ANSYS FLUENT 12.0

Theory Guide

April 2009

Contents

Using This Manual

The Contents of This Manual

The ANSYS FLUENT Theory Guide provides you with theoretical information about the models used in ANSYS FLUENT.

i Under U.S. and international copyright law, ANSYS, Inc. is unable to distribute copies of the papers listed in the bibliography, other than those published internally by ANSYS, Inc. Please use your library or a document delivery service to obtain copies of copyrighted papers.

A brief description of what is in each chapter follows:

- Chapter 1: Basic Fluid Flow, describes the governing equations and physical models used by ANSYS FLUENT to compute fluid flow (including periodic flow, swirling and rotating flows, compressible flows, and inviscid flows).
- Chapter 2: Flows with Rotating Reference Frames, describes single rotating reference frames, multiple moving reference frames, and mixing planes in ANSYS FLUENT.
- Chapter 3: Flows Using Sliding and Deforming Meshes, describes sliding and deforming meshes in ANSYS FLUENT.
- Chapter 4: Turbulence, describes ANSYS FLUENT's models for turbulent flow.
- Chapter 5: Heat Transfer, describes the physical models used by ANSYS FLUENT to compute heat transfer (including convective and conductive heat transfer, natural convection, radiative heat transfer, and periodic heat transfer).
- Chapter 6: Heat Exchangers, describes the physical models used by ANSYS FLUENT to simulate the performance of heat exchangers.
- Chapter 7: Species Transport and Finite-Rate Chemistry, describes the finite-rate chemistry models in ANSYS FLUENT. This chapter also provides information about modeling species transport in non-reacting flows.
- Chapter 8: Non-Premixed Combustion, describes the non-premixed combustion model.
- Chapter 9: Premixed Combustion, describes the premixed combustion model.

- Chapter 10: Partially Premixed Combustion, describes the partially premixed combustion model.
- Chapter 11: Composition PDF Transport, describes the composition PDF transport model.
- Chapter 12: Engine Ignition, describes the engine ignition models available in ANSYS FLUENT.
- Chapter 13: Pollutant Formation, describes the models for the formation of $\mathrm{NO_x}$, $\mathrm{SO_x}$, and soot.
- Chapter 14: Aerodynamically Generated Noise, describes the acoustics model.
- Chapter 15: Discrete Phase, describes the discrete phase models available in ANSYS FLUENT.
- Chapter 16: Multiphase Flows, describes the general multiphase models available in ANSYS FLUENT (VOF, mixture, and Eulerian).
- Chapter 17: Solidification and Melting, describes ANSYS FLUENT's model for solidification and melting.
- Chapter 18: Solver Theory, describes the ANSYS FLUENT solvers.
- Chapter 19: Adapting the Mesh, describes the solution-adaptive mesh refinement feature in ANSYS FLUENT.
- Chapter 20: Reporting Alphanumeric Data, describes how to obtain reports of fluxes, forces, surface integrals, and other solution data.

The Contents of the Other Manuals

In addition to this Theory Guide, there are several other manuals available to help you use ANSYS FLUENT and its associated programs:

- The Getting Started Guide describes the capabilities of ANSYS FLUENT, provides an overview of the problem setup steps, and presents helpful tips in order for you to create a successfull CFD simulation. The manual also includes information about accessing the ANSYS FLUENT manuals.
- The User's Guide contains detailed instructions for using ANSYS FLUENT.
- The Tutorial Guide contains a number of example problems with detailed instructions, commentary, and postprocessing of results.
- The UDF Manual contains information about writing and using user-defined functions (UDFs).
- The Text Command List provides a brief description of each of the commands in ANSYS FLUENT's text interface.

Typographical Conventions

Several typographical conventions are used in this manual's text to facilitate your learning process.

- An informational icon (*i*) marks an important note.
- Different type styles are used to indicate graphical user interface menu items and text interface menu items (e.g., Iso-Surface dialog box, `surface/iso-surface` command).
- The text interface type style is also used when illustrating exactly what appears on the screen or exactly what you need to type into a field in a dialog box. The information displayed on the screen is enclosed in a large box to distinguish it from the narrative text, and user inputs are often enclosed in smaller boxes.
- A mini flow chart is used to guide you through the navigation pane, which leads you to a specific task page or dialog box. For example,

  ✥ Models ⟶ ≣ Multiphase ⟶ Edit...

  indicates that Models is selected in the navigation pane, which then opens the corresponding task page. In the Models task page, Multiphase is selected from the list. Clicking the Edit... button opens the Multiphase dialog box.

Also, a mini flow chart is used to indicate the menu selections that lead you to a specific command or dialog box. For example,

$\boxed{\text{Define}}\longrightarrow$Injections...

indicates that the Injections... menu item can be selected from the Define pull-down menu, and

$\boxed{\texttt{display}}\longrightarrow$`mesh`

indicates that the `mesh` command is available in the `display` text menu.

In this manual, mini flow charts usually precede a description of a dialog box or command, or a screen illustration showing how to use the dialog box or command. They allow you to look up information about a command or dialog box and quickly determine how to access it without having to search the preceding material.

- The menu selections that will lead you to a particular dialog box or task page are also indicated (usually within a paragraph) using a "/". For example, Define/Materials... tells you to choose the Materials... menu item from the Define pull-down menu.

Mathematical Conventions

- Where possible, vector quantities are displayed with a raised arrow (e.g., $\vec{a}$, $\vec{A}$). Boldfaced characters are reserved for vectors and matrices as they apply to linear algebra (e.g., the identity matrix, $\mathbf{I}$).

- The operator ∇, referred to as grad, nabla, or del, represents the partial derivative of a quantity with respect to all directions in the chosen coordinate system. In Cartesian coordinates, ∇ is defined to be

$$\frac{\partial}{\partial x}\vec{\imath}+\frac{\partial}{\partial y}\vec{\jmath}+\frac{\partial}{\partial z}\vec{k}$$

∇ appears in several ways:

  - The gradient of a scalar quantity is the vector whose components are the partial derivatives; for example,

$$\nabla p=\frac{\partial p}{\partial x}\vec{\imath}+\frac{\partial p}{\partial y}\vec{\jmath}+\frac{\partial p}{\partial z}\vec{k}$$

  - The gradient of a vector quantity is a second-order tensor; for example, in Cartesian coordinates,

$$\nabla(\vec{v})=\left(\frac{\partial}{\partial x}\vec{\imath}+\frac{\partial}{\partial y}\vec{\jmath}+\frac{\partial}{\partial z}\vec{k}\right)\left(v_x\vec{\imath}+v_y\vec{\jmath}+v_z\vec{k}\right)$$

This tensor is usually written as

$$\begin{pmatrix} \frac{\partial v_x}{\partial x} & \frac{\partial v_x}{\partial y} & \frac{\partial v_x}{\partial z} \\ \\ \frac{\partial v_y}{\partial x} & \frac{\partial v_y}{\partial y} & \frac{\partial v_y}{\partial z} \\ \\ \frac{\partial v_z}{\partial x} & \frac{\partial v_z}{\partial y} & \frac{\partial v_z}{\partial z} \end{pmatrix}$$

- The divergence of a vector quantity, which is the inner product between ∇ and a vector; for example,

$$\nabla \cdot \vec{v} = \frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} + \frac{\partial v_z}{\partial z}$$

- The operator $\nabla \cdot \nabla$, which is usually written as ∇^2 and is known as the Laplacian; for example,

$$\nabla^2 T = \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2}$$

$\nabla^2 T$ is different from the expression $(\nabla T)^2$, which is defined as

$$(\nabla T)^2 = \left(\frac{\partial T}{\partial x}\right)^2 + \left(\frac{\partial T}{\partial y}\right)^2 + \left(\frac{\partial T}{\partial z}\right)^2$$

- An exception to the use of ∇ is found in the discussion of Reynolds stresses in Chapter 4: Turbulence, where convention dictates the use of Cartesian tensor notation. In this chapter, you will also find that some velocity vector components are written as u, v, and w instead of the conventional v with directional subscripts.

Technical Support

If you encounter difficulties while using ANSYS FLUENT, please first refer to the section(s) of the manual containing information on the commands you are trying to use or the type of problem you are trying to solve. The product documentation is available from the online help, or from the User Services Center (`www.fluentusers.com`).

If you encounter an error, please write down the exact error message that appeared and note as much information as you can about what you were doing in ANSYS FLUENT. Then refer to the following resources available on the User Services Center (`www.fluentusers.com`):

- Installation and System FAQs - link available from the main page on the User Services Center. The FAQs can be searched by word or phrase, and are available for general installation questions as well as for products.
- Known defects for ANSYS FLUENT - link available from the product page. The defects can be searched by word or phrase, and are listed by categories.
- Online Technical Support - link available from the main page on the User Services Center. From the Online Technical Support Portal page, there is a link to the Search Solutions & Request Support page, where the solutions can be searched by word or phrase.

Contacting Technical Support

If none of the resources available on the User Services Center help in resolving the problem, or you have complex modeling projects, we invite you to log a technical support request (`www.fluentusers.com`) to obtain further assistance. However, there are a few things that we encourage you to do before logging a request:

- Note what you are trying to accomplish with ANSYS FLUENT.
- Note what you were doing when the problem or error occurred.
- Save a journal or transcript file of the ANSYS FLUENT session in which the problem occurred. This is the best source that we can use to reproduce the problem and thereby help to identify the cause.

Chapter 1. Basic Fluid Flow

This chapter describes the theoretical background for some of the basic physical models that ANSYS FLUENT provides for fluid flow. Models for flows in moving zones (including sliding and dynamic meshes) are explained in Chapter 2: Flows with Rotating Reference Frames and Chapter 3: Flows Using Sliding and Deforming Meshes, models for turbulence are described in Chapter 4: Turbulence, and models for heat transfer (including radiation) are presented in Chapter 5: Heat Transfer. An overview of modeling species transport and reacting flows is provided in Chapter 7: Species Transport and Finite-Rate Chemistry, details about models for species transport and reacting flows are described in Chapters 7–11, and models for pollutant formation are presented in Chapter 13: Pollutant Formation. The discrete phase model is described in Chapter 15: Discrete Phase, general multiphase models are described in Chapter 16: Multiphase Flows, and the melting and solidification model is described in Chapter 17: Solidification and Melting. For information on modeling porous media, porous jumps, and lumped parameter fans and radiators, see Chapter 7: Cell Zone and Boundary Conditions in the separate User's Guide.

The information in this chapter is presented in the following sections:

1.1 Overview of Physical Models in ANSYS FLUENT

ANSYS FLUENT provides comprehensive modeling capabilities for a wide range of incompressible and compressible, laminar and turbulent fluid flow problems. Steady-state or transient analyses can be performed. In ANSYS FLUENT, a broad range of mathematical models for transport phenomena (like heat transfer and chemical reactions) is combined with the ability to model complex geometries. Examples of ANSYS FLUENT applications include laminar non-Newtonian flows in process equipment; conjugate heat transfer in turbomachinery and automotive engine components; pulverized coal combustion in utility boilers; external aerodynamics; flow through compressors, pumps, and fans; and multiphase flows in bubble columns and fluidized beds.

To permit modeling of fluid flow and related transport phenomena in industrial equipment and processes, various useful features are provided. These include porous media, lumped parameter (fan and heat exchanger), streamwise-periodic flow and heat transfer, swirl, and moving reference frame models. The moving reference frame family of models includes the ability to model single or multiple reference frames. A time-accurate sliding mesh method, useful for modeling multiple stages in turbomachinery applications, for example, is also provided, along with the mixing plane model for computing time-averaged flow fields.

Another very useful group of models in ANSYS FLUENT is the set of free surface and multiphase flow models. These can be used for analysis of gas-liquid, gas-solid, liquid-solid, and gas-liquid-solid flows. For these types of problems, ANSYS FLUENT provides the volume-of-fluid (VOF), mixture, and Eulerian models, as well as the discrete phase model (DPM). The DPM performs Lagrangian trajectory calculations for dispersed phases (particles, droplets, or bubbles), including coupling with the continuous phase. Examples of multiphase flows include channel flows, sprays, sedimentation, separation, and cavitation.

Robust and accurate turbulence models are a vital component of the ANSYS FLUENT suite of models. The turbulence models provided have a broad range of applicability, and they include the effects of other physical phenomena, such as buoyancy and compressibility. Particular care has been devoted to addressing issues of near-wall accuracy via the use of extended wall functions and zonal models.

Various modes of heat transfer can be modeled, including natural, forced, and mixed convection with or without conjugate heat transfer, porous media, etc. The set of radiation models and related submodels for modeling participating media are general and can take into account the complications of combustion. A particular strength of ANSYS FLUENT is its ability to model combustion phenomena using a variety of models, including eddy dissipation and probability density function models. A host of other models that are very useful for reacting flow applications are also available, including coal and droplet combustion, surface reaction, and pollutant formation models.

1.2 Continuity and Momentum Equations

For all flows, ANSYS FLUENT solves conservation equations for mass and momentum. For flows involving heat transfer or compressibility, an additional equation for energy conservation is solved. For flows involving species mixing or reactions, a species conservation equation is solved or, if the non-premixed combustion model is used, conservation equations for the mixture fraction and its variance are solved. Additional transport equations are also solved when the flow is turbulent.

In this section, the conservation equations for laminar flow in an inertial (non-accelerating) reference frame are presented. The equations that are applicable to rotating reference frames are presented in Chapter 2: Flows with Rotating Reference Frames. The conservation equations relevant to heat transfer, turbulence modeling, and species transport will be discussed in the chapters where those models are described.

The Euler equations solved for inviscid flow are presented in Section 1.7: Inviscid Flows.

The Mass Conservation Equation

The equation for conservation of mass, or continuity equation, can be written as follows:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{v}) = S_m \tag{1.2-1}$$

Equation 1.2-1 is the general form of the mass conservation equation and is valid for incompressible as well as compressible flows. The source S_m is the mass added to the continuous phase from the dispersed second phase (e.g., due to vaporization of liquid droplets) and any user-defined sources.

For 2D axisymmetric geometries, the continuity equation is given by

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x}(\rho v_x) + \frac{\partial}{\partial r}(\rho v_r) + \frac{\rho v_r}{r} = S_m \tag{1.2-2}$$

where x is the axial coordinate, r is the radial coordinate, v_x is the axial velocity, and v_r is the radial velocity.

Momentum Conservation Equations

Conservation of momentum in an inertial (non-accelerating) reference frame is described by [17]

$$\frac{\partial}{\partial t}(\rho\vec{v}) + \nabla\cdot(\rho\vec{v}\vec{v}) = -\nabla p + \nabla\cdot(\overline{\overline{\tau}}) + \rho\vec{g} + \vec{F} \tag{1.2-3}$$

where p is the static pressure, $\overline{\overline{\tau}}$ is the stress tensor (described below), and $\rho\vec{g}$ and $\vec{F}$ are the gravitational body force and external body forces (e.g., that arise from interaction with the dispersed phase), respectively. $\vec{F}$ also contains other model-dependent source terms such as porous-media and user-defined sources.

The stress tensor $\overline{\overline{\tau}}$ is given by

$$\overline{\overline{\tau}} = \mu\left[(\nabla\vec{v} + \nabla\vec{v}^{\,\mathrm{T}}) - \frac{2}{3}\nabla\cdot\vec{v}I\right] \tag{1.2-4}$$

where μ is the molecular viscosity, I is the unit tensor, and the second term on the right hand side is the effect of volume dilation.

For 2D axisymmetric geometries, the axial and radial momentum conservation equations are given by

$$\begin{aligned}\frac{\partial}{\partial t}(\rho v_x) + \frac{1}{r}\frac{\partial}{\partial x}(r\rho v_x v_x) + \frac{1}{r}\frac{\partial}{\partial r}(r\rho v_r v_x) = -\frac{\partial p}{\partial x} + \frac{1}{r}\frac{\partial}{\partial x}\left[r\mu\left(2\frac{\partial v_x}{\partial x} - \frac{2}{3}(\nabla\cdot\vec{v})\right)\right]\\ +\frac{1}{r}\frac{\partial}{\partial r}\left[r\mu\left(\frac{\partial v_x}{\partial r} + \frac{\partial v_r}{\partial x}\right)\right] + F_x\end{aligned} \tag{1.2-5}$$

and

$$\begin{aligned}\frac{\partial}{\partial t}(\rho v_r) + \frac{1}{r}\frac{\partial}{\partial x}(r\rho v_x v_r) + \frac{1}{r}\frac{\partial}{\partial r}(r\rho v_r v_r) = -\frac{\partial p}{\partial r} + \frac{1}{r}\frac{\partial}{\partial x}\left[r\mu\left(\frac{\partial v_r}{\partial x} + \frac{\partial v_x}{\partial r}\right)\right]\\ + \frac{1}{r}\frac{\partial}{\partial r}\left[r\mu\left(2\frac{\partial v_r}{\partial r} - \frac{2}{3}(\nabla\cdot\vec{v})\right)\right] - 2\mu\frac{v_r}{r^2} + \frac{2}{3}\frac{\mu}{r}(\nabla\cdot\vec{v}) + \rho\frac{v_z^2}{r} + F_r\end{aligned} \tag{1.2-6}$$

where

$$\nabla\cdot\vec{v} = \frac{\partial v_x}{\partial x} + \frac{\partial v_r}{\partial r} + \frac{v_r}{r} \tag{1.2-7}$$

and v_z is the swirl velocity. (See Section 1.5: Swirling and Rotating Flows for information about modeling axisymmetric swirl.)

1.3 User-Defined Scalar (UDS) Transport Equations

ANSYS FLUENT can solve the transport equation for an arbitrary, user-defined scalar (UDS) in the same way that it solves the transport equation for a scalar such as species mass fraction. Extra scalar transport equations may be needed in certain types of combustion applications or for example in plasma-enhanced surface reaction modeling.

This section provides information on how you can specify user-defined scalar (UDS) transport equations to enhance the standard features of ANSYS FLUENT. ANSYS FLUENT allows you to define additional scalar transport equations in your model in the **User-Defined Scalars** dialog box. For more information about setting up user-defined scalar transport equations in ANSYS FLUENT, see Section 9.1: User-Defined Scalar (UDS) Transport Equations in the separate User's Guide.

Information in this section is organized in the following subsections:

- Section 1.3.1: Single Phase Flow
- Section 1.3.2: Multiphase Flow

1.3.1 Single Phase Flow

For an arbitrary scalar ϕ_k, ANSYS FLUENT solves the equation

$$\frac{\partial \rho \phi_k}{\partial t} + \frac{\partial}{\partial x_i}(\rho u_i \phi_k - \Gamma_k \frac{\partial \phi_k}{\partial x_i}) = S_{\phi_k} \quad k = 1, ..., N \tag{1.3-1}$$

where Γ_k and S_{ϕ_k} are the diffusion coefficient and source term supplied by you for each of the N scalar equations. Note that $\Gamma_k$is defined as a tensor in the case of anisotropic diffusivity. The diffusion term is thus $\nabla \cdot (\mathbf{\Gamma_k} \cdot \phi_k)$

For isotropic diffusivity, Γ_k could be written as $\Gamma_k I$ where I is the identity matrix.

For the steady-state case, ANSYS FLUENT will solve one of the three following equations, depending on the method used to compute the convective flux:

- If convective flux is *not* to be computed, ANSYS FLUENT will solve the equation

  $$-\frac{\partial}{\partial x_i}(\Gamma_k \frac{\partial \phi_k}{\partial x_i}) = S_{\phi_k} \quad k = 1, ..., N \tag{1.3-2}$$

  where Γ_k and S_{ϕ_k} are the diffusion coefficient and source term supplied by you for each of the N scalar equations.

- If convective flux is to be computed with mass flow rate, ANSYS FLUENT will solve the equation

$$\frac{\partial}{\partial x_i}(\rho u_i \phi_k - \Gamma_k \frac{\partial \phi_k}{\partial x_i}) = S_{\phi_k} \quad k = 1, ..., N \tag{1.3-3}$$

- It is also possible to specify a user-defined function to be used in the computation of convective flux. In this case, the user-defined mass flux is assumed to be of the form

$$F = \int_S \rho \vec{u} \cdot d\vec{S} \tag{1.3-4}$$

where $d\vec{S}$ is the face vector area.

1.3.2 Multiphase Flow

For multiphase flows, ANSYS FLUENT solves transport equations for two types of scalars: *per phase* and *mixture*. For an arbitrary k scalar in *phase-1*, denoted by ϕ_l^k, ANSYS FLUENT solves the transport equation inside the volume occupied by *phase-l*

$$\frac{\partial \alpha_l \rho_l \phi_l^k}{\partial t} + \nabla \cdot (\alpha_l \rho_l \vec{u}_l \phi_l^k - \alpha_l \Gamma_l^k \nabla \phi_l^k) = S_l^k \quad k = 1, ..., N \tag{1.3-5}$$

where α_l, ρ_l, and $\vec{u}_l$ are the volume fraction, physical density, and velocity of *phase-l*, respectively. Γ_l^k and S_l^k are the diffusion coefficient and source term, respectively, which you will need to specify. In this case, scalar ϕ_l^k is associated only with one phase (*phase-l*) and is considered an individual field variable of *phase-l*.

The mass flux for *phase-l* is defined as

$$F_l = \int_S \alpha_l \rho_l \vec{u}_l \cdot d\vec{S} \tag{1.3-6}$$

If the transport variable described by scalar ϕ_l^k represents the physical field that is shared between phases, or is considered the same for each phase, then you should consider this scalar as being associated with a mixture of phases, ϕ^k. In this case, the generic transport equation for the scalar is

$$\frac{\partial \rho_m \phi^k}{\partial t} + \nabla \cdot (\rho_m \vec{u}_m \phi^k - \Gamma_m^k \nabla \phi^k) = S^{k_m} \quad k = 1, ..., N \tag{1.3-7}$$

where mixture density ρ_m, mixture velocity $\vec{u}_m$, and mixture diffusivity for the scalar k Γ_m^k are calculated according to

$$\rho_m = \sum_l \alpha_l \rho_l \tag{1.3-8}$$

$$\rho_m \vec{u}_m = \sum_l \alpha_l \rho_l \vec{u}_l \tag{1.3-9}$$

$$F_m = \int_S rho_m \vec{u}_m \cdot d\vec{S} \tag{1.3-10}$$

$$\Gamma_m^k = \sum_l \alpha_l \Gamma_l^k \tag{1.3-11}$$

$$S_m^k = \sum_l S_l^k \tag{1.3-12}$$

To calculate mixture diffusivity, you will need to specify individual diffusivities for each material associated with individual phases.

Note that if the user-defined mass flux option is activated, then mass fluxes shown in Equation 1.3-6 and Equation 1.3-10 will need to be replaced in the corresponding scalar transport equations.

1.4 Periodic Flows

Periodic flow occurs when the physical geometry of interest and the expected pattern of the flow/thermal solution have a periodically repeating nature. Two types of periodic flow can be modeled in ANSYS FLUENT. In the first type, no pressure drop occurs across the periodic planes. In the second type, a pressure drop occurs across translationally periodic boundaries, resulting in “fully-developed” or “streamwise-periodic” flow.

This section discusses streamwise-periodic flow. A description of no-pressure-drop periodic flow is provided in Section 7.3.16: Periodic Boundary Conditions in the separate User’s Guide, and a description of streamwise-periodic heat transfer is provided in Section 13.4: Modeling Periodic Heat Transfer in the separate User’s Guide. For more information about setting up periodic flows in ANSYS FLUENT, see Section 9.2: Periodic Flows in the separate User’s Guide.

Information about streamwise-periodic flow is presented in the following sections:

- Section 1.4.1: Overview
- Section 1.4.2: Limitations
- Section 1.4.3: Physics of Periodic Flows

1.4.1 Overview

ANSYS FLUENT provides the ability to calculate streamwise-periodic—or "fully-developed"—fluid flow. These flows are encountered in a variety of applications, including flows in compact heat exchanger channels and flows across tube banks. In such flow configurations, the geometry varies in a repeating manner along the direction of the flow, leading to a periodic fully-developed flow regime in which the flow pattern repeats in successive cycles. Other examples of streamwise-periodic flows include fully-developed flow in pipes and ducts. These periodic conditions are achieved after a sufficient entrance length, which depends on the flow Reynolds number and geometric configuration.

Streamwise-periodic flow conditions exist when the flow pattern repeats over some length L, with a constant pressure drop across each repeating module along the streamwise direction. Figure 1.4.1 depicts one example of a periodically repeating flow of this type which has been modeled by including a single representative module.

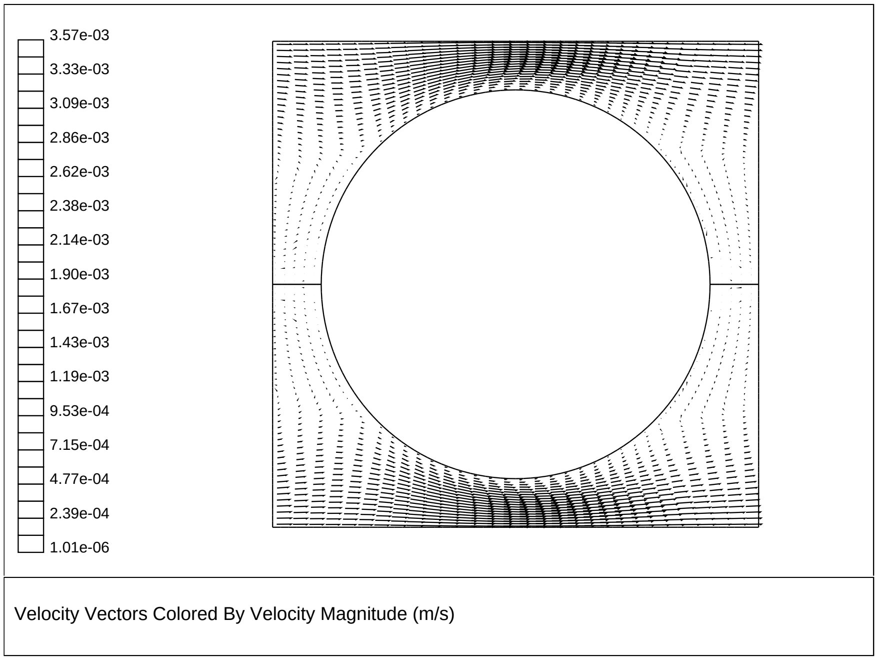


Figure 1.4.1: Example of Periodic Flow in a 2D Heat Exchanger Geometry

1.4.2 Limitations

The following limitations apply to modeling streamwise-periodic flow:

- The flow must be incompressible.
- The geometry must be translationally periodic. Note that transient simulations for fully-developed fluid flow are not valid with translational periodic flow.
- If one of the density-based solvers is used, you can specify only the pressure jump; for the pressure-based solver, you can specify either the pressure jump or the mass flow rate.
- No net mass addition through inlets/exits or extra source terms is allowed.
- Species can be modeled only if inlets/exits (without net mass addition) are included in the problem. Reacting flows are not permitted.
- Discrete phase and multiphase modeling are not allowed.

1.4.3 Physics of Periodic Flows

Definition of the Periodic Velocity

The assumption of periodicity implies that the velocity components repeat themselves in space as follows:

$$\begin{aligned} u(\vec{r}) &= u(\vec{r}+\vec{L}) = u(\vec{r}+2\vec{L}) = \cdots \\ v(\vec{r}) &= v(\vec{r}+\vec{L}) = v(\vec{r}+2\vec{L}) = \cdots \\ w(\vec{r}) &= w(\vec{r}+\vec{L}) = w(\vec{r}+2\vec{L}) = \cdots \end{aligned} \tag{1.4-1}$$

where $\vec{r}$ is the position vector and $\vec{L}$ is the periodic length vector of the domain considered (see Figure 1.4.2).

Definition of the Streamwise-Periodic Pressure

For viscous flows, the pressure is not periodic in the sense of Equation 1.4-1. Instead, the *pressure drop* between modules is periodic:

$$\Delta p = p(\vec{r}) - p(\vec{r}+\vec{L}) = p(\vec{r}+\vec{L}) - p(\vec{r}+2\vec{L}) = \cdots \tag{1.4-2}$$

If one of the density-based solvers is used, Δp is specified as a constant value. For the pressure-based solver, the local pressure gradient can be decomposed into two parts:

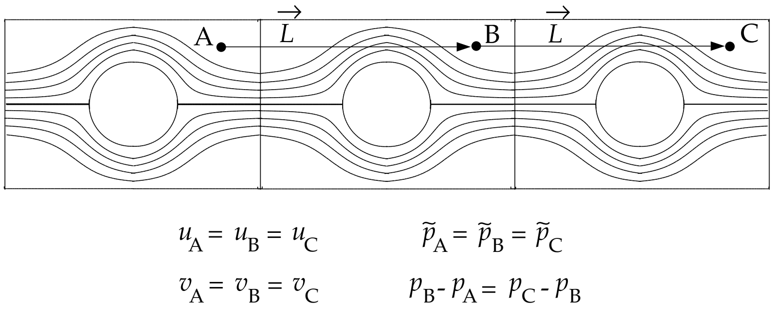


Figure 1.4.2: Example of a Periodic Geometry

the gradient of a periodic component, $\nabla\tilde{p}(\vec{r})$, and the gradient of a linearly-varying component, $\beta\frac{\vec{L}}{|\vec{L}|}$:

$$\nabla p(\vec{r}) = \beta\frac{\vec{L}}{|\vec{L}|} + \nabla\tilde{p}(\vec{r}) \tag{1.4-3}$$

where $\tilde{p}(\vec{r})$ is the periodic pressure and $\beta|\vec{r}|$ is the linearly-varying component of the pressure. The periodic pressure is the pressure left over after subtracting out the linearly-varying pressure. The linearly-varying component of the pressure results in a force acting on the fluid in the momentum equations. Because the value of β is not known a priori, it must be iterated on until the mass flow rate that you have defined is achieved in the computational model. This correction of β occurs in the pressure correction step of the SIMPLE, SIMPLEC, or PISO algorithm where the value of β is updated based on the difference between the desired mass flow rate and the actual one. You have some control over the number of sub-iterations used to update β. For more information about setting up parameters for β in ANSYS FLUENT, see Section 9.2.2: Setting Parameters for the Calculation of β in the separate User's Guide.

1.5 Swirling and Rotating Flows

Many important engineering flows involve swirl or rotation and ANSYS FLUENT is well-equipped to model such flows. Swirling flows are common in combustion, with swirl introduced in burners and combustors in order to increase residence time and stabilize the flow pattern. Rotating flows are also encountered in turbomachinery, mixing tanks, and a variety of other applications.

When you begin the analysis of a rotating or swirling flow, it is essential that you classify your problem into one of the following five categories of flow:

- axisymmetric flows with swirl or rotation
- fully three-dimensional swirling or rotating flows
- flows requiring a rotating reference frame
- flows requiring multiple rotating reference frames or mixing planes
- flows requiring sliding meshes

Modeling and solution procedures for the first two categories are presented in this section. The remaining three, which all involve "moving zones", are discussed in Chapter 2: Flows with Rotating Reference Frames.

Information about rotating and swirling flows is provided in the following subsections:

- Section 1.5.1: Overview of Swirling and Rotating Flows
- Section 1.5.2: Physics of Swirling and Rotating Flows

For more information about setting up swirling and rotating flows in ANSYS FLUENT, see Section 9.3: Swirling and Rotating Flows in the separate User's Guide.

1.5.1 Overview of Swirling and Rotating Flows

Axisymmetric Flows with Swirl or Rotation

As discussed in Section 1.5.1: Overview of Swirling and Rotating Flows, you can solve a 2D axisymmetric problem that includes the prediction of the circumferential or swirl velocity. The assumption of axisymmetry implies that there are no circumferential gradients in the flow, but that there may be non-zero circumferential velocities. Examples of axisymmetric flows involving swirl or rotation are depicted in Figures 1.5.1 and 1.5.2.

Your problem may be axisymmetric with respect to geometry and flow conditions but still include swirl or rotation. In this case, you can model the flow in 2D (i.e., solve

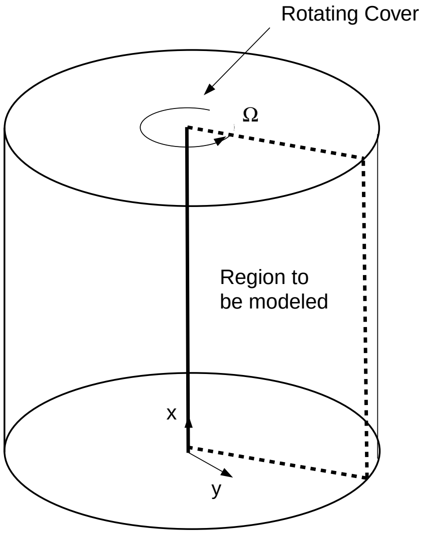


Figure 1.5.1: Rotating Flow in a Cavity

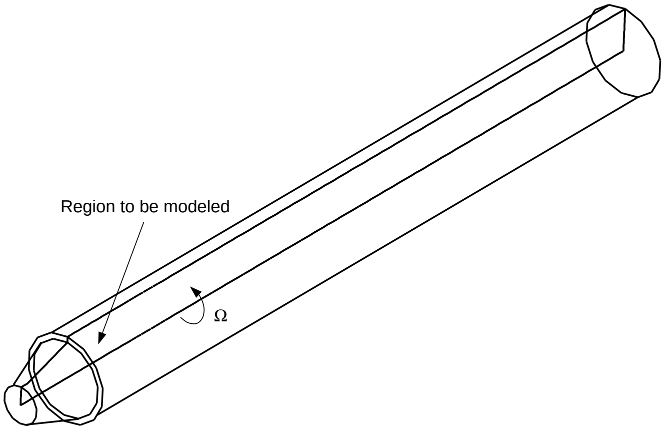


Figure 1.5.2: Swirling Flow in a Gas Burner

the axisymmetric problem) and include the prediction of the circumferential (or swirl) velocity. It is important to note that while the assumption of axisymmetry implies that there are no circumferential gradients in the flow, there may still be non-zero swirl velocities.

Momentum Conservation Equation for Swirl Velocity

The tangential momentum equation for 2D swirling flows may be written as

$$\frac{\partial}{\partial t}(\rho w)+\frac{1}{r}\frac{\partial}{\partial x}(r\rho u w)+\frac{1}{r}\frac{\partial}{\partial r}(r\rho v w) = \frac{1}{r}\frac{\partial}{\partial x}\left[r\mu\frac{\partial w}{\partial x}\right]+\frac{1}{r^2}\frac{\partial}{\partial r}\left[r^3\mu\frac{\partial}{\partial r}\left(\frac{w}{r}\right)\right]-\rho\frac{vw}{r} \tag{1.5-1}$$

where x is the axial coordinate, r is the radial coordinate, u is the axial velocity, v is the radial velocity, and w is the swirl velocity.

Three-Dimensional Swirling Flows

When there are geometric changes and/or flow gradients in the circumferential direction, your swirling flow prediction requires a three-dimensional model. If you are planning a 3D ANSYS FLUENT model that includes swirl or rotation, you should be aware of the setup constraints (Section 9.3.3: Coordinate System Restrictions in the separate User's Guide). In addition, you may wish to consider simplifications to the problem which might reduce it to an equivalent axisymmetric problem, especially for your initial modeling effort. Because of the complexity of swirling flows, an initial 2D study, in which you can quickly determine the effects of various modeling and design choices, can be very beneficial.

> For 3D problems involving swirl or rotation, there are no special inputs required during the problem setup and no special solution procedures. Note, however, that you may want to use the cylindrical coordinate system for defining velocity-inlet boundary condition inputs, as described in Section 7.3.4: Defining the Velocity in the separate User's Guide. Also, you may find the gradual increase of the rotational speed (set as a wall or inlet boundary condition) helpful during the solution process. For more information, see Section 9.3.4: Improving Solution Stability by Gradually Increasing the Rotational or Swirl Speed in the separate User's Guide.

Flows Requiring a Rotating Reference Frame

If your flow involves a rotating boundary which moves through the fluid (e.g., an impeller blade or a grooved or notched surface), you will need to use a rotating reference frame to model the problem. Such applications are described in detail in Section 2.2: Flow in a Rotating Reference Frame. If you have more than one rotating boundary (e.g., several impellers in a row), you can use multiple reference frames (described in Section 2.3.1: The Multiple Reference Frame Model) or mixing planes (described in Section 2.3.2: The Mixing Plane Model).

1.5.2 Physics of Swirling and Rotating Flows

In swirling flows, conservation of angular momentum (rw or $r^2\Omega$ = constant) tends to create a free vortex flow, in which the circumferential velocity, w, increases sharply as the radius, r, decreases (with w finally decaying to zero near $r = 0$ as viscous forces begin to dominate). A tornado is one example of a free vortex. Figure 1.5.3 depicts the radial distribution of w in a typical free vortex.

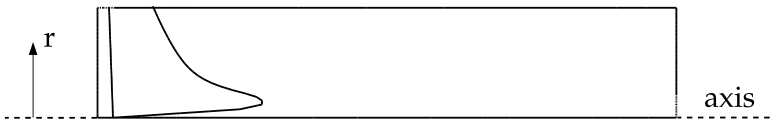


Figure 1.5.3: Typical Radial Distribution of w in a Free Vortex

It can be shown that for an ideal free vortex flow, the centrifugal forces created by the circumferential motion are in equilibrium with the radial pressure gradient:

$$\frac{\partial p}{\partial r} = \frac{\rho w^2}{r} \tag{1.5-2}$$

As the distribution of angular momentum in a non-ideal vortex evolves, the form of this radial pressure gradient also changes, driving radial and axial flows in response to the highly non-uniform pressures that result. Thus, as you compute the distribution of swirl in your ANSYS FLUENT model, you will also notice changes in the static pressure distribution and corresponding changes in the axial and radial flow velocities. It is this high degree of coupling between the swirl and the pressure field that makes the modeling of swirling flows complex.

In flows that are driven by wall rotation, the motion of the wall tends to impart a forced vortex motion to the fluid, wherein w/r or Ω is constant. An important characteristic of such flows is the tendency of fluid with high angular momentum (e.g., the flow near the wall) to be flung radially outward (Figure 1.5.4). This is often referred to as "radial pumping", since the rotating wall is pumping the fluid radially outward.

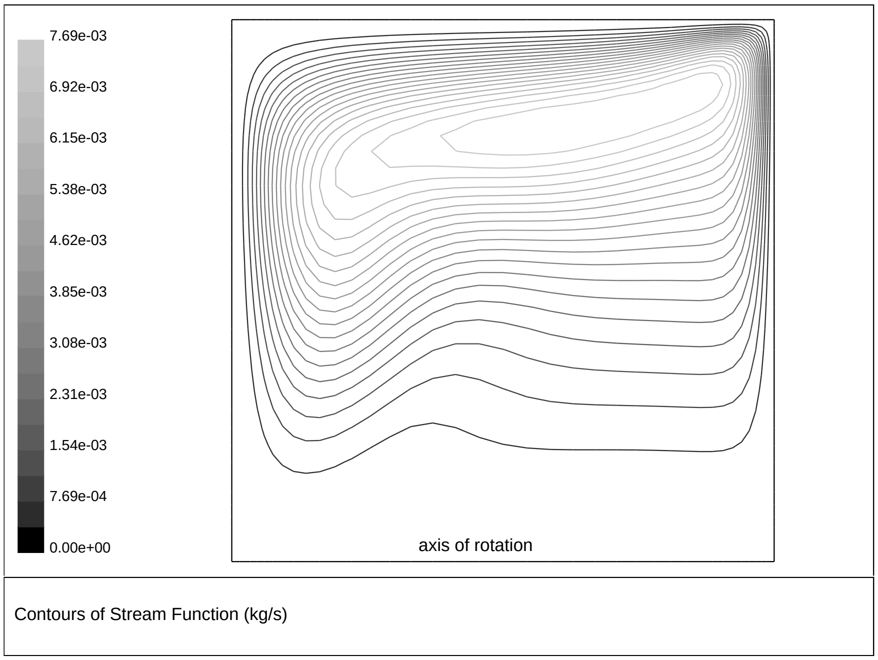


Figure 1.5.4: Stream Function Contours for Rotating Flow in a Cavity (Geometry of Figure 1.5.1)

1.6 Compressible Flows

Compressibility effects are encountered in gas flows at high velocity and/or in which there are large pressure variations. When the flow velocity approaches or exceeds the speed of sound of the gas or when the pressure change in the system ($\Delta p/p$) is large, the variation of the gas density with pressure has a significant impact on the flow velocity, pressure, and temperature. Compressible flows create a unique set of flow physics for which you must be aware of the special input requirements and solution techniques described in this section. Figures 1.6.1 and 1.6.2 show examples of compressible flows computed using ANSYS FLUENT.

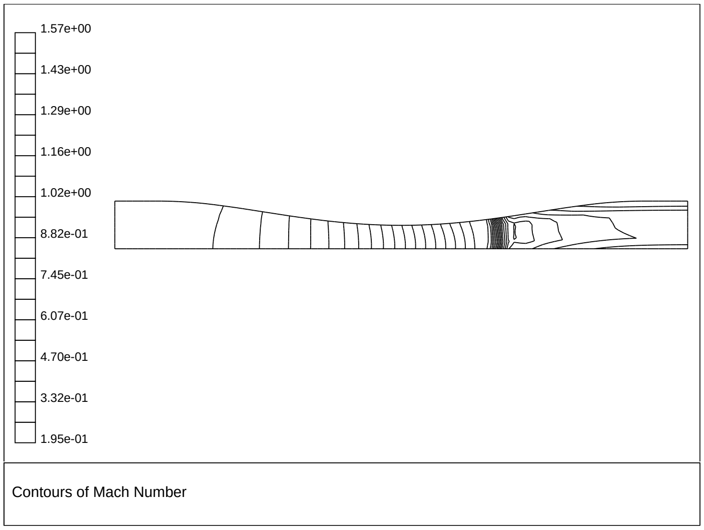


Figure 1.6.1: Transonic Flow in a Converging-Diverging Nozzle

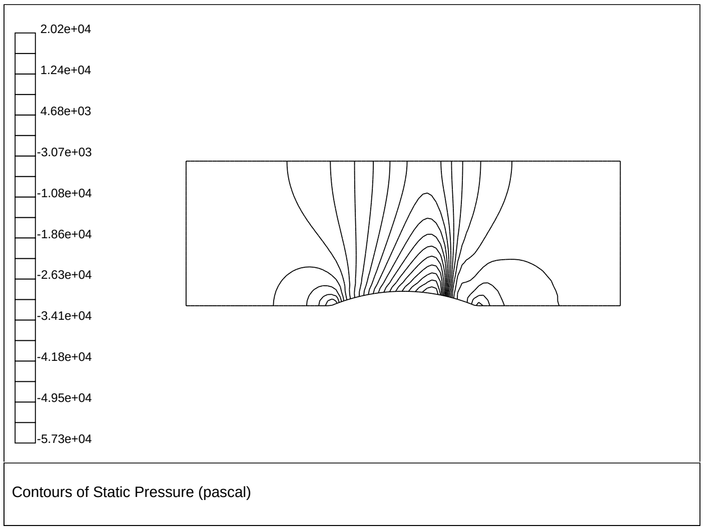


Figure 1.6.2: Mach 0.675 Flow Over a Bump in a 2D Channel

For more information about setting up compressible flows in ANSYS FLUENT, see Section 9.4: Compressible Flows in the separate User's Guide.

Information about compressible flows is provided in the following subsections:

- Section 1.6.1: When to Use the Compressible Flow Model
- Section 1.6.2: Physics of Compressible Flows

1.6.1 When to Use the Compressible Flow Model

Compressible flows can be characterized by the value of the Mach number:

$$\mathrm{M} \equiv u/c \tag{1.6-1}$$

Here, c is the speed of sound in the gas:

$$c = \sqrt{\gamma R T} \tag{1.6-2}$$

and γ is the ratio of specific heats (c_p/c_v).

When the Mach number is less than 1.0, the flow is termed subsonic. At Mach numbers much less than 1.0 (M < 0.1 or so), compressibility effects are negligible and the variation of the gas density with pressure can safely be ignored in your flow modeling. As the Mach number approaches 1.0 (which is referred to as the transonic flow regime), compressibility effects become important. When the Mach number exceeds 1.0, the flow is termed supersonic, and may contain shocks and expansion fans which can impact the flow pattern significantly. ANSYS FLUENT provides a wide range of compressible flow modeling capabilities for subsonic, transonic, and supersonic flows.

1.6.2 Physics of Compressible Flows

Compressible flows are typically characterized by the total pressure p_0 and total temperature T_0 of the flow. For an ideal gas, these quantities can be related to the static pressure and temperature by the following:

$$\frac{p_0}{p} = exp(\frac{\int_T^{T_0} \frac{C_p}{T} dT}{R}) \tag{1.6-3}$$

For constant C$_p$, Equation 1.6-3 reduces to

$$\frac{p_0}{p} = \left(1 + \frac{\gamma - 1}{2}\mathrm{M}^2\right)^{\gamma/(\gamma-1)} \tag{1.6-4}$$

$$\frac{T_0}{T} = 1 + \frac{\gamma - 1}{2}\mathrm{M}^2 \tag{1.6-5}$$

These relationships describe the variation of the static pressure and temperature in the flow as the velocity (Mach number) changes under isentropic conditions. For example, given a pressure ratio from inlet to exit (total to static), Equation 1.6-4 can be used to estimate the exit Mach number which would exist in a one-dimensional isentropic flow. For air, Equation 1.6-4 predicts a choked flow (Mach number of 1.0) at an isentropic pressure ratio, p/p_0, of 0.5283. This choked flow condition will be established at the point of minimum flow area (e.g., in the throat of a nozzle). In the subsequent area expansion the flow may either accelerate to a supersonic flow in which the pressure will continue to drop, or return to subsonic flow conditions, decelerating with a pressure rise. If a supersonic flow is exposed to an imposed pressure increase, a shock will occur, with a sudden pressure rise and deceleration accomplished across the shock.

Basic Equations for Compressible Flows

Compressible flows are described by the standard continuity and momentum equations solved by ANSYS FLUENT, and you do not need to activate any special physical models (other than the compressible treatment of density as detailed below). The energy equation solved by ANSYS FLUENT correctly incorporates the coupling between the flow velocity and the static temperature, and should be activated whenever you are solving a compressible flow. In addition, if you are using the pressure-based solver, you should activate the viscous dissipation terms in Equation 5.2-1, which become important in high-Mach-number flows.

The Compressible Form of the Gas Law

For compressible flows, the ideal gas law is written in the following form:

$$\rho = \frac{p_{\rm op} + p}{\frac{R}{M_w} T} \tag{1.6-6}$$

where $p_{\rm op}$ is the operating pressure defined in the **Operating Conditions** dialog box, p is the local static pressure relative to the operating pressure, R is the universal gas constant, and M_w is the molecular weight. The temperature, T, will be computed from the energy equation.

1.7 Inviscid Flows

Inviscid flow analyses neglect the effect of viscosity on the flow and are appropriate for high-Reynolds-number applications where inertial forces tend to dominate viscous forces. One example for which an inviscid flow calculation is appropriate is an aerodynamic analysis of some high-speed projectile. In a case like this, the pressure forces on the body will dominate the viscous forces. Hence, an inviscid analysis will give you a quick estimate of the primary forces acting on the body. After the body shape has been modified to maximize the lift forces and minimize the drag forces, you can perform a viscous analysis to include the effects of the fluid viscosity and turbulent viscosity on the lift and drag forces.

Another area where inviscid flow analyses are routinely used is to provide a good initial solution for problems involving complicated flow physics and/or complicated flow geometry. In a case like this, the viscous forces are important, but in the early stages of the calculation the viscous terms in the momentum equations will be ignored. Once the calculation has been started and the residuals are decreasing, you can turn on the viscous terms (by enabling laminar or turbulent flow) and continue the solution to convergence. For some very complicated flows, this is the only way to get the calculation started.

For more information about setting up inviscid flows in ANSYS FLUENT, see Section 9.5: Inviscid Flows in the separate User's Guide.

Information about inviscid flows is provided in the following subsections:

- Section 1.7.1: Euler Equations

1.7.1 Euler Equations

For inviscid flows, ANSYS FLUENT solves the Euler equations. The mass conservation equation is the same as for a laminar flow, but the momentum and energy conservation equations are reduced due to the absence of molecular diffusion.

In this section, the conservation equations for inviscid flow in an inertial (non-rotating) reference frame are presented. The equations that are applicable to non-inertial reference frames are described in Chapter 2: Flows with Rotating Reference Frames. The conservation equations relevant for species transport and other models will be discussed in the chapters where those models are described.

The Mass Conservation Equation

The equation for conservation of mass, or continuity equation, can be written as follows:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{v}) = S_m \qquad (1.7\text{-}1)$$

Equation 1.7-1 is the general form of the mass conservation equation and is valid for incompressible as well as compressible flows. The source S_m is the mass added to the continuous phase from the dispersed second phase (e.g., due to vaporization of liquid droplets) and any user-defined sources.

For 2D axisymmetric geometries, the continuity equation is given by

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x}(\rho v_x) + \frac{\partial}{\partial r}(\rho v_r) + \frac{\rho v_r}{r} = S_m \qquad (1.7\text{-}2)$$

where x is the axial coordinate, r is the radial coordinate, v_x is the axial velocity, and v_r is the radial velocity.

Momentum Conservation Equations

Conservation of momentum is described by

$$\frac{\partial}{\partial t}(\rho \vec{v}) + \nabla \cdot (\rho \vec{v}\vec{v}) = -\nabla p + \rho \vec{g} + \vec{F} \qquad (1.7\text{-}3)$$

where p is the static pressure and $\rho \vec{g}$ and $\vec{F}$ are the gravitational body force and external body forces (e.g., forces that arise from interaction with the dispersed phase), respectively. $\vec{F}$ also contains other model-dependent source terms such as porous-media and user-defined sources.

For 2D axisymmetric geometries, the axial and radial momentum conservation equations are given by

$$\frac{\partial}{\partial t}(\rho v_x) + \frac{1}{r}\frac{\partial}{\partial x}(r\rho v_x v_x) + \frac{1}{r}\frac{\partial}{\partial r}(r\rho v_r v_x) = -\frac{\partial p}{\partial x} + F_x \tag{1.7-4}$$

and

$$\frac{\partial}{\partial t}(\rho v_r) + \frac{1}{r}\frac{\partial}{\partial x}(r\rho v_x v_r) + \frac{1}{r}\frac{\partial}{\partial r}(r\rho v_r v_r) = -\frac{\partial p}{\partial r} + F_r \tag{1.7-5}$$

where

$$\nabla \cdot \vec{v} = \frac{\partial v_x}{\partial x} + \frac{\partial v_r}{\partial r} + \frac{v_r}{r} \tag{1.7-6}$$

Energy Conservation Equation

Conservation of energy is described by

$$\frac{\partial}{\partial t}(\rho E) + \nabla \cdot (\vec{v}(\rho E + p)) = -\nabla \cdot \left(\sum_j h_j J_j \right) + S_h \tag{1.7-7}$$

Chapter 2. Flows with Rotating Reference Frames

This chapter describes the theoretical background modeling flows in rotating reference frames. The information in this chapter is presented in the following sections:

- Section 2.1: Introduction
- Section 2.2: Flow in a Rotating Reference Frame
- Section 2.3: Flow in Multiple Rotating Reference Frames

2.1 Introduction

ANSYS FLUENT solves the equations of fluid flow and heat transfer, by default, in a stationary (or inertial) reference frame. However, there are many problems where it is advantageous to solve the equations in a moving (or non-inertial) reference frame. Such problems typically involve moving parts (such as rotating blades, impellers, and similar types of moving surfaces), and it is the flow around these moving parts that is of interest. In most cases, the moving parts render the problem unsteady when viewed from the stationary frame. With a moving reference frame, however, the flow around the moving part can (with certain restrictions) be modeled as a steady-state problem with respect to the moving frame.

ANSYS FLUENT's moving reference frame modeling capability allows you to model problems involving moving parts by allowing you to activate moving reference frames in selected cell zones. When a moving reference frame is activated, the equations of motion are modified to incorporate the additional acceleration terms which occur due to the transformation from the stationary to the moving reference frame. By solving these equations in a steady-state manner, the flow around the moving parts can be modeled.

For many problems, it may be possible to refer the entire computational domain to a single moving reference frame. This is known as the single reference frame (or SRF) approach. The use of the SRF approach is possible; provided the geometry meets certain requirements (as discussed in Section 2.2: Flow in a Rotating Reference Frame). For more complex geometries, it may not be possible to use a single reference frame. In such cases, you must break up the problem into multiple cell zones, with well-defined interfaces between the zones. The manner in which the interfaces are treated leads to two approximate, steady-state modeling methods for this class of problem: the multiple reference frame (or MRF) approach, and the mixing plane approach. These approaches will be discussed in Sections 2.3.1 and 2.3.2. If unsteady interaction between the stationary and moving parts is important, you can employ the Sliding Mesh approach to capture the transient behavior of the flow. The sliding meshing model will be discussed in Chapter 3: Flows Using Sliding and Deforming Meshes.

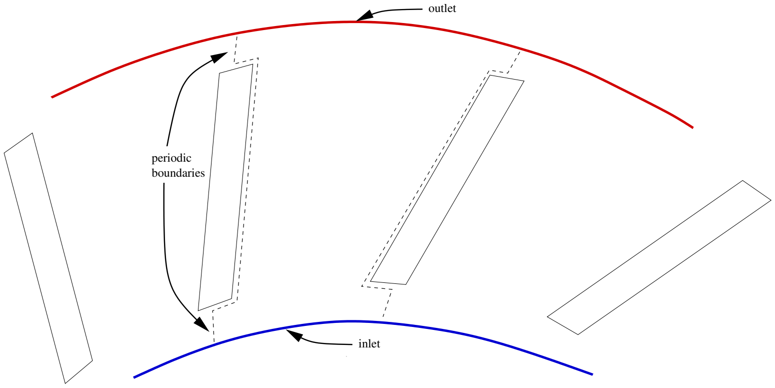


Figure 2.1.1: Single Component (Blower Wheel Blade Passage)

2.2 Flow in a Rotating Reference Frame

The principal reason for employing a moving reference frame is to render a problem which is unsteady in the stationary (inertial) frame steady with respect to the moving frame. For a steadily rotating frame (i.e., the rotational speed is constant), it is possible to transform the equations of fluid motion to the rotating frame such that steady-state solutions are possible. By default, ANSYS FLUENT permits the activation of a moving reference frame with a steady rotational speed. If the rotational speed is not constant, the transformed equations will contain additional terms which are not included in ANSYS FLUENT's formulation (although they can be added as source terms using user-defined functions). It should also be noted that you can run an unsteady simulation in a moving reference frame with constant rotational speed. This would be necessary if you wanted to simulate, for example, vortex shedding from a rotating fan blade. The unsteadiness in this case is due to a natural fluid instability (vortex generation) rather than induced from interaction with a stationary component.

The information in this section is presented in the following:

- Section 2.2.1: Equations for a Rotating Reference Frame
- Section 2.2.2: Single Rotating Reference Frame (SRF) Modeling

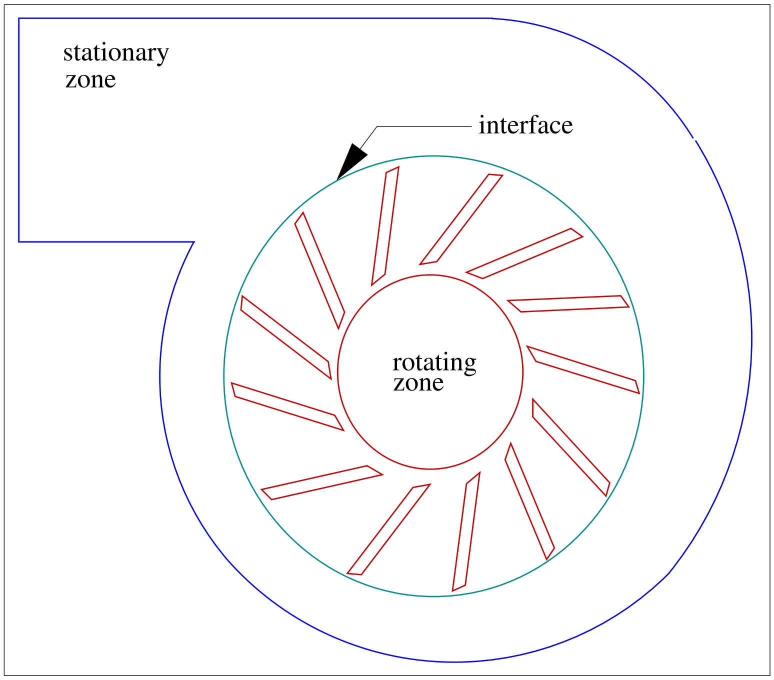


Figure 2.1.2: Multiple Component (Blower Wheel and Casing)

2.2.1 Equations for a Rotating Reference Frame

Consider a coordinate system which is rotating steadily with angular velocity $\vec{\omega}$ relative to a stationary (inertial) reference frame, as illustrated in Figure 2.2.1. The origin of the rotating system is located by a position vector $\vec{r_0}$.

The axis of rotation is defined by a unit direction vector $\hat{a}$ such that

$$\vec{\omega} = \omega\hat{a} \tag{2.2-1}$$

The computational domain for the CFD problem is defined with respect to the rotating frame such that an arbitrary point in the CFD domain is located by a position vector $\vec{r}$ from the origin of the rotating frame.

The fluid velocities can be transformed from the stationary frame to the rotating frame using the following relation:

$$\vec{v}_r = \vec{v} - \vec{u}_r \tag{2.2-2}$$

where

$$\vec{u}_r = \vec{\omega} \times \vec{r} \tag{2.2-3}$$

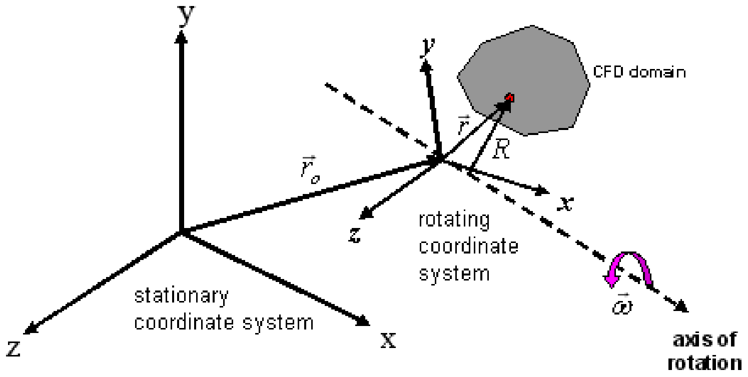


Figure 2.2.1: Stationary and Rotating Reference Frames

In the above, $\vec{v}_r$ is the relative velocity (the velocity viewed from the rotating frame), $\vec{v}$ is the absolute velocity (the velocity viewed from the stationary frame), and $\vec{u}_r$ is the "whirl" velocity (the velocity due to the moving frame).

When the equations of motion are solved in the rotating reference frame, the acceleration of the fluid is augmented by additional terms that appear in the momentum equations [17]. Moreover, the equations can be formulated in two different ways:

- Expressing the momentum equations using the relative velocities as dependent variables (known as the relative velocity formulation).
- Expressing the momentum equations using the absolute velocities as dependent variables in the momentum equations (known as the absolute velocity formulation).

The exact forms of the governing equations for these two formulations will be provided in the sections below. It can be noted here that ANSYS FLUENT's pressure-based solvers provide the option to use either of these two formulations, whereas the density-based solvers always use the absolute velocity formulation. For more information about the advantages of each velocity formulation, see Section 10.7.1: Choosing the Relative or Absolute Velocity Formulation (in the separate User's Guide).

Relative Velocity Formulation

For the relative velocity formulation, the governing equations of fluid flow for a steadily rotating frame can be written as follows:

Conservation of mass:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \rho \vec{v}_r = 0 \tag{2.2-4}$$

Conservation of momentum:

$$\frac{\partial}{\partial t}(\rho \vec{v}_r) + \nabla \cdot (\rho \vec{v}_r \vec{v}_r) + \rho(2\vec{\omega} \times \vec{v}_r + \vec{\omega} \times \vec{\omega} \times \vec{r}) = -\nabla p + \nabla \cdot \overline{\overline{\tau}}_r + \vec{F} \tag{2.2-5}$$

Conservation of energy:

$$\frac{\partial}{\partial t}(\rho E_r) + \nabla \cdot (\rho \vec{v}_r H_r) = \nabla \cdot (k \nabla T + \overline{\overline{\tau}}_r \cdot \vec{v}_r) + S_h \tag{2.2-6}$$

The momentum equation contains two additional acceleration terms: the Coriolis acceleration ($2\vec{\omega} \times \vec{v}_r$), and the centripetal acceleration ($\vec{\omega} \times \vec{\omega} \times \vec{r}$). In addition, the viscous stress ($\overline{\overline{\tau}}_r$) is identical to Equation 1.2-4 except that relative velocity derivatives are used. The energy equation is written in terms of the relative internal energy (E_r) and the relative total enthalpy (H_r), also known as the rothalpy. These variables are defined as:

$$E_r = h - \frac{p}{\rho} + \frac{1}{2}({v_r}^2 - {u_r}^2) \tag{2.2-7}$$

$$H_r = E_r + \frac{p}{\rho} \tag{2.2-8}$$

Absolute Velocity Formulation

For the absolute velocity formulation, the governing equations of fluid flow for a steadily rotating frame can be written as follows:

Conservation of mass:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \rho \vec{v}_r = 0 \tag{2.2-9}$$

Conservation of momentum:

$$\frac{\partial}{\partial t}\rho \vec{v} + \nabla \cdot (\rho \vec{v}_r \vec{v}) + \rho(\vec{\omega} \times \vec{v}) = -\nabla p + \nabla \cdot \overline{\overline{\tau}} + \vec{F} \tag{2.2-10}$$

Conservation of energy:

$$\frac{\partial}{\partial t}\rho E + \nabla \cdot (\rho \vec{v}_r H + p\vec{u}_r) = \nabla \cdot (k\nabla T + \overline{\overline{\tau}} \cdot \vec{v}) + S_h \qquad (2.2\text{-}11)$$

In this formulation, the Coriolis and centripetal accelerations can be collapsed into a single term ($\vec{\omega} \times \vec{v}$).

2.2.2 Single Rotating Reference Frame (SRF) Modeling

Many problems permit the entire computational domain to be referred to a single rotating reference frame (hence the name SRF modeling). In such cases, the equations given in Section 2.2.1: Equations for a Rotating Reference Frame are solved in all fluid cell zones. Steady-state solutions are possible in SRF models provided suitable boundary conditions are prescribed. In particular, wall boundaries must adhere to the following requirements:

- Any walls which are moving with the reference frame can assume any shape. An example would be the blade surfaces associated with a pump impeller. The no slip condition is defined in the relative frame such that the relative velocity is zero on the moving walls.
- Walls can be defined which are non-moving with respect to the stationary coordinate system, but these walls must be surfaces of revolution about the axis of rotation. Here the so slip condition is defined such that the absolute velocity is zero on the walls. An example of this type of boundary would be a cylindrical wind tunnel wall which surrounds a rotating propeller.

Rotationally periodic boundaries may also be used, but the surface must be periodic about the axis of rotation. As an example, it is very common to model flow through a blade row on a turbomachine by assuming the flow to be rotationally periodic and using a periodic domain about a single blade. This permits good resolution of the flow around the blade without the expense of modeling all blades in the blade row (see Figure 2.2.2).

Flow boundary conditions in ANSYS FLUENT (inlets and outlets) can, in most cases, be prescribed in either the stationary or rotating frames. For example, for a velocity inlet, one can specify either the relative velocity or absolute velocity, depending on which is more convenient. For additional information on these and other boundary conditions, see Section 10.7: Setting Up a Single Rotating Reference Frame Problem and Chapter 7: Cell Zone and Boundary Conditions in the separate User's Guide.

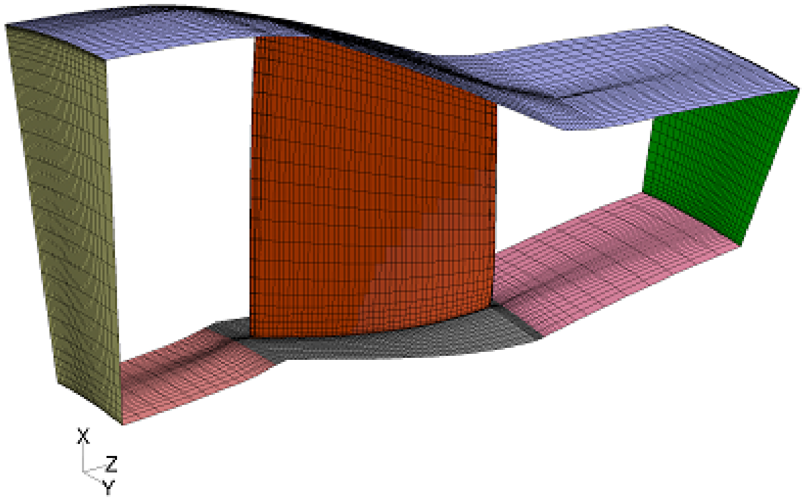


Figure 2.2.2: Single Blade Model with Rotationally Periodic Boundaries

2.3 Flow in Multiple Rotating Reference Frames

Many problems involve multiple moving parts or contain stationary surfaces which are not surfaces of revolution (and therefore cannot be used with the Single Reference Frame modeling approach). For these problems, you must break up the model into multiple fluid/solid cell zones, with interface boundaries separating the zones. Zones which contain the moving components can then be solved using the moving reference frame equations (Section 2.2.1: Equations for a Rotating Reference Frame), whereas stationary zones can be solved with the stationary frame equations. The manner in which the equations are treated at the interface lead to two approaches which are supported in ANSYS FLUENT:

- Multiple Rotating Reference Frames
  - Multiple Reference Frame model (MRF) (see Section 2.3.1: The Multiple Reference Frame Model)
  - Mixing Plane Model (MPM) (see Section 2.3.2: The Mixing Plane Model)
- Sliding Mesh Model (SMM)

Both the MRF and mixing plane approaches are steady-state approximations, and differ primarily in the manner in which conditions at the interfaces are treated. These approaches will be discussed in the sections below. The sliding mesh model approach is, on the other hand, inherently unsteady due to the motion of the mesh with time. This approach is discussed in Chapter 3: Flows Using Sliding and Deforming Meshes.

2.3.1 The Multiple Reference Frame Model

Overview

The MRF model [209] is, perhaps, the simplest of the two approaches for multiple zones. It is a steady-state approximation in which individual cell zones can be assigned different rotational and/or translational speeds. The flow in each moving cell zone is solved using the moving reference frame equations (see Section 2.2: Flow in a Rotating Reference Frame). If the zone is stationary ($\omega = 0$), the equations reduce to their stationary forms. At the interfaces between cell zones, a local reference frame transformation is performed to enable flow variables in one zone to be used to calculate fluxes at the boundary of the adjacent zone. The MRF interface formulation will be discussed in more detail in Section 2.3.1: The MRF Interface Formulation.

It should be noted that the MRF approach does not account for the relative motion of a moving zone with respect to adjacent zones (which may be moving or stationary); the mesh remains fixed for the computation. This is analogous to freezing the motion of the moving part in a specific position and observing the instantaneous flowfield with the rotor in that position. Hence, the MRF is often referred to as the "frozen rotor approach."

While the MRF approach is clearly an approximation, it can provide a reasonable model of the flow for many applications. For example, the MRF model can be used for turbomachinery applications in which rotor-stator interaction is relatively weak, and the flow is relatively uncomplicated at the interface between the moving and stationary zones. In mixing tanks, for example, since the impeller-baffle interactions are relatively weak, large-scale transient effects are not present and the MRF model can be used.

Another potential use of the MRF model is to compute a flow field that can be used as an initial condition for a transient sliding mesh calculation. This eliminates the need for a startup calculation. The multiple reference frame model should not be used, however, if it is necessary to actually simulate the transients that may occur in strong rotor-stator interactions, the sliding mesh model alone should be used (see Section 3.2: Sliding Mesh Theory).

Examples

For a mixing tank with a single impeller, you can define a rotating reference frame that encompasses the impeller and the flow surrounding it, and use a stationary frame for the flow outside the impeller region. An example of this configuration is illustrated in Figure 2.3.1. (The dashes denote the interface between the two reference frames.) Steady-state flow conditions are assumed at the interface between the two reference frames. That is, the velocity at the interface must be the same (in absolute terms) for each reference frame. *The mesh does not move.*

You can also model a problem that includes more than one rotating reference frame. Figure 2.3.2 shows a geometry that contains two rotating impellers side by side. This problem would be modeled using three reference frames: the stationary frame outside both impeller regions and two separate rotating reference frames for the two impellers. (As noted above, the dashes denote the interfaces between reference frames.)

The MRF Interface Formulation

The MRF formulation that is applied to the interfaces will depend on the velocity formulation being used. The specific approaches will be discussed below for each case. It should be noted that the interface treatment applies to the velocity and velocity gradients, since these vector quantities change with a change in reference frame. Scalar quantities, such as temperature, pressure, density, turbulent kinetic energy, etc., do not require any special treatment, and thus are passed locally without any change.

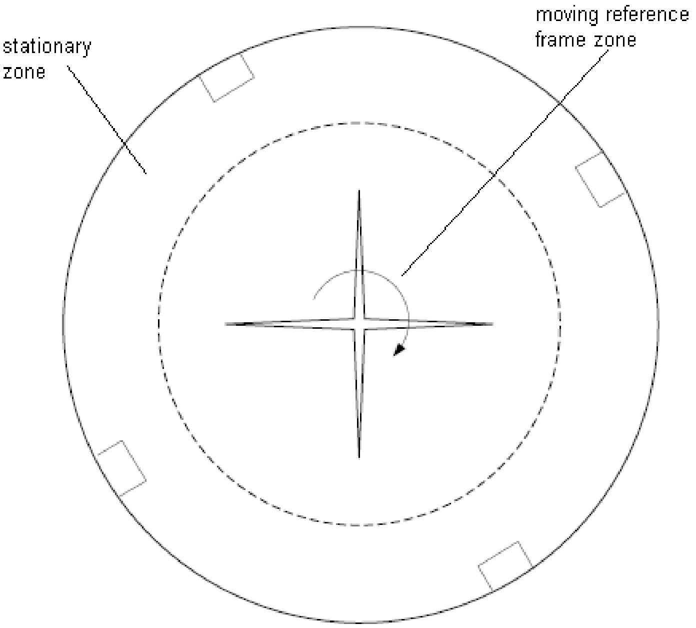


Figure 2.3.1: Geometry with One Rotating Impeller

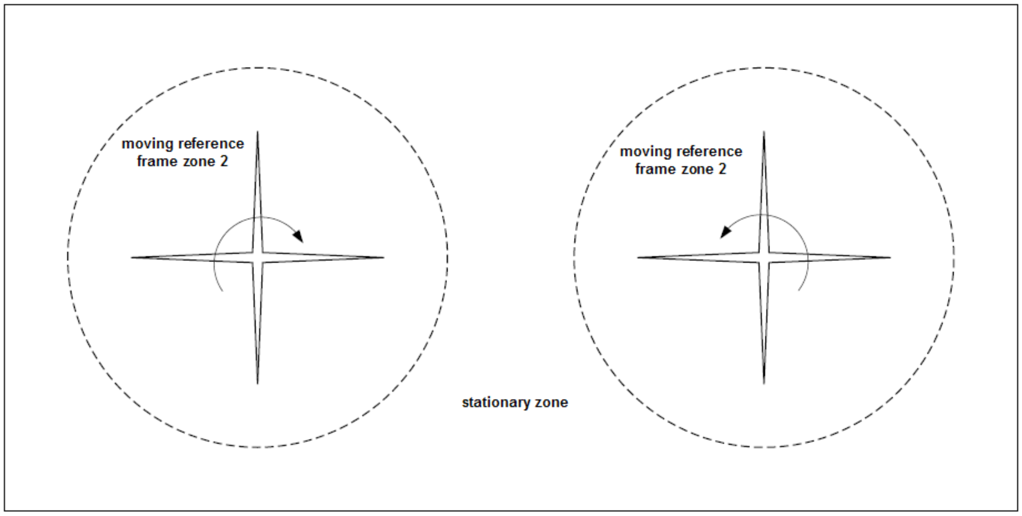


Figure 2.3.2: Geometry with Two Rotating Impellers

Interface Treatment: Relative Velocity Formulation

In ANSYS FLUENT's implementation of the MRF model, the calculation domain is divided into subdomains, each of which may be rotating and/or translating with respect to the laboratory (inertial) frame. The governing equations in each subdomain are written with respect to that subdomain's reference frame. Thus, the flow in stationary and translating subdomains is governed by the equations in Section 1.2: Continuity and Momentum Equations, while the flow in rotating subdomains is governed by the equations presented in Section 2.2.1: Equations for a Rotating Reference Frame.

At the boundary between two subdomains, the diffusion and other terms in the governing equations in one subdomain require values for the velocities in the adjacent subdomain (see Figure 2.3.3). ANSYS FLUENT enforces the continuity of the absolute velocity, $\vec{v}$, to provide the correct neighbor values of velocity for the subdomain under consideration. (This approach differs from the mixing plane approach described in Section 2.3.2: The Mixing Plane Model, where a circumferential averaging technique is used.)

When the relative velocity formulation is used, velocities in each subdomain are computed relative to the motion of the subdomain. Velocities and velocity gradients are converted from a moving reference frame to the absolute inertial frame using Equation 2.3-1.

For a translational velocity $\vec{v}_t$, we have

$$\vec{v} = \vec{v}_r + (\vec{\omega} \times \vec{r}) + \vec{v}_t \tag{2.3-1}$$

From Equation 2.3-1, the gradient of the absolute velocity vector can be shown to be

$$\nabla\vec{v} = \nabla\vec{v}_r + \nabla\left(\vec{\omega} \times \vec{r}\right) \tag{2.3-2}$$

Note that scalar quantities such as density, static pressure, static temperature, species mass fractions, etc., are simply obtained locally from adjacent cells.

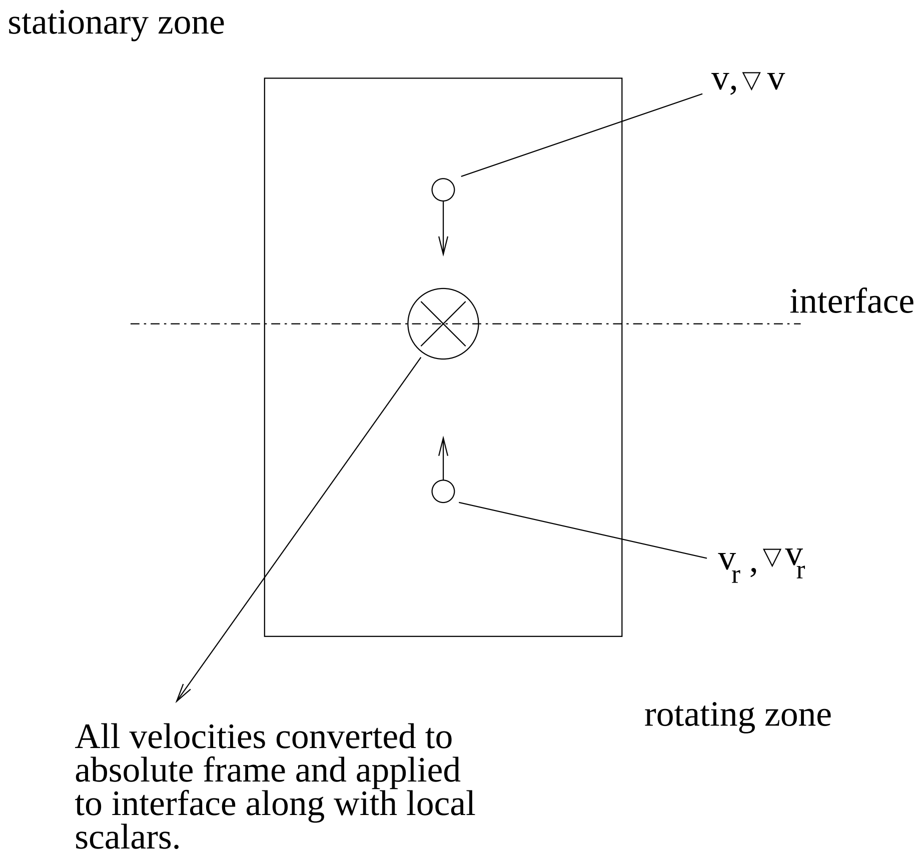

stationary zone
v, ▽ v
interface
v_r , ▽ v_r
rotating zone
All velocities converted to absolute frame and applied to interface along with local scalars.

Figure 2.3.3: Interface Treatment for the MRF Model

Interface Treatment: Absolute Velocity Formulation

When the absolute velocity formulation is used, the governing equations in each subdomain are written with respect to that subdomain's reference frame, but the velocities are stored in the absolute frame. Therefore, no special transformation is required at the interface between two subdomains. Again, scalar quantities are determined locally from adjacent cells.

2.3.2 The Mixing Plane Model

The mixing plane model in ANSYS FLUENT provides an alternative to the multiple reference frame and sliding mesh models for simulating flow through domains with one or more regions in relative motion. This section provides a brief overview of the model and a list of its limitations.

Overview

As discussed in Section 2.3.1: The Multiple Reference Frame Model, the MRF model is applicable when the flow at the interface between adjacent moving/stationary zones is nearly uniform ("mixed out"). If the flow at this interface is not uniform, the MRF model may not provide a physically meaningful solution. The sliding mesh model (see Section 3.2: Sliding Mesh Theory) may be appropriate for such cases, but in many situations it is not practical to employ a sliding mesh. For example, in a multistage turbomachine, if the number of blades is different for each blade row, a large number of blade passages is required in order to maintain circumferential periodicity. Moreover, sliding mesh calculations are necessarily unsteady, and thus require significantly more computation to achieve a final, time-periodic solution. For situations where using the sliding mesh model is not feasible, the mixing plane model can be a cost-effective alternative.

In the mixing plane approach, each fluid zone is treated as a steady-state problem. Flow-field data from adjacent zones are passed as boundary conditions that are spatially averaged or "mixed" at the mixing plane interface. This mixing removes any unsteadiness that would arise due to circumferential variations in the passage-to-passage flow field (e.g., wakes, shock waves, separated flow), thus yielding a steady-state result. Despite the simplifications inherent in the mixing plane model, the resulting solutions can provide reasonable approximations of the time-averaged flow field.

Rotor and Stator Domains

Consider the turbomachine stages shown schematically in Figures 2.3.4 and 2.3.5, each blade passage contains periodic boundaries. Figure 2.3.4 shows a constant radial plane within a single stage of an axial machine, while Figure 2.3.5 shows a constant θ plane within a mixed-flow device. In each case, the stage consists of two flow domains: the rotor domain, which is rotating at a prescribed angular velocity, followed by the stator domain, which is stationary. The order of the rotor and stator is arbitrary (that is, a situation where the rotor is downstream of the stator is equally valid).

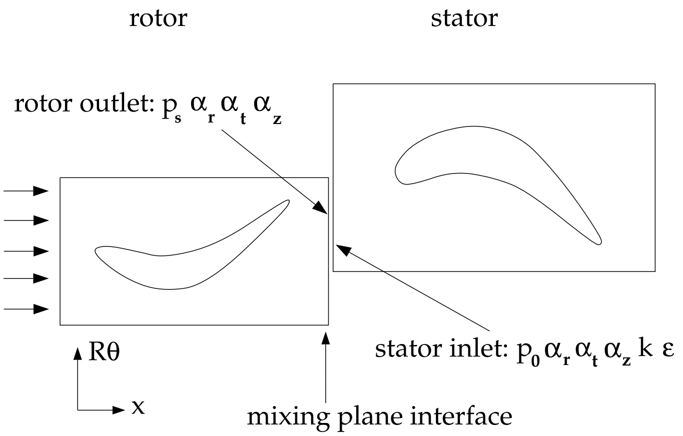

rotor
stator
rotor outlet: $p_s\ \alpha_r\ \alpha_t\ \alpha_z$
$R\theta$
x
stator inlet: $p_0\ \alpha_r\ \alpha_t\ \alpha_z\ k\ \varepsilon$
mixing plane interface

Figure 2.3.4: Axial Rotor-Stator Interaction (Schematic Illustrating the Mixing Plane Concept)

In a numerical simulation, each domain will be represented by a separate mesh. The flow information between these domains will be coupled at the mixing plane interface (as shown in Figures 2.3.4 and 2.3.5) using the mixing plane model. Note that you may couple any number of fluid zones in this manner; for example, four blade passages can be coupled using three mixing planes.

i Note that the stator and rotor passages are separate cell zones, each with their own inlet and outlet boundaries. You can think of this system as a set of SRF models for each blade passage coupled by boundary conditions supplied by the mixing plane model.

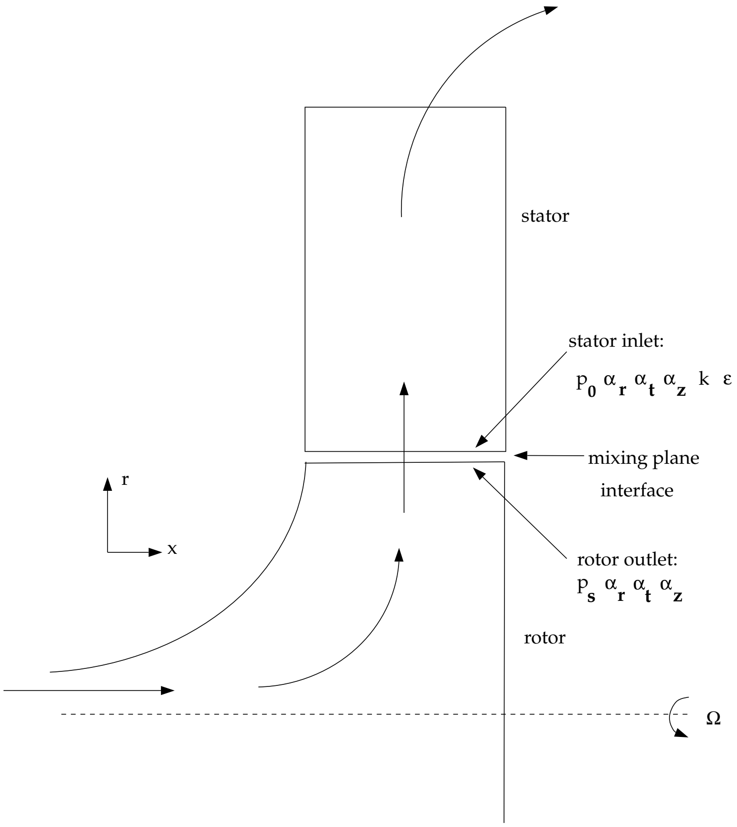

stator
stator inlet:
p_0 α_r α_t α_z k ε
mixing plane
interface
r
x
rotor outlet:
p_s α_r α_t α_z
rotor
Ω

Figure 2.3.5: Radial Rotor-Stator Interaction (Schematic Illustrating the Mixing Plane Concept)

The Mixing Plane Concept

The essential idea behind the mixing plane concept is that each fluid zone is solved as a steady-state problem. At some prescribed iteration interval, the flow data at the mixing plane interface are averaged in the circumferential direction on both the stator outlet and the rotor inlet boundaries. The ANSYS FLUENT implementation gives you the choice of three types of averaging methods: area-weighted averaging, mass averaging, and mixed-out averaging. By performing circumferential averages at specified radial or axial stations, “profiles” of boundary condition flow variables can be defined. These profiles—which will be functions of either the axial or the radial coordinate, depending on the orientation of the mixing plane—are then used to update boundary conditions along the two zones of the mixing plane interface. In the examples shown in Figures 2.3.4 and 2.3.5, profiles of averaged total pressure (p_0), direction cosines of the local flow angles in the radial, tangential, and axial directions ($\alpha_r, \alpha_t, \alpha_z$), total temperature ($T_0$), turbulence kinetic energy (k), and turbulence dissipation rate (ϵ) are computed at the rotor exit and used to update boundary conditions at the stator inlet. Likewise, a profile of static pressure (p_s), direction cosines of the local flow angles in the radial, tangential, and axial directions ($\alpha_r, \alpha_t, \alpha_z$), are computed at the stator inlet and used as a boundary condition on the rotor exit.

Passing profiles in the manner described above assumes specific boundary condition types have been defined at the mixing plane interface. The coupling of an upstream outlet boundary zone with a downstream inlet boundary zone is called a “mixing plane pair”. In order to create mixing plane pairs in ANSYS FLUENT, the boundary zones must be of the following types:

Upstream	Downstream
pressure outlet	pressure inlet
pressure outlet	velocity inlet
pressure outlet	mass flow inlet

For specific instructions about setting up mixing planes, see Section 10.10.2: Setting Up the Mixing Plane Model in the separate User’s Guide.

Choosing an Averaging Method

Three profile averaging methods are available in the mixing plane model:

- area averaging
- mass averaging
- mixed-out averaging

Area Averaging

Area averaging is the default averaging method and is given by

$$\overline{f} = \frac{1}{A}\int_A f \;\; dA \tag{2.3-3}$$

i The pressure and temperature obtained by the area average may not be representative of the momentum and energy of the flow.

Mass Averaging

Mass averaging is given by

$$\overline{f} = \frac{1}{\dot{m}}\int_A f \;\; \rho\vec{V}\cdot\hat{n}dA \tag{2.3-4}$$

where

$$\dot{m} = \int_A \rho\vec{V}\cdot\hat{n}dA$$

This method provides a better representation of the total quantities than the area-averaging method. Convergence problems could arise if severe reverse flow is present at the mixing plane. Therefore, for solution stability purposes, it is best if you initiate the solution with area averaging, then switch to mass averaging after reverse flow dies out.

i Mass averaging averaging is not available with multiphase flows.

Mixed-Out Averaging

The mixed-out averaging method is derived from the conservation of mass, momentum and energy:

$$\begin{aligned}
F &= \int_A \rho(\vec{V}\cdot\hat{n})dA \\
M_1 &= \int_A \rho(\vec{V}\cdot\hat{n})udA + \int_A P(\hat{e_x}\cdot\hat{n})dA \\
M_2 &= \int_A \rho(\vec{V}\cdot\hat{n})vdA + \int_A P(\hat{e_y}\cdot\hat{n})dA \\
M_3 &= \int_A \rho(\vec{V}\cdot\hat{n})wdA + \int_A P(\hat{e_z}\cdot\hat{n})dA \\
E &= \frac{\gamma R}{(\gamma-1)}\int_A \rho(\vec{V}\cdot\hat{n})TdA + \frac{1}{2}\int_A \rho(\vec{V}\cdot\hat{n})(u^2+v^2+w^2)dA
\end{aligned} \tag{2.3-5}$$

Because it is based on the principals of conservation, the mixed-out average is considered a better representation of the flow since it reflects losses associated with non-uniformities in the flow profiles. However, like the mass-averaging method, convergence difficulties can arise when severe reverse flow is present across the mixing plane. Therefore, it is best if you initiate the solution with area averaging, then switch to mixed-out averaging after reverse flow dies out.

Mixed-out averaging assumes that the fluid is a compressible ideal-gas with constant specific heat, C_p.

i Mixed-out averaging is not available with multiphase flows.

ANSYS FLUENT's Mixing Plane Algorithm

ANSYS FLUENT's basic mixing plane algorithm can be described as follows:

1. Update the flow field solutions in the stator and rotor domains.
2. Average the flow properties at the stator exit and rotor inlet boundaries, obtaining profiles for use in updating boundary conditions.
3. Pass the profiles to the boundary condition inputs required for the stator exit and rotor inlet.
4. Repeat steps 1–3 until convergence.

> *i* Note that it may be desirable to under-relax the changes in boundary condition values in order to prevent divergence of the solution (especially early in the computation). ANSYS FLUENT allows you to control the under-relaxation of the mixing plane variables.

Mass Conservation

Note that the algorithm described above will not rigorously conserve mass flow across the mixing plane if it is represented by a pressure outlet and pressure inlet mixing plane pair. If you use a pressure outlet and mass flow outlet pair instead, ANSYS FLUENT will force mass conservation across the mixing plane. The basic technique consists of computing the mass flow rate across the upstream zone (pressure outlet) and adjusting the mass flux profile applied at the mass flow inlet such that the downstream mass flow matches the upstream mass flow. This adjustment occurs at every iteration, thus ensuring rigorous conservation of mass flow throughout the course of the calculation.

> *i* Note that, since mass flow is being fixed in this case, there will be a jump in total pressure across the mixing plane. The magnitude of this jump is usually small compared with total pressure variations elsewhere in the flow field.

Swirl Conservation

By default, ANSYS FLUENT does not conserve swirl across the mixing plane. For applications such as torque converters, where the sum of the torques acting on the components should be zero, enforcing swirl conservation across the mixing plane is essential, and is available in ANSYS FLUENT as a modeling option. Ensuring conservation of swirl is important because, otherwise, sources or sinks of tangential momentum will be present at the mixing plane interface.

Consider a control volume containing a stationary or rotating component (e.g., a pump impeller or turbine vane). Using the moment of momentum equation from fluid mechanics, it can be shown that for steady flow,

$$T = \iint_S r v_\theta \rho \vec{v} \cdot \hat{n} dS \tag{2.3-6}$$

where T is the torque of the fluid acting on the component, r is the radial distance from the axis of rotation, v_θ is the absolute tangential velocity, $\vec{v}$ is the total absolute velocity, and S is the boundary surface. (The product rv_θ is referred to as swirl.)

For a circumferentially periodic domain, with well-defined inlet and outlet boundaries, Equation 2.3-6 becomes

$$T = \iint_{\text{outlet}} r v_\theta \rho \vec{v} \cdot \hat{n} dS + \iint_{\text{inlet}} r v_\theta \rho \vec{v} \cdot \hat{n} dS \tag{2.3-7}$$

where inlet and outlet denote the inlet and outlet boundary surfaces.

Now consider the mixing plane interface to have a finite streamwise thickness. Applying Equation 2.3-7 to this zone and noting that, in the limit as the thickness shrinks to zero, the torque should vanish, the equation becomes

$$\iint_{\text{downstream}} r v_\theta \rho \vec{v} \cdot \hat{n} dS = \iint_{\text{upstream}} r v_\theta \rho \vec{v} \cdot \hat{n} dS \tag{2.3-8}$$

where upstream and downstream denote the upstream and downstream sides of the mixing plane interface. Note that Equation 2.3-8 applies to the full area (360 degrees) at the mixing plane interface.

Equation 2.3-8 provides a rational means of determining the tangential velocity component. That is, ANSYS FLUENT computes a profile of tangential velocity and then uniformly adjusts the profile such that the swirl integral is satisfied. Note that interpolating the tangential (and radial) velocity component profiles at the mixing plane does not affect mass conservation because these velocity components are orthogonal to the face-normal velocity used in computing the mass flux.

Total Enthalpy Conservation

By default, ANSYS FLUENT does not conserve total enthalpy across the mixing plane. For some applications, total enthalpy conservation across the mixing plane is very desirable, because global parameters such as efficiency are directly related to the change in total enthalpy across a blade row or stage. This is available in ANSYS FLUENT as a modeling option.

The procedure for ensuring conservation of total enthalpy simply involves adjusting the downstream total temperature profile such that the integrated total enthalpy matches the upstream integrated total enthalpy. For multiphase flows, conservation of mass, swirl, and enthalpy are calculated for each phase. However, for the Eulerian multiphase model, since mass flow inlets are not permissible, conservation of the above quantities does not occur.

Chapter 3. Flows Using Sliding and Deforming Meshes

This chapter describes the theoretical background of the sliding and dynamic mesh models in ANSYS FLUENT. To learn more about using sliding meshes in ANSYS FLUENT, see Section 11.2: Using Sliding Meshes in the separate User's Guide. Also, for more information about using dynamic meshes in ANSYS FLUENT, see Section 11.3: Using Dynamic Meshes in the separate User's Guide.

Theoretical information about sliding and deforming mesh models is presented in the following sections:

- Section 3.1: Introduction
- Section 3.2: Sliding Mesh Theory
- Section 3.3: Dynamic Mesh Theory

3.1 Introduction

In sliding meshes, the relative motion of stationary and rotating components in a rotating machine will give rise to unsteady interactions. These interactions are illustrated in Figure 3.1.1, and generally classified as follows:

- Potential interactions: flow unsteadiness due to pressure waves which propagate both upstream and downstream.
- Wake interactions: flow unsteadiness due to wakes from upstream blade rows, convecting downstream.
- Shock interactions: for transonic/supersonic flow unsteadiness due to shock waves striking the downstream blade row.

Where the multiple reference frame (MRF) and mixing plane (MP) models, discussed in Chapter 2: Flows with Rotating Reference Frames, are models that are applied to steady-state cases, thus neglecting unsteady interactions, the sliding mesh model cannot neglect unsteady interactions. The sliding mesh model accounts for the relative motion of stationary and rotating components.

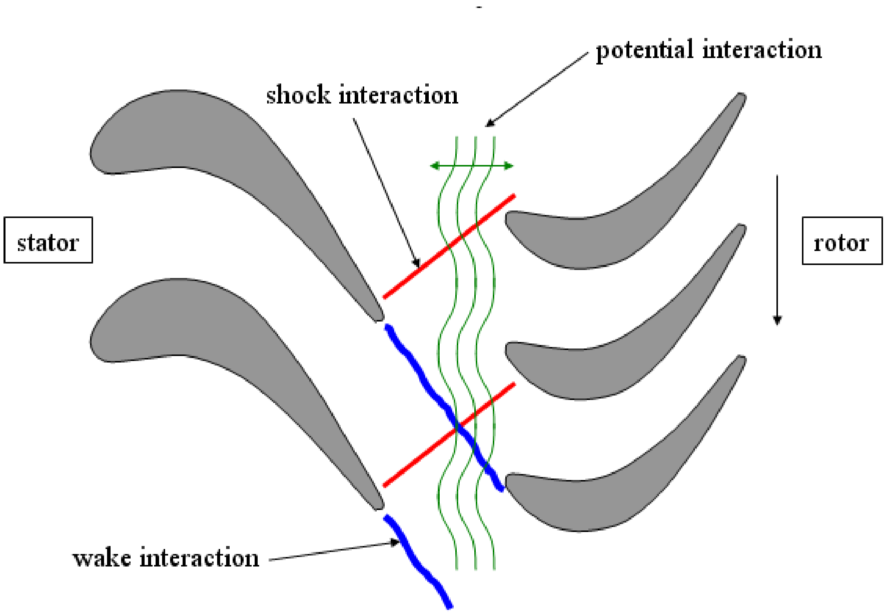


Figure 3.1.1: Illustration of Unsteady Interactions

The dynamic mesh model uses the ANSYS FLUENT solver to move boundaries and/or objects, and to adjust the mesh accordingly. The dynamic mesh model is used when boundaries move rigidly (linear or rotating) with respect to each other. For example

- A piston moving with respect to an engine cylinder.
- A flap moving with respect to an airplane wing.

The dynamic mesh model can also be used when boundaries deform or deflect. For example

- A balloon that is being inflated.
- An artificial wall responding to the pressure pulse from the heart.

Conservation Equations

With respect to dynamic meshes, the integral form of the conservation equation for a general scalar, ϕ, on an arbitrary control volume, V, whose boundary is moving can be written as

$$\frac{d}{dt}\int_V \rho\phi dV + \int_{\partial V} \rho\phi \left(\vec{u} - \vec{u}_g\right) \cdot d\vec{A} = \int_{\partial V} \Gamma\nabla\phi \cdot d\vec{A} + \int_V S_\phi dV \tag{3.1-1}$$

where ρ is the fluid density
$\vec{u}$ is the flow velocity vector
$\vec{u}_g$ is the mesh velocity of the moving mesh
Γ is the diffusion coefficient
S_ϕ is the source term of ϕ

Here ∂V is used to represent the boundary of the control volume V.

The time derivative term in Equation 3.1-1 can be written, using a first-order backward difference formula, as

$$\frac{d}{dt}\int_V \rho\phi dV = \frac{(\rho\phi V)^{n+1} - (\rho\phi V)^n}{\Delta t} \tag{3.1-2}$$

where n and $n+1$ denote the respective quantity at the current and next time level. The $(n+1)$th time level volume V^{n+1} is computed from

$$V^{n+1} = V^n + \frac{dV}{dt}\Delta t \tag{3.1-3}$$

where dV/dt is the volume time derivative of the control volume. In order to satisfy the mesh conservation law, the volume time derivative of the control volume is computed from

$$\frac{dV}{dt} = \int_{\partial V} \vec{u}_g \cdot d\vec{A} = \sum_j^{n_f} \vec{u}_{g,j} \cdot \vec{A}_j \tag{3.1-4}$$

where n_f is the number of faces on the control volume and $\vec{A}_j$ is the j face area vector. The dot product $\vec{u}_{g,j} \cdot \vec{A}_j$ on each control volume face is calculated from

$$\vec{u}_{g,j} \cdot \vec{A}_j = \frac{\delta V_j}{\Delta t} \tag{3.1-5}$$

where δV_j is the volume swept out by the control volume face j over the time step Δt.

In the case of the sliding mesh, the motion of moving zones is tracked relative to the stationary frame. Therefore, no moving reference frames are attached to the computational domain, simplifying the flux transfers across the interfaces. In the sliding mesh formulation, the control volume remains constant, therefore from Equation 3.1-3, $\frac{dV}{dt} = 0$ and $V^{n+1} = V^n$. Equation 3.1-2 can now be expressed as follows:

$$\frac{d}{dt}\int_V \rho\phi dV = \frac{[(\rho\phi)^{n+1} - (\rho\phi)^n]V}{\Delta t} \tag{3.1-6}$$

3.2 Sliding Mesh Theory

When a time-accurate solution for rotor-stator interaction (rather than a time-averaged solution) is desired, you must use the sliding mesh model to compute the unsteady flow field. As mentioned in Section 2.1: Introduction, the sliding mesh model is the most accurate method for simulating flows in multiple moving reference frames, but also the most computationally demanding.

Most often, the unsteady solution that is sought in a sliding mesh simulation is time-periodic. That is, the unsteady solution repeats with a period related to the speeds of the moving domains. However, you can model other types of transients, including translating sliding mesh zones (e.g., two cars or trains passing in a tunnel, as shown in Figure 3.2.1).

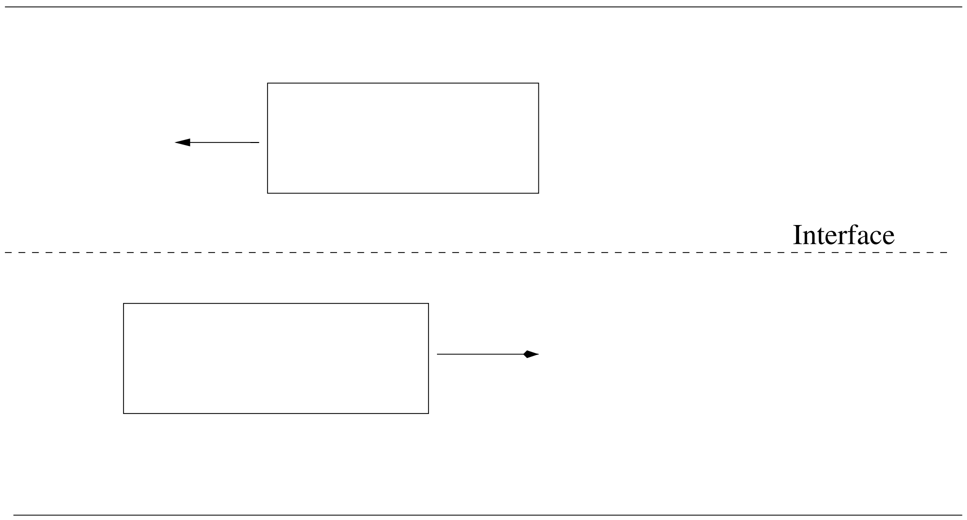


Figure 3.2.1: Two Passing Trains in a Tunnel

Note that for flow situations where there is no interaction between stationary and moving parts (i.e., when there is only a rotor), it is more efficient to use a rotating reference frame. (See Section 2.2: Flow in a Rotating Reference Frame for details.) When transient rotor-stator interaction is desired (as in the examples in Figures 3.2.2 and 3.2.3), you must use sliding meshes. If you are interested in a steady approximation of the interaction, you may use the multiple reference frame model or the mixing plane model, as described in Sections 2.3.1 and 2.3.2.

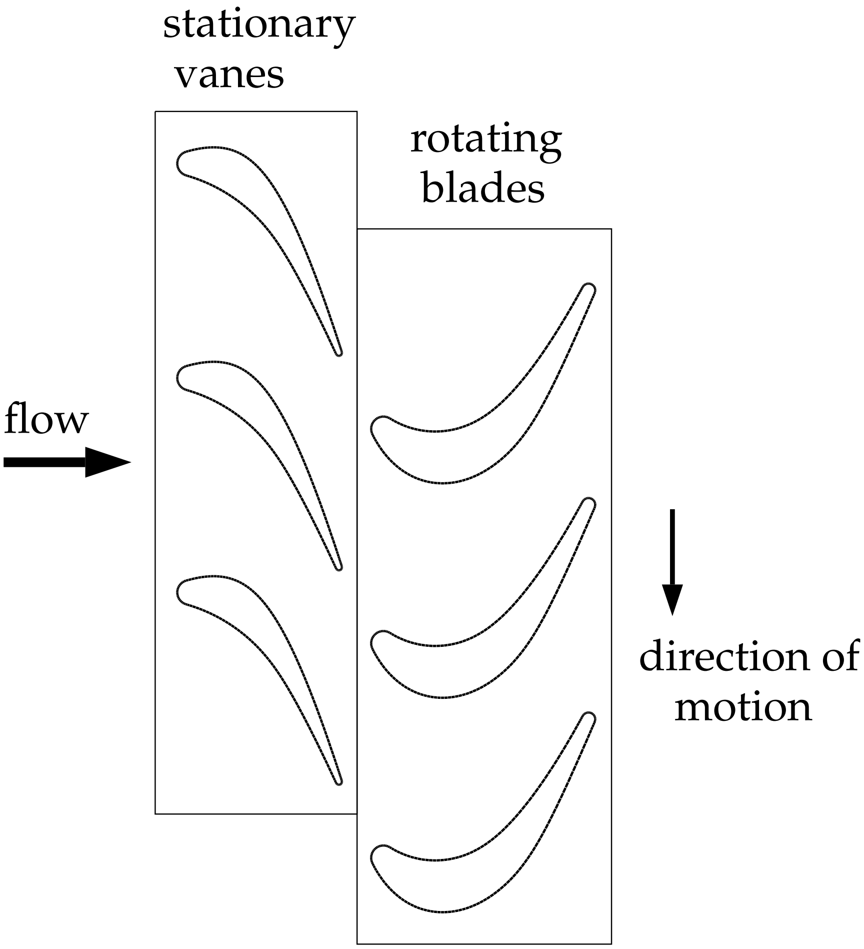


Figure 3.2.2: Rotor-Stator Interaction (Stationary Guide Vanes with Rotating Blades)

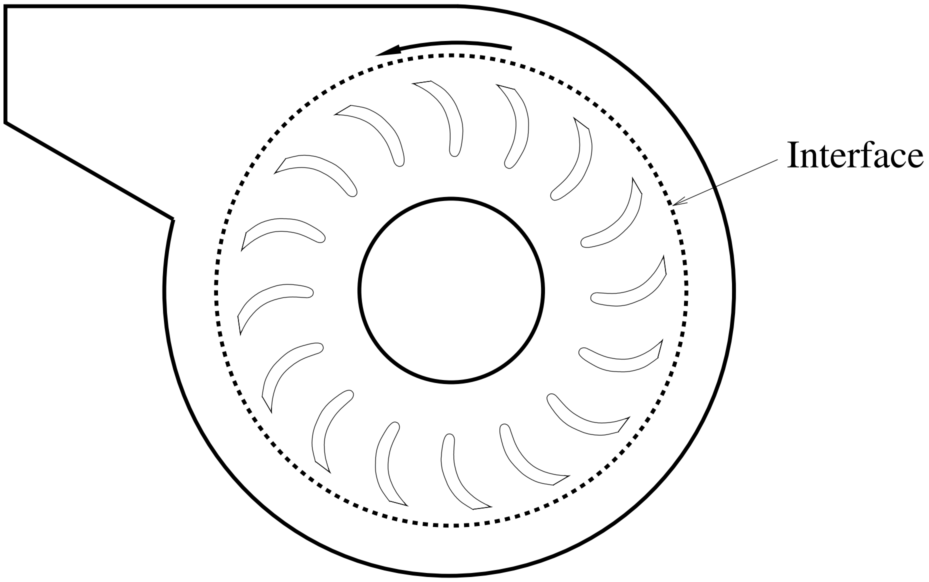


Figure 3.2.3: Blower

The Sliding Mesh Technique

In the sliding mesh technique two or more cell zones are used. (If you generate the mesh in each zone independently, you will need to merge the mesh files prior to starting the calculation, as described in Section 6.3.15: Reading Multiple Mesh/Case/Data Files in the separate User's Guide.) Each cell zone is bounded by at least one "interface zone" where it meets the opposing cell zone. The interface zones of adjacent cell zones are associated with one another to form a "mesh interface." The two cell zones will move relative to each other along the mesh interface.

During the calculation, the cell zones slide (i.e., rotate or translate) relative to one another along the mesh interface in discrete steps. Figures 3.2.4 and 3.2.5 show the initial position of two meshes and their positions after some translation has occurred.

As the rotation or translation takes place, node alignment along the mesh interface is not required. Since the flow is inherently unsteady, a time-dependent solution procedure is required.

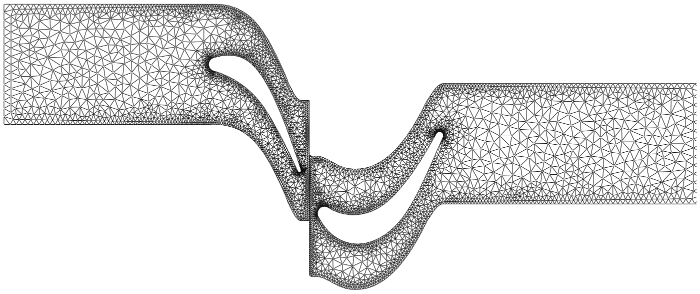


Figure 3.2.4: Initial Position of the Meshes

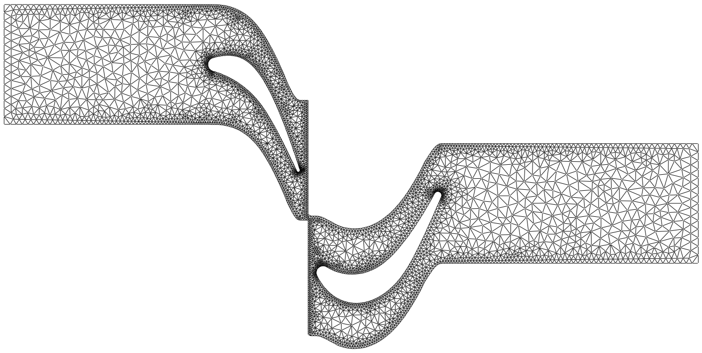


Figure 3.2.5: Rotor Mesh Slides with Respect to the Stator

Mesh Interface Shapes

The mesh interface and the associated interface zones can be any shape, provided that the two interface boundaries are based on the same geometry. Figure 3.2.6 shows an example with a linear mesh interface and Figure 3.2.7 shows a circular-arc mesh interface. (In both figures, the mesh interface is designated by a dashed line.)

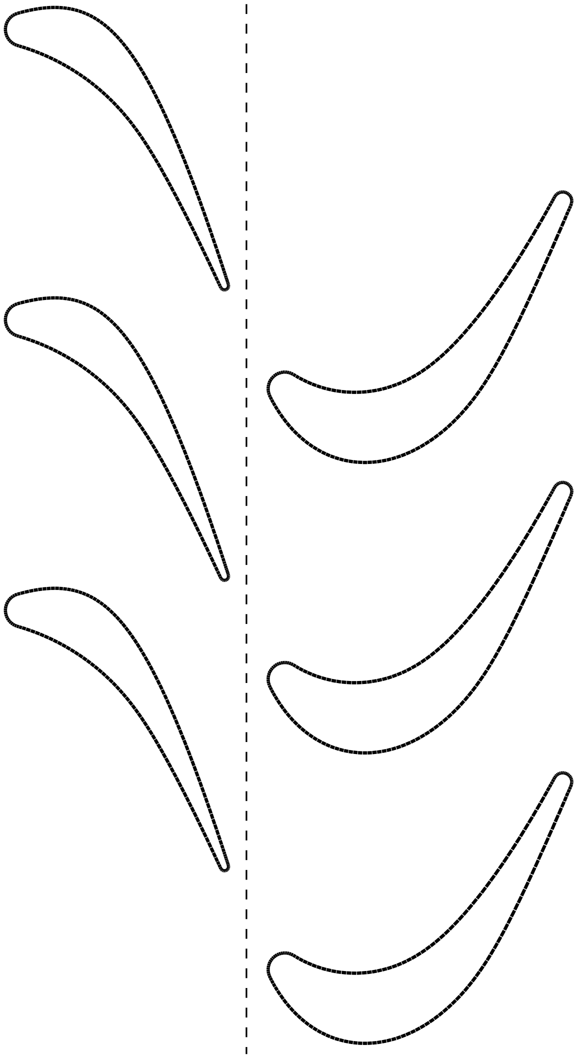


Figure 3.2.6: 2D Linear Mesh Interface

If Figure 3.2.6 was extruded to 3D, the resulting sliding interface would be a planar rectangle; if Figure 3.2.7 was extruded to 3D, the resulting interface would be a cylinder. Figure 3.2.8 shows an example that would use a conical mesh interface. (The slanted, dashed lines represent the intersection of the conical interface with a 2D plane.)

For an axial rotor/stator configuration, in which the rotating and stationary parts are

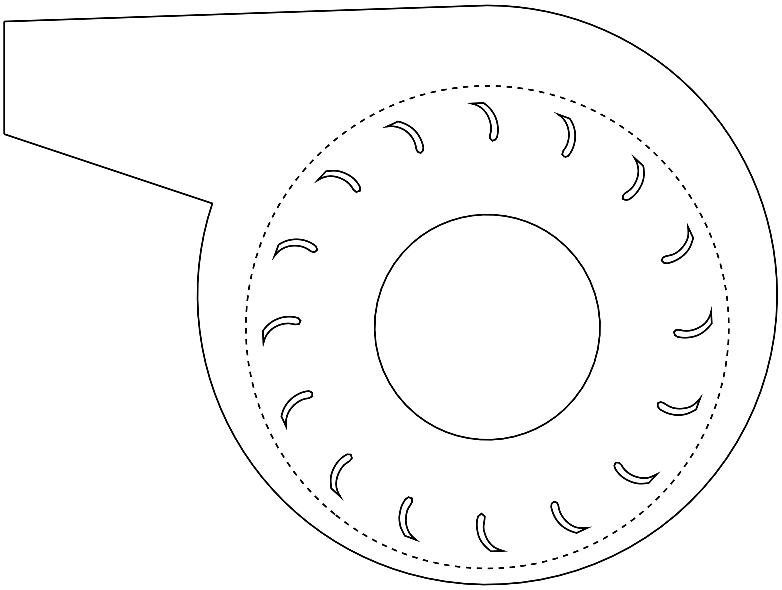


Figure 3.2.7: 2D Circular-Arc Mesh Interface

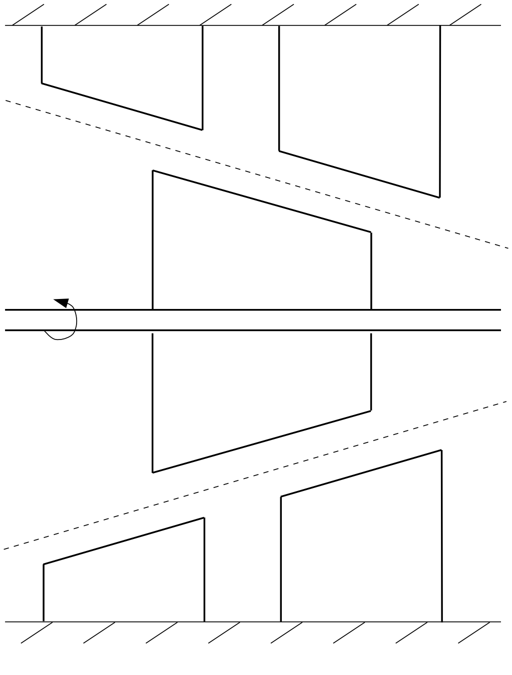


Figure 3.2.8: 3D Conical Mesh Interface

aligned axially instead of being concentric (see Figure 3.2.9), the interface will be a planar sector. This planar sector is a cross-section of the domain perpendicular to the axis of rotation at a position along the axis between the rotor and the stator.

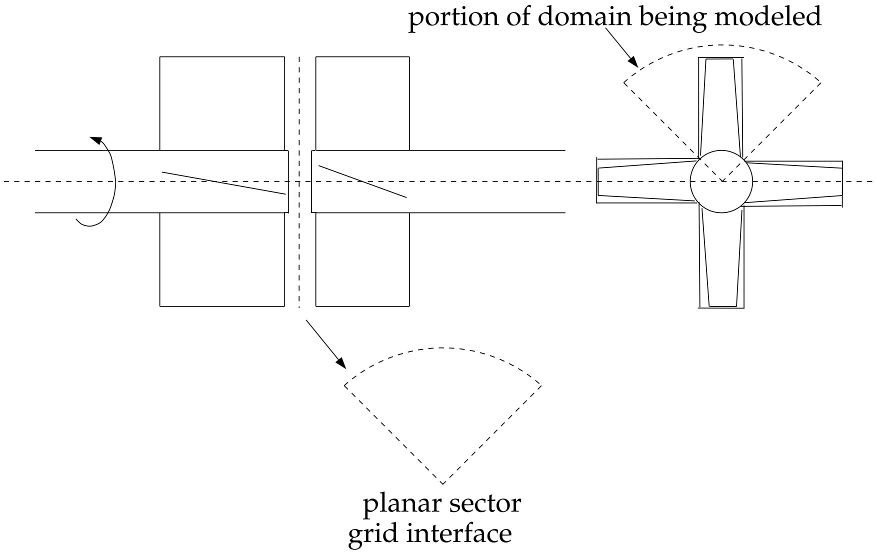


Figure 3.2.9: 3D Planar-Sector Mesh Interface

3.2.1 The Sliding Mesh Concept

As discussed in Section 3.2: Sliding Mesh Theory, the sliding mesh model allows adjacent meshes to slide relative to one another. In doing so, the mesh faces do not need to be aligned on the mesh interface. This situation requires a means of computing the flux across the two non-conformal interface zones of each mesh interface. For information about how ANSYS FLUENT handles non-conformal interfaces, see Section 6.4: Non-Conformal Meshes in the separate User's Guide.

3.3 Dynamic Mesh Theory

The dynamic mesh model in ANSYS FLUENT can be used to model flows where the shape of the domain is changing with time due to motion on the domain boundaries. The dynamic mesh model can also be used for steady-state applications, where it is beneficial to move the mesh in the steady-state solver. The motion can be a prescribed motion (e.g., you can specify the linear and angular velocities about the center of gravity of a solid body with time) or an unprescribed motion where the subsequent motion is determined based on the solution at the current time (e.g., the linear and angular velocities are calculated from the force balance on a solid body, which is what the six degree of freedom (6DOF) solver does (see Section 11.3.7: Using the Six DOF Solver in the separate User's Guide). The update of the volume mesh is handled automatically by ANSYS FLUENT at each time step based on the new positions of the boundaries. To use the dynamic mesh model, you need to provide a starting volume mesh and the description of the motion of any moving zones in the model. ANSYS FLUENT allows you to describe the motion using either boundary profiles, user-defined functions (UDFs), or the Six Degree of Freedom solver (6DOF).

ANSYS FLUENT expects the description of the motion to be specified on either face or cell zones. If the model contains moving and non-moving regions, you need to identify these regions by grouping them into their respective face or cell zones in the starting volume mesh that you generate. Furthermore, regions that are deforming due to motion on their adjacent regions must also be grouped into separate zones in the starting volume mesh. The boundary between the various regions need not be conformal. You can use the non-conformal or sliding interface capability in ANSYS FLUENT to connect the various zones in the final model.

The information in this section is presented in the following:

- Section 3.3.1: Dynamic Mesh Update Methods
- Section 3.3.2: Six DOF (6DOF) Solver Theory

3.3.1 Dynamic Mesh Update Methods

Three groups of mesh motion methods are available in ANSYS FLUENT to update the volume mesh in the deforming regions subject to the motion defined at the boundaries:

- smoothing methods
- dynamic layering
- local remeshing methods

Note that you can use ANSYS FLUENT's dynamic mesh models in conjunction with hanging node adaption, with the exception of dynamic layering and face remeshing. For more information on hanging node adaption, see Section 19.1.1: Hanging Node Adaption.

Spring-Based Smoothing Method

In the spring-based smoothing method, the edges between any two mesh nodes are idealized as a network of interconnected springs. The initial spacings of the edges before any boundary motion constitute the equilibrium state of the mesh. A displacement at a given boundary node will generate a force proportional to the displacement along all the springs connected to the node. Using Hook's Law, the force on a mesh node can be written as

$$\vec{F}_i = \sum_{j}^{n_i} k_{ij} \left(\Delta\vec{x}_j - \Delta\vec{x}_i \right) \tag{3.3-1}$$

where $\Delta\vec{x}_i$ and $\Delta\vec{x}_j$ are the displacements of node i and its neighbor j, n_i is the number of neighboring nodes connected to node i, and k_{ij} is the spring constant (or stiffness) between node i and its neighbor j. The spring constant for the edge connecting nodes i and j is defined as

$$k_{ij} = \frac{1}{\sqrt{\mid \vec{x}_i - \vec{x}_j \mid}} \tag{3.3-2}$$

At equilibrium, the net force on a node due to all the springs connected to the node must be zero. This condition results in an iterative equation such that

$$\Delta\vec{x}_i^{m+1} = \frac{\sum_j^{n_i} k_{ij} \Delta\vec{x}_j^m}{\sum_j^{n_i} k_{ij}} \tag{3.3-3}$$

Since displacements are known at the boundaries (after boundary node positions have been updated), Equation 3.3-3 is solved using a Jacobi sweep on all interior nodes. At convergence, the positions are updated such that

$$\vec{x}_i^{n+1} = \vec{x}_i^n + \Delta\vec{x}_i^{m,\text{converged}} \tag{3.3-4}$$

where $n+1$ and n are used to denote the positions at the next time step and the current time step, respectively. The spring-based smoothing is shown in Figures 3.3.1 and 3.3.2 for a cylindrical cell zone where one end of the cylinder is moving.

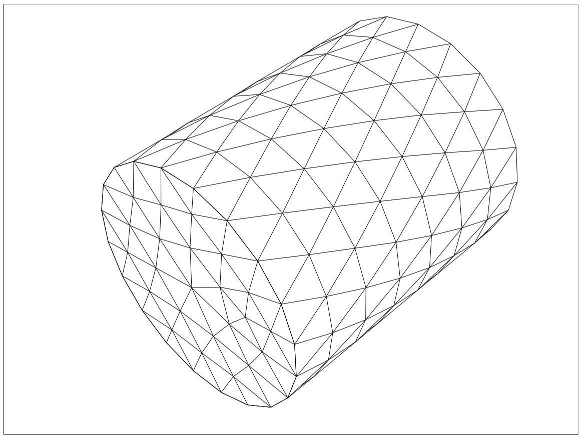


Figure 3.3.1: Spring-Based Smoothing on Interior Nodes: Start

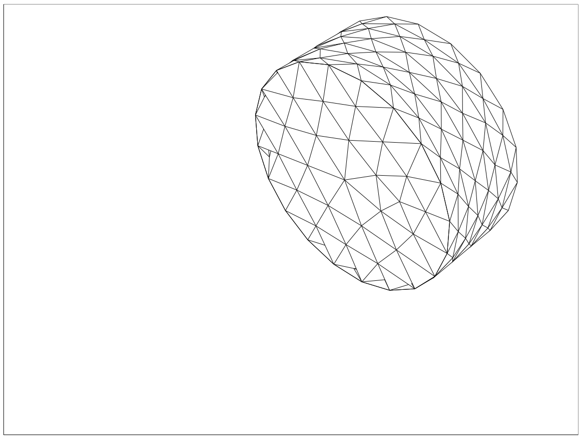


Figure 3.3.2: Spring-Based Smoothing on Interior Nodes: End

Laplacian Smoothing Method

Laplacian smoothing is the most commonly used and the simplest mesh smoothing method. This method adjusts the location of each mesh vertex to the geometric center of its neighboring vertices. This method is computationally inexpensive but it does not guarantee an improvement on mesh quality, since repositioning a vertex by Laplacian smoothing can result in poor quality elements. To overcome this problem, ANSYS FLUENT only relocates the vertex to the geometric center of its neighboring vertices if and only if there is an improvement in the mesh quality (i.e., the skewness has been improved).

This improved Laplacian smoothing can be enabled on deforming boundaries only (i.e., the zone with triangular elements in 3D and zones with linear elements in 2D). The computation of the node positions works as follows:

$$\overline{\vec{x}_i^m} = \frac{\sum_j^{n_i} \vec{x}_j^m}{n_i} \qquad (3.3\text{-}5)$$

where $\overline{\vec{x}_i^m}$ is the averaged node position of node i at iteration m, $\vec{x}_j^m$ is the node position of neighbor node of $\vec{x}_i^m$ at iteration m, and n_i is the number nodes neighboring node i. The new node position $\vec{x}_i^{m+1}$ is then computed as follows:

$$\vec{x}_i^{m+1} = \vec{x}_i^m(1-\beta) + \overline{\vec{x}_i^m}\beta \qquad (3.3\text{-}6)$$

where β is the boundary node relaxation factor.

This update only happens if the maximum skewness of all faces adjacent to $\vec{x}_i^{m+1}$ is improved in comparison to $\vec{x}_i^m$.

Boundary Layer Smoothing Method

The boundary layer smoothing method is used to deform the boundary layer during a moving-deforming mesh simulation. For cases that have a Mesh Motion UDF (see Section 11.3.9: User-Defined Motion in the separate User's Guide) applied to a face zone with adjacent boundary layers, the boundary layer will deform according to the UDF that is applied to the face zone. This smoothing method preserves the height of each boundary layer and can be applied to boundary layer zones of all mesh types (wedges and hexahedra in 3D, quadrilaterals in 2D).

Consider the example below, where a UDF of the form `DEFINE_GRID_MOTION` provides the moving-deforming mesh model with the locations of the nodes located on the compliant strip on an idealized airfoil. The node motion varies sinusoidally (Figures 3.3.5 and 3.3.6), both in time and space as seen by the deformation of the face zone and the respective boundary layer. A deforming flag is set on the adjacent cell zone, such that the cells adjacent to the deforming wall will also be deformed, in order to avoid skewness.

Compare the original mesh (Figure 3.3.3) prior to applying the Mesh Motion UDF to that mesh whose boundary layer has been deformed (Figure 3.3.5).

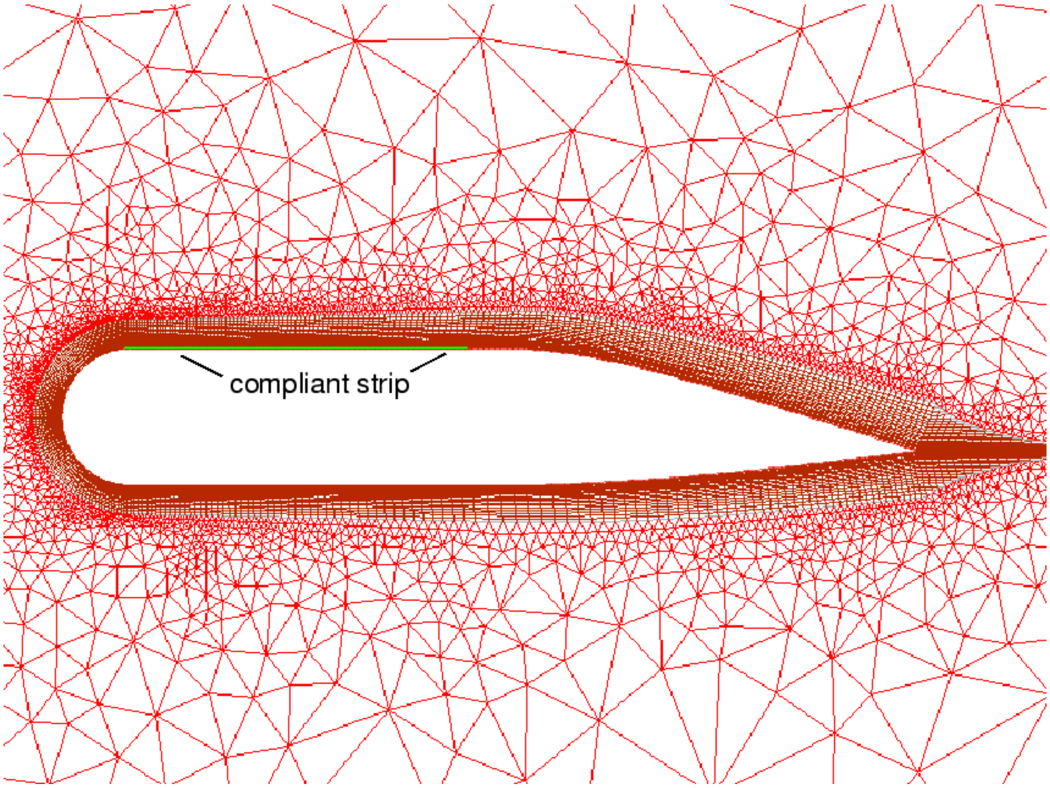


Figure 3.3.3: The Mesh Prior to Applying Boundary Layer Smoothing

To find out how to set up a deforming boundary layer for smoothing, see Section 11.3.9: Specifying Boundary Layer Deformation Smoothing (in the separate User's Guide).

Dynamic Layering Method

In prismatic (hexahedral and/or wedge) mesh zones, you can use dynamic layering to add or remove layers of cells adjacent to a moving boundary, based on the height of the layer adjacent to the moving surface. The dynamic mesh model in ANSYS FLUENT allows you to specify an ideal layer height on each moving boundary. The layer of cells adjacent to the moving boundary (layer j in Figure 3.3.7) is split or merged with the layer of cells next to it (layer i in Figure 3.3.7) based on the height (h) of the cells in layer j.

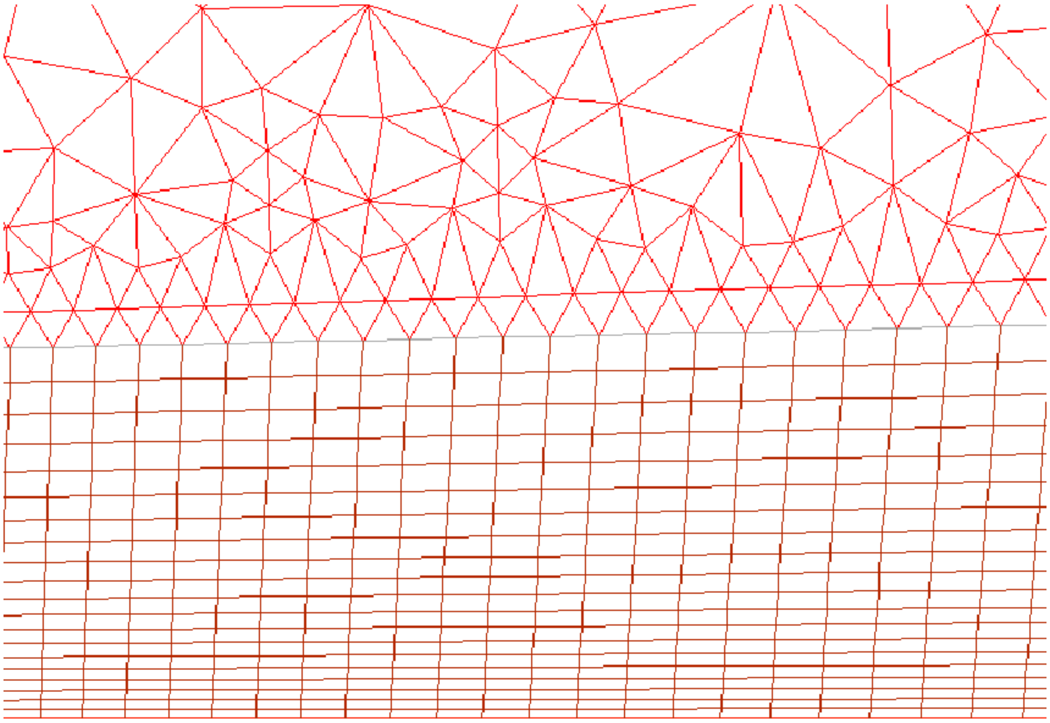


Figure 3.3.4: Zooming into the Mesh of the Compliant Strip Prior to Applying Boundary Layer Smoothing

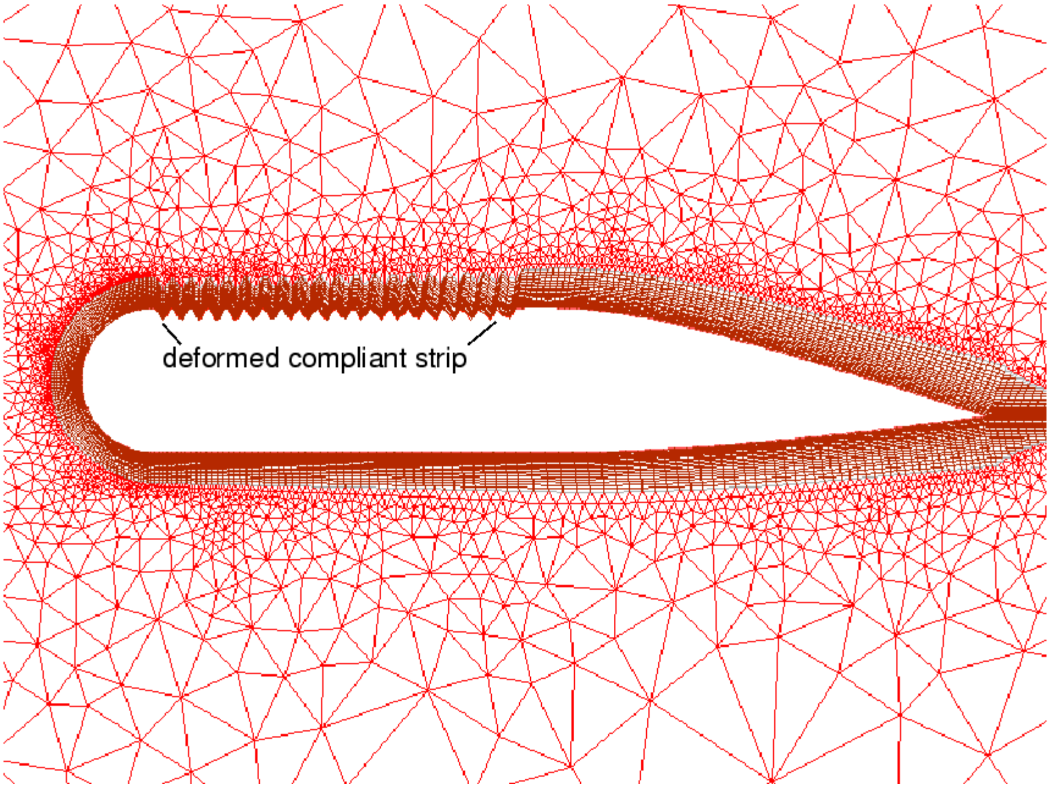


Figure 3.3.5: The Mesh After Applying Boundary Layer Smoothing

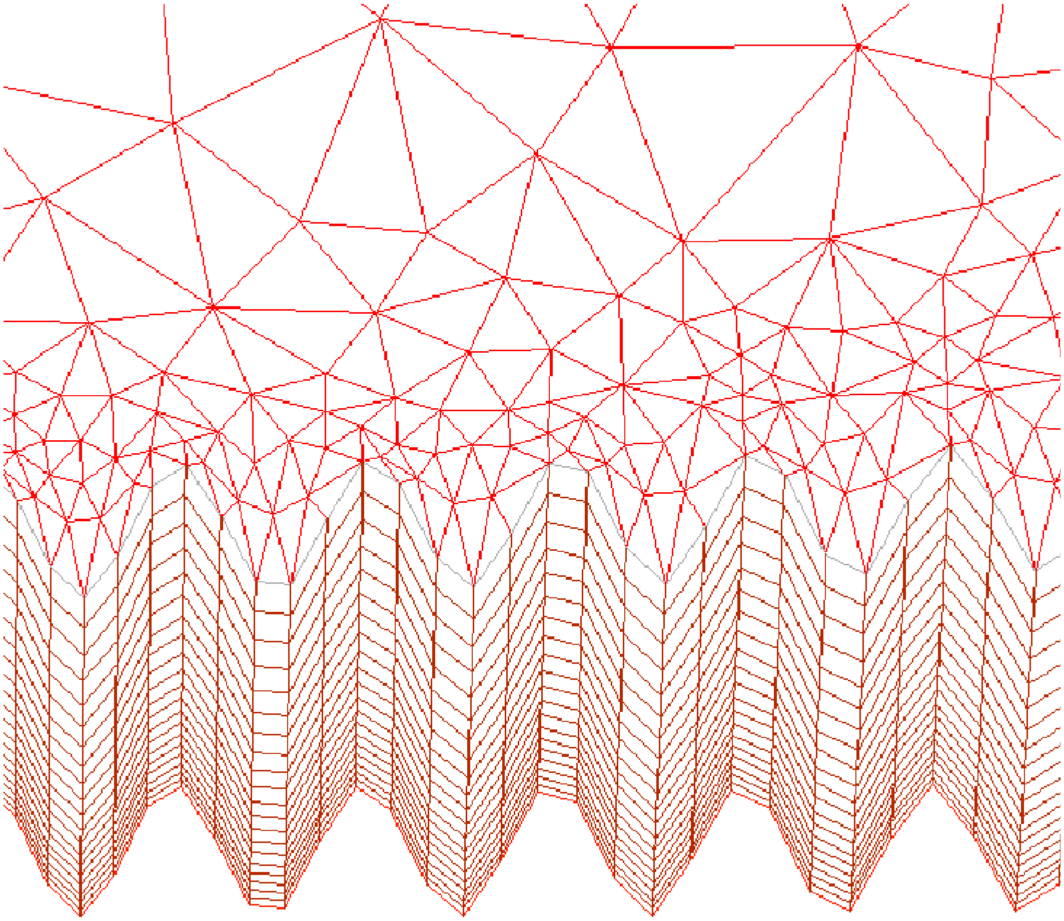


Figure 3.3.6: Zooming into the Deformed Boundary Layer of the Compliant Strip

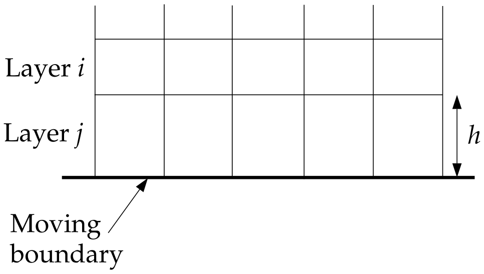


Figure 3.3.7: Dynamic Layering

If the cells in layer j are expanding, the cell heights are allowed to increase until

$$h_{\text{min}} > (1 + \alpha_s) h_{\text{ideal}} \tag{3.3-7}$$

where h_{min} is the minimum cell height of cell layer j, h_{ideal} is the ideal cell height, and α_s is the layer split factor. Note that ANSYS FLUENT allows you to define h_{ideal} as either a constant value or a value that varies as a function of time or crank angle. When the condition in Equation 3.3-7 is met, the cells are split based on the specified layering option. This option can be height based or ratio based.

With the height-based option, the cells are split to create a layer of cells with constant height h_{ideal} and a layer of cells of height $h - h_{\text{ideal}}$. With the ratio-based option, the cells are split such that locally, the ratio of the new cell heights is exactly α_s everywhere. Figures 3.3.8 and 3.3.9 show the result of splitting a layer of cells above a valve geometry using the height-based and ratio-based option.

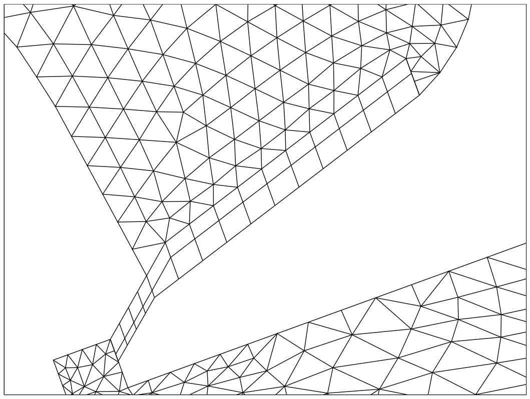


Figure 3.3.8: Results of Splitting Layer with the Height-Based Option

If the cells in layer j are being compressed, they can be compressed until

$$h_{\text{min}} < \alpha_c h_{\text{ideal}} \tag{3.3-8}$$

where α_c is the layer collapse factor. When this condition is met, the compressed layer of cells is merged into the layer of cells above the compressed layer; i.e., the cells in layer j are merged with those in layer i.

Remeshing Methods

On zones with a triangular or tetrahedral mesh, the spring-based smoothing method (described in Section 3.3.1: Spring-Based Smoothing Method) is normally used. When

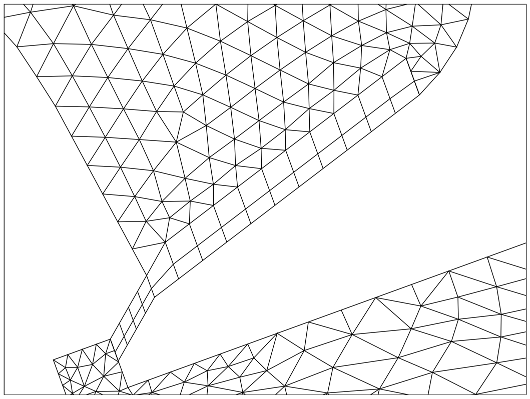


Figure 3.3.9: Results of Splitting Layer with the Ratio-Based Option

the boundary displacement is large compared to the local cell sizes, the cell quality can deteriorate or the cells can become degenerate. This will invalidate the mesh (e.g., result in negative cell volumes) and consequently, will lead to convergence problems when the solution is updated to the next time step.

To circumvent this problem, ANSYS FLUENT agglomerates cells that violate the skewness or size criteria and locally remeshes the agglomerated cells or faces. If the new cells or faces satisfy the skewness criterion, the mesh is locally updated with the new cells (with the solution interpolated from the old cells). Otherwise, the new cells are discarded.

ANSYS FLUENT includes several remeshing methods that include local remeshing, local face remeshing (for 3D flows only), face region remeshing, and 2.5D surface remeshing (for 3D flows only). The available remeshing methods in ANSYS FLUENT work for triangular-tetrahedral zones and mixed zones where the non-triangular/tetrahedral elements are skipped. The exception is the 2.5D model, where the available remeshing method only work on wedges extruded from triangular surfaces or hex meshes.

Local Remeshing Method

Using the local remeshing method, ANSYS FLUENT marks cells based on cell skewness and minimum and maximum length scales as well as an optional sizing function.

ANSYS FLUENT evaluates each cell and marks it for remeshing if it meets one or more of the following criteria:

- It has a skewness that is greater than a specified maximum skewness.
- It is smaller than a specified minimum length scale.
- It is larger than a specified maximum length scale.
- Its height does not meet the specified length scale (at moving face zones, e.g., above a moving piston).

Face Region Remeshing Method

In addition to remeshing the volume mesh, ANSYS FLUENT also allows triangular and linear faces on a deforming boundary to be remeshed. ANSYS FLUENT marks deforming boundary faces for remeshing based on moving and deforming loops of faces.

For face region remeshing, ANSYS FLUENT marks the region of faces on the deforming boundaries at the moving boundary based on minimum and maximum length scales. Once marked, ANSYS FLUENT remeshes the faces and the adjacent cells to produce a very regular mesh on the deforming boundary at the moving boundary (e.g., Figure 3.3.10). Using this method, ANSYS FLUENT is able to remesh across multiple face zones.

For 3D simulations, ANSYS FLUENT allows remeshing with symmetric boundary conditions, and across multiple face zones which includes preserving features not only between the different face zones, but also within a face zone. For more information, see Section 3.3.1: Feature Detection.

ANSYS FLUENT automatically extracts loops on the boundary of the face zone whose nodes are moving or deforming. Consider a simple tetrahedral mesh of a cylinder whose bottom wall is moving (see Figure 3.3.10). On the deforming boundary, a single loop is generated at the bottom end of the cylinder (where the nodes are moving). ANSYS FLUENT analyzes the height of the faces connected to the nodes on the loop and subsequently, splits or merges the faces depending on the specified maximum or minimum length scale.

If the faces in layer j are expanding, they are allowed to expand until the maximum length scale is reached. Conversely, if the layer is contracting, they are allowed to contract until the minimum length scale is reached. When this condition is met, the compressed layer of faces is merged into the layer of faces above it. The face remeshing is illustrated in Figure 3.3.12.

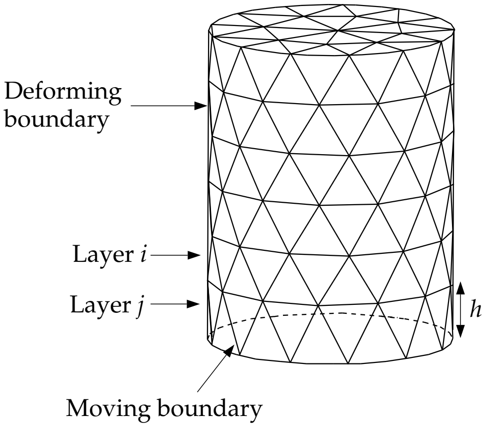


Figure 3.3.10: Remeshing at a Deforming Boundary

Local Face Remeshing Method

The local face remeshing method only applies to 3D geometries. Using this method, ANSYS FLUENT marks the faces (and the adjacent cells) on the deforming boundaries based on the face skewness. Using this method, ANSYS FLUENT is able to remesh locally at deforming boundaries, however, you are not able to remesh across multiple face zones.

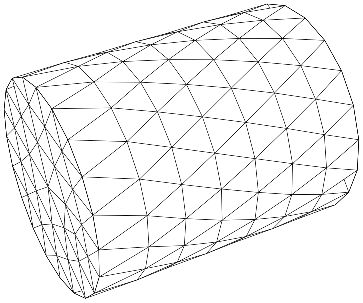


Figure 3.3.11: Expanding Cylinder Before Region Face Remeshing

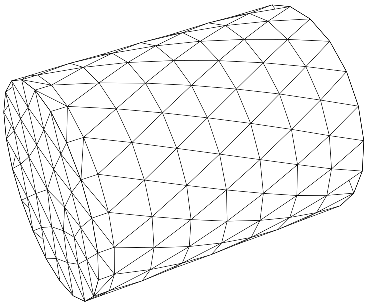


Figure 3.3.12: Expanding Cylinder After Region Face Remeshing

2.5D Surface Remeshing Method

The 2.5D surface remeshing method only applies to extruded 3D geometries and is similar to local remeshing in two dimensions on a triangular surface mesh (not a mixed zone). Faces on a deforming boundary are marked for remeshing based on face skewness, minimum and maximum length scale and an optional sizing function.

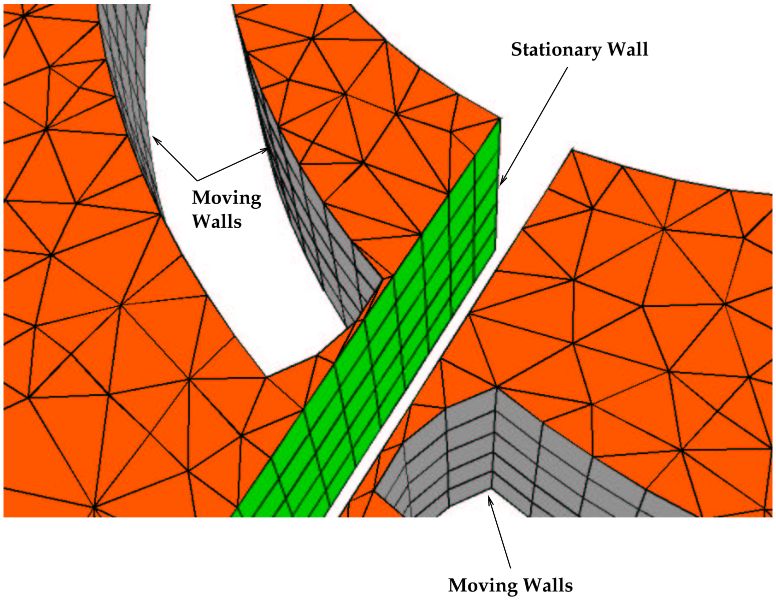


Figure 3.3.13: Close-Up of 2.5D Extruded Flow Meter Pump Geometry Before Remeshing and Laplacian Smoothing

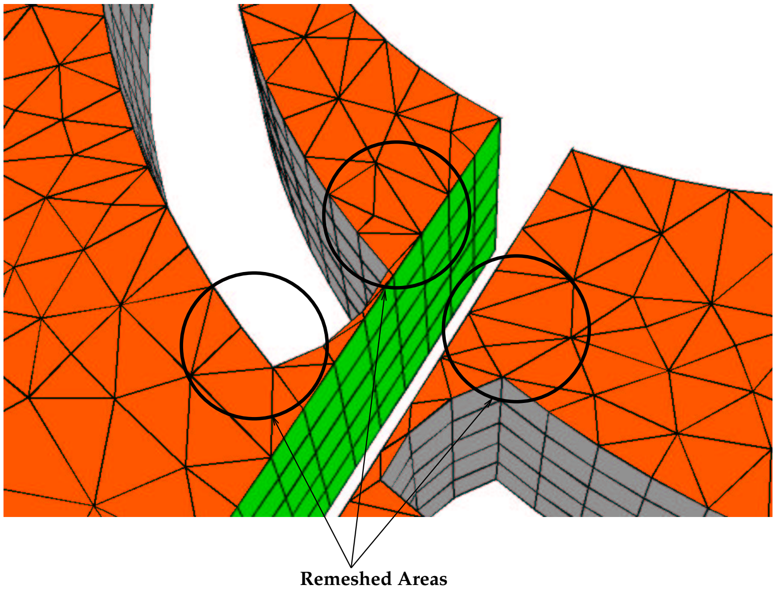


Figure 3.3.14: Close-Up of 2.5D Extruded Flow Meter Pump Geometry After Remeshing and Laplacian Smoothing

Local Remeshing Based on Size Functions

Instead of marking cells based on minimum and maximum length scales, ANSYS FLUENT can also mark cells based on the size distribution generated by sizing functions.

Local remeshing using size functions can be used with the following remeshing methods:

- local remeshing
- 2.5D surface remeshing

Figure 3.3.16 demonstrates the advantages of using size functions for local remeshing:

In determining the sizing function, ANSYS FLUENT draws a bounding box around the zone that is approximately twice the size of the zone, and locates the shortest feature length within each fluid zone. ANSYS FLUENT then subdivides the bounding box based on the shortest feature length and the Size Function Resolution that you specify. This allows ANSYS FLUENT to create a background mesh.

You control the resolution of the background mesh and a background mesh is created for each fluid zone. The shortest feature length is determined by shrinking a second box around the object, and then selecting the shortest edge on that box. The size function is evaluated at the vertex of each individual background mesh.

As seen in Figure 3.3.17, the local value of the size function SF_I is defined by

$$SF_I = \left(\frac{\Sigma \frac{1}{D_J} \Delta s_J}{\Sigma \frac{1}{D_J}} \right) \tag{3.3-9}$$

where D_J is the distance from vertex I on the background mesh to the centroid of boundary cell J and Δs_J is the mesh size (length) of boundary cell J.

The size function is then smoothed using Laplacian smoothing. ANSYS FLUENT then interpolates the value of the size function by calculating the distance L_I from a given cell centroid P to the background mesh vertices that surround the cell (see Figure 3.3.18). The intermediate value of the size function $size_b$ at the centroid is computed from

$$size_b = \left(\frac{\Sigma SF_I \frac{1}{L_I}}{\Sigma \frac{1}{L_I}} \right) \tag{3.3-10}$$

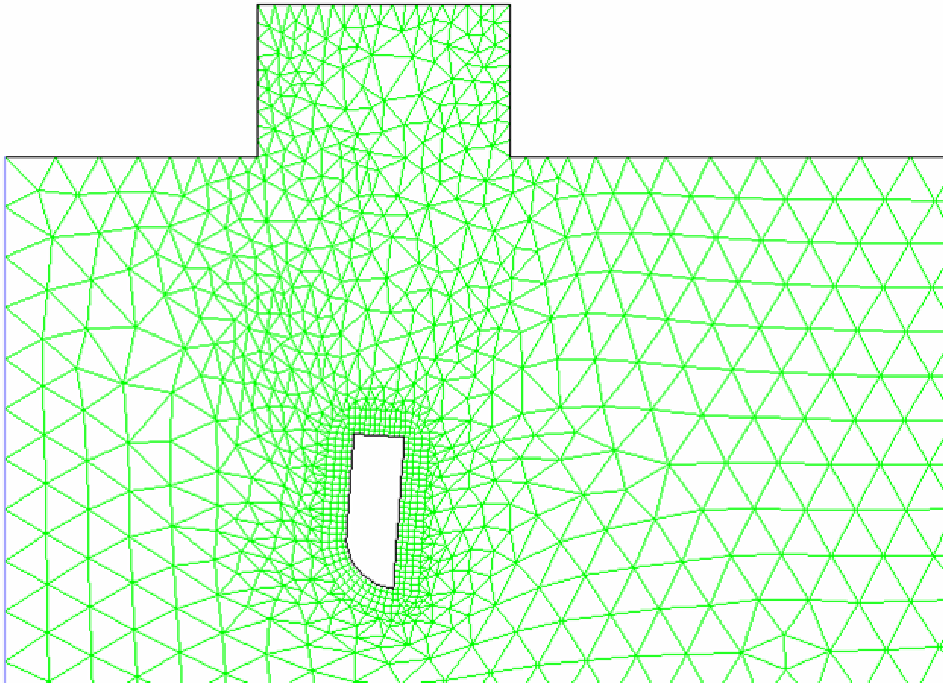


Figure 3.3.15: Mesh at the End of a Dynamic Mesh Simulation Without Size Functions

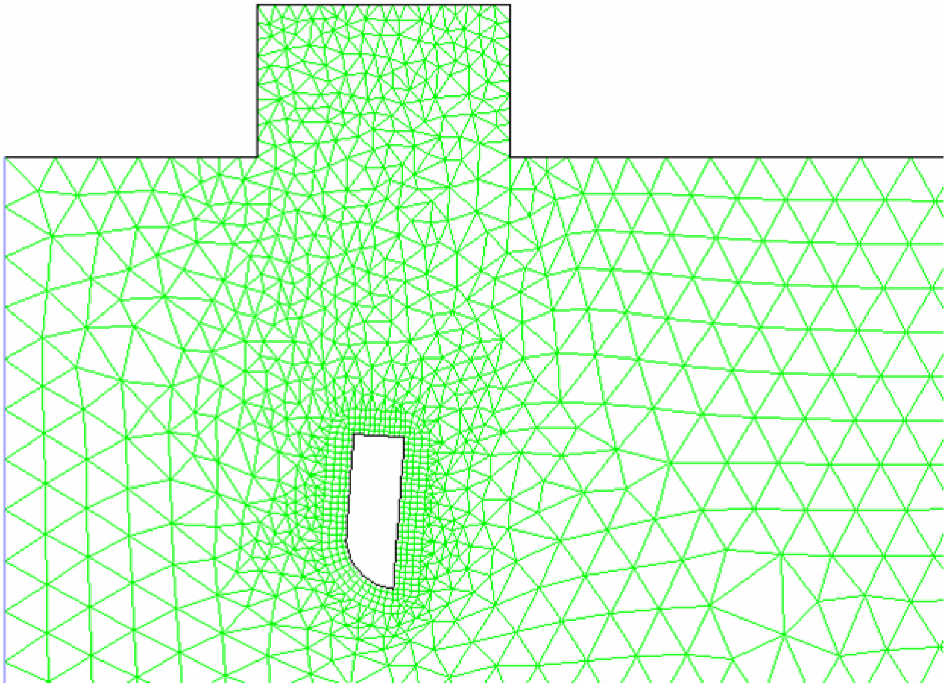


Figure 3.3.16: Mesh at the End of a Dynamic Mesh Simulation With Size Functions

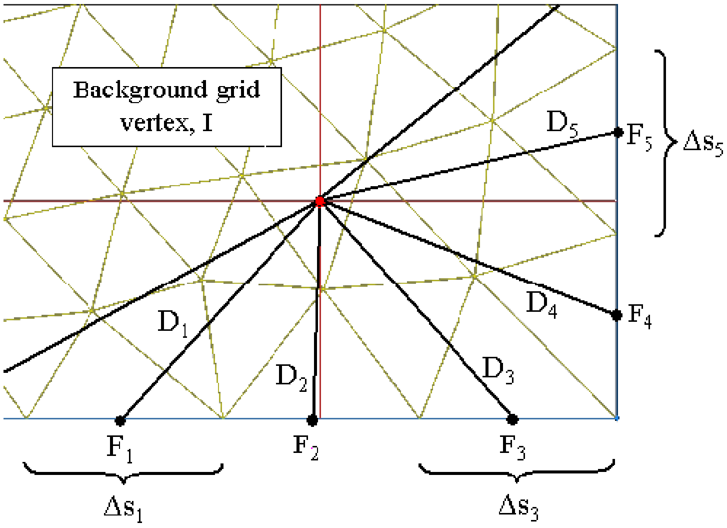


Figure 3.3.17: Size Function Determination at Background Mesh Vertex I

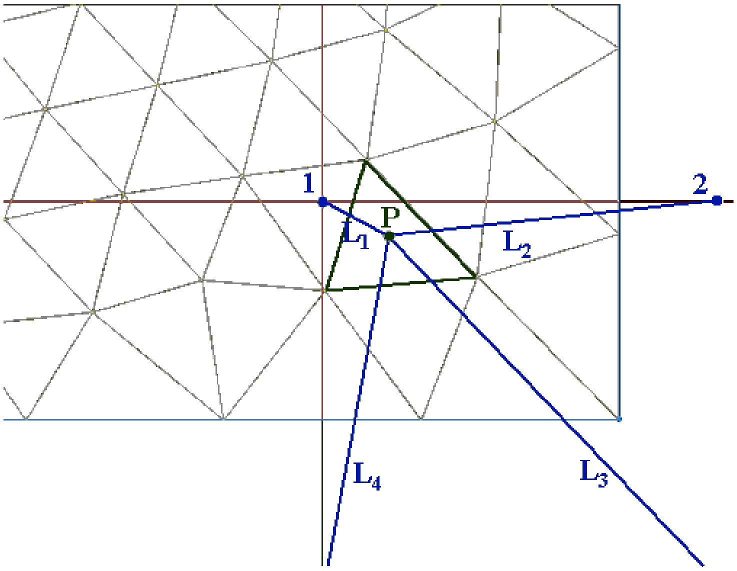


Figure 3.3.18: Interpolating the Value of the Size Function

Next, a single point Q is located within the domain (see Figure 3.3.19) that has the largest distance d_{max} to the nearest boundary to it. The normalized distance d_b for the given centroid P is given by

$$d_b = \frac{d^P_{\mathrm{min}}}{d_{\mathrm{max}}} \tag{3.3-11}$$

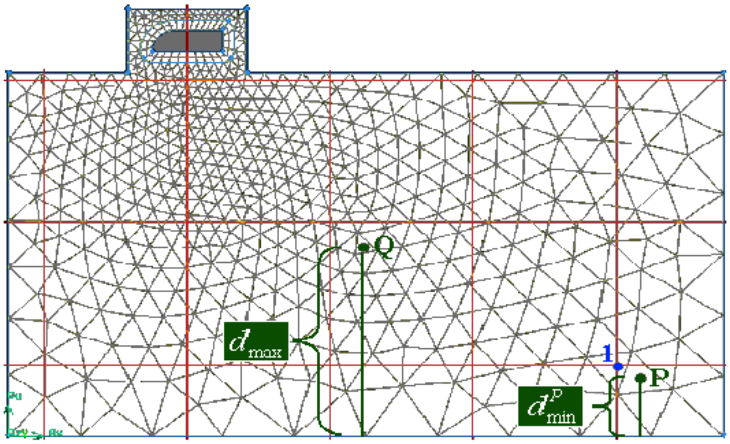


Figure 3.3.19: Determining the Normalized Distance

Using the parameters α and β (the Size Function Variation and the Size Function Rate, respectively), you can write the final value $size_P$ of the size function at point P as

$$size_P = size_b \times (1 + \alpha \times d_b^{1+2\beta}) = size_b \times \gamma \tag{3.3-12}$$

where $size_b$ is the intermediate value of the size function at the cell centroid.

Note that α is the size function *variation*. Positive values mean that the cell size increases as you move away from the boundary. Since the maximum value of d_b is one, the maximum cell size becomes

$$size_{\mathrm{P,max}} = size_b \times (1 + \alpha) = size_b \times \gamma_{\mathrm{max}} \tag{3.3-13}$$

thus, α is really a measure of the maximum cell size.

The factor γ is computed from

$$\gamma = 1 + \alpha d_b^{1+2\beta} \qquad \text{if } \alpha > 0 \tag{3.3-14}$$

$$\gamma = 1 + \alpha d_b^{\frac{1}{1-\beta}} \qquad \text{if } \alpha < 0 \tag{3.3-15}$$

You can use Size Function Variation (or α) to control how large or small an interior cell can be with respect to its closest boundary cell. α ranges from -1 to ∞, an α value of 0.5 indicates that the interior cell size can be, at most, 1.5 the size of the closest boundary cell. Conversely, an α value of -0.5 indicates that the cell size interior of the boundary can be half of that at the closest boundary cell. A value of 0 indicates a constant size distribution away from the boundary.

You can use Size Function Rate (or β) to control how rapidly the cell size varies from the boundary. The value of β should be specified such that $-0.99 < \beta < +0.99$. A positive value indicates a slower transition from the boundary to the specified Size Function Variation value. Conversely, a negative value indicates a faster transition from the boundary to the Size Function Variation value. A value of 0 indicates a linear variation of cell size away from the boundary.

You can also control the resolution of the sizing function with Size Function Resolution. The resolution determines the size of the background bins used to evaluate the size distribution with respect to the shortest feature length of the current mesh. By default, the Size Function Resolution is 3 in 2D problems, and 1 in 3D problems.

A set of default values (based on the current mesh) is automatically generated if you click Use Defaults.

In summary, the sizing function is a distance-weighted average of all mesh sizes on all boundary faces (both stationary and moving boundaries). The sizing function is based on the sizes of the boundary cells, with the size computed from the cell volume by assuming a perfect (equilateral) triangle in 2D and a perfect tetrahedron in 3D. You can control the size distribution by specifying the Size Function Variation and the Size Function Rate. If you have enabled the Sizing Function option, ANSYS FLUENT will agglomerate a cell if

$$size \notin \left[\frac{4}{5}\gamma size_b, \frac{5}{4}\gamma size_b\right] \tag{3.3-16}$$

where γ is a factor defined by Equation 3.3-14 and Equation 3.3-15.

Note that the size function is only used for marking cells *before* remeshing. The size function is not used to govern the size of the cell during remeshing.

Feature Detection

For 3D simulations, ANSYS FLUENT allows you to preserve features on deforming zones not only between the different face zones, but also within a face zone.

In the Geometry Definition tab of the **Dynamic Mesh Zones** dialog box, for any geometry definition, you can indicate whether you want to include features of a specific angle by selecting Include Features under Feature Detection and setting the Feature Angle (the zonal feature angle α) in degrees. If the angle β between adjacent faces is bigger than the specified angle, then the feature is recognized (i.e., $\cos(\beta) < \cos(\alpha)$).

3.3.2 Six DOF (6DOF) Solver Theory

The 6DOF solver in ANSYS FLUENT uses the object's forces and moments in order to compute the translational and angular motion of the center of gravity of an object. The governing equation for the translational motion of the center of gravity is solved for in the inertial coordinate system (Equation 3.3-17).

$$\dot{\vec{v_G}} = \frac{1}{m}\sum \vec{f}_G \tag{3.3-17}$$

where $\dot{\vec{v_G}}$ is the translational motion of the center of gravity, m is the mass, and $\vec{f}_G$ is the force vector due to gravity.

The angular motion of the object, $\dot{\vec{\omega_B}}$, is more easily computed using body coordinates (Equation 3.3-18).

$$\dot{\vec{\omega_B}} = \mathbf{L^{-1}}\left(\sum \vec{M_{\mathrm{B}}} - \vec{\omega_{\mathrm{B}}} \times \mathbf{L}\vec{\omega_{\mathrm{B}}}\right) \tag{3.3-18}$$

where $\mathbf{L}$ is the inertia tensor, $\vec{M_{\mathrm{B}}}$ is the moment vector of the body, and $\vec{\omega_{\mathrm{B}}}$ is the rigid body angular velocity vector.

The moments are transformed from inertial to body coordinates using

$$\vec{M_{\mathrm{B}}} = \mathbf{R}\vec{M_{\mathrm{G}}} \tag{3.3-19}$$

where $\mathbf{R}$ is the following transformation matrix:

$$\begin{array}{ccc}
C_\theta C_\psi & C_\theta S_\psi & -S_\theta \\
S_\phi S_\theta C_\psi - C_\phi S_\psi & S_\phi S_\theta S_\psi + C_\phi C_\psi & S_\phi C_\theta \\
C_\phi S_\theta C_\psi + S_\phi S_\psi & C_\phi S_\theta S_\psi - S_\phi C_\psi & C_\phi C_\theta
\end{array}$$

where, in generic terms, $C_\chi = cos(\chi)$ and $S_\chi = sin(\chi)$. The angles ϕ, θ, and ψ are Euler angles that represent the following sequence of rotations:

- rotation about the x-axis (e.g., roll for airplanes)
- rotation about the y-axis (e.g., pitch for airplanes)
- rotation about the z-axis (e.g., yaw for airplanes)

Once the angular and the translational accelerations are computed from Equation 3.3-17 and Equation 3.3-18, the rates are derived by numerical integration [328]. The angular and translational velocities are used in the dynamic mesh calculations to update the rigid body position.

Chapter 4. Turbulence

This chapter provides theoretical background about the turbulence models available in ANSYS FLUENT. Information is presented in the following sections:

For more information about using these turbulence models in ANSYS FLUENT, see Chapter 12: Modeling Turbulence in the separate User's Guide.

4.1 Introduction

Turbulent flows are characterized by fluctuating velocity fields. These fluctuations mix transported quantities such as momentum, energy, and species concentration, and cause the transported quantities to fluctuate as well. Since these fluctuations can be of small scale and high frequency, they are too computationally expensive to simulate directly in practical engineering calculations. Instead, the instantaneous (exact) governing equations can be time-averaged, ensemble-averaged, or otherwise manipulated to remove the resolution of small scales, resulting in a modified set of equations that are computationally less expensive to solve. However, the modified equations contain additional unknown variables, and turbulence models are needed to determine these variables in terms of known quantities.

ANSYS FLUENT provides the following choices of turbulence models:

- Spalart-Allmaras model
- k-ϵ models
  - Standard k-ϵ model
  - Renormalization-group (RNG) k-ϵ model
  - Realizable k-ϵ model
- k-ω models
  - Standard k-ω model
  - Shear-stress transport (SST) k-ω model
- Transition k-kl-ω model
- Transition SST model
- v^2-f model (add-on)
- Reynolds stress models (RSM)
  - Linear pressure-strain RSM model
  - Quadratic pressure-strain RSM model
  - Low-Re stress-omega RSM model
- Detached eddy simulation (DES) model, which includes one of the following RANS models.
  - Spalart-Allmaras RANS model
  - Realizable k-ϵ RANS model
  - SST k-ω RANS model
- Large eddy simulation (LES) model, which includes one of the following sub-scale models.
  - Smagorinsky-Lilly subgrid-scale model
  - WALE subgrid-scale model
  - Dynamic Smagorinsky model
  - Kinetic-energy transport subgrid-scale model

4.2 Choosing a Turbulence Model

It is an unfortunate fact that no single turbulence model is universally accepted as being superior for all classes of problems. The choice of turbulence model will depend on considerations such as the physics encompassed in the flow, the established practice for a specific class of problem, the level of accuracy required, the available computational resources, and the amount of time available for the simulation. To make the most appropriate choice of model for your application, you need to understand the capabilities and limitations of the various options.

The purpose of this section is to give an overview of issues related to the turbulence models provided in ANSYS FLUENT. The computational effort and cost in terms of CPU time and memory of the individual models is discussed. While it is impossible to state categorically which model is best for a specific application, general guidelines are presented to help you choose the appropriate turbulence model for the flow you want to model.

Information is presented in the following sections:

- Section 4.2.1: Reynolds-Averaged Approach vs. LES
- Section 4.2.2: Reynolds (Ensemble) Averaging
- Section 4.2.3: Boussinesq Approach vs. Reynolds Stress Transport Models

4.2.1 Reynolds-Averaged Approach vs. LES

Time-dependent solutions of the Navier-Stokes equations for high Reynolds-number turbulent flows in complex geometries which set out to resolve all the way down to the smallest scales of the motions are unlikely to be attainable for some time to come. Two alternative methods can be employed to render the Navier-Stokes equations tractable so that the small-scale turbulent fluctuations do not have to be directly simulated: Reynolds-averaging (or ensemble-averaging) and filtering. Both methods introduce additional terms in the governing equations that need to be modeled in order to achieve a "closure" for the unknowns.

The Reynolds-averaged Navier-Stokes (RANS) equations govern the transport of the averaged flow quantities, with the whole range of the scales of turbulence being modeled. The RANS-based modeling approach therefore greatly reduces the required computational effort and resources, and is widely adopted for practical engineering applications. An entire hierarchy of closure models are available in ANSYS FLUENT including Spalart-Allmaras, k-ϵ and its variants, k-ω and its variants, and the RSM. The RANS equations are often used to compute time-dependent flows, whose unsteadiness may be externally imposed (e.g., time-dependent boundary conditions or sources) or self-sustained (e.g., vortex-shedding, flow instabilities).

LES provides an alternative approach in which large eddies are explicitly computed (resolved) in a time-dependent simulation using the "filtered" Navier-Stokes equations. The rationale behind LES is that by modeling less of turbulence (and resolving more), the error introduced by turbulence modeling can be reduced. It is also believed to be easier to find a "universal" model for the small scales, since they tend to be more isotropic and less affected by the macroscopic features like boundary conditions, than the large eddies. Filtering is essentially a mathematical manipulation of the exact Navier-Stokes equations to remove the eddies that are smaller than the size of the filter, which is usually taken as the mesh size when spatial filtering is employed as in ANSYS FLUENT. Like Reynolds-averaging, the filtering process creates additional unknown terms that must be modeled to achieve closure. Statistics of the time-varying flow-fields such as time-averages and r.m.s. values of the solution variables, which are generally of most engineering interest, can be collected during the time-dependent simulation.

LES for high Reynolds number industrial flows requires a significant amount of computational resources. This is mainly because of the need to accurately resolve the energy-containing turbulent eddies in both space and time domains, which becomes most acute in near-wall regions where the scales to be resolved become much smaller. Wall functions in combination with a coarse near wall mesh can be employed, often with some success, to reduce the cost of LES for wall-bounded flows. However, one needs to carefully consider the ramification of using wall functions for the flow in question. For the same reason (to accurately resolve the eddies), LES also requires highly accurate spatial and temporal discretizations.

4.2.2 Reynolds (Ensemble) Averaging

In Reynolds averaging, the solution variables in the instantaneous (exact) Navier-Stokes equations are decomposed into the mean (ensemble-averaged or time-averaged) and fluctuating components. For the velocity components:

$$u_i = \bar{u}_i + u'_i \tag{4.2-1}$$

where $\bar{u}_i$ and u'_i are the mean and fluctuating velocity components ($i = 1, 2, 3$).

Likewise, for pressure and other scalar quantities:

$$\phi = \bar{\phi} + \phi' \tag{4.2-2}$$

where ϕ denotes a scalar such as pressure, energy, or species concentration.

Substituting expressions of this form for the flow variables into the instantaneous continuity and momentum equations and taking a time (or ensemble) average (and dropping the overbar on the mean velocity, $\bar{u}$) yields the ensemble-averaged momentum equations. They can be written in Cartesian tensor form as:

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_i}(\rho u_i) = 0 \tag{4.2-3}$$

$$\frac{\partial}{\partial t}(\rho u_i)+\frac{\partial}{\partial x_j}(\rho u_i u_j) = -\frac{\partial p}{\partial x_i}+\frac{\partial}{\partial x_j}\left[\mu\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}-\frac{2}{3}\delta_{ij}\frac{\partial u_l}{\partial x_l}\right)\right]+\frac{\partial}{\partial x_j}(-\rho\overline{u_i' u_j'}) \tag{4.2-4}$$

Equations 4.2-3 and 4.2-4 are called Reynolds-averaged Navier-Stokes (RANS) equations. They have the same general form as the instantaneous Navier-Stokes equations, with the velocities and other solution variables now representing ensemble-averaged (or time-averaged) values. Additional terms now appear that represent the effects of turbulence. These Reynolds stresses, $-\rho\overline{u_i' u_j'}$, must be modeled in order to close Equation 4.2-4.

For variable-density flows, Equations 4.2-3 and 4.2-4 can be interpreted as Favre-averaged Navier-Stokes equations [130], with the velocities representing mass-averaged values. As such, Equations 4.2-3 and 4.2-4 can be applied to density-varying flows.

4.2.3 Boussinesq Approach vs. Reynolds Stress Transport Models

The Reynolds-averaged approach to turbulence modeling requires that the Reynolds stresses in Equation 4.2-4 are appropriately modeled. A common method employs the Boussinesq hypothesis [130] to relate the Reynolds stresses to the mean velocity gradients:

$$-\rho\overline{u_i' u_j'} = \mu_t\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right)-\frac{2}{3}\left(\rho k+\mu_t\frac{\partial u_k}{\partial x_k}\right)\delta_{ij} \tag{4.2-5}$$

The Boussinesq hypothesis is used in the Spalart-Allmaras model, the k-ϵ models, and the k-ω models. The advantage of this approach is the relatively low computational cost associated with the computation of the turbulent viscosity, μ_t. In the case of the Spalart-Allmaras model, only one additional transport equation (representing turbulent viscosity) is solved. In the case of the k-ϵ and k-ω models, two additional transport equations (for the turbulence kinetic energy, k, and either the turbulence dissipation rate, ϵ, or the specific dissipation rate, ω) are solved, and μ_t is computed as a function of k and ϵ or k and ω. The disadvantage of the Boussinesq hypothesis as presented is that it assumes μ_t is an isotropic scalar quantity, which is not strictly true.

The alternative approach, embodied in the RSM, is to solve transport equations for each of the terms in the Reynolds stress tensor. An additional scale-determining equation (normally for ϵ) is also required. This means that five additional transport equations are required in 2D flows and seven additional transport equations must be solved in 3D.

In many cases, models based on the Boussinesq hypothesis perform very well, and the additional computational expense of the Reynolds stress model is not justified. However, the RSM is clearly superior in situations where the anisotropy of turbulence has a dominant effect on the mean flow. Such cases include highly swirling flows and stress-driven secondary flows.

4.3 Spalart-Allmaras Model

This section describes the theory behind the Spalart-Allmaras model. Information is presented in the following sections:

- Section 4.3.1: Overview
- Section 4.3.2: Transport Equation for the Spalart-Allmaras Model
- Section 4.3.3: Modeling the Turbulent Viscosity
- Section 4.3.4: Modeling the Turbulent Production
- Section 4.3.5: Modeling the Turbulent Destruction
- Section 4.3.6: Model Constants
- Section 4.3.7: Wall Boundary Conditions
- Section 4.3.8: Convective Heat and Mass Transfer Modeling

For details about using the model in ANSYS FLUENT, see Chapter 12: Modeling Turbulence and Section 12.5: Setting Up the Spalart-Allmaras Model in the separate User's Guide.

4.3.1 Overview

The Spalart-Allmaras model is a relatively simple one-equation model that solves a modeled transport equation for the kinematic eddy (turbulent) viscosity. This embodies a relatively new class of one-equation models in which it is not necessary to calculate a length scale related to the local shear layer thickness. The Spalart-Allmaras model was designed specifically for aerospace applications involving wall-bounded flows and has been shown to give good results for boundary layers subjected to adverse pressure gradients. It is also gaining popularity in the turbomachinery applications.

In its original form, the Spalart-Allmaras model is effectively a low-Reynolds-number model, requiring the viscosity-affected region of the boundary layer to be properly resolved. In ANSYS FLUENT, however, the Spalart-Allmaras model has been implemented to use wall functions when the mesh resolution is not sufficiently fine. This might make it the best choice for relatively crude simulations on coarse meshes where accurate turbulent flow computations are not critical. Furthermore, the near-wall gradients of the transported variable in the model are much smaller than the gradients of the transported variables in the k-ϵ or k-ω models. This might make the model less sensitive to numerical errors when non-layered meshes are used near walls. See Section 6.1.3: Numerical Diffusion in the separate User's Guide for a further discussion of the numerical errors.

On a cautionary note, however, the Spalart-Allmaras model is still relatively new, and no claim is made regarding its suitability to all types of complex engineering flows. For instance, it cannot be relied on to predict the decay of homogeneous, isotropic turbulence. Furthermore, one-equation models are often criticized for their inability to rapidly accommodate changes in length scale, such as might be necessary when the flow changes abruptly from a wall-bounded to a free shear flow.

In turbulence models that employ the Boussinesq approach, the central issue is how the eddy viscosity is computed. The model proposed by Spalart and Allmaras [331] solves a transport equation for a quantity that is a modified form of the turbulent kinematic viscosity.

4.3.2 Transport Equation for the Spalart-Allmaras Model

The transported variable in the Spalart-Allmaras model, $\tilde{\nu}$, is identical to the turbulent kinematic viscosity except in the near-wall (viscosity-affected) region. The transport equation for $\tilde{\nu}$ is

$$\frac{\partial}{\partial t}(\rho\tilde{\nu}) + \frac{\partial}{\partial x_i}(\rho\tilde{\nu}u_i) = G_\nu + \frac{1}{\sigma_{\tilde{\nu}}}\left[\frac{\partial}{\partial x_j}\left\{(\mu+\rho\tilde{\nu})\frac{\partial\tilde{\nu}}{\partial x_j}\right\} + C_{b2}\rho\left(\frac{\partial\tilde{\nu}}{\partial x_j}\right)^2\right] - Y_\nu + S_{\tilde{\nu}} \quad (4.3\text{-}1)$$

where G_ν is the production of turbulent viscosity, and Y_ν is the destruction of turbulent viscosity that occurs in the near-wall region due to wall blocking and viscous damping. $\sigma_{\tilde{\nu}}$ and C_{b2} are the constants and ν is the molecular kinematic viscosity. $S_{\tilde{\nu}}$ is a user-defined source term. Note that since the turbulence kinetic energy, k, is not calculated in the Spalart-Allmaras model, while the last term in Equation 4.2-5 is ignored when estimating the Reynolds stresses.

4.3.3 Modeling the Turbulent Viscosity

The turbulent viscosity, μ_t, is computed from

$$\mu_t = \rho\widetilde{\nu} f_{v1} \tag{4.3-2}$$

where the viscous damping function, f_{v1}, is given by

$$f_{v1} = \frac{\chi^3}{\chi^3 + C_{v1}^3} \tag{4.3-3}$$

and

$$\chi \equiv \frac{\widetilde{\nu}}{\nu} \tag{4.3-4}$$

4.3.4 Modeling the Turbulent Production

The production term, G_ν, is modeled as

$$G_\nu = C_{b1}\rho\widetilde{S}\widetilde{\nu} \tag{4.3-5}$$

where

$$\widetilde{S} \equiv S + \frac{\widetilde{\nu}}{\kappa^2 d^2} f_{v2} \tag{4.3-6}$$

and

$$f_{v2} = 1 - \frac{\chi}{1 + \chi f_{v1}} \tag{4.3-7}$$

C_{b1} and κ are constants, d is the distance from the wall, and S is a scalar measure of the deformation tensor. By default in ANSYS FLUENT, as in the original model proposed by Spalart and Allmaras, S is based on the magnitude of the vorticity:

$$S \equiv \sqrt{2\Omega_{ij}\Omega_{ij}} \tag{4.3-8}$$

where Ω_{ij} is the mean rate-of-rotation tensor and is defined by

$$\Omega_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i}\right) \tag{4.3-9}$$

The justification for the default expression for S is that, in the wall-bounded flows that were of most interest when the model was formulated, the turbulence production found only where vorticity is generated near walls. However, it has since been acknowledged that one should also take into account the effect of mean strain on the turbulence production, and a modification to the model has been proposed [65] and incorporated into ANSYS FLUENT.

This modification combines the measures of both vorticity and the strain tensors in the definition of S:

$$S \equiv |\Omega_{ij}| + C_{\rm prod} \; \min\left(0, |S_{ij}| - |\Omega_{ij}|\right) \tag{4.3-10}$$

where

$$C_{\rm prod} = 2.0, \quad |\Omega_{ij}| \equiv \sqrt{2\Omega_{ij}\Omega_{ij}}, \quad |S_{ij}| \equiv \sqrt{2S_{ij}S_{ij}}$$

with the mean strain rate, S_{ij}, defined as

$$S_{ij} = \frac{1}{2}\left(\frac{\partial u_j}{\partial x_i} + \frac{\partial u_i}{\partial x_j}\right) \tag{4.3-11}$$

Including both the rotation and strain tensors reduces the production of eddy viscosity and consequently reduces the eddy viscosity itself in regions where the measure of vorticity exceeds that of strain rate. One such example can be found in vortical flows, i.e., flow near the core of a vortex subjected to a pure rotation where turbulence is known to be suppressed. Including both the rotation and strain tensors more correctly accounts for the effects of rotation on turbulence. The default option (including the rotation tensor only) tends to overpredict the production of eddy viscosity and hence overpredicts the eddy viscosity itself in certain circumstances.

You can select the modified form for calculating production in the **Viscous Model** dialog box.

4.3.5 Modeling the Turbulent Destruction

The destruction term is modeled as

$$Y_\nu = C_{w1} \rho f_w \left(\frac{\widetilde{\nu}}{d}\right)^2 \tag{4.3-12}$$

where

$$f_w = g \left[\frac{1 + C_{w3}^6}{g^6 + C_{w3}^6}\right]^{1/6} \tag{4.3-13}$$

$$g = r + C_{w2} \left(r^6 - r\right) \tag{4.3-14}$$

$$r \equiv \frac{\widetilde{\nu}}{\widetilde{S} \kappa^2 d^2} \tag{4.3-15}$$

C_{w1}, C_{w2}, and C_{w3} are constants, and $\widetilde{S}$ is given by Equation 4.3-6. Note that the modification described above to include the effects of mean strain on S will also affect the value of $\widetilde{S}$ used to compute r.

4.3.6 Model Constants

The model constants $C_{b1}, C_{b2}, \sigma_{\widetilde{\nu}}, C_{v1}, C_{w1}, C_{w2}, C_{w3}$, and κ have the following default values [331]:

$$C_{b1} = 0.1355, \quad C_{b2} = 0.622, \quad \sigma_{\widetilde{\nu}} = \frac{2}{3}, \quad C_{v1} = 7.1$$

$$C_{w1} = \frac{C_{b1}}{\kappa^2} + \frac{(1 + C_{b2})}{\sigma_{\widetilde{\nu}}}, \quad C_{w2} = 0.3, \quad C_{w3} = 2.0, \quad \kappa = 0.4187$$

4.3.7 Wall Boundary Conditions

At walls, the modified turbulent kinematic viscosity, $\widetilde{\nu}$, is set to zero.

When the mesh is fine enough to resolve the viscosity-dominated sublayer, the wall shear stress is obtained from the laminar stress-strain relationship:

$$\frac{u}{u_\tau} = \frac{\rho u_\tau y}{\mu} \tag{4.3-16}$$

If the mesh is too coarse to resolve the viscous sublayer, then it is assumed that the centroid of the wall-adjacent cell falls within the logarithmic region of the boundary layer, and the law-of-the-wall is employed:

$$\frac{u}{u_\tau} = \frac{1}{\kappa} \ln E \left(\frac{\rho u_\tau y}{\mu} \right) \tag{4.3-17}$$

where u is the velocity parallel to the wall, u_τ is the shear velocity, y is the distance from the wall, κ is the von Kármán constant (0.4187), and $E = 9.793$.

4.3.8 Convective Heat and Mass Transfer Modeling

In ANSYS FLUENT, turbulent heat transport is modeled using the concept of the Reynolds' analogy to turbulent momentum transfer. The "modeled" energy equation is as follows:

$$\frac{\partial}{\partial t}(\rho E) + \frac{\partial}{\partial x_i}[u_i(\rho E + p)] = \frac{\partial}{\partial x_j}\left[\left(k + \frac{c_p \mu_t}{\mathrm{Pr}_t}\right)\frac{\partial T}{\partial x_j} + u_i(\tau_{ij})_{\rm eff}\right] + S_h \tag{4.3-18}$$

where k, in this case, is the thermal conductivity, E is the total energy, and $(\tau_{ij})_{\rm eff}$ is the deviatoric stress tensor, defined as

$$(\tau_{ij})_{\rm eff} = \mu_{\rm eff}\left(\frac{\partial u_j}{\partial x_i} + \frac{\partial u_i}{\partial x_j}\right) - \frac{2}{3}\mu_{\rm eff}\frac{\partial u_k}{\partial x_k}\delta_{ij}$$

4.4 Standard, RNG, and Realizable k-ϵ Models

This section describes the theory behind the Standard, RNG, and Realizable k-ϵ models. Information is presented in the following sections:

- Section 4.4.1: Standard k-ϵ Model
- Section 4.4.2: RNG k-ϵ Model
- Section 4.4.3: Realizable k-ϵ Model
- Section 4.4.4: Modeling Turbulent Production in the k-ϵ Models
- Section 4.4.5: Effects of Buoyancy on Turbulence in the k-ϵ Models
- Section 4.4.6: Effects of Compressibility on Turbulence in the k-ϵ Models
- Section 4.4.7: Convective Heat and Mass Transfer Modeling in the k-ϵ Models

For details about using the models in ANSYS FLUENT, see Chapter 12: Modeling Turbulence and Section 12.6: Setting Up the k-ϵ Model in the separate User's Guide.

This section presents the standard, RNG, and realizable k-ϵ models. All three models have similar forms, with transport equations for k and ϵ. The major differences in the models are as follows:

- the method of calculating turbulent viscosity
- the turbulent Prandtl numbers governing the turbulent diffusion of k and ϵ
- the generation and destruction terms in the ϵ equation

The transport equations, the methods of calculating turbulent viscosity, and model constants are presented separately for each model. The features that are essentially common to all models follow, including turbulent generation due to shear buoyancy, accounting for the effects of compressibility, and modeling heat and mass transfer.

4.4.1 Standard k-ϵ Model

Overview

The simplest "complete models" of turbulence are the two-equation models in which the solution of two separate transport equations allows the turbulent velocity and length scales to be independently determined. The standard k-ϵ model in ANSYS FLUENT falls within this class of models and has become the workhorse of practical engineering flow calculations in the time since it was proposed by Launder and Spalding [180]. Robustness, economy, and reasonable accuracy for a wide range of turbulent flows explain its popularity in industrial flow and heat transfer simulations. It is a semi-empirical model, and the derivation of the model equations relies on phenomenological considerations and empiricism.

As the strengths and weaknesses of the standard k-ϵ model have become known, improvements have been made to the model to improve its performance. Two of these variants are available in ANSYS FLUENT: the RNG k-ϵ model [384] and the realizable k-ϵ model [313].

The standard k-ϵ model [180] is a semi-empirical model based on model transport equations for the turbulence kinetic energy (k) and its dissipation rate (ϵ). The model transport equation for k is derived from the exact equation, while the model transport equation for ϵ was obtained using physical reasoning and bears little resemblance to its mathematically exact counterpart.

In the derivation of the k-ϵ model, the assumption is that the flow is fully turbulent, and the effects of molecular viscosity are negligible. The standard k-ϵ model is therefore valid only for fully turbulent flows.

Transport Equations for the Standard k-ϵ Model

The turbulence kinetic energy, k, and its rate of dissipation, ϵ, are obtained from the following transport equations:

$$\frac{\partial}{\partial t}(\rho k) + \frac{\partial}{\partial x_i}(\rho k u_i) = \frac{\partial}{\partial x_j}\left[\left(\mu + \frac{\mu_t}{\sigma_k}\right)\frac{\partial k}{\partial x_j}\right] + G_k + G_b - \rho\epsilon - Y_M + S_k \tag{4.4-1}$$

and

$$\frac{\partial}{\partial t}(\rho \epsilon) + \frac{\partial}{\partial x_i}(\rho \epsilon u_i) = \frac{\partial}{\partial x_j}\left[\left(\mu + \frac{\mu_t}{\sigma_\epsilon}\right)\frac{\partial \epsilon}{\partial x_j}\right] + C_{1\epsilon}\frac{\epsilon}{k}\left(G_k + C_{3\epsilon}G_b\right) - C_{2\epsilon}\rho\frac{\epsilon^2}{k} + S_\epsilon \tag{4.4-2}$$

In these equations, G_k represents the generation of turbulence kinetic energy due to the mean velocity gradients, calculated as described in Section 4.4.4: Modeling Turbulent Production in the k-ϵ Models. G_b is the generation of turbulence kinetic energy due to buoyancy, calculated as described in Section 4.4.5: Effects of Buoyancy on Turbulence in the k-ϵ Models. Y_M represents the contribution of the fluctuating dilatation in compressible turbulence to the overall dissipation rate, calculated as described in Section 4.4.6: Effects of Compressibility on Turbulence in the k-ϵ Models. $C_{1\epsilon}$, $C_{2\epsilon}$, and $C_{3\epsilon}$ are constants. σ_k and σ_ϵ are the turbulent Prandtl numbers for k and ϵ, respectively. S_k and S_ϵ are user-defined source terms.

Modeling the Turbulent Viscosity

The turbulent (or eddy) viscosity, μ_t, is computed by combining k and ϵ as follows:

$$\mu_t = \rho C_\mu \frac{k^2}{\epsilon} \tag{4.4-3}$$

where C_μ is a constant.

Model Constants

The model constants $C_{1\epsilon}, C_{2\epsilon}, C_\mu, \sigma_k$, and σ_ϵ have the following default values [180]:

$$C_{1\epsilon} = 1.44, \quad C_{2\epsilon} = 1.92, \quad C_\mu = 0.09, \quad \sigma_k = 1.0, \quad \sigma_\epsilon = 1.3$$

These default values have been determined from experiments with air and water for fundamental turbulent shear flows including homogeneous shear flows and decaying isotropic grid turbulence. They have been found to work fairly well for a wide range of wall-bounded and free shear flows.

Although the default values of the model constants are the standard ones most widely accepted, you can change them (if needed) in the **Viscous Model** dialog box.

4.4.2 RNG k-ϵ Model

Overview

The RNG k-ϵ model was derived using a rigorous statistical technique (called renormalization group theory). It is similar in form to the standard k-ϵ model, but includes the following refinements:

- The RNG model has an additional term in its ϵ equation that significantly improves the accuracy for rapidly strained flows.
- The effect of swirl on turbulence is included in the RNG model, enhancing accuracy for swirling flows.
- The RNG theory provides an analytical formula for turbulent Prandtl numbers, while the standard k-ϵ model uses user-specified, constant values.
- While the standard k-ϵ model is a high-Reynolds-number model, the RNG theory provides an analytically-derived differential formula for effective viscosity that accounts for low-Reynolds-number effects. Effective use of this feature does, however, depend on an appropriate treatment of the near-wall region.

These features make the RNG k-ϵ model more accurate and reliable for a wider class of flows than the standard k-ϵ model.

The RNG-based k-ϵ turbulence model is derived from the instantaneous Navier-Stokes equations, using a mathematical technique called "renormalization group" (RNG) methods. The analytical derivation results in a model with constants different from those in the standard k-ϵ model, and additional terms and functions in the transport equations for k and ϵ. A more comprehensive description of RNG theory and its application to turbulence can be found in [259].

Transport Equations for the RNG k-ϵ Model

The RNG k-ϵ model has a similar form to the standard k-ϵ model:

$$\frac{\partial}{\partial t}(\rho k) + \frac{\partial}{\partial x_i}(\rho k u_i) = \frac{\partial}{\partial x_j}\left(\alpha_k \mu_{\rm eff} \frac{\partial k}{\partial x_j}\right) + G_k + G_b - \rho\epsilon - Y_M + S_k \qquad (4.4\text{-}4)$$

and

$$\frac{\partial}{\partial t}(\rho \epsilon) + \frac{\partial}{\partial x_i}(\rho \epsilon u_i) = \frac{\partial}{\partial x_j}\left(\alpha_\epsilon \mu_{\rm eff} \frac{\partial \epsilon}{\partial x_j}\right) + C_{1\epsilon}\frac{\epsilon}{k}\left(G_k + C_{3\epsilon}G_b\right) - C_{2\epsilon}\rho\frac{\epsilon^2}{k} - R_\epsilon + S_\epsilon \quad (4.4\text{-}5)$$

In these equations, G_k represents the generation of turbulence kinetic energy due to the mean velocity gradients, calculated as described in Section 4.4.4: Modeling Turbulent Production in the k-ϵ Models. G_b is the generation of turbulence kinetic energy due to buoyancy, calculated as described in Section 4.4.5: Effects of Buoyancy on Turbulence in the k-ϵ Models. Y_M represents the contribution of the fluctuating dilatation in compressible turbulence to the overall dissipation rate, calculated as described in Section 4.4.6: Effects of Compressibility on Turbulence in the k-ϵ Models. The quantities α_k and α_ϵ are the inverse effective Prandtl numbers for k and ϵ, respectively. S_k and S_ϵ are user-defined source terms.

Modeling the Effective Viscosity

The scale elimination procedure in RNG theory results in a differential equation for turbulent viscosity:

$$d\left(\frac{\rho^2 k}{\sqrt{\epsilon\mu}}\right) = 1.72 \frac{\hat{\nu}}{\sqrt{\hat{\nu}^3 - 1 + C_\nu}} d\hat{\nu} \qquad (4.4\text{-}6)$$

where

$$\begin{aligned} \hat{\nu} &= \mu_{\rm eff}/\mu \\ C_\nu &\approx 100 \end{aligned}$$

Equation 4.4-6 is integrated to obtain an accurate description of how the effective turbulent transport varies with the effective Reynolds number (or eddy scale), allowing the model to better handle low-Reynolds-number and near-wall flows.

In the high-Reynolds-number limit, Equation 4.4-6 gives

$$\mu_t = \rho C_\mu \frac{k^2}{\epsilon} \tag{4.4-7}$$

with $C_\mu = 0.0845$, derived using RNG theory. It is interesting to note that this value of C_μ is very close to the empirically-determined value of 0.09 used in the standard k-ϵ model.

In ANSYS FLUENT, by default, the effective viscosity is computed using the high-Reynolds-number form in Equation 4.4-7. However, there is an option available that allows you to use the differential relation given in Equation 4.4-6 when you need to include low-Reynolds-number effects.

RNG Swirl Modification

Turbulence, in general, is affected by rotation or swirl in the mean flow. The RNG model in ANSYS FLUENT provides an option to account for the effects of swirl or rotation by modifying the turbulent viscosity appropriately. The modification takes the following functional form:

$$\mu_t = \mu_{t0} \;\; f\left(\alpha_s, \Omega, \frac{k}{\epsilon}\right) \tag{4.4-8}$$

where μ_{t0} is the value of turbulent viscosity calculated without the swirl modification using either Equation 4.4-6 or Equation 4.4-7. Ω is a characteristic swirl number evaluated within ANSYS FLUENT, and α_s is a swirl constant that assumes different values depending on whether the flow is swirl-dominated or only mildly swirling. This swirl modification always takes effect for axisymmetric, swirling flows and three-dimensional flows when the RNG model is selected. For mildly swirling flows (the default in ANSYS FLUENT), α_s is set to 0.07. For strongly swirling flows, however, a higher value of α_s can be used.

Calculating the Inverse Effective Prandtl Numbers

The inverse effective Prandtl numbers, α_k and α_ϵ, are computed using the following formula derived analytically by the RNG theory:

$$\left|\frac{\alpha - 1.3929}{\alpha_0 - 1.3929}\right|^{0.6321} \left|\frac{\alpha + 2.3929}{\alpha_0 + 2.3929}\right|^{0.3679} = \frac{\mu_{\rm mol}}{\mu_{\rm eff}} \tag{4.4-9}$$

where $\alpha_0 = 1.0$. In the high-Reynolds-number limit ($\mu_{\rm mol}/\mu_{\rm eff} \ll 1$), $\alpha_k = \alpha_\epsilon \approx 1.393$.

The R_ϵ Term in the ϵ Equation

The main difference between the RNG and standard k-ϵ models lies in the additional term in the ϵ equation given by

$$R_\epsilon = \frac{C_\mu \rho \eta^3 (1 - \eta/\eta_0)}{1 + \beta \eta^3} \frac{\epsilon^2}{k} \tag{4.4-10}$$

where $\eta \equiv Sk/\epsilon$, $\eta_0 = 4.38$, $\beta = 0.012$.

The effects of this term in the RNG ϵ equation can be seen more clearly by rearranging Equation 4.4-5. Using Equation 4.4-10, the third and fourth terms on the right-hand side of Equation 4.4-5 can be merged, and the resulting ϵ equation can be rewritten as

$$\frac{\partial}{\partial t}(\rho \epsilon) + \frac{\partial}{\partial x_i}(\rho \epsilon u_i) = \frac{\partial}{\partial x_j}\left(\alpha_\epsilon \mu_{\rm eff} \frac{\partial \epsilon}{\partial x_j}\right) + C_{1\epsilon} \frac{\epsilon}{k}\left(G_k + C_{3\epsilon} G_b\right) - C^*_{2\epsilon} \rho \frac{\epsilon^2}{k} \tag{4.4-11}$$

where $C^*_{2\epsilon}$ is given by

$$C^*_{2\epsilon} \equiv C_{2\epsilon} + \frac{C_\mu \eta^3 (1 - \eta/\eta_0)}{1 + \beta \eta^3} \tag{4.4-12}$$

In regions where $\eta < \eta_0$, the R term makes a positive contribution, and $C^*_{2\epsilon}$ becomes larger than $C_{2\epsilon}$. In the logarithmic layer, for instance, it can be shown that $\eta \approx 3.0$, giving $C^*_{2\epsilon} \approx 2.0$, which is close in magnitude to the value of $C_{2\epsilon}$ in the standard k-ϵ model (1.92). As a result, for weakly to moderately strained flows, the RNG model tends to give results largely comparable to the standard k-ϵ model.

In regions of large strain rate ($\eta > \eta_0$), however, the R term makes a negative contribution, making the value of $C^*_{2\epsilon}$ less than $C_{2\epsilon}$. In comparison with the standard k-ϵ model, the smaller destruction of ϵ augments ϵ, reducing k and, eventually, the effective viscosity. As a result, in rapidly strained flows, the RNG model yields a lower turbulent viscosity than the standard k-ϵ model.

Thus, the RNG model is more responsive to the effects of rapid strain and streamline curvature than the standard k-ϵ model, which explains the superior performance of the RNG model for certain classes of flows.

Model Constants

The model constants $C_{1\epsilon}$ and $C_{2\epsilon}$ in Equation 4.4-5 have values derived analytically by the RNG theory. These values, used by default in ANSYS FLUENT, are

$$C_{1\epsilon} = 1.42, \quad C_{2\epsilon} = 1.68$$

4.4.3 Realizable k-ϵ Model

Overview

The realizable k-ϵ model [313] is a relatively recent development and differs from the standard k-ϵ model in two important ways:

- The realizable k-ϵ model contains a new formulation for the turbulent viscosity.
- A new transport equation for the dissipation rate, ϵ, has been derived from an exact equation for the transport of the mean-square vorticity fluctuation.

The term "realizable" means that the model satisfies certain mathematical constraints on the Reynolds stresses, consistent with the physics of turbulent flows. Neither the standard k-ϵ model nor the RNG k-ϵ model is realizable.

An immediate benefit of the realizable k-ϵ model is that it more accurately predicts the spreading rate of both planar and round jets. It is also likely to provide superior performance for flows involving rotation, boundary layers under strong adverse pressure gradients, separation, and recirculation.

To understand the mathematics behind the realizable k-ϵ model, consider combining the Boussinesq relationship (Equation 4.2-5) and the eddy viscosity definition (Equation 4.4-3) to obtain the following expression for the normal Reynolds stress in an incompressible strained mean flow:

$$\overline{u^2} = \frac{2}{3}k - 2\,\nu_t \frac{\partial U}{\partial x} \tag{4.4-13}$$

Using Equation 4.4-3 for $\nu_t \equiv \mu_t/\rho$, one obtains the result that the normal stress, $\overline{u^2}$, which by definition is a positive quantity, becomes negative, i.e., "non-realizable", when the strain is large enough to satisfy

$$\frac{k}{\epsilon}\frac{\partial U}{\partial x} > \frac{1}{3C_\mu} \approx 3.7 \tag{4.4-14}$$

Similarly, it can also be shown that the Schwarz inequality for shear stresses ($\overline{u_\alpha u_\beta}^2 \leq \overline{u_\alpha^2 u_\beta^2}$; no summation over α and β) can be violated when the mean strain rate is large. The most straightforward way to ensure the realizability (positivity of normal stresses and Schwarz inequality for shear stresses) is to make C_μ variable by sensitizing it to the mean flow (mean deformation) and the turbulence (k, ϵ). The notion of variable C_μ is suggested by many modelers including Reynolds [291], and is well substantiated by experimental evidence. For example, C_μ is found to be around 0.09 in the inertial sublayer of equilibrium boundary layers, and 0.05 in a strong homogeneous shear flow.

Both the realizable and RNG k-ϵ models have shown substantial improvements over the standard k-ϵ model where the flow features include strong streamline curvature, vortices, and rotation. Since the model is still relatively new, it is not clear in exactly which instances the realizable k-ϵ model consistently outperforms the RNG model. However, initial studies have shown that the realizable model provides the best performance of all the k-ϵ model versions for several validations of separated flows and flows with complex secondary flow features.

One of the weaknesses of the standard k-ϵ model or other traditional k-ϵ models lies with the modeled equation for the dissipation rate (ϵ). The well-known round-jet anomaly (named based on the finding that the spreading rate in planar jets is predicted reasonably well, but prediction of the spreading rate for axisymmetric jets is unexpectedly poor) is considered to be mainly due to the modeled dissipation equation.

The realizable k-ϵ model proposed by Shih et al. [313] was intended to address these deficiencies of traditional k-ϵ models by adopting the following:

- A new eddy-viscosity formula involving a variable C_μ originally proposed by Reynolds [291].
- A new model equation for dissipation (ϵ) based on the dynamic equation of the mean-square vorticity fluctuation.

One limitation of the realizable k-ϵ model is that it produces non-physical turbulent viscosities in situations when the computational domain contains both rotating and stationary fluid zones (e.g., multiple reference frames, rotating sliding meshes). This is due to the fact that the realizable k-ϵ model includes the effects of mean rotation in the definition of the turbulent viscosity (see Equations 4.4-17–4.4-19). This extra rotation effect has been tested on single rotating reference frame systems and showed superior behavior over the standard k-ϵ model. However, due to the nature of this modification, its application to multiple reference frame systems should be taken with some caution. See Section 4.4.3: Modeling the Turbulent Viscosity for information about how to include or exclude this term from the model.

Transport Equations for the Realizable k-ϵ Model

The modeled transport equations for k and ϵ in the realizable k-ϵ model are

$$\frac{\partial}{\partial t}(\rho k) + \frac{\partial}{\partial x_j}(\rho k u_j) = \frac{\partial}{\partial x_j}\left[\left(\mu + \frac{\mu_t}{\sigma_k}\right)\frac{\partial k}{\partial x_j}\right] + G_k + G_b - \rho\epsilon - Y_M + S_k \tag{4.4-15}$$

and

$$\frac{\partial}{\partial t}(\rho \epsilon) + \frac{\partial}{\partial x_j}(\rho \epsilon u_j) = \frac{\partial}{\partial x_j}\left[\left(\mu + \frac{\mu_t}{\sigma_\epsilon}\right)\frac{\partial \epsilon}{\partial x_j}\right] + \rho\, C_1 S\epsilon - \rho\, C_2 \frac{\epsilon^2}{k + \sqrt{\nu\epsilon}} + C_{1\epsilon}\frac{\epsilon}{k}C_{3\epsilon}G_b + S_\epsilon \tag{4.4-16}$$

where

$$C_1 = \max\left[0.43, \frac{\eta}{\eta + 5}\right], \qquad \eta = S\frac{k}{\epsilon}, \qquad S = \sqrt{2S_{ij}S_{ij}}$$

In these equations, G_k represents the generation of turbulence kinetic energy due to the mean velocity gradients, calculated as described in Section 4.4.4: Modeling Turbulent Production in the k-ϵ Models. G_b is the generation of turbulence kinetic energy due to buoyancy, calculated as described in Section 4.4.5: Effects of Buoyancy on Turbulence in the k-ϵ Models. Y_M represents the contribution of the fluctuating dilatation in compressible turbulence to the overall dissipation rate, calculated as described in Section 4.4.6: Effects of Compressibility on Turbulence in the k-ϵ Models. C_2 and $C_{1\epsilon}$ are constants. σ_k and σ_ϵ are the turbulent Prandtl numbers for k and ϵ, respectively. S_k and S_ϵ are user-defined source terms.

Note that the k equation (Equation 4.4-15) is the same as that in the standard k-ϵ model (Equation 4.4-1) and the RNG k-ϵ model (Equation 4.4-4), except for the model constants. However, the form of the ϵ equation is quite different from those in the standard and RNG-based k-ϵ models (Equations 4.4-2 and 4.4-5). One of the noteworthy features is that the production term in the ϵ equation (the second term on the right-hand side of Equation 4.4-16) does not involve the production of k; i.e., it does not contain the same G_k term as the other k-ϵ models. It is believed that the present form better represents the spectral energy transfer. Another desirable feature is that the destruction term (the next to last term on the right-hand side of Equation 4.4-16) does not have any singularity; i.e., its denominator never vanishes, even if k vanishes or becomes smaller than zero. This feature is contrasted with traditional k-ϵ models, which have a singularity due to k in the denominator.

This model has been extensively validated for a wide range of flows [167, 313], including rotating homogeneous shear flows, free flows including jets and mixing layers, channel and boundary layer flows, and separated flows. For all these cases, the performance of the model has been found to be substantially better than that of the standard k-ϵ model. Especially noteworthy is the fact that the realizable k-ϵ model resolves the round-jet anomaly; i.e., it predicts the spreading rate for axisymmetric jets as well as that for planar jets.

Modeling the Turbulent Viscosity

As in other k-ϵ models, the eddy viscosity is computed from

$$\mu_t = \rho C_\mu \frac{k^2}{\epsilon} \tag{4.4-17}$$

The difference between the realizable k-ϵ model and the standard and RNG k-ϵ models is that C_μ is no longer constant. It is computed from

$$C_\mu = \frac{1}{A_0 + A_s \frac{kU^*}{\epsilon}} \tag{4.4-18}$$

where

$$U^* \equiv \sqrt{S_{ij}S_{ij} + \tilde{\Omega}_{ij}\tilde{\Omega}_{ij}} \tag{4.4-19}$$

and

$$\begin{aligned} \tilde{\Omega}_{ij} &= \Omega_{ij} - 2\epsilon_{ijk}\omega_k \\ \Omega_{ij} &= \overline{\Omega_{ij}} - \epsilon_{ijk}\omega_k \end{aligned}$$

where $\overline{\Omega_{ij}}$ is the mean rate-of-rotation tensor viewed in a rotating reference frame with the angular velocity ω_k. The model constants A_0 and A_s are given by

$$A_0 = 4.04, \quad A_s = \sqrt{6}\cos\phi$$

where

$$\phi = \frac{1}{3}\cos^{-1}(\sqrt{6}W), \quad W = \frac{S_{ij}S_{jk}S_{ki}}{\tilde{S}^3}, \quad \tilde{S} = \sqrt{S_{ij}S_{ij}}, \quad S_{ij} = \frac{1}{2}\left(\frac{\partial u_j}{\partial x_i} + \frac{\partial u_i}{\partial x_j}\right)$$

It can be seen that C_μ is a function of the mean strain and rotation rates, the angular velocity of the system rotation, and the turbulence fields (k and ϵ). C_μ in Equation 4.4-17 can be shown to recover the standard value of 0.09 for an inertial sublayer in an equilibrium boundary layer.

i In ANSYS FLUENT, the term $-2\epsilon_{ijk}\omega_k$ is, by default, not included in the calculation of $\widetilde{\Omega}_{ij}$. This is an extra rotation term that is not compatible with cases involving sliding meshes or multiple reference frames. If you want to include this term in the model, you can enable it by using the `define/models/viscous/turbulence-expert/rke-cmu-rotation-term?` text command and entering `yes` at the prompt.

Model Constants

The model constants C_2, σ_k, and σ_ϵ have been established to ensure that the model performs well for certain canonical flows. The model constants are

$$C_{1\epsilon} = 1.44, \quad C_2 = 1.9, \quad \sigma_k = 1.0, \quad \sigma_\epsilon = 1.2$$

4.4.4 Modeling Turbulent Production in the k-ϵ Models

The term G_k, representing the production of turbulence kinetic energy, is modeled identically for the standard, RNG, and realizable k-ϵ models. From the exact equation for the transport of k, this term may be defined as

$$G_k = -\rho \overline{u_i' u_j'} \frac{\partial u_j}{\partial x_i} \tag{4.4-20}$$

To evaluate G_k in a manner consistent with the Boussinesq hypothesis,

$$G_k = \mu_t S^2 \tag{4.4-21}$$

where S is the modulus of the mean rate-of-strain tensor, defined as

$$S \equiv \sqrt{2 S_{ij} S_{ij}} \tag{4.4-22}$$

i When using the high-Reynolds number k-ϵ versions, $\mu_{\rm eff}$ is used in lieu of μ_t in Equation 4.4-21.

4.4.5 Effects of Buoyancy on Turbulence in the k-ϵ Models

When a non-zero gravity field and temperature gradient are present simultaneously, the k-ϵ models in ANSYS FLUENT account for the generation of k due to buoyancy (G_b in Equations 4.4-1, 4.4-4, and 4.4-15), and the corresponding contribution to the production of ϵ in Equations 4.4-2, 4.4-5, and 4.4-16.

The generation of turbulence due to buoyancy is given by

$$G_b = \beta g_i \frac{\mu_t}{\mathrm{Pr}_t} \frac{\partial T}{\partial x_i} \tag{4.4-23}$$

where Pr_t is the turbulent Prandtl number for energy and g_i is the component of the gravitational vector in the ith direction. For the standard and realizable k-ϵ models, the default value of Pr_t is 0.85. In the case of the RNG k-ϵ model, $\mathrm{Pr}_t = 1/\alpha$, where α is given by Equation 4.4-9, but with $\alpha_0 = 1/\mathrm{Pr} = k/\mu c_p$. The coefficient of thermal expansion, β, is defined as

$$\beta = -\frac{1}{\rho}\left(\frac{\partial \rho}{\partial T}\right)_p \tag{4.4-24}$$

For ideal gases, Equation 4.4-23 reduces to

$$G_b = -g_i \frac{\mu_t}{\rho \mathrm{Pr}_t} \frac{\partial \rho}{\partial x_i} \tag{4.4-25}$$

It can be seen from the transport equations for k (Equations 4.4-1, 4.4-4, and 4.4-15) that turbulence kinetic energy tends to be augmented ($G_b > 0$) in unstable stratification. For stable stratification, buoyancy tends to suppress the turbulence ($G_b < 0$). In ANSYS FLUENT, the effects of buoyancy on the generation of k are always included when you have both a non-zero gravity field and a non-zero temperature (or density) gradient.

While the buoyancy effects on the generation of k are relatively well understood, the effect on ϵ is less clear. In ANSYS FLUENT, by default, the buoyancy effects on ϵ are neglected simply by setting G_b to zero in the transport equation for ϵ (Equation 4.4-2, 4.4-5, or 4.4-16).

However, you can include the buoyancy effects on ϵ in the **Viscous Model** dialog box. In this case, the value of G_b given by Equation 4.4-25 is used in the transport equation for ϵ (Equation 4.4-2, 4.4-5, or 4.4-16).

The degree to which ϵ is affected by the buoyancy is determined by the constant $C_{3\epsilon}$. In ANSYS FLUENT, $C_{3\epsilon}$ is not specified, but is instead calculated according to the following relation [127]:

$$C_{3\epsilon} = \tanh\left|\frac{v}{u}\right| \tag{4.4-26}$$

where v is the component of the flow velocity parallel to the gravitational vector and u is the component of the flow velocity perpendicular to the gravitational vector. In this way, $C_{3\epsilon}$ will become 1 for buoyant shear layers for which the main flow direction is aligned with the direction of gravity. For buoyant shear layers that are perpendicular to the gravitational vector, $C_{3\epsilon}$ will become zero.

4.4.6 Effects of Compressibility on Turbulence in the k-ϵ Models

For high-Mach-number flows, compressibility affects turbulence through so-called "dilatation dissipation", which is normally neglected in the modeling of incompressible flows [379]. Neglecting the dilatation dissipation fails to predict the observed decrease in spreading rate with increasing Mach number for compressible mixing and other free shear layers. To account for these effects in the k-ϵ models in ANSYS FLUENT, the dilatation dissipation term, Y_M, is included in the k equation. This term is modeled according to a proposal by Sarkar [300]:

$$Y_M = 2\rho\epsilon \mathrm{M}_t^2 \tag{4.4-27}$$

where M_t is the turbulent Mach number, defined as

$$\mathrm{M}_t = \sqrt{\frac{k}{a^2}} \tag{4.4-28}$$

where a ($\equiv \sqrt{\gamma RT}$) is the speed of sound.

This compressibility modification always takes effect when the compressible form of the ideal gas law is used.

4.4.7 Convective Heat and Mass Transfer Modeling in the k-ϵ Models

In ANSYS FLUENT, turbulent heat transport is modeled using the concept of Reynolds' analogy to turbulent momentum transfer. The "modeled" energy equation is thus given by the following:

$$\frac{\partial}{\partial t}(\rho E) + \frac{\partial}{\partial x_i}[u_i(\rho E + p)] = \frac{\partial}{\partial x_j}\left(k_{\text{eff}}\frac{\partial T}{\partial x_j} + u_i(\tau_{ij})_{\text{eff}}\right) + S_h \tag{4.4-29}$$

where E is the total energy, k_{eff} is the effective thermal conductivity, and

$(\tau_{ij})_{\text{eff}}$ is the deviatoric stress tensor, defined as

$$(\tau_{ij})_{\text{eff}} = \mu_{\text{eff}} \left(\frac{\partial u_j}{\partial x_i} + \frac{\partial u_i}{\partial x_j} \right) - \frac{2}{3} \mu_{\text{eff}} \frac{\partial u_k}{\partial x_k} \delta_{ij}$$

The term involving $(\tau_{ij})_{\text{eff}}$ represents the viscous heating, and is always computed in the density-based solvers. It is not computed by default in the pressure-based solver, but it can be enabled in the **Viscous Model** dialog box.

Additional terms may appear in the energy equation, depending on the physical models you are using. See Section 5.2.1: Heat Transfer Theory for more details.

For the standard and realizable k-ϵ models, the effective thermal conductivity is given by

$$k_{\text{eff}} = k + \frac{c_p \mu_t}{\text{Pr}_t}$$

where k, in this case, is the thermal conductivity. The default value of the turbulent Prandtl number is 0.85. You can change the value of the turbulent Prandtl number in the **Viscous Model** dialog box.

For the RNG k-ϵ model, the effective thermal conductivity is

$$k_{\text{eff}} = \alpha c_p \mu_{\text{eff}}$$

where α is calculated from Equation 4.4-9, but with $\alpha_0 = 1/\text{Pr} = k/\mu c_p$.

The fact that α varies with $\mu_{\text{mol}}/\mu_{\text{eff}}$, as in Equation 4.4-9, is an advantage of the RNG k-ϵ model. It is consistent with experimental evidence indicating that the turbulent Prandtl number varies with the molecular Prandtl number and turbulence [159]. Equation 4.4-9 works well across a very broad range of molecular Prandtl numbers, from liquid metals ($\text{Pr} \approx 10^{-2}$) to paraffin oils ($\text{Pr} \approx 10^3$), which allows heat transfer to be calculated in low-Reynolds-number regions. Equation 4.4-9 smoothly predicts the variation of effective Prandtl number from the molecular value ($\alpha = 1/\text{Pr}$) in the viscosity-dominated region to the fully turbulent value ($\alpha = 1.393$) in the fully turbulent regions of the flow.

Turbulent mass transfer is treated similarly. For the standard and realizable k-ϵ models, the default turbulent Schmidt number is 0.7. This default value can be changed in the **Viscous Model** dialog box. For the RNG model, the effective turbulent diffusivity for mass transfer is calculated in a manner that is analogous to the method used for the heat transport. The value of α_0 in Equation 4.4-9 is $\alpha_0 = 1/\text{Sc}$, where Sc is the molecular Schmidt number.

4.5 Standard and SST k-ω Models

This section describes the theory behind the Standard and SST k-ω model. Information is presented in the following sections:

- Section 4.5.1: Standard k-ω Model
- Section 4.5.2: Shear-Stress Transport (SST) k-ω Model
- Section 4.5.3: Wall Boundary Conditions

For details about using the models in ANSYS FLUENT, see Chapter 12: Modeling Turbulence and Section 12.7: Setting Up the k-ω Model in the separate User's Guide.

This section presents the standard [379] and shear-stress transport (SST) [224] k-ω models. Both models have similar forms, with transport equations for k and ω. The major ways in which the SST model [225] differs from the standard model are as follows:

- gradual change from the standard k-ω model in the inner region of the boundary layer to a high-Reynolds-number version of the k-ϵ model in the outer part of the boundary layer
- modified turbulent viscosity formulation to account for the transport effects of the principal turbulent shear stress

The transport equations, methods of calculating turbulent viscosity, and methods of calculating model constants and other terms are presented separately for each model.

4.5.1 Standard k-ω Model

Overview

The standard k-ω model in ANSYS FLUENT is based on the Wilcox k-ω model [379], which incorporates modifications for low-Reynolds-number effects, compressibility, and shear flow spreading. The Wilcox model predicts free shear flow spreading rates that are in close agreement with measurements for far wakes, mixing layers, and plane, round, and radial jets, and is thus applicable to wall-bounded flows and free shear flows. A variation of the standard k-ω model called the SST k-ω model is also available in ANSYS FLUENT, and is described in Section 4.5.2: Shear-Stress Transport (SST) k-ω Model.

The standard k-ω model is an empirical model based on model transport equations for the turbulence kinetic energy (k) and the specific dissipation rate (ω), which can also be thought of as the ratio of ϵ to k [379].

As the k-ω model has been modified over the years, production terms have been added to both the k and ω equations, which have improved the accuracy of the model for predicting free shear flows.

Transport Equations for the Standard k-ω Model

The turbulence kinetic energy, k, and the specific dissipation rate, ω, are obtained from the following transport equations:

$$\frac{\partial}{\partial t}(\rho k) + \frac{\partial}{\partial x_i}(\rho k u_i) = \frac{\partial}{\partial x_j}\left(\Gamma_k \frac{\partial k}{\partial x_j}\right) + G_k - Y_k + S_k \tag{4.5-1}$$

and

$$\frac{\partial}{\partial t}(\rho \omega) + \frac{\partial}{\partial x_i}(\rho \omega u_i) = \frac{\partial}{\partial x_j}\left(\Gamma_\omega \frac{\partial \omega}{\partial x_j}\right) + G_\omega - Y_\omega + S_\omega \tag{4.5-2}$$

In these equations, G_k represents the generation of turbulence kinetic energy due to mean velocity gradients. G_ω represents the generation of ω. Γ_k and Γ_ω represent the effective diffusivity of k and ω, respectively. Y_k and Y_ω represent the dissipation of k and ω due to turbulence. All of the above terms are calculated as described below. S_k and S_ω are user-defined source terms.

Modeling the Effective Diffusivity

The effective diffusivities for the k-ω model are given by

$$\begin{aligned} \Gamma_k &= \mu + \frac{\mu_t}{\sigma_k} && (4.5\text{-}3) \\ \Gamma_\omega &= \mu + \frac{\mu_t}{\sigma_\omega} && (4.5\text{-}4) \end{aligned}$$

where σ_k and σ_ω are the turbulent Prandtl numbers for k and ω, respectively. The turbulent viscosity, μ_t, is computed by combining k and ω as follows:

$$\mu_t = \alpha^* \frac{\rho k}{\omega} \tag{4.5-5}$$

Low-Reynolds-Number Correction

The coefficient α^* damps the turbulent viscosity causing a low-Reynolds-number correction. It is given by

$$\alpha^* = \alpha^*_\infty \left(\frac{\alpha^*_0 + \mathrm{Re}_t/R_k}{1 + \mathrm{Re}_t/R_k} \right) \tag{4.5-6}$$

where

$$\begin{aligned} \mathrm{Re}_t &= \frac{\rho k}{\mu \omega} && (4.5\text{-}7) \\ R_k &= 6 && (4.5\text{-}8) \\ \alpha^*_0 &= \frac{\beta_i}{3} && (4.5\text{-}9) \\ \beta_i &= 0.072 && (4.5\text{-}10) \end{aligned}$$

Note that, in the high-Reynolds-number form of the k-ω model, $\alpha^* = \alpha^*_\infty = 1$.

Modeling the Turbulence Production

Production of k

The term G_k represents the production of turbulence kinetic energy. From the exact equation for the transport of k, this term may be defined as

$$G_k = -\rho \overline{u'_i u'_j} \frac{\partial u_j}{\partial x_i} \tag{4.5-11}$$

To evaluate G_k in a manner consistent with the Boussinesq hypothesis,

$$G_k = \mu_t \, S^2 \tag{4.5-12}$$

where S is the modulus of the mean rate-of-strain tensor, defined in the same way as for the k-ϵ model (see Equation 4.4-22).

Production of ω

The production of ω is given by

$$G_\omega = \alpha \frac{\omega}{k} G_k \tag{4.5-13}$$

where G_k is given by Equation 4.5-11.

The coefficient α is given by

$$\alpha = \frac{\alpha_\infty}{\alpha^*} \left(\frac{\alpha_0 + \mathrm{Re}_t / R_\omega}{1 + \mathrm{Re}_t / R_\omega} \right) \tag{4.5-14}$$

where $R_\omega = 2.95$. α^* and Re_t are given by Equations 4.5-6 and 4.5-7, respectively.

Note that, in the high-Reynolds-number form of the k-ω model, $\alpha = \alpha_\infty = 1$.

Modeling the Turbulence Dissipation

Dissipation of k

The dissipation of k is given by

$$Y_k = \rho\, \beta^* f_{\beta^*}\, k\, \omega \tag{4.5-15}$$

where

$$f_{\beta^*} = \begin{cases} 1 & \chi_k \le 0 \\ \frac{1+680\chi_k^2}{1+400\chi_k^2} & \chi_k > 0 \end{cases} \tag{4.5-16}$$

where

$$\chi_k \equiv \frac{1}{\omega^3} \frac{\partial k}{\partial x_j} \frac{\partial \omega}{\partial x_j} \tag{4.5-17}$$

and

$$\beta^* = \beta_i^* \left[1 + \zeta^* F(\mathrm{M}_t) \right] \tag{4.5-18}$$

$$\beta_i^* = \beta_\infty^* \left(\frac{4/15 + (\mathrm{Re}_t / R_\beta)^4}{1 + (\mathrm{Re}_t / R_\beta)^4} \right) \tag{4.5-19}$$

$$\zeta^* = 1.5 \tag{4.5-20}$$

$$
\begin{aligned}
R_\beta &= 8 && (4.5\text{-}21)\\
\beta^*_\infty &= 0.09 && (4.5\text{-}22)
\end{aligned}
$$

where Re_t is given by Equation 4.5-7.

Dissipation of ω

The dissipation of ω is given by

$$
Y_\omega = \rho\, \beta\, f_\beta\, \omega^2 \qquad (4.5\text{-}23)
$$

where

$$
\begin{aligned}
f_\beta &= \frac{1 + 70\chi_\omega}{1 + 80\chi_\omega} && (4.5\text{-}24)\\
\chi_\omega &= \left| \frac{\Omega_{ij}\Omega_{jk}S_{ki}}{(\beta^*_\infty \omega)^3} \right| && (4.5\text{-}25)\\
\Omega_{ij} &= \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i} \right) && (4.5\text{-}26)
\end{aligned}
$$

The strain rate tensor, S_{ij} is defined in Equation 4.3-11. Also,

$$
\beta = \beta_i \left[1 - \frac{\beta^*_i}{\beta_i} \zeta^* F(\mathrm{M}_t) \right] \qquad (4.5\text{-}27)
$$

β^*_i and $F(\mathrm{M}_t)$ are defined by Equations 4.5-19 and 4.5-28, respectively.

Compressibility Correction

The compressibility function, $F(\mathrm{M}_t)$, is given by

$$
F(\mathrm{M}_t) = \begin{cases} 0 & \mathrm{M}_t \le \mathrm{M}_{t0} \\ \mathrm{M}_t^2 - \mathrm{M}_{t0}^2 & \mathrm{M}_t > \mathrm{M}_{t0} \end{cases} \qquad (4.5\text{-}28)
$$

where

$$
\begin{aligned}
\mathrm{M}_t^2 &\equiv \frac{2k}{a^2} && (4.5\text{-}29)\\
\mathrm{M}_{t0} &= 0.25 && (4.5\text{-}30)\\
a &= \sqrt{\gamma R T} && (4.5\text{-}31)
\end{aligned}
$$

Note that, in the high-Reynolds-number form of the k-ω model, $\beta_i^* = \beta_\infty^*$. In the incompressible form, $\beta^* = \beta_i^*$.

Model Constants

$$\alpha_\infty^* = 1, \quad \alpha_\infty = 0.52, \quad \alpha_0 = \frac{1}{9}, \quad \beta_\infty^* = 0.09, \quad \beta_i = 0.072, \quad R_\beta = 8$$

$$R_k = 6, \quad R_\omega = 2.95, \quad \zeta^* = 1.5, \quad \mathrm{M}_{t0} = 0.25, \quad \sigma_k = 2.0, \quad \sigma_\omega = 2.0$$

4.5.2 Shear-Stress Transport (SST) k-ω Model

Overview

The shear-stress transport (SST) k-ω model was developed by Menter [224] to effectively blend the robust and accurate formulation of the k-ω model in the near-wall region with the free-stream independence of the k-ϵ model in the far field. To achieve this, the k-ϵ model is converted into a k-ω formulation. The SST k-ω model is similar to the standard k-ω model, but includes the following refinements:

- The standard k-ω model and the transformed k-ϵ model are both multiplied by a blending function and both models are added together. The blending function is designed to be one in the near-wall region, which activates the standard k-ω model, and zero away from the surface, which activates the transformed k-ϵ model.
- The SST model incorporates a damped cross-diffusion derivative term in the ω equation.
- The definition of the turbulent viscosity is modified to account for the transport of the turbulent shear stress.
- The modeling constants are different.

These features make the SST k-ω model more accurate and reliable for a wider class of flows (e.g., adverse pressure gradient flows, airfoils, transonic shock waves) than the standard k-ω model. Other modifications include the addition of a cross-diffusion term in the ω equation and a blending function to ensure that the model equations behave appropriately in both the near-wall and far-field zones.

Transport Equations for the SST k-ω Model

The SST k-ω model has a similar form to the standard k-ω model:

$$\frac{\partial}{\partial t}(\rho k) + \frac{\partial}{\partial x_i}(\rho k u_i) = \frac{\partial}{\partial x_j}\left(\Gamma_k \frac{\partial k}{\partial x_j}\right) + \widetilde{G}_k - Y_k + S_k \tag{4.5-32}$$

and

$$\frac{\partial}{\partial t}(\rho \omega) + \frac{\partial}{\partial x_i}(\rho \omega u_i) = \frac{\partial}{\partial x_j}\left(\Gamma_\omega \frac{\partial \omega}{\partial x_j}\right) + G_\omega - Y_\omega + D_\omega + S_\omega \tag{4.5-33}$$

In these equations, $\widetilde{G}_k$ represents the generation of turbulence kinetic energy due to mean velocity gradients, calculated as described in Section 4.5.1: Modeling the Turbulence Production. G_ω represents the generation of ω, calculated as described in Section 4.5.1: Modeling the Turbulence Production. Γ_k and Γ_ω represent the effective diffusivity of k and ω, respectively, which are calculated as described below. Y_k and Y_ω represent the dissipation of k and ω due to turbulence, calculated as described in Section 4.5.1: Modeling the Turbulence Dissipation. D_ω represents the cross-diffusion term, calculated as described below. S_k and S_ω are user-defined source terms.

Modeling the Effective Diffusivity

The effective diffusivities for the SST k-ω model are given by

$$\Gamma_k = \mu + \frac{\mu_t}{\sigma_k} \tag{4.5-34}$$

$$\Gamma_\omega = \mu + \frac{\mu_t}{\sigma_\omega} \tag{4.5-35}$$

where σ_k and σ_ω are the turbulent Prandtl numbers for k and ω, respectively. The turbulent viscosity, μ_t, is computed as follows:

$$\mu_t = \frac{\rho k}{\omega} \frac{1}{\max\left[\frac{1}{\alpha^*}, \frac{S F_2}{a_1 \omega}\right]} \tag{4.5-36}$$

where S is the strain rate magnitude and

$$\sigma_k = \frac{1}{F_1/\sigma_{k,1} + (1 - F_1)/\sigma_{k,2}} \tag{4.5-37}$$

$$\sigma_\omega = \frac{1}{F_1/\sigma_{\omega,1} + (1 - F_1)/\sigma_{\omega,2}} \tag{4.5-38}$$

α^* is defined in Equation 4.5-6. The blending functions, F_1 and F_2, are given by

$$\begin{aligned} F_1 &= \tanh\left(\Phi_1^4\right) && (4.5\text{-}39)\\ \Phi_1 &= \min\left[\max\left(\frac{\sqrt{k}}{0.09\omega y}, \frac{500\mu}{\rho y^2\omega}\right), \frac{4\rho k}{\sigma_{\omega,2} D_\omega^+ y^2}\right] && (4.5\text{-}40)\\ D_\omega^+ &= \max\left[2\rho\frac{1}{\sigma_{\omega,2}}\frac{1}{\omega}\frac{\partial k}{\partial x_j}\frac{\partial \omega}{\partial x_j}, 10^{-10}\right] && (4.5\text{-}41) \end{aligned}$$

$$\begin{aligned} F_2 &= \tanh\left(\Phi_2^2\right) && (4.5\text{-}42)\\ \Phi_2 &= \max\left[2\frac{\sqrt{k}}{0.09\omega y}, \frac{500\mu}{\rho y^2\omega}\right] && (4.5\text{-}43) \end{aligned}$$

where y is the distance to the next surface and D_ω^+ is the positive portion of the cross-diffusion term (see Equation 4.5-52).

Modeling the Turbulence Production

Production of k

The term $\widetilde{G}_k$ represents the production of turbulence kinetic energy, and is defined as:

$$\widetilde{G}_k = \min(G_k, 10\rho\beta^* k\omega) \tag{4.5-44}$$

where G_k is defined in the same manner as in the standard k-ω model. See Section 4.5.1: Modeling the Turbulence Production for details.

Production of ω

The term G_ω represents the production of ω and is given by

$$G_\omega = \frac{\alpha}{\nu_t}\widetilde{G}_k \tag{4.5-45}$$

Note that this formulation differs from the standard k-ω model. The difference between the two models also exists in the way the term α_∞ is evaluated. In the standard k-ω model, α_∞ is defined as a constant (0.52). For the SST k-ω model, α_∞ is given by

$$\alpha_\infty = F_1\alpha_{\infty,1} + (1 - F_1)\alpha_{\infty,2} \tag{4.5-46}$$

where

$$\alpha_{\infty,1} = \frac{\beta_{i,1}}{\beta^*_\infty} - \frac{\kappa^2}{\sigma_{w,1}\sqrt{\beta^*_\infty}} \tag{4.5-47}$$

$$\alpha_{\infty,2} = \frac{\beta_{i,2}}{\beta^*_\infty} - \frac{\kappa^2}{\sigma_{w,2}\sqrt{\beta^*_\infty}} \tag{4.5-48}$$

where κ is 0.41.

Modeling the Turbulence Dissipation

Dissipation of k

The term Y_k represents the dissipation of turbulence kinetic energy, and is defined in a similar manner as in the standard k-ω model (see Section 4.5.1: Modeling the Turbulence Dissipation). The difference is in the way the term f_{β^*} is evaluated. In the standard k-ω model, f_{β^*} is defined as a piecewise function. For the SST k-ω model, f_{β^*} is a constant equal to 1. Thus,

$$Y_k = \rho\beta^* k\omega \tag{4.5-49}$$

Dissipation of ω

The term Y_ω represents the dissipation of ω, and is defined in a similar manner as in the standard k-ω model (see Section 4.5.1: Modeling the Turbulence Dissipation). The difference is in the way the terms β_i and f_β are evaluated. In the standard k-ω model, β_i is defined as a constant (0.072) and f_β is defined in Equation 4.5-24. For the SST k-ω model, f_β is a constant equal to 1. Thus,

$$Y_k = \rho\beta\omega^2 \tag{4.5-50}$$

Instead of having a constant value, β_i is given by

$$\beta_i = F_1\beta_{i,1} + (1 - F_1)\beta_{i,2} \tag{4.5-51}$$

and F_1 is obtained from Equation 4.5-39.

Cross-Diffusion Modification

The SST k-ω model is based on both the standard k-ω model and the standard k-ϵ model. To blend these two models together, the standard k-ϵ model has been transformed into equations based on k and ω, which leads to the introduction of a cross-diffusion term (D_ω in Equation 4.5-33). D_ω is defined as

$$D_\omega = 2\left(1 - F_1\right)\rho\sigma_{\omega,2}\frac{1}{\omega}\frac{\partial k}{\partial x_j}\frac{\partial \omega}{\partial x_j} \qquad (4.5\text{-}52)$$

For details about the various k-ϵ models, see Section 4.4: Standard, RNG, and Realizable k-ϵ Models.

Model Constants

$$\sigma_{k,1} = 1.176, \quad \sigma_{\omega,1} = 2.0, \quad \sigma_{k,2} = 1.0, \quad \sigma_{\omega,2} = 1.168$$

$$a_1 = 0.31, \quad \beta_{i,1} = 0.075 \quad \beta_{i,2} = 0.0828$$

All additional model constants (α^*_∞, α_∞, α_0, β^*_∞, R_β, R_k, R_ω, ζ^*, and M_{t0}) have the same values as for the standard k-ω model (see Section 4.5.1: Model Constants).

4.5.3 Wall Boundary Conditions

The wall boundary conditions for the k equation in the k-ω models are treated in the same way as the k equation is treated when enhanced wall treatments are used with the k-ϵ models. This means that all boundary conditions for wall-function meshes will correspond to the wall function approach, while for the fine meshes, the appropriate low-Reynolds-number boundary conditions will be applied.

In ANSYS FLUENT the value of ω at the wall is specified as

$$\omega_w = \frac{\rho\left(u^*\right)^2}{\mu}\omega^+ \qquad (4.5\text{-}53)$$

The asymptotic value of ω^+ in the laminar sublayer is given by

$$\omega^+ = \min\left(\omega_w^+, \frac{6}{\beta_i (y^+)^2}\right) \tag{4.5-54}$$

where

$$\omega_w^+ = \begin{cases} \left(\frac{50}{k_s^+}\right)^2 & k_s^+ < 25 \\ \frac{100}{k_s^+} & k_s^+ \geq 25 \end{cases} \tag{4.5-55}$$

where

$$k_s^+ = \max\left(1.0, \frac{\rho k_s u^*}{\mu}\right) \tag{4.5-56}$$

and k_s is the roughness height.

In the logarithmic (or turbulent) region, the value of ω^+ is

$$\omega^+ = \frac{1}{\sqrt{\beta_\infty^*}} \frac{du_{\text{turb}}^+}{dy^+} \tag{4.5-57}$$

which leads to the value of ω in the wall cell as

$$\omega = \frac{u^*}{\sqrt{\beta_\infty^*} \kappa y} \tag{4.5-58}$$

Note that in the case of a wall cell being placed in the buffer region, ANSYS FLUENT will blend ω^+ between the logarithmic and laminar sublayer values.

4.6 k-kl-ω Transition Model

This section describes the theory behind the k-kl-ω Transition model. For details about using the model in ANSYS FLUENT, see Chapter 12: Modeling Turbulence and Section 12.8: Setting Up the Transition k-kl-ω Model in the separate User's Guide.

4.6.1 Overview

The k-kl-ω transition model [364] is used to predict boundary layer development and calculate transition onset. This model can be used to effectively address the transition of the boundary layer from a laminar to a turbulent regime.

4.6.2 Transport Equations for the k-kl-ω Model

The k-kl-ω model is considered to be a three-equation eddy-viscosity type, which includes transport equations for turbulent kinetic energy (k_T), laminar kinetic energy (k_L), and the inverse turbulent time scale (ω)

$$\frac{Dk_T}{Dt} = P_{K_T} + R + R_{NAT} - \omega k_T - D_T + \frac{\partial}{\partial x_j}\left[\left(\nu + \frac{\alpha_T}{\alpha_k}\right)\frac{\partial k_T}{\partial x_j}\right] \tag{4.6-1}$$

$$\frac{Dk_L}{Dt} = P_{K_L} - R - R_{NAT} - D_L + \frac{\partial}{\partial x_j}\left[\nu\frac{\partial k_L}{\partial x_j}\right] \tag{4.6-2}$$

$$\frac{D_\omega}{D_t} = C_{\omega 1}\frac{\omega}{k_T}P_{k_T} + \left(\frac{C_{\omega R}}{f_W} - 1\right)\frac{\omega}{k_T}(R + R_{NAT}) - C_{\omega 2}\omega^2 + C_{\omega 3}f_\omega \alpha_T f_W^2 \frac{\sqrt{k_T}}{d^3} + \frac{\partial}{\partial x_j}\left[\left(\nu + \frac{\alpha_T}{\alpha_\omega}\right)\frac{\partial \omega}{\partial x_j}\right] \tag{4.6-3}$$

The inclusion of the turbulent and laminar fluctuations on the mean flow and energy equations via the eddy viscosity and total thermal diffusivity is as follows:

$$\overline{-u_i u_j} = \nu_{TOT}\left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i}\right) - \frac{2}{3}k_{TOT}\delta_{ij} \tag{4.6-4}$$

$$\overline{-u_i \theta} = \alpha_{\theta,TOT}\frac{\partial \theta}{\partial x_i} \tag{4.6-5}$$

The effective length is defined as

$$\lambda_{eff} = MIN(C_\lambda d, \lambda_T) \tag{4.6-6}$$

where λ_T is the turbulent length scale and is defined by

$$\lambda_T = \frac{\sqrt{k}}{\omega} \tag{4.6-7}$$

and the small scale energy is defined by

$$k_{T,s} = f_{ss} f_W k_T \tag{4.6-8}$$

$$f_W = \frac{\lambda_{eff}}{\lambda_T} \tag{4.6-9}$$

$$f_{ss} = exp\left[-\left(\frac{C_{ss}\nu\Omega}{k_T}\right)^2\right] \tag{4.6-10}$$

The large scale energy is given by

$$k_{T,l} = k_T - k_{T,s} \tag{4.6-11}$$

Note that the sum of Equations 4.6-8 and 4.6-11 yields the turbulent kinetic energy k_T. The turbulence production term generated by turbulent fluctuations is given by

$$P_{k_T} = \nu_{T,s} S^2 \tag{4.6-12}$$

where the small-scale turbulent viscosity is $\nu_{T,s}$

$$\nu_{T,s} = f_\nu f_{INT} C_\mu \sqrt{k_{T,s}} \lambda_{eff} \tag{4.6-13}$$

and

$$C_\mu = \frac{1}{A_0 + A_s(S/\omega)} \tag{4.6-14}$$

$$f_\nu = 1 - exp\left(-\frac{\sqrt{Re_{T,s}}}{A_\nu}\right) \tag{4.6-15}$$

A damping function defining the turbulent production due to intermittency is given by

$$f_{INT} = MIN\left(\frac{k_L}{C_{INT} k_{TOT}}, 1\right) \tag{4.6-16}$$

$$Re_{T,s} = \frac{f_W^2 k_T}{\nu\omega} \tag{4.6-17}$$

In Equation 4.6-2, P_{k_L} is the production of laminar kinetic energy by large scale turbulent fluctuations, such that

$$P_{k_L} = \nu_{T,l} S^2 \tag{4.6-18}$$

The large-scale turbulent viscosity $\nu_{T,1}$ is modeled as

$$\nu_{T,1} = MIN\left\{\nu^*_{T,1}, \frac{0.5(k_L + k_{T,1})}{S}\right\} \tag{4.6-19}$$

where

$$\nu^*_{T,1} = f_{\tau,1} C_{11}\left(\frac{\Omega\lambda^2_{eff}}{\nu}\right)\sqrt{k_{T,1}}\lambda_{eff} + \beta_{TS} C_{12}\phi_{NAT} d^2\Omega \tag{4.6-20}$$

The limit in Equation 4.6-19 binds the realizability such that it is not violated in the two-dimensional developing boundary layer. The time-scale-based damping function $f_{\tau,1}$ is

$$f_{\tau,1} = 1 - exp\left[-C_{\tau,1}\frac{k_{T,1}}{\lambda^2_{eff}\Omega^2}\right] \tag{4.6-21}$$

where β_{TS} from Equation 4.6-20 is

$$\beta_{TS} = 1 - exp\left(-\frac{MAX(\phi_{NAT} - C_{TS,crit}, 0)^2}{A_{TS}}\right) \tag{4.6-22}$$

$$\phi_{NAT} = \frac{d^2\Omega}{\nu} \tag{4.6-23}$$

Near-wall dissipation is given by

$$D_T = 2\nu\frac{\partial\sqrt{k_T}}{\partial x_j}\frac{\partial\sqrt{k_T}}{\partial x_j} \tag{4.6-24}$$

$$D_L = 2\nu\frac{\partial\sqrt{k_L}}{\partial x_j}\frac{\partial\sqrt{k_L}}{\partial x_j} \tag{4.6-25}$$

In Equation 4.6-1 – 4.6-3, R represents the averaged effect of the breakdown of streamwise fluctuations into turbulence during bypass transition:

$$R = C_R \beta_{BP} k_L \omega / f_W \tag{4.6-26}$$

β_{BP}, which is the threshold function controls the bypass transition process:

$$\beta_{BP} = 1 - exp\left(-\frac{\phi_{BP}}{A_{BP}}\right) \tag{4.6-27}$$

$$\phi_{BP} = MAX\left[\left(\frac{k_T}{\nu\Omega} - C_{BP,crit}\right), 0\right] \tag{4.6-28}$$

The breakdown to turbulence due to instabilities is considered to be a natural transition production term, given by

$$R_{NAT} = C_{R,NAT} \beta_{NAT} k_L \Omega \tag{4.6-29}$$

$$\beta_{NAT} = 1 - exp\left[-\frac{MAX(\phi_{NAT} - C_{NAT,crit}/f_{NAT,crit}, 0)}{A_{NAT}}\right\} \tag{4.6-30}$$

$$f_{NAT,crit} = 1 - exp\left(C_{NC}\frac{\sqrt{k_L}d}{\nu}\right) \tag{4.6-31}$$

The use of ω as the scale-determining variable can lead to a reduced intermittency effect in the outer region of a turbulent boundary layer, and consequently an elimination of the wake region in the velocity profile. From Equation 4.6-3, the following damping is defined as

$$f_\omega = 1 - exp\left[-0.41\left(\frac{\lambda_{eff}}{\lambda_T}\right)^4\right] \tag{4.6-32}$$

The total eddy viscosity and eddy diffusivity included in Equations 4.6-4 and 4.6-5 are given by

$$\nu_{TOT} = \nu_{T,s} + \nu_{T,l} \tag{4.6-33}$$

$$\alpha_{\theta,TOT} = f_W\left(\frac{k_T}{k_{TOT}}\right)\frac{\nu_{T,s}}{Pr_\theta} + (1 - f_W)C_{\alpha,\theta}\sqrt{k_T}\lambda_{eff} \tag{4.6-34}$$

The turbulent scalar diffusivity in Equations 4.6-1 – 4.6-3 is defined as

$$\alpha_T = f_\nu C_{\mu,std}\sqrt{k_{T,s}}\lambda_{eff} \tag{4.6-35}$$

$$k_{TOT} = k_T + k_L \tag{4.6-36}$$

Model Constants

The model constants for the k-kl-ω transition model are listed below [364]

$$A_0 = 4.04, \quad A_s = 2.12, \quad A_\nu = 6.75, \quad A_{BP} = 0.6$$

$$A_{NAT} = 200, \quad A_{TS} = 200, \quad C_{BP,crit} = 1.2, \quad C_{NC} = 0.1$$

$$C_{NAT,crit} = 1250, \quad C_{INT} = 0.75, \quad C_{TS,crit} = 1000, \quad C_{R,NAT} = 0.02$$

$$C_{11} = 3.4 \times 10^{-6}, \quad C_{12} = 1.0 \times 10^{-10}, \quad C_R = 0.12, \quad C_{\alpha,\theta} = 0.035$$

$$C_{SS} = 1.5, \quad C_{\tau,1} = 4360, \quad C_{\omega 1} = 0.44, \quad C_{\omega 2} = 0.92$$

$$C_{\omega 3} = 0.3, \quad C_{\omega R} = 1.5, \quad C_\lambda = 2.495, \quad C_{\mu,std} = 0.09$$

$$Pr_\theta = 0.85, \quad \sigma_k = 1, \quad \sigma_\omega = 1.17$$

4.7 Transition SST Model

This section describes the theory behind the Transition SST model. Information is presented in the following sections:

- Section 4.7.1: Overview
- Section 4.7.2: Transport Equations for the Transition SST Model
- Section 4.7.3: Specifying Inlet Turbulence Levels

For details about using the model in ANSYS FLUENT, see Chapter 12: Modeling Turbulence and Section 12.9: Setting Up the Transition SST Model in the separate User's Guide.

4.7.1 Overview

The transition SST model is based on the coupling of the SST $k-\omega$ transport equations with two other transport equations, one for the intermittency and one for the transition onset criteria, in terms of momentum-thickness Reynolds number. An ANSYS proprietary empirical correlation (Langtry and Menter) has been developed to cover standard bypass transition as well as flows in low free-stream turbulence environments.

In addition, a very powerful option has been included to allow you to enter your own user-defined empirical correlation, which can then be used to control the transition onset momentum thickness Reynolds number equation. To learn how to set up the transition SST model, see Section 12.9: Setting Up the Transition SST Model (in the separate User's Guide).

4.7.2 Transport Equations for the Transition SST Model

The transport equation for the intermittency γ is defined as:

$$\frac{\partial(\rho\gamma)}{\partial t}+\frac{\partial(\rho U_j\gamma)}{\partial x_j}=P_{\gamma 1}-E_{\gamma 1}+P_{\gamma 2}-E_{\gamma 2}+\frac{\partial}{\partial x_j}\left[\left(\mu+\frac{\mu_t}{\sigma_\gamma}\right)\frac{\partial\gamma}{\partial x_j}\right] \qquad (4.7\text{-}1)$$

The transition sources are defined as follows:

$$\begin{aligned} P_{\gamma 1} &= 2F_{length}\rho S[\gamma F_{onset}]^{c_{\gamma 3}} \\ E_{\gamma 1} &= P_{\gamma 1}\gamma \end{aligned} \qquad (4.7\text{-}2)$$

where S is the strain rate magnitude. F_{length} is an empirical correlation that controls the length of the transition region. The destruction/relaminarization sources are defined as follows:

$$\begin{aligned} P_{\gamma 2} &= (2c_{\gamma 1})\rho\Omega\gamma F_{turb} \\ E_{\gamma 2} &= c_{\gamma 2}P_{\gamma 2}\gamma \end{aligned} \qquad (4.7\text{-}3)$$

where Ω is the vorticity magnitude. The transition onset is controlled by the following functions:

$$\begin{aligned} Re_V &= \frac{\rho y^2 S}{\mu} \\ R_T &= \frac{\rho k}{\mu\omega} \end{aligned} \qquad (4.7\text{-}4)$$

$$\begin{aligned}F_{onset1} &= \frac{Re_v}{2.193 Re_{\theta c}} \\ F_{onset2} &= min(max(F_{onset1}, F_{onset1}^4), 2.0)\end{aligned} \tag{4.7-5}$$

$$\begin{aligned}F_{onset3} &= max\left(1 - \left(\frac{R_T}{2.5}\right)^3, 0\right) \\ F_{onset} &= max(F_{onset2} - F_{onset3}, 0) \\ F_{turb} &= e^{-\left(\frac{R_T}{4}\right)^4}\end{aligned} \tag{4.7-6}$$

$Re_{\theta c}$ is the critical Reynolds number where the intermittency first starts to increase in the boundary layer. This occurs upstream of the transition Reynolds number $\widetilde{Re}_{\theta t}$ and the difference between the two must be obtained from an empirical correlation. Both the F_{length} and $Re_{\theta c}$ correlations are functions of $\widetilde{Re}_{\theta t}$.

The constants for the intermittency equation are:

$$c_{\gamma 1} = 0.03; \quad c_{\gamma 2} = 50; \quad c_{\gamma 3} = 0.5; \quad \sigma_\gamma = 1.0$$

Separation Induced Transition Correction

The modification for separation-induced transition is:

$$\begin{aligned}\gamma_{sep} &= min\left(2max\left[\left(\frac{Re_v}{3.235 Re_{\theta c}}\right) - 1, 0\right] F_{reattch}, 2\right) F_{\theta t} \\ F_{reattch} &= e^{-\left(\frac{R_T}{20}\right)^4} \\ \gamma_{eff} &= max(\gamma, \gamma_{sep})\end{aligned} \tag{4.7-7}$$

The model constants in Equation 4.7-7 have been adjusted from those of Menter et al. [226] in order to improve the predictions of separated flow transition. The main difference is that the constant that controls the relation between Re_v and $Re_{\theta c}$ was changed from 2.193, its value for a Blasius boundary layer, to 3.235, the value at a separation point where the shape factor is 3.5 [226]. The boundary condition for γ at a wall is zero normal flux, while for an inlet, γ is equal to 1.0.

The transport equation for the transition momentum thickness Reynolds number $\widetilde{Re}_{\theta t}$ is

$$\frac{\partial(\rho \widetilde{Re}_{\theta t})}{\partial t} + \frac{\partial(\rho U_j \widetilde{Re}_{\theta t})}{\partial x_j} = P_{\theta t} + \frac{\partial}{\partial x_j}\left[\sigma_{\theta t}(\mu + \mu_t)\frac{\partial \widetilde{Re}_{\theta t}}{\partial x_j}\right] \tag{4.7-8}$$

The source term is defined as follows:

$$\begin{aligned} P_{\theta t} &= c_{\theta t}\frac{\rho}{t}(Re_{\theta t} - \widetilde{Re}_{\theta t})(1.0 - F_{\theta t}) \\ t &= \frac{500\mu}{\rho U^2} \end{aligned} \tag{4.7-9}$$

$$F_{\theta t} = min\left(max\left(F_{wake}e^{\left(-\frac{y}{\delta}\right)^4}, 1.0 - \left(\frac{\gamma - 1/50}{1.0 - 1/50}\right)^2\right), 1.0\right) \tag{4.7-10}$$

$$\begin{aligned} \theta_{BL} &= \frac{\widetilde{Re}_{\theta t}\mu}{\rho U} \\ \delta_{BL} &= \frac{15}{2}\theta_{BL} \\ \delta &= \frac{50\Omega y}{U}\delta_{BL} \end{aligned} \tag{4.7-11}$$

$$\begin{aligned} Re_\omega &= \frac{\rho\omega y^2}{\mu} \\ F_{wake} &= e^{-\left(\frac{Re_\omega}{1E+5}\right)^2} \end{aligned} \tag{4.7-12}$$

The model constants for the $\widetilde{Re}_{\theta t}$ equation are:

$$c_{\theta t} = 0.03 \quad \sigma_{\theta t} = 2.0$$

The boundary condition for $\widetilde{Re}_{\theta t}$ at a wall is zero flux. The boundary condition for $\widetilde{Re}_{\theta t}$ at an inlet should be calculated from the empirical correlation based on the inlet turbulence intensity.

The model contains three empirical correlations. $Re_{\Theta t}$ is the transition onset as observed in experiments. This has been modified from Menter et al. [226] in order to improve the predictions for natural transition. It is used in Equation 4.7-9. F_{length} is the length of the transition zone and is substituted in Equation 4.7-2. $Re_{\Theta c}$ is the point where the model is activated in order to match both $Re_{\Theta t}$ and F_{length}, and is used in Equation 4.7-5. At present, these empirical correlations are proprietary and are not given in this manual.

$$\begin{aligned} Re_{\Theta t} &= f(Tu, \lambda) \\ F_{length} &= f(\widetilde{Re}_{\Theta t}) \\ Re_{\Theta c} &= f(\widetilde{Re}_{\Theta t}) \end{aligned} \tag{4.7-13}$$

The first empirical correlation is a function of the local turbulence intensity, Tu, and the Thwaites' pressure gradient coefficient λ_θ is defined as

$$\lambda_\theta = (\theta^2/v)dU/ds \tag{4.7-14}$$

where dU/ds is the acceleration in the streamwise direction.

Coupling the Transition Model and SST Transport Equations

The transition model interacts with the SST turbulence model, as follows:

$$\frac{\partial}{\partial t}(\rho k) + \frac{\partial}{\partial x_j}(\rho u_j k) = \widetilde{P_k} - \widetilde{D_k} + \frac{\partial}{\partial x_j}\left((\mu + \sigma_k \mu_t)\frac{\partial k}{\partial x_j}\right) \tag{4.7-15}$$

$$\widetilde{P_k} = \gamma_{eff} P_k \tag{4.7-16}$$

$$\widetilde{D_k} = min(max(\gamma_{eff}, 0.1), 1.0)D_k \tag{4.7-17}$$

$$R_y = \frac{\rho y \sqrt{k}}{\mu} \tag{4.7-18}$$

$$F_3 = e^{-}\left(\frac{R_y}{120}\right)^3 \tag{4.7-19}$$

$$F_t = max(F_{1orig}, F_3) \tag{4.7-20}$$

where P_k and D_k are the original production and destruction terms for the SST model and F_{1orig} is the original SST blending function. Note that the production term in the ω-equation is not modified. The rationale behind the above model formulation is given in detail in Menter et al. [226].

In order to capture the laminar and transitional boundary layers correctly, the mesh must have a y^+ of approximately one. If the y^+ is too large (i.e. > 5), then the transition onset location moves upstream with increasing y^+. It is recommended to use the bounded second order upwind based discretization for the mean flow, turbulence and transition equations.

4.7.3 Specifying Inlet Turbulence Levels

It has been observed that the turbulence intensity specified at an inlet can decay quite rapidly depending on the inlet viscosity ratio (μ_t/μ) (and hence turbulence eddy frequency). As a result, the local turbulence intensity downstream of the inlet can be much smaller than the inlet value (see Figure 4.7.1). Typically, the larger the inlet viscosity ratio, the smaller the turbulent decay rate. However, if too large a viscosity ratio is specified (i.e., >100), the skin friction can deviate significantly from the laminar value. There is experimental evidence that suggests that this effect occurs physically; however, at this point it is not clear how accurately the transition model reproduces this behavior. For this reason, if possible, it is desirable to have a relatively low (i.e $\approx 1 - 10$) inlet viscosity ratio and to estimate the inlet value of turbulence intensity such that at the leading edge of the blade/airfoil, the turbulence intensity has decayed to the desired value. The decay of turbulent kinetic energy can be calculated with the following analytical solution:

$$k = k_{inlet}(1 + \omega_{inlet}\beta t)^{\frac{-\beta^*}{\beta}} \tag{4.7-21}$$

For the SST turbulence model in the freestream the constants are:

$$\beta = 0.09, \quad \beta^* = 0.0828$$

The time scale can be determined as follows:

$$t = \frac{x}{V} \tag{4.7-22}$$

where x is the streamwise distance downstream of the inlet and V is the mean convective velocity. The eddy viscosity is defined as:

$$\mu_t = \frac{\rho k}{\omega} \tag{4.7-23}$$

The decay of turbulent kinetic energy equation can be rewritten in terms of inlet turbulence intensity ($T_{u_{inlet}}$) and eddy viscosity ratio (μ_t/μ) as follows:

$$T_u = \left(T_{u_{inlet}}^2 \left[1 + \frac{3\rho V x \beta T_{u_{inlet}}^2}{2\mu(\mu_t/\mu)} \right]^{\frac{-\beta^*}{\beta}} \right)^{0.5} \tag{4.7-24}$$

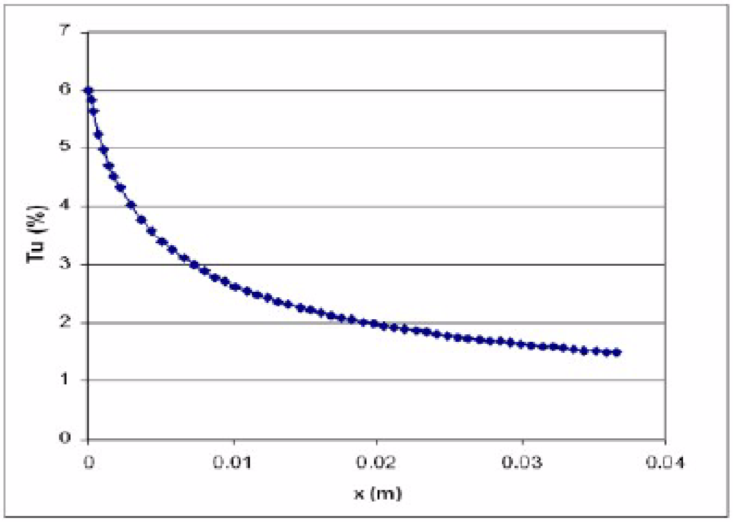


Figure 4.7.1: Decay of Turbulence Intensity (T_u) as a Function of Streamwise Distance (x)

4.8 The v^2-f Model

The v^2-f model is similar to the standard k-ϵ model, but incorporates near-wall turbulence anisotropy and non-local pressure-strain effects. A limitation of the v^2-f model is that it cannot be used to solve Eulerian multiphase problems, whereas the k-ϵ model is typically used in such applications. The v^2-f model is a general low-Reynolds-number turbulence model that is valid all the way up to solid walls, and therefore does not need to make use of wall functions. Although the model was originally developed for attached or mildly separated boundary layers [82], it also accurately simulates flows dominated by separation [24].

The distinguishing feature of the v^2-f model is its use of the velocity scale, $\overline{v^2}$, instead of the turbulent kinetic energy, k, for evaluating the eddy viscosity. $\overline{v^2}$, which can be thought of as the velocity fluctuation normal to the streamlines, has shown to provide the right scaling in representing the damping of turbulent transport close to the wall, a feature that k does not provide.

For more information about the theoretical background and usage of the v^2-f model, please visit the User Services Center (www.fluentusers.com) .

4.9 Reynolds Stress Model (RSM)

This section describes the theory behind the Reynolds Stress model (RSM). Information is presented in the following sections:

- Section 4.9.1: Overview
- Section 4.9.2: Reynolds Stress Transport Equations
- Section 4.9.3: Modeling Turbulent Diffusive Transport
- Section 4.9.4: Modeling the Pressure-Strain Term
- Section 4.9.5: Effects of Buoyancy on Turbulence
- Section 4.9.6: Modeling the Turbulence Kinetic Energy
- Section 4.9.7: Modeling the Dissipation Rate
- Section 4.9.8: Modeling the Turbulent Viscosity
- Section 4.9.9: Wall Boundary Conditions
- Section 4.9.10: Convective Heat and Mass Transfer Modeling

For details about using the model in ANSYS FLUENT, see Chapter 12: Modeling Turbulence and Section 12.10: Setting Up the Reynolds Stress Model in the separate User's Guide.

4.9.1 Overview

The Reynolds stress model (RSM) [108, 177, 178] is the most elaborate type of turbulence model that ANSYS FLUENT provides. Abandoning the isotropic eddy-viscosity hypothesis, the RSM closes the Reynolds-averaged Navier-Stokes equations by solving transport equations for the Reynolds stresses, together with an equation for the dissipation rate. This means that five additional transport equations are required in 2D flows, in comparison to seven additional transport equations solved in 3D.

Since the RSM accounts for the effects of streamline curvature, swirl, rotation, and rapid changes in strain rate in a more rigorous manner than one-equation and two-equation models, it has greater potential to give accurate predictions for complex flows. However, the fidelity of RSM predictions is still limited by the closure assumptions employed to model various terms in the exact transport equations for the Reynolds stresses. The modeling of the pressure-strain and dissipation-rate terms is particularly challenging, and often considered to be responsible for compromising the accuracy of RSM predictions.

The RSM might not always yield results that are clearly superior to the simpler models in all classes of flows to warrant the additional computational expense. However, use of the RSM is a must when the flow features of interest are the result of anisotropy in the Reynolds stresses. Among the examples are cyclone flows, highly swirling flows in combustors, rotating flow passages, and the stress-induced secondary flows in ducts.

The exact form of the Reynolds stress transport equations may be derived by taking moments of the exact momentum equation. This is a process wherein the exact momentum equations are multiplied by a fluctuating property, the product then being Reynolds-averaged. Unfortunately, several of the terms in the exact equation are unknown and modeling assumptions are required in order to close the equations.

4.9.2 Reynolds Stress Transport Equations

The exact transport equations for the transport of the Reynolds stresses, $\rho\overline{u_i' u_j'}$, may be written as follows:

$$\underbrace{\frac{\partial}{\partial t}(\rho\,\overline{u_i' u_j'})}_{\text{Local Time Derivative}} + \underbrace{\frac{\partial}{\partial x_k}(\rho u_k \overline{u_i' u_j'})}_{C_{ij} \equiv \text{Convection}} = \underbrace{-\frac{\partial}{\partial x_k}\left[\rho\,\overline{u_i' u_j' u_k'} + \overline{p\left(\delta_{kj} u_i' + \delta_{ik} u_j'\right)}\right]}_{D_{T,ij} \equiv \text{Turbulent Diffusion}}$$

$$+ \underbrace{\frac{\partial}{\partial x_k}\left[\mu \frac{\partial}{\partial x_k}(\overline{u_i' u_j'})\right]}_{D_{L,ij} \equiv \text{Molecular Diffusion}} \underbrace{-\rho\left(\overline{u_i' u_k'}\frac{\partial u_j}{\partial x_k} + \overline{u_j' u_k'}\frac{\partial u_i}{\partial x_k}\right)}_{P_{ij} \equiv \text{Stress Production}} \underbrace{-\rho\beta(g_i\overline{u_j'\theta} + g_j\overline{u_i'\theta})}_{G_{ij} \equiv \text{Buoyancy Production}}$$

$$+ \underbrace{\overline{p\left(\frac{\partial u_i'}{\partial x_j} + \frac{\partial u_j'}{\partial x_i}\right)}}_{\phi_{ij} \equiv \text{Pressure Strain}} - \underbrace{2\mu\overline{\frac{\partial u_i'}{\partial x_k}\frac{\partial u_j'}{\partial x_k}}}_{\epsilon_{ij} \equiv \text{Dissipation}}$$

$$\underbrace{-2\rho\Omega_k\left(\overline{u_j' u_m'}\epsilon_{ikm} + \overline{u_i' u_m'}\epsilon_{jkm}\right)}_{F_{ij} \equiv \text{Production by System Rotation}} + \underbrace{S_{\text{user}}}_{\text{User-Defined Source Term}} \tag{4.9-1}$$

Of the various terms in these exact equations, C_{ij}, $D_{L,ij}$, P_{ij}, and F_{ij} do not require any modeling. However, $D_{T,ij}$, G_{ij}, ϕ_{ij}, and ϵ_{ij} need to be modeled to close the equations. The following sections describe the modeling assumptions required to close the equation set.

4.9.3 Modeling Turbulent Diffusive Transport

$D_{T,ij}$ can be modeled by the generalized gradient-diffusion model of Daly and Harlow [67]:

$$D_{T,ij} = C_s \frac{\partial}{\partial x_k}\left(\rho \frac{k\overline{u'_k u'_\ell}}{\epsilon}\frac{\partial \overline{u'_i u'_j}}{\partial x_\ell}\right) \tag{4.9-2}$$

However, this equation can result in numerical instabilities, so it has been simplified in ANSYS FLUENT to use a scalar turbulent diffusivity as follows [194]:

$$D_{T,ij} = \frac{\partial}{\partial x_k}\left(\frac{\mu_t}{\sigma_k}\frac{\partial \overline{u'_i u'_j}}{\partial x_k}\right) \tag{4.9-3}$$

The turbulent viscosity, μ_t, is computed using Equation 4.9-33.

Lien and Leschziner [194] derived a value of $\sigma_k = 0.82$ by applying the generalized gradient-diffusion model, Equation 4.9-2, to the case of a planar homogeneous shear flow. Note that this value of σ_k is different from that in the standard and realizable k-ϵ models, in which $\sigma_k = 1.0$.

4.9.4 Modeling the Pressure-Strain Term

Linear Pressure-Strain Model

By default in ANSYS FLUENT, the pressure-strain term, ϕ_{ij}, in Equation 4.9-1 is modeled according to the proposals by Gibson and Launder [108], Fu et al. [104], and Launder [176, 177].

The classical approach to modeling ϕ_{ij} uses the following decomposition:

$$\phi_{ij} = \phi_{ij,1} + \phi_{ij,2} + \phi_{ij,w} \tag{4.9-4}$$

where $\phi_{ij,1}$ is the slow pressure-strain term, also known as the return-to-isotropy term, $\phi_{ij,2}$ is called the rapid pressure-strain term, and $\phi_{ij,w}$ is the wall-reflection term.

The slow pressure-strain term, $\phi_{ij,1}$, is modeled as

$$\phi_{ij,1} \equiv -C_1 \rho \frac{\epsilon}{k}\left[\overline{u'_i u'_j} - \frac{2}{3}\delta_{ij} k\right] \tag{4.9-5}$$

with $C_1 = 1.8$.

The rapid pressure-strain term, $\phi_{ij,2}$, is modeled as

$$\phi_{ij,2} \equiv -C_2 \left[(P_{ij} + F_{ij} + 5/6 G_{ij} - C_{ij}) - \frac{2}{3}\delta_{ij}(P + 5/6G - C) \right] \tag{4.9-6}$$

where $C_2 = 0.60$, P_{ij}, F_{ij}, G_{ij}, and C_{ij} are defined as in Equation 4.9-1, $P = \frac{1}{2}P_{kk}$, $G = \frac{1}{2}G_{kk}$, and $C = \frac{1}{2}C_{kk}$.

The wall-reflection term, $\phi_{ij,w}$, is responsible for the redistribution of normal stresses near the wall. It tends to damp the normal stress perpendicular to the wall, while enhancing the stresses parallel to the wall. This term is modeled as

$$\begin{aligned} \phi_{ij,w} \equiv \;& C_1' \frac{\epsilon}{k}\left(\overline{u_k' u_m'} n_k n_m \delta_{ij} - \frac{3}{2}\overline{u_i' u_k'} n_j n_k - \frac{3}{2}\overline{u_j' u_k'} n_i n_k\right) \frac{C_\ell k^{3/2}}{\epsilon d} \\ & + C_2' \left(\phi_{km,2} n_k n_m \delta_{ij} - \frac{3}{2}\phi_{ik,2} n_j n_k - \frac{3}{2}\phi_{jk,2} n_i n_k\right) \frac{C_\ell k^{3/2}}{\epsilon d} \end{aligned} \tag{4.9-7}$$

where $C_1' = 0.5$, $C_2' = 0.3$, n_k is the x_k component of the unit normal to the wall, d is the normal distance to the wall, and $C_\ell = C_\mu^{3/4}/\kappa$, where $C_\mu = 0.09$ and κ is the von Kármán constant ($= 0.4187$).

$\phi_{ij,w}$ is included by default in the Reynolds stress model.

Low-Re Modifications to the Linear Pressure-Strain Model

When the RSM is applied to near-wall flows using the enhanced wall treatment described in Section 4.12.4: Two-Layer Model for Enhanced Wall Treatment, the pressure-strain model needs to be modified. The modification used in ANSYS FLUENT specifies the values of C_1, C_2, C_1', and C_2' as functions of the Reynolds stress invariants and the turbulent Reynolds number, according to the suggestion of Launder and Shima [179]:

$$C_1 = 1 + 2.58 A {A_2}^{0.25} \left\{1 - \exp\left[-(0.0067 \mathrm{Re}_t)^2\right]\right\} \tag{4.9-8}$$

$$C_2 = 0.75\sqrt{A} \tag{4.9-9}$$

$$C_1' = -\frac{2}{3}C_1 + 1.67 \tag{4.9-10}$$

$$C_2' = \max\left[\frac{\frac{2}{3}C_2 - \frac{1}{6}}{C_2}, 0\right] \tag{4.9-11}$$

with the turbulent Reynolds number defined as $\mathrm{Re}_t = (\rho k^2/\mu\epsilon)$. The flatness parameter A and tensor invariants, A_2 and A_3, are defined as

$$
\begin{aligned}
A &\equiv \left[1-\frac{9}{8}\left(A_2-A_3\right)\right] &\quad (4.9\text{-}12)\\
A_2 &\equiv a_{ik}a_{ki} &\quad (4.9\text{-}13)\\
A_3 &\equiv a_{ik}a_{kj}a_{ji} &\quad (4.9\text{-}14)
\end{aligned}
$$

a_{ij} is the Reynolds-stress anisotropy tensor, defined as

$$a_{ij} = -\left(\frac{-\rho\overline{u_i'u_j'} + \frac{2}{3}\rho k \delta_{ij}}{\rho k}\right) \tag{4.9-15}$$

The modifications detailed above are employed only when the enhanced wall treatment is selected in the **Viscous Model** dialog box.

Quadratic Pressure-Strain Model

An optional pressure-strain model proposed by Speziale, Sarkar, and Gatski [334] is provided in ANSYS FLUENT. This model has been demonstrated to give superior performance in a range of basic shear flows, including plane strain, rotating plane shear, and axisymmetric expansion/contraction. This improved accuracy should be beneficial for a wider class of complex engineering flows, particularly those with streamline curvature. The quadratic pressure-strain model can be selected as an option in the **Viscous Model** dialog box.

This model is written as follows:

$$
\begin{aligned}
\phi_{ij} = &-\left(C_1\rho\epsilon + C_1^* P\right)b_{ij} + C_2\rho\epsilon\left(b_{ik}b_{kj} - \frac{1}{3}b_{mn}b_{mn}\delta_{ij}\right) + \left(C_3 - C_3^*\sqrt{b_{ij}b_{ij}}\right)\rho k S_{ij}\\
&+ C_4\rho k\left(b_{ik}S_{jk} + b_{jk}S_{ik} - \frac{2}{3}b_{mn}S_{mn}\delta_{ij}\right) + C_5\rho k\left(b_{ik}\Omega_{jk} + b_{jk}\Omega_{ik}\right) \qquad (4.9\text{-}16)
\end{aligned}
$$

where b_{ij} is the Reynolds-stress anisotropy tensor defined as

$$b_{ij} = -\left(\frac{-\rho\overline{u_i'u_j'} + \frac{2}{3}\rho k \delta_{ij}}{2\rho k}\right) \tag{4.9-17}$$

The mean strain rate, S_{ij}, is defined as

$$S_{ij} = \frac{1}{2}\left(\frac{\partial u_j}{\partial x_i} + \frac{\partial u_i}{\partial x_j}\right) \tag{4.9-18}$$

The mean rate-of-rotation tensor, Ω_{ij}, is defined by

$$\Omega_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i}\right) \tag{4.9-19}$$

The constants are

$$C_1 = 3.4,\ C_1^* = 1.8,\ C_2 = 4.2,\ C_3 = 0.8,\ C_3^* = 1.3,\ C_4 = 1.25,\ C_5 = 0.4$$

The quadratic pressure-strain model does not require a correction to account for the wall-reflection effect in order to obtain a satisfactory solution in the logarithmic region of a turbulent boundary layer. It should be noted, however, that the quadratic pressure-strain model is not available when the enhanced wall treatment is selected in the **Viscous Model** dialog box.

Low-Re Stress-Omega Model

The low-Re stress-omega model is a stress-transport model that is based on the omega equations and LRR model [379]. This model is ideal for modeling flows over curved surfaces and swirling flows. The low-Re stress-omega model can be selected in the **Viscous Model** dialog box and requires no treatments of wall reflections. The closure coefficients are identical to the k-ω model (Section 4.5.1: Model Constants), however, there are additional closure coefficients, C_1 and C_2, noted below.

The low-Re stress-omega model resembles the k-ω model due to its excellent predictions for a wide range of turbulent flows. Furthermore, low Reynolds number modifications and surface boundary conditions for rough surfaces are similar to the k-ω model.

Equation 4.9-4 can be re-written for the low-Re stress-omega model such that wall reflections are excluded:

$$\phi_{ij} = \phi_{ij,1} + \phi_{ij,2} \tag{4.9-20}$$

Therefore,

$$\phi_{ij} = -C_1 \rho \beta^*_{RSM} \omega \left[\overline{u_i' u_j'} - 2/3\delta_{ij} k\right] - \hat{\alpha}_0 \left[P_{ij} - 1/3 P_{kk}\delta_{ij}\right]$$

$$- \hat{\beta}_0 \left[D_{ij} - 1/3 P_{kk}\delta_{ij}\right] - k\hat{\gamma}_0 \left[S_{ij} - 1/3 S_{kk}\delta_{ij}\right] \tag{4.9-21}$$

where D_{ij} is defined as

$$D_{ij} = -\rho \left[\overline{u_i{}' u_m{}'} \frac{\partial u_m}{\partial x_j} + \overline{u_j{}' u_m{}'} \frac{\partial u_m}{\partial x_i} \right] \tag{4.9-22}$$

The mean strain rate S_{ij} is defined in Equation 4.9-18 and β^*_{RSM} is defined by

$$\beta^*_{RSM} = \beta^* f_{\beta^*} \tag{4.9-23}$$

where β^* and f^*_β are defined in the same way as for the standard $k - \omega$, using Equations 4.5-16 and 4.5-22, respectively. The only difference here is that the equation for f^*_β uses a value of 640 instead of 680, as in Equation 4.5-16.

The constants are

$$\hat{\alpha_0} = \frac{8 + C_2}{11}, \quad \hat{\beta_0} = \frac{8C_2 - 2}{11}, \quad \hat{\gamma_0} = \frac{60C_2 - 4}{55}$$

$$C_1 = 1.8, \quad C_2 = 0.52$$

The above formulation does not require viscous damping functions to resolve the near-wall sublayer. However, inclusion of the viscous damping function [379] could improve model predictions for certain flows. This results in the following changes:

$$\hat{\alpha} = \frac{1 + \hat{\alpha_0} Re_T / R_k}{1 + Re_T / R_k}$$

$$\hat{\beta} = \hat{\beta_0} \frac{Re_T / R_k}{1 + Re_T / R_k}$$

$$\hat{\gamma} = \hat{\gamma_0} \frac{0.007 + Re_T / R_k}{1 + Re_T / R_k}$$

$$C_1 = 1.8 \left[\frac{5/3 + Re_T / R_k}{1 + Re_T / R_k} \right]$$

where $\hat{\alpha}$, $\hat{\beta}$, and $\hat{\gamma}$ would replace $\hat{\alpha_0}$, $\hat{\beta_0}$, and $\hat{\gamma_0}$ in Equation 4.9-21. The constants are

$$R_\beta = 12, \quad R_k = 12, \quad R_\omega = 6.20$$

Inclusion of the low-Re viscous damping is controlled by enabling Low-Re Corrections under k-omega Options in the **Viscous Model** dialog box.

4.9.5 Effects of Buoyancy on Turbulence

The production terms due to buoyancy are modeled as

$$G_{ij} = (\overline{J_i U_j} + \overline{J_j U_i}) = -\beta(g_i \overline{u_j \theta} + g_j \overline{U_i \theta}) \tag{4.9-24}$$

$$\overline{U_i \theta} = \frac{\mu_t}{\mathrm{Pr_t}}(\frac{\partial T}{\partial X_i}) \tag{4.9-25}$$

where Pr_t is the turbulent Prandtl number for energy, with a default value of 0.85.

Using the definition of the coefficient of thermal expansion, β, given by Equation 4.4-24, the following expression is obtained for G_{ij} for ideal gases:

$$G_{ij} = -\frac{\mu_t}{\rho \mathrm{Pr}_t}\left(g_i \frac{\partial \rho}{\partial x_j} + g_j \frac{\partial \rho}{\partial x_i}\right) \tag{4.9-26}$$

Note that the buoyancy effects are not included if low-Re omega based RSM model is used.

4.9.6 Modeling the Turbulence Kinetic Energy

In general, when the turbulence kinetic energy is needed for modeling a specific term, it is obtained by taking the trace of the Reynolds stress tensor:

$$k = \frac{1}{2}\overline{u_i' u_i'} \tag{4.9-27}$$

As described in Section 4.9.9: Wall Boundary Conditions, an option is available in ANSYS FLUENT to solve a transport equation for the turbulence kinetic energy in order to obtain boundary conditions for the Reynolds stresses. In this case, the following model equation is used:

$$\frac{\partial}{\partial t}(\rho k) + \frac{\partial}{\partial x_i}(\rho k u_i) = \frac{\partial}{\partial x_j}\left[\left(\mu + \frac{\mu_t}{\sigma_k}\right)\frac{\partial k}{\partial x_j}\right] + \frac{1}{2}\left(P_{ii} + G_{ii}\right) - \rho\epsilon(1 + 2\mathrm{M}_t^2) + S_k \tag{4.9-28}$$

where $\sigma_k = 0.82$ and S_k is a user-defined source term. Equation 4.9-28 is obtainable by contracting the modeled equation for the Reynolds stresses (Equation 4.9-1). As one might expect, it is essentially identical to Equation 4.4-1 used in the standard k-ϵ model.

Although Equation 4.9-28 is solved globally throughout the flow domain, the values of k obtained are used only for boundary conditions. In every other case, k is obtained from Equation 4.9-27. This is a minor point, however, since the values of k obtained with either method should be very similar.

4.9.7 Modeling the Dissipation Rate

The dissipation tensor, ϵ_{ij}, is modeled as

$$\epsilon_{ij} = \frac{2}{3}\delta_{ij}(\rho\epsilon + Y_M) \tag{4.9-29}$$

where $Y_M = 2\rho\epsilon \mathrm{M}_t^2$ is an additional "dilatation dissipation" term according to the model by Sarkar [300]. The turbulent Mach number in this term is defined as

$$\mathrm{M}_t = \sqrt{\frac{k}{a^2}} \tag{4.9-30}$$

where a ($\equiv \sqrt{\gamma RT}$) is the speed of sound. This compressibility modification always takes effect when the compressible form of the ideal gas law is used.

The scalar dissipation rate, ϵ, is computed with a model transport equation similar to that used in the standard k-ϵ model:

$$\frac{\partial}{\partial t}(\rho\epsilon) + \frac{\partial}{\partial x_i}(\rho\epsilon u_i) = \frac{\partial}{\partial x_j}\left[\left(\mu + \frac{\mu_t}{\sigma_\epsilon}\right)\frac{\partial\epsilon}{\partial x_j}\right] C_{\epsilon 1}\frac{1}{2}\left[P_{ii} + C_{\epsilon 3}G_{ii}\right]\frac{\epsilon}{k} - C_{\epsilon 2}\rho\frac{\epsilon^2}{k} + S_\epsilon \tag{4.9-31}$$

where $\sigma_\epsilon = 1.0$, $C_{\epsilon 1} = 1.44$, $C_{\epsilon 2} = 1.92$, $C_{\epsilon 3}$ is evaluated as a function of the local flow direction relative to the gravitational vector, as described in Section 4.4.5: Effects of Buoyancy on Turbulence in the k-ϵ Models, and S_ϵ is a user-defined source term.

In the case when the Reynolds Stress model is coupled with the omega equation, the dissipation tensor ϵ_{ij} is modeled as

$$\epsilon_{ij} = 2/3\delta_{ij}\rho\beta^*_{RSM}k\omega \tag{4.9-32}$$

Where β^*_{RSM} is defined in Section 4.9.4: Modeling the Pressure-Strain Term and the specific dissipation rate ω is computed in the same way as for the standard $k - \omega$ model, using Equation 4.5-2.

4.9.8 Modeling the Turbulent Viscosity

The turbulent viscosity, μ_t, is computed similarly to the k-ϵ models:

$$\mu_t = \rho C_\mu \frac{k^2}{\epsilon} \tag{4.9-33}$$

where $C_\mu = 0.09$.

4.9.9 Wall Boundary Conditions

The RSM model in ANSYS FLUENT requires boundary conditions for individual Reynolds stresses, $\overline{u_i' u_j'}$, and for the turbulence dissipation rate, ϵ (or ω if the low-Re stress-omega model is used). These quantities can be input directly or derived from the turbulence intensity and characteristic length (Section 12.14.3: Reynolds Stress Model in the separate User's Guide).

At walls, ANSYS FLUENT computes the near-wall values of the Reynolds stresses and ϵ from wall functions (see Section 4.12.2: Standard Wall Functions, Section 4.12.3: Non-Equilibrium Wall Functions, and Section 4.12.4: Enhanced Wall Functions). ANSYS FLUENT applies explicit wall boundary conditions for the Reynolds stresses by using the log-law and the assumption of equilibrium, disregarding convection and diffusion in the transport equations for the stresses (Equation 4.9-1). Using a local coordinate system, where τ is the tangential coordinate, η is the normal coordinate, and λ is the binormal coordinate, the Reynolds stresses at the wall-adjacent cells (assuming standard wall functions or non-equilibrium wall functions) are computed from

$$\frac{\overline{u_\tau'^2}}{k} = 1.098, \quad \frac{\overline{u_\eta'^2}}{k} = 0.247, \quad \frac{\overline{u_\lambda'^2}}{k} = 0.655, \quad -\frac{\overline{u_\tau' u_\eta'}}{k} = 0.255 \tag{4.9-34}$$

To obtain k, ANSYS FLUENT solves the transport equation of Equation 4.9-28. For reasons of computational convenience, the equation is solved globally, even though the values of k thus computed are needed only near the wall; in the far field k is obtained directly from the normal Reynolds stresses using Equation 4.9-27. By default, the values of the Reynolds stresses near the wall are fixed using the values computed from Equation 4.9-34, and the transport equations in Equation 4.9-1 are solved only in the bulk flow region.

Alternatively, the Reynolds stresses can be explicitly specified in terms of wall-shear stress, instead of k:

$$\frac{\overline{u_\tau'^2}}{u_\tau^2} = 5.1, \quad \frac{\overline{u_\eta'^2}}{u_\tau^2} = 1.0, \quad \frac{\overline{u_\lambda'^2}}{u_\tau^2} = 2.3, \quad -\frac{\overline{u_\tau' u_\eta'}}{u_\tau^2} = 1.0 \tag{4.9-35}$$

where u_τ is the friction velocity defined by $u_\tau \equiv \sqrt{\tau_w/\rho}$, where τ_w is the wall-shear stress. When this option is chosen, the k transport equation is not solved.

When using enhanced wall treatments as the near-wall treatment, ANSYS FLUENT applies zero flux wall boundary conditions to the Reynolds stress equations.

4.9.10 Convective Heat and Mass Transfer Modeling

With the Reynolds stress model in ANSYS FLUENT, turbulent heat transport is modeled using the concept of Reynolds' analogy to turbulent momentum transfer. The "modeled" energy equation is thus given by the following:

$$\frac{\partial}{\partial t}(\rho E) + \frac{\partial}{\partial x_i}[u_i(\rho E + p)] = \frac{\partial}{\partial x_j}\left[\left(k + \frac{c_p \mu_t}{\mathrm{Pr}_t}\right)\frac{\partial T}{\partial x_j} + u_i(\tau_{ij})_{\mathrm{eff}}\right] + S_h \tag{4.9-36}$$

where E is the total energy and $(\tau_{ij})_{\mathrm{eff}}$ is the deviatoric stress tensor, defined as

$$(\tau_{ij})_{\mathrm{eff}} = \mu_{\mathrm{eff}}\left(\frac{\partial u_j}{\partial x_i} + \frac{\partial u_i}{\partial x_j}\right) - \frac{2}{3}\mu_{\mathrm{eff}}\frac{\partial u_k}{\partial x_k}\delta_{ij}$$

The term involving $(\tau_{ij})_{\mathrm{eff}}$ represents the viscous heating, and is always computed in the density-based solvers. It is not computed by default in the pressure-based solver, but it can be enabled in the **Viscous Model** dialog box. The default value of the turbulent Prandtl number is 0.85. You can change the value of Pr_t in the **Viscous Model** dialog box.

Turbulent mass transfer is treated similarly, with a default turbulent Schmidt number of 0.7. This default value can be changed in the **Viscous Model** dialog box.

4.10 Detached Eddy Simulation (DES)

This section describes the theory behind the Detached Eddy Simulation (DES) model. Information is presented in the following sections:

- Section 4.10.1: Spalart-Allmaras Based DES Model
- Section 4.10.2: Realizable k-ϵ Based DES Model
- Section 4.10.3: SST k-ω Based DES Model

For details about using the model in ANSYS FLUENT, see Chapter 12: Modeling Turbulence and Section 12.11: Setting Up the Detached Eddy Simulation Model in the separate User's Guide.

Overview

ANSYS FLUENT offers three different models for the detached eddy simulation: the Spalart-Allmaras model, the realizable k-ϵ model, and the SST k-ω model.

In the DES approach, the unsteady RANS models are employed in the boundary layer, while the LES treatment is applied to the separated regions. The LES region is normally associated with the core turbulent region where large unsteady turbulence scales play a dominant role. In this region, the DES models recover LES-like subgrid models. In the near-wall region, the respective RANS models are recovered.

DES models have been specifically designed to address high Reynolds number wall bounded flows, where the cost of a near-wall resolving Large Eddy Simulation would be prohibitive. The difference with the LES model is that it relies only on the required resolution in the boundary layers. The application of DES, however, may still require significant CPU resources and therefore, as a general guideline, it is recommended that the conventional turbulence models employing the Reynolds-averaged approach be used for practical calculations.

The DES models, often referred to as the hybrid LES/RANS models combine RANS modeling with LES for applications such as high-Re external aerodynamics simulations. In ANSYS FLUENT, the DES model is based on the one-equation Spalart-Allmaras model, the realizable k-ϵ model, and the SST k-ω model. The computational costs, when using the DES models, is less than LES computational costs, but greater than RANS.

4.10.1 Spalart-Allmaras Based DES Model

The standard Spalart-Allmaras model uses the distance to the closest wall as the definition for the length scale d, which plays a major role in determining the level of production and destruction of turbulent viscosity (Equations 4.3-6, 4.3-12, and 4.3-15). The DES model, as proposed by Shur et al. [314] replaces d everywhere with a new length scale $\widetilde{d}$, defined as

$$\widetilde{d} = \min(d, C_{\rm des}\Delta) \tag{4.10-1}$$

where the grid spacing, Δ, is based on the largest grid space in the x, y, or z directions forming the computational cell. The empirical constant $C_{\rm des}$ has a value of 0.65.

For a typical RANS grid with a high aspect ratio in the boundary layer, and where the wall-parallel grid spacing usually exceeds δ, where δ is the size of the boundary layer, Equation 4.10-1 will ensure that the DES model is in the RANS mode for the entire boundary layer. However, in case of an ambiguous grid definition, where $\Delta << \delta$, the DES limiter can activate the LES mode inside the boundary layer, where the grid is not fine enough to sustain resolved turbulence. Therefore, a new formulation [332] of DES is available in ANSYS FLUENT to preserve the RANS mode throughout the boundary layer. This is known as the delayed option or DDES for delayed DES.

The DES length scale $\widetilde{d}$ is re-defined according to:

$$\widetilde{d} = d - f_d \max(0, d - C_{\rm des}\Delta) \tag{4.10-2}$$

where f_d is given by:

$$f_d = 1 - \tanh\left((8r_d)^3\right) \tag{4.10-3}$$

This formulation is the default settings.

4.10.2 Realizable k-ϵ Based DES Model

This DES model is similar to the Realizable k-ϵ model discussed in Section 4.4.3: Realizable k-ϵ Model, with the exception of the dissipation term in the k equation. In the DES model, the Realizable k-ϵ RANS dissipation term is modified such that:

$$Y_k = \frac{\rho k^{\frac{3}{2}}}{l_{des}} \tag{4.10-4}$$

where

$$l_{des} = min(l_{rke}, l_{les}) \tag{4.10-5}$$

$$l_{rke} = \frac{k^{\frac{3}{2}}}{\epsilon} \tag{4.10-6}$$

$$l_{les} = C_{\rm des}\Delta \tag{4.10-7}$$

where $C_{\rm des}$ is a calibration constant used in the DES model and has a value of 0.61 and Δ is the maximum local grid spacing $(\Delta x, \Delta y, \Delta z)$.

For the case where $l_{des} = l_{rke}$, you will obtain an expression for the dissipation of the k formulation for the Realizable k-ϵ model (Section 4.4.3: Realizable k-ϵ Model):
$Y_k = \rho\epsilon$ Similarly to the Spalart-Allmaras model, the delayed concept can be applied as well to the Realizable DES model to preserve the RANS mode throughout the boundary layer. The DES length l_{des} in Equation 4.10-8 is redefined such that

$$l_{des} = l_{rke} - f_d \max(0, l_{rke} - C_{\rm des}\Delta) \tag{4.10-8}$$

4.10.3 SST k-ω Based DES Model

The dissipation term of the turbulent kinetic energy (see Section 4.5.1: Modeling the Turbulence Dissipation) is modified for the DES turbulence model as described in Menter's work [225] such that

$$Y_k = \rho \beta^* k \omega F_{DES} \tag{4.10-9}$$

where F_{DES} is expressed as

$$F_{DES} = max\left(\frac{L_t}{C_{\text{des}}\Delta}, 1\right) \tag{4.10-10}$$

where C_{des} is a calibration constant used in the DES model and has a value of 0.61, Δ is the maximum local grid spacing $(\Delta x, \Delta y, \Delta z)$.

The turbulent length scale is the parameter that defines this RANS model:

$$L_t = \frac{\sqrt{k}}{\beta^* \omega} \tag{4.10-11}$$

The DES-SST model also offers the option to "protect" the boundary layer from the limiter (delayed option). This is achieved with the help of the zonal formulation of the SST model. F_{DES} is modified according to

$$F_{DES} = max\left(\frac{L_t}{C_{\text{des}}\Delta}(1 - F_{SST}), 1\right) \tag{4.10-12}$$

with $F_{SST} = 0, F_1, F_2$, where F_1 and F_2 are the blending functions of the SST model. The default settings use F_2.

4.11 Large Eddy Simulation (LES) Model

This section describes the theory behind the Large Eddy Simulation (LES) model. Information is presented in the following sections:

- Section 4.11.1: Overview
- Section 4.11.2: Filtered Navier-Stokes Equations
- Section 4.11.3: Subgrid-Scale Models
- Section 4.11.4: Inlet Boundary Conditions for the LES Model

For details about using the model in ANSYS FLUENT, see Chapter 12: Modeling Turbulence and Section 12.12: Setting Up the Large Eddy Simulation Model in the separate User's Guide.

4.11.1 Overview

Turbulent flows are characterized by eddies with a wide range of length and time scales. The largest eddies are typically comparable in size to the characteristic length of the mean flow. The smallest scales are responsible for the dissipation of turbulence kinetic energy.

It is possible, in theory, to directly resolve the whole spectrum of turbulent scales using an approach known as direct numerical simulation (DNS). No modeling is required in DNS. However, DNS is not feasible for practical engineering problems involving high Reynolds number flows. The cost required for DNS to resolve the entire range of scales is proportional to Re_t^3, where Re_t is the turbulent Reynolds number. Clearly, for high Reynolds numbers, the cost becomes prohibitive.

In LES, large eddies are resolved directly, while small eddies are modeled. Large eddy simulation (LES) thus falls between DNS and RANS in terms of the fraction of the resolved scales. The rationale behind LES can be summarized as follows:

- Momentum, mass, energy, and other passive scalars are transported mostly by large eddies.
- Large eddies are more problem-dependent. They are dictated by the geometries and boundary conditions of the flow involved.
- Small eddies are less dependent on the geometry, tend to be more isotropic, and are consequently more universal.
- The chance of finding a universal turbulence model is much higher for small eddies.

Resolving only the large eddies allows one to use much coarser mesh and larger timesstep sizes in LES than in DNS. However, LES still requires substantially finer meshes than those typically used for RANS calculations. In addition, LES has to be run for a sufficiently long flow-time to obtain stable statistics of the flow being modeled. As a result, the computational cost involved with LES is normally orders of magnitudes higher than that for steady RANS calculations in terms of memory (RAM) and CPU time. Therefore, high-performance computing (e.g., parallel computing) is a necessity for LES, especially for industrial applications.

The following sections give details of the governing equations for LES, the subgrid-scale turbulence models, and the boundary conditions.

4.11.2 Filtered Navier-Stokes Equations

The governing equations employed for LES are obtained by filtering the time-dependent Navier-Stokes equations in either Fourier (wave-number) space or configuration (physical) space. The filtering process effectively filters out the eddies whose scales are smaller than the filter width or grid spacing used in the computations. The resulting equations thus govern the dynamics of large eddies.

A filtered variable (denoted by an overbar) is defined by

$$\overline{\phi}(\mathbf{x}) = \int_{\mathcal{D}} \phi(\mathbf{x}')G(\mathbf{x},\mathbf{x}')d\mathbf{x}' \tag{4.11-1}$$

where $\mathcal{D}$ is the fluid domain, and G is the filter function that determines the scale of the resolved eddies.

In ANSYS FLUENT, the finite-volume discretization itself implicitly provides the filtering operation:

$$\overline{\phi}(\mathbf{x}) = \frac{1}{V}\int_{\nu} \phi(\mathbf{x}')\ d\mathbf{x}', \quad \mathbf{x}' \in \nu \tag{4.11-2}$$

where V is the volume of a computational cell. The filter function, $G(\mathbf{x},\mathbf{x}')$, implied here is then

$$G(\mathbf{x},\mathbf{x}') = \left\{ \begin{array}{ll} 1/V, & \mathbf{x}' \in \nu \\ 0, & \mathbf{x}' \text{ otherwise} \end{array} \right. \tag{4.11-3}$$

The LES capability in ANSYS FLUENT is applicable to compressible flows. For the sake of concise notation, however, the theory is presented here for incompressible flows.

Filtering the Navier-Stokes equations, one obtains

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_i}(\rho \overline{u}_i) = 0 \tag{4.11-4}$$

and

$$\frac{\partial}{\partial t}(\rho \overline{u}_i) + \frac{\partial}{\partial x_j}(\rho \overline{u}_i \overline{u}_j) = \frac{\partial}{\partial x_j}(\sigma_{ij}) - \frac{\partial \overline{p}}{\partial x_i} - \frac{\partial \tau_{ij}}{\partial x_j} \tag{4.11-5}$$

where σ_{ij} is the stress tensor due to molecular viscosity defined by

$$\sigma_{ij} \equiv \left[\mu\left(\frac{\partial \overline{u}_i}{\partial x_j} + \frac{\partial \overline{u}_j}{\partial x_i}\right)\right] - \frac{2}{3}\mu\frac{\partial \overline{u}_l}{\partial x_l}\delta_{ij} \tag{4.11-6}$$

and τ_{ij} is the subgrid-scale stress defined by

$$\tau_{ij} \equiv \rho\overline{u_i u_j} - \rho\overline{u}_i\overline{u}_j \tag{4.11-7}$$

4.11.3 Subgrid-Scale Models

The subgrid-scale stresses resulting from the filtering operation are unknown, and require modeling. The subgrid-scale turbulence models in ANSYS FLUENT employ the Boussinesq hypothesis [130] as in the RANS models, computing subgrid-scale turbulent stresses from

$$\tau_{ij} - \frac{1}{3}\tau_{kk}\delta_{ij} = -2\mu_t\overline{S}_{ij} \tag{4.11-8}$$

where μ_t is the subgrid-scale turbulent viscosity. The isotropic part of the subgrid-scale stresses τ_{kk} is not modeled, but added to the filtered static pressure term. $\overline{S}_{ij}$ is the rate-of-strain tensor for the resolved scale defined by

$$\overline{S}_{ij} \equiv \frac{1}{2}\left(\frac{\partial\overline{u}_i}{\partial x_j} + \frac{\partial\overline{u}_j}{\partial x_i}\right) \tag{4.11-9}$$

For compressible flows, it is convenient to introduce the density-weighted (or Favre) filtering operator:

$$\widetilde{\phi} = \frac{\overline{\rho\phi}}{\overline{\rho}} \tag{4.11-10}$$

The Favre Filtered Navier-Stokes equation takes the same form as Equation 4.11-5. The compressible form of the subgrid stress tensor is defined as:

$$\tau_{ij} = \overline{\rho}\widetilde{u_i u_j} - \overline{\rho}\widetilde{u}_i\widetilde{u}_j \tag{4.11-11}$$

This term is split into its isotropic and deviatoric parts

$$\tau_{ij} = \underbrace{\tau_{ij} - \frac{1}{3}\tau_{kk}\delta_{ij}}_{\text{deviatoric}} + \underbrace{\frac{1}{3}\tau_{kk}\delta_{ij}}_{\text{isotropic}} \tag{4.11-12}$$

The deviatoric part of the subgrid-scale stress tensor is modeled using the compressible form of the Smagorinsky model:

$$\tau_{ij} - \frac{1}{3}\tau_{kk}\delta_{ij} = 2\mu_t(S_{ij} - \frac{1}{3}\S_{kk}\delta_{ij}) \tag{4.11-13}$$

As for incompressible flows, the term involving τ_{kk} can be added to the filtered pressure or simply neglected [89]. Indeed, this term can be re-written as $\tau_{kk} = \gamma M^2{}_{sgs}\overline{p}$ where M_{sgs} is the subgrid Mach number. This subgrid Mach number can be expected to be small when the turbulent Mach number of the flow is small.

ANSYS FLUENT offers four models for μ_t: the Smagorinsky-Lilly model, the dynamic Smagorinsky-Lilly model, the WALE model, and the dynamic kinetic energy subgrid-scale model.

The subgrid-scale turbulent flux of a scalar, ϕ, is modeled using s subgrid-scale turbulent Prandtl number by

$$q_j = -\frac{\mu_t}{\sigma_t}\frac{\partial \phi}{\partial x_j} \tag{4.11-14}$$

where q_j is the subgrid-scale flux.

In the dynamic models, the subgrid-scale turbulent Prandtl number or Schmidt number is obtained by applying the dynamic procedure originally proposed by Germano [106] to the subgrid-scale flux.

Smagorinsky-Lilly Model

This simple model was first proposed by Smagorinsky [320]. In the Smagorinsky-Lilly model, the eddy-viscosity is modeled by

$$\mu_t = \rho L_s^2 \left|\overline{S}\right| \tag{4.11-15}$$

where L_s is the mixing length for subgrid scales and $\left|\overline{S}\right| \equiv \sqrt{2\overline{S}_{ij}\overline{S}_{ij}}$. In ANSYS FLUENT, L_s is computed using

$$L_s = \min\left(\kappa d, C_s \Delta\right) \tag{4.11-16}$$

where κ is the von Kármán constant, d is the distance to the closest wall, C_s is the Smagorinsky constant, and Δ is the local grid scale. In ANSYS FLUENT, Δ is computed according to the volume of the computational cell using

$$\Delta = V^{1/3} \tag{4.11-17}$$

Lilly derived a value of 0.17 for C_s for homogeneous isotropic turbulence in the inertial subrange. However, this value was found to cause excessive damping of large-scale fluctuations in the presence of mean shear and in transitional flows as near solid boundary, and has to be reduced in such regions. In short, C_s is not a universal constant, which is the most serious shortcoming of this simple model. Nonetheless, a C_s value of around

0.1 has been found to yield the best results for a wide range of flows, and is the default value in ANSYS FLUENT.

Dynamic Smagorinsky-Lilly Model

Germano et al. [106] and subsequently Lilly [197] conceived a procedure in which the Smagorinsky model constant, C_s, is dynamically computed based on the information provided by the resolved scales of motion. The dynamic procedure thus obviates the need for users to specify the model constant C_s in advance.

The concept of the dynamic procedure is to apply a second filter (called the test filter) to the equations of motion. The new filter width $\hat{\Delta}$ is equal to twice the grid filter width Δ. Both filters produce a resolved flow field. The difference between the two resolved fields is the contribution of the small scales whose size is in between the grid filter and the test filter. The information related to these scales is used to compute the model constant. In ANSYS FLUENT, the variable density formulation of the model is considered as explained below.

At the test filtered field level, the SGS stress tensor can be expressed as:

$$T_{ij} = \widehat{\overline{\rho u_i u_j}} - (\widehat{\overline{\rho u_i}}\ \widehat{\overline{\rho u_j}} / \widehat{\overline{\rho}}) \tag{4.11-18}$$

Both T_{ij} and τ_{ij} are modeled in the same way with the Smagorinsky-Lilly model, assuming scale similarity:

$$\tau_{ij} = -2C\overline{\rho}\Delta^2\widetilde{S}|(\widetilde{S}_{ij} - \frac{1}{3}\widetilde{S}_{kk}\delta_{ij}) \tag{4.11-19}$$

$$T_{ij} = -2C\widehat{\overline{\rho}}\widehat{\Delta}^2|\widehat{\widetilde{S}}|(\widehat{\widetilde{S}}_{ij} - \frac{1}{3}\widehat{\widetilde{S}}_{kk}\delta_{ij}) \tag{4.11-20}$$

In Equation 4.11-19 and Equation 4.11-20, the coefficient C is asumed to be the same and independent of the filtering process (note that per Equation 4.11-25, $C = C_s^2$). The grid filtered SGS and the test-filtered SGS are related by the Germano identity [106] such that:

$$Ł_{ij} = T_{ij} - \widehat{\tau_{ij}} = \overline{\rho}\widehat{\widetilde{u}_i\widetilde{u}_j} - \frac{1}{\widehat{\overline{\rho}}}(\widehat{\overline{\rho}\widetilde{u}_i}\widehat{\overline{\rho}\widetilde{u}_j}) \tag{4.11-21}$$

Where L_{ij} is computable from the resolved large eddy field. Substituting the grid-filter Smagorinsky-Lilly model and Equation 4.11-20 into Equation 4.11-21, the following expressions can be derived to solve for C with the contraction obtained from the least square analysis of Lilly (1992).

$$C = \frac{(L_{ij} - L_{kk}\delta_{ij}/3)}{M_{ij}M_{ij}} \tag{4.11-22}$$

With

$$M_{ij} = -2\left(\widehat{\Delta}^2\widehat{\overline{\rho}}\ |\widehat{\widetilde{S}}|\widehat{\widetilde{S}}_{ij} - \Delta^2\overline{\rho}\ \widehat{|\widetilde{S}|\widetilde{S}_{ij}}\right) \tag{4.11-23}$$

More details of the model implementation in ANSYS FLUENT and its validation can be found in [165].

The $C_s = \sqrt{C}$ obtained using the dynamic Smagorinsky-Lilly model varies in time and space over a fairly wide range. To avoid numerical instability, both the numerator and the denominator in Equation 4.11-22 are locally averaged (or filtered) using the test-filter. In ANSYS FLUENT, C_s is also clipped at zero and 0.23 by default.

Wall-Adapting Local Eddy-Viscosity (WALE) Model

In the WALE model [248], the eddy viscosity is modeled by

$$\mu_t = \rho L_s^2 \frac{(S_{ij}^d S_{ij}^d)^{3/2}}{(\overline{S}_{ij}\overline{S}_{ij})^{5/2} + (S_{ij}^d S_{ij}^d)^{5/4}} \tag{4.11-24}$$

where L_s and S_{ij}^d in the WALE model are defined, respectively, as

$$L_s = \min\left(\kappa d, C_w V^{1/3}\right) \tag{4.11-25}$$

$$S_{ij}^d = \frac{1}{2}\left(\overline{g}_{ij}^2 + \overline{g}_{ji}^2\right) - \frac{1}{3}\delta_{ij}\overline{g}_{kk}^2\ ,\ \overline{g}_{ij} = \frac{\partial \overline{u_i}}{\partial x_j} \tag{4.11-26}$$

In ANSYS FLUENT, the default value of the WALE constant, C_w, is 0.325 and has been found to yield satisfactory results for a wide range of flow. The rest of the notation is the same as for the Smagorinsky-Lilly model. With this spatial operator, the WALE model is designed to return the correct wall asymptotic (y^3) behavior for wall bounded flows.

Dynamic Kinetic Energy Subgrid-Scale Model

The original and dynamic Smagorinsky-Lilly models, discussed previously, are essentially algebraic models in which subgrid-scale stresses are parameterized using the resolved velocity scales. The underlying assumption is the local equilibrium between the transferred energy through the grid-filter scale and the dissipation of kinetic energy at small subgrid scales. The subgrid-scale turbulence can be better modeled by accounting for the transport of the subgrid-scale turbulence kinetic energy.

The dynamic subgrid-scale kinetic energy model in ANSYS FLUENT replicates the model proposed by Kim and Menon [168].

The subgrid-scale kinetic energy is defined as

$$k_{\rm sgs} = \frac{1}{2}\left(\overline{u_k^2} - \overline{u}_k^2\right) \tag{4.11-27}$$

which is obtained by contracting the subgrid-scale stress in Equation 4.11-7.

The subgrid-scale eddy viscosity, μ_t, is computed using $k_{\rm sgs}$ as

$$\mu_t = C_k k_{\rm sgs}^{1/2} \Delta_f \tag{4.11-28}$$

where Δ_f is the filter-size computed from $\Delta_f \equiv V^{1/3}$.

The subgrid-scale stress can then be written as

$$\tau_{ij} - \frac{2}{3} k_{\rm sgs}\delta_{ij} = -2 C_k k_{\rm sgs}^{1/2} \Delta_f \overline{S}_{ij} \tag{4.11-29}$$

$k_{\rm sgs}$ is obtained by solving its transport equation

$$\frac{\partial \overline{k}_{\rm sgs}}{\partial t} + \frac{\partial \overline{u}_j \overline{k}_{sgs}}{\partial x_j} = -\tau_{ij}\frac{\partial \overline{u}_i}{\partial x_j} - C_\varepsilon \frac{k_{\rm sgs}^{3/2}}{\Delta_f} + \frac{\partial}{\partial x_j}\left(\frac{\mu_t}{\sigma_k}\frac{\partial k_{\rm sgs}}{\partial x_j}\right) \tag{4.11-30}$$

In the above equations, the model constants, C_k and C_ε, are determined dynamically [168]. σ_k is hardwired to 1.0. The details of the implementation of this model in ANSYS FLUENT and its validation is given by Kim [165].

4.11.4 Inlet Boundary Conditions for the LES Model

This section describes the three algorithms available in ANSYS FLUENT to model the fluctuating velocity at velocity inlet boundaries or pressure inlet boundaries.

No Perturbations

The stochastic components of the flow at the velocity-specified inlet boundaries are neglected if the No Perturbations option is used. In such cases, individual instantaneous velocity components are simply set equal to their mean velocity counterparts. This option is suitable only when the level of turbulence at the inflow boundaries is negligible or does not play a major role in the accuracy of the overall solution.

Vortex Method

To generate a time-dependent inlet condition, a random 2D vortex method is considered. With this approach, a perturbation is added on a specified mean velocity profile via a fluctuating vorticity field (i.e. two-dimensional in the plane normal to the streamwise direction). The vortex method is based on the Lagrangian form of the 2D evolution equation of the vorticity and the Biot-Savart law. A particle discretization is used to solve this equation. These particles, or "vortex points" are convected randomly and carry information about the vorticity field. If N is the number of vortex points and A is the area of the inlet section, the amount of vorticity carried by a given particle i is represented by the circulation Γ_i and an assumed spatial distribution η:

$$\Gamma_i(x,y) = 4\sqrt{\frac{\pi A k(x,y)}{3N[2\ln(3)-3\ln(2)]}} \quad (4.11\text{-}31)$$

$$\eta(\vec{x}) = \frac{1}{2\pi\sigma^2}\left(2e^{-|x|^2/2\sigma^2}-1\right)2e^{-|x|^2/2\sigma^2} \quad (4.11\text{-}32)$$

where k is the turbulence kinetic energy. The parameter σ provides control over the size of a vortex particle. The resulting discretization for the velocity field is given by

$$\vec{u}(\vec{x}) = \frac{1}{2\pi}\sum_{i=1}^{N}\Gamma_i\frac{((\vec{x}_i-\vec{x})\times\vec{z})(1-e^{|\vec{x}-\vec{x}'|^2/2\sigma^2})}{|\vec{x}-\vec{x}'_i|^2} \quad (4.11\text{-}33)$$

Where $\vec{z}$ is the unit vector in the streamwise direction. Originally [311], the size of the vortex was fixed by an ad hoc value of σ. To make the vortex method generally applicable, a local vortex size is specified through a turbulent mixing length hypothesis. σ is calculated from a known profile of mean turbulence kinetic energy and mean dissipation rate at the inlet according to the following:

$$\sigma = \frac{ck^{3/2}}{2\epsilon} \quad (4.11\text{-}34)$$

where $c = 0.16$. To ensure that the vortex will always belong to resolved scales, the minimum value of σ in Equation 4.11-34 is bounded by the local grid size. The sign of the circulation of each vortex is changed randomly each characteristic time scale τ. In the general implementation of the vortex method, this time scale represents the time necessary for a 2D vortex convected by the bulk velocity in the boundary normal direction to travel along n times its mean characteristic 2D size (σ_m), where n is fixed equal to 100 from numerical testing. The vortex method considers only velocity fluctuations in the plane normal to the streamwise direction.

In ANSYS FLUENT however, a simplified linear kinematic model (LKM) for the streamwise velocity fluctuations is used [219]. It is derived from a linear model that mimics the influence of the two-dimensional vortex in the streamwise mean velocity field. If the mean streamwise velocity U is considered as a passive scalar, the fluctuation u' resulting from the transport of U by the planar fluctuating velocity field v' is modeled by

$$u' = -\vec{v}' \cdot \vec{g} \tag{4.11-35}$$

where $\vec{g}$ is the unit vector aligned with the mean velocity gradient $\nabla\vec{U}$. When this mean velocity gradient is equal to zero, a random perturbation can be considered instead.

Since the fluctuations are equally distributed among the velocity components, only the prescribed kinetic energy profile can be fulfilled at the inlet of the domain. Farther downstream, the correct fluctuation distribution is recovered [219]. However, if the distribution of the normal fluctuations is known or can be prescribed at the inlet, a rescaling technique can be applied to the synthetic flow field in order to fulfill the normal statistic fluctuations $< uu >$, $< vv >$, and $< ww >$ as given at the inlet.

With the rescaling procedure, the velocity fluctuations are expressed according to:

$$u_i'^{*} = u_i' \frac{\sqrt{< u_i u_i >}}{\sqrt{2/3k}} \tag{4.11-36}$$

This also results in an improved representation of the turbulent flow field downstream of the inlet. This rescaling procedure is used only if the Reynolds-Stress Components is specified as the Reynolds-Stress Specification Method, instead of the default option K or Turbulence Intensity.

i Since the vortex method theory is based on the modification of the velocity field normal to the streamwise direction, it is imperative that you create an inlet plane normal (or as close as possible) to the streamwise velocity direction.

Spectral Synthesizer

The spectral synthesizer provides an alternative method of generating fluctuating velocity components. It is based on the random flow generation technique originally proposed by Kraichnan [171] and modified by Smirnov et al. [321]. In this method, fluctuating velocity components are computed by synthesizing a divergence-free velocity-vector field from the summation of Fourier harmonics. In ANSYS FLUENT, the number of Fourier harmonics is fixed to 100.

Both the vortex method and the spectral synthesizer are available for velocity inlet and pressure inlet boundary conditions. For the velocity inlet, the fluctuations are added on the mean specified velocity. For the pressure inlet, virtual body forces are employed in the momentum equations to add the reconstructed turbulent fluctuations to the velocity field. These virtual body forces are considered only in the first LES cells close to the inlet.
Both methods are also available for the DES models. However, note that such unsteady boundary conditions are appropriate and effective mainly for external aerodynamic flows. For internal flows, if the inlet is inside a full RANS zone, the fluctuations generated by both methods will be rapidly damped by the RANS turbulent eddy viscosity. Note also that whether the inlet will be fully or partly covered by a RANS zone will depend on the mesh and on the DES model.
Finally it should be noted that both methods require realistic inlet conditions (U,k,ϵ profiles) which can be obtained from separate RANS simulations. Unrealistic ("flat") turbulent profiles at inlets will generate unrealistic turbulent eddies at inlets.

4.12 Near-Wall Treatments for Wall-Bounded Turbulent Flows

Information about near-wall tratments for wall-bounded turbulent flows is presented in the following sections:

- Section 4.12.1: Overview
- Section 4.12.2: Standard Wall Functions
- Section 4.12.3: Non-Equilibrium Wall Functions
- Section 4.12.4: Enhanced Wall Treatment
- Section 4.12.5: User-Defined Wall Functions
- Section 4.12.6: LES Near-Wall Treatment

4.12.1 Overview

Turbulent flows are significantly affected by the presence of walls. Obviously, the mean velocity field is affected through the no-slip condition that has to be satisfied at the wall. However, the turbulence is also changed by the presence of the wall in non-trivial ways. Very close to the wall, viscous damping reduces the tangential velocity fluctuations, while kinematic blocking reduces the normal fluctuations. Toward the outer part of the near-wall region, however, the turbulence is rapidly augmented by the production of turbulence kinetic energy due to the large gradients in mean velocity.

The near-wall modeling significantly impacts the fidelity of numerical solutions, inasmuch as walls are the main source of mean vorticity and turbulence. After all, it is in the near-wall region that the solution variables have large gradients, and the momentum and other scalar transports occur most vigorously. Therefore, accurate representation of the flow in the near-wall region determines successful predictions of wall-bounded turbulent flows.

The k-ϵ models, the RSM, and the LES model are primarily valid for turbulent core flows (i.e., the flow in the regions somewhat far from walls). Consideration therefore needs to be given as to how to make these models suitable for wall-bounded flows. The Spalart-Allmaras and k-ω models were designed to be applied throughout the boundary layer, provided that the near-wall mesh resolution is sufficient.

Numerous experiments have shown that the near-wall region can be largely subdivided into three layers. In the innermost layer, called the "viscous sublayer", the flow is almost laminar, and the (molecular) viscosity plays a dominant role in momentum and heat or mass transfer. In the outer layer, called the fully-turbulent layer, turbulence plays a major role. Finally, there is an interim region between the viscous sublayer and the fully turbulent layer where the effects of molecular viscosity and turbulence are equally important. Figure 4.12.1 illustrates these subdivisions of the near-wall region, plotted in semi-log coordinates.

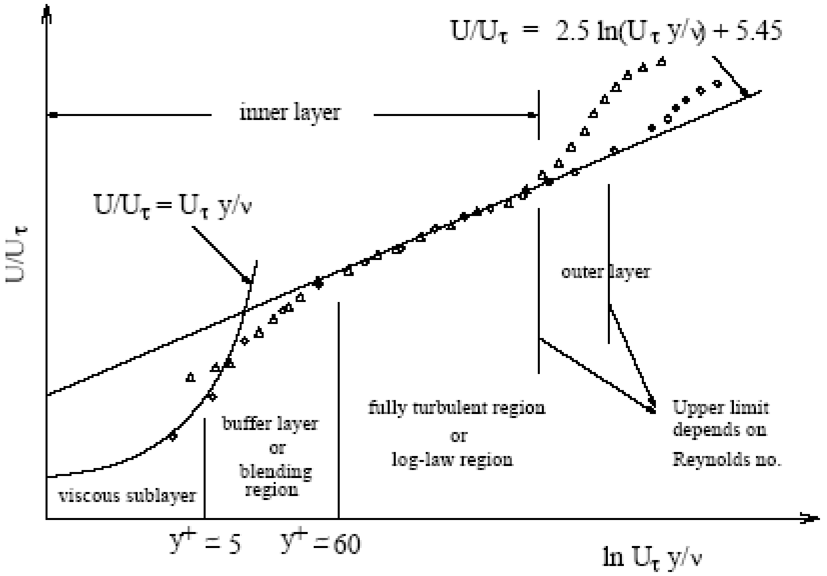


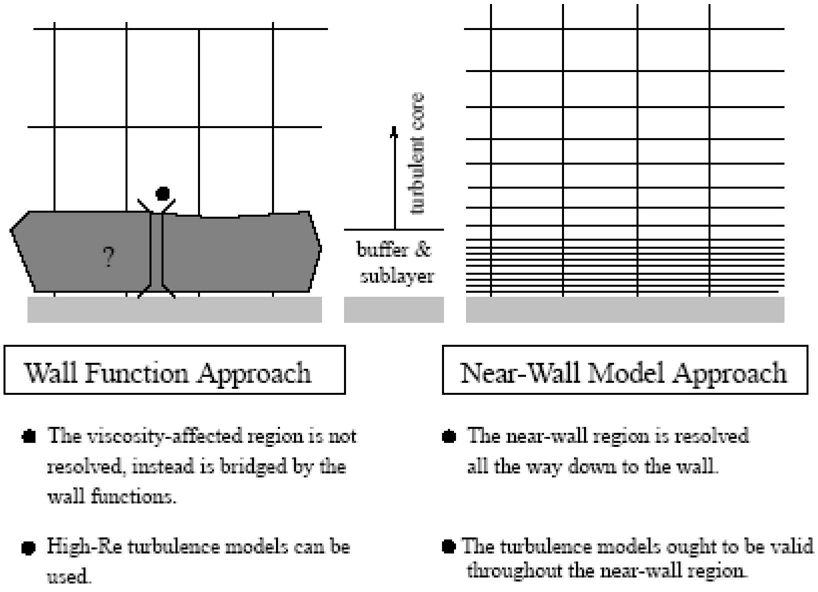
Figure 4.12.1: Subdivisions of the Near-Wall Region

In Figure 4.12.1, $y^+ \equiv \rho u_\tau y/\mu$, where u_τ is the friction velocity, defined as $\sqrt{\frac{\tau_w}{\rho}}$.

Wall Functions vs. Near-Wall Model

Traditionally, there are two approaches to modeling the near-wall region. In one approach, the viscosity-affected inner region (viscous sublayer and buffer layer) is not resolved. Instead, semi-empirical formulas called “wall functions” are used to bridge the viscosity-affected region between the wall and the fully-turbulent region. The use of wall functions obviates the need to modify the turbulence models to account for the presence of the wall.

In another approach, the turbulence models are modified to enable the viscosity-affected region to be resolved with a mesh all the way to the wall, including the viscous sublayer. For the purposes of discussion, this will be termed the “near-wall modeling” approach. These two approaches are depicted schematically in Figure 4.12.2.



- The viscosity-affected region is not resolved, instead is bridged by the wall functions.
- High-Re turbulence models can be used.

- The near-wall region is resolved all the way down to the wall.
- The turbulence models ought to be valid throughout the near-wall region.

Figure 4.12.2: Near-Wall Treatments in ANSYS FLUENT

In most high-Reynolds-number flows, the wall function approach substantially saves computational resources, because the viscosity-affected near-wall region, in which the solution variables change most rapidly, does not need to be resolved. The wall-function approach is popular because it is economical, robust, and can be reasonably accurate. It is a practical option for the near-wall treatments for industrial flow simulations.

The wall-function approach, however, is inadequate in situations where the low-Reynolds-number effects are pervasive and the assumptions underlying the wall functions cease to be valid. Such situations require near-wall models that are valid in the viscosity-affected region and accordingly integrable all the way to the wall.

ANSYS FLUENT provides both the wall-function approach and the near-wall modeling approach.

Wall Functions

Wall functions are a set of semi-empirical formulas and functions that in effect “bridge” or “link” the solution variables at the near-wall cells and the corresponding quantities on the wall. The wall functions comprise

- laws-of-the-wall for the mean velocity and temperature (or other scalars)
- formulae for the near-wall turbulent quantities

Depending on the choice of turbulent model, ANSYS FLUENT offers three to four choices of wall-function approaches:

- Standard Wall Functions
- Non-Equilibrium Wall Functions
- Enhanced Wall Functions (as a part of EWT)
- User-Defined Wall Functions

4.12.2 Standard Wall Functions

The standard wall functions in ANSYS FLUENT are based on the work of Launder and Spalding [181], and have been most widely used in industrial flows. They are provided as a default option in ANSYS FLUENT.

Momentum

The law-of-the-wall for mean velocity yields

$$U^* = \frac{1}{\kappa} \ln(Ey^*) \tag{4.12-1}$$

where

$$U^* \equiv \frac{U_P C_\mu^{1/4} k_P^{1/2}}{\tau_w/\rho} \tag{4.12-2}$$

is the dimensionless velocity.

$$y^* \equiv \frac{\rho C_\mu^{1/4} k_P^{1/2} y_P}{\mu} \tag{4.12-3}$$

is the dimensionless distance from the wall.

and κ = von Kármán constant (= 0.4187)
E = empirical constant (= 9.793)
U_P = mean velocity of the fluid at the near-wall node P
k_P = turbulence kinetic energy at the near-wall node P
y_P = distance from point P to the wall
μ = dynamic viscosity of the fluid

The logarithmic law for mean velocity is known to be valid for $30 < y^* < 300$. In ANSYS FLUENT, the log-law is employed when $y^* > 11.225$.

When the mesh is such that $y^* < 11.225$ at the wall-adjacent cells, ANSYS FLUENT applies the laminar stress-strain relationship that can be written as

$$U^* = y^* \tag{4.12-4}$$

It should be noted that, in ANSYS FLUENT, the laws-of-the-wall for mean velocity and temperature are based on the wall unit, y^*, rather than y^+ ($\equiv \rho u_\tau y/\mu$). These quantities are approximately equal in equilibrium turbulent boundary layers.

Energy

Reynolds' analogy between momentum and energy transport gives a similar logarithmic law for mean temperature. As in the law-of-the-wall for mean velocity, the law-of-the-wall for temperature employed in ANSYS FLUENT comprises the following two different laws:

- linear law for the thermal conduction sublayer, or thermal viscous sublayer, where conduction is important
- logarithmic law for the turbulent region where effects of turbulence dominate conduction

The thickness of the thermal conduction layer is, in general, different from the thickness of the (momentum) viscous sublayer, and changes from fluid to fluid. For example, the thickness of the thermal sublayer for a high-Prandtl-number fluid (e.g., oil) is much less than its momentum sublayer thickness. For fluids of low Prandtl numbers (e.g., liquid metal), on the contrary, it is much larger than the momentum sublayer thickness.

$$\frac{\delta}{\delta_T} \approx \sqrt{\mathrm{Pr}} \tag{4.12-5}$$

In highly compressible flows, the temperature distribution in the near-wall region can be significantly different from that of low subsonic flows, due to the heating by viscous dissipation. In ANSYS FLUENT, the temperature wall functions include the contribution from the viscous heating [357].

The law-of-the-wall implemented in ANSYS FLUENT has the following composite form:

$$T^* \equiv \frac{(T_w - T_P)\,\rho c_p k_P^{1/2}}{\dot{q}} = \begin{cases} \Pr y^* + \frac{1}{2}\rho \Pr \frac{C_\mu^{1/4} k_P^{1/2}}{\dot{q}} U_P^2 & (y^* < y_T^*) \\ \Pr_t \left[\frac{1}{\kappa}\ln(Ey^*) + P\right] + & \\ \frac{1}{2}\rho \frac{C_\mu^{1/4} k_P^{1/2}}{\dot{q}} \left\{\Pr_t U_P^2 + (\Pr - \Pr_t) U_c^2\right\} & (y^* > y_T^*) \end{cases} \tag{4.12-6}$$

where P is computed by using the formula given by Jayatilleke [150]:

$$P = 9.24\left[\left(\frac{\Pr}{\Pr_t}\right)^{3/4} - 1\right]\left[1 + 0.28 e^{-0.007\Pr/\Pr_t}\right] \tag{4.12-7}$$

and

k_P	=	turbulent kinetic energy at the first near-wall node P
ρ	=	density of fluid
c_p	=	specific heat of fluid
$\dot{q}$	=	wall heat flux
T_P	=	temperature at the first near-wall node P
T_w	=	temperature at the wall
Pr	=	molecular Prandtl number ($\mu c_p / k_f$)
$\Pr_t$	=	turbulent Prandtl number (0.85 at the wall)
A	=	Van Driest constant (= 26)
U_c	=	mean velocity magnitude at $y^* = y_T^*$

Note that, for the pressure-based solver, the terms

$$\frac{1}{2}\rho \Pr \frac{C_\mu^{1/4} k_P^{1/2}}{\dot{q}} U_P^2$$

and

$$\frac{1}{2}\rho \frac{C_\mu^{1/4} k_P^{1/2}}{\dot{q}} \left\{\Pr_t U_P^2 + (\Pr - \Pr_t) U_c^2\right\}$$

will be included in Equation 4.12-6 only for compressible flow calculations.

The non-dimensional thermal sublayer thickness, y_T^*, in Equation 4.12-6 is computed as the y^* value at which the linear law and the logarithmic law intersect, given the molecular Prandtl number of the fluid being modeled.

The procedure of applying the law-of-the-wall for temperature is as follows. Once the physical properties of the fluid being modeled are specified, its molecular Prandtl number is computed. Then, given the molecular Prandtl number, the thermal sublayer thickness, y_T^*, is computed from the intersection of the linear and logarithmic profiles, and stored.

During the iteration, depending on the y^* value at the near-wall cell, either the linear or the logarithmic profile in Equation 4.12-6 is applied to compute the wall temperature T_w or heat flux $\dot{q}$ (depending on the type of the thermal boundary conditions).

The function for P given by Equation 4.12-7 is relevant for the smooth walls. For the rough walls, however, this function is modified as follows:

$$P_{\text{rough}} = 3.15\text{Pr}^{0.695}\left(\frac{1}{E'} - \frac{1}{E}\right)^{0.359} + \left(\frac{E'}{E}\right)^{0.6} P \tag{4.12-8}$$

where E' is the wall function constant modified for the rough walls, defined by $E' = E/f_r$. To find a description of the roughness function f_r, you may refer to Equation 7.3-42 in Section 7.3.14: Wall Roughness Effects in Turbulent Wall-Bounded Flows in the separate User's Guide.

Species

When using wall functions for species transport, ANSYS FLUENT assumes that species transport behaves analogously to heat transfer. Similarly to Equation 4.12-6, the law-of-the-wall for species can be expressed for constant property flow with no viscous dissipation as

$$Y^* \equiv \frac{(Y_{i,w} - Y_i)\,\rho C_\mu^{1/4} k_P^{1/2}}{J_{i,w}} = \begin{cases} \text{Sc}\, y^* & (y^* < y_c^*) \\ \text{Sc}_t \left[\frac{1}{\kappa}\ln(Ey^*) + P_c\right] & (y^* > y_c^*) \end{cases} \tag{4.12-9}$$

where Y_i is the local species mass fraction, Sc and Sc_t are molecular and turbulent Schmidt numbers, and $J_{i,w}$ is the diffusion flux of species i at the wall. Note that P_c and y_c^* are calculated in a similar way as P and y_T^*, with the difference being that the Prandtl numbers are always replaced by the corresponding Schmidt numbers.

Turbulence

In the k-ϵ models and in the RSM (if the option to obtain wall boundary conditions from the k equation is enabled), the k equation is solved in the whole domain including the wall-adjacent cells. The boundary condition for k imposed at the wall is

$$\frac{\partial k}{\partial n} = 0 \tag{4.12-10}$$

where n is the local coordinate normal to the wall.

The production of kinetic energy, G_k, and its dissipation rate, ϵ, at the wall-adjacent cells, which are the source terms in the k equation, are computed on the basis of the local equilibrium hypothesis. Under this assumption, the production of k and its dissipation rate are assumed to be equal in the wall-adjacent control volume.

Thus, the production of k is based on the logarithmic law and is computed from

$$G_k \approx \tau_w \frac{\partial U}{\partial y} = \tau_w \frac{\tau_w}{\kappa \rho k_P^{1/2} y_P} \tag{4.12-11}$$

and ϵ is computed from

$$\epsilon_P = \frac{C_\mu^{3/4} k_P^{3/2}}{\kappa y_P} \tag{4.12-12}$$

The ϵ equation is not solved at the wall-adjacent cells, but instead is computed using Equation 4.12-12. ω and Reynolds stress equations are solved as detailed in Sections 4.5.3 and 4.9.9, respectively.

Note that, as shown here, the wall boundary conditions for the solution variables, including mean velocity, temperature, species concentration, k, and ϵ, are all taken care of by the wall functions. Therefore, you do not need to be concerned about the boundary conditions at the walls.

The standard wall functions described so far are provided as a default option in ANSYS FLUENT. The standard wall functions work reasonably well for a broad range of wall-bounded flows. However, they tend to become less reliable when the flow situations depart from the ideal conditions that are assumed in their derivation. Among others, the constant-shear and local equilibrium assumptions are the ones that most restrict the universality of the standard wall functions. Accordingly, when the near-wall flows are subjected to severe pressure gradients, and when the flows are in strong non-equilibrium, the quality of the predictions is likely to be compromised.

The non-equilibrium wall functions offered as an additional option that can potentially improve the results in such situations.

i Standard wall functions are available with the following viscous models:

- k-ϵ models
- Reynolds Stress models

Scalable Wall Functions

The purpose of scalable wall functions is to force the usage of the log law in conjunction with the standard wall functions approach. This is achieved by introducing a limiter in the y* calculations such that

$$\widetilde{y*} = MAX(y*, y*_{limit}) \tag{4.12-13}$$

where $y*_{limit} = 11.225$. The use of Equation 4.12-13 in the context of the scalable wall functions concept is straightforward, i.e. the y* formulation used for any standard wall function formula is replaced by $\widetilde{y*}$.

Scalable wall functions can be enabled only when the standard wall functions are used and only via the following text command:

```
define/models/viscous/near-wall-treatment/scalable-wall-functions?
```

4.12.3 Non-Equilibrium Wall Functions

In addition to the standard wall function described above (which is the default near-wall treatment) a two-layer-based, non-equilibrium wall function [166] is also available. The key elements in the non-equilibrium wall functions are as follows:

- Launder and Spalding's log-law for mean velocity is sensitized to pressure-gradient effects.
- The two-layer-based concept is adopted to compute the budget of turbulence kinetic energy ($\overline{G_k}$,$\overline{\epsilon}$) in the wall-neighboring cells.

The law-of-the-wall for mean temperature or species mass fraction remains the same as in the standard wall functions described above.

The log-law for mean velocity sensitized to the pressure gradients is

$$\frac{\widetilde{U} C_\mu^{1/4} k^{1/2}}{\tau_w/\rho} = \frac{1}{\kappa} \ln\left(E \frac{\rho C_\mu^{1/4} k^{1/2} y}{\mu}\right) \tag{4.12-14}$$

where

$$\widetilde{U} = U - \frac{1}{2}\frac{dp}{dx}\left[\frac{y_v}{\rho\kappa\sqrt{k}} \ln\left(\frac{y}{y_v}\right) + \frac{y - y_v}{\rho\kappa\sqrt{k}} + \frac{y_v^2}{\mu}\right] \tag{4.12-15}$$

and y_v is the physical viscous sublayer thickness, and is computed from

$$y_v \equiv \frac{\mu y_v^*}{\rho C_\mu^{1/4} k_P^{1/2}} \tag{4.12-16}$$

where $y_v^* = 11.225$.

The non-equilibrium wall function employs the two-layer concept in computing the budget of turbulence kinetic energy at the wall-adjacent cells, which is needed to solve the k equation at the wall-neighboring cells. The wall-neighboring cells are assumed to consist of a viscous sublayer and a fully turbulent layer. The following profile assumptions for turbulence quantities are made:

$$\tau_t = \begin{cases} 0, & y < y_v \\ \tau_w, & y > y_v \end{cases} \quad k = \begin{cases} \left(\frac{y}{y_v}\right)^2 k_P, & y < y_v \\ k_P, & y > y_v \end{cases} \quad \epsilon = \begin{cases} \frac{2\nu k}{y^2}, & y < y_v \\ \frac{k^{3/2}}{C_\ell{}^* y}, & y > y_v \end{cases} \tag{4.12-17}$$

where $C_\ell{}^* = \kappa C_\mu^{-3/4}$, and y_v is the dimensional thickness of the viscous sublayer, defined in Equation 4.12-16.

Using these profiles, the cell-averaged production of k, $\overline{G_k}$, and the cell-averaged dissipation rate, $\bar{\epsilon}$, can be computed from the volume average of G_k and ϵ of the wall-adjacent cells. For quadrilateral and hexahedral cells for which the volume average can be approximated with a depth-average,

$$\overline{G_k} \equiv \frac{1}{y_n} \int_0^{y_n} \tau_t \frac{\partial U}{\partial y} \, dy = \frac{1}{\kappa y_n} \frac{\tau_w^2}{\rho C_\mu^{1/4} k_P^{1/2}} \ln\left(\frac{y_n}{y_v}\right) \tag{4.12-18}$$

and

$$\bar{\epsilon} \equiv \frac{1}{y_n} \int_0^{y_n} \epsilon \, dy = \frac{1}{y_n} \left[\frac{2\nu}{y_v} + \frac{k_P^{1/2}}{C_\ell{}^*} \ln\left(\frac{y_n}{y_v}\right) \right] k_P \tag{4.12-19}$$

where y_n is the height of the cell ($y_n = 2y_P$). For cells with other shapes (e.g., triangular and tetrahedral grids), the appropriate volume averages are used.

In Equations 4.12-18 and 4.12-19, the turbulence kinetic energy budget for the wall-neighboring cells is effectively depends on the proportions of the viscous sublayer and the fully turbulent layer, which varies widely from cell to cell in highly non-equilibrium flows. The nonequilibrium wall functions account for the effect of pressure gradients on the distortion of the velocity profiles. In such cases the assumption of local equilibrium, when the production of the turbulent kinetic energy is equal to the rate of its distruction, is no longer valid. Therefore, the non-equilibrium wall functions, in effect, partly account for the non-equilibrium effects that are neglected in the standard wall functions.

Standard Wall Functions vs. Non-Equilibrium Wall Functions

Because of the capability to partly account for the effects of pressure gradients, the non-equilibrium wall functions are recommended for use in complex flows involving separation, reattachment, and impingement where the mean flow and turbulence are subjected to pressure gradients and rapid changes. In such flows, improvements can be obtained, particularly in the prediction of wall shear (skin-friction coefficient) and heat transfer (Nusselt or Stanton number).

ℹ Non-equilibrium wall functions are available with the following turbulence closures:

- k-ϵ models
- Reynolds Stress Transport models

Limitations of the Wall Function Approach

The standard wall functions give reasonable predictions for the majority of high-Reynolds-number wall-bounded flows. The non-equilibrium wall functions further extend the applicability of the wall function approach by including the effects of pressure gradient; however, the above wall functions become less reliable when the flow conditions depart too much from the ideal conditions underlying the wall functions. Examples are as follows:

- Pervasive low-Reynolds-number or near-wall effects (e.g., flow through a small gap or highly viscous, low-velocity fluid flow).
- Massive transpiration through the wall (blowing/suction).
- Severe pressure gradients leading to boundary layer separations.
- Strong body forces (e.g., flow near rotating disks, buoyancy-driven flows).
- High three-dimensionality in the near-wall region (e.g., Ekman spiral flow, strongly skewed 3D boundary layers).

If any of the above listed features prevail in the flow you are modeling, and if it is considered critically important for the success of your simulation, you must employ the near-wall modeling approach combined with the adequate mesh resolution in the near-wall region. ANSYS FLUENT provides the enhanced wall treatment for such situations. This approach can be used with the k-ϵ and the RSM models.

4.12.4 Enhanced Wall Treatment

Enhanced wall treatment is a near-wall modeling method that combines a two-layer model with so-called enhanced wall functions. If the near-wall mesh is fine enough to be able to resolve the viscous sublayer (typically with the first near-wall node placed at $y^+ \approx 1$), then the enhanced wall treatment will be identical to the traditional two-layer zonal model (see below for details). However, the restriction that the near-wall mesh must be sufficiently fine everywhere might impose too large a computational requirement. Ideally, one would like to have a near-wall formulation that can be used with coarse meshes (usually referred to as wall-function meshes) as well as fine meshes (low-Reynolds-number meshes). In addition, excessive error should not be incurred for the intermediate meshes where the first near-wall node is placed neither in the fully turbulent region, where the wall functions are suitable, nor in the direct vicinity of the wall at $y^+ \approx 1$, where the low-Reynold-number approach is adequate.

To achieve the goal of having a near-wall modeling approach that will possess the accuracy of the standard two-layer approach for fine near-wall meshes and that, at the same time, will not significantly reduce accuracy for wall-function meshes, ANSYS FLUENT can combine the two-layer model with enhanced wall functions, as described in the following sections.

Two-Layer Model for Enhanced Wall Treatment

In ANSYS FLUENT's near-wall model, the viscosity-affected near-wall region is completely resolved all the way to the viscous sublayer. The two-layer approach is an integral part of the enhanced wall treatment and is used to specify both ϵ and the turbulent viscosity in the near-wall cells. In this approach, the whole domain is subdivided into a viscosity-affected region and a fully-turbulent region. The demarcation of the two regions is determined by a wall-distance-based, turbulent Reynolds number, Re_y, defined as

$$\mathrm{Re}_y \equiv \frac{\rho y \sqrt{k}}{\mu} \qquad (4.12\text{-}20)$$

where y is the wall-normal distance calculated at the cell centers. In ANSYS FLUENT, y is interpreted as the distance to the nearest wall:

$$y \equiv \min_{\vec{r}_w \in \Gamma_w} \|\vec{r} - \vec{r}_w\| \qquad (4.12\text{-}21)$$

where $\vec{r}$ is the position vector at the field point, and $\vec{r}_w$ is the position vector of the wall boundary. Γ_w is the union of all the wall boundaries involved. This interpretation allows y to be uniquely defined in flow domains of complex shape involving multiple walls. Furthermore, y defined in this way is independent of the mesh topology.

In the fully turbulent region ($\mathrm{Re}_y > \mathrm{Re}_y^*$; $\mathrm{Re}_y^* = 200$), the k-ϵ models or the RSM (described in Sections 4.4 and 4.9) are employed.

In the viscosity-affected near-wall region ($\mathrm{Re}_y < \mathrm{Re}_y^*$), the one-equation model of Wolfstein [382] is employed. In the one-equation model, the momentum equations and the k equation are retained as described in Sections 4.4 and 4.9. However, the turbulent viscosity, μ_t, is computed from

$$\mu_{t,\mathrm{2layer}} = \rho\, C_\mu \ell_\mu \sqrt{k} \tag{4.12-22}$$

where the length scale that appears in Equation 4.12-22 is computed from [51]

$$\ell_\mu = y C_\ell{}^* \left(1 - e^{-\mathrm{Re}_y/A_\mu}\right) \tag{4.12-23}$$

The two-layer formulation for turbulent viscosity described above is used as a part of the enhanced wall treatment, in which the two-layer definition is smoothly blended with the high-Reynolds-number μ_t definition from the outer region, as proposed by Jongen [153]:

$$\mu_{t,\mathrm{enh}} = \lambda_\epsilon \mu_t + (1 - \lambda_\epsilon)\mu_{t,\mathrm{2layer}} \tag{4.12-24}$$

where μ_t is the high-Reynolds-number definition as described in Section 4.4: Standard, RNG, and Realizable k-ϵ Models or 4.9 for the k-ϵ models or the RSM. A blending function, λ_ϵ, is defined in such a way that it is equal to unity away from walls and is zero in the vicinity of the walls. The blending function has the following form:

$$\lambda_\epsilon = \frac{1}{2}\left[1 + \tanh\left(\frac{\mathrm{Re}_y - \mathrm{Re}_y^*}{A}\right)\right] \tag{4.12-25}$$

The constant A determines the width of the blending function. By defining a width such that the value of λ_ϵ will be within 1% of its far-field value given a variation of $\Delta\mathrm{Re}_y$, the result is

$$A = \frac{|\Delta\mathrm{Re}_y|}{artanh(0.98)} \tag{4.12-26}$$

Typically, $\Delta\mathrm{Re}_y$ would be assigned a value that is between 5% and 20% of Re_y^*. The main purpose of the blending function λ_ϵ is to prevent solution convergence from being impeded when the value of μ_t obtained in the outer layer does not match with the value of μ_t returned by the Wolfstein model at the edge of the viscosity-affected region.

The ϵ field in the viscosity-affected region is computed from

$$\epsilon = \frac{k^{3/2}}{\ell_\epsilon} \tag{4.12-27}$$

The length scales that appear in Equation 4.12-27 are computed from Chen and Patel [51]:

$$\ell_\epsilon = yC_\ell^* \left(1 - e^{-\mathrm{Re}_y/A_\epsilon}\right) \tag{4.12-28}$$

If the whole flow domain is inside the viscosity-affected region ($\mathrm{Re}_y < 200$), ϵ is not obtained by solving the transport equation; it is instead obtained algebraically from Equation 4.12-27. ANSYS FLUENT uses a procedure for the blending of ϵ that is similar to the μ_t-blending in order to ensure a smooth transition between the algebraically-specified ϵ in the inner region and the ϵ obtained from solution of the transport equation in the outer region.

The constants in Equations 4.12-23 and 4.12-28, are taken from [51] and are as follows:

$$C_\ell^* = \kappa C_\mu^{-3/4}, \quad A_\mu = 70, \quad A_\epsilon = 2C_\ell^* \tag{4.12-29}$$

Enhanced Wall Functions

To have a method that can extend its applicability throughout the near-wall region (i.e., viscous sublayer, buffer region, and fully-turbulent outer region) it is necessary to formulate the law-of-the wall as a single wall law for the entire wall region. ANSYS FLUENT achieves this by blending the linear (laminar) and logarithmic (turbulent) laws-of-the-wall using a function suggested by Kader [155]:

$$u^+ = e^\Gamma u^+_{\mathrm{lam}} + e^{\frac{1}{\Gamma}} u^+_{\mathrm{turb}} \tag{4.12-30}$$

where the blending function is given by:

$$\Gamma = -\frac{a(y^+)^4}{1 + by^+} \tag{4.12-31}$$

where $a = 0.01$ and $b = 5$.

Similarly, the general equation for the derivative $\frac{du^+}{dy^+}$ is

$$\frac{du^+}{dy^+} = e^\Gamma \frac{du^+_{\mathrm{lam}}}{dy^+} + e^{\frac{1}{\Gamma}} \frac{du^+_{\mathrm{turb}}}{dy^+} \tag{4.12-32}$$

This approach allows the fully turbulent law to be easily modified and extended to take into account other effects such as pressure gradients or variable properties. This formula also guarantees the correct asymptotic behavior for large and small values of y^+ and reasonable representation of velocity profiles in the cases where y^+ falls inside the wall buffer region ($3 < y^+ < 10$).

The enhanced wall functions were developed by smoothly blending an enhanced turbulent wall law with the laminar wall law. The enhanced turbulent law-of-the-wall for compressible flow with heat transfer and pressure gradients has been derived by combining the approaches of White and Cristoph [378] and Huang et al. [134]:

$$\frac{du^+_{\text{turb}}}{dy^+} = \frac{1}{\kappa y^+}\left[S'(1-\beta u^+ - \gamma (u^+)^2)\right]^{1/2} \tag{4.12-33}$$

where

$$S' = \begin{cases} 1+\alpha y^+ & \text{for } y^+ < y^+_s \\ 1+\alpha y^+_s & \text{for } y^+ \geq y^+_s \end{cases} \tag{4.12-34}$$

and

$$\alpha \equiv \frac{\nu_w}{\tau_w u^*}\frac{dp}{dx} = \frac{\mu}{\rho^2 (u^*)^3}\frac{dp}{dx} \tag{4.12-35}$$

$$\beta \equiv \frac{\sigma_t q_w u^*}{c_p \tau_w T_w} = \frac{\sigma_t q_w}{\rho c_p u^* T_w} \tag{4.12-36}$$

$$\gamma \equiv \frac{\sigma_t (u^*)^2}{2 c_p T_w} \tag{4.12-37}$$

where y^+_s is the location at which the log-law slope is fixed. By default, $y^+_s = 60$. The coefficient α in Equation 4.12-33 represents the influences of pressure gradients while the coefficients β and γ represent the thermal effects. Equation 4.12-33 is an ordinary differential equation and ANSYS FLUENT will provide an appropriate analytical solution. If α, β, and γ all equal 0, an analytical solution would lead to the classical turbulent logarithmic law-of-the-wall.

The laminar law-of-the-wall is determined from the following expression:

$$\frac{du^+_{\text{lam}}}{dy^+} = 1 + \alpha y^+ \tag{4.12-38}$$

Note that the above expression only includes effects of pressure gradients through α, while the effects of variable properties due to heat transfer and compressibility on the laminar wall law are neglected. These effects are neglected because they are thought to be of minor importance when they occur close to the wall. Integration of Equation 4.12-38 results in

$$u^+_{\rm lam} = y^+ \left(1 + \frac{\alpha}{2} y^+\right) \tag{4.12-39}$$

Enhanced thermal wall functions follow the same approach developed for the profile of u^+. The unified wall thermal formulation blends the laminar and logarithmic profiles according to the method of Kader [155]:

$$T^+ \equiv \frac{(T_w - T_P)\,\rho c_p u_T}{\dot{q}} \quad = \quad e^{\Gamma} T^+_{\rm lam} + e^{\frac{1}{\Gamma}} T^+_{\rm turb} \tag{4.12-40}$$

where the notation for T_P and $\dot{q}$ is the same as for standard thermal wall functions (see Equation 4.12-6). Furthermore, the blending factor Γ is defined as

$$\Gamma \quad = \quad -\frac{a(\mathrm{Pr}\, y^+)^4}{1 + b\mathrm{Pr}^3\, y^+} \tag{4.12-41}$$

where Pr is the molecular Prandtl number, and the coefficients a and b are defined as in Equation 4.12-31.

Apart from the formulation for T^+ in Equation 4.12-40, the enhanced thermal wall functions follow the same logic as for standard thermal wall functions (see Section 4.12.2: Energy), resulting in the following definition for turbulent and laminar thermal wall functions:

$$T^+_{\rm lam} = \mathrm{Pr}\left(u^+_{\rm lam} + \frac{\rho u_*}{2\dot{q}} u^2\right) \tag{4.12-42}$$

$$T^+_{\rm turb} = \mathrm{Pr_t}\left\{u^+_{\rm turb} + P + \frac{\rho u_*}{2\dot{q}}\left[u^2 - \left(\frac{\mathrm{Pr}}{\mathrm{Pr_t}} - 1\right)(u^+_c)^2 (u_*)^2\right]\right\} \tag{4.12-43}$$

where the quantity u^+_c is the value of u^+ at the fictitious "crossover" between the laminar and turbulent region. The function P is defined in the same way as for the standard wall functions.

A similar procedure is also used for species wall functions when the enhanced wall treatment is used. In this case, the Prandtl numbers in Equations 4.12-42 and 4.12-43 are replaced by adequate Schmidt numbers. See Section 4.12.2: Species for details about the species wall functions.

The boundary conditions for the turbulence kinetic energy are similar to the ones used with the standard wall functions (Equation 4.12-10). However, the production of turbulence kinetic energy, G_k, is computed using the velocity gradients that are consistent with the enhanced law-of-the-wall (Equations 4.12-30 and 4.12-32), ensuring a formulation that is valid throughout the near-wall region.

i The enhanced wall treatment is available with the following turbulence closures:

- k-ϵ models
- Realizable k-ϵ based DES model
- Reynolds Stress Transport models

The enhanced wall functions are available with the following turbulence models:

- Spalart-Allmaras model
- k-ω models
- k-ω based DES model
- Large Eddy Simulation

However, the enhanced wall functions are not available with Spalart-Allmaras model.

4.12.5 User-Defined Wall Functions

This option is only available when one of the k-ϵ model is enabled. Selecting the User-Defined Wall Functions under Near-wall Treatment allows you to hook a Law-of-the-Wall UDF. For more information about user-defined wall functions,
see Section 2.3.28: `DEFINE_WALL_FUNCTIONS` in the separate UDF Manual.

i User-defined wall functions are available with the following turbulence closures:

- k-ϵ models

4.12.6 LES Near-Wall Treatment

When the mesh is fine enough to resolve the laminar sublayer, the wall shear stress is obtained from the laminar stress-strain relationship:

$$\frac{\overline{u}}{u_\tau} = \frac{\rho u_\tau y}{\mu} \tag{4.12-44}$$

If the mesh is too coarse to resolve the laminar sublayer, it is assumed that the centroid of the wall-adjacent cell falls within the logarithmic region of the boundary layer, and the law-of-the-wall is employed:

$$\frac{\overline{u}}{u_\tau} = \frac{1}{\kappa} \ln E \left(\frac{\rho u_\tau y}{\mu} \right) \tag{4.12-45}$$

where κ is the von Kármán constant and $E = 9.793$. If the mesh is such that the first near-wall point is within the buffer region, then two above laws are blended in accordance with the Equation 4.12-30.

For the LES simulations in ANSYS FLUENT, there is an alternative near-wall approach based on the work of Werner and Wengle [375], who proposed an analytical integration of the power-law near-wall velocity distribution resulting in the following expressions for the wall shear stress:

$$|\tau_w| = \begin{cases} \frac{2\mu|u_p|}{\Delta z} & \text{for } |u_p| \le \frac{\mu}{2\rho\Delta z} A^{\frac{2}{1-B}} \\ \rho \left[\frac{1-B}{2} A^{\frac{1+B}{1-B}} \left(\frac{\mu}{\rho \Delta z} \right)^{1+B} + \frac{1+B}{A} \left(\frac{\mu}{\rho \Delta z} \right)^{B} |u_p| \right]^{\frac{2}{1+B}} & \text{for } |u_p| > \frac{\mu}{2\rho\Delta z} A^{\frac{2}{1-B}} \end{cases} \tag{4.12-46}$$

where u_p is the wall-parallel velocity, $A = 8.3, B = 1/7$ are the constants, and Δz is the near-wall control volume length scale.

The Werner-Wengle wall functions can be enabled using the `define/models/viscous/near-wall-treatment/werner-wengle-wall-fn?` text command.

Chapter 5. Heat Transfer

This chapter describes the theory behind heat transfer in ANSYS FLUENT. Information is provided in the following sections:

- Section 5.1: Introduction
- Section 5.2: Modeling Conductive and Convective Heat Transfer
- Section 5.3: Modeling Radiation

For more information about using heat transfer in ANSYS FLUENT, see Chapter 13: Modeling Heat Transfer in the separate User's Guide.

5.1 Introduction

The flow of thermal energy from matter occupying one region in space to matter occupying a different region in space is known as heat transfer. Heat transfer can occur by three main methods: conduction, convection, and radiation. Physical models involving conduction and/or convection only are the simplest (Section 5.2: Modeling Conductive and Convective Heat Transfer), while buoyancy-driven flow or natural convection (Section 5.2.2: Natural Convection and Buoyancy-Driven Flows Theory), and radiation models (Section 5.3: Modeling Radiation) are more complex. Depending on your problem, ANSYS FLUENT will solve a variation of the energy equation that takes into account the heat transfer methods you have specified. ANSYS FLUENT is also able to predict heat transfer in periodically repeating geometries (Section 13.4: Modeling Periodic Heat Transfer in the separate User's Guide), thus greatly reducing the required computational effort in certain cases.

For more information about using heat transfer models in ANSYS FLUENT, see Section 13.2: Modeling Conductive and Convective Heat Transfer, Section 13.3: Modeling Radiation, and Section 13.4: Modeling Periodic Heat Transfer in the separate User's Guide.

5.2 Modeling Conductive and Convective Heat Transfer

ANSYS FLUENT allows you to include heat transfer within the fluid and/or solid regions in your model. Problems ranging from thermal mixing within a fluid to conduction in composite solids can thus be handled by ANSYS FLUENT.

When your ANSYS FLUENT model includes heat transfer you will need to activate the relevant physical models, supply thermal boundary conditions, and input material properties that govern heat transfer and/or vary with temperature as part of the setup. For information about setting up and using heat transfer in your ANSYS FLUENT model, see Chapter 13: Modeling Heat Transfer in the separate User's Guide. Information about heat transfer theory is presented in the following subsections.

- Section 5.2.1: Heat Transfer Theory
- Section 5.2.2: Natural Convection and Buoyancy-Driven Flows Theory

5.2.1 Heat Transfer Theory

The Energy Equation

ANSYS FLUENT solves the energy equation in the following form:

$$\frac{\partial}{\partial t}(\rho E) + \nabla \cdot (\vec{v}(\rho E + p)) = \nabla \cdot \left(k_{\rm eff} \nabla T - \sum_j h_j \vec{J}_j + (\overline{\overline{\tau}}_{\rm eff} \cdot \vec{v}) \right) + S_h \tag{5.2-1}$$

where $k_{\rm eff}$ is the effective conductivity ($k + k_t$, where k_t is the turbulent thermal conductivity, defined according to the turbulence model being used), and $\vec{J}_j$ is the diffusion flux of species j. The first three terms on the right-hand side of Equation 5.2-1 represent energy transfer due to conduction, species diffusion, and viscous dissipation, respectively. S_h includes the heat of chemical reaction, and any other volumetric heat sources you have defined.

In Equation 5.2-1,

$$E = h - \frac{p}{\rho} + \frac{v^2}{2} \tag{5.2-2}$$

where sensible enthalpy h is defined for ideal gases as

$$h = \sum_j Y_j h_j \tag{5.2-3}$$

and for incompressible flows as

$$h = \sum_j Y_j h_j + \frac{p}{\rho} \tag{5.2-4}$$

In Equations 5.2-3 and 5.2-4, Y_j is the mass fraction of species j and

$$h_j = \int_{T_{\rm ref}}^{T} c_{p,j}\, dT \tag{5.2-5}$$

where $T_{\rm ref}$ is 298.15 K.

The Energy Equation for the Non-Premixed Combustion Model

When the non-adiabatic non-premixed combustion model is enabled, ANSYS FLUENT solves the total enthalpy form of the energy equation:

$$\frac{\partial}{\partial t}(\rho H) + \nabla \cdot (\rho \vec{v} H) = \nabla \cdot \left(\frac{k_t}{c_p} \nabla H \right) + S_h \tag{5.2-6}$$

Under the assumption that the Lewis number (Le) = 1, the conduction and species diffusion terms combine to give the first term on the right-hand side of the above equation while the contribution from viscous dissipation appears in the non-conservative form as the second term. The total enthalpy H is defined as

$$H = \sum_j Y_j H_j \tag{5.2-7}$$

where Y_j is the mass fraction of species j and

$$H_j = \int_{T_{{\rm ref},j}}^{T} c_{p,j} dT + h_j^0(T_{{\rm ref},j}) \tag{5.2-8}$$

$h_j^0(T_{{\rm ref},j})$ is the formation enthalpy of species j at the reference temperature $T_{{\rm ref},j}$.

Inclusion of Pressure Work and Kinetic Energy Terms

Equation 5.2-1 includes pressure work and kinetic energy terms which are often negligible in incompressible flows. For this reason, the pressure-based solver by default does not include the pressure work or kinetic energy when you are solving incompressible flow. If you wish to include these terms, use the `define/models/energy?` text command. When asked to `include pressure work in energy equation?` and `include kinetic energy in energy equation?`, respond by entering `yes` in the console window.

Pressure work and kinetic energy are *always* accounted for when you are modeling compressible flow or using the density-based solver.

Inclusion of the Viscous Dissipation Terms

Equations 5.2-1 and 5.2-6 include viscous dissipation terms, which describe the thermal energy created by viscous shear in the flow.

When the pressure-based solver is used, ANSYS FLUENT's default form of the energy equation does not include them (because viscous heating is often negligible). Viscous heating will be important when the Brinkman number, Br, approaches or exceeds unity, where

$$\mathrm{Br} = \frac{\mu U_e^2}{k \Delta T} \tag{5.2-9}$$

and ΔT represents the temperature difference in the system.

When your problem requires inclusion of the viscous dissipation terms and you are using the pressure-based solver, you should activate the terms using the Viscous Heating option in the **Viscous Model** dialog box. Compressible flows typically have Br ≥ 1. Note, however, that when the pressure-based solver is used, ANSYS FLUENT does not automatically activate the viscous dissipation if you have defined a compressible flow model.

When the density-based solver is used, the viscous dissipation terms are *always* included when the energy equation is solved.

Inclusion of the Species Diffusion Term

Equations 5.2-1 and 5.2-6 both include the effect of enthalpy transport due to species diffusion.

When the pressure-based solver is used, the term

$$\nabla \cdot \left(\sum_j h_j \vec{J}_j \right)$$

is included in Equation 5.2-1 by default. If you do not want to include it, you can disable the Diffusion Energy Source option in the **Species Model** dialog box.

When the non-adiabatic non-premixed combustion model is being used, this term does not explicitly appear in the energy equation, because it is included in the first term on the right-hand side of Equation 5.2-6.

When the density-based solver is used, this term is *always* included in the energy equation.

Energy Sources Due to Reaction

Sources of energy, S_h, in Equation 5.2-1 include the source of energy due to chemical reaction:

$$S_{h,\text{rxn}} = -\sum_j \frac{h_j^0}{M_j} \mathcal{R}_j \tag{5.2-10}$$

where h_j^0 is the enthalpy of formation of species j and $\mathcal{R}_j$ is the volumetric rate of creation of species j.

In the energy equation used for non-adiabatic non-premixed combustion (Equation 5.2-6), the heat of formation is included in the definition of enthalpy (see Equation 5.2-7), so reaction sources of energy are not included in S_h.

Energy Sources Due To Radiation

When one of the radiation models is being used, S_h in Equation 5.2-1 or 5.2-6 also includes radiation source terms. See Section 5.3: Modeling Radiation for details.

Interphase Energy Sources

It should be noted that the energy sources, S_h, also include heat transfer between the continuous and the discrete phase. This is discussed further in Section 15.12.1: Coupling Between the Discrete and Continuous Phases.

Energy Equation in Solid Regions

In solid regions, the energy transport equation used by ANSYS FLUENT has the following form:

$$\frac{\partial}{\partial t}(\rho h) + \nabla \cdot (\vec{v} \rho h) = \nabla \cdot (k \nabla T) + S_h \tag{5.2-11}$$

where ρ = density
h = sensible enthalpy, $\int_{T_\text{ref}}^{T} c_p dT$
k = conductivity
T = temperature
S_h = volumetric heat source

The second term on the left-hand side of Equation 5.2-11 represents convective energy transfer due to rotational or translational motion of the solids. The velocity field $\vec{v}$ is computed from the motion specified for the solid zone (see Section 7.2.2: Solid Conditions in the separate User's Guide). The terms on the right-hand side of Equation 5.2-11 are the heat flux due to conduction and volumetric heat sources within the solid, respectively.

Anisotropic Conductivity in Solids

When you use the pressure-based solver, ANSYS FLUENT allows you to specify anisotropic conductivity for solid materials. The conduction term for an anisotropic solid has the form

$$\nabla \cdot (k_{ij} \nabla T) \tag{5.2-12}$$

where k_{ij} is the conductivity matrix. See Section 8.5.5: Anisotropic Thermal Conductivity for Solids in the separate User's Guide for details on specifying anisotropic conductivity for solid materials.

Diffusion at Inlets

The net transport of energy at inlets consists of both the convection and diffusion components. The convection component is fixed by the inlet temperature specified by you. The diffusion component, however, depends on the gradient of the computed temperature field. Thus the diffusion component (and therefore the net inlet transport) is not specified a priori.

In some cases, you may wish to specify the net inlet transport of energy rather than the inlet temperature. If you are using the pressure-based solver, you can do this by disabling inlet energy diffusion. By default, ANSYS FLUENT includes the diffusion flux of energy at inlets. To turn off inlet diffusion, use the `define/models/energy?` text command and respond `no` when asked to `Include diffusion at inlets?`

Inlet diffusion cannot be turned off if you are using the density-based solver.

5.2.2 Natural Convection and Buoyancy-Driven Flows Theory

When heat is added to a fluid and the fluid density varies with temperature, a flow can be induced due to the force of gravity acting on the density variations. Such buoyancy-driven flows are termed natural-convection (or mixed-convection) flows and can be modeled by ANSYS FLUENT.

The importance of buoyancy forces in a mixed convection flow can be measured by the ratio of the Grashof and Reynolds numbers:

$$\frac{\mathrm{Gr}}{\mathrm{Re}^2} = \frac{g\beta\Delta T L}{v^2} \tag{5.2-13}$$

When this number approaches or exceeds unity, you should expect strong buoyancy contributions to the flow. Conversely, if it is very small, buoyancy forces may be ignored in your simulation. In pure natural convection, the strength of the buoyancy-induced flow is measured by the Rayleigh number:

$$\mathrm{Ra} = \frac{g\beta\Delta T L^3\rho}{\mu\alpha} \tag{5.2-14}$$

where β is the thermal expansion coefficient:

$$\beta = -\frac{1}{\rho}\left(\frac{\partial\rho}{\partial T}\right)_p \tag{5.2-15}$$

and α is the thermal diffusivity:

$$\alpha = \frac{k}{\rho c_p} \tag{5.2-16}$$

Rayleigh numbers less than 10^8 indicate a buoyancy-induced laminar flow, with transition to turbulence occurring over the range of $10^8 < \mathrm{Ra} < 10^{10}$.

5.3 Modeling Radiation

Information about radiation modeling theory is presented in the following sections:

- Section 5.3.1: Overview and Limitations
- Section 5.3.2: Radiative Transfer Equation
- Section 5.3.3: P-1 Radiation Model Theory
- Section 5.3.4: Rosseland Radiation Model Theory
- Section 5.3.5: Discrete Transfer Radiation Model (DTRM) Theory
- Section 5.3.6: Discrete Ordinates (DO) Radiation Model Theory
- Section 5.3.7: Surface-to-Surface (S2S) Radiation Model Theory
- Section 5.3.8: Radiation in Combusting Flows
- Section 5.3.9: Choosing a Radiation Model

For information about setting up and using radiation models in ANSYS FLUENT, see Section 13.3: Modeling Radiation in the separate User's Guide.

5.3.1 Overview and Limitations

ANSYS FLUENT provides five radiation models which allow you to include radiation, with or without a participating medium, in your heat transfer simulations.

Heating or cooling of surfaces due to radiation and/or heat sources or sinks due to radiation within the fluid phase can be included in your model using one of the following radiation models.

- Discrete Transfer Radiation Model (DTRM) [47, 312]
- P-1 Radiation Model [52, 315]
- Rosseland Radiation Model [315]
- Surface-to-Surface (S2S) Radiation Model [315]
- Discrete Ordinates (DO) Radiation Model [56, 282]

In addition to these radiation models, ANSYS FLUENT also provides a solar load model that allows you to include the effects of solar radiation in your simulation.

Typical applications well suited for simulation using radiative heat transfer include the following:

- radiative heat transfer from flames
- surface-to-surface radiant heating or cooling
- coupled radiation, convection, and/or conduction heat transfer
- radiation through windows in HVAC applications, and cabin heat transfer analysis in automotive applications
- radiation in glass processing, glass fiber drawing, and ceramic processing

You should include radiative heat transfer in your simulation when the radiant heat flux, $Q_{\rm rad} = \sigma(T_{\rm max}^4 - T_{\rm min}^4)$, is large compared to the heat transfer rate due to convection or conduction. Typically this will occur at high temperatures where the fourth-order dependence of the radiative heat flux on temperature implies that radiation will dominate.

Advantages and Limitations of the DTRM

The primary advantages of the DTRM are threefold: it is a relatively simple model, you can increase the accuracy by increasing the number of rays, and it applies to a wide range of optical thicknesses.

You should be aware of the following limitations when using the DTRM in ANSYS FLUENT:

- DTRM assumes that all surfaces are diffuse. This means that the reflection of incident radiation at the surface is isotropic with respect to the solid angle.
- The effect of scattering is not included.
- The implementation assumes gray radiation.
- Solving a problem with a large number of rays is CPU-intensive.
- DTRM is not compatible with non-conformal interface or sliding meshes.
- DTRM is not compatible with parallel processing.

Advantages and Limitations of the P-1 Model

The P-1 model has several advantages over the DTRM. For the P-1 model, the RTE (Equation 5.3-1) is a diffusion equation, which is easy to solve with little CPU demand. The model includes the effect of scattering. For combustion applications where the optical thickness is large, the P-1 model works reasonably well. In addition, the P-1 model can easily be applied to complicated geometries with curvilinear coordinates.

You should be aware of the following limitations when using the P-1 radiation model:

- P-1 model assumes that all surfaces are diffuse. This means that the reflection of incident radiation at the surface is isotropic with respect to the solid angle.
- The implementation assumes gray radiation.
- There may be a loss of accuracy, depending on the complexity of the geometry, if the optical thickness is small.
- P-1 model tends to over-predict radiative fluxes from localized heat sources or sinks.

Advantages and Limitations of the Rosseland Model

The Rosseland model has two advantages over the P-1 model. Since it does not solve an extra transport equation for the incident radiation (as the P-1 model does), the Rosseland model is faster than the P-1 model and requires less memory.

The Rosseland model can be used only for optically thick media. It is recommended for use when the optical thickness exceeds 3. Note also that the Rosseland model is not available when the density-based solver is being used; it is available with the pressure-based solver, only.

Advantages and Limitations of the DO Model

The DO model spans the entire range of optical thicknesses, and allows you to solve problems ranging from surface-to-surface radiation to participating radiation in combustion problems. It also allows the solution of radiation at semi-transparent walls. Computational cost is moderate for typical angular discretizations, and memory requirements are modest.

The current implementation is restricted to either gray radiation or non-gray radiation using a gray-band model. Solving a problem with a fine angular discretization may be CPU-intensive.

The non-gray implementation in ANSYS FLUENT is intended for use with participating media with a spectral absorption coefficient a_λ that varies in a stepwise fashion across spectral bands, but varies smoothly within the band. Glass, for example, displays banded behavior of this type. The current implementation does not model the behavior of gases such as carbon dioxide or water vapor, which absorb and emit energy at distinct wave numbers [234]. The modeling of non-gray gas radiation is still an evolving field. However, some researchers [98] have used gray-band models to model gas behavior by approximating the absorption coefficients within each band as a constant. The implementation in ANSYS FLUENT can be used in this fashion if desired.

The non-gray implementation in ANSYS FLUENT is compatible with all the models with which the gray implementation of the DO model can be used. Thus, it is possible to include scattering, anisotropy, semi-transparent media, and particulate effects. However, the non-gray implementation assumes a constant absorption coefficient within each wavelength band. The weighted-sum-of-gray-gases model (WSGGM) cannot be used to specify the absorption coefficient in each band. The implementation allows the specification of spectral emissivity at walls. The emissivity is assumed to be constant within each band.

Advantages and Limitations of the S2S Model

The surface-to-surface (S2S) radiation model is good for modeling the enclosure radiative transfer without participating media (e.g., spacecraft heat rejection systems, solar collector systems, radiative space heaters, and automotive underhood cooling systems). In such cases, the methods for participating radiation may not always be efficient. As compared to the DTRM and the DO radiation models, the S2S model has a much faster time per iteration, although the view factor calculation itself is CPU-intensive. This increased time for view factor calculation will be especially pronounced when the emitting/absorbing surfaces are the polygonal faces of polyhedral cells.

You should be aware of the following limitations when using the S2S radiation model:

- S2S model assumes that all surfaces are diffuse.
- The implementation assumes gray radiation.
- The storage and memory requirements increase very rapidly as the number of surface faces increases. This can be minimized by using a cluster of surface faces, although the CPU time is independent of the number of clusters that are used.
- S2S model cannot be used to model participating radiation problems.
- S2S model cannot be used if your model contains periodic boundary conditions.
- S2S model with hemicube/adaptive view factor methods cannot be used if your model contains symmetry boundary conditions.
- S2S model does not support non-conformal interfaces, hanging nodes, or mesh adaption.

5.3.2 Radiative Transfer Equation

The radiative transfer equation (RTE) for an absorbing, emitting, and scattering medium at position $\vec{r}$ in the direction $\vec{s}$ is

$$\frac{dI(\vec{r},\vec{s})}{ds} + (a+\sigma_s)I(\vec{r},\vec{s}) = an^2\frac{\sigma T^4}{\pi} + \frac{\sigma_s}{4\pi}\int_0^{4\pi} I(\vec{r},\vec{s}\,')\;\Phi(\vec{s}\cdot\vec{s}\,')\;d\Omega' \tag{5.3-1}$$

where	$\vec{r}$	=	position vector
	$\vec{s}$	=	direction vector
	$\vec{s}\,'$	=	scattering direction vector
	s	=	path length
	a	=	absorption coefficient
	n	=	refractive index
	σ_s	=	scattering coefficient
	σ	=	Stefan-Boltzmann constant (5.669×10^{-8} W/m^2-K^4)
	I	=	radiation intensity, which depends on position ($\vec{r}$) and direction ($\vec{s}$)
	T	=	local temperature
	Φ	=	phase function
	Ω'	=	solid angle

$(a+\sigma_s)s$ is the optical thickness or opacity of the medium. The refractive index n is important when considering radiation in semi-transparent media. Figure 5.3.1 illustrates the process of radiative heat transfer.

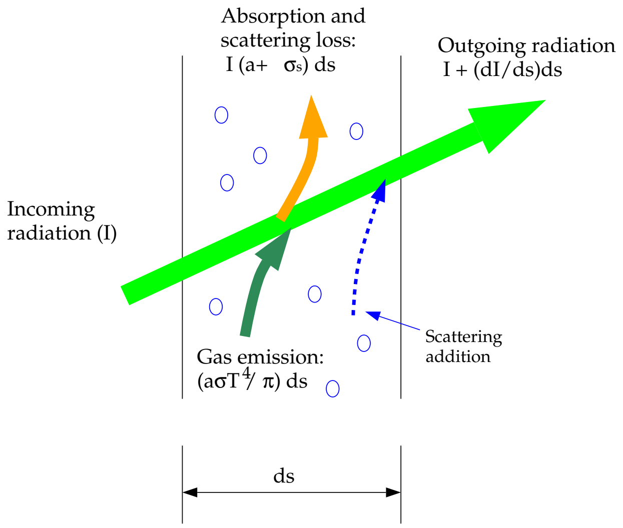


Figure 5.3.1: Radiative Heat Transfer

The DTRM and the P-1, Rosseland, and DO radiation models require the absorption coefficient a as input. a and the scattering coefficient σ_s can be constants, and a can also be a function of local concentrations of H_2O and CO_2, path length, and total pressure. ANSYS FLUENT provides the weighted-sum-of-gray-gases model (WSGGM) for computation of a variable absorption coefficient. See Section 5.3.8: Radiation in Combusting Flows for details. The discrete ordinates implementation can model radiation in semi-transparent media. The refractive index n of the medium must be provided as a part of the calculation for this type of problem. The Rosseland model also requires you to enter a refractive index, or use the default value of 1.

5.3.3 P-1 Radiation Model Theory

The P-1 radiation model is the simplest case of the more general P-N model, which is based on the expansion of the radiation intensity I into an orthogonal series of spherical harmonics [52, 315]. This section provides details about the equations used in the P-1 model. For information about setting up the model, see Section 13.3.1: Steps in Using the Radiation Models in the separate User's Guide.

The P-1 Model Equations

As mentioned above, the P-1 radiation model is the simplest case of the P-N model. If only four terms in the series are used, the following equation is obtained for the radiation flux q_r:

$$q_r = -\frac{1}{3(a+\sigma_s) - C\sigma_s}\nabla G \tag{5.3-2}$$

where a is the absorption coefficient, σ_s is the scattering coefficient, G is the incident radiation, and C is the linear-anisotropic phase function coefficient, described below. After introducing the parameter

$$\Gamma = \frac{1}{(3(a+\sigma_s) - C\sigma_s)} \tag{5.3-3}$$

Equation 5.3-2 simplifies to

$$q_r = -\Gamma\nabla G \tag{5.3-4}$$

The transport equation for G is

$$\nabla\cdot(\Gamma\nabla G) - aG + 4an^2\sigma T^4 = S_G \tag{5.3-5}$$

where n is the refractive index of the medium, σ is the Stefan-Boltzmann constant and S_G is a user-defined radiation source. ANSYS FLUENT solves this equation to determine the local radiation intensity when the P-1 model is active.

Combining Equations 5.3-4 and 5.3-5 yields the following equation:

$$-\nabla \cdot q_r = aG - 4an^2\sigma T^4 \tag{5.3-6}$$

The expression for $-\nabla \cdot q_r$ can be directly substituted into the energy equation to account for heat sources (or sinks) due to radiation.

Anisotropic Scattering

Included in the P-1 radiation model is the capability for modeling anisotropic scattering. ANSYS FLUENT models anisotropic scattering by means of a linear-anisotropic scattering phase function:

$$\Phi(\vec{s}\,' \cdot \vec{s}) = 1 + C\vec{s}\,' \cdot \vec{s} \tag{5.3-7}$$

Here, $\vec{s}$ is the unit vector in the direction of scattering, and $\vec{s}\,'$ is the unit vector in the direction of the incident radiation. C is the linear-anisotropic phase function coefficient, which is a property of the fluid. C ranges from -1 to 1. A positive value indicates that more radiant energy is scattered forward than backward, and a negative value means that more radiant energy is scattered backward than forward. A zero value defines isotropic scattering (i.e., scattering that is equally likely in all directions), which is the default in ANSYS FLUENT. You should modify the default value only if you are certain of the anisotropic scattering behavior of the material in your problem.

Particulate Effects in the P-1 Model

When your ANSYS FLUENT model includes a dispersed second phase of particles, you can include the effect of particles in the P-1 radiation model. Note that when particles are present, ANSYS FLUENT ignores scattering in the gas phase. (That is, Equation 5.3-8 assumes that all scattering is due to particles.)

For a gray, absorbing, emitting, and scattering medium containing absorbing, emitting, and scattering particles, the transport equation for the incident radiation can be written as

$$\nabla \cdot (\Gamma \nabla G) + 4\pi \left(an^2 \frac{\sigma T^4}{\pi} + E_p \right) - (a + a_p)G = 0 \tag{5.3-8}$$

where E_p is the equivalent emission of the particles, a_p is the equivalent absorption coefficient, and n is the refractive index of the medium. These are defined as follows:

$$E_p = \lim_{V \to 0} \sum_{n=1}^{N} \epsilon_{pn} A_{pn} \frac{\sigma T_{pn}^4}{\pi V} \qquad (5.3\text{-}9)$$

and

$$a_p = \lim_{V \to 0} \sum_{n=1}^{N} \epsilon_{pn} \frac{A_{pn}}{V} \qquad (5.3\text{-}10)$$

In Equations 5.3-9 and 5.3-10, ϵ_{pn}, A_{pn}, and T_{pn} are the emissivity, projected area, and temperature of particle n. The summation is over N particles in volume V. These quantities are computed during particle tracking in ANSYS FLUENT.

The projected area A_{pn} of particle n is defined as

$$A_{pn} = \frac{\pi d_{pn}^2}{4} \qquad (5.3\text{-}11)$$

where d_{pn} is the diameter of the nth particle.

The quantity Γ in Equation 5.3-8 is defined as

$$\Gamma = \frac{1}{3(a + a_p + \sigma_p)} \qquad (5.3\text{-}12)$$

where the equivalent particle scattering factor is defined as

$$\sigma_p = \lim_{V \to 0} \sum_{n=1}^{N} (1 - f_{pn})(1 - \epsilon_{pn}) \frac{A_{pn}}{V} \qquad (5.3\text{-}13)$$

and is computed during particle tracking. In Equation 5.3-13, f_{pn} is the scattering factor associated with the nth particle.

Heat sources (sinks) due to particle radiation are included in the energy equation as follows:

$$-\nabla \cdot q_r = -4\pi \left(a n^2 \frac{\sigma T^4}{\pi} + E_p \right) + (a + a_p) G \qquad (5.3\text{-}14)$$

Boundary Condition Treatment for the P-1 Model at Walls

To get the boundary condition for the incident radiation equation, the dot product of the outward normal vector $\vec{n}$ and Equation 5.3-4 is computed:

$$q_r \cdot \vec{n} = -\Gamma \nabla G \cdot \vec{n} \tag{5.3-15}$$

$$q_{r,w} = -\Gamma \frac{\partial G}{\partial n} \tag{5.3-16}$$

Thus the flux of the incident radiation, G, at a wall is $-q_{r,w}$. The wall radiative heat flux is computed using the following boundary condition:

$$I_w(\vec{r}, \vec{s}) = f_w(\vec{r}, \vec{s}) \tag{5.3-17}$$

$$f_w(\vec{r}, \vec{s}) = \epsilon_w \frac{n^2 \sigma T_w^4}{\pi} + \rho_w I(\vec{r}, -\vec{s}) \tag{5.3-18}$$

where ρ_w is the wall reflectivity. The Marshak boundary condition is then used to eliminate the angular dependence [262]:

$$\int_0^{2\pi} I_w(\vec{r}, \vec{s})\, \vec{n} \cdot \vec{s}\, d\Omega = \int_0^{2\pi} f_w(\vec{r}, \vec{s})\, \vec{n} \cdot \vec{s}\, d\Omega \tag{5.3-19}$$

Substituting Equations 5.3-17 and 5.3-18 into Equation 5.3-19 and performing the integrations yields

$$q_{r,w} = -\frac{4\pi \epsilon_w \frac{n^2 \sigma T_w^4}{\pi} - (1 - \rho_w) G_w}{2(1 + \rho_w)} \tag{5.3-20}$$

If it is assumed that the walls are diffuse gray surfaces, then $\rho_w = 1 - \epsilon_w$, and Equation 5.3-20 becomes

$$q_{r,w} = -\frac{\epsilon_w}{2\,(2 - \epsilon_w)} \left(4 n^2 \sigma T_w^4 - G_w\right) \tag{5.3-21}$$

Equation 5.3-21 is used to compute $q_{r,w}$ for the energy equation and for the incident radiation equation boundary conditions.

Boundary Condition Treatment for the P-1 Model at Flow Inlets and Exits

The net radiative heat flux at flow inlets and outlets is computed in the same manner as at walls, as described above. ANSYS FLUENT assumes that the emissivity of all flow inlets and outlets is 1.0 (black body absorption) unless you choose to redefine this boundary treatment.

ANSYS FLUENT includes an option that allows you to use different temperatures for radiation and convection at inlets and outlets. This can be useful when the temperature outside the inlet or outlet differs considerably from the temperature in the enclosure. See Section 13.3.6: Defining Boundary Conditions for Radiation in the separate User's Guide for details.

5.3.4 Rosseland Radiation Model Theory

The Rosseland or diffusion approximation for radiation is valid when the medium is optically thick ($(a + \sigma_s)L \gg 1$), and is recommended for use in problems where the optical thickness is greater than 3. It can be derived from the P-1 model equations, with some approximations. This section provides details about the equations used in the Rosseland model. For information about setting up the model, see Section 13.3.1: Steps in Using the Radiation Models in the separate User's Guide.

The Rosseland Model Equations

As with the P-1 model, the radiative heat flux vector in a gray medium can be approximated by Equation 5.3-4:

$$q_r = -\Gamma \nabla G \tag{5.3-22}$$

where Γ is given by Equation 5.3-3.

The Rosseland radiation model differs from the P-1 model in that the Rosseland model assumes that the intensity is the black-body intensity at the gas temperature. (The P-1 model actually calculates a transport equation for G.) Thus $G = 4\sigma n^2 T^4$, where n is the refractive index. Substituting this value for G into Equation 5.3-22 yields

$$q_r = -16\sigma\Gamma n^2 T^3 \nabla T \tag{5.3-23}$$

Since the radiative heat flux has the same form as the Fourier conduction law, it is possible to write

$$\begin{aligned} q &= q_c + q_r && (5.3\text{-}24) \\ &= -(k + k_r)\nabla T && (5.3\text{-}25) \\ k_r &= 16\sigma\Gamma n^2 T^3 && (5.3\text{-}26) \end{aligned}$$

where k is the thermal conductivity and k_r is the radiative conductivity. Equation 5.3-24 is used in the energy equation to compute the temperature field.

Anisotropic Scattering

The Rosseland model allows for anisotropic scattering, using the same phase function (Equation 5.3-7) described for the P-1 model in Section 5.3.3: Anisotropic Scattering.

Boundary Condition Treatment for the Rosseland Model at Walls

Since the diffusion approximation is not valid near walls, it is necessary to use a temperature slip boundary condition. The radiative heat flux at the wall boundary, $q_{r,w}$, is defined using the slip coefficient ψ:

$$q_{r,w} = -\frac{\sigma\left(T_w^4 - T_g^4\right)}{\psi} \tag{5.3-27}$$

where T_w is the wall temperature, T_g is the temperature of the gas at the wall, and the slip coefficient ψ is approximated by a curve fit to the plot given in [315]:

$$\psi = \begin{cases} 1/2 & N_w < 0.01 \\ \frac{2x^3+3x^2-12x+7}{54} & 0.01 \leq N_w \leq 10 \\ 0 & N_w > 10 \end{cases} \tag{5.3-28}$$

where N_w is the conduction to radiation parameter at the wall:

$$N_w = \frac{k(a + \sigma_s)}{4\sigma T_w^3} \tag{5.3-29}$$

and $x = \log_{10} N_w$.

Boundary Condition Treatment for the Rosseland Model at Flow Inlets and Exits

No special treatment is required at flow inlets and outlets for the Rosseland model. The radiative heat flux at these boundaries can be determined using Equation 5.3-24.

5.3.5 Discrete Transfer Radiation Model (DTRM) Theory

The main assumption of the DTRM is that the radiation leaving the surface element in a certain range of solid angles can be approximated by a single ray. This section provides details about the equations used in the DTRM. For information about setting up the model, see Section 13.3.2: Setting Up the DTRM in the separate User's Guide.

The DTRM Equations

The equation for the change of radiant intensity, dI, along a path, ds, can be written as

$$\frac{dI}{ds} + aI = \frac{a\sigma T^4}{\pi} \tag{5.3-30}$$

where a = gas absorption coefficient
I = intensity
T = gas local temperature
σ = Stefan-Boltzmann constant (5.669×10^{-8} W/m^2-K^4)

Here, the refractive index is assumed to be unity. The DTRM integrates Equation 5.3-30 along a series of rays emanating from boundary faces. If a is constant along the ray, then $I(s)$ can be estimated as

$$I(s) = \frac{\sigma T^4}{\pi}(1 - e^{-as}) + I_0 e^{-as} \tag{5.3-31}$$

where I_0 is the radiant intensity at the start of the incremental path, which is determined by the appropriate boundary condition (see the description of boundary conditions, below). The energy source in the fluid due to radiation is then computed by summing the change in intensity along the path of each ray that is traced through the fluid control volume.

The "ray tracing" technique used in the DTRM can provide a prediction of radiative heat transfer between surfaces without explicit view factor calculations. The accuracy of the model is limited mainly by the number of rays traced and the computational mesh.

Ray Tracing

The ray paths are calculated and stored prior to the fluid flow calculation. At each radiating face, rays are fired at discrete values of the polar and azimuthal angles (see Figure 5.3.2). To cover the radiating hemisphere, θ is varied from 0 to $\frac{\pi}{2}$ and ϕ from 0 to 2π. Each ray is then traced to determine the control volumes it intercepts as well as its length within each control volume. This information is then stored in the radiation file, which must be read in before the fluid flow calculations begin.

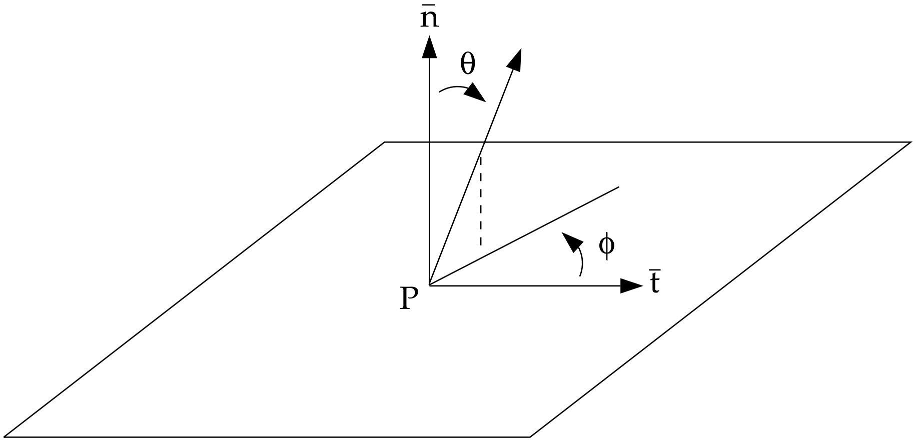


Figure 5.3.2: Angles θ and ϕ Defining the Hemispherical Solid Angle About a Point P

Clustering

DTRM is computationally very expensive when there are too many surfaces to trace rays from and too many volumes crossed by the rays. To reduce the computational time, the number of radiating surfaces and absorbing cells is reduced by clustering surfaces and cells into surface and volume "clusters". The volume clusters are formed by starting from a cell and simply adding its neighbors and their neighbors until a specified number of cells per volume cluster is collected. Similarly, surface clusters are made by starting from a face and adding its neighbors and their neighbors until a specified number of faces per surface cluster is collected.

The incident radiation flux, $q_{\rm in}$, and the volume sources are calculated for the surface and volume clusters respectively. These values are then distributed to the faces and cells in the clusters to calculate the wall and cell temperatures. Since the radiation source terms are highly non-linear (proportional to the fourth power of temperature), care must be taken to calculate the average temperatures of surface and volume clusters and distribute the flux and source terms appropriately among the faces and cells forming the clusters.

The surface and volume cluster temperatures are obtained by area and volume averaging as shown in the following equations:

$$T_{sc} = \left(\frac{\sum_f A_f T_f^4}{\sum A_f} \right)^{1/4} \tag{5.3-32}$$

$$T_{vc} = \left(\frac{\sum_c V_c T_c^4}{\sum V_c} \right)^{1/4} \tag{5.3-33}$$

where T_{sc} and T_{vc} are the temperatures of the surface and volume clusters respectively, A_f and T_f are the area and temperature of face f, and V_c and T_c are the volume and temperature of cell c. The summations are carried over all faces of a surface cluster and all cells of a volume cluster.

Boundary Condition Treatment for the DTRM at Walls

The radiation intensity approaching a point on a wall surface is integrated to yield the incident radiative heat flux, $q_{\rm in}$, as

$$q_{\rm in} = \int_{\vec{s}\cdot\vec{n}>0} I_{\rm in} \vec{s} \cdot \vec{n} d\Omega \tag{5.3-34}$$

where Ω is the hemispherical solid angle, $I_{\rm in}$ is the intensity of the incoming ray, $\vec{s}$ is the ray direction vector, and $\vec{n}$ is the normal pointing out of the domain. The net radiative heat flux from the surface, $q_{\rm out}$, is then computed as a sum of the reflected portion of $q_{\rm in}$ and the emissive power of the surface:

$$q_{\rm out} = (1 - \epsilon_w) q_{\rm in} + \epsilon_w \sigma T_w^4 \tag{5.3-35}$$

where T_w is the surface temperature of the point P on the surface and ϵ_w is the wall emissivity which you input as a boundary condition. ANSYS FLUENT incorporates the radiative heat flux (Equation 5.3-35) in the prediction of the wall surface temperature. Equation 5.3-35 also provides the surface boundary condition for the radiation intensity I_0 of a ray emanating from the point P, as

$$I_0 = \frac{q_{\rm out}}{\pi} \tag{5.3-36}$$

Boundary Condition Treatment for the DTRM at Flow Inlets and Exits

The net radiative heat flux at flow inlets and outlets is computed in the same manner as at walls, as described above. ANSYS FLUENT assumes that the emissivity of all flow inlets and outlets is 1.0 (black body absorption) unless you choose to redefine this boundary treatment.

ANSYS FLUENT includes an option that allows you to use different temperatures for radiation and convection at inlets and outlets. This can be useful when the temperature outside the inlet or outlet differs considerably from the temperature in the enclosure. See Section 13.3.6: Defining Boundary Conditions for Radiation in the separate User's Guide for details.

5.3.6 Discrete Ordinates (DO) Radiation Model Theory

The discrete ordinates (DO) radiation model solves the radiative transfer equation (RTE) for a finite number of discrete solid angles, each associated with a vector direction $\vec{s}$ fixed in the global Cartesian system (x, y, z). The fineness of the angular discretization is controlled by you, analogous to choosing the number of rays for the DTRM. Unlike the DTRM, however, the DO model does not perform ray tracing. Instead, the DO model transforms Equation 5.3-1 into a transport equation for radiation intensity in the spatial coordinates (x, y, z). The DO model solves for as many transport equations as there are directions $\vec{s}$. The solution method is identical to that used for the fluid flow and energy equations.

Two implementations of the DO model are available in ANSYS FLUENT: uncoupled and (energy) coupled. The uncoupled implementation is sequential in nature and uses a conservative variant of the DO model called the finite-volume scheme [56, 282], and its extension to unstructured meshes [242]. In the uncoupled case, the equations for the energy and radiation intensities are solved one by one, assuming prevailing values for other variables.

Alternatively, in the coupled ordinates method (or COMET) [220], the discrete energy and intensity equations at each cell are solved simultaneously, assuming that spatial neighbors are known. The advantages of using the coupled approach is that it speeds up applications involving high optical thicknesses and/or high scattering coefficients. Such applications slow down convergence drastically when the sequential approach is used. For information about setting up the model, see Section 13.3.4: Setting Up the DO Model in the separate User's Guide.

The DO Model Equations

The DO model considers the radiative transfer equation (RTE) in the direction $\vec{s}$ as a field equation. Thus, Equation 5.3-1 is written as

$$\nabla \cdot (I(\vec{r},\vec{s})\vec{s}) + (a+\sigma_s)I(\vec{r},\vec{s}) = an^2\frac{\sigma T^4}{\pi} + \frac{\sigma_s}{4\pi}\int_0^{4\pi} I(\vec{r},\vec{s}\,') \; \Phi(\vec{s}\cdot\vec{s}\,') \; d\Omega' \qquad (5.3\text{-}37)$$

ANSYS FLUENT also allows the modeling of non-gray radiation using a gray-band model. The RTE for the spectral intensity $I_\lambda(\vec{r},\vec{s})$ can be written as

$$\nabla \cdot (I_\lambda(\vec{r},\vec{s})\vec{s}) + (a_\lambda+\sigma_s)I_\lambda(\vec{r},\vec{s}) = a_\lambda n^2 I_{b\lambda} + \frac{\sigma_s}{4\pi}\int_0^{4\pi} I_\lambda(\vec{r},\vec{s}\,') \; \Phi(\vec{s}\cdot\vec{s}\,') \; d\Omega' \qquad (5.3\text{-}38)$$

Here λ is the wavelength, a_λ is the spectral absorption coefficient, and $I_{b\lambda}$ is the black body intensity given by the Planck function. The scattering coefficient, the scattering phase function, and the refractive index n are assumed independent of wavelength.

The non-gray DO implementation divides the radiation spectrum into N wavelength bands, which need not be contiguous or equal in extent. The wavelength intervals are supplied by you, and correspond to values in vacuum ($n = 1$). The RTE is integrated over each wavelength interval, resulting in transport equations for the quantity $I_\lambda \Delta\lambda$, the radiant energy contained in the wavelength band $\Delta\lambda$. The behavior in each band is assumed gray. The black body emission in the wavelength band per unit solid angle is written as

$$[F(0 \rightarrow n\lambda_2 T) - F(0 \rightarrow n\lambda_1 T)]n^2\frac{\sigma T^4}{\pi} \qquad (5.3\text{-}39)$$

where $F(0 \rightarrow n\lambda T)$ is the fraction of radiant energy emitted by a black body [234] in the wavelength interval from 0 to λ at temperature T in a medium of refractive index n. λ_2 and λ_1 are the wavelength boundaries of the band.

The total intensity $I(\vec{r},\vec{s})$ in each direction $\vec{s}$ at position $\vec{r}$ is computed using

$$I(\vec{r},\vec{s}) = \sum_k I_{\lambda_k}(\vec{r},\vec{s})\Delta\lambda_k \qquad (5.3\text{-}40)$$

where the summation is over the wavelength bands.

Boundary conditions for the non-gray DO model are applied on a band basis. The treatment within a band is the same as that for the gray DO model.

Energy Coupling and the DO Model

The coupling between energy and radiation intensities at a cell (which is also known as COMET) [220] accelerates the convergence of the finite volume scheme for radiative heat transfer. This method results in significant improvement in the convergence for applications involving optical thicknesses greater than 10. This is typically encountered in glass-melting applications. This feature is advantageous when scattering is significant, resulting in strong coupling between directional radiation intensities. This DO model implementation is utilized in ANSYS FLUENT by enabling the DO/Energy Coupling option for the DO model in the Radiation Model dialog box. The discrete energy equations for the coupled method are presented below.

The energy equation when integrated over a control volume i, yields the discrete energy equation:

$$\sum_{j=1}^{N} \mu_{ij}^{T} T_j - \beta_i^T T_i - \alpha_i^T \sum_{k=1}^{L} I_i^k \omega_k - S_i^T S_i^h \tag{5.3-41}$$

where
$\alpha_i^T = \kappa \Delta V_i$
$\beta_i^T = 16 \kappa \sigma T_i^{*3} \Delta V_i$
$S_i^T = 12 \kappa \sigma T_i^{*4} \Delta V_i$
κ = absorption coefficient
ΔV = control volume

The coefficient μ_{ij}^T and the source term S_i^h are due to the discretization of the convection and diffusion terms as well as the non-radiative source terms.

Combining the discretized form of Equation 5.3-37 and the discretized energy equation, Equation 5.3-41, yields [220]:

$$\vec{P}_i \vec{q}_i + \vec{r}_i = 0 \tag{5.3-42}$$

where

$$\vec{q_i} = \begin{bmatrix} I_i^1 \\ I_i^2 \\ : \\ : \\ I_i^L \\ T_i \end{bmatrix} \tag{5.3-43}$$

$$\vec{P_i} = \begin{bmatrix} M_{ii}^1 + \eta_i^{11} + \gamma_i^1\omega_1 & \eta_i^{12} + \gamma_i^1\omega_2 & ...\beta_i^1 \\ \eta_i^{21} + \gamma_i^2\omega_1 & M_{ii}^2 + \eta_i^{22} + \gamma_i^2\omega_2 & ...\beta_i^2 \\ : & & \\ : & & \\ -\alpha_i^T\omega_1 & -\alpha_i^T\omega_2 & ...M_{ii}^T \end{bmatrix} \tag{5.3-44}$$

$$\vec{r_i} = \begin{bmatrix} \sum_{j=1,i\neq j}^N \mu_{ij}^l I_j^l - S_i^1 - S_i^B \\ : \\ : \\ \sum_{j=1,i\neq j}^N \mu_{ij}^T T_j + S_i^T + S_i^h \end{bmatrix} \tag{5.3-45}$$

Limitations of DO/Energy Coupling

There are some instances when using DO/Energy coupling is not recommended or is incompatible with certain models:

- DO/Energy coupling is not recommended for cases with weak coupling between energy and directional radiation intensities. This may result in slower convergence of the coupled approach compared to the sequential approach.
- DO/Energy coupling is not compatible with the shell conduction model.
- DO/Energy coupling is not available when solving enthalpy equations instead of temperature equations. Typical cases would involve combustion modeling.

To find out how to apply DO/Energy coupling, refer to Section 13.3.4: Setting Up the DO Model in the separate User's Guide.

Angular Discretization and Pixelation

Each octant of the angular space 4π at any spatial location is discretized into $N_\theta \times N_\phi$ solid angles of extent ω_i, called control angles. The angles θ and ϕ are the polar and azimuthal angles respectively, and are measured with respect to the global Cartesian system (x, y, z) as shown in Figure 5.3.3. The θ and ϕ extents of the control angle, $\Delta\theta$ and $\Delta\phi$, are constant. In two-dimensional calculations, only four octants are solved due to symmetry, making a total of $4N_\theta N_\phi$ directions in all. In three-dimensional calculations, a total of $8N_\theta N_\phi$ directions are solved. In the case of the non-gray model, $4N_\theta N_\phi$ or $8N_\theta N_\phi$ equations are solved for each band.

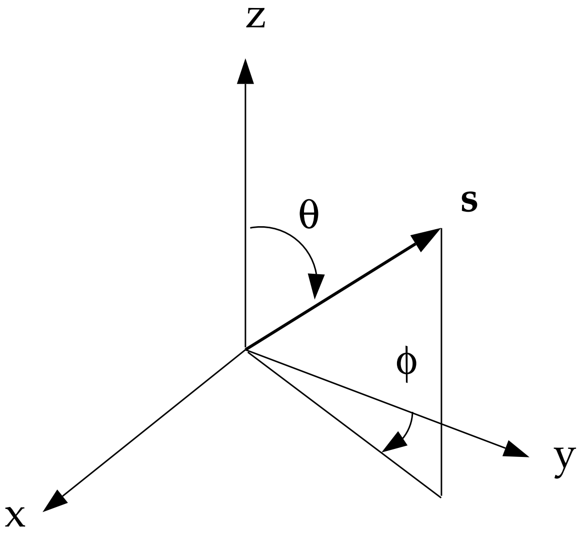


Figure 5.3.3: Angular Coordinate System

When Cartesian meshes are used, it is possible to align the global angular discretization with the control volume face, as shown in Figure 5.3.4. For generalized unstructured meshes, however, control volume faces do not in general align with the global angular discretization, as shown in Figure 5.3.5, leading to the problem of control angle overhang [242].

Essentially, control angles can straddle the control volume faces, so that they are partially incoming and partially outgoing to the face. Figure 5.3.6 shows a 3D example of a face with control angle overhang.

The control volume face cuts the sphere representing the angular space at an arbitrary angle. The line of intersection is a great circle. Control angle overhang may also occur as a result of reflection and refraction. It is important in these cases to correctly account for the overhanging fraction. This is done through the use of pixelation [242].

Each overhanging control angle is divided into $N_{\theta_p} \times N_{\phi_p}$ pixels, as shown in Figure 5.3.7.

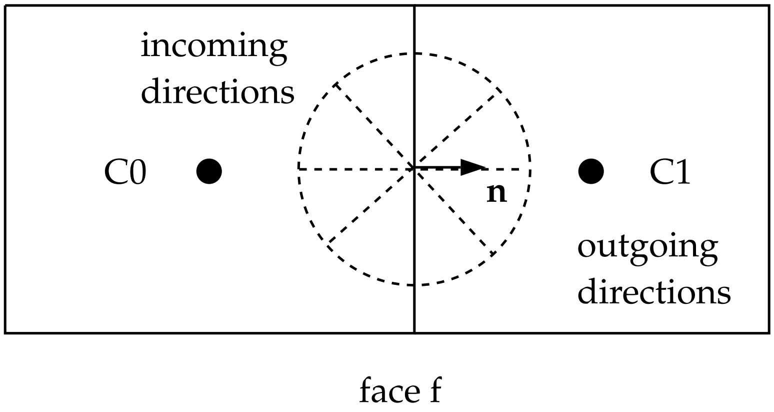


Figure 5.3.4: Face with No Control Angle Overhang

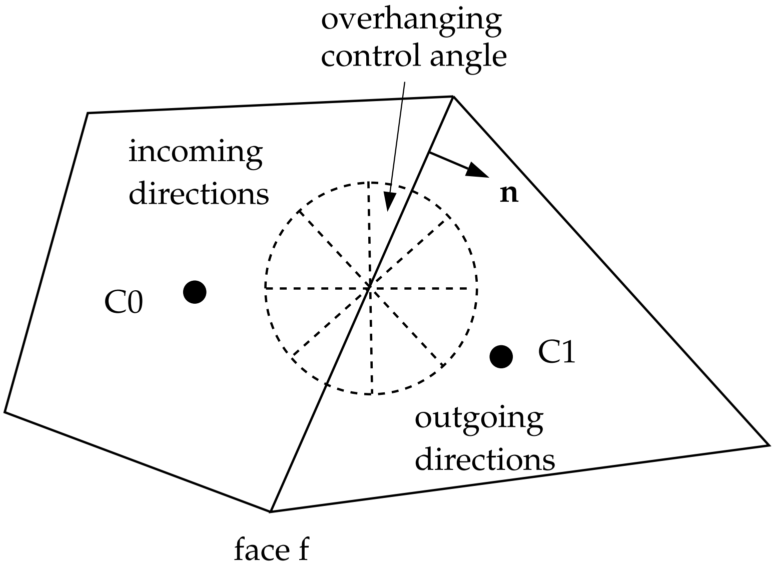


Figure 5.3.5: Face with Control Angle Overhang

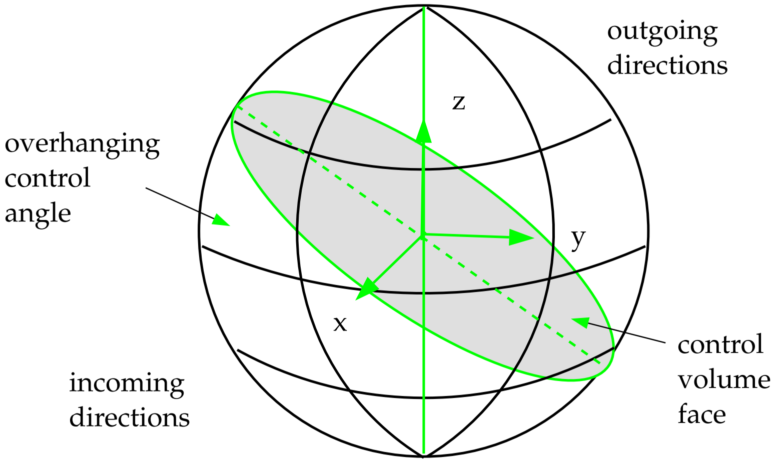


Figure 5.3.6: Face with Control Angle Overhang (3D)

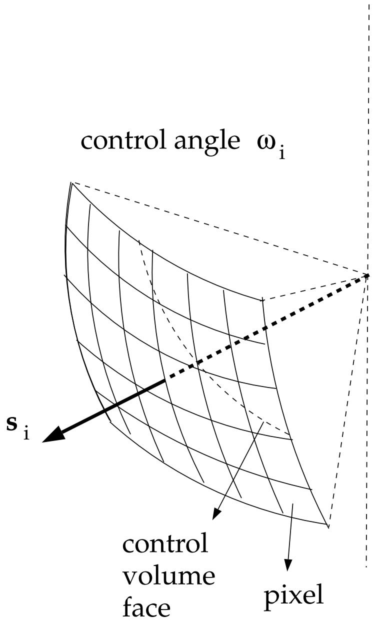


Figure 5.3.7: Pixelation of Control Angle

The energy contained in each pixel is then treated as incoming or outgoing to the face. The influence of overhang can thus be accounted for within the pixel resolution. ANSYS FLUENT allows you to choose the pixel resolution. For problems involving gray-diffuse radiation, the default pixelation of 1 × 1 is usually sufficient. For problems involving symmetry, periodic, specular, or semi-transparent boundaries, a pixelation of 3 × 3 is recommended. You should be aware, however, that increasing the pixelation adds to the cost of computation.

Anisotropic Scattering

The DO implementation in ANSYS FLUENT admits a variety of scattering phase functions. You can choose an isotropic phase function, a linear anisotropic phase function, a Delta-Eddington phase function, or a user-defined phase function. The linear anisotropic phase function is described in Equation 5.3-7. The Delta-Eddington function takes the following form:

$$\Phi(\vec{s}\cdot\vec{s}\,') = 2f\delta(\vec{s}\cdot\vec{s}\,') + (1-f)(1+C\vec{s}\cdot\vec{s}\,') \tag{5.3-46}$$

Here, f is the forward-scattering factor and $\delta(\vec{s}\cdot\vec{s}\,')$ is the Dirac delta function. The f term essentially cancels a fraction f of the out-scattering; thus, for $f=1$, the Delta-Eddington phase function will cause the intensity to behave as if there is no scattering at all. C is the asymmetry factor. When the Delta-Eddington phase function is used, you will specify values for f and C.

When a user-defined function is used to specify the scattering phase function, ANSYS FLUENT assumes the phase function to be of the form

$$\Phi(\vec{s}\cdot\vec{s}\,') = 2f\delta(\vec{s}\cdot\vec{s}\,') + (1-f)\Phi^*(\vec{s}\cdot\vec{s}\,') \tag{5.3-47}$$

The user-defined function will specify Φ^* and the forward-scattering factor f.

The scattering phase functions available for gray radiation can also be used for non-gray radiation. However, the scattered energy is restricted to stay within the band.

Particulate Effects in the DO Model

The DO model allows you to include the effect of a discrete second phase of particulates on radiation. In this case, ANSYS FLUENT will neglect all other sources of scattering in the gas phase.

The contribution of the particulate phase appears in the RTE as:

$$\nabla\cdot(I\vec{s}) + (a + a_p + \sigma_p)I(\vec{r},\vec{s}) = an^2\frac{\sigma T^4}{\pi} + E_p + \frac{\sigma_p}{4\pi}\int_0^{4\pi} I(\vec{r},\vec{s}\,')\ \Phi(\vec{s}\cdot\vec{s}\,')\ d\Omega' \tag{5.3-48}$$

where a_p is the equivalent absorption coefficient due to the presence of particulates, and is given by Equation 5.3-10. The equivalent emission E_p is given by Equation 5.3-9. The equivalent particle scattering factor σ_p, defined in Equation 5.3-13, is used in the scattering terms.

For non-gray radiation, absorption, emission, and scattering due to the particulate phase are included in each wavelength band for the radiation calculation. Particulate emission and absorption terms are also included in the energy equation.

Boundary and Cell Zone Condition Treatment at Opaque Walls

The discrete ordinates radiation model allows the specification of opaque walls that are interior to a domain (with adjacent fluid or solid zones on both sides of the wall), or external to the domain (with an adjacent fluid or solid zone on one side, only). Opaque walls are treated as gray if gray radiation is being computed, or non-gray if the non-gray DO model is being used.

Figure 5.3.8 shows a schematic of radiation on an opaque wall in ANSYS FLUENT.

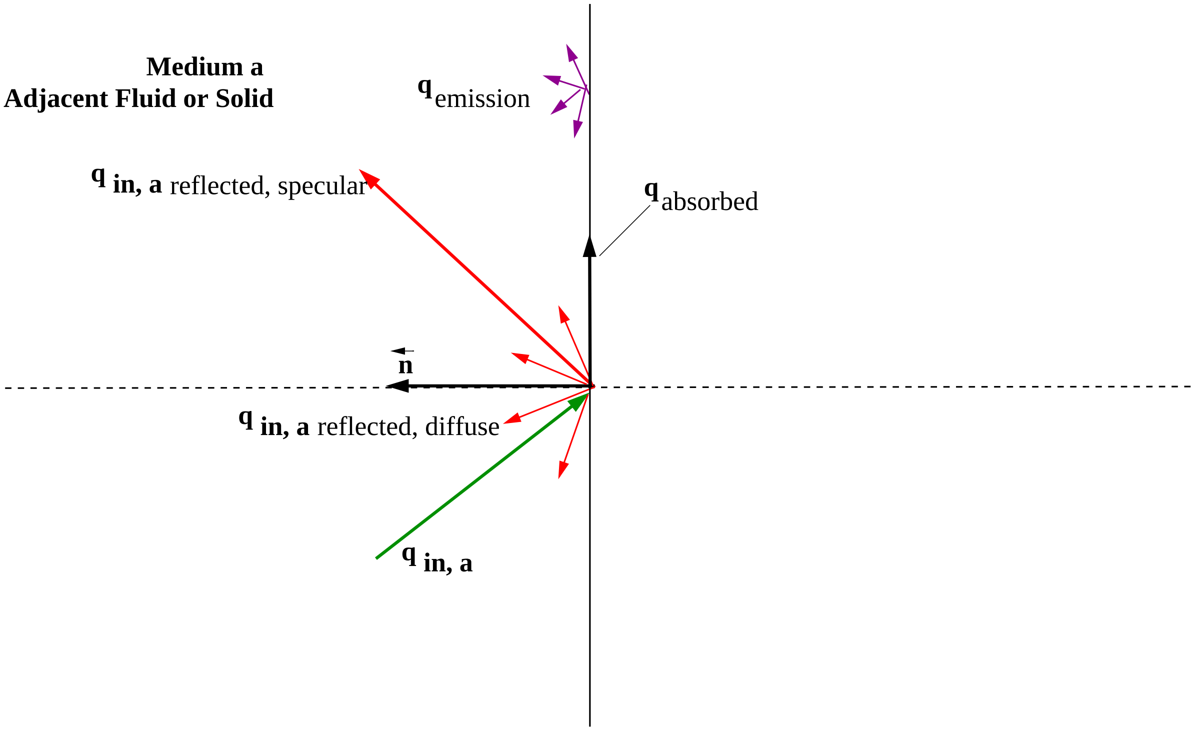


Figure 5.3.8: DO Radiation on Opaque Wall

The diagram in Figure 5.3.8 shows incident radiation $q_{\text{in,a}}$ on side a of an opaque wall. Some of the radiant energy is reflected diffusely and specularly, depending on the diffuse fraction f_d for side a of the wall that you specify as a boundary condition.

Some of the incident radiation is absorbed at the surface of the wall and some radiation is emitted from the wall surface as shown in Figure 5.3.8. The amount of incident radiation absorbed at the wall surface and the amount emitted back depends on the emissivity of that surface and the diffuse fraction. For non-gray DO models, you must specify internal emissivity for each wavelength band. Radiation is not transmitted through an opaque wall.

Radiant incident energy that impacts an opaque wall can be reflected back to the surrounding medium and absorbed by the wall. The radiation that is reflected can be diffusely reflected and/or specularly reflected, depending on the diffuse fraction f_d. If $q_{\rm in}$ is the amount of radiative energy incident on the opaque wall, then the following general quantities are computed by ANSYS FLUENT for opaque walls:

- emission from the wall surface = $n^2 \epsilon_w \sigma T_w^4$
- diffusely reflected energy = $f_d(1 - \epsilon_w) q_{\rm in}$
- specularly reflected energy = $(1 - f_d)(1 - \epsilon_w) q_{\rm in}$
- absorption at the wall surface = $\epsilon_w q_{\rm in}$

where f_d is the diffuse fraction, n is the refractive index of the adjacent medium, ϵ_w is the wall emissivity, σ is Boltzmann's Constant, and T_w is the wall temperature.

Note that although ANSYS FLUENT uses emissivity in its computation of radiation quantities, it is not available for postprocessing. Absorption at the wall surface assumes that the absorptivity is equal to the emissivity. For a purely diffused wall, f_d is equal to 1 and there is no specularly reflected energy. Similarly, for a purely specular wall, f_d is equal to 0 and there is no diffusely reflected energy. A diffuse fraction between 0 and 1 will result in partially diffuse and partially reflected energy.

Gray Diffuse Walls

For gray diffuse radiation, the incident radiative heat flux, $q_{\rm in}$, at the wall is

$$q_{\rm in} = \int_{\vec{s}\cdot\vec{n}>0} I_{\rm in} \vec{s} \cdot \vec{n} d\Omega \tag{5.3-49}$$

The net radiative flux leaving the surface is given by

$$q_{\rm out} = (1 - \epsilon_w) q_{\rm in} + n^2 \epsilon_w \sigma T_w^4 \tag{5.3-50}$$

where n is the refractive index of the medium next to the wall, ϵ_w is the wall emissivity, σ is Boltzmann's Constant, and T_w is the wall temperature. This equation is also valid for specular radiation with emissivity = 0.

The boundary intensity for all outgoing directions $\vec{s}$ at the wall is given by

$$I_0 = \frac{q_{\rm out}}{\pi} \tag{5.3-51}$$

Non-Gray Diffuse Walls

There is a special set of equations that apply uniquely to non-gray diffuse opaque walls. These equations assume that the absorptivity is equal to the emissivity for the wall surface. For non-gray diffuse radiation, the incident radiative heat flux $q_{{\rm in},\lambda}$ in the band $\Delta\lambda$ at the wall is

$$q_{{\rm in},\lambda} = \Delta\lambda \int_{\vec{s}\cdot\vec{n}>0} I_{{\rm in},\lambda}\vec{s}\cdot\vec{n} d\Omega \tag{5.3-52}$$

The net radiative flux leaving the surface in the band $\Delta\lambda$ is given by

$$q_{{\rm out},\lambda} = (1-\epsilon_{w\lambda})q_{{\rm in},\lambda} + \epsilon_{w\lambda}[F(0\rightarrow n\lambda_2 T_w) - F(0\rightarrow n\lambda_1 T_w)]n^2\sigma T_w^4 \tag{5.3-53}$$

where $\epsilon_{w\lambda}$ is the wall emissivity in the band. $F(n,\lambda,T)$ provides the Planck distribution function. This defines the emittance for each radiation band as a function of the temperature of the source surface. The boundary intensity for all outgoing directions $\vec{s}$ in the band $\Delta\lambda$ at the wall is given by

$$I_{0\lambda} = \frac{q_{{\rm out},\lambda}}{\pi\Delta\lambda} \tag{5.3-54}$$

Cell Zone and Boundary Condition Treatment at Semi-Transparent Walls

ANSYS FLUENT allows the specification of interior and exterior semi-transparent walls for the DO model. In the case of interior semi-transparent walls, incident radiation can pass through the wall and be transmitted to the adjacent medium (and possibly refracted), it can be reflected back into the surrounding medium, and absorbed through the wall thickness. Transmission and reflection can be diffuse and/or specular. You specify the diffuse fraction for all transmitted and reflected radiation; the rest is treated specularly. For exterior semi-transparent walls, there are two possible sources of radiation on the boundary wall: an irradiation beam from outside the computational domain and incident radiation from cells in adjacent fluid or solid zones.

For non-gray radiation, semi-transparent wall boundary conditions are applied on a per-band basis. The radiant energy within a band is transmitted, reflected, and refracted as in the gray case; there is no transmission, reflection, or refraction of radiant energy from one band to another.

By default the DO equations are solved in all fluid zones, but not in any solid zones. Therefore, if you have an adjacent solid zone for your thin wall, you will need to specify the solid zone as participating in radiation in the Solid dialog box as part of the boundary condition setup.

i If you are interested in the detailed temperature distribution inside your semi-transparent media, then you will need to model a semi-transparent wall as a solid zone with adjacent fluid zone(s), and treat the solid as a semi-transparent medium. This is discussed in a subsequent section.

Semi-Transparent Interior Walls

Figure 5.3.9 shows a schematic of an interior (two-sided) wall that is treated as semi-transparent in ANSYS FLUENT and has zero thickness. Incident radiant energy depicted by $q_{\text{in,a}}$ can pass through the semi-transparent wall if and only if the contiguous fluid or solid cell zones participate in radiation, thereby allowing the radiation to be coupled. Radiation coupling is set when a wall is specified as semi-transparent. Note that by default, radiation is *not* coupled and you will need to explicitly specify radiation coupling on the interior wall by changing the boundary condition type to semi-transparent in the Wall dialog box (under the Radiation tab).

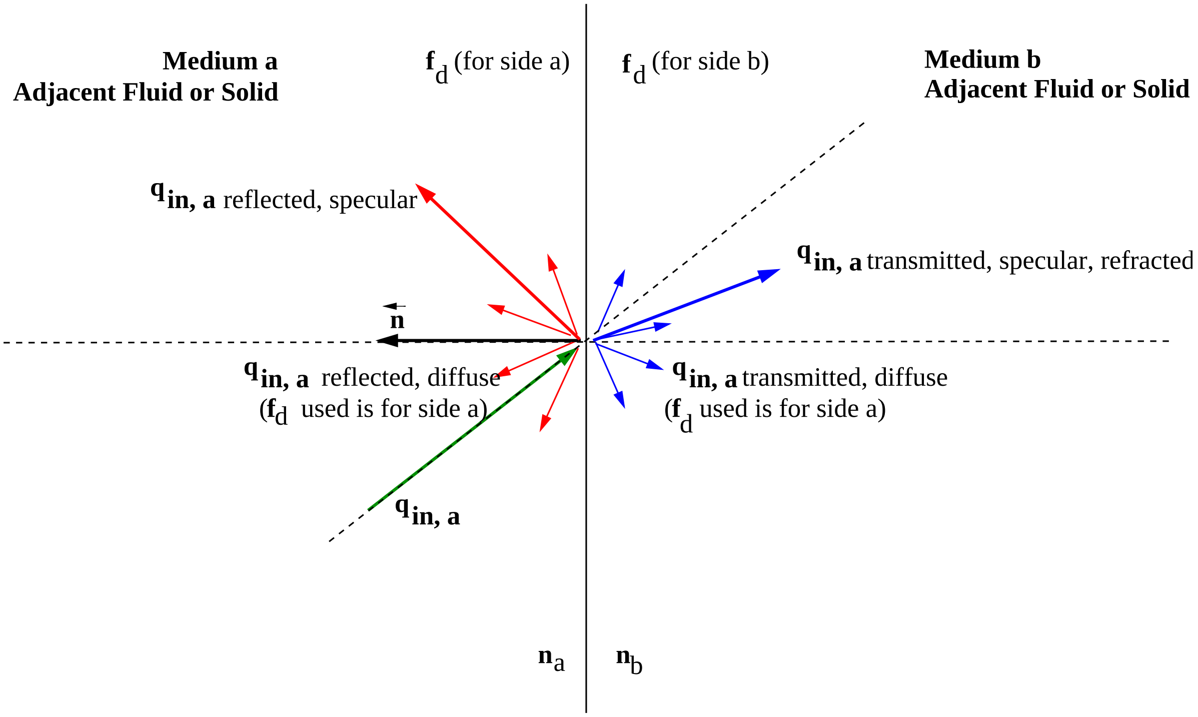


Figure 5.3.9: DO Radiation on Interior Semi-Transparent Wall

Incident radiant energy that is transmitted through a semi-transparent wall can be transmitted specularly and diffusely. Radiation can also be reflected at the interior wall back to the surrounding medium if the refractive index n_a for the fluid zone that represents medium a is different than the refractive index n_b for medium b. Reflected radiation can be reflected specularly and diffusely. The fraction of diffuse versus specular radiation that is transmitted and reflected depends on the diffuse fraction for the wall. The special cases of purely diffuse and purely specular transmission and reflection on semi-transparent walls is presented in the following sections.

If the semi-transparent wall has thickness, then the thickness and the absorption coefficient determine the absorptivity of the 'thin' wall. If either the wall thickness or absorption coefficient is set to 0, then the wall has no absorptivity. Although incident radiation can be absorbed in a semi-transparent wall that has thickness, note that by default the absorbed radiation flux does not affect the energy equation except where shell conduction is used; this can result in an energy imbalance and possibly an unexpected temperature field. The exception to this is when shell conduction is used (available in 3D only) in which case there is full correspondence between energy and radiation. If the wall is expected to have significant absorption/emission then it may be better to model the thickness explicitly with solid cells where practical. ANSYS FLUENT does not include emission from the surface of semi-transparent walls (i.e. due to defined internal emissivity) except for the case when a specified temperature boundary condition is defined.

Specular Semi-Transparent Walls

Consider the special case for a semi-transparent wall, when the diffuse fraction f_d is equal to 0 and all of the transmitted and reflected radiant energy at the semi-transparent wall is purely specular.

Figure 5.3.10 shows a ray traveling from a semi-transparent medium a with refractive index n_a to a semi-transparent medium b with a refractive index n_b in the direction $\vec{s}$. Surface a of the interface is the side that faces medium a; similarly, surface b faces medium b. The interface normal $\vec{n}$ is assumed to point into side a. We distinguish between the intensity $I_a(\vec{s})$, the intensity in the direction $\vec{s}$ on side a of the interface, and the corresponding quantity on the side b, $I_b(\vec{s})$.

A part of the energy incident on the interface is reflected, and the rest is transmitted. The reflection is specular, so that the direction of reflected radiation is given by

$$\vec{s}_r = \vec{s} - 2\left(\vec{s}\cdot\vec{n}\right)\vec{n} \qquad (5.3\text{-}55)$$

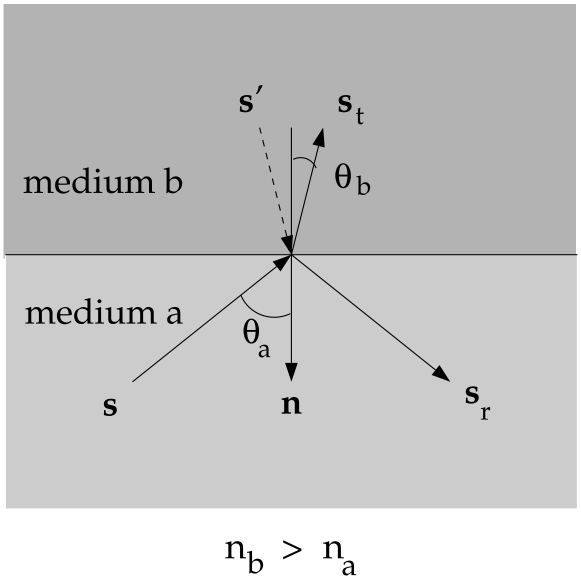


Figure 5.3.10: Reflection and Refraction of Radiation at the Interface Between Two Semi-Transparent Media

The radiation transmitted from medium a to medium b undergoes refraction. The direction of the transmitted energy, $\vec{s}_t$, is given by Snell's law:

$$\sin\theta_b = \frac{n_a}{n_b}\sin\theta_a \tag{5.3-56}$$

where θ_a is the angle of incidence and θ_b is the angle of transmission, as shown in Figure 5.3.10. We also define the direction

$$\vec{s}\,' = \vec{s}_t - 2\left(\vec{s}_t \cdot \vec{n}\right)\vec{n} \tag{5.3-57}$$

shown in Figure 5.3.10.

The interface reflectivity on side a [234]

$$r_a(\vec{s}) = \frac{1}{2}\left(\frac{n_a\cos\theta_b - n_b\cos\theta_a}{n_a\cos\theta_b + n_b\cos\theta_a}\right)^2 + \frac{1}{2}\left(\frac{n_a\cos\theta_a - n_b\cos\theta_b}{n_a\cos\theta_a + n_b\cos\theta_b}\right)^2 \tag{5.3-58}$$

represents the fraction of incident energy transferred from $\vec{s}$ to $\vec{s}_r$.

The boundary intensity $I_{w,a}(\vec{s}_r)$ in the outgoing direction $\vec{s}_r$ on side a of the interface is determined from the reflected component of the incoming radiation and the transmission from side b. Thus

$$I_{w,a}(\vec{s}_r) = r_a(\vec{s})I_{w,a}(\vec{s}) + \tau_b(\vec{s}\,')I_{w,b}(\vec{s}\,') \tag{5.3-59}$$

where $\tau_b(\vec{s}\,')$ is the transmissivity of side b in direction $\vec{s}'$. Similarly, the outgoing intensity in the direction $\vec{s}_t$ on side b of the interface, $I_{w,b}(\vec{s}_t)$, is given by

$$I_{w,b}(\vec{s}_t) = r_b(\vec{s}\,')I_{w,b}(\vec{s}\,') + \tau_a(\vec{s})I_{w,a}(\vec{s}) \tag{5.3-60}$$

For the case $n_a < n_b$, the energy transmitted from medium a to medium b in the incoming solid angle 2π must be refracted into a cone of apex angle θ_c (see Figure 5.3.11) where

$$\theta_c = \sin^{-1}\frac{n_a}{n_b} \tag{5.3-61}$$

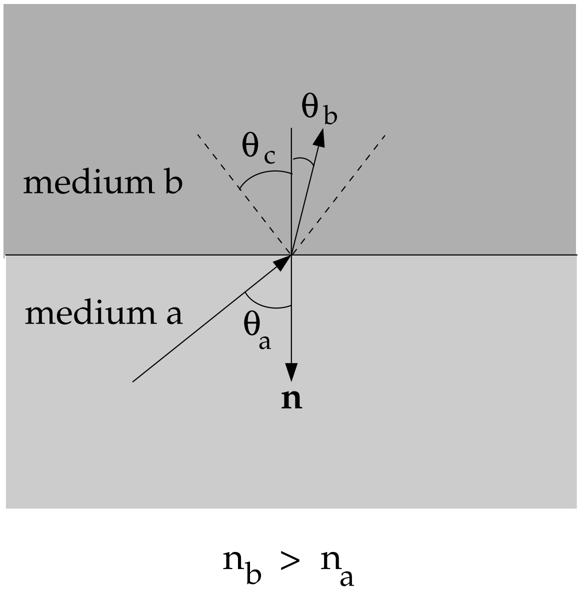


Figure 5.3.11: Critical Angle θ_c

Similarly, the transmitted component of the radiant energy going from medium b to medium a in the cone of apex angle θ_c is refracted into the outgoing solid angle 2π. For incident angles greater than θ_c, total internal reflection occurs and all the incoming energy is reflected specularly back into medium b. The equations presented above can be applied to the general case of interior semi-transparent walls that is shown in Figure 5.3.9.

When medium b is external to the domain as in the case of an external semi-transparent wall (Figure 5.3.12), $I_{w,b}(\vec{s}\ ')$ is given in Equation 5.3-59 as a part of the boundary condition inputs. You supply this incoming irradiation flux in terms of its magnitude, beam direction, and the solid angle over which the radiative flux is to be applied. Note that the refractive index of the external medium is assumed to be 1.

Diffuse Semi-Transparent Walls

Consider the special case for a semi-transparent wall, when the diffuse fraction f_d is equal to 1 and all of the transmitted and reflected radiant energy at the semi-transparent wall is purely diffuse.

In many engineering problems, the semi-transparent interface may be a diffuse reflector. For such a case, the interfacial reflectivity $r(\vec{s})$ is assumed independent of $\vec{s}$, and equal to the hemispherically averaged value r_d. For $n = n_a/n_b > 1$, $r_{d,a}$ and $r_{d,b}$ are given by [316]

$$\begin{aligned} r_{d,a} &= 1 - \frac{(1 - r_{d,b})}{n^2} && (5.3\text{-}62) \\ r_{d,b} &= \frac{1}{2} + \frac{(3n+1)(n-1)}{6(n+1)^2} + \frac{n^2(n^2-1)^2}{(n^2+1)^3} \ln\left(\frac{n-1}{n+1}\right) - \\ & \quad \frac{2n^3(n^2+2n-1)}{(n^2+1)(n^4-1)} + \frac{8n^4(n^4+1)}{(n^2+1)(n^4-1)^2} \ln(n) && (5.3\text{-}63) \end{aligned}$$

The boundary intensity for all outgoing directions on side a of the interface is given by

$$I_{w,a} = \frac{r_{d,a} q_{\text{in},a} + \tau_{d,b} q_{\text{in},b}}{\pi} \qquad (5.3\text{-}64)$$

Similarly for side b,

$$I_{w,b} = \frac{r_{d,b} q_{\text{in},b} + \tau_{d,a} q_{\text{in},a}}{\pi} \qquad (5.3\text{-}65)$$

where

$$\begin{aligned} q_{\text{in},a} &= -\int_{4\pi} I_{w,a} \vec{s} \cdot \vec{n} d\Omega, \quad \vec{s} \cdot \vec{n} < 0 && (5.3\text{-}66) \\ q_{\text{in},b} &= \int_{4\pi} I_{w,b} \vec{s} \cdot \vec{n} d\Omega, \quad \vec{s} \cdot \vec{n} \geq 0 && (5.3\text{-}67) \end{aligned}$$

When medium b is external to the domain as in the case of an external semi-transparent wall (Figure 5.3.12), $q_{\text{in,b}}$ is given as a part of the boundary condition inputs. You supply this incoming irradiation flux in terms of its magnitude, beam direction, and the solid angle over which the radiative flux is to be applied. Note that the refractive index of the external medium is assumed to be 1.

Partially Diffuse Semi-Transparent Walls

When the diffuse fraction f_d that you enter for a semi-transparent wall is between 0 and 1, the wall is partially diffuse and partially specular. In this case, ANSYS FLUENT includes the reflective and transmitted radiative flux contributions from both diffuse and specular components to the defining equations.

Semi-Transparent Exterior Walls

Figure 5.3.12 shows the general case of an irradiation beam $q_{\rm irrad}$ applied to an exterior semi-transparent wall with zero thickness and a non-zero absorption coefficient for the material property. Refer to the previous section for the radiation effects of wall thickness on semi-transparent walls.

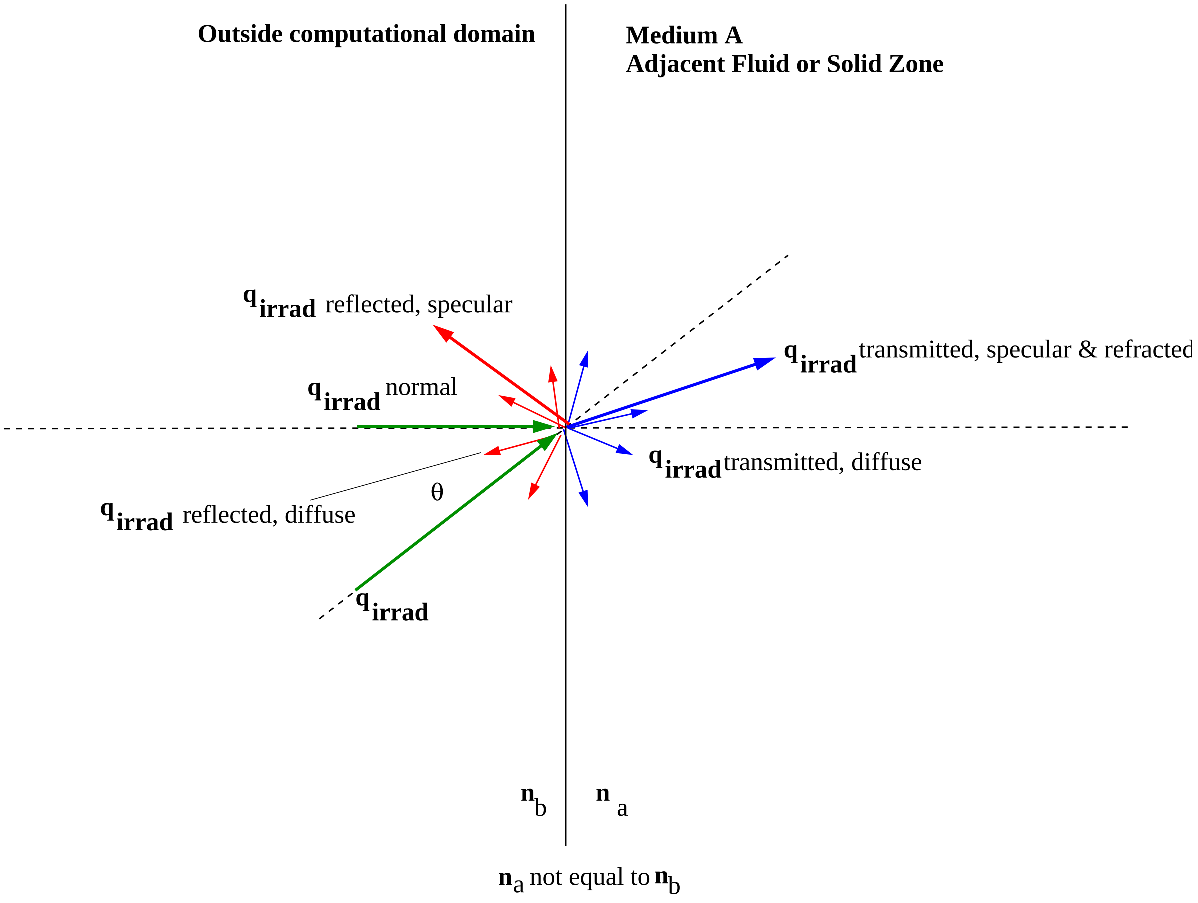

Outside computational domain
Medium A
Adjacent Fluid or Solid Zone
$\mathbf{q}_{\mathbf{irrad}}$ reflected, specular
$\mathbf{q}_{\mathbf{irrad}}$ normal
$\mathbf{q}_{\mathbf{irrad}}$ transmitted, specular & refracted
$\mathbf{q}_{\mathbf{irrad}}$ transmitted, diffuse
$\mathbf{q}_{\mathbf{irrad}}$ reflected, diffuse
θ
$\mathbf{q}_{\mathbf{irrad}}$
$\mathbf{n}_{\rm b}$
$\mathbf{n}_{\rm a}$
$\mathbf{n}_{\rm a}$ not equal to $\mathbf{n}_{\rm b}$

Figure 5.3.12: DO Irradiation on External Semi-Transparent Wall

An irradiation flux passes through the semi-transparent wall from outside the computational domain (Figure 5.3.12) into the adjacent fluid or solid medium a. The transmitted radiation can be refracted (bent) and dispersed specularly and diffusely, depending on the refractive index and the diffuse fraction that you provide as a boundary condition input. Note that there is a reflected component of $q_{\rm irrad}$ when the refractive index of the wall (n_b) is not equal to 1, as shown.

There is an additional flux beyond $q_{\rm irrad}$ that is applied when the Mixed or Radiation wall boundary conditions are selected in the Thermal tab. This external flux at the semi-transparent wall is computed by ANSYS FLUENT as

$$Q_{ext} = \epsilon_{external} \sigma T_{rad}^4 \tag{5.3-68}$$

The fraction of the above energy that will enter into the domain depends on the transmissivity of the semi-transparent wall under consideration. Note that this energy is distributed across the solid angles (i.e., similar treatment as diffuse component.)

Incident radiation can also occur on external semi-transparent walls. Refer to the previous discussion on interior walls for details, since the radiation effects are the same.

The irradiation beam is defined by the magnitude, beam direction, and beam width that you supply. The irradiation magnitude is specified in terms of an incident radiant heat flux (W/m^2). Beam width is specified as the solid angle over which the irradiation is distributed (i.e., the beam θ and ϕ extents). The default beam width in ANSYS FLUENT is $1e - 6$ degrees which is suitable for collimated beam radiation. Beam direction is defined by the vector of the centroid of the solid angle. If you select the feature Apply Direct Irradiation Parallel to the Beam in the Wall boundary condition dialog box, then you supply $q_{\rm irrad}$ for irradiation (Figure 5.3.12) and ANSYS FLUENT computes and uses the surface normal flux $q_{\rm irrad,normal}$ in its radiation calculation. If this feature is not checked, then you must supply the surface normal flux $q_{\rm irrad,normal}$ for irradiation.

Figure 5.3.13 shows a schematic of the beam direction and beam width for the irradiation beam. You provide these inputs (in addition to irradiation magnitude) as part of the boundary conditions for a semi-transparent wall.

The irradiation beam can be refracted in medium a depending on the refractive index that is specified for the particular fluid or solid zone material.

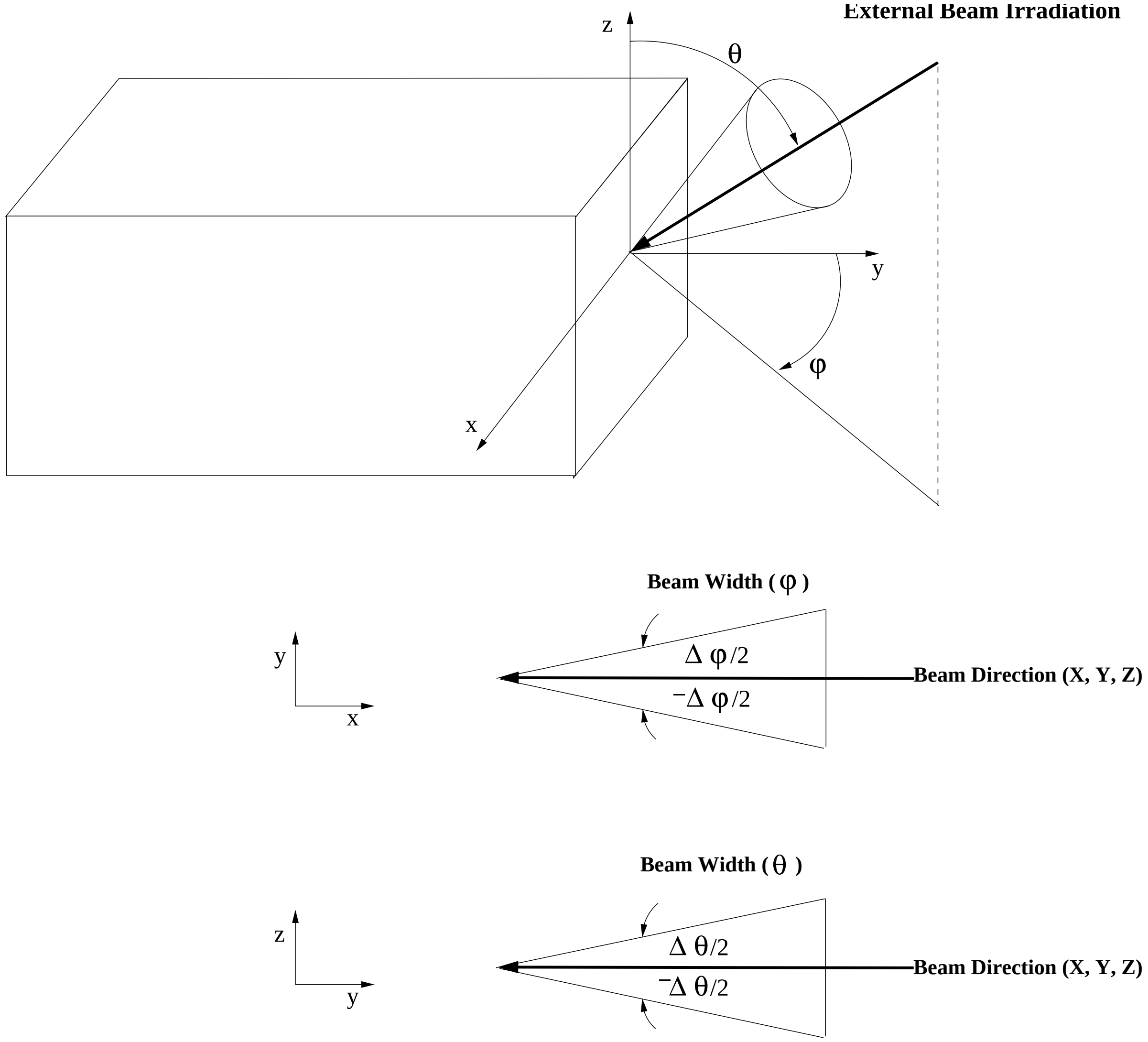


Figure 5.3.13: Beam Width and Direction for External Irradiation Beam

Limitations

Where shell conduction is not active, there is only limited support for absorbing and emitting semi-transparent thin walls. In cases with significant emission or absorption of radiation in a participating solid material, such as the absorption of long wavelength radiation in a glass window, the use of semi-transparent thin walls can result in the prediction of unphysical temperatures in the numerical solution. In a 3-dimensional model this can be overcome by activating the shell conduction option for the respective thin wall. Otherwise, where possible, it is advisable to represent the solid wall thickness explicitly with one or more layers of cells across the wall thickness.

Solid Semi-Transparent Media

The discrete ordinates radiation model allows you to model a solid zone that has adjacent fluid or solid zones on either side as a "semi-transparent" medium. This is done by designating the solid zone to participate in radiation as part of the boundary condition setup. Modeling a solid zone as a semi-transparent medium allows you to obtain a detailed temperature distribution inside the semi-transparent zone since ANSYS FLUENT solves the energy equation on a per-cell basis for the solid and provides you with the thermal results. By default however, the DO equations are solved in fluid zones, but not in any solid zones. Therefore, you will need to specify the solid zone as participating in radiation in the Solid dialog box as part of the boundary condition setup.

Boundary Condition Treatment at Specular Walls and Symmetry Boundaries

At specular walls and symmetry boundaries, the direction of the reflected ray $\vec{s}_r$ corresponding to the incoming direction $\vec{s}$ is given by Equation 5.3-55. Furthermore,

$$I_w(\vec{s}_r) = I_w(\vec{s}) \tag{5.3-69}$$

Boundary Condition Treatment at Periodic Boundaries

When rotationally periodic boundaries are used, it is important to use pixelation in order to ensure that radiant energy is correctly transferred between the periodic and shadow faces. A pixelation between 3 × 3 and 10 × 10 is recommended.

Boundary Condition Treatment at Flow Inlets and Exits

The treatment at flow inlets and exits is described in Section 5.3.5: Boundary Condition Treatment for the DTRM at Flow Inlets and Exits.

5.3.7 Surface-to-Surface (S2S) Radiation Model Theory

The surface-to-surface radiation model can be used to account for the radiation exchange in an enclosure of gray-diffuse surfaces. The energy exchange between two surfaces depends in part on their size, separation distance, and orientation. These parameters are accounted for by a geometric function called a "view factor".

The main assumption of the S2S model is that any absorption, emission, or scattering of radiation can be ignored; therefore, only "surface-to-surface" radiation need be considered for analysis.

For information about setting up the model, see Section 13.3.3: Setting Up the S2S Model in the separate User's Guide.

Gray-Diffuse Radiation

ANSYS FLUENT's S2S radiation model assumes the surfaces to be gray and diffuse. Emissivity and absorptivity of a gray surface are independent of the wavelength. Also, by Kirchoff's law [234], the emissivity equals the absorptivity ($\epsilon = \alpha$). For a diffuse surface, the reflectivity is independent of the outgoing (or incoming) directions.

The gray-diffuse model is what is used in ANSYS FLUENT. Also, as stated earlier, for applications of interest, the exchange of radiative energy between surfaces is virtually unaffected by the medium that separates them. Thus, according to the gray-body model, if a certain amount of radiant energy (E) is incident on a surface, a fraction (ρE) is reflected, a fraction (αE) is absorbed, and a fraction (τE) is transmitted. Since for most applications the surfaces in question are opaque to thermal radiation (in the infrared spectrum), the surfaces can be considered opaque. The transmissivity, therefore, can be neglected. It follows, from the conservation of energy, that $\alpha + \rho = 1$, since $\alpha = \epsilon$ (emissivity), and $\rho = 1 - \epsilon$.

The S2S Model Equations

The energy flux leaving a given surface is composed of directly emitted and reflected energy. The reflected energy flux is dependent on the incident energy flux from the surroundings, which then can be expressed in terms of the energy flux leaving all other surfaces. The energy reflected from surface k is

$$q_{\text{out},k} = \epsilon_k \sigma T_k^4 + \rho_k q_{\text{in},k} \tag{5.3-70}$$

where $q_{\text{out},k}$ is the energy flux leaving the surface, ϵ_k is the emissivity, σ is Boltzmann's constant, and $q_{\text{in},k}$ is the energy flux incident on the surface from the surroundings.

The amount of incident energy upon a surface from another surface is a direct function of the surface-to-surface "view factor," F_{jk}. The view factor F_{jk} is the fraction of energy leaving surface k that is incident on surface j. The incident energy flux $q_{\mathrm{in},k}$ can be expressed in terms of the energy flux leaving all other surfaces as

$$A_k q_{\mathrm{in},k} = \sum_{j=1}^{N} A_j q_{\mathrm{out},j} F_{jk} \tag{5.3-71}$$

where A_k is the area of surface k and F_{jk} is the view factor between surface k and surface j. For N surfaces, using the view factor reciprocity relationship gives

$$A_j F_{jk} = A_k F_{kj} \text{ for } j = 1, 2, 3, \ldots N \tag{5.3-72}$$

so that

$$q_{\mathrm{in},k} = \sum_{j=1}^{N} F_{kj} q_{\mathrm{out},j} \tag{5.3-73}$$

Therefore,

$$q_{\mathrm{out},k} = \epsilon_k \sigma T_k^4 + \rho_k \sum_{j=1}^{N} F_{kj} q_{\mathrm{out},j} \tag{5.3-74}$$

which can be written as

$$J_k = E_k + \rho_k \sum_{j=1}^{N} F_{kj} J_j \tag{5.3-75}$$

where J_k represents the energy that is given off (or radiosity) of surface k, and E_k represents the emissive power of surface k. This represents N equations, which can be recast into matrix form as

$$\boldsymbol{K}\boldsymbol{J} = \boldsymbol{E} \tag{5.3-76}$$

where $\boldsymbol{K}$ is an $N \times N$ matrix, $\boldsymbol{J}$ is the radiosity vector, and $\boldsymbol{E}$ is the emissive power vector.

Equation 5.3-76 is referred to as the radiosity matrix equation. The view factor between two finite surfaces i and j is given by

$$F_{ij} = \frac{1}{A_i} \int_{A_i} \int_{A_j} \frac{\cos\theta_i \cos\theta_j}{\pi r^2} \delta_{ij} dA_i dA_j \tag{5.3-77}$$

where δ_{ij} is determined by the visibility of dA_j to dA_i. $\delta_{ij} = 1$ if dA_j is visible to dA_i and 0 otherwise.

Clustering

The S2S radiation model is computationally very expensive when there is a large number of radiating surfaces. To reduce the computational time as well as the storage requirement, the number of radiating surfaces is reduced by creating surface "clusters". The surface clusters are made by starting from a face and adding its neighbors and their neighbors until a specified number of faces per surface cluster is collected.

An algorithm has been implemented for the creation of surface clusters which is faster and supports non-conformal interfaces, hanging nodes, or mesh adaption. This algorithm is now the default. If you wish to use the old algorithm, you may use the TUI command but adaption and non-conformal interfaces will not be supported.

The radiosity, J, is calculated for the surface clusters. These values are then distributed to the faces in the clusters to calculate the wall temperatures. Since the radiation source terms are highly non-linear (proportional to the fourth power of temperature), care must be taken to calculate the average temperature of the surface clusters and distribute the flux and source terms appropriately among the faces forming the clusters.

The surface cluster temperature is obtained by area averaging as shown in the following equation:

$$T_{sc} = \left(\frac{\sum_f A_f T_f^4}{\sum A_f} \right)^{1/4} \tag{5.3-78}$$

where T_{sc} is the temperature of the surface cluster, and A_f and T_f are the area and temperature of face f. The summation is carried over all faces of a surface cluster.

Smoothing

Smoothing can be performed on the view factor matrix to enforce the reciprocity relationship and conservation.

The reciprocity relationship is represented by

$$A_i F_{ij} = A_j F_{ji} \tag{5.3-79}$$

where A_i is the area of surface i, F_{ij} is the view factor between surfaces i and j, and F_{ji} is the view factor between surfaces j and i.

Once the reciprocity relationship has been enforced, a least-squares smoothing method [175] can be used to ensure that conservation is satisfied, i.e.,

$$\sum F_{ij} = 1.0 \tag{5.3-80}$$

5.3.8 Radiation in Combusting Flows

The Weighted-Sum-of-Gray-Gases Model

The weighted-sum-of-gray-gases model (WSGGM) is a reasonable compromise between the oversimplified gray gas model and a complete model which takes into account particular absorption bands. The basic assumption of the WSGGM is that the total emissivity over the distance s can be presented as

$$\epsilon = \sum_{i=0}^{I} a_{\epsilon,i}(T)(1 - e^{-\kappa_i p s}) \tag{5.3-81}$$

where $a_{\epsilon,i}$ is the emissivity weighting factor for the ith fictitious gray gas, the bracketed quantity is the ith fictitious gray gas emissivity, κ_i is the absorption coefficient of the ith gray gas, p is the sum of the partial pressures of all absorbing gases, and s is the path length. For $a_{\epsilon,i}$ and κ_i ANSYS FLUENT uses values obtained from [60] and [326]. These values depend on gas composition, and $a_{\epsilon,i}$ also depend on temperature. When the total pressure is not equal to 1 atm, scaling rules for κ_i are used (see Equation 5.3-87).

The absorption coefficient for $i = 0$ is assigned a value of zero to account for windows in the spectrum between spectral regions of high absorption ($\sum_{i=1}^{I} a_{\epsilon,i} < 1$) and the weighting factor for $i = 0$ is evaluated from [326]:

$$a_{\epsilon,0} = 1 - \sum_{i=1}^{I} a_{\epsilon,i} \tag{5.3-82}$$

The temperature dependence of $a_{\epsilon,i}$ can be approximated by any function, but the most common approximation is

$$a_{\epsilon,i} = \sum_{j=1}^{J} b_{\epsilon,i,j} T^{j-1} \tag{5.3-83}$$

where $b_{\epsilon,i,j}$ are the emissivity gas temperature polynomial coefficients. The coefficients $b_{\epsilon,i,j}$ and κ_i are estimated by fitting Equation 5.3-81 to the table of total emissivities, obtained experimentally [60, 70, 326].

The absorptivity α of the radiation from the wall can be approximated in a similar way [326], but, to simplify the problem, it is assumed that $\epsilon = \alpha$ [233]. This assumption is justified unless the medium is optically thin and the wall temperature differs considerably from the gas temperature.

Since the coefficients $b_{\epsilon,i,j}$ and κ_i are slowly varying functions of ps and T, they can be assumed constant for a wide range of these parameters. In [326] these constant coefficients are presented for different relative pressures of the CO_2 and H_2O vapor, assuming that the total pressure p_T is 1 atm. The values of the coefficients shown in [326] are valid for $0.001 \leq ps \leq 10.0$ atm-m and $600 \leq T \leq 2400$ K. For $T > 2400$ K, coefficient values suggested by [60] are used. If $\kappa_i ps \ll 1$ for all i, Equation 5.3-81 simplifies to

$$\epsilon = \sum_{i=0}^{I} a_{\epsilon,i} \kappa_i ps \tag{5.3-84}$$

Comparing Equation 5.3-84 with the gray gas model with absorption coefficient a, it can be seen that the change of the radiation intensity over the distance s in the WSGGM is exactly the same as in the gray gas model with the absorption coefficient

$$a = \sum_{i=0}^{I} a_{\epsilon,i} \kappa_i p \tag{5.3-85}$$

which does not depend on s. In the general case, a is estimated as

$$a = -\frac{\ln(1-\epsilon)}{s} \tag{5.3-86}$$

where the emissivity ϵ for the WSGGM is computed using Equation 5.3-81. a as defined by Equation 5.3-86 depends on s, reflecting the non-gray nature of the absorption of thermal radiation in molecular gases. In ANSYS FLUENT, Equation 5.3-85 is used when $s \leq 10^{-4}$ m and Equation 5.3-86 is used for $s > 10^{-4}$ m. Note that for $s \approx 10^{-4}$ m, the values of a predicted by Equations 5.3-85 and 5.3-86 are practically identical (since Equation 5.3-86 reduces to Equation 5.3-85 in the limit of small s).

ANSYS FLUENT allows you to specify s as the mean beam length or the characteristic cell size. The model based on the mean beam length is the recommended approach, especially when you have a nearly homogeneous medium and you are interested in the radiation exchange between the walls of the enclosure. You can specify the mean beam length or have ANSYS FLUENT compute it. If you do decide to use the WSGGM based on the characteristic cell size, note that the predicted values of a will be mesh dependent (this is a known limitation of the model). See Section 8.8.1: Inputs for a Composition-Dependent Absorption Coefficient in the separate User's Guide for details about setting properties for the WSGGM.

i The WSGGM cannot be used to specify the absorption coefficient in each band when using the non-gray DO model. If the WSGGM is used with the non-gray DO model, the absorption coefficient will be the same in all bands.

When $p_{\rm tot} \neq 1$ atm

The WSGGM, as described above, assumes that $p_{\rm tot}$—the total (static) gas pressure—is equal to 1 atm. In cases where $p_{\rm tot}$ is not unity (e.g., combustion at high temperatures), scaling rules suggested in [84] are used to introduce corrections. When $p_{\rm tot} < 0.9$ atm or $p_{\rm tot} > 1.1$ atm, the values for κ_i in Equations 5.3-81 and 5.3-85 are rescaled:

$$\kappa_i \rightarrow \kappa_i p_{\rm tot}^m \tag{5.3-87}$$

where m is a non-dimensional value obtained from [84], which depends on the partial pressures and temperature T of the absorbing gases, as well as on $p_{\rm tot}$.

The Effect of Soot on the Absorption Coefficient

When soot formation is computed, ANSYS FLUENT can include the effect of the soot concentration on the radiation absorption coefficient. The generalized soot model estimates the effect of the soot on radiative heat transfer by determining an effective absorption coefficient for soot. The absorption coefficient of a mixture of soot and an absorbing (radiating) gas is then calculated as the sum of the absorption coefficients of pure gas and pure soot:

$$a_{s+g} = a_g + a_s \tag{5.3-88}$$

where a_g is the absorption coefficient of gas without soot (obtained from the WSGGM) and

$$a_s = b_1 \rho_m [1 + b_T (T - 2000)] \tag{5.3-89}$$

with

$$b_1 = 1232.4 \text{ m}^2/\text{kg} \text{ and } b_T \approx 4.8 \times 10^{-4} \text{ K}^{-1}$$

ρ_m is the soot density in kg/m^3.

The coefficients b_1 and b_T were obtained [302] by fitting Equation 5.3-89 to data based on the Taylor-Foster approximation [348] and data based on the Smith et al. approximation [326].

See Section 8.8: Radiation Properties and Section 21.3.1: Using the Soot Models in the separate User's Guide for information about including the soot-radiation interaction effects.

The Effect of Particles on the Absorption Coefficient

ANSYS FLUENT can also include the effect of discrete phase particles on the radiation absorption coefficient, provided that you are using either the P-1 or the DO model. When the P-1 or DO model is active, radiation absorption by particles can be enabled. The particle emissivity, reflectivity, and scattering effects are then included in the calculation of the radiative heat transfer. See Section 23.5: Setting Material Properties for the Discrete Phase in the separate User's Guide for more details on the input of radiation properties for the discrete phase.

5.3.9 Choosing a Radiation Model

For certain problems one radiation model may be more appropriate than the others. When deciding which radiation model to use, consider the following:

- Optical thickness: The optical thickness aL is a good indicator of which model to use in your problem. Here, L is an appropriate length scale for your domain. For flow in a combustor, for example, L is the diameter of the combustion chamber. If $aL \gg 1$, your best alternatives are the P-1 and Rosseland models. The P-1 model should typically be used for optical thicknesses > 1. For optical thickness > 3, the Rosseland model is cheaper and more efficient. For high optical thickness cases, a second-order discretization scheme for the DO model is recommended. The DTRM and the DO model work across the full range of optical thicknesses, but are substantially more expensive to use. Consequently, you should use the "thick-limit" models, P-1 and Rosseland, if the problem allows it. For optically thin problems ($aL < 1$), the DTRM and the DO model, only, are appropriate.

- Scattering and emissivity: The P-1, Rosseland, and DO models account for scattering, while the DTRM neglects it. Since the Rosseland model uses a temperature slip condition at walls, it is insensitive to wall emissivity.

- Particulate effects: Only the P-1 and DO models account for exchange of radiation between gas and particulates (see Equation 5.3-8).
- Semi-transparent walls (interior and exterior): Only the DO model allows you to model semi-transparent walls of various types (e.g., glass).
- Specular walls: Only the DO model allows specular reflection (e.g., for dust-free mirror).
- Partially-specular walls: Only the DO model allows specular reflection (e.g., dusty mirror).
- Non-gray radiation: Only the DO model allows you to compute non-gray radiation using a gray band model.
- Localized heat sources: In problems with localized sources of heat, the P-1 model may over-predict the radiative fluxes. The DO model is probably the best suited for computing radiation for this case, although the DTRM, with a sufficiently large number of rays, is also acceptable.
- Enclosure radiative transfer with non-participating media: The surface-to-surface (S2S) model is suitable for this type of problem. The radiation models used with participating media may, in principle, be used to compute the surface-to-surface radiation, but they are not always efficient.

External Radiation

If you need to include radiative heat transfer from the exterior of your physical model, you can include an external radiation boundary condition in your model (for details, see Section 7.3.14: Thermal Boundary Conditions at Walls in the separate User's Guide). If you are not concerned with radiation within the domain, this boundary condition can be used without activating one of the radiation models.

Chapter 6. Heat Exchangers

Many engineering systems, including power plants, climate control, and engine cooling systems typically contain tubular heat exchangers. However, for most engineering problems, it is impractical to model individual fins and tubes of a heat exchanger core. In principle, heat exchanger cores introduce a pressure drop to the primary fluid stream and transfer heat to a second fluid, a coolant, referred to here as the auxiliary fluid.

In ANSYS FLUENT, lumped-parameter models are used to account for the pressure loss and auxiliary fluid heat rejection. ANSYS FLUENT provides two heat exchanger models: the macro (ungrouped and grouped) models and the dual cell model. The macro model allows you to choose between two heat transfer models, namely the simple-effectiveness-model and the number-of-transfer-units (NTU) model. The models can be used to compute auxiliary fluid inlet temperature for a fixed heat rejection or total heat rejection for a fixed auxiliary fluid inlet temperature. For the simple-effectiveness-model, the auxiliary fluid may be single-phase or two-phase. The dual cell model uses the NTU method for heat transfer calculations. This model allows the solution of auxiliary flow on a separate mesh (other than the primary fluid mesh), unlike the macro model, where the auxiliary flow is modeled as 1-D flow. The dual cell model also offers more flexibility as far as the shape of the heat exchanger is concerned, and overcomes some of the major limitations present in the macro model.

For more information about using the heat exchanger models, see Section 14: Modeling Heat Exchangers in the separate User's Guide.

The following sections contain information about the theory behind the heat exchanger models:

- Section 6.1: The Macro Heat Exchanger Models
- Section 6.2: The Dual Cell Model

6.1 The Macro Heat Exchanger Models

The following sections contain information about the theory behind the macro heat exchanger models:

- Section 6.1.1: Overview and Restrictions of the Macro Heat Exchanger Models
- Section 6.1.2: Macro Heat Exchanger Model Theory

6.1.1 Overview and Restrictions of the Macro Heat Exchanger Models

Overview

In a typical heat exchanger core, the auxiliary fluid temperature is stratified in the direction of the auxiliary fluid flow. As a result, heat rejection is not constant over the entire core. In ANSYS FLUENT, the fluid zone representing the heat exchanger core is subdivided into macroscopic cells or *macros* along the auxiliary fluid path, as in Figure 6.1.1. In this figure, the core is discretized into 3×4×2 macros. This configuration consists of 2 passes, each pass having four rows and three columns of macros. The auxiliary fluid inlet temperature to each macro is computed and then subsequently used to compute the heat rejection from each macro. This approach provides a realistic heat rejection distribution over the heat exchanger core.

To use the heat exchanger models, you must define one or more fluid zone(s) to represent the heat exchanger core. Typically, the fluid zone is sized to the dimension of the core itself. As part of the setup procedure, you will define the auxiliary fluid path, the number of macros, and the physical properties and operating conditions of the core (pressure drop parameters, heat exchanger effectiveness, auxiliary fluid flow rate, etc.).

You can also combine several fluid zones as a single heat exchanger group. In this situation each fluid zone acts as a separate heat exchanger core, and the auxiliary fluid mass flow rate of the heat exchanger group is divided among the zones in the ratio of the respective volumes. For the purpose of auxiliary fluid flow, heat exchanger groups can also be connected in series. In addition, a heat exchanger group can have an auxiliary fluid pressure drop (e.g., for pressure dependent properties) and a supplementary auxiliary fluid stream entering or leaving it. For more information on heat exchanger groups, see Section 14.4: Using the Grouped Macro Heat Exchanger Model.

The heat exchanger models were designed for "compact" heat exchangers, implying that the primary fluid side flow is unidirectional. The auxiliary fluid is assumed to flow through a large number of parallel tubes, which can optionally double back in a serpentine pattern to create a number of "passes". You can independently choose the principal auxiliary fluid flow direction, the pass-to-pass direction and the external primary fluid flow direction.

i It is highly recommended that the free-form Tet mesh is not used in the macro heat exchanger model. Instead, evenly distributed Hex/Wedge cells should be used for improved accuracy and a more robust solution process.

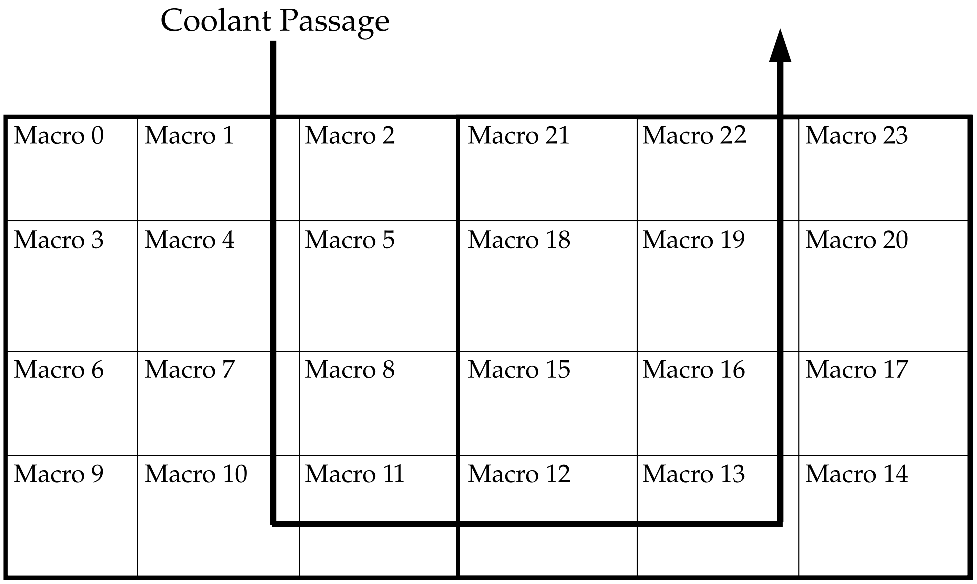


Figure 6.1.1: Core Discretized Into 3 × 4×2 Macros

Restrictions

The following restrictions exist for the macro heat exchanger models:

- The core must be approximately rectangular in shape.
- The primary fluid streamwise direction (see Equation 6.1-1) must be aligned with one of the three orthogonal axes defined by the rectangular core.
- Flow acceleration effects are neglected in calculating the pressure loss coefficient.
- For the simple-effectiveness-model, the primary fluid capacity rate must be less than the auxiliary fluid capacity rate.
- Auxiliary fluid phase change cannot be modeled using the ntu-model.
- The macro-based method requires that an equal number of cells reside in each macro of equal size and shape.
- Auxiliary fluid flow is assumed to be 1-D.
- The pass width has to be uniform.
- Accuracy is not guaranteed when the mesh is not structured or layered.
- Accuracy is not guaranteed when there is upstream diffusion of temperature at the inlet/outlet of the core.
- Non-conformal meshes cannot be attached to the inlet/outlet of the core. An extra layer has to be created to avoid it.

6.1.2 Macro Heat Exchanger Model Theory

In ANSYS FLUENT, the heat exchanger core is treated as a fluid zone with momentum and heat transfer. Pressure loss is modeled as a momentum sink in the momentum equation, and heat transfer is modeled as a heat source in the energy equation.

ANSYS FLUENT provides two heat transfer models: the default ntu-model and the simple-effectiveness-model. The simple-effectiveness-model interpolates the effectiveness from the velocity vs effectiveness curve that you provide. For the ntu-model, ANSYS FLUENT calculates the effectiveness, ϵ, from the NTU value that is calculated by ANSYS FLUENT from the heat transfer data provided by the user in tabular format. ANSYS FLUENT will automatically convert this heat transfer data to a primary fluid mass flow rate vs NTU curve (this curve will be piecewise linear). This curve will be used by ANSYS FLUENT to calculate the NTU for macros based on their size and primary fluid flow rate.

The ntu-model provides the following features:

- The model can be used to check the heat capacity for both the primary and the auxiliary fluid, and takes the lesser of the two for the calculation of heat transfer.
- The model can be used to model heat transfer to the primary fluid from the auxiliary fluid and vice versa.
- The model can be used to model primary fluid-side reverse flow.
- The model can be used with variable density of the primary fluid.
- The model can be used in either the serial or parallel ANSYS FLUENT solvers.
- Transient profiles can be used for the auxiliary fluid inlet temperature and for total heat rejection.
- Transient profiles can be used for auxiliary mass flow rates.

The simple-effectiveness-model provides the following features:

- The model can be used to model heat transfer from the auxiliary fluid to the primary fluid.
- The auxiliary fluid properties can be a function of pressure and temperature, thus allowing phase change of the auxiliary fluid.
- The model can be used by serial as well as parallel solvers.
- The model can be used to make a network of heat exchangers using a heat exchanger group (Section 14.4: Using the Grouped Macro Heat Exchanger Model).
- Transient profiles can be used for the auxiliary fluid inlet temperature and for total heat rejection.
- Transient profiles can be used for auxiliary mass flow rates.

Streamwise Pressure Drop

In both heat transfer models, pressure loss is modeled using the porous media model in ANSYS FLUENT. For the dual cell model (Section 6.2: The Dual Cell Model), pressure loss is used for both streams, while for the macro model, it is used only for the primary side.

The loss coefficients of the porous media model are computed using curve fitting of the pressure-versus-flow rate data outside of ANSYS FLUENT, which you will specify for the cell zone conditions. However, in some cases, the data for curve-fitting is not available. The macro model provides an additional means of getting the coefficients if the data is not available. The coefficients can also be automatically computed (and updated) using a known pressure loss coefficient as a function of some geometric parameters, the theory of which is defined below:

$$\Delta p = \frac{1}{2} f \rho_m U_{A_{\min}}^2 \qquad (6.1\text{-}1)$$

where

Δp = streamwise pressure drop
f = streamwise pressure loss coefficient
ρ_m = mean primary fluid density
$U_{A_{\min}}$ = primary fluid velocity at the minimum flow area

The pressure loss coefficient is computed from

$$f = (K_c + 1 - \sigma^2) - (1 - \sigma^2 - K_e)\frac{\nu_e}{\nu_i} + 2\left(\frac{\nu_e}{\nu_i} - 1\right) + f_c \frac{A}{A_c}\frac{\nu_m}{\nu_i} \qquad (6.1\text{-}2)$$

where

σ = minimum flow to face area ratio
K_c = entrance loss coefficient
K_e = exit loss coefficient
A = primary fluid-side surface area
A_c = minimum cross-sectional flow area
f_c = core friction factor
ν_e = specific volume at the exit
ν_i = specific volume at the inlet
ν_m = mean specific volume $\equiv \frac{1}{2}(\nu_e + \nu_i)$

K_c and K_e are empirical quantities obtained from experimental data. You will need to specify these parameters based on graphs that are closest to the heat exchanger configuration that you are setting up [160], [158]. These parameters are used to set up large resistances in the two non-streamwise directions, effectively forcing the primary fluid flow through the core to be unidirectional.

In Equation 6.1-2, the core friction factor is defined as

$$f_c = a\mathrm{Re}_{\mathrm{min}}^b \tag{6.1-3}$$

where

a	=	core friction coefficient
b	=	core friction exponent
$\mathrm{Re}_{\mathrm{min}}$	=	Reynolds number for velocity at the minimum flow area

a and b are empirical quantities obtained from experimental data. You will need to specify the core friction coefficient and exponent based on graphs that are closest to the heat exchanger models that you set up [160], [158].

The Reynolds number in Equation 6.1-3 is defined as

$$Re_{\mathrm{min}} = \frac{\rho_m U_{A_{\mathrm{min}}} D_h}{\mu_m} \tag{6.1-4}$$

where

ρ_m	=	mean primary fluid density
μ_m	=	mean primary fluid viscosity
D_h	=	hydraulic diameter
$U_{A_{\mathrm{min}}}$	=	primary fluid velocity at the minimum flow area

For a heat exchanger core, the hydraulic diameter can be defined as

$$D_h = 4L\left(\frac{A_c}{A}\right) \tag{6.1-5}$$

where L is the flow length of the heat exchanger. If the tubes are normal to the primary fluid flow, then L is the length in the primary fluid flow direction. Note that $U_{A_{\mathrm{min}}}$ can be calculated from

$$U_{A_{\mathrm{min}}} = \frac{U}{\sigma} \tag{6.1-6}$$

where U is the primary fluid velocity and σ is the minimum flow to face area ratio.

Heat Transfer Effectiveness

For the simple-effectiveness-model, the heat-exchanger effectiveness, ϵ, is defined as the ratio of actual rate of heat transfer from the hot to cold fluid to the maximum possible rate of heat transfer. The maximum possible heat transfer is given by

$$q_{\rm max} = C_{\rm min}(T_{\rm in,hot} - T_{\rm in,cold}) \tag{6.1-7}$$

where $T_{\rm in,hot}$ and $T_{\rm in,cold}$ are the inlet temperatures of the hot and cold fluids and

$$C_{\rm min} = \min[(\dot{m}c_p)_{\rm hot}, (\dot{m}c_p)_{\rm cold}] \tag{6.1-8}$$

Thus, the actual rate of heat transfer, q, is defined as

$$q = \epsilon C_{\rm min}(T_{\rm in,hot} - T_{\rm in,cold}) \tag{6.1-9}$$

The value of ϵ depends on the heat exchanger geometry and flow pattern (parallel flow, counter flow, cross flow, etc.). Even though the effectiveness of the primary fluid is computed using uniform conditions on the entire heat exchanger core, it is being applied to a small portion of the core represented by a computational cell. This can make it less accurate for some heat exchanger cores, where there is a strong variation in the primary flow. For a core with a strong primary flow variation, the NTU model must be used.

For the ntu-model, given the heat exchanger performance data (total heat rejection versus primary flow rate) based on uniform test conditions, ANSYS FLUENT calculates the effectiveness of the entire heat exchanger from the ratio of heat capacity and the number of transfer units using the relation

$$\epsilon = 1 - \exp\left[-\frac{1}{C_r}N_{\rm tu}^{0.22}(1 - e^{-C_r N_{\rm tu}^{0.78}})\right] \tag{6.1-10}$$

where C_r is the ratio of $C_{\rm min}$ to $C_{\rm max}$.

The heat exchanger performance data should be specified for a number of auxiliary flow rates so that ANSYS FLUENT can compute the number of transfer units versus the primary fluid flow rate for a number of auxiliary fluid flow rates. This NTU, which is based on the full heat exchanger and uniform conditions, is scaled for each macro using the ratio of their volumes and minimum heat capacities.

For each macro, the primary fluid inlet temperature is calculated using the mass average of the incoming primary fluid temperatures at the boundaries. This automatically takes into account any reverse flow of the primary fluid at the boundaries.

Heat Rejection

Heat rejection is computed for each cell within a macro and added as a source term to the energy equation for the primary fluid flow. Note that heat rejection from the auxiliary fluid to primary fluid can be either positive or negative.

For the simple-effectiveness-model, the heat transfer for a given cell is computed from

$$q_{\rm cell} = \epsilon(\dot{m}c_p)_{\rm g}(T_{\rm in,auxiliary} - T_{\rm cell}) \tag{6.1-11}$$

where

ϵ	=	heat exchanger effectiveness
$(\dot{m}c_p)_{\rm g}$	=	primary fluid capacity rate (flow rate × specific heat)
$T_{\rm in,auxiliary}$	=	auxiliary fluid inlet temperature of macro containing the cell
$T_{\rm cell}$	=	cell temperature

For the simple-effectiveness-model, the heat rejection from a macro is calculated by summing the heat transfer of all the cells contained within the macro

$$q_{\rm macro} \quad = \sum_{\text{all cells in macro}} q_{\rm cell} \tag{6.1-12}$$

For the ntu-model, the heat transfer for a macro is calculated from

$$q_{\rm macro} = \epsilon C_{\rm min}(T_{\rm in,auxiliary} - T_{\rm in,primary}) \tag{6.1-13}$$

where

ϵ	=	macro effectiveness
$T_{\rm in,auxiliary}$	=	macro auxiliary fluid inlet temperature
$T_{\rm in,primary}$	=	macro primary fluid inlet temperature

For the ntu-model, the heat transfer for a given cell is computed from

$$q_{\rm cell} = q_{\rm macro}\frac{V_{\rm cell}}{V_{\rm macro}} \tag{6.1-14}$$

For both heat exchanger models, the total heat rejection from the heat exchanger core is computed as the sum of the heat rejection from all the macros:

$$q_{\text{total}} = \sum_{\text{all macros}} q_{\text{macro}} \tag{6.1-15}$$

The auxiliary fluid inlet temperature to each macro ($T_{\text{in,auxiliary}}$ in Equations 6.1-11 and 6.1-13) is computed based on the energy balance of the auxiliary fluid at a previous macro computation. For a given macro,

$$q_{\text{macro}} = (\dot{m})_{\text{auxiliary}}(h_{\text{out}} - h_{\text{in}}) \tag{6.1-16}$$

where h_{in} and h_{out} are the inlet and outlet enthalpies of the auxiliary fluid in the macro. The auxiliary fluid outlet temperature from the macro is calculated as

$$T_{\text{out}} = \begin{cases} \frac{h_{\text{out}}}{c_{p,\text{auxiliary}}} & \text{constant specific heat method} \\ \\ f(h_{\text{out}}, p) & \text{UDF method} \end{cases} \tag{6.1-17}$$

where

f = user-defined function
p = auxiliary fluid pressure

The values of h_{out} and T_{out} then become the inlet conditions to the next macro.

The first row of macros (Macros 0, 1, and 2 in Figure 6.1.1) are assumed to be where the auxiliary fluid enters the heat exchanger core. When the fixed total heat rejection from the heat exchanger core is specified, the inlet temperature to the first row of macros is iteratively computed, so that all of the equations are satisfied simultaneously. When a fixed auxiliary fluid inlet temperature is specified, the heat transfer for the first row of macros are used to calculate their exit enthalpy, which becomes the inlet condition for the next row macros. At the end of each pass, the outlet enthalpy of each macro (in the last row) is mass averaged to obtain the inlet condition for the next pass macros.

Macro Heat Exchanger Group Connectivity

If the optional macro heat exchanger group is used, a single heat exchanger may be consist of multiple fluid zones. In this case, the auxiliary fluid is assumed to flow through these zones in parallel. Thus, after taking into account any auxiliary stream effects, the auxiliary fluid inlet mass flow rate is automatically apportioned to each zone in the group as follows:

$$\dot{m}_i = \left(\frac{\sum_k V_{i,k}}{\sum_i \sum_k V_{i,k}} \right) \dot{m} \tag{6.1-18}$$

where $\dot{m}_i$ is the total auxiliary mass flow rate for the heat exchanger group. $V_{i,k}$ refers to the volume of the kth finite volume cell within the ith fluid zone. Within each zone, the auxiliary fluid flows through each macro in series as usual.

At the outlet end of the group, the parallel auxiliary fluid streams through the individual zones are recombined, and the outlet auxiliary fluid enthalpy is calculated on a mass-averaged basis:

$$\bar{h} = \left(\frac{\sum_i \dot{m}_i h_i}{\sum_i \dot{m}_i} \right) \tag{6.1-19}$$

With user-defined functions, the simple-effectiveness-model allows you to simulate two-phase auxiliary fluid flows and other complex auxiliary fluid enthalpy relationships of the form

$$h = h(T, p, x) \tag{6.1-20}$$

where p is the absolute pressure and x is the quality (mass fraction of vapor) of a two-phase vapor-liquid mixture. When pressure-dependent auxiliary fluid properties are used, the mean pressure within each macro is calculated and passed to the user-defined function as

$$\bar{p}_j = p_{\rm in} + \left(j + \frac{1}{2} \right) \frac{\Delta p}{N} \tag{6.1-21}$$

where

j	=	macro row index
$p_{\rm in}$	=	inlet auxiliary fluid pressure
Δp	=	overall pressure drop across a heat exchanger group
N	=	number of rows per pass × number of passes.

To learn how to use the macro heat exchanger models, refer to Section 14.3: Using the Ungrouped Macro Heat Exchanger Model and Section 14.4: Using the Grouped Macro Heat Exchanger Model in the separate User's Guide.

6.2 The Dual Cell Model

The following sections contain information about the theory behind the dual cell heat exchanger model:

- Section 6.2.1: Overview and Restrictions of the Dual Cell Model
- Section 6.2.2: Dual Cell Model Theory

6.2.1 Overview and Restrictions of the Dual Cell Model

Overview

The Macro Model is quite suitable for thin rectangular heat exchanger cores, where the pass-to-pass is perpendicular to the primary flow direction and the auxiliary flow is uniform. Moreover, the mesh should be uniform and structured. However, many practical heat exchangers have a non-rectangular core and the auxiliary fluid, before reaching the core, may pass through arbitrary shaped inlet tanks, which make them highly non-uniform. It is quite possible that due to the complex shape of the core and or ease of meshing, the structured mesh may not be the obvious choice. These shortcomings of the macro model can be easily overcome by using the dual cell heat exchanger model. This model allows the solution of both the primary and auxiliary flow on separate co-located meshes and couples the two flows only through heat transfer at the heat exchanger core.

Restrictions

The following restrictions exist for the dual cell heat exchanger models:

- Heat transfer calculations are based on the NTU method only.
- Multipass heat exchangers cannot be modeled. This will require hooking a UDF.
- In the case of a heat exchanger core with non-matching meshes, the total cell count for the primary and auxiliary core should approximately be the same.

6.2.2 Dual Cell Model Theory

The dual cell heat exchanger consists of two porous fluid zones, namely a primary zone and an auxiliary zone. The two zones are solved simultaneously and are coupled only through heat transfer. The common region in each zone, where heat transfer takes place, represents the heat exchanger core. The cores for both primary and auxiliary zones occupy the same physical space, as shown in Figure 6.2.1. The cells in the two cores should overlap completely in the physical space to ensure conservative heat transfer. Heat transfer occurs between cells in close proximity based on the cell centroid. In other words, a primary zone cell exchanges heat with one, and only one, auxiliary zone cell and vice versa. Therefore, if one of the core (say primary) mesh is too coarse or fine relative to the other core (say auxiliary) conservation of heat transfer is not ensured. Heat transfer calculations in the dual cell model are based on the NTU method.

NTU Relations

In a cross-flow pattern, the NTU values are calculated as in Equation 6.1-10. The equation is solved iteratively using the Newton-Raphson. For parallel flow, the NTU value is calculated as follows:

$$NTU = \frac{-ln(1-\epsilon-\epsilon C_r)}{(C_r+1)} \tag{6.2-1}$$

and for counter flow, the following equation is used:

$$NTU_{C_r=1} = \frac{\epsilon}{1-\epsilon} \tag{6.2-2}$$

Otherwise,

$$NTU = \frac{1}{(C_r-1)} ln\left[\frac{(1-\epsilon)}{(1-\epsilon C_r)}\right] \tag{6.2-3}$$

where C_r is the heat capacity ratio and ϵ is the effectiveness.

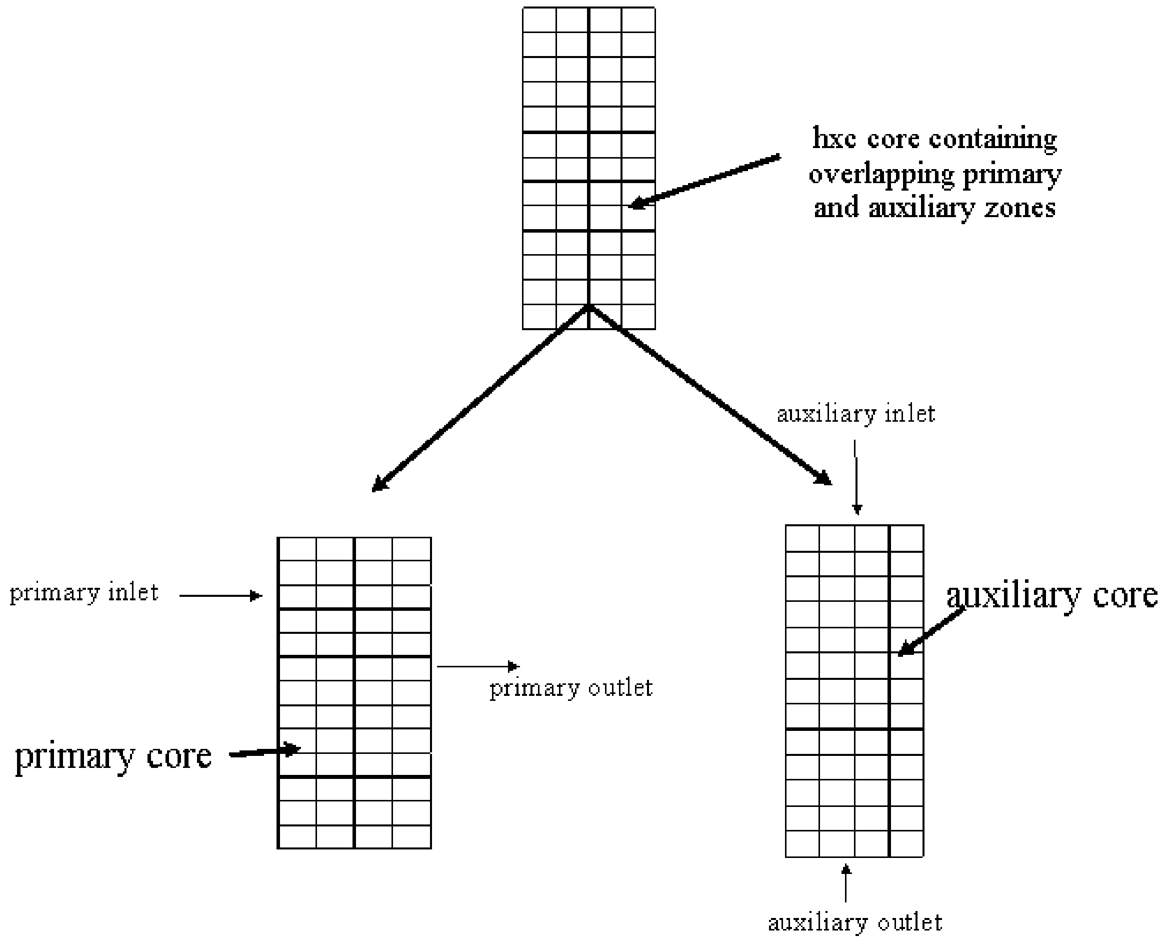


Figure 6.2.1: Core with Matching Quad Meshes for Primary and Auxiliary Zones in a Cross-Flow Pattern

Heat Rejection

Heat rejection is computed for each cell in the two cores (primary and auxiliary) and added as a source term to the energy equation for the respective flows. This is illustrated in Figure 6.2.2 and the following equations:

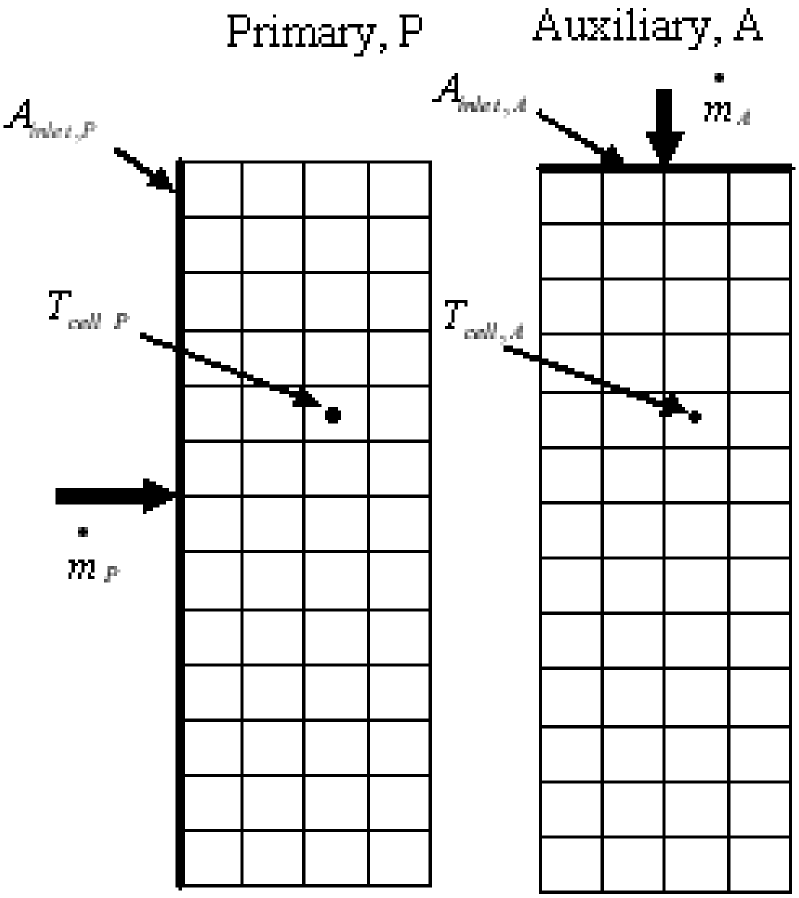


Figure 6.2.2: Core with Primary and Auxiliary Zones with Possible Overlap of Cells

$$\dot{m}_{scaled,A} = \rho_{cell,A}\vec{V}_{cell,A}A_{inlet,A} \tag{6.2-4}$$

$$\dot{m}_{scaled,P} = \rho_{cell,P}\vec{V}_{cell,P}A_{inlet,P} \tag{6.2-5}$$

$$C_{min,scaled} = min[(C_{p,cell}\dot{m}_{scale})|_P, (C_{p,cell}\dot{m}_{scale})|_A] \tag{6.2-6}$$

$$NTU_{scaled} = [NTU_{full}(\dot{m}_{scaled,P}, \dot{m}_{scaled,A})]_{bilinearinterpolation} \tag{6.2-7}$$

$$(UA)_{scaled} = NTU_{scaled}\ \ C_{min,scaled} \tag{6.2-8}$$

$$dq = q_{cell} = \frac{(UA)_{scaled}(T_{cell,A} - T_{cell,P})}{Volume_{cell}} \tag{6.2-9}$$

where $T_{cell,A}$ = auxiliary cell temperature
$T_{cell,P}$ = primary cell temperature
U = overall heat transfer coefficient
A = total heat transfer area
$C_{min,scaled}$ = scaled minimum heat capacity rate

You can supply the NTU values, or it can be calculated using the supplied raw data and the effectiveness-NTU relation that you specify.

To learn how to use the dual cell heat exchanges model, refer to Section 14.5: Using the Dual Cell Heat Exchanger Model in the separate User's Guide.

Chapter 7. Species Transport and Finite-Rate Chemistry

ANSYS FLUENT can model the mixing and transport of chemical species by solving conservation equations describing convection, diffusion, and reaction sources for each component species. Multiple simultaneous chemical reactions can be modeled, with reactions occurring in the bulk phase (volumetric reactions) and/or on wall or particle surfaces, and in the porous region. Species transport modeling capabilities, both with and without reactions, are described in this chapter.

Note that you may also want to consider modeling your turbulent reacting flame using the mixture fraction approach (for non-premixed systems, described in Chapter 8: Non-Premixed Combustion), the reaction progress variable approach (for premixed systems, described in Chapter 9: Premixed Combustion), the partially premixed approach (described in Chapter 10: Partially Premixed Combustion), or the composition PDF Transport approach (described in Chapter 11: Composition PDF Transport). Modeling multiphase species transport and finite-rate chemistry can be found in Chapter 16: Multiphase Flows.

Information is divided into the following sections:

- Section 7.1: Volumetric Reactions
- Section 7.2: Wall Surface Reactions and Chemical Vapor Deposition
- Section 7.3: Particle Surface Reactions

For more information about using these models in ANSYS FLUENT, see Chapter 15: Modeling Species Transport and Finite-Rate Chemistry in the separate User's Guide.

7.1 Volumetric Reactions

Theoretical information about species transport and finite-rate chemistry as related to volumetric reactions is presented in this section. Additional information can be found in the following sections:

- Section 7.1.1: Species Transport Equations
- Section 7.1.2: The Generalized Finite-Rate Formulation for Reaction Modeling

For more information about using species transport and finite-rate chemistry as related to volumetric reactions, see Section 15.1: Volumetric Reactions in the separate User's Guide.

7.1.1 Species Transport Equations

When you choose to solve conservation equations for chemical species, ANSYS FLUENT predicts the local mass fraction of each species, Y_i, through the solution of a convection-diffusion equation for the ith species. This conservation equation takes the following general form:

$$\frac{\partial}{\partial t}(\rho Y_i) + \nabla \cdot (\rho \vec{v} Y_i) = -\nabla \cdot \vec{J}_i + R_i + S_i \tag{7.1-1}$$

where R_i is the net rate of production of species i by chemical reaction (described later in this section) and S_i is the rate of creation by addition from the dispersed phase plus any user-defined sources. An equation of this form will be solved for $N-1$ species where N is the total number of fluid phase chemical species present in the system. Since the mass fraction of the species must sum to unity, the Nth mass fraction is determined as one minus the sum of the $N-1$ solved mass fractions. To minimize numerical error, the Nth species should be selected as that species with the overall largest mass fraction, such as N_2 when the oxidizer is air.

Mass Diffusion in Laminar Flows

In Equation 7.1-1, $\vec{J}_i$ is the diffusion flux of species i, which arises due to gradients of concentration and temperature. By default, ANSYS FLUENT uses the dilute approximation (also called Fick's law) to model mass diffusion due to concentration gradients, under which the diffusion flux can be written as

$$\vec{J}_i = -\rho D_{i,m} \nabla Y_i - D_{T,i} \frac{\nabla T}{T} \tag{7.1-2}$$

Here $D_{i,m}$ is the mass diffusion coefficient for species i in the mixture, and $D_{T,i}$ is the thermal (Soret) diffusion coefficient.

For certain laminar flows, the dilute approximation may not be acceptable, and full multicomponent diffusion is required. In such cases, the Maxwell-Stefan equations can be solved; see Section 8.9.2: Full Multicomponent Diffusion in the separate User's Guide for details.

Mass Diffusion in Turbulent Flows

In turbulent flows, ANSYS FLUENT computes the mass diffusion in the following form:

$$\vec{J}_i = -\left(\rho D_{i,m} + \frac{\mu_t}{\mathrm{Sc}_t}\right) \nabla Y_i - D_{T,i} \frac{\nabla T}{T} \tag{7.1-3}$$

where Sc_t is the turbulent Schmidt number ($\frac{\mu_t}{\rho D_t}$ where μ_t is the turbulent viscosity and D_t is the turbulent diffusivity). The default Sc_t is 0.7. Note that turbulent diffusion generally overwhelms laminar diffusion, and the specification of detailed laminar diffusion properties in turbulent flows is generally not necessary.

Treatment of Species Transport in the Energy Equation

For many multicomponent mixing flows, the transport of enthalpy due to species diffusion

$$\nabla \cdot \left[\sum_{i=1}^{n} h_i \vec{J}_i\right]$$

can have a significant effect on the enthalpy field and should not be neglected. In particular, when the Lewis number

$$\mathrm{Le}_i = \frac{k}{\rho c_p D_{i,m}} \tag{7.1-4}$$

for any species is far from unity, neglecting this term can lead to significant errors. ANSYS FLUENT will include this term by default. In Equation 7.1-4, k is the thermal conductivity.

Diffusion at Inlets

For the pressure-based solver in ANSYS FLUENT, the net transport of species at inlets consists of both convection and diffusion components. (For the density-based solvers, only the convection component is included.) The convection component is fixed by the inlet species mass fraction specified by you. The diffusion component, however, depends on the gradient of the computed species field at the inlet. Thus the diffusion component (and therefore the net inlet transport) is not specified a priori. For information about specifying the net inlet transport of species, see Section 15.1.5: Defining Cell Zone and Boundary Conditions for Species in the separate User's Guide.

7.1.2 The Generalized Finite-Rate Formulation for Reaction Modeling

The reaction rates that appear as source terms in Equation 7.1-1 are computed in ANSYS FLUENT, for turbulent flows, by one of three models:

- Laminar finite-rate model: The effect of turbulent fluctuations are ignored, and reaction rates are determined by Arrhenius kinetic expressions.
- Eddy-dissipation model: Reaction rates are assumed to be controlled by the turbulence, so expensive Arrhenius chemical kinetic calculations can be avoided. The model is computationally cheap, but, for realistic results, only one or two step heat-release mechanisms should be used.
- Eddy-dissipation-concept (EDC) model: Detailed Arrhenius chemical kinetics can be incorporated in turbulent flames. Note that detailed chemical kinetic calculations are computationally expensive.

The generalized finite-rate formulation is suitable for a wide range of applications including laminar or turbulent reaction systems, and combustion systems with premixed, non-premixed, or partially-premixed flames.

The Laminar Finite-Rate Model

The laminar finite-rate model computes the chemical source terms using Arrhenius expressions, and ignores the effects of turbulent fluctuations. The model is exact for laminar flames, but is generally inaccurate for turbulent flames due to highly non-linear Arrhenius chemical kinetics. The laminar model may, however, be acceptable for combustion with relatively slow chemistry and small turbulence-chemistry interaction, such as supersonic flames.

The net source of chemical species i due to reaction is computed as the sum of the Arrhenius reaction sources over the N_R reactions that the species participate in:

$$R_i = M_{w,i} \sum_{r=1}^{N_R} \hat{R}_{i,r} \tag{7.1-5}$$

where $M_{w,i}$ is the molecular weight of species i and $\hat{R}_{i,r}$ is the Arrhenius molar rate of creation/destruction of species i in reaction r. Reaction may occur in the continuous phase at wall surfaces.

Consider the rth reaction written in general form as follows:

$$\sum_{i=1}^{N} \nu'_{i,r}\mathcal{M}_i \underset{k_{b,r}}{\overset{k_{f,r}}{\rightleftharpoons}} \sum_{i=1}^{N} \nu''_{i,r}\mathcal{M}_i \tag{7.1-6}$$

where

N	=	number of chemical species in the system
$\nu'_{i,r}$	=	stoichiometric coefficient for reactant i in reaction r
$\nu''_{i,r}$	=	stoichiometric coefficient for product i in reaction r
$\mathcal{M}_i$	=	symbol denoting species i
$k_{f,r}$	=	forward rate constant for reaction r
$k_{b,r}$	=	backward rate constant for reaction r

Equation 7.1-6 is valid for both reversible and non-reversible reactions. (Reactions in ANSYS FLUENT are non-reversible by default.) For non-reversible reactions, the backward rate constant, $k_{b,r}$, is simply omitted.

The summations in Equation 7.1-6 are for all chemical species in the system, but only species that appear as reactants or products will have non-zero stoichiometric coefficients. Hence, species that are not involved will drop out of the equation.

For a non-reversible reaction (that is, the Include Backward Reaction button is disabled), the molar rate of creation/destruction of species i in reaction r ($\hat{R}_{i,r}$ in Equation 7.1-5) is given by

$$\hat{R}_{i,r} = \Gamma \left(\nu''_{i,r} - \nu'_{i,r}\right) \left(k_{f,r} \prod_{j=1}^{N} \left[C_{j,r}\right]^{(\eta'_{j,r}+\eta''_{j,r})} \right) \tag{7.1-7}$$

where

$C_{j,r}$	=	molar concentration of species j in reaction r (kgmol/m^3)
$\eta'_{j,r}$	=	rate exponent for reactant species j in reaction r
$\eta''_{j,r}$	=	rate exponent for product species j in reaction r

For a reversible reaction, the molar rate of creation/destruction of species i in reaction r is given by

$$\hat{R}_{i,r} = \Gamma \left(\nu''_{i,r} - \nu'_{i,r}\right) \left(k_{f,r} \prod_{j=1}^{N} \left[C_{j,r}\right]^{\eta'_{j,r}} - k_{b,r} \prod_{j=1}^{N} \left[C_{j,r}\right]^{\nu''_{j,r}} \right) \tag{7.1-8}$$

Note that the rate exponent for the reverse reaction part in Equation 7.1-8 is always the product species stoichiometric coefficient ($\nu''_{j,r}$).

For information about inputting the stoichiometric coefficients and rate exponents for both global forward (non-reversible) reactions and elementary (reversible) reactions, see Section 15.1.3: Inputs for Reaction Definition in the separate User's Guide.

Γ represents the net effect of third bodies on the reaction rate. This term is given by

$$\Gamma = \sum_{j}^{N} \gamma_{j,r} C_j \tag{7.1-9}$$

where $\gamma_{j,r}$ is the third-body efficiency of the jth species in the rth reaction. By default, ANSYS FLUENT does not include third-body effects in the reaction rate calculation. You can, however, opt to include the effect of third-body efficiencies if you have data for them.

The forward rate constant for reaction r, $k_{f,r}$, is computed using the Arrhenius expression

$$k_{f,r} = A_r T^{\beta_r} e^{-E_r/RT} \tag{7.1-10}$$

where

A_r	=	pre-exponential factor (consistent units)
β_r	=	temperature exponent (dimensionless)
E_r	=	activation energy for the reaction (J/kgmol)
R	=	universal gas constant (J/kgmol-K)

You (or the database) will provide values for $\nu'_{i,r}$, $\nu''_{i,r}$, $\eta'_{j,r}$, $\eta''_{j,r}$, β_r, A_r, E_r, and, optionally, $\gamma_{j,r}$ during the problem definition in ANSYS FLUENT.

If the reaction is reversible, the backward rate constant for reaction r, $k_{b,r}$, is computed from the forward rate constant using the following relation:

$$k_{b,r} = \frac{k_{f,r}}{K_r} \tag{7.1-11}$$

where K_r is the equilibrium constant for the rth reaction, computed from

$$K_r = \exp\left(\frac{\Delta S_r^0}{R} - \frac{\Delta H_r^0}{RT}\right)\left(\frac{p_{\rm atm}}{RT}\right)^{\sum_{i=1}^{N}(\nu''_{i,r} - \nu'_{i,r})} \tag{7.1-12}$$

where $p_{\rm atm}$ denotes atmospheric pressure (101325 Pa). The term within the exponential function represents the change in Gibbs free energy, and its components are computed as follows:

$$\frac{\Delta S_r^0}{R} = \sum_{i=1}^{N}\left(\nu''_{i,r} - \nu'_{i,r}\right)\frac{S_i^0}{R} \tag{7.1-13}$$

$$\frac{\Delta H_r^0}{RT} = \sum_{i=1}^{N}\left(\nu''_{i,r} - \nu'_{i,r}\right)\frac{h_i^0}{RT} \tag{7.1-14}$$

where S_i^0 and h_i^0 are the standard-state entropy and standard-state enthalpy (heat of formation). These values are specified in ANSYS FLUENT as properties of the mixture material.

Pressure-Dependent Reactions

ANSYS FLUENT can use one of three methods to represent the rate expression in pressure-dependent (or pressure fall-off) reactions. A “fall-off” reaction is one in which the temperature and pressure are such that the reaction occurs between Arrhenius high-pressure and low-pressure limits, and thus is no longer solely dependent on temperature.

There are three methods of representing the rate expressions in this fall-off region. The simplest one is the Lindemann [198] form. There are also two other related methods, the Troe method [111] and the SRI method [339], that provide a more accurate description of the fall-off region.

Arrhenius rate parameters are required for both the high- and low-pressure limits. The rate coefficients for these two limits are then blended to produce a smooth pressure-dependent rate expression. In Arrhenius form, the parameters for the high-pressure limit (k) and the low-pressure limit ($k_{\rm low}$) are as follows:

$$k = AT^{\beta} e^{-E/RT} \tag{7.1-15}$$

$$k_{\rm low} = A_{\rm low} T^{\beta_{\rm low}} e^{-E_{\rm low}/RT} \tag{7.1-16}$$

The net rate constant at any pressure is then taken to be

$$k_{\rm net} = k \left(\frac{p_r}{1+p_r} \right) F \tag{7.1-17}$$

where p_r is defined as

$$p_r = \frac{k_{\rm low}[M]}{k} \tag{7.1-18}$$

and $[M]$ is the concentration of the bath gas, which can include third-body efficiencies. If the function F in Equation 7.1-17 is unity, then this is the Lindemann form. ANSYS FLUENT provides two other forms to describe F, namely the Troe method and the SRI method.

In the Troe method, F is given by

$$\log F = \left\{ 1 + \left[\frac{\log p_r + c}{n - d(\log p_r + c)} \right]^2 \right\}^{-1} \log F_{\rm cent} \tag{7.1-19}$$

where

$$
\begin{aligned}
c &= -0.4 - 0.67 \log F_{\rm cent} && (7.1\text{-}20) \\
n &= 0.75 - 1.27 \log F_{\rm cent} && (7.1\text{-}21) \\
d &= 0.14 && (7.1\text{-}22)
\end{aligned}
$$

and

$$F_{\rm cent} = (1-\alpha)e^{-T/T_1} + \alpha e^{-T/T_2} + e^{-T_3/T} \qquad (7.1\text{-}23)$$

The parameters α, T_3, T_2, and T_1 are specified as inputs.

In the SRI method, the blending function F is approximated as

$$F = d\left[a \exp\left(\frac{-b}{T}\right) + \exp\left(\frac{-T}{c}\right)\right]^X T^e \qquad (7.1\text{-}24)$$

where

$$X = \frac{1}{1+\log^2 p_r} \qquad (7.1\text{-}25)$$

In addition to the three Arrhenius parameters for the low-pressure limit ($k_{\rm low}$) expression, you must also supply the parameters a, b, c, d, and e in the F expression.

Chemical kinetic mechanisms usually contain a wide range of time scales and form a set of highly non-linear, stiff coupled equations. For solution procedure guidelines, see Section 15.1.7: Solution Procedures for Chemical Mixing and Finite-Rate Chemistry in the separate User's Guide. Also, if you have a chemical mechanism in CHEMKIN [161] format, you can import this mechanism into ANSYS FLUENT (see Section 15.1.9: Importing a Volumetric Kinetic Mechanism in CHEMKIN Format in the separate User's Guide).

The Eddy-Dissipation Model

Most fuels are fast burning, and the overall rate of reaction is controlled by turbulent mixing. In non-premixed flames, turbulence slowly convects/mixes fuel and oxidizer into the reaction zones where they burn quickly. In premixed flames, the turbulence slowly convects/mixes cold reactants and hot products into the reaction zones, where reaction occurs rapidly. In such cases, the combustion is said to be mixing-limited, and the complex, and often unknown, chemical kinetic rates can be safely neglected.

ANSYS FLUENT provides a turbulence-chemistry interaction model, based on the work of Magnussen and Hjertager [216], called the eddy-dissipation model. The net rate of production of species i due to reaction r, $R_{i,r}$, is given by the smaller (i.e., limiting value) of the two expressions below:

$$R_{i,r} = \nu'_{i,r} M_{w,i} A \rho \frac{\epsilon}{k} \min_{\mathcal{R}} \left(\frac{Y_{\mathcal{R}}}{\nu'_{\mathcal{R},r} M_{w,\mathcal{R}}} \right) \tag{7.1-26}$$

$$R_{i,r} = \nu'_{i,r} M_{w,i} A B \rho \frac{\epsilon}{k} \frac{\sum_P Y_P}{\sum_j^N \nu''_{j,r} M_{w,j}} \tag{7.1-27}$$

where Y_P is the mass fraction of any product species, P
$Y_{\mathcal{R}}$ is the mass fraction of a particular reactant, $\mathcal{R}$
A is an empirical constant equal to 4.0
B is an empirical constant equal to 0.5

In Equations 7.1-26 and 7.1-27, the chemical reaction rate is governed by the large-eddy mixing time scale, k/ϵ, as in the eddy-breakup model of Spalding [333]. Combustion proceeds whenever turbulence is present ($k/\epsilon > 0$), and an ignition source is not required to initiate combustion. This is usually acceptable for non-premixed flames, but in premixed flames, the reactants will burn as soon as they enter the computational domain, upstream of the flame stabilizer. To remedy this, ANSYS FLUENT provides the finite-rate/eddy-dissipation model, where both the Arrhenius (Equation 7.1-8), and eddy-dissipation (Equations 7.1-26 and 7.1-27) reaction rates are calculated. The net reaction rate is taken as the minimum of these two rates. In practice, the Arrhenius rate acts as a kinetic “switch”, preventing reaction before the flame holder. Once the flame is ignited, the eddy-dissipation rate is generally smaller than the Arrhenius rate, and reactions are mixing-limited.

i Although ANSYS FLUENT allows multi-step reaction mechanisms (number of reactions > 2) with the eddy-dissipation and finite-rate/eddy-dissipation models, these will likely produce incorrect solutions. The reason is that multi-step chemical mechanisms are based on Arrhenius rates, which differ for each reaction. In the eddy-dissipation model, every reaction has the same, turbulent rate, and therefore the model should be used only for one-step (reactant → product), or two-step (reactant → intermediate, intermediate → product) global reactions. The model cannot predict kinetically controlled species such as radicals. To incorporate multi-step chemical kinetic mechanisms in turbulent flows, use the EDC model (described below).

i The eddy-dissipation model requires products to initiate reaction (see Equation 7.1-27). When you initialize the solution for steady flows, ANSYS FLUENT sets all species mass fractions to a maximum of the user specified initial value and 0.01. This is usually sufficient to start the reaction. However, if you converge a mixing solution first, where all product mass fractions are zero, you may then have to patch products into the reaction zone to ignite the flame. For details, see Section 15.1.7: Ignition in Combustion Simulations in the separate User's Guide.

The Eddy-Dissipation Model for LES

When the LES turbulence model is used, the turbulent mixing rate, ϵ/k in Equations 7.1-26 and 7.1-27, is replaced by the subgrid-scale mixing rate. This is calculated as

$$\tau_{sgs}^{-1} = \sqrt{2 S_{ij} S_{ij}} \tag{7.1-28}$$

where

τ_{sgs}^{-1} = subgrid-scale mixing rate (s^{-1})

S_{ij} = $\frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)$ = strain rate tensor (s^{-1})

The Eddy-Dissipation-Concept (EDC) Model

The eddy-dissipation-concept (EDC) model is an extension of the eddy-dissipation model to include detailed chemical mechanisms in turbulent flows [215]. It assumes that reaction occurs in small turbulent structures, called the fine scales. The length fraction of the fine scales is modeled as [115]

$$\xi^* = C_\xi \left(\frac{\nu\epsilon}{k^2}\right)^{1/4} \tag{7.1-29}$$

where * denotes fine-scale quantities and

C_ξ = volume fraction constant = 2.1377

ν = kinematic viscosity

The volume fraction of the fine scales is calculated as ξ^{*3}. Species are assumed to react in the fine structures over a time scale

$$\tau^* = C_\tau \left(\frac{\nu}{\epsilon}\right)^{1/2} \tag{7.1-30}$$

where C_τ is a time scale constant equal to 0.4082.

In ANSYS FLUENT, combustion at the fine scales is assumed to occur as a constant pressure reactor, with initial conditions taken as the current species and temperature in the cell. Reactions proceed over the time scale τ^*, governed by the Arrhenius rates of Equation 7.1-8, and are integrated numerically using the ISAT algorithm [277]. ISAT can accelerate the chemistry calculations by two to three orders of magnitude, offering substantial reductions in run-times. Details about the ISAT algorithm may be found in Sections 11.3.3 and 11.3.4. ISAT is very powerful, but requires some care. See Section 19.6.2: Using ISAT Efficiently in the separate User's Guide for details on using ISAT efficiently.

The source term in the conservation equation for the mean species i, Equation 7.1-1, is modeled as

$$R_i = \frac{\rho (\xi^*)^2}{\tau^* [1 - (\xi^*)^3]} (Y_i^* - Y_i) \tag{7.1-31}$$

where Y_i^* is the fine-scale species mass fraction after reacting over the time τ^*.

The EDC model can incorporate detailed chemical mechanisms into turbulent reacting flows. However, typical mechanisms are invariably stiff and their numerical integration is computationally costly. Hence, the model should be used only when the assumption of fast chemistry is invalid, such as modeling the slow CO burnout in rapidly quenched flames, or the NO conversion in selective non-catalytic reduction (SNCR).

For guidelines on obtaining a solution using the EDC model, see Section 15.1.7: Solution of Stiff Laminar Chemistry Systems in the separate User's Guide.

7.2 Wall Surface Reactions and Chemical Vapor Deposition

For gas-phase reactions, the reaction rate is defined on a volumetric basis and the rate of creation and destruction of chemical species becomes a source term in the species conservation equations. For surface reactions, the rate of adsorption and desorption is governed by both chemical kinetics and diffusion to and from the surface. Wall surface reactions thus create sources and sinks of chemical species in the gas phase, as well as on the reacting surface.

Theoretical information about wall surface reactions and chemical vapor deposition is presented in this section. Information can be found in the following sections:

- Section 7.2.1: Surface Coverage Reaction Rate Modification
- Section 7.2.2: Reaction-Diffusion Balance for Surface Chemistry
- Section 7.2.3: Slip Boundary Formulation for Low-Pressure Gas Systems

For more information about using wall surface reactions and chemical vapor deposition, see Section 15.2: Wall Surface Reactions and Chemical Vapor Deposition in the separate User's Guide.

Consider the rth wall surface reaction written in general form as follows:

$$\sum_{i=1}^{N_g} g'_{i,r} G_i + \sum_{i=1}^{N_b} b'_{i,r} B_i + \sum_{i=1}^{N_s} s'_{i,r} S_i \overset{K_r}{\rightleftharpoons} \sum_{i=1}^{N_g} g''_{i,r} G_i + \sum_{i=1}^{N_b} b''_{i,r} B_i + \sum_{i=1}^{N_s} s''_{i,r} S_i \qquad (7.2\text{-}1)$$

where G_i, B_i, and S_i represent the gas phase species, the bulk (or solid) species, and the surface-adsorbed (or site) species, respectively. N_g, N_b, and N_s are the total numbers of these species. $g'_{i,r}$, $b'_{i,r}$, and $s'_{i,r}$ are the stoichiometric coefficients for each reactant species i, and $g''_{i,r}$, $b''_{i,r}$, and $s''_{i,r}$ are the stoichiometric coefficients for each product species i. K_r is the overall forward reaction rate constant. Note that ANSYS FLUENT cannot model reversible surface reactions.

The summations in Equation 7.2-1 are for all chemical species in the system, but only species involved as reactants or products will have non-zero stoichiometric coefficients. Hence, species that are not involved will drop out of the equation.

The rate of the rth reaction is

$$\mathcal{R}_r = k_{f,r} \left(\prod_{i=1}^{N_g} [C_i]_{\text{wall}}^{\eta'_{i,g,r}} \right) \left(\prod_{j=1}^{N_s} [S_j]_{\text{wall}}^{\eta'_{j,s,r}} \right) \qquad (7.2\text{-}2)$$

where $[\ ]_{\text{wall}}$ represents molar concentrations of surface-adsorbed species on the wall. $\eta'_{i,g,r}$ is the rate exponent for the i^{th} gaseous species as reactant in the reaction and $\eta'_{j,s,r}$ is the rate exponent for the j^{th} site species as reactant in the reaction. It is assumed that the reaction rate does not depend on concentrations of the bulk (solid) species. From this, the net molar rate of production or consumption of each species i is given by

$$\hat{R}_{i,\text{gas}} = \sum_{r=1}^{N_{\text{rxn}}} (g''_{i,r} - g'_{i,r}) \mathcal{R}_r \qquad i = 1, 2, 3, \ldots, N_g \qquad (7.2\text{-}3)$$

$$\hat{R}_{i,\text{bulk}} = \sum_{r=1}^{N_{\text{rxn}}} (b''_{i,r} - b'_{i,r}) \mathcal{R}_r \qquad i = 1, 2, 3, \ldots, N_b \qquad (7.2\text{-}4)$$

$$\hat{R}_{i,\text{site}} = \sum_{r=1}^{N_{\text{rxn}}} (s''_{i,r} - s'_{i,r}) \mathcal{R}_r \qquad i = 1, 2, 3, \ldots, N_s \qquad (7.2\text{-}5)$$

The forward rate constant for reaction r ($k_{f,r}$) is computed using the Arrhenius expression,

$$k_{f,r} = A_r T^{\beta_r} e^{-E_r/RT} \tag{7.2-6}$$

where A_r = pre-exponential factor (consistent units)
β_r = temperature exponent (dimensionless)
E_r = activation energy for the reaction (J/kgmol)
R = universal gas constant (J/kgmol-K)

You (or the database) will provide values for $g'_{i,r}$, $g''_{i,r}$, $b'_{i,r}$, $b''_{i,r}$, $s'_{i,r}$, $s''_{i,r}$, β_r, A_r, and E_r.

To include the mass transfer effects and model heat release, refer to Section 15.2.3: Including Mass Transfer To Surfaces in Continuity, Section 15.2.4: Wall Surface Mass Transfer Effects in the Energy Equation, and Section 15.2.5: Modeling the Heat Release Due to Wall Surface Reactions in the separate User's Guide

7.2.1 Surface Coverage Reaction Rate Modification

ANSYS FLUENT has the option to modify the surface reaction rate as a function of species site coverages. In such cases, the forward rate constant for the rth reaction is evaluated as,

$$k_{f,r} = A_r T^{\beta_r} e^{-E_r/RT} \prod_{k_{site}} \left(10^{Z_k \eta_{k,r}}\right) \left(Z_k{}^{\mu_{k,r}}\right) \left(e^{-\epsilon_{k,r} Z_k/RT}\right) \tag{7.2-7}$$

In Equation 7.2-7, the three surface coverage rate modification parameters for specie k in reaction r are $\eta_{k,r}$, $\mu_{k,r}$ and $\epsilon_{k,r}$. These parameters default to zero for reaction species that are not surface rate modifying. The surface (coverage) site fraction, Z_k is the fraction of surface sites covered by specie k, and is defined as,

$$Z_k = [S_k]/\rho_{site} \tag{7.2-8}$$

where $[S_k]$ is the surface site concentration and ρ_{site} is the surface site density (see Equation 7.2-13).

7.2.2 Reaction-Diffusion Balance for Surface Chemistry

Reactions at surfaces change gas-phase, surface-adsorbed (site) and bulk (solid) species. On reacting surfaces, the mass flux of each gas specie due to diffusion and convection to/from the surface is balanced with its rate of consumption/production on the surface,

$$\rho_{\rm wall} D_i \frac{\partial Y_{i,{\rm wall}}}{\partial n} - \dot{m}_{\rm dep} Y_{i,{\rm wall}} = M_{w,i} \hat{R}_{i,{\rm gas}} \qquad i = 1, 2, 3, \ldots, N_g \qquad (7.2\text{-}9)$$

$$\frac{\partial \left[S_i\right]_{\rm wall}}{\partial t} = \hat{R}_{i,{\rm site}} \qquad i = 1, 2, 3, \ldots, N_s \qquad (7.2\text{-}10)$$

The wall mass fraction $Y_{i,{\rm wall}}$ is related to concentration by

$$[G_i]_{\rm wall} = \frac{\rho_{\rm wall} Y_{i,{\rm wall}}}{M_{w,i}} \qquad (7.2\text{-}11)$$

$\dot{m}_{\rm dep}$ is the net rate of mass deposition or etching as a result of surface reaction; i.e.,

$$\dot{m}_{\rm dep} = \sum_{i=1}^{N_b} M_{w,i} \hat{R}_{i,{\rm bulk}} \qquad (7.2\text{-}12)$$

$[S_i]_{\rm wall}$ is the site species concentration at the wall, and is defined as

$$[S_i]_{\rm wall} = \rho_{\rm site} Z_i \qquad (7.2\text{-}13)$$

where $\rho_{\rm site}$ is the site density and Z_i is the site coverage of species i.

Equations 7.2-9 and 7.2-10 are solved for the dependent variables $Y_{i,{\rm wall}}$ and Z_i using a point-by-point coupled Newton solver. The effective gas-phase reaction source terms are then available for solution of the gas-phase species transport Equation 7.1-1.

7.2.3 Slip Boundary Formulation for Low-Pressure Gas Systems

Most semiconductor fabrication devices operate far below atmospheric pressure, typically only a few millitorrs. At such low pressures, the fluid flow is in the slip regime and the normally used no-slip boundary conditions for velocity and temperature are no longer valid.

The Knudsen number, denoted Kn, and defined as the ratio of mean free path to a characteristic length scale of the system, is used to quantify continuum flow regimes. Since the mean free path increases as the pressure is lowered, the high end of Kn values represents *free molecular flow* and the low end the *continuum regime*. The range in between these two extremes is called the slip regime ($0.01 < \mathrm{Kn} < 0.1$) [28] In the slip regime, the gas-phase velocity at a solid surface differs from the velocity at which the wall moves, and the gas temperature at the surface differs from the wall temperature. Maxwell's models are adopted for these physical phenomena in ANSYS FLUENT for their simplicity and effectiveness.

- velocity slip

$$U_w - U_g = \left(\frac{2-\alpha_v}{\alpha_v}\right)\mathrm{Kn}\frac{\partial U}{\partial n} \approx \left(\frac{2-\alpha_v}{\alpha_v}\right)\frac{\lambda}{\delta}(U_g - U_c) \qquad (7.2\text{-}14)$$

$$V_g \equiv (\vec{V}\cdot\vec{n})_g = V_w \qquad (7.2\text{-}15)$$

Here, U and V represents the velocity component that is parallel and normal to the wall, respectively. The subscripts g, w and c indicate gas, wall and cell-center velocities. δ is the distance from cell center to the wall. α_v is the momentum accommodation coefficient of the gas mixture and its value is calculated as mass-fraction weighted average of each gas species in the system.

$$\alpha_v = \sum_{i=1}^{N_g} Y_i \alpha_i \qquad (7.2\text{-}16)$$

The mean free path, λ, is computed as follows:

$$\lambda = \frac{k_B T}{\sqrt{2}\pi\sigma^2 p} \qquad (7.2\text{-}17)$$

$$\sigma = \sum_{i=1}^{N_g} Y_i \sigma_i \qquad (7.2\text{-}18)$$

σ_i is the Lennard-Jones characteristic length of species i. k_B is the Boltzmann constant, $1.38066 \times 10^{-23} J/K$.

Equations 7.2-14 and 7.2-15 indicate that while the gas velocity component normal to the wall is the same as the wall normal velocity, the tangential components slip. The values lie somewhere between the cell-center and the wall values. These two equations can be combined to give a generalized formulation:

$$\vec{V}_g = \frac{\vec{V}_w + \frac{k}{\delta}[(\vec{V}_w \cdot \vec{n})\vec{n} + \vec{V}_c - (\vec{V}_c \cdot \vec{n})\vec{n}]}{1 + \frac{k}{\delta}} \tag{7.2-19}$$

where

$$k \equiv \lambda \left(\frac{2 - \alpha_v}{\alpha_v} \right) \tag{7.2-20}$$

- temperature jump

$$T_w - T_g = 2 \left(\frac{2 - \alpha_T}{\alpha_T} \right) \mathrm{Kn} \frac{\partial T}{\partial n} \approx 2 \left(\frac{2 - \alpha_T}{\alpha_T} \right) \frac{\lambda}{\delta} (T_g - T_c) \tag{7.2-21}$$

or equivalently

$$T_g = \frac{T_w + \beta T_c}{1 + \beta} \tag{7.2-22}$$

where

$$\beta = \frac{2(2 - \alpha_T)}{\alpha_T \delta} \tag{7.2-23}$$

α_T is the thermal accommodation coefficient of the gas mixture and is calculated as $\alpha_T = \sum Y_i \alpha_{T,i}$.

i The low-pressure slip boundary formulation is available only with the pressure-based solver.

7.3 Particle Surface Reactions

As described in Section 15.4.5: The Multiple Surface Reactions Model, it is possible to define multiple particle surface reactions to model the surface combustion of a combusting discrete-phase particle. This section provides theoretical background about particle surface reactions. Information can be found in the following sections:

- Section 7.3.1: General Description
- Section 7.3.2: ANSYS FLUENT Model Formulation
- Section 7.3.3: Extension for Stoichiometries with Multiple Gas Phase Reactants
- Section 7.3.4: Solid-Solid Reactions
- Section 7.3.5: Solid Decomposition Reactions
- Section 7.3.6: Solid Deposition Reactions
- Section 7.3.7: Gaseous Solid Catalyzed Reactions on the Particle Surface

For more information about using particle surface reactions, see Section 15.3: Particle Surface Reactions in the separate User's Guide.

7.3.1 General Description

The relationships for calculating char particle burning rates are presented and discussed in detail by Smith [324]. The particle reaction rate, $\mathcal{R}$ (kg/m^2-s), can be expressed as

$$\mathcal{R} = D_0(C_g - C_s) = R_c(C_s)^N \tag{7.3-1}$$

where

D_0 = bulk diffusion coefficient (m/s)
C_g = mean reacting gas species concentration in the bulk (kg/m^3)
C_s = mean reacting gas species concentration at the particle surface (kg/m^3)
R_c = chemical reaction rate coefficient (units vary)
N = apparent reaction order (dimensionless)

In Equation 7.3-1, the concentration at the particle surface, C_s, is not known, so it should be eliminated, and the expression is recast as follows:

$$\mathcal{R} = R_c\left[C_g - \frac{\mathcal{R}}{D_0}\right]^N \tag{7.3-2}$$

This equation has to be solved by an iterative procedure, with the exception of the cases when $N = 1$ or $N = 0$. When $N = 1$, Equation 7.3-2 can be written as

$$\mathcal{R} = \frac{C_g R_c D_0}{D_0 + R_c} \tag{7.3-3}$$

In the case of $N = 0$, if there is a finite concentration of reactant at the particle surface, the solid depletion rate is equal to the chemical reaction rate. If there is no reactant at the surface, the solid depletion rate changes abruptly to the diffusion-controlled rate. In this case, however, ANSYS FLUENT will always use the chemical reaction rate for stability reasons.

7.3.2 ANSYS FLUENT Model Formulation

A particle undergoing an exothermic reaction in the gas phase is shown schematically in Figure 7.3.1. T_p and T_∞ are the temperatures in Equation 15.4-78.

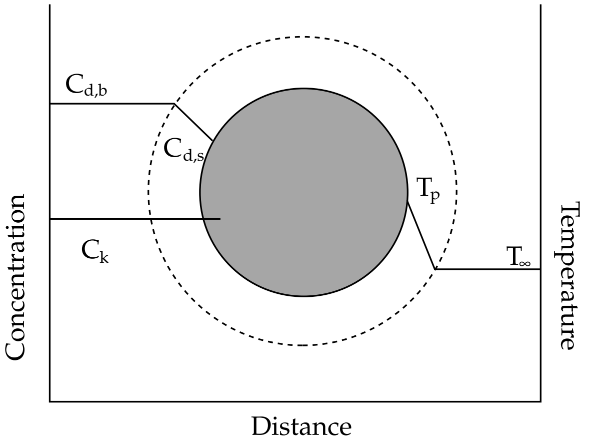


Figure 7.3.1: A Reacting Particle in the Multiple Surface Reactions Model

Based on the analysis above, ANSYS FLUENT uses the following equation to describe the rate of reaction r of a particle surface species j with the gas phase species n. The reaction stoichiometry of reaction r in this case is described by

$$\text{particle species } j(\text{s}) + \text{gas phase species } n \rightarrow \text{products}$$

and the rate of reaction is given as

$$\overline{\mathcal{R}}_{j,r} = A_p \eta_r Y_j \mathcal{R}_{j,r} \tag{7.3-4}$$

$$\mathcal{R}_{j,r} = \mathcal{R}_{\mathrm{kin},r} \left(p_n - \frac{\mathcal{R}_{j,r}}{D_{0,r}} \right)^{N} \tag{7.3-5}$$

where

$\overline{\mathcal{R}}_{j,r}$ = rate of particle surface species depletion (kg/s)
A_p = particle surface area (m^2)
Y_j = mass fraction of surface species j in the particle
η_r = effectiveness factor (dimensionless)
$\mathcal{R}_{j,r}$ = rate of particle surface species reaction per unit area (kg/m^2-s)
p_n = bulk partial pressure of the gas phase species (Pa)
$D_{0,r}$ = diffusion rate coefficient for reaction r
$\mathcal{R}_{\mathrm{kin},r}$ = kinetic rate of reaction r (units vary)
N_r = apparent order of reaction r

The effectiveness factor, η_r, is related to the surface area, and can be used in each reaction in the case of multiple reactions. $D_{0,r}$ is given by

$$D_{0,r} = C_{1,r} \frac{\left[(T_p + T_\infty)/2 \right]^{0.75}}{d_p} \tag{7.3-6}$$

The kinetic rate of reaction r is defined as

$$\mathcal{R}_{\mathrm{kin},r} = A_r {T_p}^{\beta_r} e^{-(E_r/RT_p)} \tag{7.3-7}$$

The rate of the particle surface species depletion for reaction order $N_r = 1$ is given by

$$\overline{\mathcal{R}}_{j,r} = A_p \eta_r Y_j p_n \frac{\mathcal{R}_{\mathrm{kin},r} D_{0,r}}{D_{0,r} + \mathcal{R}_{\mathrm{kin},r}} \tag{7.3-8}$$

For reaction order $N_r = 0$,

$$\overline{\mathcal{R}}_{j,r} = A_p \eta_r Y_j \mathcal{R}_{\mathrm{kin},r} \tag{7.3-9}$$

7.3.3 Extension for Stoichiometries with Multiple Gas Phase Reactants

When more than one gas phase reactant takes part in the reaction, the reaction stoichiometry must be extended to account for this case:

$$\text{particle species } j(\text{s}) + \text{gas phase species } 1 + \text{gas phase species } 2 + \ \ldots$$

$$+ \text{ gas phase species } n_{\max} \rightarrow \text{products}$$

To describe the rate of reaction r of a particle surface species j in the presence of $n_{\max}$ gas phase species n, it is necessary to define the diffusion-limited species for each solid particle reaction, i.e., the species for which the concentration gradient between the bulk and the particle surface is the largest. For the rest of the species, the surface and the bulk concentrations are assumed to be equal. The concentration of the diffusion-limited species is shown as $C_{d,b}$ and $C_{d,s}$ in Figure 7.3.1, and the concentrations of all other species are denoted as C_k. For stoichiometries with multiple gas phase reactants, the bulk partial pressure p_n in Equations 7.3-4 and 7.3-8 is the bulk partial pressure of the diffusion-limited species, $p_{r,d}$ for reaction r.

The kinetic rate of reaction r is then defined as

$$\mathcal{R}_{\text{kin},r} = \frac{A_r T^{\beta_r} e^{-(E_r/RT)}}{(p_{r,d})^{N_{r,d}}} \prod_{n=1}^{n_{\max}} p_n^{N_{r,n}} \tag{7.3-10}$$

where

p_n = bulk partial pressure of gas species n
$N_{r,n}$ = reaction order in species n

When this model is enabled, the constant $C_{1,r}$ (Equation 7.3-6) and the effectiveness factor η_r (Equation 7.3-4) are entered in the Reactions dialog box (see Section 15.3.1: User Inputs for Particle Surface Reactions in the separate User's Guide).

7.3.4 Solid-Solid Reactions

Reactions involving only particle surface reactants can be modeled, provided that the particle surface reactants and products exist on the same particle.

$$\text{particle species } 1(\text{s}) + \text{particle species } 2(\text{s}) + \ldots \rightarrow \text{products}$$

The reaction rate for this case is given by Equation 7.3-9.

7.3.5 Solid Decomposition Reactions

The decomposition reactions of particle surface species can be modeled.

$$\text{particle species } 1(\text{s}) + \text{particle species } 2(\text{s}) + \ \ldots\ + \text{particle species } n_{\max}(\text{s}) \rightarrow$$
$$\text{gas species } j + \text{products}$$

The reaction rate for this case is given by Equations 7.3-4–7.3-10, where the diffusion-limited species is now the gaseous product of the reaction. If there are more than one gaseous product species in the reaction, it is necessary to define the diffusion-limited species for the particle reaction as the species for which the concentration gradient between the bulk and the particle surface is the largest.

7.3.6 Solid Deposition Reactions

The deposition reaction of a solid species on a particle can be modeled with the following assumptions:

$$\text{gas species } 1 + \text{gas species } 2 + \ \ldots\ + \text{gas species } n_{\max} \rightarrow \text{solid species } j(\text{s}) + \text{products}$$

The theoretical analysis and Equations 7.3-4–7.3-10 are applied for the surface reaction rate calculation, with the mass fraction of the surface species set to unity in Equations 7.3-4, 7.3-8, and 7.3-9.

In ANSYS FLUENT, for the particle surface species to be deposited on a particle, a finite mass of the species must already exist in the particle. This allows for activation of the deposition reaction selectively to particular injection particles. It follows that, to initiate the solid species deposition reaction on a particle, the particle must be defined in the **Set Injection Properties** dialog box (or **Set Multiple Injection Properties** dialog box) to contain a small mass fraction of the solid species to be deposited. For details on defining the particle surface species mass fractions, see Section 15.3.3: Using the Multiple Surface Reactions Model for Discrete-Phase Particle Combustion in the separate User's Guide.

7.3.7 Gaseous Solid Catalyzed Reactions on the Particle Surface

Reactions of gaseous species catalyzed on the particle surface can also be modeled following Equations 7.3-4–7.3-10 for the surface reaction rate calculation, with the mass fraction of the surface species set to unity in Equations 7.3-4, 7.3-8, and 7.3-9. To apply this type of reaction, see Section 15.3.2: Modeling Gaseous Solid Catalyzed Reactions in the separate User's Guide. For details on defining the particle surface species mass fractions, see Section 15.3.3: Using the Multiple Surface Reactions Model for Discrete-Phase Particle Combustion in the separate User's Guide.

Chapter 8. Non-Premixed Combustion

In non-premixed combustion, fuel and oxidizer enter the reaction zone in distinct streams. This is in contrast to premixed systems, in which reactants are mixed at the molecular level before burning. Examples of non-premixed combustion include pulverized coal furnaces, diesel internal-combustion engines and pool fires.

Under certain assumptions, the thermochemistry can be reduced to a single parameter: the mixture fraction. The mixture fraction, denoted by f, is the mass fraction that originated from the fuel stream. In other words, it is the local mass fraction of burnt and unburnt fuel stream elements (C, H, etc.) in all the species (CO_2, H_2O, O_2, etc.). The approach is elegant because atomic elements are conserved in chemical reactions. In turn, the mixture fraction is a conserved scalar quantity, and therefore its governing transport equation does not have a source term. Combustion is simplified to a mixing problem, and the difficulties associated with closing non-linear mean reaction rates are avoided. Once mixed, the chemistry can be modeled as being in chemical equilibrium with the Equilibrium model, being near chemical equilibrium with the Steady Laminar Flamelet model, or significantly departing from chemical equilibrium with the Unsteady Laminar Flamelet model.

For more information about using the non-premixed combustion model, see Chapter 16: Modeling Non-Premixed Combustion in the separate User's Guide. Theoretical information about the non-premixed combustion model is presented in the following sections:

- Section 8.1: Introduction
- Section 8.2: Non-Premixed Combustion and Mixture Fraction Theory
- Section 8.3: Restrictions and Special Cases for Using the Non-Premixed Model
- Section 8.4: The Laminar Flamelet Models Theory
- Section 8.5: The Steady Laminar Flamelet Model Theory
- Section 8.6: The Unsteady Laminar Flamelet Model Theory

8.1 Introduction

Non-premixed modeling involves the solution of transport equations for one or two conserved scalars (the mixture fractions). Equations for individual species are not solved. Instead, species concentrations are derived from the predicted mixture fraction fields. The thermochemistry calculations are preprocessed and then tabulated for look-up in ANSYS FLUENT. Interaction of turbulence and chemistry is accounted for with an assumed-shape Probability Density Function (PDF).

8.2 Non-Premixed Combustion and Mixture Fraction Theory

Information about non-premixed combustion and mixture fraction theory are presented in the following sections:

- Section 8.2.1: Mixture Fraction Theory
- Section 8.2.2: Modeling of Turbulence-Chemistry Interaction
- Section 8.2.3: Non-Adiabatic Extensions of the Non-Premixed Model
- Section 8.2.4: Chemistry Tabulation

8.2.1 Mixture Fraction Theory

Definition of the Mixture Fraction

The basis of the non-premixed modeling approach is that under a certain set of simplifying assumptions, the instantaneous thermochemical state of the fluid is related to a conserved scalar quantity known as the mixture fraction, f. The mixture fraction can be written in terms of the atomic mass fraction as [319]

$$f = \frac{Z_i - Z_{i,\rm ox}}{Z_{i,\rm fuel} - Z_{i,\rm ox}} \tag{8.2-1}$$

where Z_i is the elemental mass fraction for element, i. The subscript ox denotes the value at the oxidizer stream inlet and the subscript fuel denotes the value at the fuel stream inlet. If the diffusion coefficients for all species are equal, then Equation 8.2-1 is identical for all elements, and the mixture fraction definition is unique. The mixture fraction is thus the elemental mass fraction that originated from the fuel stream.

If a secondary stream (another fuel or oxidant, or a non-reacting stream) is included, the fuel and secondary mixture fractions are simply the elemental mass fractions of the fuel and secondary streams, respectively. The sum of all three mixture fractions in the system (fuel, secondary stream, and oxidizer) is always equal to 1:

$$f_{\rm fuel} + f_{\rm sec} + f_{\rm ox} = 1 \tag{8.2-2}$$

This indicates that only points on the plane ABC (shown in Figure 8.2.1) in the mixture fraction space are valid. Consequently, the two mixture fractions, $f_{\rm fuel}$ and $f_{\rm sec}$, cannot vary independently; their values are valid only if they are both within the triangle OBC shown in Figure 8.2.1.

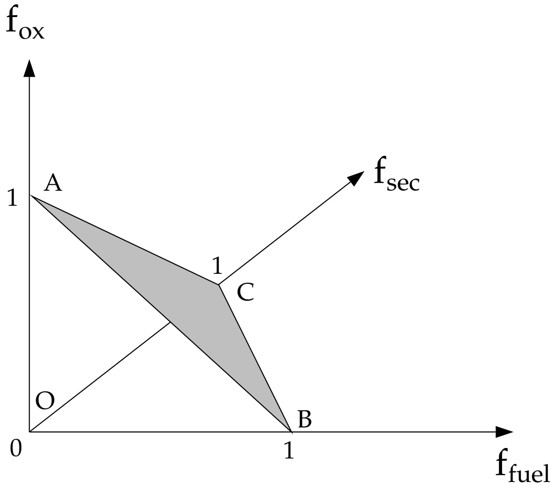


Figure 8.2.1: Relationship of $f_{\rm fuel}$, $f_{\rm sec}$, and $f_{\rm ox}$

ANSYS FLUENT discretizes the triangle OBC as shown in Figure 8.2.2. Essentially, the primary mixture fraction, $f_{\rm fuel}$, is allowed to vary between zero and one, as for the single mixture fraction case, while the secondary mixture fraction lies on lines with the following equation:

$$f_{\rm sec} = p_{\rm sec} \times (1 - f_{\rm fuel}) \tag{8.2-3}$$

where $p_{\rm sec}$ is the normalized secondary mixture fraction and is the value at the intersection of a line with the secondary mixture fraction axis. Note that unlike $f_{\rm sec}$, $p_{\rm sec}$ is bounded between zero and one, regardless of the $f_{\rm fuel}$ value.

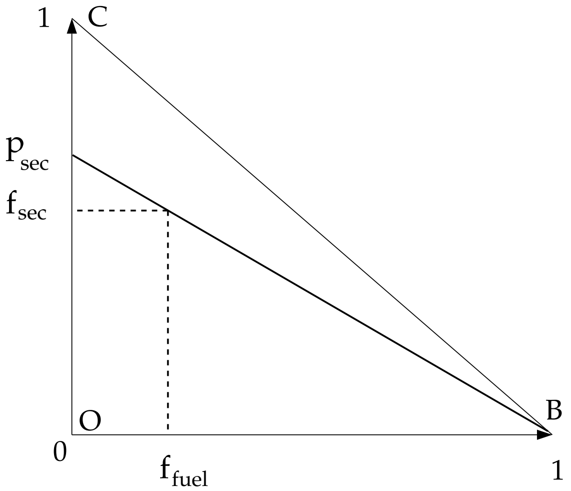


Figure 8.2.2: Relationship of $f_{\rm fuel}$, $f_{\rm sec}$, and $p_{\rm sec}$

An important characteristic of the normalized secondary mixture fraction, $p_{\rm sec}$, is its assumed statistical independence from the fuel mixture fraction, $f_{\rm fuel}$. Note that unlike $f_{\rm sec}$, $p_{\rm sec}$ is not a conserved scalar. This normalized mixture fraction definition, $p_{\rm sec}$, is used everywhere in ANSYS FLUENT when prompted for *Secondary Mixture Fraction* except when defining the rich limit for a secondary fuel stream, which is defined in terms of $f_{\rm sec}$.

Transport Equations for the Mixture Fraction

Under the assumption of equal diffusivities, the species equations can be reduced to a single equation for the mixture fraction, f. The reaction source terms in the species equations cancel (since elements are conserved in chemical reactions), and thus f is a conserved quantity. While the assumption of equal diffusivities is problematic for laminar flows, it is generally acceptable for turbulent flows where turbulent convection overwhelms molecular diffusion. The Favre mean (density-averaged) mixture fraction equation is

$$\frac{\partial}{\partial t}(\rho\overline{f}) + \nabla\cdot(\rho\vec{v}\overline{f}) = \nabla\cdot\left(\frac{\mu_t}{\sigma_t}\nabla\overline{f}\right) + S_m + S_{\rm user} \tag{8.2-4}$$

The source term S_m is due solely to transfer of mass into the gas phase from liquid fuel droplets or reacting particles (e.g., coal). $S_{\rm user}$ is any user-defined source term.

In addition to solving for the Favre mean mixture fraction, ANSYS FLUENT solves a conservation equation for the mixture fraction variance, $\overline{f'^2}$ [152]:

$$\frac{\partial}{\partial t}\left(\rho\overline{f'^2}\right) + \nabla\cdot\left(\rho\vec{v}\overline{f'^2}\right) = \nabla\cdot\left(\frac{\mu_t}{\sigma_t}\nabla\overline{f'^2}\right) + C_g\mu_t\left(\nabla\overline{f}\right)^2 - C_d\rho\frac{\epsilon}{k}\overline{f'^2} + S_{\rm user} \tag{8.2-5}$$

where $f' = f - \overline{f}$. The default values for the constants σ_t, C_g, and C_d are 0.85, 2.86, and 2.0, respectively, and $S_{\rm user}$ is any user-defined source term.

The mixture fraction variance is used in the closure model describing turbulence-chemistry interactions (see Section 8.2.2: Modeling of Turbulence-Chemistry Interaction).

For a two-mixture-fraction problem, $\overline{f_{\rm fuel}}$ and $\overline{f'^2_{\rm fuel}}$ are obtained from Equations 8.2-4 and 8.2-5 by substituting $\overline{f_{\rm fuel}}$ for $\overline{f}$ and $\overline{f'^2_{\rm fuel}}$ for $\overline{f'^2}$. $\overline{f_{\rm sec}}$ is obtained from Equation 8.2-4 by substituting $\overline{f_{\rm sec}}$ for $\overline{f}$. $\overline{p_{\rm sec}}$ is then calculated using Equation 8.2-3, and $\overline{p'^2_{\rm sec}}$ is obtained by solving Equation 8.2-5 with $\overline{p_{\rm sec}}$ substituted for $\overline{f}$. To a first-order approximation, the variances in $\overline{p_{\rm sec}}$ and $\overline{f_{\rm sec}}$ are relatively insensitive to $\overline{f_{\rm fuel}}$, and therefore the equation for $\overline{p'^2_{\rm sec}}$ is essentially the same as $\overline{f'^2_{\rm sec}}$.

> The equation for $\overline{p'^2_{\rm sec}}$ instead of $\overline{f'^2_{\rm sec}}$ is valid when the mass flow rate of the secondary stream is relatively small compared with the total mass flow rate.

The Non-Premixed Model for LES

For Large Eddy Simulations, ȓansport equation is not solved for the mixture fraction variance. Instead, it is modeled as

$$\overline{f'^2} = C_{\rm var} L_{\rm s}^2 |\nabla \overline{f}|^2 \tag{8.2-6}$$

where

$C_{\rm var}$ = constant

$L_{\rm s}$ = subgrid length scale (see Equation 4.11-16)

The constant $C_{\rm var}$ is computed dynamically when the Dynamic Stress option is enabled in the Viscous dialog box, else a constant value (with a default of 0.5) is used.

If the Dynamic Scalar Flux option is enabled, the turbulent Sc (σ_t in Equation 8.2-4) is computed dynamically.

Mixture Fraction vs. Equivalence Ratio

The mixture fraction definition can be understood in relation to common measures of reacting systems. Consider a simple combustion system involving a fuel stream (F), an oxidant stream (O), and a product stream (P) symbolically represented at stoichiometric conditions as

$$\mathrm{F} + r\ \mathrm{O} \rightarrow (1+r)\ \mathrm{P} \tag{8.2-7}$$

where r is the air-to-fuel ratio on a mass basis. Denoting the equivalence ratio as ϕ, where

$$\phi = \frac{(\text{fuel/air})_{\rm actual}}{(\text{fuel/air})_{\rm stoichiometric}} \tag{8.2-8}$$

the reaction in Equation 8.2-7, under more general mixture conditions, can then be written as

$$\phi\ \mathrm{F} + r\ \mathrm{O} \rightarrow (\phi + r)\ \mathrm{P} \tag{8.2-9}$$

Looking at the left side of this equation, the mixture fraction for the system as a whole can then be deduced to be

$$f = \frac{\phi}{\phi + r} \tag{8.2-10}$$

Equation 8.2-10 allows the computation of the mixture fraction at stoichiometric conditions ($\phi = 1$) or at fuel-rich conditions (e.g., $\phi > 1$), or fuel-lean conditions (e.g., $\phi < 1$).

Relationship of f to Species Mass Fraction, Density, and Temperature

The power of the mixture fraction modeling approach is that the chemistry is reduced to one or two conserved mixture fractions. Under the assumption of chemical equilibrium, all thermochemical scalars (species fractions, density, and temperature) are uniquely related to the mixture fraction(s).

For a single mixture fraction in an adiabatic system, the instantaneous values of mass fractions, density, and temperature depend solely on the instantaneous mixture fraction, f:

$$\phi_i = \phi_i(f) \tag{8.2-11}$$

If a secondary stream is included, the instantaneous values will depend on the instantaneous fuel mixture fraction, $f_{\rm fuel}$, and the secondary partial fraction, $p_{\rm sec}$:

$$\phi_i = \phi_i(f_{\rm fuel}, p_{\rm sec}) \tag{8.2-12}$$

In Equations 8.2-11 and 8.2-12, ϕ_i represents the instantaneous species mass fraction, density, or temperature. In the case of non-adiabatic systems, the effect of heat loss/gain is parameterized as

$$\phi_i = \phi_i(f, H) \tag{8.2-13}$$

for a single mixture fraction system, where H is the instantaneous enthalpy (see Equation 5.2-7).

If a secondary stream is included,

$$\phi_i = \phi_i(f_{\rm fuel}, p_{\rm sec}, H) \tag{8.2-14}$$

Examples of non-adiabatic flows include systems with radiation, heat transfer through walls, heat transfer to/from discrete phase particles or droplets, and multiple inlets at different temperatures. Additional detail about the mixture fraction approach in such non-adiabatic systems is provided in Section 8.2.3: Non-Adiabatic Extensions of the Non-Premixed Model.

In many reacting systems, the combustion is not in chemical equilibrium. ANSYS FLUENT offers several approaches to model chemical non-equilibrium, including the finite-rate (see Section 7.1.2: The Generalized Finite-Rate Formulation for Reaction Modeling), EDC (see Section 7.1.2: The Eddy-Dissipation-Concept (EDC) Model), and PDF transport (see Chapter 11: Composition PDF Transport) models, where detailed kinetic mechanisms can be incorporated.

There are three approaches in the non-premixed combustion model to simulate chemical non-equilibrium. The first is to use the Rich Flammability Limit (RFL) option in the Equilibrium model, where rich regions are modeled as a mixed-but-unburnt mixture of pure fuel and a leaner equilibrium burnt mixture (see Section 16.2.5: Enabling the Rich Flammability Limit (RFL) Option in the separate User's Guide). The second approach is the Steady Laminar Flamelet model, where chemical non-equilibrium due to diffusion flame stretching by turbulence can be modeled. The third approach is the Unsteady Laminar Flamelet model where slow-forming product species that are far from chemical equilibrium can be modeled. See Sections 8.4 and 8.6 for details about the Steady and Unsteady Laminar Flamelet models in ANSYS FLUENT.

8.2.2 Modeling of Turbulence-Chemistry Interaction

Equations 8.2-11 through 8.2-14 describe the instantaneous relationships between mixture fraction and species fractions, density, and temperature under the assumption of chemical equilibrium. The ANSYS FLUENT prediction of the turbulent reacting flow, however, is concerned with prediction of the averaged values of these fluctuating scalars. How these averaged values are related to the instantaneous values depends on the turbulence-chemistry interaction model. ANSYS FLUENT applies the assumed-shape probability density function (PDF) approach as its closure model when the non-premixed model is used. The assumed shape PDF closure model is described in this section.

Description of the Probability Density Function

The Probability Density Function, written as $p(f)$, can be thought of as the fraction of time that the fluid spends in the vicinity of the state f. Figure 8.2.3 plots the time trace of mixture fraction at a point in the flow (right-hand side) and the probability density function of f (left-hand side). The fluctuating value of f, plotted on the right side of the figure, spends some fraction of time in the range denoted as Δf. $p(f)$, plotted on the left side of the figure, takes on values such that the area under its curve in the band denoted, Δf, is equal to the fraction of time that f spends in this range. Written mathematically,

$$p(f)\ \Delta f = \lim_{T\rightarrow\infty} \frac{1}{T} \sum_i \tau_i \tag{8.2-15}$$

where T is the time scale and τ_i is the amount of time that f spends in the Δf band. The shape of the function $p(f)$ depends on the nature of the turbulent fluctuations in f. In practice, $p(f)$ is unknown and is modeled as a mathematical function that approximates the actual PDF shapes that have been observed experimentally.

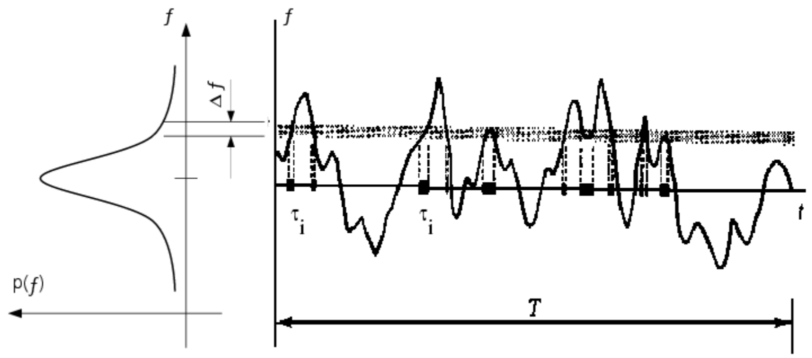


Figure 8.2.3: Graphical Description of the Probability Density Function, $p(f)$

Derivation of Mean Scalar Values from the Instantaneous Mixture Fraction

The probability density function $p(f)$, describing the temporal fluctuations of f in the turbulent flow, can be used to compute averaged values of variables that depend on f. Density-weighted mean species mass fractions and temperature can be computed (in adiabatic systems) as

$$\overline{\phi}_i = \int_0^1 p(f)\phi_i(f)df \tag{8.2-16}$$

for a single-mixture-fraction system. When a secondary stream exists, mean values are calculated as

$$\overline{\phi}_i = \int_0^1 \int_0^1 p_1(f_{\rm fuel})p_2(p_{\rm sec})\phi_i(f_{\rm fuel}, p_{\rm sec})df_{\rm fuel}dp_{\rm sec} \tag{8.2-17}$$

where p_1 is the PDF of $f_{\rm fuel}$ and p_2 is the PDF of $p_{\rm sec}$. Here, statistical independence of $f_{\rm fuel}$ and $p_{\rm sec}$ is assumed, so that $p(f_{\rm fuel}, p_{\rm sec}) = p_1(f_{\rm fuel})p_2(p_{\rm sec})$.

Similarly, the mean time-averaged fluid density, $\overline{\rho}$, can be computed as

$$\frac{1}{\overline{\rho}} = \int_0^1 \frac{p(f)}{\rho(f)} df \tag{8.2-18}$$

for a single-mixture-fraction system, and

$$\frac{1}{\overline{\rho}} = \int_0^1 \int_0^1 \frac{p_1(f_{\rm fuel}) p_2(p_{\rm sec})}{\rho(f_{\rm fuel}, p_{\rm sec})} df_{\rm fuel} dp_{\rm sec} \tag{8.2-19}$$

when a secondary stream exists. $\rho(f)$ or $\rho(f_{\rm fuel}, p_{\rm sec})$ is the instantaneous density obtained using the instantaneous species mass fractions and temperature in the ideal gas law equation.

Using Equations 8.2-16 and 8.2-18 (or Equations 8.2-17 and 8.2-19), it remains only to specify the shape of the function $p(f)$ (or $p_1(f_{\rm fuel})$ and $p_2(p_{\rm sec})$) in order to determine the local mean fluid state at all points in the flow field.

The Assumed-Shape PDF

The shape of the assumed PDF, $p(f)$, is described in ANSYS FLUENT by one of two mathematical functions:

- the double-delta function (two-mixture-fraction cases only)
- the β-function (single- and two-mixture-fraction cases)

The double-delta function is the most easily computed, while the β-function most closely represents experimentally observed PDFs. The shape produced by this function depends solely on the mean mixture fraction, $\overline{f}$, and its variance, $\overline{f'^2}$. A detailed description of each function follows.

The Double Delta Function PDF

The double delta function is given by

$$p(f) = \begin{cases} 0.5, & f = \overline{f} - \sqrt{\overline{f'^2}} \\ 0.5, & f = \overline{f} + \sqrt{\overline{f'^2}} \\ 0, & \text{elsewhere} \end{cases} \tag{8.2-20}$$

with suitable bounding near $f = 1$ and $f = 0$. One example of the double delta function is illustrated in Figure 8.2.4. As noted above, the double delta function PDF is very easy to compute but is invariably less accurate than the alternate β-function PDF because it assumes that only two states occur in the turbulent flow. For this reason, it is available only for two-mixture-fraction simulations where the savings in computational cost is significant.

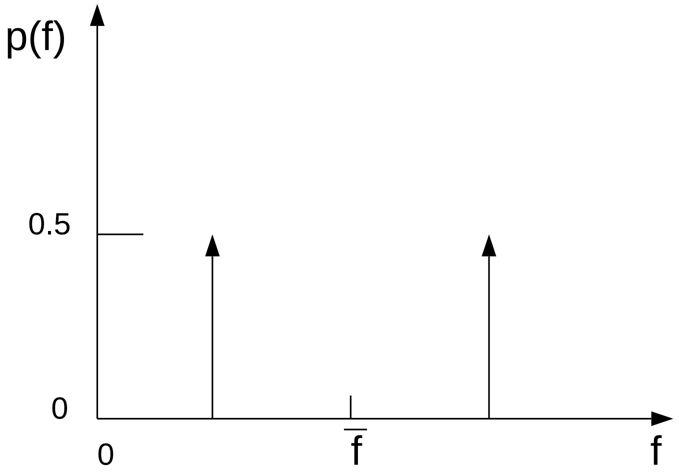


Figure 8.2.4: Example of the Double Delta Function PDF Shape

The β-Function PDF

The β-function PDF shape is given by the following function of $\overline{f}$ and $\overline{f'^2}$:

$$p(f) = \frac{f^{\alpha-1}(1-f)^{\beta-1}}{\int f^{\alpha-1}(1-f)^{\beta-1}df} \tag{8.2-21}$$

where

$$\alpha = \overline{f}\left[\frac{\overline{f}(1-\overline{f})}{\overline{f'^2}} - 1\right] \tag{8.2-22}$$

and

$$\beta = (1-\overline{f})\left[\frac{\overline{f}(1-\overline{f})}{\overline{f'^2}} - 1\right] \tag{8.2-23}$$

Importantly, the PDF shape $p(f)$ is a function of only its first two moments, namely the mean mixture fraction, $\overline{f}$, and the mixture fraction variance, $\overline{f'^2}$. Thus, given ANSYS FLUENT's prediction of $\overline{f}$ and $\overline{f'^2}$ at each point in the flow field (Equations 8.2-4 and 8.2-5), the assumed PDF shape can be computed and used as the weighting function to determine the mean values of species mass fractions, density, and temperature using, Equations 8.2-16 and 8.2-18 (or, for a system with a secondary stream, Equations 8.2-17 and 8.2-19).

This logical dependence is depicted visually in Figure 8.2.5 for a single mixture fraction.

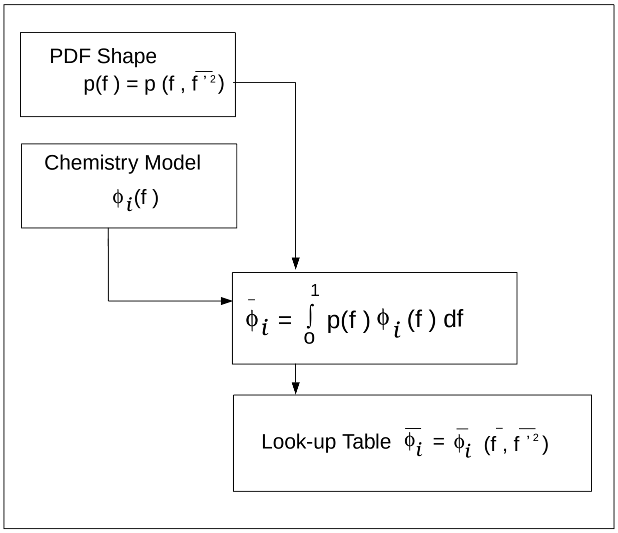

PDF Shape
p(f) = p (f , $\overline{f'^2}$)
Chemistry Model
ϕ_i(f)
$\overline{\phi}_i = \int_0^1$ p(f) ϕ_i (f) df
Look-up Table $\overline{\phi_i}$ = $\overline{\phi_i}$ ($\overline{f}$, $\overline{f'^2}$)

Figure 8.2.5: Logical Dependence of Averaged Scalars $\overline{\phi_i}$ on $\overline{f}$, $\overline{f'^2}$, and the Chemistry Model (Adiabatic, Single-Mixture-Fraction Systems)

8.2.3 Non-Adiabatic Extensions of the Non-Premixed Model

Many reacting systems involve heat transfer through wall boundaries, droplets, and/or particles. In such flows the local thermochemical state is no longer related only to f, but also to the enthalpy, H. The system enthalpy impacts the chemical equilibrium calculation and the temperature and species of the reacting flow. Consequently, changes in enthalpy due to heat loss must be considered when computing scalars from the mixture fraction, as in Equation 8.2-13.

In such non-adiabatic systems, turbulent fluctuations should be accounted for by means of a joint PDF, $p(f, H)$. The computation of $p(f, H)$, however, is not practical for most engineering applications. The problem can be simplified significantly by assuming that the enthalpy fluctuations are independent of the enthalpy level (i.e., heat losses do not significantly impact the turbulent enthalpy fluctuations). With this assumption, $p(f, H) = p(f)\delta(H - \overline{H})$ and mean scalars are calculated as

$$\overline{\phi_i} = \int_0^1 \phi_i(f, \overline{H})p(f)df \qquad (8.2\text{-}24)$$

Determination of $\overline{\phi_i}$ in the non-adiabatic system thus requires solution of the modeled transport equation for mean enthalpy:

$$\frac{\partial}{\partial t}(\rho \overline{H}) + \nabla \cdot (\rho \vec{v} \overline{H}) = \nabla \cdot \left(\frac{k_t}{c_p} \nabla \overline{H} \right) + S_h \tag{8.2-25}$$

where S_h accounts for source terms due to radiation, heat transfer to wall boundaries, and heat exchange with the dispersed phase.

Figure 8.2.6 depicts the logical dependence of mean scalar values (species mass fraction, density, and temperature) on ANSYS FLUENT's prediction of $\overline{f}$, $\overline{f'^2}$, and $\overline{H}$ in non-adiabatic single-mixture-fraction systems.

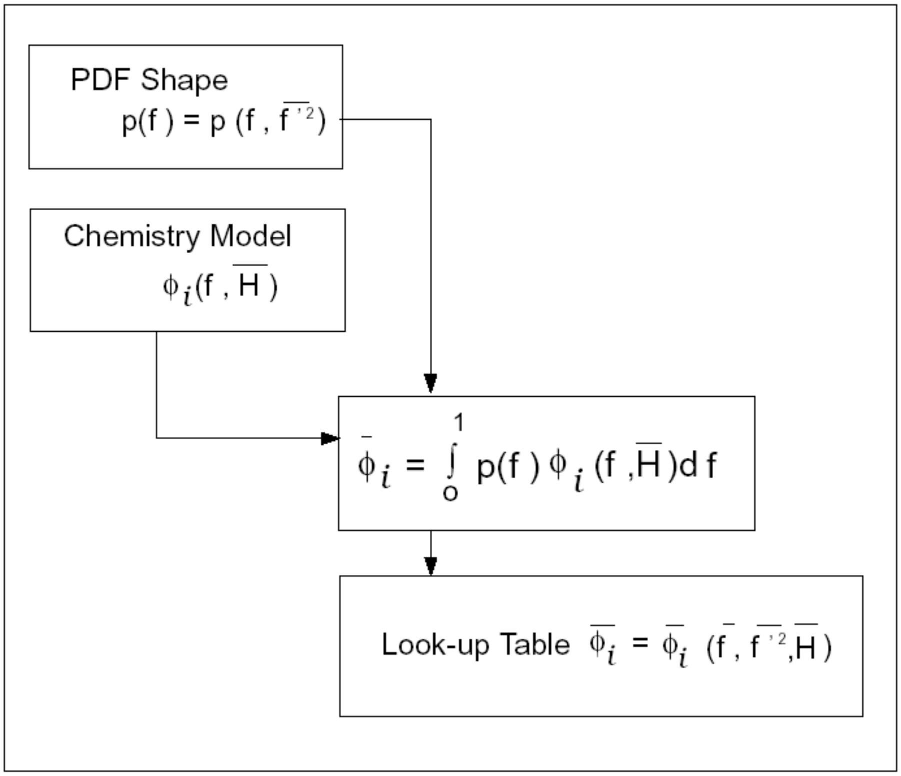
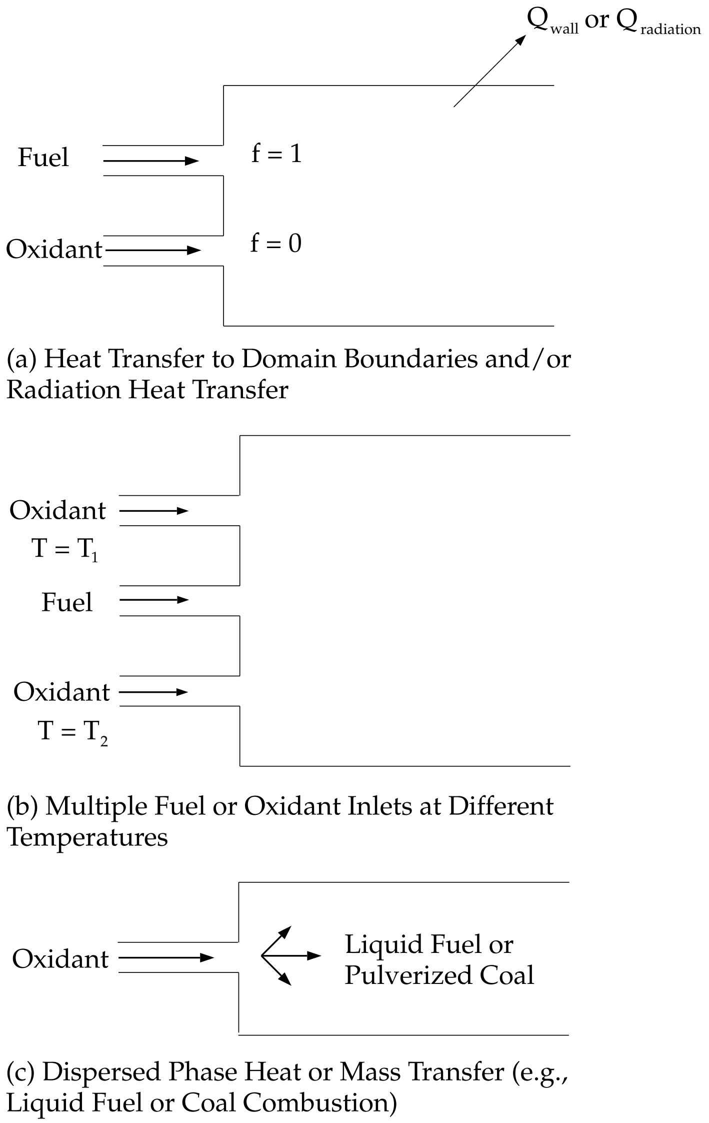

PDF Shape
p(f) = p (f , f '2)
Chemistry Model
φi(f , H)
φi = ∫0^1 p(f) φi (f ,H)d f
Look-up Table φi = φi (f , f '2,H)

Figure 8.2.6: Logical Dependence of Averaged Scalars $\overline{\phi_i}$ on $\overline{f}$, $\overline{f'^2}$, $\overline{H}$, and the Chemistry Model (Non-Adiabatic, Single-Mixture-Fraction Systems)

When a secondary stream is included, the mean values are calculated from

$$\overline{\phi_i} = \int_0^1 \int_0^1 \phi_i(f_{\rm fuel}, p_{\rm sec}, \overline{H}) p_1(f_{\rm fuel}) p_2(p_{\rm sec}) df_{\rm fuel} dp_{\rm sec} \tag{8.2-26}$$

As noted above, the non-adiabatic extensions to the PDF model are required in systems involving heat transfer to walls and in systems with radiation included. In addition, the non-adiabatic model is required in systems that include multiple fuel or oxidizer inlets with different inlet temperatures. Finally, the non-adiabatic model is required in particle-laden flows (e.g., liquid fuel systems or coal combustion systems) when such flows include heat transfer to the dispersed phase. Figure 8.2.7 illustrates several systems that must include the non-adiabatic form of the PDF model.



(a) Heat Transfer to Domain Boundaries and/or Radiation Heat Transfer

(b) Multiple Fuel or Oxidant Inlets at Different Temperatures

(c) Dispersed Phase Heat or Mass Transfer (e.g., Liquid Fuel or Coal Combustion)

Figure 8.2.7: Reacting Systems Requiring Non-Adiabatic Non-Premixed Model Approach

8.2.4 Chemistry Tabulation

Look-Up Tables for Adiabatic Systems

For an equilibrium, adiabatic, single-mixture-fraction case, the mean temperature, density, and species fraction are functions of the $\overline{f}$ and $\overline{f'^2}$ only (see Equations 8.2-16 and 8.2-21). Significant computational time can be saved by computing these integrals once, storing them in a look-up table, and retrieving them during the ANSYS FLUENT simulation.

Figure 8.2.8 illustrates the concept of the look-up tables generated for a single-mixture-fraction system. Given ANSYS FLUENT's predicted value for $\overline{f}$ and $\overline{f'^2}$ at a point in the flow domain, the mean value of mass fractions, density, or temperature ($\overline{\phi_i}$) at that point can be obtained by table interpolation.

The table, Figure 8.2.8, is the mathematical result of the integration of Equation 8.2-16. There is one look-up table of this type for each scalar of interest (species mass fractions, density, and temperature). In adiabatic systems, where the instantaneous enthalpy is a function of only the instantaneous mixture fraction, a two-dimensional look-up table, like that in Figure 8.2.8, is all that is required.

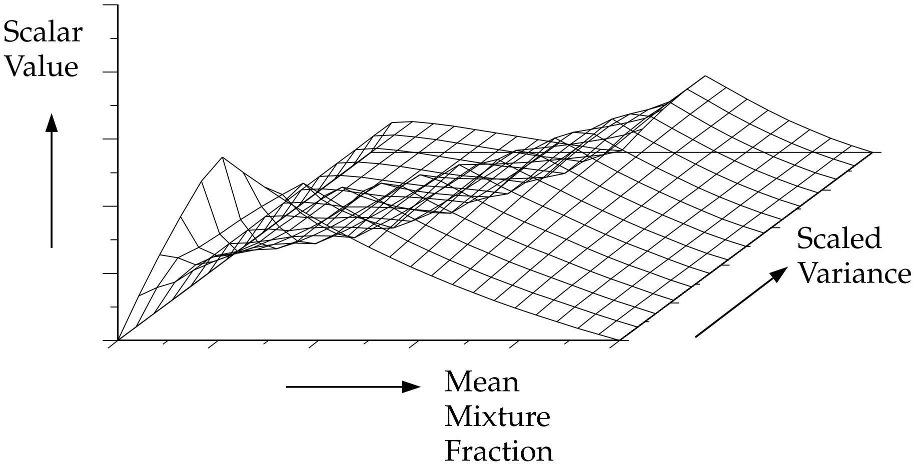


Figure 8.2.8: Visual Representation of a Look-Up Table for the Scalar $\overline{\phi_i}$ as a Function of $\overline{f}$ and $\overline{f'^2}$ in Adiabatic Single-Mixture-Fraction Systems

For systems with two mixture fractions, the creation and interpolation costs of four-dimensional look-up tables are computationally expensive. By default, the instantaneous properties ϕ_i are tabulated as a function of the fuel mixture fraction $f_{\rm fuel}$ and the secondary partial fraction $p_{\rm sec}$ (see Equation 8.2-12), and the PDF integrations (see Equation 8.2-14) are performed at run-time. This two-dimensional table is illustrated in Figure 8.2.9. Alternatively, 4D look-up tables can be created before the simulation and interpolated at run time (see Section 16.7.1: Full Tabulation of the Two-Mixture-Fraction Model in the separate User's Guide).

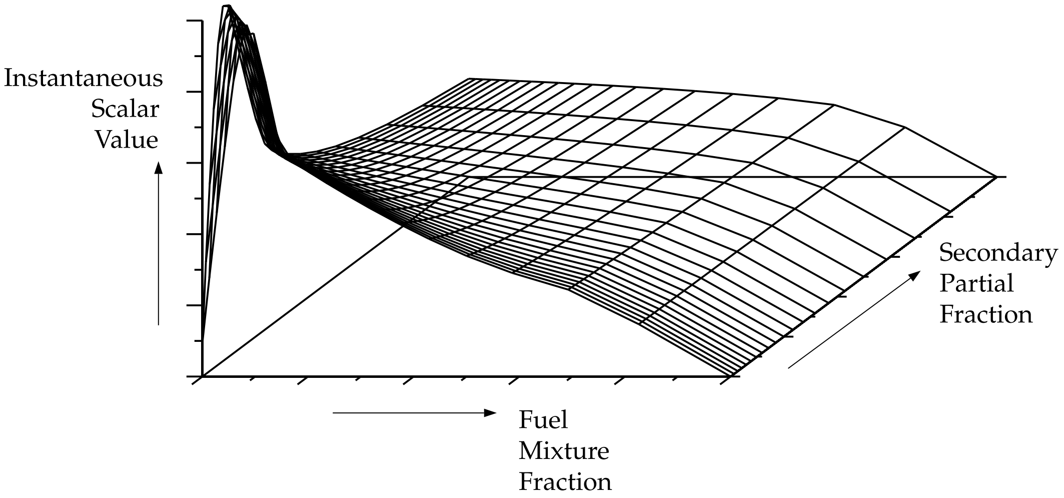


Figure 8.2.9: Visual Representation of a Look-Up Table for the Scalar ϕ_i as a Function of $f_{\rm fuel}$ and $p_{\rm sec}$ in Adiabatic Two-Mixture-Fraction Systems

3D Look-Up Tables for Non-Adiabatic Systems

In non-adiabatic systems, where the enthalpy is not linearly related to the mixture fraction, but depends also on wall heat transfer and/or radiation, a look-up table is required for each possible enthalpy value in the system. The result, for single mixture fraction systems, is a three-dimensional look-up table, as illustrated in Figure 8.2.10, which consists of layers of two-dimensional tables, each one corresponding to a normalized heat loss or gain. The first slice corresponds to the maximum heat loss from the system, the last slice corresponds to the maximum heat gain to the system, and the zero heat loss/gain slice corresponds to the adiabatic table. Slices interpolated between the adiabatic and maximum slices correspond to heat gain, and those interpolated between the adiabatic and minimum slices correspond to heat loss.

The three-dimensional look-up table allows ANSYS FLUENT to determine the value of each mass fraction, density, and temperature from calculated values of $\overline{f}$, $\overline{f'^2}$, and $\overline{H}$. This three-dimensional table in Figure 8.2.10 is the visual representation of the integral in Equation 8.2-24.

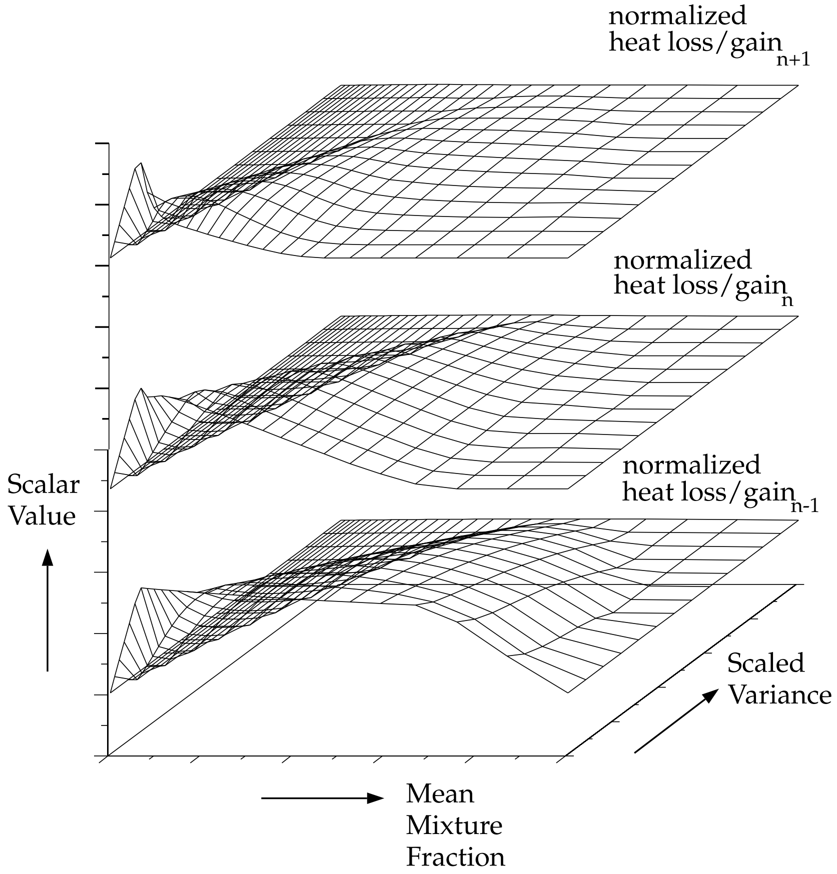


Figure 8.2.10: Visual Representation of a Look-Up Table for the Scalar $\overline{\phi_i}$ as a Function of $\overline{f}$ and $\overline{f'^2}$ and Normalized Heat Loss/Gain in Non-Adiabatic Single-Mixture-Fraction Systems

For non-adiabatic, two-mixture-fraction problems, it is very expensive to tabulate and retrieve Equation 8.2-26 since five-dimensional tables are required. By default, 3D look-up tables of the instantaneous state relationship given by Equation 8.2-14 are created. The 3D table in Figure 8.2.11 is the visual representation of Equation 8.2-14. The mean density during the ANSYS FLUENT solution is calculated by integrating the instantaneous density over the fuel and secondary mixture fraction space (see Equation 8.2-26). Alternatively, 5D look-up tables can be created before the simulation and interpolated at run time (see Section 16.7.1: Full Tabulation of the Two-Mixture-Fraction Model in the separate User's Guide). The one-time pre-generation the 5D look-up table is very expensive, but, once built, interpolating the table during ANSYS FLUENT solution is usually significantly faster than performing the integrations at run-time. This is especially true for cases with a large number of cells that require many iteration or time-steps to converge.

Note that the computation time in ANSYS FLUENT for a two-mixture-fraction case will be much greater than for a single-mixture-fraction problem. This expense should be carefully considered before choosing the two-mixture-fraction model. Also, it is usually expedient to start a two-mixture-fraction simulation from a converged single-mixture-fraction solution.

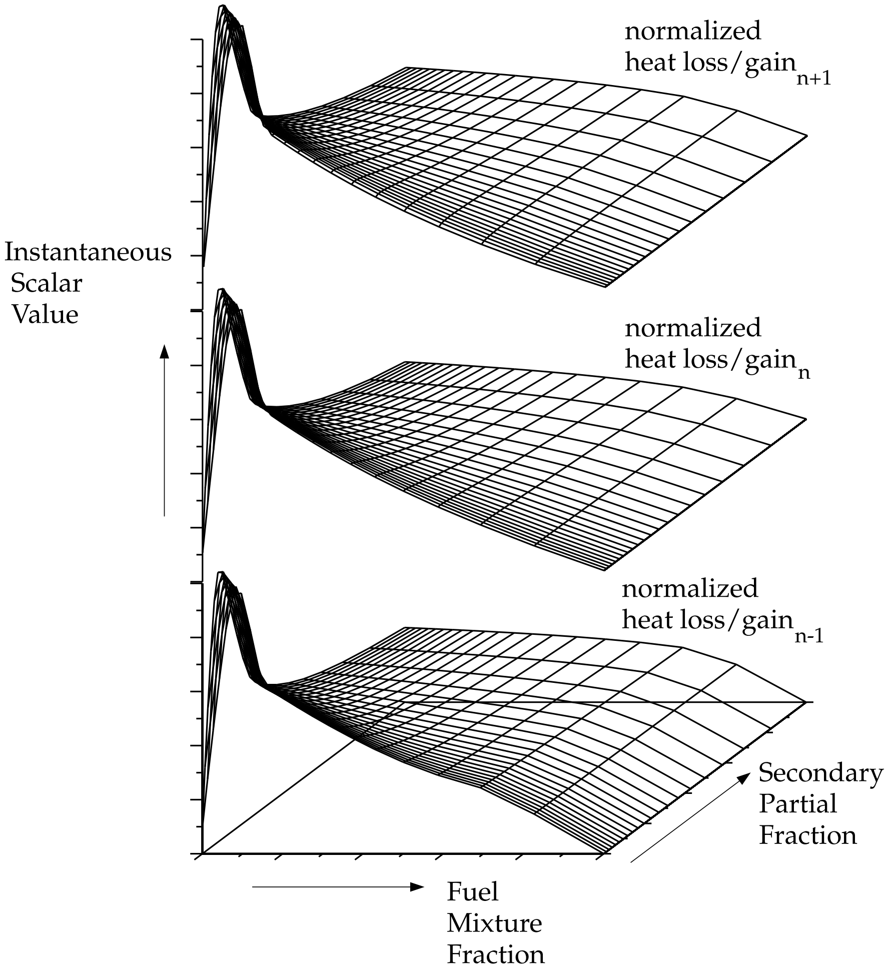


Figure 8.2.11: Visual Representation of a Look-Up Table for the Scalar ϕ_i as a Function of $f_{\rm fuel}$, $p_{\rm sec}$, and Normalized Heat Loss/Gain in Non-Adiabatic Two-Mixture-Fraction Systems

8.3 Restrictions and Special Cases for Using the Non-Premixed Model

8.3.1 Restrictions on the Mixture Fraction Approach

The unique dependence of ϕ_i (species mass fractions, density, or temperature) on f (Equation 8.2-11 or 8.2-13) requires that the reacting system meet the following conditions:

- The chemical system must be of the diffusion type with discrete fuel and oxidizer inlets (spray combustion and pulverized fuel flames may also fall into this category).
- The Lewis number must be unity. (This implies that the diffusion coefficients for all species and enthalpy are equal, a good approximation in turbulent flow).
- When a single mixture fraction is used, the following conditions must be met:
  - Only one type of fuel is involved. The fuel may be made up of a burnt mixture of reacting species (e.g., 90% CH_4 and 10% CO) and you may include multiple fuel inlets. The multiple fuel inlets must have the same composition; two or more fuel inlets with different fuel composition are not allowed (e.g., one inlet of CH_4 and one inlet of CO). Similarly, in spray combustion systems or in systems involving reacting particles, only one off-gas is permitted.
  - Only one type of oxidizer is involved. The oxidizer may consist of a mixture of species (e.g., 21% O_2 and 79% N_2) and you may have multiple oxidizer inlets. The multiple oxidizer inlets must, however, have the same composition. Two or more oxidizer inlets with different compositions are not allowed (e.g., one inlet of air and a second inlet of pure oxygen).
- When two mixture fractions are used, three streams can be involved in the system. Valid systems are as follows:
  - Two fuel streams with different compositions and one oxidizer stream. Each fuel stream may be made up of a mixture of reacting species (e.g., 90% CH_4 and 10% CO). You may include multiple inlets of each fuel stream, but each fuel inlet must have one of the two defined compositions (e.g., one inlet of CH_4 and one inlet of CO).
  - Mixed fuel systems including gas-liquid, gas-coal, or liquid-coal fuel mixtures with a single oxidizer. In systems with a gas-coal or liquid-coal fuel mixture, the coal volatiles and char can be treated as a single composite fuel stream and the secondary stream can represent another fuel. Alternatively, for coal combustion, the volatile and char off-gases are tracked separately as distinct fuel streams.

  - Two oxidizer streams with different compositions and one fuel stream. Each oxidizer stream may consist of a mixture of species (e.g. 21% O_2 and 79% N_2). You may have multiple inlets of each oxidizer stream, but each oxidizer inlet must have one of the two defined compositions (e.g., one inlet of air and a second inlet of pure oxygen).
  - A fuel stream, an oxidizer stream, and a non-reacting secondary stream.

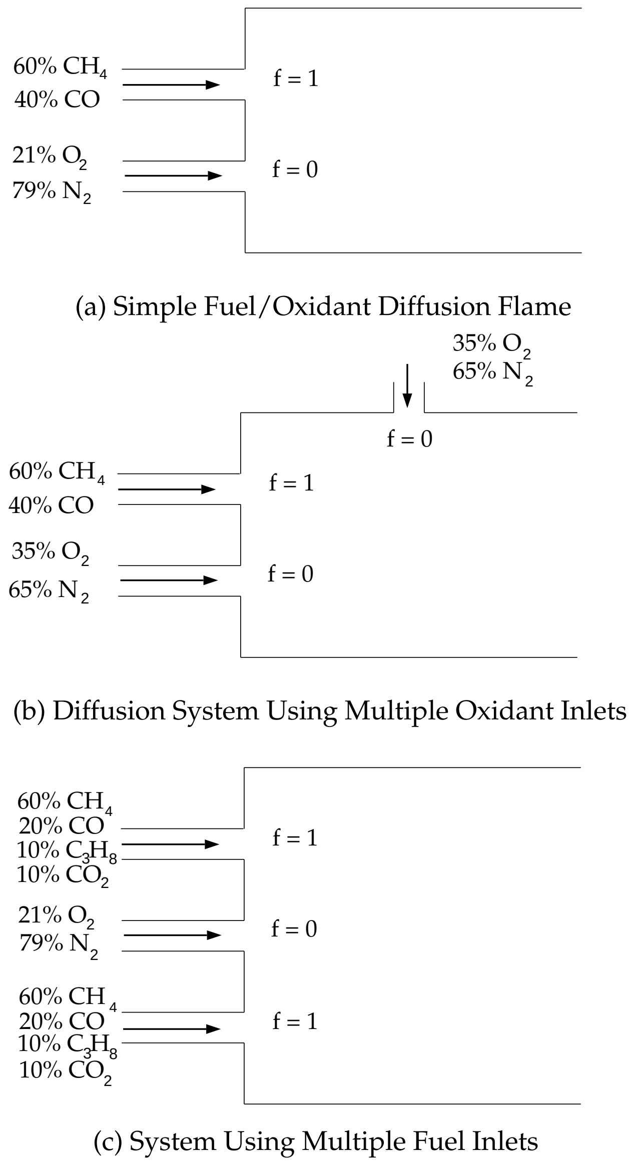
- The flow must be turbulent.

It is important to emphasize that these restrictions eliminate the use of the non-premixed approach for directly modeling premixed combustion. This is because the unburned premixed stream is far from chemical equilibrium. Note, however, that an extended mixture fraction formulation, the partially premixed model (see Chapter 10: Partially Premixed Combustion), can be applied to non-premixed (with mixed-but-unburnt regions), as well as partially premixed flames.

Figures 8.3.1 and 8.3.2 illustrate typical reacting system configurations that can be handled by the non-premixed model in ANSYS FLUENT. Figure 8.3.3 shows a premixed configuration that cannot be modeled using the non-premixed model.

8.3.2 Using the Non-Premixed Model for Liquid Fuel or Coal Combustion

You can use the non-premixed model if your ANSYS FLUENT simulation includes liquid droplets and/or coal particles. In this case, fuel enters the gas phase within the computational domain at a rate determined by the evaporation, devolatilization, and char combustion laws governing the dispersed phase. In the case of coal, the volatiles and the products of char can be defined as two different types of fuel (using two mixture fractions) or as a single composite off-gas (using one mixture fraction), as described in Section 16.4.5: Modeling Coal Combustion Using the Non-Premixed Model in the separate User's Guide.



(a) Simple Fuel/Oxidant Diffusion Flame

(b) Diffusion System Using Multiple Oxidant Inlets

(c) System Using Multiple Fuel Inlets

Figure 8.3.1: Chemical Systems That Can Be Modeled Using a Single Mixture Fraction

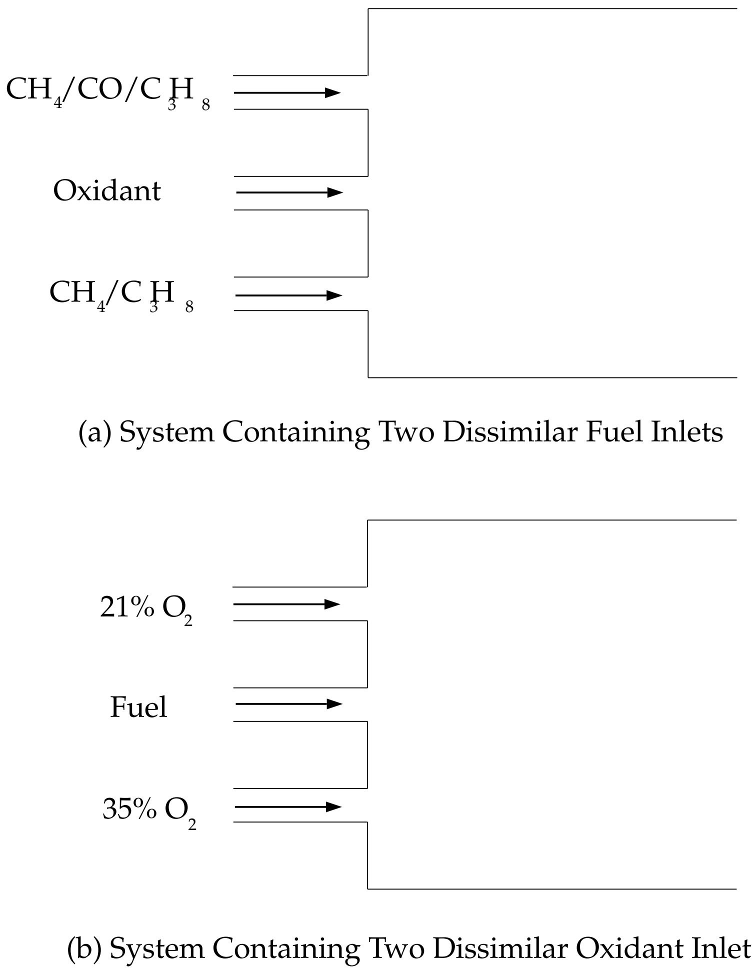

CH_4/CO/C_3H_8
Oxidant
CH_4/C_3H_8
(a) System Containing Two Dissimilar Fuel Inlets
21% O_2
Fuel
35% O_2
(b) System Containing Two Dissimilar Oxidant Inlets

Figure 8.3.2: Chemical System Configurations That Can Be Modeled Using Two Mixture Fractions

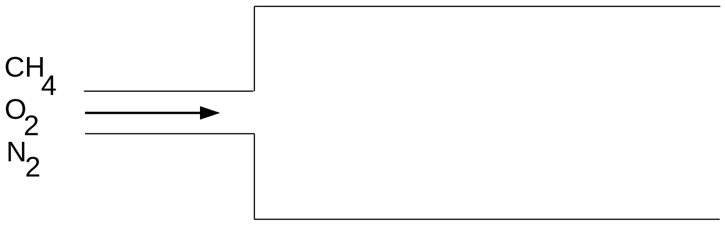


Figure 8.3.3: Premixed Systems That Cannot Be Modeled Using the Non-Premixed Model

8.3.3 Using the Non-Premixed Model with Flue Gas Recycle

While most problems you solve using the non-premixed model will involve inlets that contain either pure oxidant or pure fuel ($f = 0$ or 1), you can include an inlet that has an intermediate value of mixture fraction ($0 < f < 1$) provided that this inlet represents a completely reacted mixture. Such cases arise when there is flue gas recirculation, as depicted schematically in Figure 8.3.4. Since f is a conserved quantity, the mixture fraction at the flue gas recycle inlet can be computed as

$$\dot{m}_{\rm fuel} + \dot{m}_{\rm recyc} f_{\rm exit} = (\dot{m}_{\rm fuel} + \dot{m}_{\rm ox} + \dot{m}_{\rm recyc}) f_{\rm exit} \tag{8.3-1}$$

or

$$f_{\rm exit} = \frac{\dot{m}_{\rm fuel}}{\dot{m}_{\rm fuel} + \dot{m}_{\rm ox}} \tag{8.3-2}$$

where $f_{\rm exit}$ is the exit mixture fraction (and the mixture fraction at the flue gas recycle inlet), $\dot{m}_{\rm ox}$ is the mass flow rate of the oxidizer inlet, $\dot{m}_{\rm fuel}$ is the mass flow rate of the fuel inlet, $\dot{m}_{\rm recyc}$ is the mass flow rate of the recycle inlet.

If a secondary stream is included,

$$f_{\rm fuel,exit} = \frac{\dot{m}_{\rm fuel}}{\dot{m}_{\rm fuel} \ + \dot{m}_{\rm sec} + \dot{m}_{\rm ox}} \tag{8.3-3}$$

and

$$p_{\rm sec,exit} = \frac{\dot{m}_{\rm sec}}{\dot{m}_{\rm sec} + \dot{m}_{\rm ox}} \tag{8.3-4}$$

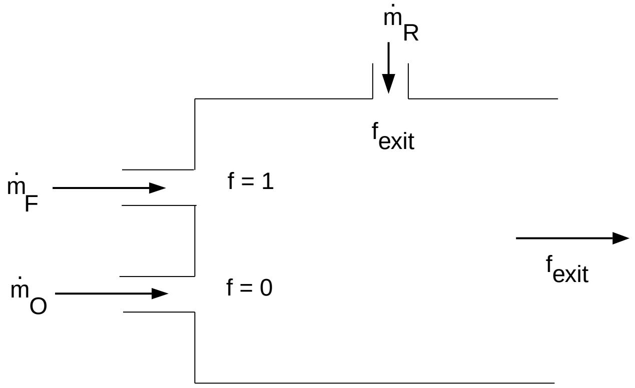


Figure 8.3.4: Using the Non-Premixed Model with Flue Gas Recycle

8.3.4 Using the Non-Premixed Model with the Inert Model

To model the effect of dilution on combustion without the expense of using two mixture fractions, ANSYS FLUENT allows the introduction of an inert stream into the domain. Unlike a secondary mixture fraction, the inert does not chemically equilibrate with the primary fuel and oxidizer - instead, its composition remains constant after mixing. However the inert stream does affect the solution due to its influence on enthalpy, specific heat, and density of the mixture. The equation for conservation of inert is written as:

$$\frac{\partial \rho Y_I}{\partial t} + \nabla \cdot (\vec{v}\rho Y_I) = \nabla \cdot \left(\frac{\mu_t}{\mathrm{Sc}_t} \nabla \left(Y_I \right) \right) \tag{8.3-5}$$

where

Y_I = inert stream tracer
Sc_t = turbulent Schmidt number
μ_t = turbulent viscosity
ρ = density

Equation 8.3-5 has no sources or sinks, because the problem is reduced to tracking a conserved scalar when it is assumed that the inert components have the same turbulent diffusivities.

Mixture Composition

The mixture properties are computed from the mean ($\bar{f}$) and variance ($\bar{f}'$) of the mixture fraction in the cell, the reaction progress variable (c, when the partially premixed model is enabled), the cell enthalpy (H, for non-adiabatic flows), and the inert tracer (Y_I). The mixture is modeled as a blend of inert and active species, but the PDF tables need to be accessed with *conditioned* variables. Conditioning is necessary to take into account the volume taken up by the inert fraction, yet still be able to utilize previously built tables by straightforward lookup. The mean mixture fraction and mixture fraction variance used to access the PDF table is given by:

$$\bar{f}_I = \frac{\bar{f}}{1 - Y_I} \tag{8.3-6}$$

$$\bar{f}_I'^2 = \frac{\bar{f}'^2}{(1 - Y_I)^2} \tag{8.3-7}$$

The reaction progress variable c is not conditioned, however the cell enthalpy must be conditioned to account for the inert enthalpy. The inert enthalpy and active enthalpy are obtained from the following relationships:

$$H(T) = (1 - Y_I)H^{pdf}(T) + Y_I H^I(T) \tag{8.3-8}$$

where H is the enthalpy of the cell at temperature T, H^{pdf} is the enthalpy of the active mixture-fraction stream and H^I is the enthalpy of the inert stream. Here it is assumed that the inert and the active streams have the same temperature, but different enthalpies. To calculate the temperature in the cell, Equation 8.3-8 is solved for the temperature and for H^{pdf}, which gives the partitioning of the energy between the inert and active streams.

The inert enthalpy is defined as

$$H^I(T) = \sum_{i=0}^{N_I} Y_i \int_{T_0}^{T} C_{p_i} dT \tag{8.3-9}$$

where Y_i refers to the mass fraction of specie i defined in the inert stream, T_0 is the reference temperature, C_{p_i} the specific heat of specie i, and N_I is the number of inert species.

The inert and PDF enthalpies are defined further in Equation 31.4-13 in the separate User's Guide.

Property Evaluation

The specific heat of the mixture is evaluated by mixing the inert and active streams in the following way:

$$C_p(T) = (1 - Y_I)C_p^{pdf}(T) + Y_I C_p^I(T) \tag{8.3-10}$$

The density of the mixture is calculated by using a harmonic average of the densities of the active and inert streams, weighted by the inert tracer:

$$\rho(T) = \left[\frac{(1 - Y_I)}{\rho^{pdf}(T)} + \frac{Y_I}{\rho^I(T)}\right]^{-1} \tag{8.3-11}$$

Here, the inert density (ρ^I) is calculated from the ideal gas law.

For information on how to set up the inert model, see Section 16.7.5: Setting Up the Inert Model in the separate User's Guide.

8.4 The Laminar Flamelet Models Theory

Information about the flamelet models are presented in the following sections:

- Section 8.4.1: Restrictions and Assumptions
- Section 8.4.2: The Flamelet Concept
- Section 8.4.3: Flamelet Generation
- Section 8.4.4: Flamelet Import

8.4.1 Restrictions and Assumptions

The following restrictions apply to all flamelet models in ANSYS FLUENT:

- Only a single mixture fraction can be modeled; two-mixture-fraction flamelet models are not allowed.
- The mixture fraction is assumed to follow the β-function PDF, and scalar dissipation fluctuations are ignored.
- Empirically-based streams cannot be used with the flamelet model.

8.4.2 The Flamelet Concept

Overview

The flamelet concept views the turbulent flame as an ensemble of thin, laminar, locally one-dimensional flamelet structures embedded within the turbulent flow field [36, 267, 268] (see Figure 8.4.1).

A common laminar flame type used to represent a flamelet in a turbulent flow is the counterflow diffusion flame. This geometry consists of opposed, axisymmetric fuel and oxidizer jets. As the distance between the jets is decreased and/or the velocity of the jets increased, the flame is strained and increasingly departs from chemical equilibrium until it is eventually extinguished. The species mass fraction and temperature fields can be measured in laminar counterflow diffusion flame experiments, or, most commonly, calculated. For the latter, a self-similar solution exists, and the governing equations can be simplified to one dimension along the axis of the fuel and oxidizer jets, where complex chemistry calculations can be affordably performed.

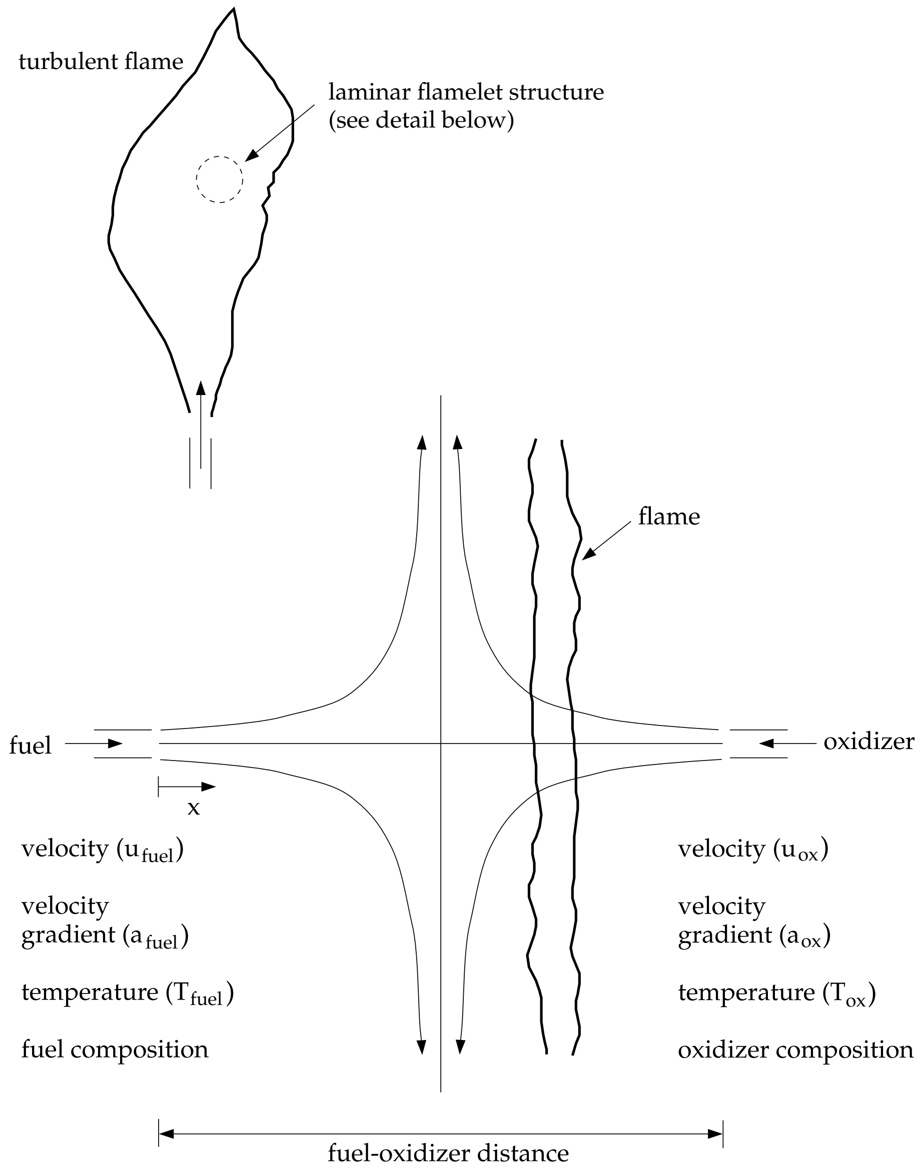


Figure 8.4.1: Laminar Opposed-Flow Diffusion Flamelet

In the laminar counterflow flame, the mixture fraction, f, (see Section 8.2.1: Definition of the Mixture Fraction for definition) decreases monotonically from unity at the fuel jet to zero at the oxidizer jet. If the species mass fraction and temperature along the axis are mapped from physical space to mixture fraction space, they can be uniquely described by two parameters: the mixture fraction and the strain rate (or, equivalently, the scalar dissipation, χ, defined in Equation 8.4-2). Hence, the chemistry is reduced and completely described by the two quantities, f and χ.

This reduction of the complex chemistry to two variables allows the flamelet calculations to be preprocessed, and stored in look-up tables. By preprocessing the chemistry, computational costs are reduced considerably.

The balance equations, solution methods, and sample calculations of the counterflow laminar diffusion flame can be found in several references. Comprehensive reviews and analyses are presented in the works of Bray and Peters, and Dixon-Lewis [36, 72].

Strain Rate and Scalar Dissipation

A characteristic strain rate for a counterflow diffusion flamelet can be defined as $a_s = v/2d$, where v is the relative speed of the fuel and oxidizer jets, and d is the distance between the jet nozzles.

Instead of using the strain rate to quantify the departure from equilibrium, it is expedient to use the scalar dissipation, denoted by χ. The scalar dissipation is defined as

$$\chi = 2D|\nabla f|^2 \tag{8.4-1}$$

where D is a representative diffusion coefficient.

Note that the scalar dissipation, χ, varies along the axis of the flamelet. For the counterflow geometry, the flamelet strain rate a_s can be related to the scalar dissipation at the position where f is stoichiometric by [267]:

$$\chi_{\rm st} = \frac{a_s \exp\left(-2[\mathrm{erfc}^{-1}(2f_{\rm st})]^2\right)}{\pi} \tag{8.4-2}$$

where

$\chi_{\rm st}$	$=$ scalar dissipation at $f = f_{\rm st}$
a_s	$=$ characteristic strain rate
$f_{\rm st}$	$=$ stoichiometric mixture fraction
erfc^{-1}	$=$ inverse complementary error function

Physically, as the flame is strained, the width of the reaction zone diminishes, and the gradient of f at the stoichiometric position $f = f_{\rm st}$ increases. The instantaneous stoichiometric scalar dissipation, $\chi_{\rm st}$, is used as the essential non-equilibrium parameter. It has the dimensions $\rm s^{-1}$ and may be interpreted as the inverse of a characteristic diffusion time. In the limit $\chi_{\rm st} \rightarrow 0$ the chemistry tends to equilibrium, and as $\chi_{\rm st}$ increases due to aerodynamic straining, the non-equilibrium increases. Local quenching of the flamelet occurs when $\chi_{\rm st}$ exceeds a critical value.

Embedding Laminar Flamelets in Turbulent Flames

A turbulent flame brush is modeled as an ensemble of discrete laminar flamelets. Since, for adiabatic systems, the species mass fraction and temperature in the laminar flamelets are completely parameterized by f and $\chi_{\rm st}$, density-weighted mean species mass fractions and temperature in the turbulent flame can be determined from the PDF of f and $\chi_{\rm st}$ as

$$\overline{\phi} = \int \int \phi(f, \chi_{\rm st}) p(f, \chi_{\rm st}) \; df \; d\chi_{\rm st} \tag{8.4-3}$$

where ϕ represents species mass fractions and temperature.

In ANSYS FLUENT, f and $\chi_{\rm st}$ are assumed to be statistically independent, so the joint PDF $p(f, \chi_{\rm st})$ can be simplified as $p_f(f)p_\chi(\chi_{\rm st})$. A β PDF shape is assumed for p_f, and transport equations for $\overline{f}$ and $\overline{f'^2}$ are solved in ANSYS FLUENT to specify p_f. Fluctuations in $\chi_{\rm st}$ are ignored so that the PDF of χ is a delta function: $p_\chi = \delta(\chi - \overline{\chi})$. The first moment, namely the mean scalar dissipation, $\overline{\chi_{\rm st}}$, is modeled in ANSYS FLUENT as

$$\overline{\chi_{\rm st}} = \frac{C_\chi \epsilon \overline{f'^2}}{k} \tag{8.4-4}$$

where C_χ is a constant with a default value of 2.

For LES, the mean scalar dissipation is modeled as

$$\overline{\chi_{\rm st}} = C_\chi \frac{(\mu_t + \mu)}{\rho \sigma_t} |\nabla \overline{f}|^2 \tag{8.4-5}$$

To avoid the PDF convolutions at ANSYS FLUENT run-time, the integrations in Equation 8.4-3 are preprocessed and stored in look-up tables. For adiabatic flows, flamelet tables have three dimensions: $\overline{f}$, $\overline{f'^2}$ and $\overline{\chi_{\rm st}}$.

For non-adiabatic steady laminar flamelets, the additional parameter of enthalpy is required. However, the computational cost of modeling steady flamelets over a range of enthalpies is prohibitive, so some approximations are made. Heat gain/loss to the system is assumed to have a negligible effect on the species mass fractions, and adiabatic mass fractions are used [27, 240]. The temperature is then calculated from Equation 5.2-7 for a range of mean enthalpy gain/loss, $\overline{H}$. Accordingly, mean temperature and density PDF tables have an extra dimension of mean enthalpy. The approximation of constant adiabatic species mass fractions is, however, not applied for the case corresponding to a scalar dissipation of zero. Such a case is represented by the non-adiabatic equilibrium solution. For $\overline{\chi_{\rm st}} = 0$, the species mass fractions are computed as functions of $\overline{f}$, $\overline{f'^2}$, and $\overline{H}$.

In ANSYS FLUENT, you can either generate your own flamelets, or import them as flamelet files calculated with other stand-alone packages. Such stand-alone codes include OPPDIF [210], CFX-RIF [15, 16, 272] and RUN-1DL [270]. ANSYS FLUENT can import flamelet files in OPPDIF format or standard flamelet file format.

Instructions for generating and importing flamelets are provided in Section 8.4.3: Flamelet Generation and Section 8.4.4: Flamelet Import.

8.4.3 Flamelet Generation

The laminar counterflow diffusion flame equations can be transformed from physical space (with x as the independent variable) to mixture fraction space (with f as the independent variable) [273]. In ANSYS FLUENT, a simplified set of the mixture fraction space equations are solved [272]. Here, N equations are solved for the species mass fractions, Y_i,

$$\rho \frac{\partial Y_i}{\partial t} = \frac{1}{2}\rho\chi \frac{\partial^2 Y_i}{\partial f^2} + S_i \tag{8.4-6}$$

and one equation for temperature:

$$\rho \frac{\partial T}{\partial t} = \frac{1}{2}\rho\chi \frac{\partial^2 T}{\partial f^2} - \frac{1}{c_p}\sum_i H_i S_i + \frac{1}{2c_p}\rho\chi \left[\frac{\partial c_p}{\partial f} + \sum_i c_{p,i}\frac{\partial Y_i}{\partial f}\right]\frac{\partial T}{\partial f} \tag{8.4-7}$$

The notation in Equations 8.4-6 and 8.4-7 is as follows: Y_i, T, ρ, and f are the ith species mass fraction, temperature, density, and mixture fraction, respectively. $c_{p,i}$ and c_p are the ith species specific heat and mixture-averaged specific heat, respectively. S_i is the ith species reaction rate, and H_i is the specific enthalpy of the ith species.

The scalar dissipation, χ, must be modeled across the flamelet. An extension of Equation 8.4-2 to variable density is used [163]:

$$\chi(f) = \frac{a_s}{4\pi}\frac{3\left(\sqrt{\rho_\infty/\rho}+1\right)^2}{2\sqrt{\rho_\infty/\rho}+1}\exp\left(-2[\mathrm{erfc}^{-1}(2f)]^2\right) \tag{8.4-8}$$

where ρ_∞ is the density of the oxidizer stream.

8.4.4 Flamelet Import

ANSYS FLUENT can import one or more flamelet files, convolute these flamelets with the assumed-shape PDFs (see Equation 8.4-3), and construct look-up tables. The flamelet files can be generated in ANSYS FLUENT, or with separate stand-alone computer codes.

Three types of flamelet files can be imported into ANSYS FLUENT: binary files generated by the OPPDIF code [210], ASCII files generated by the CFX-RIF code [15, 16, 272], and standard format files described in Section 16.7.4: Standard Flamelet Files in the separate User's Guide and in Peters and Rogg [270].

When flamelets are generated in physical space (such as with OPPDIF), the species and temperature vary in one spatial dimension. The species and temperature must then be mapped from physical space to mixture fraction space. If the diffusion coefficients of all species are equal, a unique definition of the mixture fraction exists. However, with differential diffusion, the mixture fraction can be defined in a number of ways.

ANSYS FLUENT provides four methods of computing the mixture fraction profile along the laminar flamelet:

- Average of C and H: Following the work of Drake and Blint [77], the mixture fraction is calculated as the mean value of $f_{\rm C}$ and $f_{\rm H}$, where $f_{\rm C}$ and $f_{\rm H}$ are the mixture fraction values based on the carbon and hydrogen elements.
- Hydrocarbon formula: Following the work of Bilger et al. [26], the mixture fraction is calculated as

  $$f = \frac{b - b_{\rm ox}}{b_{\rm fuel} - b_{\rm ox}} \tag{8.4-9}$$

  where

  $$b = 2\frac{Y_{\rm C}}{M_{w,{\rm C}}} + 0.5\frac{Y_{\rm H}}{M_{w,{\rm H}}} - \frac{Y_{\rm O}}{M_{w,{\rm O}}} \tag{8.4-10}$$

  $Y_{\rm C}$, $Y_{\rm H}$, and $Y_{\rm O}$ are the mass fractions of carbon, hydrogen, and oxygen atoms, and $M_{w,{\rm C}}$, $M_{w,{\rm H}}$, and $M_{w,{\rm O}}$ are the molecular weights. $b_{\rm ox}$ and $b_{\rm fuel}$ are the values of b at the oxidizer and fuel inlets.
- Nitrogen method: The mixture fraction is computed in terms of the mass fraction of the nitrogen species:

  $$f = \frac{Y_{\rm N} - Y_{\rm N,ox}}{Y_{\rm N,fuel} - Y_{\rm N,ox}} \tag{8.4-11}$$

  where $Y_{\rm N}$ is the elemental mass fraction of nitrogen along the flamelet, $Y_{\rm N,ox}$ is the mass fraction of nitrogen at the oxidizer inlet, and $Y_{\rm N,fuel}$ is the mass fraction of nitrogen at the fuel inlet.

- Read from a file (standard format files only): This option is for flamelets solved in mixture fraction space. If you choose this method, ANSYS FLUENT will search for the mixture fraction keyword Z, as specified in Peter and Roggs's work [270], and retrieve the data. If ANSYS FLUENT does not find mixture fraction data in the flamelet file, it will instead use the hydrocarbon formula method described above.

The flamelet profiles in the multiple-flamelet data set should vary only in the strain rate imposed; the species and the boundary conditions should be the same. In addition, it is recommended that an extinguished flamelet is excluded from the multiple-flamelet data set. The formats for multiple flamelets are as follows:

- OPPDIF format: The multiple-flamelet OPPDIF files should be produced using the CNTN keyword in the OPPDIF script. Alternatively, you can use ANSYS FLUENT to merge a number of single-flamelet OPPDIF files into a multiple-flamelet file.
- Standard format: If you have a set of standard format flamelet files, you can import them all at the same time, and ANSYS FLUENT will merge them internally into a multiple-flamelet file. When you import the set of flamelet files, ANSYS FLUENT will search for and count the occurrences of the HEADER keyword to determine the number of flamelets in the file.
- CFX-RIF format: A CFX-RIF flamelet file contains multiple flamelets at various strains and the file should not be modified manually. Only one CFX-RIF flamelet file should be imported.

For either type of file, ANSYS FLUENT will determine the number of flamelet profiles and sort them in ascending strain-rate order. For flamelets generated in physical space, you can select one of the four methods available for the calculation of mixture fraction. The scalar dissipation will be calculated from the strain rate using Equation 8.4-2.

8.5 The Steady Laminar Flamelet Model Theory

The steady laminar flamelet approach models a turbulent flame brush as an ensemble of discrete, steady laminar flames, called flamelets. The individual flamelets are assumed to have the same structure as laminar flames in simple configurations, and are obtained by experiments or calculations. Using detailed chemical mechanisms, ANSYS FLUENT can calculate laminar opposed-flow diffusion flamelets for non-premixed combustion. The laminar flamelets are then embedded in a turbulent flame using statistical PDF methods.

The advantage of the laminar flamelet approach is that realistic chemical kinetic effects can be incorporated into turbulent flames. The chemistry can then be preprocessed and tabulated, offering tremendous computational savings. However, the steady laminar flamelet model is limited to modeling combustion with relatively fast chemistry. The flame is assumed to respond instantaneously to the aerodynamic strain, and thus the model cannot capture deep non-equilibrium effects such as ignition, extinction, and slow chemistry (like NO_x).

Information pertaining strictly to the steady flamelet model is presented in the following sections:

- Section 8.5.1: Overview
- Section 8.5.2: Multiple Steady Flamelet Libraries
- Section 8.5.3: Steady Laminar Flamelet Automated Grid Refinement
- Section 8.5.4: Non-Adiabatic Steady Laminar Flamelets

For general information about the mixture fraction model, see Section 8.1: Introduction.

8.5.1 Overview

In a diffusion flame, at the molecular level, fuel and oxidizer diffuse into the reaction zone. Here, they encounter high temperatures and radical species and ignite. More heat and radicals are generated in the reaction zone and some diffuse out. In near-equilibrium flames, the reaction rate is much faster than the diffusion rate. However, as the flame is stretched and strained by the turbulence, species and temperature gradients increase, and radicals and heat diffuse more quickly out of the flame. The species have less time to reach chemical equilibrium, and the degree of local non-equilibrium increases.

The steady laminar flamelet model is suited to predict chemical non-equilibrium due to aerodynamic straining of the flame by the turbulence. The chemistry, however, is assumed to respond rapidly to this strain, so as the strain relaxes to zero, the chemistry tends to equilibrium.

When the chemical time-scale is comparable to the fluid mixing time-scale, the species can be considered to be in global chemical non-equilibrium. Such cases include NO_x formation and low-temperature CO oxidation. The steady laminar flamelet model is not suitable for such slow-chemistry flames. Instead, you can model slow chemistry using one of the following:

- the Unsteady Laminar Flamelet model (see Section 8.6: The Unsteady Laminar Flamelet Model Theory)
- the trace species assumption in the NO_x model (see Chapter 13: Pollutant Formation)
- the Laminar Finite-Rate model (see Section 7.1.2: The Generalized Finite-Rate Formulation for Reaction Modeling), where the turbulence-chemistry interaction is ignored.

- the EDC model (see Section 7.1.2: The Eddy-Dissipation-Concept (EDC) Model)
- the PDF transport model (see Chapter 11: Composition PDF Transport).

8.5.2 Multiple Steady Flamelet Libraries

ANSYS FLUENT can generate multiple steady flamelets over a range of strain rates to account for the varying strain field in your multi-dimensional simulation. If you specify the number of flamelets to be greater than one, flamelets are generated at scalar dissipation values as determined by Equation 8.5-1.

$$\chi_i = \begin{cases} 10\chi_{i-1} & \text{for } \chi_{i-1} < 1\ s^{-1} \\ \chi_{i-1} + \Delta\chi & \text{for } \chi_{i-1} \geq 1\ s^{-1} \end{cases} \tag{8.5-1}$$

where i ranges from 1 up to the specified maximum number of flamelets, χ_0 is the initial scalar dissipation, and $\Delta\chi$ is the scalar dissipation step. Flamelets are generated until either the maximum number of flamelets is reached, or the flamelet extinguishes. Extinguished flamelets are excluded from the flamelet library.

8.5.3 Steady Laminar Flamelet Automated Grid Refinement

By default, 1D flamelet grids are discretized by clustering a fixed number of points about the stoichiometric mixture fraction, which is approximated as the location of peak temperature. ANSYS FLUENT also has the option for Automated Grid Refinement of steady flamelets, where an adaptive algorithm inserts grid points so that the change of values, as well as the change of slopes, between successive grid points is less than user specified tolerances.

When using automated grid refinement, a steady solution is calculated on a coarse grid with a user specified Initial Number of Grid Points in Flamelet (default of 8). At convergence, a new grid point is inserted midway between a point i and its neighbor $i+1$ if,

$$|v_i - v_{i+1}| > \epsilon_v(v_{max} - v_{min}) \tag{8.5-2}$$

where v_i is the value for each temperature and species mass fraction at grid point i, ϵ_v is a user specified Maximum Change in Value Ratio (default of 0.5), and v_{max} (v_{min}) are the maximum (minimum) values over all grid points.

In addition a grid point is added if,

$$|s_i - s_{i+1}| > \epsilon_s(s_{max} - s_{min}) \tag{8.5-3}$$

where the slope s_i is defined as,

$$s_i = \frac{v_{i+1} - v_i}{f_{i+1} - f_i} \tag{8.5-4}$$

In Equations 8.5-3 and 8.5-4, ϵ_s is a user specified Maximum Change in Slope Ratio (default of 0.5), s_{max} (s_{min}) are the maximum (minimum) slopes over all grid points, and f_i is the mixture fraction value of grid point i.

The refined flamelet is reconverged, and the refinement process is repeated until no further grid points are added by Equations 8.5-2 and 8.5-3, or the user specified Maximum Number of Grid Points in Flamelet (default of 64) is exceeded.

8.5.4 Non-Adiabatic Steady Laminar Flamelets

For non-adiabatic steady flamelets, ANSYS FLUENT follows the approach of [27, 240] and assumes that flamelet species profiles are unaffected by heat loss/gain from the flamelet. No special non-adiabatic flamelet profiles need to be generated, avoiding a very cumbersome preprocessing step. In addition, the compatibility of ANSYS FLUENT with external steady flamelet generation packages (e.g., OPPDIF, CFX-RIF, RUN-1DL) is retained. The disadvantage to this model is that the effect of the heat losses on the species mass fractions is not taken into account. Also, the effect of the heat loss on the extinction limits is not taken into account.

After flamelet generation, the flamelet profiles are convoluted with the assumed-shape PDFs as in Equation 8.4-3, and then tabulated for look-up in ANSYS FLUENT. The non-adiabatic PDF tables have the following dimensions:

$\overline{T}(\overline{f}, \overline{f'^2}, \overline{H}, \overline{\chi})$
$\overline{Y_i}(\overline{f}, \overline{f'^2}, \overline{H})$ for $\chi = 0$ (i.e., equilibrium solution)
$\overline{Y_i}(\overline{f}, \overline{f'^2}, \overline{\chi})$ for $\chi \neq 0$
$\overline{\rho}(\overline{f}, \overline{f'^2}, \overline{H}, \overline{\chi})$

During the ANSYS FLUENT solution, the equations for the mean mixture fraction, mixture fraction variance, and mean enthalpy are solved. The scalar dissipation field is calculated from the turbulence field and the mixture fraction variance (Equation 8.4-4). The mean values of cell temperature, density, and species mass fraction are obtained from the PDF look-up table.

8.6 The Unsteady Laminar Flamelet Model Theory

The *steady* laminar flamelet model, described in Sections 8.4 and 8.5, models local chemical non-equilibrium due to the straining effect of turbulence. In many combustors the strain is small at the outlet and the steady flamelet model predicts all species, including slow-forming species like NO_x, to be near equilibrium, which is often inaccurate. The cause of this inaccuracy is the disparity between the flamelet time-scale, which is the inverse of the scalar dissipation, and the slow-forming species time-scale, which is the residence time since the species started accumulating after mixing in the combustor.

The *unsteady* laminar flamelet model in ANSYS FLUENT can predict slow-forming species, such as gaseous pollutants or product yields in liquid reactors, more accurately than the steady laminar flamelet model. Computationally expensive chemical kinetics are reduced to one dimension and the model is significantly faster than the laminar-finite-rate, EDC or PDF Transport models where kinetics are calculated in two or three dimensions. There are two variants of the unsteady laminar flamelet model, namely an Eulerian unsteady flamelet model (described in Section 8.6.1: The Eulerian Unsteady Laminar Flamelet Model) and a diesel unsteady flamelet model for predicting combustion in compression-ignition engines (described in Section 8.6.2: The Diesel Unsteady Laminar Flamelet Model).

Information pertaining strictly to the unsteady flamelet model is presented in the following sections:

- Section 8.6.1: The Eulerian Unsteady Laminar Flamelet Model
- Section 8.6.2: The Diesel Unsteady Laminar Flamelet Model

8.6.1 The Eulerian Unsteady Laminar Flamelet Model

The Eulerian unsteady laminar flamelet model can be used to predict slow-forming intermediate and product species which are not in chemical equilibrium. Typical examples of slow-forming species are gas-phase pollutants like NO_x, and product compounds in liquid reactors. By reducing the chemistry computation to one dimension, detailed kinetics with multiple species and stiff reactions can be economically simulated in complex 3D geometries.

The model, following the work of Barths et al. [16], postprocesses an unsteady marker probability equation on a steady-state, converged flow field. In ANSYS FLUENT, the steady flow solution must be computed with the steady laminar flamelet model (see Section 8.5: The Steady Laminar Flamelet Model Theory). Since the unsteady flamelet equations are postprocessed on a steady-state, steady flamelet solution, the effect of the unsteady flamelet species on the flow-field are neglected.

The transport equation for the unsteady flamelet marker probability, I, is

$$\frac{\partial}{\partial t}(\rho I) + \nabla \cdot (\rho \vec{v} I) = \nabla \cdot \left(\frac{\mu_t}{\sigma_t} \nabla I \right) \tag{8.6-1}$$

Equation 8.6-1 is always solved unsteady, and is initialized as

$$I = \begin{cases} 1 & \text{for } \overline{f} \geq f_{init} \\ \\ 0 & \text{for } \overline{f} < f_{init} \end{cases} \tag{8.6-2}$$

where $\overline{f}$ is the mean mixture fraction and f_{init} is a user supplied constant, which should be set greater than the stoichiometric mixture fraction. At inlet boundaries, ANSYS FLUENT always sets I toward zero, and hence the I field decreases to zero with time as I is convected and diffused out of the domain (for cases with outlet boundaries).

The unsteady flamelet species equations (Equation 8.4-6) are integrated simultaneously with the marker probability equation, I. For liquid-phase chemistry, the initial flamelet field is the mixed-but-unburnt flamelet, as liquid reactions are assumed to proceed immediately upon mixing. However, gas-phase chemistry invariably requires ignition, so the initial flamelet field is calculated from a steady flamelet solution. All slow-forming species, such as $\mathrm{NO_x}$, are zeroed in this initial flamelet profile since, at ignition, little time has elapsed for any significant formation. The slow-forming species are identified by the user before solution of the unsteady flamelet equations.

The scalar dissipation at stoichiometric mixture fraction ($\chi_{\rm st}$) is required by the flamelet species equation. This is calculated from the steady-state ANSYS FLUENT field at each time step as a probability-weighted volume integral:

$$\chi_{\rm st}(t) = \frac{\int_V I(\vec{x},t)\,\rho(\vec{x})\,\overline{\chi_{\rm st}}^{3/2}(\vec{x})\,dV}{\int_V I(\vec{x},t)\,\rho(\vec{x})\,\overline{\chi_{\rm st}}^{1/2}(\vec{x})\,dV} \tag{8.6-3}$$

where $\overline{\chi_{\rm st}}$ is defined in Equation 8.6-3, and V denotes the fluid volume. ANSYS FLUENT provides the option of limiting $\overline{\chi_{\rm st}}$ to a user-specified maximum value, which should be approximately equal to the flamelet extinction scalar dissipation (the steady flamelet solver can be used to calculate this extinction scalar dissipation in a separate simulation).

The unsteady flamelet energy equation is not solved in order to avoid flamelet extinction for high scalar dissipation, and to account for non-adiabatic heat loss or gain. For adiabatic cases, the flamelet temperature $T^{ad}(f,t)$ is calculated at each time step from the steady flamelet library at the probability-weighted scalar dissipation $\chi_{\rm st}$ from Equation 8.6-3. For non-adiabatic cases, the flamelet temperature at time t is calculated from

$$T(f,t) = T^{ad}(f,t)\, \xi(f,t) \tag{8.6-4}$$

where

$$\xi(f,t) = \frac{\int_V I\, \rho\, \overline{T}(\vec{x}|f)/T_{ad}(f,t)\, dV}{\int_V I\, \rho\, dV} \tag{8.6-5}$$

In Equation 8.6-5, $\overline{T}(\vec{x}|f)$ denotes the ANSYS FLUENT steady-state mean cell temperature conditioned on the local cell mixture fraction.

Unsteady flamelet mean species mass fractions in each cell are accumulated over time as

$$\overline{Y_k}^{ufla} = \frac{\int_0^t I\, \rho \left[\int_0^1 Y_k(f,t)\, P(f)\, df\right]\, dt}{\int_0^t I\, \rho\, dt} \tag{8.6-6}$$

where $Y_k(f,t)$ is the k'th species unsteady flamelet mass fraction, and $P(f)$ denotes the Beta PDF.

The probability marker equation (Equation 8.6-1) and the flamelet species equation (Equation 8.4-6) are advanced together in time until the probability marker has substantially convected and diffused out of the domain. The unsteady flamelet mean species, calculated from Equation 8.6-5, reaches steady-state as the probability marker I vanishes.

Liquid Reactions

Liquid reactors are typically characterized by:

- Near constant density and temperature.
- Relatively slow reactions and species far from chemical equilibrium.
- High Schmidt number (Sc) and hence reduced molecular diffusion.

The Eulerian unsteady laminar flamelet model can be used to model liquid reactions. When the Liquid Micro-Mixing model is enabled, ANSYS FLUENT uses the volume-weighted-mixing-law formula to calculate the density.

The effect of high Sc is to decrease mixing at the smallest (micro) scales and increase the mixture fraction variance, which is modeled with the Turbulent Mixer Model [11]. Three transport equations are solved for the inertial-convective ($\overline{f_{ic}'^2}$), viscous-convective ($\overline{f_{vc}'^2}$), and viscous-diffusive ($\overline{f_{vd}'^2}$) subranges of the turbulent scalar spectrum,

$$\frac{\partial}{\partial t}\left(\rho\overline{f_{ic}'^2}\right) + \nabla\cdot\left(\rho\vec{v}\overline{f_{ic}'^2}\right) = \nabla\cdot\left(\frac{\mu_t}{\sigma_t}\nabla\overline{f_{ic}'^2}\right) + C_1\mu_t\left(\nabla\overline{f}\right)^2 - C_2\rho\frac{\epsilon}{k}\overline{f_{ic}'^2} \tag{8.6-7}$$

$$\frac{\partial}{\partial t}\left(\rho\overline{f_{vc}'^2}\right) + \nabla\cdot\left(\rho\vec{v}\overline{f_{vc}'^2}\right) = \nabla\cdot\left(\frac{\mu_t}{\sigma_t}\nabla\overline{f_{vc}'^2}\right) + C_2\rho\frac{\epsilon}{k}\overline{f_{ic}'^2} - C_3\rho\sqrt{\frac{\epsilon}{\nu}}\overline{f_{vc}'^2} \tag{8.6-8}$$

$$\frac{\partial}{\partial t}\left(\rho\overline{f_{vd}'^2}\right) + \nabla\cdot\left(\rho\vec{v}\overline{f_{vd}'^2}\right) = \nabla\cdot\left(\frac{\mu_t}{\sigma_t}\nabla\overline{f_{vd}'^2}\right) + C_3\rho\sqrt{\frac{\epsilon}{\nu}}\left(\overline{f_{vc}'^2} - \left(C_4 + \frac{C_5}{Sc}\right)\overline{f_{vd}'^2}\right) \tag{8.6-9}$$

where the constants C_1 through C_5 have values of 2, 1.86, 0.058, 0.303, and 17050, respectively. The total mixture fraction variance is the sum of $\overline{f_{ic}'^2}$, $\overline{f_{vc}'^2}$ and $\overline{f_{vd}'^2}$.

In Equation 8.6-9, the cell Schmidt number, Sc, is calculated as $Sc = \mu/\rho D$ where μ is the viscosity, ρ the density, and D the mass diffusivity as defined for the *pdf-mixture* material.

8.6.2 The Diesel Unsteady Laminar Flamelet Model

In diesel engines, fuel sprayed into the cylinder evaporates, mixes with the surrounding gases, and then auto-ignites as compression raises the temperature and pressure. The diesel unsteady laminar flamelet model, based on the work of Pitsch et al. and Barths et al. [272, 15], models the chemistry in a single, one-dimensional laminar flamelet. By reducing the costly chemical kinetic calculation to 1D, substantial savings in run-time can be achieved over the laminar-finite-rate, EDC or PDF Transport models.

The flamelet species and energy equations (Equations 8.4-6 and 8.4-7) are solved simultaneously with the flow. The flamelet equations are advanced for a fractional step using properties from the flow, and then the flow is advanced for the same fractional time-step using properties from the flamelet.

The initial flamelet condition at the start of the diesel simulation is a mixed-but-unburnt distribution. For the flamelet fractional time-step, the volume-averaged scalar dissipation and pressure, as well as the fuel and oxidizer temperatures, are passed from the flow solver to the flamelet solver. To account for temperature rise during compression, the flamelet energy equation (Equation 8.4-7) has an additional term on the right-hand side as

$$\dot{q} = \frac{1}{c_p}\frac{\partial p}{\partial t} \tag{8.6-10}$$

where c_p is a the specific heat and p is the volume-averaged pressure in the cylinder. This rise in flamelet temperature due to compression eventually leads to ignition of the flamelet.

After the flamelet equations have been advanced for the fractional time-step, the PDF Table is created as a Non-Adiabatic Steady Flamelet table (see Section 8.5.4: Non-Adiabatic Steady Laminar Flamelets). Using the properties of this table, the CFD flow field is then advanced for the same fractional time-step.

The diesel unsteady flamelet approach can model ignition as well as formation of product, intermediate and pollutant species. Enabling the Diesel Unsteady Flamelet model is described in Section 16.3.6: Using the Diesel Unsteady Laminar Flamelet Model in the separate User's Guide.

Chapter 9. Premixed Combustion

ANSYS FLUENT has premixed turbulent combustion models based on the reaction-progress variable approach. For more information about using the premixed turbulent combustion model, see Chapter 17: Modeling Premixed Combustion in the separate User's Guide. Theoretical information about this model is provided in the following sections:

9.1 Overview and Limitations

9.1.1 Overview

In premixed combustion, fuel and oxidizer are mixed at the molecular level prior to ignition. Combustion occurs as a flame front propagating into the unburnt reactants. Examples of premixed combustion include aspirated internal combustion engines, lean-premixed gas turbine combustors, and gas-leak explosions.

Premixed combustion is much more difficult to model than non-premixed combustion. The reason for this is that premixed combustion usually occurs as a thin, propagating flame that is stretched and contorted by turbulence. For subsonic flows, the overall rate of propagation of the flame is determined by both the laminar flame speed and the turbulent eddies. The laminar flame speed is determined by the rate that species and heat diffuse upstream into the reactants and burn. To capture the laminar flame speed, the internal flame structure would need to be resolved, as well as the detailed chemical kinetics and molecular diffusion processes. Since practical laminar flame thicknesses are of the order of millimeters or smaller, resolution requirements are usually unaffordable.

The effect of turbulence is to wrinkle and stretch the propagating laminar flame sheet, increasing the sheet area and, in turn, the effective flame speed. The large turbulent eddies tend to wrinkle and corrugate the flame sheet, while the small turbulent eddies, if they are smaller than the laminar flame thickness, may penetrate the flame sheet and modify the laminar flame structure.

Non-premixed combustion, in comparison, can be greatly simplified to a mixing problem (see the mixture fraction approach in Section 8.1: Introduction). The essence of premixed combustion modeling lies in capturing the turbulent flame speed, which is influenced by both the laminar flame speed and the turbulence.

In premixed flames, the fuel and oxidizer are intimately mixed before they enter the combustion device. Reaction then takes place in a combustion zone that separates unburnt reactants and burnt combustion products. Partially premixed flames exhibit the properties of both premixed and diffusion flames. They occur when an additional oxidizer or fuel stream enters a premixed system, or when a diffusion flame becomes lifted off the burner so that some premixing takes place prior to combustion.

Premixed and partially premixed flames can be modeled using ANSYS FLUENT's finite-rate/eddy-dissipation formulation (see Chapter 7: Species Transport and Finite-Rate Chemistry). If finite-rate chemical kinetic effects are important, the Laminar Finite-Rate model (see Section 7.1.2: The Laminar Finite-Rate Model), the EDC model (see Section 7.1.2: The Eddy-Dissipation-Concept (EDC) Model) or the composition PDF transport model (see Chapter 11: Composition PDF Transport) can be used. For information about ANSYS FLUENT's partially premixed combustion model, see Chapter 10: Partially Premixed Combustion. If the flame is perfectly premixed (all streams entering the combustor have the same equivalence ratio), it is possible to use the premixed combustion model, as described in this chapter.

9.1.2 Limitations

The following limitations apply to the premixed combustion model:

- You must use the pressure-based solver. The premixed combustion model is not available with either of the density-based solvers.
- The premixed combustion model is valid only for turbulent, subsonic flows. These types of flames are called deflagrations. Explosions, also called detonations, where the combustible mixture is ignited by the heat behind a shock wave, can be modeled with the finite-rate model using the density-based solver. See Chapter 7: Species Transport and Finite-Rate Chemistry for information about the finite-rate model.
- The premixed combustion model cannot be used in conjunction with the pollutant (i.e., soot and NO_x) models. However, a perfectly premixed system can be modeled with the partially premixed model (see Chapter 10: Partially Premixed Combustion), which can be used with the pollutant models.
- You cannot use the premixed combustion model to simulate reacting discrete-phase particles, since these would result in a partially premixed system. Only inert particles can be used with the premixed combustion model.

9.2 Zimont Turbulent Flame Closure Theory

The turbulent premixed combustion model, based on work by Zimont et al. [390, 391, 393], involves the solution of a transport equation for the reaction progress variable. The closure of this equation is based on the definition of the turbulent flame speed.

Information in this section is provided in the following sections:

- Section 9.2.1: Propagation of the Flame Front
- Section 9.2.2: Turbulent Flame Speed

9.2.1 Propagation of the Flame Front

In many industrial premixed systems, combustion takes place in a thin flame sheet. As the flame front moves, combustion of unburnt reactants occurs, converting unburnt premixed reactants to burnt products. The premixed combustion model thus considers the reacting flow field to be divided into regions of burnt and unburnt species, separated by the flame sheet.

The flame front propagation is modeled by solving a transport equation for the density-weighted mean reaction progress variable, denoted by c:

$$\frac{\partial}{\partial t}(\rho c) + \nabla \cdot (\rho \vec{v} c) = \nabla \cdot \left(\frac{\mu_t}{\mathrm{Sc}_t} \nabla c \right) + \rho S_c \tag{9.2-1}$$

where

c = mean reaction progress variable
Sc_t = turbulent Schmidt number
S_c = reaction progress source term (s^{-1})

The progress variable is defined as a normalized sum of the product species,

$$c = \frac{\sum_{i=1}^{n} Y_i}{\sum_{i=1}^{n} Y_{i,\mathrm{eq}}} \tag{9.2-2}$$

where

n = number of products
Y_i = mass fraction of product species i
$Y_{i,\mathrm{eq}}$ = equilibrium mass fraction of product species i

Based on this definition, $c = 0$ where the mixture is unburnt and $c = 1$ where the mixture is burnt:

- $c = 0$: unburnt mixture
- $c = 1$: burnt mixture

The value of c is defined as a boundary condition at all flow inlets. It is usually specified as either 0 (unburnt) or 1 (burnt).

The mean reaction rate in Equation 9.2-1 is modeled as [391]

$$\rho S_c = \rho_u U_t |\nabla c| \tag{9.2-3}$$

where

ρ_u = density of unburnt mixture
U_t = turbulent flame speed

Many other models for turbulent flame speed exist [36], and can be specified using user-defined functions. More information about user-defined functions can be found in the separate UDF Manual.

9.2.2 Turbulent Flame Speed

The key to the premixed combustion model is the prediction of U_t, the turbulent flame speed normal to the mean surface of the flame. The turbulent flame speed is influenced by the following:

- laminar flame speed, which is, in turn, determined by the fuel concentration, temperature, and molecular diffusion properties, as well as the detailed chemical kinetics
- flame front wrinkling and stretching by large eddies, and flame thickening by small eddies

In ANSYS FLUENT, the Zimont turbulent flame speed closure is computed using a model for wrinkled and thickened flame fronts [391]:

$$U_t = A(u')^{3/4} U_l^{1/2} \alpha^{-1/4} \ell_t^{1/4} \tag{9.2-4}$$

$$= Au' \left(\frac{\tau_t}{\tau_c}\right)^{1/4} \tag{9.2-5}$$

where

A	=	model constant
u'	=	RMS (root-mean-square) velocity (m/s)
U_l	=	laminar flame speed (m/s)
$\alpha = k/\rho c_p$	=	molecular heat transfer coefficient of unburnt mixture (thermal diffusivity) (m^2/s)
ℓ_t	=	turbulence length scale (m)
$\tau_t = \ell_t/u'$	=	turbulence time scale (s)
$\tau_c = \alpha/U_l^2$	=	chemical time scale (s)

The turbulence length scale, ℓ_t, is computed from

$$\ell_t = C_D \frac{(u')^3}{\epsilon} \tag{9.2-6}$$

where ϵ is the turbulence dissipation rate.

The model is based on the assumption of equilibrium small-scale turbulence inside the laminar flame, resulting in a turbulent flame speed expression that is purely in terms of the large-scale turbulent parameters. The default value of 0.52 for A is recommended [391], and is suitable for most premixed flames. The default value of 0.37 for C_D should also be suitable for most premixed flames.

The model is strictly applicable when the smallest turbulent eddies in the flow (the Kolmogorov scales) are smaller than the flame thickness, and penetrate into the flame zone. This is called the thin reaction zone combustion region, and can be quantified by Karlovitz numbers, Ka, greater than unity. Ka is defined as

$$\mathrm{Ka} = \frac{t_l}{t_\eta} = \frac{v_\eta^2}{U_l^2} \tag{9.2-7}$$

where

t_l	=	characteristic flame time scale
t_η	=	smallest (Kolmogorov) turbulence time scale
$v_\eta = (\nu\epsilon)^{1/4}$	=	Kolmogorov velocity
ν	=	kinematic viscosity

Lastly, the model is valid for premixed systems where the flame brush width increases in time, as occurs in most industrial combustors. Flames that propagate for a long period of time equilibrate to a constant flame width, which cannot be captured in this model.

Turbulent Flame Speed for LES

For simulations that use the LES turbulence model, the Reynolds-averaged quantities in the turbulent flame speed expression (Equation 9.2-4) are replaced by their equivalent subgrid quantities. In particular, the large eddy length scale ℓ_t is modeled as

$$\ell_t = C_s \Delta \tag{9.2-8}$$

where C_s is the Smagorinsky constant and Δ is the cell characteristic length.

The RMS velocity in Equation 9.2-4 is replaced by the subgrid velocity fluctuation, calculated as

$$u' = \ell_t \tau_{sgs}^{-1} \tag{9.2-9}$$

where τ_{sgs}^{-1} is the subgrid scale mixing rate (inverse of the subgrid scale time scale), given in Equation 7.1-28.

Laminar Flame Speed

The laminar flame speed (U_l in Equation 9.2.1) can be specified as constant, or as a user-defined function. A third option appears for non-adiabatic premixed and partially-premixed flames and is based on the correlation proposed by Meghalchi and Keck [227],

$$U_l = U_{l,ref} \left(\frac{T_u}{T_{u,ref}} \right)^{\gamma} \left(\frac{p_u}{p_{u,ref}} \right)^{\beta} \tag{9.2-10}$$

In Equation 9.2-10, T_u and p_u are the unburnt reactant temperature and pressure ahead of the flame, $T_{u,ref} = 298K$ and $p_{u,ref} = 1atm$.

The reference laminar flame speed, $U_{l,ref}$, is calculated from

$$U_{l,ref} = C_1 + C_2(\phi - C_3)^2 \tag{9.2-11}$$

where ϕ is the equivalence ratio ahead of the flame front, and C_1, C_2 and C_3 are fuel-specific constants. The exponents γ and β are calculated from,

$$\begin{aligned} \gamma &= 2.18 - 0.8(\phi - 1) \\ \beta &= -0.16 + 0.22(\phi - 1) \end{aligned} \tag{9.2-12}$$

The Meghalchi-Keck laminar flame speeds are available for fuel-air mixtures of methane, methanol, propane, iso-octane and indolene fuels.

Unburnt Density and Thermal Diffusivity

The unburnt density (ρ_u in Equation 9.2.1) and unburnt thermal diffusivity (α in Equation 9.2-5) are specified constants that are set in the Materials dialog box. However, for compressible cases, such as in-cylinder combustion, these can change significantly in time and/or space. When the ideal gas model is selected for density, the unburnt density and thermal diffusivity are calculated as volume averages ahead of the flame front.

Flame Stretch Effect

Since industrial low-emission combustors often operate near lean blow-off, flame stretching will have a significant effect on the mean turbulent heat release intensity. To take this flame stretching into account, the source term for the progress variable (ρS_c in Equation 9.2-1) is multiplied by a stretch factor, G [393]. This stretch factor represents the probability that the stretching will not quench the flame; if there is no stretching ($G = 1$), the probability that the flame will be unquenched is 100%.

The stretch factor, G, is obtained by integrating the log-normal distribution of the turbulence dissipation rate, ϵ:

$$G = \frac{1}{2}\mathrm{erfc}\left\{-\sqrt{\frac{1}{2\sigma}}\left[\ln\left(\frac{\epsilon_{\mathrm{cr}}}{\epsilon}\right) + \frac{\sigma}{2}\right]\right\} \tag{9.2-13}$$

where erfc is the complementary error function, and σ and ϵ_{cr} are defined below.

σ is the standard deviation of the distribution of ϵ:

$$\sigma = \mu_{\mathrm{str}} \ln\left(\frac{L}{\eta}\right) \tag{9.2-14}$$

where μ_{str} is the stretch factor coefficient for dissipation pulsation, L is the turbulent integral length scale, and η is the Kolmogorov micro-scale. The default value of 0.26 for μ_{str} (measured in turbulent non-reacting flows) is recommended by [391], and is suitable for most premixed flames.

ϵ_{cr} is the turbulence dissipation rate at the critical rate of strain [391]:

$$\epsilon_{\mathrm{cr}} = 15\nu g_{\mathrm{cr}}^2 \tag{9.2-15}$$

By default, $g_{\rm cr}$ is set to a very high value (1×10^8) so no flame stretching occurs. To include flame stretching effects, the critical rate of strain $g_{\rm cr}$ should be adjusted based on experimental data for the burner. Numerical models can suggest a range of physically plausible values [391], or an appropriate value can be determined from experimental data. A reasonable model for the critical rate of strain $g_{\rm cr}$ is

$$g_{\rm cr} = \frac{BU_l^2}{\alpha} \tag{9.2-16}$$

where B is a constant (typically 0.5) and α is the unburnt thermal diffusivity. Equation 9.2-16 can be implemented in ANSYS FLUENTusing a property user-defined function. More information about user-defined functions can be found in the separate UDF Manual.

Gradient Diffusion

Volume expansion at the flame front can cause counter-gradient diffusion. This effect becomes more pronounced when the ratio of the reactant density to the product density is large, and the turbulence intensity is small. It can be quantified by the ratio $(\rho_u/\rho_b)(U_l/I)$, where ρ_u, ρ_b, U_l, and I are the unburnt and burnt densities, laminar flame speed, and turbulence intensity, respectively. Values of this ratio greater than one indicate a tendency for counter-gradient diffusion, and the premixed combustion model may be inappropriate. Recent arguments for the validity of the turbulent-flame-speed model in such regimes can be found in Zimont et al. [392].

Wall Damping

High turbulent kinetic energy levels at the walls in some problems can cause an unphysical acceleration of the flame along the wall. In reality, radical quenching close to walls decreases reaction rates and thus the flame speed, but is not included in the model. To approximate this effect, ANSYS FLUENT includes a constant multiplier for the turbulent flame speed, α_w, which modifies the flame speed in the vicinity of wall boundaries:

$$U_t = \alpha_w A \left(\frac{\tau_t}{\tau_c}\right)^{1/4} \tag{9.2-17}$$

The default for this constant is 1 which does not change the flame speed. Values of α_w larger than 1 increase the flame speed, while values less than 1 decrease the flame speed in the cells next to the wall boundary.

ANSYS FLUENT will solve the transport equation for the reaction progress variable c (Equation 9.2-1), computing the source term, ρS_c, based on the theory outlined above:

$$\begin{aligned}\rho S_c &= AG\rho_u I^{3/4}[U_l(\lambda_{\rm lp})]^{1/2}[\alpha(\lambda_{\rm lp})]^{-1/4}\ell_t^{1/4}|\nabla c| & (9.2\text{-}18)\\ &= AG\rho_u I\left[\frac{\tau_t}{\tau_c(\lambda_{\rm lp})}\right]^{1/4}|\nabla c| & (9.2\text{-}19)\end{aligned}$$

9.3 Extended Coherent Flamelet Model Theory

The Extended Coherent Flamelet Model (ECFM) [274] is a more refined premixed combustion model than the Zimont Turbulent Flame Closure. It has theoretically greater accuracy, but is less robust and requires greater computational effort to converge.

Information in this section is provided in the following sections:

- Section 9.3.1: Closure for ECFM Source Terms
- Section 9.3.2: Turbulent Flame Speed in ECFM

The ECFM model solves an additional equation for the flame area density, denoted Σ, which is ultimately used to model the mean reaction rate in Equation 9.2-1. The model assumes that the smallest turbulence length scales (Kolmogorov eddies) are larger than the laminar flame thickness, so the effect of turbulence is to wrinkle the laminar flame sheet, however the internal laminar flame profile is not distorted. The increased surface area of the flame results in increased net fuel consumption and an increased flame speed. The range of applicability of the ECFM model is illustrated on the Borghi diagram in Figure 9.3.1, where the wrinkled flamelets regime is indicated below the $Da = 1$ line. Typical Internal Combustion (IC) engines typically operate in this wrinkled flamelet range.

An expression for the transport of the net flame area per unit volume, or flame area density, Σ, can be derived based on these assumptions [44]:

$$\frac{\partial \Sigma}{\partial t} + \nabla\cdot(\vec{v}\Sigma) = \nabla\cdot\left(\frac{\mu_t}{{\rm Sc}_t}\nabla\left(\frac{\Sigma}{\rho}\right)\right) + (P_1 + P_2 + P_3)\,\Sigma - D \qquad (9.3\text{-}1)$$

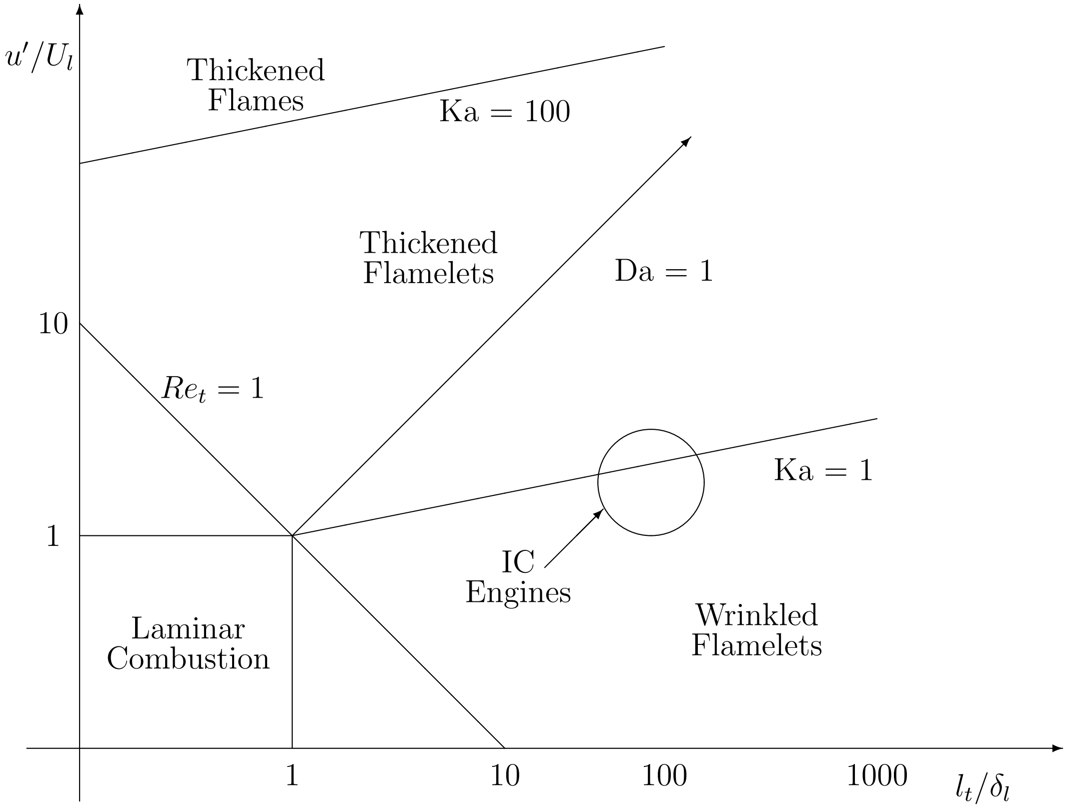


Figure 9.3.1: Borghi diagram for turbulent combustion

where

Σ	=	mean flame area density
Sc_t	=	turbulent Schmidt number
μ_t	=	turbulent viscosity
ρ	=	density
P_1	=	Source due to turbulence interaction
P_2	=	Source due to dilatation in the flame
P_3	=	Source due to expansion of burned gas
D	=	Dissipation of flame area

Equation 9.3-1 requires closure terms for the production and destruction terms for flame area density. Several families of closure terms have been put forth in the literature [274]. ANSYS FLUENT uses the closure described in the following section.

9.3.1 Closure for ECFM Source Terms

P_1 represents the production of flame area density by turbulent flame stretching, and is modeled as

$$P_1 = \alpha_1 K_t = \alpha_1 \frac{\epsilon}{k} \left[(1-\alpha_0) + \alpha_0 \Gamma_K\right] \tag{9.3-2}$$

where K_t is a turbulent time scale and α_1 is a constant with a default value of 1.6. The constant α_0 (default of 1) is a user-specified linear blending between the Intermediate Turbulent Net Flame Stretch (ITNFS) term, Γ_K, for low turbulence levels at $\alpha_0 = 1$, and a straightforward turbulent time scale source when $\alpha_0 = 0$ for high turbulence levels.

The ITNFS term, Γ_K, can be specified either as a constant or calculated as a function of the two parameters u'/U_l and l_t/δ_L^0, where u' is the turbulent velocity fluctuation, U_l is the laminar flame speed, l_t is the integral turbulent length scale and δ_L^0 is the laminar flame thickness.

The expression for Γ_K is given by:

$$\log_{10}(\Gamma_K) = -\frac{1}{(s+0.4)} \exp\left(-(s+0.4)\right) + \left(1 - \exp\left(-(s+0.4)\right)\right) \left(\sigma_1 \left(\frac{u'}{U_l}\right) s - 0.11\right) \tag{9.3-3}$$

where s is defined as

$$s = \log_{10}\left(\frac{l_t}{\delta_l^0}\right) \tag{9.3-4}$$

and σ_1 is

$$\sigma_1 \left(\frac{u'}{U_l}\right) = \frac{2}{3}\left(1 - \frac{1}{2}\exp\left(-\left(\frac{u'}{U_l}\right)^{\frac{1}{3}}\right)\right) \tag{9.3-5}$$

The ITNFS term, Γ_K, is sensitive to the laminar flame thickness. ANSYS FLUENT provides several options for the calculation of this quantity:

- Constant (user-specified) value
- Meneveau Flame Thickness [223]

  The laminar flame thickness is calculated as

$$\delta_l^0 = \frac{2\alpha}{U_l} \tag{9.3-6}$$

where α is the local unburnt thermal diffusivity.

- Poinsot Flame Thickness [274]

  The flame thickness is evaluated as in Equation 9.3-6 but an additional term, Γ_p is added to Γ_K. Γ_p is calculated as

  $$\Gamma_p = -\frac{3}{2}\frac{l_t U_l}{\delta_l u'} \log\left(\frac{1}{1-p_q}\right) \tag{9.3-7}$$

  where

  $$\begin{array}{rcl} p_q & = & -\frac{1}{2}\left(1+\tanh\left(b^3/|b|\right)\right) \\ b & = & \left[\log_{10}\left(u'/U_l\right) - g_l\right]/0.04s \\ g_l & = & \left(0.7+e^s\right)e^{-s} + \left(1+e^{-s}\right)\left(1+0.36s\right) \\ s & = & \log_{10}\left(l_t U_l/\alpha\right) \end{array}$$

- Blint Correction Flame Thickness [30]

  This includes a correction due to rapid expansion of the gas:

  $$\delta_l^0 = 2\left(\frac{T_b}{T_u}\right)^{0.7}\frac{\alpha}{U_l} \tag{9.3-8}$$

  where T_u is the unburned temperature, and T_b denotes the burned temperature.

The term P_2 in Equation 9.3-1 models the influence of dilatation on the production of flame area density. The term is given by

$$P_2 = \alpha_2 \frac{2}{3}\nabla\cdot(\rho\vec{u}) \tag{9.3-9}$$

where the constant α_2 has a default of 1.

The term P_3 models the effect of thermal expansion of the burned gas on the flame area density, and is given by

$$P_3 = \alpha_3 \frac{\rho_u}{\rho_b} U_l \frac{1-c}{c}\Sigma \tag{9.3-10}$$

where α_3 is 1 by default.

The flame area destruction term D is modeled as

$$D = \beta U_l \frac{\Sigma^2}{1-c} \tag{9.3-11}$$

where β is a constant with a default value of 1.

As formulated, the model can become singular for $c = 0$ and $c = 1$, which is handled by limiting c. Further, the production terms P_1 and P_2 can be non-zero in regions where the mixture is outside the flammability limits, which is unphysical. Accordingly, ANSYS FLUENT sets the production terms to zero when the laminar flame speed is less than a very small value. The stability of the solution is enhanced by ensuring that the laminar flame speed in the destruction term is always greater than a small, finite value. Inspection of the function for Γ_K shows that a singularity exists in Equation 9.3-3 for s = -0.4 which can occur when the turbulent integral length scale is small compared to the laminar flame thickness. To prevent the singularity, the quantity $(s + 0.4)$ is limited to a small positive number. This results in a small net turbulent flame stretch term in laminar zones. These numerical limiting constants can be adjusted in the TUI.

9.3.2 Turbulent Flame Speed in ECFM

The mean reaction rate term in the reaction progress variable Equation 9.2-1 is closed as

$$\rho S_c = \rho_u U_l \Sigma \tag{9.3-12}$$

which is the product of the unburnt density, ρ_u, laminar flame speed, U_l, and flame surface area density, Σ.

9.4 Calculation of Temperature

The calculation method for temperature will depend on whether the model is adiabatic or non-adiabatic.

9.4.1 Adiabatic Temperature Calculation

For the adiabatic premixed combustion model, the temperature is assumed to be a linear function of reaction progress between the lowset temperature of the unburnt mixture, T_u, and the highest adiabatic burnt temperature T_{ad}:

$$T = (1 - c)T_u + cT_{\mathrm{ad}} \tag{9.4-1}$$

9.4.2 Non-Adiabatic Temperature Calculation

For the non-adiabatic premixed combustion model, ANSYS FLUENT solves an energy transport equation in order to account for any heat losses or gains within the system. The energy equation in terms of sensible enthalpy, h, for the fully premixed fuel (see Equation 5.2-3) is as follows:

$$\frac{\partial}{\partial t}(\rho h) + \nabla \cdot (\rho \vec{v} h) = \nabla \cdot \left(\frac{k + k_t}{c_p} \nabla h \right) + S_{h,\text{chem}} + S_{h,\text{rad}} \tag{9.4-2}$$

$S_{h,\text{rad}}$ represents the heat losses due to radiation and $S_{h,\text{chem}}$ represents the heat gains due to chemical reaction:

$$S_{h,\text{chem}} = \rho S_c H_{\text{comb}} Y_{\text{fuel}} \tag{9.4-3}$$

where

S_c	=	normalized average rate of product formation (s^{-1})
H_{comb}	=	heat of combustion for burning 1 kg of fuel (J/kg)
Y_{fuel}	=	fuel mass fraction of unburnt mixture

9.5 Calculation of Density

ANSYS FLUENT calculates the premxed density using the ideal gas law. For the adiabatic model, pressure variations are neglected and the mean molecular weight is assumed to be constant. The burnt gas density is then calculated from the following relation:

$$\rho_b T_b = \rho_u T_u \tag{9.5-1}$$

where the subscript u refers to the unburnt cold mixture, and the subscript b refers to the burnt hot mixture. The required inputs are the unburnt density (ρ_u), the unburnt temperature (T_u), and the burnt adiabatic flame temperature (T_b).

For the non-adiabatic model, you can choose to either include or exclude pressure variations in the ideal gas equation of state. If you choose to ignore pressure fluctuations, ANSYS FLUENT calculates the density from

$$\rho T = \rho_u T_u \tag{9.5-2}$$

where T is computed from the energy transport equation, Equation 9.4-2. The required inputs are the unburnt density (ρ_u) and the unburnt temperature (T_u). Note that, from the incompressible ideal gas equation, the expression $\rho_u R T_u / p_{\text{op}}$ may be considered to be the effective molecular weight of the gas, where R is the gas constant and p_{op} is the operating pressure.

If you want to include pressure fluctuations for a compressible gas, you will need to input the effective molecular weight of the gas, and the density will be calculated from the ideal gas equation of state.

Chapter 10. Partially Premixed Combustion

ANSYS FLUENT provides a partially premixed combustion model that is based on the non-premixed combustion model described in Chapter 8: Non-Premixed Combustion and the premixed combustion model described in Chapter 9: Premixed Combustion. For information about using the partially premixed combustion model, see Chapter 18: Modeling Partially Premixed Combustion in the separate User's Guide. Information about the theory behind the partially premixed combustion model is presented in the following sections:

- Section 10.1: Overview and Limitations
- Section 10.2: Partially Premixed Combustion Theory

10.1 Overview and Limitations

10.1.1 Overview

Partially premixed combustion systems are premixed flames with non-uniform fuel-oxidizer mixtures (equivalence ratios). Such flames include premixed jets discharging into a quiescent atmosphere, lean premixed combustors with diffusion pilot flames and/or cooling air jets, and imperfectly premixed inlets.

The partially premixed model in ANSYS FLUENT is a simple combination of the non-premixed model (Chapter 8: Non-Premixed Combustion) and the premixed model (Chapter 9: Premixed Combustion). The premixed reaction-progress variable, c, determines the position of the flame front. Behind the flame front ($c = 1$), the mixture is burnt and the equilibrium or laminar flamelet mixture fraction solution is used. Ahead of the flame front ($c = 0$), the species mass fractions, temperature, and density are calculated from the mixed but unburnt mixture fraction. Within the flame ($0 < c < 1$), a linear combination of the unburnt and burnt mixtures is used.

10.1.2 Limitations

The underlying theory, assumptions, and limitations of the non-premixed and premixed models apply directly to the partially premixed model. In particular, the single-mixture-fraction approach is limited to two inlet streams, which may be pure fuel, pure oxidizer, or a mixture of fuel and oxidizer. The two-mixture-fraction model extends the number of inlet streams to three, but incurs a major computational overhead. See Sections 9.1.2 for additional limitations.

10.2 Partially Premixed Combustion Theory

The partially premixed model solves a transport equation for the mean reaction progress variable, $\overline{c}$ (to determine the position of the flame front), as well as the mean mixture fraction, $\overline{f}$ and the mixture fraction variance, $\overline{f'^2}$. Ahead of the flame ($c = 0$), the fuel and oxidizer are mixed but unburnt, and behind the flame ($c = 1$), the mixture is burnt.

10.2.1 Calculation of Scalar Quantities

Density weighted mean scalars (such as species fractions and temperature), denoted by $\overline{\phi}$, are calculated from the probability density function (PDF) of f and c as

$$\overline{\phi} = \int_0^1 \int_0^1 \phi(f, c) p(f, c)\, df\, dc \tag{10.2-1}$$

Under the assumption of thin flames, so that only unburnt reactants and burnt products exist, the mean scalars are determined from

$$\overline{\phi} = \overline{c} \int_0^1 \phi_b(f) p(f)\, df \; + (1 - \overline{c}) \int_0^1 \phi_u(f) p(f)\, df \tag{10.2-2}$$

where the subscripts b and u denote burnt and unburnt, respectively.

The burnt scalars, ϕ_b, are functions of the mixture fraction and are calculated by mixing a mass f of fuel with a mass $(1 - f)$ of oxidizer and allowing the mixture to equilibrate. When non-adiabatic mixtures and/or laminar flamelets are considered, ϕ_b is also a function of enthalpy and/or strain, but this does not alter the basic formulation. The unburnt scalars, ϕ_u, are calculated similarly by mixing a mass f of fuel with a mass $(1 - f)$ of oxidizer, but the mixture is not reacted.

Just as in the non-premixed model, the chemistry calculations and PDF integrations for the burnt mixture are performed in ANSYS FLUENT, and look-up tables are constructed.

Turbulent fluctuations are neglected for the unburnt mixture, so the mean unburnt scalars, $\overline{\phi}_u$, are functions of $\overline{f}$ only. The unburnt density, temperature, specific heat, and thermal diffusivity are fitted in ANSYS FLUENT to third-order polynomials of $\overline{f}$ using linear least squares:

$$\overline{\phi}_u = \sum_{n=0}^{3} c_n \overline{f}^n \tag{10.2-3}$$

Since the unburnt scalars are smooth and slowly-varying functions of $\overline{f}$, these polynomial fits are generally accurate. Access to polynomials is provided in case you want to modify them.

10.2.2 Laminar Flame Speed

The premixed models require the laminar flame speed (see Equation 9.2-4), which depends strongly on the composition, temperature, and pressure of the unburnt mixture. For adiabatic perfectly premixed systems as in Chapter 9: Premixed Combustion, the reactant stream has one composition, and the laminar flame speed is constant throughout the domain. However, in partially premixed systems, the laminar flame speed will change as the reactant composition (equivalence ratio) changes, and this must be taken into account.

Accurate laminar flame speeds are difficult to determine analytically, and are usually measured from experiments or computed from 1D simulations. ANSYS FLUENT uses fitted curves obtained from numerical simulations of the laminar flame speed [114]. These curves were determined for hydrogen (H_2), methane (CH_4), acetylene (C_2H_2), ethylene (C_2H_4), ethane (C_2H_6), and propane (C_3H_8) fuels. They are valid for inlet compositions ranging from the lean limit through unity equivalence ratio (stoichiometric), for unburnt temperatures from 298 K to 800 K, and for pressures from 1 bar to 40 bars.

ANSYS FLUENT fits these curves to a piecewise-linear polynomial. Mixtures leaner than the lean limit or richer than the rich limit will not burn, and have zero flame speed. The required inputs are values for the laminar flame speed at ten mixture fraction ($\overline{f}$) points. The first (minimum) and last (maximum) $\overline{f}$ inputs are the flammability limits of the mixture and the laminar flame speed is zero outside these value.

> ℹ These flame speed fits are accurate for air mixtures with pure fuels of H_2, CH_4, C_2H_2, C_2H_4, C_2H_6, and C_3H_8. If an oxidizer other than air or a different fuel is used, or if the unburnt temperature or pressure is outside the range of validity, then the curve fits will be incorrect. Although ANSYS FLUENT defaults to a methane-air mixture, the laminar flame speed polynomial and the rich and lean limits are most likely incorrect for your specified fuel/oxidizer and unburnt temperature/pressure conditions. The laminar flame speed polynomial should be determined from other sources, such as measurements from the relevant literature or detailed 1D simulations, and then input into ANSYS FLUENT.

Chapter 11. Composition PDF Transport

ANSYS FLUENT provides a composition PDF transport model for modeling finite-rate chemistry effects in turbulent flames. For information about using the composition PDF transport model, see Chapter 19: Modeling a Composition PDF Transport Problem in the separate User's Guide. Information about the theory behind this model is presented in the following sections:

- Section 11.1: Overview and Limitations
- Section 11.2: Composition PDF Transport Theory
- Section 11.3: The Lagrangian Solution Method
- Section 11.4: The Eulerian Solution Method

11.1 Overview and Limitations

The composition PDF transport model, like the Laminar Finite-Rate (see Section 7.1.2: The Laminar Finite-Rate Model) and EDC model (see Section 7.1.2: The Eddy-Dissipation-Concept (EDC) Model), should be used when you are interested in simulating finite-rate chemical kinetic effects in turbulent reacting flows. With an appropriate chemical mechanism, kinetically-controlled species such as CO and NO_x, as well as flame extinction and ignition, can be predicted. PDF transport simulations are computationally expensive, and it is recommended that you start your modeling with small meshes, and preferably in 2D.

A limitation that applies to the composition PDF transport model is that you must use the pressure-based solver as the model is not available with the density-based solver.

ANSYS FLUENT has two different discretizations of the composition PDF transport equation, namely Lagrangian and Eulerian. The Lagrangian method is strictly more accurate than the Eulerian method, but requires significantly longer run time to converge.

11.2 Composition PDF Transport Theory

Turbulent combustion is governed by the reacting Navier-Stokes equations. While this equation set is accurate, its direct solution (where all turbulent scales are resolved) is far too expensive for practical turbulent flows. In Chapter 7: Species Transport and Finite-Rate Chemistry, the species equations are Reynolds-averaged, which leads to unknown terms for the turbulent scalar flux and the mean reaction rate. The turbulent scalar flux is modeled in ANSYS FLUENT by gradient diffusion, treating turbulent convection as enhanced diffusion. The mean reaction rate can be modeled with the Laminar, Eddy-Dissipation or EDC Finite-Rate chemistry models. Since the reaction rate is invariably highly non-linear, modeling the mean reaction rate in a turbulent flow is difficult and prone to error.

An alternative to Reynolds-averaging the species and energy equations is to derive a transport equation for their single-point, joint probability density function (PDF). This PDF, denoted by P, can be considered to represent the fraction of the time that the fluid spends at each species, temperature and pressure state. P has $N+2$ dimensions for the N species, temperature and pressure spaces. From the PDF, any single-point thermo-chemical moment (e.g., mean or RMS temperature, mean reaction rate) can be calculated. The composition PDF transport equation is derived from the Navier-Stokes equations as [276]:

$$\frac{\partial}{\partial t}(\rho P) + \frac{\partial}{\partial x_i}(\rho u_i P) + \frac{\partial}{\partial \psi_k}(\rho S_k P) = -\frac{\partial}{\partial x_i}\left[\rho \langle u_i^{''} | \psi \rangle P\right] + \frac{\partial}{\partial \psi_k}\left[\rho \left\langle \frac{1}{\rho}\frac{\partial J_{i,k}}{\partial x_i} \middle| \psi \right\rangle P\right] \tag{11.2-1}$$

where

P	=	Favre joint PDF of composition
ρ	=	mean fluid density
u_i	=	Favre mean fluid velocity vector
S_k	=	reaction rate for specie k
ψ	=	composition space vector
$u_i^{''}$	=	fluid velocity fluctuation vector
$J_{i,k}$	=	molecular diffusion flux vector

The notation of $\langle \ldots \rangle$ denotes expectations, and $\langle A|B \rangle$ is the conditional probability of event A, given that event B occurs.

In Equation 11.2-1, the terms on the left-hand side are closed, while those on the right-hand side are not and require modeling. The first term on the left-hand side is the unsteady rate of change of the PDF, the second term is the change of the PDF due to convection by the mean velocity field, and the third term is the change due to chemical reactions. The principal strength of the PDF transport approach is that the highly-non-linear reaction term is completely closed and requires no modeling. The two terms on the right-hand side represent the PDF change due to scalar convection by turbulence (turbulent scalar flux), and molecular mixing/diffusion, respectively.

The turbulent scalar flux term is unclosed, and is modeled in ANSYS FLUENT by the gradient-diffusion assumption

$$-\frac{\partial}{\partial x_i}\left[\rho\langle u_i^{''}|\psi\rangle P\right] = \frac{\partial}{\partial x_i}\left(\frac{\rho\mu_t}{\mathrm{Sc}_t}\frac{\partial P}{\partial x_i}\right) \qquad (11.2\text{-}2)$$

where μ_t is the turbulent viscosity and Sc_t is the turbulent Schmidt number. A turbulence model, as described in Chapter 4: Turbulence, is required for composition PDF transport simulations, and this determines μ_t.

Since single-point PDFs are described, information about neighboring points is missing and all gradient terms, such as molecular mixing, are unclosed and must be modeled. The mixing model is critical because combustion occurs at the smallest molecular scales when reactants and heat diffuse together. Modeling mixing in PDF methods is not straightforward, and is the weakest link in the PDF transport approach. See Section 11.3.2: Particle Mixing for a description of the mixing models.

11.3 The Lagrangian Solution Method

A Lagrangian Monte Carlo method is used to solve for the $N + 1$ dimensional PDF Transport equation. Monte Carlo methods are appropriate for high-dimensional equations since the computational cost increases linearly with the number of dimensions. The disadvantage is that statistical errors are introduced, and these must be carefully controlled.

To solve the modeled PDF transport equation, an analogy is made with a stochastic differential equation (SDE) which has identical solutions. The Monte Carlo algorithm involves notional particles which move randomly through physical space due to particle convection, and also through composition space due to molecular mixing and reaction. The particles have mass and, on average, the sum of the particle masses in a cell equals the cell mass (cell density times cell volume). Since practical meshes have large changes in cell volumes, the particle masses are adjusted so that the number of particles in a cell is controlled to be approximately constant and uniform.

The processes of convection, diffusion, and reaction are treated in fractional steps as described in the sections that follow. For information on the fractional step method, refer to [45].

Information about this method is described in the following sections:

- Section 11.3.1: Particle Convection
- Section 11.3.2: Particle Mixing
- Section 11.3.3: Particle Reaction
- Section 11.3.4: The ISAT Algorithm

11.3.1 Particle Convection

A spatially second-order-accurate Lagrangian method is used in ANSYS FLUENT, consisting of two steps. At the first convection step, particles are advanced to a new position

$$x_i^{1/2} = x_i^0 + \frac{1}{2} u_i^0 \, \Delta t \tag{11.3-1}$$

where

x_i = particle position vector
u_i = Favre mean fluid-velocity vector at the particle position
Δt = particle time step

For unsteady flows, the particle time step is the physical time step. For steady-state flows, local time steps are calculated for each cell as

$$\Delta t = \min(\Delta t_{\rm conv}, \; \Delta t_{\rm diff}, \; \Delta t_{\rm mix}) \tag{11.3-2}$$

where

$\Delta t_{\rm conv}$ = *convection number* × Δx / *(cell fluid velocity)*
$\Delta t_{\rm diff}$ = *diffusion number* × $(\Delta x)^2$ / *(cell turbulent diffusivity)*
$\Delta t_{\rm mix}$ = *mixing number* × *turbulent time scale*
Δx = characteristic cell length = $volume^{1/D}$ where D is the problem dimension

After the first convection step, all other sub-processes, including diffusion and reaction are treated. Finally, the second convection step is calculated as

$$x_i^1 = x_i^{1/2} + \Delta t \left(u_i^{1/2} - \frac{1}{2} u_i^0 + \frac{1}{\rho {\rm Sc}_t} \frac{\partial \mu_t}{\partial x_i} + \xi_i \sqrt{\frac{2\mu_t}{\rho \Delta t {\rm Sc}_t}} \right) \tag{11.3-3}$$

where

ρ	=	mean cell fluid density
u_i	=	mean fluid-velocity vector at the particle position
μ_t	=	turbulent viscosity
Sc_t	=	turbulent Schmidt number
ξ_i	=	standardized normal random vector

11.3.2 Particle Mixing

Molecular mixing of species and heat must be modeled and is usually the source of the largest modeling error in the PDF transport approach. ANSYS FLUENT provides three models for molecular diffusion: the Modified Curl model [147, 250], the IEM model (which is sometimes called the LSME model) [75] and the EMST model [340].

The Modified Curl Model

For the Modified Curl model, a few particle pairs are selected at random from all the particles in a cell, and their individual compositions are moved toward their mean composition. For the special case of equal particle mass, the number of particle pairs selected is calculated as

$$N_{\text{pair}} = \frac{1.5 C_\phi N \Delta t}{\tau_t} \qquad (11.3\text{-}4)$$

where

N	=	total number of particles in the cell
C_ϕ	=	mixing constant (default = 2)
τ_t	=	turbulent time scale (for the k-ϵ model this is k/ϵ)

The algorithm in [250] is used for the general case of variable particle mass.

For each particle pair, a uniform random number ξ is selected and each particle's composition ϕ is moved toward the pair's mean composition by a factor proportional to ξ:

$$\begin{aligned}
\phi_i^1 &= (1-\xi)\phi_i^0 + \xi\frac{(\phi_i^0 m_i + \phi_j^0 m_j)}{(m_i + m_j)} \\
\phi_j^1 &= (1-\xi)\phi_j^0 + \xi\frac{(\phi_i^0 m_i + \phi_j^0 m_j)}{(m_i + m_j)} \qquad (11.3\text{-}5)
\end{aligned}$$

where ϕ_i and ϕ_j are the composition vectors of particles i and j, and m_i and m_j are the masses of particles i and j.

The IEM Model

For the Interaction by Exchange with the Mean (IEM) model, the composition of all particles in a cell are moved a small distance toward the mean composition:

$$\phi^1 = \phi^0 - \left(1 - e^{-0.5 C_\phi/\tau_t}\right)\left(\phi^0 - \tilde{\phi}\right) \tag{11.3-6}$$

where ϕ^0 is the composition before mixing, ϕ^1 is the composition after mixing, and $\tilde{\phi}$ is the Favre mean-composition vector at the particle's location.

The EMST Model

Physically, mixing occurs between fluid particles that are adjacent to each other. The Modified Curl and IEM mixing models take no account of this localness, which can be a source of error. The Euclidean Minimum Spanning Tree (EMST) model mixes particle pairs that are close to each other in composition space. Since scalar fields are locally smooth, particles that are close in composition space are likely to be close in physical space. The particle pairing is determined by a Euclidean Minimum Spanning Tree, which is the minimum length of the set of edges connecting one particle to at least one other particle. The EMST mixing model is more accurate than the Modified Curl and IEM mixing models, but incurs a slightly greater computational expense. Details on the EMST model can be found in reference [340].

Liquid Reactions

Reactions in liquids often occur at low turbulence levels (small Re), among reactants with low diffusivities (large Sc). For such flows, the mixing constant default of $C_\phi = 2$ overestimates the mixing rate. The Liquid Micro-Mixing option interpolates C_ϕ from model turbulence [278] and scalar [103] spectra.

11.3.3 Particle Reaction

The particle composition vector is represented as

$$\phi = (Y_1, Y_2, \ldots, Y_N, T, p) \tag{11.3-7}$$

where Y_k is the kth specie mass fraction, T is the temperature and p the pressure.

For the reaction fractional step, the reaction source term is integrated as

$$\phi^1 = \phi^0 + \int_0^{\Delta t} S dt \tag{11.3-8}$$

where S is the chemical source term. Most realistic chemical mechanisms consist of tens of species and hundreds of reactions. Typically, a reaction does not occur until an ignition temperature is reached, but then proceeds very quickly until reactants are consumed. Hence, some reactions have very fast time scales, in the order of 10^{-10} s, while others have much slower time scales, on the order of 1 s. This time-scale disparity results in numerical stiffness, which means that extensive computational work is required to integrate the chemical source term in Equation 11.3-8. In ANSYS FLUENT, the reaction step (i.e., the calculation of ϕ^1) can be performed either by Direct Integration or by In-Situ Adaptive Tabulation (ISAT), as described in the following paragraphs.

A typical steady-state PDF transport simulation in ANSYS FLUENT may have 50000 cells, with 20 particles per cell, and requires 1000 iterations to converge. Hence, at least 10^9 stiff ODE integrations are required. Since each integration typically takes tens or hundreds of milliseconds, on average, the direct integration of the chemistry is extremely CPU-demanding.

For a given reaction mechanism, Equation 11.3-8 may be considered as a mapping. With an initial composition vector ϕ^0, the final state ϕ^1 depends only on ϕ^0 and the mapping time Δt. In theory, if a table could be built before the simulation, covering all realizable ϕ^0 states and time steps, the integrations could be avoided by table look-ups. In practice, this *a priori* tabulation is not feasible since a full table in $N + 3$ dimensions (N species, temperature, pressure and time-step) is required. To illustrate this, consider a structured table with M points in each dimension. The required table size is M^{N+3}, and for a conservative estimate of $M = 10$ discretization points and $N = 7$ species, the table would contain 10^{10} entries.

On closer examination, the full storage of the entire realizable space is very wasteful because most regions are never accessed. For example, it would be unrealistic to find a composition of $Y_{\rm OH} = 1$ and $T = 300K$ in a real combustor. In fact, for steady-state, 3D laminar simulations, the chemistry can be parameterized by the spatial position vector. Thus, mappings must lie on a three dimensional manifold within the $N + 3$ dimensional composition space. It is, hence, sufficient to tabulate only this accessed region of the composition space.

The accessed region, however, depends on the particular chemical mechanism, molecular transport properties, flow geometry, and boundary conditions. For this reason, the accessed region is not known before the simulation and the table cannot be preprocessed. Instead, the table must be built during the simulation, and this is referred to as *in-situ* tabulation. ANSYS FLUENT employs ISAT [277] to dynamically tabulate the chemistry mappings and accelerate the time to solution. ISAT is a method to tabulate the accessed composition space region “on the fly” (in-situ) with error control (adaptive tabulation). When ISAT is used correctly, accelerations of two to three orders of magnitude are typical. However, it is important to understand how ISAT works in order to use it optimally.

11.3.4 The ISAT Algorithm

ISAT is a powerful tool that enables realistic chemistry to be incorporated in multi-dimensional flow simulations by accelerating the chemistry calculations. Typical speed-ups of 100-fold are common. This power is apparent if one considers that with a 100-fold speed-up, a simulation that would take three months without ISAT can be run in one day.

At the start of an ANSYS FLUENT simulation using ISAT, the ISAT table is empty. For the first reaction step, Equation 11.3-8 is integrated with a stiff ODE solver. This is called Direct Integration (DI). The first table entry is created and consists of:

- the initial composition ϕ^0
- the mapping ϕ^1
- the mapping gradient matrix $A = \partial\phi^1/\partial\phi^0$
- a hyper-ellipsoid of accuracy

The next reaction mapping is calculated as follows: The initial composition vector for this particle is denoted ϕ_q^0, where the subscript q denotes a *query*. The existing table (consisting of one entry at this stage) is queried by interpolating the new mapping as

$$\phi_q^1 = \phi^1 + A(\phi_q^0 - \phi^0) \qquad (11.3\text{-}9)$$

The mapping gradient is hence used to linearly interpolate the table when queried. The ellipsoid of accuracy (EOA) is the elliptical space around the table point ϕ^0 where the linear approximation to the mapping is accurate to the specified tolerance, $\epsilon_{\rm tol}$.

If the query point ϕ_q^1 is within the EOA, then the linear interpolation by Equation 11.3-9 is sufficiently accurate, and the mapping is *retrieved*. Otherwise, a direct integration (DI) is performed and the mapping error $\epsilon = |B(\phi_{DI}^1 - \phi_q^1)|$ is calculated (here, B is a scaling matrix). If this error is smaller than the specified error tolerance ($\epsilon < \epsilon_{tol}$), then the original interpolation ϕ_q^1 is accurate and the EOA is *grown* so as to include ϕ_q^0. If not, a new table entry is *added*.

Table entries are stored as leaves in a binary tree. When a new table entry is *added*, the original leaf becomes a node with two leaves—the original leaf and the new entry. A cutting hyper-plane is created at the new node, so that the two leaves are on either side of this cutting plane. A composition vector ϕ_q^0 will hence lie on either side of this hyper-plane.

The ISAT algorithm is summarized as follows:

1. The ISAT table is *queried* for every composition vector during the reaction step.
2. For each query ϕ_q^0 the table is traversed to identify a leaf whose composition ϕ^0 is close to ϕ_q^0.
3. If the query composition ϕ_q^0 lies within the EOA of the leaf, then the mapping ϕ_q^1 is *retrieved* using interpolation by Equation 11.3-9. Otherwise, Direct Integration (DI) is performed and the error ϵ between the DI and the linear interpolation is measured.
4. If the error ϵ is less than the tolerance, then the ellipsoid of accuracy is *grown* and the DI result is returned. Otherwise, a new table entry is *added*.

At the start of the simulation, most operations are *adds* and *grows*. Later, as more of the composition space is tabulated, *retrieves* become frequent. Since *adds* and *grows* are very slow whereas *retrieves* are relatively quick, initial ANSYS FLUENT iterations are slow but accelerate as the table is built.

11.4 The Eulerian Solution Method

The Lagrangian solution method solves the composition PDF transport equation by stochastically tracking Lagrangian particles through the domain. It is computationally expensive since a large number of particles are required to represent the PDF, and a large number of iterations are necessary to reduce statistical errors and explicitly convect the particles through the domain. The Eulerian PDF transport model overcomes these limitations by assuming a shape for the PDF, which allows Eulerian transport equations to be derived. Stochastic errors are eliminated and the transport equations are solved implicitly, which is computationally economical. The multi-dimensional PDF shape is assumed as a product of delta functions. As with the Lagrangian PDF model, the highly non-linear chemical source term is closed. However, the turbulent scalar flux and molecular mixing terms must be modeled, and are closed with the gradient diffusion and the IEM models, respectively.

The composition PDF of $N_s + 1$ dimension (N_s species and enthalpy) is represented as a collection of N_e delta functions (or modes). This presumed PDF has the following form:

$$P(\psi; \vec{x}, t) = \sum_{n=1}^{N_e} p_n(\vec{x}, t) \prod_{k=1}^{N_s} \delta[\psi_k - <\phi_k>_n (\vec{x}, t)] \tag{11.4-1}$$

where p_n is the probability in each mode, $<\phi_k>_n$ is the conditional mean composition of specie k in the nth mode, ψ_k is the composition space variable of specie k, and $\delta(\ldots)$ is the delta function.

The Eulerian PDF transport equations are derived by substituting Equation 11.4-1 into the closed composition PDF transport equation (Equation 11.2-1 with Equations 11.2-2 and 11.3-6). The unknown terms, p_n and $< \phi_k >_n$, are determined by forcing lower moments of this transported PDF to match the RANS lower moment transport equations, using the Direct Quadrature Method of Moments (DQMOM) approach [103, 218]. The resulting transport equations are:

- Probability (magnitude of the nth delta function):

$$\frac{\partial \rho p_n}{\partial t} + \frac{\partial}{\partial x_i}(\rho u_i p_n) = \nabla(\rho \Gamma \nabla p_n) \tag{11.4-2}$$

- Probability weighted conditional mean of composition k:

$$\frac{\partial \rho s_{k,n}}{\partial t} + \frac{\partial}{\partial x_i}(\rho u_i s_{k,n}) = \nabla(\rho \Gamma \nabla s_{k,n}) + \rho(M_{k,n} + S_{k,n} + C_{k,n}) \tag{11.4-3}$$

where p_n is the probability of the nth mode, and $s_{k,n} = p_n < \phi_k >_n$ is the kth specie probability weighted conditional mean composition of the nth mode. $\Gamma = \mu_l + \mu_t/\mathrm{Sc}_t$ is the effective turbulent diffusivity. The terms $M_{k,n}$, $S_{k,n}$ and $C_{k,n}$ represent mixing, reaction and correction terms respectively. Note that only $N_e - 1$ probability equations are solved and the Nth probability is calculated as one minus the sum of the $N_e - 1$ solved probabilities.

Reaction

The reaction source term $S_{k,n}$ in Equation 11.4-3 for the kth composition and the nth mode is calculated as,

$$S_{k,n} = p_n S(< \phi_k >_n)_k \tag{11.4-4}$$

where $S()_k$ is the net reaction rate for the kth component.

Mixing

The micro-mixing term $M_{k,n}$ is modeled with the IEM mixing model:

$$M_{k,n} = \frac{C_\phi}{\tau}(< \phi_k >_n - \psi_k) \tag{11.4-5}$$

where τ is the turbulence time-scale and C_ϕ is the mixing constant.

Hence, for the two-mode DQMOM-IEM model, the mixing terms for component k are,

$$\begin{aligned} M_{k,1} &= \frac{C_\phi}{\tau}(p_1 s_{k,2} - p_2 s_{k,1}) \\ M_{k,2} &= \frac{C_\phi}{\tau}(p_2 s_{k,1} - p_1 s_{k,2}) \end{aligned} \tag{11.4-6}$$

The default value of C_ϕ is 2, which is appropriate for gas-phase combustion. For reactions in liquids, where the diffusivities are much smaller than gases, the Liquid Micro-Mixing option interpolates C_ϕ from model turbulence [278] and scalar [103] spectra.

Correction

Using assumptions to ensure realizability and boundedness, the correction terms $C_{k,n}$ in Equation 11.4-3 for the kth composition are determined from the linear system,

$$\sum_{n=1}^{N_e} < \phi_k >_n^{m_k-1} C_{k,n} = \sum_{n=1}^{N_e} (m_k - 1) < \phi_k >_n^{m_k-2} p_n c_{k,n} \tag{11.4-7}$$

where m_k are the non-negative integer lower moments $(1..N_e)$ for each component k. Note that the condition of the matrix decreases with increasing m_k, which reduces the stability of higher mode simulations.

The dissipation term $c_{k,n}$ in Equation 11.4-7 is calculated as,

$$c_{k,n} = \Gamma \nabla (< \phi_k >_n) \cdot \nabla (< \phi_k >_n) \tag{11.4-8}$$

For the two-mode DQMOM-IEM model, the correction terms for the kth component are,

$$\begin{aligned} C_{k,1} &= \frac{\Gamma}{< \phi_k >_1 - < \phi_k >_2} \left(p_1 (\nabla < \phi_k >_1)^2 + p_2 (\nabla < \phi_k >_2)^2 \right) \\ C_{k,2} &= -C_{k,1} \end{aligned} \tag{11.4-9}$$

Calculation of Composition Mean and Variance

The mean composition (specie k or energy) is calculated as,

$$\overline{\phi_k} = \sum_{n=1}^{N_e} s_{k,n} \tag{11.4-10}$$

and its variance is calculated as

$$\overline{\phi_k'^2} = (\sum_{n=1}^{N_e} s_{k,n} < \phi_k >_n) - \overline{\phi_k}^2 \tag{11.4-11}$$

Chapter 12. Engine Ignition

This chapter discusses the theory behind the engine ignition models available in ANSYS FLUENT. Information can be found in the following sections.

- Section 12.1: Spark Model
- Section 12.2: Autoignition Models
- Section 12.3: Crevice Model

For information about using these ignition models, see Chapter 20: Modeling Engine Ignition in the separate User's Guide.

12.1 Spark Model

The spark model in ANSYS FLUENT will be described in the context of the premixed turbulent combustion model. For information about using this model, see Section 20.1: Spark Model in the separate User's Guide. Information regarding the theory behind this model is detailed in the following sections:

- Section 12.1.1: Overview and Limitations
- Section 12.1.2: Spark Model Theory

12.1.1 Overview and Limitations

Initiation of combustion at a desired time and location in a combustion chamber can be accomplished by sending a high voltage across two narrowly separated wires, creating a spark. The spark event in typical engines happens very quickly relative to the main combustion in the engine. The physical description of this simple event is very involved and complex, making it difficult to accurately model the spark in the context of a multi-dimensional engine simulation. Additionally, the energy from the spark event is several orders of magnitude less than the chemical energy release from the fuel. Despite the amount of research devoted to spark ignition physics and ignition devices, the ignition of a mixture at a point in the domain is more dependent on the local composition than on the spark energy (see Heywood [128]). Thus, for situations in which ANSYS FLUENT is utilized for combustion engine modeling, including internal combustion engines, the spark event does not need to be modeled in great detail, but simply as the initiation of combustion over a duration, which you will set.

Since spark ignition is inherently transient, the spark model is only available in the transient solver. Additionally, the spark model requires chemical reactions to be solved. The spark model is available for all of the combustion models, however, it may be most applicable to the premixed and partially premixed combustion models.

The Spark Model used in ANSYS FLUENT is based on a one-dimensional analysis by Lipatnikov [203]. The model is sensitive to perturbations and can be subject to instabilities when used in multi-dimensional simulations. The instabilities are inherent to the model and can be dependent on the mesh, especially near the beginning of the spark event when the model reduces diffusion to simulate the initial laminar spark kernel growth. The instability is susceptible to numerical errors which are increased when the mesh is not aligned with the flame propagation. As the spark kernel grows and the model allows turbulent mixing to occur, the effect of the instability decreases.

12.1.2 Spark Model Theory

The spark model in ANSYS FLUENT is based on the work done by Lipatnikov [203] and extended to other combustion models. The derivation of the model can be done in the context of the Zimont premixed combustion model.

Zimont Premixed Flame Model

The transport equation for the mean reaction progress variable, c, is given by Equation 12.1-1

$$\frac{\partial \rho c}{\partial t} + \nabla \cdot (\rho \vec{v} c) = \nabla \cdot (D_t \nabla c) + \rho_u U_t |\nabla c| \tag{12.1-1}$$

where D_t is the turbulent diffusivity, ρ_u is the density of the unburned mixture and U_t is the turbulent flame speed. Since the spark is often very small compared to the mesh size of the model and is often laminar in nature, the Zimont model is modified such that

$$\frac{\partial \rho c}{\partial t} + \nabla \cdot (\rho \vec{v} c) = \nabla \cdot ((\kappa + D_{tt}) \nabla c) + \rho_u U_t |\nabla c| \tag{12.1-2}$$

where κ is the laminar thermal diffusivity and the effective diffusivity D_{tt} is given by

$$D_{tt} = \begin{cases} D_t \left(1 - \exp\left(\frac{-t_{td}}{\tau'}\right)\right) & \text{if } t_{td} \geq 0 \\ D_t & \text{if } t_{td} < 0 \end{cases} \tag{12.1-3}$$

where is $t_{td} = t - t_{ig}$ and t_{ig} denotes the time at which the spark is initiated. Additionally, τ' is an effective diffusion time, which you can set.

Only turbulent scales that are smaller than the spark radius can contribute to turbulent spark diffusion, so the expression for the effective turbulent diffusivity, D_{tt}, is ramped up as the spark grows. This creates higher temperatures at the location of the spark and can cause convergence difficulties. In addition to convergence difficulties, small changes in the diffusion time can change the result significantly. Because of these issues, the diffusion time can be controlled by the you, and has a default value of 1e-5 seconds.

Other Combustion Models

The spark model is compatible with all combustion models in ANSYS FLUENT. However, the premixed and partially premixed models differ in that the progress variable inside the spark region is set equal to 1, a burned state, for the duration of the spark event. Other combustion models have the energy input into the cell. If the temperature exceeds 2500 K or the spark duration is exceeded, no energy from the spark model will be added to the spark cells.

The spark model can be used in models other than the premixed and partially premixed combustion models, however, you must balance energy input and diffusivity to produce a high enough temperature to initiate combustion, which can be a nontrivial undertaking. The model's use has been extended to be compatible with the other models, however, in some cases it simply creates a high temperature region and does not guarantee the initiation of combustion.

12.2 Autoignition Models

Autoignition phenomena in engines are due to the effects of chemical kinetics of the reacting flow inside the cylinder. There are two types of autoignition models considered in ANSYS FLUENT:

- knock model in spark-ignited (SI) engines
- ignition delay model in diesel engines

For information regarding using autoignition models, see Section 20.2: Autoignition Models in the separate User's Guide. The theory behind the autoignition models is described in the following sections:

- Section 12.2.1: Overview and Limitations
- Section 12.2.2: Ignition Model Theory

12.2.1 Overview and Limitations

Overview

The concept of knock has been studied extensively in the context of premixed engines, as it defines a limit in terms of efficiency and power production of that type of engine. As the compression ratio increases, the efficiency of the engine as a function of the work extracted from the fuel increases.

However, as the compression ratio increases, the temperature and pressure of the air/fuel mixture in the cylinder also increase during the cycle compressions. The temperature and pressure increase can be large enough for the mixture to spontaneously ignite and release its heat before the spark plug fires. The premature release of all of the energy in the air/fuel charge is almost never desirable, as this results in the spark event no longer controlling the combustion. As a result of the premature release of the energy, catastrophic damage to the engine components can occur. The sudden, sharp rise in pressure inside the engine can be heard clearly through the engine block as a knocking sound, hence the term “knock”. For commonly available gasoline pumps, knock usually limits the highest practical compression ratio to less than 11:1 for premium fuels and around 9:1 for less expensive fuels.

By comparison, ignition delay in diesel engines has not been as extensively studied as SI engines, mainly because it does not have such a sharply defining impact on engine efficiency. Ignition delay in diesel engines refers to the time between when the fuel is injected into the combustion chamber and when the pressure starts to increase as the fuel releases its energy. The fuel is injected into a gas which is usually air, however, it can have a considerable amount of exhaust gas mixed in (or EGR) to reduce nitrogen oxide emissions (NOx). Ignition delay depends on the composition of the gas in the cylinder, the temperature of the gas, the turbulence level, and other factors. Since ignition delay changes the combustion phasing, which in turn impacts efficiency and emissions, it is important to account for it in a diesel engine simulation.

Model Limitations

The main difference between the knock model and the ignition delay model is the manner in which the model is coupled with the chemistry. The knock model always releases energy from the fuel while the ignition delay model prevents energy from being released prematurely.

The knock model in ANSYS FLUENT is compatible with the premixed and partially premixed combustion models. The autoignition model is compatible with any volumetric combustion model, with the exception of the purely premixed models. The autoignition models are inherently transient and so are not available with steady simulations.

The autoignition models in general require adjustment of parameters to reproduce engine data and are likely to require tuning to improve accuracy. Once the model is calibrated to a particular engine configuration, then different engine speeds and loads can be reasonably well represented. Detailed chemical kinetics may be more applicable over a wider range of conditions, though are more expensive to solve. The single equation autoignition models are appropriate for the situation where geometric fidelity or resolution of particular flow details is more important than chemical effects on the simulation.

12.2.2 Ignition Model Theory

Both the knock and the ignition delay models are treated similarly in ANSYS FLUENT, in that they share the same infrastructure. These models belong to the family of *single equation* autoignition models and use correlations to account for complex chemical kinetics. They differ from the eight step reaction models, such as Halstead's "Shell" model [121], in that only a single transport equation is solved. The source term in the transport equation is typically not stiff, thus making the equation relatively inexpensive to solve.

This approach is appropriate for large simulations where geometric accuracy is more important than fully resolved chemical kinetics. The model can be used on less resolved meshes to explore a range of designs quickly, and to obtain trends before utilizing more expensive and presumably more accurate chemical mechanisms in multidimensional simulations.

Transport of Ignition Species

Autoignition is modeled using the transport equation for an *Ignition Species*, Y_{ig}, which is given by

$$\frac{\partial \rho Y_{ig}}{\partial t} + \nabla \cdot (\rho \vec{v} Y_{ig}) = \nabla \cdot \left(\frac{\mu_t}{\mathrm{Sc}_t} \nabla Y_{ig} \right) + \rho S_{ig} \tag{12.2-1}$$

where Y_{ig} is a "mass fraction" of a passive species representing radicals which form when the fuel in the domain breaks down. Sc_t is the turbulent Schmidt number. The term S_{ig} is the source term for the ignition species which has a form

$$S_{ig} = \int_{t=t_0}^{t} \frac{dt}{\tau_{ig}}$$

where t_0 corresponds to the time at which fuel is introduced into the domain. The τ_{ig} term is a correlation of ignition delay with the units of time. Ignition has occurred when the ignition species reaches a value of 1 in the domain. It is assumed that all the radical species represented by Y_{ig} diffuse at the same rate as the mean flow.

Note that the source term for these radical species is treated differently for knock and ignition delay. Furthermore, the form of the correlation of ignition delay differs between the two models. Details of how the source term is treated are covered in the following sections.

Knock Modeling

When modeling knock or ignition delay, chemical energy in the fuel is released when the ignition species reaches a value of 1 in the domain. For the knock model, two correlations are built into ANSYS FLUENT. One is given by Douaud [76], while the other is a generalized model which reproduces several correlations, given by Heywood [128].

Modeling of the Source Term

In order to model knock in a physically realistic manner, the source term is accumulated under appropriate conditions in a cell. Consider the one dimensional flame in Figure 12.2.1. Here, the flame is propagating from left to right, and the temperature is relatively low in front of the flame and high behind the flame. In this figure, T_b and T_u represent the temperatures at the burned and unburned states, respectively. The ignition species accumulates only when there is fuel. In the premixed model, the fuel is defined as $fuel = 1 - c$, where c is the progress variable. If the progress variable has a value of zero, the mixture is considered unburned. If the progress variable is 1, then the mixture is considered burned.

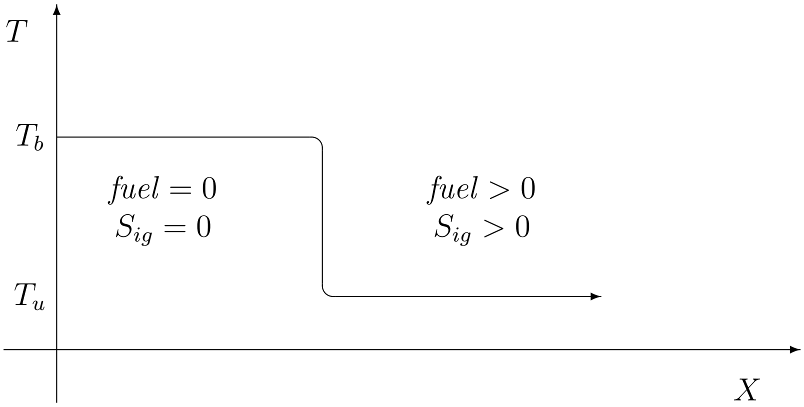


Figure 12.2.1: Flame Front Showing Accumulation of Source Terms for the Knock Model

When the ignition species reaches a value of 1 in the domain, knock has occurred at that point. The value of the ignition species can exceed unity. In fact, values well above that can be obtained in a short time. The ignition species will continue to accumulate until there is no more fuel present.

Correlations

An extensively tested correlation for knock in SI engines is given by Douaud and Eyzat [76]:

$$\tau = 0.01768 \left(\frac{ON}{100}\right)^{3.402} p^{-1.7} \exp\left(\frac{3800}{T}\right) \tag{12.2-2}$$

where ON is the octane number of the fuel, p is the absolute pressure in atmospheres and T is the temperature in Kelvin.

A generalized expression for τ is also available which can reproduce many existing Arrhenius correlations. The form of the correlation is

$$\tau = A \left(\frac{ON}{100}\right)^{a} p^{b} T^{c} \mathrm{RPM}^{d} \Phi^{d} \exp\left(\frac{-E_a}{RT}\right) \tag{12.2-3}$$

where A is the pre-exponential (with units in seconds), RPM is the engine speed in cycles per minute and Φ is the fuel/air equivalence ratio.

Energy Release

Once ignition has occurred in the domain, the knock event is modeled by releasing the remaining fuel energy with a single-step Arrhenius reaction. An additional source term, which burns the remaining fuel in that cell, is added to the rate term in the premixed model. The reaction rate is given by

$$\dot{\omega} = A_0 \exp \frac{-E_a}{RT} \tag{12.2-4}$$

where $A_0 = 8.6 \times 10^9$, and $E_a = -15078$. These values are chosen to reflect single-step reaction rates appropriate for propane as described in Amsden [4]. The rate at which the fuel is consumed is limited such that a completely unburned cell will burn during three of the current time steps. Limiting the reaction rate is done purely for numerical stability.

Ignition Delay Modeling

When modeling ignition delay in diesel engines, chemical reactions are allowed to occur when the ignition species reaches a value of 1 in the domain. For the ignition delay model, two correlations are built into ANSYS FLUENT, one given by Hardenburg and Hase [125] and the other, a generalized model which reproduces several Arrhenius correlations from the literature.

If the ignition species is less than 1 when using the ignition delay model, the chemical source term is suppressed by not activating the combustion model at that particular time step; thus, the energy release is delayed. This approach is reasonable if you have a good high-temperature chemical model, but does not wish to solve for typically expensive low temperature chemistry.

Modeling of the Source Term

In order to model ignition in a physically realistic manner, the source term is accumulated under appropriate conditions in a cell. Consider the one dimensional spray in Figure 12.2.2. Here, the spray is propagating from left to right and the fuel mass fraction is relatively low in front of the spray and high behind the spray. If there is no fuel in the cell, the model will set the local source term to zero, nevertheless, the value of Y_{ig} can be nonzero due to convection and diffusion.

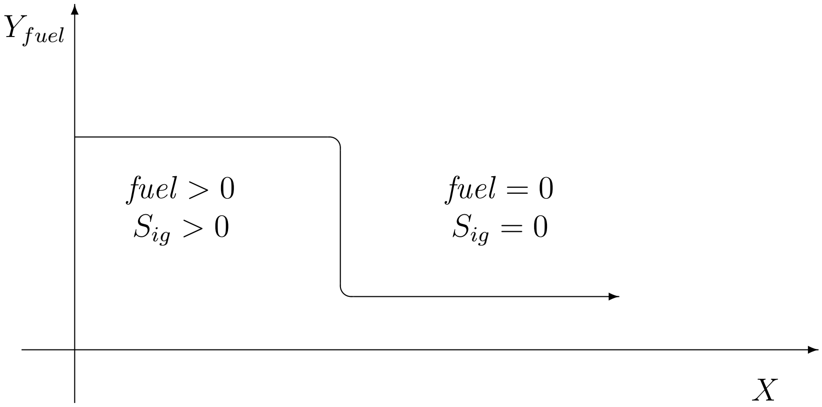


Figure 12.2.2: Propagating Fuel Cloud Showing Accumulation of Source Terms for the Ignition Delay Model

Correlations

If fuel is present in the cell, there are two built-in options in ANSYS FLUENT to calculate the local source term. The first correlation was done by Hardenburg and Hase and was developed at Daimler Chrysler for heavy duty diesel engines. The correlation works over a reasonably wide range of conditions and is given by

$$\tau_{id} = \left(\frac{C_1 + 0.22\overline{S}_p}{6N} \right) \exp \left[E_a \left(\frac{1}{RT} - \frac{1}{17,190} \right) + \left(\frac{21.2}{p - 12.4} \right)^{e_p} \right] \qquad (12.2\text{-}5)$$

where τ_{id} is in seconds, C_1 is 0.36, N is engine speed in revolutions per minute, E_a is the effective activation energy and e_p is the pressure exponent. The expression for the effective activation energy is given by

$$E_a = \frac{E_{hh}}{CN + 25} \tag{12.2-6}$$

where CN is the cetane number. The activation energy, E_{hh}, pre-exponential, C_1, pressure exponent, e_p, and cetane number, CN, are accessible from the GUI. The default values of these variables are listed in the table below.

Table 12.2.1: Default Values of the Variables in the Hardenburg Correlation

Variable	E_{hh}	CN	C_1	e_p
Default	618,840	25	0.36	0.63

The second correlation, which is the generalized correlation, is given by Equation 12.2-3 and is available for ignition delay calculations.

Energy Release

If the ignition species is greater than or equal to 1 anywhere in the domain, ignition has occurred and combustion is no longer delayed. The ignition species acts as a switch to turn on the volumetric reactions in the domain. Note that the ignition species “mass fraction” can exceed 1 in the domain, therefore, it is not truly a mass fraction, but rather a passive scalar which represents the integrated correlation as a function of time.

12.3 Crevice Model

This section describes the theory behind the crevice model. Information can be found in the following sections:

- Section 12.3.1: Overview
- Section 12.3.2: Limitations
- Section 12.3.3: Crevice Model Theory

For information regarding using the crevice model, see Section 20.3: Crevice Model in the separate User’s Guide.

12.3.1 Overview

The crevice model implemented in ANSYS FLUENT is a zero-dimensional ring-flow model based on the model outlined in Namazian and Heywood [245] and Roberts and Matthews [295]. The model is geared toward in-cylinder specific flows, and more specifically, direct-injection (DI) diesel engines, and thus is available only for time-dependent simulations.

The model takes mass, momentum, and energy from cells adjoining two boundaries and accounts for the storage of mass in the volumes of the crevices in the piston. Detailed geometric information regarding the ring and piston—typically a ring pack around the bore of an engine—is necessary to use the crevice model. An example representation is shown in Figures 12.3.1–12.3.3.

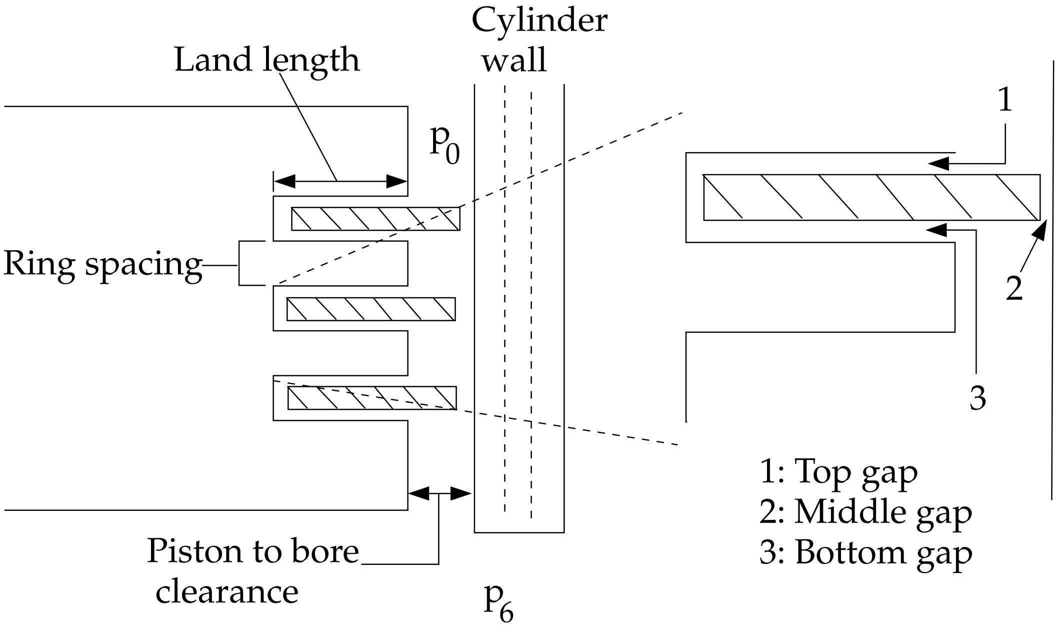


Figure 12.3.1: Crevice Model Geometry (Piston)

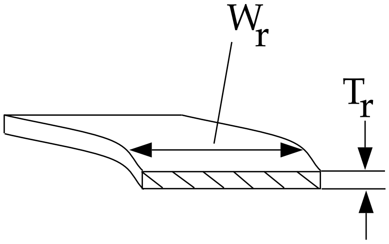


Figure 12.3.2: Crevice Model Geometry (Ring)

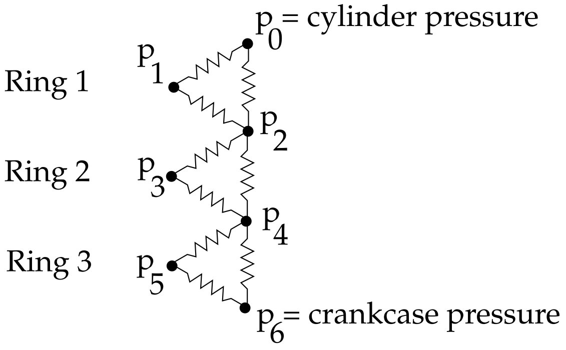


Figure 12.3.3: Crevice Model "Network" Representation

Model Parameters

- The *piston to bore clearance* is the distance between the piston and the bore. Typical values are 2 to 5 mil (80 to 120 μm) in a spark engine (SI) and 4 to 7 mil (100 to 240 μm) in some diesel engines (DI).
- The *ring thickness* is the variable T_r in Figure 12.3.2. Typical values range from 1 to 3 mm for SI engines and 2 to 4 mm for DI engines.
- The *ring width* is the variable W_r in Figure 12.3.2. Typical values range from 3 to 3.5 mm for SI engines and 4 to 6 mm for DI diesel engines.
- The *ring spacing* is the distance between the bottom of one ring land and the top of the next ring land. Typical values of the ring spacing are 3 to 5 mm for SI engines and 4 to 8 mm for DI diesel engines.
- The *land length* is the depth of the ring land (i.e., the cutout into the piston); always deeper than the width of the ring by about 1 mm. Typical values are 4 to 4.5 mm for SI engines and 5 to 7 mm for DI diesel engines.
- The *top gap* is the clearance between the ring land and the top of the ring (40 to 80 μm).
- The *middle gap* is the distance between the ring and the bore (10 to 40 μm).
- The *bottom gap* is the clearance between the ring land and the bottom of the ring (40 to 80 μm).
- The *shared boundary and leaking wall* is the piston (e.g., `wall-8`) and the cylinder wall (e.g., `wall.1`) in most in-cylinder simulations. Cells that share a boundary with the top of the piston and the cylinder wall are defined as the *crevice* cells.

The ring pack is the set of rings that seal the piston in the cylinder bore. As the piston moves upward in the cylinder when the valves are closed (e.g., during the compression stroke in a four-stroke cycle engine), the pressure in the cylinder rises and flow begins to move past the rings. The pressure distribution in the ring pack is modeled by assuming either fully-developed compressible flow through the spaces between the rings and the piston, or choked compressible flow between the rings and the cylinder wall.

Since the temperature in the ring pack is fixed and the geometry is known, once a pressure distribution is calculated, the mass in each volume can be found using the ideal gas equation of state. The overall mass flow out of the ring pack (i.e., the flow past the last ring specified) is also calculated at each discrete step in the ANSYS FLUENT solution.

12.3.2 Limitations

The limitations of the crevice model are that it is zero dimensional, transient, and currently limited to two threads that share a boundary.

A zero-dimensional approach is used because it is difficult to accurately predict lateral diffusion of species in the crevice. If the lateral diffusion of species is important in the simulation, as in when a spray plume in a DI engine is in close proximity to the boundary and the net mass flow is into the crevice, it is recommended that the full multidimensional crevice geometry be simulated in ANSYS FLUENT using a nonconformal mesh. Additionally, this approach does not specifically track individual species, as any individual species would be instantly distributed over the entire ring pack. The mass flux into the domain from the crevice is assumed to have the same composition as the cell into which mass is flowing.

The formulation of the crevice flow equations is inherently transient and is solved using ANSYS FLUENT's stiff-equation solver. A steady problem with leakage flow can be solved by running the transient problem to steady state. Additional limitations of the crevice model in its current form are that only a single crevice is allowed and only one thread can have leakage. Ring dynamics are not explicitly accounted for, although ring positions can be set during the simulation.

In this context, the crevice model solution is a stiff initial boundary-value problem. The stiffness increases as the pressure difference between the ring crevices increases and also as the overall pressure difference across the ring pack increases. Thus, if the initial conditions are very far from the solution during a time step, the ODE solver may not be able to integrate the equations successfully. One solution to this problem is to decrease the flow time step for several iterations. Another solution is to start with initial conditions that are closer to the solution at the end of the time step.

12.3.3 Crevice Model Theory

ANSYS FLUENT solves the equations for mass conservation in the crevice geometry by assuming laminar compressible flow in the region between the piston and the top and bottom faces of the ring, and by assuming an orifice flow between the ring and the cylinder wall. The equation for the mass flow through the ring end gaps is of the form

$$\dot{m}_{ij} = C_d A_{ij} \rho c \eta_{ij} \tag{12.3-1}$$

where C_d is the discharge coefficient, A_{ij} is the gap area, ρ is the gas density, c is the local speed of sound, and η_{ij} is a compressibility factor given by

$$\eta_{ij} = \begin{cases} \frac{2}{\gamma-1}\left[\left(\frac{p_i}{p_j}\right)^{\frac{2}{\gamma}} - \left(\frac{p_i}{p_j}\right)^{\frac{\gamma-1}{\gamma}}\right]^{0.5} & \frac{p_i}{p_j} > 0.52 \\ \left(\frac{2}{\gamma-1}\right)^{\frac{\gamma+1}{2(\gamma-1)}} & \frac{p_i}{p_j} \leq 0.52 \end{cases} \tag{12.3-2}$$

where γ is the ratio of specific heats, p_i the upstream pressure and p_j the downstream pressure. The equation for the mass flow through the top and bottom faces of the ring (i.e., into and out of the volume behind the piston ring) is given by

$$\dot{m}_{ij} = \frac{h_{ij}^2 \left(p_i^2 - p_j^2\right) A_{ij}}{24 W_r \mu_{\rm gas} R T} \tag{12.3-3}$$

where h_{ij} is the cross-sectional area of the gap, W_r is the width of the ring along which the gas is flowing, $\mu_{\rm gas}$ is the local gas viscosity, T is the temperature of the gas and R is the universal gas constant. The system of equations for a set of three rings is of the following form:

$$\frac{dp_1}{dt} = \frac{p_1}{m_1}\left(\dot{m}_{01} - \dot{m}_{12}\right) \tag{12.3-4}$$

$$\frac{dp_2}{dt} = \frac{p_2}{m_2}\left(\dot{m}_{02} + \dot{m}_{12} - \dot{m}_{23} - \dot{m}_{24}\right) \tag{12.3-5}$$

$$\frac{dp_3}{dt} = \frac{p_3}{m_3}\left(\dot{m}_{23} - \dot{m}_{34}\right) \tag{12.3-6}$$

$$\frac{dp_4}{dt} = \frac{p_4}{m_4}\left(\dot{m}_{24} + \dot{m}_{34} - \dot{m}_{45} - \dot{m}_{46}\right) \tag{12.3-7}$$

$$\frac{dp_5}{dt} = \frac{p_5}{m_5}\left(\dot{m}_{45} - \dot{m}_{56}\right) \tag{12.3-8}$$

where p_0 is the average pressure in the crevice cells and p_6 is the crankcase pressure input from the text interface. The expressions for the mass flows for numerically adjacent zones (e.g., 0-1, 1-2, 2-3, etc.) are given by Equation 12.3-3 and expressions for the mass flows for zones separated by two integers (e.g., 0-2, 2-4, 4-6) are given by Equations 12.3-1 and 12.3-2. Thus, there are $2n_r - 1$ equations needed for the solution to the ring-pack equations, where n_r is the number of rings in the simulation.

Chapter 13. Pollutant Formation

This chapter discusses the theory behind the models available in ANSYS FLUENT for modeling pollutant formation.

Information is presented in the following sections:

For information about using the models in ANSYS FLUENT, see Chapter 21: Modeling Pollutant Formation in the separate User's Guide.

13.1 NOx Formation

The following sections present the theoretical background of NO_x prediction. For information about using the NO_x models in ANSYS FLUENT, see Section 21.1.1: Using the NO_x Model in the separate User's Guide.

13.1.1 Overview

NO_x emission consists of mostly nitric oxide (NO), and to a lesser degree nitrogen dioxide (NO_2) and nitrous oxide (N_2O). NO_x is a precursor for photochemical smog, contributes to acid rain, and causes ozone depletion. Thus, NO_x is a pollutant. The ANSYS FLUENT NO_x model provides a tool to understand the sources of NO_x production and to aid in the design of NO_x control measures.

NOx Modeling in ANSYS FLUENT

The ANSYS FLUENT NO_x model provides the capability to model thermal, prompt, and fuel NO_x formation as well as NO_x consumption due to reburning in combustion systems. It uses rate models developed at the Department of Fuel and Energy at The University of Leeds in England as well as from the open literature. NO_x reduction using reagent injection, such as selective noncatalytic reduction (SNCR), can be modeled in ANSYS FLUENT along with an N_2O intermediate model which has also been incorporated.

To predict NO_x emissions, ANSYS FLUENT solves a transport equation for nitric oxide (NO) concentration. When fuel NO_x sources are present, ANSYS FLUENT solves additional transport equations for intermediate species (HCN and/or NH_3). When the N_2O intermediate model is activated, an additional transport equation for N_2O will be solved. The NO_x transport equations are solved based on a given flow field and combustion solution. In other words, NO_x is postprocessed from a combustion simulation. It is thus evident that an accurate combustion solution becomes a prerequisite of NO_x prediction. For example, thermal NO_x production doubles for every 90 K temperature increase when the flame temperature is about 2200 K. Great care must be exercised to provide accurate thermophysical data and boundary condition inputs for the combustion model. Appropriate turbulence, chemistry, radiation and other submodels must be employed.

To be realistic, one can only expect results to be as accurate as the input data and the selected physical models. Under most circumstances, NO_x variation trends can be accurately predicted but the NO_x quantity itself cannot be pinpointed. Accurate prediction of NO_x parametric trends can cut down on the number of laboratory tests, allow more design variations to be studied, shorten the design cycle, and reduce product development cost. That is truly the power of the ANSYS FLUENT NO_x model and, in fact, the power of CFD in general.

NOx Formation and Reduction in Flames

In laminar flames, and at the molecular level within turbulent flames, the formation of $\mathrm{NO_x}$ can be attributed to four distinct chemical kinetic processes: thermal $\mathrm{NO_x}$ formation, prompt $\mathrm{NO_x}$ formation, fuel $\mathrm{NO_x}$ formation, and intermediate $\mathrm{N_2O}$. Thermal $\mathrm{NO_x}$ is formed by the oxidation of atmospheric nitrogen present in the combustion air. Prompt $\mathrm{NO_x}$ is produced by high-speed reactions at the flame front, and fuel $\mathrm{NO_x}$ is produced by oxidation of nitrogen contained in the fuel. At elevated pressures and oxygen-rich conditions, $\mathrm{NO_x}$ may also be formed from molecular nitrogen ($\mathrm{N_2}$) via $\mathrm{N_2O}$. The reburning and SNCR mechanisms reduce the total $\mathrm{NO_x}$ formation by accounting for the reaction of NO with hydrocarbons and ammonia, respectively.

i The $\mathrm{NO_x}$ models cannot be used in conjunction with the premixed combustion model.

13.1.2 Governing Equations for NOx Transport

ANSYS FLUENT solves the mass transport equation for the NO species, taking into account convection, diffusion, production and consumption of NO and related species. This approach is completely general, being derived from the fundamental principle of mass conservation. The effect of residence time in $\mathrm{NO_x}$ mechanisms, a Lagrangian reference frame concept, is included through the convection terms in the governing equations written in the Eulerian reference frame. For thermal and prompt $\mathrm{NO_x}$ mechanisms, only the NO species transport equation is needed:

$$\frac{\partial}{\partial t}(\rho Y_{\mathrm{NO}}) + \nabla \cdot (\rho \vec{v} Y_{\mathrm{NO}}) = \nabla \cdot (\rho \mathcal{D} \nabla Y_{\mathrm{NO}}) + S_{\mathrm{NO}} \tag{13.1-1}$$

As discussed in Section 13.1.5: Fuel NOx Formation, the fuel $\mathrm{NO_x}$ mechanisms are more involved. The tracking of nitrogen-containing intermediate species is important. ANSYS FLUENT solves a transport equation for the HCN, $\mathrm{NH_3}$ or $\mathrm{N_2O}$ species in addition to the NO species:

$$\frac{\partial}{\partial t}(\rho Y_{\mathrm{HCN}}) + \nabla \cdot (\rho \vec{v} Y_{\mathrm{HCN}}) = \nabla \cdot (\rho \mathcal{D} Y_{\mathrm{HCN}}) + S_{\mathrm{HCN}} \tag{13.1-2}$$

$$\frac{\partial}{\partial t}(\rho Y_{\mathrm{NH_3}}) + \nabla \cdot (\rho \vec{v} Y_{\mathrm{NH_3}}) = \nabla \cdot (\rho \mathcal{D} Y_{\mathrm{NH_3}}) + S_{\mathrm{NH_3}} \tag{13.1-3}$$

$$\frac{\partial}{\partial t}(\rho Y_{\mathrm{N_2O}}) + \nabla \cdot (\rho \vec{v} Y_{\mathrm{N_2O}}) = \nabla \cdot (\rho \mathcal{D} Y_{\mathrm{N_2O}}) + S_{\mathrm{N_2O}} \tag{13.1-4}$$

where Y_{HCN}, $Y_{\mathrm{NH_3}}$, $Y_{\mathrm{N_2O}}$, and Y_{NO} are mass fractions of HCN, NH_3, N_2O, and NO in the gas phase, and $\mathcal{D}$ is the effective diffusion coefficient. The source terms S_{HCN}, $S_{\mathrm{NH_3}}$, $S_{\mathrm{N_2O}}$, and S_{NO} are to be determined next for different NO_x mechanisms.

13.1.3 Thermal NOx Formation

The formation of thermal NO_x is determined by a set of highly temperature-dependent chemical reactions known as the extended Zeldovich mechanism. The principal reactions governing the formation of thermal NO_x from molecular nitrogen are as follows:

$$\mathrm{O} + \mathrm{N_2} \rightleftharpoons \mathrm{N} + \mathrm{NO} \tag{13.1-5}$$

$$\mathrm{N} + \mathrm{O_2} \rightleftharpoons \mathrm{O} + \mathrm{NO} \tag{13.1-6}$$

A third reaction has been shown to contribute to the formation of thermal NO_x, particularly at near-stoichiometric conditions and in fuel-rich mixtures:

$$\mathrm{N} + \mathrm{OH} \rightleftharpoons \mathrm{H} + \mathrm{NO} \tag{13.1-7}$$

Thermal NOx Reaction Rates

The rate constants for these reactions have been measured in numerous experimental studies [29, 102, 236], and the data obtained from these studies have been critically evaluated by Baulch et al. [18] and Hanson and Salimian [124]. The expressions for the rate coefficients for Equations 13.1-5–13.1-7 used in the NO_x model are given below. These were selected based on the evaluation of Hanson and Salimian [124].

$$\begin{array}{lcl@{\qquad}lcl}
k_{f,1} & = & 1.8\times 10^{8} e^{-38370/T} & k_{r,1} & = & 3.8\times 10^{7} e^{-425/T} \\
k_{f,2} & = & 1.8\times 10^{4} T e^{-4680/T} & k_{r,2} & = & 3.81\times 10^{3} T e^{-20820/T} \\
k_{f,3} & = & 7.1\times 10^{7} e^{-450/T} & k_{r,3} & = & 1.7\times 10^{8} e^{-24560/T}
\end{array}$$

In the above expressions, $k_{f,1}$, $k_{f,2}$, and $k_{f,3}$ are the rate constants for the forward reactions 13.1-5–13.1-7, respectively, and $k_{r,1}$, $k_{r,2}$, and $k_{r,3}$ are the corresponding reverse rate constants. All of these rate constants have units of m^3/gmol-s.

The net rate of formation of NO via Reactions 13.1-5–13.1-7 is given by

$$\frac{d[\mathrm{NO}]}{dt} = k_{f,1}[\mathrm{O}][\mathrm{N_2}] + k_{f,2}[\mathrm{N}][\mathrm{O_2}] + k_{f,3}[\mathrm{N}][\mathrm{OH}] - k_{r,1}[\mathrm{NO}][\mathrm{N}] - k_{r,2}[\mathrm{NO}][\mathrm{O}] - k_{r,3}[\mathrm{NO}][\mathrm{H}] \tag{13.1-8}$$

where all concentrations have units of $gmol/m^3$.

To calculate the formation rates of NO and N, the concentrations of O, H, and OH are required.

The Quasi-Steady Assumption for [N]

The rate of formation of NO_x is significant only at high temperatures (greater than 1800 K) because fixation of nitrogen requires the breaking of the strong N_2 triple bond (dissociation energy of 941 kJ/gmol). This effect is represented by the high activation energy of reaction 13.1-5, which makes it the rate-limiting step of the extended Zeldovich mechanism. However, the activation energy for oxidation of N atoms is small. When there is sufficient oxygen, as in a fuel-lean flame, the rate of consumption of free nitrogen atoms becomes equal to the rate of its formation and therefore a quasi-steady state can be established. This assumption is valid for most combustion cases except in extremely fuel-rich combustion conditions. Hence the NO formation rate becomes

$$\frac{d[\mathrm{NO}]}{dt} = 2k_{f,1}[\mathrm{O}][\mathrm{N_2}]\frac{\left(1-\frac{k_{r,1}k_{r,2}[\mathrm{NO}]^2}{k_{f,1}[\mathrm{N_2}]k_{f,2}[\mathrm{O_2}]}\right)}{\left(1+\frac{k_{r,1}[\mathrm{NO}]}{k_{f,2}[\mathrm{O_2}]+k_{f,3}[\mathrm{OH}]}\right)} \quad (\mathrm{gmol/m^3\text{-}s}) \qquad (13.1\text{-}9)$$

Thermal NOx Temperature Sensitivity

From Equation 13.1-9 it is clear that the rate of formation of NO will increase with increasing oxygen concentration. It also appears that thermal NO formation should be highly dependent on temperature but independent of fuel type. In fact, based on the limiting rate described by $k_{f,1}$, the thermal NO_x production rate doubles for every 90 K temperature increase beyond 2200 K.

Decoupled Thermal NOx Calculations

To solve Equation 13.1-9, the concentration of O atoms and the free radical OH will be required, in addition to the concentration of stable species (i.e., O_2, N_2). Following the suggestion by Zeldovich, the thermal NO_x formation mechanism can be decoupled from the main combustion process, by assuming equilibrium values of temperature, stable species, O atoms, and OH radicals. However, radical concentrations, O atoms in particular, are observed to be more abundant than their equilibrium levels. The effect of partial equilibrium O atoms on NO_x formation rate has been investigated [232] during laminar methane-air combustion. The results of these investigations indicate that the level of NO_x emission can be underpredicted by as much as 28% in the flame zone, when assuming equilibrium O-atom concentrations.

Approaches for Determining O Radical Concentration

There has been little detailed study of radical concentration in industrial turbulent flames, but work [79] has demonstrated the existence of this phenomenon in turbulent diffusion flames. Presently, there is no definitive conclusion as to the effect of partial equilibrium on NO_x formation rates in turbulent flames. Peters and Donnerhack [269] suggest that partial equilibrium radicals can account for no more than a 25% increase in thermal NO_x and that fluid dynamics has the dominant effect on NO_x formation rate. Bilger et al. [25] suggest that in turbulent diffusion flames, the effect of O atom overshoot on the NO_x formation rate is very important.

To overcome this possible inaccuracy, one approach would be to couple the extended Zeldovich mechanism with a detailed hydrocarbon combustion mechanism involving many reactions, species, and steps. This approach has been used previously for research purposes [229]. However, long computer processing time has made the method economically unattractive and its extension to turbulent flows difficult.

To determine the O radical concentration, ANSYS FLUENT uses one of three approaches—the equilibrium approach, the partial equilibrium approach, and the predicted concentration approach—in recognition of the ongoing controversy discussed above.

Method 1: Equilibrium Approach

The kinetics of the thermal NO_x formation rate is much slower than the main hydrocarbon oxidation rate, and so most of the thermal NO_x is formed after completion of combustion. Therefore, the thermal NO_x formation process can often be decoupled from the main combustion reaction mechanism and the NO_x formation rate can be calculated by assuming equilibration of the combustion reactions. Using this approach, the calculation of the thermal NO_x formation rate is considerably simplified. The assumption of equilibrium can be justified by a reduction in the importance of radical overshoots at higher flame temperature [78]. According to Westenberg [377], the equilibrium O-atom concentration can be obtained from the expression

$$[\mathrm{O}] = k_p[\mathrm{O}_2]^{1/2} \qquad (13.1\text{-}10)$$

With k_p included, this expression becomes

$$[\mathrm{O}] = 3.97 \times 10^5 T^{-1/2}[\mathrm{O}_2]^{1/2} e^{-31090/T} \qquad \mathrm{gmol/m^3} \qquad (13.1\text{-}11)$$

where T is in Kelvin.

Method 2: Partial Equilibrium Approach

An improvement to method 1 can be made by accounting for third-body reactions in the O_2 dissociation-recombination process:

$$O_2 + M \rightleftharpoons O + O + M \tag{13.1-12}$$

Equation 13.1-11 is then replaced by the following expression [367]:

$$[O] = 36.64 T^{1/2} [O_2]^{1/2} e^{-27123/T} \qquad \text{gmol/m}^3 \tag{13.1-13}$$

which generally leads to a higher partial O-atom concentration.

Method 3: Predicted O Approach

When the O-atom concentration is well predicted using an advanced chemistry model (such as the flamelet submodel of the non-premixed model), [O] can be taken simply from the local O-species mass fraction.

Approaches for Determining OH Radical Concentration

ANSYS FLUENT uses one of three approaches to determine the OH radical concentration: the exclusion of OH from the thermal NO_x calculation approach, the partial equilibrium approach, and the use of the predicted OH concentration approach.

Method 1: Exclusion of OH Approach

In this approach, the third reaction in the extended Zeldovich mechanism (Equation 13.1-7) is assumed to be negligible through the following observation:

$$k_2 [O_2]_{\rm eq} \gg k_3 [OH]_{\rm eq}$$

This assumption is justified for lean fuel conditions and is a reasonable assumption for most cases.

Method 2: Partial Equilibrium Approach

In this approach, the concentration of OH in the third reaction in the extended Zeldovich mechanism (Equation 13.1-7) is given by [19, 376]

$$[OH] = 2.129 \times 10^2 T^{-0.57} e^{-4595/T} [O]^{1/2} [H_2O]^{1/2} \quad \text{gmol/m}^3 \tag{13.1-14}$$

Method 3: Predicted OH Approach

As in the predicted O approach, when the OH radical concentration is well predicted using an advanced chemistry model such as the flamelet model, [OH] can be taken directly from the local OH species mass fraction.

Summary

To summarize, thermal NO_x formation rate is predicted by Equation 13.1-9. The O-atom concentration needed in Equation 13.1-9 is computed using Equation 13.1-11 for the equilibrium assumption, using Equation 13.1-13 for a partial equilibrium assumption, or using the local O-species mass fraction. You will make the choice during problem setup. In terms of the transport equation for NO (Equation 13.1-1), the NO source term due to thermal NO_x mechanisms is

$$S_{\mathrm{thermal,NO}} = M_{w,\mathrm{NO}} \frac{d[\mathrm{NO}]}{dt} \tag{13.1-15}$$

where $M_{w,\mathrm{NO}}$ is the molecular weight of NO (kg/gmol), and $d[\mathrm{NO}]/dt$ is computed from Equation 13.1-9.

13.1.4 Prompt NOx Formation

It is known that during combustion of hydrocarbon fuels, the NO_x formation rate can exceed that produced from direct oxidation of nitrogen molecules (i.e., thermal NO_x).

Prompt NOx Combustion Environments

The presence of a second mechanism leading to NO_x formation was first identified by Fenimore [91] and was termed “prompt NO_x”. There is good evidence that prompt NO_x can be formed in a significant quantity in some combustion environments, such as in low-temperature, fuel-rich conditions and where residence times are short. Surface burners, staged combustion systems, and gas turbines can create such conditions [13].

At present the prompt NO_x contribution to total NO_x from stationary combustors is small. However, as NO_x emissions are reduced to very low levels by employing new strategies (burner design or furnace geometry modification), the relative importance of the prompt NO_x can be expected to increase.

Prompt NOx Mechanism

Prompt NO_x is most prevalent in rich flames. The actual formation involves a complex series of reactions and many possible intermediate species. The route now accepted is as follows:

$$\begin{aligned} CH + N_2 &\rightleftharpoons HCN + N && (13.1\text{-}16) \\ N + O_2 &\rightleftharpoons NO + O && (13.1\text{-}17) \\ HCN + OH &\rightleftharpoons CN + H_2O && (13.1\text{-}18) \\ CN + O_2 &\rightleftharpoons NO + CO && (13.1\text{-}19) \end{aligned}$$

A number of species resulting from fuel fragmentation have been suggested as the source of prompt NO_x in hydrocarbon flames (e.g., CH, CH_2, C, C_2H), but the major contribution is from CH (Equation 13.1-16) and CH_2, via

$$CH_2 + N_2 \rightleftharpoons HCN + NH \qquad (13.1\text{-}20)$$

The products of these reactions could lead to formation of amines and cyano compounds that subsequently react to form NO by reactions similar to those occurring in oxidation of fuel nitrogen, for example:

$$HCN + N \rightleftharpoons N_2 + ... \qquad (13.1\text{-}21)$$

Prompt NOx Formation Factors

Prompt NO_x formation is proportional to the number of carbon atoms present per unit volume and is independent of the parent hydrocarbon identity. The quantity of HCN formed increases with the concentration of hydrocarbon radicals, which in turn increases with equivalence ratio. As the equivalence ratio increases, prompt NO_x production increases at first, then passes a peak, and finally decreases due to a deficiency in oxygen.

Primary Reaction

Reaction 13.1-16 is of primary importance. In recent studies [304], comparison of probability density distributions for the location of the peak NO_x with those obtained for the peak CH have shown close correspondence, indicating that the majority of the NO_x at the flame base is prompt NO_x formed by the CH reaction. Assuming that Reaction 13.1-16 controls the prompt NO_x formation rate,

$$\frac{d[\mathrm{NO}]}{dt} = k_0[\mathrm{CH}][\mathrm{N_2}] \qquad (13.1\text{-}22)$$

Modeling Strategy

There are, however, uncertainties about the rate data for the above reaction. From Reactions 13.1-16–13.1-20, it can be concluded that the prediction of prompt NO_x formation within the flame requires coupling of the NO_x kinetics to an actual hydrocarbon combustion mechanism. Hydrocarbon combustion mechanisms involve many steps and, as mentioned previously, are extremely complex and costly to compute. In the present NO_x model, a global kinetic parameter derived by De Soete [69] is used. De Soete compared the experimental values of total NO_x formation rate with the rate of formation calculated by numerical integration of the empirical overall reaction rates of NO_x and N_2 formation. He showed that overall prompt formation rate can be predicted from the expression

$$\frac{d[\mathrm{NO}]}{dt} = (\text{overall prompt NO}_\mathrm{x}\text{ formation rate}) - (\text{overall prompt N}_2\text{ formation rate}) \tag{13.1-23}$$

In the early stages of the flame, where prompt NO_x is formed under fuel-rich conditions, the O concentration is high and the N radical almost exclusively forms NO_x rather than nitrogen. Therefore, the prompt NO_x formation rate will be approximately equal to the overall prompt NO_x formation rate:

$$\frac{d[\mathrm{NO}]}{dt} = k_{\mathrm{pr}}[\mathrm{O}_2]^a[\mathrm{N}_2][\mathrm{FUEL}]e^{-E_a/RT} \tag{13.1-24}$$

For C_2H_4 (ethylene)-air flames,

$$k_{\mathrm{pr}} = 1.2\times 10^7 (RT/p)^{a+1}; \quad E_a = 251151 \text{ J/gmol}$$

where a is the oxygen reaction order, R is the universal gas constant, and p is pressure (all in SI units). The rate of prompt NO_x formation is found to be of the first order with respect to nitrogen and fuel concentration, but the oxygen reaction order, a, depends on experimental conditions.

Rate for Most Hydrocarbon Fuels

Equation 13.1-24 was tested against the experimental data obtained by Backmier et al. [9] for different mixture strengths and fuel types. The predicted results indicated that the model performance declined significantly under fuel-rich conditions and for higher hydrocarbon fuels. To reduce this error and predict the prompt $\mathrm{NO_x}$ adequately in all conditions, the De Soete model was modified using the available experimental data. A correction factor, f, was developed, which incorporates the effect of fuel type, i.e., number of carbon atoms, and air-to-fuel ratio for gaseous aliphatic hydrocarbons. Equation 13.1-24 now becomes

$$\frac{d[\mathrm{NO}]}{dt} = f k'_{\mathrm{pr}}[\mathrm{O_2}]^a[\mathrm{N_2}][\mathrm{FUEL}]e^{-E'_a/RT} \qquad (13.1\text{-}25)$$

so that the source term due to prompt $\mathrm{NO_x}$ mechanism is

$$S_{\mathrm{prompt,NO}} = M_{w,\mathrm{NO}}\frac{d[\mathrm{NO}]}{dt} \qquad (13.1\text{-}26)$$

In the above equations,

$$f = 4.75 + 0.0819\ n - 23.2\phi + 32\phi^2 - 12.2\phi^3 \qquad (13.1\text{-}27)$$

$$k'_{\mathrm{pr}} = 6.4 \times 10^6 (RT/p)^{a+1}; \quad E'_a = 303474.125 \text{ J/gmol}$$

n is the number of carbon atoms per molecule for the hydrocarbon fuel, and ϕ is the equivalence ratio. The correction factor is a curve fit for experimental data, valid for aliphatic alkane hydrocarbon fuels ($\mathrm{C}_n\mathrm{H}_{2n+2}$) and for equivalence ratios between 0.6 and 1.6. For values outside the range, the appropriate limit should be used. Values of k'_{pr} and E'_a were developed at the Department of Fuel and Energy at The University of Leeds in England.

Here the concept of equivalence ratio refers to an overall equivalence ratio for the flame, rather than any spatially varying quantity in the flow domain. In complex geometries with multiple burners this may lead to some uncertainty in the specification of ϕ. However, since the contribution of prompt $\mathrm{NO_x}$ to the total $\mathrm{NO_x}$ emission is often very small, results are not likely to be significantly biased.

Oxygen Reaction Order

Oxygen reaction order depends on flame conditions. According to De Soete [69], oxygen reaction order is uniquely related to oxygen mole fraction in the flame:

$$a = \begin{cases} 1.0, & X_{\mathrm{O}_2} \leq 4.1 \times 10^{-3} \\ -3.95 - 0.9 \ln X_{\mathrm{O}_2}, & 4.1 \times 10^{-3} \leq X_{\mathrm{O}_2} \leq 1.11 \times 10^{-2} \\ -0.35 - 0.1 \ln X_{\mathrm{O}_2}, & 1.11 \times 10^{-2} < X_{\mathrm{O}_2} < 0.03 \\ 0, & X_{\mathrm{O}_2} \geq 0.03 \end{cases} \qquad (13.1\text{-}28)$$

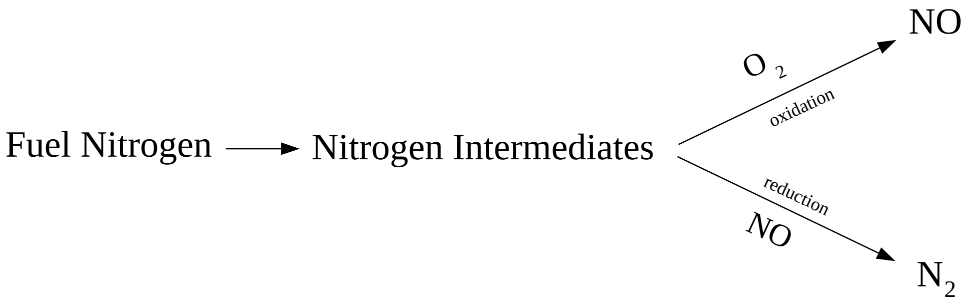
13.1.5 Fuel NOx Formation

Fuel-Bound Nitrogen

It is well known that nitrogen-containing organic compounds present in liquid or solid fossil fuel can contribute to the total $\mathrm{NO_x}$ formed during the combustion process. This fuel nitrogen is a particularly important source of nitrogen oxide emissions for residual fuel oil and coal, which typically contain 0.3–2% nitrogen by weight. Studies have shown that most of the nitrogen in heavy fuel oils is in the form of heterocycles and it is thought that the nitrogen components of coal are similar [154]. It is believed that pyridine, quinoline, and amine type heterocyclic ring structures are of importance.

Reaction Pathways

The extent of conversion of fuel nitrogen to $\mathrm{NO_x}$ is dependent on the local combustion characteristics and the initial concentration of nitrogen-bound compounds. Fuel-bound compounds that contain nitrogen are released into the gas phase when the fuel droplets or particles are heated during the devolatilization stage. From the thermal decomposition of these compounds, (aniline, pyridine, pyrroles, etc.) in the reaction zone, radicals such as HCN, $\mathrm{NH_3}$, N, CN, and NH can be formed and converted to $\mathrm{NO_x}$. The above free radicals (i.e., secondary intermediate nitrogen compounds) are subject to a double competitive reaction path. This chemical mechanism has been subject to several detailed investigations [230]. Although the route leading to fuel $\mathrm{NO_x}$ formation and destruction is still not completely understood, different investigators seem to agree on a simplified model:



Recent investigations [133] have shown that hydrogen cyanide appears to be the principal product if fuel nitrogen is present in aromatic or cyclic form. However, when fuel nitrogen is present in the form of aliphatic amines, ammonia becomes the principal product of fuel nitrogen conversion.

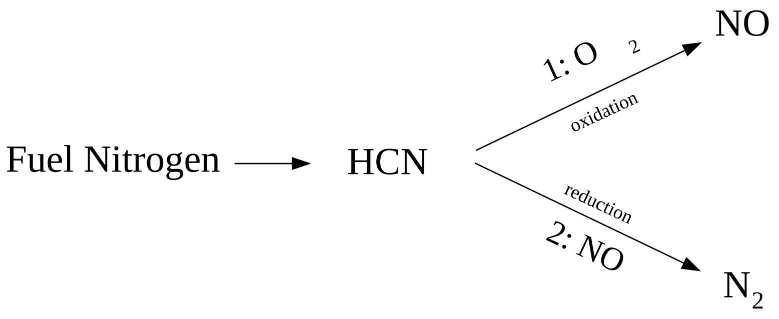
In the ANSYS FLUENT NO_x model, sources of NO_x emission for gaseous, liquid and coal fuels are considered separately. The nitrogen-containing intermediates are grouped as HCN, NH_3, or a combination of both. Transport equations (13.1-1 and 13.1-2 or 13.1-3) are solved, after which the source terms $S_{\rm HCN}$, $S_{\rm NH_3}$, and $S_{\rm NO}$ are determined for different fuel types. Discussions to follow refer to fuel NO_x sources for $S_{\rm NO}$ and intermediate HCN, NH_3 sources for $S_{\rm HCN}$ and $S_{\rm NH_3}$. Contributions from thermal and prompt mechanisms have been discussed in previous sections.

Fuel NOx from Gaseous and Liquid Fuels

The fuel NO_x mechanisms for gaseous and liquid fuels are based on different physics but the same chemical reaction pathways.

Fuel NOx from Intermediate Hydrogen Cyanide (HCN)

When HCN is used as the intermediate species:



The source terms in the transport equations can be written as follows:

$$S_{\rm HCN} = S_{\rm pl,HCN} + S_{\rm HCN-1} + S_{\rm HCN-2} \tag{13.1-29}$$

$$S_{\rm NO} = S_{\rm NO-1} + S_{\rm NO-2} \tag{13.1-30}$$

HCN Production in a Gaseous Fuel

The rate of HCN production is equivalent to the rate of combustion of the fuel:

$$S_{\rm pl,HCN} = \frac{\mathcal{R}_{\rm cf}\; Y_{\rm N,fuel}\; M_{w,\rm HCN}}{M_{w,\rm N}} \tag{13.1-31}$$

where $S_{\rm pl,HCN}$ = source of HCN (kg/m^3-s)
$\mathcal{R}_{\rm cf}$ = mean limiting reaction rate of fuel (kg/m^3-s)
$Y_{\rm N,fuel}$ = mass fraction of nitrogen in the fuel

The mean limiting reaction rate of fuel, $\mathcal{R}_{\rm cf}$, is calculated from the Magnussen combustion model, so the gaseous fuel NO$_x$ option is available only when the generalized finite-rate model is used.

HCN Production in a Liquid Fuel

The rate of HCN production is equivalent to the rate of fuel release into the gas phase through droplet evaporation:

$$S_{\rm pl,HCN} = \frac{S_{\rm fuel}\; Y_{\rm N,fuel}\; M_{w,\rm HCN}}{M_{w,\rm N} V} \tag{13.1-32}$$

where $S_{\rm pl,HCN}$ = source of HCN (kg/m^3-s)
$S_{\rm fuel}$ = rate of fuel release from the liquid droplets to the gas (kg/s)
$Y_{\rm N,fuel}$ = mass fraction of nitrogen in the fuel
V = cell volume (m^3)

HCN Consumption

The HCN depletion rates from reactions (1) and (2) in the above mechanism are the same for both gaseous and liquid fuels, and are given by De Soete [69] as

$$\mathcal{R}_1 = A_1\; X_{\rm HCN}\; X_{\rm O_2}^a\; e^{-E_1/RT} \tag{13.1-33}$$

$$\mathcal{R}_2 = A_2\; X_{\rm HCN}\; X_{\rm NO}\; e^{-E_2/RT} \tag{13.1-34}$$

where $\mathcal{R}_1$, $\mathcal{R}_2$ = conversion rates of HCN (s^{-1})
T = instantaneous temperature (K)
X = mole fractions
A_1 = 1.0 $\times 10^{10}$ s^{-1}
A_2 = 3.0 $\times 10^{12}$ s^{-1}
E_1 = 280451.95 J/gmol
E_2 = 251151 J/gmol

The oxygen reaction order, a, is calculated from Equation 13.1-28.

Since mole fraction is related to mass fraction through molecular weights of the species ($M_{w,i}$) and the mixture ($M_{w,m}$),

$$X_i = Y_i \frac{M_{w,m}}{M_{w,i}} = \frac{Y_i}{M_{w,i}} \left(\frac{\rho R T}{p} \right) \tag{13.1-35}$$

HCN Sources in the Transport Equation

The mass consumption rates of HCN which appear in Equation 13.1-29 are calculated as

$$S_{\mathrm{HCN}-1} = -\mathcal{R}_1 \frac{M_{w,\mathrm{HCN}}\, p}{R\overline{T}} \tag{13.1-36}$$

$$S_{\mathrm{HCN}-2} = -\mathcal{R}_2 \frac{M_{w,\mathrm{HCN}}\, p}{R\overline{T}} \tag{13.1-37}$$

where $S_{\mathrm{HCN}-1}$ = consumption rates of HCN in
$S_{\mathrm{HCN}-2}$ reactions 1 and 2 respectively (kg/m^3-s)
p = pressure (Pa)
$\overline{T}$ = mean temperature (K)
R = universal gas constant

NOx Sources in the Transport Equation

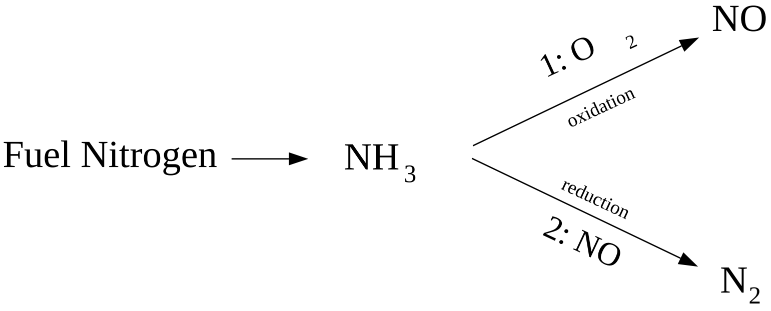
NO$_\mathrm{x}$ is produced in reaction 1 but destroyed in reaction 2. The sources for Equation 13.1-30 are the same for a gaseous as for a liquid fuel, and are evaluated as follows:

$$S_{\mathrm{NO}-1} = -S_{\mathrm{HCN}-1} \frac{M_{w,\mathrm{NO}}}{M_{w,\mathrm{HCN}}} = \mathcal{R}_1 \frac{M_{w,\mathrm{NO}}\, p}{R\overline{T}} \tag{13.1-38}$$

$$S_{\mathrm{NO}-2} = S_{\mathrm{HCN}-2} \frac{M_{w,\mathrm{NO}}}{M_{w,\mathrm{HCN}}} = -\mathcal{R}_2 \frac{M_{w,\mathrm{NO}}\, p}{R\overline{T}} \tag{13.1-39}$$

Fuel NOx from Intermediate Ammonia (NH_3)

When NH_3 is used as the intermediate species:



The source terms in the transport equations can be written as follows:

$$S_{\mathrm{NH_3}} = S_{\mathrm{pl,NH_3}} + S_{\mathrm{NH_3-1}} + S_{\mathrm{NH_3-2}} \tag{13.1-40}$$

$$S_{\mathrm{NO}} = S_{\mathrm{NO-1}} + S_{\mathrm{NO-2}} \tag{13.1-41}$$

NH_3 Production in a Gaseous Fuel

The rate of NH_3 production is equivalent to the rate of combustion of the fuel:

$$S_{\mathrm{pl,NH_3}} = \frac{\mathcal{R}_{\mathrm{cf}}\, Y_{\mathrm{N,fuel}}\, M_{w,\mathrm{NH_3}}}{M_{w,\mathrm{N}}} \tag{13.1-42}$$

where $S_{\mathrm{pl,NH_3}}$ = source of NH_3 (kg/m^3-s)
$\mathcal{R}_{\mathrm{cf}}$ = mean limiting reaction rate of fuel (kg/m^3-s)
$Y_{\mathrm{N,fuel}}$ = mass fraction of nitrogen in the fuel

The mean limiting reaction rate of fuel, $\mathcal{R}_{\mathrm{cf}}$, is calculated from the Magnussen combustion model, so the gaseous fuel NO_x option is available only when the generalized finite-rate model is used.

NH_3 Production in a Liquid Fuel

The rate of NH_3 production is equivalent to the rate of fuel release into the gas phase through droplet evaporation:

$$S_{\mathrm{pl,NH_3}} = \frac{S_{\mathrm{fuel}}\, Y_{\mathrm{N,fuel}}\, M_{w,\mathrm{NH_3}}}{M_{w,\mathrm{N}} V} \tag{13.1-43}$$

where $S_{\mathrm{pl,NH_3}}$ = source of NH_3 (kg/m^3-s)
S_{fuel} = rate of fuel release from the liquid droplets to the gas (kg/s)
$Y_{\mathrm{N,fuel}}$ = mass fraction of nitrogen in the fuel
V = cell volume (m^3)

NH_3 Consumption

The NH_3 depletion rates from reactions (1) and (2) in the above mechanism are the same for both gaseous and liquid fuels, and are given by De Soete [69] as

$$\mathcal{R}_1 = A_1\, X_{\mathrm{NH_3}}\, X_{\mathrm{O_2}}^a\, e^{-E_1/RT} \tag{13.1-44}$$

$$\mathcal{R}_2 = A_2\, X_{\mathrm{NH_3}}\, X_{\mathrm{NO}}\, e^{-E_2/RT} \tag{13.1-45}$$

where $\mathcal{R}_1$, $\mathcal{R}_2$ = conversion rates of NH_3 (s^{-1})
T = instantaneous temperature (K)
X = mole fractions
A_1 = 4.0×10^6 s^{-1}
A_2 = 1.8×10^8 s^{-1}
E_1 = 133947.2 J/gmol
E_2 = 113017.95 J/gmol

The oxygen reaction order, a, is calculated from Equation 13.1-28.

Since mole fraction is related to mass fraction through molecular weights of the species ($M_{w,i}$) and the mixture ($M_{w,m}$), X_i can be calculated using Equation 13.1-35.

NH_3 Sources in the Transport Equation

The mass consumption rates of NH_3 which appear in Equation 13.1-40 are calculated as

$$S_{\mathrm{NH_3}-1} = -\mathcal{R}_1 \frac{M_{w,\mathrm{NH_3}}\, p}{R\overline{T}} \tag{13.1-46}$$

$$S_{\mathrm{NH_3}-2} = -\mathcal{R}_2 \frac{M_{w,\mathrm{NH_3}}\, p}{R\overline{T}} \tag{13.1-47}$$

where $S_{\mathrm{NH_3}-1}$ = consumption rates of NH_3 in
$S_{\mathrm{NH_3}-2}$ reactions 1 and 2 respectively (kg/m^3-s)
p = pressure (Pa)
$\overline{T}$ = mean temperature (K)
R = universal gas constant

NOx Sources in the Transport Equation

NO_x is produced in reaction 1 but destroyed in reaction 2. The sources for Equation 13.1-41 are the same for a gaseous as for a liquid fuel, and are evaluated as follows:

$$S_{\mathrm{NO}-1} = -S_{\mathrm{NH_3}-1} \frac{M_{w,\mathrm{NO}}}{M_{w,\mathrm{NH_3}}} = \mathcal{R}_1 \frac{M_{w,\mathrm{NO}}\, p}{R\overline{T}} \tag{13.1-48}$$

$$S_{\mathrm{NO}-2} = S_{\mathrm{NH_3}-2} \frac{M_{w,\mathrm{NO}}}{M_{w,\mathrm{NH_3}}} = -\mathcal{R}_2 \frac{M_{w,\mathrm{NO}}\, p}{R\overline{T}} \tag{13.1-49}$$

Fuel NOx from Coal

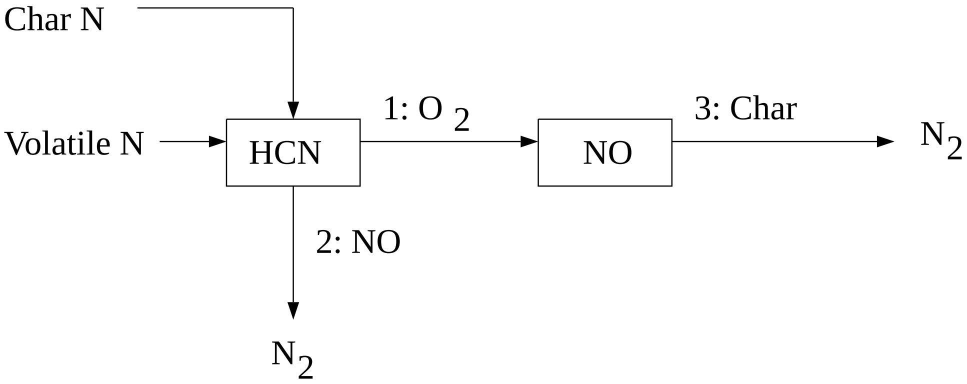
Nitrogen in Char and in Volatiles

For the coal it is assumed that fuel nitrogen is distributed between the volatiles and the char. Since there is no reason to assume that N is equally distributed between the volatiles and the char the fraction of N in the volatiles and the char should be specified separately.

When HCN is used as the intermediate species, two variations of fuel NO_x mechanisms for coal are included. When NH_3 is used as the intermediate species, two variations of fuel NO_x mechanisms for coal are included, much like in the calculation of NO_x production from the coal via HCN. It is assumed that fuel nitrogen is distributed between the volatiles and the char.

Coal Fuel NOx Scheme A

The first HCN mechanism assumes that all char N converts to HCN which is then converted partially to NO [327]. The reaction pathway is described as follows:



With the first scheme, all char-bound nitrogen converts to HCN. Thus,

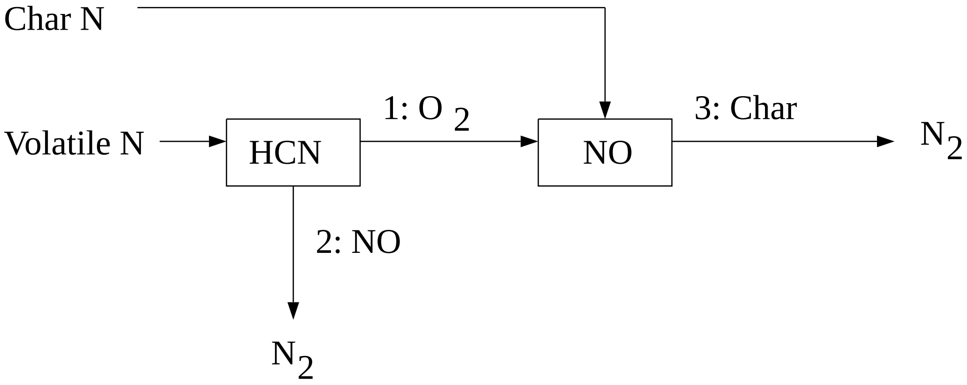
$$S_{\rm char,HCN} = \frac{S_c Y_{\rm N,char} M_{w,\rm HCN}}{M_{w,\rm N} V} \qquad (13.1\text{-}50)$$

$$S_{\rm char,NO} = 0 \qquad (13.1\text{-}51)$$

where S_c = char burnout rate (kg/s)
$Y_{\rm N,char}$ = mass fraction of nitrogen in char
V = cell volume (m^3)

Coal Fuel NOx Scheme B

The second HCN mechanism assumes that all char N converts to NO directly [207]. The reaction pathway is described as follows:



According to Lockwood [207], the char nitrogen is released to the gas phase as NO directly, mainly as a desorption product from oxidized char nitrogen atoms. If this approach is followed, then

$$
\begin{aligned}
S_{\rm char,HCN} &= 0 && (13.1\text{-}52)\\
S_{\rm char,NO} &= \frac{S_c Y_{\rm N,char} M_{w,\rm NO}}{M_{w,\rm N} V} && (13.1\text{-}53)
\end{aligned}
$$

HCN Scheme Selection

The second HCN mechanism tends to produce more $\mathrm{NO_x}$ emission than the first. In general, however, it is difficult to say which one outperforms the other.

The source terms for the transport equations are

$$
\begin{aligned}
S_{\rm HCN} &= S_{\rm pvc,HCN} + S_{\rm HCN-1} + S_{\rm HCN-2} && (13.1\text{-}54)\\
S_{\rm NO} &= S_{\rm char,NO} + S_{\rm NO-1} + S_{\rm NO-2} + S_{\rm NO-3} && (13.1\text{-}55)
\end{aligned}
$$

Source contributions $S_{\rm HCN-1}$, $S_{\rm HCN-2}$, $S_{\rm NO-1}$, and $S_{\rm NO-2}$ are described previously. Therefore, only the heterogeneous reaction source, $S_{\rm NO-3}$, the char $\mathrm{NO_x}$ source, $S_{\rm char,NO}$, and the HCN production source, $S_{\rm pvc,HCN}$, need to be considered.

NOx Reduction on Char Surface

The heterogeneous reaction of NO reduction on the char surface has been modeled according to the following [190]:

$$
\mathcal{R}_3 = A_3 e^{-E_3/R\overline{T}}\, \overline{p}_{\rm NO} \qquad (13.1\text{-}56)
$$

where $\mathcal{R}_3$ = rate of NO reduction ($\mathrm{gmol/m^2_{BET}}$-s)
$\overline{p}_{\rm NO}$ = mean NO partial pressure (atm)
E_3 = 142737.485 J/gmol
A_3 = 230 $\mathrm{gmol/m^2_{BET}}$-s-atm
$\overline{T}$ = mean temperature (K)

The partial pressure $\overline{p}_{\rm NO}$ is calculated using Dalton's law:

$$
\overline{p}_{\rm NO} = \overline{p} X_{\rm NO}
$$

The rate of NO consumption due to reaction 3 will then be

$$S_{\mathrm{NO}-3} = c_s A_{\mathrm{BET}} M_{w,\mathrm{NO}} \mathcal{R}_3$$

where A_{BET} = BET surface area (m^2/kg)
c_s = concentration of particles (kg/m^3)
$S_{\mathrm{NO}-3}$ = NO consumption (kg/m^3-s)

BET Surface Area

The heterogeneous reaction involving char is mainly an adsorption process whose rate is directly proportional to the pore surface area. The pore surface area is also known as the BET surface area due to the researchers who pioneered the adsorption theory (Brunauer, Emmett and Teller [42]). For commercial adsorbents, the pore (BET) surface areas range from 100,000 to 2 million square meters per kilogram, depending on the microscopic structure. For coal, the BET area is typically 25,000 m^2/kg which is used as the default in ANSYS FLUENT. The overall source of HCN ($S_{\mathrm{pvc,HCN}}$) is a combination of volatile contribution ($S_{\mathrm{vol,HCN}}$) and char contribution ($S_{\mathrm{char,HCN}}$):

$$S_{\mathrm{pvc,HCN}} = S_{\mathrm{vol,HCN}} + S_{\mathrm{char,HCN}}$$

HCN from Volatiles

The source of HCN from the volatiles is related to the rate of volatile release:

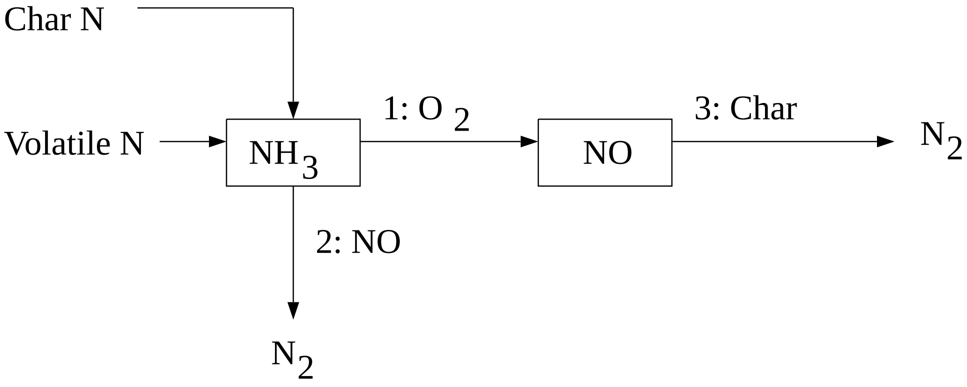
$$S_{\mathrm{vol,HCN}} = \frac{S_{\mathrm{vol}}\, Y_{\mathrm{N,vol}} M_{w,\mathrm{HCN}}}{M_{w,\mathrm{N}} V}$$

where S_{vol} = source of volatiles originating from
the coal particles into the gas phase (kg/s)
$Y_{\mathrm{N,vol}}$ = mass fraction of nitrogen in the volatiles
V = cell volume (m^3)

Calculation of sources related to char-bound nitrogen depends on the fuel NO$_{\mathrm{x}}$ scheme selection.

Coal Fuel NOx Scheme C

The first NH_3 mechanism assumes that all char N converts to NH_3 which is then converted partially to NO [327]. The reaction pathway is described as follows:



In this scheme, all char-bound nitrogen converts to NH_3. Thus,

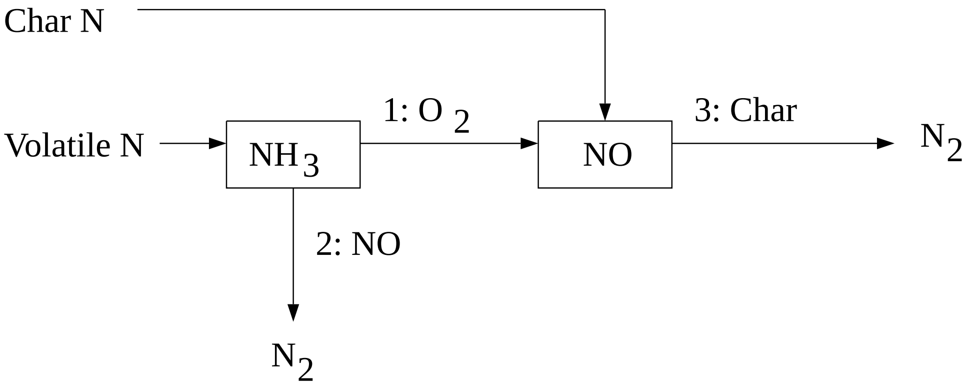
$$S_{\text{char,NH}_3} = \frac{S_c Y_{\text{N,char}} M_{w,\text{NH}_3}}{M_{w,\text{N}} V} \tag{13.1-57}$$

$$S_{\text{char,NO}} = 0 \tag{13.1-58}$$

where S_c = char burnout rate (kg/s)
$Y_{\text{N,char}}$ = mass fraction of nitrogen in char
V = cell volume (m^3)

Coal Fuel NOx Scheme D

The second NH_3 mechanism assumes that all char N converts to NO directly [207]. The reaction pathway is described as follows:



According to Lockwood [207], the char nitrogen is released to the gas phase as NO directly, mainly as a desorption product from oxidized char nitrogen atoms. If this approach is followed, then

$$S_{\text{char},\text{NH}_3} = 0 \qquad (13.1\text{-}59)$$

$$S_{\text{char},\text{NO}} = \frac{S_c Y_{\text{N},\text{char}} M_{w,\text{NO}}}{M_{w,\text{N}} V} \qquad (13.1\text{-}60)$$

NH_3 Scheme Selection

The second NH_3 mechanism tends to produce more NO_x emission than the first. In general, however, it is difficult to say which one outperforms the other.

The source terms for the transport equations are

$$S_{\text{NH}_3} = S_{\text{pvc},\text{NH}_3} + S_{\text{NH}_3-1} + S_{\text{NH}_3-2} \qquad (13.1\text{-}61)$$

$$S_{\text{NO}} = S_{\text{char},\text{NO}} + S_{\text{NO}-1} + S_{\text{NO}-2} + S_{\text{NO}-3} \qquad (13.1\text{-}62)$$

Source contributions S_{NH_3-1}, S_{NH_3-2}, $S_{\text{NO}-1}$, $S_{\text{NO}-2}$, $S_{\text{NO}-3}$, $S_{\text{char},\text{NO}}$ are described previously. Therefore, only the NH_3 production source, $S_{\text{pvc},\text{NH}_3}$, needs to be considered.

The overall production source of NH_3 is a combination of volatile contribution ($S_{\text{vol},\text{NH}_3}$), and char contribution ($S_{\text{char},\text{NH}_3}$):

$$S_{\text{pvc},\text{NH}_3} = S_{\text{vol},\text{NH}_3} + S_{\text{char},\text{NH}_3} \qquad (13.1\text{-}63)$$

NH_3 from Volatiles

The source of NH_3 from the volatiles is related to the rate of volatile release:

$$S_{\text{vol},\text{NH}_3} = \frac{S_{\text{vol}}\, Y_{\text{N},\text{vol}} M_{w,\text{NH}_3}}{M_{w,\text{N}} V}$$

where S_{vol} = source of volatiles originating from the coal particles into the gas phase (kg/s)
$Y_{\text{N},\text{vol}}$ = mass fraction of nitrogen in the volatiles
V = cell volume (m^3)

Calculation of sources related to char-bound nitrogen depends on the fuel NO_x scheme selection.

Fuel Nitrogen Partitioning for HCN and NH_3 Intermediates

In certain cases, especially when the fuel is a solid, both HCN and NH_3 can be generated as intermediates at high enough temperatures [247]. In particular, low-ranking (lignite) coal has been shown to produce 10 times more NH_3 compared to the level of HCN, whereas higher-ranking (bituminous) coal has been shown to produce only HCN [246]. Studies by Winter et al. [381] have shown that for bituminous coal, using an HCN/NH_3 partition ratio of 9:1 gave better NO_x predictions when compared to measurements than specifying only a single intermediate species. Liu and Gibbs [206] work with woody-biomass (pine wood chips), on the other hand, has suggested an HCN/NH_3 ratio of 1:9 due to the younger age of the fuel.

In total, the above work suggests the importance of being able to specify that portions of the fuel nitrogen will be converted to both HCN and NH_3 intermediates at the same time. In ANSYS FLUENT, fuel nitrogen partitioning can be used whenever HCN or NH_3 are intermediates for NO_x production, though it is mainly applicable to solid fuels such as coal and biomass. The reaction pathways and source terms for HCN and NH_3 were discussed in previous sections.

13.1.6 NOx Formation from Intermediate N_2O

Melte and Pratt [222] proposed the first intermediate mechanism for NO_x formation from molecular nitrogen (N_2) via nitrous oxide (N_2O). Nitrogen enters combustion systems mainly as a component of the combustion and dilution air. Under favorable conditions, which are elevated pressures and oxygen-rich conditions, this intermediate mechanism can contribute as much as 90% of the NO_x formed during combustion. This makes it particularly important in equipment such as gas turbines and compression-ignition engines. Because these devices are operated at increasingly low temperatures to prevent NO_x formation via the thermal NO_x mechanism, the relative importance of the N_2O-intermediate mechanism is increasing. It has been observed that about 30% of the NO_x formed in these systems can be attributed to the N_2O-intermediate mechanism.

The N_2O-intermediate mechanism may also be of importance in systems operated in flameless mode (e.g., diluted combustion, flameless combustion, flameless oxidation, and FLOX systems). In a flameless mode, fuel and oxygen are highly diluted in inert gases so that the combustion reactions and resulting heat release are carried out in the diffuse zone. As a consequence, elevated peaks of temperature are avoided, which prevents thermal NO_x. Research suggests that the N_2O-intermediate mechanism may contribute about 90% of the NO_x formed in flameless mode, and that the remainder can be attributed to the prompt NO_x mechanism. The relevance of NO_x formation from N_2O has been observed indirectly, and theoretically speculated for a number of combustion systems and by a number of researchers [12, 61, 112, 338, 344].

N_2O-Intermediate NOx Mechanism

The simplest form of the mechanism [222] takes into account two reversible elementary reactions:

$$\mathrm{N_2 + O + M} \rightleftharpoons \mathrm{N_2O + M} \tag{13.1-64}$$

$$\mathrm{N_2O + O} \rightleftharpoons \mathrm{2NO} \tag{13.1-65}$$

Here, M is a general third body. Because the first reaction involves third bodies, the mechanism is favored at elevated pressures. Both reactions involve the oxygen radical O, which makes the mechanism favored at oxygen-rich conditions. While not always justified, it is often assumed that the radical O atoms originate solely from the dissociation of molecular oxygen,

$$\frac{1}{2}\mathrm{O_2} \rightleftharpoons \mathrm{O} \tag{13.1-66}$$

According to the kinetic rate laws, the rate of $\mathrm{NO_x}$ formation via the $\mathrm{N_2O}$-intermediate mechanism is

$$\frac{d[\mathrm{NO}]}{dt} = 2\left(k_{f,2}[\mathrm{N_2O}][\mathrm{O}] - k_{r,2}[\mathrm{NO}]^2\right) \qquad \mathrm{gmol/m^3\text{-}s} \tag{13.1-67}$$

To solve Equation 13.1-67, you will need to have first calculated [O] and $[\mathrm{N_2O}]$.

It is often assumed that $\mathrm{N_2O}$ is at quasi-steady-state (i.e., $d[\mathrm{N_2O}]/dt = 0$), which implies

$$[\mathrm{N_2O}] = \frac{k_{f,1}[\mathrm{N_2}][\mathrm{O}][\mathrm{M}] + k_{r,2}[\mathrm{NO}]^2}{k_{r,1}[\mathrm{M}] + k_{f,2}[\mathrm{O}]} \tag{13.1-68}$$

The system of Equations 13.1-67–13.1-68 can be solved for the rate of $\mathrm{NO_x}$ formation when the concentration of $\mathrm{N_2}$, $\mathrm{O_2}$, and M, the kinetic rate constants for Equations 13.1-64 and 13.1-65, and the equilibrium constant of Equation 13.1-66 are known. The appearance of NO in Equation 13.1-65 entails that coupling of the $\mathrm{N_2O}$ mechanism with the thermal $\mathrm{NO_x}$ mechanism (and other $\mathrm{NO_x}$ mechanisms).

$$\begin{aligned} k_{f,1} &= 4.44 \times 10^{32} T^{-8.358} e^{-28234/T} & k_{r,1} &= 4.00 \times 10^{8} e^{-28234/T} \\ k_{f,2} &= 2.90 \times 10^{7} e^{-11651/T} & k_{r,2} &= 1.45 \times 10^{-29} T^{9.259} e^{-11651/T} \end{aligned}$$

In the above expressions, $k_{f,1}$ and $k_{f,2}$ are the forward rate constants of Equations 13.1-64 and 13.1-65, and $k_{r,1}$ and $k_{r,2}$ are the corresponding reverse rate constants. The units for $k_{f,2}$, $k_{r,1}$, and $k_{r,2}$ are $\mathrm{m^3/gmol\text{-}s}$, while $k_{f,1}$ has units of $\mathrm{m^6/gmol^2\text{-}s}$.

13.1.7 NOx Reduction by Reburning

The design of complex combustion systems for utility boilers, based on air- and fuel-staging technologies, involves many parameters and their mutual interdependence. These parameters include local stoichiometry, temperature and chemical concentration field, residence time distribution, velocity field, and mixing pattern. A successful application of the in-furnace reduction techniques requires control of these parameters in an optimum manner so as to avoid impairing the boiler performance. In the mid 1990s, global models describing the kinetics of NO_x destruction in the reburn zone of a staged combustion system became available. Two of these models are described below.

Instantaneous Approach

The instantaneous NO_x reburning mechanism is a pathway whereby NO reacts with hydrocarbons and is subsequently reduced. In general:

$$\mathrm{CH}_i + \mathrm{NO} \rightarrow \mathrm{HCN} + \mathrm{products} \tag{13.1-69}$$

Three reburn reactions are modeled by ANSYS FLUENT for $1600 \leq T \leq 2100$:

$$\mathrm{CH} + \mathrm{NO} \xrightarrow{k_1} \mathrm{HCN} + \mathrm{O} \tag{13.1-70}$$

$$\mathrm{CH_2} + \mathrm{NO} \xrightarrow{k_2} \mathrm{HCN} + \mathrm{OH} \tag{13.1-71}$$

$$\mathrm{CH_3} + \mathrm{NO} \xrightarrow{k_3} \mathrm{HCN} + \mathrm{H_2O} \tag{13.1-72}$$

i If the temperature is outside of this range, NO reburn will not be computed.

The rate constants for these reactions are taken from Bowman [32] and have units of m^3/gmol-s:

$$k_1 = 1 \times 10^8; \quad k_2 = 1.4 \times 10^6 e^{-550/T}; \quad k_3 = 2 \times 10^5$$

The NO depletion rate due to reburn is expressed as

$$\frac{d[\mathrm{NO}]}{dt} = -k_1[\mathrm{CH}][\mathrm{NO}] - k_2[\mathrm{CH_2}][\mathrm{NO}] - k_3[\mathrm{CH_3}][\mathrm{NO}] \tag{13.1-73}$$

and the source term for the reburning mechanism in the NO transport equation can be calculated as

$$S_{\text{reburn,NO}} = -M_{w,\text{NO}} \frac{d[\text{NO}]}{dt} \tag{13.1-74}$$

To calculate the NO depletion rate due to reburning, ANSYS FLUENT will obtain the concentrations of CH, CH_2, and CH_3 from the species mass fraction results of the combustion calculation. When you use this method, you must be sure to include the species CH, CH_2, and CH_3 in your problem definition.

Partial Equilibrium Approach

The partial equilibrium approach is based on the model proposed by Kandamby et al. [156] and [7]. The model adds a reduction path to De Soete's global model [69] that describes the NO_x formation/destruction mechanism in a pulverized coal flame. The additional reduction path accounts for the NO_x destruction in the fuel-rich reburn zone by CH radicals (see Figure 13.1.1).

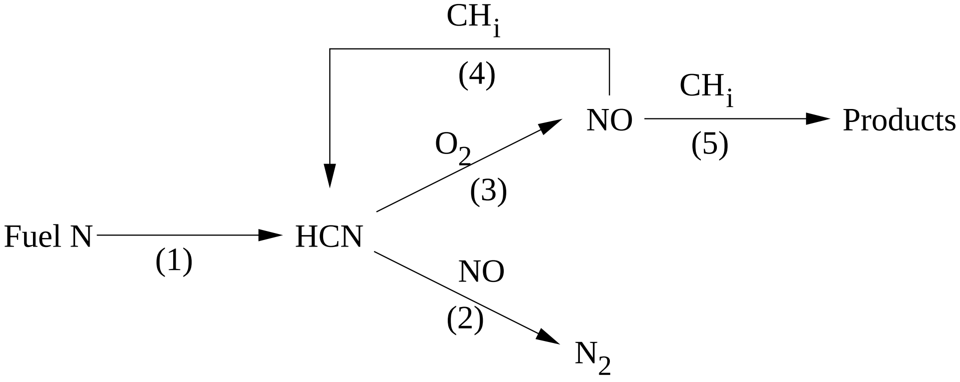


Figure 13.1.1: De Soete's Global NO_x Mechanism with Additional Reduction Path

This model can be used in conjunction with the eddy-dissipation combustion model and does not require the specification of CH radical concentrations, since they are computed based on the CH-radical partial equilibrium. The reburn fuel itself can be an equivalent of CH_4, CH_3, CH_2, or CH. How this equivalent fuel is determined is open for debate and an approximate guide would be to consider the C/H ratio of the fuel itself. A multiplicative constant of 4.0×10^{-4} has been developed for the partial equilibrium of CH radicals to reduce the rates of HCN and NO in the reburn model. This value was obtained by researchers, who developed the model, by way of predicting NO_x values for a number of test cases for which experimental data exists.

NOx Reduction Mechanism

In the fuel-rich reburn zone, the HCN oxidation is suppressed and the amount of NO formed in the primary combustion zone is decreased by the reduction reaction from HCN to N_2. However, the NO concentration may also decrease due to reactions with CH radicals, which are available in significant amounts in the reburn zone. The following are considered to be the most important reactions of NO reduction by CH radicals:

$$\begin{aligned} \mathrm{NO} + \mathrm{CH_2} &\longrightarrow \mathrm{HCN} + \mathrm{OH} && (13.1\text{-}75)\\ \mathrm{NO} + \mathrm{CH} &\longrightarrow \mathrm{HCN} + \mathrm{O} && (13.1\text{-}76)\\ \mathrm{NO} + \mathrm{C} &\longrightarrow \mathrm{CN} + \mathrm{O} && (13.1\text{-}77) \end{aligned}$$

These reactions may be globally described by the addition of pathways (4) and (5) in Figure 13.1.1, leading respectively to the formation of HCN and of minor intermediate nitrogen radicals. Assuming that methane is the reburning gas, the global NO reduction rates are then expressed as

$$\begin{aligned} \mathcal{R}_4 &= (k_a\chi_1 + k_b\chi_1^2)[\mathrm{CH_4}][\mathrm{NO}] && (13.1\text{-}78)\\ \mathcal{R}_5 &= k_c\chi_1^3\chi_2[\mathrm{CH_4}][\mathrm{NO}] && (13.1\text{-}79) \end{aligned}$$

where

$$\chi_1 = \frac{[\mathrm{H}]}{[\mathrm{H_2}]}; \quad \chi_2 = \frac{[\mathrm{OH}]}{[\mathrm{H_2O}]}$$

Therefore, the additional source terms of the HCN and NO transport equations due to reburn reactions are given by

$$\begin{aligned} \frac{d[\mathrm{HCN}]}{dt} &= 4 \times 10^{-4}\mathcal{R}_4 && (13.1\text{-}80)\\ \frac{d[\mathrm{NO}]}{dt} &= -4 \times 10^{-4}(\mathcal{R}_4 + \mathcal{R}_5) && (13.1\text{-}81) \end{aligned}$$

Certain assumptions are required to evaluate the rate constants k_a, k_b, and k_c and the factors χ_1 and χ_2. For hydrocarbon diffusion flames, the following reaction set can be reasonably considered to be in partial equilibrium:

$$\begin{aligned} CH_4 + H &\rightleftharpoons CH_3 + H_2 && (13.1\text{-}82) \\ CH_3 + OH &\rightleftharpoons CH_2 + H_2O && (13.1\text{-}83) \\ CH_2 + H &\rightleftharpoons CH + H_2 && (13.1\text{-}84) \\ CH + H &\rightleftharpoons C + H_2 && (13.1\text{-}85) \end{aligned}$$

Thus, the rate constants may be computed as

$$k_a = k_1 \frac{k_{f,4} k_{f,5}}{k_{r,4} k_{r,5}}; \quad k_b = k_2 \frac{k_{f,4} k_{f,5} k_{f,6}}{k_{r,4} k_{r,5} k_{r,6}}; \quad k_c = k_3 \frac{k_{f,4} k_{f,5} k_{f,6} k_{f,7}}{k_{r,4} k_{r,5} k_{r,6} k_{r,7}}$$

where k_1, k_2, and k_3 are the rate constants for Equations 13.1-75–13.1-77. The forward and reverse rate constants for Equations 13.1-82–13.1-85 are $k_{f,4}$–$k_{f,7}$ and $k_{r,4}$–$k_{r,7}$, respectively. In addition, it is assumed that $\chi_1 = 1$, because the H-radical concentration in the post-flame region of a hydrocarbon diffusion flame has been observed to be of the same order as [H_2]. Finally, the OH-radical concentration is estimated by considering the reaction

$$OH + H_2 \rightleftharpoons H_2O + H \qquad (13.1\text{-}86)$$

to be partially equilibrated, leading to the relationship

$$\chi_2 = \frac{k_{r,8}}{k_{f,8}}$$

Values for the rate constants k_a, k_b, and k_c for different equivalent fuel types are given in Arrhenius form ($AT^b e^{-E/RT}$) in Table 13.1.1 [189]. All rate constants have units of m^3/gmol-s, and all values of E have units of J/gmol.

Table 13.1.1: Rate Constants for Different Reburn Fuels

Equivalent	$\boldsymbol{k_a}$			$\boldsymbol{k_b}$			$\boldsymbol{k_c}$		
Fuel Type	A	b	E	A	b	E	A	b	E
CH_4	5.30×10^9	-1.54	27977	3.31×10^{13}	-3.33	15090	3.06×10^{11}	-2.64	77077
CH_3	0.37×10^9	-1.54	27977	0.23×10^{13}	-3.33	15090	0.21×10^{11}	-2.64	77077
CH_2	0.23×10^7	-1.54	27977	0.14×10^{11}	-3.33	15090	0.13×10^9	-2.64	77077
CH	0.0	0.0	0.0	0.63×10^8	-3.33	15090	0.58×10^6	-2.64	77077

For Equation 13.1-86,

$$k_{f,8} = 1.02 \times 10^5 T^{1.60} e^{-13802/RT}; \quad k_{r,8} = 4.52 \times 10^5 T^{1.60} e^{-80815/RT}$$

13.1.8 NOx Reduction by SNCR

The selective noncatalytic reduction of NO_x (SNCR), first described by Lyon [212], is a method to reduce the emission of NO_x from combustion by injecting a selective reductant such as ammonia (NH_3) or urea ($CO(NH_2)_2$) into the furnace, where it can react with NO in the flue gas to form N_2. However, the reductant can be oxidized as well to form NO_x. The selectivity for the reductive reactions decreases with increasing temperature [228] while the rate of the initiation reaction simultaneously increases. This limits the SNCR process to a narrow temperature interval, or window, where the lower temperature limit for the interval is determined by the residence time.

Ammonia Injection

Several investigators have modeled the process using a large number of elementary reactions. A simple empirical model has been proposed by Fenimore [92], which is based on experimental measurements. However, the model was found to be unsuitable for practical applications. Ostberg and Dam-Johansen [260] proposed a two-step scheme describing the SNCR process as shown in Figure 13.1.2, which is a single initiation step followed by two parallel reaction pathways: one leading to NO reduction, and the other to NO formation.

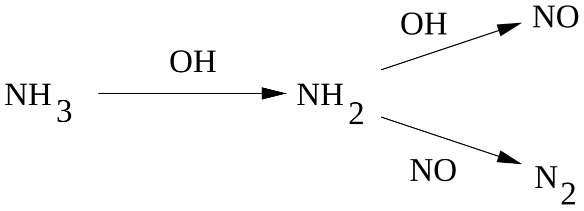


Figure 13.1.2: Simplified Reaction Mechanism for the SNCR Process

$$NO + NH_3 + \frac{1}{4}O_2 \quad \longrightarrow \quad N_2 + \frac{3}{2}H_2O \tag{13.1-87}$$

$$NH_3 + \frac{5}{4}O_2 \quad \longrightarrow \quad NO + \frac{3}{2}H_2O \tag{13.1-88}$$

The reaction orders of NO and NH_3 at 4% volume O_2 and the empirical rate constants k_r and $k_{\rm ox}$ for Equations 13.1-87 and 13.1-88, respectively, have been estimated from work done by Brouwer et al. [40]. The reaction order of NO was found to be 1 for Equation 13.1-87 and the order of NH_3 was found to be 1 for both reactions. As such, the following reaction rates for NO and NH_3, at 4% volume O_2, were proposed:

$$\mathcal{R}_{\rm NO} = -k_r[{\rm NO}][{\rm NH_3}] + k_{\rm ox}[{\rm NH_3}][{\rm O_2}] \tag{13.1-89}$$

$$\mathcal{R}_{\rm NH_3} = -k_r[{\rm NO}][{\rm NH_3}] - k_{\rm ox}[{\rm NH_3}][{\rm O_2}] \tag{13.1-90}$$

The rate constants k_r and $k_{\rm ox}$ have units of m^3/gmol-s, and are defined as

$$k_r = 4.24 \times 10^2 T^{5.30} e^{-E_r/RT}; \quad k_{\rm ox} = 3.50 \times 10^{-1} T^{7.65} e^{-E_{\rm ox}/RT}$$

where $E_r = 349937.06$ J/gmol and $E_{\rm ox} = 524487.005$ J/gmol.

This model has been shown to give reasonable predictions of the SNCR process in pulverized coal and fluidized bed combustion applications. The model also captures the influence of the most significant parameters for SNCR, which are the temperature of the flue gas at the injection position, the residence time in the relevant temperature interval, the NH_3 to NO molar ratio, and the effect of combustible additives. This model overestimates the NO reduction for temperatures above the optimum temperature by an amount similar to that of the detailed kinetic model of Miller and Bowman [228].

i The SNCR process naturally occurs when NH_3 is present in the flame as a fuel N intermediate. For this reason, even if the SNCR model is not activated and there is no reagent injection, the natural SNCR process may still occur in the flame. The temperature range or "window" at which SNCR may occur is 1073 K $<$ T $<$ 1373 K. To model your case without using the natural SNCR process, please contact your support engineer for information on how to deactivate it.

Urea Injection

Urea as a reagent for the SNCR process is similar to that of injecting ammonia and has been used in the power station combustors to reduce NO emissions successfully. However, both reagents, ammonia and urea, have major limitations as a NO_x reducing agent. The narrow temperature "window" of effectiveness and mixing limitations are difficult factors to handle in a large combustor. The use of urea instead of ammonia as the reducing agent is attractive because of the ease of storage and handling of the reagent.

The SNCR process using urea is a combination of Thermal DeNOx (SNCR with ammonia) and RAPRENOx (SNCR using cyanuric acid that, under heating, sublimes and decomposes into isocyanic acid) since urea most probably decomposes into ammonia and isocyanic acid [228].

One problem of SNCR processes using urea is that slow decay of HNCO as well as the reaction channels leading to N2O and CO can significantly increase the emission of pollutants other than NO. Urea seems to involve a significant emission of carbon-containing pollutants, such as CO and HNCO.

Also, some experimental observations [297] show that SNCR using urea is effective in a narrow temperature window that is shifted toward higher temperatures when compared to Thermal DeNOx processes at the same value of the ratio of nitrogen in the reducing agent and in NO in the feed, β. The effect of increasing the β value is to increase the efficiency of abatement, while the effect of increasing O_2 concentration depends on the temperature considered.

The model described here is proposed by Brouwer et al. [40] and is a seven-step reduced kinetic mechanism. Brouwer et al. [40] assumes that the breakdown of urea is instantaneous and 1 mole of urea is assumed to produce 1.1 moles of NH_3 and 0.9 moles of HNCO. The work of Rota et al. [297] proposed a finite rate two-step mechanism for the breakdown of urea into ammonia and HNCO.

The seven-step reduced mechanism is given in Table 13.1.2 and the two-step urea breakdown mechanism is given in Table 13.1.3.

Table 13.1.2: Seven-Step Reduced Mechanism for SNCR with Urea

Reaction	A	b	E
$NH_3 + NO \rightarrow N_2 + H_2O + H$	4.24E+02	5.30	349937.06
$NH_3 + O_2 \rightarrow NO + H_2O + H$	3.500E-01	7.65	524487.005
$HNCO + M \rightarrow H + NCO + M$	2.400E+08	0.85	284637.8
$NCO + NO \rightarrow N_2O + CO$	1.000E+07	0.00	-1632.4815
$NCO + OH \rightarrow NO + CO + H$	1.000E+07	0.00	0
$N_2O + OH \rightarrow N_2 + O_2 + H$	2.000E+06	0.00	41858.5
$N_2O + M \rightarrow N_2 + O + M$	6.900E+17	-2.5	271075.646

Table 13.1.3: Two-Step Urea Breakdown Process

Reaction	A	b	E
$CO(NH_2)_2 \rightarrow NH_3 + HNCO$	1.27E+04	0	65048.109
$CO(NH_2)_2 + H_2O \rightarrow 2NH_3 + CO_2$	6.13E+04	0	87819.133

where the units of A, in Tables 13.1.2 and 13.1.3, are m-gmol-sec and E units are J/gmol.

Transport Equations for Urea, HNCO, and NCO

When the SNCR model with urea injection is employed, in addition to the usual transport equations, ANSYS FLUENT solves the following three additional mass transport equations for urea, HNCO and NCO species.

$$\frac{\partial}{\partial t}(\rho Y_{\mathrm{CO(NH_2)_2}}) + \nabla \cdot (\rho \vec{v} Y_{\mathrm{CO(NH_2)_2}}) = \nabla \cdot (\rho \mathcal{D} Y_{\mathrm{CO(NH_2)_2}}) + S_{\mathrm{CO(NH_2)_2}} \tag{13.1-91}$$

$$\frac{\partial}{\partial t}(\rho Y_{\mathrm{HNCO}}) + \nabla \cdot (\rho \vec{v} Y_{\mathrm{HNCO}}) = \nabla \cdot (\rho \mathcal{D} Y_{\mathrm{HNCO}}) + S_{\mathrm{HNCO}} \tag{13.1-92}$$

$$\frac{\partial}{\partial t}(\rho Y_{\rm NCO}) + \nabla \cdot (\rho \vec{v} Y_{\rm NCO}) = \nabla \cdot (\rho \mathcal{D} Y_{\rm NCO}) + S_{\rm NCO} \tag{13.1-93}$$

where $Y_{\rm CO(NH_2)_2}$, $Y_{\rm HNCO}$ and $Y_{\rm NCO}$ are mass fractions of urea, HNCO and NCO in the gas phase. Source terms $S_{\rm CO(NH_2)_2}$, $S_{\rm HNCO}$ and $S_{\rm NCO}$ are determined according to the rate equations given in Tables 13.1.2 and 13.1.3 and the additional source terms due to reagent injection. These additional source terms are determined next. The source terms in the transport equations can be written as follows:

$$S_{\rm CO(NH_2)_2} = S_{\rm pl,CO(NH_2)_2} + S_{\rm CO(NH_2)_2-reac} \tag{13.1-94}$$

$$S_{\rm HNCO} = S_{\rm pl,HNCO} + S_{\rm HNCO-reac} \tag{13.1-95}$$

$$S_{\rm NCO} = S_{\rm NCO-reac} \tag{13.1-96}$$

Apart from the source terms for the above three species, additional source terms for NO, NH_3 and N_2O are also determined as follows, which should be added to the previously calculated sources due to fuel $\rm NO_x$:

$$S_{\rm NO} = S_{\rm NO-reac} \tag{13.1-97}$$

$$S_{\rm NH_3} = S_{\rm pl,NH_3} + S_{\rm NH_3-reac} \tag{13.1-98}$$

$$S_{\rm N_2O} = S_{\rm N_2O-reac} \tag{13.1-99}$$

Source terms $S_{\rm i-reac}$ for $i^{\rm th}$ species are determined from the rate equations given in Tables 13.1.2 and 13.1.3.

Urea Production due to Reagent Injection

The rate of urea production is equivalent to the rate of reagent release into the gas phase through droplet evaporation:

$$S_{\rm pl,CO(NH_2)_2} = \frac{S_{reagent}}{V} \tag{13.1-100}$$

where $S_{reagent}$ is the rate of reagent release from the liquid droplets to the gas phase (kg/s) and V is the cell volume (m^3).

NH_3 Production due to Reagent Injection

If the urea decomposition model is set to the user-specified option, then the rate of NH_3 production is proportional to the rate of reagent release into the gas phase through droplet evaporation:

$$S_{\mathrm{pl,NH_3}} = MCF_{NH_3} \frac{S_{reagent}}{V} \tag{13.1-101}$$

where $S_{reagent}$ is the rate of reagent release from the liquid droplets to the gas phase (kg/s), MCF_{NH_3} is the mole fraction of NH_3 in the $NH_3/HNCO$ mixture created from urea decomposition and V is the cell volume (m^3).

HNCO Production due to Reagent Injection

If the urea decomposition model is set to the user-specified option, then the rate of HNCO production is proportional to the rate of reagent release into the gas phase through droplet evaporation:

$$S_{\mathrm{pl,HNCO}} = MCF_{HNCO} \frac{S_{reagent}}{V} \tag{13.1-102}$$

where $S_{reagent}$, the injection source term, is the rate of reagent release from the liquid droplets to the gas phase (kg/s), MCF_{HNCO} is the mole fraction of HNCO in the $NH_3/HNCO$ mixture created from urea decomposition and V is the cell volume (m^3).

> *i* The mole conversion fractions (MCF) for species NH_3 and HNCO are determined through the user species values such that if one mole of urea decomposes into 1.1 moles of NH_3 and 0.9 moles of HNCO, then MCF_{NH_3} = 0.55 and MCF_{HNCO} = 0.45. When the user-specified option is used for urea decomposition, then $S_{\mathrm{pl,CO(NH_2)_2}} = 0$.

However, the default option for urea decomposition is through rate limiting reactions given in Table 13.1.3 and the source terms are calculated accordingly. In this case, both values of S_{pl,NH_3} and $S_{pl,HNCO}$ are zero.

13.1.9 NOx Formation in Turbulent Flows

The kinetic mechanisms of NO_x formation and destruction described in the preceding sections have all been obtained from laboratory experiments using either a laminar premixed flame or shock-tube studies where molecular diffusion conditions are well defined. In any practical combustion system, however, the flow is highly turbulent. The turbulent mixing process results in temporal fluctuations in temperature and species concentration that will influence the characteristics of the flame.

The relationships among NO_x formation rate, temperature, and species concentration are highly nonlinear. Hence, if time-averaged composition and temperature are employed in any model to predict the mean NO_x formation rate, significant errors will result. Temperature and composition fluctuations must be taken into account by considering the probability density functions which describe the time variation.

The Turbulence-Chemistry Interaction Model

In turbulent combustion calculations, ANSYS FLUENT solves the density-weighted time-averaged Navier-Stokes equations for temperature, velocity, and species concentrations or mean mixture fraction and variance. To calculate NO concentration, a time-averaged NO formation rate must be computed at each point in the domain using the averaged flow-field information.

Methods of modeling the mean turbulent reaction rate can be based on either moment methods [380] or probability density function (PDF) techniques [148]. ANSYS FLUENT uses the PDF approach.

The PDF method described here applies to the NO_x transport equations only. The preceding combustion simulation can use either the generalized finite-rate chemistry model by Magnussen and Hjertager or the non-premixed combustion model. For details on these models, refer to Chapters 7 and 8.

The PDF Approach

The PDF method has proven very useful in the theoretical description of turbulent flow [149]. In the ANSYS FLUENT NO_x model, a single- or joint-variable PDF in terms of a normalized temperature, species mass fraction, or the combination of both is used to predict the NO_x emission. If the non-premixed or partially premixed combustion model is used to model combustion, then a one- or two-variable PDF in terms of mixture fraction(s) is also available. The mean values of the independent variables needed for the PDF construction are obtained from the solution of the transport equations.

The General Expression for the Mean Reaction Rate

The mean turbulent reaction rate $\overline{w}$ can be described in terms of the instantaneous rate w and a single or joint PDF of various variables. In general,

$$\overline{w} = \int \cdots \int w(V_1, V_2, \ldots) P(V_1, V_2, \ldots) dV_1 dV_2 \ldots \qquad (13.1\text{-}103)$$

where V_1, V_2,... are temperature and/or the various species concentrations present. P is the probability density function (PDF).

The Mean Reaction Rate Used in ANSYS FLUENT

The PDF is used for weighting against the instantaneous rates of production of NO (e.g., Equation 13.1-15) and subsequent integration over suitable ranges to obtain the mean turbulent reaction rate. Hence we have

$$\overline{S}_{\rm NO} = \int S_{\rm NO}(V_1) P_1(V_1) dV_1 \qquad (13.1\text{-}104)$$

or, for two variables

$$\overline{S}_{\rm NO} = \int \int S_{\rm NO}(V_1, V_2) P(V_1, V_2) dV_1 dV_2 \qquad (13.1\text{-}105)$$

where $\overline{S}_{\rm NO}$ is the mean turbulent rate of production of NO, $\rm S_{NO}$ is the instantaneous rate of production given by, for example, Equation 13.1-15, and $P_1(V_1)$ and $P(V_1, V_2)$ are the PDFs of the variables V_1 and, if relevant, V_2. The same treatment applies for the HCN or NH_3 source terms.

Equation 13.1-104 or 13.1-105 must be integrated at every node and at every iteration. For a PDF in terms of temperature, the limits of integration are determined from the minimum and maximum values of temperature in the combustion solution. For a PDF in terms of mixture fraction, the limits of the integrations in Equation 13.1-104 or 13.1-105 are determined from the values stored in the look-up tables.

Statistical Independence

In the case of the two-variable PDF, it is further assumed that the variables V_1 and V_2 are statistically independent so that $P(V_1, V_2)$ can be expressed as

$$P(V_1, V_2) = P_1(V_1) P_2(V_2) \qquad (13.1\text{-}106)$$

The Beta PDF Option

ANSYS FLUENT can assume P to be a two-moment beta function that is appropriate for combustion calculations [123, 231]. The equation for the beta function is

$$P(V) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} V^{\alpha-1}(1-V)^{\beta-1} = \frac{V^{\alpha-1}(1-V)^{\beta-1}}{\displaystyle\int_0^1 V^{\alpha-1}(1-V)^{\beta-1}dV} \tag{13.1-107}$$

where $\Gamma(\ )$ is the Gamma function, α and β depend on $\overline{m}$, the mean value of the quantity in question, and its variance, σ^2:

$$\alpha = \overline{m}\left(\frac{\overline{m}(1-\overline{m})}{\sigma^2} - 1\right) \tag{13.1-108}$$

$$\beta = (1-\overline{m})\left(\frac{\overline{m}(1-\overline{m})}{\sigma^2} - 1\right) \tag{13.1-109}$$

The beta function requires that the independent variable V assume values between 0 and 1. Thus, field variables such as temperature must be normalized. See Section 21.1.1: Setting Turbulence Parameters in the separate User's Guide for information on using the beta PDF when using single-mixture fraction models and two-mixture fraction models.

The Gaussian PDF Option

ANSYS FLUENT can also assume P to exhibit a clipped Gaussian form with delta functions at the tails.

The cumulative density function for a Gaussian PDF ($G_{\rm CDF}$) may be expressed in terms of the error function as follows:

$$G_{\rm CDF} = \frac{1}{2}\left(1 + erf\left((m-\overline{m})/\sqrt{2\sigma^2}\right)\right) \tag{13.1-110}$$

where $erf(\ )$ is the error function, m is the quantity in question, and $\overline{m}$ and σ^2 are the mean and variance values of m, respectively. The error function may be expressed in terms of the incomplete gamma function ($gammp(\ )$):

$$\begin{aligned} \text{for } m < 0: \ erf(m) &= -gammp(0.5,\, m^2) \\ \text{for } m \geq 0: \ erf(m) &= gammp(0.5,\, m^2) \end{aligned} \tag{13.1-111}$$

The Calculation Method for σ^2

The variance, σ^2, can be computed by solving the following transport equation during the combustion calculation or pollutant postprocessing stage:

$$\frac{\partial}{\partial t}\left(\rho\sigma^2\right) + \nabla \cdot (\rho\vec{v}\sigma^2) = \nabla\left(\frac{\mu_t}{\sigma_t}\nabla\sigma^2\right) + C_g\mu_t(\nabla\overline{m})^2 - C_d\rho\frac{\epsilon}{k}\sigma^2 \qquad (13.1\text{-}112)$$

where the constants σ_t, C_g and C_d take the values 0.85, 2.86, and 2.0, respectively.

Note that the previous equation may only be solved for temperature. This solution may be computationally intensive, and therefore may not always be applicable for a postprocessing treatment of $\mathrm{NO_x}$ prediction. When this is the case or when solving for species, the calculation of σ^2 is instead based on an approximate form of the variance transport equation (also referred to as the algebraic form). The approximate form assumes equal production and dissipation of variance, and is as follows:

$$\sigma^2 = \frac{\mu_t}{\rho}\frac{k}{\epsilon}\frac{C_g}{C_d}(\nabla\overline{m})^2 = \frac{\mu_t}{\rho}\frac{k}{\epsilon}\frac{C_g}{C_d}\left[\left(\frac{\partial\overline{m}}{\partial x}\right)^2 + \left(\frac{\partial\overline{m}}{\partial y}\right)^2 + \left(\frac{\partial\overline{m}}{\partial z}\right)^2\right] \qquad (13.1\text{-}113)$$

The term in the brackets is the dissipation rate of the independent variable.

For a PDF in terms of mixture fraction, the mixture fraction variance has already been solved as part of the basic combustion calculation, so no additional calculation for σ^2 is required.

13.2 SOx Formation

The following sections include information on the theory used in the $\mathrm{SO_x}$ model. For information about using the $\mathrm{SO_x}$ models in ANSYS FLUENT, see Section 21.2.1: Using the $\mathrm{SO_x}$ Model in the separate User's Guide.

13.2.1 Overview

Sulfur exists in coal as organic sulfur, pyretic and sulfates [1], and exists in liquid fuels mostly in organic form [235], with mass fractions ranging from 0.5% to 3%. All $\mathrm{SO_x}$ emissions are produced because of the oxidation of fuel-bound sulfur. During the combustion process, fuel sulfur is oxidized to SO_2 and SO_3. A portion of the gaseous $\mathrm{SO_x}$ will condense on the particles, attaching an amount of water and thus forming sulfuric acid, or may react further to form sulfates. While $\mathrm{SO_x}$ emissions are the main cause of acid rain, SO_3 also contributes to particulate emissions, and is responsible for corrosion of combustion equipment. Furthermore, there is a growing interest in the interaction of sulfur species with the nitrogen oxide chemistry [235], as NO levels are affected by the presence of sulfur species. The evidence to date indicates that thermal NO levels (Section 13.1.3: Thermal NOx Formation) are reduced in the presence of SO_2. However, the effect of sulfur compounds on the fuel $\mathrm{NO_x}$ formation is yet to be clarified.

Sulfur emissions are regulated from stationary sources and from automotive fuels. Sulfur pollutants can be captured during the combustion process, or with after treatment methods, such as wet or dry scrubbing. Coal fired boilers are by far the biggest single $\mathrm{SO_x}$ emissions source, accounting for over 50% of total SO_2 emissions [55].

For higher sulfur concentrations in the fuel, the $\mathrm{SO_x}$ concentration field should be resolved together with the main combustion calculation using any of the ANSYS FLUENT reaction models. For cases where the sulfur fraction in fuel is low, the post-processing option can be used, which solves transport equations for H_2S, SO_2, SO, SH, and SO_3.

The Formation of SOx

The $\mathrm{SO_x}$ model incorporates the following stages:

1. Sulfur release from the fuel

   For liquid fuels, one can conveniently assume that sulfur is released as $H_2\mathrm{S}$ [235]. However, the process is more complicated in the case of coal; here some of the sulfur is decomposed into the gas phase during devolatilization as H_2S, COS, SO_2 and CS_2, while part of the sulfur is retained in the char to be oxidized at a later stage. The percentage of sulfur retained in char is rank dependent [1].

2. Sulfur reaction in the gas phase

   In oxygen rich flames the predominant sulfur species are SO, SO_2 and SO_3. At lower oxygen concentrations H_2S, S_2 and SH are also present in significant proportions, while SO_3 becomes negligible [235]. In PCGC-3 as well as in the works of Norman et al. [251] the gas phase sulfur species are assumed to be in equilibrium.

3. Sulfur retention in sorbents

   Sulfur pollutants can be absorbed by sorbent particles, injected either in situ, or in the post flame region.

For low sulfur fuels, we can assume that sulfur is mainly released as H_2S. The rate of release can be determined similarly to that of fuel-bound N. For the char S it can be assumed that SO_2 is produced directly at the same rate as that of char burnout. Transport equations for H_2S, SO_2, SO, SH, and SO_3 species are incorporated and an appropriate reaction set has been developed as described in the ensuing sections.

13.2.2 Governing Equations for SOx Transport

ANSYS FLUENT solves the mass transport equations for the SO_2 species, taking into account convection, diffusion, production and consumption of SO_2 and related species. This approach is completely general, being derived from the fundamental principle of mass conservation. The effect of residence time in $\mathrm{SO_x}$ mechanisms, a Lagrangian reference frame concept, is included through the convection terms in the governing equations written in the Eulerian reference frame. If all fuel sulfur is assumed to convert directly to SO_2 and the other product and intermediate species are assumed negligible, then only the SO_2 species transport equation is needed:

$$\frac{\partial}{\partial t}(\rho Y_{\mathrm{SO_2}}) + \nabla \cdot (\rho \vec{v} Y_{\mathrm{SO_2}}) = \nabla \cdot (\rho \mathcal{D} \nabla Y_{\mathrm{SO_2}}) + S_{\mathrm{SO_2}} \tag{13.2-1}$$

As discussed in Section 13.2.3: Reaction Mechanisms for Sulfur Oxidation, $\mathrm{SO_x}$ formation mechanisms involve multiple reactions among multiple species, and tracking sulfur-containing intermediate species is important. ANSYS FLUENT solves transport equations for the H_2S, SO_3, SO, and SH species in addition to the SO_2 species:

$$\frac{\partial}{\partial t}(\rho Y_{\mathrm{H_2S}}) + \nabla \cdot (\rho \vec{v} Y_{\mathrm{H_2S}}) = \nabla \cdot (\rho \mathcal{D} Y_{\mathrm{H_2S}}) + S_{\mathrm{H_2S}} \tag{13.2-2}$$

$$\frac{\partial}{\partial t}(\rho Y_{\mathrm{SO_3}}) + \nabla \cdot (\rho \vec{v} Y_{\mathrm{SO_3}}) = \nabla \cdot (\rho \mathcal{D} Y_{\mathrm{SO_3}}) + S_{\mathrm{SO_3}} \tag{13.2-3}$$

$$\frac{\partial}{\partial t}(\rho Y_{\mathrm{SO}}) + \nabla \cdot (\rho \vec{v} Y_{\mathrm{SO}}) = \nabla \cdot (\rho \mathcal{D} Y_{\mathrm{SO}}) + S_{\mathrm{SO}} \tag{13.2-4}$$

$$\frac{\partial}{\partial t}(\rho Y_{\mathrm{SH}}) + \nabla \cdot (\rho \vec{v} Y_{\mathrm{SH}}) = \nabla \cdot (\rho \mathcal{D} Y_{\mathrm{SH}}) + S_{\mathrm{SH}} \tag{13.2-5}$$

where $Y_{\mathrm{SO_2}}$, $Y_{\mathrm{H_2S}}$, $Y_{\mathrm{SO_3}}$, Y_{SO}, and Y_{SH} are mass fractions of SO_2, H_2S, SO_3, SO, and SH in the gas phase. The source terms $S_{\mathrm{SO_2}}$, $S_{\mathrm{H_2S}}$, $S_{\mathrm{SO_3}}$, S_{SO}, and S_{SH} are to be determined depending on the form of fuel sulfur release (SO_2 and/or H_2S) and inclusion of SO_3, SO and SH in the $\mathrm{SO_x}$ mechanism.

13.2.3 Reaction Mechanisms for Sulfur Oxidation

A detailed reaction mechanism for sulfur oxidation has been proposed by Kramlich [172]. The mechanism consists of 20 reversible reactions and includes 12 species (S, S_2, SH, SO, SO_2, H_2S, H, H_2, OH, H_2O, O and O_2). The mechanism has been reduced to 8 steps and 10 species (with S and S_2 removed), and validated in Perfectly Stirred Reactor (PSR) and Plug Flow Reactor (PFR) simulations. Table 13.2.1 lists the reduced mechanism with the modified rate constants. For reduction calculations O and OH concentrations have been calculated through partial equilibrium assumptions based on O_2 and H_2O concentrations, respectively. N_2 was used as the dilutant. Since each reaction of the eight-step reduced mechanism is reversible, for each adjacent pair of reactions given in Table 13.2.1, the second reaction is in fact the reverse reaction of the first.

The reduced mechanism given in Table 13.2.1 closely follows the SO_2 concentration levels but slightly overpredicts the H_2S concentrations at temperatures below 1500 K. Above 1500 K, both mechanisms are in close agreement for SO_2 and H_2S concentration predictions. However, SO and SH are not well correlated by the reduced mechanism when compared against the predictions using the original detailed mechanism.

A major concern in these mechanisms is the presence of H radical and the method in which to calculate its concentration in the flow field. At present, the concentration of H radical is assumed to be proportional to the O radical concentration, which can be evaluated from one of the existing methods in ANSYS FLUENT; viz. Partial Equilibrium (Section 13.1.3: Method 2: Partial Equilibrium Approach) or Equilibrium (Section 13.1.3: Method 1: Equilibrium Approach). The user is then given the option to vary the proportionality constant. Although this assumption is open to debate, the lack of simple relation to calculate the H radical concentration in a flame has prompted the present choice.

Present implementation allows the user to either include or remove $\mathrm{SO_3}$ from the calculations. Also, depending on the form of fuel sulfur release (e.g., H_2S or $\mathrm{SO_2}$) the species H_2S may or may not be present for the calculation. The user is also given the extended option of partitioning the intermediate fuel sulfur species to H_2S and SO_2. However, there is no literature to guide the user on how to select a correct partition fraction.

A is in $\mathrm{m^3}$/gmol-s, E is J/gmol (assumed 1 cal = 4.18585 J), A units for the thirteenth reaction is $\mathrm{m^6/gmol^2}$-s, and A units for the fifteenth reaction is $\mathrm{m^6/gmol^2}$-s.

In addition, the following two reactions were included in ANSYS FLUENT to complete the $\mathrm{SO_x}$ mechanism, with the rate constants taken from Hunter's work [135].

$$SO_2 + O + M \Longleftrightarrow SO_3 + M \qquad (13.2\text{-}6)$$

M = argon, nitrogen, oxygen
k_{f1} = 3.63 x 10^2 exp(+4185.85/RT) $\mathrm{m^6/gmol^2}$/sec

Table 13.2.1: Eight-Step Reduced Mechanism (Rate Constant $k = AT^b exp(-E/RT)$)

Reaction	A	b	E
$H_2S + H \rightarrow SH + H_2$	1.819702E+07	0.0E+00	7.484300E+03
$SH + H_2 \rightarrow H_2S + H$	9.375623E+06	0.0E+00	6.253660E+04
$OH + H_2S \rightarrow H_2O + SH$	1.380385E+02	0.0E+00	3.742150E+03
$H_2O + SH \rightarrow OH + H_2S$	3.104557E+07	0.0E+00	1.218543E+05
$SO + OH \rightarrow H + SO_2$	1.621810E+08	0.0E+00	2.565926E+03
$H + SO_2 \rightarrow SO + OH$	7.691299E+09	0.0E+00	1.187023E+05
$SH + O \rightarrow SO + H$	3.548135E+08	0.0E+00	2.687316E+03
$SO + H \rightarrow SH + O$	2.985385E+09	0.0E+00	1.694600E+05
$O + H_2S \rightarrow SH + OH$	4.365162E+03	0.0E+00	1.380493E+04
$SH + OH \rightarrow O + H_2S$	9.885528E+08	0.0E+00	6.035996E+04
$SO + O_2 \rightarrow SO_2 + O$	4.466832E+05	0.0E+00	2.703222E+04
$SO_2 + O \rightarrow SO + O_2$	1.663412E+06	0.0E+00	7.613643E+04
$H + SH + M \rightarrow H_2S + M$	1.096478E+03	0.0E+00	0.000000E+00
$H_2S + M \rightarrow H + SH + M$	8.669613E+14	0.0E+00	3.819463E+05
$SO + O + M \rightarrow SO_2 + M$	8.709647E+09 k	-1.8E+00	0.000000E+00
$SO_2 + M \rightarrow SO + O + M$	1.905464E+14	0.0E+00	5.207365E+05

where R = 8.313 J/gmol-K

k$_{r1}$ = 7.41 x 10^{14} exp(-346123.75/RT) m^3/gmol/sec

$$SO_3 + O \Longleftrightarrow SO_2 + O_2 \tag{13.2-7}$$

k$_{f2}$ = 1.2 x 10^6 exp(-39765.575/RT) m^3/gmol/sec

The reverse rate of Equation 13.2-7 was determined through the equilibrium constant for that equation.

13.2.4 SO_2 and H_2S Production in a Gaseous Fuel

The rate of SO_2 or H_2S production is equivalent to the rate of combustion of the fuel:

$$S_{\rm pl,i} = \frac{\mathcal{R}_{\rm cf}\; Y_{\rm S,fuel}\; M_{w,{\rm i}}}{M_{w,{\rm S}}} \tag{13.2-8}$$

where $S_{\rm pl,i}$ = source of i (kg/m^3-s), where $i = SO_2$ or H_2S
$\mathcal{R}_{\rm cf}$ = mean limiting reaction rate of fuel (kg/m^3-s)
$Y_{\rm S,fuel}$ = mass fraction of sulfur in the fuel

The mean limiting reaction rate of fuel, $\mathcal{R}_{\rm cf}$, is calculated from the Magnussen combustion model, so the gaseous fuel option for SO$_{\rm x}$ formation is available only when the generalized finite-rate model is used.

13.2.5 SO_2 and H_2S Production in a Liquid Fuel

The rate of SO_2 or H_2S production is equivalent to the rate of fuel release into the gas phase through droplet evaporation:

$$S_{\rm pl,i} = \frac{S_{\rm fuel}\; Y_{\rm S,fuel}\; M_{w,{\rm i}}}{M_{w,{\rm S}} V} \tag{13.2-9}$$

where $S_{\rm pl,i}$ = source of i (kg/m^3-s), where $i = SO_2$ or H_2S
$S_{\rm fuel}$ = rate of fuel release from the liquid droplets to the gas (kg/s)
$Y_{\rm S,fuel}$ = mass fraction of sulfur in the fuel
V = cell volume (m^3)

13.2.6 SO_2 and H_2S Production from Coal

For coal, it is assumed that sulfur is distributed between the volatiles and the char. Since there is no reason to assume that S is equally distributed between the volatiles and the char, the fraction of S in the volatiles and the char should be specified separately.

SO_2 and H_2S from Char

The source of SO_2 and H_2S from the char is related to the rate of char combustion:

$$S_{\rm char,i} = \frac{S_c \, Y_{\rm S,char} M_{w,\rm i}}{M_{w,\rm S} V} \qquad (13.2\text{-}10)$$

where S_c = char burnout rate (kg/s)
$S_{\rm char,i}$ = source of i (kg/m^3-s) in char, where $i = SO_2$ or H_2S
$Y_{\rm S,char}$ = mass fraction of sulfur in char
V = cell volume (m^3)

SO_2 and H_2S from Volatiles

The source of SO_2 and H_2S from the volatiles is related to the rate of volatile release:

$$S_{\rm vol,i} = \frac{S_{\rm vol} \, Y_{\rm S,vol} M_{w,\rm i}}{M_{w,\rm S} V} \qquad (13.2\text{-}11)$$

where $S_{\rm vol,i}$ = source of volatiles originating from
the coal particles into the gas phase (kg/s), where $i = SO_2$ or H_2S
$Y_{\rm S,vol}$ = mass fraction of sulfur in the volatiles
V = cell volume (m^3)

13.2.7 SOx Formation in Turbulent Flows

The kinetic mechanisms of SO$_{\rm x}$ formation and destruction are obtained from laboratory experiments in a similar fashion to the NO$_{\rm x}$ model. In any practical combustion system, however, the flow is highly turbulent. The turbulent mixing process results in temporal fluctuations in temperature and species concentration that will influence the characteristics of the flame.

The relationships among SO$_{\rm x}$ formation rate, temperature, and species concentration are highly nonlinear. Hence, if time-averaged composition and temperature are employed in any model to predict the mean SO$_{\rm x}$ formation rate, significant errors will result. Temperature and composition fluctuations must be taken into account by considering the probability density functions which describe the time variation.

The Turbulence-Chemistry Interaction Model

In turbulent combustion calculations, ANSYS FLUENT solves the density-weighted time-averaged Navier-Stokes equations for temperature, velocity, and species concentrations or mean mixture fraction and variance. To calculate SO_2 concentration, a time-averaged SO_2 formation rate must be computed at each point in the domain using the averaged flow-field information.

The PDF Approach

The PDF method has proven very useful in the theoretical description of turbulent flow [149]. In the ANSYS FLUENT $\mathrm{SO_x}$ model, a single- or joint-variable PDF in terms of a normalized temperature, species mass fraction, or the combination of both is used to predict the $\mathrm{SO_x}$ emission. If the non-premixed combustion model is used to model combustion, then a one- or two-variable PDF in terms of mixture fraction(s) is also available. The mean values of the independent variables needed for the PDF construction are obtained from the solution of the transport equations.

The Mean Reaction Rate

The mean turbulent reaction rate described in Section 13.1.9: The General Expression for the Mean Reaction Rate for the $\mathrm{NO_x}$ model also applies to the $\mathrm{SO_x}$ model. The PDF is used for weighting against the instantaneous rates of production of SO_2 and subsequent integration over suitable ranges to obtain the mean turbulent reaction rate as described in Equations 13.1-104 and 13.1-105 for $\mathrm{NO_x}$.

The PDF Options

As is the case with the $\mathrm{NO_x}$ model, P can be calculated as either a two-moment beta function or as a clipped Gaussian function, as appropriate for combustion calculations [123, 231]. Equations 13.1-107 – 13.1-111 apply to the $\mathrm{SO_x}$ model as well, with the variance σ^2 computed by solving a transport equation during the combustion calculation stage, using Equation 13.1-112 or Equation 13.1-113.

13.3 Soot Formation

Information about the theory behind soot formation is presented in the following sections. For information about using soot formation models in ANSYS FLUENT, see Section 21.3.1: Using the Soot Models in the separate User's Guide.

- Section 13.3.1: Overview and Limitations
- Section 13.3.2: Soot Model Theory

13.3.1 Overview and Limitations

ANSYS FLUENT provides four models for the prediction of soot formation in combustion systems. In addition, the predicted soot concentration can be coupled with radiation. That is, you can include the effect of soot on radiation absorption when you use the P-1, discrete ordinates, or discrete transfer radiation model with a variable absorption coefficient.

Predicting Soot Formation

ANSYS FLUENT predicts soot concentrations in a combustion system using one of four available models:

- the one-step Khan and Greeves model [162], in which ANSYS FLUENT predicts the rate of soot formation based on a simple empirical rate
- the two-step Tesner model [216, 349], in which ANSYS FLUENT predicts the formation of nuclei particles, with soot formation on the nuclei
- the Moss-Brookes model [39], in which ANSYS FLUENT predicts soot formation for methane flames (and higher hydrocarbon species, if appropriately modified) by solving transport equations for normalized radical nuclei concentration and the soot mass fraction
- the Moss-Brookes-Hall model [120], which is an extension of the Moss-Brookes model and is applicable for higher hydrocarbon fuels (e.g., kerosene)

The Khan and Greeves model is the default model used by ANSYS FLUENT when you include soot formation.

In the Khan and Greeves model and the Tesner model, combustion of the soot (and particle nuclei) is assumed to be governed by the Magnussen combustion rate [216]. Note that this limits the use of these soot formation models to turbulent flows. Both models are empirically-based, approximate models of the soot formation process in combustion systems. The detailed chemistry and physics of soot formation are quite complex and are only approximated in these models. You should view the results of the Khan and Greeves model and the Tesner model as qualitative indicators of your system performance unless you can undertake experimental validation of the results.

The Moss-Brookes model has less impiricism and should theoretically provide superior accuracy than the Khan and Greeves and Tesner models. The Hall extension provides further options for modeling higher hydrocarbon fuels. Note that the Moss-Brookes-Hall model is only available when the required species are present in the gas phase species list.

Restrictions on Soot Modeling

The following restrictions apply to soot formation models:

- You must use the pressure-based solver. The soot models are not available with either of the density-based solvers.
- The Khan and Greeves model and the Tesner model can model soot formation only for turbulent flows (whereas the Moss-Brookes model and the Moss-Brookes-Hall model can be used with both laminar and turbulent flows).
- The soot model cannot be used in conjunction with the premixed combustion model.

13.3.2 Soot Model Theory

The One-Step Soot Formation Model

In the one-step Khan and Greeves model [162], ANSYS FLUENT solves a single transport equation for the soot mass fraction:

$$\frac{\partial}{\partial t}(\rho Y_{\rm soot}) + \nabla \cdot (\rho \vec{v} Y_{\rm soot}) = \nabla \cdot \left(\frac{\mu_t}{\sigma_{\rm soot}} \nabla Y_{\rm soot} \right) + \mathcal{R}_{\rm soot} \tag{13.3-1}$$

where

$Y_{\rm soot}$ = soot mass fraction
$\sigma_{\rm soot}$ = turbulent Prandtl number for soot transport
$\mathcal{R}_{\rm soot}$ = net rate of soot generation (kg/m^3-s)

$\mathcal{R}_{\rm soot}$, the net rate of soot generation, is the balance of soot formation, $\mathcal{R}_{\rm soot,form}$, and soot combustion, $\mathcal{R}_{\rm soot,comb}$:

$$\mathcal{R}_{\text{soot}} = \mathcal{R}_{\text{soot,form}} - \mathcal{R}_{\text{soot,comb}} \tag{13.3-2}$$

The rate of soot formation is given by a simple empirical rate expression:

$$\mathcal{R}_{\text{soot,form}} = C_s p_{\text{fuel}} \phi^r e^{-E/RT} \tag{13.3-3}$$

where

C_s = soot formation constant (kg/N-m-s)
p_{fuel} = fuel partial pressure (Pa)
ϕ = equivalence ratio
r = equivalence ratio exponent
E/R = activation temperature (K)

The rate of soot combustion is the minimum of two rate expressions [216]:

$$\mathcal{R}_{\text{soot,comb}} = \min[\mathcal{R}_1,\ \mathcal{R}_2] \tag{13.3-4}$$

The two rates are computed as

$$\mathcal{R}_1 = A\rho Y_{\text{soot}} \frac{\epsilon}{k} \tag{13.3-5}$$

and

$$\mathcal{R}_2 = A\rho \left(\frac{Y_{\text{ox}}}{\nu_{\text{soot}}}\right) \left(\frac{Y_{\text{soot}}\nu_{\text{soot}}}{Y_{\text{soot}}\nu_{\text{soot}} + Y_{\text{fuel}}\nu_{\text{fuel}}}\right) \frac{\epsilon}{k} \tag{13.3-6}$$

where

A = constant in the Magnussen model
Y_{ox}, Y_{fuel} = mass fractions of oxidizer and fuel
ν_{soot}, ν_{fuel} = mass stoichiometries for soot and fuel combustion

The default constants for the one-step model are valid for a wide range of hydrocarbon fuels.

The Two-Step Soot Formation Model

The two-step Tesner model [349] predicts the generation of radical nuclei and then computes the formation of soot on these nuclei. ANSYS FLUENT thus solves transport equations for two scalar quantities: the soot mass fraction (Equation 13.3-1) and the normalized radical nuclei concentration:

$$\frac{\partial}{\partial t}(\rho b^*_{\rm nuc}) + \nabla \cdot (\rho \vec{v} b^*_{\rm nuc}) = \nabla \cdot \left(\frac{\mu_t}{\sigma_{\rm nuc}} \nabla b^*_{\rm nuc} \right) + \mathcal{R}^*_{\rm nuc} \qquad (13.3\text{-}7)$$

where

$b^*_{\rm nuc}$ = normalized radical nuclei concentration (particles $\times 10^{-15}$/kg)
$\sigma_{\rm nuc}$ = turbulent Prandtl number for nuclei transport
$\mathcal{R}^*_{\rm nuc}$ = normalized net rate of nuclei generation (particles $\times 10^{-15}$/m^3-s)

In these transport equations, the rates of nuclei and soot generation are the net rates, involving a balance between formation and combustion.

Soot Generation Rate

The two-step model computes the net rate of soot generation, $\mathcal{R}_{\rm soot}$, in the same way as the one-step model, as a balance of soot formation and soot combustion:

$$\mathcal{R}_{\rm soot} = \mathcal{R}_{\rm soot,form} - \mathcal{R}_{\rm soot,comb} \qquad (13.3\text{-}8)$$

In the two-step model, however, the rate of soot formation, $\mathcal{R}_{\rm soot,form}$, depends on the concentration of radical nuclei, $c_{\rm nuc}$:

$$\mathcal{R}_{\rm soot,form} = m_p(\alpha - \beta N_{\rm soot}) c_{\rm nuc} \qquad (13.3\text{-}9)$$

where

m_p = mean mass of soot particle (kg/particle)
$N_{\rm soot}$ = concentration of soot particles (particles/m^3)
$c_{\rm nuc}$ = radical nuclei concentration = $\rho b_{\rm nuc}$ (particles/m^3)
α = empirical constant (s^{-1})
β = empirical constant (m^3/particle-s)

The rate of soot combustion, $\mathcal{R}_{\rm soot,comb}$, is computed in the same way as for the one-step model, using Equations 13.3-4–13.3-6.

The default constants for the two-step model are for combustion of acetylene (C_2H_2). According to Ahmad et al. [2], these values should be modified for other fuels, since the sooting characteristics of acetylene are known to be different from those of saturated hydrocarbon fuels.

Nuclei Generation Rate

The net rate of nuclei generation in the two-step model is given by the balance of the nuclei formation rate and the nuclei combustion rate:

$$\mathcal{R}^*_{\text{nuc}} = \mathcal{R}^*_{\text{nuc,form}} - \mathcal{R}^*_{\text{nuc,comb}} \tag{13.3-10}$$

where

$\mathcal{R}^*_{\text{nuc,form}}$	= rate of nuclei formation (particles $\times 10^{-15}/\text{m}^3$-s)
$\mathcal{R}^*_{\text{nuc,comb}}$	= rate of nuclei combustion (particles $\times 10^{-15}/\text{m}^3$-s)

The rate of nuclei formation, $\mathcal{R}^*_{\text{nuc,form}}$, depends on a spontaneous formation and branching process, described by

$$\mathcal{R}^*_{\text{nuc,form}} = \eta_0 + (f - g)c^*_{\text{nuc}} - g_0 c^*_{\text{nuc}} N_{\text{soot}} \tag{13.3-11}$$

$$\eta_0 = a^*_0 c_{\text{fuel}} e^{-E/RT} \tag{13.3-12}$$

where

c^*_{nuc}	= normalized nuclei concentration ($= \rho b^*_{\text{nuc}}$)
a^*_0	= $a_0/10^{15}$
a_0	= pre-exponential rate constant (particles/kg-s)
c_{fuel}	= fuel concentration (kg/m^3)
$f - g$	= linear branching − termination coefficient (s^{-1})
g_0	= linear termination on soot particles (m^3/particle-s)

Note that the branching term, $(f - g)c^*_{\text{nuc}}$, in Equation 13.3-11 is included only when the kinetic rate, η_0, is greater than the limiting formation rate (10^5 particles/m^3-s, by default).

The rate of nuclei combustion is assumed to be proportional to the rate of soot combustion:

$$\mathcal{R}^*_{\text{nuc,comb}} = \mathcal{R}_{\text{soot,comb}} \frac{b^*_{\text{nuc}}}{Y_{\text{soot}}} \tag{13.3-13}$$

where the soot combustion rate, $\mathcal{R}_{\text{soot,comb}}$, is given by Equation 13.3-4.

The Moss-Brookes Model

The Moss-Brookes model solves transport equations for normalized radical nuclei concentration $b^*_{\rm nuc}$ and soot mass fraction $Y_{\rm soot}$:

$$\frac{\partial}{\partial t}(\rho Y_{\rm soot}) + \nabla\cdot(\rho\vec{v}Y_{\rm soot}) = \nabla\cdot\left(\frac{\mu_t}{\sigma_{\rm soot}}\nabla Y_{\rm soot}\right) + \frac{dM}{dt} \qquad (13.3\text{-}14)$$

$$\frac{\partial}{\partial t}(\rho b^*_{\rm nuc}) + \nabla\cdot(\rho\vec{v}b^*_{\rm nuc}) = \nabla\cdot\left(\frac{\mu_t}{\sigma_{\rm nuc}}\nabla b^*_{\rm nuc}\right) + \frac{1}{N_{\rm norm}}\frac{dN}{dt} \qquad (13.3\text{-}15)$$

where

$Y_{\rm soot}$ = soot mass fraction

M = soot mass concentration (kg/m^3)

$b^*_{\rm nuc}$ = normalized radical nuclei concentration (particles $\times 10^{-15}$/kg) = $\frac{N}{\rho N_{\rm norm}}$

N = soot particle number density (particles/m^3)

$N_{\rm norm}$ = 10^{15} particles

The instantaneous production rate of soot particles, subject to nucleation from the gas phase and coagulation in the free molecular regime, is given by

$$\frac{dN}{dt} = \underbrace{C_\alpha N_{\rm A}\left(\frac{X_{\rm prec}P}{RT}\right)^l \exp\left\{-\frac{T_\alpha}{T}\right\}}_{\rm Nucleation} - \underbrace{C_\beta\left(\frac{24RT}{\rho_{\rm soot}N_{\rm A}}\right)^{1/2} d_{\rm p}^{1/2}N^2}_{\rm Coagulation} \qquad (13.3\text{-}16)$$

where C_α, C_β and l are model constants. Here, $N_{\rm A}$ (= 6.022045x10^{26} kmol^{-1}) is the Avogadro number and $X_{\rm prec}$ is the mole fraction of soot precursor (for methane, the precursor is assumed to be acetylene, whereas for kerosene it is a combination of acetylene and benzene). The mass density of soot, $\rho_{\rm soot}$, is assumed to be 1800 kg/m^3 and $d_{\rm p}$ is the mean diameter of a soot particle. The nucleation rate for soot particles is taken to be proportional to the local acetylene concentration for methane. The activation temperature T_α for the nucleation reaction is that proposed by Lindstedt [199].

The source term for soot mass concentration is modeled by the expression

$$\begin{aligned}\frac{dM}{dt} \;=\; & \underbrace{M_{\rm P} C_\alpha \left(\frac{X_{\rm prec}P}{RT}\right)^{l} \exp\left\{-\frac{T_\alpha}{T}\right\}}_{\text{Nucleation}} + \underbrace{C_\gamma \left(\frac{X_{\rm sgs}P}{RT}\right)^{m} \exp\left\{-\frac{T_\gamma}{T}\right\} \left[(\pi N)^{1/3}\left(\frac{6M}{\rho_{\rm soot}}\right)^{2/3}\right]^{n}}_{\text{Surface Growth}} \\ & - \underbrace{C_{\rm oxid} C_\omega \eta_{\rm coll} \left(\frac{X_{\rm OH}P}{RT}\right)\sqrt{T}\,(\pi N)^{1/3}\left(\frac{6M}{\rho_{\rm soot}}\right)^{2/3}}_{\text{Oxidation}}\end{aligned} \tag{13.3-17}$$

where C_γ, $C_{\rm oxid}$, C_ω, m, and n are additional model constants. The constant $M_{\rm P}$ (= 144 kg/kgmol) is the mass of an incipient soot particle, here taken to consist of 12 carbon atoms. Even though the model is not found to be sensitive to this assumption, a nonzero initial mass is needed to begin the process of surface growth. Here, $X_{\rm sgs}$ is the mole fraction of the participating surface growth species. For paraffinic fuels, soot particles have been found to grow primarily by the addition of gaseous species at their surfaces, particularly acetylene that has been found in abundance in the sooting regions of laminar methane diffusion flames.

The model assumes that the hydroxyl radical is the dominant oxidizing agent in methane/air diffusion flames and that the surface-specific oxidation rate of soot by the OH radical may be formulated according to the model proposed by Fenimore and Jones [93]. Assuming a collision efficiency ($\eta_{\rm coll}$) of 0.04, the oxidation rate may be written as (Equation 13.3-17.

The process of determination of the exponents l, m, and n are explained in detail by Brookes and Moss [39]. The constants C_α and C_β are determined through numerical modeling of a laminar flame for which experimental data exists.

The set of constants proposed by Brookes and Moss for methane flames are given below:

C_α = 54 $\rm s^{-1}$ (model constant for soot inception rate)
T_α = 21000 K (activation temperature of soot inception)
C_β = 1.0 (model constant for coagulation rate)
C_γ = 11700 $\rm kg.m.kmol^{-1}.s^{-1}$ (surface growth rate scaling factor)
T_γ = 12100 K (activation temperature of surface growth rate)
C_ω = 105.8125 $\rm kg.m.kmol^{-1}.K^{-1/2}.s^{-1}$ (oxidation model constant)
$\eta_{\rm coll}$ = 0.04 (collisional efficiency parameter)
$C_{\rm oxid}$ = 0.015 (oxidation rate scaling parameter)

Note that the implementation of the Moss-Brookes model in ANSYS FLUENT uses the values listed above, except for $C_{\rm oxid}$ which is set to unity by default.

The closure for the mean soot source terms in the above equations was also described in detail by Brookes and Moss [39]. The uncorrelated closure is the preferred option for a tractable solution of the above transport equations.

Moss et al. [239] have shown the above model applied to kerosene flames by modifying only the soot precursor species (in the original model the precursor was acetylene, whereas for kerosene flames the precursor was assumed to be a combination of both acetylene and benzene) and by setting the value of oxidation scaling parameter C_{oxid} to unity. A good comparison against the experimental measurements for the lower pressure (7 bar) conditions was observed. The predictions of soot formation within methane flames have shown the Brooks and Moss [39] model to be superior compared with the standard Tesner et al. [349] formulation.

The Coal-Derived Soot Extension (Beta Feature)

The present implementation provides an extension to the Moss-Brookes soot model that accounts for coal-derived soot, based on the work of Brown [41]. This extension includes an additional transport equation for the tar evolved during coal devolatilization. The Moss-Brookes model assumes that the physical properties of tar is similar to those of volatiles, such that the combined effect of volatile and tar on the gas phase flame simulation may be replaced by a single volatile stream (consisting of volatile and tar). In reality, however, this may not be the case, and so the coal-derived soot extension allows you to treat the tar contribution similar to that of volatiles.

The following set of paths was assumed for the coal-derived soot formation [214] (see Figure 13.3.1).

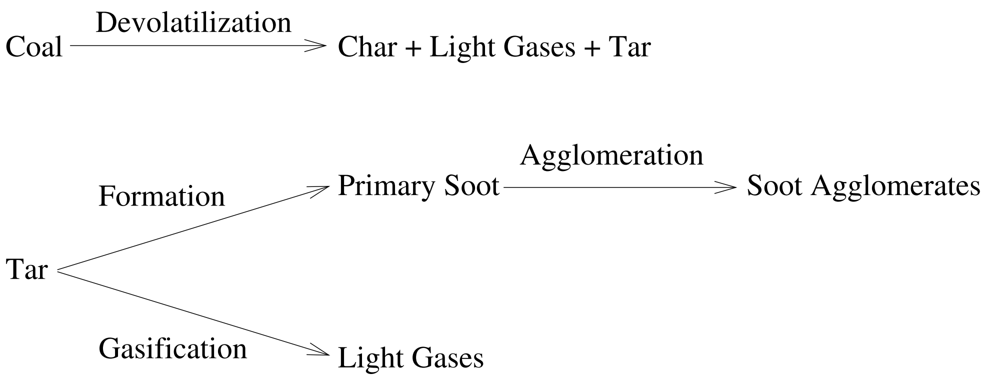


Figure 13.3.1: Presumed Path for Coal-Derived Soot

Nucleation is assumed to be the first step in formation of soot in most light gas flames, and acetylene is understood to be the major species involved. In heavier gas flames, benzene and other polycyclic aromatic hydrocarbons (PAHs) may contribute to soot formation as well. Soot formation in coal flames is thought to occur as tars or the higher molecular weight hydrocarbons given off during devolatilization combine and condense to form soot particles. This is a different mechanism to that of soot formation from gaseous fuels. The related source term for each path is given as follows:

$$S_{\text{soot}} = \text{Formation}_{\text{soot}} - \text{Oxidation}_{\text{soot}} \tag{13.3-18}$$

$$S_{\text{tar}} = \text{Formation}_{\text{tar}} - \text{Formation}_{\text{soot}} - \text{Gasification}_{\text{tar}} - \text{Oxidation}_{\text{tar}} \tag{13.3-19}$$

$$S_{\text{nuclei}} = \frac{1}{N_{\text{norm}}}\left(\frac{N_A}{M_{w,\text{soot}}}\text{Formation}_{\text{soot}} - \text{Agglomeration}_{\text{nuclei}}\right) \tag{13.3-20}$$

where N_{norm} is equal to 10^{15} particles, N_A is Avogadro's Number, and $M_{w,\text{soot}}$ is the molecular weight of the soot particle. The remaining terms in the previous overall source expressions are defined as follows:

$$\text{Formation}_{\text{tar}} = SP_{\text{tar}} \tag{13.3-21}$$

$$\text{Oxidation}_{\text{tar}} = \rho^2 Y_{\text{tar}} Y_{O_2} A_{O_{\text{tar}}} \exp\{-E_{O_{\text{tar}}}/RT\} \tag{13.3-22}$$

$$\text{Gasification}_{\text{tar}} = \rho\, Y_{\text{tar}} A_{G_{\text{tar}}} \exp\{-E_{G_{\text{tar}}}/RT\} \tag{13.3-23}$$

$$\text{Formation}_{\text{soot}} = \rho\, Y_{\text{tar}} A_{F_{\text{soot}}} \exp\{-E_{F_{\text{soot}}}/RT\} \tag{13.3-24}$$

$$\text{Agglomeration}_{\text{nuclei}} = 2C_a\left(\frac{6M_{w,C}}{\pi\rho_{\text{soot}}}\right)^{\frac{1}{6}}\left(\frac{6k_BT}{\rho_{\text{soot}}}\right)^{\frac{1}{2}}\left(\frac{\rho Y_{\text{soot}}}{M_{w,C}}\right)^{\frac{1}{6}}\left(\rho N_{\text{norm}} b^*_{\text{nuc}}\right)^{\frac{11}{6}} \tag{13.3-25}$$

The soot oxidation term ($\text{Oxidation}_{\text{soot}}$) is similar to that shown in Equation 13.3-17 in the Moss-Brookes soot model theory (Fenimore-Jones or Lee oxidation model). Soot density (ρ_{soot}) is assumed to be 1950 kg/m^3 and the collision constant C_a is set to 3.0. $M_{w,C}$ (= 12 kg/kgmol) is the molecular weight of carbon and k_B (= 1.3806503e-23 J/K) is the Boltzmann constant. An incipient soot particle is assumed to consist of 9e+04 carbon atoms, thus making $M_{w,\text{soot}}$ = 108e+04 kg/kgmol. b^*_{nuc} is the normalized radical nuclei concentration (i.e., the number of particles $\times 10^{-15}$/kg). Since the coal-derived soot particles are large, the turbulent Schmidt number used in the transport equations for soot mass fraction and the normalized number density must be modified to account for the particle size. A value of 700 for the turbulent Schmidt number is suggested for soot mass fraction and nuclei transport.

The term SP_{tar} is the tar release rate from coal (kg/m^3-s) and comes from the coal particle source computations of the discrete phase model. It is assumed that the mass fraction of tar in coal volatiles is in the range 0.3–0.5, and therefore the SP_{tar} term is related to the volatile source term via the tar mass fraction in volatiles. One of the main assumptions of this implementation is that tar may be decoupled from the flow field computations, since tar is a fraction of volatiles and volatile transport is fully coupled with the flow field.

The values used for the pre-exponential constant A and the activation energy E in Equations 13.3-22–13.3-24 are listed in Table 13.3.1.

Table 13.3.1: Rate Constants for Coal-Derived Soot

Term	A	E (kJ/kgmol)
$\text{Oxidation}_{\text{tar}}$	6.77e+05 (m^3/kg-s)	52,300
$\text{Gasification}_{\text{tar}}$	9.77e+10 (1/s)	286,900
$\text{Formation}_{\text{soot}}$	5.02e+08 (1/s)	198,900

The Moss-Brookes-Hall Model

Since the Moss-Brookes model was mainly developed and validated for methane flames, a further extension for higher hydrocarbon fuels called the Moss-Brookes-Hall model was also included in the present ANSYS FLUENT implementation. Here, the extended version is a model reported by Wen et al. [374] based on model extensions proposed by Hall et al. [120] and an oxidation model proposed by Lee et al. [184]. The work of Hall [120] is based on a soot inception rate due to two-ringed and three-ringed aromatics, as opposed to the Moss-Brookes assumption of a soot inception due to acetylene or benzene (for higher hydrocarbons).

Hall et al. [120] proposed a soot inception rate based on the formation rates of two-ringed and three-ringed aromatics ($C_{10}H_7$ and $C_{14}H_{10}$), from acetylene (C_2H_2), benzene (C_6H_6), and the phenyl radical (C_6H_5) based on the following mechanisms:

$$2\ \mathrm{C_2H_2} + \mathrm{C_6H_5} \rightleftharpoons \mathrm{C_{10}H_7} + \mathrm{H_2} \qquad (13.3\text{-}26)$$

$$\mathrm{C_2H_2} + \mathrm{C_6H_6} + \mathrm{C_6H_5} \rightleftharpoons \mathrm{C_{14}H_{10}} + \mathrm{H} + \mathrm{H_2} \qquad (13.3\text{-}27)$$

Based on their laminar methane flame data, the inception rate of soot particles was given to be eight times the formation rate of species $C_{10}H_7$ and $C_{14}H_{10}$, as shown by

$$\begin{aligned}\left(\frac{dN}{dt}\right)_{\text{inception}} &= 8C_{\alpha,1}\frac{N_\mathrm{A}}{M_\mathrm{P}}\left[\rho^2\left(\frac{Y_{C_2H_2}}{W_{C_2H_2}}\right)^2\frac{Y_{C_6H_5}W_{H_2}}{W_{C_6H_5}Y_{H_2}}\right]\exp\left\{-\frac{T_{\alpha,1}}{T}\right\} \\ &\quad +8C_{\alpha,2}\frac{N_\mathrm{A}}{M_\mathrm{P}}\left[\rho^2\frac{Y_{C_2H_2}}{W_{C_2H_2}}\frac{Y_{C_6H_6}Y_{C_6H_5}W_{H_2}}{W_{C_6H_6}W_{C_6H_5}Y_{H_2}}\right]\exp\left\{-\frac{T_{\alpha,2}}{T}\right\} \qquad (13.3\text{-}28)\end{aligned}$$

where $C_{\alpha,1} = 127\text{x}10^{8.88}\ \mathrm{s}^{-1}$, $C_{\alpha,2} = 178\text{x}10^{9.50}\ \mathrm{s}^{-1}$, $T_{\alpha,1} = 4378$ K, and $T_{\alpha,2} = 6390$ K as determined by Hall et al. [120]. In their model, the mass of an incipient soot particle was assumed to be 1200 kg/kgmol (corresponding to 100 carbon atoms, as opposed to 12 carbon atoms used by Brookes and Moss [39]). The mass density of soot was assumed to be 2000 kg/m^3, which is also slightly different from the value used by Brookes and Moss. [39]

Both the coagulation term and the surface growth term were formulated similar to those used by Brookes and Moss [39] with a slight modification to the constant C_γ so that the value is 9000.6 kg.m.kmol^{-1}.s^{-1} (based on the model developed by Lindstedt [200]).

For the soot oxidation term, oxidation due to O_2 (based on measurements and model based on Lee et al. [184]) was added, in addition to the soot oxidation due to the hydroxyl radical. By assuming that the kinetics of surface reactions is the limiting mechanism and that the particles are small enough to neglect the diffusion effect on the soot oxidation, they derived the specific rate of soot oxidation by molecular oxygen. Therefore the full soot oxidation term, including that due to hydroxyl radical, is of the form

$$\begin{aligned}\left(\frac{dM}{dt}\right)_{\text{oxidation}} &= -C_{\text{oxid}}C_{\omega,1}\eta_{\text{coll}}\left(\frac{X_{\mathrm{OH}}P}{RT}\right)\sqrt{T}\,(\pi N)^{1/3}\left(\frac{6M}{\rho_{\text{soot}}}\right)^{2/3} \\ &\quad -C_{\text{oxid}}C_{\omega,2}\left(\frac{X_{\mathrm{O_2}}P}{RT}\right)\exp\left\{-\frac{T_{\omega,2}}{T}\right\}\sqrt{T}\,(\pi N)^{1/3}\left(\frac{6M}{\rho_{\text{soot}}}\right)^{2/3} \qquad (13.3\text{-}29)\end{aligned}$$

Here, the collision efficiency is assumed to be 0.13 (compared to 0.04 used by Brookes and Moss) and the oxidation rate scaling parameter is assumed to be unity. The model constants used are as follows:

where

$C_{\omega,1}$ = 105.81 kg.m.kmol^{-1}.K$^{-1/2}$.s^{-1} (same as that used by Brookes and Moss)

$C_{\omega,2}$ = 8903.51 kg.m.kmol^{-1}.K$^{-1/2}$.s^{-1}

$T_{\omega,2}$ = 19778 K

Soot Formation in Turbulent Flows

The kinetic mechanisms of soot formation and destruction for the Moss-Brookes model and the Hall extension are obtained from laboratory experiments in a similar fashion to the NO_x model. In any practical combustion system, however, the flow is highly turbulent. The turbulent mixing process results in temporal fluctuations in temperature and species concentration that will influence the characteristics of the flame.

The relationships among soot formation rate, temperature, and species concentration are highly nonlinear. Hence, if time-averaged composition and temperature are employed in any model to predict the mean soot formation rate, significant errors will result. Temperature and composition fluctuations must be taken into account by considering the probability density functions which describe the time variation.

The Turbulence-Chemistry Interaction Model

In turbulent combustion calculations, ANSYS FLUENT solves the density-weighted time-averaged Navier-Stokes equations for temperature, velocity, and species concentrations or mean mixture fraction and variance. To calculate soot concentration for the Moss-Brookes model and the Hall extension, a time-averaged soot formation rate must be computed at each point in the domain using the averaged flow-field information.

The PDF Approach

The PDF method has proven very useful in the theoretical description of turbulent flow [149]. In the ANSYS FLUENT Moss-Brookes model and the Hall extension, a single- or joint-variable PDF in terms of a normalized temperature, species mass fraction, or the combination of both is used to predict the soot formation. If the non-premixed combustion model is used to model combustion, then a one- or two-variable PDF in terms of mixture fraction(s) is also available. The mean values of the independent variables needed for the PDF construction are obtained from the solution of the transport equations.

The Mean Reaction Rate

The mean turbulent reaction rate described in Section 13.1.9: The General Expression for the Mean Reaction Rate for the NO_x model also applies to the Moss-Brookes model and the Hall extension. The PDF is used for weighting against the instantaneous rates of production of soot and subsequent integration over suitable ranges to obtain the mean turbulent reaction rate as described in Equations 13.1-104 and 13.1-105 for NO_x.

The PDF Options

As is the case with the NO_x model, P can be calculated as either a two-moment beta function or as a clipped Gaussian function, as appropriate for combustion calculations [123, 231]. Equations 13.1-107 – 13.1-111 apply to the Moss-Brookes model and Hall extension as well, with the variance σ^2 computed by solving a transport equation during the combustion calculation stage, using Equation 13.1-112 or Equation 13.1-113.

The Effect of Soot on the Radiation Absorption Coefficient

A description of the modeling of soot-radiation interaction is provided in Section 5.3.8: The Effect of Soot on the Absorption Coefficient.

Chapter 14. Aerodynamically Generated Noise

The discipline of acoustics is intimately related to fluid dynamics. Many sounds that are technologically important in industrial applications are generated by and propagated in fluid flows. The phenomena associated with sounds can therefore be understood and analyzed in the general framework of fluid dynamics. (The governing equations for acoustics are indeed the same as the ones governing fluid flows.)

The main challenge in numerically predicting sound waves stems from the well-recognized fact that sounds have much lower energy than fluid flows, typically by several orders of magnitude. This poses a great challenge to the computation of sounds in terms of difficulty of numerically resolving sound waves, especially when one is interested in predicting sound propagation to the far field. Another challenge comes from the difficulty of predicting the very flow phenomena (e.g., turbulence) in the near field that are responsible for generating sounds.

This chapter provides an overview and theoretical background of ANSYS FLUENT's approaches to computing aerodynamically generated sound. For more information about using the acoustics model, see Chapter 22: Predicting Aerodynamically Generated Noise in the separate User's Guide.

- Section 14.1: Overview
- Section 14.2: Acoustics Model Theory

14.1 Overview

Considering the breadth of the discipline and the challenges encountered in aerodynamically generated noise, it is not surprising that a number of computational approaches have been proposed over the years whose sophistication, applicability, and cost widely vary.

ANSYS FLUENT offers three approaches to computing aerodynamically generated noise; a direct method, an integral method based on acoustic analogy and a method that utilizes broadband noise source models.

This section is divided into the following sections:

- Section 14.1.1: Direct Method
- Section 14.1.2: Integral Method Based on Acoustic Analogy
- Section 14.1.3: Broadband Noise Source Models

14.1.1 Direct Method

In this method, both generation and propagation of sound waves are directly computed by solving the appropriate fluid dynamics equations. Prediction of sound waves always requires time-accurate solutions to the governing equations. Furthermore, in most practical applications of the direct method, one has to employ governing equations that are capable of modeling viscous and turbulence effects, such as unsteady Navier-Stokes equations (i.e., DNS), RANS equations, and filtered equations used in DES and LES.

The direct method is thus computationally difficult and expensive inasmuch as it requires highly accurate numerics, very fine computational meshes all the way to receivers, and acoustically nonreflecting boundary conditions. The computational cost becomes prohibitive when sound is to be predicted in the far field (e.g., hundreds of chord-lengths in the case of an airfoil). The direct method becomes feasible when receivers are in the near field (e.g., cabin noise). In many such situations involving near-field sound, sounds (or pseudo-sounds for that matter) are predominantly due to local hydrodynamic pressure which can be predicted with a reasonable cost and accuracy.

Since sound propagation is directly resolved in this method, one normally needs to solve the compressible form of the governing equations (e.g., compressible RANS equations, compressible form of filtered equations for LES). Only in situations where the flow is low and subsonic, and the receivers in the near field consist primarily of local hydrodynamic pressure fluctuations (i.e., pseudo sound), can incompressible flow formulations be used. However, this incompressible treatment will not permit you to simulate resonance and feedback phenomena.

14.1.2 Integral Method Based on Acoustic Analogy

For predictions of mid- to far-field noise, the methods based on Lighthill's acoustic analogy [195] offer viable alternatives to the direct method. In this approach, the near-field flow obtained from appropriate governing equations such as unsteady RANS equations, DES, or LES are used to predict the sound with the aid of analytically derived integral solutions to wave equations. The acoustic analogy essentially decouples the propagation of sound from its generation, allowing one to separate the flow solution process from the acoustics analysis.

ANSYS FLUENT offers a method based on the Ffowcs Williams and Hawkings (FW-H) formulation [95]. The FW-H formulation adopts the most general form of Lighthill's acoustic analogy, and is capable of predicting sound generated by equivalent acoustic sources such as monopoles, dipoles, and quadrupoles. ANSYS FLUENT adopts a time-domain integral formulation wherein time histories of sound pressure, or acoustic signals, at prescribed receiver locations are directly computed by evaluating a few surface integrals.

Time-accurate solutions of the flow-field variables, such as pressure, velocity components, and density on source (emission) surfaces, are required to evaluate the surface integrals. Time-accurate solutions can be obtained from unsteady Reynolds-averaged Navier-Stokes (URANS) equations, large eddy simulation (LES), or detached eddy simulation (DES) as appropriate for the flow at hand and the features that you want to capture (e.g., vortex shedding). The source surfaces can be placed not only on impermeable walls, but also on interior (permeable) surfaces, which enables you to account for the contributions from the quadrupoles enclosed by the source surfaces. Both broadband and tonal noise can be predicted depending on the nature of the flow (noise source) being considered, turbulence model employed, and the time scale of the flow resolved in the flow calculation.

The FW-H acoustics model in ANSYS FLUENT allows you to select multiple source surfaces and receivers. It also permits you either to save the source data for a future use, or to carry out an "on the fly" acoustic calculation simultaneously as the transient flow calculation proceeds, or both. Sound pressure signals thus obtained can be processed using the fast Fourier transform (FFT) and associated postprocessing capabilities to compute and plot such acoustic quantities as the overall sound pressure level (SPL) and power spectra.

One important limitation of ANSYS FLUENT's FW-H model is that it is applicable only to predicting the propagation of sound toward *free space*. Thus, while the model can be legitimately used to predict far-field noise due to external aerodynamic flows, such as the flows around ground vehicles and aircrafts, it cannot be used for predicting the noise propagation inside ducts or wall-enclosed space.

14.1.3 Broadband Noise Source Models

In many practical applications involving turbulent flows, noise does not have any distinct tones, and the sound energy is continuously distributed over a broad range of frequencies. In those situations involving *broadband noise*, statistical turbulence quantities readily computable from RANS equations can be utilized, in conjunction with semi-empirical correlations and Lighthill's acoustic analogy, to shed some light on the source of broadband noise.

ANSYS FLUENT offers several such *source models* that enable you to quantify the local contribution (per unit surface area or volume) to the total acoustic power generated by the flow. They include the following:

- Proudman's formula
- jet noise source model
- boundary layer noise source model
- source terms in the linearized Euler equations
- source terms in Lilley's equation

Considering that one would ultimately want to come up with some measures to mitigate the noise generated by the flow in question, the source models can be employed to extract useful diagnostics on the noise source to determine which portion of the flow is primarily responsible for the noise generation. Note, however, that these source models do not predict the sound at receivers.

Unlike the direct method and the FW-H integral method, the broadband noise source models do not require transient solutions to any governing fluid dynamics equations. All source models require what typical RANS models would provide, such as the mean velocity field, turbulent kinetic energy (k) and the dissipation rate (ε). Therefore, the use of broadband noise source models requires the least computational resources.

14.2 Acoustics Model Theory

This section describes the theoretical background for the Ffowcs Williams and Hawkings model and the broadband noise source models.

This section is divided into the following sections:

- Section 14.2.1: The Ffowcs Williams and Hawkings Model
- Section 14.2.2: Broadband Noise Source Models

14.2.1 The Ffowcs Williams and Hawkings Model

The Ffowcs Williams and Hawkings (FW-H) equation is essentially an inhomogeneous wave equation that can be derived by manipulating the continuity equation and the Navier-Stokes equations. The FW-H [38, 95] equation can be written as:

$$\begin{aligned}\frac{1}{a_0^2}\frac{\partial^2 p'}{\partial t^2} - \nabla^2 p' &= \frac{\partial^2}{\partial x_i \partial x_j}\left\{T_{ij}H(f)\right\} \\ &- \frac{\partial}{\partial x_i}\left\{\left[P_{ij}n_j + \rho u_i\left(u_n - v_n\right)\right]\delta(f)\right\} \\ &+ \frac{\partial}{\partial t}\left\{\left[\rho_0 v_n + \rho\left(u_n - v_n\right)\right]\delta(f)\right\} \qquad (14.2\text{-}1)\end{aligned}$$

where

u_i	=	fluid velocity component in the x_i direction
u_n	=	fluid velocity component normal to the surface $f = 0$
v_i	=	surface velocity components in the x_i direction
v_n	=	surface velocity component normal to the surface
$\delta(f)$	=	Dirac delta function
$H(f)$	=	Heaviside function

p' is the sound pressure at the far field ($p' = p - p_0$). $f = 0$ denotes a mathematical surface introduced to "embed" the exterior flow problem ($f > 0$) in an unbounded space, which facilitates the use of generalized function theory and the free-space Green function to obtain the solution. The surface ($f = 0$) corresponds to the source (emission) surface, and can be made coincident with a body (impermeable) surface or a permeable surface off the body surface. n_i is the unit normal vector pointing toward the exterior region ($f > 0$), a_0 is the far-field sound speed, and T_{ij} is the Lighthill stress tensor, defined as

$$T_{ij} = \rho u_i u_j + P_{ij} - a_0^2\left(\rho - \rho_0\right)\delta_{ij} \qquad (14.2\text{-}2)$$

P_{ij} is the compressive stress tensor. For a Stokesian fluid, this is given by

$$P_{ij} = p\delta_{ij} - \mu\left[\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3}\frac{\partial u_k}{\partial x_k}\delta_{ij}\right] \qquad (14.2\text{-}3)$$

The free-stream quantities are denoted by the subscript 0.

The solution to Equation 14.2-1 is obtained using the free-space Green function ($\delta(g)/4\pi r$). The complete solution consists of surface integrals and volume integrals. The surface integrals represent the contributions from monopole and dipole acoustic sources and partially from quadrupole sources, whereas the volume integrals represent quadrupole (volume) sources in the region outside the source surface. The contribution of the volume integrals becomes small when the flow is low subsonic and the source surface encloses the source region. In ANSYS FLUENT, the volume integrals are dropped. Thus, we have

$$p'(\vec{x},t) = p'_T(\vec{x},t) + p'_L(\vec{x},t) \tag{14.2-4}$$

where

$$\begin{aligned}
4\pi p'_T(\vec{x},t) &= \int_{f=0} \left[\frac{\rho_0 \left(\dot{U}_n + U_{\dot{n}} \right)}{r\left(1 - M_r\right)^2} \right] dS \\
&+ \int_{f=0} \left[\frac{\rho_0 U_n \left\{ r\dot{M}_r + a_0 \left(M_r - M^2 \right) \right\}}{r^2 \left(1 - M_r\right)^3} \right] dS \qquad (14.2\text{-}5) \\
4\pi p'_L(\vec{x},t) &= \frac{1}{a_0} \int_{f=0} \left[\frac{\dot{L}_r}{r\left(1 - M_r\right)^2} \right] dS \\
&+ \int_{f=0} \left[\frac{L_r - L_M}{r^2\left(1 - M_r\right)^2} \right] dS \\
&+ \frac{1}{a_0} \int_{f=0} \left[\frac{L_r \left\{ r\dot{M}_r + a_0 \left(M_r - M^2 \right) \right\}}{r^2 \left(1 - M_r\right)^3} \right] dS \qquad (14.2\text{-}6)
\end{aligned}$$

where

$$U_i = v_i + \frac{\rho}{\rho_0}\left(u_i - v_i\right) \tag{14.2-7}$$

$$L_i = P_{ij}\hat{n}_j + \rho u_i (u_n - v_n) \tag{14.2-8}$$

When the integration surface coincides with an impenetrable wall, the two terms on the right in Equation 14.2-4, $p'_T(\vec{x},t)$ and $p'_L(\vec{x},t)$, are often referred to as thickness and loading terms, respectively, in light of their physical meanings. The square brackets in Equations 14.2-5 and 14.2-6 denote that the kernels of the integrals are computed at the corresponding retarded times, τ, defined as follows, given the observer time, t, and the distance to the observer, r,

$$\tau = t - \frac{r}{a_0} \tag{14.2-9}$$

The various subscripted quantities appearing in Equations 14.2-5 and 14.2-6 are the inner products of a vector and a unit vector implied by the subscript. For instance, $L_r = \vec{L} \cdot \hat{\vec{r}} = L_i r_i$ and $U_n = \vec{U} \cdot \vec{n} = U_i n_i$, where $\vec{r}$ and $\vec{n}$ denote the unit vectors in the radiation and wall-normal directions, respectively. The dot over a variable denotes source-time differentiation of that variable.

Please note the following remarks regarding the applicability of this integral solution:

- The FW-H formulation in ANSYS FLUENT can handle rotating surfaces as well as stationary surfaces.
- It is not required that the surface $f = 0$ coincide with body surfaces or walls. The formulation permits source surfaces to be permeable, and therefore can be placed in the interior of the flow.
- When a permeable source surface (either interior or nonconformal sliding interface) is placed at a certain distance off the body surface, the integral solutions given by Equations 14.2-5 and 14.2-6 include the contributions from the quadrupole sources within the region enclosed by the source surface. When using a permeable source surface, the mesh resolution needs to be fine enough to resolve the transient flow structures inside the volume enclosed by the permeable surface.

14.2.2 Broadband Noise Source Models

Proudman's Formula

Proudman [280], using Lighthill's acoustic analogy, derived a formula for acoustic power generated by isotropic turbulence without mean flow. More recently, Lilley [196] rederived the formula by accounting for the retarded time difference which was neglected in Proudman's original derivation. Both derivations yield acoustic power due to the unit volume of isotropic turbulence (in W/m^3) as

$$P_A = \alpha \rho_0 \left(\frac{u^3}{\ell} \right) \frac{u^5}{a_0^5} \tag{14.2-10}$$

where u and ℓ are the turbulence velocity and length scales, respectively, and a_0 is the speed of sound. α in Equation 14.2-10 is a model constant. In terms of k and ε, Equation 14.2-10 can be rewritten as

$$P_A = \alpha_\varepsilon \rho_0 \varepsilon M_t^5 \tag{14.2-11}$$

where

$$M_t = \frac{\sqrt{2k}}{a_0} \tag{14.2-12}$$

The rescaled constant, α_ϵ, is set to 0.1 in ANSYS FLUENT based on the calibration of Sarkar and Hussaini [301] using direct numerical simulation of isotropic turbulence.

ANSYS FLUENT can also report the acoustic power in dB, which is computed from

$$L_P = 10 \log \left(\frac{P_A}{P_{\rm ref}} \right) \tag{14.2-13}$$

where $P_{\rm ref}$ is the reference acoustic power ($P_{\rm ref} = 10^{-12} W/m^3$ by default).

The Proudman's formula gives an approximate measure of the local contribution to total acoustic power per unit volume in a given turbulence field. Proper caution, however, should be taken when interpreting the results in view of the assumptions made in the derivation, such as high Reynolds number, small Mach number, isotropy of turbulence, and zero mean motion.

The Jet Noise Source Model

This source model for axisymmetric jets is based on the works of Goldstein [113] who modified the model originally proposed by Ribner [293] to better account for anisotropy of turbulence in axisymmetric turbulent jets.

In Goldstein's model, the total acoustic power emitted by the unit volume of a turbulent jet is computed from

$$\begin{aligned} P_A(\vec{y}) &= \int_0^{2\pi} \int_0^{\pi} I(r,\theta;\vec{y}) r^2 \sin\theta d\theta \, d\psi \\ &= 2\pi r^2 \int_0^{\pi} I(r,\theta;\vec{y}) \sin\theta \, d\theta \end{aligned} \tag{14.2-14}$$

where r and θ are the radial and angular coordinates of the receiver location, and $I(r,\theta;\vec{y})$ is the directional acoustic intensity per unit volume of a jet defined by

$$I(r,\theta;\vec{y}) = \frac{12\,\rho_0\,\omega_f^4\,L_1\,L_2^2\,\overline{u_{t1}^2}^2}{5\,\pi\,a_0^5\,r^2} \frac{D_{\rm self}}{C^5} + \frac{24\,\rho_0\,\omega_f^4\,L_1\,L_2^4\,\overline{u_{t1}^2}}{\pi\,a_0^5\,r^2} \left(\frac{\partial U}{\partial r} \right)^2 \frac{D_{\rm shear}}{C^5} \tag{14.2-15}$$

C in Equation 14.2-15 is the modified convection factor defined by

$$C = 1 - M_c \cos\theta \tag{14.2-16}$$

and

$$\begin{aligned} D_{\rm self} &= 1 + 2(\frac{M}{9} - N) \cos^2\theta \, \sin^2\theta \\ &+ \frac{1}{3} \left[\frac{M^2}{7} + M - 1.5N(3 - 3N + \frac{1.5}{\Delta^2} - \frac{\Delta^2}{2}) \right] \sin^4\theta \end{aligned} \tag{14.2-17}$$

$$D_{\rm shear} = \cos^2\theta \left[\cos^2\theta + \frac{1}{2} \left(\frac{1}{\Delta^2} - 2N \right) \sin^2\theta \right] \tag{14.2-18}$$

The remaining parameters are defined as

$$\Delta = \frac{L_2}{L_1} \tag{14.2-19}$$

$$M = \left[\frac{3}{2}\left(\Delta - \frac{1}{\Delta}\right)\right]^2 \tag{14.2-20}$$

$$N = 1 - \frac{\left(\overline{u_{t2}^2}\right)}{\left(\overline{u_{t1}^2}\right)} \tag{14.2-21}$$

$$L_1 = \frac{\left(\overline{u_{t1}^2}\right)^{3/2}}{\epsilon} \tag{14.2-22}$$

$$L_2 = \frac{\left(\overline{u_{t2}^2}\right)^{3/2}}{\epsilon} \tag{14.2-23}$$

$$\omega_f = 2\pi\frac{\epsilon}{k} \tag{14.2-24}$$

where $\overline{u_{t1}^2}$ and $\overline{u_{t2}^2}$ are computed differently depending on the turbulence model chosen for the computation. When the RSM is selected, they are computed from the corresponding normal stresses. For all other two-equation turbulence models, they are obtained from

$$\overline{u_{t1}^2} = \frac{8}{9}k \tag{14.2-25}$$

$$\overline{u_{t2}^2} = \frac{4}{9}k \tag{14.2-26}$$

ANSYS FLUENT reports the acoustic power both in the dimensional units (W/m^3) and in dB computed from

$$L_P = 10 \log\left(\frac{P_A}{P_{\text{ref}}}\right) \tag{14.2-27}$$

where P_{ref} is the reference acoustic power ($P_{\text{ref}} = 10^{-12} W/m^3$ by default).

The Boundary Layer Noise Source Model

Far-field sound generated by turbulent boundary layer flow over a solid body at low Mach numbers is often of practical interest. The Curle's integral [64] based on acoustic analogy can be used to approximate the local contribution from the body surface to the total acoustic power. To that end, one can start with the Curle's integral

$$p'(\vec{x},t) = \frac{1}{4\pi a_0}\int_S \frac{(x_i - y_i)\, n_i}{r^2}\frac{\partial p}{\partial t}(\vec{y},\tau)\, dS(\vec{y}) \tag{14.2-28}$$

where τ denotes the emission time ($\tau = t - r/a_0$), and S the integration surface.

Using this, the sound intensity in the far field can then be approximated by

$$\overline{p'^2} \approx \frac{1}{16\pi^2 a_0^2} \int_S \frac{\cos^2\theta}{r^2} \overline{\left[\frac{\partial p}{\partial t}(\vec{y}, \tau)\right]^2} \mathcal{A}_c(\vec{y})\, dS(\vec{y}) \tag{14.2-29}$$

where $\mathcal{A}_c$ is the correlation area, $r \equiv |\vec{x} - \vec{y}|$, and $\cos\theta$ is the angle between $|\vec{x} - \vec{y}|$ and the wall-normal direction $\vec{n}$.

The total acoustic power emitted from the entire body surface can be computed from

$$\begin{aligned} P_A &= \frac{1}{\rho_0 a_0} \int_0^{2\pi} \int_0^{\pi} \overline{p'^2} r^2 \sin\theta \, d\theta d\psi \\ &= \int_S I(\vec{y})\, dS(\vec{y}) \end{aligned} \tag{14.2-30}$$

where

$$I(\vec{y}) \equiv \frac{\mathcal{A}_c(\vec{y})}{12\rho_0 \pi a_0^3} \overline{\left[\frac{\partial p}{\partial t}\right]^2} \tag{14.2-31}$$

which can be interpreted as the local contribution per unit surface area of the body surface to the total acoustic power. The mean-square time derivative of the surface pressure and the correlation area are further approximated in terms of turbulent quantities like turbulent kinetic energy, dissipation rate, and wall shear.

ANSYS FLUENT reports the acoustic surface power defined by Equation 14.2-31 both in physical (W/m^2) and dB units.

Source Terms in the Linearized Euler Equations

The linearized Euler equations (LEE) can be derived from the Navier-Stokes equations starting from decompositions of the flow variables into mean, turbulent, and acoustic components, and by assuming that the acoustic components are much smaller than the mean and turbulent components. The resulting linearized Euler equations for the acoustic velocity components can be written as

$$\begin{gathered} \frac{\partial u_{ai}}{\partial t} + U_j \frac{\partial u_{ai}}{\partial x_j} + u_{aj} \frac{\partial U_i}{\partial x_j} + \frac{1}{\overline{\rho}} \frac{\partial p_a}{\partial x_i} - \frac{\rho_a}{\overline{\rho}^2} \frac{\partial P}{\partial x_i} = \\ \underbrace{-U_j \frac{\partial u'_i}{\partial x_j} - u'_j \frac{\partial U_i}{\partial x_j}}_{L_{sh}} \underbrace{- u'_j \frac{\partial u'_i}{\partial x_j}}_{L_{se}} - \frac{1}{\overline{\rho}} \frac{\partial p'}{\partial x_i} - \frac{\partial u'_i}{\partial t} + \frac{\partial}{\partial x_j} \overline{u'_j u'_i} \end{gathered} \tag{14.2-32}$$

where the subscript "a" refers to the corresponding acoustic components, and the prime superscript refers to the turbulent components.

The right side of Equation 14.2-32 can be considered as effective source terms responsible for sound generation. Among them, the first three terms involving turbulence are the main contributors. The first two terms denoted by L_{sh} are often referred to as "shear-noise" source terms, since they involve the mean shear. The third term denoted by L_{se} is often called the "self-noise" source term, as it involves turbulent velocity components only.

The turbulent velocity field needed to compute the LEE source terms is obtained using the method of stochastic noise generation and radiation (SNGR) [23]. In this method, the turbulent velocity field and its derivatives are computed from a sum of N Fourier modes.

$$\vec{u}\left(\vec{x},t\right) = 2\sum_{n=1}^{N} \tilde{u}_n \cos\left(\vec{k}_n \cdot \vec{x} + \psi_n\right)\vec{\sigma}_n \tag{14.2-33}$$

where $\tilde{u}_n$, ψ_n, $\vec{\sigma}_n$ are the amplitude, phase, and directional (unit) vector of the n^{th} Fourier mode associated with the wave-number vector $\vec{k}_n$.

Note that the source terms in the LEE are vector quantities, having two or three components depending on the dimension of the problem at hand.

Source Terms in Lilley's Equation

Lilley's equation is a third-order wave equation that can be derived by combining the conservation of mass and momentum of compressible fluids. When the viscous terms are omitted, it can be written in the following form:

$$\frac{D}{Dt}\left[\frac{D^2\Pi}{Dt^2} - \frac{\partial}{\partial x_j}\left(a^2\frac{\partial\Pi}{\partial x_j}\right)\right] + 2\frac{\partial u_k}{\partial x_j}\frac{\partial}{\partial x_k}\left(a^2\frac{\partial\Pi}{\partial x_j}\right) = -2\frac{\partial u_k}{\partial x_i}\frac{\partial u_j}{\partial x_k}\frac{\partial u_i}{\partial x_j} \tag{14.2-34}$$

where $\Pi = (1/\gamma)\ln\frac{p}{p_o}$.

Lilley's equation can be linearized about the underlying steady flow as

$$u_i(\vec{x},t) = U_i(\vec{x}) + u'_i(\vec{x},t) \tag{14.2-35}$$

where $u'(\vec{x},t)$ is the turbulent velocity component.

Substituting Equation 14.2-35 into the source term of Equation 14.2-34, we have

$$\begin{aligned} S &\equiv -2\frac{\partial u_k}{\partial x_i}\frac{\partial u_j}{\partial x_k}\frac{\partial u_i}{\partial x_j} \\ &= -2\frac{\partial U_k}{\partial x_i}\frac{\partial U_j}{\partial x_k}\frac{\partial U_i}{\partial x_j} \underbrace{-2\frac{\partial u'_k}{\partial x_i}\frac{\partial u'_j}{\partial x_k}\frac{\partial u'_i}{\partial x_j}}_{\text{Self-Noise Terms}} \underbrace{-6\frac{\partial U_k}{\partial x_i}\frac{\partial U_j}{\partial x_k}\frac{\partial u'_i}{\partial x_j} - 6\frac{\partial u'_k}{\partial x_i}\frac{\partial u'_j}{\partial x_k}\frac{\partial U_i}{\partial x_j}}_{\text{Shear-Noise Terms}} \end{aligned} \tag{14.2-36}$$

The resulting source terms in Equation 14.2-36 are evaluated using the mean velocity field and the turbulent (fluctuating) velocity components synthesized by the SNGR method. As with the LEE source terms, the source terms in Equation 14.2-36 are grouped depending on whether the mean velocity gradients are involved (*shear noise* or *self noise*), and reported separately in ANSYS FLUENT.

Chapter 15. Discrete Phase

This chapter describes the theory behind the Lagrangian discrete phase capabilities available in ANSYS FLUENT. For information about how to use discrete phase models, see Chapter 23: Modeling Discrete Phase in the separate User's Guide. Information in this chapter is organized into the following sections:

15.1 Introduction

Advances in computational fluid mechanics have provided the basis for further insight into the dynamics of multiphase flows. Currently there are two approaches for the numerical calculation of multiphase flows: the Euler-Lagrange approach (discussed below) and the Euler-Euler approach (discussed in Section 16.2.1: Approaches to Multiphase Modeling).

The Euler-Lagrange Approach

The Lagrangian discrete phase model in ANSYS FLUENT (described in this chapter) follows the Euler-Lagrange approach. The fluid phase is treated as a continuum by solving the Navier-Stokes equations, while the dispersed phase is solved by tracking a large number of particles, bubbles, or droplets through the calculated flow field. The dispersed phase can exchange momentum, mass, and energy with the fluid phase.

A fundamental assumption made in this model is that the dispersed second phase occupies a low volume fraction, even though high mass loading ($\dot{m}_{\text{particles}} \geq \dot{m}_{\text{fluid}}$) is acceptable. The particle or droplet trajectories are computed individually at specified intervals during the fluid phase calculation. This makes the model appropriate for the modeling of spray dryers, coal and liquid fuel combustion, and some particle-laden flows, but inappropriate for the modeling of liquid-liquid mixtures, fluidized beds, or any application where the volume fraction of the second phase cannot be neglected.

Limitations of the discrete phase model are listed in Section 23.1.2: Limitations in the separate User's Guide.

15.2 Particle Motion Theory

This section is composed of the following:

- Section 15.2.1: Equations of Motion for Particles
- Section 15.2.2: Turbulent Dispersion of Particles
- Section 15.2.3: Integration of Particle Equation of Motion

15.2.1 Equations of Motion for Particles

Particle Force Balance

ANSYS FLUENT predicts the trajectory of a discrete phase particle (or droplet or bubble) by integrating the force balance on the particle, which is written in a Lagrangian reference frame. This force balance equates the particle inertia with the forces acting on the particle, and can be written (for the x direction in Cartesian coordinates) as

$$\frac{du_p}{dt} = F_D(u - u_p) + \frac{g_x(\rho_p - \rho)}{\rho_p} + F_x \tag{15.2-1}$$

where F_x is an additional acceleration (force/unit particle mass) term, $F_D(u - u_p)$ is the drag force per unit particle mass and

$$F_D = \frac{18\mu}{\rho_p d_p^2} \frac{C_D \text{Re}}{24} \tag{15.2-2}$$

Here, u is the fluid phase velocity, u_p is the particle velocity, μ is the molecular viscosity of the fluid, ρ is the fluid density, ρ_p is the density of the particle, and d_p is the particle diameter. Re is the relative Reynolds number, which is defined as

$$\text{Re} \equiv \frac{\rho d_p \left| u_p - u \right|}{\mu} \tag{15.2-3}$$

Inclusion of the Gravity Term

While Equation 15.2-1 includes a force of gravity on the particle, it is important to note that in ANSYS FLUENT the default gravitational acceleration is zero. If you want to include the gravitational force, you must remember to define the magnitude and direction of the gravity vector in the **Operating Conditions** dialog box.

Other Forces

Equation 15.2-1 incorporates additional forces (F_x) in the particle force balance that can be important under special circumstances. The first of these is the “virtual mass” force, the force required to accelerate the fluid surrounding the particle. This force can be written as

$$F_x = \frac{1}{2} \frac{\rho}{\rho_p} \frac{d}{dt} (u - u_p) \tag{15.2-4}$$

and is important when $\rho > \rho_p$. An additional force arises due to the pressure gradient in the fluid:

$$F_x = \left(\frac{\rho}{\rho_p} \right) u_{p_i} \frac{\partial u}{\partial x_i} \tag{15.2-5}$$

Forces in Rotating Reference Frames

The additional force term, F_x, in Equation 15.2-1 also includes forces on particles that arise due to rotation of the reference frame. These forces arise when you are modeling flows in rotating frames of reference (see Section 2.2: Flow in a Rotating Reference Frame). For rotation defined about the z axis, for example, the forces on the particles in the Cartesian x and y directions can be written as

$$\left(1 - \frac{\rho}{\rho_p} \right) \Omega^2 x + 2\Omega \left(u_{y,p} - \frac{\rho}{\rho_p} u_y \right) \tag{15.2-6}$$

where $u_{y,p}$ and u_y are the particle and fluid velocities in the Cartesian y direction, and

$$\left(1-\frac{\rho}{\rho_p}\right)\Omega^2 y - 2\Omega\left(u_{x,p}-\frac{\rho}{\rho_p}u_x\right) \tag{15.2-7}$$

where $u_{x,p}$ and u_x are the particle and fluid velocities in the Cartesian x direction.

Thermophoretic Force

Small particles suspended in a gas that has a temperature gradient experience a force in the direction opposite to that of the gradient. This phenomenon is known as thermophoresis. ANSYS FLUENT can optionally include a thermophoretic effect on particles in the additional acceleration (force/unit mass) term, F_x, in Equation 15.2-1:

$$F_x = -D_{T,p}\frac{1}{m_p T}\frac{\partial T}{\partial x} \tag{15.2-8}$$

where $D_{T,p}$ is the thermophoretic coefficient. You can define the coefficient to be constant, polynomial, or a user-defined function, or you can use the form suggested by Talbot [345]:

$$F_x = -\frac{6\pi d_p \mu^2 C_s (K + C_t \mathrm{Kn})}{\rho(1+3C_m\mathrm{Kn})(1+2K+2C_t\mathrm{Kn})}\frac{1}{m_p T}\frac{\partial T}{\partial x} \tag{15.2-9}$$

where:	Kn	= Knudsen number = 2 λ/d_p
	λ	= mean free path of the fluid
	K	= k/k_p
	k	= fluid thermal conductivity based on translational energy only = (15/4) μR
	k_p	= particle thermal conductivity
	C_S	= 1.17
	C_t	= 2.18
	C_m	= 1.14
	m_p	= particle mass
	T	= local fluid temperature
	μ	= fluid viscosity

This expression assumes that the particle is a sphere and that the fluid is an ideal gas.

Brownian Force

For sub-micron particles, the effects of Brownian motion can be optionally included in the additional force term. The components of the Brownian force are modeled as a Gaussian white noise process with spectral intensity $S_{n,ij}$ given by [191]

$$S_{n,ij} = S_0 \delta_{ij} \tag{15.2-10}$$

where δ_{ij} is the Kronecker delta function, and

$$S_0 = \frac{216 \nu k_B T}{\pi^2 \rho d_p^5 \left(\frac{\rho_p}{\rho}\right)^2 C_c} \tag{15.2-11}$$

T is the absolute temperature of the fluid, ν is the kinematic viscosity, and k_B is the Boltzmann constant. Amplitudes of the Brownian force components are of the form

$$F_{b_i} = \zeta_i \sqrt{\frac{\pi S_o}{\Delta t}} \tag{15.2-12}$$

where ζ_i are zero-mean, unit-variance-independent Gaussian random numbers. The amplitudes of the Brownian force components are evaluated at each time step. The energy equation must be enabled in order for the Brownian force to take effect. Brownian force is intended only for laminar simulations.

Saffman's Lift Force

The Saffman's lift force, or lift due to shear, can also be included in the additional force term as an option. The lift force used is from Li and Ahmadi [191] and is a generalization of the expression provided by Saffman [298]:

$$\vec{F} = \frac{2K\nu^{1/2}\rho d_{ij}}{\rho_p d_p (d_{lk} d_{kl})^{1/4}} (\vec{v} - \vec{v}_p) \tag{15.2-13}$$

where $K = 2.594$ and d_{ij} is the deformation tensor. This form of the lift force is intended for small particle Reynolds numbers. Also, the particle Reynolds number based on the particle-fluid velocity difference must be smaller than the square root of the particle Reynolds number based on the shear field. This option is recommended only for submicron particles.

15.2.2 Turbulent Dispersion of Particles

The dispersion of particles due to turbulence in the fluid phase can be predicted using the stochastic tracking model or the particle cloud model (see Section 15.2.2: Turbulent Dispersion of Particles). The stochastic tracking (random walk) model includes the effect of instantaneous turbulent velocity fluctuations on the particle trajectories through the use of stochastic methods (see Section 15.2.2: Stochastic Tracking). The particle cloud model tracks the statistical evolution of a cloud of particles about a mean trajectory (see Section 15.2.2: Particle Cloud Tracking). The concentration of particles within the cloud is represented by a Gaussian probability density function (PDF) about the mean trajectory. For stochastic tracking a model is available to account for the generation or dissipation of turbulence in the continuous phase (see Section 15.12.1: Coupling Between the Discrete and Continuous Phases).

i Turbulent dispersion of particles cannot be included if the Spalart-Allmaras turbulence model is used.

Stochastic Tracking

When the flow is turbulent, ANSYS FLUENT will predict the trajectories of particles using the mean fluid phase velocity, $\overline{u}$, in the trajectory equations (Equation 15.2-1). Optionally, you can include the instantaneous value of the fluctuating gas flow velocity,

$$u = \overline{u} + u' \tag{15.2-14}$$

to predict the dispersion of the particles due to turbulence.

In the stochastic tracking approach, ANSYS FLUENT predicts the turbulent dispersion of particles by integrating the trajectory equations for individual particles, using the instantaneous fluid velocity, $\overline{u} + u^{'}(t)$, along the particle path during the integration. By computing the trajectory in this manner for a sufficient number of representative particles (termed the "number of tries"), the random effects of turbulence on the particle dispersion can be included.

ANSYS FLUENT uses a stochastic method (random walk model) to determine the instantaneous gas velocity. In the discrete random walk (DRW) model, the fluctuating velocity components are discrete piecewise constant functions of time. Their random value is kept constant over an interval of time given by the characteristic lifetime of the eddies.

The DRW model may give nonphysical results in strongly nonhomogeneous diffusion-dominated flows, where small particles should become uniformly distributed. Instead, the DRW will show a tendency for such particles to concentrate in low-turbulence regions of the flow.

The Integral Time

Prediction of particle dispersion makes use of the concept of the integral time scale, T, which describes the time spent in turbulent motion along the particle path, ds:

$$T = \int_0^\infty \frac{u_p{}'(t) u_p{}'(t+s)}{\overline{u_p{}'^2}} ds \tag{15.2-15}$$

The integral time is proportional to the particle dispersion rate, as larger values indicate more turbulent motion in the flow. It can be shown that the particle diffusivity is given by $\overline{u_i{}' u_j{}'} T$.

For small "tracer" particles that move with the fluid (zero drift velocity), the integral time becomes the fluid Lagrangian integral time, T_L. This time scale can be approximated as

$$T_L = C_L \frac{k}{\epsilon} \tag{15.2-16}$$

where C_L is to be determined as it is not well known. By matching the diffusivity of tracer particles, $\overline{u_i{}' u_j{}'} T_L$, to the scalar diffusion rate predicted by the turbulence model, ν_t/σ, one can obtain

$$T_L \approx 0.15 \frac{k}{\epsilon} \tag{15.2-17}$$

for the k-ϵ model and its variants, and

$$T_L \approx 0.30 \frac{k}{\epsilon} \tag{15.2-18}$$

when the Reynolds stress model (RSM) is used [67]. For the k-ω models, substitute $\omega = \epsilon/k$ into Equation 15.2-16. The LES model uses the equivalent LES time scales.

The Discrete Random Walk Model

In the discrete random walk (DRW) model, or "eddy lifetime" model, the interaction of a particle with a succession of discrete stylized fluid phase turbulent eddies is simulated. Each eddy is characterized by

- a Gaussian distributed random velocity fluctuation, u', v', and w'
- a time scale, τ_e

The values of u', v', and w' that prevail during the lifetime of the turbulent eddy are sampled by assuming that they obey a Gaussian probability distribution, so that

$$u^{'} = \zeta\sqrt{\overline{u'^2}} \tag{15.2-19}$$

where ζ is a normally distributed random number, and the remainder of the right-hand side is the local RMS value of the velocity fluctuations. Since the kinetic energy of turbulence is known at each point in the flow, these values of the RMS fluctuating components can be defined (assuming isotropy) as

$$\sqrt{\overline{u'^2}} = \sqrt{\overline{v'^2}} = \sqrt{\overline{w'^2}} = \sqrt{2k/3} \tag{15.2-20}$$

for the k-ϵ model, the k-ω model, and their variants. When the RSM is used, nonisotropy of the stresses is included in the derivation of the velocity fluctuations:

$$\begin{aligned} u' &= \zeta\sqrt{\overline{u'^2}} && (15.2\text{-}21) \\ v' &= \zeta\sqrt{\overline{v'^2}} && (15.2\text{-}22) \\ w' &= \zeta\sqrt{\overline{w'^2}} && (15.2\text{-}23) \end{aligned}$$

when viewed in a reference frame in which the second moment of the turbulence is diagonal [389]. For the LES model, the velocity fluctuations are equivalent in all directions. See Section 4.11.4: Inlet Boundary Conditions for the LES Model for details.

The characteristic lifetime of the eddy is defined either as a constant:

$$\tau_e = 2T_L \tag{15.2-24}$$

where T_L is given by Equation 15.2-16 in general (Equation 15.2-17 by default), or as a random variation about T_L:

$$\tau_e = -T_L \ln(r) \tag{15.2-25}$$

where r is a uniform random number between 0 and 1 and T_L is given by Equation 15.2-17. The option of random calculation of τ_e yields a more realistic description of the correlation function.

The particle eddy crossing time is defined as

$$t_{\rm cross} = -\tau \ln\left[1 - \left(\frac{L_e}{\tau|u - u_p|}\right)\right] \tag{15.2-26}$$

where τ is the particle relaxation time, L_e is the eddy length scale, and $|u - u_p|$ is the magnitude of the relative velocity.

The particle is assumed to interact with the fluid phase eddy over the smaller of the eddy lifetime and the eddy crossing time. When this time is reached, a new value of the instantaneous velocity is obtained by applying a new value of ζ in Equation 15.2-19.

Using the DRW Model

The only inputs required for the DRW model are the value for the integral time-scale constant, C_L (see Equations 15.2-16 and 15.2-24) and the choice of the method used for the prediction of the eddy lifetime. You can choose to use either a constant value or a random value by selecting the appropriate option in the **Set Injection Properties** dialog box for each injection, as described in Section 23.3.16: Stochastic Tracking in the separate User's Guide.

i Turbulent dispersion of particles cannot be included if the Spalart-Allmaras turbulence model is used.

Particle Cloud Tracking

Particle dispersion due to turbulent fluctuations can also be modeled with the particle cloud model [21, 22, 144, 204]. The turbulent dispersion of particles about a mean trajectory is calculated using statistical methods. The concentration of particles about the mean trajectory is represented by a Gaussian probability density function (PDF) whose variance is based on the degree of particle dispersion due to turbulent fluctuations. The mean trajectory is obtained by solving the ensemble-averaged equations of motion for all particles represented by the cloud (see Section 15.2.2: Particle Cloud Tracking).

The cloud enters the domain either as a point source or with an initial diameter. The cloud expands due to turbulent dispersion as it is transported through the domain until it exits. As mentioned before, the distribution of particles in the cloud is defined by a probability density function (PDF) based on the position in the cloud relative to the cloud center. The value of the PDF represents the probability of finding particles represented by that cloud with residence time t at location x_i in the flow field. The average particle number density can be obtained by weighting the total flow rate of particles represented by that cloud, $\dot{m}$, as

$$\langle n(x_i) \rangle = \dot{m} P(x_i, t) \tag{15.2-27}$$

The PDFs for particle position are assumed to be multivariate Gaussian. These are completely described by their mean, μ_i, and variance, ${\sigma_i}^2$, and are of the form

$$P(x_i, t) = \frac{1}{(2\pi)^{3/2} \prod_{i=1}^{3} \sigma_i} e^{-s/2} \tag{15.2-28}$$

where

$$s = \sum_{i=1}^{3} \left(\frac{x_i - \mu_i}{\sigma_i} \right)^2 \tag{15.2-29}$$

The mean of the PDF, or the center of the cloud, at a given time represents the most likely location of the particles in the cloud. The mean location is obtained by integrating a particle velocity as defined by an equation of motion for the cloud of particles:

$$\mu_i(t) \equiv \langle x_i(t) \rangle = \int_0^t \langle V_i(t_1) \rangle dt_1 + \langle x_i(0) \rangle \tag{15.2-30}$$

The equations of motion are constructed using an ensemble average.

The radius of the particle cloud is based on the variance of the PDF. The variance, $\sigma_i^2(t)$, of the PDF can be expressed in terms of two particle turbulence statistical quantities:

$$\sigma_i^2(t) = 2 \int_0^t \langle u'^2_{p,i}(t_2) \rangle \int_0^{t_2} R_{p,ii}(t_2, t_1) dt_1 dt_2 \tag{15.2-31}$$

where $\langle u'^2_{p,i} \rangle$ are the mean square velocity fluctuations, and $R_{p,ij}(t_2, t_1)$ is the particle velocity correlation function:

$$R_{p,ij}(t_2, t_1) = \frac{\langle u'_{p,i}(t_2) u'_{p,j}(t_1) \rangle}{\left[\langle u'^2_{p,i}(t_2) u'^2_{p,j}(t_2) \rangle \right]^{1/2}} \tag{15.2-32}$$

By using the substitution $\tau = |t_2 - t_1|$, and the fact that

$$R_{p,ij}(t_2, t_1) = R_{p,ij}(t_4, t_3) \tag{15.2-33}$$

whenever $|t_2 - t_1| = |t_4 - t_3|$, we can write

$$\sigma_i^2(t) = 2 \int_0^t \langle u'^2_{p,i}(t_2) \rangle \int_0^{t_2} R_{p,ii}(\tau) d\tau dt_2 \tag{15.2-34}$$

Note that cross correlations in the definition of the variance ($R_{p,ij}, i \neq j$) have been neglected.

The form of the particle velocity correlation function used determines the particle dispersion in the cloud model. ANSYS FLUENT uses a correlation function first proposed by Wang [365], and used by Jain [144]. When the gravity vector is aligned with the z-coordinate direction, R_{ij} takes the form:

$$
\begin{aligned}
R_{p,11} &= \frac{u'^2}{\theta} e^{-(\tau/\tau_a)} \ \mathrm{St}_T \left(B - 0.5 m_T \gamma \frac{\mathrm{St}_T^2 B^2 + 1}{\theta} \right) \\
&+ \frac{u'^2}{\theta} e^{-(\tau B/T)} \left(-1 + \frac{m_T \mathrm{St}_T^2 \gamma B}{\theta} + 0.5 m_T \gamma \frac{\tau}{T} \right)
\end{aligned}
\tag{15.2-35}
$$

$$R_{p,22} = R_{p,11} \tag{15.2-36}$$

$$R_{p,33} = \frac{u'^2 \mathrm{St}_T B}{\theta} e^{-(\tau/\tau_a)} - \frac{u'^2}{\theta} e^{-(\tau B/T)} \tag{15.2-37}$$

where $B = \sqrt{1 + m_T^2 \gamma^2}$ and τ_a is the aerodynamic response time of the particle:

$$\tau_a = \frac{\rho_p d_p^2}{18\mu} \tag{15.2-38}$$

and

$$T = \frac{m_T T_{mE}}{m} \tag{15.2-39}$$

$$T_{fE} = \frac{C_\mu^{3/4} k^{3/2}}{\epsilon(\frac{2}{3}k)^{1/2}} \tag{15.2-40}$$

$$\gamma = \frac{\tau_a g}{u'} \tag{15.2-41}$$

$$\mathrm{St} = \frac{\tau_a}{T_{mE}} \tag{15.2-42}$$

$$\mathrm{St}_T = \frac{\tau_a}{T} \tag{15.2-43}$$

$$\theta = \mathrm{St}_T^2 (1 + m_T^2 \gamma^2) - 1 \tag{15.2-44}$$

$$m = \frac{\bar{u}}{u'} \tag{15.2-45}$$

$$T_{mE} = T_{fE} \frac{\bar{u}}{u'} \tag{15.2-46}$$

$$m_T = m \left[1 - \frac{G(m)}{(1 + \mathrm{St})^{0.4(1+0.01\mathrm{St})}} \right] \tag{15.2-47}$$

$$G(m) = \frac{2}{\sqrt{\pi}} \int_0^\infty \frac{e^{-y^2} \, dy}{\left(1 + \frac{m^2}{\pi} \left(\sqrt{\pi} \ \mathrm{erf}(y) y - 1 + e^{-y^2}\right)\right)^{5/2}} \tag{15.2-48}$$

Using this correlation function, the variance is integrated over the life of the cloud. At any given time, the cloud radius is set to three standard deviations in the coordinate directions. The cloud radius is limited to three standard deviations since at least 99.2% of the area under a Gaussian PDF is accounted for at this distance. Once the cells within the cloud are established, the fluid properties are ensemble-averaged for the mean trajectory, and the mean path is integrated in time. This is done with a weighting factor defined as

$$W(x_i,t) \equiv \frac{\int_{V_{\rm cell}} P(x_i,t)dV}{\int_{V_{\rm cloud}} P(x_i,t)dV} \tag{15.2-49}$$

If coupled calculations are performed, sources are distributed to the cells in the cloud based on the same weighting factors.

Using the Cloud Model

The only inputs required for the cloud model are the values of the minimum and maximum cloud diameters. The cloud model is enabled in the **Set Injection Properties** dialog box for each injection, as described in Section 23.3.16: Cloud Tracking in the separate User's Guide.

i The cloud model is not available for unsteady particle tracking, or in parallel, when using the message passing option for the particles.

15.2.3 Integration of Particle Equation of Motion

The trajectory equations, and any auxiliary equations describing heat or mass transfer to/from the particle, are solved by stepwise integration over discrete time steps. Integration of time in Equation 15.2-1 yields the velocity of the particle at each point along the trajectory, with the trajectory itself predicted by

$$\frac{dx}{dt} = u_p \tag{15.2-50}$$

Note that Equation 15.2-1 and Equation 15.2-50 are a set of coupled ordinary differential equations, and Equation 15.2-1 can be cast into the following general form

$$\frac{du_p}{dt} = \frac{1}{\tau_p}(u - u_p) + a \tag{15.2-51}$$

where the term a includes accelerations due to all other forces except drag force.

This set can be solved for constant u, a and τ_p by analytical integration. For the particle velocity at the new location u_p^{n+1} we get

$$u_p^{n+1} = u^n + e^{-\frac{\Delta t}{\tau_p}} \left(u_p^n - u^n \right) - a\tau_p \left(e^{-\frac{\Delta t}{\tau_p}} - 1 \right) \qquad (15.2\text{-}52)$$

The new location x_p^{n+1} can be computed from a similar relationship.

$$x_p^{n+1} = x_p^n + \Delta t(u^n + a\tau_p) + \tau_p \left(1 - e^{-\frac{\Delta t}{\tau_p}} \right) \left(u_p^n - u^n - a\tau_p \right) \qquad (15.2\text{-}53)$$

In these equations u_p^n and u^n represent particle velocities and fluid velocities at the old location. Equations 15.2-52 and 15.2-53 are applied when using the analytic discretization scheme.

The set of Equation 15.2-1 and Equation 15.2-50 can also be solved using numerical discretization schemes. When applying the Euler implicit discretization to Equation 15.2-51 we get

$$u_p^{n+1} = \frac{u_p^n + \Delta t(a + \frac{u^n}{\tau_p})}{1 + \frac{\Delta t}{\tau_p}} \qquad (15.2\text{-}54)$$

When applying a trapezoidal discretization to Equation 15.2-51 the variables u_p and u_n on the right hand side are taken as averages, while accelerations, a, due to other forces are held constant. We get

$$\frac{u_p^{n+1} - u_p^n}{\Delta t} = \frac{1}{\tau_p}(u^* - u_p^*) + a^n \qquad (15.2\text{-}55)$$

The averages u_p^* and u^* are computed from

$$\begin{aligned} u_p^* &= \frac{1}{2}(u_p^n + u_p^{n+1}) && (15.2\text{-}56) \\ u^* &= \frac{1}{2}(u^n + u^{n+1}) && (15.2\text{-}57) \\ u^{n+1} &= u^n + \Delta t u_p^n \cdot \nabla u^n && (15.2\text{-}58) \end{aligned}$$

The particle velocity at the new location $n+1$ is computed by

$$u_p^{n+1} = \frac{u_p^n(1 - \frac{1}{2}\frac{\Delta t}{\tau_p}) + \frac{\Delta t}{\tau_p} \left(u^n + \frac{1}{2}\Delta t u_p^n \cdot \nabla u^n \right) + \Delta t a}{1 + \frac{1}{2}\frac{\Delta t}{\tau_p}} \qquad (15.2\text{-}59)$$

For the implicit and the trapezoidal schemes the new particle location is always computed by a trapezoidal discretization of Equation 15.2-50.

$$x_p^{n+1} = x_p^n + \frac{1}{2}\Delta t \left(u_p^n + u_p^{n+1} \right) \tag{15.2-60}$$

Equations 15.2-51 and 15.2-50 can also be computed using a Runge-Kutta scheme which was published by Cash and Karp [48]. The ordinary differential equations can be considered as vectors, where the left hand side is the derivative $\vec{y}'$ and the right hand side is an arbitrary function $\vec{f}(t, \vec{y})$.

$$\vec{y}' = \vec{f}(t, \vec{y}) \tag{15.2-61}$$

We get

$$\vec{y}^{n+1} = \vec{y}^n + c_1\vec{k}_1 + c_2\vec{k}_2 + c_3\vec{k}_3 + c_4\vec{k}_4 + c_5\vec{k}_5 + c_6\vec{k}_6 \tag{15.2-62}$$

with

$$\begin{aligned}
\vec{k}_1 &= \Delta t\vec{f}(t, \vec{y}^n) \\
\vec{k}_2 &= \Delta t\vec{f}(t + a_2\Delta t, \vec{y}^n + b_{21}\vec{k}_1) \\
\vec{k}_3 &= \Delta t\vec{f}(t + a_3\Delta t, \vec{y}^n + b_{31}\vec{k}_1 + b_{32}\vec{k}_2) \\
\vec{k}_4 &= \Delta t\vec{f}(t + a_4\Delta t, \vec{y}^n + b_{41}\vec{k}_1 + b_{42}\vec{k}_2 + b_{43}\vec{k}_3) \\
\vec{k}_5 &= \Delta t\vec{f}(t + a_5\Delta t, \vec{y}^n + b_{51}\vec{k}_1 + b_{52}\vec{k}_2 + b_{53}\vec{k}_3 + b_{54}\vec{k}_4) \\
\vec{k}_6 &= \Delta t\vec{f}(t + a_6\Delta t, \vec{y}^n + b_{61}\vec{k}_1 + b_{62}\vec{k}_2 + b_{63}\vec{k}_3 + b_{64}\vec{k}_4 + b_{65}\vec{k}_5)
\end{aligned}$$

The coefficients $a_2 \ldots a_6$, $b_{21} \ldots b_{65}$, and $c_1 \ldots c_6$ are taken from Cash and Karp [48]

This scheme provides an embedded error control, which is switched off, when no Accuracy Control is enabled.

For rotating reference frames, the integration is carried out in the rotating frame with the extra terms described in Equation 15.2-6 and Equation 15.2-7, thus accounting for system rotation. Using the mechanisms available for accuracy control, the trajectory integration will be done accurately in time.

The analytic scheme is very efficient. It can become inaccurate for large steps and in situations where the particles are not in hydrodynamic equilibrium with the continuous flow. The numerical schemes implicit and trapezoidal, in combination with Automated Tracking Scheme Selection, consider most of the changes in the forces acting on the particles and are chosen as default schemes. The runge-kutta scheme is recommended of nondrag force changes along a particle integration step.

The integration step size of the higher-order schemes, trapezoidal and runge-kutta, is limited to a stable range. Therefore it is recommended to use them in combination with Automated Tracking Scheme Selection.

For the massless particle type, the particle velocity is equal to the velocity of the continuous phase, hence the solution of only the trajectory Equation 15.2-50 is required where the particle velocity $u_p = u$. The new particle location along the trajectory is always computed by Equations 15.2-58 and 15.2-60, with $u_p = u$.

15.3 Laws for Drag Coefficients

Several laws for drag coefficients, C_D, are available for the Euler-Lagrange Model.

Instructions for selecting the drag law are provided in Section 23.2.4: Drag Laws in the separate User's Guide.

15.3.1 Spherical Drag Law

The drag coefficient, C_D, for smooth particles can be taken from

$$C_D = a_1 + \frac{a_2}{\mathrm{Re}} + \frac{a_3}{\mathrm{Re}^2} \tag{15.3-1}$$

where a_1, a_2, and a_3 are constants that apply over several ranges of Re given by Morsi and Alexander [238].

15.3.2 Non-spherical Drag Law

For non-spherical particles Haider and Levenspiel [119] developed the correlation

$$C_D = \frac{24}{\mathrm{Re_{sph}}}\left(1 + b_1 \mathrm{Re_{sph}}^{b_2}\right) + \frac{b_3 \mathrm{Re_{sph}}}{b_4 + \mathrm{Re_{sph}}} \tag{15.3-2}$$

where

$$\begin{aligned} b_1 &= \exp(2.3288 - 6.4581\phi + 2.4486\phi^2) \\ b_2 &= 0.0964 + 0.5565\phi \\ b_3 &= \exp(4.905 - 13.8944\phi + 18.4222\phi^2 - 10.2599\phi^3) \\ b_4 &= \exp(1.4681 + 12.2584\phi - 20.7322\phi^2 + 15.8855\phi^3) \end{aligned} \tag{15.3-3}$$

The shape factor, ϕ, is defined as

$$\phi = \frac{s}{S} \tag{15.3-4}$$

where s is the surface area of a sphere having the same volume as the particle, and S is the actual surface area of the particle. The Reynolds number $\mathrm{Re}_{\mathrm{sph}}$ is computed with the diameter of a sphere having the same volume.

i The shape factor cannot exceed a value of 1.

15.3.3 Stokes-Cunningham Drag Law

For sub-micron particles, a form of Stokes' drag law is available [261]. In this case, F_D is defined as

$$F_D = \frac{18\mu}{{d_p}^2 \rho_p C_c} \tag{15.3-5}$$

The factor C_c is the Cunningham correction to Stokes' drag law, which you can compute from

$$C_c = 1 + \frac{2\lambda}{d_p}(1.257 + 0.4e^{-(1.1 d_p/2\lambda)}) \tag{15.3-6}$$

where λ is the molecular mean free path.

15.3.4 High-Mach-Number Drag Law

A high-Mach-number drag law is also available. This drag law is similar to the spherical law (Equation 15.3-1) with corrections [57] to account for a particle Mach number greater than 0.4 at a particle Reynolds number greater than 20.

15.3.5 Dynamic Drag Model Theory

Accurate determination of droplet drag coefficients is crucial for accurate spray modeling. ANSYS FLUENT provides a method that determines the droplet drag coefficient dynamically, accounting for variations in the droplet shape.

The dynamic drag model is applicable in almost any circumstance. It is compatible with both the TAB and wave models for droplet breakup. When the collision model is turned on, collisions reset the distortion and distortion velocities of the colliding droplets.

Many droplet drag models assume the droplet remains spherical throughout the domain. With this assumption, the drag of a spherical object is determined by the following [205]:

$$C_{d,\mathrm{sphere}} = \begin{cases} 0.424 & \mathrm{Re} > 1000 \\ \frac{24}{\mathrm{Re}}\left(1 + \frac{1}{6}\mathrm{Re}^{2/3}\right) & \mathrm{Re} \leq 1000 \end{cases} \tag{15.3-7}$$

However, as an initially spherical droplet moves through a gas, its shape is distorted significantly when the Weber number is large. In the extreme case, the droplet shape will approach that of a disk. The drag of a disk, however, is significantly higher than that of a sphere. Since the droplet drag coefficient is highly dependent upon the droplet shape, a drag model that assumes the droplet is spherical is unsatisfactory. The dynamic drag model accounts for the effects of droplet distortion, linearly varying the drag between that of a sphere (Equation 15.3-7) and a value of 1.54 corresponding to a disk [205]. The drag coefficient is given by

$$C_d = C_{d,\mathrm{sphere}}(1 + 2.632y) \tag{15.3-8}$$

where y is the droplet distortion, as determined by the solution of

$$\frac{d^2y}{dt^2} = \frac{C_F}{C_b}\frac{\rho_g}{\rho_l}\frac{u^2}{r^2} - \frac{C_k\sigma}{\rho_l r^3}y - \frac{C_d\mu_l}{\rho_l r^2}\frac{dy}{dt} \tag{15.3-9}$$

In the limit of no distortion ($y = 0$), the drag coefficient of a sphere will be obtained, while at maximum distortion ($y = 1$) the drag coefficient corresponding to a disk will be obtained.

Note that Equation 15.3-9 is obtained from the TAB model for spray breakup, described in Section 15.10.1: Taylor Analogy Breakup (TAB) Model, but the dynamic drag model can be used with either of the breakup models.

15.3.6 Dense Discrete Phase Model Drag Laws

The drag laws that are suitable for dense gas-solid flow are the Wen and Yu, the Gidaspow, and the Syamlal-O'Brien drag model. A detailed theoretical background to the correlations, along with a recommendation of when to use each model, can be found in Section 16.5.4: Fluid-Solid Exchange Coefficient (see also Section 24.5.2: Specifying the Drag Function in the separate User's Guide). All three correlations incorporate group effects and therefore are dependent on the particle phase volume fraction. Thus, the new drag models are only available when the DPM volume fraction is computed, i.e. in conjunction with the Dense Discrete Phase Model.

15.4 Laws for Heat and Mass Exchange

Using ANSYS FLUENT's discrete phase modeling capability, reacting particles or droplets can be modeled and their impact on the continuous phase can be examined. Several heat and mass transfer relationships, termed "laws", are available in ANSYS FLUENT and the physical models employed in these laws are described in this section.

The laws that you activate depend upon the particle type that you select. In the **Set Injection Properties** dialog box you will specify the Particle Type, and ANSYS FLUENT will use a given set of heat and mass transfer laws for the chosen type. All particle types have predefined sequences of physical laws as shown in the table below:

Particle Type	**Description**	**Laws Activated**
Massless	–	–
Inert	inert/heating or cooling	1, 6
Droplet	heating/evaporation/boiling	1, 2, 3, 6

Combusting	heating;
	evolution of volatiles/swelling;
	heterogeneous surface reaction
		1, 4, 5, 6

Multicomponent	multicomponent droplets/particles	7

In addition to the above laws, you can define your own laws using a user-defined function. More information about user-defined functions can be found in the separate UDF Manual.

You can also extend combusting particles to include an evaporating/boiling material by selecting Wet Combustion in the **Set Injection Properties** dialog box.

ANSYS FLUENT's physical laws (Laws 1 through 6), which describe the heat and mass transfer conditions listed in this table, are explained in detail in the sections that follow.

15.4.1 Inert Heating or Cooling (Law 1/Law 6)

The inert heating or cooling laws (Laws 1 and 6) are applied when the particle temperature is less than the vaporization temperature that you define, $T_{\rm vap}$, and after the volatile fraction, $f_{v,0}$, of a particle has been consumed. These conditions may be written as

Law 1:

$$T_p < T_{\rm vap} \tag{15.4-1}$$

Law 6:

$$m_p \leq (1 - f_{v,0}) m_{p,0} \tag{15.4-2}$$

where T_p is the particle temperature, $m_{p,0}$ is the initial mass of the particle, and m_p is its current mass.

Law 1 is applied until the temperature of the particle/droplet reaches the vaporization temperature. At this point a noninert particle/droplet may proceed to obey one of the mass-transfer laws (2, 3, 4, and/or 5), returning to Law 6 when the volatile portion of the particle/droplet has been consumed. (Note that the vaporization temperature, $T_{\rm vap}$, is an arbitrary modeling constant used to define the onset of the vaporization/boiling/volatilization laws.)

When using Law 1 or Law 6, ANSYS FLUENT uses a simple heat balance to relate the particle temperature, $T_p(t)$, to the convective heat transfer and the absorption/emission of radiation at the particle surface:

$$m_p c_p \frac{dT_p}{dt} = h A_p (T_\infty - T_p) + \epsilon_p A_p \sigma (\theta_R^4 - T_p^4) \tag{15.4-3}$$

where

m_p = mass of the particle (kg)
c_p = heat capacity of the particle (J/kg-K)
A_p = surface area of the particle (m^2)
T_∞ = local temperature of the continuous phase (K)
h = convective heat transfer coefficient (W/m^2-K)
ϵ_p = particle emissivity (dimensionless)
σ = Stefan-Boltzmann constant (5.67 x 10^{-8} W/m^2-K^4)
θ_R = radiation temperature, $(\frac{G}{4\sigma})^{1/4}$

Equation 15.4-3 assumes that there is negligible internal resistance to heat transfer, i.e., the particle is at uniform temperature throughout.

G is the incident radiation in W/m^2:

$$G = \int_{\Omega=4\pi} I d\Omega \tag{15.4-4}$$

where I is the radiation intensity and Ω is the solid angle.

Radiation heat transfer to the particle is included only if you have enabled the P-1 or discrete ordinates radiation model and you have activated radiation heat transfer to particles using the Particle Radiation Interaction option in the **Discrete Phase Model** dialog box.

Equation 15.4-3 is integrated in time using an approximate, linearized form that assumes that the particle temperature changes slowly from one time value to the next:

$$m_p c_p \frac{dT_p}{dt} = A_p \left\{ - \left[h + \epsilon_p \sigma T_p^3 \right] T_p + \left[h T_\infty + \epsilon_p \sigma \theta_R^4 \right] \right\} \tag{15.4-5}$$

As the particle trajectory is computed, ANSYS FLUENT integrates Equation 15.4-5 to obtain the particle temperature at the next time value, yielding

$$T_p(t + \Delta t) = \alpha_p + [T_p(t) - \alpha_p] e^{-\beta_p \Delta t} \tag{15.4-6}$$

where Δt is the integration time step and

$$\alpha_p = \frac{h T_\infty + \epsilon_p \sigma \theta_R^4}{h + \epsilon_p \sigma T_p^3(t)} \tag{15.4-7}$$

and

$$\beta_p = \frac{A_p (h + \epsilon_p \sigma T_p^3(t))}{m_p c_p} \tag{15.4-8}$$

ANSYS FLUENT can also solve Equation 15.4-5 in conjunction with the equivalent mass transfer equation using a stiff coupled solver. See Section 23.2.8: Including Coupled Heat-Mass Solution Effects on the Particles in the separate User's Guide for details.

The heat transfer coefficient, h, is evaluated using the correlation of Ranz and Marshall [284, 285]:

$$\mathrm{Nu} = \frac{h d_p}{k_\infty} = 2.0 + 0.6 \mathrm{Re}_d^{1/2} \mathrm{Pr}^{1/3} \tag{15.4-9}$$

where

d_p = particle diameter (m)
k_∞ = thermal conductivity of the continuous phase (W/m-K)
Re_d = Reynolds number based on the particle diameter and the relative velocity (Equation 15.2-3)
Pr = Prandtl number of the continuous phase ($c_p \mu / k_\infty$)

Finally, the heat lost or gained by the particle as it traverses each computational cell appears as a source or sink of heat in subsequent calculations of the continuous phase energy equation. During Laws 1 and 6, particles/droplets do not exchange mass with the continuous phase and do not participate in any chemical reaction.

15.4.2 Droplet Vaporization (Law 2)

Law 2 is applied to predict the vaporization from a discrete phase droplet. Law 2 is initiated when the temperature of the droplet reaches the vaporization temperature, $T_{\rm vap}$, and continues until the droplet reaches the boiling point, $T_{\rm bp}$, or until the droplet's volatile fraction is completely consumed:

$$T_{\rm vap} \le T_p < T_{\rm bp} \qquad (15.4\text{-}10)$$

$$m_p > (1 - f_{v,0}) m_{p,0} \qquad (15.4\text{-}11)$$

The onset of the vaporization law is determined by the setting of $T_{\rm vap}$, a modeling parameter that has no physical significance. Note that once vaporization is initiated (by the droplet reaching this threshold temperature), it will continue to vaporize even if the droplet temperature falls below $T_{\rm vap}$. Vaporization will be halted only if the droplet temperature falls below the dew point. In such cases, the droplet will remain in Law 2 but no evaporation will be predicted. When the boiling point is reached, the droplet vaporization is predicted by a boiling rate, Law 3, as described in a section that follows.

Mass Transfer During Law 2

During Law 2, the rate of vaporization is governed by gradient diffusion, with the flux of droplet vapor into the gas phase related to the difference in vapor concentration at the droplet surface and the bulk gas:

$$N_i = k_c (C_{i,s} - C_{i,\infty}) \qquad (15.4\text{-}12)$$

where

N_i	=	molar flux of vapor (kgmol/m^2-s)
k_c	=	mass transfer coefficient (m/s)
$C_{i,s}$	=	vapor concentration at the droplet surface (kgmol/m^3)
$C_{i,\infty}$	=	vapor concentration in the bulk gas (kgmol/m^3)

Note that ANSYS FLUENT's vaporization law assumes that N_i is positive (evaporation). If conditions exist in which N_i is negative (i.e., the droplet temperature falls below the dew point and condensation conditions exist), ANSYS FLUENT treats the droplet as inert ($N_i = 0.0$).

The concentration of vapor at the droplet surface is evaluated by assuming that the partial pressure of vapor at the interface is equal to the saturated vapor pressure, $p_{\rm sat}$, at the particle droplet temperature, T_p:

$$C_{i,s} = \frac{p_{\rm sat}(T_p)}{RT_p} \tag{15.4-13}$$

where R is the universal gas constant.

The concentration of vapor in the bulk gas is known from solution of the transport equation for species i for nonpremixed or partially premixed combustion calculations:

$$C_{i,\infty} = X_i \frac{p}{RT_\infty} \tag{15.4-14}$$

where X_i is the local bulk mole fraction of species i, p is the local absolute pressure, and T_∞ is the local bulk temperature in the gas. The mass transfer coefficient in Equation 15.4-12 is calculated from the Sherwood number correlation [284, 285]:

$$\mathrm{Sh}_{AB} = \frac{k_c d_p}{D_{i,m}} = 2.0 + 0.6\mathrm{Re}_d^{1/2}\mathrm{Sc}^{1/3} \tag{15.4-15}$$

where $D_{i,m}$ = diffusion coefficient of vapor in the bulk (m^2/s)
Sc = the Schmidt number, $\frac{\mu}{\rho D_{i,m}}$
d_p = particle (droplet) diameter (m)

The vapor flux given by Equation 15.4-12 becomes a source of species i in the gas phase species transport equation, (see Section 23.5: Setting Material Properties for the Discrete Phase in the separate User's Guide) or in the mixture fraction equation for nonpremixed combustion calculations.

The mass of the droplet is reduced according to

$$m_p(t + \Delta t) = m_p(t) - N_i A_p M_{w,i} \Delta t \tag{15.4-16}$$

where $M_{w,i}$ = molecular weight of species i (kg/kgmol)
m_p = mass of the droplet (kg)
A_p = surface area of the droplet (m^2)

ANSYS FLUENT can also solve Equation 15.4-16 in conjunction with the equivalent heat transfer equation using a stiff coupled solver. See Section 23.2.8: Including Coupled Heat-Mass Solution Effects on the Particles in the separate User's Guide for details.

Defining the Vapor Pressure and Diffusion Coefficient

You must define the vapor pressure as a polynomial or piecewise linear function of temperature ($p_{\rm sat}(T)$) during the problem definition. Note that the vapor pressure definition is critical, as $p_{\rm sat}$ is used to obtain the driving force for the evaporation process (Equations 15.4-12 and 15.4-13). You should provide accurate vapor pressure values for temperatures over the entire range of possible droplet temperatures in your problem. Vapor pressure data can be obtained from a physics or engineering handbook (e.g., [266]).

You must also input the diffusion coefficient, $D_{i,m}$, during the setup of the discrete phase material properties. Note that the diffusion coefficient inputs that you supply for the continuous phase are not used in the discrete phase model.

Heat Transfer to the Droplet

Finally, the droplet temperature is updated according to a heat balance that relates the sensible heat change in the droplet to the convective and latent heat transfer between the droplet and the continuous phase:

$$m_p c_p \frac{dT_p}{dt} = hA_p(T_\infty - T_p) + \frac{dm_p}{dt} h_{\rm fg} + A_p \epsilon_p \sigma ({\theta_R}^4 - {T_p}^4) \tag{15.4-17}$$

where c_p = droplet heat capacity (J/kg-K)
T_p = droplet temperature (K)
h = convective heat transfer coefficient (W/m^2-K)
T_∞ = temperature of continuous phase (K)
$\frac{dm_p}{dt}$ = rate of evaporation (kg/s)
$h_{\rm fg}$ = latent heat (J/kg)
ϵ_p = particle emissivity (dimensionless)
σ = Stefan-Boltzmann constant (5.67 x 10^{-8} W/m^2-K^4)
θ_R = radiation temperature, $(\frac{I}{4\sigma})^{1/4}$, where I is the radiation intensity

Radiation heat transfer to the particle is included only if you have enabled the P-1 or discrete ordinates radiation model and you have activated radiation heat transfer to particles using the Particle Radiation Interaction option in the **Discrete Phase Model** dialog box.

The heat transferred to or from the gas phase becomes a source/sink of energy during subsequent calculations of the continuous phase energy equation.

15.4.3 Droplet Boiling (Law 3)

Law 3 is applied to predict the convective boiling of a discrete phase droplet when the temperature of the droplet has reached the boiling temperature, $T_{\rm bp}$, and while the mass of the droplet exceeds the nonvolatile fraction, $(1 - f_{v,0})$:

$$T_p \geq T_{\rm bp} \tag{15.4-18}$$

and

$$m_p > (1 - f_{v,0}) m_{p,0} \tag{15.4-19}$$

When the droplet temperature reaches the boiling point, a boiling rate equation is applied [173]:

$$\frac{d(d_p)}{dt} = \frac{4k_\infty}{\rho_p c_{p,\infty} d_p}(1 + 0.23\sqrt{{\rm Re}_d}) \ln\left[1 + \frac{c_{p,\infty}(T_\infty - T_p)}{h_{\rm fg}}\right] \tag{15.4-20}$$

where $c_{p,\infty}$ = heat capacity of the gas (J/kg-K)
ρ_p = droplet density (kg/m^3)
k_∞ = thermal conductivity of the gas (W/m-K)

Equation 15.4-20 was derived assuming steady flow at constant pressure. Note that the model requires $T_\infty > T_{\rm bp}$ in order for boiling to occur and that the droplet remains at fixed temperature ($T_{\rm bp}$) throughout the boiling law.

When radiation heat transfer is active, ANSYS FLUENT uses a slight modification of Equation 15.4-20, derived by starting from Equation 15.4-17 and assuming that the droplet temperature is constant. This yields

$$-\frac{dm_p}{dt} h_{\rm fg} = hA_p(T_\infty - T_p) + A_p \epsilon_p \sigma({\theta_R}^4 - {T_p}^4) \tag{15.4-21}$$

or

$$-\frac{d(d_p)}{dt} = \frac{2}{\rho_p h_{\rm fg}}\left[\frac{k_\infty {\rm Nu}}{d_p}(T_\infty - T_p) + \epsilon_p \sigma(\theta_R^4 - T_p^4)\right] \tag{15.4-22}$$

Using Equation 15.4-9 for the Nusselt number correlation and replacing the Prandtl number term with an empirical constant, Equation 15.4-22 becomes

$$-\frac{d(d_p)}{dt} = \frac{2}{\rho_p h_{\rm fg}}\left[\frac{2k_\infty[1 + 0.23\sqrt{{\rm Re}_d}]}{d_p}(T_\infty - T_p) + \epsilon_p \sigma(\theta_R^4 - T_p^4)\right] \tag{15.4-23}$$

In the absence of radiation, this result matches that of Equation 15.4-20 in the limit that the argument of the logarithm is close to unity. ANSYS FLUENT uses Equation 15.4-23 when radiation is active in your model and Equation 15.4-20 when radiation is not active. Radiation heat transfer to the particle is included only if you have enabled the P-1 or discrete ordinates radiation model and you have activated radiation heat transfer to particles using the Particle Radiation Interaction option in the **Discrete Phase Model** dialog box.

The droplet is assumed to stay at constant temperature while the boiling rate is applied. Once the boiling law is entered it is applied for the duration of the particle trajectory. The energy required for vaporization appears as a (negative) source term in the energy equation for the gas phase. The evaporated liquid enters the gas phase as species i, as defined by your input for the destination species (see Section 23.5: Setting Material Properties for the Discrete Phase in the separate User's Guide).

15.4.4 Devolatilization (Law 4)

The devolatilization law is applied to a combusting particle when the temperature of the particle reaches the vaporization temperature, T_{vap}, and remains in effect while the mass of the particle, m_p, exceeds the mass of the nonvolatiles in the particle:

$$T_p \geq T_{\text{vap}} \;\text{ and }\; T_p \geq T_{\text{bp}} \qquad (15.4\text{-}24)$$

and

$$m_p > (1 - f_{v,0})(1 - f_{w,0})m_{p,0} \qquad (15.4\text{-}25)$$

where $f_{w,0}$ is the mass fraction of the evaporating/boiling material if Wet Combustion is selected (otherwise, $f_{w,0} = 0$). As implied by Equation 15.4-24, the boiling point, T_{bp}, and the vaporization temperature, T_{vap}, should be set equal to each other when Law 4 is to be used. When wet combustion is active, T_{bp} and T_{vap} refer to the boiling and evaporation temperatures for the combusting material only.

ANSYS FLUENT provides a choice of four devolatilization models:

- the constant rate model (the default model)
- the single kinetic rate model
- the two competing rates model (the Kobayashi model)
- the chemical percolation devolatilization (CPD) model

Each of these models is described, in turn, below.

Choosing the Devolatilization Model

You will choose the devolatilization model when you are setting physical properties for the combusting-particle material in the **Create/Edit Materials** dialog box, as described in Section 23.5.2: Description of the Properties in the separate User's Guide. By default, the constant rate model (Equation 15.4-26) will be used.

The Constant Rate Devolatilization Model

The constant rate devolatilization law dictates that volatiles are released at a constant rate [20]:

$$-\frac{1}{f_{v,0}(1-f_{w,0})m_{p,0}}\frac{dm_p}{dt} = A_0 \tag{15.4-26}$$

where m_p = particle mass (kg)
$f_{v,0}$ = fraction of volatiles initially present in the particle
$m_{p,0}$ = initial particle mass (kg)
A_0 = rate constant (s^{-1})

The rate constant A_0 is defined as part of your modeling inputs, with a default value of 12 s^{-1} derived from the work of Pillai [271] on coal combustion. Proper use of the constant devolatilization rate requires that the vaporization temperature, which controls the onset of devolatilization, be set appropriately. Values in the literature show this temperature to be about 600 K [20].

The volatile fraction of the particle enters the gas phase as the devolatilizing species i, defined by you (see Section 23.5: Setting Material Properties for the Discrete Phase in the separate User's Guide). Once in the gas phase, the volatiles may react according to the inputs governing the gas phase chemistry.

The Single Kinetic Rate Model

The single kinetic rate devolatilization model assumes that the rate of devolatilization is first-order dependent on the amount of volatiles remaining in the particle [10]:

$$-\frac{dm_p}{dt} = k[m_p - (1-f_{v,0})(1-f_{w,0})m_{p,0}] \tag{15.4-27}$$

where m_p = particle mass (kg)
$f_{v,0}$ = mass fraction of volatiles initially present in the particle
$f_{w,0}$ = mass fraction of evaporating/boiling material (if wet combustion is modeled)
$m_{p,0}$ = initial particle mass (kg)
k = kinetic rate (s^{-1})

Note that $f_{v,0}$, the fraction of volatiles in the particle, should be defined using a value slightly in excess of that determined by proximate analysis. The kinetic rate, k, is defined by input of an Arrhenius type pre-exponential factor and an activation energy:

$$k = A_1 e^{-(E/RT)} \tag{15.4-28}$$

ANSYS FLUENT uses default rate constants, A_1 and E [10].

Equation 15.4-27 has the approximate analytical solution:

$$m_p(t + \Delta t) = (1 - f_{v,0})(1 - f_{w,0})m_{p,0} + [m_p(t) - (1 - f_{v,0})(1 - f_{w,0})m_{p,0}]e^{-k\Delta t} \tag{15.4-29}$$

which is obtained by assuming that the particle temperature varies only slightly between discrete time integration steps.

ANSYS FLUENT can also solve Equation 15.4-29 in conjunction with the equivalent heat transfer equation using a stiff coupled solver. See Section 23.2.8: Including Coupled Heat-Mass Solution Effects on the Particles in the separate User's Guide for details.

The Two Competing Rates (Kobayashi) Model

ANSYS FLUENT also provides the kinetic devolatilization rate expressions of the form proposed by Kobayashi [169]:

$$\mathcal{R}_1 = A_1 e^{-(E_1/RT_p)} \tag{15.4-30}$$

$$\mathcal{R}_2 = A_2 e^{-(E_2/RT_p)} \tag{15.4-31}$$

where $\mathcal{R}_1$ and $\mathcal{R}_2$ are competing rates that may control the devolatilization over different temperature ranges. The two kinetic rates are weighted to yield an expression for the devolatilization as

$$\frac{m_v(t)}{(1 - f_{w,0})m_{p,0} - m_a} = \int_0^t (\alpha_1 \mathcal{R}_1 + \alpha_2 \mathcal{R}_2) \exp\left(-\int_0^t (\mathcal{R}_1 + \mathcal{R}_2)\, dt\right) dt \tag{15.4-32}$$

where $m_v(t)$ = volatile yield up to time t
$m_{p,0}$ = initial particle mass at injection
α_1, α_2 = yield factors
m_a = ash content in the particle

The Kobayashi model requires input of the kinetic rate parameters, A_1, E_1, A_2, and E_2, and the yields of the two competing reactions, α_1 and α_2. ANSYS FLUENT uses default values for the yield factors of 0.3 for the first (slow) reaction and 1.0 for the second (fast) reaction. It is recommended in the literature [169] that α_1 be set to the fraction of volatiles determined by proximate analysis, since this rate represents devolatilization at low temperature. The second yield parameter, α_2, should be set close to unity, which is the yield of volatiles at very high temperature.

By default, Equation 15.4-32 is integrated in time analytically, assuming the particle temperature to be constant over the discrete time integration step. ANSYS FLUENT can also solve Equation 15.4-32 in conjunction with the equivalent heat transfer equation using a stiff coupled solver. See Section 23.2.8: Including Coupled Heat-Mass Solution Effects on the Particles in the separate User's Guide for details.

The CPD Model

In contrast to the coal devolatilization models presented above, which are based on empirical rate relationships, the chemical percolation devolatilization (CPD) model characterizes the devolatilization behavior of rapidly heated coal based on the physical and chemical transformations of the coal structure [100, 101, 116].

General Description

During coal pyrolysis, the labile bonds between the aromatic clusters in the coal structure lattice are cleaved, resulting in two general classes of fragments. One set of fragments has a low molecular weight (and correspondingly high vapor pressure) and escapes from the coal particle as a light gas. The other set of fragments consists of tar gas precursors that have a relatively high molecular weight (and correspondingly low vapor pressure) and tend to remain in the coal for a long period of time during typical devolatilization conditions. During this time, reattachment with the coal lattice (which is referred to as crosslinking) can occur. The high molecular weight compounds plus the residual lattice are referred to as metaplast. The softening behavior of a coal particle is determined by the quantity and nature of the metaplast generated during devolatilization. The portion of the lattice structure that remains after devolatilization is comprised of char and mineral-compound-based ash.

The CPD model characterizes the chemical and physical processes by considering the coal structure as a simplified lattice or network of chemical bridges that link the aromatic clusters. Modeling the cleavage of the bridges and the generation of light gas, char, and tar precursors is then considered to be analogous to the chemical reaction scheme shown in Figure 15.4.1.

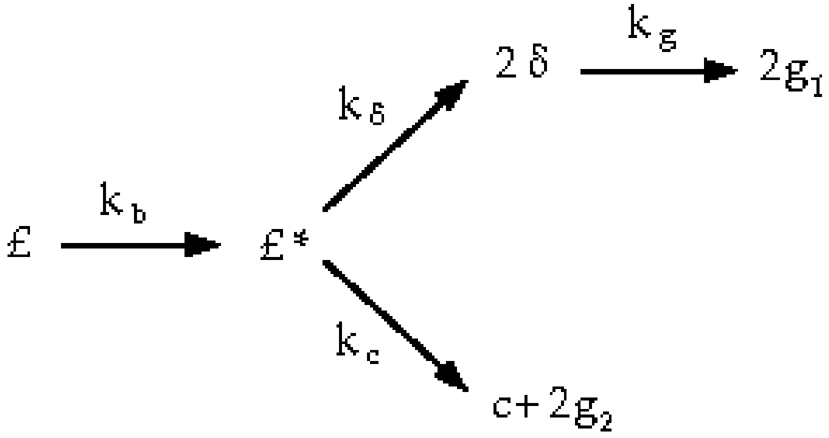


Figure 15.4.1: Coal Bridge

The variable $\pounds$ represents the original population of labile bridges in the coal lattice. Upon heating, these bridges become the set of reactive bridges, $\pounds^*$. For the reactive bridges, two competing paths are available. In one path, the bridges react to form side chains, δ. The side chains may detach from the aromatic clusters to form light gas, g_1. As bridges between neighboring aromatic clusters are cleaved, a certain fraction of the coal becomes detached from the coal lattice. These detached aromatic clusters are the heavy-molecular-weight tar precursors that form the metaplast. The metaplast vaporizes to form coal tar. While waiting for vaporization, the metaplast can also reattach to the coal lattice matrix (crosslinking). In the other path, the bridges react and become a char bridge, c, with the release of an associated light gas product, g_2. The total population of bridges in the coal lattice matrix can be represented by the variable p, where $p = \pounds + c$.

Reaction Rates

Given this set of variables that characterizes the coal lattice structure during devolatilization, the following set of reaction rate expressions can be defined for each, starting with the assumption that the reactive bridges are destroyed at the same rate at which they are created ($\frac{\partial \pounds^*}{\partial t} = 0$):

$$\frac{d\pounds}{dt} = -k_b \pounds \qquad (15.4\text{-}33)$$

$$\frac{dc}{dt} = k_b \frac{\pounds}{\rho + 1} \qquad (15.4\text{-}34)$$

$$\frac{d\delta}{dt} = \left[2\rho k_b \frac{\pounds}{\rho + 1} \right] - k_g \delta \qquad (15.4\text{-}35)$$

$$\frac{dg_1}{dt} = k_g \delta \qquad (15.4\text{-}36)$$

$$\frac{dg_2}{dt} = 2 \frac{dc}{dt} \qquad (15.4\text{-}37)$$

where the rate constants for bridge breaking and gas release steps, k_b and k_g, are expressed in Arrhenius form with a distributed activation energy:

$$k = Ae^{-(E \pm E_\sigma)/RT} \tag{15.4-38}$$

where A, E, and E_σ are, respectively, the pre-exponential factor, the activation energy, and the distributed variation in the activation energy, R is the universal gas constant, and T is the temperature. The ratio of rate constants, $\rho = k_\delta / k_c$, is set to 0.9 in this model based on experimental data.

Mass Conservation

The following mass conservation relationships are imposed:

$$\begin{aligned} g &= g_1 + g_2 && (15.4\text{-}39) \\ g_1 &= 2f - \sigma && (15.4\text{-}40) \\ g_2 &= 2(c - c_0) && (15.4\text{-}41) \end{aligned}$$

where f is the fraction of broken bridges ($f = 1 - p$). The initial conditions for this system are given by the following:

$$\begin{aligned} c(0) &= c_0 && (15.4\text{-}42) \\ \pounds(0) &= \pounds_0 = p_0 - c_0 && (15.4\text{-}43) \\ \delta(0) &= 2f_0 = 2(1 - c_0 - \pounds_0) && (15.4\text{-}44) \\ g(0) &= g_1(0) = g_2(0) = 0 && (15.4\text{-}45) \end{aligned}$$

where c_0 is the initial fraction of char bridges, p_0 is the initial fraction of bridges in the coal lattice, and $\pounds_0$ is the initial fraction of labile bridges in the coal lattice.

Fractional Change in the Coal Mass

Given the set of reaction equations for the coal structure parameters, it is necessary to relate these quantities to changes in coal mass and the related release of volatile products. To accomplish this, the fractional change in the coal mass as a function of time is divided into three parts: light gas ($f_{\rm gas}$), tar precursor fragments ($f_{\rm frag}$), and char ($f_{\rm char}$). This is accomplished by using the following relationships, which are obtained using percolation lattice statistics:

$$f_{\rm gas}(t) = \frac{r(g_1+g_2)(\sigma+1)}{4+2r(1-c_0)(\sigma+1)} \qquad (15.4\text{-}46)$$

$$f_{\rm frag}(t) = \frac{2}{2+r(1-c_0)(\sigma+1)}\left[\Phi F(p) + r\Omega K(p)\right] \qquad (15.4\text{-}47)$$

$$f_{\rm char}(t) = 1 - f_{\rm gas}(t) - f_{\rm frag}(t) \qquad (15.4\text{-}48)$$

The variables Φ, Ω, $F(p)$, and $K(p)$ are the statistical relationships related to the cleaving of bridges based on the percolation lattice statistics, and are given by the following equations:

$$\Phi = 1 + r\left[\frac{\pounds}{p} + \frac{(\sigma-1)\delta}{4(1-p)}\right] \qquad (15.4\text{-}49)$$

$$\Omega = \frac{\delta}{2(1-p)} - \frac{\pounds}{p} \qquad (15.4\text{-}50)$$

$$F(p) = \left(\frac{p'}{p}\right)^{\frac{\sigma+1}{\sigma-1}} \qquad (15.4\text{-}51)$$

$$K(p) = \left[1 - \left(\frac{\sigma+1}{2}\right)p'\right]\left(\frac{p'}{p}\right)^{\frac{\sigma+1}{\sigma-1}} \qquad (15.4\text{-}52)$$

r is the ratio of bridge mass to site mass, m_b/m_a, where

$$m_b = 2M_{w,\delta} \qquad (15.4\text{-}53)$$

$$m_a = M_{w,1} - (\sigma+1)M_{w,\delta} \qquad (15.4\text{-}54)$$

where $M_{w,\delta}$ and $M_{w,1}$ are the side chain and cluster molecular weights respectively. $\sigma+1$ is the lattice coordination number, which is determined from solid-state nuclear magnetic Resonance (NMR) measurements related to coal structure parameters, and p' is the root of the following equation in p (the total number of bridges in the coal lattice matrix):

$$p'(1-p')^{\sigma-1} = p(1-p)^{\sigma-1} \tag{15.4-55}$$

In accounting for mass in the metaplast (tar precursor fragments), the part that vaporizes is treated in a manner similar to flash vaporization, where it is assumed that the finite fragments undergo vapor/liquid phase equilibration on a time scale that is rapid with respect to the bridge reactions. As an estimate of the vapor/liquid that is present at any time, a vapor pressure correlation based on a simple form of Raoult's Law is used. The vapor pressure treatment is largely responsible for predicting pressure-dependent devolatilization yields. For the part of the metaplast that reattaches to the coal lattice, a cross-linking rate expression given by the following equation is used:

$$\frac{dm_{\text{cross}}}{dt} = m_{\text{frag}} A_{\text{cross}} e^{-(E_{\text{cross}}/RT)} \tag{15.4-56}$$

where m_{cross} is the amount of mass reattaching to the matrix, m_{frag} is the amount of mass in the tar precursor fragments (metaplast), and A_{cross} and E_{cross} are rate expression constants.

CPD Inputs

Given the set of equations and corresponding rate constants introduced for the CPD model, the number of constants that must be defined to use the model is a primary concern. For the relationships defined previously, it can be shown that the following parameters are coal independent [100]:

- A_b, E_b, $E_{\sigma b}$, A_g, E_g, and $E_{\sigma g}$ for the rate constants k_b and k_g
- A_{cross}, E_{cross}, and ρ

These constants are included in the submodel formulation and are not input or modified during problem setup.

There are an additional five parameters that are coal-specific and must be specified during the problem setup:

- initial fraction of bridges in the coal lattice, p_0
- initial fraction of char bridges, c_0
- lattice coordination number, $\sigma + 1$
- cluster molecular weight, $M_{w,1}$
- side chain molecular weight, $M_{w,\delta}$

The first four of these are coal structure quantities that are obtained from NMR experimental data. The last quantity, representing the char bridges that either exist in the parent coal or are formed very early in the devolatilization process, is estimated based on the coal rank. These quantities are entered in the Create/Edit Materials dialog box, as described in Section 23.5.2: Description of the Properties in the separate User's Guide. Values for the coal-dependent parameters for a variety of coals are listed in Table 15.4.1.

Table 15.4.1: Chemical Structure Parameters for ^{13}C NMR for 13 Coals

Coal Type	$\sigma + 1$	p_0	$M_{w,1}$	$M_{w,\delta}$	c_0
Zap (AR)	3.9	.63	277	40	.20
Wyodak (AR)	5.6	.55	410	42	.14
Utah (AR)	5.1	.49	359	36	0
Ill6 (AR)	5.0	.63	316	27	0
Pitt8 (AR)	4.5	.62	294	24	0
Stockton (AR)	4.8	.69	275	20	0
Freeport (AR)	5.3	.67	302	17	0
Pocahontas (AR)	4.4	.74	299	14	.20
Blue (Sandia)	5.0	.42	410	47	.15
Rose (AFR)	5.8	.57	459	48	.10
1443 (lignite, ACERC)	4.8	.59	297	36	.20
1488 (subbituminous, ACERC)	4.7	.54	310	37	.15
1468 (anthracite, ACERC)	4.7	.89	656	12	.25

AR refers to eight types of coal from the Argonne premium sample bank [329, 363]. Sandia refers to the coal examined at Sandia National Laboratories [99]. AFR refers to coal examined at Advanced Fuel Research. ACERC refers to three types of coal examined at the Advanced Combustion Engineering Research Center.

Particle Swelling During Devolatilization

The particle diameter changes during devolatilization according to the swelling coefficient, $C_{\rm sw}$, which is defined by you and applied in the following relationship:

$$\frac{d_p}{d_{p,0}} = 1 + (C_{\rm sw} - 1)\frac{(1 - f_{w,0})m_{p,0} - m_p}{f_{v,0}(1 - f_{w,0})m_{p,0}} \qquad (15.4\text{-}57)$$

where $d_{p,0}$ = particle diameter at the start of devolatilization
d_p = current particle diameter

The term $\frac{(1-f_{w,0})m_{p,0}-m_p}{f_{v,0}(1-f_{w,0})m_{p,0}}$ is the ratio of the mass that has been devolatilized to the total volatile mass of the particle. This quantity approaches a value of 1.0 as the devolatilization law is applied. When the swelling coefficient is equal to 1.0, the particle diameter stays constant. When the swelling coefficient is equal to 2.0, the final particle diameter doubles when all of the volatile component has vaporized, and when the swelling coefficient is equal to 0.5 the final particle diameter is half of its initial diameter.

Heat Transfer to the Particle During Devolatilization

Heat transfer to the particle during the devolatilization process includes contributions from convection, radiation (if active), and the heat consumed during devolatilization:

$$m_p c_p \frac{dT_p}{dt} = hA_p(T_\infty - T_p) + \frac{dm_p}{dt}h_{\rm fg} + A_p\epsilon_p\sigma({\theta_R}^4 - {T_p}^4) \qquad (15.4\text{-}58)$$

Radiation heat transfer to the particle is included only if you have enabled the P-1 or discrete ordinates radiation model and you have activated radiation heat transfer to particles using the Particle Radiation Interaction option in the **Discrete Phase Model** dialog box.

By default, Equation 15.4-58 is solved analytically, by assuming that the temperature and mass of the particle do not change significantly between time steps:

$$T_p(t + \Delta t) = \alpha_p + [T_p(t) - \alpha_p]e^{-\beta_p \Delta t} \qquad (15.4\text{-}59)$$

where

$$\alpha_p = \frac{hA_pT_\infty + \frac{dm_p}{dt}h_{\rm fg} + A_p\epsilon_p\sigma{\theta_R}^4}{hA_p + \epsilon_pA_p\sigma{T_p}^3} \qquad (15.4\text{-}60)$$

and

$$\beta_p = \frac{A_p(h + \epsilon_p \sigma {T_p}^3)}{m_p c_p} \tag{15.4-61}$$

ANSYS FLUENT can also solve Equation 15.4-58 in conjunction with the equivalent mass transfer equation using a stiff coupled solver. See Section 23.2.8: Including Coupled Heat-Mass Solution Effects on the Particles in the separate User's Guide for details.

15.4.5 Surface Combustion (Law 5)

After the volatile component of the particle is completely evolved, a surface reaction begins which consumes the combustible fraction, $f_{\rm comb}$, of the particle. Law 5 is thus active (for a combusting particle) after the volatiles are evolved:

$$m_p < (1 - f_{v,0})(1 - f_{w,0}) m_{p,0} \tag{15.4-62}$$

and until the combustible fraction is consumed:

$$m_p > (1 - f_{v,0} - f_{\rm comb})(1 - f_{w,0}) m_{p,0} \tag{15.4-63}$$

When the combustible fraction, $f_{\rm comb}$, has been consumed in Law 5, the combusting particle may contain residual "ash" that reverts to the inert heating law, Law 6 (described previously).

With the exception of the multiple surface reactions model, the surface combustion law consumes the reactive content of the particle as governed by the stoichiometric requirement, S_b, of the surface "burnout" reaction:

$$\text{char(s)} + S_b \text{ox(g)} \longrightarrow \text{products(g)} \tag{15.4-64}$$

where S_b is defined in terms of mass of oxidant per mass of char, and the oxidant and product species are defined in the **Set Injection Properties** dialog box.

ANSYS FLUENT provides a choice of four heterogeneous surface reaction rate models for combusting particles:

- the diffusion-limited rate model (the default model)
- the kinetics/diffusion-limited rate model
- the intrinsic model
- the multiple surface reactions model

Each of these models is described in detail below. You will choose the surface combustion model when you are setting physical properties for the combusting-particle material, as described in Section 23.5.2: Description of the Properties in the separate User's Guide. By default, the diffusion-limited rate model will be used.

The Diffusion-Limited Surface Reaction Rate Model

The diffusion-limited surface reaction rate model which is the default model in ANSYS FLUENT, assumes that the surface reaction proceeds at a rate determined by the diffusion of the gaseous oxidant to the surface of the particle:

$$\frac{dm_p}{dt} = -4\pi d_p D_{i,m} \frac{Y_{\rm ox} T_\infty \rho}{S_b (T_p + T_\infty)} \tag{15.4-65}$$

where $D_{i,m}$ = diffusion coefficient for oxidant in the bulk (m^2/s)
$Y_{\rm ox}$ = local mass fraction of oxidant in the gas
ρ = gas density ($\mathrm{kg/m}^3$)
S_b = stoichiometry of Equation 15.4-64

Equation 15.4-65 is derived from the model of Baum and Street [20] with the kinetic contribution to the surface reaction rate ignored. The diffusion-limited rate model assumes that the diameter of the particles does not change. Since the mass of the particles is decreasing, the effective density decreases, and the char particles become more porous.

The Kinetic/Diffusion Surface Reaction Rate Model

The kinetic/diffusion-limited rate model assumes that the surface reaction rate is determined either by kinetics or by a diffusion rate. ANSYS FLUENT uses the model of Baum and Street [20] and Field [96], in which a diffusion rate coefficient

$$D_0 = C_1 \frac{[(T_p + T_\infty)/2]^{0.75}}{d_p} \tag{15.4-66}$$

and a kinetic rate

$$\mathcal{R} = C_2 e^{-(E/RT_p)} \tag{15.4-67}$$

are weighted to yield a char combustion rate of

$$\frac{dm_p}{dt} = -A_p p_{\rm ox} \frac{D_0 \mathcal{R}}{D_0 + \mathcal{R}} \tag{15.4-68}$$

where A_p is the surface area of the droplet (πd_p^2), $p_{\rm ox}$ is the partial pressure of oxidant species in the gas surrounding the combusting particle, and the kinetic rate, $\mathcal{R}$, incorporates the effects of chemical reaction on the internal surface of the char particle (intrinsic reaction) and pore diffusion. In ANSYS FLUENT, Equation 15.4-68 is recast in terms of the oxidant mass fraction, $Y_{\rm ox}$, as

$$\frac{dm_p}{dt} = -A_p \frac{\rho R T_\infty Y_{\rm ox}}{M_{w,\rm ox}} \; \frac{D_0 \mathcal{R}}{D_0 + \mathcal{R}} \tag{15.4-69}$$

The particle size is assumed to remain constant in this model while the density is allowed to decrease.

When this model is enabled, the rate constants used in Equations 15.4-66 and 15.4-67 are entered in the **Create/Edit Materials** dialog box, as described in Section 23.5: Setting Material Properties for the Discrete Phase in the separate User's Guide.

The Intrinsic Model

The intrinsic model in ANSYS FLUENT is based on Smith's model [324], assuming the order of reaction is equal to unity. Like the kinetic/diffusion model, the intrinsic model assumes that the surface reaction rate includes the effects of both bulk diffusion and chemical reaction (see Equation 15.4-69). The intrinsic model uses Equation 15.4-66 to compute the diffusion rate coefficient, D_0, but the chemical rate, $\mathcal{R}$, is explicitly expressed in terms of the intrinsic chemical and pore diffusion rates:

$$\mathcal{R} = \eta \frac{d_p}{6} \rho_p A_g k_i \tag{15.4-70}$$

η is the effectiveness factor, or the ratio of the actual combustion rate to the rate attainable if no pore diffusion resistance existed [182]:

$$\eta = \frac{3}{\phi^2}(\phi \coth \phi - 1) \tag{15.4-71}$$

where ϕ is the Thiele modulus:

$$\phi = \frac{d_p}{2} \left[\frac{S_b \rho_p A_g k_i p_{\rm ox}}{D_e \rho_{\rm ox}} \right]^{1/2} \tag{15.4-72}$$

$\rho_{\rm ox}$ is the density of oxidant in the bulk gas (kg/m^3) and D_e is the effective diffusion coefficient in the particle pores. Assuming that the pore size distribution is unimodal and the bulk and Knudsen diffusion proceed in parallel, D_e is given by

$$D_e = \frac{\theta}{\tau^2}\left[\frac{1}{D_{\rm Kn}} + \frac{1}{D_0}\right]^{-1} \tag{15.4-73}$$

where D_0 is the bulk molecular diffusion coefficient and θ is the porosity of the char particle:

$$\theta = 1 - \frac{\rho_p}{\rho_t} \tag{15.4-74}$$

ρ_p and ρ_t are, respectively, the apparent and true densities of the pyrolysis char.

τ (in Equation 15.4-73) is the tortuosity of the pores. The default value for τ in ANSYS FLUENT is $\sqrt{2}$, which corresponds to an average intersecting angle between the pores and the external surface of 45° [182].

$D_{\rm Kn}$ is the Knudsen diffusion coefficient:

$$D_{\rm Kn} = 97.0\overline{r}_p\sqrt{\frac{T_p}{M_{w,{\rm ox}}}} \tag{15.4-75}$$

where T_p is the particle temperature and $\overline{r}_p$ is the mean pore radius of the char particle, which can be measured by mercury porosimetry. Note that macropores ($\overline{r}_p > 150$ Å) dominate in low-rank chars while micropores ($\overline{r}_p < 10$ Å) dominate in high-rank chars [182].

A_g (in Equations 15.4-70 and 15.4-72) is the specific internal surface area of the char particle, which is assumed in this model to remain constant during char combustion. Internal surface area data for various pyrolysis chars can be found in [323]. The mean value of the internal surface area during char combustion is higher than that of the pyrolysis char [182]. For example, an estimated mean value for bituminous chars is 300 m^2/g [50].

k_i (in Equations 15.4-70 and 15.4-72) is the intrinsic reactivity, which is of Arrhenius form:

$$k_i = A_i e^{-(E_i/RT_p)} \tag{15.4-76}$$

where the pre-exponential factor A_i and the activation energy E_i can be measured for each char. In the absence of such measurements, the default values provided by ANSYS FLUENT (which are taken from a least squares fit of data of a wide range of porous carbons, including chars [323]) can be used.

To allow a more adequate description of the char particle size (and hence density) variation during combustion, you can specify the burning mode α, relating the char particle diameter to the fractional degree of burnout U (where $U = 1 - m_p/m_{p,0}$) by [322]

$$\frac{d_p}{d_{p,0}} = (1 - U)^\alpha \tag{15.4-77}$$

where m_p is the char particle mass and the subscript zero refers to initial conditions (i.e., at the start of char combustion). Note that $0 \leq \alpha \leq 1/3$ where the limiting values 0 and 1/3 correspond, respectively, to a constant size with decreasing density (zone 1) and a decreasing size with constant density (zone 3) during burnout. In zone 2, an intermediate value of $\alpha = 0.25$, corresponding to a decrease of both size and density, has been found to work well for a variety of chars [322].

When this model is enabled, the rate constants used in Equations 15.4-66, 15.4-70, 15.4-72, 15.4-73, 15.4-75, 15.4-76, and 15.4-77 are entered in the **Create/Edit Materials** dialog box, as described in Section 23.5: Setting Material Properties for the Discrete Phase in the separate User's Guide.

The Multiple Surface Reactions Model

Modeling multiple particle surface reactions follows a pattern similar to the wall surface reaction models, where the surface species is now a "particle surface species". For the mixture material defined in the Species Model dialog box, the particle surface species can be depleted or produced by the stoichiometry of the particle surface reaction (defined in the Reactions dialog box). The particle surface species constitutes the reactive char mass of the particle, hence, if a particle surface species is depleted, the reactive "char" content of the particle is consumed, and in turn, when a surface species is produced, it is added to the particle "char" mass. Any number of particle surface species and any number of particle surface reactions can be defined for any given combusting particle.

Multiple injections can be accommodated, and combusting particles reacting according to the multiple surface reactions model can coexist in the calculation, with combusting particles following other char combustion laws. The model is based on oxidation studies of char particles, but it is also applicable to gas-solid reactions in general, not only to char oxidation reactions.

See Section 7.3: Particle Surface Reactions for information about particle surface reactions.

Limitations

Note the following limitations of the multiple surface reactions model:

- The model is not available together with the unsteady tracking option.
- The model is available only with the species transport model for volumetric reactions, and not with the nonpremixed, premixed, or partially premixed combustion models.

Heat and Mass Transfer During Char Combustion

The surface reaction consumes the oxidant species in the gas phase; i.e., it supplies a (negative) source term during the computation of the transport equation for this species. Similarly, the surface reaction is a source of species in the gas phase: the product of the heterogeneous surface reaction appears in the gas phase as a user-selected chemical species. The surface reaction also consumes or produces energy, in an amount determined by the heat of reaction defined by you.

The particle heat balance during surface reaction is

$$m_p c_p \frac{dT_p}{dt} = hA_p(T_\infty - T_p) - f_h \frac{dm_p}{dt} H_{\rm reac} + A_p \epsilon_p \sigma ({\theta_R}^4 - {T_p}^4) \tag{15.4-78}$$

where $H_{\rm reac}$ is the heat released by the surface reaction. Note that only a portion $(1 - f_h)$ of the energy produced by the surface reaction appears as a heat source in the gas-phase energy equation: the particle absorbs a fraction f_h of this heat directly. For coal combustion, it is recommended that f_h be set to 1.0 if the char burnout product is CO and 0.3 if the char burnout product is CO_2 [33].

Radiation heat transfer to the particle is included only if you have enabled the P-1 or discrete ordinates radiation model and you have activated radiation heat transfer to particles using the Particle Radiation Interaction option in the **Discrete Phase Model** dialog box.

By default, Equation 15.4-78 is solved analytically, by assuming that the temperature and mass of the particle do not change significantly between time steps. ANSYS FLUENT can also solve Equation 15.4-78 in conjunction with the equivalent mass transfer equation using a stiff coupled solver. See Section 23.2.8: Including Coupled Heat-Mass Solution Effects on the Particles in the separate User's Guide for details.

15.4.6 Multicomponent Particle Definition (Law 7)

Multicomponent particles are described in ANSYS FLUENT as a mixture of species within droplets/particles. The particle mass m is the sum of the masses of the components

$$m = \sum_i m_i \qquad (15.4\text{-}79)$$

The density of the particle ρ_p can be either constant, or volume-averaged:

$$\rho_p = \left(\sum_i \frac{m_i}{m \rho_i} \right)^{-1} \qquad (15.4\text{-}80)$$

For particles containing more than one component it is difficult to assign the whole particle to one process like boiling or heating. Therefore it can be only modeled by a law integrating all processes of relevance in one equation. The source terms for temperature and component mass are the sum of the sources from the partial processes:

$$m_p c_p \left(\frac{dT_p}{dt} \right) = A_p \epsilon_p \sigma (\theta_R^4 - T_p^4) + h A_p (T_\infty - T_p) + \sum_i \frac{dm_i}{dt} (h_{i,p} - h_{i,g}) \qquad (15.4\text{-}81)$$

$$\left(\frac{dm_i}{dt} \right) = A_p M_{w,i} k_{c,i} (C_{i,s} - C_{i,\infty}) \qquad (15.4\text{-}82)$$

The equation for the particle temperature T consists of terms for radiation, convective heating (Equation 15.4-3) and vaporization. Radiation heat transfer to the particle is included only if you have enabled P-1 or Discrete-Ordinates (DO) radiation and you have activated radiation heat transfer to the particles using the Particle Radiation Interaction option in the Discrete Phase Model dialog box.

The mass of the particle components m_i is only influenced by the vaporization (Equation 15.4-12), where $M_{w,i}$ is the molecular weight of species i. The mass transfer coefficient $k_{c,i}$ of component i is calculated from the Sherwood correlation (Equation 15.4-15). The concentration of vapor at the particle surface $C_{i,s}$ depends on the saturation pressure of the component.

Raoult's Law

The correlation between the vapor concentration of a species $C_{i,s}$ over the surface and its mole fraction in the condensed phase X_i^L is described by Raoult's law:

$$C_{i,s} = \frac{p_i}{RT} = \frac{X_i^L p}{RT} \qquad (15.4\text{-}83)$$

Peng-Robinson Real Gas Model

For the calculation of the vapor concentration of a species $C_{i,S}$ over the surface depends on whether the compressability of the vapor phase Z^V is taken into account:

$$C_{i,S} = {x_i}^V \frac{p}{Z^V RT} \tag{15.4-84}$$

The data for the vapor pressure are no longer available, when a droplet material is chosen as the component of a mixture, because it is not necessary for the calculation.

Besides using Raoult's Law and the Peng-Robinson equation of state, you can define your own user-defined function for delivering the vapor concentration at the particle surface .

For more information, see Section 2.5.15: `DEFINE_DPM_VP_EQUILIB` in the separate UDF Manual.

15.5 Vapor Liquid Equilibrium Theory

A number of industrially important processes, such as distillation, absorption and extraction, bring into contact two phases which are not at equilibrium. The rate at which a species is transferred from one phase to the other depends on the departure of the system from equilibrium. The quantitative treatment of these rate processes requires knowledge of the equilibrium states of the system. Apart from these cases, vapor-liquid equilibrium (VLE) relationships in multicomponent systems are needed for the solution of many other classes of engineering problems, such as the computation of evaporation rates in spray combustion applications.

In ANSYS FLUENT the rate of vaporization of a single component droplet is computed from Equation 15.4-12, where $C_{i,s}$ is the equilibrium concentration of the droplet species in the gas phase, and is computed in Equation 15.4-13 as:

$$C_{i,s} = p_{sat}/RT_p \tag{15.5-1}$$

where T_p is the droplet temperature, and p_{sat} is the saturation pressure of the droplet species at T_p.

For the general case where N components are evaporating from a droplet (distillation), the evaporation rate of each species is again given by Equation 15.4-12; however, p_{sat} in Equation 15.5-1 must be replaced by p_i, the partial pressure of species i, to calculate the concentration of i at the droplet surface.

The partial pressure of species i can be obtained from the general expression for two phase equilibrium, equating the fugacity of the liquid and vapor mixture components [275]:

$${f_i}^V = {x_i}^V {\phi_i}^V p = {x_i}^L {\phi_i}^L p = {f_i}^L \tag{15.5-2}$$

where x_i is the mole fraction, ϕ_i is the fugacity coefficient for the species i in the mixture, and p is the absolute pressure. The superscripts V and L are the vapor and the liquid phase variables, respectively. The fugacity coefficients account for the nonideality in the gas and liquid mixture. The fugacity of the liquid phase can be calculated from the pure component's saturation pressure $p_{sat,i}$ [325]:

$$f_i{}^L = x_i{}^L \phi_i{}^L p = \gamma_i x_i^L \phi_{sat,i}{}^L p_{sat,i} exp\left[\frac{V_i^L(p - p_{sat,i})}{RT}\right] \tag{15.5-3}$$

Here, $\phi_{sat,i}{}^L$ is the fugacity coefficient for pure i at the saturation pressure; γ_i is the activity coefficient for species i in the mixture, and accounts for the nonideality in the liquid phase; T is the particle surface temperature. We assume perfect thermal conductivity inside the particle, so the particle temperature is used instead; R is the universal gas constant; V_i^L is the molar volume of the liquid. The exponential term is the Poynting correction factor and accounts for the compressibility effects within the liquid. Except at high pressures, the Poynting factor is usually negligible. Under low pressure conditions where the gas phase may be assumed to be ideal, $\phi_i{}^V \approx 1$ and $\phi_{sat,i} \approx 1$. Furthermore, if the liquid is also assumed to be ideal, $\gamma \approx 1$, then Equation 15.5-2 reduces to Raoult's law,

$$x_i{}^V p = x_i{}^L p_{sat,i} \tag{15.5-4}$$

Raoult's law is the default vapor-liquid equilibrium expression used in the ANSYS FLUENT multicomponent droplet model. However, there is a UDF hook available for user-defined vapor-liquid equilibrium models.

While Raoult's law represents the simplest form of the VLE equation, keep in mind that it is of limited use, as the assumptions made for its derivation are usually unrealistic. The most critical assumption is that the liquid phase is an ideal solution. This is not likely to be valid, unless the system is made up of species of similar molecular sizes and chemical nature, such as in the case of benzene and toluene, or n-heptane and n-hexane.

When Raoult's law is applicable, the vaporization rate of each species from a multicomponent droplet can be computed from Equation 15.4-12, with the equilibrium concentration of species i in the gas phase $C_{i,s}$ computed as:

$$C_{i,s} = x_i p_{sat,i}/RT_p \tag{15.5-5}$$

where T_p is the droplet temperature, x_i is the mole fraction of species i in the droplet, and $p_{sat,i}$ is the saturation pressure of species i at T_p.

For higher pressures, especially near or above the critical point of the components, real gas effects must be considered. Most models describing the fugacity coefficients use a cubic equation of state with the general form:

$$p = \frac{RT}{V-b} - \frac{a(V-\eta)}{(V-b)(V^2+\delta V-\epsilon)} \tag{15.5-6}$$

where V is the molar volume. For in-cylinder applications, the Peng-Robinson equation of state is often used [265], where $\delta = 2b$, $\epsilon = -b^2$, and $\eta = b$:

$$p = \frac{RT}{V-b} - \frac{a}{V^2+2bV-b^2} \tag{15.5-7}$$

This equation defines the compressibility

$$Z = \frac{pV}{RT} = \frac{RT}{V-b} - \frac{aV/RT}{V^2+2bV-b^2} \tag{15.5-8}$$

The implementation of the Peng-Robinson equation of state in ANSYS FLUENT uses this expression for both phases, the particle liquid and the vapor phase. The parameters a and b are determined by the composition using a simple mixing law:

$$\begin{aligned} a &= \sum_{i=1}^{N}\sum_{j=1}^{N} x_i x_j \sqrt{a_i a_j} \\ b &= \sum_{i=1}^{N} x_i b_i \end{aligned} \tag{15.5-9}$$

where N is the number of components in the mixture. The pure component parameters can be obtained using the relationship with the Peng-Robinson constants:

$$\begin{aligned} a_i &= \left(4.57235\frac{R^2T_{c,i}^2}{p_{c,i}}\right)\left[1+(0.480+1.574\omega_i-0.176{\omega_i}^2)\left(1-\frac{T}{T_{c,i}}\right)^{1/2}\right]^2 \\ b_i &= 0.077796\frac{RT_{c,i}}{P_{c,i}} \end{aligned} \tag{15.5-10}$$

where $T_{c,i}$ is the critical temperature, $p_{c,i}$ is the critical pressure and ω_i is the accentric factor of the component i.

The fugacities of the components depend on the compressibility of the liquid and vapor phase:

$$ln\phi_i = \left(\frac{\partial(A^\gamma/RT)}{\partial N_i}\right)_{T,V,N_{j\neq i}} - lnZ \tag{15.5-11}$$

where A^{γ} is the residual Helmholtz energy, which is a function of the compressibility:

$$\frac{A^{\gamma}}{RT} = \int_{V}^{\infty} [1 - Z] \frac{dV}{V} + lnZ \qquad (15.5\text{-}12)$$

In summary, the vapor mole fraction ${x_i}^V$, the pressure p, and the compressibilities of the vapor (Z^V) and the liquid (Z^L) phase at the surface of the particle are determined from the liquid particle mole fraction of the components ${x_i}^L$ and the particle temperature T_p. The surface vapor concentrations are calculated using the following equation:

$$C_{i,S} = {x_i}^V \frac{p}{Z^V RT} \qquad (15.5\text{-}13)$$

15.6 Wall-Jet Model Theory

The direction and velocity of the droplet particles are given by the resulting momentum flux, which is a function of the impingement angle, ϕ, and Weber number. See Figure 15.6.1.

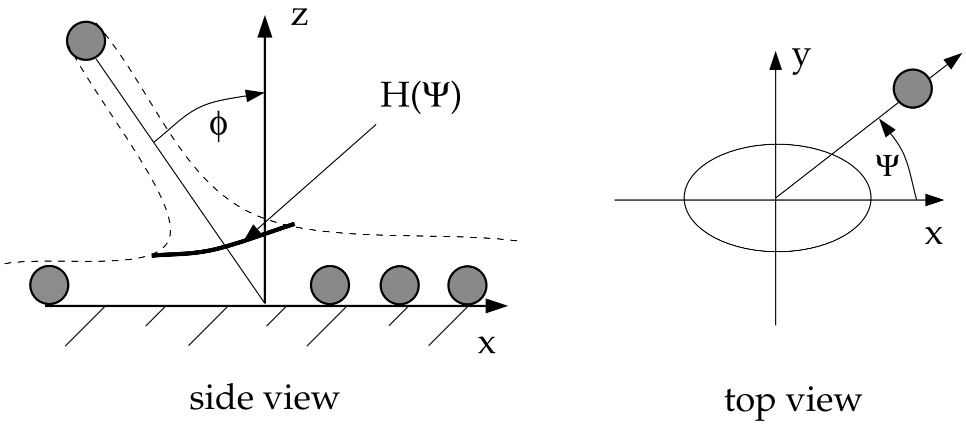


Figure 15.6.1: "Wall Jet" Boundary Condition for the Discrete Phase

The wall-jet type boundary condition assumes an analogy with an inviscid jet impacting a solid wall. Equation 15.6-1 shows the analytical solution for an axisymmetric impingement assuming an empirical function for the sheet height (H) as a function of the angle that the drop leaves the impingement (Ψ).

$$H(\Psi) = H_{\pi} e^{\beta(1-\frac{\Psi}{\pi})} \qquad (15.6\text{-}1)$$

where H_{π} is the sheet height at $\Psi = \pi$ and β is a constant determined from conservation of mass and momentum. The probability that a drop leaves the impingement point at an angle between Ψ and $\Psi + \delta\Psi$ is given by integrating the expression for $H(\Psi)$

$$\Psi = -\frac{\pi}{\beta} \ln[1 - P(1 - e^{-\beta})] \tag{15.6-2}$$

where P is a random number between 0 and 1. The expression for β is given in Naber and Reitz [244] as

$$\sin(\phi) = \frac{e^{\beta} + 1}{(e^{\beta} - 1)(1 + (\frac{\pi}{\beta})^2)} \tag{15.6-3}$$

15.7 Wall-Film Model Theory

This section is composed of the following:

- Section 15.7.1: Introduction
- Section 15.7.2: Interaction During Impact with a Boundary
- Section 15.7.3: Splashing
- Section 15.7.4: Separation Criteria
- Section 15.7.5: Conservation Equations for Wall-Film Particles

15.7.1 Introduction

ANSYS FLUENT has a specific boundary condition for simulation of internal combustion engines, called the wall-film model.

Spray-wall interaction is an important part of the mixture formation process in port fuel injected (PFI) engines. A fuel spray impinges on a surface, usually at the intake port near the intake valve, as well as at the intake valve itself, where it splashes and subsequently evaporates. The evaporated mixture is entrained into the cylinder of the engine, where it is mixed with the fresh charge and any residual gas in the cylinder. The mixture that is compressed and burned, finally exits through the exhaust port. The process repeats itself between 200 and 8000 times per minute, depending on the engine.

Several cycles worth of fuel remain in the intake tract due to film formation on the walls. This in turn makes the film important in hydrocarbon emissions for PFI engines. Additionally, film can form inside combustion chambers of direct injection (DI) types of engines. In a direct injection engine, fuel is injected directly into the combustion chamber, where the spray can impinge upon the piston if the injection event is early or late in the cycle. The modeling of the wall-film inside a DI engine, especially in diesel engines, is compounded by the presence of carbon deposits on the surfaces of the engine. This carbon deposit absorbs the liquid film as it impinges upon it. It is believed that the carbon deposits adsorb the fuel later in the cycle, however this phenomena is very complex and is not well understood.

DPM particles are used to model the wall-film. The wall-film model in ANSYS FLUENT allows a single component liquid drop to impinge upon a boundary surface and form a thin film. The model can be broken down into four major subtopics: interaction during the initial impact with a wall boundary, subsequent tracking on surfaces, calculation of film variables, and coupling to the gas phase. Figure 15.7.1 schematically shows the basic mechanisms considered for the wall-film model.

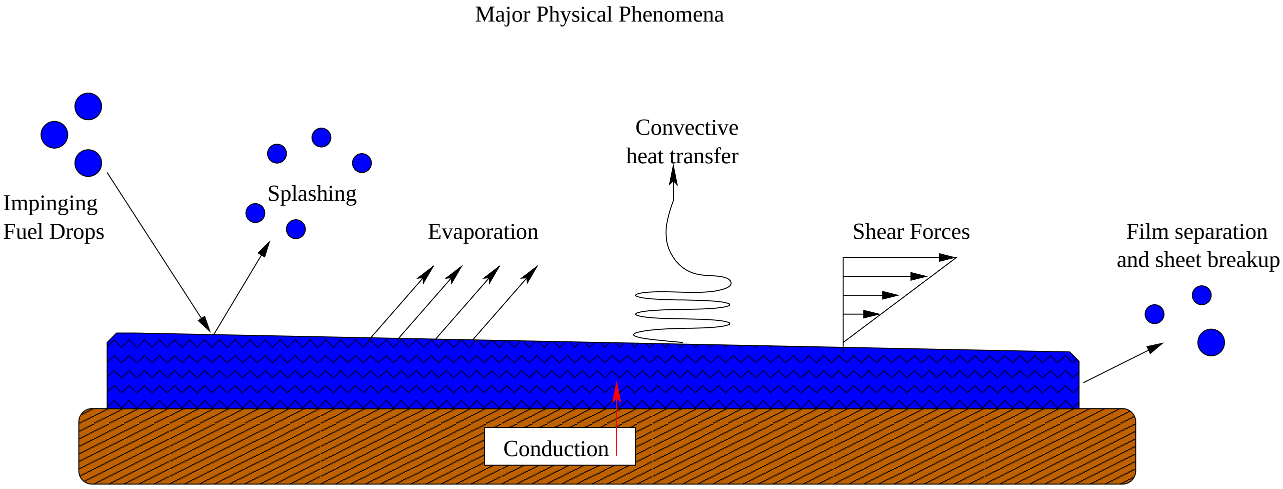


Figure 15.7.1: Mechanisms of Splashing, Momentum, Heat and Mass Transfer for the Wall-Film

The main assumptions and restrictions for the wall-film model are as follows:

- The layer is thin, less than 500 microns in thickness. This limitation is due to the assumption of a linear velocity profile in the film.
- The temperature in the film particles change relatively slowly so that an analytical integration scheme can be utilized.
- Film particles are assumed to be in direct contact with the wall surface and the heat transfer from the wall to the film takes place by conduction.
- The film temperature never exceeds the boiling temperature for the liquid.
- The simulation is transient.
- The wall-film model is not available with the Workpile Algorithm shared memory option in parallel processing.

If you wish to model a spray impacting a very hot wall, the wall-jet model may be more appropriate as the assumption in the wall-jet impingement model is that there is a vapor layer underneath the drops which keeps them from making direct contact with the boundary surface. This may be a more accurate assumption for in-cylinder diesel computations at typical operating conditions.

15.7.2 Interaction During Impact with a Boundary

The wall interaction is based on the work of Stanton [336] and O'Rourke [258], where the regimes are calculated for a drop-wall interaction based on local information. The four regimes, stick, rebound, spread, and splash are based on the impact energy and wall temperature. The following chart is helpful in showing the cutoffs.

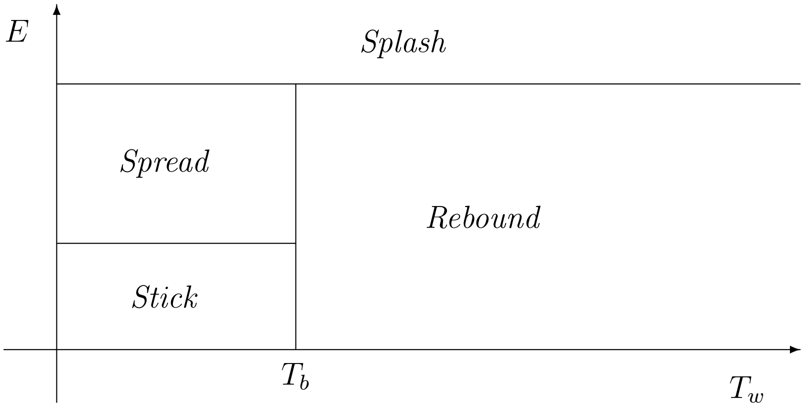


Figure 15.7.2: Simplified Decision Chart for Wall Interaction Criterion.

Below the boiling temperature of the liquid, the impinging droplet can either stick, spread or splash, while above the boiling temperature, the particle can either rebound or splash.

The criteria by which the regimes are partitioned are based on the impact energy and the boiling temperature of the liquid. The impact energy is defined by

$$E^2 = \frac{\rho V_r^2 D}{\sigma}\left(\frac{1}{\min\left(h_0/D, 1\right) + \delta_{bl}/D}\right) \tag{15.7-1}$$

where ρ is the liquid density, V_r is the relative velocity of the particle in the frame of the wall (i.e. $V_r^2 = (V_p - V_w)^2$), D is the diameter of the droplet, and σ is the surface tension of the liquid. Here, δ_{bl} is a boundary layer thickness, defined by

$$\delta_{bl} = \frac{D}{\sqrt{\mathrm{Re}}} \tag{15.7-2}$$

where the Reynolds number is defined as $Re = \rho V_r D/\mu$. By defining the energy as in Equation 15.7-1, the presence of the film on the wall suppresses the splash, but does not give unphysical results when the film height approaches zero.

The sticking regime is applied when the dimensionless energy E is less than 16, and the particle velocity is set equal to the wall velocity. In the spreading regime, the initial direction and velocity of the particle are set using the wall-jet model, where the probability of the drop having a particular direction along the surface is given by an analogy of an inviscid liquid jet with an empirically defined radial dependence for the momentum flux.

If the wall temperature is above the boiling temperature of the liquid, impingement events below a critical impact energy (E_{cr}) results in the particles rebounding from the wall. For the rebound regime, the particle rebounds with the following coefficient of restitution:

$$e = 0.993 - 1.76\Theta_I + 1.56\Theta_I^2 - 0.49\Theta_I^3 \qquad (15.7\text{-}3)$$

where Θ_I is the impingement angle as measured from the wall surface.

Splashing occurs when the impingement energy is above a critical energy threshold, defined as $E_{cr} = 57.7$. The number of splashed droplet parcels is set in the Wall boundary condition dialog box with a default number of 4, however, you can select numbers between zero and ten. The splashing algorithm follows that described by Stanton [336] and is detailed in Section 15.7.3: Splashing.

15.7.3 Splashing

If the particle impinging on the surface has a sufficiently high energy, the particle splashes and several new particles are created. You can explicitly set the number of particles created by each impact in the DPM tab of the Wall boundary condition dialog box. The number of splashed parcels may be set to an integer value between zero and ten. The properties (diameter, magnitude, and direction) of the splashed parcels are randomly sampled from the experimentally obtained distribution functions described in the following sections. Setting the number of splashed parcels to zero turns off the splashing calculation. Bear in mind that each splashed parcel can be considered a discrete sample of the distribution curves and that selecting the number of splashed drops in the Wall boundary condition dialog box does not limit the number of splashed drops, only the number of *parcels* representing those drops.

Therefore, for each splashed parcel, a different diameter is obtained by sampling a cumulative probability distribution function (CPDF), which is obtained from a Weibull distribution function and fitted to the data from Mundo, et al. [241]. The equation is

$$pdf\left(\frac{d}{D}\right) = 2\frac{d}{D^2}\exp\left[-\left(\frac{d}{D}\right)^2\right] \qquad (15.7\text{-}4)$$

and it represents the probability of finding drops of diameter d_i in a sample of splashed drops. This distribution is similar to the Nakamura-Tanasawa distribution function used by O'Rourke [258], but with the peak of the distribution function being $D = d_{max}/\sqrt{2}$. To ensure that the distribution functions produce physical results with an increasing

Weber number, the following expression for d_{max} from O'Rourke [258] is used. The peak of the splashed diameter distribution is

$$d_{max}/d_0 = \mathrm{MAX}\left(\frac{E_{crit}^2}{E^2}, \frac{6.4}{We}, 0.06\right) \tag{15.7-5}$$

where the expression for energy is given by Equation 15.7-1. Low Weber number impacts are described by the second term in Equation 15.7-5 and the peak of the minimum splashed diameter distribution is never less than 0.06 for very high energy impacts in any of the experiments analyzed by O'Rourke [258]. The Weber number in Equation 15.7-5 is defined using the relative velocity and drop diameter:

$$We = \frac{\rho V_r^2 D}{\sigma} \tag{15.7-6}$$

The cumulative probability distribution function (CPDF) is needed so that a diameter can be sampled from the experimental data. The CPDF is obtained from integrating Equation 15.7-4 to obtain

$$cpdf\left(\frac{d}{D}\right) = 1 - \exp\left[-\left(\frac{d}{D}\right)^2\right] \tag{15.7-7}$$

which is bounded by zero and one. An expression for the diameter (which is a function of D, the impingement Weber number We, and the impingement energy) is obtained by inverting Equation 15.7-7 and sampling the CPDF between zero and one. The expression for the diameter of the i^{th} splashed parcel is therefore given by,

$$d_i = D\sqrt{-\ln(1 - c_i)}$$

where c_i is the i^{th} random sample. Once the diameter of the splashed drop has been determined, the probability of finding that drop in a given sample is determined by evaluating Equation 15.7-4 at the given diameter. The number of drops per parcel can be expressed as a function of the total number of splashed drops:

$$N_i = N_{tot} pdf_i \tag{15.7-8}$$

where the pdf_i is for the i^{th} sample. The values of pdf_i are then normalized so that their sum is one. Both the number per parcel (N_i) and the total number of splashed drops (N_{tot}) is unknown, but an expression for N_{tot} can be obtained from the conservation of mass if the total splashed mass is known.

The amount of mass splashed from the surface is a quadratic function of the splashing energy, obtained from the experimental data from Mundo [241]. The splashed mass fraction y_s is given by

$$y_s = \begin{cases} 1.8\text{x}10^{-4}(E^2 - E_{crit}^2) & , E_{crit}^2 < E^2 < 7500 \\ 0.70 & , E^2 > 7500 \end{cases}$$

The authors (O'Rourke et al. [258]) noted that nearly all of the impacts for typical diesel sprays are well above the upper bound and so the splashing event nearly always ejects 70% of the mass of the impinging drop. To obtain an expression for the total number of drops, we note that overall conservation of mass requires that the sum of the total mass of the splashed parcel(s) must equal the splashed mass fraction, or

$$\frac{\rho\pi}{6} N_{tot} \sum_{n=1}^{N_{parcels}} \left(pdf_n d_n^3 \right) = y_s m_0 \qquad (15.7\text{-}9)$$

where m_0 is the total mass of the impinging parcel. The expression for the total number of splashed drops is

$$N_{tot} = \frac{y_s m_0}{\frac{\rho\pi}{6} \sum_{n=1}^{N_{parcels}} \left(pdf_n d_n^3 \right)}$$

The number of splashed drops per parcel is then determined by Equation 15.7-8 with the values of pdf_i given by Equation 15.7-4.

To calculate the velocity with which the splashed drops leave the surface, additional correlations are sampled for the normal component of the velocity. A Weibull function, fit to the data from Mundo [241], is used as the PDF for the normal component. The probability density is given by

$$pdf\left(\frac{V_{ni}}{V_{nd}}\right) = \left[\frac{b_v}{\Theta_v} \left(\frac{V_{ni}/V_{nd}}{\Theta_v} \right)^{b_v - 1} \right] \exp\left[- \left(\frac{V_{ni}/V_{nd}}{\Theta_v} \right)^{b_v} \right] \qquad (15.7\text{-}10)$$

where

$$b_v = \left\{ \begin{array}{cc} 2.1, & \Theta_I \leq 50^\circ \\ 1.10 + 0.02\Theta_I, & \Theta_I > 50^\circ \end{array} \right. \qquad (15.7\text{-}11)$$

and

$$\Theta_v = 0.158 e^{0.017\Theta_I} \qquad (15.7\text{-}12)$$

where Θ_I is the angle at which the parcel impacts the surface, or the impingement angle. The tangential component of the velocity is obtained from the expression for the reflection angle Θ_s:

$$\Theta_s = 65.4 + 0.226\Theta_l \qquad (15.7\text{-}13)$$

combined with

$$V_{ti} = \frac{V_{ni}}{\tan(\Theta_s)} \qquad (15.7\text{-}14)$$

Finally, an energy balance is performed for the new parcels so that the sum of the kinetic and surface energies of the new drops does not exceed that of the old drops. The energy balance is given by

$$\frac{1}{2} \sum_{i=1}^{N_{parcel}} \left(m_i V_i^2 \right) + \pi\sigma \sum_{i=1}^{N_{parcel}} \left(N_i d_i^2 \right) = \frac{1}{2} m_d V_d^2 \left(m_i V_i^2 \right) + \pi\sigma \left(m_d d_i^2 \right) - E_{crit}$$

where E_{crit} is the threshold energy for splashing to occur. To ensure conservation of energy, the following correction factor is computed:

$$K = \frac{\frac{1}{2} m_d V_d^2 \left(m_i V_i^2 \right) + \pi\sigma \left(m_d d_i^2 \right) - E_{crit} - \pi\sigma \sum_{i=1}^{N_{parcel}} \left(N_i d_i^2 \right)}{\frac{1}{2} \sum_{i=1}^{N_{parcel}} \left(m_i V_i^2 \right)}. \qquad (15.7\text{-}15)$$

This correction factor is needed due to the relatively small number of sampled points for the velocity of the splashed drops (see Stanton [337] for more detail). The components of the splashed parcel are multiplied by the square root of K in Equation 15.7-15 so that energy will be conserved. The normal and tangential velocity components of the splashed parcels are therefore given by

$$V'_{ni} = \sqrt{K} V_{ni} \quad \text{and} \quad V'_{ti} = \sqrt{K} V_{ti}$$

ANSYS FLUENT will limit the velocity of the splashed parcels so that they do not exceed the impact velocity of the original parcel. It is important to note that splashing events are inherently transient, so the splashing submodel is only available with unsteady tracking in ANSYS FLUENT. Splashing can also cause large increases in source terms in the cells in which it occurs, which can cause difficulty in convergence between time steps. Thus, it may be necessary to use a smaller time step during the simulation when splashing is enabled.

15.7.4 Separation Criteria

The film can separate from the wall when the stress at an edge of the film exceeds the adhesion forces holding the film to the wall. These forces are complex and highly dependent on local surface conditions. An order of magnitude analysis of a film rounding a sharp corner shows that the stresses at the edge of a film are proportional to the angle which the film negotiates. In ANSYS FLUENT, you can specify the maximum angle that the film can negotiate using a special text command. For more information, contact your ANSYS FLUENT support engineer.

15.7.5 Conservation Equations for Wall-Film Particles

Conservation equations for momentum, mass, and energy for individual parcels in the wall-film are described below. The particle-based approach for thin films was first formulated by O'Rourke [257] and most of the following derivation is based closely on that work.

Momentum

The equation for the momentum of a parcel on the film is

$$\rho h \frac{d\vec{u}_p}{dt} + h(\nabla_s p_f)_\alpha = \tau_g \vec{t_g} + \tau_w \vec{t_w} + \dot{\vec{P}}_{imp,\alpha} - \dot{M}_{imp,\alpha}\vec{u}_p + \dot{\vec{F}}_{n,\alpha} + \rho h(\vec{g} - \vec{a}_w) \quad (15.7\text{-}16)$$

where α denotes the current face on which the particle resides, h denotes the current film height at the particle location, ∇_s is the gradient operator restricted to the surface, and p_f is the pressure on the surface of the film. On the right-hand side of Equation 15.7-16, τ_g denotes the magnitude of the shear stress of the gas flow on the surface of the film, $\vec{t_g}$ is the unit vector in the direction of the relative motion of the gas and the film surface, μ_l is the liquid viscosity, and τ_w is the magnitude of the stress that the wall exerts on the film. Similarly to the expression for $\vec{t_g}$, $\vec{t_w}$ is the unit vector in the direction of the relative motion of the film and the wall. The remaining expressions on the right-hand side of Equation 15.7-16 are $\dot{\vec{P}}_{imp,\alpha}$ which denotes the impingement pressure on the film surface, $\dot{\vec{M}}_{imp,\alpha}$ is the impingement momentum source, and $\dot{\vec{F}}_{n,\alpha}$ is the force necessary to keep the film on the surface, as determined by

$$\vec{u}_p \cdot \hat{n}_\alpha = 0. \quad (15.7\text{-}17)$$

Here, $\rho h(\vec{g} - \vec{a}_w)$ is the body force term. Note that the body force term can be very significant, despite the small values of film thickness due to the very high acceleration rates seen in simulations with moving boundaries. The requirement represented by Equation 15.7-17 is explicitly enforced at each time step in ANSYS FLUENT for all particles representing the wall-film.

The term $h(\nabla_s p_f)_\alpha$ is the surface gradient of the pressure on the face, p_f. This pressure, p_f, is the sum of the fluid pressure and the impingement pressure from the drops on the face, given by

$$p_f = P_{cell} - \dot{\vec{P}}_{imp,\alpha} \cdot \hat{n} + \dot{M}_{imp,\alpha}\vec{u}_p \cdot \hat{n}$$

where the impingement mass $\dot{M}_{imp,\alpha}$ is given by

$$\dot{M}_{imp,\alpha} = \iiint \rho_l V_p \vec{v} \cdot \hat{n} f(\vec{x}_s, \vec{v}, r, T_d, t) dr d\vec{v} dT_d \quad (15.7\text{-}18)$$

and the impingement pressure is given by

$$\dot{\vec{P}}_{imp,\alpha} = \iiint \rho_l V_p \vec{v}\vec{v} \cdot \hat{n} f(\vec{x}_s, \vec{v}, r, T_d, t) dr d\vec{v} dT_d \quad (15.7\text{-}19)$$

where V_p is the volume of the drop. An approximation of the impingement mass in Equation 15.7-18 is given by

$$\dot{M}_{imp,\alpha} = \left(\sum_{n=0}^{N_s} \rho V_p \right) \Big/ A_\alpha \Delta t, \tag{15.7-20}$$

and the corresponding expression of the impingement pressure in Equation 15.7-19 is given by

$$\dot{\vec{P}}_{imp,\alpha} = \left(\sum_{n=0}^{N_i} \rho V_p (\vec{u}_p^{n+1} - \vec{u}_p^{n}) \right) \Big/ A_\alpha \Delta t. \tag{15.7-21}$$

The summation in Equation 15.7-20 is over all the drops which actually stick to the face α during the time step (N_s). The summation in Equation 15.7-21 is over *all* the particles which impinge upon the face during the same interval (N_i).

The expression for the stress that the gas exerts on the surface of the wall-film, τ_g, in Equation 15.7-16 is given by

$$\tau_g = C_f (\vec{u}_g - 2\vec{u}_p)^2 = C_f V_{rel_g}^2$$

where C_f is the skin friction coefficient and $\vec{u}_g$ is the gas velocity evaluated at the film height above the wall. The assumption made in evaluating the skin friction coefficient is that the wall shear stress from the gas is constant over the thickness of the film and the boundary layer above the film (in the normal direction from the face). The stress is tangent to the wall in the direction of the difference between the wall-film velocity and the gas velocity, so the unit vector in the direction of the velocity difference along the surface is

$$\hat{t}_g = \frac{\vec{V}_{rel_g} - (\vec{V}_{rel_g} \cdot \hat{n})\hat{n}}{|\vec{V}_{rel_g} - (\vec{V}_{rel_g} \cdot \hat{n})\hat{n}|}$$

where $\hat{n}$ is the normal face . The expression for the stress that the wall exerts on the film, τ_w, in Equation 15.7-16 is given by

$$\tau_w = -\frac{\mu_l}{h} |2\vec{u}_p - \vec{u}_w| = -\frac{\mu_l}{h} |\vec{V}_{rel_w}|$$

where μ_l is the liquid viscosity and $\vec{u}_w$ is the velocity of the wall. Here, τ_w acts in the direction of the velocity difference between the wall and the film, as given by

$$\hat{t}_w = \frac{\vec{V}_{rel_w} - (\vec{V}_{rel_w} \cdot \hat{n})\hat{n}}{|\vec{V}_{rel_w} - (\vec{V}_{rel_w} \cdot \hat{n})\hat{n}|}.$$

Note that the tangential unit vectors, $\hat{t}_g$ and $\hat{t}_w$, are independent and can point in completely different directions.

Since ANSYS FLUENT solves a particle position equation of the form

$$\frac{d\vec{u}_p}{dt} = \alpha - \beta \vec{u}_p,$$

Equation 15.7-16 must be rearranged. The film particle acceleration is then given by

$$\frac{d\vec{u}_p}{dt} = \left(\frac{C_f|V_{rel_g}|}{\rho h}\hat{t}_g - \frac{(\nabla_s p_f)_\alpha}{\rho} + \frac{\mu_l|\vec{u_w}|}{\rho h^2}\hat{t}_w + \frac{\vec{P}_{imp}}{\rho h} + \vec{g} \right) - \left(\frac{2C_f|V_{rel_g}|}{\rho h} + \frac{2\mu_l}{\rho h^2} + \frac{\dot{M}_{imp}}{\rho h} \right) \vec{u}_p. \tag{15.7-22}$$

The terms for M_{imp} and $\vec{P}_{imp}$ are used from the previous time step and the differential equations for the particle motion are solved with the existing integration routines.

Mass Transfer from the Film

The film vaporization law is applied when the film particle is above the vaporization temperature T_{vap}. A wall particle has the temperature limited by the boiling temperature T_{bp} and does not have a specific boiling law associated with the physics of film boiling.

The vaporization rate of the film is governed by gradient diffusion from the surface exposed to the gas phase. The gradient of vapor concentration between the film surface and the gas phase is

$$\dot{N}_i = B_f\left(C_{i,s} - C_{i,\infty}\right) \tag{15.7-23}$$

where $\dot{N}_i$ is the molar flux of vapor (with units of kgmol/m^2-s), B_f is the mass transfer coefficient (in m/s), and $C_{i,s}$ and $C_{i,\infty}$ are the vapor concentrations on the film surface and in the bulk gas, respectively. The units of vapor concentration are kgmol/m^3.

The vapor concentration at the surface is evaluated using the saturated vapor pressure at the film surface temperature and the bulk gas concentration is obtained from the flow field solution. The vaporization rate is sensitive to the saturated vapor pressure, similar to droplet vaporization.

The mass transfer coefficient is obtained using a Nusselt correlation for the heat transfer coefficient and replacing the Prandtl number with the Schmidt number. The equation is

$$Nu_x = \frac{B_f x}{k_f} = \begin{cases} 0.332 Re_x^{1/2} Sc^{1/3} & Re_x < 2500, 0.6 < Sc < 50 \\ 0.0296 Re_x^{4/5} Sc^{1/3} & Re_x > 2500, 0.6 < Sc < 60 \end{cases} \tag{15.7-24}$$

where the Reynolds number is based on a representative length derived from the face area. The temperature for the film surface is equal to the gas temperature, but is limited by the boiling temperature of the liquid. The particle properties are evaluated at the surface temperature when used in correlation 15.7-24.

For multicomponent vaporization, the Schmidt number based on the diffusivity of each species is used to calculate the correlation in equation 15.7-24 for each component.

The mass of the particle is decreased by

$$m_p(t + \Delta t) = m_p(t) - N_i A_p M_{w,i} \Delta t \tag{15.7-25}$$

where $M_{w,i}$ is the molecular weight of the gas phase species to which the vapor from the liquid is added. The diameter of the film particle is decreased to account for the mass loss in the individual parcel. This keeps the number of drops in the parcel constant and acts only as a place holder. When the parcel detaches from the boundary, the diameter is set to the height of the film and the number in the parcel is adjusted so that the overall mass of the parcel is conserved.

Energy Transfer from the Film

To obtain an equation for the temperature in the film, energy flux from the gas side as well as energy flux from the wall side must be considered. The assumed temperature profile in the liquid is bilinear, with the surface temperature T_s being the maximum temperature of the gas at the film height. Furthermore, the boiling point of the liquid and the wall temperature will be the maximum of the wall face temperature T_w, and will be the same boiling temperature as the liquid. An energy balance on a film particle yields

$$\frac{d}{dt}\{m_p C_p T_p\} = Q_{cond} + Q_{conv} \tag{15.7-26}$$

where Q_{cond} is the conduction from the wall, given by

$$Q_{cond} = \frac{\kappa A_p}{h}(T_w - T_p)$$

where κ is the thermal conductivity of the liquid and h is the film height at the location of the particle, as seen in Figure 15.7.3. The convection from the top surface, Q_{conv} is given by

$$Q_{conv} = h_f A_p (T_g - T_p)$$

where h_f is the film heat transfer coefficient given by Equation 15.7-24 and A_p is the area represented by a film particle, taken to be a mass weighted percentage of the face area, A_f. Contributions from the impingement terms are neglected in this formulation, as well as contributions from the gradients of the mean temperature on the edges of the film.

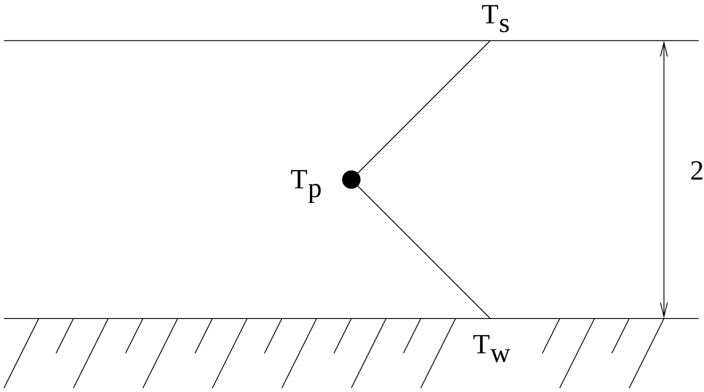


Figure 15.7.3: Assumption of a Bilinear Temperature Profile in the Film

Assuming that the temperature changes slowly for each particle in the film, the equation for the change in temperature of a non vaporizing particle can be written as

$$m_p C_p \frac{dT_p}{dt} = A_p \left(- \left[h_f + \frac{\kappa}{h} \right] T_p + hT_g + \frac{\kappa}{h} T_w \right) \tag{15.7-27}$$

As the particle trajectory is computed, ANSYS FLUENT integrates Equation 15.7-27 to obtain the particle temperature at the next time value, yielding

$$T_p(t + \Delta t) = \alpha_p + [T_p(t) - \alpha_p] e^{-\beta_p \Delta t} \tag{15.7-28}$$

where Δt is the integration time step and α_p and β_p are given by

$$\alpha_p = \frac{h_f T_g + \frac{\kappa}{h} T_w}{h_f + \frac{\kappa}{h}} \tag{15.7-29}$$

and

$$\beta_p = \frac{A_p (h_f + \frac{\kappa}{h})}{m_p C_p} \tag{15.7-30}$$

When the particle changes its mass during vaporization, an additional term is added to Equation 15.7-27 to account for the enthalpy of vaporization, which is given by

$$m_p C_p \frac{dT_p}{dt} = A_p \left(- \left[h_f + \frac{\kappa}{h} \right] T_p + hT_g + \frac{\kappa}{h} T_w \right) + \dot{m}_p h_{fg} \tag{15.7-31}$$

where h_{fg} is the latent heat of vaporization (with units of J/kg) and the expression $\dot{m}_p$ is the rate of evaporation in kg/s. This alters the expression for α_p in Equation 15.7-29 so that

$$\alpha_p = \frac{h_f T_g + \frac{\kappa}{h} T_w + \dot{m}_p h_{fg} / A_p}{h_f + \frac{\kappa}{h}} \tag{15.7-32}$$

When the wall-film model is active, the heat flux from the wall to the liquid film is subtracted from the heat flux from the wall to the gas phase. Additionally, enthalpy from vaporization of the liquid from the wall is subtracted from the cell to which the vapor mass goes. Since film boiling is modeled by limiting the liquid phase temperature to the boiling point of the material, energy in excess of that absorbed by the liquid will be put into the gas phase. When the thermal boundary conditions on the wall are set to a constant heat flux, the local temperature of the wall face is used as the thermal boundary condition for the wall-film particles.

i The wall-film model has been specifically implemented for in-cylinder flows and should be used with caution for other applications.

15.8 Particle Erosion and Accretion Theory

Particle erosion and accretion rates can be monitored at wall boundaries. The erosion rate is defined as

$$R_{\text{erosion}} = \sum_{p=1}^{N_{\text{particles}}} \frac{\dot{m}_p C(d_p) f(\alpha) v^{b(v)}}{A_{\text{face}}} \tag{15.8-1}$$

where $C(d_p)$ is a function of particle diameter, α is the impact angle of the particle path with the wall face, $f(\alpha)$ is a function of impact angle, v is the relative particle velocity, $b(v)$ is a function of relative particle velocity, and A_{face} is the area of the cell face at the wall. Default values are $C = 1.8 \times 10^{-9}$, $f = 1$, and $b = 0$.

Since C, f, and b are defined as boundary conditions at a wall, rather than properties of a material, the default values are not updated to reflect the material being used. You will need to specify appropriate values at all walls. Values of these functions for sand eroding both carbon steel and aluminum are given by Edwards et al. [86].

The erosion rate as calculated above is displayed in units of removed material/(area-time), i.e., mass flux, and can therefore be changed accordingly to the defined units in ANSYS FLUENT. The functions C and f have to be specified in consistent units to build a dimensionless group with the relative particle velocity and its exponent. To compute an erosion rate in terms of length/time (mm/year, for example) you can either define a custom field function to divide the erosion rate by the density of the wall material or include this division in the units for C and/or f. Note that the units given by ANSYS FLUENT when displaying the erosion rate are no longer valid in the latter case.

A variety of erosion models [97, 221, 85, 249, 126, 299] containing model constants [126, 85] and angle functions can be easily implemented into ANSYS FLUENT. The equations describing some of the erosion models can be modified to appear in the form of the general equation describing the erosion rate, Equation 15.8-1. For example, the Tulsa Angle Dependent Model [85] described by Equation 15.8-2

$$ER = 1559 B^{-0.59} F_s v^{1.73} f(\alpha) \tag{15.8-2}$$

can be rewritten in the form of Equation 15.8-1 with the following substitutions:

$$\begin{aligned} v^{1.73} &= v^{b(v)} \\ 1559 B^{-0.59} F_s &= C(d_p) \end{aligned}$$

where ER is the erosion rate, B is the Brinell hardness, and F_s is a particle shape coefficient.

User-defined functions can be used to describe erosion models of any form. For more complex models, such as those models with varying function angles, $f(\alpha)$, the default Erosion Model in the Wall boundary condition dialog box cannot be used. Hence, a user-defined function should be used instead. For information on how to apply user-defined functions for DPM erosion models, see Section 2.5.4: `DEFINE_DPM_EROSION` in the separate UDF Manual, or contact your support engineer for further assistance.

Note that the particle erosion and accretion rates can be displayed only when coupled calculations are enabled.

The accretion rate is defined as

$$R_{\text{accretion}} = \sum_{p=1}^{N_{\text{particles}}} \frac{\dot{m}_p}{A_{\text{face}}} \tag{15.8-3}$$

15.9 Atomizer Model Theory

In addition to the simple injection types described in Section 23.3.1: Injection Types in the separate User's Guide, ANSYS FLUENT also provides more complex injection types for sprays describing primary breakup phenomena. For most types of injections, you will need to provide the initial diameter, position, and velocity of the particles. For sprays, however, there are models available to predict the droplet size and velocity distributions.

All of the atomization models use physical atomizer parameters, such as orifice diameter and mass flow rate, to calculate initial droplet size, velocity, and position.

For realistic atomizer simulations, the droplets must be randomly distributed, both spatially through a dispersion angle and in their time of release. For other types of injections in ANSYS FLUENT (nonatomizer), all of the droplets are released along fixed trajectories at the beginning of the time step. The atomizer models use stochastic trajectory selection and staggering to attain a random distribution. Further information on staggering can be found in section Section 23.2.8: Staggering of Particles in Space and Time in the separate User's Guide.

Stochastic trajectory selection is the random dispersion of initial droplet directions. All of the atomizer models provide an initial dispersion angle, and the stochastic trajectory selection picks an initial direction within this angle. This approach improves the accuracy of the results for spray-dominated flows. The droplets will be more evenly spread among the computational cells near the atomizer, which improves the coupling to the gas phase by spreading drag more smoothly over the cells near the injection. Source terms in the energy and species conservation equations are also more evenly distributed among neighboring cells, improving solution convergence.

Five atomizer models are available in ANSYS FLUENT to predict the spray characteristics from knowledge of global parameters such as nozzle type and liquid flow rate. You can choose them as injection types and define the associated parameters in the Set Injection Properties dialog box, as described in Section 23.3.1: Injection Types in the separate User's Guide. Details about the atomizer models are provided below.

Information is organized into the following subsections:

- Section 15.9.1: The Plain-Orifice Atomizer Model
- Section 15.9.2: The Pressure-Swirl Atomizer Model
- Section 15.9.3: The Air-Blast/Air-Assist Atomizer Model
- Section 15.9.4: The Flat-Fan Atomizer Model
- Section 15.9.5: The Effervescent Atomizer Model

15.9.1 The Plain-Orifice Atomizer Model

The plain-orifice is the most common type of atomizer and the most simply made. However there is nothing simple about the physics of the internal nozzle flow and the external atomization. In the plain-orifice atomizer model in ANSYS FLUENT, the liquid is accelerated through a nozzle, forms a liquid jet and then breaks up to form droplets. This apparently simple process is dauntingly complex. The plain orifice may operate in three different regimes: single-phase, cavitating and flipped [330]. The transition between regimes is abrupt, producing dramatically different sprays. The internal regime determines the velocity at the orifice exit, as well as the initial droplet size and the angle of droplet dispersion. Diagrams of each case are shown in Figures 15.9.1, 15.9.2, and 15.9.3.

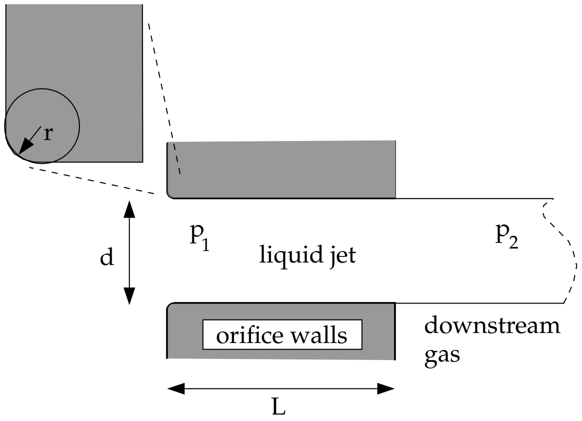


Figure 15.9.1: Single-Phase Nozzle Flow (Liquid Completely Fills the Orifice)

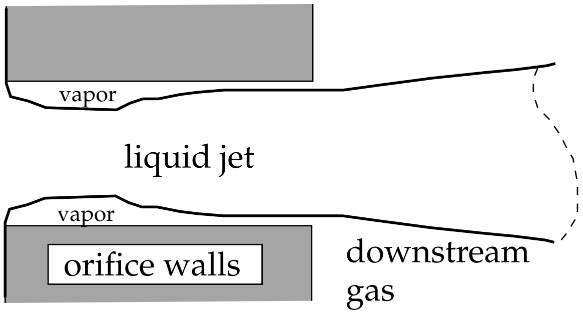


Figure 15.9.2: Cavitating Nozzle Flow (Vapor Pockets Form Just After the Inlet Corners)

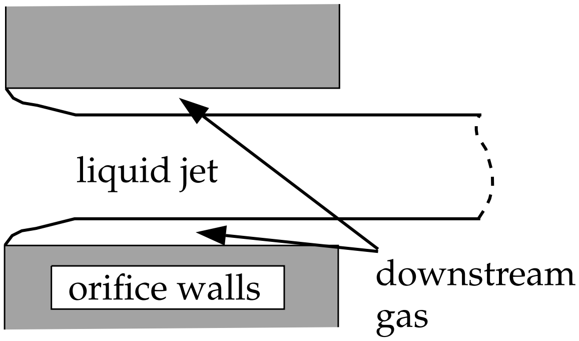


Figure 15.9.3: Flipped Nozzle Flow (Downstream Gas Surrounds the Liquid Jet Inside the Nozzle)

Internal Nozzle State

To accurately predict the spray characteristics, the plain-orifice model in ANSYS FLUENT must identify the correct state of the internal nozzle flow because the nozzle state has a tremendous effect on the external spray. Unfortunately, there is no established theory for determining the nozzle state. One must rely on empirical models obtained from experimental data. ANSYS FLUENT uses several dimensionless parameters to determine the internal flow regime for the plain-orifice atomizer model. These parameters and the decision-making process are summarized below.

A list of the parameters that control internal nozzle flow is given in Table 15.9.1. These parameters may be combined to form nondimensional characteristic lengths such as r/d and L/d, as well as nondimensional groups like the Reynolds number based on hydraulic "head" (Re_h) and the cavitation parameter (K).

Table 15.9.1: List of Governing Parameters for Internal Nozzle Flow

nozzle diameter	d
nozzle length	L
radius of curvature of the inlet corner	r
upstream pressure	p_1
downstream pressure	p_2
viscosity	μ
liquid density	ρ_l
vapor pressure	p_v

$$\mathrm{Re}_h = \frac{d\rho_l}{\mu}\sqrt{\frac{2(p_1 - p_2)}{\rho_l}} \tag{15.9-1}$$

$$K = \frac{p_1 - p_v}{p_1 - p_2} \tag{15.9-2}$$

The liquid flow often contracts in the nozzle, as can be seen in Figures 15.9.2 and 15.9.3. Nurick [252] found it helpful to use a coefficient of contraction (C_c) to represent the reduction in the cross-sectional area of the liquid jet. The coefficient of contraction is defined as the area of the stream of contracting liquid over the total cross-sectional area of the nozzle. ANSYS FLUENT uses Nurick's fit for the coefficient of contraction:

$$C_c = \frac{1}{\sqrt{\frac{1}{C_{\mathrm{ct}}^2} - \frac{11.4r}{d}}} \tag{15.9-3}$$

Here, $C_{\rm ct}$ is a theoretical constant equal to 0.611, which comes from potential flow analysis of flipped nozzles.

Coefficient of Discharge

Another important parameter for describing the performance of nozzles is the coefficient of discharge (C_d). The coefficient of discharge is the ratio of the mass flow rate through the nozzle to the theoretical maximum mass flow rate:

$$C_d = \frac{\dot{m}_{\rm eff}}{A\sqrt{2\rho_l(p_1 - p_2)}} \tag{15.9-4}$$

where $\dot{m}_{\rm eff}$ is the *effective* mass flow rate of the nozzle, defined by

$$\dot{m}_{\rm eff} = \frac{2\pi\dot{m}}{\Delta\phi} \tag{15.9-5}$$

Here, $\dot{m}$ is the mass flow rate specified in the user interface, and $\Delta\phi$ is the difference between the azimuthal stop angle and the azimuthal start angle

$$\Delta\phi = \phi_{\rm stop} - \phi_{\rm start} \tag{15.9-6}$$

as input by you (see Section 23.3.7: Point Properties for Plain-Orifice Atomizer Injections in the separate User's Guide). Note that the mass flow rate that you input should be for the appropriate start and stop angles, in other words the correct mass flow rate for the sector being modeled. Note also that for $\Delta\phi$ of 2π, the effective mass flow rate is identical to the mass flow rate in the interface.

The cavitation number (K in Equation 15.9-2) is an essential parameter for predicting the inception of cavitation. The inception of cavitation is known to occur at a value of $K_{\rm incep} \approx 1.9$ for short, sharp-edged nozzles. However, to include the effects of inlet rounding and viscosity, an empirical relationship is used:

$$K_{\rm incep} = 1.9\left(1 - \frac{r}{d}\right)^2 - \frac{1000}{{\rm Re}_h} \tag{15.9-7}$$

Similarly, a critical value of K where flip occurs is given by

$$K_{\rm crit} = 1 + \frac{1}{\left(1 + \frac{L}{4d}\right)\left(1 + \frac{2000}{{\rm Re}_h}\right)e^{70r/d}} \tag{15.9-8}$$

If r/d is greater than 0.05, then flip is deemed impossible and $K_{\rm crit}$ is set to 1.0.

The cavitation number, K, is compared to the values of K_{incep} and K_{crit} to identify the nozzle state. The decision tree is shown in Figure 15.9.4. Depending on the state of the nozzle, a unique closure is chosen for the above equations.

For a single-phase nozzle $(K > K_{\mathrm{incep}}, K \geq K_{\mathrm{crit}})$ [193], the coefficient of discharge is given by

$$C_d = \frac{1}{\frac{1}{C_{\mathrm{du}}} + 20\frac{(1+2.25L/d)}{\mathrm{Re}_h}} \tag{15.9-9}$$

where C_{du} is the ultimate discharge coefficient, and is defined as

$$C_{\mathrm{du}} = 0.827 - 0.0085\frac{L}{d} \tag{15.9-10}$$

For a cavitating nozzle $(K_{\mathrm{crit}} \leq K \leq K_{\mathrm{incep}})$ [252] the coefficient of discharge is determined from

$$C_d = C_c\sqrt{K} \tag{15.9-11}$$

For a flipped nozzle $(K < K_{\mathrm{crit}})$ [252],

$$C_d = C_{\mathrm{ct}} = 0.611 \tag{15.9-12}$$

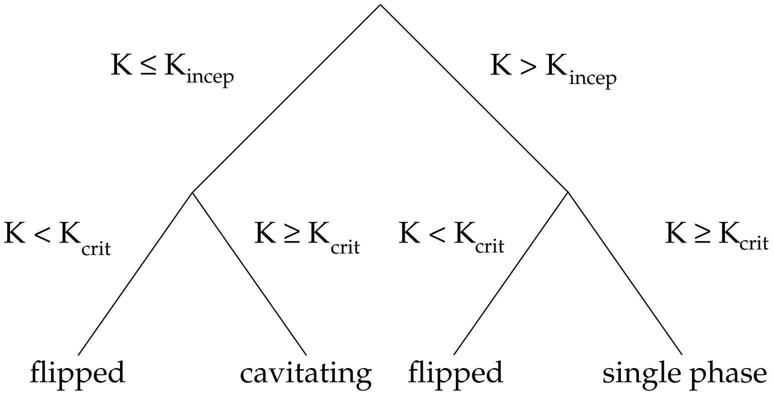


Figure 15.9.4: Decision Tree for the State of the Cavitating Nozzle

All of the nozzle flow equations are solved iteratively, along with the appropriate relationship for coefficient of discharge as given by the nozzle state. The nozzle state may change as the upstream or downstream pressures change. Once the nozzle state is determined, the exit velocity is calculated, and appropriate correlations for spray angle and initial droplet size distribution are determined.

Exit Velocity

For a single-phase nozzle, the estimate of exit velocity (u) comes from the conservation of mass and the assumption of a uniform exit velocity:

$$u = \frac{\dot{m}_{\rm eff}}{\rho_l A} \tag{15.9-13}$$

For the cavitating nozzle, Schmidt and Corradini [306] have shown that the uniform exit velocity is not accurate. Instead, they derived an expression for a higher velocity over a reduced area:

$$u = \frac{2C_c p_1 - p_2 + (1 - 2C_c)p_v}{C_c\sqrt{2\rho_l(p_1 - p_v)}} \tag{15.9-14}$$

This analytical relation is used for cavitating nozzles in ANSYS FLUENT. For the case of flipped nozzles, the exit velocity is found from the conservation of mass and the value of the reduced flow area:

$$u = \frac{\dot{m}_{\rm eff}}{\rho_l C_{\rm ct} A} \tag{15.9-15}$$

Spray Angle

The correlation for the spray angle (θ) comes from the work of Ranz [283]:

$$\frac{\theta}{2} = \begin{cases} \tan^{-1}\left[\frac{4\pi}{C_A}\sqrt{\frac{\rho_g}{\rho_l}}\frac{\sqrt{3}}{6}\right] & \text{single phase, cavitating} \\ 0.01 & \text{flipped} \end{cases} \tag{15.9-16}$$

The spray angle for both single-phase and cavitating nozzles depends on the ratio of the gas and liquid densities and also the parameter C_A. For flipped nozzles, the spray angle has a constant value.

The parameter C_A, which you must specify, is thought to be a constant for a given nozzle geometry. The larger the value, the narrower the spray. Reitz [288] suggests the following correlation for C_A:

$$C_A = 3 + \frac{L}{3.6d} \tag{15.9-17}$$

The spray angle is sensitive to the internal flow regime of the nozzle. Hence, you may wish to choose smaller values of C_A for cavitating nozzles than for single-phase nozzles. Typical values range from 4.0 to 6.0. The spray angle for flipped nozzles is a small, arbitrary value that represents the lack of any turbulence or initial disturbance from the nozzle.

Droplet Diameter Distribution

One of the basic characteristics of an injection is the distribution of drop size. For an atomizer, the droplet diameter distribution is closely related to the nozzle state. ANSYS FLUENT's spray models use a two-parameter Rosin-Rammler distribution, characterized by the most probable droplet size and a spread parameter. The most probable droplet size, d_0 is obtained in ANSYS FLUENT from the Sauter mean diameter, d_{32} [186]. For more information about the Rosin-Rammler size distribution, see Section 23.3.13: Using the Rosin-Rammler Diameter Distribution Method in the separate User's Guide.

For single-phase nozzle flows, the correlation of Wu et al. [383] is used to calculate d_{32} and relate the initial drop size to the estimated turbulence quantities of the liquid jet:

$$d_{32} = 133.0 \lambda \mathrm{We}^{-0.74}, \tag{15.9-18}$$

where $\lambda = d/8$, λ is the radial integral length scale at the jet exit based upon fully-developed turbulent pipe flow, and We is the Weber number, defined as

$$\mathrm{We} \equiv \frac{\rho_l u^2 \lambda}{\sigma}. \tag{15.9-19}$$

Here, σ is the droplet surface tension. For a more detailed discussion of droplet surface tension and the Weber number, see Section 15.10: Secondary Breakup Model Theory. For more information about mean particle diameters, see Section 23.7.8: Summary Reporting of Current Particles in the separate User's Guide.

For cavitating nozzles, ANSYS FLUENT uses a slight modification of Equation 15.9-18. The initial jet diameter used in Wu's correlation, d, is calculated from the effective area of the cavitating orifice exit, and thus represents the effective diameter of the exiting liquid jet, d_{eff}. For an explanation of effective area of cavitating nozzles, please see Schmidt and Corradini [306].

The length scale for a cavitating nozzle is $\lambda = d_{\mathrm{eff}}/8$, where

$$d_{\mathrm{eff}} = \sqrt{\frac{4\dot{m}_{\mathrm{eff}}}{\pi \rho_l u}}. \tag{15.9-20}$$

For the case of the flipped nozzle, the initial droplet diameter is set to the diameter of the liquid jet:

$$d_0 = d\sqrt{C_{\rm ct}} \tag{15.9-21}$$

where d_0 is defined as the most probable diameter.

The second parameter required to specify the droplet size distribution is the spread parameter, s. The values for the spread parameter are chosen from past modeling experience and from a review of experimental observations. Table 15.9.2 lists the values of s for the three nozzle states. The larger the value of the spread parameter, the narrower the droplet size distribution.

Table 15.9.2: Values of Spread Parameter for Different Nozzle States

State	**Spread Parameter**
single phase	3.5
cavitating	1.5
flipped	∞

Since the correlations of Wu et al. provide the Sauter mean diameter, d_{32}, these are converted to the most probable diameter, d_0. Lefebvre [186] gives the most general relationship between the Sauter mean diameter and most probable diameter for a Rosin-Rammler distribution. The simplified version for s=3.5 is as follows:

$$d_0 = 1.2726 d_{32} \left(1 - \frac{1}{s}\right)^{1/s} \tag{15.9-22}$$

At this point, the droplet size, velocity, and spray angle have been determined and the initialization of the injections is complete.

15.9.2 The Pressure-Swirl Atomizer Model

Another important type of atomizer is the pressure-swirl atomizer, sometimes referred to by the gas-turbine community as a simplex atomizer. This type of atomizer accelerates the liquid through nozzles known as swirl ports into a central swirl chamber. The swirling liquid pushes against the walls of the swirl chamber and develops a hollow air core. It then emerges from the orifice as a thinning sheet, which is unstable, breaking up into ligaments and droplets. The pressure-swirl atomizer is very widely used for liquid-fuel combustion in gas turbines, oil furnaces, and direct-injection spark-ignited automobile engines. The transition from internal injector flow to fully-developed spray can be divided into three steps: film formation, sheet breakup, and atomization. A sketch of how this process is thought to occur is shown in Figure 15.9.5.

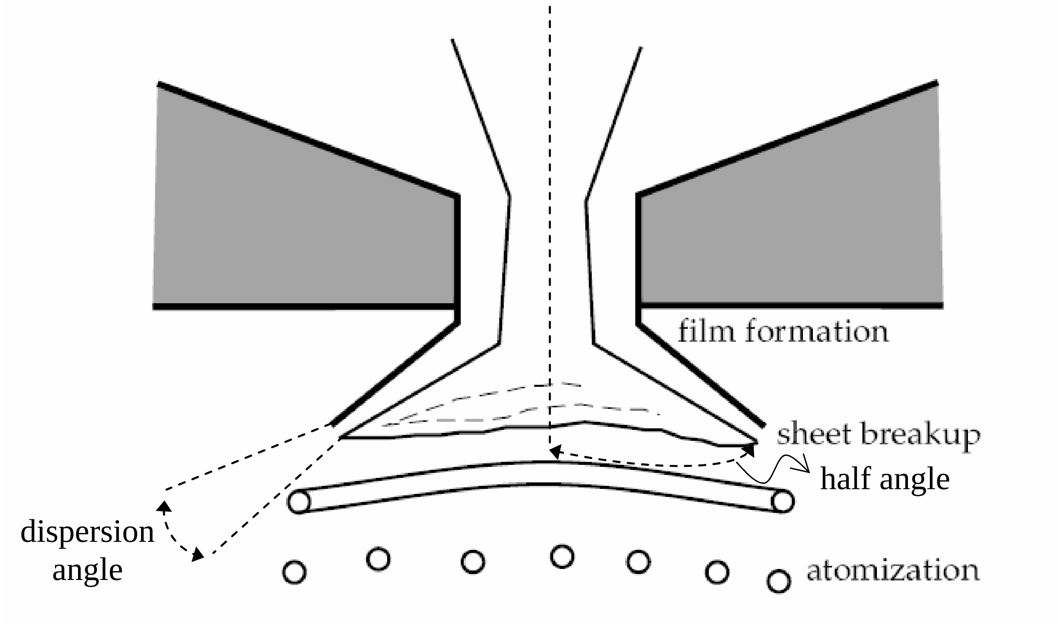


Figure 15.9.5: Theoretical Progression from the Internal Atomizer Flow to the External Spray

The interaction between the air and the sheet is not well understood. It is generally accepted that an aerodynamic instability causes the sheet to break up. The mathematical analysis below assumes that Kelvin-Helmholtz waves grow on the sheet and eventually break the liquid into ligaments. It is then assumed that the ligaments break up into droplets due to varicose instability. Once the liquid droplets are formed, the spray evolution is determined by drag, collision, coalescence, and secondary breakup.

The pressure-swirl atomizer model used in ANSYS FLUENT is called the Linearized Instability Sheet Atomization (LISA) model of Schmidt et al. [308]. The LISA model is divided into two stages:

1. film formation
2. sheet breakup and atomization

Both parts of the model are described below.

Film Formation

The centrifugal motion of the liquid within the injector creates an air core surrounded by a liquid film. The thickness of this film, t, is related to the mass flow rate by

$$\dot{m}_{\text{eff}} = \pi \rho u t (d_{\text{inj}} - t) \tag{15.9-23}$$

where d_{inj} is the injector exit diameter, and $\dot{m}_{\text{eff}}$ is the effective mass flow rate, which is defined by Equation 15.9-5 . The other unknown in Equation 15.9-23 is u, the axial component of velocity at the injector exit. This quantity depends on internal details of the injector and is difficult to calculate from first principles. Instead, the approach of Han et al. [122] is used. The total velocity is assumed to be related to the injector pressure by

$$U = k_v \sqrt{\frac{2\Delta p}{\rho_l}} \tag{15.9-24}$$

where k_v is the velocity coefficient. Lefebvre [186] has noted that k_v is a function of the injector design and injection pressure. If the swirl ports are treated as nozzles and if it is assumed that the dominant portion of the pressure drop occurs at those ports, k_v is the expression for the discharge coefficient (C_d). For single-phase nozzles with sharp inlet corners and L/d ratios of 4, a typical C_d value is 0.78 or less [193]. If the nozzles are cavitating, the value of C_d may be as low as 0.61. Hence, 0.78 should be a practical upper bound for k_v. Reducing k_v by 10% to 0.7 approximates the effect of other momentum losses on the discharge coefficient.

Physical limits on k_v require that it be less than unity from conservation of energy, yet be large enough to permit sufficient mass flow. To guarantee that the size of the air core is non-negative, the following expression is used for k_v:

$$k_v = \max\left[0.7, \frac{4\dot{m}_{\text{eff}}}{d_0^2 \rho_l \cos\theta} \sqrt{\frac{\rho_l}{2\Delta p}}\right] \tag{15.9-25}$$

Assuming that Δp is known, Equation 15.9-24 can be used to find U. Once U is determined, u is found from

$$u = U \cos\theta \tag{15.9-26}$$

where θ is the spray angle, which is assumed to be known. At this point, the thickness and axial component of the liquid film are known at the injector exit. The tangential component of velocity ($w = U \sin\theta$) is assumed to be equal to the radial velocity component of the liquid sheet downstream of the nozzle exit. The axial component of velocity is assumed to remain constant.

Sheet Breakup and Atomization

The pressure-swirl atomizer model includes the effects of the surrounding gas, liquid viscosity, and surface tension on the breakup of the liquid sheet. Details of the theoretical development of the model are given in Senecal et al. [310] and are only briefly presented here. For a more robust implementation, the gas-phase velocity is neglected in calculating the relative liquid-gas velocity and is instead set by you. This avoids having the injector parameters depend too heavily on the usually under-resolved gas-phase velocity field very near the injection location.

The model assumes that a two-dimensional, viscous, incompressible liquid sheet of thickness $2h$ moves with velocity U through a quiescent, inviscid, incompressible gas medium. The liquid and gas have densities of ρ_l and ρ_g, respectively, and the viscosity of the liquid is μ_l. A coordinate system is used that moves with the sheet, and a spectrum of infinitesimal wavy disturbances of the form

$$\eta = \eta_0 e^{ikx+\omega t} \tag{15.9-27}$$

is imposed on the initially steady motion. The spectrum of disturbances results in fluctuating velocities and pressures for both the liquid and the gas. In Equation 15.9-27, η_0 is the initial wave amplitude, $k = 2\pi/\lambda$ is the wave number, and $\omega = \omega_r + i\omega_i$ is the complex growth rate. The most unstable disturbance has the largest value of ω_r, denoted here by Ω, and is assumed to be responsible for sheet breakup. Thus, it is desired to obtain a dispersion relation $\omega = \omega(k)$ from which the most unstable disturbance can be calculated as a function of wave number.

Squire [335], Li and Tankin [192], and Hagerty and Shea [118] have shown that two solutions, or modes, exist that satisfy the governing equations subject to the boundary conditions at the upper and lower interfaces. The first solution, called the sinuous mode, has waves at the upper and lower interfaces in phase. The second solution is called the varicose mode which has the waves at the upper and lower interfaces π radians out of phase. It has been shown by numerous authors (e.g., Senecal et. al. [310]) that the sinuous mode dominates the growth of varicose waves for low velocities and low gas-to-liquid density ratios. In addition, it can be shown that the sinuous and varicose modes become indistinguishable for high-velocity flows. As a result, the atomization model in ANSYS FLUENT is based upon the growth of sinuous waves on the liquid sheet.

As derived in Senecal et al. [310], the dispersion relation for the sinuous mode is given by

$$\begin{gathered}\omega^2[\tanh(kh)+Q]+[4\nu_l k^2\tanh(kh)+2iQkU]+\\ 4\nu_l k^4\tanh(kh)-4\nu_l^2k^3\ell\tanh(\ell h)-QU^2k^2+\frac{\sigma k^3}{\rho_l}=0\end{gathered}\tag{15.9-28}$$

where $Q=\rho_g/\rho_l$ and $\ell^2=k^2+\omega/\nu_l$.

Above a critical Weber number of $\mathrm{We}_g=27/16$ (based on the liquid velocity, gas density, and sheet half-thickness), the fastest-growing waves are short. For $\mathrm{We}_g<27/16$, the wavelengths are long compared to the sheet thickness. The speed of modern high pressure fuel injection systems is high enough to ensure that the film Weber number is well above this critical limit.

An order-of-magnitude analysis using typical values shows that the terms of second order in viscosity can be neglected in comparison to the other terms in Equation 15.9-28. Using this assumption, Equation 15.9-28 reduces to

$$\begin{gathered}\omega_r=\frac{1}{\tanh(kh)+Q}\Bigg\{-2\nu_l k^2\tanh(kh)+\\ \left.\sqrt{4\nu_l^2k^4\tanh^2(kh)-Q^2U^2k^2-[\tanh(kh)+Q]\left[-QU^2k^2+\frac{\sigma k^3}{\rho_l}\right]}\right\}\end{gathered}\tag{15.9-29}$$

For waves that are long compared with the sheet thickness, a mechanism of sheet disintegration proposed by Dombrowski and Johns [74] is adopted. For long waves, ligaments are assumed to form from the sheet breakup process once the unstable waves reach a critical amplitude. If the surface disturbance has reached a value of η_b at breakup, a breakup time, τ, can be evaluated:

$$\eta_b=\eta_0 e^{\Omega\tau}\Rightarrow\frac{1}{\Omega}\ln\left(\frac{\eta_b}{\eta_0}\right)\tag{15.9-30}$$

where Ω, the maximum growth rate, is found by numerically maximizing Equation 15.9-29 as a function of k. The maximum is found using a binary search that checks the sign of the derivative. The sheet breaks up and ligaments will be formed at a length given by

$$L_b = U\tau = \frac{U}{\Omega} \ln\left(\frac{\eta_b}{\eta_0}\right) \qquad (15.9\text{-}31)$$

where the quantity $\ln(\frac{\eta_b}{\eta_0})$ is an empirical *sheet* constant that you must specify. The default value of 12 was obtained theoretically by Weber [369] for liquid jets. Dombrowski and Hooper [73] showed that a value of 12 for the sheet constant agreed favorably with experimental sheet breakup lengths over a range of Weber numbers from 2 to 200.

The diameter of the ligaments formed at the point of breakup can be obtained from a mass balance. If it is assumed that the ligaments are formed from tears in the sheet twice per wavelength, the resulting diameter is given by

$$d_L = \sqrt{\frac{8h}{K_s}} \qquad (15.9\text{-}32)$$

where K_s is the wave number corresponding to the maximum growth rate, Ω. The ligament diameter depends on the sheet thickness, which is a function of the breakup length. The film thickness is calculated from the breakup length and the radial distance from the center line to the mid-line of the sheet at the atomizer exit, r_0:

$$h_{\mathrm{end}} = \frac{r_0 h_0}{r_0 + L_b \sin\left(\frac{\theta}{2}\right)} \qquad (15.9\text{-}33)$$

This mechanism is not used for waves that are short compared to the sheet thickness. For short waves, the ligament diameter is assumed to be linearly proportional to the wavelength that breaks up the sheet,

$$d_L = \frac{2\pi C_L}{K_s} \qquad (15.9\text{-}34)$$

where C_L, or the ligament constant, is equal to 0.5 by default.

In either the long-wave or the short-wave case, the breakup from ligaments to droplets is assumed to behave according to Weber's [369] analysis for capillary instability.

$$d_0 = 1.88 d_L (1 + 3\mathrm{Oh})^{1/6} \qquad (15.9\text{-}35)$$

Here, Oh is the Ohnesorge number which is a combination of the Reynolds number and the Weber number (see Section 15.10.2: Jet Stability Analysis for more details about Oh). Once d_0 has been determined from Equation 15.9-35, it is assumed that this droplet diameter is the most probable droplet size of a Rosin-Rammler distribution with a spread parameter of 3.5 and a default dispersion angle of 6° (which can be modified in the GUI). The choice of spread parameter and dispersion angle is based on past modeling experience [307]. It is important to note that the spray cone angle must be specified by you when using this model.

15.9.3 The Air-Blast/Air-Assist Atomizer Model

In order to accelerate the breakup of liquid sheets from an atomizer, an additional air stream is often directed through the atomizer. The liquid is formed into a sheet by a nozzle, and air is then directed against the sheet to promote atomization. This technique is called air-assisted atomization or air-blast atomization, depending on the quantity of air and its velocity. The addition of the external air stream past the sheet produces smaller droplets than without the air. Though the exact mechanism for this enhanced performance is not completely understood, it is thought that the assisting air may accelerate the sheet instability. The air may also help disperse the droplets, preventing collisions between them. Air-assisted atomization is used in many of the same applications as pressure-swirl atomization, where especially fine atomization is required.

ANSYS FLUENT's air-blast atomization model is a variation of the pressure-swirl model. One important difference between them is that the sheet thickness is set directly in the air-blast atomizer model. This input is necessary because of the variety of sheet formation mechanisms used in air-blast atomizers. Hence the air-blast atomizer model does not contain the sheet formation equations that were included in the pressure-swirl atomizer model (Equations 15.9-23–15.9-26). You will also specify the maximum relative velocity that is produced by the sheet and air. Though this quantity could be calculated, specifying a value relieves you from the necessity of finely resolving the atomizer internal flow. This feature is convenient for simulations in large domains, where the atomizer is very small by comparison.

An additional difference is that the air-blast atomizer model assumes that the sheet breakup is always due to short waves. This assumption is a consequence of the greater sheet thickness commonly found in air-blast atomizers. Hence the ligament diameter is assumed to be linearly proportional to the wavelength of the fastest-growing wave on the sheet, and is calculated from Equation 15.9-34.

Other inputs are similar to the pressure-swirl model – you must provide the mass flow rate and spray angle. The angle in the case of the air-blast atomizer is the initial trajectory of the film as it leaves the end of the orifice. The value of the angle is negative if the initial film trajectory is inward, towards the centerline. Specification of the inner and outer diameters of the film at the atomizer exit are also required, in addition to the dispersion angle whose default value is 6° (which can be modified in the GUI).

Since the air-blast atomizer model does not include internal gas flows, you must create the air streams surrounding the injector as boundary conditions within the ANSYS FLUENT simulation. These streams are ordinary continuous-phase flows and require no special treatment.

15.9.4 The Flat-Fan Atomizer Model

The flat-fan atomizer is very similar to the pressure-swirl atomizer, but it makes a flat sheet and does not use swirl. The liquid emerges from a wide, thin orifice as a flat liquid sheet that breaks up into droplets. The primary atomization process is thought to be similar to the pressure-swirl atomizer. Some researchers believe that flat-fan atomization, because of jet impingement, is very similar to the atomization of a flat sheet. The flat-fan model could serve doubly for this application.

The flat-fan atomizer is available only for 3D models. An image of the three-dimensional flat fan is shown in Figure 15.9.6. The model assumes that the fan originates from a virtual origin. You need to provide the location of this origin, which is the intersection of the lines that mark the sides of the fan as well as the location of the center point of the arc from which the fan originates. ANSYS FLUENT will find the vector that points from the origin to the center point in order to determine the direction of the injection. You also need to provide the half-angle of the fan arc, the width of the orifice (in the normal direction) and the mass flow rate of the liquid to use the flat-fan atomizer model.

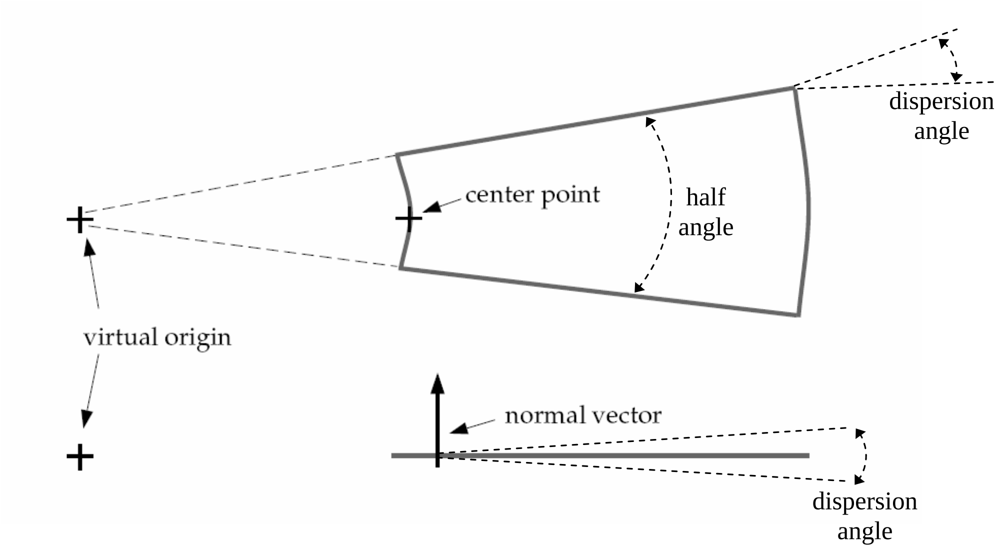


Figure 15.9.6: Flat Fan Viewed from Above and from the Side

The breakup of the flat fan is calculated very much like the breakup of the sheet in the pressure-swirl atomizer. The sheet breaks up into ligaments which then form individual droplets. The only difference is that for short waves, the flat fan sheet is assumed to form ligaments at half-wavelength intervals. Hence the ligament diameter for short waves is given by

$$d_L = \sqrt{\frac{16h}{K_s}} \tag{15.9-36}$$

In this case, d_L in Equation 15.9-36 is taken to be the most probable diameter, with a Rosin-Rammler spread parameter of 3.5 and a default dispersion angle of 6°. This angle can be set in the Set Injection Properties dialog box. In all other respects, the flat-fan atomizer model is like the sheet breakup portion of the pressure-swirl atomizer.

15.9.5 The Effervescent Atomizer Model

Effervescent atomization is the injection of liquid infused with a super-heated (with respect to downstream conditions) liquid or propellant. As the volatile liquid exits the nozzle, it rapidly changes phase. This phase change quickly breaks up the stream into small droplets with a wide dispersion angle. The model also applies to cases where a very hot liquid is discharged.

Since the physics of effervescence is not well understood, the model must rely on rough empirical fits. The photographs of Reitz and Bracco [288] provide some insights. These photographs show a dense liquid core to the spray, surrounded by a wide shroud of smaller droplets.

The initial velocity of the droplets is computed from conservation of mass, assuming the exiting jet has a cross-sectional area that is C_{ct} times the nozzle area, where C_{ct} is a fixed constant, equal to `0.611` as specified in Equations 15.9-3 and 15.9-12.

$$u = \frac{\dot{m}_{\mathrm{eff}}}{\rho_l C_{\mathrm{ct}} A} \tag{15.9-37}$$

The maximum droplet diameter is set to the effective diameter of the exiting jet:

$$d_{\max} = d\sqrt{C_{\mathrm{ct}}} \tag{15.9-38}$$

The droplet size is then sampled from a Rosin-Rammler distribution with a spread parameter of 4.0. (See Section 23.3.13: Using the Rosin-Rammler Diameter Distribution Method in the separate User's Guide for details on the Rosin-Rammler distribution.) The most probable droplet size depends on the angle, θ, between the droplet's stochastic trajectory and the injection direction:

$$d_0 = d_{\text{max}} e^{-(\theta/\Theta_s)^2} \tag{15.9-39}$$

The dispersion angle multiplier, Θ_s, is computed from the quality, x, and the specified value for the dispersion constant, C_{eff}:

$$x = \frac{\dot{m}_{\text{vapor}}}{(\dot{m}_{\text{vapor}} + \dot{m}_{\text{liquid}})} \tag{15.9-40}$$

$$\Theta_s = \frac{x}{C_{\text{eff}}} \tag{15.9-41}$$

This technique creates a spray with large droplets in the central core and a shroud of smaller surrounding droplets. The droplet temperature is initialized to 0.99 times the saturation temperature, such that the temperature of the droplet is close to boiling. To complete the model, the flashing vapor must also be included in the calculation. This vapor is part of the continuous phase and not part of the discrete phase model. You must create an inlet at the point of injection when you specify boundary conditions for the continuous phase. When the effervescent atomizer model is selected, you will need to specify the nozzle diameter, mass flow rate, mixture quality, saturation temperature of the volatile substance, spray half-angle and dispersion constant in addition to specifying the position and direction of the injector.

15.10 Secondary Breakup Model Theory

ANSYS FLUENT offers two droplet breakup models: the Taylor analogy breakup (TAB) model and the wave model. The TAB model is recommended for low-Weber-number injections and is well suited for low-speed sprays into a standard atmosphere. For Weber numbers greater than 100, the wave model is more applicable. The wave model is popular for use in high-speed fuel-injection applications. Details for each model are provided below.

15.10.1 Taylor Analogy Breakup (TAB) Model

Introduction

The Taylor analogy breakup (TAB) model is a classic method for calculating droplet breakup, which is applicable to many engineering sprays. This method is based upon Taylor's analogy [347] between an oscillating and distorting droplet and a spring mass system. Table 15.10.1 illustrates the analogous components.

Table 15.10.1: Comparison of a Spring-Mass System to a Distorting Droplet

Spring-Mass System	**Distorting and Oscillating Droplet**
restoring force of spring	surface tension forces
external force	droplet drag force
damping force	droplet viscosity forces

The resulting TAB model equation set, which governs the oscillating and distorting droplet, can be solved to determine the droplet oscillation and distortion at any given time. As described in detail below, when the droplet oscillations grow to a critical value the "parent" droplet will break up into a number of smaller "child" droplets. As a droplet is distorted from a spherical shape, the drag coefficient changes. A drag model that incorporates the distorting droplet effects is available in ANSYS FLUENT. See Section 15.3.5: Dynamic Drag Model Theory for details.

Use and Limitations

The TAB model is best for low-Weber-number sprays. Extremely high-Weber-number sprays result in shattering of droplets, which is not described well by the spring-mass analogy.

Droplet Distortion

The equation governing a damped, forced oscillator is [256]

$$F - kx - d\frac{dx}{dt} = m\frac{d^2x}{dt^2} \tag{15.10-1}$$

where x is the displacement of the droplet equator from its spherical (undisturbed) position. The coefficients of this equation are taken from Taylor's analogy:

$$\frac{F}{m} = C_F\frac{\rho_g u^2}{\rho_l r} \tag{15.10-2}$$

$$\frac{k}{m} = C_k \frac{\sigma}{\rho_l r^3} \tag{15.10-3}$$

$$\frac{d}{m} = C_d \frac{\mu_l}{\rho_l r^2} \tag{15.10-4}$$

where ρ_l and ρ_g are the discrete phase and continuous phase densities, u is the relative velocity of the droplet, r is the undisturbed droplet radius, σ is the droplet surface tension, and μ_l is the droplet viscosity. The dimensionless constants C_F, C_k, and C_d will be defined later.

The droplet is assumed to break up if the distortion grows to a critical ratio of the droplet radius. This breakup requirement is given as

$$x > C_b r \tag{15.10-5}$$

Where C_b is a constant equal to 0.5, if breakup is assumed to occur when the distortion is equal to half the droplet radius, i.e., oscillations at the north and south pole with this amplitude will meet at the droplet center. This implicitly assumes that the droplet is undergoing only one (fundamental) oscillation mode. Equation 15.10-1 is nondimensionalized by setting $y = x/(C_b r)$ and substituting the relationships in Equations 15.10-2–15.10-4:

$$\frac{d^2y}{dt^2} = \frac{C_F}{C_b}\frac{\rho_g}{\rho_l}\frac{u^2}{r^2} - \frac{C_k \sigma}{\rho_l r^3} y - \frac{C_d \mu_l}{\rho_l r^2}\frac{dy}{dt} \tag{15.10-6}$$

where breakup now occurs for $y > 1$. For under-damped droplets, the equation governing y can easily be determined from Equation 15.10-6 if the relative velocity is assumed to be constant:

$$y(t) = \mathrm{We}_c + e^{-(t/t_d)} \left[(y_0 - \mathrm{We}_c)\cos(\omega t) + \frac{1}{\omega}\left(\frac{dy_0}{dt} + \frac{y_0 - \mathrm{We}_c}{t_d} \right) \sin(\omega t) \right] \tag{15.10-7}$$

where

$$\mathrm{We} = \frac{\rho_g u^2 r}{\sigma} \tag{15.10-8}$$

$$\mathrm{We}_c = \frac{C_F}{C_k C_b}\mathrm{We} \tag{15.10-9}$$

$$y_0 = y(0) \tag{15.10-10}$$

$$\frac{dy_0}{dt} = \frac{dy}{dt}(0) \tag{15.10-11}$$

$$\frac{1}{t_d} = \frac{C_d}{2}\frac{\mu_l}{\rho_l r^2} \tag{15.10-12}$$

$$\omega^2 = C_k \frac{\sigma}{\rho_l r^3} - \frac{1}{t_d^2} \qquad (15.10\text{-}13)$$

In Equation 15.10-7, u is the relative velocity between the droplet and the gas phase and We is the droplet Weber number, a dimensionless parameter defined as the ratio of aerodynamic forces to surface tension forces. The droplet oscillation frequency is represented by ω. The default value of y_0 is 0, based upon the work of Liu et al. [205]. The constants have been chosen to match experiments and theory [174]:

$$\begin{aligned} C_k &= 8 \\ C_d &= 5 \\ C_F &= \frac{1}{3} \end{aligned}$$

If Equation 15.10-7 is solved for all droplets, those with $y > 1$ are assumed to break up. The size and velocity of the new child droplets must be determined.

Size of Child Droplets

The size of the child droplets is determined by equating the energy of the parent droplet to the combined energy of the child droplets. The energy of the parent droplet is [256]

$$E_{\rm parent} = 4\pi r^2 \sigma + K \frac{\pi}{5} \rho_l r^5 \left[\left(\frac{dy}{dt} \right)^2 + \omega^2 y^2 \right] \qquad (15.10\text{-}14)$$

where K is the ratio of the total energy in distortion and oscillation to the energy in the fundamental mode, of the order $(\frac{10}{3})$. The child droplets are assumed to be nondistorted and nonoscillating. Thus, the energy of the child droplets can be shown to be

$$E_{\rm child} = 4\pi r^2 \sigma \frac{r}{r_{32}} + \frac{\pi}{6} \rho_l r^5 \left(\frac{dy}{dt} \right)^2 \qquad (15.10\text{-}15)$$

where r_{32} is the Sauter mean radius of the droplet size distribution. r_{32} can be found by equating the energy of the parent and child droplets (i.e., Equations 15.10-14 and 15.10-15), setting $y = 1$, and $\omega^2 = 8\sigma/\rho_l r^3$:

$$r_{32} = \frac{r}{1 + \frac{8Ky^2}{20} + \frac{\rho_l r^3 (dy/dt)^2}{\sigma} \left(\frac{6K-5}{120} \right)} \qquad (15.10\text{-}16)$$

Once the size of the child droplets is determined, the number of child droplets can easily be determined by mass conservation.

Velocity of Child Droplets

The TAB model allows for a velocity component normal to the parent droplet velocity to be imposed upon the child droplets. When breakup occurs, the equator of the parent droplet is traveling at a velocity of $dx/dt = C_b r(dy/dt)$. Therefore, the child droplets will have a velocity normal to the parent droplet velocity given by

$$v_{\text{normal}} = C_v C_b r \frac{dy}{dt} \tag{15.10-17}$$

where C_v is a constant of order (1).

Droplet Breakup

To model droplet breakup, the TAB model first determines the amplitude for an undamped oscillation ($t_d \approx \infty$) for each droplet at time step n using the following:

$$A = \sqrt{(y^n - \mathrm{We}_c)^2 + \left(\frac{(dy/dt)^n}{\omega}\right)^2} \tag{15.10-18}$$

According to Equation 15.10-18, breakup is possible only if the following condition is satisfied:

$$\mathrm{We}_c + A > 1 \tag{15.10-19}$$

This is the limiting case, as damping will only reduce the chance of breakup. If a droplet fails the above criterion, breakup does not occur. The only additional calculations required then, are to update y using a discretized form of Equation 15.10-7 and its derivative, which are both based on work done by O'Rourke and Amsden [256]:

$$y^{n+1} = \mathrm{We}_c + e^{-(\Delta t/t_d)} \left\{ (y^n - \mathrm{We}_c)\cos(\omega t) + \frac{1}{\omega}\left[\left(\frac{dy}{dt}\right)^n + \frac{y^n - \mathrm{We}_c}{t_d}\right]\sin(\omega t) \right\} \tag{15.10-20}$$

$$\left(\frac{dy}{dt}\right)^{n+1} = \frac{\mathrm{We}_c - y^{n+1}}{t_d} + \omega e^{-(\Delta t/t_d)} \left\{ \frac{1}{\omega}\left[\left(\frac{dy}{dt}\right)^n + \frac{y^n - \mathrm{We}_c}{t_d}\right]\cos(\omega \Delta t) - (y^n - \mathrm{We}_c)\sin(\omega \Delta t) \right\} \tag{15.10-21}$$

All of the constants in these expressions are assumed to be constant throughout the time step.

If the criterion of Equation 15.10-19 is met, then breakup is possible. The breakup time, $t_{\rm bu}$, must be determined to see if breakup occurs within the time step Δt. The value of $t_{\rm bu}$ is set to the time required for oscillations to grow sufficiently large that the magnitude of the droplet distortion, y, is equal to unity. The breakup time is determined under the assumption that the droplet oscillation is undamped for its first period. The breakup time is therefore the smallest root greater than t^n of an undamped version of Equation 15.10-7:

$$\mathrm{We}_c + A\cos[\omega(t - t^n) + \phi] = 1 \tag{15.10-22}$$

where

$$\cos\phi = \frac{y^n - \mathrm{We}_c}{A} \tag{15.10-23}$$

and

$$\sin\phi = -\frac{(dy/dt)^n}{A\omega} \tag{15.10-24}$$

If $t_{\rm bu} > t^{n+1}$, then breakup will not occur during the current time step, and y and (dy/dt) are updated by Equations 15.10-20 and 15.10-21. The breakup calculation then continues with the next droplet. Conversely, if $t^n < t_{\rm bu} < t^{n+1}$, then breakup will occur and the child droplet radii are determined by Equation 15.10-16. The number of child droplets, N, is determined by mass conservation:

$$N^{n+1} = N^n \left(\frac{r^n}{r^{n+1}}\right)^3 \tag{15.10-25}$$

It is assumed that the child droplets are neither distorted nor oscillating; i.e., $y = (dy/dt) = 0$. The child droplets are represented by a number of child parcels which are created from the original parcel. These child parcels are distributed equally along the equator of the parent droplet in a plane normal to the parent relative velocity vector. The diameter of each of the child parcels is sampled from a Rosin Rammler distribution based on the Sauter mean radius (Equation 15.10-16) and a spread parameter of 3.5.

A velocity component normal to the relative velocity vector, with magnitude computed by Equation 15.10-17, is imposed upon the child droplets. It is decomposed at the equator into components pointing radially outward.

i A large number of child parcels ensures a smooth distribution of particle diameters and source terms which is needed when simulating, for example, evaporating sprays.

15.10.2 Wave Breakup Model

Introduction

An alternative to the TAB model that is appropriate for high-Weber-number flows is the wave breakup model of Reitz [287], which considers the breakup of the droplets to be induced by the relative velocity between the gas and liquid phases. The model assumes that the time of breakup and the resulting droplet size are related to the fastest-growing Kelvin-Helmholtz instability, derived from the jet stability analysis described below. The wavelength and growth rate of this instability are used to predict details of the newly-formed droplets.

Use and Limitations

The wave model is appropriate for high-speed injections, where the Kelvin-Helmholtz instability is believed to dominate droplet breakup (We $>$ 100). Because this breakup model can increase the number of computational parcels, you may wish to inject a modest number of droplets initially.

Jet Stability Analysis

The jet stability analysis described in detail by Reitz and Bracco [289] is presented briefly here. The analysis considers the stability of a cylindrical, viscous, liquid jet of radius a issuing from a circular orifice at a velocity v into a stagnant, incompressible, inviscid gas of density ρ_2. The liquid has a density, ρ_1, and viscosity, μ_1, and a cylindrical polar coordinate system is used which moves with the jet. An arbitrary infinitesimal axisymmetric surface displacement of the form

$$\eta = \eta_0 e^{ikz+\omega t} \tag{15.10-26}$$

is imposed on the initially steady motion and it is thus desired to find the dispersion relation $\omega = \omega(k)$ which relates the real part of the growth rate, ω, to its wave number, $k = 2\pi/\lambda$.

In order to determine the dispersion relation, the linearized equations for the hydrodynamics of the liquid are solved assuming wave solutions of the form

$$\phi_1 = C_1 I_0(kr) e^{ikz+\omega t} \quad (15.10\text{-}27)$$

$$\psi_1 = C_2 I_1(Lr) e^{ikz+\omega t} \quad (15.10\text{-}28)$$

where ϕ_1 and ψ_1 are the velocity potential and stream function, respectively, C_1 and C_2 are integration constants, I_0 and I_1 are modified Bessel functions of the first kind, $L^2 = k^2 + \omega/\nu_1$, and ν_1 is the liquid kinematic viscosity [287]. The liquid pressure is obtained from the inviscid part of the liquid equations. In addition, the inviscid gas equations can be solved to obtain the fluctuating gas pressure at $r = a$:

$$-p_{21} = -\rho_2 (U - i\omega k)^2 k\eta \frac{K_0(ka)}{K_1(ka)} \quad (15.10\text{-}29)$$

where K_0 and K_1 are modified Bessel functions of the second kind and u is the relative velocity between the liquid and the gas. The linearized boundary conditions are

$$v_1 = \frac{\partial \eta}{\partial t} \quad (15.10\text{-}30)$$

$$\frac{\partial u_1}{\partial r} = -\frac{\partial v_1}{\partial z} \quad (15.10\text{-}31)$$

and

$$-p_1 + 2\mu_1 - \frac{\sigma}{a^2}\left(\eta + a^2 \frac{\partial^2 \eta}{\partial z^2}\right) + p_2 = 0 \quad (15.10\text{-}32)$$

which are mathematical statements of the liquid kinematic free surface condition, continuity of shear stress, and continuity of normal stress, respectively. Note that u_1 is the axial perturbation liquid velocity, v_1 is the radial perturbation liquid velocity, and σ is the surface tension. Also note that Equation 15.10-31 was obtained under the assumption that $v_2 = 0$.

As described by Reitz [287], Equations 15.10-30 and 15.10-31 can be used to eliminate the integration constants C_1 and C_2 in Equations 15.10-27 and 15.10-28. Thus, when the pressure and velocity solutions are substituted into Equation 15.10-32, the desired dispersion relation is obtained:

$$\omega^2 + 2\nu_1 k^2 \omega \left[\frac{I_1'(ka)}{I_0(ka)} - \frac{2kL}{k^2+L^2} \frac{I_1(ka)}{I_0(ka)} \frac{I_1'(La)}{I_1(La)} \right] =$$

$$\frac{\sigma k}{\rho_1 a^2}(1-k^2a^2)\left(\frac{L^2-a^2}{L^2+a^2}\right)\frac{I_1(ka)}{I_0(ka)} + \frac{\rho_2}{\rho_1}\left(U - i\frac{\omega}{k}\right)^2 \left(\frac{L^2-a^2}{L^2+a^2}\right)\frac{I_1(ka)}{I_0(ka)}\frac{K_0(ka)}{K_1(ka)} \quad (15.10\text{-}33)$$

As shown by Reitz [287], Equation 15.10-33 predicts that a maximum growth rate (or most unstable wave) exists for a given set of flow conditions. Curve fits of numerical solutions to Equation 15.10-33 were generated for the maximum growth rate, Ω, and the corresponding wavelength, Λ, and are given by Reitz [287]:

$$\frac{\Lambda}{a} = 9.02 \frac{(1+0.45\mathrm{Oh}^{0.5})(1+0.4\mathrm{Ta}^{0.7})}{(1+0.87\mathrm{We}_2^{1.67})^{0.6}} \quad (15.10\text{-}34)$$

$$\Omega\left(\frac{\rho_1 a^3}{\sigma}\right) = \frac{(0.34+0.38\mathrm{We}_2^{1.5})}{(1+\mathrm{Oh})(1+1.4\mathrm{Ta}^{0.6})} \quad (15.10\text{-}35)$$

where $\mathrm{Oh} = \sqrt{\mathrm{We}_1}/\mathrm{Re}_1$ is the Ohnesorge number and $\mathrm{Ta} = \mathrm{Oh}\sqrt{\mathrm{We}_2}$ is the Taylor number. Furthermore, $\mathrm{We}_1 = \rho_1 U^2 a/\sigma$ and $\mathrm{We}_2 = \rho_2 U^2 a/\sigma$ are the liquid and gas Weber numbers, respectively, and $\mathrm{Re}_1 = Ua/\nu_1$ is the Reynolds number.

Droplet Breakup

In the wave model, breakup of droplet parcels is calculated by assuming that the radius of the newly-formed droplets is proportional to the wavelength of the fastest-growing unstable surface wave on the parent droplet. In other words,

$$r = B_0 \Lambda \quad (15.10\text{-}36)$$

where B_0 is a model constant set equal to 0.61 based on the work of Reitz [287]. Furthermore, the rate of change of droplet radius in the parent parcel is given by

$$\frac{da}{dt} = -\frac{(a-r)}{\tau}, \quad r \le a \quad (15.10\text{-}37)$$

where the breakup time, τ, is given by

$$\tau = \frac{3.726 B_1 a}{\Lambda\Omega} \tag{15.10-38}$$

and Λ and Ω are obtained from Equations 15.10-34 and 15.10-35, respectively. The breakup time constant, B_1, is set to a value of 1.73 as recommended by Liu et al. [205]. Values of B_1 can range between 1 and 60, depending on the injector characterization.

In the wave model, mass is accumulated from the parent drop at a rate given by Equation 15.10-38 until the shed mass is equal to 5% of the initial parcel mass. At this time, a new parcel is created with a radius given by Equation 15.10-36. The new parcel is given the same properties as the parent parcel (i.e., temperature, material, position, etc.) with the exception of radius and velocity. The new parcel is given a component of velocity randomly selected in the plane orthogonal to the direction vector of the parent parcel, and the momentum of the parent parcel is adjusted so that momentum is conserved. The velocity magnitude of the new parcel is the same as the parent parcel.

You must also specify the model constants which determine how the gas phase interacts with the liquid droplets. For example, the breakup time constant B1 is the constant multiplying the time scale which determines how quickly the parcel will loose mass. Therefore, a larger number means that it takes longer for the particle to loose a given amount. A larger number for B1 in the context of interaction with the gas phase would mean that the interaction with the subgrid is less intense. B0 is the constant for the drop size and is generally taken to be 0.61.

15.11 Droplet Collision and Coalescence Model Theory

15.11.1 Introduction

When your simulation includes tracking of droplets, ANSYS FLUENT provides an option for estimating the number of droplet collisions and their outcomes in a computationally efficient manner. The difficulty in any collision calculation is that for N droplets, each droplet has $N-1$ possible collision partners. Thus, the number of possible collision pairs is approximately $\frac{1}{2}N^2$. (The factor of $\frac{1}{2}$ appears because droplet A colliding with droplet B is identical to droplet B colliding with droplet A. This symmetry reduces the number of possible collision events by half.)

An important consideration is that the collision algorithm must calculate $\frac{1}{2}N^2$ possible collision events at every time step. Since a spray can consist of several million droplets, the computational cost of a collision calculation from first principles is prohibitive. This motivates the concept of parcels. Parcels are statistical representations of a number of individual droplets. For example, if ANSYS FLUENT tracks a set of parcels, each of which represents 1000 droplets, the cost of the collision calculation is reduced by a factor of 10^6. Because the cost of the collision calculation still scales with the square of N, the reduction of cost is significant; however, the effort to calculate the possible intersection of so many parcel trajectories would still be prohibitively expensive.

The algorithm of O'Rourke [255] efficiently reduces the computational cost of the spray calculation. Rather than using geometry to see if parcel paths intersect, O'Rourke's method is a stochastic estimate of collisions. O'Rourke also makes the assumption that two parcels may collide only if they are located in the same continuous-phase cell. These two assumptions are valid only when the continuous-phase cell size is small compared to the size of the spray. For these conditions, the method of O'Rourke is second-order accurate at estimating the chance of collisions. The concept of parcels together with the algorithm of O'Rourke makes the calculation of collision possible for practical spray problems.

Once it is decided that two parcels of droplets collide, the algorithm further determines the type of collision. Only coalescence and bouncing outcomes are considered. The probability of each outcome is calculated from the collisional Weber number (We_c) and a fit to experimental observations. Here,

$$We_c = \frac{\rho U_{rel}^2 \overline{D}}{\sigma} \tag{15.11-1}$$

where U_{rel} is the relative velocity between two parcels and $\overline{D}$ is the arithmetic mean diameter of the two parcels. The state of the two colliding parcels is modified based on the outcome of the collision.

15.11.2 Use and Limitations

The collision model assumes that the frequency of collisions is much less than the particle time step. If the particle time step is too large, then the results may be time-step-dependent. You should adjust the particle length scale accordingly. Additionally, the model is most applicable for low-Weber-number collisions where collisions result in bouncing and coalescence. Above a Weber number of about 100, the outcome of collision could be shattering.

Sometimes the collision model can cause mesh-dependent artifacts to appear in the spray. This is a result of the assumption that droplets can collide only within the same cell. These tend to be visible when the source of injection is at a mesh vertex. The coalescence of droplets tends to cause the spray to pull away from cell boundaries. In two dimensions, a finer mesh and more computational droplets can be used to reduce these effects. In three dimensions, best results are achieved when the spray is modeled using a polar mesh with the spray at the center.

If the collision model is used in a transient simulation, multiple DPM iterations per time step cannot be specified in the Number of Continuous Phase Iterations per DPM Iteration field in the **Discrete Phase Model** dialog box. In such cases, only one DPM iteration per time step will be calculated.

15.11.3 Theory

As noted above, O'Rourke's algorithm assumes that two droplets may collide only if they are in the same continuous-phase cell. This assumption can prevent droplets that are quite close to each other, but not in the same cell, from colliding, although the effect of this error is lessened by allowing some droplets that are farther apart to collide. The overall accuracy of the scheme is second-order in space.

Probability of Collision

The probability of collision of two droplets is derived from the point of view of the larger droplet, called the collector droplet and identified below with the number 1. The smaller droplet is identified in the following derivation with the number 2. The calculation is in the frame of reference of the larger droplet so that the velocity of the collector droplet is zero. Only the relative distance between the collector and the smaller droplet is important in this derivation. If the smaller droplet is on a collision course with the collector, the centers will pass within a distance of $r_1 + r_2$. More precisely, if the smaller droplet center passes within a flat circle centered around the collector of area $\pi(r_1 + r_2)^2$ perpendicular to the trajectory of the smaller droplet, a collision will take place. This disk can be used to define the collision volume, which is the area of the aforementioned disk multiplied by the distance traveled by the smaller droplet in one time step, namely $\pi(r_1 + r_2)^2 v_{\rm rel}\Delta t$.

The algorithm of O'Rourke uses the concept of a collision volume to calculate the probability of collision. Rather than calculating whether or not the position of the smaller droplet center is within the collision volume, the algorithm calculates the probability of the smaller droplet being within the collision volume. It is known that the smaller droplet is somewhere within the continuous-phase cell of volume V. If there is a uniform probability of the droplet being anywhere within the cell, then the chance of the droplet being within the collision volume is the ratio of the two volumes. Thus, the probability of the collector colliding with the smaller droplet is

$$P_1 = \frac{\pi(r_1 + r_2)^2 v_{\rm rel}\Delta t}{V} \tag{15.11-2}$$

Equation 15.11-2 can be generalized for parcels, where there are n_1 and n_2 droplets in the collector and smaller droplet parcels, respectively. The collector undergoes a mean expected number of collisions given by

$$\bar{n} = \frac{n_2\pi(r_1 + r_2)^2 v_{\rm rel}\Delta t}{V} \tag{15.11-3}$$

The actual number of collisions that the collector experiences is not generally the mean expected number of collisions. The probability distribution of the number of collisions follows a Poisson distribution, according to O'Rourke, which is given by

$$P(n) = e^{-\bar{n}} \frac{\bar{n}^n}{n!} \tag{15.11-4}$$

where n is the number of collisions between a collector and other droplets.

Collision Outcomes

Once it is determined that two parcels collide, the outcome of the collision must be determined. In general, the outcome tends to be coalescence if the droplets collide head-on, and bouncing if the collision is more oblique. In the reference frame being used here, the probability of coalescence can be related to the offset of the collector droplet center and the trajectory of the smaller droplet. The critical offset is a function of the collisional Weber number and the relative radii of the collector and the smaller droplet.

The critical offset is calculated by O'Rourke using the expression

$$b_{\rm crit} = (r_1 + r_2)\sqrt{\min\left(1.0, \frac{2.4f}{\rm We}\right)} \tag{15.11-5}$$

where f is a function of r_1/r_2, defined as

$$f\left(\frac{r_1}{r_2}\right) = \left(\frac{r_1}{r_2}\right)^3 - 2.4\left(\frac{r_1}{r_2}\right)^2 + 2.7\left(\frac{r_1}{r_2}\right) \tag{15.11-6}$$

The value of the actual collision parameter, b, is $(r_1 + r_2)\sqrt{Y}$, where Y is a random number between 0 and 1. The calculated value of b is compared to $b_{\rm crit}$, and if $b < b_{\rm crit}$, the result of the collision is coalescence. Equation 15.11-4 gives the number of smaller droplets that coalesce with the collector. The properties of the coalesced droplets are found from the basic conservation laws.

In the case of a grazing collision, the new velocities are calculated based on conservation of momentum and kinetic energy. It is assumed that some fraction of the kinetic energy of the droplets is lost to viscous dissipation and angular momentum generation. This fraction is related to b, the collision offset parameter. Using assumed forms for the energy loss, O'Rourke derived the following expression for the new velocity:

$$v_1' = \frac{m_1 v_1 + m_2 v_2 + m_2(v_1 - v_2)}{m_1 + m_2}\left(\frac{b - b_{\rm crit}}{r_1 + r_2 - b_{\rm crit}}\right) \tag{15.11-7}$$

This relation is used for each of the components of velocity. No other droplet properties are altered in grazing collisions.

15.12 One-Way and Two-Way Coupling

You can use ANSYS FLUENT to predict the discrete phase patterns based on a fixed continuous phase flow field (an *uncoupled* approach or "one-way coupling"), or you can include the effect of the discrete phase on the continuum (a *coupled* approach or "two-way coupling"). In the coupled approach, the continuous phase flow pattern is impacted by the discrete phase (and vice versa), and you can alternate calculations of the continuous phase and discrete phase equations until a converged coupled solution is achieved.

15.12.1 Coupling Between the Discrete and Continuous Phases

As the trajectory of a particle is computed, ANSYS FLUENT keeps track of the heat, mass, and momentum gained or lost by the particle stream that follows that trajectory and these quantities can be incorporated in the subsequent continuous phase calculations. Thus, while the continuous phase always impacts the discrete phase, you can also incorporate the effect of the discrete phase trajectories on the continuum. This two-way coupling is accomplished by alternately solving the discrete and continuous phase equations until the solutions in both phases have stopped changing. This interphase exchange of heat, mass, and momentum from the particle to the continuous phase is depicted qualitatively in Figure 15.12.1. Note that no interchange terms are computed for particles defined as massless, where the discrete phase trajectories have no impact on the continuum.

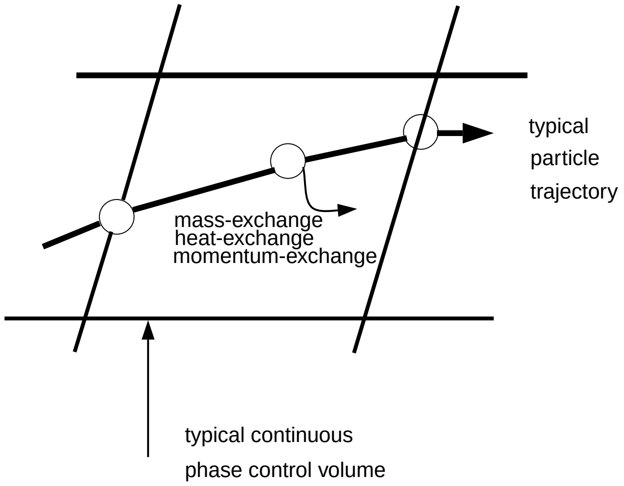


Figure 15.12.1: Heat, Mass, and Momentum Transfer Between the Discrete and Continuous Phases

15.12.2 Momentum Exchange

The momentum transfer from the continuous phase to the discrete phase is computed in ANSYS FLUENT by examining the change in momentum of a particle as it passes through each control volume in the ANSYS FLUENT model. This momentum change is computed as

$$F = \sum \left(\frac{18 \mu C_D \mathrm{Re}}{\rho_p d_p^2 24} (u_p - u) + F_{\mathrm{other}} \right) \dot{m}_p \Delta t \tag{15.12-1}$$

where

μ	=	viscosity of the fluid
ρ_p	=	density of the particle
d_p	=	diameter of the particle
Re	=	relative Reynolds number
u_p	=	velocity of the particle
u	=	velocity of the fluid
C_D	=	drag coefficient
$\dot{m}_p$	=	mass flow rate of the particles
Δt	=	time step
F_{other}	=	other interaction forces

This momentum exchange appears as a momentum source in the continuous phase momentum balance in any subsequent calculations of the continuous phase flow field and can be reported by ANSYS FLUENT as described in Section 23.7: Postprocessing for the Discrete Phase in the separate User's Guide.

15.12.3 Heat Exchange

The heat transfer from the continuous phase to the discrete phase is computed in ANSYS FLUENT by examining the change in thermal energy of a particle as it passes through each control volume in the ANSYS FLUENT model. In the absence of a chemical reaction (i.e., for all particle laws except Law 5) the heat exchange is computed as

$$Q = \frac{\dot{m}_{p,0}}{m_{p,0}} \left[(m_{p_{\mathrm{in}}} - m_{p_{\mathrm{out}}})[-H_{lat\mathrm{ref}} + H_{\mathrm{pyrol}}] - m_{p_{\mathrm{out}}} \int_{T_{\mathrm{ref}}}^{T_{p_{\mathrm{out}}}} c_{p_p} dT + m_{p_{\mathrm{in}}} \int_{T_{\mathrm{ref}}}^{T_{p_{\mathrm{in}}}} c_{p_p} dT \right] \tag{15.12-2}$$

where

$\dot{m}_{p,0}$ = initial mass flow rate of the particle injection (kg/s)
$m_{p,0}$ = initial mass of the particle (kg)
$m_{p_{\rm in}}$ = mass of the particle on cell entry (kg)
$m_{p_{\rm out}}$ = mass of the particle on cell exit (kg)
c_{p_p} = heat capacity of the particle (J/kg-K)
$H_{\rm pyrol}$ = heat of pyrolysis as volatiles are evolved (J/kg)
$T_{p_{\rm in}}$ = temperature of the particle on cell entry (K)
$T_{p_{\rm out}}$ = temperature of the particle on cell exit (K)
$T_{\rm ref}$ = reference temperature for enthalpy (K)
$H_{lat{\rm ref}}$ = latent heat at reference conditions (J/kg)

The latent heat at the reference conditions $H_{lat{\rm ref}}$ for droplet particles is computed as the difference of the liquid and gas standard formation enthalpies, and can be related to the latent heat at the boiling point as follows:

$$H_{lat{\rm ref}} = H_{lat} - \int_{T_{\rm ref}}^{T_{bp}} c_{p_g} dT + \int_{T_{\rm ref}}^{T_{bp}} c_{p_p} dT \qquad (15.12\text{-}3)$$

where

c_{p_g} = heat capacity of gas product species (J/kg-K)
T_{bp} = boiling point temperature (K)
H_{lat} = latent heat at the boiling point temperature (J/kg)

For the volatile part of the combusting particles, some constraints are applied to ensure that the enthalpy source terms do not depend on the particle history. The formulation should be consistent with the mixing of two gas streams, one consisting of the fluid and the other consisting of the volatiles. Hence $H_{lat{\rm ref}}$ is derived by applying a correction to H_{lat}, which accounts for different heat capacities in the particle and gaseous phase:

$$H_{lat{\rm ref}} = H_{lat} - \int_{T_{\rm ref}}^{T_{p,init}} c_{p_g} dT + \int_{T_{\rm ref}}^{T_{p,init}} c_{p_p} dT \qquad (15.12\text{-}4)$$

where

$T_{p,init}$ = particle initial temperature (K)

15.12.4 Mass Exchange

The mass transfer from the discrete phase to the continuous phase is computed in ANSYS FLUENT by examining the change in mass of a particle as it passes through each control volume in the ANSYS FLUENT model. The mass change is computed simply as

$$M = \frac{\Delta m_p}{m_{p,0}} \; \dot{m}_{p,0} \qquad (15.12\text{-}5)$$

This mass exchange appears as a source of mass in the continuous phase continuity equation and as a source of a chemical species defined by you. The mass sources are included in any subsequent calculations of the continuous phase flow field and are reported by ANSYS FLUENT as described in Section 23.7: Postprocessing for the Discrete Phase in the separate User's Guide.

15.12.5 Under-Relaxation of the Interphase Exchange Terms

Note that the interphase exchange of momentum, heat, and mass is under-relaxed during the calculation, so that

$$F_{\rm new} = F_{\rm old} + \alpha(F_{\rm calculated} - F_{\rm old}) \tag{15.12-6}$$

$$Q_{\rm new} = Q_{\rm old} + \alpha(Q_{\rm calculated} - Q_{\rm old}) \tag{15.12-7}$$

$$M_{\rm new} = M_{\rm old} + \alpha(M_{\rm calculated} - M_{\rm old}) \tag{15.12-8}$$

where α is the under-relaxation factor for particles/droplets. The default value for α is 0.5. This value may be reduced to improve the stability of coupled calculations. Note that the value of α does not influence the predictions obtained in the final converged solution.

Two options exist when updating the new particle source terms $F_{\rm new}$, $Q_{\rm new}$ and $M_{\rm new}$. The first option is to compute the new source terms and the particle source terms, $F_{\rm calculated}$, $Q_{\rm calculated}$ and $M_{\rm calculated}$, at the same time. The second option is to update the new source terms, $F_{\rm new}$, $Q_{\rm new}$ and $M_{\rm new}$, every flow iteration, while the particle source terms, $F_{\rm calculated}$, $Q_{\rm calculated}$ and $M_{\rm calculated}$, are calculated every Discrete Phase Model iteration. The latter option is recommended for transient flows, where the particles are updated once per flow time step.

Figure 15.12.2 shows how the source term, S, when applied to the flow equations, changes with the number of updates for varying under-relaxation factors. In Figure 15.12.2, S_∞ is the final source term for which a value is reached after a certain number of updates and S_0 is the initial source term at the start of the computation. The value of S_0 is typically zero at the beginning of the calculation.

Figure 15.12.2 can be applied to this option as well. Keep in mind that the DPM source terms are updated every continuous flow iteration.

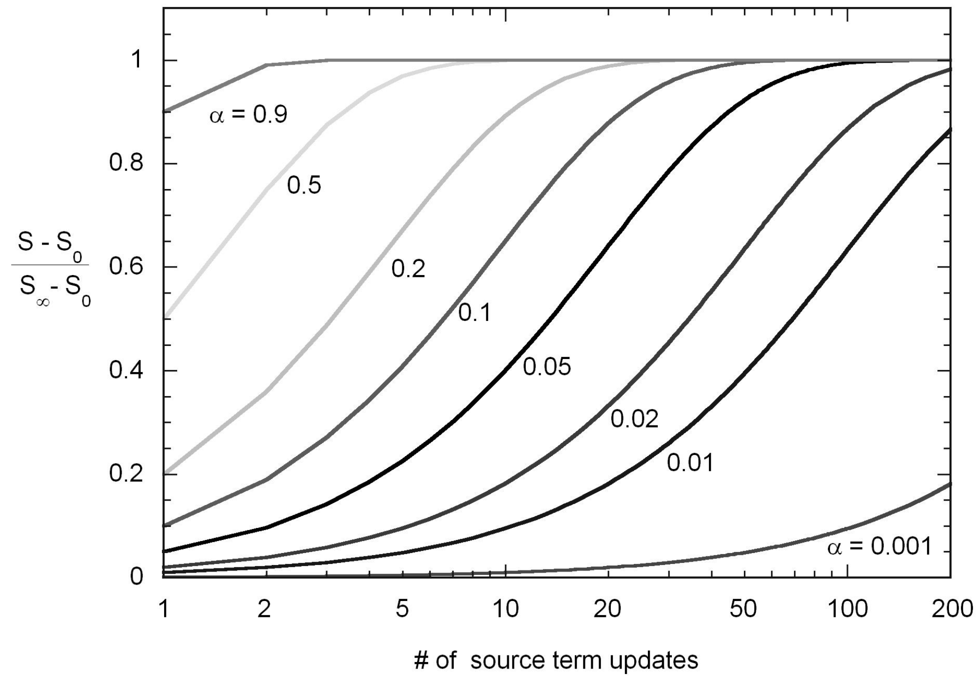


Figure 15.12.2: Effect of Number of Source Term Updates on Source Term Applied to Flow Equations

15.12.6 Interphase Exchange During Stochastic Tracking

When stochastic tracking is performed, the interphase exchange terms, computed via Equations 15.12-1 to 15.12-8, are computed for each stochastic trajectory with the particle mass flow rate, $\dot{m}_{p0}$, divided by the number of stochastic tracks computed. This implies that an equal mass flow of particles follows each stochastic trajectory.

15.12.7 Interphase Exchange During Cloud Tracking

When the particle cloud model is used, the interphase exchange terms are computed via Equations 15.12-1 to 15.12-8 based on ensemble-averaged flow properties in the particle cloud. The exchange terms are then distributed to all the cells in the cloud based on the weighting factor defined in Equation 15.2-49.

Chapter 16. Multiphase Flows

This chapter discusses the general multiphase models that are available in ANSYS FLUENT. Section 16.1: Introduction provides a brief introduction to multiphase modeling, Chapter 15: Discrete Phase discusses the Lagrangian dispersed phase model, and Chapter 17: Solidification and Melting describes ANSYS FLUENT's model for solidification and melting. For information about using the general multiphase models in ANSYS FLUENT, see Chapter 24: Modeling Multiphase Flows in the separate User's Guide. Information about the various theories behind the multiphase models is presented in the following sections:

- Section 16.1: Introduction
- Section 16.2: Choosing a General Multiphase Model
- Section 16.3: Volume of Fluid (VOF) Model Theory
- Section 16.4: Mixture Model Theory
- Section 16.5: Eulerian Model Theory
- Section 16.6: Wet Steam Model Theory
- Section 16.7: Modeling Mass Transfer in Multiphase Flows
- Section 16.8: Modeling Species Transport in Multiphase Flows

16.1 Introduction

A large number of flows encountered in nature and technology are a mixture of phases. Physical phases of matter are gas, liquid, and solid, but the concept of phase in a multiphase flow system is applied in a broader sense. In multiphase flow, a phase can be defined as an identifiable class of material that has a particular inertial response to and interaction with the flow and the potential field in which it is immersed. For example, different-sized solid particles of the same material can be treated as different phases because each collection of particles with the same size will have a similar dynamical response to the flow field.

Information is organized into the following subsections:

- Section 16.1.1: Multiphase Flow Regimes
- Section 16.1.2: Examples of Multiphase Systems

16.1.1 Multiphase Flow Regimes

Multiphase flow regimes can be grouped into four categories: gas-liquid or liquid-liquid flows; gas-solid flows; liquid-solid flows; and three-phase flows.

Gas-Liquid or Liquid-Liquid Flows

The following regimes are gas-liquid or liquid-liquid flows:

- Bubbly flow: This is the flow of discrete gaseous or fluid bubbles in a continuous fluid.
- Droplet flow: This is the flow of discrete fluid droplets in a continuous gas.
- Slug flow: This is the flow of large bubbles in a continuous fluid.
- Stratified/free-surface flow: This is the flow of immiscible fluids separated by a clearly-defined interface.

See Figure 16.1.1 for illustrations of these regimes.

Gas-Solid Flows

The following regimes are gas-solid flows:

- Particle-laden flow: This is flow of discrete particles in a continuous gas.
- Pneumatic transport: This is a flow pattern that depends on factors such as solid loading, Reynolds numbers, and particle properties. Typical patterns are dune flow, slug flow, and homogeneous flow.
- Fluidized bed: This consists of a vertical cylinder containing particles, into which a gas is introduced through a distributor. The gas rising through the bed suspends the particles. Depending on the gas flow rate, bubbles appear and rise through the bed, intensifying the mixing within the bed.

See Figure 16.1.1 for illustrations of these regimes.

Liquid-Solid Flows

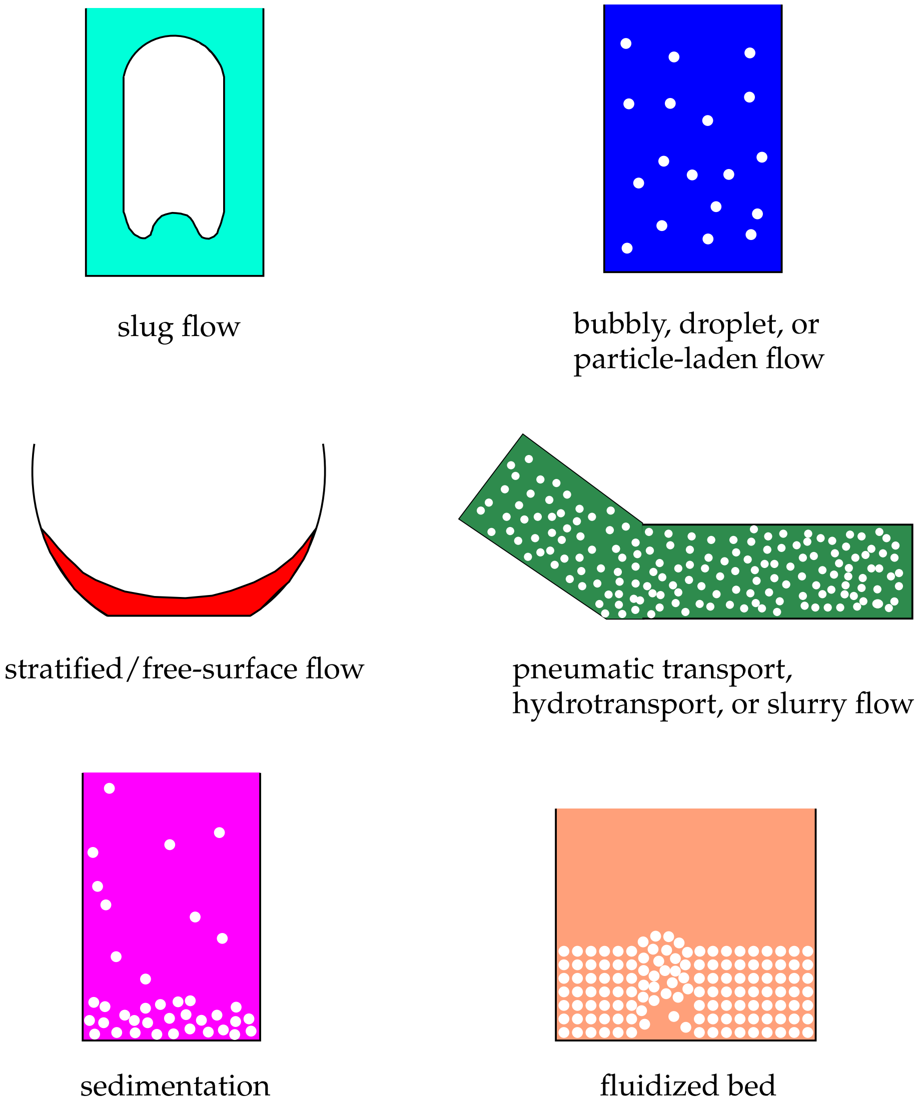
The following regimes are liquid-solid flows:

- Slurry flow: This flow is the transport of particles in liquids. The fundamental behavior of liquid-solid flows varies with the properties of the solid particles relative to those of the liquid. In slurry flows, the Stokes number (see Equation 16.2-4) is normally less than 1. When the Stokes number is larger than 1, the characteristic of the flow is liquid-solid fluidization.
- Hydrotransport: This describes densely-distributed solid particles in a continuous liquid
- Sedimentation: This describes a tall column initially containing a uniform dispersed mixture of particles. At the bottom, the particles will slow down and form a sludge layer. At the top, a clear interface will appear, and in the middle a constant settling zone will exist.

See Figure 16.1.1 for illustrations of these regimes.

Three-Phase Flows

Three-phase flows are combinations of the other flow regimes listed in the previous sections.



slug flow

bubbly, droplet, or particle-laden flow

stratified/free-surface flow

pneumatic transport, hydrotransport, or slurry flow

sedimentation

fluidized bed

Figure 16.1.1: Multiphase Flow Regimes

16.1.2 Examples of Multiphase Systems

Specific examples of each regime described in Section 16.1.1: Multiphase Flow Regimes are listed below:

- Bubbly flow examples include absorbers, aeration, air lift pumps, cavitation, evaporators, flotation, and scrubbers.
- Droplet flow examples include absorbers, atomizers, combustors, cryogenic pumping, dryers, evaporation, gas cooling, and scrubbers.
- Slug flow examples include large bubble motion in pipes or tanks.
- Stratified/free-surface flow examples include sloshing in offshore separator devices and boiling and condensation in nuclear reactors.
- Particle-laden flow examples include cyclone separators, air classifiers, dust collectors, and dust-laden environmental flows.
- Pneumatic transport examples include transport of cement, grains, and metal powders.
- Fluidized bed examples include fluidized bed reactors and circulating fluidized beds.
- Slurry flow examples include slurry transport and mineral processing
- Hydrotransport examples include mineral processing and biomedical and physiochemical fluid systems
- Sedimentation examples include mineral processing.

16.2 Choosing a General Multiphase Model

The first step in solving any multiphase problem is to determine which of the regimes provides some broad guidelines for determining appropriate models for each regime, and how to determine the degree of interphase coupling for flows involving bubbles, droplets, or particles, and the appropriate model for different amounts of coupling.

Information is organized into the following subsections:

- Section 16.2.1: Approaches to Multiphase Modeling
- Section 16.2.2: Model Comparisons
- Section 16.2.3: Time Schemes in Multiphase Flow
- Section 16.2.4: Stability and Convergence

16.2.1 Approaches to Multiphase Modeling

Advances in computational fluid mechanics have provided the basis for further insight into the dynamics of multiphase flows. Currently there are two approaches for the numerical calculation of multiphase flows: the Euler-Lagrange approach (discussed in Section 15.1: Introduction) and the Euler-Euler approach (discussed in the following section).

The Euler-Euler Approach

In the Euler-Euler approach, the different phases are treated mathematically as interpenetrating continua. Since the volume of a phase cannot be occupied by the other phases, the concept of phasic volume fraction is introduced. These volume fractions are assumed to be continuous functions of space and time and their sum is equal to one. Conservation equations for each phase are derived to obtain a set of equations, which have similar structure for all phases. These equations are closed by providing constitutive relations that are obtained from empirical information, or, in the case of granular flows, by application of kinetic theory.

In ANSYS FLUENT, three different Euler-Euler multiphase models are available: the volume of fluid (VOF) model, the mixture model, and the Eulerian model.

The VOF Model

The VOF model (described in Section 16.3: Volume of Fluid (VOF) Model Theory) is a surface-tracking technique applied to a fixed Eulerian mesh. It is designed for two or more immiscible fluids where the position of the interface between the fluids is of interest. In the VOF model, a single set of momentum equations is shared by the fluids, and the volume fraction of each of the fluids in each computational cell is tracked throughout the domain. Applications of the VOF model include stratified flows, free-surface flows, filling, sloshing, the motion of large bubbles in a liquid, the motion of liquid after a dam break, the prediction of jet breakup (surface tension), and the steady or transient tracking of any liquid-gas interface.

The Mixture Model

The mixture model (described in Section 16.4: Mixture Model Theory) is designed for two or more phases (fluid or particulate). As in the Eulerian model, the phases are treated as interpenetrating continua. The mixture model solves for the mixture momentum equation and prescribes relative velocities to describe the dispersed phases. Applications of the mixture model include particle-laden flows with low loading, bubbly flows, sedimentation, and cyclone separators. The mixture model can also be used without relative velocities for the dispersed phases to model homogeneous multiphase flow.

The Eulerian Model

The Eulerian model (described in Section 16.5: Eulerian Model Theory) is the most complex of the multiphase models in ANSYS FLUENT. It solves a set of n momentum and continuity equations for each phase. Coupling is achieved through the pressure and interphase exchange coefficients. The manner in which this coupling is handled depends upon the type of phases involved; granular (fluid-solid) flows are handled differently than nongranular (fluid-fluid) flows. For granular flows, the properties are obtained from application of kinetic theory. Momentum exchange between the phases is also dependent upon the type of mixture being modeled. ANSYS FLUENT's user-defined functions allow you to customize the calculation of the momentum exchange. Applications of the Eulerian multiphase model include bubble columns, risers, particle suspension, and fluidized beds.

16.2.2 Model Comparisons

In general, once you have determined the flow regime that best represents your multiphase system, you can select the appropriate model based on the following guidelines:

- For bubbly, droplet, and particle-laden flows in which the phases mix and/or dispersed-phase volume fractions exceed 10%, use either the mixture model (described in Section 16.4: Mixture Model Theory) or the Eulerian model (described in Section 16.5: Eulerian Model Theory).
- For slug flows, use the VOF model. See Section 16.3: Volume of Fluid (VOF) Model Theory for more information about the VOF model.
- For stratified/free-surface flows, use the VOF model. See Section 16.3: Volume of Fluid (VOF) Model Theory for more information about the VOF model.
- For pneumatic transport, use the mixture model for homogeneous flow (described in Section 16.4: Mixture Model Theory) or the Eulerian model for granular flow (described in Section 16.5: Eulerian Model Theory).
- For fluidized beds, use the Eulerian model for granular flow. See Section 16.5: Eulerian Model Theory for more information about the Eulerian model.
- For slurry flows and hydrotransport, use the mixture or Eulerian model (described, respectively, in Sections 16.4 and 16.5).
- For sedimentation, use the Eulerian model. See Section 16.5: Eulerian Model Theory for more information about the Eulerian model.
- For general, complex multiphase flows that involve multiple flow regimes, select the aspect of the flow that is of most interest, and choose the model that is most appropriate for that aspect of the flow. Note that the accuracy of results will not be as good as for flows that involve just one flow regime, since the model you use will be valid for only part of the flow you are modeling.

As discussed in this section, the VOF model is appropriate for stratified or free-surface flows, and the mixture and Eulerian models are appropriate for flows in which the phases mix or separate and/or dispersed-phase volume fractions exceed 10%. (Flows in which the dispersed-phase volume fractions are less than or equal to 10% can be modeled using the discrete phase model described in Chapter 15: Discrete Phase.)

To choose between the mixture model and the Eulerian model, you should consider the following guidelines:

- If there is a wide distribution of the dispersed phases (i.e., if the particles vary in size and the largest particles do not separate from the primary flow field), the mixture model may be preferable (i.e., less computationally expensive). If the dispersed phases are concentrated just in portions of the domain, you should use the Eulerian model instead.
- If interphase drag laws that are applicable to your system are available (either within ANSYS FLUENT or through a user-defined function), the Eulerian model can usually provide more accurate results than the mixture model. Even though you can apply the same drag laws to the mixture model, as you can for a nongranular Eulerian simulation, if the interphase drag laws are unknown or their applicability to your system is questionable, the mixture model may be a better choice. For most cases with spherical particles, then the Schiller-Naumann law is more than adequate. For cases with nonspherical particles, then a user-defined function can be used.
- If you want to solve a simpler problem, which requires less computational effort, the mixture model may be a better option, since it solves a smaller number of equations than the Eulerian model. If accuracy is more important than computational effort, the Eulerian model is a better choice. Keep in mind, however, that the complexity of the Eulerian model can make it less computationally stable than the mixture model.

ANSYS FLUENT's multiphase models are compatible with ANSYS FLUENT's dynamic mesh modeling feature. For more information on the dynamic mesh feature, see Section 3: Flows Using Sliding and Deforming Meshes. For more information about how other ANSYS FLUENT models are compatible with ANSYS FLUENT's multiphase models, see Appendix A: ANSYS FLUENT Model Compatibility in the separate User's Guide.

Detailed Guidelines

For stratified and slug flows, the choice of the VOF model, as indicated in Section 16.2.2: Model Comparisons, is straightforward. Choosing a model for the other types of flows is less straightforward. As a general guide, there are some parameters that help to identify the appropriate multiphase model for these other flows: the particulate loading, β, and the Stokes number, St. (Note that the word “particle” is used in this discussion to refer to a particle, droplet, or bubble.)

The Effect of Particulate Loading

Particulate loading has a major impact on phase interactions. The particulate loading is defined as the mass density ratio of the dispersed phase (d) to that of the carrier phase (c):

$$\beta = \frac{\alpha_d \rho_d}{\alpha_c \rho_c} \tag{16.2-1}$$

The material density ratio

$$\gamma = \frac{\rho_d}{\rho_c} \tag{16.2-2}$$

is greater than 1000 for gas-solid flows, about 1 for liquid-solid flows, and less than 0.001 for gas-liquid flows.

Using these parameters it is possible to estimate the average distance between the individual particles of the particulate phase. An estimate of this distance has been given by Crowe et al. [62]:

$$\frac{L}{d_d} = \left(\frac{\pi}{6} \frac{1+\kappa}{\kappa} \right)^{1/3} \tag{16.2-3}$$

where $\kappa = \frac{\beta}{\gamma}$. Information about these parameters is important for determining how the dispersed phase should be treated. For example, for a gas-particle flow with a particulate loading of 1, the interparticle space $\frac{L}{d_d}$ is about 8; the particle can therefore be treated as isolated (i.e., very low particulate loading).

Depending on the particulate loading, the degree of interaction between the phases can be divided into the following three categories:

- For very low loading, the coupling between the phases is one-way (i.e., the fluid carrier influences the particles via drag and turbulence, but the particles have no influence on the fluid carrier). The discrete phase (Chapter 15: Discrete Phase), mixture, and Eulerian models can all handle this type of problem correctly. Since the Eulerian model is the most expensive, the discrete phase or mixture model is recommended.
- For intermediate loading, the coupling is two-way (i.e., the fluid carrier influences the particulate phase via drag and turbulence, but the particles in turn influence the carrier fluid via reduction in mean momentum and turbulence). The discrete phase(Chapter 15: Discrete Phase) , mixture, and Eulerian models are all applicable in this case, but you need to take into account other factors in order to decide which model is more appropriate. See below for information about using the Stokes number as a guide.
- For high loading, there is two-way coupling plus particle pressure and viscous stresses due to particles (four-way coupling). Only the Eulerian model will handle this type of problem correctly.

The Significance of the Stokes Number

For systems with intermediate particulate loading, estimating the value of the Stokes number can help you select the most appropriate model. The Stokes number can be defined as the relation between the particle response time and the system response time:

$$\mathrm{St} = \frac{\tau_d}{t_s} \tag{16.2-4}$$

where $\tau_d = \frac{\rho_d d_d^2}{18\mu_c}$ and t_s is based on the characteristic length (L_s) and the characteristic velocity (V_s) of the system under investigation: $t_s = \frac{L_s}{V_s}$.

For $\mathrm{St} \ll 1.0$, the particle will follow the flow closely and any of the three models (discrete phase(Chapter 15: Discrete Phase) , mixture, or Eulerian) is applicable; you can therefore choose the least expensive (the mixture model, in most cases), or the most appropriate considering other factors. For $\mathrm{St} > 1.0$, the particles will move independently of the flow and either the discrete phase model (Chapter 15: Discrete Phase) or the Eulerian model is applicable. For $\mathrm{St} \approx 1.0$, again any of the three models is applicable; you can choose the least expensive or the most appropriate considering other factors.

Examples

For a coal classifier with a characteristic length of 1 m and a characteristic velocity of 10 m/s, the Stokes number is 0.04 for particles with a diameter of 30 microns, but 4.0 for particles with a diameter of 300 microns. Clearly the mixture model will not be applicable to the latter case.

For the case of mineral processing, in a system with a characteristic length of 0.2 m and a characteristic velocity of 2 m/s, the Stokes number is 0.005 for particles with a diameter of 300 microns. In this case, you can choose between the mixture and Eulerian models. (The volume fractions are too high for the discrete phase model (Chapter 15: Discrete Phase), as noted below.)

Other Considerations

Keep in mind that the use of the discrete phase model (Chapter 15: Discrete Phase) is limited to low volume fractions. Also, the discrete phase model is the only multiphase model that allows you to specify the particle distribution or include combustion modeling in your simulation.

16.2.3 Time Schemes in Multiphase Flow

In many multiphase applications, the process can vary spatially as well as temporally. In order to accurately model multiphase flow, both higher-order spatial and time discretization schemes are necessary. In addition to the first-order time scheme in ANSYS FLUENT, the second-order time scheme is available in the Mixture and Eulerian multiphase models, and with the VOF Implicit Scheme.

The second-order time scheme cannot be used with the VOF Explicit Schemes.

The second-order time scheme has been adapted to all the transport equations, including mixture phase momentum equations, energy equations, species transport equations, turbulence models, phase volume fraction equations, the pressure correction equation, and the granular flow model. In multiphase flow, a general transport equation (similar to that of Equation 18.3-15) may be written as

$$\frac{\partial(\alpha\rho\phi)}{\partial t} + \nabla \cdot (\alpha\rho\vec{V}\phi) = \nabla \cdot \overline{\overline{\tau}} + S_\phi \tag{16.2-5}$$

Where ϕ is either a mixture (for the mixture model) or a phase variable, α is the phase volume fraction (unity for the mixture equation), ρ is the mixture phase density, $\vec{V}$ is the mixture or phase velocity (depending on the equations), $\overline{\overline{\tau}}$ is the diffusion term, and S_ϕ is the source term.

As a fully implicit scheme, this second-order time-accurate scheme achieves its accuracy by using an Euler backward approximation in time (see Equation 18.3-17). The general transport equation, Equation 16.2-5 is discretized as

$$\frac{3(\alpha_p\rho_p\phi_p Vol)^{n+1} - 4(\alpha_p\rho_p\phi_p Vol)^{n} + (\alpha_p\rho_p\phi_p)^{n-1}}{2\Delta t} = \tag{16.2-6}$$

$$\sum[A_{nb}(\phi_{nb} - \phi_p)]^{n+1} + {S_U}^{n+1} - {S_p}^{n+1}{\phi_p}^{n+1}$$

Equation 16.2-6 can be written in simpler form:

$$A_p\phi_p = \sum A_n b\phi_n b + S_\phi \tag{16.2-7}$$

where

$A_p = \sum {A_{nb}}^{n+1} + {S_p}^{n+1} + \frac{1.5(\alpha_p\rho_p Vol)^{n+1}}{\Delta t}$

$S_\phi = {S_U}^{n+1} + \frac{2(\alpha_p\rho_p\phi_p Vol)^{n} - 0.5(\alpha_p\rho_p\phi_p Vol)^{n-1}}{\Delta t}$

This scheme is easily implemented based on ANSYS FLUENT's existing first-order Euler scheme. It is unconditionally stable, however, the negative coefficient at the time level t_{n-1}, of the three-time level method, may produce oscillatory solutions if the time steps are large.

This problem can be eliminated if a bounded second-order scheme is introduced. However, oscillating solutions are most likely seen in compressible liquid flows. Therefore, in this version of ANSYS FLUENT, a bounded second-order time scheme has been implemented for compressible liquid flows only. For single phase and multiphase compressible liquid flows, the second-order time scheme is, by default, the bounded scheme.

16.2.4 Stability and Convergence

The process of solving a multiphase system is inherently difficult and you may encounter some stability or convergence problems.

When solving a time-dependent problem, a proper initial field is required to avoid instabilities, which usually arise from poor initial fields. If the CPU time is a concern for transient problems, then the best option is to use PC SIMPLE. When body forces are significant, or if the solution requires higher order numerical schemes, it is recommended that you start with a small time step, which can be increased after performing a few time steps to get a better approximation of the pressure field.

For a steady solution, it is recommended that you use the Multiphase Coupled solver, described in detail in Section 24.7.1: Coupled Solution for Multiphase Flows in the separate User's Guide. The iterative nature of this solver requires a good starting patched field.

If difficulties are encountered due to higher order schemes, or due to the complexities of the problem, you may need to reduce the Courant number. The default Courant number is 200 but it can be reduced to as low as 4. This can later be increased if the iteration process runs smoothly. In addition, there are explicit under-relaxation factors for velocities and pressure. All other under-relaxation factors are implicit. Lower under-relaxation factors for the volume of fraction equation may delay the solution dramatically with the Coupled solver (any value 0.5 or above is adequate); on the contrary, PC SIMPLE would normally need a low under-relaxation for the volume fraction equation.

In addition, ANSYS FLUENT offers a Full Multiphase Coupled solver where all velocities, pressure correction and volume fraction correction are solved simultaneously, which currently is not as robust as the others.

Furthermore, ANSYS FLUENT has an option to solve stratified immiscible fluids within the Eulerian multiphase formulation. This feature is similar to the single fluid VOF solution, but in the context of multiple velocities.

16.3 Volume of Fluid (VOF) Model Theory

Information is organized into the following subsections:

16.3.1 Overview and Limitations of the VOF Model

Overview

The VOF model can model two or more immiscible fluids by solving a single set of momentum equations and tracking the volume fraction of each of the fluids throughout the domain. Typical applications include the prediction of jet breakup, the motion of large bubbles in a liquid, the motion of liquid after a dam break, and the steady or transient tracking of any liquid-gas interface.

Limitations

The following restrictions apply to the VOF model in ANSYS FLUENT:

- You must use the pressure-based solver. The VOF model is not available with the density-based solver.
- All control volumes must be filled with either a single fluid phase or a combination of phases. The VOF model does not allow for void regions where no fluid of any type is present.
- Only one of the phases can be defined as a compressible ideal gas. There is no limitation on using compressible liquids using user-defined functions.
- Streamwise periodic flow (either specified mass flow rate or specified pressure drop) cannot be modeled when the VOF model is used.
- The second-order implicit time-stepping formulation cannot be used with the VOF explicit scheme.
- When tracking particles in parallel, the DPM model cannot be used with the VOF model if the shared memory option is enabled (Section 23.8: Parallel Processing for the Discrete Phase Model in the separate User's Guide). (Note that using the message passing option, when running in parallel, enables the compatibility of all multiphase flow models with the DPM model.)

Steady-State and Transient VOF Calculations

The VOF formulation in ANSYS FLUENT is generally used to compute a time-dependent solution, but for problems in which you are concerned only with a steady-state solution, it is possible to perform a steady-state calculation. A steady-state VOF calculation is sensible only when your solution is independent of the initial conditions and there are distinct inflow boundaries for the individual phases. For example, since the shape of the free surface inside a rotating cup depends on the initial level of the fluid, such a problem must be solved using the time-dependent formulation. On the other hand, the flow of water in a channel with a region of air on top and a separate air inlet can be solved with the steady-state formulation.

The VOF formulation relies on the fact that two or more fluids (or phases) are not interpenetrating. For each additional phase that you add to your model, a variable is introduced: the volume fraction of the phase in the computational cell. In each control volume, the volume fractions of all phases sum to unity. The fields for all variables and properties are shared by the phases and represent volume-averaged values, as long as the volume fraction of each of the phases is known at each location. Thus the variables and properties in any given cell are either purely representative of one of the phases, or representative of a mixture of the phases, depending upon the volume fraction values. In other words, if the $q^{\rm th}$ fluid's volume fraction in the cell is denoted as α_q, then the following three conditions are possible:

- $\alpha_q = 0$: The cell is empty (of the $q^{\rm th}$ fluid).
- $\alpha_q = 1$: The cell is full (of the $q^{\rm th}$ fluid).
- $0 < \alpha_q < 1$: The cell contains the interface between the $q^{\rm th}$ fluid and one or more other fluids.

Based on the local value of α_q, the appropriate properties and variables will be assigned to each control volume within the domain.

16.3.2 Volume Fraction Equation

The tracking of the interface(s) between the phases is accomplished by the solution of a continuity equation for the volume fraction of one (or more) of the phases. For the $q^{\rm th}$ phase, this equation has the following form:

$$\frac{1}{\rho_{\rm q}}\left[\frac{\partial}{\partial t}(\alpha_q\rho_q) + \nabla\cdot(\alpha_q\rho_q\vec{v}_q) = S_{\alpha_q} + \sum_{p=1}^{n}(\dot{m}_{\rm pq} - \dot{m}_{\rm qp})\right] \qquad (16.3\text{-}1)$$

where $\dot{m}_{\rm qp}$ is the mass transfer from phase q to phase p and $\dot{m}_{\rm pq}$ is the mass transfer from phase p to phase q. By default, the source term on the right-hand side of Equation 16.3-1,

S_{α_q}, is zero, but you can specify a constant or user-defined mass source for each phase. See Section 16.7: Modeling Mass Transfer in Multiphase Flows for more information on the modeling of mass transfer in ANSYS FLUENT's general multiphase models.

The volume fraction equation will not be solved for the primary phase; the primary-phase volume fraction will be computed based on the following constraint:

$$\sum_{q=1}^{n} \alpha_q = 1 \tag{16.3-2}$$

The volume fraction equation may be solved either through implicit or explicit time discretization.

The Implicit Scheme

When the implicit scheme is used for time discretization, ANSYS FLUENT's standard finite-difference interpolation schemes, QUICK, Second Order Upwind and First Order Upwind, and the Modified HRIC schemes, are used to obtain the face fluxes for all cells, including those near the interface.

$$\frac{\alpha_q^{n+1}\rho_q^{n+1} - \alpha_q^n \rho_q^n}{\Delta t} V + \sum_f (\rho_q^{n+1} U_f^{n+1} \alpha_{q,f}^{n+1}) = \left[S_{\alpha_{\mathrm{q}}} + \sum_{p=1}^{n} (\dot{m}_{\mathrm{pq}} - \dot{m}_{\mathrm{qp}}) \right] V \tag{16.3-3}$$

Since this equation requires the volume fraction values at the current time step (rather than at the previous step, as for the explicit scheme), a standard scalar transport equation is solved iteratively for each of the secondary-phase volume fractions at each time step.

The implicit scheme can be used for both time-dependent and steady-state calculations. See Section 24.2.2: Choosing a Volume Fraction Formulation in the separate User's Guide for details.

The Explicit Scheme

In the explicit approach, ANSYS FLUENT's standard finite-difference interpolation schemes are applied to the volume fraction values that were computed at the previous time step.

$$\frac{\alpha_q^{n+1}\rho_q^{n+1} - \alpha_q^n \rho_q^n}{\Delta t} V + \sum_f (\rho_q U_f^n \alpha_{q,f}^n) = \left[\sum_{p=1}^{n} (\dot{m}_{\mathrm{pq}} - \dot{m}_{\mathrm{qp}}) + S_{\alpha_{\mathrm{q}}} \right] V \tag{16.3-4}$$

where $n+1$ = index for new (current) time step
n = index for previous time step
$\alpha_{q,f}$ = face value of the q^{th} volume fraction, computed from the first- or second-order upwind, QUICK, modified HRIC, or CICSAM scheme
V = volume of cell
U_f = volume flux through the face, based on normal velocity

This formulation does not require iterative solution of the transport equation during each time step, as is needed for the implicit scheme.

i When the explicit scheme is used, a time-dependent solution must be computed.

When the explicit scheme is used for time discretization, the face fluxes can be interpolated either using interface reconstruction or using a finite volume discretization scheme (Section 16.3.2: Interpolation near the Interface). The reconstruction based schemes available in ANSYS FLUENT are Geo-Reconstruct and Donor-Acceptor. The discretization schemes available with explicit scheme for VOF are First Order Upwind, Second Order Upwind, CICSAM, Modified HRIC, and QUICK.

Interpolation near the Interface

ANSYS FLUENT's control-volume formulation requires that convection and diffusion fluxes through the control volume faces be computed and balanced with source terms within the control volume itself.

In the geometric reconstruction and donor-acceptor schemes, ANSYS FLUENT applies a special interpolation treatment to the cells that lie near the interface between two phases. Figure 16.3.1 shows an actual interface shape along with the interfaces assumed during computation by these two methods.

The explicit scheme and the implicit scheme treat these cells with the same interpolation as the cells that are completely filled with one phase or the other (i.e., using the standard upwind (Section 18.3.1: First-Order Upwind Scheme), second-order (Section 18.3.1: Second-Order Upwind Scheme), QUICK (Section 18.3.1: QUICK Scheme, modified HRIC (Section 18.3.1: Modified HRIC Scheme), or CICSAM scheme), rather than applying a special treatment.

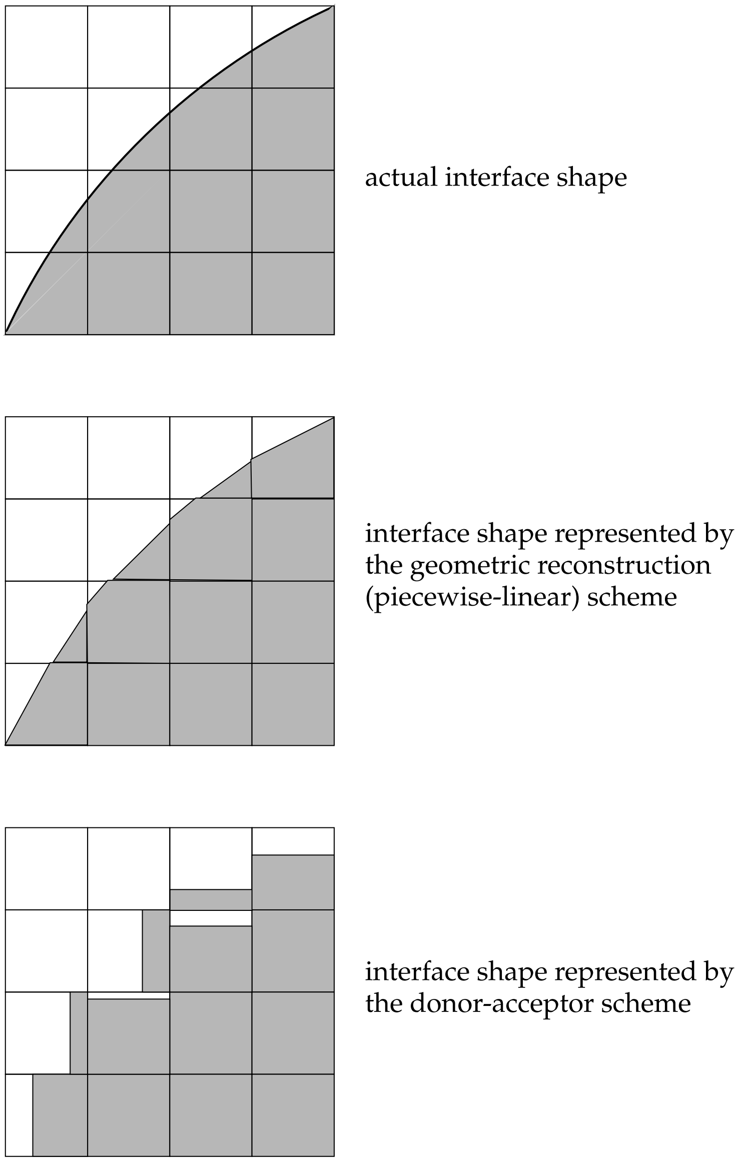

actual interface shape
interface shape represented by the geometric reconstruction (piecewise-linear) scheme
interface shape represented by the donor-acceptor scheme

Figure 16.3.1: Interface Calculations

The Geometric Reconstruction Scheme

In the geometric reconstruction approach, the standard interpolation schemes that are used in ANSYS FLUENT are used to obtain the face fluxes whenever a cell is completely filled with one phase or another. When the cell is near the interface between two phases, the geometric reconstruction scheme is used.

The geometric reconstruction scheme represents the interface between fluids using a piecewise-linear approach. In ANSYS FLUENT this scheme is the most accurate and is applicable for general unstructured meshes. The geometric reconstruction scheme is generalized for unstructured meshes from the work of Youngs [388]. It assumes that the interface between two fluids has a linear slope within each cell, and uses this linear shape for calculation of the advection of fluid through the cell faces. (See Figure 16.3.1.)

The first step in this reconstruction scheme is calculating the position of the linear interface relative to the center of each partially-filled cell, based on information about the volume fraction and its derivatives in the cell. The second step is calculating the advecting amount of fluid through each face using the computed linear interface representation and information about the normal and tangential velocity distribution on the face. The third step is calculating the volume fraction in each cell using the balance of fluxes calculated during the previous step.

i When the geometric reconstruction scheme is used, a time-dependent solution must be computed. Also, if you are using a conformal mesh (i.e., if the mesh node locations are identical at the boundaries where two subdomains meet), you must ensure that there are no two-sided (zero-thickness) walls within the domain. If there are, you will need to slit them, as described in Section 6.8.6: Slitting Face Zones in the separate User's Guide.

The Donor-Acceptor Scheme

In the donor-acceptor approach, the standard interpolation schemes that are used in ANSYS FLUENT are used to obtain the face fluxes whenever a cell is completely filled with one phase or another. When the cell is near the interface between two phases, a "donor-acceptor" scheme is used to determine the amount of fluid advected through the face [131]. This scheme identifies one cell as a donor of an amount of fluid from one phase and another (neighbor) cell as the acceptor of that same amount of fluid, and is used to prevent numerical diffusion at the interface. The amount of fluid from one phase that can be convected across a cell boundary is limited by the minimum of two values: the filled volume in the donor cell or the free volume in the acceptor cell.

The orientation of the interface is also used in determining the face fluxes. The interface orientation is either horizontal or vertical, depending on the direction of the volume fraction gradient of the q^{th} phase within the cell, and that of the neighbor cell that shares the face in question. Depending on the interface's orientation as well as its motion, flux values are obtained by pure upwinding, pure downwinding, or some combination of the two.

i When the donor-acceptor scheme is used, a time-dependent solution must be computed. Also, the donor-acceptor scheme can be used only with quadrilateral or hexahedral meshes. In addition, if you are using a conformal mesh (i.e., if the mesh node locations are identical at the boundaries where two subdomains meet), you must ensure that there are no two-sided (zero-thickness) walls within the domain. If there are, you will need to slit them, as described in Section 6.8.6: Slitting Face Zones in the separate User's Guide.

The Compressive Interface Capturing Scheme for Arbitrary Meshes (CICSAM)

The compressive interface capturing scheme for arbitrary meshes (CICSAM), based on Ubbink's work [351], is a high resolution differencing scheme. The CICSAM scheme is particularly suitable for flows with high ratios of viscosities between the phases. CICSAM is implemented in ANSYS FLUENT as an explicit scheme and offers the advantage of producing an interface that is almost as sharp as the geometric reconstruction scheme.

16.3.3 Material Properties

The properties appearing in the transport equations are determined by the presence of the component phases in each control volume. In a two-phase system, for example, if the phases are represented by the subscripts 1 and 2, and if the volume fraction of the second of these is being tracked, the density in each cell is given by

$$\rho = \alpha_2\rho_2 + (1 - \alpha_2)\rho_1 \tag{16.3-5}$$

In general, for an n-phase system, the volume-fraction-averaged density takes on the following form:

$$\rho = \sum \alpha_q\rho_q \tag{16.3-6}$$

All other properties (e.g., viscosity) are computed in this manner.

16.3.4 Momentum Equation

A single momentum equation is solved throughout the domain, and the resulting velocity field is shared among the phases. The momentum equation, shown below, is dependent on the volume fractions of all phases through the properties ρ and μ.

$$\frac{\partial}{\partial t}(\rho\vec{v}) + \nabla\cdot(\rho\vec{v}\vec{v}) = -\nabla p + \nabla\cdot\left[\mu\left(\nabla\vec{v} + \nabla\vec{v}^{\rm T}\right)\right] + \rho\vec{g} + \vec{F} \tag{16.3-7}$$

One limitation of the shared-fields approximation is that in cases where large velocity differences exist between the phases, the accuracy of the velocities computed near the interface can be adversely affected.

Note that if the viscosity ratio is more than $1x10^3$, this may lead to convergence difficulties. The compressive interface capturing scheme for arbitrary meshes (CICSAM) (Section 16.3.2: The Compressive Interface Capturing Scheme for Arbitrary Meshes (CICSAM)) is suitable for flows with high ratios of viscosities between the phases, thus solving the problem of poor convergence.

16.3.5 Energy Equation

The energy equation, also shared among the phases, is shown below.

$$\frac{\partial}{\partial t}(\rho E) + \nabla\cdot(\vec{v}(\rho E + p)) = \nabla\cdot(k_{\rm eff}\nabla T) + S_h \tag{16.3-8}$$

The VOF model treats energy, E, and temperature, T, as mass-averaged variables:

$$E = \frac{\displaystyle\sum_{q=1}^{n}\alpha_q\rho_q E_q}{\displaystyle\sum_{q=1}^{n}\alpha_q\rho_q} \tag{16.3-9}$$

where E_q for each phase is based on the specific heat of that phase and the shared temperature.

The properties ρ and $k_{\rm eff}$ (effective thermal conductivity) are shared by the phases. The source term, S_h, contains contributions from radiation, as well as any other volumetric heat sources.

As with the velocity field, the accuracy of the temperature near the interface is limited in cases where large temperature differences exist between the phases. Such problems also arise in cases where the properties vary by several orders of magnitude. For example, if a model includes liquid metal in combination with air, the conductivities of the materials can differ by as much as four orders of magnitude. Such large discrepancies in properties lead to equation sets with anisotropic coefficients, which in turn can lead to convergence and precision limitations.

16.3.6 Additional Scalar Equations

Depending upon your problem definition, additional scalar equations may be involved in your solution. In the case of turbulence quantities, a single set of transport equations is solved, and the turbulence variables (e.g., k and ϵ or the Reynolds stresses) are shared by the phases throughout the field.

16.3.7 Time Dependence

For time-dependent VOF calculations, Equation 16.3-1 is solved using an explicit time-marching scheme. ANSYS FLUENT automatically refines the time step for the integration of the volume fraction equation, but you can influence this time step calculation by modifying the Courant number. You can choose to update the volume fraction once for each time step, or once for each iteration within each time step. These options are discussed in more detail in Section 24.3.5: Setting Time-Dependent Parameters for the VOF Model in the separate User's Guide.

16.3.8 Surface Tension and Wall Adhesion

The VOF model can also include the effects of surface tension along the interface between each pair of phases. The model can be augmented by the additional specification of the contact angles between the phases and the walls. You can specify a surface tension coefficient as a constant, as a function of temperature, or through a UDF. The solver will include the additional tangential stress terms (causing what is termed as Marangoni convection) that arise due to the variation in surface tension coefficient. Variable surface tension coefficient effects are usually important only in zero/near-zero gravity conditions.

Surface Tension

Surface tension arises as a result of attractive forces between molecules in a fluid. Consider an air bubble in water, for example. Within the bubble, the net force on a molecule due to its neighbors is zero. At the surface, however, the net force is radially inward, and the combined effect of the radial components of force across the entire spherical surface is to make the surface contract, thereby increasing the pressure on the concave side of the surface. The surface tension is a force, acting only at the surface, that is required to maintain equilibrium in such instances. It acts to balance the radially inward intermolecular attractive force with the radially outward pressure gradient force across the surface. In regions where two fluids are separated, but one of them is not in the form of spherical bubbles, the surface tension acts to minimize free energy by decreasing the area of the interface.

The surface tension model in ANSYS FLUENT is the continuum surface force (CSF) model proposed by Brackbill et al. [34]. With this model, the addition of surface tension to the VOF calculation results in a source term in the momentum equation. To understand the origin of the source term, consider the special case where the surface tension is constant along the surface, and where only the forces normal to the interface are considered. It can be shown that the pressure drop across the surface depends upon the surface tension coefficient, σ, and the surface curvature as measured by two radii in orthogonal directions, R_1 and R_2:

$$p_2 - p_1 = \sigma \left(\frac{1}{R_1} + \frac{1}{R_2} \right) \tag{16.3-10}$$

where p_1 and p_2 are the pressures in the two fluids on either side of the interface.

In ANSYS FLUENT, a formulation of the CSF model is used, where the surface curvature is computed from local gradients in the surface normal at the interface. Let n be the surface normal, defined as the gradient of α_q, the volume fraction of the q^{th} phase.

$$n = \nabla \alpha_q \tag{16.3-11}$$

The curvature, κ, is defined in terms of the divergence of the unit normal, $\hat{n}$ [34]:

$$\kappa = \nabla \cdot \hat{n} \tag{16.3-12}$$

where

$$\hat{n} = \frac{n}{|n|} \tag{16.3-13}$$

The surface tension can be written in terms of the pressure jump across the surface. The force at the surface can be expressed as a volume force using the divergence theorem. It is this volume force that is the source term which is added to the momentum equation. It has the following form:

$$F_{\text{vol}} = \sum_{\text{pairs } ij,\, i<j} \sigma_{ij} \frac{\alpha_i \rho_i \kappa_j \nabla \alpha_j + \alpha_j \rho_j \kappa_i \nabla \alpha_i}{\frac{1}{2}(\rho_i + \rho_j)} \tag{16.3-14}$$

This expression allows for a smooth superposition of forces near cells where more than two phases are present. If only two phases are present in a cell, then $\kappa_i = -\kappa_j$ and $\nabla \alpha_i = -\nabla \alpha_j$, and Equation 16.3-14 simplifies to

$$F_{\text{vol}} = \sigma_{ij} \frac{\rho \kappa_i \nabla \alpha_i}{\frac{1}{2}(\rho_i + \rho_j)} \tag{16.3-15}$$

where ρ is the volume-averaged density computed using Equation 16.3-6. Equation 16.3-15 shows that the surface tension source term for a cell is proportional to the average density in the cell.

Note that the calculation of surface tension effects on triangular and tetrahedral meshes is not as accurate as on quadrilateral and hexahedral meshes. The region where surface tension effects are most important should therefore be meshed with quadrilaterals or hexahedra.

When Surface Tension Effects Are Important

The importance of surface tension effects is determined based on the value of two dimensionless quantities: the Reynolds number, Re, and the capillary number, Ca; or the Reynolds number, Re, and the Weber number, We. For Re $\ll$ 1, the quantity of interest is the capillary number:

$$\text{Ca} = \frac{\mu U}{\sigma} \tag{16.3-16}$$

and for Re $\gg$ 1, the quantity of interest is the Weber number:

$$\text{We} = \frac{\rho L U^2}{\sigma} \tag{16.3-17}$$

where U is the free-stream velocity. Surface tension effects can be neglected if Ca $\gg$ 1 or We $\gg$ 1.

To include the effects of surface tension in your model, refer to Section 24.3.4: Including Surface Tension and Wall Adhesion Effects in the separate User's Guide.

Wall Adhesion

An option to specify a wall adhesion angle in conjunction with the surface tension model is also available in the VOF model. The model is taken from work done by Brackbill et al. [34]. Rather than impose this boundary condition at the wall itself, the contact angle that the fluid is assumed to make with the wall is used to adjust the surface normal in cells near the wall. This so-called dynamic boundary condition results in the adjustment of the curvature of the surface near the wall.

If θ_w is the contact angle at the wall, then the surface normal at the live cell next to the wall is

$$\hat{n} = \hat{n}_w \cos\theta_w + \hat{t}_w \sin\theta_w \tag{16.3-18}$$

where $\hat{n}_w$ and $\hat{t}_w$ are the unit vectors normal and tangential to the wall, respectively. The combination of this contact angle with the normally calculated surface normal one cell away from the wall determine the local curvature of the surface, and this curvature is used to adjust the body force term in the surface tension calculation.

To include wall adhesion in your model, refer to Section 24.3.4: Including Surface Tension and Wall Adhesion Effects in the separate User's Guide.

16.3.9 Open Channel Flow

ANSYS FLUENT can model the effects of open channel flow (e.g., rivers, dams, and surface-piercing structures in unbounded stream) using the VOF formulation and the open channel boundary condition. These flows involve the existence of a free surface between the flowing fluid and fluid above it (generally the atmosphere). In such cases, the wave propagation and free surface behavior becomes important. Flow is generally governed by the forces of gravity and inertia. This feature is mostly applicable to marine applications and the analysis of flows through drainage systems.

Open channel flows are characterized by the dimensionless Froude Number, which is defined as the ratio of inertia force and hydrostatic force.

$$Fr = \frac{V}{\sqrt{gy}} \tag{16.3-19}$$

where V is the velocity magnitude, g is gravity, and y is a length scale, in this case, the distance from the bottom of the channel to the free surface. The denominator in Equation 16.3-19 is the propagation speed of the wave. The wave speed as seen by the fixed observer is defined as

$$V_{\mathrm{w}} = V \pm \sqrt{gy} \tag{16.3-20}$$

Based on the Froude number, open channel flows can be classified in the following three categories:

- When $Fr < 1$, i.e., $V < \sqrt{gy}$ (thus $V_{\mathrm{w}} < 0$ or $V_{\mathrm{w}} > 0$), the flow is known to be *subcritical* where disturbances can travel upstream as well as downstream. In this case, downstream conditions might affect the flow upstream.
- When $Fr = 1$ (thus $V_{\mathrm{w}} = 0$), the flow is known to be critical, where upstream propagating waves remain stationary. In this case, the character of the flow changes.
- When $Fr > 1$, i.e., $V > \sqrt{gy}$ (thus $V_{\mathrm{w}} > 0$), the flow is known to be *supercritical* where disturbances cannot travel upstream. In this case, downstream conditions do not affect the flow upstream.

Upstream Boundary Conditions

There are two options available for the upstream boundary condition for open channel flows:

- pressure inlet
- mass flow rate

Pressure Inlet

The total pressure p_0 at the inlet can be given as

$$p_0 = \frac{1}{2}(\rho - \rho_0)V^2 + (\rho - \rho_0)|\overrightarrow{g}|(\hat{g} \cdot (\overrightarrow{b} - \overrightarrow{a})) \tag{16.3-21}$$

where $\overrightarrow{b}$ and $\overrightarrow{a}$ are the position vectors of the face centroid and any point on the free surface, respectively, Here, free surface is assumed to be horizontal and normal to the direction of gravity. $\overrightarrow{g}$ is the gravity vector, $|\overrightarrow{g}|$ is the gravity magnitude, $\hat{g}$ is the unit vector of gravity, V is the velocity magnitude, ρ is the density of the mixture in the cell, and ρ_0 is the reference density.

From this, the dynamic pressure q is

$$q = \frac{\rho - \rho_0}{2}V^2 \tag{16.3-22}$$

and the static pressure p_{s} is

$$p_{\mathrm{s}} = (\rho - \rho_0)|\overrightarrow{g}|(\hat{g} \cdot (\overrightarrow{b} - \overrightarrow{a})) \tag{16.3-23}$$

which can be further expanded to

$$p_{\mathrm{s}} = (\rho - \rho_0)|\overrightarrow{g}|((\hat{g} \cdot \overrightarrow{b}) + y_{\mathrm{local}}) \qquad (16.3\text{-}24)$$

where the distance from the free surface to the reference position, y_{local}, is

$$y_{\mathrm{local}} = -(\overrightarrow{a} \cdot \hat{g}) \qquad (16.3\text{-}25)$$

Mass Flow Rate

The mass flow rate for each phase associated with the open channel flow is defined by

$$\dot{m}_{\mathrm{phase}} = \rho_{\mathrm{phase}}(Area_{\mathrm{phase}})(Velocity) \qquad (16.3\text{-}26)$$

Volume Fraction Specification

In open channel flows, ANSYS FLUENT internally calculates the volume fraction based on the input parameters specified in the boundary conditions dialog box, therefore this option has been disabled.

For subcritical inlet flows (Fr $<$ 1), ANSYS FLUENT reconstructs the volume fraction values on the boundary by using the values from the neighboring cells. This can be accomplished using the following procedure:

- Calculate the node values of volume fraction at the boundary using the cell values.
- Calculate the volume fraction at the each face of boundary using the interpolated node values.

For supercritical inlet flows (Fr $>$ 1), the volume fraction value on the boundary can be calculated using the fixed height of the free surface from the bottom.

Downstream Boundary Conditions

Pressure Outlet

Determining the static pressure is dependent on the Pressure Specification Method. Using the Free Surface Level, the static pressure is dictated by Equation 16.3-23 and Equation 16.3-25, otherwise you must specify the static pressure as the Gauge Pressure.

For subcritical outlet flows (Fr $<$ 1), if there are only two phases, then the pressure is taken from the pressure profile specified over the boundary, otherwise the pressure is taken from the neighboring cell. For supercritical flows (Fr $>$1), the pressure is always taken from the neighboring cell.

Outflow Boundary

Outflow boundary conditions can be used at the outlet of open channel flows to model flow exits where the details of the flow velocity and pressure are not known prior to solving the flow problem. If the conditions are unknown at the outflow boundaries, then ANSYS FLUENT will extrapolate the required information from the interior.

It is important, however, to understand the limitations of this boundary type:

- You can only use single outflow boundaries at the outlet, which is achieved by setting the flow rate weighting to 1. In other words, outflow splitting is not permitted in open channel flows with outflow boundaries.
- There should be an initial flow field in the simulation to avoid convergence issues due to flow reversal at the outflow, which will result in an unreliable solution.
- An outflow boundary condition can only be used with mass flow inlets. It is not compatible with pressure inlets and pressure outlets. For example, if you choose the inlet as pressure-inlet, then you can only use pressure-outlet at the outlet. If you choose the inlet as mass-flow-inlet, then you can use either outflow or pressure-outlet boundary conditions at the outlet. Note that this only holds true for open channel flow.
- Note that the outflow boundary condition assumes that flow is fully developed in the direction perpendicular to the outflow boundary surface. Therefore, such surfaces should be placed accordingly.

Backflow Volume Fraction Specification

ANSYS FLUENT internally calculates the volume fraction values on the outlet boundary by using the neighboring cell values, therefore, this option is disabled.

16.3.10 Open Channel Wave Boundary Conditions

The open channel wave boundary condition allows you to simulate the propagation of waves, which is useful in the marine industry. This is an upstream boundary condition and is applied to the velocity inlet of the VOF model. The wave profile for an incident wave can be described as follows:

$$\zeta = A\cos(k_x x + k_y y - \omega_e t + \epsilon) \tag{16.3-27}$$

where z is the wave height, A is the wave amplitude, ϵ is the phase difference, t is the time, and k_x and k_y are the wave numbers in the x and y directions, respectively, such that $k_x = k\cos\theta$ and $k_y = k\sin\theta$.

The wave number k is defined as

$$k = \frac{2\pi}{\lambda} \tag{16.3-28}$$

where λ is the wave length and the effective wave frequency ω_e is

$$\omega_e = \omega + kU \tag{16.3-29}$$

U is the uniform incident wave velocity and ω, the wave frequency, is defined as

$$\omega = \sqrt{gk \tanh(kh)} \tag{16.3-30}$$

where h is the liquid height and g is the gravity magnitude.

The velocity components for the incident wave boundary condition can be described in terms of shallow waves and short gravity waves.

Shallow waves are defined as

$$\begin{pmatrix} u \\ v \end{pmatrix} = \frac{gkA}{\omega} \frac{\cosh[k(z+h)]}{\cosh(kh)} \begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix} \cos(k_x x + k_y y - \omega_e t + \epsilon) \tag{16.3-31}$$

$$w = \frac{gkA}{\omega} \frac{\sinh[k(z+h)]}{\cosh(kh)} \sin(k_x x + k_y y - \omega_e t + \epsilon) \tag{16.3-32}$$

Short gravity waves are defined as

$$\begin{pmatrix} u \\ v \end{pmatrix} = \frac{gkA}{\omega} e^{kz} \begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix} \cos(k_x x + k_y y - \omega_e t + \epsilon) \tag{16.3-33}$$

$$w = \frac{gkA}{\omega} e^{kz} \sin(k_x x + k_y y - \omega_e t + \epsilon) \tag{16.3-34}$$

where u, v, and w are the velocity components. Note that direction specifications for the velocity components are such that u is based on the flow direction specified in the wave velocity specification method, w is based on the gravity direction, and v is in the cross direction of the flow and gravity direction. For more information on how to use and set up this model, refer to Section 24.3.2: Modeling Open Channel Wave Boundary Conditions in the separate User's Guide.

16.4 Mixture Model Theory

Information is organized into the following subsections:

16.4.1 Overview and Limitations of the Mixture Model

Overview

The mixture model is a simplified multiphase model that can be used in different ways. It can be used to model multiphase flows where the phases move at different velocities, but assume local equilibrium over short spatial length scales. It can be used to model homogeneous multiphase flows with very strong coupling and phases moving at the same velocity and lastly, the mixture models are used to calculate non-Newtonian viscosity.

The mixture model can model n phases (fluid or particulate) by solving the momentum, continuity, and energy equations for the mixture, the volume fraction equations for the secondary phases, and algebraic expressions for the relative velocities. Typical applications include sedimentation, cyclone separators, particle-laden flows with low loading, and bubbly flows where the gas volume fraction remains low.

The mixture model is a good substitute for the full Eulerian multiphase model in several cases. A full multiphase model may not be feasible when there is a wide distribution of the particulate phase or when the interphase laws are unknown or their reliability can be questioned. A simpler model like the mixture model can perform as well as a full multiphase model while solving a smaller number of variables than the full multiphase model.

The mixture model allows you to select granular phases and calculates all properties of the granular phases. This is applicable for liquid-solid flows.

Limitations

The following limitations apply to the mixture model in ANSYS FLUENT:

- You must use the pressure-based solver. The mixture model is not available with the density-based solver.
- Only one of the phases can be defined as a compressible ideal gas. There is no limitation on using compressible liquids using user-defined functions.
- When the mixture model is used, do not model streamwise periodic flow with specified mass flow rate.
- Do not model solidification and melting in conjunction with the mixture model.
- The Singhal et al. cavitation model (available with the mixture model) is not compatible with the LES turbulence model.
- Do not use the relative formulation in combination with the MRF and mixture model (see Section 10.3.1: Limitations in the separate User's Guide).
- The mixture model does not allow for inviscid flows.
- The shell conduction model for walls is not allowed with the mixture model.
- When tracking particles in parallel, do not use the DPM model with the mixture model if the shared memory option is enabled (Section 23.8: Parallel Processing for the Discrete Phase Model in the separate User's Guide). (Note that using the message passing option, when running in parallel, enables the compatibility of all multiphase flow models with the DPM model.)

The mixture model, like the VOF model, uses a *single-fluid* approach. It differs from the VOF model in two respects:

- The mixture model allows the phases to be interpenetrating. The volume fractions α_q and α_p for a control volume can therefore be equal to any value between 0 and 1, depending on the space occupied by phase q and phase p.
- The mixture model allows the phases to move at different velocities, using the concept of slip velocities. (Note that the phases can also be assumed to move at the same velocity, and the mixture model is then reduced to a homogeneous multiphase model.)

The mixture model solves the continuity equation for the mixture, the momentum equation for the mixture, the energy equation for the mixture, and the volume fraction equation for the secondary phases, as well as algebraic expressions for the relative velocities (if the phases are moving at different velocities).

16.4.2 Continuity Equation

The continuity equation for the mixture is

$$\frac{\partial}{\partial t}(\rho_m) + \nabla \cdot (\rho_m \vec{v}_m) = 0 \tag{16.4-1}$$

where $\vec{v}_m$ is the mass-averaged velocity:

$$\vec{v}_m = \frac{\sum_{k=1}^{n} \alpha_k \rho_k \vec{v}_k}{\rho_m} \tag{16.4-2}$$

and ρ_m is the mixture density:

$$\rho_m = \sum_{k=1}^{n} \alpha_k \rho_k \tag{16.4-3}$$

α_k is the volume fraction of phase k.

16.4.3 Momentum Equation

The momentum equation for the mixture can be obtained by summing the individual momentum equations for all phases. It can be expressed as

$$\begin{aligned}\frac{\partial}{\partial t}(\rho_m \vec{v}_m) + \nabla \cdot (\rho_m \vec{v}_m \vec{v}_m) = -\nabla p \ + \nabla \cdot \left[\mu_m \left(\nabla \vec{v}_m + \nabla \vec{v}_m^{\mathrm{T}}\right)\right] + \\ \rho_m \vec{g} + \vec{F} + \nabla \cdot \left(\sum_{k=1}^{n} \alpha_k \rho_k \vec{v}_{\mathrm{dr},k} \vec{v}_{\mathrm{dr},k}\right)\end{aligned} \tag{16.4-4}$$

where n is the number of phases, $\vec{F}$ is a body force, and μ_m is the viscosity of the mixture:

$$\mu_m = \sum_{k=1}^{n} \alpha_k \mu_k \tag{16.4-5}$$

$\vec{v}_{\mathrm{dr},k}$ is the drift velocity for secondary phase k:

$$\vec{v}_{\mathrm{dr},k} = \vec{v}_k - \vec{v}_m \tag{16.4-6}$$

16.4.4 Energy Equation

The energy equation for the mixture takes the following form:

$$\frac{\partial}{\partial t}\sum_{k=1}^{n}(\alpha_k\rho_k E_k) + \nabla\cdot\sum_{k=1}^{n}(\alpha_k\vec{v}_k(\rho_k E_k + p)) = \nabla\cdot(k_{\rm eff}\nabla T) + S_E \qquad (16.4\text{-}7)$$

where $k_{\rm eff}$ is the effective conductivity ($\sum\alpha_k(k_k + k_t)$), where k_t is the turbulent thermal conductivity, defined according to the turbulence model being used). The first term on the right-hand side of Equation 16.4-7 represents energy transfer due to conduction. S_E includes any other volumetric heat sources.

In Equation 16.4-7,

$$E_k = h_k - \frac{p}{\rho_k} + \frac{v_k^2}{2} \qquad (16.4\text{-}8)$$

for a compressible phase, and $E_k = h_k$ for an incompressible phase, where h_k is the sensible enthalpy for phase k.

16.4.5 Relative (Slip) Velocity and the Drift Velocity

The relative velocity (also referred to as the slip velocity) is defined as the velocity of a secondary phase (p) relative to the velocity of the primary phase (q):

$$\vec{v}_{pq} = \vec{v}_p - \vec{v}_q \qquad (16.4\text{-}9)$$

The mass fraction for any phase (k) is defined as

$$c_k = \frac{\alpha_k\rho_k}{\rho_m} \qquad (16.4\text{-}10)$$

The drift velocity and the relative velocity ($\vec{v}_{qp}$) are connected by the following expression:

$$\vec{v}_{{\rm dr},p} = \vec{v}_{pq} - \sum_{k=1}^{n} c_k\vec{v}_{qk} \qquad (16.4\text{-}11)$$

ANSYS FLUENT's mixture model makes use of an algebraic slip formulation. The basic assumption of the algebraic slip mixture model is that to prescribe an algebraic relation for the relative velocity, a local equilibrium between the phases should be reached over a short spatial length scale. Following Manninen et al. [217], the form of the relative velocity is given by:

$$\vec{v}_{pq} = \frac{\tau_p}{f_{\rm drag}} \frac{(\rho_p - \rho_m)}{\rho_p} \vec{a} \tag{16.4-12}$$

where τ_p is the particle relaxation time

$$\tau_p = \frac{\rho_p d_p^2}{18 \mu_q} \tag{16.4-13}$$

d is the diameter of the particles (or droplets or bubbles) of secondary phase p, $\vec{a}$ is the secondary-phase particle's acceleration. The default drag function $f_{\rm drag}$ is taken from Schiller and Naumann [305]:

$$f_{\rm drag} = \begin{cases} 1 + 0.15\ {\rm Re}^{0.687} & {\rm Re} \leq 1000 \\ 0.0183\ {\rm Re} & {\rm Re} > 1000 \end{cases} \tag{16.4-14}$$

and the acceleration $\vec{a}$ is of the form

$$\vec{a} = \vec{g} - (\vec{v}_m \cdot \nabla)\vec{v}_m - \frac{\partial \vec{v}_m}{\partial t} \tag{16.4-15}$$

The simplest algebraic slip formulation is the so-called drift flux model, in which the acceleration of the particle is given by gravity and/or a centrifugal force and the particulate relaxation time is modified to take into account the presence of other particles.

In turbulent flows the relative velocity should contain a diffusion term due to the dispersion appearing in the momentum equation for the dispersed phase. ANSYS FLUENT adds this dispersion to the relative velocity:

$$\vec{v}_{pq} = \frac{(\rho_p - \rho_m) d_p^2}{18 \mu_q f_{\rm drag}} \vec{a} - \frac{\eta_t}{\sigma_t} \left(\frac{\nabla \alpha_p}{\alpha_p} - \frac{\nabla \alpha_q}{\alpha_q} \right) \tag{16.4-16}$$

where σ_t is a Prandtl/Schmidt number set to 0.75 and η_t is the turbulent diffusivity. This diffusivity is calculated from the continuous-dispersed fluctuating velocity correlation, such that

$$\eta_t = C_\mu \frac{k^2}{\epsilon} \left(\frac{\gamma_\gamma}{1 + \gamma_\gamma} \right] (1 + C_\beta \zeta_\gamma^2)^{-1/2} \tag{16.4-17}$$

$$\zeta_\gamma = \frac{|\vec{v}_{pq}|}{\sqrt{2/3k}} \tag{16.4-18}$$

where

$$C_\beta = 1.8 - 1.35 \cos^2 \theta$$

and

$$\cos\theta = \frac{\vec{v}_{pq} \cdot \vec{v}_p}{|\vec{v}_{pq}||\vec{v}_p|}$$

γ_γ is the time ratio between the time scale of the energetic turbulent eddies affected by the crossing-trajectories effect and the particle relaxation time.

When you are solving a mixture multiphase calculation with slip velocity, you can directly prescribe formulations for the drag function. The following choices are available:

- Schiller-Naumann (the default formulation)
- Morsi-Alexander
- symmetric
- constant
- user-defined

See Section 16.5.4: Interphase Exchange Coefficients for more information on these drag functions and their formulations, and Section 24.4.1: Defining the Phases for the Mixture Model in the separate User's Guide for instructions on how to enable them.

Note that, if the slip velocity is not solved, the mixture model is reduced to a homogeneous multiphase model. In addition, the mixture model can be customized (using user-defined functions) to use a formulation other than the algebraic slip method for the slip velocity. See the separate UDF Manual for details.

16.4.6 Volume Fraction Equation for the Secondary Phases

From the continuity equation for secondary phase p, the volume fraction equation for secondary phase p can be obtained:

$$\frac{\partial}{\partial t}(\alpha_p \rho_p) + \nabla \cdot (\alpha_p \rho_p \vec{v}_m) = -\nabla \cdot (\alpha_p \rho_p \vec{v}_{\mathrm{dr},p}) + \sum_{q=1}^{n} (\dot{m}_{qp} - \dot{m}_{pq}) \tag{16.4-19}$$

16.4.7 Granular Properties

Since the concentration of particles is an important factor in the calculation of the effective viscosity for the mixture, we may use the granular viscosity to get a value for the viscosity of the suspension. The volume weighted averaged for the viscosity would now contain shear viscosity arising from particle momentum exchange due to translation and collision.

The collisional and kinetic parts, and the optional frictional part, are added to give the solids shear viscosity:

$$\mu_s = \mu_{s,\mathrm{col}} + \mu_{s,\mathrm{kin}} + \mu_{s,\mathrm{fr}} \tag{16.4-20}$$

Collisional Viscosity

The collisional part of the shear viscosity is modeled as [110, 343]

$$\mu_{s,\mathrm{col}} = \frac{4}{5}\alpha_s \rho_s d_s g_{0,ss}(1+e_{ss})\left(\frac{\Theta_s}{\pi}\right)^{1/2}\alpha_s \tag{16.4-21}$$

Kinetic Viscosity

ANSYS FLUENT provides two expressions for the kinetic viscosity.

The default expression is from Syamlal et al. [343]:

$$\mu_{s,\mathrm{kin}} = \frac{\alpha_s d_s \rho_s \sqrt{\Theta_s \pi}}{6\left(3-e_{ss}\right)}\left[1+\frac{2}{5}\left(1+e_{ss}\right)\left(3e_{ss}-1\right)\alpha_s g_{0,ss}\right] \tag{16.4-22}$$

The following optional expression from Gidaspow et al. [110] is also available:

$$\mu_{s,\mathrm{kin}} = \frac{10\rho_s d_s \sqrt{\Theta_s \pi}}{96\alpha_s\left(1+e_{ss}\right)g_{0,ss}}\left[1+\frac{4}{5}g_{0,ss}\alpha_s\left(1+e_{ss}\right)\right]^2\alpha_s \tag{16.4-23}$$

16.4.8 Granular Temperature

The viscosities need the specification of the granular temperature for the s^{th} solids phase. Here we use an algebraic equation from the granular temperature transport equation. This is only applicable for dense fluidized beds where the convection and the diffusion term can be neglected under the premise that production and dissipation of granular energy are in equilibrium.

$$0 = (-p_s \overline{\overline{I}} + \overline{\overline{\tau}}_s) : \nabla \vec{v}_s - \gamma_{\Theta_s} + \phi_{ls} \tag{16.4-24}$$

where

$(-p_s \overline{\overline{I}} + \overline{\overline{\tau}}_s) : \nabla \vec{v}_s$ = the generation of energy by the solid stress tensor
γ_{Θ_s} = the collisional dissipation of energy
ϕ_{ls} = the energy exchange between the l^{th} fluid or solid phase and the s^{th} solid phase

The collisional dissipation of energy, γ_{Θ_s}, represents the rate of energy dissipation within the s^{th} solids phase due to collisions between particles. This term is represented by the expression derived by Lun et al. [208]

$$\gamma_{\Theta m} = \frac{12(1-e_{ss}^2)g_{0,ss}}{d_s\sqrt{\pi}} \rho_s \alpha_s^2 \Theta_s^{3/2} \tag{16.4-25}$$

The transfer of the kinetic energy of random fluctuations in particle velocity from the s^{th} solids phase to the l^{th} fluid or solid phase is represented by ϕ_{ls} [110]:

$$\phi_{ls} = -3K_{ls}\Theta_s \tag{16.4-26}$$

ANSYS FLUENT allows you to solve for the granular temperature with the following options:

- algebraic formulation (the default)

  This is obtained by neglecting convection and diffusion in the transport equation (Equation 16.4-24) [343].

- constant granular temperature

  This is useful in very dense situations where the random fluctuations are small.

- UDF for granular temperature

16.4.9 Interfacial Area Concentration

Interfacial area concentration is defined as the interfacial area between two phases per unit mixture volume. This is an important parameter for predicting mass, momentum and energy transfers through the interface between the phases. In two-fluid flow systems, one discrete (particles) and one continuous, the size and its distribution of the discrete phase or particles can change rapidly due to growth (mass transfer between phases), expansion due to pressure changes, coalescence, breakage and/or nucleation mechanisms. The Population Balance model (see the separate Population Balance Module Manual) ideally captures this phenomenon, but is computationally expensive since several transport equations need to be solved using moment methods, or more if the discrete method is used. The interfacial area concentration model uses a single transport equation per secondary phase and is specific to bubbly flows in liquid at this stage.

The transport equation for the interfacial area concentration can be written as

$$\frac{\partial(\rho_g \chi_p)}{\partial t} + \nabla \cdot (\rho_g \vec{u}_g \chi_p) = \frac{1}{3}\frac{D\rho_g}{Dt}\chi_p + \frac{2}{3}\frac{\dot{m}_g}{\alpha_g}\chi_p + \rho_g(S_{RC} + S_{WE} + S_{TI}) \tag{16.4-27}$$

where χ_p is the interfacial area concentration (m^2/m^3), and α_g is the gas volume fraction. The first two terms on the right hand side of Equation 16.4-27 are of gas bubble expansion due to compressibility and mass transfer (phase change). $\dot{m}_g$ is the mass transfer rate into the gas phase per unit mixture volume ($kg/m^3/s$). S_{RC} and S_{WE} are the coalescence sink terms due to random collision and wake entrainment, respectively. S_{TI} is the breakage source term due to turbulent impact.

Two sets of models, the Hibiki-Ishii model [129] and the Ishii-Kim model [281, 139], exist for those source and sink terms for the interfacial area concentration, which are based on the works of Ishii et al. [129, 281]. According to their study, the mechanisms of interactions can be summarized in five categories:

- Coalescence due to random collision driven by turbulence.
- Breakage due to the impact of turbulent eddies.
- Coalescence due to wake entrainment.
- Shearing-off of small bubbles from large cap bubbles.
- Breakage of large cap bubbles due to flow instability on the bubble surface.

In ANSYS FLUENT, only the first three effects will be considered.

Hibiki-Ishii Model

$$\begin{aligned} S_{RC} &= -\frac{1}{3\phi}(\frac{\alpha_g}{\chi_p})^2 f_c n_b \lambda_c \\ &= -(\frac{\alpha_g}{\chi_p})^2 \frac{\Gamma_c {\alpha_g}^2 \epsilon^{1/3}}{{d_b}^{11/3}(\alpha_{gmax}-\alpha_g)} exp(-K_c \frac{{d_b}^{5/6}{\rho_f}^{1/2}\epsilon^{1/3}}{\sigma^{1/2}}) \\ &= -\frac{\Gamma_c}{\psi^{11/3}} \frac{\epsilon^{1/3}}{(\alpha_{gmax}-\alpha_g)} {\alpha_g}^{1/3}{\chi_p}^{5/3} exp[-K_c \psi^{5/6} \frac{{\rho_f}^{1/2}\epsilon^{1/3}}{\sigma^{1/2}} (\frac{\alpha_g}{\chi_p})^{5/6}] \end{aligned} \tag{16.4-28}$$

where f_c, λ_c and n_b are the frequency of particle/bubble collision, the efficiency of coalescence from the collision, and the number of particles per unit mixture volume, respectively. The averaged size of the particle/bubble d_b is assumed to be calculated as

$$d_b = \psi \frac{\alpha_g}{\chi_p} \tag{16.4-29}$$

and

$$\lambda_c = exp(-K_c \frac{{d_b}^{5/6}{\rho_f}^{1/2}\epsilon^{1/3}}{\sigma^{1/2}}) \tag{16.4-30}$$

$$S_{TI} = \frac{1}{3\phi}(\frac{\alpha_g}{\chi_p})^2 f_B n_e \lambda_B \tag{16.4-31}$$

$$\begin{aligned} &= (\frac{\alpha_g}{\chi_p})^2 \frac{\Gamma_B \alpha_g (1-\alpha_g)\epsilon^{1/3}}{{d_b}^{11/3}(\alpha_{gmax}-\alpha_g)} exp(-K_B \frac{\sigma}{\rho_f {d_b}^{5/3}\epsilon^{2/3}}) \\ &= \frac{\Gamma_B}{\psi^{11/3}} \frac{(1-\alpha_g)\epsilon^{1/3}{\chi_p}^{5/3}}{{\alpha_g}^{2/3}(\alpha_{gmax}-\alpha_g)} exp[-\frac{K_B}{\psi^{5/3}} \frac{\sigma}{\rho_f \epsilon^{2/3}} (\frac{\chi_p}{\alpha_g})^{5/3}] \end{aligned}$$

where f_B, λ_B and n_e are the frequency of collision between particles/bubbles and turbulent eddies of the primary phase, the efficiency of breakage from the impact, and the number of turbulent eddies per unit mixture volume, respectively. In Equation 16.4-31

$$\lambda_B = exp(-K_B \frac{\sigma}{\rho_f {d_b}^{5/3}\epsilon^{2/3}}) \tag{16.4-32}$$

The experimental adjustable coefficients are given as follows:

$\Gamma_C = 0.188$; $K_C = 0.129$; $\Gamma_B = 0.264$; $K_B = 1.37$.

The shape factor ψ is given as 6 and ϕ as $\frac{1}{36\pi}$ for spherical particles/bubbles. There is no model for S_{WE} in the Hibiki-Ishii formulation.

Ishii-Kim Model

$$\begin{aligned} S_{RC} &= -\frac{1}{3\phi}(\frac{\alpha_g}{\chi_p})^2 C_{RC}[\frac{{n_b}^2 u_t {d_b}^2}{{\alpha_{gmax}}^{1/3}({\alpha_{gmax}}^{1/3} - {\alpha_g}^{1/3})}] \\ & \quad [1 - exp(-C\frac{{\alpha_{gmax}}^{1/3}{\alpha_g}^{1/3}}{{\alpha_{gmax}}^{1/3} - {\alpha_g}^{1/3}})] \\ &= -\frac{1}{3\pi} C_{RC} u_t {\chi_p}^2 [\frac{1}{{\alpha_{gmax}}^{1/3}({\alpha_{gmax}}^{1/3} - {\alpha_g}^{1/3})}][1 - exp(-C\frac{{\alpha_{gmax}}^{1/3}{\alpha_g}^{1/3}}{{\alpha_{gmax}}^{1/3} - {\alpha_g}^{1/3}})] \end{aligned} \tag{16.4-33}$$

$$S_{WE} = -\frac{1}{3\phi}(\frac{\alpha_g}{\chi_p})^2 {n_b}^2 {d_b}^2 u_r {C_D}^{1/3} = -\frac{1}{3\pi} C_{WE} u_r {\chi_p}^2 {C_D}^{1/3} \tag{16.4-34}$$

$$\begin{aligned} S_{TI} &= \frac{1}{3\phi}(\frac{\alpha_g}{\chi_p})^2 C_{TI}(\frac{n_b u_t}{d_b})(1 - \frac{We_{cr}}{We})^{1/2} exp(-\frac{We_{cr}}{We}) \\ &= \frac{1}{18} C_{TI} u_t \frac{{\chi_p}^2}{\alpha_g}(1 - \frac{We_{cr}}{We})^{1/2} exp(-\frac{We_{cr}}{We}) \end{aligned} \tag{16.4-35}$$

where the mean bubble fluctuating velocity, u_t , is given by $\epsilon^{1/3}{d_b}^{1/3}$. The bubble terminal velocity, u_r , is a function of the bubble diameter and local time-averaged void fraction.

$$u_r = (\frac{d_b g \Delta\rho}{3 C_D \rho_f})^{1/2} \tag{16.4-36}$$

$$C_D = 24\frac{(1 + 0.1{Re_D}^{0.75})}{Re_D} \quad and \quad Re_D \equiv \frac{\rho_f u_r d_b}{\mu_f}(1 - \alpha_g) \tag{16.4-37}$$

$$We = \frac{\rho_f {u_t}^2 d_b}{\sigma} \tag{16.4-38}$$

where μ_f is the molecular viscosity of the fluid phase, g is the gravitational acceleration and σ is the interfacial tension. In this model, when the Weber number, We, is less than the critical Weber number, We_{cr}, the breakage rate equals zero, i.e. $S_{TI} = 0$. The coefficients used are given as follows [139]:

$$\begin{aligned} C_{RC} &= 0.004 \\ C_{WE} &= 0.002 \\ C_{TI} &= 0.085 \\ C &= 3.0 \\ We_{cr} &= 6.0 \\ \alpha_{g_max} &= 0.75 \end{aligned}$$

> Currently, this model is only suitable for two-phase flow regimes, one phase being gas and another liquid, i.e. bubbly column applications. However, you can always use UDFs to include your own interfacial area concentration models, which can apply to other flow regimes.

See the separate UDF Manual for details.

16.4.10 Solids Pressure

The total solid pressure is calculated and included in the mixture momentum equations:

$$P_{s,total} = \sum_{q=1}^{N} p_q \tag{16.4-39}$$

where p_q is presented in the section for granular flows by equation Equation 16.5-67

16.5 Eulerian Model Theory

Details about the Eulerian multiphase model are presented in the following subsections:

- Section 16.5.1: Overview and Limitations of the Eulerian Model
- Section 16.5.2: Volume Fraction Equation
- Section 16.5.3: Conservation Equations
- Section 16.5.4: Interphase Exchange Coefficients
- Section 16.5.5: Solids Pressure
- Section 16.5.6: Maximum Packing Limit in Binary Mixtures
- Section 16.5.7: Solids Shear Stresses
- Section 16.5.8: Granular Temperature
- Section 16.5.9: Interfacial Area Concentration
- Section 16.5.10: Description of Heat Transfer
- Section 16.5.11: Turbulence Models
- Section 16.5.12: Solution Method in ANSYS FLUENT
- Section 16.5.13: Dense Discrete Phase Model
- Section 16.5.14: Immiscible Fluid Model

16.5.1 Overview and Limitations of the Eulerian Model

Overview

The Eulerian multiphase model in ANSYS FLUENT allows for the modeling of multiple separate, yet interacting phases. The phases can be liquids, gases, or solids in nearly any combination. An Eulerian treatment is used for each phase, in contrast to the Eulerian-Lagrangian treatment that is used for the discrete phase model.

With the Eulerian multiphase model, the number of secondary phases is limited only by memory requirements and convergence behavior. Any number of secondary phases can be modeled, provided that sufficient memory is available. For complex multiphase flows, however, you may find that your solution is limited by convergence behavior. See Section 24.7.5: Eulerian Model in the separate User's Guide for multiphase modeling strategies.

ANSYS FLUENT's Eulerian multiphase model does not distinguish between fluid-fluid and fluid-solid (granular) multiphase flows. A granular flow is simply one that involves at least one phase that has been designated as a granular phase.

The ANSYS FLUENT solution is based on the following:

- A single pressure is shared by all phases.
- Momentum and continuity equations are solved for each phase.
- The following parameters are available for granular phases:
  - Granular temperature (solids fluctuating energy) can be calculated for each solid phase. You can select either an algebraic formulation, a constant, a user-defined function, or a partial differential equation.
  - Solid-phase shear and bulk viscosities are obtained by applying kinetic theory to granular flows. Frictional viscosity for modeling granular flow is also available. You can select appropriate models and user-defined functions for all properties.
- Several interphase drag coefficient functions are available, which are appropriate for various types of multiphase regimes. (You can also modify the interphase drag coefficient through user-defined functions, as described in the separate UDF Manual.)
- All of the k-ϵ turbulence models are available, and may apply to all phases or to the mixture.

Limitations

All other features available in ANSYS FLUENT can be used in conjunction with the Eulerian multiphase model, except for the following limitations:

- The Reynolds Stress turbulence model is not available on a per phase basis.
- Particle tracking (using the Lagrangian dispersed phase model) interacts only with the primary phase.
- Streamwise periodic flow with specified mass flow rate cannot be modeled when the Eulerian model is used (the user is allowed to specify a pressure drop).
- Inviscid flow is not allowed.
- Melting and solidification are not allowed.
- When tracking particles in parallel, the DPM model cannot be used with the Eulerian multiphase model if the shared memory option is enabled (Section 23.8: Parallel Processing for the Discrete Phase Model in the separate User's Guide). (Note that using the message passing option, when running in parallel, enables the compatibility of all multiphase flow models with the DPM model.)

To change from a single-phase model, where a single set of conservation equations for momentum, continuity and (optionally) energy is solved, to a multiphase model, additional sets of conservation equations must be introduced. In the process of introducing additional sets of conservation equations, the original set must also be modified. The modifications involve, among other things, the introduction of the volume fractions $\alpha_1, \alpha_2, \ldots\ \alpha_n$ for the multiple phases, as well as mechanisms for the exchange of momentum, heat, and mass between the phases.

16.5.2 Volume Fraction Equation

The description of multiphase flow as interpenetrating continua incorporates the concept of phasic volume fractions, denoted here by α_q. Volume fractions represent the space occupied by each phase, and the laws of conservation of mass and momentum are satisfied by each phase individually. The derivation of the conservation equations can be done by ensemble averaging the local instantaneous balance for each of the phases [5] or by using the mixture theory approach [31].

The volume of phase q, V_q, is defined by

$$V_q = \int_V \alpha_q dV \tag{16.5-1}$$

where

$$\sum_{q=1}^{n} \alpha_q = 1 \tag{16.5-2}$$

The effective density of phase q is

$$\hat{\rho}_q = \alpha_q \rho_q \tag{16.5-3}$$

where ρ_q is the physical density of phase q.

The volume fraction equation may be solved either through implicit or explicit time discretization. For detailed information about both VOF schemes, refer to Section 16.3.2: The Implicit Scheme and Section 16.3.2: The Explicit Scheme.

16.5.3 Conservation Equations

The general conservation equations from which the equations solved by ANSYS FLUENT are derived are presented in this section, followed by the solved equations themselves.

Equations in General Form

Conservation of Mass

The continuity equation for phase q is

$$\frac{\partial}{\partial t}(\alpha_q \rho_q) + \nabla \cdot (\alpha_q \rho_q \vec{v}_q) = \sum_{p=1}^{n} (\dot{m}_{\rm pq} - \dot{m}_{\rm qp}) + S_q \tag{16.5-4}$$

where $\vec{v}_q$ is the velocity of phase q and $\dot{m}_{pq}$ characterizes the mass transfer from the $p^{\rm th}$ to $q^{\rm th}$ phase, and $\dot{m}_{\rm qp}$ characterizes the mass transfer from phase q to phase p, and you are able to specify these mechanisms separately.

By default, the source term S_q on the right-hand side of Equation 16.5-4 is zero, but you can specify a constant or user-defined mass source for each phase. A similar term appears in the momentum and enthalpy equations. See Section 16.7: Modeling Mass Transfer in Multiphase Flows for more information on the modeling of mass transfer in ANSYS FLUENT's general multiphase models.

Conservation of Momentum

The momentum balance for phase q yields

$$\frac{\partial}{\partial t}(\alpha_q \rho_q \vec{v}_q) + \nabla \cdot (\alpha_q \rho_q \vec{v}_q \vec{v}_q) = -\alpha_q \nabla p + \nabla \cdot \overline{\overline{\tau}}_q + \alpha_q \rho_q \vec{g} + \sum_{p=1}^{n} (\vec{R}_{pq} + \dot{m}_{pq} \vec{v}_{pq} - \dot{m}_{qp} \vec{v}_{qp}) + (\vec{F}_q + \vec{F}_{\text{lift},q} + \vec{F}_{\text{vm},q}) \tag{16.5-5}$$

where $\overline{\overline{\tau}}_q$ is the q^{th} phase stress-strain tensor

$$\overline{\overline{\tau}}_q = \alpha_q \mu_q (\nabla \vec{v}_q + \nabla \vec{v}_q^{\text{T}}) + \alpha_q (\lambda_q - \frac{2}{3}\mu_q) \nabla \cdot \vec{v}_q \overline{\overline{I}} \tag{16.5-6}$$

Here μ_q and λ_q are the shear and bulk viscosity of phase q, $\vec{F}_q$ is an external body force, $\vec{F}_{\text{lift},q}$ is a lift force, $\vec{F}_{\text{vm},q}$ is a virtual mass force, $\vec{R}_{pq}$ is an interaction force between phases, and p is the pressure shared by all phases.

$\vec{v}_{pq}$ is the interphase velocity, defined as follows. If $\dot{m}_{pq} > 0$ (i.e., phase p mass is being transferred to phase q), $\vec{v}_{pq} = \vec{v}_p$; if $\dot{m}_{pq} < 0$ (i.e., phase q mass is being transferred to phase p), $\vec{v}_{pq} = \vec{v}_q$. Likewise, if $\dot{m}_{qp} > 0$ then $v_{qp} = v_q$, if $\dot{m}_{qp} < 0$ then $v_{qp} = v_p$.

Equation 16.5-5 must be closed with appropriate expressions for the interphase force $\vec{R}_{pq}$. This force depends on the friction, pressure, cohesion, and other effects, and is subject to the conditions that $\vec{R}_{pq} = -\vec{R}_{qp}$ and $\vec{R}_{qq} = 0$.

ANSYS FLUENT uses a simple interaction term of the following form:

$$\sum_{p=1}^{n} \vec{R}_{pq} = \sum_{p=1}^{n} K_{pq} (\vec{v}_p - \vec{v}_q) \tag{16.5-7}$$

where K_{pq} ($= K_{qp}$) is the interphase momentum exchange coefficient (described in Section 16.5.4: Interphase Exchange Coefficients).

Lift Forces

For multiphase flows, ANSYS FLUENT can include the effect of lift forces on the secondary phase particles (or droplets or bubbles). These lift forces act on a particle mainly due to velocity gradients in the primary-phase flow field. The lift force will be more significant for larger particles, but the ANSYS FLUENT model assumes that the particle diameter is much smaller than the interparticle spacing. Thus, the inclusion of lift forces is not appropriate for closely packed particles or for very small particles.

The lift force acting on a secondary phase p in a primary phase q is computed from [80]

$$\vec{F}_{\mathrm{lift}} = -0.5\rho_q\alpha_p(\vec{v}_q - \vec{v}_p) \times (\nabla \times \vec{v}_q) \tag{16.5-8}$$

The lift force $\vec{F}_{\mathrm{lift}}$ will be added to the right-hand side of the momentum equation for both phases ($\vec{F}_{\mathrm{lift},q} = -\vec{F}_{\mathrm{lift},p}$).

In most cases, the lift force is insignificant compared to the drag force, so there is no reason to include this extra term. If the lift force is significant (e.g., if the phases separate quickly), it may be appropriate to include this term. By default, $\vec{F}_{\mathrm{lift}}$ is not included. The lift force and lift coefficient can be specified for each pair of phases, if desired.

i It is important that if you include the lift force in your calculation, you need not include it everywhere in the computational domain since it is computationally expensive to converge. For example, in the wall boundary layer for turbulent bubbly flows in channels, the lift force is significant when the slip velocity is large in the vicinity of high strain rates for the primary phase.

Virtual Mass Force

For multiphase flows, ANSYS FLUENT includes the "virtual mass effect" that occurs when a secondary phase p accelerates relative to the primary phase q. The inertia of the primary-phase mass encountered by the accelerating particles (or droplets or bubbles) exerts a "virtual mass force" on the particles [80]:

$$\vec{F}_{\mathrm{vm}} = 0.5\alpha_p\rho_q\left(\frac{d_q\vec{v}_q}{dt} - \frac{d_p\vec{v}_p}{dt}\right) \tag{16.5-9}$$

The term $\frac{d_q}{dt}$ denotes the phase material time derivative of the form

$$\frac{d_q(\phi)}{dt} = \frac{\partial(\phi)}{\partial t} + (\vec{v}_q \cdot \nabla)\phi \tag{16.5-10}$$

The virtual mass force $\vec{F}_{\mathrm{vm}}$ will be added to the right-hand side of the momentum equation for both phases ($\vec{F}_{\mathrm{vm},q} = -\vec{F}_{\mathrm{vm},p}$).

The virtual mass effect is significant when the secondary phase density is much smaller than the primary phase density (e.g., for a transient bubble column). By default, $\vec{F}_{\mathrm{vm}}$ is not included.

Conservation of Energy

To describe the conservation of energy in Eulerian multiphase applications, a separate enthalpy equation can be written for each phase:

$$\frac{\partial}{\partial t}(\alpha_q \rho_q h_q) + \nabla \cdot (\alpha_q \rho_q \vec{u}_q h_q) = \alpha_q \frac{\partial p_q}{\partial t} + \overline{\overline{\tau}}_q : \nabla \vec{u}_q - \nabla \cdot \vec{q}_q + S_q + \sum_{p=1}^{n} (Q_{pq} + \dot{m}_{pq} h_{pq} - \dot{m}_{qp} h_{qp}) \tag{16.5-11}$$

where h_q is the specific enthalpy of the $q^{\rm th}$ phase, $\vec{q}_q$ is the heat flux, S_q is a source term that includes sources of enthalpy (e.g., due to chemical reaction or radiation), Q_{pq} is the intensity of heat exchange between the $p^{\rm th}$ and $q^{\rm th}$ phases, and h_{pq} is the interphase enthalpy (e.g., the enthalpy of the vapor at the temperature of the droplets, in the case of evaporation). The heat exchange between phases must comply with the local balance conditions $Q_{pq} = -Q_{qp}$ and $Q_{qq} = 0$.

Equations Solved by ANSYS FLUENT

The equations for fluid-fluid and granular multiphase flows, as solved by ANSYS FLUENT, are presented here for the general case of an n-phase flow.

Continuity Equation

The volume fraction of each phase is calculated from a continuity equation:

$$\frac{1}{\rho_{rq}} \left(\frac{\partial}{\partial t}(\alpha_q \rho_q) + \nabla \cdot (\alpha_q \rho_q \vec{v}_q) = \sum_{p=1}^{n} (\dot{m}_{pq} - \dot{m}_{qp}) \right) \tag{16.5-12}$$

where ρ_{rq} is the phase reference density, or the volume averaged density of the $q^{\rm th}$ phase in the solution domain.

The solution of this equation for each secondary phase, along with the condition that the volume fractions sum to one (given by Equation 16.5-2), allows for the calculation of the primary-phase volume fraction. This treatment is common to fluid-fluid and granular flows.

Fluid-Fluid Momentum Equations

The conservation of momentum for a fluid phase q is

$$\begin{aligned}\frac{\partial}{\partial t}(\alpha_q \rho_q \vec{v}_q) + \nabla \cdot (\alpha_q \rho_q \vec{v}_q \vec{v}_q) &= -\alpha_q \nabla p + \nabla \cdot \overline{\overline{\tau}}_q + \alpha_q \rho_q \vec{g} + \\ & \sum_{p=1}^{n} (K_{pq}(\vec{v}_p - \vec{v}_q) + \dot{m}_{pq} \vec{v}_{pq} - \dot{m}_{qp} \vec{v}_{qp}) + \\ & (\vec{F}_q + \vec{F}_{\text{lift},q} + \vec{F}_{\text{vm},q}) \qquad (16.5\text{-}13)\end{aligned}$$

Here $\vec{g}$ is the acceleration due to gravity and $\overline{\overline{\tau}}_q$, $\vec{F}_q$, $\vec{F}_{\text{lift},q}$, and $\vec{F}_{\text{vm},q}$ are as defined for Equation 16.5-5.

Fluid-Solid Momentum Equations

Following the work of [3, 49, 71, 110, 183, 208, 254, 343], ANSYS FLUENT uses a multi-fluid granular model to describe the flow behavior of a fluid-solid mixture. The solid-phase stresses are derived by making an analogy between the random particle motion arising from particle-particle collisions and the thermal motion of molecules in a gas, taking into account the inelasticity of the granular phase. As is the case for a gas, the intensity of the particle velocity fluctuations determines the stresses, viscosity, and pressure of the solid phase. The kinetic energy associated with the particle velocity fluctuations is represented by a "pseudothermal" or granular temperature which is proportional to the mean square of the random motion of particles.

The conservation of momentum for the fluid phases is similar to Equation 16.5-13, and that for the s^{th} solid phase is

$$\begin{aligned}\frac{\partial}{\partial t}(\alpha_s \rho_s \vec{v}_s) + \nabla \cdot (\alpha_s \rho_s \vec{v}_s \vec{v}_s) &= -\alpha_s \nabla p - \nabla p_s + \nabla \cdot \overline{\overline{\tau}}_s + \alpha_s \rho_s \vec{g} + \\ & \sum_{l=1}^{N} (K_{ls}(\vec{v}_l - \vec{v}_s) + \dot{m}_{ls} \vec{v}_{ls} - \dot{m}_{sl} \vec{v}_{sl}) + \\ & (\vec{F}_s + \vec{F}_{\text{lift},s} + \vec{F}_{\text{vm},s}) \qquad (16.5\text{-}14)\end{aligned}$$

where p_s is the s^{th} solids pressure, $K_{ls} = K_{sl}$ is the momentum exchange coefficient between fluid or solid phase l and solid phase s, N is the total number of phases, and $\vec{F}_q$, $\vec{F}_{\text{lift},q}$, and $\vec{F}_{\text{vm},q}$ are as defined for Equation 16.5-5.

Conservation of Energy

The equation solved by ANSYS FLUENT for the conservation of energy is Equation 16.5-11.

16.5.4 Interphase Exchange Coefficients

It can be seen in Equations 16.5-13 and 16.5-14 that momentum exchange between the phases is based on the value of the fluid-fluid exchange coefficient K_{pq} and, for granular flows, the fluid-solid and solid-solid exchange coefficients K_{ls}.

Fluid-Fluid Exchange Coefficient

For fluid-fluid flows, each secondary phase is assumed to form droplets or bubbles. This has an impact on how each of the fluids is assigned to a particular phase. For example, in flows where there are unequal amounts of two fluids, the predominant fluid should be modeled as the primary fluid, since the sparser fluid is more likely to form droplets or bubbles. The exchange coefficient for these types of bubbly, liquid-liquid or gas-liquid mixtures can be written in the following general form:

$$K_{pq} = \frac{\alpha_q \alpha_p \rho_p f}{\tau_p} \tag{16.5-15}$$

where f, the drag function, is defined differently for the different exchange-coefficient models (as described below) and τ_p, the "particulate relaxation time", is defined as

$$\tau_p = \frac{\rho_p d_p^2}{18\mu_q} \tag{16.5-16}$$

where d_p is the diameter of the bubbles or droplets of phase p.

Nearly all definitions of f include a drag coefficient (C_D) that is based on the relative Reynolds number (Re). It is this drag function that differs among the exchange-coefficient models. For all these situations, K_{pq} should tend to zero whenever the primary phase is not present within the domain. To enforce this, the drag function f is always multiplied by the volume fraction of the primary phase q, as is reflected in Equation 16.5-15.

- For the model of Schiller and Naumann [305]

$$f = \frac{C_D \mathrm{Re}}{24} \tag{16.5-17}$$

where

$$C_D = \begin{cases} 24(1 + 0.15\ \mathrm{Re}^{0.687})/\mathrm{Re} & \mathrm{Re} \leq 1000 \\ 0.44 & \mathrm{Re} > 1000 \end{cases} \tag{16.5-18}$$

and Re is the relative Reynolds number. The relative Reynolds number for the primary phase q and secondary phase p is obtained from

$$\mathrm{Re} = \frac{\rho_q |\vec{v}_p - \vec{v}_q| d_p}{\mu_q} \tag{16.5-19}$$

The relative Reynolds number for secondary phases p and r is obtained from

$$\mathrm{Re} = \frac{\rho_{rp} |\vec{v}_r - \vec{v}_p| d_{rp}}{\mu_{rp}} \tag{16.5-20}$$

where $\mu_{rp} = \alpha_p \mu_p + \alpha_r \mu_r$ is the mixture viscosity of the phases p and r.

The Schiller and Naumann model is the default method, and it is acceptable for general use for all fluid-fluid pairs of phases.

- For the Morsi and Alexander model [238]

$$f = \frac{C_D \mathrm{Re}}{24} \tag{16.5-21}$$

where

$$C_D = a_1 + \frac{a_2}{\mathrm{Re}} + \frac{a_3}{\mathrm{Re}^2} \tag{16.5-22}$$

and Re is defined by Equation 16.5-19 or 16.5-20. The a's are defined as follows:

$$a_1,\ a_2,\ a_3 = \begin{cases} 0,\ 24,\ 0 & 0 < \text{Re} < 0.1 \\ 3.690,\ 22.73,\ 0.0903 & 0.1 < \text{Re} < 1 \\ 1.222,\ 29.1667,\ -3.8889 & 1 < \text{Re} < 10 \\ 0.6167,\ 46.50,\ -116.67 & 10 < \text{Re} < 100 \\ 0.3644,\ 98.33,\ -2778 & 100 < \text{Re} < 1000 \\ 0.357,\ 148.62,\ -47500 & 1000 < \text{Re} < 5000 \\ 0.46,\ -490.546,\ 578700 & 5000 < \text{Re} < 10000 \\ 0.5191,\ -1662.5,\ 5416700 & \text{Re} \geq 10000 \end{cases} \tag{16.5-23}$$

The Morsi and Alexander model is the most complete, adjusting the function definition frequently over a large range of Reynolds numbers, but calculations with this model may be less stable than with the other models.

- For the symmetric model

$$K_{pq} = \frac{\alpha_p(\alpha_p\rho_p + \alpha_q\rho_q)f}{\tau_{pq}} \tag{16.5-24}$$

where

$$\tau_{pq} = \frac{(\alpha_p\rho_p + \alpha_q\rho_q)(\frac{d_p+d_q}{2})^2}{18(\alpha_p\mu_p + \alpha_q\mu_q)} \tag{16.5-25}$$

and

$$f = \frac{C_D\text{Re}}{24} \tag{16.5-26}$$

where

$$C_D = \begin{cases} 24(1 + 0.15\ \text{Re}^{0.687})/\text{Re} & \text{Re} \leq 1000 \\ 0.44 & \text{Re} > 1000 \end{cases} \tag{16.5-27}$$

and Re is defined by Equation 16.5-19 or 16.5-20. Note that if there is only one dispersed phase, then $d_p = d_q$ in Equation 16.5-25.

The symmetric model is recommended for flows in which the secondary (dispersed) phase in one region of the domain becomes the primary (continuous) phase in another. Thus for a single dispersed phase, $d_p = d_q$ and $\frac{(d_p+d_q)}{2} = d_p$. For example, if air is injected into the bottom of a container filled halfway with water, the air is the dispersed phase in the bottom half of the container; in the top half of the container, the air is the continuous phase. This model can also be used for the interaction between secondary phases.

You can specify different exchange coefficients for each pair of phases. It is also possible to use user-defined functions to define exchange coefficients for each pair of phases. If the exchange coefficient is equal to zero (i.e., if no exchange coefficient is specified), the flow fields for the fluids will be computed independently, with the only "interaction" being their complementary volume fractions within each computational cell.

Fluid-Solid Exchange Coefficient

The fluid-solid exchange coefficient K_{sl} can be written in the following general form:

$$K_{sl} = \frac{\alpha_s \rho_s f}{\tau_s} \tag{16.5-28}$$

where f is defined differently for the different exchange-coefficient models (as described below), and τ_s, the "particulate relaxation time", is defined as

$$\tau_s = \frac{\rho_s d_s^2}{18\mu_l} \tag{16.5-29}$$

where d_s is the diameter of particles of phase s.

All definitions of f include a drag function (C_D) that is based on the relative Reynolds number (Re_s). It is this drag function that differs among the exchange-coefficient models.

- For the Syamlal-O'Brien model [342]

$$f = \frac{C_D \mathrm{Re}_s \alpha_l}{24 v_{r,s}^2} \tag{16.5-30}$$

where the drag function has a form derived by Dalla Valle [66]

$$C_D = \left(0.63 + \frac{4.8}{\sqrt{\mathrm{Re}_s / v_{r,s}}}\right)^2 \tag{16.5-31}$$

This model is based on measurements of the terminal velocities of particles in fluidized or settling beds, with correlations that are a function of the volume fraction and relative Reynolds number [294]:

$$\mathrm{Re}_s = \frac{\rho_l d_s |\vec{v}_s - \vec{v}_l|}{\mu_l} \tag{16.5-32}$$

where the subscript l is for the l^{th} fluid phase, s is for the s^{th} solid phase, and d_s is the diameter of the s^{th} solid phase particles.

The fluid-solid exchange coefficient has the form

$$K_{sl} = \frac{3\alpha_s\alpha_l\rho_l}{4v_{r,s}^2 d_s} C_D \left(\frac{\mathrm{Re}_s}{v_{r,s}}\right) |\vec{v}_s - \vec{v}_l| \tag{16.5-33}$$

where $v_{r,s}$ is the terminal velocity correlation for the solid phase [105]:

$$v_{r,s} = 0.5\left(A - 0.06\,\mathrm{Re}_s + \sqrt{(0.06\,\mathrm{Re}_s)^2 + 0.12\,\mathrm{Re}_s\,(2B - A) + A^2}\right) \tag{16.5-34}$$

with

$$A = \alpha_l^{4.14} \tag{16.5-35}$$

and

$$B = 0.8\alpha_l^{1.28} \tag{16.5-36}$$

for $\alpha_l \leq 0.85$, and

$$B = \alpha_l^{2.65} \tag{16.5-37}$$

for $\alpha_l > 0.85$.

This model is appropriate when the solids shear stresses are defined according to Syamlal et al. [343] (Equation 16.5-83).

- For the model of Wen and Yu [373], the fluid-solid exchange coefficient is of the following form:

$$K_{sl} = \frac{3}{4} C_D \frac{\alpha_s\alpha_l\rho_l|\vec{v}_s - \vec{v}_l|}{d_s} \alpha_l^{-2.65} \tag{16.5-38}$$

where

$$C_D = \frac{24}{\alpha_l \mathrm{Re}_s}\left[1 + 0.15(\alpha_l \mathrm{Re}_s)^{0.687}\right] \tag{16.5-39}$$

and Re_s is defined by Equation 16.5-32.

This model is appropriate for dilute systems.

- The Gidaspow model [110] is a combination of the Wen and Yu model [373] and the Ergun equation [88].

  When $\alpha_l > 0.8$, the fluid-solid exchange coefficient K_{sl} is of the following form:

  $$K_{sl} = \frac{3}{4} C_D \frac{\alpha_s \alpha_l \rho_l |\vec{v}_s - \vec{v}_l|}{d_s} \alpha_l^{-2.65} \tag{16.5-40}$$

  where

  $$C_D = \frac{24}{\alpha_l \mathrm{Re}_s} \left[1 + 0.15 (\alpha_l \mathrm{Re}_s)^{0.687}\right] \tag{16.5-41}$$

  When $\alpha_l \leq 0.8$,

  $$K_{sl} = 150 \frac{\alpha_s (1 - \alpha_l) \mu_l}{\alpha_l d_s^2} + 1.75 \frac{\rho_l \alpha_s |\vec{v}_s - \vec{v}_l|}{d_s} \tag{16.5-42}$$

  This model is recommended for dense fluidized beds.

Solid-Solid Exchange Coefficient

The solid-solid exchange coefficient K_{ls} has the following form [341]:

$$K_{ls} = \frac{3 \left(1 + e_{ls}\right) \left(\frac{\pi}{2} + C_{\mathrm{fr},ls} \frac{\pi^2}{8}\right) \alpha_s \rho_s \alpha_l \rho_l \left(d_l + d_s\right)^2 g_{0,ls}}{2\pi \left(\rho_l d_l^3 + \rho_s d_s^3\right)} |\vec{v}_l - \vec{v}_s| \tag{16.5-43}$$

where

e_{ls}	=	the coefficient of restitution
$C_{\mathrm{fr},ls}$	=	the coefficient of friction between the l^{th} and s^{th} solid-phase particles ($C_{\mathrm{fr},ls} = 0$)
d_l	=	the diameter of the particles of solid l
$g_{0,ls}$	=	the radial distribution coefficient

Note that the coefficient of restitution is described in Section 16.5.5: Solids Pressure and the radial distribution coefficient is described in Section 16.5.5: Radial Distribution Function.

Universal Drag Laws for Bubble-Liquid and Droplet-Gas Flows

The universal drag laws [170] are suitable for the calculation of the drag coefficients in bubble-liquid or droplet-gas flow regimes. The drag laws can apply to non-spherical particles with the constraint of a pool flow regime, i.e. the hydraulic diameter of the flow domain which is far larger than the averaged size of the particles.

The exchange coefficient for bubbly and droplet flows can be written in the general form

$$K_{pq} = \frac{\alpha_q \alpha_p \rho_p f}{\tau_p} \tag{16.5-44}$$

Where q represents the primary phase and p the particulate phase. The particulate relaxation time τ_p is defined as

$$\tau_p = \frac{\rho_p {d_p}^2}{18\mu_e} \tag{16.5-45}$$

The drag function f is defined as

$$f = \frac{C_D Re}{24} \tag{16.5-46}$$

The relative Reynolds number for the primary phase q and the secondary phase p is obtained based on the relative velocity of the two phases.

$$Re = \frac{\rho_q |\vec{v_q} - \vec{v_p}| d_p}{\mu_e} \tag{16.5-47}$$

Where μ_e is the effective viscosity of the primary phase accounting for the effects of family of particles in the continuum.

The Rayleigh-Taylor instability wavelength is

$$\lambda_{RT} = \left(\frac{\sigma}{g \Delta\rho_{pq}} \right)^{0.5} \tag{16.5-48}$$

Where σ is the surface tension, g the gravity, and $\Delta\rho_{pq}$ the absolute value of the density difference between phases p and q.

The drag coefficient is defined differently for bubbly and droplet flows.

Bubble-Liquid Flow

$$C_{D_{vis}} = \frac{24}{Re}(1 + 0.1Re^{0.75}) \tag{16.5-49}$$

$$C_{D_{dis}} = 2/3(\frac{d_p}{\lambda_{RT}})\left\{\frac{1 + 17.67f^{*6/7}}{18.67f^{*}}\right\}^2 ; f^{*} = (1 - \alpha_p)^{1.5} \tag{16.5-50}$$

$$C_{D_{cap}} = \frac{8}{3}(1 - \alpha_p)^2 \tag{16.5-51}$$

- In the viscous regime, the following condition is satisfied:

$$C_{D_{dis}} < C_{D_{vis}} \tag{16.5-52}$$

  The drag coefficient, C_D, is computed as

$$C_D = C_{D_{vis}} \tag{16.5-53}$$

- In the distorted bubble regime, the following condition is satisfied:

$$C_{D_{vis}} < C_{D_{dis}} < C_{D_{cap}} \tag{16.5-54}$$

  The drag coefficient is calculated as

$$C_D = C_{D_{dis}} \tag{16.5-55}$$

- In the regime of strongly deformed, capped bubbles, the following condition is satisfied:

$$C_{D_{dis}} > C_{D_{cap}} \tag{16.5-56}$$

  The drag coefficient can be written as

$$C_D = C_{D_{cap}} \tag{16.5-57}$$

The effective viscosity for the bubble-liquid mixture is

$$\mu_e = \frac{\mu_q}{1 - \alpha_p} \tag{16.5-58}$$

Droplet-Gas Flow

- When Re < 1, the drag coefficient for the stokes regime is

$$C_D = \frac{24}{Re} \tag{16.5-59}$$

- When $1 \leq$ Re ≤ 1000, the drag coefficient for the viscous regime is

$$C_D = \frac{24}{Re}(1 + 0.1Re^{0.75}) \tag{16.5-60}$$

- For the Newton's regime (Re ≥ 1000), the drag coefficient is

$$C_D = 2/3(\frac{d_p}{\lambda_{RT}})\left\{\frac{1 + 17.67 f^{*6/7}}{18.67 f^*}\right\}^2 ; f^* = (1 - \alpha_p)^3 \tag{16.5-61}$$

The effective viscosity for a bubble-liquid mixture is

$$\mu_e = \frac{\mu_q}{(1 - \alpha_p)^{2.5}} \tag{16.5-62}$$

i The drag model is currently suitable for bubble-liquid and/or droplet-gas flow when the characteristic length of the flow domain is much larger than the averaged size of the particles.

16.5.5 Solids Pressure

For granular flows in the compressible regime (i.e., where the solids volume fraction is less than its maximum allowed value), a solids pressure is calculated independently and used for the pressure gradient term, ∇p_s, in the granular-phase momentum equation. Because a Maxwellian velocity distribution is used for the particles, a granular temperature is introduced into the model, and appears in the expression for the solids pressure and viscosities. The solids pressure is composed of a kinetic term and a second term due to particle collisions:

$$p_s = \alpha_s \rho_s \Theta_s + 2\rho_s(1 + e_{ss})\alpha_s^2 g_{0,ss} \Theta_s \tag{16.5-63}$$

where e_{ss} is the coefficient of restitution for particle collisions, $g_{0,ss}$ is the radial distribution function, and Θ_s is the granular temperature. ANSYS FLUENT uses a default value of 0.9 for e_{ss}, but the value can be adjusted to suit the particle type. The granular temperature Θ_s is proportional to the kinetic energy of the fluctuating particle motion, and will be described later in this section. The function $g_{0,ss}$ (described below in more detail) is a distribution function that governs the transition from the "compressible" condition with $\alpha < \alpha_{s,\max}$, where the spacing between the solid particles can continue to decrease, to the “incompressible” condition with $\alpha = \alpha_{s,\max}$, where no further decrease in the spacing can occur. A value of 0.63 is the default for $\alpha_{s,\max}$, but you can modify it during the problem setup.

Other formulations that are also available in ANSYS FLUENT are [343]

$$p_s = 2\rho_s(1+e_{ss})\alpha_s^2 g_{0,ss}\Theta_s \tag{16.5-64}$$

and [213]

$$p_s = \alpha_s\rho_s\Theta_s[(1+4\alpha_s g_{0,ss}) + \frac{1}{2}[(1+e_{ss})(1-e_{ss}+2\mu_{fric})]] \tag{16.5-65}$$

When more than one solids phase are calculated, the above expression does not take into account the effect of other phases. A derivation of the expressions from the Boltzman equations for a granular mixture are beyond the scope of this manual, however there is a need to provide a better formulation so that some properties may feel the presence of other phases. A known problem is that N solid phases with identical properties should be consistent when the same phases are described by a single solids phase. Equations derived empirically may not satisfy this property and need to be changed accordingly without deviating significantly from the original form. From [109], a general solids pressure formulation in the presence of other phases could be of the form

$$p_q = \alpha_q\rho_q\Theta_q + \sum_{p=1}^{N}\frac{\pi}{3}g_{0,pq}d_{qp}^3 n_q n_p(1+e_{qp})f(m_p, m_q, \Theta_p, \Theta_q) \tag{16.5-66}$$

where $d_{pq} = \frac{d_p+d_q}{2}$ is the average diameter, n_p, n_q are the number of particles, m_p and m_q are the masses of the particles in phases p and q, and f is a function of the masses of the particles and their granular temperatures. For now, we have to simplify this expression so that it depends only on the granular temperature of phase q

$$p_q = \alpha_q\rho_q\Theta_q + \sum_{p=1}^{N} 2\frac{d_{pq}^3}{d_q^3}(1+e_{pq})g_{0,pq}\alpha_q\alpha_p\rho_q\Theta_q \tag{16.5-67}$$

Since all models need to be cast in the general form, it follows that

$$p_q = \alpha_q\rho_q\Theta_q + (\sum_{p=1}^{N}\frac{d_{pq}^3}{d_q^3}p_{c,qp})\rho_q\Theta_q \tag{16.5-68}$$

where $p_{c,qp}$ is the collisional part of the pressure between phases q and p.

The above expression reverts to the one solids phase expression when $N = 1$ and $q = p$ but also has the property of feeling the presence of other phases.

Radial Distribution Function

The radial distribution function, g_0, is a correction factor that modifies the probability of collisions between grains when the solid granular phase becomes dense. This function may also be interpreted as the nondimensional distance between spheres:

$$g_0 = \frac{s + d_p}{s} \tag{16.5-69}$$

where s is the distance between grains. From Equation 16.5-69 it can be observed that for a dilute solid phase $s \rightarrow \infty$, and therefore $g_0 \rightarrow 1$. In the limit when the solid phase compacts, $s \rightarrow 0$ and $g_0 \rightarrow \infty$. The radial distribution function is closely connected to the factor χ of Chapman and Cowling's [49] theory of nonuniform gases. χ is equal to 1 for a rare gas, and increases and tends to infinity when the molecules are so close together that motion is not possible.

In the literature there is no unique formulation for the radial distribution function. ANSYS FLUENT has a number of options:

- For one solids phase, use [254]:

  $$g_0 = \left[1 - \left(\frac{\alpha_s}{\alpha_{s,\max}}\right)^{\frac{1}{3}}\right]^{-1} \tag{16.5-70}$$

  This is an empirical function and does not extend easily to n phases. For two identical phases with the property that $\alpha_q = \alpha_1 + \alpha_2$, the above function is not consistent for the calculation of the partial pressures p_1 and p_2, $p_q = p_1 + p_2$. In order to correct this problem, ANSYS FLUENT uses the following consistent formulation:

  $$g_{0,ll} = \left[1 - \left(\frac{\alpha_s}{\alpha_{s,\max}}\right)^{\frac{1}{3}}\right]^{-1} + \frac{1}{2} d_l \sum_{k=1}^{N} \frac{\alpha_k}{d_k} \tag{16.5-71}$$

  where

  $$\alpha_s = \sum_{k=1}^{N} \alpha_k \tag{16.5-72}$$

  and k are solid phases only.

- The following expression is also available [137]:

  $$g_{0,ll} = \frac{1}{(1 - \frac{\alpha_s}{\alpha_{s,\max}})} + \frac{3}{2} d_l \sum_{k=1}^{N} \frac{\alpha_k}{d_k} \tag{16.5-73}$$

- Also available [213], slightly modified for n solids phases, is the following:

$$g_{0,ll} = \frac{1 + 2.5\alpha_s + 4.59\alpha_s^2 + 4.52\alpha_s^3}{\left(1 - \left(\frac{\alpha_s}{\alpha_{s,\max}}\right)^3\right)^{0.678}} + \frac{1}{2} d_l \sum_{k=1}^{N} \frac{\alpha_k}{d_k} \tag{16.5-74}$$

- The following equation [343] is available:

$$g_{0,kl} = \frac{1}{(1-\alpha_s)} + \frac{3(\sum_{k=1}^{N} \frac{\alpha_k}{d_k})}{(1-\alpha_s)^2 (d_j + d_k)} d_k d_l \tag{16.5-75}$$

When the number of solid phases is greater than 1, Equation 16.5-71, Equation 16.5-73 and Equation 16.5-74 are extended to

$$g_{0,lm} = \frac{d_m g_{0,ll} + d_l g_{0,mm}}{d_m + d_l} \tag{16.5-76}$$

It is interesting to note that Equation 16.5-73 and Equation 16.5-74 compare well with [3] experimental data, while Equation 16.5-75 reverts to the [46] derivation.

16.5.6 Maximum Packing Limit in Binary Mixtures

The packing limit is not a fixed quantity and may change according to the number of particles present within a given volume and the diameter of the particles. Small particles accumulate in between larger particles increasing the packing limit. For a binary mixture ANSYS FLUENT uses the correlations proposed by [90].

For a binary mixture with diameters $d_1 > d_2$, the mixture composition is defined as $X_1 = \frac{\alpha_1}{\alpha_1 + \alpha_2}$

where

$$X_1 <= \frac{\alpha_{1,max}}{(\alpha_{1,max} + (1 - \alpha_{1,max})\alpha_{2,max})} \tag{16.5-77}$$

as this is a condition for application of the maximum packing limit for binary mixtures.

The maximum packing limit for the mixture is given by

$$\begin{aligned} \alpha_{s,max} &= (\alpha_{1,max} - \alpha_{2,max} + [1 - \sqrt{\frac{d2}{d1}}](1 - \alpha_{1,max})\alpha_{2,max}) && (16.5\text{-}78) \\ & *(\alpha_{1,max} + (1 - \alpha_{1,max})\alpha_{2,max}) \frac{X_1}{\alpha_{1,max}} \\ & +\alpha_{2,max} && (16.5\text{-}79) \end{aligned}$$

otherwise, the maximum packing limit for the binary mixture is

$$[1-\sqrt{\frac{d2}{d1}}](\alpha_{1,max}+(1-\alpha_{1,max})\alpha_{2,max})(1-X_1)+\alpha_{1,max} \tag{16.5-80}$$

The packing limit is used for the calculation of the radial distribution function.

16.5.7 Solids Shear Stresses

The solids stress tensor contains shear and bulk viscosities arising from particle momentum exchange due to translation and collision. A frictional component of viscosity can also be included to account for the viscous-plastic transition that occurs when particles of a solid phase reach the maximum solid volume fraction.

The collisional and kinetic parts, and the optional frictional part, are added to give the solids shear viscosity:

$$\mu_s = \mu_{s,\mathrm{col}} + \mu_{s,\mathrm{kin}} + \mu_{s,\mathrm{fr}} \tag{16.5-81}$$

Collisional Viscosity

The collisional part of the shear viscosity is modeled as [110, 343]

$$\mu_{s,\mathrm{col}} = \frac{4}{5}\alpha_s\rho_s d_s g_{0,ss}(1+e_{ss})\left(\frac{\Theta_s}{\pi}\right)^{1/2}\alpha_s \tag{16.5-82}$$

Kinetic Viscosity

ANSYS FLUENT provides two expressions for the kinetic part.

The default expression is from Syamlal et al. [343]:

$$\mu_{s,\mathrm{kin}} = \frac{\alpha_s d_s \rho_s\sqrt{\Theta_s\pi}}{6\left(3-e_{ss}\right)}\left[1+\frac{2}{5}\left(1+e_{ss}\right)\left(3e_{ss}-1\right)\alpha_s g_{0,ss}\right] \tag{16.5-83}$$

The following optional expression from Gidaspow et al. [110] is also available:

$$\mu_{s,\mathrm{kin}} = \frac{10\rho_s d_s\sqrt{\Theta_s\pi}}{96\alpha_s\left(1+e_{ss}\right)g_{0,ss}}\left[1+\frac{4}{5}g_{0,ss}\alpha_s\left(1+e_{ss}\right)\right]^2\alpha_s \tag{16.5-84}$$

Bulk Viscosity

The solids bulk viscosity accounts for the resistance of the granular particles to compression and expansion. It has the following form from Lun et al. [208]:

$$\lambda_s = \frac{4}{3}\alpha_s \rho_s d_s g_{0,ss}(1+e_{ss})\left(\frac{\Theta_s}{\pi}\right)^{1/2} \tag{16.5-85}$$

Note that the bulk viscosity is set to a constant value of zero, by default. It is also possible to select the Lun et al. expression or use a user-defined function.

Frictional Viscosity

In dense flow at low shear, where the secondary volume fraction for a solid phase nears the packing limit, the generation of stress is mainly due to friction between particles. The solids shear viscosity computed by ANSYS FLUENT does not, by default, account for the friction between particles.

If the frictional viscosity is included in the calculation, ANSYS FLUENT uses Schaeffer's [303] expression:

$$\mu_{s,\mathrm{fr}} = \frac{p_s \sin\phi}{2\sqrt{I_{2D}}} \tag{16.5-86}$$

where p_s is the solids pressure, ϕ is the angle of internal friction, and I_{2D} is the second invariant of the deviatoric stress tensor. It is also possible to specify a constant or user-defined frictional viscosity.

In granular flows with high solids volume fraction, instantaneous collisions are less important. The application of kinetic theory to granular flows is no longer relevant since particles are in contact and the resulting frictional stresses need to be taken into account. ANSYS FLUENT extends the formulation of the frictional viscosity and employs other models, as well as providing new hooks for UDFs. See the separate UDF Manual for details.

The frictional stresses are usually written in Newtonian form:

$$\tau_{friction} = -P_{friction}\vec{\vec{I}} + \mu_{friction}(\nabla\vec{u}_s + (\nabla\vec{u}_s)^T) \tag{16.5-87}$$

The frictional stress is added to the stress predicted by the kinetic theory when the solids volume fraction exceeds a critical value. This value is normally set to 0.5 when the flow is three-dimensional and the maximum packing limit is about 0.63. Then

$$P_S = P_{kinetic} + P_{friction} \tag{16.5-88}$$

$$\mu_S = \mu_{kinetic} + \mu_{friction} \tag{16.5-89}$$

The derivation of the frictional pressure is mainly semi-empirical, while the frictional viscosity can be derived from the first principles. The application of the modified Coulomb law leads to an expression of the form

$$\mu_{friction} = \frac{P_{friction}\sin\phi}{2\sqrt{I_{2D}}} \tag{16.5-90}$$

Where ϕ is the angle of internal friction and I_{2D} is the second invariant of the deviatoric stress tensor.

Two additional models are available in ANSYS FLUENT: the Johnson and Jackson [151] model for frictional pressure and Syamlal et al [343].

The Johnson and Jackson [151] model for frictional pressure is defined as

$$P_{friction} = Fr\frac{(\alpha_s - \alpha_{s,min})^n}{(\alpha_{s,max} - \alpha_s)^p} \tag{16.5-91}$$

With coefficient Fr = 0.05, n=2 and p = 5 [253]. The critical value for the solids volume fraction is 0.5. The coefficient Fr was modified to make it a function of the volume fraction:

$$Fr = 0.1\alpha_s \tag{16.5-92}$$

The frictional viscosity for this model is of the form

$$\mu_{friction} = P_{friction}\sin\phi \tag{16.5-93}$$

The second model that is employed is Syamlal et al. [343], described in Equation 16.5-83. Comparing the two models results in the frictional normal stress differing by orders of magnitude.

The radial distribution function is an important parameter in the description of the solids pressure resulting from granular kinetic theory. If we use the models of Lun et al. [208] or Gidaspow [109] the radial function tends to infinity as the volume fraction tends to the packing limit. It would then be possible to use this pressure directly in the calculation of the frictional viscosity, as it has the desired effect. This approach is also available in ANSYS FLUENT by default.

i The introduction of the frictional viscosity helps in the description of frictional flows, however a complete description would require the introduction of more physics to capture the elastic regime with the calculation of the yield stress and the use of the flow-rule. These effects can be added by the user via UDFs to model static regime. Small time steps are required to get good convergence behavior.

16.5.8 Granular Temperature

The granular temperature for the s^{th} solids phase is proportional to the kinetic energy of the random motion of the particles. The transport equation derived from kinetic theory takes the form [71]

$$\frac{3}{2}\left[\frac{\partial}{\partial t}(\rho_s\alpha_s\Theta_s)+\nabla\cdot(\rho_s\alpha_s\vec{v}_s\Theta_s)\right] = (-p_s\overline{\overline{I}}+\overline{\overline{\tau}}_s):\nabla\vec{v}_s+\nabla\cdot(k_{\Theta_s}\nabla\Theta_s)-\gamma_{\Theta_s}+\phi_{ls} \tag{16.5-94}$$

where

$(-p_s\overline{\overline{I}}+\overline{\overline{\tau}}_s):\nabla\vec{v}_s$ = the generation of energy by the solid stress tensor

$k_{\Theta_s}\nabla\Theta_s$ = the diffusion of energy ($k_{\Theta s}$ is the diffusion coefficient)

γ_{Θ_s} = the collisional dissipation of energy

ϕ_{ls} = the energy exchange between the l^{th} fluid or solid phase and the s^{th} solid phase

Equation 16.5-94 contains the term $k_{\Theta_s}\nabla\Theta_s$ describing the diffusive flux of granular energy. When the default Syamlal et al. model [343] is used, the diffusion coefficient for granular energy, k_{Θ_s} is given by

$$k_{\Theta_s} = \frac{15d_s\rho_s\alpha_s\sqrt{\Theta_s\pi}}{4(41-33\eta)}\left[1+\frac{12}{5}\eta^2(4\eta-3)\alpha_s g_{0,ss}+\frac{16}{15\pi}(41-33\eta)\eta\alpha_s g_{0,ss}\right] \tag{16.5-95}$$

where

$$\eta = \frac{1}{2}(1+e_{ss})$$

ANSYS FLUENT uses the following expression if the optional model of Gidaspow et al. [110] is enabled:

$$k_{\Theta_s} = \frac{150\rho_s d_s\sqrt{(\Theta\pi)}}{384(1+e_{ss})g_{0,ss}}\left[1+\frac{6}{5}\alpha_s g_{0,ss}(1+e_s)\right]^2 + 2\rho_s{\alpha_s}^2 d_s(1+e_{ss})g_{0,ss}\sqrt{\frac{\Theta_s}{\pi}} \qquad (16.5\text{-}96)$$

The collisional dissipation of energy, γ_{Θ_s}, represents the rate of energy dissipation within the s^{th} solids phase due to collisions between particles. This term is represented by the expression derived by Lun et al. [208]

$$\gamma_{\Theta m} = \frac{12(1-e_{ss}^2)g_{0,ss}}{d_s\sqrt{\pi}}\rho_s\alpha_s^2\Theta_s^{3/2} \qquad (16.5\text{-}97)$$

The transfer of the kinetic energy of random fluctuations in particle velocity from the s^{th} solids phase to the l^{th} fluid or solid phase is represented by ϕ_{ls} [110]:

$$\phi_{ls} = -3K_{ls}\Theta_s \qquad (16.5\text{-}98)$$

ANSYS FLUENT allows the user to solve for the granular temperature with the following options:

- algebraic formulation (the default)

  It is obtained by neglecting convection and diffusion in the transport equation, Equation 16.5-94 [343].

- partial differential equation

  This is given by Equation 16.5-94 and it is allowed to choose different options for it properties.

- constant granular temperature

  This is useful in very dense situations where the random fluctuations are small.

- UDF for granular temperature

For a granular phase s, we may write the shear force at the wall in the following form:

$$\vec{\tau_s} = -\frac{\pi}{6}\sqrt{3}\phi\frac{\alpha_s}{\alpha_{s,max}}\rho_s g_0\sqrt{\Theta_s}\vec{U_{s,||}} \qquad (16.5\text{-}99)$$

Here $\vec{U_{s,||}}$ is the particle slip velocity parallel to the wall, ϕ is the specularity coefficient between the particle and the wall, $\alpha_{s,max}$ is the volume fraction for the particles at maximum packing, and g_0 is the radial distribution function that is model dependent.

The general boundary condition for granular temperature at the wall takes the form [151]

$$q_s = \frac{\pi}{6}\sqrt{3}\phi\frac{\alpha_s}{\alpha_{s,max}}\rho_s g_0 \sqrt{\Theta_s})\vec{U}_{s,||} \cdot \vec{U}_{s,||} - \frac{\pi}{4}\sqrt{3}\frac{\alpha_s}{\alpha_{s,max}}(1 - e_{sw}^2)\rho_s g_0 \Theta_s^{\frac{3}{2}} \qquad (16.5\text{-}100)$$

16.5.9 Interfacial Area Concentration

Interfacial area concentration is defined as the interfacial area between two phases per unit mixture volume. This is an important parameter for predicting mass, momentum and energy transfers through the interface between the phases. The models that are inplemented in ANSYS FLUENT are discussed in Section 16.4.9: Interfacial Area Concentration.

16.5.10 Description of Heat Transfer

The internal energy balance for phase q is written in terms of the phase enthalpy, Equation 16.5-11, defined by

$$H_q = \int c_{p,q} dT_q \qquad (16.5\text{-}101)$$

where $c_{p,q}$ is the specific heat at constant pressure of phase q. The thermal boundary conditions used with multiphase flows are the same as those for a single-phase flow. See Chapter 7: Cell Zone and Boundary Conditions in the separate User's Guide for details.

The Heat Exchange Coefficient

The rate of energy transfer between phases is assumed to be a function of the temperature difference

$$Q_{pq} = h_{pq}(T_p - T_q) \qquad (16.5\text{-}102)$$

where $h_{pq}(= h_{qp})$ is the heat transfer coefficient between the p^{th} phase and the q^{th} phase. The heat transfer coefficient is related to the p^{th} phase Nusselt number, Nu_p, by

$$h_{pq} = \frac{6\kappa_q \alpha_p \alpha_q \text{Nu}_p}{{d_p}^2} \qquad (16.5\text{-}103)$$

Here κ_q is the thermal conductivity of the q^{th} phase. The Nusselt number is typically determined from one of the many correlations reported in the literature. In the case of fluid-fluid multiphase, ANSYS FLUENT uses the correlation of Ranz and Marshall [284, 285]:

$$\mathrm{Nu}_p = 2.0 + 0.6\mathrm{Re}_p^{1/2}\mathrm{Pr}^{1/3} \tag{16.5-104}$$

where Re_p is the relative Reynolds number based on the diameter of the p^{th} phase and the relative velocity $|\vec{u_p} - \vec{u_q}|$, and Pr is the Prandtl number of the q^{th} phase:

$$\mathrm{Pr} = \frac{c_{p_q}\mu_q}{\kappa_q} \tag{16.5-105}$$

In the case of granular flows (where $p = s$), ANSYS FLUENT uses a Nusselt number correlation by Gunn [117], applicable to a porosity range of 0.35–1.0 and a Reynolds number of up to 10^5:

$$\mathrm{Nu}_s = (7 - 10\alpha_f + 5\alpha_f^2)(1 + 0.7\mathrm{Re}_s^{0.2}\mathrm{Pr}^{1/3}) + (1.33 - 2.4\alpha_f + 1.2\alpha_f^2)\mathrm{Re}_s^{0.7}\mathrm{Pr}^{1/3} \tag{16.5-106}$$

The Prandtl number is defined as above with $q = f$. For all these situations, h_{pq} should tend to zero whenever one of the phases is not present within the domain. To enforce this, h_{pq} is always multiplied by the volume fraction of the primary phase q, as reflected in Equation 16.5-103.

16.5.11 Turbulence Models

To describe the effects of turbulent fluctuations of velocities and scalar quantities in a single phase, ANSYS FLUENT uses various types of closure models, as described in Chapter 4: Turbulence. In comparison to single-phase flows, the number of terms to be modeled in the momentum equations in multiphase flows is large, and this makes the modeling of turbulence in multiphase simulations extremely complex.

ANSYS FLUENT provides three methods for modeling turbulence in multiphase flows within the context of the k-ϵ models. In addition, ANSYS FLUENT provides two turbulence options within the context of the Reynolds stress models (RSM).

The k-ϵ turbulence model options are:

- mixture turbulence model (the default)
- dispersed turbulence model
- turbulence model for each phase

Note that the descriptions of each method below are presented based on the standard k-ϵ model. The multiphase modifications to the RNG and realizable k-ϵ models are similar, and are therefore not presented explicitly.

The RSM turbulence model options are:

- mixture turbulence model (the default)
- dispersed turbulence model

For either category, the choice of model depends on the importance of the secondary-phase turbulence in your application.

k-ϵ Turbulence Models

ANSYS FLUENT provides three turbulence model options in the context of the k-ϵ models: the mixture turbulence model (the default), the dispersed turbulence model, or a per-phase turbulence model.

k-ϵ Mixture Turbulence Model

The mixture turbulence model is the default multiphase turbulence model. It represents the first extension of the single-phase k-ϵ model, and it is applicable when phases separate, for stratified (or nearly stratified) multiphase flows, and when the density ratio between phases is close to 1. In these cases, using mixture properties and mixture velocities is sufficient to capture important features of the turbulent flow.

The k and ϵ equations describing this model are as follows:

$$\frac{\partial}{\partial t}(\rho_m k) + \nabla \cdot (\rho_m \vec{v}_m k) = \nabla \cdot \left(\frac{\mu_{t,m}}{\sigma_k} \nabla k \right) + G_{k,m} - \rho_m \epsilon \tag{16.5-107}$$

and

$$\frac{\partial}{\partial t}(\rho_m \epsilon) + \nabla \cdot (\rho_m \vec{v}_m \epsilon) = \nabla \cdot \left(\frac{\mu_{t,m}}{\sigma_\epsilon} \nabla \epsilon \right) + \frac{\epsilon}{k}(C_{1\epsilon} G_{k,m} - C_{2\epsilon} \rho_m \epsilon) \tag{16.5-108}$$

where the mixture density and velocity, ρ_m and $\vec{v}_m$, are computed from

$$\rho_m = \sum_{i=1}^{N} \alpha_i \rho_i \tag{16.5-109}$$

and

$$\vec{v}_m = \frac{\sum_{i=1}^{N} \alpha_i \rho_i \vec{v}_i}{\sum_{i=1}^{N} \alpha_i \rho_i} \tag{16.5-110}$$

the turbulent viscosity, $\mu_{t,m}$, is computed from

$$\mu_{t,m} = \rho_m C_\mu \frac{k^2}{\epsilon} \tag{16.5-111}$$

and the production of turbulence kinetic energy, $G_{k,m}$, is computed from

$$G_{k,m} = \mu_{t,m}(\nabla \vec{v}_m + (\nabla \vec{v}_m)^T) : \nabla \vec{v}_m \tag{16.5-112}$$

The constants in these equations are the same as those described in Section 4.4.1: Standard k-ϵ Model for the single-phase k-ϵ model.

k-ϵ Dispersed Turbulence Model

The dispersed turbulence model is the appropriate model when the concentrations of the secondary phases are dilute. In this case, interparticle collisions are negligible and the dominant process in the random motion of the secondary phases is the influence of the primary-phase turbulence. Fluctuating quantities of the secondary phases can therefore be given in terms of the mean characteristics of the primary phase and the ratio of the particle relaxation time and eddy-particle interaction time.

The model is applicable when there is clearly one primary continuous phase and the rest are dispersed dilute secondary phases.

Assumptions

The dispersed method for modeling turbulence in ANSYS FLUENT assumes the following:

- a modified k-ϵ model for the continuous phase

  Turbulent predictions for the continuous phase are obtained using the standard k-ϵ model supplemented with extra terms that include the interphase turbulent momentum transfer.

- Tchen-theory correlations for the dispersed phases

  Predictions for turbulence quantities for the dispersed phases are obtained using the Tchen theory of dispersion of discrete particles by homogeneous turbulence [130].

- interphase turbulent momentum transfer

  In turbulent multiphase flows, the momentum exchange terms contain the correlation between the instantaneous distribution of the dispersed phases and the turbulent fluid motion. It is possible to take into account the dispersion of the dispersed phases transported by the turbulent fluid motion.

- a phase-weighted averaging process

  The choice of averaging process has an impact on the modeling of dispersion in turbulent multiphase flows. A two-step averaging process leads to the appearance of fluctuations in the phase volume fractions. When the two-step averaging process is used with a phase-weighted average for the turbulence, however, turbulent fluctuations in the volume fractions do not appear. ANSYS FLUENT uses phase-weighted averaging, so no volume fraction fluctuations are introduced into the continuity equations.

Turbulence in the Continuous Phase

The eddy viscosity model is used to calculate averaged fluctuating quantities. The Reynolds stress tensor for continuous phase q takes the following form:

$$\overline{\overline{\tau}}''_q = -\frac{2}{3}(\rho_q k_q + \rho_q \nu_{t,q} \nabla \cdot \vec{U}_q)\overline{\overline{I}} + \rho_q \nu_{t,q}(\nabla \vec{U}_q + \nabla \vec{U}_q^{\,T}) \tag{16.5-113}$$

where $\vec{U}_q$ is the phase-weighted velocity.

The turbulent viscosity $\mu_{t,q}$ is written in terms of the turbulent kinetic energy of phase q:

$$\mu_{t,q} = \rho_q C_\mu \frac{k_q^2}{\epsilon_q} \tag{16.5-114}$$

and a characteristic time of the energetic turbulent eddies is defined as

$$\tau_{t,q} = \frac{3}{2} C_\mu \frac{k_q}{\epsilon_q} \tag{16.5-115}$$

where ϵ_q is the dissipation rate and $C_\mu = 0.09$.

The length scale of the turbulent eddies is

$$L_{t,q} = \sqrt{\frac{3}{2}} C_\mu \frac{k_q^{3/2}}{\epsilon_q} \tag{16.5-116}$$

Turbulent predictions are obtained from the modified k-ϵ model:

$$\frac{\partial}{\partial t}(\alpha_q\rho_q k_q) + \nabla\cdot(\alpha_q\rho_q\vec{U}_q k_q) = \nabla\cdot(\alpha_q\frac{\mu_{t,q}}{\sigma_k}\nabla k_q) + \alpha_q G_{k,q} - \alpha_q\rho_q\epsilon_q + \alpha_q\rho_q\Pi_{k_q} \quad (16.5\text{-}117)$$

and

$$\frac{\partial}{\partial t}(\alpha_q\rho_q\epsilon_q) + \nabla\cdot(\alpha_q\rho_q\vec{U}_q\epsilon_q) = \nabla\cdot(\alpha_q\frac{\mu_{t,q}}{\sigma_\epsilon}\nabla\epsilon_q) + \alpha_q\frac{\epsilon_q}{k_q}(C_{1\epsilon}G_{k,q} - C_{2\epsilon}\rho_q\epsilon_q) + \alpha_q\rho_q\Pi_{\epsilon_q} \quad (16.5\text{-}118)$$

Here Π_{k_q} and Π_{ϵ_q} represent the influence of the dispersed phases on the continuous phase q, and $G_{k,q}$ is the production of turbulent kinetic energy, as defined in Section 4.4.4: Modeling Turbulent Production in the k-ϵ Models. All other terms have the same meaning as in the single-phase k-ϵ model.

The term Π_{k_q} can be derived from the instantaneous equation of the continuous phase and takes the following form, where M represents the number of secondary phases:

$$\Pi_{k_q} = \sum_{p=1}^{M}\frac{K_{pq}}{\alpha_q\rho_q}(< \vec{v}''_q\cdot\vec{v}''_p > +(\vec{U}_p - \vec{U}_q)\cdot\vec{v}_{\rm dr}) \quad (16.5\text{-}119)$$

which can be simplified to

$$\Pi_{k_q} = \sum_{p=1}^{M}\frac{K_{pq}}{\alpha_q\rho_q}(k_{pq} - 2k_q + \vec{v}_{pq}\cdot\vec{v}_{\rm dr}) \quad (16.5\text{-}120)$$

where k_{lq} is the covariance of the velocities of the continuous phase q and the dispersed phase l (calculated from Equation 16.5-128 below), $\vec{v}_{pq}$ is the relative velocity, and $\vec{v}_{\rm dr}$ is the drift velocity (defined by Equation 16.5-133 below).

Π_{ϵ_q} is modeled according to Elgobashi et al. [87]:

$$\Pi_{\epsilon_q} = C_{3\epsilon}\frac{\epsilon_q}{k_q}\Pi_{k_q} \quad (16.5\text{-}121)$$

where $C_{3\epsilon} = 1.2$.

Turbulence in the Dispersed Phase

Time and length scales that characterize the motion are used to evaluate dispersion coefficients, correlation functions, and the turbulent kinetic energy of each dispersed phase.

The characteristic particle relaxation time connected with inertial effects acting on a dispersed phase p is defined as

$$\tau_{F,pq} = \alpha_p \rho_q K_{pq}^{-1} \left(\frac{\rho_p}{\rho_q} + C_V \right) \tag{16.5-122}$$

The Lagrangian integral time scale calculated along particle trajectories, mainly affected by the crossing-trajectory effect [63], is defined as

$$\tau_{t,pq} = \frac{\tau_{t,q}}{\sqrt{(1 + C_\beta \xi^2)}} \tag{16.5-123}$$

where

$$\xi = \frac{|\vec{v}_{pq}| \tau_{t,q}}{L_{t,q}} \tag{16.5-124}$$

and

$$C_\beta = 1.8 - 1.35 \cos^2 \theta \tag{16.5-125}$$

where θ is the angle between the mean particle velocity and the mean relative velocity. The ratio between these two characteristic times is written as

$$\eta_{pq} = \frac{\tau_{t,pq}}{\tau_{F,pq}} \tag{16.5-126}$$

Following Simonin [317], ANSYS FLUENT writes the turbulence quantities for dispersed phase p as follows:

$$\begin{aligned}
k_p &= k_q \left(\frac{b^2 + \eta_{pq}}{1 + \eta_{pq}} \right) && (16.5\text{-}127)\\
k_{pq} &= 2k_q \left(\frac{b + \eta_{pq}}{1 + \eta_{pq}} \right) && (16.5\text{-}128)\\
D_{t,pq} &= \frac{1}{3} k_{pq} \tau_{t,pq} && (16.5\text{-}129)\\
D_p &= D_{t,pq} + \left(\frac{2}{3} k_p - b \frac{1}{3} k_{pq} \right) \tau_{F,pq} && (16.5\text{-}130)\\
b &= (1 + C_V) \left(\frac{\rho_p}{\rho_q} + C_V \right)^{-1} && (16.5\text{-}131)
\end{aligned}$$

and $C_V = 0.5$ is the added-mass coefficient.

Interphase Turbulent Momentum Transfer

The turbulent drag term for multiphase flows ($K_{pq}(\vec{v}_p - \vec{v}_q)$ in Equation 16.5-7) is modeled as follows, for dispersed phase p and continuous phase q:

$$K_{pq}(\vec{v}_p - \vec{v}_q) = K_{pq}(\vec{U}_p - \vec{U}_q) - K_{pq}\vec{v}_{\rm dr} \qquad (16.5\text{-}132)$$

The second term on the right-hand side of Equation 16.5-132 contains the drift velocity:

$$\vec{v}_{\rm dr} = -\left(\frac{D_p}{\sigma_{pq}\alpha_p}\nabla\alpha_p - \frac{D_q}{\sigma_{pq}\alpha_q}\nabla\alpha_q\right) \qquad (16.5\text{-}133)$$

Here D_p and D_q are diffusivities, and σ_{pq} is a dispersion Prandtl number. When using Tchen theory in multiphase flows, ANSYS FLUENT assumes $D_p = D_q = D_{t,pq}$ and the default value for σ_{pq} is 0.75.

The drift velocity results from turbulent fluctuations in the volume fraction. When multiplied by the exchange coefficient K_{pq}, it serves as a correction to the momentum exchange term for turbulent flows. This correction is not included, by default, but you can enable it during the problem setup, as discussed in Section 24.5.4: Modeling Turbulence in the separate User's Guide.

k-ϵ Turbulence Model for Each Phase

The most general multiphase turbulence model solves a set of k and ϵ transport equations for each phase. This turbulence model is the appropriate choice when the turbulence transfer among the phases plays a dominant role.

Note that, since ANSYS FLUENT is solving two additional transport equations for each secondary phase, the per-phase turbulence model is more computationally intensive than the dispersed turbulence model.

Transport Equations

The Reynolds stress tensor and turbulent viscosity are computed using Equations 16.5-113 and 16.5-114. Turbulence predictions are obtained from

$$\frac{\partial}{\partial t}(\alpha_q\rho_q k_q) + \nabla\cdot(\alpha_q\rho_q\vec{U}_q k_q) = \nabla\cdot(\alpha_q(\mu_q + \frac{\mu_{t,q}}{\sigma_k})\nabla k_q) + (\alpha_q G_{k,q} - \alpha_q\rho_q\epsilon_q) +$$
$$\sum_{l=1}^{N} K_{lq}(C_{lq}k_l - C_{ql}k_q) - \sum_{l=1}^{N} K_{lq}(\vec{U}_l - \vec{U}_q)\cdot\frac{\mu_{t,l}}{\alpha_l\sigma_l}\nabla\alpha_l + \sum_{l=1}^{N} K_{lq}(\vec{U}_l - \vec{U}_q)\cdot\frac{\mu_{t,q}}{\alpha_q\sigma_q}\nabla\alpha_q \quad (16.5\text{-}134)$$

and

$$\frac{\partial}{\partial t}(\alpha_q \rho_q \epsilon_q) + \nabla \cdot (\alpha_q \rho_q \vec{U}_q \epsilon_q) = \nabla \cdot (\alpha_q \frac{\mu_{t,q}}{\sigma_\epsilon} \nabla \epsilon_q) + \frac{\epsilon_q}{k_q} \Bigg[C_{1\epsilon} \alpha_q G_{k,q} - C_{2\epsilon} \alpha_q \rho_q \epsilon_q + C_{3\epsilon} \left(\sum_{l=1}^{N} K_{lq}(C_{lq} k_l - C_{ql} k_q) - \sum_{l=1}^{N} K_{lq}(\vec{U}_l - \vec{U}_q) \cdot \frac{\mu_{t,l}}{\alpha_l \sigma_l} \nabla \alpha_l + \sum_{l=1}^{N} K_{lq}(\vec{U}_l - \vec{U}_q) \cdot \frac{\mu_{t,q}}{\alpha_q \sigma_q} \nabla \alpha_q \right) \Bigg] \tag{16.5-135}$$

The terms C_{lq} and C_{ql} can be approximated as

$$C_{lq} = 2, \;\; C_{ql} = 2 \left(\frac{\eta_{lq}}{1 + \eta_{lq}} \right) \tag{16.5-136}$$

where η_{lq} is defined by Equation 16.5-126.

Interphase Turbulent Momentum Transfer

The turbulent drag term $(K_{pq}(\vec{v}_p - \vec{v}_q)$ in Equation 16.5-7) is modeled as follows, where l is the dispersed phase (replacing p in Equation 16.5-7) and q is the continuous phase:

$$\sum_{l=1}^{N} K_{lq}(\vec{v}_l - \vec{v}_q) = \sum_{l=1}^{N} K_{lq}(\vec{U}_l - \vec{U}_q) - \sum_{l=1}^{N} K_{lq} \vec{v}_{\mathrm{dr},lq} \tag{16.5-137}$$

Here $\vec{U}_l$ and $\vec{U}_q$ are phase-weighted velocities, and $\vec{v}_{\mathrm{dr},lq}$ is the drift velocity for phase l (computed using Equation 16.5-133, substituting l for p). Note that ANSYS FLUENT will compute the diffusivities D_l and D_q directly from the transport equations, rather than using Tchen theory (as it does for the dispersed turbulence model).

As noted above, the drift velocity results from turbulent fluctuations in the volume fraction. When multiplied by the exchange coefficient K_{lq}, it serves as a correction to the momentum exchange term for turbulent flows. This correction is not included, by default, but you can enable it during the problem setup.

The turbulence model for each phase in ANSYS FLUENT accounts for the effect of the turbulence field of one phase on the other(s). If you want to modify or enhance the interaction of the multiple turbulence fields and interphase turbulent momentum transfer, you can supply these terms using user-defined functions.

RSM Turbulence Models

Multiphase turbulence modeling typically involves two equation models that are based on single-phase models and often cannot accurately capture the underlying flow physics. Additional turbulence modeling for multiphase flows is diminished even more when the basic underlying single-phase model cannot capture the complex physics of the flow. In such situations, the logical next step is to combine the Reynolds stress model with the multiphase algorithm in order to handle challenging situations in which both factors, RSM for turbulence and the Eulerian multiphase formulation, are a precondition for accurate predictions [59].

The phase-averaged continuity and momentum equations for a continuous phase are:

$$\frac{\partial}{\partial t}(\overline{\alpha_{\rm c}}\rho_{\rm c}) + \nabla\cdot(\overline{\alpha_{\rm c}}\rho_{\rm c}\tilde{U}_{\rm c}) = 0 \tag{16.5-138}$$

$$\frac{\partial}{\partial t}(\overline{\alpha_{\rm c}}\rho_{rmc}\tilde{U}_{\rm c}) + \nabla\cdot(\overline{\alpha_{\rm c}}\rho_{rmc}\tilde{U}_{\rm c}\bigotimes\tilde{U}_{\rm c}) = -\overline{\alpha_{\rm c}}\nabla\tilde{p} + \nabla\cdot\tilde{\tau}^{t}_{\rm c} + F_{\rm Dc} \tag{16.5-139}$$

For simplicity, the laminar stress-strain tensor and other body forces such as gravity have been omitted from Equations 16.5-138-16.5-139. The tilde denotes phase-averaged variables while an overbar (e.g., $\overline{\alpha_{\rm c}}$) reflects time-averaged values. In general, any variable Φ can have a phase-average value defined as

$$\tilde{\Phi}_{\rm c} = \frac{\overline{\alpha_{\rm c}\Phi_{\rm c}}}{\overline{\alpha_{\rm c}}} \tag{16.5-140}$$

Considering only two phases for simplicity, the drag force between the continuous and the dispersed phases can be defined as:

$$F_{\rm Dc} = K_{\rm dc}\left[(\tilde{U}_{\rm d} - \tilde{U}_{\rm c}) - \left(\frac{\overline{\alpha_{\rm d}u'_{\rm d}}}{\overline{\alpha_{\rm d}}} - \frac{\overline{\alpha_{\rm c}u'_{\rm c}}}{\overline{\alpha_{\rm c}}}\right)\right] \tag{16.5-141}$$

where $K_{\rm dc}$ is the drag coefficient. Several terms in the Equation 16.5-141 need to be modeled in order to close the phase-averaged momentum equations. Full descriptions of all modeling assumptions can be found in [58]. This section only describes the different modeling definition of the turbulent stresses $\tilde{\tau}^t$ that appears in Equation 16.5-139.

The turbulent stress that appears in the momentum equations need to be defined on a per-phase basis and can be calculated as:

$$\tau^{\tilde{t}}{}_{\rm k} = -\overline{\alpha_{\rm k}}\rho_{\rm k}\tilde{R}_{\rm k,ij} \tag{16.5-142}$$

where the subscript k is replaced by c for the primary (i.e., continuous) phase or by d for any secondary (i.e., dispersed) phases. As is the case for single-phase flows, the current multiphase Reynolds stress model (RSM) also solves the transport equations for Reynolds stresses $R_{\rm ij}$. ANSYS FLUENT includes two methods for modeling turbulence in multiphase flows within the context of the RSM model: the dispersed turbulence model, and the mixture turbulence model.

RSM Dispersed Turbulence Model

The dispersed turbulence model is used when the concentrations of the secondary phase are dilute and the primary phase turbulence is regarded as the dominant process. Consequently, the transport equations for turbulence quantities are only solved for the primary (continuous) phase, while the predictions of turbulence quantities for dispersed phases are obtained using the Tchen theory. The transport equation for the primary phase Reynolds stresses in the case of the dispersed model are:

$$\begin{aligned}
\frac{\partial}{\partial t}(\overline{\alpha}\rho\tilde{R}_{\rm ij}) + \frac{\partial}{\partial x_{\rm k}}(\overline{\alpha}\rho\tilde{U}_{\rm k}\tilde{R}_{\rm ij}) &= -\overline{\alpha}\rho\left(\tilde{R}_{\rm ik}\frac{\partial\tilde{U}_{\rm j}}{\partial x_{\rm k}} + \tilde{R}_{\rm jk}\frac{\partial\tilde{U}_{\rm i}}{\partial x_{\rm k}}\right) + \frac{\partial}{\partial x_{\rm k}}\left[\overline{\alpha}\mu\frac{\partial}{\partial x_{\rm k}}(\tilde{R}_{\rm ij})\right] \\
&- \frac{\partial}{\partial x_{\rm k}}[\overline{\alpha}\rho\overline{u'_{\rm i}u'_{\rm j}u'_{\rm k}}] + \overline{\alpha}\overline{p(\frac{\partial u'_{\rm i}}{\partial x_{\rm j}} + \frac{\partial u'_{\rm j}}{\partial x_{\rm i}})} \\
&- \overline{\alpha}\rho\tilde{\epsilon}_{\rm ij} + \Pi_{\rm R,ij} \qquad (16.5\text{-}143)
\end{aligned}$$

The variables in Equation 16.5-143 are per continuous phase c and the subscript is omitted for clarity. The last term of Equation 16.5-143, $\Pi_{\rm R,ij}$, takes into account the interaction between the continuous and the dispersed phase turbulence. A general model for this term can be of the form:

$$\Pi_{\rm R,ij} = K_{\rm dc}C_{\rm 1,dc}(R_{\rm dc,ij} - R_{\rm c,ij}) + K_{\rm dc}C_{\rm 2,dc}a_{\rm dc,i}b_{\rm dc,j} \qquad (16.5\text{-}144)$$

where C_1 and C_2 are unknown coefficients, $a_{\rm dc,i}$ is the relative velocity, $b_{\rm dc,j}$ represents the drift or the relative velocity, and $R_{\rm dc,ij}$ is the unknown particulate-fluid velocity correlation. To simplify this unknown term, the following assumption has been made:

$$\Pi_{\rm R,ij} = \frac{2}{3}\delta_{\rm ij}\Pi_k \qquad (16.5\text{-}145)$$

where $\delta_{\rm ij}$ is the Kronecker delta, and Π_k represents the modified version of the original Simonin model [317].

$$\Pi_{\rm kc} = K_{\rm dc}(\tilde{k}_{\rm dc} - 2\tilde{k}_{\rm c} + \tilde{V}_{\rm rel}\cdot\tilde{V}_{\rm drift}) \qquad (16.5\text{-}146)$$

where $\tilde{K}_{\rm c}$ represents the turbulent kinetic energy of the continuous phase, $\tilde{k}_{\rm dc}$ is the continuous-dispersed phase velocity covariance and finally, $\tilde{V}_{\rm rel}$ and $\tilde{V}_{\rm drift}$ stand for the relative and the drift velocities, respectively. In order to achieve full closure, the transport equation for the turbulent kinetic energy dissipation rate ($\tilde{\epsilon}$) is required. The modeling of $\tilde{\epsilon}$ together with all other unknown terms in Equation 16.5-146 are modeled in the same way as in [58].

RSM Mixture Turbulence Model

The main assumption for the mixture model is that all phases share the same turbulence field which consequently means that the term $\Pi_{\rm R}$ in the Reynolds stress transport equations (Equation 16.5-143) is neglected. Apart from that, the equations maintain the same form but with phase properties and phase velocities being replaced with mixture properties and mixture velocities. The mixture density, for example, can be expressed as

$$\rho_{\rm m} = \sum_{\rm i=1}^{N} \overline{\alpha_{\rm i}} \rho_i \tag{16.5-147}$$

while mixture velocities can be expressed as

$$\tilde{U}_{\rm m} = \frac{\sum_{\rm i=1}^{N} \overline{\alpha_{\rm i}} \rho_i \tilde{U}_{\rm i}}{\sum_{\rm i=1}^{N} \overline{\alpha_{\rm i}} \rho_i} \tag{16.5-148}$$

where N is the number of species.

16.5.12 Solution Method in ANSYS FLUENT

For Eulerian multiphase calculations, ANSYS FLUENT can solve the phase momentum equations, the shared pressure, and phasic volume fraction equations in a coupled and segregated fashion. The coupled solution for multiphase flows is discussed in detail in Section 24.7.1: Coupled Solution for Multiphase Flows in the separate User's Guide. When solving the equations in a segregated manner, ANSYS FLUENT uses the phase coupled SIMPLE (PC-SIMPLE) algorithm [354] for the pressure-velocity coupling. PC-SIMPLE is an extension of the SIMPLE algorithm [264] to multiphase flows. The velocities are solved coupled by phases, but in a segregated fashion. The block algebraic multigrid scheme used by the density-based solver described in [371] is used to solve a vector equation formed by the velocity components of all phases simultaneously. Then, a pressure correction equation is built based on total volume continuity rather than mass continuity. Pressure and velocities are then corrected so as to satisfy the continuity constraint.

The Pressure-Correction Equation

For incompressible multiphase flow, the pressure-correction equation takes the form

$$\sum_{k=1}^{n} \frac{1}{\rho_{\rm rk}} \left\{ \frac{\partial}{\partial t} \alpha_k \rho_k + \nabla \cdot \alpha_k \rho_k \vec{v}_k' + \nabla \cdot \alpha_k \rho_k \vec{v}_k^* - (\sum_{l=1}^{n} (\dot{m}_{lk} - \dot{m}_{kl})) \right\} = 0 \tag{16.5-149}$$

where $\rho_{\rm rk}$ is the phase reference density for the $k^{\rm th}$ phase (defined as the total volume average density of phase k), $\vec{v}_k'$ is the velocity correction for the $k^{\rm th}$ phase, and $\vec{v}_k^*$ is the value of $\vec{v}_k$ at the current iteration. The velocity corrections are themselves expressed as functions of the pressure corrections.

Volume Fractions

The volume fractions are obtained from the phase continuity equations. In discretized form, the equation of the $k^{\rm th}$ volume fraction is

$$a_{p,k} \alpha_k = \sum_{nb} (a_{nb,k} \alpha_{nb,k}) + b_k = R_k \tag{16.5-150}$$

In order to satisfy the condition that all the volume fractions sum to one,

$$\sum_{k=1}^{n} \alpha_k = 1 \tag{16.5-151}$$

16.5.13 Dense Discrete Phase Model

In the standard formulation of the Largangian multiphase model, described in Chapter 15: Discrete Phase, the assumption is that the volume fraction of the discrete phase is sufficiently low: it is not taken into account when assembling the continuous phase equations. The general form of the mass and momentum conservation equations in ANSYS FLUENT is given in Equations 16.5-152 and 16.5-153 (and also defined in Section 1.2: Continuity and Momentum Equations).

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{v}) = S_{DPM} + S_{other} \tag{16.5-152}$$

$$\frac{\partial \rho \vec{v}}{\partial t} + \nabla \cdot (\rho \vec{v} \vec{v}) = -\nabla p + \nabla \cdot \tau + \rho \vec{g} + \vec{F}_{DPM} + \vec{F}_{other} \tag{16.5-153}$$

To overcome this limitation of the Lagrangian multiphase model, the volume fraction of the particulate phase is accounted for by extending Equations 16.5-152 and 16.5-153 to the following set of equations (see also Section 16.5.3: Conservation Equations, written for phase p):

$$\frac{\partial}{\partial t}\left(\alpha_p \rho_p\right) + \nabla \cdot \left(\alpha_p \rho_p \vec{v}_p\right) = \sum_{q=1}^{nphases} \left(\dot{m}_{qp} - \dot{m}_{pq}\right) \tag{16.5-154}$$

$$\begin{aligned}\frac{\partial}{\partial t}\left(\alpha_p \rho_p \vec{v}_p\right) + \nabla \cdot \left(\alpha_p \rho_p \vec{v}_p \vec{v}_p\right) &= -\alpha_p \nabla p + \nabla \cdot \left[\alpha_p \mu_p \left(\nabla \vec{v}_p + \nabla \vec{v}_p^T\right)\right] \\ &+\alpha_p \rho_p \vec{g} + F_{vm,lift,user} + \sum_{q=1}^{nphases} \left(\vec{K}_{qp}\left(\vec{v}_q - \vec{v}_p\right) + \dot{m}_{qp}\vec{v}_{qp} - \dot{m}_{qp}\vec{v}_{qp}\right) \\ &+ K_{DPM}\left(\vec{v}_{DPM} - \vec{v}_p\right) + S_{DPM,explicit}\end{aligned} \tag{16.5-155}$$

Here, Equation 16.5-154 is the mass conservation equation for an individual phase p and Equation 16.5-155 is the corresponding momentum conservation equation. Currently, the momentum exchange terms (denoted by DPM) are considered only in the primary phase equations.

In the resulting set of equations (one continuity and one momentum conservation equation per phase), those corresponding to a discrete phase are not solved. The solution, such as volume fraction or velocity field, is taken from the Lagrangian tracking solution.

In the context of the phase coupled SIMPLE algorithm (Section 16.5.12: Solution Method in ANSYS FLUENT) and the coupled algorithm for pressure-velocity coupling (Section 24.7.5: Selecting the Solution Method in the separate User's Guide), a higher degree of implicitness is achieved in the treatment of the drag coupling terms. All drag related terms appear as coefficients on the left hand side of the linear equation system.

Limitations

Since the given approach makes use of the Eulerian multiphase model framework, all its limitations are adopted:

- The turbulence models: LES, DES and $k - \omega$ turbulence models are not available.
- The combustion models: PDF Transport model, Premixed, Non-premixed and partially premixed combustion models are not available.
- The radiation models are not available.
- The solidification and melting models are not available.
- The Wet Steam model is not available.
- The real gas model (pressure-based and density-based) is not available.
- The density-based solver and models dependent on it are not available.
- Parallel DPM with the shared memory option is disabled.

Granular Temperature

The solids stress acting on particles in a dense flow situation is modeled via an additional acceleration in the particle force balance Equation 15.2-1.

$$\frac{du_p}{dt} = F_D\left(u - u_p\right) + \frac{g_x\left(\rho_p - \rho\right)}{\rho_p} + F_x + F_{interaction} \tag{16.5-156}$$

The term $F_{interaction}$ models the additional acceleration acting on a particle, resulting from interparticle interaction. It is computed from the stress tensor given by the Kinetic Theory of Granular Flows as

$$F_{interaction} = -\frac{1}{\rho_p}\nabla \cdot \bar{\bar{\tau}}_s \tag{16.5-157}$$

The conservation equation for the granular temperature (kinetic energy of the fluctuating particle motion) is solved with the averaged particle velocity field. Therefore, a sufficient statistical representation of the particle phase is needed to ensure the stable behavior of the granular temperature equation. For details on the Kinetic Theory of Granular Flows, please refer to Section 16.5.3: Conservation Equations – Section 16.5.8: Granular Temperature.

The main advantage over the Eulerian model is that, there is no need to define classes to handle particle size distributions. This is done in a natural way in the Lagrangian formulation [279].

16.5.14 Immiscible Fluid Model

The immiscible fluid model for Eulerian multiphase allows you to use the Geo-Reconstruct and CICSAM sharpening schemes with the explicict VOF option. This model should be enabled only for cases requiring sharp interface treatment between phases. This model might help in overcoming some limitations of the VOF model because of the shared velocity and temperature formulation.

The immiscible fluid model for the Eulerian multiphase model provides the anisotropic drag law, which can be used when modeling free surface flow. This drag law is also used when there is higher drag in the normal direction to the interface and lower drag in the tangential direction to the interface. This model may help in overcoming some limitations of the VOF model because of the shared velocity and temperature formulation.

In some cases, where the flow for a particular phase is important in both the directions (tangential and normal to the interface), using a higher anisotropy ratio will result in numerical instability. Therefore, in those cases, an anisotropy ratio of up to 1000 is recommended, where the anisotropy ratio is defined as

$$\text{anisotropy ratio} = \frac{\text{friction factor}_{\text{normal to interface}}}{\text{friction factor}_{\text{tangential to interface}}}$$

In cases, where flow for a particular phase is important only in one direction (tangential or normal to the interface), a higher anisotropy ratio could be used. The principal directions for this drag are based on the normal and tangential direction to the interface.

Two types of drag formulations exist within the anisotropic drag law: one that is based on the symmetric drag law and the other is based on different viscosity options.

Formulation 1

This is based on the symmetric drag law, where the effective drag coefficient in the principal direction p is described as follows:

$$K, p = K_{symmetric} \;\; \lambda, p \tag{16.5-158}$$

where λ is the friction factor vector in the principal direction. $K_{symmetric}$ is the isotropic drag coefficient obtained from the symmetric drag law.

Formulation 2

The effective drag coefficient in the principal direction p is described as follows:

$$K, p = K, visc, p * vof_i * vof_j K, p = (K_{visc}, p \;\; vof_i \;\; vof_j \tag{16.5-159}$$

where vof_i is the volume fraction for phase i and vof_j is the volume fracion for phase j.

The viscous drag component in the principal direction K_{visc}, p is

$$K_{visc}, p = \frac{\mu}{(l_c l_c)} \lambda, p \tag{16.5-160}$$

where the viscosity options can be any one of the following:

$$\begin{aligned}
\mu &= 0.5(\mu_i + \mu_j) \\
\mu &= \frac{2\mu_i \mu_j}{(\mu_i + \mu_j)} \\
\mu &= \frac{\mu_i \mu_j}{(\mu_i vof_j + \mu_j vof_i)} \\
\mu &= \mu_i vof_i + \mu_j vof_j \\
\mu &= \mu_i \\
\mu &= \mu_j
\end{aligned}$$

and l_c is the length scale.

To learn how to use the immiscible fluid model and the two drag formulations, refer to Section 24.5.8: Including the Immiscible Fluid Model in the separate User's Guide.

16.6 Wet Steam Model Theory

Information is organized into the following subsections:

- Section 16.6.1: Overview and Limitations of the Wet Steam Model
- Section 16.6.2: Wet Steam Flow Equations
- Section 16.6.3: Phase Change Model
- Section 16.6.4: Built-in Thermodynamic Wet Steam Properties

16.6.1 Overview and Limitations of the Wet Steam Model

Overview

During the rapid expansion of steam, a condensation process will take place shortly after the state path crosses the vapor-saturation line. The expansion process causes the super-heated dry steam to first subcool and then nucleate to form a two-phase mixture of saturated vapor and fine liquid droplets known as wet steam.

Modeling wet steam is very important in the analysis and design of steam turbines. The increase in steam turbine exit wetness can cause severe erosion to the turbine blades at the low-pressure stages, and a reduction in aerodynamic efficiency of the turbine stages operating in the wet steam region [237].

ANSYS FLUENT has adopted the Eulerian-Eulerian approach for modeling wet steam flow. The flow mixture is modeled using the compressible Navier-Stokes equations, in addition to two transport equations for the liquid-phase mass-fraction (β), and the number of liquid-droplets per unit volume (η). The phase change model, which involves the formation of liquid-droplets in a homogeneous nonequilibrium condensation process, is based on the classical nonisothermal nucleation theory.

This section describes the theoretical aspects of the wet steam model. Information about enabling the model and using your own property functions and data with the wet steam model is provided in Section 24.6: Setting Up the Wet Steam Model in the separate User's Guide. Solution settings and strategies for the wet steam model can be found in Section 24.7.6: Wet Steam Model in the separate User's Guide. Postprocessing variables are described in Section 24.8.1: Model-Specific Variables in the separate User's Guide.

Limitations

The following restrictions and limitations currently apply to the wet steam model in ANSYS FLUENT:

- The wet steam model is available for the density-based solver only.
- Pressure inlet, mass-flow inlet, and pressure outlet are the only inflow and outflow boundary conditions available.
- When the wet steam model is active, the access to the Create/Edit Materials dialog box is restricted because the fluid mixture properties are determined from the built-in steam property functions or from the user-defined wet steam property functions. Therefore, if solid properties need to be set and adjusted, then it must be done in the Create/Edit Materials dialog box before activating the wet steam model.

16.6.2 Wet Steam Flow Equations

The wet steam is a mixture of two-phases. The primary phase is the gaseous-phase consisting of water-vapor (denoted by the subscript v) while the secondary phase is the liquid-phase consisting of condensed-water droplets (denoted by the subscript l).

The following assumptions are made in this model:

- The velocity slip between the droplets and gaseous-phase is negligible.
- The interactions between droplets are neglected.
- The mass fraction of the condensed phase, β (also known as wetness factor), is small ($\beta < 0.2$).
- Since droplet sizes are typically very small (from approximately 0.1 microns to approximately 100 microns), it is assumed that the volume of the condensed liquid phase is negligible.

From the preceding assumptions, it follows that the mixture density (ρ) can be related to the vapor density (ρ_v) by the following equation:

$$\rho = \frac{\rho_v}{(1-\beta)} \tag{16.6-1}$$

In addition, the temperature and the pressure of the mixture will be equivalent to the temperature and pressure of the vapor-phase.

The mixture flow is governed by the compressible Navier-Stokes equations given in vector form by Equation 18.5-4:

$$\frac{\partial \mathbf{W}}{\partial \mathbf{Q}} \frac{\partial}{\partial t} \int_V \mathbf{Q}\, dV + \oint [\mathbf{F} - \mathbf{G}] \cdot d\mathbf{A} = \int_V \mathbf{H}\, dV \tag{16.6-2}$$

where **Q**=(P,u,v,w,T) are mixture quantities. The flow equations are solved using the same density-based solver algorithms employed for general compressible flows.

To model wet steam, two additional transport equations are needed [138]. The first transport equation governs the mass fraction of the condensed liquid phase (β):

$$\frac{\partial \rho\beta}{\partial t} + \nabla \cdot (\rho \vec{v} \beta) = \Gamma \tag{16.6-3}$$

where Γ is the mass generation rate due to condensation and evaporation (kg per unit volume per second). The second transport equation models the evolution of the number density of the droplets per unit volume:

$$\frac{\partial \rho\eta}{\partial t} + \nabla \cdot (\rho \vec{v} \eta) = \rho I \tag{16.6-4}$$

where I is the nucleation rate (number of new droplets per unit volume per second).

To determine the number of droplets per unit volume, Equation 16.6-1 and the average droplet volume V_d are combined in the following expression:

$$\eta = \frac{\beta}{(1-\beta)V_d(\rho_l/\rho_v)} \tag{16.6-5}$$

where ρ_l is the liquid density and the average droplet volume is defined as

$$V_d = \frac{4}{3}\pi\overline{r}_d^3 \tag{16.6-6}$$

where r_d is the droplet radius.

Together, Equation 16.6-2, Equation 16.6-3, and Equation 16.6-4 form a closed system of equations which, along with Equation 16.6-1, permit the calculation of the wet steam flow field.

16.6.3 Phase Change Model

The following is assumed in the phase change model:

- The condensation is homogeneous (i.e., no impurities present to form nuclei).
- The droplet growth is based on average representative mean radii.
- The droplet is assumed to be spherical.
- The droplet is surrounded by infinite vapor space.
- The heat capacity of the fine droplet is negligible compared with the latent heat released in condensation.

The mass generation rate Γ in the classical nucleation theory during the nonequilibrium condensation process is given by the sum of mass increase due to nucleation (the formation of critically sized droplets) and also due to growth/demise of these droplets [138].

Therefore, Γ is written as:

$$\Gamma = \frac{4}{3}\pi\rho_l I {r_*}^3 + 4\pi\rho_l\eta\overline{r}^2\frac{\partial\overline{r}}{\partial t} \tag{16.6-7}$$

where $\overline{r}$ is the average radius of the droplet, and r_* is the Kelvin-Helmholtz critical droplet radius, above which the droplet will grow and below which the droplet will evaporate. An expression for r_* is given by [387].

$$r_* = \frac{2\sigma}{\rho_\mathrm{l} RT \ln S} \qquad (16.6\text{-}8)$$

where σ is the liquid surface tension evaluated at temperature T, ρ_l is the condensed liquid density (also evaluated at temperature T), and S is the super saturation ratio defined as the ratio of vapor pressure to the equilibrium saturation pressure:

$$s = \frac{P}{P_\mathrm{sat}(T)} \qquad (16.6\text{-}9)$$

The expansion process is usually very rapid. Therefore, when the state path crosses the saturated-vapor line, the process will depart from equilibrium, and the supersaturation ratio S can take on values greater than one.

The condensation process involves two mechanisms, the transfer of mass from the vapor to the droplets and the transfer of heat from the droplets to the vapor in the form of latent heat. This energy transfer relation was presented in [385] and used in [138] and can be written as:

$$\frac{\partial \overline{r}}{\partial t} = \frac{P}{h_\mathrm{lv}\rho_\mathrm{l}\sqrt{2\pi RT}}\frac{\gamma+1}{2\gamma}C_\mathrm{p}(T_0 - T) \qquad (16.6\text{-}10)$$

where T_0 is the droplet temperature.

The classical homogeneous nucleation theory describes the formation of a liquid-phase in the form of droplets from a supersaturated phase in the absence of impurities or foreign particles. The nucleation rate described by the steady-state classical homogeneous nucleation theory [387] and corrected for nonisothermal effects, is given by:

$$I = \frac{q_\mathrm{c}}{(1+\theta)}\left(\frac{\rho_\mathrm{v}^2}{\rho_\mathrm{l}}\right)\sqrt{\frac{2\sigma}{{M_\mathrm{m}}^3\pi}}\,e^{-\left(\frac{4\pi {r_*}^2\sigma}{3K_\mathrm{b}T}\right)} \qquad (16.6\text{-}11)$$

where q_c is evaporation coefficient, k_b is the Boltzmann constant, M_m is mass of one molecule, σ is the liquid surface tension, and ρ_l is the liquid density at temperature T.

A nonisothermal correction factor, θ, is given by:

$$\theta = \frac{2(\gamma-1)}{(\gamma+1)}\left(\frac{h_\mathrm{lv}}{RT}\right)\left(\frac{h_\mathrm{lv}}{RT} - 0.5\right) \qquad (16.6\text{-}12)$$

where h_lv is the specific enthalpy of evaporation at pressure p and γ is the ratio of specific heat capacities.

16.6.4 Built-in Thermodynamic Wet Steam Properties

There are many equations that describe the thermodynamic state and properties of steam. While some of these equations are accurate in generating property tables, they are not suitable for fast CFD computations. Therefore, ANSYS FLUENT uses a simpler form of the thermodynamic state equations [386] for efficient CFD calculations that are accurate over a wide range of temperatures and pressures. These equations are described below.

Equation of State

The steam equation of state used in the solver, which relates the pressure to the vapor density and the temperature, is given by [386]:

$$P = \rho_v RT(1 + B\rho_v + C{\rho_v}^2) \tag{16.6-13}$$

where B, and C are the second and the third virial coefficients given by the following empirical functions:

$$B = a_1(1 + \frac{\tau}{\alpha})^{-1} + a_2 e^{\tau}(1 - e^{-\tau})^{\frac{5}{2}} + a_3\tau \tag{16.6-14}$$

where B is given in m^3/kg, $\tau = \frac{1500}{T}$ with T given in Kelvin, α = 10000.0, a_1= 0.0015, a_2= -0.000942, and a_3= -0.0004882.

$$C = a(\tau - \tau_0)e^{-\alpha\tau} + b \tag{16.6-15}$$

where C is given in m^6/kg^2, $\tau = \frac{T}{647.286}$ with T given in Kelvin, τ_o= 0.8978, α=11.16, a= 1.772, and b= 1.5E-06.

The two empirical functions that define the virial coefficients B and C cover the temperature range from 273 K to 1073 K.

The vapor isobaric specific heat capacity Cp_v is given by:

$$Cp_v = Cp_0(T) + R\left[[(1 - \alpha_v T)(B - B_1) - B_2]\rho_v + \left[(1 - 2\alpha_v T)C + \alpha_v T C_1 - \frac{C_2}{2}\right]{\rho_v}^2\right] \tag{16.6-16}$$

The vapor specific enthalpy, h_v is given by:

$$h_v = h_0(T) + RT\left[(B - B_1)\rho_v + (C - \frac{C_1}{2}){\rho_v}^2\right] \tag{16.6-17}$$

The vapor specific entropy, s_v is given by:

$$s_v = s_0(T) - R\left[\ln\rho_{\mathrm{v}} + (B+B_1)\rho_{\mathrm{v}} + \frac{(C+C_1)}{2}{\rho_{\mathrm{v}}}^2\right] \tag{16.6-18}$$

The isobaric specific heat at zero pressure is defined by the following empirical equation:

$$Cp_0(T) = \sum_{i=1}^{6} a_i T^{i-2} \tag{16.6-19}$$

where Cp_0 is in KJ/kg K, $a_1 = 46.0$, $a_2 = 1.47276$, $a_3 = 8.38930\text{E-}04$, $a_4 = \text{-}2.19989\text{E-}07$, $a_5 = 2.46619\text{E-}10$, and $a_6 = \text{-}9.70466\text{E-}14$.

and

$B_1 = T\frac{dB}{dT}$, $C_1 = T\frac{dC}{dT}$, $B_2 = T^2\frac{dB^2}{dT^2}$, and $C_2 = T^2\frac{dC^2}{dT^2}$.

Both $h_0(T)$ and $s_0(T)$ are functions of temperature and they are defined by:

$$h_0(T) = \int Cp_0 dT + h_c \tag{16.6-20}$$

$$s_0(T) = \int \frac{Cp_0}{T} dT + s_c \tag{16.6-21}$$

where h_c and s_c are arbitrary constants.

The vapor dynamic viscosity μ_v and thermal conductivity Kt_v are also functions of temperature and were obtained from [385].

Saturated Vapor Line

The saturation pressure equation as a function of temperature was obtained from [290]. The example provided in Section 24.6.5: UDWSPF Example in the separate User's Guide contains a function called `wetst_satP()` that represents the formulation for the saturation pressure.

Saturated Liquid Line

At the saturated liquid-line, the liquid density, surface tension, specific heat Cp, dynamic viscosity, and thermal conductivity must be defined. The equation for liquid density, ρ_{l}, was obtained from [290]. The liquid surface tension equation was obtained from [385]. While the values of Cp_l , μ_l and Kt_l were curve fit using published data from [83] and then written in polynomial forms. The example provided in Section 24.6.5: UDWSPF Example in the separate User's Guide contains functions called `wetst_cpl()`, `wetst_mul()`, and `wetst_ktl()` that represent formulations for Cp_l , μ_l and Kt_l.

Mixture Properties

The mixture properties are related to vapor and liquid properties via the wetness factor using the following mixing law:

$$\phi_m = \phi_l \beta + (1 - \beta)\phi_v \tag{16.6-22}$$

where ϕ represents any of the following thermodynamic properties: h, s, Cp, Cv, μ or Kt.

16.7 Modeling Mass Transfer in Multiphase Flows

This section describes the modeling of mass transfer in the framework of ANSYS FLUENT's general multiphase models (i.e., Eulerian multiphase, mixture multiphase, VOF multiphase). There are numerous kinds of mass transfer processes that can be modeled in ANSYS FLUENT. You can use models available in ANSYS FLUENT (e.g. ANSYS FLUENT's cavitation model), or define your own mass transfer model via user-defined functions. See Section 16.7.3: UDF-Prescribed Mass Transfer and the separate UDF Manual for more information about the modeling of mass transfer via user-defined functions.

Information about mass transfer is presented in the following subsections:

- Section 16.7.1: Source Terms due to Mass Transfer
- Section 16.7.2: Unidirectional Constant Rate Mass Transfer
- Section 16.7.3: UDF-Prescribed Mass Transfer
- Section 16.7.4: Cavitation Models
- Section 16.7.5: Evaporation-Condensation Model

16.7.1 Source Terms due to Mass Transfer

ANSYS FLUENT adds contributions due to mass transfer only to the momentum, species, and energy equations. No source term is added for other scalars such as turbulence or user-defined scalars.

Let $m_{p^i q^j}$ be the mass transfer rate per unit volume from the i^{th} species of phase p to the j^{th} species of phase q. In case a particular phase does not have a mixture material associated with it, the mass transfer will be with the bulk phase.

Mass Equation

The contribution to the mass source for phase p in a cell is

$$m_p = -m_{p^i q^j} \tag{16.7-1}$$

and for phase q is

$$m_q = m_{p^i q^j} \tag{16.7-2}$$

Momentum Equation

For VOF or mixture models, there is no momentum source.

For the Eulerian model, the momentum source in a cell for phase p is

$$m_p \vec{u}_p = -m_{p^i q^j} \vec{u}_p \tag{16.7-3}$$

and for phase q is

$$m_q \vec{u}_q = m_{p^i q^j} \vec{u}_p \tag{16.7-4}$$

Energy Equation

For all multiphase models, the following energy sources are added.

The energy source in a cell for phase p is

$$H_p = -m_{p^i q^j} (h_p^i) \tag{16.7-5}$$

and for phase q is

$$H_q = m_{p^i q^j} (h_p^i + {h^f}_p^i - {h^f}_q^j) \tag{16.7-6}$$

where ${h^f}_p^i$ and ${h^f}_q^j$ are the formation enthalpies of species i of phase p and species j of phase q respectively and h_p^i is the enthalpy of species i of phase p (with reference to the formation enthalpy).

Species Equation

The species source in a cell for species i of phase p is

$$m_p^i = -m_{p^i q^j} \tag{16.7-7}$$

and for species j of phase q is

$$m_q^j = m_{p^i q^j} \tag{16.7-8}$$

Other Scalar Equations

No source/sink terms are added for turbulence quantities and other scalars. The transfer of these scalar quantities due to mass transfer could be modeled using user-defined source terms.

16.7.2 Unidirectional Constant Rate Mass Transfer

The unidirectional mass transfer model defines a positive mass flow rate per unit volume from phase p to phase q:

$$\dot{m_{pq}} = \max[0, \lambda_{pq}] - \max[0, -\lambda_{pq}] \tag{16.7-9}$$

where

$$\lambda_{pq} = \dot{r}\alpha_p\rho_q \tag{16.7-10}$$

and $\dot{r}$ is a constant rate of particle shrinking or swelling, such as the rate of burning of a liquid droplet. This is not available for the VOF model.

If phase p is a mixture material and a mass transfer mechanism is defined for species i of phase p, then

$$\lambda_{pq} = \dot{r}\alpha_p y_{\rm p,i}\rho_q \tag{16.7-11}$$

where $y_{\rm p,i}$ is the mass fraction of species i in phase p.

16.7.3 UDF-Prescribed Mass Transfer

Because there is no universal model for mass transfer, ANSYS FLUENT provides a UDF that you can use to input models for different types of mass transfer, e.g. evaporation, condensation, boiling, etc. Note that when using this UDF, ANSYS FLUENT will automatically add the source contribution to all relevant momentum and scalar equations. This contribution is based on the assumption that the mass “created” or “destroyed” will have the same momentum and energy of the phase from which it was created or destroyed. If you would like to input your source terms directly into momentum, energy, or scalar equations, then the appropriate path is to use UDFs for user-defined sources for all equations, rather than the UDF for mass transfer. See the separate UDF Manual for more information about UDF-based mass transfer in multiphase.

16.7.4 Cavitation Models

A liquid at constant temperature can be subjected to a decreasing pressure, which may fall below the saturated vapor pressure. The process of rupturing the liquid by a decrease of pressure at constant temperature is called cavitation. The liquid also contains the micro-bubbles of noncondensable (dissolved or ingested) gases, or nuclei, which under decreasing pressure may grow and form cavities. In such processes, very large and steep density variations happen in the low-pressure/cavitating regions.

This section provides information about the following three cavitation models used in ANSYS FLUENT.

- Singhal et al. model: You can use this model to include cavitation effects in two-phase flows when the mixture model is used. This is also known as the Full Cavitaton Model, which has been implemented in ANSYS FLUENTsince Version 6.1.
- Zwart-Gerber-Belamri model: You can use this model in both the mixture and Eulerian multiphase models.
- Schnerr and Sauer model: This is the default model. You can use this model in both the mixture and Eulerian multiphase models.

The following assumptions are made in the standard two-phase cavitation models:

- The system under investigation must consist of a liquid and a vapor phase.
- Mass transfer between the liquid and vapor phase is assumed to take place. Both bubble formation (evaporation) and collapse (condensation) are taken into account in the cavitation models.
- The cavitation models are based on the Rayleigh-Plesset equation, describing the growth of a single vapor bubble in a liquid.
- In the Singhal et al. model, noncondensable gases have been introduced into the system. The mass fraction of the noncondensable gases is assumed to be a known constant.
- The input material properties used in the cavitation models can be constants, functions of temperature or user-defined.

The cavitation models offer the following capabilities:

- The Singhal et al. model can be used to account for the effect of noncondensable gases. The Zwart-Gerber-Belamri and Schnerr and Sauer models do not include the noncondensable gases in the basic model terms.

- The Zwart-Gerber-Belamri and Schnerr and Sauer models are compatible with all the turbulence models available in ANSYS FLUENT.
- Both the pressure-based segregated and coupled solvers are available with the cavitation models.
- They are all fully compatible with dynamic mesh and non-conformal interfaces.
- Both liquid and vapor phases can be incompressible or compressible. For compressible liquids, the density is described using a user-defined function. See the separate UDF Manual for more information on user-defined density functions.

Limitations of the Cavitation Models

The following limitations apply to the cavitation models in ANSYS FLUENT:

- None of the cavitation models can be used with the VOF model because the surface tracking schemes for the VOF model are incompatible with the interpenetrating continua assumption of the cavitation models.
- They can only be used for cavitating flows occurring in a single liquid fluid.
- The Singhal et al. model requires the primary phase to be a liquid and the secondary phase to be a vapor. This model is only compatible with the multiphase mixture model.
- The Singhal et al. model cannot be used with the Eulerian multiphase model.
- The Singhal et al. model is not compatible with the LES turbulence model.
- The Zwart-Gerber-Belamri and Schnerr and Sauer models do not take the effect of noncondensable gases into account by default.

Vapor Transport Equation

With the multiphase cavitation modeling approach, a basic two-phase cavitation model consists of using the standard viscous flow equations governing the transport of mixture (Mixture model) or phases (Eulerian multiphase), and a conventional turbulence model (k-ϵ model). In cavitation, the liquid-vapor mass transfer (evaporation and condensation) is governed by the vapor transport equation:

$$\frac{\partial}{\partial t}(\alpha\rho_{\rm v}) + \nabla.(\alpha\rho_{\rm v}\vec{\rm V}_{\rm v}) = {\rm R_e} - {\rm R_c} \qquad (16.7\text{-}12)$$

where,

v	=	vapor phase
α	=	vapor volume fraction
ρ_v	=	vapor density
$\vec{V}_v$	=	vapor phase velocity
R_e, R_c	=	mass transfer source terms connected to the growth and collapse of the vapor bubbles repectively

In Equation 16.7-12, the terms R_e and R_c account for the mass transfer between the vapor and liquid phases in cavitation. In ANSYS FLUENT, they are modeled based on the Rayleigh-Plesset equation describing the growth of a single vapor bubble in a liquid.

Bubble Dynamics Consideration

In most engineering situations we assume that there are plenty of nuclei for the inception of cavitation. Thus, our primary focus is on proper accounting of bubble growth and collapse. In a flowing liquid with zero velocity slip between the fluid and bubbles, the bubble dynamics equation can be derived from the generalized Rayleigh-Plesset equation as [37]

$$\Re_B \frac{D^2\Re_B}{Dt^2} + \frac{3}{2}(\frac{D\Re_B}{Dt})^2 = (\frac{P_B - P}{\rho_\ell}) - \frac{4\mathrm{v}_\ell}{\Re_B}\Re_B - \frac{2S}{\rho_\ell \Re_\mathrm{B}} \qquad (16.7\text{-}13)$$

where,

$\Re_B$	=	bubble radius
σ	=	liquid surface tension coefficient
ρ_ℓ	=	liquid density
P_B	=	bubble surface pressure
P	=	local far-field pressure

Neglecting the second-order terms and the surface tension force, Equation 16.7-13 is simplified to

$$\frac{D\Re_B}{Dt} = \sqrt{\frac{2}{3}\frac{P_B - P}{\rho_\ell}} \qquad (16.7\text{-}14)$$

This equation provides a physical approach to introduce the effects of bubble dynamics into the cavitation model. It can also be considered to be an equation for void propagation and, hence, mixture density.

Singhal et al. Model

This cavitation model is based on the "full cavitation model", developed by Singhal et al. [318]. It accounts for all first-order effects (i.e., phase change, bubble dynamics, turbulent pressure fluctuations, and noncondensable gases). It has the capability to account for multiphase (N-phase) flows or flows with multiphase species transport, the effects of slip velocities between the liquid and gaseous phases, and the thermal effects and compressibility of both liquid and gas phases. The cavitation model can be used with the mixture multiphase model, with or without slip velocities. However, it is always preferable to solve for cavitation using the mixture model without slip velocity; slip velocities can be turned on if the problem suggests that there is significant slip between phases.

To derive an expression of the net phase change rate, Singhal et al. [318] uses the following two-phase continuity equations:

Liquid phase:

$$\frac{\partial}{\partial t}[(1-\alpha)\rho_\ell] + \nabla.[(1-\alpha)\rho_\ell \vec{V}] = -R \qquad (16.7\text{-}15)$$

Vapor phase:

$$\frac{\partial}{\partial t}(\alpha\rho_v) + \nabla.(\alpha\rho_v \vec{V}) = R \qquad (16.7\text{-}16)$$

Mixture:

$$\frac{\partial}{\partial t}(\rho) + \nabla.(\rho\vec{V}) = 0 \qquad (16.7\text{-}17)$$

where,

ℓ = liquid phase
ρ = mixture density (function of phase volume fraction and density)

Mixture density ρ is defined as

$$\rho = \alpha\rho_v + (1-\alpha)\rho_\ell \qquad (16.7\text{-}18)$$

Combining Equations 16.7-15, 16.7-16, and 16.7-17 yields a relationship between the mixture density and vapor volume fraction (α):

$$\frac{D\rho}{Dt} = -(\rho_\ell - \rho_v)\frac{D\alpha}{Dt} \qquad (16.7\text{-}19)$$

The vapor volume fraction (α) can be related to the bubble number density (n) and the radius of bubble ($\Re_B$) as

$$\alpha = n \times (\frac{4}{3}\pi {\Re_B}^3) \tag{16.7-20}$$

Substituting Equation 16.7-20 into Equation 16.7-19 gives the following:

$$\frac{D\rho}{Dt} = -(\rho_\ell - \rho_v)(n4\pi)^{\frac{1}{3}}(3\alpha)^{\frac{2}{3}}\frac{D\Re_B}{Dt} \tag{16.7-21}$$

Using the Equation 16.7-14, and combining Equations 16.7-15, 16.7-16, 16.7-19, and 16.7-21, the expression for the net phase change rate (R) is finally obtained as

$$R = (n4\pi)^{\frac{1}{3}}(3\alpha)^{\frac{2}{3}}\frac{\rho_v\rho_\ell}{\rho}[\frac{2}{3}(\frac{P_B - P}{\rho_\ell})]^{\frac{1}{2}} \tag{16.7-22}$$

Here R represents the vapor generation or evaporation rate, i.e. the source term R_e in Equation 16.7-12. All terms, except n, are either known constants or dependent variables. In the absence of a general model for estimation of the bubble number density, the phase change rate expression is rewritten in terms of bubble radius ($\Re_B$), as follows:

$$R = \frac{3\alpha}{\Re_B}\frac{\rho_v\rho_\ell}{\rho}\sqrt{\frac{2}{3}\frac{(P_B - P)}{\rho_\ell}} \tag{16.7-23}$$

Equation 16.7-23 indicates that the unit volume mass transfer rate is not only related to the vapor density (ρ_v), but the function of the liquid density (ρ_ℓ), and the mixture density (ρ) as well. Since Equation 16.7-23 is derived directly from phase volume fraction equations, it is exact and should accurately represent the mass transfer from liquid to vapor phase in cavitation (bubble growth or evaporation). As for bubble collapse or the condensation process, though it is expected to be different from that of bubble growth, Equation 16.7-23 is, as a first approximation, also often used to model the bubble collapse by using the absolute value of the pressure difference and treating the right side as a sink term.

It may be noted that in practical cavitation models, the local far-field pressure P is usually taken to be the same as the cell center pressure. The bubble pressure P_B is equal to the saturation vapor pressure in the absence of dissolved gases, mass transport and viscous damping, i.e., $P_B = P_v$.

where,

P_B = bubble pressure
P_v = saturation vapor pressure

Based on Equation 16.7-23, Singhal et al. [394] proposed a model where the vapor mass fraction is the dependent variable in the transport equation. This model accommodates also a single phase formulation where the governing equations is given by:

$$\frac{\partial}{\partial t}(f_v \rho) + \nabla.(f_v \rho \vec{V}_v) = \nabla.(\Gamma \nabla f_v) + R_e - R_c \tag{16.7-24}$$

where,

f_v = vapor mass fraction
f_g = noncondensable gases
Γ = diffusion coefficient

The rates of mass exchange are given by the following equations:

If $P \leq P_v$

$$R_e = F_{vap} \frac{max(1.0, \sqrt{k})(1 - f_v - f_g)}{\sigma} \rho_l \rho_v \sqrt{\frac{2}{3} \frac{(P_v - P)}{\rho_{ell}}} \tag{16.7-25}$$

If $P \, P_v$

$$R_c = F_{cond} \frac{max(1.0, \sqrt{k}) f_v}{\sigma} \rho_l \rho_l \sqrt{\frac{2}{3} \frac{(P_v - P)}{\rho_l}} \tag{16.7-26}$$

The saturation pressure is corrected by an estimation of the local values of the turbulent pressure fluctuations:

$$P_v = P_{sat} + \frac{1}{2}(0.39k) \tag{16.7-27}$$

The constants have the values $F_{vap} = 0.02$ and $F_{cond} = 0.01$. In this model, the liquid-vapor mixture is assumed to be compressible. Also, the effects of turbulence and the noncondensable gases have been taken into account.

Zwart-Gerber-Belamri Model

Assuming that all the bubbles in a system have the same size, Zwart-Gerber-Belamri [395] proposed that the total interphase mass transfer rate per unit volume (R) is calculated using the bubble density numbers (n), and the mass change rate of a single bubble:

$$R = n \times (4\pi {\Re_B}^2 \rho_v \frac{D\Re_B}{Dt}) \tag{16.7-28}$$

Substituting the value of n in Equation 16.7-28 into Equation 16.7-20, we have the expression of the net mass transfer:

$$R = \frac{3\alpha\rho_v}{\Re_B}\sqrt{\frac{2}{3}\frac{P_B - P}{\rho_\ell}} \tag{16.7-29}$$

Comparing Equations 16.7-29 and 16.7-23, you will notice that the difference is only in the density terms in the mass transfer rate. In Equation 16.7-29, the unit voume mass transfer rate is only related to the vapor phase density (ρ_v). Unlike Equation 16.7-23, R has no relation with the liquid phase and mixture densities in this model.

As in Equation 16.7-23, Equation 16.7-29 is derived assuming bubble growth (evaporation). To apply it to the bubble collapse process (condensation), the following generalized formulation is used:

$$R_e = F\frac{3\alpha\rho_v}{\Re_B}\sqrt{\frac{2}{3}\frac{|P_B - P|}{\rho_\ell}}sign(P_B - P) \tag{16.7-30}$$

where F is an empirical calibration coefficient. Though it is originally derived from evaporation, Equation 16.7-30 only works well for condensation. It is physically incorrect and numerically unstable if applied to evaporation. The fundamental reason is that one of the key assumptions is that the cavitation bubble does not interact with each other. This is plausible only during the earliest stage of cavitation when the cavitation bubble grows from the nucleation site. As the vapor volume fraction increases, the nucleation site density must decrease accordingly. To model this process, Zwart-Gerber-Belamri proposed to replace α_v with $\alpha_{\rm nuc}(1-\alpha_v)$ in Equation 16.7-30. Then the final form of this cavitation model is as follows:

If $P \leq P_v$

$$R_e = F_{\rm vap}\frac{3\alpha_{\rm nuc}(1-\alpha_v)\rho_v}{\Re_B}\sqrt{\frac{2}{3}\frac{P_v - P}{\rho_\ell}} \tag{16.7-31}$$

If $P \geq P_v$

$$R_c = F_{\rm cond}\frac{3\alpha_v\rho_v}{\Re_B}\sqrt{\frac{2}{3}\frac{P - P_v}{\rho_\ell}} \tag{16.7-32}$$

where,

$\Re_B$	=	bubble radius	=	$10^{-6}m$
α_{nuc}	=	nucleation site volume fraction	=	5×10^{-4}
$F_{\rm vap}$	=	evaporation coefficient	=	50
$F_{\rm cond}$	=	condensation coefficient	=	0.001

Schnerr and Sauer Model

As in the Singhal et al. model, Schnerr and Sauer [309] follow a similar approach to derive the exact expression for the net mass transfer from liquid to vapor. The equation for the vapor volume fraction has the general form:

$$\frac{\partial}{\partial t}(\alpha \rho_v) + \nabla.(\alpha \rho_v \vec{V}) = \frac{\rho_v \rho_l}{\rho} \frac{D\alpha}{Dt} \tag{16.7-33}$$

Here, the net mass source term is as follows:

$$R = \frac{\rho_v \rho_l}{\rho} \frac{d\alpha}{dt} \tag{16.7-34}$$

Unlike Zwart-Gerber-Belamri and Singhal et al., Schnerr and Sauer use the following expression to connect the vapor volume fraction to the number of bubbles per volume of liquid:

$$\alpha = \frac{n_b \frac{4}{3} \pi {\Re_B}^3}{1 + n_b \frac{4}{3} \pi {\Re_B}^3} \tag{16.7-35}$$

Following a similar approach to Singhal et al., they derived the following equation:

$$R = \frac{\rho_v \rho_l}{\rho} \alpha(1-\alpha) \frac{3}{\Re_B} \sqrt{\frac{2}{3} \frac{(P_v - P)}{\rho_l}} \tag{16.7-36}$$

$$\Re_B = (\frac{\alpha}{1-\alpha} \frac{3}{4\pi} \frac{1}{n})^{\frac{1}{3}} \tag{16.7-37}$$

where,

R	=	mass transfer rate
$\Re_B$	=	bubble radius

Comparing Equation 16.7-36 with Equations 16.7-23 and 16.7-29, it is obvious that unlike the two previous models, the mass transfer rate in the Schnerr and Sauer model is proportional to $\alpha_v(1-\alpha_v)$. Moreover, the function $f(\alpha_v, \rho_v, \rho_l) = \frac{\rho_v \rho_l}{\rho} \alpha(1-\alpha)$ has the interesting property that it approaches zero when $\alpha = 0$ and $\alpha = 1$, and reaches the maximum in between. Also in this model, the only parameter which must be determined is the number of spherical bubbles per volume of liquid. If you assume that no bubbles are created or destroyed, the bubble number density would be constant. The initial conditions for the nucleation site volume fraction and the equilibrium bubble radius would therefore be sufficient to specify the bubble number density (n) from Equation 16.7-35 and then the phase transition by Equation 16.7-36.

As in the two other models, Equation 16.7-36 is also used to model the condensation process. The final form of the model is as follows:

When $P_v \geq P$,

$$R_e = \frac{\rho_v \rho_l}{\rho} \alpha(1-\alpha) \frac{3}{\Re_B} \sqrt{\frac{2}{3} \frac{(P_v - P)}{\rho_l}} \tag{16.7-38}$$

When $P_v \leq P$,

$$R_c = \frac{\rho_v \rho_l}{\rho} \alpha(1-\alpha) \frac{3}{\Re_B} \sqrt{\frac{2}{3} \frac{(P - P_v)}{\rho_l}} \tag{16.7-39}$$

Additional Guidelines for the Cavitation Models

In practical applications of a cavitation model, several factors greatly influence numerical stability. For instance, the high pressure difference between the inlet and exit, large ratio of liquid to vapor density, and large phase change rates between the liquid and vapor all have unfavorable effects on solution convergence. In addition, poor initial conditions very often lead to an unrealistic pressure field and unexpected cavitating zones, which, once present, are usually difficult to correct. You may consider the following factors/tips when choosing a cavitation model and addressing potential numerical problems:

- Choice of the cavitation models

  In ANSYS FLUENT, there are three available cavitation models. The Zwart-Gerber-Belamri and the Schnerr and Sauer models have been implemented following an entirely different numerical procedure from the Singhal et al. model developed in ANSYS FLUENT 6.1. Numerically, these two models are robust and converge quickly. It is therefore highly recommended that you should use the Schnerr and Sauer or the Zwart-Gerber-Bleamri model. The Singhal et al. model, though physically similar to the other two, is numerically less stable and more difficult to use.

- Choice of the solvers

  In ANSYS FLUENT, both the segregated (SIMPLE, SIMPLEC, and PISO) and coupled pressure-based solvers can be used in cavitation. As usual, the coupled solver is generally more robust and converges faster, particularly for cavitating flows in rotating machinery (liquid pumps, inducers, imprellers, etc). For fuel injector applications, however, the segregated solver also performs very well with the Schnerr and Sauer and the Zwart-Gerber-Belamri models.

  As for the Singhal et al. model, since the coupled solver does not show any significant advantages, it is suggested that the segregated solver is used.

- Initial conditions

  Though no special initial condition settings are required, we suggest that the vapor fraction is always set to inlet values. The pressure is set close to the highest pressure among the inlets and outlets to avoid unexpected low pressure and cavitating spots. In gereral, the Schnerr and Sauer and Zwart-Gerber-Belamri models are robust enough so that there is no need for specific initial conditions. But in some very complicated cases, it may be beneficial to obtain a realistic pressure field before substantial cavities are formed. This can be achieved by obtaining a converged/near-converged solution for a single phase liquid flow, and then enabling the cavitation model. Again, the Singhal et al. model is much more sensitive to initial conditions. The above mentioned treatments are generally required.

- Pressure discretization schemes

  As for general multiphase flows, it is more desirable to use the following pressure discretization schemes in cavitation applications in this order:

  – PRESTO!
  – body force weighted
  – second order

  The standard and linear schemes generally are not very effective in complex cavitating flows and you should avoid using them.

- Relaxation factors

  – Schnerr and Sauer and Zwart-Gerber-Belamri models

    The default settings generally work well. To achieve numerical efficiency, however, the following values may be recommended:

    ∗ The relaxation factor for vapor is 0.5 or higher unless the solution diverges or all the residuals oscillate excessively.
    ∗ The density and the vaporization mass can be relaxed, but in general set them to 1.
    ∗ For the segregated solver, the relaxation factor for pressure should be no less than the value for the momentum equations.
    ∗ For the coupled solver, the default value for the Courant number (200) may need to be reduced to 20-50 in some complex 3D cases.

  – Singhal et al. model

    In general, small relaxation factors are recommended for momentum equations, usually between 0.05 – 0.4. The relaxation factor for the pressure-correction equation should generally be larger than those for momentum equations, say in the range of 0.2 – 0.7. The density and the vaporization mass (source term in the vapor equation) can also be relaxed to improve convergence. Typically, the relaxation factor for density is set between the values of 0.3 and 1.0, while for the vaporization mass, values between 0.1 and 1.0 may be appropriate. For some extreme cases, even smaller relaxation factors may be required for all the equations.

- Special tips for the Singal et al. model
  - Noncondensable gases

    Noncondensable gases are usually present in liquids. Even a small amount (e.g., 15 ppm) of noncondensable gases can have significant effects on both the physical results and the convergence characteristics of the solution. A value of zero for the mass fraction of noncondensable gases should generally be avoided. In some cases, if the liquid is purified of noncondensable gases, a much smaller value (e.g., 10^{-8}) may be used to replace the default value of 1.5×10^{-5}. In fact, higher mass fractions of the noncondensable gases may in many cases enhance numerical stability and lead to more realistic results. In particular, when the saturation pressure of a liquid at a certain temperature is zero or very small, noncondensable gases will play a crucial role both numerically and physically.
  - Limits for dependent variables

    In many cases, setting the pressure upper limit to a reasonable value can help convergence greatly at the early stage of the solution. It is advisable to always limit the maximum pressure when possible. By default, ANSYS FLUENT sets the maximum pressure limit to 5.0×10^{10} Pascal.
  - Relaxation factor for the pressure correction equation

    For cavitating flows, a special relaxation factor is introduced for the pressure correction equation. By default, this factor is set to 0.7, which should work well for most of the cases. For some very complicated cases, however, you may experience the divergence of the AMG solver. Under those circumstances, this value may be reduced to no less than 0.4. You can set the value of this relaxation factor by typing a text command. For more information, contact your ANSYS FLUENT support engineer.

Extended Cavitation Model Capabilites

When cavitation occurs, in many practical applications other gaseous species exist in the systems. For instance, in a ventilated supercavitating vehicle, air is injected into a liquid to stabilize or increase the cavitation along the vehicle surfaces. In some cases, the incoming flow is a mixture of a liquid and some gaseous species. To predict those types of cavitating flows, the basic two-phase cavitation model needs to be extended to a multiphase (N-phase) flow, or a multiphase species transport cavitation model.

Multiphase Cavitation Models

The multiphase cavitation models are the extensions of the three basic two-phase cavitation models to multiphase flows. In addition to the primary liquid and secondary vapor phase, more secondary gaseous phases can be included into the computational system under the following assumptions:

- Mass transfer (cavitation) only occurs between the liquid and vapor phase.
- The basic two-phase cavitation models are still used to model the phase changes between the liquid and vapor.
- Only one secondary phase can be defined as compressible gas phase, while a user-defined density may be applied to all the phases.
- In the Singhal et al. model, the predescribed noncondensable gases can be included in the system. To exclude noncondensable gases from the system, the mass fraction needs to be set to 0, and the noncondensable gas needs to be modeled by a separate compressible gas phase.
- For noncavitating phase (i), the general transport equation governing the vapor phase is the volume fraction equation in the Zwart-Gerber-Belamri and Schnerr and Sauer models, while in the Singhal et al. model, a mass transfer equation is solved and the vapor must be the second phase.

Multiphase Species Transport Cavitation Model

In some cases, there are several gas phase components in a system which can be considered compressible. Since only one compressible gas phase is allowed in the general multiphase approach, the multiphase species transport approach offers an option to handle these types of applications by assuming that there is one compressible gas phase with multiple species.

The detailed description of the multiphase species transport approach can be found in Section 16.8: Modeling Species Transport in Multiphase Flows. The multiphase species transport cavitation model can be summarized as follows:

- All the assumptions/limitations for the multiphase cavitation model apply here.
- The primary phase can only be a single liquid.
- All the secondary phases allow more than one species.
- The mass transfer between a liquid and a vapor phase/species is modeled by the basic cavitation models.

- The mass transfer between other phases or species are modeled with the standard mass transfer approach. In the standard model, zero constant rates should be chosen.
- For the phases with multiple species, the phase shares the same pressure as the other phases, but each species has its own pressure (i.e., partial pressure). As a result, the vapor density and the pressure used in Equation 16.7-26 are the partial density and pressure of the vapor.

16.7.5 Evaporation-Condensation Model

The evaporation-condensation model is a mechanistic model [185], with a physical basis. It is available with the mixture and Eulerian multiphase models.

Based on the following temperature regimes, the mass transfer can be described as follows:

If $T > T_{sat}$

$$\dot{m}_{e\rightarrow v} = coeff * \alpha_l \rho_l \frac{(T - T_{sat})}{T_{sat}} \tag{16.7-40}$$

If $T < T_{sat}$

$$\dot{m}_{e\rightarrow v} = coeff * \alpha_v \rho_v \frac{(T - T_{sat})}{T_{sat}} \tag{16.7-41}$$

$\dot{m}_{e\rightarrow v}$ represents the rate of mass transfer from the liquid phase to the vapor phase, with units of $kg/s/m^3$. $coeff$ is a coefficient that needs to be fine tuned and can be interpreted as a relaxation time. α and ρ are the phase volume fraction and density, respectively. The source term for the energy equation can be obtained by multiplying the rate of mass transfer by the latent heat.

Consider the Hertz Knudsen formula, which gives the evaporation-condensation flux based on the kinetic theory for a flat interface:

$$F = \beta \sqrt{\frac{M}{2\pi R T_{sat}}} (P^* - P_{sat}) \tag{16.7-42}$$

The flux has units of $kg/s/m^2$, P is the pressure, T is the temperature, and R is the universal gas constant. The coefficient β is the so-called accommodation coefficient that shows the portion of vapor molecules going into the liquid surface and adsorbed by this surface. P^* represents the vapor partial pressure at the interface on the gas side. The Clapeyron-Clausius equation relates the pressure to the temperature for the saturation condition. (It is obtained by equating the vapor and liquid chemical potentials):

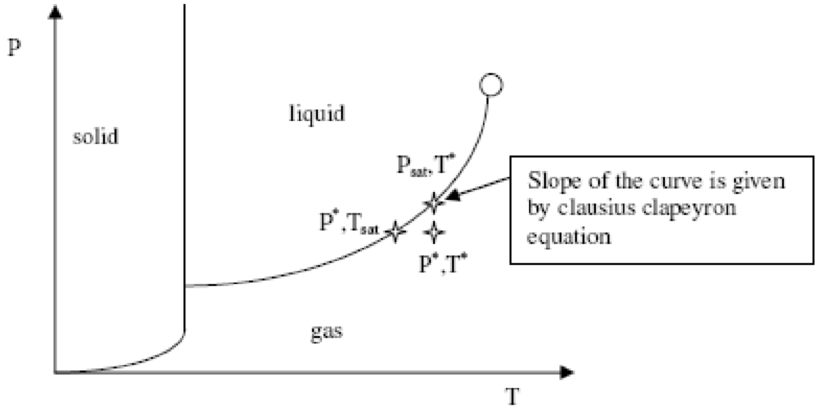


Figure 16.7.1: The Stability Phase Diagram

$$\frac{dP}{dT} = \frac{L}{T(v_g - v_l)} \tag{16.7-43}$$

v_g and v_l are the inverse of the density for the gas and liquid (volume per mass unit), respectively. L is the latent heat (J/kg).

Based on this differential expression, we can obtain variation of temperature from variation of pressure close to the saturation condition.

The Clausius Clapeyron equation yields the following formula as long as P^* and T^* are close to the saturation condition:

$$(P^* - P_{sat}) = -\frac{L}{T(v_g - v_l)}(T^* - T_{sat}) \tag{16.7-44}$$

Using this relation in the above Hertz Knudsen equation yields [346]

$$F = \beta\sqrt{\frac{M}{2\pi RT_{sat}}}L\left(\frac{\rho_g\rho_l}{\rho_l - \rho_g}\right)\frac{(T^* - T_{sat})}{T_{sat}} \tag{16.7-45}$$

The factor β is defined by means of the accomodation coefficient and the physical characteristics of the gas. β approaches 1.0 at near equilibrium conditions.

In the Eulerian and mixture multiphase models, the flow regime is assumed to be dispersed. If we assume that all vapor bubbles, for example, have the same diameter, then the interfacial area density is given by the following formula:

$$\frac{A_i}{V_{cell}} = \frac{6\alpha_v}{d} \tag{16.7-46}$$

where V_{cell} is the cell volume and the phase source term $(kg/s/m^3)$ should be of the form:

$$F\frac{A_i}{V_{cell}} = \frac{6}{d}\beta\sqrt{\frac{M}{2\pi R T_{sat}}}L\left(\frac{\rho_l}{\rho_l - \rho_g}\right)\left[\rho_g\alpha_v\frac{(T^* - T_{sat}}{T_{sat}}\right] \tag{16.7-47}$$

From the above equation, $coeff$, which is the inverse of the relaxation time (1/s) is defined as

$$coeff = \frac{6}{d}\beta\sqrt{\frac{M}{2\pi R T_{sat}}}L\left(\frac{\rho_l}{\rho_l - \rho_g}\right) \tag{16.7-48}$$

This leads to the final expression for the vaporization, defined in Equation 16.7-41. It can be treated implicitly as a source term in the phase conservation equation.

A similar expression can be obtained for condensation. In this case, we consider small droplets in a continuous vapor phase even if your primary phase is a liquid.

Note that the coefficient $coeff$ should theoretically be different for the condensation and evaporation expression. Furthermore, the theoretical expression is based on a few strong assumptions:

- flat interface
- dispersed regime with constant diameter
- known β

The bubble diameter and accommodation coefficient are usually not very well known, which is why the coefficient $coeff$ can be fine tuned to match experimental data. By default, the coefficient for both evaporation and condensation is 0.1.

16.8 Modeling Species Transport in Multiphase Flows

Species transport, as described in Chapter 7: Species Transport and Finite-Rate Chemistry, can also be applied to multiphase flows. You can choose to solve the conservation equations for chemical species in multiphase flows by having ANSYS FLUENT, for each phase k, predict the local mass fraction of each species, Y_i^k, through the solution of a convection-diffusion equation for the i^{th} species. The generalized chemical species conservation equation (Equation 7.1-1), when applied to a multiphase mixture can be represented in the following form:

$$\frac{\partial}{\partial t}(\rho^q\alpha^q Y_i^{\,q})+\nabla\cdot(\rho^q\alpha^q\vec{v}^q Y_i^{\,q}) = -\nabla\cdot\alpha^q\vec{J}_i^{\,q}+\alpha^q R_i^{\,q}+\alpha^q S_i^{\,q}+\sum_{p=1}^{n}(\dot{m}_{\mathrm{p^i q^j}}-\dot{m}_{\mathrm{q^j p^i}})+\mathcal{R} \quad (16.8\text{-}1)$$

where $R_i^{\,q}$ is the net rate of production of homogeneous species i by chemical reaction for phase q, $\dot{m}_{\mathrm{q^j p^i}}$ is the mass transfer source between species i and j from phase q to p, and $\mathcal{R}$ is the heterogeneous reaction rate. In addition, α^q is the volume fraction for phase q and $S_i^{\,q}$ is the rate of creation by addition from the dispersed phase plus any user-defined sources.

ANSYS FLUENT treats homogeneous gas phase chemical reactions the same as a single-phase chemical reaction. The reactants and the products belong to the same mixture material (set in the Species Model dialog box), and hence the same phase. The reaction rate is scaled by the volume fraction of the particular phase in the cell.

The set-up of a homogeneous gas phase chemical reaction in ANSYS FLUENT is the same as it is for a single phase. For more information, see Chapter 7: Species Transport and Finite-Rate Chemistry. For most multiphase species transport problems, boundary conditions for a particular species are set in the associated phase boundary condition dialog box (see Section 24.2.9: Defining Multiphase Cell Zone and Boundary Conditions in the separate User's Guide), and postprocessing and reporting of results is performed on a per-phase basis (see Section 24.8: Postprocessing for Multiphase Modeling in the separate User's Guide).

For multiphase species transport simulations, the Species Model dialog box allows you to include Volumetric, Wall Surface, and Particle Surface reactions. ANSYS FLUENT treats multiphase surface reactions as it would a single-phase reaction. The reaction rate is scaled with the volume fraction of the particular phase in the cell. For more information, see Chapter 7: Species Transport and Finite-Rate Chemistry.

ℹ To turn off reactions for a particular phase, while keeping the reactions active for other phases. turn on Volumetric under Reactions in the Species Model dialog box. Then, in the Create/Edit Materials dialog box, select none from the Reactions drop-down list.

The species of different phases is entirely independent. There is no implicit relationship between them even if they share the same name. Explicit relationships between species of different phases can be specified through mass transfer and heterogeneous reactions. For more information on mass transfer and heterogeneous reactions, see Section 24.2.8: Including Mass Transfer Effects and Section 24.2.7: Specifying Heterogeneous Reactions in the separate User's Guide, respectively.

Some phases may have a fluid material associated with them instead of a mixture material. The species equations are solved in those phases that are assigned a mixture material. The species equation above is solved for the mass fraction of the species in a particular phase. The mass transfer and heterogeneous reactions will be associated with the bulk fluid for phases with a single fluid material.

Additional information about modeling species transport is presented in the following subsections:

- Section 16.8.1: Limitations
- Section 16.8.2: Mass and Momentum Transfer with Multiphase Species Transport
- Section 16.8.3: The Stiff Chemistry Solver

16.8.1 Limitations

The following limitations exist for the modeling of species transport for multiphase flows:

- The nonpremixed, premixed, partially-premixed combustion, or the composition PDF transport species transport models are not available for multiphase species reactions.
- Only the laminar finite-rate, finite-rate/eddy-dissipation and eddy-dissipation turbulence-chemistry models of homogeneous reactions are available for multiphase species transport.
- The discrete phase model (DPM) is not compatible with multiphase species transport.

16.8.2 Mass and Momentum Transfer with Multiphase Species Transport

The ANSYS FLUENT multiphase mass transfer model accommodates mass transfer between species belonging to different phases. Instead of a matrix-type input, multiple mass transfer mechanisms need to be input. Each mass transfer mechanism defines the mass transfer phenomenon from one entity to another entity. An entity is either a particular species in a phase, or the bulk phase itself if the phase does not have a mixture material associated with it. The mass transfer phenomenon could be specified either through the inbuilt unidirectional "constant-rate" mass transfer (Section 16.7.2: Unidirectional Constant Rate Mass Transfer) or through user-defined functions.

ANSYS FLUENT loops through all the mass transfer mechanisms to compute the net mass source/sink of each species in each phase. The net mass source/sink of a species is used to compute species and mass source terms. ANSYS FLUENT will also automatically add the source contribution to all relevant momentum and energy equations based on that assumption that the momentum and energy carried along with the transferred mass. For other equations, the transport due to mass transfer needs to be explicitly modeled by the user.

Source Terms due to Heterogeneous Reactions

Consider the following reaction:

$$aA + bB \rightarrow cC + dD \tag{16.8-2}$$

Let as assume that A and C belong to phase 1 and B and D to phase 2.

Mass Transfer

Mass source for the phases are given by:

$$\begin{aligned} S_1 &= \mathcal{R}(cM_c - aM_a) & (16.8\text{-}3) \\ S_2 &= \mathcal{R}(dM_d - bM_b) & (16.8\text{-}4) \end{aligned}$$

where S is the mass source, M is the molecular weight, and $\mathcal{R}$ is the reaction rate.

The general expression for the mass source for the i^{th} phase is

$$\begin{aligned} S_{r_i} &= -\mathcal{R} \sum_{r_i} \gamma_j^r M_j^r & (16.8\text{-}5) \\ S_{p_i} &= \mathcal{R} \sum_{p_i} \gamma_j^p M_j^p & (16.8\text{-}6) \\ S_i &= S_{p_i} + S_{r_i} & (16.8\text{-}7) \end{aligned}$$

where γ is the stoichiometric coefficient, p represents the product, and r represents the reactant.

Momentum Transfer

Momentum transfer is more complicated, but we can assume that the reactants mix (conserving momentum) and the products take momentum in the ratio of the rate of their formation.

The net velocity, $\vec{u}_{net}$, of the reactants is given by:

$$\vec{u}_{net} = \frac{aM_a\vec{u}_1 + bM_b\vec{u}_2}{aM_a + bM_b} \tag{16.8-8}$$

The general expression for the net velocity of the reactants is given by:

$$\vec{u}_{net} = \frac{\sum_r \gamma_j^r M_j^r \vec{u}_{r_j}}{\sum_r \gamma_j^r M_j^r} \tag{16.8-9}$$

where j represents the j^{th} item (either a reactant or a product).

Momentum transfer for the phases is then given by:

$$S_1^{\vec{u}} = \mathcal{R}(cM_c\vec{u}_{net} - aM_a\vec{u}_1) \tag{16.8-10}$$

$$S_2^{\vec{u}} = \mathcal{R}(dM_d\vec{u}_{net} - bM_b\vec{u}_2) \tag{16.8-11}$$

The general expression is

$$S_i^{\vec{u}} = S_{p_i}\vec{u}_{net} - \mathcal{R}\sum_{r_i} \gamma_j^r M_j^r \vec{u}_i \tag{16.8-12}$$

If we assume that there is no momentum transfer, then the above term will be zero.

Species Transfer

The general expression for source for k^{th} species in the j^{th} phase is

$$S_{r_i^k} = -\mathcal{R}\sum_{r_i^k} \gamma_j^{r^k} M_j^{r^k} \tag{16.8-13}$$

$$S_{p_i^k} = \mathcal{R}\sum_{p_i^k} \gamma_j^{p^k} M_j^{p^k} \tag{16.8-14}$$

$$S_i^k = S_{p_i^k} + S_{r_i^k} \tag{16.8-15}$$

Heat Transfer

For heat transfer, we need to consider the formation enthalpies of the reactants and products as well:

The net enthalpy of the reactants is given by:

$$H_{net} = \frac{aM_a(H_a + h_a^f) + bM_b(H_b + h_b^f)}{aM_a + bM_b} \qquad (16.8\text{-}16)$$

where h^f represents the formation enthalpy, and H represents the enthalpy.

The general expression for H_{net} is:

$$H_{net} = \frac{\sum_r \gamma_j^r M_j^r (H_j^r + {h^f}_j^r)}{\sum_r \gamma_j^r M_j^r} \qquad (16.8\text{-}17)$$

If we assume that this enthalpy gets distributed to the products in the ratio of their mass production rates, heat transfer for the phases are given by:

$$\begin{aligned} S_1^H &= \mathcal{R}(cM_c H_{net} - aM_a H^a - cM_c h_f^c) \qquad (16.8\text{-}18) \\ S_2^H &= \mathcal{R}(dM_d H_{net} - bM_b H^b - dM_d h_f^d) \qquad (16.8\text{-}19) \end{aligned}$$

The last term in the above equations appears because our enthalpy is with reference to the formation enthalpy.

The general expression for the heat source is:

$$S_i^H = S_{p_i} H_{net} - \mathcal{R}\left(\sum_{r_i} \gamma_j^r M_j^r H_j^r + \sum_{p_i} \gamma_j^p M_j^p {h^f}_j^p\right) \qquad (16.8\text{-}20)$$

If we assume that there is no heat transfer, we can assume that the different species only carry their formation enthalpies with them. Thus the expression for H_{net} will be:

$$H_{net} = \frac{\sum_r \gamma_j^r M_j^r {h^f}_j^r}{\sum_r \gamma_j^r M_j^r} \qquad (16.8\text{-}21)$$

The expression S_i^H will be

$$S_i^H = S_{p_i} H_{net} - \mathcal{R} \sum_{p_i} \gamma_j^p M_j^p {h^f}_j^p \qquad (16.8\text{-}22)$$

16.8.3 The Stiff Chemistry Solver

ANSYS FLUENT has the option of solving *intra*phase and *inter*phase chemical reactions with a stiff chemistry solver. This option is only available for unsteady cases, where a fractional step scheme is applied. In the first fractional step, the multiphase species Equation 16.8-1 is solved spatially with the reaction term ${R_i}^q$ set to zero. In the second fractional step, the reaction term is integrated in every cell using a stiff ODE solver.

Chapter 17. Solidification and Melting

This chapter describes how you can model solidification and melting in ANSYS FLUENT. For information about using the model, see Chapter 25: Modeling Solidification and Melting in the separate User's Guide. Information about the theory behind the model is organized into the following sections:

17.1 Overview

ANSYS FLUENT can be used to solve fluid flow problems involving solidification and/or melting taking place at one temperature (e.g., in pure metals) or over a range of temperatures (e.g., in binary alloys). Instead of tracking the liquid-solid front explicitly, ANSYS FLUENT uses an enthalpy-porosity formulation. The liquid-solid mushy zone is treated as a porous zone with porosity equal to the liquid fraction, and appropriate momentum sink terms are added to the momentum equations to account for the pressure drop caused by the presence of solid material. Sinks are also added to the turbulence equations to account for reduced porosity in the solid regions.

ANSYS FLUENT provides the following capabilities for modeling solidification and melting:

- calculation of liquid-solid solidification/melting in pure metals as well as in binary alloys
- modeling of continuous casting processes (i.e., "pulling" of solid material out of the domain)
- modeling of the thermal contact resistance between solidified material and walls (e.g., due to the presence of an air gap)
- modeling of species transport with solidification/melting
- postprocessing of quantities related to solidification/melting (i.e., liquid fraction and pull velocities)

These modeling capabilities allow ANSYS FLUENT to simulate a wide range of solidification/melting problems, including melting, freezing, crystal growth, and continuous casting. The physical equations used for these calculations are described in Section 17.3: Introduction, and instructions for setting up and solving a solidification/melting problem are provided in Section 25: Modeling Solidification and Melting in the separate User's Guide.

17.2 Limitations

As mentioned in Section 17.1: Overview, the formulation in ANSYS FLUENT can be used to model the solidification/melting of pure materials, as well as alloys. ANSYS FLUENT offers two rules to determine the liquid fraction versus temperature relationship, namely the linear Lever rule and the non-linear Scheil rule.

The following limitations apply to the solidification/melting model in ANSYS FLUENT:

- The solidification/melting model can be used only with the pressure-based solver; it is not available with the density-based solvers.
- The solidification/melting model cannot be used for compressible flows.
- Of the general multiphase models (VOF, mixture, and Eulerian), only the VOF model can be used with the solidification/melting model.
- With the exception of species diffusivities, you cannot specify material properties separately for the solid and liquid materials.
- When using the solidification/melting model in conjunction with modeling species transport with reactions, there is no mechanism to restrict the reactions to only the liquid region; i.e., the reactions are solved everywhere.

17.3 Introduction

An enthalpy-porosity technique [358, 360, 361] is used in ANSYS FLUENT for modeling the solidification/melting process. In this technique, the melt interface is not tracked explicitly. Instead, a quantity called the liquid fraction, which indicates the fraction of the cell volume that is in liquid form, is associated with each cell in the domain. The liquid fraction is computed at each iteration, based on an enthalpy balance.

The mushy zone is a region in which the liquid fraction lies between 0 and 1. The mushy zone is modeled as a "pseudo" porous medium in which the porosity decreases from 1 to 0 as the material solidifies. When the material has fully solidified in a cell, the porosity becomes zero and hence the velocities also drop to zero.

In this section, an overview of the solidification/melting theory is given. Refer to Voller and Prakash [361] for details on the enthalpy-porosity method.

17.4 Energy Equation

The enthalpy of the material is computed as the sum of the sensible enthalpy, h, and the latent heat, ΔH:

$$H = h + \Delta H \tag{17.4-1}$$

where

$$h = h_{\rm ref} + \int_{T_{\rm ref}}^{T} c_p dT \tag{17.4-2}$$

and $h_{\rm ref}$ = reference enthalpy
$T_{\rm ref}$ = reference temperature
c_p = specific heat at constant pressure

The liquid fraction, β, can be defined as

$$\begin{aligned}
\beta &= 0 \quad \text{if} \quad T < T_{\rm solidus} \\
\beta &= 1 \quad \text{if} \quad T > T_{\rm liquidus} \\
\beta &= \frac{T - T_{\rm solidus}}{T_{\rm liquidus} - T_{\rm solidus}} \quad \text{if} \quad T_{\rm solidus} < T < T_{\rm liquidus}
\end{aligned} \tag{17.4-3}$$

The latent heat content can now be written in terms of the latent heat of the material, L:

$$\Delta H = \beta L \tag{17.4-4}$$

The latent heat content can vary between zero (for a solid) and L (for a liquid).

For solidification/melting problems, the energy equation is written as

$$\frac{\partial}{\partial t}(\rho H) + \nabla \cdot (\rho \vec{v} H) = \nabla \cdot (k \nabla T) + S \qquad (17.4\text{-}5)$$

where H = enthalpy (see Equation 17.4-1)
ρ = density
$\vec{v}$ = fluid velocity
S = source term

The solution for temperature is essentially an iteration between the energy equation (Equation 17.4-5) and the liquid fraction equation (Equation 17.4-3). Directly using Equation 17.4-3 to update the liquid fraction usually results in poor convergence of the energy equation. In ANSYS FLUENT, the method suggested by Voller and Swaminathan [362] is used to update the liquid fraction. For pure metals, where T_{solidus} and T_{liquidus} are equal, a method based on specific heat, given by Voller and Prakash [361], is used instead.

17.5 Momentum Equations

The enthalpy-porosity technique treats the mushy region (partially solidified region) as a porous medium. The porosity in each cell is set equal to the liquid fraction in that cell. In fully solidified regions, the porosity is equal to zero, which extinguishes the velocities in these regions. The momentum sink due to the reduced porosity in the mushy zone takes the following form:

$$S = \frac{(1-\beta)^2}{(\beta^3 + \epsilon)} A_{\text{mush}} (\vec{v} - \vec{v}_p) \qquad (17.5\text{-}1)$$

where β is the liquid volume fraction, ϵ is a small number (0.001) to prevent division by zero, A_{mush} is the mushy zone constant, and $\vec{v}_p$ is the solid velocity due to the pulling of solidified material out of the domain (also referred to as the pull velocity).

The mushy zone constant measures the amplitude of the damping; the higher this value, the steeper the transition of the velocity of the material to zero as it solidifies. Very large values may cause the solution to oscillate.

The pull velocity is included to account for the movement of the solidified material as it is continuously withdrawn from the domain in continuous casting processes. The presence of this term in Equation 17.5-1 allows newly solidified material to move at the pull velocity. If solidified material is not being pulled from the domain, $\vec{v}_p = 0$. More details about the pull velocity are provided in Section 17.8: Pull Velocity for Continuous Casting.

17.6 Turbulence Equations

Sinks are added to all of the turbulence equations in the mushy and solidified zones to account for the presence of solid matter. The sink term is very similar to the momentum sink term (Equation 17.5-1):

$$S = \frac{(1-\beta)^2}{(\beta^3+\epsilon)} A_{\rm mush} \phi \tag{17.6-1}$$

where ϕ represents the turbulence quantity being solved (k, ϵ, ω, etc.), and the mushy zone constant, $A_{\rm mush}$, is the same as the one used in Equation 17.5-1.

17.7 Species Equations

For solidification and melting of a pure substance, phase change occurs at a distinct melting temperature, $T_{\rm melt}$. For a multicomponent mixture, however, a mushy freeze/melt zone exists between a lower solidus and an upper liquidus temperature. When a multicomponent liquid solidifies, solutes diffuse from the solid phase into the liquid phase. This effect is quantified by the partition coefficient of solute i, denoted K_i, which is the ratio of the mass fraction in the solid to that in the liquid at the interface.

ANSYS FLUENT computes the solidus and liquidus temperatures in a species mixture as,

$$T_{\rm solidus} = T_{\rm melt} + \sum_{\rm solutes} m_i Y_i / K_i \tag{17.7-1}$$

$$T_{\rm liquidus} = T_{\rm melt} + \sum_{\rm solutes} m_i Y_i \tag{17.7-2}$$

where K_i is the partition coefficient of solute i, Y_i is the mass fraction of solute i, and m_i is the slope of the liquidus surface with respect to Y_i. It is assumed that the last species material of the mixture is the solvent and that the other species are the solutes.

The liquidus slope of species i, m_i, is calculated from the Eutectic temperature and the Eutectic mass fraction as,

$$m_i = \frac{T_{\rm Eut} - T_{\rm melt}}{Y_{i,{\rm Eut}}} \tag{17.7-3}$$

Updating the liquid fraction via Equation 17.4-3 can cause numerical errors and convergence difficulties in multicomponent mixtures. Instead, the liquid fraction is updated as,

$$\beta^{n+1} = \beta^n - \lambda \frac{a_p \left(T - T^*\right) \Delta t}{\rho V L - a_p \Delta t L \frac{\partial T^*}{\partial \beta}} \tag{17.7-4}$$

where the superscript n indicates the iteration number, λ is a relaxation factor with a default value if 0.9, a_p is the cell matrix co-efficient, Δt is the time-step, ρ is the current density, V is the cell volume, T is the current cell temperature and T^* is the interface temperature.

ANSYS FLUENT offers two models for species segregation at the micro-scale, namely the Lever and Scheil rules. The former assumes infinite diffusion of the solute species in the solid, and the latter assumes zero diffusion. For the Lever rule, the interface temperature, T^*, is calculated for a binary mixture as,

$$T^* = \frac{T_{\rm liquidus} - T_{\rm melt}(1-\beta)(1-P)}{1 - T_{\rm melt}(1-\beta)(1-P)} \tag{17.7-5}$$

where

$$P = \frac{T_{\rm melt} - T_{\rm liquidus}}{T_{\rm melt} - T_{\rm solidus}} \tag{17.7-6}$$

The Scheil rule evaluates T^* as,

$$T^* = T_{\rm melt} - (T_{\rm melt} - T_{\rm liquidus})\beta^{(P-1)} \tag{17.7-7}$$

For the Lever rule, species transport equations are solved for the total mass fraction of species i, Y_i:

$$\frac{\partial}{\partial t}(\rho Y_i) + \nabla \cdot (\rho\,[\beta \vec{v}_{\rm liq} Y_{i,\rm liq} + (1-\beta)\vec{v}_p Y_{i,\rm sol}]) = -\nabla \cdot \vec{J}_i + R_i \tag{17.7-8}$$

where R_i is the reaction rate and $\vec{J}_i$ is given by

$$\vec{J}_i = -\rho[\beta D_{i,m,\rm liq} \nabla Y_{i,\rm liq} + (1-\beta) D_{i,m,\rm sol} \nabla Y_{i,\rm sol}] \tag{17.7-9}$$

$\vec{v}_{\rm liq}$ is the velocity of the liquid and $\vec{v}_p$ is the solid (pull) velocity. $\vec{v}_p$ is set to zero if pull velocities are not included in the solution. The liquid velocity can be found from the average velocity (as determined by the flow equation) as

$$\vec{v}_{\rm liq} = \frac{(\vec{v} - \vec{v}_p(1-\beta))}{\beta} \tag{17.7-10}$$

The liquid ($Y_{i,\rm liq}$) and solid ($Y_{i,\rm sol}$) mass fractions are related to each other by the partition coefficient K_i:

$$Y_{i,\rm sol} = K_i Y_{i,\rm liq} \tag{17.7-11}$$

When the Scheil model is selected, ANSYS FLUENT solves for $Y_{i,\text{liq}}$ as the dependent variable [359]:

$$\frac{\partial}{\partial t}(\rho Y_{i,\text{liq}}) + \nabla \cdot (\rho\, [\beta \vec{v}_{\text{liq}} Y_{i,\text{liq}} + (1-\beta)\vec{v}_p Y_{i,\text{sol}}]) = R_i + \nabla \cdot (\rho \beta D_{i,m,\text{liq}} \nabla Y_{i,\text{liq}}) - K_i Y_{i,\text{liq}} \frac{\partial}{\partial t}(\rho(1-\beta)) + \frac{\partial}{\partial t}(\rho(1-\beta) Y_{i,\text{liq}}) \tag{17.7-12}$$

17.8 Pull Velocity for Continuous Casting

In continuous casting processes, the solidified matter is usually continuously pulled out from the computational domain, as shown in Figure 17.8.1. Consequently, the solid material will have a finite velocity that needs to be accounted for in the enthalpy-porosity technique.

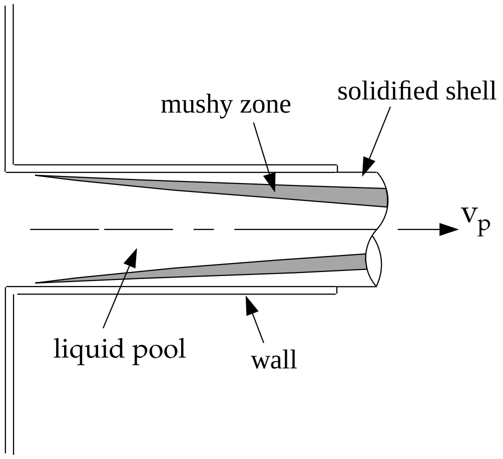


Figure 17.8.1: “Pulling” a Solid in Continuous Casting

As mentioned in Section 17.5: Momentum Equations, the enthalpy-porosity approach treats the solid-liquid mushy zone as a porous medium with porosity equal to the liquid fraction. A suitable sink term is added in the momentum equation to account for the pressure drop due to the porous structure of the mushy zone. For continuous casting applications, the relative velocity between the molten liquid and the solid is used in the momentum sink term (Equation 17.5-1) rather than the absolute velocity of the liquid.

The exact computation of the pull velocity for the solid material is dependent on the Young’s modulus and Poisson’s ratio of the solid and the forces acting on it. ANSYS FLUENT uses a Laplacian equation to approximate the pull velocities in the solid region based on the velocities at the boundaries of the solidified region:

$$\nabla^2 \vec{v}_p = 0 \tag{17.8-1}$$

ANSYS FLUENT uses the following boundary conditions when computing the pull velocities:

- At a velocity inlet, a stationary wall, or a moving wall, the specified velocity is used.
- At all other boundaries (including the liquid-solid interface between the liquid and solidified material), a zero-gradient velocity is used.

The pull velocities are computed only in the solid region.

Note that ANSYS FLUENT can also use a specified constant value or custom field function for the pull velocity, instead of computing it. See Section 25.2: Procedures for Modeling Continuous Casting in the separate User's Guide for details.

17.9 Contact Resistance at Walls

ANSYS FLUENT's solidification/melting model can account for the presence of an air gap between the walls and the solidified material, using an additional heat transfer resistance between walls and cells with liquid fractions less than 1. This contact resistance is accounted for by modifying the conductivity of the fluid near the wall. Thus, the wall heat flux, as shown in Figure 17.9.1, is written as

$$q = \frac{(T - T_w)}{(l/k + R_c(1 - \beta))} \tag{17.9-1}$$

where T, T_w, and l are defined in Figure 17.9.1, k is the thermal conductivity of the fluid, β is the liquid volume fraction, and R_c is the contact resistance, which has the same units as the inverse of the heat transfer coefficient.

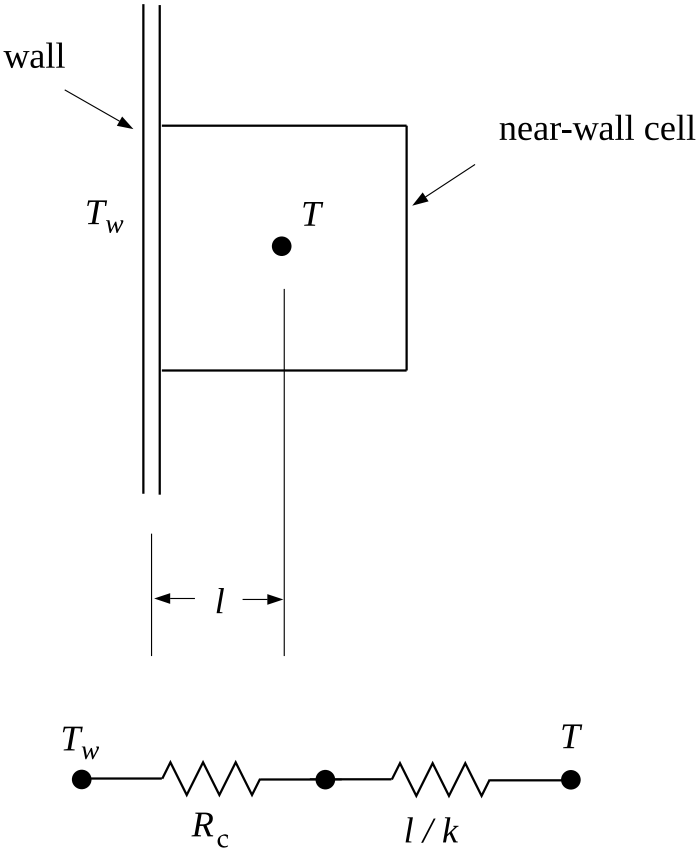


Figure 17.9.1: Circuit for Contact Resistance

Chapter 18. Solver Theory

This chapter describes the ANSYS FLUENT solver theory. Details about the solver algorithms used by ANSYS FLUENT are provided in Sections 18.1–18.6. For more information about using the solver, see Chapter 26: Using the Solver in the separate User's Guide.

18.1 Overview of Flow Solvers

ANSYS FLUENT allows you to choose one of the two numerical methods:

- pressure-based solver (see Section 18.1.1: Pressure-Based Solver)
- density-based solver (see Section 18.1.2: Density-Based Solver)

Historically speaking, the pressure-based approach was developed for low-speed incompressible flows, while the density-based approach was mainly used for high-speed compressible flows. However, recently both methods have been extended and reformulated to solve and operate for a wide range of flow conditions beyond their traditional or original intent.

In both methods the velocity field is obtained from the momentum equations. In the density-based approach, the continuity equation is used to obtain the density field while the pressure field is determined from the equation of state.

On the other hand, in the pressure-based approach, the pressure field is extracted by solving a pressure or pressure correction equation which is obtained by manipulating continuity and momentum equations.

Using either method, ANSYS FLUENT will solve the governing integral equations for the conservation of mass and momentum, and (when appropriate) for energy and other scalars such as turbulence and chemical species. In both cases a control-volume-based technique is used that consists of:

- Division of the domain into discrete control volumes using a computational grid.
- Integration of the governing equations on the individual control volumes to construct algebraic equations for the discrete dependent variables ("unknowns") such as velocities, pressure, temperature, and conserved scalars.
- Linearization of the discretized equations and solution of the resultant linear equation system to yield updated values of the dependent variables.

The two numerical methods employ a similar discretization process (finite-volume), but the approach used to linearize and solve the discretized equations is different.

The general solution methods are described in Sections 18.1.1 and 18.1.2. To learn how to apply the solvers, see Section 26.1.1: Choosing the Solver in the separate User's Guide.

18.1.1 Pressure-Based Solver

The pressure-based solver employs an algorithm which belongs to a general class of methods called the projection method [54]. In the projection method, wherein the constraint of mass conservation (continuity) of the velocity field is achieved by solving a pressure (or pressure correction) equation. The pressure equation is derived from the continuity and the momentum equations in such a way that the velocity field, corrected by the pressure, satisfies the continuity. Since the governing equations are nonlinear and coupled to one another, the solution process involves iterations wherein the entire set of governing equations is solved repeatedly until the solution converges.

Two pressure-based solver algorithms are available in ANSYS FLUENT. A segregated algorithm, and a coupled algorithm. These two approaches are discussed in the sections below.

The Pressure-Based Segregated Algorithm

The pressure-based solver uses a solution algorithm where the governing equations are solved sequentially (i.e., segregated from one another). Because the governing equations are non-linear and coupled, the solution loop must be carried out iteratively in order to obtain a converged numerical solution.

In the segregated algorithm, the individual governing equations for the solution variables (e.g., u, v, w, p, T, k, ϵ, etc.) are solved one after another . Each governing equation, while being solved, is "decoupled" or "segregated" from other equations, hence its name. The segregated algorithm is memory-efficient, since the discretized equations need only be stored in the memory one at a time. However, the solution convergence is relatively slow, inasmuch as the equations are solved in a decoupled manner.

With the segregated algorithm, each iteration consists of the steps illustrated in Figure 18.1.1 and outlined below:

1. Update fluid properties (e,g, density, viscosity, specific heat) including turbulent viscosity (diffusivity) based on the current solution.

2. Solve the momentum equations, one after another, using the recently updated values of pressure and face mass fluxes.

3. Solve the pressure correction equation using the recently obtained velocity field and the mass-flux.

4. Correct face mass fluxes, pressure, and the velocity field using the pressure correction obtained from Step 3.

5. Solve the equations for additional scalars, if any, such as turbulent quantities, energy, species, and radiation intensity using the current values of the solution variables.

6. Update the source terms arising from the interactions among different phases (e.g., source term for the carrier phase due to discrete particles).

7. Check for the convergence of the equations.

These steps are continued until the convergence criteria are met.

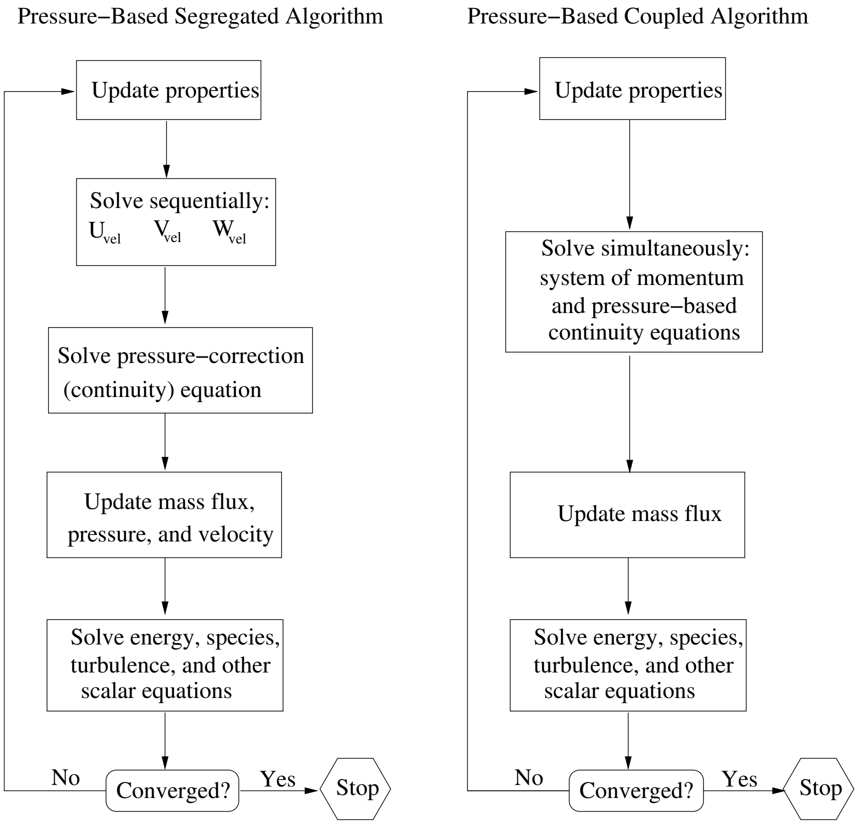

Pressure−Based Segregated Algorithm
Update properties
Solve sequentially:
U_{vel} V_{vel} W_{vel}
Solve pressure−correction (continuity) equation
Update mass flux, pressure, and velocity
Solve energy, species, turbulence, and other scalar equations
No
Converged?
Yes
Stop
Pressure−Based Coupled Algorithm
Update properties
Solve simultaneously: system of momentum and pressure−based continuity equations
Update mass flux
Solve energy, species, turbulence, and other scalar equations
No
Converged?
Yes
Stop

Figure 18.1.1: Overview of the Pressure-Based Solution Methods

The Pressure-Based Coupled Algorithm

Unlike the segregated algorithm described above, the pressure-based coupled algorithm solves a coupled system of equations comprising the momentum equations and the pressure-based continuity equation. Thus, in the coupled algorithm, Steps 2 and 3 in the segregated solution algorithm are replaced by a single step in which the coupled system of equations are solved. The remaining equations are solved in a decoupled fashion as in the segregated algorithm.

Since the momentum and continuity equations are solved in a closely coupled manner, the rate of solution convergence significantly improves when compared to the segregated algorithm. However, the memory requirement increases by 1.5 – 2 times that of the segregated algorithm since the discrete system of all momentum and pressure-based continuity equations needs to be stored in the memory when solving for the velocity and pressure fields (rather than just a single equation, as is the case with the segregated algorithm).

18.1.2 Density-Based Solver

The density-based solver solves the governing equations of continuity, momentum, and (where appropriate) energy and species transport simultaneously (i.e., coupled together). Governing equations for additional scalars will be solved afterward and sequentially (i.e., segregated from one another and from the coupled set) using the procedure described in Section 18.2: General Scalar Transport Equation: Discretization and Solution. Because the governing equations are non-linear (and coupled), several iterations of the solution loop must be performed before a converged solution is obtained. Each iteration consists of the steps illustrated in Figure 18.1.2 and outlined below:

1. Update the fluid properties based on the current solution. (If the calculation has just begun, the fluid properties will be updated based on the initialized solution.)

2. Solve the continuity, momentum, and (where appropriate) energy and species equations simultaneously.

3. Where appropriate, solve equations for scalars such as turbulence and radiation using the previously updated values of the other variables.

4. When interphase coupling is to be included, update the source terms in the appropriate continuous phase equations with a discrete phase trajectory calculation.

5. Check for convergence of the equation set.

These steps are continued until the convergence criteria are met.

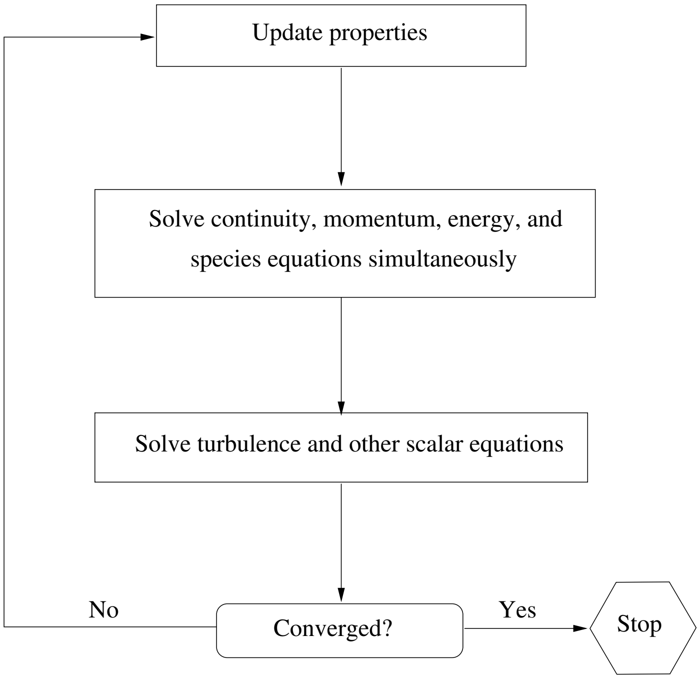

Update properties
Solve continuity, momentum, energy, and species equations simultaneously
Solve turbulence and other scalar equations
Converged?
No
Yes
Stop

Figure 18.1.2: Overview of the Density-Based Solution Method

In the density-based solution method, you can solve the coupled system of equations (continuity, momentum, energy and species equations if available) using, either the coupled-explicit formulation or the coupled-implicit formulation. The main distinction between the density-based explicit and implicit formulations is described next.

In the density-based solution methods, the discrete, non-linear governing equations are linearized to produce a system of equations for the dependent variables in every computational cell. The resultant linear system is then solved to yield an updated flow-field solution.

The manner in which the governing equations are linearized may take an “implicit” or “explicit” form with respect to the dependent variable (or set of variables) of interest. By implicit or explicit we mean the following:

- implicit: For a given variable, the unknown value in each cell is computed using a relation that includes both existing and unknown values from neighboring cells. Therefore each unknown will appear in more than one equation in the system, and these equations must be solved simultaneously to give the unknown quantities.

- explicit: For a given variable, the unknown value in each cell is computed using a relation that includes only existing values. Therefore each unknown will appear in only one equation in the system and the equations for the unknown value in each cell can be solved one at a time to give the unknown quantities.

In the density-based solution method you have a choice of using either an implicit or explicit linearization of the governing equations. This choice applies only to the coupled set of governing equations. Transport equations for additional scalars are solved segregated from the coupled set (such as turbulence, radiation, etc.). The transport equations are linearized and solved implicitly using the method described in section Section 18.2: General Scalar Transport Equation: Discretization and Solution. Regardless of whether you choose the implicit or explicit methods, the solution procedure shown in Figure 18.1.2 is followed.

If you choose the implicit option of the density-based solver, each equation in the coupled set of governing equations is linearized implicitly with respect to all dependent variables in the set. This will result in a system of linear equations with N equations for each cell in the domain, where N is the number of coupled equations in the set. Because there are N equations per cell, this is sometimes called a “block” system of equations.

A point implicit linear equation solver (Incomplete Lower Upper (ILU) factorization scheme or a symmetric block Gauss-Seidel) is used in conjunction with an algebraic multigrid (AMG) method to solve the resultant block system of equations for all N dependent variables in each cell. For example, linearization of the coupled continuity, x-, y-, z-momentum, and energy equation set will produce a system of equations in which p, u, v, w, and T are the unknowns. Simultaneous solution of this equation system (using the block AMG solver) yields at once updated pressure, u-, v-, w-velocity, and temperature fields.

In summary, the coupled implicit approach solves for all variables (p, u, v, w, T) in all cells at the same time.

If you choose the explicit option of the density-based solver, each equation in the coupled set of governing equations is linearized explicitly. As in the implicit option, this too will result in a system of equations with N equations for each cell in the domain and likewise, all dependent variables in the set will be updated at once. However, this system of equations is explicit in the unknown dependent variables. For example, the x-momentum equation is written such that the updated x velocity is a function of existing values of the field variables. Because of this, a linear equation solver is not needed. Instead, the solution is updated using a multi-stage (Runge-Kutta) solver. Here you have the additional option of employing a full approximation storage (FAS) multigrid scheme to accelerate the multi-stage solver.

In summary, the density-based explicit approach solves for all variables (p, u, v, w, T) one cell at a time.

Note that the FAS multigrid is an optional component of the explicit approach, while the AMG is a required element in both the pressure-based and density-based implicit approaches.

18.2 General Scalar Transport Equation: Discretization and Solution

ANSYS FLUENT uses a control-volume-based technique to convert a general scalar transport equation to an algebraic equation that can be solved numerically. This control volume technique consists of integrating the transport equation about each control volume, yielding a discrete equation that expresses the conservation law on a control-volume basis.

Discretization of the governing equations can be illustrated most easily by considering the unsteady conservation equation for transport of a scalar quantity ϕ. This is demonstrated by the following equation written in integral form for an arbitrary control volume V as follows:

$$\int_V \frac{\partial \rho\phi}{\partial t}\, dV + \oint \rho\phi\, \vec{v}\cdot d\vec{A} = \oint \Gamma_\phi\, \nabla\phi \cdot d\vec{A} + \int_V S_\phi\, dV \qquad (18.2\text{-}1)$$

where

ρ = density
$\vec{v}$ = velocity vector ($= u\,\hat{\imath} + v\,\hat{\jmath}$ in 2D)
$\vec{A}$ = surface area vector
Γ_ϕ = diffusion coefficient for ϕ
$\nabla\phi$ = gradient of ϕ ($= (\partial\phi/\partial x)\,\hat{\imath} + (\partial\phi/\partial y)\,\hat{\jmath}$ in 2D)
S_ϕ = source of ϕ per unit volume

Equation 18.2-1 is applied to each control volume, or cell, in the computational domain. The two-dimensional, triangular cell shown in Figure 18.2.1 is an example of such a control volume. Discretization of Equation 18.2-1 on a given cell yields

$$\frac{\partial \rho\phi}{\partial t} V + \sum_f^{N_{\text{faces}}} \rho_f \vec{v}_f \phi_f \cdot \vec{A}_f = \sum_f^{N_{\text{faces}}} \Gamma_\phi\, \nabla\phi_f \cdot \vec{A}_f + S_\phi V \qquad (18.2\text{-}2)$$

where

N_{faces} = number of faces enclosing cell
ϕ_f = value of ϕ convected through face f
$\rho_f \vec{v}_f \cdot \vec{A}_f$ = mass flux through the face
$\vec{A}_f$ = area of face f, $|A|$ ($= |A_x\hat{\imath} + A_y\hat{\jmath}|$ in 2D)
$\nabla\phi_f$ = gradient of ϕ at face f
V = cell volume

Where $\frac{\partial\rho\phi}{\partial t} V$ is defined in Section 18.3.2: Temporal Discretization. The equations solved by ANSYS FLUENT take the same general form as the one given above and apply readily to multi-dimensional, unstructured meshes composed of arbitrary polyhedra.

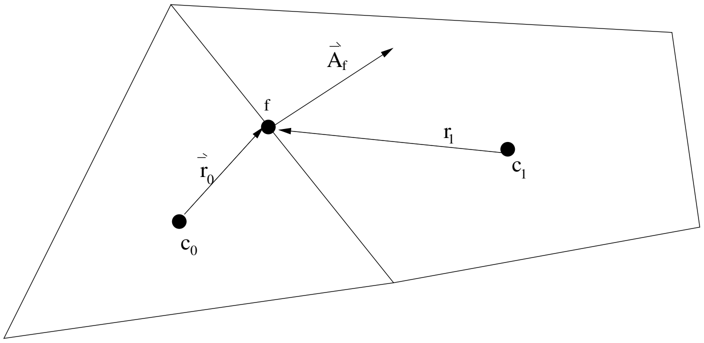


Figure 18.2.1: Control Volume Used to Illustrate Discretization of a Scalar Transport Equation

18.2.1 Solving the Linear System

The discretized scalar transport equation (Equation 18.2-2) contains the unknown scalar variable ϕ at the cell center as well as the unknown values in surrounding neighbor cells. This equation will, in general, be non-linear with respect to these variables. A linearized form of Equation 18.2-2 can be written as

$$a_P\,\phi = \sum_{\rm nb} a_{\rm nb}\phi_{\rm nb} + b \tag{18.2-3}$$

where the subscript *nb* refers to neighbor cells, and a_P and $a_{\rm nb}$ are the linearized coefficients for ϕ and $\phi_{\rm nb}$.

The number of neighbors for each cell depends on the mesh topology, but will typically equal the number of faces enclosing the cell (boundary cells being the exception).

Similar equations can be written for each cell in the mesh. This results in a set of algebraic equations with a sparse coefficient matrix. For scalar equations, ANSYS FLUENT solves this linear system using a point implicit (Gauss-Seidel) linear equation solver in conjunction with an algebraic multigrid (AMG) method which is described in Section 18.6.3: Algebraic Multigrid (AMG).

18.3 Discretization

Information is organized into the following subsections:

- Section 18.3.1: Spatial Discretization
- Section 18.3.2: Temporal Discretization
- Section 18.3.3: Evaluation of Gradients and Derivatives
- Section 18.3.4: Gradient Limiters

18.3.1 Spatial Discretization

By default, ANSYS FLUENT stores discrete values of the scalar ϕ at the cell centers (c_0 and c_1 in Figure 18.2.1). However, face values ϕ_f are required for the convection terms in Equation 18.2-2 and must be interpolated from the cell center values. This is accomplished using an upwind scheme.

Upwinding means that the face value ϕ_f is derived from quantities in the cell upstream, or “upwind,” relative to the direction of the normal velocity v_n in Equation 18.2-2. ANSYS FLUENT allows you to choose from several upwind schemes: first-order upwind, second-order upwind, power law, and QUICK. These schemes are described in Sections 18.3.1–18.3.1.

The diffusion terms in Equation 18.2-2 are central-differenced and are always second-order accurate.

For information on how to use the various spatial discretization schemes, see Section 26.2: Choosing the Spatial Discretization Scheme in the separate User’s Guide.

First-Order Upwind Scheme

When first-order accuracy is desired, quantities at cell faces are determined by assuming that the cell-center values of any field variable represent a cell-average value and hold throughout the entire cell; the face quantities are identical to the cell quantities. Thus when first-order upwinding is selected, the face value ϕ_f is set equal to the cell-center value of ϕ in the upstream cell.

> *i* First-order upwind is available in the pressure-based and density-based solvers.

Power-Law Scheme

The power-law discretization scheme interpolates the face value of a variable, ϕ, using the exact solution to a one-dimensional convection-diffusion equation

$$\frac{\partial}{\partial x}(\rho u \phi) = \frac{\partial}{\partial x}\Gamma\frac{\partial \phi}{\partial x} \tag{18.3-1}$$

where Γ and ρu are constant across the interval ∂x. Equation 18.3-1 can be integrated to yield the following solution describing how ϕ varies with x:

$$\frac{\phi(x) - \phi_0}{\phi_L - \phi_0} = \frac{\exp(\text{Pe}\frac{x}{L}) - 1}{\exp(\text{Pe}) - 1} \tag{18.3-2}$$

where

$$\begin{aligned} \phi_0 &= \phi|_{x=0} \\ \phi_L &= \phi|_{x=L} \end{aligned}$$

and Pe is the Peclet number:

$$\text{Pe} = \frac{\rho u L}{\Gamma} \tag{18.3-3}$$

The variation of $\phi(x)$ between $x = 0$ and $x = L$ is depicted in Figure 18.3.1 for a range of values of the Peclet number. Figure 18.3.1 shows that for large Pe, the value of ϕ at $x = L/2$ is approximately equal to the upstream value. This implies that when the flow is dominated by convection, interpolation can be accomplished by simply letting the face value of a variable be set equal to its "upwind" or upstream value. This is the standard first-order scheme for ANSYS FLUENT.

If the power-law scheme is selected, ANSYS FLUENT uses Equation 18.3-2 in an equivalent "power law" format [264], as its interpolation scheme.

As discussed in Section 18.3.1: First-Order Upwind Scheme, Figure 18.3.1 shows that for large Pe, the value of ϕ at $x = L/2$ is approximately equal to the upstream value. When Pe=0 (no flow, or pure diffusion), Figure 18.3.1 shows that ϕ may be interpolated using a simple linear average between the values at $x = 0$ and $x = L$. When the Peclet number has an intermediate value, the interpolated value for ϕ at $x = L/2$ must be derived by applying the "power law" equivalent of Equation 18.3-2.

i The power-law scheme is available in the pressure-based solver and when solving additional scalar equations in the density-based solver.

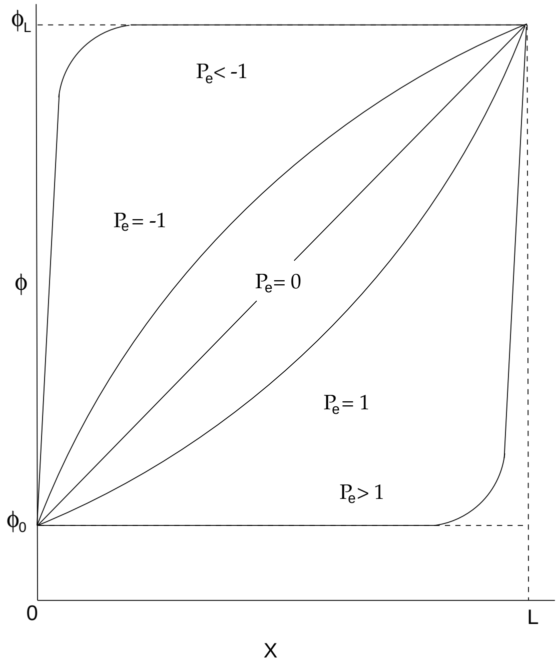


Figure 18.3.1: Variation of a Variable ϕ Between $x = 0$ and $x = L$ (Equation 18.3-1)

Second-Order Upwind Scheme

When second-order accuracy is desired, quantities at cell faces are computed using a multidimensional linear reconstruction approach [14]. In this approach, higher-order accuracy is achieved at cell faces through a Taylor series expansion of the cell-centered solution about the cell centroid. Thus when second-order upwinding is selected, the face value ϕ_f is computed using the following expression:

$$\phi_{f,SOU} = \phi + \nabla\phi \cdot \vec{r} \qquad (18.3\text{-}4)$$

where ϕ and $\nabla\phi$ are the cell-centered value and its gradient in the upstream cell, and $\vec{r}$ is the displacement vector from the upstream cell centroid to the face centroid. This formulation requires the determination of the gradient $\nabla\phi$ in each cell, as discussed in Section 18.3.3: Evaluation of Gradients and Derivatives. Finally, the gradient $\nabla\phi$ is limited so that no new maxima or minima are introduced.

i Second-order upwind is available in the pressure-based and density-based solvers.

First-to-Higher Order Blending

In some instances, and at certain flow conditions, a converged solution to steady-state may not be possible with the use of higher-order discretization schemes due to local flow fluctuations (physical or numerical). On the other hand, a converged solution for the same flow conditions maybe possible with a first-order discretization scheme. For this type of flow and situation, if a better than first-order accurate solution is desired, then first-to-higher-order blending can be used to obtain a converged steady-state solution.

The first-order to higher-order blending is applicable only when higher-order discretization is used. It is applicable with the following discretization schemes: second-order upwinding, central-differencing schemes, QUICK, and third-order MUSCL. The blending is not applicable to first-order, power-law, modified HRIC schemes, or the Geo-reconstruct and CICSAM schemes.

In the density-based solver, the blending is applied as a scaling factor to the reconstruction gradients. While in the pressure-based solver, the blending is applied to the higher-order terms for the convective transport variable.

To learn how to apply this option, refer to Section 26.2.1: First-to-Higher Order Blending in the separate User's Guide.

Central-Differencing Scheme

A second-order-accurate central-differencing discretization scheme is available for the momentum equations when you are using the LES turbulence model. This scheme provides improved accuracy for LES calculations.

The central-differencing scheme calculates the face value for a variable (ϕ_f) as follows:

$$\phi_{f,\mathrm{CD}} = \frac{1}{2}\left(\phi_0 + \phi_1\right) + \frac{1}{2}\left(\nabla\phi_0 \cdot \vec{r}_0 + \nabla\phi_1 \cdot \vec{r}_1\right) \qquad (18.3\text{-}5)$$

where the indices 0 and 1 refer to the cells that share face f, $\nabla\phi_{r,0}$ and $\nabla\phi_{r,1}$ are the reconstructed gradients at cells 0 and 1, respectively, and $\vec{r}$ is the vector directed from the cell centroid toward the face centroid.

It is well known that central-differencing schemes can produce unbounded solutions and non-physical wiggles, which can lead to stability problems for the numerical procedure. These stability problems can often be avoided if a deferred approach is used for the central-differencing scheme. In this approach, the face value is calculated as follows:

$$\phi_f = \underbrace{\phi_{f,\mathrm{UP}}}_{\text{implicit part}} + \underbrace{\left(\phi_{f,\mathrm{CD}} - \phi_{f,\mathrm{UP}}\right)}_{\text{explicit part}} \qquad (18.3\text{-}6)$$

where UP stands for upwind. As indicated, the upwind part is treated implicitly while the difference between the central-difference and upwind values is treated explicitly. Provided that the numerical solution converges, this approach leads to pure second-order differencing.

The central differencing scheme is available only in the pressure-based solver.

Bounded Central Differencing Scheme

The central differencing scheme described in Section 18.3.1: Central-Differencing Scheme is an ideal choice for LES in view of its meritoriously low numerical diffusion. However, it often leads to unphysical oscillations in the solution fields. In LES, the situation is exacerbated by usually very low subgrid-scale turbulent diffusivity. The bounded central differencing scheme is essentially based on the normalized variable diagram (NVD) approach [187] together with the convection boundedness criterion (CBC). The bounded central differencing scheme is a composite NVD-scheme that consists of a pure central differencing, a blended scheme of the central differencing and the second-order upwind scheme, and the first-order upwind scheme. It should be noted that the first-order scheme is used only when the CBC is violated.

The bounded central differencing scheme is the default convection scheme for LES. When you select LES, the convection discretization schemes for all transport equations are automatically switched to the bounded central differencing scheme.

The bounded central differencing scheme is available only in the pressure-based solver.

QUICK Scheme

For quadrilateral and hexahedral meshes, where unique upstream and downstream faces and cells can be identified, ANSYS FLUENT also provides the QUICK scheme for computing a higher-order value of the convected variable ϕ at a face. QUICK-type schemes [188] are based on a weighted average of second-order-upwind and central interpolations of the variable. For the face e in Figure 18.3.2, if the flow is from left to right, such a value can be written as

$$\phi_e = \theta \left[\frac{S_d}{S_c + S_d} \phi_P + \frac{S_c}{S_c + S_d} \phi_E \right] + (1 - \theta) \left[\frac{S_u + 2S_c}{S_u + S_c} \phi_P - \frac{S_c}{S_u + S_c} \phi_W \right] \qquad (18.3\text{-}7)$$

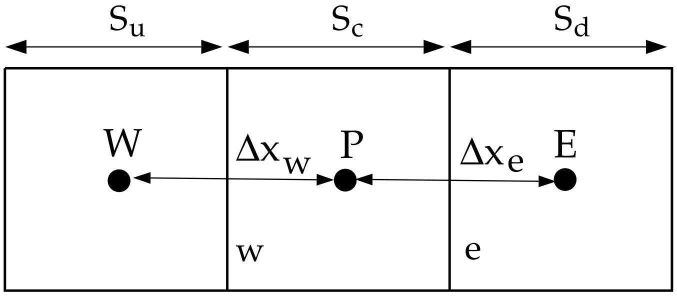


Figure 18.3.2: One-Dimensional Control Volume

$\theta = 1$ in the above equation results in a central second-order interpolation while $\theta = 0$ yields a second-order upwind value. The traditional QUICK scheme is obtained by setting $\theta = 1/8$. The implementation in ANSYS FLUENT uses a variable, solution-dependent value of θ, chosen so as to avoid introducing new solution extrema.

The QUICK scheme will typically be more accurate on structured meshes aligned with the flow direction. Note that ANSYS FLUENT allows the use of the QUICK scheme for unstructured or hybrid meshes as well; in such cases the usual second-order upwind discretization scheme (described in Section 18.3.1: Second-Order Upwind Scheme) will be used at the faces of non-hexahedral (or non-quadrilateral, in 2D) cells. The second-order upwind scheme will also be used at partition boundaries when the parallel solver is used.

i The QUICK scheme is available in the pressure-based solver and when solving additional scalar equations in the density-based solver.

Third-Order MUSCL Scheme

This third-order convection scheme was conceived from the original MUSCL (Monotone Upstream-Centered Schemes for Conservation Laws) [352] by blending a central differencing scheme and second-order upwind scheme as

$$\phi_f = \theta \phi_{f,\text{CD}} + (1-\theta)\phi_{f,\text{SOU}} \tag{18.3-8}$$

where $\phi_{f,\text{CD}}$ is defined in Equation 18.3-5, and $\phi_{f,\text{SOU}}$ is computed using the second-order upwind scheme as described in Section 18.3.1: Second-Order Upwind Scheme.

Unlike the QUICK scheme which is applicable to structured hex meshes only, the MUSCL scheme is applicable to arbitrary meshes. Compared to the second-order upwind scheme, the third-order MUSCL has a potential to improve spatial accuracy for all types of meshes by reducing numerical diffusion, most significantly for complex three-dimensional flows, and it is available for all transport equations.

i The third-order MUSCL currently implemented in ANSYS FLUENT does not contain any flux-limiter. As a result, it can produce undershoots and overshoots when the flow-field under consideration has discontinuities such as shock waves.

i The MUSCL scheme is available in the pressure-based and density-based solvers.

Modified HRIC Scheme

For simulations using the VOF multiphase model, upwind schemes are generally unsuitable for interface tracking because of their overly diffusive nature. Central differencing schemes, while generally able to retain the sharpness of the interface, are unbounded and often give unphysical results. In order to overcome these deficiencies, ANSYS FLUENT uses a modified version of the High Resolution Interface Capturing (HRIC) scheme. The modified HRIC scheme is a composite NVD scheme that consists of a non-linear blend of upwind and downwind differencing [243].

First, the normalized cell value of volume fraction, $\tilde{\phi}_c$, is computed and is used to find the normalized face value, $\tilde{\phi}_f$, as follows:

$$\tilde{\phi}_c = \frac{\phi_D - \phi_U}{\phi_A - \phi_U} \tag{18.3-9}$$

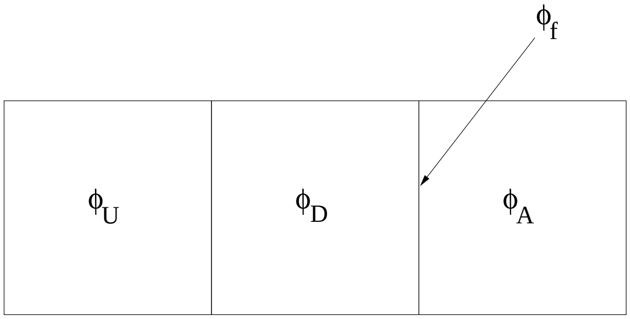


Figure 18.3.3: Cell Representation for Modified HRIC Scheme

where A is the acceptor cell, D is the donor cell, and U is the upwind cell, and

$$\tilde{\phi}_f = \begin{cases} \tilde{\phi}_c & \tilde{\phi}_c < 0 \text{ or } \tilde{\phi}_c > 1 \\ 2\tilde{\phi}_c & 0 \leq \tilde{\phi}_c \leq 0.5 \\ 1 & 0.5 \leq \tilde{\phi}_c \leq 1 \end{cases} \tag{18.3-10}$$

Here, if the upwind cell is not available (e.g., unstructured mesh), an extrapolated value is used for ϕ_U. Directly using this value of $\tilde{\phi}_f$ causes wrinkles in the interface, if the flow is parallel to the interface. So, ANSYS FLUENT switches to the ULTIMATE QUICKEST scheme (the one-dimensional bounded version of the QUICK scheme [187]) based on the angle between the face normal and interface normal:

$$\phi_f^{\tilde{U}Q} = \begin{cases} \tilde{\phi}_c & \tilde{\phi}_c < 0 \text{ or } \tilde{\phi}_c > 1 \\ MIN\left(\tilde{\phi}_f, \frac{6\tilde{\phi}_c+3}{8}\right) & 0 \leq \tilde{\phi}_c \leq 1 \end{cases} \tag{18.3-11}$$

This leads to a corrected version of the face volume fraction, $\tilde{\phi}_f^*$:

$$\tilde{\phi}_f^* = \tilde{\phi}_f \sqrt{\cos\theta} + (1 - \sqrt{\cos\theta})\phi_f^{\tilde{U}Q} \tag{18.3-12}$$

where

$$\cos\theta = \frac{\nabla\phi \cdot \vec{\mathbf{d}}}{|\nabla\phi||\vec{\mathbf{d}}|} \tag{18.3-13}$$

and $\vec{\mathbf{d}}$ is a vector connecting cell centers adjacent to the face f.

The face volume fraction is now obtained from the normalized value computed above as follows:

$$\phi_f = \tilde{\phi}_f^*(\phi_A - \phi_U) + \phi_U \tag{18.3-14}$$

The modified HRIC scheme provides improved accuracy for VOF calculations when compared to QUICK and second-order schemes, and is less computationally expensive than the Geo-Reconstruct scheme.

18.3.2 Temporal Discretization

For transient simulations, the governing equations must be discretized in both space and time. The spatial discretization for the time-dependent equations is identical to the steady-state case. Temporal discretization involves the integration of every term in the differential equations over a time step Δt. The integration of the transient terms is straightforward, as shown below.

A generic expression for the time evolution of a variable ϕ is given by

$$\frac{\partial\phi}{\partial t} = F(\phi) \tag{18.3-15}$$

where the function F incorporates any spatial discretization. If the time derivative is discretized using backward differences, the first-order accurate temporal discretization is given by

$$\frac{\phi^{n+1} - \phi^n}{\Delta t} = F(\phi) \tag{18.3-16}$$

and the second-order discretization is given by

$$\frac{3\phi^{n+1} - 4\phi^n + \phi^{n-1}}{2\Delta t} = F(\phi) \tag{18.3-17}$$

where

ϕ = a scalar quantity
$n+1$ = value at the next time level, $t + \Delta t$
n = value at the current time level, t
$n-1$ = value at the previous time level, $t - \Delta t$

Once the time derivative has been discretized, a choice remains for evaluating $F(\phi)$: in particular, which time level values of ϕ should be used in evaluating F?

Implicit Time Integration

One method is to evaluate $F(\phi)$ at the future time level:

$$\frac{\phi^{n+1} - \phi^n}{\Delta t} = F(\phi^{n+1}) \qquad (18.3\text{-}18)$$

This is referred to as "implicit" integration since ϕ^{n+1} in a given cell is related to ϕ^{n+1} in neighboring cells through $F(\phi^{n+1})$:

$$\phi^{n+1} = \phi^n + \Delta t F(\phi^{n+1}) \qquad (18.3\text{-}19)$$

This implicit equation can be solved iteratively at each time level before moving to the next time step.

The advantage of the fully implicit scheme is that it is unconditionally stable with respect to time step size.

Explicit Time Integration

A second method is available when the density-based explicit solver is used. This method evaluates $F(\phi)$ at the current time level:

$$\frac{\phi^{n+1} - \phi^n}{\Delta t} = F(\phi^n) \qquad (18.3\text{-}20)$$

and is referred to as "explicit" integration since ϕ^{n+1} can be expressed explicitly in terms of the existing solution values, ϕ^n:

$$\phi^{n+1} = \phi^n + \Delta t F(\phi^n) \qquad (18.3\text{-}21)$$

Here, the time step Δt is restricted to the stability limit of the underlying solver (i.e., a time step is limited by the Courant-Friedrich-Lewy condition). In order to be time-accurate, all cells in the domain must use the same time step. For stability, this time step must be the minimum of all the local time steps in the domain. This method is also referred to as "global time stepping".

The use of explicit time stepping is fairly restrictive. It is used primarily to capture the transient behavior of moving waves, such as shocks, because it is more accurate and less expensive than the implicit time stepping methods in such cases. You *cannot* use explicit time stepping in the following cases:

- Calculations with the pressure-based solver or density-based implicit formulation. The explicit time stepping formulation is available only with the density-based explicit formulation. ANSYS FLUENT also uses multi-stage Runge-Kutta explicit time integration for the density-based solver, as detailed in Sections 18.5.4 and 18.5.5
- Incompressible flow. Explicit time stepping cannot be used to compute time-accurate incompressible flows (i.e., gas laws other than ideal gas). Incompressible solutions must be iterated to convergence within each time step.
- Convergence acceleration. FAS multigrid and residual smoothing cannot be used with explicit time stepping because they destroy the time accuracy of the underlying solver.

18.3.3 Evaluation of Gradients and Derivatives

Gradients are needed not only for constructing values of a scalar at the cell faces, but also for computing secondary diffusion terms and velocity derivatives. The gradient $\nabla\phi$ of a given variable ϕ is used to discretize the convection and diffusion terms in the flow conservation equations. The gradients are computed in ANSYS FLUENT according to the following methods:

- Green-Gauss Cell-Based
- Green-Gauss Node-Based
- Least Squares Cell-Based

To learn how to apply the various gradients, see Section 26.2: Choosing the Spatial Discretization Scheme in the separate User's Guide.

Green-Gauss Theorem

When the Green-Gauss theorem is used to compute the gradient of the scalar ϕ at the cell center $c0$, the following discrete form is written as

$$(\nabla\phi)_{c0} \quad = \quad \frac{1}{\mathcal{V}}\sum_f \overline{\phi}_f \, \vec{A}_f \tag{18.3-22}$$

where ϕ_f is the value of ϕ at the cell face centroid, computed as shown in the sections below. The summation is over all the faces enclosing the cell.

Green-Gauss Cell-Based Gradient Evaluation

By default, the face value, $\overline{\phi}_f$, in Equation 18.3-22 is taken from the arithmetic average of the values at the neighboring cell centers, i.e.,

$$\overline{\phi}_f = \frac{\phi_{c0} + \phi_{c1}}{2} \tag{18.3-23}$$

Green-Gauss Node-Based Gradient Evaluation

Alternatively, $\overline{\phi}_f$ can be computed by the arithmetic average of the nodal values on the face.

$$\overline{\phi}_f = \frac{1}{N_f} \sum_{n}^{N_f} \overline{\phi}_n \tag{18.3-24}$$

where N_f is the number of nodes on the face.

The nodal values, $\overline{\phi}_n$ in Equation 18.3-24, are constructed from the weighted average of the cell values surrounding the nodes, following the approach originally proposed by Holmes and Connel[132] and Rauch *et al.*[286]. This scheme reconstructs exact values of a linear function at a node from surrounding cell-centered values on arbitrary unstructured meshes by solving a constrained minimization problem, preserving a second-order spatial accuracy.

The node-based gradient is known to be more accurate than the cell-based gradient particularly on irregular (skewed and distorted) unstructured meshes, however, it is relatively more expensive to compute than the cell-based gradient scheme.

i The node-based gradient method is not available with polyhedral meshes.

Least Squares Cell-Based Gradient Evaluation

In this method the solution is assumed to vary linearly. In Figure 18.3.4, the change in cell values between cell $c0$ and ci along the vector δr_i from the centroid of cell $c0$ to cell ci, can be expressed as

$$(\nabla\phi)_{c0} \cdot \Delta r_i = (\phi_{ci} - \phi_{c0}) \tag{18.3-25}$$

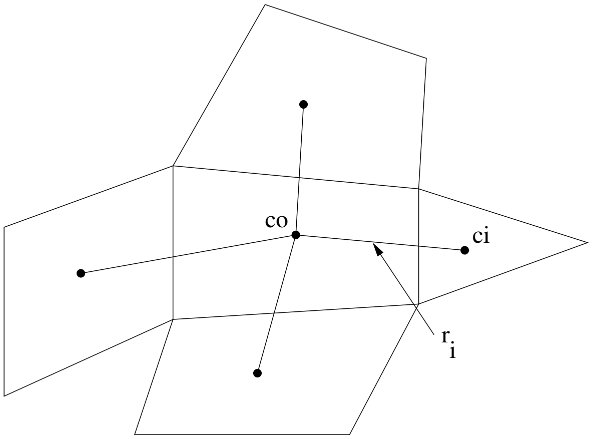


Figure 18.3.4: Cell Centroid Evaluation

If we write similar equations for each cell surrounding the cell c0, we obtain the following system written in compact form:

$$[J](\nabla\phi)_{c0} = \Delta\phi \tag{18.3-26}$$

Where [J] is the coefficient matrix which is purely a function of geometry.

The objective here is to determine the cell gradient ($\nabla\phi_0 = \phi_x\hat{\imath}+\phi_y\hat{\jmath}+\phi_z\hat{k}$) by solving the minimization problem for the system of the non-square coefficient matrix in a least-squares sense.

The above linear-system of equation is over-determined and can be solved by decomposing the coefficient matrix using the Gram-Schmidt process [6]. This decomposition yields a matrix of weights for each cell. Thus for our cell-centered scheme this means that the three components of the weights (${W^x}_{i0}, {W^y}_{i0}, {W^z}_{i0}$) are produced for each of the faces of cell c0.

Therefore, the gradient at the cell center can then be computed by multiplying the weight factors by the difference vector $\Delta\phi = (\phi_{c1} - \phi_{c0})$,

$$\begin{aligned}
(\phi_x)_{c0} &= \sum_{i=1}^{n} W^x{}_{i0} \cdot (\phi_{ci} - \phi_{c0}) && (18.3\text{-}27)\\
(\phi_y)_{c0} &= \sum_{i=1}^{n} W^y{}_{i0} \cdot (\phi_{ci} - \phi_{c0}) && (18.3\text{-}28)\\
(\phi_z)_{c0} &= \sum_{i=1}^{n} W^z{}_{i0} \cdot (\phi_{ci} - \phi_{c0}) && (18.3\text{-}29)
\end{aligned}$$

On irregular (skewed and distorted) unstructured meshes, the accuracy of the least-squares gradient method is comparable to that of the node-based gradient (and both are much more superior compared to the cell-based gradient). However, it is less expensive to compute the least-squares gradient than the node-based gradient. Therefore, it has been selected as the default gradient method in the ANSYS FLUENT solver.

18.3.4 Gradient Limiters

Gradient limiters, also known as slope limiters, are used on the second-order upwind (SOU) scheme to prevent spurious oscillations, which would otherwise appear in the solution flow field near shocks, discontinuities, or near rapid local changes in the flow field. The gradient limiter attempts to invoke and enforce the monotonicity principle by prohibiting the linearly reconstructed field variable on the cell faces to exceed the maximum or minimum values of the neighboring cells.

There are three gradient limiters in the ANSYS FLUENT solvers:

- Standard limiter
- Multidimensional limiter
- Differentiable limiter

Gradient limiters can be categorized into two general groups: non-differentiable limiters and differentiable limiters. Both, the standard limiter and multidimensional limiter are of the non-differentiable form, since they use minimum and maximum types of functions for limiting the solution variables. The third limiter in ANSYS FLUENT, as the name indicates, is a differentiable type of limiter, which uses a smooth function to impose the monotonicity principle.

For each of the above mentioned limiter methods, ANSYS FLUENT provides two limiting directions:

- cell to face limiting, where the limited value of the reconstruction gradient is determined at cell face centers. This is the default method.
- cell to cell limiting, where the limited value of the reconstruction gradient is determined along a scaled line between two adjacent cell centroids. On an orthogonal mesh (or when cell-to-cell direction is parallel to face area direction) this method becomes equivalent to the default cell to face method. For smooth field variation, cell to cell limiting may provide less numerical dissipation on meshes with skewed cells.

For more information about how to access the limiter functions in ANSYS FLUENT through the GUI or TUI, see Section 26.8: Selecting Gradient Limiters in the separate User's Guide.

i On unstructured meshes, ANSYS FLUENT uses the scalar form of the gradient limiter given by the following equation:

$$\Phi_{fSOU} = \Phi + \psi \nabla \phi \cdot r \tag{18.3-30}$$

Where ψ is a scalar value which limits the gradient $\nabla \Phi$.

Standard Limiter

The standard limiter is the default limiter function in ANSYS FLUENT and is derived from the work of Barth and Jespersen [14]. This limiter is of a non-differentiable type and uses the Minmod function (Minimum Modulus) to limit and clip the reconstructed solution overshoots and undershoots on the cell faces.

Multidimensional Limiter

The multidimensional limiter in ANSYS FLUENT [164] has a similar form to the standard limiter. Since the multidimensional limiter uses a Minmod function for limiting the gradient, it is also classified as a non-differentiable type of limiter. However, in the standard limiter formulation, if limiting took place on any face of the cell, then this will cause the cell gradient to be clipped in an equal manner, in all directions, regardless of whether or not limiting is needed on the other cell faces. This limiting method is rather severe and adds unnecessary dissipation to the numerical scheme. The multidimensional limiter, on the other hand, attempts to lessen the severity of the gradient limiting by carefully examining the gradient on each cell and clipping only the normal components of the gradient to the cell faces. For this procedure to work on a scalar form limiter, the normal components of gradients on cell faces are first sorted out in ascending order of their magnitude so that only the necessary clipping can be applied. The multidimensional limiter is therefore less dissipative than the standard limiter.

Differentiable Limiter

One disadvantage with non-differentiable limiters is that they tend to stall the apparent residual's convergence after a few orders of reduction in residual magnitude. Note that this does not mean that the solution is not converging, but rather the solution continues to converge while the residuals are stalling. This annoying behavior can be directly traced to the non-differentiable nature of the limiting functions. Therefore, the differentiable limiter uses a smooth function to impose the monotonicity condition while allowing the residuals to converge. The differentiable limiter used in ANSYS FLUENT is a modified form [366] of a limiter which was originally proposed by Venkatakrishnan[355].

> ANSYS FLUENT uses gradient or slope limiters and not flux limiters. Gradient limiters are applied to the gradients of the variable field being linearly reconstructed at the cell faces, while flux limiters are used on the system fluxes.

18.4 Pressure-Based Solver

In this section, special practices related to the discretization of the momentum and continuity equations and their solution by means of the pressure-based solver are addressed. Information is organized into the following subsections:

- Section 18.4.1: Discretization of the Momentum Equation
- Section 18.4.2: Discretization of the Continuity Equation
- Section 18.4.3: Pressure-Velocity Coupling
- Section 18.4.4: Steady-State Iterative Algorithm
- Section 18.4.5: Time-Advancement Algorithm

These special practices are most easily described by considering the steady-state continuity and momentum equations in integral form:

$$\oint \rho \, \vec{v} \cdot d\vec{A} = 0 \tag{18.4-1}$$

$$\oint \rho \vec{v} \, \vec{v} \cdot d\vec{A} = - \oint p \mathbf{I} \cdot d\vec{A} + \oint \overline{\overline{\tau}} \cdot d\vec{A} + \int_V \vec{F} \, dV \tag{18.4-2}$$

where $\mathbf{I}$ is the identity matrix, $\overline{\overline{\tau}}$ is the stress tensor, and $\vec{F}$ is the force vector.

18.4.1 Discretization of the Momentum Equation

The discretization scheme described in Section 18.3: Discretization for a scalar transport equation is also used to discretize the momentum equations. For example, the x-momentum equation can be obtained by setting $\phi = u$:

$$a_P\, u = \sum_{\rm nb} a_{\rm nb}\, u_{\rm nb} + \sum p_f \mathrm{A} \cdot \hat{\imath} + S \tag{18.4-3}$$

If the pressure field and face mass fluxes are known, Equation 18.4-3 can be solved in the manner outlined in Section 18.3: Discretization, and a velocity field obtained. However, the pressure field and face mass fluxes are not known a priori and must be obtained as a part of the solution. There are important issues with respect to the storage of pressure and the discretization of the pressure gradient term; these are addressed next.

ANSYS FLUENT uses a co-located scheme, whereby pressure and velocity are both stored at cell centers. However, Equation 18.4-3 requires the value of the pressure at the face between cells $c0$ and $c1$, shown in Figure 18.2.1. Therefore, an interpolation scheme is required to compute the face values of pressure from the cell values.

Pressure Interpolation Schemes

The default scheme in ANSYS FLUENT interpolates the pressure values at the faces using momentum equation coefficients [292]:

$$P_f = \frac{\frac{P_{c_0}}{a_{p,c_0}} + \frac{P_{c_1}}{a_{p,c_1}}}{\frac{1}{a_{p,c_0}} + \frac{1}{a_{p,c_1}}} \tag{18.4-4}$$

This procedure works well as long as the pressure variation between cell centers is smooth. When there are jumps or large gradients in the momentum source terms between control volumes, the pressure profile has a high gradient at the cell face, and cannot be interpolated using this scheme. If this scheme is used, the discrepancy shows up in overshoots/undershoots of cell velocity.

Flows for which the standard pressure interpolation scheme will have trouble include flows with large body forces, such as in strongly swirling flows, in high-Rayleigh-number natural convection and the like. In such cases, it is necessary to pack the mesh in regions of high gradient to resolve the pressure variation adequately.

Another source of error is that ANSYS FLUENT assumes that the normal pressure gradient at the wall is zero. This is valid for boundary layers, but not in the presence of body forces or curvature. Again, the failure to correctly account for the wall pressure gradient is manifested in velocity vectors pointing in/out of walls.

Several alternate methods are available for cases in which the standard pressure interpolation scheme is not valid:

- The linear scheme computes the face pressure as the average of the pressure values in the adjacent cells.
- The second-order scheme reconstructs the face pressure in the manner used for second-order accurate convection terms (see Section 18.3.1). This scheme may provide some improvement over the standard and linear schemes, but it may have some trouble if it is used at the start of a calculation and/or with a bad mesh. The second-order scheme is not applicable for flows with discontinuous pressure gradients imposed by the presence of a porous medium in the domain or the use of the VOF or mixture model for multiphase flow.
- The body-force-weighted scheme computes the face pressure by assuming that the normal gradient of the difference between pressure and body forces is constant. This works well if the body forces are known a priori in the momentum equations (e.g., buoyancy and axisymmetric swirl calculations).

  > *i* When a case contains porous media, the body-force-weighted scheme is applied only for non-porous faces, where the scheme takes into account the discontinuity of explicit body forces (e.g., gravity, swirl, Coriolis) and the discontinuity of pressure gradients for flows with rapidly changing densities (e.g., natural convection, VOF). All interior and exterior porous faces are treated with a special scheme that preserves the continuity of the normal velocity across cell faces in spite of the discontinuity of the resistance.

- The PRESTO! (PREssure STaggering Option) scheme uses the discrete continuity balance for a “staggered” control volume about the face to compute the “staggered” (i.e., face) pressure. This procedure is similar in spirit to the staggered-grid schemes used with structured meshes [264]. Note that for triangular, tetrahedral, hybrid, and polyhedral meshes, comparable accuracy is obtained using a similar algorithm. The PRESTO! scheme is available for all meshes.

For recommendations on when to use these alternate schemes, see Section 26.2.3: Choosing the Pressure Interpolation Scheme in the separate User’s Guide.

18.4.2 Discretization of the Continuity Equation

Equation 18.4-1 may be integrated over the control volume in Figure 18.2.1 to yield the following discrete equation

$$\sum_{f}^{N_{\text{faces}}} J_f A_f = 0 \tag{18.4-5}$$

where J_f is the mass flux through face f, ρv_n.

In order to proceed further, it is necessary to relate the face values of velocity, $\vec{v}_n$, to the stored values of velocity at the cell centers. Linear interpolation of cell-centered velocities to the face results in unphysical checker-boarding of pressure. ANSYS FLUENT uses a procedure similar to that outlined by Rhie and Chow [292] to prevent checkerboarding. The face value of velocity is not averaged linearly; instead, momentum-weighted averaging, using weighting factors based on the a_P coefficient from equation 18.4-3, is performed. Using this procedure, the face flux, J_f, may be written as

$$J_f = \rho_f \frac{a_{p,c_0} v_{n,c_0} + a_{p,c_1} v_{n,c_1}}{a_{p,c_0} + a_{p,c_1}} + d_f ((p_{c_0} + (\nabla p)_{c_0} \cdot \vec{r_0}) - (p_{c_1} + (\nabla p)_{c_1} \cdot \vec{r_1})) = \hat{J}_f + d_f \left(p_{c_0} - p_{c_1}\right) \tag{18.4-6}$$

where p_{c_0}, p_{c_1} and v_{n,c_0}, v_{n,c_1} are the pressures and normal velocities, respectively, within the two cells on either side of the face, and $\hat{J}_f$ contains the influence of velocities in these cells (see Figure 18.2.1). The term d_f is a function of $\bar{a}_P$, the average of the momentum equation a_P coefficients for the cells on either side of face f.

Density Interpolation Schemes

For incompressible flows, ANSYS FLUENT uses arithmetic averaging for density. For compressible flow calculations (i.e., calculations that use the ideal gas law for density), ANSYS FLUENT applies upwind interpolation of density at cell faces. Several interpolation schemes are available for the density upwinding at cell faces: first-order upwind (default), second-order-upwind, QUICK, MUSCL, and when applicable, central differencing and bounded central differencing.

The first-order upwind scheme (based on [157]) sets the density at the cell face to be the upstream cell-center value. This scheme provides stability for the discretization of the pressure-correction equation, and gives good results for most classes of flows. The first-order scheme is the default scheme for compressible flows. Although this scheme provides the best stability for compressible flow calculations, it gives very diffusive representations of shocks.

The second-order upwind scheme provides stability for supersonic flows and captures shocks better than the first-order upwind scheme. The QUICK scheme for density is similar to the QUICK scheme used for other variables. See Section 18.3.1: QUICK Scheme for details.

> ℹ In the case of multiphase flows, the selected density scheme is applied to the compressible phase and arithmetic averaging is used for incompressible phases.

> ℹ For stability reasons, it is recommended that you achieve a solution with a first order scheme and then switch to a higher order scheme for compressible flow calculations.

For recommendations on choosing an appropriate density interpolation scheme for your compressible flow, see Section 26.2.4: Choosing the Density Interpolation Scheme in the separate User's Guide.

18.4.3 Pressure-Velocity Coupling

Pressure-velocity coupling is achieved by using Equation 18.4-6 to derive an additional condition for pressure by reformatting the continuity equation (Equation 18.4-5). The pressure-based solver allows you to solve your flow problem in either a segregated or coupled manner. ANSYS FLUENT provides the option to choose among five pressure-velocity coupling algorithms: SIMPLE, SIMPLEC, PISO, Coupled, and (for unsteady flows using the non-iterative time advancement scheme (NITA)) Fractional Step (FSM). All the aforementioned schemes, except the "coupled" scheme, are based on the predictor-corrector approach. For instructions on how to select these algorithms, see Section 26.3.1: Choosing the Pressure-Velocity Coupling Method in the separate User's Guide.

Note that SIMPLE, SIMPLEC, PISO, and Fractional Step use the pressure-based segregated algorithm, while Coupled uses the pressure-based coupled solver.

> ℹ The pressure-velocity coupling schemes that are applicable when using the Eulerian multiphase model are Phase Coupled SIMPLE, Multiphase Coupled, and Full Multiphase Coupled. These are discussed in detail in Section 24.7.5: Selecting the Solution Method in the separate User's Guide.

Segregated Algorithms

SIMPLE

The SIMPLE algorithm uses a relationship between velocity and pressure corrections to enforce mass conservation and to obtain the pressure field.

If the momentum equation is solved with a guessed pressure field p^*, the resulting face flux, J_f^*, computed from Equation 18.4-6

$$J_f^* = \hat{J}_f^* + d_f \left(p_{c0}^* - p_{c1}^*\right) \tag{18.4-7}$$

does not satisfy the continuity equation. Consequently, a correction J_f' is added to the face flux J_f^* so that the corrected face flux, J_f

$$J_f = J_f^* + J_f' \tag{18.4-8}$$

satisfies the continuity equation. The SIMPLE algorithm postulates that J_f' be written as

$$J_f' = d_f \left(p_{c0}' - p_{c1}'\right) \tag{18.4-9}$$

where p' is the cell pressure correction.

The SIMPLE algorithm substitutes the flux correction equations (Equations 18.4-8 and 18.4-9) into the discrete continuity equation (Equation 18.4-5) to obtain a discrete equation for the pressure correction p' in the cell:

$$a_P \, p' = \sum_{\rm nb} a_{\rm nb} \, p_{\rm nb}' + b \tag{18.4-10}$$

where the source term b is the net flow rate into the cell:

$$b = \sum_{f}^{N_{\rm faces}} J_f^* A_f \tag{18.4-11}$$

The pressure-correction equation (Equation 18.4-10) may be solved using the algebraic multigrid (AMG) method described in Section 18.6.3: Algebraic Multigrid (AMG). Once a solution is obtained, the cell pressure and the face flux are corrected using

$$p = p^* + \alpha_p \, p' \tag{18.4-12}$$

$$J_f = J_f^* + d_f \,(p'_{c0} - p'_{c1}) \tag{18.4-13}$$

Here α_p is the under-relaxation factor for pressure (see Section 18.4.4: Under-Relaxation of Variables for information about under-relaxation). The corrected face flux, J_f, satisfies the discrete continuity equation identically during each iteration.

SIMPLEC

A number of variants of the basic SIMPLE algorithm are available in the literature. In addition to SIMPLE, ANSYS FLUENT offers the SIMPLEC (SIMPLE-Consistent) algorithm [353]. SIMPLE is the default, but many problems will benefit from the use of SIMPLEC, as described in Section 26.3.1: SIMPLE vs. SIMPLEC in the separate User's Guide.

The SIMPLEC procedure is similar to the SIMPLE procedure outlined above. The only difference lies in the expression used for the face flux correction, J'_f. As in SIMPLE, the correction equation may be written as

$$J_f = J_f^* + d_f \,(p'_{c0} - p'_{c1}) \tag{18.4-14}$$

However, the coefficient d_f is redefined as a function of $(\overline{a_P - \sum_{\rm nb} a_{\rm nb}})$. The use of this modified correction equation has been shown to accelerate convergence in problems where pressure-velocity coupling is the main deterrent to obtaining a solution.

Skewness Correction

For meshes with some degree of skewness, the approximate relationship between the correction of mass flux at the cell face and the difference of the pressure corrections at the adjacent cells is very rough. Since the components of the pressure-correction gradient along the cell faces are not known in advance, an iterative process similar to the PISO neighbor correction described below is desirable. After the initial solution of the pressure-correction equation, the pressure-correction gradient is recalculated and used to update the mass flux corrections. This process, which is referred to as "skewness correction", significantly reduces convergence difficulties associated with highly distorted meshes. The SIMPLEC skewness correction allows ANSYS FLUENT to obtain a solution on a highly skewed mesh in approximately the same number of iterations as required for a more orthogonal mesh.

PISO

The Pressure-Implicit with Splitting of Operators (PISO) pressure-velocity coupling scheme, part of the SIMPLE family of algorithms, is based on the higher degree of the approximate relation between the corrections for pressure and velocity. One of the limitations of the SIMPLE and SIMPLEC algorithms is that new velocities and corresponding fluxes do not satisfy the momentum balance after the pressure-correction equation is solved. As a result, the calculation must be repeated until the balance is satisfied. To improve the efficiency of this calculation, the PISO algorithm performs two additional corrections: neighbor correction and skewness correction.

Neighbor Correction

The main idea of the PISO algorithm is to move the repeated calculations required by SIMPLE and SIMPLEC inside the solution stage of the pressure-correction equation [141]. After one or more additional PISO loops, the corrected velocities satisfy the continuity and momentum equations more closely. This iterative process is called a momentum correction or "neighbor correction". The PISO algorithm takes a little more CPU time per solver iteration, but it can dramatically decrease the number of iterations required for convergence, especially for transient problems.

Skewness Correction

For meshes with some degree of skewness, the approximate relationship between the correction of mass flux at the cell face and the difference of the pressure corrections at the adjacent cells is very rough. Since the components of the pressure-correction gradient along the cell faces are not known in advance, an iterative process similar to the PISO neighbor correction described above is desirable [94]. After the initial solution of the pressure-correction equation, the pressure-correction gradient is recalculated and used to update the mass flux corrections. This process, which is referred to as "skewness correction", significantly reduces convergence difficulties associated with highly distorted meshes. The PISO skewness correction allows ANSYS FLUENT to obtain a solution on a highly skewed mesh in approximately the same number of iterations as required for a more orthogonal mesh.

Skewness - Neighbor Coupling

For meshes with a high degree of skewness, the simultaneous coupling of the neighbor and skewness corrections at the same pressure correction equation source may cause divergence or a lack of robustness. An alternate, although more expensive, method for handling the neighbor and skewness corrections inside the PISO algorithm is to apply one or more iterations of skewness correction for each separate iteration of neighbor correction. For each individual iteration of the classical PISO algorithm from [141], this technique allows a more accurate adjustment of the face mass flux correction according to the normal pressure correction gradient.

Fractional-Step Method (FSM)

In the FSM, the momentum equations are decoupled from the continuity equation using a mathematical technique called operator-splitting or approximate factorization. The resulting solution algorithm is similar to the segregated solution algorithms described earlier. The formalism used in the approximate factorization allows you to control the order of splitting error. Because of this, the FSM is adopted in ANSYS FLUENT as a velocity-coupling scheme in a non-iterative time-advancement (NITA) algorithm (Section 18.4.5: Non-Iterative Time-Advancement Scheme).

Coupled Algorithm

As previously mentioned, the pressure-based solver allows you to solve your flow problem in either a segregated or coupled manner. Using the coupled approach offers some advantages over the non-coupled or segregated approach. The coupled scheme obtains a robust and efficient single phase implementation for steady-state flows, with superior performance compared to the segregated solution schemes. This pressure-based coupled algorithm offers an alternative to the density-based and pressure-based segregated algorithm with SIMPLE-type pressure-velocity coupling. For transient flows, using the coupled algorithm is necessary when the quality of the mesh is poor, or if large time steps are used.

The pressure-based segregated algorithm solves the momentum equation and pressure correction equations separately. This semi-implicit solution method results in slow convergence.

The coupled algorithm solves the momentum and pressure-based continuity equations together. The full implicit coupling is achieved through an implicit discretization of pressure gradient terms in the momentum equations, and an implicit discretization of the face mass flux, including the Rhie-Chow pressure dissipation terms.

In the momentum equations (18.4-3), the pressure gradient for component k is of the form

$$\sum_f p_f A_k = -\sum_j a^{u_k p} p_j \tag{18.4-15}$$

Where $a^{u_k p}$ is the coefficient derived from the Gauss divergence theorem and coefficients of the pressure interpolation schemes (Equation 18.4-4). Finally, for any ith cell, the discretized form of the momentum equation for component u_k is defined as

$$\sum_j a_{ij}{}^{u_k u_k} u_{kj} + \sum_j a_{ij}{}^{u_k p} p_j = b_i{}^{u_k} \tag{18.4-16}$$

In the continuity equation, Equation 18.4-5, the balance of fluxes is replaced using the flux expression in Equation 18.4-6, resulting in the discretized form

$$\sum_k \sum_j a_{ij}{}^{pu_k} u_{kj} + \sum_j a_{ij}{}^{pp} p_j = b_i{}^p \tag{18.4-17}$$

As a result, the overall system of equations (18.4-16 and 18.4-17), after being transformed to the δ-form, is presented as

$$\sum_j [A]_{ij} \vec{X}_j = \vec{B}_i \tag{18.4-18}$$

where the influence of a cell i on a cell j has the form

$$A_{ij} = \begin{bmatrix} a_{ij}^{pp} & a_{ij}^{pu} & a_{ij}^{pv} & a_{ij}^{pw} \\ a_{ij}^{up} & a_{ij}^{uu} & a_{ij}^{uv} & a_{ij}^{uw} \\ a_{ij}^{vp} & a_{ij}^{vu} & a_{ij}^{vv} & a_{ij}^{vw} \\ a_{ij}^{wp} & a_{ij}^{wu} & a_{ij}^{wv} & a_{ij}^{ww} \end{bmatrix} \tag{18.4-19}$$

and the unknown and residual vectors have the form

$$\vec{X}_j = \begin{bmatrix} p_i^{'} \\ u_i^{'} \\ v_i^{'} \\ w_i^{'} \end{bmatrix} \tag{18.4-20}$$

$$\vec{B}_i = \begin{bmatrix} -r_i^p \\ -r_i^u \\ -r_i^v \\ -r_i^w \end{bmatrix} \tag{18.4-21}$$

Note that Equation 18.4-18 is solved using the coupled AMG, which is detailed in Section 18.6.3: The Coupled and Scalar AMG Solvers.

Limitations

The pressure-based coupled algorithm is not compatible with

- The non-iterative time advancement solver (NITA)
- Periodic mass-flow boundary conditions
- The fixed velocity option

18.4.4 Steady-State Iterative Algorithm

If you are performing a steady-state calculation, the governing equations for the pressure-based solver do not contain time-dependent terms. For steady-state flows,
Section 18.3: Discretization describes control-volume-based discretization of the steady-state transport equation (see Equation 18.2-1).

Under-Relaxation of Variables

The under-relaxation of variables is used in all cases for some material properties, in the NITA solver for solution variables, and in the pressure-based coupled algorithm where this explicit under-relaxation is used for momentum and pressure.

Because of the nonlinearity of the equation set being solved by ANSYS FLUENT, it is necessary to control the change of ϕ. This is typically achieved by under-relaxation of variables (also referred to as explicit relaxation), which reduces the change of ϕ produced during each iteration. In a simple form, the new value of the variable ϕ within a cell depends upon the old value, $\phi_{\rm old}$, the computed change in ϕ, $\Delta\phi$, and the under-relaxation factor, α, as follows:

$$\phi = \phi_{\rm old} + \alpha\Delta\phi \tag{18.4-22}$$

Under-Relaxation of Equations

The under-relaxation of equations, also known as implicit relaxation, is used in the pressure-based solver to stabilize the convergence behavior of the outer nonlinear iterations by introducing selective amounts of ϕ in the system of discretized equations. This is equivalent to the location-specific time step.

$$\frac{a_p\phi}{\alpha} = \sum_{nb} a_{nb}\phi_{nb} + b + \frac{1-\alpha}{\alpha} a_p\phi_{\rm old} \tag{18.4-23}$$

The CFL number is a solution parameter in the pressure-based coupled algorithm and can be written in terms of α:

$$\frac{1-\alpha}{\alpha} = \frac{1}{CFL} \tag{18.4-24}$$

18.4.5 Time-Advancement Algorithm

For time-dependent flows, the discretized form of the generic transport equations is of the following form:

$$\int_V \frac{\partial \rho\phi}{\partial t}\, dV + \oint \rho\phi\, \vec{v} \cdot d\vec{A} = \oint \Gamma_\phi\, \nabla\phi \cdot d\vec{A} + \int_V S_\phi\, dV \qquad (18.4\text{-}25)$$

where

$\frac{\partial\rho\phi}{\partial t}$ = conservative form of transient derivative of transported variable ϕ
ρ = density
$\vec{v}$ = velocity vector ($= u\,\hat{\imath} + v\,\hat{\jmath}$ in 2D)
$\vec{A}$ = surface area vector
Γ_ϕ = diffusion coefficient for ϕ
$\nabla\phi$ = gradient of ϕ ($= (\partial\phi/\partial x)\,\hat{\imath} + (\partial\phi/\partial y)\,\hat{\jmath}$ in 2D)
S_ϕ = source of ϕ per unit volume

The temporal discretization of the transient derivative in the Equation 18.4-25 is described in Section 18.3.2: Temporal Discretization, including first-order and second-order schemes in time. The pressure-based solver in ANSYS FLUENT uses an implicit discretization of the transport equation (Equation 18.4-25). As a standard default approach, all convective, diffusive, and source terms are evaluated from the fields for time level *n+1*.

$$\int_V \frac{\partial \rho\phi}{\partial t}\, dV + \oint \rho^{n+1}\phi^{n+1}\, \vec{v}^{n+1} \cdot d\vec{A} = \oint {\Gamma_\phi}^{n+1}\, \nabla\phi^{n+1} \cdot d\vec{A} + \int_V {S_\phi}^{n+1}\, dV \qquad (18.4\text{-}26)$$

In the pressure-based solver, the overall time-discretization error is determined by both the choice of temporal discretization (e.g., first-order, second-order) and the manner in which the solutions are advanced to the next time step (time-advancement scheme). Temporal discretization introduces the corresponding truncation error; $O(\Delta t)$, $O\left[(\Delta t)^2\right]$, for first-order and second-order, respectively. The segregated solution process by which the equations are solved one by one introduces splitting error. There are two approaches to the time-advancement scheme depending on how you want to control the splitting error.

Iterative Time-Advancement Scheme

In the iterative scheme, all the equations are solved iteratively, for a given time-step, until the convergence criteria are met. Thus, advancing the solutions by one time-step normally requires a number of outer iterations as shown in Figure 18.1.1 and Figure 18.4.1. With this iterative scheme, non-linearity of the individual equations and inter-equation couplings are fully accounted for, eliminating the splitting error. The iterative scheme is the default in ANSYS FLUENT.

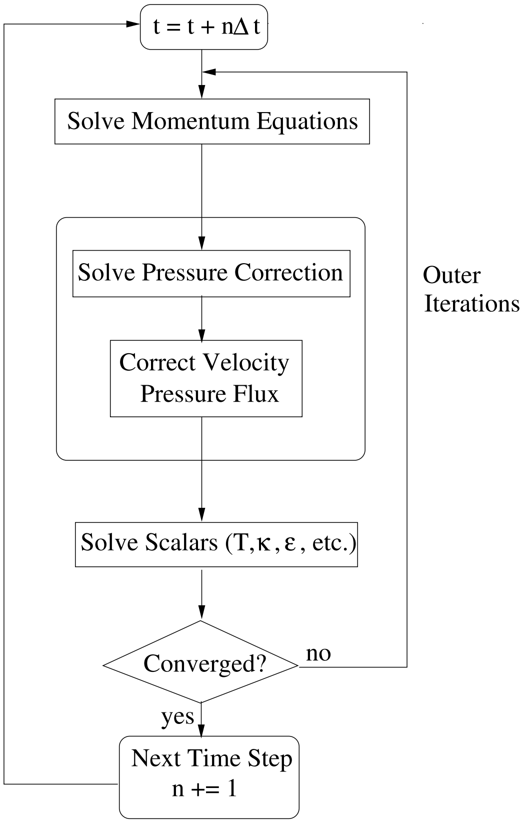


Figure 18.4.1: Overview of the Iterative Time Advancement Solution Method For the Segregate Solver

The Frozen Flux Formulation

The standard fully-implicit discretization of the convective part of Equation 18.4-26 produces non-linear terms in the resulting equations. In addition, solving these equations generally requires numerous iterations per time step. As an alternative, ANSYS FLUENT provides an optional way to discretize the convective part of Equation 18.4-25 using the mass flux at the cell faces from the previous time level n.

$$\oint \rho\phi\,\vec{v}\cdot d\vec{A} = \oint \rho^n \phi^{n+1}\,\vec{v}^n \cdot d\vec{A} \qquad (18.4\text{-}27)$$

The solution still has the same order of accuracy but the non-linear character of the discretized transport equation is essentially reduced and the convergence within each time step is improved.

The Frozen Flux Formulation option is accessible from the Solution Methods dialog box.

This option is only available for single-phase transient problems that use the segregated iterative solver and do not use a moving/deforming mesh model.

Non-Iterative Time-Advancement Scheme

The iterative time-advancement scheme requires a considerable amount of computational effort due to a large number of outer iterations performed for each time-step. The idea underlying the non-iterative time-advancement (NITA) scheme is that, in order to preserve overall time accuracy, you do not really need to reduce the splitting error to zero, but only have to make it the same order as the truncation error. The NITA scheme, as seen in Figure 18.4.2, does not need the outer iterations, performing only a single outer iteration per time-step, which significantly speeds up transient simulations. However, the NITA scheme still allows for an inner iteration to solve the individual set of equations.

ANSYS FLUENT offers two versions of NITA schemes; the non-iterative fractional step method (FSM) ([8], [81], [142], and [143]) and the non-iterative PISO method [140]. Both NITA schemes are available for first-order and second-order time integration.

In general, the NITA solver is not recommended for highly viscous fluid flow.

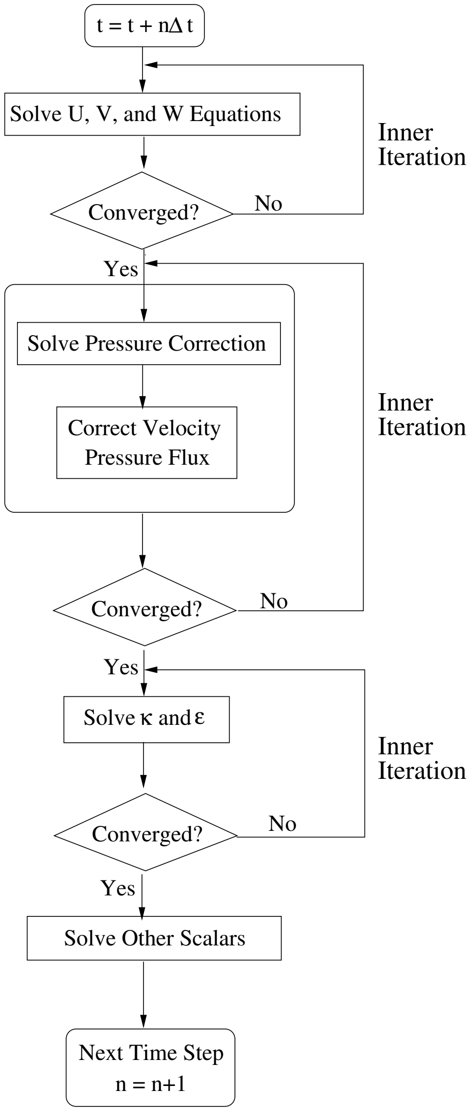


Figure 18.4.2: Overview of the Non-Iterative Time Advancement Solution Method

18.5 Density-Based Solver

The density-based solver in ANSYS FLUENT solves the governing equations of continuity, momentum, and (where appropriate) energy and species transport simultaneously as a set, or vector, of equations. Governing equations for additional scalars will be solved sequentially (i.e., segregated from one another and from the coupled set), in a manner described in Section 18.2: General Scalar Transport Equation: Discretization and Solution. Two algorithms are available for solving the coupled set of equations, the coupled-explicit formulation and the coupled-implicit formulation.

Information is organized into the following subsections:

- Section 18.5.1: Governing Equations in Vector Form
- Section 18.5.2: Preconditioning
- Section 18.5.3: Convective Fluxes
- Section 18.5.4: Steady-State Flow Solution Methods
- Section 18.5.5: Unsteady Flows Solution Methods

18.5.1 Governing Equations in Vector Form

The system of governing equations for a single-component fluid, written to describe the mean flow properties, is cast in integral Cartesian form for an arbitrary control volume V with differential surface area $d\mathbf{A}$ as follows:

$$\frac{\partial}{\partial t}\int_V \mathbf{W}\,dV + \oint [\mathbf{F} - \mathbf{G}]\cdot d\mathbf{A} = \int_V \mathbf{H}\,dV \tag{18.5-1}$$

where the vectors $\mathbf{W}, \mathbf{F}$, and $\mathbf{G}$ are defined as

$$\mathbf{W} = \left\{\begin{matrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E \end{matrix}\right\}, \; \mathbf{F} = \left\{\begin{matrix} \rho\mathbf{v} \\ \rho\mathbf{v}u \; + p\hat{\mathbf{i}} \\ \rho\mathbf{v}v \; + p\hat{\mathbf{j}} \\ \rho\mathbf{v}w + p\hat{\mathbf{k}} \\ \rho\mathbf{v}E + p\mathbf{v} \end{matrix}\right\}, \; \mathbf{G} = \left\{\begin{matrix} 0 \\ \tau_{xi} \\ \tau_{yi} \\ \tau_{zi} \\ \tau_{ij}v_j + \mathbf{q} \end{matrix}\right\}$$

and the vector $\mathbf{H}$ contains source terms such as body forces and energy sources.

Here ρ, $\mathbf{v}$, E, and p are the density, velocity, total energy per unit mass, and pressure of the fluid, respectively. τ is the viscous stress tensor, and $\mathbf{q}$ is the heat flux.

Total energy E is related to the total enthalpy H by

$$E = H - p/\rho \tag{18.5-2}$$

where

$$H = h + |\mathbf{v}|^2/2 \tag{18.5-3}$$

The Navier-Stokes equations as expressed in Equation 18.5-1 become (numerically) very stiff at low Mach number due to the disparity between the fluid velocity $\mathbf{v}$ and the acoustic speed c (speed of sound). This is also true for incompressible flows, regardless of the fluid velocity, because acoustic waves travel infinitely fast in an incompressible fluid (speed of sound is infinite). The numerical stiffness of the equations under these conditions results in poor convergence rates. This difficulty is overcome in ANSYS FLUENT's density-based solver by employing a technique called (time-derivative) preconditioning [372].

18.5.2 Preconditioning

Time-derivative preconditioning modifies the time-derivative term in Equation 18.5-1 by pre-multiplying it with a preconditioning matrix. This has the effect of re-scaling the acoustic speed (eigenvalue) of the system of equations being solved in order to alleviate the numerical stiffness encountered in low Mach numbers and incompressible flow.

Derivation of the preconditioning matrix begins by transforming the dependent variable in Equation 18.5-1 from conserved quantities $\mathbf{W}$ to primitive variables $\mathbf{Q}$ using the chain-rule as follows:

$$\frac{\partial \mathbf{W}}{\partial \mathbf{Q}} \frac{\partial}{\partial t} \int_V \mathbf{Q}\, dV + \oint [\mathbf{F} - \mathbf{G}] \cdot d\mathbf{A} = \int_V \mathbf{H}\, dV \tag{18.5-4}$$

where $\mathbf{Q}$ is the vector $\{p, u, v, w, T\}^T$ and the Jacobian $\partial\mathbf{W}/\partial\mathbf{Q}$ is given by

$$\frac{\partial \mathbf{W}}{\partial \mathbf{Q}} = \begin{bmatrix} \rho_p & 0 & 0 & 0 & \rho_T \\ \rho_p u & \rho & 0 & 0 & \rho_T u \\ \rho_p v & 0 & \rho & 0 & \rho_T v \\ \rho_p w & 0 & 0 & \rho & \rho_T w \\ \rho_p H - \delta & \rho u & \rho v & \rho w & \rho_T H + \rho C_p \end{bmatrix} \tag{18.5-5}$$

where

$$\rho_p = \left.\frac{\partial \rho}{\partial p}\right|_T, \ \rho_T = \left.\frac{\partial \rho}{\partial T}\right|_p$$

and $\delta = 1$ for an ideal gas and $\delta = 0$ for an incompressible fluid.

The choice of primitive variables $\mathbf{Q}$ as dependent variables is desirable for several reasons. First, it is a natural choice when solving incompressible flows. Second, when we use second-order accuracy we need to reconstruct $\mathbf{Q}$ rather than $\mathbf{W}$ in order to obtain more accurate velocity and temperature gradients in viscous fluxes, and pressure gradients in inviscid fluxes. And finally, the choice of pressure as a dependent variable allows the propagation of acoustic waves in the system to be singled out [356].

We precondition the system by replacing the Jacobian matrix $\partial\mathbf{W}/\partial\mathbf{Q}$ (Equation 18.5-5) with the preconditioning matrix Γ so that the preconditioned system in conservation form becomes

$$\Gamma \frac{\partial}{\partial t}\int_V \mathbf{Q}\, dV + \oint [\mathbf{F} - \mathbf{G}] \cdot d\mathbf{A} = \int_V \mathbf{H}\, dV \tag{18.5-6}$$

where

$$\Gamma = \begin{bmatrix} \Theta & 0 & 0 & 0 & \rho_T \\ \Theta u & \rho & 0 & 0 & \rho_T u \\ \Theta v & 0 & \rho & 0 & \rho_T u \\ \Theta w & 0 & 0 & \rho & \rho_T u \\ \Theta H - \delta & \rho u & \rho v & \rho w & \rho_T H + \rho C_p \end{bmatrix} \tag{18.5-7}$$

The parameter Θ is given by

$$\Theta = \left(\frac{1}{U_r^2} - \frac{\rho_T}{\rho C_p} \right) \tag{18.5-8}$$

The reference velocity U_r appearing in Equation 18.5-8 is chosen locally such that the eigenvalues of the system remain well conditioned with respect to the convective and diffusive time scales [372].

The resultant eigenvalues of the preconditioned system (Equation 18.5-6) are given by

$$u,\ u,\ u,\ u' + c',\ u' - c' \tag{18.5-9}$$

where

$$\begin{aligned} u &= \mathbf{v} \cdot \hat{n} \\ u' &= u\,(1 - \alpha) \\ c' &= \sqrt{\alpha^2 u^2 + U_r^2} \end{aligned}$$

$$\begin{aligned}\alpha &= \left(1-\beta U_r^2\right)/2 \\ \beta &= \left(\rho_p + \frac{\rho_T}{\rho C_p}\right)\end{aligned}$$

For an ideal gas, $\beta = (\gamma RT)^{-1} = 1/c^2$. Thus, when $U_r = c$ (at sonic speeds and above), $\alpha = 0$ and the eigenvalues of the preconditioned system take their traditional form, $u \pm c$. At low speed, however, as $U_r \rightarrow 0$, $\alpha \rightarrow 1/2$ and all eigenvalues become of the same order as u. For constant-density flows, $\beta = 0$ and $\alpha = 1/2$ regardless of the values of U_r. As long as the reference velocity is of the same order as the local velocity, all eigenvalues remain of the order u. Thus, the eigenvalues of the preconditioned system remain well conditioned at all speeds.

Note that the non-preconditioned Navier-Stokes equations are recovered exactly from Equation 18.5-6 by setting $1/U_r^2$ to ρ_p, the derivative of density with respect to pressure. In this case Γ reduces exactly to the Jacobian $\partial\mathbf{W}/\partial\mathbf{Q}$.

Although Equation 18.5-6 is conservative in the steady state, it is not, in a strict sense, conservative for time-dependent flows. This is not a problem, however, since the preconditioning has already destroyed the time accuracy of the equations and we will not employ them in this form for unsteady calculations.

For unsteady calculations, an unsteady preconditioning is available when the dual-time stepping method is used (Section 18.5.5: Implicit Time Stepping (Dual-Time formulation)). The unsteady preconditioning enhances the solution accuracy by improving the scaling of the artificial dissipation and maximizes the efficiency by optimizing the number of sub-iterations required at each physical time step [263]. For low Mach number flows in particular, for both low frequency problems (large time steps) and high frequency problems (small time step), significant savings in computational time are possible when compared with the non-preconditioned case.

The unsteady preconditioning adapts the level of preconditioning based on the user specified time-step and on the local advective and acoustic time scales of the flow. For acoustic problems, the physical time-step size is small as it is based on the acoustic CFL number. In this case the preconditioning parameter U_r^2 will approach c^2, which in effect will turn off the low-Mach preconditioning almost completely. For advection dominated problems, like the transport of turbulent vortical structures, etc., the physical time-step is large as it is based on the particle CFL number. The corresponding unsteady preconditioning parameter U_r^2 will then approach u^2, which corresponds to the steady preconditioning choice. For intermediate physical time-step sizes, the unsteady preconditioning parameter will be adapted to provide optimum convergence efficiency of the pseudo-time iterations and accurate scaling of the artificial dissipation terms, regardless of the choice of the physical time step.

18.5.3 Convective Fluxes

Roe Flux-Difference Splitting Scheme

The inviscid flux vector $\mathbf{F}$ appearing in Equation 18.5-6 is evaluated by a standard upwind, flux-difference splitting [296]. This approach acknowledges that the flux vector $\mathbf{F}$ contains characteristic information propagating through the domain with speed and direction according to the eigenvalues of the system. By splitting $\mathbf{F}$ into parts, where each part contains information traveling in a particular direction (i.e., characteristic information), and upwind differencing the split fluxes in a manner consistent with their corresponding eigenvalues, we obtain the following expression for the discrete flux at each face:

$$\mathbf{F} = \frac{1}{2}\left(\mathbf{F}_R + \mathbf{F}_L\right) - \frac{1}{2}\Gamma\,|\hat{\mathrm{A}}|\,\delta\mathbf{Q} \qquad (18.5\text{-}10)$$

Here $\delta\mathbf{Q}$ is the spatial difference $\mathbf{Q}_R - \mathbf{Q}_L$. The fluxes $\mathbf{F}_R = \mathbf{F}\left(\mathbf{Q}_R\right)$ and $\mathbf{F}_L = \mathbf{F}\left(\mathbf{Q}_L\right)$ are computed using the (reconstructed) solution vectors $\mathbf{Q}_R$ and $\mathbf{Q}_L$ on the "right" and "left" side of the face. The matrix $|\hat{\mathrm{A}}|$ is defined by

$$|\hat{\mathrm{A}}| = M\,|\Lambda|\,M^{-1} \qquad (18.5\text{-}11)$$

where Λ is the diagonal matrix of eigenvalues and M is the modal matrix that diagonalizes $\Gamma^{-1}\mathrm{A}$, where A is the inviscid flux Jacobian $\partial\mathbf{F}/\partial\mathbf{Q}$.

For the non-preconditioned system (and an ideal gas) Equation 18.5-10 reduces to Roe's flux-difference splitting [296] when Roe-averaged values are used to evaluate $\Gamma\,|\hat{\mathrm{A}}|$. At present, arithmetic averaging of states $\mathbf{Q}_R$ and $\mathbf{Q}_L$ is used.

In its current form, Equation 18.5-10 can be viewed as a second-order central difference plus an added matrix dissipation. The added matrix dissipation term is not only responsible for producing an upwinding of the convected variables, and of pressure and flux velocity in supersonic flow, but it also provides the pressure-velocity coupling required for stability and efficient convergence of low-speed and incompressible flows.

AUSM+ Scheme

An alternative way to compute the flux vector **F** appearing in Equation 18.5-6 is by using a flux-vector splitting scheme [53]. The scheme, called Advection Upstream Splitting Method (AUSM), was first introduced by Liou and Steffen in 1993 [202]. The AUSM scheme first computes a cell interface Mach number based on the characteristic speeds from the neighboring cells. The interface Mach number is then used to determine the upwind extrapolation for the convection part of the inviscid fluxes. A separate Mach number splitting is used for the pressure terms. Generalized Mach number based convection and pressure splitting functions were proposed by Liou [201] and the new scheme was termed AUSM+. The AUSM+ scheme has several desirable properties:

1. Provides exact resolution of contact and shock discontinuities
2. Preserves positivity of scalar quantities
3. Free of oscillations at stationary and moving shocks

The AUSM+ scheme avoids using an explicit artificial dissipation, by proposing a numerical flux of the form:

$$F = m_f \phi + p_i \tag{18.5-12}$$

Here m_f is the mass flux through the interface, which is computed using the fourth order polynomial functions of the left and right side (of the interface) Mach numbers.

ANSYS FLUENT utilizes an all-speed version of the AUSM+ scheme based on the low-Mach preconditioning.

Low Diffusion Roe Flux Difference Splitting Scheme

In order to reduce dissipation in LES calculations, ANSYS FLUENT uses a modified version of the Roe Flux Difference Splitting scheme, called the Low Diffusion Roe Flux Difference Splitting scheme. The scheme includes low Mach number preconditioning, in which the artificial dissipation term has been reduced [43] through the use of a hybrid scheme that combines a central scheme and a second-order upwind scheme (Roe's Flux Difference scheme).

i The low diffusion discretization must be used only for subsonic flows. For high Mach number flows, you should switch to the second-order upwind scheme.

i The low diffusion discretization is only available with the implicit-time formulation (dual-time-stepping). When running LES with the explicit time formulation, you will need to use the second-order upwind scheme.

To learn how to apply the different convective fluxes, see Section 26.4.2: Convective Flux Types in the separate User's Guide.

18.5.4 Steady-State Flow Solution Methods

The coupled set of governing equations (Equation 18.5-6) in ANSYS FLUENT is discretized in time for both steady and unsteady calculations. In the steady case, it is assumed that time marching proceeds until a steady-state solution is reached. Temporal discretization of the coupled equations is accomplished by either an implicit or an explicit time-marching algorithm. These two algorithms are described below. To learn how to apply the two formulations, see Section 26.4: Density-Based Solver Settings in the separate User's Guide.

Explicit Formulation

In the explicit scheme a multi-stage, time-stepping algorithm [146] is used to discretize the time derivative in Equation 18.5-6. The solution is advanced from iteration n to iteration $n+1$ with an m-stage Runge-Kutta scheme, given by

$$\begin{aligned}
\mathbf{Q}^0 &= \mathbf{Q}^n \\
\Delta\mathbf{Q}^i &= -\alpha_i \Delta t \Gamma^{-1} \mathbf{R}^{i-1} \\
\mathbf{Q}^{n+1} &= \mathbf{Q}^m
\end{aligned}$$

where $\Delta\mathbf{Q}^i \equiv \mathbf{Q}^i - \mathbf{Q}^n$ and $i = 1, 2, \ldots, m$ is the stage counter for the m-stage scheme. α_i is the multi-stage coefficient for the i^{th} stage. The residual $\mathbf{R}^i$ is computed from the intermediate solution $\mathbf{Q}^i$ and, for Equation 18.5-6, is given by

$$\mathbf{R}^i = \sum^{N_{\rm faces}} \Big(\mathbf{F}(\mathbf{Q}^i) - \mathbf{G}(\mathbf{Q}^i)\Big) \cdot \mathbf{A} - V\mathbf{H} \tag{18.5-13}$$

The time step Δt is computed from the CFL (Courant-Friedrichs-Lewy) condition

$$\Delta t = \frac{2\mathrm{CFL} \cdot V}{\sum_f {\lambda_{\rm f}}^{max} A_f} \tag{18.5-14}$$

where V is the cell volume, A_f is the face area, and ${\lambda_{\rm f}}^{max}$ is the maximum of the local eigenvalues defined by Equation 18.5-9.

For steady-state solutions, convergence acceleration of the explicit formulation can be achieved with the use of local time stepping, residual smoothing, and full-approximation storage multigrid.

Local time stepping is a method by which the solution at each control volume is advanced in time with respect to the cell time step, defined by the local stability limit of the time-stepping scheme.

Residual smoothing, on the other hand, increases the bound of stability limits of the time-stepping scheme and hence allows for the use of a larger CFL value to achieve fast convergence (Section 18.5.4: Implicit Residual Smoothing).

The convergence rate of the explicit scheme can be accelerated through use of the full-approximation storage (FAS) multigrid method described in Section 18.6.4: Full-Approximation Storage (FAS) Multigrid.

By default, ANSYS FLUENT uses a 3-stage Runge-Kutta scheme based on the work by Lynn [211] for steady-state flows that use the density-based explicit solver.

Implicit Residual Smoothing

The maximum time step can be further increased by increasing the support of the scheme through implicit averaging of the residuals with their neighbors. The residuals are filtered through a Laplacian smoothing operator:

$$\bar{R}_i = R_i + \epsilon \sum (\bar{R}_j - \bar{R}_i) \tag{18.5-15}$$

This equation can be solved with the following Jacobi iteration:

$$\bar{R}_i^m = \frac{R_i + \epsilon \sum \bar{R}_j^{m-1}}{1 + \epsilon \sum 1} \tag{18.5-16}$$

Two Jacobi iterations are usually sufficient to allow doubling the time step with a value of $\epsilon = 0.5$.

Implicit Formulation

In the implicit scheme, an Euler implicit discretization in time of the governing equations (Equation 18.5-6) is combined with a Newton-type linearization of the fluxes to produce the following linearized system in delta form [370]:

$$\left[\mathrm{D} + \sum_{j}^{N_{\text{faces}}} \mathrm{S}_{j,k} \right] \Delta \mathbf{Q}^{n+1} = -\mathbf{R}^n \tag{18.5-17}$$

The center and off-diagonal coefficient matrices, D and $\mathrm{S}_{j,k}$ are given by

$$\mathrm{D} = \frac{V}{\Delta t}\Gamma + \sum_{j}^{N_{\text{faces}}} \mathrm{S}_{j,i} \tag{18.5-18}$$

$$\mathrm{S}_{j,k} = \left(\frac{\partial \mathbf{F}_j}{\partial \mathbf{Q}_k} - \frac{\partial \mathbf{G}_j}{\partial \mathbf{Q}_k}\right) A_j \tag{18.5-19}$$

and the residual vector $\mathbf{R}^n$ and time step Δt are defined as in Equation 18.5-13 and Equation 18.5-14, respectively.

Equation 18.5-17 is solved using either Incomplete Lower Upper factorization (ILU) by default or symmetric point Gauss-Seidel algorithm, in conjunction with an algebraic multigrid (AMG) method (see Section 18.6.3: Algebraic Multigrid (AMG)) adapted for coupled sets of equations.

Explicit relaxation can improve the convergence to steady state of the implicit formulation. By default, explicit relaxation is enabled for the implicit solver and uses a factor of 0.75. You can specify a factor α to control the amount that the solution vector Q changes between iterations after the end of the algebraic multigrid (AMG) cycle:

$$Q_{\text{new}} = Q_{\text{old}} + \alpha \Delta Q \tag{18.5-20}$$

By specifying a value less than the default value of 1 for α, the variables in the solution vector will be under-relaxed and the convergence history can be improved. For information on how to set this value, see Section 26.4.3: Specifying the Explicit Relaxation in the separate User's Guide.

Note that explicit relaxation is available for the density-based implicit solver in steady state mode only.

18.5.5 Unsteady Flows Solution Methods

For time-accurate calculations, explicit and implicit time-stepping schemes are available. (The time-implicit approach is also referred to as "dual time stepping".)

Explicit Time Stepping

The explicit time stepping approach, is available only for the explicit scheme described above. The time step is determined by the CFL condition. To maintain time accuracy of the solution the explicit time stepping employs the same time step in each cell of the domain (this is also known as global-time step), and with preconditioning disabled. By default, ANSYS FLUENT uses a 4-stage Runge-Kutta scheme for unsteady flows.

Implicit Time Stepping (Dual-Time formulation)

The implicit-time stepping method (also known as dual-time formulation) is available in the density-based explicit and implicit formulation.

When performing unsteady simulations with implicit-time stepping (dual-time stepping), ANSYS FLUENT uses a low Mach number time-derivative unsteady preconditioner to provide accurate results both for pure convective processes (e.g., simulating unsteady turbulence) and for acoustic processes (e.g., simulating wave propagation) [350, 263].

Here we introduce a preconditioned pseudo-time-derivative term into Equation 18.5-1 as follows:

$$\frac{\partial}{\partial t}\int_V \mathbf{W}\,dV + \Gamma\frac{\partial}{\partial \tau}\int_V \mathbf{Q}\,dV + \oint [\mathbf{F}-\mathbf{G}]\cdot d\mathbf{A} = \int_V \mathbf{H}\,dV \qquad (18.5\text{-}21)$$

where t denotes physical-time and τ is a pseudo-time used in the time-marching procedure. Note that as $\tau \to \infty$, the second term on the left side of Equation 18.5-21 vanishes and Equation 18.5-1 is recovered.

The time-dependent term in Equation 18.5-21 is discretized in an implicit fashion by means of either a first- or second-order accurate, backward difference in time.

The dual-time formulation is written in semi-discrete form as follows:

$$\begin{aligned}\left[\frac{\Gamma}{\Delta\tau} + \frac{\epsilon_0}{\Delta t}\frac{\partial \mathbf{W}}{\partial \mathbf{Q}}\right]\Delta\mathbf{Q}^{k+1} \quad &+ \quad \frac{1}{V}\oint [\mathbf{F}-\mathbf{G}]\cdot d\mathbf{A} \\ = \mathbf{H} \quad &- \quad \frac{1}{\Delta t}\left(\epsilon_0\mathbf{W}^k - \epsilon_1\mathbf{W}^n + \epsilon_2\mathbf{W}^{n-1}\right)\end{aligned}$$

where $\{\epsilon_0 = \epsilon_1 = 1/2, \epsilon_2 = 0\}$ gives first-order time accuracy, and $\{\epsilon_0 = 3/2, \epsilon_1 = 2, \epsilon_2 = 1/2\}$ gives second-order. k is the inner iteration counter and n represents any given physical-time level.

The pseudo-time-derivative is driven to zero at each physical time level by a series of inner iterations using either the implicit or explicit time-marching algorithm.

Throughout the (inner) iterations in pseudo-time, $\mathbf{W}^n$ and $\mathbf{W}^{n-1}$ are held constant and $\mathbf{W}^k$ is computed from $\mathbf{Q}^k$. As $\tau \to \infty$, the solution at the next physical time level $\mathbf{W}^{n+1}$ is given by $\mathbf{W}(\mathbf{Q}^k)$.

Note that the physical time step Δt is limited only by the level of desired temporal accuracy. The pseudo-time-step $\Delta\tau$ is determined by the CFL condition of the (implicit or explicit) time-marching scheme.

Table 18.5.1 summarizes all operation modes for the density-based solver from the iterative scheme in steady-state calculations to time-marching schemes for transient calculations.

Table 18.5.1: Summary of the Density-Based Solver

Solution Method	**Density-Based Solver - Explicit Formulation**	**Density-Based Solver - Implicit Formulation**
Steady-State	– 3-stages Runge-Kutta – local time step – time-derivative preconditioning – FAS	– local time step – time-derivative preconditioning
Unsteady - Explicit Time Stepping	– 4-stages Runge-Kutta – global time step – no time-derivative preconditioning – No FAS	N/A
Unsteady - Implicit Time Stepping (dual-time formulation) **First Order**	– dual-time formulation – Physical time: first order Euler backward – preconditioned pseudo-time derivative – inner iteration: explicit pseudo-time marching, 3-stage Runge-Kutta	– dual-time formulation – Physical time: first order Euler backward – preconditioned pseudo-time derivative – inner iteration: implicit pseudo-time marching
Unsteady - Implicit Time Stepping (dual-time formulation) **Second Order**	– dual-time formulation – Physical time: second order Euler backward – preconditioned pseudo-time derivative – inner iteration: explicit pseudo-time marching, 3-stage Runge-Kutta	– dual-time formulation – Physical time: second order Euler backward – preconditioned pseudo-time derivative – inner iteration: implicit pseudo-time marching

18.6 Multigrid Method

The ANSYS FLUENT solver contains two forms of multigrid: algebraic (AMG) and full-approximation storage (FAS). AMG is an essential component of both the pressure-based and density-based implicit solvers, while FAS is an important, but optional, component of the density-based explicit solver. (Note that when the density-based explicit solver is used, AMG will also be used, since the scalar equations (e.g., turbulence) are solved using the approach described in Section 18.2: General Scalar Transport Equation: Discretization and Solution.)

This section describes the mathematical basis of the multigrid approach. Common aspects of AMG and FAS are presented first, followed by separate sections that provide details unique to each method.

Information is organized into the following subsections:

- Section 18.6.1: Approach
- Section 18.6.2: Multigrid Cycles
- Section 18.6.3: Algebraic Multigrid (AMG)
- Section 18.6.4: Full-Approximation Storage (FAS) Multigrid

For information about user inputs and controls for the multigrid solver, see
Section 26.18.3: Modifying Algebraic Multigrid Parameters and Section 26.5.2: Setting FAS Multigrid Parameters in the separate User's Guide.

18.6.1 Approach

ANSYS FLUENT uses a multigrid scheme to accelerate the convergence of the solver by computing corrections on a series of coarse grid levels. The use of this multigrid scheme can greatly reduce the number of iterations and the CPU time required to obtain a converged solution, particularly when your model contains a large number of control volumes.

The Need for Multigrid

Implicit solution of the linearized equations on unstructured meshes is complicated by the fact that there is no equivalent of the line-iterative methods that are commonly used on structured meshes. Since direct matrix inversion is out of the question for realistic problems and "whole-field" solvers that rely on conjugate-gradient (CG) methods have robustness problems associated with them, the methods of choice are point implicit solvers like Gauss-Seidel and ILU. Although the Gauss-Seidel and ILU schemes rapidly remove local (high-frequency) errors in the solution, global (low-frequency) errors are reduced at a rate inversely related to the mesh size. Thus, for a large number of nodes, the solver "stalls" and the residual reduction rate becomes prohibitively low.

The multi-stage scheme used in the density-based explicit solver can efficiently remove local (high-frequency) errors as well. That is, the effect of the solution in one cell is communicated to adjacent cells relatively quickly. However, the scheme is less effective at reducing global (low-frequency) errors–errors which exist over a large number of control volumes. Thus, global corrections to the solution across a large number of control volumes occur slowly, over many iterations. This implies that performance of the multi-stage scheme will deteriorate as the number of control volumes increases.

Multigrid techniques allow global error to be addressed by using a sequence of successively coarser meshes. This method is based upon the principle that global (low-frequency) error existing on a fine mesh can be represented on a coarse mesh where it again becomes accessible as local (high-frequency) error: because there are fewer coarse cells overall, the global corrections can be communicated more quickly between adjacent cells. Since computations can be performed at an exponentially decaying expense in both CPU time and memory storage on coarser meshes, there is the potential for very efficient elimination of global error. The fine-grid relaxation scheme or "smoother", in this case either the point-implicit linear solvers (Section 18.6.3: The Coupled and Scalar AMG Solvers) or the explicit multi-stage scheme, is not required to be particularly effective at reducing global error and can be tuned for efficient reduction of local error.

The Basic Concept in Multigrid

Consider the set of discretized linear (or linearized) equations given by

$$A\,\phi_e + b = 0 \tag{18.6-1}$$

where ϕ_e is the exact solution. Before the solution has converged there will be a defect d associated with the approximate solution ϕ:

$$A\,\phi + b = d \tag{18.6-2}$$

We seek a correction ψ to ϕ such that the exact solution is given by

$$\phi_e = \phi + \psi \tag{18.6-3}$$

Substituting Equation 18.6-3 into Equation 18.6-1 gives

$$\begin{aligned} A\,(\phi + \psi) + b &= 0 && (18.6\text{-}4) \\ A\,\psi + (A\,\phi + b) &= 0 && (18.6\text{-}5) \end{aligned}$$

Now using Equations 18.6-2 and 18.6-5 we obtain

$$A\,\psi + d = 0 \tag{18.6-6}$$

which is an equation for the correction in terms of the original fine level operator A and the defect d. Assuming the local (high-frequency) errors have been sufficiently damped by the relaxation scheme on the fine level, the correction ψ will be smooth and therefore more effectively solved on the next coarser level.

Restriction and Prolongation

Solving for corrections on the coarse level requires transferring the defect down from the fine level (restriction), computing corrections, and then transferring the corrections back up from the coarse level (prolongation). We can write the equations for coarse level corrections ψ^H as

$$A^H\,\psi^H + R\,d = 0 \tag{18.6-7}$$

where A^H is the coarse level operator and R the restriction operator responsible for transferring the fine level defect down to the coarse level. Solution of Equation 18.6-7 is followed by an update of the fine level solution given by

$$\phi^{\mathrm{new}} = \phi + P\,\psi^H \tag{18.6-8}$$

where P is the prolongation operator used to transfer the coarse level corrections up to the fine level.

Unstructured Multigrid

The primary difficulty with using multigrid on unstructured meshes is the creation and use of the coarse grid hierarchy. On a structured mesh, the coarse meshes can be formed simply by removing every other mesh line from the fine meshes and the prolongation and restriction operators are simple to formulate (e.g., injection and bilinear interpolation).

The difficulties of applying multigrid on unstructured meshes are overcome in a separate fashion by each of the two multigrid methods used in ANSYS FLUENT. While the basic principles discussed so far and the cycling strategy described in Section 18.6.2: Multigrid Cycles are the same, the techniques for construction of restriction, prolongation, and coarse mesh operators are different, as discussed in Section 18.6.3: Algebraic Multigrid (AMG) and Section 18.6.4: Full-Approximation Storage (FAS) Multigrid for the AMG and FAS methods, respectively.

18.6.2 Multigrid Cycles

A multigrid cycle can be defined as a recursive procedure that is applied at each grid level as it moves through the grid hierarchy. Four types of multigrid cycles are available in ANSYS FLUENT: the V, W, F, and flexible (“flex”) cycles. The V and W cycles are available in both AMG and FAS, while the F and flexible cycles are restricted to the AMG method only. (The W and flexible AMG cycles are not available for solving the coupled equation set due to the amount of computation required.)

The V and W Cycles

Figures 18.6.1 and 18.6.2 show the V and W multigrid cycles (defined below). In each figure, the multigrid cycle is represented by a square, and then expanded recursively to show the individual steps that are performed within the cycle. The individual steps are represented by a circle, one or more squares, and a triangle, connected by lines: circle-square-triangle for a V cycle, or circle-square-square-triangle for a W cycle. The squares in this group expand again, into circle-square-triangle or circle-square-square-triangle, and so on. You may want to follow along in the figures as you read the steps below.

For the V and W cycles, the traversal of the hierarchy is governed by three parameters, β_1, β_2, and β_3, as follows:

1. First, iterations are performed on the current grid level to reduce the high-frequency components of the error (local error). For AMG, one iteration consists of one forward and one backward Gauss-Seidel sweep. For FAS, one iteration consists of one pass of the multi-stage scheme (described in Section 18.5.4: Explicit Formulation).

   These iterations are referred to as pre-relaxation sweeps because they are performed before moving to the next coarser grid level. The number of pre-relaxation sweeps is specified by β_1.

   In Figures 18.6.1 and 18.6.2 this step is represented by a circle and marks the start of a multigrid cycle. The high-wave-number components of error should be reduced until the remaining error is expressible on the next coarser mesh without significant aliasing.

   If this is the coarsest grid level, then the multigrid cycle on this level is complete. (In Figures 18.6.1 and 18.6.2 there are 3 coarse grid levels, so the square representing the multigrid cycle on level 3 is equivalent to a circle, as shown in the final diagram in each figure.)

   i In the AMG method, the default value of β_1 is zero (i.e., no pre-relaxation sweeps are performed).

2. Next, the problem is “restricted” to the next coarser grid level using Equation 18.6-7.

   In Figures 18.6.1 and 18.6.2, the restriction from a finer grid level to a coarser grid level is designated by a downward-sloping line.

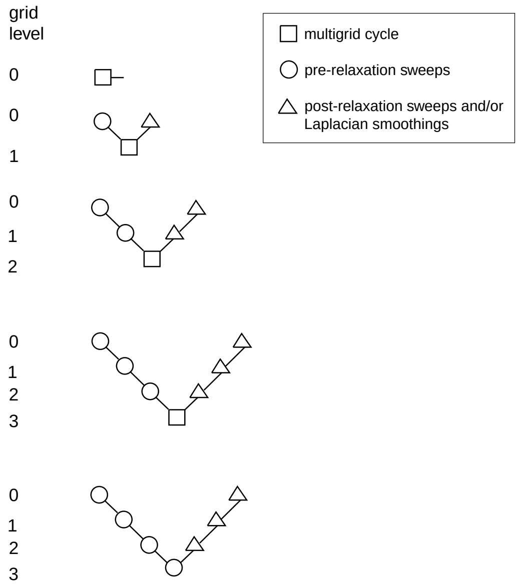


Figure 18.6.1: V-Cycle Multigrid

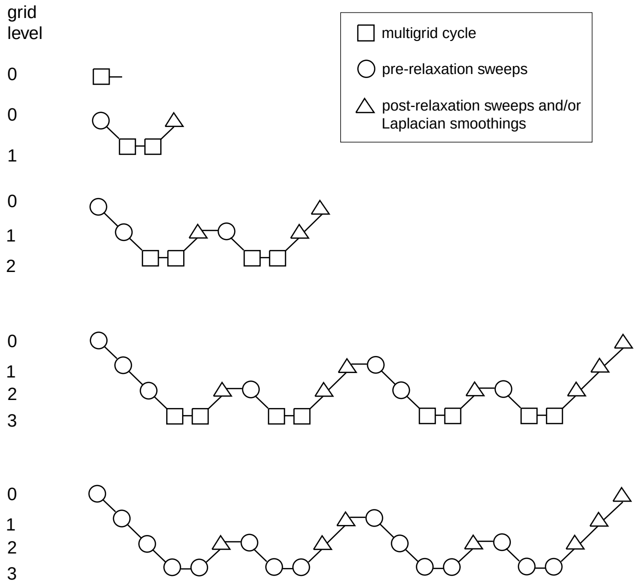


Figure 18.6.2: W-Cycle Multigrid

3. The error on the coarse grid is reduced by performing a specified number (β_2) of multigrid cycles (represented in Figures 18.6.1 and 18.6.2 as squares). Commonly, for fixed multigrid strategies β_2 is either 1 or 2, corresponding to V-cycle and W-cycle multigrid, respectively.

4. Next, the cumulative correction computed on the coarse grid is "interpolated" back to the fine grid using Equation 18.6-8 and added to the fine grid solution. In the FAS method, the corrections are additionally smoothed during this step using the Laplacian smoothing operator discussed in Section 18.5.4: Implicit Residual Smoothing.

   In Figures 18.6.1 and 18.6.2 the prolongation is represented by an upward-sloping line.

   The high-frequency error now present at the fine grid level is due to the prolongation procedure used to transfer the correction.

5. In the final step, iterations are performed on the fine grid to remove the high-frequency error introduced on the coarse grid by the multigrid cycles. These iterations are referred to as post-relaxation sweeps because they are performed after returning from the next coarser grid level. The number of post-relaxation sweeps is specified by β_3.

   In Figures 18.6.1 and 18.6.2, this relaxation procedure is represented by a single triangle.

   For AMG, the default value of β_3 is 1.

   i Note, however, that if you are using AMG with V-cycle to solve an energy equation with a solid conduction model presented with anisotropic or very high conductivity coefficient, there is a possibility of divergence with a default post-relaxation sweep of 1. In such cases you should increase the post-relaxation sweep (e.g., to 2) in the AMG section for better convergence, or change the cycle type to F-cycle or W-cycle, with an under-relaxation factor set to 1. This is especially effective when calculating pure heat conduction or conjugate heat transfer. Any instability observed when using the F-cycle or W-cycle can be remedied by increasing the pre-relaxation sweep count to 1. Although the default value of 0 is optimal for most cases, increasing the pre-relaxation sweep value to 1 or 2 can improve convergence.

   Since the default value for β_1 is 0 (i.e., pre-relaxation sweeps are not performed),this procedure is roughly equivalent to using the solution from the coarse level as the initial guess for the solution at the fine level.

For FAS, the default value of β_3 is zero (i.e., post-relaxation sweeps are not performed); post-relaxation sweeps are never performed at the end of the cycle for the finest grid level, regardless of the value of β_3. This is because for FAS, post-relaxation sweeps at the fine level are equivalent to pre-relaxation sweeps during the next cycle.

18.6.3 Algebraic Multigrid (AMG)

This algorithm is referred to as an "algebraic" multigrid scheme because, as we shall see, the coarse level equations are generated without the use of any geometry or re-discretization on the coarse levels; a feature that makes AMG particularly attractive for use on unstructured meshes. The advantage being that no coarse meshes have to be constructed or stored, and no fluxes or source terms need to be evaluated on the coarse levels. This approach is in contrast with FAS (sometimes called "geometric") multigrid in which a hierarchy of meshes is required and the discretized equations are evaluated on every level. In theory, the advantage of FAS over AMG is that the former should perform better for non-linear problems since non-linearities in the system are carried down to the coarse levels through the re-discretization; when using AMG, once the system is linearized, non-linearities are not "felt" by the solver until the fine level operator is next updated.

AMG Restriction and Prolongation Operators

The restriction and prolongation operators used here are based on the additive correction (AC) strategy described for structured meshes by Hutchinson and Raithby [136]. Inter-level transfer is accomplished by piecewise constant interpolation and prolongation. The defect in any coarse level cell is given by the sum of those from the fine level cells it contains, while fine level corrections are obtained by injection of coarse level values. In this manner the prolongation operator is given by the transpose of the restriction operator

$$P = R^T \tag{18.6-9}$$

The restriction operator is defined by a coarsening or "grouping" of fine level cells into coarse level ones. In this process each fine level cell is grouped with one or more of its "strongest" neighbors, with a preference given to currently ungrouped neighbors. The algorithm attempts to collect cells into groups of fixed size, typically two or four, but any number can be specified. In the context of grouping, strongest refers to the neighbor j of the current cell i for which the coefficient A_{ij} is largest. For sets of coupled equations A_{ij} is a block matrix and the measure of its magnitude is simply taken to be the magnitude of its first element. In addition, the set of coupled equations for a given cell are treated together and not divided amongst different coarse cells. This results in the same coarsening for each equation in the system.

AMG Coarse Level Operator

The coarse level operator A^H is constructed using a Galerkin approach. Here we require that the defect associated with the corrected fine level solution must vanish when transferred back to the coarse level. Therefore we may write

$$R\, d^{\text{new}} = 0 \tag{18.6-10}$$

Upon substituting Equations 18.6-2 and 18.6-8 for d^{new} and ϕ^{new} we have

$$\begin{aligned} R\left[A\,\phi^{\text{new}} + b\right] &= 0 \\ R\left[A\left(\phi + P\,\psi^H\right) + b\right] &= 0 \end{aligned} \tag{18.6-11}$$

Now rearranging and using Equation 18.6-2 once again gives

$$\begin{aligned} R\,A\,P\,\psi^H + R\,(A\,\phi + b) &= 0 \\ R\,A\,P\,\psi^H + R\,d &= 0 \end{aligned} \tag{18.6-12}$$

Comparison of Equation 18.6-12 with Equation 18.6-7 leads to the following expression for the coarse level operator:

$$A^H = R\,A\,P \tag{18.6-13}$$

The construction of coarse level operators thus reduces to a summation of diagonal and corresponding off-diagonal blocks for all fine level cells within a group to form the diagonal block of that group's coarse cell.

The F Cycle

The multigrid F cycle is essentially a combination of the V and W cycles described in Section 18.6.2: The V and W Cycles.

Recall that the multigrid cycle is a recursive procedure. The procedure is expanded to the next coarsest grid level by performing a single multigrid cycle on the current level. Referring to Figures 18.6.1 and 18.6.2, this means replacing the square on the current level (representing a single cycle) with the procedure shown for the 0-1 level cycle (the second diagram in each figure). We see that a V cycle consists of:

pre sweep → restrict → V cycle → prolongate → post sweep

and a W cycle:

pre sweep → restrict → W cycle → W cycle → prolongate → post sweep

An F cycle is formed by a W cycle followed by a V cycle:

pre sweep → restrict → W cycle → V cycle → prolongate → post sweep

As expected, the F cycle requires more computation than the V cycle, but less than the W cycle. However, its convergence properties turn out to be better than the V cycle and roughly equivalent to the W cycle. The F cycle is the default AMG cycle type for the coupled equation set.

The Flexible Cycle

For the flexible cycle, the calculation and use of coarse grid corrections is controlled in the multigrid procedure by the logic illustrated in Figure 18.6.3. This logic ensures that coarser grid calculations are invoked when the rate of residual reduction on the current grid level is too slow. In addition, the multigrid controls dictate when the iterative solution of the correction on the current coarse grid level is sufficiently converged and should thus be applied to the solution on the next finer grid. These two decisions are controlled by the parameters α and β shown in Figure 18.6.3, as described in detail below. Note that the logic of the multigrid procedure is such that grid levels may be visited repeatedly during a single global iteration on an equation. For a set of 4 multigrid levels, referred to as 0, 1, 2, and 3, the flex-cycle multigrid procedure for solving a given transport equation might consist of visiting grid levels as 0-1-2-3-2-3-2-1-0-1-2-1-0, for example.

The main difference between the flexible cycle and the V and W cycles is that the satisfaction of the residual reduction tolerance and termination criterion determine when and how often each level is visited in the flexible cycle, whereas in the V and W cycles the traversal pattern is explicitly defined.

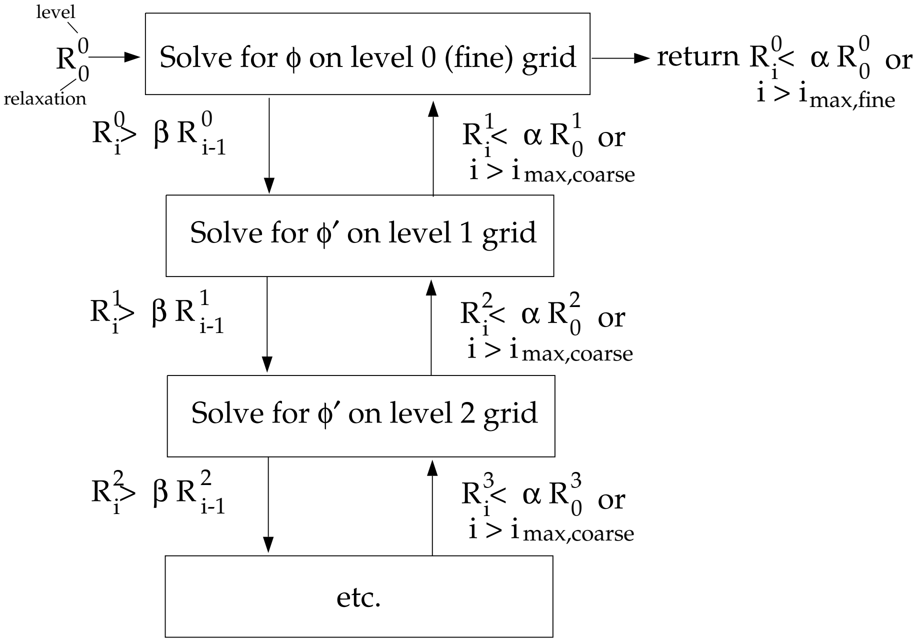

level
relaxation
R_0^0
Solve for φ on level 0 (fine) grid
return $R_i^0 < \alpha R_0^0$ or $i > i_{max,fine}$
$R_i^0 > \beta R_{i-1}^0$
$R_i^1 < \alpha R_0^1$ or $i > i_{max,coarse}$
Solve for φ′ on level 1 grid
$R_i^1 > \beta R_{i-1}^1$
$R_i^2 < \alpha R_0^2$ or $i > i_{max,coarse}$
Solve for φ′ on level 2 grid
$R_i^2 > \beta R_{i-1}^2$
$R_i^3 < \alpha R_0^3$ or $i > i_{max,coarse}$
etc.

Figure 18.6.3: Logic Controlling the Flex Multigrid Cycle

The Residual Reduction Rate Criteria

The multigrid procedure invokes calculations on the next coarser grid level when the error reduction rate on the current level is insufficient, as defined by

$$R_i > \beta R_{i-1} \tag{18.6-14}$$

Here R_i is the absolute sum of residuals (defect) computed on the current grid level after the ith relaxation on this level. The above equation states that if the residual present in the iterative solution after i relaxations is greater than some fraction, β (between 0 and 1), of the residual present after the $(i-1)$th relaxation, the next coarser grid level should be visited. Thus β is referred to as the residual reduction tolerance, and determines when to “give up” on the iterative solution at the current grid level and move to solving the correction equations on the next coarser grid. The value of β controls the frequency with which coarser grid levels are visited. The default value is 0.7. A larger value will result in less frequent visits, and a smaller value will result in more frequent visits.

The Termination Criteria

Provided that the residual reduction rate is sufficiently rapid, the correction equations will be converged on the current grid level and the result applied to the solution field on the next finer grid level.

The correction equations on the current grid level are considered sufficiently converged when the error in the correction solution is reduced to some fraction, α (between 0 and 1), of the original error on this grid level:

$$R_i < \alpha R_0 \tag{18.6-15}$$

Here, R_i is the residual on the current grid level after the ith iteration on this level, and R_0 is the residual that was initially obtained on this grid level at the current global iteration. The parameter α, referred to as the termination criterion, has a default value of 0.1. Note that the above equation is also used to terminate calculations on the lowest (finest) grid level during the multigrid procedure. Thus, relaxations are continued on each grid level (including the finest grid level) until the criterion of this equation is obeyed (or until a maximum number of relaxations has been completed, in the case that the specified criterion is never achieved).

The Coupled and Scalar AMG Solvers

The scalar AMG solver is used for the solution of linear systems obtained from the discretization of the individual transport equations.

$$a_{ij} x_j = b_i \tag{18.6-16}$$

where the above equation contains scalar variables.

The coupled AMG solver is used to solve linear transport equations using implicit discretization from coupled systems such as flow variables for the density-based solver, pressure-velocity variables for the coupled pressure-based schemes and inter-phase coupled individual equations for Eulerian multiphase flows.

$$[A]_{ij} \vec{X}_j = \vec{B}_i \tag{18.6-17}$$

where the influence of a cell i on a cell j has the form

$$A_{ij} = \begin{bmatrix} a_{ij}^{11} & a_{ij}^{12} & \ldots & a_{ij}^{1N} \\ a_{ij}^{21} & . & & . \\ \vdots & & . & . \\ a_{ij}^{N1} & & & a_{ij}^{NN} \end{bmatrix} \tag{18.6-18}$$

and the unknown and source vectors have the form

$$\vec{X}_j = \begin{bmatrix} x_j^1 \\ . \\ . \\ x_j^N \end{bmatrix} \tag{18.6-19}$$

$$\vec{B}_i = \begin{bmatrix} b_i^1 \\ . \\ . \\ b_i^N \end{bmatrix} \tag{18.6-20}$$

The above resultant system of equations is solved in ANSYS FLUENT using either the Gauss-Seidel smoother or the Incomplete Lower Upper decomposition (ILU) smoother. If a scalar system of equations is to be solved then the point-method (Gauss-Seidel or ILU) smoother is used, while for a coupled system of equations the block-method (Gauss-Seidel or ILU) smoother is used.

Gauss-Seidel

The Gauss-Seidel method is a technique for solving a linear system of equations one at a time and in sequence. It uses the previously computed results as soon as they become available. It performs two sweeps on the unknowns in forward and backward directions. Both point or block method Gauss-Seidel smoothers are available in ANSYS FLUENT to solve for either the scalar AMG system of equations or the coupled AMG system of equations.

The Gauss-Seidel procedure can be illustrated using the scalar system, Equation 18.6-16. The forward sweep can be written as:

$$x_i{}^{k+1/2} = (b_i - \sum_{j<i} a_{ij} x_j{}^{k+1/2} - \sum_{j>i} a_{ij} x_j{}^{k})/a_{ii} \tag{18.6-21}$$

$$(i = 1, ..., N)$$

where N is the number of unknowns. The forward sweep is followed by a backward sweep which can be written as:

$$x_i{}^{k+1} = (b_i - \sum_{j<i} a_{ij} x_j{}^{k+1/2} - \sum_{j>i} a_{ij} x_j{}^{k+1})/a_{ii} \tag{18.6-22}$$

Following from Equations 18.6-21 and 18.6-22, symmetric Gauss-Seidel can be expressed in matrix form as a two-step recursive solution of the system

$$(D_A + L_A){D_A}^{-1}(D_A + U_A)(x^{k+1} - x^k) = b - Ax^k \tag{18.6-23}$$

where D_A, L_A, and U_A represent diagonal, lower tridiagonal, and upper tridiagonal parts of matrix A, respectively.

Symmetric Gauss-Seidel has a somewhat limited rate of smoothing of residuals between levels of AMG, unless the coarsening factor is set to 2.

Incomplete Lower Upper (ILU)

A more effective AMG smoother is based on the ILU decomposition technique. In general, any iterational method can be represented as

$$M(x^{k+1} - x^k) = b - Ax^k \tag{18.6-24}$$

where matrix M is some approximation of the original matrix A from

$$Ax = b \tag{18.6-25}$$

M should be close to A and the calculation of M^{-1} should have a low operation count. We consider M as an incomplete lower upper factorization of the matrix A such that

$$M = LU = (D + L_A)D^{-1}(D + U_A) \tag{18.6-26}$$

where L_A and U_A are the lower tridiagonal and upper tridiagonal parts of matrix A. The diagonal matrix D is calculated in a special way to satisfy the following condition for diagonal D_M of matrix M:

$$D_M = D_A \tag{18.6-27}$$

In this case, the ith element of the diagonal of D will be calculated using

$$d_{ii} = a_{ii} - \sum_{j<1}\left(\frac{a_{ij}a_{ji}}{d_{jj}}\right) \tag{18.6-28}$$

The calculation of the new solution x^{k+1} is then performed in two symmetric recursive sweeps, similar to Gauss-Seidel sweeps. Diagonal elements d_{ii} of the ILU decomposition are calculated during the construction of levels and stored in the memory. ILU smoother is slightly more expensive compared to Gauss-Seidel, but has better smoothing properties, especially for block-coupled systems solved by coupled AMG. In this case, coarsening of levels can be more aggressive using coarsening factors between 8 and 12 for 3D problems compared to 2 for Gauss-Seidel.

i When solving the coupled systems, shorter solution times and more robust performance can be obtained by using the default ILU smoother, rather than the Gauss-Seidel smoother, which is the default for scalar systems. ILU is recommended whenever the coupled AMG solver is used.

18.6.4 Full-Approximation Storage (FAS) Multigrid

ANSYS FLUENT's approach to forming the multigrid grid hierarchy for FAS is simply to coalesce groups of cells on the finer grid to form coarse grid cells. Coarse grid cells are created by agglomerating the cells surrounding a node, as shown in Figure 18.6.4. Depending on the grid topology, this can result in cells with irregular shapes and variable numbers of faces. The grid levels are, however, simple to construct and are embedded, resulting in simple prolongation and relaxation operators.

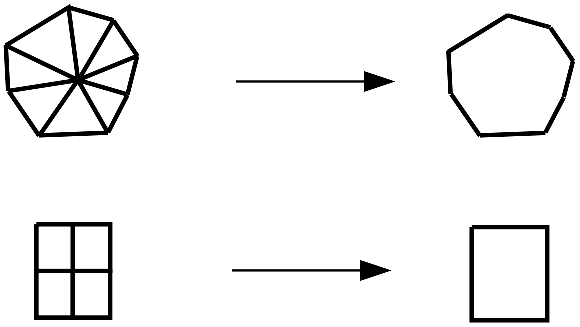


Figure 18.6.4: Node Agglomeration to Form Coarse Grid Cells

It is interesting to note that although the coarse grid cells look very irregular, the discretization cannot "see" the jaggedness in the cell faces. The discretization uses only the area projections of the cell faces and therefore each group of "jagged" cell faces separating two irregularly-shaped cells is equivalent to a single straight line (in 2D) connecting the endpoints of the jagged segment. (In 3D, the area projections form an irregular, but continuous, geometrical shape.) This optimization decreases the memory requirement and the computation time.

FAS Restriction and Prolongation Operators

FAS requires restriction of both the fine grid solution ϕ and its residual (defect) d. The restriction operator R used to transfer the solution to the next coarser grid level is formed using a full-approximation scheme [35]. That is, the solution for a coarse cell is obtained by taking the volume average of the solution values in the embedded fine grid cells. Residuals for the coarse grid cell are obtained by summing the residuals in the embedded fine grid cells.

The prolongation operator P used to transfer corrections up to the fine level is constructed to simply set the fine grid correction to the associated coarse grid value.

The coarse grid corrections ψ^H, which are brought up from the coarse level and applied to the fine level solution, are computed from the difference between the solution calculated on the coarse level ϕ^H and the initial solution restricted down to the coarse level $R\phi$. Thus correction of the fine level solution becomes

$$\phi^{\text{new}} = \phi + P\left(\phi^H - R\phi\right) \tag{18.6-29}$$

FAS Coarse Level Operator

The FAS coarse grid operator A^H is simply that which results from a re-discretization of the governing equations on the coarse level mesh. Since the discretized equations presented in Sections 18.3 and 18.5 place no restrictions on the number of faces that make up a cell, there is no problem in performing this re-discretization on the coarse grids composed of irregularly shaped cells.

There is some loss of accuracy when the finite-volume scheme is used on the irregular coarse grid cells, but the accuracy of the multigrid solution is determined solely by the finest grid and is therefore not affected by the coarse grid discretization.

In order to preserve accuracy of the fine grid solution, the coarse level equations are modified to include source terms [145] which insure that corrections computed on the coarse grid ϕ^H will be zero if the residuals on the fine grid d^h are zero as well. Thus, the coarse grid equations are formulated as

$$A^H\phi^H + d^H = d^H(R\phi) - Rd^h \tag{18.6-30}$$

Here d^H is the coarse grid residual computed from the current coarse grid solution ϕ^H, and $d^H(R\phi)$ is the coarse grid residual computed from the restricted fine level solution $R\phi$. Initially, these two terms will be the same (because initially we have $\phi^H = R\phi$) and cancel from the equation, leaving

$$A^H\phi^H = -Rd^h \tag{18.6-31}$$

So there will be no coarse level correction when the fine grid residual d^h is zero.

18.7 Full Multigrid (FMG) Initialization

For many complex flow problems such as those found in rotating machinery, or flows in expanding or spiral ducts, flow convergence can be accelerated if a better initial solution is used at the start of the calculation. The Full Multigrid initialization (FMG initialization) can provide this initial and approximate solution at a minimum cost to the overall computational expense.

For more information about using FMG initialization in ANSYS FLUENT, see Section 26.10: Using Full Multigrid (FMG) Initialization in the separate User's Guide.

18.7.1 Overview of FMG Initialization

FMG initialization utilizes the ANSYS FLUENT FAS Multigrid technology (see Section 18.6.4: Full-Approximation Storage (FAS) Multigrid) to obtain the initial solution. Starting from a uniform solution (after performing standard initialization), the FMG initialization procedure constructs the desirable number of geometric grid levels using the procedure outlined in Section 18.6.4: Full-Approximation Storage (FAS) Multigrid. To begin the process, the initial solution is restricted all the way down to the coarsest level. The FAS multigrid cycle is then applied until a given order of residual reduction is obtained or the maximum number of cycles is reached. The solution is then interpolated one grid level up and the FAS multigrid cycle is applied again between the current level all the way down to the coarsest level. This process will repeat until the finest level is reached. The FMG initialization iteration is illustrated in Figure 18.7.1.

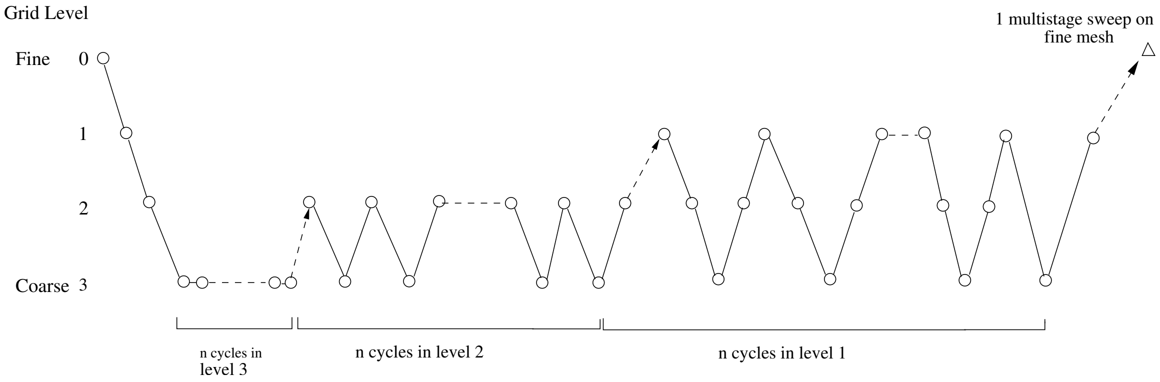


Figure 18.7.1: The FMG Initialization

Since FMG initialization does most of the work on coarse levels, this initialization procedure is computationally inexpensive and, for large problems, a good initial solution can be obtained in a fraction of the time spent to converge on a final solution. Note that FMG initialization can be used with the pressure-based and density-based solvers.

When FMG initialization is started, the algorithm will perform the following steps:

1. Records the current solver selection and all current solver parameters.
2. Switches from the selected solver to the density-based explicit formulation.
3. Performs one FMG iteration using the FMG parameters given in the text command interface (see below).
4. Switches back to the initially selected solver and resets all solver parameters back to the original solver settings.

In the FMG iteration, the inviscid Euler equations are solved using first order-discretization to obtain the approximate solution. If species are present, then the FMG initialization will solve the species equations. However, turbulence equations or any other transport scalars are not solved in the FMG initialization.

18.7.2 Limitations of FMG Initialization

- FMG initialization is not available for unsteady flows.

  i If an initial solution is needed for an unsteady calculation, then you must first switch to the steady state solver, perform FMG initialization and other necessary iterations to get an initial solution, and then perform unsteady calculations.

- FMG will not initialize turbulence or other transport equations field variables.

  i If your are solving for turbulent flow, then you can still use FMG initialization, however the FMG will not initialize the turbulence field variables. For this reason, a good initial guess of the turbulence field quantities are important for achieving fast convergence after the FMG initialization has been performed.

- FMG should not be used with multiphase flow.

Chapter 19. Adapting the Mesh

The solution-adaptive mesh refinement feature of ANSYS FLUENT allows you to refine and/or coarsen your mesh based on geometric and numerical solution data. In addition, ANSYS FLUENT provides tools for creating and viewing adaption fields customized to particular applications. For information about using mesh adaption in ANSYS FLUENT, see Chapter 27: Adapting the Mesh in the separate User's Guide. Theoretical information about the adaption process is described in detail in the following sections.

19.1 Static Adaption Process

The adaption process is separated into two distinct tasks.

1. The individual cells are marked for refinement or coarsening based on the adaption function, which is created from geometric and/or solution data.
2. The cell is refined or considered for coarsening based on these adaption marks.

   The primary advantages of this modularized approach are the abilities to create sophisticated adaption functions and to experiment with various adaption functions without modifying the existing mesh.

i Write a case and data file before starting the adaption process. If you generate an undesirable mesh, you can restart the process with the saved files.

19.1.1 Hanging Node Adaption

Hanging node adaption is the procedure used in ANSYS FLUENT. Meshes produced by this method are characterized by nodes on edges and faces that are not vertices of all the cells sharing those edges or faces, as shown in Figure 19.1.1.

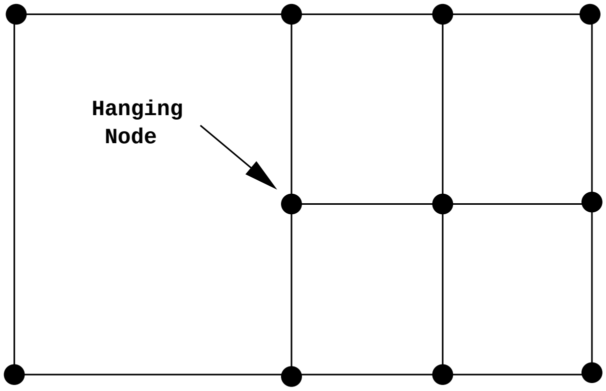


Figure 19.1.1: Example of a Hanging Node

Hanging node mesh adaption provides the ability to operate on meshes with a variety of cell shapes, including hybrid meshes. Although the hanging node scheme provides significant mesh flexibility, it requires additional memory to maintain the mesh hierarchy which is used by the rendering and mesh adaption operations.

Hanging Node Refinement

The cells marked for refinement are divided as described here:

- A triangle is split into 4 triangles.
- A quadrilateral is split into 4 quadrilaterals.
- A tetrahedron is split into eight tetrahedra. The subdivision consists of trimming each corner of the tetrahedron, and subdividing the enclosed octahedron by introducing the shortest diagonal.
- A hexahedron is split into 8 hexahedra.
- A wedge (prism) is split into 8 wedges.
- A pyramid is split into 6 pyramids and 4 tetrahedra.

Figures 19.1.2 and 19.1.3 illustrate the division of the supported cell shapes.

To maintain accuracy, neighboring cells are not allowed to differ by more than one level of refinement. This prevents the adaption from producing excessive cell volume variations (reducing truncation error) and ensures that the positions of the parent (original) and child (refined) cell centroids are similar (reducing errors in the flux evaluations).

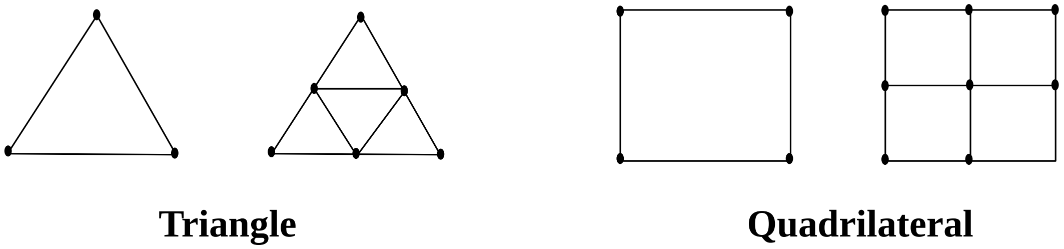
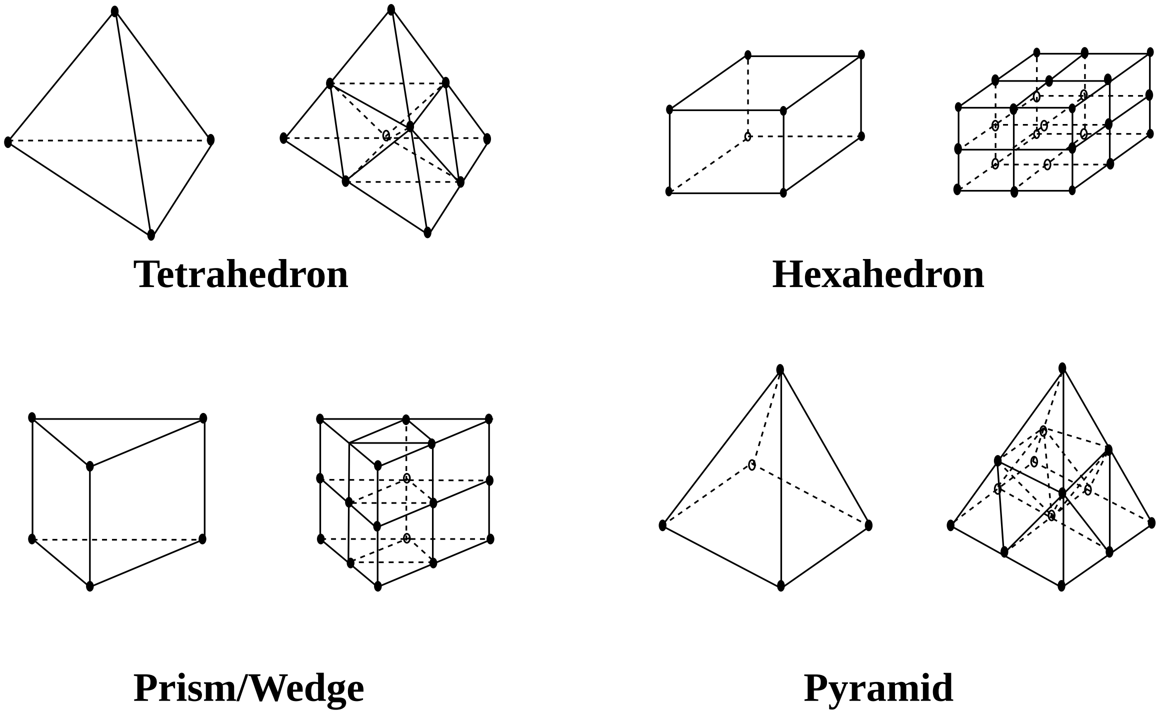


Figure 19.1.2: Hanging Node Adaption of 2D Cell Types

Hanging Node Coarsening

The mesh is coarsened by reintroducing inactive parent cells (uniting the child cells to reclaim the previously subdivided parent cell). An inactive parent cell is reactivated if all its children are marked for coarsening. You will eventually reclaim the original mesh with repeated application of the hanging node coarsening. Using the hanging node adaption process, you cannot coarsen the mesh further than the original mesh.



Tetrahedron

Hexahedron

Prism/Wedge

Pyramid

Figure 19.1.3: Hanging Node Adaption of 3D Cell Types

19.2 Boundary Adaption

If more cells are required on a boundary, they can be added using boundary adaption, which allows you to mark or refine cells in the proximity of the selected boundary zones. The ability to refine the mesh near one or more boundary zones is provided because important fluid interactions often occur in these regions. Example, development of strong velocity gradients in the boundary layer near a wall.

An example of a mesh that can be improved with boundary adaption is shown in Figure 19.2.1. This mesh has only two cells on the vertical face of a step. Boundary adaption on the zone corresponding to the face of the step can be used to increase the number of cells, as shown in Figure 19.2.2. This procedure cannot increase the resolution of a curved surface. Therefore, if more cells are required on a curved surface where the shape of the surface is important, create the mesh with sufficient surface nodes before reading it into the solver.

19.3 Gradient Adaption

The gradient adaption function allows you to mark cells or adapt the mesh based on the gradient, curvature, or isovalue of the selected field variables.

Information can be found in the following subsections:

- Section 19.3.1: Gradient Adaption Approach
- Section 19.3.2: Example of Steady Gradient Adaption

19.3.1 Gradient Adaption Approach

Solution-adaptive mesh refinement is performed to efficiently reduce the numerical error in the digital solution, with minimal numerical cost. Unfortunately, direct error estimation for point-insertion adaption schemes is difficult because of the complexity of accurately estimating and modeling the error in the adapted meshes. A comprehensive mathematically rigorous theory for error estimation and convergence is not yet available for CFD simulations. Assuming that maximum error occurs in high-gradient regions, the readily available physical features of the evolving flow field may be used to drive the mesh adaption process.

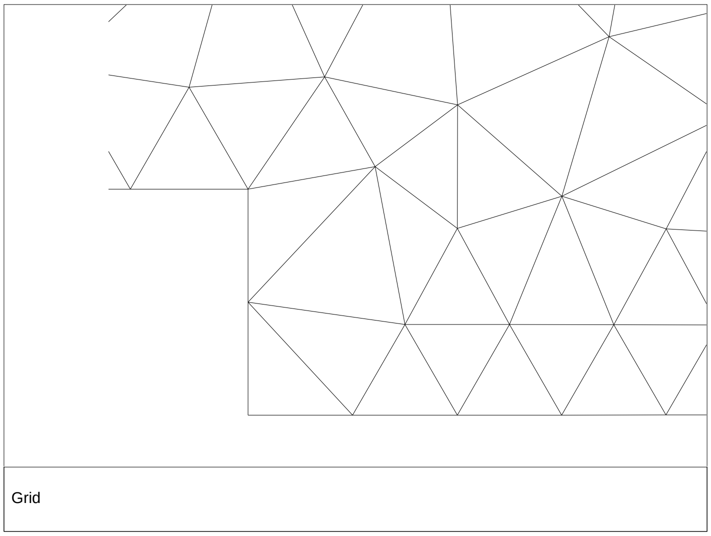


Figure 19.2.1: Mesh Before Adaption

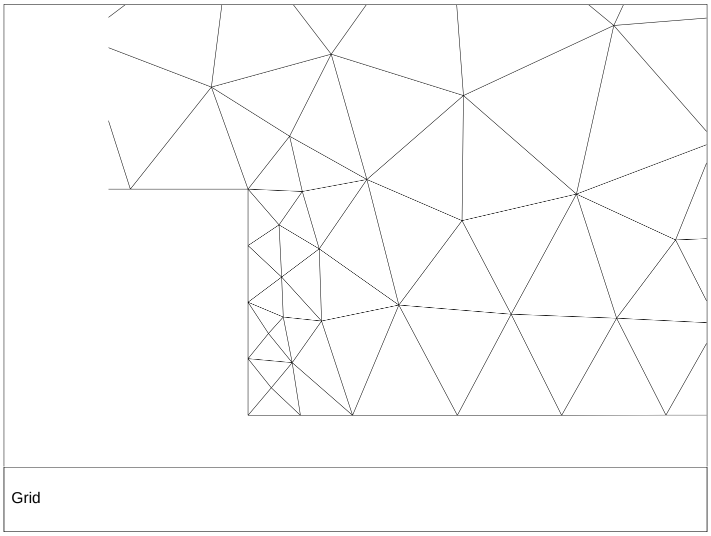


Figure 19.2.2: Mesh after Boundary Adaption

Three approaches for using this information for mesh adaption are available in ANSYS FLUENT:

- **Gradient approach**: In this approach, ANSYS FLUENT multiplies the Euclidean norm of the gradient of the selected solution variable by a characteristic length scale [68]. For example, the gradient function in two dimensions has the following form:

  $$|e_{i1}| = (A_{\text{cell}})^{\frac{r}{2}} |\nabla f| \tag{19.3-1}$$

  where e_{i1} is the error indicator, A_{cell} is the cell area, r is the gradient volume weight, and ∇f is the Euclidean norm of the gradient of the desired field variable, f.

  The default value of the gradient volume weight is unity, which corresponds to full volume weighting. A value of zero will eliminate the volume weighting, and values between 0 and 1 will use proportional weighting of the volume.

  If you specify adaption based on the gradient of a scalar, then the value of $|e_{i1}|$ is displayed when you plot contours of the adaption function.

  This approach is recommended for problems with strong shocks, e.g., supersonic inviscid flows.

- **Curvature approach**: This is the equidistribution adaption technique formerly used by ANSYS FLUENT, that multiplies the undivided Laplacian of the selected solution variable by a characteristic length scale [368].

  For example, the gradient function in two dimensions has the following form:

  $$|e_{i2}| = (A_{\text{cell}})^{\frac{r}{2}} |\nabla^2 f| \tag{19.3-2}$$

  where e_{i2} is the error indicator, A_{cell} is the cell area, r is the gradient volume weight, and $\nabla^2 f$ is the undivided Laplacian of the desired field variable (f).

  The default value of the gradient volume weight is unity, which corresponds to full volume weighting. A value of zero will eliminate the volume weighting, and values between 0 and 1 will use proportional weighting of the volume.

  This approach is recommended for problems with smooth solutions.

- **Isovalue approach**: This approach is not based on derivatives. Instead, the isovalues of the required field variable f, are used to control the adaption. Therefore, the function is of the form:

  $$e_{i3} = f \tag{19.3-3}$$

  where e_{i3} is the error indicator. This approach is recommended for problems where derivatives are not helpful. For example, if you want to refine the mesh where the reaction is taking place, you can use the isovalues of the reaction rate and mark for refinement at high reaction rates. This approach also allows you to customize the criteria for controlling the adaption using custom field functions, user-defined scalars, etc.

The length scale is the square (2D) or cube (3D) root of the cell volume. Introducing the length scale allows resolution of both strong and weak disturbances, increasing the potential for more accurate solutions. However, you can reduce or eliminate the volume weighting by changing the gradient Volume Weight in the **Mesh Adaption Controls** dialog box (see Section 27.12: Mesh Adaption Controls in the separate User's Guide for details).

Any of the field variables available for contouring can be used in the gradient adaption function. These scalar functions include, both geometric and physical features of the numerical solution. Therefore, in addition to traditional adaption to physical features, such as the velocity, you may choose to adapt to the cell volume field to reduce rapid variations in cell volume.

In addition to the Standard (no normalization) approach formerly used by ANSYS FLUENT, two options are available for Normalization [107]:

- Scale, which scales the values of e_{i1}, e_{i2}, or e_{i3} by their average value in the domain, i.e.:

  $$\frac{|e_i|}{\overline{|e_i|}} \tag{19.3-4}$$

  when using the Scale option, suitable first-cut values for the Coarsen Threshold and the Refine Threshold are 0.3 to 0.5, and 0.7 to 0.9, respectively. Smaller values will result in larger adapted regions.

- Normalize, which scales the values of e_{i1}, e_{i2}, or e_{i3} by their maximum value in the domain, therefore always returning a problem-independent range of [0, 1] for any variable used for adaption, i.e.:

  $$\frac{|e_i|}{\max |e_i|} \tag{19.3-5}$$

when using the Normalize option, suitable first-cut values for the Coarsen Threshold and the Refine Threshold are 0.2 to 0.4, and 0.5 to 0.9, respectively. Smaller values will result in larger adapted regions.

19.3.2 Example of Steady Gradient Adaption

An example of the use of steady gradient adaption is the solution of the supersonic flow over a circular cylinder. The initial mesh, shown in Figure 19.3.1, is very coarse, even though it contains sufficient cells to adequately describe the shape of the cylinder. The mesh ahead of the cylinder is too coarse to resolve the shock wave that forms in front of the cylinder. In this instance, pressure is a suitable variable to be used in gradient adaption. This is because there will be a jump in pressure across the shock. However, several adaptions are necessary before the shock can be properly resolved. After several adaptions the mesh will be as shown in Figure 19.3.2.

A typical application of gradient adaption for an incompressible flow might be a mixing layer, which involves a discontinuity.

19.4 Dynamic Gradient Adaption

In contrast with the static gradient adaption (Section 19.3: Gradient Adaption) dynamic gradient adaption is a fully automated process. For time dependent and for steady state problems, you can perform the entire solution without changing the initial settings. That is, you can let the solver periodically perform adaptions without changing/entering any parameter.

19.5 Isovalue Adaption

Some flows may contain flow features that are easy to identify based on values of a certain quantity. For instance, wakes represent a total pressure deficit, and jets are identifiable by a region of relatively high-velocity fluid. Since it is known that these regions also contain large gradients of important flow quantities (such as k and ϵ in turbulent flows), it is convenient to perform an isovalue adaption on the relevant flow quantity than to refine on gradients of the individual flow variables.

The isovalue adaption function allows you to mark or refine cells inside or outside a specified range of a selected field variable function. The mesh can be refined or marked for refinement based on geometric and/or solution vector data. Specifically, any quantity in the display list of field variables can be used for the isovalue adaption. Some examples of how you might use the isovalue marking/adaption feature include the following:

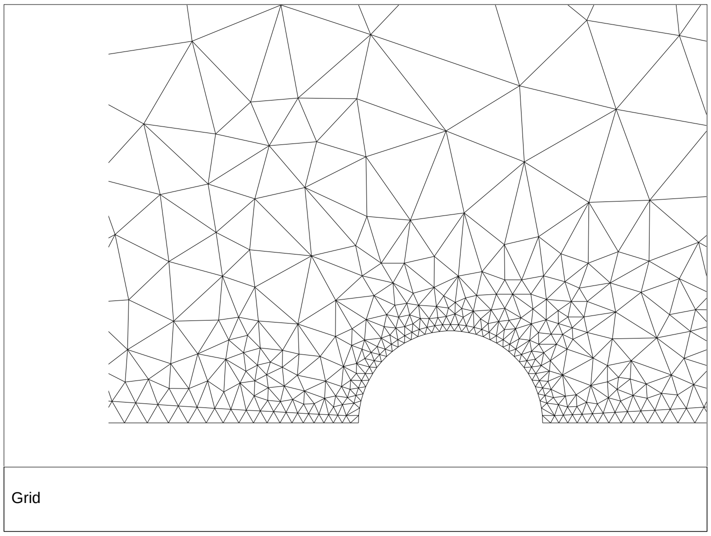


Figure 19.3.1: Bluff-Body Mesh Before Adaption

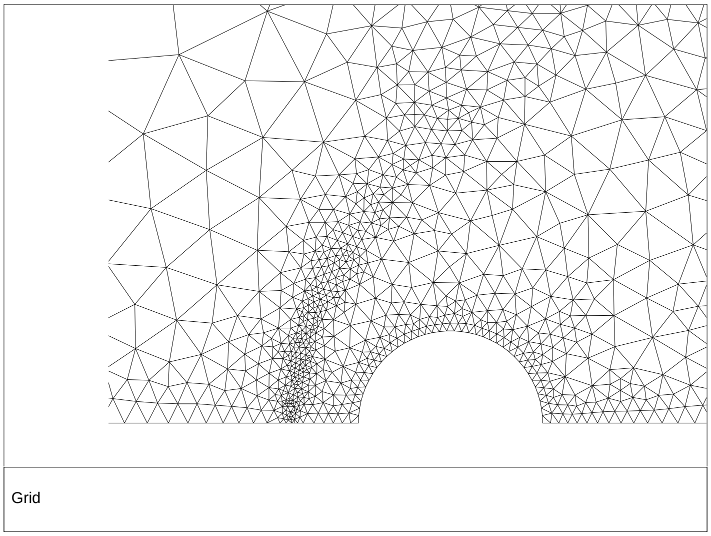


Figure 19.3.2: Bluff-Body Mesh after Gradient Adaption

- Create masks using coordinate values or the quadric function.
- Refine cells that have a velocity magnitude within a specified range.
- Mark and display cells with a pressure or continuity residual outside of a desired range to determine where the numerical solution is changing rapidly.

The approach used in isovalue adaption function is to compute the specified value for each cell (velocity, quadric function, centroid x coordinate, etc.), and then visit each cell, marking for refinement the cells that have values inside (or outside) the specified ranges.

An example of a problem in which isovalue adaption is useful is shown in Figure 19.5.1. The mesh for an impinging jet is displayed along with contours of x velocity. An isovalue adaption based on x velocity allows refinement of the mesh only in the jet (Figure 19.5.2).

Note: *When adapting to isovalues take care to prevent large gradients in cell volume. This can affect accuracy and impede convergence (Section 27.1: Using Adaption in the separate User's Guide). To rectify large gradients in cell volume, adapt to cell-volume change, as demonstrated in Section 19.7.2: Volume Adaption Example.*

19.6 Region Adaption

Many mesh generators create meshes with cell volumes that grow very rapidly with distance from boundaries. While this avoids a dense mesh as a matter of course, it might also create problems if the mesh is not fine enough to resolve the flow. But if it is known a priori that a finer mesh is required in a certain region of the solution domain, the mesh can be refined using region adaption.

The region adaption function marks or refines cells inside or outside a region defined by text or mouse input. Presently, the mesh can be refined or marked inside or outside a hexahedron (quadrilateral in 2D), a sphere (circle in 2D), or a cylinder. The region-based marking/adaption feature is particularly useful for refining regions that intuitively require good resolution: e.g., the wake region of a blunt-body flow field. In addition, you can use the region marking to create mask adaption registers that can be used to limit the extent of the refinement and coarsening.

Information can be found in the following subsections:

- Section 19.6.1: Defining a Region
- Section 19.6.2: Region Adaption Example

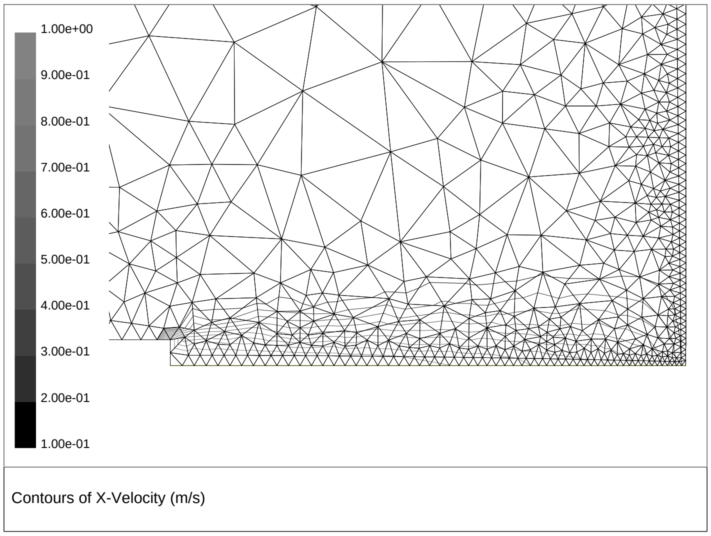


Figure 19.5.1: Impinging Jet Mesh Before Adaption

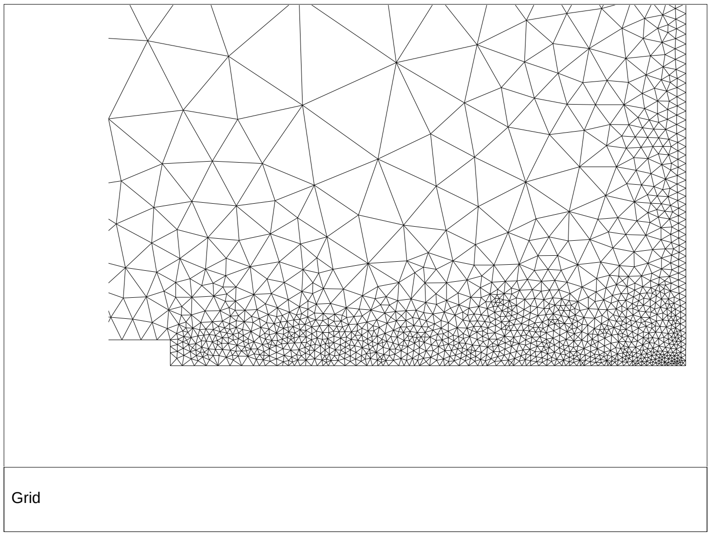


Figure 19.5.2: Impinging Jet Mesh after Isovalue Adaption

19.6.1 Defining a Region

The basic approach to the region adaption function is to first define a region:

- The hexahedron (quadrilateral) is defined by entering the coordinates of two points defining the diagonal.
- The sphere (circle) is defined by entering the coordinates of the center of the sphere and its radius.
- To define a cylinder, specify the coordinates of the points defining the cylinder axis, and the radius. In 3D this will define a cylinder. In 2D, you will have an arbitrarily oriented rectangle with length equal to the cylinder axis length and width equal to the radius.

  A rectangle defined using the cylinder option differs from one defined with the quadrilateral option in that the former can be arbitrarily oriented in the domain while the latter must be aligned with the coordinate axes.

  You can either enter the exact coordinates into the appropriate real entry fields or select the locations with the mouse on displays of the mesh or solution field. After the region is defined, each cell that has a centroid inside/outside the specified region is marked for refinement.

19.6.2 Region Adaption Example

Figure 19.6.1 shows a mesh that was created for solving the flow around a flap airfoil. The mesh is very fine near the surface of the airfoil so that the viscous-affected region may be resolved. However, the mesh grows very rapidly away from the airfoil, because of which the flow separation known to occur on the suction surface of the flap will not be properly predicted. To avoid this problem, the mesh is adapted within circular regions (selected by mouse probe) surrounding the flap. The result is shown in Figure 19.6.2. When the region adaption is performed, the minimum cell volume for adaption is limited (as described in Section 27.12: Mesh Adaption Controls in the separate User's Guide) to prevent the very small cells near the surface from being refined further.

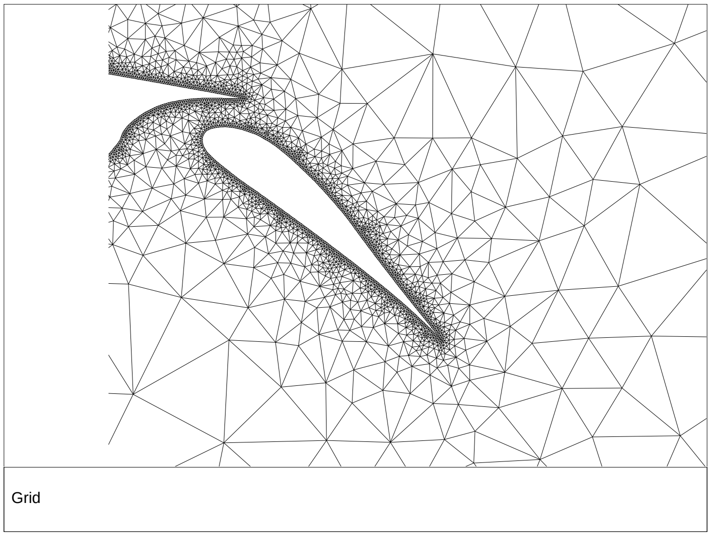


Figure 19.6.1: Flap-Airfoil Mesh Before Adaption

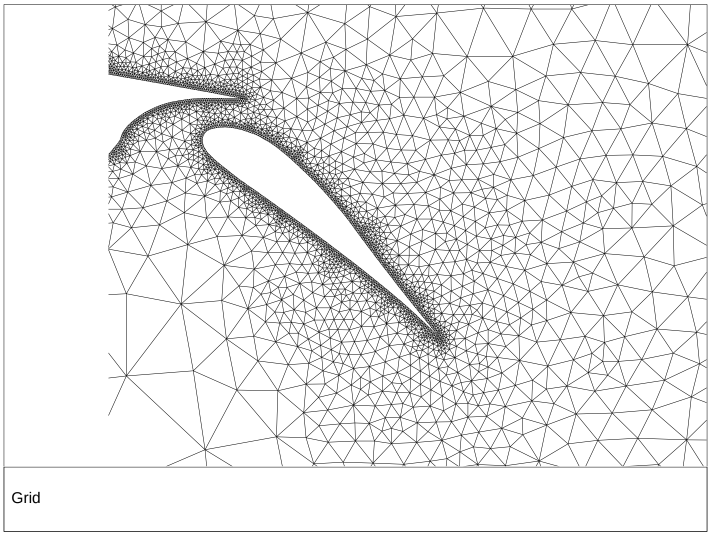


Figure 19.6.2: Flap-Airfoil Mesh after Region Adaption

19.7 Volume Adaption

As mentioned in Section 27.1: Using Adaption in the separate User's Guide, it is best for both accuracy and convergence to have a mesh in which the changes in cell volume are gradual. If the mesh creation or adaption process results in a mesh that does not have this property, the mesh can be improved by using volume adaption with the option of refining, based on either the cell volume or the change in volume between the cell and its neighbors.

Information can be found in the following subsections:

- Section 19.7.1: Volume Adaption Approach
- Section 19.7.2: Volume Adaption Example

19.7.1 Volume Adaption Approach

Marking or refining the mesh based on volume magnitude is often used to remove large cells or to globally refine the mesh. The procedure is to mark for refinement any cell with a volume greater than the specified threshold value.

Marking or refining the mesh based on the change in cell volume is used to improve the smoothness of the mesh. The procedure is to mark for refinement any cell that has a volume change greater than the specified threshold value. The volume change is computed by looping over the faces and comparing the ratio of the cell neighbors to the face.

For example, in Figure 19.7.1 the ratio of V1/V2 and the ratio of V2/V1 is compared to the threshold value. If V2/V1 is greater than the threshold, then C2 is marked for refinement.

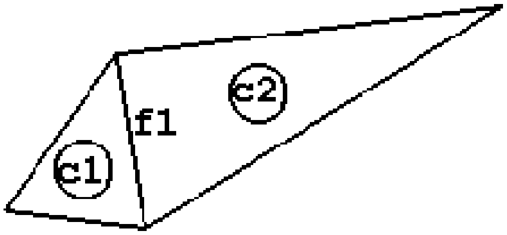


Figure 19.7.1: Volume Change—Ratio of the Volumes of the Cells

19.7.2 Volume Adaption Example

The mesh in Figure 19.7.2 was created for computing a turbulent jet. Local refinement was used in TGrid to create a mesh that is fine in the region of the jet, but coarse elsewhere. This created a very sharp change in cell volume at the edge of the jet.

To improve the mesh, it was refined using volume adaption with the criterion that the maximum cell volume change should be less than 50%. The minimum cell volume for adaption was also limited. The resulting mesh, after smoothing and swapping, is shown in Figure 19.7.3. It can be seen that the interface between the refined region within the jet and the surrounding mesh is no longer as sharp.

19.8 Yplus/Ystar Adaption

ANSYS FLUENT provides three different options for near-wall modeling of turbulence, standard wall functions, nonequilibrium wall functions, and the enhanced wall treatment. As described in Section 12.3: Mesh Considerations for Turbulent Flow Simulations in the separate User's Guide, there are certain mesh requirements for each of these near-wall modeling options.

It is often difficult to gauge the near-wall resolution requirements when creating the mesh. Hence, Yplus and Ystar adaption have been provided to appropriately refine or coarsen the mesh along the wall during the solution process.

19.8.1 Yplus/Ystar Adaption Approach

The approach is to compute y^+ or y^* for boundary cells on the specified viscous wall zones, define the minimum and maximum allowable y^+ or y^*, and mark and/or adapt the appropriate cells. Cells with y^+ or y^* values below the minimum allowable threshold will be marked for coarsening and cells with y^+ or y^* values above the maximum allowable threshold will be marked for refinement (unless coarsening or refinement has been disabled).

Figure 19.8.1 shows the mesh for a duct flow, where the top boundary is the wall and the bottom boundary is the symmetry plane. After an initial solution, it was determined that y^+ values of the cells on the wall boundary were too large, and y^+ adaption was used to refine them. The resulting mesh is shown in Figure 19.8.2. This figure shows that the height of the cells along the wall boundary has been reduced during the refinement process. However, the cell-size distribution on the wall after refinement is much less uniform than in the original mesh, which is an adverse effect of y^+ adaption.

See Section 12.3: Mesh Considerations for Turbulent Flow Simulations in the separate User's Guide for guidelines on recommended values of y^+ or y^* for different near-wall treatments.

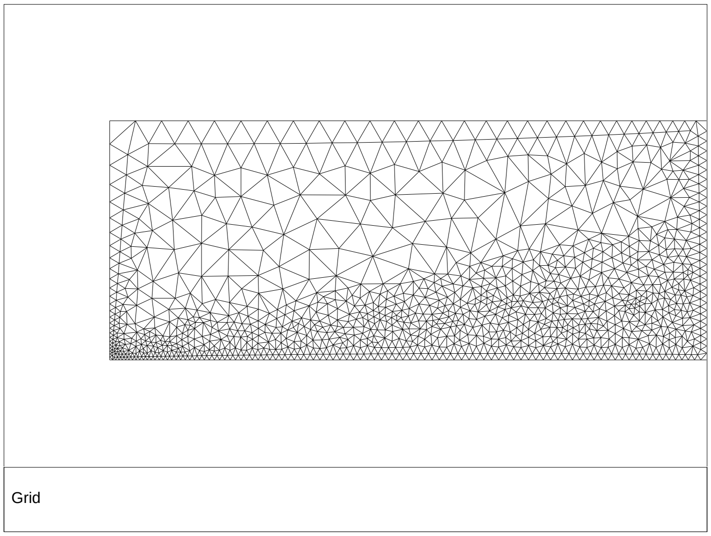


Figure 19.7.2: Jet Mesh Before Adaption

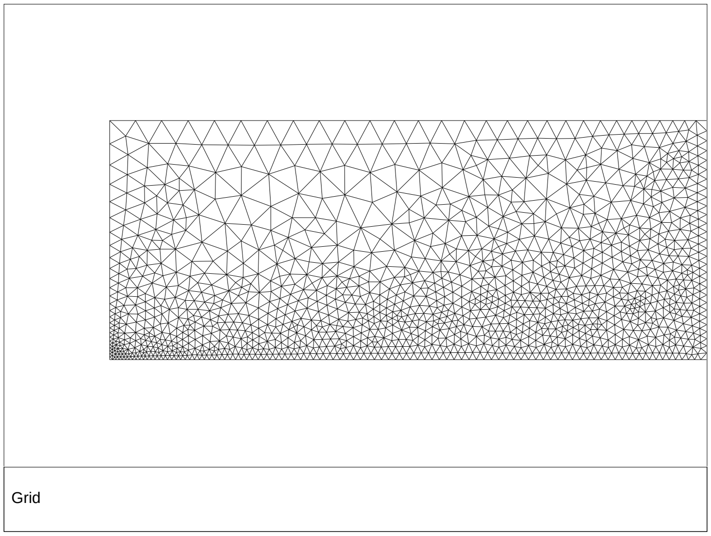


Figure 19.7.3: Jet Mesh after Volume Adaption Based on Change in Cell Volume

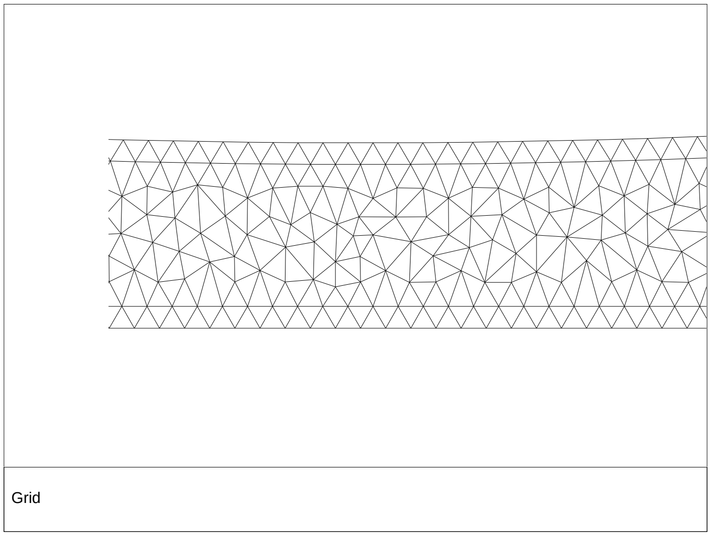


Figure 19.8.1: Duct Flow Mesh Before Adaption

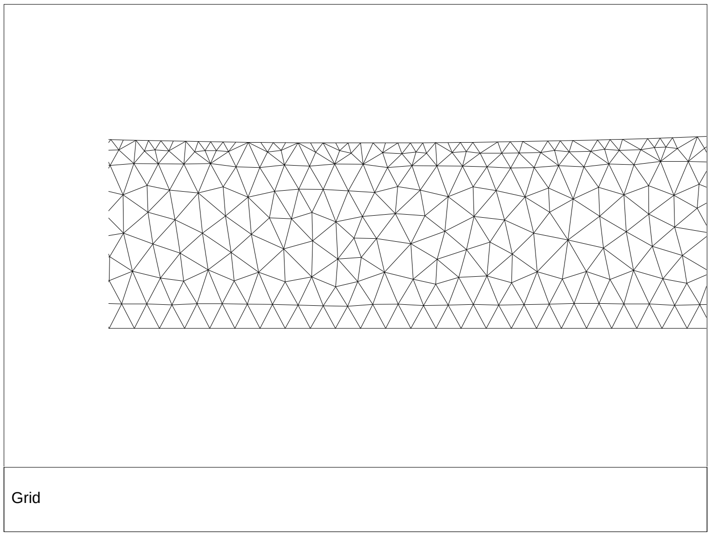


Figure 19.8.2: Duct Flow Mesh after y^+ Adaption

19.9 Anisotropic Adaption

The purpose of anisotropic adaption is to refine hexahedral or prism layer cells in 3D meshes. Anisotropic adaption is considered to be more of a mesh manipulation tool rather than an adaption feature, allowing you to refine some hexahedral or prism cells that are adjacent to one or a few boundary face zones using the GUI. The hexahedral or prism cells are split in one direction each time, giving you control of the different splitting ratios, thus achieving anisotropic refinement.

Note that anisotropic adaption is different from other mesh adaption features because it only refines the mesh and cannot be coarsened after refinement.

For information on how to use anisotropic adaption, see Section 27.9: Anisotropic Adaption in the separate User's Guide

19.10 Geometry-Based Adaption

The purpose of adaption is to produce a mesh that is fine enough and adequately represents all important features of the geometry. However, when you have a coarse mesh of a geometry that has curved profiles and sharp corners, the adapted mesh may not recover the curved profiles and corners at the perimeter of the geometry. In such cases, use geometry-based adaption to reconstruct the geometry (or to recover the finer details of the geometry at its extends) along with performing the adaption process.

19.10.1 Geometry-Based Adaption Approach

Geometry-based adaption works on the principle of *geometry reconstruction*. In this approach, the cell count of the mesh is increased by creating the new nodes in the domain in between the existing nodes of the mesh. The newly created nodes are projected in such a way that the resulting mesh is finer and it's shape is closer to the original geometry.

The following sections explain how nodes are projected and the parameters that control the node propagation.

Node Projection

Consider a coarse mesh created for a circular geometry. A section of the mesh close to the circular edge is shown in Figure 19.10.1. The edge is not smooth and has sharp corners, because of which its shape is not closer to that of the original geometry. Using boundary adaption along with the geometry reconstruction option will result in a mesh with smoother edges as shown in Figure 19.10.2.

In Figure 19.10.2, the dotted lines represent the original edge of the mesh. The boundary adaption process creates new nodes in between the original nodes. These nodes are projected towards the edge of the geometry, because of which the resulting mesh has smooth edges and its shape is closer to the original geometry.

i Only the nodes created in the adaption process (newly created nodes) are projected and the original nodes retain their positions.

The following parameters control node projection and are specified in **Geometry Based Adaption** dialog box.

- Levels of Projection Propagation: This parameter allows you to specify the number of node layers across which node propagation should take place for geometry reconstruction. A value of 1 means only the nodes at the boundary will be projected, a value of 2 means the nodes at the boundary and the nodes in the next layer will be projected, and so on.

  Note: *The nodes in the first level are projected by a maximum magnitude and the node in the last level are projected by a minimum magnitude. The magnitude of projection decreases gradually from the first level to the last level.*

  For example, a value of 3 *for* Levels of Projection Propagation *means, the level 1 node is projected by maximum magnitude and level 3 node is projected by minimum magnitude. Figure 19.10.3 illustrates the level of propagation and magnitude of projection of newly created nodes.*

- Direction of Projection: This parameter allows you to specify the direction, X, Y, or Z (for 3D), for node projection. If you do not specify any direction, the node projection takes place at the nearest point of the newly created node.

- Background Mesh: This option allows you to use a fine surface mesh as a background mesh, based on which the geometry is reconstructed. When you read the surface mesh, the node projection will take place based on the node positions of the background mesh.

  This option is useful when the mesh you want to adapt is very coarse and geometry is highly curved. In such cases, node projection, only by specifying the parameters may not result in a good quality mesh. However, you can also modify the propagation criteria by specifying the parameters.

  i You can read only one surface mesh at a time. The various zones of the surface mesh will be listed in the Background Mesh drop-down list.

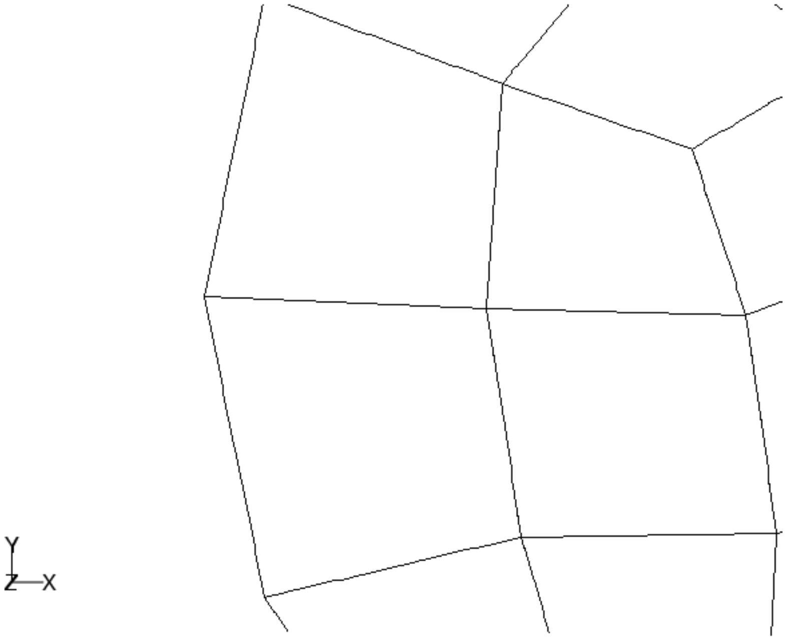


Figure 19.10.1: Mesh Before Adaption

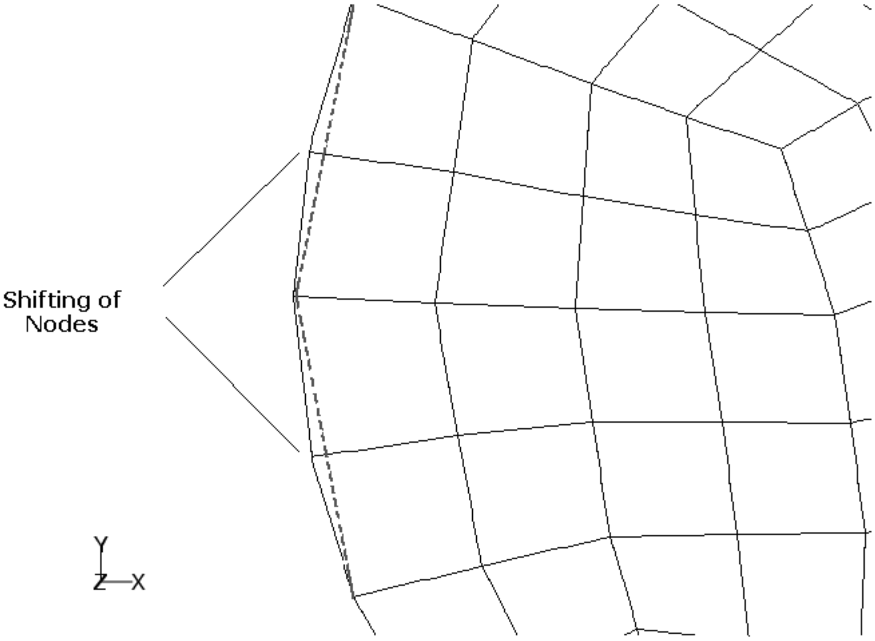


Figure 19.10.2: Projection of Nodes

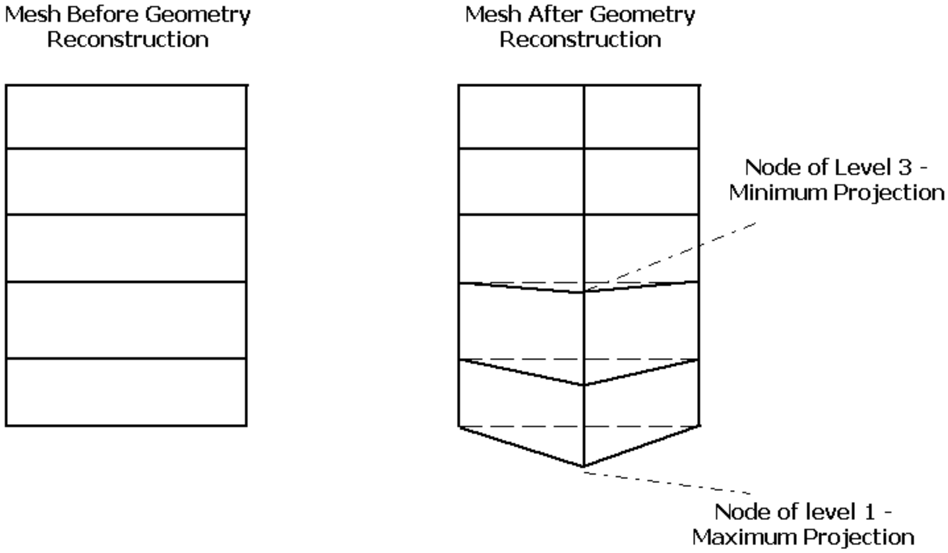

Mesh Before Geometry Reconstruction
Mesh After Geometry Reconstruction
Node of Level 3 - Minimum Projection
Node of level 1 - Maximum Projection

Figure 19.10.3: Levels Projection Propagation and Magnitude

Example of Geometry-Based Adaption

Consider a mesh created for a spherical geometry. The initial mesh is very coarse, because of which it has sharp corners (as in Figure 19.10.4). It does not represent the spherical geometry accurately. To recover the original spherical geometry from this coarse mesh use geometry-based adaption.

If you adapt boundaries of the domain without activating the Reconstruct Geometry option, the resulting mesh (see Figure 19.10.5) has sufficient number of cells, but the boundary of the domain still contains sharp corners.

Boundary adaption only creates new nodes in between the existing nodes to increase the cell count of the mesh. Since it does not project the nodes, the shape of the mesh remains as it is.

If you adapt the boundary with Reconstruct Geometry option. The resulting mesh (Figure 19.10.6) has more number of cells and less sharp corners at boundary. In addition, the newly created nodes are projected in a direction such that it's shape is closer to the original geometry (i.e., sphere with smooth boundary).

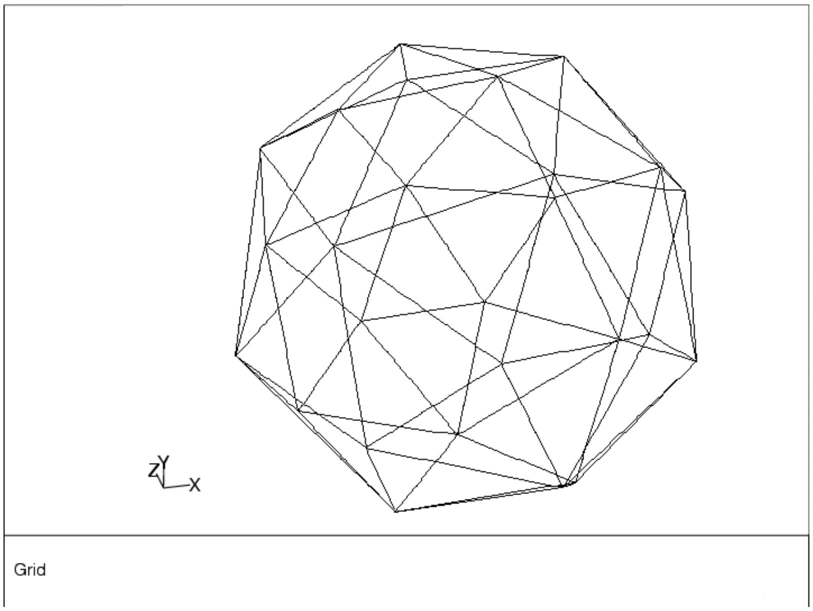


Figure 19.10.4: Coarse Mesh of a Sphere

19.11 Registers

A register is the group of cells that are marked for refinement/coarsening but not adapted.

There are two types of registers:

- Adaption Register
- Mask Register

Adaption Register

An adaption register is basically a list of identifiers for each cell in the domain. The identifiers designate whether a cell is neutral (not marked), marked for refinement, or marked for coarsening. Invoking the Mark command creates an adaption register. It is called a *register* because it is used in a manner similar to the way memory registers are used in calculators. For example, one adaption register holds the result of an operation, another register holds the results of a second operation, and these registers can be used to produce a third register.

The adaption function is used to set the appropriate identifier. For example, to refine the cells based on pressure gradient, the solver computes the gradient adaption function for each cell. The cell value is compared with the refining and coarsening threshold values and assigned the appropriate identifier.

- If the cell value < coarsen threshold value, the cell is marked for coarsening.
- If the coarsen threshold value < cell value < refine threshold value, the cell is neutral (not marked).
- If the cell value > refine threshold value, the cell is marked for refinement.

Adaption registers can be created using geometric data, physical features of the flow field, and combinations of these information. After they are created, the adaption registers can be listed, displayed, deleted, combined, exchanged, inverted, and changed to mask registers.

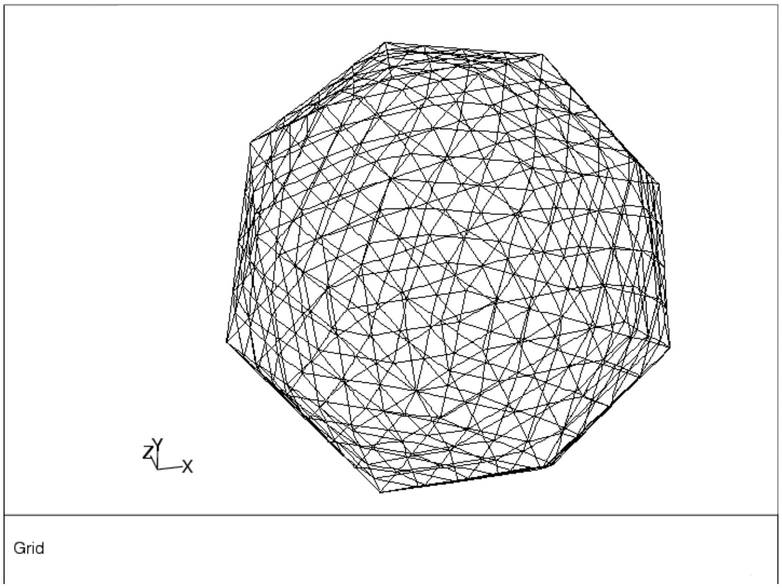


Figure 19.10.5: Adapted Mesh Without Geometry Reconstruction

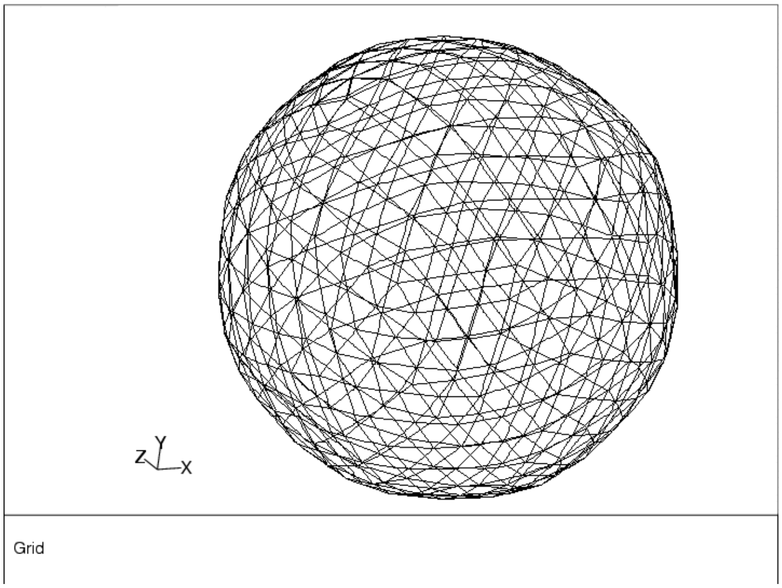


Figure 19.10.6: Mesh after Geometry-Based Adaption

Hybrid Adaption Functions

The hybrid adaption functions are created to confine the adaption to a specific region (using masks) and/or create a more accurate error indicator. ANSYS FLUENT provides a few basic tools to aid in creating hybrid adaption functions.

1. Create the initial adaption registers using geometric and/or solution vector information.
2. Manipulate these registers and their associated refinement and coarsening marks.
   - Manipulate the registers by changing the type and/or combining them to create the desired hybrid function.
   - Manipulate the marks by using Exchange, Invert, Limit, and Fill operations.
3. Delete, display adapt to the hybrid adaption functions.

For example, you can capture the shock wave generated on a wedge in a supersonic flow field by adapting the mesh to the gradients of pressure. The pressure gradient near the surface of the wedge, however, is relatively small. You can therefore use the velocity field to resolve the equally important boundary layer near the surface of the wedge.

- If you adapt to pressure, regions near the surface might be coarsened.
- If you subsequently adapted to velocity, these regions may be refined, but the net result will not gain in resolution.
- If you *combine* the velocity and pressure gradient adaption functions, the new adaption function will allow increased resolution in both regions.

The relative weight of the two functions in the hybrid function is determined by the values of the refinement and coarsening thresholds you specify for each of the flow field variables.

To refine the shock and boundary layer only near the leading edge of the wedge, create a circle at the leading edge of the wedge using the region adaption function, change this new register to a mask, and combine it with the hybrid gradient function.

The GUI and text interface commands generate adaption registers that designate the cells marked for refinement or coarsening. These registers can be converted to mask registers.

Mask Register

Mark registers maintain only two states: ACTIVE and INACTIVE. If the adaption register is converted to a mask, cells marked for refinement become ACTIVE cells, while those that are unmarked or marked for coarsening become INACTIVE.

You can use a mask register to limit adaption to cells within a certain region.

This process is illustrated in Figures 19.11.1, 19.11.2, and 19.11.3.

Figure 19.11.1 shows a cloud of cells representing an adaption register (shaded cells are marked cells). Figure 19.11.2 illustrates the active cells associated with a mask register. If the mask is applied to (combined with) the adaption register, the new adaption register formed from the combination has the marked cells shown in Figure 19.11.3.

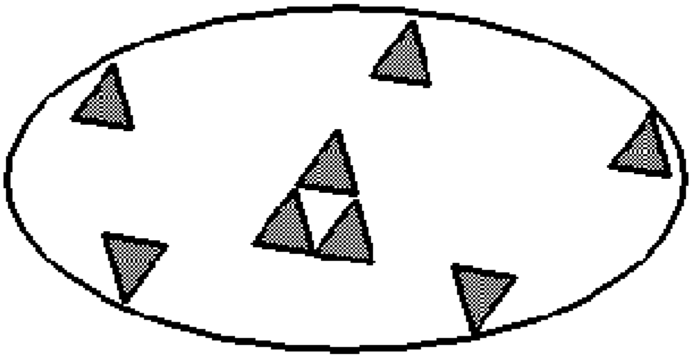


Figure 19.11.1: Adaption Register with Marked Cells

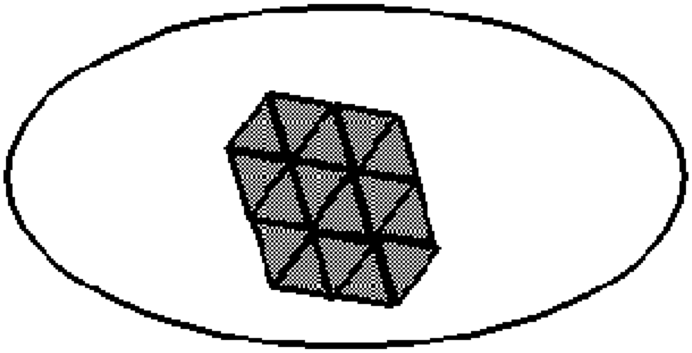


Figure 19.11.2: Mask Register with Active Cells

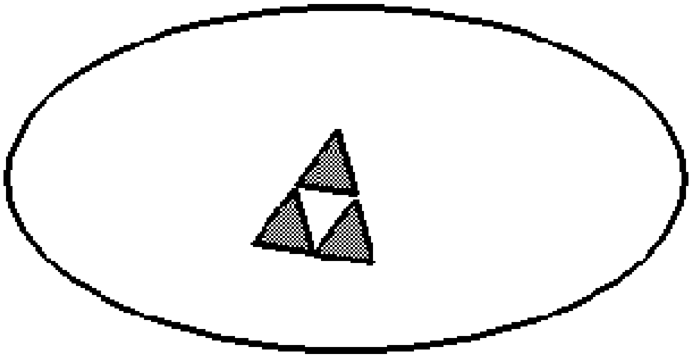


Figure 19.11.3: New Adaption Register Created from Application of Mask

This example does not differentiate between refinement or coarsening marks because the mask is applied to both types of marks. For more information on combining registers, see Section 27.11.1: Manipulating Adaption Registers in the separate User's Guide.

Chapter 20. Reporting Alphanumeric Data

ANSYS FLUENT provides tools for computing and reporting integral quantities at surfaces and boundaries. These tools enable you to find the mass flow rate and heat transfer rate through boundaries, the forces and moments on boundaries, and the area, integral, flow rate, average, and mass average (among other quantities) on a surface or in a volume. In addition, you can print histograms of geometric and solution data, set reference values for the calculation of nondimensional coefficients, and compute projected surface areas. You can also print or save a summary report of the models, boundary conditions, and solver settings in the current case.

This chapter describes some of the background behind ANSYS FLUENT's reporting features. These features are described in the following sections.

- Section 20.1: Fluxes Through Boundaries
- Section 20.2: Forces on Boundaries
- Section 20.3: Surface Integration
- Section 20.4: Volume Integration

Additional information about using these reporting tools can be found in Chapter 30: Reporting Alphanumeric Data in the separate User's Guide. Reporting tools for the discrete phase are described in Section 23.7: Postprocessing for the Discrete Phase in the separate User's Guide.

20.1 Fluxes Through Boundaries

This section describes the quantities that you can compute for boundaries. For more information about generating flux reports, see Section 30.3.1: Generating a Flux Report in the separate User's Guide.

For selected boundary zones, you can compute the following quantities:

- The mass flow rate through a boundary is computed by summing the dot product of the density times the velocity vector and the area projections over the faces of the zone.
- The total heat transfer rate through a boundary is computed by summing the total heat transfer rate, $q = q_c + q_r$, over the faces, where q_c is the convective heat transfer rate and q_r is the radiation heat transfer rate. The computation of the heat transfer through the face depends on the specified boundary condition. For example, the conduction heat transfer on a constant-temperature wall face would be the product of the thermal conductivity with the dot product of the area projection and the temperature gradient. For flow boundaries, the total heat transfer rate is the flow rate of the conserved quantity. Depending on the models that are being used, the total heat transfer rate may include the convective flow of sensible or total enthalpy, diffusive flux of energy, etc. Note that the reference temperature in all enthalpy calculations is always 298.15K
- The radiation heat transfer rate through a boundary is computed by summing the radiation heat transfer rate q_r over the faces. The computation of the radiation heat transfer depends on the radiation model used.

For example, you might use flux reporting to compute the resulting mass flow through a duct with pressure boundaries specified at the inlet and exit.

20.2 Forces on Boundaries

For selected wall zones, you can compute and report the forces along a specified vector, the moments about a specified center and along a specified axis, and the coordinates of the center of pressure. This feature is useful for reporting, for instance, aerodynamic quantities such as lift, drag, and moment coefficients, as well as the center of pressure for an airfoil.

For information about how ANSYS FLUENT computes forces, moments, and centers of pressure, see Section 20.2.1: Computing Forces, Moments, and the Center of Pressure. Otherwise, for more information about generating a report on forces, moments, or centers of pressure, see Section 30.4.1: Generating a Force, Moment, or Center of Pressure Report in the separate User's Guide.

20.2.1 Computing Forces, Moments, and the Center of Pressure

The total force component along the specified force vector $\vec{a}$ on a wall zone is computed by summing the dot product of the pressure and viscous forces on each face with the specified force vector. The terms in this summation represent the pressure and viscous force components in the direction of the vector $\vec{a}$:

$$\underbrace{F_a}_{total\ force\ component} = \underbrace{\vec{a}\cdot\vec{F}_p}_{pressure\ force\ component} + \underbrace{\vec{a}\cdot\vec{F}_v}_{viscous\ force\ component} \qquad (20.2\text{-}1)$$

where

$\vec{a}$ = specified force vector
$\vec{F}_p$ = pressure force vector
$\vec{F}_v$ = viscous force vector

In addition to the actual pressure, viscous, and total forces, the associated force coefficients are also computed for each of the selected wall zones, using the reference values (as described in Section 30.11: Reference Values in the separate User's Guide). The force coefficient is defined as force divided by $\frac{1}{2}\rho v^2 A$, where ρ, v, and A are the density, velocity, and area. Finally, the net values of the pressure, viscous, and total forces and coefficients for all of the selected wall zones are also computed.

The total moment vector about a specified center A is computed by summing the cross products of the pressure and viscous force vectors for each face with the moment vector $\vec{r}_{AB}$, which is the vector from the specified moment center A to the force origin B (see Figure 20.2.1). The terms in this summation represent the pressure and viscous moment vectors:

$$\underbrace{\vec{M}_A}_{total\ moment} = \underbrace{\vec{r}_{AB}\times\vec{F}_p}_{pressure\ moment} + \underbrace{\vec{r}_{AB}\times\vec{F}_v}_{viscous\ moment} \qquad (20.2\text{-}2)$$

where

A = specified moment center
B = force origin
$\vec{r}_{AB}$ = moment vector
$\vec{F}_p$ = pressure force vector
$\vec{F}_v$ = viscous force vector

Direction of the total moment vector follows the right hand rule for cross products.

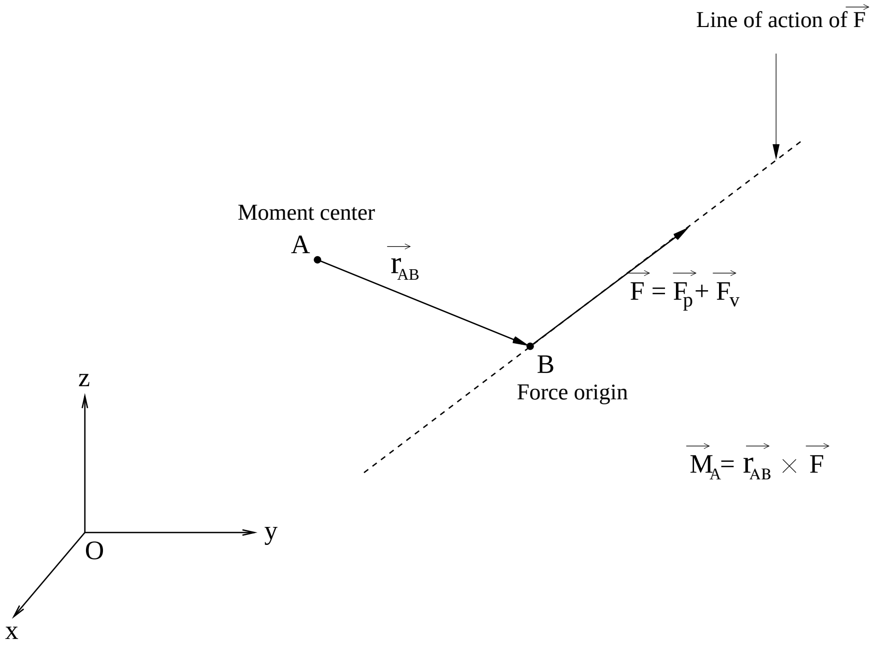

Line of action of $\vec{F}$
Moment center
A
$\vec{r}_{AB}$
$\vec{F} = \vec{F}_p + \vec{F}_v$
B
Force origin
$\vec{M}_A = \vec{r}_{AB} \times \vec{F}$
z
y
O
x

Figure 20.2.1: Moment About a Specified Moment Center

Along with the actual components of the pressure, viscous, and total moments, the moment coefficients are computed for each of the selected wall zones, using the reference values (as described in Section 30.11: Reference Values in the separate User's Guide). The moment coefficient is defined as the moment divided by $\frac{1}{2}\rho v^2 AL$, where ρ, v, A, and L are the density, velocity, area, and length. The coefficient values for the individual wall zones are also summed to yield the net values of the pressure, viscous, and total moments and coefficients for all of the selected wall zones.

Furthermore, the moments along a specified axis are computed. These moments, also known as torques, are defined as the dot product of a unit vector in the direction of the specified axis and the individual and net values of the pressure, viscous, and total moments and coefficients.

To reduce round-off error, a reference pressure is used to normalize the cell pressure for computation of the pressure force. For example, the net pressure force vector, acting on a wall zone, is computed as the vector sum of the individual force vectors for each cell face:

$$\vec{F}_p = \sum_{i=1}^{n} (p - p_{\rm ref}) A\hat{n} \tag{20.2-3}$$

$$= \sum_{i=1}^{n} pA\hat{n} - p_{\rm ref} \sum_{i=1}^{n} A\hat{n} \tag{20.2-4}$$

where n is the number of faces, A is the area of the face, and $\hat{n}$ is the unit normal to the face.

The center of pressure is the average location of the pressure. The pressure varies around the surface of an object, such that $P = p(x)$. The general expression for determining the center of pressure is

$$cp = \frac{\int xp(x)dx}{\int p(x)dx} \tag{20.2-5}$$

However, the center of pressure of a wall is also defined as the point about which the moment on the wall(s) will be zero, that is, the point on the wall where all of the forces balance. It is computed as follows:

$$\vec{M} = \vec{r} \times \vec{F} \tag{20.2-6}$$

$$\vec{M} = 0 \tag{20.2-7}$$

where $\vec{F}$ is the force acting on the selected wall and $\vec{r}$ is the position vector from the center of pressure to the point where the force is applied.

20.3 Surface Integration

You can compute the area or mass flow rate, or the integral, standard deviation, flow rate, volume flow rate, area-weighted average, flow rate, mass-weighted average, sum, facet average, facet maximum, facet minimum, vertex average, vertex minimum, and vertex maximum for a selected field variable on selected surfaces in the domain. These surfaces are sets of data points created by ANSYS FLUENT for each of the zones in your model, or defined by you using the methods described in Chapter 28: Creating Surfaces for Displaying and Reporting Data in the separate User's Guide, or by using the Text User Interface, described in Chapter 3: Text User Interface (TUI) in the separate User's Guide.

The following list provides information on the data reported for the various surface integrals:

- For the vertex average, vertex maximum, and the vertex minimum, ANSYS FLUENT reports the node values of the selected variable on the selected surface.
- For mass flow rate, volume flow rate, and flow rate, ANSYS FLUENT reports the rate. Of these, flow rate is the only one associated with a selected variable. The values used in the computation depend on the kind of surface selected:
  - The face flux values are used for face zone surfaces.
  - The cell values are used for postprocessing surfaces.
- For all other surface integrals, ANSYS FLUENT reports the integral, using values that are appropriate for the particular surface:
  - For face zone surfaces, the face values are used when they are available, that is, when they are calculated by the solver or specified as a boundary condition. Otherwise, the cell values are used. The cell value for noninternal faces (i.e., faces that only have `c0` and no `c1`) is the `c0` value. The cell value for internal faces (i.e., faces that have `c0` and `c1`) is the average of the `c0` and `c1` values.
  - For postprocessing surfaces, the cell values are used. See Section 31.1.1: Cell Values in the separate User's Guide for further details.

Example uses of several types of surface integral reports are given below:

- Area: You can compute the area of a velocity inlet zone, and then estimate the velocity from the mass flow rate:

$$v = \frac{\dot{m}}{\rho A} \tag{20.3-1}$$

- Area-weighted average: You can find the average value on a solid surface, such as the average heat flux on a heated wall with a specified temperature.
- Mass average: You can find the average value on a surface in the flow, such as average enthalpy at a velocity inlet.
- Mass flow rate: You can compute the mass flow rate through a velocity inlet zone, and then estimate the velocity from the area, as described above.
- Flow rate: To calculate the heat transfer rate through a surface, you can calculate the flow rate of enthalpy.
- Integral: You can use integrals for more complex calculations, which may involve the use of the **Custom Field Function Calculator** dialog box, described in Section 31.5: Custom Field Functions in the separate User's Guide, to calculate a function that requires integral computations (e.g., swirl number).
- Standard deviation: You can find the standard deviation of a specified field variable on a surface, such as enthalpy, viscosity, and velocity.
- Volume flow rate: This will report the volume flow rate through the specified surface.

For information about how ANSYS FLUENT computes surface integrals, see Section 20.3.1: Computing Surface Integrals. Otherwise, for more information about generating a surface integral report, see Section 30.6.1: Generating a Surface Integral Report in the separate User's Guide.

20.3.1 Computing Surface Integrals

Area

The area of a surface is computed by summing the areas of the facets that define the surface. Facets on a surface are either triangular or quadrilateral in shape.

$$\int dA = \sum_{i=1}^{n} |A_i| \tag{20.3-2}$$

Integral

An integral on a surface is computed by summing the product of the facet area and the selected field variable facet value, such as density or pressure. For details on the computation of the facet values, see Section 20.3: Surface Integration.

Area-Weighted Average

The area-weighted average of a quantity is computed by dividing the summation of the product of the selected field variable and facet area by the total area of the surface:

$$\frac{1}{A}\int \phi dA = \frac{1}{A}\sum_{i=1}^{n}\phi_i|A_i| \tag{20.3-3}$$

Flow Rate

The flow rate of a quantity through a surface is computed by summing the value of the selected field variable multiplied by the density and the dot product of the facet area vector and the facet velocity vector.

$$\int \phi\rho\vec{v}\cdot d\vec{A} = \sum_{i=1}^{n}\phi_i\rho_i\vec{v_i}\cdot\vec{A}_i \tag{20.3-4}$$

Mass Flow Rate

The mass flow rate through a surface is computed by summing the value of the selected field variable multiplied by the density and the dot product of the facet area vector and the facet velocity vector.

$$\int \rho\vec{v}\cdot d\vec{A} = \sum_{i=1}^{n}\rho_i\vec{v_i}\cdot\vec{A}_i \tag{20.3-5}$$

Mass-Weighted Average

The mass-weighted average of a quantity is computed by dividing the summation of the value of the selected field variable multiplied by the absolute value of the dot product of the facet area and momentum vectors by the summation of the absolute value of the dot product of the facet area and momentum vectors (surface mass flux):

$$\frac{\int \phi\rho\left|\vec{v}\cdot d\vec{A}\right|}{\int \rho\left|\vec{v}\cdot d\vec{A}\right|} = \frac{\sum_{i=1}^{n}\phi_i\rho_i\left|\vec{v_i}\cdot\vec{A}_i\right|}{\sum_{i=1}^{n}\rho_i\left|\vec{v_i}\cdot\vec{A}_i\right|} \tag{20.3-6}$$

Sum of Field Variable

The sum of a specified field variable on a surface is computed by summing the value of the selected variable at each facet:

$$\sum_{i=1}^{n} \phi_i \tag{20.3-7}$$

Facet Average

The facet average of a specified field variable on a surface is computed by dividing the summation of the facet values of the selected variable by the total number of facets. See Section 31.1: Node, Cell, and Facet Values in the separate User's Guide for definitions of facet values.

$$\frac{\sum_{i=1}^{n} \phi_i}{n} \tag{20.3-8}$$

Facet Minimum

The facet minimum of a specified field variable on a surface is the minimum facet value of the selected variable on the surface. See Section 31.1: Node, Cell, and Facet Values in the separate User's Guide for definitions of facet values.

Facet Maximum

The facet maximum of a specified field variable on a surface is the maximum facet value of the selected variable on the surface. See Section 31.1: Node, Cell, and Facet Values in the separate User's Guide for definitions of facet values.

Vertex Average

The vertex average of a specified field variable on a surface is computed by dividing the summation of the vertex values of the selected variable by the total number of vertices. See Section 31.1: Node, Cell, and Facet Values in the separate User's Guide for definitions of vertex values.

$$\frac{\sum_{i=1}^{n} \phi_i}{n} \tag{20.3-9}$$

Vertex Minimum

The vertex minimum of a specified field variable on a surface is the minimum vertex value of the selected variable on the surface. See Section 31.1: Node, Cell, and Facet Values in the separate User's Guide for definitions of vertex values.

Vertex Maximum

The vertex maximum of a specified field variable on a surface is the maximum vertex value of the selected variable on the surface. See Section 31.1: Node, Cell, and Facet Values in the separate User's Guide for definitions of vertex values.

Standard-Deviation

The standard deviation of a specified field variable on a surface is computed using the mathematical expression below:

$$\sqrt{\frac{\sum_{i=1}^{n}(x-x_0)^2}{n}} \tag{20.3-10}$$

where x is the cell value of the selected variables at each facet, x_0 is the mean of x

$$x_0 = \frac{\sum_{i=1}^{n} x}{n}$$

and n is the total number of facets. See Section 31.1: Node, Cell, and Facet Values in the separate User's Guide for definitions of facet values.

Volume Flow Rate

The volume flow rate through a surface is computed by summing the value of the facet area vector multiplied by the facet velocity vector:

$$\int \vec{v} \cdot d\vec{A} = \sum_{i=1}^{n} \vec{v_i} \cdot \vec{A}_i \tag{20.3-11}$$

20.4 Volume Integration

The volume, sum, maximum, minimum, volume integral, volume-weighted average, mass integral, and mass-weighted average can be obtained for a selected field variable in selected cell zones in the domain.

Example uses of the different types of volume integral reports are given below:

- Volume: You can compute the total volume of a fluid region.
- Sum: You can add up the discrete-phase mass or energy sources to determine the net transfer from the discrete phase. You can also sum user-defined sources of mass or energy.
- Maximum: The maximum value of the selected variable at each cell in the selected zone.
- Minimum: The minimum value of the selected variable at each cell in the selected zone.
- Volume Integral: For quantities that are stored per unit volume, you can use volume integrals to determine the net value (e.g., integrate density to determine mass).
- Volume-Average: You can obtain volume averages of mass sources, energy sources, or discrete-phase exchange quantities.
- Mass Integral: You can determine the total mass of a particular species by integrating its mass fraction.
- Mass-Average: You can find the average value (such as average temperature) in a fluid zone.

For information about how ANSYS FLUENT computes volume integrals, see Section 20.4.1: Computing Volume Integrals. Otherwise, for more information about generating a volume integral report, see Section 30.7.1: Generating a Volume Integral Report in the separate User's Guide.

20.4.1 Computing Volume Integrals

Volume

The volume of a cell zone is computed by summing the volumes of the cells that comprise the zone:

$$\int dV = \sum_{i=1}^{n} |V_i| \tag{20.4-1}$$

Sum

The sum of a specified field variable in a cell zone is computed by summing the value of the selected variable at each cell in the selected zone:

$$\sum_{i=1}^{n} \phi_i \tag{20.4-2}$$

Volume Integral

A volume integral is computed by summing the product of the cell volume and the selected field variable:

$$\int \phi dV = \sum_{i=1}^{n} \phi_i |V_i| \tag{20.4-3}$$

Volume-Weighted Average

The volume-weighted average of a quantity is computed by dividing the summation of the product of the selected field variable and cell volume by the total volume of the cell zone:

$$\frac{1}{V} \int \phi dV = \frac{1}{V} \sum_{i=1}^{n} \phi_i |V_i| \tag{20.4-4}$$

Mass-Weighted Integral

The mass-weighted integral is computed by summing the product of density, cell volume, and the selected field variable:

$$\int \phi \rho dV = \sum_{i=1}^{n} \phi_i \rho_i |V_i| \tag{20.4-5}$$

Mass-Weighted Average

The mass-weighted average of a quantity is computed by dividing the summation of the product of density, cell volume, and the selected field variable by the summation of the product of density and cell volume:

$$\frac{\int \phi \rho dV}{\int \rho dV} = \frac{\sum_{i=1}^{n} \phi_i \rho_i |V_i|}{\sum_{i=1}^{n} \rho_i |V_i|} \tag{20.4-6}$$

Nomenclature

A	Area (m^2, ft^2)
$\vec{a}$	Acceleration (m/s^2, ft/s^2)
a	Local speed of sound (m/s, ft/s)
c	Concentration (mass/volume, moles/volume)
C_D	Drag coefficient, defined different ways (dimensionless)
c_p, c_v	Heat capacity at constant pressure, volume (J/kg-K, Btu/lb$_m$-°F)
d	Diameter; d_p, D_p, particle diameter (m, ft)
D_H	Hydraulic diameter (m, ft)
$\mathcal{D}_{ij}, D$	Mass diffusion coefficient (m^2/s, ft^2/s)
E	Total energy, activation energy (J, kJ, cal, Btu)
f	Mixture fraction (dimensionless)
$\vec{F}$	Force vector (N, lb$_f$)
F_D	Drag force (N, lb$_f$)
$\vec{g}$	Gravitational acceleration (m/s^2, ft/s^2); standard values = 9.80665 m/s^2, 32.1740 ft/s^2
Gr	Grashof number $\equiv$ ratio of buoyancy forces to viscous forces (dimensionless)
H	Total enthalpy (energy/mass, energy/mole)
h	Heat transfer coefficient (W/m^2-K, Btu/ft^2-h-°F)
h	Species enthalpy; h^0, standard state enthalpy of formation (energy/mass, energy/mole)
I	Radiation intensity (energy per area of emitting surface per unit solid angle)
J	Mass flux; diffusion flux (kg/m^2-s, lb$_m$/ft^2-s)
K	Equilibrium constant = forward rate constant/backward rate constant (units vary)
k	Kinetic energy per unit mass (J/kg, Btu/lb$_m$)
k	Reaction rate constant, e.g., k_1, k_{-1},$k_{f,r}$, $k_{b,r}$ (units vary)
k	Thermal conductivity (W/m-K, Btu/ft-h-°F)
k_B	Boltzmann constant (1.38×10^{-23} J/molecule-K)
k,k_c	Mass transfer coefficient (units vary); also K, K_c
ℓ, l, L	Length scale (m, cm, ft, in)

Le Lewis number ≡ ratio of thermal diffusivity to mass diffusivity (dimensionless)

m Mass (g, kg, lb_m)

$\dot{m}$ Mass flow rate (kg/s, lb_m/s)

M_w Molecular weight (kg/kgmol, $\mathrm{lb}_m/\mathrm{lb}_m$mol)

M Mach number ≡ ratio of fluid velocity magnitude to local speed of sound (dimensionless)

Nu Nusselt number ≡ dimensionless heat transfer or mass transfer coefficient (dimensionless); usually a function of other dimensionless groups

p Pressure (Pa, atm, mm Hg, $\mathrm{lb}_f/\mathrm{ft}^2$)

Pe Peclet number ≡ Re × Pr for heat transfer, and ≡ Re × Sc for mass transfer (dimensionless)

Pr Prandtl number ≡ ratio of momentum diffusivity to thermal diffusivity (dimensionless)

Q Flow rate of enthalpy (W, Btu/h)

q Heat flux ($\mathrm{W/m}^2$, $\mathrm{Btu/ft}^2$-h)

R Gas-law constant (8.31447×10^3 J/kgmol-K, 1.98588 Btu/lb_mmol-°F)

r Radius (m, ft)

$\mathcal{R}$ Reaction rate (units vary)

Ra Rayleigh number ≡ Gr × Pr; measure of the strength of buoyancy-induced flow in natural (free) convection (dimensionless)

Re Reynolds number ≡ ratio of inertial forces to viscous forces (dimensionless)

S Total entropy (J/K, J/kgmol-K, Btu/lb_mmol-°F)

s Species entropy; s^0, standard state entropy (J/kgmol-K, Btu/lb_mmol-°F)

Sc Schmidt number ≡ ratio of momentum diffusivity to mass diffusivity (dimensionless)

S_{ij} Mean rate-of-strain tensor (s^{-1})

T Temperature (K, °C, °R, °F)

t Time (s)

U Free-stream velocity (m/s, ft/s)

u, v, w Velocity magnitude (m/s, ft/s); also written with directional subscripts (e.g., v_x, v_y, v_z, v_r)

V Volume (m^3, ft^3)

$\vec{v}$ Overall velocity vector (m/s, ft/s)

We Weber number ≡ ratio of aerodynamic forces to surface tension forces (dimensionless)

X	Mole fraction (dimensionless)
Y	Mass fraction (dimensionless)
α	Permeability (m^2, ft^2)
α	Thermal diffusivity (m^2/s, ft^2/s)
α	Volume fraction (dimensionless)
β	Coefficient of thermal expansion (K^{-1})
γ	Porosity (dimensionless)
γ	Ratio of specific heats, $\mathrm{c}_p/\mathrm{c}_\mathrm{v}$ (dimensionless)
Δ	Change in variable, final $-$ initial (e.g., Δp, Δt, ΔH, ΔS, ΔT)
δ	Delta function (units vary)
ϵ	Emissivity (dimensionless)
ϵ	Lennard-Jones energy parameter (J/molecule)
ϵ	Turbulent dissipation rate ($\mathrm{m}^2/\mathrm{s}^3$, $\mathrm{ft}^2/\mathrm{s}^3$)
ϵ	Void fraction (dimensionless)
η	Effectiveness factor (dimensionless)
η', η''	Rate exponents for reactants, products (dimensionless)
θ_r	Radiation temperature (K)
λ	Molecular mean free path (m, nm, ft)
λ	Wavelength (m, nm, Å, ft)
μ	Dynamic viscosity (cP, Pa-s, lb_m/ft-s)
ν	Kinematic viscosity (m^2/s, ft^2/s)
ν', ν''	Stoichiometric coefficients for reactants, products (dimensionless)
ρ	Density (kg/m^3, $\mathrm{lb}_m/\mathrm{ft}^3$)
σ	Stefan-Boltzmann constant (5.67×10^8 W/m^2-K^4)
σ	Surface tension (kg/m, dyn/cm, lb_f/ft)
σ_s	Scattering coefficient (m^{-1})
$\overline{\overline{\tau}}$	Stress tensor (Pa, $\mathrm{lb}_f/\mathrm{ft}^2$)
τ	Shear stress (Pa, $\mathrm{lb}_f/\mathrm{ft}^2$)
τ	Time scale, e.g., τ_c, τ_p, τ_c (s)
τ	Tortuosity, characteristic of pore structure (dimensionless)
ϕ	Equivalence ratio (dimensionless)
ϕ	Thiele modulus (dimensionless)
Ω	Angular velocity; Ω_{ij}, Mean rate of rotation tensor (s^{-1})

ω	Specific dissipation rate (s^{-1})
Ω, Ω'	Solid angle (degrees, radians, gradians)
Ω_D	Diffusion collision integral (dimensionless)

Bibliography

[1] User manual for 1993 version pulverized coal gasification and combustion 3-dimensional (pcgc-3). Advanced Combustion Engineering Research Center, Brigham Young University, 1993.

[2] T. Ahmad, S. L. Plee, and J. P. Myers. Computation of Nitric Oxide and Soot Emissions from Turbulent Diffusion Flames. *J. of Engineering for Gas Turbines and Power*, 107:48–53, 1985.

[3] B. J. Alder and T. E. Wainwright. Studies in Molecular Dynamics II: Behaviour of a Small Number of Elastic Spheres. *J. Chem. Phys.*, 33:1439, 1960.

[4] A. A. Amsden. KIVA-3: A KIVA Program with Block-Structured Mesh for Complex Geometries. Technical Report LA-12503-MS, UC-361, Los Alamos National Laboratory, Los Alamos, New Mexico, March 1993.

[5] T. B. Anderson and R. Jackson. A Fluid Mechanical Description of Fluidized Beds. *I & EC Fundam.*, 6:527–534, 1967.

[6] W. Anderson and D. L. Bonhus. An Implicit Upwind Algorithm for Computing Turbulent Flows on Unstructured Grids. *Computers Fluids*, 23(1):1–21, 1994.

[7] A. Antifora, M. Sala, A. Perera, and L. Vigevano. NO_x Emissions in Combustion Systems of Coal Fired Furnaces with a Reducing Environment: Predictions and Measurements. In *Fourth International Conference on Technologies and Combustion for a Clean Environment*, Lisbon, Portugal, 1997.

[8] S. Armsfield and R. Street. The Fractional-Step Method for the Navier-Stokes Equations on Staggered Grids: Accuracy of Three Variations. *Journal of Computational Physics*, 153:660–665, 1999.

[9] F. Backmier, K. H. Eberius, and T. Just. *Comb. Sci. Tech.*, 7:77, 1973.

[10] S. Badzioch and P. G. W. Hawksley. Kinetics of Thermal Decomposition of Pulverized Coal Particles. *Ind. Eng. Chem. Process Design and Development*, 9:521–530, 1970.

[11] J. Baldyga. Turbulent mixer model with application to homogenous instantaneous chemical reactions. *Chem. Eng. Sci.*, 44:1175–1182, 1989.

[12] R. S. Barlow, G. J. Fiechtner, C. D. Carter, and J. Y. Chen. Experiments on the Scalar Structure of Turbulent CO/H_2/N_2 Jet Flames. *Combustion and Flame*, 120:549–569, 2000.

[13] F. J. Barnes, J. H. Bromly, T. J. Edwards, and R. Madngezewsky. NO_x Emissions from Radiant Gas Burners. *Journal of the Institute of Energy*, 155:184–188, 1988.

[14] T. J. Barth and D. Jespersen. The design and application of upwind schemes on unstructured meshes. Technical Report AIAA-89-0366, AIAA 27th Aerospace Sciences Meeting, Reno, Nevada, 1989.

[15] H. Barths, C. Antoni, and N. Peters. Three-Dimensional Simulation of Pollutant Formation in a DI-Diesel Engine Using Multiple Interactive Flamelets. SAE Paper, SAE, 1998.

[16] H. Barths, N. Peters, N. Brehm, A. Mack, M. Pfitzner, and V. Smiljanowski. Simulation of pollutant formation in a gas turbine combustor using unsteady flamelets. In *27th Symp. (Int'l.) on Combustion*, pages 1841–1847. The Combustion Institute, 1998.

[17] G. K. Batchelor. *An Introduction to Fluid Dynamics.* Cambridge Univ. Press, Cambridge, England, 1967.

[18] D. L. Baulch, D. D. Drysdall, D. G. Horne, and A. C. Lloyd. *Evaluated Kinetic Data for High Temperature Reactions*, volume 1,2,3. Butterworth, 1973.

[19] D. L. Baulch et al. Evaluated Kinetic Data for Combustion Modelling. *J. Physical and Chemical Reference Data*, 21(3), 1992.

[20] M. M. Baum and P. J. Street. Predicting the Combustion Behavior of Coal Particles. *Combust. Sci. Tech.*, 3(5):231–243, 1971.

[21] L. L. Baxter. *Turbulent Transport of Particles.* PhD thesis, Brigham Young University, Provo, Utah, 1989.

[22] L. L. Baxter and P. J. Smith. Turbulent Dispersion of Particles: The STP Model. *Energy & Fuels*, 7:852–859, 1993.

[23] W. Bechara, C. Bailly, P. Lafon, and S. Candel. Stochastic Approach to Noise Modeling for Free Turbulent Flows. *AIAA Journal*, 32:3, 1994.

[24] M. Behnia, S. Parneix, Y. Shabany, and P. A. Durbin. Numerical Study of Turbulent Heat Transfer in Confined and Unconfined Impinging Jets. *International Jounal of Heat and Fluid Flow*, 20:1–9, 1999.

[25] R. W. Bilger and R. E. Beck. In *15th Symp. (Int'l.) on Combustion*, page 541. The Combustion Institute, 1975.

[26] R. W. Bilger, M. B. Esler, and S. H. Starner. On Reduced Mechanisms for Methane-Air Combustion. In *Lecture Notes in Physics*, volume 384, page 86. Springer-Verlag, 1991.

[27] B. Binniger, M. Chan, G. Paczkko, and M. Herrmann. Numerical Simulation of Turbulent Partially Premixed Hydrogen Flames with the Flamelet Model. Technical report, Advanced Combustion Gmbh, Internal Report, 1998.

[28] G. A. Bird. *Molecular Gas Dynamics and the direct Simulation of Gas Flows.* Oxford University Press, New York, 1994.

[29] J. Blauvens, B. Smets, and J. Peters. In *16th Symp. (Int'l.) on Combustion.* The Combustion Institute, 1977.

[30] Blint. *Principles of Combustion.* Wiley-Interscience, New York, 1988.

[31] R. M. Bowen. Theory of Mixtures. In A. C. Eringen, editor, *Continuum Physics*, pages 1–127. Academic Press, New York, 1976.

[32] C. T. Bowman. Chemistry of Gaseous Pollutant Formation and Destruction. In W. Bartok and A. F. Sarofim, editors, *Fossil Fuel Combustion.* J. Wiley and Sons, Canada, 1991.

[33] R. K. Boyd and J. H. Kent. Three-dimensional furnace computer modeling. In *21st Symp. (Int'l.) on Combustion*, pages 265–274. The Combustion Institute, 1986.

[34] J. U. Brackbill, D. B. Kothe, and C. Zemach. A Continuum Method for Modeling Surface Tension. *J. Comput. Phys.*, 100:335–354, 1992.

[35] A. Brandt. Multi-level Adaptive Computations in Fluid Dynamics. Technical Report AIAA-79-1455, AIAA, Williamsburg, VA, 1979.

[36] K. N. Bray and N. Peters. Laminar Flamelets in Turbulent Flames. In P. A. Libby and F. A. Williams, editors, *Turbulent Reacting Flows*, pages 63–114. Academic Press, 1994.

[37] C.E. Brennen. Cavitation and Bubble Dynamics. *Oxford University Press*, 1995.

[38] K. S. Brentner and F. Farassat. An Analytical Comparison of the Acoustic Analogy and Kirchhoff Formulations for Moving Surfaces. *AIAA Journal*, 36(8), 1998.

[39] S. J. Brookes and J. B. Moss. Prediction of Soot and Thermal Radiation in Confined Turbulent Jet Diffusion Flames. *Combustion and Flame*, 116:486–503, 1999.

[40] J. Brouwer, M. P. Heap, D. W. Pershing, and P. J. Smith. A Model for Prediction of Selective Non-Catalytic Reduction of Nitrogen Oxides by Ammonia, Urea, and Cyanuric Acid with Mixing Limitations in the Presence of CO. In *26th Symposium (Int'l) on Combustion, The Combustion Institute*, 1996.

[41] A. L. Brown. *Modeling Soot in Pulverized Coal Flames.* MSc thesis, Brigham Young University, Utah, USA, 1997.

[42] S. Brunauer. *The Absorption of Gases and Vapors.* Princeton University Press, Princeton, NJ, 1943.

[43] Trong T. Bui. A Parallel, Finite-Volume Algorithm for Large-Eddy Simulation of Turbulent Flows. Technical Memorandum NASA/TM-1999-206570, 1999.

[44] S. Candel and T. Poinsot. Flame stretch and the balance equation for the flame area. *Combustion Science and Technology*, 70:1–15, 1990.

[45] R. Cao and S. B. Pope. Numerical Integration of Stochastic Differential Equations: Weak Second-Order Mid-Point Scheme for Application in the Composition PDF Method. *Journal of Computational Physics*, 185(1):194–212, 2003.

[46] N. F. Carnahan and K. E. Starling. Equations of State for Non-Attracting Rigid Spheres. *J. Chem. Phys.*, 51:635–636, 1969.

[47] M. G. Carvalho, T. Farias, and P. Fontes. Predicting Radiative Heat Transfer in Absorbing, Emitting, and Scattering Media Using the Discrete Transfer Method. In W. A. Fiveland et al., editor, *Fundamentals of Radiation Heat Transfer*, volume 160, pages 17–26. ASME HTD, 1991.

[48] J. R. Cash and A. H. Karp. A variable order Runge-Kutta method for initial value problems with rapidly varying right-hand sides. *ACM Transactions on Mathematical Software*, 16:201–222, 1990.

[49] S. Chapman and T. G. Cowling. *The Mathematical Theory of Non-Uniform Gases.* Cambridge University Press, Cambridge, England, 3rd edition, 1990.

[50] S. Charpenay, M. A. Serio, and P. R. Solomon. In *24th Symp. (Int'l.) on Combustion*, pages 1189–1197. The Combustion Institute, 1992.

[51] H. C. Chen and V. C. Patel. Near-Wall Turbulence Models for Complex Flows Including Separation. *AIAA Journal*, 26(6):641–648, 1988.

[52] P. Cheng. Two-Dimensional Radiating Gas Flow by a Moment Method. *AIAA Journal*, 2:1662–1664, 1964.

[53] R. V. Chima and M. S. Liou. Comparison of the AUSM+ and H-CUSP schemes for turbomachinery applications. NASA TM-2003-212457, 2003.

[54] A. J. Chorin. Numerical solution of navier-stokes equations. *Mathematics of Computation*, 22:745–762, 1968.

[55] B. H. Chowdhury. Emission Control Alternatives for Electric Utility Power Plants. *Energy Sources*, 18(4):393–406, October 1996.

[56] E. H. Chui and G. D. Raithby. Computation of Radiant Heat Transfer on a Non-Orthogonal Mesh Using the Finite-Volume Method. *Numerical Heat Transfer, Part B*, 23:269–288, 1993.

[57] Clift, Grace, and Weber. Bubbles, Drops, and Particles. Technical report, Academic Press, 1978.

[58] D. Cokljat, V. A. Ivanov, F. J. Sarasola, and S. A. Vasquez. Multiphase k-epsilon Models for Unstructured Meshes. In *ASME 2000 Fluids Engineering Division Summer Meeting*, Boston, USA, 2000.

[59] D. Cokljat, M. Slack, and S. A. Vasquez. Reynolds-Stress Model for Eulerian Multiphase. In Y. Nagano K. Hanjalic and M. J. Tummers, editors, *Proceedings of the 4th International Symposium on Turbulence Heat and Mass Transfer*, pages 1047–1054. Begell House, Inc., 2003.

[60] A. Coppalle and P. Vervisch. The Total Emissivities of High-Temperature Flames. *Combustion and Flame*, 49:101–108, 1983.

[61] S. M. Correa. A Review of NO_x Formation Under Gas-Turbine Combustion Conditions. *Combustion Science and Technology*, 87:329–362, 1992.

[62] C. Crowe, M. Sommerfield, and Yutaka Tsuji. *Multiphase Flows with Droplets and Particles.* CRC Press, 1998.

[63] G. T. Csanady. Turbulent Diffusion of Heavy Particles in the Atmosphere. *J. Atmos. Science*, 20:201–208, 1963.

[64] N. Curle. The Influence of Solid Boundaries upon Aerodynamic Sound. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 231:505–514, 1955.

[65] J. Dacles-Mariani, G. G. Zilliac, J. S. Chow, and P. Bradshaw. Numerical/Experimental Study of a Wingtip Vortex in the Near Field. *AIAA Journal*, 33(9):1561–1568, 1995.

[66] J. M. Dalla Valle. *Micromeritics.* Pitman, London, 1948.

[67] B. J. Daly and F. H. Harlow. Transport Equations in Turbulence. *Phys. Fluids*, 13:2634–2649, 1970.

[68] J. F. Daunenhofer and J. R. Baron. Grid Adaption for the 2D Euler Equations. Technical Report AIAA-85-0484, American Institute of Aeronautics and Astronautics, 1985.

[69] G. G. De Soete. Overall Reaction Rates of NO and N_2 Formation from Fuel Nitrogen. In *15th Symp. (Int'l.) on Combustion*, pages 1093–1102. The Combustion Institute, 1975.

[70] M. K. Denison and B. W. Webb. A Spectral Line-Based Weighted-Sum-of-Gray-Gases Model for Arbitrary RTE Solvers. *J. Heat Transfer*, 115:1002–1012, 1993.

[71] J. Ding and D. Gidaspow. A Bubbling Fluidization Model Using Kinetic Theory of Granular Flow. *AIChE J.*, 36(4):523–538, 1990.

[72] G. Dixon-Lewis. Structure of Laminar Flames. In *23rd Symp. (Int'l.) on Combustion*, pages 305–324. The Combustion Institute, 1990.

[73] N. Dombrowski and P. C. Hooper. The effect of ambient density or drop formation in sprays. *Chemical Engineering Science*, 17:291–305, 1962.

[74] N. Dombrowski and W. R. Johns. The aerodynamic Instability and Disintegration of Viscous Liquid Sheets. *Chemical Engineering Science*, 18:203, 1963.

[75] C. Dopazo and E. E. O'Brien. Functional formulation of nonisothermal turbulent reactive flows. *Phys. Fluids*, 17:1968, 1975.

[76] A. M. Douaud and P. Eyzat. Four-Octane-Number Method for Predicting the Anti-Knock Behavior of Fuels in Engines. SAE Technical Paper, v87 780080, SAE, 1978.

[77] M. C. Drake and R. J. Blint. Relative Importance of Nitrogen Oxide Formation Mechanisms in Laminar Opposed-Flow Diffusion Flames. *Combustion and Flame*, 83:185–203, 1991.

[78] M. C. Drake, S. M. Correa, R. W. Pitz, W. Shyy, and C. P. Fenimore. Superequilibrium and Thermal Nitric Oxide Formation in Turbulent Diffusion Flames. *Combustion and Flame*, 69:347–365, 1987.

[79] M. C. Drake, R. W. Pitz, M. Lapp, C. P. Fenimore, R. P. Lucht, D. W. Sweeney, and N. M. Laurendeau. In *20th Symp. (Int'l.) on Combustion*, page 327. The Combustion Institute, 1984.

[80] D. A. Drew and R. T. Lahey. In *Particulate Two-Phase Flow*, pages 509–566. Butterworth-Heinemann, Boston, 1993.

[81] J. K. Dukowwicz and A. S. Dvinsky. Approximate Factorization as a High-Order Splitting for the Implicit Incompressible Flow Equations. *Journal of Computational Physics*, 102:336–347, 1992.

[82] P. A. Durbin. Separated Flow Computations with the k-ϵ-v^2 Model. *AIAA Journal*, 33(4):659–664, 1995.

[83] E. R. G. Eckert and R. M. Drake. *Analysis of Heat and Mass Transfer*. McGraw-Hill Co., 1972.

[84] D. K. Edwards and R. Matavosian. Scaling Rules for Total Absorptivity and Emissivity of Gases. *J. Heat Transfer*, 106:684–689, 1984.

[85] J. K. Edwards, B. S. McLaury, and S. A. Shirazi. Supplementing a CFD Code with Erosion Prediction Capabilities. In *Proceedings of ASME FEDSM'98: ASME 1998 Fluids Engineering Division Summer Meeting*, Washington DC, June 1998.

[86] J. K. Edwards, B. S. McLaury, and S. A. Shirazi. Evaluation of Alternative Pipe Bend Fittings in Erosive Service. In *Proceedings of ASME FEDSM'00: ASME 2000 Fluids Engineering Division Summer Meeting*, Boston, June 2000.

[87] S. E. Elgobashi and T. W. Abou-Arab. A Two-Equation Turbulence Model for Two-Phase Flows. *Phys. Fluids*, 26(4):931–938, 1983.

[88] S. Ergun. Fluid Flow through Packed Columns. *Chem. Eng. Prog.*, 48(2):89–94, 1952.

[89] G. Erlebacher, M. Y. Hussaini, C. G. Speziale, and T. A. Zang. Toward the Large-Eddy Simulation of Compressible Turbulent Flows. *J. Fluid Mech.*, 238:155–185, 1992.

[90] R. F. Fedors and R. F. Landell. An Empirical Method of Estimating the Void Fraction in Mixtures of Uniform Particles of Different Size. *Powder Technology*, 23:225–231.

[91] C. P. Fenimore. Formation of Nitric Oxide in Premixed Hydrocarbon Flames. In *13th Symp. (Int'l.) on Combustion*, page 373. The Combustion Institute, 1971.

[92] C. P. Fenimore. Destruction of NO by NH_3 in Lean Burnt Gas. *Combustion and Flame*, 37:245, 1980.

[93] C. P. Fenimore and G. W. Jones. Oxidation of soot by hydroxyl radicals. *J. Phys. Chem.*, 71:593–597, 1967.

[94] J. L. Ferzieger and M. Peric. *Computational Methods for Fluid Dynamics*. Springer-Verlag, Heidelberg, 1996.

[95] J. E. Ffowcs-Williams and D. L. Hawkings. Sound Generation by Turbulence and Surfaces in Arbitrary Motion. *Proc. Roy. Soc. London*, A264:321–342, 1969.

[96] M. A. Field. Rate of Combustion Of Size-Graded Fractions of Char from a Low Rank Coal between 1200 K–2000 K. *Combustion and Flame*, 13:237–252, 1969.

[97] I. Finnie. Erosion of Surfaces by Solid Particles. *Wear*, 3:87–103, 1960.

[98] W. A. Fiveland and A. S. Jamaluddin. Three-Dimensional Spectral Radiative Heat Transfer Solutions by the Discrete Ordinates Method. HTD Vol. 106, Heat Transfer Phenomena in Radiation, Combustion and Fires, pp. 43–48, 1989.

[99] T. H. Fletcher and D. R. Hardesty. Compilation of Sandia coal devolatilization data: Milestone report. Sandia Report SAND92-8209, 1992.

[100] T. H. Fletcher and A. R. Kerstein. Chemical percolation model for devolatilization: 3. Direct use of ^{13}C NMR data to predict effects of coal type. *Energy and Fuels*, 6:414, 1992.

[101] T. H. Fletcher, A. R. Kerstein, R. J. Pugmire, and D. M. Grant. Chemical percolation model for devolatilization: 2. Temperature and heating rate effects on product yields. *Energy and Fuels*, 4:54, 1990.

[102] W. L. Flower, R. K. Hanson, and C. H. Kruger. In *15th Symp. (Int'l.) on Combustion*, page 823. The Combustion Institute, 1975.

[103] R. O. Fox. *Computational Models for Turbulent Reacting Flows.* Cambridge University Press, Cambridge, England, 2003.

[104] S. Fu, B. E. Launder, and M. A. Leschziner. Modeling Strongly Swirling Recirculating Jet Flow with Reynolds-Stress Transport Closures. In *Sixth Symposium on Turbulent Shear Flows*, Toulouse, France, 1987.

[105] J. Garside and M. R. Al-Dibouni. Velocity-Voidage Relationships for Fluidization and Sedimentation. *I & EC Process Des. Dev.*, 16:206–214, 1977.

[106] M. Germano, U. Piomelli, P. Moin, and W. H. Cabot. Dynamic Subgrid-Scale Eddy Viscosity Model. In *Summer Workshop, Center for Turbulence Research*, Stanford, CA, 1996.

[107] T. Gessner. *Dynamic Mesh Adaption for Supersonic Combustion Waves Modeled with Detailed Reaction Mechanisms.* PhD thesis, University of Freiburg, Freiburg, Germany, 2001.

[108] M. M. Gibson and B. E. Launder. Ground Effects on Pressure Fluctuations in the Atmospheric Boundary Layer. *J. Fluid Mech.*, 86:491–511, 1978.

[109] D. Gidaspow. *Multiphase Flow and Fluidization.* Academic Press, Boston, 1994.

[110] D. Gidaspow, R. Bezburuah, and J. Ding. Hydrodynamics of Circulating Fluidized Beds, Kinetic Theory Approach. In *Fluidization VII, Proceedings of the 7th Engineering Foundation Conference on Fluidization*, pages 75–82, 1992.

[111] R. G. Gilbert, K. Luther, and J. Troe. *Ber. Bunsenges. Phys. Chem.*, 87, 1983.

[112] P. Glarborg, J. E. Johnsson, and K. Dam-Johansen. Kinetics of Homogeneous Nitrous Oxide Decomposition. *Combustion and Flame*, 99:523–532, 1994.

[113] M. E. Goldstein and B. Rosenbaum. Effect of Anisotropic Turbulence on Aerodynamic Noise. *Journal of the Acoustical Society of America*, 54:630–645, 1973.

[114] J. Gottgens, F. Mauss, and N. Peters. Analytic Approximations of Burning Velocities and Flame Thicknesses of Lean Hydrogen, Methane, Ethylene, Ethane, Acetylene and Propane Flames. In *Twenty-Fourth Symposium (Int.) on Combustion*, pages 129–135, Pittsburgh, 1992.

[115] I. R. Gran and B. F. Magnussen. A numerical study of a bluff-body stabilized diffusion flame. part 2. influence of combustion modeling and finite-rate chemistry. *Combustion Science and Technology*, 119:191, 1996.

[116] D. M. Grant, R. J. Pugmire, T. H. Fletcher, and A. R. Kerstein. Chemical percolation model of coal devolatilization using percolation lattice statistics. *Energy and Fuels*, 3:175, 1989.

[117] D. J. Gunn. Transfer of Heat or Mass to Particles in Fixed and Fluidized Beds. *Int. J. Heat Mass Transfer*, 21:467–476, 1978.

[118] W. W. Hagerty and J. F. Shea. A Study of the Stability of Plane Fluid Sheets. *Journal of Applied Mechanics*, 22:509, 1955.

[119] A. Haider and O. Levenspiel. Drag Coefficient and Terminal Velocity of Spherical and Nonspherical Particles. *Powder Technology*, 58:63–70, 1989.

[120] R. J. Hall, M. D. Smooke, and M. B. Colket. *Physical and Chemical Aspects of Combustion.* Gordon and Breach, 1997.

[121] M. P. Halstead, L. J. Kirsch, and C. P. Quinn. Autoignition of Hydrocarbon Fuels at High Temperatures and Pressures – Fitting of a Mathematical Model. *Combustion and Flame*, 30:45–60, 1977.

[122] Z. Han, S. Perrish, P. V. Farrell, and R. D. Reitz. Modeling Atomization Processes of Pressure-Swirl Hollow-Cone Fuel Sprays. *Atomization and Sprays*, 7(6):663–684, Nov.-Dec. 1997.

[123] G. Hand, M. Missaghi, M. Pourkashanian, and A. Williams. Experimental Studies and Computer Modelling of Nitrogen Oxides in a Cylindrical Furnace. In *Proceedings of the Ninth Members Conference*, volume 2. IFRF Doc No K21/g/30, 1989.

[124] R. K. Hanson and S. Salimian. Survey of Rate Constants in H/N/O Systems. In W. C. Gardiner, editor, *Combustion Chemistry*, page 361, 1984.

[125] H. O. Hardenburg and F. W. Hase. An Empirical Formula for Computing the Pressure Rise Delay of a Fuel from its Cetane Number and from the Relevant Parameters of Direct Injection Diesel Engines. SAE Technical Paper 790493, SAE, 1979.

[126] K. Haugen, O. Kvernvold, A. Ronald, and R. Sandberg. Sand Erosion of Wear-resistant Materials: Erosion in Choke Valves. *Wear*, 186-187:179–188, 1995.

[127] R. A. W. M. Henkes, F. F. van der Flugt, and C. J. Hoogendoorn. Natural Convection Flow in a Square Cavity Calculated with Low-Reynolds-Number Turbulence Models. *Int. J. Heat Mass Transfer*, 34:1543–1557, 1991.

[128] J. B. Heywood. *Internal Combustion Engine Fundamentals*. McGraw-Hill, New York, rev. edition, 1988.

[129] T. Hibiki and M. Ishii. One-group Interfacial Area Transport of Bubbly Flows in Vertical Round Tubes. *International Journal of Heat and Mass Transfer*, 43:2711–2726, 2000.

[130] J. O. Hinze. *Turbulence*. McGraw-Hill Publishing Co., New York, 1975.

[131] C. W. Hirt and B. D. Nichols. Volume of Fluid (VOF) Method for the Dynamics of Free Boundaries. *J. Comput. Phys.*, 39:201–225, 1981.

[132] D. G. Holmes and S. D. Connell. Solution of the 2D Navier-Stokes Equations on Unstructured Adaptive Grids. Presented at the AIAA 9th Computational Fluid Dynamics Conference, June, 1989.

[133] T. J. Houser, M. Hull, R. Alway, and T. Biftu. *Int. Journal of Chem. Kinet.*, 12:579, 1980.

[134] P. Huang, P. Bradshaw, and T. Coakley. Skin Friction and Velocity Profile Family for Compressible Turbulent Boundary Layers. *AIAA Journal*, 31(9):1600–1604, September 1993.

[135] S. C. Hunter. Formation of so3 in gas turbines. *Transactions of the ASME*, 104:44–51, 1982.

[136] B. R. Hutchinson and G. D. Raithby. A Multigrid Method Based on the Additive Correction Strategy. *Numerical Heat Transfer*, 9:511–537, 1986.

[137] H. Ibdir and H. Arastoopour. Modeling of multi-type particle flow using kinetic approach. *AICHE Journal*, May 2005.

[138] K. Ishazaki, T. Ikohagi, and H. Daiguji. A High-Resolution Numerical Method for Transonic Non-equilibrium Condensation Flows Through a Steam Turbine Cascade. In *Proceedings of the 6th International Symposium on Computational Fluid Dynamics*, volume 1, pages 479–484, 1995.

[139] M. Ishii and S. Kim. Micro Four-Sensor Probe Measurement of Interfacial Area Transport for Bubbly Flow in Round Pipes. *Nuclear Engineering and Design*, 205:123–131, 2001.

[140] R. I. Issa. Solution of the Implicitly Discretized Fluid Flow Equations by Operator-Splitting. *Journal of Computational Physics*, 62:40–65, 1985.

[141] R. I. Issa. Solution of Implicitly Discretized Fluid Flow Equations by Operator Splitting. *J. Comput. Phys.*, 62:40–65, 1986.

[142] H. M. Glaz J. B. Bell, P. Colella. Second-Order Projection Method for the Incompressible Navier-Stokes Equations. *Journal of Computational Physics*, 85:257, 1989.

[143] H. M. Glaz J. B. Bell, P. Colella. An Analysis of the Fractional-Step Method. *Journal of Computational Physics*, 108:51–58, 1993.

[144] S. Jain. *Three-Dimensional Simulation of Turbulent Particle Dispersion.* PhD thesis, University of Utah, Utah, 1995.

[145] A. Jameson. Solution of the Euler Equations for Two Dimensional Transonic Flow by a Multigrid Method. MAE Report 1613, Princeton University, June 1983.

[146] A. Jameson, W. Schmidt, and E. Turkel. Numerical Solution of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time-Stepping Schemes. Technical Report AIAA-81-1259, AIAA 14th Fluid and Plasma Dynamics Conference, Palo Alto, California, June 1981.

[147] J. Janicka, W. Kolbe, and W. Kollmann. Closure of the transport equation for the pdf of turbulent scalar fields. *Journal Non-Equilibrium Thermodynamics*, 4:47, 1978.

[148] J. Janicka and W. Kollmann. A Two-Variable Formulation for the Treatment of Chemical Reactions in Turbulent H_2-Air Diffusion Flames. In *17th Symp. (Int'l.) on Combustion*. The Combustion Institute, 1978.

[149] J. Janicka and W. Kollmann. A Numerical Study of Oscillating Flow Around a Circular Cylinder. *Combustion and Flame*, 44:319–336, 1982.

[150] C. Jayatillaka. The Influence of Prandtl Number and Surface Roughness on the Resistance of the Laminar Sublayer to Momentum and Heat Transfer. *Prog. Heat Mass Transfer*, 1:193–321, 1969.

[151] P. C. Johnson and R. Jackson. Frictional-Collisional Constitutive Relations for Granular Materials, with Application to Plane Shearing. *J. Fluid Mech.*, 176:67–93, 1987.

[152] W. P. Jones and J. H. Whitelaw. Calculation Methods for Reacting Turbulent Flows: A Review. *Combustion and Flame*, 48:1–26, 1982.

[153] T. Jongen. *Simulation and Modeling of Turbulent Incompressible Flows.* PhD thesis, EPF Lausanne, Lausanne, Switzerland, 1992.

[154] T. Just and S. Kelm. *Die Industry*, 38:76, 1986.

[155] B. Kader. Temperature and Concentration Profiles in Fully Turbulent Boundary Layers. *Int. J. Heat Mass Transfer*, 24(9):1541–1544, 1981.

[156] N. Kandamby, G. Lazopoulos, F. C. Lockwood, A. Perera, and L. Vigevano. Mathematical Modeling of NO_x Emission Reduction by the Use of Reburn Technology in Utility Boilers. In *ASME Int. Joint Power Generation Conference and Exhibition*, Houston, Texas, 1996.

[157] K. C. Karki and S. V. Patankar. Pressure-Based Calculation Procedure for Viscous Flows at All Speeds in Arbitrary Configurations. *AIAA Journal*, 27:1167–1174, 1989.

[158] W. M. Kays. Loss coefficients for abrupt changes in flow cross section with low reynolds number flow in single and multiple tube systems. *Transactions of the ASME*, 72:1067–1074, January 1950.

[159] W. M. Kays. Turbulent Prandtl Number - Where Are We? *J. Heat Transfer*, 116:284–295, 1994.

[160] W. M. Kays and A. L. London. *Compact Heat Exchangers.* McGraw-Hill, New York, 1964.

[161] R. J. Kee, F. M. Rupley, J. A. Miller, M. E. Coltrin, J. F. Grcar, E. Meeks, H. K. Moffat, A. E. Lutz, G. Dixon-Lewis, M. D. Smooke, J. Warnatz, G. H. Evans, R. S. Larson, R. E. Mitchell, L. R. Petzold, W. C. Reynolds, M. Caracotsios, W. E. Stewart, P. Glarborg, C. Wang, O. Adigun, W. G. Houf, C. P. Chou, S. F. Miller, P. Ho, and D. J. Young. CHEMKIN v. 4.0. Technical Report San Diego, CA, Reaction Design, Inc., 2004.

[162] I. M. Khan and G. Greeves. A Method for Calculating the Formation and Combustion of Soot in Diesel Engines. In N. H. Afgan and J. M. Beer, editors, *Heat Transfer in Flames*, chapter 25. Scripta, Washington DC, 1974.

[163] J. S. Kim and F. A. Williams. Extinction of Diffusion Flames with Non-Unity Lewis Number. *Eng. Math*, 31:101–118, 1997.

[164] S. Kim, D. Caraeni, and B. Makarov. A Multidimensional Linear Reconstruction Scheme for Arbitrary Unstructured Grids. Technical report, American Institute of Aeronautics and Astronautics, AIAA 16th Computational Fluid Dynamics Conference, Orlando, Florida, June 2003.

[165] S.-E. Kim. Large eddy simulation using unstructured meshes and dynamic subgrid-scale turbulence models. Technical Report AIAA-2004-2548, American Institute of Aeronautics and Astronautics, 34th Fluid Dynamics Conference and Exhibit, June 2004.

[166] S.-E. Kim and D. Choudhury. A Near-Wall Treatment Using Wall Functions Sensitized to Pressure Gradient. In *ASME FED Vol. 217, Separated and Complex Flows.* ASME, 1995.

[167] S.-E. Kim, D. Choudhury, and B. Patel. Computations of Complex Turbulent Flows Using the Commercial Code ANSYS FLUENT. In *Proceedings of the ICASE/LaRC/AFOSR Symposium on Modeling Complex Turbulent Flows*, Hampton, Virginia, 1997.

[168] W.-W. Kim and S. Menon. Application of the localized dynamic subgrid-scale model to turbulent wall-bounded flows. Technical Report AIAA-97-0210, American Institute of Aeronautics and Astronautics, 35th Aerospace Sciences Meeting, Reno, NV, January 1997.

[169] H. Kobayashi, J. B. Howard, and A. F. Sarofim. Coal Devolatilization at High Temperatures. In *16th Symp. (Int'l.) on Combustion.* The Combustion Institute, 1976.

[170] N.I. Kolev. *Multiphase Flow Dynamics 2: Thermal and Mechanical Interactions.* Springer, Berlin, Germany, 2nd edition edition, 20051994.

[171] R. Kraichnan. Diffusion by a Random Velocity Field. *Physics of Fluids*, 11:21–31, 1970.

[172] J. C. Kramlich. *The Fate and Behavior of Fuel-Sulfur in Combustion Systems.* PhD thesis, Washington State University, Washington, USA, 1980.

[173] K. K. Y. Kuo. *Principles of Combustion.* John Wiley and Sons, New York, 1986.

[174] H. Lamb. *Hydrodynamics, Sixth Edition.* Dover Publications, New York, 1945.

[175] M. E. Larsen and J. R. Howell. Least Squares Smoothing of Direct Exchange Areas in Zonal Analysis. *J. Heat Transfer*, 108:239–242, 1986.

[176] B. E. Launder. Second-Moment Closure and Its Use in Modeling Turbulent Industrial Flows. *International Journal for Numerical Methods in Fluids*, 9:963–985, 1989.

[177] B. E. Launder. Second-Moment Closure: Present... and Future? *Inter. J. Heat Fluid Flow*, 10(4):282–300, 1989.

[178] B. E. Launder, G. J. Reece, and W. Rodi. Progress in the Development of a Reynolds-Stress Turbulence Closure. *J. Fluid Mech.*, 68(3):537–566, April 1975.

[179] B. E. Launder and N. Shima. Second-Moment Closure for the Near-Wall Sublayer: Development and Application. *AIAA Journal*, 27(10):1319–1325, 1989.

[180] B. E. Launder and D. B. Spalding. *Lectures in Mathematical Models of Turbulence.* Academic Press, London, England, 1972.

[181] B. E. Launder and D. B. Spalding. The Numerical Computation of Turbulent Flows. *Computer Methods in Applied Mechanics and Engineering*, 3:269–289, 1974.

[182] N. M. Laurendeau. Heterogeneous Kinetics of Coal Char Gasification and Combustion. *Prog. Energy Comb. Sci.*, 4:221–270, 1978.

[183] J. L. Lebowitz. Exact Solution of Generalized Percus-Yevick Equation for a Mixture of Hard Spheres. *The Phy. Rev.*, 133(4A):A895–A899, 1964.

[184] K. B. Lee, M. W. Thring, and J. M. Beer. *Combustion and Flame*, 6:137–145, 1962.

[185] W.H. Lee. A Pressure Iteration Scheme for Two-Phase Modeling. Technical Report LA-UR 79-975, Los Alamos Scientific Laboratory, Los Alamos, New Mexico, 1979.

[186] A. H. Lefebvre. *Atomization and Sprays.* Hemisphere Publishing Corporation, 1989.

[187] B. P. Leonard. The ULTIMATE conservative difference scheme applied to unsteady one-dimensional advection. *Comp. Methods Appl. Mech. Eng.*, 88:17–74, 1991.

[188] B. P. Leonard and S. Mokhtari. ULTRA-SHARP Nonoscillatory Convection Schemes for High-Speed Steady Multidimensional Flow. NASA TM 1-2568 (ICOMP-90-12), NASA Lewis Research Center, 1990.

[189] K. M. Leung and R. P. Lindsted. Detailed Kinetic Modeling of C1-C3 Alkane Diffusion Flames. *Combustion and Flame*, 102:129–160, 1995.

[190] J. M. Levy, L. K. Chen, A. F. Sarofim, and J. M. Beer. NO/Char Reactions at Pulverized Coal Flame Conditions. In *18th Symp. (Int'l.) on Combustion*. The Combustion Institute, 1981.

[191] A. Li and G. Ahmadi. Dispersion and Deposition of Spherical Particles from Point Sources in a Turbulent Channel Flow. *Aerosol Science and Technology*, 16:209–226, 1992.

[192] X. Li and R. S. Tankin. On the Temporal Instability of a Two-Dimensional Viscous Liquid Sheet. *Journal of Fluid Mechanics*, 226:425, 1991.

[193] A. K. Lichtarowicz, R. K. Duggins, and E. Markland. Discharge Coefficients for Incompressible Non-Cavitating Flow Through Long Orifices. *Journal of Mechanical Engineering Science*, 7:2, 1965.

[194] F. S. Lien and M. A. Leschziner. Assessment of Turbulent Transport Models Including Non-Linear RNG Eddy-Viscosity Formulation and Second-Moment Closure. *Computers and Fluids*, 23(8):983–1004, 1994.

[195] M. J. Lighthill. On Sound Generated Aerodynamically. *Proc. Roy. Soc. London*, A211:564–587, 1952.

[196] G. M. Lilley. The radiated noise from isotropic turbulence revisited. NASA Contract Report 93-75, NASA Langley Research Center, Hampton, VA, 1993.

[197] D. K. Lilly. A Proposed Modification of the Germano Subgrid-Scale Closure Model. *Physics of Fluids*, 4:633–635, 1992.

[198] F. Lindemann. *Trans. Faraday Soc.*, 7, 1922.

[199] R. P. Lindstedt. In *Proc. IUTAM Conf. on Aerothermo-Chemistry in Combustion*, Taipei, Taiwan, 1991.

[200] R. P. Lindstedt. *Soot Formation in Combustion.* Springer-Verlag, Berlin, 1994.

[201] M. S. Liou. A sequel to AUSM: AUSM+. *Journal of Computational Physics*, 129:364–382, 1996.

[202] M. S. Liou and C. J. Steffen, Jr. A new flux splitting scheme. *Journal of Computational Physics*, 107(1):23–39, 1993.

[203] A. N. Lipatnikov and J. Chomiak. Turbulent Flame Speed and Thickness: Phenomenology, Evaluation and Application in Multi-dimensional Simulations. *Progress in Energy & Combustion Science*, 28:1–74, January 2002.

[204] R. J. Litchford and S. M. Jeng. Efficient Statistical Transport Model for Turbulent Particle Dispersion in Sprays. *AIAA Journal*, 29:1443, 1991.

[205] A. B. Liu, D. Mather, and R. D. Reitz. Modeling the Effects of Drop Drag and Breakup on Fuel Sprays. SAE Technical Paper 930072, SAE, 1993.

[206] H. Liu and B. M. Gibbs. Modeling of NO and N_2O Emissions from Biomass-Fired Circulating Fluidized Bed Combustors. *Fuel*, 81:271–280, 2002.

[207] F. C. Lockwood and C. A. Romo-Millanes. Mathematical Modelling of Fuel - NO Emissions From PF Burners. *J. Int. Energy*, 65:144–152, September 1992.

[208] C. K. K. Lun, S. B. Savage, D. J. Jeffrey, and N. Chepurniy. Kinetic Theories for Granular Flow: Inelastic Particles in Couette Flow and Slightly Inelastic Particles in a General Flow Field. *J. Fluid Mech.*, 140:223–256, 1984.

[209] J. Y. Luo, R. I. Issa, and A. D. Gosman. Prediction of Impeller-Induced Flows in Mixing Vessels Using Multiple Frames of Reference. In *IChemE Symposium Series*, number 136, pages 549–556, 1994.

[210] A. E. Lutz, R. J. Kee, J. F. Grcar, and F. M. Rupley. OPPDIF: A FORTRAN Program for Computing Opposed-Flow Diffusion Flames. Sandia National Laboratories Report SAND96-8243, 1997.

[211] J. F. Lynn. *Multigrid Solution of the Euler Equations with Local Preconditioning.* PhD thesis, University of Michigan, 1995.

[212] R. K. Lyon. The NH_3-NO-O_2 Reaction. *Int. Journal of Chem. Kinetics*, 8:315–318, 1976.

[213] D. Ma and G. Ahmadi. A Thermodynamical Formulation for Dispersed Multiphase Turbulent Flows. *Int. J. Multiphase Flow*, 16:323–351, 1990.

[214] J. Ma. *Soot Formation and Soot Secondary Reactions During Coal Pyrolysis.* PhD thesis, Brigham Young University, Utah, USA, 1996.

[215] B. F. Magnussen. On the Structure of Turbulence and a Generalized Eddy Dissipation Concept for Chemical Reaction in Turbulent Flow. Nineteeth AIAA Meeting, St. Louis, 1981.

[216] B. F. Magnussen and B. H. Hjertager. On mathematical models of turbulent combustion with special emphasis on soot formation and combustion. In *16th Symp. (Int'l.) on Combustion.* The Combustion Institute, 1976.

[217] M. Manninen, V. Taivassalo, and S. Kallio. On the mixture model for multiphase flow. VTT Publications 288, Technical Research Centre of Finland, 1996.

[218] D. L. Marchisio and R. O. Fox. Solution of Population Balance Equations Using the Direct Quadrature Method of Moments. *Aerosol Science and Technology*, 36:43–73, 2005.

[219] F. Mathey, D. Cokljat, J. P. Bertoglio, and E. Sergent. Specification of LES Inlet Boundary Condition Using Vortex Method. In K. Hanjalić, Y. Nagano, and M. Tummers, editors, *4th International Symposium on Turbulence, Heat and Mass Transfer*, Antalya, Turkey, 2003. Begell House, Inc.

[220] S. R. Mathur and J. Y. Murthy. Coupled ordinates method for multigrid acceleration of radiation calculations. *J. of Thermophysics and Heat Transfer*, 13(4):467–473, 1999.

[221] B. S. McLaury, J. Wang, S. A. Shirazi, J. R. Shadley, and E. F. Rybicki. Solid Particle Erosion in Long Radius Elbows and Straight Pipes. SPE Paper 38842, SPE Annual Technical Conference and Exhibition, II Production Operations and Engineering/General, San Antonio, Texas, October 1997.

[222] P. C. Melte and D. T. Pratt. Measurement of Atomic Oxygen and Nitrogen Oxides in Jet Stirred Combustion. In *15th Symposium (Int'l) on Combustion, The Combustion Institute*, pages 1061–1070, 1974.

[223] C. Meneveau and T. Poinsot. Stretching and Quenching of Flamelets in Premixed Turbulent Combustion. *Combustion and Flame*, 86:311–332, 1991.

[224] F. R. Menter. Two-Equation Eddy-Viscosity Turbulence Models for Engineering Applications. *AIAA Journal*, 32(8):1598–1605, August 1994.

[225] F. R. Menter, M. Kuntz, and R. Langtry. Ten Years of Experience with the SST Turbulence Model. In K. Hanjalic, Y. Nagano, and M. Tummers, editors, *Turbulence, Heat and Mass Transfer 4*, pages 625–632. Begell House Inc., 2003.

[226] F. R. Menter, R. B. Langtry, S. R. Likki, Y. B. Suzen, P. G. Huang, and S. Volker. A Correlation Based Transition Model Using Local Variables Part 1 - Model Formulation. (ASME-GT2004-53452), 2004.

[227] M. Metghalchi and J. C. Keck. Burning velocities of mixtures of air with methanol, isooctane and indolene at high pressures and temperatures. *Combustion and Flame*, 48:191–210, 1982.

[228] J. A. Miller and C. T. Bowman. Mechanism and Modeling of Nitrogen Chemistry in Combustion. *Prog. in Energy and Comb. Sci.*, 15:287–338, 1989.

[229] J. A. Miller, M. C. Branch, W. J. McLean, D. W. Chandler, M. D. Smooke, and R. J. Kee. In *20th Symp. (Int'l.) on Combustion*, page 673. The Combustion Institute, 1985.

[230] J. A. Miller and G. A. Fisk. *Chemical and Engineering News*, 31, 1987.

[231] M. Missaghi. *Mathematical Modelling of Chemical Sources in Turbulent Combustion.* PhD thesis, The University of Leeds, England, 1987.

[232] M. Missaghi, M. Pourkashanian, A. Williams, and L. Yap. In *Proceedings of American Flame Days Conference*, USA, 1990.

[233] M. F. Modest. The Weighted-Sum-of-Gray-Gases Model for Arbitrary Solution Methods in Radiative Transfer. *J. Heat Transfer*, 113:650–656, 1991.

[234] M. F. Modest. *Radiative Heat Transfer.* Series in Mechanical Engineering. McGraw-Hill, 1993.

[235] L. J. Molero de Blas. *Pollutant Formation and Interaction in the Combustion of Heavy Liquid Fuels.* PhD thesis, University of London, London, England, 1998.

[236] J. P. Monat, R. K. Hanson, and C. H. Kruger. In *17th Symp. (Int'l.) on Combustion*, page 543. The Combustion Institute, 1979.

[237] M. J. Moore and C. H. Sieverding. *Two-Phase Steam Flow in Turbines and Separator.* McGraw-Hill, 1976.

[238] S. A. Morsi and A. J. Alexander. An Investigation of Particle Trajectories in Two-Phase Flow Systems. *J. Fluid Mech.*, 55(2):193–208, September 26 1972.

[239] J. B. Moss, S. A. L. Perera, C. D. Stewart, and M. Makida. Radiation Heat Transfer in Gas Turbine Combustors. In *Proc. 16th (Int'l.) Symp. on Airbreathing Engines*, Cleveland, OH, 2003.

[240] C. M. Muller, H. Breitbach, and N. Peters. Partially Premixed Turbulent Flame Propagation in Jet Flames. Technical report, 25th Symposium (Int) on Combustion, The Combustion Institute, 1994.

[241] C. Mundo, M. Sommerfeld, and C. Tropea. Droplet-Wall Collisions: Experimental Studies of the Deformation and Breakup Process. *International Journal of Multiphase Flow*, 21(2):151–173, 1995.

[242] J. Y. Murthy and S. R. Mathur. A Finite Volume Method For Radiative Heat Transfer Using Unstructured Meshes. AIAA-98-0860, January 1998.

[243] S. Muzaferija, M. Peric, P. Sames, and T. Schellin. A Two-Fluid Navier-Stokes Solver to Simulate Water Entry. In *Proc 22nd Symposium on Naval Hydrodynamics*, pages 277–289, Washington, DC, 1998.

[244] J. D. Naber and R. D. Reitz. Modeling Engine Spray/Wall Impingement. Technical Report 880107, Society of Automotive Engineers, General Motors Research Laboratories, Warren, MI, 1988.

[245] M. Namazian and J. B. Heywood. Flow in the Piston Cylinder Ring Crevices of a Spark Ignition Engine: Effect on Hydrocarbon Emissions, Efficiency, and Power. SAE Technical Paper 820088, SAE, 1982.

[246] I. Naruse, Y. Yamamoto, Y. Itoh, and K. Ohtake. Fundamental Study on N_2O Formation/Decomposition Characteristics by Means of Low-Temperature Pulverized Coal Combustion. In *26th Symposium (Int'l) on Combustion, The Combustion Institute*, pages 3213–3221, 1996.

[247] P. F. Nelson, A. N. Buckley, and M. D. Kelly. Functional Forms of Nitrogen in Coals and the Release of Coal Nitrogen as NO_x Precursors (HCN and NH_3). In *24th Symposium (Int'l) on Combustion, The Combustion Institute*, page 1259, 1992.

[248] F. Nicoud and F. Ducros. Subgrid-Scale Stress Modelling Based on the Square of the Velocity Gradient Tensor. *Flow, Turbulence, and Combustion*, 62(3):183–200, 1999.

[249] L. Nokleberg and T. Sontvedt. Erosion of Oil and Gas Industry Choke Valves Using Computational Fluid Dynamics and Experiment. *International Journal of Heat and Fluid Flow*, 19:636–643, 1998.

[250] P. A. Nooren, H. A. Wouters, T. W. J. Peeters, D. Roekaerts, U. Maas, and D. Schmidt. Monte carlo pdf modeling of a turbulent natural-gas diffusion flame. *Combustion Theory and Modeling*, 1:79–96, 1997.

[251] J. Norman, M. Porkashanian, and A. Williams. Modelling the Formation and Emission of Environmentally Unfriendly Coal Species in Some Gasification Processes. *Fuel*, 76(13):1201–1216, October 1997.

[252] W. H. Nurick. Orifice Cavitation and Its Effects on Spray Mixing. *Journal of Fluids Engineering*, page 98, 1976.

[253] R. Ocone, S. Sundaresan, and R. Jackson. Gas-particle flow in a duct of arbitrary inclination with particle-particle interaction. *AIChE J.*, 39:1261–1271, 1993.

[254] S. Ogawa, A. Umemura, and N. Oshima. On the Equation of Fully Fluidized Granular Materials. *J. Appl. Math. Phys.*, 31:483, 1980.

[255] P. J. O'Rourke. *Collective Drop Effects on Vaporizing Liquid Sprays.* PhD thesis, Princeton University, Princeton, New Jersey, 1981.

[256] P. J. O'Rourke and A. A. Amsden. The TAB Method for Numerical Calculation of Spray Droplet Breakup. SAE Technical Paper 872089, SAE, 1987.

[257] P. J. O'Rourke and A. A. Amsden. A Particle Numerical Model for Wall Film Dynamics in Port-Fuel Injected Engines. SAE Paper 961961, 1996.

[258] P. J. O'Rourke and A. A. Amsden. A Spray/Wall Interaction Submodel for the KIVA-3 Wall Film Model. SAE Paper 2000-01-0271, 2000.

[259] S.A. Orszag, V. Yakhot, W.S. Flannery, F. Boysan, D. Choudhury, J. Maruzewski, and B. Patel. Renormalization Group Modeling and Turbulence Simulations. In *International Conference on Near-Wall Turbulent Flows*, Tempe, Arizona, 1993.

[260] M. Ostberg and K. Dam-Johansen. Empirical Modeling of the Selective Non-Catalytic Reduction of NO: Comparison with Large-Scale Experiments and Detailed Kinetic Modeling. *Chem. Engineering Science*, 49(12):1897–1904, 1994.

[261] H. Ounis, G. Ahmadi, and J. B. McLaughlin. Brownian Diffusion of Submicrometer Particles in the Viscous Sublayer. *Journal of Colloid and Interface Science*, 143(1):266–277, 1991.

[262] M. N. Ozisik. *Radiative Transfer and Interactions with Conduction and Convection.* Wiley, New York, 1973.

[263] S. A. Pandya, S. Venkateswaran, and T. H. Pulliam. Implementation of dual-time procedures in overflow. Technical Report AIAA-2003-0072, American Institute of Aeronautics and Astronautics, 2003.

[264] S. V. Patankar. *Numerical Heat Transfer and Fluid Flow.* Hemisphere, Washington, DC, 1980.

[265] D.Y. Peng and D.B. Robinson. A New Two-Constant Equation of State. *Industrial and Engineering Chemistry: Fundamentals*, 15:59–64, 1976.

[266] R. H. Perry, D. W. Gree, and J. O. Maloney. *Perry's Chemical Engineers' Handbook*. McGraw-Hill, New York, 6th edition, 1984.

[267] N. Peters. Laminar Diffusion Flamelet Models in Non Premixed Combustion. *Prog. Energy Combust. Sci.*, 10:319–339, 1984.

[268] N. Peters. Laminar Flamelet Concepts in Turbulent Combustion. In *21st Symp. (Int'l.) on Combustion*, pages 1231–1250. The Combustion Institute, 1986.

[269] N. Peters and S. Donnerhack. In *18th Symp. (Int'l.) on Combustion*, page 33. The Combustion Institute, 1981.

[270] N. Peters and B. Rogg. Reduced Kinetic Mechanisms for Applications in Combustion Systems. In *Lecture Notes in Physics*, volume m15. Springer-Verlag, 1992.

[271] K. K. Pillai. The Influence of Coal Type on Devolatilization and Combustion in Fluidized Beds. *J. Inst. Energy*, page 142, 1981.

[272] H. Pitsch, H. Barths, and N. Peters. Three-Dimensional Modeling of NOx and Soot Formation in DI-Diesel Engines Using Detailed Chemistry Based on the Interactive Flamelet Approach. SAE Paper 962057, SAE, 1996.

[273] H. Pitsch and N. Peters. A Consistent Flamelet Formulation for Non-Premixed Combustion Considering Differential Diffusion Effects. *Combustion and Flame*, 114:26–40, 1998.

[274] T. Poinsot and D. Veynante. *Theoretical and Numerical Combustion*. McGraw-Hill, New York, 2001.

[275] B.E. Poling, J.M. Prausnitz, and J.P. OConnell. *The properties of Gases and Liquids*. McGraw-Hill, New York, 5th edition, 2001.

[276] S. B. Pope. Pdf methods for turbulent reactive flows. *Progress Energy Combustion Science*, 11:119, 1985.

[277] S. B. Pope. Computationally efficient implementation of combustion chemistry using in-situ adaptive tabulation. *Combustion Theory and Modeling*, 1:41–63, 1997.

[278] S. B. Pope. *Turbulent Flows*. Cambridge University Press, Cambridge, England, 2000.

[279] B. Popoff and M. Braun. A Lagrangian Approach to Dense Particulate Flows. In *International Conference on Multiphase Flow*, Leipzig, Germany, 2007.

[280] I. Proudman. The Generation of Noise by Isotropic Turbulence. *Proc. Roy. Soc.*, A214:119, 1952.

[281] S.Kim Q. Wu and M. Ishii. One-group Interfacial Area Transport in Vertical Bubbly Flow. *International Journal of Heat and Mass Transfer*, 41:1103–1112, 1997.

[282] G. D. Raithby and E. H. Chui. A Finite-Volume Method for Predicting a Radiant Heat Transfer in Enclosures with Participating Media. *J. Heat Transfer*, 112:415–423, 1990.

[283] W. E. Ranz. Some Experiments on Orifice Sprays. *Canadian Journal of Chemical Engineering*, page 175, 1958.

[284] W. E. Ranz and W. R. Marshall, Jr. Evaporation from Drops, Part I. *Chem. Eng. Prog.*, 48(3):141–146, March 1952.

[285] W. E. Ranz and W. R. Marshall, Jr. Evaporation from Drops, Part II. *Chem. Eng. Prog.*, 48(4):173–180, April 1952.

[286] R. D. Rauch, J. T. Batira, and N. T. Y. Yang. Spatial Adaption Procedures on Unstructured Meshes for Accurate Unsteady Aerodynamic Flow Computations. Technical Report AIAA-91-1106, aiaa, 1991.

[287] R. D. Reitz. Mechanisms of Atomization Processes in High-Pressure Vaporizing Sprays. *Atomization and Spray Technology*, 3:309–337, 1987.

[288] R. D. Reitz and F. V. Bracco. Mechanism of Atomization of a Liquid Jet. *Phys. Fluids.*, 26(10), 1982.

[289] R. D. Reitz and F. V. Bracco. Mechanisms of Breakup of Round Liquid Jets. *The Encyclopedia of Fluid Mechanics, ed. N. Cheremisnoff*, 3:223–249, 1986.

[290] W. C. Reynolds. Thermodynamic Properties in SI: Graphs, Tables, and Computational Equations for 40 Substances. Department of Mechanical Engineering, Stanford University, 1979.

[291] W. C. Reynolds. Fundamentals of turbulence for turbulence modeling and simulation. Lecture Notes for Von Karman Institute Agard Report No. 755, 1987.

[292] C. M. Rhie and W. L. Chow. Numerical Study of the Turbulent Flow Past an Airfoil with Trailing Edge Separation. *AIAA Journal*, 21(11):1525–1532, November 1983.

[293] H. S. Ribner. The Generation of Sound by turbulent jets. In *Advances in Applied Mathematics*. Academic, New York, 1964.

[294] J. R. Richardson and W. N. Zaki. Sedimentation and Fluidization: Part I. *Trans. Inst. Chem. Eng.*, 32:35–53, 1954.

[295] C. E. Roberts and R. D. Matthews. Development and Application of an Improved Ring Pack Model for Hydrocarbon Emissions Studies. SAE Technical Paper 961966, SAE, 1996.

[296] P. L. Roe. Characteristic based schemes for the Euler equations. *Annual Review of Fluid Mechanics*, 18:337–365, 1986.

[297] R. Rota, D. Antos, E. F. Zanoelo, and M. Morbidelli. Experimental and Modeling Analysis of the NOxOUT Process. *Chemical Engineering Science*, 57:27–38, 2002.

[298] P. G. Saffman. The Lift on a Small Sphere in a Slow Shear Flow. *J. Fluid Mech.*, 22:385–400, 1965.

[299] M. M. Salama and E. S. Venkatesh. Evaluation of api rp14e erosional velocity limitations for offshore gas wells. In *OTC Conference*, pages 371–376. Houston, May 1983.

[300] S. Sarkar and L. Balakrishnan. Application of a Reynolds-Stress Turbulence Model to the Compressible Shear Layer. ICASE Report 90-18, NASA CR 182002, 1990.

[301] S. Sarkar and M. Y. Hussaini. Computation of the sound generated by isotropic turbulence. NASA Contract Report 93-74, NASA Langley Research Center, Hampton, VA, 1993.

[302] S. S. Sazhin. An Approximation for the Absorption Coefficient of Soot in a Radiating Gas. Manuscript, Fluent Europe, Ltd., 1994.

[303] D. G. Schaeffer. Instability in the Evolution Equations Describing Incompressible Granular Flow. *J. Diff. Eq.*, 66:19–50, 1987.

[304] R. W. Schefer, M. Namazian, and J. Kelly. In *Combustion Research Facility News*, volume 3, number 4. Sandia, 1991.

[305] L. Schiller and Z. Naumann. *Z. Ver. Deutsch. Ing.*, 77:318, 1935.

[306] D. P. Schmidt and M. L. Corradini. Analytical Prediction of the Exit Flow of Cavitating Orifices. *Atomization and Sprays*, 7:6, 1997.

[307] D. P. Schmidt, M. L. Corradini, and C. J. Rutland. A Two-Dimensional, Non-Equilibrium Model of Flashing Nozzle Flow. In *3rd ASME/JSME Joint Fluids Engineering Conference*, 1999.

[308] D. P. Schmidt, I. Nouar, P. K. Senecal, C. J. Rutland, J. K. Martin, and R. D. Reitz. Pressure-Swirl Atomization in the Near Field. SAE Paper 01-0496, SAE, 1999.

[309] G.H. Schnerr and J. Sauer. Physical and Numerical Modeling of Unsteady Cavitation Dynamics. In *Fourth International Conference on Multiphase Flow*, New Orleans, USA,, 2001.

[310] P. K. Senecal, D. P. Schmidt, I. Nouar, C. J. Rutland, and R. D. Reitz. Modeling High Speed Viscous Liquid Sheet Atomization. *International Journal of Multiphase Flow*, in press.

[311] E. Sergent. *Vers une methodologie de couplage entre la Simulation des Grandes Echelles et les modeles statistiques.* PhD thesis, L'Ecole Centrale de Lyon, Lyon, France, 2002.

[312] N. G. Shah. *A New Method of Computation of Radiant Heat Transfer in Combustion Chambers.* PhD thesis, Imperial College of Science and Technology, London, England, 1979.

[313] T.-H. Shih, W. W. Liou, A. Shabbir, Z. Yang, and J. Zhu. A New k-ϵ Eddy-Viscosity Model for High Reynolds Number Turbulent Flows - Model Development and Validation. *Computers Fluids*, 24(3):227–238, 1995.

[314] M. Shur, P. R. Spalart, M. Strelets, and A. Travin. Detached-Eddy Simulation of an Airfoil at High Angle of Attack. In *4th Int. Symposium on Eng. Turb. Modeling and Experiments*, Corsica, France, May 1999.

[315] R. Siegel and J. R. Howell. *Thermal Radiation Heat Transfer.* Hemisphere Publishing Corporation, Washington DC, 1992.

[316] R. Siegel and C. M. Spuckler. Effect of Refractive Index and Diffuse or Specular Boundaries on a Radiating Isothermal Layer. *J. Heat Transfer*, 116:787–790, 1994.

[317] C. Simonin and P. L. Viollet. Predictions of an Oxygen Droplet Pulverization in a Compressible Subsonic Coflowing Hydrogen Flow. *Numerical Methods for Multiphase Flows*, FED91:65–82, 1990.

[318] A. K. Singhal, H. Y. Li, M. M. Athavale, and Y. Jiang. Mathematical Basis and Validation of the Full Cavitation Model. ASME FEDSM'01, New Orleans, Louisiana, 2001.

[319] Y. R. Sivathanu and G. M. Faeth. Generalized State Relationships for Scalar Properties in Non-Premixed Hydrocarbon/Air Flames. *Combustion and Flame*, 82:211–230, 1990.

[320] J. Smagorinsky. General Circulation Experiments with the Primitive Equations. I. The Basic Experiment. *Month. Wea. Rev.*, 91:99–164, 1963.

[321] R. Smirnov, S. Shi, and I. Celik. Random Flow Generation Technique for Large Eddy Simulations and Particle-Dynamics Modeling. *Journal of Fluids Engineering*, 123:359–371, 2001.

[322] I. W. Smith. *Combustion and Flame*, 17:421, 1971.

[323] I. W. Smith. The Intrinsic Reactivity of Carbons to Oxygen. *Fuel*, 57:409–414, 1978.

[324] I. W. Smith. The Combustion Rates of Coal Chars: A Review. In *19th Symp. (Int'l.) on Combustion*, pages 1045–1065. The Combustion Institute, 1982.

[325] J. M. Smith and H. C. Van Ness. *Introduction to Chemical Engineering Thermodynamics*. McGraw Hill, New Jersey, 1986.

[326] T. F. Smith, Z. F. Shen, and J. N. Friedman. Evaluation of Coefficients for the Weighted Sum of Gray Gases Model. *J. Heat Transfer*, 104:602–608, 1982.

[327] L. D. Smoot and P. J. Smith. NO_x Pollutant Formation in a Turbulent Coal System. In *Coal Combustion and Gasification*, page 373, Plenum, Plenum, NY, 1985.

[328] D. O. Snyder, E. K. Koutsavdis, and J. S. R. Anttonen. Transonic store separation using unstructured cfd with dynamic meshing. Technical Report AIAA-2003-3913, American Institute of Aeronautics and Astronautics, 33th AIAA Fluid Dynamics Conference and Exhibit, 2003.

[329] M. S. Solum, R. J. Pugmire, and D. M. Grant. *Energy and Fuels*, 3:187, 1989.

[330] C. Soteriou, R. Andrews, and M. Smith. Direct Injection Diesel Sprays and the Effect of Cavitation and Hydraulic Flip on Atomization. SAE Paper 950080, SAE, 1995.

[331] P. Spalart and S. Allmaras. A one-equation turbulence model for aerodynamic flows. Technical Report AIAA-92-0439, American Institute of Aeronautics and Astronautics, 1992.

[332] P. R. Spalart, S. Deck, M. L. Shur, K. D. Squires, M. K. Strelets, and A. Travin. A new version of detached-eddy simulation, resistant to ambiguous grid densities. *Theoretical and Computational Fluid Dynamics*, 20:181–195, 2006.

[333] D. B. Spalding. Mixing and chemical reaction in steady confined turbulent flames. In *13th Symp. (Int'l.) on Combustion*. The Combustion Institute, 1970.

[334] C. G. Speziale, S. Sarkar, and T. B. Gatski. Modelling the Pressure-Strain Correlation of Turbulence: An Invariant Dynamical Systems Approach. *J. Fluid Mech.*, 227:245–272, 1991.

[335] H. B. Squire. Investigation of the Instability of a Moving Liquid Film. *British Journal of Applied Physics*, 4:167, 1953.

[336] D. W. Stanton and C. J. Rutland. Modeling Fuel Film Formation and Wall Interaction in Diesel Engines. SAE Paper 960628, 1996.

[337] D. W. Stanton and C. J. Rutland. Multi-Dimensional Modeling of Thin Liquid Films and Spray-Wall Interactions Resulting from Impinging Sprays. *International Journal of Heat and Mass Transfer*, 41:3037–3054, 1998.

[338] R. C. Steele, P. C. Malte, D. G. Nichol, and J. C. Kramlich. NO_x and N_2O in Lean-Premixed Jet-Stirred Flames. *Combustion and Flame*, 100:440–449, 1995.

[339] P. H. Steward, C. W. Larson, and D. Golden. *Combustion and Flame*, 75, 1989.

[340] S. Subramaniam and S. B. Pope. A Mixing Model for Turbulent Reactive Flows Based on Euclidean Minimum Spanning Trees. *Combustion and Flame*, 115:487–514, 1998.

[341] M. Syamlal. *The Particle-Particle Drag Term in a Multiparticle Model of Fluidization*. National Technical Information Service, Springfield, VA, 1987. DOE/MC/21353-2373, NTIS/DE87006500.

[342] M. Syamlal and T. J. O'Brien. Computer Simulation of Bubbles in a Fluidized Bed. *AIChE Symp. Series*, 85:22–31, 1989.

[343] M. Syamlal, W. Rogers, and O'Brien T. J. *MFIX Documentation: Volume 1, Theory Guide*. National Technical Information Service, Springfield, VA, 1993. DOE/METC-9411004, NTIS/DE9400087.

[344] D. Tabacco, C. Innarella, and C. Bruno. Theoretical and Numerical Investigation on Flameless Combustion. *Combustion Science and Technology*, 2002.

[345] L. Talbot et al. Thermophoresis of Particles in a Heated Boundary Layer. *J. Fluid Mech.*, 101(4):737–758, 1980.

[346] I. Tanasawa. Advances in Condensation Heat Transfer. *Advances in Heat Transfer*, 21:55–139, 1991.

[347] G. I. Taylor. The Shape and Acceleration of a Drop in a High Speed Air Stream. Technical report, In the Scientific Papers of G. I. Taylor, ed., G. K. Batchelor, 1963.

[348] P. B. Taylor and P. J. Foster. Some Gray Weighting Coefficients for CO_2-H_2O-Soot Mixtures. *Int. J. Heat Transfer*, 18:1331–1332, 1974.

[349] P. A. Tesner, T. D. Snegiriova, and V. G. Knorre. Kinetics of Dispersed Carbon Formation. *Combustion and Flame*, 17:253–260, 1971.

[350] E. Turkel and V. N. Vatsa. Choice of variables and preconditioning for time dependent problems. Technical Report AIAA-2003-3692, American Institute of Aeronautics and Astronautics, 16th AIAA Computational Fluid Dynamics Conference, Orlando, Florida, June 2003.

[351] O. Ubbink. *Numerical Prediction of Two Fluid Systems With Sharp Interfaces*. PhD thesis, Imperial College of Science, Technology and Medicine, London, England, 1997.

[352] B. Van Leer. Toward the Ultimate Concervative Difference Scheme. IV. A Second Order Sequel to Godunov's Method. *Jounal of Computational Physics*, 32:101–136, 1979.

[353] J. P. Vandoormaal and G. D. Raithby. Enhancements of the SIMPLE Method for Predicting Incompressible Fluid Flows. *Numer. Heat Transfer*, 7:147–163, 1984.

[354] S. A. Vasquez and V. A. Ivanov. A Phase Coupled Method for Solving Multiphase Problems on Unstructured Meshes. In *Proceedings of ASME FEDSM'00: ASME 2000 Fluids Engineering Division Summer Meeting*, Boston, June 2000.

[355] V. Venkatakrishnan. On The Accuracy of Limiters and Convergence to Steady State Solutions. Technical Report AIAA-93-0880, American Institute of Aeronautics and Astronautics, January 1993.

[356] S. Venkateswaran, J. M. Weiss, and C. L. Merkle. Propulsion Related Flowfields Using the Preconditioned Navier-Stokes Equations. Technical Report AIAA-92-3437, AIAA/ASME/SAE/ASEE 28th Joint Propulsion Conference, Nashville, TN, July 1992.

[357] J. R. Viegas, M. W. Rubesin, and C. C. Horstman. On the Use of Wall Functions as Boundary Conditions for Two-Dimensional Separated Compressible Flows. Technical Report AIAA-85-0180, AIAA 23rd Aerospace Sciences Meeting, Reno, Nevada, 1985.

[358] V. R. Voller. Modeling Solidification Processes. Technical report, Mathematical Modeling of Metals Processing Operations Conference, American Metallurgical Society, Palm Desert, CA, 1987.

[359] V. R. Voller, A. D. Brent, and C. Prakash. The Modeling of Heat, Mass and Solute Transport in Solidification Systems. *Int. J. Heat Mass Transfer*, 32(9):1719–1731, 1989.

[360] V. R. Voller, A. D. Brent, and K. J. Reid. A Computational Modeling Framework for the Analysis of Metallurgical Solidification Process and Phenomena. Technical report, Conference for Solidification Processing, Ranmoor House, Sheffield, September 1987.

[361] V. R. Voller and C. Prakash. A Fixed-Grid Numerical Modeling Methodology for Convection-Diffusion Mushy Region Phase-Change Problems. *Int. J. Heat Mass Transfer*, 30:1709–1720, 1987.

[362] V. R. Voller and C. R. Swaminathan. Generalized Source-Based Method for Solidification Phase Change. *Numer. Heat Transfer B*, 19(2):175–189, 1991.

[363] K. S. Vorres. User's handbook for the Argonne premium coal sample bank. Argonne National Laboratory, supported by DOE contract W-31-109-ENG-38, September 1989. Also K. S. Vorres, ACS Div. Fuel Chem. preprint, 32:4, 1987.

[364] D. Keith Walters and Davor Cokljat. A three-equation eddy-viscosity model for reynolds-averaged navier-stokes simulations of transitional flows. *Journal of Fluids Engineering*, 130, December 2008.

[365] L. P. Wang. *On the Dispersion of Heavy Particles by Turbulent Motion.* PhD thesis, Washington State University, 1990.

[366] Z. J. Wang. A Fast Nested Multi-grid Viscous Flow Solver for Adaptive Cartesian/Quad Grids. *International Journal for Numerical Methods in Fluids*, 33:657–680, 2000.

[367] J. Warnatz. NO_x Formation in High Temperature Processes. University of Stuttgart, Germany.

[368] G. P. Warren, W. K. Anderson, J. L. Thomas, and S. L. Krist. Grid convergence for adaptive methods. Technical Report AIAA-91-1592, American Institute of Aeronautics and Astronautics, AIAA 10th Computational Fluid Dynamics Conference, Honolulu, Hawaii, June 1991.

[369] C. Weber. Zum Zerfall eines Flüssigkeitsstrahles. *ZAMM*, 11:136–154, 1931.

[370] J. M. Weiss, J. P. Maruszewski, and W. A. Smith. Implicit Solution of the Navier-Stokes Equations on Unstructured Meshes. Technical Report AIAA-97-2103, 13th AIAA CFD Conference, Snowmass, CO, July 1997.

[371] J. M. Weiss, J. P. Maruszewski, and W. A. Smith. Implicit Solution of Preconditioned Navier-Stokes Equations Using Algebraic Multigrid. *AIAA Journal*, 37(1):29–36, 1999.

[372] J. M. Weiss and W. A. Smith. Preconditioning Applied to Variable and Constant Density Flows. *AIAA Journal*, 33(11):2050–2057, November 1995.

[373] C.-Y. Wen and Y. H. Yu. Mechanics of Fluidization. *Chem. Eng. Prog. Symp. Series*, 62:100–111, 1966.

[374] Z. Wen, S. Yun, M. J. Thomson, and M. F. Lightstone. Modeling Soot Formation in Turbulent Kerosene/Air Jet Diffusion Flames. *Combustion and Flame*, 135:323–340, 2003.

[375] H. Werner and H. Wengle. Large-Eddy Simulation of Turbulent Flow Over and Around a Cube in a Plate Channel. In *Eighth Symposium on Turbulent Shear Flows*, Munich, Germany, 1991.

[376] C. Westbrook and F. Dryer. Chemical Kinetic Modelling of Hydrocarbon Combustion. *Prog. Energy Comb. Sci.*, page 1, 1984.

[377] A. A. Westenberg. *Comb. Sci. Tech.*, 4:59, 1971.

[378] F. White and G. Christoph. A Simple New Analysis of Compressible Turbulent Skin Friction Under Arbitrary Conditions. Technical Report AFFDL-TR-70-133, February 1971.

[379] D. C. Wilcox. *Turbulence Modeling for CFD.* DCW Industries, Inc., La Canada, California, 1998.

[380] F. A. Williams. *Turbulent Mixing in Nonreactive and Reactive Flows.* Plenum Press, New York, 1975.

[381] F. Winter, C. Wartha, G. Loffler, and H. Hofbauer. The NO and N_2O Formation Mechanism During Devolatilization and Char Combustion Under Fluidized Bed Conditions. In *26th Symposium (Int'l) on Combustion, The Combustion Institute*, pages 3325–3334, 1996.

[382] M. Wolfshtein. The Velocity and Temperature Distribution of One-Dimensional Flow with Turbulence Augmentation and Pressure Gradient. *Int. J. Heat Mass Transfer*, 12:301–318, 1969.

[383] P.-K. Wu, L.-K. Tseng, and G. M. Faeth. Primary Breakup in Gas/Liquid Mixing Layers for Turbulent Liquids. *Atomization and Sprays*, 2:295–317, 1995.

[384] V. Yakhot and S. A. Orszag. Renormalization Group Analysis of Turbulence: I. Basic Theory. *Journal of Scientific Computing*, 1(1):1–51, 1986.

[385] J. B. Young. The Spontaneous Condensation od Steam in Supersonic Nozzles. *Physico Chemical Hydrodynamics*, 3(2):57–82, July 1982.

[386] J. B. Young. An Equation of State for Steam for Turbomachinery and Other Flow Calculations. *Journal of Engineering for Gas Turbines and Power*, 110:1–7, January 1988.

[387] J. B. Young. Two-Dimensional, Nonequilibrium, Wet-Steam Calculations for Nozzles and Turbine Cascades. *Journal of Turbomachinery*, 114:569–579, July 1992.

[388] D. L. Youngs. Time-Dependent Multi-Material Flow with Large Fluid Distortion. In K. W. Morton and M. J. Baines, editors, *Numerical Methods for Fluid Dynamics.* Academic Press, 1982.

[389] Q. Zhou and M. A. Leschziner. Technical report, 8th Turbulent Shear Flows Symp., Munich, 1991.

[390] V. Zimont. Gas Premixed Combustion at High Turbulence. Turbulent Flame Closure Model Combustion Model. *Experimental Thermal and Fluid Science*, 21:179–186, 2000.

[391] V. Zimont, W. Polifke, M. Bettelini, and W. Weisenstein. An Efficient Computational Model for Premixed Turbulent Combustion at High Reynolds Numbers Based on a Turbulent Flame Speed Closure. *J. of Gas Turbines Power*, 120:526–532, 1998.

[392] V. L. Zimont, F. Biagioli, and K. J. Syed. Modelling Turbulent Premixed Combustion in the Intermediate Steady Propagation Regime. *Progress in Computational Fluid Dynamics*, 1(1):14–28, 2001.

[393] V. L. Zimont and A. N. Lipatnikov. A Numerical Model of Premixed Turbulent Combustion of Gases. *Chem. Phys. Report*, 14(7):993–1025, 1995.

[394] P.J. Zwart. Numerical Modelling of Free Surface and Cavitating Flows. *VKI Lecture Series*, 2005.

[395] P.J. Zwart, A.G. Gerber, and T. Belamri. A Two-Phase Flow Model for Predicting Cavitation Dynamics. In *Fifth International Conference on Multiphase Flow*, Yokohama, Japan, 2004.

Index