PyVASCO User guide

Patricia Ribes Metidieri
Ida Aichinger
Christina Yin Vallgren

TE-VSC
CERN - Geneva, Switzerland

Contents

Chapter 1

About PyVASCO

PyVASCO (VAcuum Stability COde written in Python) is a code integrally developed at CERN for the simulation of pressure profiles in cylindrical geometries considering beam induced effects.

The first version of this program was distributed under the name *IdaVac*. This program constitutes an update of VASCO (presented in [1]) and seeks to optimize the performance of the original code for large geometries [2].

This program has been integrally developed in Python 2.7 and tested on Windows 10.

1.1 Getting started

Installation

This version of PyVASCO includes an installer, called 'setup.exe'. In order to install PyVASCO in your machine, launch the installer and follow the specified instructions. Even if recommended, the installation using the setup is not compulsory in order to launch PyVASCO. To launch the application without installing, enter in the folder PyVASCO and double-click on the application ('PyVASCO.exe').

Developer tools

This version of PyVASCO is distributed together with its source code (in `PyVASCO/PyVASCO_Code/`) and a portable python interpreter `PyVASCO/WinPython-64bit-2.7.6.4` with all the required dependencies already installed.

There's also an API documentation available in web format in the directory `docs/` under the name 'API.html' or opening the program and selecting the option 'Documentation' in the menu Help or pressing the keyboard key combination Ctrl+U.

To build a stand-alone python application from the source code:

- Make sure that Pyinstaller is installed in your computer:
  - Open a command prompt and type pyinstaller.
  - If you don't have pyinstaller in your computer, in the same command prompt, type:

```
pip install pyinstaller
```

- Enter in the directory containing the source code (`PyVASCO_Code/`) and open the file 'PyVASCO.spec'. Paste the full path of the location of the directory PyVASCO_Code in the tag *pathex* and save the changes.
- Open a command prompt in this directory and type:

```
pyinstaller PyVASCO.spec
```

1.2 Authors and contributors

List of authors:

- Ida Aichinger
- Patricia Ribes Metidieri

List of contributors:

- Christina Yin Vallgren
- Giuseppe Bregliozzi
- Simone Callegari

1.3 Contact

In case of problems,if a bug is detected or if you have suggestions for further development, please send an emalil to the following addresses:

- patricia.ribes.metidieri@cern.ch

Chapter 2

Basic concepts

2.1 Motivation and main advantages with respect to VASCO

PyVASCO is an upgrade of a preexisting program at CERN: VASCO [1], and it solves the same vacuum model. Even though this program is not as precise as other simulations tools for ultra high vacuum (UHV) based on Montecarlo techniques, its main advantages to are twofold: first, PyVASCO can easily simulate several gas species at a time and cross-desorption between gas species. Second, it solves the vacuum model in Eq. 2.1 analytically, which allows to simulate large portions of an accelerator in minutes.

The original VASCO, however, didn't allow for simulations with more than around 40 segments. The reason is that, in order to solve a system of N_{segments} it holds in memory matrices of dimension $8 \cdot N_{\text{gas}} \cdot N_{\text{segments}} \times 8 \cdot N_{\text{gas}} \cdot N_{\text{segments}}$. PyVASCO takes advantage of the sparse structure of the solving matrices to store the information of the system in arrays of dimension $8 \cdot N_{\text{gas}} \cdot N_{\text{segments}} \times 17$. Thus, the memory storage is reduced from $\mathcal{O}(N^2)$ to $\mathcal{O}(N)$. This step is fundamental in order to simulate large geometries.

For more information on the computer implementation of PyVASCO, see Ref. [2].

2.2 Dynamic vacuum model

PyVASCO uses the same vacuum model as presented in Ref. [1] and, as in Ref. [1], PyVASCO solves the equations of the model analytically. The purpose of this section is to give a fast overview of the equations solved by PyVASCO and to present some of the limitations of this vacuum model.

The model used in PyVASCO assumes the the rate of change of molecules per unit volume depends uniquely on:

- molecular diffusion due to a density gradient;
- beam induced effects such as ion, photon and electron stimulated desorption;
- gas pumping through a distributed pumping (NEG of cryo-pumping) or through lumped pumps located in the interconnection of segments;
- addition of gas to the system through lumped leaks and through thermal outgassing.

PyVASCO assumes that the simulated vacuum system is made of cylindrical finite elements characterized with constant, time invariant parameters (material properties, radius, pumps...), and it can take into account the cross- desorption of gas of one specie by ions of other gas species (mu. PyVASCO considers four gas species: H_2, CH_4, CO and CO_2, since these are the dominant gas species in a backed ultra-high vacuum (UHV) system.

These assumptions allow to solve analytically the stationary equations presented in 2.1 in 1-dimension.

As presented in Refs. [1] and [2], the main equation governing the stationary behavior of the vector volume density $\vec{n} = (n_{H_2}, n_{CH_4}, n_{CO}, n_{CO_2})$ is

$$0 = \vec{\vec{c}}_{spec}\frac{\partial^2 \vec{n}}{\partial x^2} + \vec{\vec{\eta}}_i \vec{\vec{\sigma}}_{i-I}\frac{I}{e}\vec{n} - \vec{\vec{\sigma}}_{i-I}\vec{n} - (\vec{\vec{S}}_{wall} + \vec{\vec{C}}_{dis})\vec{n} - \vec{\vec{S}}_{cryo}(\vec{n} - \vec{n}_e) + \vec{\vec{\eta}}_{ph}\dot{\Gamma}_{ph} + \vec{\vec{\eta}}_e \dot{N}_e + a\vec{q}_{th} \quad (2.1)$$

The definition, dimension and units of the quantities in Eq. 2.1 are defined in Tab.

2.3 Electron stimulated desorption

The electron-stimulated desorption (ESD), the desorption process initiated by electronic excitation, of atoms and molecules is an important factor in determining the pressure profile under beam induced effects.

In order to empirically characterize this effect for different gases, the so-called ESD yield, η_e is defined as:

$$\eta_e = \frac{N_i}{N_e} \quad (2.2)$$

where N_i is the number of desorbed molecules of a given gas specie and N_e is the number of incident electrons.

Magnitude	Units	Dimension	Definition
$\vec{n}$	m^{-3}	4×1	Molecular density of the different gas species
$\overleftrightarrow{c}_{spec}$	m^4/s	4×4	Specific conductance. (Diagonal matrix)
$\overleftrightarrow{\eta}_i$	-	4×4	Ion stimulated desorption yield
$\overleftrightarrow{\sigma}_{i-I}$	m^2	4×4	Ionization cross section of the gas-proton interaction
$\overleftrightarrow{S}_{wall}$	$\frac{m^2}{s}$	4×4	Distributed pumping speed per unit length due to NEG $S^i_{wall} = \alpha(T)\frac{a\cdot\bar{v}_i}{4}$
$\overleftrightarrow{C}_{dis}$	$\frac{m^2}{s}$	4×4	Distributed pumping through holes per unit length
$\overleftrightarrow{S}_{cryo}$	$\frac{m^2}{s}$	4×4	Cryogenic pumping due to cryo-condensation, per unit length
$\vec{n}_e$	m^{-3}	4×1	Thermal equilibrium density
$\overleftrightarrow{\eta}_{ph}$	-	4×4	Photon stimulated desorption yield (diagonal matrix)
$\dot{\Gamma}_{ph}$	$\frac{1}{m\,s}$	1×1	Photon flux to the wall per unit length
$\overleftrightarrow{\eta}_e$	-	4×4	Electron stimulated desorption yield (diagonal matrix)
$\dot{N}_e$	$\frac{1}{ms}$	1×1	Electron flux to the walls per unit length
a	m		Surface area per unit length
$\vec{q}_{th}$	$\frac{1}{m^2 s}$	4×1	Thermal outgassing. (The user introduces the thermal outgassing in mbar l/s)
I	A	1×1	Proton beam current
e	C	1×1	Charge of the electron

Table 2.1: Symbol, units and description of the magnitudes used in PyVASCO's model.

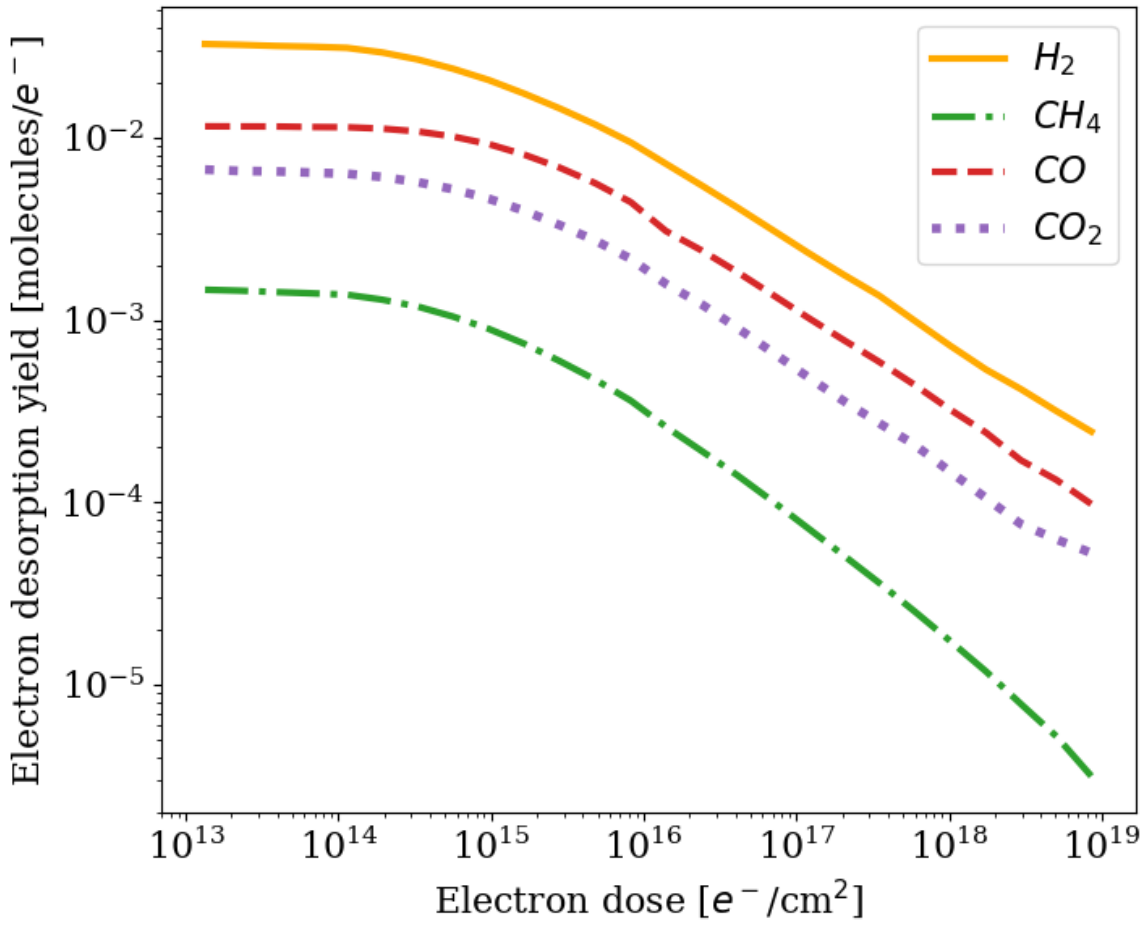

Figure 2.1: ESD curve for backed copper.

The ESD yields of different gases depend on the properties of the surface where the molecules of the studied gases are adsorbed, on the temperature and on gas specie.

The curve representing the ESD yields for a material as a function of the accumulated electron dose is the ESD curve for that material, and it has been observed that the ESD yields for different gases on materials relevant for UHV systems decrease with the accumulation of incident electron dose (in electrons/cm^2), as presented in Fig. 2.1 for backed copper.

This phenomenon of decrease of the ESD with the accumulated electron dose received in the walls is typically called *conditioning effect* (or *scrubbing effect*).

This phenomenon is relevant for the vacuum performance of UHV systems under electron bombardment due to beam induced effects, like the LHC, which performs dedicated scrubbing runs at the beginning of operation periods [3].

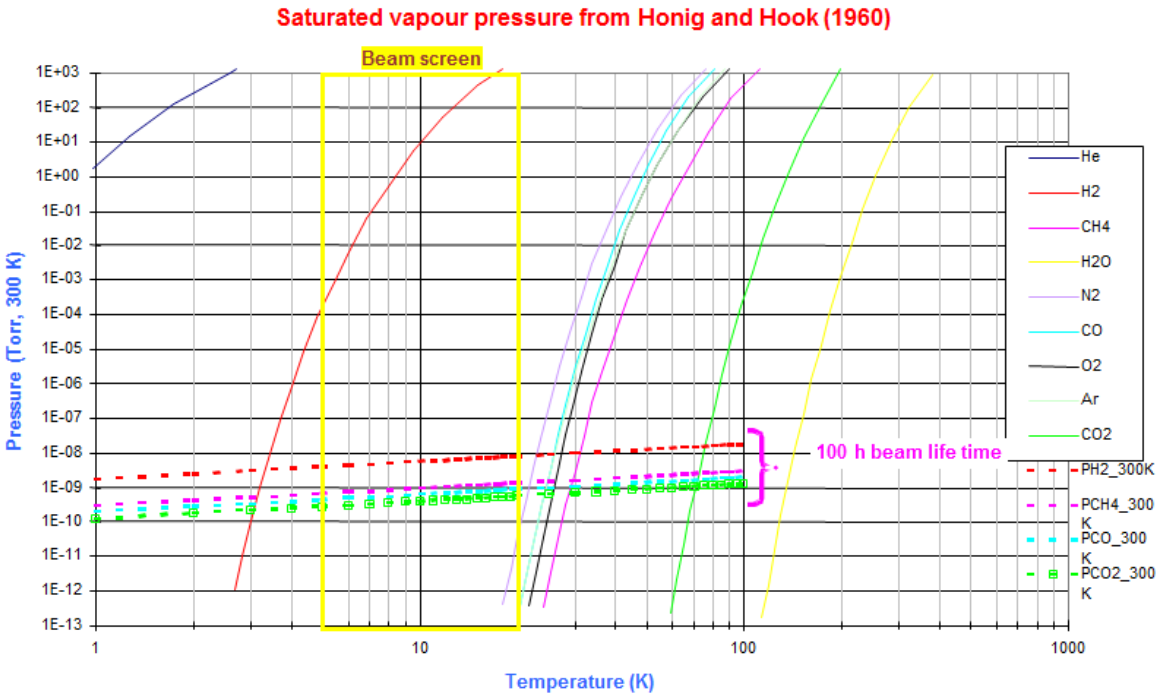

Figure 2.2: Equilibrium vapor pressure for different gases relevant in UHV.

2.4 Treatment of cryogenic surfaces

Basics concepts

The molecules of the gas species present in an UHV system mainly interact with the surfaces of vacuum chambers through van der Waals forces.

The binding energies between the surface and the gas molecules increases when the temperature of the system decreases, increasing the number of adsorbed molecules. For sufficiently low temperatures, the surface coverage increases sufficiently for the van der Waals forces to start acting between the molecules themselves. This regime is called cryocondensation. Once in the cryocondensation regime at a fixed temperature, a dynamic equilibrium might be reached between the adsorbed and desorbed molecules of a given gas specie, which translates to an equilibrium pressure of the gas over its condensed phase, called the equilibrium vapor pressure (Fig. . 2.2).

The sticking coefficient for cryosorption and cryocondensation is defined as

$$\alpha = \frac{\text{Number of molecules "sticking" on a surface}}{\text{Total number of molecules impinging on the surface}} \tag{2.3}$$

The sticking coefficient ignores the effects of the vapor pressure and it is close to unity at sufficiently low temperatures.

A cold surface acts as a pump with the characteristic pumping speed pre-

sented in Eq. 2.4:

$$S_i^{cryo} = \alpha_i \frac{A}{4} \bar{v}_i, \tag{2.4}$$

where the α_i is the sticking coefficient for the gas specie i, A is the area of the cold surface and $\bar{v}_i$ is the average molecular speed of the gas specie i.

Cryogenic surfaces in PyVASCO

As explained in Sec. 3.2, PyVASCO divides a simulation in a main input describing the geometry and in basic components of a vacuum system, i.e., Materials, Pumps and Gasources. In order to avoid confusion, the properties of all materials are defined at room temperature and a cryogenic behavior is set by default. In order to change the cryogenic behavior of a given material, see ...

> IMPORTANT!: For temperatures below 100 K, PyVASCO doesn't consider thermal outgassing (it is set to zero). However, the value of the thermal outgassing is not changed with temperature. Therefore, in order to simulate a system at a temperature considerably larger than room temperature, the user has to modify the outgassing in the definition of the material or to define another material with the correct outgassing for the temperature of interest.

Below 100 K, PyVASCO computes the distributed pumping speed of the cryogenic surfaces as in Eq. 2.4, using for the sticking factor the corresponding value as a function of the temperature set for the corresponding material (in ...) . If this information is not provided, the following default options are assumed:

- The dependence of the sticking factor with temperature for CH_4, CO and CO_2 are assumed to be the step functions presented in Fig.
- At temperatures below 20 K, the values of the sticking factor for H_2 as a function of the temperature are taken from Ref. [4]. Due to the scarce experimental data concerning the sticking coefficient of H_2 on technical surfaces at cryogenic temperatures, the choice of the reference has been arbitrary.

As can be seen in Eq. , the cryogenic pumping speed used in the vacuum model implemented in PyVASCO is modulated by the equilibrium density, n_e. This term takes into account the reduction of the efficiency of the cryogenic pumping due to the desorption of molecules from the considered surface. This term also implies that the cryogenic pumping becomes null when

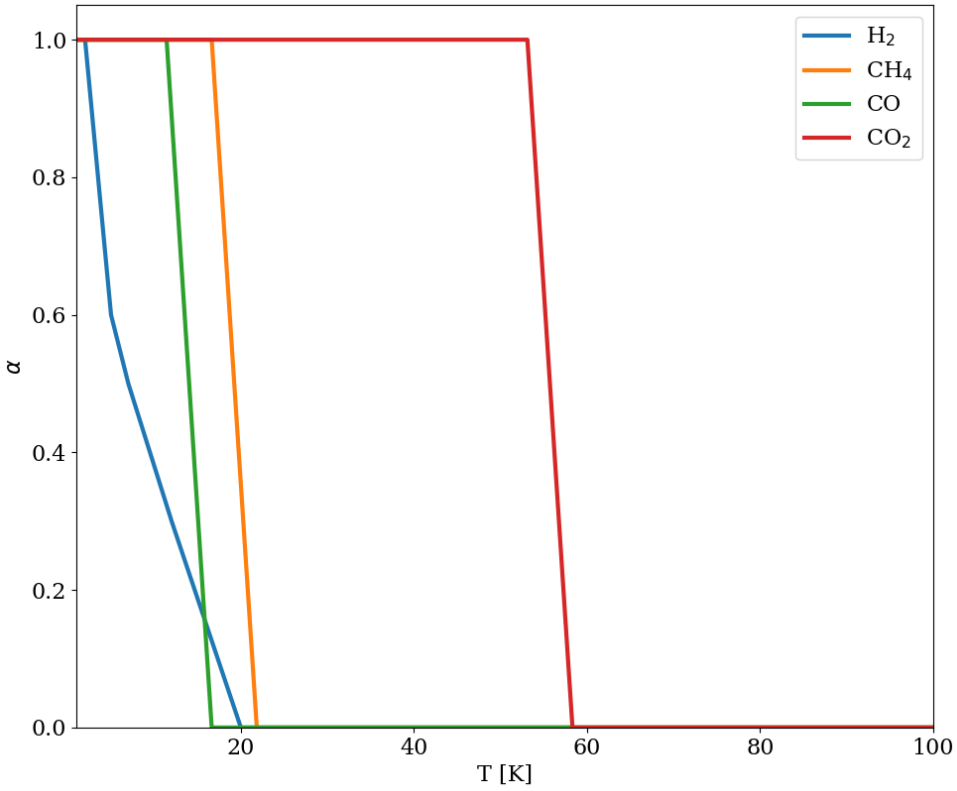

Figure 2.3: Default dependence of the sticking coefficient with temperature at low temperatures.

the computed gas density in the considered segment equals the equilibrium density for the temperature of the segment.

For a given temperature, PyVASCO estimates the values of the equilibrium density for the considered gas species for using the curves of Fig. 2.2, truncated at a certain pressure to ensure that n_e is always smaller than the computed density.

Chapter 3

Inputs of PyVASCO

3.1 'Old' Input format

As already mentioned, PyVASCO is based in VASCO, thus 'old' input format refers to VASCO's input format, detailed in [1].

To ease the comparison between VASCO and PyVASCO, PyVASCO can accept CSV files written with the format of the first program, as mentioned in Subsection 4.2 and transform this files to the native format of PyVASCO.

3.2 'New' Input format

Writing input simulation files with a large number of segments in VASCO's input format might be tedious and the it might be difficult to detect mistakes. For this reason, PyVASCO uses a new input format, which seeks to make an input file more readable.

A simulation in PyVASCO is built through a main simulation table plus three different constituents: Materials, Pumps and Gas sources. These three components are defined and stored as independent building blocks, making it possible to reuse them in different simulations.

Main input file

A geometric model in PyVASCO is built by cylindrical segments stuck together. An example of input file with the new format is shown in Tab. 3.1.

Input files in this format can be written using the integrated Input editor (in the menu *Add* → *Input file* or by pressing Ctrl+I) or in spreadsheet programs (like Excel) or in a plain text editor (like WordPad or Notepad). An example of how a simple input file would look like when using the integrated editor is shown in Fig. 3.1.

The new input format consists in:

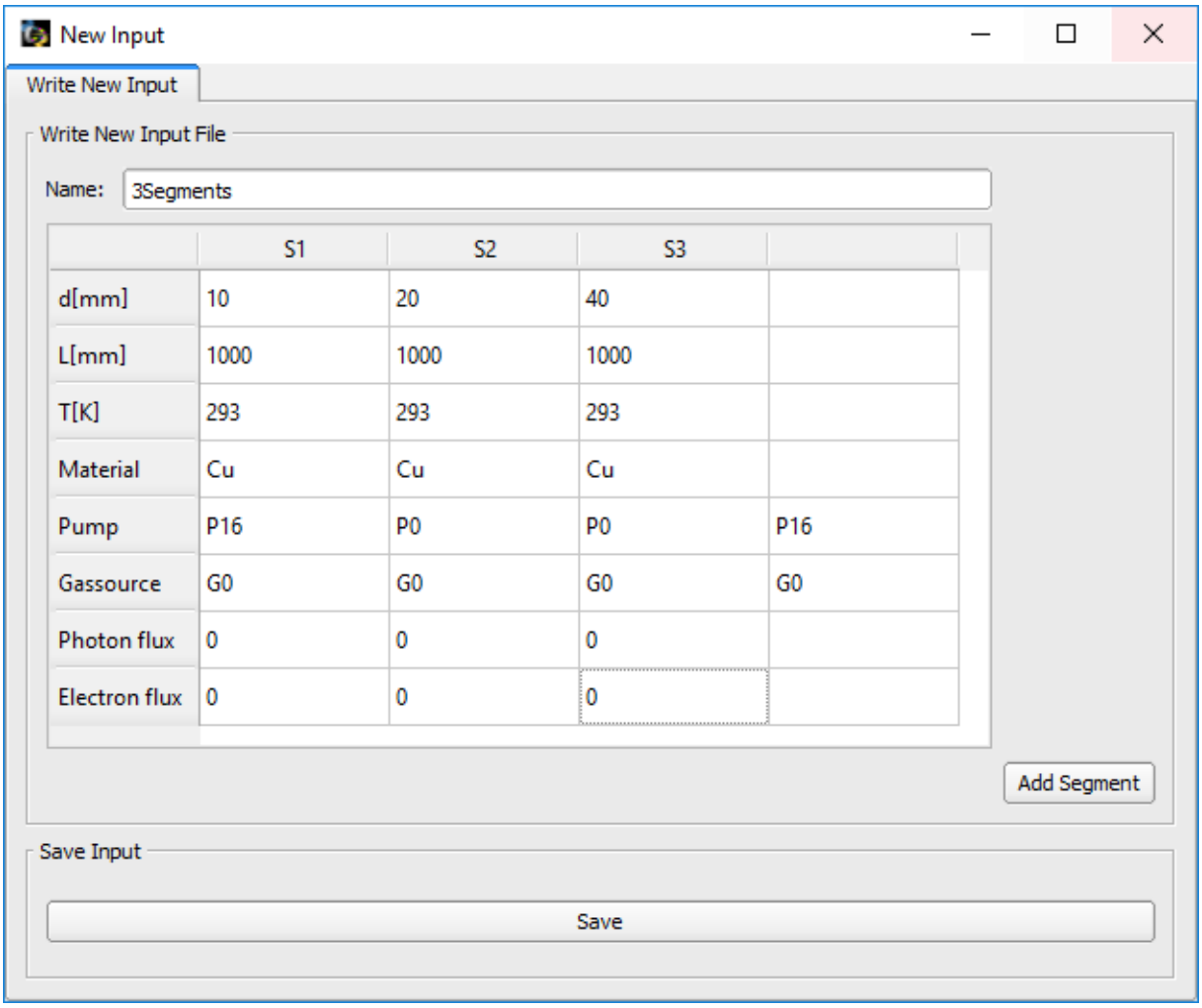

Figure 3.1: Input example written in the integrated editor.

Name of the model

3Segments	S1	S2	S3	
d[mm]	10	20	40	
L[mm]	1000	1000	1000	
T[K]	293	293	293	
Material	Cu	Cu	Cu	
Pump	P16	P0	P0	P16
Gassource	G0	G0	G0	G0
Photon flux	0	0	0	
Electron flux	0	0	0	

Segment 1 Segment 2 Segment 3 Right side of end segment

Table 3.1: Example of input file written in the 'new' format.

- **Name of the simulation:**
  The name of the simulation must be written in the dedicated line edit when using the integrated editor or, when using external editors like Excel, it must be written in the first row, first column of the input file.

- **Columns, labeled S_1...S_N, for the defined segments:**
  Each column labeled S_1...S_N in the input file represents a segment in the simulated geometry. For each segment the following information has to be provided:

  - **d[mm]**: The diameter of the segment (in mm), assumed to be cylindrical.
  - **L[mm]**: The length of the segment (in mm).
  - **T[k]**: The average temperature of the segment.
  - **Material**: Name of the material of the segment. The list of registered materials can be visualized in the menu *File* → *Show Components* or pressing Ctrl+S. (See Show components in Section 4.1 and Materials in Section 3.2 for more details).
  - **Pump**: The pumps specified in the main input file are lumped pumps located on the left side of the segment where they are indicated. The list of registered pumps can be visualized in the menu *File* → *Show Components* or pressing Ctrl+S. In order to simulate the union of two segments without a lumped pump in between, P0 has to be written in the corresponding cell.
  - *Gas source*: The gas sources in the main input file represent localized leaks in the interconnections between segments and, in particular, located on the left side of the segment where they are indicated.The list of registered pumps can be visualized in the menu *File* → *Show Components* or pressing Ctrl+S. In order indicate to PyVASCO that no leak exists in a certain location, G0 has to be written in the corresponding cell.
  - **Photon flux**: PyVASCO accepts an homogeneous, constant photon flux (in photons/m/s) impinging the walls of every segment.
  - **Electron flux** An homogeneous, constant electron flux (in electrons/m/s) can be added to the different simulated segments.

- **End column:**
  This column is exclusively used to indicate the lumped pump and the gas source located at the right side of the last simulated segment.

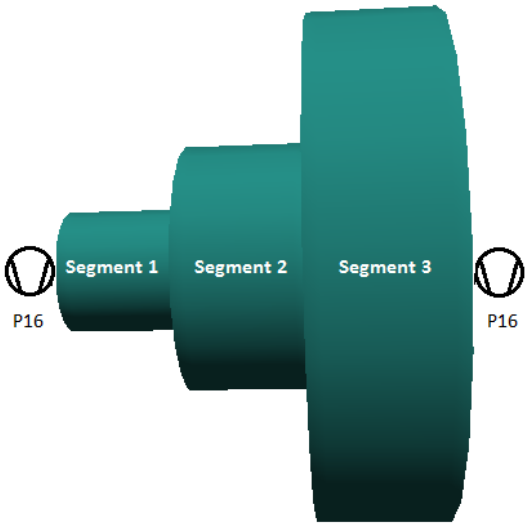

Figure 3.2: 3D model of the simulated geometry presented in Tab. 3.1. (For aesthetic reasons, the geometry has been shrinked in the longitudinal direction).

Thus, the example of Fig. 3.1 and Tab. 3.1, is composed by 3 copper segments of the same same length and increasing diameters of 10, 20 and 40 mm, respectively. The three segments are hold at room temperature and two lumped pumps (called P16) are connected at the beginning of the first segment (left side) and at the end of the third segment (right side). There are no lumped pumps connected nor on the right side of the first segment nor on the right side of the second segment, and there are no leaks along the geometry. A 3D model of the described system can be seen in Fig. 3.2.

IMPORTANT!: If the user writes the main input in an external editor, the same format as shown in Tab. 3.1 has to be used. The input file has to be saved in CSV format.

Materials

The materials used in the main input file for the PyVASCO simulations have to be defined previously to their usage. PyVASCO offers the possibility of defining new materials in the dedicated Material editor, but the user can also import a CSV file with the format shown in Tab. 3.2.

A material file defined in PyVASCO consists in:

- **Name of the material**
  The name with which this material will be called in the main input files.

- **Colums:**
  A material file always have 4 data columns, corresponding to the behavior of the defined material with respect to the main dominant gases in UHV, i.e. H_2, CH_4 CO and CO_2.

- **Rows:**
  The different rows specified in the a materials file are:

  - **alpha**: Sticking factor or sticking coefficient (adimensional).This quantity represents the probability which each of the defined gases have of sticking onto the surface of the segment. In the case of the LHC, this parameter is used to represent the pumping due to NEG in the warm sections and to physisorption (cryopumping) in the cold sections.

    IMPORTANT!: The sticking coefficient defined in this section is always defined at room temperature! PyVASCO scales the sticking coefficient with the temperature for the defined cryogenic behavior of the material (See , for more information).

  - **eta_ion:** Ion stimulated desorption yields $\vec{\eta}_I$ (in molecules/incident ion) at a chosen ion impact energy (4×4 matrix, occupying from row 2 to row 5).

  - **eta_e:** Electron stimulated desorption yields (in molecules/ incident electron) at a chosen impact energy.

  - **eta_ph:** Photon stimulated desorption yields (in molecules/incident photon) at a chosen photon energy.

  - **Cbs:** distributed pumping speed per unit length (in $l \cdot s^{-1} \cdot m^{-1}$). In the case of the LHC, this input parameter can be used to simulate the pumping through pumping slots.

  - **Qth_total:** Thermal outgassing rate per unit area at a chosen temperature (in $mbar \cdot l \cdot s^{-1} \cdot cm^{-2}$).

IMPORTANT!: If the user writes a material file in an external editor (Excel, for example) the name of the material written in the first row and column of the material table has to match the name of the file. Moreover, the material file has to be saved in CSV format.

Cu	H2	CH4	CO	CO2
alpha	1.00E-12	1.00E-12	1.00E-12	1.00E-12
eta_ion	0.54	0.54	0.54	0.54
	0.04	0.05	0.07	0.11
	0.25	0.29	0.29	0.33
	0.14	0.14	0.14	0.14
eta_e	0.04	0.001	0.003	0.003
eta_ph	0.00015	4.00E-06	1.50E-05	2.50E-05
Cbs	3377.03	1.18E+02	1.04E+03	1.19E+03
Qth_total	6.66E-07	1.58E-09	9.30E-09	2.14E-09

Table 3.2: Example of defined material in PyVASCO.

IMPORTANT!: All the properties defined for a certain material depend on the temperature! For this reason it is important to register the same material held at different temperatures as different entries by including the temperature in the definition name. For example: use Cu@RT and Cu@5K to define copper at room temperature and at 5 K, respectively.

IMPORTANT!: The outgassing rate of a given material is internally converted to total outgassing by multiplying this quantity by the surface area of the cylindrical segment considered. If you are trying to simulate a geometry which considerably differs from a cylinder, it might turn out that the real outgassing area is much bigger than the computed area, and you should scale the outgassing rate accordingly to give the real total outgassing when multiplied by the computed area.

Pumps

The lumped pumps used in the main input file for the PyVASCO simulations have to be defined previously to their usage. The same pump in PyVASCO can present different pumping speeds for different pressure ranges. PyVASCO offers the possibility of defining new pumps in the dedicated Pump editor, but the user can also import a CSV file with the format shown in Tab. 3.3. However, in the later case only simple pumps (with pumping speeds for the different considered gas species independent of the pressure range) can be defined.

The pumping speed for each of the gas species has to be in l/s.

P1	
S_H2 [l/s]	335
S_CH4 [l/s]	15
S_CO [l/s]	120
S_CO2 [l/s]	120

Table 3.3: Example of a defined simple pump in PYVASCO.

G1	
g_H2 [mbar l/s]	6.00E-09
g_CH4 [mbar l/s]	6.00E-11
g_CO [mbar l/s]	7.50E-10
g_CO2 [mbar l/s]	7.50E-10

Table 3.4: Example of a defined local gas release (gas source) in PyVASCO.

Gas sources

The gas sources used in the main input file for the PyVASCO simulations have to be defined previously to their usage. PyVASCO offers the possibility of defining new gas sources in the dedicated Gas source editor, but the user can also import a CSV file with the format shown in Tab. 3.4.

The gas release for each of the gas species has to be in mbar l/s.

3.3 ESD curves format

In order to easily quantify the impact of the reduction of the ESD yields with the accumulated electron dose, PyVASCO offers the possibility of solving the dynamic vacuum model for different accumulated electron doses. (See Dynamic pressure due to ESD in Section 4.2, for more details on the simulation).

The ESD input files for PyVASCO must be CSV files containing 5 columns:

- The first column must be labeled *DOSe/cm2*, and contain the accumulated electron dose (in electrons/cm^2).
- The second to fifth columns must include the ESD yields of H_2, CH_4, CO and CO_2, respectively.

  An example of the format for a ESD curve in PyVASCO can be seen in Tab. 3.5

DOSe/cm2	H2	CH4	CO	CO2
1.34E+13	0.0326625	0.001466961	0.0115575	0.006693333
2.28E+13	0.03234	0.001449682	0.0115425	0.006576667
3.90E+13	0.03189	0.001424937	0.01153	0.006536667
6.66E+13	0.0315825	0.001402831	0.0114575	0.00644
1.14E+14	0.0311275	0.001378118	0.011425	0.00635
1.94E+14	0.0293675	0.00129371	0.0112225	0.006093333
3.32E+14	0.0268725	0.001186265	0.010825	0.005706667
5.66E+14	0.0238925	0.00104455	0.01015	0.00522
9.67E+14	0.0207325	0.00089523	0.00923	0.00463
1.65E+15	0.0174675	0.000742215	0.0080825	0.00398
2.82E+15	0.0144875	0.000595861	0.0068575	0.00332
4.82E+15	0.0118225	0.000471321	0.0056325	0.00273
8.23E+15	0.0094375	0.000361661	0.0044375	0.002166667
1.40E+16	0.0072	0.000259877	0.00308	0.001578909
2.40E+16	0.0054775	0.000191541	0.00241	0.001220288
4.10E+16	0.00415	0.000141125	0.0018375	0.000907256
7.00E+16	0.0031175	0.000101452	0.00137686	0.000664671
1.19E+17	0.002345	7.23E-05	0.00103295	0.000487825
2.04E+17	0.001775	5.13E-05	0.000778363	0.000360405
3.48E+17	0.001365	3.63E-05	0.000591423	0.000272406
5.95E+17	0.00099326	2.53E-05	0.00044162	0.000207247
1.02E+18	0.000723494	1.74E-05	0.000322753	0.000147206
1.75E+18	0.00053548	1.18E-05	0.000241401	0.000105682
2.96E+18	0.000418254	7.84E-06	0.000170316	7.60E-05
5.06E+18	0.000317149	5.21E-06	0.0001329	6.27E-05
8.56E+18	0.000244796	3.15E-06	9.80E-05	5.29E-05

Table 3.5: Example of the format for an ESD curve required by PyVASCO.

IMPORTANT!: In order to properly run this simulation, all the materials used in the geometry model must have an associated ESD curve. To associate an ESD curve to a given material, select the option *Add* → *ESD curve* in PyVASCO menus or press Ctrl+D. See ESD curve in Subsection 4.1 for more details on how to use this option.

Chapter 4

Layout and functionality

4.1 Menus

The current version of PyVASCO (2.0) presents 4 menus, named: File, Add, Analysis and Help. In this section, a detailed description of the different menus in PyVASCO and their functionality is provided.

File

The File menu of PyVASCO contains 4 options:

- **Load...**:
  This option reloads all the registered materials, pumps and gas sources when selecting it or pressing the keyboard key combination Ctr+L.

- **Properties**:
  When selected or on pressing the keyboard key combination Ctr+P, this option will launch the *Properties* window, shown in Fig. **??**. The properties window allows to select the pressure unit (mbar or torr) of the input. The native pressure unit of PyVASCO is mbar, while the input pressure unit in VASCO is torr. This option was added in order to ease the benchmark between both programs.

  IMPORTANT! : Please note that changing the pressure unit in the Properties window won't change the pressure unit in the output of the simulation (the results will still be given in mbar). Changing this value will only affect the interpreted units of all the gas sources and the thermal outgassing and the linear pumping of all materials.

Add and Edit

The Add menu of PyVASCO contains 5 options:

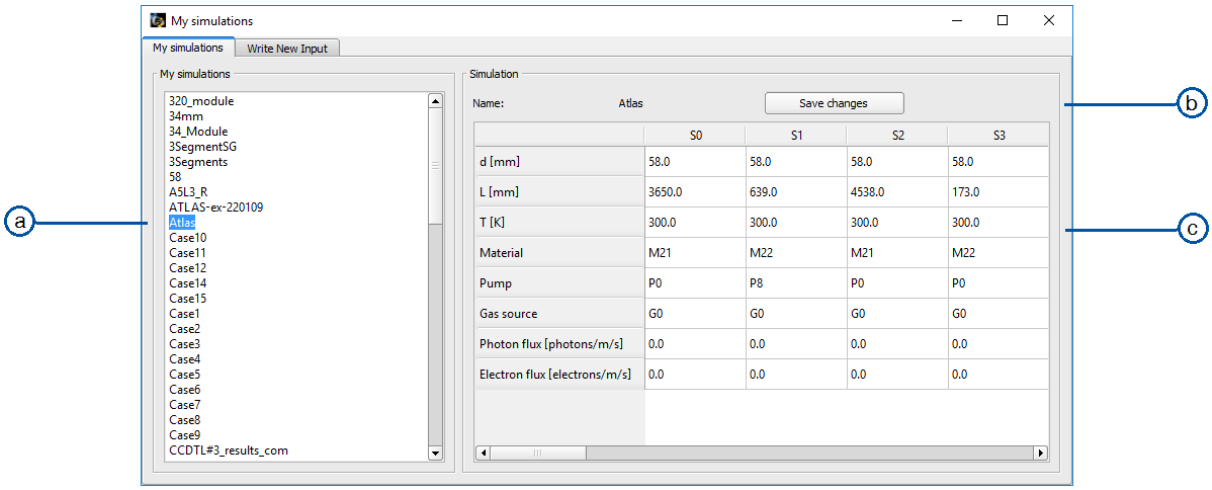

Figure 4.1: Tab 1 of "My Simulations" window.

- **Simulation**:
  When this option is selected or the keyboard key combination Ctrl+I is pressed, the *My simualtions* window will be launched. This window allows the user to view and edit the preexisting simulations in the first tab (Fig. 4.1) and to write a new input model in the second tab (Fig. 4.2).

  (a) List of the existing simulations. A single click on a list elements makes that it appears in the table on the right. A double click on a name makes it possible to rename a simulation.

  (b) Pressing the "button save changes" save the modifications done in the simulation.

  (c) Table showing the components of the simulation selected on the right. The elements of the table can be manually edited and the changes saved by pressing the Save button.

  (d) Defines the name of the simulation

  (e) Simulation components in the 'New format' (see Section 3.2 for a detailed explanation).

  (f) Allows to add a segment to the model.

  (g) Saves the new input under the name specified in (a) plus the suffix "_New" .

- **Material**:
  When this option is selected or the keyboard key combination Ctrl+M is pressed, the *New Material* window is launched. This window allows

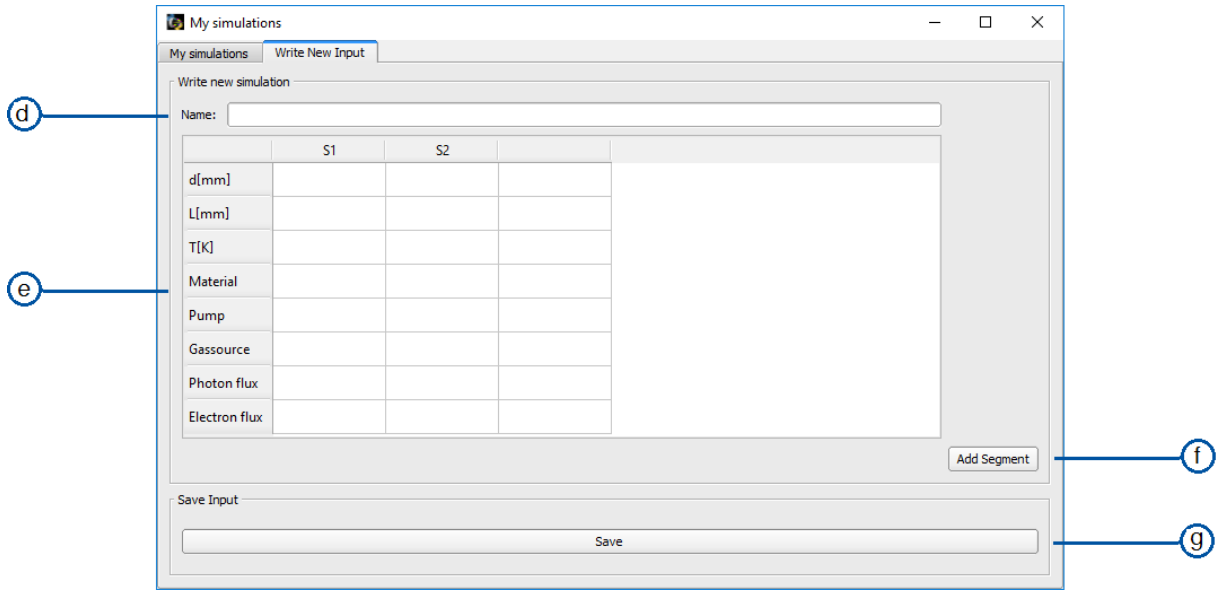

Figure 4.2: Tab 2 of "My Simulations" window.

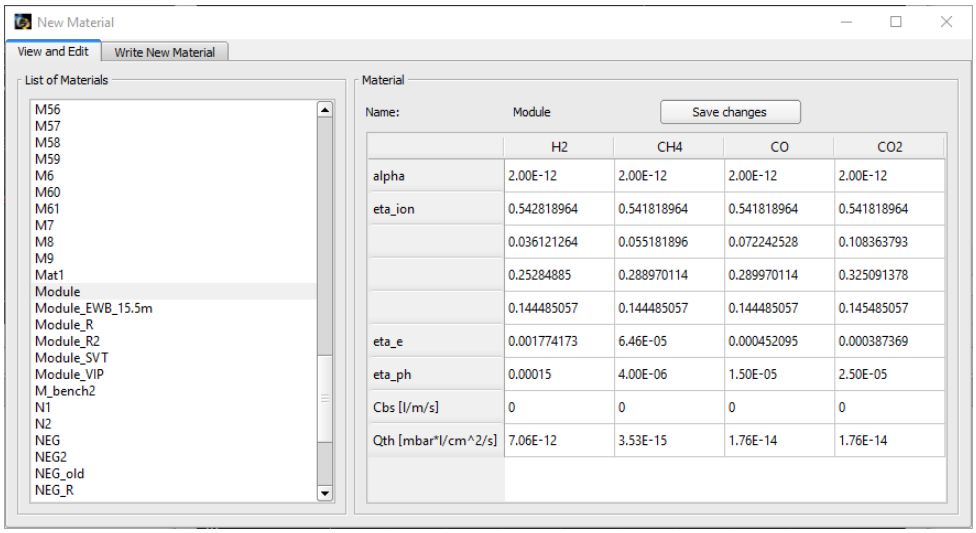

Figure 4.3: Tab 1 of the "New Material" window

the user to view the preexisting materials and to define a new material. It contains 2 tabs:

- **View and Edit**: (Fig. 4.3) with this tab, the user can see all the defined materials by selecting them in the list on the left, edit their names by double-clicking them in the list and edit their properties in the table. To save the changes done in a material, the user has to press the button "Save changes".
- **Write New Material**: (Fig. 4.4) with this tab, the user can define a new material in PyVASCO, which will be available for all the simulations once saved (pressing the 'Save' button).

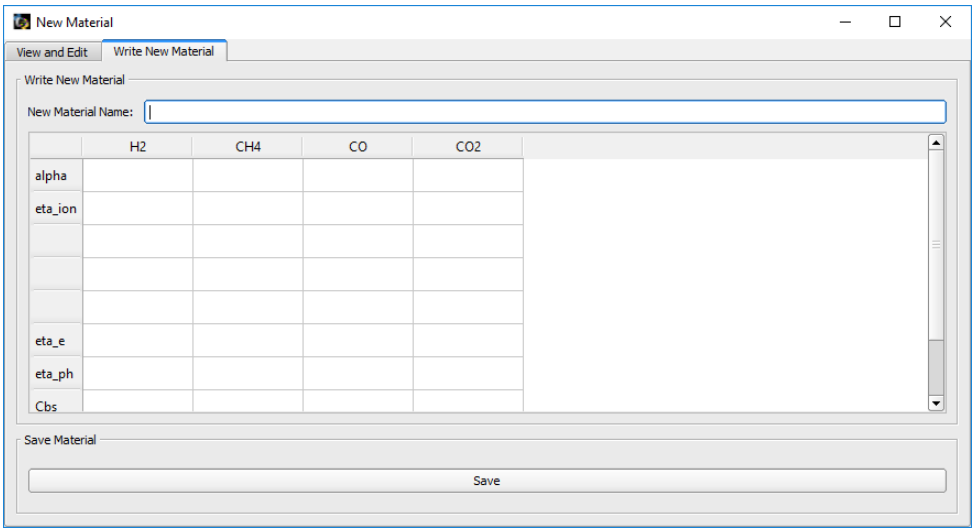

Figure 4.4: Tab 2 of the "New Material" window

- **Pump**:
  When this option is selected or the keyboard key combination Alt+P is pressed, the *New Pump* window will be launched. This window allows the user to define a new pump. It contains 2 tabs:

  - **View and Edit**: (Fig. 4.5) with this tab, the user can see all the defined pumps by selecting them in the list on the left, edit their names by double-clicking them in the list and edit their properties in the table. To save the changes done in a material, the user has to press the button "Save changes".
  - **Write New Pump**: (Fig. 4.6) with this tab, the user can define a new pump in PyVASCO, which will be available for all the simulations once saved (pressing the 'Save' button). Different pumping speeds can be associated to the same pumps for different pressure ranges by writing a pressure value (in mbar) in the second line edit and pressing the button 'Add pumping speed p [mbar]:'.

- **Gas source**:
  When this option is selected or the keyboard key combination Ctrl+G is pressed, the *New Gas Source* window will be launched. This window allows the user to view and edit preexisting gas sources and to define a new gas source. It contains 2 tabs:

  - **Data**: View and Edit: (Fig. 4.7) with this tab, the user can see all the defined pumps by selecting them in the list on the left, edit their names by double-clicking them in the list, and edit their proper-

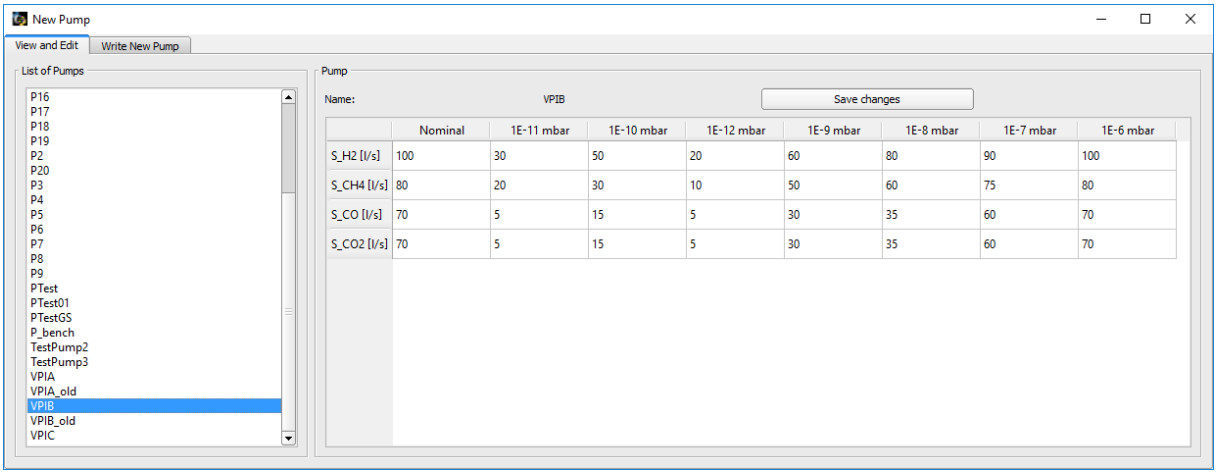

Figure 4.5: Tab 1 of the "New Pump" window.

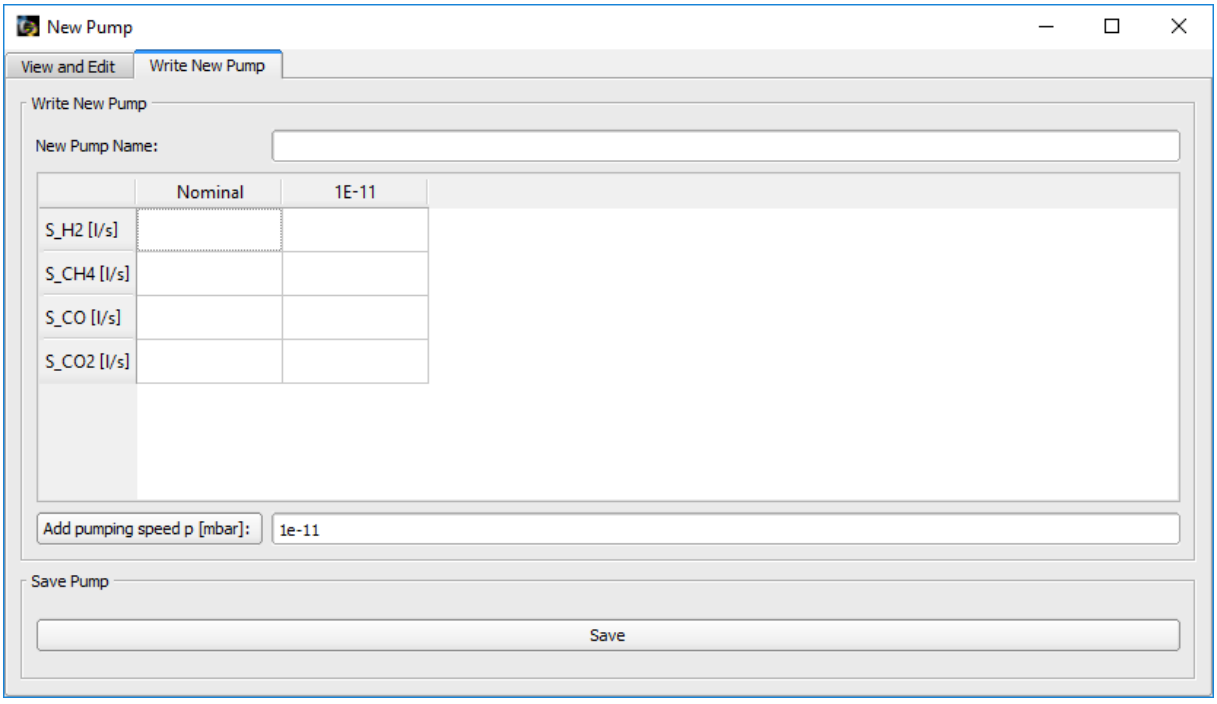

Figure 4.6: Tab 1 of the "New Pump" window.

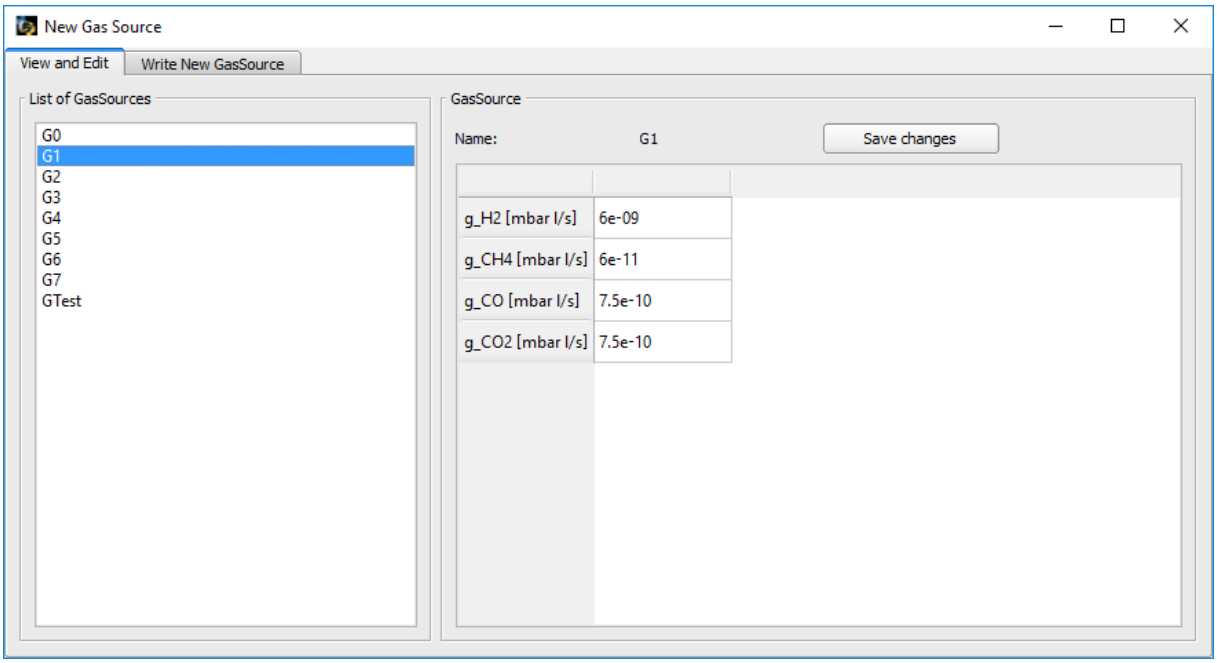

Figure 4.7: Tab 1 of the "New Gas SOurce" window.

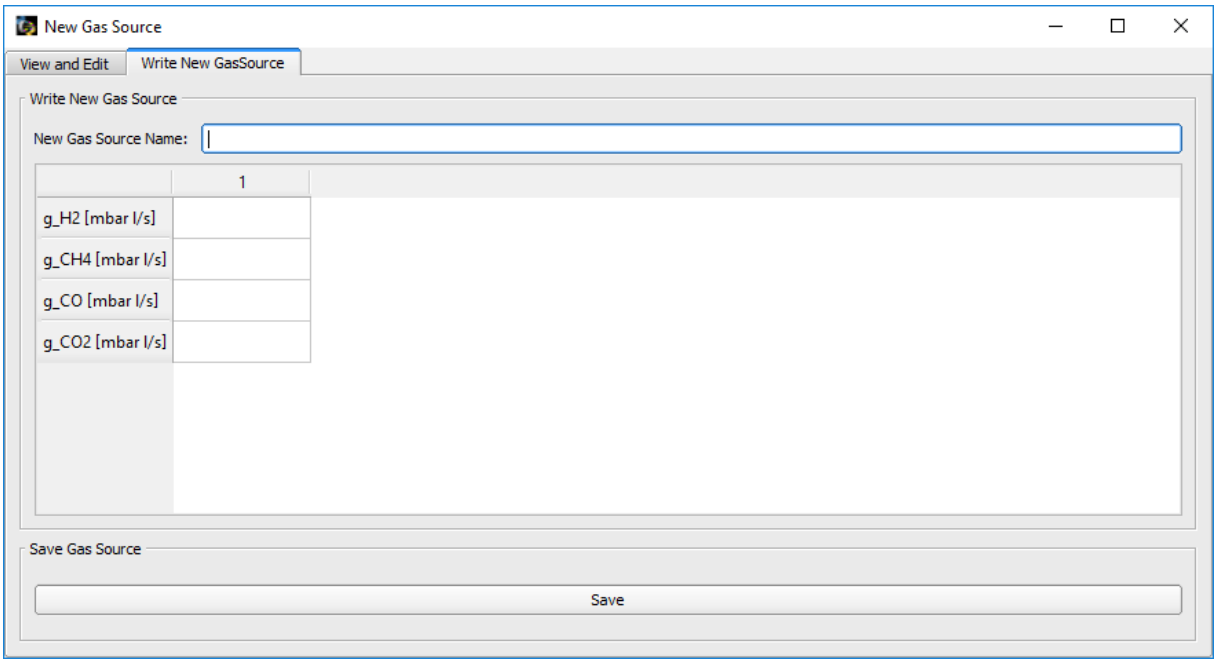

Figure 4.8: Tab 2 of the "New Gas SOurce" window.

ties in the table. To save the changes done in a material, the user has to press the button "Save changes".

– **Write New Gas Source**: (Fig. 4.8) with this tab, the user can define a new gas source in PyVASCO, which will be available for all the simulations once saved (pressing the 'Save' button).

- **ESD curve**:

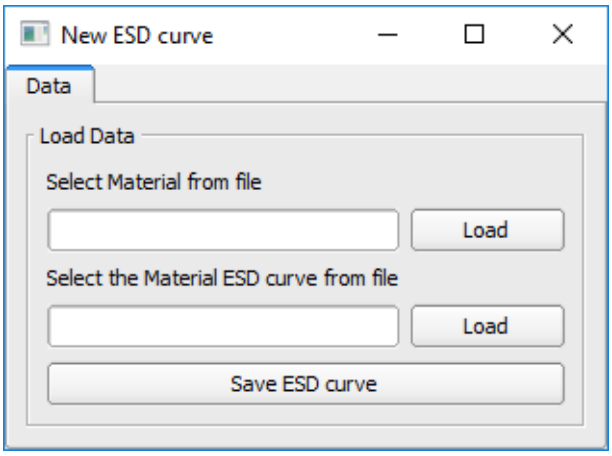

Figure 4.9: *New ESD curve* window.

When this option is selected or the keyboard key combination Ctrl+D is pressed, the *New ESD curve* window will be launched. This window, shown in Fig. 4.9 links an existing material with the experimental data of its corresponding ESD curve. See Section 3.3 for more information on the format of this file.

Analysis

The Analysis menu of PYVASCO launches the *Analysis menu* window. This window contains 3 tabs:

- **Analysis Configuration** (Fig. 4.10) and **Analysis and Comparison** (Fig. 4.11):
  These two tabs allow the user to upload two different simulation results in CSV format and plot them together in the *Analysis and Comparison* tab. In the **Analysis Configuration** tab, the user can select the result files clicking on the buttons 'Directory to Upload...', and has to manually indicate the format and units in those files using the format and unit dopdowns, and pressing 'Run Analysis'.

- **TDIS**:
  This tab was used to carry out the study on the TDIS presented in [5], and has been kept in order to ease the generation of the this results.

Help

The Help menu of PyVASCO contains 2 options:

- **User's guide**:
  When this option is selected or the keyboard key combination Ctrl+H

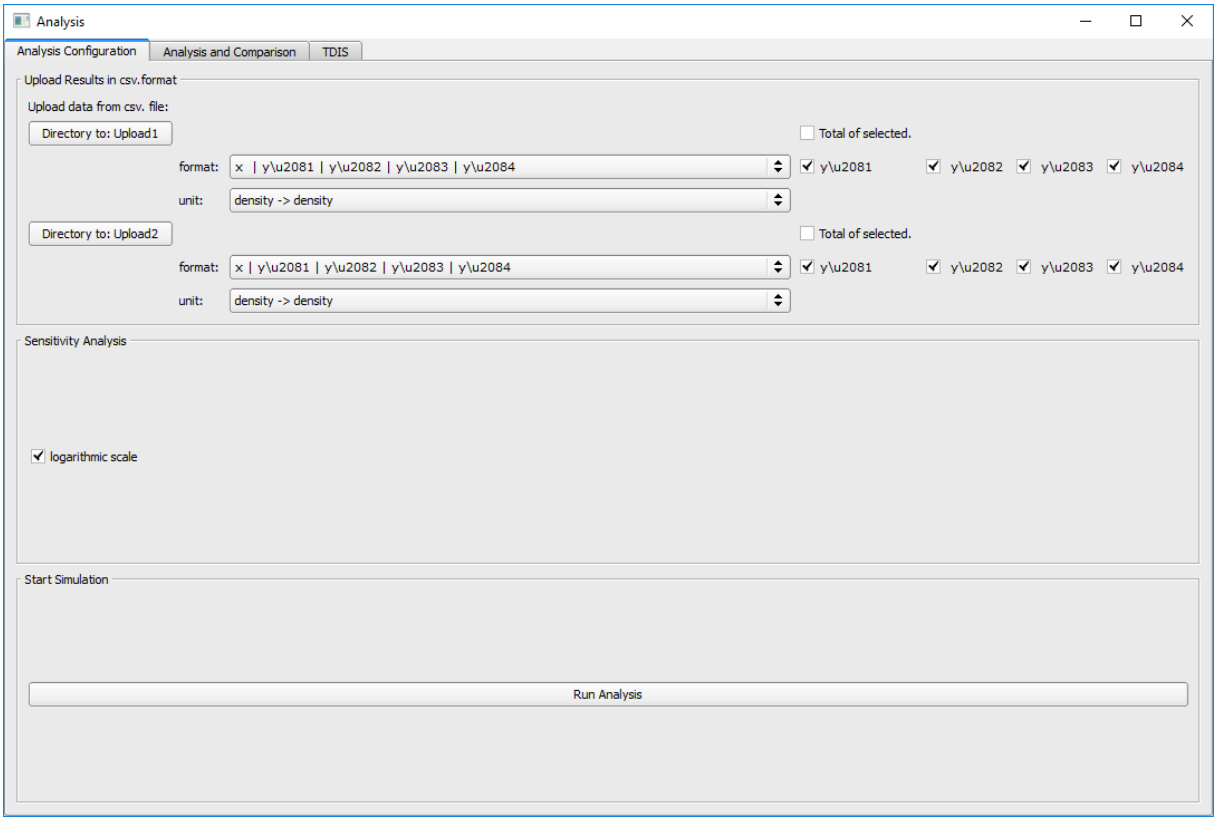

Figure 4.10: *Analysis* window, *Analysis Configuration* tab.

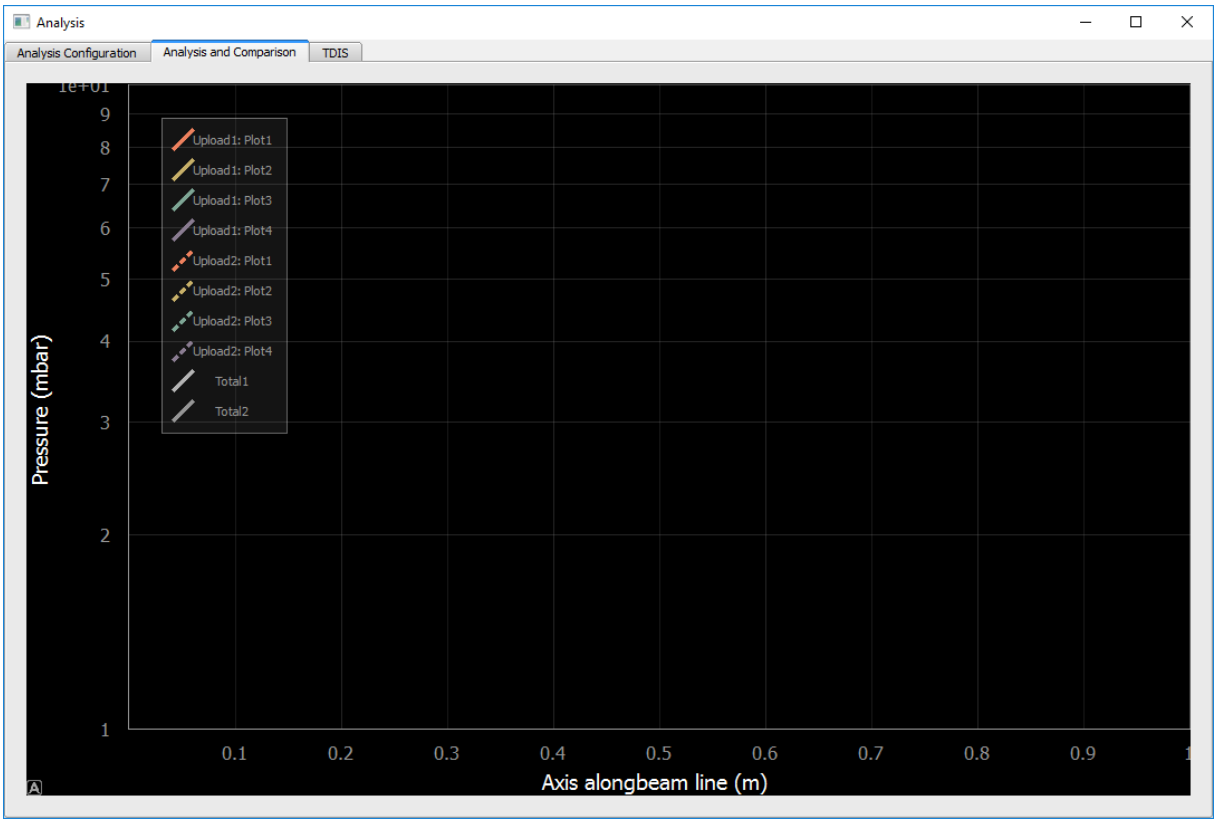

Figure 4.11: *Analysis* window, *Analysis and Comparison* tab.

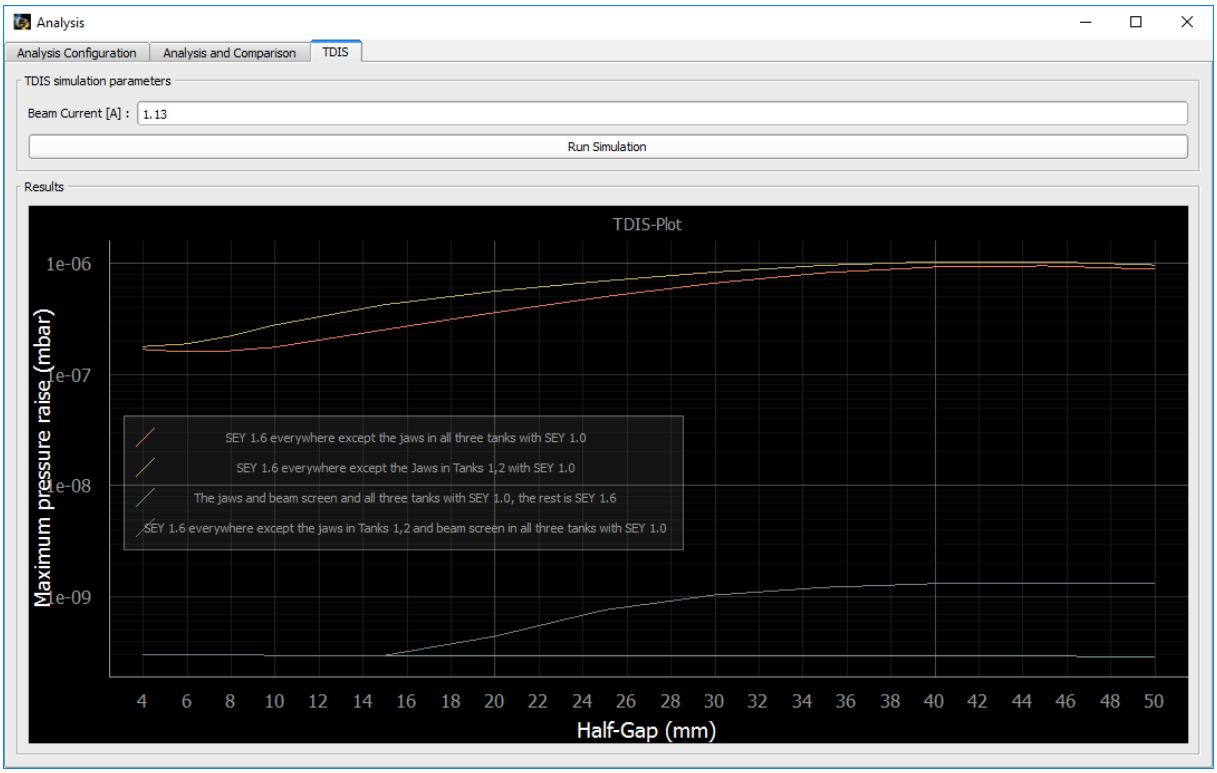

Figure 4.12: *Analysis* window, *TDIS* tab.

is pressed, the current document (PyVASCO User's guide) is launched and shown in the default web browser.

- **Documentation**:
  When this option is selected or the keyboard key combination Ctrl+U is pressed, the API documentation is launched in the default web browser.

4.2 Tabs

The current version of PyVASCO (2.0) contains 4 tabs, named: Data, Simulation, Critical Current and Dynamic pressure due to ESD, respectively. In this section, a detailed description of the different tabs in PyVASCO and their functionality is provided.

Data

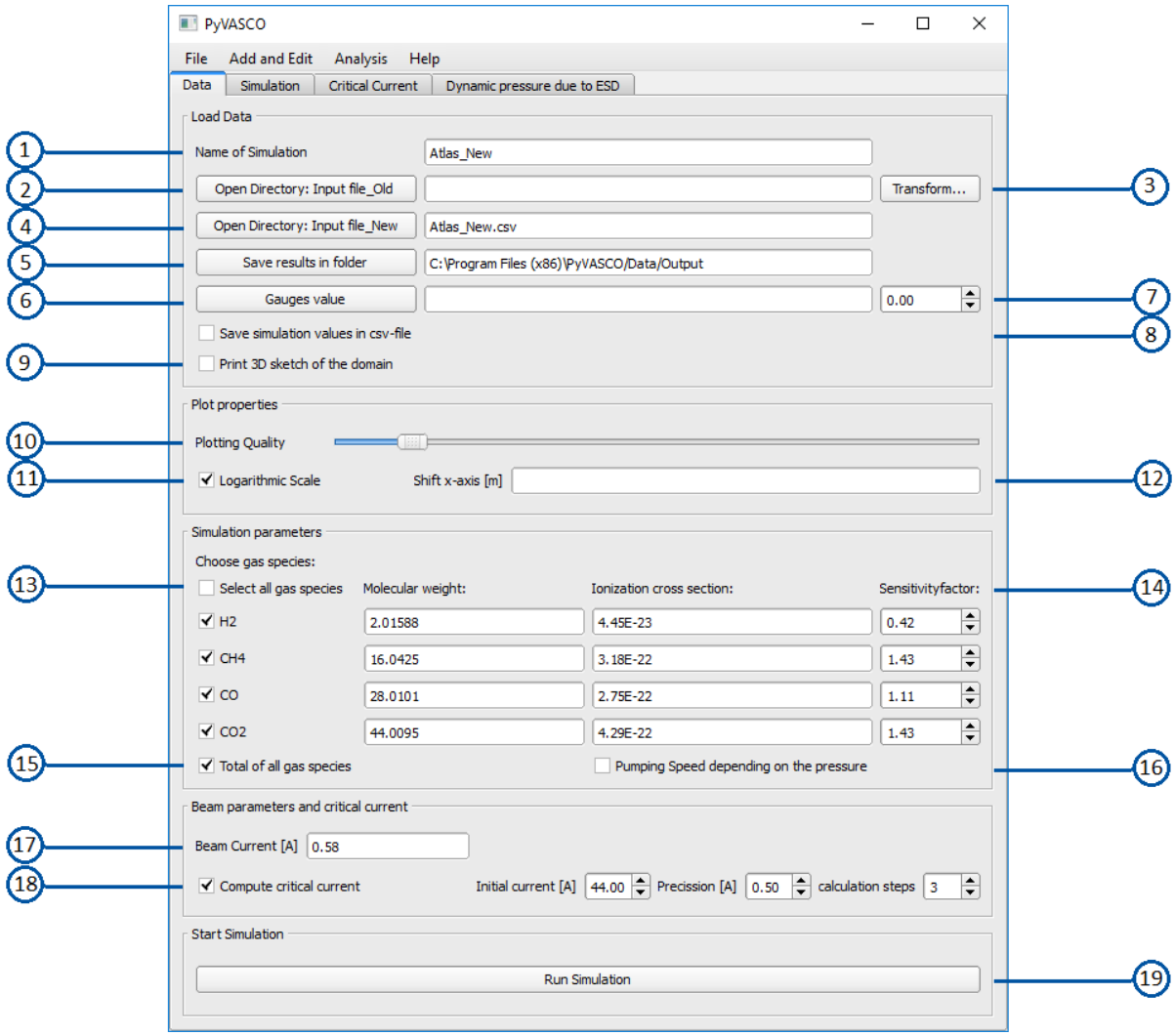

Figure 4.13: *Data* tab of PyVASCO.

The numbers in Fig. 4.13 represent:

1. **Name of the simulation**:
The name of the simulation is automatically set to the name of the input file selected in 4, and can be manually modified by the user. This name is used for the automatic saving in CSV format in the directory specified in 5 if option 8 is selected.

2. **'Old' format input file**:
As mentioned in Chapter 1 and Chapter 3, PyVASCO is based in VASCO, but the format of the input files has been changed in order to ease the writing of the input files for large simulations. However, it is still possible to upload a CSV input file written in the same format as the input in VASCO [1] with this option.

(3) **Transform to new format**:
After selecting an input file written with the same format as used in VASCO (see [1] and Chapter 3 for more details) in (2), this option allows to generate a new input file written in the native PyVASCO format containing the same information as the one previously selected and named as the old file with the suffix "_New". The new input file is saved in the default input directory of PyVASCO, i.e., *Data/Input/*.

(4) **'New' format input file**:
Upload an input file in the native format of PyVASCO (see Chapter 3 for more details).

(5) **Default output directory**:
If option (8) is selected, the result of the simulation will be automatically saved in the directory selected using this option under the name specified in (1).

(6) **Upload data from gauges**:
This option allows to upload experimental data from different gauges and plot it together with the simulation results in Simulation (see for more details on the format of the gauges data).

(7) **Shift the data from gauges**:
Typically, PyVASCO assumes that the geometry starts in $x = 0$ m, while the data from gauges extracted from, for example, the LHC, might start at a longitudinal coordinate (s) different from 0 m, depending on the reference point used. In order to effectively compare the simulation results with the experimental data in the tab Simulation, his option allows to shift horizontally the experimental data uploaded in (6). The number indicated in this slot corresponds to the shift in meters to the right (if the value is positive) or to the left (if the value is negative).

(8) **Save results in CSV format**:
If this option is selected, the results of the simulation, i.e., the molecular density of the different considered gas species considered (in m^{-3}) will be automatically saved in CSV format in the directory indicated in (5) under the name indicated in (1).

(9) **Print a 3D sketch of the geometry**:
If this option is selected, a 3D sketch of the simulated geometry will be saved in PNG format under the name indicated in (1) in the directory *Data/*.

(10) **Plotting quality**:
This option specifies the number of points with which the density profile for the different gas species is calculated and presented in tab Simulation.

(11) **Logarithmic scale**:
If selected, the Y axis of the density profile plot in tab Simulation is set to logarithmic scale.

(12) **Move horizontally**:
Similarly to (7), this option shifts horizontally the simulated density profile and the geometry. Thus, if a value different than 0 m is indicated, the geometry and the simulated density profile will be assumed to start at the indicated x coordinate (in m).

(13) **Gas species**:
This option allows to select the gas species to simulate and their ionization cross section (in m^2). The default values indicated for the ionization cross sections of the different gas species correspond to those calculated in [6] for a proton beam at 7 TeV.

(14) **Sensitivity factor**:
If option (15) is selected, the total pressure is computed using the specified sensitivity factors for each gas specie.

(15) **Total of all gas species**:
If this option is selected, the total pressure is computed using the sensitivity factors specified in (14) and plotted in the tab Simulation.

(16) **Variable pumping speed**: If this option is selected, the change in pumping speed of ion pumps for different pressure ranges will be taken into account. After performing an initial simulation with the nominal (maximum after saturation) pumping speed for the different gases, the pumping speed of the ion pumps located along the geometry is recalculated and the gas density is recomputed.

(17) **Beam current**:
Current of the circulating proton beam (in A). The default value is 0.582 A, which corresponds to the nominal average beam current in the LHC [7].

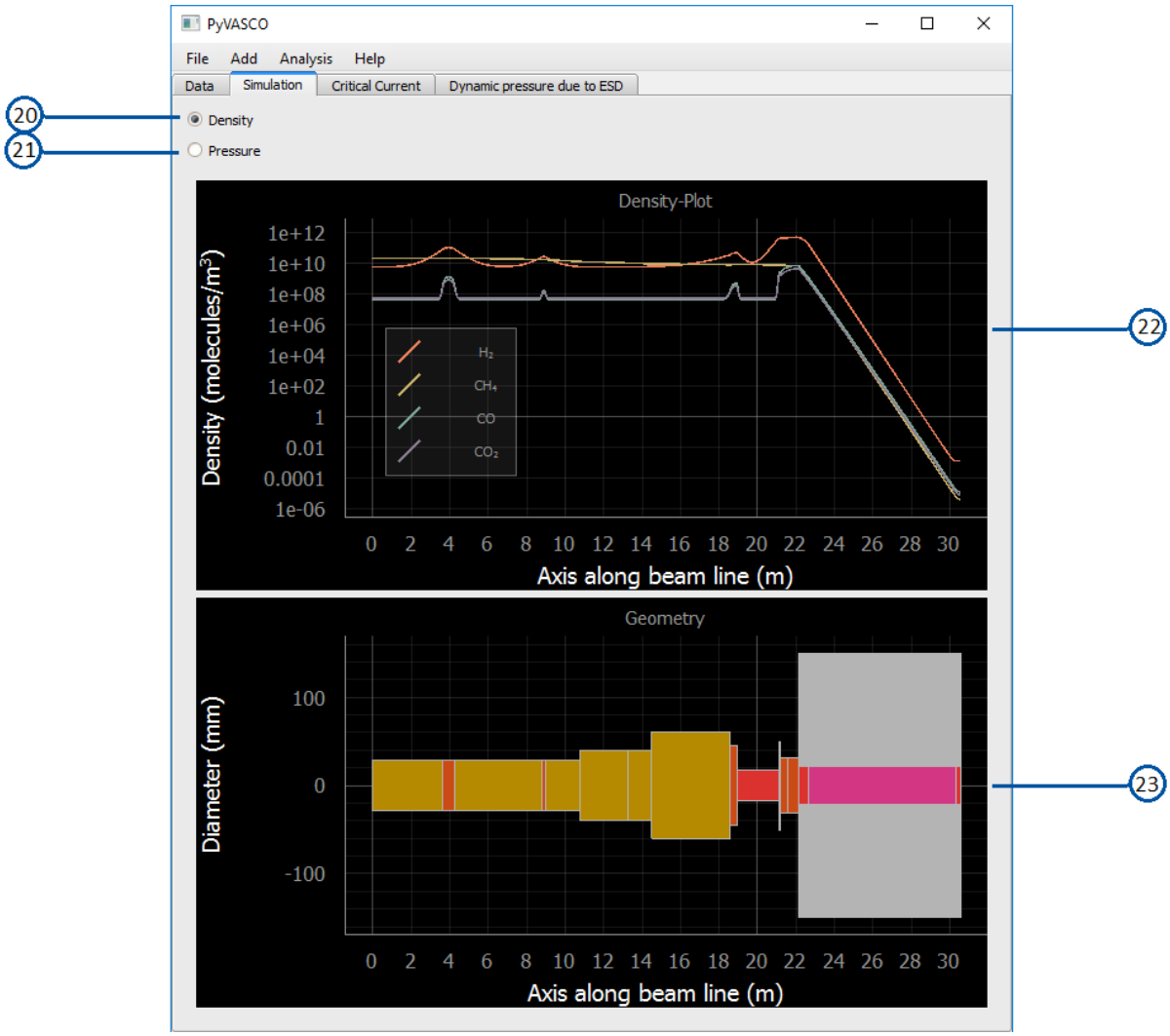

Figure 4.14: *Simulation* tab of PyVASCO.

18 **Compute critical current** :
If selected, the critical current for the selected model will be computed and plotted in the tab Critical Current. PyVASCO looks for a divergence in the gas density as a function of the beam current from the indicated initial current and increases the test beam current as indicated by precision for the indicated number of steps.

19 **Start simulation**:
Pressing this button will launch the simulation with the setup specified above. The results of the simulation are shown in the tabs Simulation and Critical Current (if option 18 has been selected).

Simulation

The numbers in Fig. 4.14 represent:

(20) **Density**:

If selected, the plot in (22) will show the density profile of the different gas species selected in the tab Data in molecules/m^3.

(21) **Pressure**:

If selected, the plot in (22) will show the pressure profile of the different gas species selected in the tab Data in mbar and the total pressure computed using the sensitivity factors in (14) (if the option (15) is selected).

(22) **Density/Pressure plot**:
This plot shows the simulated density or pressure profile, if the option (20) or (21) is selected, respectively.

(23) **Geometry plot**:
This plot shows a block diagram of the simulated system. Different colors correspond to different materials.

Critical Current

Ion stimulated desorption in the presence of a high intensity proton beam can lead to the so-called ion induced pressure instability [8]. Above a certain beam current, the so called the critical current, I_C, the pressure in the system diverges. PyVASCO allows to simulate the evolution of the pressure for different beam currents, and to estimate the value of the critical current when a divergence in the pressure of the system is found.

The numbers in Fig. 4.15 represent:

(24) **Critical current value**:
Computed value of critical current for the simulated system.

(25) **Total density profile plot**:
This plot shows the total molecular density profile for the different computed beam currents.

(26) **Dynamic current plot** :
This plot shows the maximum density of the different gas species as a function of the beam current.

In order to compute the critical current for the simulated system for an explanation on the critical current), the dynamic vacuum model presented

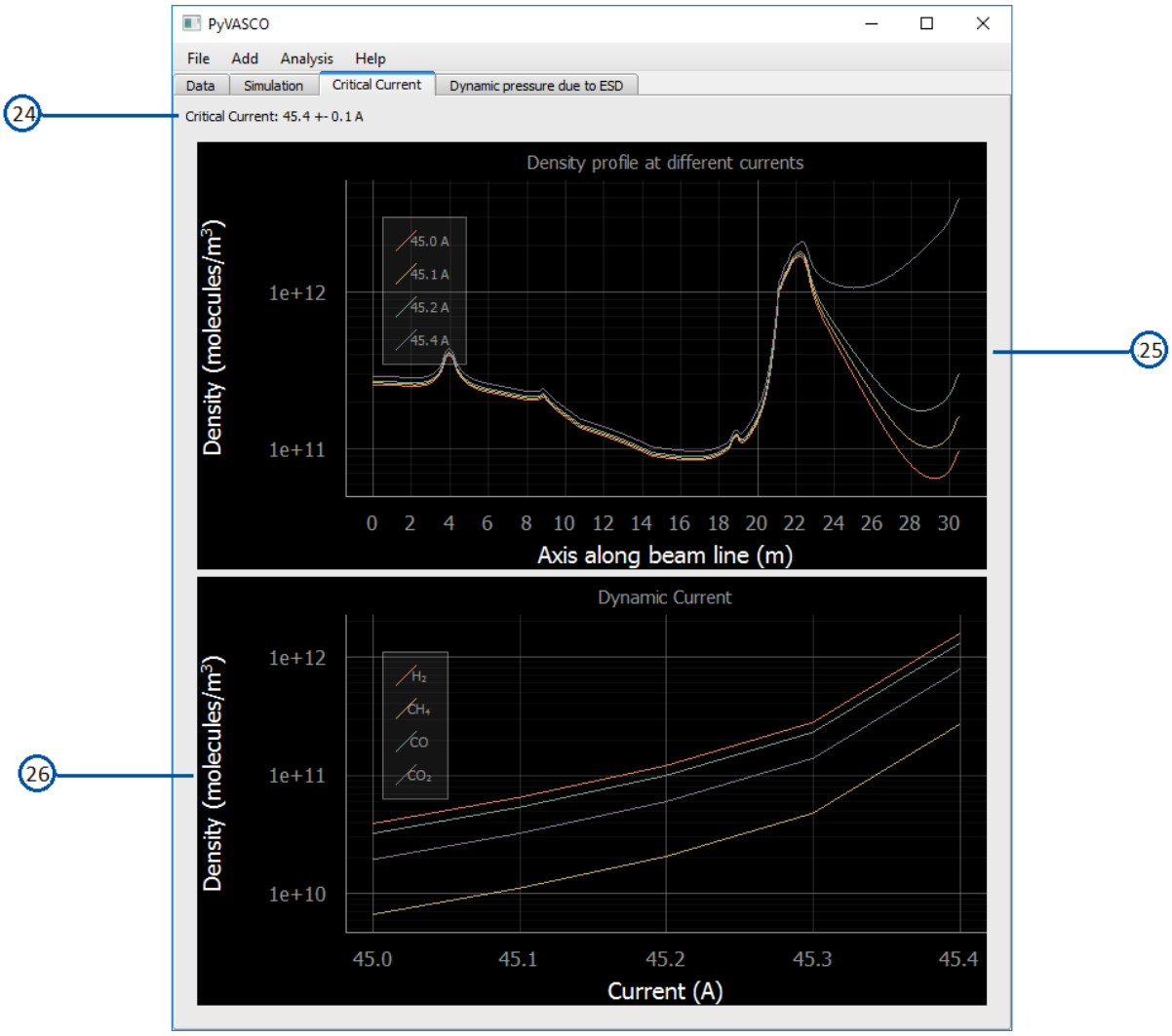

Figure 4.15: *Critical Current* tab of PyVASCO.

in [1, 2] and summarized in Section 2.2 is solved for different tested beam currents. The first value of the beam current used for the computation is the value set in the box 'Initial current [A]' in (18). The subsequent used values of current are computed by repeatedly increasing the initial value by the indicated precision until the number of steps entered by the user is reached. If a divergence in the density has been found for a given beam current, this value ± the precision are set as the critical current. If the divergence is found in the first step of the computation (the current set in 'Initial current [A]' in (18)), the shown critical current value will be ≤ Initial current. On the opposite, if a divergence in the density is not found after the indicated number of steps, the value of critical current shown in this tag will be ≥ than the last tried beam current.

Dynamic pressure due to ESD

The tab *Dynamic pressure due to ESD* of PyVASCO allows to perform two different simulations varying the ESD of the materials used in the simulation. The different setups of this tab are shown in Figs. 4.16 and 4.17, and the numbers in these figures indicate:

27 **Open Directory**:
Select the input file (in the native PyVASCO format).

28 **ESD from accumulated electron dose**:
This option uses the ESD yields of the different UHV gas species for the selected accumulated electron dose while solving the dynamic vacuum model for the input file selected in 27. If option 28 is selected, the slider 29 will appear in the box 'Simulation parameters'.

29 **Estimated electron dose**:
This slider allows the user to set the accumulated electron dose received homogeneously along the simulated geometry.

30 **Start simulation**:
Launches the simulation.

31 **Scrubbing Plot**:
This plot shows the density profile of the different selected gas species for the accumulated electron dose selected in 29 if option 28 is selected. On the contrary, if option 32 is selected, the plot will show the total density profile for the different accumulated electron doses set in 33.

32 **Scan ESD for different accumulated electron doses**:
This option solves the dynamic vacuum model presented in Section 2.2 for a range of accumulated electron doses specified in 33.

33 **Electron dose values**:
These 3 boxes allow the user to introduce the range of accumulated electron doses of interest for the simulation.

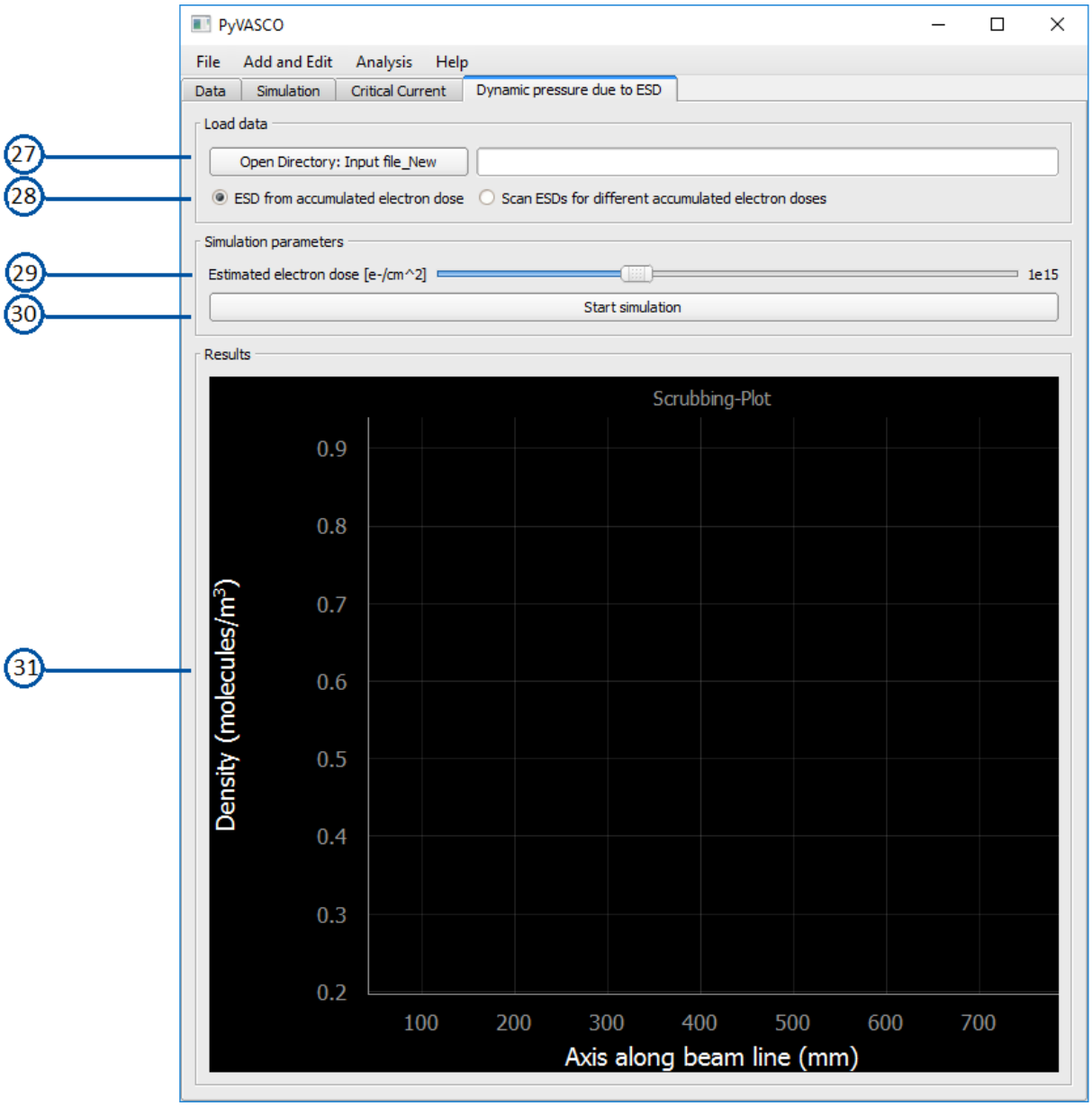

Figure 4.16: *Dynamic pressure due to ESD* tab of PyVASCO, layout for the simulation after receiving a fix accumulated electron doses.

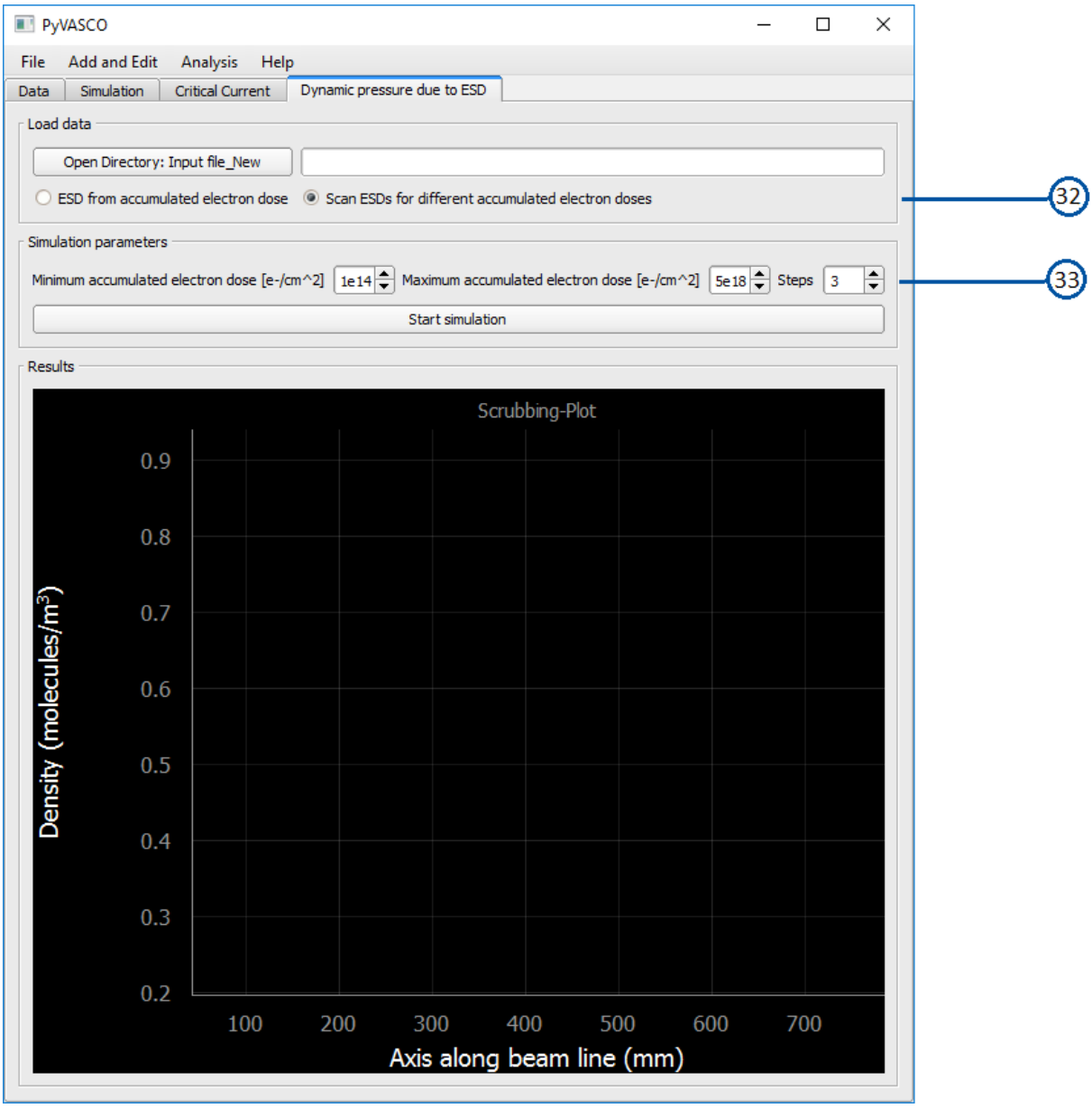

Figure 4.17: *Dynamic pressure due to ESD* tab of PyVASCO, layout for the simulation of the conditioning effect.

IMPORTANT! : Please note that this simulation will be properly performed if an ESD curve for each of the materials used in the simulation has already been defined. If this is not the case, please link the concerned materials with an ESD curve pressing on the menu Add → ESD curve.

Chapter 5

Extracting results with PyVASCO

By right-clicking on any plot of PyVASCO, the menu in Fig. 5.1 will appear. This menu allows the user to directly modify some properties of the presented plot and to export it in several formats. In this chapter, a fast overview of the relevant options available for the extraction of data from PyVASCO will be given.

5.1 Management and plot options

The plot menu contains the following options:

- **View All** : This option centers the plotted data and adjusts the axis scale to optimize the occupied space in the plot. Selecting this option is equivalent to press the A which appears in the left-down corner in every plot.

- **XAxis and YAxis**: Selecting these option will open the menus in Figs. 5.2a and 5.2b, respectively. The first and second cell under the option *Manual* allow to set the minimum and maximum values in the x and y axis, respectively.

- **Mouse Mode**: In 3 button, the left mouse button pans the view and the right button scales. In 1 button mode, the left button draws a rectangle which updates the visible region (this mode is more suitable for single-button mice).

- **Plot options**: Several options are available. The most relevant are *Log X* and *Log Y* (under the Plot options > Transforms), which allows to change the scales in both the X and Y axis from linear to log and vice versa, and*Grid*, which allows to show, hide and modulate the intensity of the plot grid in both x and y axes.

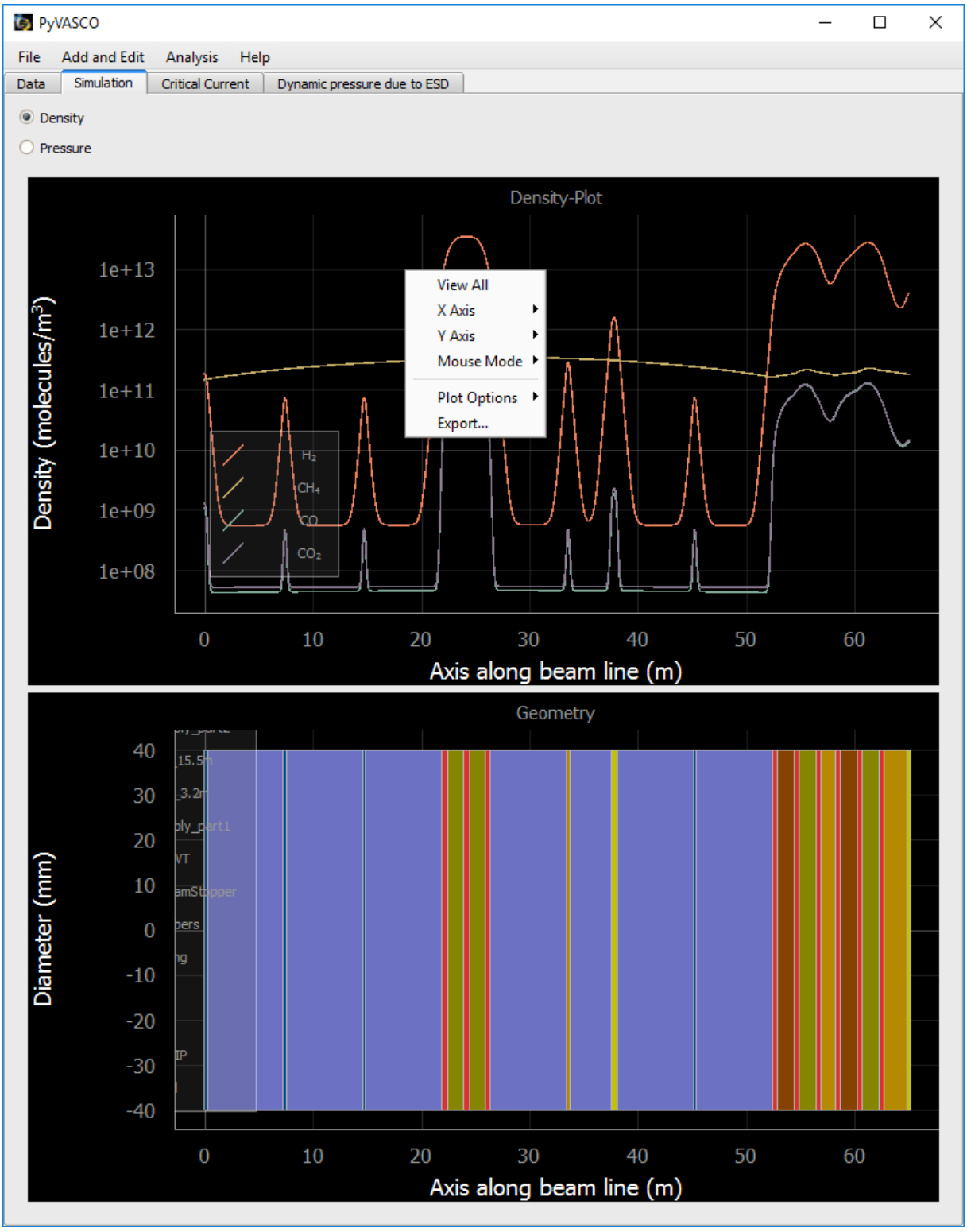

Figure 5.1: Plot menu (by right-clicking anywhere in the plot).

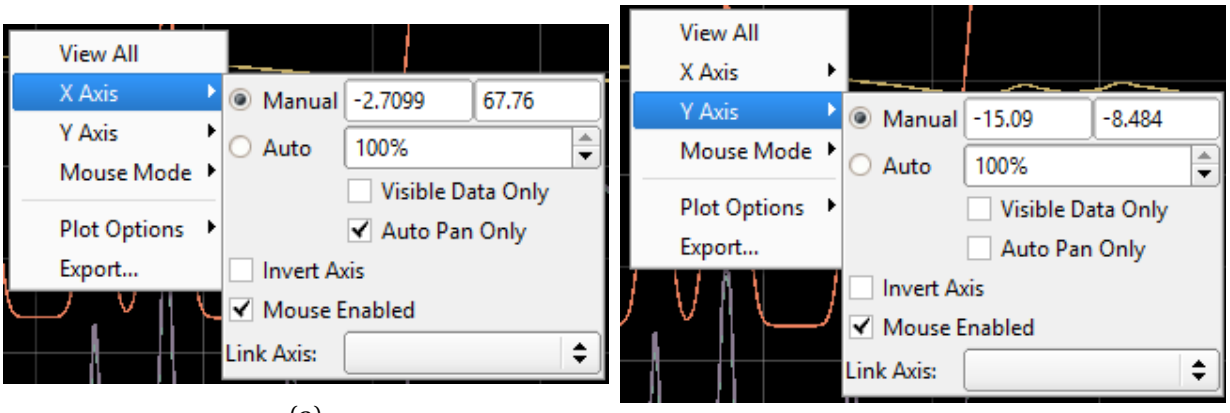

Figure 5.2: Options XAxis and YAxis in the plot menu.

5.2 Exporting plots in different formats

Pressing the option *Export* in the plot menu of Fig. 5.1 will launch the menu in Fig. 5.3.

The *Export* menu allows the user to save the plots in PyVASCO in different formats, listed in the frame *Export format* in Fig. 5.3. It is interesting to mention the following formats:

- **Image File**: The user can export any plot in PyVASCO as an image file. This options allows for the manual modification of the size of the resulting file and the background color.
- **Scalable Vector Graphics**: The advantage of this option is that the resulting plot can be opened and edited directly in programs like *InkScape*.
- **Matplotlib Window**: Matplotlib is one of the most used plotting libraries in Python. This option allows the user to further customize the plot changing, for instance, the width, shape and markers of the lines, the labels in both axis, etc.
- **CSV**: The data from all PyVASCO plots can be exported to CSV format with this option.

IMPORTANT!: If the X or Y scale in the plot to export to CSV data is in logarithmic scale, only the logarithm of the real value will be exported. The real values can be recovered by doing $10^{\text{exported values}}$.

Alternatively, the data of the pressure profile in the Tab Simulation of PyVASCO is automatically saved in CSV format in the directory selected in (5) of Tab Data if option (8) in the same tab is selected.

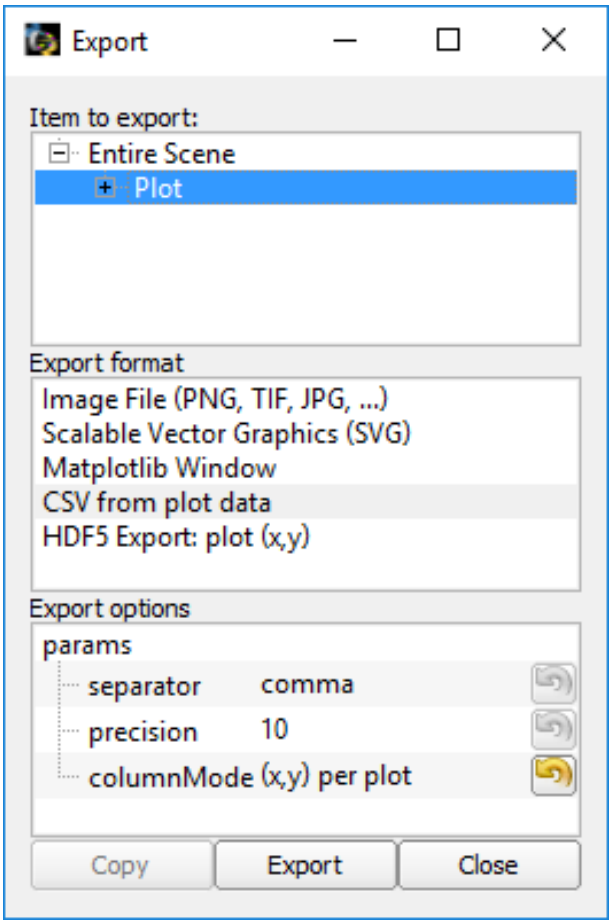

Figure 5.3: Export menu.

Chapter 6

Benchmark with VASCO and Molflow+

We have benchmarked PyVASCO with two other programs: VASCO and Molflow+ [9] for the simple case of a single segment of length 100 cm and a radius of 1 cm, with lumped pumps of 5 l/s connected in the extremes. Fig. 6.1 shows the 3D model built in Molflow+ for this simulation and Figs. 6.2 and 6.3 show a comparison between the results of PyVASCO and Molflow+ and PyVASCO and VASCO, respectively, for this simple model.

> IMPORTANT! : Please note that more complex tests for comparing PyVASCO with other simulation tools have not been done due to lack of time.

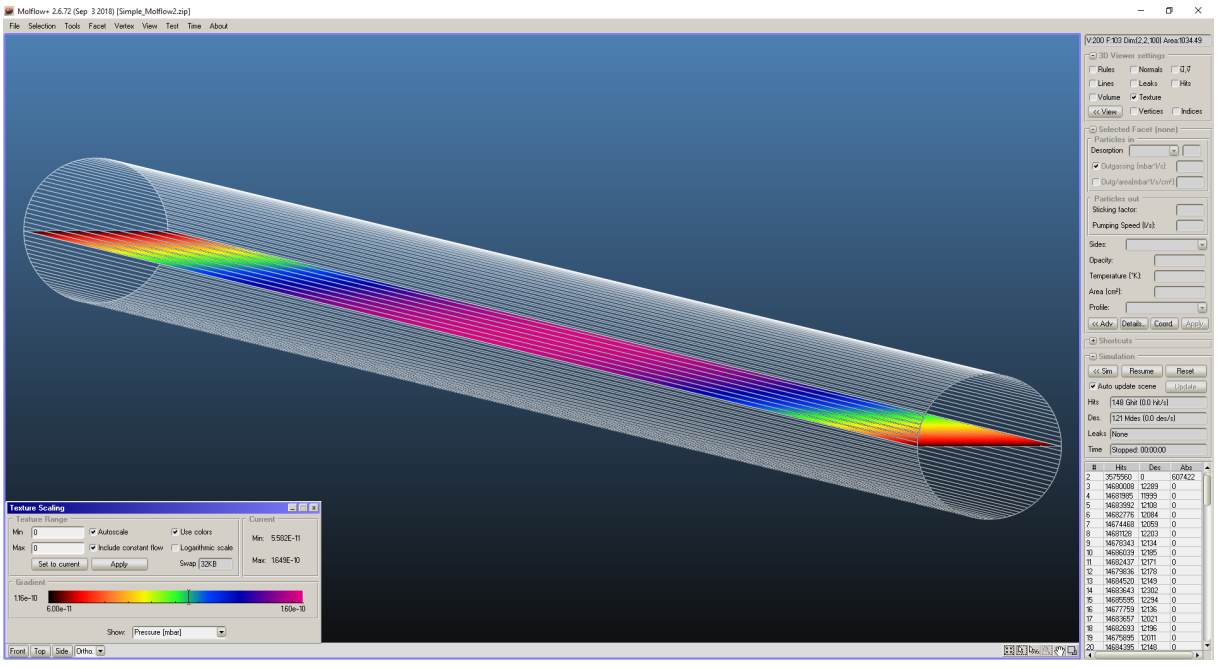

Figure 6.1: Molflow+ model for benchmark with PyVASCO.

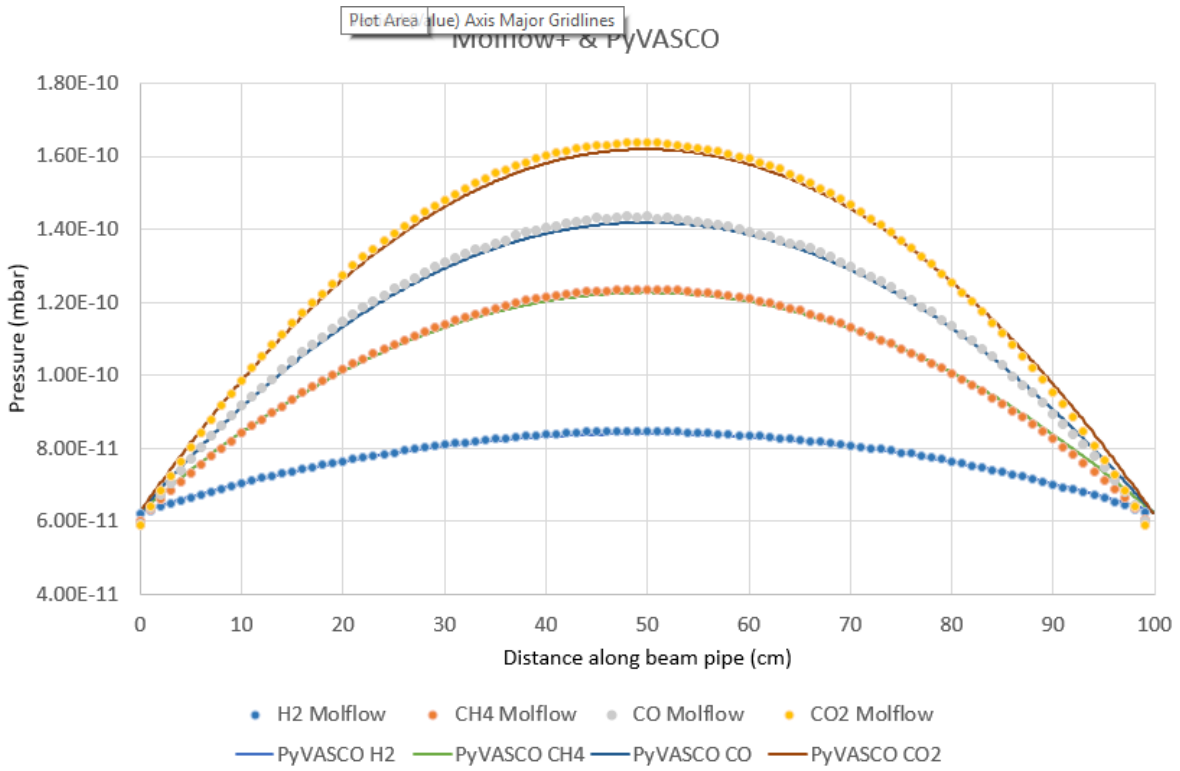

Figure 6.2: Comparison between the results obtained with Molflow+ and PyVASCO.

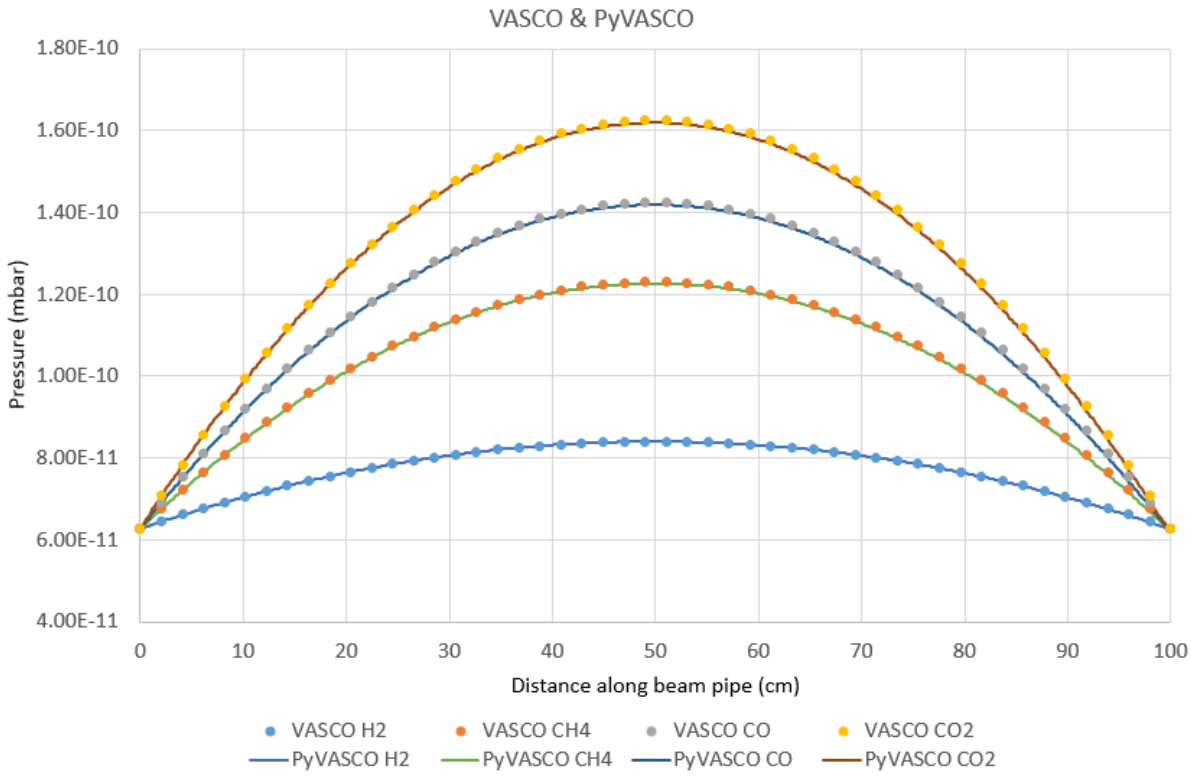

Figure 6.3: Comparison between the results obtained with VASCO and PyVASCO.

Bibliography

[1] A. Rossi, "VASCO (VAcuum Stability COde): multi-gas code to calculate gas density profile in a UHV system," Tech. Rep. LHC-Project-Note-341. CERN-LHC-Project-Note-341, CERN, Geneva, Mar 2004.

[2] I. Aichinger, G. Larcher, and R. Kersevan, "Vacuum Simulations in High Energy Accelerators and Distribution Properties of Continuous and Discrete Particle Motions," 2017. Presented 2017.

[3] O. Bruning, F. Caspers, I. R. Collins, O. Grobner, B. Henrist, N. Hilleret, J. . Laurent, M. Morvillo, M. Pivi, F. Ruggiero, and X. Zhang, "Electron cloud and beam scrubbing in the lhc," in *Proceedings of the 1999 Particle Accelerator Conference (Cat. No.99CH36366)*, vol. 4, pp. 2629–2631 vol.4, March 1999.

[4] C. Day, "Basics and applications of cryopumps," 01 2006.

[5] P. Ribes Metidieri, C. Yin Vallgren, G. Skripka, G. Iadarola, and G. Bregliozzi, "TDIS pressure profile simulations after LS2," tech. rep., CERN, Geneva, 2018.

[6] A. G. Mathewson and S. Zhang, "Beam-gas ionisation cross sections at 7.0 TeV," Tech. Rep. LHC-VAC/AGM. Vacuum-Technical-Note-96-01, CERN, Geneva, Jan 1996.

[7] Brüning, Oliver Sim and Collier, Paul and Lebrun, P and Myers, Stephen and Ostojic, Ranko and Poole, John and Proudlock, Paul, *LHC Design Report*. CERN Yellow Reports: Monographs, Geneva: CERN, 2004.

[8] E. Fischer and K. Zankel, "The stability of the residual gas density in the ISR in presence of high density proton beams," Tech. Rep. CERN-ISR-VA-73-52. ISR-VA-73-52, CERN, Geneva, Nov 1973.

[9] M. Ady and R. Kersevan, "Molflow+ user guide," tech. rep., CERN, Geneva, June 2014.