AZURE2 User Manual

Version 1.0

E. Uberseder[1,2] and R. J. deBoer[1]

March 2, 2015

[1] Joint Institute for Nuclear Astrophysics, University of Notre Dame, Notre Dame, Indiana 46556, USA
[2] Present Address: Cyclotron Institute, Texas A&M University, College Station, TX 77843 USA

Contents

List of Figures

1 Attribution

The authors of the code AZURE2 are indebted to the developers of the original AZURE code. In particular to R. E. Azuma who is responsible for the original concept of the AZURE collaboration and who headed the effort to develop the original code until its completion. While this code is written in a different language than the original, many of the algorithms developed for increased computational speed remain.

The development of AZURE2 has been funded mainly by the Joint Institute of Nuclear Astrophysics. In order to acknowledge such strong support and the exceptional work of those who developed the original code, the authors of AZURE2 request that all published works that incorporate results generated by AZURE2, or that incorporate portions of code from AZURE2, be acknowledged by citing the following reference [1]:
AZURE: An R-matrix code for nuclear astrophysics
R.E.Azuma, E.Uberseder, E.C.Simpson, C.R.Brune, H.Costantini, R.J.deBoer, J.Görres, M.Heil, P.J.Leblanc, C.Ugalde and M.Wiescher, Physical Review C **81**, 045805 (2010)

2 Forward

While many R-matrix codes have been developed over the years, most have been designed for specific reactions and have therefore been limited in their scope. It has been the goal of the AZURE project to create a fully multiple channel and multiple level R-matrix code capable of analyzing a wide variety of experimental data. The `FORTRAN` version of AZURE, released in 2010 went a long way toward reaching this goal. The code was able to analyze, simultaneously, one particle entrance channel with multiple particle exit channels. The code also incorporated the external capture model in order to account for direct capture γ-ray reactions, often significant contributions to the cross section in reactions important for nuclear astrophysics.

In pushing the capabilities of the `FORTRAN` version of the code, several limitations were realized. Because of the coding structure used, only a limited amount of data and levels could be analyzed simultaneously. The amount of data is even more severely limited if target effect corrections are necessary. Another limitation was the single particle entrance channel. With the ever increasing amount of experimental nuclear data, a multiple entrance channel analysis is often possible. Therefore, a code that makes this kind of analysis readily available seems especially useful for the community.

With the above motivations (as well as others), a new version of AZURE has been developed: AZURE2. The code has been re-written in `C++` in order to utilize the language's object oriented features and take advantage of modern third party mathematical libraries. It should be emphasized that the new version of the code is not simply a translation of the previous `FORTRAN` version into `C++`. While many of the same ideas for improved computational efficiency, a corner stone of the `FORTRAN` version, are still utilized, the structure of the code has been heavily altered. Because of the packages' widespread use and reliability, MINUIT2 (the `C++` version of MINUIT) [4] is utilized as the minimization routine and the GNU Scientific Libraries [6] libraries implement much of the mathematical operations.

The code now allows the user to setup the input for the *R*-matrix calculation, control the execution of the code, and plot the results, all from a single graphical interface. Numerous other changes, improvements, and additions to the code have also been implemented, which will hopefully result in a better user experience with more comprehensive analysis options.

As always, an important aspect of our philosophy is that AZURE2 should remain "open source." While the developers have worked hard to make the code as comprehensive as possible, we recognize that users may need to alter it for their particular needs. This is readily encouraged and the source code has been documented. It is further hoped that through wide spread use of the code and by comparison with other existing *R*-matrix codes, a critical review of the implementation of the underlying physics in AZURE2 will occur. In this way, errors or insufficient mathematical approximations may be more readily realized.

While the developers of AZURE2 have worked very hard to create a fully-functional *R*-matrix package, we can give no warranty concerning the operation, quality, or functionality of any files included in the AZURE2 distribution. The developers also disclaim all responsibility for any damages or losses arising from the usage of any files included with the AZURE2 distribution.

3 Introduction

AZURE2 is a code that is used principally to calculate differential and angle integrated cross sections for resonant nuclear reactions relevant to nuclear and nuclear-astrophysical applications. The *R*-matrix portions of the code are based on the formulation of A. M. Lane and R. G. Thomas, Reviews of Modern Physics **30**, 257 (1958) (here after referred to as LT) [5]. Other portions of the code have been drawn extensively from different sources, and are cited as they are described below. For brevity, many of the mathematical and structural concepts that were presented in the user manual for the `FORTRAN` version of the code, which remain mathematically unchanged, are omitted from this document.

The AZURE2 code has underwent several years of testing and development before its public release. The authors have not found many physics related code bugs in several years and believe the code to be very reliable if used correctly. A systematic comparison of the original code (AZURE), AZURE2, and the DREAM code of P. Descouvemont was made by D. Mountford and is published in Nuclear Instruments and Methods in Physics Research A **767** 359363 (2014).

In general, AZURE2 is designed with two modes of operation in mind: a data driven mode and a purely calculation mode. The latter may also be used to extrapolate or interpolate the results of a data driven fit to other energy regions not covered by the data (e.g. to extrapolate to relevant astrophysical energies). The parameter optimization routine, MINUIT2, is used to perform the least squares minimization so that the theoretical R-matrix calculations can be fitted to the data.

The following manual describes the operation of AZURE2 but assumes a basic familiarity with nuclear physics and R-matrix. If the user is new to the topic of R-matrix, the authors recommend reading LT and the more recent review article Descouvemont and Baye, Reports on Progress in Physics **73**, 036301 (2010) [3].

4 New Code Features

Most of the R-matrix concepts that are necessary for the operation of AZURE2 remain unchanged from the previous version. For this reason, this section of the manual omits many of the discussions already presented in the `FORTRAN` version user manual. A user new to both AZURE2 and R-matrix is encouraged to first read the section entitled "Some R-matrix Concepts" presented in that document. Users familiar with R-matrix can likely continue without difficulty.

This section describes new features that have been implemented in AZURE2. General discussions of these topics are presented here, while the actual implementation of the code is discussed in the later sections.

4.1 Simplified Operation

One of the major goals of AZURE2 was to improve accessibility. The code is now executed through the same graphical user interface that is used for the setup of calculations and the setup information has been condensed into a single Input file. Improved memory management means that the code can operate for a wide range of different reactions without needing to be recompiled. As a result of this restructuring AZURE2 is not backwards compatible with AZURE.

4.2 Multiprocessing

AZURE2 is capable of multiprocessing using the OpenMP API, which is also implemented in MINUIT2. This allows for much more computationally intensive analysis that may involve many levels, large amounts of data, and target corrections computations. These kinds of analysis become more typical when multiple entrance and exit channel data is simultaneously analyzed.

4.3 *R*-matrix Parameterization

AZURE2 offers the option of dealing with input parameters (i.e. energies, partial widths, etc.) in a few different ways. Formal *R*-matrix parameters can be used as well as observable parameters. For physical parameters, calculations can be done using either the classical *R*-matrix or the alternate parameterization given by C.R. Brune, Physical Review C **66**, 044611 (2002), here after referred to as the Brune parameterization. In both approaches the physical parameters must be transformed to formal on-resonance parameters. The Brune parameterization uses these formal on-resonance parameters directly but in the classical approach a second transformation must be performed to shift parameters relative to a single boundary condition. The transformation back from *R*-matrix to on-resonance parameters can sometimes fail because an iterative method must be employed since the physical energies are no longer known a priori, therefore the preferred method of calculation is with the Brune parameterization, but all three methods are available in the code options.

4.4 Reich Moore Capture Formalism

One of the limitations of the standard perturbation theory method of the γ-ray channel in *R*-matrix is that the γ width must always be much smaller than the particle widths. This condition is often not met in low energy neutron capture reactions so an alternate approach has been developed to analyze these data [8]. The methods of Reich and Moore, Physical Review **111**, 929 (1958) are implemented in the code but are currently limited to only (n, γ) reactions.

4.5 Multiple Entrance Channels

One of the fundamental assumptions of *R*-matrix theory given in LT is that the reactions obey time reversal invariance. Because of this fundamental assumption, the following reaction equality is assumed to be true for compound nucleus reactions

$$a + X \rightarrow C^* \rightarrow b + Y = b + Y \rightarrow C^* \rightarrow a + X \tag{1}$$

where A, x, B, and y are all nucleons and C^* represents the excited state compound nucleus of the interaction. Because of this, the levels observed in the compound nucleus, C^*, correspond to those levels populated in both reaction sequences. Therefore, both entrance and exit *particle pairs* are treated equivalently. AZURE2 has been restructured to reflect this assumption. This has naturally lead to the support of multiple entrance channel calculations in the code.

Radiative capture reactions also obey time reversal invariance, but the current version of AZURE2 *does not* support γ-ray entrance channels.

4.6 Phase Shifts

Often elastic scattering data is published in the form of phase shifts. In order to readily include this kind of data in an R-matrix analysis, the code now allows for the input of phase shift data directly. This is one of the options available when a *segment* is defined (see Section 7.6.1). The adopted convention is that the domain is $-90^\circ < \theta_{\text{lab}} < 90^\circ$.

4.7 Angle Integrated Total Capture

New experimental techniques, often using recoil separators, have begun to make measurements of angle integrated total capture cross sections. This is equivalent to measuring the sum of all the γ-ray transitions. The implementation of this in AZURE2 is done by summing over all user defined γ-cascade transitions (see Sections 7.4 and 7.6.1). It is therefore the responsibility of the user to include all of the important γ-ray *particle pairs* in their setup if they wish to also fit total capture data.

4.8 Beta Delayed Particle Emission

Beta delayed particle emission data can be analyzed in the framework of R-matrix theory using perturbation theory [2]. This method is similar to that of incorporating γ-ray decay except that, instead of the electromagnetic force, the weak force is the perturbation on the nuclear force. The formalism follows that of F.C. Barker and E.K. Warburton, Nuclear Physics **A487**, 269-278 (1988). The total number of counts as a function of energy is given by

$$N(E) = \sum_c f_\beta P_c \sum_x \left| \sum_{\lambda\mu} B_{\lambda x} \gamma_{\mu c} A_{\lambda\mu} \right|^2 \tag{2}$$

Here f_β is the Fermi function, P_c is the penetrability, $\gamma_{\lambda c}$ is the reduced width amplitude of the level λ, $A_{\lambda\mu}$ is the A-matrix as defined in LT, and $B_{\lambda x}$ are the Beta-decay channel fit parameters. The index x is simply over the Fermi and Gammow-Teller transitions.

Note that while Barker and Warburton provide an approximate transformation of the beta-decay R-matrix fit parameters to observable branching ratios, this has not yet been implemented in the code.

4.9 Angular Distribution Co-Efficients

Angular distributions are often determined by fitting a series of Legendre polynomials to experimental data. The angular distributions are assumed to have the form

$$W(\theta) = \sum_{i=0}^{\infty} a_i P_i(cos(\theta)) \tag{3}$$

where θ is the angle of reference, P_i are the Legendre polynomials, and a_i are the co-efficients. This method is very useful because the infinite series can often be truncated at only a few terms, and $4\pi a_0$ gives the angle integrated cross section. Comparison with theory can then yield the different momentum components that make up the differential cross section. Acting in reverse of this method, AZURE2 can output the co-efficients from a calculation based on the relative strengths of the different reaction components assumed. This option is only implemented for output, AZURE2 can not take previously extracted co-efficients directly as input, instead the original angular distribution data should be used.

5 Obtaining AZURE2

The AZURE2 distribution can be found online at:

`azure.nd.edu`

All users are asked to register before downloading the distribution. Registration allows the developers to keep the users apprised of any news and updates that may occur concerning the AZURE2 code and allows for the collection of usage statistics.

6 Compiling AZURE2

Compiling AZURE2 requires several external packages to first be installed. The program is built using the CMake toolkit, available from `http://www.cmake.org`. The interface for AZURE is written using Qt. At this time AZURE2 requires Qt4, which is available for download from `https://download.qt.io/archive/qt/`. The minimization in AZURE2 is performed using Minuit2. This package is distributed within ROOT, though is not a build option enabled by default. It can also be built as a standalone package available at `http://seal.web.cern.ch/seal/snapshot/work-packages/mathlibs/minuit/`. Finally, most of the mathematical functions of AZURE2 are performed using the GNU Scientific Library. This library can be obtained from `http://www.gnu.org/software/gsl/`. AZURE2 includes a convenient plotting tool that can optionally be enabled. This tool is based upon the Qwt distribution, which can be downloaded from `http://qwt.sourceforge.net`. To enable compilation of the plotting tool, add the option -DUSE_QWT=ON to step 4 enumerated below.

After the external dependencies have been installed, AZURE2 can be build with the following steps:

1. Unpack the AZURE2 archive in your directory of choice (`tar zxvf azure2_v1.tar.gz`)
2. Change directory to the new archive, and create a build folder (`cd AZURE2 && mkdir build`)
3. Change to the build directory (`cd build`)
4. Run CMake against the source root directory (`cmake ..`) to create a Makefile
5. Build AZURE using the `make && make install` command

If the user encounters compilation problems, please contact the developers via email at azure@nd.edu.

7 Using AZURE2

The setup and operation of AZURE2 is designed to be done primarily through the graphical user interface (GUI). This interface was created using the cross platform

API `Qt` (`qt.nokia.com`). `QT` is available on many different operating systems including Windows, Mac, and Linux. Because of platform specific implementation the interface may appear different than shown here.

AZURE2's graphical interface is organized by tabs. Each of the different tabs has a different set of information that may be required for an R-matrix calculation. These tabs are organized, from left to right, in the order in which information should be entered into AZURE2. The user should *always* start by entering information into the ***Particle Pairs*** tab. This is because AZURE2 automatically makes calculations based on the information entered in this tab in order to help populate the information in the other tabs. If the user makes a change in any of the tabs, the results of those changes will automatically take effect on all other tabs. For example, if the user enters in a new *particle pair* under the ***Particle Pairs*** tab and already has levels defined on the ***Levels and Channels*** tab, the new allowed channels will automatically be calculated and displayed.

All setup information for an AZURE2 calculation is stored in a single Input File. This file can have any name as specified by the user, but the developers usually use the extension `.azr` to differentiate the setup file. This file has a rather complicated format and it is assumed that the user will use the GUI to create and edit this file. However, this file is a simple text formatted file and advanced users may wish to modify this file directly, for example, to perform some batch type calculations, though the format is not described in this manual.

In the following sections, the menus and each of the tabs of AZURE2 are described in detail.

7.1 File Menu

In the upper left corner of AZURE2 is the File menu (see Fig. 1). This menu contains the standard options for opening and saving the input file for AZURE2. The input file can be named anything the user wishes and can be saved to any directory. AZURE2 can be used to either create a new Input File or open and edit a pre-existing one.

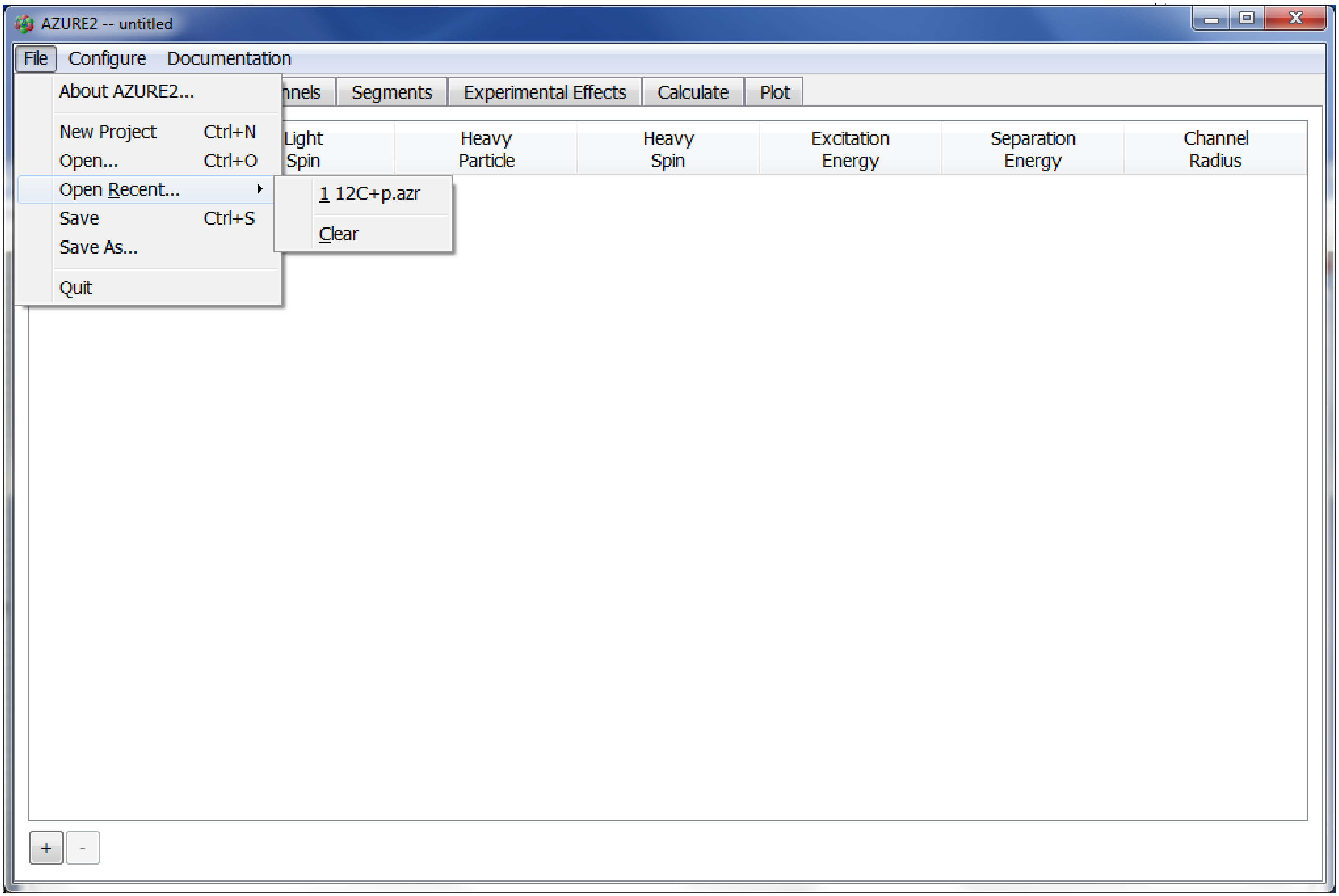


Figure 1: File menu.

While not necessary for operation, the developers have found that a standard file format works well for use of the code and for sharing calculations with others. The suggested format is that projects be contained in their own directory and that the Input file be stored there. The directory should then contain an output, checks, and data directory. The output and checks directories should be selected as detailed in Section 7.2.3 and data files (Section 7.6.1) should be placed in the data directory. All pathways should be relative to the primary directory where the Input file is stored.

7.2 Configure Menu

7.2.1 Formalism: A- or R-matrix

To the right of the File Menu on the upper left corner of AZURE2 is the Configure Menu (see Fig. 2). The first option, labeled Formalism, (as shown selected in Figure 1) allows the user to choose between R-matrix and A-matrix formalisms. The two formalisms produce equivalent results but the R-matrix is a channel matrix while the A-matrix is a level matrix. The option is purely one of computational efficiency. For the case of many channels and few levels, the A-matrix formalism is preferred. For the case of many levels and few channels, the R-matrix is preferred.

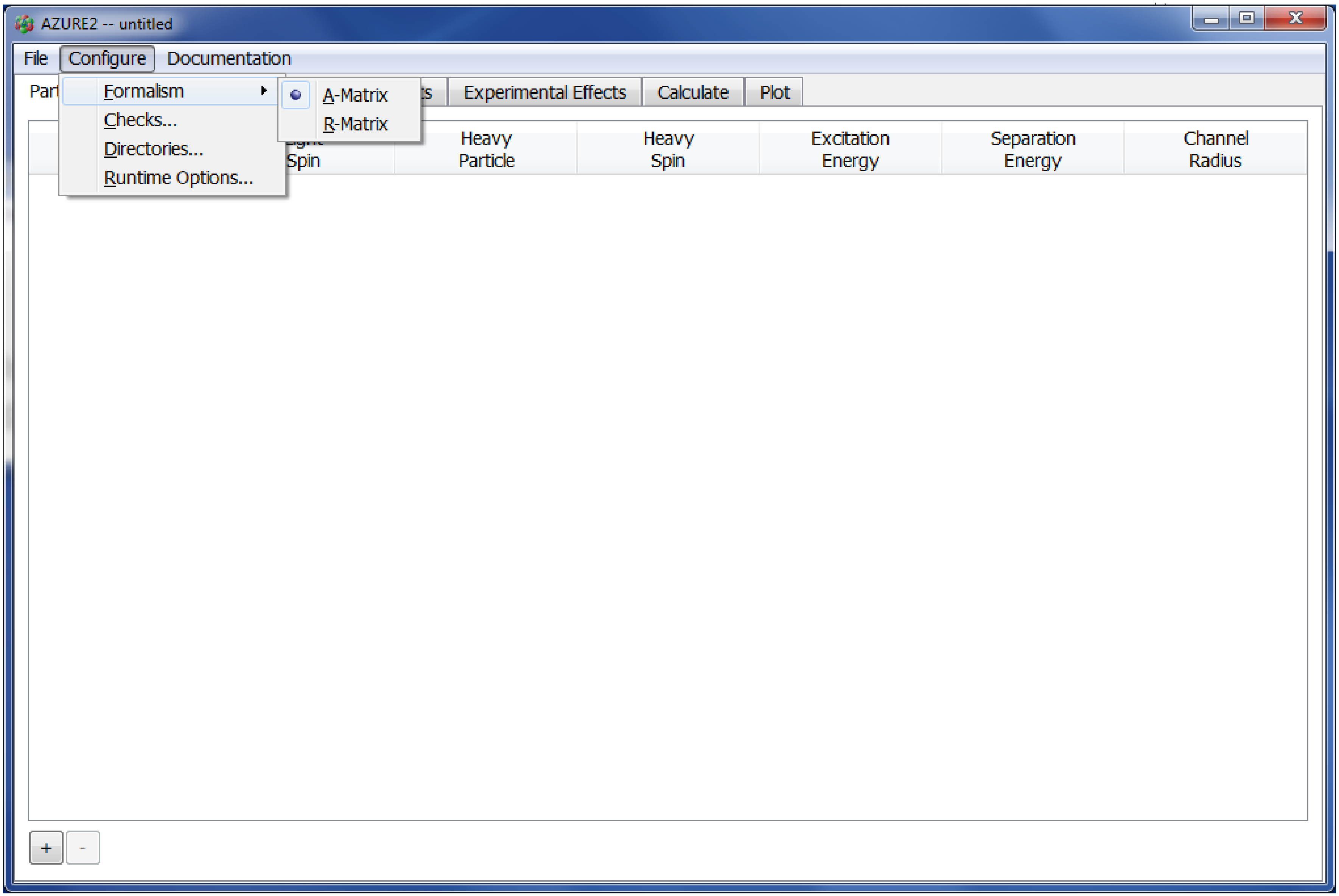
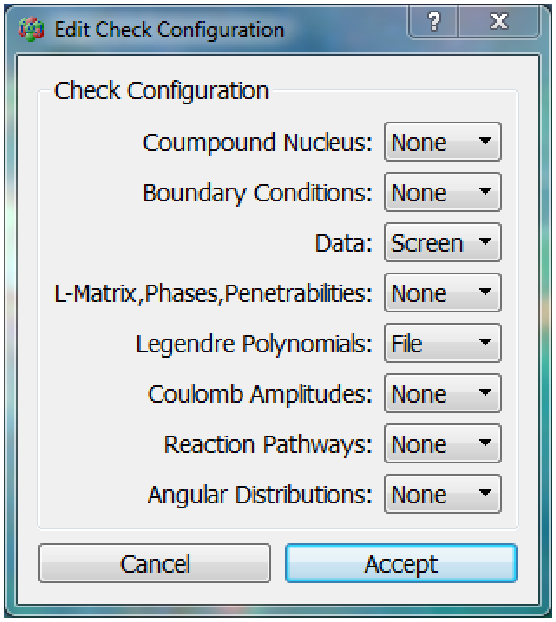


Figure 2: Configure menu.



Edit Check Configuration

Check Configuration

Coumpound Nucleus: None

Boundary Conditions: None

Data: Screen

L-Matrix,Phases,Penetrabilities: None

Legendre Polynomials: File

Coulomb Amplitudes: None

Reaction Pathways: None

Angular Distributions: None

Figure 3: Check File control under the Configure menu.

7.2.2 Check Files

AZURE2 provides Check Files to help in debugging and diagnostics for advanced users. Writing out these files can drastically increase the amount of time required for a calculation. For this reason, the Check File output can be controlled from the Checks menu that is located under the Configure menu. The toggle option for each Check File can switched between "None", "Screen", and "File" (see Figure 3). The default option is "None", providing no check file output. The option "Screen" will write a Check File's contents to the terminal and the option "File" will write to a files of prespecified names. Specification of the check file directory is controlled through the option labeled Directories, also under the Configure menu. The specific names of the Check Files are hard coded. For individual descriptions of the files' contents the user must consult the source code.

7.2.3 File Directories

The directory paths where AZURE2 writes Output Files and Check Files are specified under the Directories option (see Figure 4). These pathways may be either relative to the directory of the Input file or absolute. When the Directories

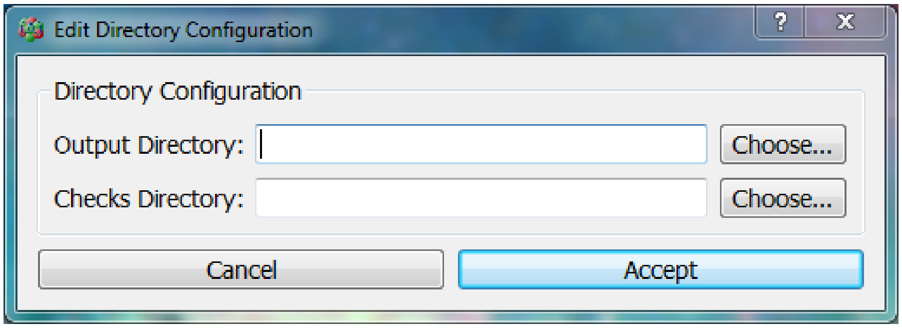


Figure 4: File directory control under the Configure menu.

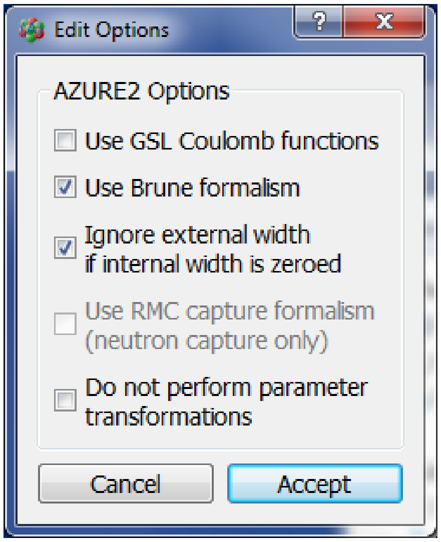
option is selected, a pop-up window appears that allows for the specification of these directories. Note that these directories should be defined prior to an AZURE2 calculation. If no file directories are given, the code will write the Output Files to the directory where the Input File is saved. The directory path must exist prior to specification because AZURE2 will not create new directories automatically.

7.2.4 Runtime Options

The final options under the Configuration menu are the Runtime options (see Figure 5). The first option, labeled "Use GSL Coulomb functions" controls the computational method by which the Coulomb functions are calculated. The GNU Scientific Library [6] (used under the GNU General Public License, `www.gnu.org/software/gsl/`) routines are used for many of the mathematical manipulations performed by the code. However, for the calculation of the Coulomb functions, GSL's methods of calculation vary and are sometimes insufficiently accurate. For this reason, an improved method [7] described by N. Michel, Computer Physics Communications **176**, 232 (2007) is used. By default AZURE2 uses the more accurate method, but computation using this method is much slower than the standard GSL method. Therefore, the GSL method may be selected if the user finds that their accuracy is sufficient.

The second option, labeled "Use Brune formalism", controls which R-matrix parameterization is used, classical or Brune (as described in Section 4.3). The developers recommend the use of the Brune parameterization and it is the default option.

Often the user wishes to ignore the external γ width of a level if no total γ width is specified. In AZURE2, this can be accomplished by zeroing the partial width of the corresponding channel if this option is specified.



AZURE2 Options

☐ Use GSL Coulomb functions

☑ Use Brune formalism

☑ Ignore external width if internal width is zeroed

☐ Use RMC capture formalism (neutron capture only)

☐ Do not perform parameter transformations

Figure 5: Run time options under the Configure menu.

The next option changes AZURE2's method of calculation to the RMC formalism. This is currently only supported for (n, γ) calculations. Unexpected errors may occur if the user selects this option and tries to perform other kinds of calculations.

The final option, labeled "Do not perform parameter transformations", allows the user to directly input R-matrix formal widths and pole energies. This may be useful if starting from an older calculation. Remember that the formal parameters are radius and boundary condition dependent. In AZURE2, the adopted boundary condition is the Shift function of the first level of each J^π. The code will still perform the final transformations from formal widths to on-resonance widths to physical widths. See Section 8 below for descriptions of which Output files contain the transformed and untransformed parameters.

7.3 Documentation Menu

Some documentation is provided within the code. This can be accessed by selecting the Documentation menu and then the Show Documentation For Current Tab option. This documentation is not meant to be comprehensive and just provides some basic information and tips. A link to the AZURE website can be accessed through the Open AZURE website... menu option.

7.4 Particle Pairs Tab

During an initial setup of a calculation using AZURE2, the user must specify the reaction *particle pairs* that will be used to define the R-matrix channels in the ***Particle Pairs*** tab (see Figure 6). Note that a (Particle, Particle) type must be specified as the first *particle pair*. After this is defined, further *particle pairs* may be in any order. To create a new *particle pair*, the user selects the + sign located in the lower left hand corner of the GUI. A pop-up window will then appear that allows the user to input the defining nuclear characteristics for the *particle pair* (see Figure 7). After the user selects "Accept", a *particle pair* will be created, displaying a summary of the nuclear information. Note that the numerical key of the *particle pair* is automatically assigned and displayed on the far left. Once a *particle pair* has been created, it can be edited by first highlighting the line and then double clicking. A *particle pair* may be deleted by highlighting and then clicking the "-" button on the bottom left of the tab.

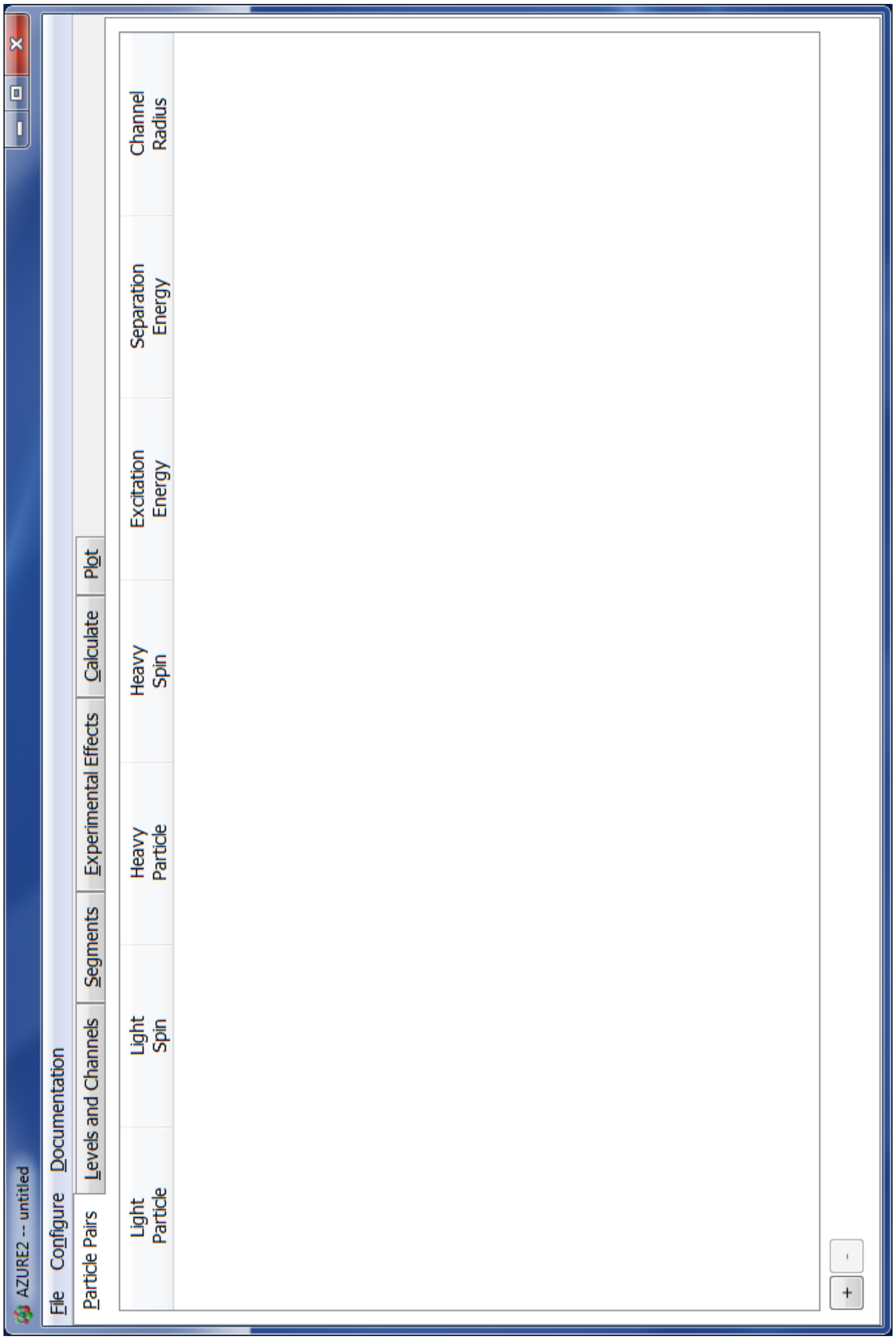


Figure 6: ***Particle Pairs*** tab.

The following describes the information that is entered into a *particle pair*. First, the Type of *particle pair* should be selected from the drop down menu located at the top of the pop-up window. The first *particle pair* must always be a (Particle, Particle) type. Note that when the (Particle, Gamma) and (Beta Decay) *particle pairs* are selected, some of the values are automatically populated and can not be edited. Spins, Proton number, Mass, and Magnetic Moment for the Light and Heavy Particle are defined followed by the Excitation Energy, Separation Energy, and Channel Radius. For the (Particle, Gamma) *particle pair*, the External Capture Multipolarities ($E1$ or $E2$) that the user wishes to consider can be selected.

- *Particle Pair Type*

  AZURE2 currently supports three types of *particle pairs*: (Particle, Particle), (Particle, Gamma), and (Beta Decay).

- *Spin (J)*

  The spin of the particle. Half integer spin should be entered as its decimal value.

- *Parity*

  The parity of the particle. Positive or negative parity should be selected from the drop-down menu.

- *Proton Number (Z)*

  The proton number of the particle.

- *Mass (M)*

  The mass number of the particle.

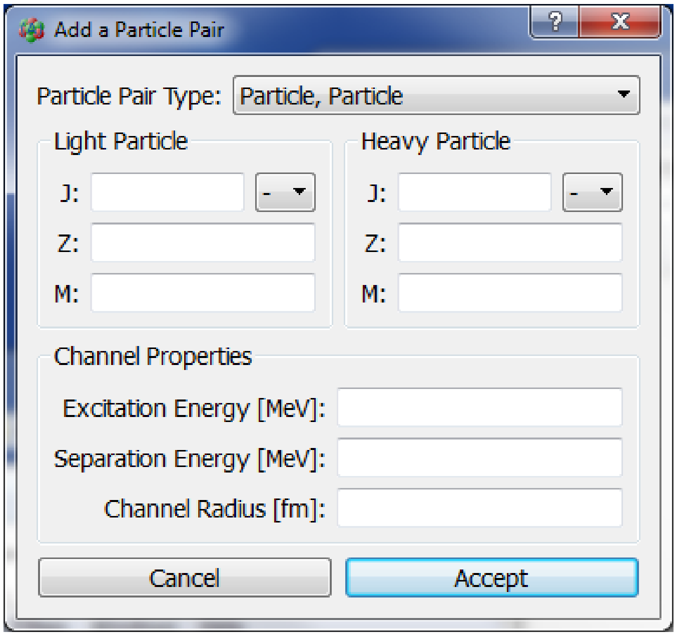
- *Excitation Energy*

  The final excitation energy of the heavy particle. This value will be nonzero for any *particle pair* representing transitions to any state other than the ground state. The energy should be entered in MeV.

- *Separation Energy*

  The amount of energy necessary to separate the compound system (in the ground state) into its constituent *particle pair*, again in MeV.

- *Channel Radius*



Particle Pair Type: Particle, Particle

Figure 7: Add *particle pair* pop-up under the ***Particle Pairs*** tab.

While the channel radius is a necessary parameter throughout an R-matrix calculation, each *particle pair* need not have the same value. The user is free to choose whatever value they desire, with a useful estimate being

$$R = R_0 \times (A_p^{1/3} + A_t^{1/3}).$$

Because the channel radius is a model parameter and does not correspond to an actual nuclear radius, it is always necessary to test the effect of the channel radius on the fitted parameters (and extrapolations). This can only be done by re-running the calculation with a new value of the radius entered into this tab. The channel radius can not currently be used as a fit parameter.

- *External Capture Multipolarities*

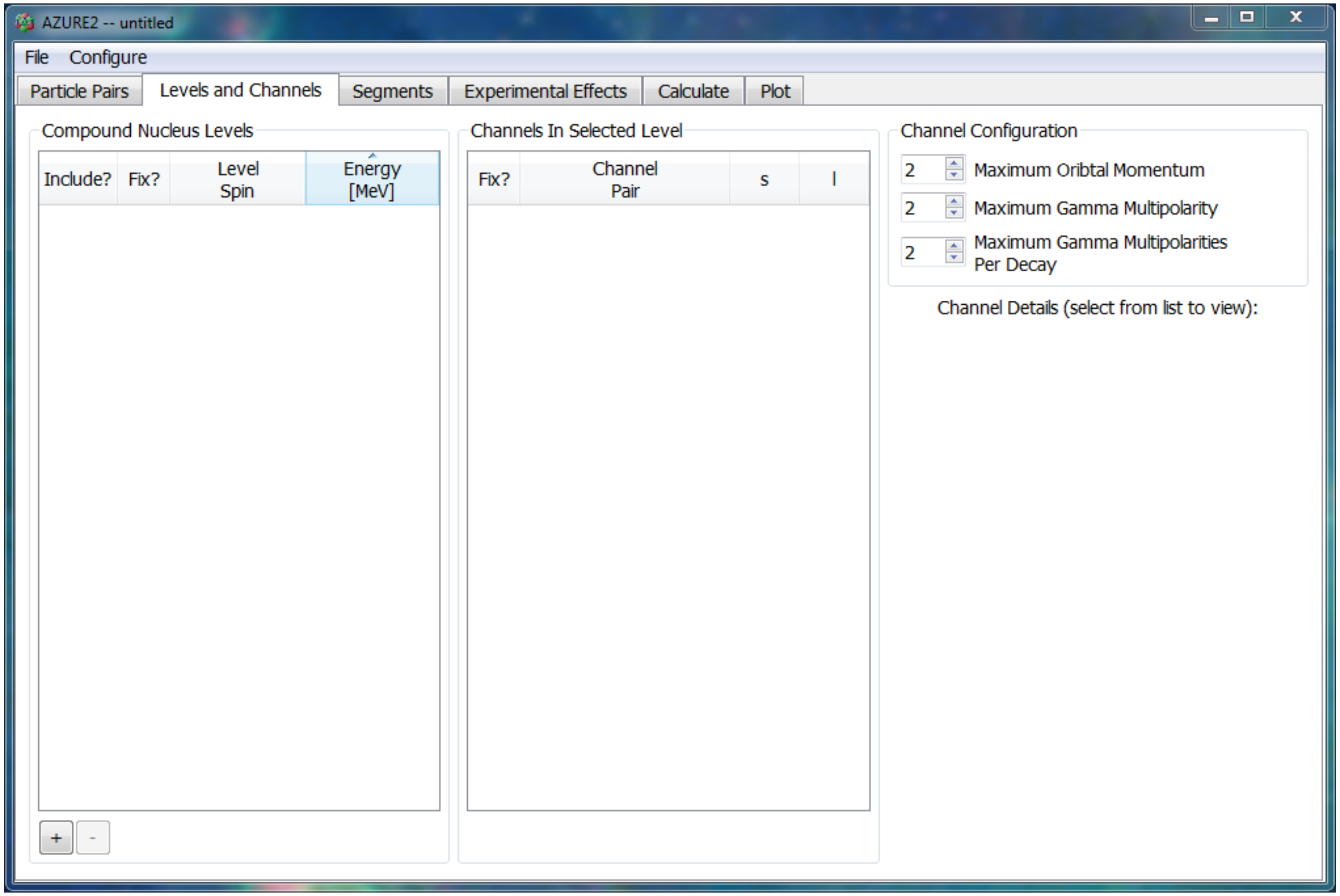
  AZURE2 currently supports $E1$ and $E2$ multipolarities for the external capture calculation. When a multipolarity is selected, the code automatically determines the allowed intrinsic and angular momentum combinations that lead to the given multipolarity based on the resonances that are specified.

With this tab completed, the user can move to the ***Levels and Channels*** tab to enter the resonance information.

7.5 Levels and Channels Tab

The ***Levels and Channels*** tab is used to enter the nuclear level information used by AZURE2. It draws from information contained in the previous ***Particle Pairs*** tab to calculate the allowed channels automatically.

This tab is divided into four frames (see Figure 8). In the left frame, labeled “Compound Nucleus Levels”, the information for the nuclear levels is added. The second frame labeled “Channels in Selected Levels”, located in the center of the tab, displays the allowed R-matrix channels. The third frame labeled “Channel Configuration”, found in the top right section of the tab, contains options to limit the number of R-matrix channels. The final frame labeled “Channel Details”, located on the bottom right, reprints the specific information used from the ***Levels and Channels*** tab for a given R-matrix channel. The width parameter (partial width, reduced width amplitude or ANC) for this channel is also entered here. The different options are described in detail below.



2 Maximum Oribtal Momentum

2 Maximum Gamma Multipolarity

2 Maximum Gamma Multipolarities Per Decay

Channel Details (select from list to view):

Figure 8: ***Levels and Channels*** tab.

Both bound and unbound levels are defined here. The inclusion of bound levels is necessary to control the parameters of these levels (γ widths, ANCs, etc.). This is especially important for (Particle, Gamma) reactions. In general it is recommended that the user define all known bound levels in the compound nucleus. Note that the energies of this bound levels should be exactly the same as the γ ray *particle pair* excitation levels defined in the ***Particle Pairs*** tab.

Note that the options specified in the "Channel Configuration" must be checked carefully by the user or errors can occur. These options exist because the orbital momentum must be truncated at some finite value. It is the user's responsibility to ensure that high enough orbital momenta are allowed for the calculations to be performed correctly. For example, if a high enough momentum is not specified, no channel may appear for a given *particle pair*. Further, even if one channel appears, the user should check to see if higher orbital angular momentum channels are possible. It is left to the judgment of the user whether these channels are important and need to be included in the calculations. In general it is recommended that all allowed channels be used, but this may slow computation speed. At least one channel must be specified for each *particle pair* for each J^π for proper functionality. Currently the highest angular momentum allowed by the code is $l = 10$, limiting the maximum J^π's that can be included.

Because the number of levels and orbital angular momenta of an R-matrix calculation must be truncated, it is up to the user to define which J^π's will be included. The code only includes those J^π's that are created in the ***Levels and Channels*** tab. For this reason the user must often define "dummy levels" to include all channels of interest. This is critical in order to include all of the desired hard sphere phase shifts that are needed for scattering and external capture calculations.

- *Add a Level*

  A *level* is created by clicking on the + button located in the bottom left corner of the tab. A pop-up window will appear and the user will be prompted for the relevant information concerning that level: "Excitation Energy", "Spin", and "Parity" (see Figure 9). The specified energy should be the excitation energy of the compound nucleus (in units of MeV). Half integer spins should be given as decimal values. The parity, either + or −, can be selected from the drop down menu. By clicking "Accept", a new level is added to the list, and the allowed channels that can populate the level are automatically calculated from the information contained in the ***Particle Pairs*** tab. These channels can be viewed by selecting the *level segment* of interest from the list under the Compound Nucleus Levels frame. The allowed channels will then appear in the "Channels In Selected Level" frame.

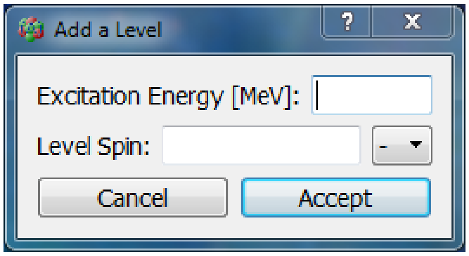


Figure 9: Add level pop-up under the ***Levels and Channels*** tab.

When one of the channels is then selected, under the "Channels In Selected Level" frame, further details about the channel will appear in the area to the right labeled "Channel Details". An entry box where the partial width parameter can be entered is found at the bottom of the "Channel Details" section. The order in which the *levels* are created is not important.

- *Edit a Level*

  Once a *level* has been added, it can be modified by selecting the *level* list under the "Compound Nucleus Levels" frame and then double clicking on it. If the user wishes to delete a *level*, simply select that *level* and click on the − sign button located just to the right of the + sign button in the lower left corner of the tab.

- *Include or Exclude a Level from Calculation*

  On the far left side of each *level* in the "Compound Nucleus Levels" frame, a check box appears under the header labeled "Include?". This controls whether a *level* will be included or excluded from the calculation. To remove a level from the calculation, simply remove the check from this box.

- *Fix a Level's Energy*

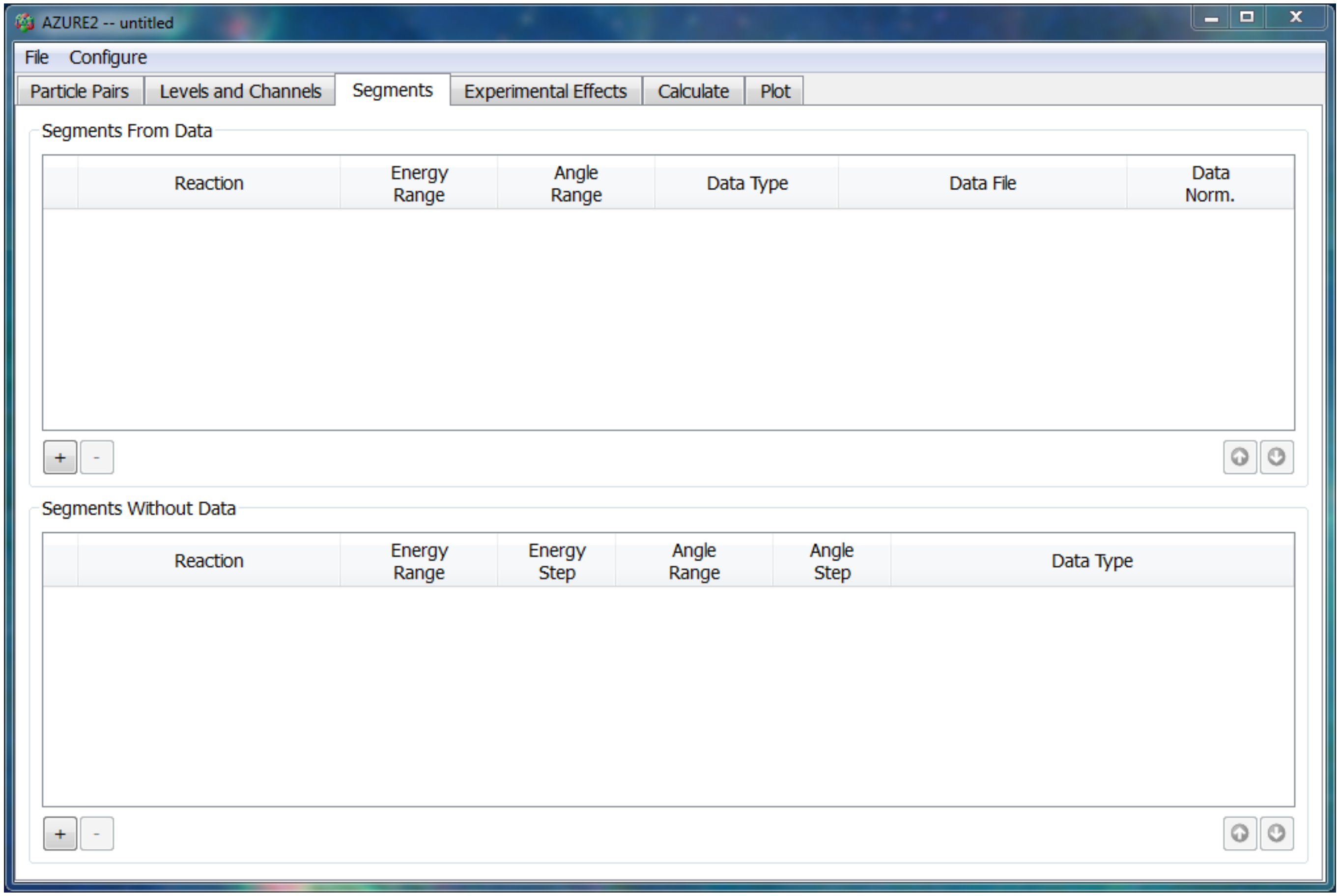
  Next to the Include column is a column labeled "Fix?". When this box is checked the energy of the resonance will not be allowed to vary as a fee parameter in the fit.

- *Fix a Width Parameter*

  In the Channels frame another column appears labeled "Fix?". When this box is checked the width parameter of the corresponding channel will be fixed. Note that if the width parameter is left zero, it is automatically held fixed.

7.6 Segments Tab

The ***Segments*** tab is used to select the energies and angles at which cross sections will be calculated by AZURE2 (see Figure 10). As mentioned earlier, AZURE2 has two different main modes of operation: calculations with data and calculations without data. The upper frame, labeled "Segments From Data", is used to select *calculation segments* to be included while operating in "Fit Segments From Data" or "Calculate Segments From Data" mode (see Section 7.8). Both of these calculation modes consider experimental data and cross section calculations are only made at the energies and angles of the experimental points. The lower frame, labeled "Segments Without Data", can be used to define specific energies and angles when operating using "Calculate Segments Without Data" mode. This mode is used when the user wishes make a pure calculation without directly considering data. In this mode the user can make calculations at any energy or angle they wish.



Segments From Data

	Reaction	Energy Range	Angle Range	Data Type	Data File	Data Norm.

Segments Without Data

	Reaction	Energy Range	Energy Step	Angle Range	Angle Step	Data Type

Figure 10: ***Segments*** tab.

7.6.1 Creating Segments With Data

- *Creating User Data Files*

  A Data File consists of four white space-delimited columns of real numbers (decimal or scientific format):

  - Col.1 - Forward kinematic laboratory frame energy (MeV)
  - Col.2 - Forward kinematic laboratory frame angle (degrees) (dummy index still required for angle integrated data)
  - Col.3 - Forward kinematic laboratory frame cross section (barns or barns/sr)
  - Col.4 - Forward kinematic laboratory frame cross section uncertainty (barns or barns/sr)

  Note that the angle must still be entered even for data that is angle integrated. The angle then simply acts as a dummy index and is not actually used for calculation.

  Data may be entered in any order, it *does not* have to be sorted in any particular way. However, for plotting purposes, it is useful to sort the data either by energy or angle if the user wishes to plot an excitation curve or angular distribution respectively.

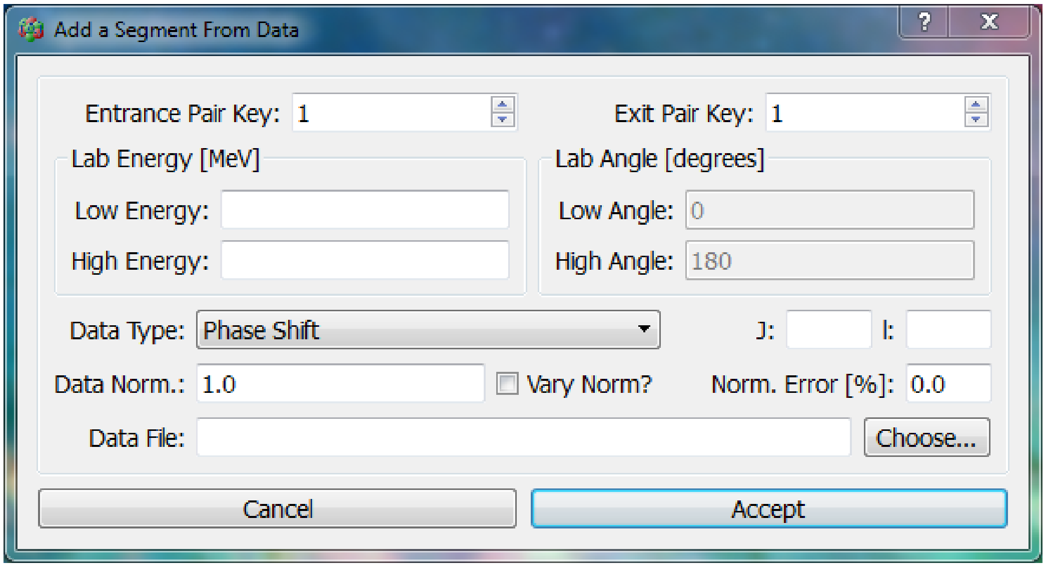
- *Creating Segments From Data*

  Similar to the way *particle pairs* are created in the ***Particle Pairs*** tab, a new *calculation segment* is created by clicking on the + sign on the lower left section of the "Segments" frame. A pop-up window appears prompting the user to specify the information that will characterize the *calculation segment* (see Figure 11).

  - *Particles Pair Selection*

    The entrance *particle pair* is chosen from the list of *particle pair* keys that were created automatically when the user created the *particle pairs*. These can be read off of the far left side of each of the *particle pair* rows on the ***Particle Pairs*** tab. The exit *particle pair* is also selected from the list of *particle pairs* in the same way as the entrance *particle pair*. The value can either be entered by hand or using the spinner on the left of each of the key boxes.

  - *Energy or Angle Range Selection*



Entrance Pair Key: 1
Exit Pair Key: 1
Low Energy:
High Energy:
Low Angle: 0
High Angle: 180
Data Type: Phase Shift
J:
l:
Data Norm.: 1.0
□ Vary Norm?
Norm. Error [%]: 0.0
Data File:

Figure 11: Add *calculation segment* with data pop-up under the ***Segments*** tab.

The boxes labeled "Low Energy and High Energy" are used to select a range of data by energy from a data file. If the user wishes to select all the data in the file, values for low and high energy should include the entire range of data in the file. Selection of angle range for differential cross sections are done in a similar manner. Note that if the "Data Type" is "Angle Integrated" or "Phase Shift", the angle range boxes are grayed out.

- *Data Type Selection*

  AZURE2 can currently accept four kinds of "Data Types": "Angle Integrated", "Differential", "Phase Shift", and "Angle Integrated Total Capture". These are selected by the drop down menu to the right of the "Data Type" label.

  The implementation of the "Angle Integrated Total Capture" "Data Type" requires further explanation. This option should be selected when data exists that represents the sum over all possible γ-ray transitions. This kind of data has become more common with the implementation of recoil mass separator experiments. Each of the significant γ-ray cascade transitions must be specified as individual particle-pairs in the ***Particle Pairs*** tab. A *calculation segment* can then be defined with the “Data Type” option “Angle Integrated Total Capture” and will automatically sum over the contributions from each of these γ-ray cascade channels.

- *Total and Orbital Angular Momentum Selection*

  For the “Phase Shift” “Data Type” only, the total (J) and angular (l) momentum must be specified. Boxes for these appear to the right of the “Data Type” selector when the “Phase Shift Data Type” is selected. Note that fractional values must be given as decimal numbers.

- *Data Normalizations and Systematic Uncertainties*

  The uncertainties of data can often be separated, at least to some approximation, into systematic and statistical contributions. If this is the case, the statistical uncertainties should be reflected by the uncertainty of each data point. Since the relative systematic uncertainty should cover the range of the entire data set, it can be entered in the box labeled “Norm. Error”. For convenience, an over all normalization factor can also be multiplied by the yield of the data. This is entered into the box labeled “Data Norm”, the default value is set to 1. It should be noted that the relative systematic uncertainty is calculated using the *initial* value of the normalization factor. If the user wishes to vary the normalization factor as a free parameter in the fit, the box labeled “Vary Norm?” should be checked. The systematic uncertainty is included in the calculation of the χ^2 using the method of G. D’Agostini, Nuclear Instruments & Methods in Physics Research A **346**, 306-311 (1994)

$$\chi^2 = \sum_i \left(\sum_j \frac{(f(x_{i,j}) - c_i n_i y_{i,j})^2}{(c_i n_i \sigma_{i,j})^2} + \frac{((c_i - n_i)/n_i)^2}{\delta^2_{c_{exp,i}}} \right) \tag{4}$$

  where c_i is the normalization fit parameter, n_i is the starting normalization, $f(x_{i,j})$ is the calculated quantity form the R-matrix (i.e. cross section, S-factor, phase shift, etc.), $y_{i,j}$ is the data point value, $\sigma_{i,j}$ is the statistical uncertainty of the data point, and $\delta_{c_{exp,i}}$ is the *percent* systematic uncertainty of the data set.

- *Data File Selection*

  The final entry box requires the path to the data file from which the data for this segment should be drawn. This path can either be entered directly or can be chosen using a file directory interface by selecting the “Choose...” option. The path can be either an absolute reference or relative to the directory of the Input file.

When all options have been entered in the calculation segment pop-up window, clicking the Accept button will create the segment. The segment appears as a line in the "Segments" frame and is given a numerical key to the far left.

- *Editing Segments.*

  If a *calculation segment* exists, it can be edited by selecting the *segment* and double clicking. The same pop-up window that was used to create the *calculation segment* will appear and the different entries can be edited freely. A *calculation segment* can be deleted by selecting the segment from the "Segments" frame and then clicking the − button on the bottom left of the frame.

It is recommended that data from different experiments be placed in different files. The *calculation segments* can be used to calculate excitation curves by selecting an energy range but only a single angle. Angular distributions can be calculated by selecting an angular range and a single energy.

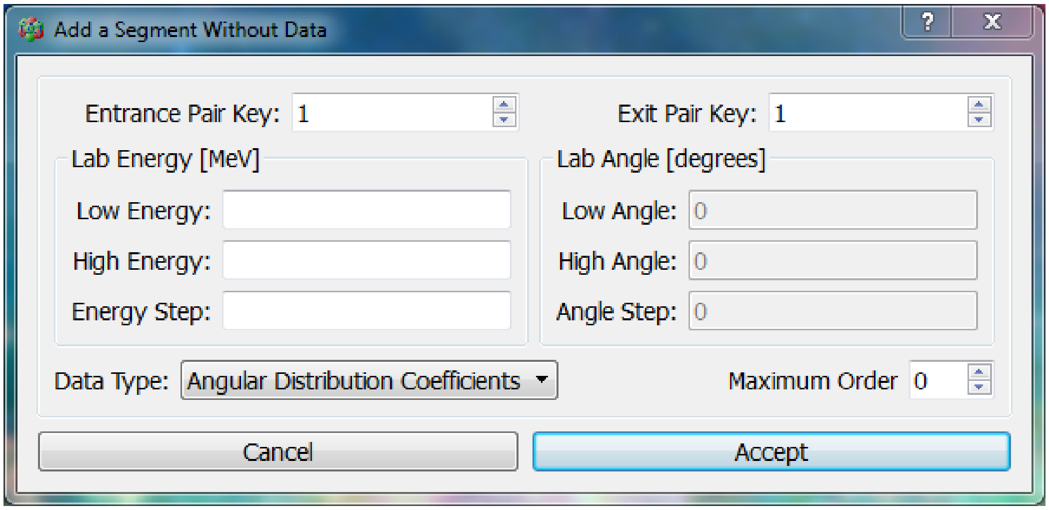
Because *calculation segments* in AZURE2 can automatically pick out specific angles or energies (or angular or energy ranges), further division of data sets by energy or angle is not necessary. Therefore if a single experimental measurement, where all measurements are made under the same conditions, has data at multiple angles and excitation curves are to be plotted, different *calculation segments* can simply be created for each angle but they may reference the same Data File. Angular distribution *calculation segments* may be created similarly but with the roles of angles and energies reversed.

Note that for careful final fitting when the systematic uncertainties are included, the user needs to carefully consider the segmentation scheme of the data so that the systematic uncertainties are not counted more than once.

7.6.2 Creating Segments Without Data

The lower half of the ***Segments*** tab entitled "Segments Without Data" is used for making calculations where no data are present or for making cross section and S-factor extrapolations after a fit has been completed. Both are done using the "Calculate Segments Without Data" mode.

Creating this kind of *calculation segment* is very similar to creating a *calculation segment* where data is considered. To create a new segment click on the + sign on the bottom left corner of the "Segments Without Data" frame. This brings up a pop-up window where the characteristics of the extrapolation can be defined (see Figure 12).



Entrance Pair Key: 1

Exit Pair Key: 1

Low Angle: 0

High Angle: 0

Angle Step: 0

Data Type: Angular Distribution Coefficients

Maximum Order 0

Figure 12: Add *calculation segment* without data pop-up under the ***Segments*** tab.

Note that errors may occur if the user specifies an energy or angular range that is not kinematically allowed. A an error may also occur if the user makes the energy too low and the penetrability become too small.

- *Particles Pair Selection*

  The procedure to select entrance and exit *particle pairs* is identical that given for *calculation segments*.

- *Energy and Angle Ranges*

  Selection of the range of energies and angles that will be included in the calculation is done by specifying a low value, high value and step size for the energy or angle. To specify that the calculation should be done at a single energy or angle, the low and high values should be set equal and the step value should be set to zero.

- *Data Type*

  Four data types are allowed for calculation: “Angle Integrated”, “Differential”, “Phase Shift”, and “Angular Distribution Co-Efficient”. Specifications for “Angle Integrated”, “Differential”, and “Phase Shift” are the same as those described for the *calculation segments* considering data. For “Angular

Distribution Co-Efficient", the order of the Legendre Polynomial must also be specified.

The order of the *segments* can be adjusted by selecting a *segment* and then using the arrow buttons on the far right of each frame to adjust the *segment's* position.

7.7 Experimental Effects Tab

The target integration and beam resolution convolution techniques implemented in the code are by no means intended to cover all cases. The developers strongly recommend that these sections of the code be evaluated by the user on a case-by-case basis. The user may need to make modifications to these sections of the code depending on their particular experimental results, as it is quite difficult to produce fully versatile target integration and convolution routines. As such, this section of the code has been designed to provide a simple ground work for target integration and beam resolution convolution effects and for custom changes. Target effect corrections can drastically increase computation time, especially for (Particle, Gamma) reactions.

A common approach to correct for beam particle energy losses in the target and the limited energy resolution of the beam is given by

$$F(E_0) = \int_{E_0-\Delta}^{E_0} \int_{E=-\infty}^{+\infty} \frac{\sigma(E')}{\epsilon(E')} g(E - E_0) dE' dE \tag{5}$$

where $\sigma(E')$ is the cross section devoid of experimental effects. The function $g(E' - E)$ is the *spreading* function that represents the energy distribution of the beam particles and the function $\epsilon(E')$ is the stopping cross section that represents the energy distribution resulting from beam particle scattering within the target. The function $F(E_0)$ is the final simulated spectrum where E_0 is the mean beam energy.

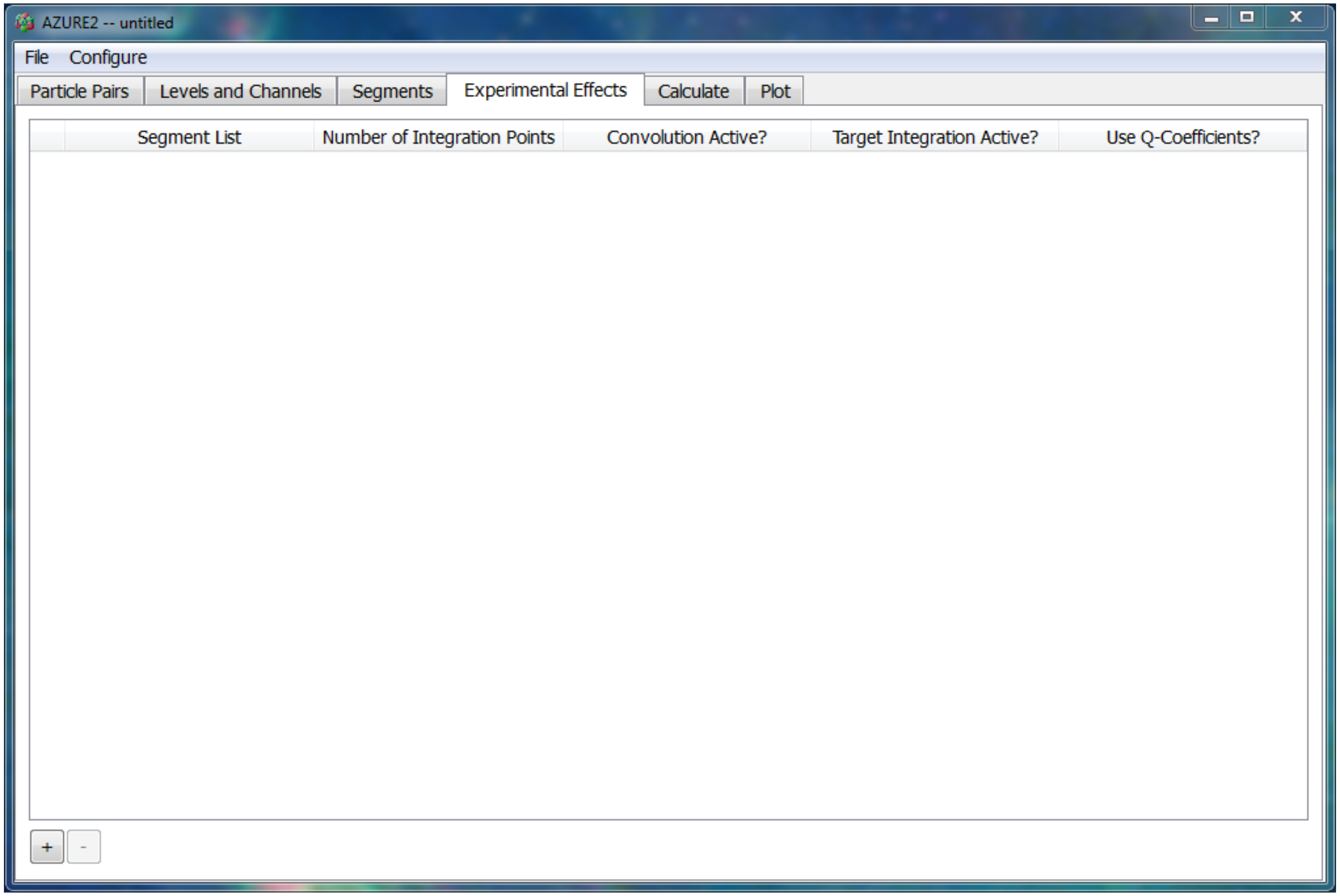
As implemented in AZURE2, the spreading function is given by the Gaussian function

$$g(E - E_0) = \frac{1}{\sqrt{2\pi}} \sigma_b \exp\left[-\frac{(E - E_0)}{2\sigma_b^2} \right] \tag{6}$$

where σ_b defines the energy width of the beam.

The ***Experimental Effects*** tab is shown in Fig. 13. To create a new *experimental effects segment* click the + sign in the lower left hand corner of the tab.

This brings up a pop-up window where the different convolution options can be specified (see Figure 14).



Segment List	Number of Integration Points	Convolution Active?	Target Integration Active?	Use Q-Coefficients?

Figure 13: ***Experimental Effects*** tab.

- *Specifying Associated Calculation Segments*

  Often a *experimental effects segment* may be applicable for several *calculation segments*. The *calculation segment* number to which the *experimental effects segment* should be applied is entered into the box labeled "Segments List". Here the *calculation segment* numbers correspond to the numbers automatically assigned to each *calculation segment* upon their creation. These are listed on the far left of each of the *calculation segment* lines on the ***Segments*** tab. The *calculation segments* can be listed either separated by a comma or, if sequential, a range can be specified using a − between the first and last segment number. For example, a user could specify segments 3,4,5,7,8, and 9 by

  - 3,4,5,7,8,9
  - 3-9

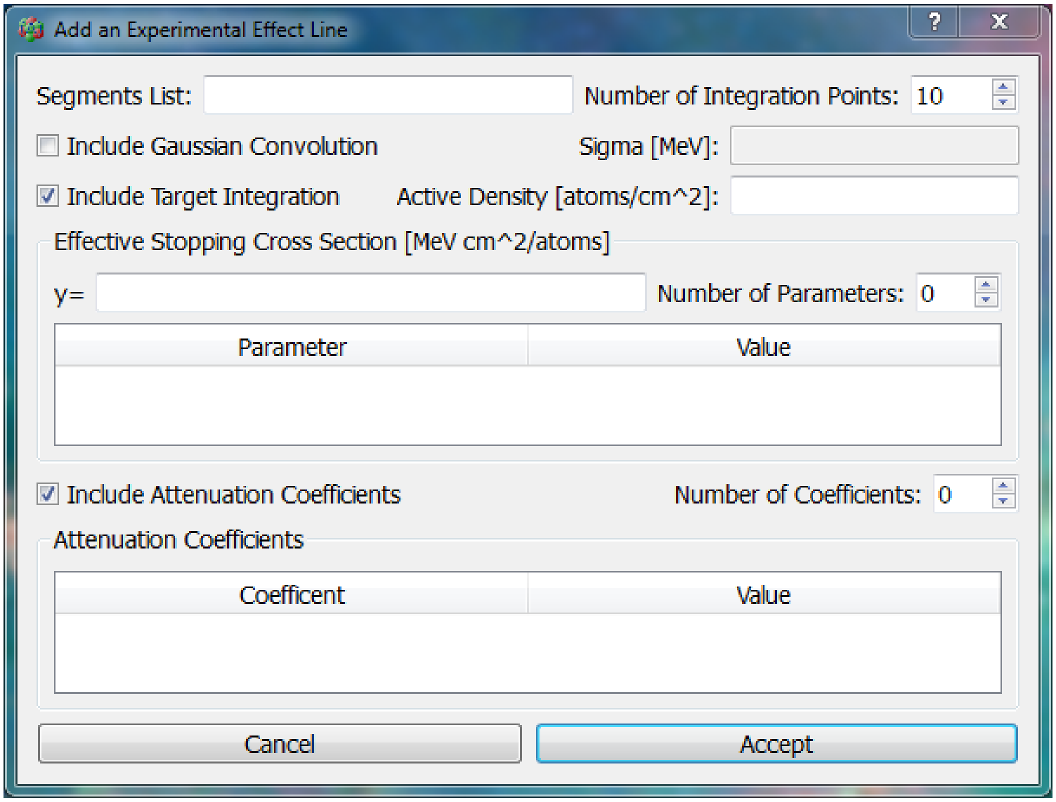
  Note, the format of 3-4,5-9 would not currently work properly.

- *Integration Points*

  When using either the "Target Integration" or the "Gaussian Convolution" options, a number of "Integration Points" must be specified in order to perform the numerical integration. The number of "Integration Points" required is determined by how rapidly the cross section is changing as a function of energy. This number can be set using the box located in the upper right section of the experimental effects pop-up window. The value can either be entered manually or can be set using the up and down arrows.

- *Energy Convolution*

  In any standard accelerator type experiment, the true cross section is never measured directly. The observed yield is unavoidably effected by the finite energy resolution of the beam and the finite thickness of the target. If the cross section changes slowly in energy, compared to the energy resolution and the target thickness, then the shape of the yield approaches the shape of the true cross section. It is often the case that the distortion of the cross section by these two effects (and perhaps others) can be simulated by convoluting the R-matrix cross section with a Gaussian function (see Equation 5). Since the beam resolution is usually roughly constant over the usual energy range of low energy nuclear physics experiments (a few MeV), the Gaussian function can be energy independent. This method can be included in an AZURE2 *experimental effects segment* by checking the box labeled "Include Gaussian Convolution". The full width at half maximum of the Gaussian convolution



Add an Experimental Effect Line

Segments List:

Number of Integration Points: 10

□ Include Gaussian Convolution

Sigma [MeV]:

☑ Include Target Integration

Active Density [atoms/cm^2]:

Effective Stopping Cross Section [MeV cm^2/atoms]

y=

Number of Parameters: 0

Parameter	Value

☑ Include Attenuation Coefficients

Number of Coefficients: 0

Attenuation Coefficients

Coefficent	Value

Figure 14: Add *experimental effect segments* pop-up under the ***Segments*** tab.

function can then be entered into the box labeled “Sigma”. While the value of “Sigma” is usually of the order keV, keeping with the standard units of AZURE2, this value is entered in MeV. The input can accept either decimal or scientific notation.

- *Target Integration*

  As a particle beam passes through a target, it experiences energy loss as the beam particles scatter (Coulomb and nuclear) off the target material. These energy losses (described by stopping cross sections) are often very well tabulated (e.g. SRIM, `www.srim.org`). In many experiments, the target energy thickness, calculated from the stopping cross sections, is designed to be “thin” compared to the width of the observed resonances, the so called the thin target yield approximation. However, the count rate must also be considered when designing the target thickness, resulting in the need for thicker targets. For this reason of target thickness compromise (and many others), targets are often made that only approximate a thin target yield. To correct for stopping cross section effects, basic target integration corrections can be added to a *experimental effects segment* by checking the box labeled “Include Target Integration”. The density of the *active* target material (i.e. the density of the nuclei from which the reactions of interest are produced in a mixed material target) is entered in the box labeled “Active Density”. The density must be in units of atoms/cm^2. The *effective* stopping cross section (i.e. the stopping cross section that includes all materials in the target) must also be entered in the section labeled “Effective Stopping Cross Section”. The stopping cross sections must be entered as a continuous function of energy so that they can be determined at an arbitrary energy when the numerical integration is performed. The functional representation of the stopping cross section can be entered in a very flexible manner. The equation is parameterized by y as the stopping cross section, x as the energy, and any number of constants a_i. Any kind of standard equation can be used. Here the functional representation is arbitrary, it must simply describe the cross section over the energy range of the data to the accuracy desired. For simplicity, a polynomial is often used.

  For example, consider a cross section that has been fit using a second order polynomial. Three constants are required to describe the function. The box labeled “Number of Parameters” should be set to 3 using the up and down arrows. In the table, three lines will appear that are labeled by “Parameter” a0, a1, and a2 in the parameter column. The value of each parameter can be entered by highlighting the value area to the right of the parameter and then double clicking. The equation for the polynomial can be entered into the box labeled “y=” and, for this case, would be entered as “a0+a1*x+a2*x^2”.

- *Attenuation Coefficients*

  Attenuation coefficients are used to compensate for the finite angle of detectors that are placed in close geometry with the target as presented in M. E. Rose, Physical Review **91**, 610 (1953) [9].

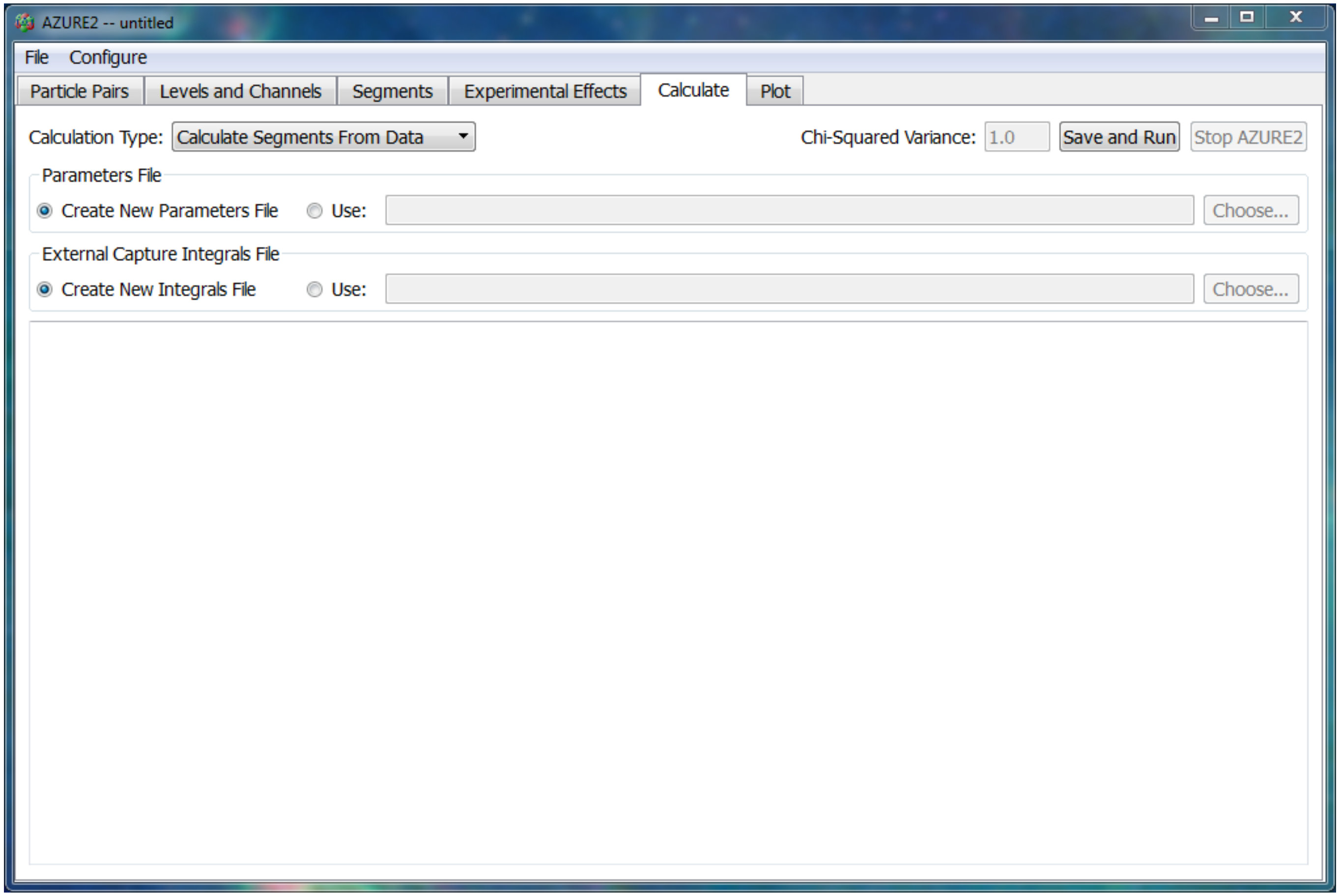
  The correction is made by adding correction terms to the angular distribution coefficients of Eq. (3)

$$W(\theta) = \sum_{i=0}^{\infty} a_i Q_i P_i(\cos(\theta)) \tag{7}$$

  where Q_i are the attenuation coefficients.

7.8 Calculate Tab

Once options have been specified in the previous tabs, the user is ready to perform an R-matrix calculation. Execution of AZURE2 is controlled from the ***Calculate*** tab (see Figure 15). The code can be run in five different modes of operation selected from the “Calculation Type” drop down menu: “Calculate With Data”, “Fit With Data”, “Extrapolate Without Data”, “Perform MINOS Error Analysis”, and “Calculate Reaction Rate”. The user may use these options in any order they see fit but they are listed in the drop down in roughly the order that they are typically employed.



Calculation Type: Calculate Segments From Data

Chi-Squared Variance: 1.0

Parameters File

● Create New Parameters File ○ Use:

External Capture Integrals File

● Create New Integrals File ○ Use:

Figure 15: ***Calculate*** tab.

To execute any of the "Calculation Types", select the type desired from the drop down menu and then click the button on the right side of the ***Calculate*** tab labeled "Save and Run". When this button is pressed, AZURE2 first saves to the Input File all of the different options that have been changed by the user, then the code performs the selected calculation. Information about the selected calculation will be displayed, in real time as the steps are preformed, in the lower half of the tab. When using the "Fit Data" mode, the code updates the printed χ^2 information every 1000 minimization iterations and also prints out to the AZURE-Out_aa=*_R=*.out files so that the progress of the fit can be monitored visually using the ***Plot*** tab. The final χ^2 output is printed to this tab and is also printed to the file chiSquared.out. Several other files are also printed when calculations are completed. When a calculation is complete, the line "Thanks for using AZURE2" is printed and the "Save and Run" button will become available again.

When performing a lengthy "Fit With Data" or "Perform MINOS Error Analysis" mode calculation, it is sometimes desirable to stop the calculation before the minimization procedure has completed. This can be done by clicking on the "Stop AZURE2" button located on the upper right part of the ***Calculate*** tab. Because of the code structure and the way AZURE2 communicates with MINUIT2, the user may have to wait a moment before the calculation actually concludes. A calculation has concluded when the "Save and Run" button becomes accessible again and the message "Thanks for using AZURE2" appears at the bottom of the ***Calculate*** tab output box. In addition to the files that are printed out every 1000 minimization iterations, the other files that are printed when the calculation concludes are also printed out. If the user desires to simply kill a calculation quickly, this can be accomplished by simply closing AZURE2.

For many of the calculation types described below, the option is given to "Create New Parameters File" or the "Use" option where a parameter file may be specified. If the "Create New Parameters File" option is selected, AZURE2 will use the parameters entered into AZURE2 as the basis for its calculation. Most often, these parameters are only initial values and a best fit set of parameters has been obtained using the "Fit Data" option. When the "Fit Data" option is complete it writes its best fit parameters into the file called param.sav. Therefore, the user may want to use this file or another parameter file that they have given another name, for a calculation. For this reason, the user has the option to choose a file that contains some set of stored fitted parameters.

- *Calculate With Data*

  The Calculate With Data mode allows the user to make a single R-matrix calculation based on the starting parameters (i.e. energies, partial widths, etc.) they have entered into the previous tabs and considering the data that have been specified in the ***Segments*** . Practically speaking, this mode allows the user to make an initial check of the R-matrix calculation in order to determine the quality of their initial parameter values compared with the experimental data. Often when parameter values, especially relative interference signs, are unknown the user will want to fit "manually" using this option until a suitably close representation of the data is obtained and the automated minimization routine can be used.

- *Fit Data*

  After the user has obtained a reasonable starting point for the parameter values, AZURE2 can perform a least squares (χ^2) fit to the data by selecting the "Fit With Data" option and then clicking the "Save and Run" button. During this calculation, AZURE2 calls the minimization package MINUIT2 to automatically adjust the different fit parameters until the best fit has been reached. Every 1000 minimization iterations, the χ^2/N values and cross section information of each *calculation segment* being fit are output so that the progress of the fit can be monitored through the text box at the bottom of the ***Calculate*** tab and by viewing plots in the ***Plot*** tab.

  It is important to note that every time the user completes a fit, the output in all of the standard Output Files is over written. Therefore, if the user is testing different fitting options, the previous Output Files should be renamed or saved to another directory other than the specified AZURE2 output directory.

- *Extrapolate Without Data*

  The "Extrapolate Without Data" mode has variety of different uses. This mode is similar to the "Calculate With Data" mode in that only a single calculation is performed (i.e. no minimizations are performed). The difference is that here no experimental data are considered. The extent of the calculation must be specified using the *calculation segments* defined in the ***Segments*** tab. Possible uses for the extrapolation mode include, but are not limited to:

  – Interpolate a cross section curve using finer energy points than the measurement provides. This is very useful for plotting purposes.
  – Extrapolate the cross section into a near by unobserved region.
  – Calculate the total cross section after a fit has been obtained for differential cross section data or vice versa.
  – Calculate a previously unobserved cross section using only level parameters.

  - Using fitted data, calculate the cross section for the inverse reaction.

- *Perform MINOS Error Analysis*

  One of the more challenging aspects of an R-matrix analysis is to estimate the uncertainties on the fitted parameters. Since AZURE2 already utilizes MINUIT2 for its minimization procedure, it is natural to also use the uncertainty estimate subroutines that are available. In particular, MINOS, a comprehensive uncertainty routine available for MINUIT2 is used for the calculation. The details of the uncertainty calculation are not specific to the R-matrix method but are common to any kind of multiple parameter fit. For this reason the details of the uncertainty calculation are not reproduced here but can be found in the MINUIT2 user manual and in the publications of the Particle Data Group (`pdg.lbl.gov`). In particular the exact value of $\Delta\chi^2$ that corresponds to a 68% confidence level is debatable.

  One additional use of the MINOS uncertainty calculation is to test the constraints on the level parameters of the fit. If the MINOS calculation crashes, this very often indicates that a fit parameter is unconstrained by the data. The user can fix different parameters in the calculation (see Section 7.5 on how to fix parameters) in order to determine the offending parameter(s).

- *Calculate Reaction Rate*

  The final calculation option available in AZURE2 calculates the reaction rate using the specified set of parameters. In addition, the user must specify the entrance and exit *particle pairs* for the calculation by filling in the boxes labeled “Entrance Key” and “Exit Key” respectively. The user has the option to calculate the reaction rate over an even spacing of temperature by filling in the “Minimum Temperature”, “Maximum Temperature”, and “Temperature Step” boxes (temperatures should be in GK) or a file can be created with a list of specific temperatures. This is often more useful since the reaction rate changes rapidly with temperature at low temperatures. This file should list individual temperatures in a single column and can be specified by clicking the “Choose...” button after the “Use Temperature File” option has been selected. Reaction rates are output into the file reactionrates.dat.

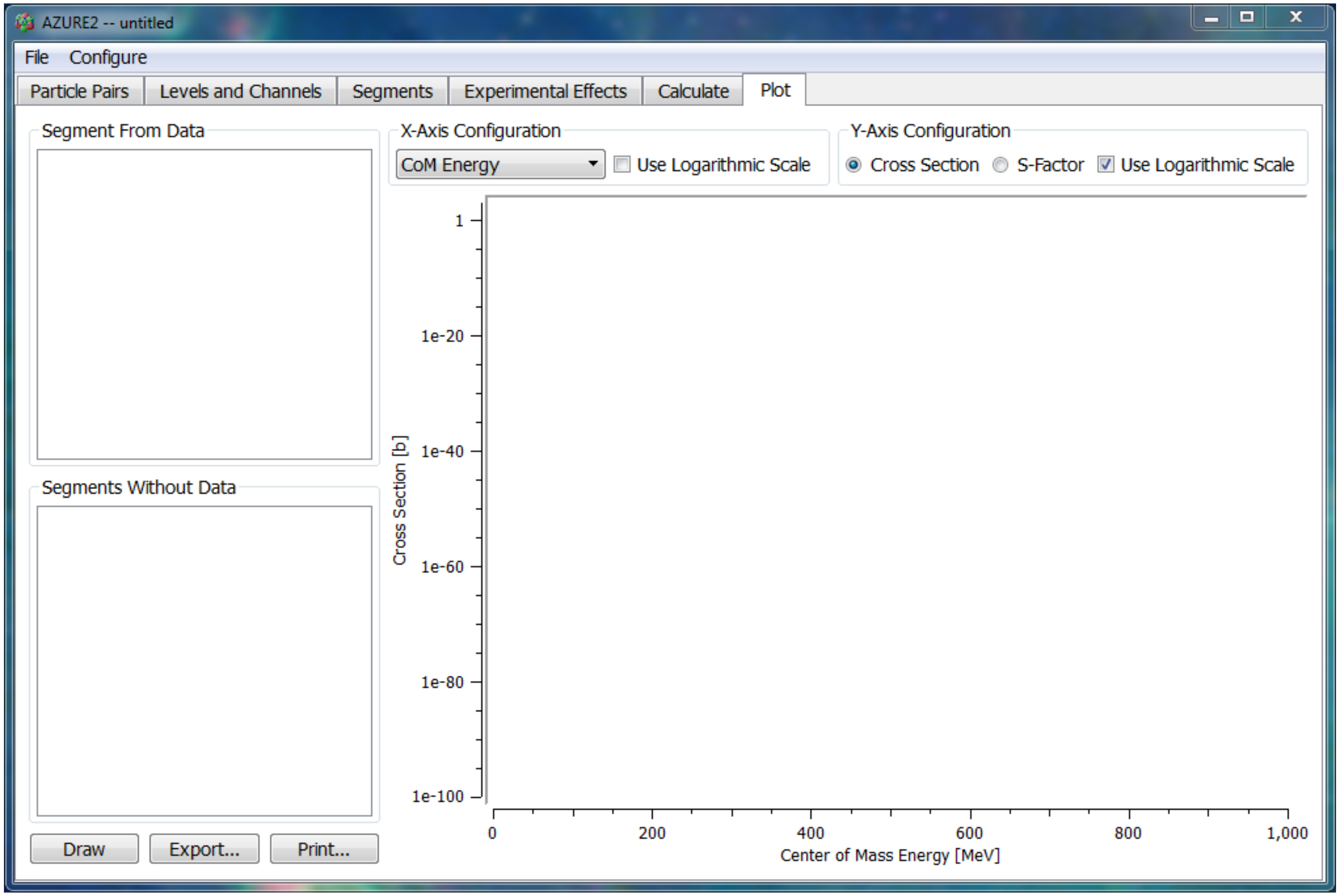
  The reaction rate is calculated using GSL’s adaptive step size integration routine to perform the integral

$$N_A\langle\sigma\nu\rangle = \left(\frac{8}{\pi\mu}\right)^{\frac{1}{2}} \frac{N_A}{(kT)^{\frac{3}{2}}} \int_0^\infty \sigma(E) E e^{-E/kT} dE \tag{8}$$

where μ is the reduced mass, N_A is Avogadro's number, and k is Boltzmann's constant. Units are in $cm^3 mole^{-1} s^{-1}$. Note that numerical integration of an excitation curve is limited to broad resonance structures. If the resonances under consideration become too narrow, the numerical integration may fail. In general it is suggested that the user use caution if the total width of the level is less than ≈1 keV. In the case of narrow resonances, the contributions to the reaction rate are often calculated by taking the sum of the single level narrow approximation.

7.9 Plot Tab

For quick simple plotting when using AZURE2, the plot feature in the ***Plot*** tab is provided (see Figure 16). A list of the *calculation segments* used with data that were defined in the ***Segments*** tab are given in the frame labeled "Segments From Data". Likewise, a list of *calculation segments* used without data, also defined in the ***Segments*** tab, are given in the frame labeled "Segments Without Data". To make a plot of the R-matrix calculation and the data for a *calculation segment*, click once on the segment to be plotted in either the "Segments From Data" or "Segments Without Data" frame. Then click the "Draw" button located at the bottom left of the tab. A figure will appear in the plot section on the right hand side of the tab. The options on the top of the plot area control what quantity is plotted on the horizontal axis (X-Axis Configuration frame), either "CoM Energy", "Excitation Energy", or "CoM Angle". The Y-Axis Configuration frame can be use to switch the vertical axis between "Cross Section" and "S-factor". Both axis frames have the option to be plotted in log scale by checking the appropriate box.



AZURE2 -- untitled

X-Axis Configuration: CoM Energy ▾ □ Use Logarithmic Scale

Y-Axis Configuration: ● Cross Section ○ S-Factor ☑ Use Logarithmic Scale

Figure 16: ***Plot*** tab.

The plot utility is also very useful for plotting several *calculation segments* simultaneously. When a *calculation segment* is selected by clicking on the segment line in the "Segments From Data" or "Segments Without Data" frame, it remains selected until the same line is clicked on again. If multiple *calculation segment* are selected simultaneously, and the "Draw" button is clicked, all the selected *calculation segment* will be plotted at once. This feature is very useful for comparing multiple *calculation segments* that contain data from different experiments. If the horizontal axis is set to "Excitation Energy", simultaneous plotting is also very useful for comparing data sets from different entrance and exit particle channels.

When first plotted, the axes will be automatically scaled. The user may zoom in by clicking and holding down the left mouse button and then highlighting a region. A zoom may be undone by then right clicking anywhere on the plot area. The user may also shift the central position of the plot manually by holding down the center mouse button and then dragging.

For simplicity, all plots of the R-matrix calculations are shown by red lines, while plots of data points are shown in black. Data points from different segments have different shapes that are selected automatically.

The plotter in AZURE2 is meant to be used as a quick plotting utility while doing calculations. For higher quality plots, the output from the AZUREOut_aa=*_R=*.out and AZUREOut_aa=*_R=*.extrap files can be used in an external plotting program.

Plots created in the plot window may also be exported and printed. These options are located in the lower left corner of the ***Plot*** tab, just to the right of the "Draw" button and are labeled "Export.." and "Print..". Plots can be exported to most of the standard image formats. Exported and Printed plots appear exactly as they are in the viewer allowing the user to zoom to customize the plotted area.

8 Output Files

The AZURE2 code outputs several files that contain the results of the R-matrix calculations. A very important difference between the Input Files and the Output Files is that while AZURE2 input is always in the laboratory frame, *quantities in the Output Files are always in the center of mass frame.*

- param.par:

  This file contains the initial fit parameters (these are formal parameters) that will be used by the code. These include resonance energies, reduced width amplitudes, etc. The quantities in this file are determined from the parameters in the Input File. The param.par file has limited use for the user, the file is created mainly as a check file.

- param.sav

  When AZURE2 finishes a fit, the final best fit parameters (again formal) are output into the file param.sav. The file contains the parameters in a form that is easily re-read by the code. In this way the user may select this file in order to reproduce the fit at a later time.

- parameters.out

  This output file contains the physical parameters that result from the fit.

- normalizations.out

  If the user has specified that normalization parameters be varied in the fit, the final fitted values are output in this file.

- intEC.dat

  This file contains information related to the external capture integrals that may be calculated when analyzing γ-ray data. This file is created because the calculation of the external capture integrals can take a significant amount of time. If the user does not change any of the *calculation segments* and does not add/remove any resonances of a new J^π (or add/remove any channels), this file may be loaded using the "Use" option under the "External Capture Integrals File" frame to speed up calculations. The user can change any of the attributes of a level (i.e. energy, partial width, ANC, etc.) and still use the same external capture integrals file. This is especially useful when fitting by hand at the beginning of an analysis when searching for good starting parameters or proper interference combinations. It is usually obvious when one must recalculate as the capture integrals as the calculated cross sections often change dramatically if improper input is given. But when in doubt, recalculate.

- intEC.extrap

  This file is similar to the file intEC.dat, except that it contains the information for the *calculation segments*. Similar rules apply for its use, except that now the user must not change any of the *calculation segments* between uses.

- param.errors

  MINUIT2 has built in methods for parameter uncertainty calculations. AZURE2 uses the MINOS routine when an uncertainty calculation is specified. The reduced width amplitudes and their associated uncertainties (asymmetric) are written out to this file.

- covariance_matrix.out

  This file contains the covariance and correlation matricies that result from a MINOS calculation, which give more complete descriptions of the fit parameter uncertainties.

- AZUREOut_aa=*_=R=*.out

  The numerical output of energy, angle, cross sections and uncertainties of the fit when the "Fit With Data" or "Calculate With Data" options are specified are contained in files of the type AZUREOut_aa=*_R=*.out, where the * is a place holder for the specific numerical value of the entrance or exit channel. The naming convention is a legacy of the original AZURE code where aa is the entrance channel *particle pair* index and R is the exit channel *particle pair* index. These values are specified during the creation of the *particle pairs* under the ***Particle Pairs*** tab. They are listed on the far left hand side of that tab. The file contains the following nine columns of information:

  - Col.1 - Center of Mass Energy (MeV)
  - Col.2 - Excitation Energy (MeV)
  - Col.3 - Center of Mass Angle (degrees)
  - Col.4 - Fit Center of Mass Cross Section (barns or barns/sr)
  - Col.5 - Fit Center of Mass S-factor (MeV b or MeV b/sr)
  - Col.6 - Data Center of Mass Cross Section (barns or barns/sr)
  - Col.7 - Data Center of Mass Cross Section Uncertainty (barns or barns/sr)
  - Col.8 - Data Center of Mass S-factor (MeV b or MeV b/sr)
  - Col.9 - Data Center of Mass S-factor Uncertainty (MeV b or MeV b/sr)

  When multiple *calculation segments* are present that share the same entrance and exit particle-pairs, these data are all output to the same AZUREOut_aa=*_R=*.out file. The data are printed to the file in the order in which they are sorted in the ***Segments*** tab and are separated by a double space.

- AZUREOut_aa=*_R=*.extrap

  These Output Files are similar to the AZUREOut_aa=*_R=*.out files, except that they are created when the "Extrapolate Without Data" option is used.

  - Col.1 - Center of Mass Energy (MeV)

  - Col.2 - Excitation Energy (MeV)
  - Col.3 - Center of Mass Angle (degrees)
  - Col.4 - Extrapolated Center of Mass Cross Section (barns or barns/sr)
  - Col.5 - Extrapolated Center of Mass S-factor (MeV b or MeV b/sr)

- chiSquared.out

  This file contains a list of the final χ^2/N (χ^2 divided by the number of data points) values obtained after the user has run a calculation using the “Calculate With Data” or “Fit With Data” options. On the final line the total χ^2 of the fit is printed.

- reactionrates.dat

  This file contains the temperatures and their associated reaction rates after running the “Calculate Reaction Rates” option.

9 Execution of AZURE2

The authors assume that normally AZURE2 can be executed graphically by clicking on the icon created after compilation. To launch AZURE2 in its usual graphical interface mode through the command line simply enter

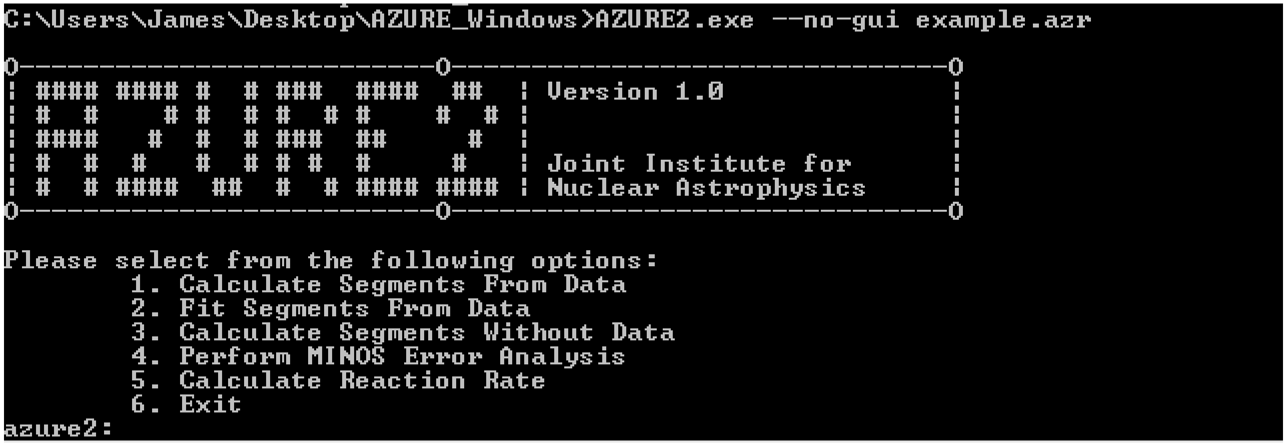
```
AZURE2 Input_File.azr
```

while in the file directory which contains the executable, where “Input_File.azr” is the name the user has decided to give their Input File (see Section 7.1). This Input file is optional.

Although AZURE2 is usually most conveniently accessed though its graphical interface, it may also be executed though the command line (Note that this option is not available on Windows or Mac). This is accomplished by entering

```
AZURE2 --no-gui Input_File.azr
```

where the option “--no-gui” signals AZURE2 to launch in the command line interface. The command line interface is intentionally similar to the interface of the `FORTRAN` version of the code. All of the Calculation Types that are available under the ***Calculate*** tab of the graphical interface are available here (see Figure 17).



```
C:\Users\James\Desktop\AZURE_Windows>AZURE2.exe --no-gui example.azr

O--------------------------O-------------------------------O
¦ #### #### #  # ###  ####  ##  ¦ Version 1.0              ¦
¦ #  #    # #  # #  # #    #  # ¦                          ¦
¦ ####   #  #  # ###  ##     #  ¦                          ¦
¦ #  #  #   #  # # #  #     #   ¦ Joint Institute for      ¦
¦ #  # ####  ##  #  # #### #### ¦ Nuclear Astrophysics     ¦
O--------------------------O-------------------------------O

Please select from the following options:
        1. Calculate Segments From Data
        2. Fit Segments From Data
        3. Calculate Segments Without Data
        4. Perform MINOS Error Analysis
        5. Calculate Reaction Rate
        6. Exit
azure2:
```

Figure 17: Command line execution of AZURE2.

One very important difference when running the code through the command line versus running through the graphical interface is that the Runtime options that are specified under the Configure Menu of the graphical interface, see Section 7.2.4, are not applied when AZURE2 is executed through command line. These options must be selected each time the user runs AZURE2 through the command line. The options are specified in the same way that the "--no-gui" option is selected except that more than one option may be selected at a time by separating each option flag with a space (the standard method that LINUX programs take command line options). The different options are

- --help

  This option lists all of the command line options available for AZURE2. It is the only option that may be given to AZURE2 without also specifying an Input File.

- --no-gui

  Executes AZURE2 in command line mode.

- --no-readline

- --no-transform

  This option corresponds to the "Do not perform parameter transformation" discussed in Section 7.2.4.

- --use-brune

  This option corresponds to the "Use Brune formalism" option discussed in Section 7.2.4.

- --ignore-externals

  This option corresponds to the "Ignore external width of internal width is zeroed" option discussed in Section 7.2.4.

- --use-rmc

  This option corresponds to the "Use RMC capture formalism" option discussed in Section 7.2.4.

- --gsl-coul

  This option corresponds to the "Use GSL Coulomb functions" option discussed in Section 7.2.4.

For example, if the user wanted to use the Brune parameterization and also ignore external widths for zeroed internal widths while running though the command line, they would enter

```
AZURE2.exe --no-gui --use-brune --ignore-externals Input_File.azr
```

into the command line.

10 Contact and Support

Questions regarding the AZURE2 package can be sent to

azure@nd.edu

11 Trouble Shooting

While AZURE2 is fairly stable, it is still possible to cause errors in a variety of ways. Here we give some examples of the most common issues the developers have experienced.

- When making a pure calculation it is often easy to accidentally specify an energy or angular range which is not kinematically allowed, especially when extrapolating cross sections to low energies. This usually results in a GSL error that can be caught by the code or may result in a hard crash.
- If multiple entrance channels are defined, problems can occur if calculations are performed very close to these thresholds of another reaction channel. The code will try to calculate the penetrability for each channel and if it is too small the GSL functions will give an error message. There is currently no work around for this issue and the user must exclude these points from the calculation. This issue is unfortunately exacerbated by the target integration, convolution, and reaction rate routines that create subpoints.
- In some problems where the fit parameters are very unconstrained the parameters may be varied to unphysical values causing the transformation codes to crash. This often occures during the actual fitting routine. The Brune parameterization is more stable in this regard which is one reason why

it is the recommended mode of operation. If continued crashes are experienced, it is likely that one or more of the fit parameters is unconstrained by the data. The user can try to fix different parameter(s) in the fit to isolate the offending parameter(s).

- The MINOS error analysis routine is very picky and can easily crash if a calculation has unconstrained parameters. It is often a good idea to try to run the MINOS analysis because its success strongly indicates that the fit is robust.
- Because of the issues of thresholds mentioned above, the reaction rate calculation can also crash if it is numerically integrating over a threshold. If this occurs, the user may have to calculate the rate by themselves by make a fine energy extrapolation and doing their own numerical integration (e.g. Simpson's rule).
- If the user does not format the input data files correctly the program can hang, not crashing but never finishing the calculations.

12 Acknowledgments

This work has been supported extensively by the Joint Institute for Nuclear Astrophysics (JINA) under NSF grant PHY0216783. In addition, the AZURE2 project has received developmental support from R. E. Azuma, C. R. Brune, J. Görres, D. Mountford, E.C. Simpson, and M. Wiescher.

References

[1] R. E. Azuma, E. Uberseder, E. C. Simpson, C. R. Brune, H. Costantini, R. J. de Boer, J. Görres, M. Heil, P. J. LeBlanc, C. Ugalde, and M. Wiescher. Azure: An R-matrix code for nuclear astrophysics. *Phys. Rev. C*, 81:045805, Apr 2010.

[2] F. Barker and E. Warburton. The beta-decay of 8he. *Nuclear Physics A*, 487(2):269 – 278, 1988.

[3] P. Descouvemont and D. Baye. The r -matrix theory. *Reports on Progress in Physics*, 73(3):036301, 2010.

[4] F. James and M. Roos. Minuit - a system for function minimization and analysis of the parameter errors and correlations. *Computer Physics Communications*, 10(6):343 – 367, 1975.

[5] A. M. Lane and R. G. Thomas. r-matrix theory of nuclear reactions. *Rev. Mod. Phys.*, 30:257–353, Apr 1958.

[6] M. Galassi *et al. GNU Scientific Library Reference Manual (3rd Ed.)*.

[7] N. Michel. Precise coulomb wave functions for a wide range of complex , and z. *Computer Physics Communications*, 176(3):232 – 249, 2007.

[8] C. W. Reich and M. S. Moore. Multilevel formula for the fission process. *Phys. Rev.*, 111:929–933, Aug 1958.

[9] M. E. Rose. The analysis of angular correlation and angular distribution data. *Phys. Rev.*, 91:610–615, Aug 1953.