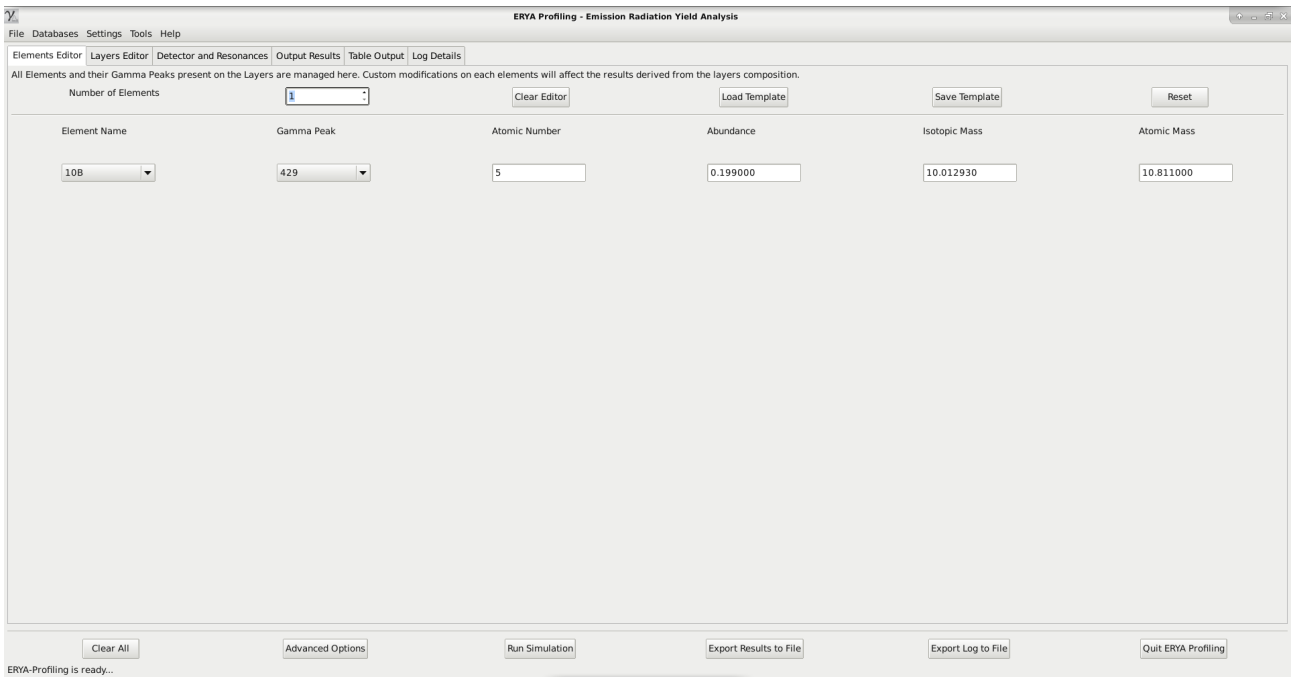
# ERYA Profiling

## User Guide & Tutorial




Manteigas Fonseca Pedro de Jesus

Version 3.00

ERYA Profiling USER’S MANUAL

SOFTWARE VERSION 3.00

https://sites.fct.unl.pt/nuclear/software

## About the authors:

Members of LibPhys-UNL, Departamento de Física da Faculdade de Ciências e Tecnologia - Universidade Nova de Lisboa (UNL), 2829-516 Caparica, Monte de Caparica, Portugal.

**June, 2019**

# Contents

# Introducing ERYA Profiling

## Welcome!

ERYA (Emitted Radiation Yield Analysis) Profiling is a software application designed to aid the PIGE analysis of thick samples, including inhomogeneous ones. This software are the natural companion of ERYA Profiling, and had new features related to a more accurate physical simulation of the experimental setup related to the detector and the interaction with the matter.

PIGE refers to Particle (mainly protons) Induced Gamma-ray Emission, resulting from nuclear reactions between the sample and the impinging particles. As the cross sections related to these reactions, at proton energies in the range 2- 5 MeV, favor nuclear reactions in the lighter isotopes, PIGE deals mainly with light element analysis.

The ERYA code calculates the elemental concentrations from ab-initio principles, surpassing the need of a comparative method based on standards, and achieving quick and easily the pretended results.

The performance of PIGE analysis based on this software depends on the physical model, experimental setup and measurement accuracy.

**ERYA is available for Windows, Linux and Mac OS X platforms.**

## Physical Theory of Gamma Ray Emission Yield Analysis on Heterogeneous Samples

The program evaluates the composition of a heterogeneous sample based on the corresponding nuclear reaction induced gamma-ray yields.

For a very thin target (meaning one for which the energy loss of the incoming beam is small), the yield corresponding to an isotropic gamma-line may be written as:

$$Y(E) = \varepsilon_{abs}(E_\gamma).n_p.\sigma(E).N_i.\Delta x \qquad (1)$$

where $\varepsilon_{abs}(E_\gamma)$ is the detector efficiency corresponding to the gamma-ray emission energy, $n_p$ is the number of incident protons, $\sigma(E)$ are the cross section at incident energy $E$ of the relevant nuclear reaction, $N_i$ is the number of nuclei of the relevant isotope per volume unit and $\Delta x$ is the target linear thickness. For a thick target, the above equation still applies for each thin layer parallel to the target surface (for normal incidence). For each layer of linear thickness $dx$ or mass thickness (mass per surface unit) $dx$, we may write a similar equation that may be expressed in terms of the relevant element mass fraction $f_m$:

$$dY(E) = \varepsilon_{abs}(E_\gamma).n_p.\sigma(E).f_m(E).f_i(E).N_{av}.A^{-1}d\Gamma \qquad (2)$$

being $N_{av}$ the Avogrado number and $f_i$, and $A$ the isotopic abundance and the atomic mass of the relevant element, respectively. Notice that unlike homogeneous samples trated on companion ERYA Profiling program, the atomic fraction and abundance are functions of energy.

The thick target yield results from the integration of eq. (2) over the range, $R$, of the protons in the target:

$$Y(E) = \varepsilon_{abs}(E_\gamma).n_p.N_{av}.A^{-1}\int_0^R f_m(E).f_i(E).\sigma(E)\,d\Gamma \qquad (3)$$

or:
$$Y(E) = \varepsilon_{abs}(E_\gamma).n_p.N_{av}.A^{-1}\int_0^{E_0} f_m(E).f_i(E).\sigma(E)/S_m(E)\,dE \qquad (4)$$

where $E_0$ is the incident beam energy and $S_m(E)$ represents the mass stopping power, expressed in energy per areal mass units.

ERYA Profiling take a step further to describe more accurately the interaction of the beam along the sample. The program will simulate the energy loss of the beam due to the interaction with the sample matter (straggling), simultaneously with the physical proprieties of the beam itself.

Once consequence of this physical model is that some physical quantities pass from discrete to a continuous distribution, even they already depend from the energy.

Since at a first approximation all distributions varies around an average value $\overline{E}$ that is around their maximum, it is possible the define the energy dispersion $\Delta E$ which is the distribution variable, in order to make the assumption that the energy variable on (4) can be represented by:

$$E = \overline{E} - \Delta E \tag{5}$$

Since the cross-section vary much more that the stopping-power of other quantities around the distribution peak, it is fairly accurate to assume that:

$$Y(E) = \varepsilon_{\text{abs}}(E_\gamma).n_p.N_{av}.A^{-1}\int_0^{E_0}\int_{-\infty}^{+\infty} f_m(E).f_i(E).F(\overline{E}).\sigma(\overline{E})/S_m(E)\,dE\,\overline{dE} \tag{6}$$

Where F is the distribution related to the proton beam and cross-section, that can be interpreted as the convolution of two different distributions: one related to the beam dispersion ($F_T$) and another to the straggling ($F_S$), each one with their one independent variables, albeit constrained by physical energy variable, due to (5). Joining together, it will result a triple integral given by:

$$Y(E) = \varepsilon_{\text{abs}}(E_\gamma).n_p.N_{av}.A^{-1}\int_0^{E_0}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f_m(E).f_i(E).F_T(\overline{E}).F_S(\overline{E}'-\overline{E}).\sigma(\overline{E}')/S_m(E)\,dE\,\overline{dE}\,\overline{dE'} \tag{7}$$

In order to evaluate (7) numerically this triple integral to return the yield, the program make the following simplifications:

- The sample are sliced by several layers with a constant depth and fixed atomic density $n_L$ using $10^{15}$ at/cm$^2$ units. And since energy loss of each layer $\overline{\Delta E_L}$ are related to the stopping-power by the formula:

$$\overline{\Delta E_L} = n_L.\epsilon(\overline{E_L}).(\overline{E_{L+1}} - \overline{E_L}) \tag{8}$$

The yield integration around the energy are replaced by a sequence of sums along the elementary layers, where each one depends from the stopping-power and depth. At each layer, the program will evaluate the energy loss, and subtract to the previous energy from the previous one.

In the end, the global yield calculation are the sum of all elementary yields given by each layer:

$$Y(E_0) = \sum_{L=1}^{N} Y_L(\overline{E_L}) \tag{9}$$

With this approach, many physical quantities depending from energy, can be treated as constants, since the energy at an elementary layer are constant at a fairy approximation.

The only exception are the cross-section that depends from the integration of two distributions, although the distribution variables can be scaled against the constant layer energy:

$$Y(E) = \varepsilon_{\text{abs}}(E_\gamma).n_p.N_{av}.f_m(E).f_i(E).A^{-1}/S_m(\overline{E_L})\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} F_T(x).F_S(y-x).\sigma(\overline{E_L}-y)\,dx\,dy \tag{10}$$

- The second simplification is to solve the unlimited domain of integration on (10), and since it is a double integral defined by two distributions the usual method is to apply an cut-off on their domain.

Taking the average measurement accuracy of many Nuclear Physics laboratories, it is satisfactory precise to make measurements until reaches 99.5 % confidence level, which are equivalent to a Gaussian Distribution until 3-sigma. ERYA will approximate the beam resolution as a Gaussian Distribution, while the straggling are detailed by a Vavilov Distribution or a Gaussian one, depending from the user settings on ERYA.

With this predicament the calculation of (10) are computed by evaluating first the factor:

$$Y_L\left(\overline{E_L}\right)=\varepsilon_{\text{abs}}\left(E_\gamma\right).n_p.N_{av}.f_m\left(\overline{E_L}\right).f_i\left(\overline{E_L}\right).A^{-1}.Z\left(\overline{E_L}\right)/S_m\left(\overline{E_L}\right) \tag{11}$$

And evaluate the weighted cross-section $Z\left(\overline{E_L}\right)$by ratio of two integrals with finite integration domain, to compensate the cut-off:

$$Z\left(\overline{E_L}\right)=\frac{\int_{x_0}^{x_1}\int_{y_0}^{y_1}F_T(x).F_S(y-x).\sigma\left(\overline{E_L}-y\right)dx\,dy}{\int_{x_0}^{x_1}\int_{y_0}^{y_1}F_T(x).F_S(y-x)dx\,dy} \tag{12}$$

The double integral domain are giving by a rectangular shape with boundaries:

$$x\in\left[-3\Omega,3\Omega\right];\ y\in\left[\lambda_0,\lambda_{0.995}\right] \tag{13}$$

Which are related to the 99.5% confidence level of the Gaussian and Vavilov distribution.

Some details about the ERYA implementation on Vavilov distribution are described on Appendix 3.

A more detailed description of the physical theory can be obtained from IAEA-TECDOC-1822, http://www-pub.iaea.org/books/IAEABooks/12235/Development-of-a-Reference-Database-for-Particle-Induced-Gamma-Ray-Emission-PIGE-Spectroscopy, Chap. 2 and 5.

## What's New on ERYA-Profiling?

This application is a full rewrite of a previous program package that was composed by a GUI handler written on LabView (are reuse parts of labView based ERYA), and a companion console-based program to run the simulation code.

The new ERYA is now fully implemented on C++ language, while the core framework, including the Graphic User Interface, is derived from a cross-platform library called wxWidgets.

wxWidgets is an open-source framework inspired on the Microsoft Foundation Classes, and compliant with the Standard Template Library, extending the compatibility to several operating systems and computer architectures, easing the port to several systems.

The new ERYA Profiling is available for Windows, Linux and Mac OS X, with support of most common architectures.

Windows and Linux versions are available for 32 and 64-bit Intel/AMD versions, where the 64-bit versions are the recommended versions. It was tested successfully on Windows 7 and 10.

Mac OS X are only available for 64-bit x86-64 Intel architecture, since Apple guidelines strongly discourage a 32-bit version. ERYA will not check Mac OS X version, but it was reported to work on Mac OS X 10.10 until 10.14 versions.

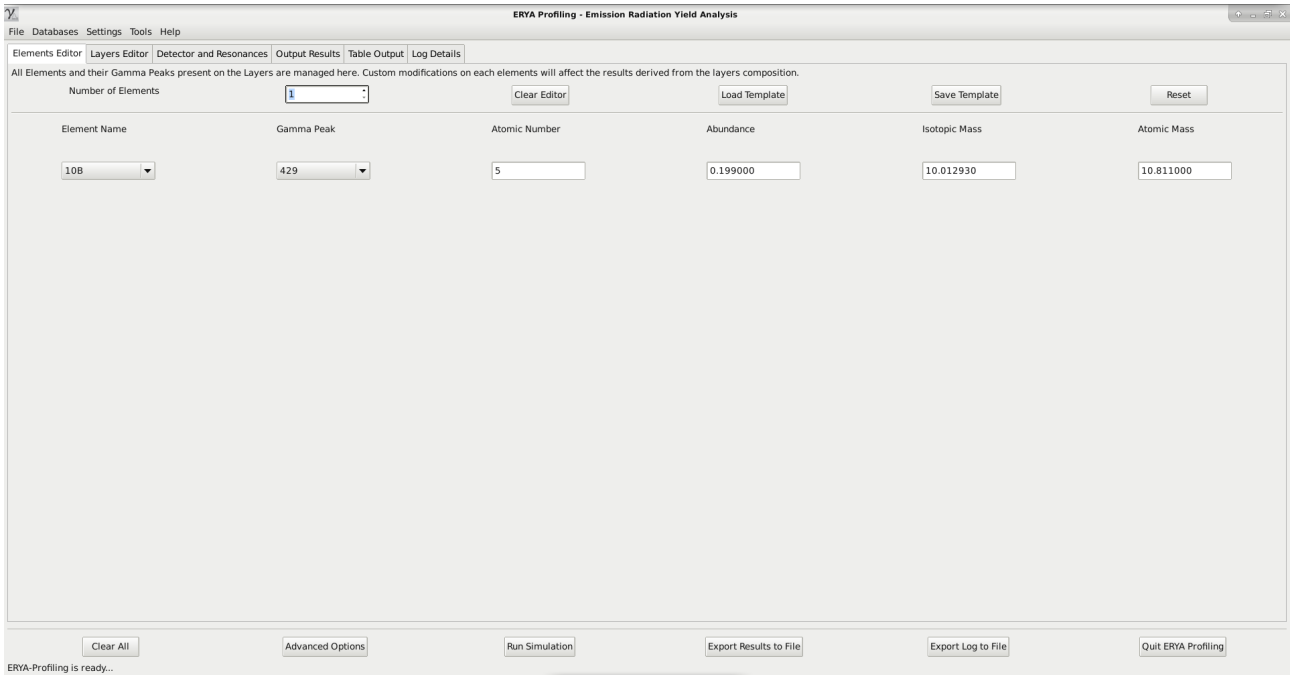
A special build for 32 and 64-bit ARM machines are also available for Linux computers, and are dedicated for ARM computers like the Raspberry 3 models.

All Linux packages are designed to work with any Debian derived Distribution, as long it contains the wxWidgets 3.0.4 libraries updated. It requires at least Debian 10 Buster, or Ubuntu 18.04 LTS versions to work.

Some databases are bundled on ERYA-Profiling package. The default database contains cross-sections as described in Appendix 2. It contains a ready to use Element's database file, three stopping-power files (for Ziegler's and SRIM parameters), and a Detector Efficiency file.

ERYA contains import tools intended to convert several source files (IBANDL, SRIM, plain text files, …) to the native ERYA database file formats.

The new ERYA-Profiling Main Program User Interface...




Some features of ERYA Profiling are:

1. All fundamental database files: Elements, the Detector Efficiency and Stopping-Power, use the XML document format.

2. ERYA-Profiling also handles IBANDL files, SRIM files, generic spreadsheet-like ASCII source files and Excel 2007 (and beyond) .xlsx files.

3. Compared to the old LabView based ERYA program, ERYA-Profiling handles more options to tune the simulation at user’s needs. Some additional tools are also available to the user interface.

## Quick Start!

The installation of ERYA on any supported operating system is straightforward.

Select the adequate package which matches your operating system and architecture and download it to your computer. As a rule of thumb, Linux versions should have the *amd64, i386, arm64, armhf* suffixes (first half are for Intel, and later half are for ARM machines). Windows versions are identified with *win64* or *win32* versions. Mac OS X version are identified by *osx* suffix.

**Windows:** All Windows packages contains the wxWidgets libraries and GNU C++ Runtime Library inside the executable package. The installation wizard program will install ERYA to the default Program Files folder and will create a Start Menu entry.

**Linux:** Linux versions of ERYA are available on Debian package format (deb), and any Linux distribution compatible with Ubuntu 18.04 LTS or Debian 10 Buster will install automatically.

► It is highly recommendable to use the apt/dpkg package manager from the Terminal application to minimize potential installation errors:

$ **sudo apt update** (Refresh the repositories)

$ **sudo apt dist-upgrade** (Update manually your system)

$ **sudo dpkg -i *.deb** (Install the package)

$ **sudo apt -f install** (Install the missing libraries from your Linux repository.)

► Debian distributions have more tight security features, and require to open a root shell from Terminal:

$ **su** (And type your root password)

# **apt update** (Refresh the repositories)

# **apt dist-upgrade** (Update manually your system)

# **dpkg -i *.deb** (Install the package)

# **apt -f install** (Install the missing libraries from your Linux repository.)

> **Warning:** ERYA was not tested on Linux distributions tailored to perpetual beta staging, like Arch Linux or their derivatives.

**Mac OS X:** For Apple Macintosh, once downloaded the Mac OS X package and unzipped it, the users just need to copy the ERYA-Profiling.app package to the Applications folder. Finally, click on ERYA-Profiling icon to run the software on Mac OS X.

## Uninstall ERYA

To remove ERYA from your system completely, follow these instructions:

**Windows:** Open the list of installed programs on Control Panel (usually, the path is: Start Menu > Control Panel > Programs and Features). Once the program list is loaded, select “ERYA-Profiling”. It will ask whether to uninstall the program, and answer “Yes”.

To remove any program profiles, open a Command Prompt (cmd), and execute the command:

$ **rmdir -s C:\users\name\Local Settings\Application Data\ERYA-Profiling-Win64** (64-bit)

**$ rmdir -s C:\users\name\Local Settings\Application Data\ERYA-Profiling-Win32** (32-bit)

**Linux:** Open a Terminal application and type the following commands:

$ **sudo apt remove eryaprofiling**

$ **rm -rf ~/.ERYA-Profiling**

**Mac OS X:** Delete the application from the Applications folder on your system.

Open the Terminal.app and execute the following command:

$ **rm –rf ~/Library/Application Support/ERYA-Profiling-OSX/**

## Running ERYA for the First Time

**1.** Once ERYA-Profiling is successfully installed, it will need a configuration file in order to work properly.

**2.** When ERYA-Profiling runs for the first time, it will display a warning that no configuration file was found, triggering the Setup Wizard. In case of a previous ERYA-Profiling installation, the program will load the old configuration file, and the associated databases. To avoid this, read the Changing Setup Settings section, and restart these steps.

**3.** If no previous configuration file exists, ERYA will start a wizard, guiding the user for the next steps. The wizard will ask the user to select from the bundled database files, which ones will be the Element Database, Detector Efficiency and Ziegler Parameters databases. While it is possible to skip the wizard, it is not recommendable to use ERYA without any loaded databases.

**Important Note:** The wizard on next three steps will use the native operating system's file dialog controls, and the user should take note:

- Linux versions will open the file dialog on ***/opt*** directory where ERYA-Profiling software locates. For each database the user should click the “ERYA-Profiling” directory, and then the correct file as indicated on each step.

- Windows versions will open the file dialog on “Program Files”, and the user should open the “ERYA-Profiling” folder to access the database files.

- Mac OS X will point the file dialog inside the hidden “Contents” folder of ERYA-Profiling-OSX.app bundle. All databases are located inside “MacOS” folder, and the user should open the ones dictated by the next steps.

**4.** ERYA’s wizard will ask the user to select the start-up Element Database from the ERYA’s package. Normally the file opening dialog will point to the program directory, and here the user should select the file with an **epd** extension.

**5.** The same logic will apply for the Detector Efficiency file, with an **epsd** extension. The bundled file corresponds to the HPGe detector used at Lisbon. The user must create or change the file according to the detector’s efficiency of his laboratory. Further details are described on the Detector chapter of this manual.

**6.** Finally, choose the file for the Stopping-Powers, with an **epsz** extension. ERYA’s package contains three options (including the converted SRIM tables, that is the recommend one), leaving to the user to choose among them.

**7.** Once selected all necessary files, the wizard will ask the user to select a local or portable profile.

The key differences are displayed on this table:

| **Operating System** | **Windows** | **Linux** | **Mac OS X** |
|---|---|---|---|
| **Portable Setting** | **In the same program directory** | | |
| **Local User Profile** | **C:\users\name\Local Settings\Application Data\ERYA-Profiling-Win64*** | **/home/name/.ERYA-Profiling/*** | **/Users/name/Library/Application Support/ERYA-Profiling-OSX/*** |
| **Config Name** | **ERYA-Profiling-Win64.conf** | **ERYA-Profiling.conf** | **ERYA-Profiling-OSX.conf** |

**Portable Profile Setting:** Once the configuration file is written on the root program directory, the setup is complete. Any changes on default Detector, Elements and Ziegler will overwrite the original ones used by the program, since the same working directory was chosen.

**Local User Profile Setting:** This is the recommended configuration, since the wizard will store a copy of the default Database, Ziegler and Detector files on the directory defined on the previous table. Even if the user changes the local databases, the original ones are left unchanged, being available as an informal backup allowing a program reset.

**8.** Once all wizard steps are done, the configuration file is created and tested. If ERYA do not find any problems during the Databases loading phase, the main GUI interface will appear.

**9.** ERYA-Profiling is now ready to use.

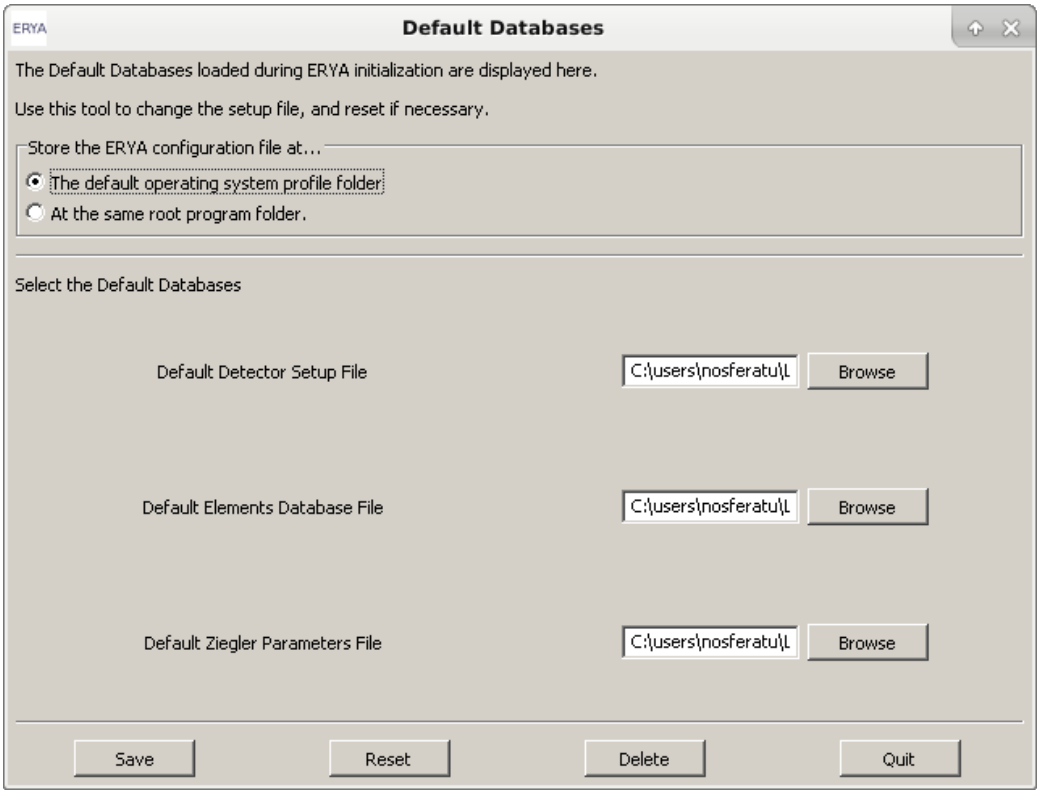
**Warning:** In case of any errors related to the databases or configuration files, read the *Troubleshooting* guide to deal with them.

► On next start-ups, the program will load the configuration file automatically, and the main GUI starts without additional user input.

## Changing Setup Settings

When needed, the user can change any start-up databases at any time, and ERYA provides a tool accessible from the menu: Database > Setup Default Databases.

The actual configuration will be displayed on the different widget controls:




► Once the parameters are changed, select the following buttons to:

"Save" will create a new configuration file directly.

"Reset" will clear the widget contents, without affecting the actual configuration.

"Delete" will delete all configurations, and force to start a new Setup Wizard.

"Quit" will close the widget leaving the actual configuration untouched.

**Tip:** To erase all damaged configuration files, and local copies of ERYA's databases, select the *Delete* button. This will replace the need of a manual deletion of the local profile folder described on the *Quick Start* chapter.

## Setup Troubleshooting Guide

► This chapter is a quick guide to fix some common problems that may occur with the utilization of this software.

**1. ERYA warns with the message "Invalid Database", when one or all databases are absent or cannot be loaded at all.**

►At top menu, select: Database > Setup Default Databases, and select the correct files.

- If the problem persists, select "Delete" to delete all local data, and start a new Setup Wizard (all custom changes will be lost!)

- Alternatively, delete the configuration files and profile folders manually.

- Avoid using third-source database files to perform a clean installation, since it could be the source of problems.

**2. ERYA cannot write any file, even the configuration file, on certain folders.**

► This problem occurs when ERYA is trying to write on a protected folder that requires additional privileges. To solve this problem, select other folder that does not require special permissions.

- Warning: the user should not run ERYA-Bulk with administrator privileges, in order to improve stability and security!

**3. ERYA's dialog sizes are too small/big.**

**- Label's font size on dialog are too tiny or cropped.**

**- The whole interface does not fit to the computer screen.**

► ERYA GUI's code relies on the automatic window sizing delivered by the wxWidgets framework, but in some cases, issues are inevitable. To correct this problem, check the following:

- Select the default operating system's DPI (around 72 or 90 dots per inch) to avoid abnormal font sizes on dialog labels.

- ERYA's optimal screen size is between 1280*720 and 1920*1080. Avoid screen resolutions below 800*600, since it will not display many dialog controls correctly.

- ERYA's main window will allocate scroll bars for smaller screens.

- If necessary, use the mouse to grab and change the dialogs windows sizes manually.

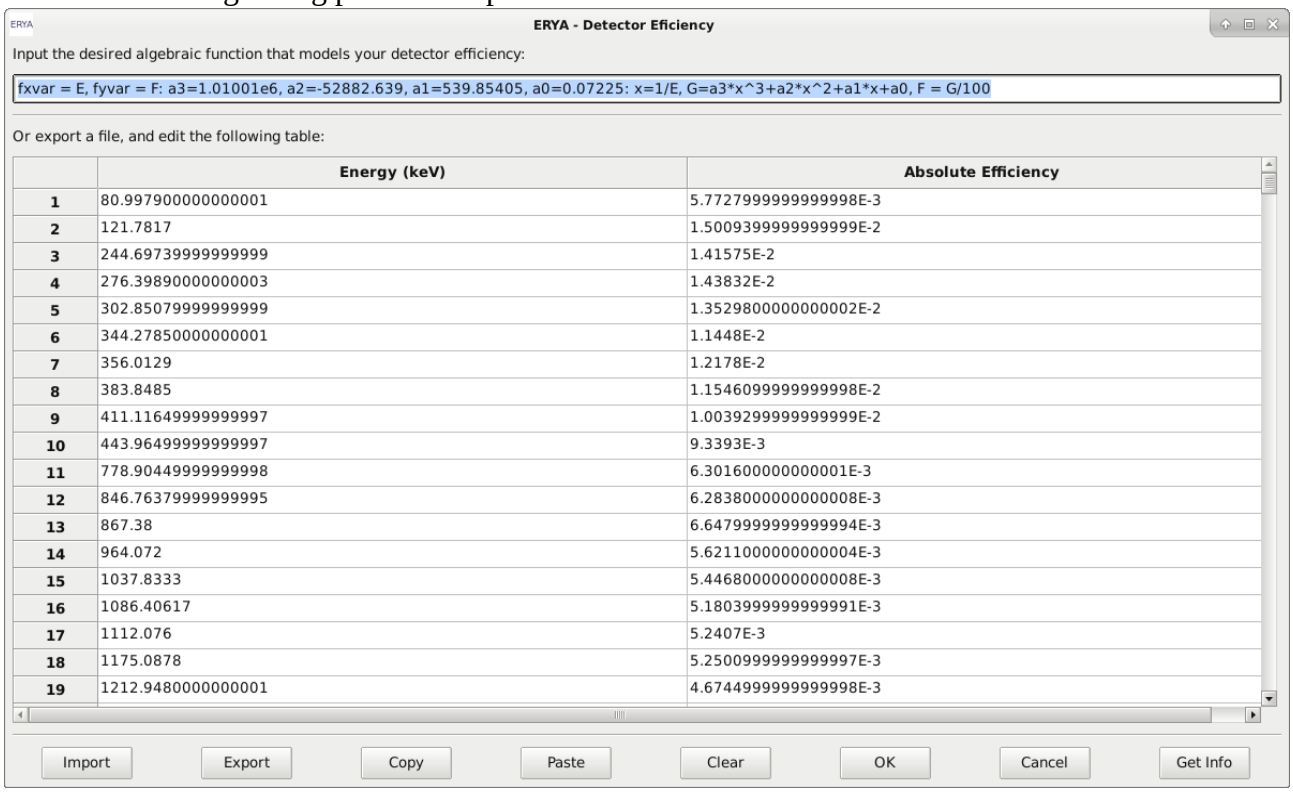
**►Manual resizing of dialog windows may be needed when ERYA runs on computers with 4K (3840*2160) monitors, or higher.**

# Explore the Database Management Tools

## Detector Efficiency

One of the key core databases is the Detector Efficiency file.

To manage the Detector Efficiency file, open the widget from Databases>Detector Efficiency, and then the following dialog panel will open.




| | Energy (keV) | Absolute Efficiency |
|---|---|---|
| 1 | 80.997900000000001 | 5.7727999999999998E-3 |
| 2 | 121.7817 | 1.5009399999999999E-2 |
| 3 | 244.69739999999999 | 1.41575E-2 |
| 4 | 276.39890000000003 | 1.43832E-2 |
| 5 | 302.85079999999999 | 1.3529800000000002E-2 |
| 6 | 344.27850000000001 | 1.1448E-2 |
| 7 | 356.0129 | 1.2178E-2 |
| 8 | 383.8485 | 1.1546099999999998E-2 |
| 9 | 411.11649999999997 | 1.0039299999999999E-2 |
| 10 | 443.96499999999997 | 9.3393E-3 |
| 11 | 778.90449999999998 | 6.3016000000000001E-3 |
| 12 | 846.76379999999995 | 6.2838000000000008E-3 |
| 13 | 867.38 | 6.6479999999999994E-3 |
| 14 | 964.072 | 5.6211000000000004E-3 |
| 15 | 1037.8333 | 5.4468000000000008E-3 |
| 16 | 1086.40617 | 5.1803999999999991E-3 |
| 17 | 1112.076 | 5.2407E-3 |
| 18 | 1175.0878 | 5.2500999999999997E-3 |
| 19 | 1212.9480000000001 | 4.6744999999999998E-3 |

This is straightforward to use, with all possible operations clearly visible.

This editor combines a text box, intended for the user to code an algebraic function, describing the variation of the Detector Efficiency with Energy, and a two-column table to fill any experimental values for the Detector Efficiency that will serve the same purpose.

Both fields can be filled on the built-in editor and stored in the profile file. However, ERYA will choose the function field by default, if it is correctly defined, and will ignore the table.

In absence of the algebraic function, the user should fill the highest possible number of experimental values on the built-in spreadsheet editor. The accuracy will be better if the difference between energy values is smaller, since the program will make a segmented linear interpolation between the values of the efficiency table.

## Managing the Detector Efficiency profile file

While it is possible to use the built-in spreadsheet editor, the user can also import any compatible experimental values from external files.

The native Detector Efficiency file is a XML file, which stores the experimental efficiency points, and the additional user defined efficiency function. This file has an **epsd** extension.

It is possible to export and import the experimental efficiency (the custom function is discarded) to or from an ASCII file, required to be a full numerical two columns of data, separated by a space or tab.

Excel files are also supported in a similar way. The Excel file should be a simple sheet with only two columns of numbers. There is no problem in adding labels on the first row, which will be ignored. Any graphics or special attachments to Excel file will also be ignored. [The ERYA's Excel file filter only supports Excel 2007 and beyond version files (with **xlsx** extension), and only recognizes cells with names and numbers.]

It is also possible to copy the built-in table contents to a text or spreadsheet file, but this feature is very error prone. In order to copy the table, select the contents from the editor, and click "Copy", then paste to your text editor. The reverse is similar, once copied from the source, it requires to click on the first target cell, and then click "Paste". If necessary, clear the built-in table by clicking "Clear".

Modifications can be stored by using the "Export" button, and then choose among the native XML (epsd), ASCII (txt) and Excel (xlsx) formats. You should overwrite the start-up Detector Profile, as you defined in the initial setup and described in the "Quick Setup" chapter, to make permanent changes.

Stored and additional Detector profiles can be loaded from the "Import" button, and they will overwrite the previous content on the editor.

In order to make modifications valid only for a session, after the alteration of values made by importing or filling up the spreadsheet, the user should click "OK". After finishing the session, the changed values are lost, since the profile file was nor changed.

To discard any editions made by this editor, click "Cancel", leaving the original data untouched.

Optionally the user can view or edit a brief commentary by clicking "Get Info" button, that will open a plain text editor capable to made simple editions. This text are stored on the respective XML databases used by ERYA.

## Custom Efficiency Function

The optional algebraic efficiency function is implemented in ERYA as an text box for the user to code a custom function, overriding the table.

In technical terms, it is a trimmed-down macro language designed to code custom functions without needing to recompile the application for different kinds of functions. The functionality and syntax are inspired from the BASIC language dialects found on programmable calculators. A more detailed documentation about the language may be find in the appendix 1.

By default, a function will be inscribed in the upper window of the Detector Efficiency widget, as, for example:

**fxvar = E, fyvar = F: a3=-2078.4, a2=-312.84, a1=5.006, a0=0.001: x=1/E, F=a3*x^3+a2*x^2+a1*x+a0**

It represents symbolically the following function:

$$\epsilon_{\text{abs}}\left(E_\gamma\right) = 0.001 + 5.006\,E_\gamma^{-1} - 312.84\,E_\gamma^{-2} - 2078.4\,E_\gamma^{-3}$$

The special commands **fxvar** and **fyvar** followed by the assignment sign (=) and the variable name, are used to define the independent and dependent function variable, respectively. The following entry, separated by commas (a3 to a0), refer to the coefficients of the function. The final entry defines the function after redefining the independent variable.

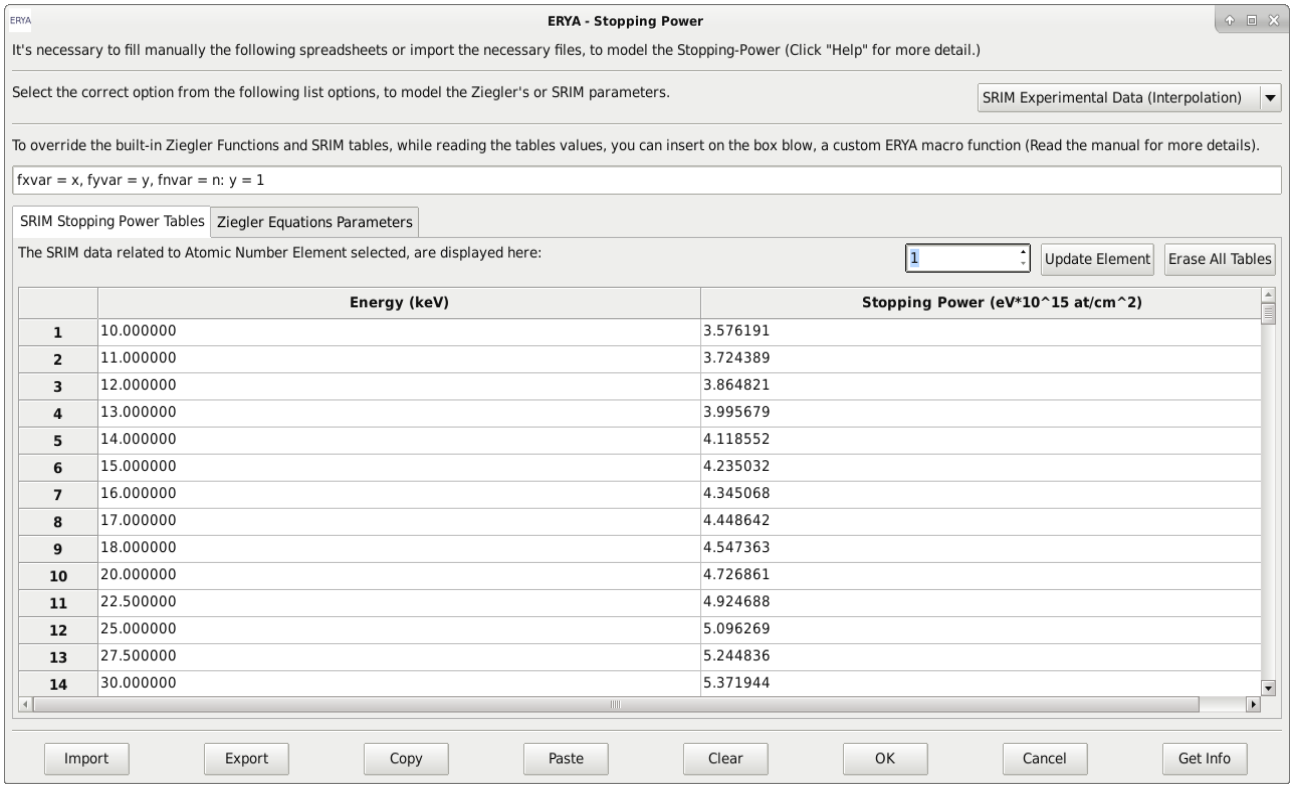
If the user is willing to use a function of the type referred above, then he (she) only has to modify the coefficients overwriting the ones in the window. If more coefficients or other type of function is required, it must be typed in the window, respecting separation spaces and symbols, as the ones shown in the example.

## Stopping-Power Database

As the Detector counterpart, the Stopping-Power widget follows the same layout and functions, albeit exists some key differences.

ERYA-Bulk inherited the original Ziegler's Parameters model from the previous LabView ERYA software. In order to expand from the old software limitations, ERYA now supports custom algebraic functions and the SRIM tables.

The Stopping Powers is opened from Databases > Stopping Powers:




The Stopping Powers profile file stores not only the Ziegler's Parameters derived from 1977 and 1991 models, but also the converted SRIM tables for all elements, and a custom ERYA macro.

The data is stored in a XML file with **epsz** extension and contains all data editable by this panel.

The top pull-down menu selects the stopping-power model applicable to all numerical simulations. The first two options correspond to the Ziegler's model (either the 1977 or 1991 equations, hard-coded on program), that requires to fill the "Ziegler" tab with the parameters on the built-in spreadsheet as displayed on the figure.

The algebraic function and SRIM options are explained with more detail on the next sections.

## Defining a Custom Stopping-Power Function

The user can define a custom function for the stopping-power in function of energy. To use this functionality, select the "Custom Function" mode on the Stopping-Power mode menu, and fill the required macro code on the input box. When correctly defined, ERYA will make a syntax check and validate the custom function code.

ERYA expects the custom function to represent the stopping-power in $10^{15}$ at*eV/cm$^2$ units in function of energy in keV.

The coding standards of the custom stopping-power function are the same of the detector's efficiency, and the only real difference is a third obligatory command called **fnvar** that create a vector of 16 elements. The first variable of such vector had the same name as defined by the assignment sign (=). The other 15 variables are simply the same name base, appended with a number from 1 to 15.

This means that if the user defines a *zn* variable as **fnvar**, as displayed here:

**fxvar = x, fyvar = y, fnvar = zn : …**

Then the following variables are created on the interpreter's memory stack: *zn, zn1, zn2, …, zn15*.

All 16 variables created by the **fnvar** command will be used by ERYA to copy the entire line of the Ziegler's Parameters spreadsheet automatically to those variables:

► The first base variable will be the Element's Atomic number;

► The next 12 variables will be the A-1 to A-12 parameters;

► The variable with *13* suffix will be the Atomic Mass.

► The variable with *14* suffix will be the Atomic Density.

► The variable with *15* suffix will be the Mean Ionization Potential (Bloch Parameter).

**Example:** To use the following function, defined by the combination of two expressions as follows:

$$S(E) = S_{low}, 0 < E < 1000$$

$$S(E) = \frac{S_{high} S_{low}}{S_{high} + S_{low}}, E \geq 1000$$

$$S_{low} = a_1 E^{a_2} + a_3 E^{a_4}; S_{high} = \frac{a_5}{E^{a_6}} \ln\left(\frac{a_7}{E} + a_8 E\right)$$

The constants $a_1$, … , $a_8$ are placed on the Ziegler's Parameters table and ERYA will load automatically according to the element's atomic number line.

Using the *fnvar* command, all necessary constants are created from the base variable "a".

Applying all ERYA’s Macro language rules, the custom function should be written as the following:

**fxvar = x, fyvar = y, fnvar = a : low = a1*x^a2 + a3*x^a4 , high = (a5/x^a6) * ln (a7/x + a8*x) , y = (x>0) * (x<1000) * (low) + (x>=1000)*((high*low)/(high+low))**

Since the two auxiliary functions depends on the independent variable, and pass their values to the main function, such algebraic expressions should never be separated by the colons symbol.

ERYA’s Macro interpreter stores the last expression (including all terms separated by commas) after the last colon, as the main function, and considers everything left as fixed constants.

**Warning:** If the user define, wrongly, the example with:

**fxvar = x, fyvar = y, fnvar = a : low = a1*x^a2 + a3*x^a4 , high = (a5/x^a6) * ln (a7/x + a8*x): y = (x>0) * (x<1000) * (low) + (x>=1000)*((high*low)/(high+low))**

The interpreter will store the “high” and “low” functions as fixed valued constants, and give wrong results for each value “y” in function of “x”, since the auxiliary functions were incorrectly declared as constants.

For more details about the language see the appendix.

Piecewise functions, such as this example, must be defined with conditional terms, that use simple relational expressions inside parenthesis multiplied by each piecewise function inside parenthesis too. All piecewise functions are joined together by sums.

**Note:** The Custom Stopping-Power Function is mainly intended for program tests, where it may be useful to use a constant stopping-power for certain samples.

For example, to define a constant stopping-power of 1.77 $10^{15}$ at/cm$^2$, just write the following code:

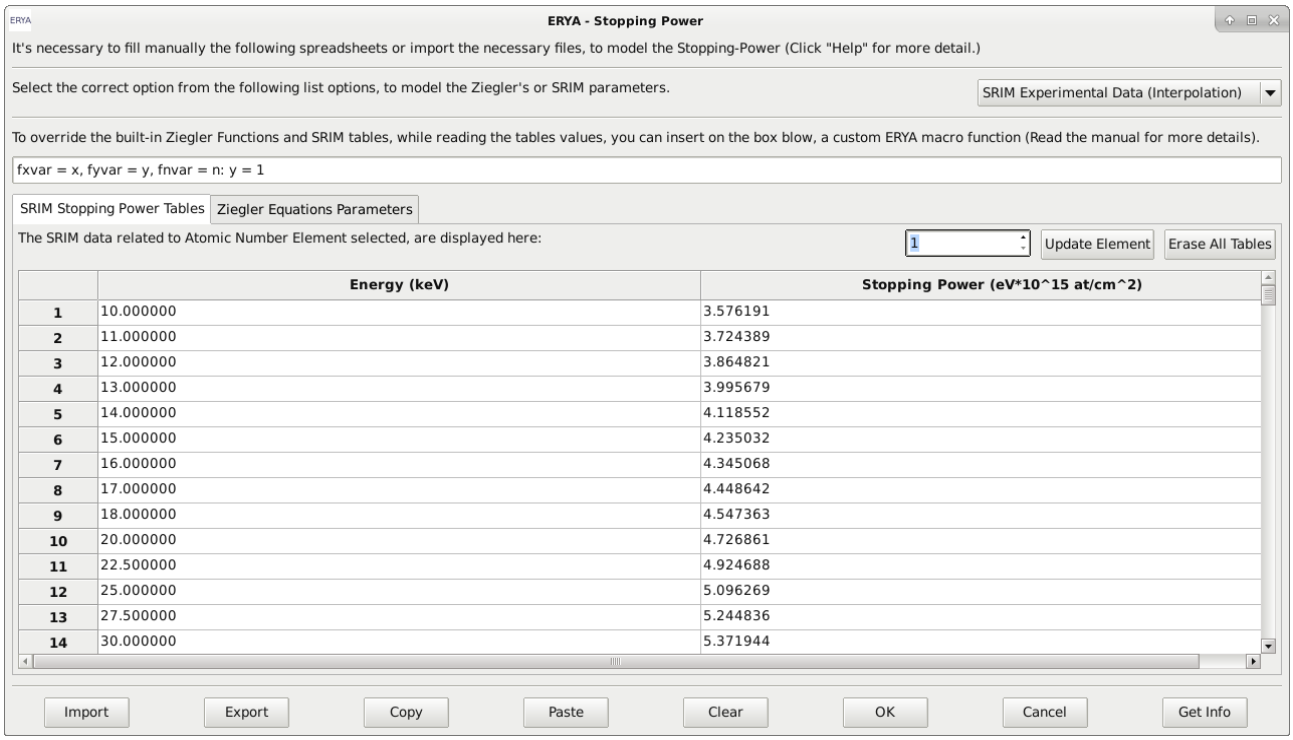
**fxvar = x, fyvar = y, fnvar = zn : y = 1.77**

Or alternatively, write with a symbolic constant on function:

**fxvar = x, fyvar = y, fnvar = zn : c = 1.77 : y = c**

## Using the SRIM Stopping-Power Tables

To manage the converted SRIM tables, select the pull-down menu to "SRIM", and select the "SRIM" tab to edit the chain of spreadsheets in a fashion like the Detector's Efficiency.




The SRIM's Spreadsheet editor can only display an element at a time, even several elements reside on memory, in the original file. In order to display other tables, just change the numerical value on the switch-box that corresponds to the Element's Atomic Number.

Any edition of the selected SRIM table will only be stored in memory, if the user clicks "Update Element" button. The other "Erase All Tables" button will delete all tables from memory.

If the user wants to update the tables from the original SRIM® stopping-power ASCII files, he must follow the import instructions on the next section.

All SRIM tables will have experimental values for the stopping-power in $eV/10^{15}$ $at/cm^2$ units versus energy in keV units; interpolation between tabled values will be done during the numerical calculation.

## Import and Export Stopping-Power to different formats

The native Stopping-Power file is a XML file with **epsz** extension. If the user desires, it is possible, like on the Detector widget, to import and export the current built-in table content to a text file or an Excel file.

►ERYA can parse directly any unedited SRIM Stopping-Power output files generated by SRIM® software. (**And this is highly recommendable, to avoid conversion mistakes**)

When the Import button is activated with ASCII import, and ERYA detects that is a SRIM Stopping-Power table, it will convert to the correct Element's atomic number table on the SRIM tab, since all necessary information like the atomic number, unit conversion constants and some additional information are directly available.

**► ERYA cannot save stopping-power tables back to SRIM® tables file formats.**

Instead, it will export the active element's stopping-power directly to an ASCII file, exactly as ERYA displays.

**► When the user exports to ASCII files, the program will take attention to the selected tab mode beforehand**

This means that it will apply the Ziegler's rules when the tab is the Zielger's Parameters, and the SRIM's rules when the correspondent tab is the SRIM ones. In particular, the active atomic number SRIM table is also considered.

If the user opts to export Ziegler's Parameters to an ASCII file format, any custom "Element" names are replaced by the number of the corresponding row. When imported from an ASCII file, it will rename to the chemical symbol, since ERYA contains a hard-coded dictionary of all elements.

►**When the user imports a raw numerical data table**, for example a new Ziegler's Parameters table or a stopping power interpolation data, as ASCII, the user should select the Ziegler's tab or the correct atomic numbers on the SRIM tab **before importing the file.** An error message or an erroneous validation of the import will happen otherwise.

►The same cautions and procedures are also applied when export or import Ziegler's and SRIM tables using Excel xlsx files, since ERYA requires to know manually the element's atomic number if the source file corresponds to a SRIM table. When the user exports an Excel file with Ziegler's Parameters, the Elements names are also stored on the Excel files.

## Cautions between ERYA-Profiling and ERYA-Profiling Ziegler’s Parameters

Both programs use the same XML based document with *epsz* file extension, still exists a small difference: the ERYA-Profiling version had two additional columns for the density and mean ionization constants.

► This means that if ERYA-Bulk reads the ERYA-Profiling Ziegler’s database it will discard the two unused columns (and save it will rebuild to the ERYA-Bulk version).

► And ERYA-Profiling will read the ERYA-Bulk version if the user accept the warning dialog, and then ERYA-Profiling will add a filler value “1” to all densities, and “10” for all mean ionization parameters.

For other formats (Excel and ASCII), ERYA-Profiling will process the files by the following rules:

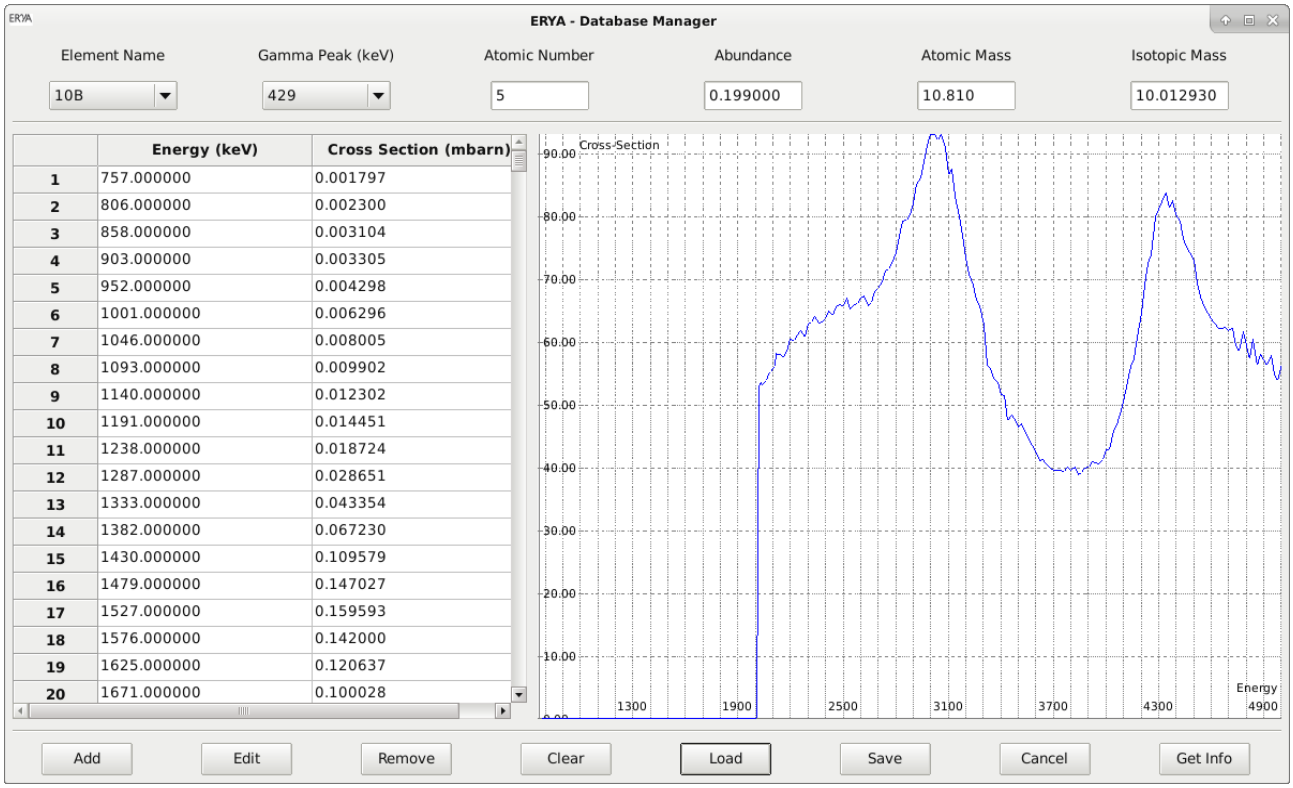
► ERYA-Profiling will load all ASCII, XML and Excel files created by ERYA-Bulk, but will warn that the absence of the density and mean ionization columns, asking to continue the import, or abort at all. If the ASCII or Excel files corresponds to a interpolated table with two columns (derived from SRIM), there’s no ambiguity.

► When the user saves a Ziegler’s Parameters to ASCII and Excel files, it always save the density and mean ionization columns. Those files will fail to open on ERYA-Bulk.

## Elements Database

The Elements Database contains all the necessary elemental parameters, including cross sections for the numerical calculations. The default database contains cross-sections as described in Appendix 2.

To open the Database Manager, just click on menu: Databases > Elements:




Using this tool, it is possible to handle the contents of one database at a time (normally the default loading database when ERYA starts).

►By clicking on "Element Name" pull-down box, the selected element/isotope will refresh the "Gamma Peak" pull-down box, and then the user can select the available gamma emission peak for that element.

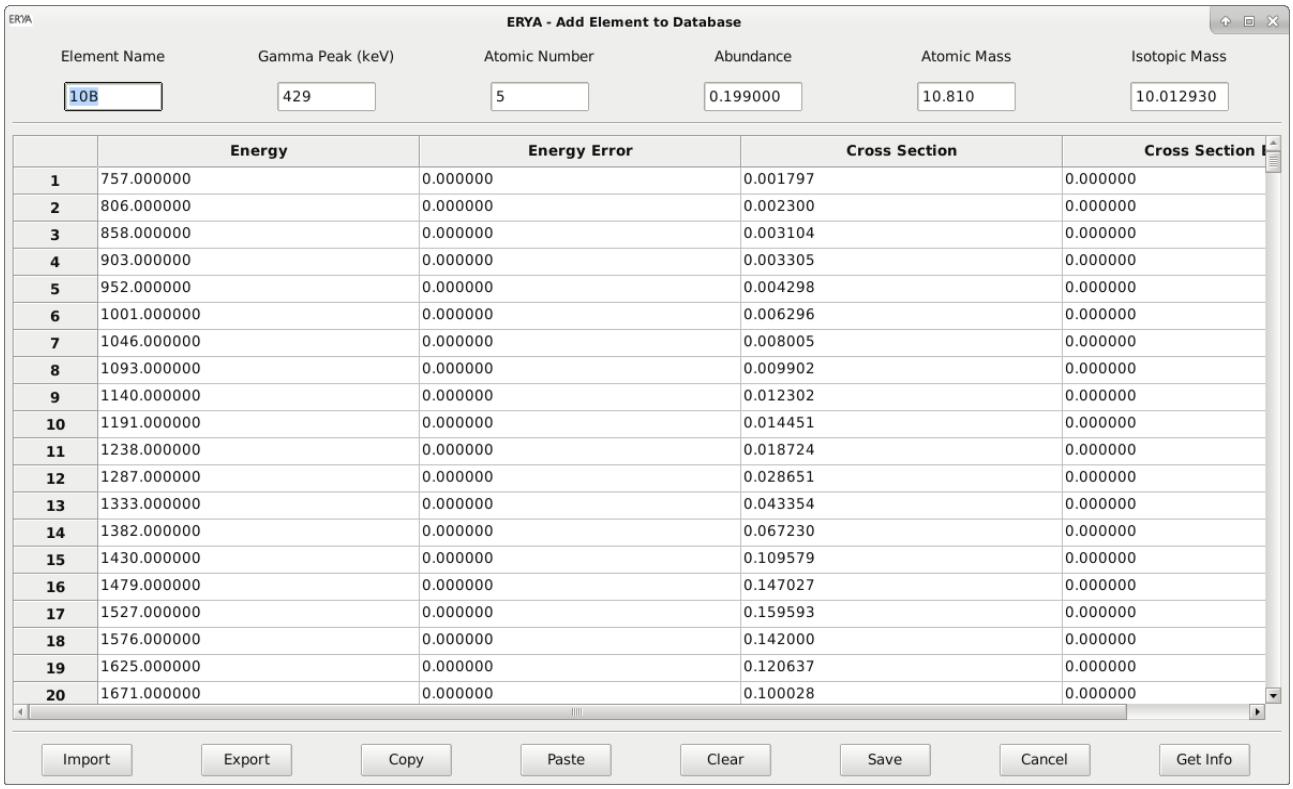
For each Element/Gamma pair, a read-only Energy/Cross-Section table and graphical plot will be drawn. Up to four element physical proprieties will also be displayed.

The graphic plot can be zoomed or fitted to the screen, using the mouse. Additional commands are available on a right-click menu.

## Editing the Elements Database

A common operation on database management is the possibility to add or edit elements.

►To edit an element content, select the desired element and gamma peak from the *Database Manager* dialog panel, and click on "Edit" button, that will open a new dialog window:




**Note:** The same widget is displayed when the user clicks on "Add", but it will open a blanked dialog. Further explanation about elements editing will be detailed later.

►To delete an element, select the element and gamma peak to be removed, and click "Remove". ERYA will ask if you want to delete the selected element and gamma. Notice that the changes are only permanent if the edited database is saved.

►Select "Load" to load a database from a file. The current database on memory will be replaced.

►Select "Save" to update the database file content.

►To discard any editions made to the current database, hit the "Cancel" button.

►Select "Get Info" to fill additional information about the active database.

## Adding/Editing new Elements

The user can create new elements or edit old ones using the adequate controls on the *Database Manager*, that will open a new dialog dedicated to the edition of all necessary parameters.

► Any element should have a unique name and gamma peak energy value. There's no harm to add suffixes to the Element's name, if they are supported characters. (See warning below.)

Every element (isotope) register must have an Atomic Number, its Abundance (ranging from 0 to 1), its Atomic and Isotopic Mass.

► The element's cross-section spreadsheet can be manually typed, filling four columns: Energy, Energy Uncertainty, Cross-section and Cross-section Uncertainty. The energy should be placed in keV units, and the cross-section (total cross section) in millibarns.

►It is rarely useful to manually type the values on built-in spreadsheet editor, since the cross-section values may be filled in by importing them from file sources, using the "Import" button to enable direct file parsing of some file formats.

ERYA allows import and export of R33 (as those in IBANDL), ASCII and Excel's xlsx files, where the last two ones require additional options to make a successful import, due to the requirement to convert units.

►Click "Get Info" to fill additional information about the element, using the text editor bundled for this task. ERYA will copy automatically the "Comment" field when the original data are imported from an IBANDL file.

**Warning:** ERYA will check all physical parameters values written on top window, when the user exports or saves the new element register to the database.

All of them are numerical values, except the Element Name that has its own strict rules: it cannot contain spaces or any character that is not a number or a Latin letter, although an underscore "_" is accepted. ERYA will warn the user to fix the error.

## How to Import from External Files

To import cross-section data from external files, select "Import", then select one of the following modes.

**► R33 Import**

Select an **r33** format file, for example an IBANDL file, and ERYA will parse automatically without additional information.

Notice that if the file contains ambiguous data, or incompatible units, it will deliver an error message, aborting the process.

**► ASCII files import**

Usually these are text files that contain two columns of numeric data, and additional non-numeric lines of text.

When ERYA tries to parse such ASCII files, it will open an additional dialog window asking the original cross-section units (millibarn or barn), or if the cross-section is differential or total. If ERYA fails to parse ASCII files due to the presence of non-numeric data, repeat the import procedure while selecting an additional option on the same import dialog to ignore any lines that are non-numeric data.

Once the parsing of the source file is completed, only numerical data related to the cross-section are placed on the built-in spreadsheet.

**►Excel files import**

Import cross-section data from Excel files is also possible. An additional dialog will open in order to define the original cross-section units in the same manner as for the ASCII case, and to choose the import profile with three different options:

- First option – leaves ERYA to select automatically the import format from the original data. This is the recommended case, unless ERYA fails to decode the original Excel file structure.

- Second option – force ERYA to use the two columns import format, where the energy is placed in the first column, and the cross-section in the second column.

- Third option – force ERYA to use the four columns version, where the energy is place in the first column, and the cross-section in the third column. The second and fourth columns are reserved for the energy and cross section measurement uncertainties.

►To avoid import errors from Excel files, apply these rules:

- Store every relevant information on the first spreadsheet page and group all relevant data on a single compact matrix block, not necessarily starting from the first cell.

- Any rows with non-numerical data, or anything beyond the fourth relevant column are ignored.

## How to Export to External Files

To export the cross-section data from the current Element data, select “Export”, and select one of the following modes.

**► R33 Export**

ERYA can store the cross-section and some elemental physics parameters (including the name and gamma energy) on r33 (IBANDL) file format. The cross-section is given as total in millibarn units.

The comments fields contain the ERYA software version and the six original element references. If re-imported the r33 file again, such fields will be ignored.

Notice that ERYA will assume the nuclear reaction to be an inelastic scattering reaction, requiring manual editing of the exported IBANDL file to fix some inaccuracies.

**► ASCII files export:**

ERYA will export only the cross-section data to a text file with two columns of numeric data, without any additional information. The total cross-section is expressed in barn and energy in keV.

**►Excel files export:**

ERYA will store only the cross-section as a single sheet numeric spreadsheet of data. The first row contains the original row labels of the ERYA’s cross-section editor. When selected this format, the program will ask to choose between a two or four-column file export, but it will always give total cross-section in millibarn units.

## A Simple Import Example

To explain the basic work-flow of creating or editing the Element's Database, we will show how to create a new element using an IBANDL file and make a backup to a file.

**1.** Go to IBANDL website at: https://www-nds.iaea.org/exfor/ibandl.htm

**2.** Select a nucleon, which can be any of the list, but choose a proton projectile (ERYA only supports proton projectiles). It is highly recommendable to filter data types by PIGE only and select the ones with "mb" or "tot" units format.

►Selecting IBANDL files with non-supported units or non-proton projectiles will be rejected by ERYA, warning the user about the import failure cause.

**3.** Save the relevant cross-section files as r33 files. (Click the "Save" button from the web-page.)

**4.** Start ERYA, then open the *Database Manager*, and finally, click "Add".

The Element Editor widget, which should be blank, opens.

**5.** Click on "Import" button to load the file dialog widget. Then select the "IBANDL" file format from the file dialog and pick the correct IBANDL file.

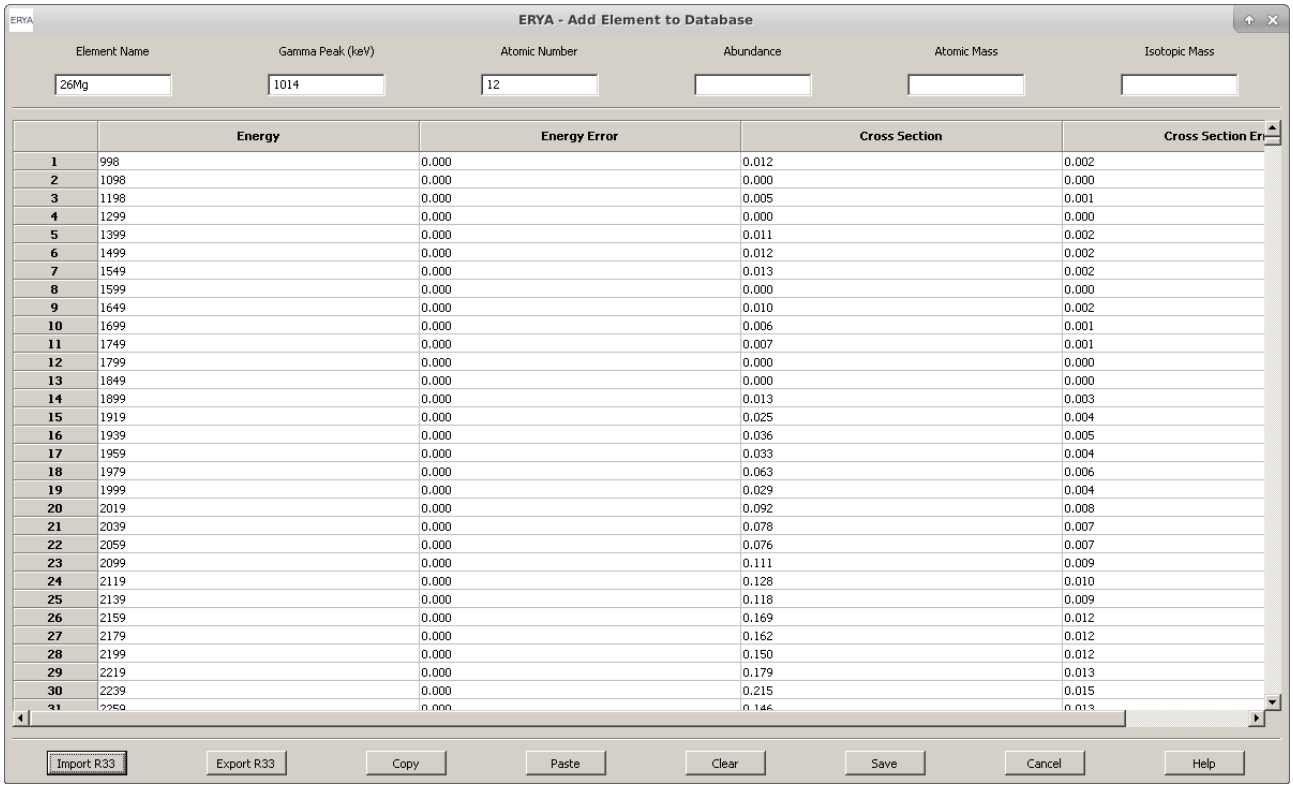
If everything goes right, the cross-section data will be placed inside the built-in spreadsheet of the Element Editor automatically. Usually all necessary unit conversions are made automatically, since the IBANDL file contains all necessary conversion parameters explicitly declared.

Since the IBANDL file contains some additional data, it will also place some of the six Elements parameters on the top window widget.

►Normally, the chemical or isotope symbol and their atomic number will be placed automatically.

►Some IBANDL files may contain more than a value for the gamma peak energy (And are declared by a chain of numbers linked with plus "+" sign). ERYA will only parse the first value.

►On this example, the widget fills the following data, as displayed here:




**6.** Using an isotope table, or an online reference material, fill the remaining gaps, like Abundance or the Isotopic Mass.

**7.** Click "Save", and the new Database entry will be sorted and added.

The new element information will be displayed on the Database Manager.

**8.** Repeat the steps 4 to 7 to add another entry.

**9.** Once returned to the main Database Manager, click on "Save", and select the start-up Database file, if you want to make permanent changes, or choose another name if you want to keep the custom database separated.

**10.** The new elements are now available to use.

**Warning:** If ERYA detects another Database register with the same Name and Gamma Peak, it will asks to accept or abort. In case of acceptance, it will add a number suffix after an underscore (<x>_0, <x>_1, … ).

**How to Export to External Files:**

To export the cross-section data from the current Element data, select “Export”, and select one of those three models:

**► R33 Export:**

ERYA can store the cross-section and some elemental physics parameters (including the name and gamma chosen on the *Element Editor*) on r33 (IBANDL) file format. The cross-section is given as total in milibarn units.

The comments fields contains ERYA software version, and the six original element references. If re-imported the r33 file again, such fields will be ignored.

Notice that ERYA will assume the reaction not change the nuclear species, requiring manual editing of the exported IBANDL file to fix some inaccuracies.

**► ASCII files export:**

ERYA will export only the cross-section data to a text file with two column of numeric data, without any additional information. The total cross-section are converted to barn units automatically and energy to keV.

**►Excel files export:**

ERYA will store only the cross-section as a single sheet numeric spreadsheet of data. The first row contains the original row labels of the ERYA’s cross-section editor. When selected this format, the program will ask to choose between a two or four-column file export, but it will always stored on total cross-section in milibarn units.

# Using ERYA-Profiling's Tools

## ERYA Calculator

This widget is intended to serve as a debug tool for the ERYA macro language, that are available on some options of this program. To open the calculator use the menu: "Tools" > "Calculator"

The widget can also be used as a scientific calculator, if the user wanted to do so. The user can click on the mathematical functions, symbols or numbers keys to avoid manual typing.

To use this tool, it's possible to copy any ERYA macro expression stored on a compatible section, and past on the input line (the top one). Hitting "Return" will display the results on the output line (the bottom one) or any error found during the macro execution.

A full description of ERYA macro language are available on Appendix 1, and the user should run the examples giving with this calculator widget to learn the basics.

In general, the calculator will return a numerical value from the input line once the user click "Return", specially the pure algebraic expressions.

Sequence of commands related to functions that requires arguments as inputs (that during the simulation routines are automatically handled without user input), requires to use the function argument button "f(x=?)" and fill a numerical value on the new input window. Once validated, the result will be displayed on the calculator's output line.

The user can also fill the vector variables defined by *fnvar* parameter (See the Stopping-Power Custom Function chapter) with the "InpVec" button. Then fill on the new input window up to 16 numbers separated by commas (,). This should be done before using the "f(x=?)" button.

Other function keys such as the variable input "InpVar" will ask to insert a valid variable name, and then inserts directly on the input line once confirmed their validity.

To display all memory contents, click "DspVars" and it will writes to the output line (still can be copied) all the contents using the format <var #1> = <value #1> , … , <var #n> = <value #n>.

The user can use "ClrProg" and "ClrAns" to clear the input and output line respectively.

Finally to set the number of decimal places (for bigger numbers, it will switch to the scientific notation), use the function "SetDec" key and select from the list the required number.

Once complete all operations, click "Quit" to close the calculator widget.

# LabView ERYA Import Wizard

**Note:** This feature are deprecated and left undocumented, and it is a leftover of ERYA development to ease the import of LabView binary databases to the new program. To run the wizard, open from the menu: Tools > Import ERYA LabView.

**1.** Before starting the wizard, verify if you have the following five files:

- The original LabView binary Ziegler's Parameters file;

- The original LabView binary Elements Database file;

- A plain text file that should contains two columns of numerical data, for the Detector's Efficiency.

- A plain text file for the Atomic Density (single column) for all elements of the periodic table.

- A plain text file for the Bloch Constants (single column) for all elements of the periodic table.

**Tip:** Using the Database Management tools described on this manual make a text or Excel export of Detector's or Ziegler's Parameters, and make the three text files required by this wizard instead.

**2.** Once started the wizard, and pass the introduction page, the program will ask to select a compatible file from the following components, briefly explained by each wizard's page:

- Element Database;

- Ziegler Parameters;

- Detector Efficiency;

- Element Atomic densities;

- Bloch's Parameters;

**3.** Once inserted all source files, the wizard will displays three additional pages that will asks to give a name for each new database files with the native format handled by ERYA:

- Elements Database;

- Detector Efficiency;

- Stopping Power;

**4.** Once reached the final wizard's page, click "Finish" to start the real conversion job. The whole process may take some seconds to finish, but will display an information dialog once completed the task.

**5.** You should use the Database Management tools to handle the new files. Read the adequate manual chapters about how to edit the files, and fill any additional data.

Normally, it may need to add a custom Detector Efficiency function, and save with the updated information. The user can copy an original function, and paste on the same place after load the new detector file.

To merge the SRIM tables on the new Stopping-Power file, read the *Batch SRIM Import Wizard* section to ease the whole process.

**Attention:** While the wizard enables to import old databases in a single step, the Database Manager and the Stopping-Power Editor can load directly the old binary files if the user select “Legacy (txt)” on “Load” button. However any further saving will always be stored on XML document format.

## SRIM Import Wizard

In order to ease the import of several SRIM Stopping Power tables created by the SRIM® software itself, an auxiliary wizard was created to batch the conversion of several tables at once.

The user need to select pure element stopping powers, where ERYA will only support the incident proton (hydrogen) beams. Once created all necessary files from the SRIM® software, start the *SRIM Import Wizard* itself, and follow the instructions:

**1.** Once passed the introduction page, the first relevant page will display a file manager and associated controls to load several source SRIM® files at once. To start, click “Load”.

► To select more than one file, hold down the Ctrl key, and select multiple files with the mouse.

All selected files will appear on the file manager list. It is also possible to delete one of the list, requiring to select one and click “Remove”. The “Clear” button will clear the file manager list.

**2.** On next wizard page will give the possibility to retrieve an existent Stopping-Power’s database file, or to use the current active Stopping-Power database from ERYA memory, leaving also the possibility to not copy any external data at all.

Notice that only the Ziegler’s Parameters are actually imported by this wizard.

**3.** Finally, the wizard will asks to name the new Stopping-Power database file.

**4.** Once make all main three steps on wizard, the batch conversion should be done in a couple of seconds. When completed the conversion, the new file once opened by the Stopping-Power Editor should display the contents on the “SRIM” tab by the atomic number’s order.

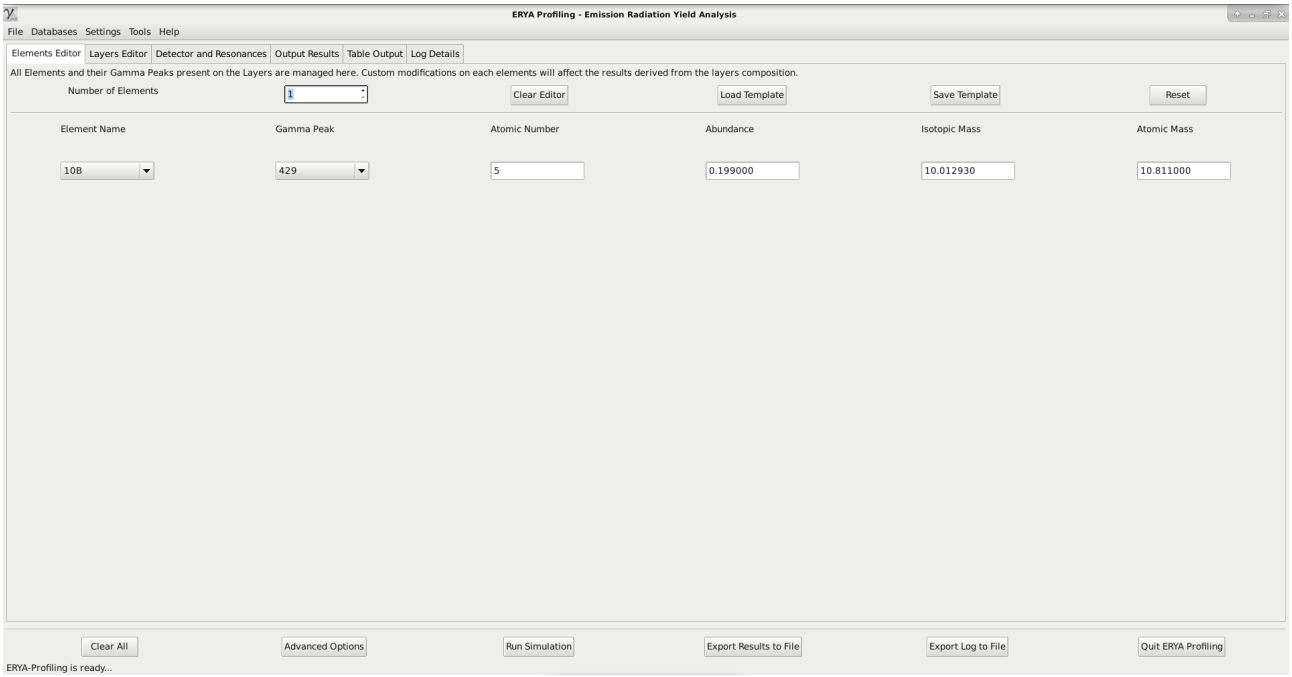
**Note:** If the batch list contains repeated files with the same atomic number, only the last one are stored, while overwrites the previous ones.

**Attention:** Take notice that incorrect SRIM® files from the batch are discarded by the wizard without warning the user.

# How to make a Profiling Analysis

The main purpose of ERYA - Profiling is to make a yield spectrum in function of energy, using the sample previously defined by the user.

If ERYA-Profiling are correctly configured, then all default databases will be loaded automatically on start-up, displaying a full window screen like this example below:




The program uses a tabbed interface, where each tab had own functions and tools that will be described on this chapter.

Database management are handled by the menu interface, as already described on previous chapters of this manual.

The bottom screen contains some buttons dedicated to the simulation and some special configuration settings.

At the bottom of main window, exists a status bar for some internal information, as will be briefly useful during the numerical evaluations.

## ERYA Profiling Quick Guide

**First Step: Define the Number of Elements**

**1.** Select the “Elements Editor” tab, and then select the number of Elements required to the sample.

**2.** Once selected the adequate number of Elements in order to change the Element Name, then:

- Once select an element from the Elements switch widget, it will reload the Gamma Peaks pull-downs that corresponds to the selected element. It will also update the basic information for the selected element from the Database.

- Notice that adding or removing won’t reset the remaining contents of the dynamic table.

**3.** Only select the relevant Gamma Peak of the Element, once finishes all previous element selections.

**4.** It is possible to change the four physical elements parameters, overridden the default values giving by the database.

- This feature can be useful on samples with abnormal isotopic abundances, while also avoid unnecessary Databases editions.

**5.** The current elements and gamma peaks selection will update the tables of other tabs, as it will detail later.

**Extra:** It’s possible to store the custom sample composition on a template file. It can be a normal Excel file, or a native XML based file with extension **epcs**. (ERYA Profiling Compound Sample)

- This is useful to load template sample files without manually typing all over again.

**Second Step: Define the Number of Layers, and Detail their Composition**

**1.** Select the “Layer Editor” tab, and once confirmed the selected elements name and gamma on the new table are correctly assigned, select the number of layers on this step.

**2.** Define the depth of each layer on $10^{15}$ at/cm$^2$ units, on the first column. And for each element column fill the relative atomic mass, taking care to fill the correct values to the corresponding layers. ERYA will renormalize the inputted mass fractions to be equal to one without any additional user intervention.

**Extra:** It’s possible to store the layer contents on an Excel file, or a native XML file with **epls** (ERYA Profiling Layer Sample) extension. This may avoid manual typing of every sample when ERYA starts, when only requires to change some values between a series of simulation, once loaded a layer template from memory.

Notice that loading templates on the “Layer Editor” will reload the “Element Editor” to the default database values, at this current program implementation.

- To avoid a custom data reset from the first tab, the user can write the numerical data directly on the layer built-in spreadsheet editor, or load a text file with the numerical data that should detail the sample.

**Extra Step: Using an Excel File or a ASCII file to Import a Template**

Previous ERYA versions (including the LabView version) could load a text file with a sample composition and import directly, avoiding a manual typing.

On ERYA-Profiling, this feature was greatly improved, where the major improvement was the Excel file import of a template, as it will detail here.

**1. Excel File Option:**

The Excel option had an additional advantage of the text import, since it is possible to use the first row as labels, and define directly the Elements and Gammas on their respective numerical columns.

Select any initial cell of an empty Excel file on the first worksheet, while maintain a compact matrix of data to avoid importation errors.

The first column of the first row will be the depth column label, and fill any work as ERYA will ignore it. Then place on next cells at next right columns of the same row, their names and gammas.

The most usual label format is the form <Symbol> <Gamma>, separated by a space, like "*19F 197"* on a cell to represent the 197 keV gamma of the isotope 19F from the Database. Other valid representations would be *19F (197)* or *19F (197 keV)*.

Then write the correct values for the depth and mass fraction of each element aligned to the correct column label, like the built-in editor of ERYA-Profiling.

Once written all data, save the contents in an Excel Xlsx File Format (version 2007 or greater).

On ERYA, use "Load Table" and select the new created Excel File. It should extract the data and fill the "Element" and "Layer" editor automatically.

It is necessary to take care to not select an non-existent element or gamma, or the program will warn about a non-existent isotope from the current database, aborting the import.

To fix that, the user should open the Excel file again and make the necessary corrections.

It's possible to export a pure numerical table on an Excel file, but require the user to define manually the number and names of all elements previously, or load an additional Excel file for the "Elements Editor" tab forehand.

**2. ASCII File Option:**

To load a text file with the sample composition, define all number of elements and gammas forehand, then load the file.

The sample file should have only numerical data, where each column element are separated by tabs. A mistake to previous elements declaration when load the file will give a failed importation error.

**Third Step: Define the Detector Parameters, and Resonances**

**1.** Select the "Detector and Resonances" tab once completed the previously steps.

**2.** The user can now fill the main Detector physical parameters, and the number of energy samples, by defining a Energy Step along a Minimum and Maximum Energy sequence.

**3.** Optionally is possible to define a Lorentzian Resonance, or even a custom Function Resonance using the ERYA macro language described on Appendix 1.

**3.1** The Lorentzian resonance accept several values on Resonance peak, width and energy, as long that had the same number of elements separated by commas (,), and placed on the same order.

- The optional Maximum and Minimum Energy for the Lorentzian, even they had multiple peaks, are always a single valued input.

**3.2** A custom resonance function should be declared with four keywords, like the following example:

**fxvar = x, fyvar = y, fxmin = a, fxmax = b: a= 1100, b= a + 10, h = 100: y = h * exp(-1*(x-a)^2)**

- Since ERYA needs to know the absolute values for the minimum (fxmin), and maximum (fxmax) values for energy for the resonance function, along the usual fxvar and fyvar keywords for the function variables.

**4.** Select the correct resonance mode using the bottom menu of this tab, where the default setting is to no resonance, or choose the Lorentzian or Custom Resonance function.

**Extra:** It's possible to save the current Detector and Resonance values on a XML file with **eprs** (ERYA Profiling Resonance Settings) extension, that don't affect other tabs data.

**Additional Tip.**

To save all contents of "Element Editor", "Layers Editor" and "Detector and Resonance" tabs, along some additional values at once, use "File" > "Save" and save on a XML file with extension **epgs**. (ERYA Profiling Global Sample). This also saves any custom change on Elements physical parameters.

When load a **epgs** file, with "File" > "Load", it will load all three tabs contents automatically, including all custom Elements physical parameters.

It's up to the user prefer a single sample file, or three separated files to archive all sample data and user inputs.

**Last Step: Make the Simulation**

**1.** Once everything is done, according to the previous steps, hit the “Run Simulation” button.

- If any kind of error are not reported, the simulation will begin while displaying the percentage on the bottom status bar, and also on a progress bar window.

- Since a complex sample may take a long time to complete (the user can change the precision setting on the “Advanced” button, as it will explained latter.), the progress bar will inform the user about the real task in progress.

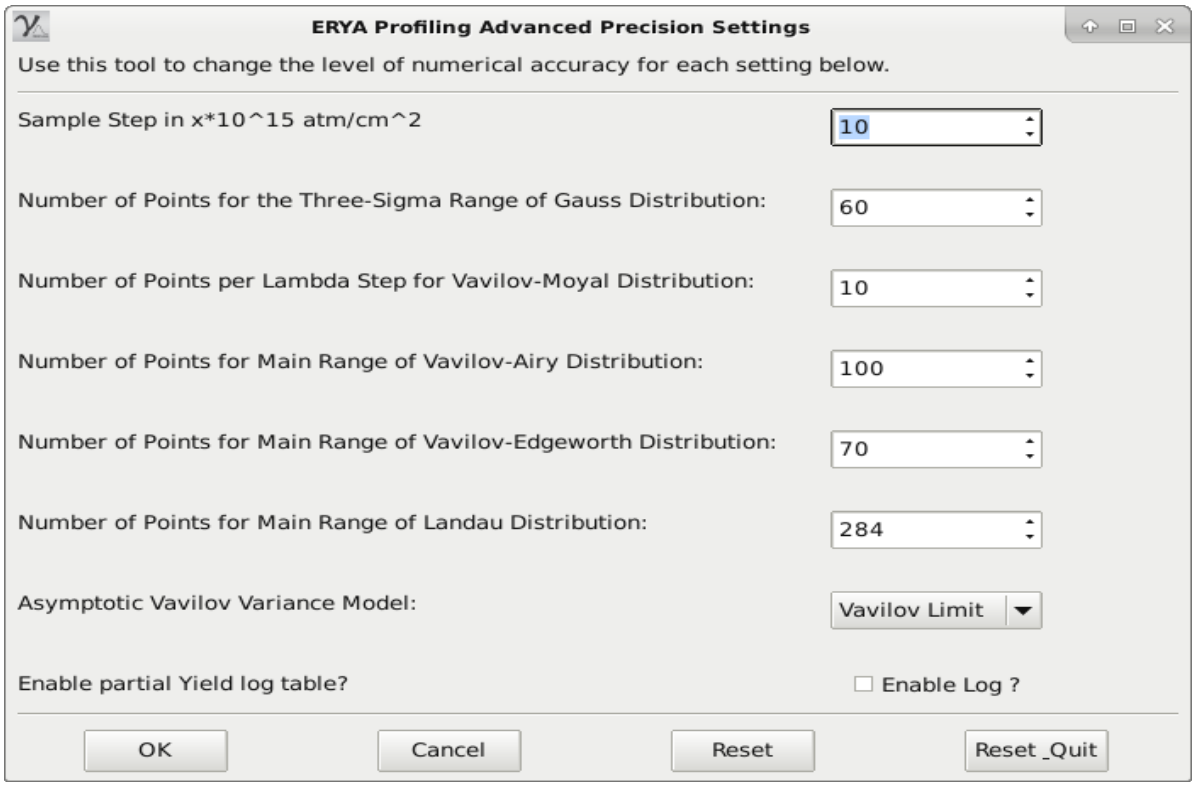
**2.** Once completed, the results are displayed on a graphic plot on “Output Results”, a numerical table of the same plot on “Table Output”, and a log of several intermediate values on the “Log Details” tab.

- Normally, the “Log” table will be empty, as the logging are disabled by default.

- To change some precision parameters, and other features, read the next sections of this manual.

# Changing Accuracy Parameters (Optional)

To change some internal program settings, the user should click the "Advanced Options" button on the bottom main page controls, which opens a simple widget:




On most realistic scenarios, the user only change the Sample Step (Using $10^{15}$ at/cm$^2$ units), where their range are from 1 to 1000 (default is 10). A smaller number on Sample Step may be needed with samples that contains sharp resonances, but it will increase the the simulation time.

Smooth cross-sections can be computed more fast, by choosing a greater Sample Step without losing much accuracy.

The next four parameters are related to the number of points of the intervals of integration related to the distributions. The user can read a brief reference on Appendix 3.

The pull-down option is to select between the Vavilov or Bohr variance formulas for the straggling distribution, applicable when all distributions goes to the Gaussian Limit. An additional pure Gaussian for straggling is also available.

On Profiling samples, the results of both three options are approximately the same, but it will computationally much faster if the "Gaussian Only" option are selected, in some cases between ten to a hundred times faster than the Vavilov distribution.

The last option is a check-box that enables or disables the partial yields logging, which are filled on the “Log” tab. Disabling it can save time and memory, specially when an huge sample ends the simulation and delays the outputs due to the processing of hundreds of thousands lines from memory to the table. This why are disabled by default.

Normally the Log Table are only useful for small samples, and when needed to get the distributions parameters for each step layer to make additional analysis.

►To validate any custom changes, click “OK”. When open this widget again, the new values are automatically displayed on this tool.

►Clicking “Cancel” will close the widget without changing the actual settings running on ERYA.

►To display the default settings, click “Reset” and their values are placed on the widget.

►By selecting “Reset & Quit”, the program reset to the default settings and close the widget in a single action.

# Import Experimental Data

Once the results are displayed on “Output Results” (graphic plot) and “Table Output” (numerical table), ERYA had the feature to the user add experimental data to compare with the simulation results.

► To import experimental data to ERYA it will requires an Excel file with two columns of numerical data (any rows with non-numerical data will be ignored and trimmed out), or an ASCII filled with two tabbed columns of numerical data.

Then click on “Import Experimental Data” and select the correct file. If the import are done correctly, it will make a new plot with the experimental values on “Output Results” and write numerical results on “Table Output”. If the energy column values from the imported file don’t match the simulated energy range from ERYA, the program will make a piecewise interpolation of the original yield column values and displays on the numerical table. The experimental plot output retrieves the original data as is, without any kind of adjustment.

► To add another experimental data, simply repeat the same procedure again.

► ERYA data memory are only deleted if the user click on “Clear All” button.

# Saving Results

ERYA Profiling provides three possible output files, as described below:

► The first option are to save the current graphic plot as an image image, using the “Save As Image” button from the “Output Results” tab page, and then select one of the four images types supported by this program (BMP, GIF, JPEG, TIFF).

If you zoom the image or crop the plot, when save the changed plot to the file, the changed plot will be stored as an image file as expected to be.

Using the “Table Output” and “Log Details” tabs pages buttons, it is possible to save the tables either by an ASCII text file, or an Excel file. Either way, the Excel format is more friendly for further data processing, but with a caveat.

Since the log file of huge samples can creates tables with several thousands of lines, and ERYA-Profiling can take some minutes to save an Excel file with some hundred thousand rows, in this case are preferable to use the ASCII file format.

The ERYA’s own Excel library can crash with such gigantic volume of data, since it was not designed to create enormous files and also ignore the Excel file limitations.

# Useful Tips

- Use the built-in tools described on this tutorial to make everything needed. It’s much better than export from text or Excel files.

- Save all samples as **epgs** files form the “File” > “Save As”, in order to create a collection of master samples. This is useful to run multiple simulations when only a few values are changed, since once the file are loaded by the program avoids to fill everything manually.

Any custom accuracy parameters are also stored on **epgs** file.

- If the elements belonging to a sample have a smooth cross-section, increase the Sample Step, on the “Advanced Options” button. This will reduce the computation time.

- If the simulated sample are bulk, without sharp resonances or not needed much accuracy, select the “Gaussian Only” straggling mode. It will perform the simulation much faster than the Vavilov distribution mode. Combined with a greater Sample Step, the simulation will run an hundred times faster that the default settings.

- On samples with elements that have sharp resonances, reduce the Sample Step in order to avoid loss of accuracy. The main drawback are the increasing simulation time.

- At last resort, run several ERYA-Profiling instances (as much as the number of processors cores your computer have) where each one runs a partial energy range from the same sample (Save the template first, and then it is just needed to change the energy range.).

Once completed all simultaneously simulations, save the results and combine them into a single file.

# A Profiling Example

A good (and long time consuming) example is to simulate the sharp 105 eV resonance of Aluminum-27 at 992 keV.

►To reduce the computation time, start some ERYA-Profiling instances, where each one run a partial energy spectrum. Then run both of them at the same time, since modern computers had several processors on the same chip making possible to run some simulations in parallel.

**1.** Define a sample with pure 27Al, with $3*10^{20}$ at/cm$^2$ (300000 units) of depth, enough to absorb the proton beam at energies around 1 MeV.

**2.** Define the Detector Setup (0.1 keV Detector Resolution, with 0K for the thermal dispersion, and 1 μC of charge) and a Lorentzian Ressonance (1000 mbarn maximum peak, 0.105 keV width at 992 keV).

**3.** With a sharp resonance, reduce the sample step (Advanced) to $10^{15}$ at/cm$^2$ or $5*10^{15}$at/cm$^2$ (1 to 5 units), but maintain a Vavilov distribution for straggling.

**4.** Save the actual settings to a profile file (File > Save As), that will store all inputs.

**5.** Load several ERYA-Profiling (the same number of processors that your computer had) at once, and at each one load the same profile file. For each ERYA-Profiling running, define the initial and final energy of each partial segment with a step of 0.1 keV.

►To run a simulation from 990 to 1006 keV, where each partial simulation spans over 2 keV, it would require 8 simultaneous simulations. Reducing to 1 keV, requires 16. If the the number of total simulations exceeds the number of processors, don't run them at once, instead start a first batch, and wait until finishes to start another ones.

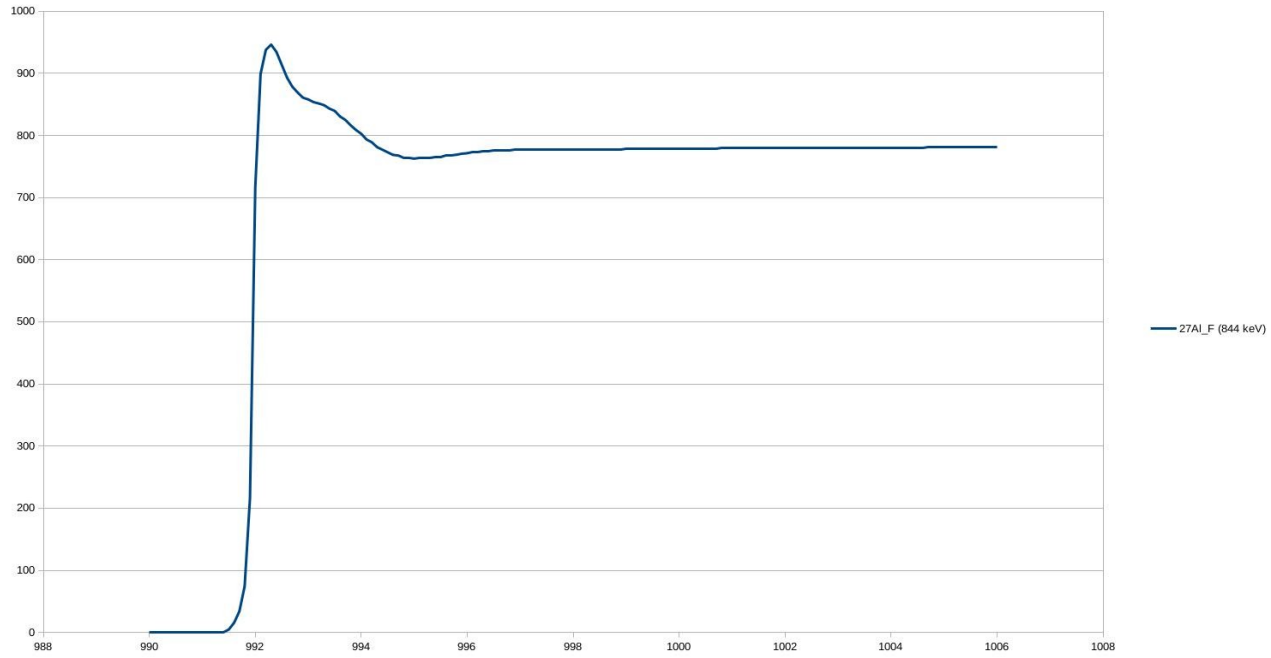
**6.** Once each simulation finishes, save the yields to a file (An Excel file will be fine).

►Once all batched simulations finishes, merge all partial results into a single file.

**Note:** While it would be possible to run the entire profiling simulation in a single instance (for 990 to 1006 keV) the computational time will be huge (several hours).

Currently, ERYA-Profiling don't provide any kind of real multi-threading, and therefore only a full processor will be constantly used by this software. There's some internal discussions about the possibility to create a multi-threaded algorithm for ERYA-Profiling, but until this time none of this was implemented.

Although the results may vary for each simulation setup (different detector’s efficiency, for example), the simulated sharp resonance of 27Al will be similar to the following picture:




The yield profiling displays a fast yield increase until the resonance peak, followed by a fast decay as the energy increases, as expected for a sharp resonance.

More accurate models that uses stochastic simulation, deliver a more realistic simulation for sharp resonances, giving a better treatment of the yield decay post the resonance.

The Vavilov model that predicts the Lewis Peak around the resonance are not fairly accurate compared to the stochastic model, but still are a first approximation.

# Additional Reading

- R. Mateus, A.P. Jesus, B. Braizinha, J. Cruz, J.V. Pinto, J.P. Ribeiro, Nucl. Instr. and Meth. B 190 (2002) 117-121.

- R. Mateus, A.P. Jesus, J.P. Ribeiro, Nucl. Instr. and Meth. B 229 (2005) 302.

- PhD thesis: "Análise de elementos leves por reacções nucleares com produção de radiação gama", Maria Micaela Leal da Fonseca, Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa (2010).

- PhD thesis: "Recolha Automática de aerossóis e a sua análise por técnicas analíticas nucleares", Rodrigo Clemente Velez Mateus, Faculdade de Ciências da Universidade de Lisboa (2003).

- PhD thesis: “Programa para análise em profundidade por técnicas nucleares”, Luís de Souto Martins, Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa (2013).

IAEA-TECDOC-1822, http://www-pub.iaea.org/books/IAEABooks/12235/Development-of-a-Reference-Database-for-Particle-Induced-Gamma-Ray-Emission-PIGE-Spectroscopy

- J. F. Ziegler, J. P. Biersack, U. Littmark, in "The Stopping and Ranges of ions in Solids", Vol. 1, Pergamon Press, New York, 1985.

- Stephen M. Seltzer, Martin J. Berger, in “Energy Loss Straggling of Protons and Mesons: Tabulation of the Vavilov Distribution”, “Studies in Penetration of Charged Particles in Matter (1964)”, The National Academies Press.

- SRIM website: http://www.srim.org/; J.F. Ziegler, Nucl. Instr. and Meth. B219-220 (2004) 1027.

- IBANDL website: http://www-nds.iaea.org/ibandl/

- wxWidgets website: http://www.wxwidgets.org/

- wxWidgets 3.0 reference manual: http://docs.wxwidgets.org/3.0/

- wxWidgets reference book: “Cross-Platform GUI Programming with wxWidgets”, Julian Smart and Kevin Hock with Stefan Csomor, Prentice Hall (2005).

- wxMathPlot add-on: http://wxmathplot.sourceforge.net/

# Appendix 1
# ERYA Macro Programming Language Reference

For some advanced options on ERYA software, it's possible to write small programs (ERYA Macro) to handle custom functions or algebraic expressions that during the numerical simulation routines will deliver numerical values.
This feature resembles the BASIC dialect found on some calculators and it was designed to make single-line programs and tailored to write algebraic functions.
Several characteristics like flow control, code branches, subroutines and other features present on more advanced BASIC interpreters are absent.
This documents will detail the syntax rules and recommend code practices.

**ERYA Macro Code Standards**

An ERYA Macro is composed by a linear sequence of sentences, separated by commas (,) forming a chain function. It is possible to write several chain functions, requiring to use colons (:) to separate them. These conventions are inherited from BASIC language, when multiple sentences are placed on the same line, using the colon symbol to separate them.

ERYA Macro interpreter will consider the last sentence as the *return* sentence of the custom function, and will cache in order to optimize the execution time code.

ERYA Macro will consider two types of sentences:
- **Function Sentences**: Composed by a variable and a number or algebraic expression linked by an assignment sign (=). ERYA will store the numerical value of the variable, according to the numerical evaluation of the algebraic equation.
**Example 1: a = 3 : b = 1**
(Creates two variables on stack memory)
- **Algebraic Sentences:** Are made with a simple numerical expression, that ERYA will evaluate and store on an answer memory variable. Chaining multiple algebraic sentences will overwrite the previous answer variable content.
This means that only the last Algebraic Sentence returned value can be retrieved by the interpreter code to the ERYA numerical routines.
**Example 2: 2/3 :  1 + 4**
(Only the value "5" is stored on the answer memory)

**A Typical User Defined Function using ERYA Macro**

A typical user-defined function, made exclusively by Function Sentences, should follow this template:

**<Function Variables>:<Function Constants >:<Function Expression>**

- The internal interpreter will parse the line starting from the left. You should define several variables or parameters (as explained above) separated by commas (,), and each major section should by separated by colons (:).

- After each section separated by colons (:), the interpreter will flush the stack cache, and store or update the variable stack memory. The last chain of commands are maintained on memory, and renamed as function stack.
- This method was intended to make a single compilation step, creating the function, variables and parameters stack. Once compiled, ERYA numerical routines can return several function values for different arguments, by running the compiled code several times as necessary.

**Get Acquaintance about Parameters, Constants, Variables and Functions**

**1. Parameters**

A *parameter* is a special variable which only accepts alphanumeric words as variables and are intended to make variable declarations.
Each kind of parameter should only be declared a single time, otherwise ERYA will warn the user when a parameter is declared more than one time.
Do declare a *parameter*, just follow the template:

<Parameter Name> = <Parameter Value>

The list of parameters are separated by commas. The supported parameters and respective arguments are the following:

- fxvar = <any valid variable>
Specifies the function independent variable, like x.

- fyvar = <any valid variable>
Specifies the function dependent variable, like y = f(x).

- fxmin = <any valid variable>
Specifies a constant value that should be declared as the absolute minimum value of the function independent variable x.

- fxmax = <any valid variable>
Specifies a constant value that should be declared as the absolute maximum value of the function independent variable x.

- fnvar = <any valid variable>
Defines a vector of 16 elements, where the initial vector element starts with the user-defined variable name. The next vector elements are placed by merging the numbers 1 to 15 as suffixes.

† On both five types, any variable should be an alphanumerical one, but the first character should be a letter.

On ERYA-Bulk program, only the *fxvar* and *fyvar* are obligatory to define a custom function, specially the Detector Efficiency, otherwise it will display an error about an undefined function.
Any custom Stopping-Power function also requires the *fnvar* parameter in order to copy the Ziegler's Parameters table to the vector memories.

The *fxmax* and *fxmin* parameters are optional on ERYA-Bulk program, and serve to define a global limiting function domain at any context.

**2. Constants**

A *constant* is a fixed variable value that the user can declare for best convenience.
A constant name should be an alphanumerical one, but the first character should be a letter.
Once declared, with the assign symbol (=), the user can put any valid arithmetic expression, or a plain numerical value on the right side of the expression:

<Constant Name> = <Constant Value>
<Constant Name> = <Arithmetic Expression>

During the list parsing, the built-in interpreter will create a list of constants and store them into memory. Groups of constants should be separated by commas. When the user defines any constants it is also possible to use an algebraic expression that contains previous defined constants, as long are declared in first place. Notice that constants cannot be defined twice.

**Example 3:** The following example will work, due to the fact the last variable depends from the first ones previously declared.

a0 = 1, a1 = 3, a2 = 4, a3 = a0+a1*a2

**Example 4:** The following example will not work, and will pop-up a warning due to an undeclared constant.

a0 = 1 , a1 = 3 , a2 = 4 , r = a0+a1*a2-a3
(It will warn of the non-existence of "a3")

**Example 5**: It is not possible to define the same variable more than once:

a0 = 1 , a1 = 3 , a2 = 4 , a0 = a0+a1*a2
(It will warn that "a0" value cannot be overwritten)

**3. Variables**

A *variable* is a special kind of constant whose value can be changed during execution. On ERYA, all variables should be declared previously as an argument of a parameter, as described on the respective section.

**Note**: In the program context, the variables are the function variables defined by the "fxvar" and "fyvar" Parameters. Constants are useful to make the macro code more easily readable.

**4. Functions**

On the last code block, the user should define the *algebraic function.*
It is possible to define a chain of functions separated by commas, that can be useful to define composite functions.

**Example 6:** Take the following model-function:

$$\text{Efficiency} = a_3 x^3 + a_2 x^2 + a_1 x + a_0 \text{ , where } x = 1 / \text{Energy}$$

Where the symbolic constants are: a3=-3000; a2=-234.5; a1=10; a0=$1.5*10^{-4}$

- Then implement the following chain of instructions:

**fxvar = E, fyvar = F : a3=-3000, a2=-234.5, a1=10, a0=1.5E-4 : F = a0 + a1/E + a2/E^2 + a3/E^3**

- **Note:** Since the function independent variable can be treated as a composite function respective to the reciprocal of energy, it is possible to write the same function as:

**fxvar = E, fyvar = F : a3=-3000, a2=-234.5, a1=10, a0=1.5E-4 : x=1/E, F = a0 + a1*x + a2*x^2 + a3*x^3**

**Example 7:** A chain of functions can be implemented as follows:
Consider the following analytical formula that models some detectors (A log-polynomial interpolation formula):

$$\ln f(x) = a0 + a1 \ln(x) + a2 \ln^2(x) + a3 \ln^3(x) + a4 \ln^4(x)$$

Where: a0=a1=a2=a3=a4=$3*10^{-4}$.

- Using all ERYA Macro proprieties, it's possible to define the function as the following:

**fxvar = x, fyvar = y : a0 = 3e-4, a1 = a0, a2 = a0, a3 = a0, a4 = a0: z = ln(x), w = a0 + a1*z + a2*z^2 + a3*z^3 + a4*z^4, y = exp(w)**

**List of Admissible Functions and Symbols**

- ERYA Macro's interpreters will recognize a valid block of letters (case-sensitive) and numbers (as long the first character is a letter) as a word, and joined to another vector of similar words, also called tokens. Special symbols are quickly recognized as a token by is own.
Each new token are compared to a table of reserved words (functions or parameters) to further classification. When none of tokens belongs to a reserved word or special symbol, it will become a variable. Failure to follow the naming rules will trigger a syntax error about an incorrect word, or an invalid symbol.

- Numbers should follow the standard format for scientific or float point number according to the C language standard, such as -2.5e+34 to represent $-2.5\times10^{34}$, or decimal numbers ( 22.456) , or even integers ( 23).

- Notice that the interpreter will evaluate negative numbers as one token, which means that ERYA on algebraic evaluations will apply -3^2 as equal 9.
A expression such as 5-3^2 will get -4, since the negative sign are now a minus arithmetic operator.

- ERYA recognizes the BASIC arithmetic operations, like "+", "-", "*", "/". Exponentiation is done by the "^", such as 3^5 to evaluate $3^5$ , following the BASIC standard.

- ERYA support the BASIC relational operators less "<" , more ">", equal less "<=", equal more ">=", equal "==" and different "<>".
Any true relational operation will return "1", and any false statement will get "0".
It is intended to define limited domain functions on the following format:
**Example 8:**

**fxvar = x, fyvar = y : y = (x >= 100) * ( x <= 200 ) * ( x^3+1)**

**Warning**: Do not chain relational operators like 2<5<8, since it will make false positives, due to the conversion of each operation to a single number. ERYA macro language only recognizes double precision numbers as elements, not true boolean variables.
A simple example to display this incorrectness are the expression, 2<5<8, that ERYA will evaluate as 1<8, resulting 1. (Which are coincidentally correct.)
But with the reciprocal, 8>5>2, ERYA will gives 1>2, turning 0.

- The number of left or right parenthesis should be equal, or you will get a parenthesis mismatch error.

- ERYA recognizes several trigonometric functions, both circular and hyperbolic, as well as their inverses:
sin, cos, tan, sinh, cosh, tanh, asin, acos, atan, asinh, acosh, atanh.

- ERYA also recognizes the following exponential and logarithmic functions:
sqrt (square root), ln (natural logarithm), log (decimal logarithm) and exp (natural exponential).

- Any function argument should at last be placed under a pair of parenthesis, like sqrt(4) .

**Note:** To make the macro more readable, the user can separate the numbers, operators, functions and variables with at least one empty space.

**Some Technical Information about the ERYA Macro Interpreter**

The built-in interpreter uses a custom made parser that reads several groups of characters and groups to form a dynamic vector of strings, called tokens.
The token list are also classified by their associativity and order of precedence, as indicated by the following table below:

| **Token Type** | **Priority Order** | **Associativity** |
|---|---|---|
| Variables and Symbols “:”, “,”, “(“, “)” | 0 | Left |
| Assignment Operator “=” | 1 | Left |
| All relational operators | 2 | Left |
| Sum “+”, Minus “-” | 3 | Left |
| Product “*”, Division “/” | 4 | Left |
| Power “^” | 5 | Right |
| Functions | 6 | Left |

Automatically the tokens list are subject to a special algorithm derived from the Shunting-Yard Algorithm in order to convert the original expression to the RPN format (Reversed Polish Notation), also called postfix notation. This process also creates a vector of variables, parameters and constants.

The algebraic function numerical evaluation in postfix notation are a trivial task, since the order of operations are placed linearly without any branches.

# Appendix 2
# Table of Excitation Functions included in the Default Database.

| Element/ Isotope | Nuclear Reaction | Gamma-ray (keV) | Angle * | Excitation functions |
|---|---|---|---|---|
| Lithium | $^{7}Li(p,p_1)^{7}Li$ | | | |
| $^{7}Li$ | | 478 | 130º | R.Mateus et al. [1]: 652 – 2349 keV |
| | | | 130º | M. Fonseca et al. [2]: 2398 – 4175 keV |
| Beryllium | $^{9}Be(p,p_1)^{10}B$ | | | |
| $^{9}Be$ | | 718 | 130º | M. Fonseca et al.: Unpublished results. |
| Boron | | | | |
| $^{10}B$ | $^{10}B(p,\ )^{7}Be$ | 429 | 90º | R.B. Day et al. [3]: 757 – 2007 keV |
| | | 429 | 90º | A. Lagoyannis et al. [4]: 2020 – 5001 keV |
| | $^{10}B(p,p_1)^{10}B$ | 718 | 90º | R.B. Day et al. [3]: 1520 – 2534 keV |
| $^{11}B$ | | 718 | 90º | A. Lagoyannis et al. [4]: 2542 – 5002 keV |
| | $^{11}B(p,p_1)^{11}B$ | 2125 | 90º | K. Preketes-Sigalas et al [5]: 2580 – 5000 keV |
| Fluorine | | | | |
| $^{19}F$ | $^{19}F(p,p_1)^{19}F$ | 110 | 130º | A. P. Jesus et al [6]: 821 – 2139 keV |
| | | 110 | 130º | A. P. Jesus et al [7]: 2144 – 4059 keV |
| | $^{19}F(p,p_2)^{19}F$ | 197 | 130º | A. P. Jesus et al [6]: 821 – 2160 keV |
| | | 197 | 130º | A. P. Jesus et al [7]: 2174 – 4059 keV |
| Sodium | | | | |
| $^{23}Na$ | $^{23}Na(p,p_1)^{23}Na$ | 440 | 130º | R.Mateus et al. [8]: 1260 – 2410 keV |
| | | 440 | 130º | M. Chiari et al. [9]: 2410 – 4060 keV |
| | $^{23}Na(p,p_2)^{23}Na+$ $^{23}Na(p,\ )^{20}Ne$ | 1636+1634 | 130º | M. Chiari et al. [9]: 2410 – 4060 keV |
| Magnesium | | | | |
| $^{24}Mg$ | $^{24}Mg(p,p_1)^{24}Mg$ | 1369 | 90º | N. Sharifzadeh et al. [10]: 998 – 2999 keV |
| | | | 135º | I. Zamboni et al. [11]: 3007 – 3053 keV |
| | $^{25}Mg(p,p)^{25}Mg$ | 390 | 90º | N. Sharifzadeh et al. [10]: 998 – 2449 keV |
| $^{25}Mg$ | | 390 | 90º | K. Preketes-Sigalas et al [12]: 2456 – 4546 ke |
| | | 585 | 130º | A. P. Jesus et al [7]: 869 – 4017 keV |
| | | 975 | 90º | N. Sharifzadeh et al. [10]: 1899 – 2459 keV |
| | | 975 | 90º | K. Preketes-Sigalas et al [12]: 2476 – 4546 ke |
| Aluminum | | | | |
| $^{27}Al$ | $^{27}Al(p,p_1)^{27}Al$ | 844 | 130º | A. P. Jesus et al [7]: 1497 – 3000 keV |
| | | 844 | 90º | M. Chiari et al. [13]: 3007 – 4089 keV |
| | $^{27}Al(p,p_2)^{23}Al$ | 1014 | 130º | A. P. Jesus et al [7]: 1497 – 3000 keV |
| | | 1014 | 90º | M. Chiari et al. [13]: 3007 – 4089 keV |
| Silicon | | | | |
| $^{28}Si$ | $^{28}Si(p,p_1)^{28}Si$ | 1779 | 90º | A. Jokar et al. [14]: 2098 – 3047 keV |
| | | 1779 | 55º | L. Csedreki et al. [15]: 3052 – 4015 keV |
| $^{29}Si$ | $^{29}Si(p,p_1)^{29}Si$ | 1273 | 90º | A. Jokar et al. [14]: 2098 – 3047 keV |
| | | 1273 | 55º | L. Csedreki et al. [15]: 3052 – 4015 keV |

The excitation functions shown above are, with the exception of $^{9}$Be, available in IBANDL. As indicated in the table some merges between two excitation functions were made to obtain wider energy range cross-sections.

* Angle used for the measurement of cross sections; the gamma-ray lines indicated in the table are either isotropic or have a small anisotropy, except for $^{28}$Si 1779 keV line which above 3 MeV show a strong anisotropy.

References:

[1] R.Mateus et al. Nucl. Instr. & Meth. B 190 (2002) 117

[2] M. Fonseca et al., IAEA-TECDOC-1822, Vienna (2017)

[3] R.B. Day et al., J. Phys. Rev. 95 (1957) 1003

[4] A. Lagoyannis et al., Nucl. Instr. & Meth. B 342 (2015) 271-276

[5] K. Preketes-Sigalas et al., Nucl. Instr. & Meth. B 368 (2016) 71

[6] A. P. Jesus et al, Nucl. Instr. & Meth. B 161-163 (2000) 186

[7] A. P. Jesus et al, IAEA-TECDOC-1822, Vienna (2017)

[8] R.Mateus et al. Nucl. Instr. & Meth. B 219-220 (2004) 307

[9] M. Chiari et al., Nucl. Instr. & Meth. B 441 (2019) 108-118

[10] N. Sharifzadeh et al., Nucl. Instr. & Meth. B 372 (2016) 109

[11] I. Zamboni et al., Nucl. Instr. & Meth. B 342 (2015) 266-270

[12] K. Preketes-Sigalas et al, Nucl. Instr. & Meth. B 386 (2016) 4

[13] M. Chiari et al., Nucl. Instr. & Meth. B 332 (2014) 355

[14] A. Jokar et al., Nucl. Instr. & Meth B 371 (2016) 37

[15] L. Csedreki et al., Nucl. Instr. & Meth. B 443 (2019) 48-56

# Appendix 3
# The Vavilov Distribution on ERYA-Profiling

ERYA-Profiling use a numerical approximation of Vavilov distribution described on [1] and [2], where it will display a very brief description before reading the cited bibliography.

Taking the interaction between the incident beams and the matter as collisions, it is possible to define a maximum transferable energy of a collision $\epsilon_{MAX}$, and from Relativistic Kinematics to evaluate the relativistic velocity $\beta$, and Lorentz factor $\gamma$, resulting the following Landau parameters:

$$k = \xi / \epsilon_{MAX} \tag{1}$$

$$\xi = \frac{2.5507 * 10^{-7} Z M n}{A \beta^2} [keV] \tag{2}$$

$$\epsilon_{MAX} = \frac{2 m_e \beta^2 \gamma^2}{1 + 2\gamma \frac{m_e}{m_p}} [keV] \tag{3}$$

Which can be evaluated by each individual layers in the sense of the model applied by ERYA, since the **k** and **ξ** can be summed along each individual layer of the sample.

The major physical parameters are total electric charge **Z**, molar mass **M**, and isotopic mass **A**, along the depth of the layer **n** in $10^{15}$ at/cm$^2$ units).

With this definitions, it's now possible to define the Landau Variable **λ:**

$$\lambda = \frac{\Delta}{\xi} - 1 + \gamma - \beta^2 - \ln(k) \tag{4}$$

On (4), the energy gap $\Delta$ are defined by the difference between the average distribution energy $\overline{\Delta E}$ that is also the distribution argument, and the actual energy loss $\Delta E$, obtaining:

$$\Delta = \overline{\Delta E} - \Delta E \tag{5}$$

Since the energy loss are always a positive value, this means the following expression is positive:

$$\Delta E = \underbrace{\xi \left( \lambda + 1 - \gamma + \beta^2 + \ln(k) \right)}_{\Delta} + \overline{\Delta E} > 0 \tag{6}$$

Retrieving from the theoretical introduction on this manual, the energy beam and energy loss and related to the following formula:

$$E = \overline{E_k} - \Delta E = \overline{E_k} - \xi \left( \lambda + 1 - \gamma + \beta^2 + \ln(k) \right) - \overline{\Delta E} \tag{7}$$

As briefly explained on introduction the beam distribution are a Gaussian shaped distribution with average equal to zero, but with a standard deviation given by:

$$\sigma_T = \sqrt{\Omega_{rel}^2 + \Omega_{th}^2} [keV] \tag{8}$$

Where the thermal resolution of the sample are defined by the following formula, and user defined temperature T, and beam resolution:

$$\Omega_{th} = 2.355 \sqrt{\frac{2 m_H k T E_0}{M}} [keV] \tag{9}$$

The Straggling Distribution are modeled by the Vavilov Distribution which depends from the Landau variable formula on (4) and (6), where the energy average loss are related to the stopping-power and layer depth:

$$\overline{\Delta E} = \epsilon\left(E_0\right) n\left[keV\right] \tag{10}$$

The Vavilov Distribution don't have a finite average and standard deviation. Such concepts are applied when the Vavilov distribution tends to the Gaussian Distribution, with an average of:

$$\mu = \overline{\Delta E} \tag{11}$$

And the standard deviation defined by Vavilov itself [1]:

$$\sigma = \sqrt{\frac{\xi^2\left(1-\beta^2/2\right)}{k}} \tag{12}$$

The user can change the Vavilov formula (12) on "Advanced" by the H.Bichsel formula [1]:

$$\sigma = \sqrt{\frac{8\xi}{3n}\sum_{i=1}^{N} S_i I_i \ln\left(2 m_e \beta^2 / I_i\right)} \tag{13}$$

Where it takes account the mean ionization **I** of all elements of each layer, and their stoichiometries **S**, to evaluate the desired formula (13). Negative values for the logarithm are cut off from the sum. Selecting "Gaussian Only" will apply a Gaussian with (11) and (12) as parameters.

The Vavilov Distribution on ERYA are composed by a sequence of functions that depends from **k**:

| Landau k-factor | Approximate Distribution |
|---|---|
| **k = 0** | **Dirac's Function** |
| **0 < k < 0.02** | **Landau Function** |
| **0.02 < k < 0.24** | **Vavilov-Moyal Function** |
| **0.24 < k < 22** | **Vavilov-Airy Function** |
| **22 < k <25** | **Vavilov-Edgeworth Function** |
| **k > 25** | **Gaussian Function** |

The user on "Advanced Settings" widget can change the number of integration steps along the distribution's integration domain. Since all distributions are unbounded, the numerical integration spans over a 3-sigma cut-off domain (99.5 % confidence level).

If the user selects a pure Gaussian distribution for straggling, only the user's custom setting will be applied. The same happens to the beam resolution distribution that are always Gaussian.

The degenerate Dirac's Distribution are always a single point, and obliviously absent to the user change.

All mathematical details of any of those functions are related to [1] and [2], for further reading.

References:

[1] "Programa para análise em profundidade por técnicas nucleares", Luís de Souto Martins, Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa (2013).

[2] "Energy Loss Straggling of Protons and Mesons: Tabulation of the Vavilov Distribution", Stephen M. Seltzer, Martin J. Berger, The National Academies Press (1964).