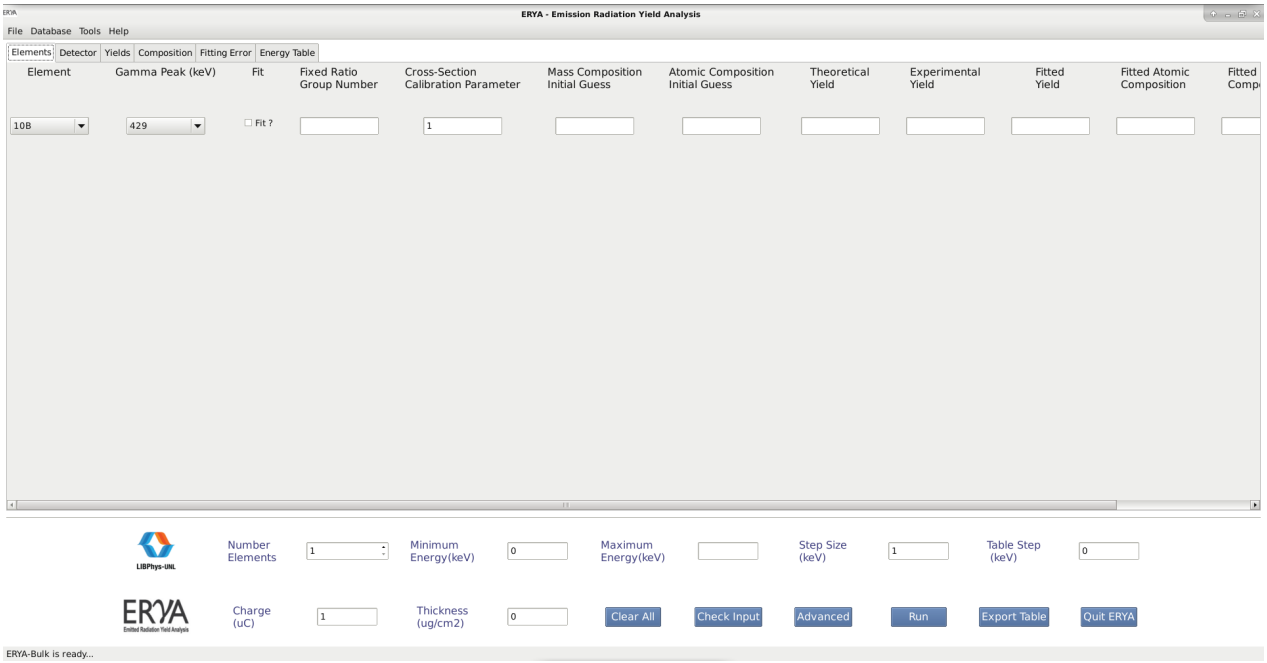
ERYA Bulk

User’s Manual



Manteigas Fonseca Pedro de Jesus

Version 5.00

ERYA Bulk USER’S MANUAL

SOFTWARE VERSION 5.00

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

Table of Contents

Introducing ERYA Bulk

Welcome!

ERYA (Emitted Radiation Yield Analysis) Bulk is a software application designed to aid the PIGE analysis of bulk samples (i.e. in-depth homogeneous samples), when the main task is to determine their composition in terms of their elements and/or isotopes quantities.

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

The program evaluates the composition of a homogeneous 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 Δx 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:

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

being N_{av} the Avogrado number and f_i, and A the isotopic abundance and the atomic mass of the relevant element, respectively.

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

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

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

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

In order to get the mass fraction from a given yield, it is required to know the major composition of the target which affects the stopping powers. From a first guess of this composition, ERYA provides the tools to optimize the composition in order to fit the calculated yields to the experimental ones.

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-Bulk?

This application is a full rewrite of a previous LabView program that handles the PIGE measurements of a bulk homogeneous sample (hence the ERYA-Bulk program name).

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 Bulk 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.

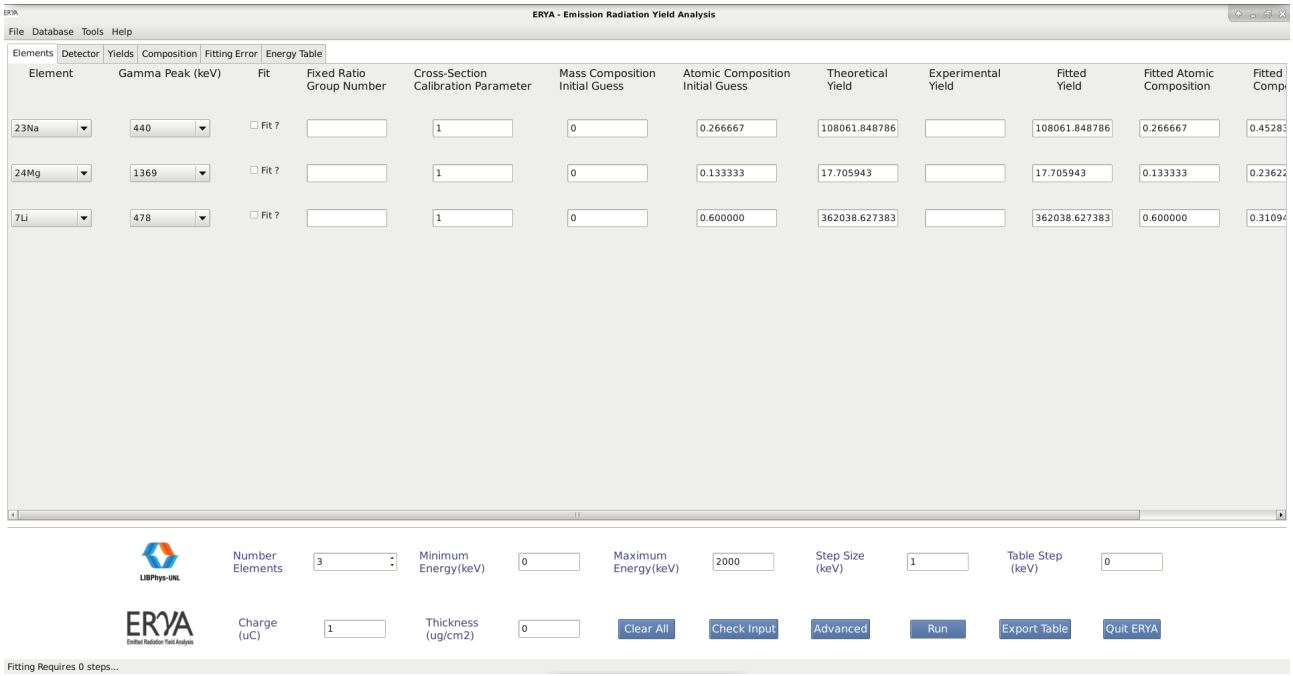
A special build for 32 and 64-bit ARM machines are also available for Linux computers, and are dedicated for ARM computers like the Raspberry 3 models.

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-Bulk 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 ERYA-Bulk main window interface:



Some features of ERYA Bulk are:

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

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

3. ERYA-Bulk can adjust the simulated yields to the experimental yields, using the Levenberg-Marquardt algorithm. The program has an optional utility to tune the fitting algorithm, in terms of numerical precision and number of steps. Several elemental yields can be fitted simultaneously.

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-Bulk.app package to the Applications folder. Finally, click on ERYA-Bulk 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-Bulk". 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-Bulk-Win64** (64-bit)

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

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

$ **sudo apt remove eryabulk**

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

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-Bulk-OSX/**

Running ERYA for the First Time

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

2. When ERYA-Bulk 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-Bulk 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-Bulk software locates. For each database the user should click the “ERYA-Bulk” 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-Bulk” folder to access the database files.

- Mac OS X will point the file dialog inside the hidden “Contents” folder of ERYA-Bulk-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-Bulk-Win64***	**/home/name/.ERYA-Bulk/***	**/Users/name/Library/Application Support/ERYA-Bulk-OSX/***
Config Name	**ERYA-Bulk-Win64.conf**	**ERYA-Bulk.conf**	**ERYA-Bulk-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-Bulk is now ready to use.

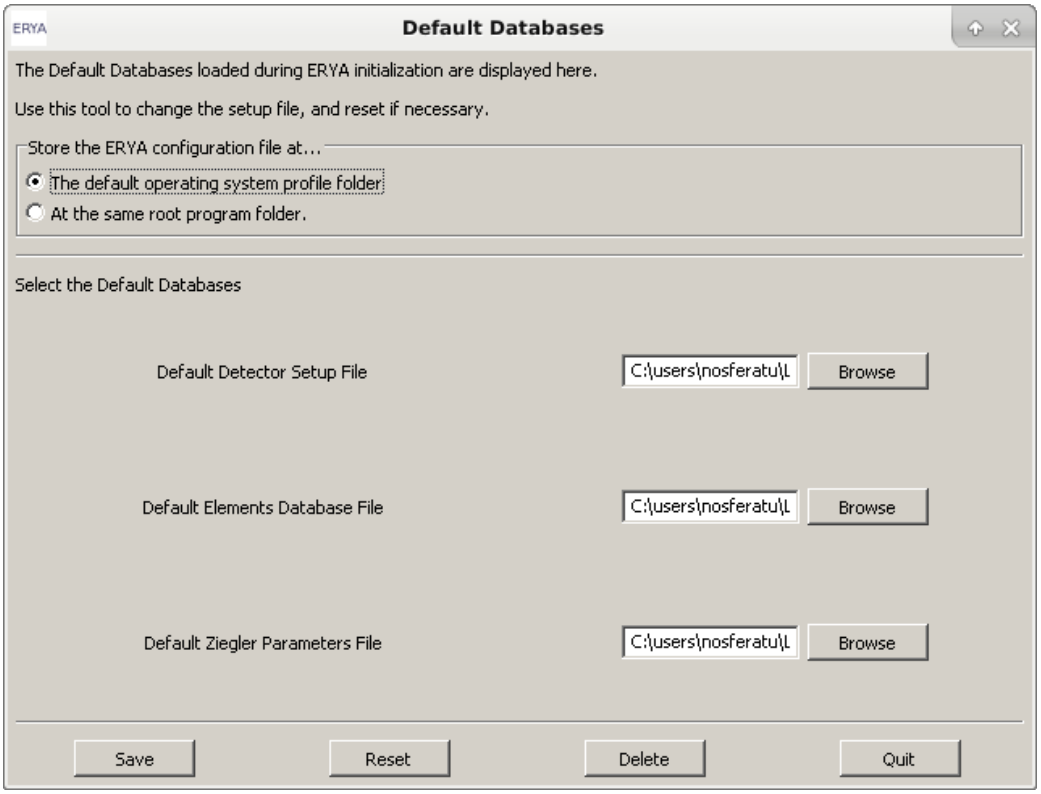
Warning: In case of any errors related to the databases or configuration files, read the *Troubleshooting* guide to deal with them.

► 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. Alternatively, use the menu tool at Tools > Reset ERYA Settings. 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.

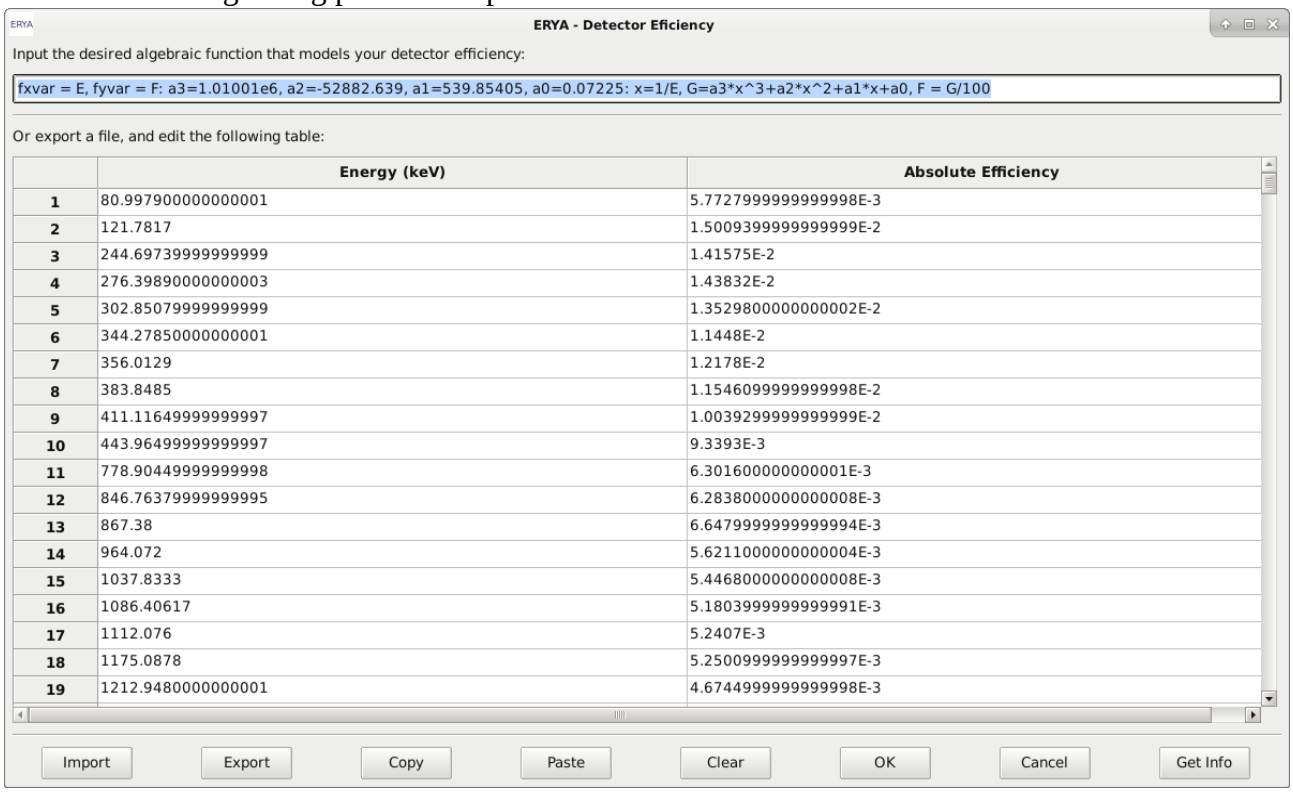
►Manual resizing of dialog windows may be needed when ERYA runs on computers with 4K (3840*2160) monitors, or higher.

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.

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.

Tip: The user can test any ERYA macro code using the internal interpreter by opening the ERYA Macro Calculator available on Tools menu. This widget accept any paste from any valid macro code from a compatible source, and execute directly for testing purposes.

To evaluate any Macro expressions, fill the input box or paste from the source, and hit *Return* .

Any arithmetic expression (means no function keywords) should appear on the output box, which is not editable.

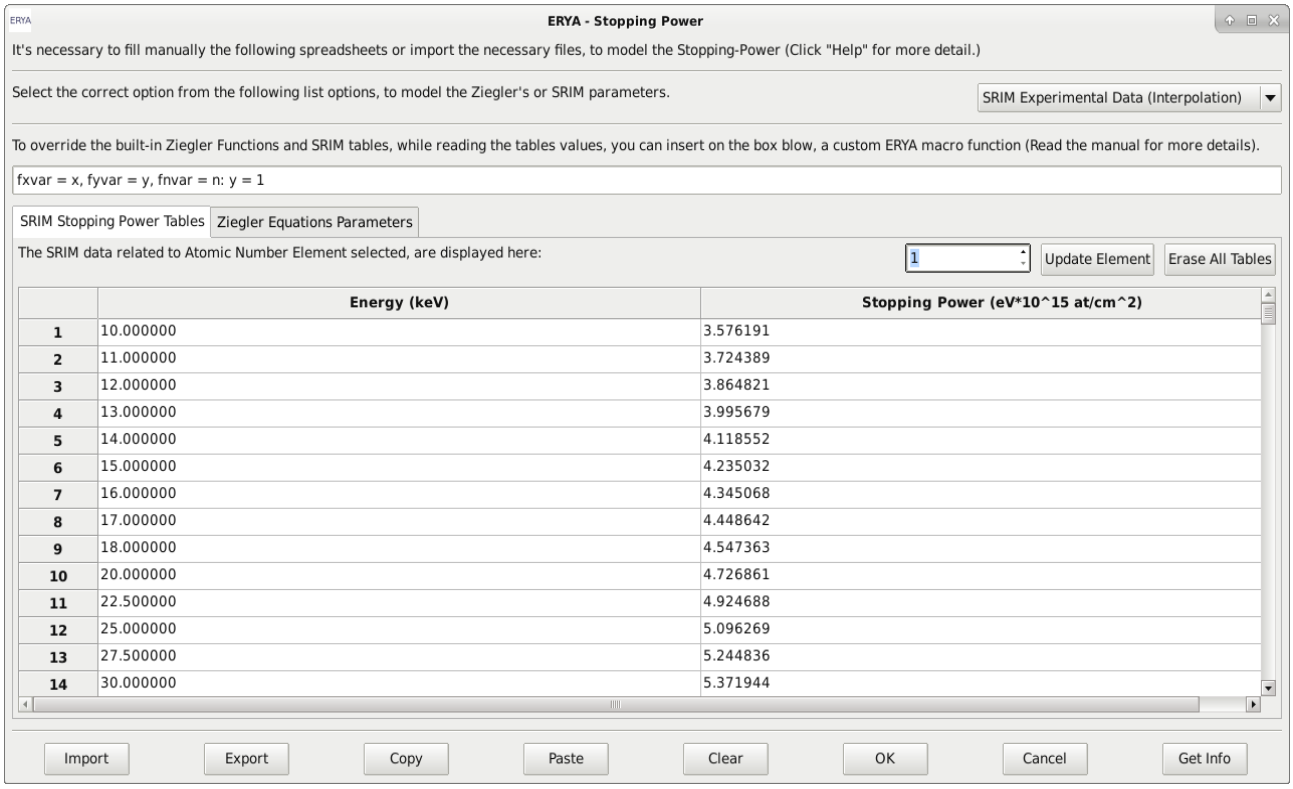
Notice that a function requires an argument, and requires to click on *f(x=?)* button and fill the function argument requested by an additional dialog. The function image will be placed on the output box.

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 variables with *14* and *15* suffix are not used by ERYA-Bulk.

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:

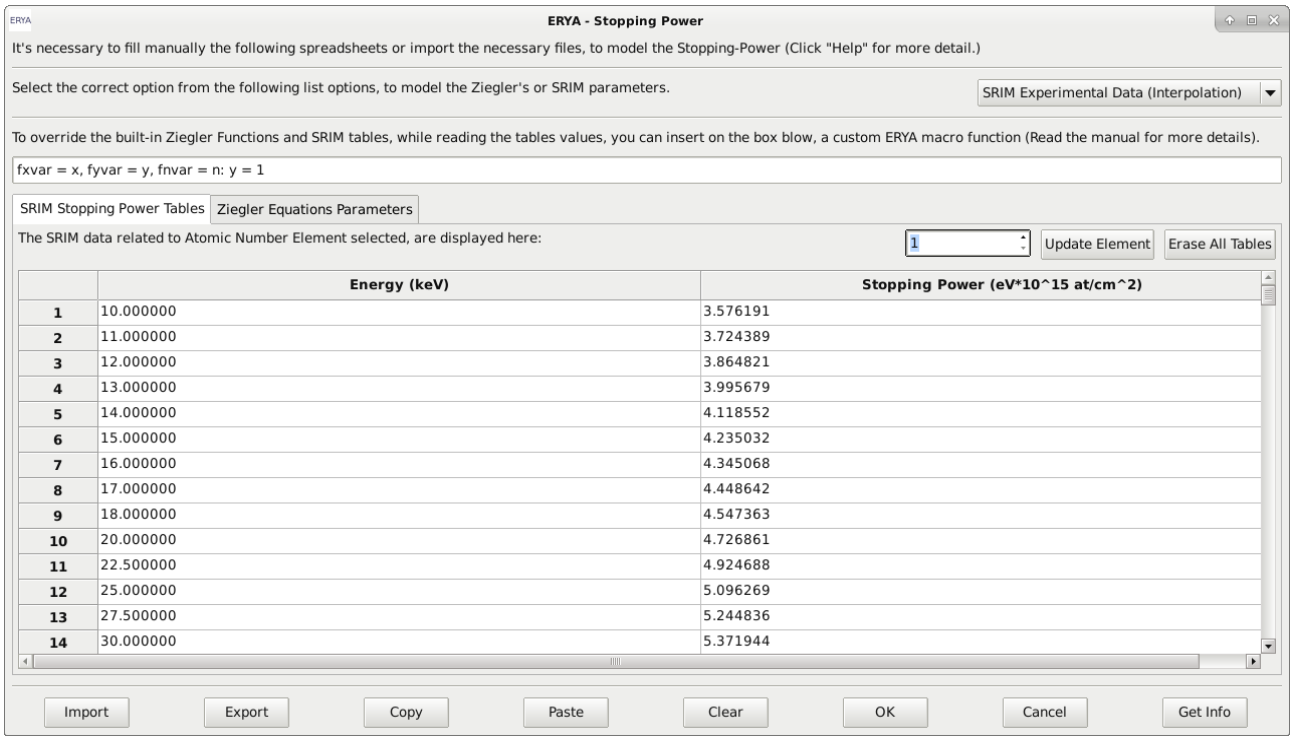
fxvar = x, fyvar = y, fnvar = zn : y = 1.77

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.

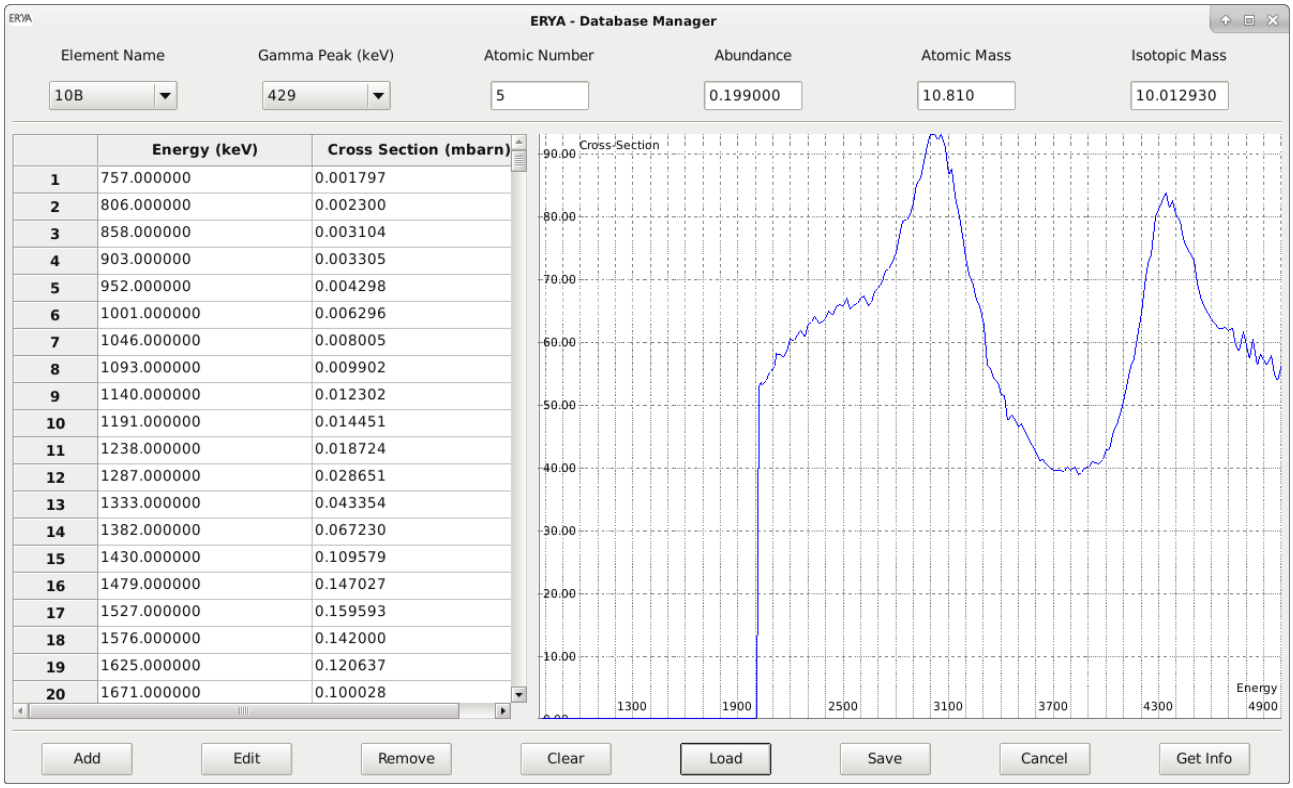
►**When the user imports a raw numerical data table**, for example a new Ziegler's Parameters table or a stopping power interpolation data, as ASCII, the user should select the Ziegler's tab or the correct atomic numbers on the SRIM tab **before importing the file.** An error message or an erroneous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

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.

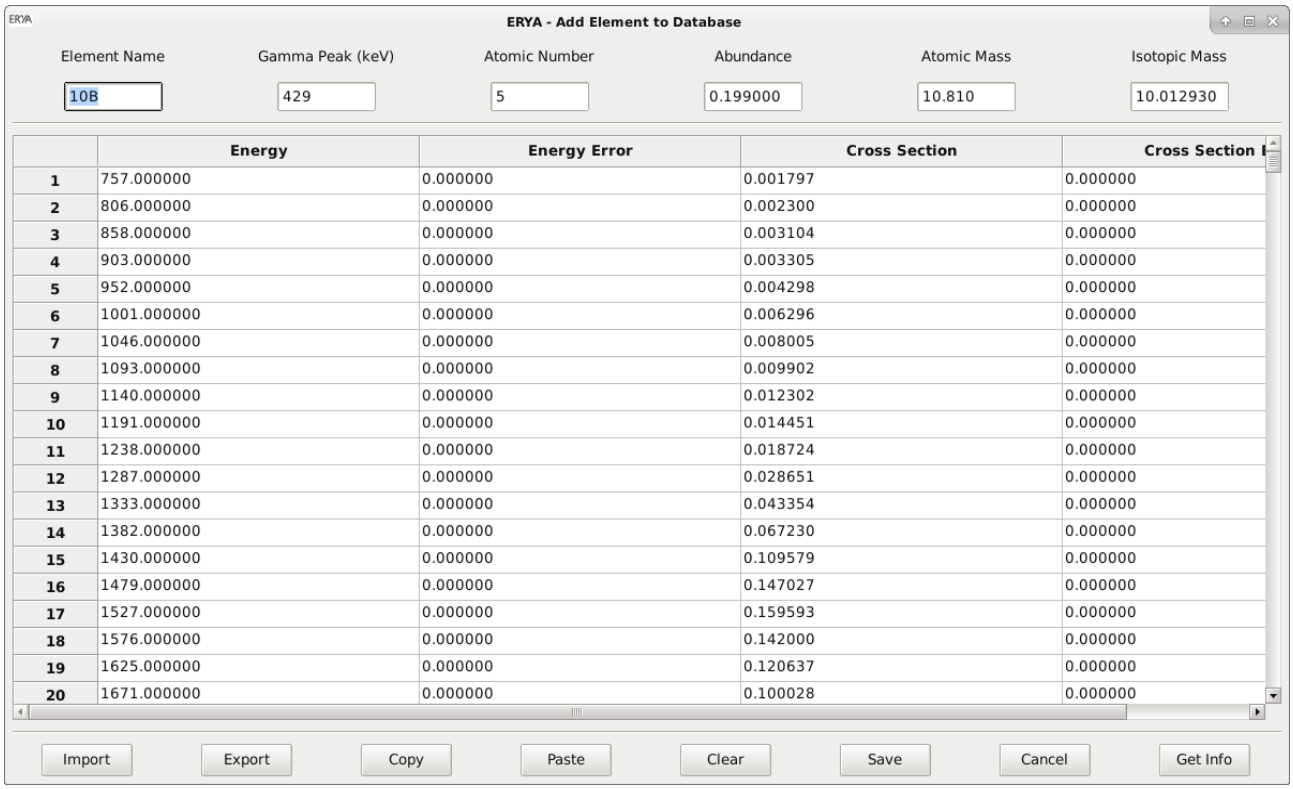
For each Element/Gamma pair, a read-only Energy/Cross-Section table and graphical plot will be drawn. Up to four element physical proprieties will also be displayed.

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.

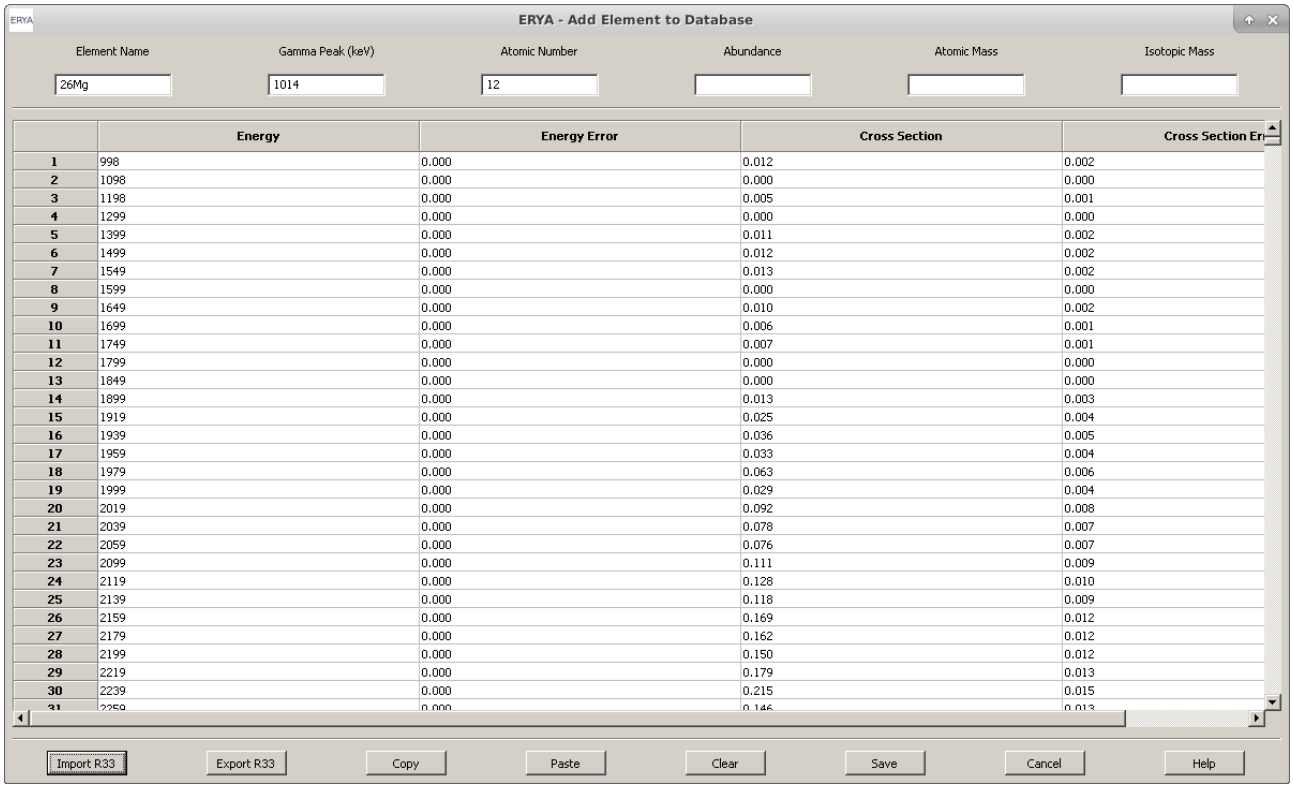
If everything goes right, the cross-section data will be placed inside the built-in spreadsheet of the Element Editor automatically. Usually all necessary unit conversions are made automatically, since the IBANDL file contains all necessary conversion parameters explicitly declared.

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:



	Energy	Energy Error	Cross Section	Cross Section Er
1	998	0.000	0.012	0.002
2	1098	0.000	0.000	0.000
3	1198	0.000	0.005	0.001
4	1299	0.000	0.000	0.000
5	1399	0.000	0.011	0.002
6	1499	0.000	0.012	0.002
7	1549	0.000	0.013	0.002
8	1599	0.000	0.000	0.000
9	1649	0.000	0.010	0.002
10	1699	0.000	0.006	0.001
11	1749	0.000	0.007	0.001
12	1799	0.000	0.000	0.000
13	1849	0.000	0.000	0.000
14	1899	0.000	0.013	0.003
15	1919	0.000	0.025	0.004
16	1939	0.000	0.036	0.005
17	1959	0.000	0.033	0.004
18	1979	0.000	0.063	0.006
19	1999	0.000	0.029	0.004
20	2019	0.000	0.092	0.008
21	2039	0.000	0.078	0.007
22	2059	0.000	0.076	0.007
23	2099	0.000	0.111	0.009
24	2119	0.000	0.128	0.010
25	2139	0.000	0.118	0.009
26	2159	0.000	0.169	0.012
27	2179	0.000	0.162	0.012
28	2199	0.000	0.150	0.012
29	2219	0.000	0.179	0.013
30	2239	0.000	0.215	0.015
31	2259	0.000	0.146	0.013

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.

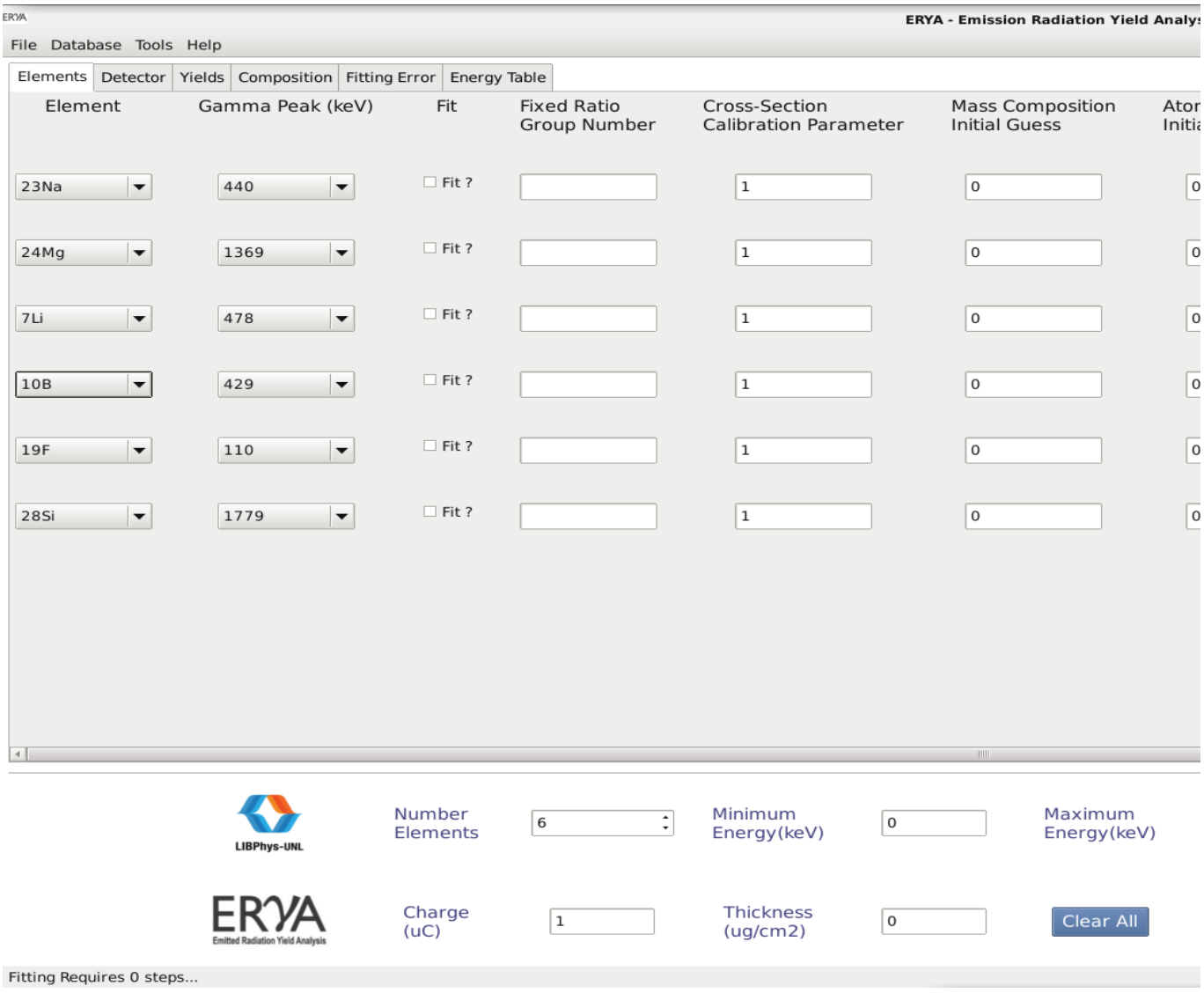
10. The new elements are now available to use.

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 make a PIGE Analysis

The main purpose of ERYA is to make an analysis of an homogeneous sample, in order to determine its composition in terms of its elemental mass concentrations, providing the possibility to fit the experimental values to the simulated ones, for several elements simultaneously.

In the last chapter we described the databases of the input data. Now we will address the additional steps to proceed to the calculation.



How to define your Sample

The user should fill all information related to the sample composition using the tools provided by the main screen interface, following the steps below.

1. Select the number of elements (first entry of the bottom bar), which will lead to a corresponding number of table lines.
2. In the first column of this table select the element names from the pull-down menus.
3. In the second column select the corresponding gamma-ray line.
4. If the sample contains a well-defined chemical compound, then a unique positive integer should be assigned on the fourth column, for all elements belonging to such a compound.

   Example: If you have a Li_2WO_4 compound mixed to the sample, assign the three elements (Li,W,O) to a Number Group "1" or other number of your choice.

Note: Unsigned elements will have a Number Group equal to zero by default, and ERYA treats them as independent elements.

5. The Atomic and Massic Composition columns may be filled with numeric values or algebraic expressions that the ERYA macro interpreter will evaluate to numerical values automatically. The following examples are valid:

   1 – 0.2 for **[0.8]**

   1/5 for **[0.2]**

   1 – (1 / 2 + 1 / 3) for **[0.166667]**

   It is also possible to use symbolic variables, as long the final arithmetic expression is the last block of code after the rightmost colon (:)

   a = 1, b = 5: 1 – a / b for **[0.8]**

   Any column entry empty on those two columns will be perceived by ERYA as the default "1" value for atomic, and default "0" for massic.

   This dual column input are intended to the user define the sample composition either by atomic fractions, or mass fractions. Both modes had some rules to be followed:

   Atomic Mode: The user can only fill the atomic column and clean entirely the massic column, which ERYA will interpret that the sample was measured by the atomic composition, which is the oldest standard on similar software.

   ERYA will apply a renormalization of the column contents in order to ensure that the sum of all atomic fractions are equal to "1", without additional user inputs.

Massic Mode: To enable this mode, the user should fill the Atomic column and avoid to fill any value on the Atomic column. The renormalization rules from the Atomic Mode are also applied.

Compound Mode: A third mode are activated when the user fill values for the massic and atomic columns within certain rules, related to the values of the Compound column (Step 4).

This input mode are useful for a sample made for a mixture of isolated elements, along compounds with a known chemical stoichiometry and global mass fraction. With this feature ERYA will make the conversion of everything to atomic fractions automatically.

For a compound with two or more elements (sharing the same Compound number), it requires the user to fill their mass (or fraction of it) at one of the Massic Column cell. Then fill their stoichiometry (or atomic fraction) related to the compound itself. ERYA will use the compound atomic fractions and their relative mass fraction to convert automatically to atomic fractions of the whole sample.

Placing more than a mass value for the same Compound will deliver an error message alerting the user that the sample composition are not consistent.

For isolated elements, just fill the mass fraction term, since the atomic value will be ignored.

Example: To input a sample made with 20 μg of lithium carbonate (Li_2CO_3) and 400 μg of silver (Ag), the user should define 4 elements, when the isolated silver will have "400" for the mass column, while no matter the atomic column value inserted, it will be treated as "1".

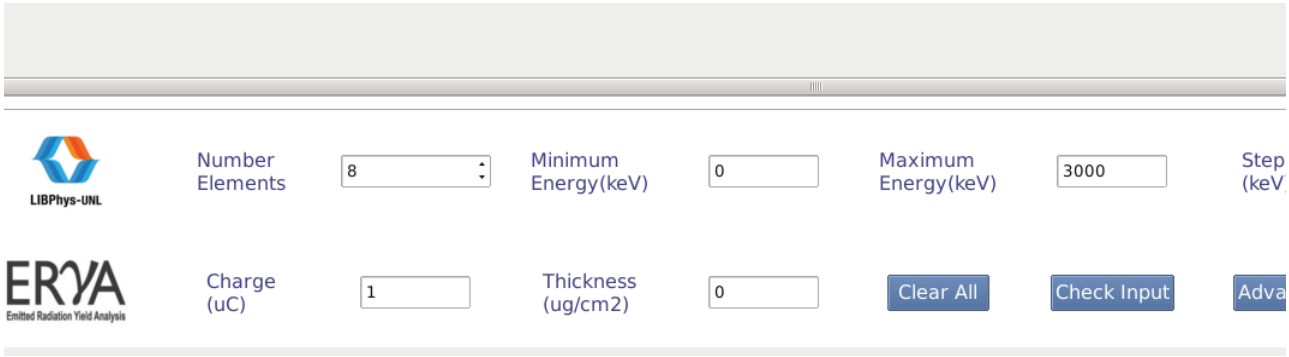
The lithium carbonate compound only need a single "20" value on mass for the first element (Li), then fill "2", "1" and "3" on atomic column that corresponds to Li, C and O elements.

Element	Gamma Peak (keV)	Fit	Fixed Ratio Group Number	Cross-Section Calibration Parameter	Mass Composition Initial Guess	Atomic Composition Initial Guess
Ag	0	☐ Fit ?		1	400	1
7Li	478	☐ Fit ?	1	1	20	2
12C	0	☐ Fit ?	1	1		1
16O	0	☐ Fit ?	1	1		3

6. Another important column is the cross-section calibration parameter, which default is "1". Systematic uncertainties related to cross section values, detector efficiency, beam charge collection, may be minimized by calibration of the experimental set-up made with known samples, leading to calibration parameters for each isotope (see for example [1]).

How to define your experimental set-up

Most of the input needed about the experimental set-up must be filled in the bottom bar of the main screen interface:

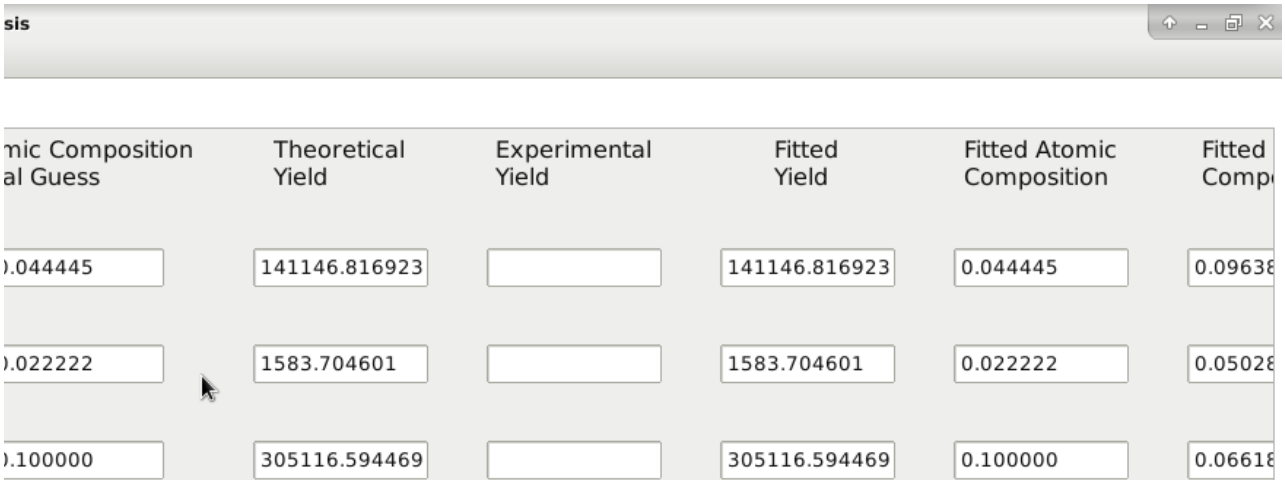


1. Maximum Energy – proton energy used for the analysis.
2. Minimum Energy – the energy cut-off for the sample integration. This can be zero, or below any initial energy referenced by the element's cross-section, although ERYA is actually summing zeros until reaching points that have non-zero values for the cross-section.
3. Energy Step – The integration step between the minimum and maximum energy values defined earlier.
4. Table Step – Set zero to skip this table. Optionally, define a non-zero value on "Table Step", to create a table (on the last tab of the main screen) to display a list of yields between the minimum and maximum energy, using the user defined value as step. This table displays always the results after a successful fit, or the initial composition values if the user do not choose any fitting at all. (A fit will be triggered if at least a single "Fit" checkbox is marked.)
5. Charge – fill in μC units.
6. Sample thickness (μg/cm^2) – set to zero for a thick sample (wider than the proton range in the sample); enter the thickness of the sample for non-thick samples.

►Now click "Run" to get the theoretical yields, which will appear on the correspondent columns of the main table. The whole progress runs with an additional progress bar gauge window.

How to Fit experimental results

ERYA provides a resource to optimize the composition of the sample in order to fit the experimental yields obtained for the sample analysis. This process needs the following steps.

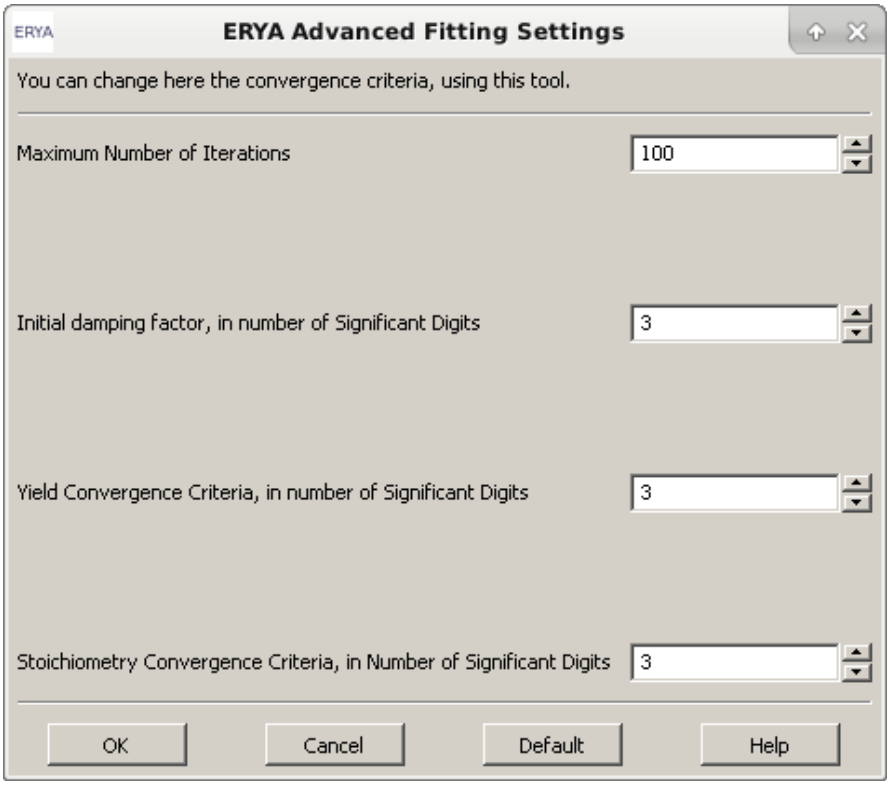


1. Fill the "Experimental Yield" column with the experimental results.
2. Click on the third column the "Fit" check box, to choose the elements whose atomic (and mass) concentration you want to optimize. If for a given element there is no click, ERYA will keep for that element the atomic (and mass) concentration provided by the user.
   This is helpful, mainly in relation to those elements for which there are no gamma yield and/or gamma cross sections, but for which the user knows (from PIXE, for example) their concentrations.
3. Elements which are part of a compound and for which the user has assigned the same Group Number (column four), will be treated as a single fitting variable. If all of them are chosen to be fitted, their relative composition ratios will not change.
4. Now click on "Run" to start. ERYA will display a progress bar window during the fitting (indicating the current fit step until the maximum number of steps).

   Note: If the user define a non-zero value for "Table Step", once the fit procedure ends, it will use the fitted composition values to evaluate the table yields.
5. Once finished all numerical calculations, ERYA will ask if the user wants to copy the fitted sample composition to the initial column (for example, for a chain of successive fits), leaving the user to choose the best option on his (her) context.

About the fitting procedure

The program applies the Levenberg-Marquardt algorithm to perform the fitting procedure, and the fitting computation times depends on the number of fittings parameters, initial values and additional fitting settings. Normally, it only takes a few seconds for simple samples. At the bottom bar of the main screen interface the user may choose "Advanced" to access and modify details about the fit, such as the maximum number of iterations and parameters of calculation precision.



Maximum number of iterations – 100 (which is also the default value).

Initial damping factor (a damping gradient norm) – number of significant digits from 0 to 3 (default is 3),

Yield convergence criteria – number of significant digits from 0 to 6 (default is 3).

Composition convergence criteria – number of significant digits from 0 to 6 (default is 3).

As always, precision and computation time represent a trade-off which should be taken into account by the user.

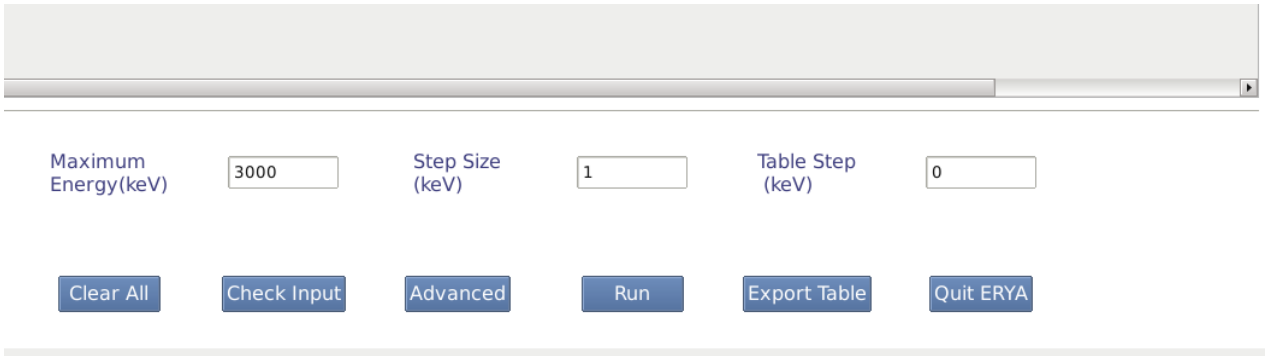
Loading and Saving Results

Once the numerical simulation ends successfully, the user can store the results, both the ones on the main screen and the table of yields if created, in two major file formats:

►Excel file (extension **xlsx)**, by clicking on the "Export Table" button and giving a name and directory location of the file.

You need Microsoft Excel 2007 or newer version, or a compatible alternative such LibreOffice or Google Sheets to open the ERYA's own created Excel files.

Third party programs capable to save and load Excel 2007 files should read the files created by ERYA, as this program only use a few functionalities of this proprietary format.



►A XML file (extension **epz**) from File > Save As.

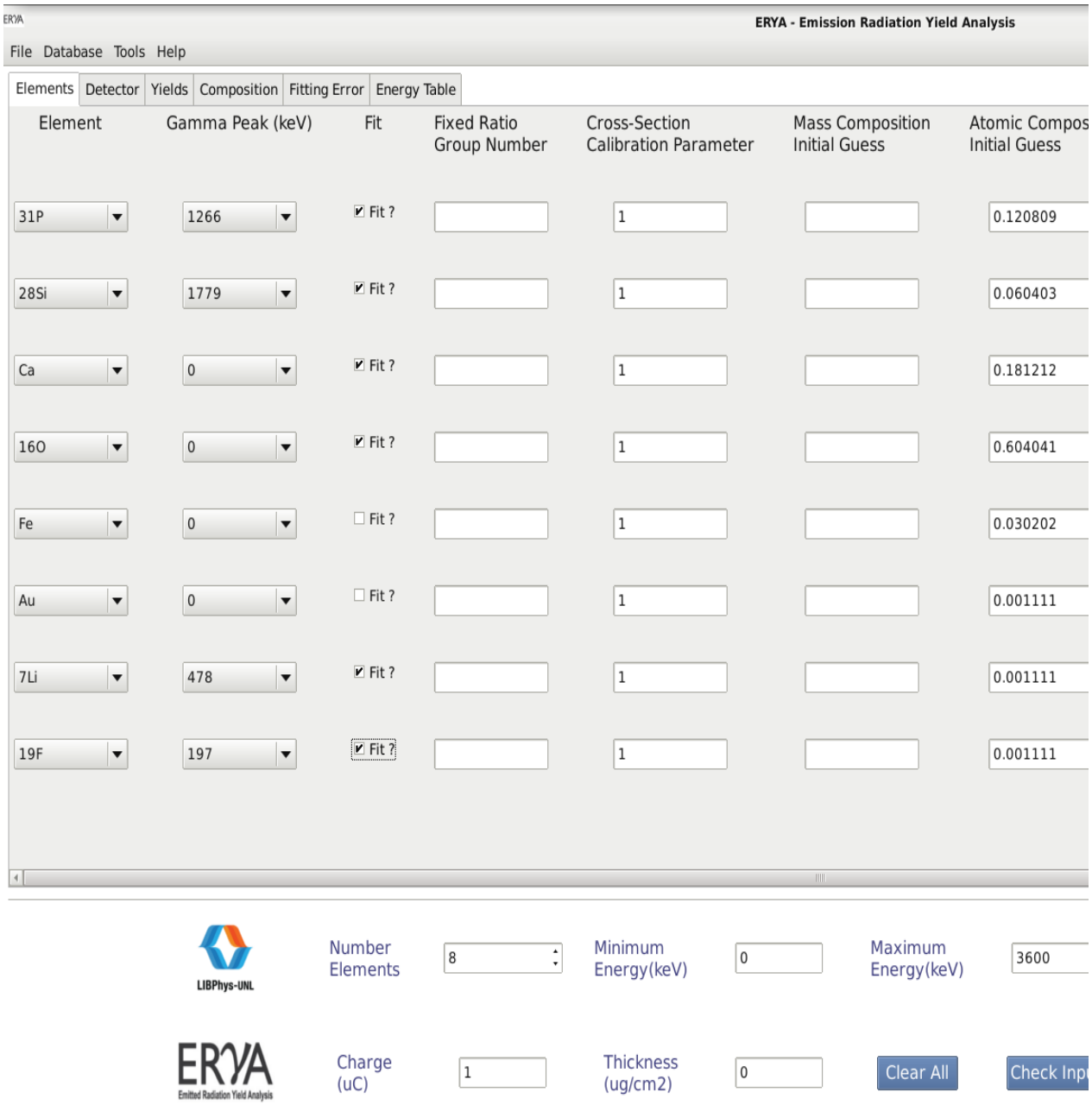
Opening output files in ERYA is only possible for native **epz** files. Once loaded, ERYA asks to run a new simulation with the initial values loaded from the file.

Example of a PIGE analysis

Sample composition: major composition – calcium phosphate mixed with silica; minor elements – iron and gold; minor (trace elements) – lithium and fluorine. The user may have an idea of the composition given by other methods, as for example, PIXE, to use as a first guess.

PIGE spectra collected for 3600 keV protons. Gamma-ray available only for ^{15}P, ^{28}Si, ^{19}F and ^{7}Li. Corresponding yields obtained for 1.0 μC of collected charge.

Step 1. On the main interface (see below) introduce 8 for the number of elements and then choose the elements (as Ca, Fe, Au) and the isotopes with corresponding gamma-rays.



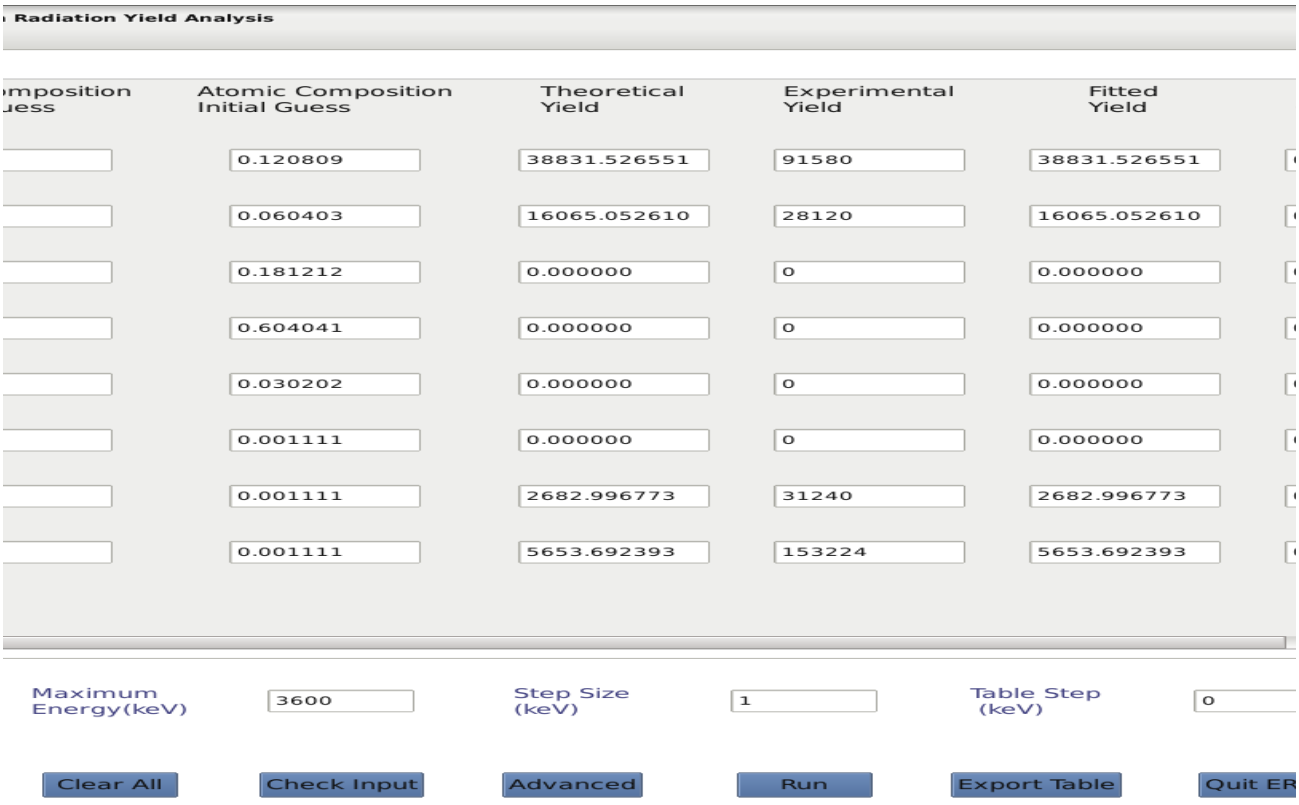
Step 2. As gamma-ray yields and excitation functions are available for ^{15}P, ^{28}Si, ^{19}F and ^{7}Li, check the boxes of column 3 related to these isotopes in order to fit. Every major element must also be chosen to fit, in order to obtain the adequate stopping power for the calculations. As no yields are given for these elements, which would constraint the fitting procedure, their number must be as small as possible, in order to get an unequivocal answer. In this example, Ca may be grouped with phosphorous (column 4 – assign the same number) in order to keep their atomic ratio fixed during the fitting and have one free parameter less. Oxygen may not be grouped with these because it is part of silica besides entering the phosphate composition and may be linked to other elements. So, it should be chosen to fit, but not grouped. In relation to Fe and Au which are minor elements, they should not be chosen to fit.

Step 3. If you have calibrated your PIGE experimental system and have calibration parameters for ^{15}P, ^{28}Si, ^{19}F and ^{7}Li, fill-in on column 5; otherwise fill in the number 1.

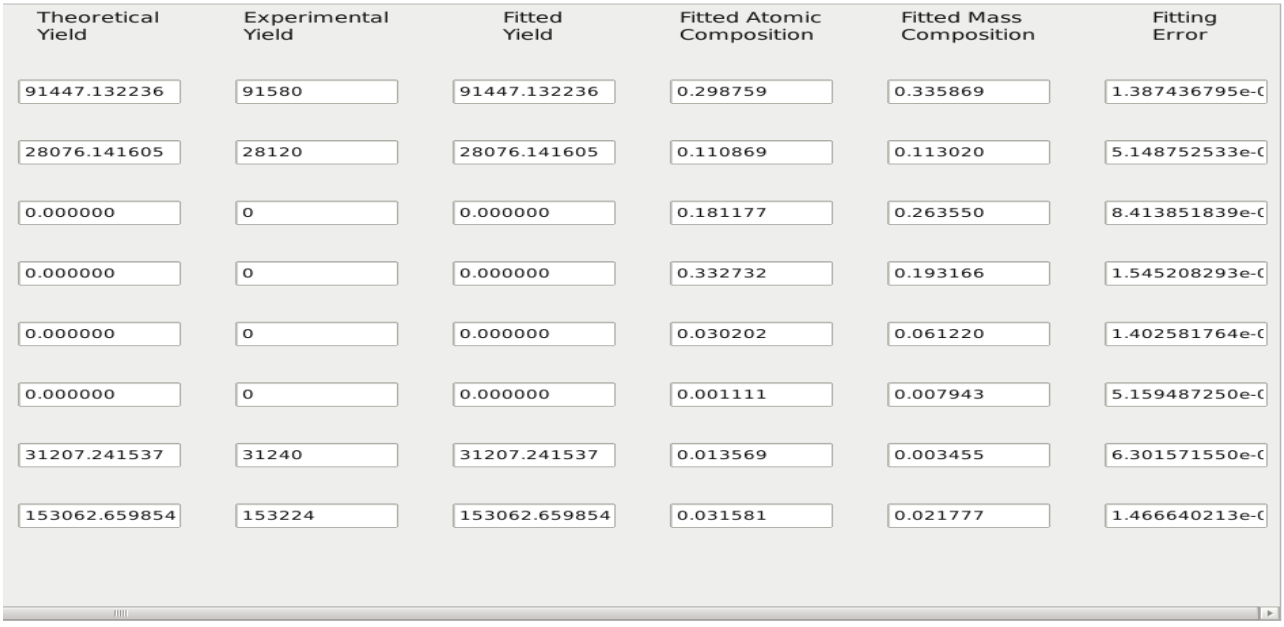
Step 4. Give in atomic fractions your initial guess of the sample composition column 6. You may use numbers higher than one. ERYA will renormalize the sum of the sample composition fractions to one.

Step 5. Enter the experimental gamma-ray yields – column 8. The yields of column 7 will be calculated by ERYA.

Step 6. Fill in the bottom bar of the main screen introducing maximum energy, step size (integration step), charge, thickness (= 0 for a thick target).



Step 7. Press "Run". After the fitting procedure ERYA will fill-in information about fitted yields, fitted composition in atomic fractions, fitted composition in mass fraction together with its fitting uncertainty.



Theoretical Yield	Experimental Yield	Fitted Yield	Fitted Atomic Composition	Fitted Mass Composition	Fitting Error
91447.132236	91580	91447.132236	0.298759	0.335869	1.387436795e-(
28076.141605	28120	28076.141605	0.110869	0.113020	5.148752533e-(
0.000000	0	0.000000	0.181177	0.263550	8.413851839e-(
0.000000	0	0.000000	0.332732	0.193166	1.545208293e-(
0.000000	0	0.000000	0.030202	0.061220	1.402581764e-(
0.000000	0	0.000000	0.001111	0.007943	5.159487250e-(
31207.241537	31240	31207.241537	0.013569	0.003455	6.301571550e-(
153062.659854	153224	153062.659854	0.031581	0.021777	1.466640213e-(

Step 8. Save the results, using the "Export Table" button, which stores the data to an Excel file.

Extra Step. If the user would like to have the gamma-ray yields of the fitted isotopes as function of the proton energy, he (she) must change the table step for the number in keV the user chooses and a table with this data will also be provided.

Extra Step. In order to have less computational time (if needed), the user may open the "Advance" button and reduce the composition and yield precision to two decimal places, and reduce the gradient damping to two decimal places too.

Tip: Since ERYA gives the chance to copy the Fitted Composition values to the Composition Initial Guess column after each simulation, the user can deselect the "Fit" flag for elements which have fitted yields and experimental yields very close to each other and then make another "Run", now with less fitting variables, and less computational time.

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).

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.

- Ananth Ranganathan, "The Levenberg-Marquardt Algorithm"

- K. Madsen, H.B.Nilsen, O. Tingleff, "Methods for Non-Linear Least Square Problems".

- 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 -2.5×10^{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				
	$^{24}Mg(p,p_1)^{24}Mg$	1369	90º	N. Sharifzadeh et al. [10]: 998 – 2999 keV
^{24}Mg			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