ERYA Bulk

User Guide & Tutorial

Nuclear Reactions Group, LibPhys-UNL (2018)
Departamento de Física da FCT/UTL, Lisboa

Contents

Introducing ERYA

Welcome!

ERYA is a software application designed to aid the analytical analysis of bulk samples, where the main task is determine their composition in terms do their elements and/or isotopes quantities.

The efficiency of this implemented method by this software highly depends from the physical model, experimental setup and measurement accuracy.

In general, the measurement of samples beamed by light particles are more precise when the sample had lighter elements than weightier ones, and this fact make possible to use the excitation function of lighter elements by their gamma emissions to determine the composition of the sample.

Since this method are based from the spectrum analysis of gamma emissions of the sample's elements, it justifies why this analytical method is called PIGE (Particle Induced Gamma Emission).

This program avoids the problem-prone and time consuming tasks to evaluate the results, that once require several printed tables, and an huge man-powered tedious calculation task. With this software it gives the results in a matter of seconds.

ERYA-Bulk was developed at LibPhys-UNL, by the institution *Faculdade de Ciências e Tecnologia* (FCT) - *Universidade Nova de Lisboa* (UNL), on *Departamento de Física* (Physics Department), postal code address 2829-516 Caparica, Monte da Caparica, Portugal.

ERYA-Bulk was programmed by Vasco Manteigas (vm.manteigas @ campus.fct.unl.pt), with advice from Micaela Fonseca (micaelafonseca @ fct.unl.pt), and also the original programmer of the previous LabView ERYA version.

Disclaimer: All Software screen-shoots are made from the Windows version running on a Linux computer using the Wine compatibility layer. Some screen-shoots from Mac OS X version are provided for additional acquaintance.

Physical Theory of Gamma Ray Emission Yield Analysis

The program evaluate the stoichiometry composition of an homogeneous sample, although their composition can be made from a mixture of several elements, by evaluate the nuclear reaction yield of each element. Since in theory the yields depends of their relative stoichiometry, it is possible to find the correct sample composition using this method.

Each individual yield can be evaluated from the following integral:

$$Y(E_0) = \epsilon_{abs}(E_\gamma) n_p f_m f_i N_A A^{-1} \int_0^{E_0} \frac{\sigma(E)}{\epsilon(E)} dE$$

Where it depends from the Detector Efficiency (ε_{abs}) at the element gamma ray emission energy, the number of protons (n_p), the element/isotope relative abundance (f_i), the relative fraction mass (f_m), the isotope/element atomic mass (A) and Avogadro's Number (N_A).

Both multiplied by the proper integral in function of energy, where depends from the element's excitation function cross-section (σ) and the stopping power (ε). Each of relevant functions depends from the relative stoichiometry of the sample.

Taking the required number of experimental yields from each relevant element, it is possible to adjust the stoichiometry for all elements to minimize the difference between the experimental values and the conjectured composition. This process is done automatically by ERYA, and returns the best stoichiometric ratios fitted.

A more detailed description of physical theory can be obtained from the own program's online help reference.

What's New on ERYA Bulk?

This application is a full rewrite of a previous LabView program that handles the PIGE measurements on 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, are derived from a cross-platform library called wxWidgets.

wxWidgets is an open-source framework inspired from the Microsoft Foundation Classes, and compliant with the Standard Template Library, while extends to be compatible to several operating systems and computer architectures, easing the port to several systems.

The new ERYA Bulk (Emitted Radiation Yield Analysis) are available natively for Windows, Linux and Mac OS X, and several modern computer architectures, since it runs on 32-bit and 64-bit CPU, either x86-64 or ARMv8.

Binary packages are available for Windows, Linux and Mac OS X, with support of most common architectures.

Windows and Linux versions had 32-bit and 64-bit Intel/AMD x86 binary packages, and require Windows 7 or a better version to work. (Windows 10 works without known issues.)

Mac OS X are only available on 64-bit x86-64 Intel architecture, and requires version 10.11 or greater to work. (ERYA will not check the Mac OS X version, and can run on some older versions as well.)

Linux packages for ARM machines, both 32-bit or 64-bit, are also supported and intended to use with ARM mini-computers like the Raspberry Pi 3, or based on Snapdragon ARM CPU.

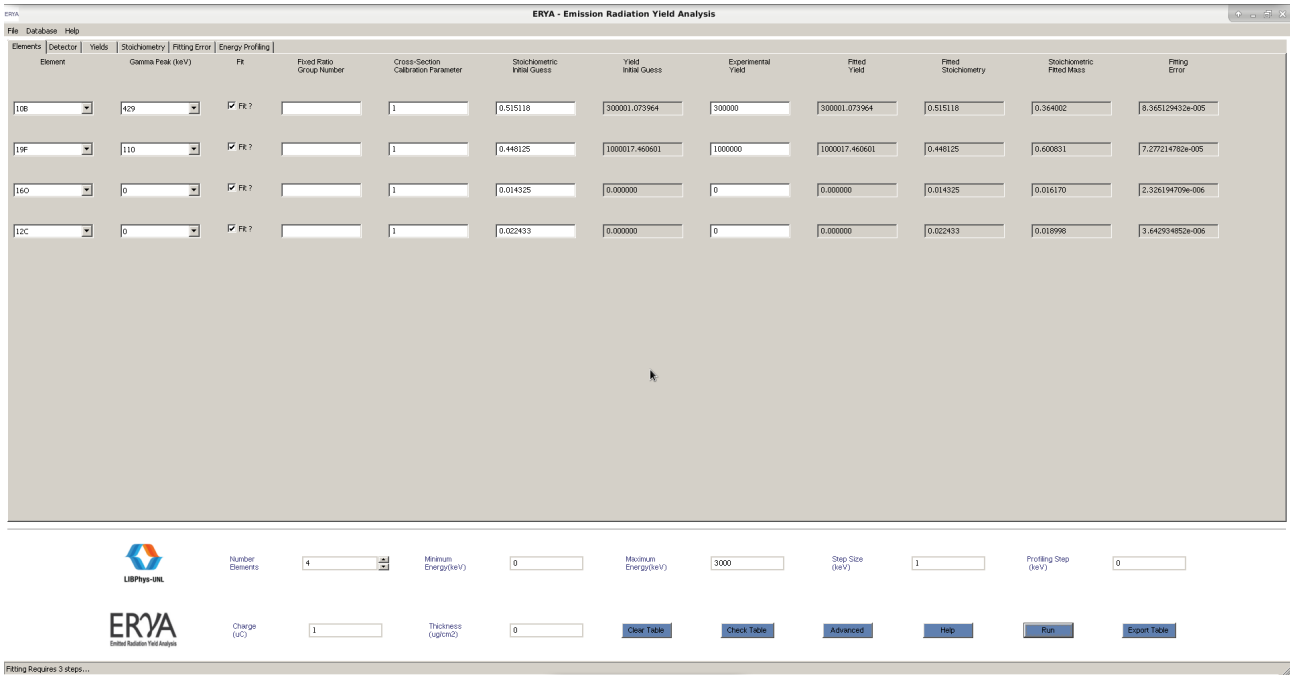
All Linux packages are designed to work with any Debian derived Distribution, as long contains the wxWidgets 3.0.4 libraries updated, and require at least Debian 9 Stretch with backports repositories enabled, or Ubuntu 18.04 LTS versions to work.

If you are interested to contribute to maintain the wxWidgets ERYA Bulk source-code, or any issue related to the program use, please contact the programmer by it's e-mail.

The standard package contains a Database with over 100 elements and isotopes, the 1977 and 1991 Ziegler Parameters, the 2013 SRIM Stopping Power tables, and two sample Detector profiles. Both of all those files can be edited by the user, using the tools provided by this software that will be described on next chapters on this tutorial.

The software can export the original LabView ERYA Databases, and convert them to the native wxWidgets ERYA Database format, saving countless hours of manual workarounds to export the databases.

A brief display of the ERYA-Bulk user interface:



At a technical level, the major features are:

1. A new unified Database format for the Elements, the Detector Efficiency and Ziegler Parameters, that use the standard XML document format, given flexible data storage, an hierarchical data content structure, and a more standardized method to read and write the data.

2. An on-line help, based on HTML pages, and directly accessed by the program, which explain the main program features, and major tasks.

3. New file filter imports for IBANDL files, SRIM files, generic spreadsheet-like ASCII source files and Excel 2007 Xlsx files, in order to the user ease the extraction of the relevant data, and minimize the steps needed to convert the original data to the native ERYA database structures.

4. The whole GUI was written from scratch, and use the new wxWidgets objects which make more easy to handle events, and implement a more concise work-flow. And also simplifies the porting to several operating systems.

More details can be obtain from the ERYA-Bulk own on-line help system.

ERYA v4.10 (ERYA/PIGE Bulk Tutorial)

Quick Start!

To install ERYA on any supported operating system are straightforward simple…

- Select the adequate package which matches your operating system and architecture, and download it to your computer. Both versions are, by default, installable programs, which means it just you need to run the package installer that will move the asset files to the default protected program directory. Once installed, just open the menu entry of ERYA in order to run the application.

Windows: For Windows users, all versions contains the wxWidgets libraries and GNU C++ Runtime Library on the application installation package. The installation program should install ERYA to the default Program Files folder, and will create an Start Menu entry.

Linux: All supported Linux packages versions are available as an installable Debian package. Ubuntu or Mate distributions usually gives a GUI tool to install packages, such as Synaptic or the Gnome Software Center.

Without install Synaptic, Debian will require the user to use the command-line prompt to install.

The advantage of Synaptic, is that will automatically download the necessary libraries automatically, also called dependencies, without additional steps.

No matter the Linux distribution officially supported, Linux users can open a Terminal application, update your Linux distribution, and install ERYA using the apt utility.

$ sudo apt update (Refresh the repositories)

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

$ sudo dpkg -i eryabulk_4.00-5_amd64.deb (If your system is a 64-bit Intel system.)

$ sudo dpkg -i eryabulk_4.00-5_i386.deb (If your system is a 32-bit Intel system.)

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

$ sudo apt -t stretch-backports -f install (Required for Debian 9 Stretch install the libraries.)

Note: You need a Linux distribution compatible with Debian 9 Stretch to install ERYA correctly.

Note: On most recent Debian systems, the sudo command may not work, then login a root shell first:

$ su (And type your root password)

apt update … (And use the previous commands without the sudo, since it is a root shell.)

Mac OS X: For Apple Macintosh users, it only requires to download the zip file to your computer.

Once extracted the zip file, just copy the ERYA-Bulk.app package to the Applications folder.

Finally, click on ERYA-Bulk icon to run the software on Mac OS X.

Nuclear Reactions Group, LibPhys-UNL (2018)
Departamento de Física da FCT/UTL, Lisboa

Linux ARM: If you have a Linux mini-computer that use an ARM processor, select the 32-bit *armhf* package, or the 64-bit *arm64* package, if your system or operating system supports it.

To install ERYA on Linux ARM, is the same procedure as the Linux Intel, as described before:

$ sudo apt update (Refresh the repositories)

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

$ sudo dpkg -i eryabulk_4.00-6_armhf.deb (For 32-bit ARM)

$ sudo dpkg -i eryabulk_4.00-6_arm64.deb (For 64-bit ARM)

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

$ sudo apt -t stretch-backports -f install (Required for Debian 9 Stretch install the libraries.)

Note: Only recent ARM mini-computer boards support 64-bit binaries!

For example, the Raspberry 3 model support both the 32-bit and the 64-bit versions, but require to install the Debian 9 Stretch *arm64* version to work on 64-bit.

Tip: ERYA on Linux requires a X11 Window server to run, or any replacement compatible to X11 Window Server. On ARM computers that usually had little RAM, it is recommendable to use a low-memory Desktop Environment like the LXDE or XFCE to run ERYA properly, even this software requires little memory to run correctly.

Uninstall ERYA:

Windows: From Start Menu, selecting Uninstall Program from Control Panel. Once the program list as loaded, select “ERYA-Bulk”. It will ask to uninstall the program, and answer “Yes”.

Open the user profile directory from Windows Explorer, and delete the contents of “ERYA-Bulk”.

Using the Command Prompt, just type:

$ **rmdir -s C:\users\name\Local Settings\Application Data\ERYA-Bulk**

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/**

Initial Setup

When ERYA-Bulk is started for the first time, it will start a wizard to guide the user to create a configuration file, that should include some databases to work properly.

During the database selection, the wizard will asks sequentially, from which bundled files, are the default Element Database files, Detector Efficiency files and Ziegler Parameters files.

And on final step of configuration wizard, it will asks if the user want 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-Linux/***	**/Users/name/Library/Application Support/ERYA-Bulk-OSX/***
Config Name	**ERYA-Bulk-Win64.conf**	**ERYA-Bulk-Linux.conf**	**ERYA-Bulk-OSX.conf**

Warning! Do not choose a Portable Setting if you install the program to a protected directory.

Once the setup wizard got all necessary information, ERYA will check the options, and source files, and convert them to the native XML format if necessary.

If all steps are done correctly, the configuration file are created, and the selected three Databases on that configuration file are loaded and the main GUI interface will appear.

The Configuration File are also a XML file, and their placement depends from the following profiles, decided by the user during the setup wizard forehand:

Portable Setting: Once the configuration file are written on the root program directory, the setup is complete. Any changes on selected Detector, Elements and Ziegler will overwritten the original program ones, since you set the same working directory.

Local User Profile Setting: Once the profile folder are created, according to the previous table, and the configuration file are written, a copy of Database, Ziegler and Detector files are created on the same local user profile folder. Any changes by the user will change, by default, the local copies, not the original ones on the program folder. At least you can save the original files as an informal backup.

Note: If the initial setup are completed correctly, the GUI interface should start automatically.

ERYA will warn with an error message if any essential database cannot be loaded.

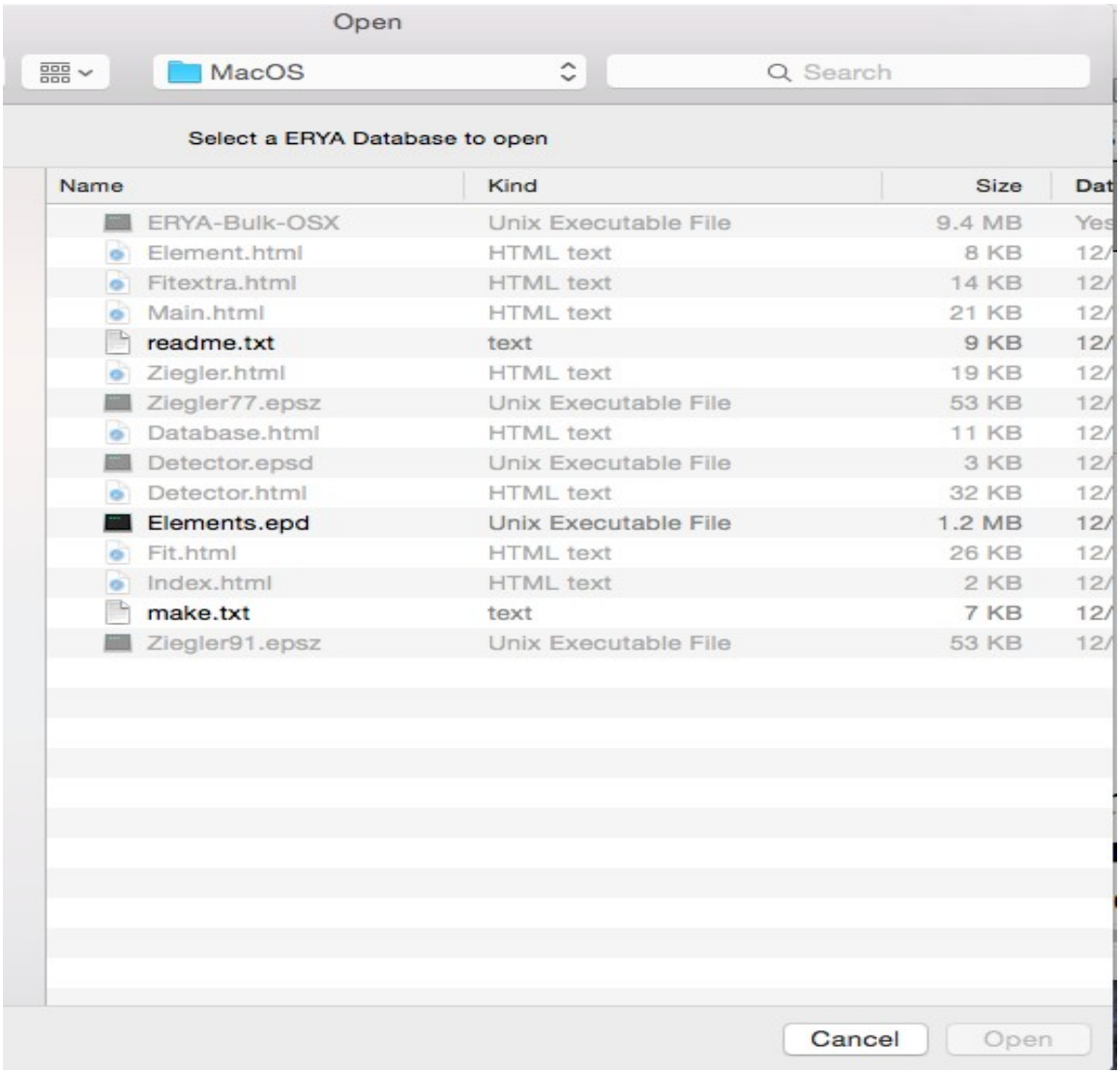
Initial Setup Step-by-Step

1. Once ERYA are started for the first time, the application will display a warning about absent configuration files. If you have installed before, it is highly recommended to delete first the local user profile folder listed on the table on the previous chapter, then repeat this step again.

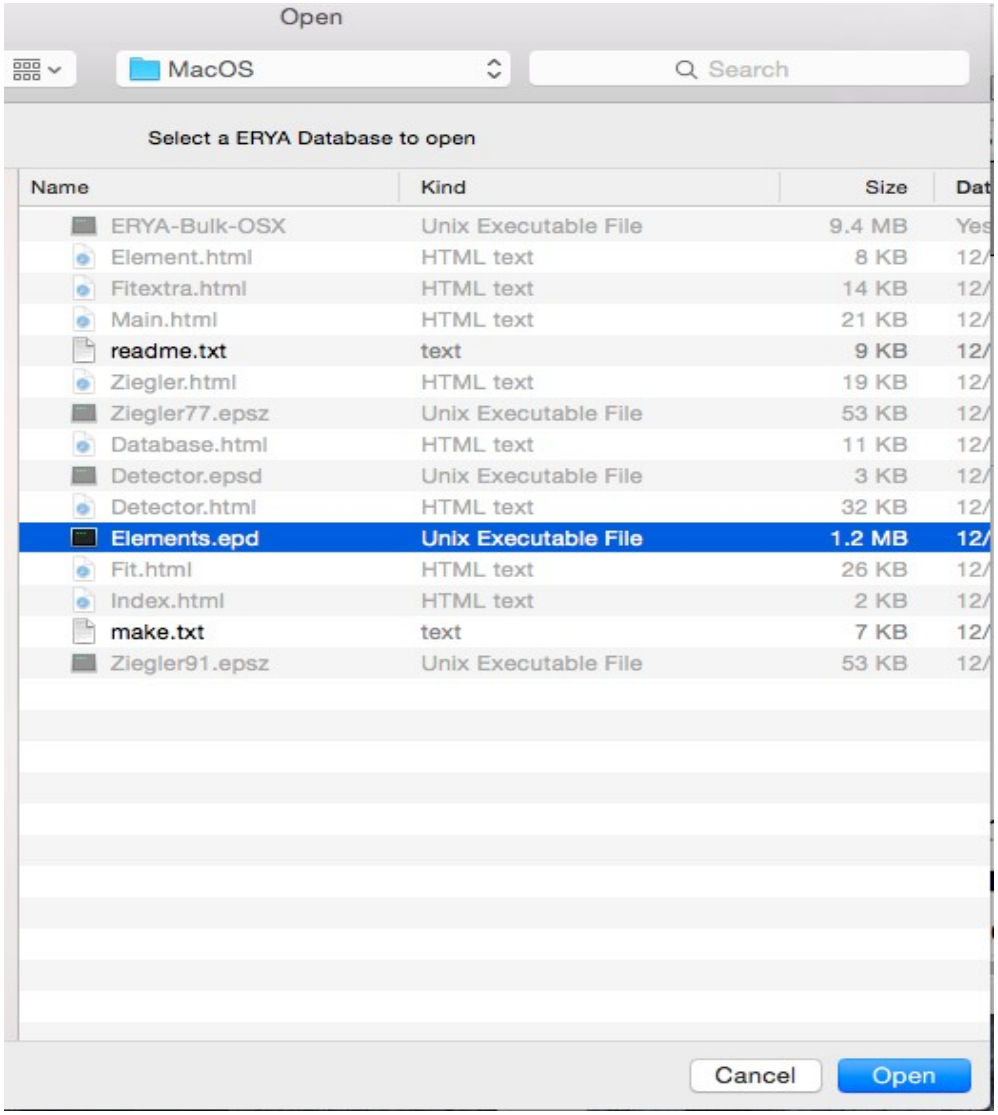
2. It will start a wizard, guiding the user for the next steps. Just click on “Next” to go forward, or “Back” to go back. The wizard will not proceeded until the user choose all obligatory settings.

3. Now the program will asks from the ERYA’s package, which file should be the default start-up Element Database. Since Windows and Linux standard dialogs are more easy to filter files to the supported extensions, just select the file with **epd** extension.

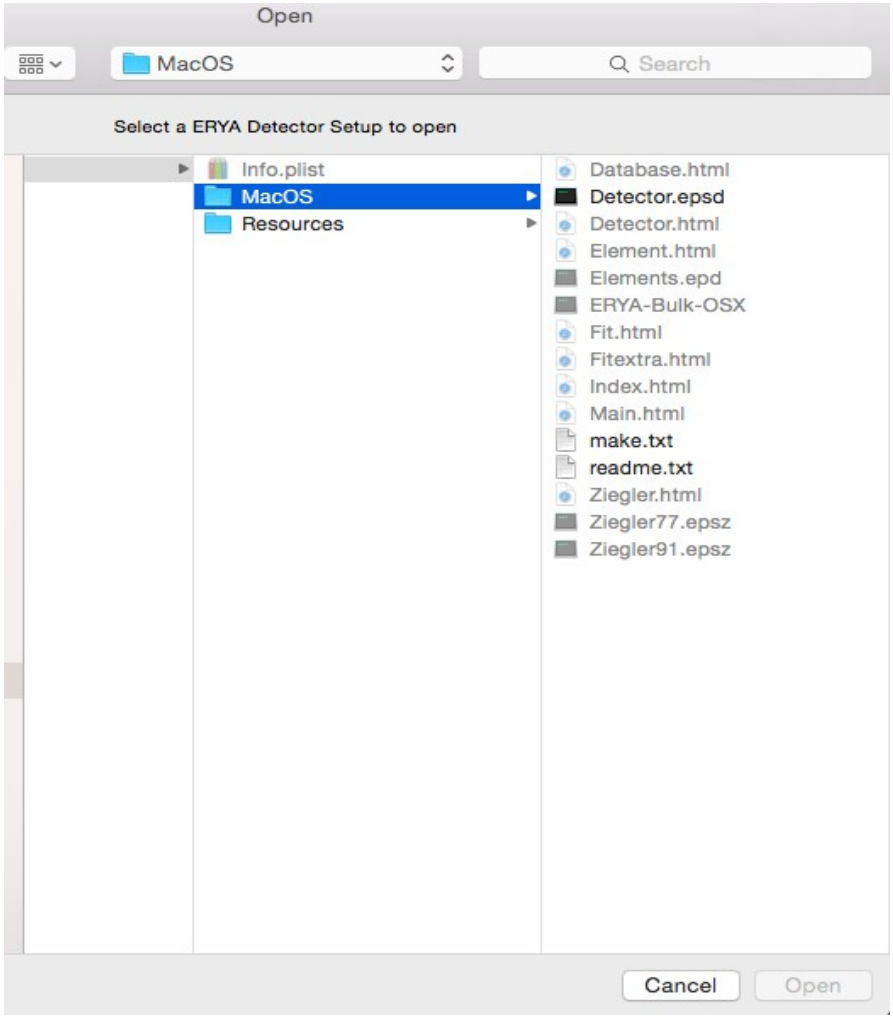
On Mac OS X, the standard dialog will display all files, but shades the incompatible ones:



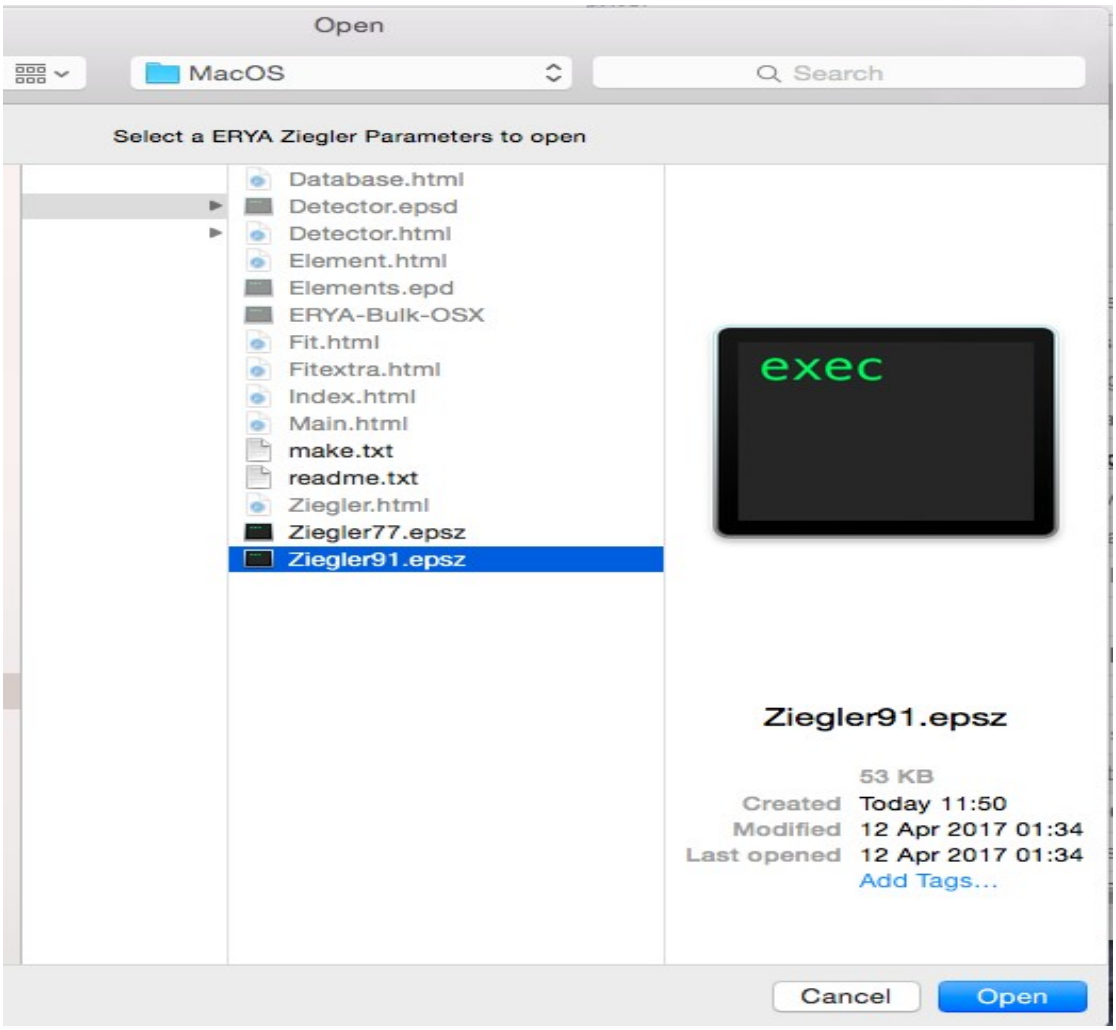
And select the file with **epd** extension, as indicated:



4. The same logic will apply for the Detector Profile, and select the file with **epsd** extension:



5. Finally, select the file for the Ziegler Parameters, with **epsz** extension, where it is recommendable to choose the most recent version:



6. At the last page, the user can selected the local (default) folder profile, providing by the default operating system's settings, or the root folder (the same of the main program). It is highly recommendable to select the local profile setting.

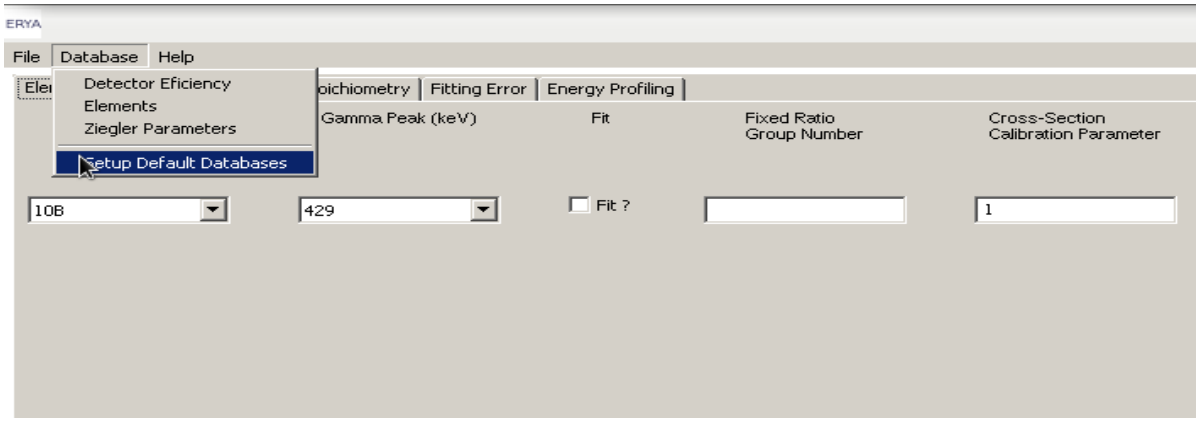
7. The selected files are analyzed, and placed on the target directory.

8. The configuration file are created, and the main GUI is loaded automatically. ERYA-Bulk is now ready to use, unless an error occurs (see the Troubleshooting section for additional details).

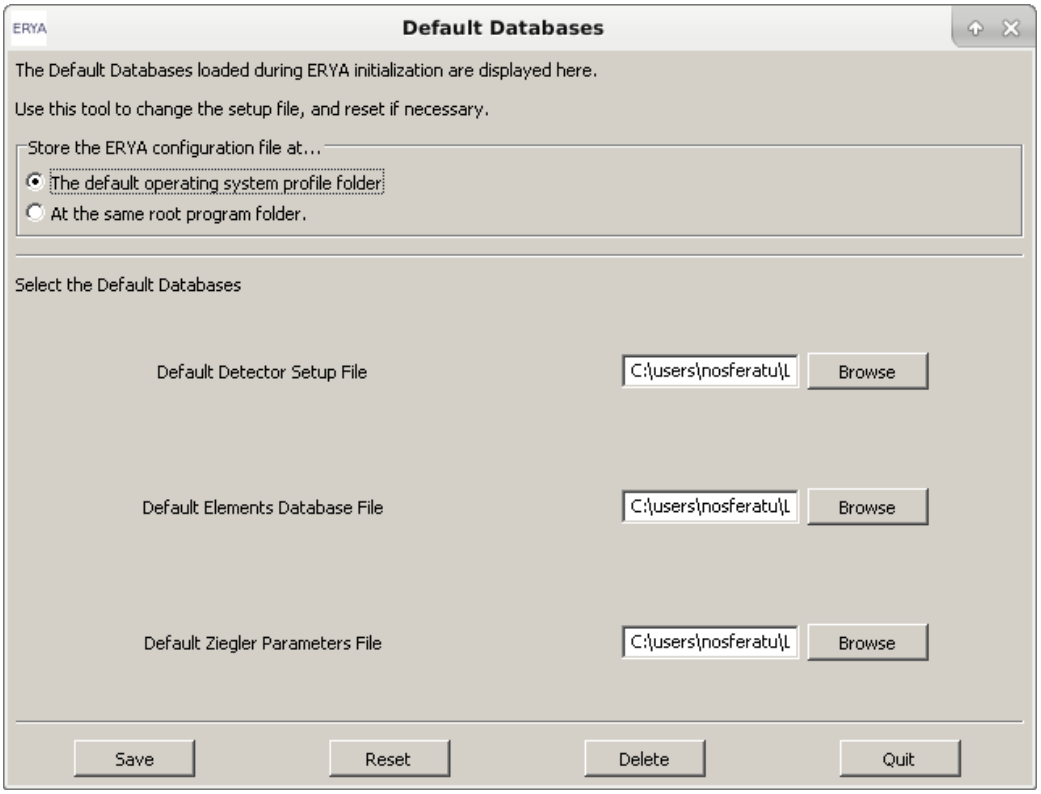
On next start-ups, the program will load the configuration file automatically, and the main GUI starts without additional user input.

Changing Setup Settings

To change the Default Databases, you can either delete the configuration file from the selected settings folder, edit it with a normal ASCII text editor, as long you understand the XML tree, or use the built-in widget from Database > Setup Default Databases...



Where a new window opens, and the user can change the current settings, and update the new ones.



ERYA will check the new default databases, and update the configuration file, if successfully.

The program will warn the user when the new settings are updated, and can resume from it.

Setup Troubleshooting Guide

1. If you make a mistake, either on setup wizard or when edits the configuration file, that makes ERYA warns about corrupted or non-existent databases, then the program will start without any databases.

In this case you can use the Setup widget tool as explained before, and then select “Delete” to delete the wrong configuration file, and it will force ERYA to restart the setup wizard again.

A manual configuration can be made, by clicking “Reset” on the setup widget described on last chapter, and select all necessary databases from any source. Once click “Save”, it will create a new configuration file, along a copy of the selected databases on the same setup profile directory.

If necessary delete manually the user profile folder as detailed on “Initial Setup”, to force ERYA to start the initial setup routine.

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

2. If the ERYA warns that cannot write any file, even the configuration file, to the selected folder, this means you are trying to write on a protected folder that requires additional privileges.

ERYA-Bulk’s author recommends that in any circumstance, **you should not run** ERYA-Bulk with administrator privileges, since it is not necessary, and can make a mistake that could overwrite important operating system files unnecessarily by user’s own mistake.

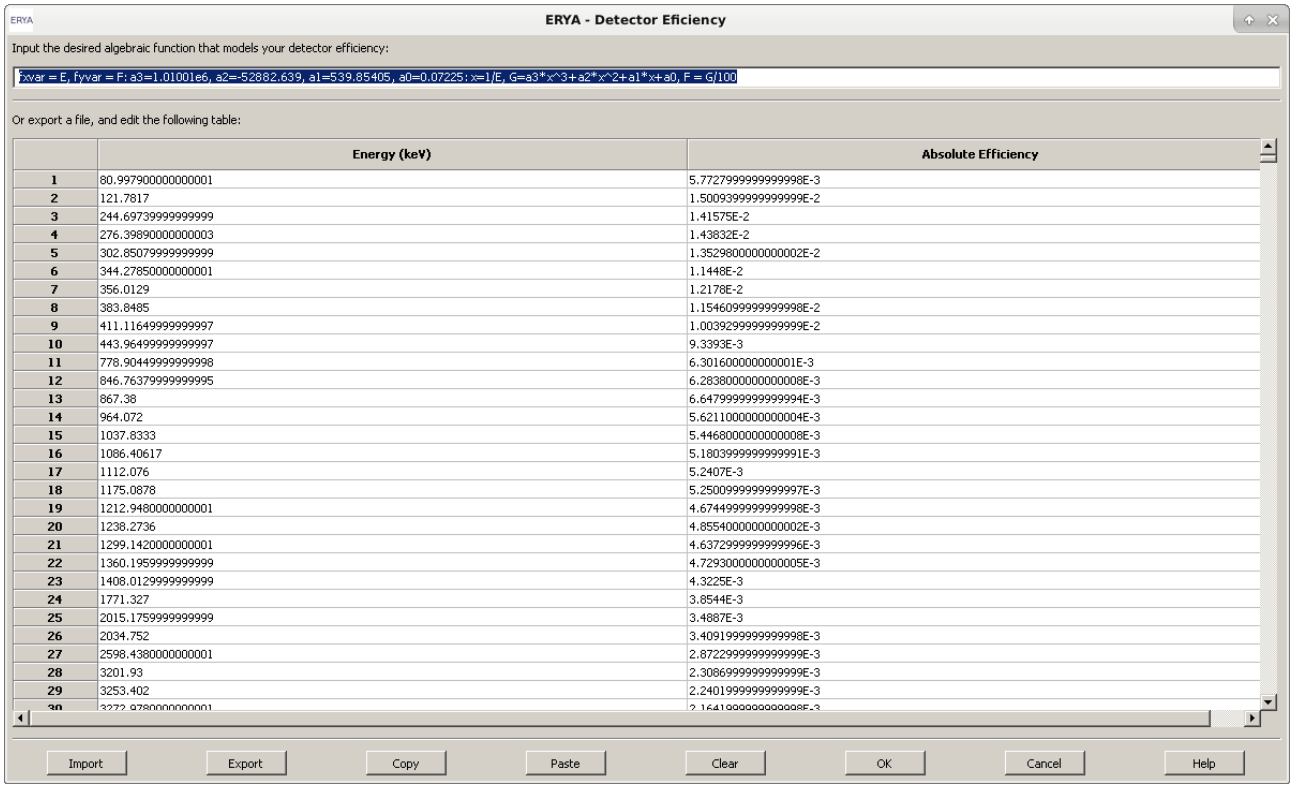
3. The optimal screen size for ERYA is a FullHD screen (1080p), but can work on a 720p screen at minimum. The main window screen will create scroll bars in case when the screen don’t fit all the elements on the same time. If the other windows appear cropped, then you need to increase the screen resolution.

Explore the Database Management Tools

Detector Efficiency

One of key core databases required to ERYA operate is the Database Efficiency profile file.

To manage the Detector profile, open the widget from Databases>Detector Efficiency, and then should open the following dialog panel:



	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
20	1238.2736	4.8554000000000002E-3
21	1299.1420000000001	4.6372999999999996E-3
22	1360.1959999999999	4.7293000000000005E-3
23	1408.0129999999999	4.3225E-3
24	1771.327	3.8544E-3
25	2015.1759999999999	3.4887E-3
26	2034.752	3.4091999999999998E-3
27	2598.4380000000001	2.8722999999999999E-3
28	3201.93	2.3086999999999999E-3
29	3253.402	2.2401999999999999E-3
30	3272.9780000000001	2.1641999999999998E-3

The operation are straightforward to use, when all possible operations clearly visible. Additional information can be retrieved from the on-line help, by clicking the “Help” button.

In practice, this editor combines a text field, intended to the user code an algebraic function, that should give the Detector Efficiency in function of Energy, and a two-column table to fill any experimental values for the Detector Efficiency that serve for the same purpose.

The origin of this dual configuration is mainly for compatibility with the previous LabView ERYA, although the algebraic function was hard-coded to a polynomial function, or simply load a text file with two columns of numerical data. In both cases, represents a function of efficiency in terms of energy.

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

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

Managing the Detector Efficiency profile file

While it is possible to fill directly the detector's efficiency data by using the built-in spreadsheet editor, the user can also export any compatible experimental values from external files.

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

By other hand, it is possible to export and import the experimental efficiency (the custom function are discarded!) from or to an ASCII file, which requires to be a full numerical two columns of data, separated by a space or tab (Any decimal number should use the C-like standard, like 5.5), or even an Excel file.

The Excel file should be a simple sheet (or fill the first sheet only) with only two columns of numbers, and it is highly recommendable to avoid any cell formatting, or attachments, but it is fine to add labels on the first row, which will 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 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".

Any edition can be stored by using the "Export" button, and then choose the native XML (epsd), ASCII (txt) and Excel (xlsx). You should overwrite the Default Detector Profile, as you defined on initial setup and described on "Quick Setup" chapter, to make permanent changes.

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

By clicking "OK", the current edition will be used by ERYA program as a new default Detector Profile, overridden the one loaded at start-up, and will use it until ERYA closes.

Hitting the "Cancel" cancel, any edition will be discarded.

Custom Efficiency Function

The optional algebraic efficiency function is the new ERYA feature, and on this widgets interface is just a input line box to the user code a custom function, overriding the table interpolation.

In technical terms, is a trimmed-down macro language designed to code custom functions without needing to recompile the application for different kinds of functions.

Taking the following example, as a good starting point:

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:

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

In reality this feature are a very simple programming language built into ERYA, implemented as an interpreter, and an extensive explanation are described on Detector's on-line help, requiring to click "Help" to read it.

A ERYA macro program requires at least two fields, and optionally three or more, depending of the convenience of readability.

The first field should be used to define the function variables names (It can be any sequence of letters and numbers, as long the first character is a letter), as long don't override the reserved words.

The special commands **fxvar** and **fyvar** followed by the assignment sign (=) and the variable name, is used to define the independent and dependent function variable, respectively. All sequences should be separated by commas (,), in order to the interpreter chain the variables to the stacks.

Any major field are separated by a colon (:) operator, where the interpreter will flush the main stack, and store the assigned variables values to the respective memory stacks. The last field, normally not flushed, are cached in order to reuse each time the ERYA main code calls it, acting as a compiling function in order to evaluate the function in terms of energy inputs.

An optional middle field can be used to define numerical constants, placing any valid variable name, followed by an assignment sign (=) and a numerical value, or any numerical or algebraic expression that can depends from previous variables values, defined earlier.

Like on the variable declaration field, and chain of variables are separated by commas (,).

In case of use any undefined constant on any expression, specially on the right side of the assignment sign (=), will trigger an undefined variable declaration, once you click the "OK" button, as it will make a basic syntax check before releasing the Detector's dialog panel.

The user then should fix the wrong declarations, and should try again until clear all macro code bugs.

The rightmost field are reserved for the algebraic function itself, and should contains only the function variables defined on the leftmost field, and the optional constant values created by pure convenience.

To simplify the function, it is possible to make a chain of functions separated by commas (,), as long all relevant variables and partial functions are previously declared.

Like the own example already shown, by defining a compound function by the variable *x* as x=1/E, the main function can be written as a normal polynomial.

On this context, if the user fail to properly define a function (omitting the **fxvar** or **fyvar** commands, for example), a fatal declaration error will be triggered once ERYA starts to check the macro code.

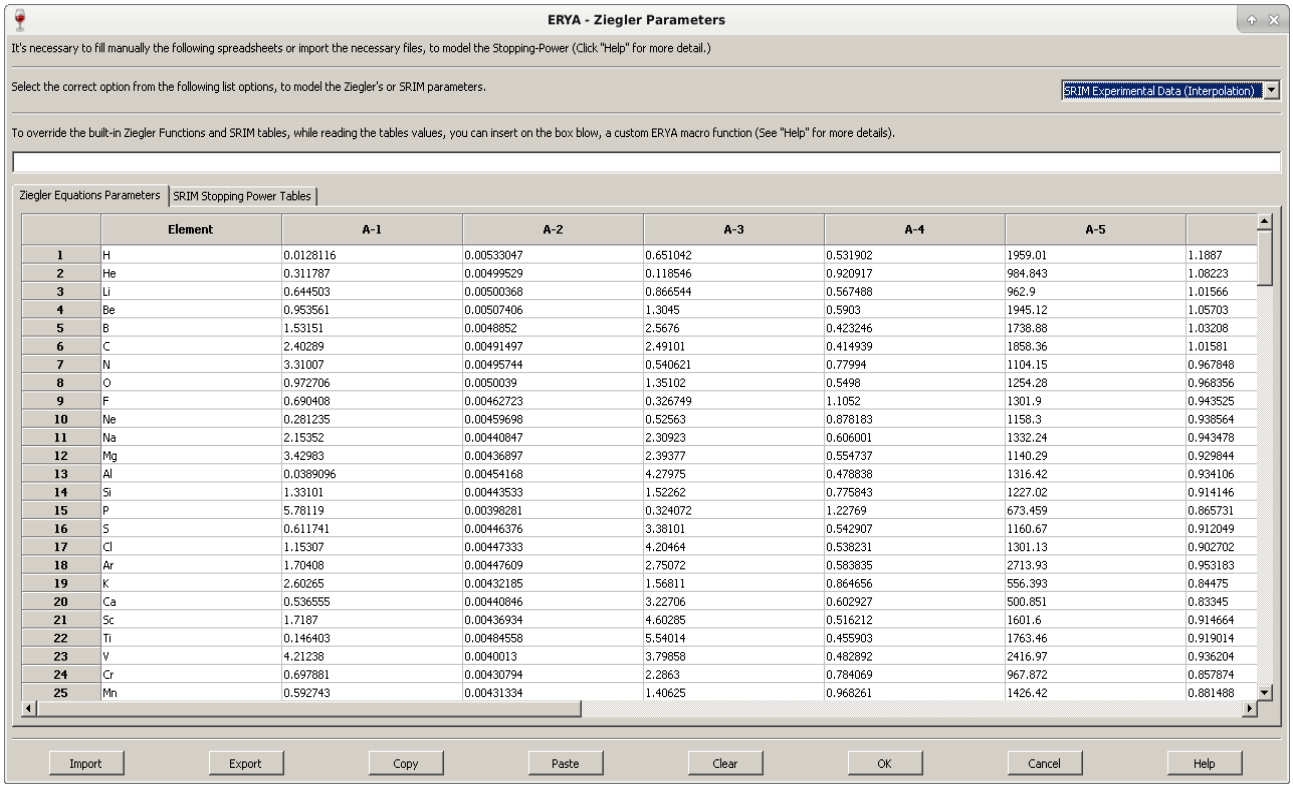
When ERYA loads the macro, it compiles to a chain of tokens, and caches the last field, making a kind of computer program that once got the input value for the independent variable, it will run the code until got a value for the dependent variable function, or an error that will trigger a simulation halt during the main numerical calculations.

Ziegler Parameters

Like the Detector counterpart, the Ziegler Parameter widget follows the same layout and functions, albeit exist some differences. The ERYA accepts the 1977 and 1991 Ziegler models, and both of them are hard-coded on the ERYA code. This editor manages the experimental parameters required by this model.

In order to expand from LabView ERYA original limitations, additional options to support custom algebraic functions, like on Detector's Efficiency, and also the SRIM tables are also supported.

The Ziegler Editor are opened from Databases > Ziegler Parameters



	Element	A-1	A-2	A-3	A-4	A-5	
1	H	0.0128116	0.00533047	0.651042	0.531902	1959.01	1.1887
2	He	0.311787	0.00499529	0.118546	0.920917	984.843	1.08223
3	Li	0.644503	0.00500368	0.866544	0.567488	962.9	1.01566
4	Be	0.953561	0.00507406	1.3045	0.5903	1945.12	1.05703
5	B	1.53151	0.0048852	2.5676	0.423246	1738.88	1.03208
6	C	2.40289	0.00491497	2.49101	0.414939	1858.36	1.01581
7	N	3.31007	0.00495744	0.540621	0.77994	1104.15	0.967848
8	O	0.972706	0.0050039	1.35102	0.5498	1254.28	0.968356
9	F	0.690408	0.00462723	0.326749	1.1052	1301.9	0.943525
10	Ne	0.281235	0.00459698	0.52563	0.878183	1158.3	0.938564
11	Na	2.15352	0.00440847	2.30923	0.606001	1332.24	0.943478
12	Mg	3.42983	0.00436897	2.39377	0.554737	1140.29	0.929844
13	Al	0.0389096	0.00454168	4.27975	0.478838	1316.42	0.934106
14	Si	1.33101	0.00443533	1.52262	0.775843	1227.02	0.914146
15	P	5.78119	0.00398281	0.324072	1.22769	673.459	0.865731
16	S	0.611741	0.00446376	3.38101	0.542907	1160.67	0.912049
17	Cl	1.15307	0.00447333	4.20464	0.538231	1301.13	0.902702
18	Ar	1.70408	0.00447609	2.75072	0.583835	2713.93	0.953183
19	K	2.60265	0.00432185	1.56811	0.864656	556.393	0.84475
20	Ca	0.536555	0.00440846	3.22706	0.602927	500.851	0.83345
21	Sc	1.7187	0.00436934	4.60285	0.516212	1601.6	0.914664
22	Ti	0.146403	0.00484558	5.54014	0.455903	1763.46	0.919014
23	V	4.21238	0.0040013	3.79858	0.482892	2416.97	0.936204
24	Cr	0.697881	0.00430794	2.2863	0.784069	967.872	0.857874
25	Mn	0.592743	0.00431334	1.40625	0.968261	1426.42	0.881488

When a new parameters file are loaded using this tool, ERYA will guess the correct version from the 1977 and 1991 Ziegler models, by counting the number of nonzero columns, but the user can override the program own guess by selecting the desired version from the pull-down menu located above the spreadsheet editor.

Using the Custom Ziegler's Functions

Using the same pull-down menu, it is possible to activate the custom algebraic function, as long exists a macro program on the respective input-box. When selected, once the user try to validate with the "OK" button, ERYA will trigger the same syntax check like as on Detector's efficiency widget, and will display the first error found during the tests.

The overall syntax rules on Ziegler's Parameters are equal to the ones on Detector's efficiency, and the only main difference is a third obligatory command called **fnvar** that create a sequence of 16 variables, instead of one. The first variable 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, and the final variable with *13* suffix will get the Atomic Mass. The last two variables are only used on the ERYA-Profiling version of this software, and here it only give a fixed "1" value.

As a final reference about the algebraic function macros, it is possible to define functions with limited domain using the relational operators (< , >), define each function as the following chain of sequences:

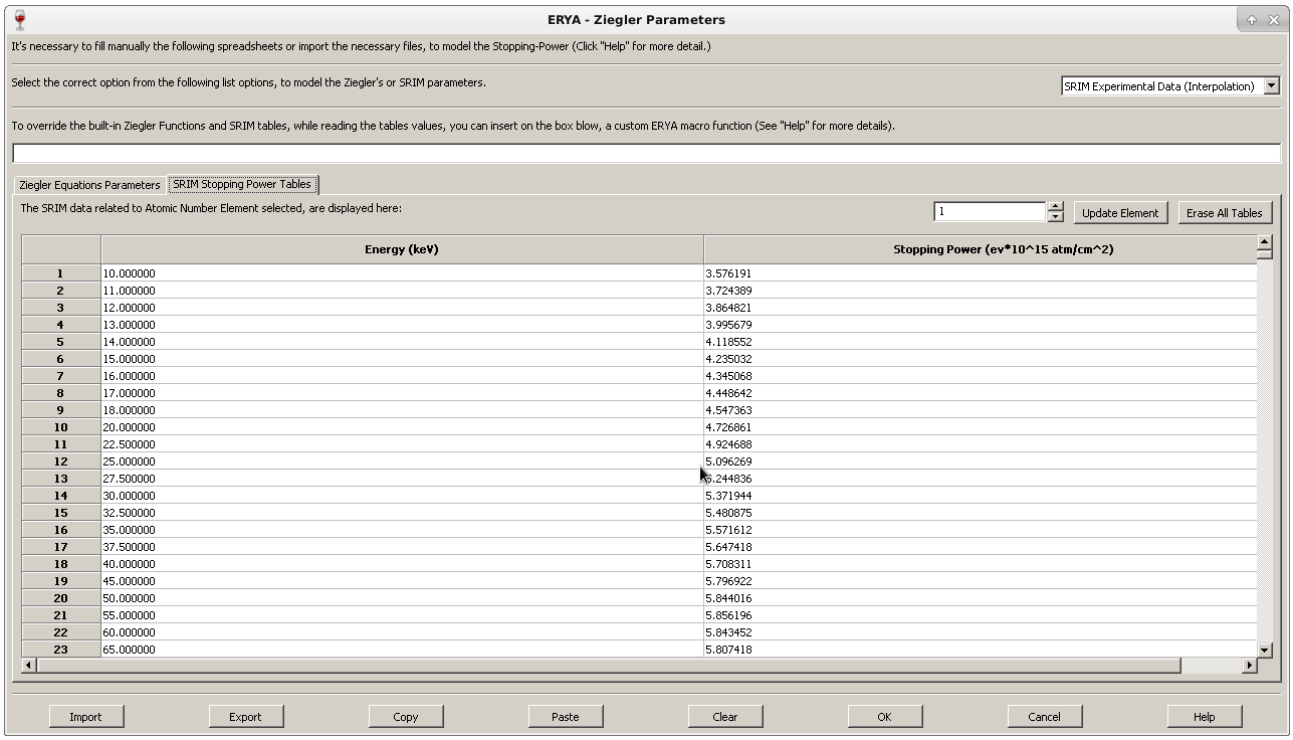
(x < [x_a]) * ([Any function]) + (x > [x_a]) * (x < [x_b]) * … *([Any function]) + …

Since the interpreter only recognizes floating point numbers, any relational evaluation inside parenthesis will gives "0" to a false statement, and "1" to a true statement. It is highly recommended to use single statement formulas inside inside closed parenthesis, in order to give only a "0" or "1" value. Multiple chains such as -3 < x < 5, should be implemented as (x > -3) * (x < 5) * …, and the product sign (*) should be applied to each individual logic statements. The partial function should be enclosed inside parenthesis, and linked with a product sign (*).

A complete chain of partial functions with limited domain should be linked with the plus (+) sign.

Using the SRIM Stopping-Power Tables

To manage the SRIM tables, it is first necessary to select the pull-down menu to “SRIM”, and select the “SRIM” tables to edit the chain of spreadsheets in a fashion similar to the Detector’s Efficiency.



The only main difference on SRIM tables is that is actually an array of tables, and the editor only display a single of them on the SRIM’s Spreadsheet editor. In order to display other tables, just change the numerical value on the switch-box that corresponds to the Elements Atomic Number.

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

The native XML based Ziegler’s Parameters file used by ERYA can store all SRIM tables, Ziegler’s Parameters, and the current model settings in a single file, with **epsz** extension.

Import and Export Ziegler's Parameters to different formats

The native Ziegler Parameters file is a XML file with **epsz** extension. If the user desires, it is possible, like the Detector widget, to import and export the current built-in table content to a text file or an Excel file, but there's some caveats that should be take attention.

Due to compatibility reasons, the "Element" column on the Ziegler's tab are a free user editable zone, since it can be a simple number (like the atomic number) or their Element name.

When it was converted to the ASCII file format, any custom "Element" name are replaced to their row number. In reverse, it renames to the chemical name, since ERYA contains an hard-coded dictionary of all elements.

The ASCII file format exportation will read the active spreadsheet, that can be the one of SRIM tables (when the SRIM tab is active), or the Ziegler's tab. When import a raw numerical data as ASCII, the user should select the Ziegler's or the correct atomic number's on the SRIM table first **before importing the file**, or it could get an import error, since the software had strict import and export rules.

The same caution are applied to export and import Excel Xlsx files, as ERYA requires to know manually the element's atomic number if the source file corresponds to a SRIM table. When the user import or export an Excel file with Ziegler's Parameters, the Elements names are also stored on Excel files.

It is also possible to read the original LabView ERYA's Ziegler Parameters file, both in text and binary form, as long it respects the expected format. The ERYA when the user click "Import" and select "ASCII txt" from the file dialog, as long it is selected the "Ziegler's" tab forehand, it will check if the source file are an ASCII or a LabView DataLog file automatically, and convert the data in order to display on the built-in editor.

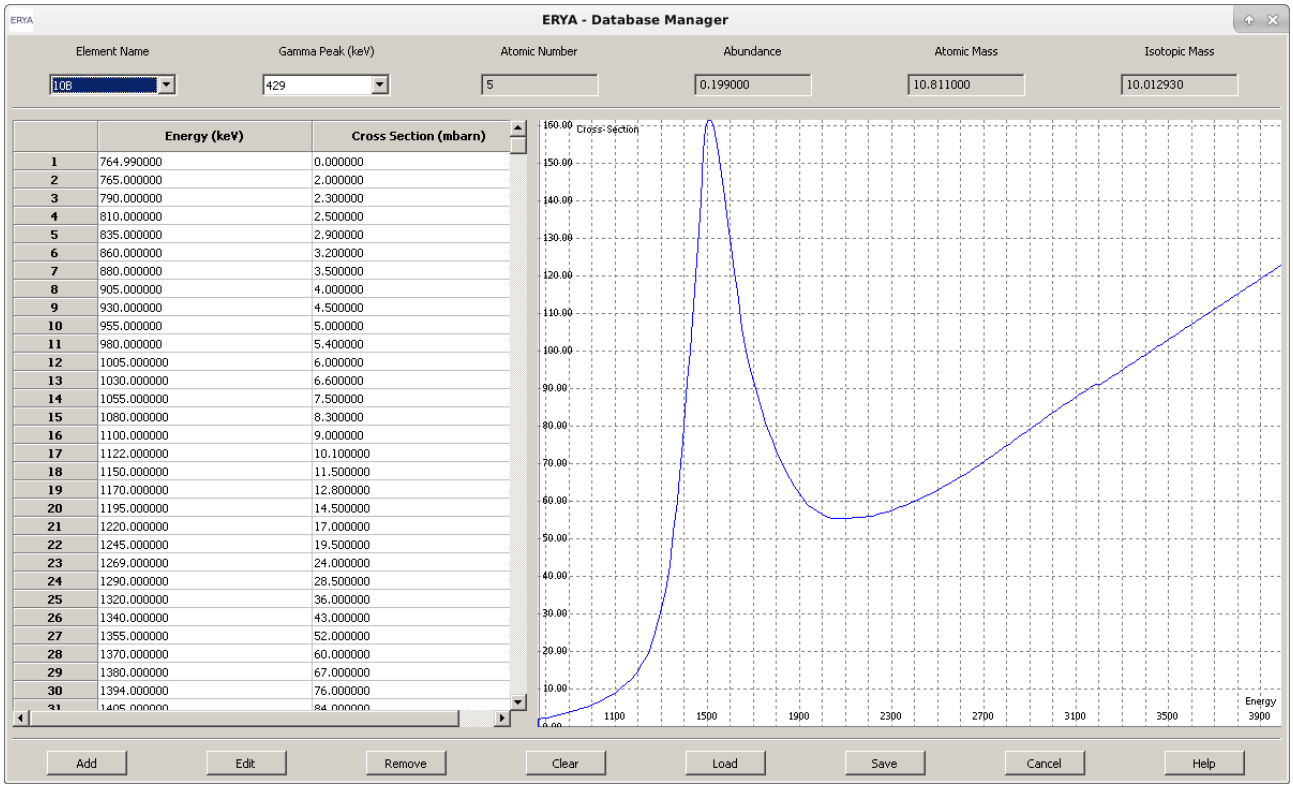
The same applies to unedited SRIM Stopping-Power output files, that is plain ASCII text files, as long corresponds to single elements tables related to proton beams. When the Import button are activated with ASCII import, and ERYA detects that is a SRIM table, it will convert to the correct Element's atomic number table on the SRIM tab without needing additional steps, even if the user select manually the correct atomic number.

Further documentation can be readied from the on-line help widget, by clicking the "Help" button.

Element Database

Apart of the Detector and Ziegler profile files, the real ordeal are the Element Database, which should manage a collection of Elements, along the necessary parameters to handle the numerical calculations.

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



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

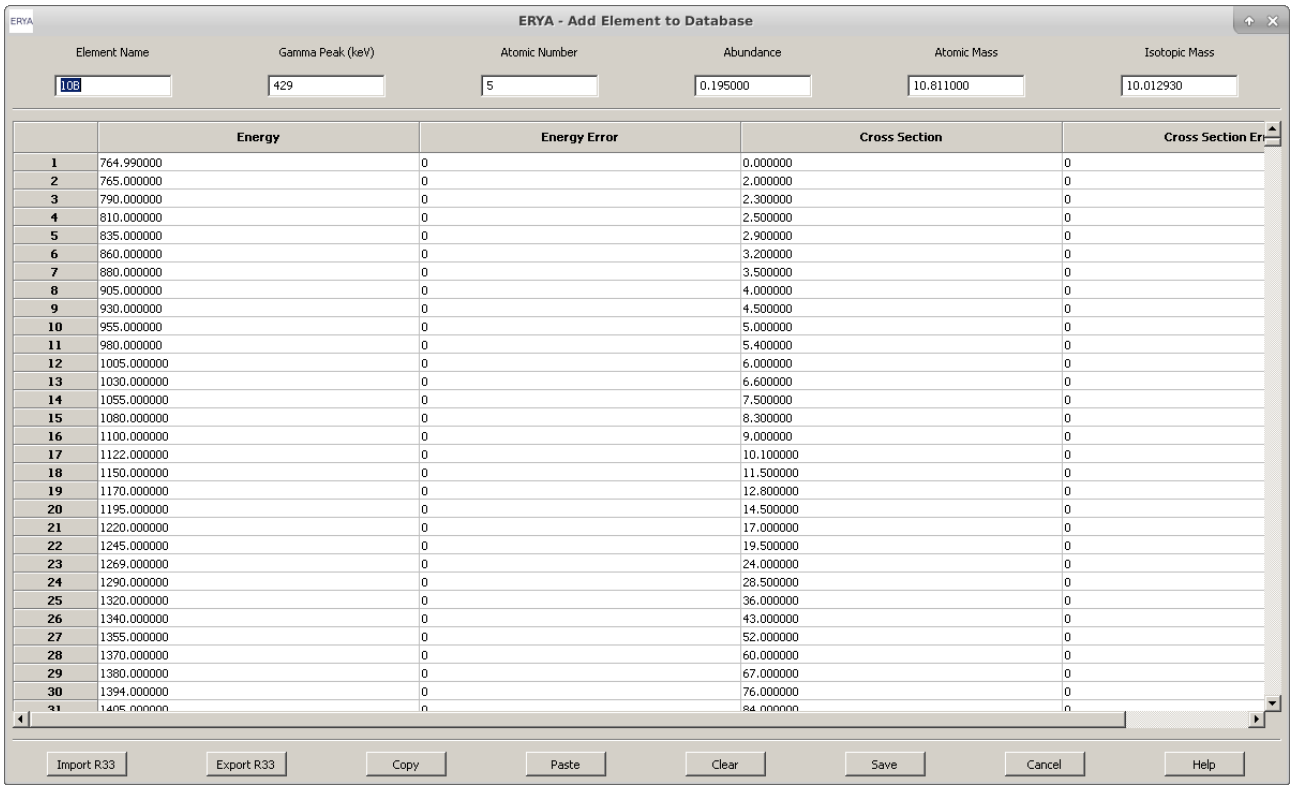
By clicking on “Element Name” pull-down box, the selected element/isotope will refresh the “Gamma Peak” pull-down box, and then the user can select the available gamma emission peak form the same element. For each Element/Gamma pair, a read-only Energy/Cross-Section table and graphical plot will be drawn. Up to four global element physical proprieties will also be displayed.

Editing the Elements Database

The real interest here, is the possibility to add or edit elements to the database.

Selecting any element from the list, then the select and displayed Element can be removed (click on “Remove”), or edited (“Edit”).

It will open a new dialog, and now the Element contents can be changed:



Note: The same widget are displayed when the user click on “Add”, but it will open a blanked dialog.

The other Database Manager available controls is the option that can delete the entire Database from memory (“Clear”), or load(“Load”) from and save(“Save”) to a file which should contains the source database related to the Element Database.

Compatibility with LabView Binary formats

Like the other essential files, the native ERYA Database is a XML file with extension **epd.**

ERYA also supports other file formats for compatibility reasons.

ERYA have a read-only support to export the original LabView Database on binary form, and have a good compatibility to read the LabView ASCII Source Database files. In general ERYA also write to the ASCII format, although it splits any elements with different gamma emission values into separated registers.

Either way, the LabView support is mainly a legacy feature intended to ease the import of binary databases, and currently it is expected to work without major issues.

However it is not possible to use the LabView format as a default Database, simply because ERYA don’t support the rules to write a file of this particular binary file format.

When the user choose to save a new or edited Database, ERYA will optimize the database structure and write a new copy, or overwrite it, to the user selected file. Once completed, the changed Database are reloaded to the main memory.

Additional details can be obtained from the on-line help system, by clicking “Help”.

Adding/Editing new Elements

When the user wants to edit or create a new element from scratch, using the cited controls from “Database Manager”, it should opens the new Element Editor.

Any element should have a name and a gamma peak emission value. Also the user should type their Atomic Number, Abundance and the Atomic and Isotopic Mass.

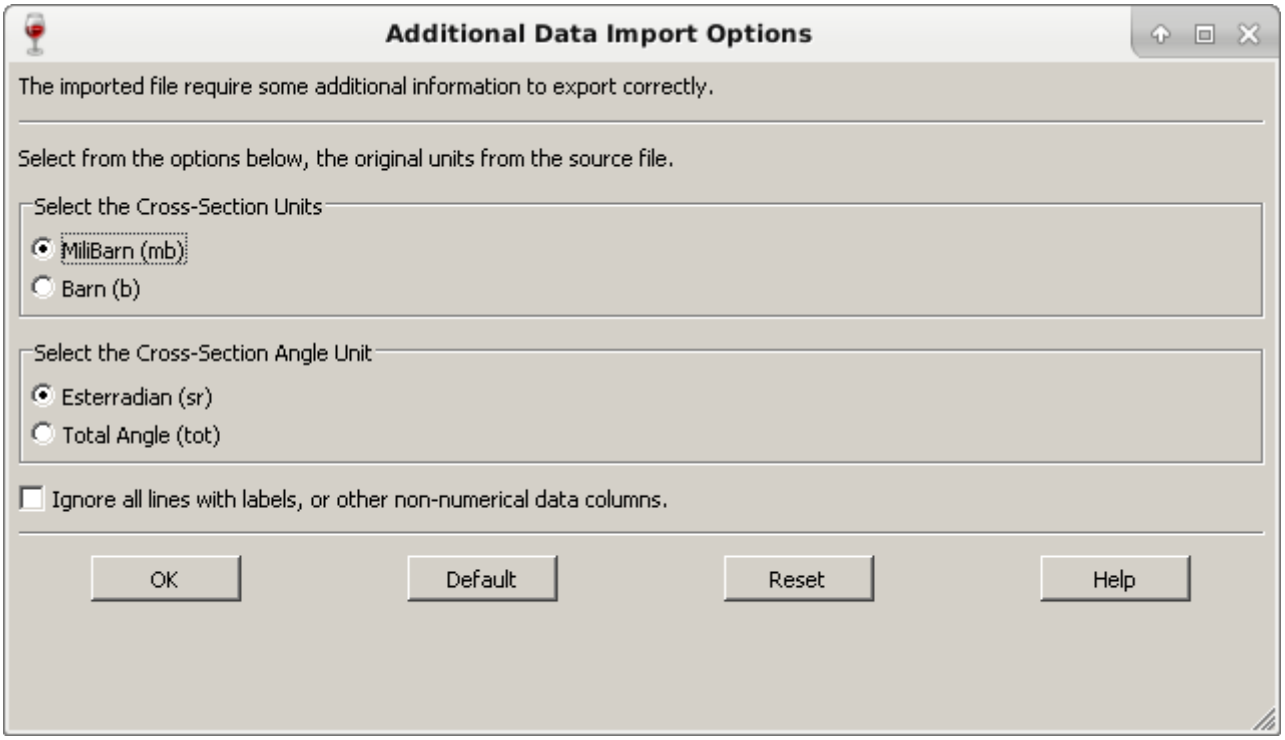
The element cross-section should could now be typed, by placing the energy/sigma pairs, and their experimental errors. The default units are the Energy in keV, and the cross-section in milibarn.

Normally, the cross-section are experimental values stored on several file sources, and it is very rare to manually type on the built-in sigma table editor.

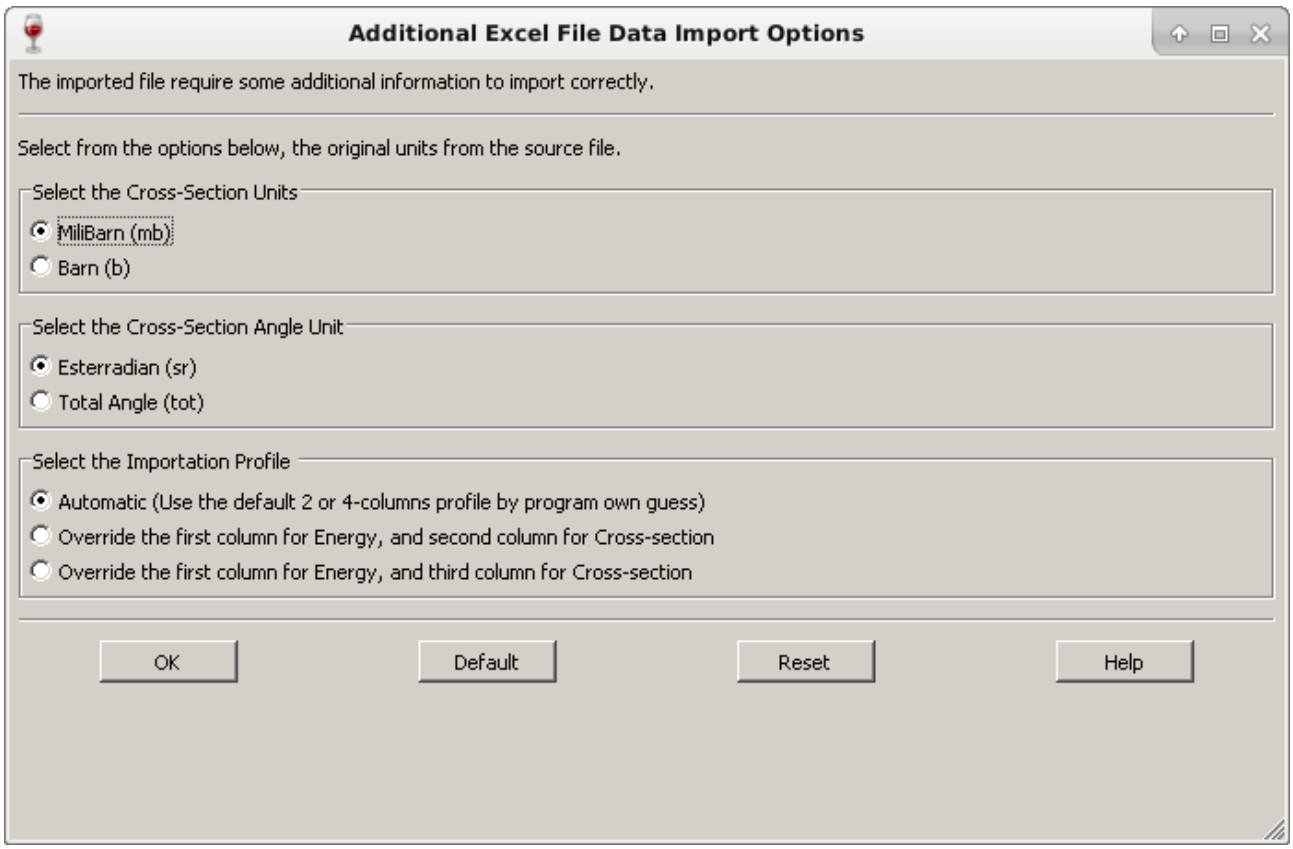
It is possible to export the cross-section from a r33 file from IBANDL (where also fills the Element Name, and the Gamma Emission), from a ASCII file (usually in two-columns text files, but the cross-section are in barn.keV), and even from an Excel file (As long had a xlsx extension, and can have two or four columns, grouping the IBANDL and ASCII profiles.)

To ease the conversion between units, when ERYA load an ASCII or Excel file, an additional window will open to the user select the original cross-section units, and aid the parsing.

On ASCII files, as long they had two columns of numerical data, the user can select the cross-section units, and if the values are differential or integration cross-sections. An optional check-box to ignore parsing errors will ignore any invalid lines, but won't extract any real data.



The same unit conversion logic applies to source Excel files, but it is possible to aid ERYA to tell if the table are intended to be with two or four columns, specially if the automatic guess gives erroneous imports.



When the user save the cross-section to an Excel file, the program will asks if the user want a file with two or four columns, before starts the actual export.

Any ASCII file export will convert back to barn units, and store only two columns.

Either way, when the user export or imports to ASCII or Excel files, no reference of Elements or it's six physical characteristics are stored.

It is possible to save the current edited Element as an IBANDL r33 file, but it will assume always an inelastic proton collision, and store some elements characteristics, but the six elements parameters are stored on the Comments fields, that ERYA will ignore them.

A Basic Example: Magnesium-26

To explain the basic work-flow of create or editing Database's Elements, we will show how to create a new element using an IBANDL file, and make a backup to a file, if you need to do so.

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

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

In this example save two or three samples with the highest number of cross-section points, where the target are the magnesium-26 isotope.

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.

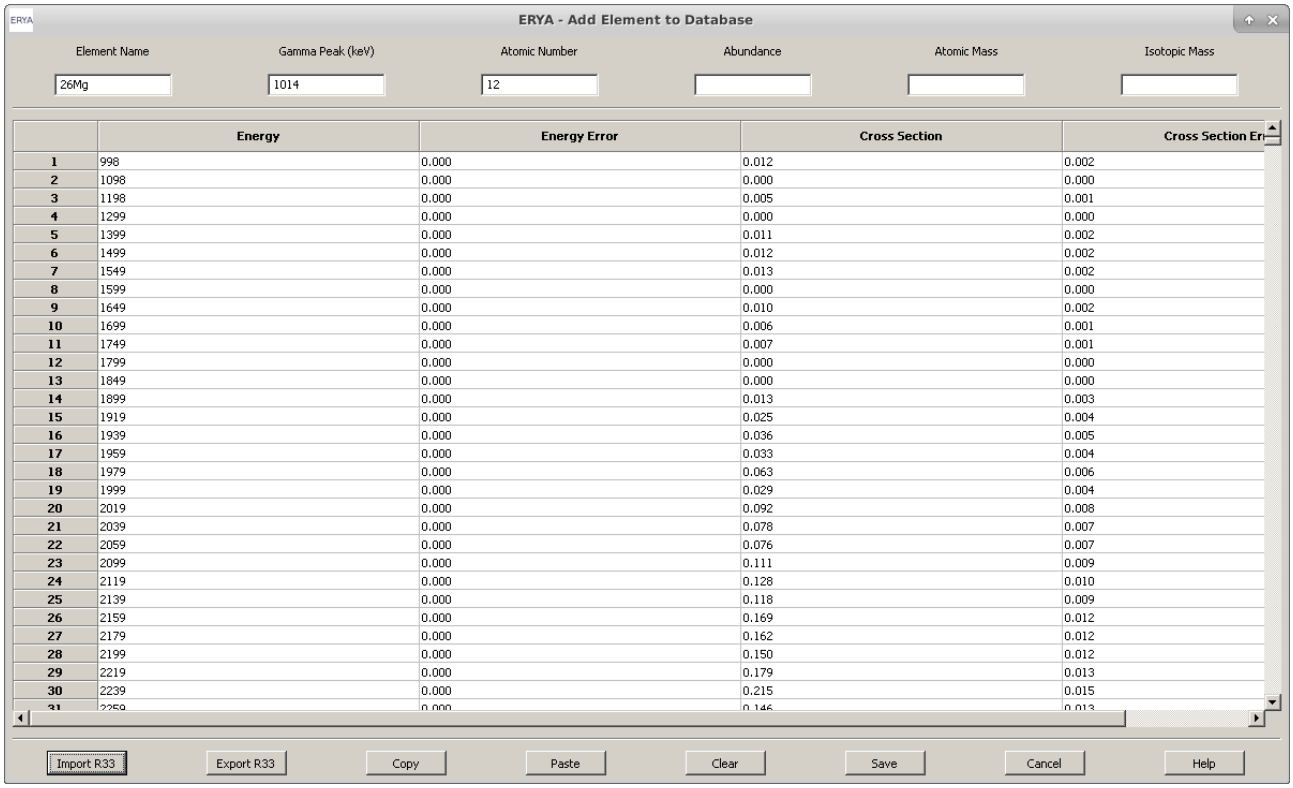
5. Select the relevant r33 files, from the "Import R33" button.

The ERYA will only display r33 files, if you select "IBANDL" files from the standard dialog window.

Select the adequate r33 file, and click "OK". The program may ask to delete the main physical parameters located below the top window, and choose "Yes", anyways.

And it will fill the cross-section and some of the element's parameters.

On this example, the widget fills the following data, as display 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 (even today the physical basic characteristics of several isotopes are available freely), 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 Database Manager.

8. Repeat the steps 4 to 7 to add another entries.

9. Once return to the main Database Manager, click on “Save”, and select the current default Database file, if you want to make permanent changes, or choose another name, if you want to keep the custom database separated. (And save potential data losses!)

10. The new elements are now available to use.

Additional Tasks:

When you successfully complete a new element profile, at step 6, you can click on “Export R33” to save the current data by the following three file formats:

1. Create a backup IBANDL file, while writes on “Comments” section the ERYA references, and the stored Elements physical characteristics.

The ERYA will assume, in some fields, that the cross-section represents an inelastic proton-nuclide collision, without changing the nuclear species, which can be incorrect. To fix that unaccuracy it requires a additional manual editions of the resultant r33 file.

2. When the user choose to export as an ASCII file, only the cross-section are stored, and converted the cross-section column to barn.keV.

Every element name or physical references will be lost, and should choose this option only for compatibility purposes.

3. It is possible to store the cross-section table as an Excel file, where the user can choose a four-column version, or a two-column version where in the last option the cross-section are converted to barn.keV. Like on previous text option, no additional information beyond the cross-section data are actually stored.

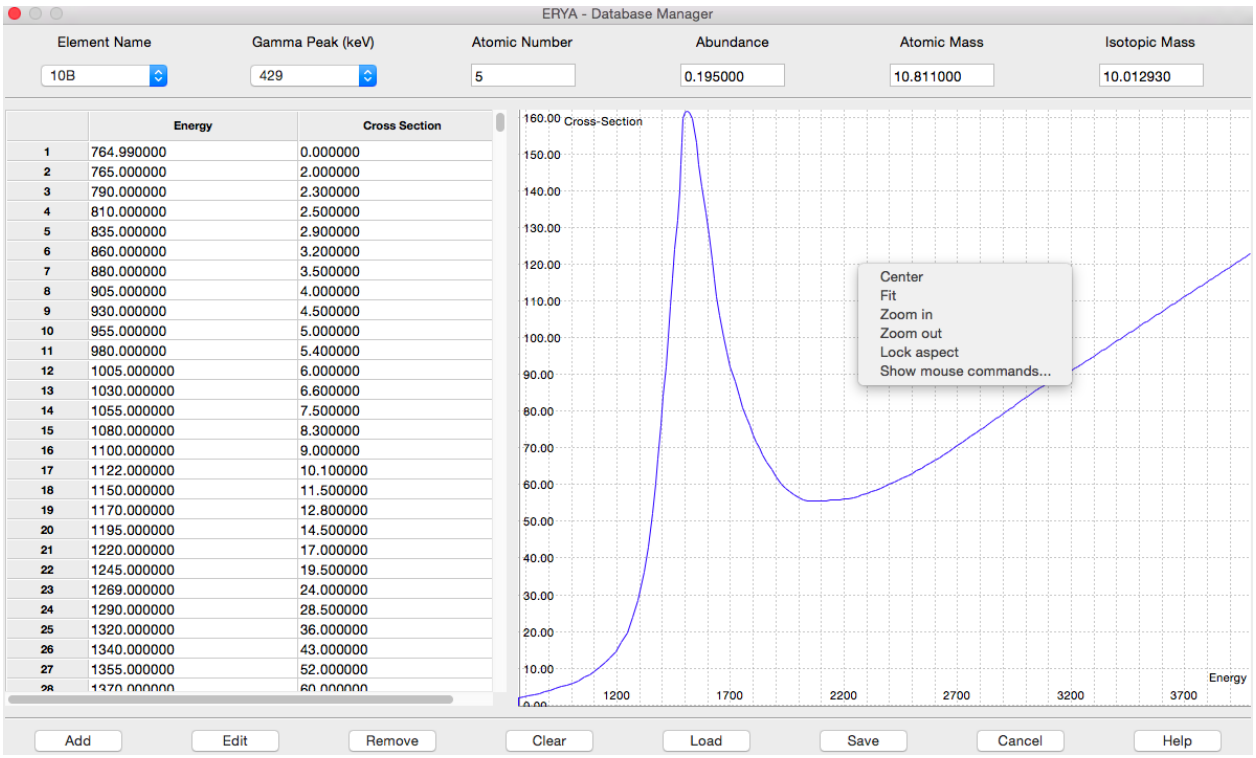
The Excel file export is more friendly for further data manipulation

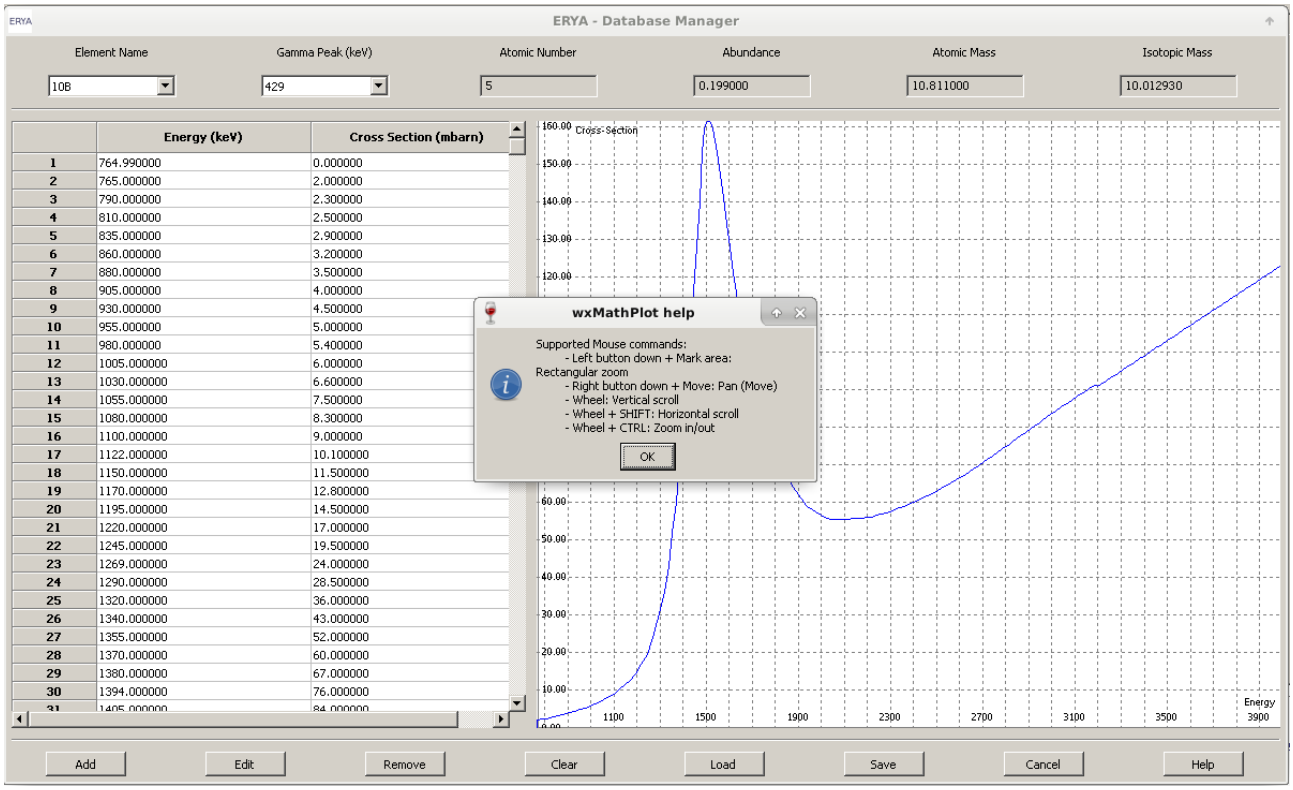
4. You can the graphic plot feature to zoom, fit or select a portion of the graphic by your convenience. The main commands are available by right-clicking the mouse over the plot.

You can select and drag the zoom window, while hold the left mouse button, in order to magnify the selected portion of the graphic.

Note: The graphics plotting library is derived from the third-party wxMathPlot for wxWidgets under GPLv3 license.

Screen-shots of graphics menu tools:





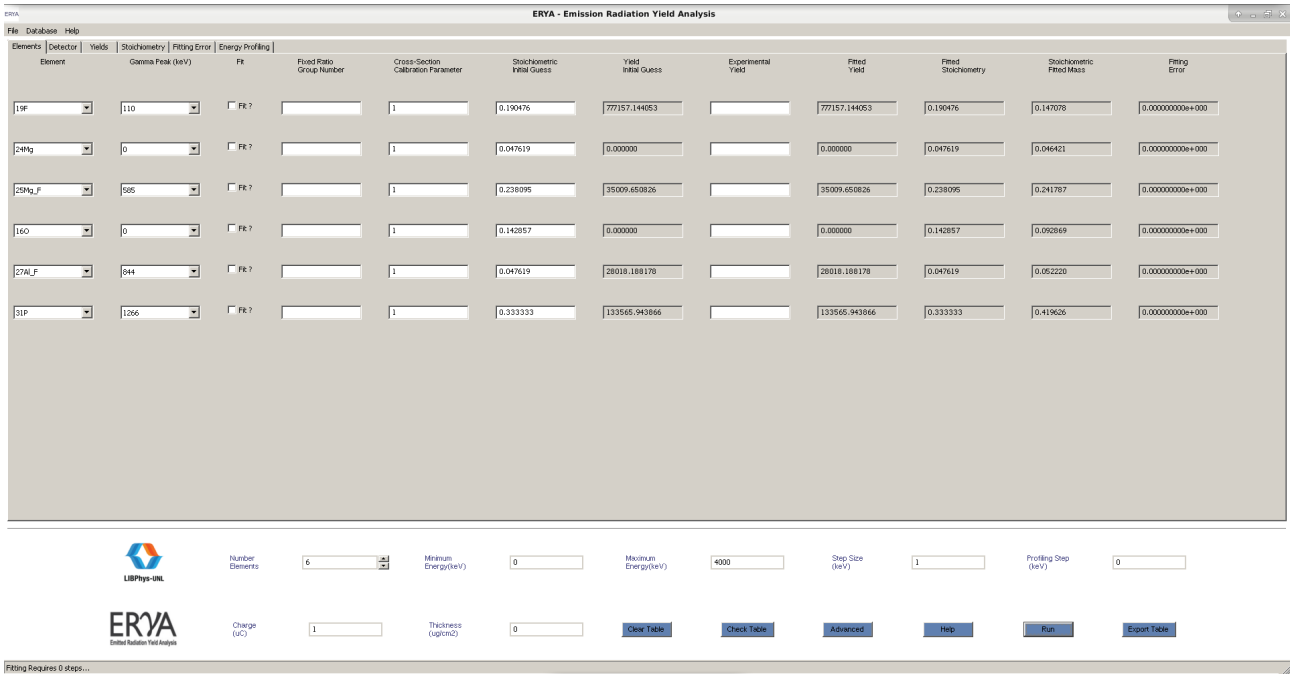
Nuclear Reactions Group, LibPhys-UNL (2018)
Departamento de Física da FCT/UTL, Lisboa

How to make a Stoichiometric Analysis

The main purpose of ERYA is to make an analytical analyses of a bulk sample, in order to determine their components in terms of their stoichiometries. In order to work, it requires a functional Database of several elements, a well-defined Detector efficiency profile and the Ziegler's Stopping-Powers parameters. The previous chapters handles about the Databases, and now the main task is to make the program working.

How to define your Sample

When the program starts, and then the Databases are checked to make ready, in order to make an analysis, the user should define and fill the necessary number of elements and their initial parameters, as like this screen:



1. Select the number of elements, which will fill the corresponding number of table lines, where the user should first select all Element Names from the first column of pull-down menus, then select from their Gamma Emission, the correct values on the second columns.

2. If you sample have a defined chemical composition, then you should assign an unique positive integer on Number Group column, for each element belonging to the same 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 will make by special purposes an independent element.

3. The Stoichiometry column will be automatically renormalized by the program, so you must take attention of this part. If you not fill any number, the program will assume the default “1” value.

It is also possible to fill an algebraic expression on that column, following the same syntax rules of the ERYA macro language. In this case, it requires a normal arithmetic expression like the following examples:

1 – 0.2 **[0.8]**

1/5 **[0.2]**

1 – (1 / 2 + 1 / 3) **[0.166667]**

Or even with symbolic variables, as long the final arithmetic expression are the last block of code before the rightmost colon (:), as this simple example:

a = 1 , b = 5 : 1 – a / b **[0.8]**

ERYA will evaluate all algebraic expressions, and then make the renormalization, where it is simply to sum all numerical values of the columns, and divide all of them by this value.

4. Another important column is the cross-section calibration parameter, which default is “1”.

If your experimental setup requires corrections due to different calibration sources, change the values for this column in order to adjust the expected yield with your experimental values.

5. If you intended to made a fitting, you will need to fill the experimental yields, as it will be described next section of this guide.

6. The bottom controls expects to fill the energy range (maximum, minimum and integration step energy), electrical charge (on micro-columb: μC), and optionally the sample thickness in micro-grams per square centimeter ($\mu g/cm^2$).

Note: if you want to use a non-thin sample, fill the sample’s thickness value as zero.

7. Optionally, define a non-zero value on “Profile Step”, to create an additional table (on the last tab of the main screen) to display a list of partials yields between the minimum and maximum energy, along the user selected energy step. This table displays always the results after a successful fit, or the initial stoichiometric values if the user don’t mark any fitting at all. (A fit will be trigger if exists at least a single “Fit” checkbox mark.)

This feature is intended to reduce the gap between this program and the more powerful ERYA Profiling.

7. Now click “Run” to get the theoretical yields, where it will appear on the adequate columns of the main table, and also on four additional tabs filled with graphics.

Calibration Parameters

The calibration parameters feature was also present on LabView ERYA and serve as a correction factor to the adjustment between the theoretical yields and the experimental ones, due to the inevitable presence of systematic measurement errors, and the physical uncertainties related to several physical quantities like collected charge, cross-section measurement errors, and so on.

To determine the adequate corrections on the simulation it is required to make measurements with calibration samples, where the composition are well determined and documented. Measuring the simulated yields against the ones derived from the calibration sample, the ratio between them is the calibration parameter, and should take note on it.

Notice that this option is only meaningful if the experimental setup conditions are not changed during all measurements.

How to Fit experimental results

A last program resource is the possibility to fit experimental yields to adjust the stoichiometric composition of the sample. To make an stoichiometric fitting, apply the steps described on last section, and now fill the “Experimental Yield” column with all experimental data you obtain.

1. Once filled the “Experimental Yield” column, also click of their respective “Fit” check box, to mark the element as a fitting parameter.

2. Elements grouped as Compounds, which have the same Group Number, will be treated as a single fitting variable, and then their relative stoichiometric ratios between the selected elements of the same Compound (as long they have both the “Fit” enabled) will not change.

Except the initial global renormalization applied to all elements, done to ensure the sum of all relative stoichimetries are equal to “1”. All elements that don’t had the “Fit” checkbox, their stoichiometries will never change.

3. To produce a profiling table once the fitting are done, set a different value for the Profiling Step than “0”. This table are the evaluating of the yields from the initial value to the maximum value, selected by the user, along the equally spaced interval defined by this setting.

4. Now click on “Run” to start.

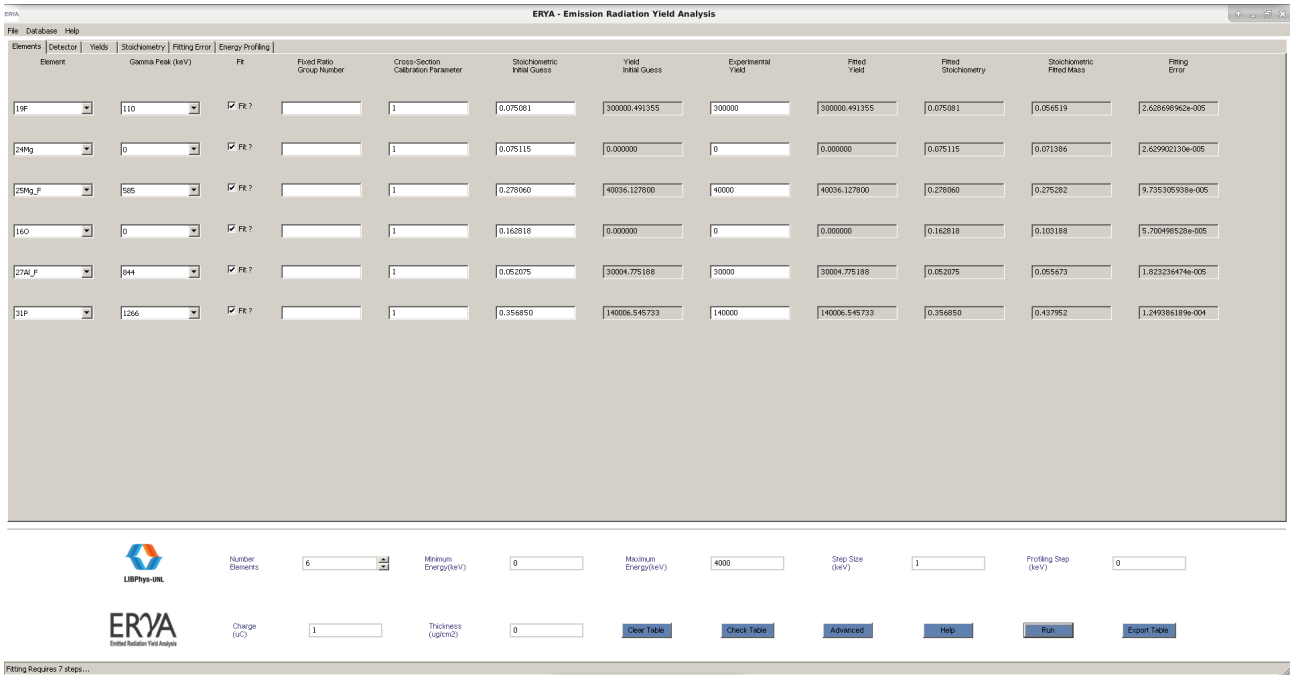
A small window with a progress bar will appear, and displays the current number of steps during the fitting procedure. The total steps is the maximum number of steps defined by the user (It requires to change the fitting values, as explained on “Advanced” section), before quits, not the predicted number of steps required to achieve the pretended solution.

5. If applicable, once the fitting process ends, the profiling procedure begins with their own window dialog. The progress bar quantity is actually the relative value of all evaluated partial integrals along the sample.

6. Once completed, it asks if you wanted to copy the fitted stoichiometry to the initial guess.

Choose the option that most benefits on your context.

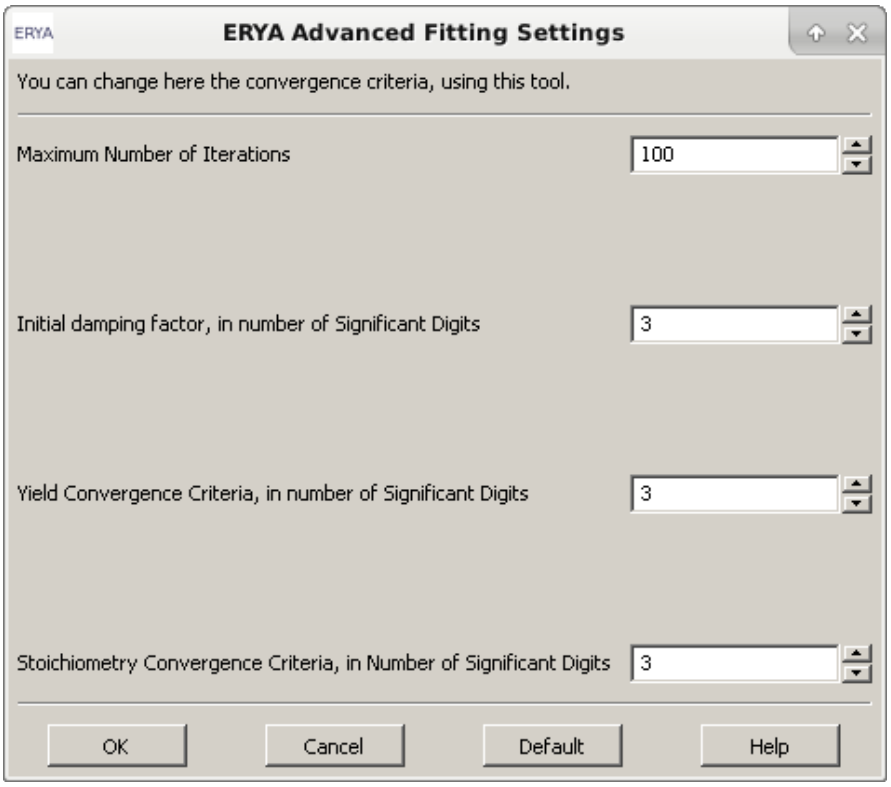
The main windows screen, when completes all tasks, should display all relevant numeric data:



Note: The program applies the Levenberg-Marquardt algorithm to do the fitting procedure, and the fitting computation times depends of the number of fittings parameters, initial values and additional fitting settings. Normally, it takes few seconds for simple examples, and takes a little more time for larger samples. At the bottom status bar, it also displays the number of iterations required to achieve the best fit possible.

Changing Fitting Settings (Optional!)

If the user knows what are doing, it is possible to change the fitting convergence criteria by clicking the “Advance” button on the bottom main page controls, which opens a simple widget:



Where the user can set the maximum number of iterations before aborts (up to 100, which is also the default value), and the other setting are integer values that represent the number of significant digits of precision.

The first is a damping gradient norm, where can range from 0 to 3 digits (default is 3),

The second is the relative yield convergence criteria, ranging from 0 to 6 digits (default is 3),

And the third is the relative stoichiometry convergence criteria, ranging from 0 to 6 digits (default is 3).

Changing the default values in order to reduce precision can avoid unnecessary computation time for poor samples, but increasing the precision could increase the computation time without gain much benefit.

Saving Results

Once you got the results of the numerical simulation, it's possible to save it as a file, where the preferred file format is an Excel file.

Just click the "Export Table" button, give a name and directory location, and it is done!

A XML file output format alternative , with extension **epz**, are accessible from File > Save As.

Opening output files on ERYA are only possible for native **epz** files, and also will ask to reevaluate the the results in the same way as the manual user input and the fitting procedure.

You need Microsoft Excel 2007 or greater version to open the ERYA 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.

Note: The profiling data are also stored on **epz** and **xlsx** files. The main advantage of **epz** files is that is possible to read the file by ERYA (File>Open) and edit the contents.

Warning: If you need to use Excel files as an import source on ERYA (when it is possible), make sure that the source Excel file is a **xlsx** file. ERYA also will read only the first sheet, and all data must be packed together as a single matrix block. (Additional requirements are described on Database Management Tools chapters, and on on-line help documentation.)

Example: A 7LiF:Na sample embedded on Ag

On this simple exercise, let's create a sample made with 1% lithium, 1% fluoride, where belongs to a single LiF compound, blended with 18% sodium. All three elements are mixed with a silver subtract (up to 80%).

Based on PIGE spectrum analysis, suppose that the lithium yield are 8500, the fluoride are 13500, and sodium are 200000, rounded. The energy beam was up to 3 MeV. From this results, determine the rectified sample composition.

First Step:

Open ERYA, and define four elements. Select "7Li", "F", "Na" and "Ag". (The gamma emissions are only one on initial Databases, so skip details for now.).

Since lithium and fluoride belongs to a single compound, add a number "1" to "7Li" and "F", respective to the "Group Number" column. It is not necessary to assign any number to "Na" or "Ag", since it is an isolated element.

Fill on "Initial Stoichiometry", the relative stoichiometry, which 1 to lithium and fluoride, 18 to sodium, and 80 to silver. The sum is 1+1+18+80=100, so the stoichiometric input will be renormalized by dividing both column entries by 100. (This is done automatically by the program.)

Second Step:

Define on the bottom man page controls, the energy range: "Maximum"=3000, "Minimum"=0, "Step"=1, "Charge"=1, "Sample"=0, "Profile"=0.

Third Step:

Fill on "Experimental Yields" column, the 8500 value to "7Li", 13500 to "F", and 200000 to "Na".

Then mark the "Fit" column, to lithium, fluoride, sodium and silver.

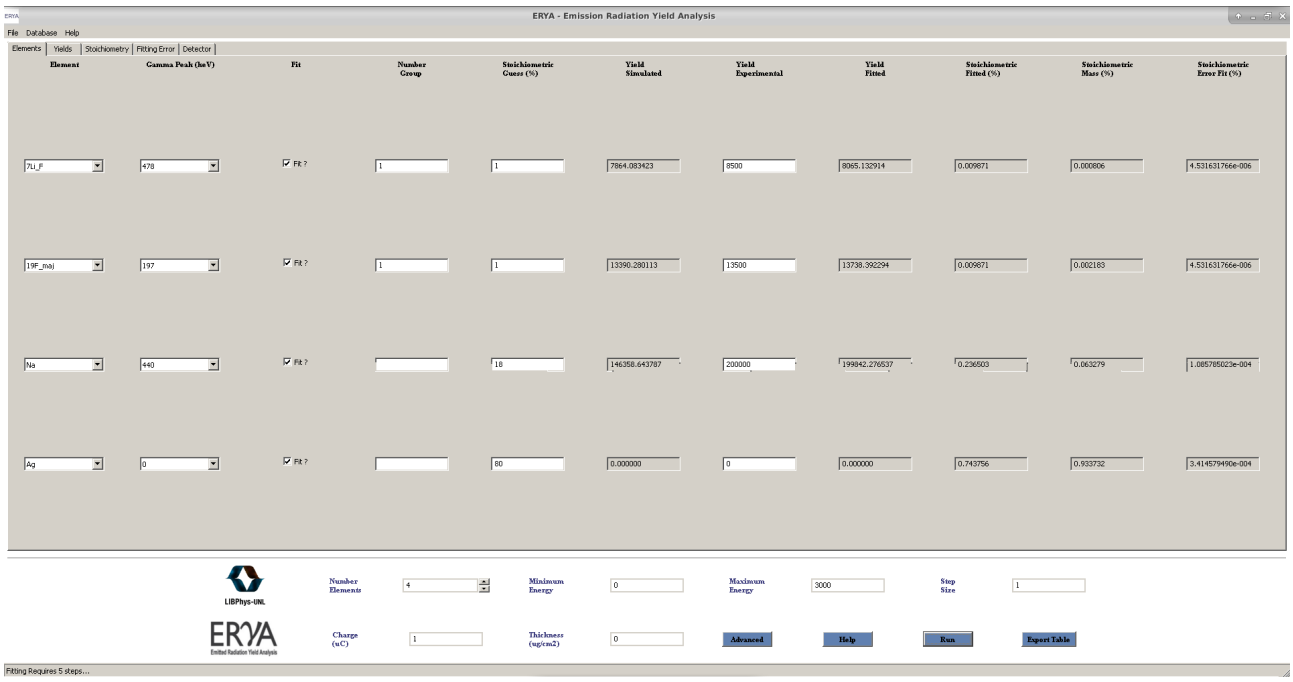
Fourth Step:

Finally, click on "Run" button, and will get the results. Save the results quickly, using the "Export Table" button, which store the data to an Excel file.

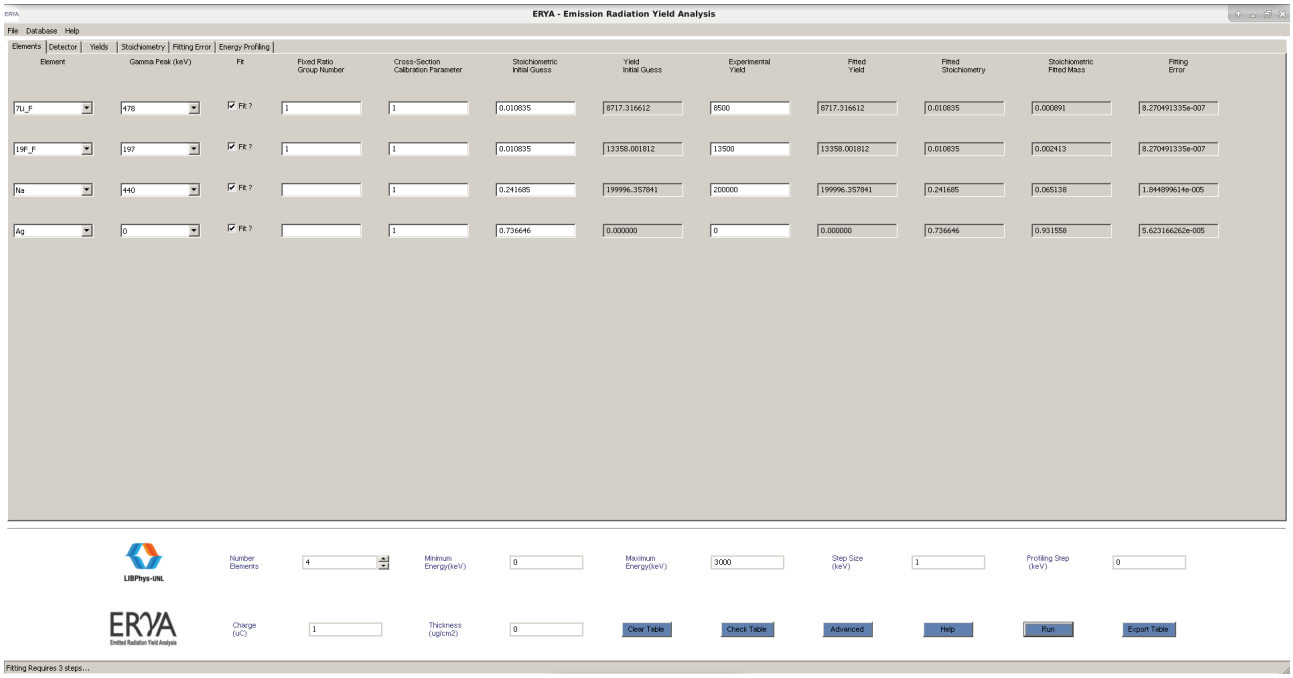
Extra Step:

If you click the "Advanced" button, and change the Damping factor from 3 to 4, we got much more computation time, without real benefit, since it only increases the number of iterations to almost triple without increase the precision.

First try:



Second try:



ERYA take only 3 iterations, but there's not much improvement when change the accuracy settings.

Conclusion: The LiF stoichiometry maintains at 1% each one, the sodium increases to 24.1%, and silver drops to 73.6%.

A more realistic Example: A Holquistite Sample

The Holquistite is a lithium magnesium aluminum inosilicate mineral that contains up to six or eight different elements, making a more complex model and more realistic problem that ERYA should handle.

Suppose our holquistite sample can be modeled by the chemical formula: $Li_2 Mg_3 Al_2 (Si_8 O_{22}) F$

The chemical formula already specify the relative stoichiometric values between the seven elements, and putting that values on the “Relative Stoichiometry” are valid, as ERYA will apply all renormalization automatically, as explained on the previous topics.

Contaminations of additional elements can be evaluated by placing small numbers, at least with two or three orders of magnitude lower that the main elements stoichiometry.

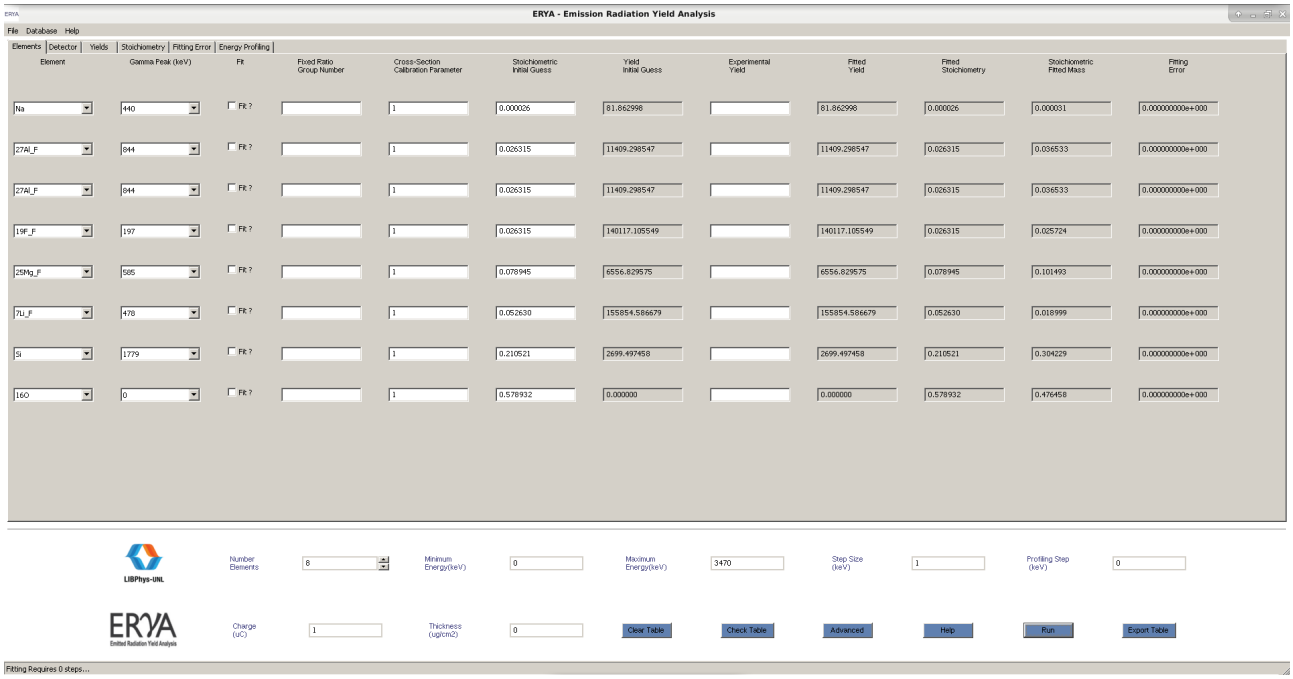
Let’s suppose that the sample was placed on chamber and bombarded by a beam of protons with 3470 keV energy.

The detector computer measured seven different yield peaks, where two of them are two different gamma peaks from the same element, and then compare to the six elements that belongs to the mineral. The oxygen are assumed to be without relevant yield measurement.

A more careful analysis on the PIGE spectrum reveals the presence of a trace amount of sodium.

With all information gathered, we conclude that the simulation should have eight elements.

1. The inputed ERYA window should be one like this example:



The initial yield are defined by the expected relative stoichiometry defined by the holquistite chemical formula. Since the detector distinguishes two different gamma peaks of aluminum, the relative aluminum stoichiometry was halved between the two gamma peaks. The sodium trace was a little guess by placing a small number. (0.001 is fine).

By hitting “Run”, the program redefine a renormalized global stoichiometry straight on the initial step.

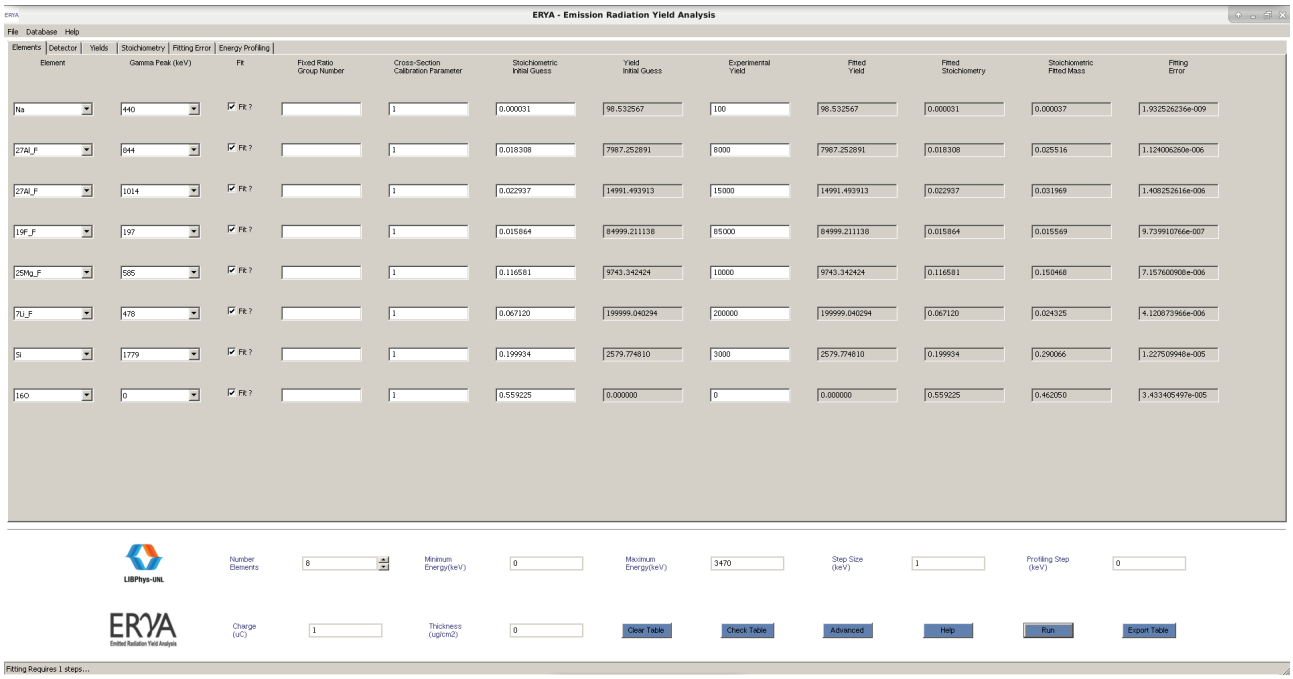
To make the fitting work, it requires to gather the experimental yield of all seven peaks from the PIGE spectrum, normally generated by the computer that controls all experimental apparatus, and place them on the “Experimental Yield” column.

Since the sample may deviate from the expected composition, we assume that all elements are independent.

This means that all elements should be fitted, by selecting the “Fit” checkbox.

Since some Elements such as lithium or fluoride had a theoretical yield discrepancy of up to two magnitude orders, it is a good advice to open the “Advance” button and reduce the stoichiometry and yield precision to two decimal places, and reduce the gradient damping to two decimal places too.

2. Hitting Fit gives a good approximation with only 2 iterations, which is a nice result for 8 elements!

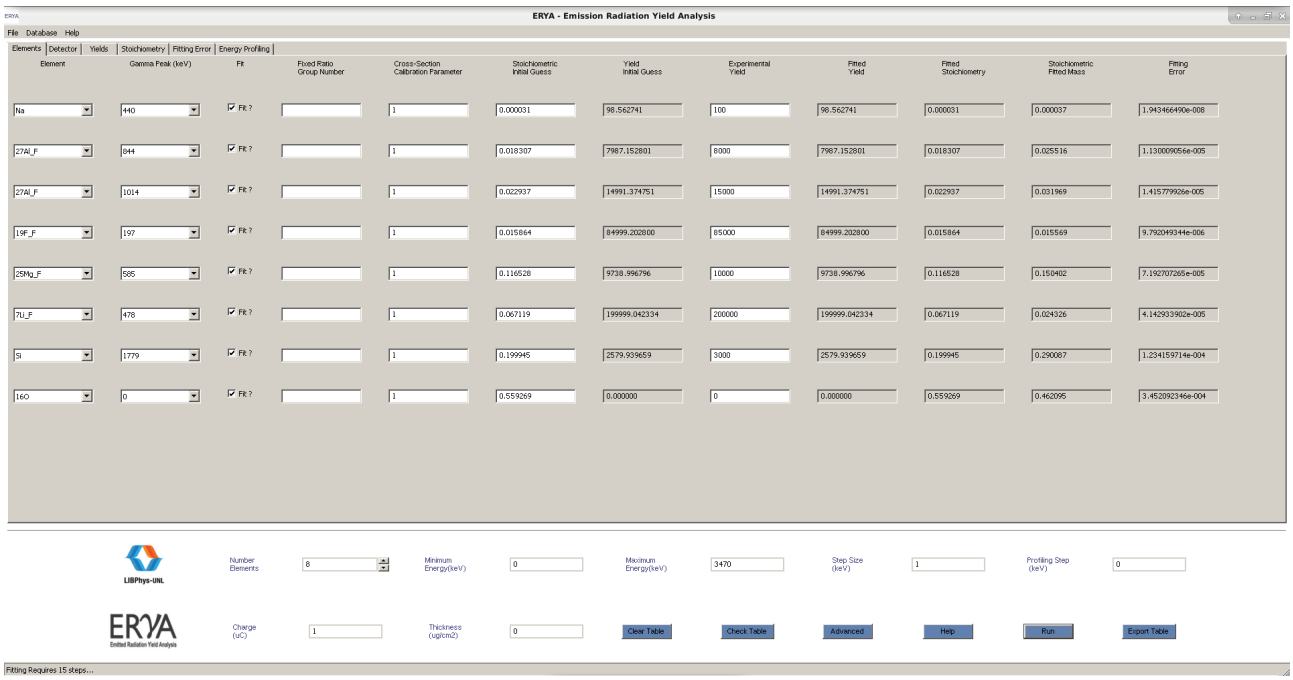


Some elements like magnesium and silicon suggest that are still room for improvement, however the convergence will be much slower, and can require more computation time, where it take up to several seconds, or even a few minutes, depending how fast is the computer processor.

Tip: Since ERYA will ask to copy the fitted Stoichiometric values to the Stoichiometric Guess columns once done each simulation, you an deselected the “Fit” flag for elements which their fitted yields and experimental yields are almost identical. Then made another “Run”, now with less fitting variables, and less computational time.

Once made a chain of successive fits with this trick, remark all fits to made a final fit, that should be almost instantly.

3. When increasing the gradient, yield and stoichiometry precision to 3 decimal places, it gives:



It requires 15 iterations to achieve better precision, and fixes the silicon/oxygen ratio, reveling that on real samples, it should begin a trial measurement with lower precision to check if your sample are beyond the initial expected stoichiometry, before make another fit with greater precision to fix the trial one.

Bibliography

Note: The research materials are the same of the LabView ERYA program.

- 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).

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

- 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/

Additional reference:

- Ananth Ranganathan, “The Levenberg-Marquardt Algorithm”

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

- 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/

- LabView DataLog reverse-engeenering file structure reference: https://sccn.ucsd.edu/svn/software/tags/EEGLAB7_0_2_6beta/external/fieldtrip-20090727/private/read_labview_dtlg.m

- Excel 2007 file documentation: https://www.codeproject.com/Articles/208075/How-to-read-and-write-xlsx-Excel-file-Part-I ; https://www.codeproject.com/Articles/210014/How-to-read-and-write-xlsx-Excel-file-Part-II

Note: All C# and MatLab code was ported to C++, while the Excel file internal structures are implemented using the native wxWidgets classes for XML and ZIP streams.