ERYA Profiling

User Guide & Tutorial

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

Contents

Introducing ERYA Profiling

Welcome!

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

It is the natural evolution of the ERYA-Bulk program, where the main difference is that a more realistic physical model are applied.

The efficiency of the method implemented by this software depends highly of 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 weighted 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 it use the 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, now it gives the results in a matter of seconds.

ERYA-Profiling 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-Profiling was programmed by Vasco Manteigas (vm.manteigas @ campus.fct.unl.pt), with advice from Micaela Fonseca (micaelafonseca @ fct.unl.pt), and Luís de Souto Martins, where first one was the original programmer of the previous LabView ERYA version, and the second one implements the numerical routines of the first version of ERYA-Profiling, bundled on LabView 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 on Heterogeneous Samples

The program evaluate the stoichiometry composition of an heterogeneous sample, make by several layers, where each one composition can be made 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.

On the homogeneous case, each individual yield could be evaluated from the following integral:

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

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 cross-section excitation function (σ) and the stopping power (ε). Each of relevant functions depends from the relative stoichiometry of the sample.

When the sample are heterogeneous, the integral is a little more complicated, since the the terms related to the relative stoichiometry now depends on the depth, or energy, where it turns to be:

$$Y(E_0) = \epsilon_{abs}(E_\gamma)\, n_p N_A A^{-1} \int_0^{E_0} \frac{f_m(E) f_i(E) \sigma(E)}{\epsilon(E)} dE \qquad (2)$$

And a more realistic model of energy dissipation require to take account of the energy loss of the beam. This require to model a distribution of the energy straggling dissipation, and another distribution related to the own detector's physics, where on on-line help is called, by convenience, the thermal distribution.

Since the energy variable can be spitted by the sum:

$$E = \overline{E} - \Delta E \qquad (3)$$

Making a distinction of the average energy $\overline{E}$, and energy dispersion ΔE , the integral in (2), can now be written:

$$Y(E_0) = \epsilon_{abs}(E_\gamma)\, n_p N_A A^{-1} \int_0^{E_0} \int_E^{\infty} \frac{f_m(E) f_i(E) F(\overline{E}) \sigma(\overline{E})}{\epsilon(E)} dE\, \overline{dE} \qquad (4)$$

Where the cross-section on ERYA-Profiling is now an integrable function along the dispersion distribution F. And since the actual distribution is the convolution of a thermal and a straggling one, the integral (4) is now a triple integral:

$$Y(E_0)=\epsilon_{\text{abs}}(E_\gamma)\,n_p\,N_A\,A^{-1}\int_0^{E_0}\int_E^{\infty}\int_{\overline{E}}^{\infty}\frac{f_m(E)f_i(E)\,F_T(\overline{E}')F_S(\overline{E}-\overline{E}')\sigma(\overline{E})}{\epsilon(E)}\,dE\,\overline{dE}\,\overline{dE}' \qquad (5)$$

In order to simplify (5) to be reasonable to implement the numerical integration of the yields, the program make the following simplifications:

- The sample are sliced by several layers with a constant depth *x,* where the stoichiometry can be constant of each element, then the yield of an element is the sum of all partial yields respective to each layer, this replace the energy integration, by a sum of partial yields:

$$Y(E_0)=\sum_{L=1}^{N}Y_L(\overline{E_L}) \qquad (6)$$

And the average energy at the current layer $\overline{E_L}$ are a constant along the main terms of the integral, leaving the integral of the distributions separated, since:

$$Y_L(E_L)=\epsilon_{\text{abs}}(E_\gamma)\,n_p f_m(\overline{E_L})f_i(\overline{E_L})N_A\,A^{-1}\int_{x_0}^{x_1}\int_{y_0}^{y_1}\frac{F_T(x)F_S(y-x)\sigma(\overline{E_L}-y)\overline{\Delta E_L}}{\epsilon(\overline{E_L})}\,dx\,dy \qquad (7)$$

Where the double integral domain $x=[-3\Omega_T,3\Omega_T];\ y=[\lambda_0,\lambda_{0.995}]$ are truncated along the 3-sigma approximation of the relevant distributions.

The thermal distribution is a Gaussian Distribution, truncated at 3-sigma, where it is numerically trivial to evaluate. However, the straggling is modeled by a Vavilov Distribution, which require additional numerical routines to handle properly.

A full discussion of the numerical Vavilov Distribution is out of scope of this manual, but some additional information can be retrieved on on-line help, and a full description are elucidated on the program bibliography.

ERYA profiling evaluate the yields integral by calculating the integral (7) for each step on (6), where the average energy are evaluated by an energy loss step defined by:

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

Where the first layer had the initial incident energy E_0 , and n_L are atomic density of the current layer.

What's New on ERYA Profiling?

This application was a full rewrite of a previous LabView program that handles the PIGE database management, and an additional program that made the numerical profiling of the sample. Actually such program was only a prototype and never released outside of CTN.

ERYA Profiling 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 Profiling (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 Intel or ARM.

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 Profiling 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.

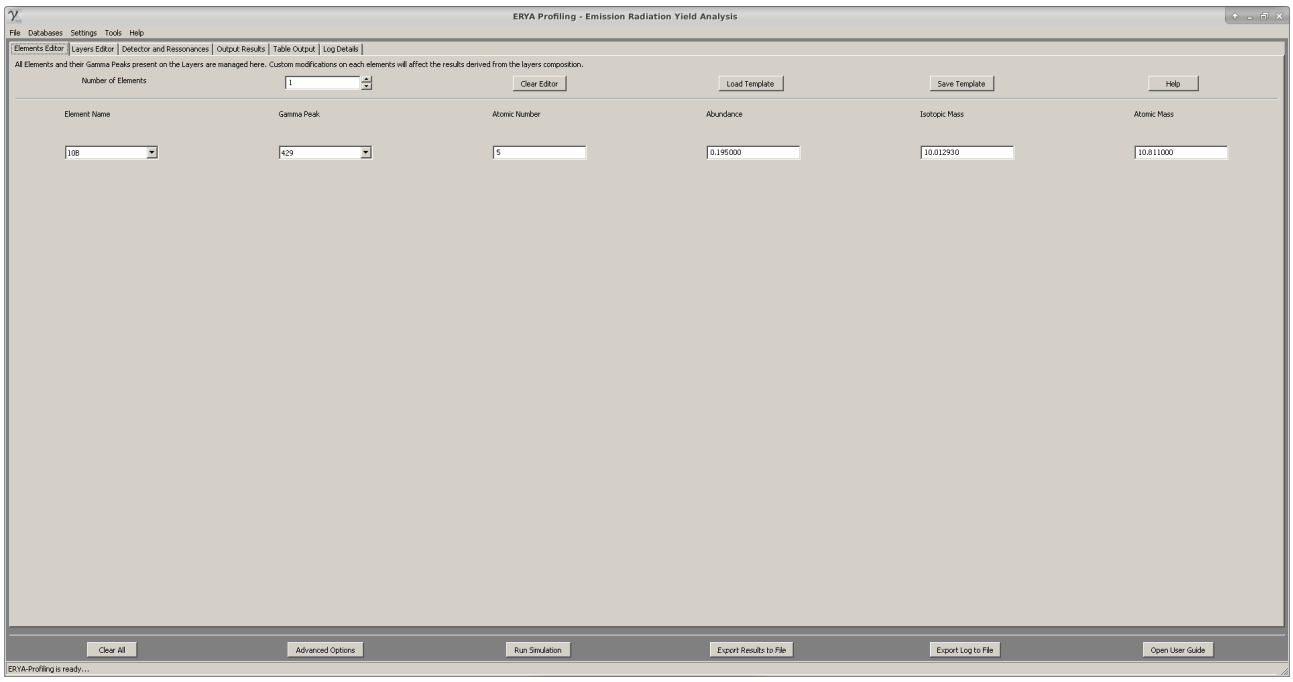
All Linux packages are designed to work with any Debian derived Distribution, as long contains the wxWidgets 3.0.4 libraries updated, and require at least Debian 9 Stretch with backports repositories enabled, or Ubuntu 18.04 LTS versions to work.

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. Even this feature are maintained for legacy support, all conversion features will be described on the *LabView Export* chapter.

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

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



Like the previous ERYA-Bulk companion program, ERYA-Profiling shares the following core features:

1. A new unified Database format for the Elements, Detector and Ziegler Parameters, that use the standard XML document format, and inter-changeable between the wxWidgets ERYA-Bulk and ERYA-Profiling.

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 data, and implement a more concise work-flow. And also simplifies the porting to several operating systems.

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

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. All packages require, by default design, to install on computer's local storage. Normally, the user just 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 to run the application.

Windows: For Windows users, the wxWidgets libraries and GNU C++ Runtime Library are already bundled on the application installation package.

The Windows version package will install ERYA on 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 eryaprofiling_2.40-5_amd64.deb (64-bit Intel)

$ sudo dpkg -i eryaprofiling_2.40-5_i386.deb (32-bit Intel)

$ 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-Profiling.app package to the Applications folder.

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

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 eryaprofiling_2.40-6_armhf.deb (32-bit ARM)

$ sudo dpkg -i eryaprofiling_2.40-6_arm64.deb (64-bit ARM)

$ sudo apt -f install (To install the library dependencies.)

$ 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!

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.

Warning: ERYA was not tested on Linux distributions tailored to perpetual beta staging, like Arch or their derivatives. Also alternative Linux Window Servers replacements like Wayland or Mir was not tested at all, even using a X11 emulation layer.

Starting ERYA-Profiling:

- On Windows, select Start > ERYA-Profiling, and ERYA-Profiling main icon.

(On Windows 10, type ERYA on search-box to grad the launcher quicker)

- On Linux, select the icon from Programs > Education/Science > ERYA-Profiling.

(The real category can vary according to the default's Desktop Manager of your Linux distribution.)

- On Mac OS X, open the Applications folder from Finder, and then find and click on ERYA-Profiling icon to start the program.

Removing ERYA-Profiling:

- On Windows, select the uninstaller icon, or from "Control Panel">"Program and Features", select the ERYA-Profiling group, and follow the instructions.

Open a Command prompt to delete the user profile folder:

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

- On Linux, open a terminal application and type the following command as root:

$ sudo apt-get remove eryaprofiling

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

- On Mac OS X, simply delete the "ERYA-Profiling" program from "Applications" folder on Finder.

Open the Terminal.app to delete the user profile directory:

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

- Consult the table on next "Initial Setup" section, to find and delete the local profile folder, before adapt the commands described above.

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-Profiling-Win64***	**/home/name/.ERYA-Profiling-Linux/***	**/Users/name/Library/Application Support/ERYA-Profiling-OSX/***
Config Name	**ERYA-Profiling-Win64.conf**	**ERYA-Profiling-Linux.conf**	**ERYA-Profiling-OSX.conf**

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

Note: A guided step-by-step setup for Windows and Mac OS X are displayed on the next session.

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.

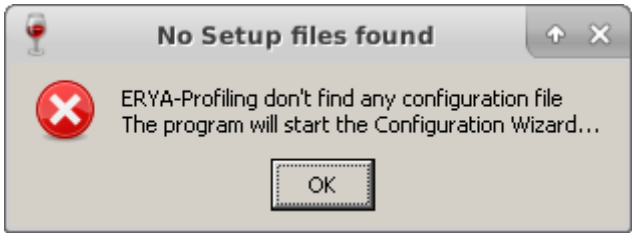
Local User Profile Setting: Once the profile folder are created, according to the previous table, and the configuration file are written, a copy of Database, Ziegler and Detector files are created on the same local user profile folder. Any changes by the user will change, by default, the local copies, 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.

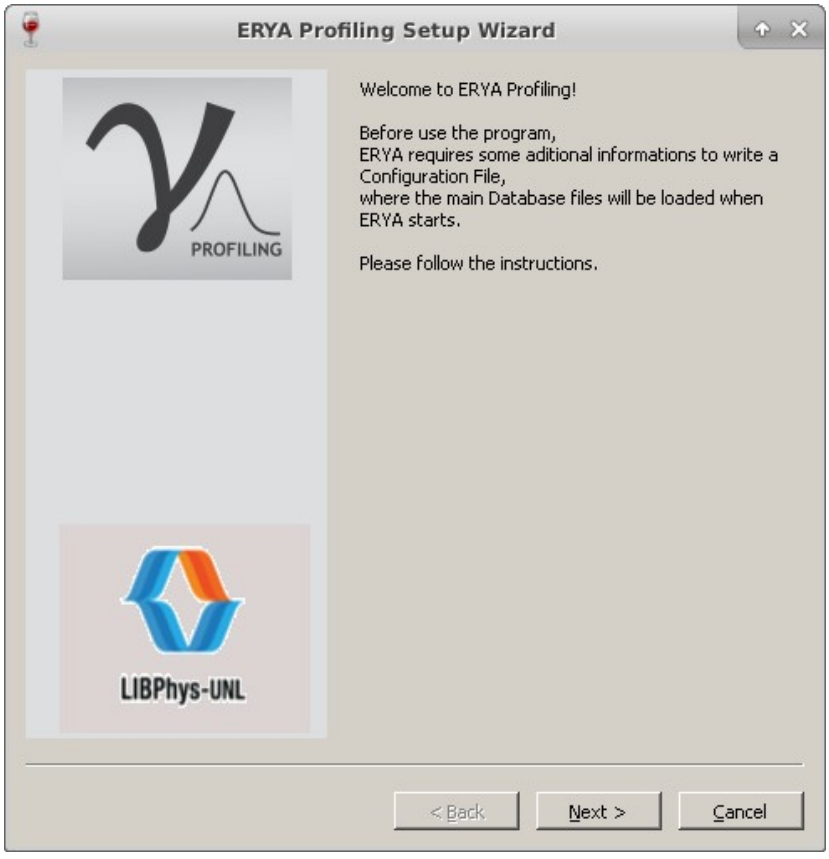
ERYA will warn with an error message if any essential database cannot be loaded.

Initial Setup Step-by-Step

1. Once ERYA are started by 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 has described by the table on the previous chapter, then repeat this step again.



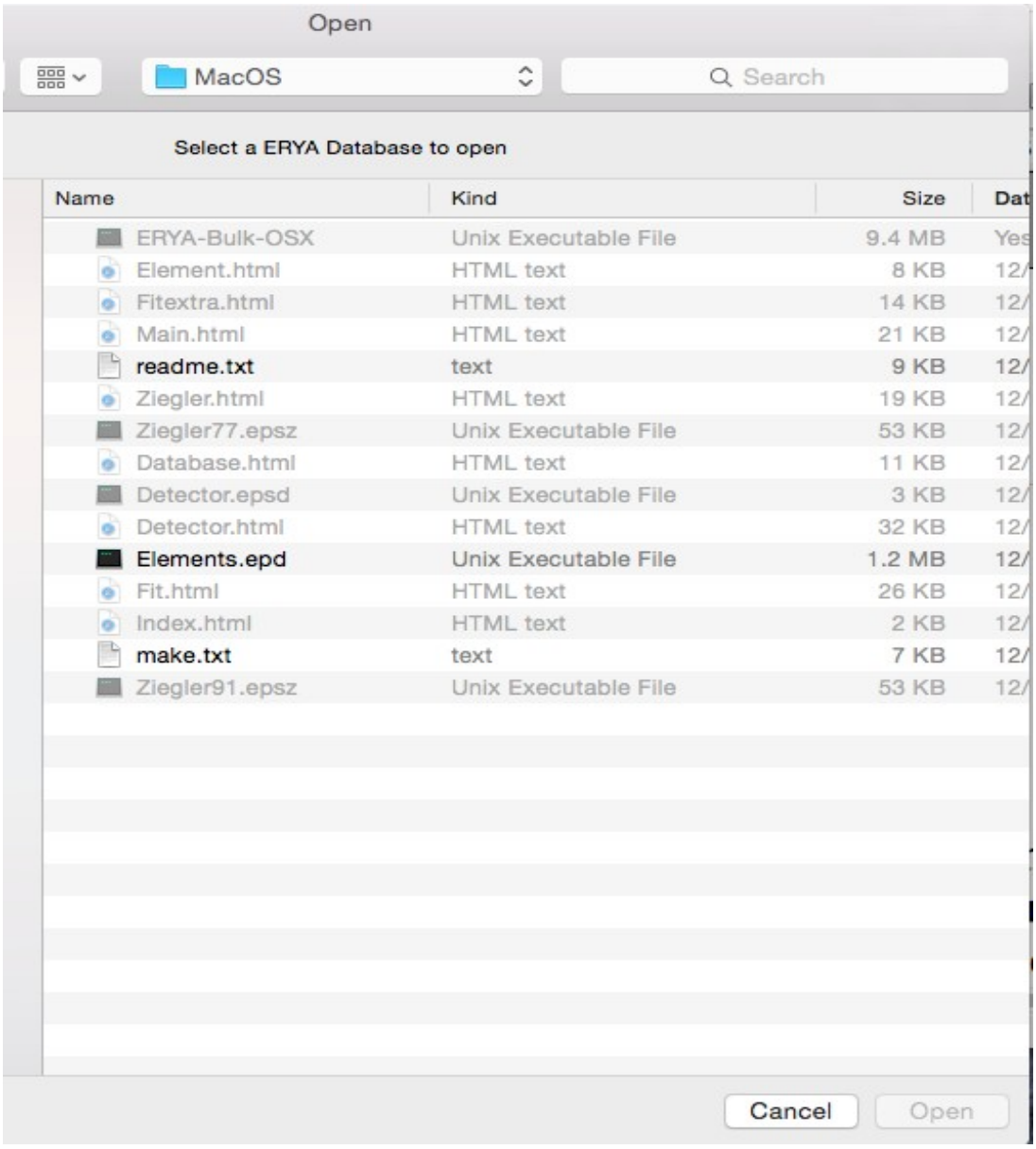
2. It will start a wizard, guiding the user for the next steps. Just click on “Next” to go forward, or “Prev” to go back. The wizard will not make any real work until the user choose all obligatory settings.



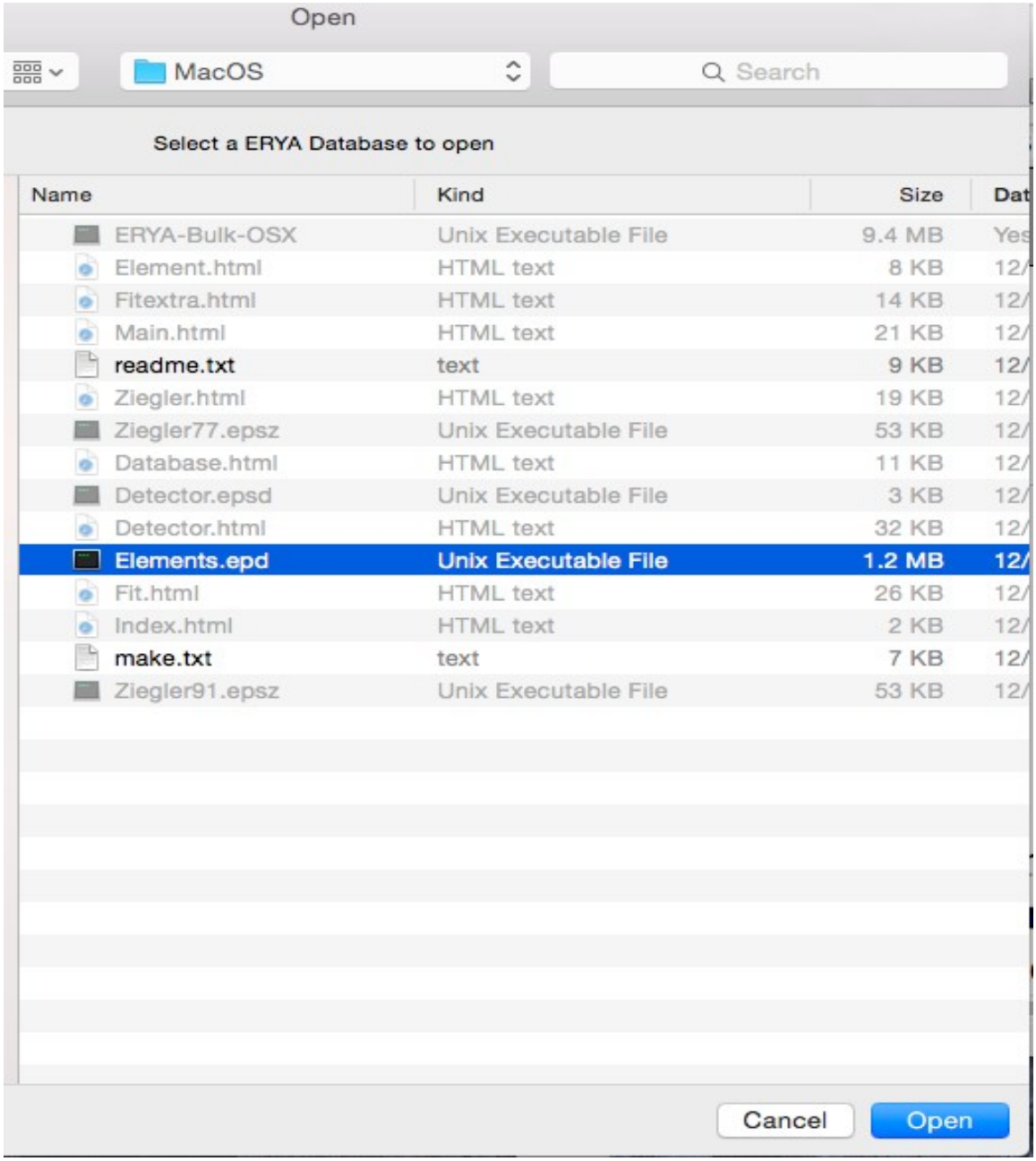
Notice that each wizard pages will ask to select a compatible database file.

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.

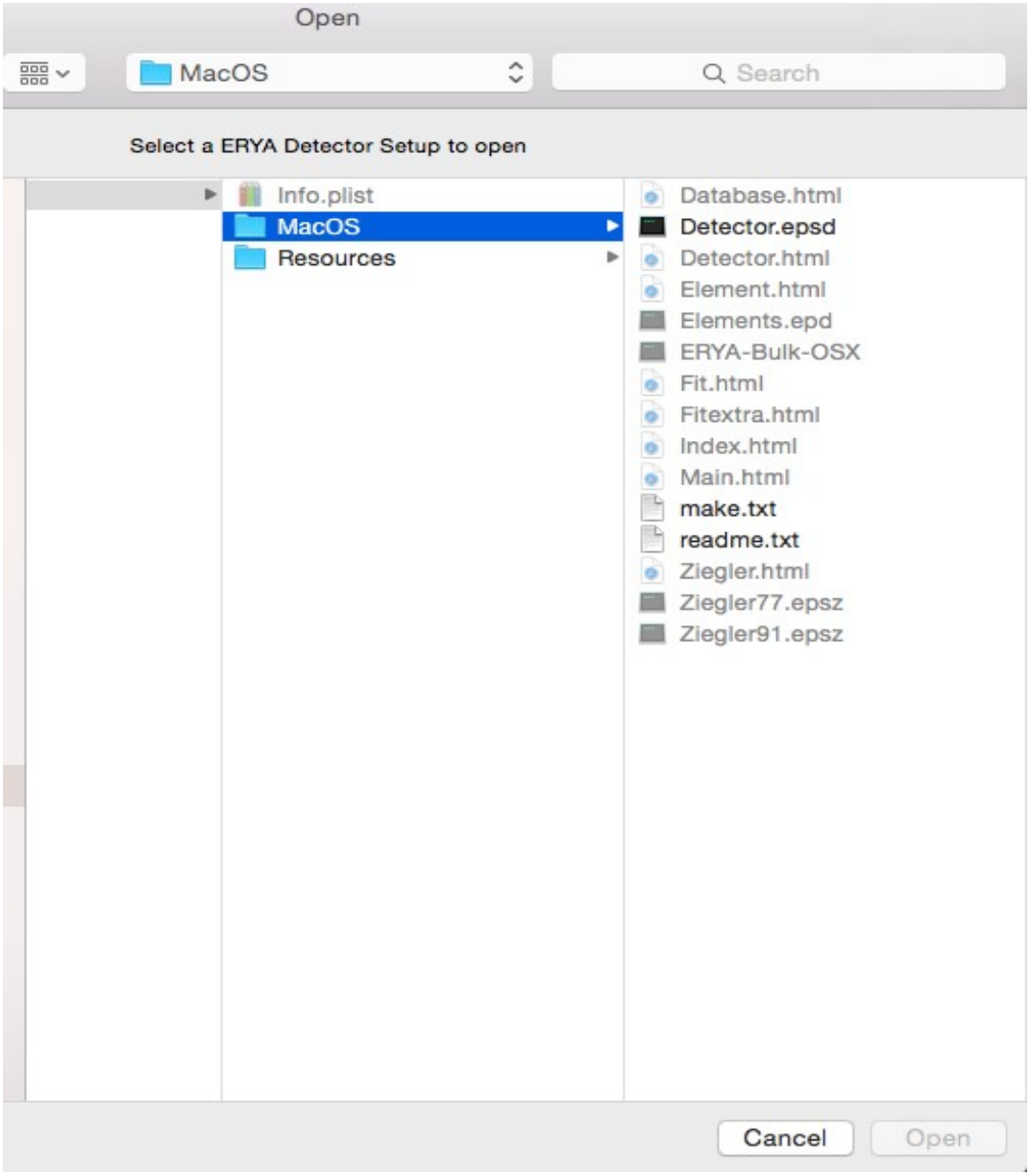
On Mac OS X, the standard dialog will display all files, but shades the incompatible ones:



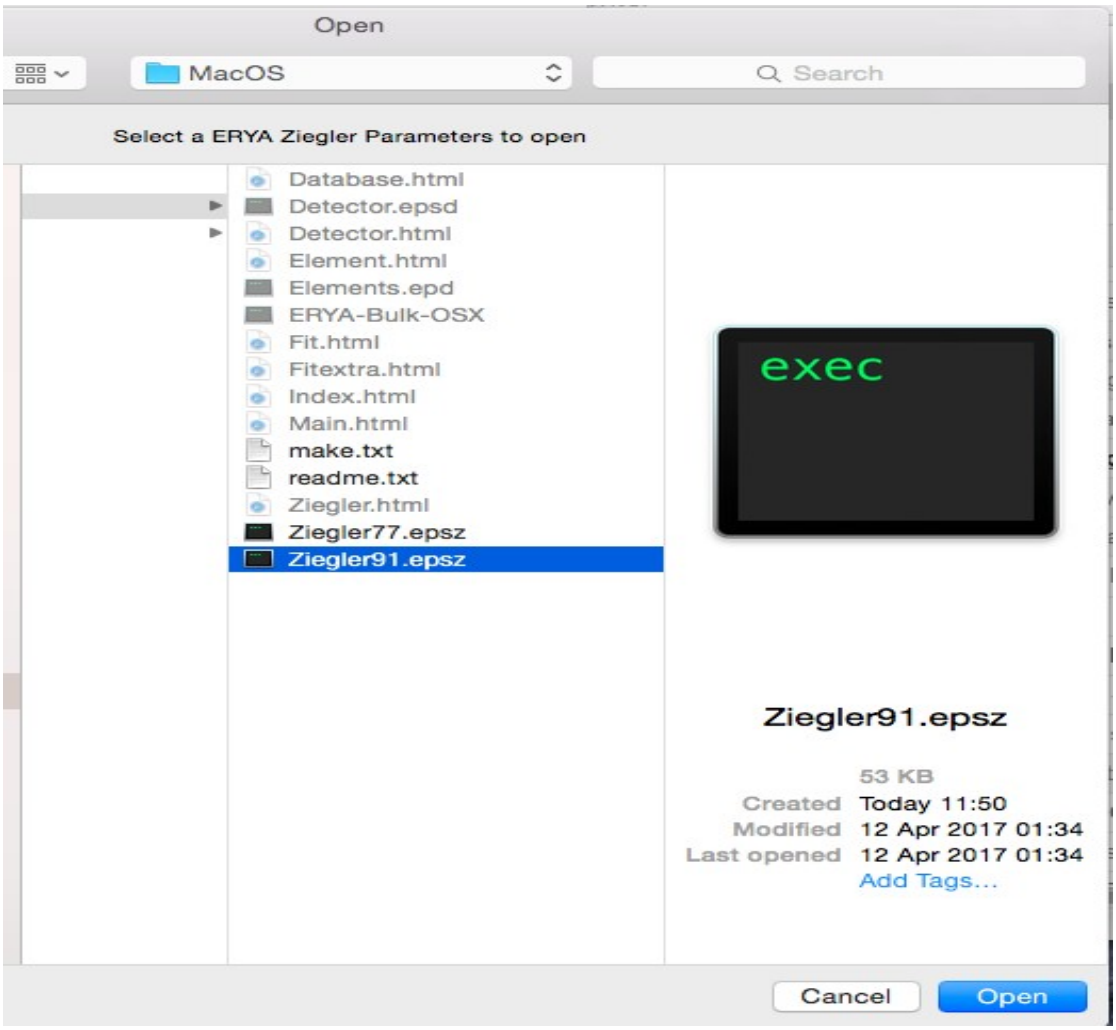
And select the file with **epd** extension, as indicated:



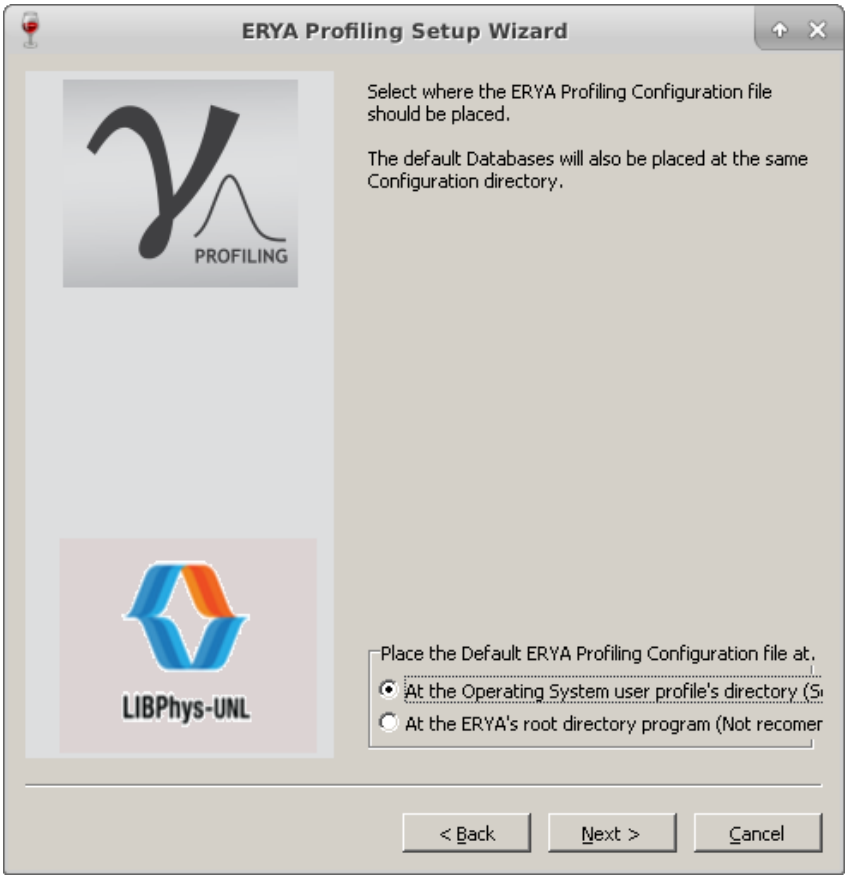
4. The same logic will apply for the Detector Profile, and select the file with **epsd** extension:



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.



Once selected, click "Next" to read the final informations, and click on "Finish" to start the real configuration procedure.

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

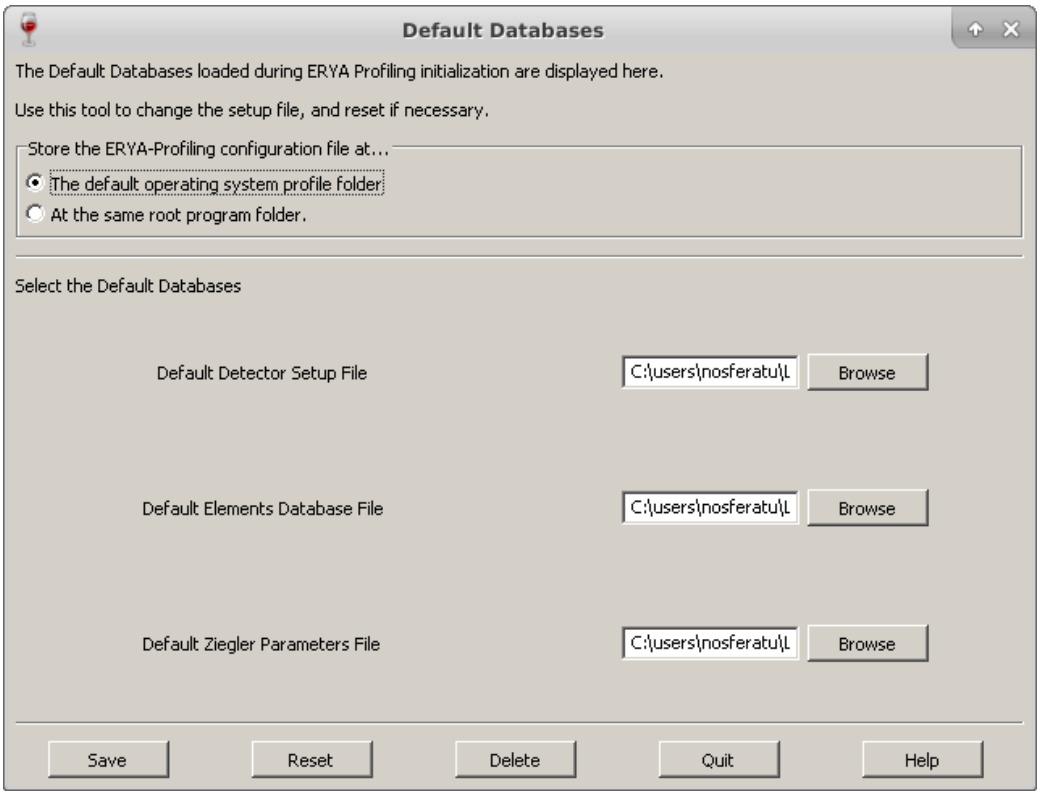
8. The configuration file are created, and the main GUI is loaded automatically. ERYA-Profiling are now ready to use, unless an error occurs (see the Troubleshooting section for additional details).

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 use the built-in widget from Settings > Program Settings...

Where a new widgets 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. In case of some error, whatever the motive, that makes ERYA warns about corrupted or non-existent databases, then 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, where it will force ERYA to repeat the initial setup wizard again. The same can be made, by the Settings > Reset Settings menu option.

If necessary delete manually the user profile folder as detailed on “Initial Setup”, to force ERYA to start the initial setup routine again. Also, delete any configuration file on root program folder.

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-Profiling’s author recommends that in any circumstance, **you should not run** ERYA-Profiling 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.

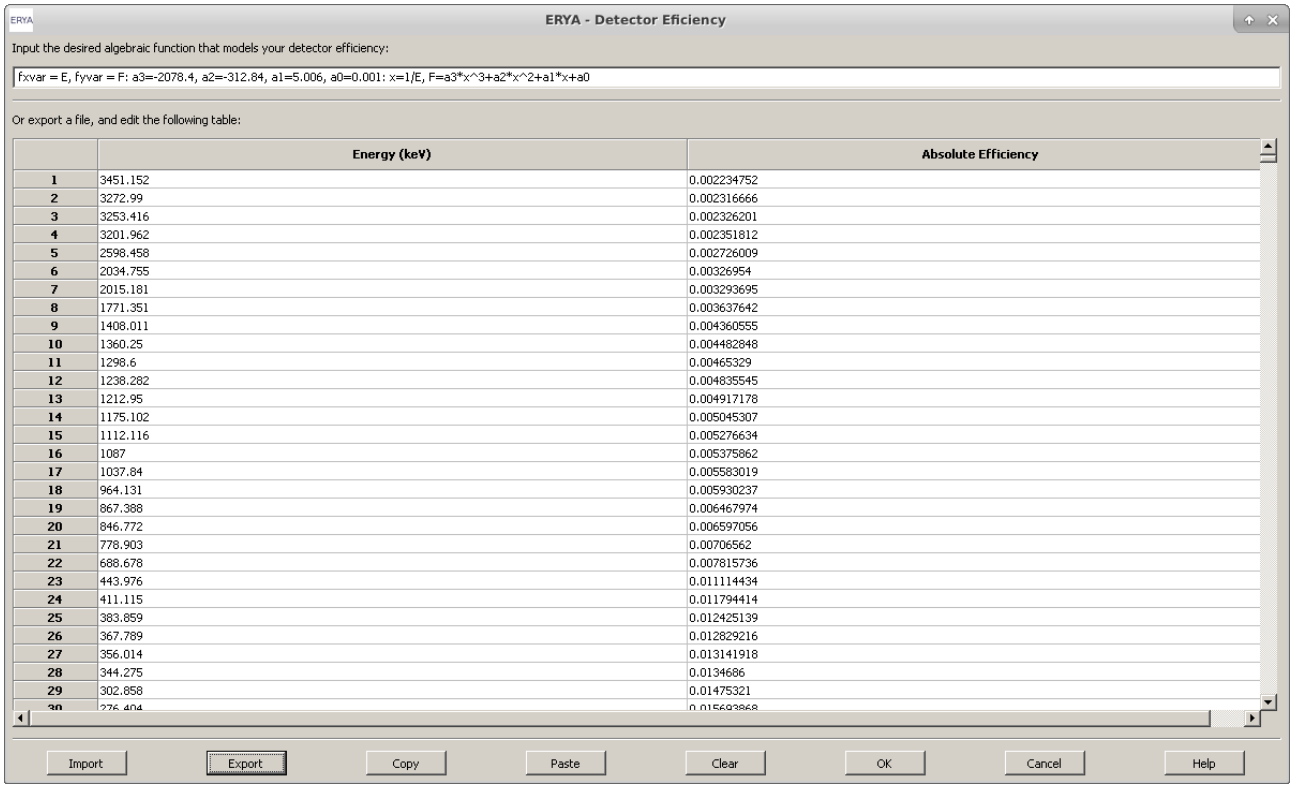
3. The optimal screen size for ERYA is a FullHD screen (1080p), but can work on a 720p screen at minimum, and main window screen will create scroll bars in case when the screen don’t fit all the elements on the same time. If the other windows appear cropped, then you need to increase the screen resolution.

Explore the Database Management Tools

Detector Efficiency

One of the essential Databases needed 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:



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.

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.

Deprecated Features:

It is also possible to copy the table contents to a text or spreadsheet file, but this feature is very error prone, and officially deprecated.

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".

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.

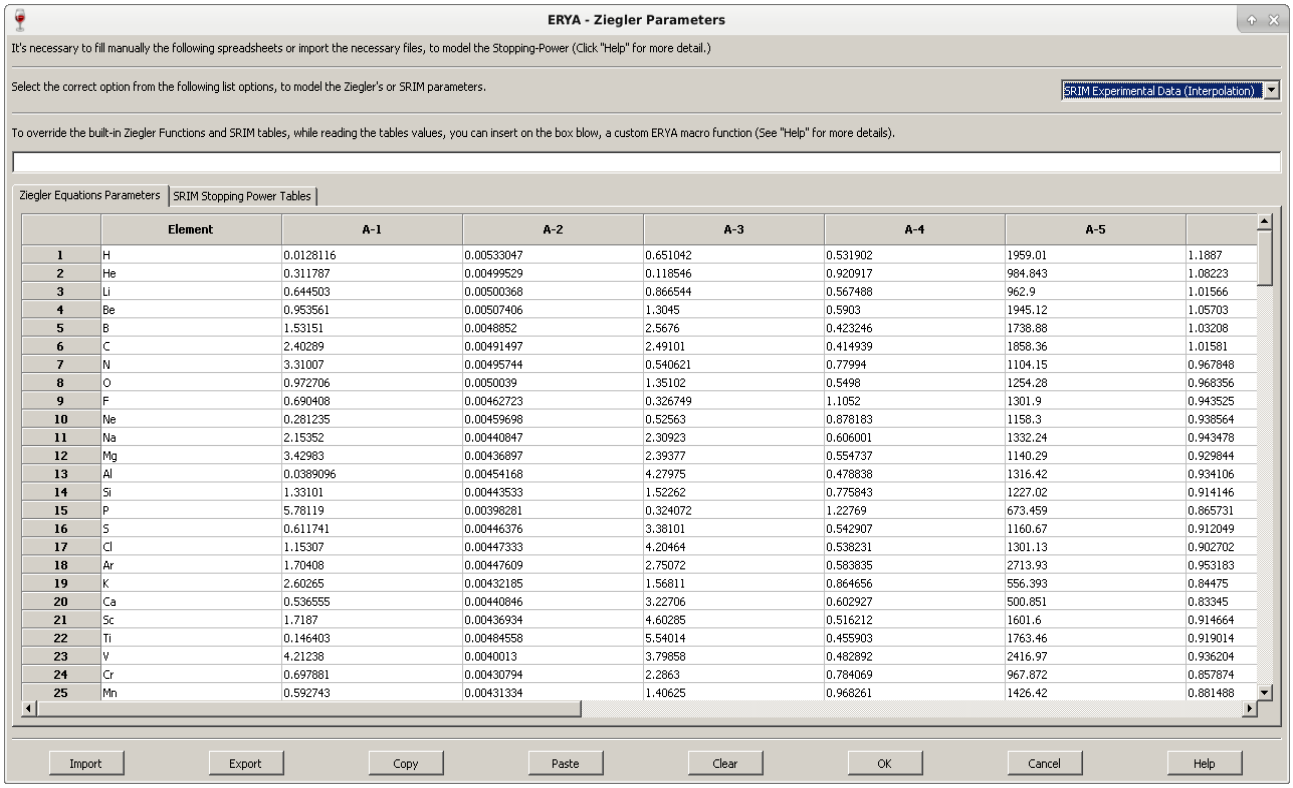
When ERYA loads the macro, it compiles to a chain of tokens, and caches the last field, making a kind of computer program that once got the input value for the independent variable, it will run the code until got a value for the dependent variable function, or an error that will trigger a simulation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

Is also the only Database where the ERYA-Profiling differs slightly form the ERYA-Bulk, since on this program, it contains 16 columns, due to the extra "Element Density" and "Bloch Parameter".

Cautions between ERYA-Bulk and ERYA-Profiling Ziegler's Databases

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

This means, that if ERYA-Bulk reads the ERYA-Profiling Ziegler's database it will discard the two unused columns (and save it will rebuild as the expected ERYA-Bulk version).

And ERYA-Profiling when read the ERYA-Bulk version will add a filler value "1" to all densities, and "10" for all mean ionization parameters.

On the prototyped LabView ERYA Profiling application, those additional columns are two additional text files, since it use the same binary Ziegler file from the LabView ERYA.

On wxWidgets based ERYA Profiling program, the extended Ziegler database combine all former binary and text files into a single database.

To avoid manual merging, considerer to use the "Tools" > "LabView Import" wizard described on the *LabView Import Tools* chapter.

It's possible to export and import the Ziegler's parameters using an Excel *xlsx* file, but avoid to share between programs, to avoid unexpected results.

Using the Custom Ziegler's Functions

Using the Ziegler's version 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 next variable with *13* suffix will get the Atomic Mass. Finally the variable with *14* suffix will be assigned to the Atomic Density, while the last variable with *15* suffix will be the mena ionization energy by atomic number (Bloch's Constant).

As an additional 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:

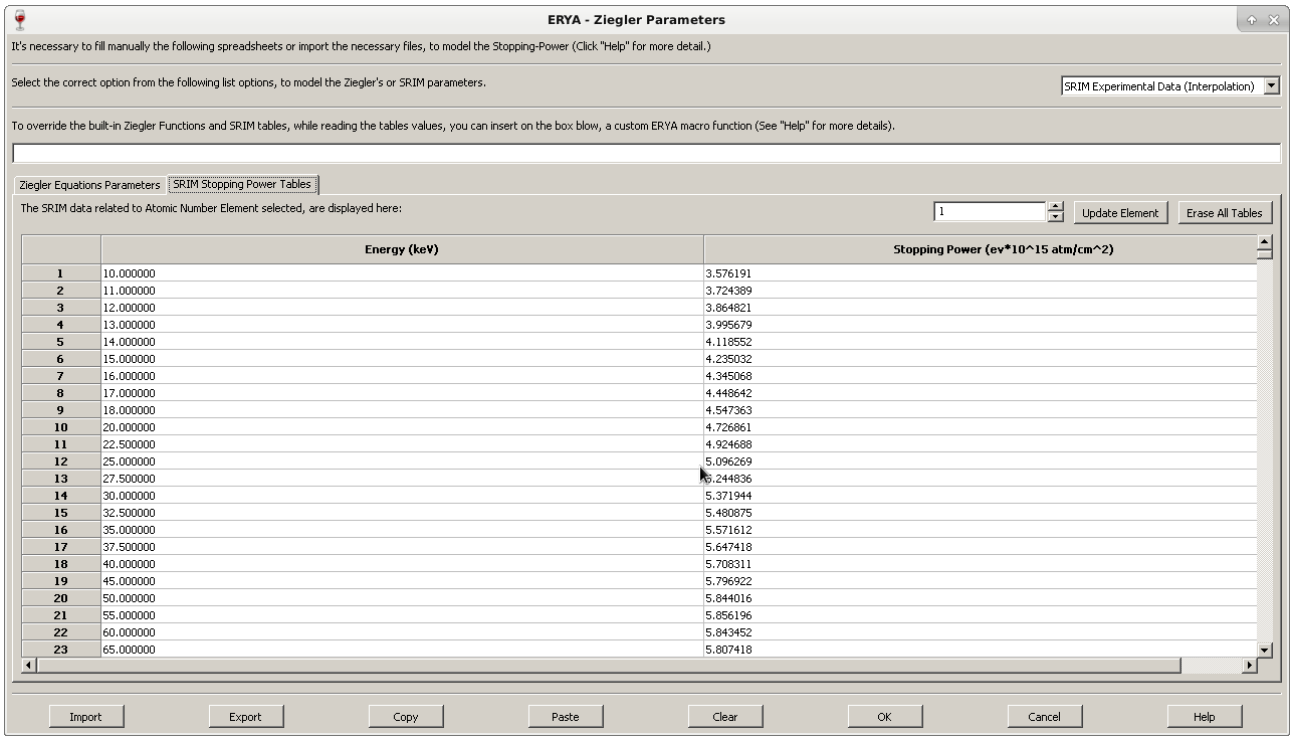
(x < [x_a]) * ([Any function]) + (x > [x_a]) * (x < [x_b]) * … *([Any function]) + …

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.

Notice than when a LabView binary Ziegler's file are parsed, it will place two placeholder values for Density ("1"), and Bloch("10").

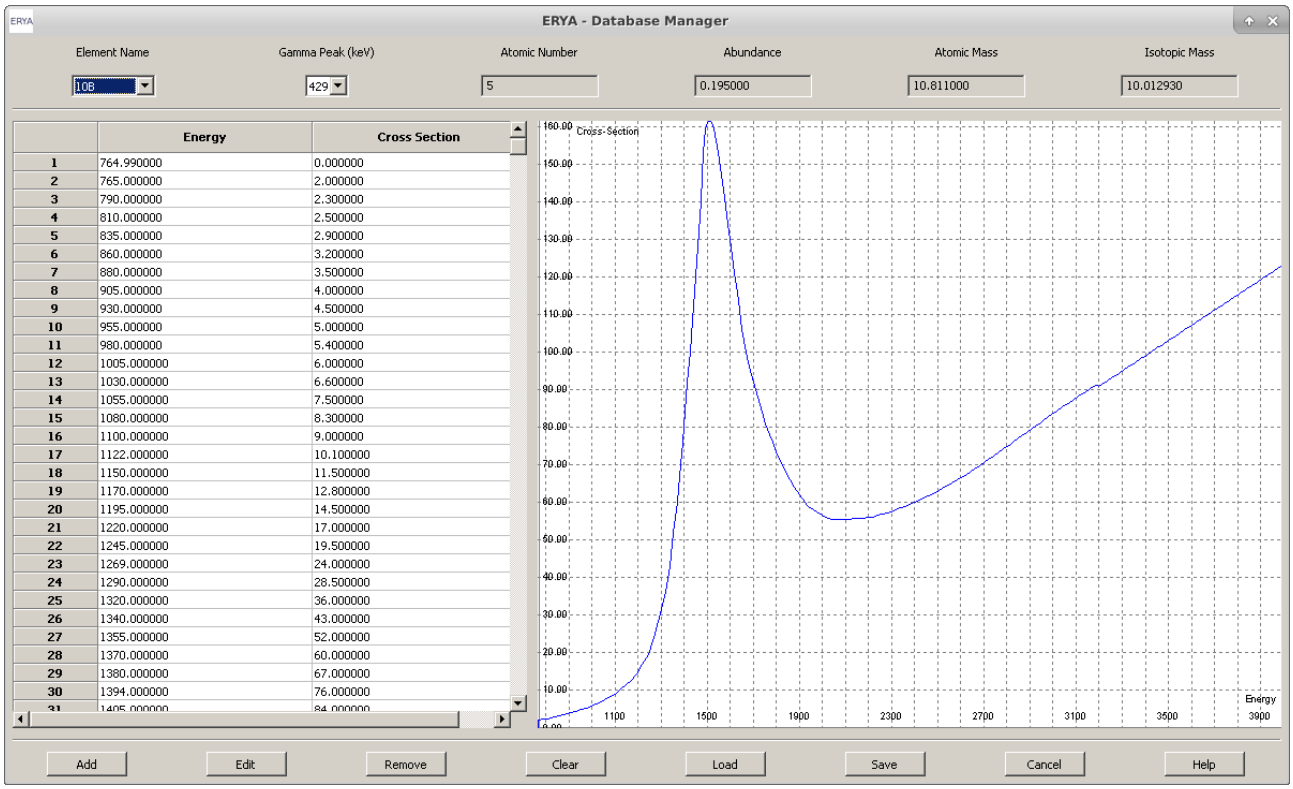
The same applies to unedited SRIM Stopping-Power output files, that is plain ASCII text files, as long corresponds to single elements tables related to proton beams. When the Import button are activated with ASCII import, and ERYA detects that is a SRIM table, it will convert to the correct Element's 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

For major convenience, the physical characteristics of several elements/isotopes are handled by the Database Manager widget.

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



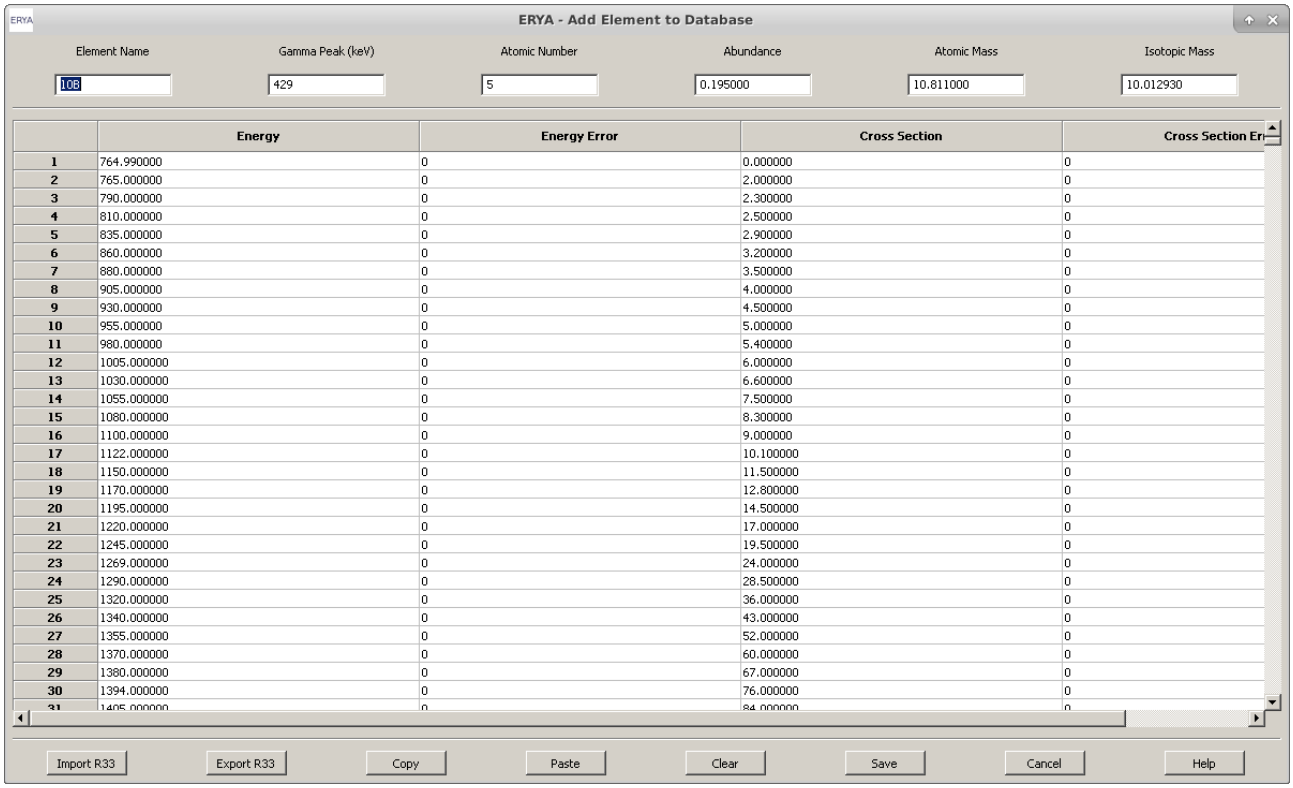
Using this tool, it is possible to handle one database at time (normally the default database).

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.

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.

Other Database Manager Options

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.

Notice that only overwriting the default database defined by the configuration file will be automatically reloaded when ERYA-Profiling starts.

Compatibility with LabView Binary formats

Like the other essential files, the native ERYA Database is a XML file with extension **epd**, but 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, and also write to the ASCII format, although it splits any elements with different gamma emission values to separated registers.

It is recommend to use the *LabView Import Tools* to handle conversions of LabView databases, even it can open by this tool directly, since all further LabView support are deprecated.

However it is not possible to use the LabView format as a default Database, since it will convert everything to the XML 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. Then, the changed Database are reloaded to the main memory.

Additional details can be obtained from the “Help” button.

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.

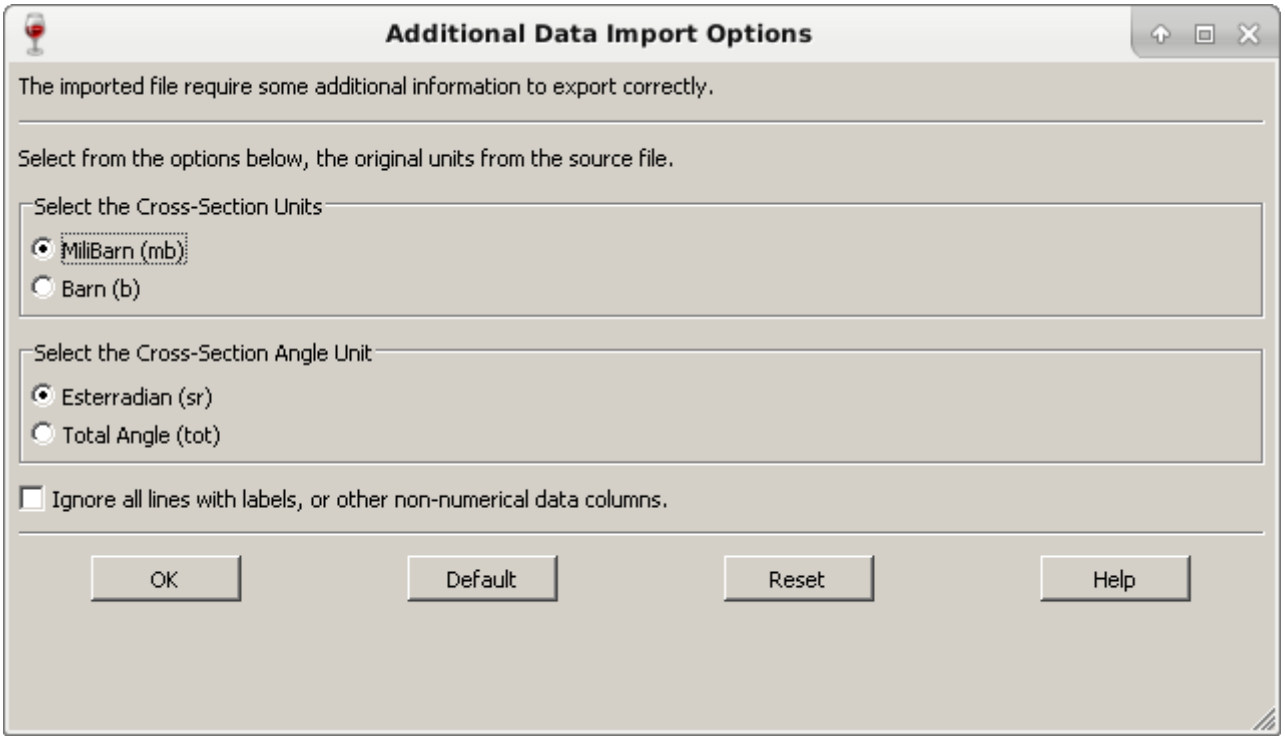
Normally, the cross-section are experimental values stored on several file sources, and it is very rare to manually type on the built-in sigma table editor.

(It is fine for about half a dozen of lines, but impractical for hundreds of them!)

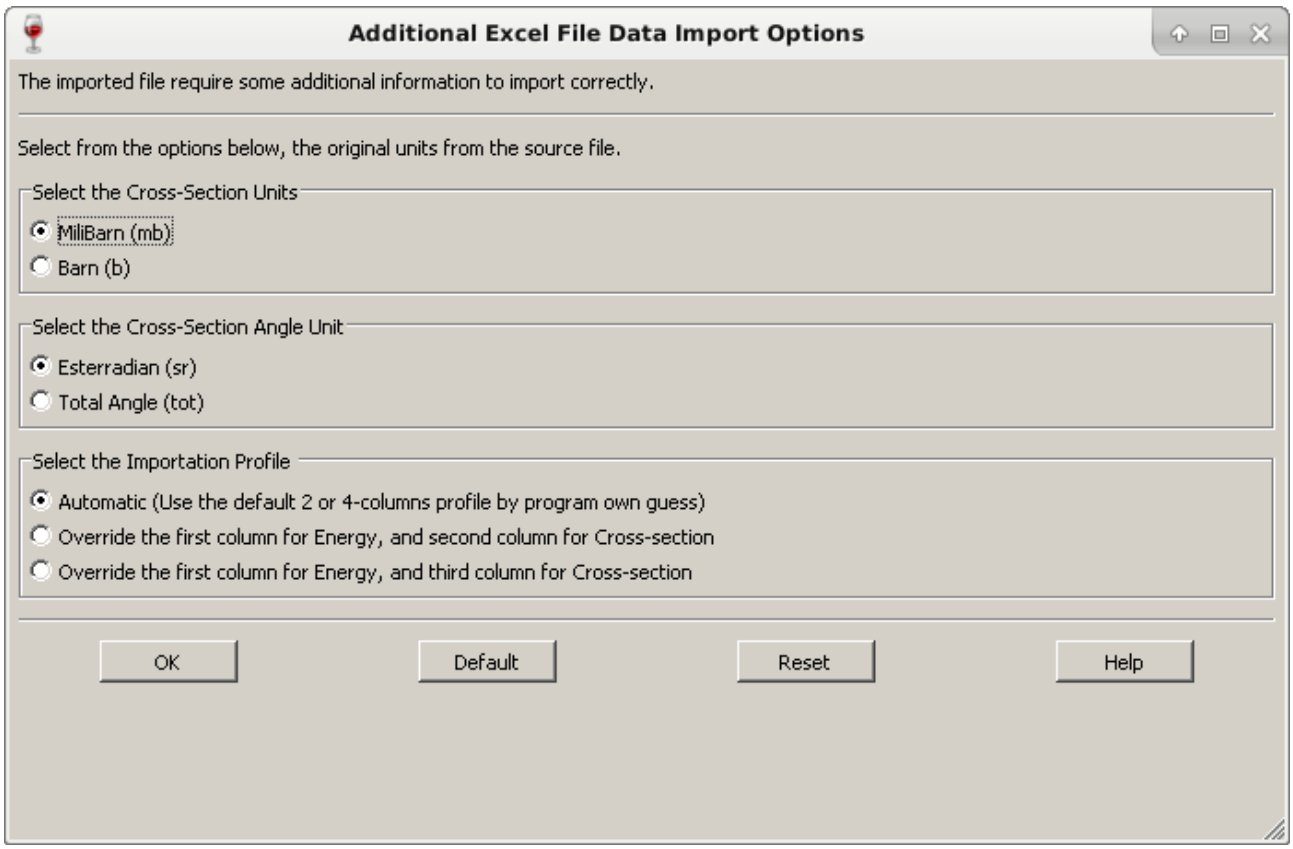
It is possible to export the cross-section from a r33 file from IBANDL (where also fills the Element Name, and the Gamma Emission), from a ASCII file (usually in two-columns text files, and normally the cross-section are in barn.keV), and even from an Excel file (As long had a xlsx extension, and can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

To ease the conversion between units, and avoid manual conversions using intermediate files, 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 if re-exported.

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" 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.

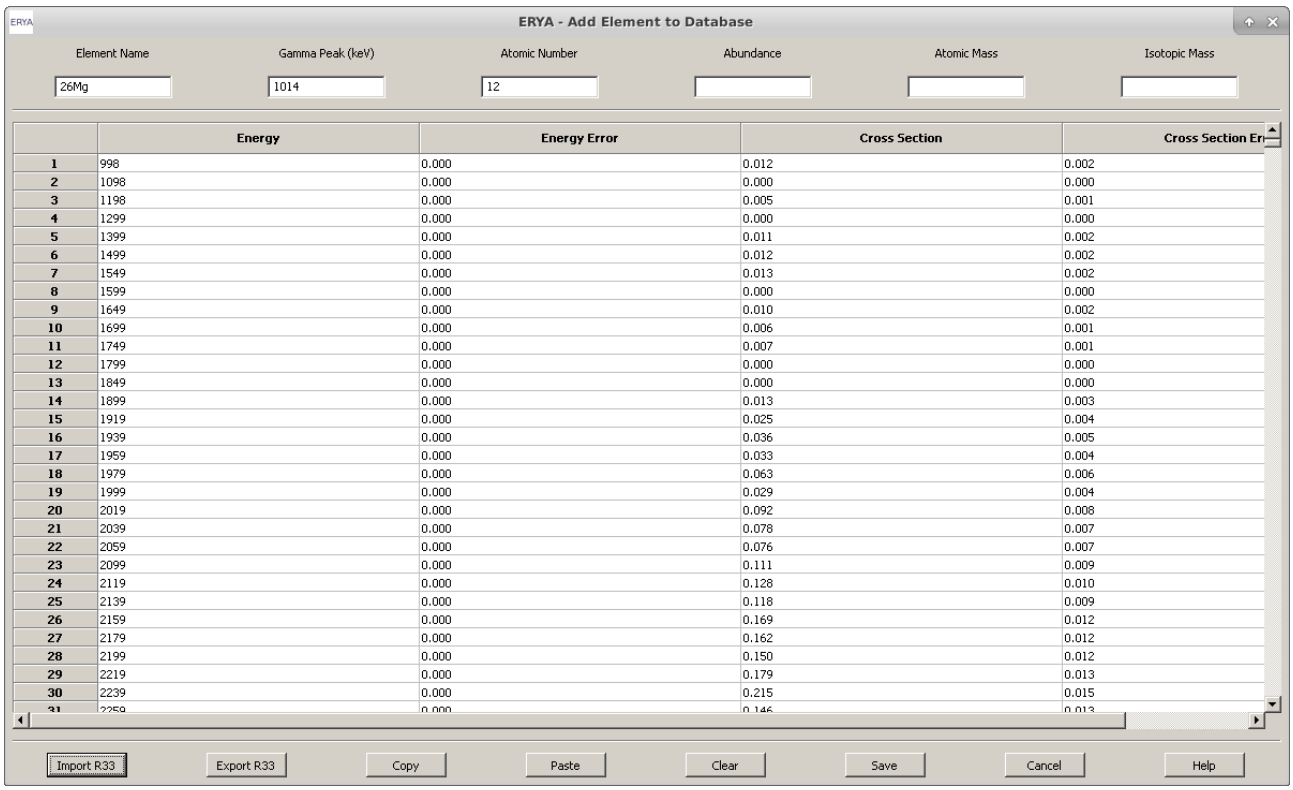
5. Select the relevant r33 files, from the "Import R33" button.

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.

In your 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

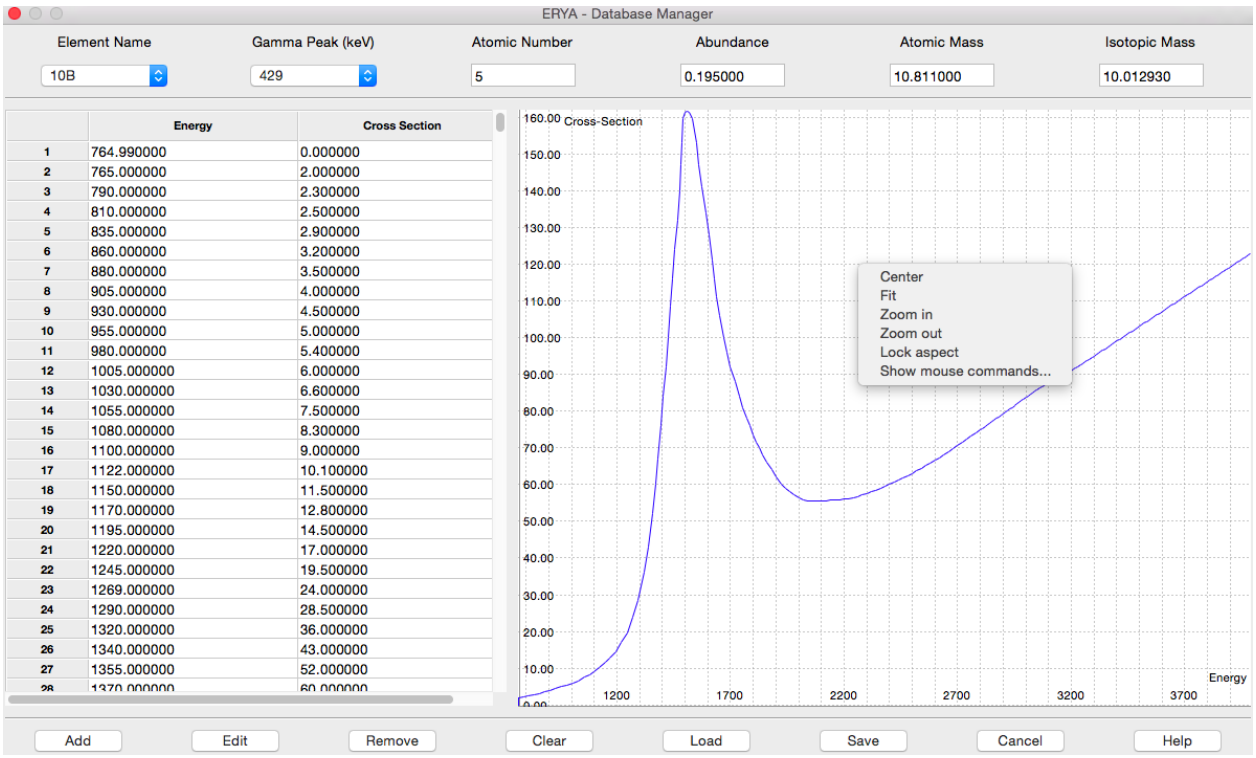
4. You can the graphic plot feature to zoom, fit or select a portion of the graphic by your convenience. The main commands are available by right-clicking the mouse over the plot.

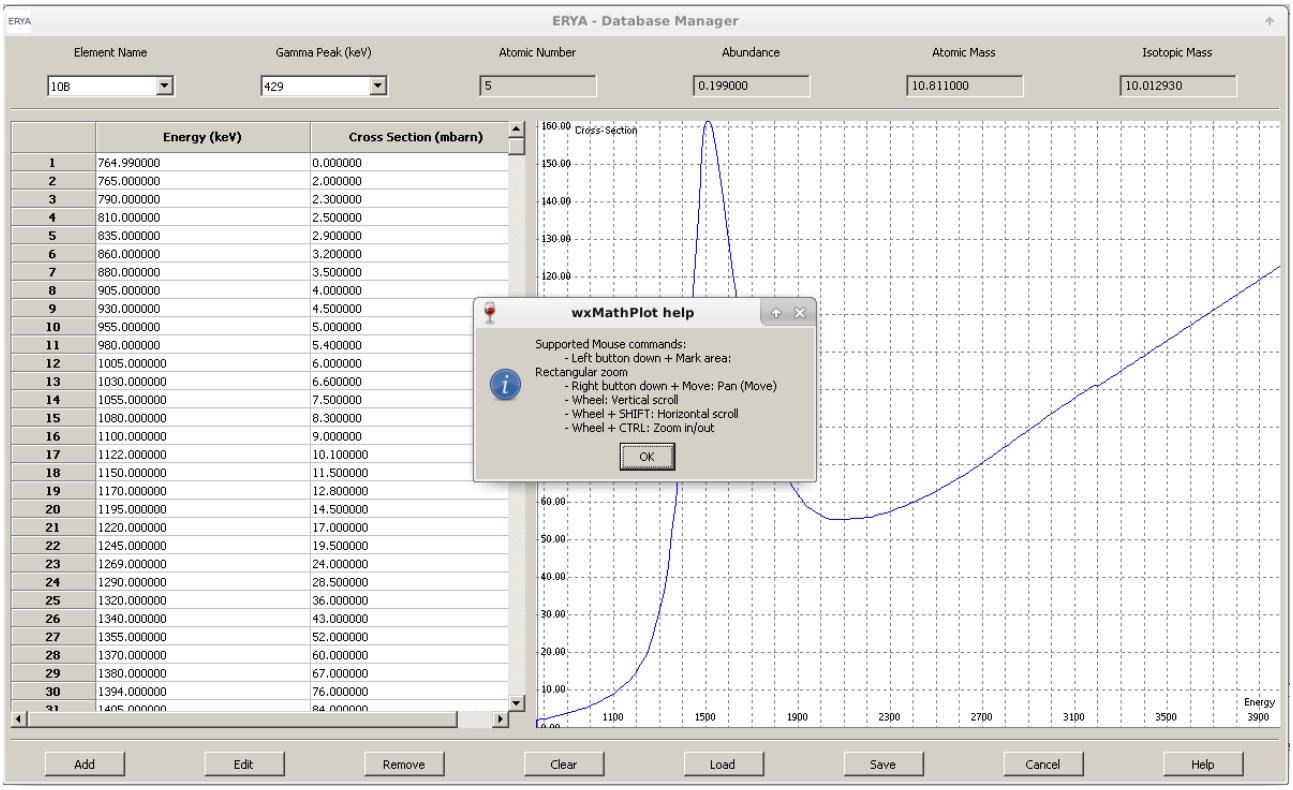
You can select and drag the zoom window, while hold the left mouse button, in order to magnify the selected portion of the graphic.

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

ERYA v2.40 (ERYA/Profiling Tutorial)

Screen-shots of graphics menu tools, on Mac OS X and Windows/Linux:





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

Useful Utilities

How to Export LabView Databases Safely

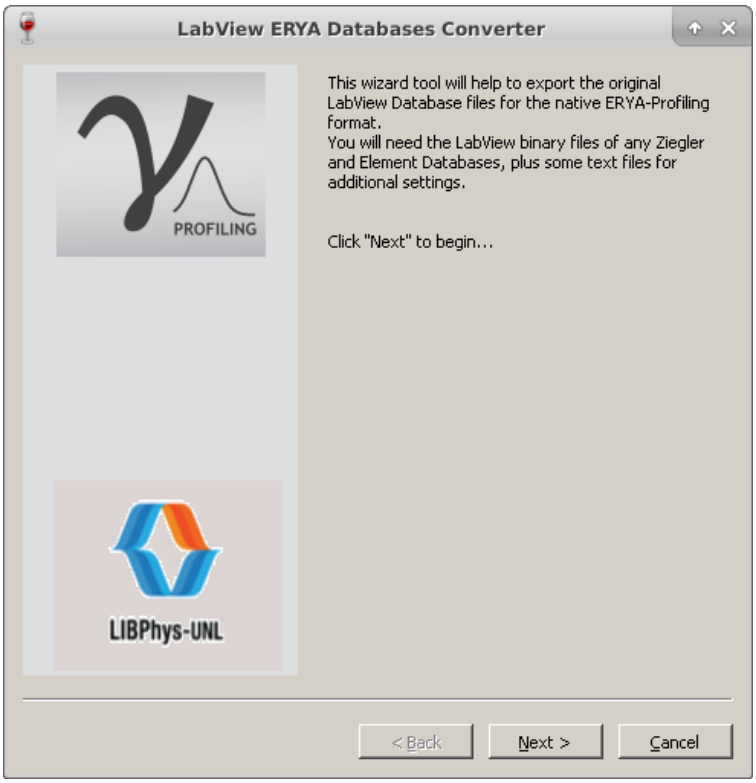
The original ERYA Profiling software package was a main program that use a GUI made with LabView Runtime, and additional working programs that makes the actual numerical simulation. The program package need five files to make them work, where ERYA-Profiling can export it.

It contains two binary files for the Elements Database, and Ziegler Parameters, and plain text files for the Detector, densities and mean ionization potential (also called as Bloch's Parameter).

Some LabView builds ever hard-codes the Detector efficiency function, and this require to fill manually the custom function text-box as described on the *Custom Efficiency Function* section of the Detector Profile widget.

If you ever needed to export the former databases, it is highly recommendable to use the following wizard tool available on Tools > Import ERYA LabView

1. Once selected the LabView Import Wizard, it should display the following wizard page:



2. On the next five pages, the program will ask to select a compatible file from the following components, as ever explained by own wizard's page:

- Element Database;

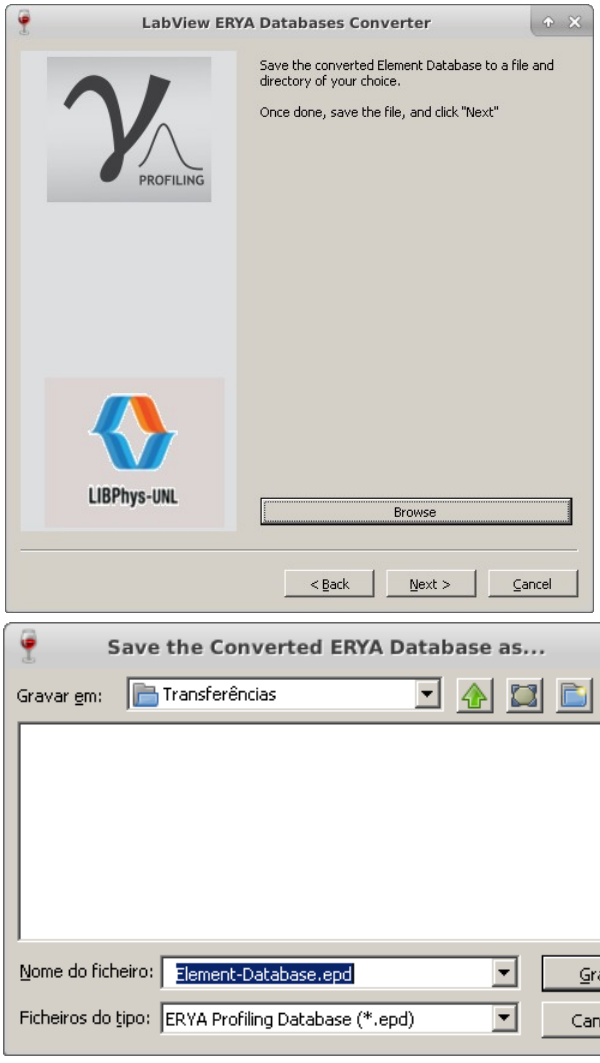
- Ziegler Parameters;

- Detector Efficiency;

- Element densities (can be a single column of values, as long are in the same atomic number order);

- Bloch's Parameters (can be a single column of values, as long are in the same atomic number order);

3. At this stage, the wizard will ask to use the define a series of three new files, using the same template, like this example:



(Click on Browse button, then fill a name on an adequate folder, and click "Save")

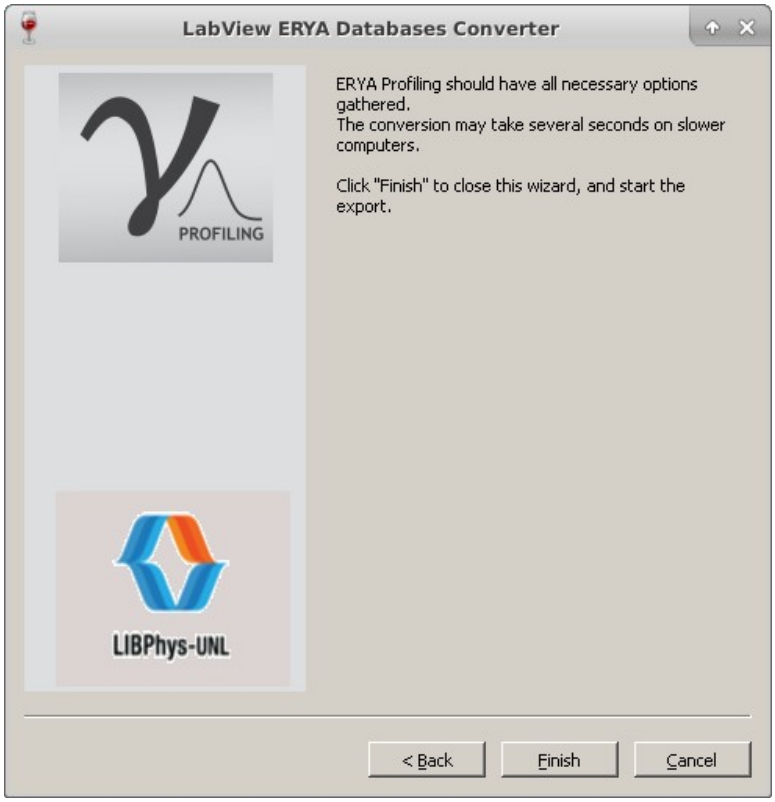
4. Repeat the template from step 3 for the following files:

- Elements Database;

- Detector Efficiency;

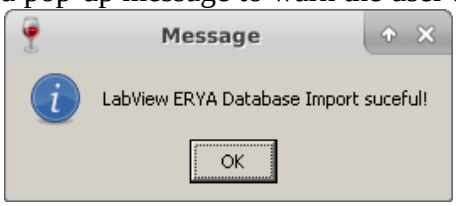
- Ziegler Parameters;

5. Once reached the final wizard's page, as described below, click "Finish" to start the real conversion.



6. All previous databases are converted and written to the new three native format ERYA-Profiling formats. The process can take several seconds, where the Element's Database are analyzed, and sorted, taking the most time on the conversion procedure.

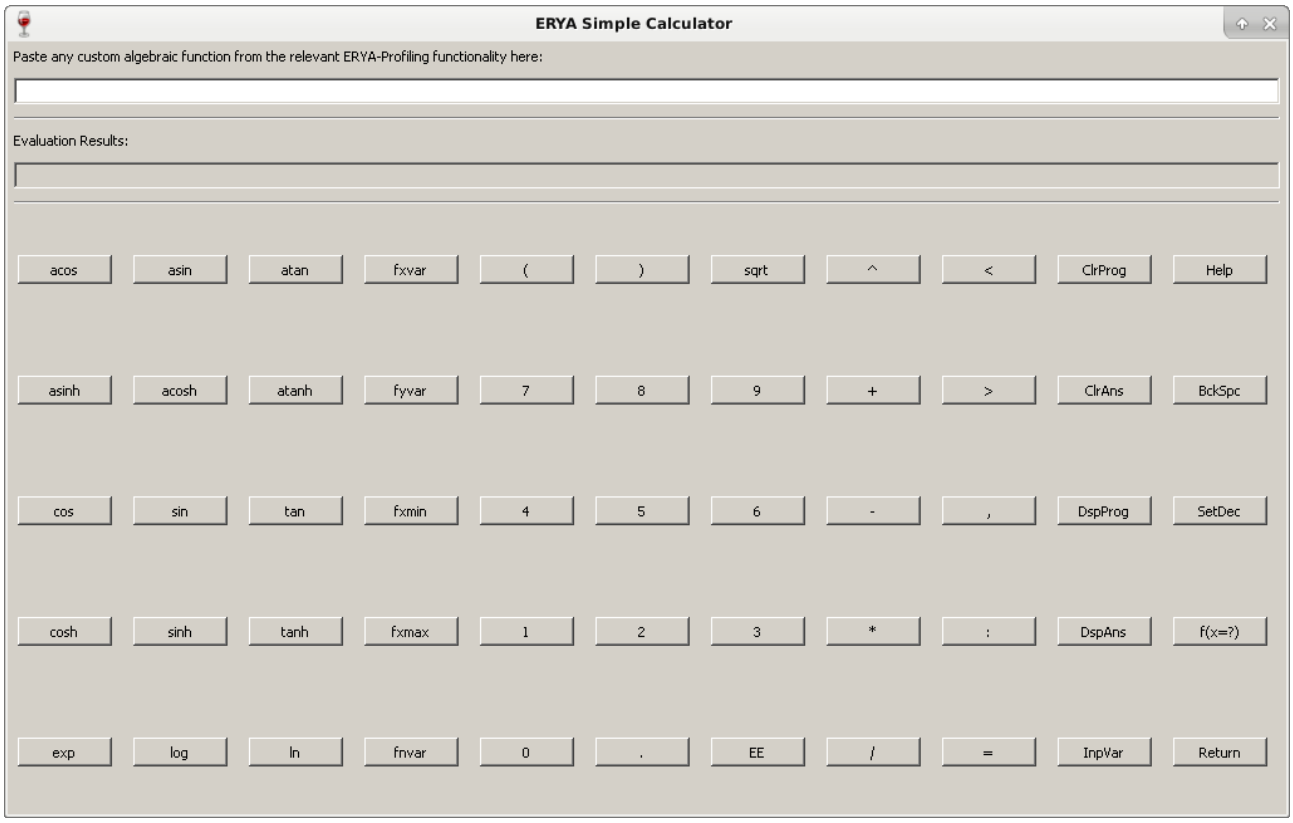
Once finished, it display a pop-up message to warn the user that the conversion was successful.



7. However, the new converted files will not substitute the original ones. You should use the Database Management tools to check each one, and use the *Setup Tool* to replace the converted databases as the new default ones.

Using the ERYA Calculator

This widget is intended to serve as a debug tool for some advanced options that use the internal ERYA macro language, where the user can open a scientific calculator from “Tools” > “Calculator”



Either way, it can be used as a standard calculator anyway. Just fill any algebraic expression on first box, or input box, and then hit “Return” to evaluate numerically the expression on the second box.

Any syntax error will be displayed on the second box, also called output box.

It support a *direct mode*, and a *program mode*, where the difference is when the user defines a small program on the input box, more precisely a defined function, where can be:

fxvar = x , fyvar = y : a = 10 , b = 20 : a * x + b

Or with a limited domain, as it is needed on the *Resonances* tab on the main menu interface:

fxvar = x , fyvar = y , fxmin = a , fxmax = b : a = 10 , b = 20 , g = a + b : g * x

The variable names related to the *fxmin* and *fxmax* needed to be initialized, otherwise it use the default null values.

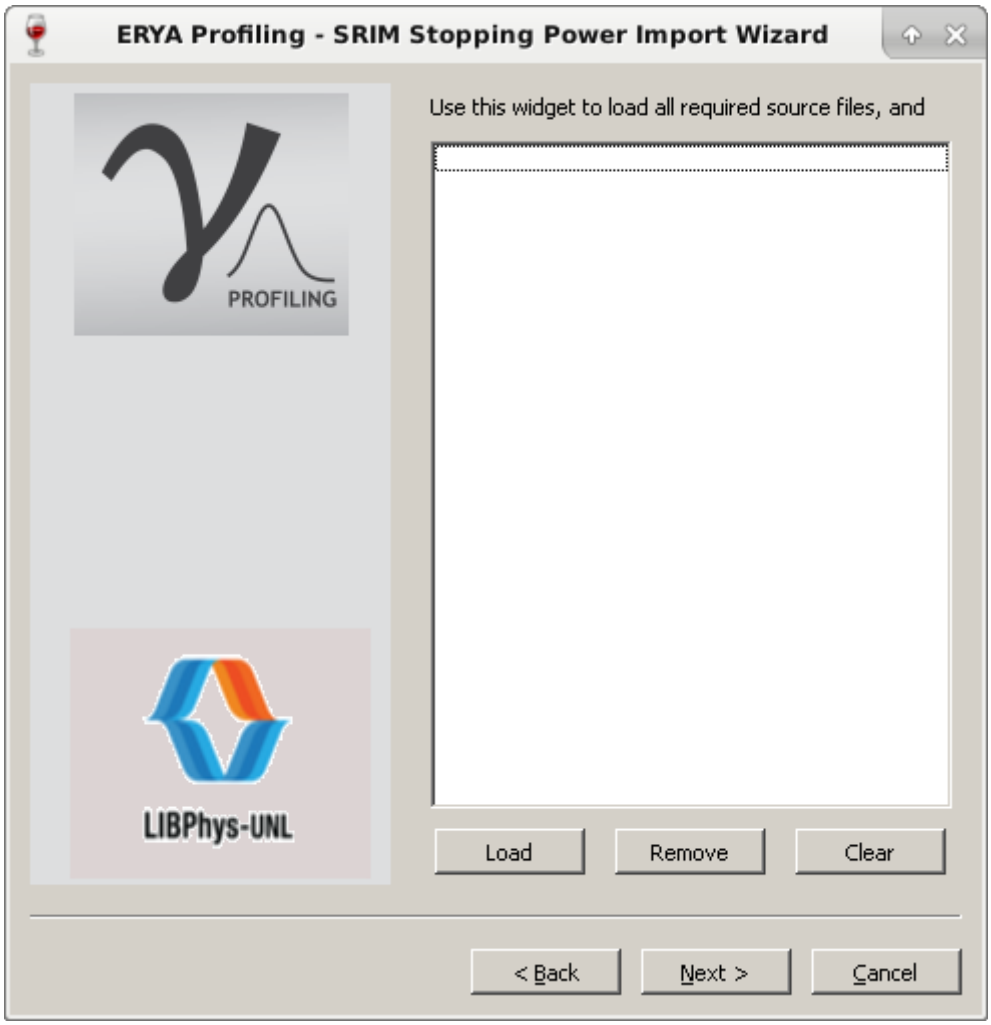
When the calculator is on *program mode*, it should hit the “f(x=?)” button, to fill a value, and it will display the result on output box.

More information are available on the on-line help.

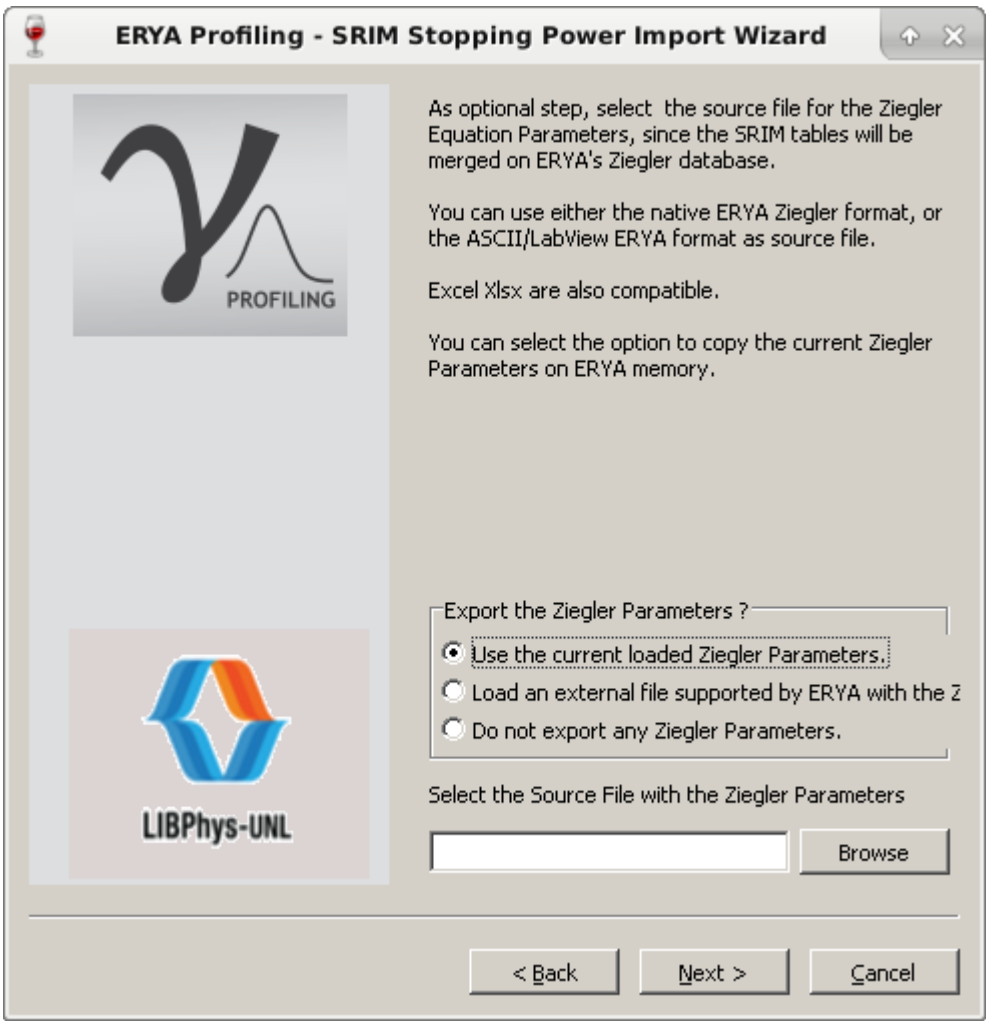
SRIM Import Wizard

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

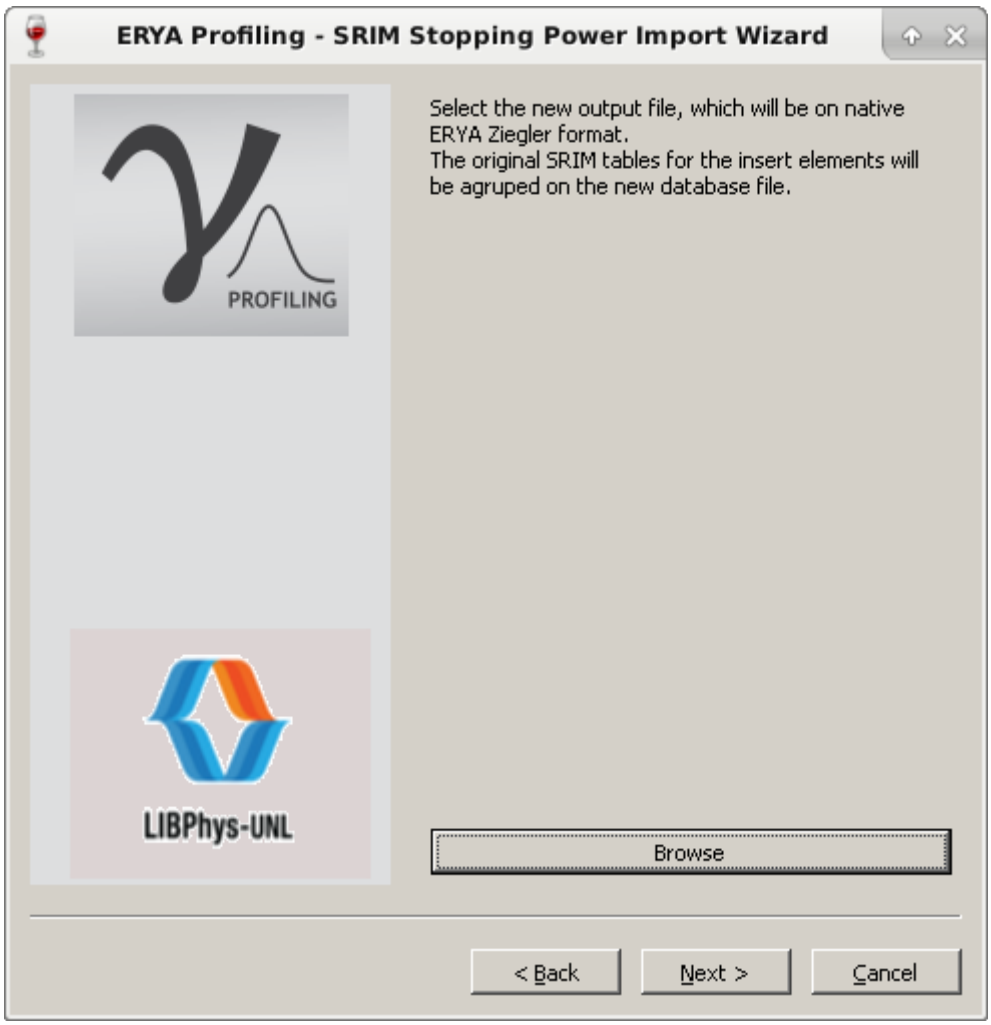
The user need to select pure element stopping powers, where ERYA will only support the incident proton (hydrogen) beams. Once exported the outputs from the SRIM® software, just select the desired number of files, and follow the on-line instructions:



The wizard will also asks to get the Ziegler's parameters from several sources (including the current loaded on memory), or loading an external file.



Finally, it asks to fill a new native **epsz** file name to output:



The whole batch conversion should be done in a matter of a couple of seconds, and only the valid SRIM files will be actually stored on the converted file.

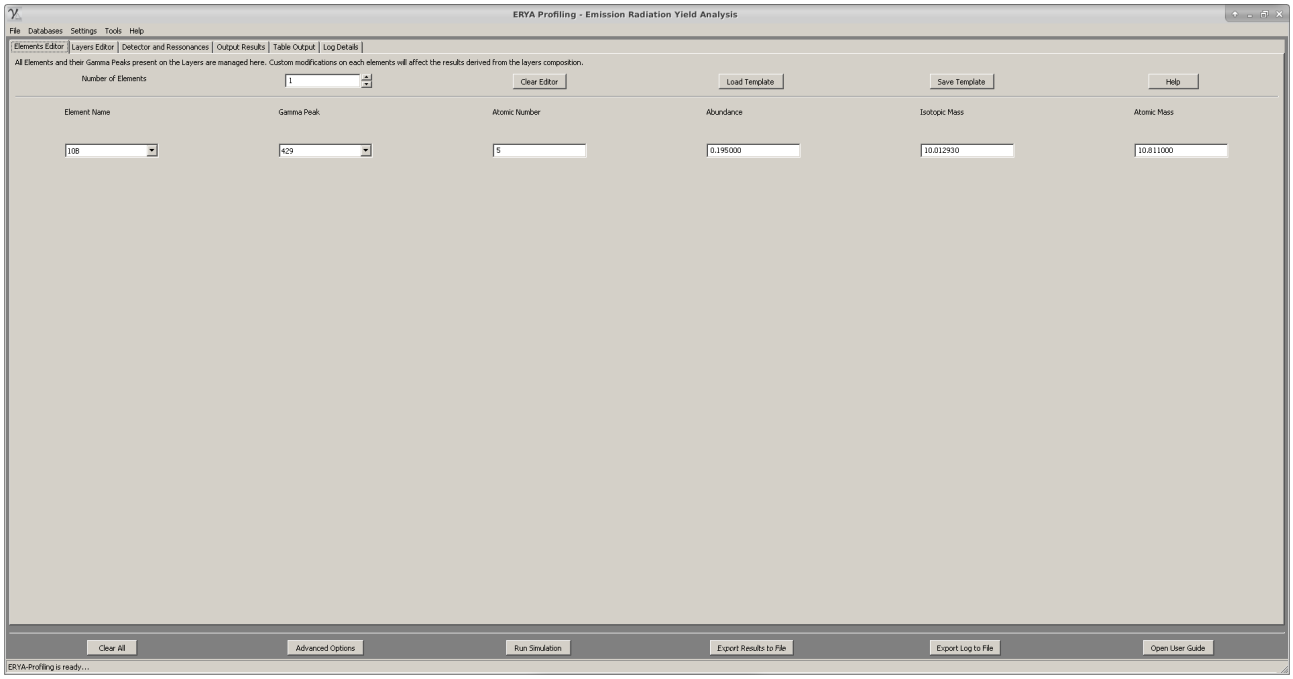
Note: If the batch list contains repeated files with the same atomic number, only the last one are stored, overwritten the previous ones.

How to make a Profiling Analysis

The main purpose of ERYA - Profiling is to make a yield spectrum in function of energy, and the depth of the sample. Future improvements are under current discussion, so this chapter will detail the implemented features.

Main Interface Features

If ERYA-Profiling was correctly configured, all default databases will load automatically on start-up, and wait for the user input, where it should be like this screen:



The program follow a modeled tabbed interface, where each tab had own functions and tools.

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

The bottom screen contains some buttons for the major tasks, and a status bar for some internal information, as will be briefly useful during the numerical evaluations.

The program require the user to fill the first three tabs, and then hit “Run Simulation” to start the whole process, where a percentage progress will be updated on the bottom status bar, since this process can take a long time to finish.

By clicking “Open User Guide”, it will display a resumed how-to tutorial of this current printed guide, with links for additional topics. Also each tab had own on-line help for additional information.

First Step: Define the Number of Elements

1. Select the “Elements Editor” tab, and select the number of Elements.

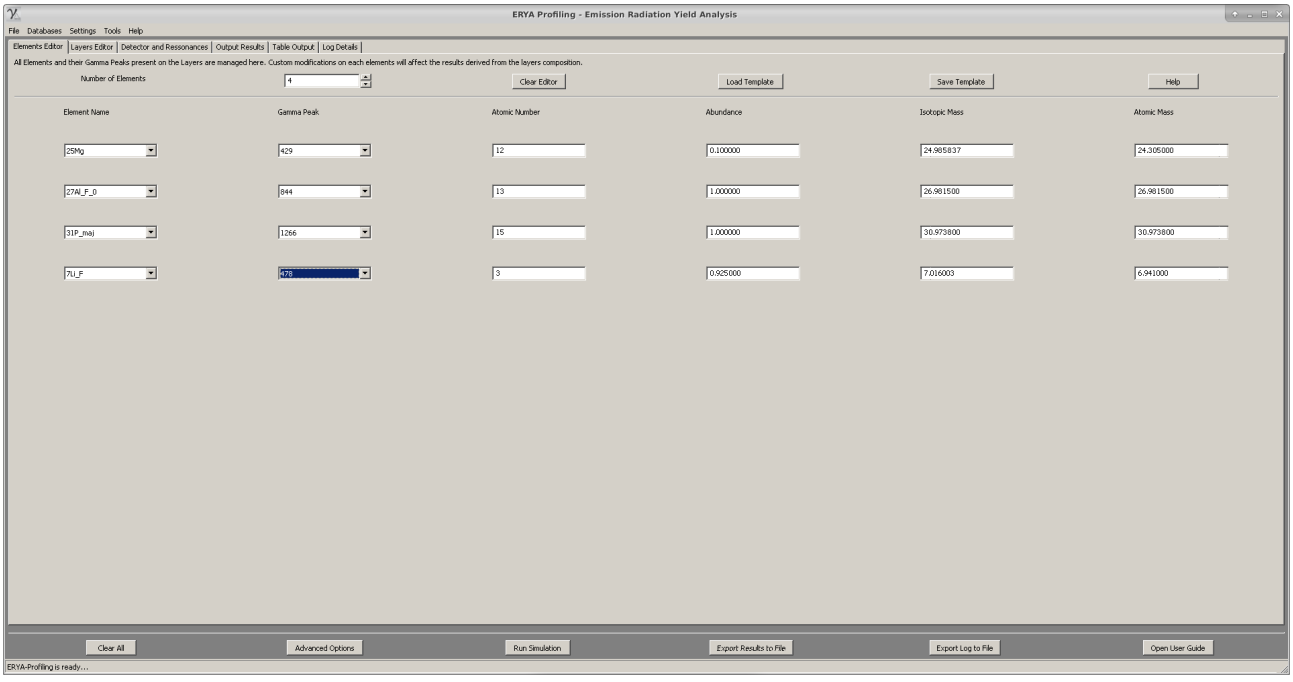
2. Once selected the adequate number of Elements, as notice that adding or removing won’t reset the remaining ones, change the Element Name from the pull-down inputs, and it will reload the Gamma Peaks pull-downs, and update the basic information from the Database.

3. Select the relevant Gamma Peak, as explained before.

4. Once selected all elements, if necessary, edit the basic Elements parameters, overridden the default values, without affecting the original values from the database. This feature can be useful on samples with abnormal isotopic abundances, and avoid unneeded Databases editions.

5. The current elements selection will update the tables from other tabs, as it will detail later.

Extra: It’s possible to store the custom sample composition on a template file. It can be a normal Excel file, or a native XML based file with extension **epcs**. And loading the stored file, will fill the current “Elements Editor” tab without filling all over again.



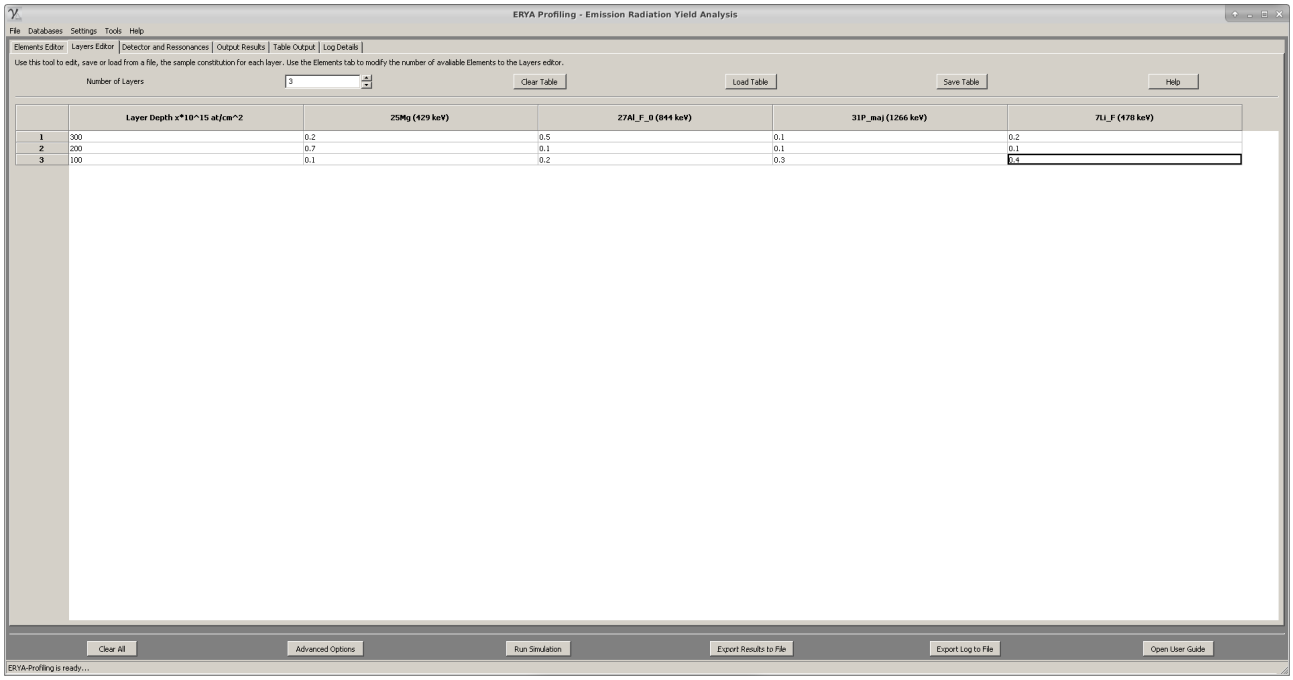
Second Step: Define the Number of Layers, and Detail their Composition

1. Select the “Layer Editor” tab, confirm the selected elements on the new table, and select the number of layers.

2. Define the depth of each layer on 10^{15} at/cm^2 units, on the first column, and then the relative stoichiometry of each element. The program will make an internal renormalization during the main numerical simulation, then there’s no need to comply a manual input as long their sum should be equal to one, since the ERYA will fix without any additional user intervention.

Extra: It’s possible to store the table contents on an Excel file, or a native XML file with **epls** extension. This may make more easy to avoid manual inserted of a standard sample, by saving the initial values as template, and loading again.

Notice that loading templates on the “Layer Editor” will reload the “Element Editor” to the default values.



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

An additional option, inherited from the LabView ERYA program, was to fill a text file with a sample composition and export directly, instead to follow the manual steps of First and Second Steps.

On ERYA-Profiling, this feature was greatly improved, while warns the main issues of each option.

The major improvement was the Excel file import of a template, as it will detail here.

1. Excel File Option:

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

Select any initial cell of an empty Excel file, but maintain a compact matrix of data, in order to avoid importation errors, due to a wrong matrix dimension.

On first column of the firs row, write any word, since it will be the depth column, and then place the elements on the rightmost columns of the same row, the names and gammas. Normally, just use the patter <Symbol> <Gamma>, separated by a space, like "*19F 197"* on a cell to represent the 197 keV gamma of the isotope 19F from the Database. Other variations are permitted, and are detailed on the on-line help reference.

Finally, write on the next adjacent rows, taking attention to assign correctly from each column, the depth and stoichiometry, like the built-in tale editor form ERYA-Profiling.

Once written, save as Excel 2007 Xlsx File Format, and then use the "Load Table" to select the new created Excel File. It should extract the data and fill the "Element" and "Layer" editor automatically.

It is necessary to take care to not select an non-existent element of gamma, or the program will warn about a non-existent isotope from the current database, and the user should open the Excel file on your favorite spreadsheet program, and make the necessary corrections.

It's possible to export a pure numerical table on an Excel file, but require the user to define manually the number and names of all elements, making more prone to mistakes.

2. ASCII File Option:

In this case, create a numerical matrix separated by tabs and newlines, taking care to maintain a coherent number of columns. Also the user should define the number of Elements, and their names manually, before export the table.

A mistake can get an import error, or a garbled table, since the program don't know the expected matrix dimension forehand.

Third Step: Define the Detector Parameters, and Resonances

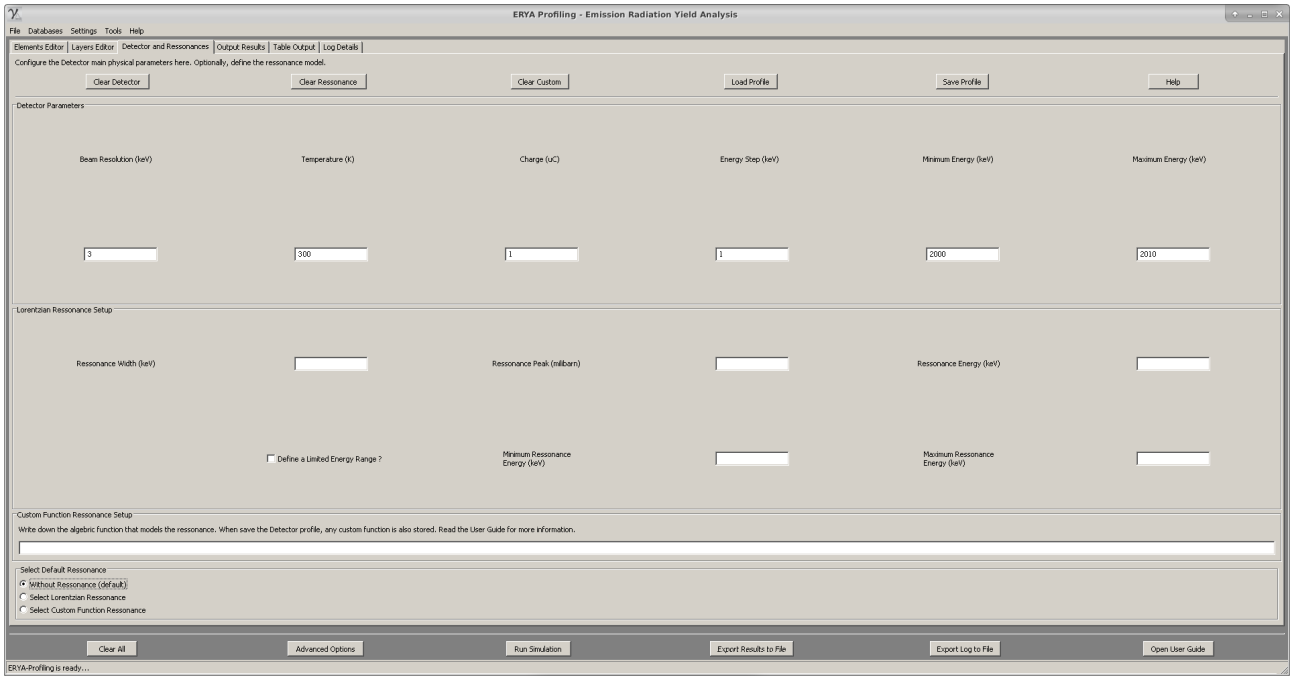
1. To complete the necessary data to start a simulation, select the “Detector and Resonances” tab.

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

3. Optionally is possible to define a Lorentzian Resonance, or even a custom Function Resonance using the same ERYA macro language on *Detector Efficiency* or on *ERYA Calculator*, but require to use all four reserved keywords for a function to be valid, as it expects a defined resonance at a restricted domain of energies. The on-line help contains major details about the ERYA macro language syntax and examples.

4. Select the main resonance mode, using the bottom control, where the default is to ignore any custom resonance at all, or choose the Lorentzian or Custom Resonance function.

Extra: It’s possible to save the current Detector and Resonance values on a XML file with **eprs** extension. Load it, will recover the stored values.



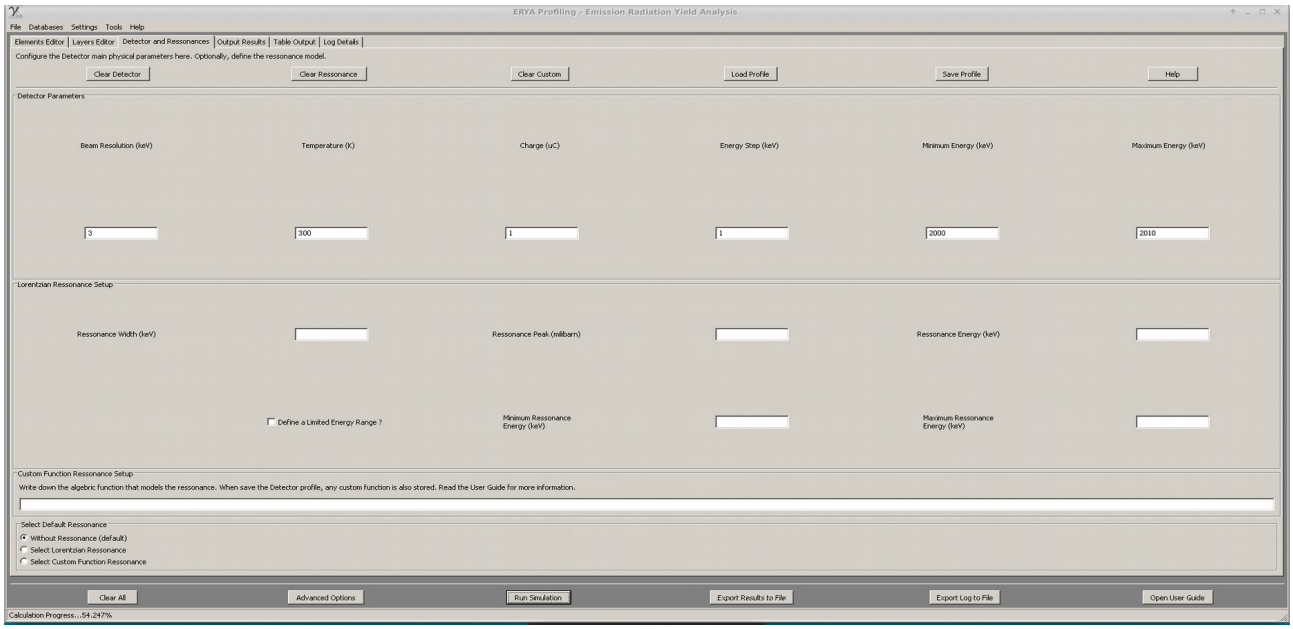
Additional Tip.

To save all contents of “Element Editor”, “Layers Editor” and “Detector and Resonance” tabs, along some additional values at once, use “File” > “Save” and save on a XML file with extension **epgs**.

When load a **epgs** file, with “File” > “Load”, it will load all three tabs contents automatically.

Last Step: Make the Simulation

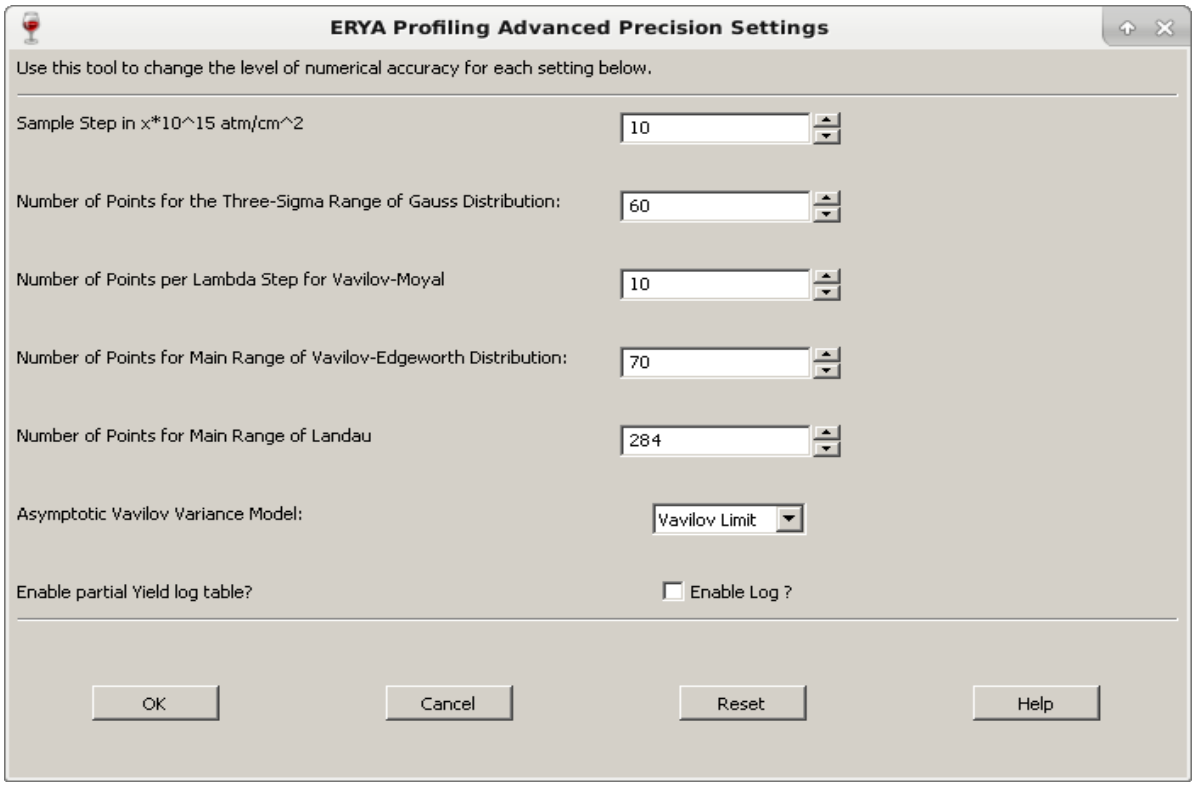
Once everything done, according to the previous steps, hit the “Run Simulation” button, and if no error are reported, the simulation will begin, displaying the percentage on the bottom status bar:



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

Changing Accuracy Parameters (Optional)

If the user knows what are doing, it is possible to change the accuracy parameters by clicking the "Advanced Options" button on the bottom main page controls, which opens a simple widget:



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

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

The next four parameters are related to the number of points of the intervals of integration related to the distributions, as explained on the brief theoretical chapter. The on-line reference contains some more detailed information.

The pull-down option is to select between the Vavilov or Bohr Variance, applicable when all distributions goes to the Gaussian Limit. An additional pure Gaussian option is also available. The numerical differences rarely make one to be great than the double of the other, and only on special contexts the evaluated yields will got a significant difference between the two modes.

On bulk samples, the results of both three options are the same, but it will computationally much faster if the "Gaussian Only" option are selected.

The checkbox option enables or disables the partial yields logging, which are filled on the “Log” tab. Disabling it can save time and memory, specially when an huge sample ends the simulation, and delays the outputs due to the processing of hundreds of thousands lines of spreadsheet data!

Import Experimental Data

Once the results are displayed on “Output Results”, mainly the graphical plot, it is possible to add additional plots of experimental data.

The current implementation only accept a single line per time, and only requires an Excel file with two columns of numerical data (the first row can be any labels, that will be ignored), or an ASCII filled with two tabbed columns of numerical data.

Then click on “Import Experimental Data”, and the correct file, and a new plot with a template name will be displayed, along the interpolated data on the “Table Output”.

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

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

Attention: Notice that the experimental data, once plotted on the cited tab page, will also appears on the rightmost columns of the “Table Output” tab, but interpolated to match the energy spectrum created by the numerical simulation.

Saving Results

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

At first glance, it's possible to save the current graphic plot as an image image, using the "Save As Image" button from the "Output" tab page, and then select one of the four images types supported by this program.

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

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

The log files of huge samples can creates tables with several thousands of lines, and ERYA-Profiling can take some minutes to save an Excel file with some hundred thousand rows. On this case, it is preferable to use the ASCII file. The ERYA's Excel library can crash with such gigantic volume of data.

Some Useful Tips

- Use the built-in tools described on this tutorial, when applied to the first three tabs, to fill the sample composition. It's much better than export from text or Excel files.

- Save all samples as **epgs** files form the "File" > "Save As", in order to create a collection of master samples. This is useful to use test different stoichiometries, once the file are load by the program, avoiding to fill everything manually. Instead, only a few values are changed at all.

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

- On samples with elements that have sharp resonances, reduce the Sample Step, in order to avoid loss of accuracy.

Example: A Sampling with a smooth cross-section: 10-Boron

- Fill a single layer with a single element with 3 million units of depth.

- Increase the Sample Step up to 1000. Later change the asymptotic model from Vavilov to Bohr, and finally Gaussian.

- Use a beam resolution of 3 keV, with 300 K, along a energy spectrum from 1000 to 3000, with 100 keV Step.

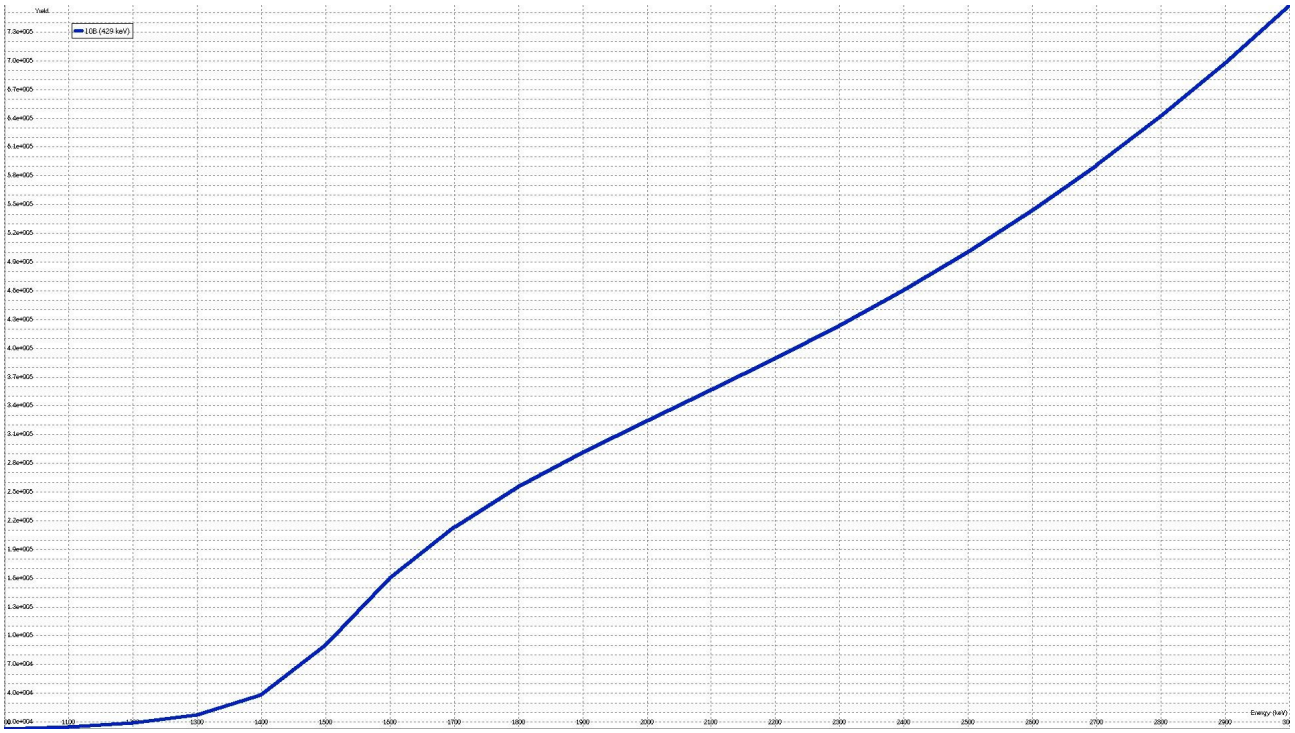
- If necessary, save the two experiments as **epgs** files, just in case to avoid manual typesetting for each experiment.

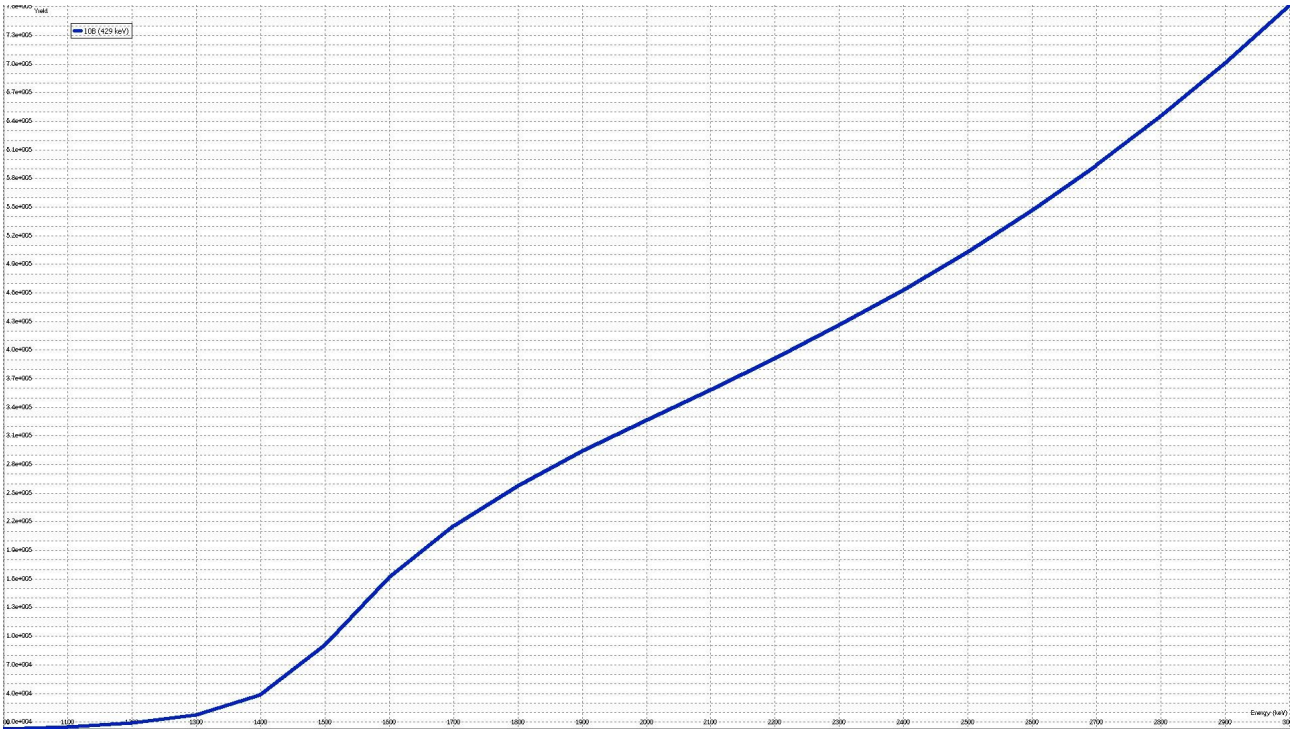
- Start the simulations.

Note: On bulk samples it could use the Gaussian option, since it will takes less than 10% of time than using the Vavilov distribution, and gives the same results!

Results brief analysis:

The simulated yields for Bohr and Vavilov limits, respectively, on a gross sample are almost indistinguishable:





The differences are less than 0.5%, and the plot are almost the same on the two models.

The tables are placed by the same order, first the Vavilov limit, then the Bohr limit.

	Energy (keV)	10B (429 keV)
1	1000.000000	2717.228415
2	1100.000000	5274.544323
3	1200.000000	9601.553814
4	1300.000000	17922.758726
5	1400.000000	38615.154252
6	1500.000000	90365.577384
7	1600.000000	160144.869170
8	1700.000000	213775.193351
9	1800.000000	256027.441638
10	1900.000000	292138.246598
11	2000.000000	324802.368436
12	2100.000000	356764.333349
13	2200.000000	389811.209142
14	2300.000000	424459.115374
15	2400.000000	461457.276585
16	2500.000000	501314.893491
17	2600.000000	544471.656737
18	2700.000000	591373.382015
19	2800.000000	642486.716552
20	2900.000000	698059.382653
21	3000.000000	758273.208885

	Energy (keV)	10B (429 keV)
1	1000.000000	2740.925196
2	1100.000000	5318.580543
3	1200.000000	9681.677921
4	1300.000000	18071.251579
5	1400.000000	38930.729757
6	1500.000000	91174.020336
7	1600.000000	162052.541618
8	1700.000000	216420.103288
9	1800.000000	258642.115446
10	1900.000000	294678.120413
11	2000.000000	327258.665583
12	2100.000000	359137.273657
13	2200.000000	392146.413576
14	2300.000000	426832.124868
15	2400.000000	463896.784290
16	2500.000000	503860.147319
17	2600.000000	547173.008336
18	2700.000000	594266.135610
19	2800.000000	645629.219719
20	2900.000000	701483.722185
21	3000.000000	762041.969388

Example: A Sampling with a thin AlTi layer

- Fill two layers layer with two elements: 150 units of pure Aluminum, and 200 units of pure Titanium. Here, the units are the depth in 10^{15} at/cm^2, as already acquainted.

Note: To do this, create two elements, and then two layers.

On the first layer, select 150 units for the depth, then a relative stoichiometry of “1” for Aluminum, and “0” for Titanium.

Finally, on the second layer, select 200 units for the depth, then a relative stoichiometry of “0” for Aluminum and “1” for Titanium.

- If necessary, reduce the Sample Step to 10. Also, make two experiments for the Vavilov and Bohr limits.

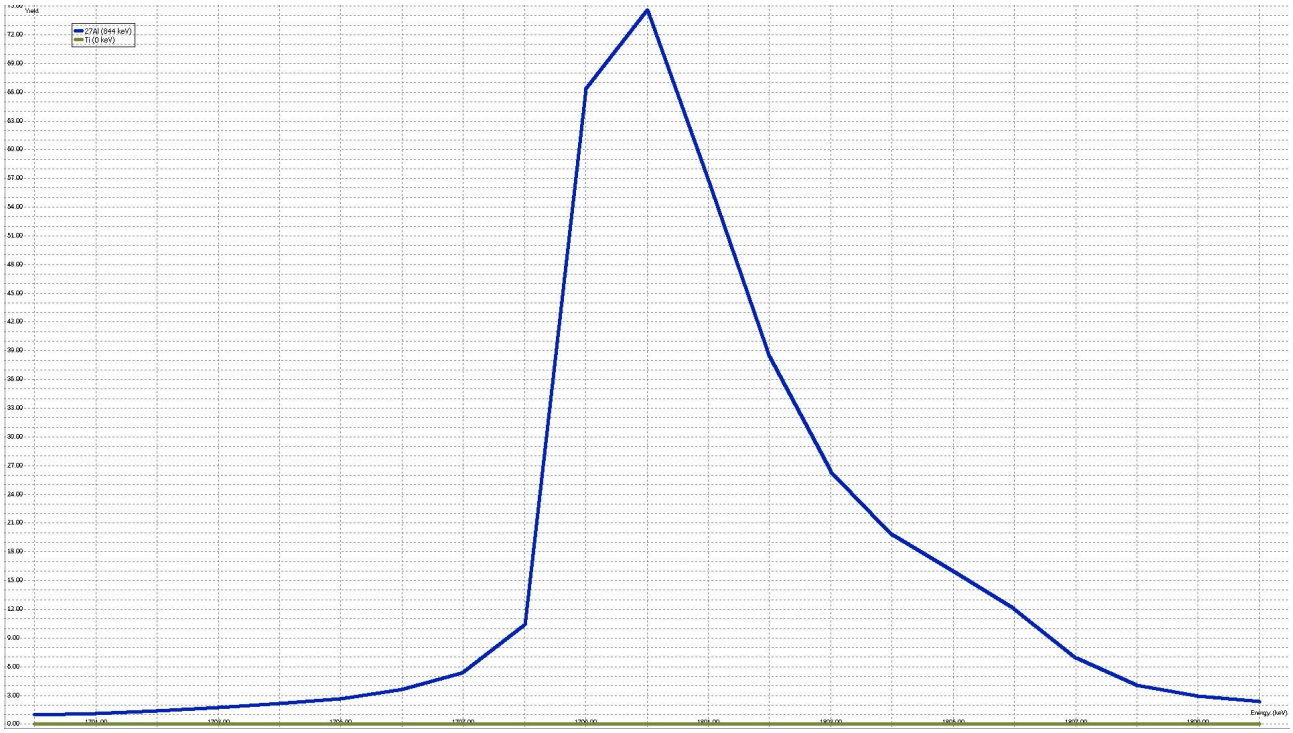
- Use a beam resolution of 3 keV, with 300 K, 1μC, along a energy spectrum from 1790 to 1810, with 1 keV Step.

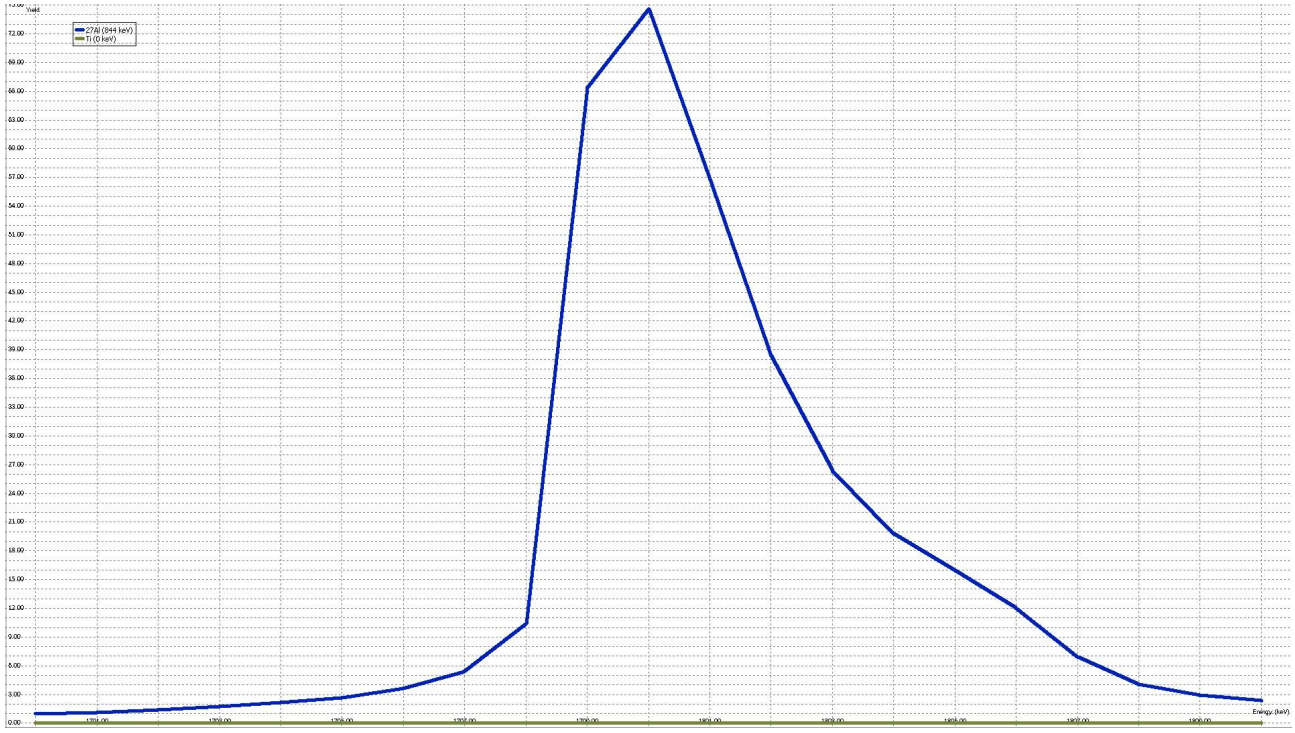
- Start the simulations. (and notice that the incident beam will never be totally absorbed…)

- Once complete, try to reduce the Energy Step to a smaller value, like 0.1 keV, in order to obtain a better plot of the resonance at 1800 keV. (It leave here as an exercise.)

Brief visualization of the Results:

Like the previous example, it will display the yield spectrum plot, first using the Vavilov limit, then the Bohr model:





And the numerical tables follows the same order: first, Vavilov, then, Bohr.

	Energy (keV)	27Al (844 keV)	Ti (0 keV)
1	1790.000000	1.030773	0.000000
2	1791.000000	1.162767	0.000000
3	1792.000000	1.438590	0.000000
4	1793.000000	1.799537	0.000000
5	1794.000000	2.209915	0.000000
6	1795.000000	2.650835	0.000000
7	1796.000000	3.674698	0.000000
8	1797.000000	5.426299	0.000000
9	1798.000000	10.370724	0.000000
10	1799.000000	66.359140	0.000000
11	1800.000000	74.613382	0.000000
12	1801.000000	56.802435	0.000000
13	1802.000000	38.367960	0.000000
14	1803.000000	26.345006	0.000000
15	1804.000000	19.756846	0.000000
16	1805.000000	15.915519	0.000000
17	1806.000000	12.011785	0.000000
18	1807.000000	6.889559	0.000000
19	1808.000000	4.039940	0.000000
20	1809.000000	2.896744	0.000000
21	1810.000000	2.340699	0.000000

	Energy (keV)	27Al (844 keV)	Ti (0 keV)
1	1790.000000	1.030773	0.000000
2	1791.000000	1.162767	0.000000
3	1792.000000	1.438590	0.000000
4	1793.000000	1.799537	0.000000
5	1794.000000	2.209915	0.000000
6	1795.000000	2.650835	0.000000
7	1796.000000	3.674698	0.000000
8	1797.000000	5.426299	0.000000
9	1798.000000	10.370724	0.000000
10	1799.000000	66.359140	0.000000
11	1800.000000	74.613382	0.000000
12	1801.000000	56.802435	0.000000
13	1802.000000	38.367960	0.000000
14	1803.000000	26.345006	0.000000
15	1804.000000	19.756846	0.000000
16	1805.000000	15.915519	0.000000
17	1806.000000	12.011785	0.000000
18	1807.000000	6.889559	0.000000
19	1808.000000	4.039940	0.000000
20	1809.000000	2.896744	0.000000
21	1810.000000	2.340699	0.000000

- Unsurprising, the plots are the same, because on thin sample, rarely, if ever, the Vavilov Distribution never went to the asymptotic Gaussian Distribution.

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).

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

- 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:

- “Numerical Analysis”, Richard L. Burden, J. Douglas Faires, Brooks/Cole Publisher Company.

- 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.