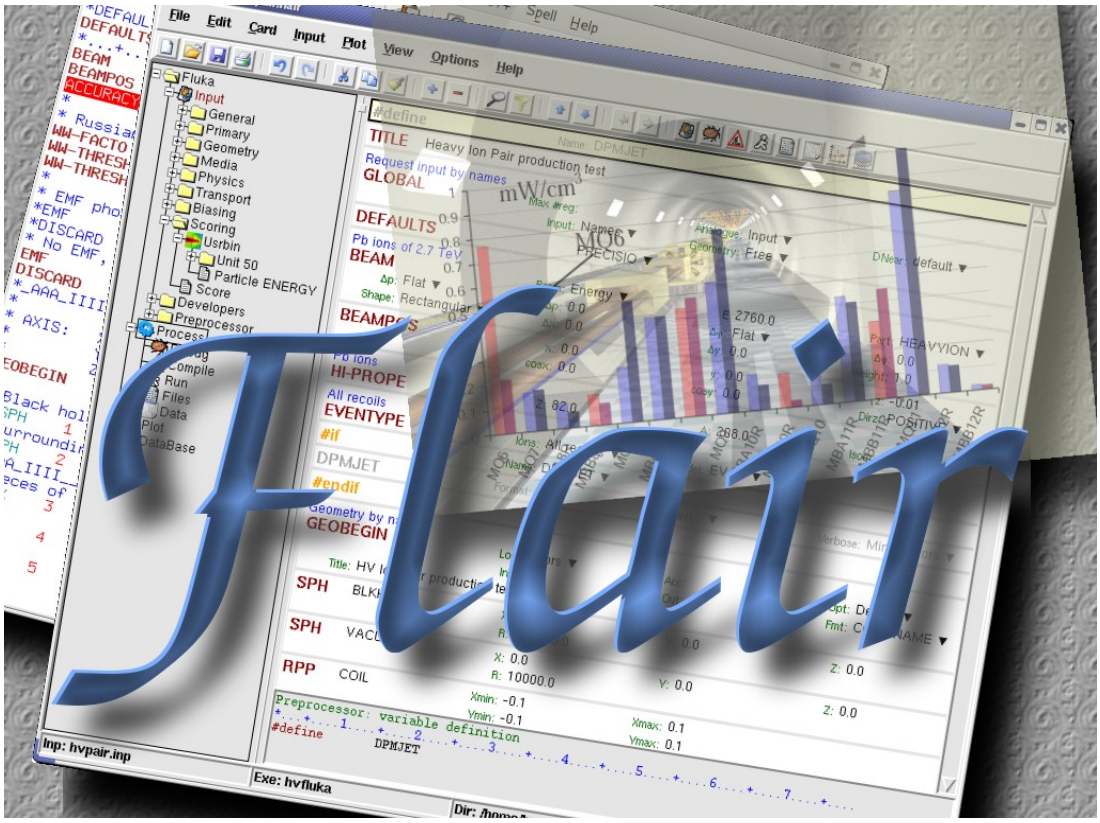


flair for FLUKA

/fleə(r)/ n [U,C][1] natural or instinctive ability (to do something well, to select or recognize what is best, more useful, etc.
[*Oxford Advanced Dictionary of Current English*]

author: Vasilis Vlachoudis
email: Vasilis.Vlachoudis@cern.ch
www: http://www.fluka.org/flair

version: 1.0
last change: April 2012
date: May 2006

1 n: noun, U=Uncountable, C=Countable

Contents

1. Introduction

flair is an advanced user interface for FLUKA (http://www.fluka.org) to facilitate the editing of FLUKA input files, execution of the code and visualization of the output files. It is based entirely on python and Tkinter. Flair provides the following functionality:

1. front-end interface for an easy and almost error free editing as well as validation of the input file during editing;
2. interactive geometry editor and debugger;
3. debugging, compiling, running and monitoring of the status during a run;
4. back-end interface for post-processing of the output files and plot generation through an interface with gnuplot and 3D photo-realistic images;
5. library of materials and geometrical objects, for easier editing, storing and sharing among other users and projects;
6. python API for manipulating the input files, post processing of the results and interfacing to gnuplot;

The philosophy of flair was to work on an intermediate level of user interface. Not too high, that hides the inner functionality of FLUKA from the user, and not so low that the user is in constant need of the FLUKA manual to verify the options needed for each card. Flair works directly with the input file of FLUKA and is able to read/write all acceptable FLUKA input formats. Inside the flair editor the user is working directly with the FLUKA cards having a small dialog for each card that displays the card information in an interpreted human readable way. The only exception is that the cards in flair are called "extended cards" where each card is not composed only by 6 *whats* and 1 *sdum,* but rather it contains all related information in one unit (comments preceding the card, continuation cards, titles etc).

For installation instructions please consult the README (http://www.fluka/flair/README.txt)

1.1. Project File

Flair is operating with the concept of the "*FLUKA projects*". A flair FLUKA project contains the following information:

1. general project information like: title, notes, override formatting options for the input file;
2. links to the filenames for the default input, optional geometry files and executable;
3. it will maintain a list of debugging regions;
4. links to auxiliary Fortran files and libraries for compiling a user FLUKA executable if necessary;
5. list of runs; For each run the user can change the input name, override the default preprocessor defines, title, random seed, number of start particles, run cycles and executable file;
6. list of output files and rules for merging the output of scoring cards;
7. list of user defined plots, for Geometry, all USRxxx cards, RESNUCLEi and visualization of the input file information i.e. graphical representation of the beam profile, particle thresholds, weight-windows etc.

All the above information is stored in an text file with the extension .flair, and is editable with the flair program.

1.2. Input File

Flair is able to read and write all formats recognized by FLUKA, but internally it works always

in the names format, and treats the input as a list of extended cards. The default format for saving is always fixed with names for the input and free with names for the geometry. The user can override the default exporting format either by the appropriate use FLUKA cards (like FREE, GLOBAL, GEOBEGIN) or by overriding the format in the project definition.

The positioning of the input cards is not so important, flair will try to reorganize the input file during saving and move all the geometry cards in the correct position. The only exception is that geometry cards (bodies, regions and lattices) controlled by the preprocessor outside the GEOBEGIN .. GEOEND block while be moved inside the geometry definition. While the preprocessor cards if any will remain at their initial position. All geometry cards that are correctly placed inside the GEOBEGIN .. GEOEND block while not be affected.

1.3. Extended input cards

All FLUKA cards in flair, are described by the extended input card (class Input.Card). Each extended card is composed by:

1. comment lines, the ones preceding the card definition, as well the in-line comments
2. a tag, which corresponds to the FLUKA card name. With a few additions: regions are defined with the REGION card. All preprocessor cards have tag names like #define, #undef, #if, ...
3. a variable number of whats. Starting from what[0] which corresponds to the sdum, what[1]..what[6] are the same with the FLUKA whats, what[7]-what[12] correspond to the what(1)-what(6) of the first continuation line etc.
4. Some cards what an extra field of information called extra or accessible as what[-1]. This is used to store long strings event multi-line information used by some cards, like TITLE, GEOBEGIN, PLOTGEOM or REGION
5. state of the card can be either enabled or disabled. Since cards cannot be commented with flair, the only way of excluding them from the input without deleting them is to disable. The program while place the card around an #if 0 .. #endif block. On the contrary all commented cards present in the input file, (with no space between the * and the card tag) will be converted to disabled cards.
6. All obsolete cards present in the input will be converted to the closest match if any, otherwise will be treated as error
7. All unknown cards will be converted to the card "error" and be disabled.

1.4. Card Grouping

FLUKA cards in flair are grouped into the following categories for easier accessing and more flexible editing of the input file:

Category	*Description*
General	Cards of general purpose (like TITLE, DEFAULTS, GLOBAL, etc..)
Primary	Cards dealing with the definition of the primary starting particles
Geometry	Cards related to the definition of the geometry bodies/regions/lattices plotting and rotations/translations
Media	Cards for the definition of materials
Physics	Cards defining physics properties for the simulation
Transport	Cards that modify the way particles are transported in FLUKA

Category	*Description*
Biasing	Cards for importance biasing definition
Scoring	Cards related to scoring
Developers	Cards reserved by the developers
Preprocessor	Preprocessor definitions for creating conditional input files

The user is encouraged to give a look first on the menu "Add" and browse the various categories to become familiar with the grouping.

1.5. Geometry Editing

Flair contains a graphical Geometry Editor as an additional plugin to flair that permits the user to visually inspect, edit, and debug the FLUKA geometry. It works with 2D cross sections of the geometry with some fast 3D ray tracing capabilities producing photo realistic images. At first glance 2D might seem as a handicap but is not a big problem since most of the objects are 2D and extruded in the 3rd dimension. However the strong points of the geometry editor is the instant geometry debugging, and the region definition directly in zones in an easy and graphical way without the need to type any mathematical expression.

The module can be started from flair by pressing the F4 button or from

Menu → View → Geometry Editor

2. Quick Start

2.1. n_TOF target

In the present section we will go through a step-by-step example and we will build from scratch without the use of the Geometry Editor, run and analyze the results of a simple FLUKA example.

We will simulate the neutron production and energy deposition of a lead target like n_TOF at CERN (http://cern.ch/n_TOF). The target has a rectangular shape of 80x80x60 cm^3 and is submerged in a water container with a ~5 cm layer of water that is used both for cooling and moderation. The neutrons are produced by a 20 GeV a proton beam, impacting with 10º angle on the horizontal plane.

To follow this tutorial, basic knowledge of FLUKA is required.

The user must make a choice on the coordinate system. The general tendency is to use the Z-axis collinear with the beam axis (usually lying on the horizontal plane), and then select the vertical and horizontal axis. In this example we will use the following convention:

X – horizontal axis, pointing to the left, with respect to beam direction

Y – vertical axis, pointing upwards

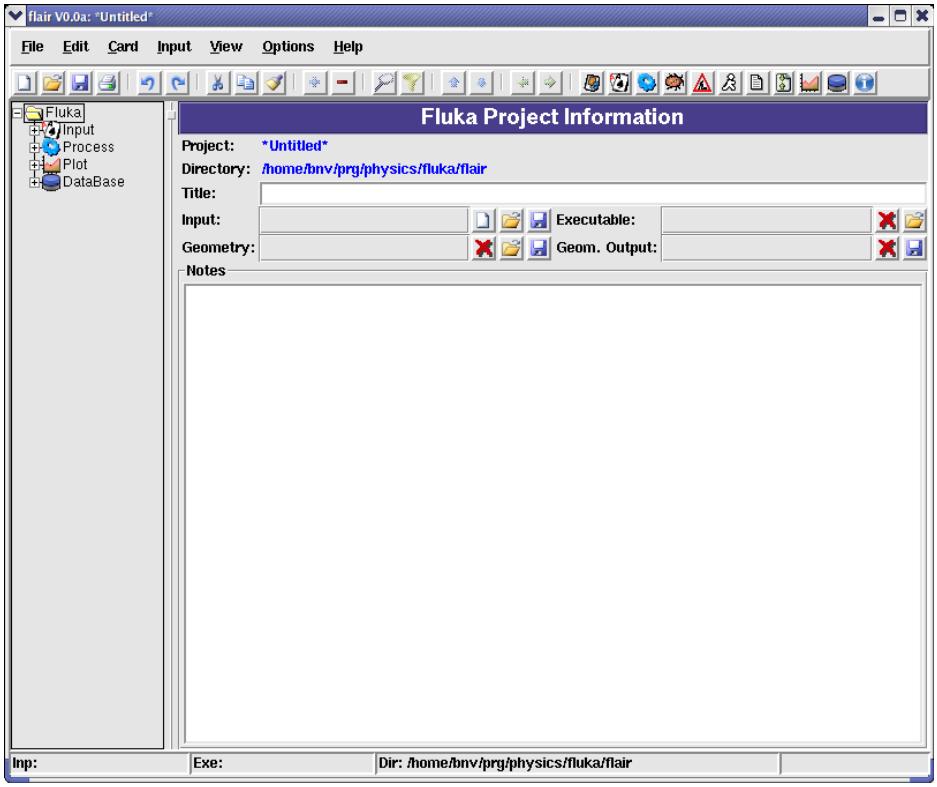
Z – beam axis, usually horizontal

2.1.1. Start a flair Project

- Launch `flair` by typing the following command, or by clicking the flair icon 

```
$ flair
```

The flair main window and the output window will appear:



On the flair main window one can find:

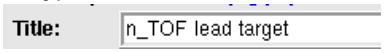
i. a menu bar on top
ii. a tool-bar with icons for fast access of the most common commands
iii. a status bar at the bottom to display some useful information for the current frame
iv. and three frames in the center separated by a horizontal and a vertical splitter:
    - on the left frame, there is a tree browser for the various sections of the project. By expanding and clicking on the appropriate node of the tree a different frame appears on the right side.
    - the right frames encapsulates all the project frames used for editing the information stored in the project file.

The window-title displays the program name, version and the name of the project we are currently are working on. Since we didn't specify any name it will show the word *Untitled*.

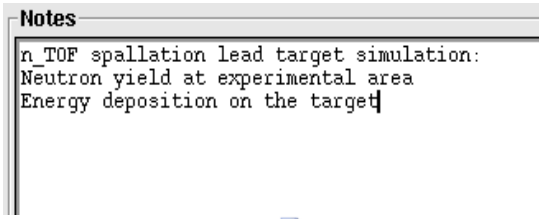
The Output window displays all information that are printed on the standard output and standard error unit.

> If you want to start flair without the output window, use the -X as command line argument `flair -X`

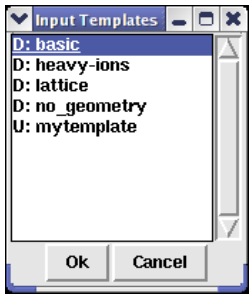
- We start by selecting the root node  on the left frame (it should be already selected). The right frame should show the general FLUKA project information, like title, input, geometry and executable filenames and notes.
- In the title entry field, type a title like:



- add also a small note describing the present project

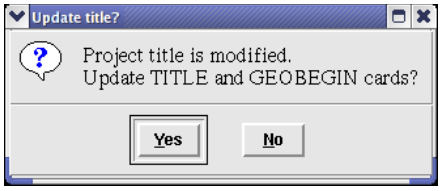


- Click on the button with the new icon (on the right of the Input Filename field) to create a new input file. The new icon on the toolbox creates a new project. A list box will pop up to select one of the various standard templates. Select the basic by double clicking on it.



> The dialog will show the default templates that exist in the flair package prefixed with the label "D:" and all the user defined templates with the label "U:". The user defined templates are located in the user directory ~/.flair/templates

- An information dialog box will appear asking if you want to update the TITLE and GEOBEGIN cards inside the FLUKA input using the title that was entered in the project. Click yes and automatically all FLUKA cards that require a title string will be set with the project title.



2.1.2. Edit the input file

- Next we move the Input editor by selecting and clicking on the tree the Input 

node . The right frame will change and the input editor will appear.

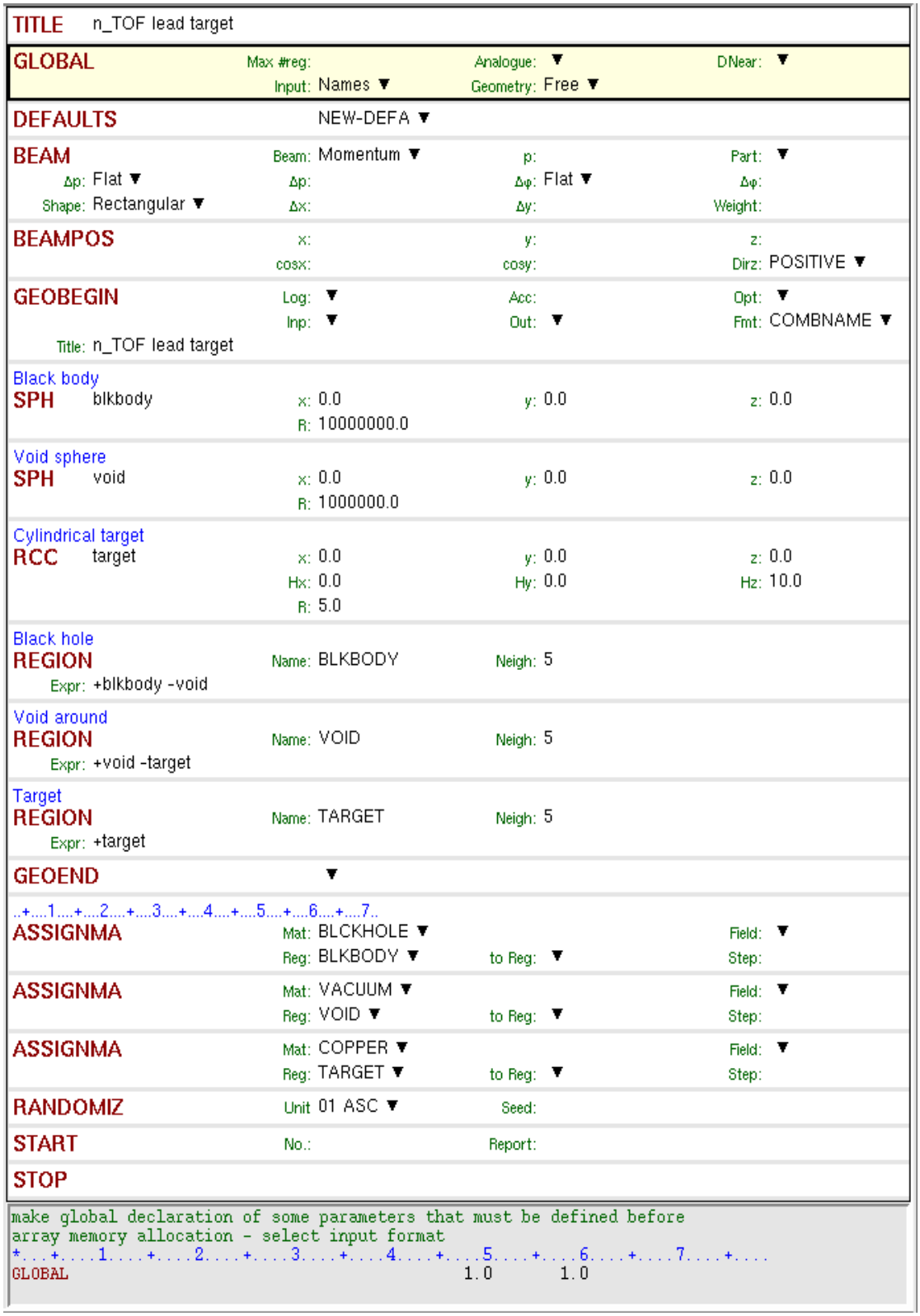
> The editor and all the project frames can be accessed with various ways:
> i. by selecting the appropriate node in the project tree;
> ii. with the function F2 – F9 keys. F3 is for the Input editor;
> iii. with the tool bar buttons 
> iv. From the menu item View

> During input editing there are two modes:

i. Card mode where you can manipulate the cards as a single object
i.e. Drag and drop, move, delete, insert, copy, paste...,
ii. Field editing mode, to modify the contents of a card.
To start editing the fields of a card, first select the card with the Up/Down arrows or Ctrl-F (Find) and press ENTER-key to start the field editing mode. To exit editing press ESCAPE-key, and you will return to Card mode.

- The editor now contains the default template, with the TITLE and GEOBEGIN card filled with our Project title. The selected cards are highlighted with a Light Yellow background color, while the active card has a thick black border around it.

We start editing the file by going through one by one the input cards:



We skip the cards TITLE, GLOBAL and DEFAULTS they are already filled with the default values to enable the input in Fixed format with Names and Free format for the geometry.

- Select the BEAM card and start editing either by pressing ENTER-key or by clicking

with the mouse on the appropriate field. Use the TAB-key to move to the next field:

- Select Energy on beam type. They next label will change to "E:"
- Type 20 at the beam energy labeled "E:"

> You will notice that the card display at the bottom of the screen will start to fill in with the values you typed, highlighting with yellow the changes from the previous state. Flair always converted the numbers into floating point format using the best representation of the number to ensure the maximum accuracy.

- Select PROTON as particle type.

> In all flair list boxes you can key-in the starting characters of the item you are searching and the closest match will be selected

- Select Gauss as momentum distribution from the field labeled "Δp"
- Type 0.082425 in the "Δp(FWHM)" as momentum spread (GeV/c)
- Select Gauss as angular distribution from the field labeled "Δφ"
- Type 1.7 in the next field "Δφ" as angular spread in mrad
- Type 1 in the field of the particle "Weight". The particle weight is omitted by FLUKA but the sign is important. Therefore is wise to set it always to one.

The card should look like the following:

BEAM	Beam: Energy ▼	E: 20.0	Part: PROTON ▼
Δp: Gauss ▼	Δp(FWHM): 0.082425	Δφ: Gauss ▼	Δφ: 1.7
Shape: Gauss ▼	x(FWHM):	y(FWHM):	Weight: 1

> At any time you can hit F1 to browse the FLUKA manual for the active card

- Fill up the BEAMPOS card with the following values:

BEAMPOS	x: 2.2632	y: -0.5	z: -10.0
	cosx: -0.17365	cosy: 0.0	Dirz: POSITIVE ▼

- Next we start building the geometry. To de-clatter the display expand the project tree (Left frame) and select the Geometry node 

. Now the right frame will show only the cards belonging to the geometry group.

The template generated a default geometry consisting of two concentric huge spheres named blkbody and void, and a cylindrical target named target. We will replace the target by a right parallelepiped named watercnt and add two parallelepipeds named pbtarget and niche

- Select the card RCC target either by clicking on it or using the up/down keys
- Select from the menu the command:

  Add → Change To → Geometry → Bodies → Rpp

> WARNING: Change To commands change the type of card, while at the same time trying to keep as much as possible from the whats. All exceeding whats in the new type will be discarded.

The above command will convert the RCC to RPP

- Start editing the card by pressing the ENTER-key.
  - Change the comment to Water container
  - Press tab to move to next field and change the body name to watercnt. The program will ask you to change the name of all references to body target renamed to watercnt. Click Yes.

When changing a body name, region name, material, or detector all cards that refer to this name will be changed also.

- Create two new RPP bodies. Either by right clicking and selecting

  Add → Bodies → Rpp

  or from the menu bar

  Add → Geometry → Bodies → Rpp

  or even by hitting Ctrl-Enter and selecting again

  Bodies → Rpp

  or by clicking the icon ✚ in the tool bar

To add a comment on a card, right click the card and select Insert Comment or from the Input menu select the Insert Comment. From the configuration panel you can change the default behavior and add a comment line by default on every newly inserted card.

Is always a good practice to avoid touching (co-planar) bodies in FLUKA. Either try to cut the bodies with the use of infinite planes, or slightly overlap the bodies and then performing the appropriate logical operation in the region definition.

- Now modify the TARGET region to WATERCNT. Click on Yes to the pop up dialog requesting a name change, and type the expression as shown in the next image. Create also a new REGION named TARGET as shown below.

Hitting the +, -, "Insert" keys or the icon ✚ while editing a REGION's expression shows a list of bodies to select from. Press the ESCape key if you are not interested in adding any body. This behavior can change from the configuration panel.

Lists in flair are search-able. Type the beginning of the item you are looking for and the closest match will be highlighted. Ctrl-N or Ctrl-G repeats the last search

- Now is time to create the water material. There are two ways:
  - create manually the necessary MATERIAL and COMPOUND cards, directly in the input editor;

- use directly the material from the Material Database and then import it to the input editor (recommended). Using the Material Database you can benefit from the pre-existing materials, and create or store your own materials to be exchanged between various input files.

To create manually the water material:

Select from the project tree the Input node  , to show all cards. Click to select the GEOEND card, in order to add a new card immediately after using the menu command

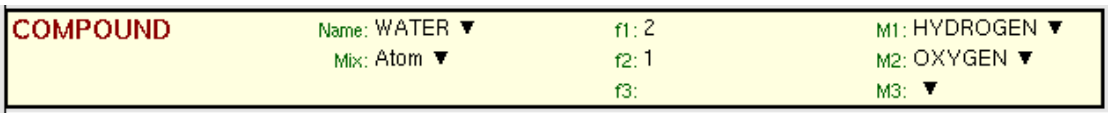
Add → Media → Material

Enter WATER as name and density of 1.0

- Add a new COMPOUND card.

Add → Media → Compound

Select the WATER from the material list, set the mixing type to Atom and fill up with the following values:



To create/ use the Material Database:

- Expand and select the node Material, which is found under the Database. The Material Database frame will appear. In this frame you can create or modify the material database, which is global for all flair projects.

  The materials are divided into groups where one material can belong to many groups. Each material contains:
  1. a title which should be unique
  2. some notes explaining the material
  3. a list of possible FLUKA names with the preference order in case of doubles
  4. the stoichiometric composition given as fraction in atoms, mass or volume
  5. density in g/cm^3
  6. Sternheimer parameters.

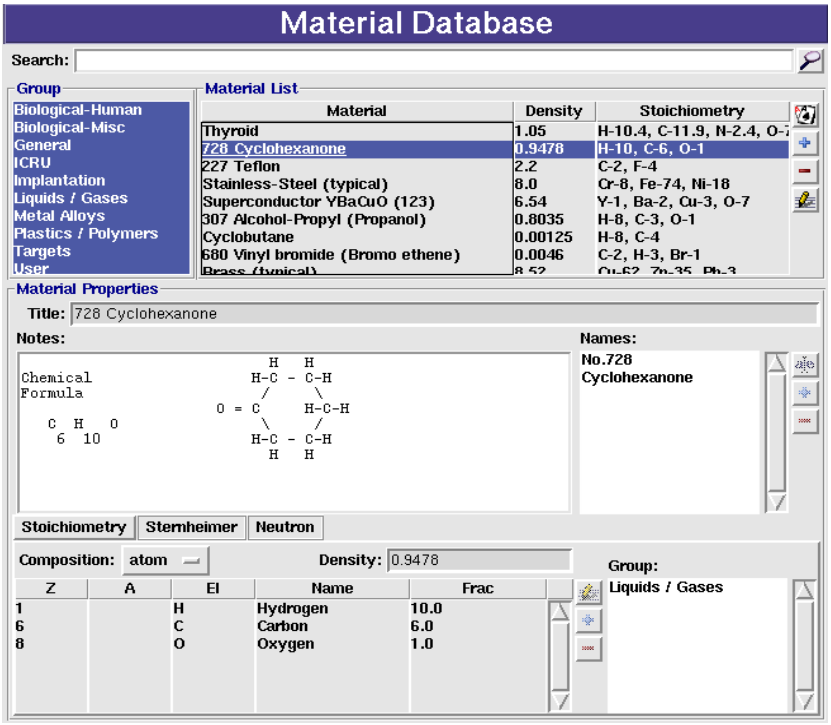
  By default all groups are selected

- Enter the word "water" in the Search: field and click the search icon or press Enter. The Material List will show only the materials in the selected groups matching the search string. A few choices will appear.
- Click with the mouse on the "Water (liquid)" to select the material and then click on the icon or from menu item "Tools → Insert to Input", or by right-clicking and select the "Insert to Input". The WATER material will be inserted after the last MATERIAL/COMPOUND/GEOEND card in the input. The action will be confirmed by a message box as well printing the inserted cards in the output window

  WARNING: The existing materials are there for reference. Please verify the correctness of the information.

- To assign the materials to the regions we need to modify and add the necessary ASSIGNMAt cards as shown below

ASSIGNMA	Mat: WATER ▼		Field: ▼
	Reg: WATERCNT ▼	to Reg: ▼	Step:
ASSIGNMA	Mat: LEAD ▼		Field: ▼
	Reg: TARGET ▼	to Reg: ▼	Step:



- After the COMPOUND card add a LOW-MAT card from the Media group, to specify that we want to use Self Shielded Lead for the low-energy neutron cross section.

USRBIN		Unit: 50 BIN ▼	Name: EneDep
Type: X-Y-Z ▼	Xmin: -45.0	Xmax: 45.0	NX: 100.0
Part: ENERGY ▼	Ymin: -54.0	Ymax: 54.0	NY: 100.0
	Zmin: -33.0	Zmax: 36.0	NZ: 100.0
Proton fluence			
USRCOLL		Unit: 51 BIN ▼	Name: Proton
Type: Log ▼	Reg: TARGET ▼		Vol: 1.0
Part: PROTON ▼	Emin: 0.001	Emax: 20.0	Bins: 100.0
Neutron fluence			
USRCOLL		Unit: 51 BIN ▼	Name: Neutron
Type: Log ▼	Reg: TARGET ▼		Vol: 1.0
Part: NEUTRON ▼	Emin: 1e-09	Emax: 20.0	Bins: 100.0
RESNUCLE	Type: All ▼	Unit: 52 BIN ▼	Name: Target
Max Z:	Max M:	Reg: TARGET ▼	Vol: 1.0

- Add a few scoring cards, a USRBIN, USRTRACK and RESNUCLEi as shown above
- Set some primary particles in the START card for a test run

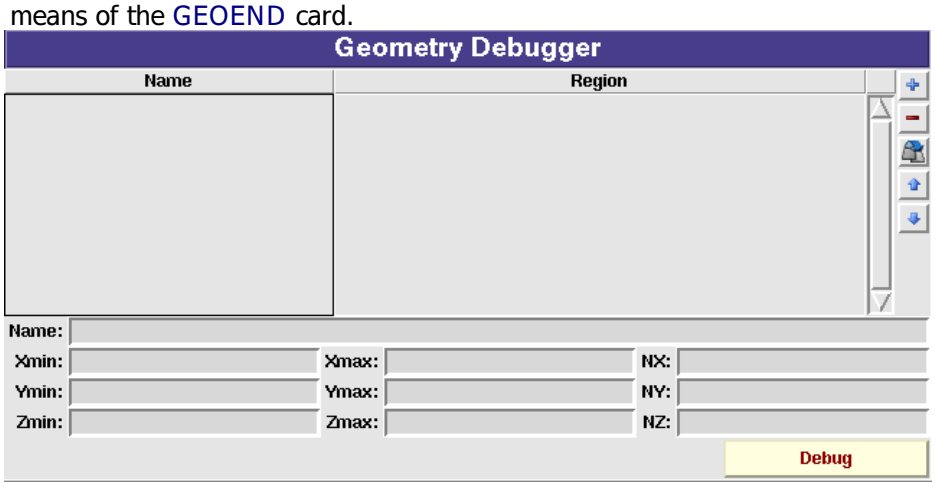
START	No.: 100.0	Report:

- Finally click on the save icon from the tool bar. Flair will ask you first to save the input file, use tutorial.inp; and then the project file, use tutorial.flair.

2.1.3. Debugging the input file

- Select the Debug node which is under the Process Debug or by clicking directly the F4 key. The Geometry Debugger will appear. This frame is composed by a listbox

and a few entry fields where the user can add various debugging regions. Debugging regions in flair have nothing to do with the regions defined in the geometry of FLUKA. They refer to a volume in the geometry to be scanned for geometry errors with means of the GEOEND card.

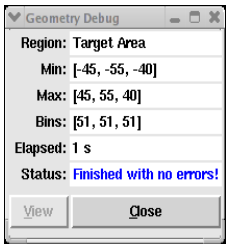


- Start by adding a new debugging region by clicking the add button or the Insert key.
- The region will be automatically named as Region #1. Select it and enter the following information in the fields below. To define a region around our target scanned in a grid of 51x51x51 steps.

Name:	Target Area				
Xmin:	-45.0	Xmax:	45.0	NX:	51
Ymin:	-55.0	Ymax:	55.0	NY:	51
Zmin:	-40.0	Zmax:	40.0	NZ:	51

It is always a good practice to use odd numbers in the number of steps, to decrease the probability of checking a position falling directly on a region boundary.

- When ready click the DEBUG button (or Ctrl-Enter) to start debugging. Flair will build a temporary input file containing only the geometry information and launch a FLUKA run. A dialog will pop up to monitor the progress of debugging. The dialog will show the current region being debugged elapsed time and status. When the run is finished, if there are errors in the geometry, click the View button to display the output of the Run, otherwise if the geometry is error free, the only available option



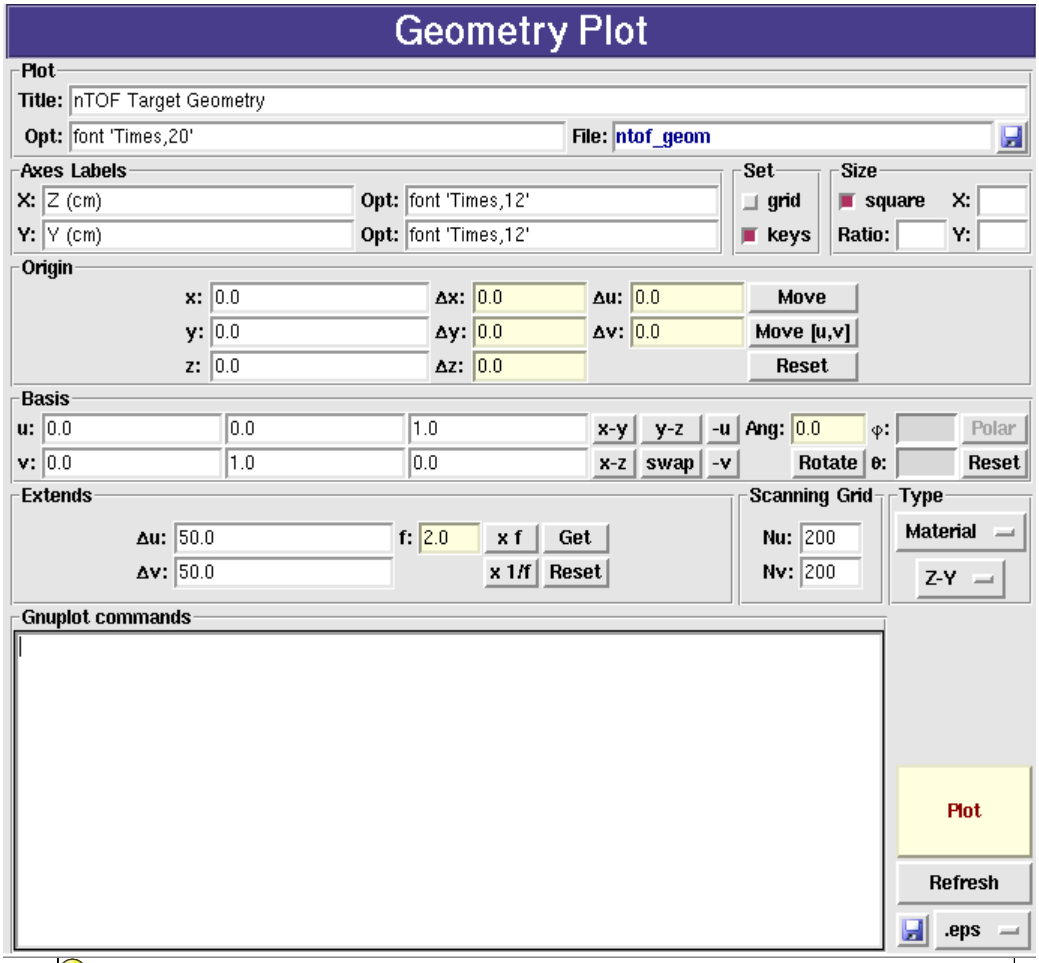
will be to close the window.

2.1.4. Geometry Plotting

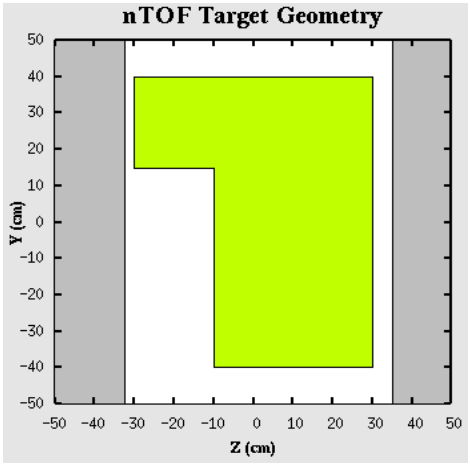
- Select the Plot  node under Fluka, or by clicking directly the F9 key. The Plot List frame will appear.

The frame contains a list of all the plots that are associated with the project.

- To create a new plot click on the + button. Now the Title will contain the string "Plot #1" and the the File will be "plot001" with format .eps.
- Change the title field "*nTOF Target Geometry*"
- Click on the save icon + after the File field at the bottom of the frame, to change the filename to "*geometry*". This filename will be used for all auxiliary files that will be created during the plot.
- Select and double click the geometry plot entry from the listbox, or from the project tree on the left frame. The Geometry Plot frame will appear.
- The title and the filename should be already filled and the coordinates are set to the axes origin.
  - Click on the Size frame "square" to get a square aspect ratio
  - Click on the button labeled Y-Z in the Basis frame
  - and then click on the Swap button. This will set the plotting axis to Z for horizontal and Y for vertical.
  - Set the value 2 in the f: field of the Extends frame and click the 1/f button to reduce the plot extends to 50 cm.
  - Select the Type Material to be used for the coloring of the regions. The dialog should resemble like the following figure
  - Optionally fill the Axes labels and the "Opt:" fields on the labels to select a different font, color and/or other option from gnuplot. Please look the gnuplot manual for the available options.
- When ready click the Plot button (or press Ctrl-Enter). A temporary input file will be created containing the geometry and a PLOTGEOM card. Flair will launch FLUKA with this temporary file and convert the output PLOTGEOM.STORE file into a file with a gnuplot acceptable format. Finally the gnuplot will be called to plot the geometry.



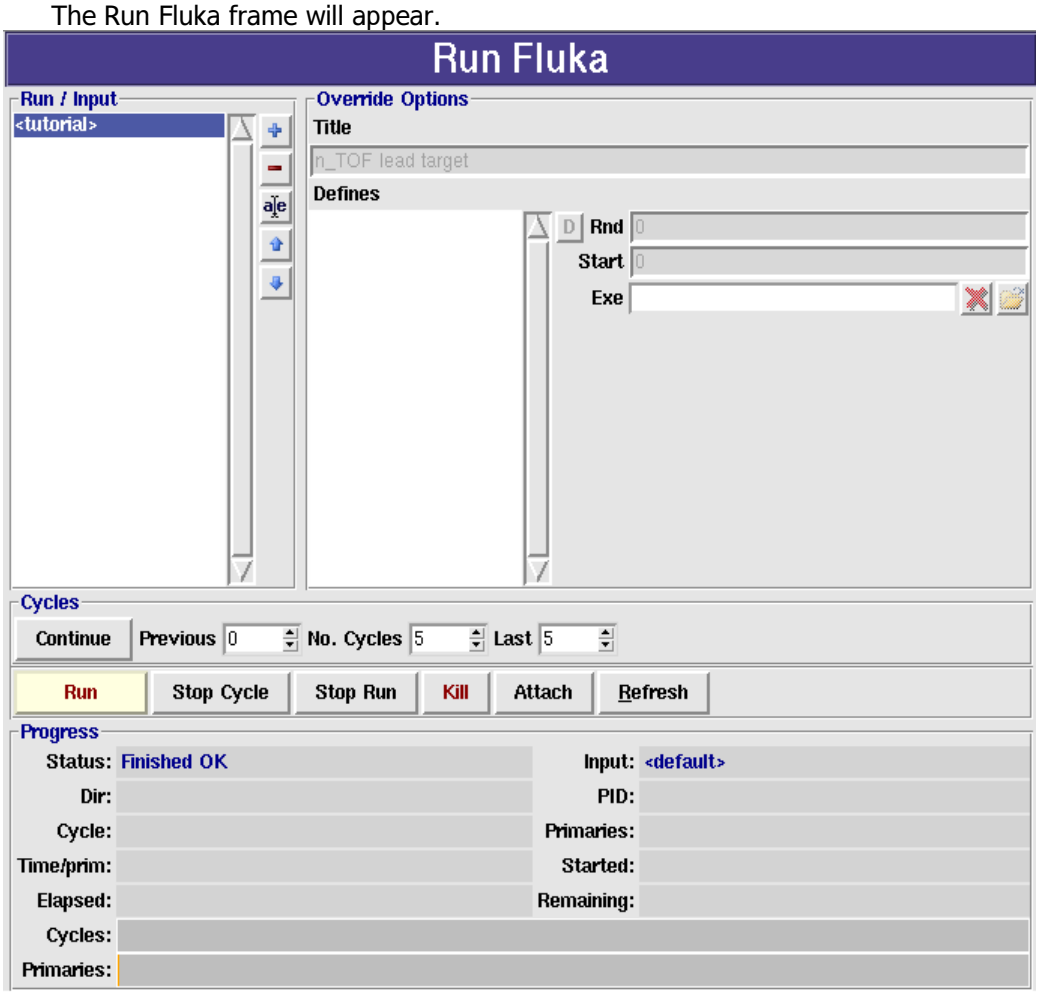
When changing the axes system flair tries to select the most appropriate representation in the Type frame. The user is free to modify the plotting system based on his needs.



2.1.5. Running the simulation

- Select the Run 

node under the Process, or by clicking directly the F6 key. The Run Fluka frame will appear.



The top-left listbox shows the runs associated with the present project. All runs share the same input file, and for the additional runs there is the possibility to tweak some of the contents of the input file like:

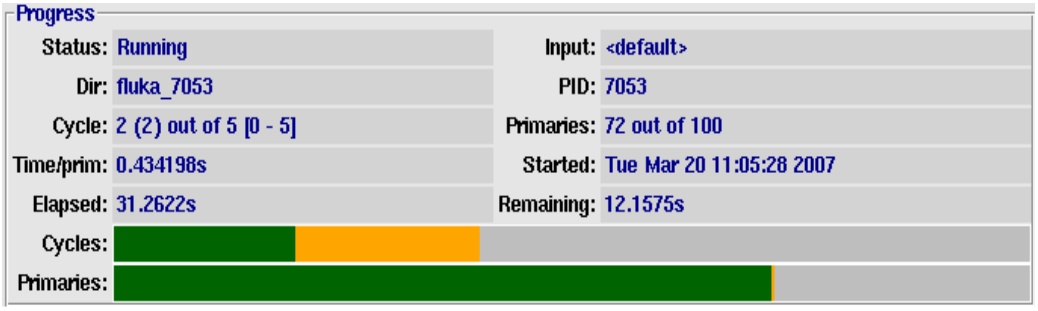
1. run title
2. preprocessor defines to enable/disable blocks of cards
3. random seed
4. number of starting primaries
5. executable file to use

There is already a <tutorial> Run, in which we cannot override any option. This is the run using the input file as it described directly in the Input editor.

- Click the Run button (or Ctrl-Enter) to start the run with the default 5 cycles.

The dialog below the control buttons while change and it will be refreshed every half a minute. During the execution the "Status" will change, initially to Waiting to attach, followed by Running and finally in Finished OK. You can even click on the Refresh button to refresh at any time the progress information. The run is submitted using the defined submit program (the default is nohup). The program is running decoupled from the flair editor, therefore if you click save on the project and exit the program. The next time you will

open the program, flair will try to attach and display the current run status.



Flair is trying to peek the run information only by looking the status of the output files. It doesn't make use of the system process information. This way it increases portability across different platforms, and batch systems (see the qfluka example for a substitution of the submit command). Flair will be able to monitor the status only if the run takes place on the same directory. The drawback of this method is that takes some time to attach.

The Stop Cycle/Run button will try to make a clean stop by creating a fluka.stop or rfluka.stop file in the running directory.

The Kill button will try to issue a kill -SIGHUP command to the running process. If the user is using a batch system he has to substitute the kill command with the appropriate program

The Attach button tries to re-attach to the running process in the case that flair failed to attach correctly, or lost the running status.

The Refresh button can be used to update the progress information at any interval.

2.1.6. Viewing Output Files

During the FLUKA run or when a cycle is completed, the user can inspect and/or delete the output files generated by FLUKA from the "Files" frame.

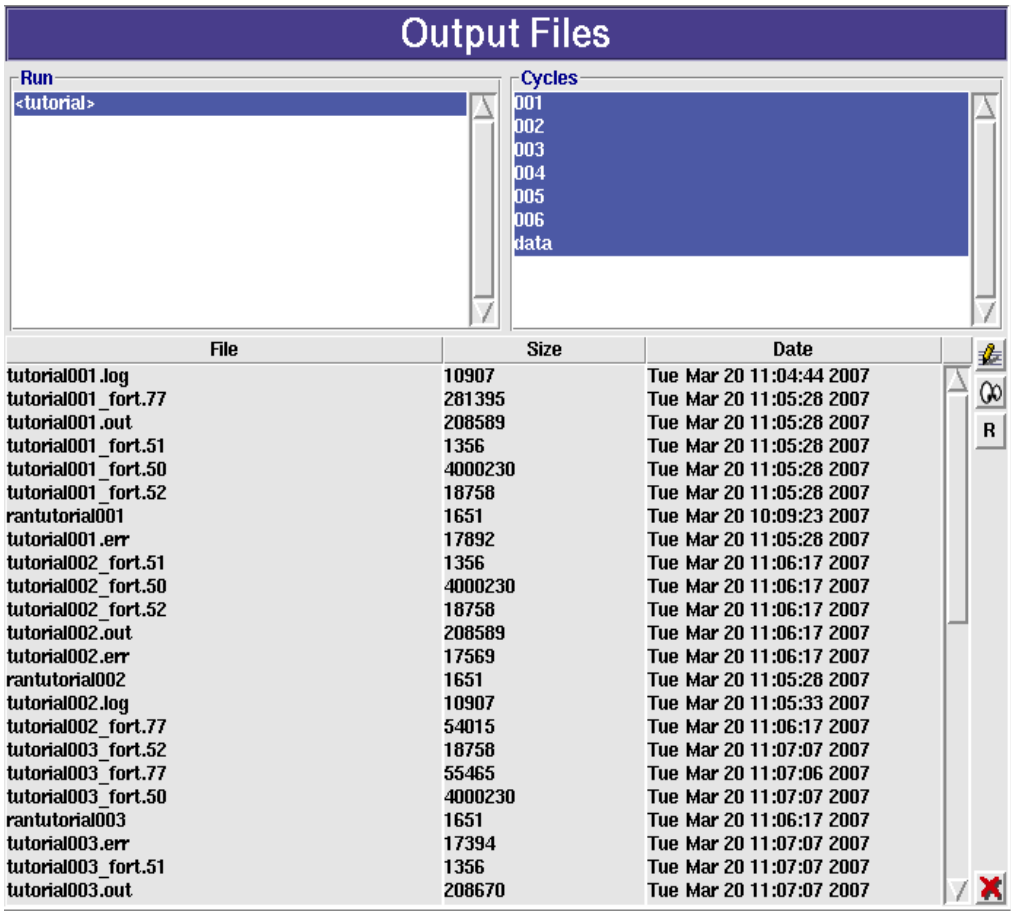
- Select the Files 

node under the Process, or by clicking directly the F7 key. The Output Files frame will appear.
- The frame is composed by 3 list-boxes and a couple of action buttons on the side. The first listbox with the label "Run", contains a list of the different runs present in this project. In our case only one run will appear named "<tutorial>" which is the default run with the input filename as it is described in the input editor.
- By selecting a run from the list (The <tutorial> is already selected) the other two list will be updated accordingly. The second list called "Cycles" contains all the cycles that are performed and output files exists for the specific run. The last cycle called "data" contains all the files that are generated by mering the output files (see the section on processing the data). By selecting one or multiple cycles the files are updated accordingly.

Shortcuts in listboxes: Ctrl-A selects everything; Ctrl-I inverses the selection; Ctrl-C clears the selection; typing the beginning of an item the

> listbox will highlight the closest match. Ctrl-G or Ctrl-N repeats the last search. In the multi-column listboxes by clicking on the header sorts the current column Ascending or Descending.

The last multi-listbox contains a list of files for the run & cycles selected in the previous listboxes.



- By double clicking on any of the files i.e. tutorial001.out, the file will be opened in the internal viewer of flair for inspection. The action is equivalent to selecting one or several files and clicking the eyes icon. By clicking on the editor icon the file will be opened on the user defined external editor. The button [R] (hotkey Ctrl-R) is used to refresh the display.

> WARNING: Do not try to open binary files. Could be rather huge for the editor or viewer and in any case incomprehensible.

2.1.7. Data Merging

The next step before is to merge the output data files of the run in order to create the files contain the average values and the statistical error.

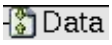
One would expect that the simulation is equivalent to a counting experiment, therefore the data will follow a Poisson distribution and the error will be the square root of the number of events collected. This is true provided that no biasing is used in the simulation. When importance scoring is involved (quite typical and recommended way of working) to calculate correctly the statistical error, apart from the final value one has to record also the square

number of events/hits for every value needed. This doubles the memory and increases the complexity for special estimators. Therefore, FLUKA is making use of the Central Limit Theorem for calculating the mean value of a quantity scored and the error on the determination of the mean. The theorem states:

> The distribution of an average tends to be Normal, even when the distribution from which the average is computed is decidedly non-Normal.

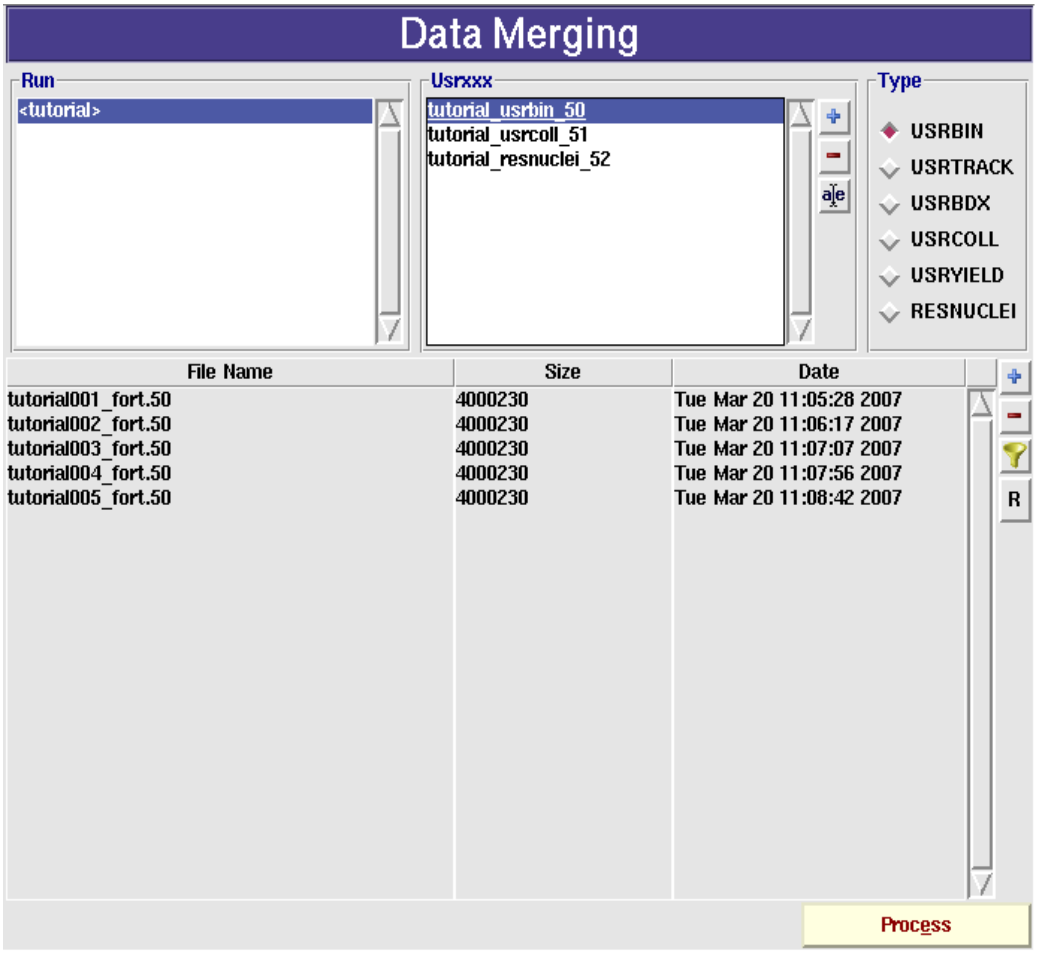
This is the main reason we have to perform several cycles, minimum 5 is recommended to simulate correctly a Normal distribution, and then sum-up and average the results. In FLUKA this is done automatically with the us?suw utilities (where ? can be: b=USRBIN, r=RESNUCLEi, t=USRTRACK or USRCOLL, x=USRBDX, y=USRYIELD). These programs expect as input a list of binary files generated from FLUKA with the respective card and using as unit a negative number, and in the end they generate a set of output files both binary, text and tabulated with the results.

Flair makes this process transparent to the user with the Process Data frame.

- Select the Data 

node under the Process, or by clicking directly the F8 key. The following frame will appear.

  The frame is composed with the following:
  1. a listbox for the runs described in the flair project
  2. the list of scoring units (Usrxxx) requested for each run. Automatically a default name will be assigned in the form of "input-name"_"card"_"unit". The user with the buttons on the right can delete the file, create a new one or rename it.
  3. Radial selection buttons (Type) for changing the type of the card that the file is referring to. Normally the program selects the correct one based on the input file. A multi-column listbox with the list of files corresponding to each summary file in the Usrxxx listbox.
  4. With the side buttons the user can add/remove or set a multiple filter rule to modify the list of data files.



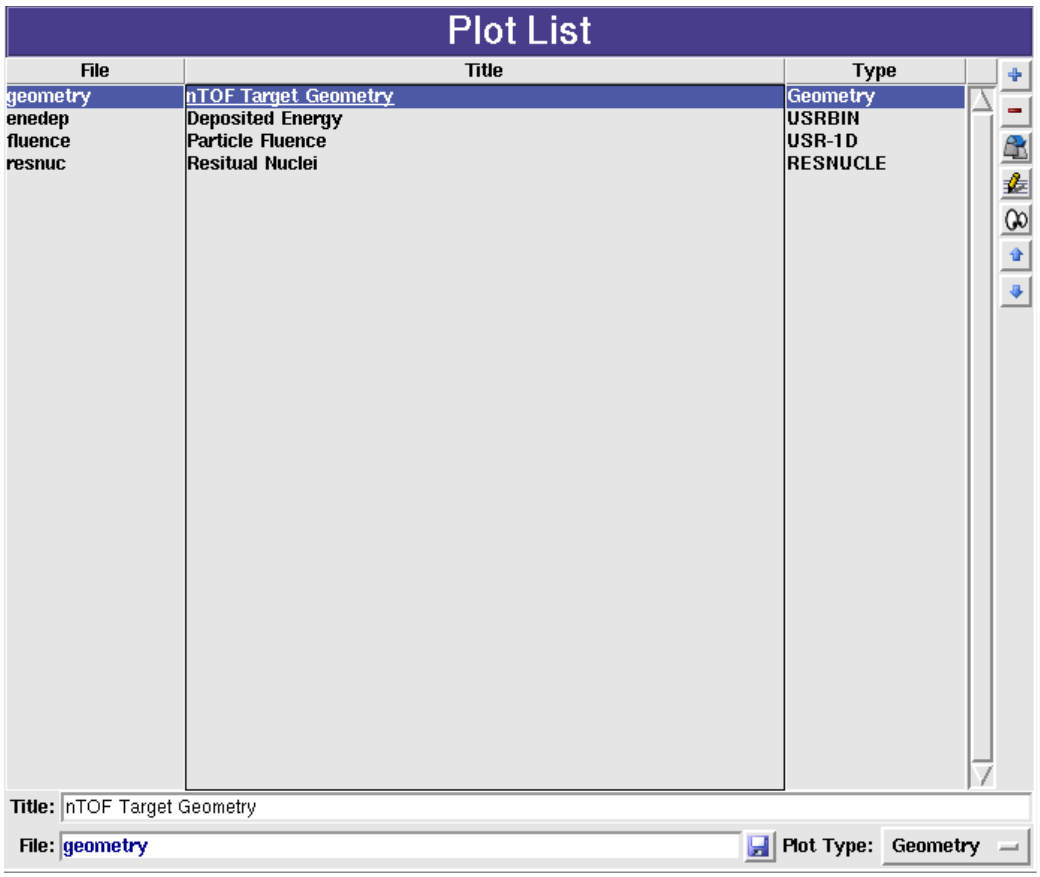
- By pressing the Process button (or clicking Ctrl-Enter) flair will run the appropriate FLUKA utilities to merge the data files. A dialog will confirm the outcome of the operation.

> The FLUKA utilities usually generate more than one output files. Typically the merge binary data file has the requested name while for a text file is generated with the extension _sum.lis, and a tabulated one with the extension _tab.lis

2.1.8. Data Plotting

- The last step is to plot the data, we will create 3 plots one for the USRBIN file that contains the energy deposition on the spallation target, one for the USRCOLL estimators with the particle fluences and one with the RESNUCLEi card containing the residual nuclei produced. We start by creating a plot in the "Plot" frame (F9) like what we did for the geometry plotting.
- Click on the + button, a new plot will be added to the list with Title "Plot #2" and the File will be "plot002" with format .eps, and type "Geometry".
- Change the title to "Deposited Energy"
- the file to "enedep". The filename is important since all auxiliary files and the final plot will share the same filename. The extension will only change.
- Finally set the type to "USRBIN"

- Repeat the process until you have create the following plots:
  - File: geometry Title: nTOF Target Geometry Type: Geometry
  - File: enedep Title: Deposited Energy Type: USR-1D
  - File: resnuc Title: Residual Nuclei Type: RESNUCLE

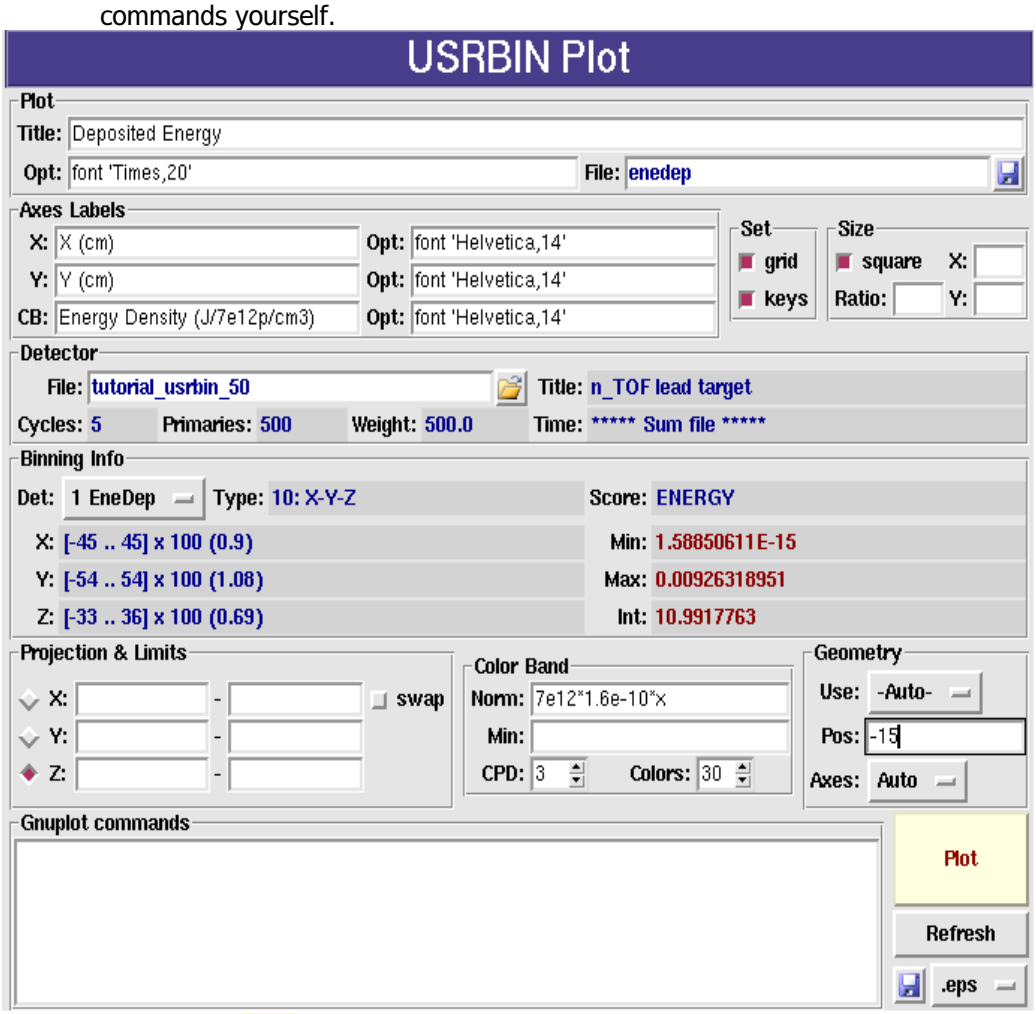


USRBIN Plotting

- By double clicking on the list or clicking the edit icon, the USRBIN Plot frame will appear. By default the Title, and File will be filled in with the values we filled in the Plot List frame.
- Fill in the values as you see in the following image.
  - Opt: under the title to supply additional gnuplot options to the title like changing the default font, color etc.
  - Axes Labels, supply axes labels and additional options if needed
  - Size, click on the square to have a square aspect ratio
  - Detector click on the button and select the "tutorial_usrbin_50" file we created in the Process frame. Automatically the run and detector information in the file will be filled.
  - Binning Info select from the drop down list box the first detector
  - Projection & Limits select projection on the Z-axis without suppling any limits. This will generate a projection from -33 to +36cm on the XY plane.
  - Color Band select as normalization the formula "7e12*1.6e-10*x", this way every value will be converted from GeV/cm^3/p to J/cm^3/pulse where a pulse has 7e12 protons. Select the Minimum plotting value and colors per

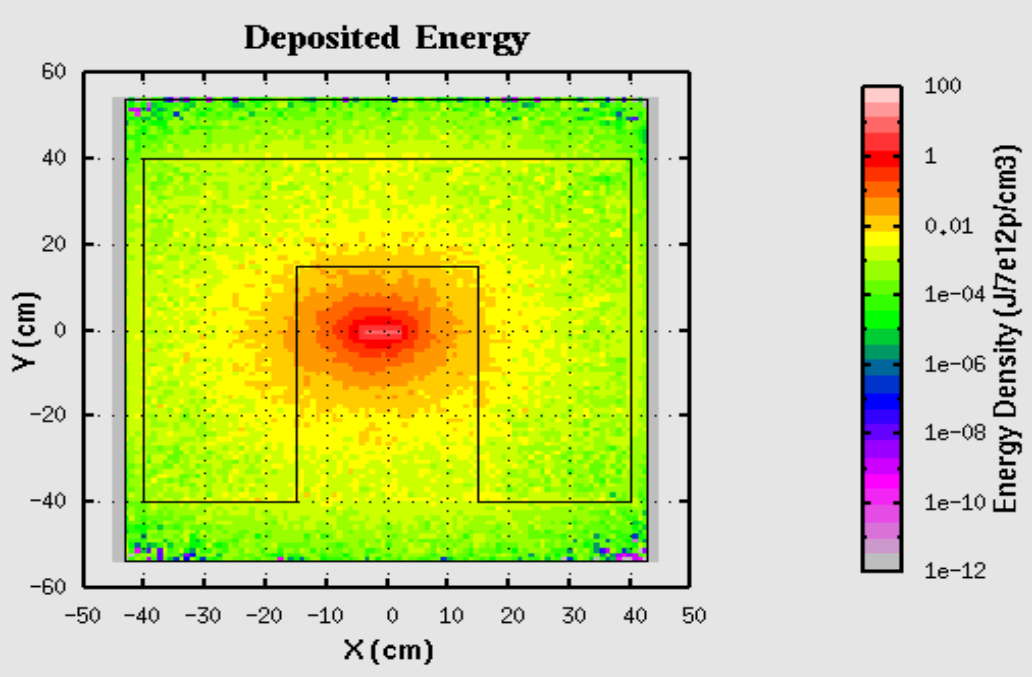
decade (CPD) and the total number of colors as you wish.

- Geometry use the automatic generator for the geometry at position -15. If we don't supply a position the mean value of the projection limits will be used. In this case the binning is directly fitting the box of the spallation target and the geometry will be empty. Therefore we move the position of the geometry to somewhere that makes sense.
- Gnuplot commands you can leave empty or specify additional plotting commands if you don't like the default ones. i.e. set palette.. to change the default color palette, set arrow ... to generate arrows to point regions of interest, set ?range... to change the default plotting ranges, and in principle any gnuplot command. Even "reset" and supply all needed commands yourself.



- By clicking the Plot button (or Ctrl-Enter), flair will create a temporary input file for FLUKA with the appropriate PLOTGEOM card, will perform a FLUKA run to create the geometry information, and then process the USRBIN data file with the gplevbin program to create the requested projection. Finally it will call gnuplot to create the plot. Once the plot is created you can save it as image by clicking the button at the bottom-right part of the frame.

Prefer to use the .eps format for higher quality figures. The .png format for gnuplot up to version 4.0 is broken and the color band will not appear properly.

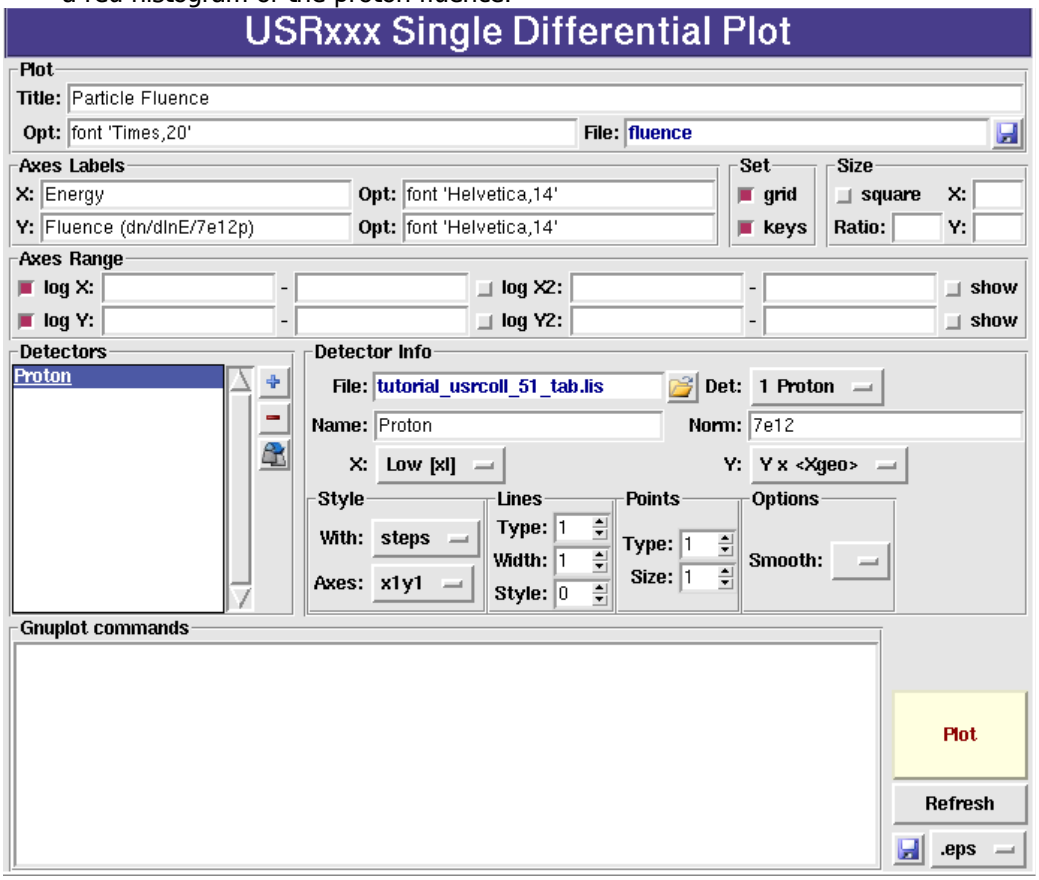


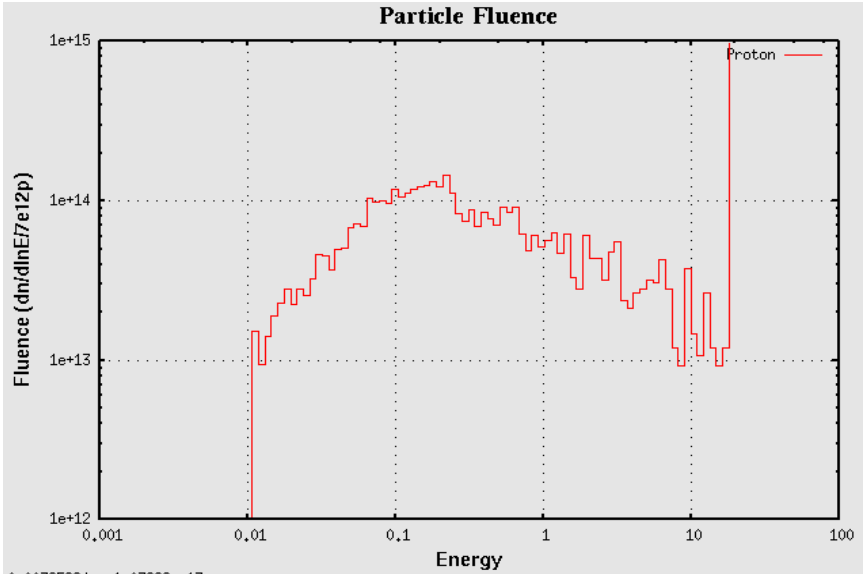
Single Differential Quantities Plot

The scoring cards USRBDX, USRCOLL, USRTRACK, USRYIELD after the data merging are producing a single differential quantities that can be plotted with the "USR-1D" plot frame in flair. This frame is using the _tab.lis file and many data can be super imposed one on top of the other.

- Select the "fluence" plot from the Plot List or from the Project Tree.
- Fill in the values as you see in the following image.
  - Opt: under the title to supply additional gnuplot options to the title like changing the default font
  - Axes Labels: supply axes labels and additional options if needed
  - Axes Range: click on the "log X" and "log Y"
- To add now the detectors click on the + inside the Detectors frame. The File Dialog will appear with all the Fluka _tab.lis files. Select the "tutorial_usrcoll_51_tab.lis" file. Now the Detectors listbox will contain an entry named "#Detector 1" and the "Detector Info" frame will be filled with some basic values.
- Select from the "Det:" drop-down box the "Proton" detector
- Rename the detector to "Proton" with no (#) symbol in front. The (#) symbol is used as a comment not to display the label of the detector in the plot.
- Select from X: the "Low [xl]" value to plot and on the Y: leave the default "Yx<Xgeo>" which will plot the isolethargic fluence.
- Finally for the proton detector, set the normalization to 7e12, select as style with: "steps"

- You can already see the plot by clicking the Plot button (or Ctrl-Enter). It will show a red histogram of the proton fluence.

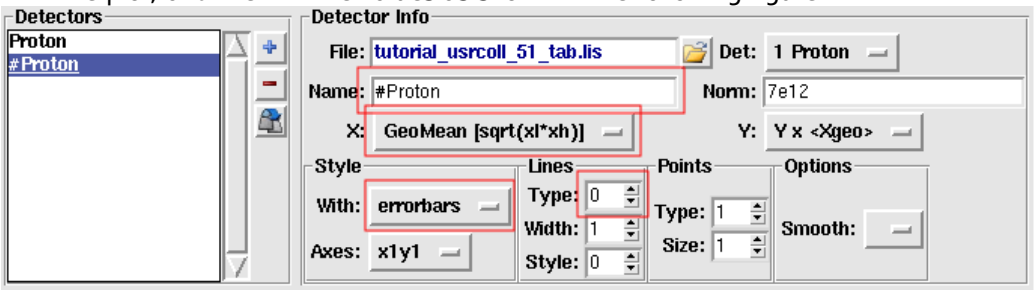




To add the statistical error bars and the neutron fluence you will have to create a couple of other detectors.

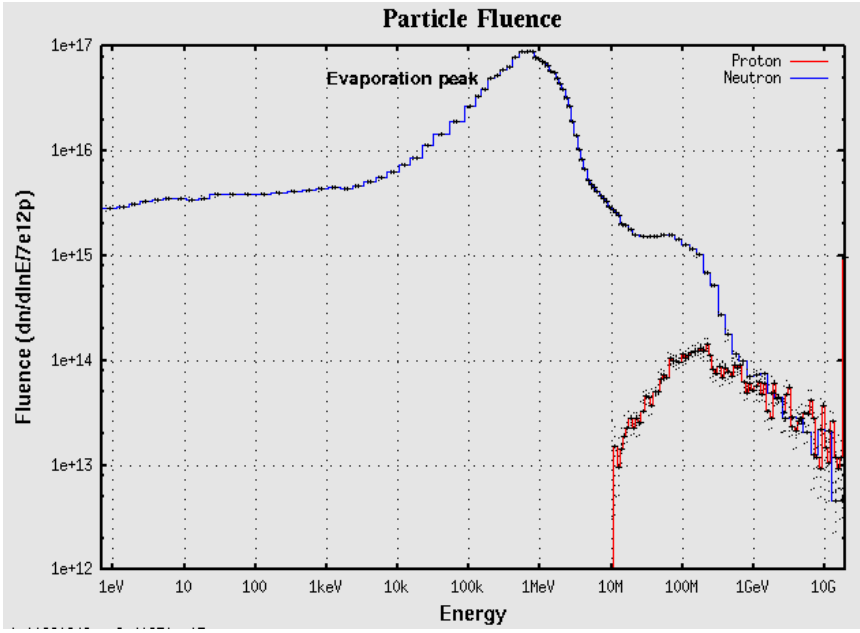
- Clone the "Proton" detector by selecting in the Detector listbox the "Proton" detector and clicking the  icon or the Ctrl-D button.

- Insert the # character in the name (#Proton), to avoid having a duplicate label on the plot, and then fill the values as shown in the following figure



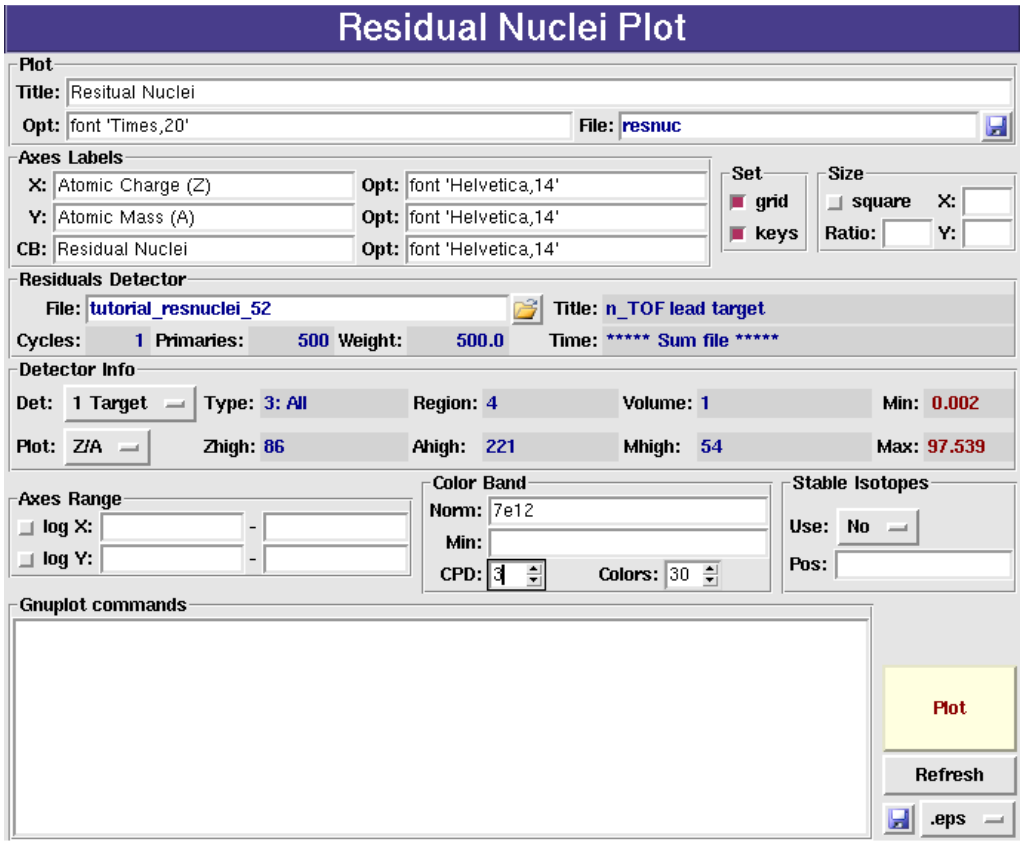
- Click on the ✚ inside the Detectors frame to add an extra detector. Select again the same file "tutorial_usrcoll_51_tab.lis" and from the "Det:" select the number "2 Neutron". Or even you can clone the first detector "Proton" and change the corresponding fields.
- Change the name of the detector to "Neutron" use on X: "Low [xl]", Style with: "steps" Type: 3.
- Clone the "Neutron" detector as before with the Protons to add the errorbars. Name the new detector as "#Neutron" X: "GeoMean [sqrt(xl*xh)]", and style with: "errorbars", Type: 0.
- Clicking on the Plot button you can see the plot.
- Optionally to make it even nicer you can add a custom x-scale and a label, in the Gnuplot commands:

```
set xtics ('1meV' 1e-12, '10m' 1e-11, '100m' 1e-10, '1eV' 1e-9, '10' 1e-8,'100' 1e-7,'1keV' 1e-6, '10k' 1e-5, '100k' 1e-4, '1MeV' 1e-3, '10M' 0.01, '100M' 0.1, '1GeV' 1, '10G' 10, '100G' 100, '1TeV' 1000, '10T' 1e4, '100T' 1e5)
set label 'Evaporation peak' at 1e-6,5e16 font 'Arial,14'
```

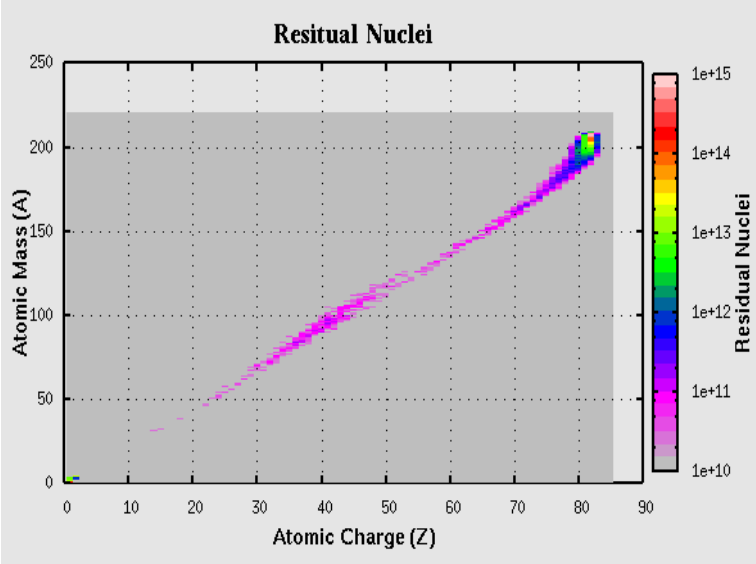


Residual Nuclei Plot

As a last example we will make a plot of the residual nuclei scored with the RESNUCLEi card. The Residual Nuclei Plot frame is able to do two dimensional plots of Z/A, Z/N or histograms of Z or A. Click on the "resnuc" plot from the Plot List or from the Project Tree, and fill the fields with the information as it is shown on the following figure



We are requesting a 2-dimensional Z/A plot of the first detector using a normalization factor of 7E12 primary particles. Clicking the Plot will generate the following plot:



A.Index