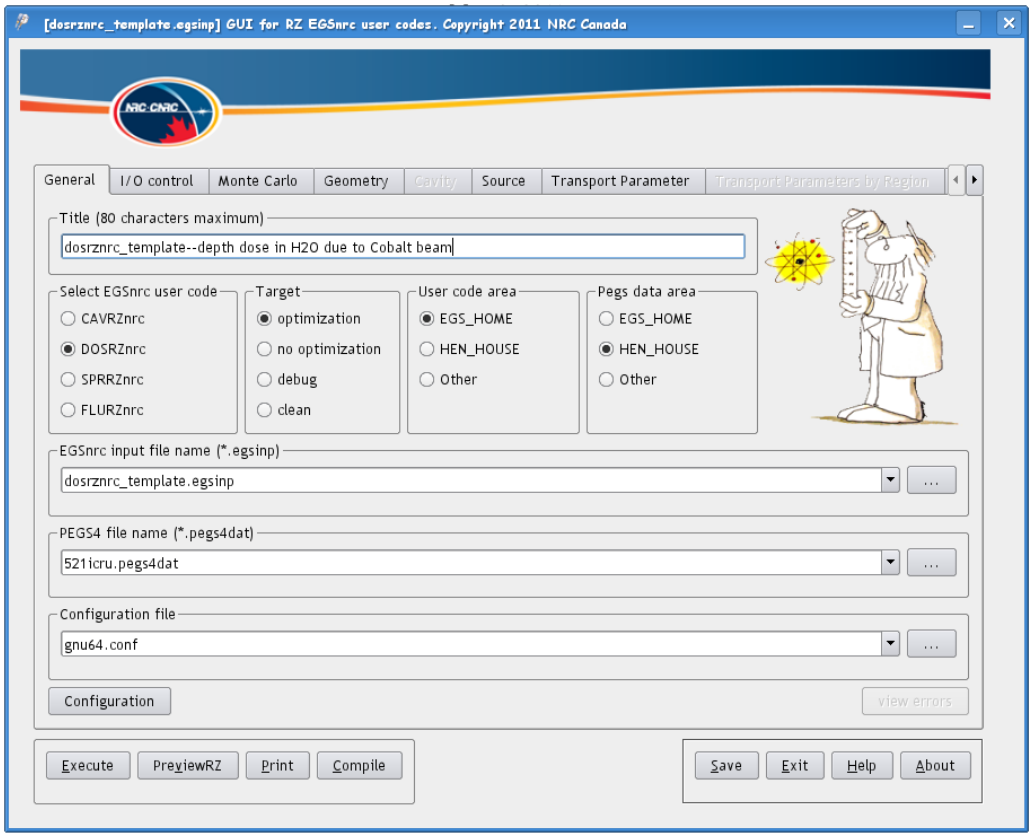
User Manual for egs_inprz, a GUI for the NRC RZ user-codes

Ernesto Mainegra-Hing
Ionizing Radiation Standards
National Research Council of Canada
Ottawa, K1A OR6

November 26, 2015

NRCC Report PIRS-801(RevB)



Front page of the egs_inprz GUI for the RZ EGSnrc User Codes.

Abstract

This is the reference user manual for `egs_inprz`, a graphical user interface for the EGSnrc RZ user-codes suite. It briefly introduces the GUI and describes how to install it and work with it. Descriptions and snapshots of each of the input blocks are provided.

Contents

1 Introduction

One of the major improvements in the RZ user-codes was moving from an input format based on a long series of numbers to a text based input which is easier to use. This text based system for input files was then used to create a single routine (`get_inputs`) to read inputs entries for all the user codes so that now one can cut and paste entire input blocks from one user code to another. This routine is now part of the EGSnrc system and can be used in any user-code to parse through `key=value` pairs in an input file. As a consecuence, input files look very similar and, more importantly, they are much easier to read and know exactly what the simulation is about without having the description of the inputs open on the desk. The idea behind using a GUI for working with EGSnrc input files is to further extend the above mentioned improvements. Although the input files are currently very readable, one must still remember what the keys used in an input file mean. By using this GUI, a user can immediately get a description about any input parameter by means of tool-tips.

`egs_inprz` is a Graphical User Interface (GUI), originally created for manipulating (reading, creating, modifying, printing and visualizing) input files for the RZ suite of EGSnrc user-codes: DOSRZnrc, CAVRZnrc, SPRRZnrc and FLURZnrc (see NRCC report PIRS-702[1]). Furthermore it can be also used for compiling and executing these user-codes. `egs_inprz` is user friendly, offering more flexibility, on-line help and therefore, increases the efficiency in getting hands-on experience with the EGSnrc user codes.

This GUI was developed using Qt, a multi-platform, C++ Graphical User Interfaces toolkit that enables building efficient, portable and maintainable GUI applications quickly and easily. Qt is a fully object-oriented, easily extensible C++ application framework that enables rapid building of state-of-the-art GUI applications. For more information please see http://www.qt.io/.

2 Installation

This GUI is part of the multi-platform version[2] of the EGSnrc Monte Carlo simulation system[3]. For its development we used the Qt library and therefore, users wanting to build it, will have to install this library. Most Linux distributions include the Qt library these days, since the popular Desktop Environment KDE is based on this library. However, if the Qt library is not available in the user's system, one needs to install it first

Only Linux/Unix users need to built this GUI since it is distributed as a binary executable on Windows. We might start distributing binary executables for Linux/Unix as well in the near future. The only requirement for this to happen is that most Linux distributions and Unixes are binary compatible and the GUI is linked statically to the Qt library.

2.1 Building `egs_inprz`

The user can have this and the other GUI's built during installation of the EGSnrc system, provided `QTDIR` is properly set. At any time the user can go to `$HEN_HOUSE/gui/egs_inprz` and type

```
./make [EGS_CONFIG=desired_config]
```

A C++ compiler will have to be installed on your computer in order to build the GUIs. On Windows one **must** have installed either MS C++ 6.0 or Borland C++.

Note: You only need to pass `EGS_CONFIG` to `make` if it is not set or you want/need to build the GUI for a different configuration as the current one. In principle, all Makefiles provided in the new EGSnrcMP environment are for `GNU make`. Although they might also work with other Unix `make` versions.

It is important that the environment variable `QTDIR` points to the location where Qt was installed. This can be checked by issuing the command:

`echo $QTDIR` on Unix/Linux or

`echo %QTDIR%` on a Windows console.

One can change this environment variable by issuing the command

`setenv QTDIR Qt_location` for the C shell, or

`export QTDIR = Qt_location` for Bash, or

`set QTDIR=Qt_location` on a Windows console.

On Unix/Linux this variable can be set on a system wide basis by including the corresponding statement above in the `.cshrc` resource file for the C-shell or the `.basrc` resource file for Bash.

On Windows the user can also set the QTDIR environment variable system wide by right clicking on the `My Computer` icon, selecting `Properties` and clicking on the `Environment Variables` button in the `Advanced` tab.

3 Using egs_inprz

3.1 Running egs_inprz

After installing EGSnrc, egs_inprz is located on HEN_HOUSE/bin/my_machine/. my_machine stands for the name of the configuration used to build the GUI. For more information about configurations the user is referred to the PIRS-877 report on the new multi-platform environment[2].

On Windows the user can invoke directly the binary executable from a DOS console, since its location will be on the user's PATH environment variable. If requested by the user, there will be also shortcuts to the GUI's distributed with the EGSnrc system on the Desktop and Start Menu.

On Unix/Linux the user can also invoke directly the binary executable from a shell console, since its location is added to the user's PATH environment variable when the corresponding egsnrc_[cshrc|bashrc]_additions is sourced, which **must** have been done after installing the EGSnrc system. The alias egsinprz is also available, which points to HEN_HOUSE/bin/my_machine/egs_inprz and starts the GUI in the background. If requested by the user, shortcuts for the KDE desktop environment are also created by the installation GUI.

Once all the necessary information is entered, the user can perform different operations from within the GUI provided the input file has been saved to the disk since all other operations use the disk version of the input file.

3.2 Reading EGSnrc RZ input files

Existing input files can be read directly from the command line by passing the file name as argument, i.e., by invoking

egs_inprz *filename[.egsinp]*

where the file name can be with or without extension. If the file does not exist, a warning message is shown and the file name new_file.egsinp is used instead. Note, that in this case no input file will exist. To have an actual input file and be able to run a calculation, the user **must** have saved it. Saving new_file.egsinp without modifying any entry will leave a default input file for use with the RZ user code dosrznrc.mortran.

Once an existing input file is loaded, it is searched to identify the user-code it belongs to. If no user-code is identified, DOSRZnrc is used by default. Once the user-code to be used is known, its location becomes the place where the GUI will look for input files.

Regarding location, the EGSnrc system relies on having the input file on the EGSnrc user area, *i.e.*, EGS_HOME/user-code. This is so because for execution, temporary directories and output files are created, moved and deleted and all these operations are relative to the EGS_HOME/user-code location.

For this reason, this GUI will only store input files in the user's EGSnrc area, i.e., `EGS_HOME/user-code`. If the EGSnrc user area does not exist, the GUI creates it and issues a warning.

Input files can be also read in from the GUI's General tab. Once the GUI is loaded, a list of available **.egsinp* input files in the current directory is offered to the user through the EGSnrc input file name combo box. By default the input file template *dosrznrc_template.egsinp*, distributed with the EGSnrc system, is loaded. Alternatively, the user can click on the button to the right of the combo box to invoke an open file dialog to open any **.egsinp* file located anywhere.

The GUI verifies that all media used in the input file are available in the selected PEGS4 data set. By default this file is set to be `521icru.pegs4dat`, a standard data file, that comes with the EGSnrc distribution. If any medium is not found in the current PEGS4 data file, an error message pops up recommending that the user corrects the media names and/or find the appropriate data file.

The user-code area is the location where `egs_inprz` will look for input files. Initially, `egs_inprz` assumes that the user-code area is `EGS_HOME/user-code`, where `user-code` is by default *dosrznrc*. If the GUI is started from any user-code location, `user-code` is changed to the corresponding user-code. If a valid input file name is passed as argument to `egs_inprz`, then after identifying the user-code, `user-code` is updated properly. The user-code area can be later changed by the user in the general input tab (see figure 5 in section 5.1).

Similarly, the PEGS4 data area is the location where `egs_inprz` will look for PEGS4 data sets. Since there are some data sets in the EGSnrc distribution, we chose to set this area to be in `HEN_HOUSE/pegs4/data` by default. Later on, when users have created their own data sets, they can switch to `EGS_HOME/pegs4/data` or any other location of their preference.

3.3 Creating EGSnrc RZ input files

As mentioned above, when starting the RZ GUI, the template `dosrznrc_template.egsinp` is read in, which contains defaults for all possible entries. Saving this template under any other name is a possible way for getting started. In similar fashion, one can switch to another RZ user-code (see section 5.1) and select the corresponding input template file.

3.4 Porting input files between NRC RZ user-codes

Sometimes different user-codes share common input blocks like transport parameters, geometry, variance reduction parameters, and so on. For instance, the user might want to run a CAVRZnrc calculation to obtain the dose inside the air cavity of an ion chamber and also run a FLURZnrc calculation to obtain the spectrum inside the cavity for the same chamber. This is easily acomplished with `egs_inprz` by loading the input file for the CAVRZnrc calculation, switching to the other user code input by clicking on the corresponding radio button in the user-code group box (see figure 5 in section 5.1). One will have to modify

the default entries for the selected user-code to suit the user's problem if needed. Once the proper entries are made, the input file can be saved by clicking on the *Save* or *Save&Exit* button in the user's EGSnrc user-code area (if it doesn't exist, it is created automatically, and a warning is issued to the user).

3.5 Viewing the geometry with previewRZ

Once an existing input file has been loaded or created from scratch and saved on the hard drive, the user can invoke `previewRZ`, a tool supplied with the EGSnrc distribution, which allows one to visualize the geometry and material data (see figure 1). The `PreviewRZ` button, placed in the left bottom corner of the GUI (see any GUI snapshot in section 5), becomes enabled if Tcl/Tk is installed on your computer, the input file exists and there were no errors reading the geometry. Pressing this button is equivalent to typing on the command line of a console (Windows or Unix/Linux)

```
HEN_HOUSE/previewRZ/previewRZ name[.egsinp]
```

where the input filename can be entered with or without extension.

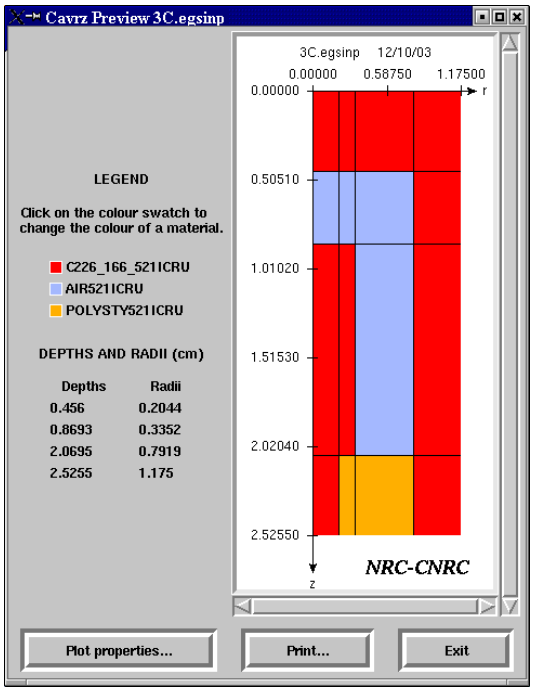


Figure 1: View of a 3C cylindrical ionization chamber using `previewRZ`.

`previewRZ` is a Tcl/Tk script which had been previously used at NRC only on Unix/Linux.

We have now successfully used `previewRZ` on Windows 2000/XP after downloading and installing a Tcl/Tk self-extracting distribution. To find out whether Tcl/Tk is available on the user's system, `egs_inprz` tries to find the binary executable `wish.exe` on Windows or `wish` on Unix/Linux in any of the locations defined on the user's PATH environment variable. The Tcl/Tk package is **FREELY** distributed for HP-UX, Linux, Solaris, and Windows by ActiveState Corp. To obtain Tcl/Tk go to http://www.activestate.com/Products/ActiveTcl/ and click on the `Download` link of the page. For more information and useful links on Tcl/Tk please visit http://www.tcl.tk/software/tcltk/

Future versions of the `egs_inprz` GUI will use its own previewing tool, but for now, users wishing to have the feature of looking at the geometry they are defining, will have to install the Tcl/Tk package.

3.6 Printing *.egsinp input files

To produce a hard copy of the input file, users have the option to print the file by pressing the `Print` button located in the button group on the lower left corner of the GUI (see any GUI snapshot in section 5). A Print Dialog pops up with a list of available printers and a printer and paper format setup among other options (see figure 2).

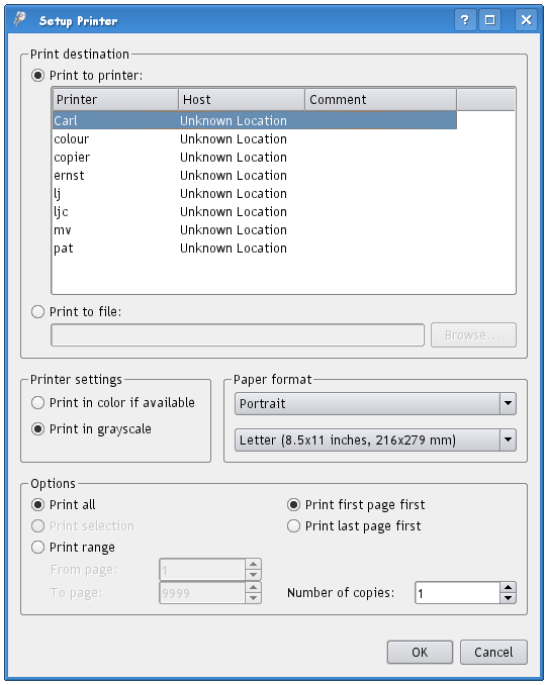


Figure 2: Printer Setup Dialog on SuSE Linux 10.3 KDE 3.5.9

3.7 Compiling the RZ user codes

When one modifies the user-codes, these need to be re-compiled. The user can perform this operation from within the GUI by pressing the *Compile* button on the lower left corner of the GUI (see for instance figure 5 in section 5.1). On the `General Information` tab there is a `Target` radio button group box where one can choose the type of compilation desired. By default it is set to `optimization` which uses the optimization option defined in the active config file generated during the EGSnrc installation process or the configuration utility available in all the EGSnrcMP GUI's. The other available options are `no optimization, debug and clean`. Optimization is recommended for production runs after the user-code and the input file have been thoroughly tested.

3.8 Executing the RZ user codes

After all necessary information has been entered and stored, one can execute the EGSnrc RZ user-code from within the GUI by pressing the *Execute* button on the lower left corner of the GUI (see figure 5 in section 5.1). A dialog appears where one can define the different execution parameters (see figure 3). There are two modes for running an EGSnrc RZ user-code, *interactive* or *batch*, *i.e.*, using a batch queuing system. The execution mode defaults to *interactive*. The *batch* execution mode is only available on Unix/Linux since it has not been implemented on Windows yet. At NRC the *PBS batch system* is currently used to send jobs to a queue where they are remotely executed, returning the results after completion to the user EGSnrc area.

On Unix/Linux if the batch execution mode is selected, a pane becomes enabled where queue input parameters can be entered such as the queueing system, type of queue and number of jobs to submit (see figure 4). The GUI recognizes which queueing systems are available by looking up on `$HEN_HOUSE/scripts` for batch definition files in the form `batch_options.queueing_system`, where `queueing_system` stands for either `at`, `nqs` or `pbs`. The user can add any other batch submission system by creating a batch definition file in a similar fashion to the ones in the EGSnrcMP distribution.

The default batch submission system assumed in the GUI is the standard Unix job submission tool `at`. The batch definition files provided in the directory `$HEN_HOUSE/scripts` contain specific definitions for the `at`, `NQS` and `PBS` batch submission systems. If the user wants to make NQS, PBS or any other system the default job submission system, he/she can define the environment variable `EGS_BATCH_SYSTEM` to be nqs, pbs or the name of the other queueing system.

These are the batch definition files distributed with the EGSnrc system:

```
batch_options.at

batch_options.nqs

batch_options.pbs
```

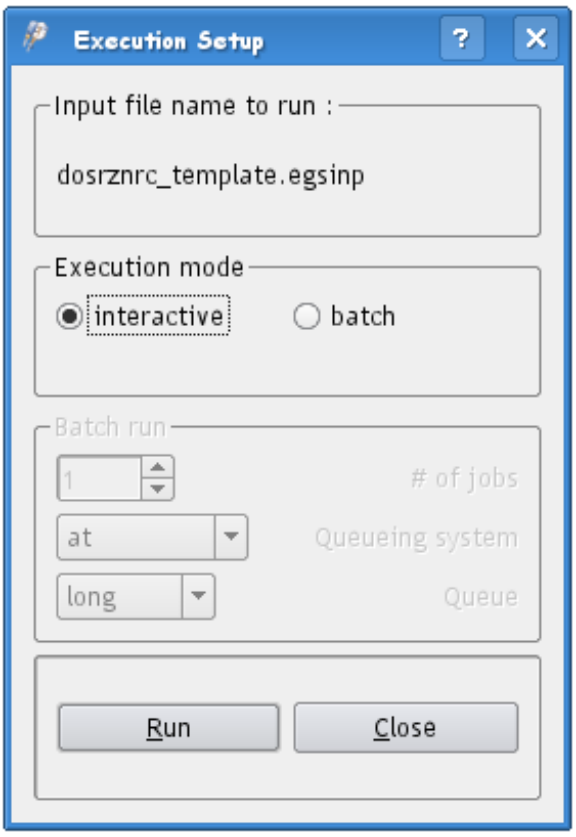


Figure 3: Execution Setup Dialog.

Queue names are installation especific and at NRC the names *short*, *medium* and *long* have been adopted for `PBS` and `NQS`. To change these, edit the names in the proper batch definition file.

For more information on the implementation of parallel runs in the new EGSnrc system, the reader is refered to the NRCC report PIRS-877.

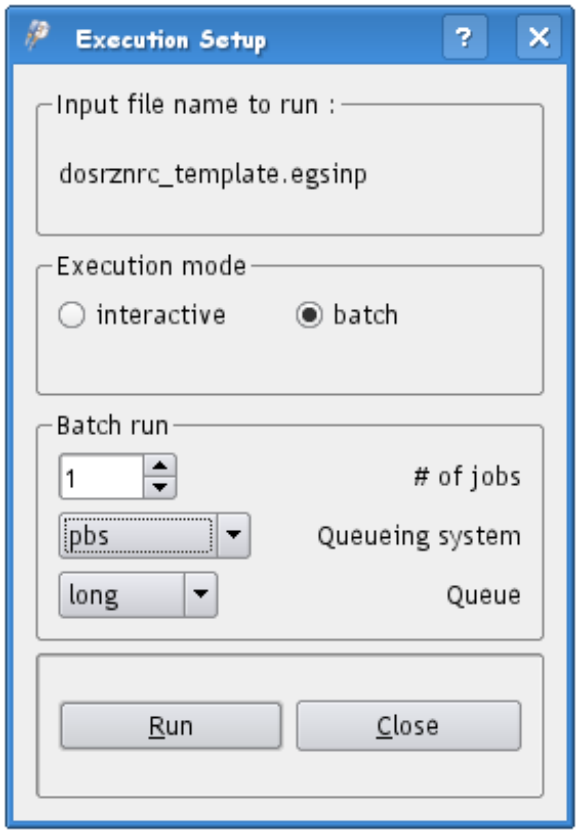


Figure 4: Execution Setup Dialog in batch mode.

4 Getting help

One of the advantages of a graphical user interface is the possibility of providing information in an interactive way. `egs_inprz` uses this feature extensively by activating so called *tool tips* when the user positions the mouse over a given area in the GUI. A dialog pops up *temporarily* with information, if available, about the corresponding input quantity.

There is also the possibility of activating these *tool tips permanently* (until another action is performed: mouse click or key press). For this, the user must set the focus on the relevant location and press `Shift+F1`. The help text appears immediately; it goes away as soon as the user does something else.

More general information is provided in html format through the *Help* button located in the lower right corner of the GUI (*html* version of this document). `egs_inprz` attemps to run *Internet Explorer* on Windows and *Konqueror* or *Netscape* on Unix/Linux to show the document. If none of these are available an error message is displayed. In that case the user can go to `$HEN_HOUSE/gui/egs_inprz/html` and load the index page *index.html* with an html browser of his/her choice.

5 Input blocks description and screenshots

In this section we describe briefly the different input blocks that are used in the NRC RZ user codes. We have also included screenshots of the different input tabs of the GUI. In each of these tabs, there are input options, common to all the RZ user codes. But some of them are specific to one user code and remain disabled when one selects a different user code. The active file name is always displayed on the GUI's caption. This can be useful to recognize whether the current file name in the input box is the same as the active one.

Note that the bottom row of buttons are available with all tabs.

5.1 The General Information Tab

As its name suggests, this section of the tabbed dialog is intended to collect general information not contemplated inside the input file itself like the input and pegs4 data file names, the areas to search for those files, compilation mode, execution mode and its parameters, the user code name, etc. The title constitutes an exception, since it is part of the input file, but does not fit in any of the different input blocks.

A very useful feature in this GUI is the ability to set the location of the input and pegs4 files automatically to be in the `HEN_HOUSE` or `EGS_HOME` area. This saves time by not having to browse all the way to the location of different files, acting like a shortcut. If the files are loaded from a location different than the above mentioned, the `Other` radio button is checked. Note that upon loading an input file from a location other than the `HEN_HOUSE` or `EGS_HOME` user code area, it will be only saved on the `EGS_HOME` user code area.

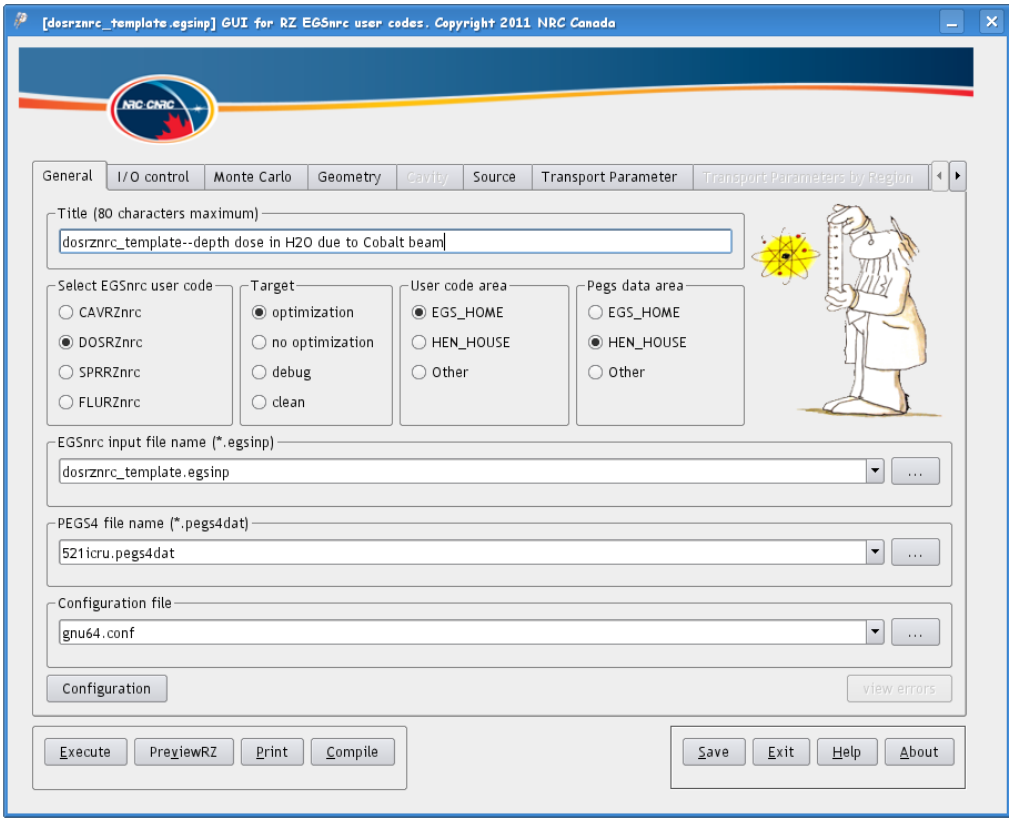


Figure 5: General Input for the RZ EGSnrc User Codes.

5.2 The I/O Control Tab

This block contains information relevant to the I/O controls of the NRC user codes. Many of the inputs are common to all codes, but there are some which are specific to only some of them.

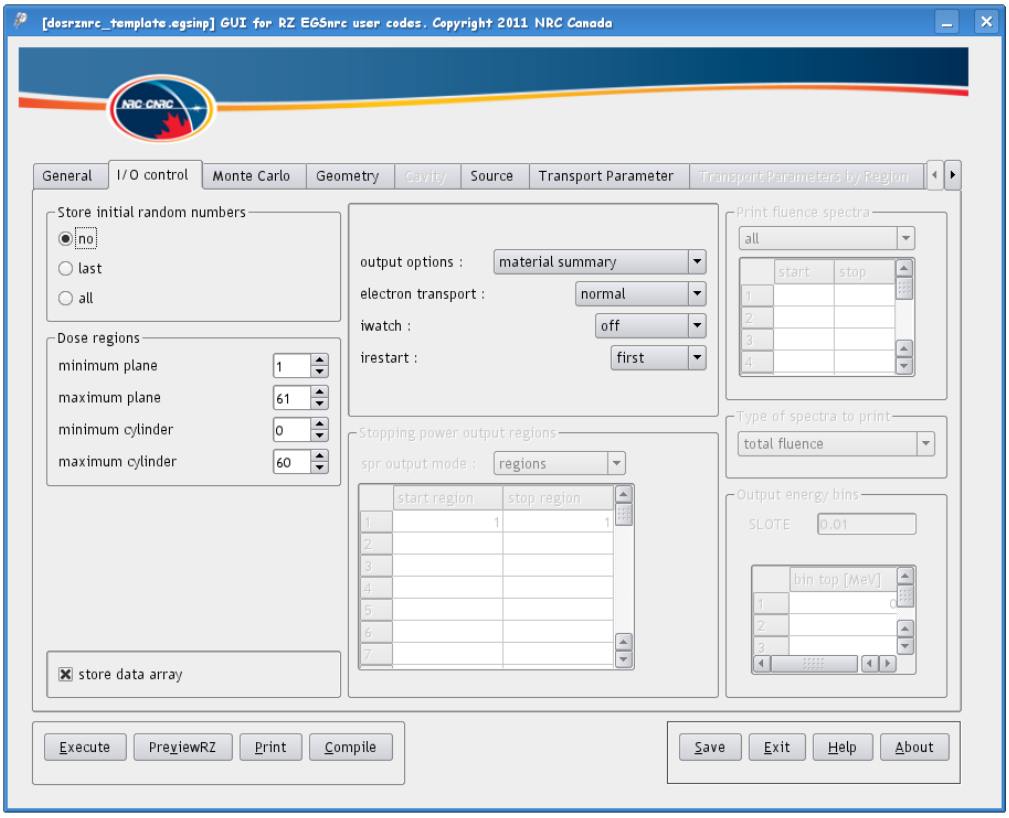


Figure 6: I/O control for the RZ EGSnrc User Codes.

5.3 The Monte Carlo Parameter Tab

This input tab collects the typical information required in a Monte Carlo simulation like the number of histories to run, the initial random number seeds, desired statistical accuracy, and the maximum CPU time for the calculation. There are also more user code specific entries that are enabled or disabled depending on the user code selected.

An input block required only by the user code DOSRZnrc when the calculation type is *pulse height distribution* is also included in this tab. For other calculation types and user codes, this box remains disabled.

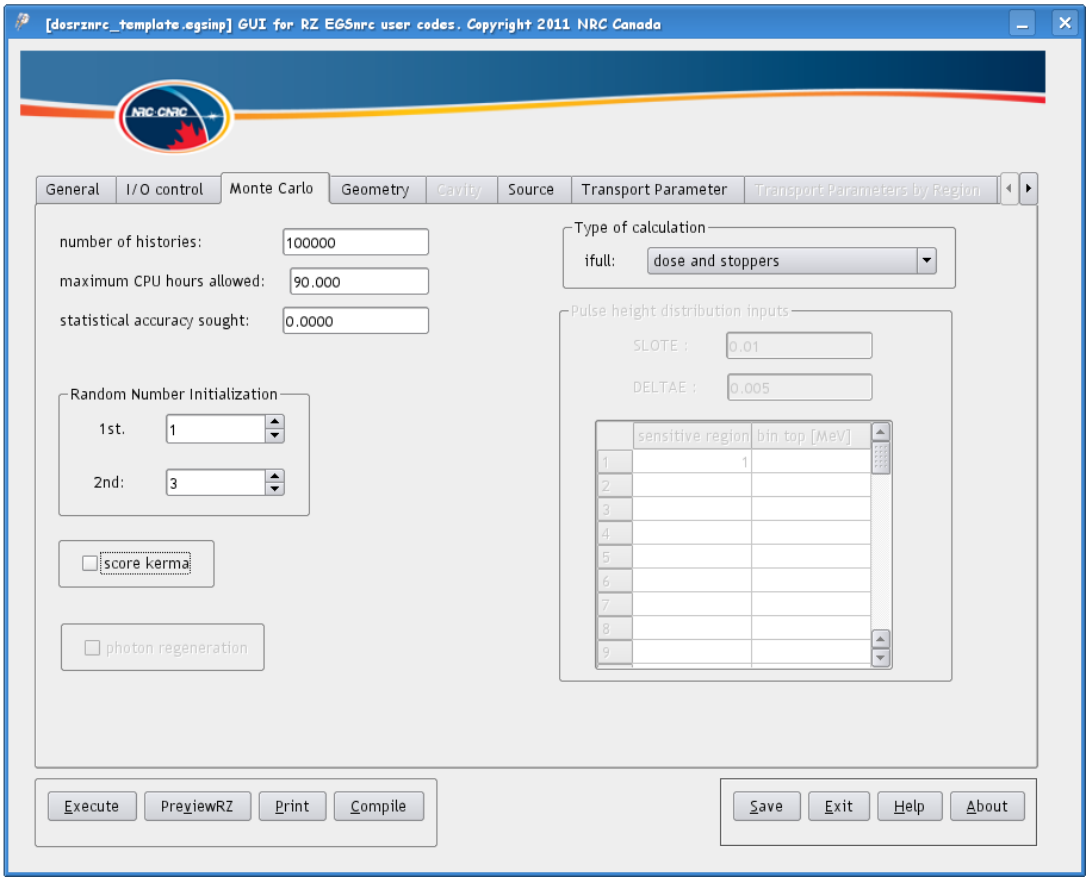


Figure 7: Monte Carlo parameters for the RZ EGSnrc User Codes.

5.4 The Geometry Tab

This input block contains all the necessary inputs for defining a RZ geometry (cylindrical symmetry) and the media present in the different regions. It is important to notice that for the user code CAVRZnrc an option is available to define the geometry in a simpler way. If the input method selected (upper left corner of the tab) is *cavity description*, then the rest of the input fields in this tab are disabled and the whole geometry input occurs through the next tab, `the cavity tab`.

Only media present in the current PEGS4 data set can be set in the media table. This is assured by activating a combo box in the first column of the media table as soon as the user tries to type or double click on it.

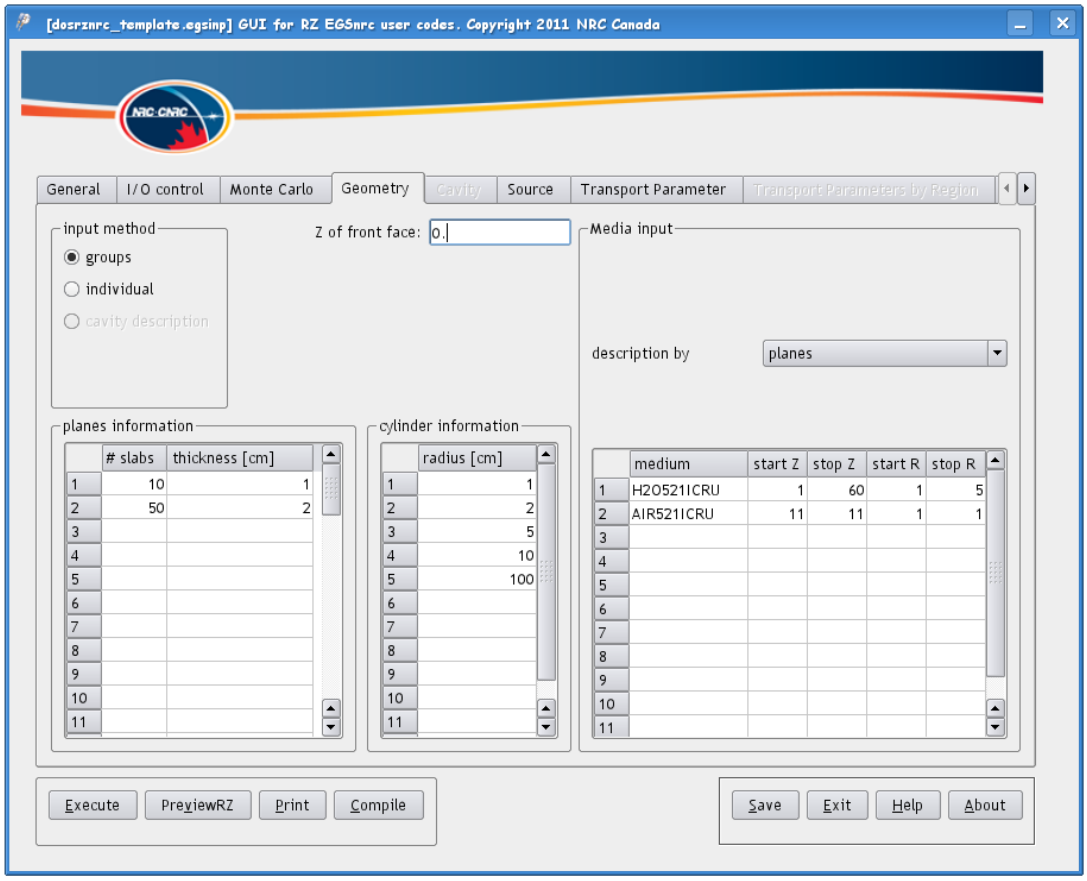


Figure 8: Geometry Input for the RZ EGSnrc User Codes.

5.5 The Cavity Tab

This tab is only enabled for the user code CAVRZnrc. If the input method selected in the `geometry tab` (upper left corner) is *groups* or *individual*, the user can define the regions comprising the cavity there. If on the other hand, the input method selected is *cavity description*, then the rest of the input fields in the `the geometry tab` are disabled and the the whole geometry input occurs here. The materials for the chamber wall and the electrode can be selected from available media in the current PEGS4 data file. This option was useful for early calculations but is not adequate for chambers in which one wants to include much detail.

Beware: If the input method is cavity description, the material name inside the cavity is assumed to be **AIR** by the user-code CAVRZnrc, *i.e.*, CAVRZnrc will search for this medium in the pegs4 data file.

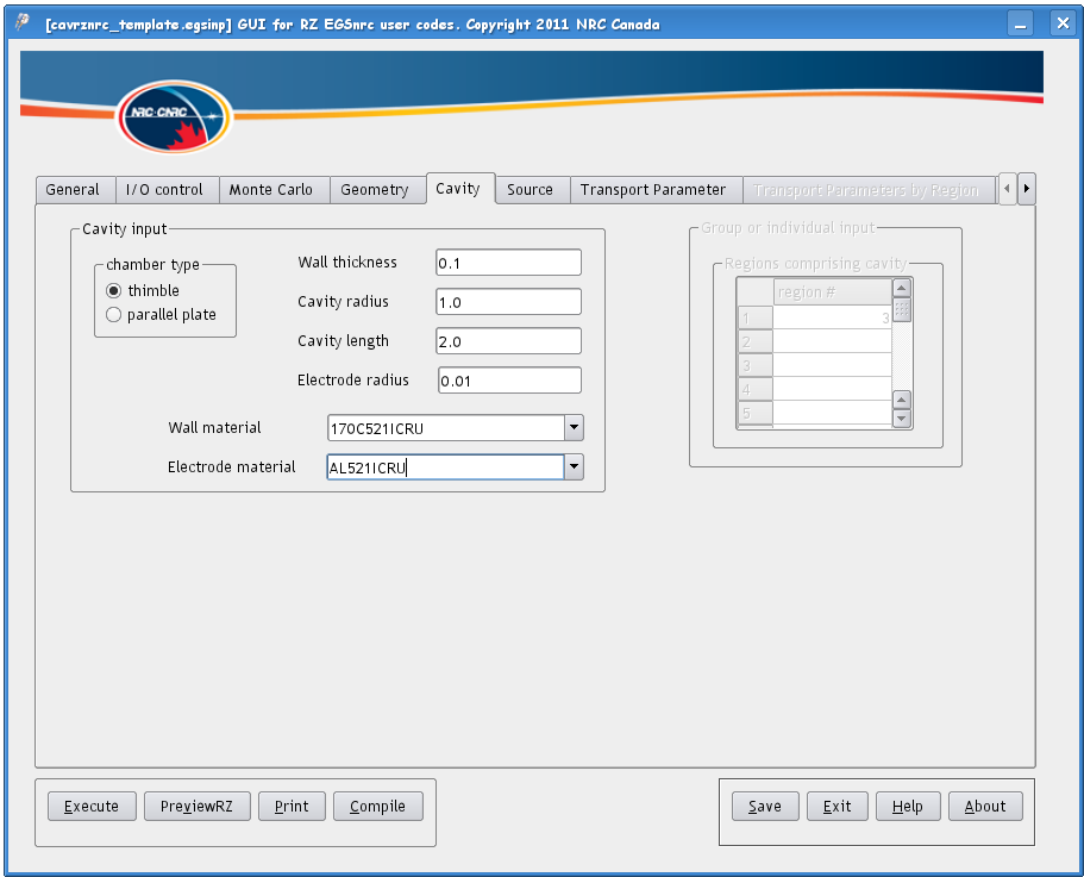


Figure 9: Cavity Input for the RZ EGSnrc User Code CAVRZnrc.

5.6 The Source Tab

Any input related to the initial characteristics of the beam or phase space file are entered here. There are 15 different types of source geometries that can be entered. A detailed description of each source can be found in the NRC User Codes Manual (NRCC report PIRS-702[1]) and directly through the `Tool Tips` help feature offered by this GUI.

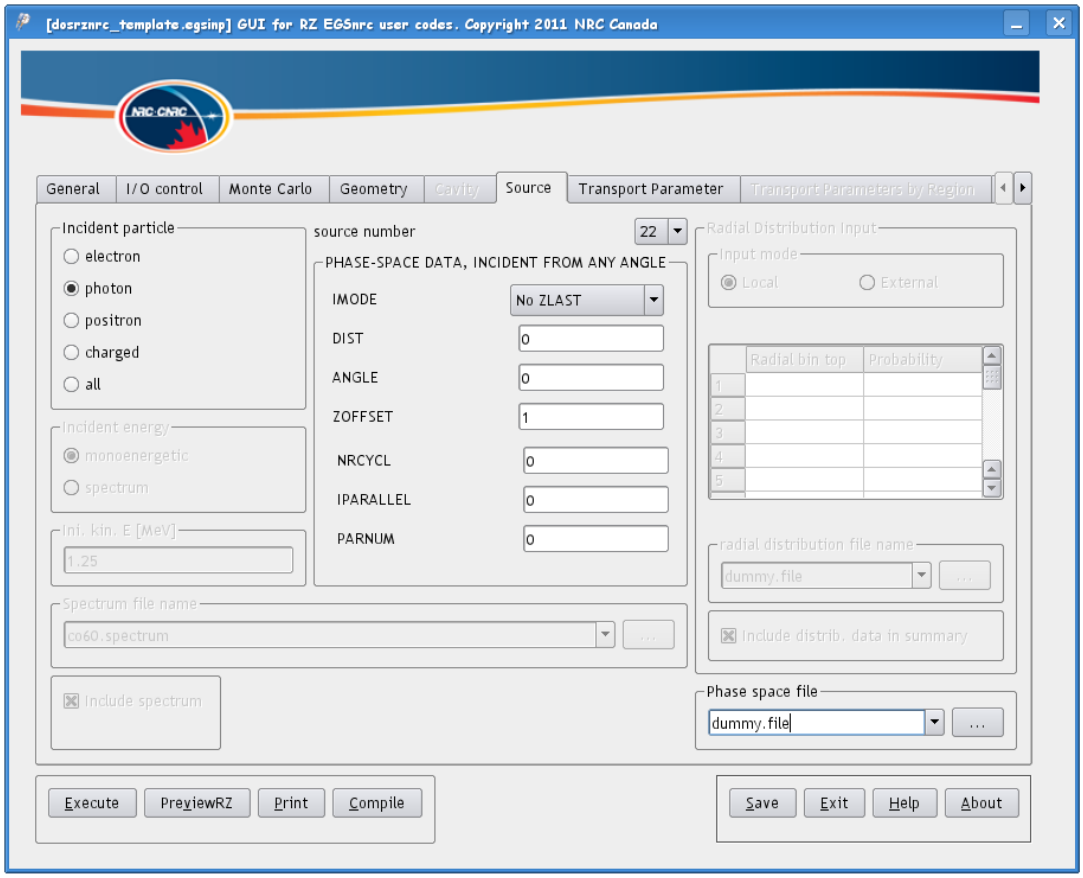


Figure 10: Source Input for the RZ EGSnrc User Codes.

If the user selects source 21 or 22, the Phase-space file edit line becomes enabled and one can either type the name of a phase-space file or one can use the Open File Dialog to navigate throught the directories to get the desired file. In the latter case, the path is stripped from the file name, but it is still rembered and properly added to the file name when saving the input file. Although EGSnrc accepts phase-space files with arbitrary extensions, it is customary to use the *.egsphsp1 extension for regular phase-space files and *.IAEAphsp for phase-space files using the IAEA format (in this latter case, the extension is improtant). Although the default filter for searching for files uses these extensions (see figure 11), the All

files (*) filter is still available.

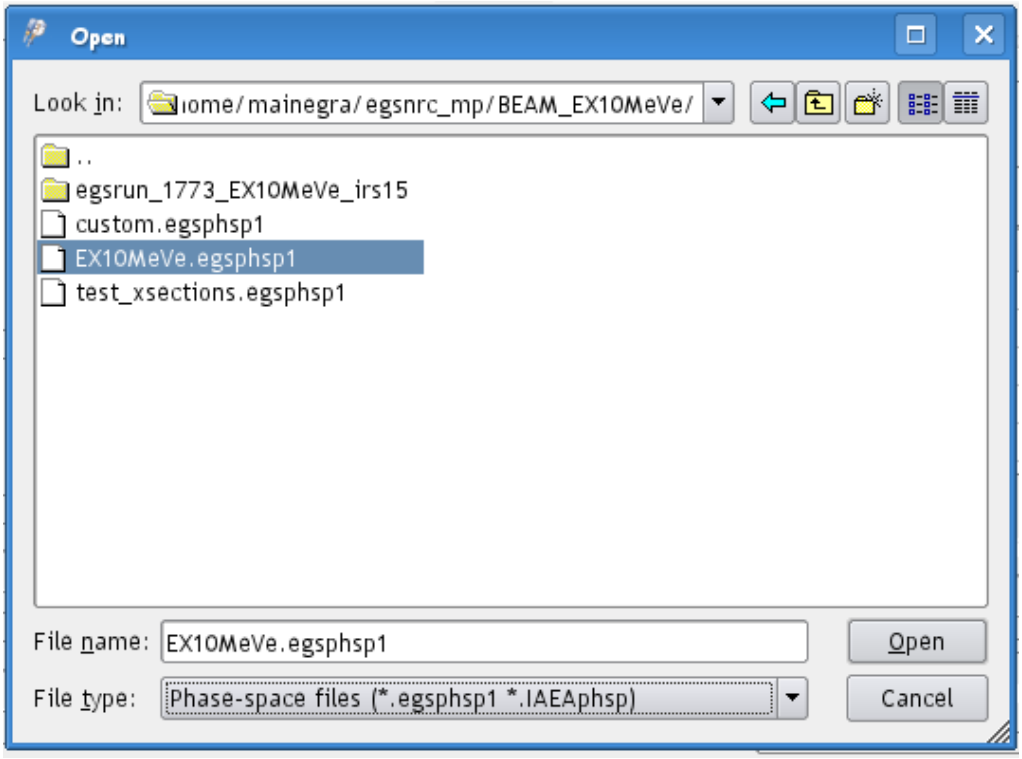


Figure 11: Phase-space open file dialog.

5.6.1 Setting up a BEAM Source

Alternatively to phase-space files, EGSnrc can now use a BEAMnrc simulation as a particle source (figure 12). This source (source 23) needs to be set up in a separate dialog. When the user selects this source, a button appears with a red text, prompting the user to enter the source parameters and warning that unless there are BEAM user-codes compiled as a library on the system, no BEAM user-code will be available.

Clicking on the above mentioned button brings a new dialog (figure 13), where the user can enter the name of the BEAM user-code, the BEAM input file and the required PEGS4 data file based on the current PEGS4 directory. The user can also define a weight window for the particles and the positioning and orientation of the source.

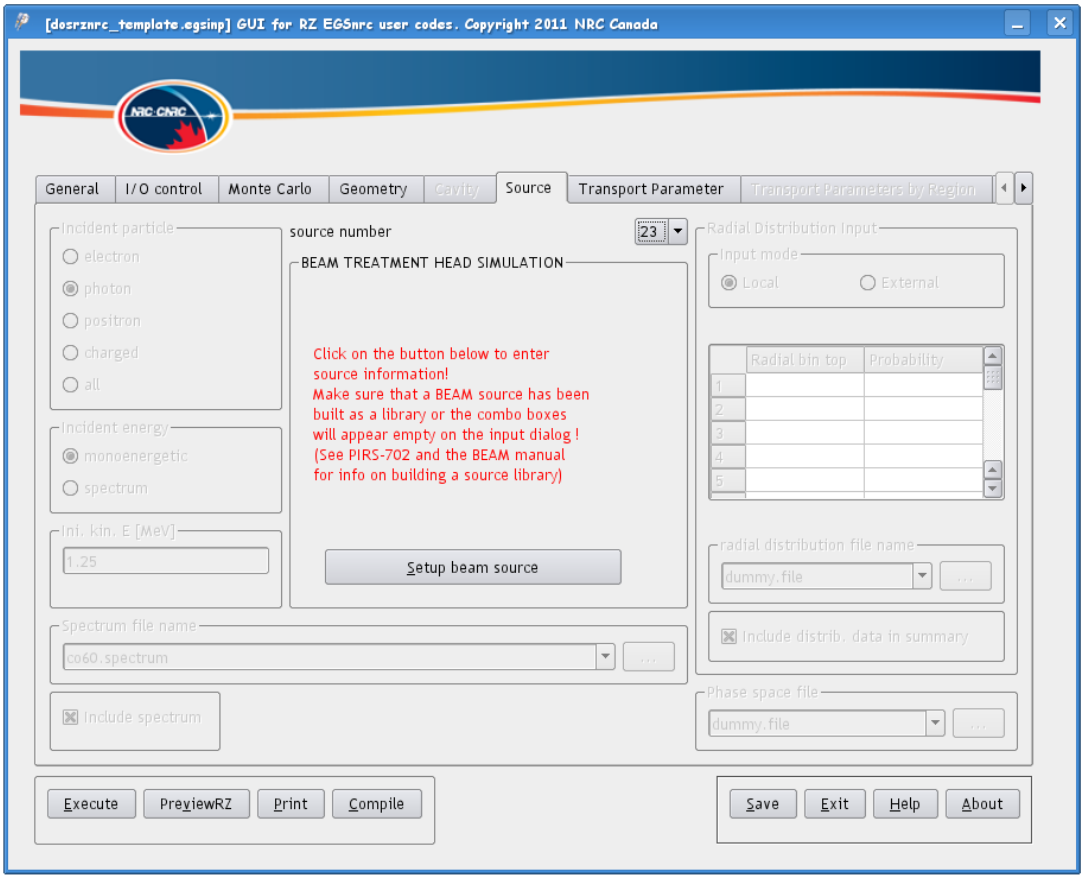


Figure 12: Selecting BEAMnrc as a source.

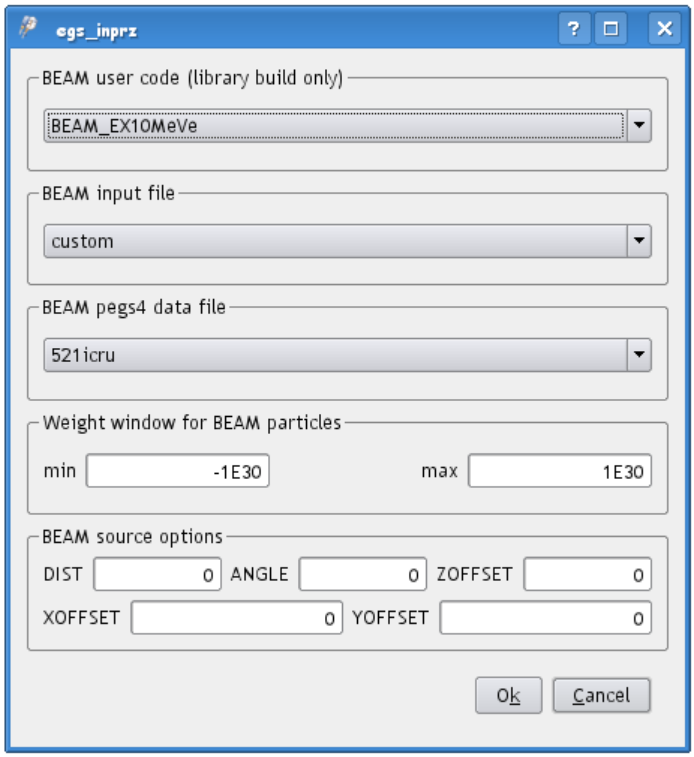


Figure 13: BEAMnrc source definition dialog.

5.7 The Transport Parameters Tab

This input section gathers information inherent to the physics of the transport of electromagnetic radiation through matter. Threshold energies for photons and electrons, electron transport algorithm to be used, as well as the cross section data and angular distributions to be used are entries that are defined here. By default, EGSnrc uses threshold energies given by AP and AE in each region for photons and electrons respectively. The electron transport is originally set to the EGSnrc default algorithm, which is independent of electron step size. The user can also choose to turn *on* and *off* other effects in the simulation, like Compton binding effects, spin effects, Rayleigh scattering, atomic relaxations and angular sampling of the photo-electrons. For a detailed reading on the physics of the transport of photons and electrons the user is refered to the EGSnrc system manual (NRCC Report PIRS-701[3]).

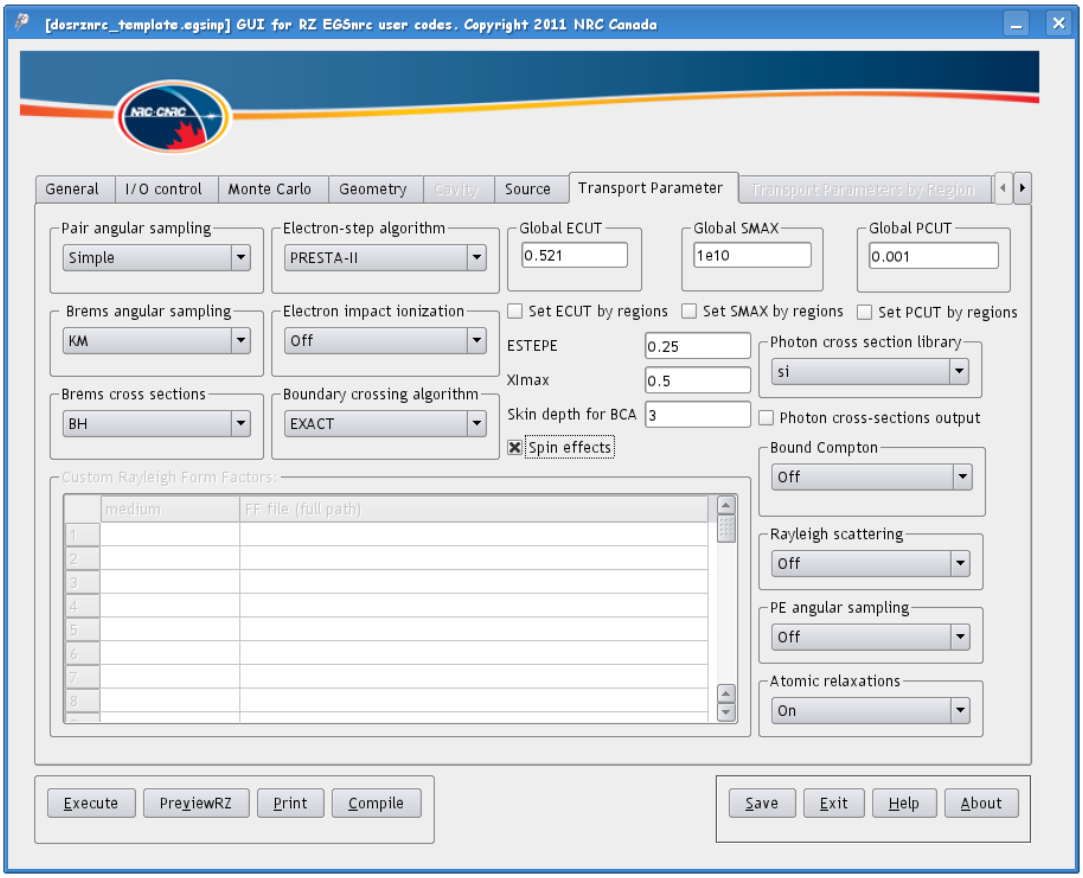


Figure 14: Monte Carlo Transport Parameter Input for the RZ EGSnrc User Codes.

This tab has been updated to most of the latest additions to the `MC trasport parameters` input block. Notably, one can now enter the medium and file names for using custom coherent scattering form factors. Currently the option for defining an arbitrary file with Compton

cross sections is not available in the GUI. This wizard tab is already overloaded and will be split in individual tabs for photon and electron/positron inputs in future releases.

5.8 The Transport Parameters by Regions Tab

For some applications it might be desirable to have some of the quantities defined on a region by region basis. This can be done by checking the corresponding check box or radio button of the quantity chosen in the *Transport Parameter* tab. As soon as the user selects a quantity to be set by region this tab is enabled. Here are tables for each of the quantities than can be set up on a region by region basis. Tables will *only* be enabled for those quantities selected in the previous tab.

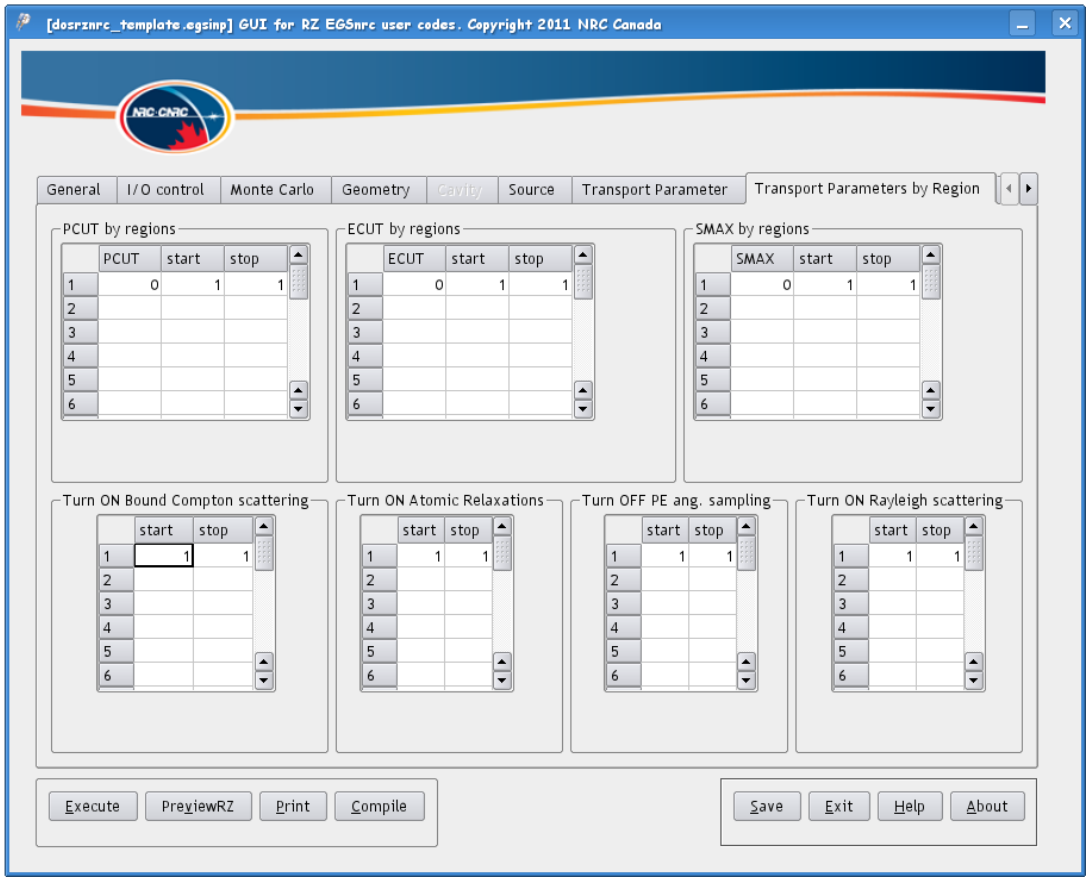


Figure 15: Transport Parameter per Regions Input for the RZ EGSnrc User Codes.

5.9 The Variance Reduction Tab

In this tab the user can define the parameters for the different variance reduction techniques incorporated in the specific user-codes. Techniques like electron range rejection, bremsstrahlung splitting and Russian Roulette are now implemented in EGSnrc. Pathlength biasing and photon forcing are implemented in all user-codes except FLURZnrc, which only includes photon forcing. Additionally photon cross section enhancement is available in DOSRZnrc and CAVRZnrc and a photon splitting technique is also available in CAVRZnrc. See the NRC User Codes Manual for information on these techniques (NRCC report PIRS-702[1]).

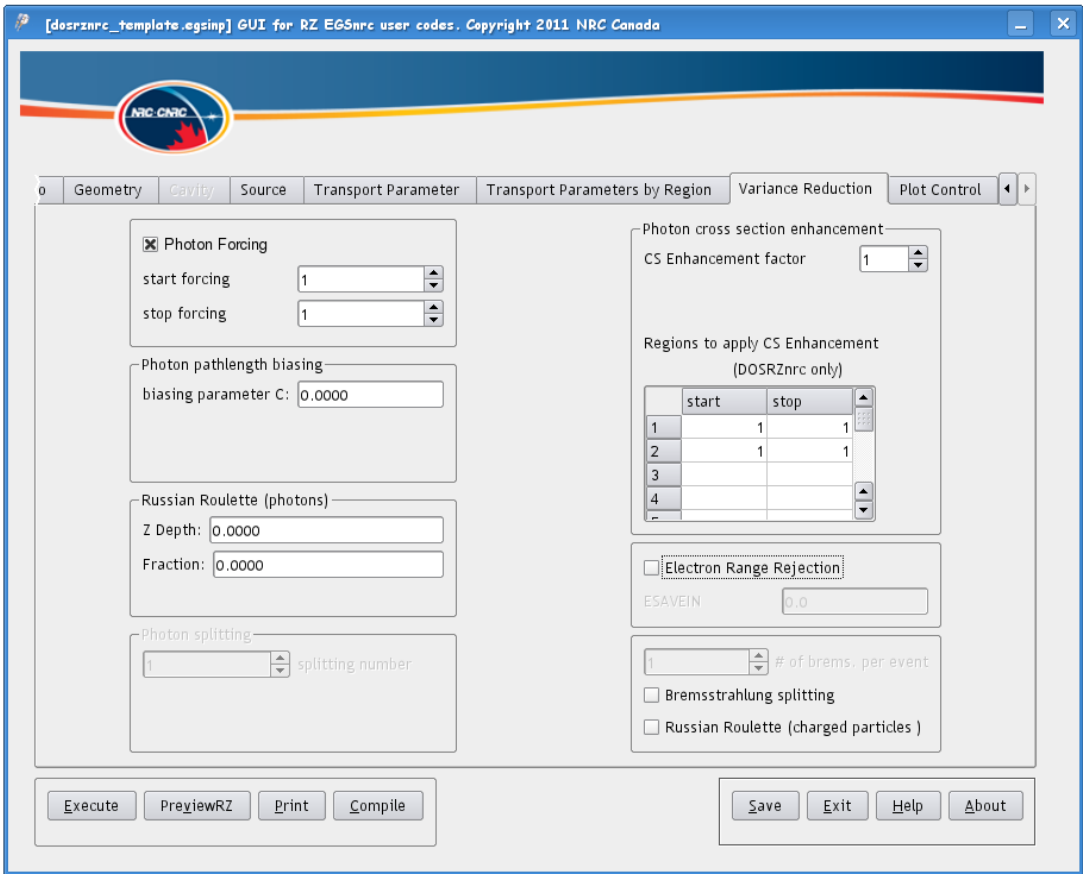


Figure 16: Variance Reduction Parameters for the RZ EGSnrc User Codes.

5.10 The Plot Control Tab

This input block is only relevant for two of the user codes, DOSRZnrc and FLURZnrc. DOSRZnrc has a section of inputs to control plotting of dose *vs* depth/radius results (see figure 17).

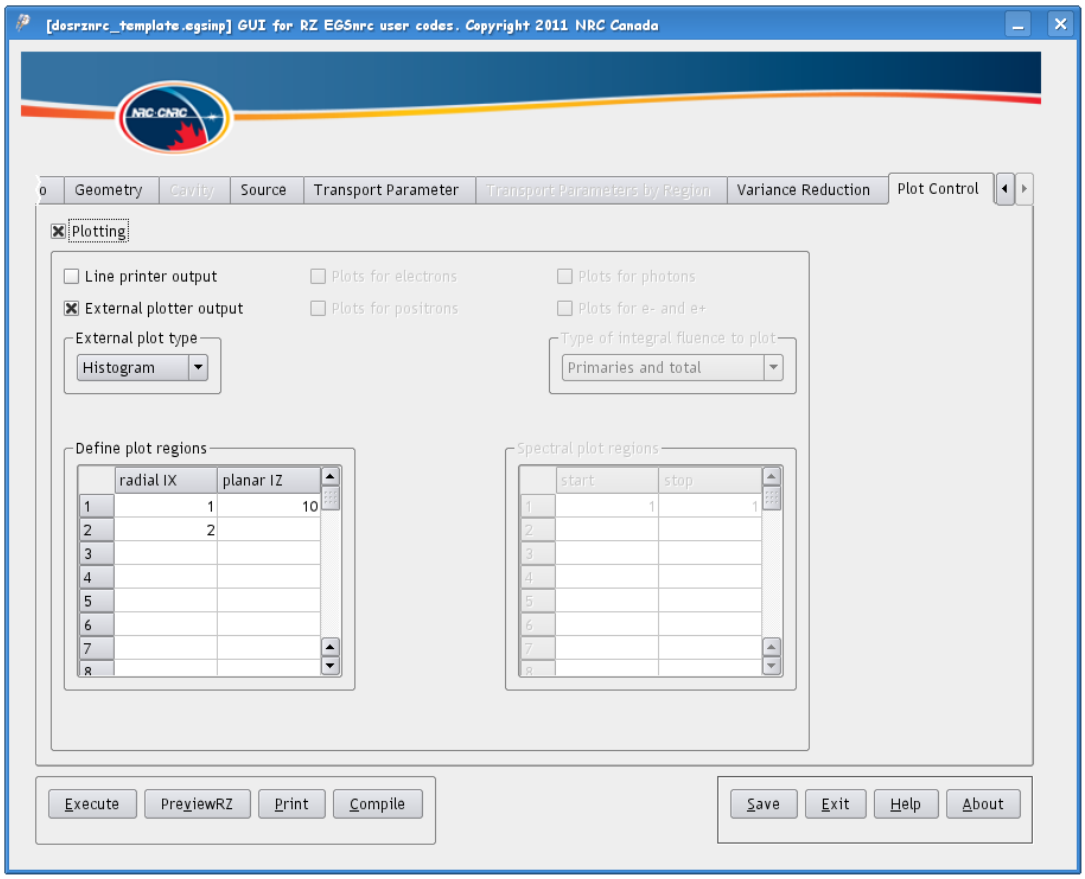


Figure 17: Plot Inputs for the RZ EGSnrc User Code DOSRZnrc.

FLURZnrc has two distinct types of plotting outputs. One class of plots gives integral fluence *vs* position plots in various ways (*vs* depth, *vs* radius). The code also ouputs fluence spectra in specified regions (see figure 18).

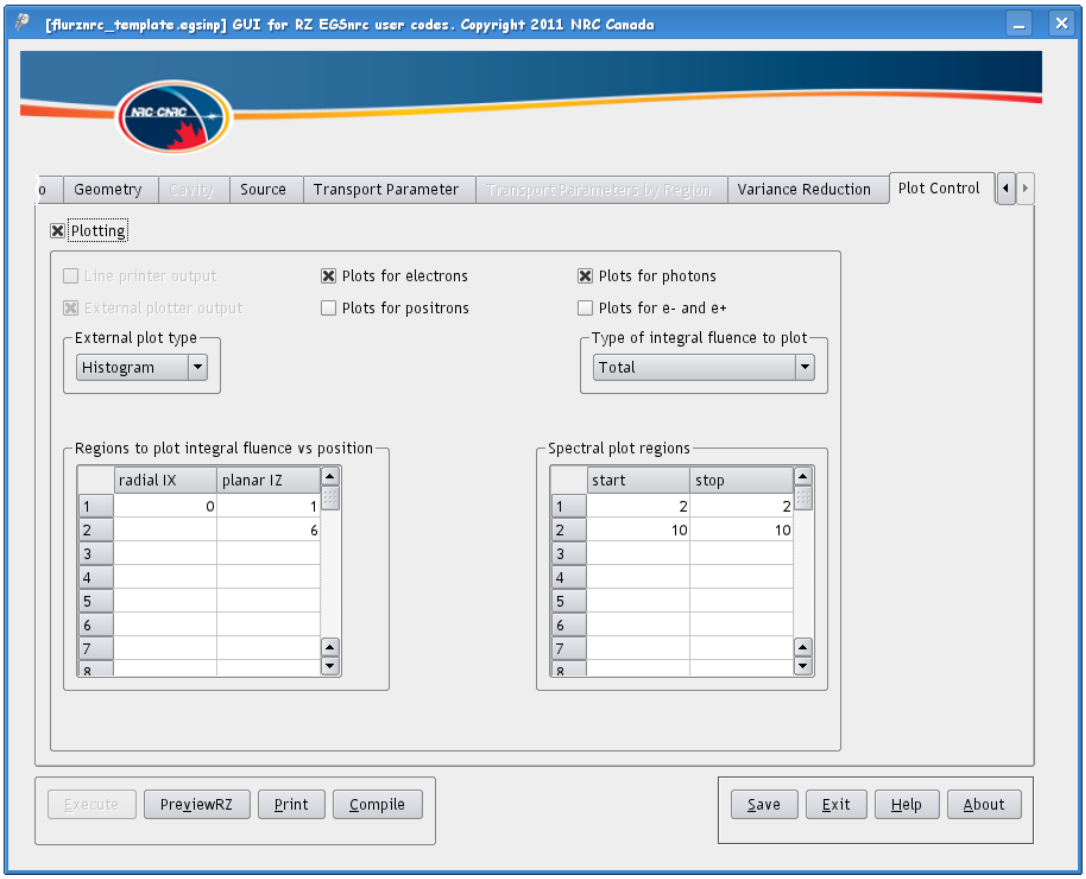


Figure 18: Plot Inputs for the RZ EGSnrc User Code FLURZnrc.

[1] D. W. O. Rogers, I. Kawrakow, J. P. Seuntjens, B. R. B. Walters, and E. Mainegra-Hing. NRC User Codes for EGSnrc. Technical Report PIRS–702(RevC), National Research Council of Canada, Ottawa, Canada, 2010.

[2] I. Kawrakow, E. Mainegra-Hing, and D. W. O. Rogers. EGSnrcMP: the multi-platform environment for EGSnrc. Technical Report PIRS–877, National Research Council of Canada, Ottawa, Canada, 2003.

[3] I. Kawrakow, E. Mainegra-Hing, D. W. O. Rogers, F. Tessier, and B. R. B. Walters. The EGSnrc Code System: Monte Carlo simulation of electron and photon transport. Technical Report PIRS–701 (5th printing), National Research Council of Canada, Ottawa, Canada, 2009.

Index