Accelerator-Driven System Design FLUKA Exercise

Yacine Kadi
(Yacine.kadi@cern.ch)
Department of Energy Science, SungKyunKwan University

Introduction

The goal of these exercises is to give a basic introduction on how to run the Monte Carlo code FLUKA (acronym for FLUktuierende KAskade) [1], a general-purpose tool for simulating the interaction of radiation with matter. This physics code is applied to solve problems on a wide range of fields, from high-energy physics (e.g. study of the ATLAS detector at CERN) to medical and radiation physics (e.g. calculation of doses to astronauts by NASA). Namely, it has been thoroughly used for the design of ADS for waste transmutation and energy production. In fact, the development of the Energy Amplifier, carried out by the Emerging Energy Technologies group at CERN, was performed using FLUKA, among other computer programs. This code was simulating the nuclear cascade produced by high-energy protons (occurring mainly in the spallation target of the device), coupling its results with EA-MC [2], to accurately transport the low-energy neutrons and calculate the burn-up evolution, using the latest point-wise cross-section libraries.

The present exercise is a simplified version of the FLUKA model used in the design of a 4 MW proton-to-neutron converter, performed within Task #2 of the EURopean Isotope Separation On-Line Radioactive Ion Beam Facility Design Study (EURISOL DS) [3]. For more detailed analysis of this spallation target, the final documents of this study may be downloaded from Reference [4].

The FLUKA input defines a cylindrical target, where the type of incident particle, its kinetic energy and the target material composition may be varied, in order to study the effect of these parameters on the general neutronics of the system.

The output files shall show the particle yields and spatial distributions, energy deposition, fission density and particle energy spectrum in the target, residual nuclei distributions, absorbed doses, etc... allowing us to extract useful conclusions concerning the design of the system.

Model Layout

Initially, the beam particles considered are protons with a kinetic energy of 1GeV, following a Gaussian distribution with a 1.0 cm standard deviation in both x and y directions. The target is a 60 cm long and 10 cm radius, PbBi (density: 10.5 g/cm^3) cylinder. Figure 1 illustrates an artistic view of the target, with a 20 cm long, 2 cm radius vacuum along the proton line in order to reduce the possible particle backscattering through the front cap.

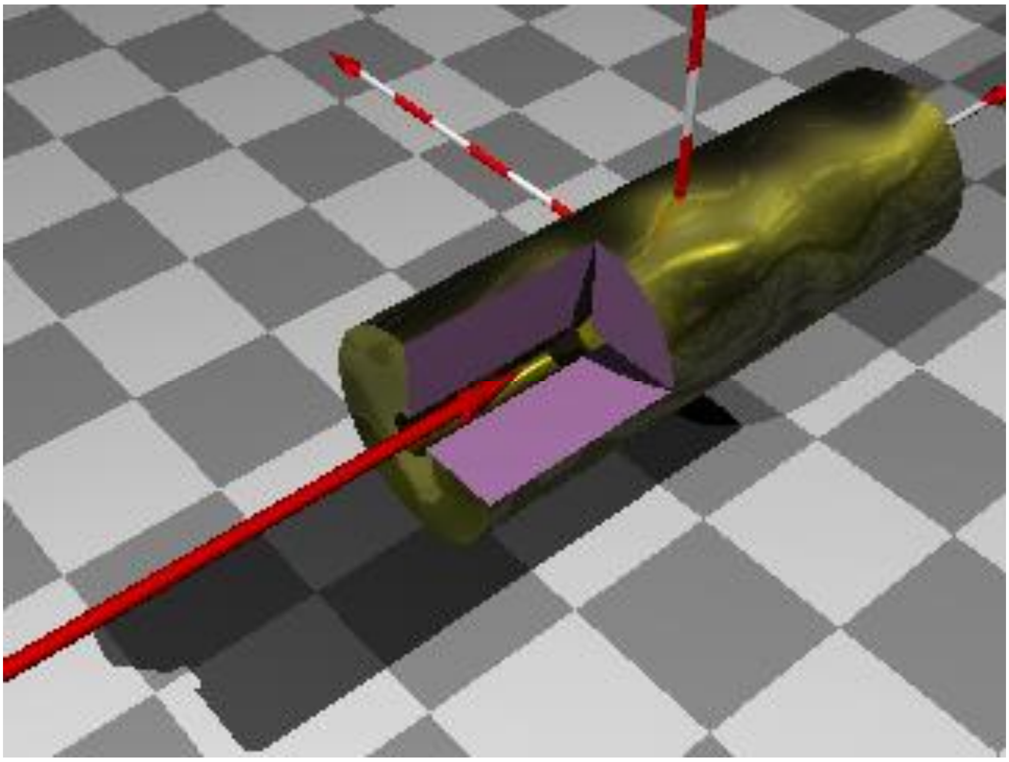

Figure 1. 3-D view of the target used in the exercise.

Generalities about the FLUKA input

FLUKA reads user input from an ASCII “standard input” file with extension .inp. The input consists of a variable number of “commands” (also called “options”), each consisting of one or more “lines” (also called “cards” for historical reasons). The typical structure of a FLUKA input file is the following:

- Titles and comments for documentation purposes (optional, but recommended)
- Definition of the particle source (mandatory)
- Description of the problem geometry (solid bodies and surfaces, combined to partition space into regions) (mandatory)
- Definition of the materials (mandatory unless pre-defined materials are used)
- Material assignments (correspondence material-region, mandatory)
- Definition of the requested “detectors”. Each of these is a phase space domain (region of space, particle direction and energy) where the user wants to calculate the expectation value of a physical quantity such as dose, fluence, etc. Various kinds of detectors are available, corresponding to different quantities and to different algorithms used to estimate them (“estimators”). Detectors are optional, but one at least is expected
- Definition of biasing schemes (variance reduction technique, optional)
- Definition of problem settings such as energy cut-offs, step size, physical effects not simulated by default, particles not to be transported, etc. (optional)
- Initialisation of the random number sequence (mandatory if an estimation of the statistical error is desired)
- Starting signal and number of requested histories (mandatory)

In addition, special commands are available in FLUKA for more advanced problems involving magnetic fields, time-dependent calculations, writing of history files (so-called "collision tapes"), transport of optical photons, event-by-event scoring, calling user-written routines, etc. These options are expected to be requested only by users having some previous experience with the more common commands: therefore they will be mostly ignored in this beginner's guide.

The general structure of the FLUKA command lines or cards (except for the geometry commands) contains:

- one keyword,
- six floating point values (called WHATs),
- one character string (called SDUM)

Not necessarily all WHATs and SDUMs are used. In some cases, a command line can be followed by a line of text (for instance a filename path or a title). Any line having an asterisk (*) in the first position is treated as a comment. All lines (commands, text strings and comments) are echoed on the standard output (the file with extension .out). In case of problems, it is a good idea to check how every line has been printed in the standard output. More information at: http://www.fluka.org/manual/Toc.html

FLUKA Input Cards

This section briefly defines the input cards related to the present exercise, explaining their most relevant options. Therefore, in order to fully understand the implications of each card in the FLUKA input, the user should read the definition of each command, from the on-line manual:

http://www.fluka.org/manual/sect/s010/text.html

The card explanation follows the order of the exercise input, starting by the title and beam definition, followed by the geometry, material assignment and detector cards. The so-called WHATs (from 1 to 6) are the command options, with a trailing text field called SDUM.

Title and Beam Definition

```
***
TITLE
Spallation cylindrical target
DEFAULTS                                                              PRECISIO
***
*
*
* 23456789 123456789 123456789 123456789 123456789 123456789 123456789 123456789
BEAM            -3.0       0.0       0.0  -3.53223  -3.53223       1.0    PROTON
BEAMPOS         0.00      0.00     -30.1
*
*
```

WHAT-1 is the BEAM card that defines the kinetic energy of the primary particles, whereas WHAT-4&5 define the FWHM (2.35*sigma of the PROTON beam). BEAMPOS defines the initial position and direction cosines of the beam particles.

Geometry Definition (GEOBEGIN – GEOEND)

```
GEOBEGIN                                                              COMBNAME
    0    0          0    0          Spallation target
* ******************************************
* BODIES DEFINITION
*
* Blackhole and vacuum
```

```
RPP B1         -1100. 1100. -1100. 1100. -1100. 1100.
RPP B2         -1000. 1000. -1000. 1000. -1000. 1000.
*
* spallation target and case, liquid moderator
ZCC spall      0.0 0.0 10.
XYP VP1        -30.
XYP VP2        30.
*
  END
* ******************************************
* REGIONS DEFINITION
*
* Black hole around the geometry
BlckHole     5   +B1          -B2
* spallation target
Spall        5   +spall       -VP1        +VP2
* Vacuum
Vacuum       5   +B2          -(spall -VP1 +VP2)
  END
GEOEND
```

FLUKA uses a Boolean approach to define geometries (similar to MCNP), first declaring the primitive bodies, e.g. SPH for sphere, XYP for XY infinite planes, ZCC for infinite cylinders in the Z direction etc. The first (consecutive) numbers in all definition lines are the names of the bodies (in our case, they range from 1 to 7). The next numbers in the body definition state some geometrical parameters for each body (such as its centre, radius etc.).

The Boolean operations (addition and subtraction) between bodies are defined in the second part of GEOBEGIN. The initial number (i.e. 001, 002…. 006) defines the name/number of each spatial region. The positive or negative numbers show the inclusion or exclusion of a body, for each region. The ORs indicate the addition of several sub-regions to form a single region. For example, region 003 is formed by ZCC 4 minus ZCC 3, between XYP 6 and 5, plus ZCC 3 between XYP 7 and 6.

Material Definition and Assignment

```
*23456789 123456789 123456789 123456789 123456789 123456789 123456789 123456789
*CARD     WHAT-1    WHAT-2    WHAT-3    WHAT-4    WHAT-5    WHAT-6    SDUM
*_________+_________+_________+_________+_________+_________+_________+
MATERIAL        83.0  208.9804      9.78                             0.0BISMUTH
MATERIAL         1.0     2.014    1.8D-4                             2.0DEUTERIU
*
* COMPOUND MATERIALS
* Lead - Bismuth Eutectic
MATERIAL         0.0       0.0      10.5                             0.0PbBi
COMPOUND        0.55   BISMUTH      0.45      LEAD       0.0            PbBi
* Heavy Water
MATERIAL                             1.0                                HWATER
COMPOUND         2.0  DEUTERIU       1.0    OXYGEN                      HWATER
* ******************************************************************
* ============================= ASSIGNMAT =================================
* ******************************************************************
* ..+....1....+....2....+....3....+....4....+....5....+....6....+....7..
ASSIGNMA    BLCKHOLE  BlckHole                           0.0
ASSIGNMA        PbBi     Spall                           0.0
**ASSIGNMA     MERCURY     Spall                           0.0
ASSIGNMA      VACUUM    Vacuum                           0.0
```

The element materials are defined using the MATERIAL card, where WHAT-1 defines the atomic number of the element (Z), WHAT-2 defines its mass number (A), WHAT-3 defines the material's density and WHAT-4 and SDUM respectively assign a number and a name to the material. Note the upper line, commented by using an * as first character, which defines the alignment of each one of the option cards (WHATs).

The COMPOUND cards (always combined with a MATERIAL card) define a material composed by several elements. In the example above, a mixture of bismuth and natural lead is defined as PbBi, presenting and isotopic composition of 0.55 Bi and 0.45 Pb. Its density is defined as 10.5 g/cm^3, in the preceding MATERIAL card.

The assignment of materials to each region is performed using the ASSIGMAT command, where WHAT-1 shows the material number and WHAT-2&3 define the range of regions on which that material is applied.

Detector Cards (RESNUCLE USRBIN, USRBDX, USRTRACK)

```
*23456789 123456789 123456789 123456789 123456789 123456789 123456789 123456789
* RESIDUAL NUCLEI
*
RESNUCLE     3.0     -50.0                            3.0   100279.63 TarResNuc
*
* Userbins (two dimensional plots)
USRBIN     11.00       8.0    -41.00      100.0             110.0    Neutrons
USRBIN                        -50.00      200.0             320.0      &
*
* USRTRACK: Fluence Estimators; Neutron flux
USRTRACK   -1.0        8.0     -71.0        3.0    100279.63  120.0   Tar-N-Flux
USRTRACK  1.0000       1.E-9                                           &
*
* USRBDX: User Boundary Crossings; Radial Neutron flux
USRBDX      99.0       8.0    -21.000       3.0       7.0   10053.096 Rad-N-Flux
USRBDX    1.0000       1.E-9     120.0                          6.0     &
```

The ultimate objective of this type of the simulations is to understand the effect of radiation interaction with matter. These results are mainly obtained by defining some "detectors", which will give the user an idea on the real interaction. There are several types of detectors, although only four are used in these exercises.

In order to compute the residual nuclei, the RESNUCLE card is used. Note that only static results are obtained, thus no pile-up of elements or on-line decay is accounted for. WHAT-1 defines the type of products to be scored, WHAT-2 defines the logical output file where the results will be dumped (explained in the post-processing), WHAT-5 defines the region for which the residual nuclei are followed, WHAT-6 declares the volume (in cm^3) of that region and SDUM states the name of the detector.

The USRBIN card produces 3D mappings of a certain parameter. In the example above, the R-PHI-Z (WHAT-1 defines the type of binning) neutron flux (WHAT-2 states the particle to be studied, given by the FLUKA number of that particle) is dumped on a file with the extension 41 (WHAT-3, logical output file). The following cards define the R (or X), PHI (or Y) and Z geometrical dimensions of the detector.

The USRTRACK card scores the energy spectrum of a certain particle, inside a region. Thus, WHAT-2 defines the particle type, WHAT-4 defines the region where the particles shall be tracked and WHAT-4 declares the volume of the aforementioned region. WHAT-5 and the WHAT-1&2 in the second card respectively define the number of energy bins and the lower and higher thresholds of the detector.

Finally, the USRBDX card also scores the energy distribution of a certain particle type, but only those crossing the boundary surface between two regions (declared in WHAT-4&5).

Running FLUKA

At this point, the only card necessary to finalise the FLUKA input is the declaration of the number of particles to run, which will impact on the running time and the statistical error of the Monte Carlo simulation. WHAT-1 in the START card declares the number of primary particles to compute (1,000 in the following example).

```
* The first number is the number of events to be processed
START      1.000E+03 1.0000E+9                                1.000E+07
* End the run
STOP
```

FLAIR

Flair is an advanced user interface for FLUKA 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. debugging, compiling, running and monitoring of the status during a run;
3. back-end interface for post-processing of the output files and plot generation through an interface to gnuplot or 3D photorealistic images with povray;
4. library of materials and geometrical objects, for easier editing, storing and sharing among other users and projects;
5. python API for manipulating the input files, post processing of the results and interfacing to gnuplot;

The philosophy of Flair is 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 for each card. Flair works directly with the input file of FLUKA and is able to read/write all acceptable FLUKA input formats. In the input editor the user is working directly with the FLUKA cards using a small dialog for each card. The program displays the card information in an interpreted human readable way. The only exception is that the cards in Flair are so called extended cards where each card is not composed only by 6 *whats* and 1 *sdum* but rather they contains all related information in one unit (comments preceding the card, continuation cards, titles etc).

Running FLUKA

To facilitate the exercise to the students who are unfamiliar with Unix environments, the instructions which should be typed in the command line (i.e. >) are also included ***using this font***.

First, get into the working directory:

>mkdir fluka_exercise1

>cd fluka_exercise1

At this stage, the user must define an environmental variable FLUPRO pointing to the directory where the distribution tar file has been opened. This directory is made available as follows:

Bash shell: use the export command	C or tc shell: use the setenv command
...	...
export FLUPRO=/opt/fluka	setenv FLUPRO /opt/fluka

Of course the definition of FLUPRO can be placed once for ever in the login script.

To start with, the FLUKA input file should be copied to your working directory.

>cp /home/members/ykadi/case1.inp ./

Open the FLUKA input file using a text editor (e.g. kwrite):

>kwrite case1.inp &

Using this editor, one can modify the input and save the changes. Once the changes are done, FLUKA is run by typing:

>$FLUPRO/flutil/rfluka –N0 –M1 case1 &

where case1 is the name of the FLUKA input (without the .inp extension!)

After a while, and if everything run properly, we should obtain a whole set of results, to list these results (list command):

>ls -ltr

```
[ykadi@master ~]$ ls -ltr
total 304
-rw-r--r-- 1 ykadi mb   6442 Sep 22 20:19 case1_allgeom.inp
-rw-r--r-- 1 ykadi mb   3946 Sep 22 20:19 case1.inp
-rw-r--r-- 1 ykadi mb   5062 Sep 22 20:19 case2.inp
-rw-r--r-- 1 ykadi mb   7346 Sep 22 20:19 case3.inp
-rw-r----- 1 ykadi mb   1651 Sep 22 20:34 rancase1001
-rw-r--r-- 1 ykadi mb   9536 Sep 22 20:34 case1001.log
-rw-r--r-- 1 ykadi mb   1651 Sep 22 20:48 rancase1002
-rw-r--r-- 1 ykadi mb 236402 Sep 22 20:48 case1001.out
-rw-r--r-- 1 ykadi mb  22512 Sep 22 20:48 case1001.err
```

FLUKA produces a text output of the run called ***case1001.out***. This file contains useful information on the loading of the input file, the statistics of the run and about the general physics undergone. At the end of this file (open with a text editor ***case1001.out***) we can extract, for example, the neutron yield:

```
Number of secondaries generated in inelastic interactions per beam particle:
 Prompt radiation       Radioactive decays
   5.1901E+01 (100.%)    0.0000E+00 (100.%)
   1.5600E-01 ( 0.3%)    0.0000E+00 ( 0.0%) 4-HELIUM
   0.0000E+00 ( 0.0%)    0.0000E+00 ( 0.0%) 3-HELIUM
   1.6000E-02 ( 0.0%)    0.0000E+00 ( 0.0%) TRITON
   3.9000E-02 ( 0.1%)    0.0000E+00 ( 0.0%) DEUTERON
   0.0000E+00 ( 0.0%)    0.0000E+00 ( 0.0%) HEAVYION
   0.0000E+00 ( 0.0%)    0.0000E+00 ( 0.0%) OPTIPHOT
   0.0000E+00 ( 0.0%)    0.0000E+00 ( 0.0%) RAY
   3.5500E+00 ( 6.8%)    0.0000E+00 ( 0.0%) PROTON
   0.0000E+00 ( 0.0%)    0.0000E+00 ( 0.0%) APROTON
   0.0000E+00 ( 0.0%)    0.0000E+00 ( 0.0%) ELECTRON
   0.0000E+00 ( 0.0%)    0.0000E+00 ( 0.0%) POSITRON
   0.0000E+00 ( 0.0%)    0.0000E+00 ( 0.0%) NEUTRIE
   0.0000E+00 ( 0.0%)    0.0000E+00 ( 0.0%) ANEUTRIE
   1.7448E+01 (33.6%)    0.0000E+00 ( 0.0%) PHOTON
-> 3.0368E+01 (58.5%)    0.0000E+00 ( 0.0%) NEUTRON
   0.0000E+00 ( 0.0%)    0.0000E+00 ( 0.0%) ANEUTRON
   0.0000E+00 ( 0.0%)    0.0000E+00 ( 0.0%) MUON+
   0.0000E+00 ( 0.0%)    0.0000E+00 ( 0.0%) MUON-
   0.0000E+00 ( 0.0%)    0.0000E+00 ( 0.0%) KAONLONG
   1.0300E-01 ( 0.2%)    0.0000E+00 ( 0.0%) PION+
   7.1000E-02 ( 0.1%)    0.0000E+00 ( 0.0%) PION-
```

In our case, 3 GeV protons interacting with the previously defined PbBi target produce approximately 70 neutrons per primary proton. The neutron yield strongly depends on the type of primary particles, the energy of these particles and the target material and dimension. Therefore, the students should now modify the following parameters:

- energy of the primary particle
- type of target material
- type of primary particle
- length, radius of the target

This exercise may be performed by using the shell commands ***mkdir*** *dirname* (to create a directory dirname), ***cp*** *filename dirname/* (to copy a file [filename] into a directory [dirname]), ***cd*** *dirname* (to change directory [into dirname]). Once the directories are set and fluka_example.inp is copied into each one of them, you should edit the example file and change the aforementioned parameters, one by one, filling up tables 1 and 2.

Table 1 will show the evolution of the neutron yield as a function of the proton energy and target material. The materials used are already defined in the input file (C, Al, Fe, Ta, Hg, PbBi and natural U). Hence only the ASSIGNMAT card of region 3 needs to be changed. The students should choose 4 of them (for example, Al, Ta, Hg and nat U) and run for the proton energies suggested in Table 1, by modifying WHAT-1 in the BEAM card of the input file.

Table 1. Neutron yields for different materials and proton energies.

		Proton energy (GeV)					
		0.1	0.25	0.5	1.0	1.5	2.0
Target material	C						
	Al						
	Fe						
	Ta						
	Hg						

	PbBi						
	Nat U						

Likewise, Table 2 should be filled with the neutron yields for different source particles (protons, deuterons, helium-4 and electrons). The name of the primary particles should be changed in the SDUM field of the BEAM card, using FLUKA's particle name (on-line manual: http://www.fluka.org/manual/Toc.html, section 5).

The data in these tables will bring forward that, for energies below 1 GeV, the increase in neutron yield as a function of energy follows an exponential pattern, whereas for energies above ~1.5 GeV this progression is attenuated, roughly to a linear increase. Gnuplot may be used to plot this data, producing results similar to those illustrated by Figure 1.(a), where the neutron yield evolution as a function of energy is shown.

Figure 1.(b) presents the evolution of the neutron yield efficiency by normalising the previous results over the incident particle energy. This procedure effectively allows us to infer the optimum energy of neutron production through spallation.

Table 2. Neutron yields for projectile particles and incident energies.

		Energy of the projectile (GeV)					
		0.1	0.25	0.5	1.0	1.5	2.0
Projectile particle	Proton						
	Deuteron						
	Helium-3						
	Electron						

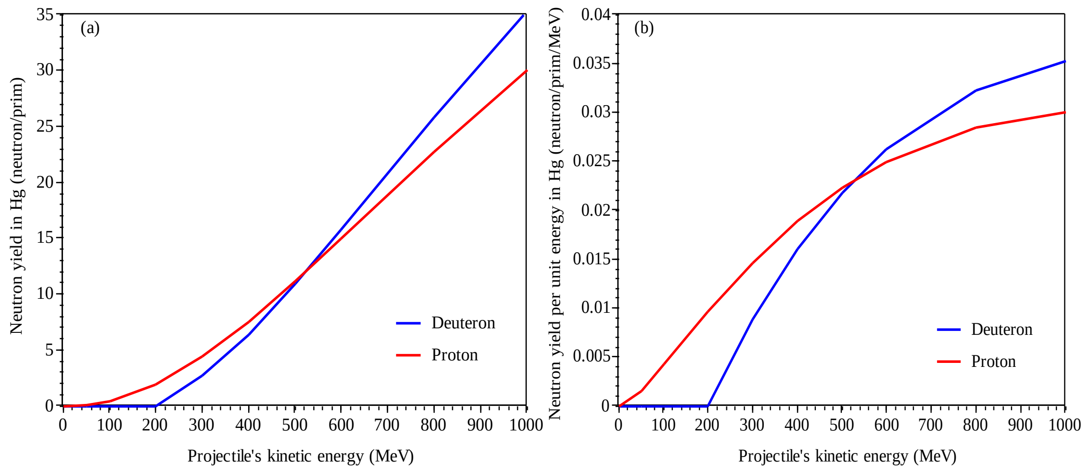

Figure 1. Neutron yield in Hg for different types of projectile.

Table 3. Neutron yields for increasing radii and incident energies: protons on PbBi (60 cm long).

		Energy of the projectile (GeV)					
		0.1	0.25	0.5	1.0	1.5	2.0
Target Radius cm	10						
	15						
	20						
	25						

Table 4. Neutron yields for increasing length and incident energies: protons on PbBi (10 cm radius).

		Energy of the projectile (GeV)					
		0.1	0.25	0.5	1.0	1.5	2.0
Target length cm	60						
	80						
	100						
	120						

Conclusion

We have now finished this brief introduction to the use of FLUKA. You should now be able to configure basic models and extract a wide range of results. For this purpose, the scripts used in the previous exercises are included in this document as annexes.

For more in-depth information on FLUKA and the physics behind it, visit the official website at http://www.fluka.org.

References

[1] *"FLUKA: Status and Prospective for Hadronic Applications"*, A. Fassò, A. Ferrari, J. Ranft, P.R. Sala, invited talk in the Proceedings of the MonteCarlo 2000 Conference, Lisbon, October 23--26 2000, A. Kling, F. Barao, M. Nakagawa, L. Tavora, P. Vaz eds., Springer-Verlag Berlin, p. 955-960, 2001. Also: *"Electron-photon transport in FLUKA: Status"*, A. Fassò, A. Ferrari, P.R. Sala, invited talk in the Proceedings of the MonteCarlo 2000 Conference, Lisbon, October 23--26 2000, A. Kling, F. Barao, M. Nakagawa, L. Tavora, P. Vaz eds., Springer-Verlag Berlin, p. 159-164, 2001. For additional information: *http://www.fluka.org*

[2] Yacine Kadi, *"The EA-MC Monte Carlo Code Package"*, in Proc. of the Fifth International Meeting on Simulating Accelerator Radiation Environment — SARE-5: Models and Codes for Spallation Neutron Sources, OECD Headquarters; Paris, France (July 2000).

[3] *"EURISOL DS; EURopean Isotope Separation On-Line Radioactive Ion Beam Facility Design Study"*, EC – FP6 Research Infrastructure Action-Structuring the European Research Area, Project Contract no. 515768 RIDS.

[4] http://www.eurisol.org, refer to task#2 activities