

Chapter 2

The FLUKA Monte Carlo Code

Monte Carlo (MC) calculations are a standard tool in several areas of research, and particularly in physics. This name is used to refer to any technique in which complex models are evaluated by generating successive random samples and interpreting statistically the global result after numerous iterations. This technique dates back to the 40s, having being developed in the framework of the Manhattan Project, and owes its name to the famous casino in Monaco [1].

Even if born in the frame of gamble, MC is now used in a variety of contexts, including for instance economics: it is fact today at the base of risk measurements computation, portfolios sensitivity evaluation and so on.

In particle physics, the main application of this method is to simulate particle transport and interactions in media. From the knowledge of the physics of the elementary collision processes (which accuracy strongly influences the one of the entire technique), MC simulations actually create and track a high number of primary particles, as well as the secondaries created in interactions.

A certain amount of information (depending of the desired depth of the study undertaken) is than stored and analyzed.

The main advantage of this approach is that it permits to study various different configurations (for example geometries or primary particles) before (or even without) doing specific measurements.

The other use of simulation is to support experimental data comprehension, by helping to recognize unexpected structures and phenomena.

In parallel, MC techniques find wide applications even in medicine, and in particular in nuclear medicine. Each radiotherapeutic treatment is in fact preceded by MC simulations aimed at verifying the exact localization of energy deposition within the body.

In this Thesis work I used MC simulations to study the response to different kinds of signal of various prototypes of probe, thus testing and optimizing its design, and to evaluate radio protection issues due to the injected activity.

FLUKA (*FLUktuierende KAskade*) is a general purpose tool for calculations of particle transport and interactions with matter, developed within a collaboration between

INFN (*Istituto Nazionale di Fisica Nucleare*) and CERN (*European Council for Nuclear Research*) [2].

Its original application was accelerator shielding, but its development in the past years led it to be a common tool for numerous applications, from ion beam and electron acceleration to calorimetry, activation and dosimetry studies, being able to simulate about 60 different particles with energies ranging from the keV up to thousands of TeV.

The history of FLUKA began in 1962, when Prof. Johannes Ranft at CERN wrote the first high-energy Monte Carlo transport code. Starting from those early pioneer attempts, it is possible to distinguish three different generation of FLUKA codes along the years, which can be roughly identified as the FLUKA of the 70s (main authors J. Ranft and J. Routti), the FLUKA of the 80s (P. Aarnio, A. Fassò, H.J. Møhring, J. Ranft, G.R. Stevenson), and the FLUKA of today developed under the INFN CERN Collaboration Agreement (A. Fassò, A. Ferrari, J. Ranft and P.R. Sala).

Even if every new generation of the code originates from the previous one, each new version represented not only an improvement of the existing program, but rather a quantum jump in the code physics, design and goals. In this sense, the name FLUKA has been preserved as a reminder of this historical development, even if the present code has pretty much nothing to do with the versions which were released before 1990.

FLUKA is developed using the FORTRAN 77 language. A graphical user interface named *Flair* (FLUKA Advanced Interface) has been developed using Python [3]. For most applications, no programming is required from the user, even if a number of Fortran-77 routines are available in case of special requirements.

Efficiency is achieved by relying heavily on to table look-up sampling and a systematic use of double precision has a great impact on overall accuracy. To obtain a reasonable flexibility while minimising the need for user-written code, the program has been provided with a large number of options available to the user, and has been completely restructured introducing dynamical dimensioning.

2.1 Physics in FLUKA

The strength of FLUKA is in its physical models: microscopic models are adopted whenever possible, consistency among all the reaction steps and/or reaction types is ensured, conservation laws are enforced at each step, and crosschecks between results and experimental data are done at single interaction level. As a result, final predictions are obtained with a minimal set of free parameters fixed for all energy, target and projectile combinations. Therefore results in complex cases, as well as properties and scaling laws, arise naturally from the underlying physical models, predictivity is provided where no experimental data are directly available, and correlations within interactions and among eventual shower components are preserved.

The transport of electrons and photons in FLUKA (EMF, for Electro Magnetic FLUKA) handles all interactions and scattering processes, including photon nuclear interactions.

The electromagnetic sector is fully coupled to the hadronic one, for instance photons from nuclear deexcitation are directly transported by EMF, and photonuclear interactions are treated in the same framework as hadronic interactions.

For each kind of particle, a set of relative effects are taken into account regarding transport and interactions with matter.

2.1.1 Electrons and Positrons

Since the origins, the algorithm used by FLUKA for charged particles includes:

- A complete multiple-Coulomb scattering treatment giving the correct lateral displacement even near a boundary;
- the Landau–Pomeranchuk–Migdal suppression effect;
- Ter-Mikaelyan polarisation effect in the soft part of the bremsstrahlung spectrum;
- positron annihilation in flight and at rest;
- electrohadron production via virtual photon spectrum and Vector Meson Dominance Model;
- the bremsstrahlung differential cross sections of Seltzer and Berger have been extended to include the finite value at tip energy, and the angular distribution of bremsstrahlung photons is sampled accurately;
- the variations with energy of the discrete event cross sections and of the continuous energy loss in each transport step are taken into account exactly;
- differences between electron and positron are taken into account concerning both Bremsstrahlung and Stopping-Power;
- delta-ray production via Bhabha and Moller scattering.

2.1.2 Photons

Photons interactions are taken into account as carefully as charged particles ones. In each interaction, FLUKA is able to describe:

- pair production with actual angular distribution of electrons and positrons;
- photoelectric effect with actual photoelectron angular distribution, detailed interaction on six K and L single sub-shells, optional emission of fluorescence photons and approximate treatment of Auger electrons;
- Rayleigh scattering;
- photon polarization taken into account for Compton, Rayleigh and Photoelectric effects;

- Compton scattering, considering both binding effects and orbital motion of all electronics shells of all elements (particularly important for low energy photons).

2.1.3 *Cut-Off Energy*

An important aspect in Monte Carlo simulations is the correct tuning of cut-off energies for both transported and produced particles. In FLUKA several sets of these parameters are available via optional card `DEFAULTS` (e.g. for hadrontherapy or dosimetry applications), which takes a default value even if not declared. However, it is often convenient to override some of the default cut-offs according to particular needs in term of performance or accuracy.

Transport cut-offs, or thresholds, are set with command `PART-THRes` for hadrons and muons, with `EMFCUT` for electrons, positrons and photons, and with `LOW-BIAS` for low-energy neutrons. Despite the similar functionality of the three commands, there are important differences in their syntax and in the way the threshold is implemented for the three families of particles.

`PART-THRes` can assign different transport thresholds to different particles, but the thresholds are the same in all materials and regions. When the hadron or muon energy becomes lower than the relevant threshold, the particle is not stopped but ranged out in a simplified way.

`EMFCUT` can assign instead transport thresholds on a region basis for electrons positrons and photons: on the other hand no ranging out is performed, due to the difficulty to clearly define electron ranges.

`LOW-BIAS` is used to set the transport threshold for low-energy neutrons, also on a region basis, but as a group number rather than an energy.

Two input commands can set particle production cut-offs, respectively for heavy particles and for electrons, positrons and photons. Thresholds for delta ray production by charged hadrons and muons are assigned, on a material basis, by means of option `DELTARAY`. Energy transfers to electrons lower than the threshold are handled in the continuous slowing down approximation. Production of bremsstrahlung by electrons and of Moller/Bhabha secondary electrons is simulated explicitly above thresholds set on a material basis with option `EMFCUT`.

2.2 **Geometry in FLUKA**

FLUKA is able to manage very complex geometries via an improved version of the Combinatorial Geometry package. There are two fundamental concepts in CG: bodies and regions.

Originally, bodies were defined as convex solid bodies (finite portions of space completely delimited by surfaces). In FLUKA, the definition has been now extended to include infinite cylinders (circular and elliptical) and planes (half-spaces).

Regions are defined as combinations of bodies obtained by boolean operations: Union, Subtraction and Intersection. Each region is not necessarily simply connected (it can be made of two or more non contiguous parts), but must be of homogeneous material composition.

Being not possible for the ray tracing routines to track particles across the outermost boundary, all the regions must be contained within a surrounding *blackhole* (an infinitely absorbing material), so that all escaping particles are absorbed. It is suggested to make the external blackhole region rather big, in order not to interfere with possible further modifications of the problem layout.

Really important, and not so easy condition to respect, is that each point of space MUST belong to one and only one region.

This combinatorial approach paves the way to complex geometries and powerful capabilities, based on Voxel Geometry.

2.2.1 Voxel Geometry—CT Import

In FLUKA it is possible to describe a complex geometry in terms of *voxels*, that are tiny parallelepipeds forming a 3-dimensional grid.

In principle this can be done with any geometry, but it is especially useful when translating a CT scan of a human body into a phantom. In this context, the word “organ” is used to indicate a contiguous group of voxels (or even more than one group) made of the same material. The code handles each organ as a Combinatorial Geometry region, possibly in addition to other conventional “non-voxel” regions defined by the user, and assigns automatically to each organ a new region number.

To this aim, FLUKA is able to receive in input directly the *DICOM* files of the scan. The conversion between Hounsfield unit (HU, the unit used in CT to parametrize the stopping power of a certain tissue, i.e. the grade of gray in a image) and the material is performed by means of two text files. A first one is a table that assigns to each HU value a tissue density, while a second file assigns to each HU interval a material. This is a significantly nontrivial conversion, due to very large variations between patients and body zones. Different of these files are thus available, in relation to the imported ones (head, body, etc.).

The result of such importation process are showed in Fig. 2.1.

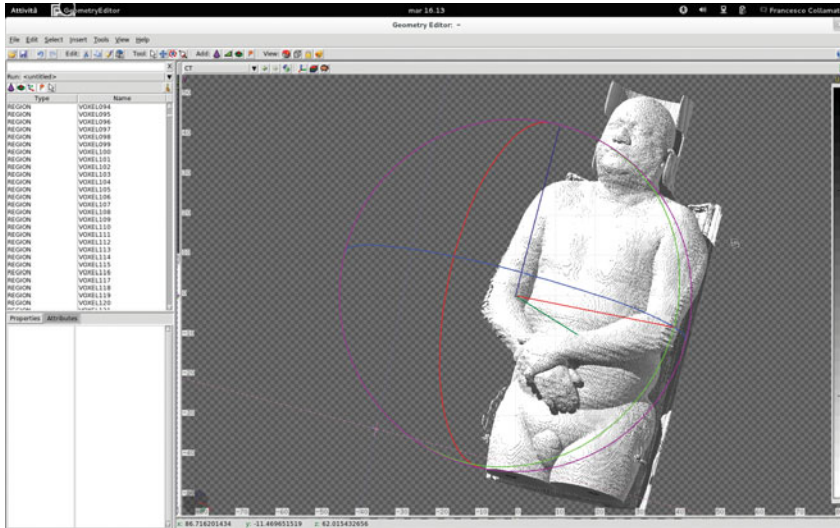


Fig. 2.1 Result of the importation of CT dicom files in FLUKA. On the *left* several voxel volumes automatically created by FLUKA are listed

2.3 Input File Structure in FLUKA

The input file of a FLUKA simulation consist in an ASCII file with extension `.inp`. In this file a series of instructions (named *cards*) are written in each line, following this order:

- Title and comments;
- Description of the problem geometry;
- Definition of the materials;
- Material assignment;
- Definition of the particle source;
- Definition of requested detectors that the user wants to calculate various physical quantities such as dose, fluence, etc.;
- Definition of biasing schemes;
- Definition of problem settings such as energy cut-offs, step size, etc.;
- Initialization of the random number sequence;
- Starting signal and number of requested histories.

FLUKA provides a BEAM card by which it is possible to choose the particle kind, its energy or momentum (with the eventual spread), its starting position and direction. For more complex sources a devoted *user routine* must be used.

2.4 Output in FLUKA

Every FLUKA simulation automatically produces three text output files, named as the .inp file, with the extensions .out, .log and .err.

These last two files contain useful informations in case of errors during the simulation helping to solve them.

The FLUKA standard output file (.out) instead contains plenty of information about the executed run:

- A header with the FLUKA version and the time when the output was printed;
- A straight echo of the input cards and the geometry input, useful to quickly check if all the cards are correctly read and interpreted;
- Informations about basic nuclear data files and physical models used in the simulation process;
- Material parameters related to multiple scattering;
- Memory allocation information;
- Table of correspondence between materials used in the run and materials in the low-energy neutron cross section library;
- Information on the low-energy neutron cross section
- Material-dependent parameters: ionization energy losses, δ -rays and Bremssstrahlung threshold, transport thresholds;
- List of FLUKA particles
- Interpreted input summary
- Scoring summary: list of each estimators requested;
- Materials scattering length;
- Energy balance: percentage of the initial energy deposited and/or lost on each region of the simulation.

Beside the .out file, a number of scoring possibilities are available by means of SCORE cards. Via this cards FLUKA can score particle fluences, current, track length, energy spectra, Z spectra, energy deposition, dose deposition etc.

The possible detectors are:

- USRTRACK and USRCOLL: score average $d\phi/dE$ (differential fluence) of a given type or family of particles in a given region.
- USRBDX: score average $d^2\phi/dEd\Sigma$ (double-differential fluence or current) of a given type or family of particles on a given surface.
- USRBIN scores the spatial distribution of energy deposited, or total fluence (or star density, or momentum transfer) in a regular mesh (cylindrical or Cartesian) described by the user. Usually the results from USRBIN are normalised per unit volume and per unit primary weight.
- USRYIELD scores a double differential yield of particles escaping from a surface. The distribution can be with respect to energy and angle, or other specific quantities.

References

1. *The Beginning of the Monte Carlo Method*. Special Issue (Los Alamos: Los Alamos Science, 1987)
2. G. Battistoni et al., The fluka code: description and benchmarking, in *Proceedings of the Hadronic Shower Simulation Workshop 2006*, Fermilab, 6–8 Sept 2006 (2007), pp. 31–49
3. V. Vlachoudis, FLAIR: a powerful but user friendly graphical interface for FLUKA, in *Proceedings of the International Conference on Mathematics, Computational Methods and Reactor Physics*, (Saratoga Springs, New York, 2009)

An Intraoperative Beta–Probe for Cancer Surgery

Collamati, F.

2016, XIV, 100 p. 55 illus., 24 illus. in color., Hardcover

ISBN: 978-3-319-33698-5