Il Codice Monte Carlo FLUKA e le sue applicazioni in radioterapia e adroterapia

Giuseppe Battistoni
INFN Milano

Overview:

General Monte Carlo concepts:

- Monte Carlo foundations
- Sampling techniques
- Statistical issues

The FLUKA code

- History, general content, design criteria
- Distribution and licensing
- Short review of applications
- A few words on physics models
- The FLUKA geometry
- The graphical user interface

The application to particle therapy

- The relevant physics
- The voxel geometry
- Import of CT scans
- Treatment Planning and PET in-beam, examples
- Simulation of instruments

A demonstrative example

Monte Carlo* mathematical foundation:

Several possible ways of defining Monte Carlo (MC):

- A mathematical method for Numerical Integration
  - Random sampling techniques
  - Convergence, variance reduction techniques…
- A computer simulation of a Physical Process
  - Physics
  - Tracking
  - Scoring…

Both are valid, depending on the problem one or the other can be more effective

* Monte Carlo method "inventors": Von Neumann, Ulam, Fermi, Metropolis in the late 40's

Integration efficiency:

- Traditional numerical integration methods (Simpson, etc), converge to the true values as $N^{-1/n}$ where N = number of "points" (interval), and n = number of dimensions
- Monte Carlo converges instead as $1/\sqrt{N}$

Number of dimensions	Traditional methods	Monte Carlo	Remark
$n = 1$	$1/N$	$1/\sqrt{N}$	MC not convenient
$n = 2$	$1/\sqrt{N}$	$1/\sqrt{N}$	About equivalent
$n > 2$	$1/\sqrt[n]{N}$	$1/\sqrt{N}$	MC converges faster

A typical particle transport Monte Carlo problem is a 7-D problem!
x, y, z, p_x, p_y, p_z and t !!

Random Sampling: the key to Monte Carlo!

The central problem of the Monte Carlo method:

Given a Probability Density Function (pdf), f(x), generate a sequence of x's distributed according to f(x) (x can be multi-dimensional)

The use of random sampling techniques is the distinctive feature of Monte Carlo

The use of Monte Carlo to solve the integral Boltzmann transport equation consists of:

- Random sampling of the outcome of physical events
- Geometry and material description of the problem

(Pseudo) Random numbers:

- Basis for all Monte Carlo integrations are random numbers, *i.e.* values of a variable distributed according to a pdf (probability distribution function).
- In real world: the random outcome of a physical process
- In computer world: pseudo-random numbers
- The basic pdf is the uniform distribution:

$$f(\xi) = 1 \qquad 0 \le \xi < 1$$

- Pseudo-random numbers are sequences that reproduce the uniform distribution, constructed from mathematical algorithms.
- All computers provide a pseudo-random number generator (or even several of them). In most computer languages (e.g., Fortran 90, C) a PRNG is even available as an intrinsic routine

Sampling from a distribution:

Sampling from a discrete distribution:

- Suppose to have a *discrete* random variable x, that can assume values $x_1, x_2, \ldots, x_n, \ldots$ with probability $p_1, p_2, \ldots, p_n, \ldots$
- Assume $\sum_i p_i = 1$, or normalize it
- Divide the interval [0,1) in n subintervals, with limits

$$y_0 = 0, \; y_1 = p_1, \; y_2 = p_1 + p_2, \; \ldots$$

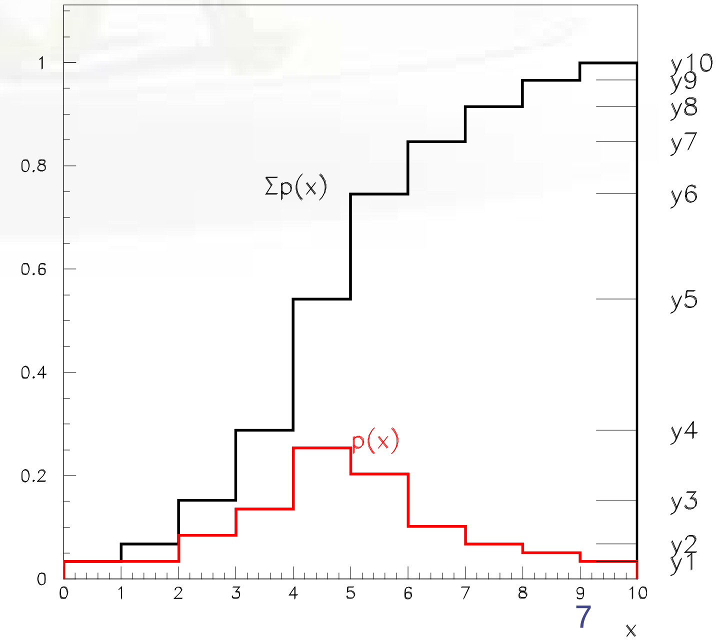
- Generate a uniform pseudo-random number ξ
- Find the interval i^{th} y-interval such that

$$y_{i-1} \le \xi < y_i$$

- Select $X = x_i$ as the sampled value

Since ξ is uniformly random:

$$P(x_i) = P(y_{i-1} \le \xi < y_i) = y_i - y_{i-1} = p_i$$



Sampling from a distribution:

Sampling from a generic continuous distribution:

- Integrate the distribution function ***f(x)***, analytically or numerically, and normalize to 1 to obtain the normalized cumulative distribution

$$F(\xi) = \frac{\int_{x_{\min}}^{\xi} f(x)dx}{\int_{x_{\min}}^{x_{\max}} f(x)dx}$$

- Generate a uniform pseudo-random number ξ
- Get the desired result by finding the inverse value $X = F^{-1}(\xi)$, analytically or most often numerically, i.e. by interpolation (table look-up)

Since ξ is uniformly random:

$$P(a < x < b) = P(F(a) \leq \xi < F(b)) = F(b) - F(a) = \int_a^b f(x)dx$$

Example:

Take $f(x) = e^{-\frac{x}{\lambda}}$, x ∈ [0,-∞)

Cumulative distribution:

$$F(t) = \int_0^t e^{-\frac{x}{\lambda}} dx = \lambda \times \left(1 - e^{-\frac{t}{\lambda}}\right)$$

Normalized:

$$F'(t) = \int_0^t \frac{e^{-\frac{x}{\lambda}}}{\lambda} dx = 1 - e^{-\frac{t}{\lambda}}$$

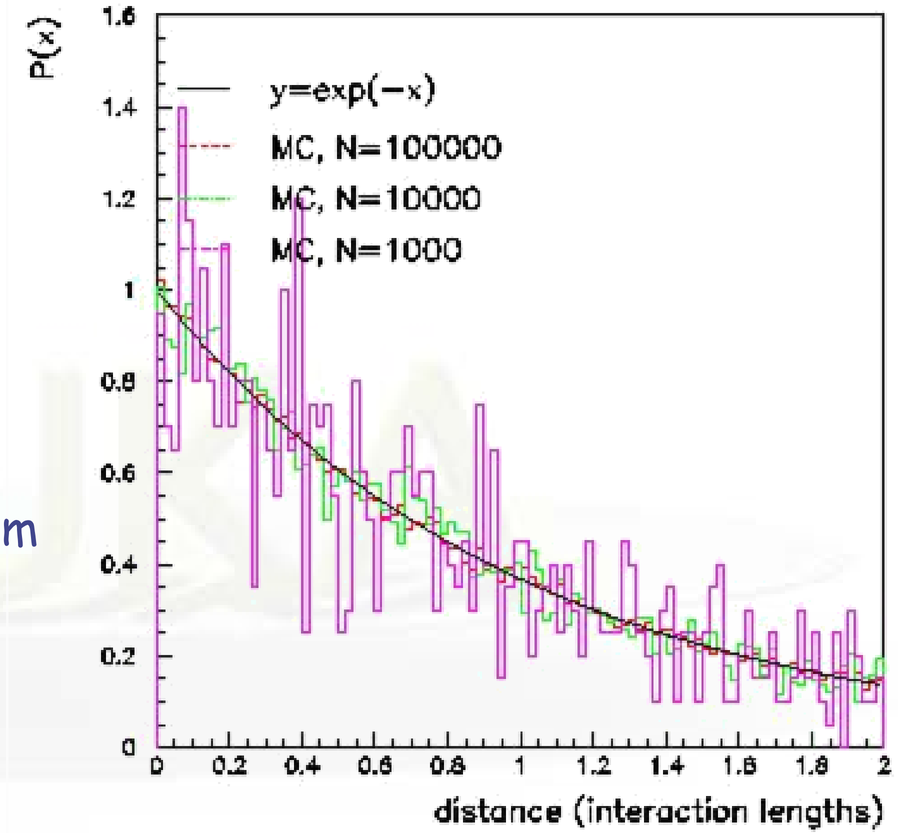
Generate a uniform pseudo-random number $\xi \in [0,1)$

$$1 - e^{-\frac{t}{\lambda}} = \xi$$

Sample *t* by inverting

$$t = -\lambda \ln(1 - \xi)$$

Repeat N times



Practical rule: a distribution can be sampled directly if and only if its pdf can be integrated and the integral inverted

Sampling from a distribution: rejection technique

Rejection procedure:

- Let be $f'(x)$, a normalized distribution function, which cannot be sampled by integration and inversion
- Let be $g'(x)$, a normalized distribution function, which can be sampled, and such that $Cg'(x) \geq f'(x), \forall\ x \in [x_{min}, x_{max}]$
- Sample X from g'(x), and generate a uniform pseudo-random number $\xi \in [0,1)$
- Accept X if $\xi < f'(X)/Cg'(X)$, if not repeat the previous step
- The overall efficiency (accepted/rejected) is given by:

$$R = \int \frac{f'(x)}{Cg'(x)} g'(x)\,dx = \frac{1}{C}$$

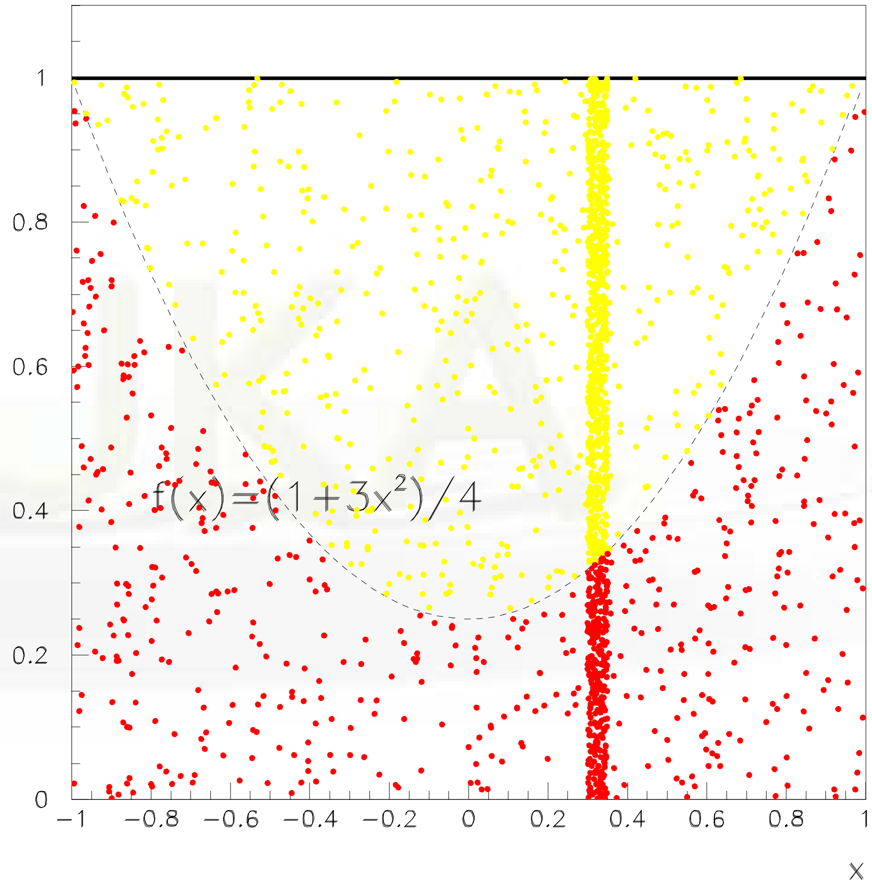
- and the probability that X is accepted is unbiased:

$$P(X)dX = \frac{1}{R} g'(X)dX \times \frac{f'(X)}{Cg'(X)} = f'(X)dX$$

Sampling from a distribution: example

Rejection procedure:

- Let be $f'(x) = (1+3x^2)/4$, $x \in [-1,1]$,
- Take $g'(x)=1/2$, $C=2$
- Generate two uniform pseudo-random numbers $\xi_1, \xi_2 \in [0,1)$
- Accept $X=2\xi_1-1$ if $\xi_2 < (1+3X^2)/4$, if not repeat



Particle transport Monte Carlo:

Assumptions:

- Static, homogeneous, isotropic, and amorphous media (and geometry)
- Markovian process: the fate of a particle depends only on its actual properties, not on previous events or histories
- Particles do not interact with each other
- Particles interact with individual atoms/nuclei/molecules (invalid at low energies)
- Material properties are not affected by particle reactions

⇓

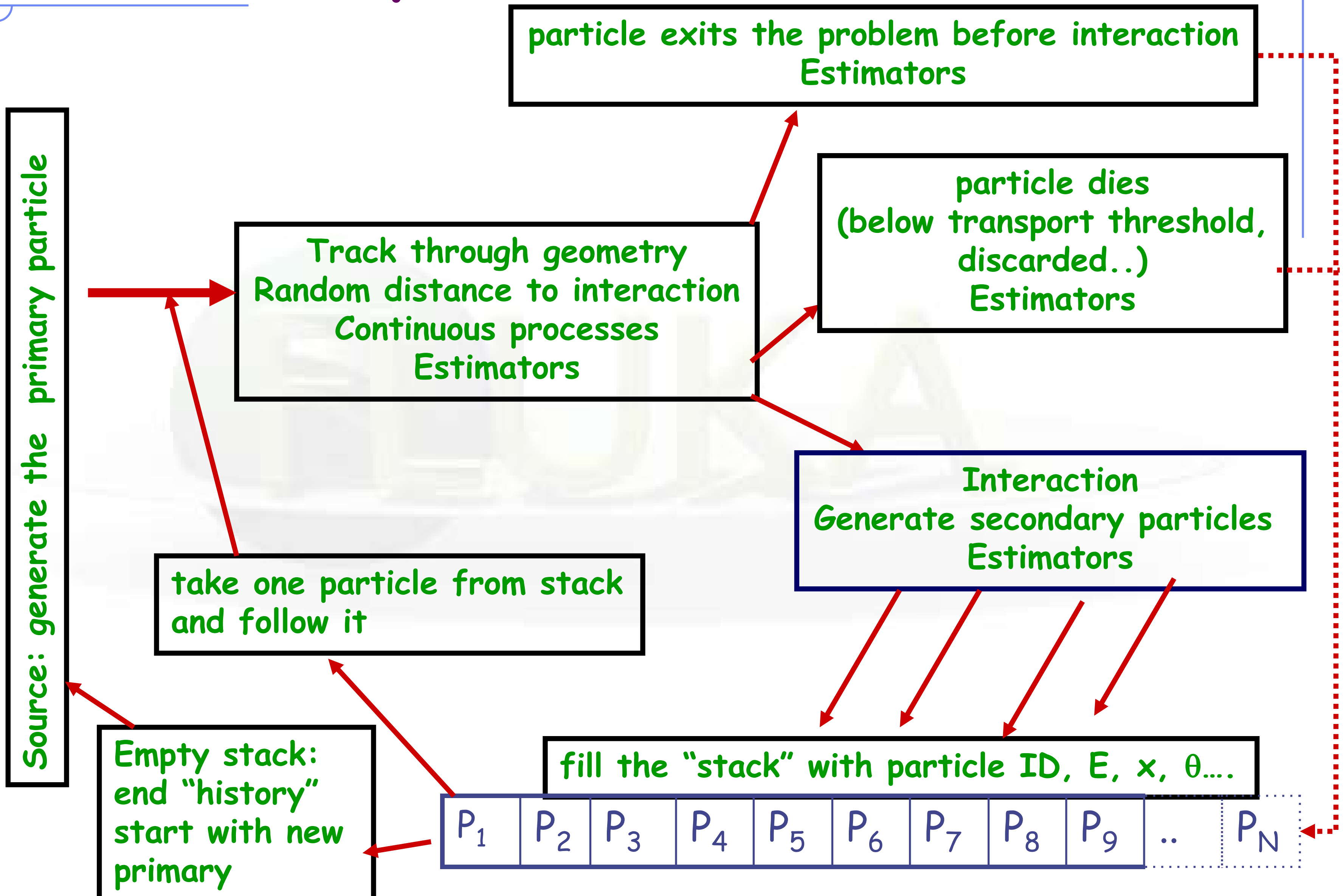
The superposition principle can be used

Particle transport Monte Carlo:

Application of Monte Carlo to particle transport and interaction:

- Each particle is followed on its path through matter.
- At each step the occurrence and outcome of interactions are decided by random selection from the appropriate probability distributions.
- All the secondaries issued from the same primary are transported before a new history is started.
- The accuracy and reliability of a Monte Carlo depends on the models or data on which the pdfs are based
- Statistical accuracy of results depends on the number of “histories"
- Statistical convergence can be accelerated by “biasing" techniques.

Practical implementations



Statistical Errors:

- Can be calculated for single histories, or for batches of several histories each
- Distribution of scoring contributions by single histories can be very asymmetric (many histories contribute little or zero)
- Scoring distribution from batches tends to Gaussian for $N \to \infty$, provided $\sigma^2 \neq \infty$ (thanks to Central Limit Theorem)
- The standard deviation of an estimator calculated from batches or from single histories is an estimate of the standard deviation of the actual distribution ("error of the mean")
- How good is such an estimate depends on the type of estimator and on the particular problem (but it converges to the true value for $N \to \infty$)

Relative error	Quality of Tally *(from the MCNP Manual)*
50 to 100%	Garbage
20 to 50%	Factor of a few
10 to 20%	Questionable
< 10%	Generally reliable except for point detectors

Analog Monte Carlo:

In an analog Monte Carlo calculation ("honest" simulation), not only the mean of the contributions converges to the mean of the real distribution, but also the variance and all moments of higher order

$$\overline{\mu}^m = \int_x \int_y \int_z \ldots \int \left[A(x,y,z,\ldots) - \overline{A}\right]^m f'(x,y,z,\ldots) g'(x,y,z,\ldots) h'(x,y,z,\ldots) \, dx \, dy \, dz \ldots$$

converge as well:

$$\lim_{N\to\infty} \left[\frac{\sum_{i=1}^{N} (A_i - S_n)^m}{N} \right]^{\frac{1}{m}} = \overline{\mu}^m$$

Biased Monte Carlo approach

The Analog Monte Carlo

- samples from actual phase space distributions
- predicts average quantities and all statistical moments of any order
- preserves correlations and reproduces fluctuations (provided the physics is correct…)
- is *(almost)* safe and can *(sometimes)* be used as “black box”

BUT

- is inefficient and converges very slowly
- fails to predict important contributions due to rare events

Biased Monte Carlo:

- samples from artificial distributions and applies a weight to the particles to correct for the bias
- predicts average quantities, but not the higher moments *(on the contrary, its goal is to minimize the second moment)*
- same mean with smaller variance, *i.e.*, faster convergence

BUT

- cannot reproduce correlations and fluctuations
- requires physical judgment, experience and a good understanding of the problem (it is not a "black box"!)
- in general, a user does not get the definitive result after the first run, but needs to do a series of test runs in order to optimize the biasing parameters

→ **balance between user's time and CPU time**

Reduce variance or CPU time ?

A Figure of Merit

Computer cost of an estimator = σ^2 x t

(σ^2 = Variance, t =CPU time per primary particle)

- some biasing techniques are aiming at reducing σ, others at reducing t
- often reducing σ increases t, and *viceversa*
- therefore, minimizing $\sigma^2 x\, t$ means to reduce σ at a faster rate than t increases or *viceversa*
- the choice depends on the problem, and sometimes a combination of several techniques is most effective
- bad judgment, or excessive "forcing" on one of the two variables can have catastrophic consequences on the other one, making computer cost explode

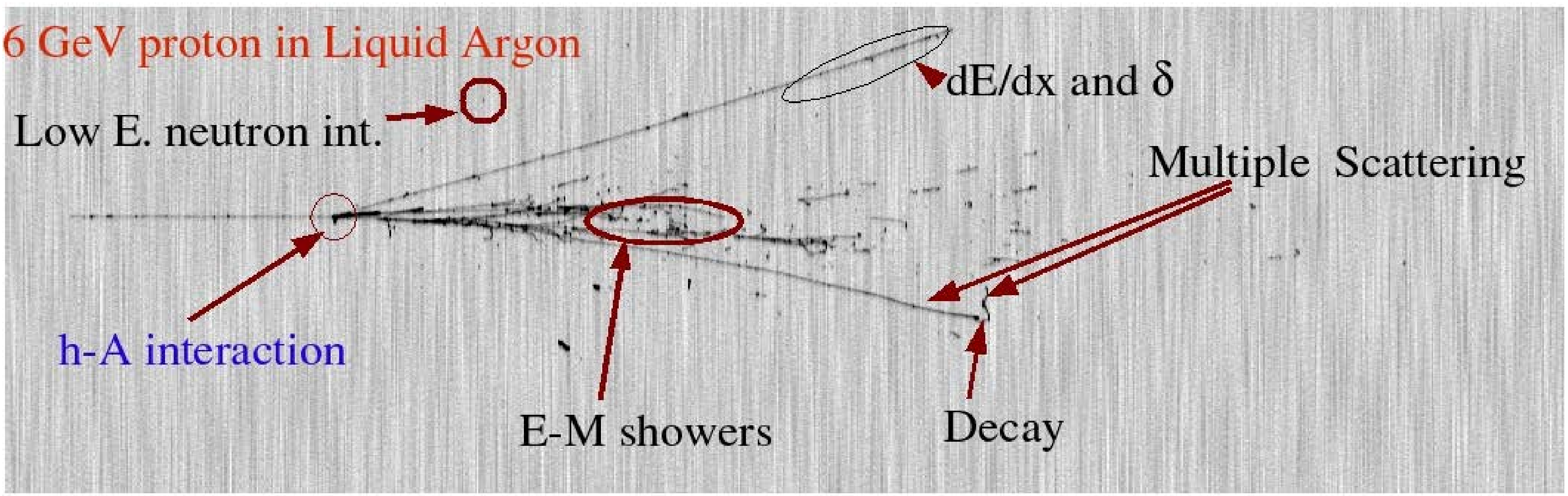
The FLUKA Code

An Introduction to FLUKA:
a multipurpose Interaction and Transport MC code

FLUKA

Main authors: A. Fassò, A. Ferrari, J. Ranft, P.R. Sala

Contributing authors: G. Battistoni, F. Cerutti, M. Chin,T. Empl, M.V. Garzelli, M. Lantz, A. Mairani, V. Patera, S. Roesler, G. Smirnov,



Developed and maintained under an INFN-CERN agreement Copyright 1989-2011 CERN and INFN

>4000 users

http://www.fluka.org

The FLUKA international Collaboration

V. Boccone, T. Boehlen, M. Brugger, F. Cerutti, M. Chin, Alfredo Ferrari, A. Mereghetti, S. Roesler, G. Smirnov, C. Theis, R. Versaci, Heinz Vincke, Helmut Vincke, V. Vlachoudis, J.Vollaire, CERN

A. Fassò, Jefferson Lab, USA

J. Ranft, Univ. of Siegen, Germany

G. Battistoni, F. Broggi, M. Campanella, F. Cappucci, E. Gadioli, S. Muraro, P.R. Sala, INFN & Univ. Milano, Italy

L. Sarchiapone, INFN Legnaro, Italy

G. Brunetti, A. Margiotta, M. Sioli, INFN & Univ. Bologna, Italy

V. Patera, INFN Frascati & Univ. Roma La Sapienza, Ital

. Pelliccioni, INFN Frascati & CNAO, Pavia, Italy

A. Mairani, CNAO Pavia, Italy

M. Santana, SLAC, USA

M.C. Morone, Univ. Roma II, Italy

. Parodi, F. Sommerer, HIT, Heidelberg, Germany

A. Empl, L. Pinsky, B. Reddell, Univ. of Houston, USA

K.T. Lee, T. Wilson, N. Zapp, NASA-Houston, USA

S. Rollet, AIT, Austria

M. Lantz, Uppsala Univ., Sweden

S. Trovati, PSI, Switzerland

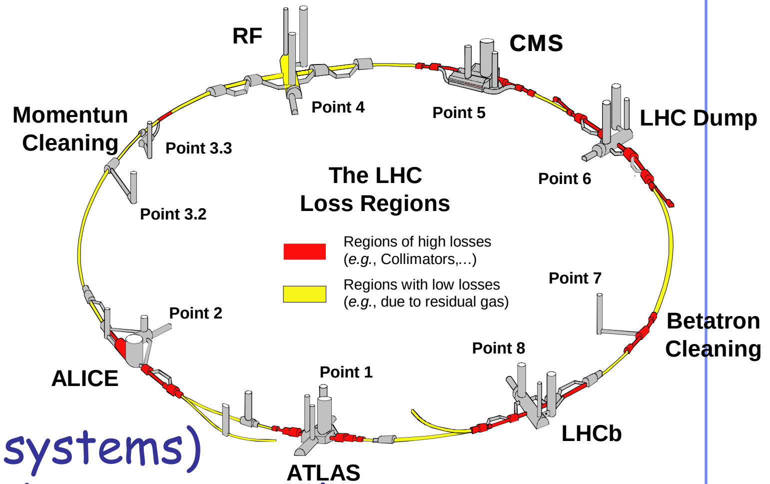
G. Lukasik, Poland

P. Colleoni, Bergamo, Italy

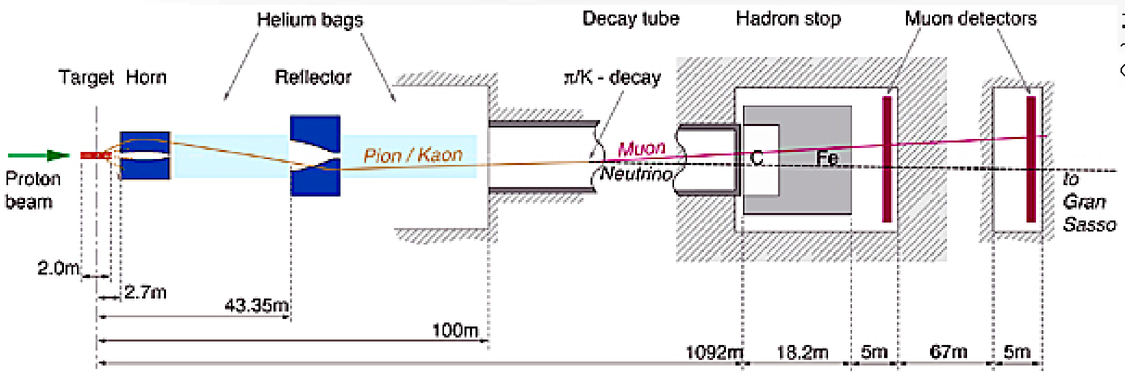
M.V. Garzelli, Granada Univ., Spain

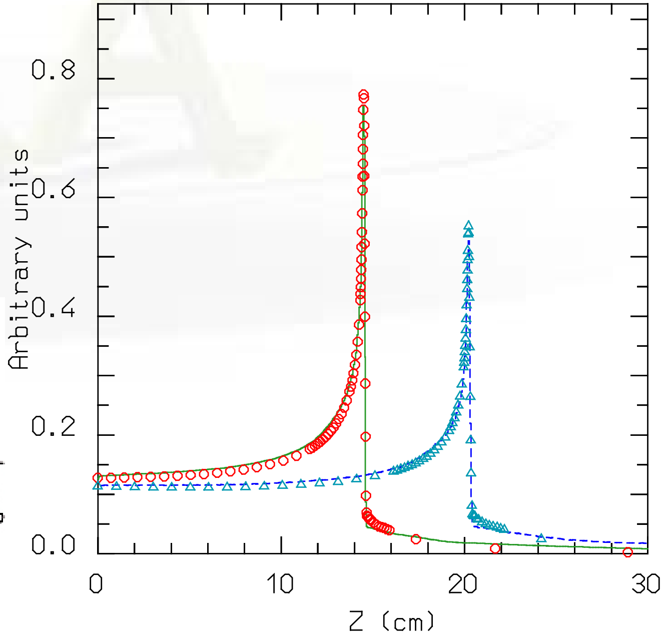
Anna Ferrari, FZR Rossendorf, Germany

FLUKA Applications



- Cosmic ray physics
- Neutrino physics
- Accelerator design (→ n_ToF, CNGS, LHC systems)
- Particle physics: calorimetry, tracking and detector simulation etc. (→ ALICE, ICARUS, ...)
- ADS systems, waste transmutation, (→"Energy amplifier", FEAT, TARC,...)
- Shielding design
- Dosimetry and radioprotection
- Space radiation
- Hadrontherapy
- Neutronics





The History

The early days

The beginning:

1962: Johannes Ranft (Leipzig) and Hans Geibel (CERN): Monte Carlo for high-energy proton beams

The name:

1970: study of event-by-event fluctuations in a NaI calorimeter (FLUktuierende KAskade)

Early 70's to ≈1987: J. Ranft and coworkers (Leipzig University) with contributions from Helsinki University of Technology (J. Routti, P. Aarnio) and CERN (G.R. Stevenson, A. Fassò)

Link with EGS4 in 1986, later abandoned

The modern code: some dates

Since 1989: mostly INFN Milan (A. Ferrari, P.R. Sala): little or no remnants of older versions. Link with the past: J. Ranft and A. Fassò

1990: LAHET / MCNPX: high-energy hadronic FLUKA generator *No further update*

1993: G-FLUKA (the FLUKA hadronic package in GEANT3). *No further update*

1998: FLUGG, interface to GEANT4 geometry

2000: grant from NASA to develop heavy ion interactions and transport

2001: the INFN FLUKA Project

2003: official CERN-INFN collaboration to develop, maintain and distribute FLUKA

The FLUKA Code design - 1

- Sound and updated physics models
  - Based, as far as possible, on original and well-tested microscopic models
  - Optimized by comparing with experimental data at single interaction level: *"theory driven, benchmarked with data"*
  - Final predictions obtained with minimal free parameters fixed for all energies, targets and projectiles
  - Basic conservation laws fulfilled "a priori"

    ➔ *Results in complex cases, as well as properties and scaling laws*, arise naturally from the underlying physical models

    ➔ *Predictivity where no experimental data are directly available*

It is a "condensed history" MC code, with the possibility use of single instead of multiple scattering

The FLUKA Code design - 2

- Self-consistency
  - Full cross-talk between all components: hadronic, electromagnetic, neutrons, muons, heavy ions
  - Effort to achieve the same level of accuracy:
    - for each component
    - for all energies

➔ Correlations preserved fully within interactions and among shower components

➔ ***FLUKA is NOT a toolkit! Its physical models are fully integrated***

The Physics Content of FLUKA

- 60 different particles + Heavy Ions
  - Nucleus-nucleus interactions from Coulomb barrier up to 10000 TeV/n
  - Electromagnetic and μ interactions 1 keV – 10000 TeV
  - Hadron-hadron and hadron-nucleus interactions 0–10000 TeV
  - Neutrino interactions
  - Charged particle transport including all relevant processes
  - Transport in magnetic fields
  - Neutron multigroup transport and interactions 0 – 20 MeV
  - *Analog calculations, or with variance reduction*

INSTALLING FLUKA

How to download and install FLUKA

Two ways of downloading the FLUKA software:

- From the FLUKA website http://www.fluka.org
- From NEA databank http://www.nea.fr through the liaison officer from your institute

It is mandatory to be registered as FLUKA user. Follow the link:

http://www.fluka.org/download.html

After registration (or using your user-id and password) you can proceed in downloading the latest official release version.

How to download and install FLUKA

First identify the location of the FLUKA distribution file:
fluka2011.2-linuxAA.tar.gz

The user will create a directory **FLUKA** (or any other name) and there will expand the tar file. Example:

```
mkdir FLUKA      # creates a directory called FLUKA
cd FLUKA         # changes to the FLUKA directory
tar xzvf "some path" /fluka2011.2-linuxAA.tar.gz
                 # expands the FLUKA package
```

Alternatively a RPM file is available for linux RedHat like installation

There is also the image to allow the use of FLUKA by means of a Virtual Machine

Persistent settings

A **FLUPRO** environment variable, pointing to the FLUKA directory must be defined

Example for **bash users:**

```
export FLUPRO=${HOME}/FLUKA
```

In case of rpm installation, FLUKA will be installed in **/usr/local**
and the FLUPRO variable will be automaticcaly defined

FLUKA release: main directory $FLUPRO

Main Library:

libflukahp.a (object collection)

Physics data files:

sigmapi.bin
elasct.bin
brems_fin.bin
cohff.bin
gxsect.bin
neuxsc-ind_260.bin
nuclear.bin
fluodt.dat
e6r1nds3.fyi
jef2.fyi
jendl3.fyi
xnloan.dat
Fad/*
DDS/*

Basic Scripts: (in $FLUPRO/flutil)

rfluka
lfluka
fff

Random Number seed

random.dat

Important Directories

flukapro/	all FLUKA commons
usermvax/	user routines
flutil/	general utilities

Available Documentation

- fluka2011.manual ASCII version of the manual (easy to edit)
- FM.pdf current version of the FLUKA manual
- CERN-2005-10.pdf official reference for FLUKA

- or navigate the manual, online version (www.fluka.org)

- or (when using FLAIR) press F1 to get an interactive manual (which can be also called on prompt level by calling '*fm.py*')

- or (at a further stage) the FAQ available at: http://www.fluka.org/fluka.php?id=faq&mm2=3
- or (at a further stage) the archive of fluka-discuss: http://www.fluka.org/MailingList.html
- Release notes

A glimpse of FLUKA

The FLUKA version

FLUKA20xx.n(y)(.m)

- 20xx: Major version
- .n: Minor version
- (y): Patch level
- (.m): Respin

Since 2006 each version is going to be maintained for 2 years max.

we are now distributing FLUKA2011.2.3

The FLUKA license (it is not GPL):

- Standard download: binary library + user routines.
  - FLUKA can be used freely for scientific and academic purposes, ad-hoc agreement for commercial purposes
  - It cannot be used for weapon related applications
  - It is not permitted to redistribute the code (single user, single site)
  - User can add their own scoring, sources etc through a wide set of user routines, provided they don't modify the physics
  - Relevant references for each FLUKA version can be found in the documentation
- It is possible, by explicit signature of license, to download the source for researchers of scientific/academic Institutions. *(!!! now from NEA as well !!!)*
  - FLUKA cannot be copied, even in part, into other codes, or translated into another language without permission.
  - The user cannot publish results with modified code, unless explicit authorization is granted in advance.

Using FLUKA

Platform: Linux with g77

Under test: Linux and Mac OSX (gfortran), Windows-Cygwin (g95)

The code can be compiled/run only on with operating systems, compilers (and associated) options tested and approved by the development team

Standard Input:

- Command/options driven by "data cards" (ascii file) .
  Graphical interface is available!!!!

- Standard Geometry ("Combinatorial geometry"): input by "data cards"

Standard Output and Scoring:

- Apparently limited but highly flexible and powerful
- Output processing and plotting interface available

FLUKA Description

- FLUKA is a general purpose tool for calculations of particle transport and interactions with matter, covering an extended range of applications: from proton and electron accelerator shielding to target design, calorimetry, activation, dosimetry, detector design, Accelerator Driven Systems, cosmic rays, neutrino physics, radiotherapy etc.
- 60 different particles + Heavy Ions
  - Hadron-hadron and hadron-nucleus interaction "0"-10000 TeV
  - Electromagnetic and μ interactions 1 keV – 10000 TeV
  - Nucleus-nucleus interaction up to 10000 TeV/n
  - Charged particle transport and energy loss
  - Neutron multi-group transport and interactions 0-20 MeV
  - n interactions
  - Transport in magnetic field
  - Combinatorial (boolean) and Voxel geometries
  - Double capability to run either fully analogue and/or biased calculations
  - On-line evolution of induced radioactivity and dose
  - User-friendly GUI interface thanks to the Flair interface

Full mixed field capability

- Maintained and developed under CERN-INFN agreement and copyright 1989-2011
- More than 4000 users all over the world

http://www.fluka.org

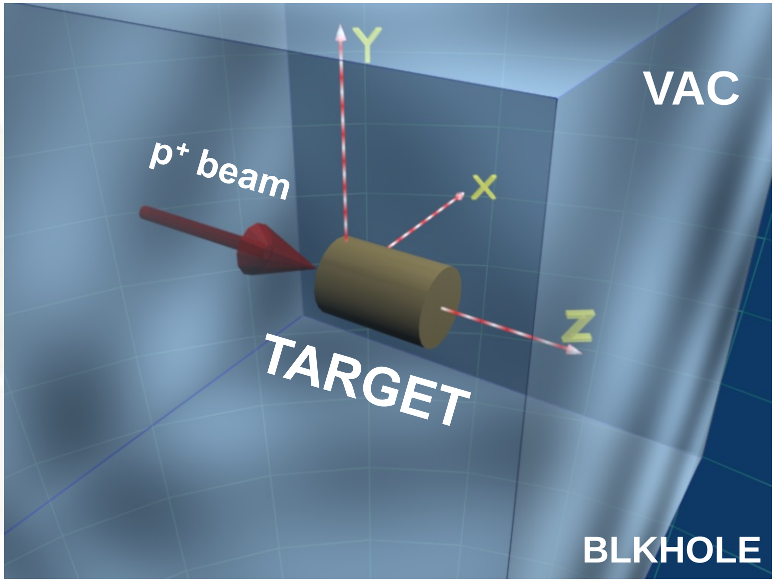
A Simple Example of basic input

```
TITLE
FLUKA Course Exercise
*...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....*
DEFAULTS                                                                NEW-DEFA
BEAM             -3.5 -0.082425      -1.7       0.0       0.0       1.0PROTON
BEAMPOS           0.0       0.0       0.1       0.0       0.0       0.0
*...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....*
GEOBEGIN                                                                COMBNAME
    0    0                    Cylindrical Target
SPH BLK 0.0  0.0  0.0  10000.
* vacuum box
RPP VOI -1000. 1000. -1000. 1000. -1000. 1000.
* Lead target
RCC TARG0.0 0.0 0.0 0.0 0.0 10. 5.
END
* Regions
* Black Hole
BLKHOLE 5   +BLK -VOI
* Void around
VAC     5   +VOI -TARG
* Target
TARGET  5   +TARG
END
GEOEND
*...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....*
ASSIGNMA    BLCKHOLE   BLKHOLE
ASSIGNMA      VACUUM       VAC
ASSIGNMA        LEAD    TARGET
*...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....*
RANDOMIZ          1.0
START            10.0       0.0
STOP
```

Primary beam

Geometry

Assignin materials



FLUKA Scoring & Results - Estimators

- It is often said that Monte Carlo (MC) is a "mathematical experiment" The MC equivalent of the result of a real experiment (*i.e.*, of a measurement) is called an estimator.
- Just as a real measurement, an estimator is obtained by sampling from a statistical distribution and has a statistical error (and in general also a systematic one).
- There are often several different techniques to measure the same physical quantity: in the same way the same quantity can be calculated using different kinds of estimators.
- FLUKA offers numerous different estimators, *i.e.*, directly from the input file the users can request scoring the respective quantities they are interested in.
- As the latter is implemented in a very complete way, users are strongly encouraged to preferably use the built-in estimators with respect to user-defined scoring
- For additional requirements FLUKA user routines are provided

Built-In and User Scoring

- Several pre-defined estimators can be activated in FLUKA.
- One usually refers to these estimators as “scoring” capabilities
- Users have also the possibility to build their own scoring through user routines, HOWEVER:
  - Built-in scoring covers most of the **common needs**
  - Built-in scoring has been **extensively tested**
  - Built-in scoring takes BIASING **weights automatically into account**
  - Built-in scoring has **refined algorithms** for track subdivision
  - Built-in scoring comes with **utility programs** that allow to evaluate statistical errors
- Scoring can be geometry dependent AND/OR geometry independent FLUKA can score particle fluences, current, track length, energy spectra, Z spectra, energy deposition...
- Either integrated over the “run”, with proper normalization, OR event-by event
- Standard scoring can be weighted by means of simple user routines

Related Scoring Commands (main cases)

- USRTRACK, USRCOLL score average $d\Phi/dE$ (differential fluence) of a given type or family of particles in a given region
- USRBDX scores average $d^2\Phi/dEd\Omega$ (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
- USRYIELD scores a double differential yield of particles escaping from a surface. The distribution can be with respect to energy and angle, but also other more "exotic" quantities
- SCORE scores energy deposited (or star density) in all regions
- The output of SCORE will be printed in the main (standard) output, written on logical output unit LUNOUT (pre-defined as **11** by default)
- All other detectors write their results into logical output units assigned by the user (the unit numbers must be **>20**)

USRBIN

** energy deposition

```
USRBIN      11.0     ENERGY     -40.0      10.0           15.0 TargEne
USRBIN       0.0                 -5.0     100.0          200.0 &
```

- This is an R-Z-Φ binning (what(1)=11), scoring energy deposition (generalized particle ENERGY, or 208), writing the unformatted output on unit 40, spanning 0<R<10 in 100 bins, $0<\Phi<2\pi$ in 1 bin (default), -5<z<15 in 200 bins.

** neutron fluence

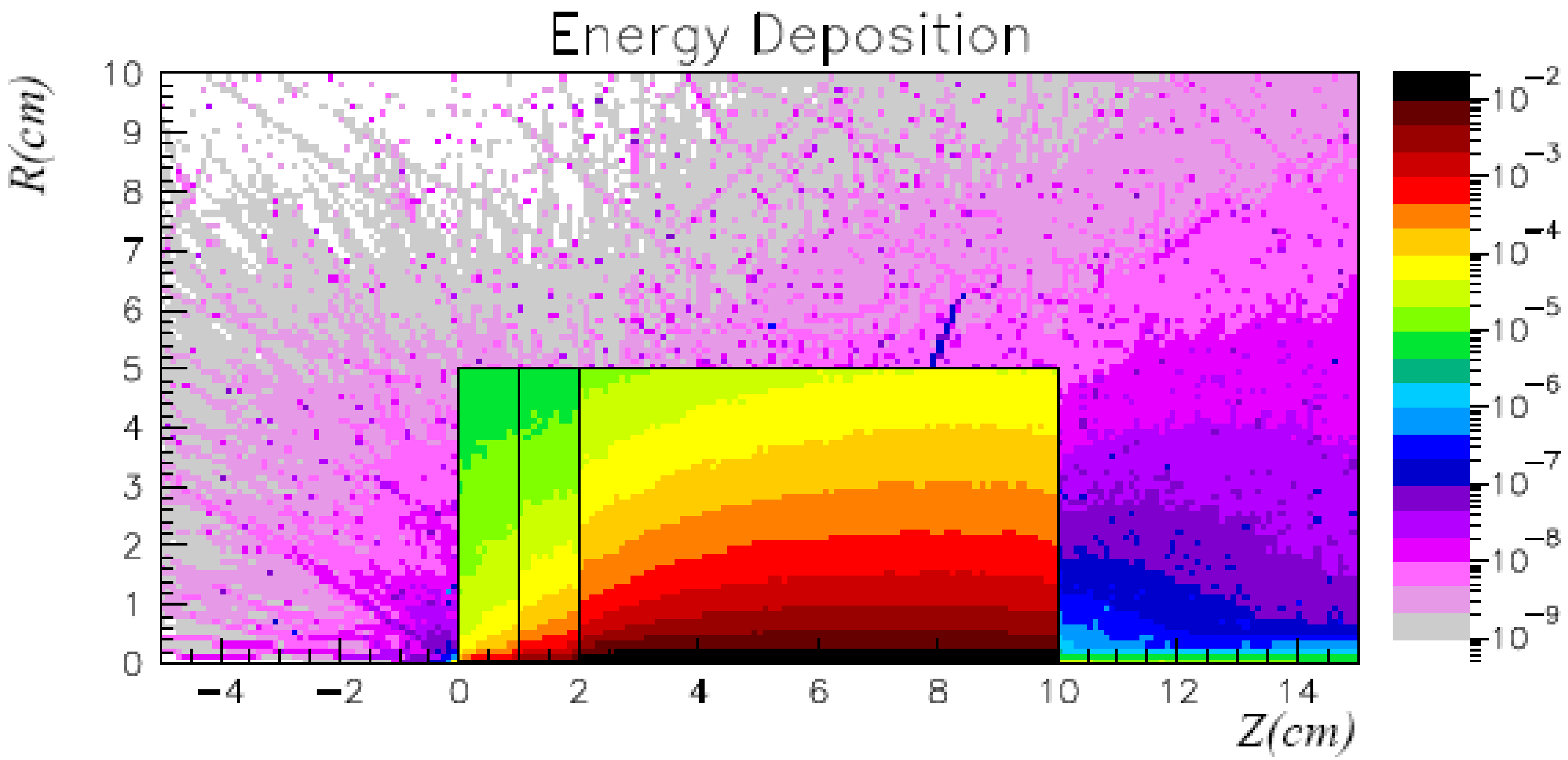
```
*            R-Z  EM energy  output unit    Rmax    axis Y    Zmax
*           Rmin     axis X         Zmin  # R-bins # Phi-bins # Z-bins
USRBIN      11.0    NEUTRON     -40.0      10.0           15.0 TargNeu
USRBIN       0.0                 -5.0     100.0          200.0 &
```

- This is a R-Z-Φ binning (what(1)=11), scoring neutron fluence, writing the unformatted output on unit 40, spanning 0<R<10 in 100 bins, $0<\Phi<2\pi$ in 1 bin (default), -5<z<15 in 200 bins.

USRBIN → The Result

WHAT(2) = ENERGY :Energy deposition from a 3.5 GeV proton beam hitting at [0.,0.,0.] directed along z

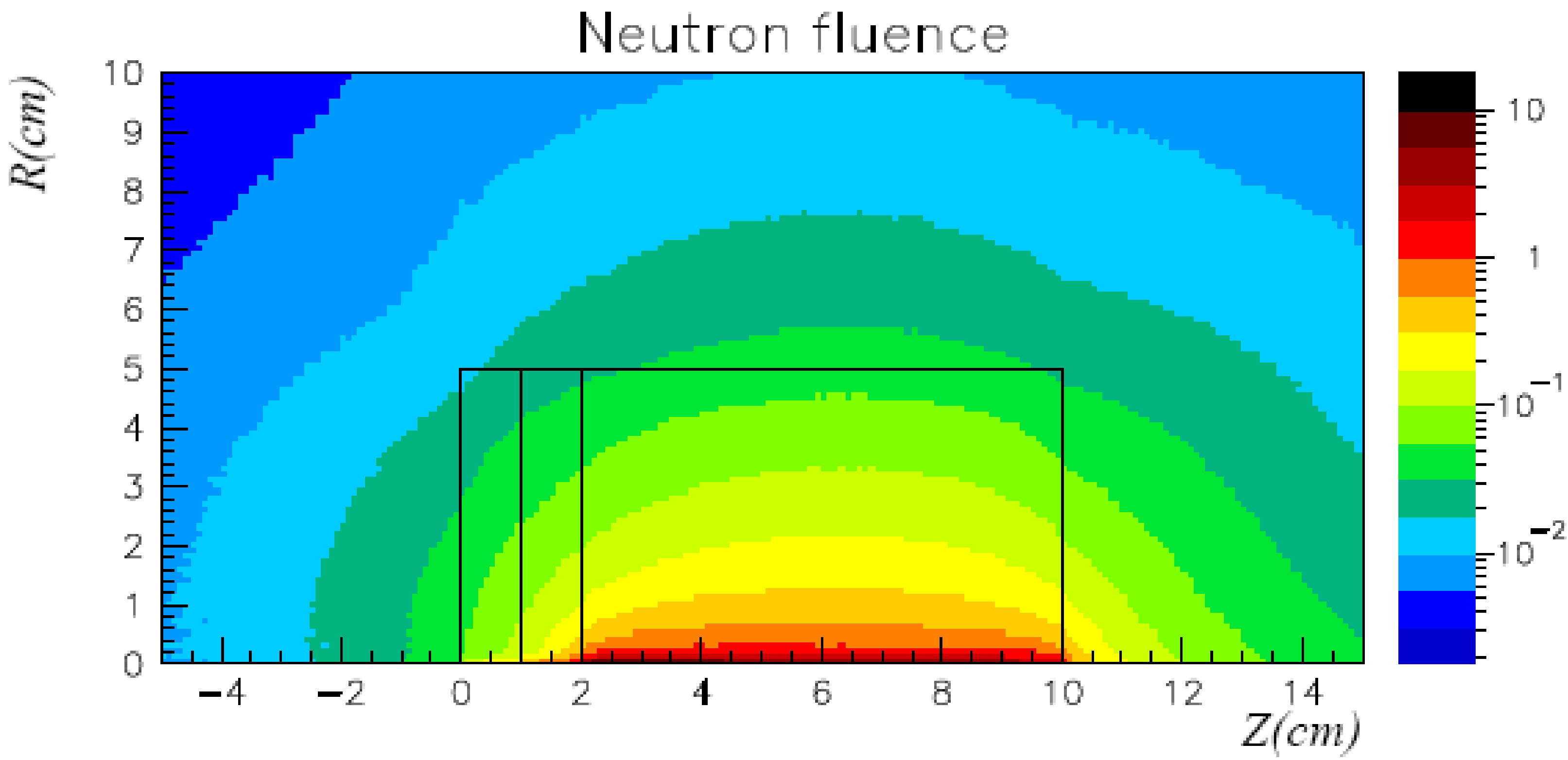
results are normalized to GeV/cm³ per primary



USRBIN → The Result

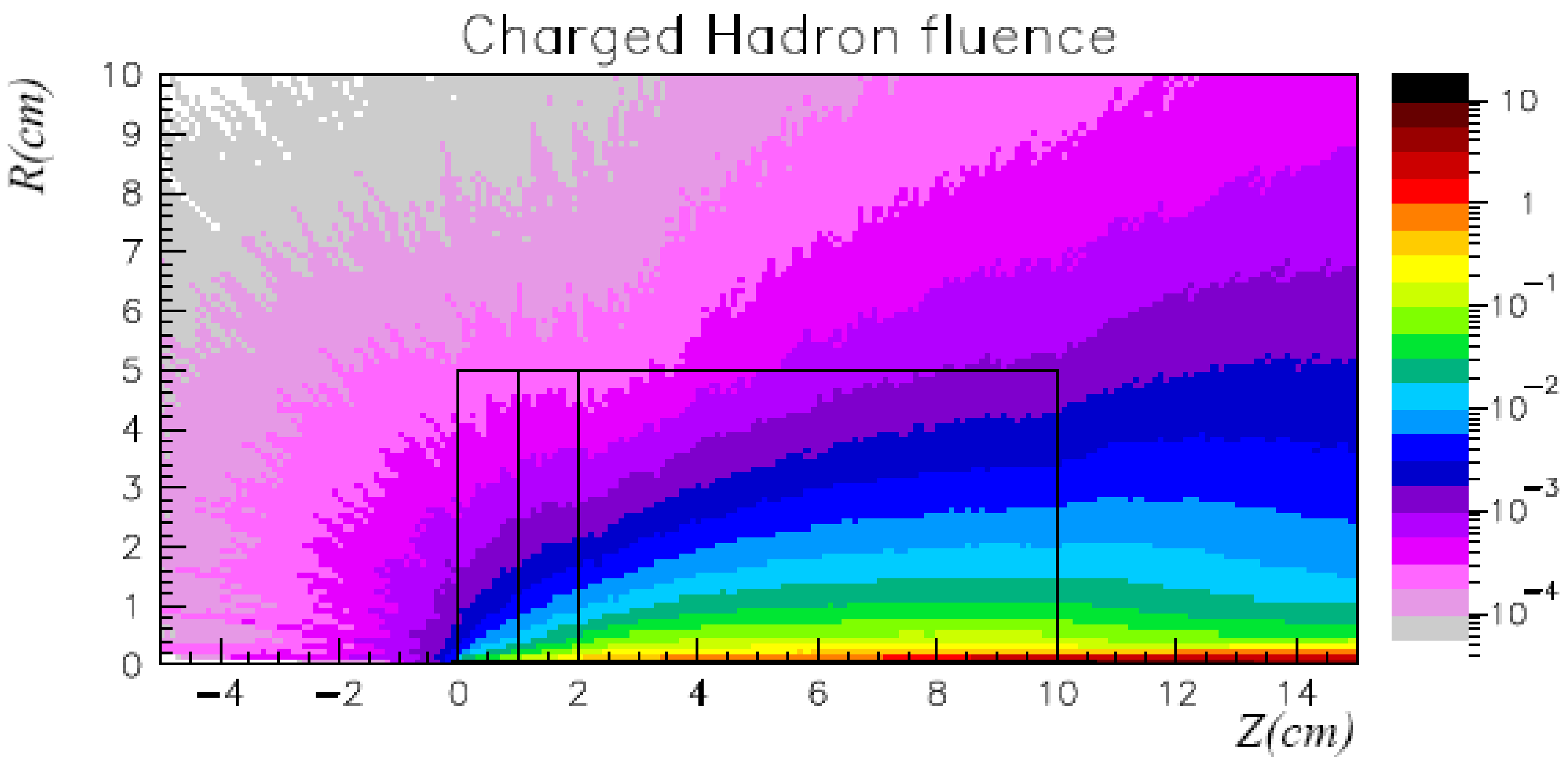
Same, WHAT(2)= NEUTRON to get neutron fluence

results are normalized to particles/cm^2 per primary



USRBIN → The Result

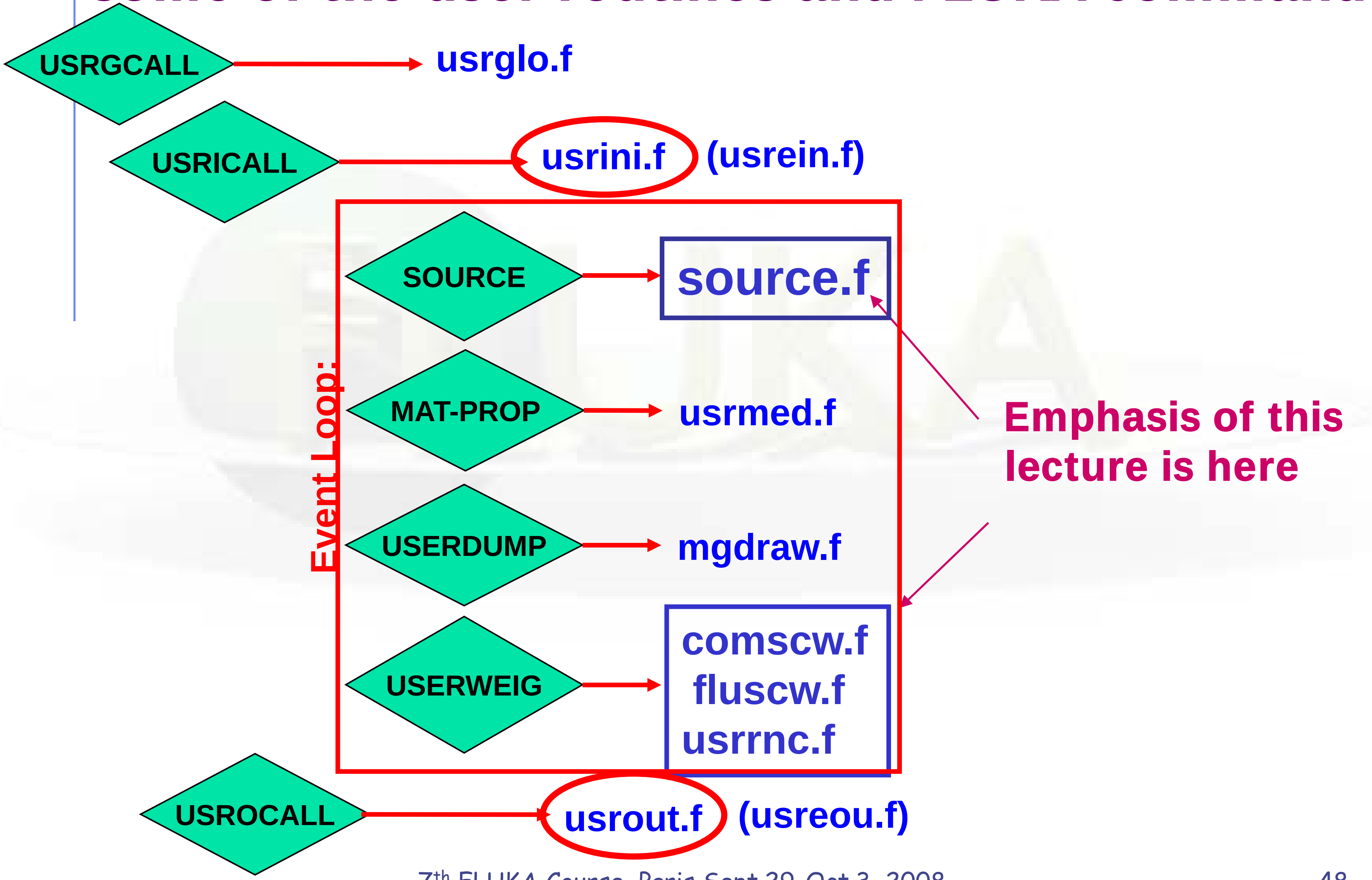
Same, WHAT(2)= HAD-CHAR to get charged hadron fluence
results are normalized to particles/cm^2 per primary



User Routines

- Fluka offers a rich choice of options for scoring most quantities and for applying variance reduction techniques, without requiring the users to write a single line of code.
- However there are special cases where “ad-hoc” routines are unavoidable, because the required information cannot be obtained through standard options.
- A number of template of user routines (available in the usermvax directory) can be modified/activated by the user allow to fulfill non-standard tasks

A first look at the correspondence between some of the user routines and FLUKA commands



AN INTRODUCTION TO THE FLUKA COMBINATORIAL GEOMETRY

Introduction

Principle of Combinatorial Geometry: Basic convex shapes (bodies) such as cylinders, spheres, parallelepipeds, etc. are combined to more complex shapes called regions. This combination is done by the boolean operations union, intersection and subtraction.

The Combinatorial Geometry of FLUKA was initially similar to the package developed at ORNL for the neutron and gamma-ray transport program Morse (M.B. Emmett ORNL-4972 1975) which was based on the original combinatorial geometry by MAGI (Mathematical Applications Group, Inc., W. Guber et al, MAGI-6701 1967).

Basic Concepts

Four concepts are fundamental in the FLUKA CG:

- Bodies - basic convex objects, plus infinite planes, infinite cylinders and generic quadric surfaces
- Zones - sub-regions defined only with intersection and subtraction of bodies
- Regions - defined as boolean operations of bodies (union of zones)
- Lattices - duplication of existing objects (translated & rotated), will be explained in a separate lecture

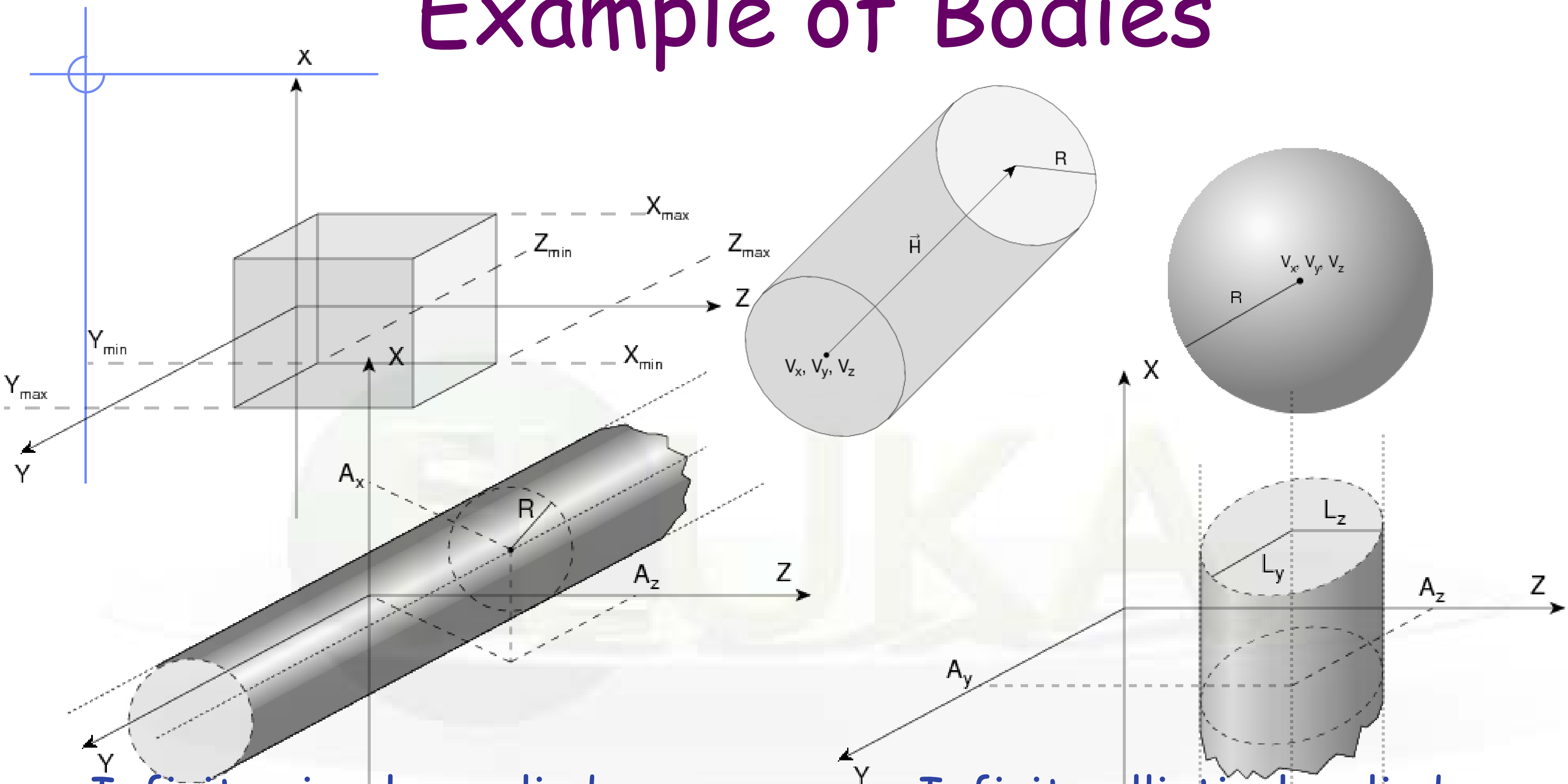
In the original description (Morse) bodies were convex solid bodies (finite portions of space completely delimited by surfaces of first or second degree, i.e. planes or quadrics). In FLUKA, the definition has been extended to include infinite cylinders (circular and elliptical), planes (half-spaces), and generic quadrics (surfaces described by 2nd degree equations)

Use of such “infinite bodies” is encouraged since it makes input less error-prone. They also provide a more accurate and faster tracking.

Bodies

- Each body divides the space into two domains inside and outside. The outside part is pointed to by the normal to the surface.
- 3-character code of available bodies:
  - RPP: Rectangular ParallelePiped
  - SPH: SPHere
  - XYP, XZP, YZP: Infinite half space delimited by a coordinate plane
  - PLA: Generic infinite half-space, delimited by a PLAne
  - XCC, YCC, ZCC: Infinite Circular Cylinder, parallel to coordinate axis
  - XEC, YEC, ZEC: Infinite Elliptical Cylinder, parallel to coordinate axis
  - RCC: Right Circular Cylinder
  - REC: Right Elliptical Cylinder
  - TRC: Truncated Right angle Cone
  - ELL: ELLipsoid of revolution
  - QUA: QUAdric

Example of Bodies



Infinite circular cylinder

Infinite elliptical cylinder

Infinite half-space parallel to coordinate axis

Arbitrarily oriented infinite half-space

Arbitrary generic quadric: corresponding to the equation:

$$A_{xx}\, x^2 + A_{yy}\, y^2 + A_{zz}\, z^2 + A_{xy}\, xy + A_{xz}\, xz + A_{yz}\, yz + A_x\, x + A_y\, y + A_z\, x + A_0 = 0$$

The Black Hole

To avoid infinite tracking the particles must be stopped somewhere. This has to be insured by the user by defining a region surrounding the geometry and assigning the material BLCKHOLE to it.

The outer surface of this region must be defined by a single closed body (generally an RPP or a Sphere).

All particles that enter the blackhole are absorbed (they disappear). Further blackhole regions can be defined by the user if necessary.

The blackhole is the outermost boundary of the geometry. Inside its outer surface:

Each point of space must belong to one and only one region!

Combinatorial Geometry Input

CG input must respect the following sequential order:

- GEOBEGIN card
  - VOXELS card (optional, see Voxel lecture)
  - Geometry title (and reading format options)
    - Body data
  - END card (not needed in flair)
    - Region data
  - END card (not needed in flair)
  - LATTICE cards (optional, see Lattice lecture)
  - Region volumes

    (optionally requested by a flag in the Geometry title, used together with the SCORE command)
- GEOEND card

Cards having a * in column 1 are treated as comments.

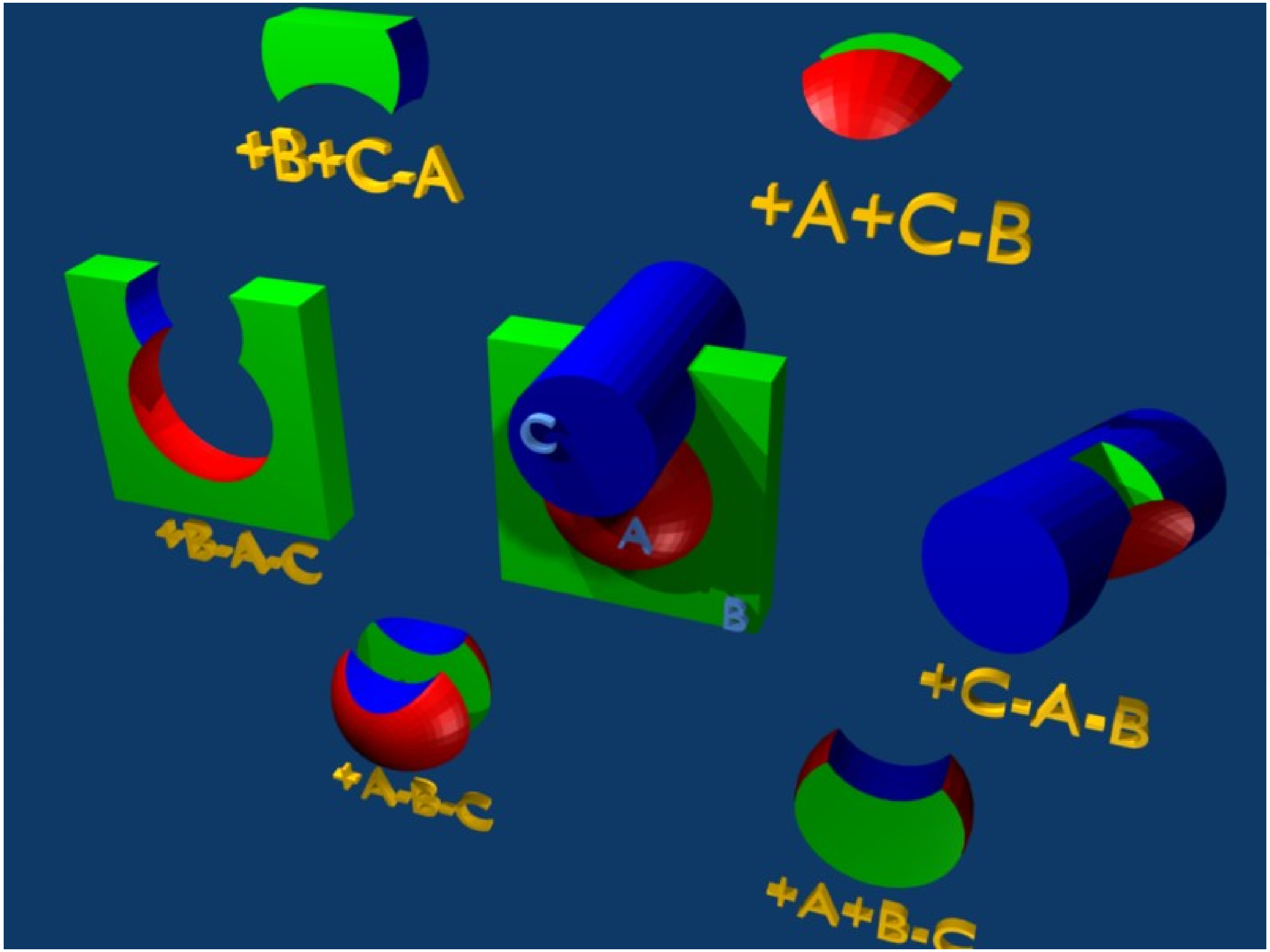
Concept of Region

Regions are defined as combinations of bodies obtained by boolean operations:

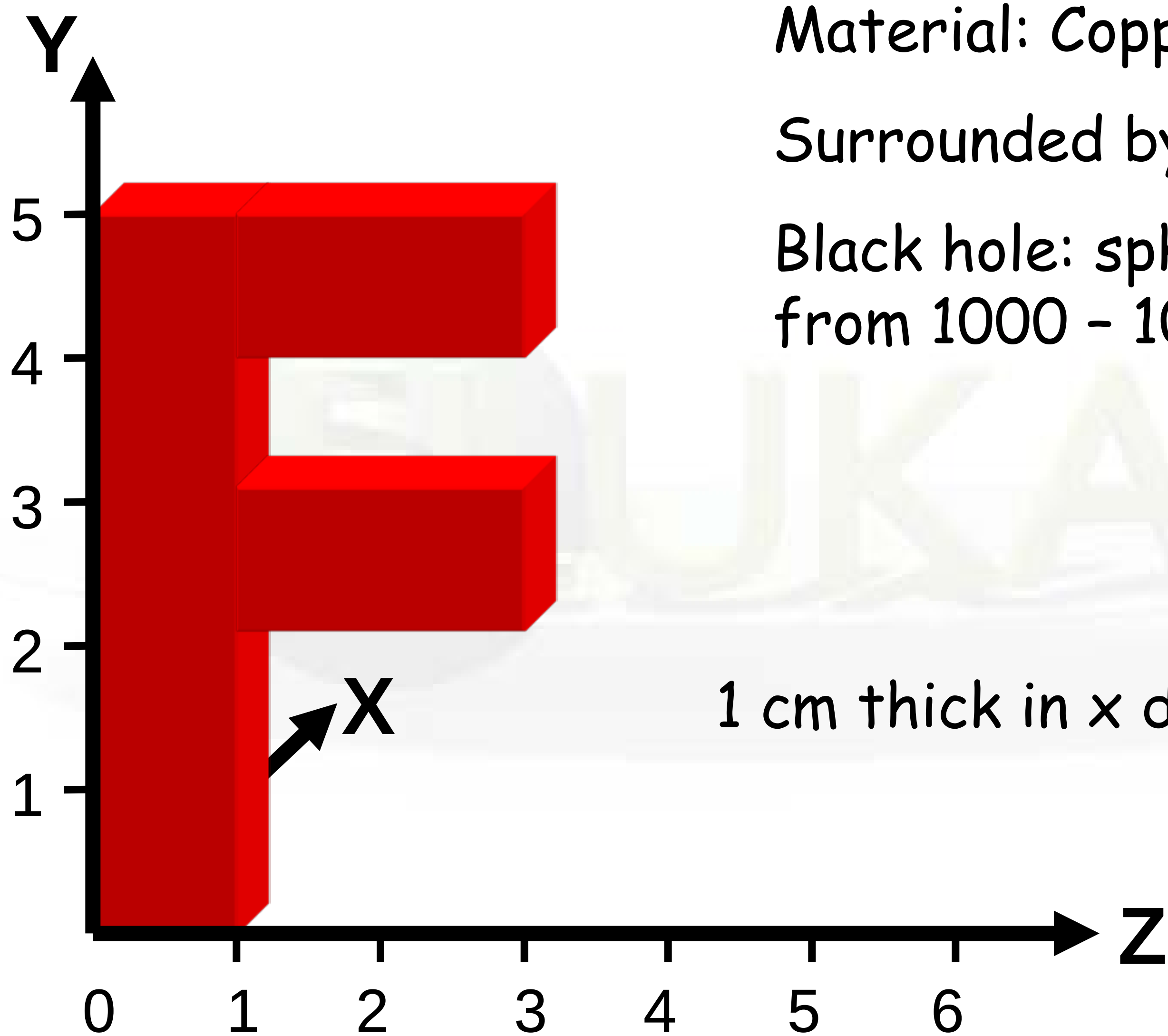
	Union	Subtraction	Intersection
Free Format	|	–	+
Fixed format	OR	–	+
Mathematically	$\cup$	-	$\cap$

Regions are not necessarily simply connected (they can be made as the union of two or more non contiguous or partially overlapping zones) but must be of homogeneous material composition.

Illustration of the + and – operators



Geometry Example "F" shaped target



Material: Copper

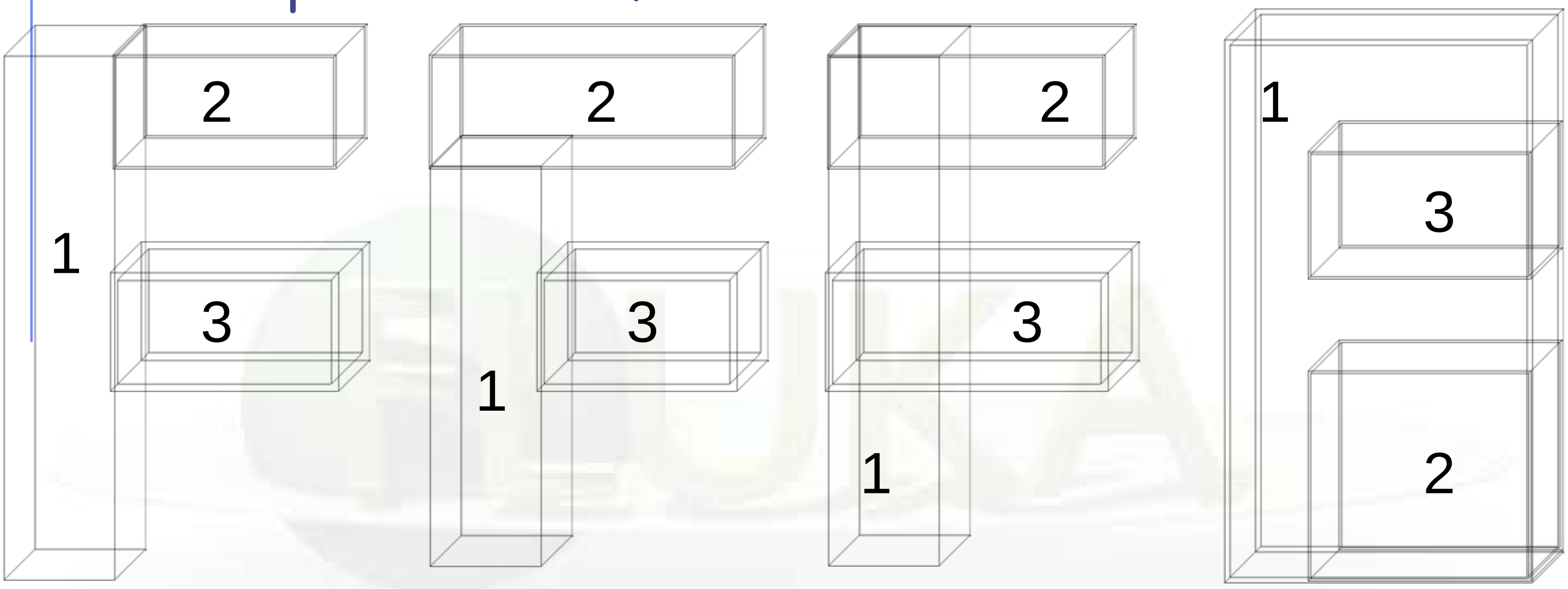
Surrounded by vacuum

Black hole: spherical shell from 1000 - 10000 cm

1 cm thick in x direction

Geometry example "F": Bodies

Several possibilities for bodies:



(A) 3 bodies (B) 3 bodies (C) overlapping (D) subtraction

If we use the 3 bodies to make a single region, the 4 options are equivalent. We will use C.

If 1, 2 and 3 are separate regions, avoid A and B: there are surfaces shared by different regions

Export to POVray

POV ray is a ray tracing program used for **3D** visualization or movies

Info: http://www.povray.org

Auxiliary program: Simple Geo

SimpleGeo is an interactive solid modeler which allows for flexible and easy creation of the models via drag & drop, as well as on-the-fly inspection

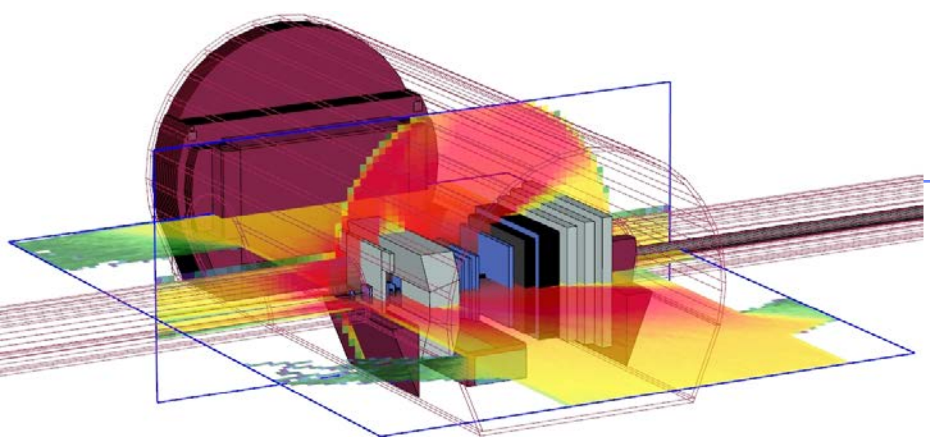
Imports existing geometries for viewing

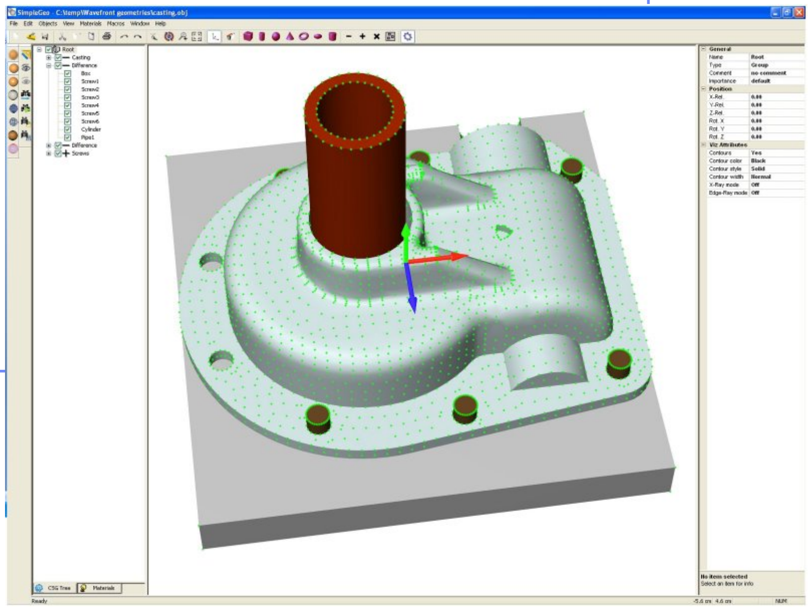
Creating new geometries from scratch

Export to various formats (FLUKA, MCNP, MCNPX)

Download, Tutorials, etc.: http://theis.web.cern.ch/theis/simplegeo

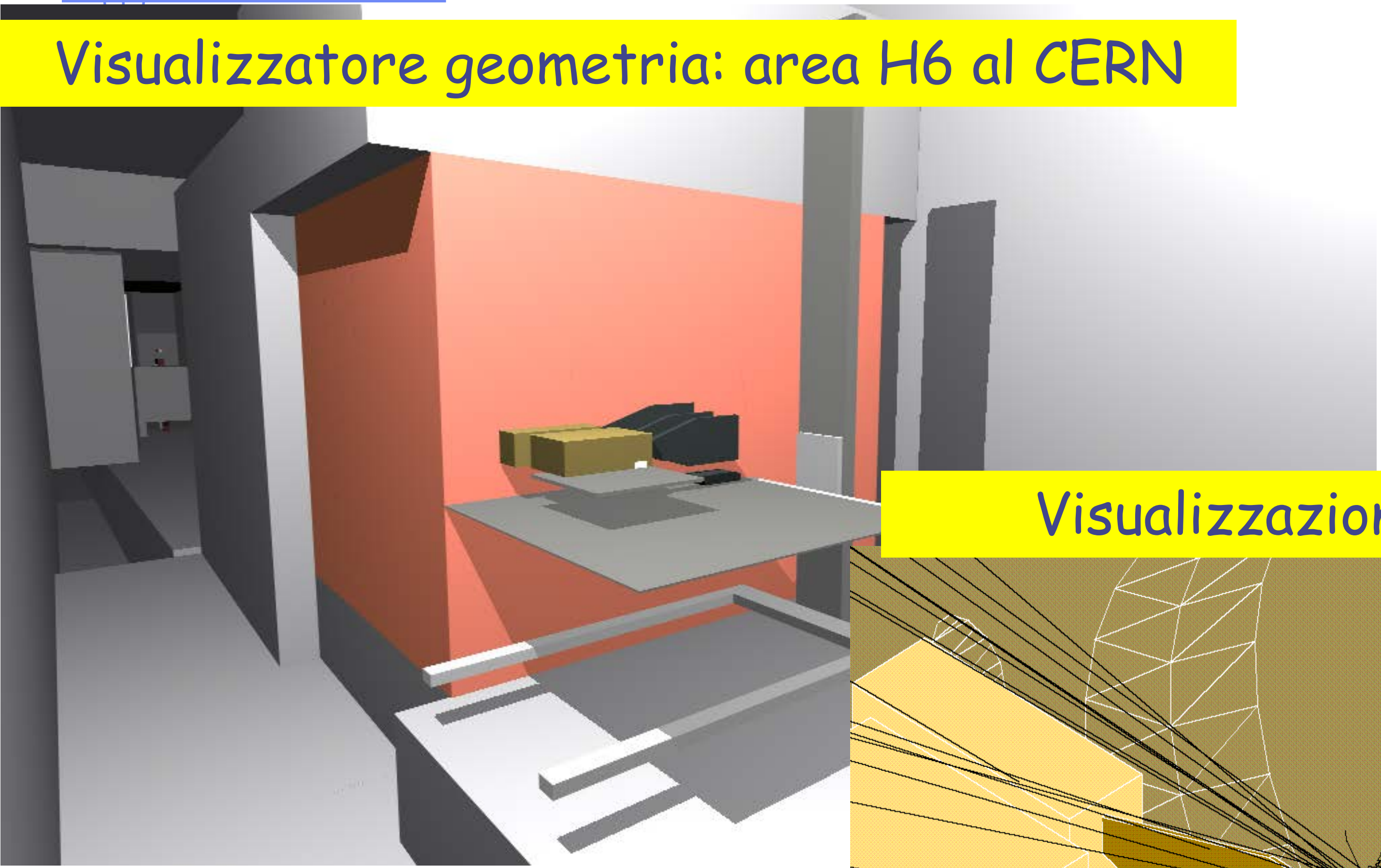
Operating system: Windows only



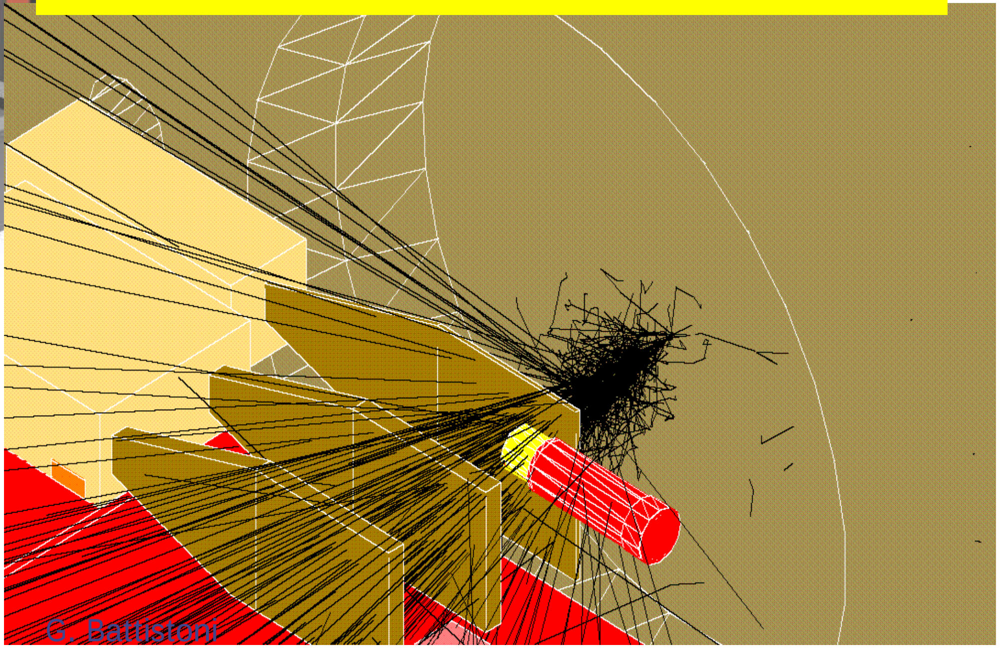


FLUKA + AutoCad

Visualizzatore geometria: area H6 al CERN



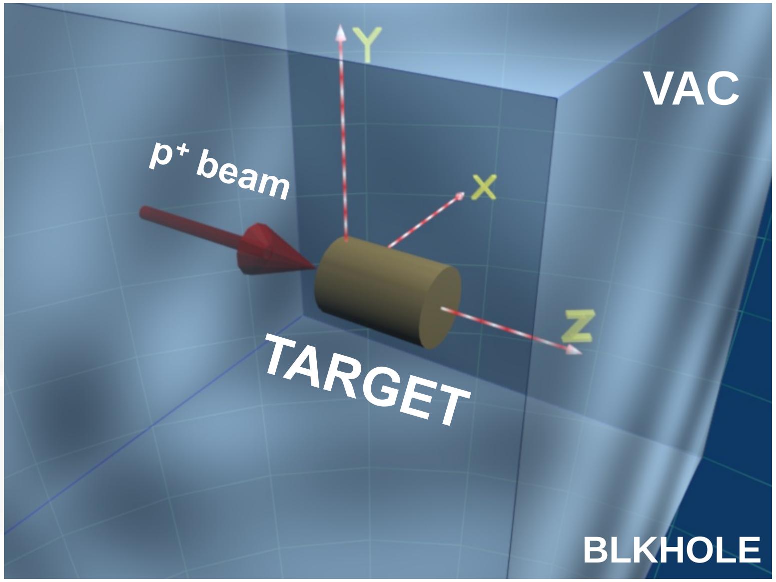
Visualizzazione eventi



A Simple Example

Geometry

```
TITLE
FLUKA Course Exercise
*...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....*
DEFAULTS                                                            NEW-DEFA
BEAM            -3.5 -0.082425      -1.7       0.0       0.0       1.0PROTON
BEAMPOS          0.0       0.0       0.1       0.0       0.0       0.0
*...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....*
GEOBEGIN                                                            COMBNAME
    0    0                  Cylindrical Target
SPH BLK 0.0  0.0  0.0  10000.
* vacuum box
RPP VOI -1000. 1000. -1000. 1000. -1000. 1000.
* Lead target
RCC TARG0.0 0.0 0.0 0.0 0.0 10. 5.
END
* Regions
* Black Hole
BLKHOLE 5   +BLK -VOI
* Void around
VAC     5   +VOI -TARG
* Target
TARGET  5   +TARG
END
GEOEND
*...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....*
ASSIGNMA    BLCKHOLE   BLKHOLE
ASSIGNMA      VACUUM       VAC
ASSIGNMA        LEAD    TARGET
*...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....*
RANDOMIZ         1.0
START           10.0       0.0
STOP
```



SOMETHING ABOUT THE PHYSICS CONTENT OF FLUKA

The FLUKA hadronic model(s)

Hadron-Hadron

Elastic, exchange Phase shifts data, eikonal	P<3-5GeV/c Resonance prod and decay	low E π, K Special	High Energy DPM hadronization

Hadron-Nucleus

PEANUT
Sophisticated GINC
Gradual onset of Glauber-Gribov multiple interactions
Preequilibrium
Coalescence

Nucleus-Nucleus

E< 0.1GeV/u	0.1< E< 5 GeV/u	E> 5 GeV/u
BME Complete fusion+ peripheral	rQMD-2.4 modified new QMD	DPMJET DPM+ Glauber+ GINC

Evaporation/Fission/Fermi break-up
γ deexcitation

The Nuclear environment: PEANUT

PreEquilibrium Approach to NUclear Thermalization

- PEANUT handles hadron-nucleus interactions from threshold (or 20 MeV neutrons) ~~up to 5 GeV~~ NEW

Sophisticated Generalized IntraNuclear Cascade

Smooth transition (all non-nucleons emitted/absorbed/decayed + all secondaries below 30-50 MeV)

↓

Prequilibrium stage

Standard Assumption on exciton number or excitation energy

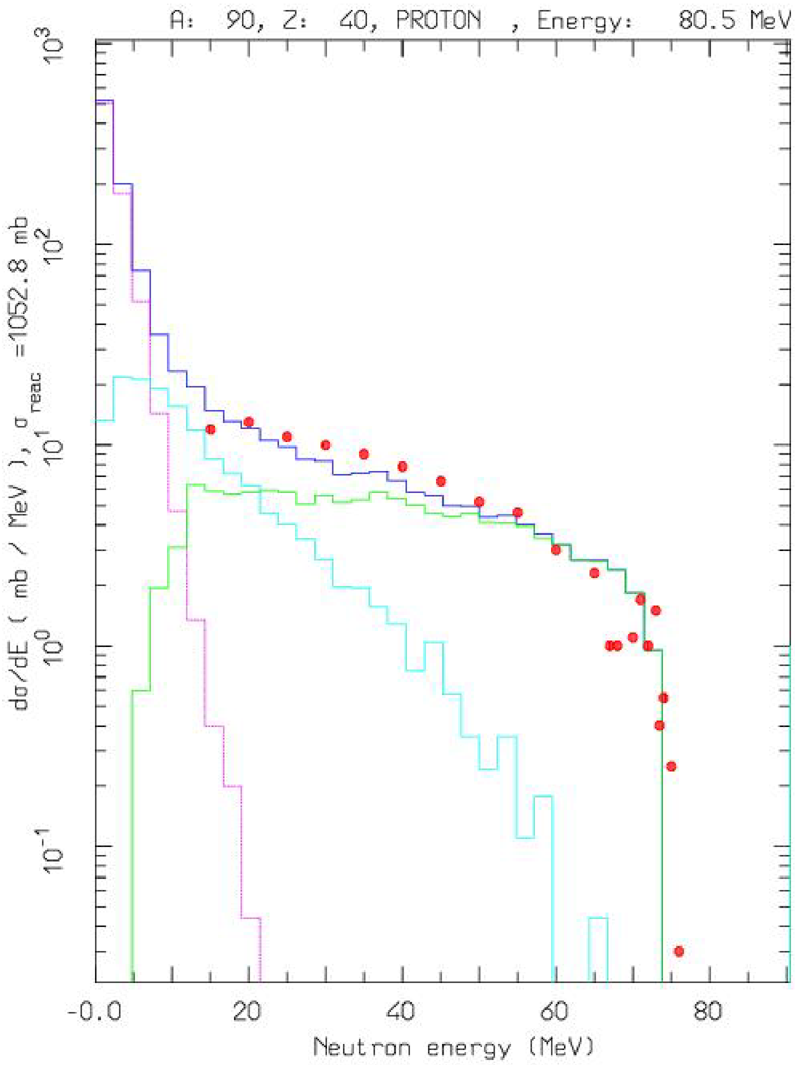
↓

Common FLUKA Evaporation/fission/fragmentation model

Peanut has proven to be a precise and reliable tool for intermediate energy hadron-nucleus reactions

Its "nuclear environment" is also used in the modelization of (real and virtual) photonuclear reactions, neutrino interactions, nucleon decays, muon captures..

Thin target example



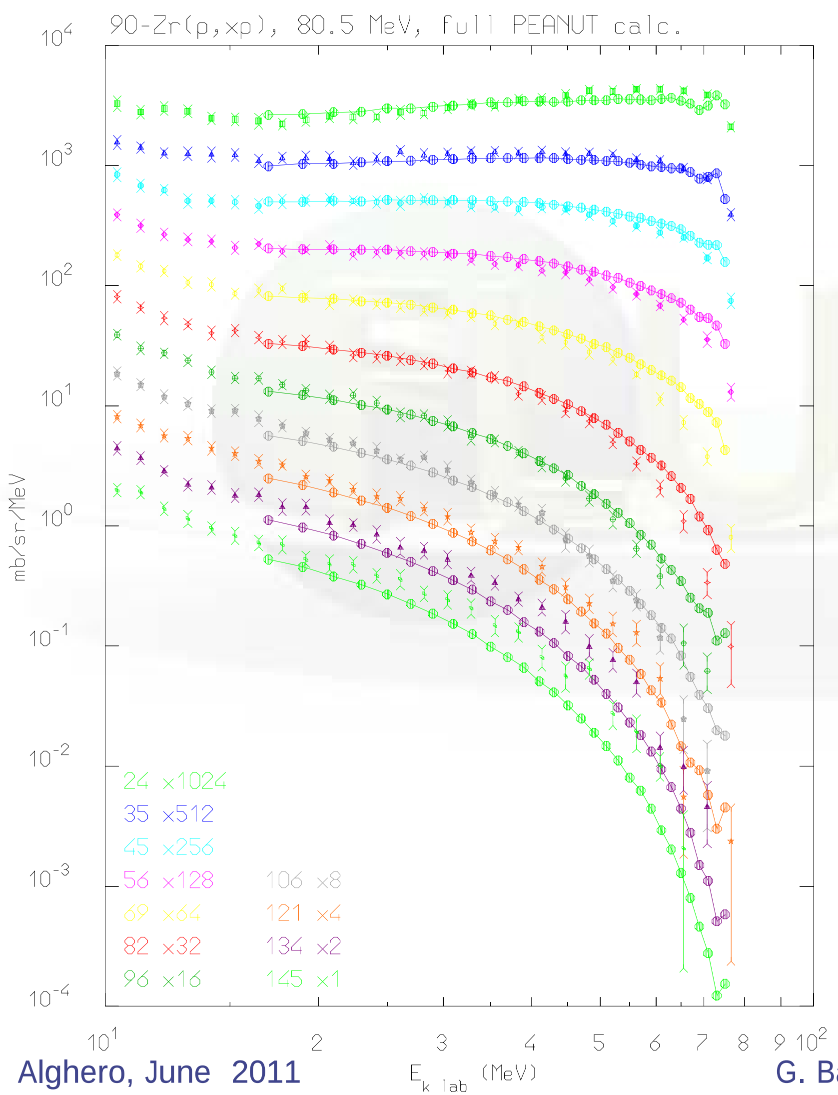
Angle-integrated $^{90}Zr(p,xn)$ at 80.5 MeV

The various lines show the total, INC, preequilibrium and evaporation contributions

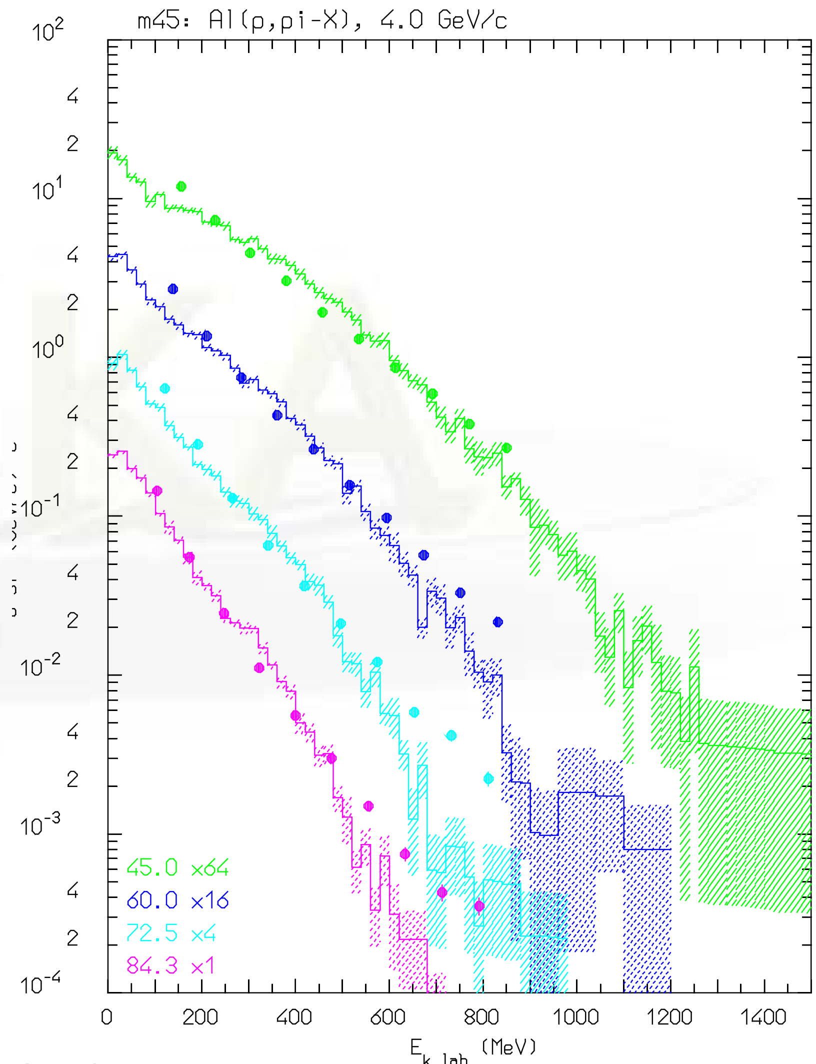
Experimental data from M. Trabandt et al., Phys. Rev. C39, 452 (1989)

Thin target examples

p + ^{80}Zr $\rightarrow$ p + X (80 MeV)

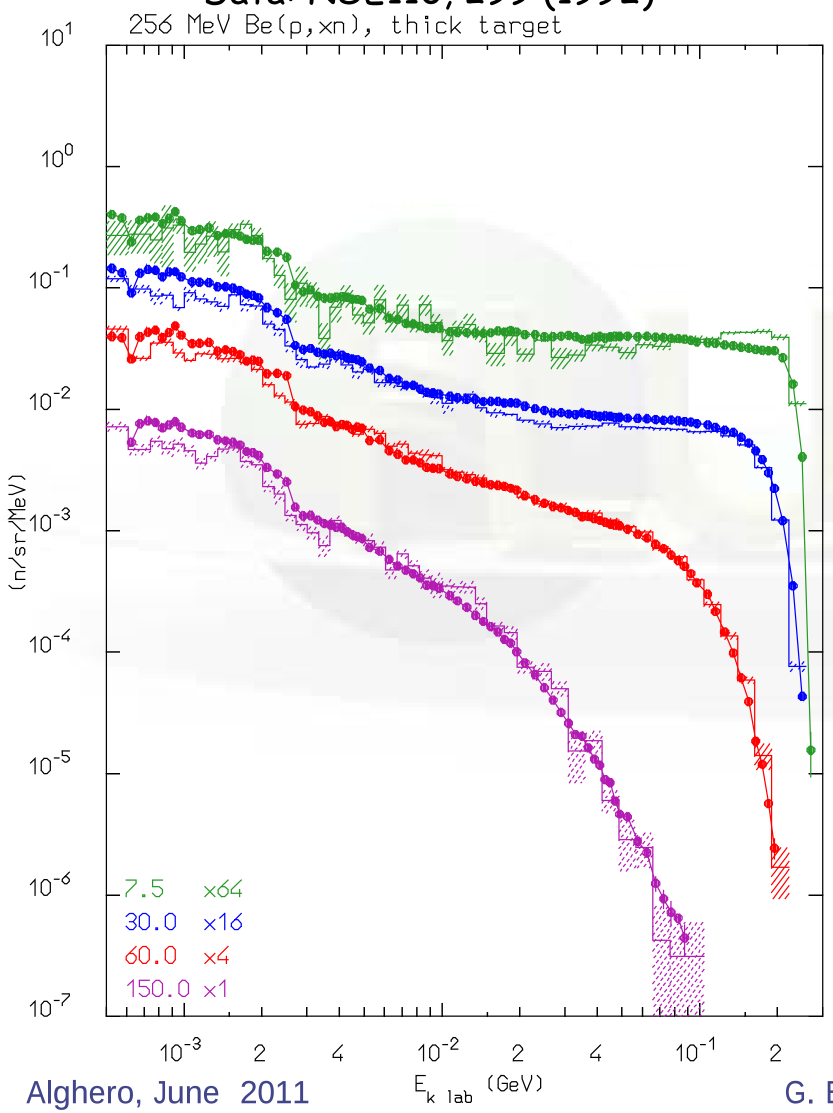


p + Al $\rightarrow$ π^- + X (4 GeV/c)

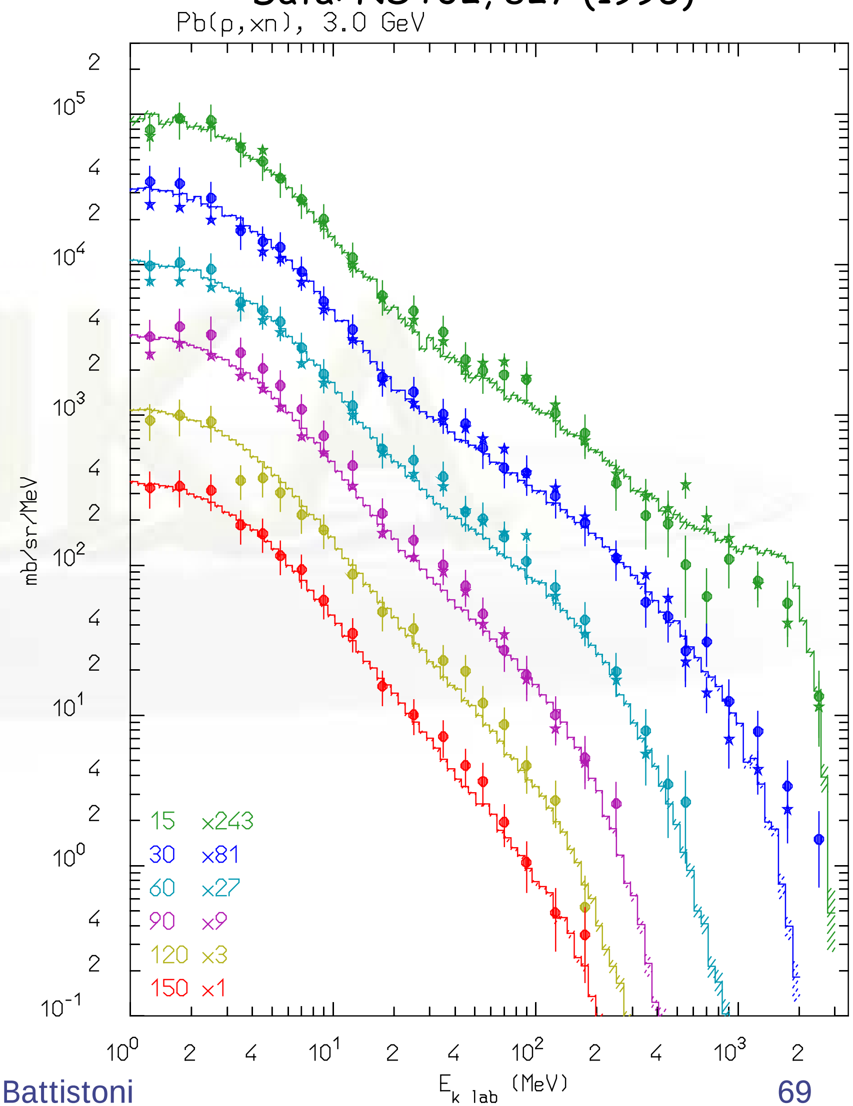


Thick/Thin target examples: neutrons

$^9Be(p,xn)$ @ 256 MeV, stopping target
Data: NSE110, 299 (1992)



Pb(p,xn) @ 3 GeV, thin target
Data: NST32, 827 (1995)



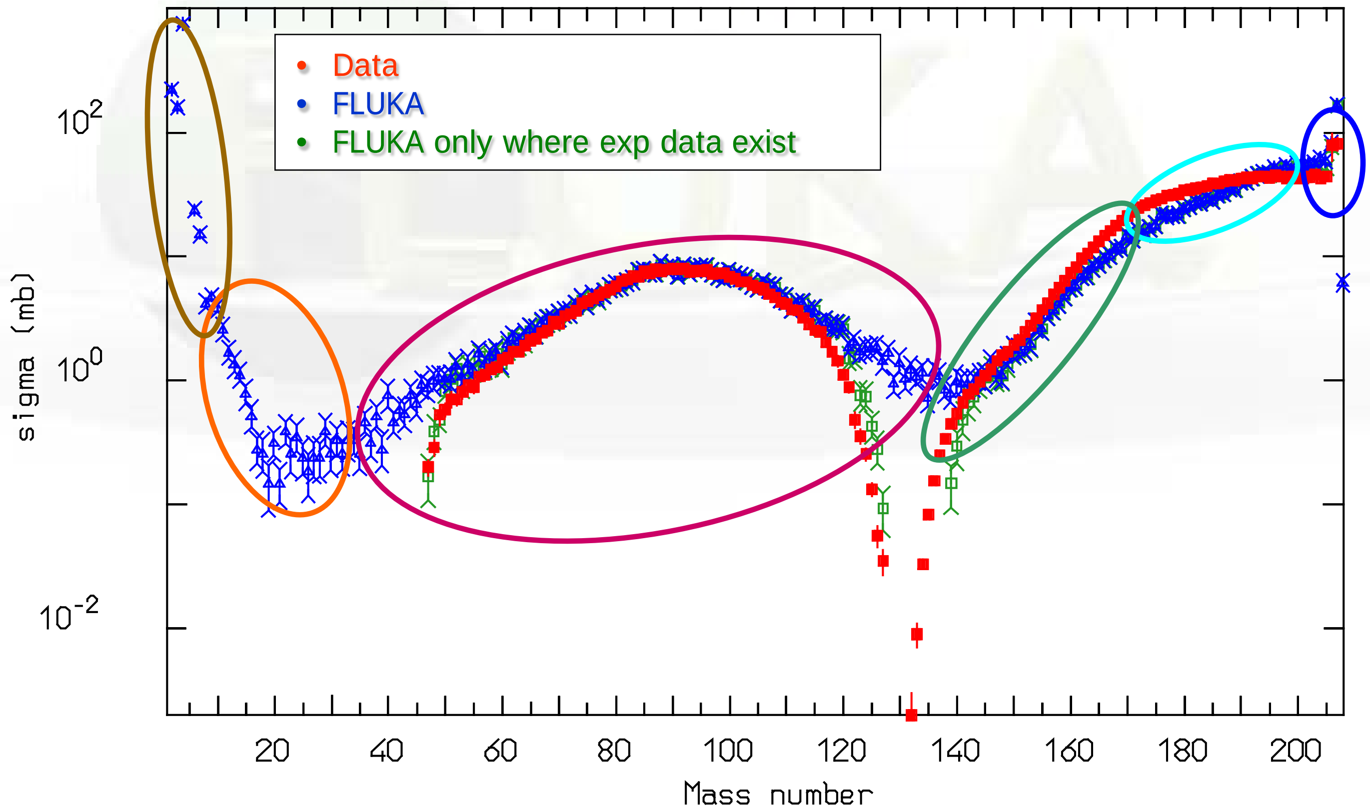
Equilibrium particle emission

- **Evaporation**: Weisskopf-Ewing approach
  - 600 possible emitted particles/states (A<25) with an extended evaporation/fragmentation formalism
  - Full level density formula
  - Inverse cross section with proper sub-barrier
  - Analytic solution for the emission widths
  - Emission energies from the width expression with no. approx.
  - NEW! New energy dependent self-consistent evaporation level densities (IAEA recommendations)
  - NEW! New pairing energies consistent with the above point
  - NEW! Extension of mass tables till A=330 using available offline calculations
  - NEW! New shell corrections coherent with the new masses
- **Fission**:
  - NEW! Actinide fission done on first principles
  - NEW! New fission barrier calculations (following Myers & Swiatecki)
  - NEW! Fission level density enhancement at saddle point washing out with excitation energy (following IAEA recommendations)
  - NEW! Fission product widths and asymmetric versus symmetric probabilities better parameterized
- **Fermi Break-up** for A<18 nuclei
  - ~ 50000 combinations included with up to 6 ejectiles
- γ **de-excitation**: statistical + rotational + tabulated levels

Example of fission/evaporation

- Quasi-elastic products
- Spallation products
- Deep spallation products
- Fission products
- Fragmentation products
- Evaporation products

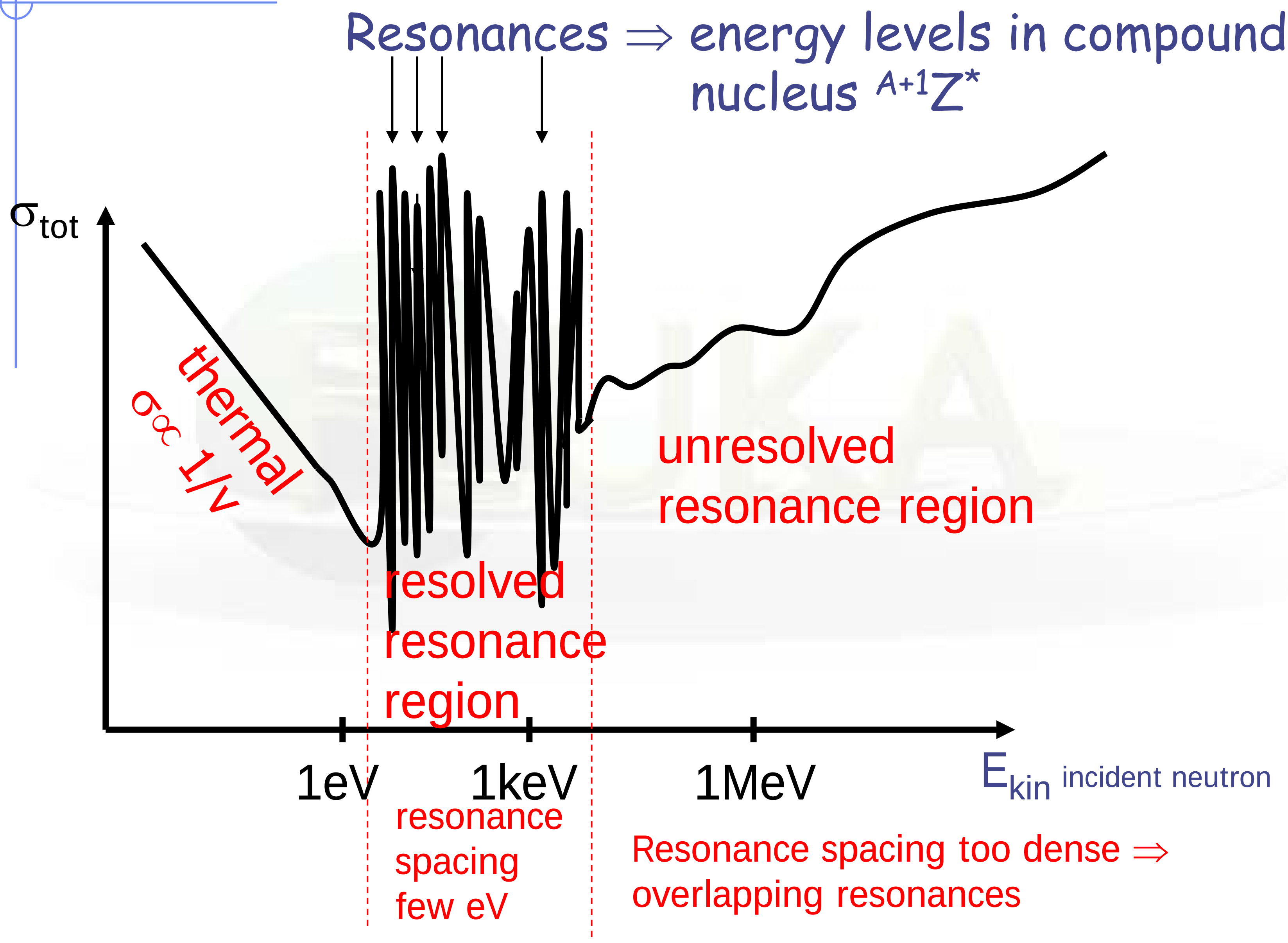
1 A GeV ^{208}Pb + p reactions Nucl. Phys. A 686 (2001) 481-524



Heavy ion interaction models

- DPMJET-III for energies ≥ 5 GeV/n
  - DPMJET (R. Engel, J. Ranft and S. Roesler) Nucleus-Nucleus interaction model
  - Energy range: from 5-10 GeV/n up to the highest Cosmic Ray energies (10^{18}-10^{20} eV)
  - Used in many Cosmic Ray shower codes
  - Based on the Dual Parton Model and the Glauber model, like the high-energy FLUKA hadron-nucleus event generator
- Modified and improved version of rQMD-2.4 for 0.1 < E < 5 GeV/n
  - rQMD-2.4 (H. Sorge et al.) Cascade-Relativistic QMD model
  - Energy range: from 0.1 GeV/n up to several hundred GeV/n
  - Successfully applied to relativistic A-A particle production
- BME (BoltzmannMasterEquation) for E< 0.1 GeV/n
  - FLUKA implementation of BME from E.Gadioli et al (Milan)
  - Now under test for A≤ 16
- Standard FLUKA evaporation/fission/fragmentation used in both Target/Projectile final deexcitation ⇒ ***Projectile-like evaporation is responsible for the most energetic fragments***
- Electromagnetic dissociation

Typical neutron cross section



Evaluated Nuclear Data Files

- Evaluated nuclear data files (ENDF, JEFF, JENDL...)
  - typically provide neutron σ (cross sections) for E<20MeV for all channels
  - σ are stored as continuum + resonance parameters
  - Complex programs like NJOY, PREPRO convert the ENDF file to P-ENDF (point-wise cross sections), or G-ENDF (group-wise) including Doppler broadening etc.

Point-wise and Group-wise cross sections

- In neutron transport codes in general two approaches used: point-wise ("continuous" cross sections) and group-wise transport
- Point-wise follows cross section precisely but is can be time and memory consuming
- Group approach is widely used in neutron transport codes because it is fast and gives good results for most applications

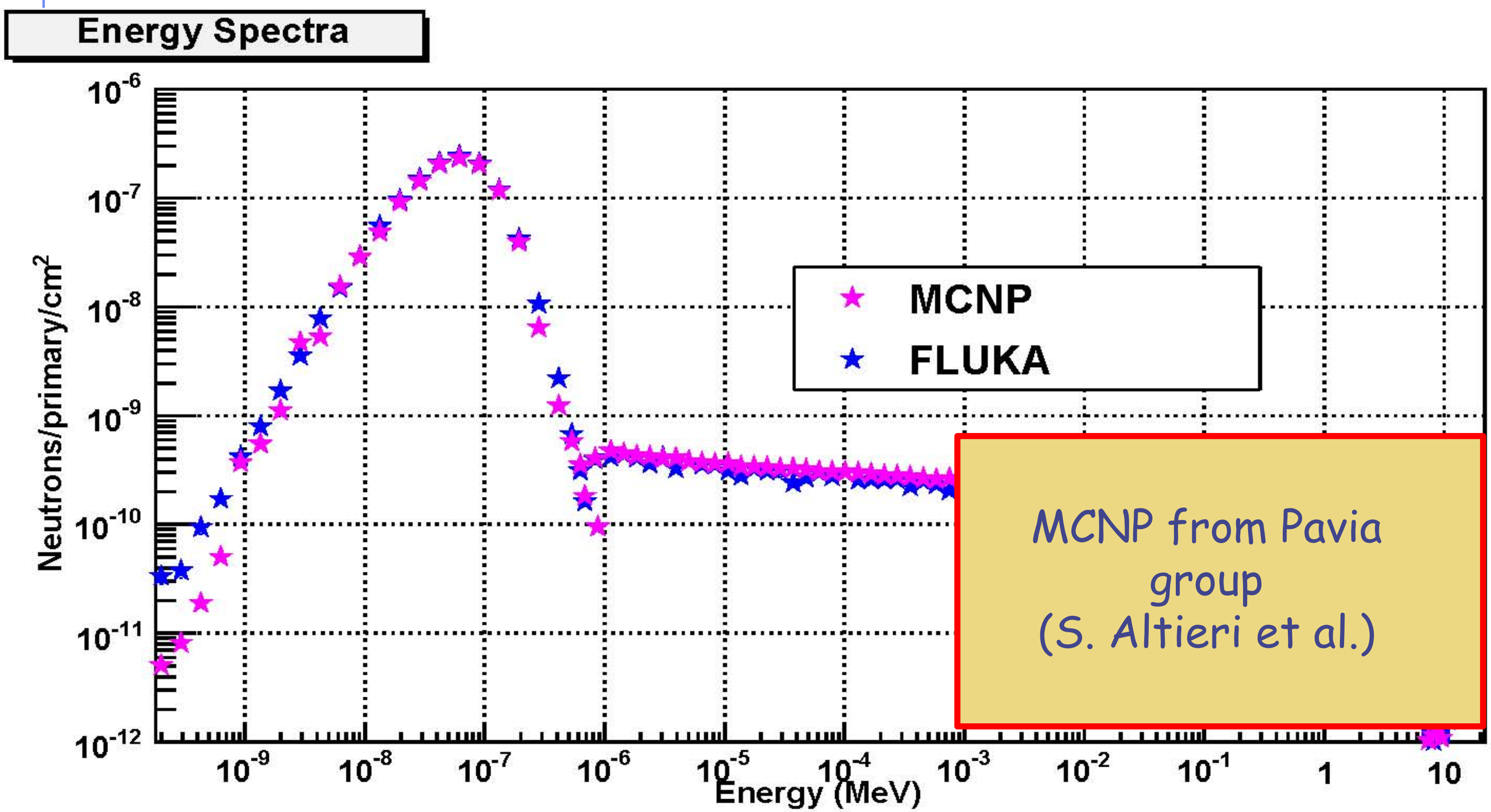
Group Transport Technique

- The energy range of interest is divided in a given number of discrete intervals ("energy groups")
- Elastic and inelastic reactions simulated not as exclusive processes, but by group-to-group transfer probabilities (downscattering matrix)
- Downscattering matrix: if a neutron in a given group undergoes a scattering event and loses energy, it will be transferred to a group of lower energy (each of the lower energy groups having a different probability)
- If the neutron does not lose enough energy to be in another group, it will stay in the same group (in-scattering).
- In thermal region neutrons can gain energy. This is taken into account by an upscattering matrix, containing the transfer probability to a group of higher energy

The FLUKA Low Energy Neutron Library

- FLUKA uses the multigroup transport technique
- The energy boundary below which multigroup transport takes over depends in principle on the cross section library used. In the present library it is 20 MeV.
- Both fully biased and semi-analog approaches are available
- Number of groups: 260 of approximately equal logarithmic with, the actual energies limits of each group can be found in the manual (or can be printed to *.out file)
- N.B. the group with the highest energy has the number 1, the group with the lowest energy has number 260
- 31 thermal groups, with 30 upscattering groups
- Energy range of library: 0.01 meV - 20 MeV

Simulation of neutron spectrum from reactor (Pavia)



Radiation Protection Aspects of the SPES facility at LNL

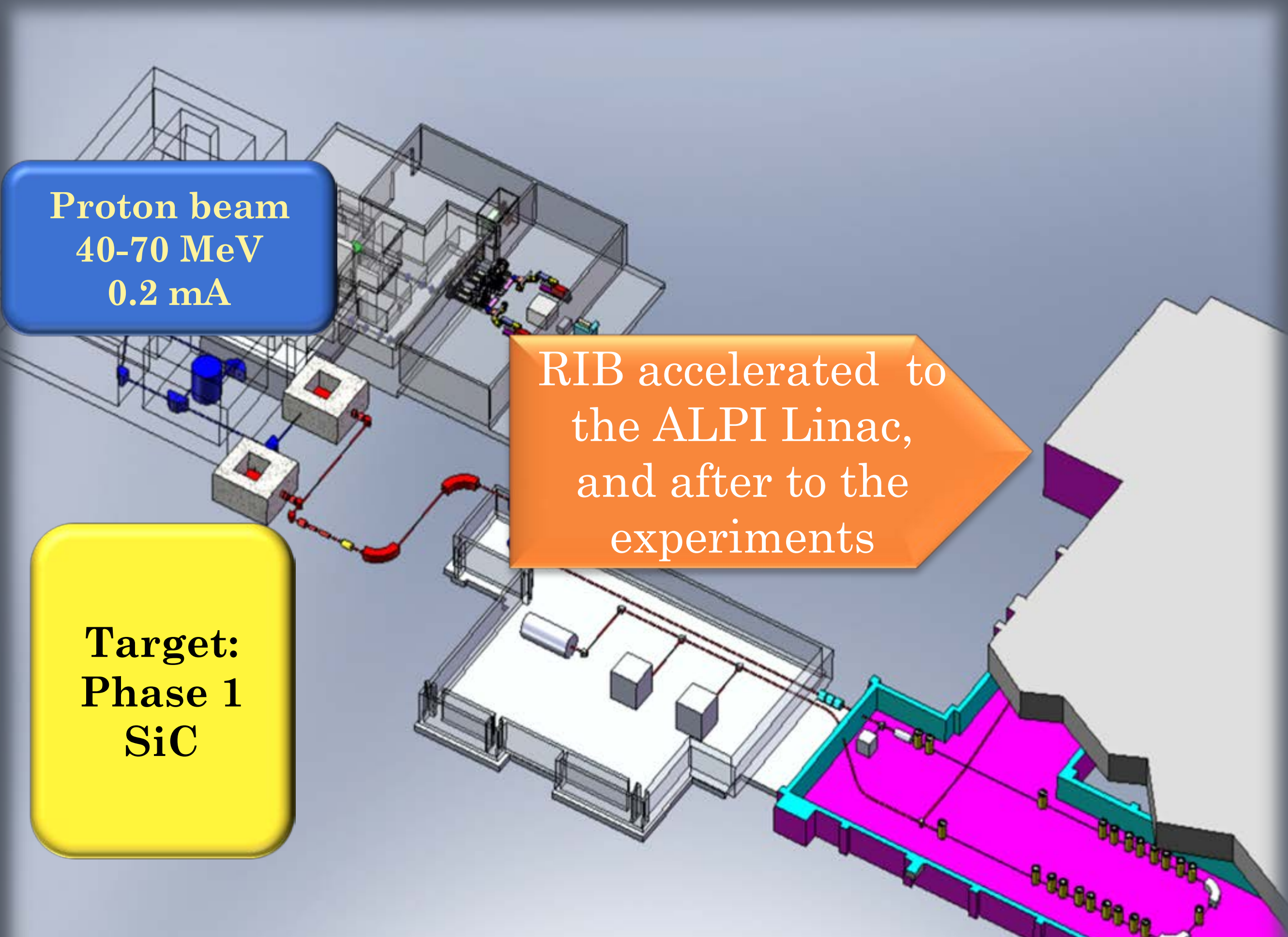
L. Sarchiapone, D. Zafiropoulos

INFN, Laboratori Nazionali di Legnaro, Italy

Shielding Aspects of Accelerators, Targets and Irradiation Facilities, 2-4 June 2010

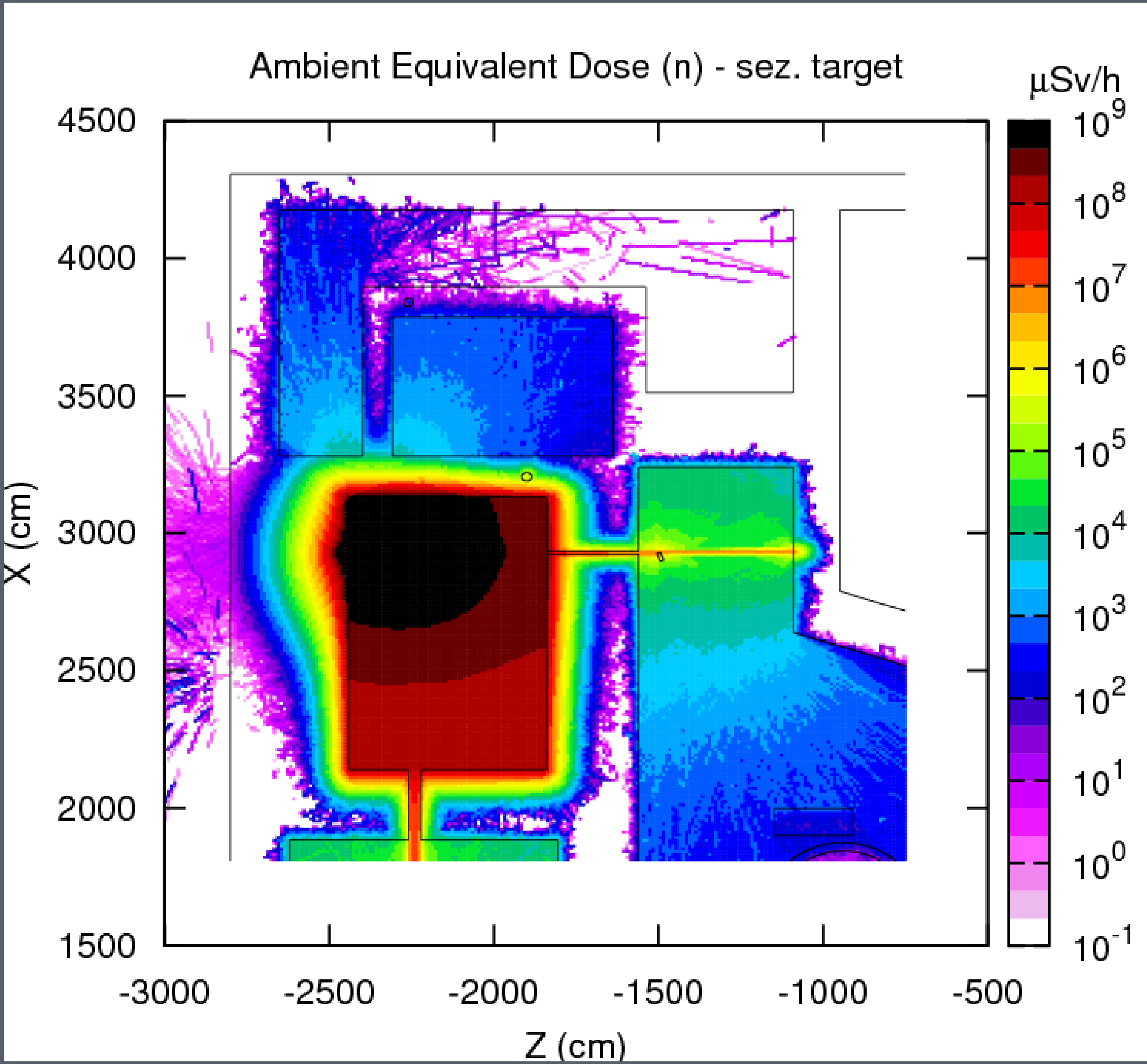
THE SPES PROJECT

Selective Production of Exotic Species



All the radiation protection aspect object of this work have been evaluated using the FLUKA Monte Carlo code.

SHIELDING ASPECTS: TARGET UC_2



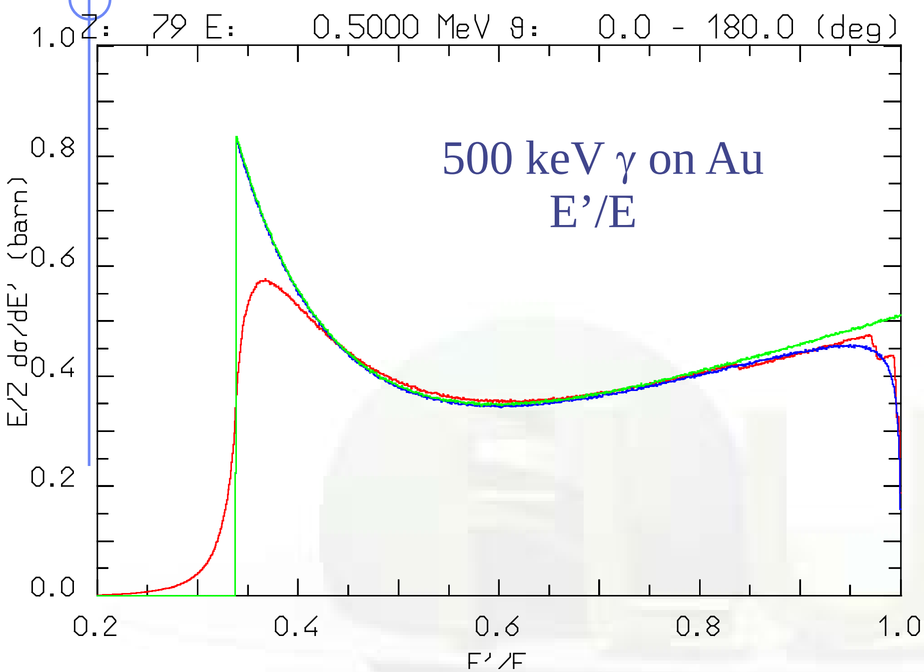
EM Physics

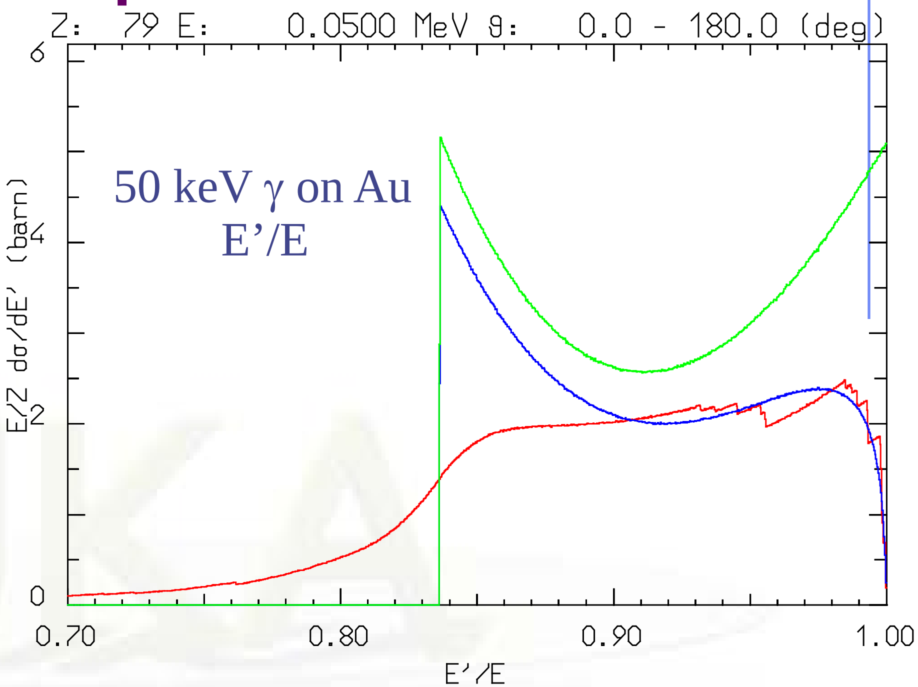
- General settings
- Interactions of leptons/photons
  - Photon interactions
    - Photoelectric
    - Compton
    - Rayleigh
    - Pair production
    - Photonuclear
    - Photomuon production
  - Electron/positron interactions
    - Bremsstrahlung
    - Scattering on electrons
  - Muon interactions
    - Bremsstrahlung
    - Pair production
    - Nuclear interactions

- Ionization energy losses
  - Continuous
  - Delta-ray production
- Transport
  - Multiple scattering
  - Single scattering

These are common to all charged particles, although traditionally associated with EM

Compton profile examples





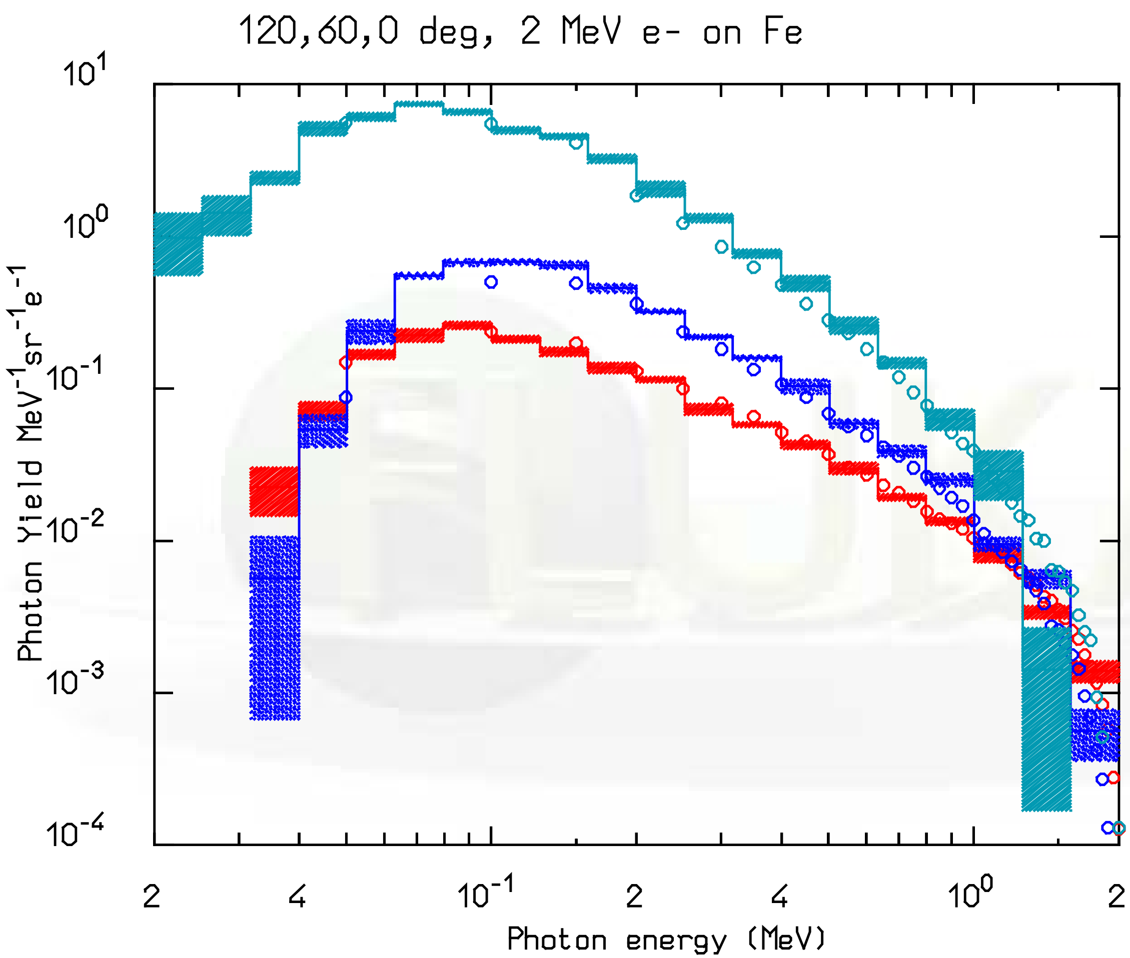
green = free electron
blue = binding with form factors
red = binding with shells and orbital motion

Larger effect at very low energies, where, however, the dominant process is photoelectric.
Visible: shell structure near E'=E, smearing from motion at low E'

Compton and Rayleigh

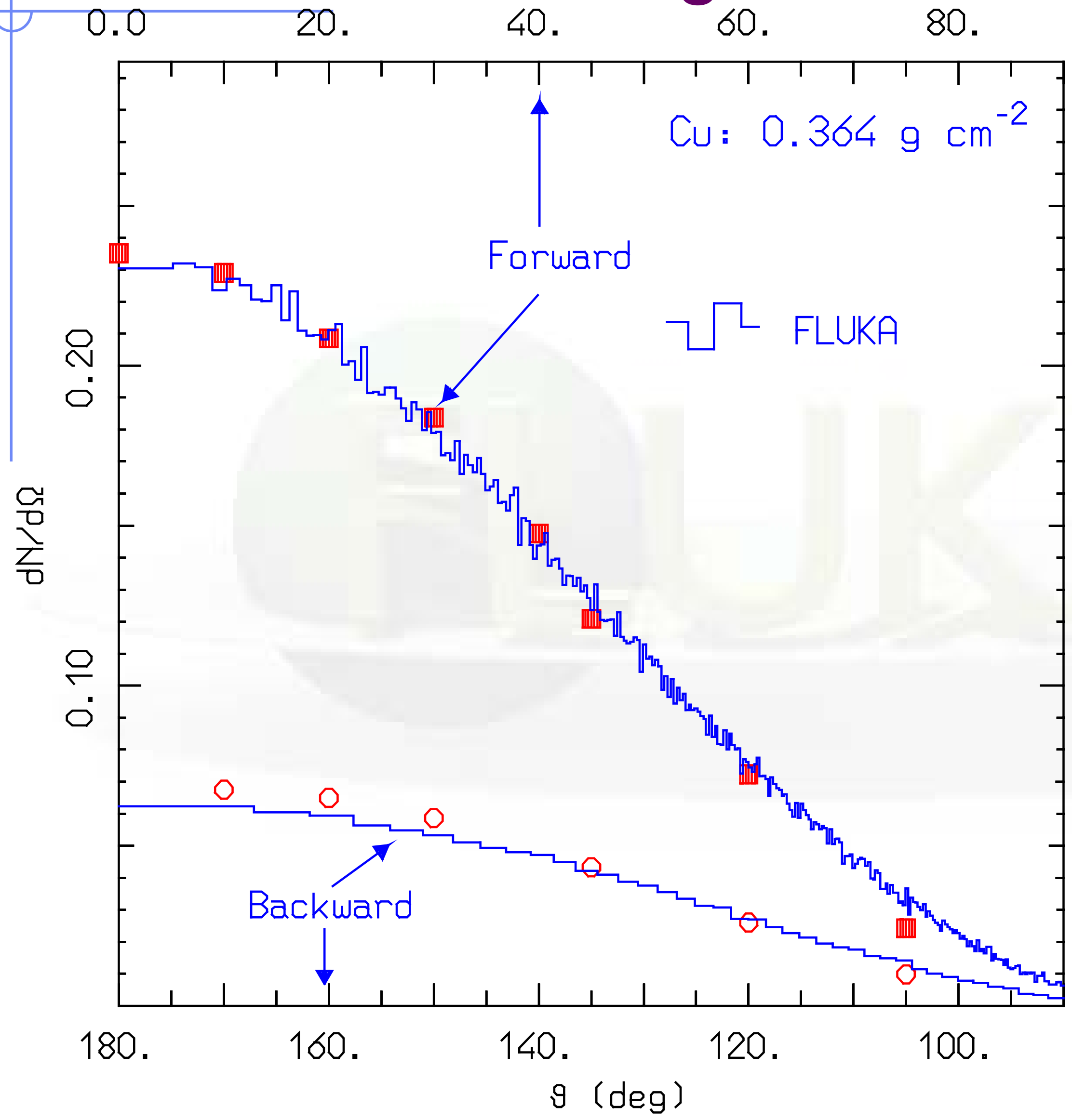
- Account for atomic bonds using inelastic Hartree-Fock form factors (very important at low E in high Z materials)
- NEW : Compton with atomic bonds and orbital motion (as better alternative to form factors)
  - Atomic shells from databases
  - Orbital motion from database + fit
  - Followed by fluorescence
- Account for effect of photon polarization

Bremsstrahlung: benchmark



2 MeV electrons on Iron, Bremsstrahlung photon spectra measured (dots) and simulated (histos) at three different angles

Electron scattering:



Transmitted (forward) and backscattered (backward) electron angular distributions for 1.75 MeV electrons on a 0.364 g/cm^2 thick Copper foil Measured (dots) and simulated (histos) data

Photonuclear int.: example

Reaction:

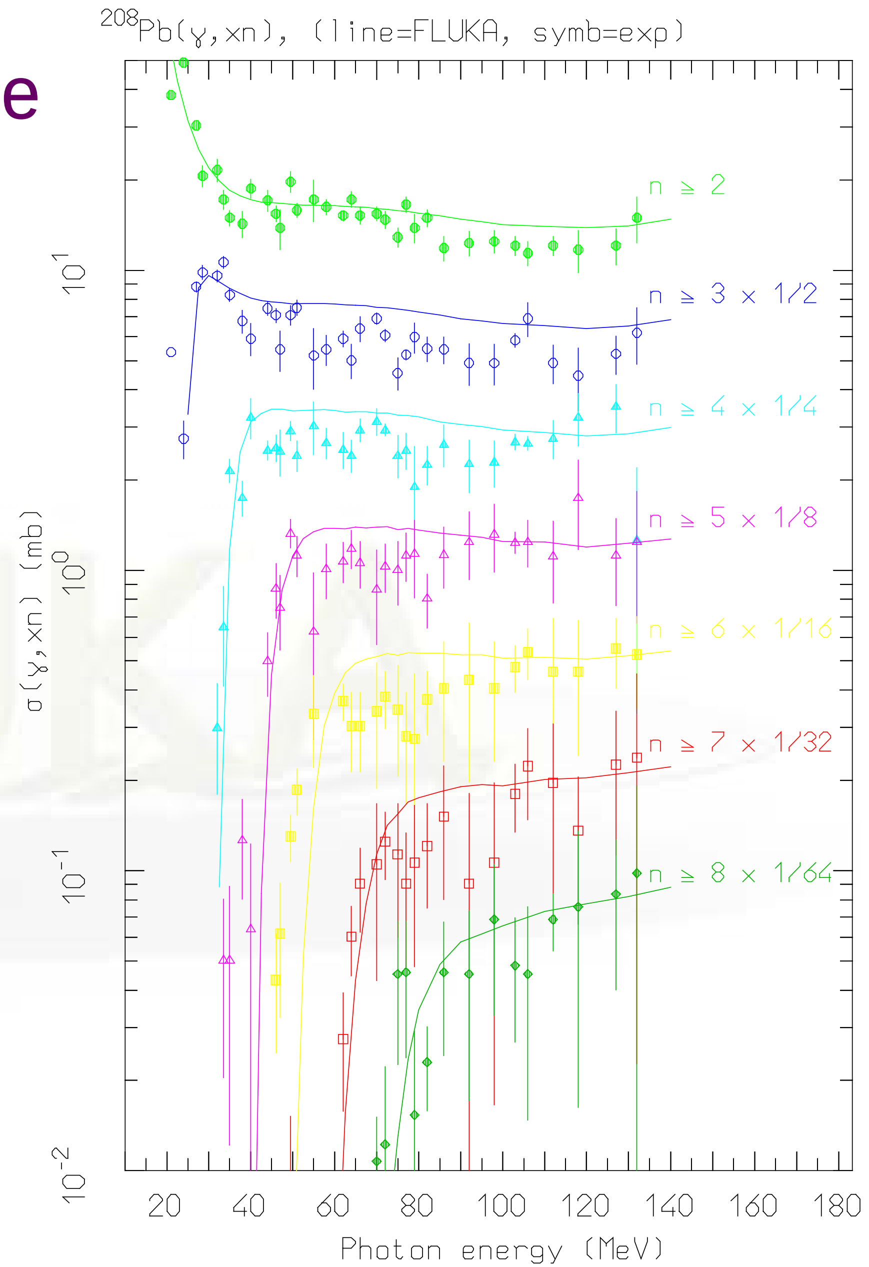
^{208}Pb(γ,x n)

20≤Eγ ≤140 MeV

Cross section for multiple neutron emission as a function of photon energy, Different colors refer to neutron multiplicity ≥ *n* , with 2 ≤ *n* ≤ 8

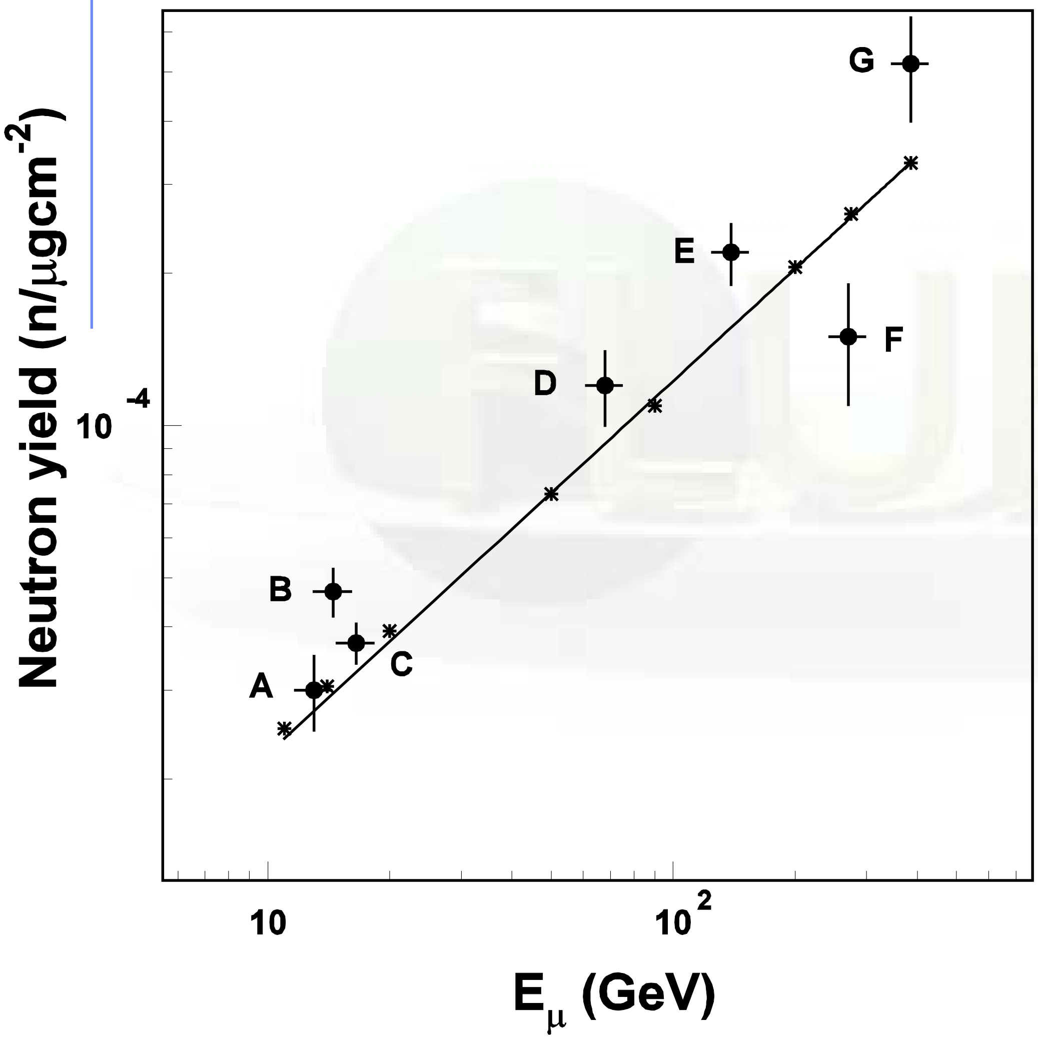
Symbols: exp. data (NPA367, 237 (1981) ; NPA390, 221 (1982))

Lines: FLUKA



Muon-induced neutron background in underground labs

PRD64 (2001) 013012



Neutron production rate as a function of muon energy

Stars+line : FLUKA simulation with a fit to a power law.

Exp. points:
abscissa → average μ energy at the experiment's depth:

A) 20 m.w.e.
B) 25 m.w.e.
C) 32 m.w.e. (Palo Verde)
D) 316 m.w.e.
E) 750 m.w.e.
F) 3650 m.w.e. (LVD)
G) 5200 m.w.e. (LSD)

Charged particle dE/dx: Bethe-Bloch

Spin 0
(spin1 is similar):

~ln $\beta^4\gamma^4$
relativistic rise

$$\left(\frac{dE}{dx}\right)_0 = \frac{2\pi n_e r_e^2 m_e c^2 z^2}{\beta^2}\left[\ln\left(\frac{2m_e c^2 \beta^2 T_{\max}}{I^2(1-\beta^2)}\right) - 2\beta^2 + 2zL_1(\beta) + 2z^2 L_2(\beta) - 2\frac{C}{Z} - \delta + G\right]$$

- I : mean excitation energy , material-dependent
- δ : density correction
- C : is the shell correction, important at low energies
- T_{max} : maximum energy transfer to an electron (from kinematics)

Higher order corrections implemented in FLUKA

- L1 : Barkas (z^3) correction responsible for the difference in stopping power for particles-antiparticles
- L2 the Bloch (z^4) correction
- G : Mott corrections

Valid for m>>m_e, However, the formulation for electron/positrons is similar, with the exception of "energetic" collisions with atomic electrons.

Discrete ionization events

Above a pre-set threshold, ionization is modeled as δ ray production (free electrons)

- Spin 0 or 1/2 δ-ray production (charged hadrons, muons)
- Mott for heavy ions
- Bhabha scattering (e^+)
- Møller scattering (e^-)

Below the pre-set threshold for δ ray production:

Restricted energy losses

For particles much heavier than electrons and charge z, with energy transfers to atomic electrons restricted at T_δ

Continuous energy losses

Below the δ-ray threshold, energy losses are treated as "continuous", with some special features:

- Fluctuations of energy loss are simulated with a FLUKA- specific algorithm
- The energy dependence of cross sections and dE/dx is taken into account exactly (see later)
- Latest recommended values of ionization potential and density effect parameters implemented for elements (Sternheimer, Berger & Seltzer), but can be overridden by the user with (set yourself for compounds!)

Ionization fluctuations - I

The Landau distribution is limited in several respects:

- Max. energy of δ rays assumed to be ∞ $\Longrightarrow$ cannot be applied for long steps or low velocities
- cross section for close collisions assumed equal for all particles
- fluctuations connected with distant collisions neglected $\Longrightarrow$ cannot be applied for short steps
- incompatible with explicit δ-ray production

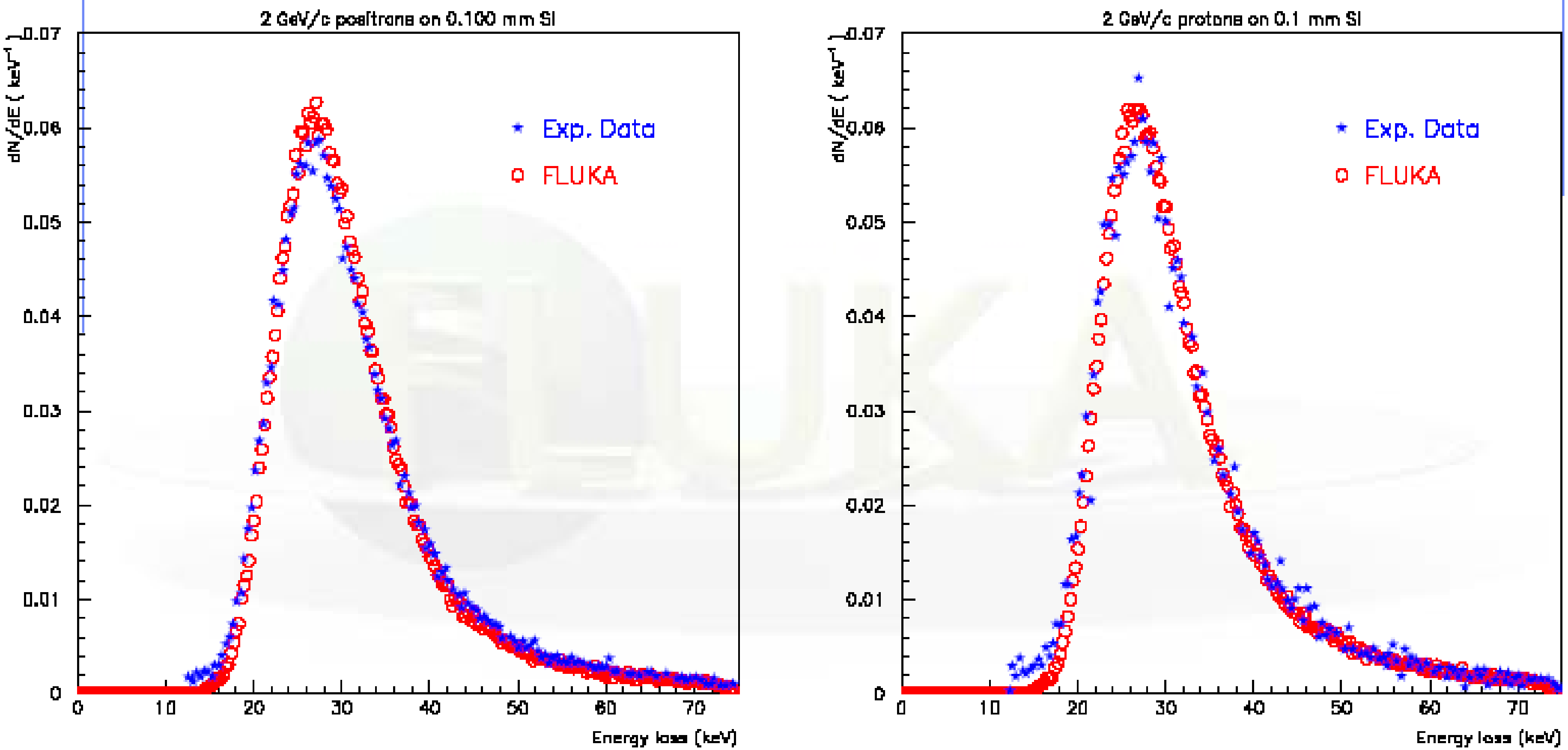
The Vavilov distribution overcomes some of the Landau limitations, but is difficult to compute if step length or energy are not known *a priori*.

Ionization fluctuations - II

The FLUKA approach:

- based on general statistical properties of the cumulants of a distribution (in this case a Poisson distribution convoluted with $\mathrm{d}\boldsymbol{\sigma}/\mathrm{d}\boldsymbol{E}$)
- integrals can be calculated analytically and exactly *a priori* $\Longrightarrow$ minimal CPU time
- applicable to any kind of charged particle, taking into account the proper (spin-dependent) cross section for δ ray production
- the first 6 moments of the energy loss distribution are reproduced ($k_n =< (x- < x >)^n >$)

Ionization fluctuations -III



Experimental [1] and calculated energy loss distributions for 2 GeV/c positrons (left) and protons (right) traversing 100μm of Si J.Bak et al. NPB288, 681 (1987)

Nuclear stopping power (NEW)

- Besides Coulomb scattering with atomic electrons, particles undergo Coulomb scattering also with atomic nuclei
- The resulting energy losses, called nuclear stopping power, are smaller than the atomic ones, but are important for
  - Heavy particles (i.e. ions)
  - Damage to materials

Code complexity

- Inelastic h-N: ~*72000 lines*
- Cross sections (h-N and h-A), and elastic (h-N and h-A): *~32000 lines*
- (G)INC and preequilibrium (PEANUT): *~114000 lines*
- Evap./Fragm./Fission/Deexc.: *~27000 lines*
- ν-N interactions: *~35000 lines*
- A-A interactions:
  - ✓ FLUKA native (including BME): *~8000 lines*
  - ✓ DPMJET-3: *~130000 lines*
  - ✓ (modified) rQMD-2.4: *~42000 lines*

- **FLUKA** in total (including transport, EM, geometry, scoring): *~680000 lines*
- … + *~20000 lines* of ancillary off-line codes used for data pre-generation
- … and *~30000 lines* of post-processing codes

THE FLUKA GRAPHICAL USER INTERFACE: AN INTRODUCTION TO FLAIR

Why is UI design important

- User Interfaces are what allows end users to interact with an application.
- A good UI will make an application intuitive and easy to use
- Excellent applications without good UI will be less popular than inferior ones with a good UI

What makes a good UI?

General:

- Simple
- Intuitive
- Respects the commonly accepted conventions
- Visually organized
- Native look
- Easily install and setup
- Extensible / Programmable

FLUKA:

- Do not hide the inner functionality
- Provide a platform for working/analyzing results

Language Choice

	Python	Java	Root/cint	C/C++
Distribution	Fedora: Pre-Installed M$ Win: installer, cygwin	Linux: package M$ Win: Installer, no-gygwin	Linux: package M$ Win: procedure no-cygwin	Linux: Pre-installed M$ Win: cygwin, djgpp
Flavors	Single	Several	Single	Many
Interpreted	√	√ VM	√	
Compiled		√ VM	√	√
Source Portability	√	√	√	
Binary Portability	√	√		
Interactive	√		√	

What is Python?

Python is a scripting language which is:

- interpreted
- interactive
- object-oriented
- like pseudo code
- dynamically typed
- available for many platforms
- extensible with C-API

Free from: http://www.python.org

GUI toolkits for Python

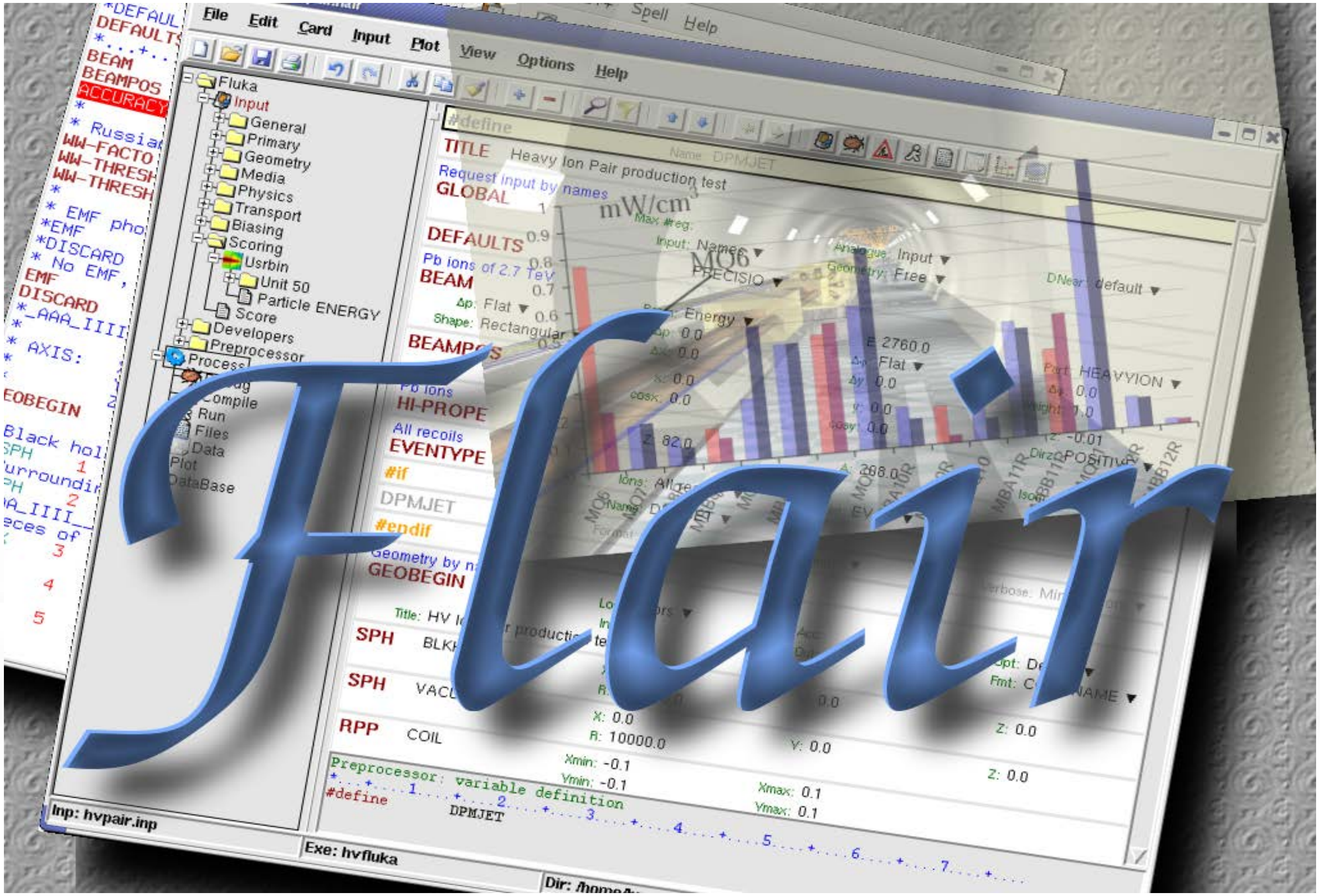
1st Choice

- Tkinter — default GUI toolkit for Python. Good for simple UIs. Portable, wrapper around tk/tcl
- wxPython — Most popular. Good for complex UIs. Wrapper on Win32, GTK
- JPython — Access to the Swing library
- PyGTK — Access to the well-known GTK toolkit
- PyQt — Access to the well-known Qt library
- win32all — Access to MFC from python (MS-Win only)
- WPY — MFC style, both also available for UNIX
- X11 — Limited to X Windows.

Plotting Engine

matplotlib	python 2D plotting library http://matplotlib.sourceforge.net
gnuplot-py	Python interface to gnuplot http://gnuplot-py.sourceforge.net
pyROOT	Python interface to ROOT

About



/fleə(r)/ n [U,C] natural or instinctive ability (to do something well, to select or recognize what is best, more useful, etc.

[*Oxford Advanced Dictionary of Current English*]

What is flair [1/2]

FLUKA Advanced Interface [http://www.fluka.org/flair]

- **All-in-one** User friendly graphical Interface;
- Minimum requirements on additional software;
- Working in an intermediate level
  Not hiding the inner functionality of FLUKA

Front-End interface:

- Fully featured **Input file Editor**
  - Mini-dialogs for each card, allows easy and almost error free editing
  - Uniform treatment of all FLUKA cards
  - Card grouping in categories and card filtering
  - Error checking and validation of the input file during editing
- **Geometry:** transformation, optimizations and debugging
- **Compilation** of the FLUKA Executable
- **Running** and **monitoring** of the status of a/many run(s)

What is flair [2/2]

Back-End interface:

- Inspection of the output files (core dumps and directories)
- Output file viewer dividing into sections
- Post processing (merging) the output data files
- Plot generation through an interface with gnuplot;

Other Goodies:

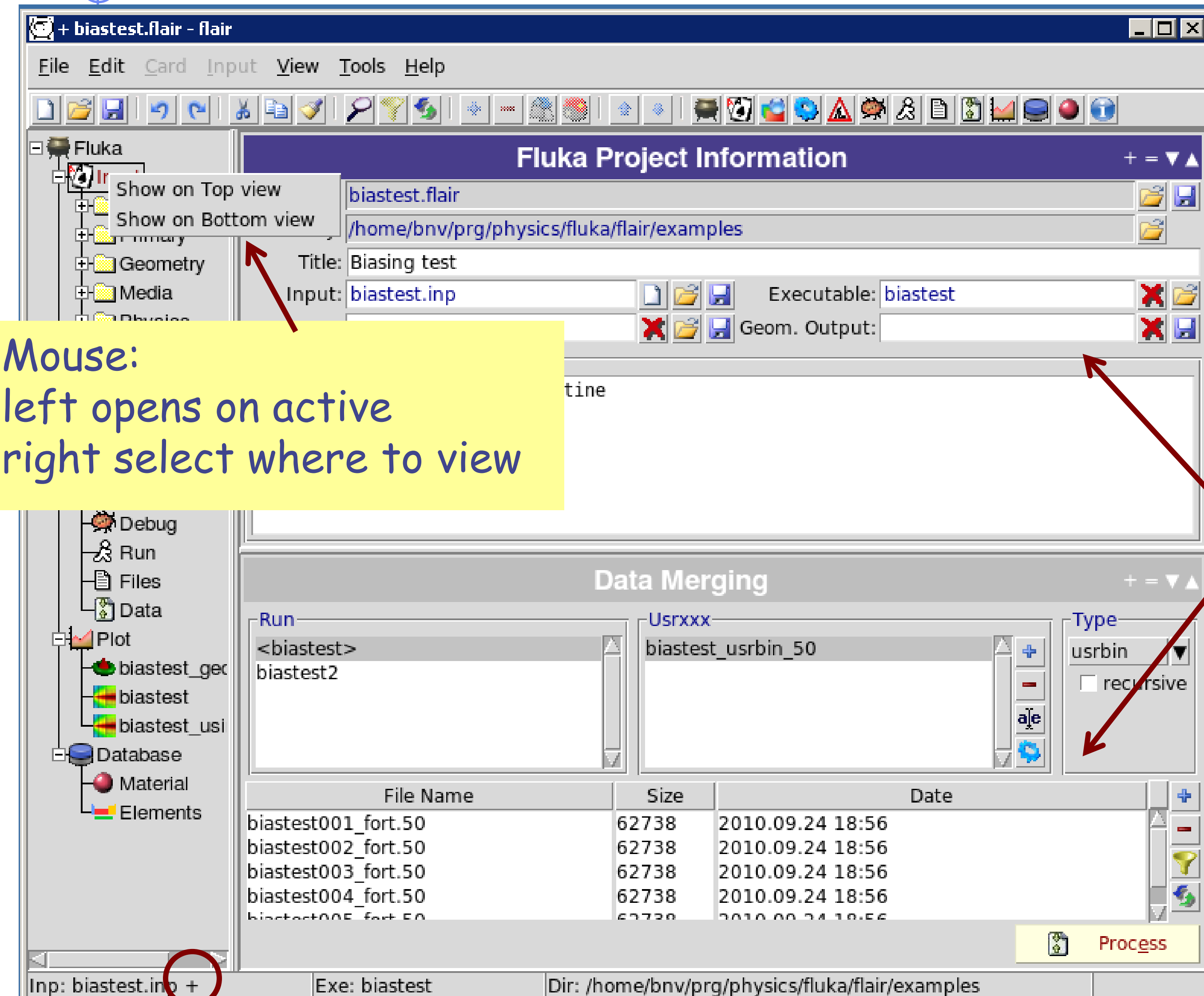
- Geometry viewer/debugger with 3D rendering (see talk tomorrow);
- Access to FLUKA manual as hyper text
- Checking for release updates of FLUKA and flair
- Nuclear wallet cards
- Library of materials
- Database of geometrical objects (Not yet completed)
- Programming python **API**

Concepts: Flair Project

- Store in a single file all relevant information:
  - Project notes
  - Links to needed files: input file, source routines, output files …
  - Multiple runs from the same input file, as well running status
  - Procedures on how to run the code
  - Rules on how to perform data merging
  - Information on how to post process and create plots of the results

- You can consider Flair as an editor and manager of the project files.

Interface



Mouse:
left opens on active
right select where to view

active

+ vertical/horizontal
= equalize
▼ minimize
▲ maximize

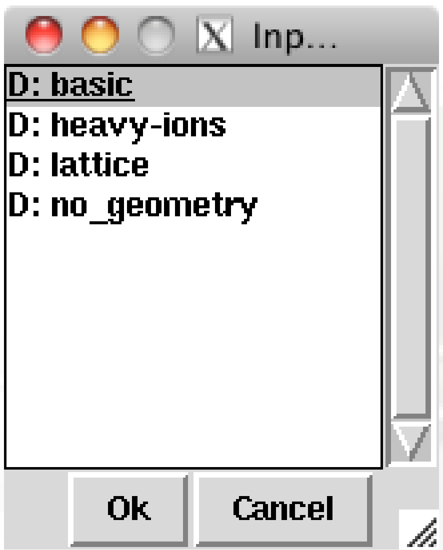
2 working frames

inactive
click to activate

input modified and not saved

Input Templates

- When requesting a new input or a new project flair will prompt to select an input template:



Default template: basic.inp

```
TITLE

GLOBAL                                          1.0       1.0
DEFAULTS                                                               NEW-DEFA
BEAM
BEAMPOS
GEOBEGIN                                                               COMBNAME
    0    0
* Black body
SPH blkbody    0.0 0.0 0.0 10000000.0
* Void sphere
SPH void       0.0 0.0 0.0 1000000.0
* Cylindrical target
RCC target     0.0 0.0 0.0 0.0 0.0 10.0 5.0
END
* Black hole
BLKBODY      5   +blkbody -void
* Void around
VOID         5   +void -target
* Target
TARGET       5   +target
END
GEOEND
* ..+....1....+....2....+....3....+....4....+....5....+....6....+....7..
ASSIGNMA     BLCKHOLE    BLKBODY
ASSIGNMA       VACUUM       VOID
ASSIGNMA       COPPER     TARGET
RANDOMIZ          1.0
START
STOP
```

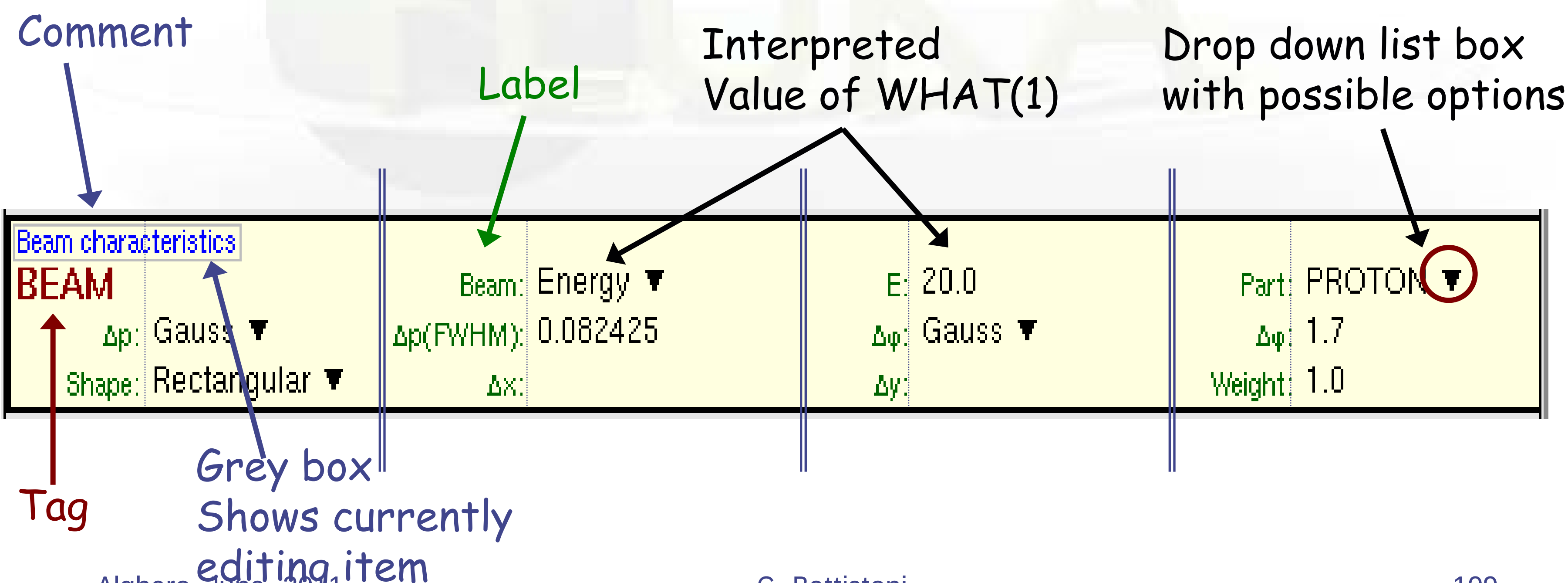
- Flair default templates are prefixed with "D:"
- User templates will be prefixed with "U:"

The user can create his own set of input templates. They are normal FLUKA input files and they have to be placed in the directory ~/.flair/templates (create the directory if not existing)

Anatomy of a card mini-dialog [1/2]

- For each extended card flair has a mini dialog (currently in 4 columns), interpreting all information stored in the card

```
* Beam characteristics
BEAM          -20.0 -0.082425      -1.7                                 1.0PROTON
```



Anatomy of a card mini-dialog [2/2]

```
* Energy deposition in 3D binning
USRBIN          10.0    ENERGY     -50.0      45.0      54.0      36.0EneDep
USRBIN         -45.0     -54.0     -33.0     100.0     100.0     100.0&
```

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

```
* Polypyromellitimide Polyimide, Kapton
* Chemical      O = C   H-C   C = O
* Formula          / \  // \ / \      H-C - C-H        H-C - C-H
*                 /   \C    C   \    //     \\        //     \\
*           ---- N     |    ||   N - C        C - O - C        C ----
*                 \   /C    C   /    \      /         \      /
*  C  H  N O       \ /  \\ / \ /      H-C = C-H        H-C = C-H
*   22 10 2 5   O = C   H-C    C = O
MATERIAL                                 1.43                              Polyimid
COMPOUND        10.0  HYDROGEN           22.0    CARBON        2.0  NITROGENPolyimid
COMPOUND         5.0    OXYGEN                                              Polyimid
```

MATERIAL	Name: Polyimid	#	ρ: 1.43
Z:	Am:	A:	dE/dx: ▼

COMPOUND	Name: Polyimid ▼	Mix: Atom ▼	Elements: 6 ▼
f1: 10.0	M1: HYDROGEN ▼	f2: 22.0	M2: CARBON ▼
f3: 2.0	M3: NITROGEN ▼	f4: 5.0	M4: OXYGEN ▼
f5:	M5: ▼	f6:	M6: ▼

Material Database

- Flair contains an internal database of ~500 predefined materials and/or compounds;
- Some (~300) with the Sternheimer parameters;
- The database can be edited, and populated with your own materials. In this case a local copy of the database will be made in ~/.flair directory.