



The FLUKA Code: Insight and features



www.fluka.org



[Fluka >>](#)

[Documentation >>](#)

[Download](#)

[Tools >>](#)

[Discuss >>](#)

[Team >>](#)

Quick launch:

[Download](#)
[Mailing list](#)
[Manual Online](#)
[Courses](#)
[Flair](#)
[Contact us](#)

Last version:

FLUKA 2011.2.15,
September 9th 2012
(last respin)
FLAIR 1.0.0

News:

**Fluka Release
(10.08.2012)**
FLUKA 2011.2.14 has
been released.

A A A



Main Authors: A.Fasso`¹, A.Ferrari², J.Ranft³, P.R.Sala⁴

¹ Jefferson Lab, ² CERN, ³ Siegen University, ⁴ INFN Milan

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

Developed and maintained under an INFN-CERN agreement
Copyright 1989-2014 CERN and INFN

--
0

The FLUKA international Collaboration

M. Brugger, M. Calviani, F. Cerutti, A. Fedynitch, Alfredo Ferrari, L. Esposito, R. Esposito, R. GarciaAlia, P. GarciaOrtega, L. Lari, A. Lechner, M. Magistris, C. Mancini, A. Manousos, A. Mereghetti, E. Nowak, S. Roesler, P. Schoofs, L. Skordis, N. Shetty, G. Smirnov, C. Theis, T. Viana Miranda, Heinz Vincke, Helmut Vincke, V. Vlachoudis, J. Vollaie, C. Weiss **CERN Switzerland**

M. Chin, M. Nozar, A. Trudel, M. Trinkzec **TRIUMF Canada**

A. Fassò, R. Versaci **ELI beamlines, Czech** J. Ranft, **Univ. of Siegen, Germany**

T. Boehlen, **MedAustron, Austria** P. Degtiarenko, G. Kharashvili, **JLAB, USA**

G. Battistoni, F. Broggi, M. Campanella, S. Muraro, P.R. Sala, **INFN Milano, Italy**

A. Margiotta, M. Sioli, **Univ. and INFN Bologna, Italy**

V. Patera **Univ. Roma La Sapienza, Italy** V. Boccone **Switzerland**

M. Pelliccioni, **INFN Frascati & CNAO, Italy** A. Mairani, **CNAO, Italy**

M. Santana, **SLAC, USA** M.C. Morone, **Univ. Roma II, Italy**

I. Rinaldi, K. Parodi, G. Dedes **LMU Munich, Germany and HIT, Heidelberg, Germany**

A. Empl, S. Hoang, J. Idarraga, M. Kroupa, L. Pinsky, B. Reddell, **Houston Univ., USA**

A.A. Bahadori, K.T. Lee, N. Stoffe, E. Semones, N. Zapp, **NASA-Houston, USA**

S. Trovati, **PSI Switzerland** S. Rollet, A. Sipaj **AIT, Austria**

M. Lantz, **Uppsala Univ., Sweden** G. Lukasik, **Poland**

E. Gadioli, M.V. Garzelli, **Italy** L. Sarchiapone, **INFN Legnaro, Italy**

P. Colleoni, **Ospedale Papa Giovanni XXIII Bergamo, Italy**

A. Fontana V.E. Bellinzona **INFN Pavia** Anna Ferrari, S. Mueller **HZDR, Germany**



PLEASE: JOIN THE MAILING LIST, READ THE LICENSE (NON-GPL, MILITARY USE FORBIDDEN, ...)






[Fluka >>](#)
[Documentation >>](#)
[Download](#)
[Tools >>](#)
[Discuss >>](#)
[Team >>](#)

Quick launch:

[Download](#)

[Mailing list](#)

[Manual Online](#)

[Courses](#)

[Flair](#)

[Contact us](#)

A A A


Download and use of the FLUKA package are subject to the conditions put forth in the [FLUKA user license](#). You received a copy of the conditions when you established your personal FLUKA User ID (**FUID**) for use and download of the FLUKA package.

Only registered users can download!
Please enter your **FUID** (i.e. **fuid-XXXX**) as "User Name:"

It is strongly suggested to subscribe to the FLUKA mailing list. The list offers help and informs about new releases and available updated (mainly BUG fixes) versions of the package. To subscribe, simply send email to Majordomo@fisica.unimi.it with a single line of text: **subscribe fluka-discuss**.

Name	Comment	Date	Size
fluka-2011.2-15.i686.rpm	rpm for fluka-2011.2-15	10th of September, 2012	135 MB
flupix-vdi.zip	Auxiliary files for FLUPIX	28th of September, 2009	526 kB
fluka2011.2-linuxAA.tar.gz	FLUKA Package Version 2011.2.15	10th of September, 2012	137 MB

Last version:

[FLUKA](#) 2011.2.15,
September 9th 2012
(last respin)

[FLAIR](#) 1.0.0

News:

Fluka Release (10.08.2012)

[FLUKA](#) 2011.2.14 has been released.

whenever there is a respin (announced on the mailing list)
please update your installation

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
- Maintained and developed under CERN-INFN agreement and copyright 1989-2014
- More than 7000 registered users

Full mixed field capability

<http://www.fluka.org>

Preliminary considerations:

What this course is *not* about:

This is an advanced course, *no detailed instructions* will be given on

- Installing and running the code
- Using the basic Flair features
- Writing/debugging a simple geometry
- Writing/debugging a simple input file
- Use the built-in scoring, and process the results

Moreover, there will be no lecture on the physics embedded in the code.

However, a few reminders / summaries will be provided

What this course is about:

- New features in **FLUKA**
- A bit of the internal structure of the code
- Advanced geometry issues
- Advanced biasing
- User routines, with examples
- And in general how to exploit at best the code

This course relies heavily on the experience of past advanced courses: feedback is welcome!!!

Release / Registration

- ❑ You received a USB stick with **Fluka2014.5pre** (pre-release/alpha version of Fluka2015), that is **not** yet publicly distributed. The public release will occur in 2015
- ❑ If you are not a registered user, **PLEASE REGISTER NOW** on the **fluka** web site
- ❑ This pre-release ***should not be used for scientific work, neither*** results obtained with this pre-release should be used ***for any project, publication or comparison***. Its use is strictly limited at getting a flavour of the new features
- ❑ Please download the public fluka2015 version as soon as it will available on the web site sometimes next year
- ❑ Bugs/questions related to this pre-release version should be addressed to **fluka-discuss@fluka.org** only if strictly necessary, with the tag **[fluka2014]** in the subject

Important notice:

- PEANUT has been extended to cover the whole energy range since several years. The extended PEANUT is ***NOW THE DEFAULT*** at all energies, that is the default is now the equivalent of:

```
PHYSICS 100000. 100000. 100000. 100000. 100000. 100000. PEATHRESH
```

- Coalescence and heavy evaporation are still required to be activated explicitly for precise calculations (hopefully changed for the final release at least for coalescence):

```
PHYSICS 1.
```

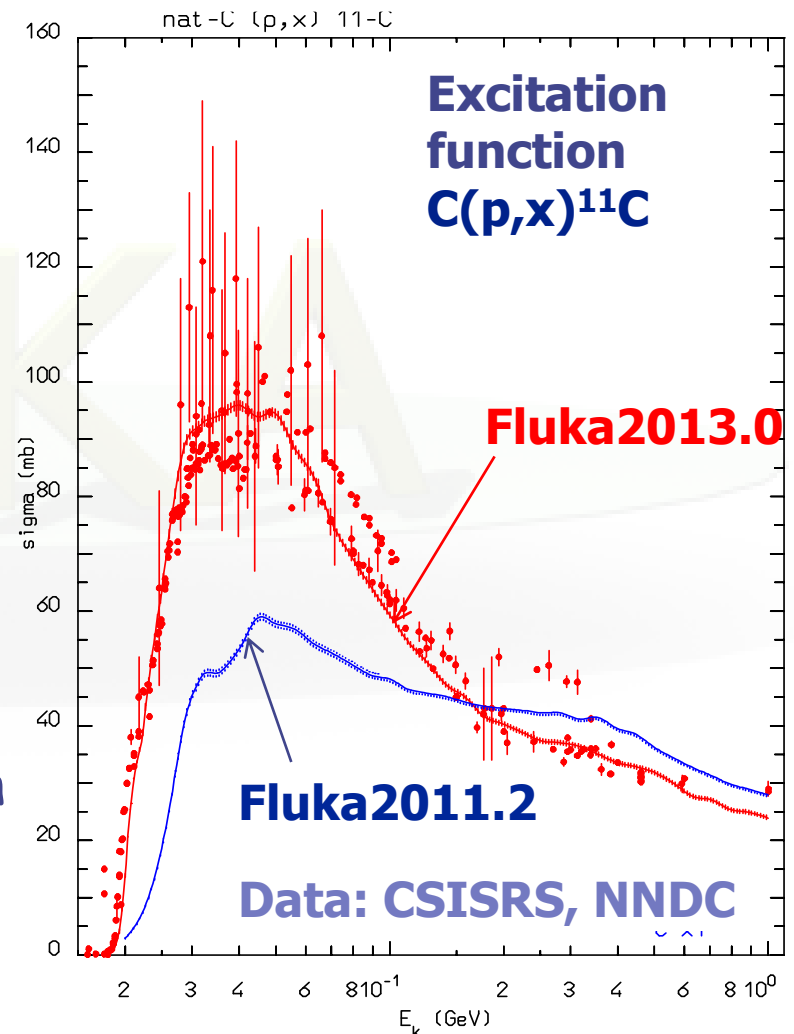
```
COALESCE
```

```
PHYSICS 3.
```

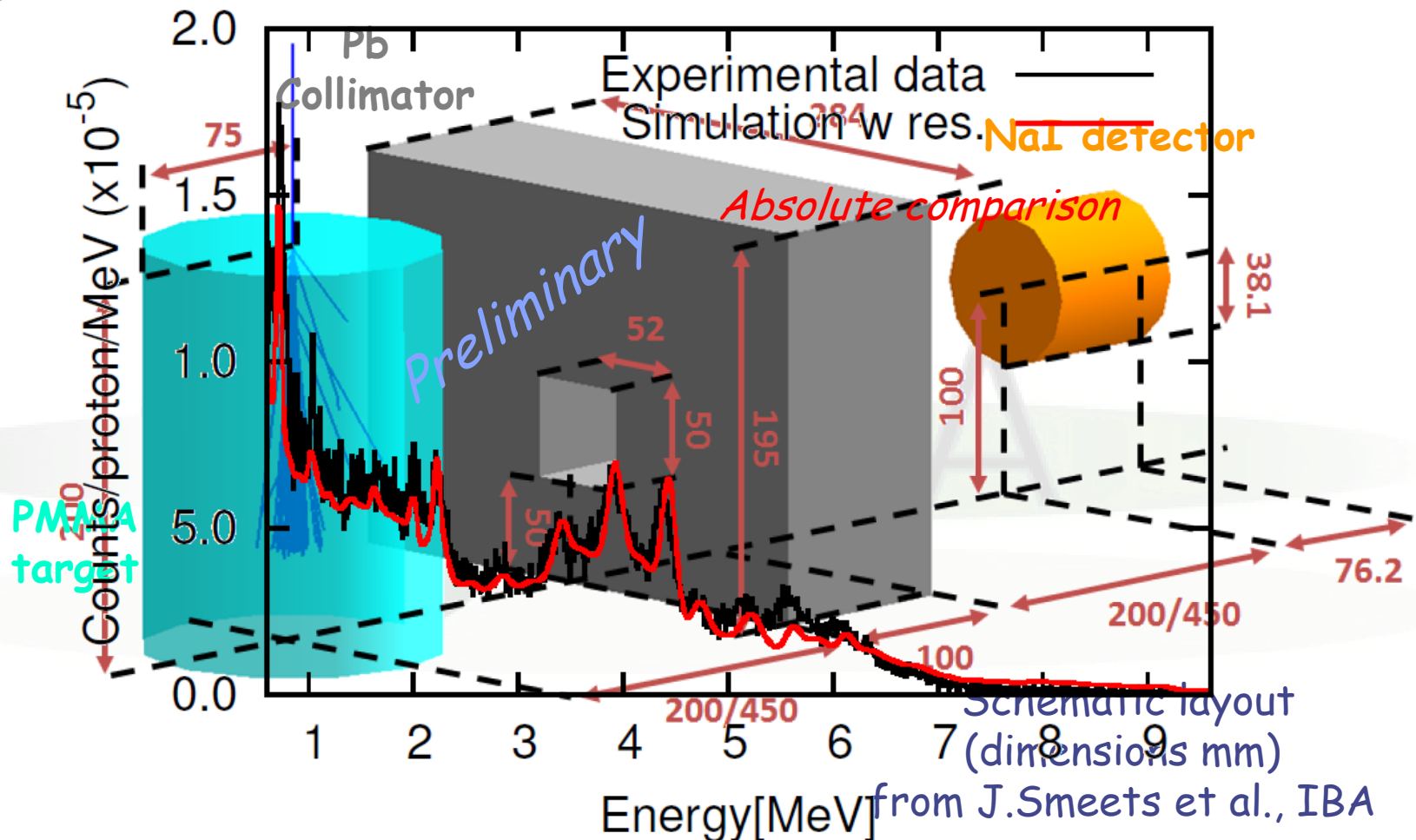
```
EVAPORAT
```


New features in low energy nuclear int.

- Direct (p, α) reactions introduced for light nuclei
- New model for "first interaction" deuteron production for light nuclei
- General cleanup/retuning of PEANUT for p/n at low energies
 - ✓ Example: effect on the production of β^+ emitters, important for medical (PET on line) applications
- De-excitation γ angular distribution according to multipolarity and spin orientation
- Example of prompt photon emission: see next slide



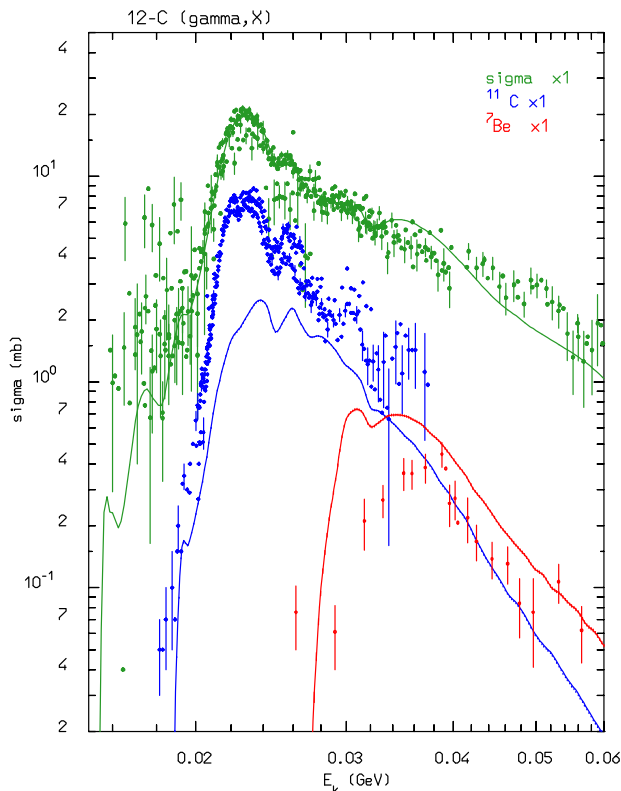
Photon yields by 160 MeV p in PMMA



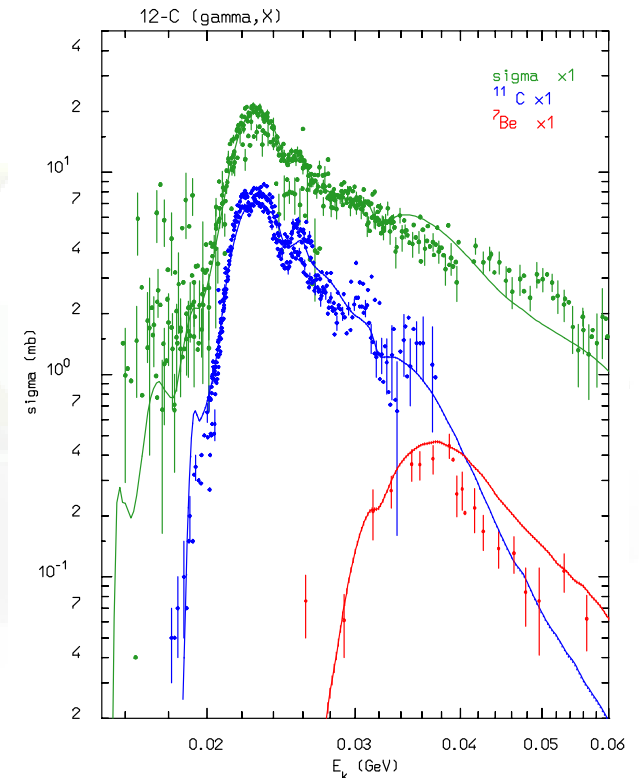
Energy spectrum of "photons" after background subtraction (collimator open - collimator closed) for 160 MeV p on PMMA. FLUKA **red line**, data **black line** (J.Smeets et al., IBA, ENVISION WP3)

New features in Fermi break-up

- Account for initial spin and parity, when known, in Fermi break-up
→ major improvement for (γ, X) reactions on light targets.



12C + γ in GDR
→ $J^\pi = 1^-$
→ 3α and $\alpha + {}^8\text{Be}$
impossible in
 $L=0$
→ Factor 3 on ${}^{11}\text{C}$
production



- Broad 8-Be^* 1st excited state now accounted for in Fermi break-up

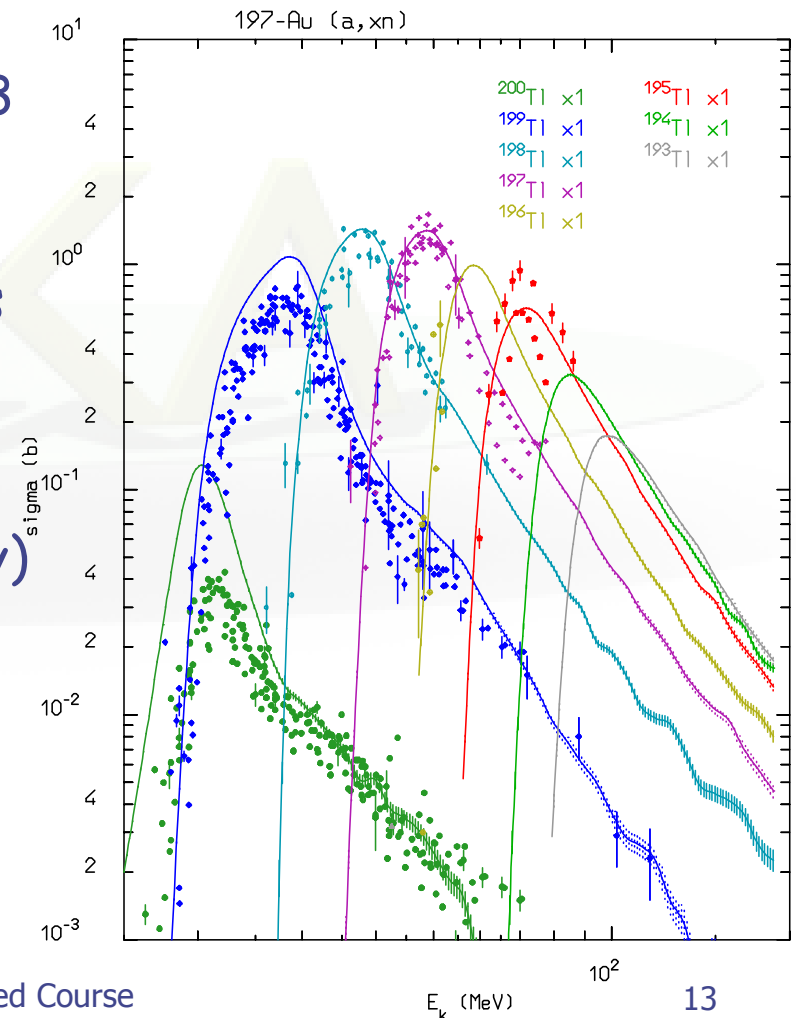
- Emission of α particles in radioactive decays added
- Special decay model for 9-C

- ## 3rd FLUKA Advanced Course

New features in ion-ion generators

- Nuclear discrete levels accounted for in BME, rQMD and DPMJET interactions when considering fragment excitation
- (Initial) extension of BME to $A=3$ projectiles
- BME coupled with Peanut preequilibrium for configurations where the BME native preequilibrium is not available \rightarrow major improvement in the description of alpha (and not only) induced reactions

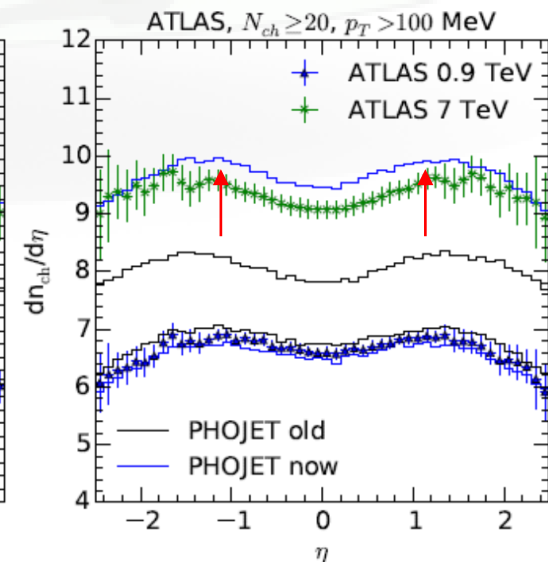
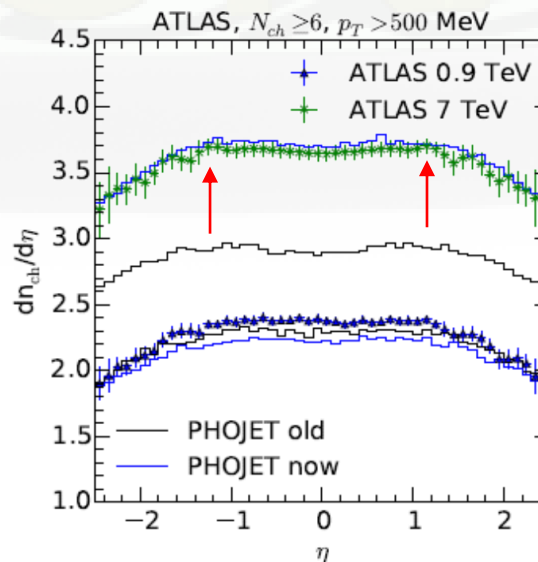
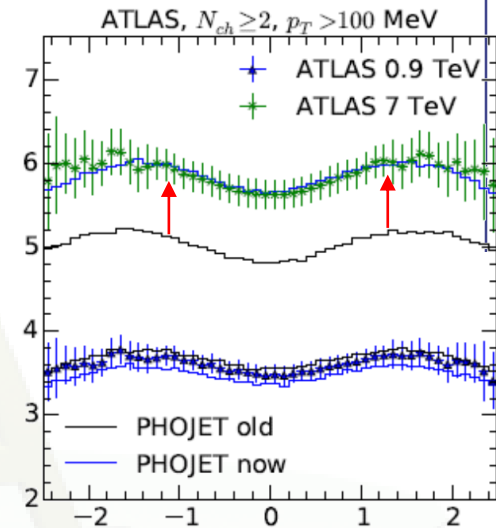
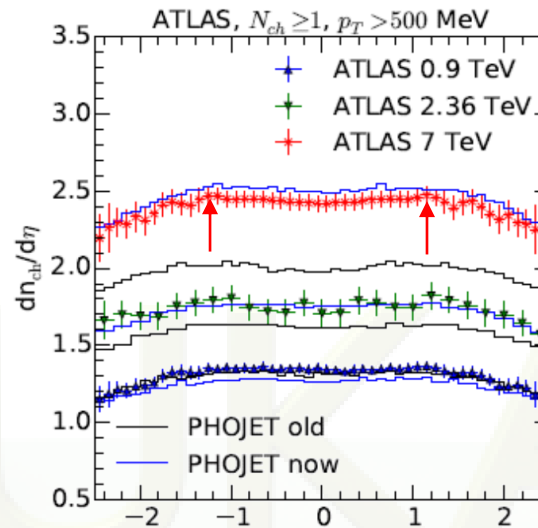
Excitation functions for the production of radioisotopes from α interactions on Au (left) and Pb (right) (Data: CSISRS, NNDC)



New features in Dpmjet-3

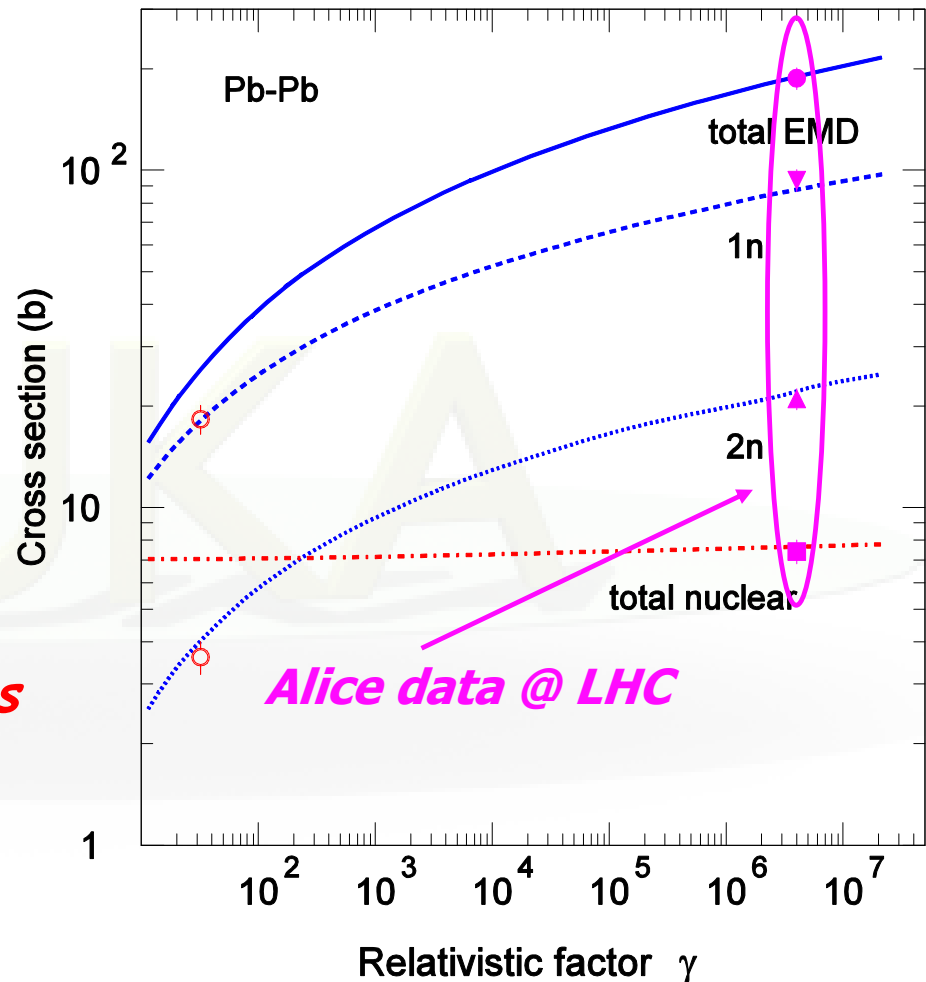
ATLAS - Charged particle pseudorapidity density.

- Dpmjet-3 used for heavy ion interactions above 5 GeV/A, pp and hA above 20 TeV
- Major revision with significant improvement of agreement with LHC data
- New model for high energy p/pbar-p elastic and total cross sections and elastic scattering



New features in EMD

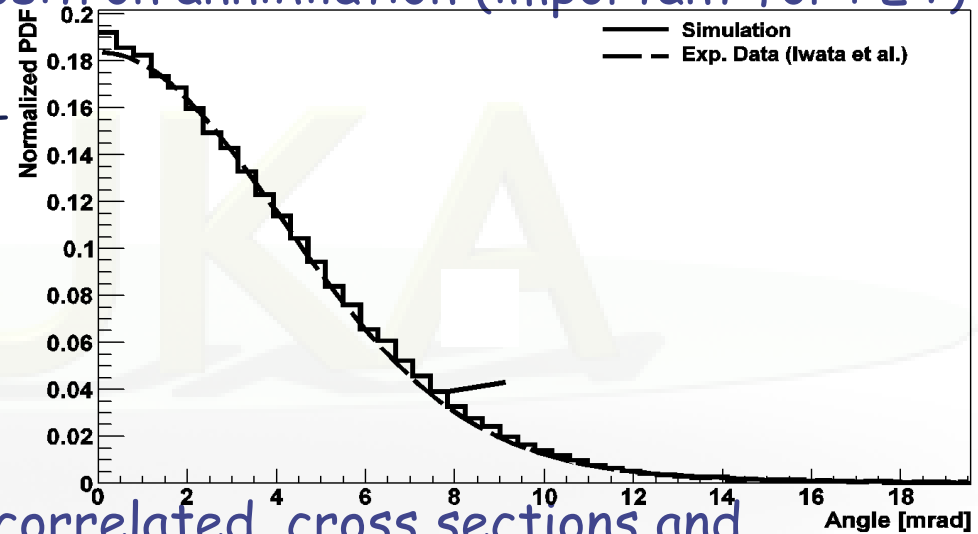
- General improvement of ElectroMagnetic Dissociation (EMD), now including the *E2 multipolarity* → important at low energies and for μ^\pm and e^\pm
- EMD extended to μ^\pm below the γ -N threshold
- EMD extended to e^\pm → *electronuclear interactions* now implemented (see exercise)
- *Deuteron Coulomb dissociation* (stripping). Please note *nuclear stripping not yet implemented*



More new features..

- ❑ Improvement of the transition between the resonance and DPM models for Nucleon-Nucleon interactions
- ❑ Acolinearity model for positron annihilation (important for PET)

Example: deviation from γ - γ co-linearity after positron annihilation in H_2O , FLUKA(line) vs data (histo)



- ❑ Neutron pointwise, fully correlated, cross sections and interactions for 2-H, 3-He and 4-He
- ❑ Synchrotron radiation generation through a new SPECSOUR option (see lecture on sources)
- ❑ Offset dipole and IGRF11 magnetic field tracking for cosmic ray module (see lecture on cosmic rays)

Technical improvements

- Alternate material for radioactive decays extended to support whichever material, not only vacuum
- New **DEFAULT: DAMAGE** for DPA calculations
- **Revised DEFAULTS**, towards better precision in general
- Possibility to **assign** numeric or character **values** to **preprocessor codewords** and then use them in the input file
- Extension of **signed** rotation indexes to lattices and geometry, even in preprocessor parameterized form
- Multiple **DETECT** estimators with arbitrary number of channels
- New generalized estimators for (rough) Single Event Upset estimates: **HEHAD-EQ, THNEU-EQ**

Technical improvements -II

- New generalized estimators: LGH-IONS, HVY-IONS, E+E-GAMM, DOSE-H2O, LOWENNEU, NTLOWENE, ALL-IONS
- Blank common (initial) *dynamic memory allocation* now possible, and further run-time dynamic allocation for usrbins and voxels, obviously on the *gfortran version only*
- Alarm set up to catch TERM signals and stop runs cleanly
- EMD cross sections for the projectile and all charged hadrons/leptons tabulated at initialization (improved speed)

Some reminders:

- ❑ The present version works with the g77 (32 bit) and gfortran compilers (64 bit). For 64 bit computers, if you wish to use the 32 bit, g77, version compatibility packages are required
- ❑ gfortran ≥ 4.6 required, to avoid mis-compilations
- ❑ On Windows, a virtual-machine package can be installed
- ❑ The code is in fortran, as well as all user routines.
- ❑ The high energy heavy ion interaction generators are external, if needed they have to be linked with the program using the **ldpmqmd** script. There are two of them: DPMJET-3 ($E/A > 5$ GeV), rQMD ($E/A > 100$ MeV)
- ❑ The low energy ion interaction generator (BME) is part of the standard Fluka library (no need for specific linking)
- ❑ Units: **GeV, g, cm, second, radian**, with a few exceptions (for instance the Ionization potential is in eV, as well the DPA damage threshold)

Some reminder: the FLUKA input file jargon

The FLUKA input file is an **ascii** file containing the **COMMANDS**

Command:

One keyword, 6 floating point numbers, one keyword

Example (fixed format, FREE format is available as well):

```
*...+...1...+...2...+...3...+...4...+...5...+...6...+...7...+...
BEAM          1.E+04          0.0          0.0          0.0          0.0          0.0PROTON
*
*keyword      momentum mom.spread  diverg.    X-width    Y-width    flag    particle
*            WHAT (1)    WHAT (2)    WHAT (3)    WHAT (4)    WHAT (5)    WHAT (6)    SDUM
```

- We refer to commands also as: **cards**, **options**, **directives**, **definitions**
- We refer to input parameters as **WHAT's**
- Command keywords must be in uppercase, fixed or free format
- Some commands require more than one "card"
- Generally, with few exceptions, the order of commands is irrelevant
- Most commands can be issued several times and each next commands adds information or overrides (in total or in part) the previous ones
- A line with a * character in column 1 is treated as a **comment**
- Nearly always there are default values for WHAT() values!
- Now most of the difficulties in building of the input file are managed by the **FLAIR graphical interface**

Some reminders:

- ❑ The code works under IMPLICIT DOUBLE PRECISION for variables in the range (A-H,O-Z).
 - *Don't forget ..D+/-xx* (eg 2.3D+00, 7.8D-03) in all numerical settings in user routines, and *be careful in passing variables* to/from Fluka or external packages (eg CERNLIB) routines
- ❑ Most mathematical and physical constants are predefined inside the (DBLPRC) include,
 - *use them whenever possible!*
- ❑ Compilation flags are already included in the **fff** script
 - The script should be used for **user routines** as well
- ❑ *Floating point exceptions* are enabled (hard-wired!!) and *dump core* size set to infinity at the start of each run
- ❑ If *high precision input* (> 10 digit) is required, FREE format can be invoked → no limit

Some reminder: the FLUKA particles

The list of particles transported by fluka is in the manual. Each particle is defined by a NAME and a NUMBER.

Only a few heavy ions have a predefined name and a number, the others are defined by A and Z

FLUKA name	FLUKA number	Symbol	Common name	Standard PDG number (Particle Data Group) [120]
4-HELIUM ⁽¹⁾	-6	α	Alpha	—
3-HELIUM ⁽¹⁾	-5	^3He	Helium 3	—
TRITON ⁽¹⁾	-4	^3H	Triton	—
DEUTERON ⁽¹⁾	-3	^2H	Deuteron	—
HEAVYION ⁽¹⁾	-2	—	Generic Heavy Ion (see command HI-PROPE)	—
OPTIPHOT	-1	—	Optical Photon	—
RAY ⁽²⁾	0	—	Pseudoparticle	—
PROTON	1	p	Proton	2212
APROTON	2	\bar{p}	Antiproton	-2212
ELECTRON	3	e^-	Electron	11
POSITRON	4	e^+	Positron	-11
NEUTRIE	5	ν_e	Electron Neutrino	12
ANEUTRIE	6	$\bar{\nu}_e$	Electron Antineutrino	-12

**Beginning
Of the list,
More follows**

There exists also GENERALIZED particles, mostly used for scoring: ex

ALL-NEGA	204	All negative particles			
ALL-POSI	205	All positive particles			
NUCLEONS	206	Protons and neutrons	DPA-SCO	239	Displacements per atoms
NUC&PI+-	207	Protons, neutrons and charged pions	DOSE-EQ	240	Dose Equivalent (pSv) ⁽⁶⁾
ENERGY	208	For dose scoring: Deposited energy			
		For energy fluence scoring: Kinetic energy			

Some reminders: neutrons

- Transport and interactions of neutrons with energies below 20 MeV are handled by a *dedicated library*
- Neutron interactions at higher energy are handled by FLUKA nuclear models
- In the FLUKA jargon neutrons below 20 MeV are called **low energy neutrons**
- The low energy neutron library uses a **multigroup** approach
- About **280 material/temperature** combinations are available
- The library handles also gamma generation, energy deposition by kerma factors, residual nuclei production (now including isomers), secondary neutrons, fission neutrons, and NIEL
- For some isotopes/materials: self shielding, molecular binding, correlated gamma generation, point-wise transport

Reminder: radioactive isotopes

- ❑ In FLUKA, the production, build-up and decay of **radioactive isotopes** can be simulated within the same run
- ❑ Radioactive isotopes can also be used as source particles
- ❑ **Caveat:** the production of **metastable** states is not yet simulated by the Fluka nuclear models*. When radioactive build-up/decay is requested, it is assumed that the initial isotope production is **equally distributed (half-half)** between the ground state and the (possible) metastable state. However, **metastable** states in the subsequent decay chain are populated and decayed according to the correct branching ratios

*Except low-energy neutrons

Events, statistics, normalization

- ❑ In a Monte Carlo code, the result is an **estimator** of the desired quantity, and is obtained as the **average over many trials**
- ❑ Mathematically, the MC treatment is based on the **central limit theorem**
- ❑ The higher the number of trials, the better the **error** on the estimator
- ❑ In particle transport MC, a **trial** is the full history following the **primary event**
- ❑ The primary event may be represented by a single particle in a beam (the most common situation), or by a more complex source event, like for instance the decay of a radioactive isotope or the products of an interaction. Therefore, a primary, or **source**, event may be composed by several particles.
- ❑ The estimators are obtained **averaging** over the number of primary events: therefore all results in FLUKA are given **normalized per primary event**.
- ❑ Nevertheless, **event-by-event** quantities are also available from the code and/or can be accessed through user routines. Event-by-event information is useful when **correlations among estimators** are required.

Reminder: biased and analogue

- ❑ Fluka can run both in **fully analog** and in **biased** mode
- ❑ BIASING techniques allow to **improve the statistical convergence** of results in a **selected region** of the problem phase space (see lecture)
- ❑ This is done using modified distributions, and associating corresponding weights to particles
- ❑ However, the statistical convergence usually worsens in other phase space regions
- ❑ BIASING does NOT reproduce correlations among different components of the same event
- ❑ BIASING may not reproduce fluctuations of physical quantities
- ❑ The type of calculation has to be chosen with care!

Initialization

- ❑ The input cards are parsed according to an optimized ordering different from the order in the input file.
- ❑ Names are converted to numbers for the internal use. The correspondence is kept and is accessible
- ❑ Geometry data are decoded and stored
- ❑ User scoring is decoded, checked, memory space is allocated
- ❑ External data files (cross sections etc) are read in and processed
- ❑ Neutron cross section sets are read in for used materials
- ❑ Tabulations of partial and total cross sections are generated for the materials in use: dE/dx , bremsstrahlung, pair production..
- The energy range and the granularity of these tabulations depend on the energy limits of the problem, essentially on the BEAM card definition and on the production thresholds
- All these quantities, including allocations for scoring, are stored in the Fluka **BLANK COMMON**. Pointers are kept to the different areas.

Consequences

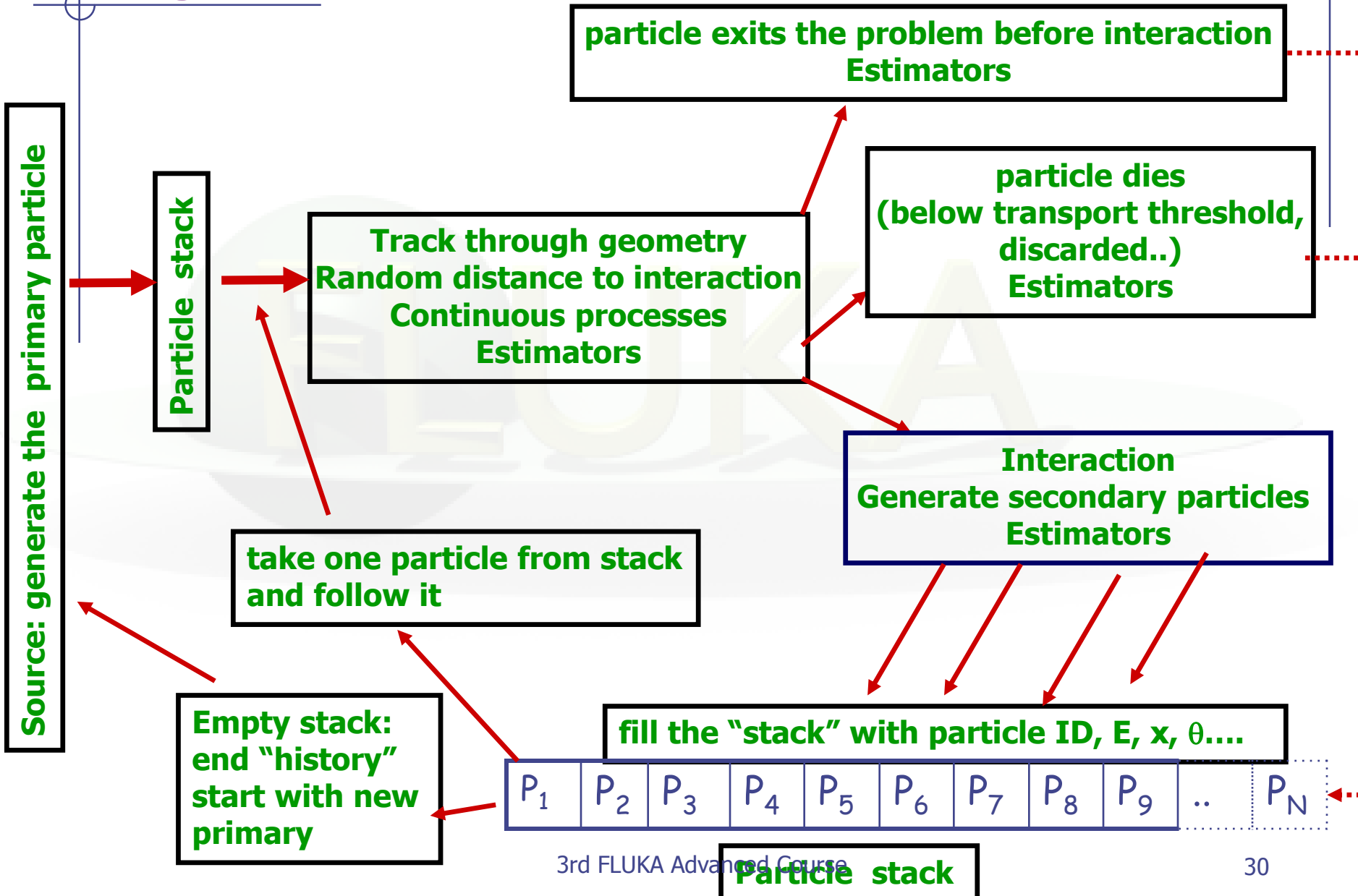
- The TOTAL amount of memory is limited. At present the blank common dimension is about 500 MB: to be kept in mind when asking for estimators *
- The amount of information, thus of memory used, grows with the number of regions and materials used
- User settings have an impact on initialization of physical processes

** Most of the memory limitations disappeared in this (gfortran) pre-release where dynamic memory allocation is introduced with a maximum set a priori by the user, or automatically for scoring and geometry, with a global limit around 4 GB*

Order of input cards parsing

1. DEFAULTS GLOBAL ROT-DEFI TITLE USRGCALL
2. BEAM BEAMAXES BEAMPOS BME DISCARD DPMJET EMF
EVENTYPE GEOBEGIN IONTRANS HI-PROPE MATERIAL
MCSTHRES PART-THR PHYSICS POLARIZAtion RQMD SOURCE
SPECSOUR THRESHOLD
3. COMPOUND RADDECAY RANDOMIZe
4. DETECT WW-FACTO WW-PROFI WW-THRES ASSIGNMat
CORRFACt DCYTIMES DELTARAY ELCFIELD EMF-BIAS EMFCUT
EMFFIX EMFFLUO EXPTRANS FLUKAFIX IONFLUCT IRRPROFI
LAM-BIAS LOW-BIAS LOW-DOWN LOW-MAT MAT-PROP
MGNFIELD MULSOPT MUPHOTON OPT-PROD OPT-PROP
PAIRBREM PHOTONUC STERNHEI
5. EVENTBIN EVENTDAT RESNUCLEi SCORE TIME-CUT USERDUMP
USERWEIG USRBDX USRBIN USRCOLL USRTRACK USRYIELD
6. AUXSCORE DCYSCORE ROTPRBIN TCQUENCH
7. PLOTGEOM START
8. USROCALL STOP

A generic MC scheme



The stack, secondary particles, tracks

- ❑ The properties of all the particles to be tracked are stored in the "stack": /flkstk/ in '(FLKSTK)'
- ❑ NPFLKA counts the particles on stack
- ❑ The *kaskad* routine loops on NPFLKA until the stack is empty, going from bottom (npflka) to top (1)
- ❑ The "current particle" properties are copied from the stack to the TRACKR common, and updated during tracking
- ❑ At each interaction, secondaries are first stored in temporary stacks (GENSTK, FHEAVY..), then loaded onto the main stack. The primary particle, if surviving, is loaded on the stack exactly like the others
- ❑ The particle on top of the stack is followed *first*, generally it is the less energetic, → avoid stack explosion
- ❑ The treatment of the stack for EM particles is slightly different, due to historical reasons EM secondaries are kept on the EMF particle stack, which is emptied before the normal stack

Consequences

- Steps related to the same particle track will (almost) always be non-consecutive in the program flow.
- Primary particles lose their identity as soon as an interaction occurs (this is physical!)
- Therefore, “follow a particle track” may be not straightforward
- However, a “track number” is associated to each “new” particle and it is propagated to the stack and the TRACKR common (see dedicated lecture)
- Moreover, the generation level of each particle is recorded

Main loop

- The loop on events is controlled by the **FEEDER** routine.
 - It checks for run termination conditions (number of primaries)
 - Calls the standard fluka **source**(s) or the user source,
 - May call the **SODRAW**, user routine
 - Then gives the control to the **KASKAD** routine
- **KASKAD** keeps the control until the stack is empty. It handles directly the tracking of hadrons, ions and muons, while it dispatches
 - E.M particles to **KASEMF**,
 - Optical photons to **KASOPH**
 - Low energy neutrons to **KASNEU**
 - Heavy particles to **KASHEA** if approximate treatment is asked for
- Tracking is performed in **steps**, limited by
 - Maximum percentage energy loss in a step
 - Boundary crossing
 - Interaction probability (elastic, non-elastic, δ rays ...)
 - Decay probability

Discrete or continuous processes

During, and at the end, of a step: **discrete** and **continuous** processes

- **Continuous:** energy deposition by Ionization, bremsstrahlung, and pair production (below explicit production thresholds)
- **Continuous:** multiple scattering, deflection by magnetic field
- **Discrete:** interaction (including low energy neutron ones), particle decay, δ ray production, radioactive decay...
- **Discrete:** track termination conditions, such as time cutoff, energy cutoff, escape in the black hole, boundary crossing
- Estimators can be activated for each of these processes, either built-in, or through user routines
- **Tricky:** energy deposition by recoil nuclei after elastic and inelastic reactions (with some settings) and energy deposition by low-energy neutron reaction products (with exceptions..) are treated as **discrete** events.

Biasing

- At every interactions/boundary crossing/ step biasing is applied if required.
- If necessary, the particle weight is modified, and stored in TRACKR and propagated to the stack.
- Particle weights are automatically taken into account by built-in estimators
- User scoring routines must take care of proper weight handling.

FLUKA provides a large variety of biasing techniques. A proper understanding of their use is essential for many shielding (and not only) problems!



end