FLUKA
Advanced Scoring

FLUKA Advanced Course

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 (some of which mentioned afterwards), 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, particle 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

A Reminder on Flux/Fluence/Current

Reaction Rate and Cross Section [1/3]

- We call mean free path $\lambda [cm]$ the average distance travelled by a particle in a material before an interaction. Its inverse, $\Sigma\ [cm^{-1}]$ is the probability of interaction per unit distance, and is called macroscopic cross section. Both λ and Σ depend on the material and on the particle type and energy.

- For N identical particles, the number of reactions R occurring in a given time interval will be equal to the total distance travelled l times the probability per unit distance Σ: $R = l\Sigma$

- The reaction rate will be $\dot{R} = \mathrm{d}l/\mathrm{d}t\ \Sigma = v\Sigma$, where v is the average particle velocity.

Reaction Rate and Cross Section [2/3]

- Assume now that $n(\mathbf{r},v)=dN/dV$ $[cm^{-3}]$ be the density of particles with velocity $v=dl/dt$ $[cm/s]$, at a spatial position $\mathbf{r}$. The reaction rate inside the volume element $\mathrm{d}V$ will be: $\mathrm{d}\dot{R}/\mathrm{d}V = n(\mathbf{r},v)v\Sigma$

- The quantity $\dot{\Phi}(\mathbf{r},v) = n(\mathbf{r},v)v$ is called fluence rate or flux density and has dimensions $[cm^{-3}\ cm\ t^{-1}]=[cm^{-2}\ t^{-1}]$.

- The time integral of the flux density $\Phi(\mathbf{r},v) = n(\mathbf{r},v)\mathrm{d}l$ is the fluence $[cm^{-2}]$

- Fluence is measured in particles per cm^2 but in reality it describes the density of particle tracks

- The number of reactions inside a volume V is given by the formula: $R = \Sigma\Phi V$ (where both Σ and Φ are integrated over energy or velocity)

Reaction Rate and Cross Section [3/3]

- Dividing the macroscopic cross section by N_0, the number of atoms per unit volume, one obtains the microscopic cross section σ [barn=10^{-24}cm^2].

$$\frac{\text{probability/cm}}{\text{atoms/cm}^3} = \frac{\text{probability} \times \text{cm}^2}{\text{atom}} = \text{atom effective area}$$

- i.e., the area of an atom weighted with the probability of interaction (hence the name "cross section").

- But it can also be understood as the probability of interaction per unit length, with the length measured in atoms/cm^2 (the number of atoms contained in a cylinder with a 1 cm^2 base).
- In this way, both microscopic and macroscopic cross section are shown to have a similar physical meaning of "probability of interaction per unit length", with length measured in different units. Thus, the number of interaction can be obtained by both by multiplying by the corresponding particle track-length.

Fluence estimation [1/2]

- Track length estimation:

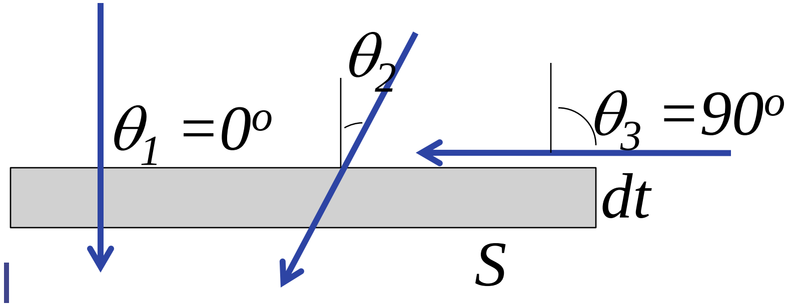
$$\dot{\Phi}(v)\,dt = n(v)\,v\,dt = \frac{dN(v)}{dV}\frac{dl(v)}{dt}dt = \lim_{\Delta V \to 0}\frac{\sum_i l_i(v)}{\Delta V}$$

- Collision density estimation:

$$\dot{\Phi}(v) = \frac{\dot{R}(v)}{\sigma(v)\,N_o} = \frac{\dot{R}(v)}{\Sigma(v)} = \dot{R}(v)\lambda(v)$$

Fluence estimation [2/2]

Surface crossing estimation



- Imagine a surface having an infinitesimal thickness dt
  A particle incident with an angle θ with respect to the normal of the surface S will travel a segment $dt/cos\theta$.
- Therefore, we can calculate an average surface fluence by adding $dt/cos\,\theta$ for each particle crossing the surface, and dividing by the volume $S\,dt$

$$\Phi = \lim_{dt\to 0} \frac{\sum_i \frac{dt}{\cos\theta_i}}{S\,dt}$$

- While the current J will be to count the number of particles crossing the surface divided by the surface

$$J = dN/dS$$

The fluence is independent from the orientation of surface S, while the current is NOT!

In an isotropic field can be easily seen that on a flat surface $J = \Phi/2$

Built-in Conversions and AUXSCORE

Scoring Cards (see Beginner's Course)

- SCORE scores energy deposited (or star density) in all regions
- 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
- RESNUCLEi scores residual nuclei in a given region

Scoring cards (most explained in the following slides)

- EVENTBIN is like USRBIN, but prints the binning output after each event instead of an average over histories
- ROTPRBIN sets the storage precision (single or double) and assigns rotations/translations for a given user-defined binning (USRBIN or EVENTBIN)
  - linked to lecture about the use of LATTICE
- USERDUMP defines the events to be written onto a “collision tape” file
  - see **mgdraw** part later in this lecture
- AUXSCORE defines filters and conversion coefficients
- TCQUENCH sets scoring time cut-offs and/or Birks quenching parameters for binnings (USRBIN or EVENTBIN) indicated by the user
- DETECT scores energy deposition in coincidence or anti-coincidence with a trigger, separately for each “event" (primary history)
  - dedicated post-processing routine is now available

Lattice Related Scorings

EVENTBIN or USRBIN with WHAT(1)=**8** :

Special user-defined 3D binning. Two variables are discontinuous (*e.g.*, region number), the third one is continuous, but not necessarily a space coordinate.

Variable	Type	Default	Override Routine
1^{st}	integer	region number	MUSRBR
2^{nd}	integer	lattice cell number	LUSRBL
3^{rd}	float	Before used as η, now set to zero*	FUSRBV

* In the past it scored: $n=-ln(tan(0.5\ arctan(sqrt(x^2+y^2)/z))$

ROTPRBIN can assign rotations/translations (as defined by ROT-DEFI) for a given user-defined binning (USRBIN or EVENTBIN):

- this allows *e.g.*, defining a 'normal' scoring around a prototype and then 'replicating' the scoring to the respective lattices

"FILTER" : AUXSCORE

WARNING!!
all energy deposition
<u>ionization+NIEL</u> by the selected particle

There is the possibility to filter the estimators, restricting the scoring to a selected subset of particles.

For instance: USRBIN energy deposition by muons only

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

Assign the "muons" filter to the USRBIN estimator named TargEne

Another example: score the yield of 56-Iron ions (very useful: there is no separate name for each ion specie, except light ones. HEAVYION score all isotopes heavier than alpha's together!)

```
USRYIELD    124.0  ALL-PART   -87.    TARGS3     INAIR      1.0 Fe56
USRYIELD    180.0       0.0    18.      10.0       0.0       3.0   &
AUXSCORE USRYIELD -5602600.             Fe56       Fe56
```

The requested ion is coded in what(2) according to its **A**, **Z** and (optionally) isomeric state **m**:

what(2) = - (100***Z** + 100000***A** + **m***100000000)

with 0==all , i.e. 2600 == all Iron isotopes

Dose-Equivalent (not Dose)

For some quantities, there is the possibility to get built-in conversions, without the need for user routines: done with generalized particles, the most commonly used is dose equivalent (ambient dose equivalent or effective dose):

DOSE-EQ Dose Equivalent [pSv]

DOSEQLET Dose Equivalent LET (Q(LET) according to ICRP60)

!!!! Different to !!!:

(DOSE total absorbed dose in (obviously…) GeV/g!
DOSE-EM dose as above (electromagnetic part only))

The set of conversion coefficients used to calculate DOSE-EQ can be selected by the user among a list (see manual) with AUXSCORE:

```
USRBIN      11.0    DOSE-EQ    -40.0      10.0              15.0 TargDEQ
USRBIN       0.0                -5.0     100.0             200.0 &
AUXSCORE USRBIN                        TargDEQ TargDEQ           AMB74
```

Scores equivalent dose by folding the particle fluences with the “AMB74” conversion coefficients

WARNING : DOSE-EQ no coefficients available for heavy ions (ok for DOSEQLET) !!!

Available Conversion Coefficients

The following dose conversion coefficients sets are available:

1) Effective dose sets from ICRP74 and Pelliccioni data calculated with ICRP radiation weighting factors Wr
   (a) **EAP74** : Anterior-Posterior irradiation
   (b) **ERT74** : Rotational irradiation geometry
   (c) **EWT74** : WORST possible geometry for the irradiation
2) Effective dose sets from ICRP74 and Pelliccioni data calculated with the Pelliccioni radiation weighting factors Wr
   (a) **EAPMP** : Anterior-Posterior irradiation
   (b) **ERTMP** : Rotational irradiation geometry
   (c) **EWTMP** : WORST possible geometry for the irradiation
3) Ambient dose equivalent from ICRP74 and Pelliccioni data
   (a) **AMB74** : **[Default]**
4) Ambient dose equivalent with old "GRS"-conversion factors
   (a) **AMBGS**

(**see backup slides for details**)

Radiation Damage to Electronics

- All important quantities to estimate risks of damage to electronics can be directly scored in FLUKA (***see Materials lecture***):

  Cumulative damage:

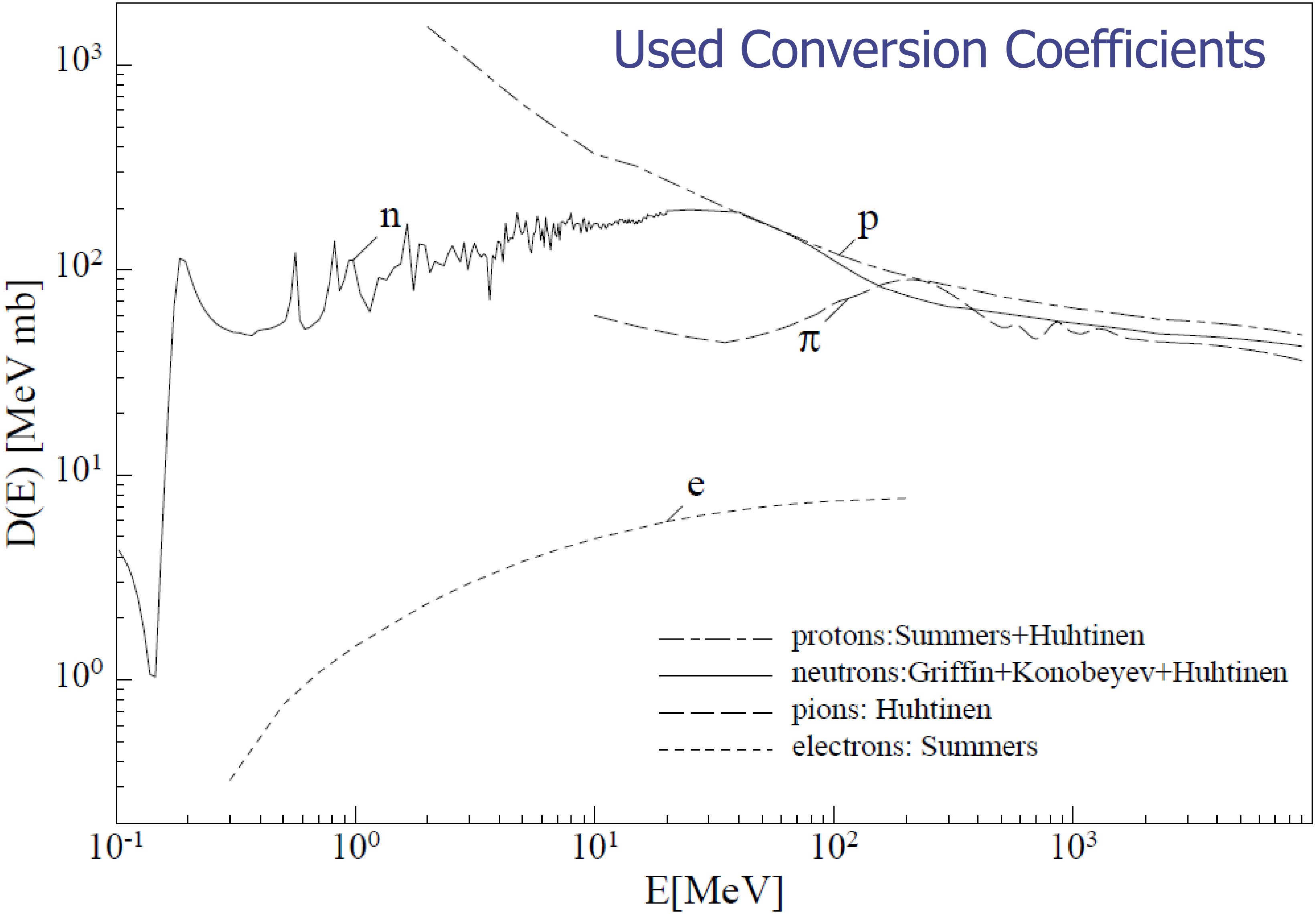
  - Energy deposition (total dose) by scoring DOSE with any 'energy deposition like estimator' (*e.g.*, USRBIN)
  - Si Lattice displacement (1-MeV neutron equivalent particle fluxes) with any 'fluence like estimator' (*e.g.*, USRTRACK)

  Stochastic failures (SEU):

  - "high" energy hadron fluences ("E>20 MeV") with any 'fluence like estimator' (*e.g.*, USRTRACK)
    (**the option of special threshold functions [user defined] is currently in development and will be included in the next release together with the scoring related to the "damage by thermal neutrons"**)

- The powerful FLUKA scoring options together with the analysis of particle energy spectra allows a detailed study in order to select best possible locations for electronics or efficiently design shielding implementations

1MeV Neutron Equivalent



Electronic Damage - Related Scoring

DOSE	total absorbed dose in (obviously...) GeV/g!
SI1MEVNE	Silicon 1 MeV-neutron equivalent fluence
HADGT20M	Hadrons fluence with energy > 20 MeV

- USRTRACK scores average $d\Phi/dE$ (differential fluence) in a given region (SI1MEVNE, HADGT20M or any particle type)
- USRBDX scores for the same quantities average $d^2\Phi/dEd\Omega$ (double-differential fluence or current) on a given surface (between two regions)
- USRBIN scores the spatial distribution either of deposited dose, or fluence (1MeV or 20MeV) in a regular mesh (cylindrical or Cartesian) described by the user
- USRBIN also scores the same quantites on a region basis

```
* 1) high-energy hadron fluence spectrum
USRTRACK          -1.   HADGT20M      -31.   RADMON1      125.       170.Ust20MeV
USRTRACK          1D3      1D-14                                              &
* 2) displacement damage spectrum
USRBDX            98.   SI1MEVNE      -41.      TAIR   RADMON1       150.Usx1MeV
USRBDX            1D3      1D-14      170.                                    &
* 3) dose distribution in a regular mesh through the geometry
USRBIN            10.       DOSE      -21.      100.       20.       200.UsbDose
USRBIN          -100.       -20.     -100.      100.       20.       150.&
* 4) integrated high-energy hadron fluence on a region basis
USRBIN           18.0   HADGT20M     -37.0    LSTREG     300.0   10000.0UsbReg20
USRBIN         FSTREG        0.0  -10000.0       1.0       1.0       1.0    &
```

Displacement Damage / Charge Collection

For all charged particles and Heavy Ions FLUKA calculates the recoil as a normal particle. During transport it calculates the restricted and unrestricted nuclear stopping power, allowing to score dpa's and non-ionizing energy loss (NIEL):

NIEL-DEP	Non Ionizing Energy Loss deposition
DPA-SCO	Displacements per atoms

(details see **Material's Lecture**)

In addition (not necessarily linked to displacement damage) the following can be useful in order to get the net charge deposition in a given region:

NET-CHRG	Net Charge

Activation Scoring (Reminder!)

Input card: RADDECAY

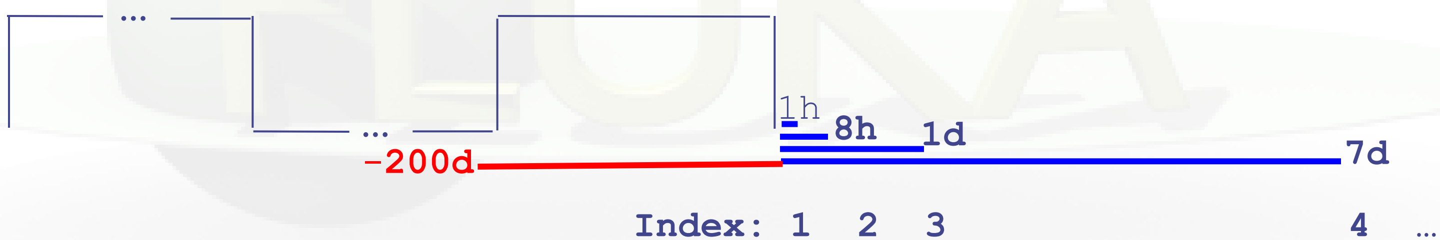
requests simulation of decay of produced radioactive nuclides and allows to modify biasing and transport thresholds (defined with other cards) for the transport of decay radiation

Input card: IRRPROFI

definition of an irradiation profile (irradiation times and intensities)

Input card: DCYTIMES

definition of decay (cooling) times



Input card: DCYSCORE

associates scoring detectors (radio-nuclides, fluence, dose) with different cooling times

Input card: AUXSCORE

allows to associate scoring estimators with dose equivalent conversion factors or/and to filter them according to (generalized) particle identity

Before: Change Of 'Geometry' ->Two-Step Method

Two separate FLUKA simulations were necessary in order to change the geometry between the prompt and the decay part (see *e.g.*, CERN FLUKA Course for explanation of use [S. Roesler]):

1st step

- simulation of production of radioactive nuclides and of their build-up and decay for a certain irradiation pattern and different cooling times
- write-out of all information on produced radio-nuclides at each cooling time into external file via user-routine **usrrnc.f**
- uses the analytical solution of the Bateman equation in FLUKA (*i.e.*, radioactive build-up and decay identical to 1-step method)

2nd step

- simulation(s) of radioactive decay and transport of decay radiation
- information on radio-nuclides read in from file created in 1st step via user-routine **source.f**
- individual simulations for each requested cooling time

Now: Change Of 'Geometry'

The latest FLUKA version (this course) contains the possibility of selectively changing regions to vacuum/blackhole and/or switching on/off possible fields) when transporting radioactive decay products. Radioactive decay products originating from regions switched to vacuum/blackhole are ignored. This is helpful for situations where the emissions of an activated object in a complex environment have to be evaluated standalone.

Through Input card: ASSIGNMA

(a (single-element or compound) material is assigned to each geometry region)

Example

```
*...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+...
ASSIGNMA       GOLD      REG1      REG2        1.0       0.0
```

MATERIAL from REGION to REGION in steps of

WHAT(5) = 1/2/3: a magnetic/electric/both field(s) is(are) present in the region(s) defined by WHAT(2), (3), and (4), for both prompt and radioactive decay products

= 4/5/6: same as above, but for prompt products only

= 7/8/9: same as above, but for radioactive decay products only

Electr/magn. field is present, either in both/prompt or decay part (material set to vacuum in the other case!

Note: so far distinction between lattices (all instances are affected)

Interception through User Routines

Routines Linked to the FLUKA Output

in association to FLUKA output

- comscw.f ... weighting energy deposition or star production
- fluscw.f ... weighting fluence, current and yield
- mgdraw.f ... general scoring tracking interface
- usrrnc.f ... isotope production and possible conversion
- endscp.f ...
- fldscp.f ...
- musrbr.f ...
- lusrbl.f ...
- fusrbv.f ...

Possibly Related:
(Intercepting Particle Stack)

- mdstck.f
- stupre.f
- stuprf.f

$\Big\}$ **See Lecture User-Routines**

comscw.f

(weighting energy deposition or star production)

Argument list (all variables are input only)

IJ	:	particle type (1 = proton, 8 = neutron, etc.: see code in 5.1)
XA,YA,ZA	:	current particle position
MREG	:	current geometry region
RULL	:	amount to be deposited (unweighted)
LLO	:	particle generation
ICALL	:	internal code calling flag (not for general use)

Activated by option USERWEIG with WHAT(6) > 0.0. Energy and stars obtained via SCORE, USRBIN and EVENTBIN, and production of residual nuclei obtained via RESNUCLEi are multiplied by the value returned by this function. The user can implement any desired logic according to the argument list (particle type, position, region, amount deposited, particle generation), or information available in COMMON SCOHLP (binning number, type of scored quantity). The scored quantity is given by the flag ISCRNG (in SCOHLP):

ISCRNG = 1 → Energy density binning
ISCRNG = 2 → Star density binning
ISCRNG = 3 → Residual nuclei scoring
ISCRNG = 4 → Momentum transfer
ISCRNG = 5 → Activity density binning
ISCRNG = 6 → Net charge density

comscw.f – Tips & Tricks

The binning/detector number is given by JSCRNG (in SCOHLP) and is printed in output between the estimator type and the detector name.

Note that a detector of residual nuclei can have the same JSCRNG number as a binning (use the value of ISCRNG to discriminate).

Further information can be obtained including COMMON TRACKR (for instance particle's total energy, direction cosines, age). TRACKR contains also special user variables (both integer and in double precision) which can be used to save information about particles which have undergone some particular event.

If data concerning the current material are needed, it can be accessed as MEDIUM(MREG) if (FLKMAT) is included.

fluscw.f

(weighting fluence, current and yield)

Argument list (all variables are input only)

IJ :	particle type
PLA :	particle momentum (if > 0.0) or -PLA = kinetic energy (if < 0.0)
TXX, TYY, TZZ :	particle current direction cosines
WEE :	particle weight
XX, YY, ZZ :	particle position
NRGFLK :	current region (after boundary crossing)
IOLREG :	previous region (before boundary crossing). Useful only with boundary crossing estimators (for other estimators it has no meaning)
LLO :	particle generation
NSURF :	internal code calling flag (not for general use)

Similar to COMSCW. Function FLUSCW is activated by option USERWEIG, with WHAT(3) > 0.0. Yields obtained via USRYIELD, fluences calculated with USRBDX, USRTRACK, USRCOLL, USRBIN, and currents calculated with USRBDX are multiplied by the value returned by this function.

fluscw.f - Tips & Tricks

The user can implement any desired logic according to the argument list (particle type, energy, direction, weight, position, region, boundary, particle generation), or information available in COMMON SCOHLP (binning or detector number, estimator type). The estimator type is given by the flag ISCRNG (in COMMON SCOHLP):

ISCRNG = 1 → Boundary crossing estimator
ISCRNG = 2 → Track-length binning
ISCRNG = 3 → Track-length estimator
ISCRNG = 4 → Collision density estimator
ISCRNG = 5 → Yield estimator

Useful (*e.g.*, in fluscw/comscw)

When interception of regions is required, thus the conversion of region name to number (or opposite, but rarely) might be important:

CALL GEON2R) (REGNAM, NREG, IERR)

<u>Input variable:</u>

Regnam = region name (CHAR*8)

<u>Output variables:</u>

Nreg = region number

Ierr = error code (0 on success, 1 on failure)

Conversion of region number to name

CALL GEOR2N (NREG, REGNAM, IERR)

<u>Input variable:</u>

Nreg = region number

<u>Output variables:</u>

Regname = region name (CHAR*8)

Ierr = error code (0 on success, 1 on failure)

mgdraw.f

(General interface to FLUKA transport and scoring)

Argument list (all variables are input only)

ICODE	:	FLUKA physical compartment originating the call
		= 1: call from subroutine KASKAD (hadrons and muons)
		= 2: call from subroutine EMFSCO (e^-, e^+ and photons)
		= 3: call from subroutine KASNEU (low-energy neutrons)
		= 4: call from subroutine KASHEA (heavy ions)
		= 5: call from subroutine KASOPH (optical photons)
MREG	:	current region

See Lecture Code-Structure

Subroutine MGDRAW, activated by option USERDUMP with WHAT(1) ≥ 100.0, usually writes a "collision tape", *i.e.*, a file where all or selected transport events are recorded. The default version (unmodified by the user) offers several possibilities, selected by WHAT(3) in USERDUMP.

mgdraw.f

The different ENTRY points of MGDRAW

Additional flexibility is offered by a user entry USDRAW, interfaced with the most important physical events happening during particle transport.

The user can modify also any other entry of this subroutine:

BXDRAW called at boundary crossings,
EEDRAW called at event end,
MGDRAW called at each step, for trajectory drawing and dE/dx energy deposition events,
ENDRAW for recording of point energy deposition events,
SODRAW for recording of source events

mgdraw.f: the SODRAW entry

Argument list
No arguments

SODRAW writes by default, for each source or beam particle:

NCASE:	(in **COMMON CASLIM**, with a minus sign to identify SODRAW output) number of primaries followed so far
NPFLKA:	(in **COMMON FLKSTK**) stack pointer
NSTMAX:	(in **COMMON FLKSTK**) highest value of the stack pointer encountered so far
TKESUM:	(in **COMMON SOURCM**) total kinetic energy of the primaries of a user written source, if applicable. Otherwise = 0.0
WEIPRI:	(in **COMMON SUMCOU**) total weight of the primaries handled so far

NPFLKA times:	ILOFLK:	type of source particle
(all variables in	TKEFLK + AM:	total particle energy (kinetic+mass)
COMMON FLKSTK)	WTFLK:	source particle weight
	XFLK, YFLK, ZFLK:	source particle position
	TXFLK, TYFLK, TZFLK:	source particle direction cosines

mgdraw.f: the MGDRAW entry

MTRACK: number of energy deposition events along the track
JTRACK: type of particle
ETRACK: total energy of the particle
WTRACK: weight of the particle
NTRACK: values of XTRACK, YTRACK, ZTRACK: end of each track segment
MTRACK: values of DTRACK: energy deposited at each deposition event
CTRACK: total length of the curved path

Other variables are available in TRACKR (but not written by MGDRAW unless the latter is modified by the user: particle momentum, direction cosines, cosines of the polarisation vector, age, generation, etc. see a full list in the comment in the INCLUDE file).

mgdraw.f: the ENDRAW entry

Called at point-like Energy Deposition dumps

(for example: stopping particles, photoelectric eff., etc.)

Argument list (all variables are input only)

ICODE : type of event originating energy deposition
ICODE = 1x: call from subroutine KASKAD (hadrons and muons);
= 10: elastic interaction recoil
= 11: inelastic interaction recoil
= 12: stopping particle
= 14: particle escaping (energy deposited in blackhole)
ICODE = 2x: call from subroutine EMFSCO (electrons, positrons and photons)
= 20: local energy deposition (i.e. photoelectric)
= 21 or 22: particle below threshold
= 23: particle escaping (energy deposited in blackhole)
ICODE = 3x: call from subroutine KASNEU (low-energy neutrons)
= 30: target recoil
= 31: neutron below threshold
= 32: neutron escaping (energy deposited in blackhole)
ICODE = 4x: call from subroutine KASHEA (heavy ions)
= 40: ion escaping (energy deposited in blackhole)
ICODE = 5x: call from subroutine KASOPH (optical photons)
= 50: optical photon absorption
= 51: optical photon escaping (energy deposited in blackhole)
MREG : current region
RULL : energy amount deposited
XSCO, YSCO, ZSCO : point where energy is deposited

mgdraw.f: the BXDRAW entry

Called at Boundary Crossings

Argument list (all variables are input only)

`ICODE` :	physical compartment originating the call, as in the `MGDRAW` entry
`MREG` :	region from which the particle is exiting
`NEWREG` :	region the particle is entering
`XSCO, YSCO, ZSCO` :	point where the boundary crossing occurs

mgdraw.f: the USDRAW entry

USDRAW is called after each particle interaction (requested by the user with option USERDUMP, WHAT(4) ≥ 1.0)

Argument list (all variables are input only)

ICODE : type of event

ICODE = $10x$: call from subroutine KASKAD (hadron and muon interactions);
= 100: elastic interaction secondaries
= 101: inelastic interaction secondaries
= 102: particle decay secondaries
= 103: delta ray generation secondaries
= 104: pair production secondaries
= 105: bremsstrahlung secondaries

ICODE = $20x$: call from subroutine EMFSCO (electron, positron and photon interactions)
= 208: bremsstrahlung secondaries
= 210: Møller secondaries
= 212: Bhabha secondaries
= 214: in-flight annihilation secondaries
= 215: annihilation at rest secondaries
= 217: pair production secondaries
= 219: Compton scattering secondaries
= 221: photoelectric secondaries
= 225: Rayleigh scattering secondaries

ICODE = $30x$: call from subroutine KASNEU (low-energy neutron interactions)
= 300: neutron interaction secondaries

ICODE = $40x$: call from subroutine KASHEA (heavy ion interactions)
= 400: delta ray generation secondaries

MREG : current region

XSCO, YSCO, ZSCO : interaction point

mgdraw.f: the EEDRAW entry

Called at Event End

Argument list (all variables are input only)		
ICODE	:	physical compartment originating the call, as in the MGDRAW entry

mgdraw.f - Tips & Tricks

- An interesting aspect of the routine is that the six entries (all of which, if desired, can be activated at the same time by setting USERDUMP with WHAT(3) = 0.0 and WHAT(4) $\geq$ 1.0) constitute a complete interface to the entire Fluka transport.
  - Therefore, MGDRAW can be used not only to write a collision tape, but to do any kind of complex analysis. (*e.g.*, event by event output as used in HEP applications).
- When mgdraw should better <u>not be used</u>
  - When biasing is requested (non-analogue run)
  - Whenever low-energy neutrons (E<20 MeV) are used, unless one has a deep knowledge of the peculiarities of their transport and quantities (*i.e.*, kerma, etc)

(User) Bugs & Problems (Reminder)

Scoring: USRBIN/EVENTBIN

Error Message:

*** Activity/fission/neutron balance binnings cannot be track-length!!!

USRBIN scoring method:

- Track-length quantities: that can be distributed along a track,
  e.g. fluence, energy deposition…
  WHAT(1)>=10

- Point-wise quantities: that have to be scored on a point, or in the middle of the step!
  e.g Activity, Fission, Neutron balance…
  WHAT(1)<10

Other Error in Scoring Definitions

- Example: badly defined USRBIN limits

Error Message:

******* Fluka stopped in Usrbin: "usr/eventbin" n. 1 *******
******* with zero width 0.000 for axis R ******

- Never use unit numbers smaller than 20 or higher than 99
  <20 They are reserved by FLUKA
  >99 Depends on fortran
- Never mix the output of different scoring cards in the same unit

automatically checked by FLAIR

Merging Cycles from Different Inputs

<u>Error Message:</u>

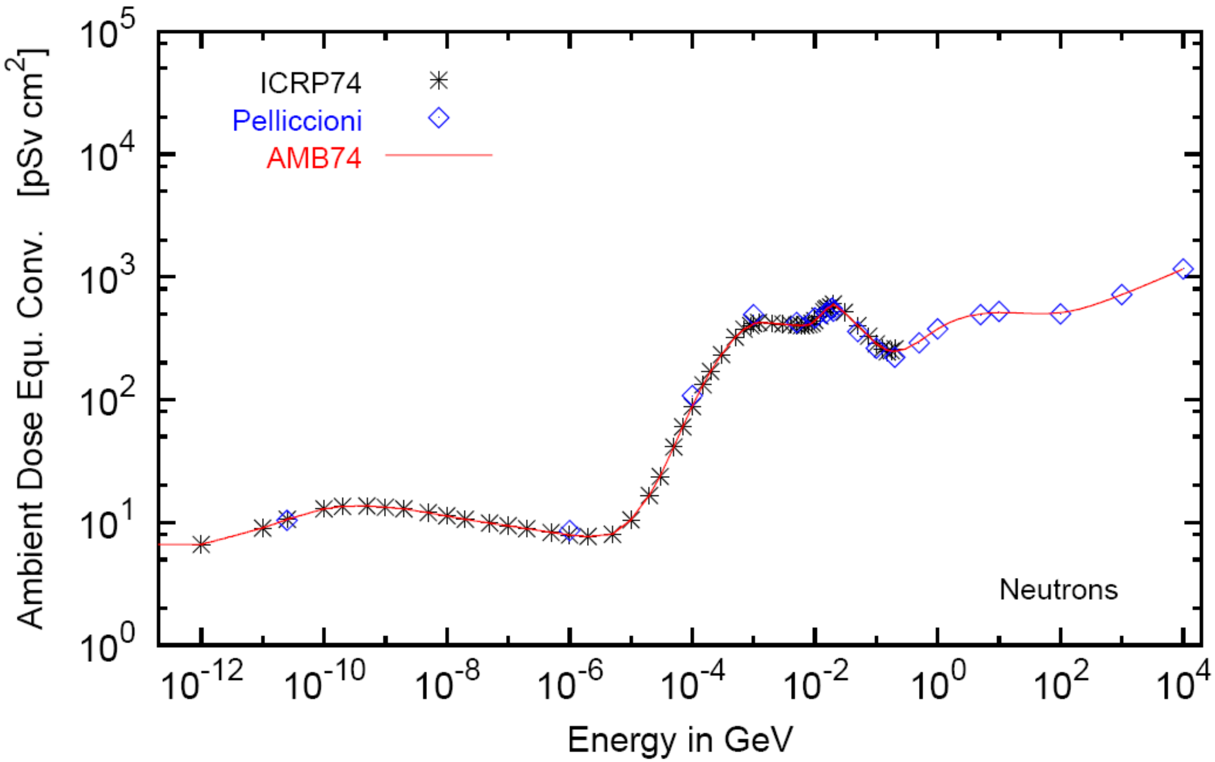
none

- Verify that you didn't merge cycles from different runs that the input has been modified.
- It's a good habit to clean the files before starting a new run with a modified input file.
- Flair offers this possibility from the "Output Files" frame.
- It's good to develop the habit to clean the output files from test runs.
- ….change the name of the input file for every new problem!

Backup

Conversion Coefficients

Conversion coefficients from fluence to ambient dose equivalent are based on ICRP74 values and values calculated by M.Pelliccioni. They are implemented for protons, neutrons, charged pions, muons, photons, electrons (conversion coefficients for other particles are approximated by these). AMB74 is the default choice for dose equivalent calculation.

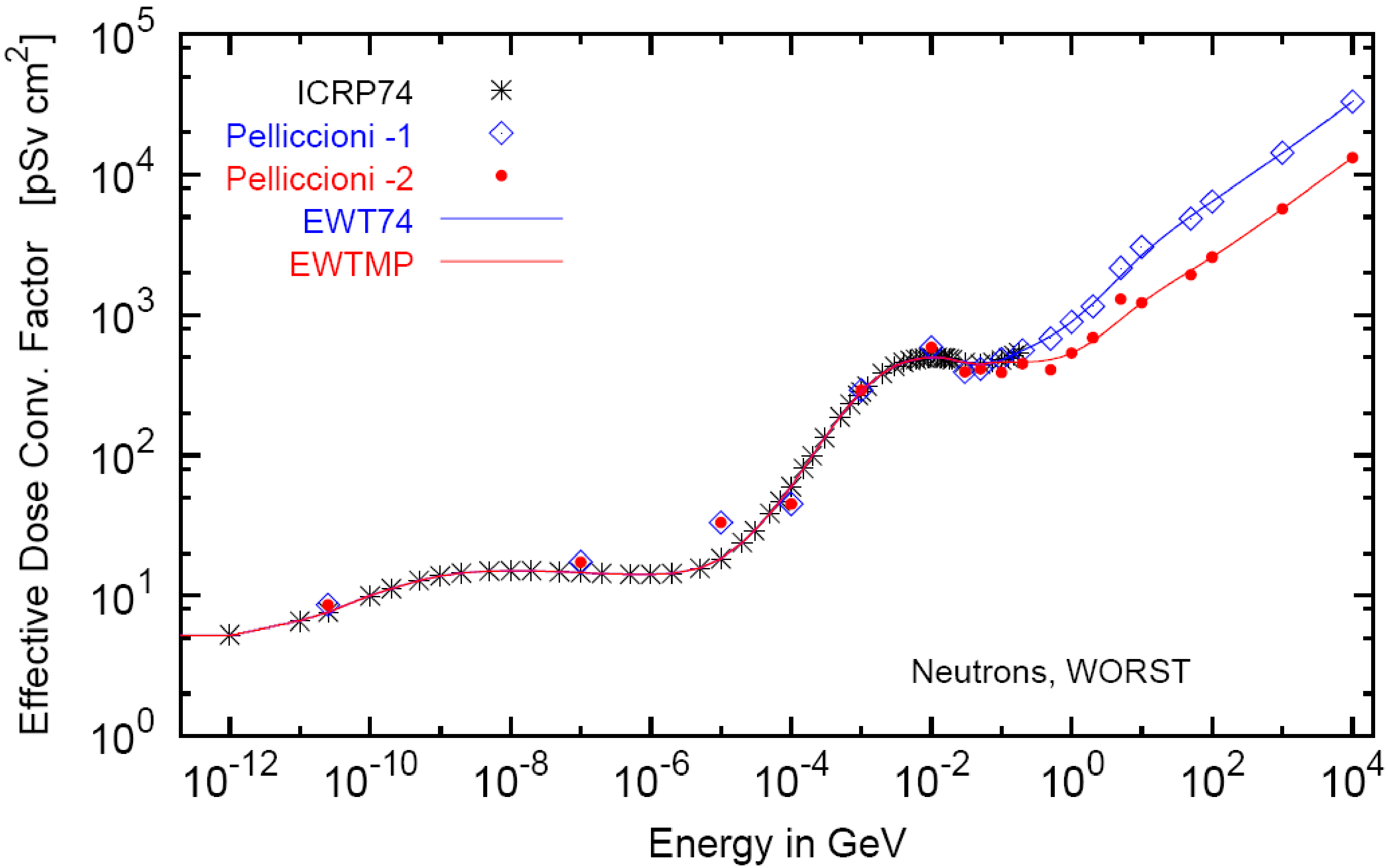


For more info: http://cern.ch/info-fluka-discussion/download/deq2.pdf

Fluence to effective dose coefficients

- Conversion coefficients from fluence to effective dose are implemented for three different irradiation geometries:
  - anterior-posterior
  - rotational
  - WORST ("Working Out Radiation Shielding Thicknesses") is the maximum coefficient of anterior-posterior, posterior-anterior, right-lateral and left-lateral geometries. It is recommended to be used for shielding design.
- Implemented for radiation weighting factors recommended by ICRP60 (e.g., SDUM=ETW74) and recommended by M.Pelliccioni (e.g., SDUM=EWTMP). The latter anticipate the 2007 recommendations of ICRP.
- Implemented for protons, neutrons, charged pions, muons, photons, electrons (conversion coefficients for other particles are approximated by these)
- Zero coefficient is applied to all heavy ions

Fluence to effective dose coefficients



USRYIELD

- Scores a double-differential particle yield around an extended or a point target.
- "Energy-like" quantities

Kinetic energy , total momentum , total energy , longitudinal momentum in the lab frame , longitudinal momentum in the c.m.s. frame LET

- "Angle-like" quantities (in degrees or radians)

Rapidity in the lab frame , rapidity in the c.m.s. frame , pseudorapidity in the lab frame , pseudorapidity in the c.m.s. frame , Feynman-x in the lab frame , Feynman-x in the c.m.s. frame , transverse momentum , transverse mass , polar angle (*) in the lab frame , polar angle (*) in the c.m.s. frame , square transverse momentum , charge , weighted angle in the lab frame , weighted transverse momentum