Sources

Advanced FLUKA Course

Overview

1. Built-in sources

- Beam definition
- Extended sources
- Colliding beams
- Synchrotron radiation (SPECSOUR)

2. User-defined sources

- User routine SOURCE
- Useful auxiliary routines
- Sampling techniques
- Two-step methods

Built-in sources

Beam definition – 1

Input card: BEAM

defines several *beam characteristics*:
type of particle, energy, divergence, profile

Example

```
*...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+...
BEAM                 3.5 -0.082425        -1.7        0.0        0.0        0.0PROTON
```

- 3.5 GeV/c [**WHAT(1)**] proton beam [**SDUM**] with weight 1 [**WHAT(6)**]
- Gaussian momentum distribution: 0.082425 GeV/c FWHM [**WHAT(2)**]
- Gaussian angular distribution: 1.7 mrad FWHM [**WHAT(3)**]
- no beam width along x (point-like source) [**WHAT(4)**]
- no beam width along y (point-like source) [**WHAT(5)**]

Beam definition - 2

Input card: BEAMPOS

If **SDUM** = blank:

defines the coordinates of the centre of the beam spot and the beam direction

Example

```
*...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+...
BEAMPOS          0.0       0.0       -0.1       0.0       0.0       0.0
```

- x-coordinate: 0.0 [**WHAT(1)**]
- y-coordinate: 0.0 [**WHAT(2)**]
- z-coordinate: -0.1 cm [**WHAT(3)**]
- direction cosine with respect to the x-axis: 0.0 [**WHAT(4)**]
- direction cosine with respect to the y-axis: 0.0 [**WHAT(5)**]
- **WHAT(6)** is not used !

→ beam points in the positive z-direction starting at (0.,0.,-0.1)

Beam definition - 3

Input card: BEAMAXES

defines the beam reference frame which all parameters defined with BEAM and BEAMPOS refer to (angular divergence, transverse profile, polarization, extended sources)

Example

```
*...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+...
BEAMAXES             1.0       0.0       0.0       0.0 0.7071068 0.7071068
```

- cosine of angle between x-axis of beam and x-axis of geometry frame [WHAT(1)]
- cosine of angle between x-axis of beam and y-axis of geometry frame [WHAT(2)]
- cosine of angle between x-axis of beam and z-axis of geometry frame [WHAT(3)]

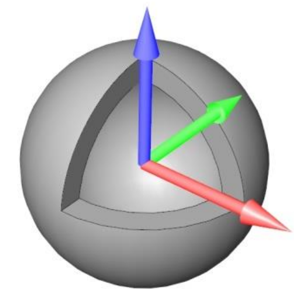
  (1.,0,0) → x-axes of beam and geometry frames are parallel

- cosine of angle between z-axis of beam and x-axis of geometry frame [WHAT(4)]
- cosine of angle between z-axis of beam and y-axis of geometry frame [WHAT(5)]
- cosine of angle between z-axis of beam and z-axis of geometry frame [WHAT(6)]

  (0.,0.7071068,0.7071068) → z-axis of beam frame is at 45deg to both y- and z-axes of geometry frame

Extended sources - *Spherical shell source*

Input card: BEAMPOS



If **SDUM** = **SPHE-VOL**:

defines a spatially extended source in a spherical shell

Example

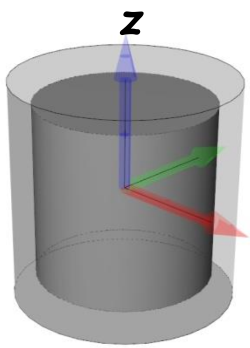
```
*...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+...
BEAMPOS          0.0       0.0       0.0       0.0       0.0       0.0
BEAMPOS          0.0       1.0       0.0       0.0       0.0       0.0SPHE-VOL
```

- radius (in cm) of the inner sphere shell: 0.0 cm [**WHAT(1)**]
- radius (in cm) of the outer sphere shell: 1.0 cm [**WHAT(2)**]
- **WHAT(3)** - **WHAT(6)** are not used !

The shell is centred at the (x,y,z) point defined by another BEAMPOS card with **SDUM** = blank (or = **NEGATIVE**). The particle direction or angular distribution are those defined by BEAM, BEAMAXES and another BEAMPOS cards.

Extended sources - *Cylindrical shell source*

Input card: BEAMPOS



If **SDUM** = **CYLI-VOL**:

defines a spatially extended source in a cylindrical shell with the height parallel to the z-axis of the beam frame

Example

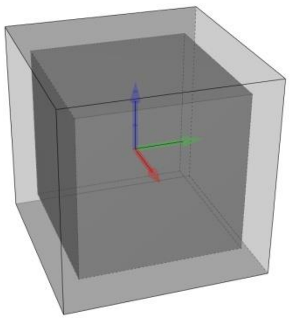
```
*...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+...
BEAMPOS          0.0       0.0       0.0       0.0       0.0       0.0
BEAMPOS          0.0       1.0       0.0       1.0       0.0       0.0CYLI-VOL
```

- radius (in cm) of the inner cylinder defining the shell: 0.0 cm [**WHAT(1)**]
- radius (in cm) of the outer cylinder defining the shell: 1.0 cm [**WHAT(2)**]
- height (in cm) of the inner cylinder defining the shell: 0.0 cm [**WHAT(3)**]
- height (in cm) of the outer cylinder defining the shell: 1.0 cm [**WHAT(4)**]
- **WHAT(5)** - **WHAT(6)** are not used !

The shell is centred at the (x,y,z) point defined by another BEAMPOS card with **SDUM** = blank (or = **NEGATIVE**). The particle direction or angular distribution are those defined by BEAM, BEAMAXES and another BEAMPOS cards.

Extended sources - *Cartesian shell source*

Input card: BEAMPOS



If **SDUM** = **CART-VOL**:

defines a spatially extended source in a Cartesian shell with the sides parallel to the beam frame axes

Example

```
*...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+...
BEAMPOS          0.0       0.0       0.0       0.0       0.0       0.0
BEAMPOS          0.0       1.0       0.0       1.0       0.0       1.0CART-VOL
```

- length (in cm) of the x-side of the inner parallelepiped defining the shell: 0.0 cm [**WHAT(1)**]
- length (in cm) of the x-side of the outer parallelepiped defining the shell: 1.0 cm [**WHAT(2)**]
- length (in cm) of the y-side of the inner parallelepiped defining the shell: 0.0 cm [**WHAT(3)**]
- length (in cm) of the y-side of the outer parallelepiped defining the shell: 1.0 cm [**WHAT(4)**]
- length (in cm) of the z-side of the inner parallelepiped defining the shell: 0.0 cm [**WHAT(5)**]
- length (in cm) of the z-side of the outer parallelepiped defining the shell: 1.0 cm [**WHAT(6)**]

The shell is centred at the (x,y,z) point defined by another BEAMPOS card with **SDUM** = blank (or = **NEGATIVE**). The particle direction or angular distribution are those defined by BEAM, BEAMAXES and another BEAMPOS cards.

Extended sources – *Spherical surface source*

Input card: BEAMPOS

If **SDUM** = **FLOOD**:

defines a source distribution on a spherical surface

Example

```
*...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+...
BEAMPOS          0.0       0.0       0.0       0.0       0.0       0.0
BEAMPOS          1.0       0.0       0.0       0.0       0.0       0.0FLOOD
```

- radius (in cm) of the sphere: 1.0 cm [**WHAT(1)**]
- **WHAT(2)** - **WHAT(6)** are not used !

The surface is centred at the (x,y,z) point defined by another BEAMPOS card with **SDUM** = blank (or = **NEGATIVE**). The particle direction is sampled according to a diffusive distribution so as to generate a uniform fluence equal to $1/(\pi R^2)$ inside the sphere (in absence of materials)

Extended sources - *Example*

Radioactive source of ^{60}Co (two main γ-emissions: 1332.5 keV and 1173.2 keV)

cylindrical shape, 2cm diameter, 2mm height along z, centre of cylinder base at origin

```
*...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+...
BEAM             0.0                                                  ISOTOPE
HI-PROPE        27.0      60.0
BEAMPOS          0.0       0.0       0.1       0.0       0.0       0.0
BEAMPOS          0.0       1.0       0.0       0.2       0.0       0.0CYLI-VOL
```

or

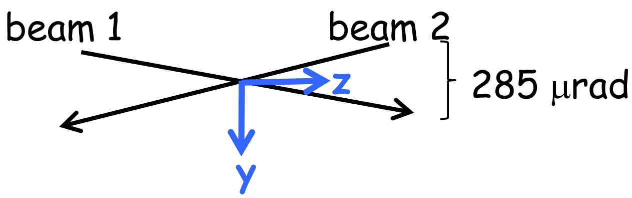
```
*...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+...
BEAM        1252.8E-6            10000.                               PHOTON
BEAMPOS          0.0       0.0       0.1       0.0       0.0       0.0
BEAMPOS          0.0       1.0       0.0       0.2       0.0       0.0CYLI-VOL
```

If height along x (instead of z) add

```
*...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+...
BEAMAXES         0.0       0.0      -1.0       1.0       0.0       0.0
```

Special sources – hadron-nucleus *collision*

Input card: SPECSOUR



Example: LHC

7 TeV/c, full crossing angle of 285 μrad in yz-plane

Momentum vectors of colliding beams: *three possibilities*

1) If SDUM = **PPSOURCE**:

```
SPECSOUR          0.    0.9975 6999.9999       0.0     0.9975-6999.9999PPSOURCE
```

- x, y, z-components of lab momentum for beam 1 particle [**WHAT(1-3)**]
- x, y, z-components of lab momentum for beam 2 particle [**WHAT(4-6)**]

2) If SDUM = **CROSSASY**:

```
SPECSOUR       7000.  142.5E-6      90.0     7000.  142.5E-6       0.0CROSSASY
```

- lab momentum for beam 1 particle [**WHAT(1)**]
- polar angle (rad) between beam 1 particle momentum and positive z-direction [**WHAT(2)**]
- azimuth angle (deg!) defining crossing plane [**WHAT(3)**]
- lab momentum for beam 2 particle [**WHAT(4)**]
- polar angle (rad) between beam 2 particle momentum and negative z-direction [**WHAT(5)**]

Special sources – hadron-nucleus *collision*

3) If SDUM = CROSSSYM:

```
*...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+...
SPECSOUR       7000.  142.5E-6      90.0         0.0         0.0         0.0CROSSSYM
```

- lab momentum for beam 1 and 2 particle [WHAT(1)]
- half crossing angle (rad) [WHAT(2)]
- azimuth angle (deg!) defining crossing plane [WHAT(3)]
- WHAT(4) - WHAT(6) are not used !

Interaction point of colliding beams (continuation card):

```
SPECSOUR       7000.  142.5E-6      90.0         0.0         0.0         0.0CROSSSYM
SPECSOUR      12.E-4    12.E-4       5.0                                   &
```

- sigma_x in cm for Gaussian sampling around XBEAM: 12 um [WHAT(7)]
- sigma_y in cm for Gaussian sampling around YBEAM: 12 um [WHAT(8)]
- sigma_z in cm for Gaussian sampling around ZBEAM: 5 cm [WHAT(9)]

(XBEAM,YBEAM,ZBEAM) defined with BEAMPOS card

- sampling limit, in sigma, applying along x, y, and z [WHAT(10)]
  =< 0 no limit

Special sources – hadron-nucleus *collision*

```
BEAM           3000.0                                                     HEAVYION
HI-PROPE         82.0      208.0
...
SPECSOUR      574000.   142.5E-6       90.0       0.0       0.0       0.0CROSSSYM
SPECSOUR       12.E-4     12.E-4        5.0                          208.0&
SPECSOUR          82.                                                     &
```

- ID of beam 1 particle (default: the one of BEAM) [WHAT(11)]
- mass number of beam 2 particle (default: proton) [WHAT(12)]
- charge of beam 2 particle [WHAT(13)]

```
SPECSOUR       7000.0   0.000335      180.0       0.0       0.0       0.0CROSSSYM
SPECSOUR          0.0        0.0       5.34       0.0                 0.0&
SPECSOUR          1.0  1.057E-5   1.057E-5  1.057E-5  1.057E-5       0.0&&
```

- sigma_th_C (rad) for the Gaussian sampling of the beam 1 particle angle [WHAT(14)]
  wrt the ideal momentum in the Crossing plane
- sigma_th_O (rad) for the Gaussian sampling of the beam 1 particle angle [WHAT(15)]
  wrt the ideal momentum in the Orthogonal plane
- the same as WHAT(14) for beam 2 particle [WHAT(16)]
- the same as WHAT(15) for beam 2 particle [WHAT(17)]

Special sources – hadron-nucleus *collision*

Three interaction types are available:

- i0 + i1 * 10 + i2 * 100 [**WHAT(18)**]

i0 = flag for **nuclear nonelastic** interactions

i1 = flag for **nuclear elastic** interactions

i2 = flag for **electromagnetic dissociation** (EMD) interactions

default: nuclear nonelastic + EMD if selected with the **PHYSICS(SDUM = EM-DISSO)** card

For collisions in the DPMJET energy range, don't forget (to link it as well as to input) the following card

```
*...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+...
PHYSICS        8000.0                                                   LIMITS
```

WHAT(1) [GeV/c] must be larger than the maximum nucleon centre-of-mass momentum

Special sources – *Synchrotron radiation*

- Sophisticated low energy photon transport including polarization effects for Compton, photoelectric and coherent scattering, and full account for bound electron effects: already available in FLUKA since several years

- FLUKA can model the emission of Synchrotron Radiation:
  - by any charged particle, with **arbitrary orientation** vs magnetic field and traversing **up to 2 circular arcs or helical paths,**

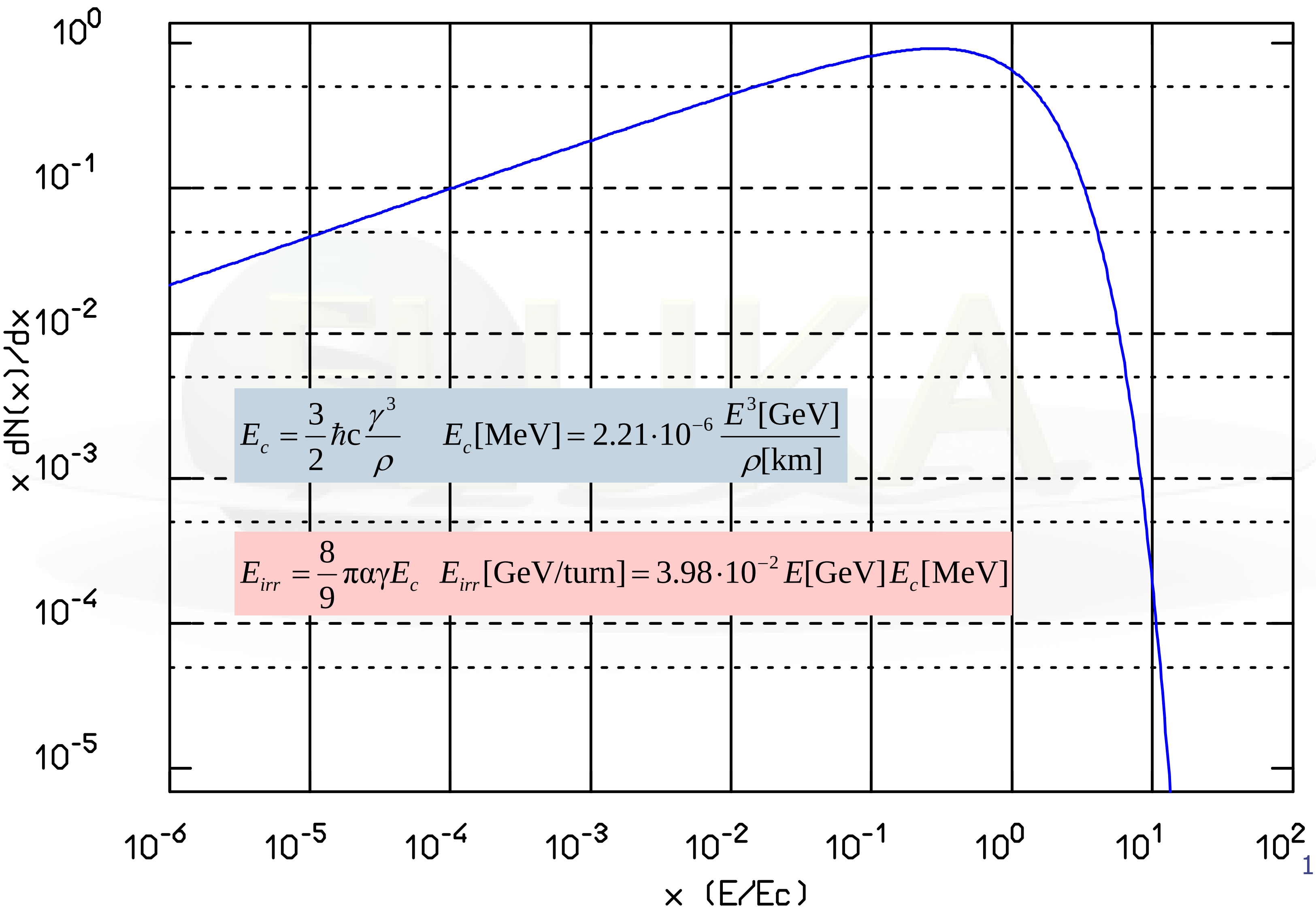
➡ accounting for the **emitted photon polarization,** as a function of the emitted photon energy, and

➡ **sampling SR photon energy and SR photon angle**

The emitting charged particle is NOT transported:
SR photons are sampled directly.

Readily usable for bending magnets and wigglers (two steps so far).

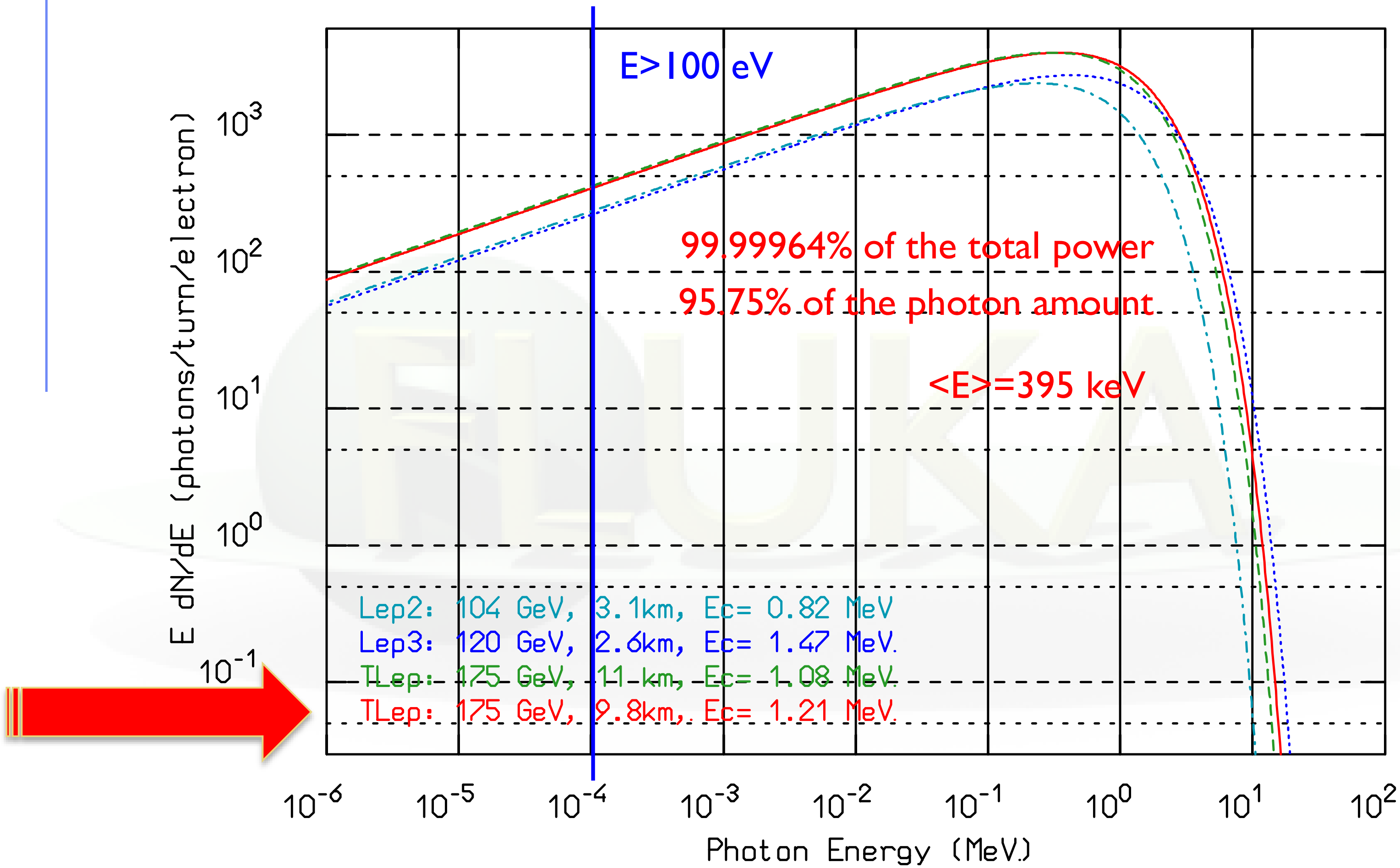
Special sources – Synchrotron radiation



$$E_c = \frac{3}{2}\hbar c \frac{\gamma^3}{\rho} \qquad E_c[\mathrm{MeV}] = 2.21 \cdot 10^{-6} \frac{E^3[\mathrm{GeV}]}{\rho[\mathrm{km}]}$$

$$E_{irr} = \frac{8}{9}\pi\alpha\gamma E_c \quad E_{irr}[\mathrm{GeV/turn}] = 3.98 \cdot 10^{-2} E[\mathrm{GeV}] E_c[\mathrm{MeV}]$$

Special sources – *synchrotron radiation*



- ΔE= 8.5 GeV/turn (dE/ds=1.375 keV/cm in the dipoles)
- P = 8.5 x I[mA] MW = 8.5 x 10mA = 85 MW in the whole accelerator (dP/ds= 1.375 x I[mA] W/cm in the dipoles)

Special sources – *synchrotron radiation*

```
FREE
SPECSOUR  , ELECTRON, 175.0, 979948.86, 0.0000001, 0.0, -1.0, SYNC-RAD
SPECSOUR  , 1050.0, -0.59467382951, 0., 1134.9997568, -.10714843289E-02, 0.0, &
FIXED
```

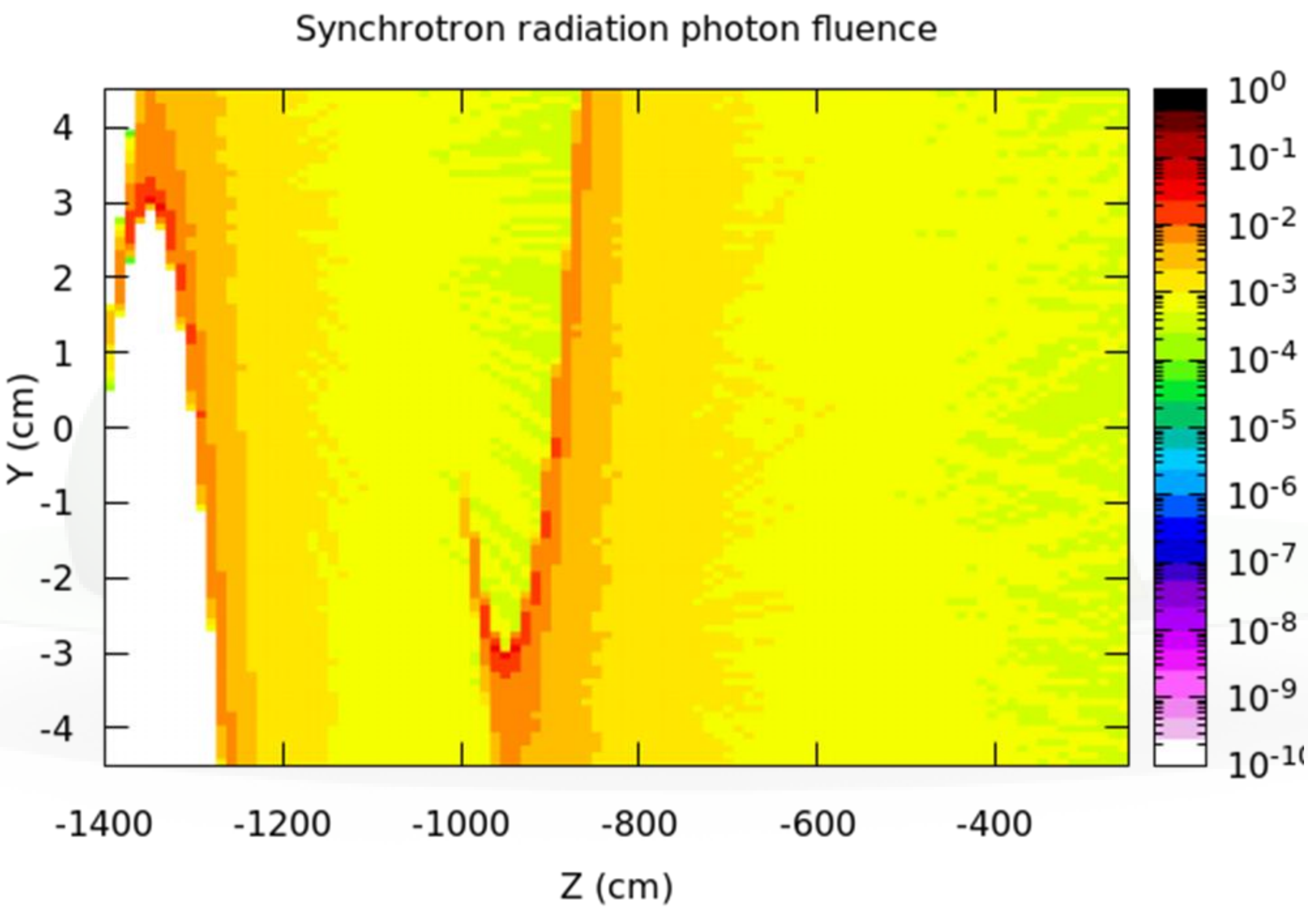
- particle emitting the radiation [**WHAT(1)**]
- emitting particle momentum [GeV/c if >0] or kinetic energy [GeV if <0] [**WHAT(2)**]
- curvature radius [cm if >0] or magnetic field [T if <0] [**WHAT(3)**]
- photon spectrum lower limit [GeV] [**WHAT(4)**]
- x/y-components of the magnetic field versor [**WHAT(5/6)**]

  The z-component sign is positive for SYNC-RAD and negative for SYNC-RDN

- length [cm] of the emission arc [**WHAT(7)**]
- coordinates (x/y/z [cm]) of the starting point of a possible second arc of same length [**WHAT(8/9/10)**]
- x/y-components of the emitting particle direction versor at the beginning of the second arc [**WHAT(11/12)**]

The starting point of the first arc as well the initial direction of the emitting particle are defined in the BEAMPOS card

Synchrotron radiation: 2-arc example




```
BEAMPOS                         0.5    -1400.0                0.100
SPECSOUR    ELECTRON            3.0       -2.0 0.0000001      1.000        0.0SYNC-RAS
SPECSOUR       150.0            0.0       -0.5     -1000.       0.0     -0.100&
```

A comment about the units

All simulation results for the synchrotron radiation SPECSOUR are quoted **per simulated synchrotron radiation photon**.

From the output file (previous ex.):

```
<<< Synchrotron radiation source n.    1 >>>

Emitting particle: ELECTRON P:      3.00000 GeV/c
  Initial position :    0.0000000      0.50000000      -1400.0000     cm
  Initial direction:    0.0000000      0.10000000      0.99498744

Magnetic field:    2.0000000       0.0000000       0.0000000     T
Nominal curvature radius:    500.34614     cm
Nominal arc:    150.00000     cm
Arc angle:   0.29979246     rad
Actual  curvature radius:    500.34614     cm
Actual arc:    150.00000     cm
Transverse p_T:      3.00000 GeV/c  and gamma:       5870.85237

Critical energy:    0.0000119705 GeV

Photon emission threshold        :   1.00000000E-07 GeV
Photons >1 eV/nominal unit length:   0.11693748     cm^-1
Photons/unit length 1 eV - thres.:   2.38764527E-02 cm^-1
Photons/unit length above  thres.:   9.30610323E-02 cm^-1

Total energy/nominal unit length:   4.55537630E-07 GeV/cm
Energy/unit length below thresh.:   7.54228751E-10 GeV/cm
Energy/unit length above thresh.:   4.54783401E-07 GeV cm
```

in this specific case
we have to scale results
by 150*.093061
to obtain results
per primary emitting particle

BEAM Visualization

USRBIN

- Create a **USRBIN** covering the beam position (preferentially Cartesian X-Y-Z) with BEAMPART as scoring particle
- Set all materials to VACUUM (to speed up calculation)
- Make one run of 1 cycle
- Visualize the results:
  - in flair as USRBIN plot
  - in the geometry editor as a custom USRBIN layer (don't forget to set properly the colorband)

With USERDUMP

- Add a **`USERDUMP`** card selecting ONLY Source particles
- Make one run of 1 cycle
- Create a USERDUMP plot in flair:

  Select the “Source” tab

  You have the ability to make

  - 1D histogram plots of any of the source quantities
  - 2D scattered plots for any of the source quantities with even the possibility to overlay on a geometry image

User-defined sources

Source routine - 1

- Allows the definition of primary particle properties (in space, energy, time, direction, or mixture of particles) which cannot be described with built-in sources

- Activated with input card SOURCE. The parameter list of that card (two continuation cards possible!) allows the user to pass on up to 18 numerical values `WHASOU(1-18)` and one 8-character string `SDUSOU` via `COMMON /SOURCM/`

- At each call, one (or more) particle(s) must be loaded onto `COMMON /FLKSTK/` (particle bank) before returning control. The relevant variable values can be read from a file, generated by some sampling algorithm, or just assigned.

- Argument list: if `NOMORE=1` (output variable) the run will be terminated after exhausting the primary particles loaded onto the stack in the present call. The history number limit set with card START will be overridden.

Source routine - 2

```
...
      LOGICAL LFIRST
*
      SAVE LFIRST
      DATA LFIRST / .TRUE. /
...
      NOMORE = 0
*  +-------------------------------------------------------------------*
*  |  First call initializations:
      IF ( LFIRST ) THEN
*  |  *** The following 3 cards are mandatory ***
         LFIRST = .FALSE.
         TKESUM = ZERZER
         LUSSRC = .TRUE.
*  |  *** User initialization ***
```

Any first-time initialization can be inserted here, for example
- setting up parameters passed on via SOURCE card
- reading spectra from data files

```
      END IF
...
```

Source routine - 3

```
...
      NPFLKA = NPFLKA + 1
* Wt is the weight of the particle
      WTFLK  (NPFLKA) = ONEONE
      WEIPRI = WEIPRI + WTFLK (NPFLKA)
* Particle type (1=proton.....). Ijbeam is the type set by the BEAM
* card
*  +-------------------------------------------------------------------*
*  |  (Radioactive) isotope:
      IF ( IJBEAM .EQ. -2 .AND. LRDBEA ) THEN
         IARES  = IPROA
         IZRES  = IPROZ
         IISRES = IPROM
         CALL STISBM ( IARES, IZRES, IISRES )
         IJHION = IPROZ  * 1000 + IPROA
         IJHION = IJHION * 100 + KXHEAV
         IONID  = IJHION
         CALL DCDION ( IONID )
         CALL SETION ( IONID )
*  |
*  +-------------------------------------------------------------------*
*  |  Heavy ion:
      ELSE IF ( IJBEAM .EQ. -2 ) THEN
         IJHION = IPROZ  * 1000 + IPROA
         IJHION = IJHION * 100 + KXHEAV
         IONID  = IJHION
         CALL DCDION ( IONID )
         CALL SETION ( IONID )
         ILOFLK (NPFLKA) = IJHION
*  |  Flag this is prompt radiation
         LRADDC (NPFLKA) = .FALSE.
*  |  Group number for "low" energy neutrons, set to 0 anyway
         IGROUP (NPFLKA) = 0
*  |
*  +-------------------------------------------------------------------*
*  |  Normal hadron:
      ELSE
         IONID = IJBEAM
         ILOFLK (NPFLKA) = IJBEAM
*  |  Flag this is prompt radiation
         LRADDC (NPFLKA) = .FALSE.
*  |  Group number for "low" energy neutrons, set to 0 anyway
         IGROUP (NPFLKA) = 0
      END IF
*  |
*  +-------------------------------------------------------------------*
...
```

increase pointer in FLKSTK

weight of particle
(if varying -> biased source)

total weight of primaries (don't change)

Definition of particle type

- The template sets the type of particle equal to the one defined by the BEAM card (and HI-PROPE, if used).

- Whichever valid particle type can be set inside the source (may be varying event by event)

Source routine - 4

```
...
* Particle age (s)
      AGESTK (NPFLKA) = +ZERZER
      AKNSHR (NPFLKA) = -TWOTWO
* Kinetic energy of the particle (GeV)
      TKEFLK (NPFLKA) = SQRT ( PBEAM**2 + AM (IONID)**2 ) - AM (IONID)
* Particle momentum
      PMOFLK (NPFLKA) = PBEAM
* Cosines (tx,ty,tz)
      TXFLK  (NPFLKA) = UBEAM
      TYFLK  (NPFLKA) = VBEAM
      TZFLK  (NPFLKA) = WBEAM
*     TZFLK  (NPFLKA) = SQRT ( ONEONE - TXFLK (NPFLKA)**2
*    &                       - TYFLK (NPFLKA)**2 )
* Polarization cosines:
      TXPOL  (NPFLKA) = -TWOTWO
      TYPOL  (NPFLKA) = +ZERZER
      TZPOL  (NPFLKA) = +ZERZER
* Particle coordinates
      XFLK   (NPFLKA) = XBEAM
      YFLK   (NPFLKA) = YBEAM
      ZFLK   (NPFLKA) = ZBEAM
...
```

momentum and energy

- by default taken from BEAM card (PBEAM in COMMON /BEAMCM/)
- the user can set (consistently!) any momentum or energy here (either from file or sampled)
- **NOTE:** BEAM card is always mandatory for initialization purposes. Momentum/energy set here must not exceed the respective BEAM card value.

direction cosines and coordinates

- by default taken from BEAMPOS card (COMMON /BEAMCM/)
- ensure proper normalization of cosines!

polarization

- `TXPOL = -2` flag for "no polarization"

Source routine - 5

```
* User dependent flag:
      LOUSE   (NPFLKA) = 0
...
* User dependent spare variables:
      DO 100 ISPR = 1, MKBMX1
         SPAREK (ISPR,NPFLKA) = ZERZER
 100  CONTINUE
* User dependent spare flags:
      DO 200 ISPR = 1, MKBMX2
         ISPARK (ISPR,NPFLKA) = 0
 200  CONTINUE
```

Variables that allow to store additional information in `COMMON /FLKSTK/`,
such as information on ancestors of a certain particle

Auxiliary routines - *Random numbers*

```
... = FLRNDM (XDUMMY)
```

returns a 64-bit random number [0-1)

NOTE: Fundamental for SOURCE! No other external random generators must be used, otherwise the history reproducibility will be lost.

```
CALL FLNRRN (RGAUSS)
```

returns a normally distributed random number RGAUSS

```
CALL FLNRR2 (RGAUS1,RGAUS2)
```

returns an uncorrelated pair of normally distributed random numbers RGAUS1 and RGAUS2

```
CALL SFECFE (SINT,COST)
```

returns SINT and COST, sine and cosine of a random azimuthal angle
SINT**2 + COST**2 = 1.D+00

```
CALL RACO (TXX, TYY, TZZ)
```

returns a random 3D direction (TXX, TYY, TZZ)
TXX**2 + TYY**2 + TZZ**2 = 1.D+00

Auxiliary routines – Name <-> number conv.

Conversion of region name to number

CALL GEON2R (REGNAM, NREG, IERR)

Input variable:

REGNAM = region name (CHARACTER*8)

Output variables:

NREG = region number

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

Conversion of region number to name

CALL GEOR2N (NREG, REGNAM, IERR)

Input variable:

NREG = region number

Output variables:

REGNAM = region name (CHARACTER*8)

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

Auxiliary routines - *Others*

CALL OAUXFI ('file', LUN, 'CHOPT', IERR)

to open an auxiliary file (to read data or parameters) looking automatically for the file in some default locations (temporary directory, working directory)

CALL FLABRT ('routine_name','message')

this allows to force a FLUKA abort on user request: it might be useful to perform a debugging (using gdb for instance)

CALL SFLOOD (XXX, YYY, ZZZ, UXXX, VYYY, WZZZ)

returns a random position XXX, YYY, ZZZ on the surface of a sphere of radius 1 and centre 0 (multiply XXX, YYY, ZZZ by the actual radius and add the centre coordinates) and a random direction UXXX, VYYY, WZZZ (cosines) so as to generate a uniform fluence inside the sphere, equal to $1/(\pi R^2)$, being R the actual sphere radius.

Sampling from a distribution – *Discrete*

1) From the cumulative 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

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

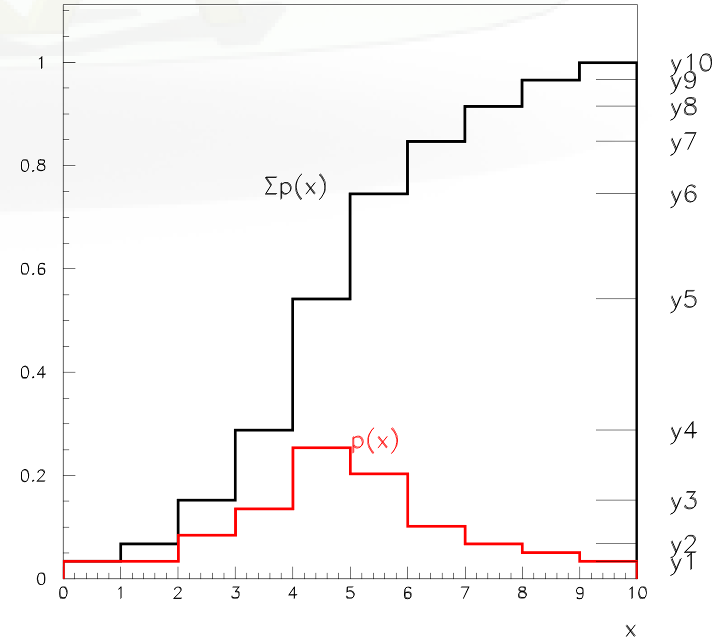
- Generate a uniform pseudo-random number ξ
- Find the i^{th} y-interval such that

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

- Select $X = x_i$ as the sampled value

Since ξ is uniformly random:

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



Sampling from a distribution – *Discrete*

2) By adjusting weights

- Suppose to have a fluence energy spectrum Φ given in N discrete energy bins between E_0 and E_N : $\Phi_1,...,\Phi_N$
- Generate a uniform pseudo-random number ξ
- Find the i^{th} energy bin such that

$$E_{i-1} \leq \xi (E_N - E_0) < E_i$$

- Generate another uniform pseudo-random number $\xi \in [0,1)$ and sample an energy uniformly within the i^{th} energy bin
- assign a weight Φ_i to that primary particle

Note: This method is often used for spectra steeply decreasing with energy (e.g., Φ ~1/E), where the result depends significantly on the particle cascades initiated by high energy primaries, as it ensures faster convergence to the true value.

Example Sampling from a histogram - 1

```
      PARAMETER (NMAX=1000)
      DIMENSION ERG(NMAX), CUM(NMAX)
      CHARACTER*250 LINE
      SAVE  N, ERG, CUM

       IF ( LFIRST ) THEN
       …
         LUNRD = NINT(WHASOU(1))
         N = 0
         SUM   = ZERZER
         EPREV = ZERZER
 10      CONTINUE
           READ (LUNRD, '(A)', ERR=9999, END=20 ) LINE
           READ (LINE, *, ERR=10) E, H
           N = N + 1
           IF (N .GT. NMAX)
     &          CALL FLABRT('SOURCE','Please increase NMAX')
           IF (N .EQ. 1 .AND. ABS(H).GT.AZRZRZ)
     &          CALL FLABRT(
     &               'SOURCE','ZERO was expected as first value')
*** Create cumulative sum of dE*V
           SUM = SUM + H*(E-EPREV)
           EPREV = E
           ERG(N) = E
           CUM(N) = SUM
         GO TO 10
 20      CONTINUE
         CLOSE (LUNRD)
      END IF
9999 CALL FLABRT('SOURCE', 'Error reading source file')
```

Logical unit from input file as WHAT(1) of the SOURCE card. Use OPEN card to open the file which contains *pairs Energy-Value*. First value is supposed to be 0 in order to set the *lower energy limit*.

Example: hsource.f

Example Sampling from a histogram - 2

```
*  From this point .....

*** Select a random energy interval
      C = CUM(N) * FLRNDM(C)
*** Find interval (CUM(1)=0)
      DO I=2,N
        IF (CUM(I) .GT. C) THEN
***         Found interval I, select a random energy inside
            E = ERG(I-1) + (ERG(I)-ERG(I-1))*FLRNDM(C)
            GO TO 90
        END IF
      END DO
```

Select a random cumulative value

Sampling from a distribution – *Continuous*

1) By integration

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

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

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

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

Sampling from a distribution – *Continuous*

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:

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

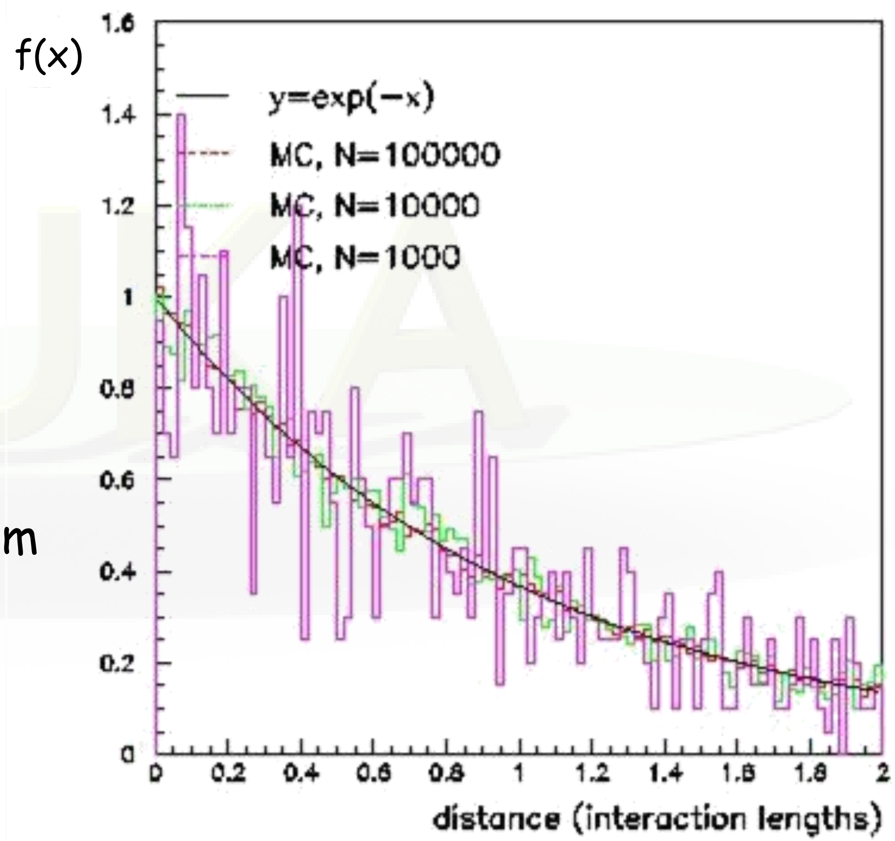
Generate a uniform pseudo-random number $\xi \in [0,1)$

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

Sample *t* by inverting

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

Repeat N times



Sampling from a distribution – *Continuous*

2) By rejection

- 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/sampled) is given by:

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

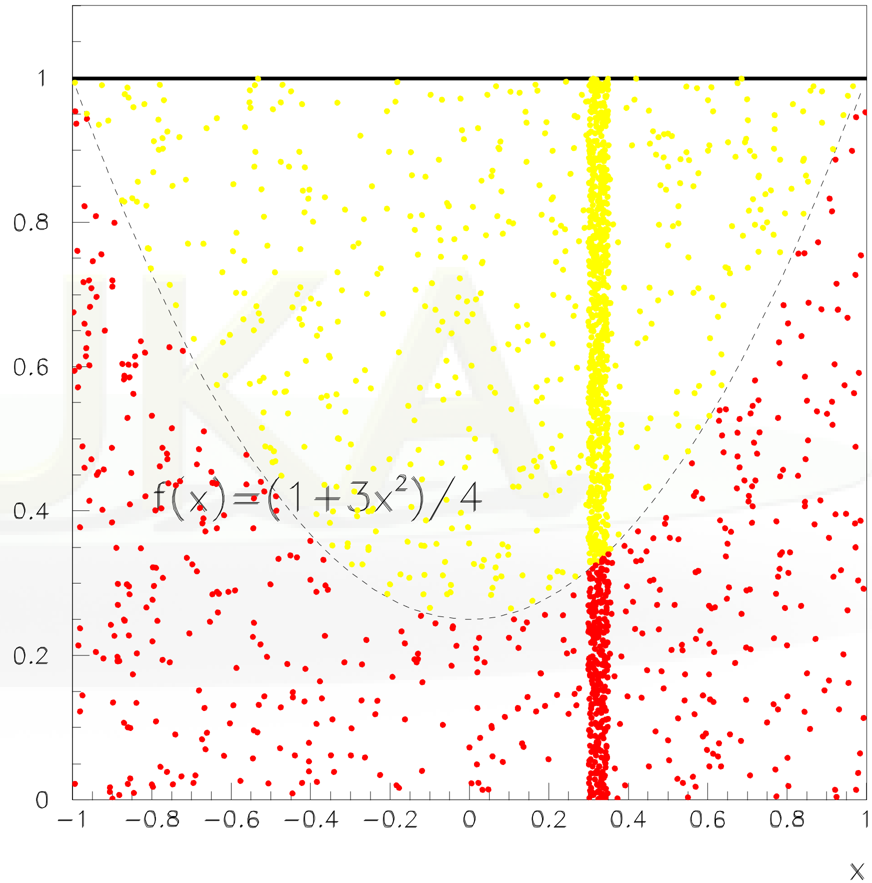
and the probability that X is accepted is unbiased:

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

Sampling from a distribution – *Continuous*

Example

- 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 ξ_1, $\xi_2 \in [0,1)$
- Accept $X=2\xi_1-1$ if $\xi_2 < (1+3X^2)/4$, if not repeat



Sampling from a distribution – *Continuous*

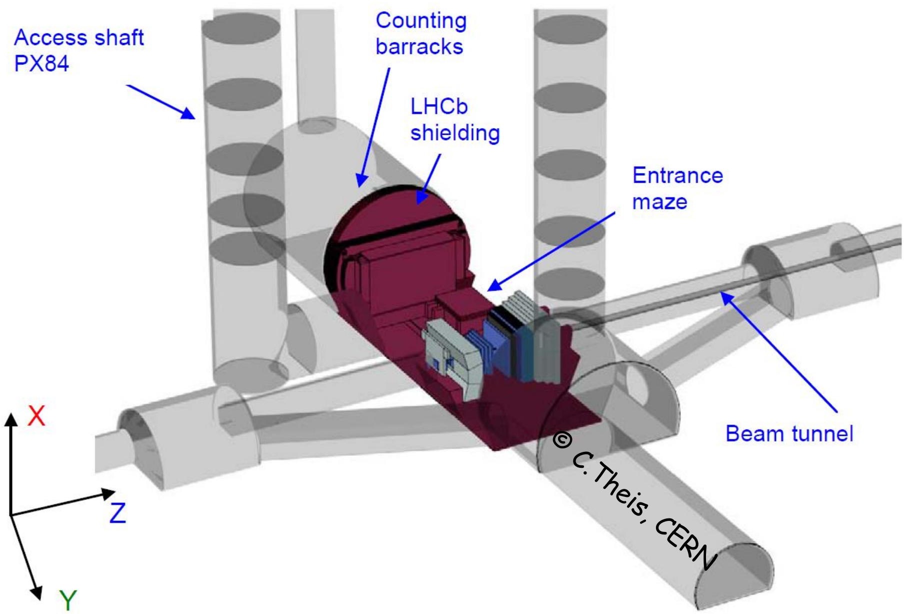
3) By adjusting weights

- Suppose to have a fluence energy spectrum $\Phi(E)$ given in between E_0 and E_1
- Generate a uniform pseudo-random number $\xi \in [0,1)$ and calculate the sampled energy $E = E_0 + \xi\,(E_1 - E_0)$
- Assign a weight $\Phi(E)$ to that primary particle

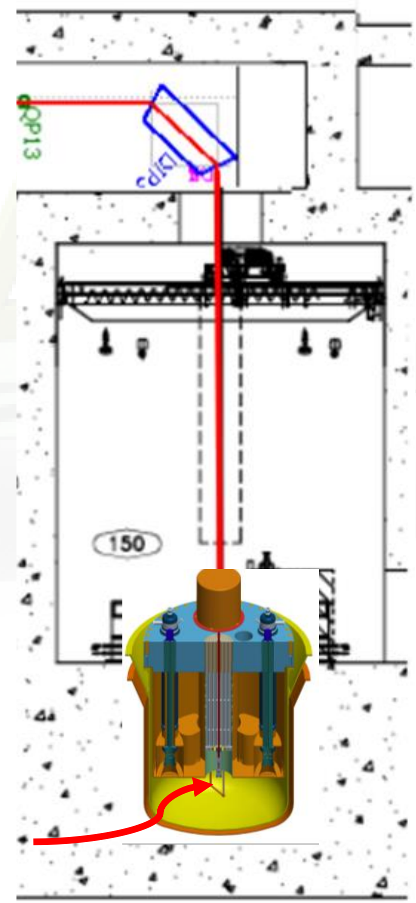
Two-step methods

Goal: predict radiation fields and observables in remote locations in a huge geometry

Ex.1: LHCb experiment



Ex.2: MYRRHA ADS



Spallation target

Problem:

direct calculation in one step
can be highly inefficient due to the small affected phase-space

Two-step methods

Solution: split simulation into two steps.

Two different approaches are possible:

✓ (1) Calculation of radiation fields in a suitable location:
for each particle type, calculate fluence distributions, double-differential in energy and angle

(2) Sample from the calculated distributions in a user-defined source, and score the interested quantities at the location of interest

Obs: - it must be clear that in this way the correlations within one full event are lost. All quantities are calculated as average quantities
- pay attention to calculate the radiation fields in step (1) covering all the phase space relevant for step (2)

✓ (1) Dumping particles at the location of interest:
write all information on particles entering it (type,energy, position, direction) into an external file

(2) Read the information from the external with a user-defined source

First approach: an example

In this example, a source term built calculating the neutrons backward emitted at the location (A) was important to calculate the average neutron and photon fluence on detectors very far from the neutron production point