Biasing techniques

23rd FLUKA Beginner's Course
Lanzhou University
Lanzhou, China

June 2–7, 2024

Outline

Outline

Motivation I

If you are new to Monte Carlo:

- This is not the most important lecture for you
  - you will definitely not need it in order to start using FLUKA
  - do not try to remember everything
  - just take this information into account so that later you know where to find it

Motivation II

- Once you gain experience with FLUKA
  and be dealing with more difficult problems,
- you will most likely realise that all available to you computational resources are not enough to solve them
- Then it's time to revisit this lecture and explore biasing techniques

Motivation III

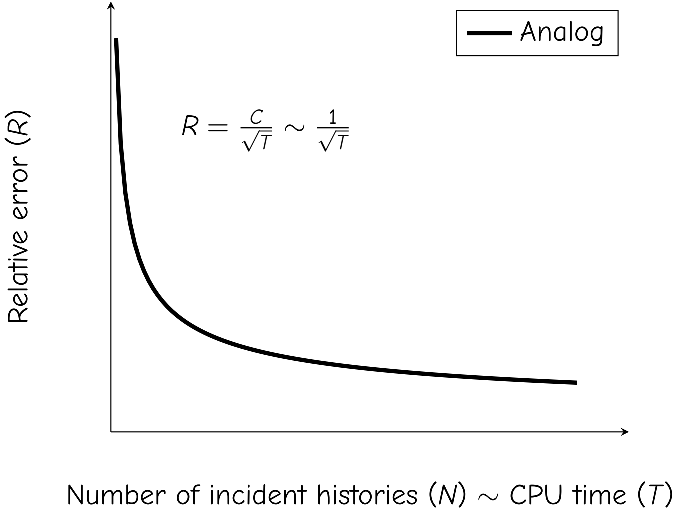
- Biasing — the use of variance reduction techniques that distort distributions and apply weights to particles to correct for the bias
- It's exchange of human time for computational efficiency
  - few hours of human time can reduce computational time by orders of magnitude
- The use of biasing techniques gives you a lot of power, but also requires the same amount of responsibility in return

Some definitions

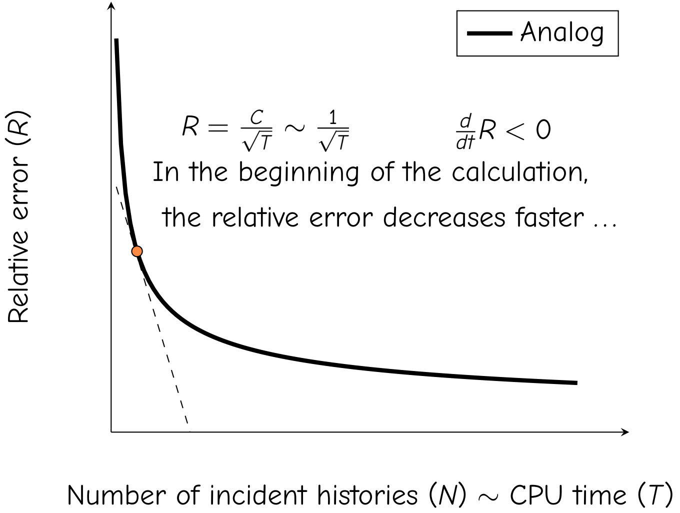
Analog calculation A calculation performed without using biasing methods

Biased calculation A calculation performed with biasing methods

History Calculation of transport of a single incident particle and all its secondaries through Monte Carlo geometry



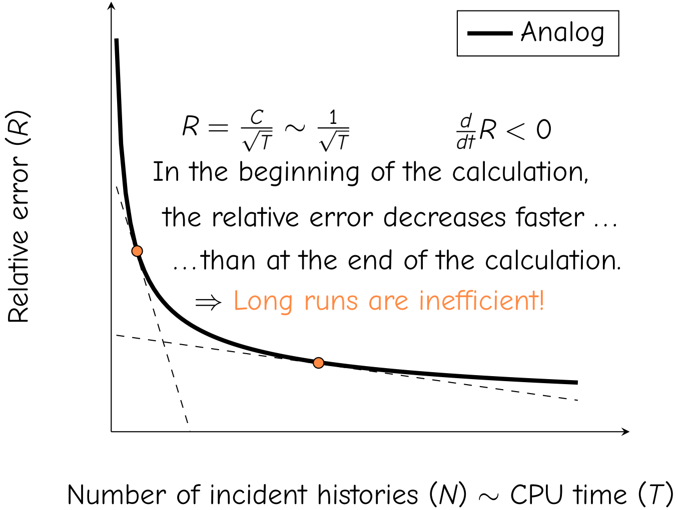
Analog
$R = \frac{C}{\sqrt{T}} \sim \frac{1}{\sqrt{T}}$
Relative error (R)
Number of incident histories $(N) \sim$ CPU time (T)



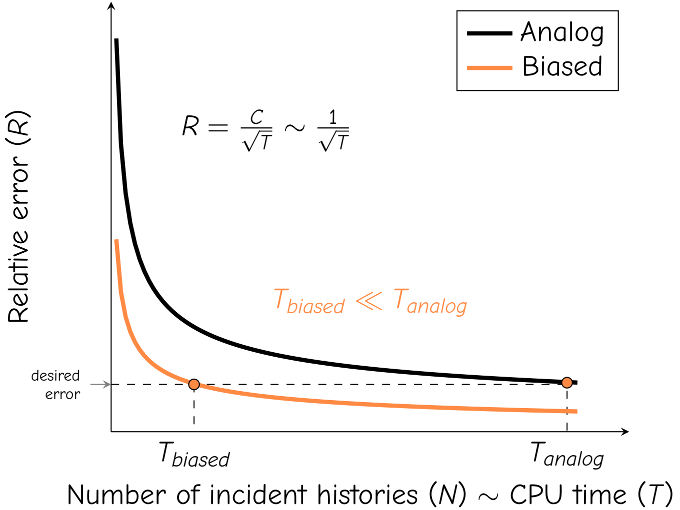
Analog
$R = \frac{C}{\sqrt{T}} \sim \frac{1}{\sqrt{T}}$
$\frac{d}{dt}R < 0$
In the beginning of the calculation,
the relative error decreases faster …
Relative error (R)
Number of incident histories (N) $\sim$ CPU time (T)

Motivation

Analog and biased calculations



Motivation
Analog and biased calculations



How to reduce relative error?

Relative error: $R = C/\sqrt{T}$

- C is a constant, depends on the simulation setup
- T is the CPU time, proportional to the number of primary histories: $T \sim N$

How to reduce relative error?

Relative error: $R = C/\sqrt{T}$

- C is a constant, depends on the simulation setup
- T is the CPU time, proportional to the number of primary histories: $T \sim N$

How to reduce R?

Increase T

run longer

$\uparrow$

inefficient

Decrease C

Choose right estimator

Choose wise sampling

Biasing

How to reduce relative error?

Relative error: $R = C/\sqrt{T}$

- C is a constant, depends on the simulation setup
- T is the CPU time, proportional to the number of primary histories: $T \sim N$

How to reduce R?

Increase T

run longer

$\uparrow$

inefficient

Decrease C

Choose right estimator

Choose wise sampling

Biasing

How to reduce relative error?
Example: Choosing right estimator

Hadron fluence

USRTRACK
(track-length estimator)

Distance travelled in the region

Get scores from every particle that enter the region

USRCOLL
(collision estimator)

$1/\Sigma_{total}$ at each collision

With small Σ_{total} few particles collide, but give enormous scores producing high variance

Test run: 20 MeV protons on Lead

Same CPU time (same run), but USRTRACK relative error is 17 times smaller

How to reduce relative error?

Relative error: $R = C/\sqrt{T}$

- C is a constant, depends on the simulation setup
- T is the CPU time, proportional to the number of primary histories: $T \sim N$

How to reduce R?

- Increase T
  - run longer
    - $\uparrow$ inefficient
- Decrease C
  - Choose right estimator
  - Choose wise sampling
    - Biasing

How to reduce relative error?
Choosing wise sampling

Monte Carlo sampling

- Analog
  - Events are sampled according to their natural cross-sections
- Biased
  - Bias the PDFs in order to sample the estimator more efficiently

Give more opportunity to increase efficiency of the calculation

Our today's focus

PDFs — probability density functions

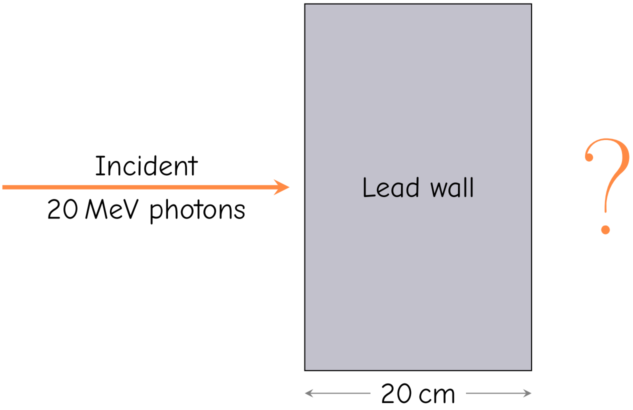
Efficiency of calculation

$$R = C/\sqrt{T} \quad \Rightarrow \quad R\sqrt{T} = C = const$$

- Calculation efficiency: a measure how quickly the desired precision is achieved
  - if CPU time T is fixed, smaller relative error R indicates the more efficient calculation $\leftarrow$ we will use it in the biasing exercise
- The number of histories is irrelevant
  - usually, running many histories shows ignorance
- Most biasing techniques decrease R faster than $\sqrt{T}$ is increased

  $\Rightarrow$ you need to run less histories to achieve the same relative error
- However, efficient calculation does not mean it is accurate!
  - use biasing with care

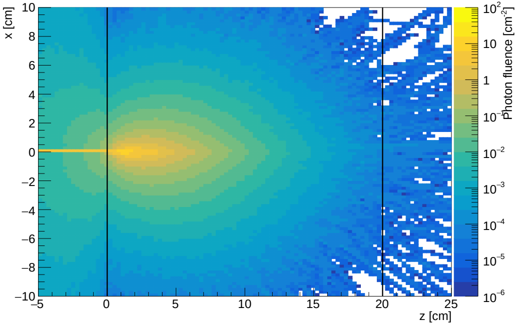
Example: photons on lead
Geometry



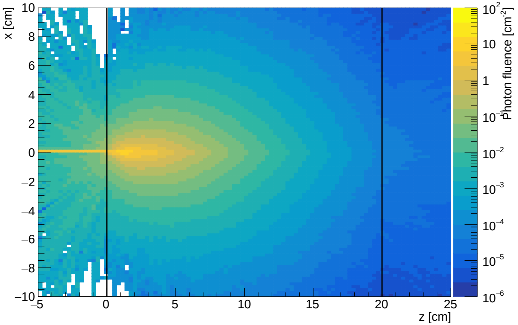
Example: photons on lead

Results: fluence

Analog run (no biasing used)



Biased run

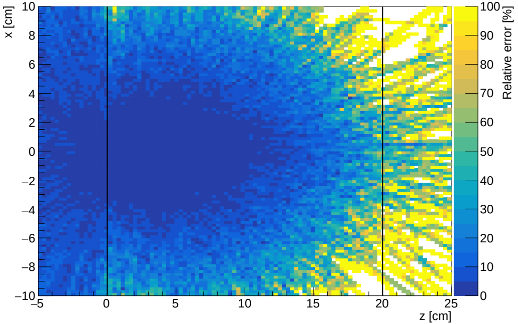


- Same CPU time (2.5 min)
- Number of histories: Analog: 37000 *vs* Biased: 600 (1.6 %)
- About 15 min is spent to setup variance reduction

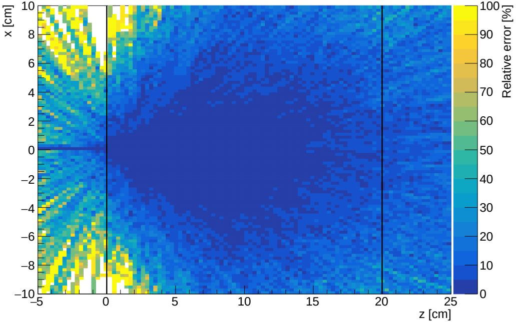
Example: photons on lead

Results: relative errors

Analog run (no biasing used)



Biased run



- Notice different track statistics on both sides of the wall
- Relative error at the wall exit plane is improved by the factor of 3

  $\Rightarrow$ biased calculation until the same uncertainty is 3×3 = 9 times faster

How to achieve this?

Main idea:

- You have tell FLUKA which tracks are more important than others
  - more important tracks are those which will eventually cause contribution to the estimator score
- This is done by adjusting the sampling of the random walk so that important random walks, which will contribute to the estimator score, are preferentially sampled over others
  - in our example, forward-going tracks are more important than backward-going tracks
- FLUKA will spend more computational effort on particles that contribute significantly to the quantities of interest and less effort on those that do not

How to achieve this?

- FLUKA has a rich set of tools to achieve this
- These tools are referred as biasing techniques
  - each technique has its own peculiarities:
    - area of use
    - advantages and disadvantages
- Some techniques cooperate with each other
- Others should not be used together
- To be successful, user must understand both
  - physics of the problem
  - and biasing methodology

Outline

Biasing techniques

FLUKA biasing techniques

- basic
  - Importance biasing (particle or region) — BIASING
  - Leading particle biasing ($e^{\pm}$ and photon interactions) — EMF-BIAS
  - Multiplicity tuning (hadronic interactions) — BIASING
  - Biasing mean free paths (hadronic interactions, decays) — LAM-BIAS
- advanced
  - Weight window
    - WW-FACTOr
    - WW-THRESh
    - WW-PROFIle
  - Biased downscattering for neutrons — LOW-DOWN
  - Non-alagoue absorptions — LOW-BIAS
  - User-defined biasing
    - usbset.f
    - usimbs.f

Biasing techniques

FLUKA biasing techniques

- basic
  - Importance biasing (particle or region) — BIASING
  - Leading particle biasing ($e^{\pm}$ and photon interactions) — EMF-BIAS
  - Multiplicity tuning (hadronic interactions) — BIASING
  - Biasing mean free paths (hadronic interactions, decays) — LAM-BIAS
- advanced
  - Weight window
    - WW-FACTOr
    - WW-THRESh
    - WW-PROFIle
  - Biased downscattering for neutrons — LOW-DOWN
  - Non-alagoue absorptions — LOW-BIAS
  - User-defined biasing
    - usbset.f
    - usimbs.f

Outline

Importance biasing
Usage area

Importance biasing is helpful in:

- Large geometries
- Problems with strong attenuation
  - e.g. deep penetration calculations

Importance biasing
Usage area

- It is simplest, "safest" and easiest to use of all biasing techniques
- The user assigns an importance to each geometry region, based on
  1. expected fluence attenuation with respect to other regions
  2. probability of contribution to detectors by particles entering the region
- Importances are relative to each other, i.e. only their ratios matter

Rule of thumb

- In deep penetration calculations, assign importance to a region **inversely proportional to its attenuation factor**
- This compensates for the dilution of particle density, resulting in a **more uniform particle population** across space

- Can be tuned per particle type
- Combines two processes:
  - Surface splitting — reduces R but increases T
  - Russian roulette — does the opposite

$$\boxed{R\sqrt{T} = C}$$

Three fundamental concepts

- Surface splitting
- Russian roulette
- Particle weight

Surface splitting and Russian roulette

General overview

- The number of particles is increased in important regions for better sampling and decreased in other regions

Splitting

- Splitting **increases** the number of particles where they are needed
- The new particles have **reduced weight to conserve the total weight**

Russian roulette

- Russian roulette **reduces** the number of particles where they are not needed
- To maintain the expected weight, **some particles survive but with increased weight**

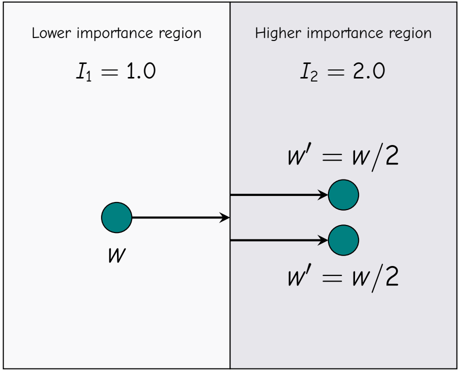
- The total weight of particles is conserved in both processes

Particle weight

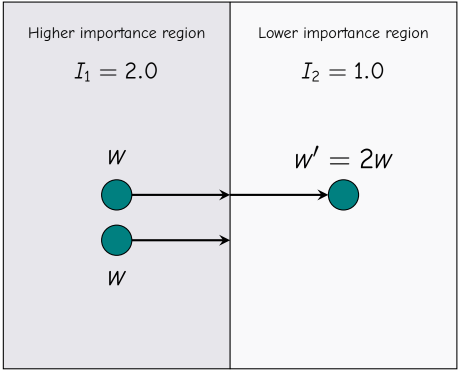
- Particle weight is a crucial component used to manage and optimise the simulation process
- It is a numerical factor assigned to each simulated particle
  - represents the statistical significance or contribution of that particle to the estimator score
  - normally, the user does not set the particle weight explicitly, but it is calculated by the code based on other biasing parameters (e.g. region importance)
- In an analog simulation, all particles are equally important
  - and therefore their weight is the same
- In a biased simulation, not all particles are equally important
  - particle weight helps to effectively manage the computational effort by forcing resources on more significant particles

Surface splitting

- When a particle enters a region of higher importance, it can be split into several particles, each with a reduced weight
  - the particle is split into $n = I_2/I_1$ identical particles with the same characteristics
- This increases the number of particles in important regions without increasing the total weight
  - the weight of each "daughter" particle is divided by I_2/I_1

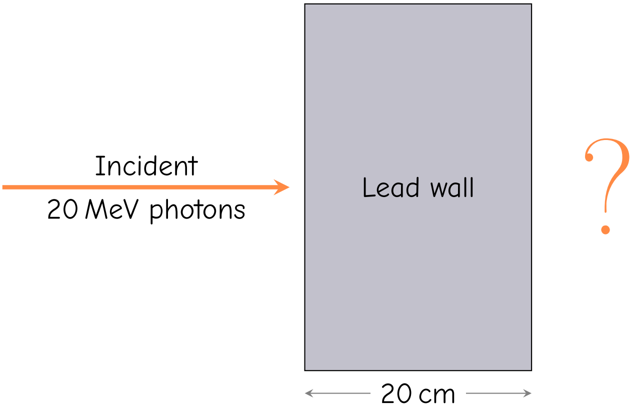


- When a particle enters a less important region, it may be subjected to Russian roulette
  - where it either survives with probability I_2/I_1
    - and its weight is increased by I_1/I_2 (to conserve the overall weight)
  - or it is terminated with probability $1 - I_2/I_1$
- Total weight of incoming and surviving particles is the same



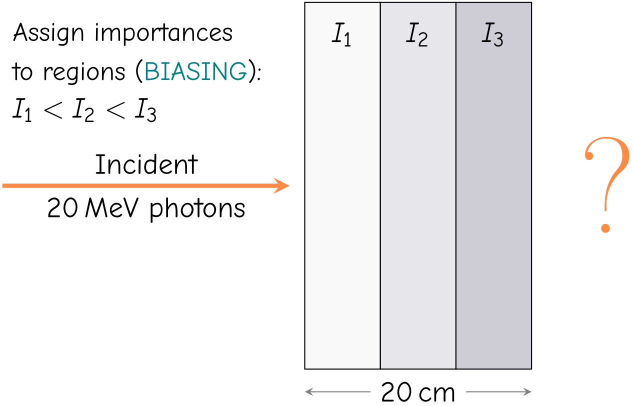
- When particles contribute to an estimator, their contributions are weighted by their particle weight
  - this ensures that the final results reflect the adjusted statistical significance of each particle

Example: photons on lead
Geometry



Example: photons on lead
Region importances

Assign importances to regions (BIASING):
$I_1 < I_2 < I_3$

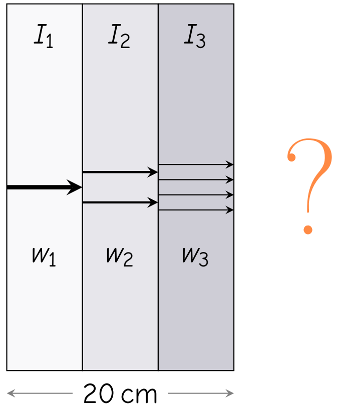


Example: photons on lead
Forward-going tracks: surface splitting

Assign importances
to regions (BIASING):
$I_1 < I_2 < I_3$

Incident
20 MeV photons

Particle weights:
$w_1 = 1$ (normally)
$w_2 = w_1 \cdot I_1 / I_2$
$w_3 = w_2 \cdot I_2 / I_3$

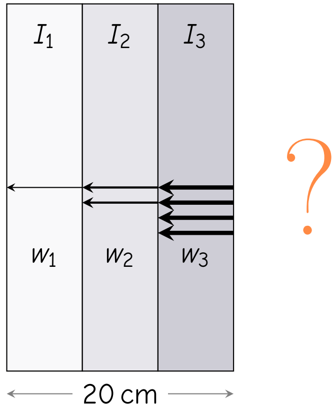


Example: photons on lead

Backward-going tracks: Russian roulette

Assign importances to regions (BIASING): $I_1 < I_2 < I_3$

Incident 20 MeV photons



Particle weights:

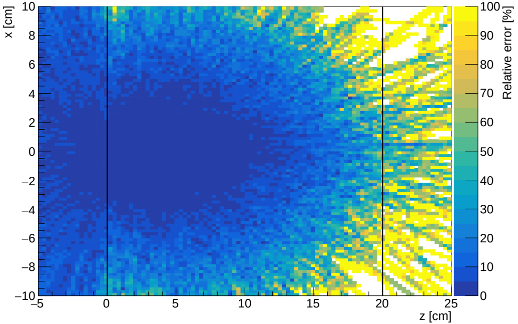
$w_3 = w_3$ (not changed)

$w_2 = w_3 \cdot I_3/I_2$

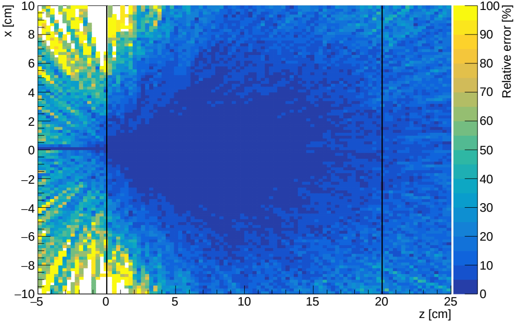
$w_1 = w_2 \cdot I_2/I_1$

Example: photons on lead
Results: relative errors

Analog run (no biasing used)



Biased run



- Notice different track statistics on both sides of the wall
- Relative error at the wall exit plane is improved by the factor of 3

  $\Rightarrow$ biased calculation until the same uncertainty is 3×3 = 9 times faster

Importance biasing
Input cards

BIASING	WHAT1	WHAT2	WHAT3	WHAT4	WHAT5	WHAT6	**SDUM**

The meaning of WHATs and SDUM depends on the WHAT(1) sign:

WHAT(1)

≥ 0.0 → region-based biasing

< 0.0 → particle-based biasing

Importance biasing

Input cards: region-based biasing

BIASING	WHAT1	WHAT2	WHAT3	WHAT4	WHAT5	WHAT6	**SDUM**

The meaning of WHATs and SDUM depends on the WHAT(1) sign:

WHAT(1)

≥ 0.0 → region-based biasing

< 0.0 → particle-based biasing

Importance biasing

Input cards: region-based biasing

BIASING	Particle	Mult	Imp	Region1	Region2	Step	**SDUM**

If WHAT(1) $\geq$ 0.0

WHAT(1) particles to be biased

- 0.0 all particles
- 1.0 hadrons and muons
- 2.0 electrons, positrons and photons
- 3.0 low energy neutrons

Importance biasing
Input cards: region-based biasing

BIASING	Particle	Mult	Imp	Region1	Region2	Step	**SDUM**

If WHAT(1) $\geq$ 0.0

WHAT(1) particles to be biased

- 0.0 all particles
- 1.0 hadrons and muons
- 2.0 electrons, positrons and photons
- 3.0 low energy neutrons

WHAT(2) see Multiplicity tuning

Importance biasing

Input cards: region-based biasing

BIASING	Particle	Mult	Imp	Region1	Region2	Step	SDUM

If WHAT(1) $\geq$ 0.0

WHAT(1) particles to be biased

- 0.0 all particles
- 1.0 hadrons and muons
- 2.0 electrons, positrons and photons
- 3.0 low energy neutrons

WHAT(2) see Multiplicity tuning

WHAT(3) region importance

- allowed range: 10^{-4} to 10^{5}

BIASING	Particle	Mult	Imp	Region1	Region2	Step	SDUM

If WHAT(1) $\geq$ 0.0

WHAT(4) first region

WHAT(5) last region

WHAT(6) region index step

BIASING	Particle	Mult	Imp	Region1	Region2	Step	SDUM

If WHAT(1) $\geq$ 0.0

SDUM option:

- PRINT: importance biasing counters are printed
- NOPRINT: importance biasing counters are not printed
- USER: use the USer-defined IMportance BiaSing subroutine USIMBS ($FLUPRO/usermvax/usimbs.f)
- NOUSER: cancel any previous USER request
- PRPRONLY: multiplicity biasing for PRimary PaRticles ONLY
  - otherwise biasing is applied to all generations of particles

Importance biasing
Input cards: particle-based biasing

BIASING	WHAT1	WHAT2	WHAT3	WHAT4	WHAT5	WHAT6	**SDUM**

The meaning of WHATs and SDUM depends on the WHAT(1) sign:

WHAT(1)

≥ 0.0 → region-based biasing

< 0.0 → particle-based biasing

Importance biasing
Input cards: particle-based biasing

BIASING	Flag	Factor	Particle1	Particle2	Step	—	**SDUM**

If WHAT(1) < 0.0

WHAT(1) flag indicating that all region importances shall be modified by a particle-dependent factor

BIASING	Flag	Factor	Particle1	Particle2	Step	—	**SDUM**

If WHAT(1) < 0.0

WHAT(1) flag indicating that all region importances shall be modified by a particle-dependent factor

WHAT(2) modifying factor

Importance biasing

Input cards: particle-based biasing

BIASING	Flag	Factor	Particle1	Particle2	Step	—	SDUM

If WHAT(1) $<$ 0.0

- WHAT(1) flag indicating that all region importances shall be modified by a particle-dependent factor
- WHAT(2) modifying factor
- WHAT(3) lower bound of particle indexes/names
- WHAT(4) upper bound of particle indexes/names
- WHAT(5) particle index step length

Importance biasing

Input cards: particle-based biasing

BIASING	Flag	Factor	Particle1	Particle2	Step	—	**SDUM**

If WHAT(1) < 0.0

WHAT(1)	flag indicating that all region importances shall be modified by a particle-dependent factor
WHAT(2)	modifying factor
WHAT(3)	lower bound of particle indexes/names
WHAT(4)	upper bound of particle indexes/names
WHAT(5)	particle index step length
WHAT(6)	not used

Importance biasing

Input cards: particle-based biasing

BIASING	Flag	Factor	Particle1	Particle2	Step	—	SDUM

If WHAT(1) < 0.0

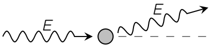
SDUM option

- NOPRIMARy importance biasing is applied only to the secondary particles
- PRIMARY importance biasing is applied also to primaries

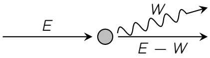
Outline

Leading particle biasing (LPB)

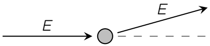
- Only available for electrons, positrons and photons
- Generally used to prevent the geometric increase in the number of particles in an electromagnetic shower with energy
- It is characteristic of most electromagnetic interactions that two particles are present in the final state:



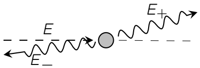
Rayleigh scattering



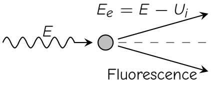
Bremßtrahlung



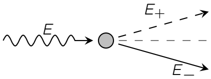
Elastic scattering



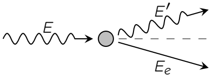
Positron annihilation



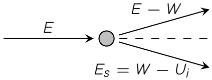
Photoelectric effect



Pair production



Compton scattering



Inelastic scattering

Leading particle biasing (LPB)

- Only one of the two particles is randomly retained, and its weight is adjusted to conserve the product of weight and probability
  - The most energetic of the two particles is kept with higher probability, as it is more efficient in propagating the shower
- LPB is aimed at reducing the mean CPU time per history (T) rather than the relative error R by introducing weight fluctuations.
- If its application is not limited below a suitable energy threshold, it should be backed up by weight windows (see Advanced course).

Leading particle biasing (LPB)

Input cards

EMF-BIAS	WHAT1	WHAT2	WHAT3	WHAT4	WHAT5	WHAT6	**SDUM**

The meaning of WHATs depends on the SDUM card value:

SDUM

LPBEMF   Lambda biasing

LAMBEMF   LAMBCOMP   LAMBBREM   Russian Roulette

LBRREMF   LBRRCOMP   LBRRBREM

- Only LPBEMF is covered in this course
  - EMF = ElectroMagnetic FLUKA

EMF-BIAS	Reaction	$e^{\pm}$	Photons	Region1	Region2	Step	**LPBEMF**

WHAT(1) defines reactions to be biased by the binary mask: $\sum_{i=0}^{9} 2^i b_i$

$b_0 = 1$ bremsstrahlung and pair production
$b_1 = 1$ bremsstrahlung
$b_2 = 1$ pair production
$b_3 = 1$ positron annihilation at rest
$b_4 = 1$ Compton scattering
$b_5 = 1$ Bhabha & Møller scattering $(e^+e^-\ \&\ e^-e^-)$
$b_6 = 1$ photoelectric effect
$b_7 = 1$ positron annihilation in flight
$b_{8,9}$ not used (reserved)

Note: $(1111111110)_2 = 1022$ – activates leading particle biasing for all physical effects

EMF-BIAS	Reaction	$e^{\pm}$	Photons	Region1	Region2	Step	**LPBEMF**

WHAT(2) energy threshold (max. energy) for electrons and positrons

- >0.0 energy threshold below which LPB is applied to $e^{\pm}$
- <0.0 resets any previously defined threshold to infinity (i.e. LPB is applied at all energies, default)

EMF-BIAS	Reaction	$e^{\pm}$	Photons	Region1	Region2	Step	**LPBEMF**

WHAT(2) energy threshold (max. energy) for electrons and positrons

- >0.0 energy threshold below which LPB is applied to $e^{\pm}$
- <0.0 resets any previously defined threshold to infinity (i.e. LPB is applied at all energies, default)

WHAT(3) energy threshold (max. energy) for photons

- >0.0 energy threshold below which LPB is applied to photons
- <0.0 resets any previously defined threshold to infinity (i.e. LPB is applied at all energies, default)

Leading particle biasing (LPB)
Input cards

EMF-BIAS	Reaction	$e^{\pm}$	Photons	Region1	Region2	Step	**LPBEMF**

WHAT(4) first region

WHAT(5) last region

WHAT(6) region index step

- EMF-BIAS concerns electromagnetic interactions only.
  - Photonuclear interaction biasing is provided by LAM-BIAS
- It is mainly used to:
  - Estimate the shower punch-through
    - but even better efficiency can be obtained with Importance splitting
  - Reduce time (T) spent in simulating secondary electromagnetic showers produced by π^0 in hadronic cascades
- Must be used with care (as any other kind of biasing):
  - The radial shower profile might be reproduced less accurately (Molière radius)
    - this is not very important for hadronic showers since their lateral spread is governed essentially by hadrons
  - Few low-energy particles might result with a very large weight causing large energy deposition fluctuations
    - therefore, LPB should be avoided in regions when using energy deposition bins of very small volume
    - for energy deposition calculations, recommended to be backed up by weight windows

Outline

Multiplicity tuning

- Multiplicity tuning is meant to be for hadrons what Leading Particle Biasing is for $e^{\pm}$ and photons

Multiplicity tuning

- Multiplicity tuning is meant to be for hadrons what Leading Particle Biasing is for $e^{\pm}$ and photons
- A high-energy hadronic nuclear interaction can produce numerous secondaries

Multiplicity tuning

- Multiplicity tuning is meant to be for hadrons what Leading Particle Biasing is for $e^{\pm}$ and photons
- A high-energy hadronic nuclear interaction can produce numerous secondaries
  - Therefore, simulating an entire hadronic cascade in bulk matter may be very time consuming

Multiplicity tuning

- Multiplicity tuning is meant to be for hadrons what Leading Particle Biasing is for $e^{\pm}$ and photons
- A high-energy hadronic nuclear interaction can produce numerous secondaries
  - Therefore, simulating an entire hadronic cascade in bulk matter may be very time consuming
- Many secondaries in the hadronic cascade are of the same type and have similar energies and other characteristics

Multiplicity tuning

- Multiplicity tuning is meant to be for hadrons what Leading Particle Biasing is for $e^{\pm}$ and photons
- A high-energy hadronic nuclear interaction can produce numerous secondaries
  - Therefore, simulating an entire hadronic cascade in bulk matter may be very time consuming
- Many secondaries in the hadronic cascade are of the same type and have similar energies and other characteristics
- Therefore, it is feasible to discard a predefined average fraction of them, given that the weight of those retained and transported is adjusted to maintain total weight conservation (though the leading particle is always retained)

Multiplicity tuning

- Multiplicity tuning is meant to be for hadrons what Leading Particle Biasing is for $e^{\pm}$ and photons
- A high-energy hadronic nuclear interaction can produce numerous secondaries
  - Therefore, simulating an entire hadronic cascade in bulk matter may be very time consuming
- Many secondaries in the hadronic cascade are of the same type and have similar energies and other characteristics
- Therefore, it is feasible to discard a predefined average fraction of them, given that the weight of those retained and transported is adjusted to maintain total weight conservation (though the leading particle is always retained)
- The user can adjust the average multiplicity for selected particle types by specifying a region-dependent factor (normally < 1)

Multiplicity tuning
Input cards

Same options as Importance biasing

BIASING	Particle	Mult	Imp	Region1	Region2	Step	**SDUM**

WHAT(2) Russian Roulette (or splitting) factor by which the average number of secondaries produced in a collision should be reduced (or increased):

$I_2 < I_1$ Russian roulette

$I_2 > I_1$ Splitting

Outline

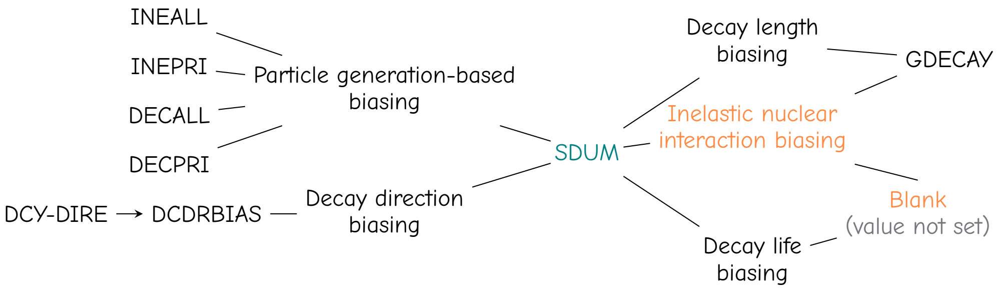
Biasing mean free paths

- In a similar way, the hadron or photon mean free path for non-elastic nuclear interactions can be artificially decreased by a predefined particle- or material-dependent factor
- This option is useful to increase the probability for beam interaction in a
  - very thin target
  - material of very low density
  - material with very low interaction cross section
    - e.g. to simulate photonuclear reactions with acceptable statistics since the photonuclear cross section is much smaller than that for electromagnetic processes

Biasing mean free paths

Input cards

LAM-BIAS	WHAT1	WHAT2	WHAT3	WHAT4	WHAT5	WHAT6	SDUM



- The meaning of WHATs depend on the SDUM value
- In this lecture, we will only consider Inelastic nuclear interaction biasing

Biasing mean free paths
Input cards

LAM-BIAS	$T_{1/2}$	MFP	Material	Particle1	Particle2	Step

WHAT(1) decay time biasing factor (for unstable particles only)

LAM-BIAS	$T_{1/2}$	MFP	Material	Particle1	Particle2	Step

WHAT(1) decay time biasing factor (for unstable particles only)

WHAT(2) biasing factor for hadronic inelastic interaction length (normally < 1)

Biasing mean free paths
Input cards

LAM-BIAS	$T_{1/2}$	MFP	Material	Particle1	Particle2	Step

WHAT(1) decay time biasing factor (for unstable particles only)

WHAT(2) biasing factor for hadronic inelastic interaction length (normally < 1)

WHAT(3) material to which the biasing is to be applied

LAM-BIAS	$T_{1/2}$	MFP	Material	Particle1	Particle2	Step

- WHAT(1) decay time biasing factor (for unstable particles only)
- WHAT(2) biasing factor for hadronic inelastic interaction length (normally < 1)
- WHAT(3) material to which the biasing is to be applied
- WHAT(4) first particle
- WHAT(5) last particle
- WHAT(6) particle index step

Outline

Other optimisation checks

- Check CPU-expensive physics options:
  - energy cuts
  - step sizes
  - list of transported particles
  - see the Ionization and Transport lecture and the Cutoffs exercise
- Geometry truncation
  - avoid modelling entire Universe!
    - limit your geometry to the areas relevant to your problem

Outline

Biasing techniques

FLUKA biasing techniques

- basic
  - Importance biasing (particle or region) — BIASING
  - Leading particle biasing ($e^{\pm}$ and photon interactions) — EMF-BIAS
  - Multiplicity tuning (hadronic interactions) — BIASING
  - Biasing mean free paths (hadronic interactions, decays) — LAM-BIAS
- advanced
  - Weight window
    - WW-FACTOr
    - WW-THRESh
    - WW-PROFIle
  - Biased downscattering for neutrons — LOW-DOWN
  - Non-alagoue absorptions — LOW-BIAS
  - User-defined biasing
    - usbset.f
    - usimbs.f

Conclusions

- All these biasing techniques are intended to improve statistics in some parts of phase space at the expenses of the other parts
- Biased runs in particular can neither accelerate convergence in all regions, nor reproduce natural fluctuations and correlations
- Biasing can be very powerful, but …
  - do not bias unless you know what you are doing ← you are warned!
  - always check whether your results make sense

www.fluka.org

Bibliography

Alfredo Ferrari, Paola R. Sala, Alberto Fassò, Johannes Ranft
FLUKA: a multi-particle transport code (FLUKA Manual)
2024

Joel A. Kulesza *et al*
MCNP 6.3.0 Theory & User Manual
2022
Available online, contains an excellent description of general Monte Carlo theory, most of which applies also to FLUKA

E. J. McGrath and D. C. Irving
Techniques for efficient Monte Carlo simulation. Volume III: Variance reduction.
1975