Geant 4

Biasing

http://geant4.org

Event biasing (1/2)

- What is **analogue** simulation ?
  - Sample using natural probability distribution, N(x)
  - Predicts mean with correct fluctuations
  - Can be inefficient for certain applications
- What is non-analogue/**event biased** simulation ?
  - **Cheat** - apply artificial biasing probability distribution, B(x) in place of natural one, N(x)
    - B(x) enhances production of whatever it is that is interesting
    - To get meaningful results, must apply a **weight correction**
  - Predicts same analogue mean with smaller variance
    - Increases efficiency of the Monte Carlo
  - Does **not** predict correct fluctuations
  - Should be used **with care**

Event biasing (2/2)

- Geant4 provides **built-in** general use **biasing techniques**
- The effect consists in producing a small number of secondaries, which are artificially recognized as a huge number of particles by their statistical weights → **reduce CPU time**
- Event biasing can be used, for instance, for the transportation of particles through a thick shielding
- An utility class `G4WrapperProcess` supports user-defined biasing

Event biasing techniques (1)

- Production cuts / threshold
  - This is a biasing technique – most popular for many applications: set high cuts to reduce secondary production
- Geometry based biasing
  - Importance weighting for volume/region
  - Duplication or sudden death of tracks
- Primary event biasing
  - Biasing primary events and/or primary particles in terms of type of event, momentum distribution → generate *only primaries* that can produce *events that are interesting for you*

Event biasing techniques (2)

- Forced interaction
  - Force a particular interaction, e.g. within a volume
- Enhanced process or channel and physics-based biasing
  - Increasing cross section for a given process (e.g.bremsstrahlung)
  - Biasing secondary production in terms of particle type, momentum distribution, cross-section, etc.
- Leading particle biasing
  - Take into account only the most energetic (or most important) secondary
  - Currently NOT supported in Geant4

Variance Reduction

- Use variance reduction techniques to **reduce computing time** taken to calculate a result with a given variance (= statistic error)
- Want to **increase efficiency** of the Monte Carlo
- Measure of efficiency is given by

$$\varepsilon = \frac{1}{s^2 T}$$

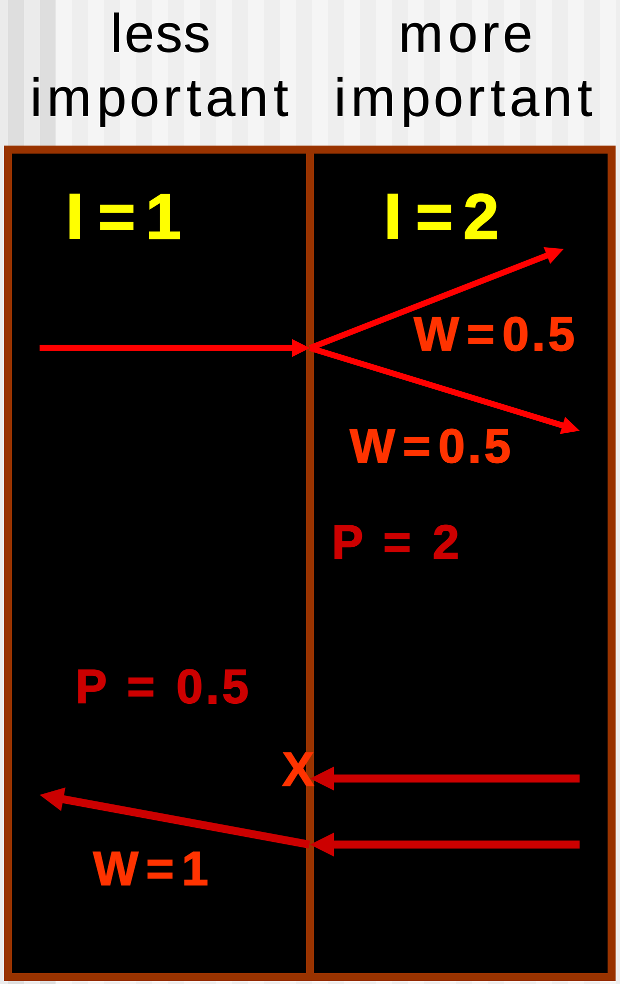
s = variance on calculated quantity
T = computing time

Geometric Biasing

The purpose of geometry-based event biasing is to save computing time by sampling less often the particle histories entering “less important” geometry regions, and **more often in more “important” regions**.

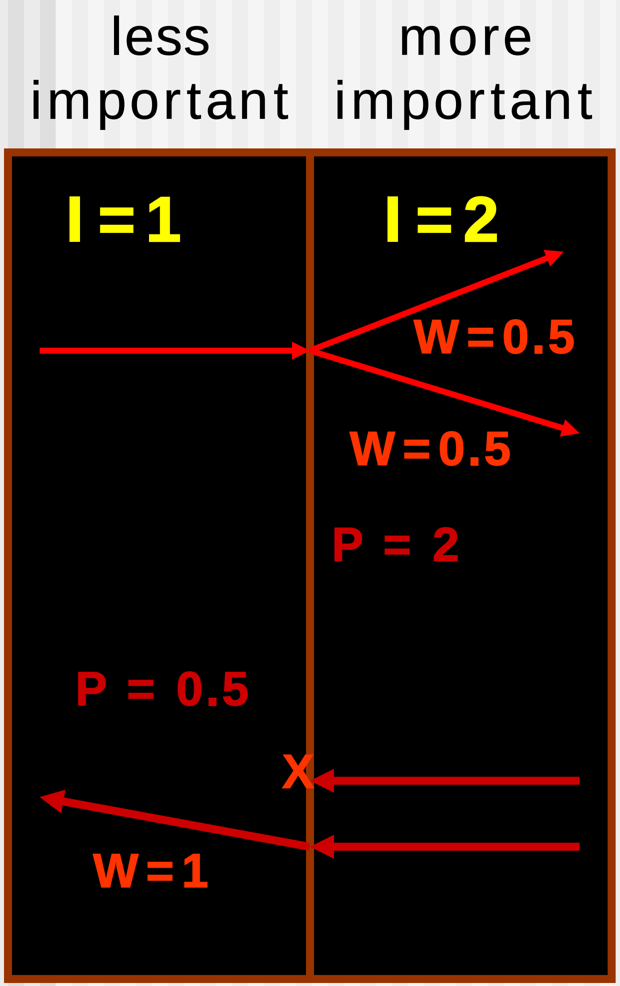
* Importance sampling technique
* Weight window technique

Importance sampling technique (1)



- Importance sampling acts on particles **crossing boundaries** between “importance cells”.
- The action taken depends on the importance value (I) assigned to the cell.
- In general, a track is played either **split** or **Russian roulette** at the geometrical boundary depending on the importance value assigned to the cell.

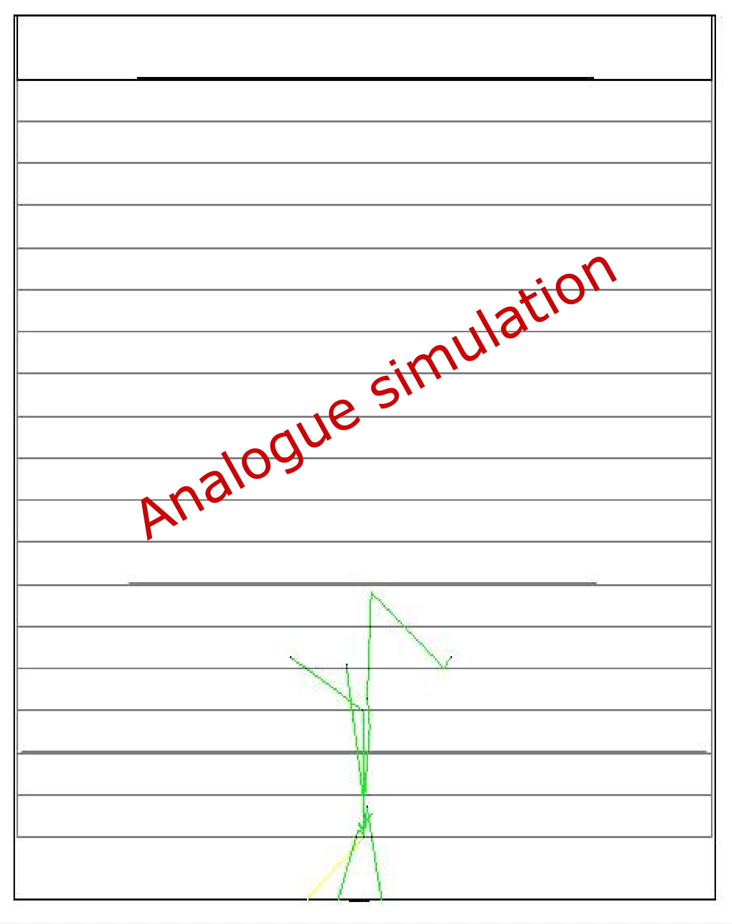
Importance sampling technique (2)

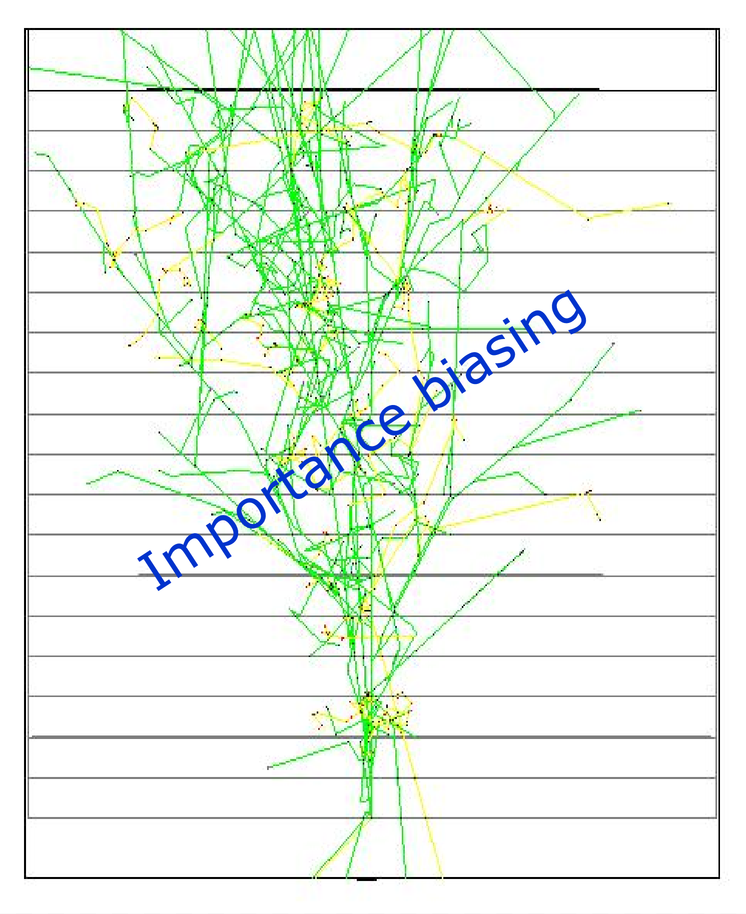


- Survival probability (P) is defined by the ratio of importance value
  $P = I_{post} / I_{pre}$
- The **track weight** is changed to W/P (weight necessary to get correct results at the end!)
- If **P>1**: splitting a track
  - E.g. creating two particles with half the 'weight' if it moves into volume with double importance value.
- If **P<1**: Russian-roulette in opposite direction
  - E.g. Kill particles according to the survival probability (1 - P).

Importance biasing

increasing importance





10 MeV neutron in thick concrete cylinder

Physics biasing

- Built-in cross section biasing for PhotoInelastic, ElectronNuclear and PositronNuclear processes

```
G4ElectroNuclearReaction * theeReaction = new G4ElectroNuclearReaction;
G4ElectronNuclearProcess theElectronNuclearProcess;
theElectronNuclearProcess.RegisterMe(theeReaction);
theElectronNuclearProcess.BiasCrossSectionByFactor(100);
```

- Similar tool for rare EM processes (e^+e^- annihilation to μ pair or hadrons, γ conversion to $\mu^+\mu^-$)

```
G4AnnihiToMuPair* theProcess = new G4AnnihiToMuPair();
theProcess->SetCrossSecFactor(100);
```

- It is possible to introduce these factors for all EM processes, with a definition of customized processes that inherit from the "normal" ones (→ extended example)

- **Artificially** enhance/reduce cross section of a process (useful for thin layer interactions or thick layer shielding)

How to learn more about biasing

There are **examples** in Geant4, to show how to use the most common biasing techniques:

examples/extended/biasing

geometry-based biasing

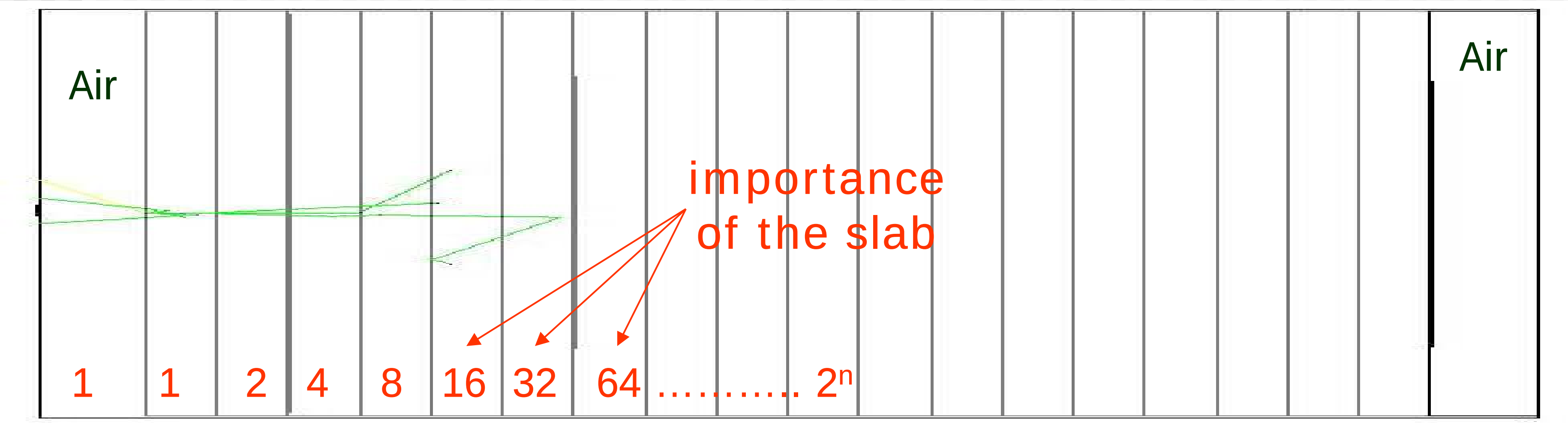
examples/extended/medical/fanoCavity

cross-section biasing (Compton scattering)

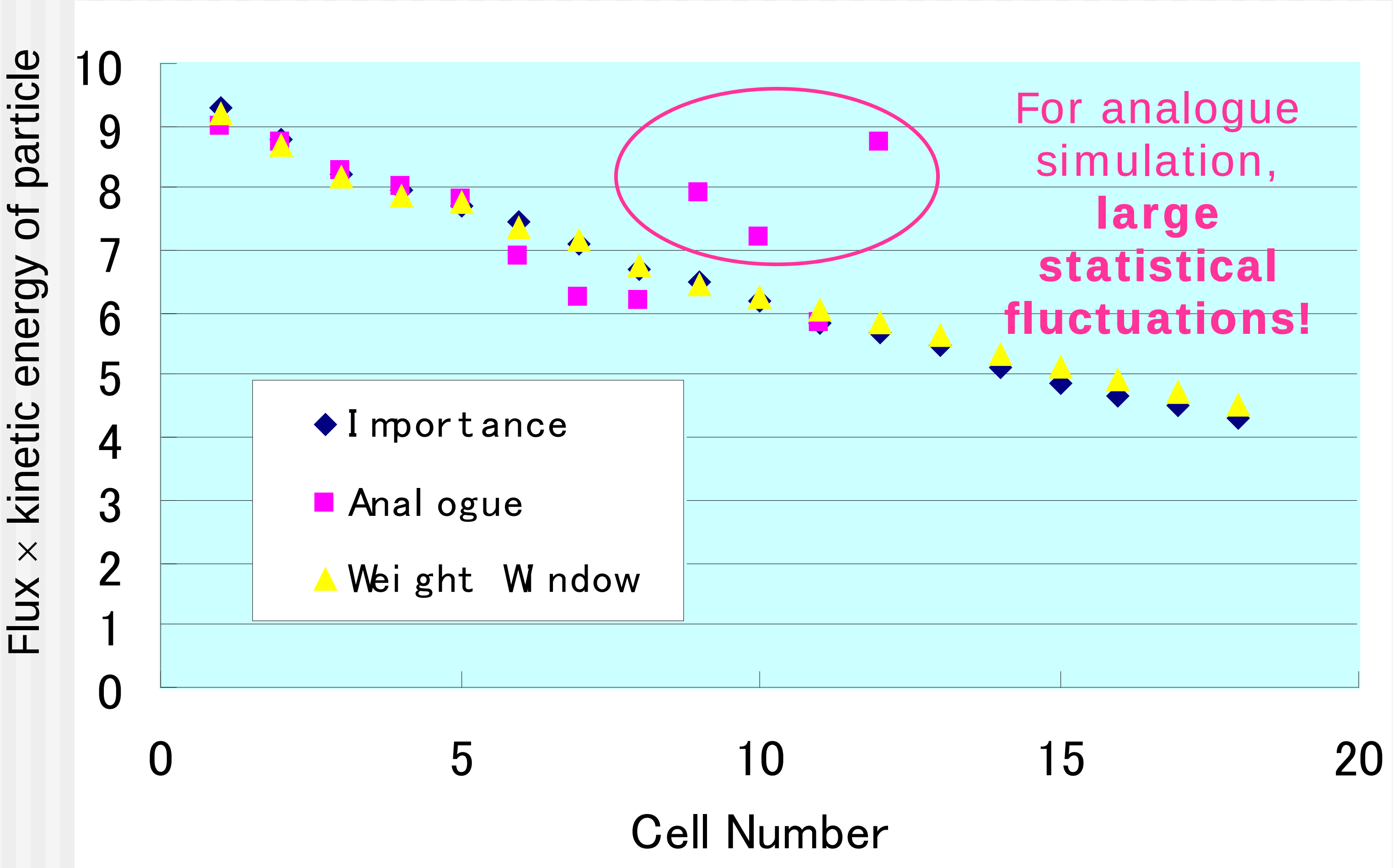
Additional documentation about biasing techniques available in the Geant4 User Guide, section 3.7

Biasing example B01

- Shows the **importance sampling** in the mass (tracking) geometry
- 10 MeV neutron shielding by cylindrical thick concrete
- 80 cm high concrete cylinder divided into 18 slabs (importance values assigned in the DetectorConstruction for simplicity)



Results of example B01



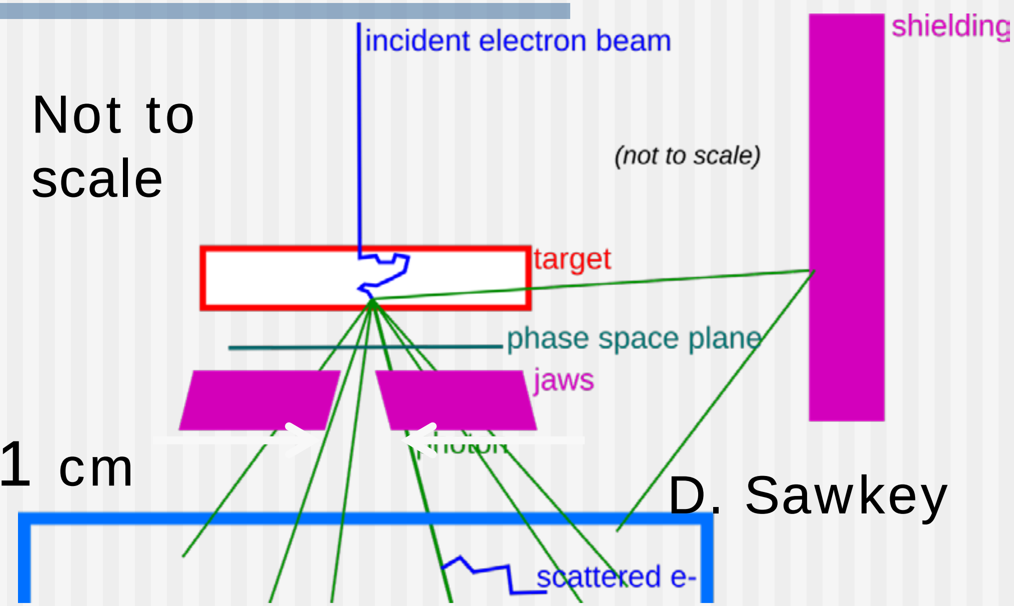
Built-in biasing options

- Cross section biasing, forced interactions, splitting of final state, **Russian roulette**
- **Common interface** (UI and C++) to apply them on the top of any EM configuration

Built-in biasing options

- Example of secondary **particle splitting**: medical linear accelerator
  - N photons (with weight 1/N) created in each **bremsstrahlung** interaction
  - Sampled independently
  - Energy spectrum is reproduced with high accuracy
  - Speed-up factor depends on geometry and cuts (up to factor on 8.5 here)

Not to scale



1 cm

D. Sawkey