Event biasing – An Overview

1. **Geometrical Event Biasing – Overview**
2. **Importance Sampling**
3. **Weight Window and Energy biasing**
4. **Examples**
5. **Parallel Navigation**
6. **Physics Based Biasing**
7. **Scoring**
8. **Summary**

Alex Howard, CERN

Event Biasing Overview

GATE meeting, Hebden Bridge 12th September 2007

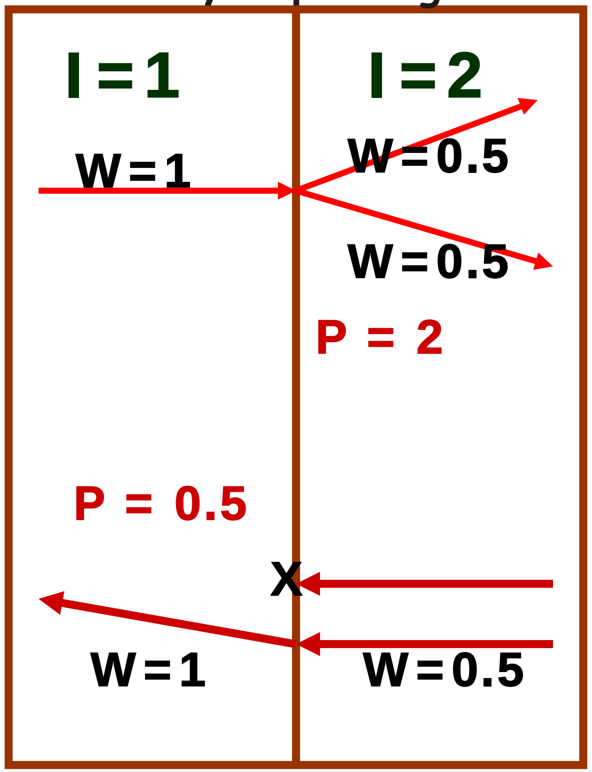
Geometric Biasing

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

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



- Survival probability (P) is defined by the ratio of importance value.
  $P = I_{post} / I_{pre}$
- The track weight is changed to W/P.

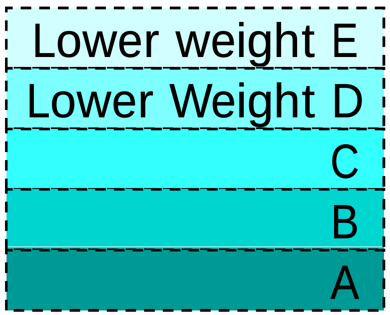
- Splitting a track (P > 1)
  - E.g. creating two particles with half the 'weight' if it moves into volume with double importance value.

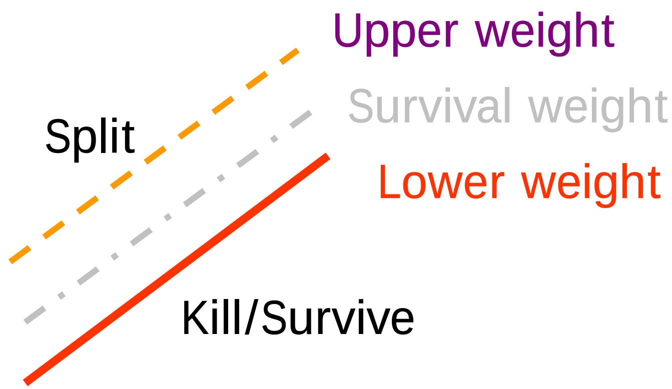
- Russian-roulette (P < 1) in opposite direction
  - E.g. Kill particles according to the survival probability (1 - P).

The Weight Window Technique

- **The weight window technique is a weight-based algorithm – generally used together with other techniques as an alternative to importance sampling:**
  - **It applies splitting and Russian roulette depending on space (cells) and energy**
  - **User defines weight windows in contrast to defining importance values as in importance sampling**

- ***A weight window may be specified for every cell and for several energy regions: space-energy cell .***

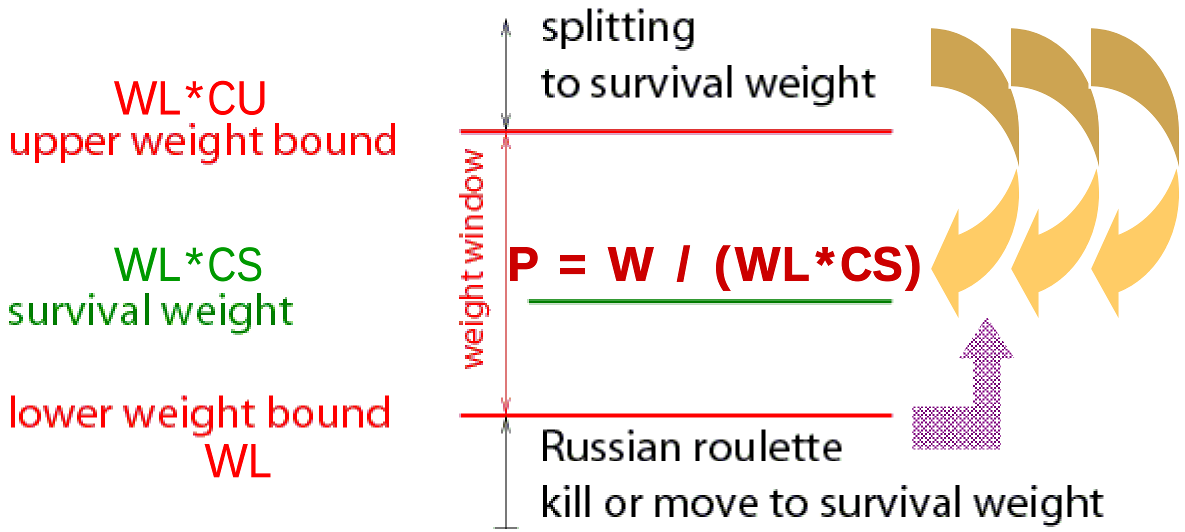




- Apply in combination with other techniques such as cross-section biasing, leading particle and implicit capture, or combinations of these.

The weight window technique (continue)

- Checks the particle weight
  - Compare the particle weight with a ‘window’ of weights defined for the current *energy-space* cell
  - Play splitting or roulette in case if it is outside, resulting in 0 or more particles ‘inside’ the window
    - E.g. WL is a lower weight bound of a cell.
      CU and CS are upper limit factor and survival factor, respectively.
      - W > WL*CU Split track
      - W < WL*WL Roulette

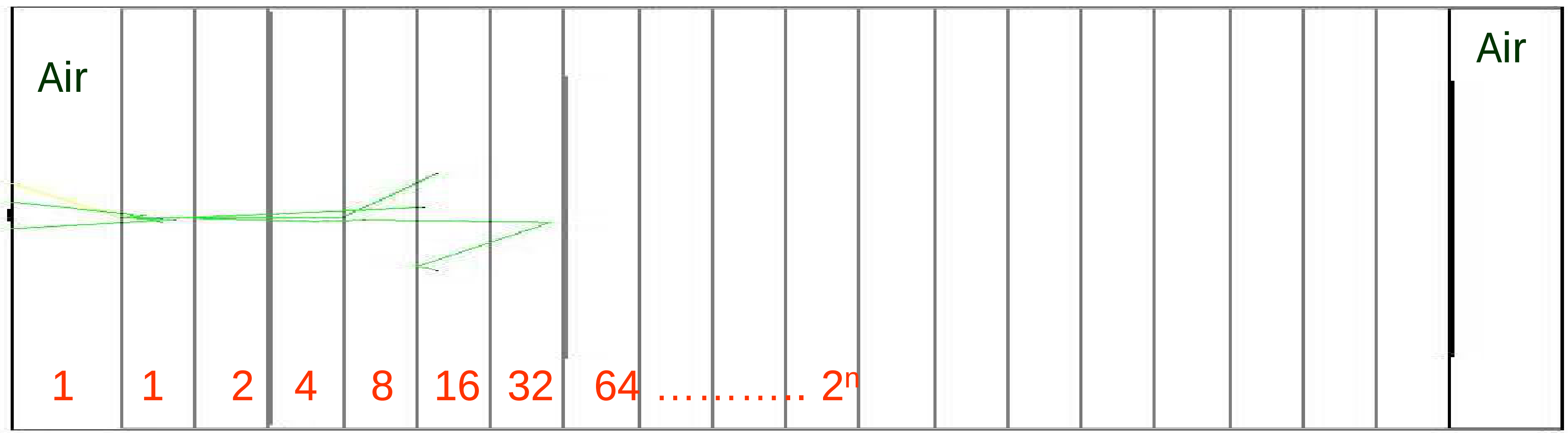


Examples

- **Extended/Biasing contains 2 examples of biasing**
  - **B01 (one geometry biasing)**
  - **B02 (biasing in the parallel world)**
    - **Both looking at 10MeV neutrons travelling through concrete shielding**

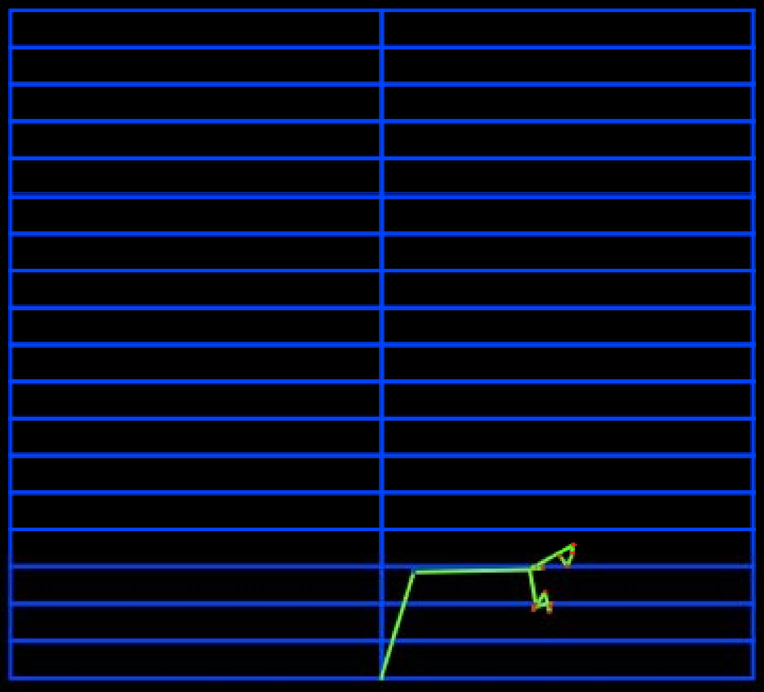
Biasing example B01

- Shows the **importance sampling** in the mass (tracking) geometry
- Option to show weight window
- 10 MeV neutron shielding by cylindrical thick concrete material
- Geometry
  - 80 cm high concrete cylinder divided into 18 slabs
  - Importance values assigned to 18 concrete slabs in the DetectorConstruction for simplicity.
  - The G4Scorer is used for the checking result
    - Top level class uses the framework provided for scoring.

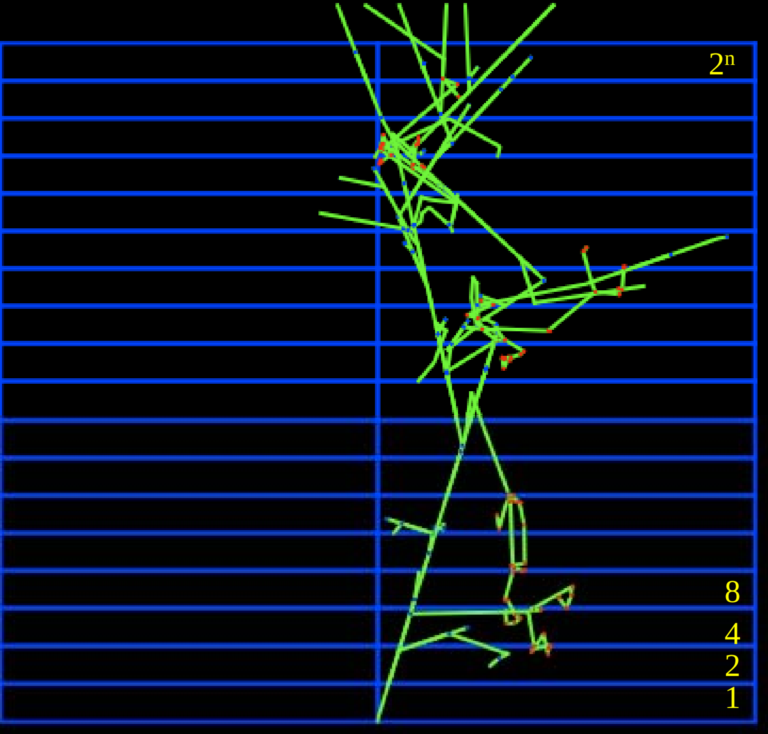


Example B01 - 10 MeV neutrons, thick concrete cylinder

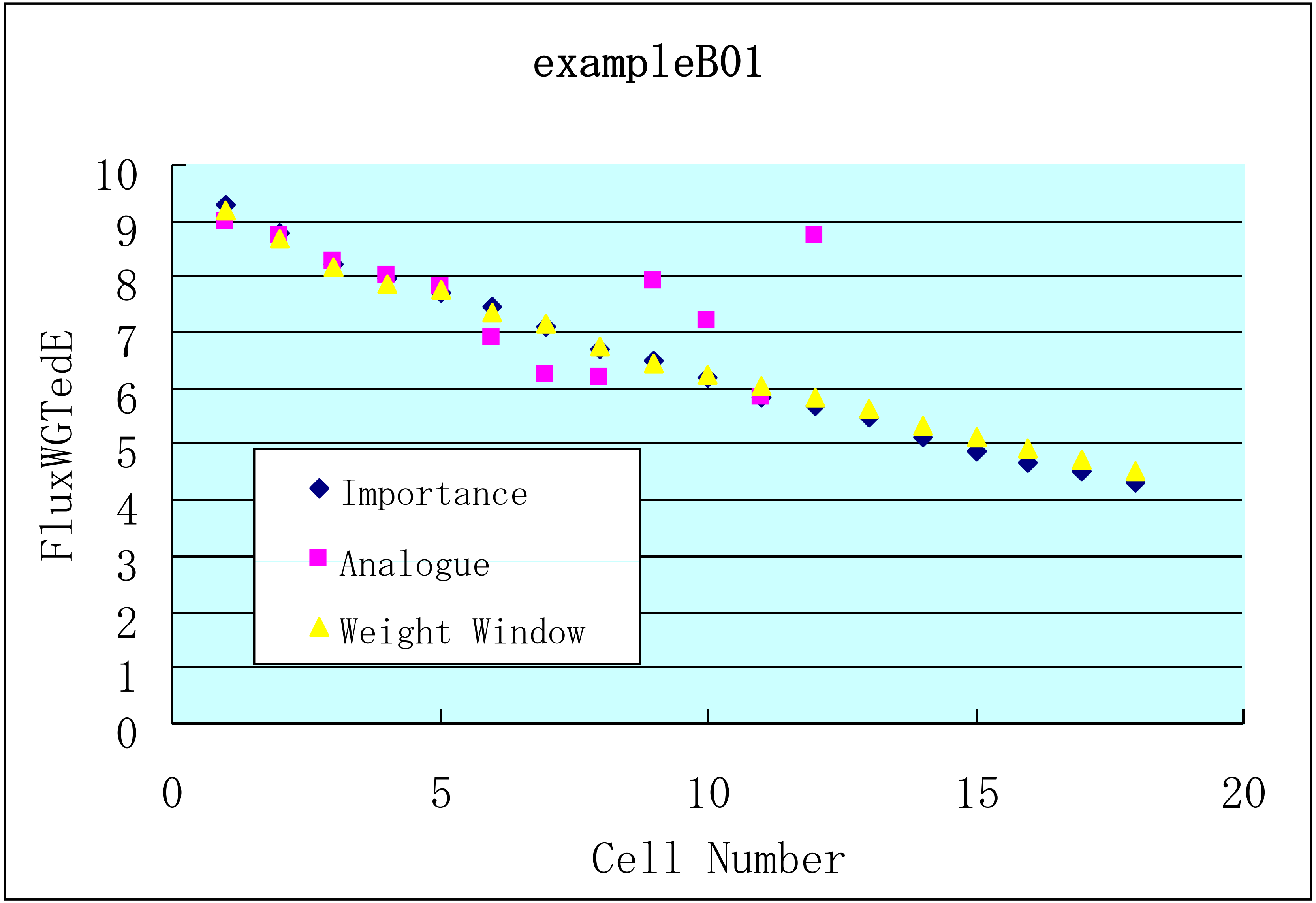
Analogue Simulation



Importance Sampled



Geant 4


exampleB01
FluxWGTedE
0
1
2
3
4
5
6
7
8
9
10
Importance
Analogue
Weight Window
0
5
10
15
20
Cell Number

Example B02

- Shows importance sampling in a parallel geometry
- Includes a customized scoring making use of the scoring framework
- Mass geometry consists of a 180 cm high simple bulk concrete cylinder
- A parallel geometry is created to hold importance values for slabs of width 10cm and for scoring
- The scoring uses the G4CellSCorer and one customized scorer for the last slab

Coupled Transportation

- Since release 8.2 coupled transportation has been included
- This is a generic form of parallel navigation
- Geometrical biasing was migrated/copied to this formalism
  - No longer duplicate mass and parallel classes/samplers
  - Switch between mass and parallel world is through transportation assignation
  - Examples also migrated – release 9.0
- Examples also run for charged particles (and should handle magnetic fields)

Future Developments?
Other forms of geometrical biasing

- Point tallies? Particles directed to a point?
- Modified sampling methods? (exponential transform, implicit capture, forced collisions, source biasing): to sample from any arbitrary distribution rather than the physical probability as long as the particle weights are then adjusted to compensate
- Others?

Summary

- **Two main geometrical importance biasing techniques are implemented within Geant4**
- **Extension to charged particles and fields is possible within the context of Coupled Transportation**
- **Examples provided of functionality within the extended and advanced example categories**
- **Other use cases, developments required?**

Physics Based Biasing

- Built in biasing options
  - Primary particle biasing
  - Radioactive decay biasing
  - General hadronic leading particle biasing
  - Hadronic cross section biasing

- User defined biasing
  - G4WrapperProcess

- See release documentation for detailed instructions on how to implement the various biasing techniques

Built in Biasing Options

Biasing Technique	Since Release
Primary particle biasing	3.0
Radioactive decay biasing	3.0
General hadronic lead particle biasing	4.3
Hadronic cross section biasing	4.3
Geometrical Importance sampling	5.0
Geometrical weight window and weight cutoff	5.2

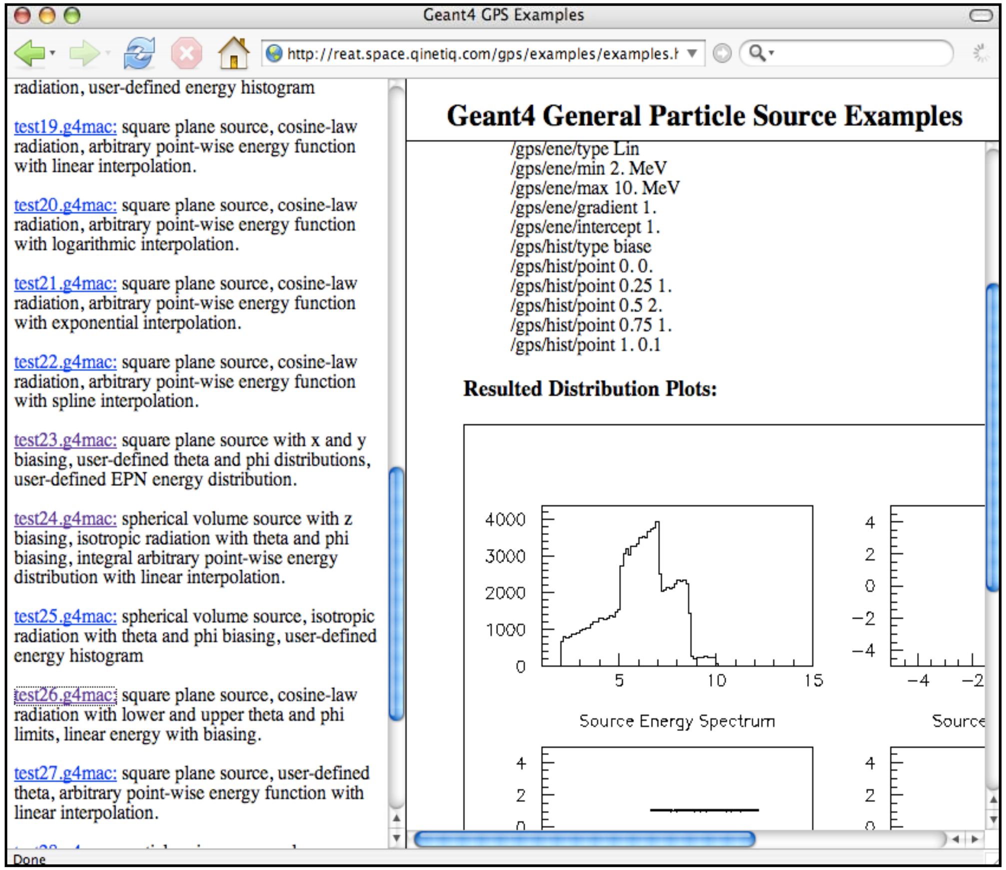
Primary Particle Biasing

- Use case:
  - Increase number of high energy particles in cosmic ray spectrum

- Increase number of primary particles generated in a particular phase space region of interest
  - Weight of primary particle modified as appropriate

- General implementation provided by G4GeneralParticleSource class
  - Bias position, angular and energy distributions

- Online documentation & examples
  - http://reat.space.qinetiq.com/gps/



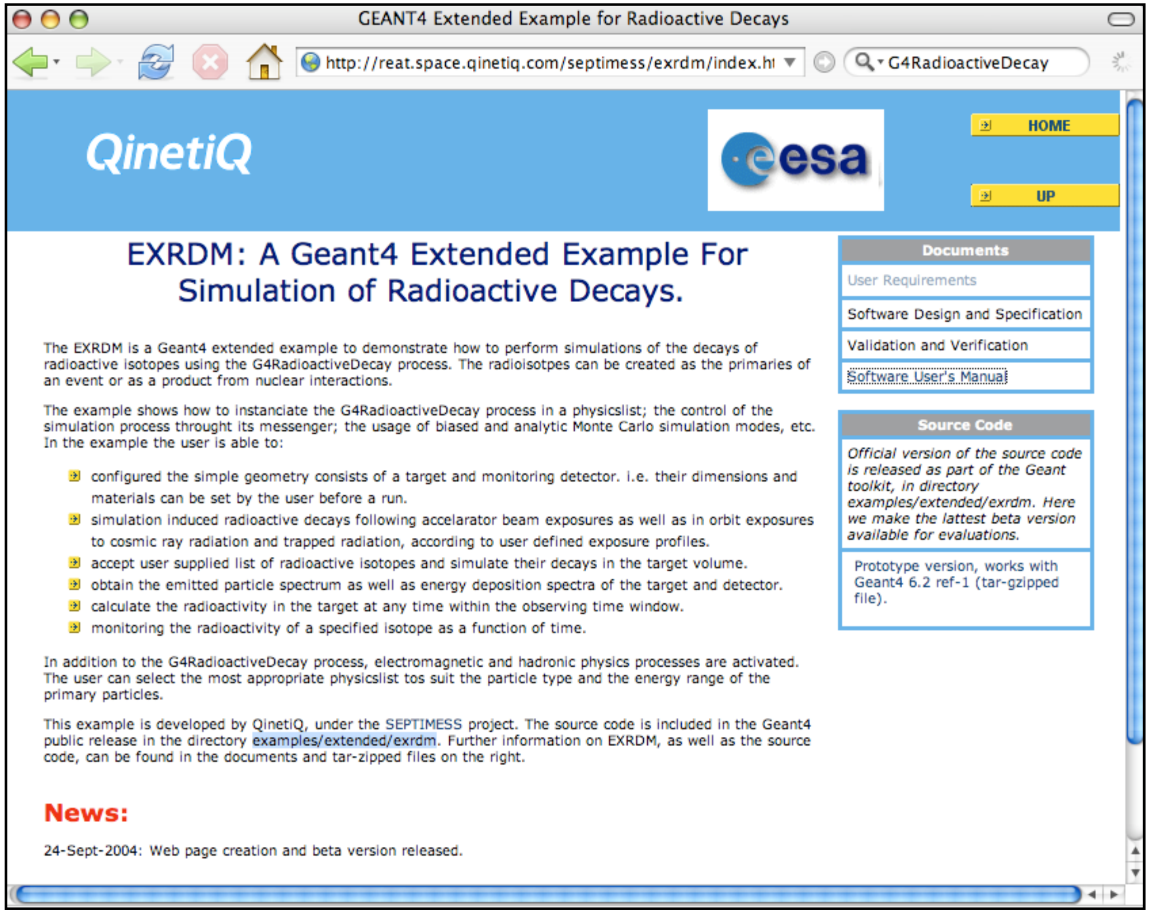
- Geant4 examples:
  - examples/extended/eventgenerator/exgps

Radioactive Decay Biasing

- G4RadioactiveDecay simulates decay of radioactive nuclei

- Implements the following biasing methods
  - Increase sampling rate of radioculides within observation times
    - User defined probability distribution function
  - Nuclear splitting
    - Parent nuclide is split into user defined number of nuclides
  - Branching ratio biasing
    - For a particular decay mode, sample branching ratios with equal probability

- Online documentation & examples
  - http://reat.space.qinetiq.com/septimess/exrdm/



- Geant4 examples:
  - examples/extended/radioactivedecay/exrdm

General Hadronic Leading Particle Biasing

- Built in utility for hadronic processes
  - Implemented in G4HadLeadBias class

- Keep only the most important part of the event, and representative tracks of given particle types
  - Keep track with highest energy
    - I.e, the leading particle
  - Of the remaining tracks, select one from each of the following types if they exist:
    - Baryon's, π^0's, mesons, leptons
  - Apply appropriate weight

Hadronic Cross Section Biasing

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

- Artificially enchance/reduce cross section of a process

- Useful for studying
  - Thin layer interactions
  - Thick layer shielding

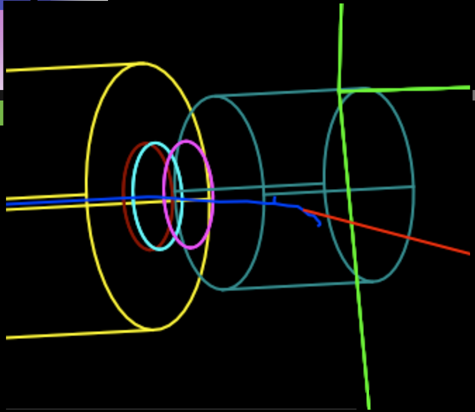
User Defined Biasing: G4WrapperProcess

- Implement user defined biasing through G4WrapperProcess
  - A process itself, I.e, inherits from G4VProcess
  - Wraps an existing process
    - By default, function calls are forwarded to existing process
  - Non-invasive way to manipulate the behaviour of a process
- To use:
  - Subclass G4WrapperProcess and override appropriate methods, e.g, PostStepDoIt
  - Register sublcass with process manager in place of existing process
  - Register existing process with G4WrapperProcess

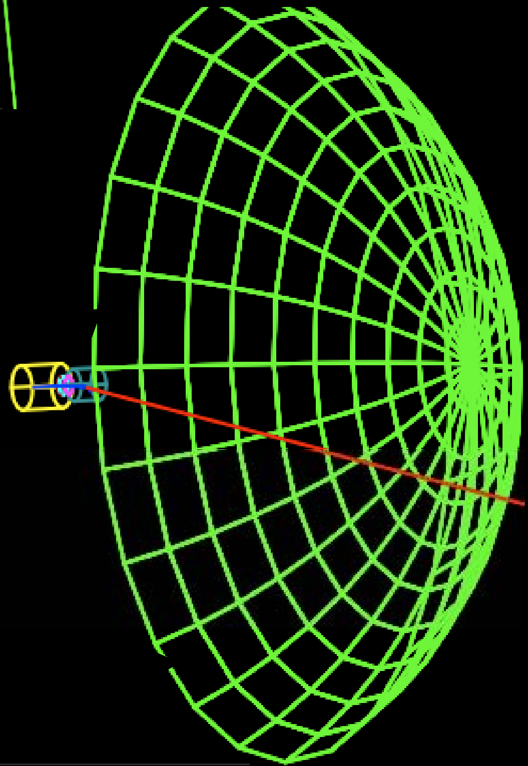
G4WrapperProcess Structure

```
class G4WrapperProcess : public G4VProcess {

protected:
  G4VProcess* pRegProcess;
...

inline void
G4WrapperProcess::RegisterProcess(G4VProcess* process)
{
  pRegProcess = process;
...

}

inline G4VParticleChange*
G4WrapperProcess::PostStepDoIt(const G4Track& track,
                               const G4Step&  stepData)
{
  return pRegProcess->PostStepDoIt(track, stepData);
}
```

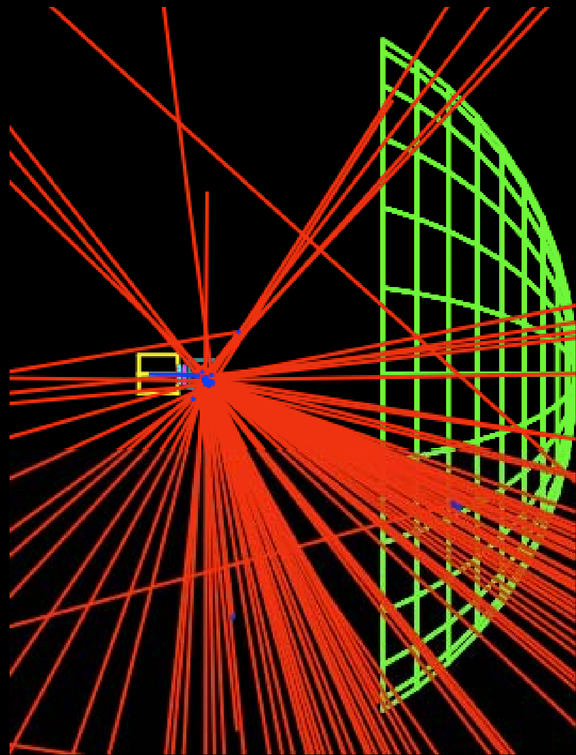
Uniform Bremsstrahlung Splitting



No splitting



Splitting factor = 100



Scoring
Geometry

Recent Developments

http://geant4.slac.stanford.edu/EBMS/

Geant 4

Geant4 Event Biasing and Scoring Mini-Workshop

Stanford Linear Accelerator Center
Menlo Park, California, U.S.A.
March 19 - 23, 2007

The Geant4 Event Biasing and Scoring Mini-Workshop addresses the emergent needs of event biasing and scoring options.
It targets to:

- Understand the users needs of event biasing and scoring options,
- Sort out the options currently available in Geant4,
- Understand the potential design iteration required in Geant4, and
- Discuss and plan the work for the coming releases for design, implementation, documentation and examples.

This mini-workshop is closed to the members of Geant4 Collaboration.

---

workshop local organizer 01/10/2007 19:58:48

Workshop Brief Summary & Work Plan

- Detailed summary and work plan at
  http://geant4.slac.stanford.edu/EBMS/material/Summary_EBminiworkshop.ppt
- Geometrical biasing
  - Updated to use parallel navigation developments
  - With release v9.0 should be able to do full geometrical biasing in parallel worlds
    - At the moment limited to neutrals
    - Biasing examples to be updated
  - Producing validation examples
- Scoring for biasing
  - Use in place of depreciated G4VScorer used in geometrical biasing
  - Development of new scorers

Scoring/Tallies – brief overview

Tallies	MCNPX	GEANT4	FLUKA	MARS	PHITS
Standard					
Flux					
Volume	Yes	Yes	Yes	Yes	Yes
Surface	Yes	Yes	Yes	Yes	Yes
Point/ring	Yes	No	No	Yes (neutrons)	No
Current	Yes	Yes	Yes	Yes	Yes
Charge	Yes	Yes	Yes	Yes	Yes
Kinetic energy	Yes	Yes	Yes	Yes	Yes
Particle density	Yes	Yes	No	No	No
Reaction rates	Yes	No	Star (inelastic)	Yes	Yes
Energy dep.	Yes	Yes	Yes	Yes	Yes
Rapidity	No	Yes	Yes	Yes.	No
DPA	Some	??	~~Yes~~	??	??
Momentum	No	Yes	Yes	Yes	No
Pulse-height	Yes	User input	Yes	No	Yes
Termination	Partial	??	??	??	Yes
Modifiers	10	3 >11	2	2	2
Special					
Mesh	rec, cyl, sph	Arbitrary	rec, cyl	rec, cyl, sph	rec,cyl
Coincidence	Yes	(USER)	Yes	Yes	Yes
Residuals	Yes	No	Yes	Yes	Yes
Event logs	Yes	Yes	Yes	Yes	Yes
Convergence Tests	10	Error >1	Error	Error	Error

Review of Monte Carlo All-Particle Transport Codes and ..
G. W. McKinney (*Los Alamos National Laboratory*) et al,
FNDA 2006, Cape Town, South Africa, April 3-6, 2006

9

Summary

- Number of popular event biasing techniques built into Geant4
- User defined biasing supported through G4WrapperProcess
- Ongoing developments aim to improve exiting Geant4 biasing, and provide new event biasing and scoring methods
- Geometrical based biasing is now extended to charged particles and magnetic fields (by default since release 9.0 for parallel geometries)
- Physics based biasing offers significant improvement in cases where full sampling is unnecessary
- Scoring is very flexible within Geant4 and offers extension
- Further requirements?
- Documentation at:

http://cern.ch/geant4/UserDocumentation/UsersGuides/ForApplicationDeveloper/html/ch03s07.html

- **Spare slides**

New Biasing Scheme (default since v9.0)

- Parallel geometry now must inherit from G4VUserParallelWorld
- Parallel and mass sampler classes now combined into one G4GeometrySampler class:

  e.g.: mgs(detector->GetWorldVolume(),"neutron");

  An additional set method defines whether or not the sampler is in a parallel geometry: mgs.SetParallel(false);
- When a parallel world is created, parallel navigation must be activated. An application with physics list that uses the AddTransportation method will automatically use G4CoupledTransportation which is picked up by the creation of the parallel world copy.
- All scoring can be implemented *only* through the Primitive Scorer classes.
- Users utilising the older scoring classes will need to migrate to using Primitive Scorers. The same functionality is provided apart from G4ScoreTable. Scorers are now attached to logical volumes (in place of geometry cells). This requires explicit copy numbers to be utilised for the same logical volume and the GeometryCell methods should be accessed through the physical volume and replica number method.
- A demonstration of the new biasing and scoring is available in examples/extended/biasing/B01 and B02.

- Physics biasing
  - Existing physics based biasing fragmented
  - Identify missing biasing methods & variations between methods in other Monte Carlo codes
    - Implicit capture
    - General cross section biasing
    - Interaction forcing
    - Path length biasing
    - Advanced bremsstrahlung splitting
    - Leading particle biasing
  - Look at developing dedicated framework to provide general physics biasing in analogy with geometrical biasing
    - Manipulating physics processes/lists