Quick Status of Generic Biasing

Marc Verderi
LLR/Ecole polytechnique
Ferrara G4 Collaboration Meeting
September 2016

Reminder About Existing Functionalities

- Design of generic biasing relies on:
  - two main abstract classes:
    - G4VBiasingOperation:
      - Biasing of physics processes
        - Change of process interaction law
        - Change of final state generation
      - Splitting/killing
    - G4VBiasingOperator:
      - Which takes decisions on what biasing operation to apply
      - At the beginning of the step, and at the post step
  - One concrete class : G4BiasingProcessInterface
    - Makes the connection between the biasing and the tracking
    - Gets instructions from the biasing operator about operations to apply

First released in 10.0 then consolidation in 10.1 and 10.2. In particular in 10.2 : use of track auxiliary information (forced collision scheme) & easy access to phys. process XS to operator.

- Concrete implementations:
  - Biasing operation to change a process cross-section
  - Forced collision scheme à la MCNP
  - Both functionalities validated with neutral particles
- Set of examples example/extended/biasing/GBXX
  - GB01 : change of XS
  - GB02 : force collision
  - GB03 : geom. importance based biasing
  - GB04 : bremsstralhung splitting

Status as of Today

- Not much happened this year
- Several items are planned
  - Not all will be delivered this year

  ⇒Take opportunity to discuss needs/priorities
- Prioritized list for now (not all for this year !):
  - Statistical tests / statistical test suite
  - Refactor existing generic biasing bremsstrahlung splitting example to source
  - Implicit capture
  - Biasing of charged particles
  - Allow use of parallel worlds
  - Use of occurrence biasing to allow continuous density change inside a same volume
  - DXTRAN-like biasing
  - Material/isotope biasing
  - Woodcock tracking

Status as of Today

- Not much happened this year
- Several items are planned
  - Not all will be delivered this year

  ⇒Take opportunity to discuss needs/priorities
- Prioritized list for now (not all for this year !):
  - Statistical tests / statistical test suite
  - Refactor existing generic biasing bremsstrahlung splitting example to source
  - Implicit capture
  - Biasing of charged particles
  - Allow use of parallel worlds
  - Use of occurrence biasing to allow continuous density change inside a same volume
  - DXTRAN-like biasing
  - Material/isotope biasing
  - Woodcock tracking

Statistical Test Suite

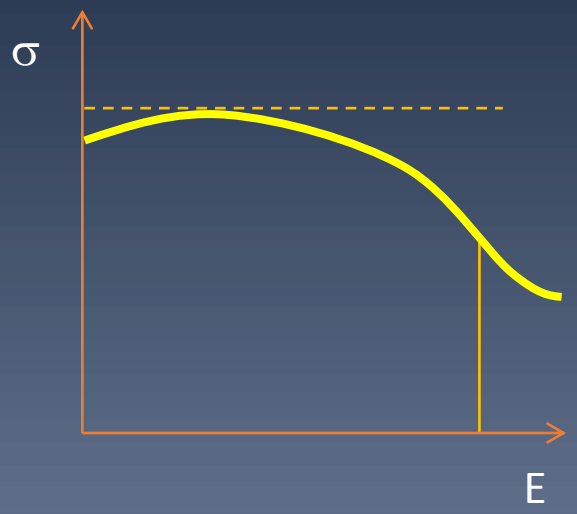
- This is the priority
  - as needed to validate any biasing developments
- Goal : perform a statistical validation of the biased simulation against the analog one
  - Hence, need to run heavy analog statistics
    - Clusters ? Grid ?
- Simple in principle:
  - Record “same” histograms and some observables
  - With quantities entered with proper weights
  - And make statistical tests
- In practice:
  - Might be a same test that is run with:
    - A light statistics mode, for testing purposes
    - A heavy statistics mode, for statistical validation
      - and a script/macro to perform the comparisons
  - Can/must be done with already existing options

Implicit Capture

- MCNP option in neutron transport
  - "Implicit capture," "survival biasing," and "absorption by weight reduction" stand for the same technique
- Keep neutrons alive wrt absorption process(es)
  - Makes a same neutron "exploring" more phase space
- From MCNP manual:
  - variance reduction technique applied after the collision nuclide has been selected. Let:
    - σ_{ti} = **total** microscopic cross section for nuclide i and
    - σ_{ai} = microscopic **absorption** cross section for nuclide I
  - particle weight changed as $W \rightarrow W * (1 - \sigma_{ai} / \sigma_{ti})$
- Technically, need to get the nuclide selected
  - Might not be straightforward
- MCNP also advertises "Implicit Absorption Along a Flight Path"
  - Variant of option, where absorption processes are suppressed.
    - This is simply equivalent to put to zero absorption cross-sections
    - We call "force free flight" : used today (in force collision scheme)
  - This second scheme should be possible today

Interaction Law Biasing for Charged Particles

- Change of interaction law by:
  - cross-section change
  - Force collision
  - Force non-collision (force free flight)
  - …
- Options that already exist for neutral particles
- Difficulty:
  - Weight calculation involves integral of XS over the step…
  - … and cross-section value changes over the step
    - Because of energy loss

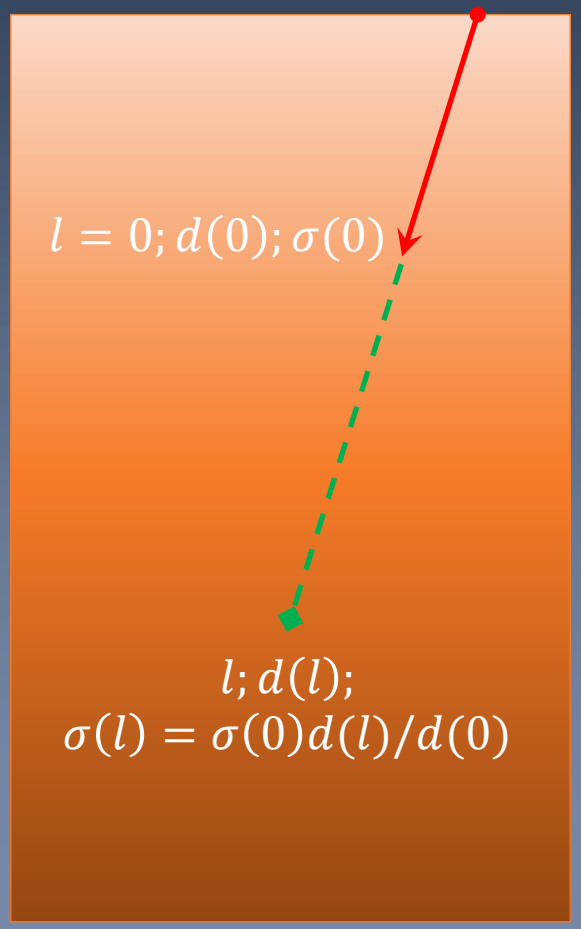


- Laurent Desorgher developed a method to compute the weight by MC
  - Validated on toy MC
  - Need to implement in G4
- Note:
  - XS variation over a step is taken into account by the “integral approach”
  - EM uses “integral approach” for processes
    - to take into account XS change for many years
  - Hadronics just starts to use this

  ⇒ Taking into account XS variation becomes hence a mandatory feature

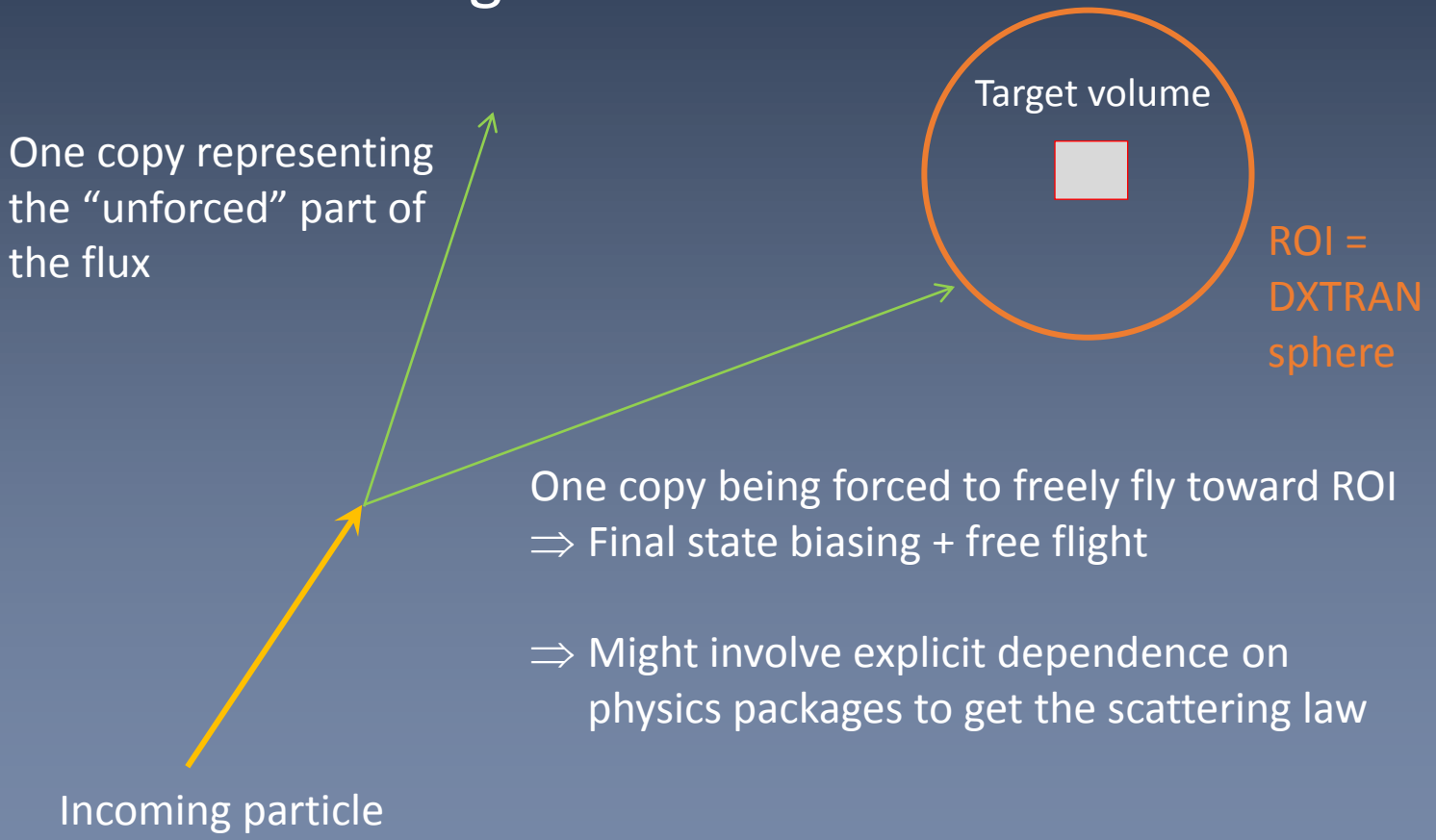
Use of Occurrence Biasing to Allow Continuous Density Change Inside a Same Volume

- Volumes hold a constant density material
- But occurrence biasing may be used to mimic a continuously varying density.
- Idea:
  - Sample:
    $p(l) = \sigma(l) \cdot \int_0^l \exp\big(-\sigma(s)\big)\, ds$
  - Instead of:
    $p(l) = \sigma \cdot \exp(-\sigma \cdot l)$
  - Using $\sigma(l) \propto \text{density}(l)$
- In this problem, all particles keep :
  - weight = 1



DXTRAN-like Biasing

› Option in MCNP to scatter particles toward a preferred solid angle



Dependencies on Other Physics Packages

- Dependencies between biasing and other physics packages must be introduced
  - Because of absence of generic interfaces for XS and differential XS
  - XS : interaction law biasing
    - G4VProcess provides the interaction length in a generic way
    - But getting the cross-section at the end of the step to apply the “integral approach” will likely involve explicit dependencies on physics packages
  - Differential XS : final state biasing:
    - Splitting/killing of primary/secondaries could be made using G4VProcess interface only
    - But any change of final state distribution will likely require dependencics on physics packages to access related laws
- Could consider:
  - Biasing depending on other physics packages
    - And all concrete biasing classes reside under biasing

    ** Or (exclusive) **
  - Physics packages depend on biasing
    - To access the biasing interfaces
    - Having the concrete biasing classes residing in physics packages
- Must be discussed.