# UPDATE ON GENERIC BIASING

*PARALLEL 5B*

Marc Verderi

LLR/Ecole polytechnique

Lund Collaboration Meeting

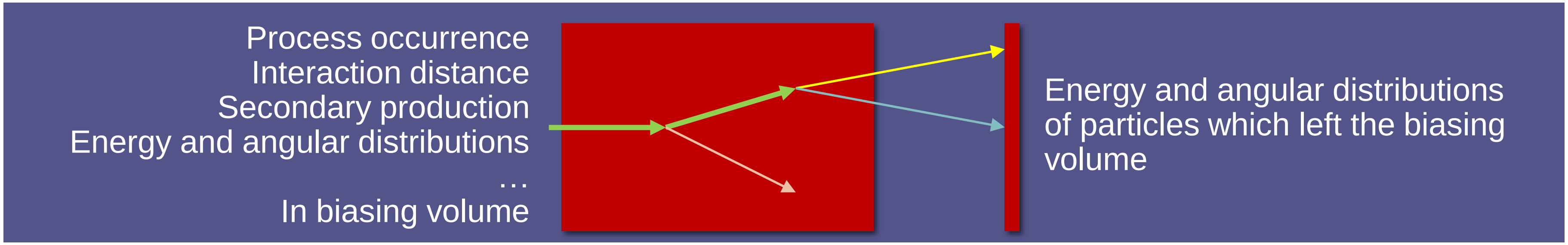
August 2018

# Statistical test suite

- Development ongoing under test49
  - Some code committed at the trunk
  - But not ready to go at this point !

- Aim at verifying statistical correctness of weight application
  - Verifications done with private tests up to now
  - But need high statistics from time to time for analog vs biased simulation comparison

- Observables:
  - Many variables are common to the various biasing options



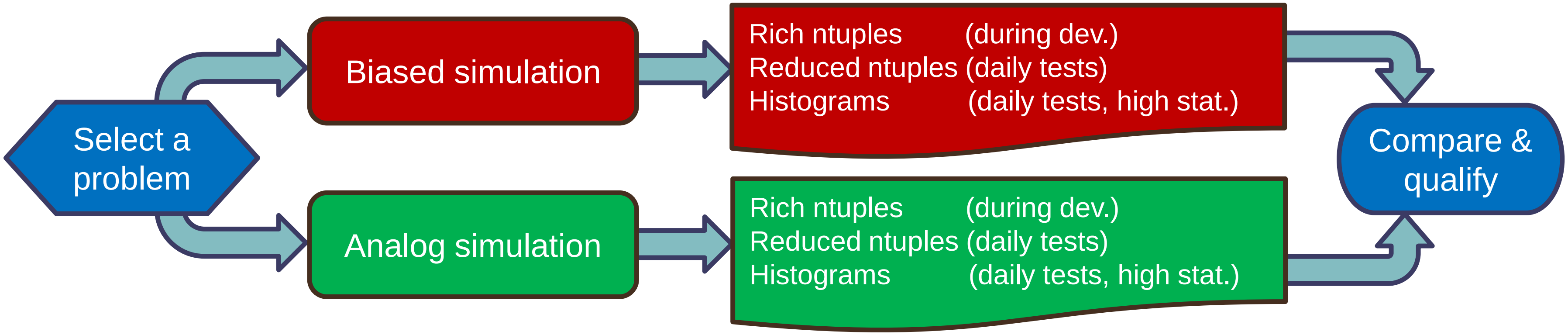
Process occurrence
Interaction distance
Secondary production
Energy and angular distributions
…
In biasing volume

Energy and angular distributions of particles which left the biasing volume

- So many options can use the same "testing framework"
  - Also desirable to share as much as possible : better robustness of test, and limited manpower

# Statistical test suite : requirements

- It is a Monte Carlo only test
- Run “rave events” simulation problems, and compare distributions from
  - High statistics analog samples
  - With moderate or small biased ones
  - Note we can also make the reverse:
    - Making rare some topologies with biasing
    - Not useful for physics, but is a valid way to verify differently the correctness of weight calculations
- Test should be able to run under configurations for:
  1. Development, with tests ran privately
  2. Daily testing:
     - Running with moderate biasing
     - Addressable with moderate statistics
  3. Heavy testing, for reference tags or releases, or dedicated runs
     - Running with strong biasing
     - Requiring high statistics
- Outputs depend on these configurations:
  1. Development: rich ntuples
  2. Daily testing : reduced nutples, histograms
  3. Heavy testing: histograms
- Comparison:
  - After the MC stage, the analog-biased comparison must occur
  - What outputs possible ?
- ***I do need technical help to setup this test49 in the testing environment !***
- NB: similar scheme needed for fast simulation.



Select a problem

Biased simulation

Rich ntuples (during dev.)
Reduced ntuples (daily tests)
Histograms (daily tests, high stat.)

Analog simulation

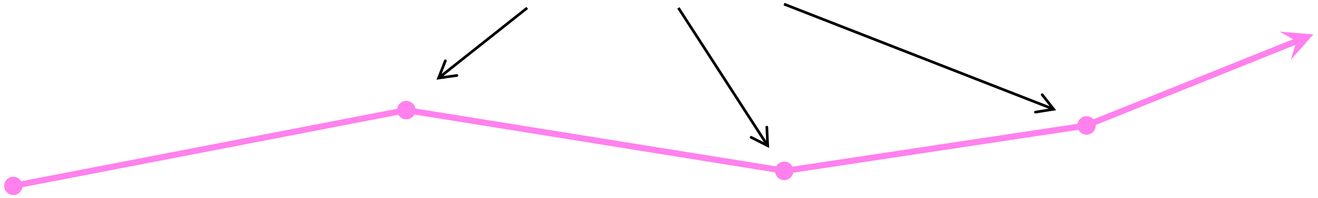
Rich ntuples (during dev.)
Reduced ntuples (daily tests)
Histograms (daily tests, high stat.)

Compare & qualify

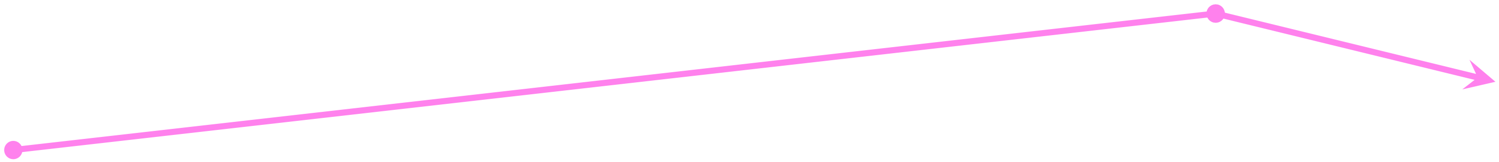
# Implicit Capture

- Aka “survival biasing,” and “absorption by weight reduction”
- Biasing technique used for neutrons (and gammas) to avoid “loosing” neutrons by absorption.
  - Keep neutrons alive wrt absorption to make it exploring more detector domains.

(A) - Suppress capture but update track weight to reflect this suppression



(B) - Or sample the interaction length without absorption ($\sigma_{abs.} = 0$) and by setting it to some value.

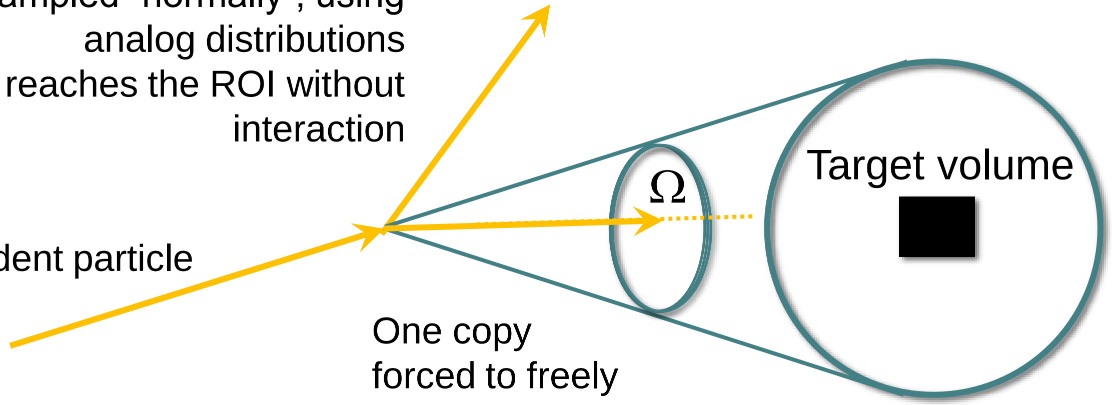


# Implicit Capture

- Technique implemented in MCNP and FLUKA under different flavors:
  - MCNP:
    1. After collision nucleus has been selected, ignore $\sigma_{abs.}$ and let neutron continues with $w' = w \cdot \left(1 - \frac{\sigma_{abs.}}{\sigma_{tot.}}\right)$
       - Rephrasing : conserve total XS, and scale the scattering XS to become the total XS – case (A) before
         - ☞ Saying this way allows us to find back the MCNP weight using the formalism presented in the general paper
    2. Cancel $\sigma_{abs.}$ when sampling interaction distance – case (B)
       - Hence changing the total XS (a case of “exponential transform”)
       - Useful in highly absorbing media, to avoid making many steps limited by the (ignored) absorption
         - ☞ A case already treated in GB01 example
    - These two schemes actually “bias” the simulation (ie: make it “wrong” as loosing some process)
      - As secondaries from absorption are not produced
      - ☞ Technique to be applied far enough from the region of interest, so that ignored secondaries have to effect on this area
  - FLUKA:
    3. Add the ability to set the cross-section to a desired value – case (B)
       - Something possible in G4 too (GB01).
- Main task is then to refactor these options in easy to use biasing class(es)
- Tentative implementation last year, but I got finally confused by the weight calculation (1.)
  - Reanalyzed this year, and came to above (understandable to me) rephrasing.
- In the work plan since few years, but should come this year !

# DXTRAN

- An other option also re-conducted in work plan several times…
- Option in MCNP to scatter particles toward a preferred solid angle
  - Has some similarities in its use-case with the Reverse MC as it targets small ROI
- DXTRAN = stands for deterministic transportation



One copy sampled “normally”, using analog distributions
Killed if reaches the ROI without interaction

Incident particle

One copy
forced to freely
fly up to ROI
Called “DXTRAN particle”

ROI =
DXTRAN
sphere

# DXTRAN

- Used only for elastic scattering in MCNP
  - If I'm correct…
- In Geant4, we intend it only for elastic (or quasi-elastic) scattering too.
  - In particular neutrons

- Main difficulty:
  - We must compute $w = \frac{p^a(\Omega)}{p^b(\Omega)}$
    - For $p^b(\Omega)$, the biased angular distribution, we can chose whatever convenient
    - For $p^a(\Omega)$, the analog angular distribution, we need the physics process calculation.
  - That without introducing dependencies
  - Agreed last year:
    - Having an abstract class that some process may implement to provide concrete distributions
    - That will be used by the biasing, messaging only the abstract interface
    - In slow progress (see next presentation)

- DXTRAN scheme will be difficult to deliver this year, a priori
  - But appear of interest to medical during discussion this week
  - So, adding some motivation…

# Geometry Importance for various particle types

- Daren reported a bug:
    - Problem 1941 - Cannot use importance sampling for more than one particle type
- Problem addressed to the geometry importance biasing scheme as primarily provided in Geant4
    - Alex investigates if this is a bug or a design limitation

- Alternative solution can a priori be provided with generic biasing:
    - Will be demonstrated in extented/biasing/GB03 how to use more than one particle type
        - Only neutron for now.
    - GB03 :
        - implements a technique inspired by geometry-based biasing
        - Adding some other options
            - like probability to apply splitting, to make non-integer effective splitting factors
    - No show-stoppers a priori.
        - As several generic biasing examples already handle several particle types.

# Other items on wish list:

- Extension of biasing scheme for AtRest
  - At present, generic biasing acts at PostStep level methods
    - Either for deciding and applying splitting/killing
    - Or by modifying the interaction probability of a physics process and/or its final state generation.
  - This is handled by the (pure virtual) methods of the base class for biasing operators, that returns the biasing operation to be applied:

```
ProposeNonPhysicsBiasingOperation( const G4Track*,
                                   const G4BiasingProcessInterface* );
ProposeOccurenceBiasingOperation( const G4Track*,
                                  const G4BiasingProcessInterface* );
ProposeFinalStateBiasingOperation( const G4Track*,
                                   const G4BiasingProcessInterface* );
```

  - Idea if to add the AtRest version of these methods: ie, AtRestProposeXXXX
    - Together with the machinery to handle these in the biasing operations,
    - And in the G4BiasingProcessInterface class.

- Cross-volume biasing:
  - For now, biasing if applied to one volume, without daughters
  - By there are use-cases to –say- force the interaction inside a mother volume
    - In what case, the internal volume structure must be taken into account