Biasing and Scoring parallel session

- Following-up the mini-workshop @ SLAC in March
  - Enhancing biasing and scoring options as rich as MCNP, if not more
  - Unifying usages of geometry biasing and physics biasing, and simplifying them
  - Offering ready-to-use scorers
- Changes
  - Parallel geometry now must inherit from G4VUserParallelWorld
  - Parallel and mass sampler classes now combined into one G4GeometrySampler
  - All scoring can be implemented only through the Primitive Scorer
- New developments
  - Physics biasing framework
    - Great potential extendibility also for processes-per-region
  - Command-base scorer
- Issues
  - Robustness and performance of the new parallel navigation mechanism
  - Use-relation between Processes and Digits_hits for scoring

Developments since mini-workshop (1)

- New module for process-based biasing
- Migrated classes from processes/scoring and transportation modules
- Removed classes from old biasing scheme
- These are now replaced by either G4CoupledTransportation (parallel navigation) or in processes/biasing module
- The scoring is implemented with the new scoring framework (in digits_hits and processes/scoring modules).
- The following classes were removed: G4ParallelNavigator, G4ParallelStepper, G4ParallelWorld, G4VParallelStepper, G4VPGeoDriver, G4VScorer, G4ImportanceSplitExaminer, G4VImportanceSplitExaminer, G4VWeightWindowExaminer, G4WeightWindowExaminer.

Developments since mini-workshop (2)

- Examples B01 and B02 migrated to new scheme
  - And found to reproduce the old system
- B03 removed (python/obsolete)
- TIARA still needs updating / re-writing / migrating / maintaining
- Benchmarking vs. data with Figure of Merits still need developing for biasing
- Biasing documentation updated, but not extended

Scoring updates regarding with event biasing

Tsukasa Aso

Toyama National College of Mari. Tech.

Perspective

- Several primitive scorers had been introduced in geant4.9.0 for the purpose of migrating the biasing scorers to the primitive scorers.

- The other scorers are to be introduced in order to provide similar functionality as MCNPX.
  - Residual Nuclei
  - Reaction rate by process counter
  - DPA (Displacement per atom)

- 1D/2D counter plot should be provided for visualizing the result of a scorer.
  => Cooperative work with M.Asai, A.Kimura, T.Aso.
  This will be presented by Makoto.

New scorers - prototype -

- Reaction rate: Number of Hadronic inelastic interactions
  - Counts number of steps by a particular process
    - G4PSNofCollision and a process filter will be applied.
    - (Created) SD-filter by process name/type and particle name. G4SDProcessFilter/G4SDProcessTypeFilter.
- Termination: Number of tracks killed by a particular process
  - This may be useful in the other MC for evaluating the energy threshold for tracking, while it may be less important in G4 because G4 transports a track down to zero energy.
    - G4PSTermination and a process filter will be applied.
    - (Created) G4PSTermination scores only at a last step.
    - (Created) SD-filter by process name/type and particle.

- Residual: Number of recoil nuclei/produced nuclei
  - Counts number of nuclei.
    - G4NofSecondary will be applied with specifying the secondary particle
    - (Modified) G4NofSecondary for a particular secondary particle.
    - (Created) SD-filter by creator process name/type and parent particle. G4SDCreatorProcessFilter
- DPA: Displacement per atom (Pending)
    - Under investigation

Physics Biasing Framework

- Aim to provide flexibility through common physics biasing framework
  - Make life a little easier for the users

- Two levels to biasing
  - Processing (G4SteppingManager) level :
    - Manippulating physics & processing lists, taking into account when a physics list/process is active (triggered)
      - Independent of process type
        - Although whatever process grouping constraints currently imposed must still be applied
  - Process level:
    - Actual biasing code
    - Executed in GPIL/DoIt methods
    - Avoid “do it all” interface classes to simplify biasing while allowing access to underlying processes when need to do more complex biasing
      - More complex biasing working on process type level is limited as to what interfaces the actual process provides

Prototype

- Prototype code in CVS under geant4/source/processes/biasing/test/physics_biasing/
  - GPR stands for generalised processing since biasing processes don't need to inherit from anything
- Relevant directories
  - gpr_base : basic general use building blocks
  - gpr_core : more complex structures forming gpr processing
  - gpr_configuration : user interface stuff
  - gpr_geant4_modifications : modifications to geant4
  - gpr_examples : A01 example demonstrating biasing
- It's development code.
  - Lots of debugging print
  - Subset of desired features implemented
  - Probably buggy
  - Just an example of how things could be done

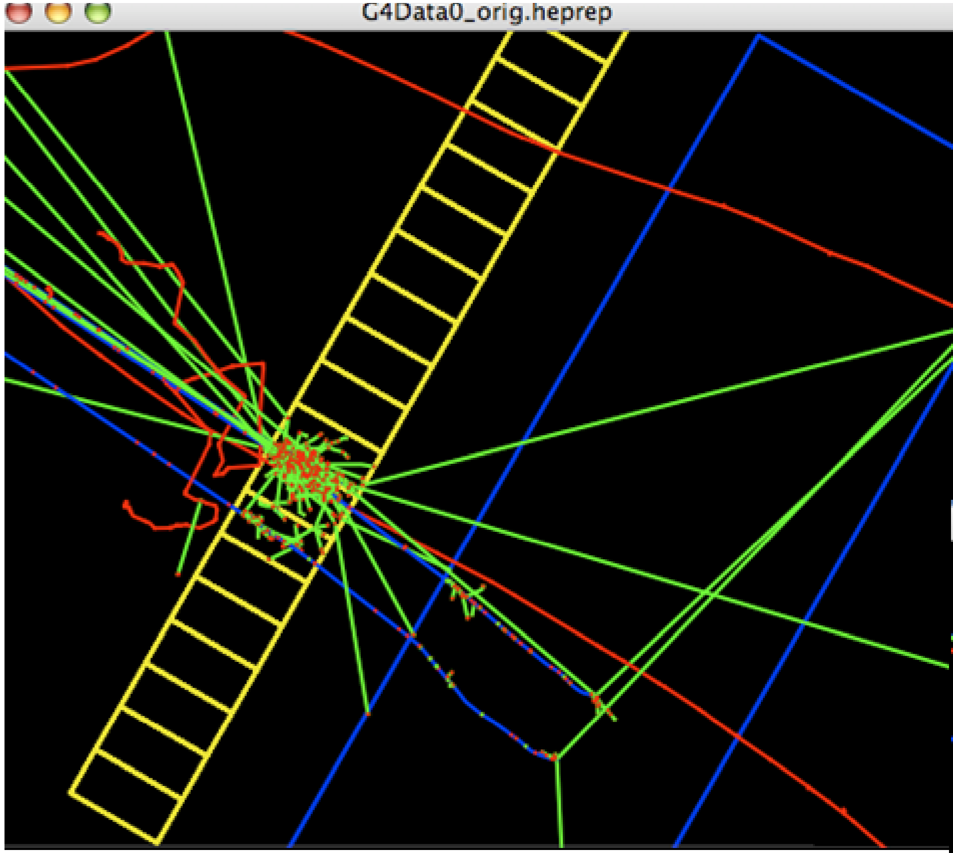
Leading Particle Biasing

- processes/biasing/test/physics_biasing/gpr_examples/A01/include/A01Biasing_Leading_Particle_Biasing.hh

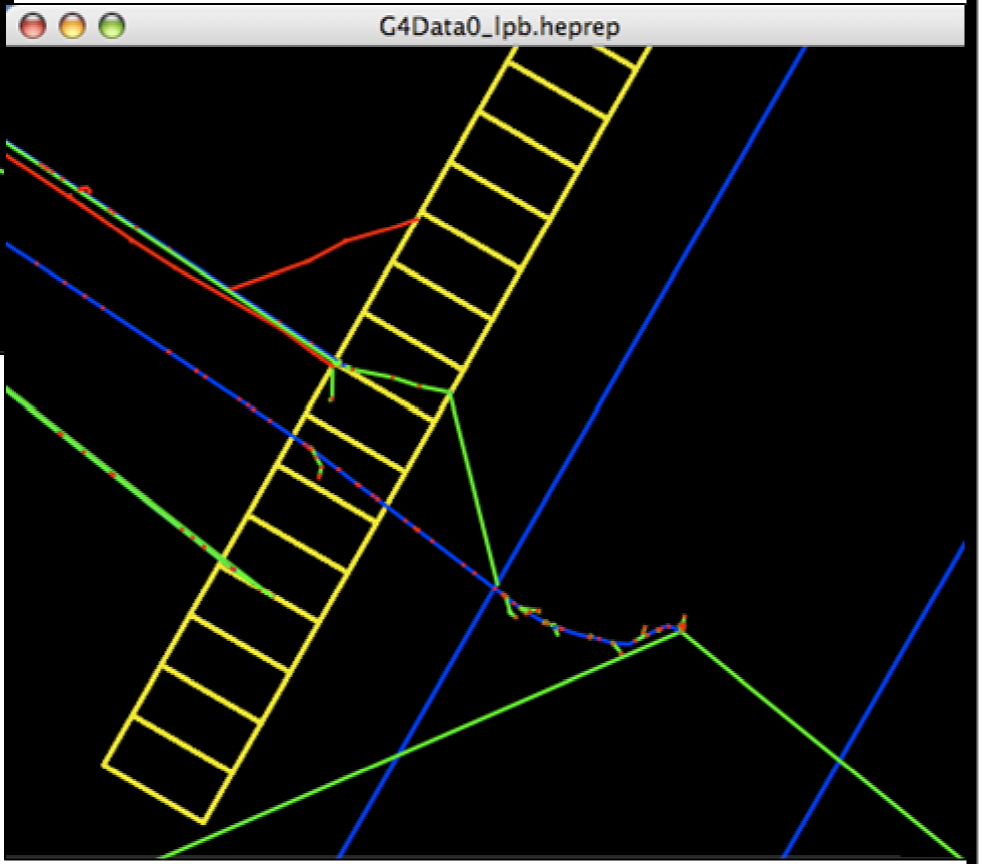
- Implements simple (EGS style) electromagnetic leading particle biasing for e-, e+ and gammas in the electromagnetic calorimeter

- Implements equivalent of existing hadronic leading particle biasing
  - Bias only G4HadronicProcesses
  - Only apply to incoming tracks with energy < 5 GeV

Random Primaries



Regular processing



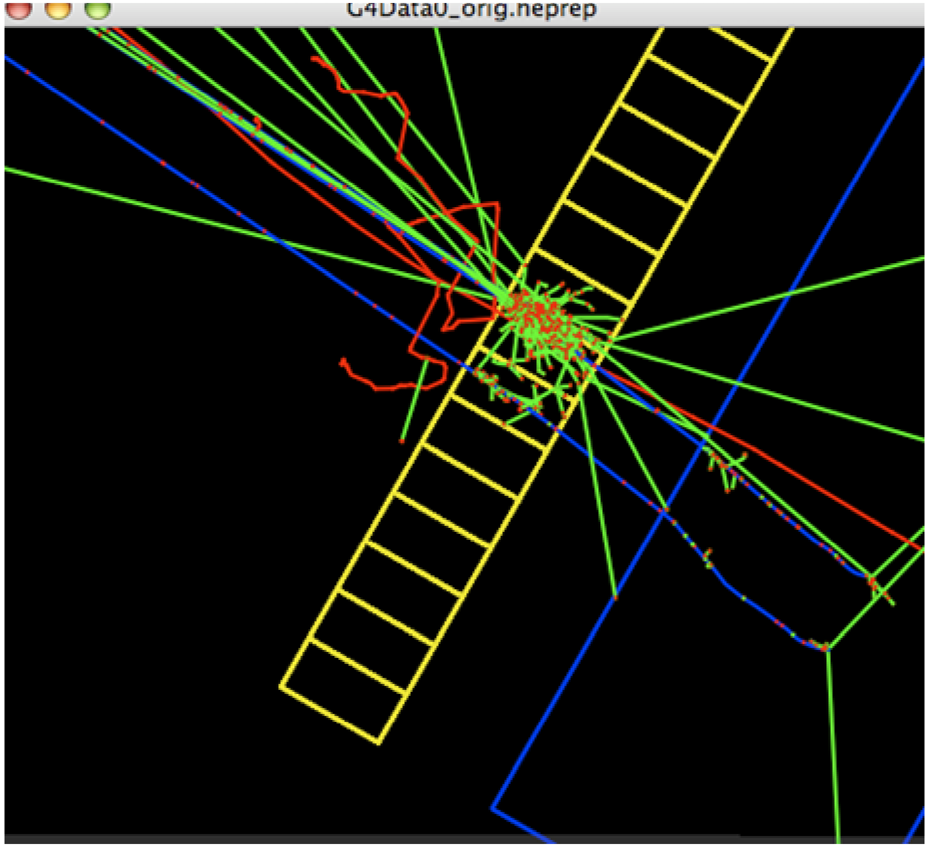
Leading particle biasing

Jane Tinslay, SLAC

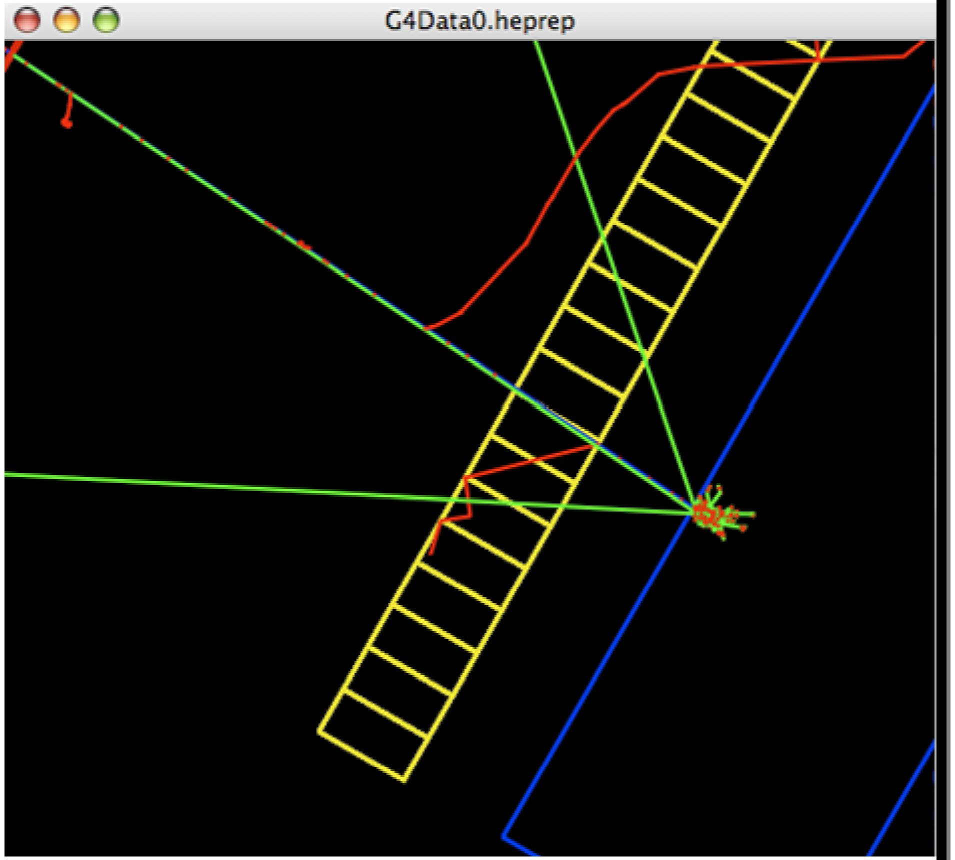
New calorimeter Physics list

- processes/biasing/test/physics_biasing/gpr_examples/A01/include/A01Biasing_New_Calorimeter_PhysicsLists.hh
- Create new physics lists for photon, e+, e- triggered only in calorimeter
- No photon, e-, e+ interactions in calorimeter

Random Primaries



Regular processing



No e+, e-, gamma
Interactions in calorimeter

Jane Tinslay, SLAC