Stanford Linear Accelerator Center

Updates in Kernel

Makoto Asai (SLAC/SCCS)

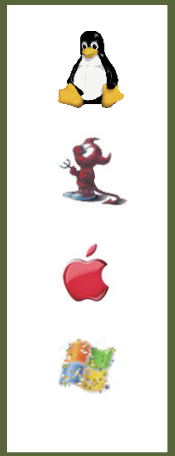
5th Geant4 Space Users' Workshop
February 15th, 2008

Geant4

- Contents
  - Recent releases and supporting platforms / OS
  - New features in kernel
    - Parallel geometries
    - Command-based scoring
    - Other new features in kernel
- Releases since the previous space users workshop
  - Dec 15th, '06 - Geant4 version 8.2 release
    - Feb 23rd, '07 - Geant4 8.2-patch01 release
  - May 4th, '07 - Geant4 version 8.3 release
    - Aug 17th, '07 - Geant4 8.3-patch01 release
    - Feb 5th, '08 - Geant4 8.3-patch02 release
  - June 29th, '07 - Geant4 version 9.0 release
    - Aug 28th, '07 - Geant4 9.0-patch01 release
    - Feb 5th, '08 - Geant4 9.0-patch02 release
  - Dec 14th, '07 - Geant4 version 9.1 release ← Current version
    - Feb 5th, '08 - Geant4 9.1-patch01 release

Platforms & compilers currently supported

- Linux systems
  - Scientific Linux SLC4, g++ gcc 3.4.6
    - G4SYSTEM: Linux-g++
- UNIX systems
  - SUN-SunOS v.5.8, CC v.5.5
    - G4SYSTEM: SUN-CC
- MacOS systems
  - MacOS 10.4 / 10.5, g++ gcc 4.0
    - G4SYSTEM: Darwin-g++
- Windows systems
  - Win/XP & Cygwin32, MSVC++ 8.0
    - G4SYSTEM: WIN32-VC



Parallel geometries

Parallel navigation

- Occasionally, it is not straightforward to define sensitivity, importance or envelope (for shower parameterization) to be assigned to volumes in the mass geometry.
  - Generally a geometry built machinery by CAD, GDML, DICOM, etc. has this difficulty.
  - Also scoring particle flux on an artificial plane, independent to any volume in complex detector set-up, has this difficulty, too.
- New parallel navigation functionality allows the user to define more than one worlds simultaneously.
  - New G4CoupledTransportation process sees all worlds simultaneously.
    - Once a parallel world is defined, G4CoupledTransportation is automatically used instead of conventional G4Transportation.
  - A step is limited not only by the boundary of the mass geometry but also by the boundaries of parallel geometries.
  - Materials, production thresholds and EM field are used only from the mass geometry.
  - In a parallel world, the user can define volumes in arbitrary manner with sensitivity, regions with shower parameterization, and/or importance field for biasing.
    - Volumes in different worlds may overlap.

Parallel world

- G4VUserParallelWorld is the new abstract base class by which the user implements a parallel world.
  - The user should implement the geometry of this parallel world in the Construct() virtual method.
  - The world physical volume of the parallel world is provided as a clone of the world volume of the mass geometry.
    - GetWorld() method (defined and implemented in the base class) returns the pointer to the cloned world physical volume.
  - Since it's an artificial parallel world, a logical volume needs not to have a valid pointer to G4Material.

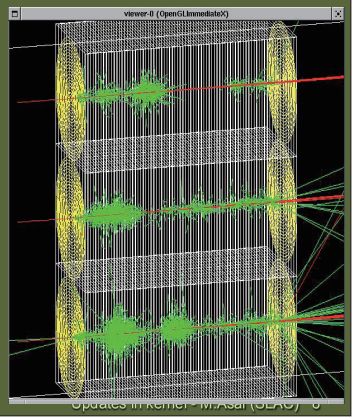
```
class G4VUserParallelWorld
{
  public:
    G4VUserParallelWorld(G4String worldName);
    virtual ~G4VUserParallelWorld();
  public:
    virtual void Construct() = 0;
  protected:
    G4String fWorldName;
  protected:
    G4VPhysicalVolume* GetWorld();
};
```


Parallel navigation

- All UserParallelWorlds must be registered to UserDetectorConstruction.
  - G4VUserDetectorConstruction::RegisterParallelWorld()
  - Each parallel world must have its unique name. It must be given as an argument of G4VUserParallelWorld base class constructor. It is then used as the name of the world physical volume.
  - Each parallel world has its dedicated G4Navigator object, that is taking care of the touchable of this particular world.
- Though all worlds will be comprehensively taken care by G4CoupledTransportation process for their navigations, each parallel world must have its own process to achieve its purpose.
  - For example, in case the user defines a sensitive detector in a parallel world, a process dedicated to this world is responsible to invoke this detector.
    - G4SteppingManager sees only the detectors in the mass geometry.
    - Use G4ParallelWorldScoringProcess. This process alternates G4Step with its own touchable.

New exampleN07

- Fully revised ExampleN07 is a good starting point for scoring in parallel world.
- It has rectangular layers of absorber and sensitive material in mass geometry (drawn in white) and cylindrical layers to measure the lateral energy distribution (drawn in yellow).
- ExN07DetectorConstruction class defines the geometry and scorers in the mass geometry, while ExN07ParallelWorld class defines them in the parallel world.
- ExN07Run class accumulates scores of both mass and parallel worlds, and ExN07RunAction class makes the report at the end of a run.
- ExN07SteppingVerbose measures the timing spent for each region.
  - Because of this, you cannot get the ordinary tracking verbose output for this example.

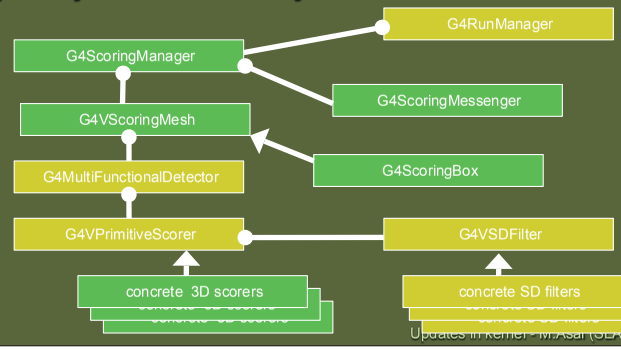


Command-based scoring

Updates in kernel - M.Asai (SLAC)

Command-based scoring

- Thanks to the newly developed parallel navigation, now we can define an arbitrary scoring mesh geometry which is independent to the volumes in the mass geometry.
- Also, G4MultiFunctionalDetector and primitive scorer classes now offer the built-in scoring of most-common quantities.
- Then, why not offering command-based scoring functionality to free the user from implementing C++ source code for scoring.

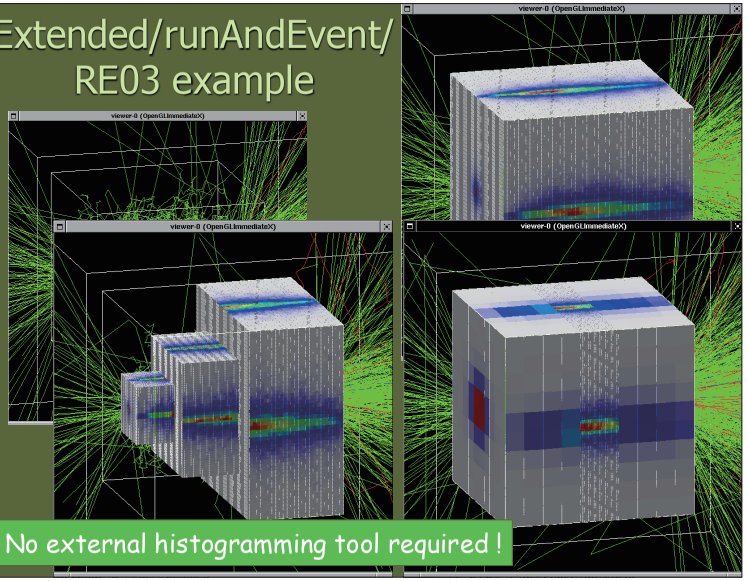


UI commands for scoring

- Define a scoring mesh
  - /score/create/boxMesh <mesh_name>
  - /score/open, /score/close
- Define mesh parameters
  - /score/mesh/boxSize <dx> <dy> <dz> <unit>
  - /score/mesh/nBin <nx> <ny> <nz>
  - /score/mesh/translate, /score/mesh/rotate
- Define primitive scorers
  - /score/quantity/energyDeposit <scorer_name>
  - /score/quantity/cellFlux <scorer_name>
  - /score/quantity/nOfStep <scorer_name>
  - currently **16 scorers** are available
- Define filters
  - /score/filter/particle <filter_name> <particle_list>
  - /score/filter/kineticEnergy <filter_name> <Emin> <Emax> <unit>
  - currently **5 filters** are available
- Output
  - /score/drawProjection, /score/drawColumn <mesh_name> <scorer_name>
  - /score/dump, /score/list, /score/dumpQuantityToFile

Note:
One mesh may take arbitrary number of primitive scorers.

Extended/runAndEvent/ RE03 example



No external histogramming tool required !

Command-based scoring

- It is still a beta release. There occasionally appear warning messages for stuck track on the volume boundaries.
  - We believe the scoring result is correct regardless of these warning messages.
  - We are working to fix this problem.
- Since it's a beta release, this functionality is not available by default. To use this, please add a couple of lines in your main().
  - #include "G4ScoringManager.hh"
  - G4ScoringManager::GetScoringManager();
    immediately after the instantiation of G4RunManager.
  - Refer to /examples/extended/runAndEvent/RE03
- Avoid too granular and/or too many meshes. It's the user's responsibility to consider the mesh size in the memory. Also, too many boundaries may cause a performance concern.
- The default file format is CSV (comma-separated values). The user may override the output method to any file format.
- Only the linear coloring map is currently provided. The user may override the coloring scheme.
  - Color legend is missing. It will be available in the next release.
- Currently, only the rectangular scoring mesh is supported. Cylindrical mesh is under development.

Other new features in kernel

Run, Event, Tracking and Track

- 8.2 G4Run object is kept until the beginning of the next run, so that the user can have an access to the valid G4Run pointer during the Idle state after the run. Events can be kept un-deleted, stored in G4Run, and persist until the deletion of the G4Run objects.
- 8.2 G4Step has a vector of secondary particles created in that particular step.
- 8.2 Added methods for setting polarization and modifying weight for G4ParticleChangeForGamma and G4ParticleChangeForLoss.
- 8.2 New methods IsFirst/LastStepInVolume() in G4Step, with accessors and modifiers in G4Step and G4VParticleChange.
- 9.0 Removed obsolete scorers dedicated to geometry biasing.
- 9.0 Added non-ionizing energy deposit in G4Step and G4VParticleChange.
- 9.1 Added class member particle_momentum and method SetParticleMomentum(). Also added UI commands /gun/momentum and /gun/momentumAmp

Particle

- 8.2 Updated mass/width/encoding of particles to match with PDG2006:
- 8.2 Introduced PDG encoding for nuclei.
- 8.2 Modified GetMass() and GetCharge() of G4PrimaryParticle to return PDG mass and charge if users do not specify them.
- 9.0 Added magnetic moment in G4ParticleDefinition, G4DynamicParticle and G4IsotopeProperty. Updated magnetic moment for leptons. Added G4IsotopeMagneticMomentTable and data file.
- 9.0 Allow for any Z,A combination (A>=Z) for G4IonTable::GetIon().
- 9.1 Added G4HyperNucleiProperties class so that hyper-nuclei can be created for G4Ions. G4ParticleTable can return a pointer to hyper-nucleus by using GetIon() and FindIon() methods.
- 9.1 Modified G4PDGCodeChecker to support PDG code for hyper-nuclei.
- 9.1 Added new classes G4PionRadiativeDecayChannel and G4MuonRadiativeDecayChannelWithSpin.

Material

- 8.2 Added methods GetAtomicMass(), AddIsotopeMass(), returning atom mass and nuclear mass in Geant4 units.
- 8.2 Added GetTotalBindingEnergy() method to G4AtomicShells.
- 8.2 Added Get/SetNaturalAbundancesFlag().
- 9.0 Added implementation file for G4SurfaceProperty.
- 9.1 G4NistManager: extended interfaces to build new materials; added G4State, temperature, pressure (for gases) with default values. Simplified logic of Print() methods.