New Scoring Functionalities

Makoto Asai (SLAC)

Recent developments in scoring functionalities

- Command-based real-world scorer
  - User may assign a scorer to a logical volume defined in the mass world.
- Command-based probe scorer
  - User may locate scoring "probes" at arbitrary locations. A "probe" is a virtual cube, to which any Geant4 primitive scorers could be assigned. Probes are placed in a dedicated parallel world so that they may overlap to the volumes defined in the mass geometry.
  - In addition, the user may optionally set a material to the probe. Once a material is set to the probe, it overwrites the material(s) defined in the mass geometry when a track enters the probe cube.
- 1-D histogram directly filled by a primitive scorer
  - Track-by-track or step-by-step filling allows command-based histogram such as energy spectrum.
  - Histogram must be predefined by G4Analysis.

- N.B. Due to the design reasons, real-world scorer and probe scorer are treated as a kind of "scoring mesh".
  - Treated as an alternative to box mesh or cylindrical mesh.
  - Thus, their names are occasionally referred as "mesh name".

Command-based real-world scorer

- With a UI command, user can define a scorer to a logical volume in the mass world.

- Once a real-world scorer is defined, user can associate arbitrary number of primitive scorers and filters like the conventional scoring mesh.

  - All physical volumes have the same primitive scorers but score individually.

  - Logical volume name is used as the scorer (mesh) name.

/score/create/realWorldLogVol *rad_monitor*

/score/quantity/energyDeposit eDep MeV

/score/quantity/doseDeposit dose mGy

/score/close
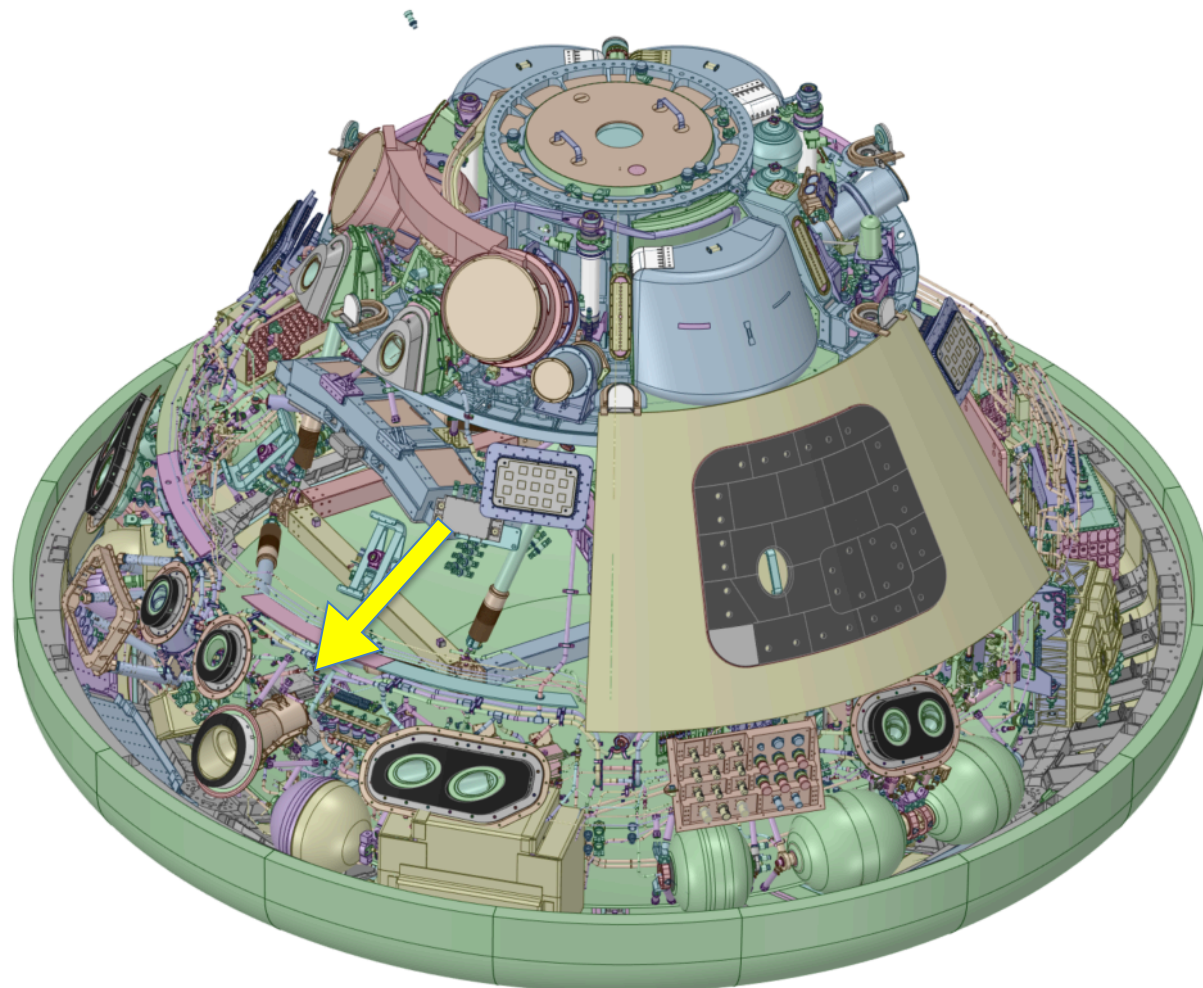
Command-based probe scorer

- User may locate scoring "probes" at arbitrary locations. A "probe" is a virtual cube, to which any Geant4 primitive scorers could be assigned.

- Given these probes are located in an artificial "parallel world", probes may overlap to the volumes defined in the mass geometry, as long as probes themselves are not overlapping to each other or protruding from the world volume.

- In addition, the user may optionally set a material to the probe. Once a material is set to the probe, it overwrites the material(s) defined in the mass geometry when a track enters the probe cube.

  - Because of this overwriting, physics quantities that depend on material or density, e.g. energy deposition or dose, would be measured accordingly to the specified material.

  - The user does not need to care the overlaps against real-world volumes.

  - Please note that this overwriting material obviously affects to the simulation results, so the size and number of probes should be reasonably small to avoid significant side effects.

- Once a probe is defined, user can associate arbitrary number of primitive scorers and filters like the conventional scoring mesh.

  - All probes have the same scorers but score individually.

Command-based probe scorer

/score/create/probe Probes 5. cm

/score/probe/material G4_WATER

/score/probe/locate 0. 0. 0. cm

/score/probe/locate 25. 0. 0. cm

/score/probe/locate 0. 25. 0. cm

/score/probe/locate 0. 0. 25. cm

/score/quantity/energyDeposit eDep MeV

/score/quantity/doseDeposit dose mGy

/score/quantity/volumeFlux volFlx

/score/quantity/volumeFlux protonFlux

/score/filter/particle protonFilter proton

/score/close
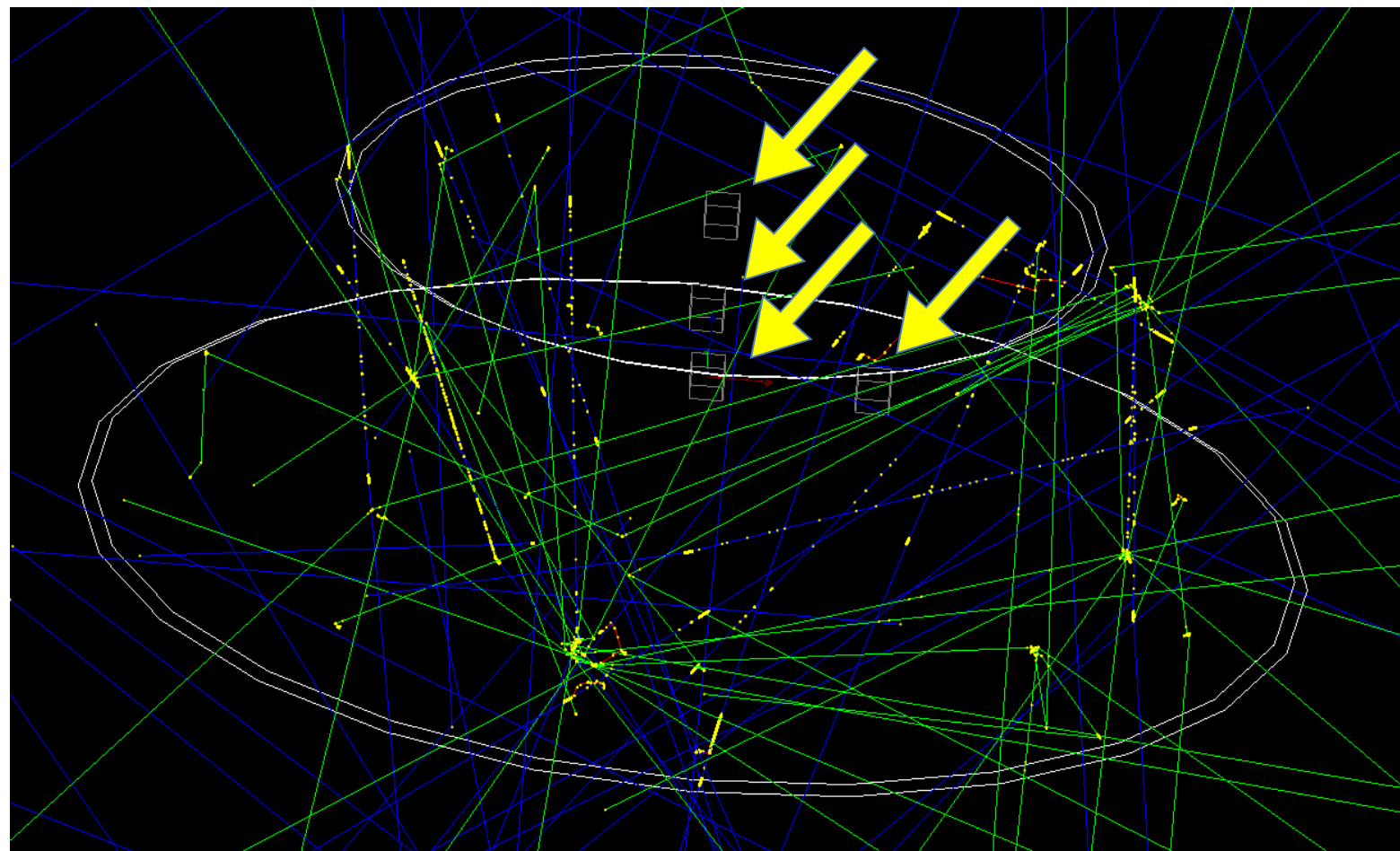
Note: To visualize the probes defined in a parallel world, the following command is required.
/vis/drawVolume worlds

1-D histogram directly filled by a primitive scorer

- Through a newly introduced interface class (G4TScoreHistFiller) a primitive scorer can directly fill a 1-D histogram defined by G4Analysis.

    – Track-by-track or step-by-step filling allows command-based histogram such as energy spectrum.

- G4TScoreHistFiller template class must be instantiated in the user's code with his/her choice of analysis data format.

```
#ifndef MyAnalysis_h
#define MyAnalysis_h 1
//#include "g4root.hh"
#include "g4csv.hh"
//#include "g4xml.hh"
#endif
```

```
#include "MyAnalysis.hh"
#include "G4TScoreHistFiller.hh"

auto histFiller =
        new G4TScoreHistFiller<G4AnalysisManager>;
```

- Primitive scorer must be defined in advance to setting a histogram.

- Histogram must be defined through /analysis/h1/create command in advance to setting it to a primitive scorer.

- This functionality is available only for primitive scorers defined in real-world scorer or probe scorer.

    – Not available for box or cylindrical mesh scorer due to memory consumption concern.

- Currently, only a few primitive scorers offer this functionality but will be extended to most of them before 10.7 release.

1-D histogram directly filled by a primitive scorer

/score/create/probe Probes 5. cm

/score/probe/locate 0. 0. 0. cm

/score/quantity/volumeFlux volFlux

/score/quantity/volumeFlux protonFlux

/score/filter/particle protonFilter proton

/score/close

/analysis/h1/create volFlux Probes_volFlux

                 100 0.01 2000. MeV ! log

/score/fill1D 1 Probes volFlux

/analysis/h1/create protonFlux Probes_protonFlux

                 100 0.01 2000. MeV ! log

/score/fill1D 2 Probes protonFlux

N.B. If probe is placed more than once, *fill1D* command should be called to each *copyNo.*

/score/fill1D 1 Probes volFlux 0

To sum up
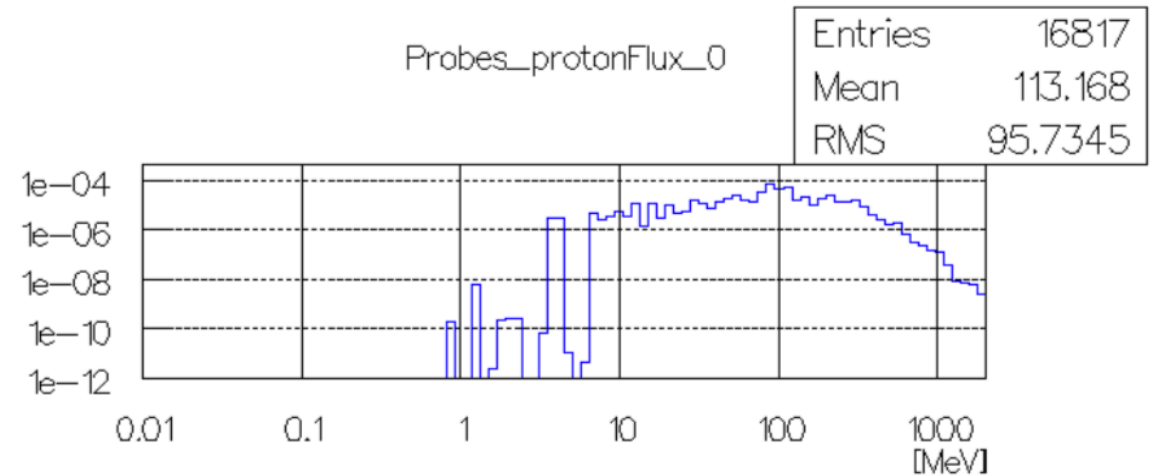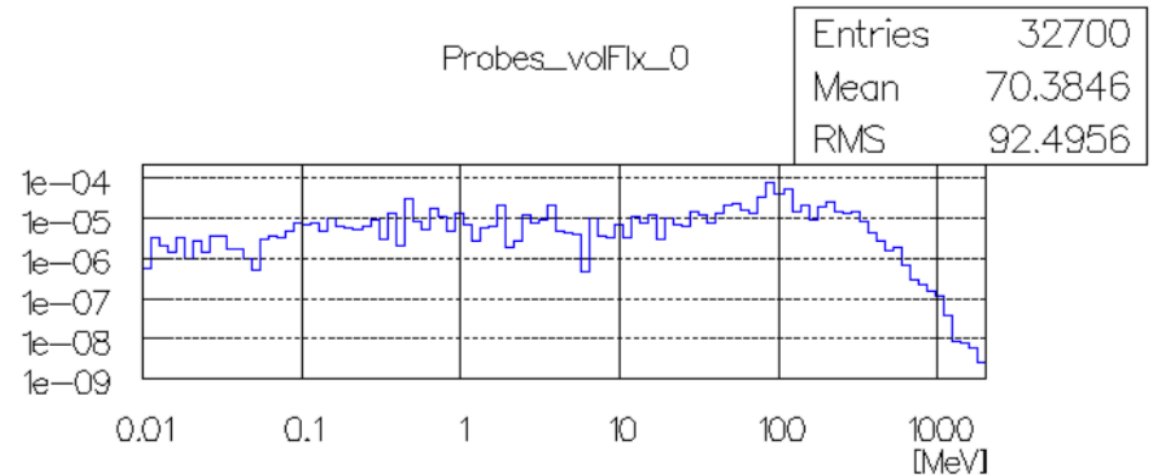
- Some new functionalities are added to scoring.
  - Command-based real-world scorer
  - Command-based probe scorer
  - 1-D histogram directly filled by a primitive scorer
- They all are controlled by UI commands. They will simplify some use-cases, for example
  - Real-world scorer to score dose in a volume read from GDML,
  - Probe scorer to score particle flux at the location where no easy space for real-world volume, and
  - Energy spectrum histogram of a particle type at a certain position
- These functionalities are demonstrated in a new advanced example ***Gorad***.
  - Updates of User's guide underway.
  - An additional extended example will be added in 2021.
- They are already in the Git repository and will be released as a part of version 10.7.