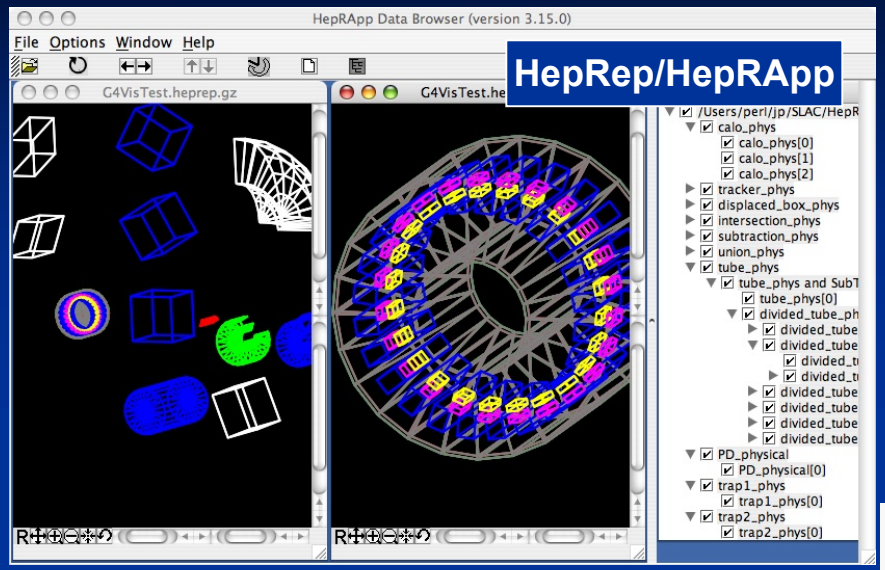
Geant4 Visualization Commands

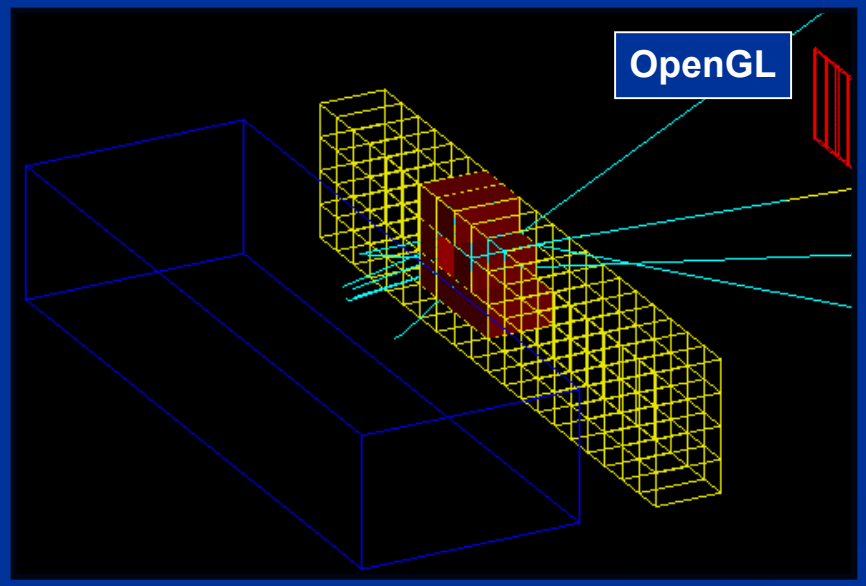
Geant4 v9.4



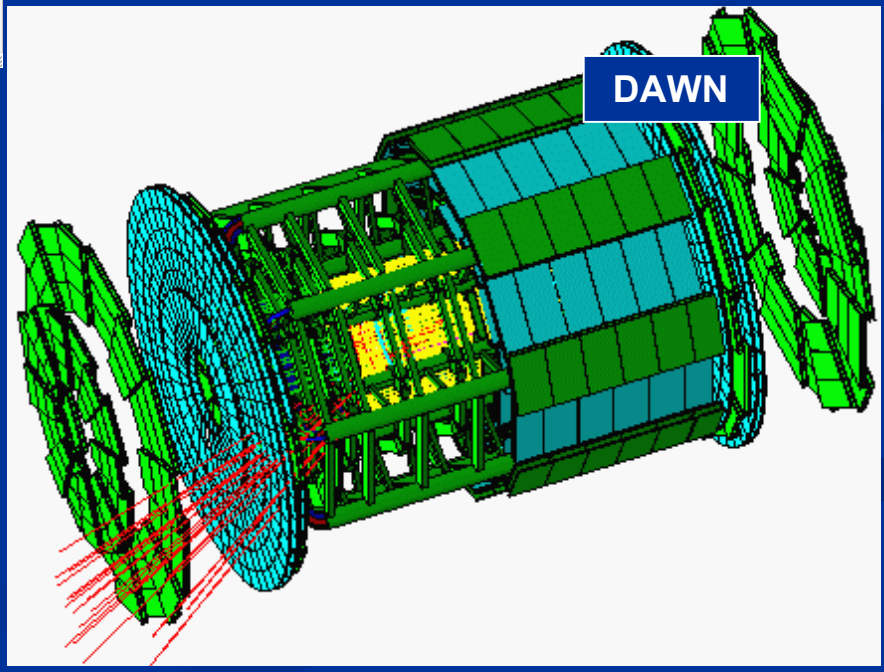
HepRep/HepRApp

Joseph Perl, SLAC

Basic Visualization Commands



OpenGL



DAWN

How this Document Fits with Other Tutorial Materials

- This presentation can be used on its own, but gives the most comprehensive introduction to Geant4 visualization when used as part of the following full set of documents:
  - Introduction to Geant4 Visualization
  - Geant4 Installation Guides
  - Geant4 Visualization Tutorial using the HepRApp HepRep Browser
  - Geant4 Visualization Tutorial using the DAWN Event Display
  - Geant4 Visualization Tutorial using the OpenGL Event Display
  - Geant4 Visualization Commands
  - Geant4 Advanced Visualization
  - See the URLS at the end of this presentation

What's in this Presentation

- This presentation covers the most basic commands for Geant Visualization (plus a few interesting small digressions)
- Further presentations cover more advanced topics (see references at the end of this presentation)
  - Geant4 Advanced Visualization:
    - Visualization Attributes
      - to control color, line style, etc.
      - to represent particle type, charge, etc.
    - Trajectory Modeling
    - Advanced Trajectory Modeling
    - Trajectory and Hit Filtering
    - Controlling level of detail in geometry
    - Section planes
    - Reviewing kept events
    - Standalone visualization
  - How to Make a Movie

Simplest Example

- Visualize your geometry in OpenGL:
  - /vis/open OGL
  - /vis/drawVolume

Most examples will come with a visualization macro as follows (1 of 3)

```
# Use this open statement to create an OpenGL view:
/vis/open OGL 600x600-0+0
#
# Use this open statement to create a .prim file suitable for viewing in DAWN:
#/vis/open DAWNFILE
#
# Use this open statement to create a .heprep file suitable for viewing in HepRApp:
#/vis/open HepRepFile
#
# Use this open statement to create a .wrl file suitable for viewing in a VRML viewer:
#/vis/open VRML2FILE
#
# Disable auto refresh and quieten vis messages whilst scene and trajectories are established:
/vis/viewer/set/autoRefresh false
/vis/verbose errors
#
# Draw geometry:
/vis/drawVolume
```

Most examples will come with a visualization macro as follows (2 of 3)

```
# Specify view angle:
#/vis/viewer/set/viewpointThetaPhi 90. 0.
#
# Specify zoom value:
#/vis/viewer/zoom 2.
#
# Specify style (surface or wireframe):
#/vis/viewer/set/style wireframe
#
# Draw coordinate axes:
#/vis/scene/add/axes 0 0 0 1 m
#
# Draw smooth trajectories at end of event, showing trajectory points as markers 2 pixels wide:
/vis/scene/add/trajectories smooth
/vis/modeling/trajectories/create/drawByCharge
/vis/modeling/trajectories/drawByCharge-0/default/setDrawStepPts true
/vis/modeling/trajectories/drawByCharge-0/default/setStepPtsSize 2
# (if too many tracks cause core dump => /tracking/storeTrajectory 0)
#
# Draw hits at end of event:
#/vis/scene/add/hits
```

Most examples will come with a visualization macro as follows (3 of 3)

```
# # To draw only gammas:
#/vis/filtering/trajectories/create/particleFilter
#/vis/filtering/trajectories/particleFilter-0/add gamma
#
# To invert the above, drawing all particles except gammas, keep the above two lines but also add:
#/vis/filtering/trajectories/particleFilter-0/invert true
#
# Many other options are available with /vis/modeling and /vis/filtering.
# For example, to select colour by particle ID:
#/vis/modeling/trajectories/create/drawByParticleID
#/vis/modeling/trajectories/drawByParticleID-0/set e- blue
#
# To superimpose all of the events from a given run:
#/vis/scene/endOfEventAction accumulate
#
# Re-establish auto refreshing and verbosity:
/vis/viewer/set/autoRefresh true
/vis/verbose warnings
#
# For file-based drivers, use this to create an empty detector view:
#/vis/viewer/flush
```

```
/vis/open OGL 600x600-0+0
#/vis/open DAWNFILE
#/vis/open HepRepFile
#/vis/open VRML2FILE
/vis/viewer/set/autoRefresh false
/vis/verbose errors
/vis/drawVolume
#/vis/viewer/set/viewpointThetaPhi 90. 0.
#/vis/viewer/zoom 2.
#/vis/viewer/set/style wireframe
#/vis/scene/add/axes 0 0 0 1 m
/vis/scene/add/trajectories smooth
/vis/modeling/trajectories/create/drawByCharge
/vis/modeling/trajectories/drawByCharge-0/default/setDrawStepPts true
/vis/modeling/trajectories/drawByCharge-0/default/setStepPtsSize 2
#/vis/scene/add/hits
#/vis/filtering/trajectories/create/particleFilter
#/vis/filtering/trajectories/particleFilter-0/add gamma
#/vis/filtering/trajectories/particleFilter-0/invert true
#/vis/modeling/trajectories/create/drawByParticleID
#/vis/modeling/trajectories/drawByParticleID-0/set e- blue
#/vis/scene/endOfEventAction accumulate
/vis/viewer/set/autoRefresh true
/vis/verbose warnings
#/vis/viewer/flush
```

Here's that long macro with the comments removed.

I'll cover all of this commands in this presentation.

/vis/open OGL 600x600-0+0

#/vis/open DAWNFILE

#/vis/open HepRepFile

#/vis/open VRML2FILE

/vis/viewer/set/autoRefresh false

/vis/verbose errors

/vis/drawVolume

#/vis/viewer/set/viewpointThetaPhi 90. 0.

#/vis/viewer/zoom 2.

#/vis/viewer/set/style wireframe

#/vis/scene/add/axes 0 0 0 1 m

/vis/scene/add/trajectories smooth

/vis/modeling/trajectories/create/drawByCharge

/vis/modeling/trajectories/drawByCharge-0/default/setDrawStepPts true

/vis/modeling/trajectories/drawByCharge-0/default/setStepPtsSize 2

#/vis/scene/add/hits

#/vis/filtering/trajectories/create/particleFilter

#/vis/filtering/trajectories/particleFilter-0/add gamma

#/vis/filtering/trajectories/particleFilter-0/invert true

#/vis/modeling/trajectories/create/drawByParticleID

#/vis/modeling/trajectories/drawByParticleID-0/set e- blue

#/vis/scene/endOfEventAction accumulate

/vis/viewer/set/autoRefresh true

/vis/verbose warnings

#/vis/viewer/flush

Simplest Command Sequence

To Open Visualization

- To Open a Driver
  - /vis/open <driver name>
- for example
  - /vis/open OGL
  - /vis/open DAWNFILE
  - /vis/open HepRepFile
  - /vis/open VRML2FILE
- The set of available drivers is listed when you first start Geant4, but you can also get this list with the command:
  - help /vis/open
- For the OpenGL drivers, there are options with very slightly different behavior:
  - OGLS: use OpenGL's "Stored" mode.
  - OGLI: use OpenGL's "Immediate" mode.
  - OGL: same as OGLS unless starts using too much memory, then switches to OGLI
  - Most users will not notice any difference. It has to do with whether Geant4 needs to resend all of the graphics information to the OpenGL driver for every new picture (Immediate mode) or can just send the new parts of the picture, getting the rest from storage already in the OpenGL driver (Stored mode).
  - And if you have Qt or Motif libraries installed and configured, OGL will use these.
  - Don't worry about this for now, but if you have problems with one mode, try the other.

More about Open, and about DrawVolume

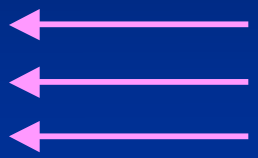
- Some drivers have additional options at open
  - e.g., for OpenGL, can specify size and location of window
    /vis/open OGL 600x600-0+0
- You can open more than one viewer at a time:
  - /vis/open OGL
  - /vis/open HepRepFile
  - To see what viewers you then have:
    - /vis/viewer/list
  - To select which viewer is the current one:
    - /vis/viewer/select viewer-0
    - /vis/viewer/select viewer-1
  - All other vis commands affect only the currently selected viewer
- To draw the entire detector geometry:
  - /vis/drawVolume
- There are many additional options to let you draw only a selected subset of the detector geometry.
  - See separate presentation, Geant4 Advanced Visualization

What we've covered so far

```
/vis/open OGL 600x600-0+0
#/vis/open DAWNFILE
#/vis/open HepRepFile
#/vis/open VRML2FILE
/vis/viewer/set/autoRefresh false
/vis/verbose errors
/vis/drawVolume
#/vis/viewer/set/viewpointThetaPhi 90. 0.
#/vis/viewer/zoom 2.
#/vis/viewer/set/style wireframe
#/vis/scene/add/axes 0 0 0 1 m
/vis/scene/add/trajectories smooth
/vis/modeling/trajectories/create/drawByCharge
/vis/modeling/trajectories/drawByCharge-0/default/setDrawStepPts true
/vis/modeling/trajectories/drawByCharge-0/default/setStepPtsSize 2
#/vis/scene/add/hits
#/vis/filtering/trajectories/create/particleFilter
#/vis/filtering/trajectories/particleFilter-0/add gamma
#/vis/filtering/trajectories/particleFilter-0/invert true
#/vis/modeling/trajectories/create/drawByParticleID
#/vis/modeling/trajectories/drawByParticleID-0/set e- blue
#/vis/scene/endOfEventAction accumulate
/vis/viewer/set/autoRefresh true
/vis/verbose warnings
#/vis/viewer/flush
```

/vis/open OGL 600x600-0+0
#/vis/open DAWNFILE
#/vis/open HepRepFile
#/vis/open VRML2FILE
/vis/viewer/set/autoRefresh false
/vis/verbose errors
/vis/drawVolume
/vis/viewer/set/viewpointThetaPhi 90. 0.
/vis/viewer/zoom 2.
/vis/viewer/set/style wireframe
#/vis/scene/add/axes 0 0 0 1 m
/vis/scene/add/trajectories smooth
/vis/modeling/trajectories/create/drawByCharge
/vis/modeling/trajectories/drawByCharge-0/default/setDrawStepPts true
/vis/modeling/trajectories/drawByCharge-0/default/setStepPtsSize 2
#/vis/scene/add/hits
#/vis/filtering/trajectories/create/particleFilter
#/vis/filtering/trajectories/particleFilter-0/add gamma
#/vis/filtering/trajectories/particleFilter-0/invert true
#/vis/modeling/trajectories/create/drawByParticleID
#/vis/modeling/trajectories/drawByParticleID-0/set e- blue
#/vis/scene/endOfEventAction accumulate
/vis/viewer/set/autoRefresh true
/vis/verbose warnings
#/vis/viewer/flush

Controlling the viewpoint and zoom



The /vis/viewer/… Commands

- Only needed if using an immediate viewer, such as OpenGL
  - For HepRepFile or DAWNFILE, these sorts of adjustments are made later, in the HepRApp or DAWN viewer programs
- Set view angles
  - /vis/viewer/set/viewpointThetaPhi <theta_angle> <phi_angle>
  - for example
    - /vis/viewer/set/viewpointThetaPhi 90. 0.
- Zoom
  - /vis/viewer/zoom <scale factor>
  - for example
    - /vis/viewer/zoom 2.
- Reset viewpoint
  - /vis/viewer/reset
- Set drawing style
  - /vis/viewer/set/style <style>
  - for example
    - /vis/viewer/set/style wireframe
    - /vis/viewer/set/style surface
  - but note that this will not affect volumes that have style explicitly forced by “setForceWireframe” or “setForceSolid” commands in the c++ code

What we've covered so far

```
/vis/open OGL 600x600-0+0
#/vis/open DAWNFILE
#/vis/open HepRepFile
#/vis/open VRML2FILE
/vis/viewer/set/autoRefresh false
/vis/verbose errors
/vis/drawVolume
/vis/viewer/set/viewpointThetaPhi 90. 0.
/vis/viewer/zoom 2.
/vis/viewer/set/style wireframe
#/vis/scene/add/axes 0 0 0 1 m
/vis/scene/add/trajectories smooth
/vis/modeling/trajectories/create/drawByCharge
/vis/modeling/trajectories/drawByCharge-0/default/setDrawStepPts true
/vis/modeling/trajectories/drawByCharge-0/default/setStepPtsSize 2
#/vis/scene/add/hits
#/vis/filtering/trajectories/create/particleFilter
#/vis/filtering/trajectories/particleFilter-0/add gamma
#/vis/filtering/trajectories/particleFilter-0/invert true
#/vis/modeling/trajectories/create/drawByParticleID
#/vis/modeling/trajectories/drawByParticleID-0/set e- blue
#/vis/scene/endOfEventAction accumulate
/vis/viewer/set/autoRefresh true
/vis/verbose warnings
#/vis/viewer/flush
```

```
/vis/open OGL 600x600-0+0
#/vis/open DAWNFILE
#/vis/open HepRepFile
#/vis/open VRML2FILE
/vis/viewer/set/autoRefresh false
/vis/verbose errors
/vis/drawVolume
/vis/viewer/set/viewpointThetaPhi 90. 0.
/vis/viewer/zoom 2.
/vis/viewer/set/style wireframe
/vis/scene/add/axes 0 0 0 1 m
/vis/scene/add/trajectories smooth
/vis/modeling/trajectories/create/drawByCharge
/vis/modeling/trajectories/drawByCharge-0/default/setDrawStepPts true
/vis/modeling/trajectories/drawByCharge-0/default/setStepPtsSize 2
/vis/scene/add/hits
#/vis/filtering/trajectories/create/particleFilter
#/vis/filtering/trajectories/particleFilter-0/add gamma
#/vis/filtering/trajectories/particleFilter-0/invert true
#/vis/modeling/trajectories/create/drawByParticleID
#/vis/modeling/trajectories/drawByParticleID-0/set e- blue
#/vis/scene/endOfEventAction accumulate
/vis/viewer/set/autoRefresh true
/vis/verbose warnings
#/vis/viewer/flush
```

Add axes, trajectories and hits

Axes, Trajectories and Hits

- Axes
  - /vis/scene/add/axes <x_origin> <y_origin> <z_origin> <size> <units>
  - for example
    - /vis/scene/add/axes 0 0 0 1 m
- Trajectories
  - /vis/scene/add/trajectories
  - By default, trajectories are redrawn at every event
    - /run/beamOn 1
- Hits
  - /vis/scene/add/hits
    - Note that not all examples contain code to create hits, so in some cases this command will add nothing to the display

/vis/open OGL 600x600-0+0
#/vis/open DAWNFILE
#/vis/open HepRepFile
#/vis/open VRML2FILE
/vis/viewer/set/autoRefresh false
/vis/verbose errors
/vis/drawVolume
/vis/viewer/set/viewpointThetaPhi 90. 0.
/vis/viewer/zoom 2.
/vis/viewer/set/style wireframe
/vis/scene/add/axes 0 0 0 1 m
/vis/scene/add/trajectories smooth
/vis/modeling/trajectories/create/drawByCharge
/vis/modeling/trajectories/drawByCharge-0/default/setDrawStepPts true
/vis/modeling/trajectories/drawByCharge-0/default/setStepPtsSize 2
/vis/scene/add/hits
#/vis/filtering/trajectories/create/particleFilter
#/vis/filtering/trajectories/particleFilter-0/add gamma
#/vis/filtering/trajectories/particleFilter-0/invert true
#/vis/modeling/trajectories/create/drawByParticleID
#/vis/modeling/trajectories/drawByParticleID-0/set e- blue
#/vis/scene/endOfEventAction accumulate
/vis/viewer/set/autoRefresh true
/vis/verbose warnings
#/vis/viewer/flush

What we've covered so far

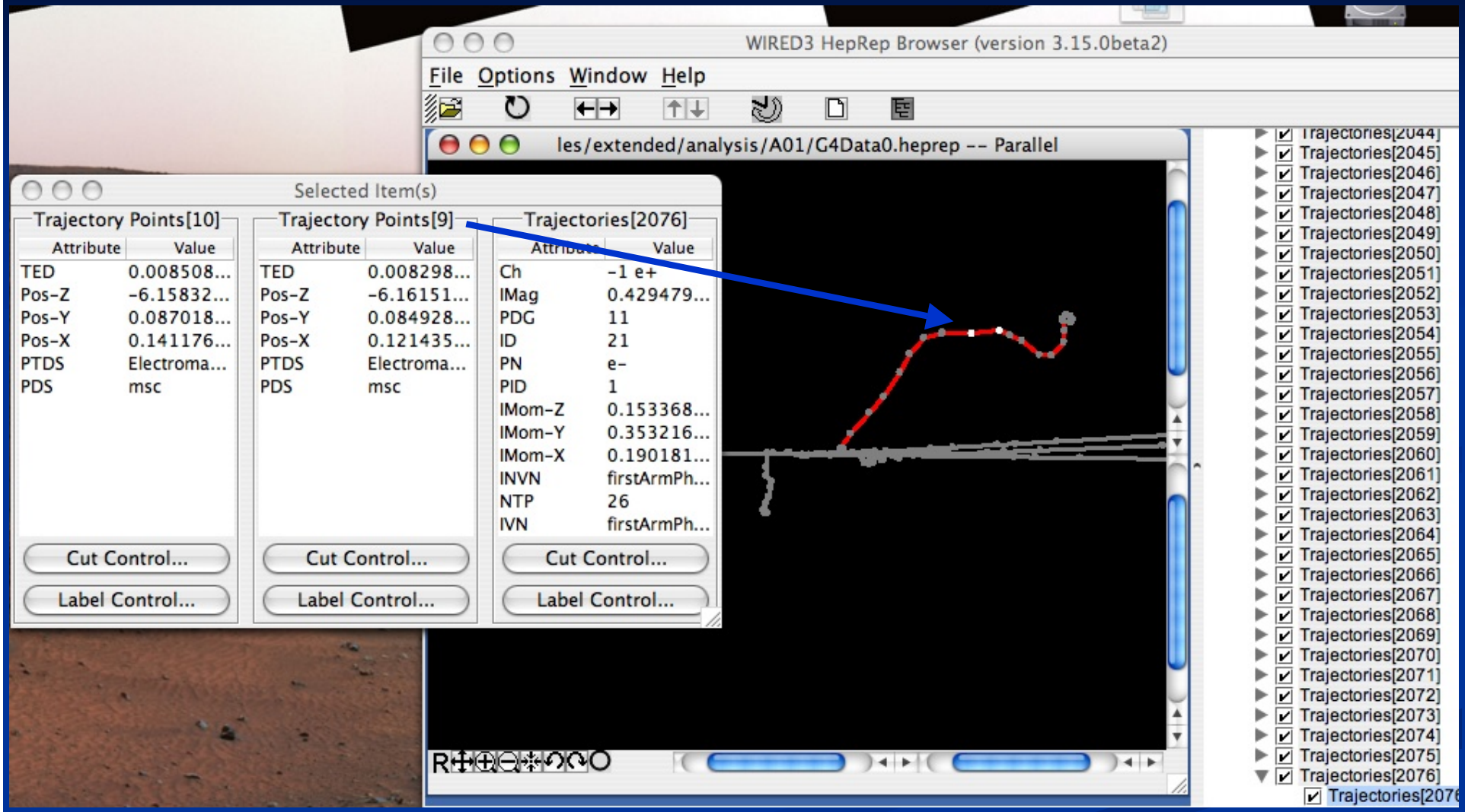
Visualizing step points

```
/vis/open OGL 600x600-0+0
#/vis/open DAWNFILE
#/vis/open HepRepFile
#/vis/open VRML2FILE
/vis/viewer/set/autoRefresh false
/vis/verbose errors
/vis/drawVolume
/vis/viewer/set/viewpointThetaPhi 90. 0.
/vis/viewer/zoom 2.
/vis/viewer/set/style wireframe
/vis/scene/add/axes 0 0 0 1 m
/vis/scene/add/trajectories smooth
/vis/modeling/trajectories/create/drawByCharge
/vis/modeling/trajectories/drawByCharge-0/default/setDrawStepPts true
/vis/modeling/trajectories/drawByCharge-0/default/setStepPtsSize 2
/vis/scene/add/hits
#/vis/filtering/trajectories/create/particleFilter
#/vis/filtering/trajectories/particleFilter-0/add gamma
#/vis/filtering/trajectories/particleFilter-0/invert true
/vis/modeling/trajectories/create/drawByParticleID
/vis/modeling/trajectories/drawByParticleID-0/set e- blue
/vis/scene/endOfEventAction accumulate
/vis/viewer/set/autoRefresh true
/vis/verbose warnings
#/vis/viewer/flush
```

Visualizing Step Points

- By default, the trajectory is drawn just as a line
- To also show the step points:
  - /vis/modeling/trajectories/create/drawByCharge
  - /vis/modeling/trajectories/drawByCharge-0/default/setDrawStepPts true
  - /vis/modeling/trajectories/drawByCharge-0/default/setStepPtsSize 2
  - This syntax is complicated because it actually supports many more options on how trajectories and step points should be modeled.
    - See separate presentation, Geant4 Advanced Visualization
- Trajectories and step points can contain additional, non-displayed information
  - such as particle id, momentum, etc.
  - shown when you pick on the trajectory in some visualization drivers.
- This set of information can be made richer by specifying rich trajectories:
  - /vis/scene/add/trajectories rich

Rich Trajectory Has Details on Every Step Point



Rich Trajectory and Points

G4RichTrajectory:

- Extra:
  - Creator Process Name
  - Creator Process Type Name
  - Charge (Ch): unit: e+
  - Ending Process Name
  - Ending Process Type Name
  - Final kinetic energy
  - Final Next Volume Path
  - Final Volume Path
- Already in regular Trajectory:
  - Track ID
  - Initial kinetic energy
  - Initial momentum magnitude
  - Initial momentum
  - Initial Next Volume Path
  - Initial Volume Path
  - No. of points
  - PDG Encoding
  - Parent ID
  - Particle Name

G4RichTrajectoryPoint:

- Extra:
  - Auxiliary Point Position
  - Process Defined Step
  - Process Type Defined
  - Position
  - Post-step-point global time
  - Post-step Volume Path
  - Pre-step-point global time
  - Pre-step Volume Path
  - Remaining Energy
  - Total Energy Deposit
- Already in regular TrajectoryPoint
  - nothing is included by default

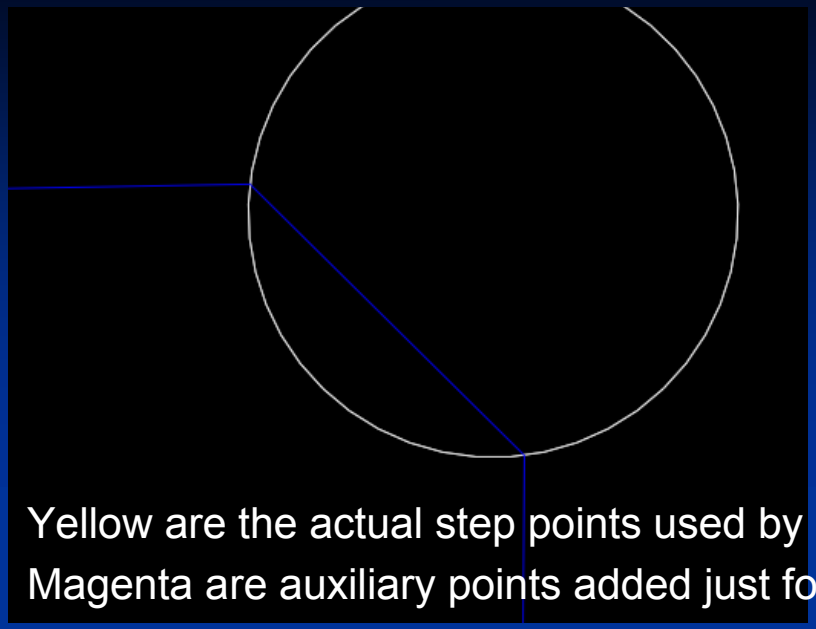
What we've covered so far

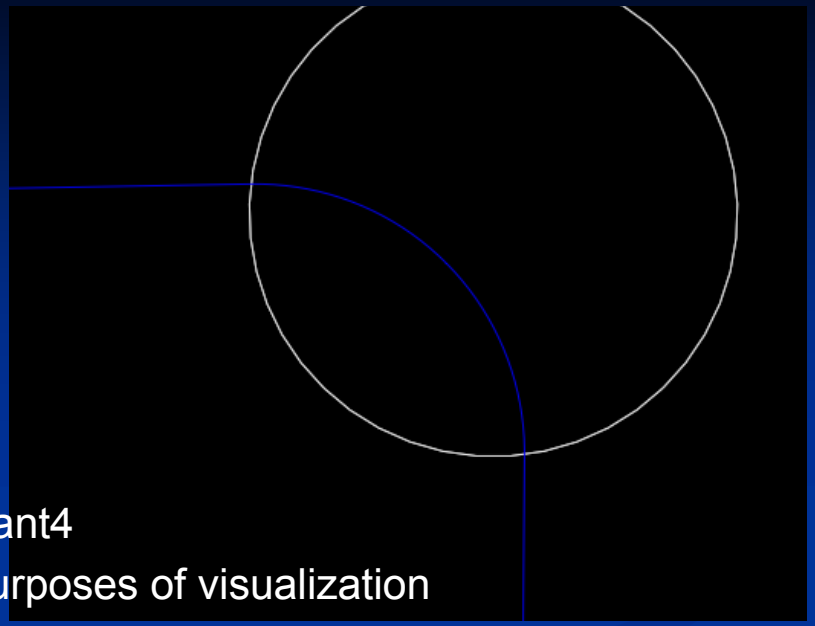
```
/vis/open OGL 600x600-0+0
#/vis/open DAWNFILE
#/vis/open HepRepFile
#/vis/open VRML2FILE
/vis/viewer/set/autoRefresh false
/vis/verbose errors
/vis/drawVolume
/vis/viewer/set/viewpointThetaPhi 90. 0.
/vis/viewer/zoom 2.
/vis/viewer/set/style wireframe
/vis/scene/add/axes 0 0 0 1 m
/vis/scene/add/trajectories smooth
/vis/modeling/trajectories/create/drawByCharge
/vis/modeling/trajectories/drawByCharge-0/default/setDrawStepPts true
/vis/modeling/trajectories/drawByCharge-0/default/setStepPtsSize 2
/vis/scene/add/hits
#/vis/filtering/trajectories/create/particleFilter
#/vis/filtering/trajectories/particleFilter-0/add gamma
#/vis/filtering/trajectories/particleFilter-0/invert true
#/vis/modeling/trajectories/create/drawByParticleID
#/vis/modeling/trajectories/drawByParticleID-0/set e- blue
#/vis/scene/endOfEventAction accumulate
/vis/viewer/set/autoRefresh true
/vis/verbose warnings
#/vis/viewer/flush
```

Smooth Trajectories

```
/vis/open OGL 600x600-0+0
#/vis/open DAWNFILE
#/vis/open HepRepFile
#/vis/open VRML2FILE
/vis/viewer/set/autoRefresh false
/vis/verbose errors
/vis/drawVolume
/vis/viewer/set/viewpointThetaPhi 90. 0.
/vis/viewer/zoom 2.
/vis/viewer/set/style wireframe
/vis/scene/add/axes 0 0 0 1 m
/vis/scene/add/trajectories smooth
/vis/modeling/trajectories/create/drawByCharge
/vis/modeling/trajectories/drawByCharge-0/default/setDrawStepPts true
/vis/modeling/trajectories/drawByCharge-0/default/setStepPtsSize 2
/vis/scene/add/hits
#/vis/filtering/trajectories/create/particleFilter
#/vis/filtering/trajectories/particleFilter-0/add gamma
#/vis/filtering/trajectories/particleFilter-0/invert true
/vis/modeling/trajectories/create/drawByParticleID
/vis/modeling/trajectories/drawByParticleID-0/set e- blue
/vis/scene/endOfEventAction accumulate
/vis/viewer/set/autoRefresh true
/vis/verbose warnings
#/vis/viewer/flush
```

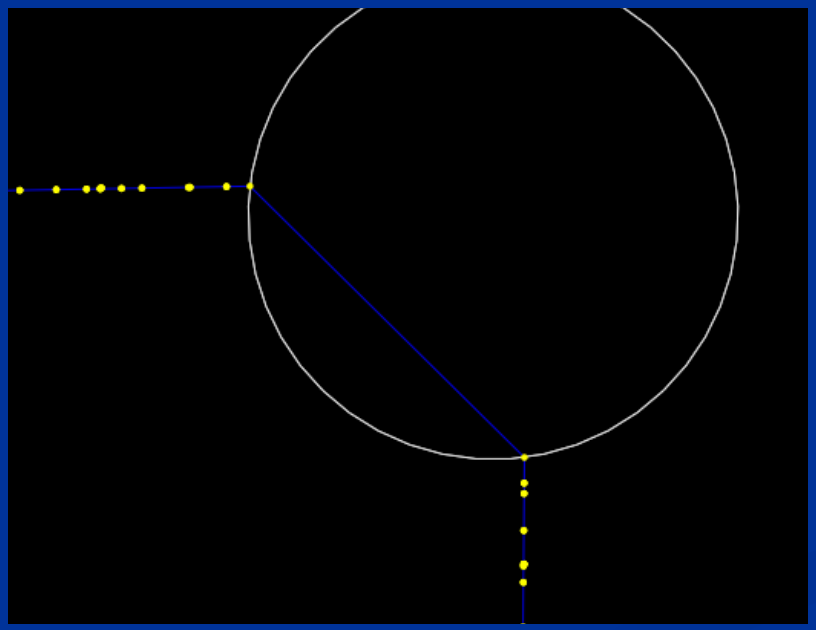
Regular versus Smooth Trajectory

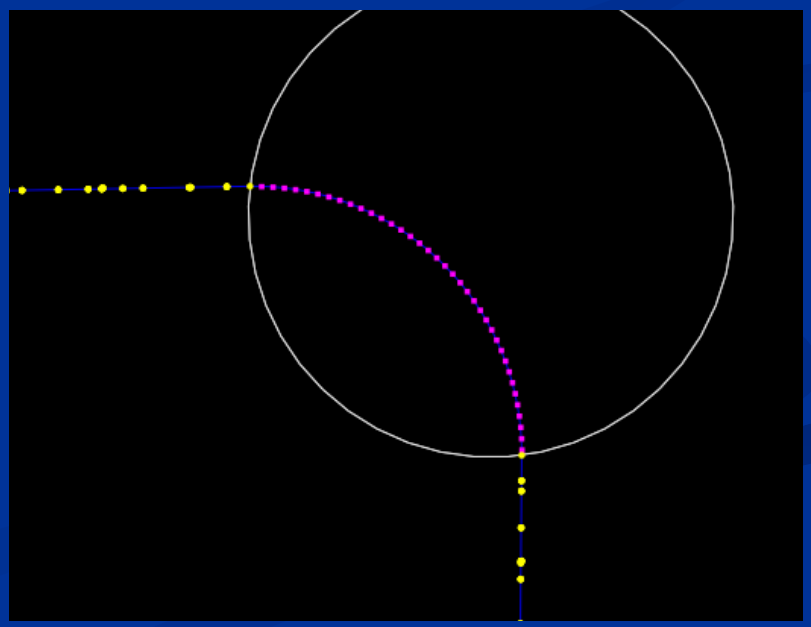




Yellow are the actual step points used by Geant4

Magenta are auxiliary points added just for purposes of visualization

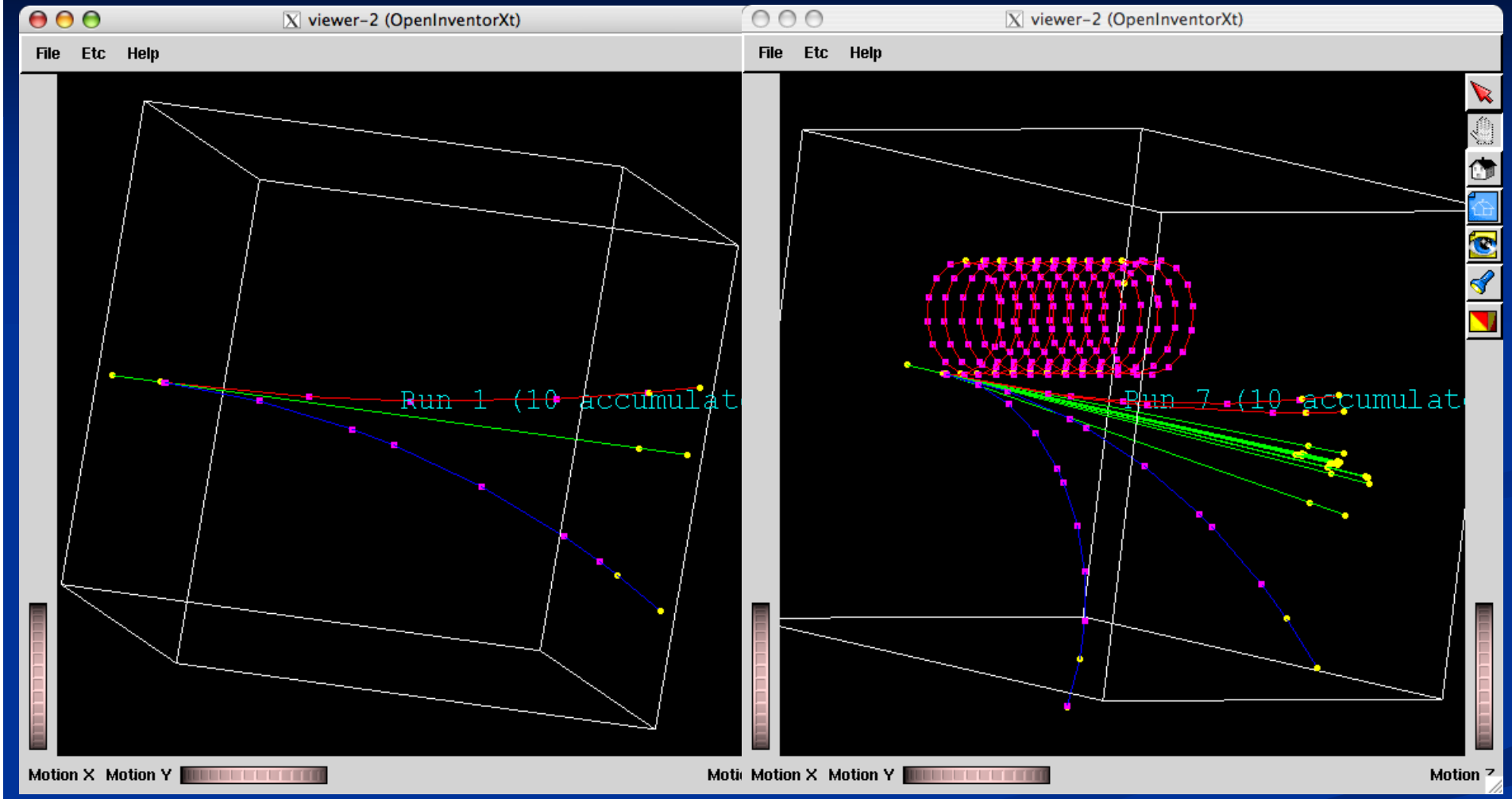




Smooth Trajectories

- By default, the trajectory is represented as a series of line segments from one step point to the next.
  - For the case of strong fields, this may result in jagged looking tracks.
- Can ask visualization to smooth the lines with:
  - /vis/scene/add/trajectories smooth
  - The extra points are not actual Geant4 step points.
    Smooth does not change how Geant4 actually does its stepping.
    These extra “auxiliary points” are only added to make a smoother line.
- Trajectories can be smooth, rich or both:
  - /vis/scene/add/trajectories smooth rich
- We’ll cover more details on this in a later presentation

Smooth Trajectory Makes Big Difference for Trajectories that Loop in a Magnetic Field



- Yellow dots are the actual step points used by Geant4
- Magenta dots are auxiliary points added just for purposes of visualization

/vis/open OGL 600x600-0+0
#/vis/open DAWNFILE
#/vis/open HepRepFile
#/vis/open VRML2FILE
/vis/viewer/set/autoRefresh false
/vis/verbose errors
/vis/drawVolume
/vis/viewer/set/viewpointThetaPhi 90. 0.
/vis/viewer/zoom 2.
/vis/viewer/set/style wireframe
/vis/scene/add/axes 0 0 0 1 m
/vis/scene/add/trajectories smooth
/vis/modeling/trajectories/create/drawByCharge
/vis/modeling/trajectories/drawByCharge-0/default/setDrawStepPts true
/vis/modeling/trajectories/drawByCharge-0/default/setStepPtsSize 2
/vis/scene/add/hits
#/vis/filtering/trajectories/create/particleFilter
#/vis/filtering/trajectories/particleFilter-0/add gamma
#/vis/filtering/trajectories/particleFilter-0/invert true
/vis/modeling/trajectories/create/drawByParticleID
/vis/modeling/trajectories/drawByParticleID-0/set e- blue
/vis/scene/endOfEventAction accumulate
/vis/viewer/set/autoRefresh true
/vis/verbose warnings
#/vis/viewer/flush

What we've covered so far

Basic trajectory modeling

```
/vis/open OGL 600x600-0+0
#/vis/open DAWNFILE
#/vis/open HepRepFile
#/vis/open VRML2FILE
/vis/viewer/set/autoRefresh false
/vis/verbose errors
/vis/drawVolume
/vis/viewer/set/viewpointThetaPhi 90. 0.
/vis/viewer/zoom 2.
/vis/viewer/set/style wireframe
/vis/scene/add/axes 0 0 0 1 m
/vis/scene/add/trajectories smooth
/vis/modeling/trajectories/create/drawByCharge
/vis/modeling/trajectories/drawByCharge-0/default/setDrawStepPts true
/vis/modeling/trajectories/drawByCharge-0/default/setStepPtsSize 2
/vis/scene/add/hits
#/vis/filtering/trajectories/create/particleFilter
#/vis/filtering/trajectories/particleFilter-0/add gamma
#/vis/filtering/trajectories/particleFilter-0/invert true
/vis/modeling/trajectories/create/drawByParticleID
/vis/modeling/trajectories/drawByParticleID-0/set e- blue
#/vis/scene/endOfEventAction accumulate
/vis/viewer/set/autoRefresh true
/vis/verbose warnings
#/vis/viewer/flush
```

Basic Trajectory Modeling

- By default, trajectories are color-coded by charge
  - positive = blue
  - neutral = green
  - negative = red
- But you can choose other modeling options, such as color by particle ID
  - /vis/modeling/trajectories/create/drawByParticleID
  - /vis/modeling/trajectories/drawByParticleID-0/set e- blue
- Right now, when you first turn on drawByParticleID, all particles are set to grey, and you then have to assign any colors you want.
- Starting in the next release, the default will be to already assign some color codes (which you can then override if you want to).
- The new defaults will be equivalent to:
  - /vis/modeling/trajectories/drawByParticleID-0/set e- red
  - /vis/modeling/trajectories/drawByParticleID-0/set e+ blue
  - /vis/modeling/trajectories/drawByParticleID-0/set proton cyan
  - /vis/modeling/trajectories/drawByParticleID-0/set gamma green
  - /vis/modeling/trajectories/drawByParticleID-0/set neutron yellow
  - /vis/modeling/trajectories/drawByParticleID-0/set pi+ magenta
  - /vis/modeling/trajectories/drawByParticleID-0/set pi- magenta
  - /vis/modeling/trajectories/drawByParticleID-0/set pi0 magenta
  - # and everything else still grey

What we've covered so far

```
/vis/open OGL 600x600-0+0
#/vis/open DAWNFILE
#/vis/open HepRepFile
#/vis/open VRML2FILE
/vis/viewer/set/autoRefresh false
/vis/verbose errors
/vis/drawVolume
/vis/viewer/set/viewpointThetaPhi 90. 0.
/vis/viewer/zoom 2.
/vis/viewer/set/style wireframe
/vis/scene/add/axes 0 0 0 1 m
/vis/scene/add/trajectories smooth
/vis/modeling/trajectories/create/drawByCharge
/vis/modeling/trajectories/drawByCharge-0/default/setDrawStepPts true
/vis/modeling/trajectories/drawByCharge-0/default/setStepPtsSize 2
/vis/scene/add/hits
#/vis/filtering/trajectories/create/particleFilter
#/vis/filtering/trajectories/particleFilter-0/add gamma
#/vis/filtering/trajectories/particleFilter-0/invert true
/vis/modeling/trajectories/create/drawByParticleID
/vis/modeling/trajectories/drawByParticleID-0/set e- blue
/vis/scene/endOfEventAction accumulate
/vis/viewer/set/autoRefresh true
/vis/verbose warnings
#/vis/viewer/flush
```

Accumulating trajectories and hits

```
/vis/open OGL 600x600-0+0
#/vis/open DAWNFILE
#/vis/open HepRepFile
#/vis/open VRML2FILE
/vis/viewer/set/autoRefresh false
/vis/verbose errors
/vis/drawVolume
/vis/viewer/set/viewpointThetaPhi 90. 0.
/vis/viewer/zoom 2.
/vis/viewer/set/style wireframe
/vis/scene/add/axes 0 0 0 1 m
/vis/scene/add/trajectories smooth
/vis/modeling/trajectories/create/drawByCharge
/vis/modeling/trajectories/drawByCharge-0/default/setDrawStepPts true
/vis/modeling/trajectories/drawByCharge-0/default/setStepPtsSize 2
/vis/scene/add/hits
#/vis/filtering/trajectories/create/particleFilter
#/vis/filtering/trajectories/particleFilter-0/add gamma
#/vis/filtering/trajectories/particleFilter-0/invert true
/vis/modeling/trajectories/create/drawByParticleID
/vis/modeling/trajectories/drawByParticleID-0/set e- blue
/vis/scene/endOfEventAction accumulate
/vis/viewer/set/autoRefresh true
/vis/verbose warnings
#/vis/viewer/flush
```

Accumulating Trajectories and Hits

- By default, you will get a drawing after each event. To instead get just one drawing with all of the accumulated events from that run
  - /vis/scene/endOfEventAction accumulate
- This overrides the default
  - /vis/scene/endOfEventAction refresh

- To even suppress that one drawing from the end of the /run/beamOn, use
  - /vis/scene/endOfRunAction accumulate
- This overrides the default
  - /vis/scene/endOfRunAction refresh
- When you actually want to draw, you then have to explicitly issue the command
  - /vis/viewer/flush

What we've covered so far

```
/vis/open OGL 600x600-0+0
#/vis/open DAWNFILE
#/vis/open HepRepFile
#/vis/open VRML2FILE
/vis/viewer/set/autoRefresh false
/vis/verbose errors
/vis/drawVolume
/vis/viewer/set/viewpointThetaPhi 90. 0.
/vis/viewer/zoom 2.
/vis/viewer/set/style wireframe
/vis/scene/add/axes 0 0 0 1 m
/vis/scene/add/trajectories smooth
/vis/modeling/trajectories/create/drawByCharge
/vis/modeling/trajectories/drawByCharge-0/default/setDrawStepPts true
/vis/modeling/trajectories/drawByCharge-0/default/setStepPtsSize 2
/vis/scene/add/hits
#/vis/filtering/trajectories/create/particleFilter
#/vis/filtering/trajectories/particleFilter-0/add gamma
#/vis/filtering/trajectories/particleFilter-0/invert true
/vis/modeling/trajectories/create/drawByParticleID
/vis/modeling/trajectories/drawByParticleID-0/set e- blue
/vis/scene/endOfEventAction accumulate
/vis/viewer/set/autoRefresh true
/vis/verbose warnings
#/vis/viewer/flush
```

Filtering Trajectories

```
/vis/open OGL 600x600-0+0
#/vis/open DAWNFILE
#/vis/open HepRepFile
#/vis/open VRML2FILE
/vis/viewer/set/autoRefresh false
/vis/verbose errors
/vis/drawVolume
/vis/viewer/set/viewpointThetaPhi 90. 0.
/vis/viewer/zoom 2.
/vis/viewer/set/style wireframe
/vis/scene/add/axes 0 0 0 1 m
/vis/scene/add/trajectories smooth
/vis/modeling/trajectories/create/drawByCharge
/vis/modeling/trajectories/drawByCharge-0/default/setDrawStepPts true
/vis/modeling/trajectories/drawByCharge-0/default/setStepPtsSize 2
/vis/scene/add/hits
/vis/filtering/trajectories/create/particleFilter
/vis/filtering/trajectories/particleFilter-0/add gamma
/vis/filtering/trajectories/particleFilter-0/invert true
/vis/modeling/trajectories/create/drawByParticleID
/vis/modeling/trajectories/drawByParticleID-0/set e- blue
/vis/scene/endOfEventAction accumulate
/vis/viewer/set/autoRefresh true
/vis/verbose warnings
#/vis/viewer/flush
```

Filtering Trajectories

- By default, all trajectories are drawn
- You apply a filter so that only certain trajectories are drawn:
  - /vis/filtering/trajectories/create/particleFilter
  - /vis/filtering/trajectories/particleFilter-0/add gamma
- The above adds a filter that only allows gammas to draw
- To instead do the opposite, drawing everying except gammas, include the above, but also add the following:
  - /vis/filtering/trajectories/particleFilter-0/invert true
- We'll cover more details on this in a later presentation (including how to filter hits)

What we've covered so far

```
/vis/open OGL 600x600-0+0
#/vis/open DAWNFILE
#/vis/open HepRepFile
#/vis/open VRML2FILE
/vis/viewer/set/autoRefresh false
/vis/verbose errors
/vis/drawVolume
/vis/viewer/set/viewpointThetaPhi 90. 0.
/vis/viewer/zoom 2.
/vis/viewer/set/style wireframe
/vis/scene/add/axes 0 0 0 1 m
/vis/scene/add/trajectories smooth
/vis/modeling/trajectories/create/drawByCharge
/vis/modeling/trajectories/drawByCharge-0/default/setDrawStepPts true
/vis/modeling/trajectories/drawByCharge-0/default/setStepPtsSize 2
/vis/scene/add/hits
/vis/filtering/trajectories/create/particleFilter
/vis/filtering/trajectories/particleFilter-0/add gamma
/vis/filtering/trajectories/particleFilter-0/invert true
/vis/modeling/trajectories/create/drawByParticleID
/vis/modeling/trajectories/drawByParticleID-0/set e- blue
/vis/scene/endOfEventAction accumulate
/vis/viewer/set/autoRefresh true
/vis/verbose warnings
#/vis/viewer/flush
```

```
/vis/open OGL 600x600-0+0
#/vis/open DAWNFILE
#/vis/open HepRepFile
#/vis/open VRML2FILE
/vis/viewer/set/autoRefresh false
/vis/verbose errors
/vis/drawVolume
/vis/viewer/set/viewpointThetaPhi 90. 0.
/vis/viewer/zoom 2.
/vis/viewer/set/style wireframe
/vis/scene/add/axes 0 0 0 1 m
/vis/scene/add/trajectories smooth
/vis/modeling/trajectories/create/drawByCharge
/vis/modeling/trajectories/drawByCharge-0/default/setDrawStepPts true
/vis/modeling/trajectories/drawByCharge-0/default/setStepPtsSize 2
/vis/scene/add/hits
/vis/filtering/trajectories/create/particleFilter
/vis/filtering/trajectories/particleFilter-0/add gamma
/vis/filtering/trajectories/particleFilter-0/invert true
/vis/modeling/trajectories/create/drawByParticleID
/vis/modeling/trajectories/drawByParticleID-0/set e- blue
/vis/scene/endOfEventAction accumulate
/vis/viewer/set/autoRefresh true
/vis/verbose warnings
/vis/viewer/flush
```

To force output of a new file

To Force Output of a New File

- There are two classes of Geant4 visualization drivers:
  - Immediate drivers: draw directly to the screen (such as OpenGL)
  - File-based drivers: create a file on disk (HepRepFile, VRML2FILE)
    - You open this file later in that separate application
- For immediate drivers
  - you see the results or your /vis commands immediately
- For the file-based drivers
  - the default is to only create a new file (showing your changes) when you do /run/beamOn
- If you want to see visualization at some other time, such as after you set up geometry, but before you do /run/beamOn, use:
  - /vis/viewer/flush

What we've covered so far

```
/vis/open OGL 600x600-0+0
#/vis/open DAWNFILE
#/vis/open HepRepFile
#/vis/open VRML2FILE
/vis/viewer/set/autoRefresh false
/vis/verbose errors
/vis/drawVolume
/vis/viewer/set/viewpointThetaPhi 90. 0.
/vis/viewer/zoom 2.
/vis/viewer/set/style wireframe
/vis/scene/add/axes 0 0 0 1 m
/vis/scene/add/trajectories smooth
/vis/modeling/trajectories/create/drawByCharge
/vis/modeling/trajectories/drawByCharge-0/default/setDrawStepPts true
/vis/modeling/trajectories/drawByCharge-0/default/setStepPtsSize 2
/vis/scene/add/hits
/vis/filtering/trajectories/create/particleFilter
/vis/filtering/trajectories/particleFilter-0/add gamma
/vis/filtering/trajectories/particleFilter-0/invert true
/vis/modeling/trajectories/create/drawByParticleID
/vis/modeling/trajectories/drawByParticleID-0/set e- blue
/vis/scene/endOfEventAction accumulate
/vis/viewer/set/autoRefresh true
/vis/verbose warnings
/vis/viewer/flush
```

```
/vis/open OGL 600x600-0+0
#/vis/open DAWNFILE
#/vis/open HepRepFile
#/vis/open VRML2FILE
/vis/viewer/set/autoRefresh false
/vis/verbose errors
/vis/drawVolume
/vis/viewer/set/viewpointThetaPhi 90. 0.
/vis/viewer/zoom 2.
/vis/viewer/set/style wireframe
/vis/scene/add/axes 0 0 0 1 m
/vis/scene/add/trajectories smooth
/vis/modeling/trajectories/create/drawByCharge
/vis/modeling/trajectories/drawByCharge-0/default/setDrawStepPts true
/vis/modeling/trajectories/drawByCharge-0/default/setStepPtsSize 2
/vis/scene/add/hits
/vis/filtering/trajectories/create/particleFilter
/vis/filtering/trajectories/particleFilter-0/add gamma
/vis/filtering/trajectories/particleFilter-0/invert true
/vis/modeling/trajectories/create/drawByParticleID
/vis/modeling/trajectories/drawByParticleID-0/set e- blue
/vis/scene/endOfEventAction accumulate
/vis/viewer/set/autoRefresh true
/vis/verbose warnings
#/vis/viewer/flush
```

To avoid excessive redrawing on immediate viewers

To Avoid Excessive Redrawing

- By default, immediate viewers will redraw after every vis command that might change the view, such as:
  - /vis/drawVolume
  - /vis/viewer/set/viewpointThetaPhi 90. 0.
  - /vis/viewer/zoom 2.
  - /vis/viewer/set/style wireframe
  - /vis/scene/add/axes 0 0 0 1 m
- If the geometry is very complex (such as in some imported patient geometries for medical applications), this can result in slow performance
  - the above would redraw five times
- To temporarily turn off this redrawing:
  - /vis/viewer/set/autoRefresh false
- And then once everything is set up:
  - /vis/viewer/set/autoRefresh true
- Note that there is no need for a /vis/viewer/flush at that point, since the autoRefresh true will trigger a redraw

/vis/open OGL 600x600-0+0
#/vis/open DAWNFILE
#/vis/open HepRepFile
#/vis/open VRML2FILE
/vis/viewer/set/autoRefresh false
/vis/verbose errors
/vis/drawVolume
/vis/viewer/set/viewpointThetaPhi 90. 0.
/vis/viewer/zoom 2.
/vis/viewer/set/style wireframe
/vis/scene/add/axes 0 0 0 1 m
/vis/scene/add/trajectories smooth
/vis/modeling/trajectories/create/drawByCharge
/vis/modeling/trajectories/drawByCharge-0/default/setDrawStepPts true
/vis/modeling/trajectories/drawByCharge-0/default/setStepPtsSize 2
/vis/scene/add/hits
/vis/filtering/trajectories/create/particleFilter
/vis/filtering/trajectories/particleFilter-0/add gamma
/vis/filtering/trajectories/particleFilter-0/invert true
/vis/modeling/trajectories/create/drawByParticleID
/vis/modeling/trajectories/drawByParticleID-0/set e- blue
/vis/scene/endOfEventAction accumulate
/vis/viewer/set/autoRefresh true
/vis/verbose warnings
#/vis/viewer/flush

What we've covered so far

```
/vis/open OGL 600x600-0+0
#/vis/open DAWNFILE
#/vis/open HepRepFile
#/vis/open VRML2FILE
/vis/viewer/set/autoRefresh false
/vis/verbose errors
/vis/drawVolume
/vis/viewer/set/viewpointThetaPhi 90. 0.
/vis/viewer/zoom 2.
/vis/viewer/set/style wireframe
/vis/scene/add/axes 0 0 0 1 m
/vis/scene/add/trajectories smooth
/vis/modeling/trajectories/create/drawByCharge
/vis/modeling/trajectories/drawByCharge-0/default/setDrawStepPts true
/vis/modeling/trajectories/drawByCharge-0/default/setStepPtsSize 2
/vis/scene/add/hits
/vis/filtering/trajectories/create/particleFilter
/vis/filtering/trajectories/particleFilter-0/add gamma
/vis/filtering/trajectories/particleFilter-0/invert true
/vis/modeling/trajectories/create/drawByParticleID
/vis/modeling/trajectories/drawByParticleID-0/set e- blue
/vis/scene/endOfEventAction accumulate
/vis/viewer/set/autoRefresh true
/vis/verbose warnings
#/vis/viewer/flush
```

To turn off unwanted visualization messages on the console

To Turn off Unwanted Visualization Messages

- You can control how many messages visualization puts on the console by:
  - /vis/verbose <level>

0) quiet, // Nothing is printed.
1) startup, // Startup and endup messages are printed...
2) errors, // ...and errors...
3) warnings, // ...and warnings...
4) confirmations, // ...and confirming messages...
5) parameters, // ...and parameters of scenes and views...
6) all // ...and everything available.

```
/vis/open OGL 600x600-0+0
#/vis/open DAWNFILE
#/vis/open HepRepFile
#/vis/open VRML2FILE
/vis/viewer/set/autoRefresh false
/vis/verbose errors
/vis/drawVolume
/vis/viewer/set/viewpointThetaPhi 90. 0.
/vis/viewer/zoom 2.
/vis/viewer/set/style wireframe
/vis/scene/add/axes 0 0 0 1 m
/vis/scene/add/trajectories smooth
/vis/modeling/trajectories/create/drawByCharge
/vis/modeling/trajectories/drawByCharge-0/default/setDrawStepPts true
/vis/modeling/trajectories/drawByCharge-0/default/setStepPtsSize 2
/vis/scene/add/hits
/vis/filtering/trajectories/create/particleFilter
/vis/filtering/trajectories/particleFilter-0/add gamma
/vis/filtering/trajectories/particleFilter-0/invert true
/vis/modeling/trajectories/create/drawByParticleID
/vis/modeling/trajectories/drawByParticleID-0/set e- blue
/vis/scene/endOfEventAction accumulate
/vis/viewer/set/autoRefresh true
/vis/verbose warnings
#/vis/viewer/flush
```

We've covered all of it

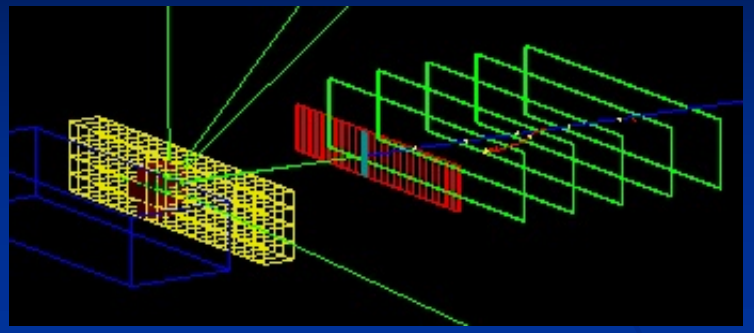
Now on to some extra topics

Printing from OpenGL

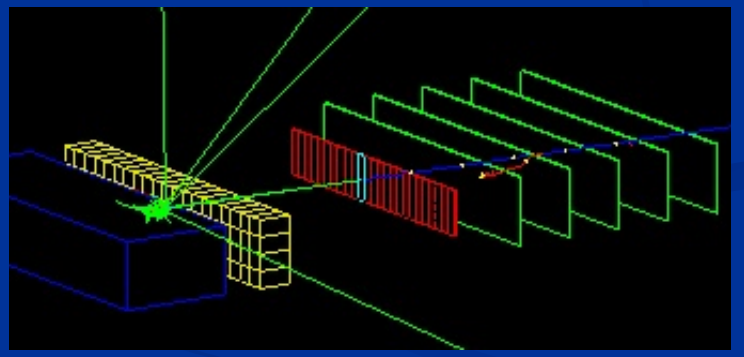
- Each of the visualization drivers provides its own way to print files.
  - RayTracer directly produces a gif file
  - OpenInventor, HepRep, DAWN and VRML rely on external application to provide a rich set of printing options
  - OpenGL printing is less obvious but quite powerful
- The OpenGL drivers can directly produce PostScript graphics
  - Options for both Vector and Bitmapped
- Open your GL viewer and set up the view as usual
  - /vis/open OGL
  - /vis/drawVolume
  - /vis/viewer/zoom 2.
  - /vis/viewer/set/viewpointThetaPhi 30. 30.
- Then print
  - /vis/ogl/printEPS
- Extra options allows you to control output style and transparency
  - /vis/ogl/set/printMode vectored or pixmap
  - /vis/ogl/set/transparency True or False

Hidden Line Removal

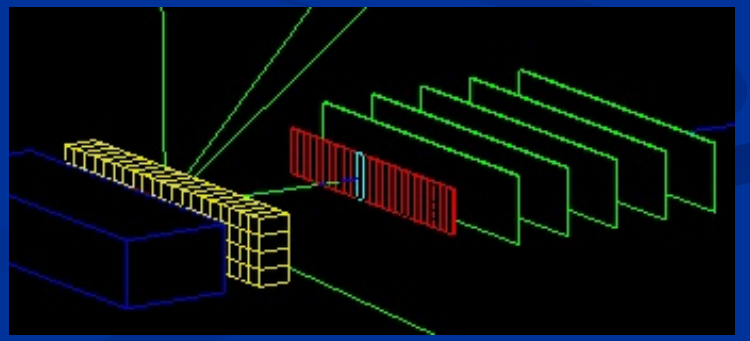
- OpenGL supports hidden line removal.
- You can control whether this removal is done and whether trajectories and hits are affected by this feature.
- By default, hidden line removal is disabled



- To turn on hidden line removal
  - /vis/viewer/set/hiddenEdge 1
- This hides edges of geometry, but lets trajectories through.



- To hide trajectories and hits as well
  - /vis/viewer/set/hiddenMarker 1



Transfer View Settings Between Views

- You can transfer the view settings from one viewer to another.
  - Allows you to use a fast viewer to select a special view, then tell a slower viewer to render this same view
- For example, set up a view in OpenGL
  - /vis/open OGL
    - by default, it will give this new viewer the name “viewer-0”
  - /vis/drawVolume
- Try a few values for zoom to find one that looks good
  - /vis/viewer/zoom 2.
  - /vis/viewer/zoom 1.2
- Try a few view angles to find one that looks good
  - /vis/viewer/set/viewpointThetaPhi 30. 30.
  - /vis/viewer/set/viewpointThetaPhi 20. 20.
- Now render the same view in RayTracer
  - /vis/open RayTracer
  - /vis/drawVolume
  - /vis/viewer/set/all viewer-0
    - where viewer-0 is the default name that was assigned to the OGL view
  - /vis/viewer/flush

Compound Commands

- To allow you to work quickly, Geant4 visualization lets you issue the equivalent of several common commands at one time by using a “compound command”.
- Some of the commands you have already seen in this presentation are actually compound commands:
  - /vis/open
    - /vis/sceneHandler/create
    - /vis/viewer/create
  - /vis/drawVolume
    - /vis/scene/create
    - /vis/scene/add/volume
  - /vis/viewer/flush
    - /vis/viewer/refresh
    - /vis/viewer/update
- I mention this just so that you will understand other people’s examples you see that may not contain the familiar /vis/open or /vis/drawVolume

Command Guidance

- Complete guidance on all commands is available from the command line:
  - Idle> help
  - Command directory path : /
  - Sub-directories :
  - 1) /control/ UI control commands.
  - 2) /units/ Available units.
  - 3) /geometry/ Geometry control commands.
  - 4) /tracking/ TrackingManager and SteppingManager control commands.
  - 5) /event/ EventManager control commands.
  - 6) /run/ Run control commands.
  - 7) /random/ Random number status control commands.
  - 8) /particle/ Particle control commands.
  - 9) /process/ Process Table control commands.
  - 10) /vis/ Visualization commands.
  - 11) /mydet/ A01 detector setup control commands.
  - 12) /hits/ Sensitive detectors and Hits
  - 13) /gun/ Particle Gun control commands.
  - Commands :
  - Type the number (0:end, -n:n level back) :

Guidance Detail

- Guidance is hierarchical, providing full detail on all commands.
  - Sub-directories :
  - 1) /vis/ASCIITree/ Commands for ASCIITree control.
  - 2) /vis/GAGTree/ Commands for GAGTree control.
  - 3) /vis/heprep/ HepRep commands.
  - 4) /vis/rayTracer/ RayTracer commands.
  - 5) /vis/scene/ Operations on Geant4 scenes.
  - 6) /vis/sceneHandler/ Operations on Geant4 scene handlers.
  - 7) /vis/viewer/ Operations on Geant4 viewers.
  - Commands :
  - 8) enable * Enables/disables visualization system.
  - 9) disable * Disables visualization system.
  - 10) verbose * Simple graded message scheme - digit or string (1st character defines):
  - 11) drawTree * (DTREE) Creates a scene consisting of this physical volume and produces a representation of the geometry hierarchy.
  - 12) drawView * Draw view from this angle, etc.
  - 13) drawVolume * Creates a scene consisting of this physical volume and asks the current viewer to draw it.
  - 14) open * Creates a scene handler ready for drawing.
  - 15) specify * Draws logical volume with Boolean components, voxels and readout geometry.

- Command /vis/open
- Creates a scene handler ready for drawing.
- The scene handler becomes current (the name is auto-generated).
- Parameter : graphics-system-name
  - Parameter type : s
  - Omittable : False
  - Candidates : ATree DAWNFILE HepRepXML HepRepFile RayTracer VRML1FILE VRML2FILE gMocrenFile OGL OGLI OGLS RayTracerX
- Parameter : window-size-hint
- pixels
  - Parameter type : i
  - Omittable : True
  - Default value : 600

Complete List of Commands

- This presentation has shown only a very small subset of Geant4 vis commands. Even for those commands shown, only a few of the options have been presented.
- Each visualization driver may have its own set of additional commands.
- To see the complete set of commands, use the interactive command guidance (i.e., type help and then type the appropriate number for “vis”).
- Note that many of the command details are only loaded into the help system once you start using the given command
  - e.g., when you first look at the help for /vis/modeling, you will see only
    - /vis/modeling/trajectories/create
    - /vis/modeling/trajectories/list
  - But once you have done your first
    - /vis/modeling/trajectories/create/drawByParticleID
  - you will see many subcommands such as
    - /vis/modeling/trajectories/drawByParticleID-0/set
    - /vis/modeling/trajectories/drawByParticleID-0/setRGBA
    - etc.

Geant4 Visualization Resources

Geant4 Installation Guides

- http://geant4.slac.stanford.edu/installation

Hands on HepRApp Tutorial

- http://geant4.slac.stanford.edu/Presentations/vis/G4HepRAppTutorial/G4HepRAppTutorial.html

Hands on DAWN Tutorial

- http://geant4.slac.stanford.edu/Presentations/vis/G4DAWNTutorial/G4DAWNTutorial.html

Hands on OpenGL Tutorial

- http://geant4.slac.stanford.edu/Presentations/vis/G4OpenGLTutorial/G4OpenGLTutorial.html

Geant4 Visualization Commands

- http://geant4.slac.stanford.edu/Presentations/vis/G4VisCommands.ppt (and .pdf)

Geant4 Advanced Visualization

- http://geant4.slac.stanford.edu/Presentations/vis/G4VisAdvanced.ppt (and .pdf)

How to Make a Movie

- http://geant4.slac.stanford.edu/Presentations/vis/HowToMakeAMovie.ppt (and .pdf)

Visualization Chapter of the Geant4 User's Guide for Application Developers

- http://geant4.web.cern.ch/geant4/UserDocumentation/UsersGuides/ForApplicationDeveloper/html/

List of Visualization Commands:

- http://geant4.web.cern.ch/geant4/UserDocumentation/UsersGuides/ForApplicationDeveloper/html/AllResources/Control/UIcommands/_vis_.html

For Questions or Comments: Geant4 Visualization Online Forum:

- http://geant4-hn.slac.stanford.edu:5090/HyperNews/public/get/visualization.html