# Geant4 Visualization
## Advanced topics

Slides created by Joseph Perl

U.S. DEPARTMENT OF ENERGY
Office of Science

SLAC NATIONAL ACCELERATOR LABORATORY

# Simplest Example

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

Slides from J. Perl (SLAC)

```
/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
```

Many examples contain a vis.mac that demonstrates many commands. Here's that 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
```

# 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
  ```

```
/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
```

Controlling the viewpoint and zoom

# The /vis/viewer/… Commands

- Set view angles

  ```
  /vis/viewer/set/viewpointThetaPhi <theta_angle>
  <phi_angle>
  ```

- Zoom

  ```
  /vis/viewer/zoom <scale factor>
  ```

- Reset viewpoint

  ```
  /vis/viewer/reset
  ```

- Set drawing style

  ```
  /vis/viewer/set/style <style>
  ```

  - Options for style: wireframe , surface
  - but note that this will not affect volumes that have style explicitly forced by “setForceWireframe” or “setForceSolid” commands in the c++ code

```
/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
```



Add axes, trajectories and hits

# Axes, Trajectories and Hits

- Axes

  `/vis/scene/add/axes <x_origin> <y_origin> <z_origin> <size> <units>`

- Trajectories

  `/vis/scene/add/trajectories`

  - By default, trajectories are redrawn at every event, try: `/run/beamOn 1`

- Hits (if application has hits defined)

  `/vis/scene/add/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
```

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
```

## Visualizing step points

# Visualizing Step Points

- By default, the trajectory is drawn just as a line
- To also show the step points:

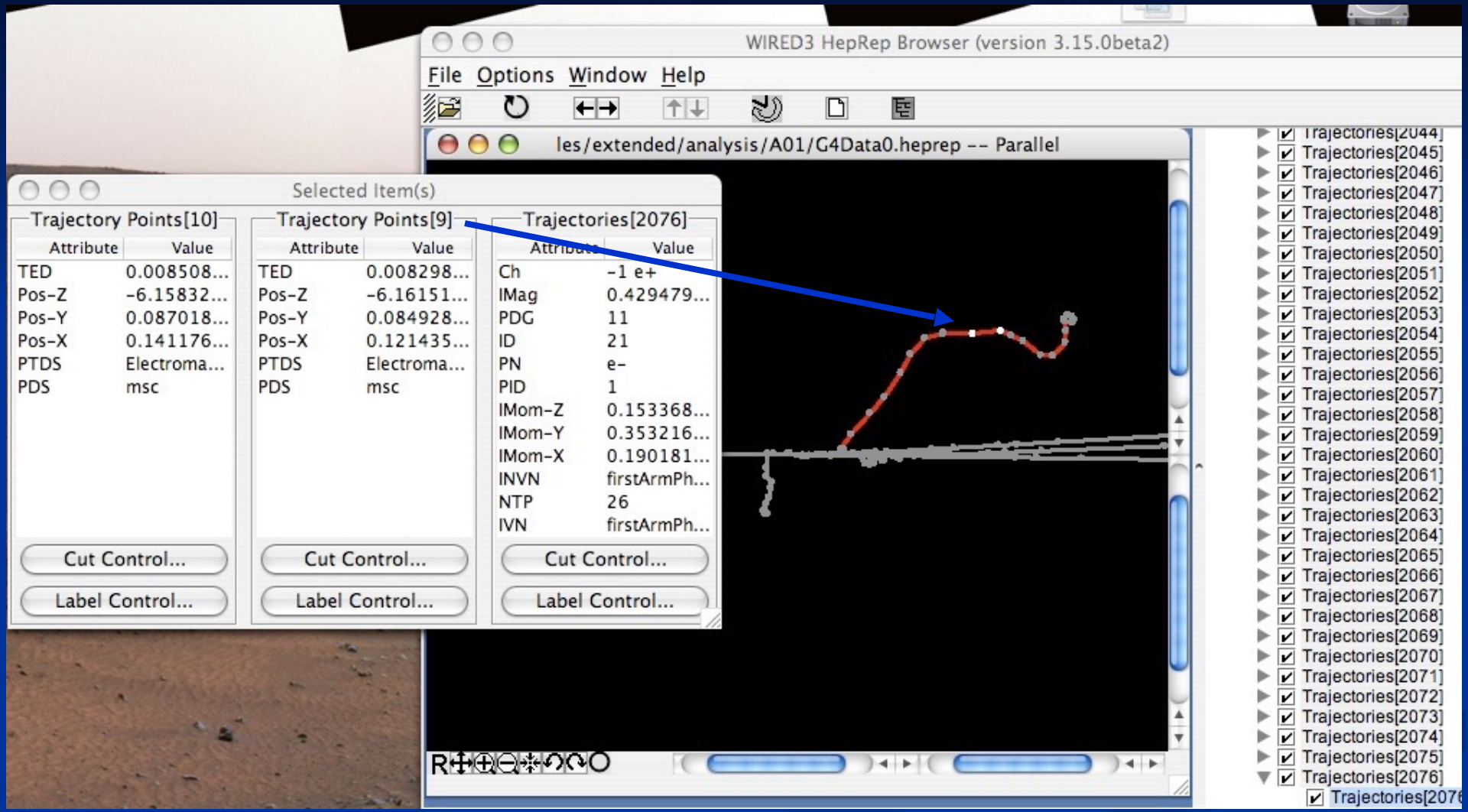
```
/vis/modeling/trajectories/create/drawByCharge
/vis/modeling/trajectories/drawByCharge-0/default/
setDrawStepPts true
/vis/modeling/trajectories/drawByCharge-0/default/
setStepPtsSize 2
```

  This syntax is complicated because it actually supports many more options on how trajectories and step points should be modeled.

- 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.
- Turn on extra info with parameter rich:

```
/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

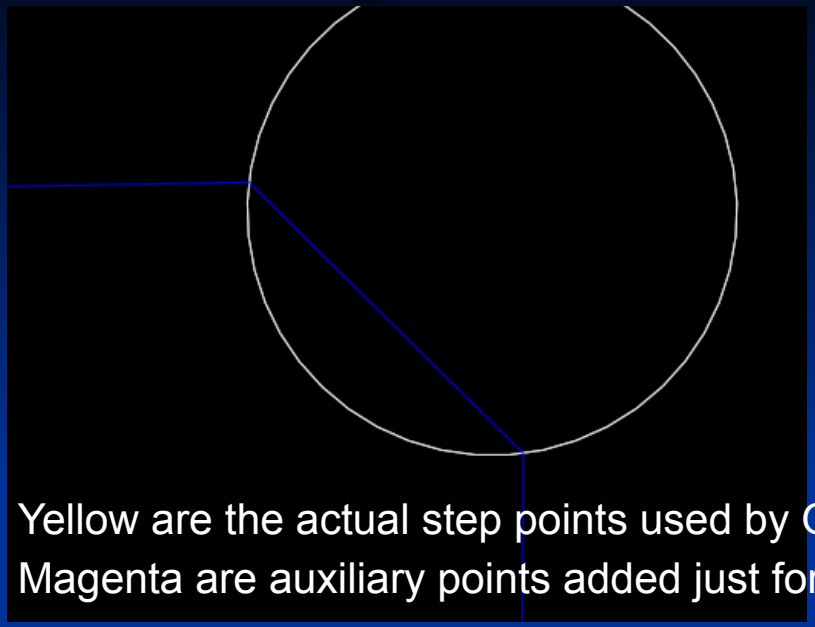
/vis/open OGL 600x600-0+0
#/vis/open DAWNFILE
#/vis/open HepRepFile
#/vis/open VRML2FILE
/vis/viewer/set/autoRefresh false
/vis/verbose errors
/vis/drawVolume
/vis/viewer/set/viewpointThetaPhi 90. 0.
/vis/viewer/zoom 2.
/vis/viewer/set/style wireframe
/vis/scene/add/axes 0 0 0 1 m
/vis/scene/add/trajectories 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

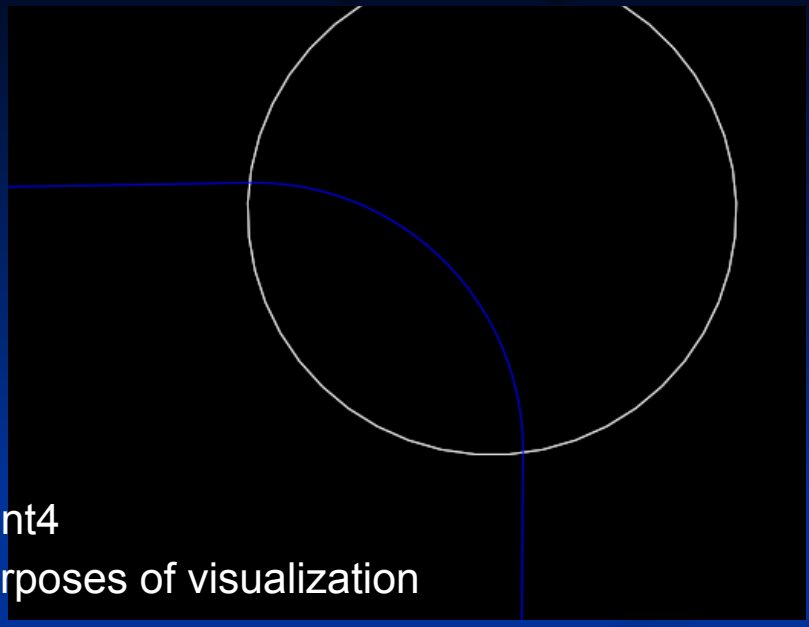
# What we've covered so far

```
/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
```

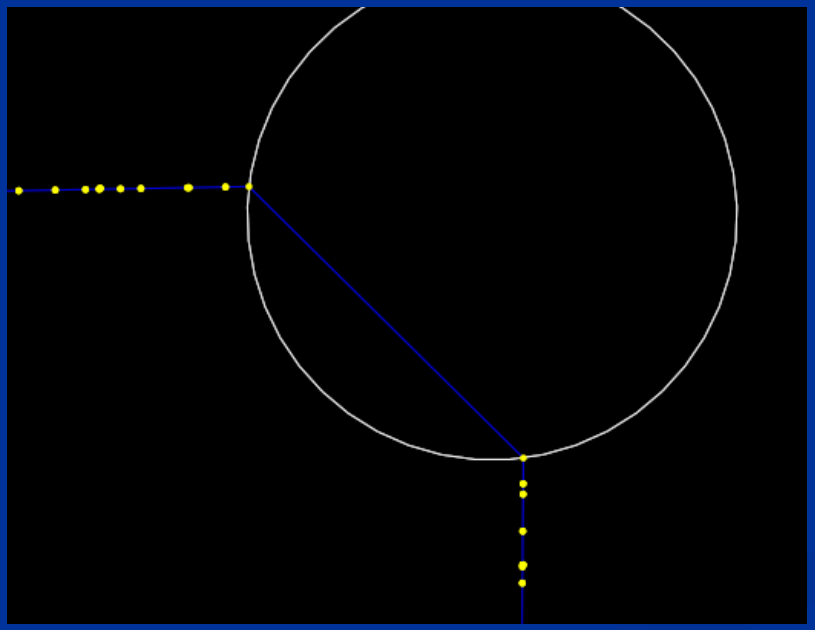
Smooth Trajectories

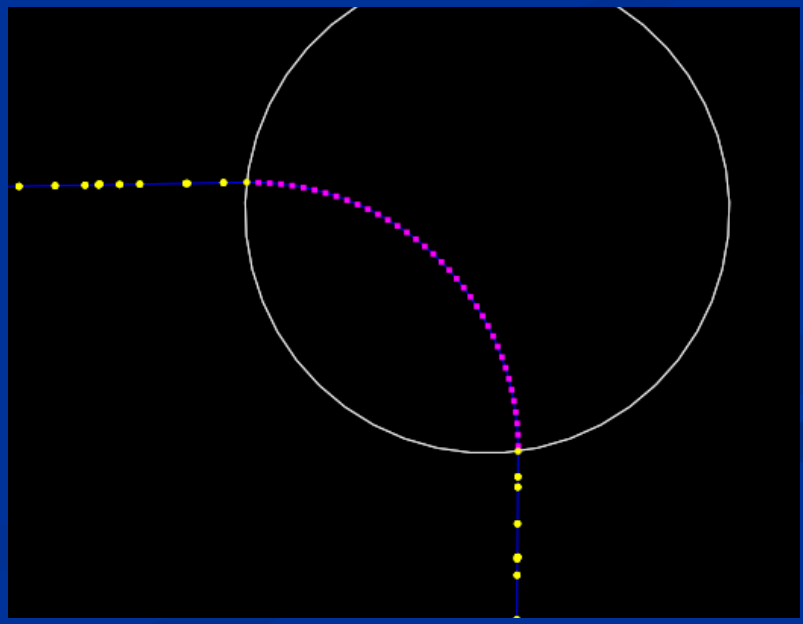
# Regular versus Smooth Trajectory





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:

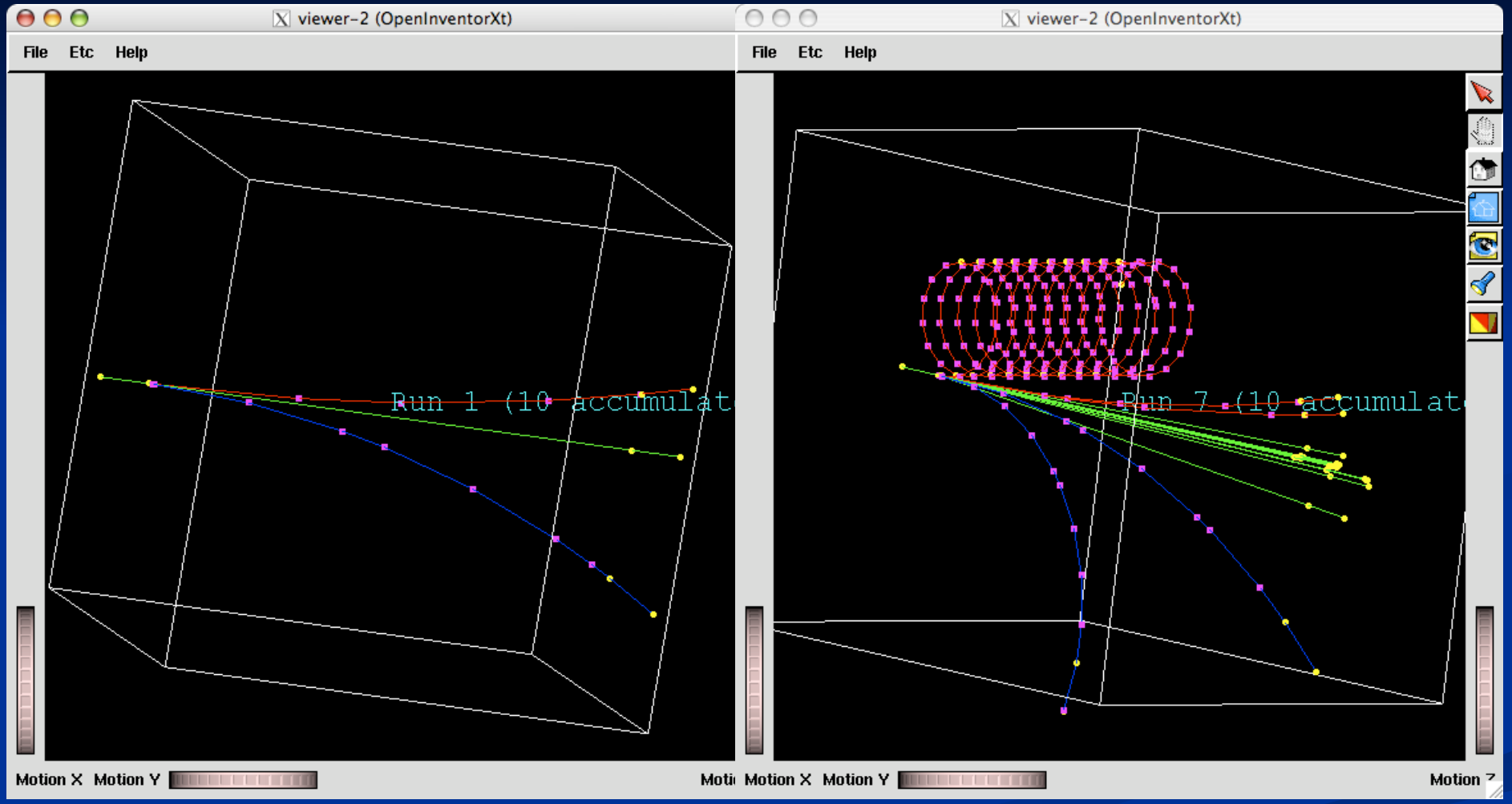
```
/vis/scene/add/trajectories smooth
```

  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
```

# 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

```
/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

# Basic Trajectory Modeling

- By default, trajectories are color-coded by charge
  - /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
- 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

/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
```

Accumulating trajectories and hits

# 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
```

- To even suppress that one drawing from the end of the /run/beamOn, use

```
/vis/scene/endOfRunAction accumulate
```

- When you actually want to draw, you then have to explicitly issue the command

```
/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

What we've covered so far

## 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 everything except gammas,
  include the above, but also add the following:

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

/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 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)
- 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`

/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 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
- To temporarily turn off this redrawing:

```
/vis/viewer/set/autoRefresh false
```

- And then once everything is set up:

```
/vis/viewer/set/autoRefresh true
```

/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

- Open your OGL 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

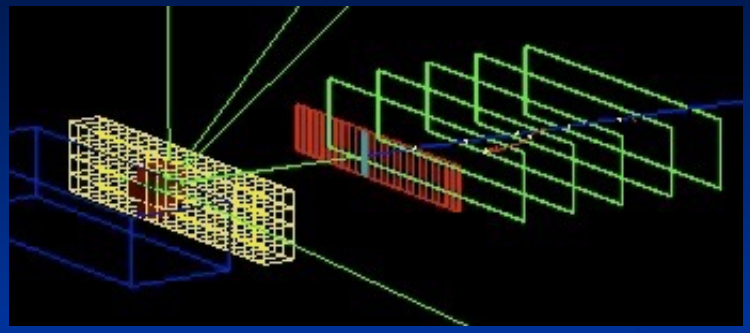
```
/vis/ogl/printEPS
```

- 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.

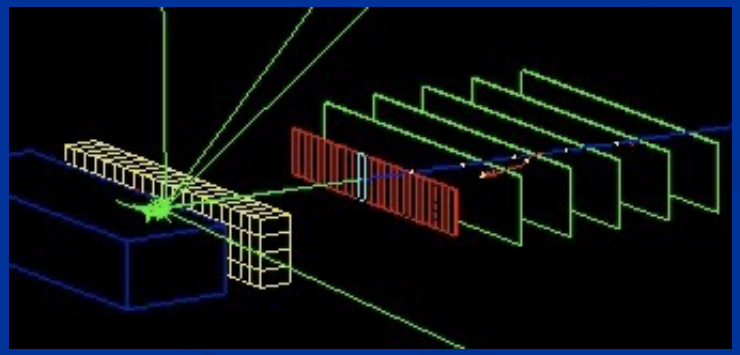


- By default, hidden line removal is disabled

- To turn on hidden line removal

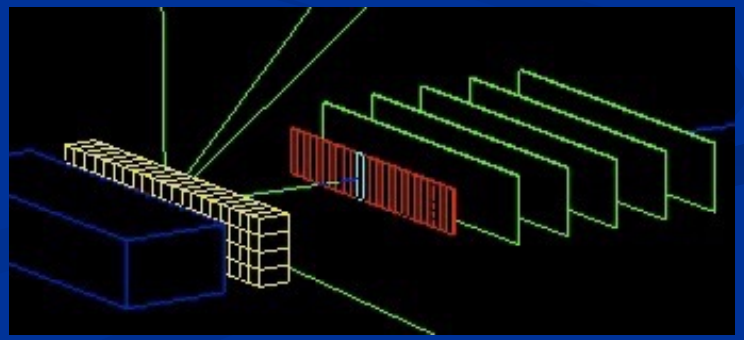
```
/vis/viewer/set/hiddenEdge 1
```

- This hides edges of geometry, but lets trajectories through.



- To hide trajectories and hits as well

```
/vis/viewer/set/hiddenMarker 1
```



# Reviewing Kept Events

- If you have accumulated several events in your visualization, you can still go back afterwards and view the events individually. For each event, you can execute various vis commands to rotate, zoom, output to a different vis driver, etc.

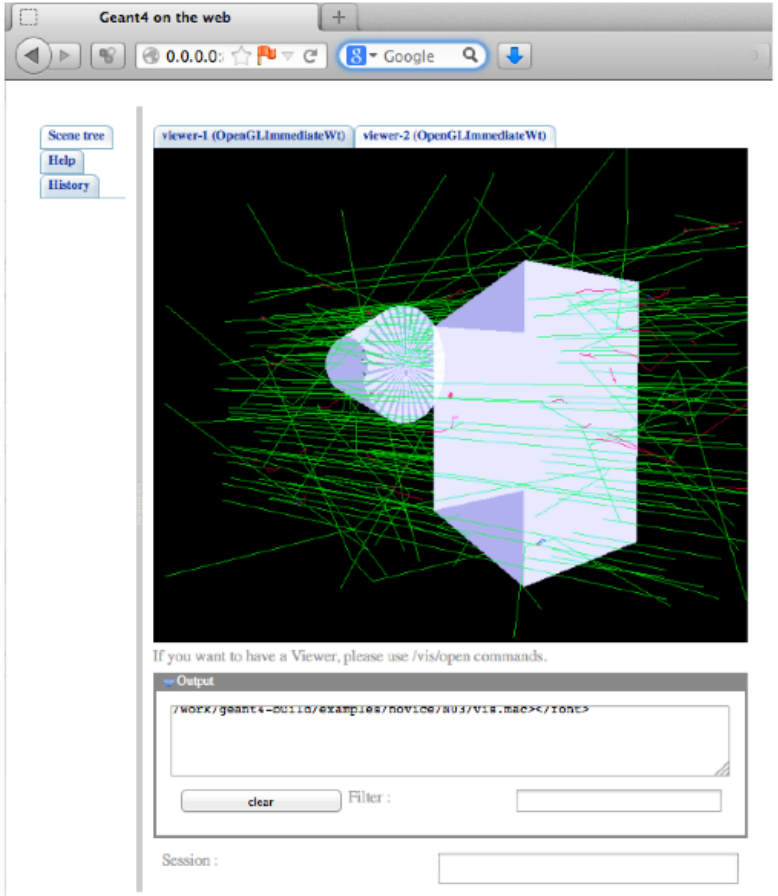
  `/vis/reviewKeptEvents`
  - Each time you type “continue”, you will get to the next kept event.
- To quit reviewing events:

  `/vis/abortReviewKeptEvents`
  - and then again type “continue”
- You can also use a command or c++ calls to force keeping of specific events regardless of how visualization is accumulating them.
  - e.g., keep events based on a particular hit or trigger pattern
- From the command line:
  - `/event/keepCurrentEvent`
- From C++
  - G4EventManager->KeepTheCurrentEvent()
- This feature makes it easy to do a large run and then recall for visualization only those events that are of interest

# References

- Geant4 Qt Home Page http://geant4.in2p3.fr/spip.php?rubrique25&lang=en
- gMocren Home Page http://geant4.kek.jp/gMocren
- DAWN Home Page http://geant4.kek.jp/~tanaka/DAWN/About_DAWN.html
  - DAWNCUT Home Page http://geant4.kek.jpJ/~tanaka/DAWN/About_DAWNCUT.html
  - DAVID Home Page http://geant4.kek.jp/~tanaka/DAWN/About_DAVID.html
  - Satoshi Tanaka's GEANT4 Ritsumeikan University Group Home Page (more information on DAWN, sample PRIM files, images, etc.) http://geant4.kek.jp/~tanaka/
- HepRApp HepRep Browser http://www.slac.stanford.edu/~perl/HepRApp
- OpenScientist Home Page http://openscientist.lal.in2p3.fr

- Under Development: Wt driver, support for browser based visualizaiton (e.g. visualization in a dynamic webpage)




Journal of Physics: Conference Series 513 (2014) 06201