

Visualization Parallel Summary

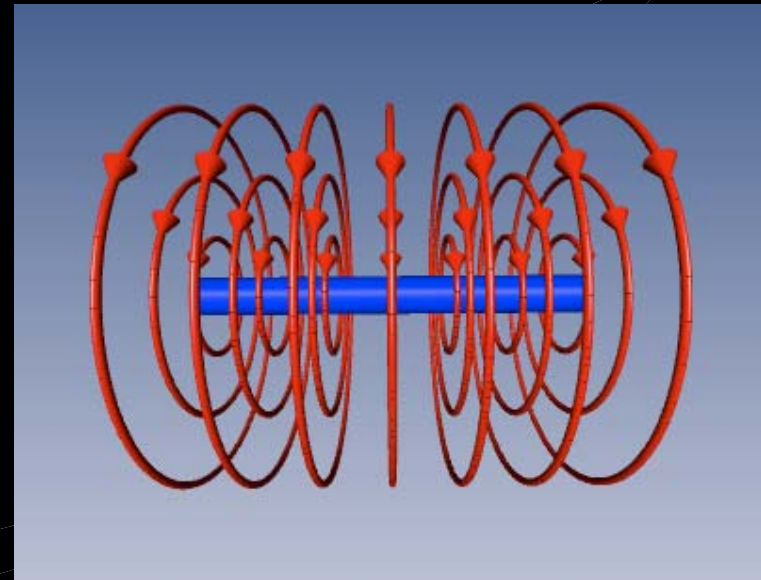
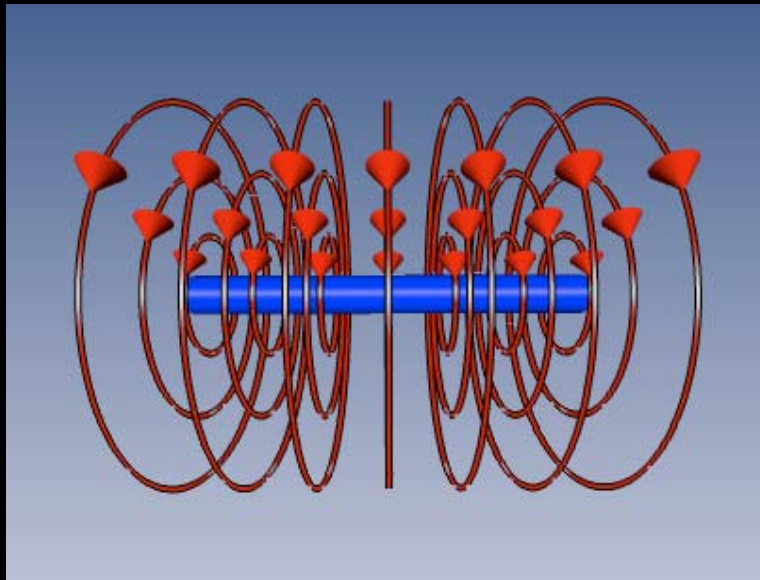
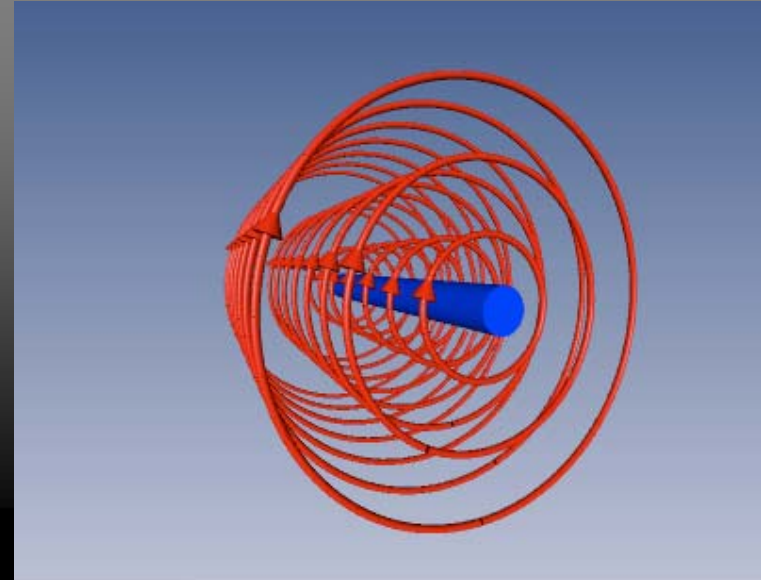
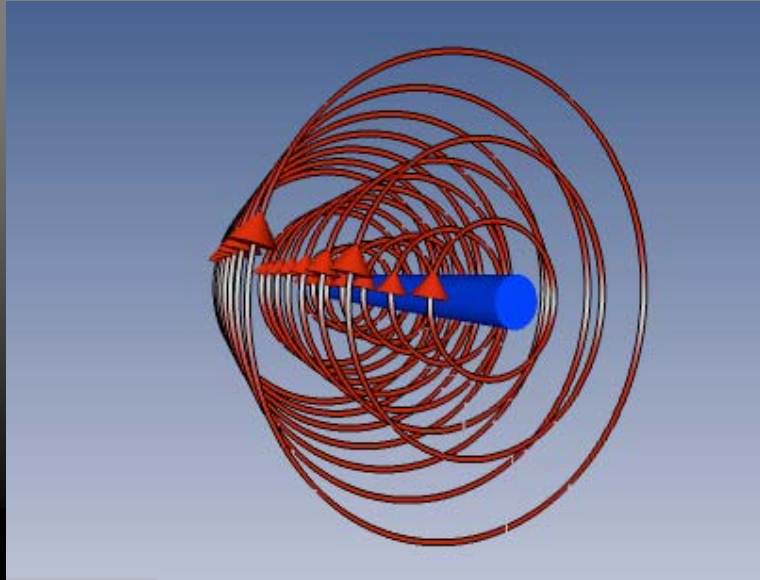
- A working session. For each topic there was both presentation and significant discussions.

Redrawing Events on Switching to a Different Vis Driver

- Presentation from John Allison.
- Needed to allow same transients in two different vis drivers.
- Use cases:
 - user declares from start that they want to send all events to two different vis drivers (e.g., OpenGL plus DAWNFILE)
 - user who scans many events in one driver (e.g. OpenGL) and then decides to send one interesting event to a different driver (e.g. DAWNFILE)
 - user displays event and then changes to different trajectory model and wants to re-display same event
- Solution will be based on changes in run management to allow keeping events

Visualization of Fields (Electric, Magnetic, etc.)

- Presentation from Jane Tinslay.
- Work is at an early stage.
- But already some impressive demonstrations...



Visualization of Fields (Electric, Magnetic, etc.)

- Current studies are using Open Inventor for testing, but eventual solution will be for all visualization drivers.
- Previous example showed evenly spaced seeding of streamlines, but eventual solution will include options for alternate seeding strategies (such as by field density)
- Aim to release some of this in June

Making a Geant4 Movie

- Presentation from John Allison.
- To make movies with just changes to camera position, has been possible for years using macros.
- Conversion to mpeg somewhat complicated
 - will be documented in next release
- Movies shown in plenary session involved additional technique of “time-slicing”, breaking trajectories into individual slices, each with a time attribute.
 - required newer visualization features, rich trajectory and some extensions to the OpenGL driver
 - available in next release
 - you can run these animations Directly from Geant4, does NOT involve stitching together a movie by hand

Movies: Stitched Together from Multiple Stills

QuickTime™ and a
YUV420 codec decompressor
are needed to see this picture.

Movies: Live from Geant4 OpenGL

QuickTime™ and a
decompressor
are needed to see this picture.

G4RichTrajectory and G4SmoothTrajectory

- Rich Trajectory encodes additional information at every step point
- Smooth Trajectory adds auxiliary points to allow smoother line in visualization (not Geant4 Steps, no physics at auxiliary points)
- Currently have to write user tracking action and instantiate either of these special kinds of trajectory
- We will move towards having this selectable from interactive commands, such as:
 - `/tracking/storeTrajectory rich`
 - `/tracking/storeTrajectory smooth`

G4RichTrajectory

The screenshot displays the WIRED3 HepRep Browser (version 3.15.0beta2) interface. The main window shows a 3D visualization of particle trajectories. A red trajectory is highlighted, and a blue arrow points from the 'Trajectories[2076]' table to it. The interface includes a menu bar (File, Options, Window, Help), a toolbar, and a file path: `les/extended/analysis/A01/G4Data0.heprep -- Parallel`. On the right, a list of trajectories is shown, with checkboxes for each item from Trajectories[2044] to Trajectories[2076].

The 'Selected Item(s)' dialog box is open, showing three tables of attributes for selected items:

Trajectory Points[10]	
Attribute	Value
TED	0.008508...
Pos-Z	-6.15832...
Pos-Y	0.087018...
Pos-X	0.141176...
PTDS	Electroma...
PDS	msc

Trajectory Points[9]	
Attribute	Value
TED	0.008298...
Pos-Z	-6.16151...
Pos-Y	0.084928...
Pos-X	0.121435...
PTDS	Electroma...
PDS	msc

Trajectories[2076]	
Attribute	Value
Ch	-1 e+
IMag	0.429479...
PDG	11
ID	21
PN	e-
PID	1
IMom-Z	0.153368...
IMom-Y	0.353216...
IMom-X	0.190181...
INVN	firstArmPh...
NTP	26
IVN	firstArmPh...

Other Topics Discussed in the Session

- `i_mode` option from `/vis/scene/add/trajectories` will be removed at 9.0
 - handled better by Trajectory Modeling Options since release 8.0
- Evaluate, Improve and Extend Visualization Examples
 - and clean up use of visualization within all novice examples
- Implementing more basic features (2/3D text, background colour, window positioning, line width, `G4Atts`,...) in all vis drivers
- Status of the Geant4 Visualization Paper
 - circulated among the vis group about three weeks ago
- Support Qt as Wrapper for Visualization
 - something to study
- Clarify/simplify user options for OGL drivers
 - explore why need separate `OGLIX`, `OGLSX`, how to clarify to users
 - why need to expose user to distinctions of `OGLxWin32` versus `OGLxX`
 - and how can we get OGL window to refresh when uncovered
- Capture and handle `ctrl-c` to return to idle prompt during visualization
 - end event loop or `RayTracerX` process but don't end the entire Geant4 session
- Need a Realistic Complex Geometry for Visualization Benchmarking
 - problem is most experiments embed geometry construction in own framework
- Parallel worlds
 - require some enhancements to visualization

Detailed List of What Was Recently Done and What is Still To Do

For the complete list of what has been done since release 8.0 and what is to do, see:

[http://geant4.slac.stanford.edu/Presentations/
vis/workplans/VisToDoList_20061011d.doc](http://geant4.slac.stanford.edu/Presentations/vis/workplans/VisToDoList_20061011d.doc)
and .pdf