

Visualisation of Geant 4 Events Within Mokka

An Application for Plugins

Adrian Vogel
DESY FLC

Visualisation Settings

Display some background in the forward region!
Play around with different settings...

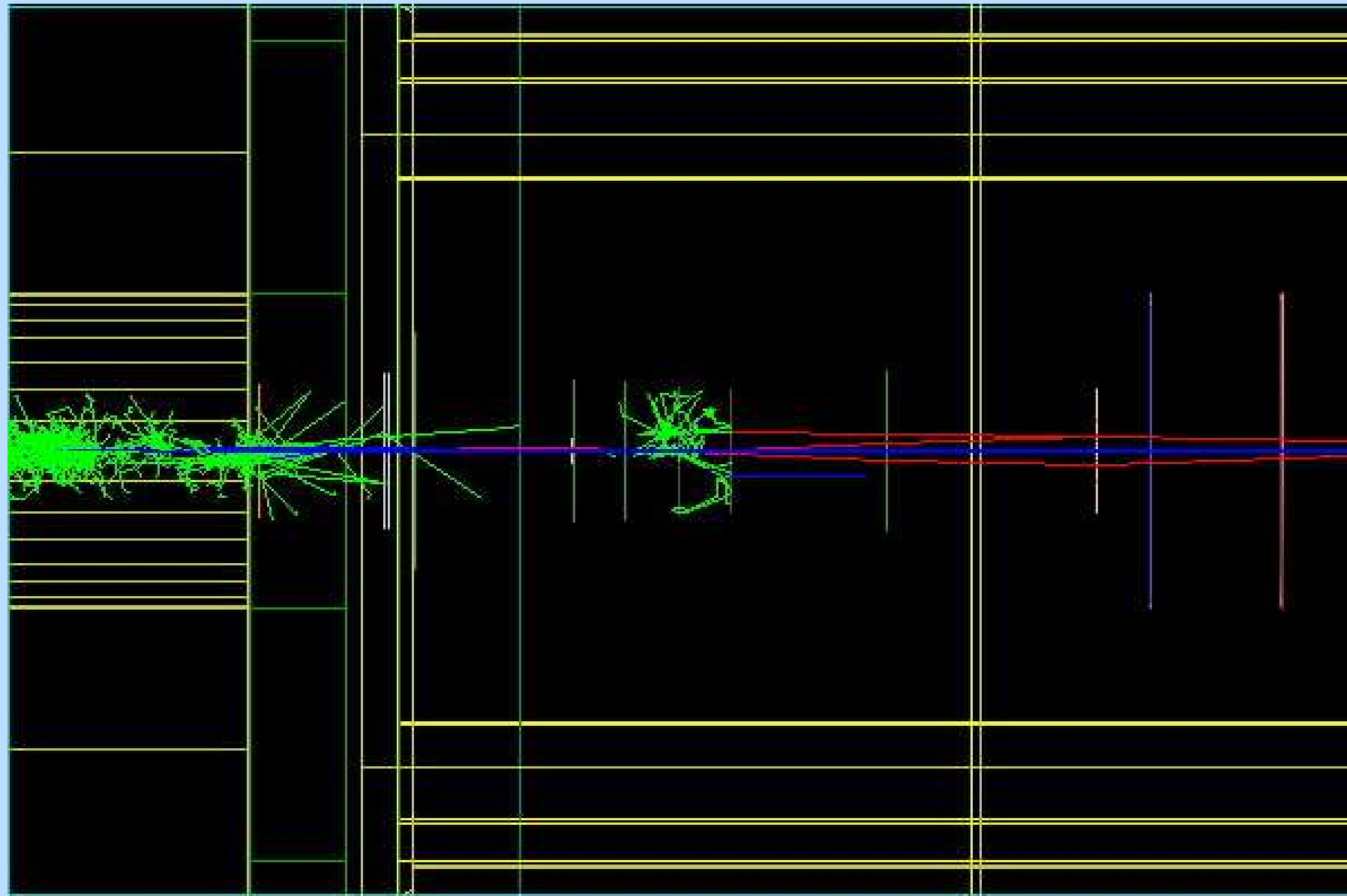
Visualisation Settings

Display some background in the forward region!

Play around with different settings:

- wireframe view (Mokka default)

Visualisation Settings – Example 1



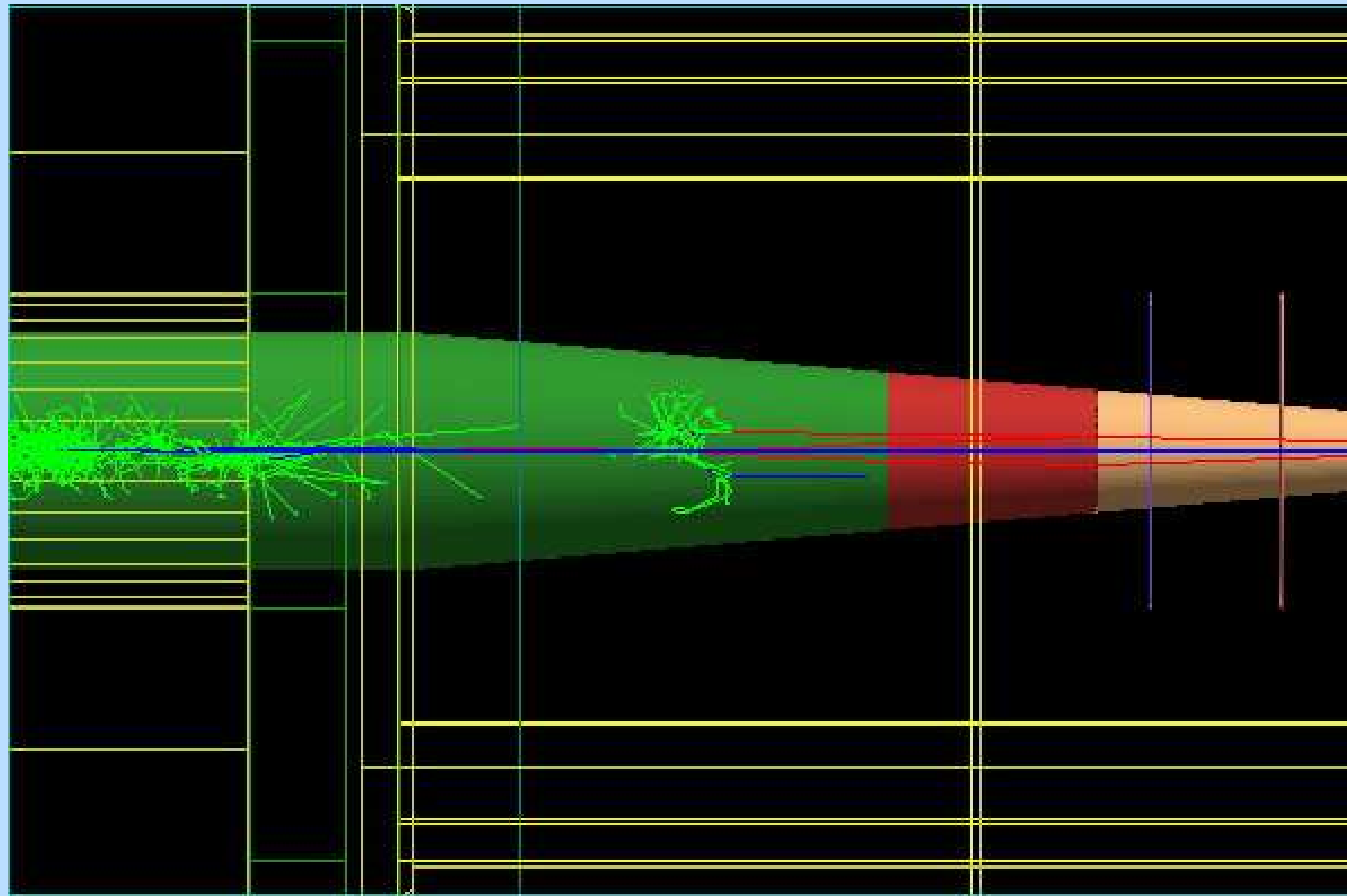
Visualisation Settings

Display some background in the forward region!

Play around with different settings:

- wireframe view (Mokka default)
- surface view (partially overridden)

Visualisation Settings – Example 2



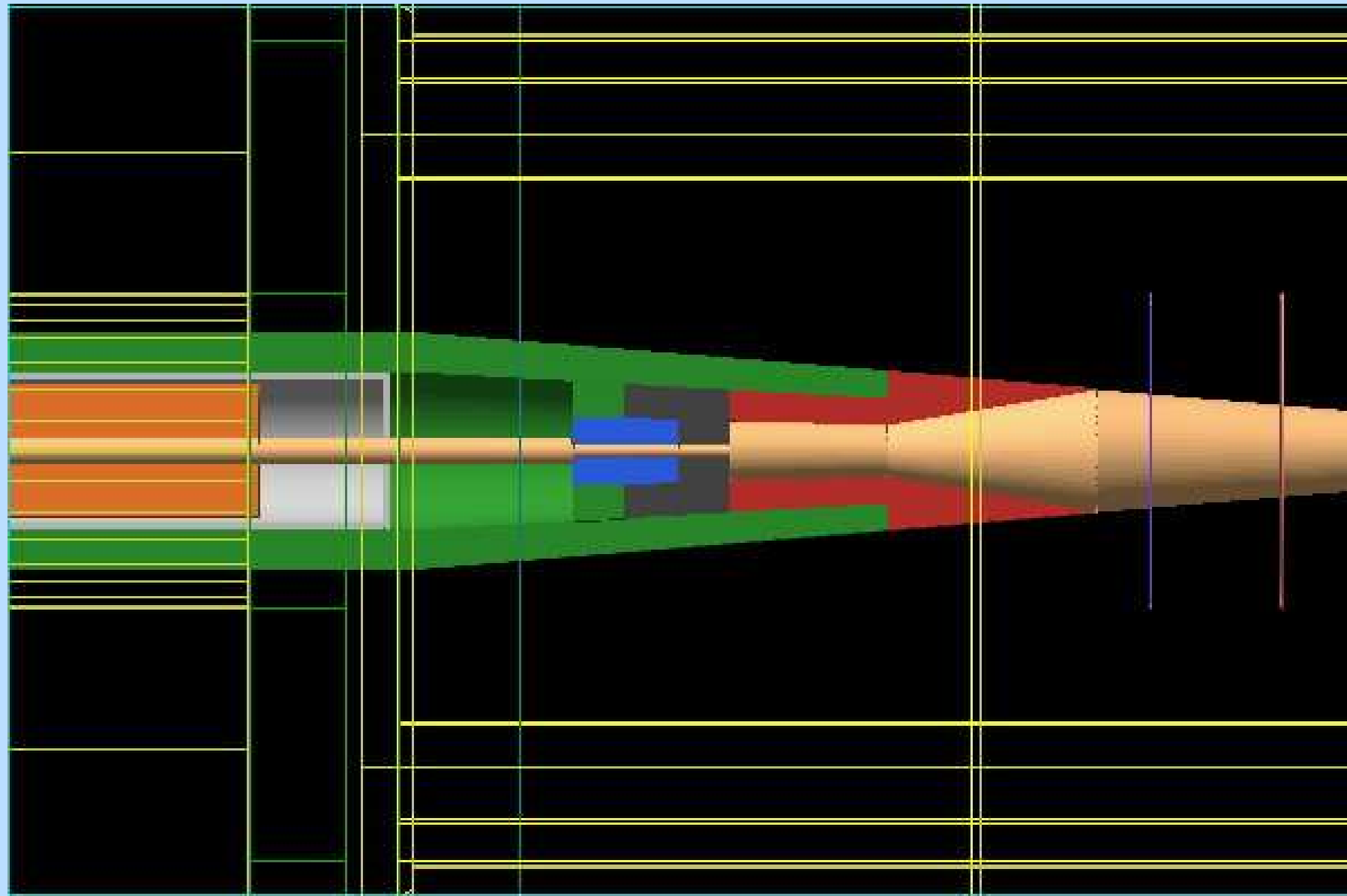
Visualisation Settings

Display some background in the forward region!

Play around with different settings:

- wireframe view (Mokka default)
- surface view (partially overridden)
- construct only one half of the detector

Visualisation Settings – Example 3



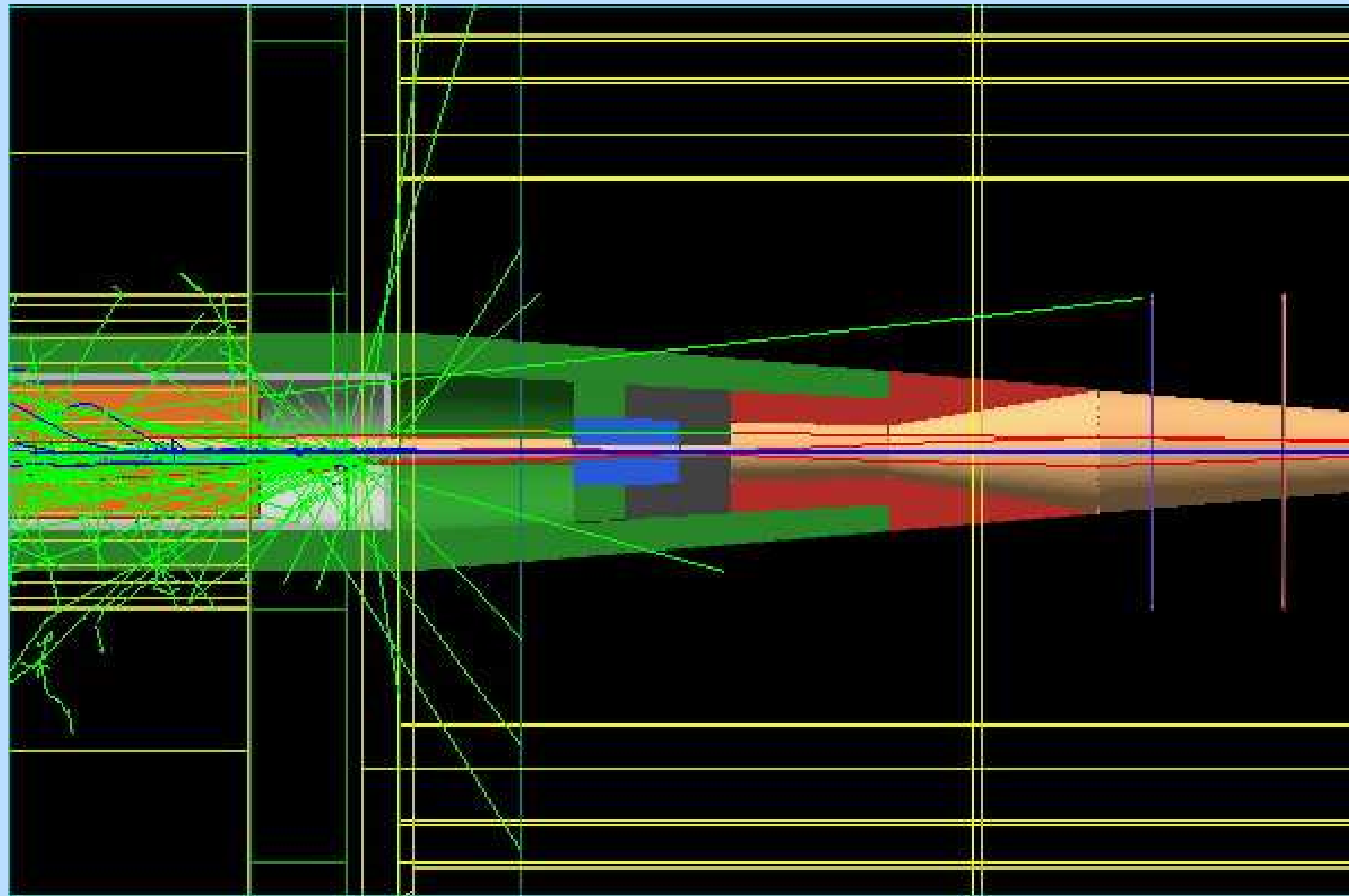
Visualisation Settings

Display some background in the forward region!

Play around with different settings:

- wireframe view (Mokka default)
- surface view (partially overridden)
- construct only one half of the detector
(but this changes the physics behaviour!)

Visualisation Settings – Example 4



Visualisation Settings

Display some background in the forward region!

Play around with different settings:

- wireframe view (Mokka default)
- surface view (partially overridden)
- construct only one half of the detector
- make the other half invisible

Visualisation Settings

Display some background in the forward region!

Play around with different settings:

- wireframe view (Mokka default)
- surface view (partially overridden)
- construct only one half of the detector
- make the other half invisible
(quite tedious, increases the complexity of driver code and simulation geometry)

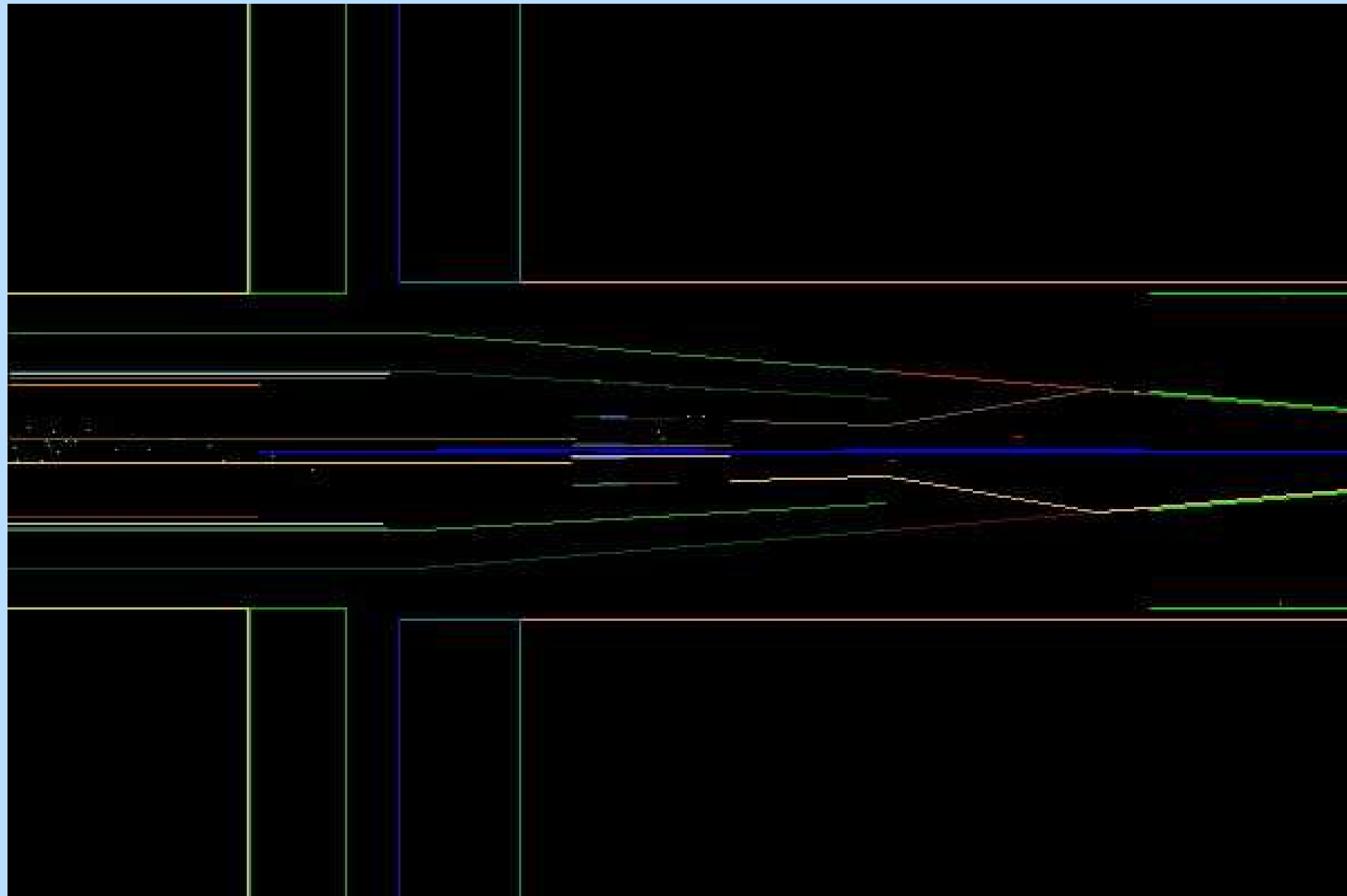
Visualisation Settings

Display some background in the forward region!

Play around with different settings:

- wireframe view (Mokka default)
- surface view (partially overridden)
- construct only one half of the detector
- make the other half invisible
- apply a cross section

Visualisation Settings – Example 5



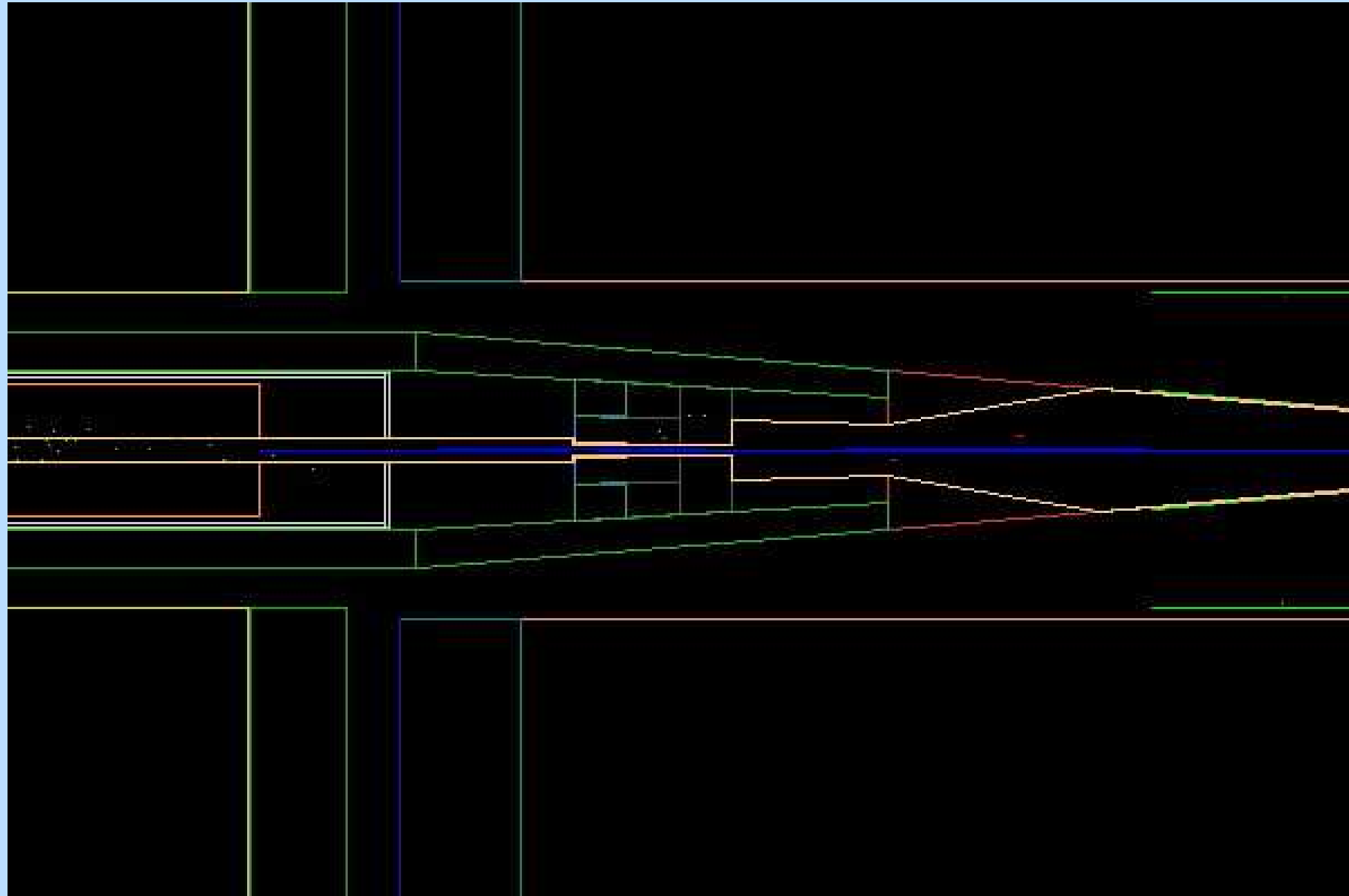
Visualisation Settings

Display some background in the forward region!

Play around with different settings:

- wireframe view (Mokka default)
- surface view (partially overridden)
- construct only one half of the detector
- make the other half invisible
- apply a cross section...
- ... but do not use the surface view!

Visualisation Settings – Example 6



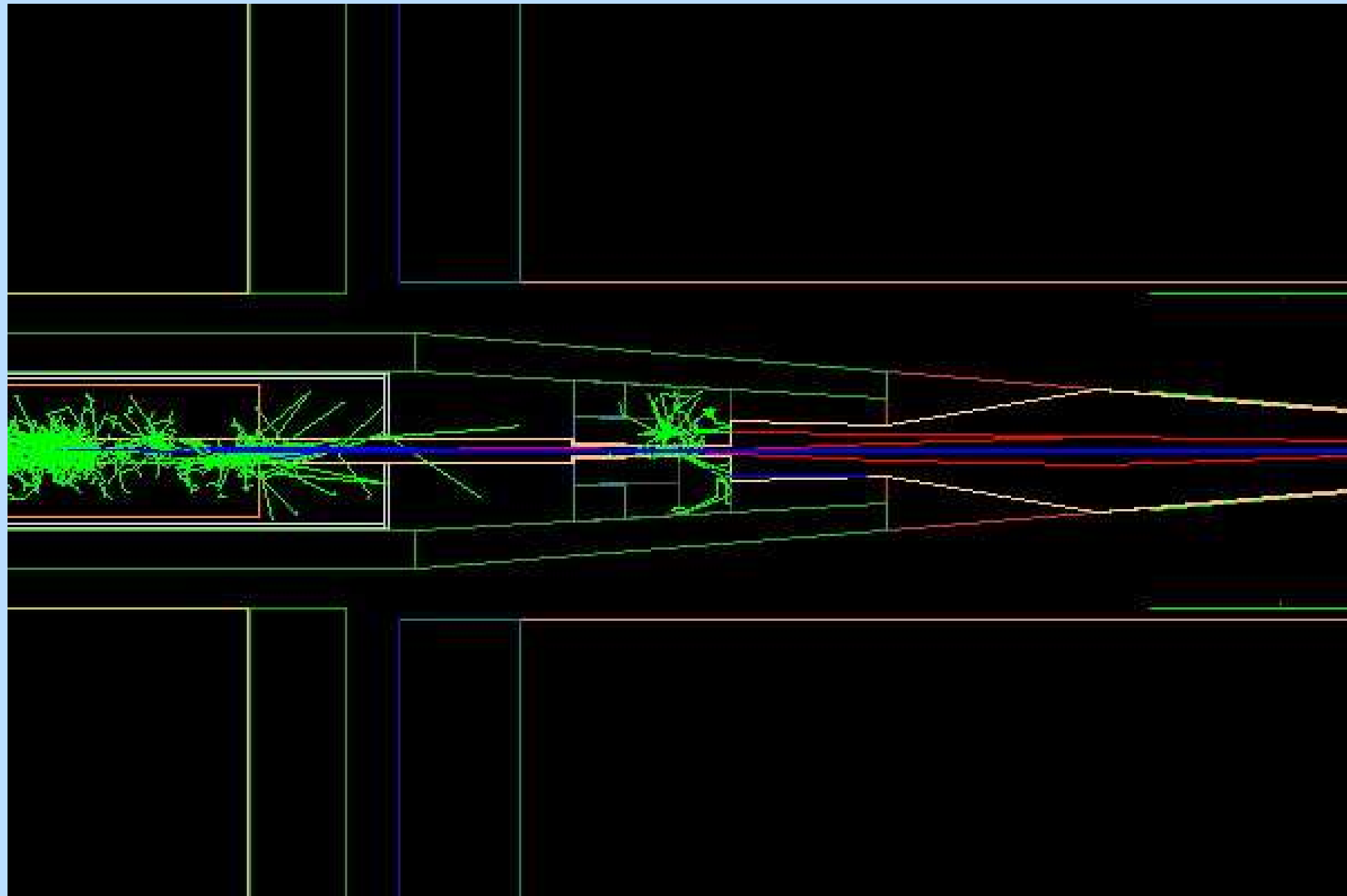
Visualisation Settings

Display some background in the forward region!

Play around with different settings:

- wireframe view (Mokka default)
- surface view (partially overridden)
- construct only one half of the detector
- make the other half invisible
- apply a cross section...
- ... but do not use the surface view!
- project trajectories onto the section plane

Visualisation Settings – Example 7



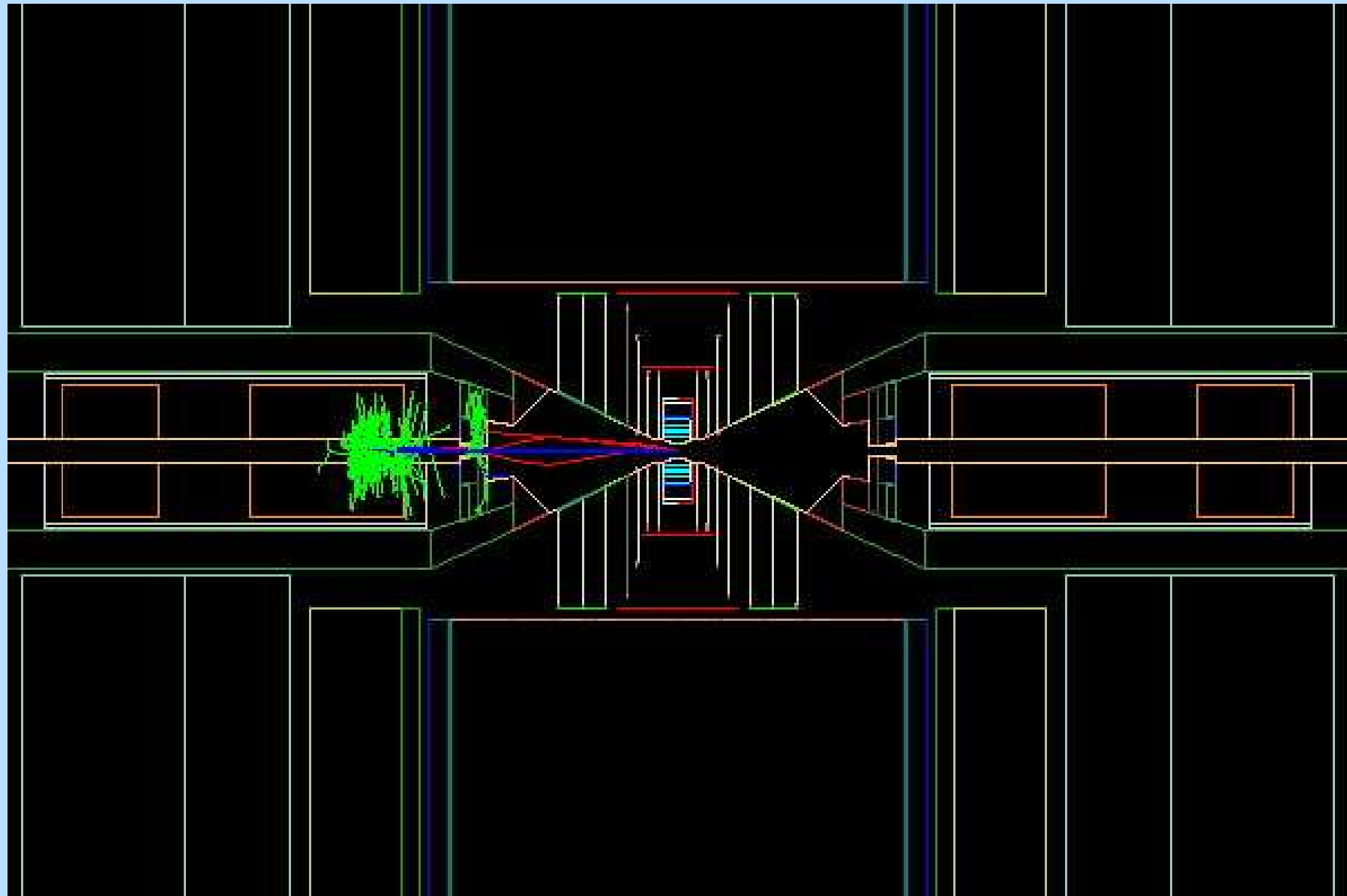
Visualisation Settings

Display some background in the forward region!

Play around with different settings:

- wireframe view (Mokka default)
- surface view (partially overridden)
- construct only one half of the detector
- make the other half invisible
- apply a cross section...
- ... but do not use the surface view!
- project trajectories onto the section plane
- tilt the view to distort the aspect ratio

Visualisation Settings – Example 8



Drawing With a Plugin

Disable automatic drawing:

- `/step/draw 0` (disable on-the-fly drawing of steps)
- `/Mokka/Draw 0` (disable drawing of whole events)

Rewrite some Geant 4 classes with extended features:

- `G4VTrajectoryPoint`: store related volumes
- `G4VTrajectory`: keep a record of steps

Do the drawing from within your plugin:

- `SteppingAction`: store each step point
- `EndOfEventAction`: project each point to $x = 0$ and draw a coloured polyline into the event display

Creating Your Own Plugin

Write the source code:

- inherit from the base class `Plugin`
- implement the virtual methods
- perform whatever actions you like

Include it into Mokka:

- add it to the top-level makefile
- link it into the Mokka executable

Call and configure it in your steering file:

- `/Mokka/init/registerPlugin PluginName`
- `/Mokka/init/userInitType param value`

Pros and Cons

Advantages:

- full control over each step, track, event, run
- drawing is only a side effect!

Disadvantages and dangers:

- you have to know exactly what is happening inside Mokka and in the depths of Geant 4
- multiple plugins may interfere with each other
- one multi-purpose plugin offends the original idea of plugins as independent little helpers

Conclusion

This approach:

- it works – or seems to work, at least
- it is just an ugly hack
- displaying things is the task of the viewer and not of the simulation code
- there's a long way ahead to catch up with the visualisation power of Geant 3

Alternatives:

- HepRep together with the WIRED4 event display
- IGUANA visualisation system (used for CMS)

Bonus Slide – Things to Try

Use an interactive, Motif-based viewer:

- `/vis/open OGLSXm`
- `/vis/viewer/update` (to enter the GUI)

Display multiple events at once:

- `/vis/scene/endOfEventAction accumulate`

Adjust some fancy settings:

- `/vis/viewer/set/style w|s`
- `/vis/viewer/set/sectionPlane on|off`
- `/vis/viewer/set/lineSegmentsPerCircle n`
- `/vis/viewer/set/projection o|p [n]`