Integrating Tracking and Beam-Matter Interaction for Medical Beam Lines

Dresden ENLITE 09

Lucas Clemente

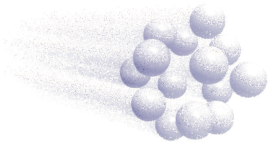
FZD, Dresden

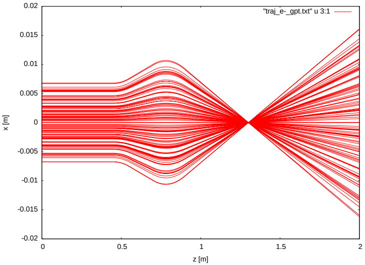
April 3rd 2009

The General Particle Tracer (GPT)

GPT

- "A 3D code for accelerator and beamline design"
- Developed for 12 years (in C)
- Sophisticated tracking algorithm
- Good space-charge and multiple particle tracing
- Big repository of elements accessible





The General Particle Tracer (GPT)

How does it work?

- 5^{th} order Runge-Kutta algorithm
- Stepwise tracking of *multiple* particles through user-defined fields
- User interference possible using "custom elements"
- User may specify field components ($\vec{E}$, $\vec{B}$)
- No random but reproducable trajectories of *real* particles

A Problem

Little to no particle matter interaction possible

The General Particle Tracer (GPT)

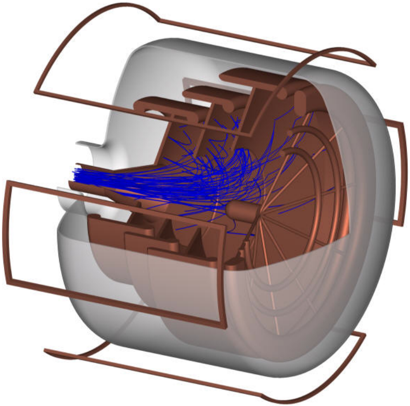
How does it work?

- 5^{th} order Runge-Kutta algorithm
- Stepwise tracking of *multiple* particles through user-defined fields
- User interference possible using "custom elements"
- User may specify field components ($\vec{E}$, $\vec{B}$)
- No random but reproducable trajectories of *real* particles

A Problem

Little to no particle matter interaction possible

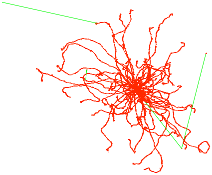
The General Particle Tracer (GPT)

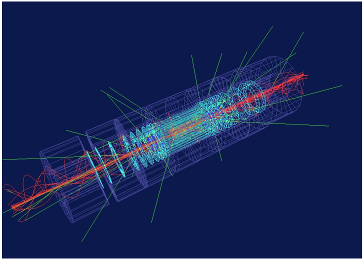


GEANT4

GEometry ANd Tracking 4

- Toolkit for the simulation of particle-matter interactions
- Developed for 15 years at CERN (C++)
- Advanced geometry system
- Proofed correct in many experiments





GEANT4

How does it work?

- Monte-Carlo algorithm
- Stepwise tracking of *single* particles through user-defined geometries
- User can overload nearly any class or supply his own (Open-source)
- Random particle trajectories, statistical simulation of *Monte-Carlo particles*

Some Problems

- Only single-particle simulations possible (no space charge, ...)
- Modest electromagnetic fields

GEANT4

How does it work?

- Monte-Carlo algorithm
- Stepwise tracking of *single* particles through user-defined geometries
- User can overload nearly any class or supply his own (Open-source)
- Random particle trajectories, statistical simulation of *Monte-Carlo particles*

Some Problems

- Only single-particle simulations possible (no space charge, ...)
- Modest electromagnetic fields

Challenges of a beamline modelling software

Medical beamlines

- Compact proton / ion sources require a compact beamline
- Scaling of beamlines not possible
- Radiation protection!

State of the art

There is no code combining advanced tracking and particle-matter interactions!

Challenges of a beamline modelling software

Medical beamlines

- Compact proton / ion sources require a compact beamline
- Scaling of beamlines not possible
- Radiation protection!

State of the art

There is no code combining advanced tracking and particle-matter interactions!

What is Flint?

FZD **L**aser **In**teractor and **T**racker

A simulation software fundamentally combining GPT and GEANT4

Features

- Tracking massive amounts of particles through complex geometries
- All of GPT's space charge algorithms
- Full GEANT4 capabilities in particle-matter interaction
- All geometry elements and fields of GPT and GEANT4
- Customizability
- High-Speed due to parallelization using OpenMP

What is Flint?

FZD Laser Interactor and Tracker

A simulation software fundamentally combining GPT and GEANT4

Features

- Tracking massive amounts of particles through complex geometries
- All of GPT's space charge algorithms
- Full GEANT4 capabilities in particle-matter interaction
- All geometry elements and fields of GPT and GEANT4
- Customizability
- High-Speed due to parallelization using OpenMP

A simple calculation using GPT...

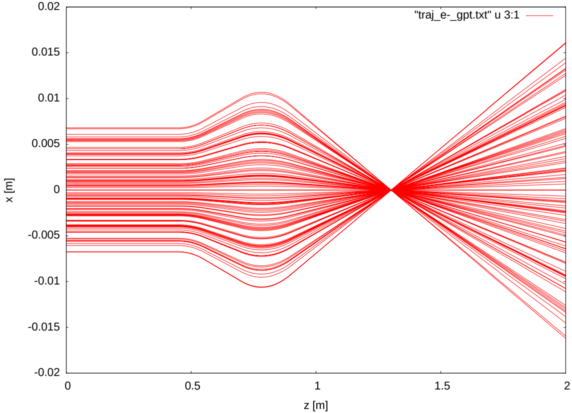


Figure: Electron beam ($\gamma = 100$) passing through two quadrupoles ($G = 3.9, -3.25\ \frac{T}{m}$)

... and Flint

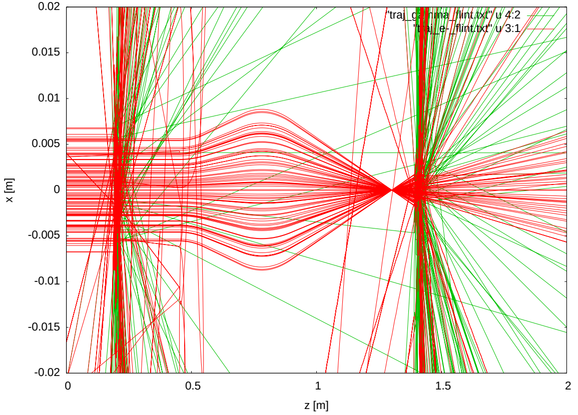


Figure: Electron beam ($\gamma = 100$) passing through two quadrupoles ($G = 3.9, -3.25\ \frac{T}{m}$) and two pinholes ($Cu,\ dz = 2\ cm$)

Architecture

How is it built?

- Problem: GPT only limited extendable
  $\Rightarrow$ Custom field elements
- GPT is C, GEANT4 is C++
- Solution: Link a GEANT4 based library
- Other used libraries:
  - xerces for XML-configuration
  - CLHEP
  - OpenMP (included in gcc4)

What is flint?

Flint consists of a GEANT4 based library to be linked into GPT, a GPT custom element and a bunch of tools for modelling, run and analysis.

Architecture

How is it built?

- Problem: GPT only limited extendable
  $\Rightarrow$ Custom field elements
- GPT is C, GEANT4 is C++
- Solution: Link a GEANT4 based library
- Other used libraries:
  - xerces for XML-configuration
  - CLHEP
  - OpenMP (included in gcc4)

What is flint?

Flint consists of a GEANT4 based library to be linked into GPT, a GPT custom element and a bunch of tools for modelling, run and analysis.

Architecture

The element "G4Virtual"

Include GEANT4 just by typing

```
G4Virtual("WCS", "I");
```

Program flow

- Calculate normal GPT step
- After each succesful step do for each particle:
  - Check for intersection with a GEANT4 solid (pre and post step point)
  - Use GEANT4 for recalculation of the step
  - Inject the results of GEANT4 into GPT
  - Output the particle's data (e.g. trajectories, dose)

Architecture

The element “G4Virtual”

Include GEANT4 just by typing

```
G4Virtual("WCS", "I");
```

Program flow

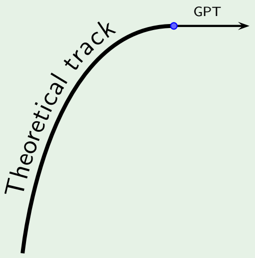
- Calculate normal GPT step
- After each succesful step do for each particle:
  - Check for intersection with a GEANT4 solid (pre and post step point)
  - Use GEANT4 for recalculation of the step
  - Inject the results of GEANT4 into GPT
  - Output the particle's data (e.g. trajectories, dose)

Combined Stepping

A step in Flint

1. Use GPT to track the step
2. Recalculate the step using GEANT4
   - Start at pre-GPT-point
   - Track in direction of post-GPT point
   - Stop when time exceeds the GPT time
3. Inject the results of GEANT4 into GPT
   - Take the new position and velocities
   - Calculate the rotation of momentum between pre-GEANT4 and post-GEANT4
   - Rotate the post-GPT momentum accordingly

An example

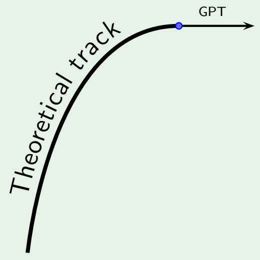


Combined Stepping

A step in Flint

1. Use GPT to track the step
2. Recalculate the step using GEANT4
   - Start at pre-GPT-point
   - Track in direction of post-GPT point
   - Stop when time exceeds the GPT time
3. Inject the results of GEANT4 into GPT
   - Take the new position and velocities
   - Calculate the rotation of momentum between pre-GEANT4 and post-GEANT4
   - Rotate the post-GPT momentum accordingly

An example

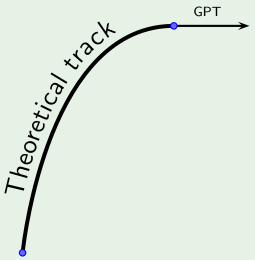


Combined Stepping

A step in Flint

1. Use GPT to track the step
2. Recalculate the step using GEANT4
   - Start at pre-GPT-point
   - Track in direction of post-GPT point
   - Stop when time exceeds the GPT time
3. Inject the results of GEANT4 into GPT
   - Take the new position and velocities
   - Calculate the rotation of momentum between pre-GEANT4 and post-GEANT4
   - Rotate the post-GPT momentum accordingly

An example

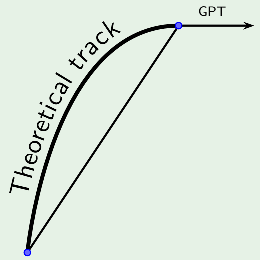


Combined Stepping

A step in Flint

1. Use GPT to track the step
2. Recalculate the step using GEANT4
   - Start at pre-GPT-point
   - Track in direction of post-GPT point
   - Stop when time exceeds the GPT time
3. Inject the results of GEANT4 into GPT
   - Take the new position and velocities
   - Calculate the rotation of momentum between pre-GEANT4 and post-GEANT4
   - Rotate the post-GPT momentum accordingly

An example

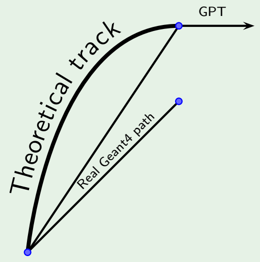


Combined Stepping

A step in Flint

1. Use GPT to track the step
2. Recalculate the step using GEANT4
   - Start at pre-GPT-point
   - Track in direction of post-GPT point
   - Stop when time exceeds the GPT time
3. Inject the results of GEANT4 into GPT
   - Take the new position and velocities
   - Calculate the rotation of momentum between pre-GEANT4 and post-GEANT4
   - Rotate the post-GPT momentum accordingly

An example

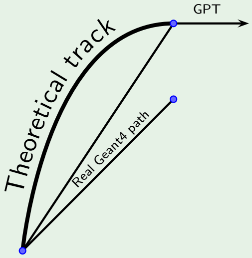


Combined Stepping

A step in Flint

1. Use GPT to track the step
2. Recalculate the step using GEANT4
   - Start at pre-GPT-point
   - Track in direction of post-GPT point
   - Stop when time exceeds the GPT time
3. Inject the results of GEANT4 into GPT
   - Take the new position and velocities
   - Calculate the rotation of momentum between pre-GEANT4 and post-GEANT4
   - Rotate the post-GPT momentum accordingly

An example

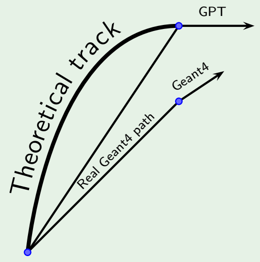


Combined Stepping

A step in Flint

1. Use GPT to track the step
2. Recalculate the step using GEANT4
   - Start at pre-GPT-point
   - Track in direction of post-GPT point
   - Stop when time exceeds the GPT time
3. Inject the results of GEANT4 into GPT
   - Take the new position and velocities
   - Calculate the rotation of momentum between pre-GEANT4 and post-GEANT4
   - Rotate the post-GPT momentum accordingly

An example

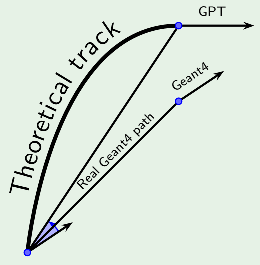


Combined Stepping

A step in Flint

1. Use GPT to track the step
2. Recalculate the step using GEANT4
   - Start at pre-GPT-point
   - Track in direction of post-GPT point
   - Stop when time exceeds the GPT time
3. Inject the results of GEANT4 into GPT
   - Take the new position and velocities
   - Calculate the rotation of momentum between pre-GEANT4 and post-GEANT4
   - Rotate the post-GPT momentum accordingly

An example

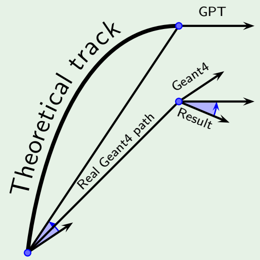


Combined Stepping

A step in Flint

1. Use GPT to track the step
2. Recalculate the step using GEANT4
   - Start at pre-GPT-point
   - Track in direction of post-GPT point
   - Stop when time exceeds the GPT time
3. Inject the results of GEANT4 into GPT
   - Take the new position and velocities
   - Calculate the rotation of momentum between pre-GEANT4 and post-GEANT4
   - Rotate the post-GPT momentum accordingly

An example



Secondaries

Secondary particles

- GEANT4 produces all kind of secondaries
- GPT can only handle massive and charged particles

The solution

- Massive and charged particles are injected into GPT
- Massless or chargeless particles are tracked *and outputted* solely by GEANT4

Secondaries

Secondary particles

- GEANT4 produces all kind of secondaries
- GPT can only handle massive and charged particles

The solution

- Massive and charged particles are injected into GPT
- Massless or chargeless particles are tracked *and outputted* solely by GEANT4

Design of beamlines

Input

Only one input file (XML) consisting of:

- GDML part (**G**eometry **D**escription **M**arkup **L**anguage, supported naturally by GEANT4)
- GPT part (In general GPT `ini` file)
- Some additional tags controlling Flint

Example: Configuring particles used in GEANT4

```
<particles>
  <particle name="gamma" />
  <particle name="e-" />
  <particle name="proton" />
</particles>
```

Design of beamlines

Input

Only one input file (XML) consisting of:

- GDML part (**G**eometry **D**escription **M**arkup **L**anguage, supported naturally by GEANT4)
- GPT part (In general GPT `ini` file)
- Some additional tags controlling Flint

Example: Configuring particles used in GEANT4

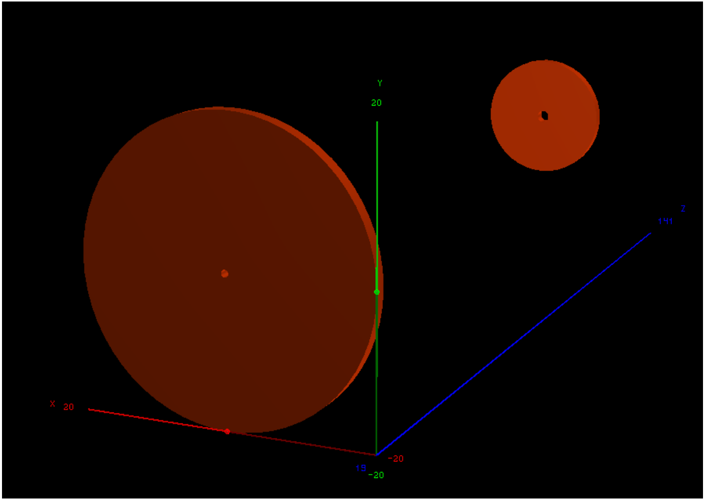
```
<particles>
  <particle name="gamma" />
  <particle name="e-" />
  <particle name="proton" />
</particles>
```

Design of beamlines

Example: A pinhole

```
<tube name="pinhole" z="20.0" rmin="10.0" rmax="100.0" />
<volume name="pinhole">
  <materialref ref="Cu" />
  <solidref ref="pinhole" />
</volume>
<position name="ph" z="1400.0" />
<physvol>
  <volumeref ref="pinhole" />
  <positionref ref="ph" />
</physvol>
```

Design of beamlines



Design of beamlines

Example: Two quadrupoles

```
<gpt>
  radius = 6e-3;
  setparticles("beam", 100, me, qe, 0.0);
  setrxydist("beam", "u", radius/2, radius);
  setphidist("beam", "u", 0, 2*pi);
  setGdist("beam", "u", 100, 0);
  quadrupole("wcs", "z", 0.2, 0.1, 3.90);
  quadrupole("wcs", "z", 0.5, 0.2, -3.25);
  G4Virtual("wcs", "i");
  tout(0, 4e-9, 0.05e-9);
</gpt>
```

Output

Current outputs availible

- GPT outputs (e.g. Divergence, Emittance, Standard deviations, ...)
- Trajectories (seperated by particle type, massless only here) as ASCII
- Dose and energy deposition in the geometry as ASCII

Configuration of outputs

```
<outputs>
  <output name="trajectories" />
  <output name="dose" x_min="-1e3" x_max="1e3"
          y_min="-1e3" y_max="1e3" z_min="0" z_max="2e3"
          nx="1000" ny="1000" nz="2000" />
</outputs>
```

Output

Current outputs availible

- GPT outputs (e.g. Divergence, Emittance, Standard deviations, ...)
- Trajectories (seperated by particle type, massless only here) as ASCII
- Dose and energy deposition in the geometry as ASCII

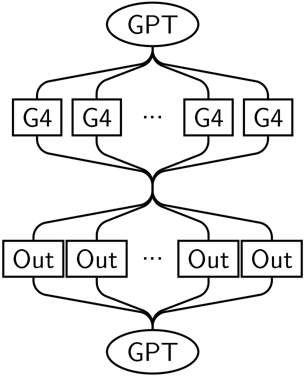
Configuration of outputs

```
<outputs>
  <output name="trajectories" />
  <output name="dose" x_min="-1e3" x_max="1e3"
          y_min="-1e3" y_max="1e3" z_min="0" z_max="2e3"
          nx="1000" ny="1000" nz="2000" />
</outputs>
```

Optimization

Parallization of the GEANT4 part

- Single particles are independant
- GEANT4 and output distributed over several threads using OpenMP (previous versions also MPI)
- Configurable with environment variables



Other optimizations

There are additional optimizations, e.g. for output, particle exchange, ...

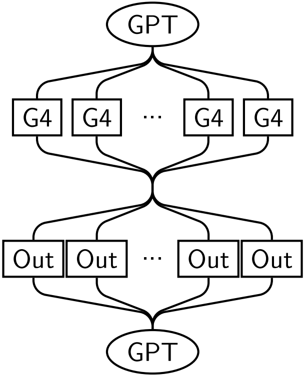
Optimization

Parallization of the GEANT4 part

- Single particles are independant
- GEANT4 and output distributed over several threads using OpenMP (previous versions also MPI)
- Configurable with environment variables

Other optimizations

There are additional optimizations, e.g. for output, particle exchange, ...



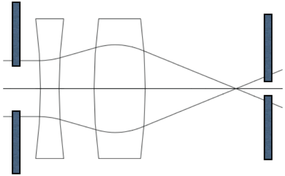
Comparison to pure GPT

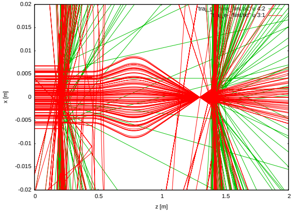
The quadrupole example

- Basically the same world as in a GPT example: Two quadrupoles ($G = 3.9, -3.25\ \frac{T}{m}$) focus a beam of electrons ($\gamma = 100$)
- Empty space filled with weak interacting gas (H_2), two pinholes (Cu)
- Possibility to test the stepping algorithm

The result

Flint and GPT both focus at $z = 1.3\ m$

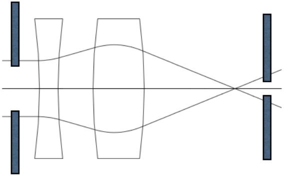


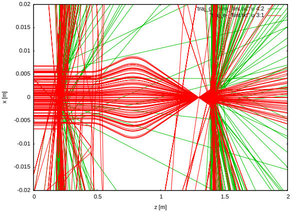


Comparison to pure GPT

The quadrupole example

- Basically the same world as in a GPT example: Two quadrupoles ($G = 3.9, -3.25\ \frac{T}{m}$) focus a beam of electrons ($\gamma = 100$)
- Empty space filled with weak interacting gas (H_2), two pinholes (Cu)
- Possibility to test the stepping algorithm

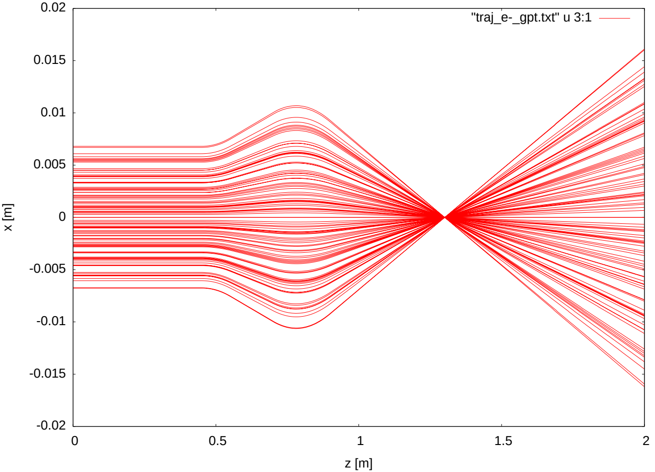




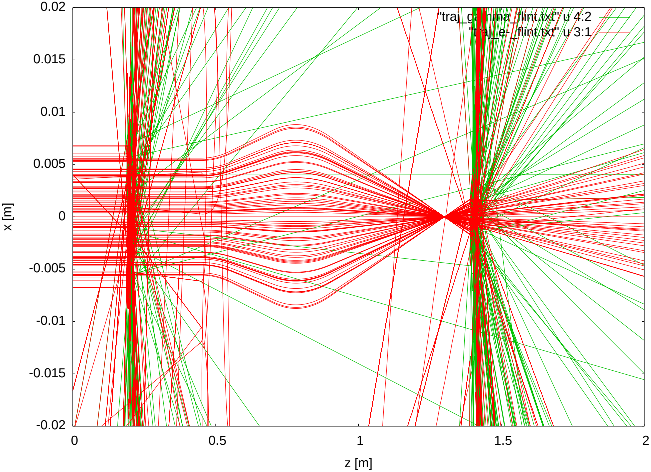
The result

Flint and GPT both focus at $z = 1.3\ m$

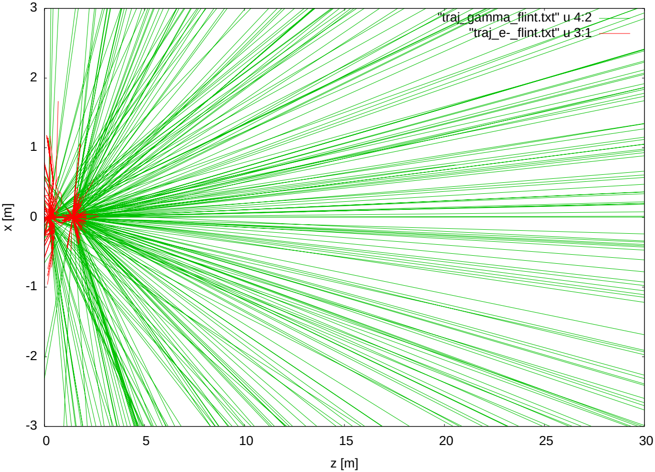
Comparison to pure GPT



Comparison to pure GPT



Comparison to pure GPT



Comparison to pure GEANT4

Emittance calculation

- Electron-beam with a Gaussian space-distribution with $\sigma = 0.3$ mm, $\varepsilon_0 = 1$ mm mrad, divergence of 190 μrad, $\gamma = 440$
- Emittance after a thin ($d = (1, 4, 15)$ μm) aluminium foil is compared to GEANT4

The result

Both, Flint and GEANT4, produce the same change in emittance (for 15 μm Al): $\varepsilon_0 = 81$ mm mrad.

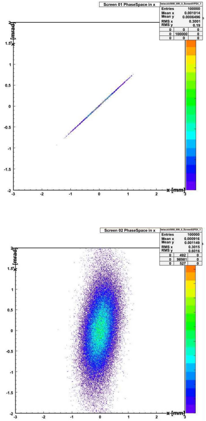


Figure: Phase space

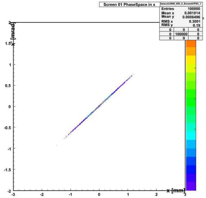
Comparison to pure GEANT4

Emittance calculation

- Electron-beam with a Gaussian space-distribution with $\sigma = 0.3$ mm, $\varepsilon_0 = 1$ mm mrad, divergence of 190 μrad, $\gamma = 440$
- Emittance after a thin ($d = (1, 4, 15)$ μm) aluminium foil is compared to GEANT4

The result

Both, Flint and GEANT4, produce the same change in emittance (for 15 μm Al): $\varepsilon_0 = 81$ mm mrad.



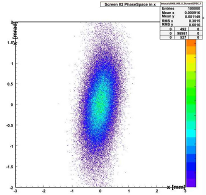
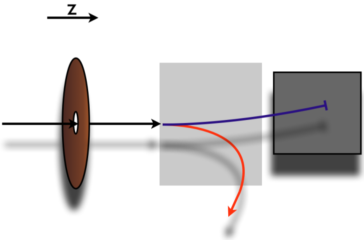


Figure: Phase space

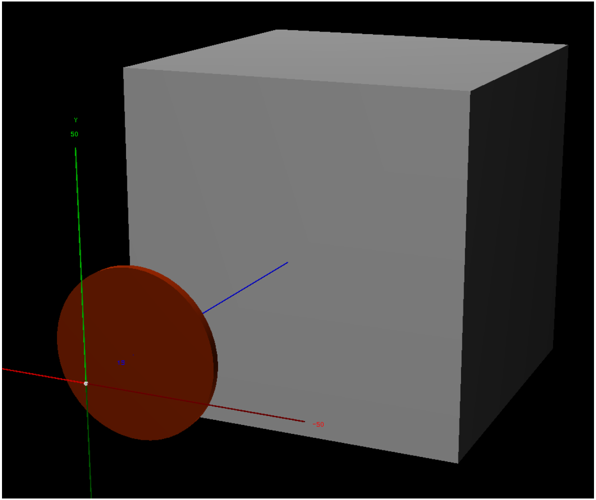
Design of a sample beamline

Designing a sample beamline

- Beam: electrons ($\gamma = 100$) and protons ($\gamma = 5$)
- A pinhole (Cu)
- A separating magnetic field ($B = 2\ T$)
- Protons collide into a beam dump (Pb)



Design of a sample beamline



Design of a sample beamline

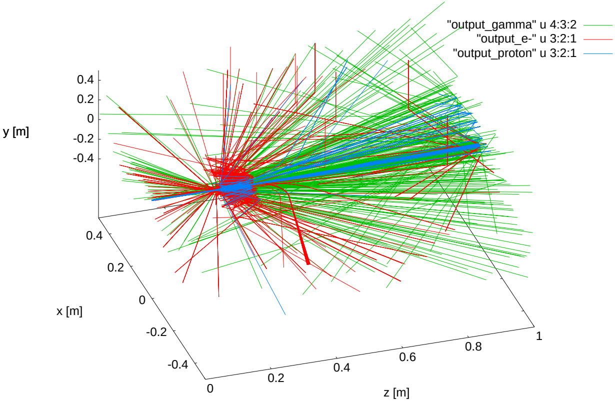
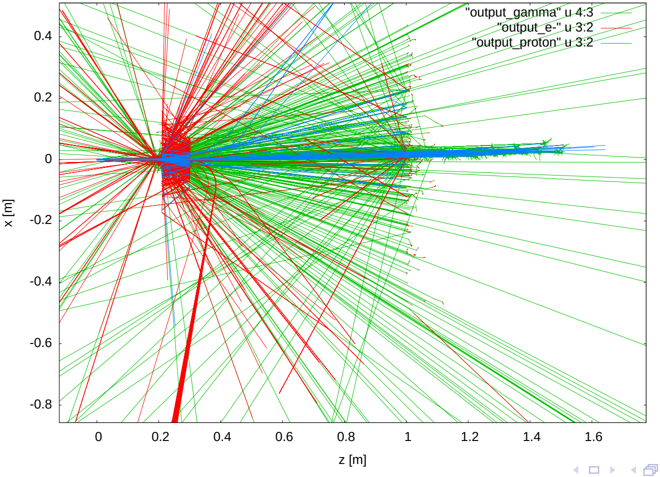
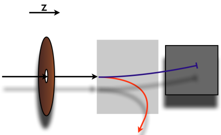


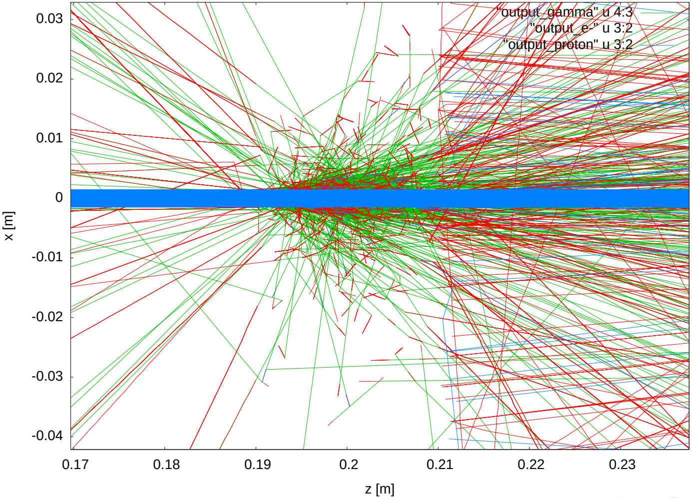
Figure: Also produced: α, e^+, neutrons

Design of a sample beamline

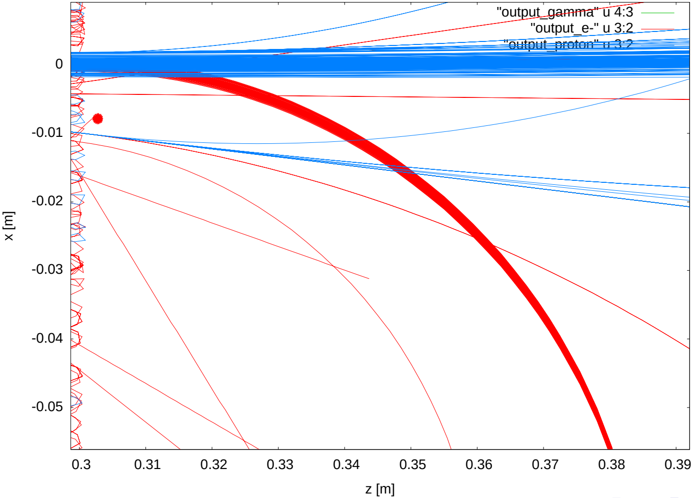




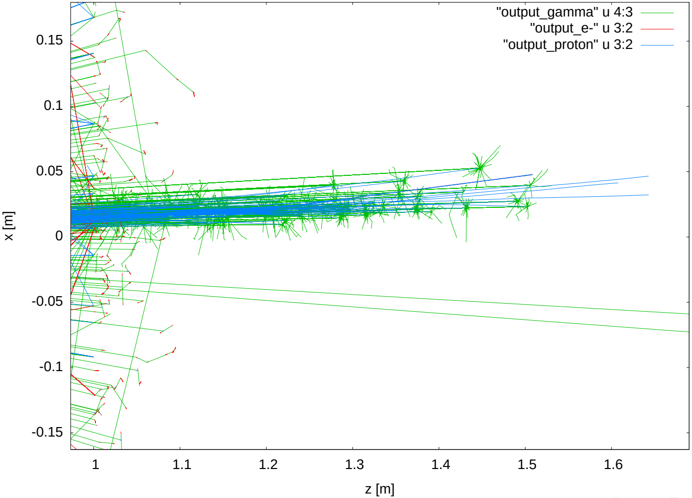
Design of a sample beamline



Design of a sample beamline



Design of a sample beamline



Outlook

Software impropvements

- Some minor and major optimizations
- More configurability through XML (perhaps replace GPT `ini` file)

Major improvements

- Use GEANT4's proposed step length in GPT
- More output options (e.g. detailed interaction information)
- Use another format than ASCII for output (speedup)
- More testing (e.g. tracking ions, fission processes, complex space charge effects)

Outlook

Software impropvements

- Some minor and major optimizations
- More configurability through XML (perhaps replace GPT `ini` file)

Major improvements

- Use GEANT4's proposed step length in GPT
- More output options (e.g. detailed interaction information)
- Use another format than ASCII for output (speedup)
- More testing (e.g. tracking ions, fission processes, complex space charge effects)