Chicane simulation update

Pavel Snopok

Front end phone meeting

April 1, 2014

Chicane field map

- In order to speed up G4beamline simulation it was proposed to use a pre-calculated field map based on the analytic expressions for bent solenoids.
- There are minor discontinuities at the ends of the chicane and at the center, but those are expected.
- Original script is in Python, it takes ~7 seconds to save a 32 Mb field map with grid sizes $\Delta x=\Delta z=10$ mm.

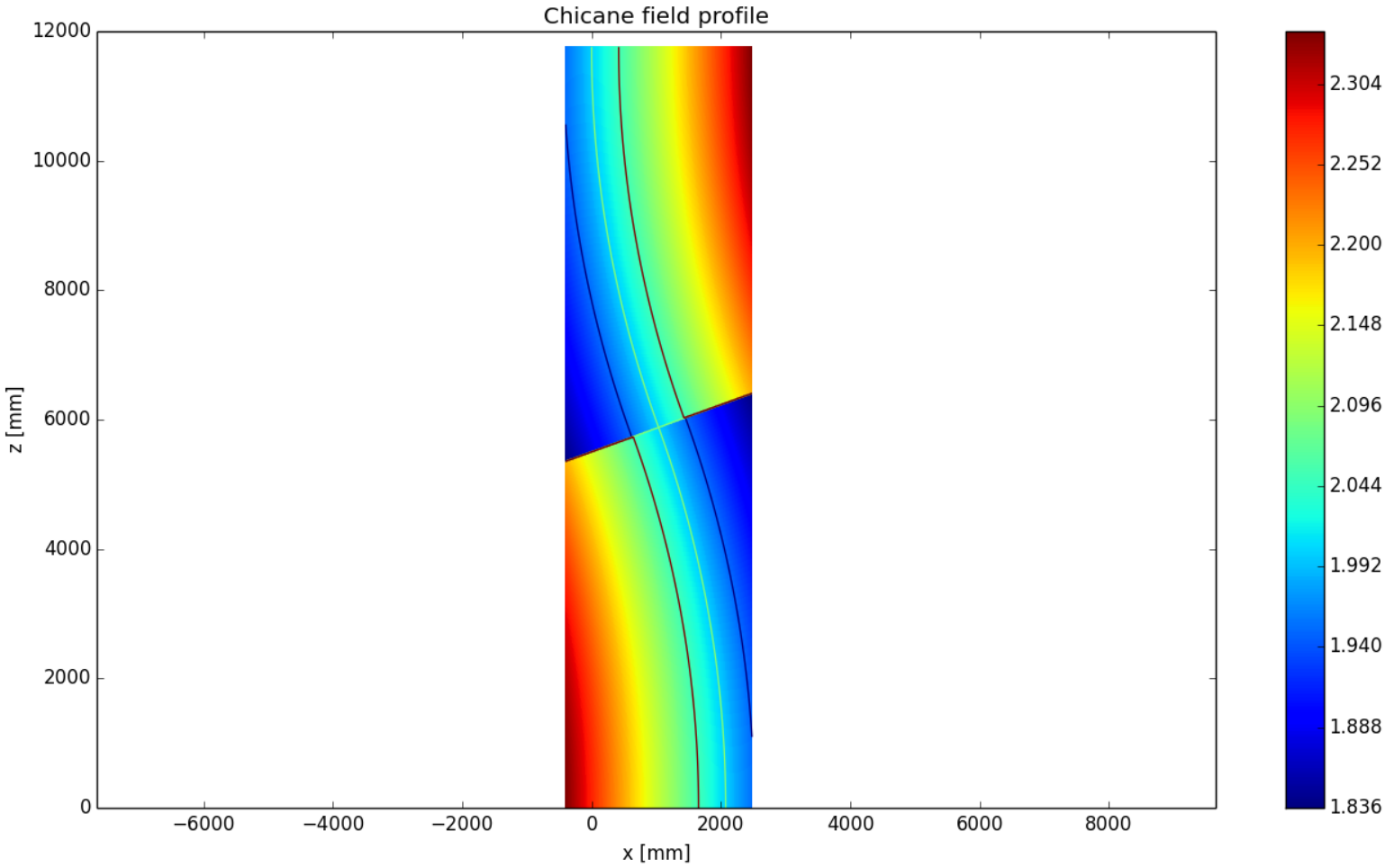
Scripting for NERSC

- I created and tested a bit of C++ code that does the same the Python script does.
- Depends on stdio, stdlib, math, hope those are available.
- Benchmark: ~2 sec for the same size field map.
- Requires six parameters: filename, field on axis [T], chicane curvature [1/m], chicane half-length [m], aperture [m], x grid size [m], z grid size [m].
- Meaningful set of defaults provided.
- Will need some help/feedback getting it up and running on NERSC.

Look ma, no coils

- I kept coils in G4beamline but set their current to zero.
- It was suggested that I should remove the coils altogether to see if that reduces simulation time.
- So I did, and magically the time to run the same number of particles went down sharply from 540 seconds (with coils and currents) to 43 seconds (with coils but no current) to 13 seconds (with no coils) for 50k particles (+ 2 seconds to calculate the field map prior to running G4beamline).

Analytic expression contour



Lines correspond to +/- 0.05 T

Buncher-Rotator-Cooler

- There is still a mystery there.
- Alexey ran the ICOOL distribution at rotator through G4beamline:

@ rotator	eps_perp	eps_long	eps_6D	n_0	n_1	n_2
ICOOL	15.85	121.2	30.39	1703	61.26	178.1
G4beamline, phase 0	15.49	133.7	31.99	1606	61.26	173.0
G4beamline, phase -1 deg	15.59	132.2	32.0	1721	58.73	177.5

- And the same results if starting at buncher:

@ rotator	eps_perp	eps_long	eps_6D	n_0	n_1	n_2
ICOOL	15.85	121.4	30.46	1702	59.99	176.2
G4beamlline	15.91	123.2	31.13	1599	56.20	156.0