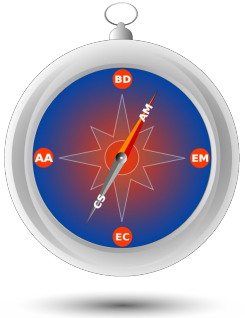


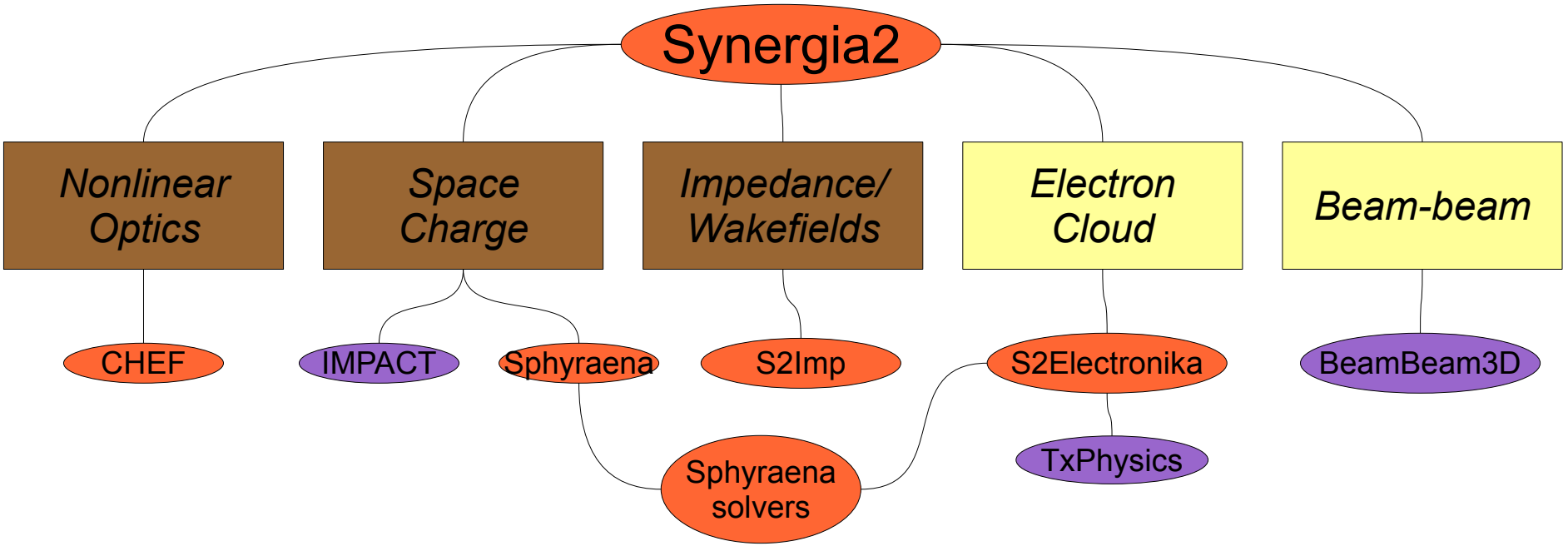
Progress in Synergia2 Development

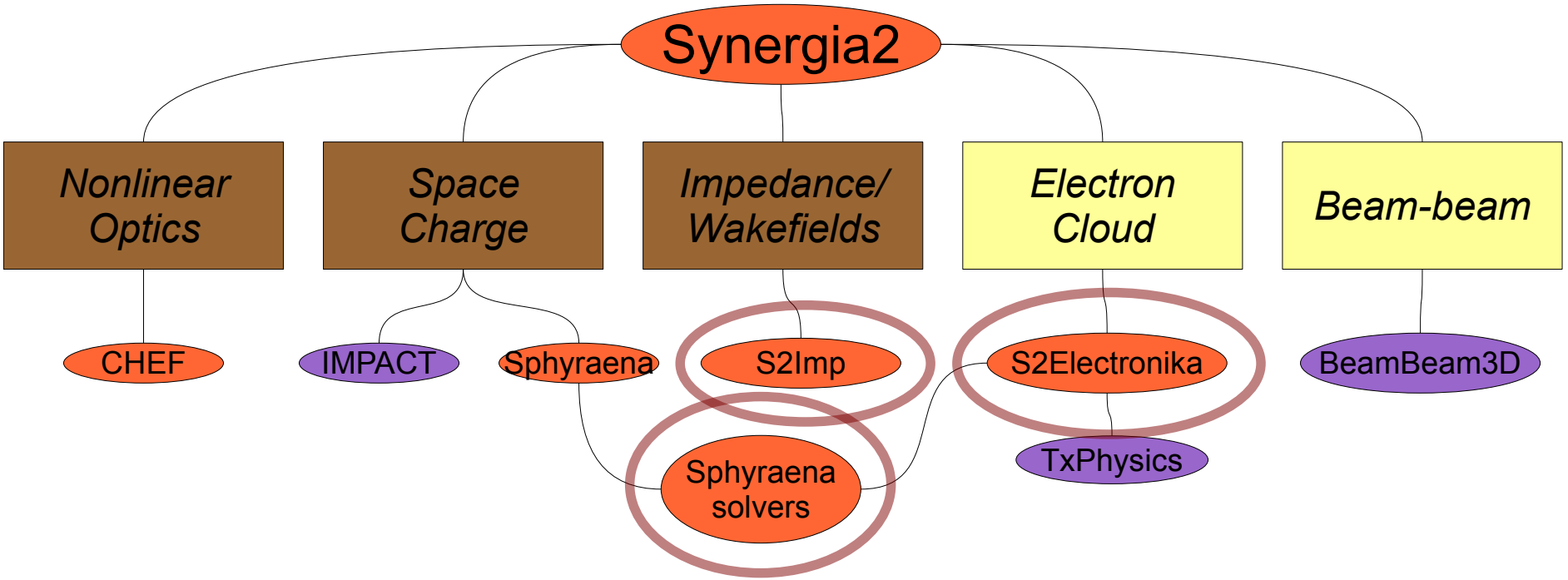
James Amundson
Fermilab

work done in collaboration with
Panagiotis Spentzouris,
Eric Stern and Paul Lebrun
Fermilab

Overview

- Solver development
  - FFT issues
  - Testing
  - Elliptical pipe
- Capability development
  - Resistive wall
  - Electron cloud
- Infrastructure
  - Refactoring
- Porting
    - IBM BlueGene and Cray XT4


Synergia2
Nonlinear Optics
Space Charge
Impedance/ Wakefields
Electron Cloud
Beam-beam
CHEF
IMPACT
Sphyraena
S2Imp
S2Electronika
BeamBeam3D
TxPhysics
Sphyraena solvers

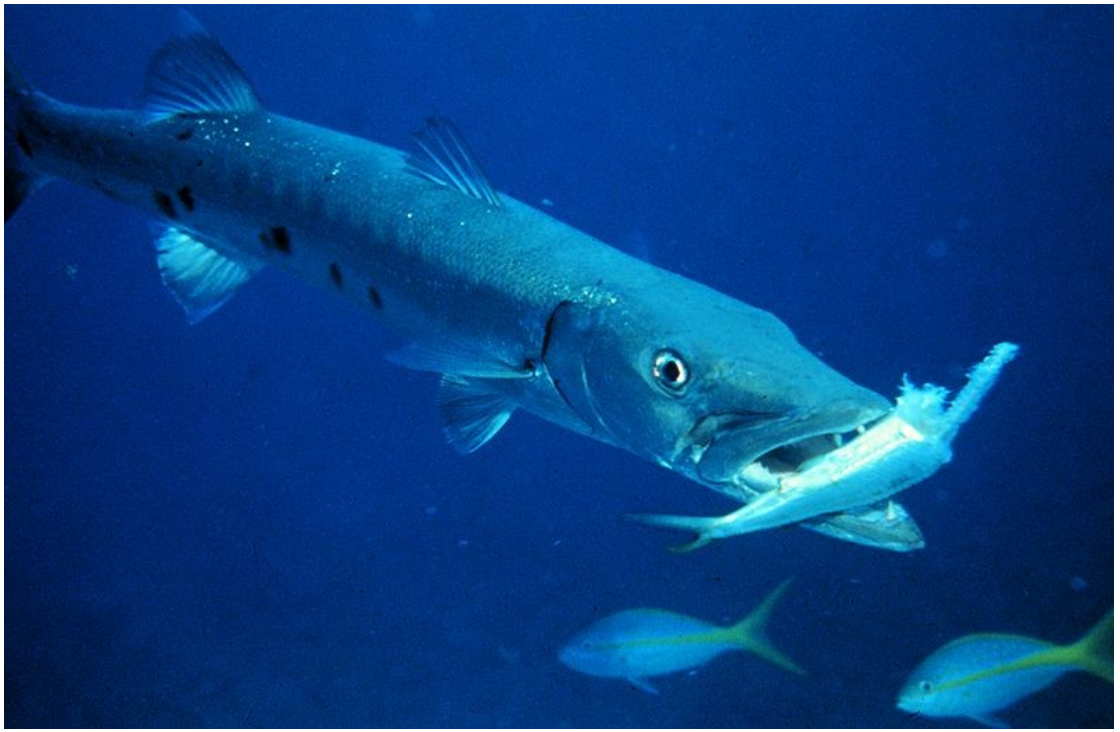

Synergia2
Nonlinear Optics
Space Charge
Impedance/ Wakefields
Electron Cloud
Beam-beam
CHEF
IMPACT
Sphyraena
S2Imp
S2Electronika
BeamBeam3D
Sphyraena solvers
TxPhysics

Sphyraena solver suite

Barracuda

From Wikipedia, the free encyclopedia

The barracuda is a ray-finned fish known for its large size (up to 6 ft in length and up to a foot in width, for some species[2]) and fearsome appearance. Its body is long, fairly compressed, and covered with small, smooth scales. It is found in tropical and subtropical oceans worldwide. It is of the genus Sphyraena, the only genus in the family Sphyraenidae.



Sphyraena, a suite of fast Poisson solvers

Sphyraena solvers

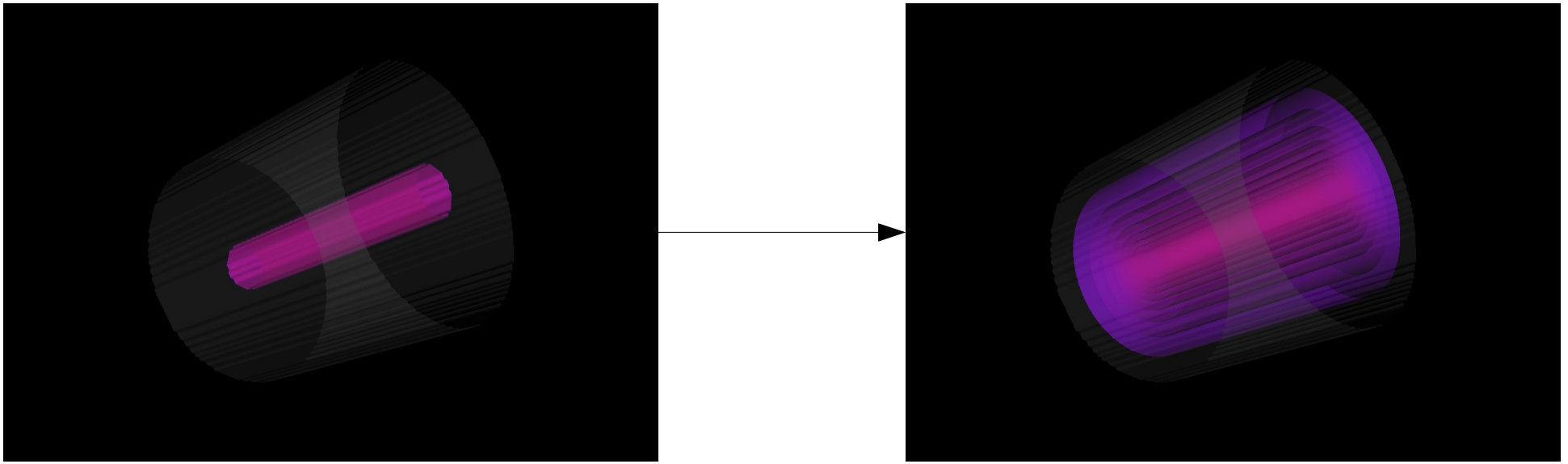
- 3D Open BC's
  - FFT + Green Functions a la Hockney
    - Similar to IMPACT, but with different parallel decomposition strategy
- 3D Cylindrical Closed
  - FFT (z,theta), finite difference in r
    - Likewise
- 3D Elliptical
  - Finite differences, stretched grid
    - PETSc

Sphyraena FFT issues

- Utilize FFTW
  - Take advantage of all FFTW optimizations, including real-to-complex/complex-to-real
- Parallel (MPI) FFTW
  - V2: OK
  - V3: 3.2alpha only, now 3.3alpha!
    - 3.2 was alpha for a long time (> 1 yr). Finally, 3.3 is out, but MPI still considered alpha!
- Steve Plimpton's parallel FFT package?
  - Very general
  - Complex-to-complex only...

Sphyraena testing

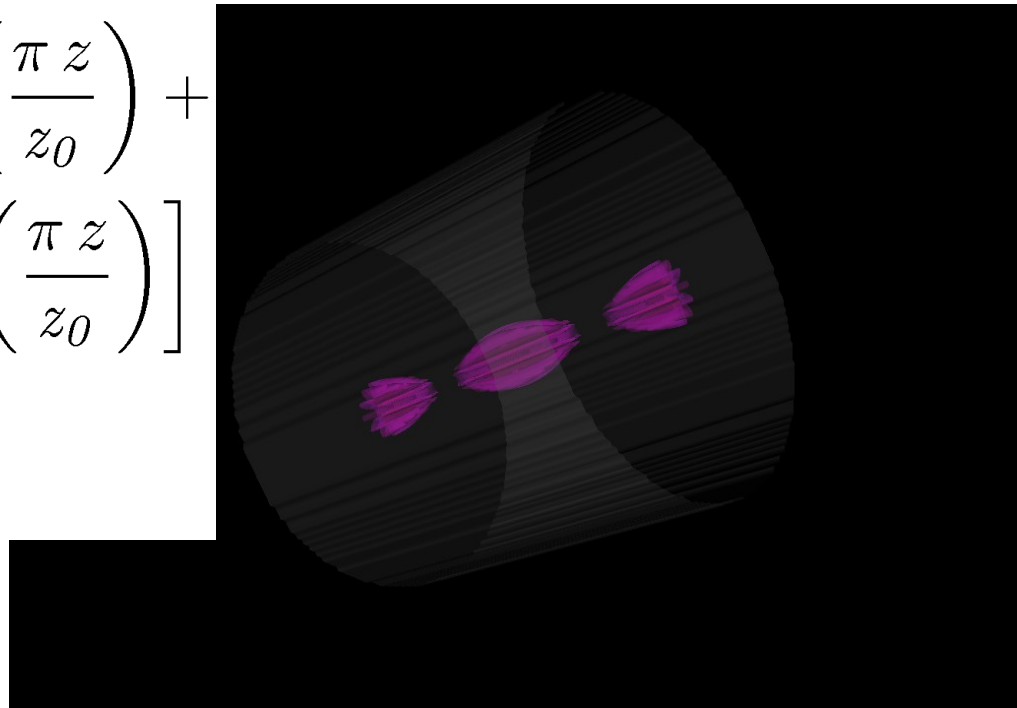
- Straightforward approach is to test Poisson solvers on analytically solvable problems, e.g., (non-)uniform conducting cylinder
  - High degree of symmetry makes for poor tests
    - also makes for solvable problem



Sphyraena testing, cont.

- Better test: take a non-trivial charge density

$$\rho(r,\theta,z) =$$
$$\Big[\Big(\big(18\,{r_0}^2 - 14\,r^2\big)\,\sin^2(3\,\theta) +$$
$$\big(18\,r^2 - 18\,{r_0}^2\big)\,\cos^2(3\,\theta)\Big)\,\cos^2\left(\frac{\pi\,z}{z_0}\right)\,{z_0}^2 +$$
$$\big(2\,\pi^2\,r^4 - 2\,\pi^2\,r^2\,{r_0}^2\big)\,\sin^2(3\,\theta)\,\sin^2\left(\frac{\pi\,z}{z_0}\right) +$$
$$\big(2\,\pi^2\,r^2\,{r_0}^2 - 2\,\pi^2\,r^4\big)\,\sin^2(3\,\theta)\,\cos^2\left(\frac{\pi\,z}{z_0}\right)\Big]$$
$$\Big/\,\big(r^2\,{r_0}^2\,{z_0}^2\big)$$

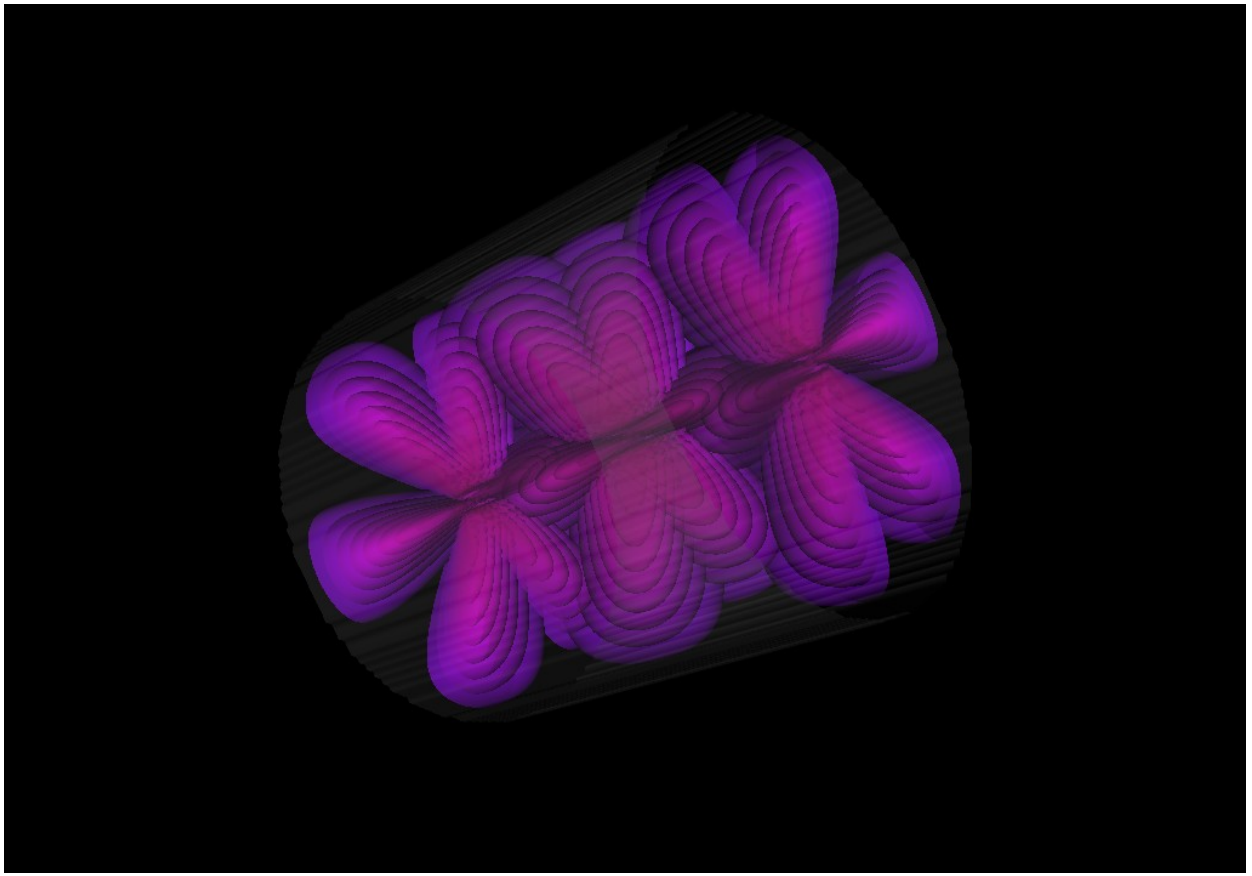


Sphyraena testing, cont.

- ... and produce a test in all coordinates

$$\phi(r,\theta,z) = \left(1 - \frac{r^2}{r_0^2}\right) \sin^2(3\theta) \cos^2\left(\pi \frac{z}{z_0}\right)$$

Procedure may be easily extended to any boundaries described by analytical expressions



Sphyraena elliptical solver

- Simplest transverse boundary conditions for Poisson solvers are open, rectangular closed and cylindrical closed
  - Unfortunately, the beam pipe currently of most interest at Fermilab is the Main Injector's

The Main Injector beam pipe

From a simulator's perspective, the F117 Stealth Fighter is ideal:it doesn't even really look like an airplane, but it was constructed with an easy-to-simulate geometry

In contrast, the Main Injector beam pipe is a challenge to simulators: 12.3 cm wide by 5.31 cm high (5.08 cm high under vacuum), it isn't even exactly elliptical.

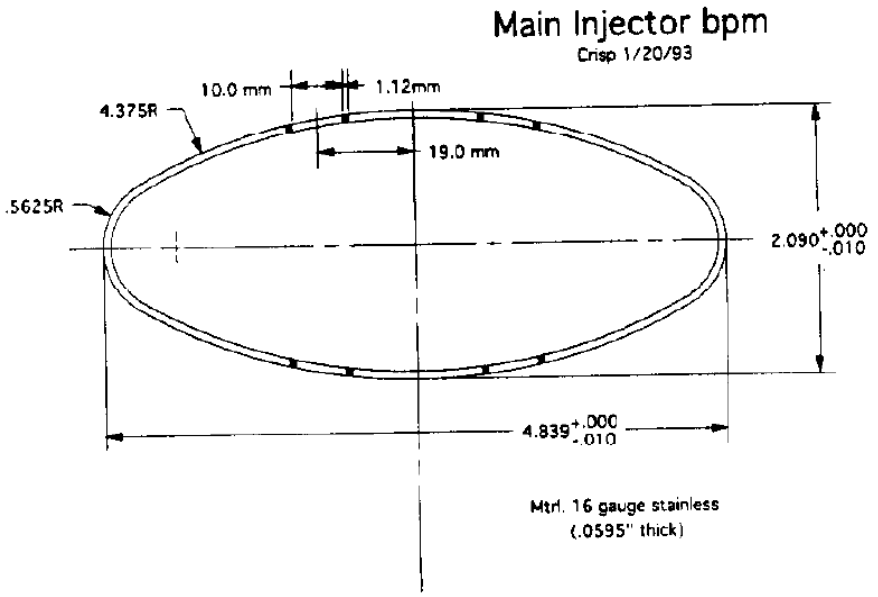
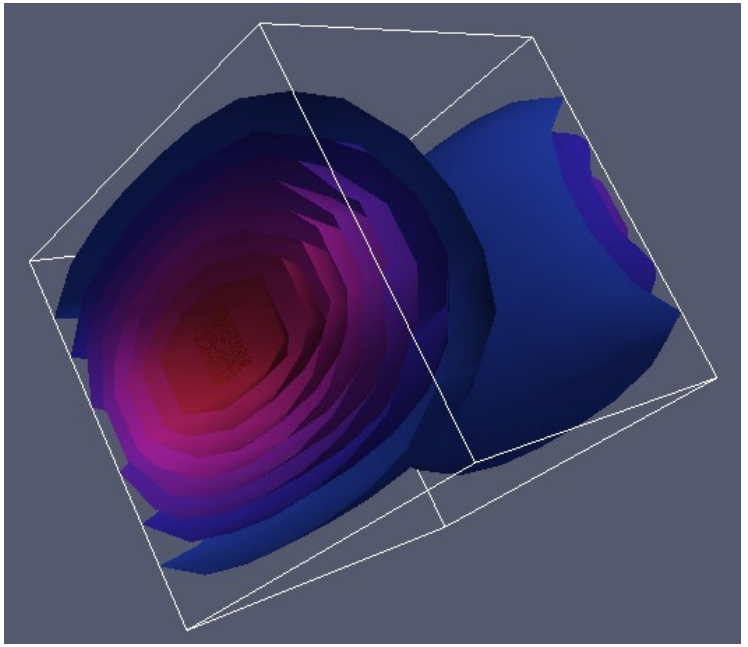


Figure 1: Geometry of Main Injector Beampipe and Beam Position Monitor.

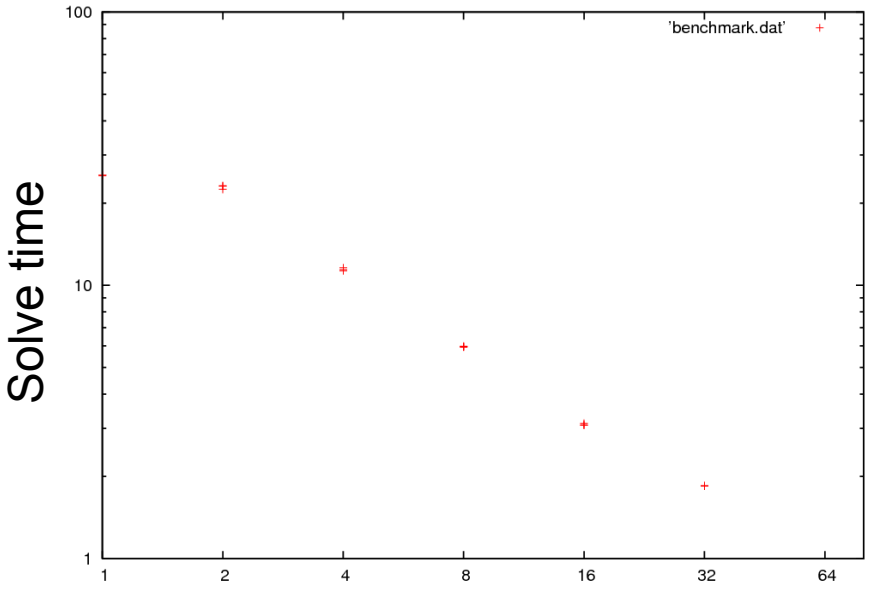
Sphyraena elliptical solver, cont.

- New, FD-based elliptical solver
  - Uses PETSc
  - Work done by Pavel Aleynikov - *Summer student from Russia*
    - Currently completing integration with electric field calculation, etc.

Field solution for benchmark problem



excellent parallel scaling performance provided by PETSc libraries



Resistive Wall Impedance Module

- Developed for BeamBeam3D simulations of Tevatron
- Dipole component of resistive-wall wakefields
- Includes true multiple bunch implementation in Synergia2
  - Bunches are coupled only through resistive wall
- Kicks are applied to each particle from all earlier slices

$$\frac{\Delta\vec{p}_\perp}{p} = \frac{2}{\pi b^3}\sqrt{\frac{4\pi\epsilon_0 c}{\sigma}}\,\frac{N_j r_p <\vec{r}_j>}{\beta\gamma}\,\frac{L}{\sqrt{z_{ij}}}$$

(A few words on) Electron Cloud module

- Electron cloud module under development
  - Primary developer Paul Lebrun (so far)
    - New hire starting in January
  - Electron production
    - TxPhysics
  - Cloud evolution
    - Single-particle transport currently being benchmarked against Vorpal
  - Beam-cloud interaction
    - Sphyraena solver

Infrastructure

- Refactoring for a more interchangeable set of modules
  - Break ties with CHEF and IMPACT at the *interface* level
    - General beamline interface more complex than expected
  - Cleaner language separation; eliminate mixed-language modules
    - Python driver
    - Python modules
    - C++ modules
    - (Fortran modules)

Infrastructure, cont.

- Current Python-F90 wrapping done with Forthon
  - I hate Forthon
    - It's not F90, it's Forthon!
    - Complex portability issues
      - Either Just Works or Just Does Not Work
- Next-generation wrapping to be done with Babel
  - Current design utilizes F90 structs
    - Babel F90 struct implementation requires F2003
      - Portability issues
    - Better interface probably should hide structs
  - This effort currently stalled
    - Priorities, personnel changes
    - Difficulty: undo Forthonization

Porting

- BlueGene and (Cray) XT4 continue to pose a porting problem
  - Neither BlueGene/L nor XT4 support shared libraries, the default method for implementing loadable modules in Python
  - BlueGene/P supports shared libraries in “Linux mode”
    - Python is an advertised feature of BlueGene/P
    - First stab at Synergia port failed at building PyMPI
      - Cross-compilation of Python modules not a fully solved problem

Porting, cont.

- Porting Synergia2 to capability platforms is a high priority for the next year of development
  - We will make the compromises necessary
    - Need to understand the differences between unsolved, inconvenient and impossible
  - We hope to get some help from CS by utilizing Babel
    - Certain chicken-egg issues

Conclusions

- Solvers progressing
  - Significant work done w/ summer student
  - Non-trivial testing providing significant boost
- Capabilities added and being added
  - Resistive wall in place
  - Electron cloud in progress
- Infrastructure getting cleaner
  - Refactoring ongoing
- Porting
  - Capability machines major focus of this year's work