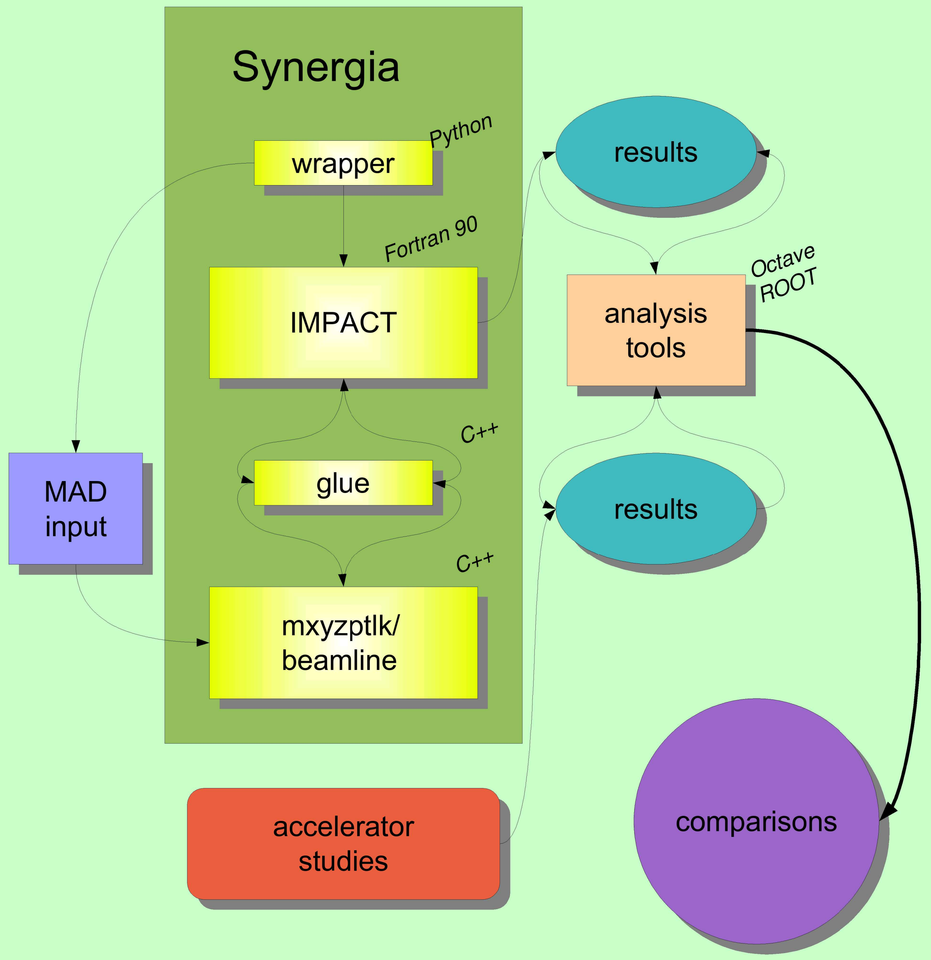
Synergia: a Hybrid, Parallel Beam Dynamics Code with 3D Space Charge

J. Amundson and P. Spentzouris
Fermi National Accelerator Laboratory

Part of the "Advanced Computing for 21st Century Accelerator Science and Technology" Collaboration



3D space-charge code for circular and linear machines

- Reuses existing beam dynamics packages
  - IMPACT
  - mxyzptlk/beamline libraries
- Advanced parallelism
- Modern build system and code distribution
- Humane interface and MAD lattice input

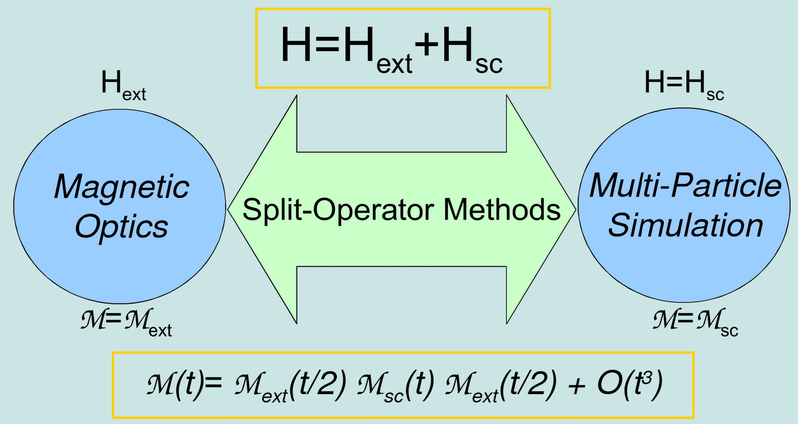
Modern build system

portability is key

- Builds without modification on Linux
  - Intel or Portland F90
  - g++
  - MPICH or lam
- GNU Autotools
  - automake
  - autoconf
- Builds without modification on AIX
  - XL Fortran
  - Visual Age C++
  - POE

IMPACT*

- Fortran 90
- Fully parallel
- Linear accelerators
- 3D space charge via split-operator technique



*Ji Qiang, Robert D. Ryne and Salman Habib

mxyzptlk/beamline libraries†

- C++
- Flexible libraries
- Many features
  - Lie-algebraic maps
  - MAD parser
  - Element splitting

Libraries

- basic_toolkit
- physics_toolkit
- mxyzptlk
- beamline
- Machines

†Leo Michelotti

Humane interface

Utilizes Python for simplicity and flexibility

```
ip = impact_parameters.Impact_parameters()
ip.processors(16,4)
ip.space_charge_BC("trans finite, long periodic round")
ip.input_distribution("6d gaussian")
ip.pipe_dimensions(0.04,0.04)
ip.kinetic_energy(0.400)
ip.scaling_frequency(201.0e6)
ip.x_params(sigma = .004 , lam = 1.0e-4)
ip.y_params(sigma = .004 , lam = 1.0e-4)
pz = ip.gamma() * ip.beta()*ip.mass_GeV
ip.z_params(sigma = 0.10, lam = 3.0e-4 * pz)
ip.particles(2700000)
ip.space_charge_grid(65,65,65)
booster = impact_elements.External_element(length=474.2,
                                           kicks=100, steps=1, radius=0.04,
                                           mad_file_name="booster.mad")
for turn in range(1,11):
    ip.add(booster)
my_impact = impact.Impact(ip)
currents = ( 1.0e-10, 0.011, 0.020, 0.042 )
for current in currents:
    ip.current(current)
    my_impact.prepare_run("bs1_I%0.3f" % current,
                          clock_limit = "01:30:00")
    my_impact.submit()
```

Identifiable names

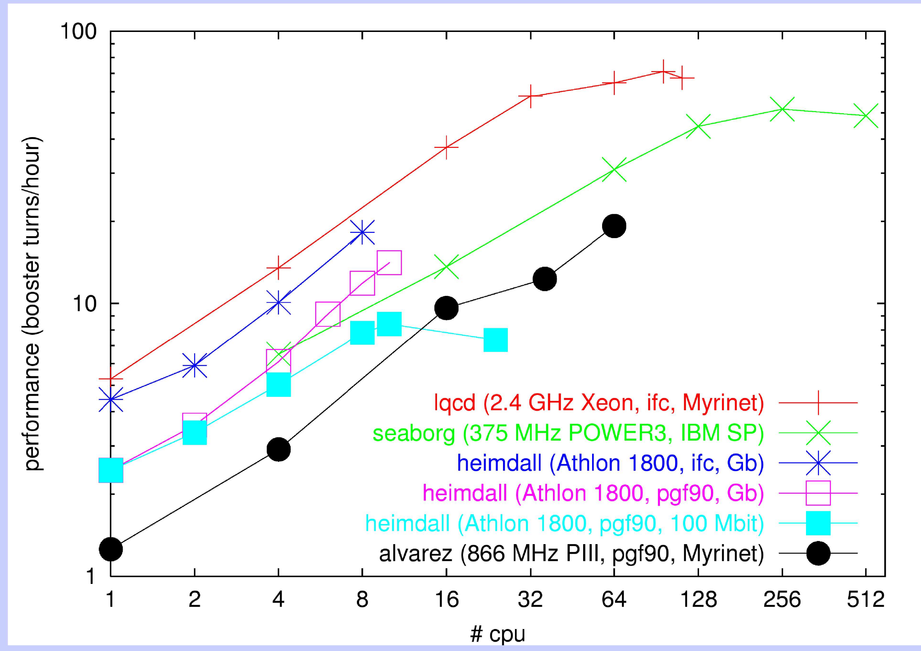
Beam parameters specified in m, GeV

Trivial use of Python

multiple runs in one script

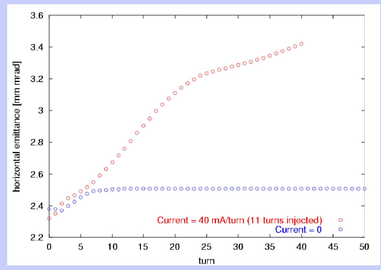
Parallel Performance

Simulation of 2.7M particles on a 65^3 grid for one complete Booster revolution including 100 space-charge kicks



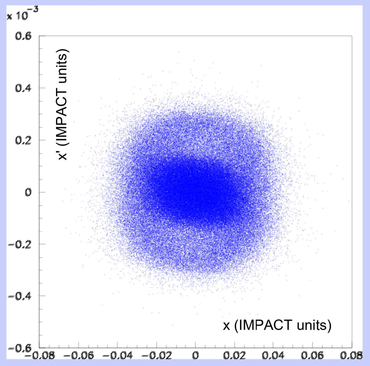
Application to FNAL Booster

Simulated horizontal emittance in the FNAL Booster

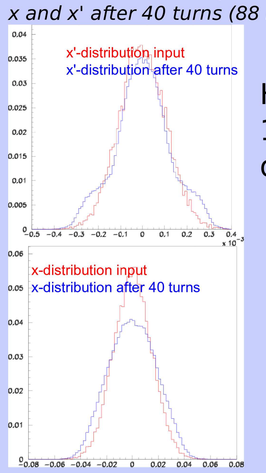


Simulations of horizontal emittance show growth with space charge, no growth without space charge

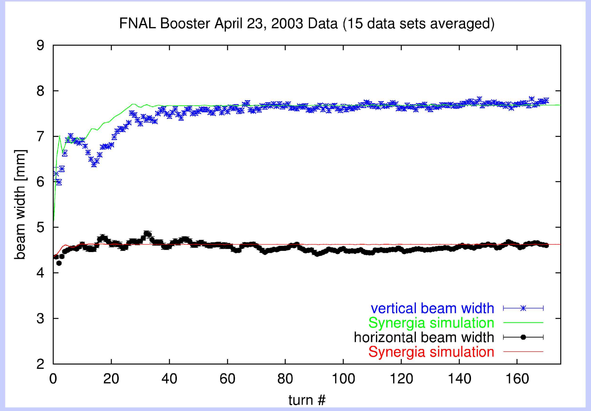
Simulated transverse phase space projected onto horizontal plane after 40 turns (88 ms)



Simulated transverse phase space projected onto x and x' after 40 turns (88 ms)



Halo formation seen in 1D and 2D distributions



Synergia simulation compared with measured beam widths in the FNAL Booster