# Synergia Status

James Amundson and Eric Stern, *et al.*

Fermilab

August 21, 2013

# Status

- Synergia code status
  - Who, what and why
  - End-user release
  - Cluster and supercomputer status
  - Plans for new architectures
- Synergia application status

# What: Synergia

- Accelerator simulation package
  - independent-particle physics
  - collective effects

https://cdcvs.fnal.gov/redmine/projects/synergia2/wiki

- Designed for range of computing resources
  - laptops and desktops
  - clusters (not farms!)
  - supercomputers
  - GPU/Intel Phi accelerated platforms

# Who: Personnel

Synergia is developed and maintained by the

**Accelerator Simulation Group**

of the

**Scientific Computing Simulation Department**

*virtual member of*

**APC**

James Amundson, Paul Lebrun, Qiming Lu, Alex Macridin, Leo Michelotti (CHEF), Chong Shik Park, (Panagiotis Spentzouris) and Eric Stern

# What: Physics

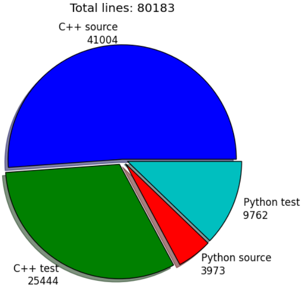
- Single-particle physics are provided by CHEF
  - C++ libraries developed by Leo Michelotti
  - direct symplectic tracking
    - magnets, cavities, drifts, etc.
  - (and/or) arbitrary-order polynomial maps
  - many advanced analysis features
    - nonlinear map analysis, including normal forms
    - lattice functions (multiple definitions)
    - tune and chromaticity calculation and adjustment
    - etc.
- Apertures
- Collective effects (single and multiple bunches)
  - space charge (3D, 2.5D, semi-analytic, multiple boundary conditions)
  - wake fields
    - can accommodate arbitrary wake functions
  - electron cloud
    - proof of principle only

# Synergia 2.1: first end-user release

- Synergia 2.1 is a major milestone
  - very different from Synergia 1
  - significantly different from Synergia 2
  - **designed for widespread use**
- Synergia is a mix of C++ and Python
  - all computationally-intensive code is written in C++
  - user-created simulations are usually written in Python
    - pure-C++ simulations are possible
- Synergia provides a set of functions and classes for creating simulations
  - many examples available
- Virtually every aspect of Synergia is designed to be extendable by the end-user
  - code in C++ and/or Python

# Testing and release status

- We do a great deal of testing in Synergia
    - ≈ 80k lines of code
        - excluding CHEF
    - $>$ 40% tests
    - Integrated testing ongoing
        - Space-charge trapping benchmark in collaboration with CERN and GSI



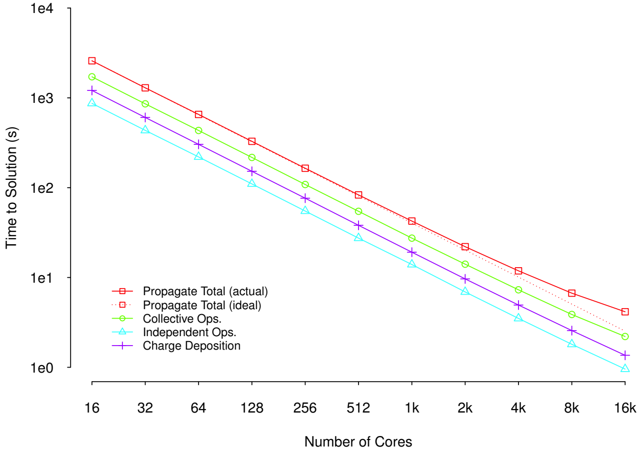

- Current status
    - Beta release 1
    - Manual in process (content already available on web)
    - API will be finalized with final release

# Scalability on clusters and supercomputers

- Why?
  - Simulations take too long on single computers
  - Require fast inter-node communication (no farms)
- Why is it hard?
  - Typical simulation is not very large
    - 2e5 - 2e6 degrees of freedom
    - supercomputer scale is 1e4-1e6 cores
- Breakthroughs
  - Communication avoidance
    - redundant solves improve overall scaling
  - Some simulations can be huge
    - statistics (many particles, requires communication avoidance)
    - multiple bunches
    - parameter scans
- We can now fully utilize current and (near?) future supercomputers

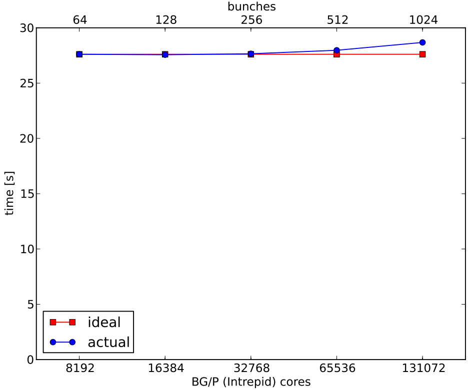
# Strong scaling

- Single-bunch Synergia simulations have been shown to scale past 8192 cores on ALCF's Mira, a BlueGene/Q supercomputer
  - Strong scaling, *i.e.*, fixed problem size
    - ($32 \times 32 \times 1024$ grid, 100 grid cells per particle, trivial apertures)

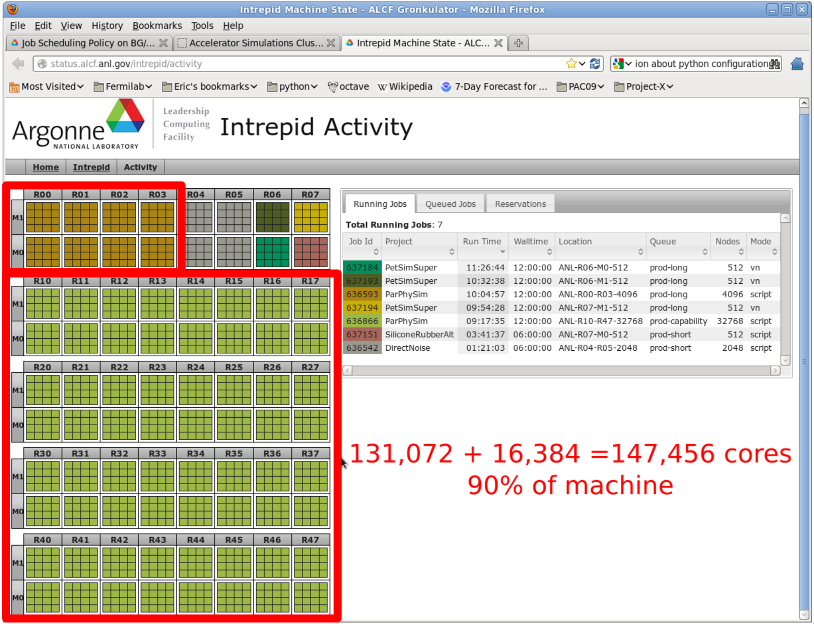


# Weak scaling

Performed large-scale scaling benchmarks on production BlueGene/P machine at Argonne Leadership Computing Facility: Weak scaling, *i.e.*, fixed ratio (problem size)/(compute size) ($32 \times 32 \times 1024$ grid, 100 grid cells per particle, trivial apertures)

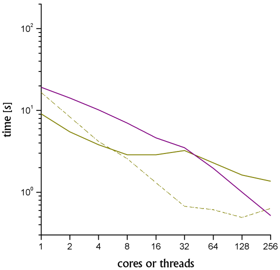


# Production runs on supercomputers

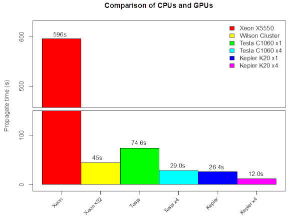


# Work on multicore and GPUs

OpenMP




GPU




- Promising results so far
  - not yet production
- *Best performance in a single box* a promising new direction
  - Also has supercomputer applications, *e.g.*, Titan
  - Intel Phi platform is a promising new variation
  - Possible applications include Main Control Room use

# Status

- Synergia code status
- Synergia application status

# Synergia applications

## Status and plans

1. Main Injector – Eric Stern

2. Mu2e Extraction – Chong Shik Park

3. Booster Multi-bunch Instabilities – Alexandru Macridin

$\sqrt{-1}$. LHC Injector Upgrades

# Available Computing Resoures

In-house Wilson Cluster

- 24 12-core Intel Nehalem nodes
- 34 32-core AMD Opteron nodes

INCITE 2013 Award

- 50M core-hours Intrepid BG/P
- 30M core-hours Mira BG/Q

INCITE 2014 Proposal

- 80M core-hours Mira BG/Q

NERSC Hopper and Edison

# Main Injector

Increased intensity in the Project-X era could lead to unacceptable particle losses due to space charge and wakefield effects.

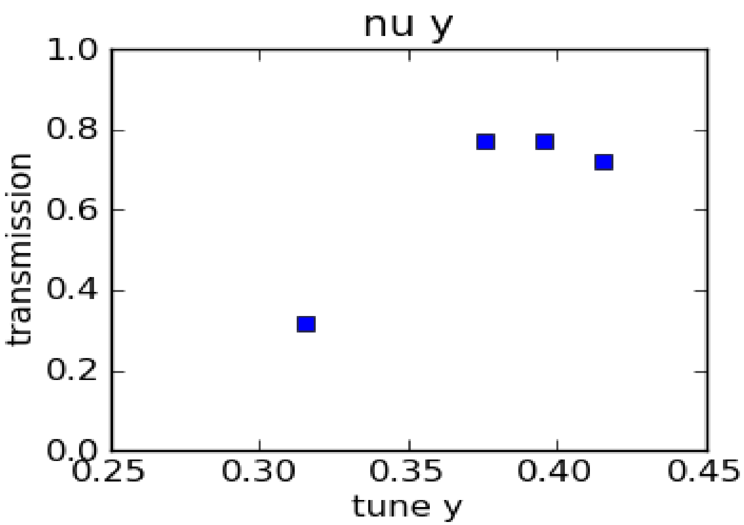
Study with Synergia to understand the magnitude and characteristics of losses.

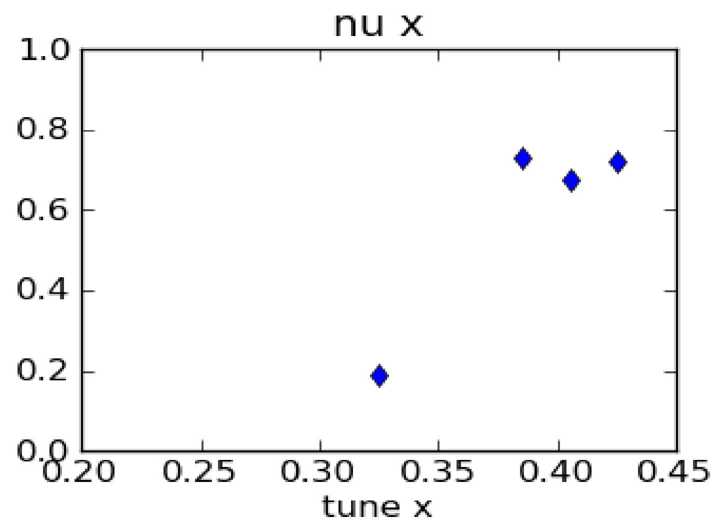
# Main Injector Simulations

- Understand current losses with apertures
- Survey losses in tune space
- Sensitivity to multipole errors

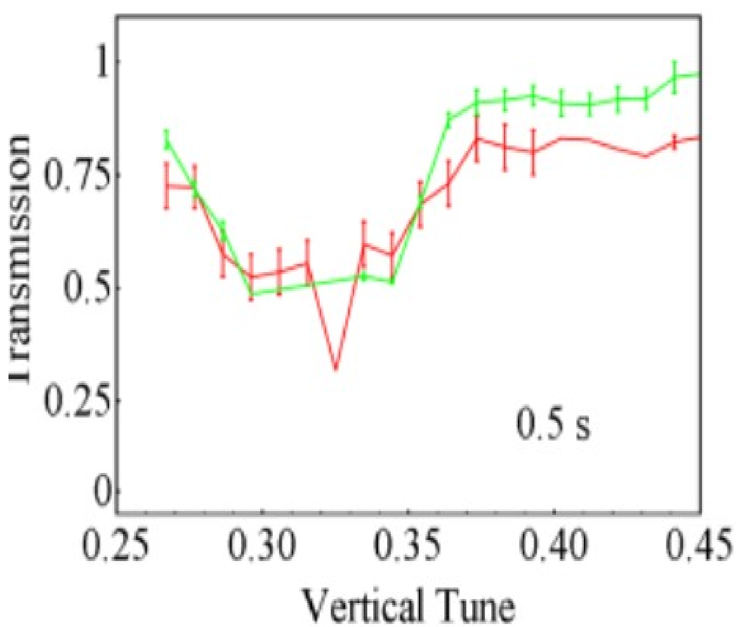
# Plot transmission (0.5 s) vs tune

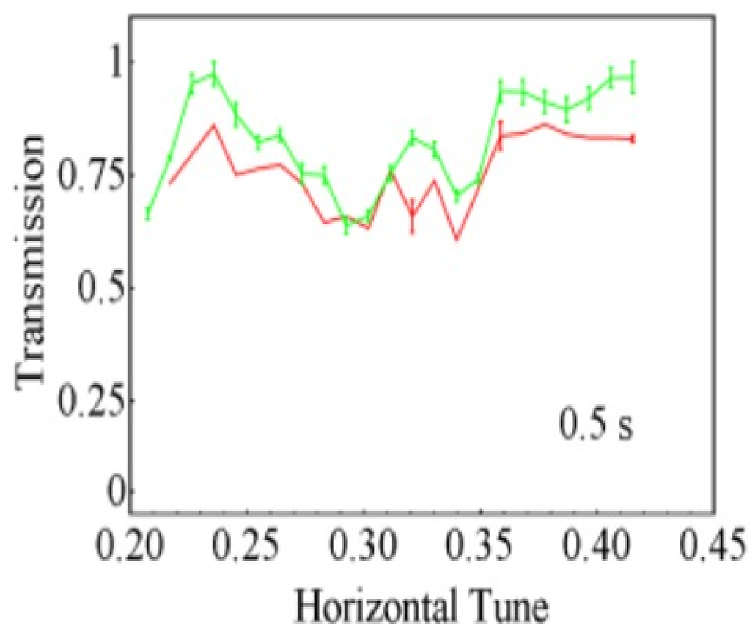
7 runs 256 cores/run 1000 turns/run takes2 weeks on the Wilson cluster




simulation



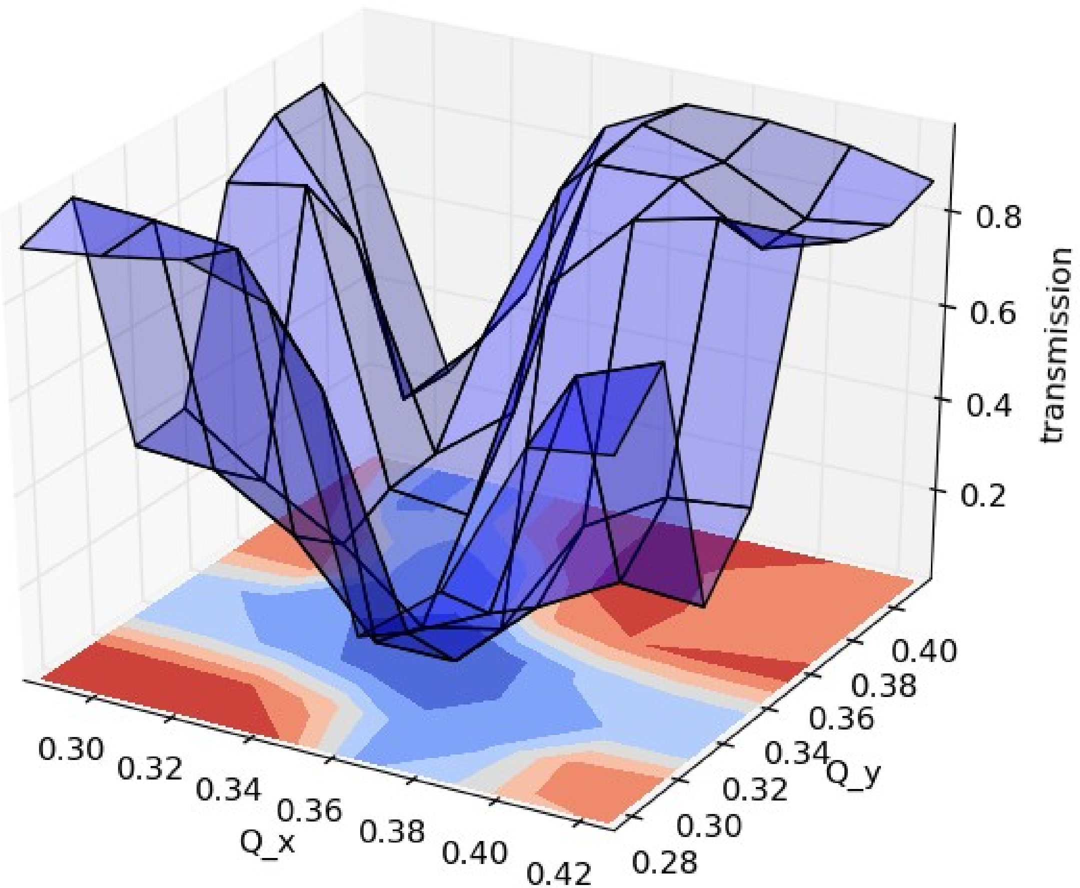

measurement

Short bunch

Long bunch

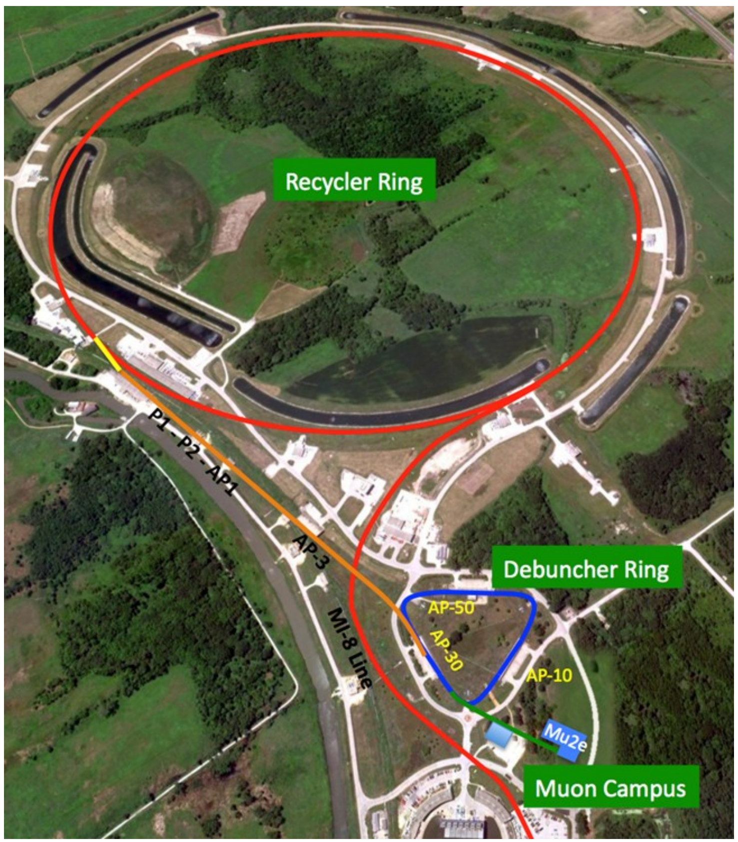
# Detailed Tune Space Survey

64 runs 2048 cores/run 6000 turns/run takes 1 week on BG/P



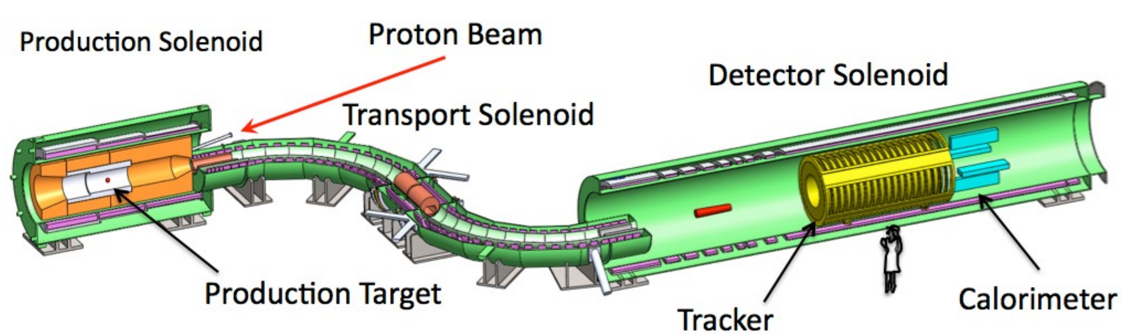
# Mu2e Extraction Simulations

The experiment requires a constant stream of protons during its data taking period.



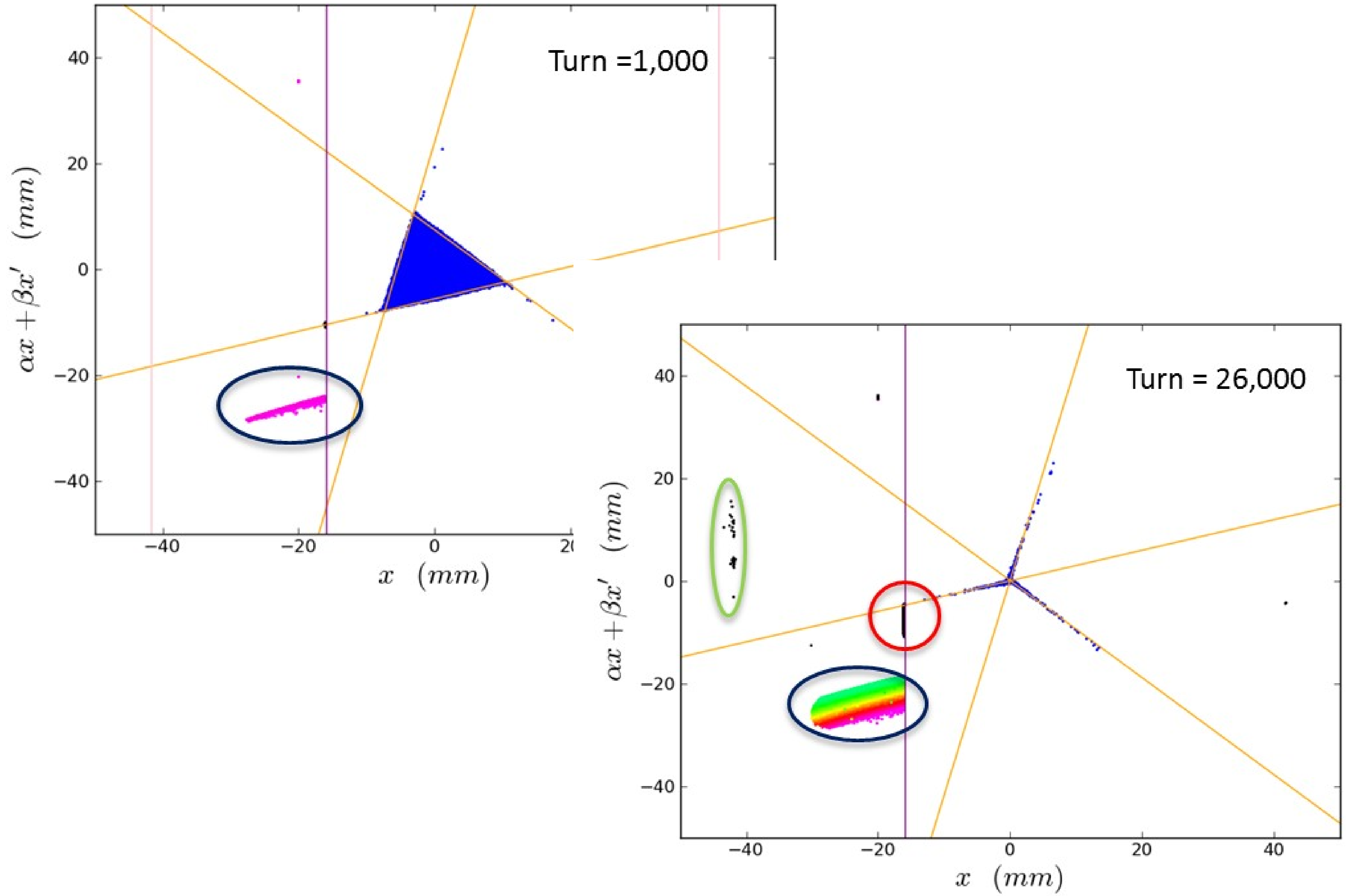

In resonant extraction, the machine is tuned so that extreme particles in phase space are extracted.

Space charge alters the phase space distribution and complicates the extraction process.



# Resonant Extraction process...

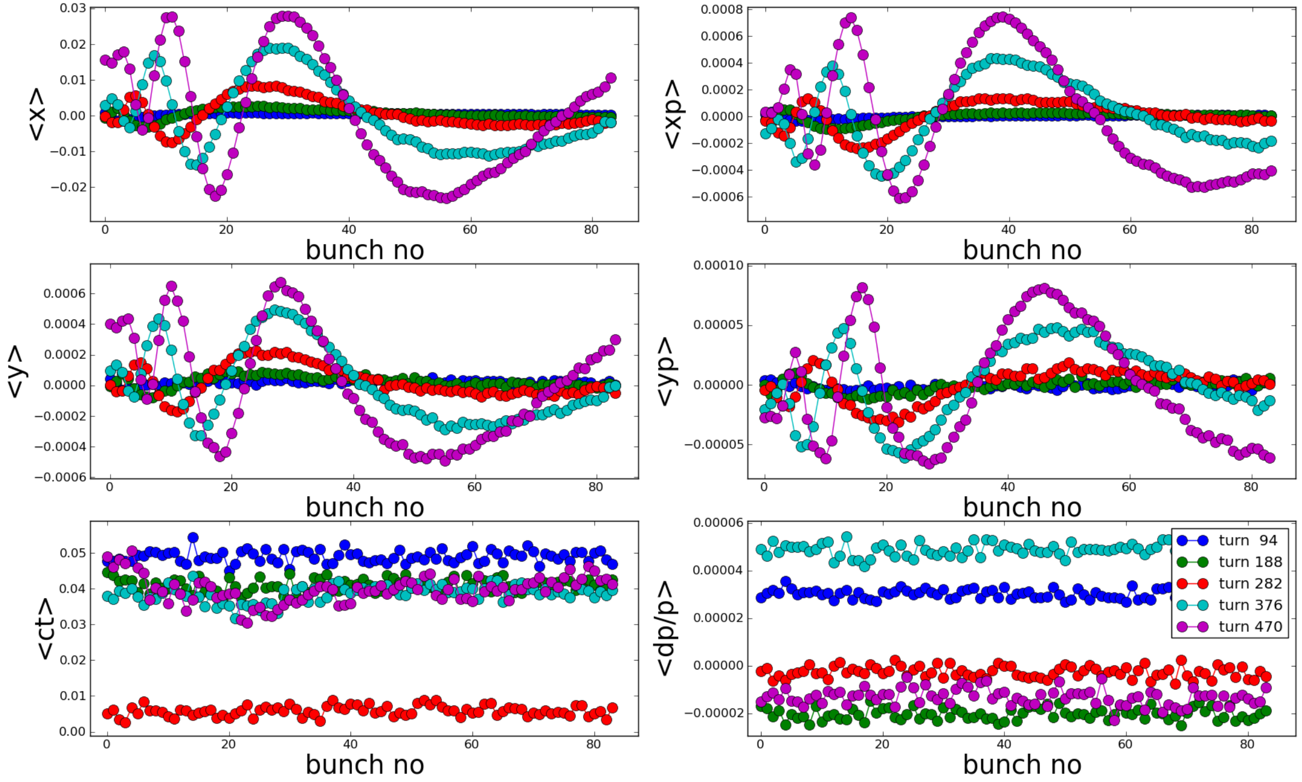
## Showing extraction aperture, septum wire, beampipe



# Booster Simulations

Wakefields* incorporated in Synergia reproduce observed multi-bunch instability.

Effect is only evident in simulation with many bunches

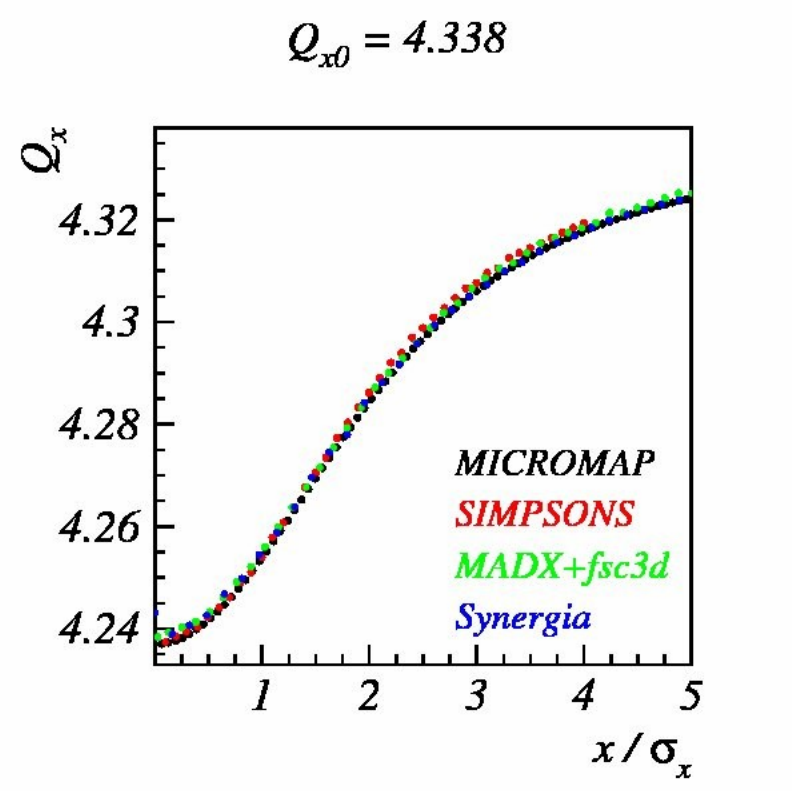


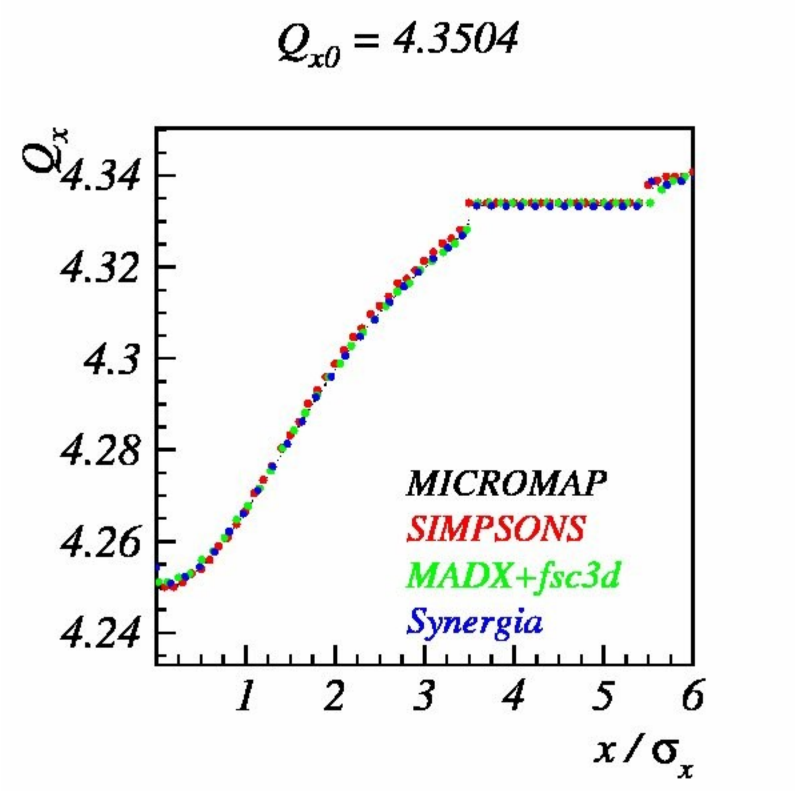

* Invited talk at Particle Accelerator Conference '13

# LHC Injector Upgrades

In benchmarking and validation for possible future work.

- Possible CERN contributions
- Prestige for Fermilab



# Issues

- Growing number of end users
  - Great!
  - Support requests are increasing
- We do not have control over end-user platforms
  - Individual user machines
    - doable
  - Supercomputer centers
    - at their mercy
    - things like compiling our own gcc can be impossible
    - I anticipate using C++11 circa 2021
    - Future directions could be radical
- New architectures are promising, but require a substantial amount of work/research