Synergia 2.1 status

James Amundson Alexandru Macridin Chong Shik Park
Panagiotis Spentzouris Eric Stern

Fermilab

ComPASS Meeting 2012

Background

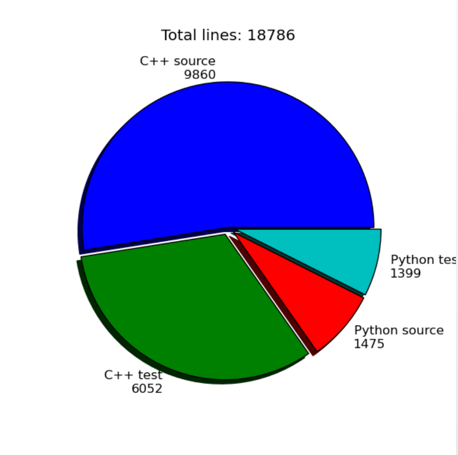
- Synergia is a research project.
  - It is always in development.
  - The API has been continually evolving.
- We want to expand our base of end users.
  - End users hate continually evolving APIs.
  - Developers hate them, too.

Defining a stable API is our top development priority.

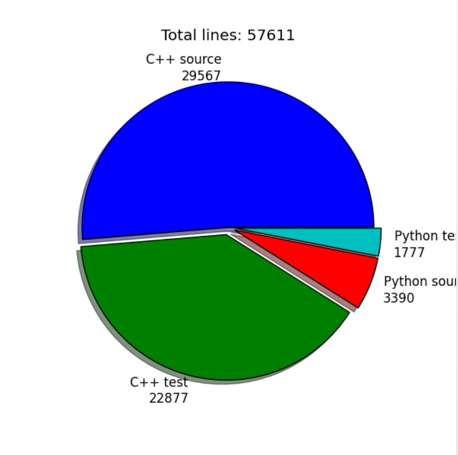
Developing an API flexible accommodate future physics models and code optimizations has been our primary challenge for Synergia 2.1.

Progress since 2010 meeting

- 2010: *Synergia refactor* with preliminary examples



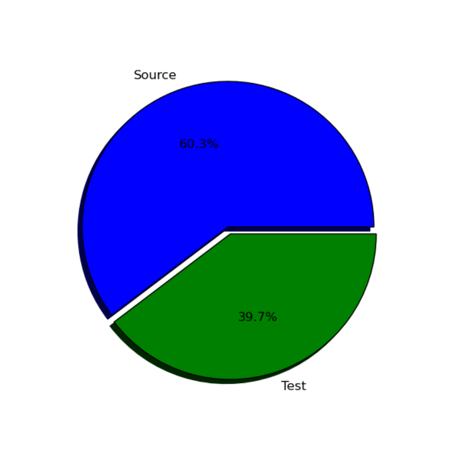
- 2012: *Synergia 2.1* used in production



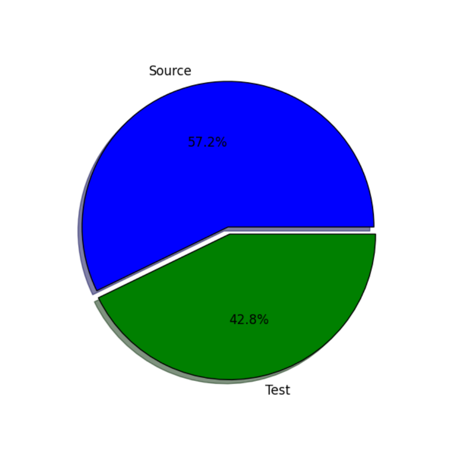
$\gtrsim 3x$ LOC, primarily C++

Progress since 2010 meeting

- 2010



- 2012



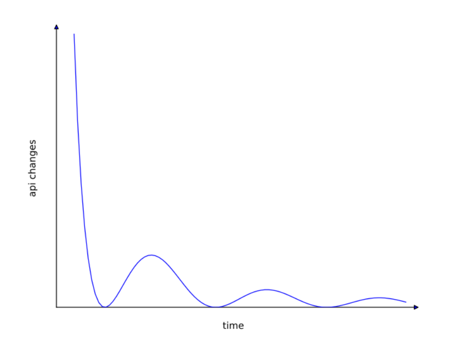
We have maintained our commitment to rigorous testing.

Abstract API

Core concepts

- Bunch
  - dense array of phase space coordinates with id
  - maintains State
    - fixed-t or fixed-z
    - Lorentz frame
- Operator
  - Independent operator
    - Lattice (abstract, extensible)
  - Collective operator
- Step
  - series of Operators
    - e.g., leading-order split operator

- Propagator
  - applies Steps
  - applies various Actions (lattice ramping, diagnostics) after selected steps
  - checkpoints

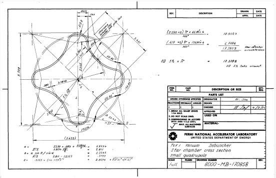


Synergia 2.1 in practice

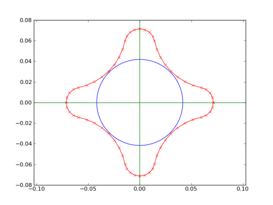
- Users can utilize the full API in either C++ or Python.
  - Python for flexibility
  - C++ to make life easier in some situations, e.g., debugging, novel architectures, etc.
- Some classes can be extended trivially.
  - Propagate actions allows for ramping, etc.
  - Diagnostics actions is already very flexible, but can be extended to make arbitrarily complicated decisions.
- All operators can be extended.
  - new Collective operators
  - new Independent operators implementations
- New simulation schemes can be employed.
  - new generators of Steps

New Feature: Advanced aperture model

- Apertures can be associated with elements and/or steps
- Geometric
  - Circular
  - Elliptical
  - Rectangular
  - Polygon
  - Wire
- Abstract
  - Finite
    - ordinarily, a finite aperture is applied each step
  - Phase space
  - Lambertson
    - removes particles



Engineering drawing of FNAL Debuncher quad cross-section



Synergia implementation, including inscribed circle optimization

New Feature: Automatic checkpointing

- All simulation objects are serializable
  - implementation uses Boost Serialization library
  - even, e.g., objects with open files
- Multiple file formats are supported
  - Binary
    - efficient
  - XML
    - useful for debugging
  - this functionality comes for free
- Serialization available for both C++ and Python objects
  - *including end-user objects*
- User specifies checkpointing parameters
  - on-the-fly
  - do p out of q total turns

Status and Landmarks

Synergia 2.1 is being used for all production work in our group at Fermilab.
Status:

- Fermilab Main Injector
  - Space charge, multipoles, detailed apertures, orbit bumps
  - See Eric Stern's talk.
- Fermilab Booster
  - Space charge, wakes, multiple bunches
  - See second half of this talk.
- Fermilab Debuncher
  - Space charge, ramping, resonant extraction
  - See Chong Shik Park's talk.
- Hybrid MPI-OpenMP and MPI-GPU versions
  - See Qiming Lu's talk.

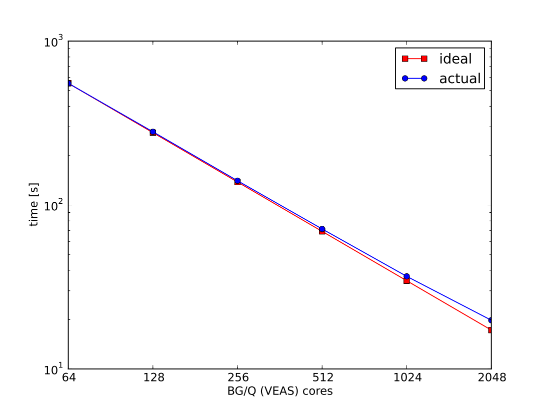
Status and Landmarks

Landmarks:

- Full, 3D simulation of resonant extraction from the Fermilab Debuncher for Mu2e
  - 30,000 turns
  - 7,200,000 time steps
  - 20 chained checkpointed jobs
- Optimized Mu2e simulation
  - 2.5D space charge
    - benchmarked against 3D space charge
  - Many optimizations
    - Inscribed circle approach to polygon aperture sped up that portion of the calculation by $\sim 100x$.
  - Now able to do 30,000 turns in 24 hours

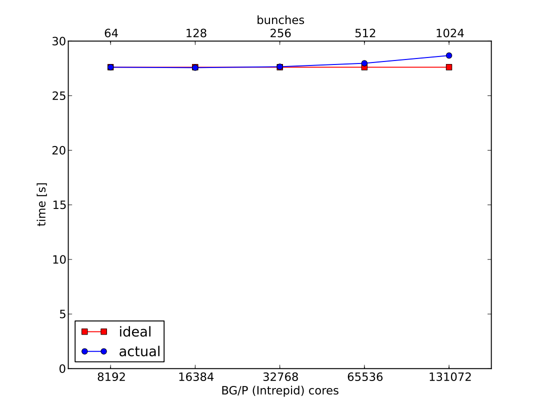
Scaling

Performed large-scale scaling benchmarks on ALCF machines:
Strong scaling on BG/Q ($32 \times 32 \times 1024$ grid, 100 grid cells per particle, trivial apertures)



Scaling

Performed large-scale scaling benchmarks on ALCF machines:
Weak scaling on BG/P ($32 \times 32 \times 1024$ grid, 100 grid cells per particle, trivial apertures)



Largest simulation included 13,421,772,800 particles

Wakes and Fermilab Booster simulations

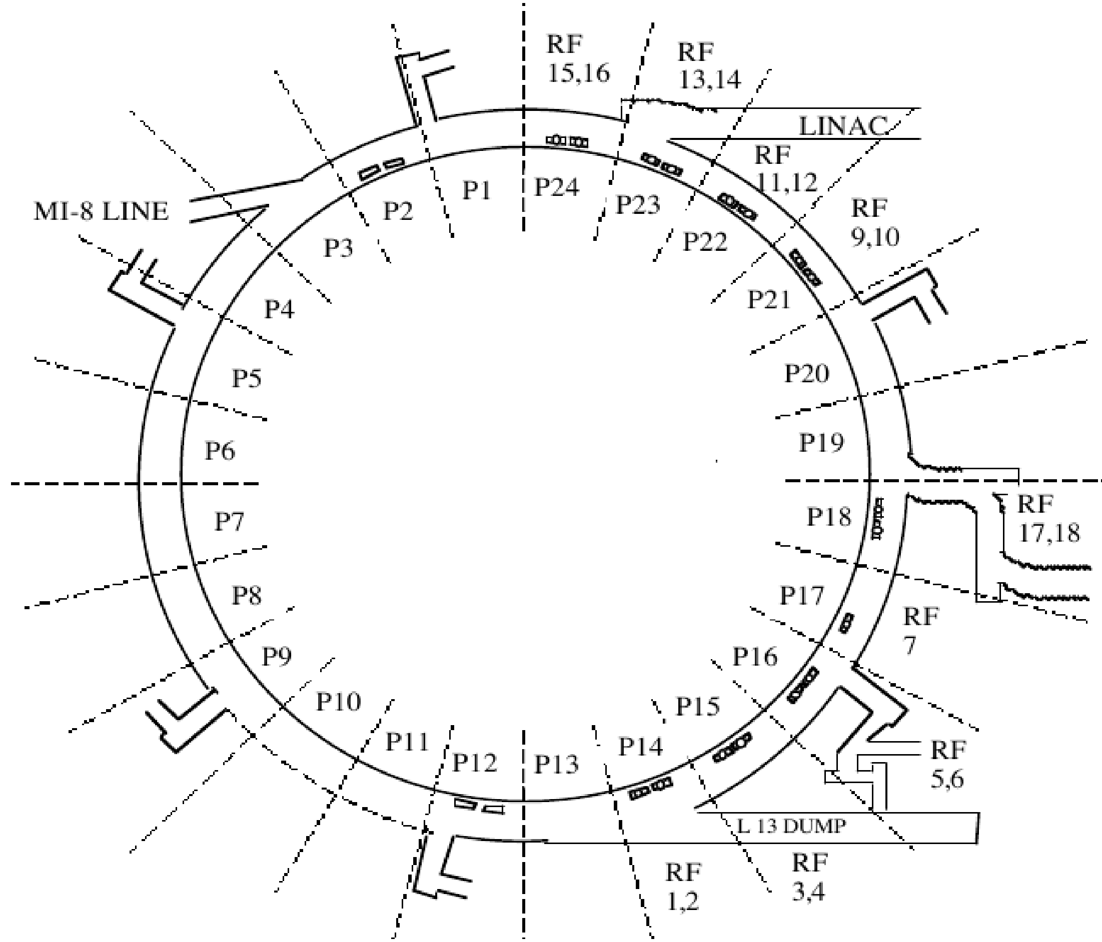
Alexandru Macridin
James Amundson
Panagiotis Spentzouris

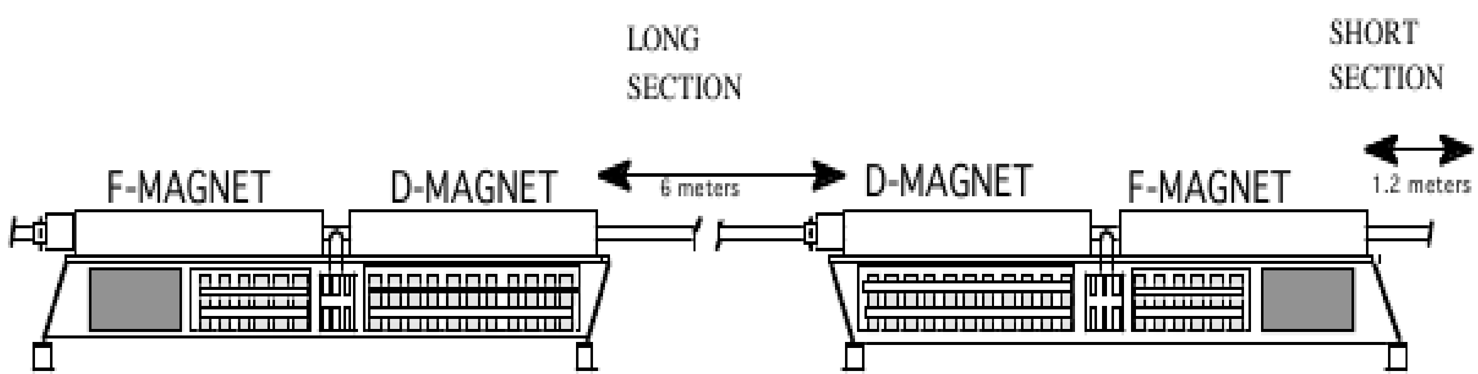
Fermilab Booster:

- Intensity ≈ 6×10^{12} p, much greater than the design value
- At injection, E= 400 MeV, γ=1.4, non-ultrarelativistic
- Laminated magnets
- Collective effects

  space-charge

  wake fields



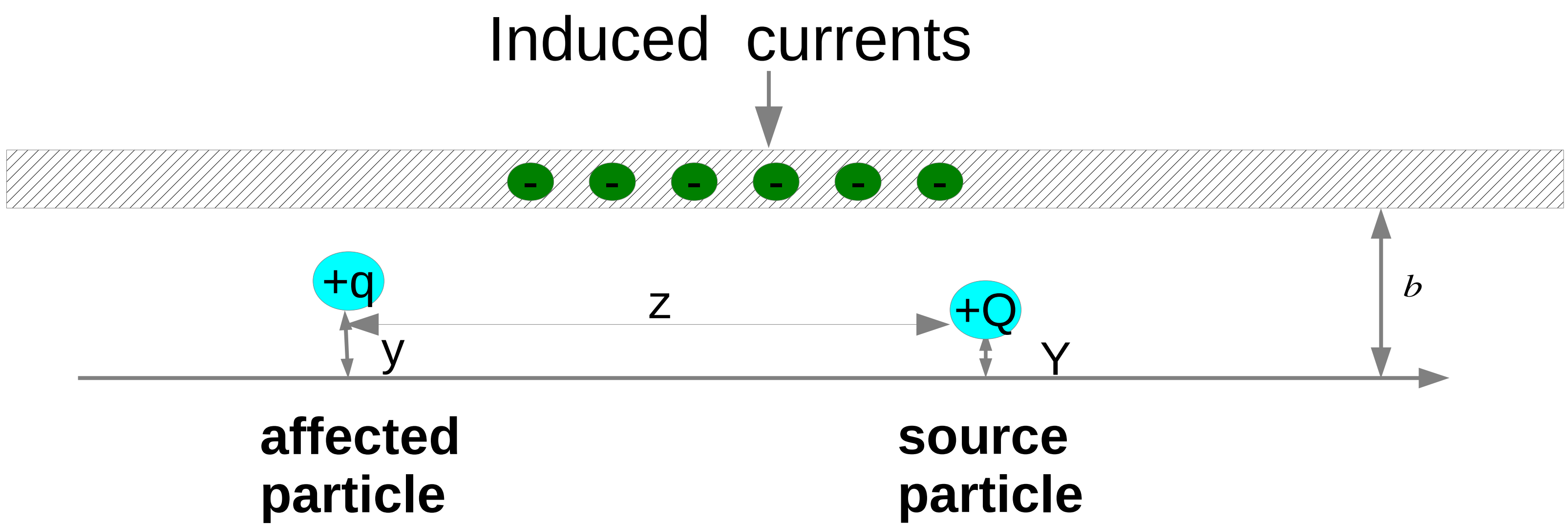


1. Calculation of the wake functions:
   - Parallel-plane geometry
   - Laminated chambers
   - Non-ultrarelativistic beam
   - Frequency dependent permeability in magnets

2. Booster simulations:
   - Nonlinear optics
   - Multi-bunch simulations
   - Various space-charge solvers and wakes specific for the lattice elements

3. Comparison with measurements:
   - Coherent tune shift
   - Kick attenuation

Wake functions



$$\beta c \Delta p_z = -qQW^{\parallel}(z)$$

$$\beta c \Delta p_x = -qQ\left(W_X^{\perp}(z)X + W_x^{\perp}(z)x\right)$$

$$\beta c \Delta p_y = -qQ\left(W_Y^{\perp}(z)Y + W_y^{\perp}(z)y\right)$$

$W_x^{\perp}(z) = -W_X^{\perp}(z)$ for parallel plane

For simulations we need: $W^{||}(z)$, $W_x^{\perp}(z)$, $W_Y^{\perp}(z)$, $W_y^{\perp}(z)$

Wake calculation:

- Solve the Maxwell's equations in the frequency domain for a point source moving with speed βc.
- The impedance Z(ω) is proportional to the force acting on the affected (trailing) particle .
- The wakes are obtain via Fourier transforms.

$$W^{\|}(z)=\frac{1}{2\pi}\int d\omega\, Z^{\|}(\omega)\, e^{-i\frac{\omega z}{\beta c}}$$

$$W^{\perp}_{x,y}(z)=\frac{i}{2\pi}\int d\omega\, Z_{x,y}(\omega)\, e^{-i\frac{\omega z}{\beta c}}$$

Boundary conditions:

The electromagnetic field is determined by the boundary conditions at chamber's walls:

We write Z(ω) =Z[*R*(ω)], where the wall surface impedance

$R(\omega)=E_z/H_x$, at wall boundary

- **The expression of impedance is valid for all types of walls with a particular geometry**
- **The wall's specific properties are included implicitly in *R*(ω)**

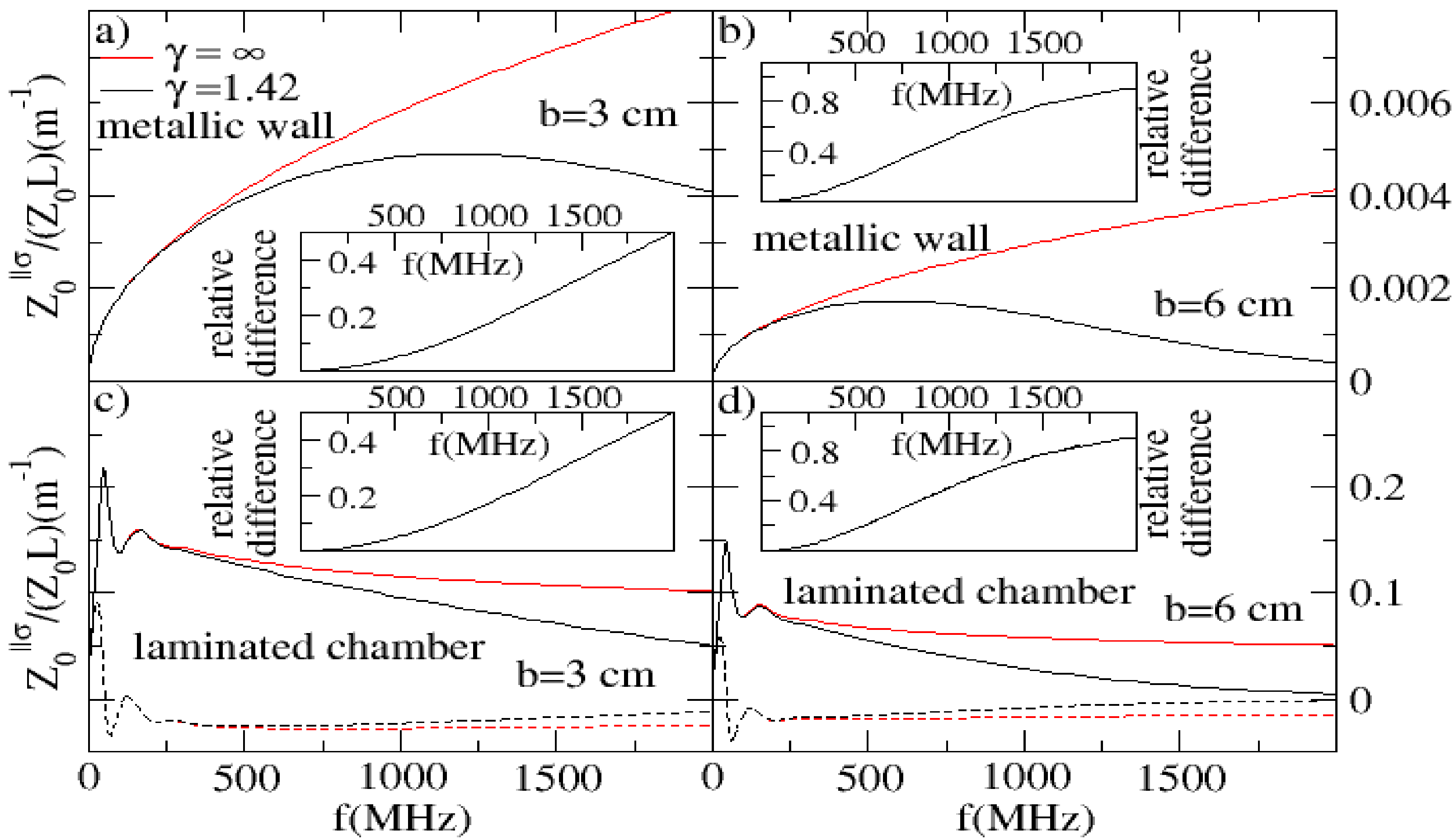
Examples:

for metallic wall $$R(\omega)=\frac{1+i}{\delta\sigma}$$

for the laminated chamber $$R(\omega)=i\frac{\eta^2+q^2}{q\frac{\omega}{c}\epsilon_{r1}}\tan q(d-b)$$

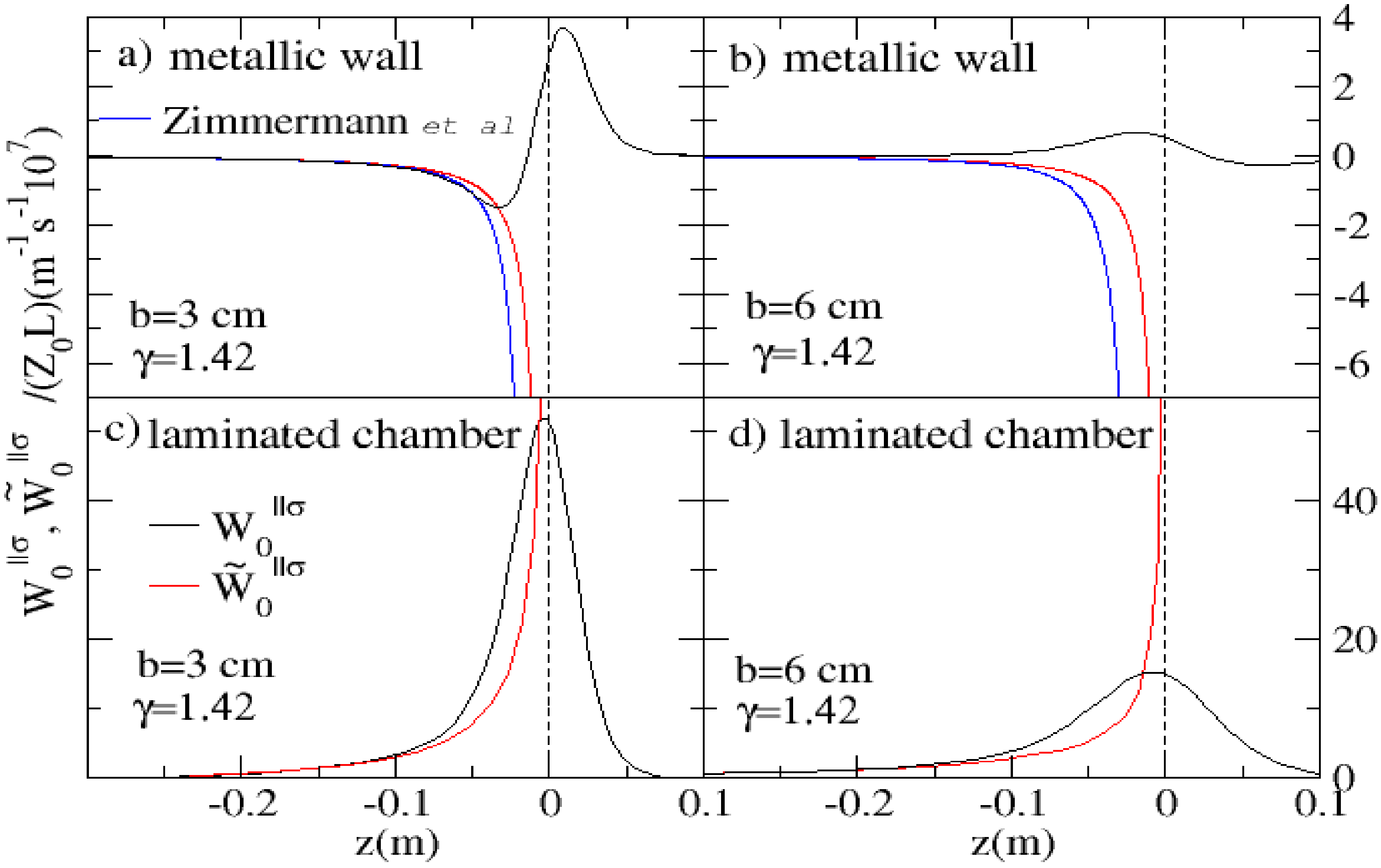
(see *A. Macridin et al*, PRSTAB 14, 061003, 2011)

Longitudinal impedance



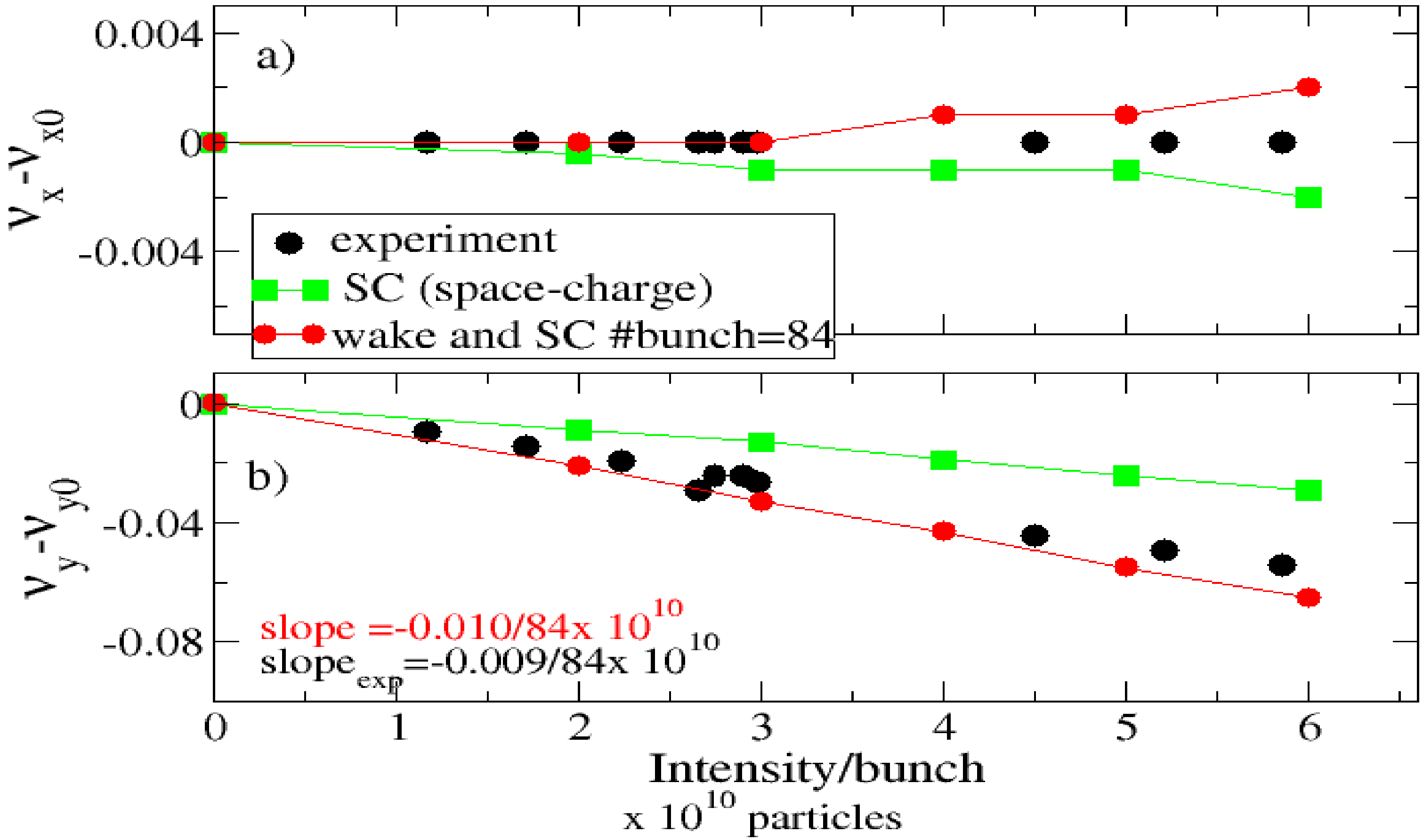
The difference between the ultrarelativistic approximation and the non-ultrarelativistic impedance increases with increasing frequency and the chamber radius

Longitudinal wake

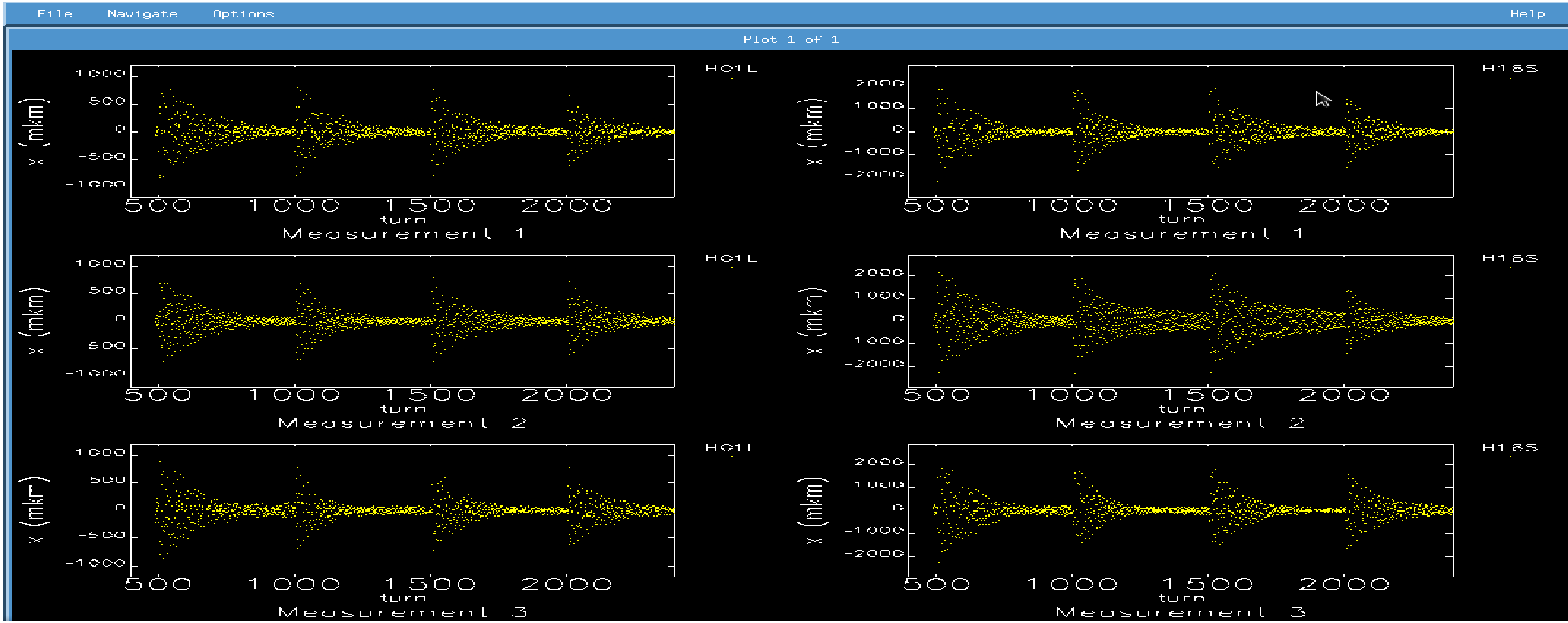


- In the small distance region, |z|< ≈0.1m the ultrarelativistic approximation (red) fails
- Behind the source, the ultrarelativistic approximation overestimates the true wake (black).
- The wake is non-zero in front of the source

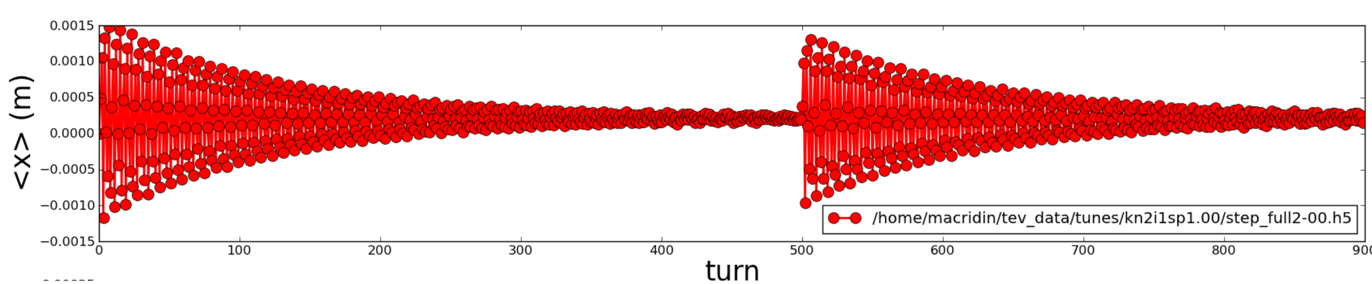
Coherent tune shift:



Horizontal kick attenuation (preliminary):



Experiment



Synergia simulations, 1 bunch, 900 turns