Status of PY-ORBIT: Benchmarking and Noise Control in PIC Codes

J. Holmes, S. Cousineau,

and A. Shishlo

54th ICFA Beam Dynamics Workshop
HB2014
Michigan State University
November 12, 2014

U.S. DEPARTMENT OF ENERGY

OAK RIDGE NATIONAL LABORATORY
MANAGED BY UT-BATTELLE FOR THE DEPARTMENT OF ENERGY

Python ORBIT (PY-ORBIT)

- PY-ORBIT is a collection of computational beam dynamics models for accelerators, designed to work together in a common framework.
- It was started as a "friendly" version of the ORBIT Code, written using publicly available supported software.
  - Users interface with code via Python scripts.
  - Computationally intensive code in C++ (mostly, PTC is in Fortran).
  - Wrappers make C++ routines available to users at Python level.
  - Uses MPI for multiprocessing.
- PY-ORBIT source code is publicly available via Google Codes:
  - svn checkout https://py-orbit.googlecode.com/svn/trunk py-orbit --username youraccount@gmail.com
  - It is not difficult to develop your own extensions to PY-ORBIT. We welcome responsible participation in developing PY-ORBIT models.
- PY-ORBIT people at present:
  - Owners: Andrei Shishlo (ORNL), Sarah Cousineau (ORNL), Jeff Holmes (ORNL)
  - Committers: Sabrina Appel (GSI), Oliver Boine-Frankenheim (GSI), Hannes Bartosik (CERN), Timofey Gorlov (ORNL)
  - Additional user: Sasha Molodozhentsev (KEK)

What Does PY-ORBIT Do?

- Most things that ORBIT does for rings and transfer lines
  - Single particle tracking
    - Native symplectic tracker
    - PTC tracking
    - 3D field tracker
    - Linear tracking with matrices (coming)
  - Space charge
    - Longitudinal
    - 2D potential and direct force
    - Full 3D (not parallel) and 3D ellipses (mostly for linacs and transfer lines)
    - 2.5D (Recently added by Hannes Bartosik)
  - Impedances
    - Longitudinal
    - Transverse dipole (partially completed)
  - Injection, painting, RF cavities, collimation, apertures
- Linac Modeling
  - RF cavities
  - Magnets
  - Full 3D space charge (not parallel) and 3D ellipses
- Laser Stripping, Nonlinear Optics, Electron Cloud

Appealing Features

- Entire source code is available. Uses standard Python, C++, and Fortran (PTC). Extra libraries only for FFTs and PTC.
- User is free to develop specialized or extended models to suit his/her own needs.
- With permission of owners, users' models can be incorporated into public version.
- Many examples demonstrate use of models in scripts. Some documentation in Google Code wikis.
- Bunch class is easily extendable:
  - Basic bunch has macroparticle coordinates in 6D.
  - User can add various properties:
    - Particle ID tag
    - Spin
    - Species, ionization number, excited state, etc.

Remaining Issues

- Code is not yet complete. Missing:
  - Linear and second order tracking with matrices (coming)
  - Transverse dipole impedance (in progress)
  - Alignment errors
  - Electron Cloud
  - Other?
- Documentation is incomplete:
  - Although lots of examples illustrate use of modules and making of scripts,
  - Some models are documented in Google Code wikis, but the rest needs to be done.
- No “professional” full-time support:
  - PY-ORBIT development has been carried out by working accelerator physicists (owners) trying to model and solve problems.
  - We all have to work at our “day jobs”.

Space Charge Models for Long Times

- Space charge physics has been successfully incorporated into computational particle tracking studies of linacs, transfer lines, accumulator rings, and rapid cycling synchrotrons.
- Computations for these machines all involve tracking particles on short to moderate time scales.
- With the advent of higher beam intensities, calculations incorporating space charge effects must now be undertaken for storage rings.
- Storage ring calculations require following beams for far longer times than are necessary for linacs, accumulator rings, or rapid cycling synchrotrons.
  - This will place more severe requirements on the speed and accuracy of the physics models.
  - Progress for modeling collective effects, such as space charge, is necessary.
- Issues in choosing a model:
  - Representation of beam
  - Numerical properties (discretization)
  - Computational requirements vs available resources

State of the Art a Decade Ago

S. Cousineau, J. A. Holmes, J. Galambos, A. Fedotov, J. Wei, and R. Macek,
Physical Review Special Topics – Accelerators and Beams 6, (2003) 074202 .

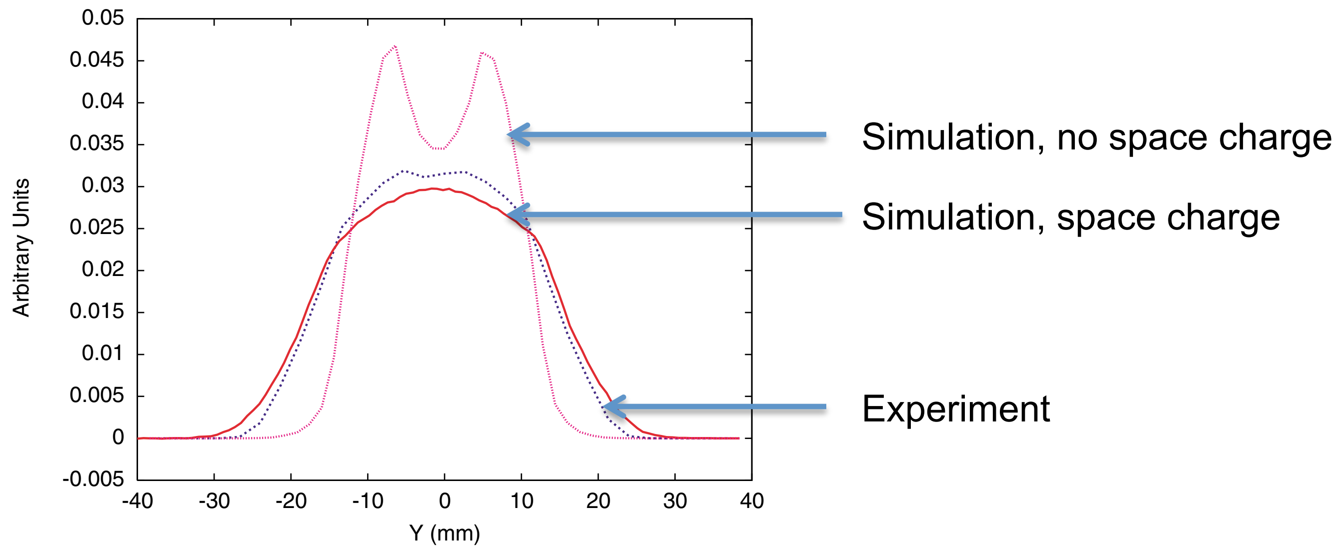


FIG. 1. (Color) Benchmark of vertical experimental profile measurements with PIC simulations. The solid (red) curve is the experimental profile. The dotted (blue) curve is the PIC result with space charge. The (pink) dashed line is the PIC result without space charge.

Transverse profiles: experiment and simulation with and without space charge in PSR ring.

Choice of Model

- **Do we calculate space charge from assumed simple beam distribution or directly from tracked particle bunch?**

- **Representation of beam**
  - **Assumed simple distributions will miss details, such as real beam profiles and nonuniformites, and cannot accurately depict many processes, such as injection into accumulator rings.**
  - **Direct use of the tracked macroparticle bunch can, in principle, faithfully represent real beam distributions.**

- **Numerical Properties**
  - **With assumed simple distributions, applied forces are smooth and collisions eliminated. Numerical noise comes from time discretization only.**
  - **Direct use of the macroparticle distribution for space charge force calculation involves discretization errors in space and time that lead to numerical diffusion and emittance growth. The implications of controlling this will be discussed.**

- **Computational requirements**
  - **Forces from assumed simple distributions can be evaluated quickly, and it may be necessary to track relatively few particles.**
  - **Because of numerical issues, the direct use of macroparticles is typically more demanding on computer resources. Even so, in previous work on short to medium time scales, accurate calculations could be performed on workstations and small clusters in reasonable time.**

- **Therefore, methods of choice have used the tracked distribution directly to obtain the force.**
  - **Particle-in-cell (PIC) and fast multipole (FMM) methods are of order *O(~M)*, where *M* is the macroparticle number.**
  - **Methods come in various dimensionality (1D longitudinal, 2D transverse, 2.5D transverse and optionally longitudinal, or full 3D).**
  - **Complexity increases with dimension. Use simplest model containing desired physics.**

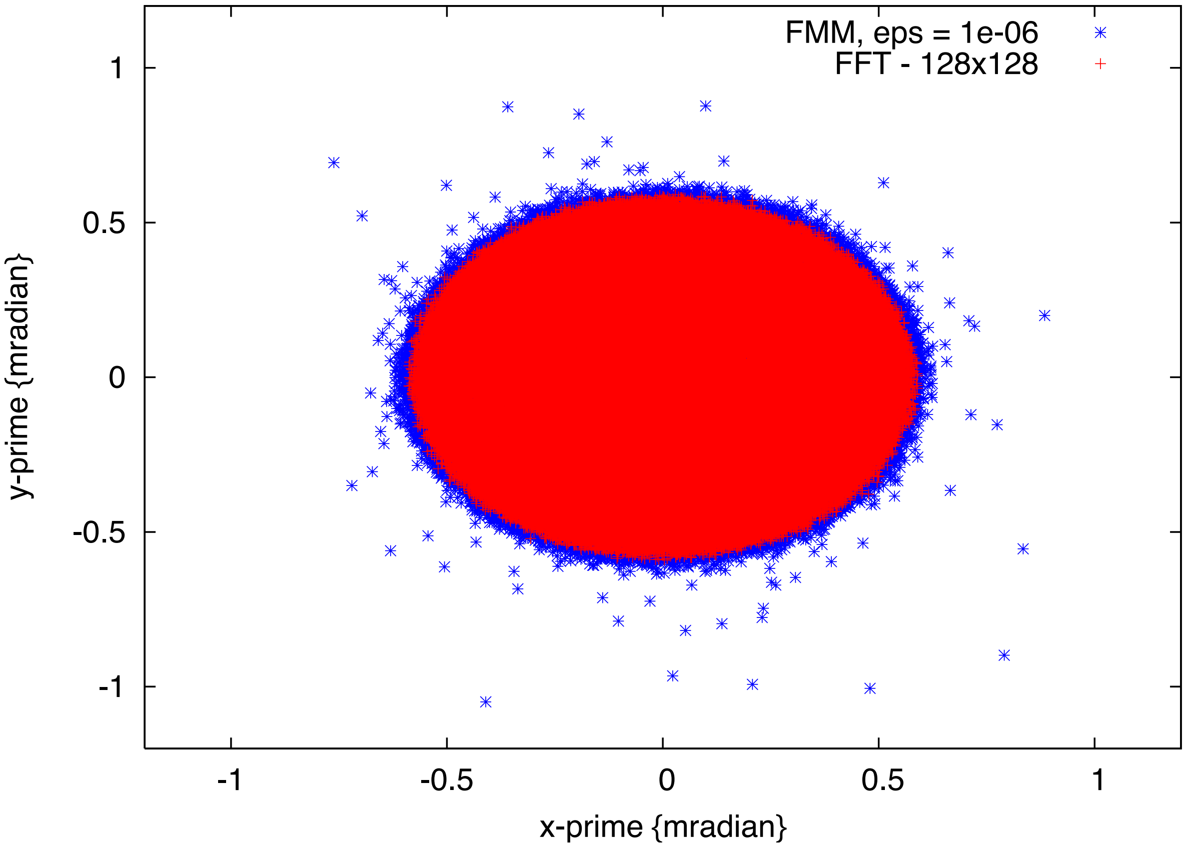
Bunch-Based Forces in Time

- 2D or 2.5D bunch-based models are appropriate candidates for long time scale calculations (storage rings). Valid for long bunches $L_{Bunch} >> r_{BeamPipe}$.
- Most calculations use FFT-based PIC methods.
  - Discretization errors due to time, coarseness of distribution , gridding.
- An alternative is the Fast Multipole Method (FMM)
  - Discretization errors due to time, coarseness of distribution, but not gridding.
  - Accurate in principle to machine precision.
- Based on the above properties, we programmed and tested fast multipoles.
- References on numerical solution of Poisson's equation:
  - R. W. Hockney and J. W. Eastwood, "Computer Simulation Using Particles", Institute of Physics Publishing, Bristol, 1988.
  - J. Demmel, NSF-CBMS Short Course on Parallel Numerical Linear Algebra, especially Lectures 24-27, http://www.cs.berkeley.edu/~demmel/cs267-1995/
  - L. Greengard and V. Rokhlin, "A Fast Algorithm for Particle Simulations", Journal of Computational Physics 73, (1987), 325.

Fast Multipoles: The Method

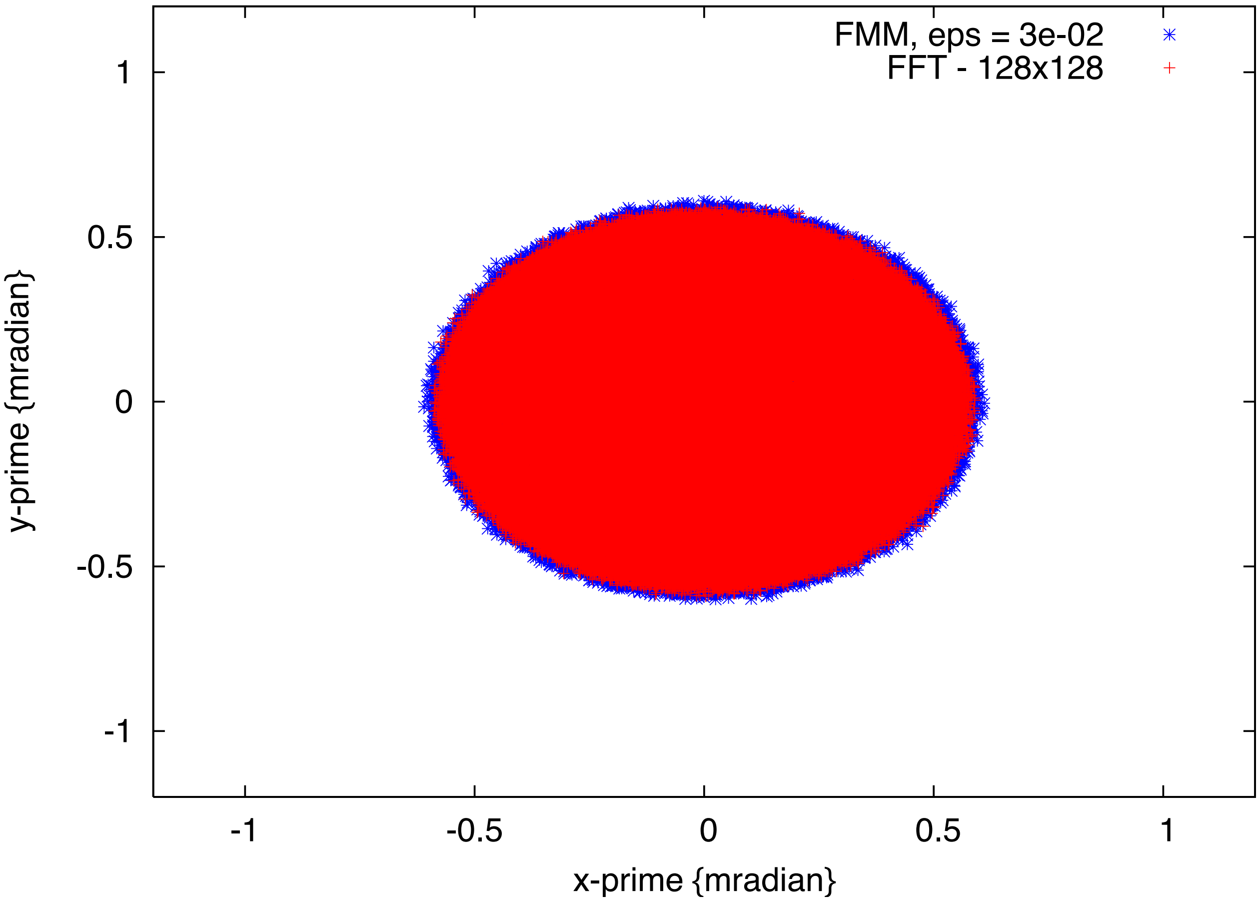
- Create series of levels, each with grid of $2^n \times 2^n$ square cells, n = N->2.
  - Grids of successive levels align.
  - Each cell of level n < N contains 4 child cells from n+1 level.
- Grid of cells must cover beam and boundary (if specified).
- At finest level, N, bin macroparticle charges to center of containing cell.
  - Expand force/potential of each charge as series about center.
  - Can keep as many terms as necessary – machine precision in principle.
  - Series converges outside of cell corner radii.
- Successively shift multipole expansions to centers of parent cells.
  - n = N-1 -> 2.
  - Accumulated, shifted multipole series converge outside of parent cells.
- Expand multipole series from non-adjacent parent cells as Taylor series about center of cells.
  - Again, can keep as many terms as necessary – machine precision in principle.
  - n = 3 -> N.
  - Pick up contributions of non-adjacent sibling cells (same level) that were missed.
- At finest level, N:
  - Apply Taylor series kicks to macroparticles.
  - Sum force kicks from same and adjacent cell macroparticles pairwise. Smoothing?!
- Treat boundary
  - Calculate beam potential at boundary points as sum of Taylor series and adjacent Cell macroparticles.
  - Use method of Fred Jones (ACCSIM) to add kicks due to conducting wall BC.
- Parameters: N levels, K terms, eps (smoothing parameter for pairwise).
  - Others: boundary, longitudinal discretization, etc.

Fast Multipoles vs FFT



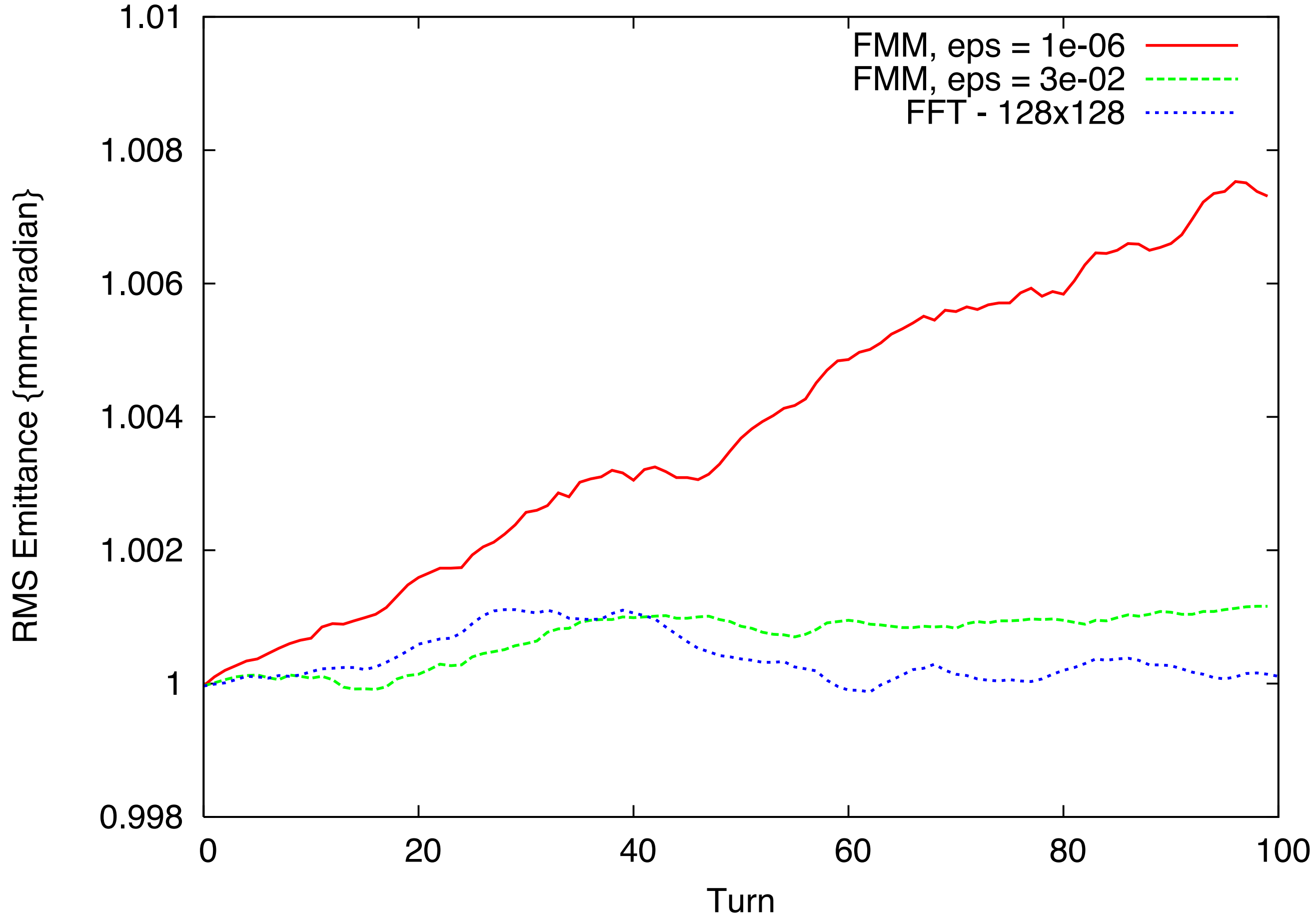
No smoothing -> pairwise collisions. Smaller time steps, more macroparticles -> increase computer time.

Fast Multipoles vs FFT



Some pairwise smoothing -> less Scattering agreement with FFT.

Fast Multipoles vs FFT



RMS emmitance: FFT and FFM with varied smoothing.

Fast Multipoles vs FFT

Property	FFT	FMM
Numerical order (work)	n	n
Poisson solution	Grid and interpolate	Up to machine precision
Poisson domain noise	Particle distribution Grid size Interpolation scheme	Particle distribution Expansion order Binary smoothing parameter
Longitudinal noise	Grid size Interpolation scheme Particle distribution	Grid size Interpolation scheme Particle distribution
Time step discretization noise	Yes	Yes Anomalous pairwise forces

Based on initial experiments with 10^5 macroparticles, FMM takes anywhere from 10-50 times more computer time, depending on the number of levels and order of expansion, than FFT with 128×128 cells.

Numerical Discretization Effects

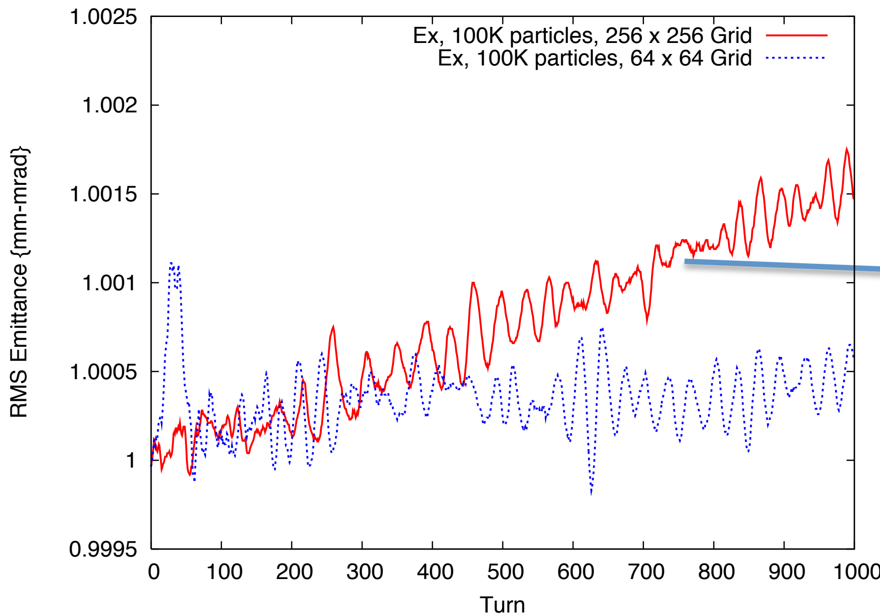
- Discretization in time and graininess of the charge/current distribution -> diffusion.
- Spatial meshes and smoothing also effect discretization.
- Boine-Frankenheim, Hofmann, Struckmeier, and Appel studied numerically generated collisions and emittance growth in 2D space charge calculations for FODO lattices.
  - Based on earlier theoretical studies using Fokker-Planck analysis.
  - Derived an empirical scaling law for collisionality, or entropy or emittance growth, in space charge calculations.

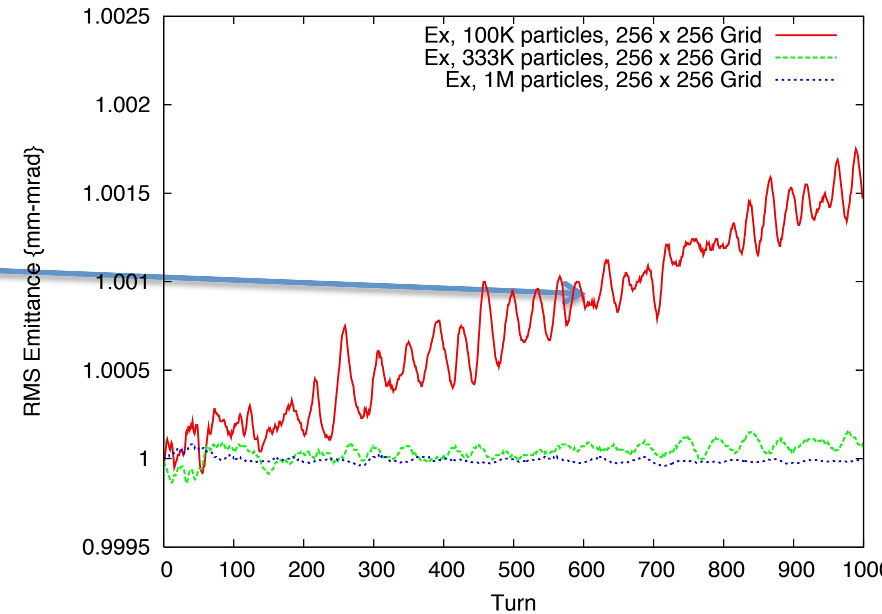
$$\nu \propto \frac{N^2}{M} \times \left(1 - \frac{\Delta}{\lambda}\right)$$

- ν = collision frequency, *N* = beam intensity (number of real particles), *M* = number of macroparticles, Δ = smoothing size (grid spacing), λ = cutoff scale length.
- Numerical entropy growth is proportional to beam intensity squared, is inversely proportional to number of macroparticles, and is decreased by smoothing.
- References on collisions and numerically-induced emittance growth:
  - Jurgen Struckmeier, Phys. Rev. E 54, 830, (1996).
  - O. Boine-Frankenheim, I. Hofmann, J. Struckmeier, S. Appel, Nucl. Instr. and Meth. in Phys.Research A., in press.

Observations of the Empirical Scaling

- **Numerical diffusion in PIC codes leads to emittance growth in time. The growth is dependent on the number of macroparticles and the grid spacing.**





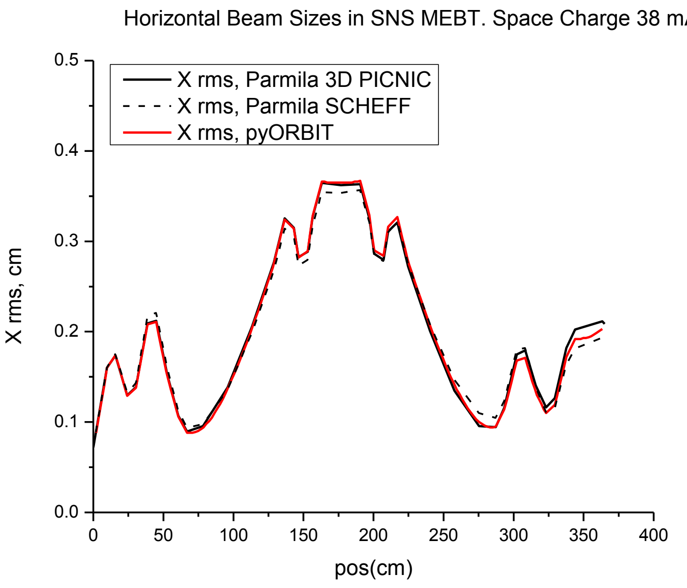
100K particles, no space charge and space charge with various grids.

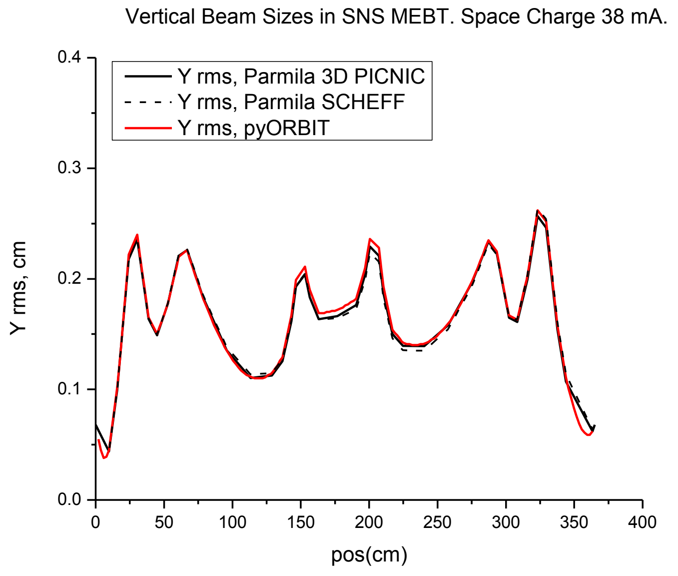
256x256 grid and various numbers of particles.

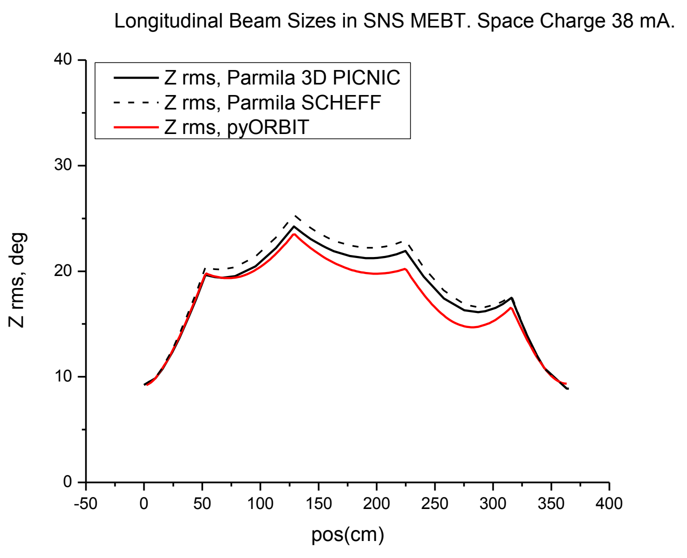
What to Do to Reduce Collisions?

- Refinement: More macroparticles, smaller time steps…
  - Feasible to some extent with the help of modern computers: large clusters, supercomputers, GPUs, etc.
  - Still expensive and “brute force”.
- Reduce discretization effects
  - For FFT solvers, try different binning or smoothing algorithms for distributing space charge from the numerical particle distributions to the grid. Like low pass filters, can alleviate grid and distribution discretization effects.
  - Simplified or analytic distributions, based on statistically calculated parameters of the macroparticle distribution, provide another type of low pass filter to reduce grid and distribution discretization effects. Such methods can be used to evaluate space charge forces quickly.
  - Questions for the latter approaches: How much physics is lost in the simplification? How important is that physics? To what problems are analytic approaches applicable.
- We have just begun exploring these approaches in the ORBIT Code.

MEBT 38 mA. PARMILA vs. pyORBIT







Water Bag 3D, 38 mA
2,000 macro-particles for Ellipse SC
20,000 macro-particles for 3D FFT
32 x 32 x 32 grids
Bunchers RF is on, 90^0 phases

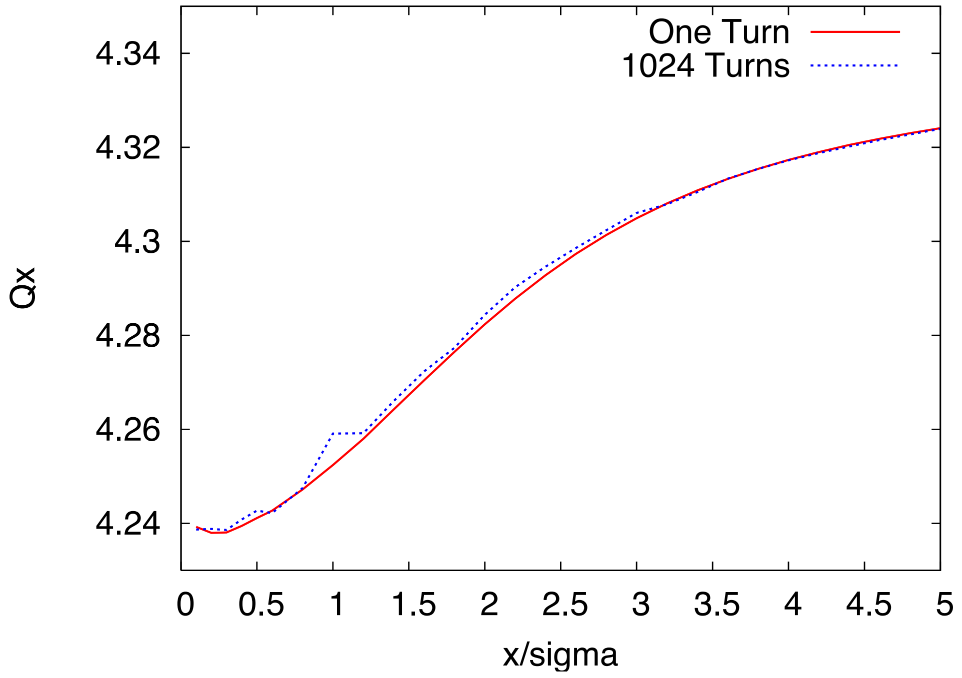
Timing:
Parmila SHEFF – about 4 sec
PARMILA 3D PICNIC – 8 sec
pyORBIT 1 Ellipsoid – 1.6 sec

Benchmarking

- ORBIT has been extensively benchmarked, including all its space charge models, through its 15 year tenure.
  - Benchmark tests include comparisons with analytic results, with experimental observations, and with other codes.
  - PY-ORBIT has been thoroughly benchmarked with ORBIT as models have been completed and tested.
- Formal benchmark tests
  - An excellent suite of benchmarks on several time scales that involves space charge induced resonance trapping can be found on Giuliano Franchetti's web site at http://web-docs.gsi.de/~giuliano/
    - ORBIT has successfully completed 8 of the 9 benchmark tests on this site. One case is under study.
    - Some of the longer time scale tests, as posed, are accompanied by significant numerically-induced emittance growth.

Numerical Diffusion Effects Are Apparent in Individual Particle Behavior

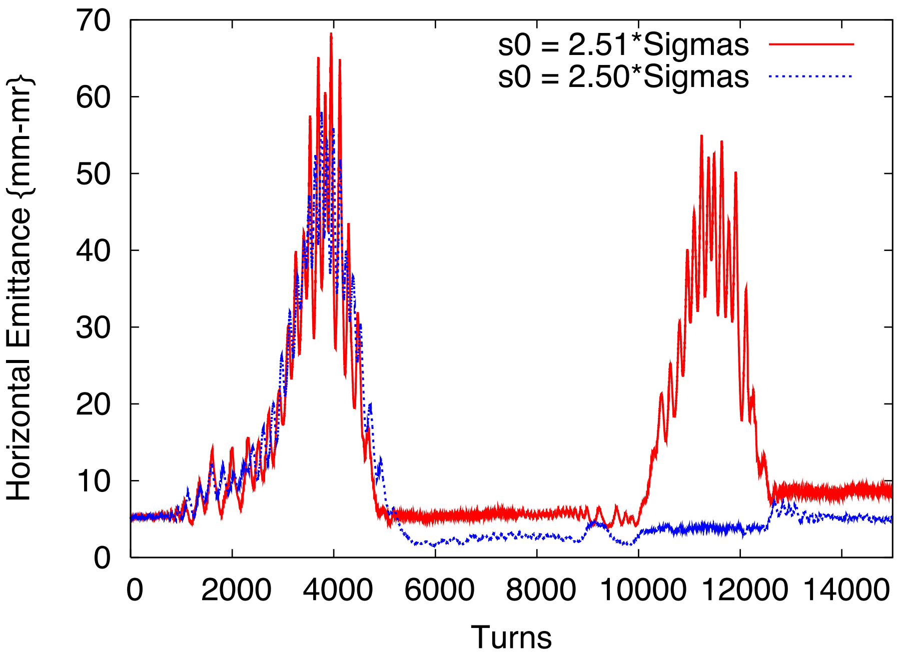
- **Individual particle behavior is difficult predict, due to numerically-induced collisions.**



Calculated tunes of individual particles are sensitive their diffusion.

Numerical Diffusion Effects Are Apparent in Individual Particle Behavior

- **Individual particle behavior is difficult predict, due to numerical noise.**



Individual particle orbits diffuse in time.

For long times, convergence criteria are hard to satisfy.

2.5D PIC Space Charge Model

- **"Comparison Between Measurements, Simulations, and Theoretical Predictions of the Extraction Kicker Transverse Dipole Instability in the Spallation Neutron Source," J.A. Holmes, S. Cousineau, V. Danilov, and L. Jain, *Phys. Rev. Special Topics – Accelerators and Beams* 14, (2011), 074401 .**

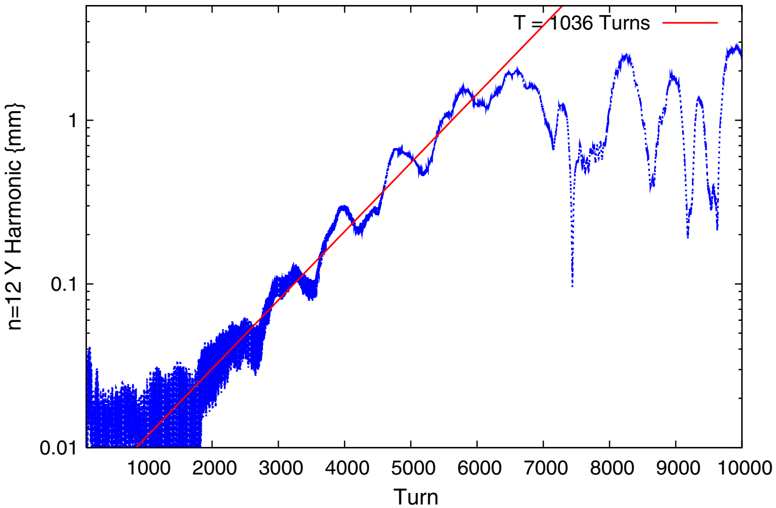


FIG. 8. Vertical $n = 12$ harmonic (in blue) versus turn number in the ORBIT extraction kicker instability simulation. The red line depicts an exponential growth time of 1036 turns.

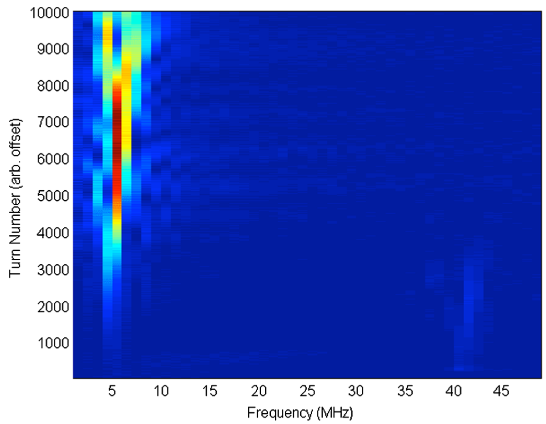


FIG. 7. Evolution of experimental turn-by-turn vertical harmonic spectrum of the extraction kicker instability.

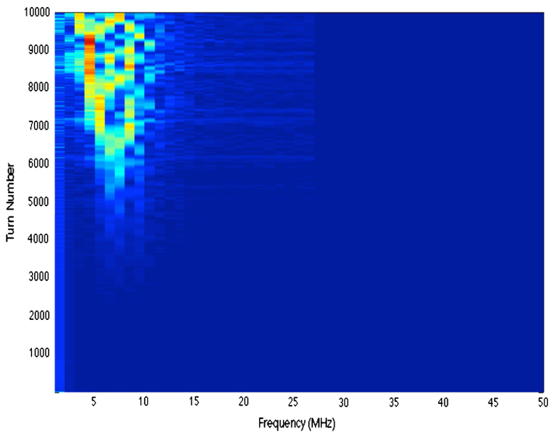


FIG. 9. Evolution of simulated turn-by-turn vertical harmonic spectrum of the extraction kicker instability.

Left:	Growth of dominant harmonic matches measured and predicted values.
Right:	Experimental (top) and simulated (bottom) spectral evolution.

Back Up Slides:

- **Back Up Slides:**

Longitudinal PIC Space Charge Model

- "Space-Charge-Sustained Microbunch Structure in the Los Alamos Proton Storage Ring", S. Cousineau, V. Danilov, J. A. Holmes, and R. Macek, *Physical Review Special Topics – Accelerators and Beams* 7, (2004), 094201.

Experiment

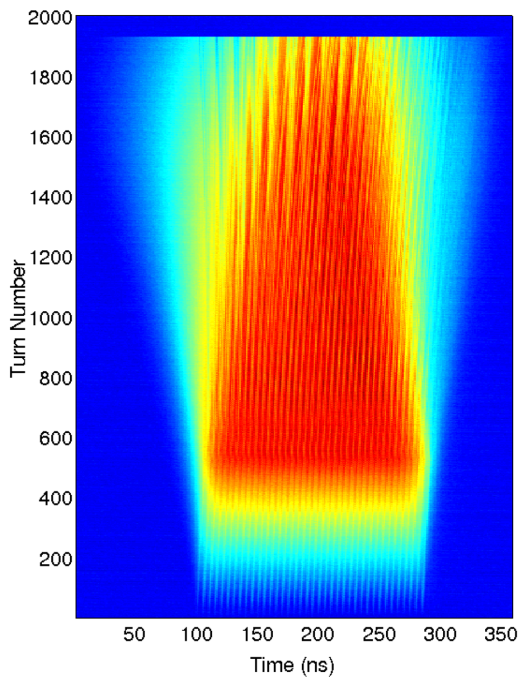


FIG. 1. (Color) Waterfall plot of the wall current monitor signal for the chopped beam experiment. The time in a single pulse is plotted on the horizontal axis, and the turn number is plotted on the vertical axis. Beam injection ends at 559 turns. The high frequency variations in color indicate the microbunch structure of the beam. The beam gap region is displayed in blue.

Simulations

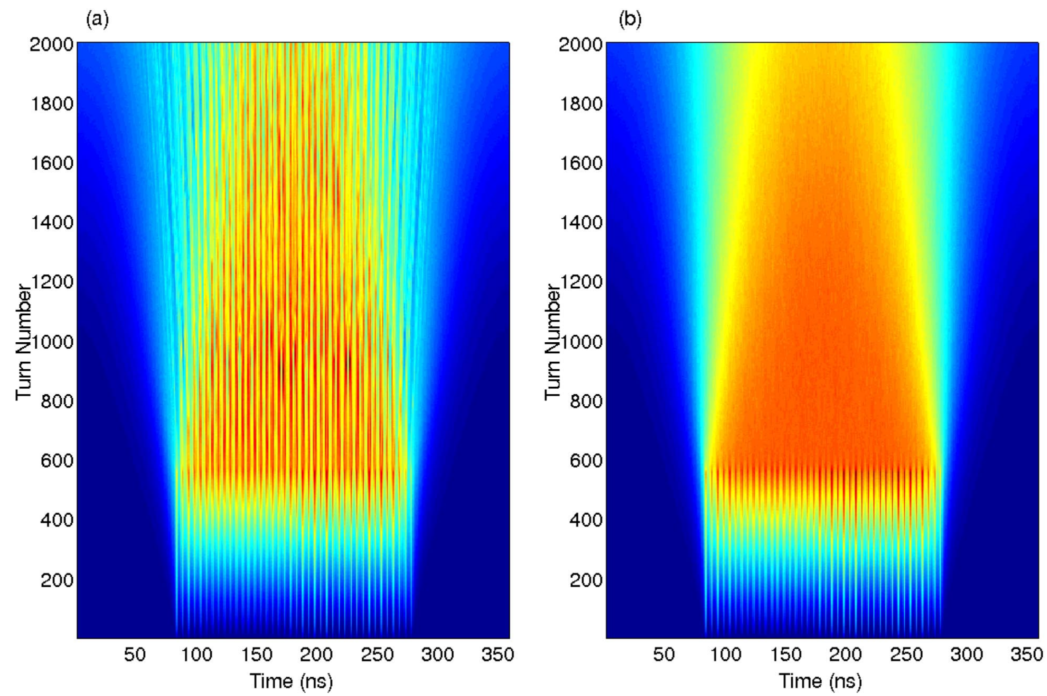


FIG. 5. (Color) (a) Waterfall plot of simulated wall current monitor signal for chopped beam experiments. Space-charge effects are included. (b) Same as in (a) but with no space charge included.

Space Charge No Space Charge

2D PIC Space Charge Model

- "Comparison of Simulated and Observed Beam Profile Broadening in the PSR and the Role of Space Charge", J. D. Galambos, S. Danilov, D. Jeon, J. A. Holmes, D. K. Olsen, F. Neri, and M. Plum, Physical Review Special Topics – Accelerators and Beams 3, (2000), 034201.
- "Resonant Beam Behavior Studies in the Proton Storage Ring", S. Cousineau, J. A. Holmes, J. Galambos, A. Fedotov, J. Wei, and R. Macek, Physical Review Special Topics – Accelerators and Beams 6, (2003) 074202 .

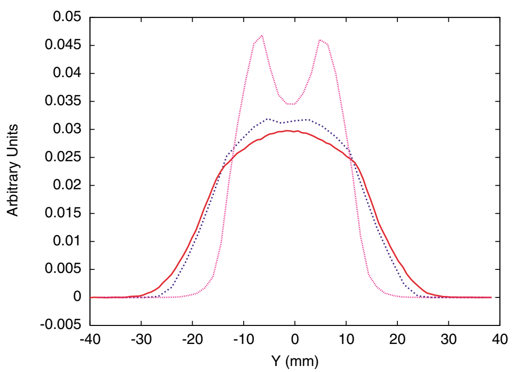


FIG. 1. (Color) Benchmark of vertical experimental profile measurements with PIC simulations. The solid (red) curve is the experimental profile. The dotted (blue) curve is the PIC result with space charge. The (pink) dashed line is the PIC result without space charge.

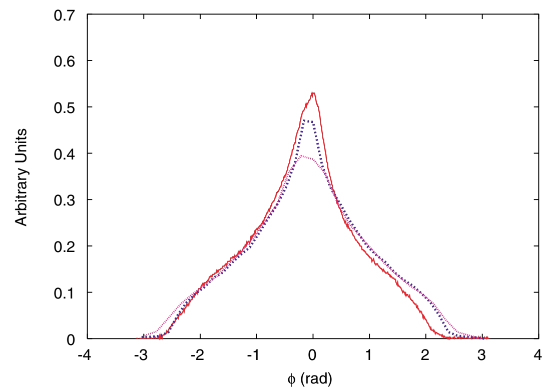


FIG. 2. (Color) Benchmark of longitudinal profile measurements with PIC simulations. The solid (red) curve is the experimental profile. The dashed (blue) curve is the PIC result with longitudinal space charge and impedance. The (pink) dotted line is the PIC result with longitudinal space charge but without other impedance.

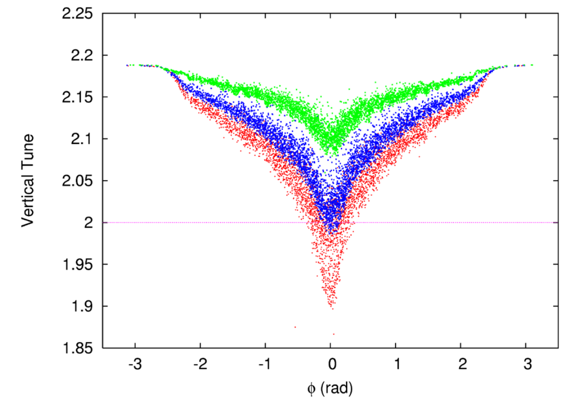


FIG. 5. (Color) Incoherent particle tune shifts versus longitudinal coordinate after the accumulation of the full intensity PSR beam (red points), the half intensity beam (blue points), and the quarter intensity beam (green points), obtained from PIC simulations.

Transverse profiles: experiment and simulation with and without space charge

Longitudinal profiles: note bunch factor.

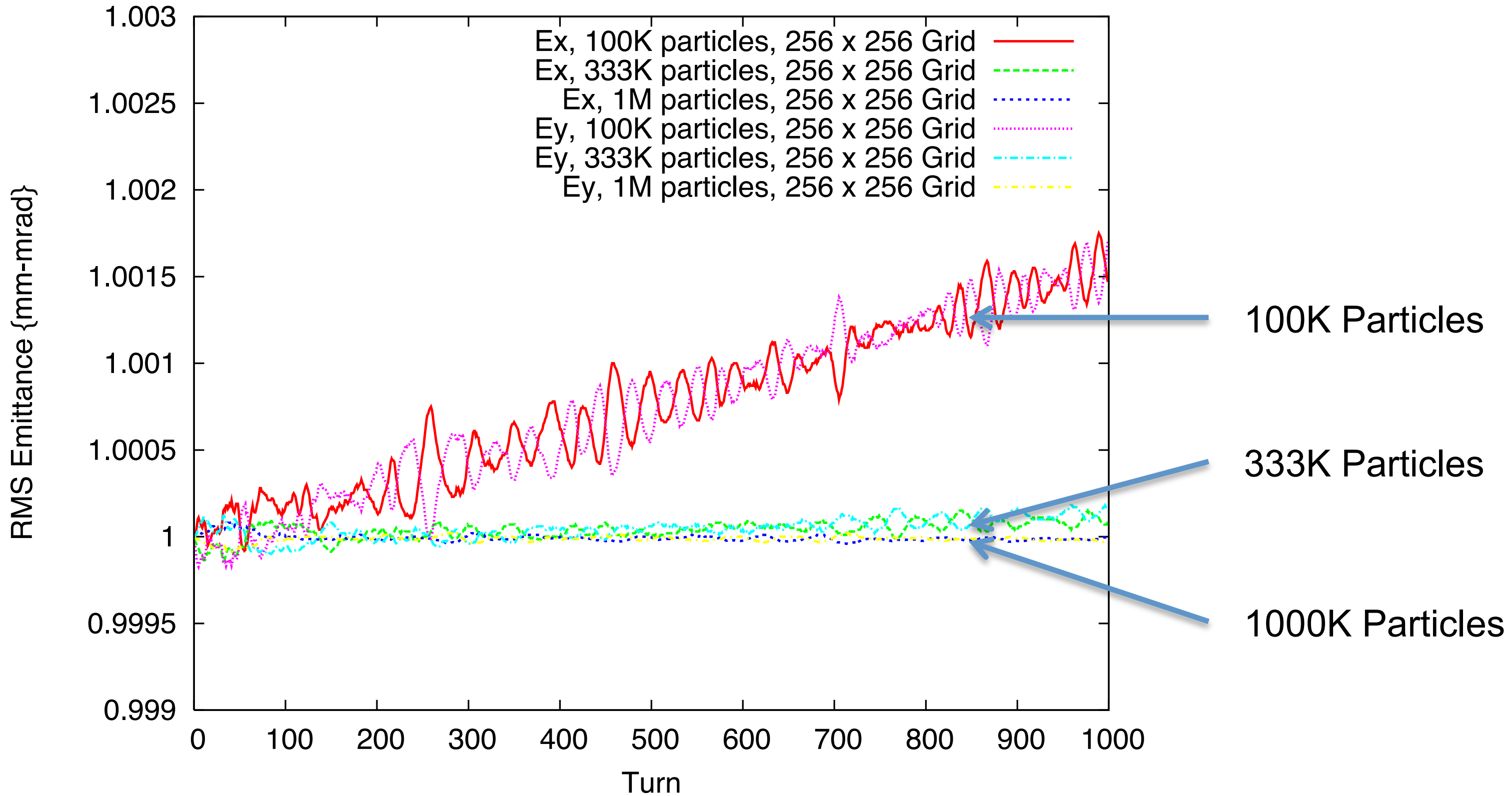
Incoherent tunes versus longitudinal position.

Choice of Independent Variable

- Time or position.
  - Time
    - Natural choice for evaluation of space charge.
    - Requires knowledge of accelerator environment for the entire bunch -> complicated for long bunches spanning multiple lattice elements.
  - Position
    - Since all particles are at the same place, relative transverse positions of longitudinally separated particles are incorrect for space charge evaluation. The particles exist at different times. This effect grows with longitudinal particle separation, but forces decrease with separation.
    - Convenient from the standpoint of accelerator environment, especially for long bunches.
    - To work, space charge forces must depend mostly on local, rather than distant, longitudinal separation.

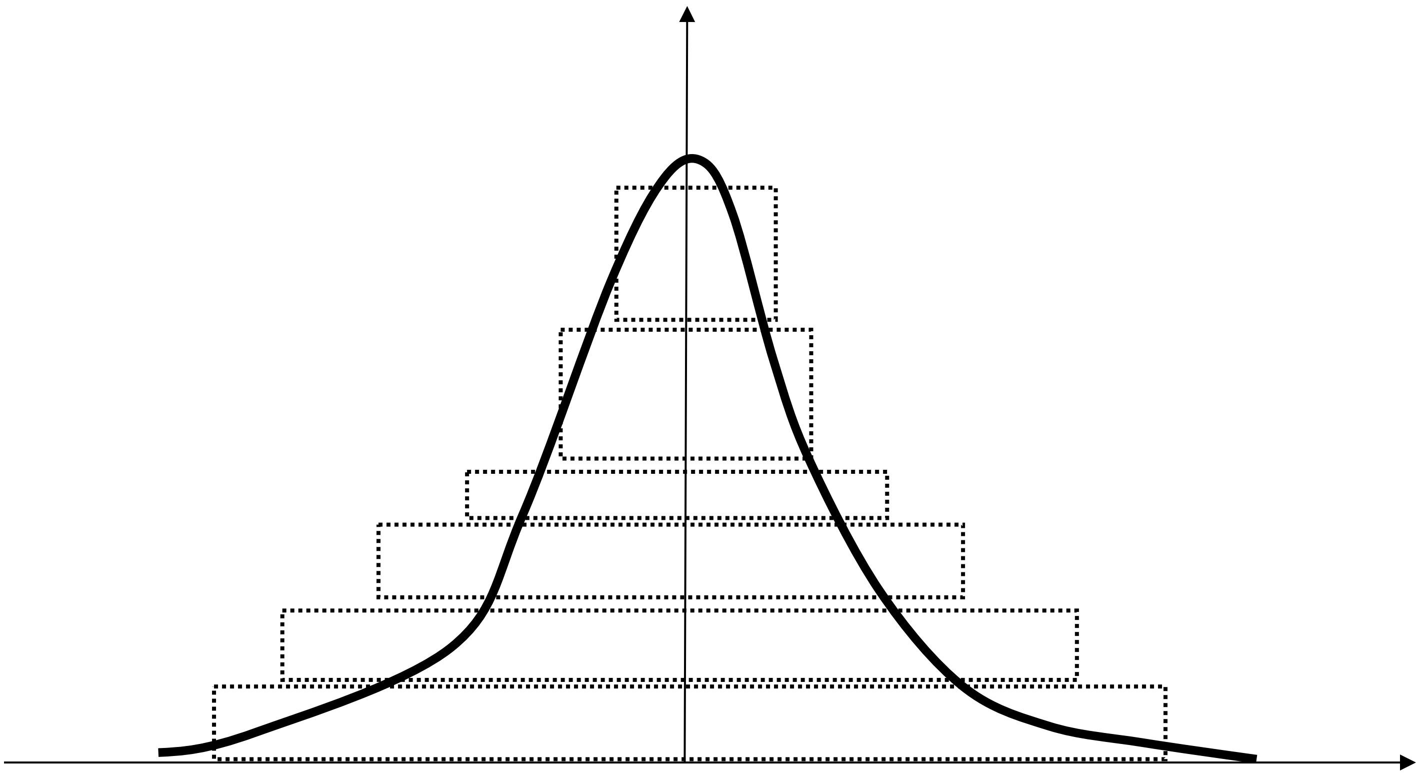
Applicability of Model Depends on Information Sought

- **Bulk quantities, such as RMS emittances, can be obtained for short to medium time scales with moderate numbers of particles, but numerical diffusion leads to slow growth over long times. Even for short times, it is necessary to be sure the numerics converge.**



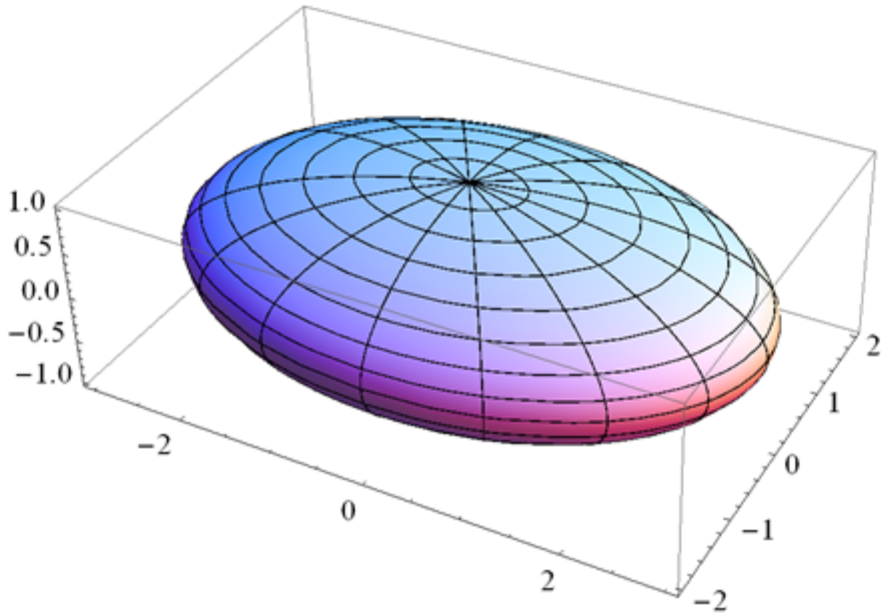
256x256 grid with various numbers of particles.

Set of Ellipsoids



pyORBIT Space Charge Solver can use arbitrary number of ellipsoids

Electric Field of Uniformly Charged Ellipse



$$\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1$$

$$\phi(\mathbf{x}) \equiv \int_{V_0} \frac{\rho}{|\mathbf{x} - \mathbf{x}'|} \, dV(\mathbf{x}')$$

$$\phi(x, y, z) = \pi \, abc \, \rho \int_0^\infty \left[1 - \frac{x^2}{a^2+s} - \frac{y^2}{b^2+s} - \frac{z^2}{c^2+s}\right] \frac{ds}{\sqrt{\varphi(s)}}$$ inside

$$\phi(x, y, z) = \pi \, abc \, \rho \int_\lambda^\infty \left[1 - \frac{x^2}{a^2+s} - \frac{y^2}{b^2+s} - \frac{z^2}{c^2+s}\right] \frac{ds}{\sqrt{\varphi(s)}}$$ outside

$$\varphi(s) \equiv (a^2+s)(b^2+s)(c^2+s) \qquad f(s) \equiv \frac{x^2}{a^2+s} + \frac{y^2}{b^2+s} + \frac{z^2}{c^2+s} - 1$$

where λ is the greatest root of the equation $f(s) = 0$