

Computation Lecture #1
June 12, 2006
Computation for Intense Beams

USPAS - "Boston"
Rami Kishek,
University of Maryland

Rm. 206
ramiak@umd.edu

Displaying these Slides Major Team Effort

Thanks to:

- My wife: sent copies from home
- John Barnard: using memory stick
- Diktys Stratakis: Used laptop
- Christos Papadopoulos: Using Laptop
- Steve Lund: Using Notes

Suggested Bible

Charles K. Birdsall and A. Bruce Langdon, Plasma Physics via Computer Simulation, (McGraw Hill 1985). *Chaps. 1 and 2 suffice.*

3

All you need to know about beams ...

Maxwell's Equations:

$$\bar{\nabla} \times \bar{E} = -\frac{\partial \bar{B}}{\partial t} \qquad \bar{\nabla} \cdot \bar{E} = \frac{\rho}{\epsilon_0}$$

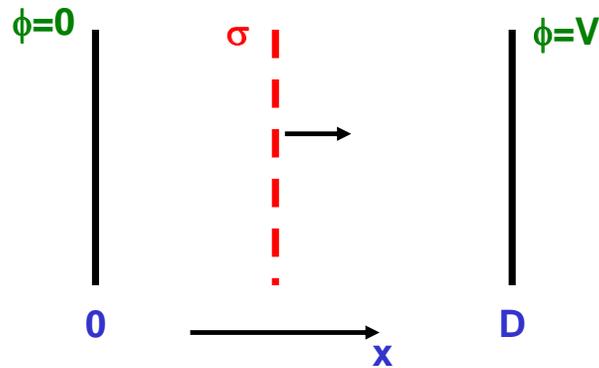
$$\bar{\nabla} \times \bar{B} = \frac{1}{c^2} \frac{\partial \bar{E}}{\partial t} + \mu_0 \bar{J} \qquad \bar{\nabla} \cdot \bar{B} = 0$$

Lorentz Force Law:

$$\bar{F} = \frac{d}{dt}(\gamma m \bar{v}) = q(\bar{E} + \bar{v} \times \bar{B})$$

4

Some simple cases don't need simulation



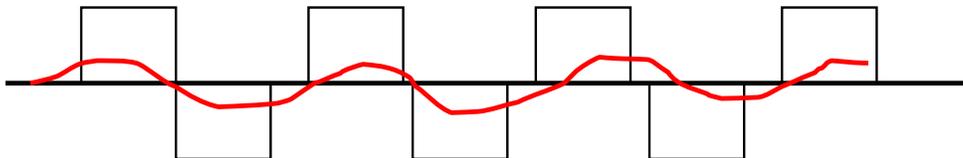
σ is surface charge density

5

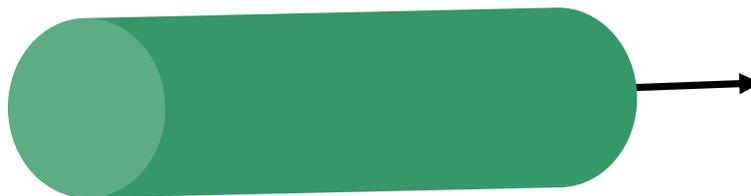
Theoretical Approach (this course, Reiser's book)

Simplify problem sufficiently to solve, e.g.:

- Single particle wandering through known lattice



- “Cylinder” beam with uniform density distribution

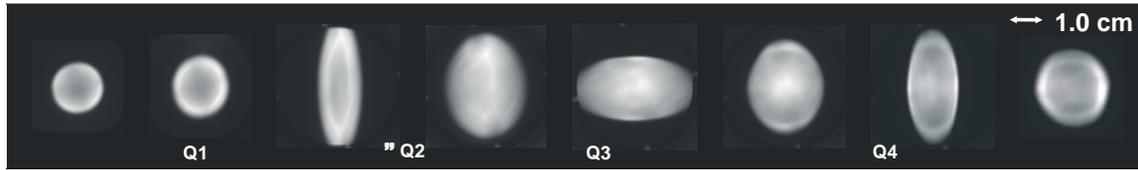


Even “analytical” cases like this often require computers to solve (numerical integration).

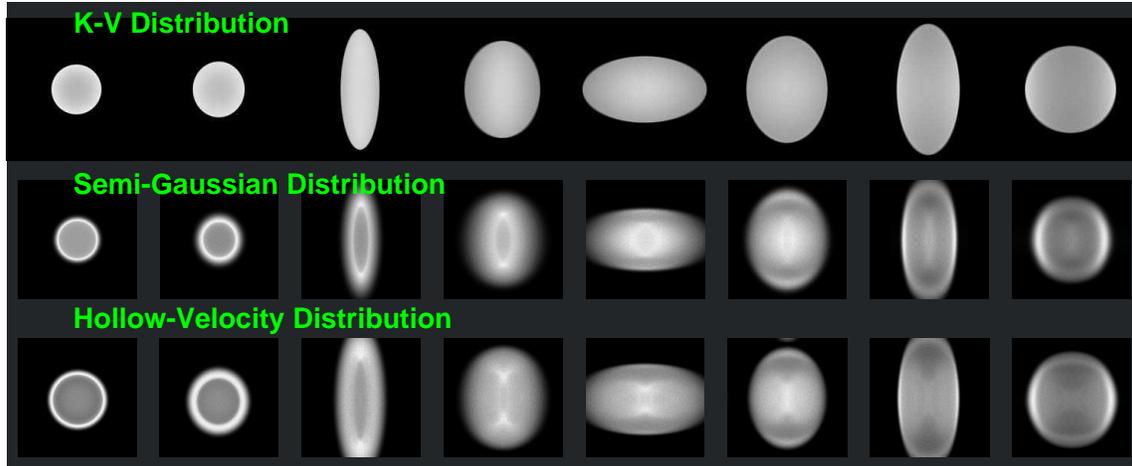
6

Problem 1: Real Beams are not K-V

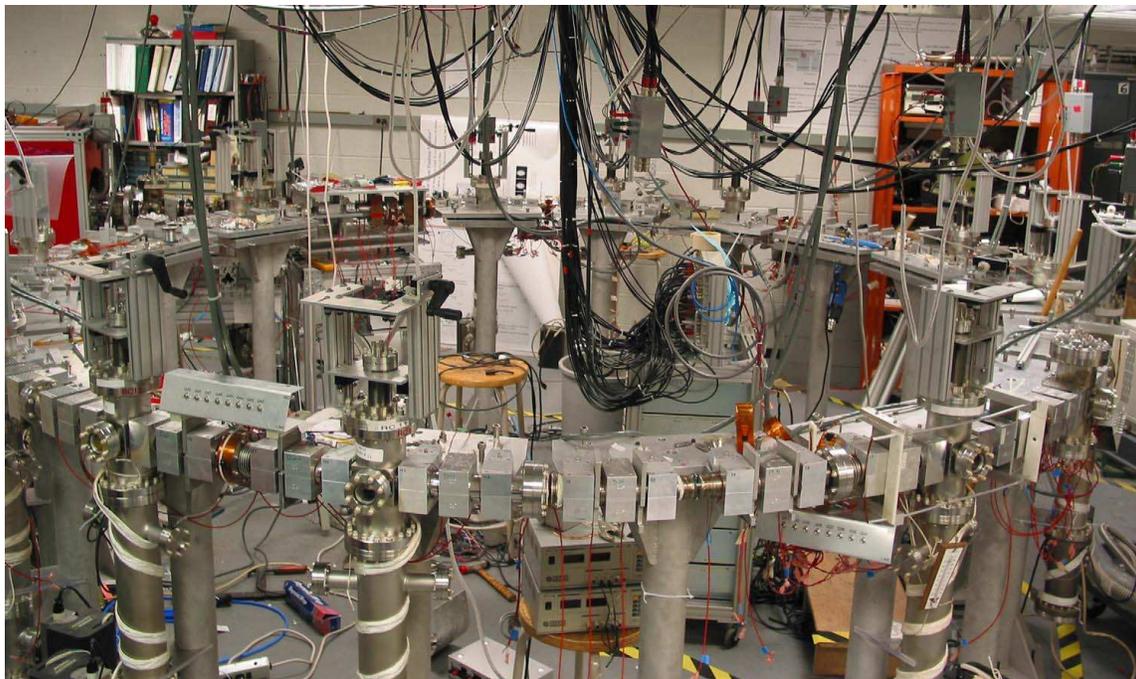
Experiment (100 mA) (top) [Bernal]



WARP Simulation (below) [Kishek]



Problem 2: Real Accelerators are Very Complex

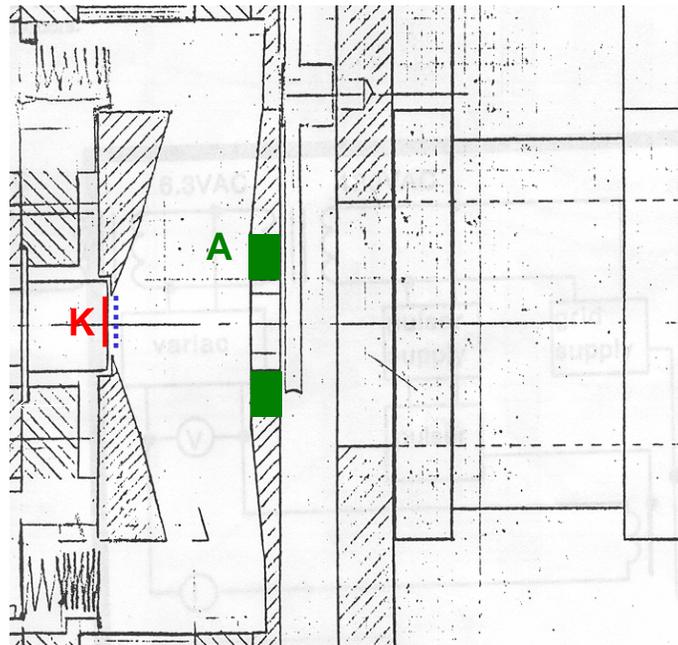


The University of Maryland Electron Ring (UMER)

Problem 3: Realistic Geometries May be Difficult

Electron Gun

Grid



9

Problem 4: Real Accelerators are Expensive



10

Computation has Many Advantages:

1. Verification of simpler analytic models.
Layers of complexity can be added or removed at will from the computer model.
2. Understanding and interpreting experimental results.
Simulation is much easier (and cheaper) to set up and perform than experiment.
3. To facilitate design of large, expensive accelerators.
The computer model can be used to study the physics of a large accelerator well before such an accelerator can be constructed.

11

Simulation

Central issue: What happens to charged particles in an environment of electric and magnetic fields?

For real beams:

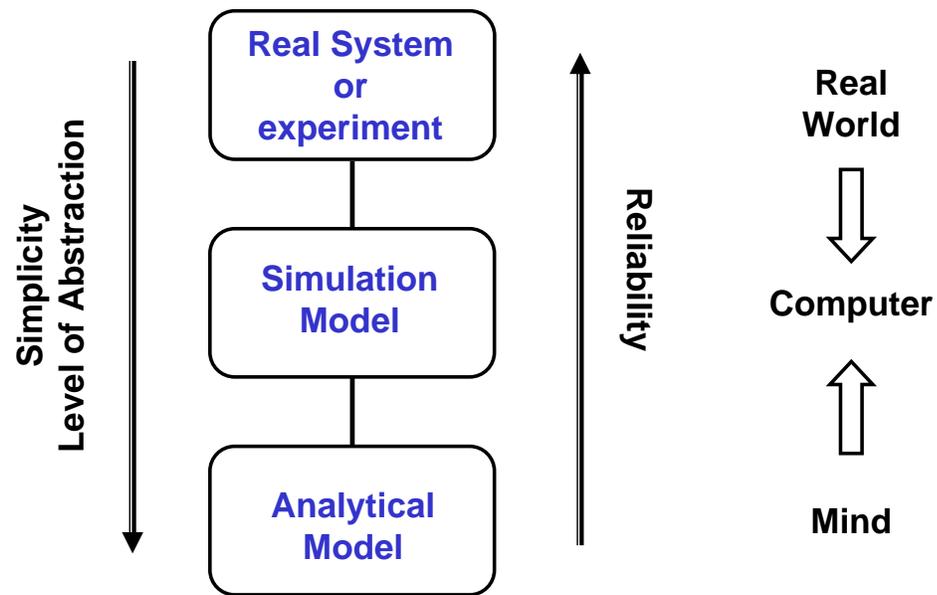
- Self-Fields linear only for uniform beam density
- Emittance conserved only for linear fields
- Envelope equation valid only if emittance conserved.

In general, evolution of beam size, emittance, and particle trajectories can be complicated and difficult to predict analytically.

Simulation: Applying physical laws (approximately) to model reality on a computer

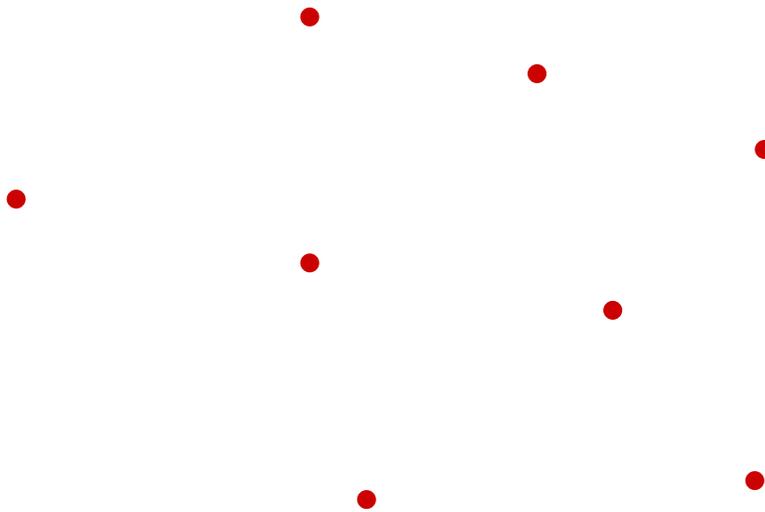
12

Where Does Simulation Fit in?



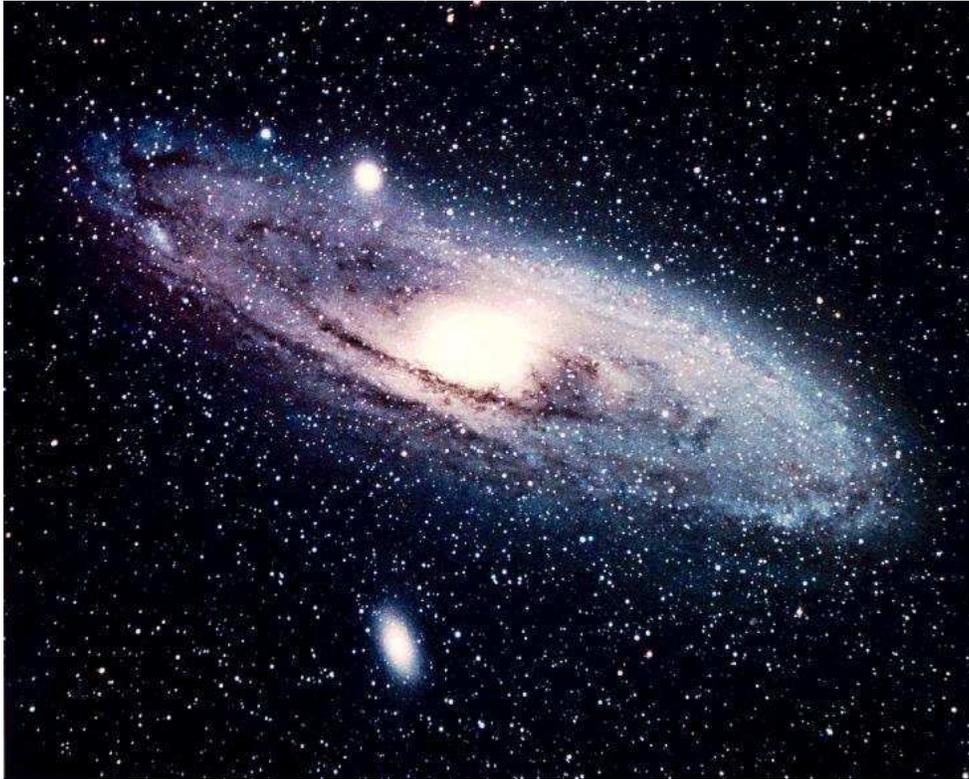
13

Particle Simulations



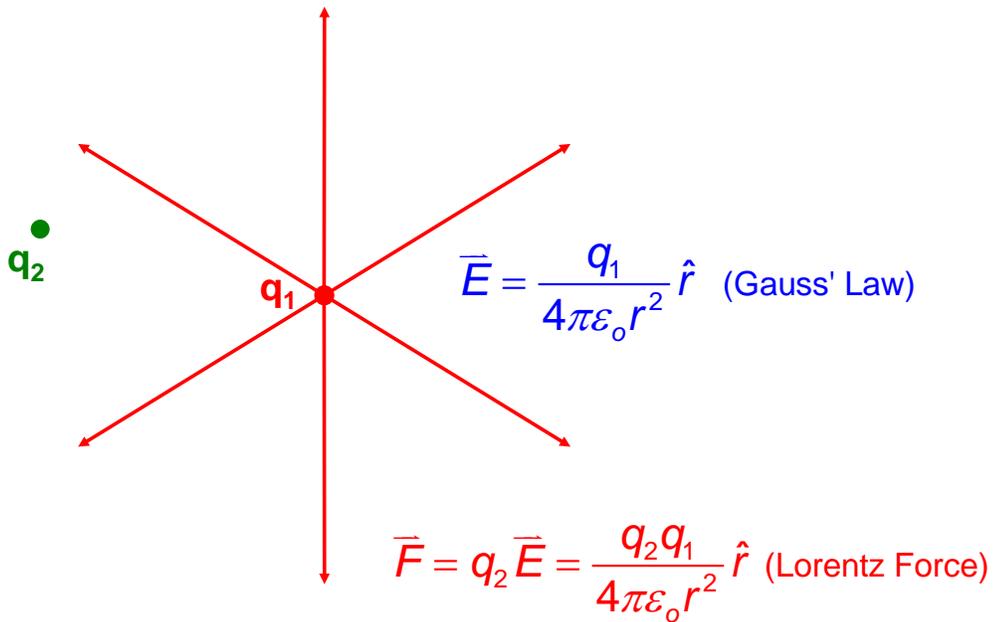
14

Galaxies face similar issue-how to model 10^{11} stars?



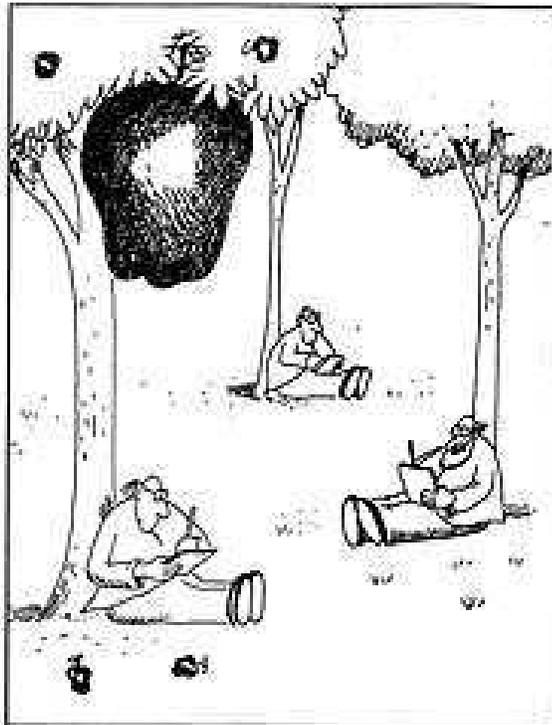
15

Coulomb's Law



16

Gravity - same form, different sign



"Nothing yet. ...How about you, Newton?"

$$\bar{g} = -\frac{GM}{r^2} \hat{r}$$

$$\bar{F} = m\bar{g} = -\frac{GmM}{r^2} \hat{r}$$

17

Classification of Computing Methods

Lund Notes - classify by the way the beam is represented:

- Particle Methods
- Distribution Methods
- Moment Methods

18

Approach

- Start with simplest brute force approach
- Find out problems with that approach
- Add some sophistication
- Find out problems

Each technique has its own strengths and weaknesses

Need to be aware of the boutique of available computational methods

19

Rami: Classification by Physical Laws (Loose)

Maxwell's Equations:

$$\bar{\nabla} \times \bar{E} = -\frac{\partial \bar{B}}{\partial t} \qquad \bar{\nabla} \cdot \bar{E} = \frac{\rho}{\epsilon_0}$$

$$\bar{\nabla} \times \bar{B} = \frac{1}{c^2} \frac{\partial \bar{E}}{\partial t} + \mu_0 \bar{J} \qquad \bar{\nabla} \cdot \bar{B} = 0$$

Lorentz Force Law:

$$\bar{F} = \frac{d}{dt}(\gamma m \bar{v}) = q(\bar{E} + \bar{v} \times \bar{B})$$

20

Specific Needs Behind Computing

- **Field Solvers:** Calculating the electromagnetic fields affecting the particles.
- **Particle Trackers:** Calculating particle trajectories in those fields.
- Calculating electric and magnetic fields generated by those particles. Self-Consistency → **Particle-in-cell (PIC) Codes.**

In most realistic situations, these factors are difficult to derive analytically.

Analytical models typically employ drastic simplification, which may or may not result in correct predictions.

21

Types of Accelerator Codes

Field Solvers

Static
Poisson
COULOMB

Eigenmode
YAP

Time Domain
MAFIA
AMOS
GDFIDL
Maxwell

Other
SuperFISH

Magnet Design
MAGPC
Maxwell 3D

Particle-in-Cell

Electrostatic
pdp1 / pdc2 / etc.
WARP

Electromagnetic
MAGIC / SOS
OOPIC
ARGUS
VORPAL

Gun Codes

E-Gun

Particle Trackers

Transfer Maps
MaryLie

Matrices
MAD / DIMAD
TRANSPORT

Multi-Particle
PARMELLA/ILLA
COZY

Other
SIMION

Envelope

TRACE
PBOLab
EMATCH

22

Other Approaches

Plasma Codes

Fluid model
Vlasov-Poisson model -
 δf Methods
Green's Function model

Other Accelerator Codes

Mechanical: structural / thermal
/ vacuum (*ProE*)
Radiation
Optics; FELs
Controls (*LabView*; *EPICS*)
Data Processing
Systems

Hybrid Codes

Particle-core model:
Envelope or PIC model for
beam bulk (core);
Tracking of halo particles.

PIC + map codes:
use PIC for self-fields;
transfer maps for external

Fluid + Envelope: (*CIRCE*)
Fluid for longitudinal;
Envelope for transverse

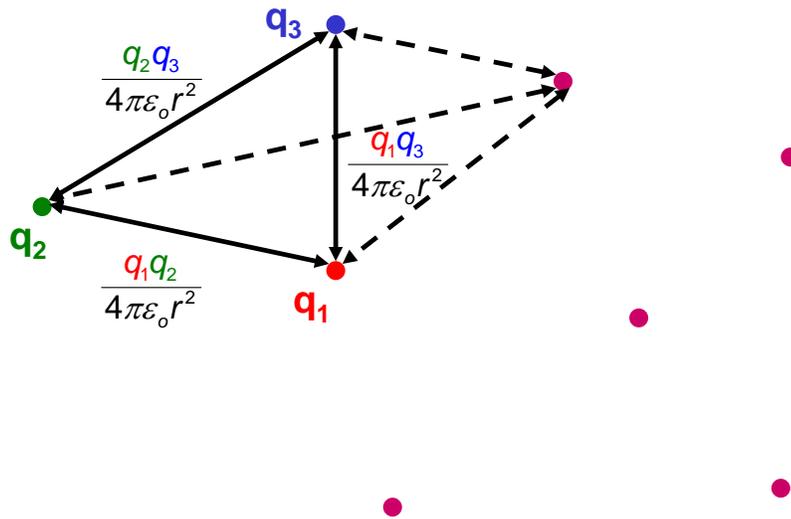
23

A Few Notes of Caution

- Successful codes are usually tremendous projects
- You really don't want to develop a new code if an existing one contains suitable approximations
- Sometimes a "less accurate" approximation can work better than more sophisticated models
- GIGO: The results you get are (at best) as good as the input you put in

24

Multiparticle Model



25

N-Body Problem

- Typical beam contains $\sim 10^9$ particles.
- Typical galaxy contains $\sim 10^{11}$ stars.
- Laboratory plasmas may contain $10^{12} - 10^{15} \text{ cm}^{-3}$ and inertial fusion plasmas up to $\sim 10^{22} \text{ cm}^{-3}$.
- Even with 10^9 particles, calculation of particle-particle interaction for every particle implies 10^{18} calculations to determine the forces.
- ***Once particles are advanced, forces change, and all 10^{18} interactions have to be recalculated.***

26

How Do We Simplify?

27

Approximation #1: Macroparticles

- Instead of accounting for every particle, take a **statistical sample** of particles → **Macroparticles**.
- A macroparticle is a special particle which carries the weight of a large number of particles when used to calculate the fields exerted on other particles.
- In response to a given field, the macroparticle moves as if it were a regular test particle.
- E.g, 10^9 particles → 10^5 particles.
Still have to deal with 10^{10} calculations.

Number of calculations $\sim n_p^2$.

28

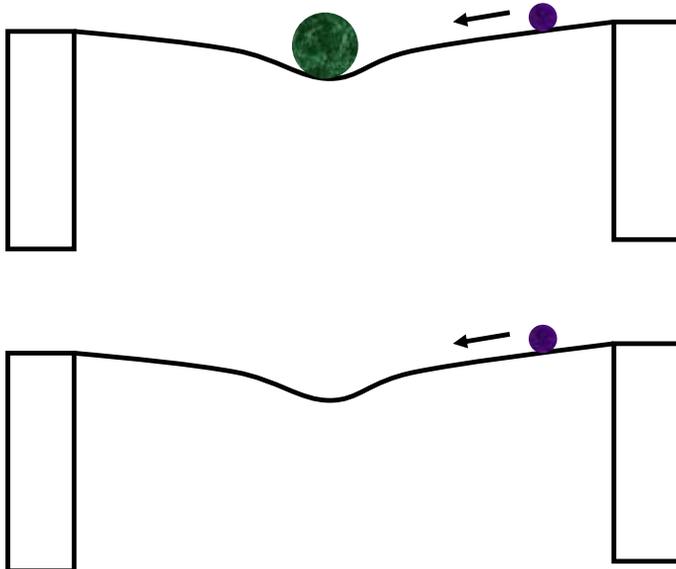
Notes on Sampling

For Approximation to be valid, need to ensure:

- Number of macroparticles should be sufficiently large to give good statistics.
- Average behavior and distribution of macroparticles should resemble that of real beam.

29

Observation



30

Approximation #2: Fields not Forces

- Do not deal with every possible interaction.
- Instead, use the concept of fields or potentials:
 - i. Interpolate particle positions onto grid to determine charge density.
 - ii. Calculate Fields as in previous lecture.
 - iii. Interpolate fields to position of each particle to determine force.

Total number of calculations $\sim n_p$

- In addition, have to do field calculation which for a 2D FFT $\sim n_x^2 \log n_x$

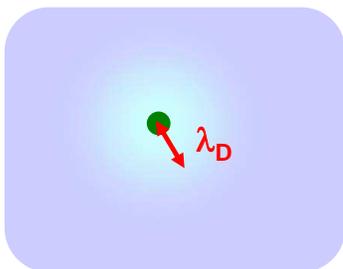
31

How (When) is This Possible?

Reiser Sec. 4.1
Birdsall Chap 1

Grid method ignores close-collisions (interactions between particles within same cell).

This is justified in reality when large number of particles



**Uniform distribution
of particles confined
by external field**
 $\phi(r) = \phi_e(r) + \phi_s(r)$

Boltzmann distribution:

$$\rho(r) = en(r) \exp\left(\frac{e\phi(r)}{k_B T}\right) \approx \frac{e^2 n(r)}{k_B T} \phi_s(r)$$

Solve for Potential from Poisson Equation:

$$\nabla^2 \phi = \frac{1}{r^2} \frac{d}{dr} \left(r^2 \frac{d\phi_s}{dr} \right) = -\frac{\rho(r)}{\epsilon_0} = -\frac{e^2 n(r)}{\epsilon_0 k_B T} \phi_s(r)$$

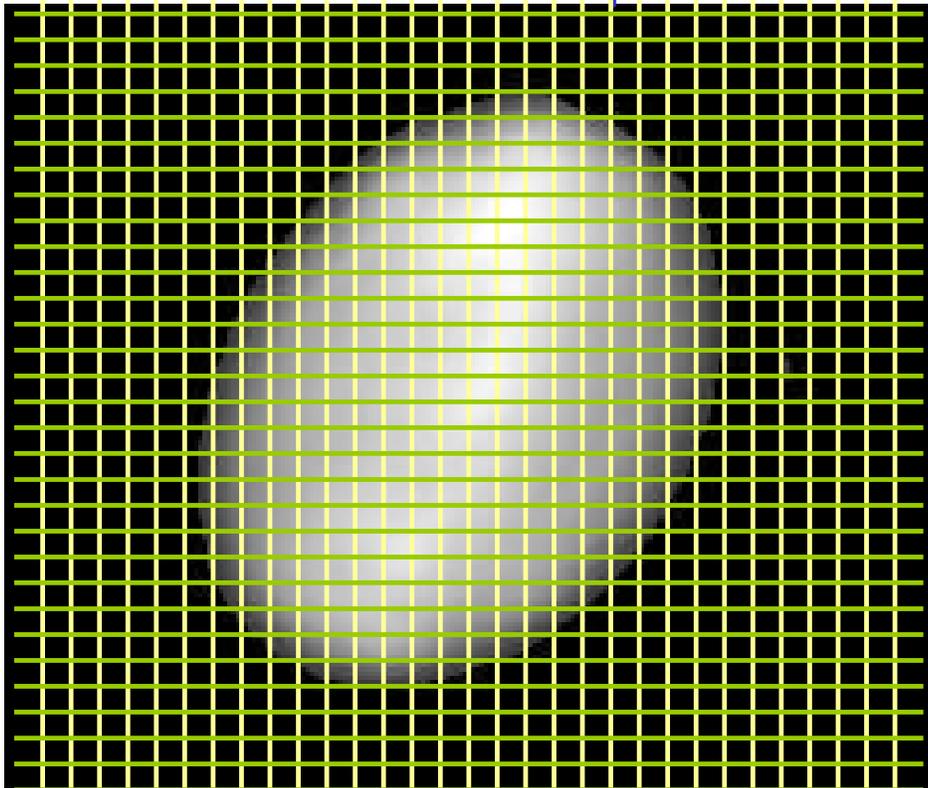
$$\phi(r) = \frac{q}{4\pi\epsilon_0 r} \exp\left(-\frac{r}{\lambda_D}\right)$$

Debye Length

$$\lambda_D = \sqrt{\frac{\epsilon_0 k_B T}{e^2 n}}$$

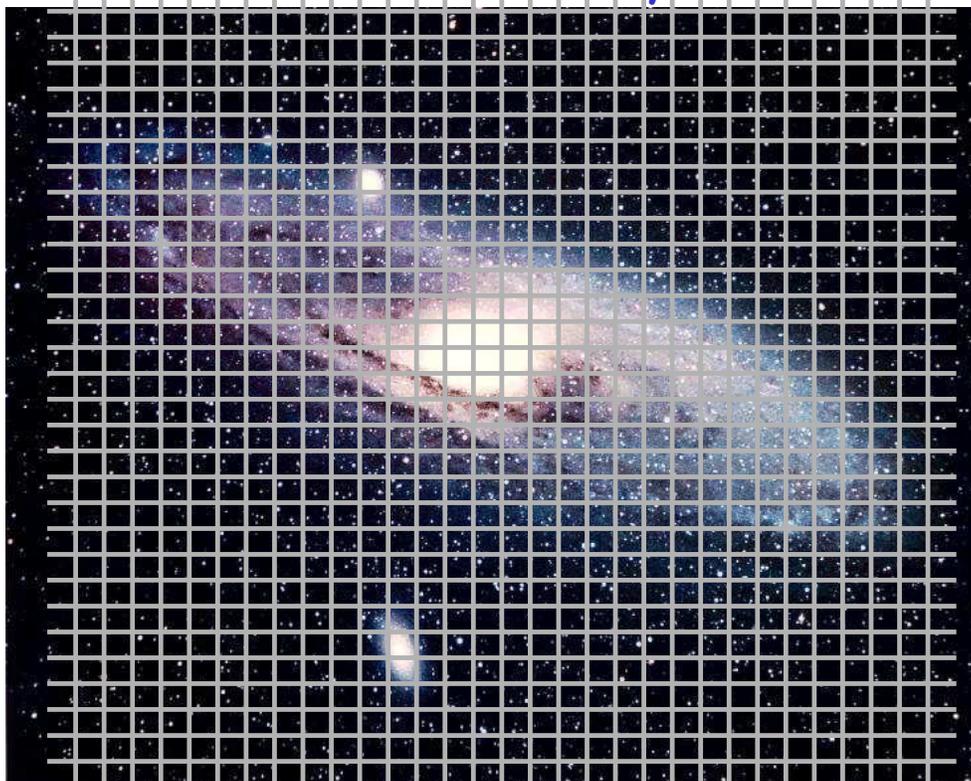
32

How Do We Model 10^9 particles?



33

Galaxies Can Be Similarly Treated



34

Debye Shielding

If:

Large number of particles (# particles in Debye sphere = $n\lambda_D^3 \gg 1$)

$\lambda_D < R$ (Space-Charge-Dominated)

Force from “collective” potential \gg force from nearest neighbors

Probability of individual particle-particle interaction

(Coulomb collisions) \ll collective interaction

Can use grid to model space-charge-force

Grid needs to be small enough to resolve potential variations (**Rule of thumb** $\Delta x \sim \lambda_D/3$)

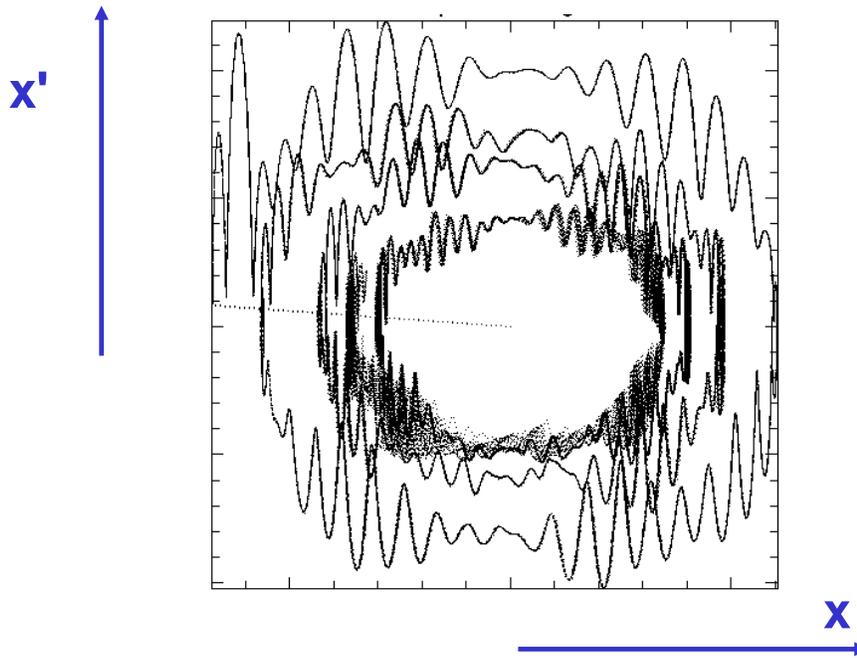
Otherwise, individual “collisions” important and need be modeled

35

Particle Tracking and Numerical Methods

Rami Kishek

Particle Trajectories: x - x' Phase Space



2

Pushing Particles (Tracking)

Maxwell's Equations:

$$\bar{\nabla} \times \bar{E} = -\frac{\partial \bar{B}}{\partial t} \quad \bar{\nabla} \cdot \bar{E} = \frac{\rho}{\epsilon_0}$$

$$\bar{\nabla} \times \bar{B} = \frac{1}{c^2} \frac{\partial \bar{E}}{\partial t} + \mu_0 \bar{J} \quad \bar{\nabla} \cdot \bar{B} = 0$$

Lorentz Force Law:

$$\bar{F} = \frac{d}{dt}(\gamma m \bar{v}) = q(\bar{E} + \bar{v} \times \bar{B})$$

3

Computer Limitations

- Computers can only handle **numbers** (binary, no less).
- Computers can only perform **simple arithmetic** (no differentiation).
- Furthermore, computers can only handle a **finite number of data**. Continuous functions (made of infinite number of “points”) must be discretized.
- Can only deal with **bounded problems**.

4

Particle Pushing

$$\bar{v} = \frac{d\bar{x}}{dt}$$

$$\bar{F} = \frac{d}{dt}(\gamma m \bar{v})$$

**Equations of Motion
(Newton)**

Know derivatives of function – want to predict function’s future values

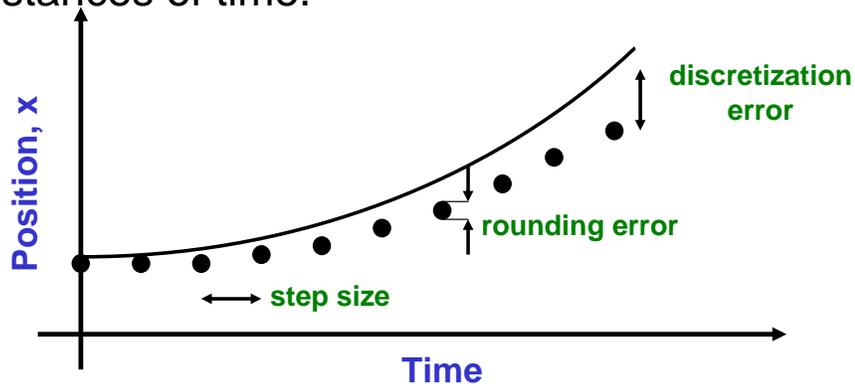
To determine position of a particle as a function of time in response to a given force, need to integrate equations of motion.

Note: Forces can be time-dependent.

5

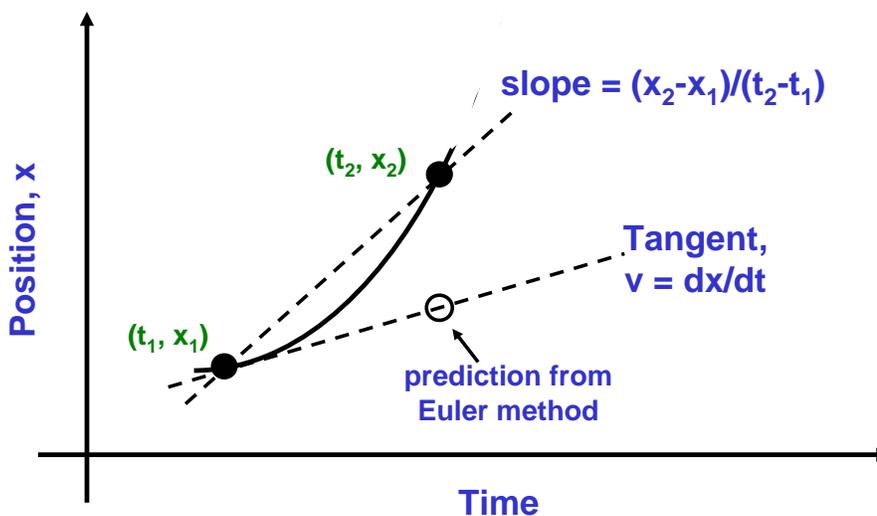
Time Discretization

- Particle positions and velocities change continuously with respect to time, but computer can hold only finite number of quantities.
- Hence treat time (or s) as a discrete quantity: care about particle positions and velocities only at discrete instances of time.



6

Euler Method



As $\Delta t = (t_2 - t_1) \rightarrow 0$, then slope $\rightarrow dx/dt$

7

Forward-, Backward-, & Central-Differences

$$f'(s) \approx \frac{f(s + \Delta s) - f(s)}{\Delta s} \quad \text{Forward-Difference}$$

$$f'(s) \approx \frac{f(s) - f(s - \Delta s)}{\Delta s} \quad \text{Backward-Difference}$$

$$f'(s) \approx \frac{f(s + \frac{\Delta s}{2}) - f(s - \frac{\Delta s}{2})}{\Delta s} \quad \text{Central-Difference}$$

Forward- and Backward- Differences Converge as Δs
Central-Difference Converges as $(\Delta s)^2$

8

Accuracy of Difference Method

Taylor Expand arbitrary function:

$$f(s + \Delta s) = f(s) + \Delta s f'(s) + \frac{\Delta s^2}{2!} f''(s) + \frac{\Delta s^3}{3!} f'''(s) + \frac{\Delta s^4}{4!} f^{iv}(s) + O(\Delta s^5)$$

$$f(s - \Delta s) = f(s) - \Delta s f'(s) + \frac{\Delta s^2}{2!} f''(s) - \frac{\Delta s^3}{3!} f'''(s) + \frac{\Delta s^4}{4!} f^{iv}(s) + O(\Delta s^5)$$

Subtract 2 Equations:

$$f'(s) = \frac{f(s + \Delta s) - f(s - \Delta s)}{2\Delta s} - \frac{\Delta s^2}{6} f'''(s) + \dots$$

Add 2 Equations:

$$f''(s) = \frac{f(s + \Delta s) + f(s - \Delta s) - 2f(s)}{\Delta s^2} - \frac{\Delta s^2}{12} f^{iv}(s) + \dots$$

9

Discretization Errors

Can reduce errors in 2 ways:

- Use smaller time steps
- Use higher-order integration technique which converges faster as a function of time step.

Examples,

Euler method converges as Δt or Δs

4th-order Runge-Kutta converges as $(\Delta t)^4$.

Center-Difference Methods converge as $(\Delta t)^2$.

10

LeapFrog Methods

Eqns of Motion: 2 first-order differential equation
center-difference expansion of each

$$x(t + \Delta t) \cong x(t) + (\Delta t)v(t + \Delta t/2)$$

$$\gamma mv(t + \Delta t/2) \cong \gamma mv(t - \Delta t/2) + (\Delta t)F(t)$$

Notes:

Positions and Velocities not known at same points.

Electric forces depend only on particle positions.

Get $(\Delta t)^2$ convergence for only one calculation per step per equation.

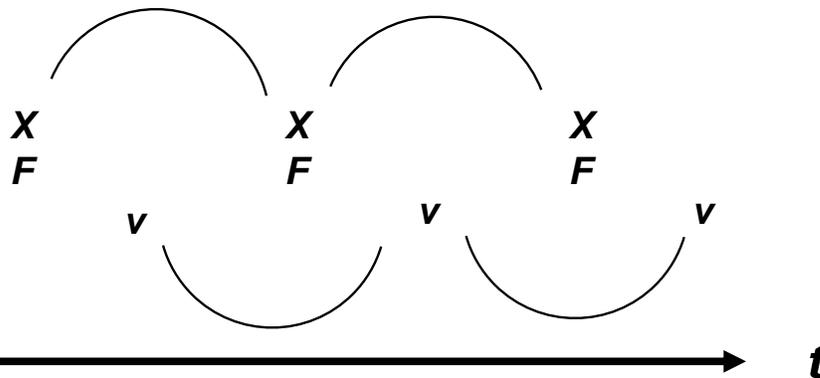
Pair **not** self-starting. Need to integrate backwards 1/2 step to start.

11

The Leap Frog

$$x(t + \Delta t) = x(t) + \Delta t \times v(t + \Delta t/2)$$

$$v(t + \Delta t/2) = v(t - \Delta t/2) + \Delta t \times F(t) / m$$



Note that the forces depend only on the particle positions.

12

Why use leapfrog ?

-
- Only second order but simple (and appropriate because of truncation errors in field solution).
 - Solves an easy to visualize physical problem, i.e. gives the correct answer for a force which is constant over the integration interval
 - Symplectic
 - time-centered and reversible, i.e. simulation can be run backward
 - Langdon called these characteristics “virtuous”.

13

Implications of Leapfrog

- To obtain useable output, velocity and position of particles must be measured at the same time.
- Therefore, at all instances of time in which diagnostic output is desired, the calculation must be advanced 1/2 step to synchronize position and velocity → **semi-Leapfrog**.
- This is expensive.

14

Integrating the envelope equation

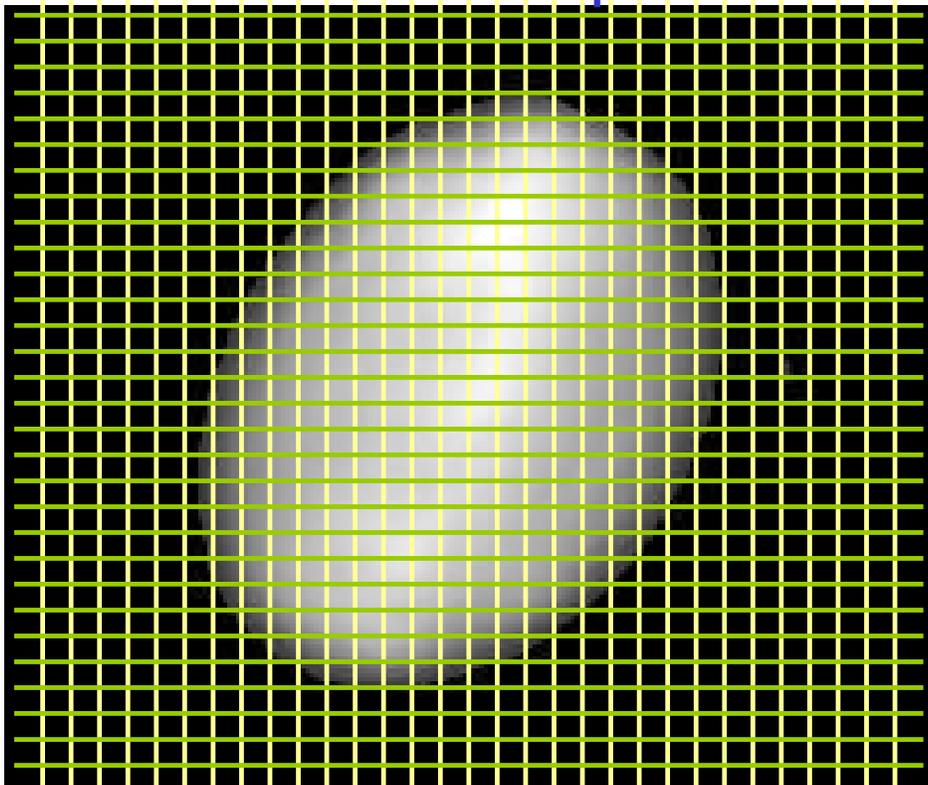
- Can use same methods for general purpose integration

15

Computation in Beams: Field Solvers

Rami Kishek

Review: **How Do We Model 10^9 particles?**



Solving for Fields

Humphries CPA 4.2, 4.5
Birdsall Chap 2

Maxwell's Equations:

$$\vec{\nabla} \times \vec{E} = -\frac{\partial \vec{B}}{\partial t} \qquad \vec{\nabla} \cdot \vec{E} = \frac{\rho}{\epsilon_0}$$

$$\vec{\nabla} \times \vec{B} = \frac{1}{c^2} \frac{\partial \vec{E}}{\partial t} + \mu_0 \vec{J} \qquad \vec{\nabla} \cdot \vec{B} = 0$$

Lorentz Force Law:

$$\vec{F} = \frac{d}{dt}(\gamma m \vec{v}) = q(\vec{E} + \vec{v} \times \vec{B})$$

3

Static Approximation (steady state) $\frac{\partial}{\partial t} \rightarrow 0$:

$\vec{\nabla} \times \vec{E} = 0 \frac{\partial \vec{B}}{\partial t}$	$\vec{\nabla} \cdot \vec{E} = \frac{\rho}{\epsilon_0}$	<p>Electrostatic</p>
<p style="color: red; font-weight: bold;">Source Terms</p>		
$\vec{\nabla} \times \vec{B} = \frac{1}{c^2} \frac{\partial \vec{E}}{\partial t} + \mu_0 \vec{J}$	$\vec{\nabla} \cdot \vec{B} = 0$	<p>Magnetostatic</p>

4

Electrostatic Case

First Equation, $\bar{\nabla} \times \bar{E} = 0$ allows us to define electrostatic potential, ϕ :

$$\bar{E} = -\bar{\nabla}\phi$$

$$\bar{\nabla} \cdot \bar{E} = -\bar{\nabla} \cdot \bar{\nabla}\phi$$

Poisson Eqn

$$\nabla^2 \phi = \frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} + \frac{\partial^2 \phi}{\partial z^2} = -\frac{\rho}{\epsilon_0}$$

If no space charge, $\rho = 0$,

$$\nabla^2 \phi = 0$$

Laplace's Eqn

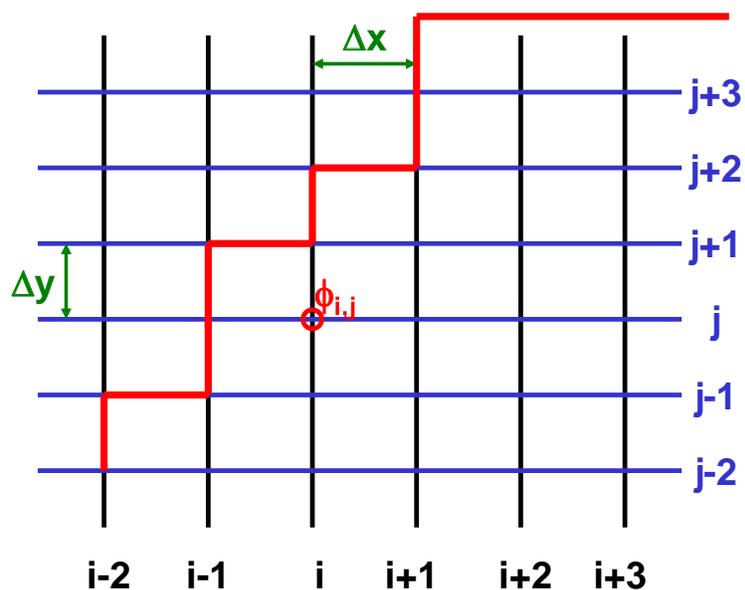
5

Solving for E-S Potential numerically

Divide space into grid.

Quantities (e.g., charge density, potential) defined on the grid.

Boundary



6

Boundary Conditions

Dirichlet B.C.: $\phi(\text{border point}) = \text{constant}$

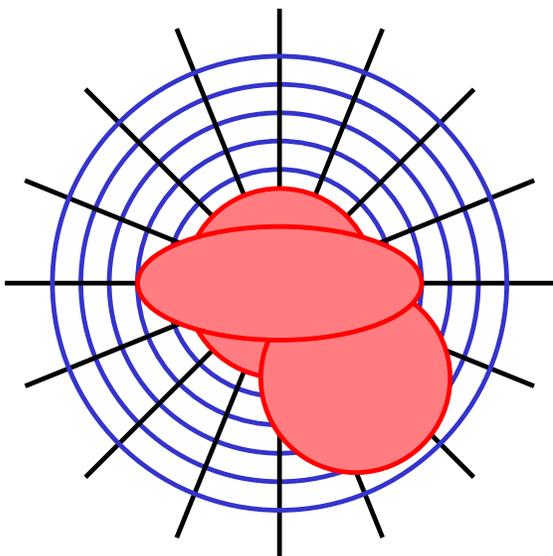
0 for conducting
boundary

Neumann B.C.: $\frac{\partial \phi}{\partial x} = \text{const}$

7

Other Gridding Geometries Possible

HW Problem



Alternative gridding
may be more suitable

However, Pay a Price:

- Difference equations more complex
- Resolution varies in different parts of beam
- Gridding may introduce unphysical behavior

Adaptive Mesh (grid size changes with location to model details at higher resolution). Again more complex.

8

Forward-, Backward-, & Central-Differences

$$f'(x) \approx \frac{f(x + \Delta x) - f(x)}{\Delta x} \quad \text{Forward-Difference}$$

$$f'(x) \approx \frac{f(x) - f(x - \Delta x)}{\Delta x} \quad \text{Backward-Difference}$$

$$f'(x) \approx \frac{f(x + \frac{\Delta x}{2}) - f(x - \frac{\Delta x}{2})}{\Delta x} \quad \text{Central-Difference}$$

Forward- and Backward- Differences Converge as Δx

Central-Difference Converges as $(\Delta x)^2$

$$\text{Here } ' \equiv \frac{\partial}{\partial x}$$

9

Accuracy of Difference Method

Taylor Expand arbitrary function:

$$f(x + \Delta x) = f(x) + \Delta x f'(x) + \frac{\Delta x^2}{2!} f''(x) + \frac{\Delta x^3}{3!} f'''(x) + \frac{\Delta x^4}{4!} f^{iv}(x) + o(\Delta x^5)$$

$$f(x - \Delta x) = f(x) - \Delta x f'(x) + \frac{\Delta x^2}{2!} f''(x) - \frac{\Delta x^3}{3!} f'''(x) + \frac{\Delta x^4}{4!} f^{iv}(x) + o(\Delta x^5)$$

Subtract 2 Equations:

$$f'(x) = \frac{f(x + \Delta x) - f(x - \Delta x)}{2\Delta x} - \frac{\Delta x^2}{6} f'''(x) + \dots$$

Add 2 Equations:

$$f''(x) = \frac{f(x + \Delta x) + f(x - \Delta x) - 2f(x)}{\Delta x^2} - \frac{\Delta x^2}{12} f^{iv}(x) + \dots$$

10

Transform Poisson's Equation

Derivative \Rightarrow Finite Difference

$$\phi(x, y) = \phi(i\Delta x, j\Delta y) = \phi(i, j) = \phi_{i,j}$$

$$\phi'_{i+\frac{1}{2},j} = \frac{\partial}{\partial x} \phi_{i+\frac{1}{2},j} \approx \frac{\phi_{i+1,j} - \phi_{i,j}}{\Delta x} \quad \text{Central-Difference Estimate Error} \sim O(\Delta x^2)$$

$$\frac{\partial^2 \phi_{i,j}}{\partial x^2} \approx \frac{\phi'_{i+\frac{1}{2},j} - \phi'_{i-\frac{1}{2},j}}{\Delta x} \quad \text{Error in 2nd derivative} \sim O(\Delta x^2)$$

11

Finite-Difference Form of Laplacian Operator

$$\begin{aligned} \frac{\partial^2 \phi_{i,j}}{\partial x^2} &\approx \frac{(\phi_{i+1,j} - \phi_{i,j})/\Delta x - (\phi_{i,j} - \phi_{i-1,j})/\Delta x}{\Delta x} \\ &= \frac{\phi_{i+1,j} + \phi_{i-1,j} - 2\phi_{i,j}}{(\Delta x)^2} \end{aligned}$$

$$\mathbf{2-D} \quad \nabla^2 \phi_{i,j} \approx \frac{\phi_{i+1,j} + \phi_{i-1,j} - 2\phi_{i,j}}{(\Delta x)^2} + \frac{\phi_{i,j+1} + \phi_{i,j-1} - 2\phi_{i,j}}{(\Delta y)^2}$$

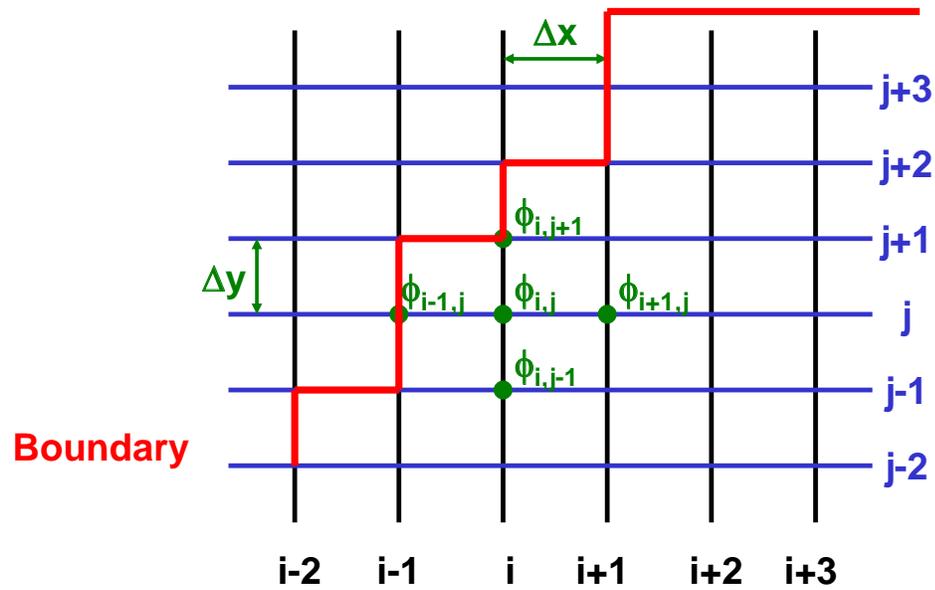
Typically, $\Delta x = \Delta y = \Delta$

$$\phi_{i,j} \approx \frac{1}{4} (\phi_{i+1,j} + \phi_{i-1,j} + \phi_{i,j+1} + \phi_{i,j-1})$$

Potential at each point is average of that at neighboring points

12

$$\phi_{i,j} \approx \frac{1}{4} (\phi_{i+1,j} + \phi_{i-1,j} + \phi_{i,j+1} + \phi_{i,j-1})$$



13

Poisson's Equation

$$\left(\phi_{i+1,j} + \phi_{i-1,j} + \phi_{i,j+1} + \phi_{i,j-1} \right) - 4\phi_{i,j} \approx -\frac{\rho_{i,j} \Delta^2}{\epsilon_0}$$

Q_{ij} - charge in cell

$$\phi_{i,j} \approx \frac{1}{4} \left(\phi_{i+1,j} + \phi_{i-1,j} + \phi_{i,j+1} + \phi_{i,j-1} \right) + \frac{1}{4} \frac{Q_{i,j}}{\epsilon_0}$$

Can generalize to 3-D ...

14

Algorithms for Solving Coupled Difference Equations

A. Tridiagonal matrices: 1-D Problems

B. Finite-Difference Methods: (for higher-D)

- **Gauss-Seidel:** Iteration
- **Successive Over-Relaxation (SOR):** Enhanced Iteration
Most general method.
- **Multi-grid** Methods: SOR with changing grid size each iteration.
- **Conjugate Gradient:** Sparse Matrices

C. Fourier Transforms.

15

A. Tridiagonal Matrices

$$\phi_{i-2} - 2\phi_{i-1} + \phi_i = -\frac{\Delta^2 \rho_{i-1}}{\epsilon_0} \quad \mathbf{1-D}$$

$$\phi_{i-1} - 2\phi_i + \phi_{i+1} = -\frac{\Delta^2 \rho_i}{\epsilon_0}$$

$$\phi_i - 2\phi_{i+1} + \phi_{i+2} = -\frac{\Delta^2 \rho_{i+1}}{\epsilon_0}$$

$$\begin{pmatrix} -2 & 1 & 0 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 & 0 \\ 0 & 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & 0 & 1 & -2 \end{pmatrix} \begin{pmatrix} \phi_1 \\ \vdots \\ \phi_6 \end{pmatrix} = \begin{pmatrix} -\phi_0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -\phi_7 \end{pmatrix} + \frac{\Delta^2}{\epsilon_0} \begin{pmatrix} \rho_1 \\ \vdots \\ \rho_6 \end{pmatrix}$$

Source
Boundaries

16

B. Finite Difference Methods

1. Write Difference equations for each point on the grid.
2. Results in system of N coupled *algebraic* equations with N unknowns where N is the number of gridpoints.
3. Can solve, in principle at least.

In practice, for large N , or for complicated boundaries, system can be difficult to solve. For example, inversion of large matrix not computationally efficient.

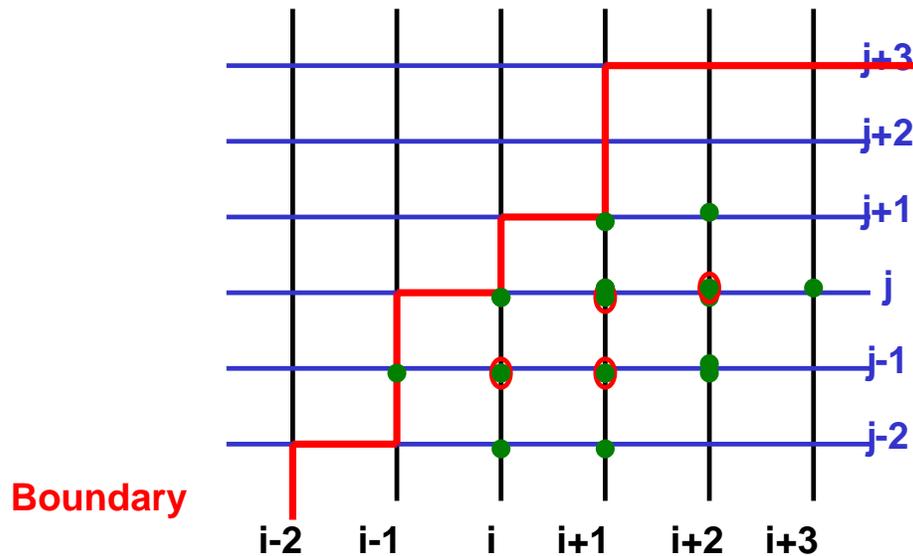
17

Iterative Solution Procedure (Gauss-Seidel)

1. Initial guess of ϕ at all points.
 2. Calculate charge density and load on grid.
 3. Impose Boundary Conditions
 4. Generate Next Guess by calculating potential at each point from existing data.
 5. Iterate until solution converges (typically some 200 iterations).
- 

18

Demonstration



19

Successive Over-Relaxation (SOR) Method

Define **Residual** for each iteration, n :

$$\begin{aligned} {}^n R_{i,j} &= {}^{calc} \phi_{i,j} - {}^{n-1} \phi_{i,j} \\ &= \frac{1}{4} \left(\phi_{i+1,j} + \phi_{i-1,j} + \phi_{i,j+1} + \phi_{i,j-1} \right) - \phi_{i,j} \end{aligned}$$

$${}^n \phi = {}^{n-1} \phi + \omega {}^n R$$

**Over-relaxation
coefficient, $1 \leq \omega \leq 2$**

Obviously, solution converges when $R = 0$

20

Notes on SOR method

Check for yourselves:

$\omega = 1$ corresponds to simple averaging and iteration
(Gauss-Seidl)

$\omega = 2$ corresponds to overshooting so as to achieve faster convergence. Gives more weight to new value of ϕ .

Convergence still SLOW!

E.g., for an $n \times n$ 2-D grid, calculation time $\sim n^3$, since need order n iterations to propagate errors out of mesh.

21

So Why use SOR?

Finite-Difference gives closed set of equations, so can solve in principle.

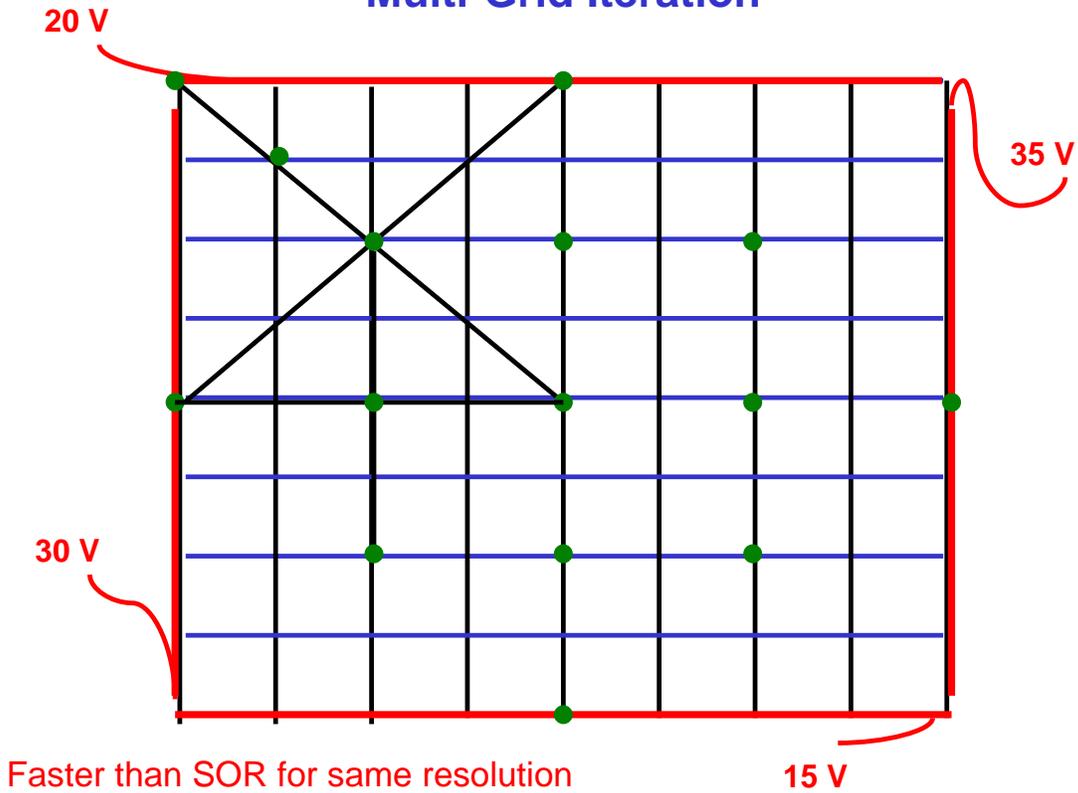
Solution can get messy for complicated geometries.

SOR does not care about complexity of boundaries!

Takes about as much time to solve a complicated problem as a simple one, for a given number of grid cells.

22

Multi-Grid Iteration



23

C. FFT Methods

See Lund Notes

Much faster than SOR - no iterations are needed.

Basic idea:

Transform to Fourier Domain $\rho_{i,j} \rightarrow \rho(k_x, k_y)$

$\phi_{i,j} \rightarrow \phi(k_x, k_y)$

Differentiation becomes multiplication

$$\nabla^2 \phi_{i,j} \rightarrow -(k_x^2 + k_y^2) \phi(k_x, k_y)$$

$$\phi(k_x, k_y) = \frac{\rho(k_x, k_y)}{\epsilon_0 (k_x^2 + k_y^2)}$$

Transform Back to spatial domain

24

Why FFT?

Takes advantage of Fast Fourier Transform routines,
hence converges much faster:

FFT of N points takes $\sim N \log N$ calculations

For $n \times n$ 2-D grid, total number of points $N = n^2$, so FFT
converges as $n^2 \log n^2 = 2 n^2 \log n$

Better than n^3

25

Limitation of FFT Method

Can only implement simple boundaries, e.g., round pipe.

To include conducting boundary, calculate equivalent
image charge needed to produce zero-potential on it.

1. Calculate a **capacitance matrix** relating potential on boundary points to image charges at the boundary.
2. Solve for the potential without any images.
3. Multiply the potential at the boundary points by the capacitance matrix to get the induced images.
4. Add the images to the total charge and recalculate the potential.

26

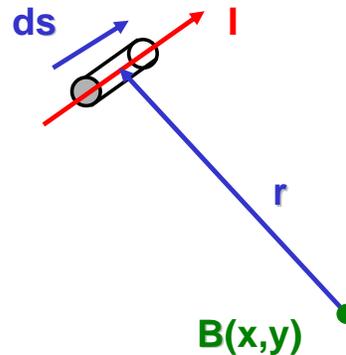
Magnetostatic Calculations

Frequently desire to know magnetic field of a DC electromagnet. Know current configuration, want B-field.

Possible methods:

- Vector Potential
- Biot-Savart Law

$$\vec{B} = \frac{\mu}{4\pi} \sum \left(\frac{I d\vec{s} \times \hat{r}}{r^2} \right)$$



27

Electromagnetic

Use full set of equations.

$$\vec{\nabla} \times \vec{E} = -\frac{\partial \vec{B}}{\partial t}$$

$$\vec{\nabla} \cdot \vec{E} = \frac{\rho}{\epsilon_0} \rightarrow 0$$

$$\vec{\nabla} \times \vec{B} = \frac{1}{c^2} \frac{\partial \vec{E}}{\partial t} + \mu_0 \vec{J} \rightarrow 0$$

$$\vec{\nabla} \cdot \vec{B} = 0$$

Absence of space charge leads to wave equation in free space

$$\frac{\partial^2 \vec{E}}{\partial t^2} = -c^2 \nabla \times \nabla \times \vec{E}$$

28

Wave Equation

$$\frac{\partial^2 \eta}{\partial t^2} = c^2 \frac{\partial^2 \eta}{\partial x^2}$$

29

Notes on Electromagnetic Calculations

- In general more difficult to solve.
- Time needs to be discretized as well.
- For leapfrog algorithms, x and v not known simultaneously. Hence, ρ , and \mathbf{J} , and consequently \mathbf{E} and \mathbf{B} not known simultaneously either.
- Leads to instability if $c \Delta t > \Delta x$

Courant Condition: $(c\Delta t)^2 \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right) < 1$

30

Particle-in-Cell Codes (PIC)

Rami Kishek

Particle-in-Cell (PIC)

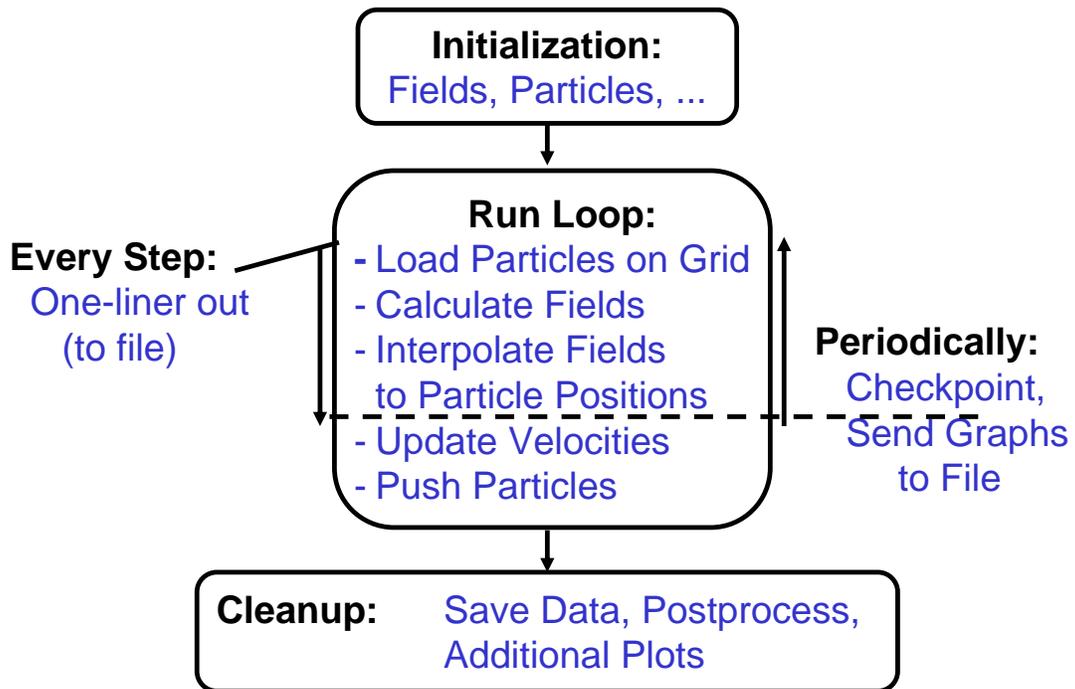
Self-Consistent method for solving

Maxwell's Equations and **Lorentz Force Law:**

Uses **macroparticles** to sample actual particles

Represents **fields on a grid.**

Structure of a PIC Code



3

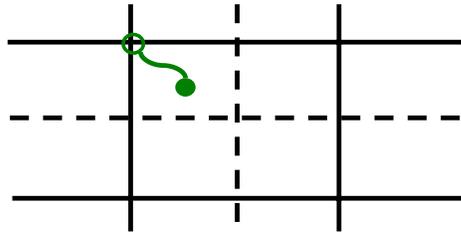
Components of a PIC code

- Field Calculation
- Particle Pushing
- Interpolation schemes
- Particle Loading
- Diagnostics and Output.

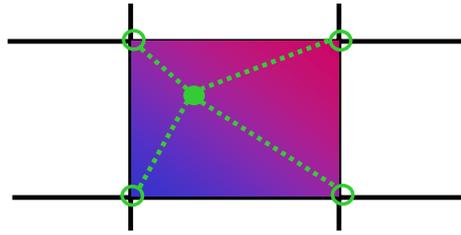
4

Interpolation Schemes

Nearest Gridpoint (NGP)



Cloud-in-cell



Higher-Order

5

Particle Loading

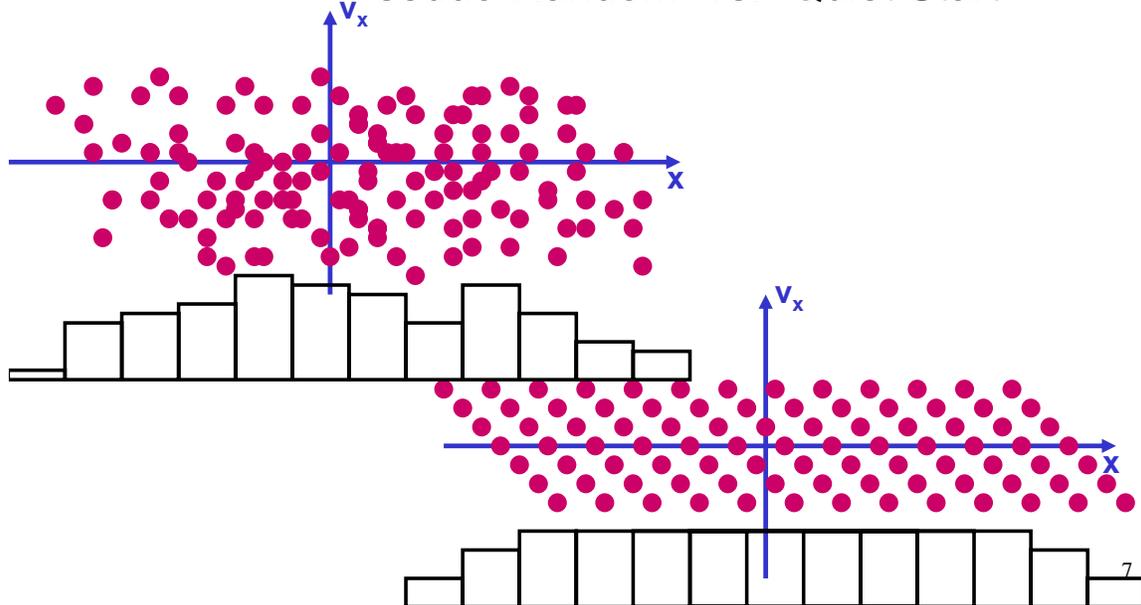
- Populating Phase Space
- Random number generation
- Initial distributions

6

Particle Simulation Caveat: Particle Loading

Populating Phase Space:

“Pseudo-Random” vs. “Quiet Start”

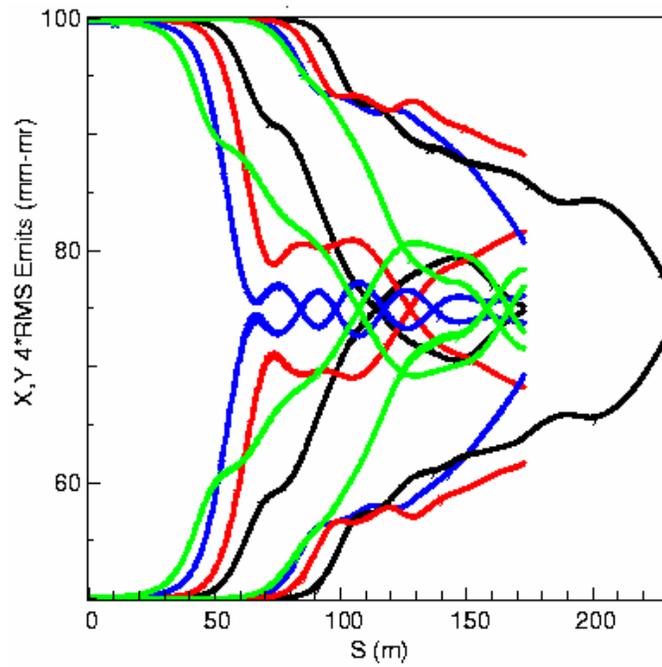


Particle Simulation Caveat: Particle Loading

Populating Phase Space:

“Pseudo-Random” vs. “Quiet Start”

The Dangers of "Quiet-Start"



9

Enhancing Speed

- Symmetry
- Filtering

10

Self-Consistency

- With sufficient space charge, forces generated between particles need to be included in model.
- Motion of the particles will change these self-forces, hence the calculation needs to be updated as the particles move, e.g., every time step.
- Space charge forces can be represented in several different ways, e.g., as an averaged linear force, etc.
- We are primarily interested in a representation of space charge forces that is self-consistent with the particle model.

11

Output

- 1/2 Time step to synchronize position and velocity before recording output. Visualization is an important aspect.
- Real-time interactive graphics vs.. No Output!

12

One-liners: a Suitable Compromise

One line of text for each step listing values of selected variables. Useful for keeping an eye in case anything goes wrong, e.g.,

```
it = 30 zbeam = 0.4500 2*xrms = 9.52 emity 58.55 nplive = 1000
it = 40 zbeam = 0.6000 2*xrms = 9.81 emity 58.56 nplive = 1000
it = 50 zbeam = 0.7500 2*xrms = 10.11 emity 58.56 nplive = 1000
```

13

Graphical Output

- Graphical output is for human consumption.
- Need graphics in order to condense the massive amounts of data generated by simulation.
- Need to display it in presentable form.
- Graphical output also useful for comparison of different simulations.

14

Sources of Error

- Macroparticle Statistics
- Discrete Time Step
- Discrete Potential (gridcells)
- Input and initialization errors; particle loading
- Modeling issues

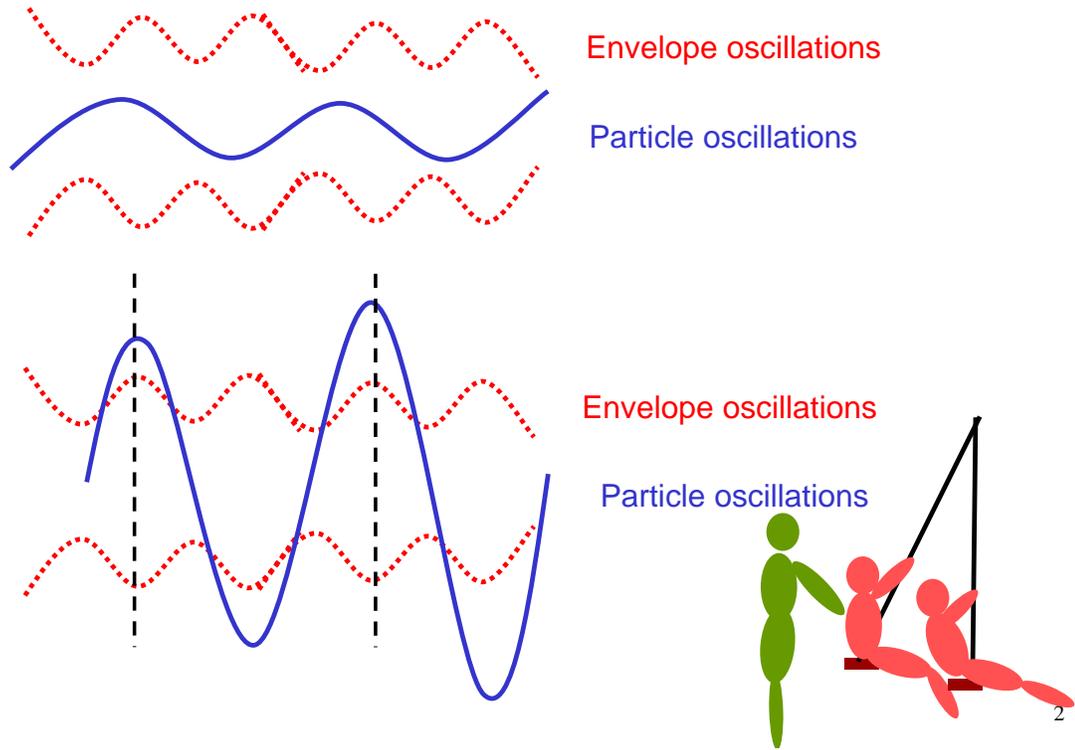
15

Advanced Computing Topics

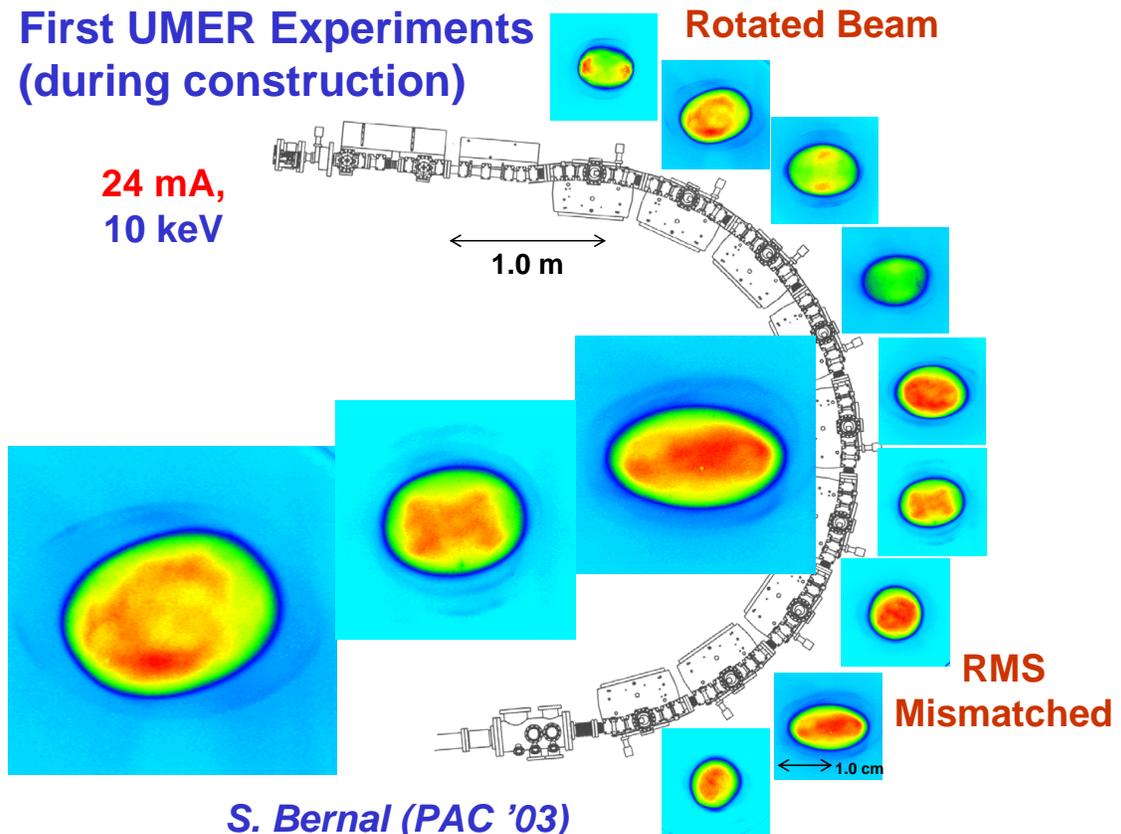
Rami Kishek

1

Parametric Resonance



First UMER Experiments (during construction)



Dynamic Range Problem

- Interested in halos.
- Halos can contain as little as 10^{-5} particle density as in the main beam.
- Resolving that accurately requires very good particle statistics. For a PIC code, this means 10^9 particles, or more.

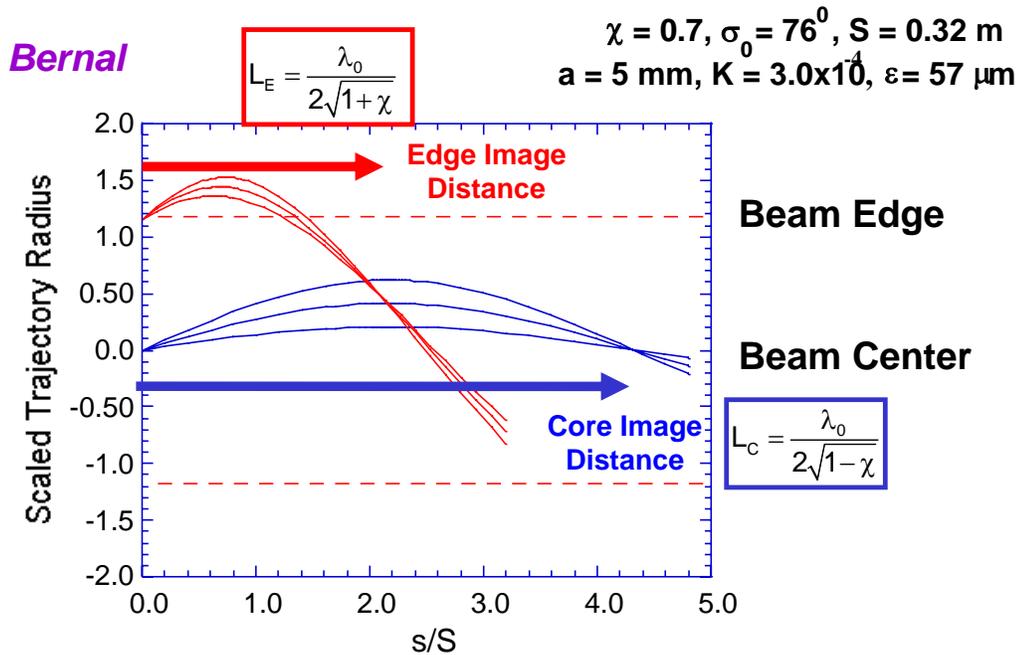
4

Possible Solutions

1. Enhance Speed of PIC codes:
 - Sub-Cycling: do not solve for fields at every step
 - Split-operator techniques (maps + PIC)
2. Particle-Core Models
3. Subtract out main beam particles, i.e., look only at halo or perturbation $\Downarrow \delta f$
4. Direct Solution in terms of particle “density” \Downarrow
“Vlasov Solvers”
5. Hybrid Codes

5

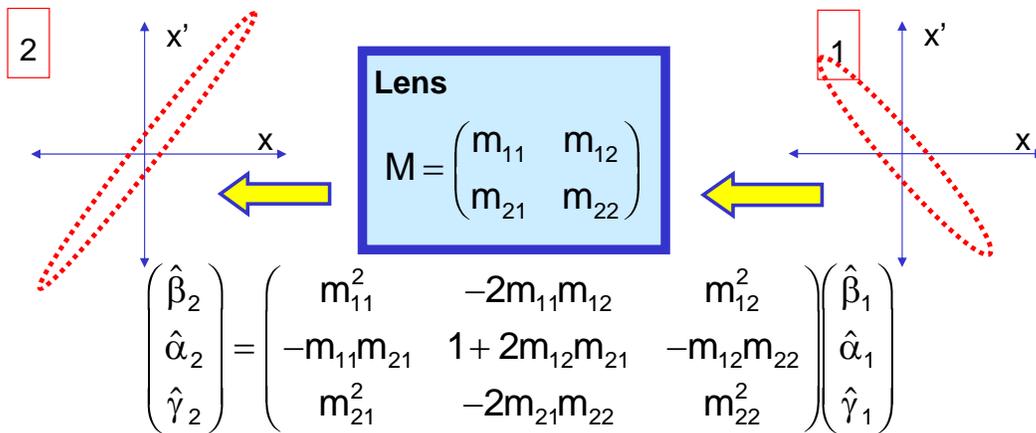
2. Particle Core Model: Edge Lensing



Theory / P-Core Model

6

Matrix Techniques and Maps



TRANSPORT: Library of lattice element matrices

TRACE: Advances beam ellipse using matrix models of lattice

MARYLIE: Lie mapping of lattice elements, can handle nonlinearity

7

Following slides courtesy of VNL ...

*Alex Friedman, Ron Davidson, Eric Sonnendrucker,
Wei W. Lee, Bill Sharp, & Dave Grote*

8

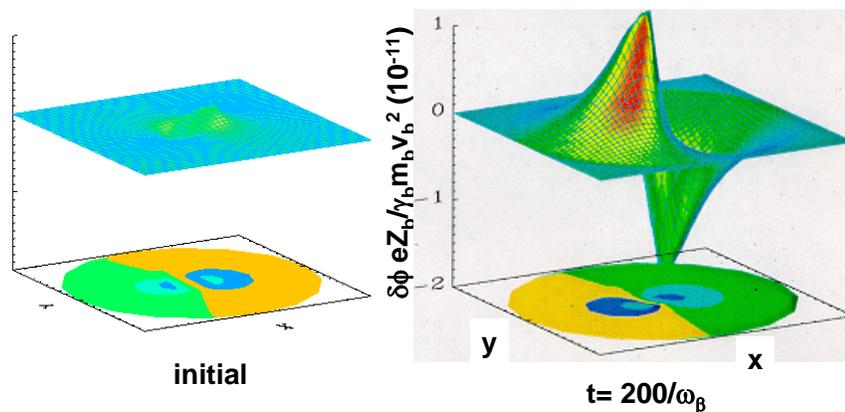
3. Perturbative (δf) Method: PPPL code BEST

E.g., Electron-Proton 2-Stream Instability, growing from initial noise
Dipole “surface mode” can be destabilized by background electrons

For each particle,
 $f_i = f_{i0} + \delta f_i$

Evolve δf_i step-by-step along orbit

In a nonlinear calculation, orbit is computed using full field $E_0 + \delta E$



Offers reduced noise for detailed studies of instability, beam-plasma, and electron processes.

Review: Computation: Vlasov Solvers

No particles!

6-D (4-D) grid

- Define density in phase space, $f(x,y,z, v_x, v_y, v_z)$
- Evolve by numerically solving Vlasov Equation

Advantages:

- Get smooth distribution at all times, no numerical collisions.

Disadvantages:

- Grid very large compared to PIC code with same resolution (e.g. $128^2 = 16.3k$; $128^4 = 268M$).
- Advancing each cell requires interpolation every time step.

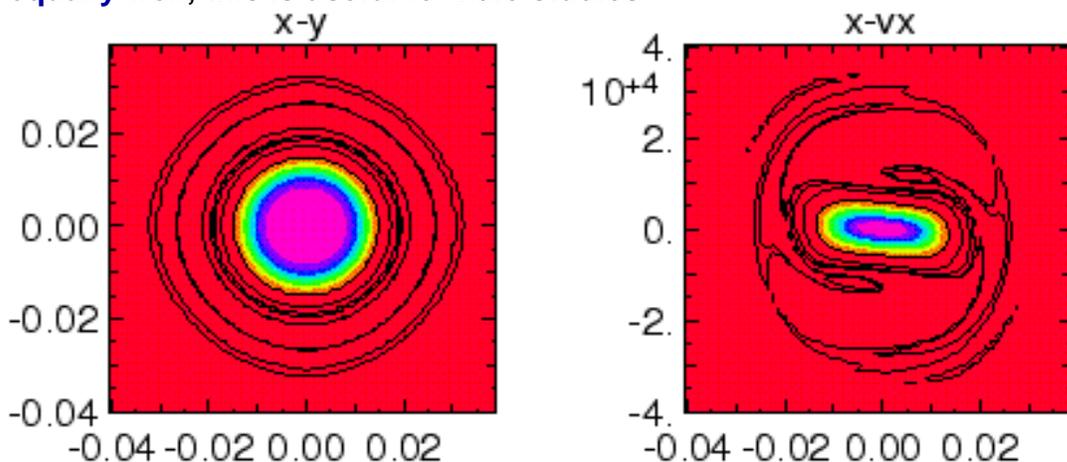
10

4. Semi-Lagrangian Vlasov Solver (SLV)

The distribution function $f(\underline{x}, \underline{v})$ is retained at nodes of a 4-D mesh

The calculation reaches backward in time along a characteristic (orbit in phase space) to obtain the current value of f at each node

Thus, **Low-density and high-density regions of phase space are tracked equally well**; this is useful for halo studies



Black contour lines: 0.1, 0.01, 0.001, 0.0001, and 0.00001 of peak

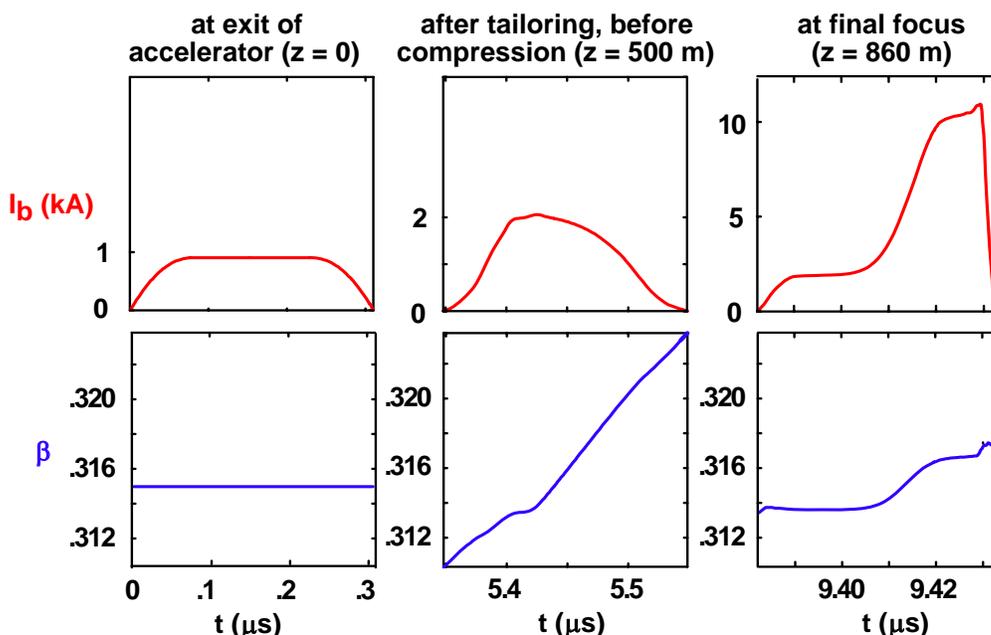
11

5. Moment methods (e.g. CIRCE)

- **Physics model:**
 - **longitudinal:** hundreds of “slices” in a Lagrangian fluid model
 - **transverse:** each slice evolved using envelope and centroid equations
- **Applications:**
 - acceleration and compression schedules, “ears”
 - error tolerances
 - beam sensing and steering
- **Limitations (perhaps not fundamental ...):**
 - no model for emittance growth or phase-mixing of “mismatch”
 - no module impedance model
 - slow variation along beam assumed

12

The beam's current and velocity must be tailored to achieve compression and pulse shaping (CIRCE calculation)



13

Testing: A Case Study on Validation

Rami Kishek

Quote of the Day

“After you break up with your girlfriend, it takes twice as much time to get over her as the time you actually spent together!”

- a friend of mine

“After you think you’re done writing a program, it takes you twice as much time to debug it as the time you actually spent writing it.”

- my rule of thumb

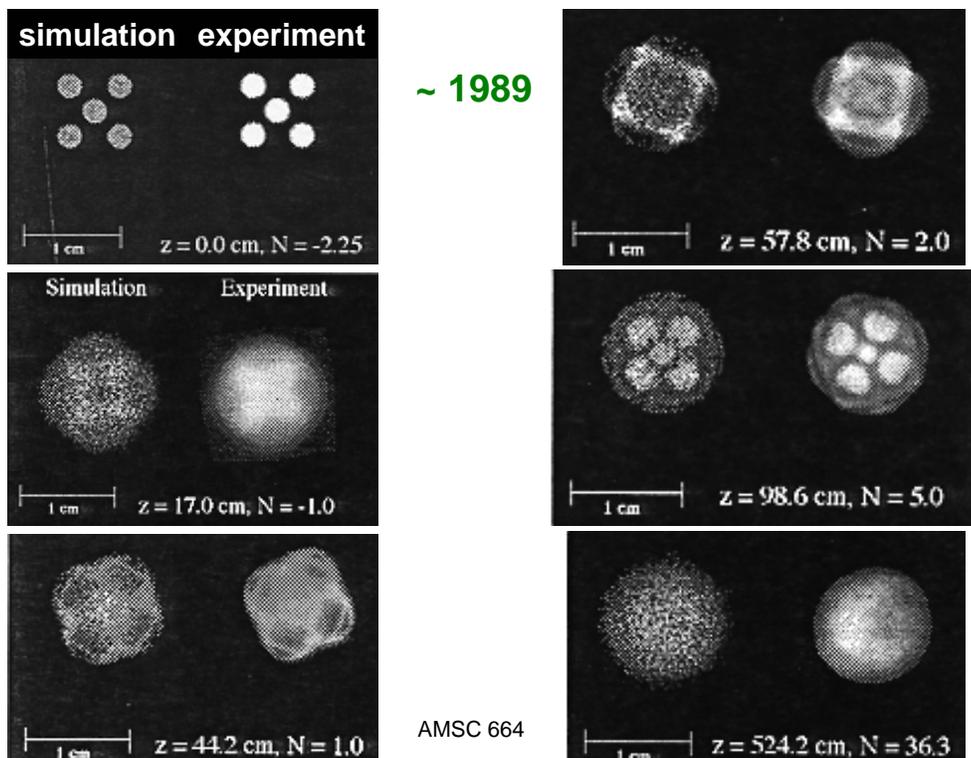
WARP Code

- Electrostatic Particle-in-Cell (PIC) Code.
- Flexible Geometries: 2-D (x-y, r-z) and 3-D.
- Contains various accelerator models for representing external lenses.
- Includes many different Poisson solvers (SOR, FFT, multi-grid, adaptive meshing). Subgrid interpolation allows for curved boundaries.
- Follows the beam around bends.

AMSC 664

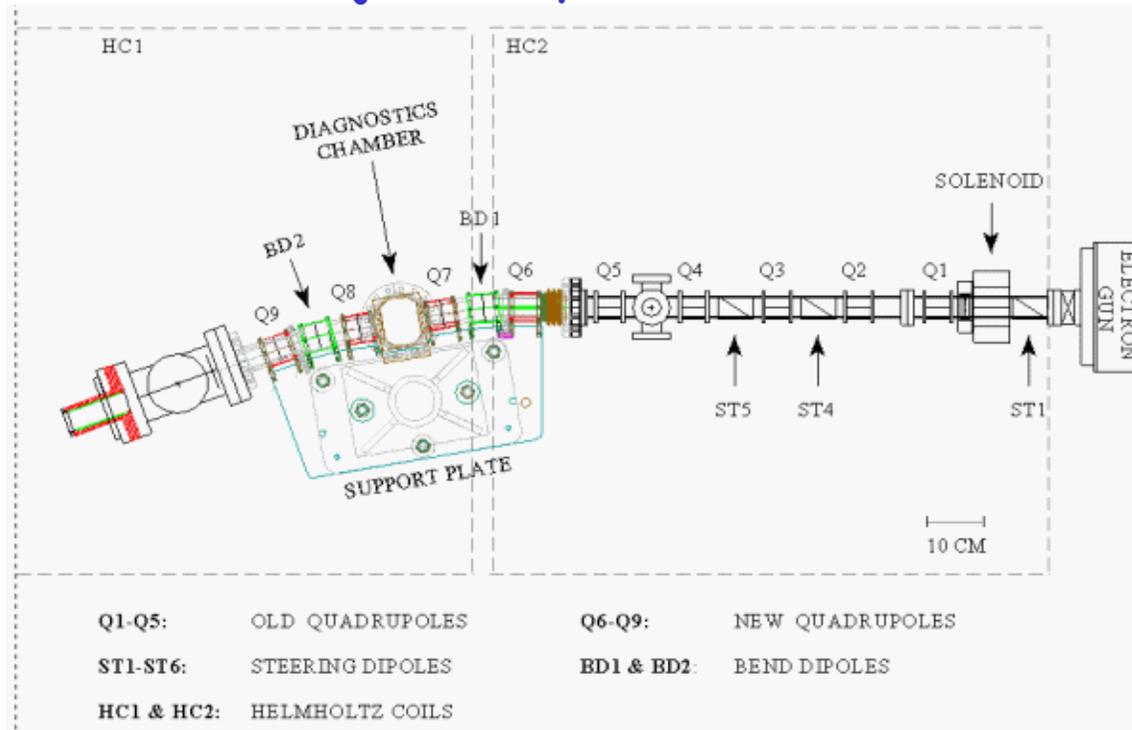
3

5-Beamlet Experiment



4

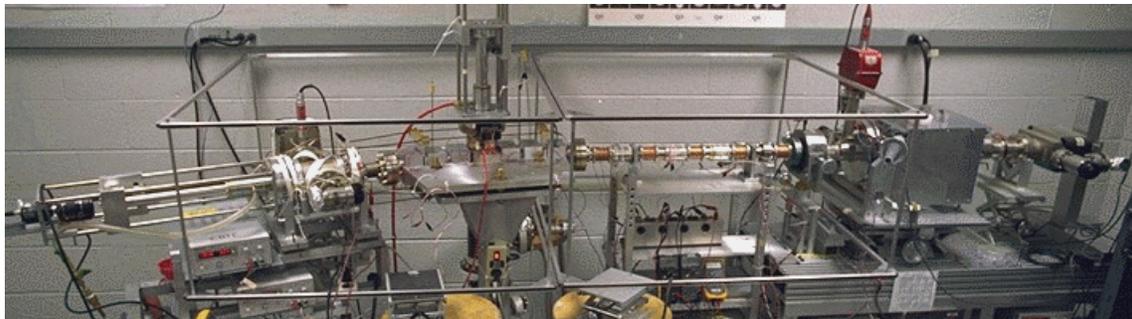
e-Beam Injector Experiment, 1998-1999



AMSC 664

5

Experimental setup complicated

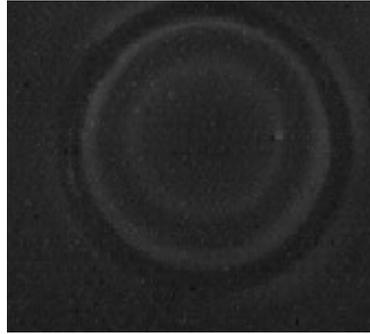


<http://www.ireap.umd.edu/umer/ringdesign/protoinjexper.html>

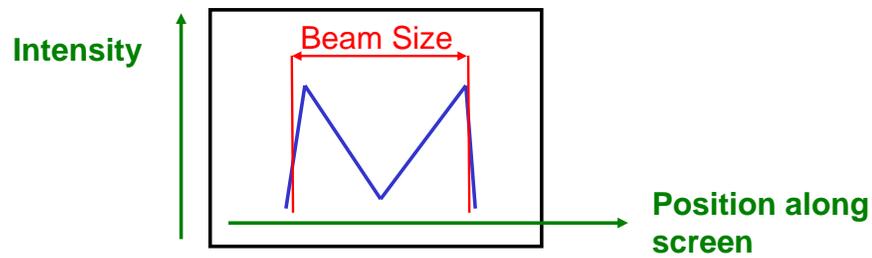
AMSC 664

6

Phosphor screen photo used to determine size



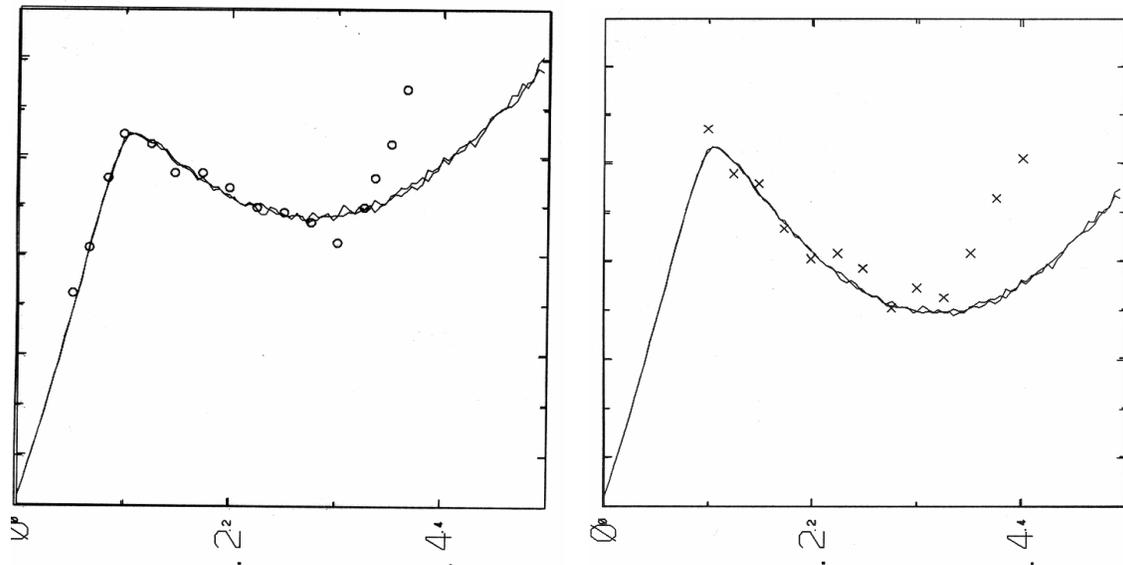
Profile across cross-section is hollowed:



AMSC 664

7

Initial Results Disagree



— Simulation
o o o o Experiment
x x x x

AMSC 664

8

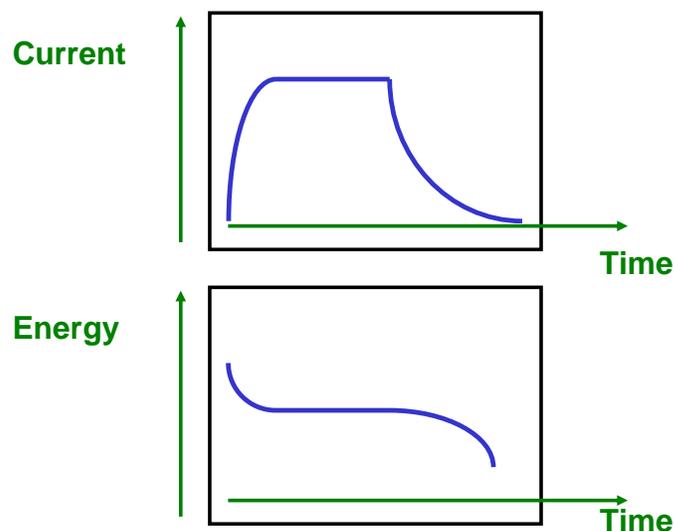
Attempted different models to reproduce data

1. Slightly varied initial conditions
2. Slightly varied magnet strength
3. Refined description of magnet fields
4. Added relativistic effects
5. Superimposed effect of beam ends

AMSC 664

9

Beam Ends - a Clever Explanation

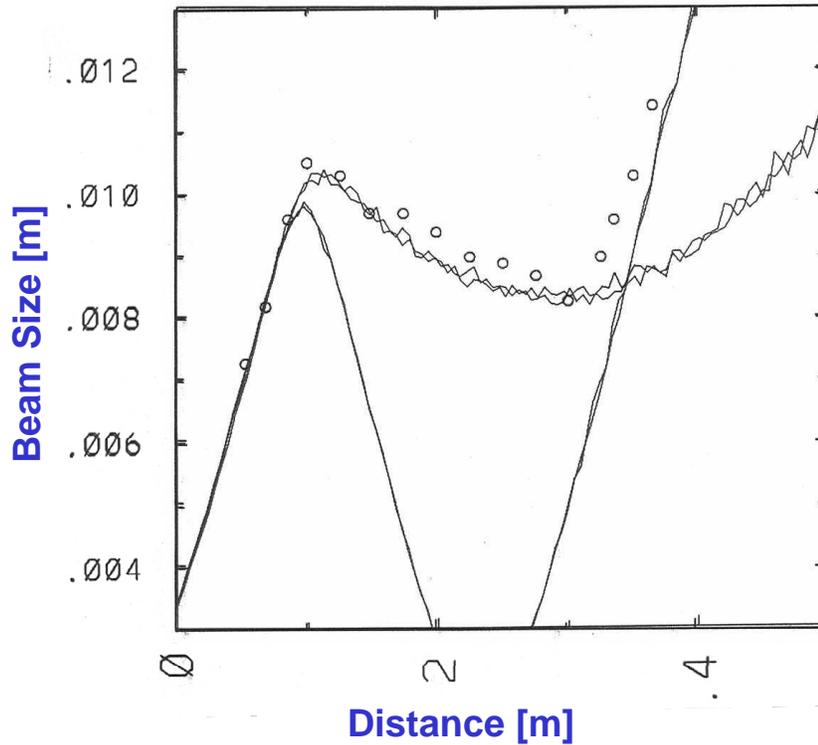


P-screen an **Integrating Diagnostic**

AMSC 664

10

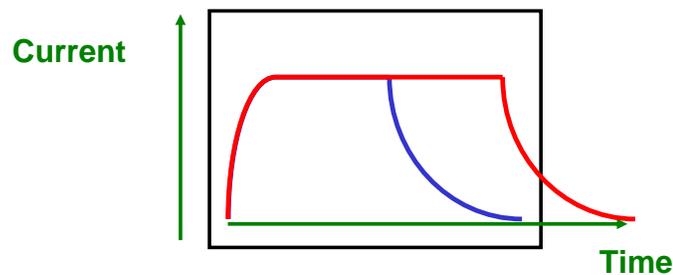
Near Perfect Fit



11

Beam Ends - an Explanation?

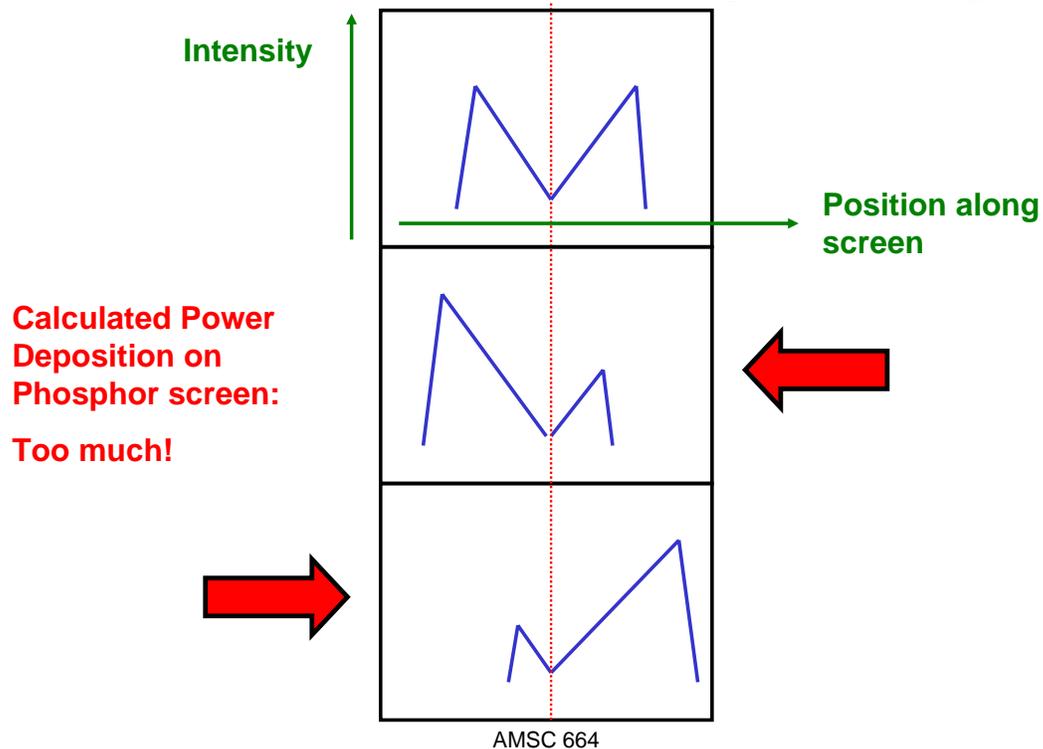
- Perfect agreement between simulation code and experiment ...
- ... without either of them agreeing with reality
- Testing the Hypothesis: Experiment with different pulse lengths



AMSC 664

12

One day in the lab, however, caught a strong hint



13

Attempted Explanations of Disagreement

- Wrong magnetic fields or initial conditions - **Sensitivity Studies** (Solenoid Strength, Initial Slope, Emittance)
- Possibility of head and tail particles smearing measurement - P-screen an **Integrating Diagnostic**
- Code problems? Benchmarked some other published experiment - OK.
- Experiment: P-screen damage? Power deposition

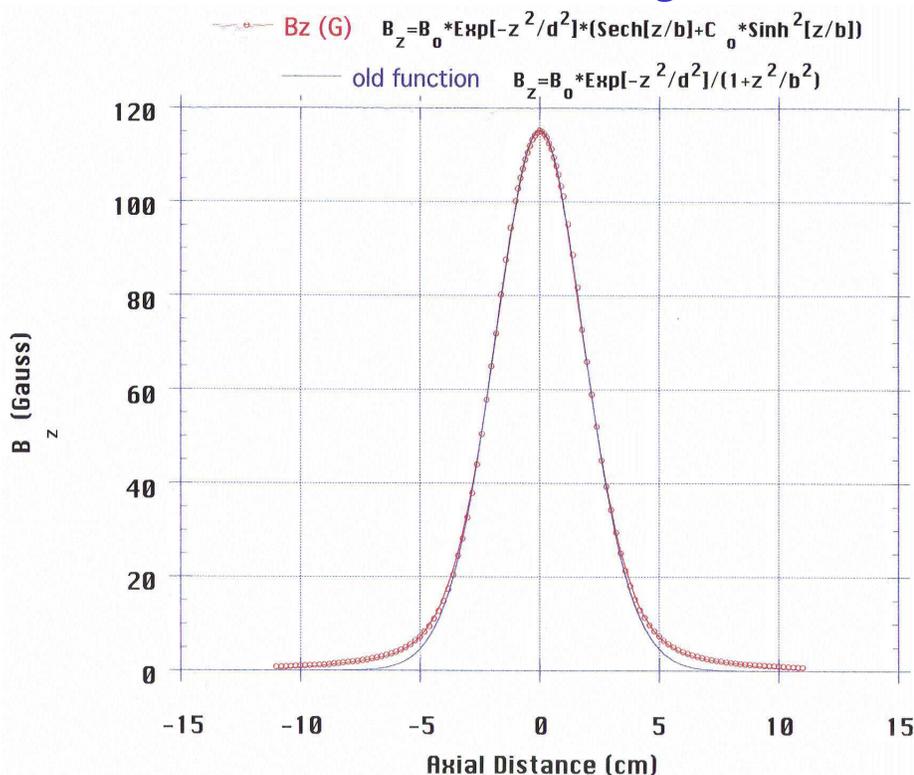
Opened Up Experiment: found many things wrong

1. Phosphor-screen was damaged
2. Initial conditions (measured using screen) were wrong (by factor of 2)!
3. Hence, magnets were also set for wrong values in experiment
4. Magnetic field used in simulation not exactly as experiment
5. Camera system dated to 1980s, needed replacement
6. Processing of experimental data slow – used image processing to automate and enhance

AMSC 664

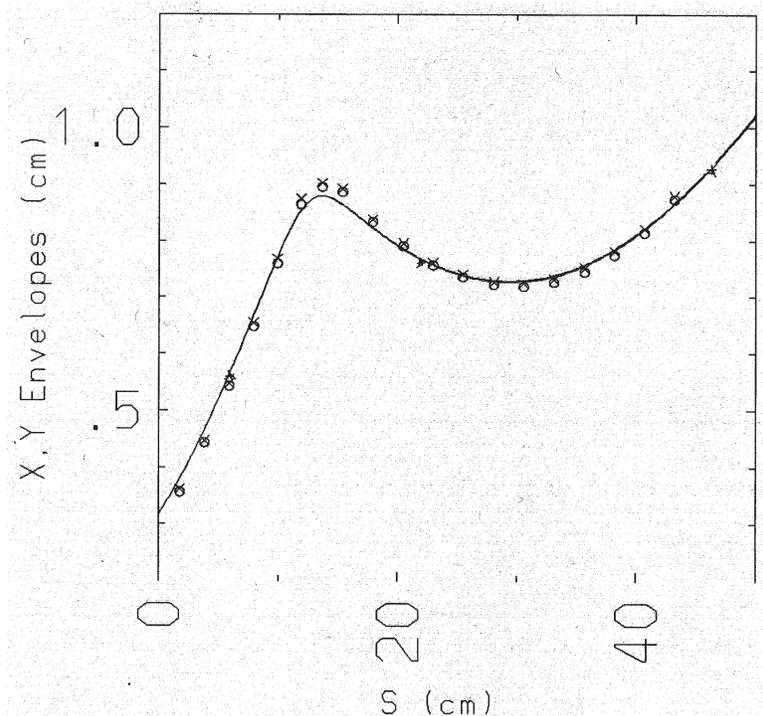
15

Another Problem: Solenoid Magnet Field Profile



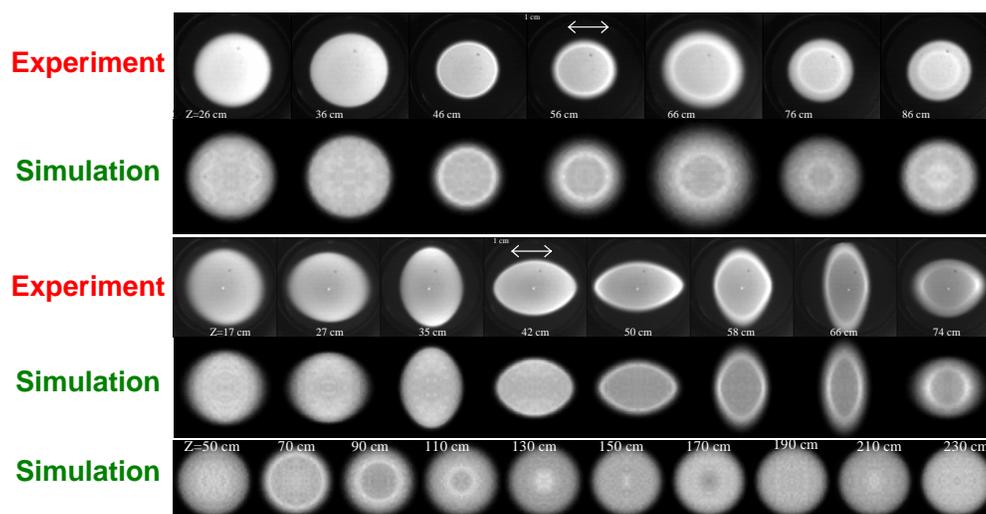
16

Revision of experiment and simulation resulted in much better agreement



17

Agreement also in density profile



S. Bernal, et. al., PRL, **82**, 4002 (1999).