J.J. Barnard and S.M. Lund, USPAS, Jan. 2004

Simulation Techniques for Intense Beams

Outline

1. Why Numerical Simulation

2. Classes of Intense Beam Simulations
   - A. Particle Methods
   - B. Distribution Methods
   - C. Moment Methods.

3. Numerical Methods
   - A. Discritizations
     - Derivatives
     - Integrals
   - B. Application to Moment Methods
     - Euler and Runge-kutta advances
     - Example Application

4. Numerical Methods for Particle and Distribution Metho
   - A. Field discritizations
   - B. Particle Methods and "PIC" Codes for solving Vlasov's equation.
     - Leapfrog advance
     - Field solution
     - Particle weighting
     - Advance cycle
     - Initialization
     - Numerical convergance
     - Examples

C. Distribution Methods

- Similarities with particle methods

Will not cover in class - only mention qualitatively

- Distribution advance
- Examples

5. Overview of the WARP code

References.

Simulation Techniques for Intense Beams

§1 Overview: Why Numerical simulation?

- Builds intuition on intense beam dynamics.
  The primary purpose of simulation is to gain insight on trends, sensitivities, etc. – not specific numbers!
  In contrast to laboratory experiments:
  - Full nonintrusive beam diagnostics are possible
  - Effects can be turned on and off.

  This insight can also be applied to motivate approximations in analytical analysis.
- Allows analysis of more realistic situations that can be intractable in analytical analysis.
  - Realistic geometries
  - Non-ideal distributions
  - Combined effects
  - Large amplitude (nonlinear) effects.

- Expected performance of specific machines can be quantified.

- Computers and numerical methods / libraries are becoming more powerful enabling both analysis of more realistic problems and better numerical convergance.
  - Processor speed increasing
  - Parallel machine architectures
  - Memory larger

  } Hardware bigger and faster

  - Numerical methods improved
  - Libraries of debugged code modules

  } Software more developed

S.M. Lund

Although simulations are increasingly powerful and valuable in the analysis of intense beams, simulations should not be used to exclusion:

- Parametric scaling is <u>very</u> important in machine design. Often it is hardest to understand what specific choices should be made in physical aperture sizes, etc. Although scaling can be explored with simulation. Analytical analysis often best illustrates the trade offs, sensitivities, and relevant combinations of parameters.

- Concepts often fail due to limits of technology (fabrication tolerances, material failures, unanticipated properties, ...) and hence full laboratory testing is vital. Many understood classes of error can be probed with simulation though – unanticipated sources are most dangerous.

- Economic realities often severely limit what can be constructed and such limitations must be understood so you do not simulate something unobtainable.

The highest understanding and confidence is achived when results from analytic theory, numerical simulation, and experiment all converge.

These days numerical simulation skills also make you highly employable in many areas of accelerator and beam physics!

Numerous programing languages are employed in numerical simulations of intense beams. The most common today are:

Fortran, Fortran 90
C, C++
Java

Results are analyzed with a variety of graphics packages:

NCAR, gist, gnuplot, idl, narcisse, ...

Perhaps the most modern and flexible way simulation packages can be structured is to link routines programmed in fast, compiled code with an interactive interpreter such as:

python, basis, yorick

- Allows routines to be coded in mixed languages and (for python) in modern, object-oriented structure.
- Flexible reconfiguration of code modules possible to adapt for specific (unanticipated) needs
  - reduces need for re-compilation and cumbersome structures for special uses.
  - aids cross-checking problems and debugging
- "Steering" of code during runs to address unanticipated effects.

Discussing particular programing languages and graphics packages is beyond the scope of this class. Here our goal is to survey numerical simulation methods employed without presenting details of specific implementations

However,— We will show examples based on the "WARP" particle-in-cell code developed for intense beam simulation at LLNL and LBNL.

WARP - Named since it works on a "warped" Cartesian mesh with bends.

WARP is a family of particle-in-cell code tools built around a common python interpeter for flexable operation. It is optimized for the simulation of intense beams with self-consistent electrostatic space-charge forces

More on WARP later after discussion of methods etc.

J.J. Barnard and S. M. Lund
USPAS, Jan 2004

§2 Classes of Intense Beam Simulations

There are 3 distinct classes of modeling of intense ion beams applicable to numerical simulation

1/ Particle Methods
2/ Distribution Methods
3/ Moment Methods

All of these draw heavily on methods developed for the simulation of neutral plasmas.
The main differences:

- Lack of overall charge neutrality
  - Single species typical (though electron + ion simulations are common too.
- Directed motion of the beam.
- Applied field descriptions are setup for beam optical elements and accelerating structures.

Review/contrast these methods before discussing specific numerical implementations.

1/ Particle Methods

Classical point particles are advanced with self-consistent interactions given by the Maxwell Equations.

- Most general: If actual number of particles are used, this is approx. the physical beam.

Equations of motion (time domain, 3d, for generality)

i'th particle:

$$\boxed{\begin{aligned} \frac{d\vec{p}_i}{dt} &= \vec{F}_i = q\left(\vec{E} + \frac{d\vec{x}_i}{dt} \times \vec{B}\right) \\ m\gamma_i \frac{d\vec{x}_i}{dt} &= \vec{p}_i \; ; \quad \gamma_i = \left[1 + \vec{p}_i^{\,2}/m^2c^2\right]^{1/2} \end{aligned}}$$

initial conditions:

$\{\vec{x}_i(t=0), \vec{p}_i(t=0)\}$

particle orbits

$\vec{x}_i(t), \vec{p}_i(t)$ solved as a function of time.

Fields (Electromagnetic in most general form)

$$\nabla \cdot \vec{E} = \frac{\rho}{\varepsilon_0}$$

charge density

$$\rho(\vec{x},t) = \rho_{ext}(\vec{x},t) + \sum_i q\,\delta[\vec{x} - \vec{x}_i(t)]$$

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

external (applied) — particle beam

$$\nabla \times \vec{B} = \mu_0 \vec{J} + \mu_0\varepsilon_0 \frac{\partial \vec{E}}{\partial t}$$

current density

$$\vec{J}(\vec{x},t) = \vec{J}_{ext}(\vec{x},t) + \sum_i q \frac{d\vec{x}_i(t)}{dt}\,\delta[\vec{x} - \vec{x}_i(t)]$$

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

+ Boundary Conditions on $\vec{E}, \vec{B}$.

2/ Distribution Methods:

- Based on reduced (statistical) continuum models of the beam.
- Two classes (microscopic) kinetic models and (macroscopic) fluid models.
- Here distribution means a function of cont. variables e.g. $f(\vec{x},\vec{p},t)$ in Vlasov model.

2a Kinetic Model (Vlasov, neglect collisions)

- Obtained from statistical averages of particle formulation

Equations of Motion

$$\boxed{\left\{\frac{\partial}{\partial t} + \vec{v}\cdot\frac{\partial}{\partial \vec{x}} + q\left(\vec{E} + \vec{v}\times\vec{B}\right)\right\} f(\vec{x},\vec{p},t) = 0}$$

$$\vec{v} = \frac{\vec{p}}{\gamma m} = \frac{\vec{p}/m}{\left(1+\vec{p}^2/m^2c^2\right)^{1/2}}$$

initial conditions:

$f(\vec{x},\vec{p},t=0)$

Distribution evolution

$f(\vec{x},\vec{p},t)$ evolved from $t=0$.

$\vec{x},\vec{p},t$ indep. variables.

Fields

Same as particle case with different expressions of ρ, $\vec{J}$:

$$\nabla\cdot\vec{E} = \frac{\rho}{\varepsilon_0} \qquad \rho(\vec{x},t) = \rho_{ext}(\vec{x},t) + q\int d^3p\, f(\vec{x},\vec{p},t)$$

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

external (applied) — beam

$$\nabla\times\vec{B} = \mu_0\vec{J} + \mu_0\varepsilon_0\frac{\partial\vec{E}}{\partial t} \qquad \vec{J}(\vec{x},t) = \vec{J}_{ext}(\vec{x},t) + q\int d^3p\, \vec{v}\, f(\vec{x},\vec{p},t)$$

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

+ Boundary conditions on $\vec{E},\vec{B}$

The Vlasov equation is essentially a continuity equation for an incompressible "fluid" f in 6D phase-space. To see this, note that:

$$\frac{\partial}{\partial \vec{p}} \cdot \vec{v} \times \vec{B} = 0$$

so the Vlasov equation can be expressed as:

$$\boxed{\frac{\partial f}{\partial t} + \frac{\partial}{\partial \vec{x}} \cdot \underbrace{\left(\vec{v} f\right)}_{= \left.d\vec{x}/dt\right|_{orbit}} + \frac{\partial}{\partial \vec{p}} \cdot \underbrace{\left(q\left[\vec{E} + \vec{v} \times \vec{B}\right] f\right)}_{= \left.d\vec{p}/dt\right|_{orbit}} = 0}$$

which is manifestly the form of a continuity equation in 6D phase-space. $\Rightarrow$ probability is not created or destroyed!

The effect of collisions can be added to this formulation by adding a collision operator:

$$\boxed{\frac{\partial f}{\partial t} + \vec{v} \cdot \frac{\partial}{\partial \vec{x}} f + q\left[\vec{E} + \vec{v} \times \vec{B}\right] \cdot \frac{\partial}{\partial \vec{p}} f = \left.\frac{\partial f}{\partial t}\right|_{coll}}$$

For most applications in beam physics

$$\left.\frac{\partial f}{\partial t}\right|_{coll}$$

can be neglected (see Intro. lectures). For exceptional cases specific forms of collision terms can be found in Nicholson, "Intro to Plasma Theory," Wiley 1983.

We will find later that the common Particle-in-Cell (PIC) is <u>not</u> really a particle method but rather a method that uses a collection of smoothed "macro" particles to simulate Vlasov's equation. This roughly can be understood by noting that Vlasov's equation can be interpreted as

$$\frac{d}{dt} f(\vec{x}, \vec{p}, t) = 0$$

total derivative along a test particle.

$\Rightarrow$ Advance test particles in a continuous field "fluid" to erase collisions.

But Remember:

PIC is a method to solve Vlasov's equation <u>not</u> a discrete particle method.

This notion should become clear after these lectures.

2b Fluid Model

- Obtained from further averages of kinetic models.
- Described in terms of "macroscopic" variables (density, Flow velocity, Pressure, ...)
- Models must be closed (truncated) at some order via physically motivated assumptions (cold or negligible heat flow, ...)

Density n: $$n(\vec{x},t) = \int d^3p\, f(\vec{x},\vec{p},t)$$

Flow velocity $\vec{V}$: $$n\vec{V}(\vec{x},t) = \int d^3p\, \vec{v}\, f(\vec{x},\vec{p},t)$$

$\vec{P}$ $$n\vec{P}(\vec{x},t) = \int d^3p\, \vec{p}\, f(\vec{x},\vec{p},t)$$

Pressure Tensor $\mathcal{P}_{ij}$: $$n\mathcal{P}_{ij}(\vec{x},t) = \int d^3p\, [p_i - P_i(\vec{x},t)][v_j - V_j(\vec{x},t)] \cdot f(\vec{x},\vec{p},t)$$

Higher Rank Objects

Equations of Motion (Eulerian approach)

continuity: $$\frac{\partial}{\partial t} n + \frac{\partial}{\partial \vec{x}} \cdot [n\vec{V}] = 0$$

Force: ith comp. $$n\left(\frac{\partial}{\partial t} + \vec{V}\cdot\frac{\partial}{\partial \vec{x}}\right)P_i + \sum_j \frac{\partial}{\partial x_j}\mathcal{P}_{ij} = nq\left[\vec{E} + \vec{V}\times\vec{B}\right]_i$$

Higher order – Models must be truncated.
Simplist: cold fluid ($\mathcal{P}_{ij} \equiv 0$)

Field

Same as the kinetic and particle case with

$$\rho(\vec{x},t) = \rho_{ext}(\vec{x},t) + qn(\vec{x},t)$$

$$\vec{J}(\vec{x},t) = \vec{J}_{ext}(\vec{x},t) + qn(\vec{x},t)\,\vec{V}(\vec{x},t)$$

Aside: Lagrangian Formulations of Distribution Methods

In certain kinetic and especially fluid models it can be convenient to adopt so-called "Lagrangian" models.

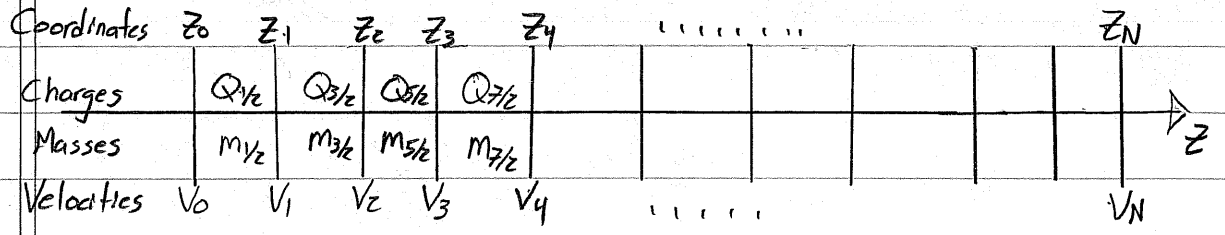
Eulerian fluid model: Flow quantities specified in terms of space ($\vec{x}$) and time (t). $\vec{V} = \vec{V}(\vec{x}, t)$.

Lagrangian fluid model: Identify parts of the flow with objects (material element) and follow the flow in time (t). (shape of elements must generally change in t)

In reality many particle methods are really hybrid Lagrangian too:

- Smoothing methods in the common PIC (Particle In Cell) method (to be covered) typically employ shaped particles that may be thought of as Lagrangian elements representing a Vlasov fluid.

<u>Example:</u> 1D Lagrangian model of the longitudinal evolution of a cold beam.



z_i	slice boundaries	$Q_{i+1/2}$	Fixed	$\frac{Q_{i+1/2}}{m_{i+1/2}} = \frac{q}{m}$
V_i	velocities of slice boundaries.	$m_{i+1/2}$	Fixed	$= \text{const}$ for single species. (set initial cond.)

Solve the equations of motion:

$$\frac{dz_i(t)}{dt} = V_p(t)$$

$$\frac{dV_i(t)}{dt} = \left(\frac{q}{m}\right) E_z(z_i, t)$$

for all the slice boundaries. Several methods might be used to calculate E_z:

1) Take "slices" to have some radial extent modeled by a $\perp$ envelope etc. and deposit the $Q_{i+1/2}$ onto a grid and solve:

$$\nabla^2 \phi = \frac{-\rho}{\varepsilon_0} \qquad E_z = -\frac{\partial \phi}{\partial z}$$

subject to $E_z \to 0$ as $|z| \to \infty$

2) Employ a "g"-factor model

$$E_z = \frac{-g}{2\pi\varepsilon_0} \frac{\partial \lambda}{\partial z}$$

λ calculated from Q_i and radial extent of elements etc.

3) Pure-1d model and use Gauss' Law

//

3/ Moment Methods (with all phase space variations integrated over)

- Most reduced description of intense beam.
- Beam represented by a finite (closed and truncated) set of moments that are advanced from initial values.
  Here by moments, we mean functions of a single variable s or t.
- Such models usually are not fully self-consistent but are commonly used for lattice design.
  - Some special cases such as a stable $\perp$ KV equilibrium distribution are consistent with truncated moment description (rms envelope eqn.)
  - Typically derived from assumed distributions with self-similar evolution.

Many moment models exist. Illustrate with examples for $\perp$ beam evolution:

$$\langle \cdots \rangle_\perp = \frac{\int dx dy \int dx' dy' \cdots f}{\int dx dy \int dx' dy' f}$$

1st order moments

$\vec{x}_c = \langle \vec{x} \rangle_\perp$	centroid coordinate
$\vec{x}_c' = \langle \vec{x}' \rangle_\perp$	centroid angle
$\Delta = \langle \frac{\delta p}{p} \rangle_\perp \equiv \langle \delta \rangle_\perp$	off momentum
$\vdots$	$\vdots$

2nd order moments

x moments	y moments	x-y cross moments	dispersive moments
$\langle x^2 \rangle_\perp$	$\langle y^2 \rangle_\perp$	$\langle xy \rangle_\perp$	$\langle x\delta \rangle_\perp, \langle y\delta \rangle_\perp$
$\langle xx' \rangle_\perp$	$\langle yy' \rangle_\perp$	$\langle x'y \rangle_\perp, \langle xy' \rangle_\perp$	$\langle x'\delta \rangle_\perp, \langle y'\delta \rangle_\perp$
$\langle x'^2 \rangle_\perp$	$\langle y'^2 \rangle_\perp$	$\langle x'y' \rangle_\perp$	$\langle \delta^2 \rangle_\perp$
			....

Note:
Typically convenient to subtract centroid from higher order moments

$$\tilde{x} \equiv x - x_c \quad , \quad \tilde{x}' \equiv x' - x_c' \quad \cdots\cdots$$

$$\tilde{y} \equiv y - y_c \quad \quad \tilde{y}' \equiv y' - y_c'$$

$$\langle \tilde{x}^2 \rangle_\perp = \langle (x - x_c)^2 \rangle_\perp = \langle x^2 \rangle_\perp - x_c^2 \quad , \text{etc.}$$

3rd order moments

Analogous. $\langle x^3 \rangle_\perp$, $\langle x^2 y \rangle_\perp$, ...

Many quantities of physical interest are expressed in terms of moments:

Statistical Beam size:
$$\begin{cases} r_x = 2\langle \tilde{x}^2 \rangle_\perp^{1/2} \\ r_y = 2\langle \tilde{y}^2 \rangle_\perp^{1/2} \end{cases}$$

Statistical Emittances
$$\begin{cases} \varepsilon_x = 4\left[\langle \tilde{x}^2 \rangle_\perp \langle \tilde{x}'^2 \rangle_\perp - \langle \tilde{x}\tilde{x}' \rangle_\perp^2 \right]^{1/2} \\ \varepsilon_y = 4\left[\langle \tilde{y}^2 \rangle_\perp \langle \tilde{y}'^2 \rangle_\perp - \langle \tilde{y}\tilde{y}' \rangle_\perp^2 \right]^{1/2} \end{cases}$$

Equations of Motion

- Can be expressed in terms of moments or combinations of moments that are of physical interest.
- Moments are advanced from specified initial values.

Ex: $\delta = 0$, 2nd order truncation, zero x-y cross moments
linear focusing lattice, elliptical beam with uniform charge.

$$\frac{dx_c'}{ds} = -\kappa_x(s) x_c + \text{image terms} \quad ; \quad \frac{dx_c}{ds} = x_c'$$

$$\frac{dy_c'}{ds} = -\kappa_y(s) y_c + \text{image terms} \quad ; \quad \frac{dy_c}{ds} = y_c'$$

$$\frac{d}{ds}\begin{bmatrix} \langle \tilde{x}^2 \rangle_\perp \\ \langle \tilde{x}\tilde{x}' \rangle_\perp \\ \langle \tilde{x}'^2 \rangle_\perp \\ \langle \tilde{y}^2 \rangle_\perp \\ \langle \tilde{y}\tilde{y}' \rangle_\perp \\ \langle \tilde{y}'^2 \rangle_\perp \end{bmatrix} = \begin{bmatrix} 2\langle \tilde{x}\tilde{x}' \rangle_\perp \\ \langle \tilde{x}'^2 \rangle_\perp - \kappa_x(s)\cdot\langle \tilde{x}^2 \rangle_\perp + Q\langle x^2 \rangle^{1/2} / \left[4\langle \tilde{x}^2 \rangle_\perp^{1/2}\left(\langle \tilde{x}^2 \rangle_\perp^{1/2} + \langle \tilde{y}^2 \rangle_\perp^{1/2}\right)\right] \\ -2\kappa_x(s)\langle \tilde{x}\tilde{x}' \rangle_\perp + 2Q\langle \tilde{x}\tilde{x}' \rangle_\perp / \left[4\langle \tilde{x}^2 \rangle_\perp^{1/2}\left(\langle \tilde{x}^2 \rangle_\perp^{1/2} + \langle \tilde{y}^2 \rangle_\perp^{1/2}\right)\right] \\ 2\langle \tilde{y}\tilde{y}' \rangle_\perp \\ \langle \tilde{y}'^2 \rangle_\perp - \kappa_y(s)\langle \tilde{y}^2 \rangle_\perp + Q\langle \tilde{y}^2 \rangle_\perp^{1/2} / \left[4\langle \tilde{y}^2 \rangle_\perp^{1/2}\left(\langle \tilde{x}^2 \rangle_\perp^{1/2} + \langle \tilde{y}^2 \rangle_\perp^{1/2}\right)\right] \\ -2\kappa_y(s)\langle \tilde{y}\tilde{y}' \rangle_\perp + 2Q\langle \tilde{y}\tilde{y}' \rangle_\perp / \left[4\langle \tilde{y}^2 \rangle_\perp^{1/2}\left(\langle \tilde{x}^2 \rangle_\perp^{1/2} + \langle \tilde{y}^2 \rangle_\perp^{1/2}\right)\right] \end{bmatrix}$$

or using $\frac{d}{ds}\varepsilon_x^2 = 0 = \frac{d}{ds}\varepsilon_y^2$, the 2nd order moment equations can be equivalently expressed as:

$$\frac{d}{ds} r_x' + \kappa_x r_x - \frac{Q}{r_x + r_y} - \frac{\varepsilon_x^2}{r_x^3} = 0 \quad ; \quad \frac{dr_x}{ds} = r_x'$$

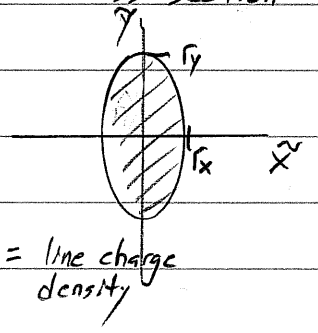
$$\frac{d}{ds} r_y' + \kappa_y r_y - \frac{Q}{r_x + r_y} - \frac{\varepsilon_y^2}{r_y^3} = 0 \quad ; \quad \frac{dr_y}{ds} = r_y'$$

Field

- In moment methods fields are typically calculated using averages over assumed distribution forms.
  - often analytic solutions employed.

Example

Envelope model employed above assumes an elliptical cross-section beam with uniform charge in free space.



For: $\tilde{x}^2/r_x^2 + \tilde{y}^2/r_y^2 < 1$

$$E_{\tilde{x}} = \frac{\lambda \tilde{x}}{\pi\varepsilon_0 r_x (r_x + r_y)}, \quad E_{\tilde{y}} = \frac{\lambda \tilde{y}}{\pi\varepsilon_0 r_y (r_x + r_y)}$$

these forms are employed in the derivation of the moment equations of motion presented (KV Envelope Equation).

Moment models generally form an infinite chain of equations that do <u>not</u> truncate. To be useful the moments must be truncated. Truncations are usually carried out via field model assumptions. For example, the KV model field assumption above leads to a truncation of moments at 2nd order with

$$\varepsilon_x^2 = 16\left[\langle x^2\rangle_\perp \langle x'^2\rangle_\perp - \langle xx'\rangle_\perp^2\right] = \text{const.}$$

$$\varepsilon_y^2 = 16\left[\langle y^2\rangle_\perp \langle y'^2\rangle_\perp - \langle yy'\rangle_\perp^2\right] = \text{const.}$$

Beyond the 3 levels of modeling outlined above:

1/ Particle Methods

2/ Distribution Methods

3/ Moment Methods

There exist numerous "hybrid" models that combine features of several methods. Also, moments are typically calculated in particle and distribution methods as diagnostics.

- Hybrids: δf, gyro-kinetic, ... PIC with shaped particles; etc.

In general:

- Particle and Distribution methods are appropriate for higher levels of detail
- Moment methods are used for rapid iteration of machine design.
- Even within a single method (particle say) there exist many levels of description.
  - Electromagnetic, electrostatic + many field sol. numerical methods
  - 1d, 2d, 3d, ...

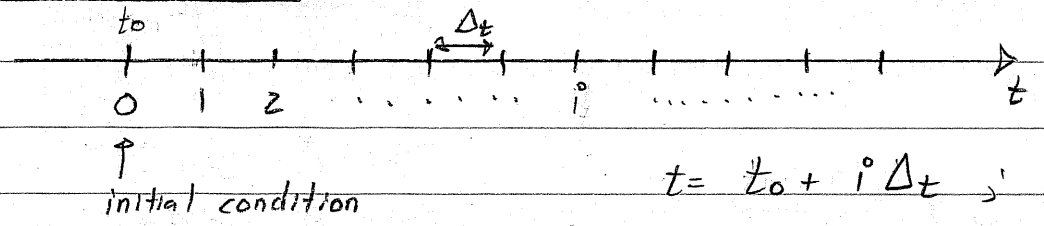
    ⋮
- Employing a hiearchy of models with full diagnostics allows cross-checking (both in numerics and physics) and aids understanding.
  - No single method is best in all cases.

J.J. Barnard and S. M. Lund
USPAS, Jan 2004

§3 Numerical Methods:

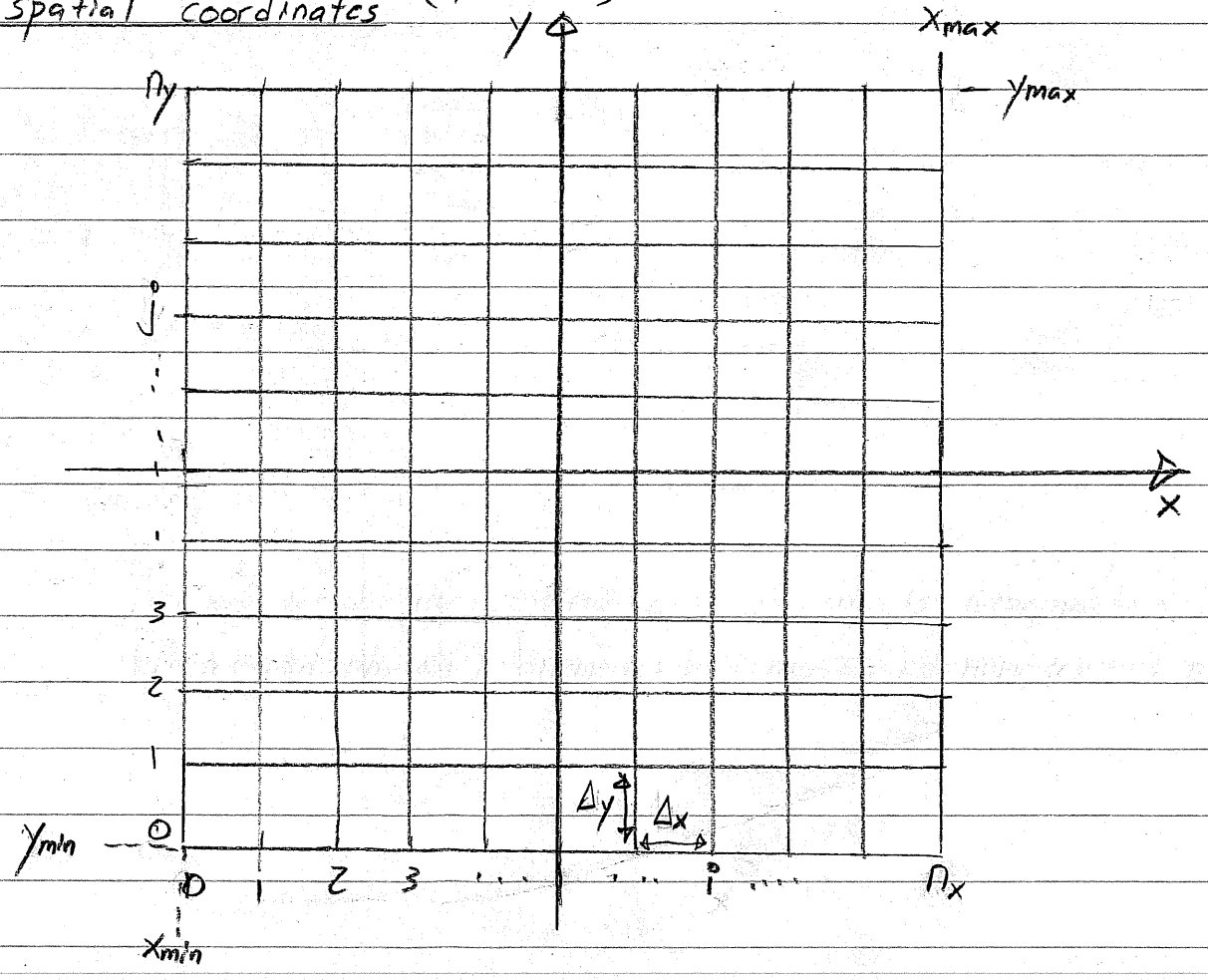
General approach is to discretize independent variables in each of the methods and solve for dependant variables which may in cases also be discretized.

time (or s)



$$t = t_0 + i\,\Delta_t \;;\quad i = 0, 1, 2, 3, \ldots$$

spatial coordinates (transverse)



$$x_i = x_{min} + \Delta x \cdot i \;;\quad \Delta x = (x_{max} - x_{min})/n_x \;;\quad i = 0, 1, 2, \ldots n_x$$

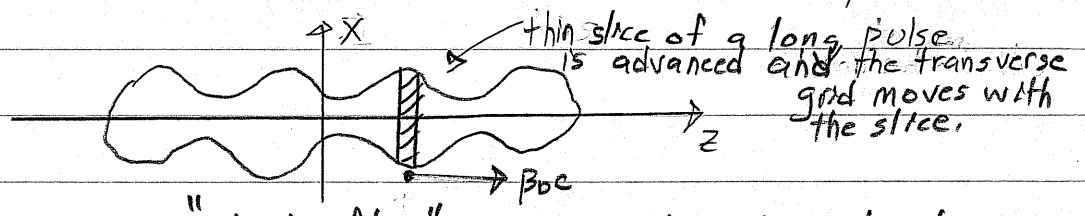
$$y_j = y_{min} + \Delta y \cdot j \;;\quad \Delta y = (y_{min} - y_{max})/n_y \;;\quad j = 0, 1, 2, \ldots n_y$$

analogous for 3d, momentum coordinates, etc...

Spatial coordinates (longitudinal)

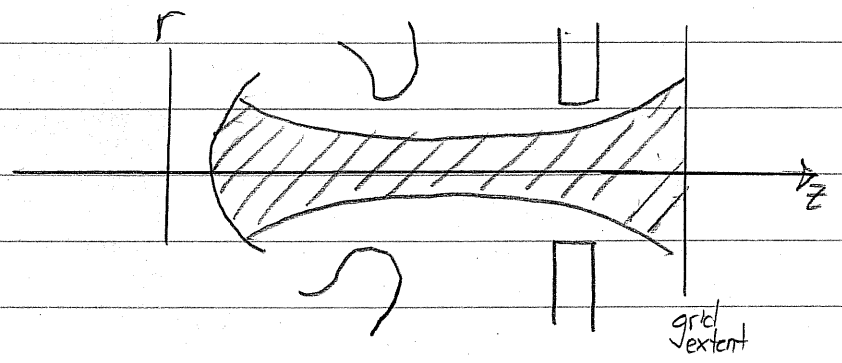
In typical applications may:

- Move a ⊥ thin "slice" of beam along the axial coordinate s of a reference particle.



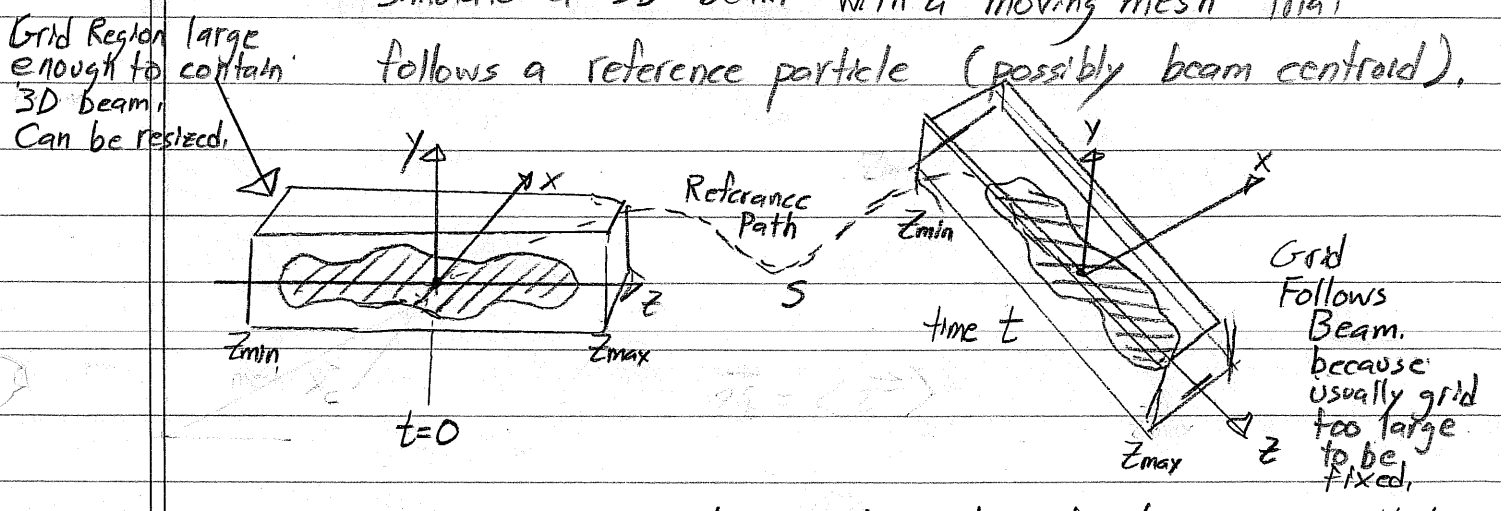
– This "steady flow" approximation is not always possible – some problems are intrinsic 3D.

- Simulate a steady state situation like in a mid-pulse diode where the beam fills the grid.



In these situations the mesh is stationary.

- Simulate a 3D beam with a moving mesh that follows a reference particle (possibly beam centroid).



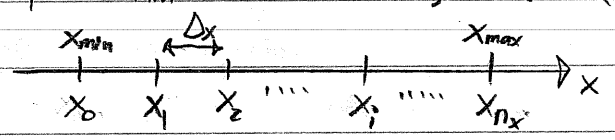
– Grid may be moved in discretized jumps so that applied fields maintain alignment with the grid.

Particle and distribution methods require further levels of discritization in field and other quantities to be rendered practical. Before discussing this we will first discuss basic discrete numerical operations employed. Then we will apply these operations to the simplist case, moment methods, before discussing the more complicated particle and distribution methods.

§3A: Discrete Numerical Operations

Let x represent a space or time coordinate and $f(x)$ some continuous function of x.

$$x_i = x_{min} + \Delta x \cdot i \quad ; \quad \Delta x = (x_{max} - x_{min})/n_x \quad ; \quad i = 0, 1, 2, \ldots, n_x$$



Denote

$$f_i \equiv f(x = x_i) \quad , \text{ etc.}$$

and taylor expand:

$$f_{i+1} = f_i + \left.\frac{\partial f}{\partial x}\right|_i \Delta x + \frac{1}{2!}\left.\frac{\partial^2 f}{\partial x^2}\right|_i \Delta x^2 + \frac{1}{3!}\left.\frac{\partial^3 f}{\partial x^3}\right|_i \Delta x^3 + \cdots$$

$$f_{i-1} = f_i - \left.\frac{\partial f}{\partial x}\right|_i \Delta x + \frac{1}{2!}\left.\frac{\partial^2 f}{\partial x^2}\right|_i \Delta x^2 - \frac{1}{3!}\left.\frac{\partial^3 f}{\partial x^3}\right|_i \Delta x^3 + \cdots$$

Derivatives

Low order derivatives follow immediately

$$\boxed{\begin{aligned} &\text{Foward:} \quad \left.\frac{\partial f}{\partial x}\right|_i = \frac{f_{i+1} - f_i}{\Delta_x} + \mathcal{O}(\Delta_x) \\ &\text{Backward:} \quad \left.\frac{\partial f}{\partial x}\right|_i = \frac{f_i - f_{i-1}}{\Delta_x} + \mathcal{O}(\Delta_x) \end{aligned}}$$

A more accurate, centered, schemes is obtained by subtracting the two expansions: But needs more grid points for accuracy.

$$\boxed{\begin{aligned} &\text{Centered:} \\ &\text{(3 point)} \end{aligned} \quad \left.\frac{\partial f}{\partial x}\right|_i = \frac{f_{i+1} - f_{i-1}}{2\Delta_x} + \mathcal{O}(\Delta_x^2)}$$

The expansions can be relabled ($i \to i+1$, etc) and the resulting equations analyzed to obtain 5-point and other higher-order forms such as:

$$\boxed{\text{5 Point:} \quad \left.\frac{\partial f}{\partial x}\right|_i = \frac{f_{i-2} - 8f_{i-1} + 8f_{i+1} - f_{i+2}}{12\Delta_x} + \mathcal{O}(\Delta_x^4)}$$

Still higher order forms are possible but rapidly become more cumbersome and require more points. Higher order derivatives can be similiarly obtained. For example, adding the two expansions shows:

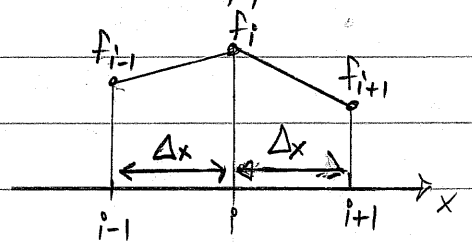
$$\boxed{\left.\frac{\partial^2 f}{\partial x^2}\right|_i = \frac{f_{i+1} - 2f_i + f_{i-1}}{\Delta_x^2} + \mathcal{O}(\Delta_x^2)}$$

Integrals (Quadrature)

Take n_x even, then $\int_{x_{min}}^{x_{max}} dx\, f(\tilde{x})$ can be composed as sub-integrals of form:

$$\int_{x_{i-1}}^{x_{i+1}} d\tilde{x}\, f(\tilde{x})$$

Using a linear approximation: (Trapezoidal Rule)



$$\boxed{\int_{x_{i-1}}^{x_{i+1}} d\tilde{x}\, f(x) = \frac{(f_{i-1} + 2f_i + f_{i+1})}{2}\Delta x + \mathcal{O}(\Delta x^3)}$$

Better approximatations can be found (Simpsons' Rule) using Taylor series expansions and the previous discrete derivatives:

$$f(x) = f_i + \frac{f_{i+1} - f_{i-1}}{2\Delta x} x + \frac{f_{i+1} - 2f_i + f_{i-1}}{\Delta x^2} x^2 + \cdots$$

giving:

$$\boxed{\int_{x_{i-1}}^{x_{i+1}} dx\, f(x) = \frac{(f_{i+1} + 4f_i + f_{i-1})}{3}\Delta x + \mathcal{O}(\Delta x^5)}$$

Numerous other higher order formulas can be derived but are more complicated.

In the examples given, uniform meshes have been employed and formulas are readily generalized to multi-dimensions. In practice, nonuniform meshes are also often used since many applications need resolution concentrated in particular regions. Unfortunately, this makes the differencing formulas more complicated. Particularly with respect to ordering errors. An example of derivative calculations on a nonuniform 1d grid is included in the homework to illustrate methods.

§3B Numerical Solution of Moment Methods

We now have the tools to discuss the numerical solution of moment methods. The moment equations may always be written as an N-dimensional set of coupled 1st order ordinary differential equations:

$$\boxed{\begin{aligned} \vec{M} &= (x_c, \langle \tilde{x}^2 \rangle_\perp, \ldots\ldots) \\ \frac{d\vec{M}}{ds} &= \vec{F}(\vec{M}, s) \end{aligned}}$$

// Example:

Axisymmetric envelope equation for a continuously focused beam

$$\frac{d^2R}{ds^2} + k_{\beta 0}^2 R - \frac{Q}{R} - \frac{\varepsilon_x^2}{R^3} = 0 \quad , \quad k_{\beta 0}^2, Q, \varepsilon_x^2 \text{ constants.}$$

$$\frac{d}{ds}\begin{bmatrix} R' \\ R \end{bmatrix} = \begin{bmatrix} -k_{\beta 0}^2 R + \frac{Q}{R} + \frac{\varepsilon_x^2}{R^3} \\ R' \end{bmatrix}$$

///

Euler's Method:

Apply the forward diff. formula

$$\left.\frac{d\vec{M}}{ds}\right|_i = \frac{\vec{M}_{i+1} - \vec{M}_i}{\Delta_s} + \mathcal{O}(\Delta_s) = \vec{F}(\vec{M}_i, s_i)$$

1st order Euler Method:

$$\boxed{\vec{M}_{i+1} = \vec{M}_i + \Delta_s \vec{F}(\vec{M}_i, s_i) + \mathcal{O}(\Delta_s^2)}$$

The moments can then be advanced from their initial values.

Note that for N_s steps will lead to a total error $\sim N_s \cdot \mathcal{O}(\Delta_s^2) \sim \frac{(s_{max}-s_{min})}{\Delta_s} \cdot \mathcal{O}(\Delta_s^2) \sim \mathcal{O}(\Delta_s)$.

- Error decreases only linearly with step size.
- Numerical work for each step is only 1 evaluation of $\vec{F}$.

<u>Definition</u>

A discrete advance with error $\mathcal{O}(\Delta_s^n)$ is an $n-1$'th order method.

Thus Euler's method is a 1st order method.

Higher order methods are generally used for ODE's

$$\boxed{\frac{d\vec{M}}{ds} = \vec{F}(\vec{M}, s)}$$

<u>Runge-Kutta Method:</u>

Integrate:

$$\vec{M}_{i+1} = \vec{M}_i + \int_{s_i}^{s_{i+1}} ds\, \vec{F}(\vec{M}, s)$$

Approximate $\vec{F}$ by a taylor expansion through the midpoint of the interval. The linear term integrates to zero leaving: $\vec{F}(\vec{M}, s) = \vec{F}(\vec{M}_{i+1/2}, s_{i+1/2}) + \left.\frac{\partial \vec{F}}{\partial s}\right|_{i+1/2} \cdot s + \cdots$

$$\Rightarrow \quad \vec{M}_{i+1} = \vec{M}_i + \vec{F}(\vec{M}_{i+1/2}, s_{i+1/2}) \cdot \Delta_s + \mathcal{O}(\Delta_s^3)$$

Note: only need $\vec{M}_{i+1/2}$ to $\mathcal{O}(\Delta_s^2)$ for $\mathcal{O}(\Delta_s^3)$ accuracy. Apply Euler's method for the 2-step procedure:

2nd Order Runge-Kutta Method:	$\vec{K} \equiv \Delta_s \vec{F}(\vec{M}_i, s_i)$ $\vec{M}_{i+1} = \vec{M}_i + \Delta_s \cdot \vec{F}\left(\vec{M}_i + \frac{\vec{K}}{2}, s_i + \frac{\Delta_s}{2}\right) + \mathcal{O}(\Delta_s^3)$

- Requires 2 evaluations of $\vec{F}$ per advance.
- 2nd order accurate in Δ_s.

Higher order Runge-Kutta Schemes are analogously derived from various quadrature formulas. Many such formulas are derived in standard numerical texts.

- Typically, methods with local error $\mathcal{O}(\Delta s^{N+1})$ will require N evaluations of $\vec{F}$.

For beam physics, many methods are employed to advance moments and particle orbits. A general survey of these methods is beyond the scope of this lecture. But some general comments can be made:

- Many high-order methods with adaptive stepsize exist that refine accuracy to specified tolerances and are optimized for specific classes of equations.
- Choice of methods often involves numerical work and stability considerations.
- Certain methods can be formulated to exactly preserve relevant single-particle invariants.
  - "Symplectic" methods preserve Hamiltonian structure of dynamics.
- Accelerator problems can be demanding due to multiple frequency scales and long tracking times/distances.

Numerical stability is key — but is difficult to analyze in general.

One simple test often used is a "Numerical Reversibility" check. In this method, the final value of an advance is used as an initial condition. Then the problem is run backwards to the original start point and deviations analyzed.

- Often a stringent test of accuracy.
- Will ultimately fail due to roundoff errors and cases where there is a sensitive dependance on initial conditions.
- Orbits can be wrong but qualitatively right ..... we will quantify this notion better later. So lack of full convergance does not necessarily mean that useless results will be obtained.

We now briefly overview an application of Moment equations, namely the KV envelope equations, to a practical high current transport lattice that was designed for heavy ion fusion applications at Lawrence Berkeley National Laboratory.

Moment Equation Application: Pep. RMS Envelope Equations

Neglect image charges and nonlinear self-fields (emittance constant) to obtain moment equations for the evolution of the beam envelope radii

$$\frac{d^2 r_x}{ds} + \kappa_q r_x - \frac{2Q}{r_x + r_y} - \frac{\varepsilon_x^2}{r_x^3} = 0$$

$$\frac{d^2 r_y}{ds} - \kappa_q r_y - \frac{2Q}{r_x + r_y} - \frac{\varepsilon_y^2}{r_y^3} = 0$$

statistical beam envelope radii

$$r_x = 2\sqrt{\langle x^2 \rangle}$$

$$r_y = 2\sqrt{\langle y^2 \rangle}$$

$$Q = \frac{qI}{2\pi\varepsilon_0 mc^3 \gamma_b^3 \beta_b^3}$$

... **Dimensionless Perveance**
measures space-charge strength

$$\varepsilon_x = 4[\langle x^2 \rangle \langle x'^2 \rangle - \langle xx' \rangle^2]^{1/2}$$

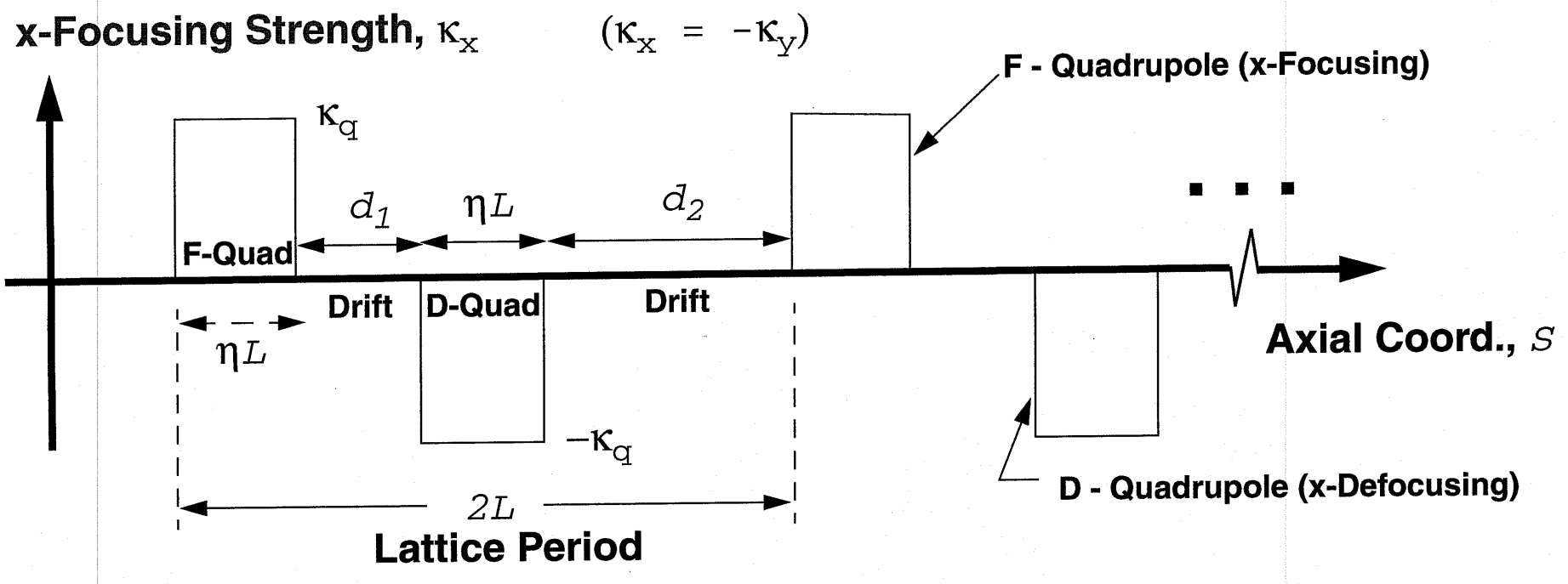
($\varepsilon_{xn} = \gamma_b \beta_b \varepsilon_x$ **normalized**)

... **RMS Edge Emittance**
measures x-x' phase-space area
~ (beam size)sqrt(thermal temperature)

The matched beam solution together with parametric constraints from engineering, higher-order theory, and simulations are used to design the lattice

S.M. Lund 11/

Alternating Gradient Focusing Lattice with Syncopation



Drifts:

$$d_1 = \alpha(1-\eta)2L$$

$$d_2 = (1-\alpha)(1-\eta)2L$$

η = **Quadrupole Occupancy** $(0 < \eta < 1)$

α = **Syncopation Factor** $(0 < \alpha < 1)$

$\alpha = 1/2$ => **Symmetric FODO,** $d_1 = d_2$

$\alpha = 0,1$ => **F touches D**

Focusing Strength:

$$\kappa_q = \begin{cases} \dfrac{1}{[B\rho]}\left|\dfrac{dB_x}{dy}\right| & \text{, Magnetic Quadrupole} \\ \dfrac{1}{[B\rho]V_b}\left|\dfrac{dE_x}{dx}\right| & \text{, Electric Quadrupole} \end{cases}$$

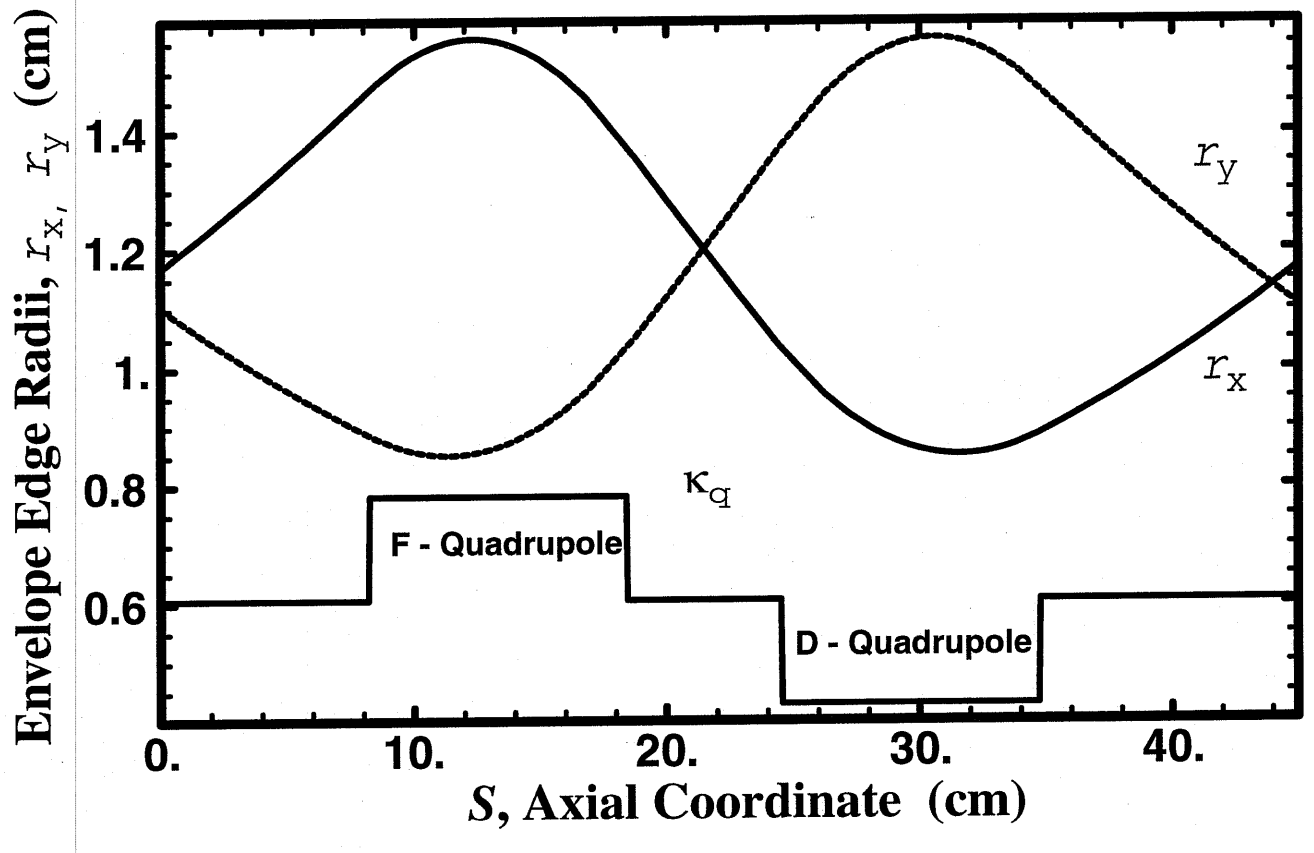
Rigidity

$$[B\rho] = m\gamma_b V_b / q$$

S.M. Lund 12/

Matched Envelope Solution and Particle Phase Advances

$2L$ = 45 cm, Lattice Period



Ion

K^+, E = 2 meV

Current

I = 800 mA

Quadrupole:

occupancy
η = 0.453

syncopation factor
α = 0.248

Low Emittance Case: $\varepsilon_{x,rms}$ = 33.0 mm-mrad, rms edge

$\overline{r_x} = \overline{r_y}$ = 1.184 cm

Max[r_x] = **Max**[r_y] = 1.558 cm **Max**[r_x'] = **-Min**[r_y'] = 38.9 mrad $\sigma_{\beta 0}$ = 80°/Lattice-Period

Min[r_x] = **Min**[r_y] = 0.853 cm **Max**[r_y'] = **-Min**[r_x'] = 58.0 mrad σ_β = 6.88°/Lattice-Period

High Emittance Case: $\varepsilon_{x,rms}$ = 100.0 mm-mrad, rms edge

$\overline{r_x} = \overline{r_y}$ = 1.218 cm

Max[r_x] = **Max**[r_y] = 1.605 cm **Max**[r_x'] = **-Min**[r_y'] = 40.2 mrad $\sigma_{\beta 0}$ = 80°/Lattice-Period

Min[r_x] = **Min**[r_y] = 0.875 cm **Max**[r_y'] = **-Min**[r_x'] = 60.0 mrad σ_β = 19.8°/Lattice-Period

S.M. Lund 13/

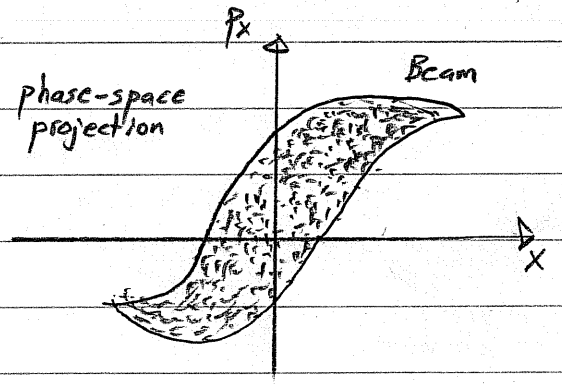
J.J. Barnard and S.M. Lund
USPAS, Jan 04

§4 Numerical Solution of Particle and Distribution Methods

Particle methods – generally not used at high space-charge intensity

Distribution methods – pretered – mainly PIC – for high space-charge. We will motivate why now.

Particle Methods – Why they are not a good choice:



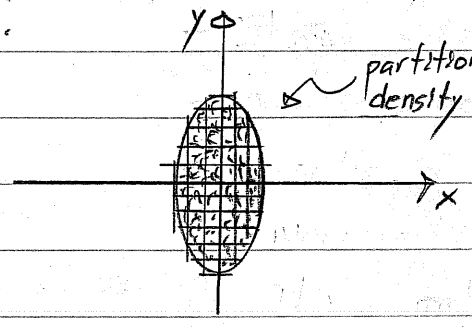
N particle coordinates
$\{\vec{x}_i, \vec{p}_i\}$

Physical Beam: (typical)
$N \sim 10^{10} - 10^{14}$ particles

Although larger problems are possible every year with more powerful computers, processor speed and memory limitations presently limit us to

$$N \lesssim 10^8 \text{ particles.}$$

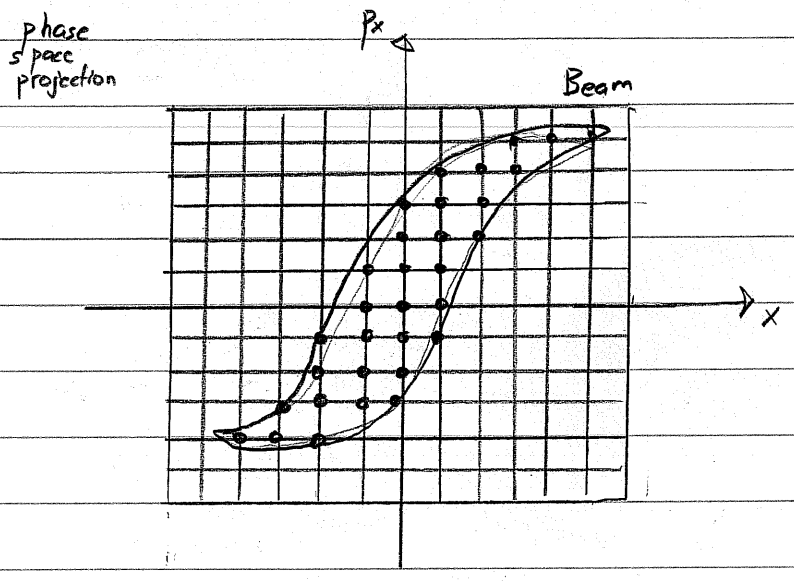
Represent the beam by "macroparticles" and advance macroparticles in time.



- Same q/m ratio as real particle $\Rightarrow$ same single particle dynamics in applied field.
- Results in more collisions due to macroparticles having more close approaches.
  - Enhanced collisionality is unphysical
  - Controlled by smoothing the macroparticle interaction with the self-field. More on this later!

Distribution Methods

Vlasov as an example.



Discrete grid points $\{\vec{x}_i, \vec{p}_i\}$

Advance distribution $f(\vec{x}, \vec{p}, t)$ at discrete grid points in time.

- Continuum distribution advanced on a discrete phase-space mesh.
  - Extreme memory for high resolution. Ex. for 4d x-p_x, y-p_y with 100 mesh points on each axis $\Rightarrow 100^4 = 10^8$ values to store in fast memory.
- Discritization errors can lead to aliasing and unphysical behaviour (negative probability etc.).

Both particle and distribution methods can be broken up into 2 basic parts:

1) Moving particles or distribution evaluated at grid points through a finite time (or axial space) step.

2) Calculation of beam self-fields consistently with the distribution of particles.

In both methods, significant fractions of the run time may also be <u>devoted to diagnostics</u>:

- Moments — can be computationally intensive and may be accumulated frequently for evolution of evolution "histories."
- Phase space projections ("snapshot" in time)
- Fields (snapshot in time)

Diagnostics are also critical!

- Without appropriate diagnostics runs are useless even if correct.
- Must accumulate and present mass data in an understandable format.

Significant source code development may also be devoted to creating (loading) the initial distribution of particles to simulate.

- Such parts of the simulation will usually only take a small fraction of total run time.

Particle Methods – Integration of the Equations of Motion

Higher order methods require more storage and numerical work per time step

- Fieldsolves are expensive, especially in 3D, and several fieldsolves per step can be necessary for higher order accuracy.

$\therefore$

Typically, low-order methods are used for self-consistent space-charge. The "leapfrog" method is the most common.

- Only need to store prior position and velocity
- One fieldsolve per timestep.

Illustrate for

Nonrelativistic equations of motion:

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

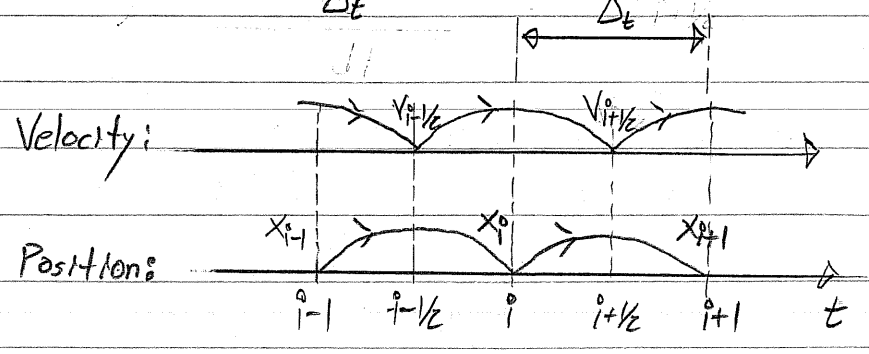
$$\frac{d\vec{x}}{dt} = \vec{v}$$

Leapfrog advance (time centered)

1) $$m\frac{\vec{V}_{i+1/2} - \vec{V}_{i-1/2}}{\Delta_t} = \vec{F}_i$$

2) $$\frac{\vec{X}_{i+1} - \vec{X}_i}{\Delta_t} = \vec{V}_{i+1/2}$$

Note: $\vec{x}$ and $\vec{v}$ advanced out of phase.



For purposes of analysis of the properties of the leapfrog method it is often convenient to write the map in an alternative form:

$$\begin{cases} \dfrac{\vec{x}_{i+1} - \vec{x}_i}{\Delta_t} = \vec{V}_{i+1/2} \\[2ex] i \to i-1 \quad \dfrac{\vec{x}_i - \vec{x}_{i-1}}{\Delta_t} = \vec{V}_{i-1/2} \end{cases}$$

Subtract the equations and apply the other formula:

$$\Rightarrow \quad m\,\frac{\vec{V}_{i+1/2} - \vec{V}_{i-1/2}}{\Delta_t} = \boxed{m\,\frac{\vec{x}_{i+1} - 2\vec{x}_i + \vec{x}_{i-1}}{\Delta_t^2} = \vec{F}_i}$$

Note:

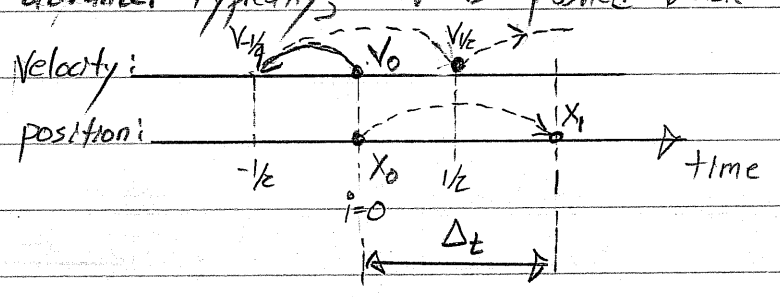
$$\left.\frac{\partial^2 f}{\partial x^2}\right|_i = \frac{f_{i+1} - 2f_i + f_{i-1}}{\Delta_x^2} + \mathcal{O}(\Delta_x^2)$$

$\Rightarrow$ $\vec{x}_{i+1}$ fixed from $\vec{x}_i, \vec{x}_{i-1}, \vec{F}_i$ to $\mathcal{O}(\Delta_t^4)$

<u>Synchronization and desynchronization:</u>

Since $\vec{x}$ and $\vec{v}$ are not evaluated at the same time in the leapfrog method:

- Initial conditions <u>must be adjusted</u> to start the advance. Typically, $\vec{v}$ is pushed back a half cycle.



- When evaluating diagnostic quantities such as moments the particle coordinates and velocities should first be synchronized using a similar procedure.

Another complication is in evaluation of the force which has velocity dependence when magnetic fields are present since $\vec{x}$ and $\vec{v}$ are advanced out of phase in the leap-frog method.

$$\boxed{\vec{F} = q\vec{E} + q\underbrace{\vec{v}}_{\text{Velocity term}} \times \vec{B}}$$

Note:

- Electric field $\vec{E}$ accelerates.
- Magnetic field $\vec{B}$ bends particle trajectory without change in speed $\vec{v}$.

A commonly implemented time centered scheme for this is the following 3-step "Boris" method:

1) <u>Half-step acceleration</u> in electric field

$$\vec{V}^{(1)}_{i+1/2} = \vec{V}_{i-1/2} + \frac{q}{m}\vec{E}_i \cdot \frac{\Delta_t}{2}$$

2) <u>Rotation</u> in magnetic field. Here choose coordinates such that $\vec{B}_i = B_i\hat{z}$, $\omega_{ci} = qB_i/m$

$\parallel B_i$: $V^{(2)}_{z\,i+1/2} = V^{(1)}_{z\,i+1/2}$ [Parallel Magnetic Field to $\vec{v}$ does not bend trajectory]

$\perp B_i$: $$\begin{pmatrix} V^{(2)}_{x\,i+1/2} \\ V^{(2)}_{y\,i+1/2} \end{pmatrix} = \begin{bmatrix} \cos(\omega_{ci}\Delta_t) & \sin(\omega_{ci}\Delta_t) \\ -\sin(\omega_{ci}\Delta_t) & \cos(\omega_{ci}\Delta_t) \end{bmatrix} \begin{pmatrix} V^{(1)}_{x\,i+1/2} \\ V^{(1)}_{x\,i+1/2} \end{pmatrix}$$

3) <u>Half-step acceleration</u> in electric field

$$\vec{V}_{i+1/2} = \vec{V}^{(3)}_{i+1/2} = \vec{V}^{(2)}_{i+1/2} + \frac{q}{m}\vec{E}_i \cdot \frac{\Delta_t}{2}$$

Complication: On startup how does one generate the out of phase $\vec{x}, \vec{v}$ advance from initial conditions?

- Calculate $\vec{E}, \vec{B}$ with initial conditions.
- Move $\vec{v}$ backward a half-step $(-\Delta t/2)$
  - Rotate with $\vec{B}$ a half-step.
  - Deaccelerate a half-step in $\vec{E}$.

Similar comments hold for synchronization of $\vec{x}, \vec{v}$ for diagnostic accumulation.

Now we will look at the numerical properties of the leapfrog advance cycle.

- Only use a simple example to illustrate issues.

Leapfrog Advance Errors and Numerical Stability

To better understand the leapfrog method consider the simple harmonic oscillator:

$$\boxed{\frac{d^2x}{dt^2} = -\omega^2 x \qquad \omega = \text{const.}}$$

Exact Solution $\Rightarrow$ $x = C_0 \cos\omega t + C_1 \sin\omega t$ $\qquad$ C_0, C_1 constants

$= X_0 \cos(\omega t + \psi_0)$ $\qquad$ set from initial conds.

In the leapfrog method:

$$\boxed{\frac{x_{i+1} - 2x_i + x_{i-1}}{\Delta_t^2} = -\omega^2 x_i}$$

Try a solution of form

$$x_i = C_0 \cos\tilde{\omega} t_i + C_1 \sin\tilde{\omega} t_i$$

Substitute

$$x_i = C e^{j\tilde{\omega} i \Delta_t} \qquad j \equiv \sqrt{-1}$$

Use j to not confuse with other index choices.

$$\Rightarrow \quad e^{j\tilde{\omega}\Delta_t} - 2 + e^{-j\tilde{\omega}\Delta_t} = -\omega^2 \Delta_t^2$$

$$2 - 2\cos(\tilde{\omega}\Delta_t) = \omega^2 \Delta_t^2$$

$$\sin^2\left(\frac{\tilde{\omega}\Delta_t}{2}\right) = \frac{\omega^2 \Delta_t^2}{4} \qquad \text{take + root}$$

$$\Rightarrow \quad \boxed{\sin\left(\frac{\tilde{\omega}\Delta_t}{2}\right) = \frac{\omega\Delta_t}{2}}$$

This has solutions for $\omega\Delta_t < 2$ and it is straightforward to show (expansion) that for $\omega\Delta_t$ small:

$$\boxed{\tilde{\omega}\Delta_t = \omega\Delta_t + \frac{(\omega\Delta_t)^3}{24} + \mathcal{O}(\omega\Delta_t^5)}$$

It follows immediately for the leapfrog method:

- For $\omega\Delta_t < 2$ the method is <u>stable</u>
- There is <u>no</u> amplitude error in the integration
- For $\omega dt \ll 1$, the phase error is

Actual phase: $\psi \equiv \omega i \Delta_t$

Simulated phase: $\tilde{\psi} \equiv \tilde{\omega} i \Delta_t \cong \omega i \Delta_t + \frac{(\omega\Delta_t)^3}{24} i$

Error phase: $\Delta\psi = \tilde{\psi} - \psi = \frac{(\omega\Delta_t)^3}{24} i$

Note: i to get to a fixed time $\sim 1/\Delta_t$ and therefore phase errors decrease as $\mathcal{O}(\Delta_t^2)$

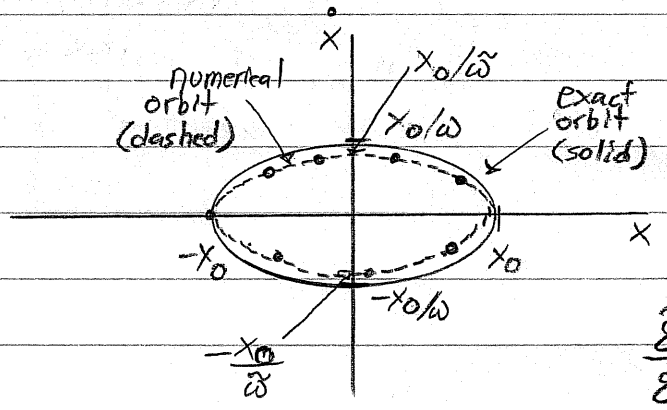
// Example: $\omega = 2\pi/\tau$

$\Delta_t = 0.1\,\tau$: steps for a π phase error $\cong 24\pi/(0.1\cdot 2\pi)^3 \approx 304 \Rightarrow \sim 30.4$ cycles

$\Delta_t = 0.01\,\tau$: steps for a π phase error $\cong 24\pi/(0.01\cdot 2\pi)^3 \approx 3.04\times 10^5 \Rightarrow \sim 3{,}040$ cycles

///

<u>Contrast: Numerical and Actual Orbit</u>



Emittances = (Phase space Areas)/π

exact: $\mathcal{E} = x_0^2/\omega$

Numerical $\tilde{\mathcal{E}} = x_0^2/\tilde{\omega}$

$$\frac{\tilde{\mathcal{E}}}{\mathcal{E}} \cong 1 - \frac{(\omega\Delta_t)^2}{24}$$

The numerical orbit conserves phase space area regardless of the number of steps taken! <u>The slight differences between the numerical and actual orbit can be removed by rescaling ω to account for the discrete step.</u>

- More general analysis of the leapfrog map shows it has "symplectic" structure – meaning it preserves the Hamiltonian nature of the dynamics.
- Symplectic methods are important for long tracking problems (typical in accelerators) to obtain the right orbit structure – Runge-Kutta is not symplectic

Particle Methods – Field solution

The self-consistant calculation of self-fields is of primary importance to accurately simulate intense particle beams. Techniques outlined here are also applicable to distribution methods.

$$\vec{F} = q\vec{E} + q\vec{v} \times \vec{B}$$

Fields can be resolved into externally applied and self (i.e., beam generated) components:

$$\boxed{\begin{array}{ll} \vec{E} = \vec{E}_a + \vec{E}_s & \vec{E}_a, \vec{B}_a : \text{Applied} \\ \vec{B} = \vec{B}_a + \vec{B}_s & \vec{E}_s, \vec{B}_s : \text{Self} \end{array}}$$

- $\vec{E}_a, \vec{B}_a$ applied fields generated from magnets and electrodes
  - sometimes calculated at high resolution in external codes and imported or specified via analytic formulae
  - sometimes calculated from code fieldsolve via applied charges and currents and boundary conditions

- $\vec{E}_s, \vec{B}_s$ self fields generated from beam charges and currents.
  - At high beam intensities can be a large fraction (on average) of the applied fields
  - Important to calculate with realistic boundary conditions.

For simplicity, we restrict analysis to electrostatic problems to illustrate methods:

$$\vec{B} = \vec{B}_a \qquad \text{(specified via another code or theory)}$$

$$\vec{E} = \vec{E}_a + \vec{E}_s \qquad \left(\begin{array}{l}\vec{E}_a \text{ due to biased electrodes and} \\ \vec{E}_s \text{ due to beam space-charge}\end{array}\right)$$

The Maxwell equations to be solved for $\vec{E}$ are

$$\boxed{\begin{array}{ll} \nabla \cdot \vec{E} = \dfrac{\rho}{\varepsilon_0} & + \text{ Boundary conditions on } \vec{E}. \\ \nabla \times \vec{E} = 0 & \end{array}}$$

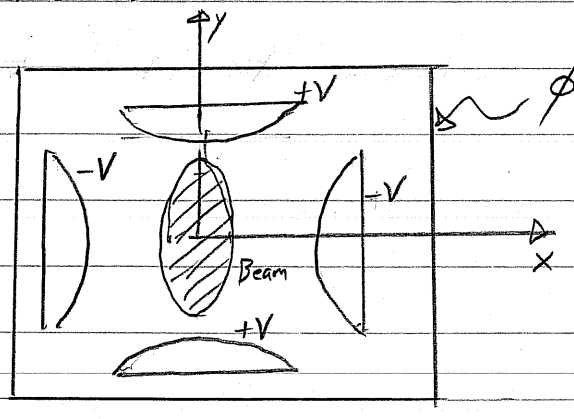
$\nabla \times \vec{E}$ implies that we can always take

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

and we have

$$\boxed{\begin{array}{ll} \nabla^2 \phi = -\dfrac{\rho}{\varepsilon_0} & + \text{ Boundary conditions on } \phi \\ \vec{E} = -\nabla \phi & \end{array}}$$

As an explicit example, it might be necessary to solve (2D) problems like:



ϕ or $\frac{\partial \phi}{\partial n}$ specified on domain boundary or in infinite space.

Formally, the solution to ϕ can be constructed with a Green's Function: (Illustrated for Dirichlet Boundary Conditions)

$$\boxed{\begin{aligned} \nabla^2 G(\vec{x},\vec{x}') &= -4\pi\delta(\vec{x}-\vec{x}') \\ G(\vec{x},\vec{x}')\Big|_{\vec{x}' \text{ on boundaries}} &= 0 \end{aligned}}$$

which gives:

$$\boxed{\phi(\vec{x}) = \underbrace{\int_V d^3x' \frac{\rho(\vec{x}')}{4\pi\varepsilon_0} G(\vec{x},\vec{x}')}_{\text{Self-field component } \equiv \phi_s} - \underbrace{\frac{1}{4\pi}\oint_S d^2x' \, \phi(\vec{x}') \frac{\partial G(\vec{x},\vec{x}')}{\partial n'}}_{\text{Applied Field from electrode potentials } \equiv \phi_a}}$$

$\hat{n}$ = normal to conductor surface

$$\vec{E}_s = -\frac{\partial \phi_s}{\partial \vec{x}} \qquad \vec{E}_a = -\frac{\partial \phi_a}{\partial \vec{x}}$$

- ϕ_a can be calculated in advance and need not be recalculated.
  - may be analytical in simple situations.

Let: $q_M, \vec{x}_i$ = macro-particle charge and coordinate
N_p = Macro particle number.

$$\phi_s = \int d^3x' \frac{\rho(\vec{x}')}{4\pi\varepsilon_0} G(\vec{x},\vec{x}') = \sum_{i=1}^{N_p} \int d^3x' \frac{q_M \delta(\vec{x}'-\vec{x}_i)}{4\pi\varepsilon_0} G(\vec{x},\vec{x}')$$

$$\boxed{\phi_s = \sum_{i=1}^{N_p} \frac{q_M}{4\pi\varepsilon_0} G(\vec{x},\vec{x}_i)}$$

Then the field at the jth macro-particle is

$$\boxed{\vec{E}_{sj} = \left.\frac{\partial \phi_s}{\partial \vec{x}}\right|_{\vec{x}=\vec{x}_j} = \frac{q_M}{4\pi\varepsilon_0} \sum_{\substack{i=1 \\ i\neq j}}^{N_p} \left.\frac{\partial G(\vec{x}, \vec{x}_i)}{\partial \vec{x}}\right|_{\vec{x}=\vec{x}_j}}$$

This, in general, will be a numerically intensive expression to evaluate at <u>each</u> macroparticle.

- $N_p(N_p-1)$ terms to evaluate <u>and G itself will in general be complicated</u>, and may require many costly numerical operations for each term.

- Small number of macroparticles that this procedure may be carried out for will result in a noisy field
  - Enhanced, unphysically high, close approaches (collisions) can change the physics.

- Special "fast multipole" methods based on Greens' functions can reduce the scaling to $\sim N_p$ or $\sim N_p \ln N_p$. However the coefficient is large and smoothing is not easily implimented, usually rendering such methods inferior to gridded methods to be covered shortly.

// Example – Self Fields in free space:

$$G(\vec{x}, \vec{x}') = \frac{1}{|\vec{x}-\vec{x}'|} \quad ; \quad \vec{E}_{sj} = \frac{q_M}{4\pi\varepsilon_0} \sum_{i\neq j}^{N_p} \frac{(\vec{x}_j - \vec{x}_i)}{|\vec{x}_j - \vec{x}_i|^3}$$

//

An alternative Procedure is Needed to:

1) <u>Calculate fields efficiently</u>
2) <u>Smooth interactions</u> to compensate for limited particle numbers.

Approach:

Solve the Maxwell Equations on a discrete spatial grid and then smooth the interactions calculated from the gridded field. We will outline this method for 2D interactions, 3D is analogous.

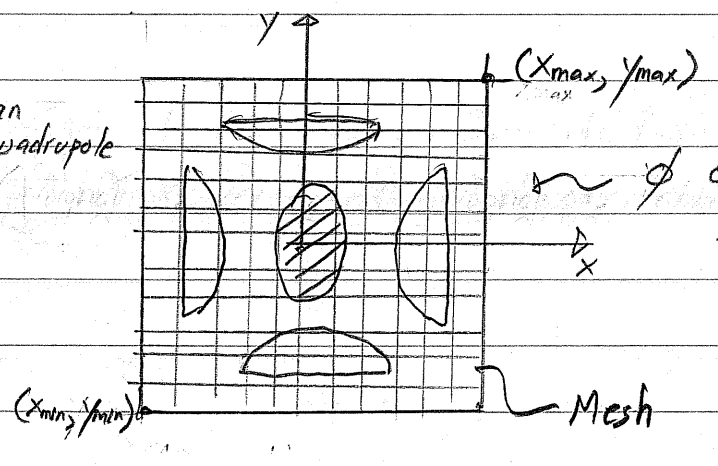
$$x_i = x_{min} + \Delta x \cdot i \qquad \Delta x = (x_{max} - x_{min})/n_x \qquad i = 0, 1, 2, \ldots, n_x$$

$$y_i = y_{min} + \Delta y \cdot j \qquad \Delta y = (y_{max} - y_{min})/n_y \qquad j = 0, 1, 2, \ldots, n_y$$

$$\left.\begin{aligned} \vec{E}_{ij} &= \vec{E}(x_i, y_j) \\ \phi_{ij} &= \phi(x_i, y_j) \\ \rho_{ij} &= \rho(x_i, y_j) \end{aligned}\right\} \text{Field components, potential, and charge are gridded}$$

Ex

Beam in an electric quadrupole (2D)



ϕ or $\frac{\partial \phi}{\partial n}$ specified on domain boundaries and possibly on surfaces within.

Note: ρ_{ij} must be calculated from macro-particle not necessarily on grid points. Also fields will ultimately be needed at macro-particle coordinates not on the grid. These issues will be covered later under "particle weighting"

For low order Differancing, the Poisson Equation becomes:

$$\boxed{\frac{\phi_{i+1,j} - 2\phi_{i,j} + \phi_{i-1,j}}{\Delta_x^2} + \frac{\phi_{i,j+1} - 2\phi_{i,j} + \phi_{i,j-1}}{\Delta_y^2} = -\frac{\rho_{i,j}}{\varepsilon_0}}$$

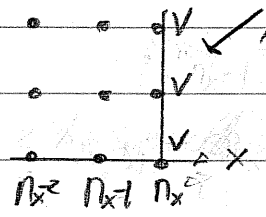
with the gridded field components calculated as:

$$\boxed{\begin{aligned} E_{x_{ij}} &= \frac{\phi_{i+1,j} - \phi_{i-1,j}}{2\Delta_x} \\ E_{y_{ij}} &= \frac{\phi_{i,j+1} - \phi_{i,j-1}}{2\Delta_y} \end{aligned}}$$

Boundary conditions must also be incorporated as equations of constraint.

1) <u>Dirichlet Conditions</u>: ϕ specified on surfaces.

Example: $\phi = V = \text{const}$ at right grid edge:



$\phi_{n_x,j} = V = \text{const}$

$i = n_x - 1$ cells:

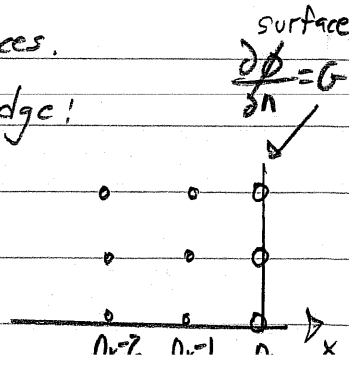
$$\boxed{\frac{V - 2\phi_{n_x-1,j} + \phi_{n_x-2,j}}{\Delta_x^2} + \frac{\phi_{n_x-1,j+1} - 2\phi_{n_x-1,j} + \phi_{n_x-1,j-1}}{\Delta_y^2} = \frac{-\rho_{n_x-1,j}}{\varepsilon_0}}$$

; etc.

2) <u>Neumann Conditions</u>: $\partial\phi/\partial n$ specified on surfaces.

Example: $\partial\phi/\partial x = \sigma = \text{const}$ at right grid edge:

$$\boxed{\frac{\phi_{n_x-1,j} - \phi_{n_x,j}}{\Delta_x} = \sigma}$$



The finite differenced Poisson equation and the boundary conditions can be expressed in Matrix Form:

$$\boxed{\bar{M} \cdot \vec{\phi} = \vec{S}}$$

$\bar{M}$ = coefficients matrix from <u>local</u> finite differences
This matrix will be sparse (i.e., most elements zero)

$\vec{\phi}$ = vector of potentials at grid points.

$\vec{S}$ = "source" terms resulting from 1) <u>beam charge deposited on the grid</u> (ρ_{ij}) and 2) "known" potentials from <u>boundary condition constraints</u>.

Formal solution found by Matrix inversion:

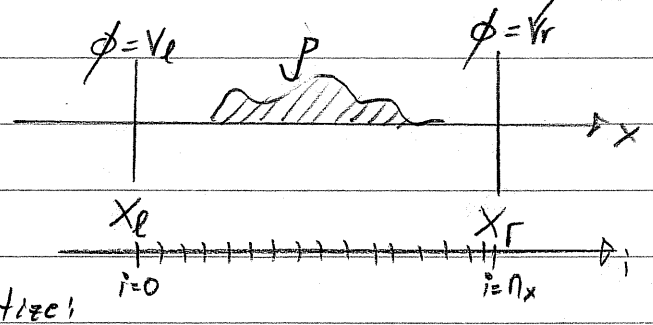
$$\boxed{\vec{\phi} = \bar{M}^{-1} \cdot \vec{S}}$$

Direct inversion of $\bar{M}^{-1}$ is not practical due to large dimension of problem. However:

- $\bar{M}$ will in general be very sparse due to the use of local, low-order finite differencing.
- Many fast, numerically efficient inversion methods exist for sparse matrices.
  - Specific method best used depends on type of differencing and boundary conditions.

To illustrate this procedure, consider a simple 1-d example with Dirichlet boundary conditions.

$$\frac{d^2\phi}{dx^2} = -\frac{\rho}{\varepsilon_0} \qquad \phi(x_\ell) = V_\ell \quad \text{left B.C.}$$
$$\phi(x_r) = V_r \quad \text{right B.C.}$$



Discretize:

$$\frac{\phi_{i+1} - 2\phi_i + \phi_{i-1}}{\Delta_x^2} = -\frac{\rho_i}{\varepsilon_0}$$

$$x_i = x_\ell + \Delta_x \cdot i, \quad \Delta_x = (x_r - x_\ell)/n_x; \quad i = 0, 1, 2, \ldots, n_x$$

$$\phi_0 = V_\ell$$
$$\phi_{n_x} = V_r$$

note: ρ_0 is irrelevant – and will have a surface term to fix boundary cond.

$$\begin{bmatrix} -2 & 1 & & & & & \\ 1 & -2 & 1 & & & 0 & \\ & 1 & -2 & 1 & & & \\ & & & \ddots & & & \\ & & & 1 & -2 & 1 & \\ & 0 & & & 1 & -2 & 1 \\ & & & & & 1 & -2 \end{bmatrix} \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \vdots \\ \phi_{n_x-3} \\ \phi_{n_x-2} \\ \phi_{n_x-1} \end{bmatrix} = -\frac{\Delta_x^2}{\varepsilon_0} \begin{bmatrix} \rho_1 + \frac{\varepsilon_0}{\Delta_x^2} V_\ell \\ \rho_2 \\ \rho_3 \\ \vdots \\ \rho_{n_x-3} \\ \rho_{n_x-2} \\ \rho_{n_x-1} + \frac{\varepsilon_0}{\Delta_x^2} V_r \end{bmatrix}$$

Matrix has tri-diagonal structure and may be rapidly inverted to find the ϕ_i.

- Sparse matrices need not be stored in full (waste of memory)

Many other methods exist to solve the discretized field equations. These methods fall into 3 broad classes:

1) Direct Matrix Methods
   - Fast inversion of sparse matrices.

2) Spectral Methods
   - Fast Fourier Transform (FFT)
     - Periodic boundary conditions
     - sine transform [$\phi=0$ on grid boundary]
     - FFT + capacity matrix for arbitrary conductors
     - Free space boundary conditions

3) Relaxation Methods
   - Successive overrelaxation (SOR)
     - General boundary conditions and structures.
   - Multigrid (good, fast, accurate method for complicated boundaries)

Sometimes these methods may be combined. For example, one might employ spectral methods transversely and invert the tri-diagonal matrix longitudinally.

Other discretization procedures are also widely employed:
- Finite elements
- Variational
- Monte-Carlo ''''''

The various methods of field solution are central to the efficient numerical solution of intense beams. It is not possible to review them all here. But before discussing particle weighting, we will first overview the important spectral methods and FFTs

Spectral Methods and the FFT

The spectral approach combined with numerically efficient Fast Fourier Transforms (FFTs) is commonly used to solve the Maxwell Equations on a discrete spatial grid.

- Approach provides spectral information on fields that can smooth the interactions.

Illustrate in 1-d for simplicity (multidimensions analogous)

$$\boxed{\frac{d^2\phi}{dx^2} = -\frac{\rho}{\varepsilon_0}}$$

Continuous Fourier Transforms: (Reminder)

$$\boxed{\begin{aligned} \tilde{\phi}(k) &= \int_{-\infty}^{\infty} dx\, e^{ikx}\, \phi(x) & \tilde{\rho}(k) &= \int_{-\infty}^{\infty} dx\, e^{ikx}\, \rho(x) \\ \phi(x) &= \int_{-\infty}^{\infty} \frac{dk}{2\pi}\, e^{-ikx}\, \tilde{\phi}(k) & \rho(x) &= \int_{-\infty}^{\infty} \frac{dk}{2\pi}\, e^{-ikx}\, \tilde{\rho}(k) \end{aligned}}$$

Transform Poisson Eqn. $\Rightarrow$ $$k^2 \tilde{\phi}(k) = \frac{\tilde{\rho}(k)}{\varepsilon_0}$$

$$\boxed{\phi(x) = \int_{-\infty}^{\infty} \frac{dk}{2\pi}\, e^{-ikx}\, \frac{\tilde{\rho}(k)}{\varepsilon_0 k^2}}$$

Similar procedures work for the field on a discrete and <u>finite</u> spatial grid.

Discrete Fourier Transform:

$$x_j = x_{min} + \Delta_x \cdot j \quad , \quad dx = (x_{max} - x_{min})/n_x \; ; \; j = 0, 1, 2, \ldots n_x$$

$$\phi_j = \phi(x_j)$$

$n_x + 1$ grid points. (take even)

$$k_n \equiv \frac{2\pi n}{(n_x+1)\Delta_x} \qquad n = -\frac{(n_x+1)}{2}, \ldots\ldots, \frac{n_x+1}{2}$$

$$\tilde{\phi}(k_n) = \int_{-\infty}^{\infty} dx \, e^{ik_n x} \phi(x) \approx \Delta_x \sum_{j=0}^{n_x} e^{ik_n x_j} \phi_j$$

Discrete Transform:

$$\approx \Delta_x \sum_{j=0}^{n_x} e^{\frac{i 2\pi n j}{n_x+1}} \phi_j$$

$$\boxed{\begin{aligned} \tilde{\phi}_n &\equiv \tilde{\phi}(k_n) \\ &\equiv \Delta_x \sum_{j=0}^{n_x} e^{\frac{i 2\pi n j}{n_x+1}} \phi_j \end{aligned}}$$

Note that $\tilde{\phi}_n$ is periodic in n with period $n_x + 1$

$$\tilde{\phi}_{-n} = \tilde{\phi}_{n_x+1-n}$$

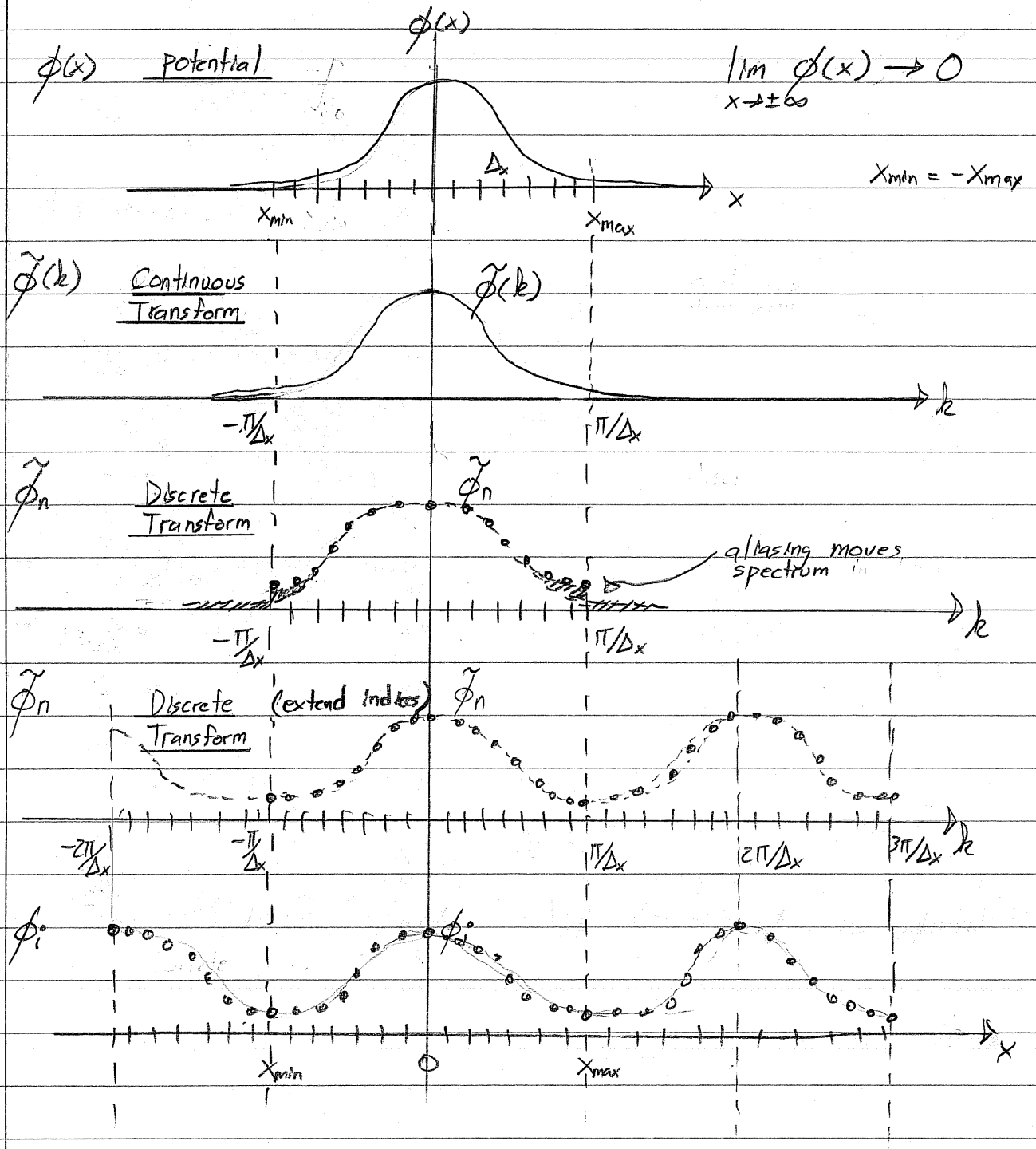
Let n vary from: $n = 0, 1, 2, \ldots, n_x$
so n and j have same ranges.

Then an inverse transform can be <u>exactly</u> constructed:

$$\boxed{\phi_j = \frac{1}{(n_x+1)\Delta_x} \sum_{n=0}^{n_x} e^{\frac{-i 2\pi n j}{n_x+1}} \tilde{\phi}_n}$$

The discrete transform constructed describes a periodic problem if indices are extended.

- Discretization errors (aliasing) can occur.



Application of Discrete Fourier transforms to solve Poisson's Equation:

$$E_x = -\frac{\partial \phi}{\partial x} \quad \Longleftrightarrow \quad E_{x_j} = -\frac{\phi_{j+1} - \phi_{j-1}}{2\Delta x}$$

$$\frac{d^2\phi}{dx^2} = \frac{-\rho}{\varepsilon_0} \quad \Longrightarrow \quad \frac{\phi_{j+1} - 2\phi_j + \phi_{j-1}}{\Delta x^2} = \frac{-\rho_j}{\varepsilon_0}$$

Applying the discrete transform one obtains:

$$\boxed{\begin{aligned} \tilde{E}_{x_n} &= -i K_n \tilde{\phi}_n \qquad K_n = k_n \left[\frac{\sin(k_n \Delta x)}{k_n \Delta x}\right] \\ & \qquad K_n = k_n \, \mathrm{dif}(k_n \Delta x) \\ & \qquad k_n \equiv \frac{2\pi n}{(n_x+1)\Delta x} \end{aligned}}$$

$$\mathrm{dif}(x) \equiv \frac{\sin x}{x} \quad \text{from diffraction theory}$$

and Poisson's eqn becomes

$$\boxed{\begin{aligned} \tilde{\phi}_n &= \frac{\tilde{\rho}_n}{\varepsilon_0 K_n^2} \qquad K_n^2 = k_n^2 \left[\frac{\sin\left(\frac{k_n \Delta x}{2}\right)}{\frac{k_n \Delta x}{2}}\right]^2 \\ & \qquad = k_n^2 \, \mathrm{dif}^2(k_n \Delta x/2) \end{aligned}}$$

Note:

- Factors of K_n^2 need only be calculated once per simulation (store values).

// Aside – Example Derivation of Discrete Transform Formula.

$$E_{x_j} = -\frac{\phi_{j+1} - \phi_{j-1}}{2\Delta_x}$$

$$\begin{cases} \phi_j = \dfrac{1}{(n_x+1)\Delta_x} \displaystyle\sum_{n=0}^{n_x} e^{\frac{-i2\pi n j}{n_x+1}} \tilde{\phi}_n \\ E_{x_j} = \dfrac{1}{(n_x+1)\Delta_x} \displaystyle\sum_{n=0}^{n_x} e^{\frac{-i2\pi n j}{n_x+1}} \tilde{E}_{x_n} \end{cases}$$

Substitute transforms in difference formula:

$$2\Delta_x \sum_{n=0}^{n_x} e^{\frac{-i2\pi n j}{n_x+1}} \tilde{E}_n =$$

$$= -\sum_{n=0}^{n_x} e^{\frac{-i2\pi n j}{n_x+1}} \tilde{\phi}_n \left\{ e^{\frac{-i2\pi n}{n_x+1}} - e^{\frac{i2\pi n}{n_x+1}} \right\}$$

$$\Delta_x \sum_{n=0}^{n_x} e^{\frac{-i2\pi n j}{n_x+1}} \tilde{E}_n = i \sum_{n=0}^{n_x} e^{\frac{-i2\pi n j}{n_x+1}} \sin\left(\frac{2\pi n}{n_x+1}\right) \tilde{\phi}_n$$

This equation must hold true for each complex term $\propto e^{\frac{-i2\pi n j}{n_x+1}}$ to be valid for general j.

$$\Rightarrow \quad \tilde{E}_n = \frac{i}{\Delta_x} \sin\left(\frac{2\pi n}{n_x+1}\right) \cdot \tilde{\phi}_n$$

$$k_n = \frac{2\pi n}{(n_x+1)\Delta_x}$$

$$\Rightarrow \quad \tilde{E}_n = i k_n \left[\frac{\sin(k_n \Delta_x)}{k_n \Delta_x} \right] \tilde{\phi}_n$$

//

Typical discrete Fourier Transform Field Solution (not optimized)

$$\rho_j \xrightarrow[\text{FT}]{\text{Foward Trans.}} \tilde{\rho}_n \xrightarrow[\cdot \frac{1}{k_n^2}]{\text{Multiply}} \tilde{\phi}_n \xrightarrow[\text{IFT}]{\text{Inverse Trans.}} \phi_j \xrightarrow{\text{Finite Diff.}} E_{xj}$$

The Fast Fourier Transform (FFT) makes this procedure numerically efficient.

- Discrete transform (no optimization) $\sim (n_x+1)^2$ complex operations.

- FFT exploits symmetries to reduce needed operations to $\sim (n_x+1)\ln(n_x+1)$,
  - Huge savings for large n_x.

- The needed symmetries exist only for certain numbers of grid points. In the simplest manifestations:

$$n_x + 1 = 2^p \qquad p = 1, 2, 3, \ldots$$

  - reduced gridding freedom.
  - other manifestations allow 2^p and products of prime numbers for more possibilities.

The FFT can be combined with other procedures such as capacity matrices to implement boundary conditions for interior conductors, etc. This allows rapid field solutions in complicated geometries.

FFT is fastest method for simple geometry.

Particle Weighting

We have outlined methods to solve Electrostatic! Maxwell's equations on a discrete spatial grid. To complete the description we must:

- Specify how to deposit macro-particle charges and currents on the mesh.
- Specify how to interpolate fields on the spatial grid points to the macroparticle coordinates (not generally on the mesh) to apply in the particle advance.
- Smooth interactions resulting from the small number of macro-particles to reduce artifical collisions resulting from the use of an unphysically small number of macro-particles.

This is called the "particle weighting" problem.

Birdsall & Langdon pg 10

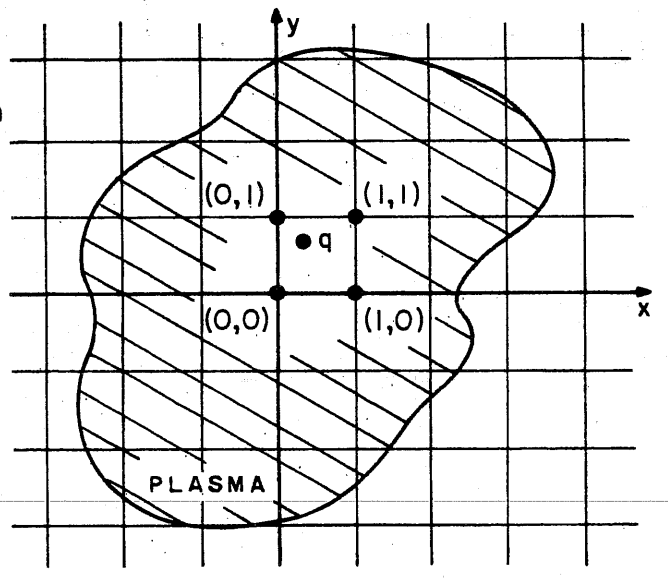


Figure 2-2b A mathematical grid is set into the plasma region in order to measure charge and current densities ρ, **J**; from these we will obtain the electric and magnetic fields **E**, **B** on the grid. A charged particle q at (x,y) will typically be counted in terms of ρ at the nearby grid points (0,0), (1,0), (1,1), (0,1) and in terms of **J** at the faces between these points. The force on q will also be obtained from the fields at these nearby points.

It is found that it is usually better to employ the same weighting schemes to both deposit the Macro-particles charges and currents on the mesh and to extrapolate the fields at gridded points to the Macro-particles.

- Avoids unphysical self-forces where the particle accelerates itself.

Many methods of particle weighting exist. Can be grouped into 4 catagories:

1) <u>Nearest Grid Point</u>
2) <u>Cloud in Cell</u> (CIC)
   - Shaped particles. Linear ⇒ Particle-in-Cell PIC
3) <u>Multipole</u>
   - Dipole, subtracted dipole, etc.
4) <u>Higher order Methods</u>
   - Splines
   - $\vec{k}$ space cutoffs in discrete transforms

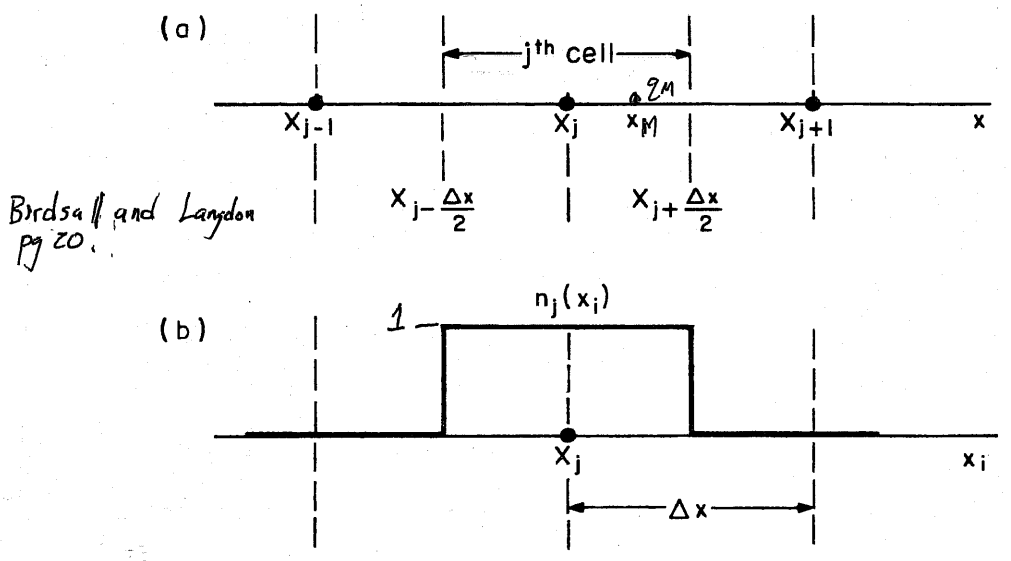
   ⋮

Possible hybrid methods also exist. We will illustrate methods 1) and 2) in 1D for electrostatic problems. Generalizations to higher dimensions are straightforward. Descriptions of methods can be found in the literature.

1) Nearest Grid Point:

Assign charges to the nearest grid cell.

- Fast and simple
- Noisy



Birdsall and Langdon pg 20.

Figure 2-6a (a) Zero-order particle and field weighting, also called nearest-grid-point, or, NGP. Particles in the jth cell, that is, with positions $x_i \in X_j \pm \Delta x/2$, are assigned to X_j to obtain grid density $n(X_j)$. All of these particles are acted on by the field at X_j, $E(X_j)$. (b) The density $n_j(X_i)$ at point X_j due to a particle at x_M as the particle moves through the cell centered on X_j. This density may be interpreted as the effective particle shape.

q_M, X_M = charge and coordinate of Macro-particle

X_j = closest grid cell.

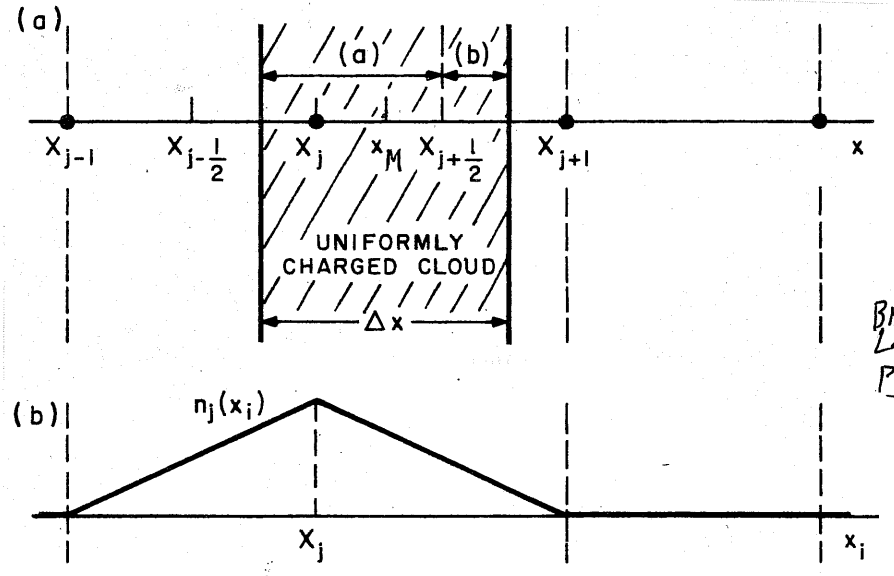
charge Deposition :	$q_j = q_M$
Field :	$E_x\big\|_{x=X_M} = E_{x_j}$

2) Cloud in Cell

Shaped macroparticles pass freely through each-other.

- Smoother than Nearest Grid Point.
- For linear interpolation results in simple, commonly used "Particle-in-Cell" (PIC) method.

Illustrate 1D PIC "Area" weighting:



Birdsall & Langdon pg 21

Figure 2-6b (a) First-order particle weighting, or cloud-in-cell model CIC. The nominal finite-size charged particle, or cloud, is one cell wide, with center at x_i. This weighting puts that part of the cloud which is in the jth cell at X_j, fraction (a), and that part which is in the $(j+1)^{th}$ cell at X_{j+1}, fraction (b). This weighting is the same as applying NGP interpolation to each elemental part. (b) The grid density $n_j(x_i)$ at point x_M as the particle moves past X_j, again displaying the effective particle shape $S(x)$.

q_M, X_M = charge and coordinate of macro-particle

X_j = closest grid cell

Charge Deposition

$$q_j = q_M \left[\frac{\Delta_x - (X_M - X_j)}{\Delta_x} \right] = \frac{X_{j+1} - X_M}{\Delta_x}$$

$$q_{j+1} = q_M \left[\frac{X_M - X_j}{\Delta_x} \right]$$

Field:

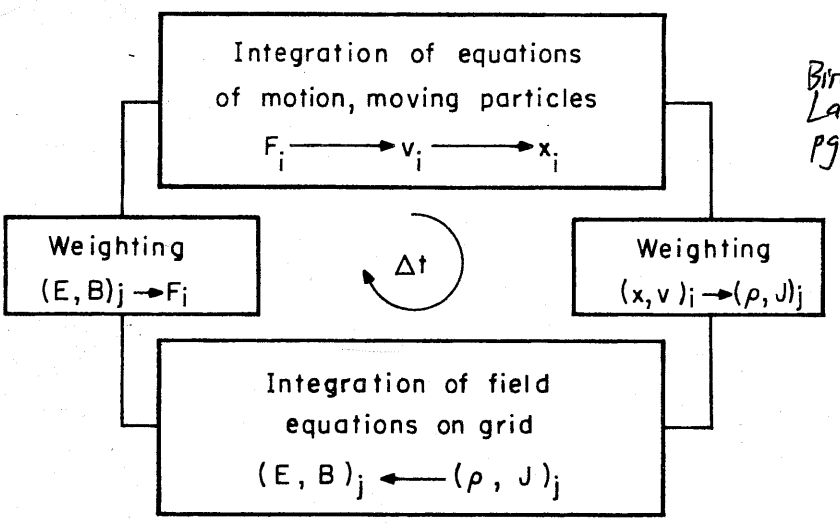
$$E_x \Big|_{x=X_M} = \left[\frac{X_{j+1} - X_M}{\Delta_x} \right] E_j + \left[\frac{X_M - X_j}{\Delta_x} \right] E_{j+1}$$

Computational Cycle for Particle Simulations

We now have (simplified) notions of the parts that make up a particle simulation.

1) Particle Moving
2) Field Solver on a discrete mesh
3) Weighting of particles and Fields to and from the mesh.

These are combined in a computational cycle:



Birdsall & Langdon pg 11

Figure 2-3a A typical cycle, one time step, in a particle simulation program. The particles are numbered $i = 1, 2, \ldots, NP$; the grid indices are j, which become vectors in 2 and 3 dimensions.

At synchronization steps, particle coordinates, velocities, and fields will all be in-phase and diagnostics may be accumulated.

Diagnostics:

Depend on what is analyzed. Typical choices:

- **Moments** – statistical sums over the particle distribution (slices and full beam), often plotted as time histories as the beam evolves in the accelerator:

centroid: $x_c = \langle x \rangle_\perp$ .....

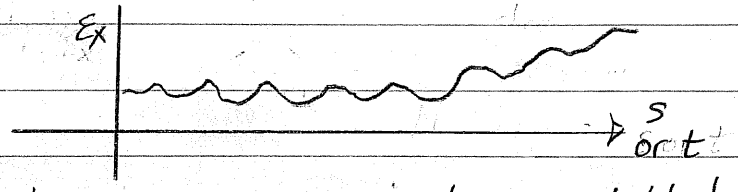
RMS Widths: Beam Size $r_x = 2\langle (x-x_c)^2 \rangle^{1/2}$, ... etc.

Emittance: $\varepsilon_x = 16\left[\langle (x-x_c)^2\rangle\langle (x'-x_c')^2\rangle - \langle (x-x_c)(x'-x_c')\rangle^2\right]^{1/2}$,

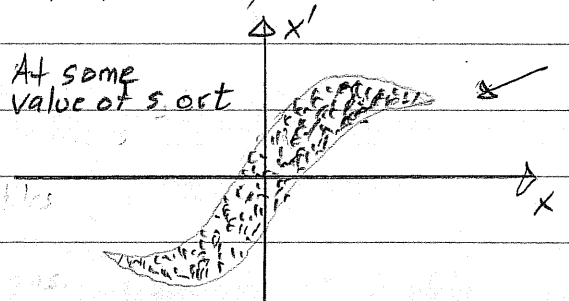
etc.

Here; $\langle \cdots \rangle_\perp = \frac{1}{N_s}\sum_i^{\text{slice}} \{\cdots\}_i$ ..... etc.

Slice emittance "history"

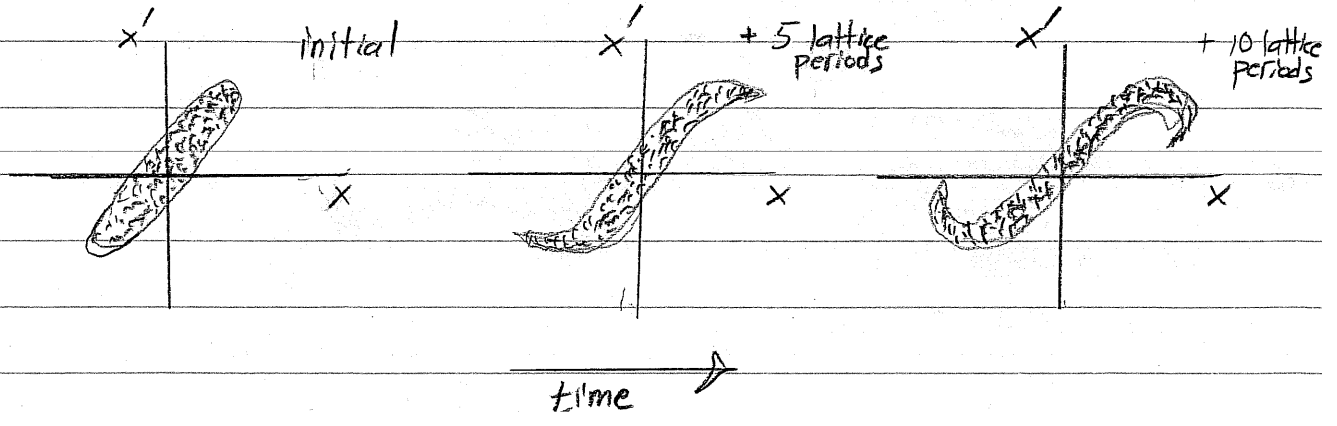


- **Particle phase-space projections** plotted as snapshots in time (x-x', y-y', x-y, x'-y', ...).



scatter plot of enough macro-particles to visualize the phase-volume and structure of the beam.

Plotting snapshots at periodic locations allows vizualization of the evolution of beam distortions:

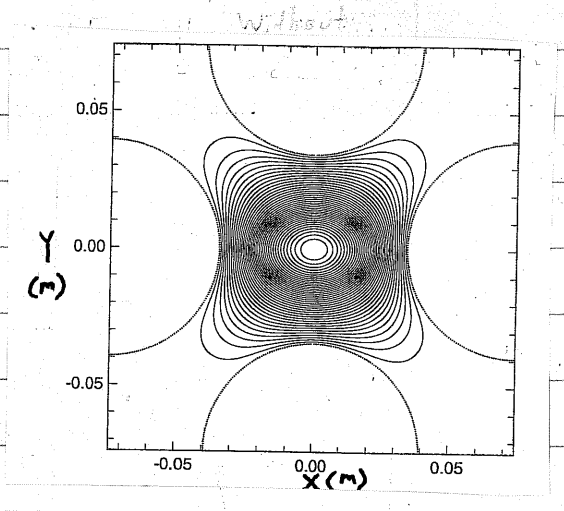


- <u>Field</u> Diagnostics:

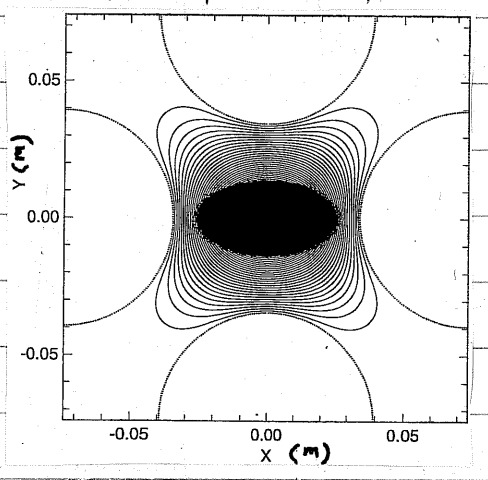
Contours of self and applied fields illustrate field structure.

Examples:

1) Total potential contors of ϕ (self and applied) of an elliptical beam in an electric quadrupole formed by 4 biased rods.

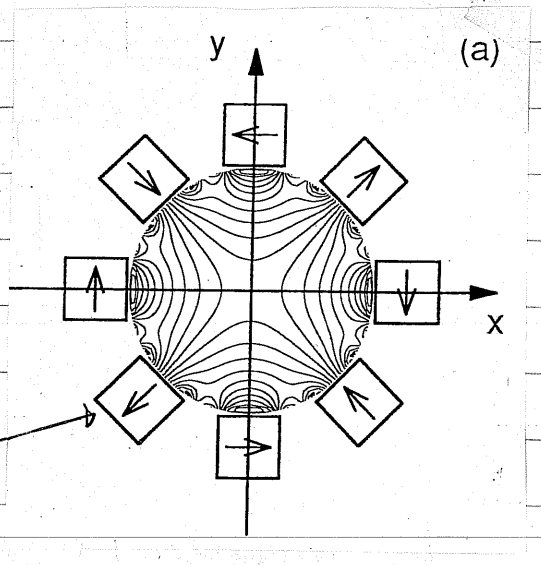


same ϕ contours with beam particles plotted



2) <u>Applied Field</u> Scalar magnetic potential contours of a permanant magnet lense illustrate the structure of applied field nonlinearities:

8 Block Permanent Magnet Quadrupole.



Note: contours deviate from hyperbolic near apeture edge.

Numerous other field diagnostics are possible:

- Field energy, multipole moments,

Particle Loads

To start the simulation, one must specify and "load" the initial distribution function.

In realistic accelerators, focusing elements are s-varying. In such situations there are in general no known equilibrium distribution functions. Moreover, it is unclear in most cases if the beam is even best thought of as an equilibrium + perturbations as is typical in plasma physics. Rather, in accelerators, the beam is injected from a source and may only reside in the machine (especially a linac) for a small number of characteristic oscillation periods and may not fully relax to an equilibrium-like state. In such situations, so-called "source-to-target" simulations where the particles are simulated off the source and tracked to the target can be most realistic if carried out with realistic focusing fields, accelerating waveforms, alignment errors, etc.

Unfortunately, such idealized source-to-target simulations can rarely be carried out due to computer limitations.

- Memory limits
- Numerical convergence and accuracy ....

Two ways around this limitation:

1) Load experimentally measured distribution at some point in the machine and advance as an initial condition.
2) Load an idealized initial distribution.

The 1st option can have practical difficulties:

- Diagnostics often are far from an ideal 6D "snapshot" of the beam phase-space.
  - Much information typically lost.
- Process of measuring the beam can itself change the beam.

Most commonly, some experimental measures such as:

- rms beam sizes r_x, r_y, r_x', r_y'
- rms emittances $\varepsilon_x, \varepsilon_y$

are loaded in the form of idealized distributions.

It can be insightful to initialize the beam in a simplified manner

- Fewer simultaneous processes can allow one to more clearly see how limits arise.
- Seed perturbations of relevance when analyzing resonance effects, instabilities, halo, etc.

In these situations it is often useful to load a round, continuously focused distribution such as:

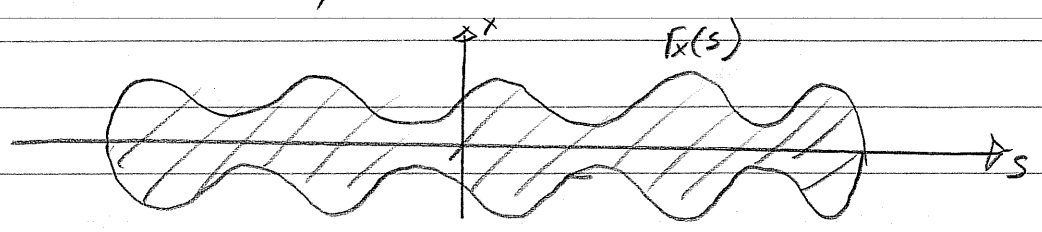
- Thermal Equilibrium
- Waterbag
- KV

and then transform the particle coordinates to "match" the local focusing structure of the lattice using a local KV envelope solution:

$$\frac{d^2}{ds^2} r_x(s) + \kappa_x(s) r_x(s) - \frac{2Q(s)}{r_x(s)+r_y(s)} - \frac{\varepsilon_x^2(s)}{r_x^3(s)} = 0$$

$$\frac{d^2}{ds^2} r_y(s) + \kappa_y(s) r_y(s) - \frac{2Q(s)}{r_x(s)+r_y(s)} - \frac{\varepsilon_y^2(s)}{r_y^3(s)} = 0$$

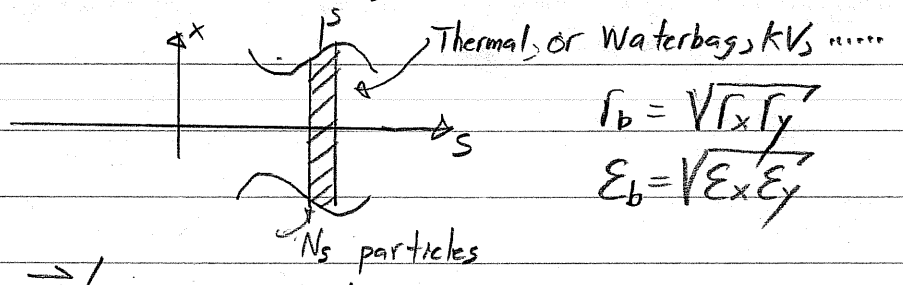
$\kappa_x(s), \kappa_y(s)$ = lattice focusing constants
$Q(s)$ = local perveance (specified)
$\varepsilon_x(s), \varepsilon_y(s)$ = local emittances (specified)



Procedure:

1st

Load in each $\perp$ slice round, continuous distribution:



$$r_b = \sqrt{r_x r_y}$$

$$\varepsilon_b = \sqrt{\varepsilon_x \varepsilon_y}$$

$\Rightarrow$ $\vec{X}_i$, $\vec{X}_i'$ specified

2nd

Transform the spatial coordinates to match the elliptical envelope structure

$$x_i \longrightarrow \frac{r_x}{r_b} x_i$$

$$y_i \longrightarrow \frac{r_y}{r_b} y_i$$

3rd

Transform the local thermal velocity spreads to obtain the right average thermal force.

$$x_i' \longrightarrow \frac{\varepsilon_x}{\varepsilon_b} \frac{r_b}{r_x} x_i'$$

$$y_i' \longrightarrow \frac{\varepsilon_y}{\varepsilon_b} \frac{r_b}{r_y} y_i'$$

4th

Add the correct coherent velocity to match the needed envelope angle.

$$x_i' \longrightarrow x_i' + r_x' \frac{x_i}{r_x}$$

$$y_i' \longrightarrow y_i' + r_y' \frac{y_i}{r_y}$$

// Aside – The semi-Gaussian Distribution

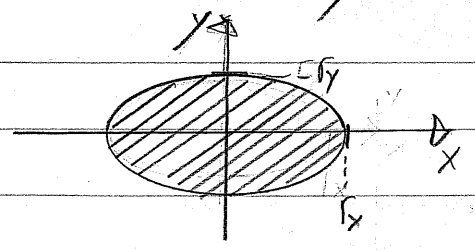
It is not necessary to always load something based on a transformation of an equilibrium distribution to get a good quiescent load. Note that for high space-charge intensities:

- Beam space charge will be more or less uniform out to the edge, where the density will rapidly fall to zero.

- If the beam is injected off a uniform temperture source or has relaxed, one expects roughly uniform thermal velocity spread across the cross-section of the beam.

This suggests the so-called "semi-Gaussian" load specified as follows:

- Uniform density within an elliptical beam envelope.



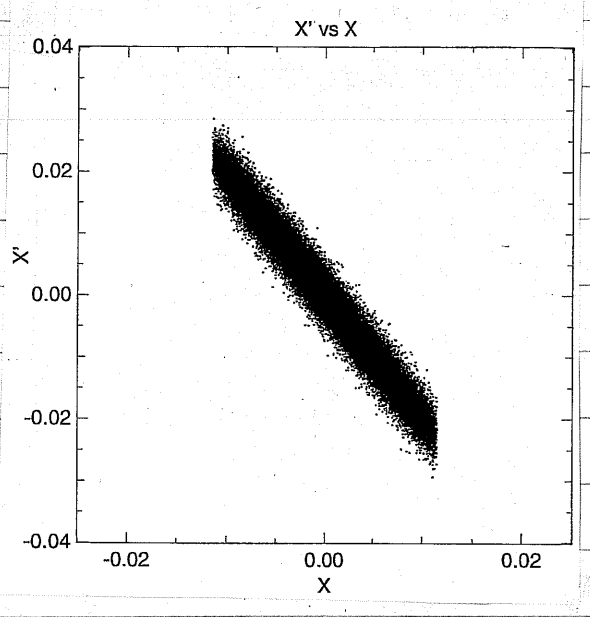
x_i, y_i uniformly distributed for $(x/r_x)^2 + (y/r_y)^2 \leq 1$.

- Gaussian distributed thermal velocity spread with the correct coherent velocity to match the needed envelope angles

$$x_i' = r_x' \frac{x_i}{r_x} + \frac{\varepsilon_x}{2r_x} \tilde{r}_{x_i}$$
$$y_i' = r_y' \frac{y_i}{r_y} + \frac{\varepsilon_y}{2r_y} \tilde{r}_{y_i}$$

$\tilde{r}_{x,y_i}$ Gaussian distributed with unit variance $\left(\frac{1}{N_s} \sum_i^{\text{slice}} \tilde{r}_{x,y_i}^2 = 1 \right)$

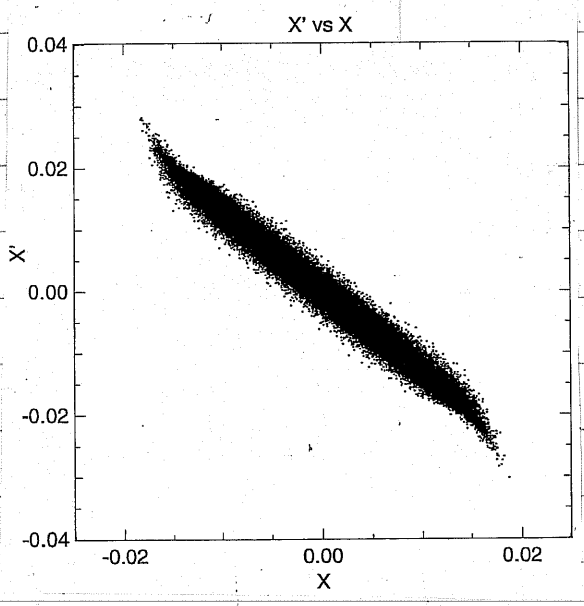
The semi-Gaussian load results in "squared" initial x-x' and y-y' phase space projections:



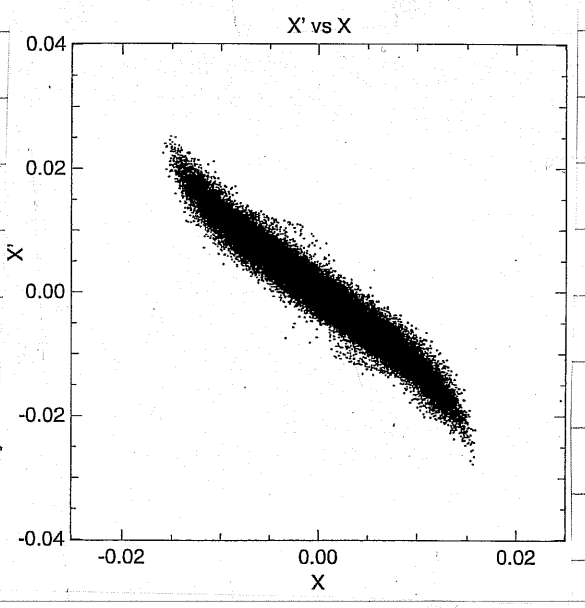
Initial load in a symmetric FODO AG Quadrupole transport lattice

The unphysical edges typically relax rapidly without perturbing the low order RMS structure of the beam (envelope match, emittance, etc.).

1 lattice period advance



4 lattice period advance



Numerous other loading techniques exist to address specific issues. //

Numerical Convergence

Numerical simulations must be checked for proper resolution and statistics to be confident that answers obtained are correct and physical.

Resolution on discritized quantities
- Time step in particle advance
- Spatial grid for fields

Statistics (number of macroparticles) to control noise.

More resolution and statistics require more computer time and memory. To be practical, one generally wishes to solve problems with the minimum resources required to achieve correct, converged answers. Unfortunately, there are no set rules on what resolution and statistics are required. This depends on what one is examining, how long the simulations are run, what numerical methods are employed, ...

Some general guidance:

- The statistical rms emittances:

$$\varepsilon_x = 4\left[\langle x^2\rangle_\perp\langle x'^2\rangle_\perp - \langle xx'\rangle_\perp^2\right]^{1/2}$$

$$\varepsilon_y = 4\left[\langle y^2\rangle_\perp\langle y'^2\rangle_\perp - \langle yy'\rangle_\perp\right]^{1/2}$$

often prove to be sensitive and easy to interpret measures of numerical differences when plotted as overlaid time histories.
  - picks up small phase space distortions induced by numerical errors.

general guidance continued ....

- To get started, find results from similar problems using similar methods.

- Benchmark code and methods against problems with known analytical solutions, established codes using both similar and different numerical methods, ...

- Recheck numerical convergence whenever runs differ significantly <u>or</u> when differing quantities are analyzed.

  - What is adequate for one measure of the beam (say image charge structure) may not be for another (say collective modes).

- Although it is common to increase resolution and statistics till relevant quantities do not vary, it is <u>also useful to purposefully analyze poor resolution and statistics regimes so the characteristics of unphysical numerical errors can be recognized.</u>

- Expect to make many setup, convergence, and debugging runs for each useful series of simulations carried out.

Specific Numerical Convergence Comments:

Time Resolution.

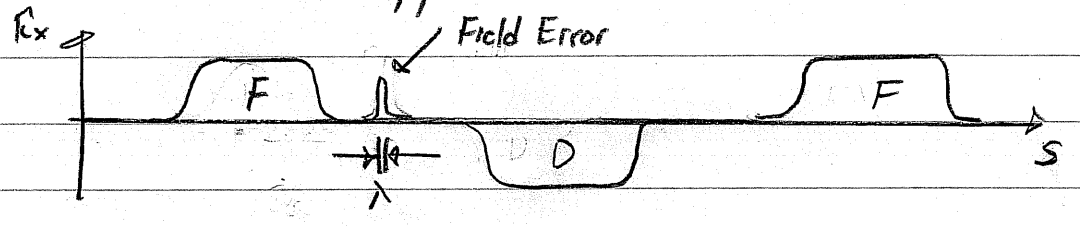
These comments are applicable to both spatial or time steps of the particle advance. We will frame estimates in terms of timesteps.

- Particle coordinates should not move through more than one cell in a single step. This is a standard "Courant" condition:

$$\boxed{\begin{aligned} v_x \Delta_t &< \Delta_x \\ v_y \Delta_t &< \Delta_y \\ v_z \Delta_t &< \Delta_z \end{aligned}}$$

for all particles.

- Enough steps should be taken to resolve variations in applied field structures.



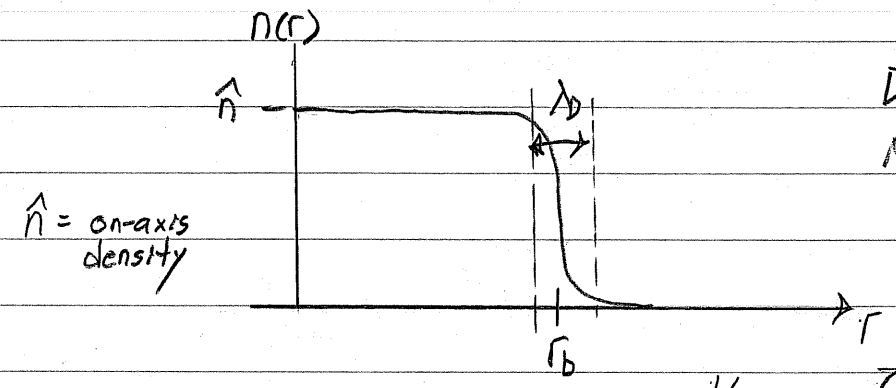
$$\boxed{v_s \Delta_t < \lambda \quad ; \quad \lambda = \text{shortest wavelength of field structures to be modeled}}$$

- Phase variations in collective waves (if of interest) should be resolved. For a leap-frog mover this requires:

$$\boxed{\begin{aligned} &\omega = \text{mode frequency component} \\ &\frac{\omega \Delta_t}{2\pi} < \frac{1}{2} \end{aligned}}$$

Spatial Resolution

For cold beams the beam edge can be sharp for most reasonable distribution functions:



Density falls off in $\mathcal{O}(\lambda_D)$

Debye length: $\lambda_D = \dfrac{v_t}{\omega_p} = \left(\dfrac{\varepsilon_0 k_B T}{\hat{n} q^2}\right)^{1/2}$ $\qquad v_t = (k_B T/m)^{1/2}$, $\quad \omega_p = \left(\dfrac{q^2 \hat{n}}{\varepsilon_0 m}\right)^{1/2}$

$$r_b = \sim 5\lambda_D \rightarrow 30\lambda_D \quad \text{typical.}$$

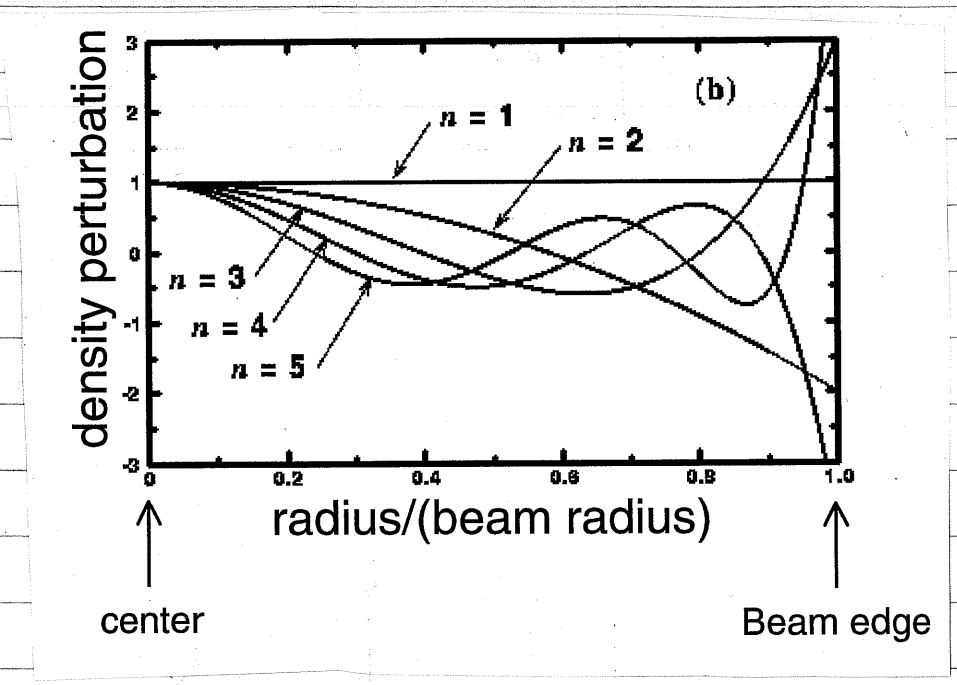
To resolve edge physics, the mesh should have several cells across the rapid density variation near the edge of the beam.

$$\boxed{dx, dy < \frac{\lambda_{Dx,y}}{2} \qquad \lambda_{Dx,y} = \left(\frac{k_B T_{x,y}}{m}\right)^{1/2} \qquad T_{x,y} = \text{local kinetic temperature}}$$

The mesh should also resolve relevant spatial scales associated with processes of interest:

- If applied electrostatic fields are calculated from biased conductors, the mesh should resolve conductor structures or special corrections should be made.
- Self-field fluctuations induced by collective modes should be resolved.

Collective mode resolution: From Eigenfunctions to be presented later in the class:



Collective mode density perturbation on a uniform density beam.

$\Rightarrow$ $dx, dy < \lambda$ ~ shortest characteristic wavelength of modes of interest.

Note that higher order modes (n larger) will become hard to resolve. Moreover, such perturbations also oscillate rapidly making time (s) stepsize resolution likewise difficult.

Statistics:

Collective effects typically require having a significant number of particles N_D within the characteristic screening radius characterized by the Debye length:

2D: $$N_D = \sum_i \int_{\substack{\text{circle} \\ |\vec{x}|<\lambda_D}} d^2x \, \delta^{(2)}(\vec{x}-\vec{x}_i) \gg 1$$

$\vec{x}_i$ = macro-particle coordinate.

3D: $$N_D = \sum_i \int_{\substack{\text{sphere} \\ |\vec{x}|<\lambda_D}} d^3x \, \delta^{(3)}(\vec{x}-\vec{x}_i) \gg 1$$

where:

$$\lambda_D = \frac{v_t}{\omega_p} = \left(\frac{\varepsilon_0 k_B T}{\hat{n} q^2}\right)^{1/2} \qquad v_t = (k_B T/m)^{1/2} \qquad \omega_p = \left(\frac{q^2 \hat{n}}{\varepsilon_0 m}\right)^{1/2}$$

$\sum_i \Rightarrow$ sum over all macro particles.

In simulations of higher order collective modes it may also be necessary to have a significant number of particles per cell on a mesh that resolves the relevant spatial variations of mode induced self-field fluctuations.

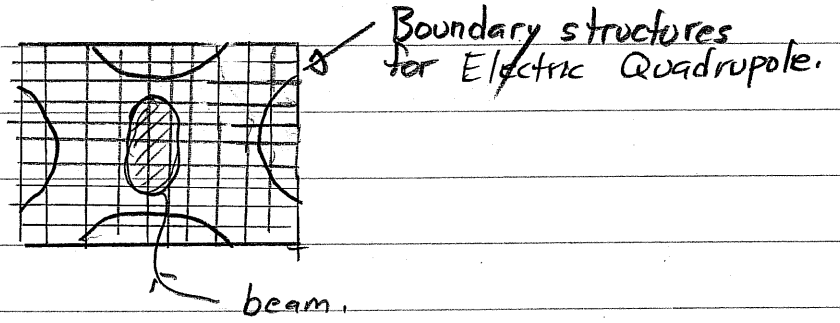
2D: $$N_{cell} = \sum_i \int_{cell} d^2x \, \delta^{(2)}(\vec{x}-\vec{x}_i) \gg 1$$

3D: $$N_{cell} = \sum_i \int_{cell} d^3x \, \delta^{(3)}(\vec{x}-\vec{x}_i) \gg 1$$

- Larger N_{cell} prevents local self-fields from being noise dominated.
- Larger N_{cell} leads to larger N_D, typically $N_D > N_{cell}$ since λ_D must be resolved on the grid.

Good statistics are <u>only</u> needed in the beam core with the possible exception of certain beam-halo problems and near the beam edge.

- Most beams will only occupy a fraction of the full grid.



statistics should be evaluated in the cells that the beam occupies rather than average grid measures.

No comprehensive rules exist for how good the statistics must be. <u>Individual problems must be checked and verified.</u> Some general comments:

- What is adequate will typically depend on what is analyzed
  - Image fields may be resolved with few particles
  - Collective waves may take many particles if low noise (interpetable) diagnostic projections are needed.
- Longer runs generally require increased statistics
- Poor statistics result in unphysical collisionality that is often characterized by a linear rise in beam emittances with simulation time.

Classes of Particle Simulations.

How important is smoothing?

3D Beam: $N \sim 10^{10} - 10^{14}$ particles typical

Simulations: $N \lesssim 10^{8}$ practical (modern parallel computers), typical $10^{3} - 10^{6}$

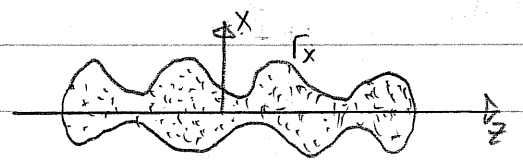
Each simulation particle may represent: $10^{3} \rightarrow 10^{11}$ particles in the real beam for 3D simulations.

- Smoothing involved with particle weightings are key to obtaining physical answers and limiting collisionality.

Is the situation really this bad?

- Lower dimensonal models typically simulated.

3D Model



N point particles with smoothed interactions

Phase Space:

$$\left.\begin{array}{l} x, y, z \\ p_x, p_y, p_z \end{array}\right\} \text{6D}$$

Physical charge – point charges

$$\rho = \sum_i q\, \delta(x-x_i)\,\delta(y-y_i)\,\delta(z-z_i)$$

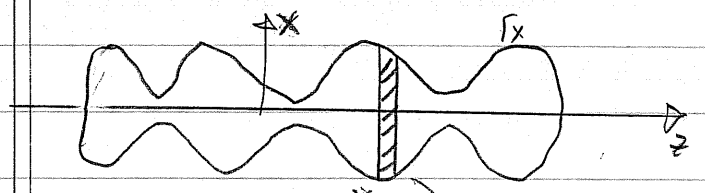
Smoothed charge

$$\rho = \sum_i q_M\, f(x-x_i, y-y_i, z-z_i)$$

q_M: Macro particle charge

f: smoothed shape function

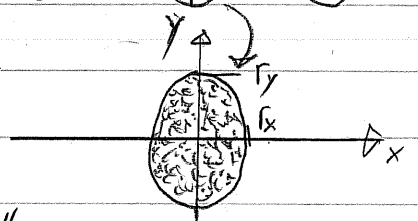
2D ⊥ Thin Slice Model



N line charges with smoothed interactions.

Thin slice

$\frac{\partial}{\partial z} \ll \nabla_\perp$



Phase Space:

x, y } 4D + possible 1D
p_x, p_y } p_z

4D or 5D

"Physical" charge - line charges

$$\rho = \sum_i \lambda_p \delta(x-x_i)\delta(y-y_i)$$

Smoothed charge -

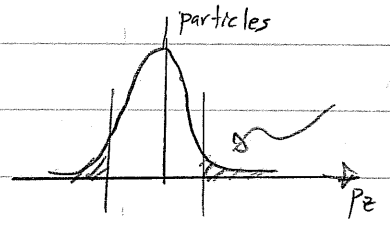
$$\rho = \sum_i \lambda_M f(x-x_i, y-y_i)$$

λ_M: Macro particle line charge

f: smoothing function

The slice must be tracked in s with each particle moving the same increment in s with each step so that a slice maps to a slice.

- If p_z is included the velocity distribution must be assumed "frozen in".

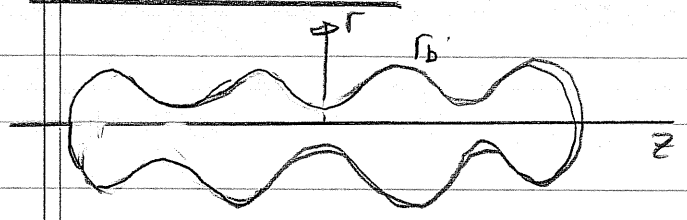


parts that would leave assumed replenished by particles from adjacent slices.

- Response to acceleration may be modeled with

- "Thick" slice models also possible with periodic boundary conditions on the "slice" to try to recover some 3D effects of a long pulse in a periodic lattice.

2D r-z Model



N charged Rings with smoothed interactions

$\partial/\partial\theta = 0$ Axisymmetric

Phase Space.

$\left.\begin{matrix} r, z \\ p_r, p_z \end{matrix}\right\}$ 4D + possible 1D p_θ (angular mom.)

4D or 5D.

physical charge – cylindrical rings

$$\rho = \sum_i \frac{Q_i}{2\pi} \frac{\delta(r-r_i)\,\delta(z-z_i)}{r_i}$$

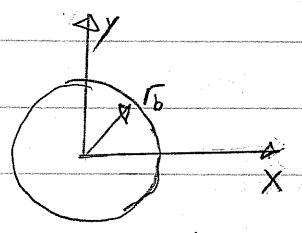
smoothed charge

$$\rho = \sum_i \frac{Q_M}{2\pi} \frac{f(r-r_i,\, z-z_i)}{r_i}$$

Q_M : Macro particle charge
f : Smoothing function.

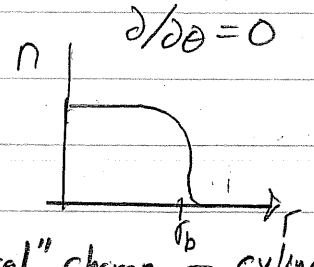
- Used to model solenoidal transport of an initial axisymmetric beam.
- Sometimes used to model AG beams with an approximately equivalent, s-dependant focusing force. $\vec{x}_\perp'' = k_{\beta 0}^2(s)\,\vec{x}_\perp + \cdots\cdots$

1D Axisymmetric Model



N charged cylinders with smoothed interactions

$\partial/\partial\theta = 0$



Phase – Space

$\left.\begin{matrix} r \\ p_r \end{matrix}\right\}$ 2D + possible p_θ, p_z

2D to 4D

"physical" charge – cylindrical sheets

$$\rho = \sum_i \frac{Q_i}{2\pi} \frac{\delta(r-r_i)}{r_i}$$

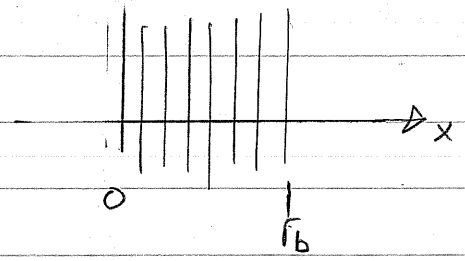
smoothed charge

$$\rho = \sum_i \frac{Q_M}{2\pi} \frac{f(r-r_i)}{r_i}$$

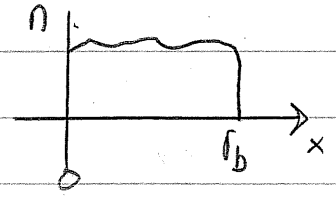
Q_M : Macro charge
f : Smoothing function

- Simple model for continuously focused, axisymmetric beams

1D slab Model



N charged slabs with smoothed interactions.



Phase-Space

$\left.\begin{matrix} x \\ p_x \end{matrix}\right\}$ 2D + possible p_y, p_z

2D to 4D

"physical" charge - sheets

$$\rho = \sum_i \sigma\, \delta(x - x_i)$$

smoothed charge

$$\rho = \sum_i \sigma_M\, f(x - x_i)$$

σ_M: macro particle charge
f: smoothing function

- Most simple model, but slab geometry is least physical.

It is not immediately clear how such different models can in many cases represent qualitatively similar collective interactions since force laws can change form with dimension. For example, in free space, we find that:

Free Space.

Model	Field due to ith "particle"
3D	$\vec{E} = \dfrac{q_i}{4\pi\varepsilon_0} \dfrac{(\vec{x} - \vec{x}_i)}{\lvert\vec{x} - \vec{x}_i\rvert^3}$
2D	$\vec{E} = \dfrac{\lambda_i}{2\pi\varepsilon_0} \dfrac{(\vec{x} - \vec{x}_i)}{\lvert\vec{x} - \vec{x}_i\rvert^2}$ $\quad \lambda_i$ = line-charge "particle"
1D	$E_x = \dfrac{\sigma_i}{2\varepsilon_0} \dfrac{(x - x_i)}{\lvert x - x_i\rvert}$ $\quad \sigma_i$ = sheet charge "particle"

The reason these radically different interactions can give similar physics is that the screening associated with collective interactions is found to be similar:

- Debye screening has similar characteristics in each dimension.

showed 2D form, in class. Will show in final that the 3D scaling obtains the same Debye length definition. $\Rightarrow$

$$\lambda_D = \frac{v_{thermal}}{\omega_p} = \left(\frac{\varepsilon_0 k_B T}{\hat{n} q^2}\right)^{1/2} \qquad v_t = (k_B T/m)^{1/2} \qquad \omega_p = \left(\frac{q^2 \hat{n}}{\varepsilon_0 m}\right)^{1/2}$$

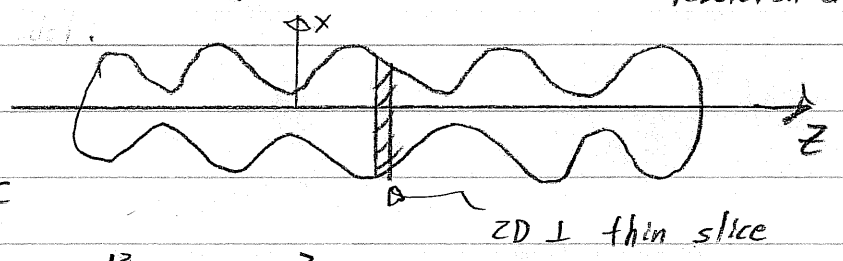
- <u>It is much easier to have a significant number of particles within the characteristic screening distances for lower dimensional problems.</u>

  - Lower dimensional simulations can more easily resolve collective effects!
    (Sometimes people run 3D simulations for collective modes and present garbage answers due to resolution difficulties)

// Example

$q = Ze$ ions

$e = 1.6 \times 10^{-19}$ C



$\lambda \sim 10^{-13} \rightarrow 10^{-7}$ C/m typical for intense beams

$$\frac{\#\text{ particles}}{\text{cm}} = \frac{\lambda}{Ze \cdot 100} \sim \frac{10^4}{Z} \rightarrow \frac{10^{10}}{Z} \qquad q = Ze$$

charge state

- Smoothing still important in lower dimensions and real beam is 3D

//

Practical Considerations

Computer Memory

Computer memory dictates how large a problem can be run.

- If the problem will not fit into fast memory (RAM), computer performance will be severely compromised.
  - writes to hard disks are slow.

3 main contributions to memory:

1) Particle phase space coordinates
2) Gridded field.
3) General code overhead.

} Dominate for large problems

Generally, particle and field quantities are stored in double precision:

Representation	digits (Floating Point)	Bytes memory
Single Precision	8	4
Double Precision	16	8

Relevant case for most problems

1) Particle Phase Space Coordinates

N_p = # macro particles

D = Dimension of variables characterizing particles.

$$\boxed{\text{Memory} = 8 \times N_p \times D \quad \text{Bytes}}$$

The dimension D depends on the specific type of simulation!

Examples:

3D $D = 7$: x, y, z; $p_x, p_y, p_z, \gamma^{-1}$ (for opt.)

2D $D = 4$: $\perp$ slice in s: x, y; p_x, p_y

2) Gridded field

For electrostatic problems typically need only store ϕ and ρ on the mesh.

- Discrete Fourier Transform complex, but transform is of real functions. Proper optimization can be carried out using only ϕ and ρ arrays.
- Electric field is typically not stored and is calculated for each particle only where needed.
  - Some methods do store gridded $\vec{E}$ to optimize specific problems / computers.

$$
\begin{aligned}
N_m &= \#\text{ mesh points.} \\
&= (n_x+1)(n_y+1) && \text{2D} \perp \text{Slice} \\
&= (n_x+1)(n_y+1)(n_z+1) && \text{3D}
\end{aligned}
$$

$\vdots$

$$\boxed{\text{Memory} = \overset{\text{Bytes}}{8} \cdot \overset{\phi,\rho}{2} * N_m \text{ Bytes}}$$

Dimensonality is also critical in field solves. One typically finds:

1D Simulations:

Fieldsolve trivial - only a small fraction of the work involved in moving particles.

- Green's functions can be used. (Gauss' Law)

2D Simulations

Fieldsolve is typically a small fraction of the work involved in moving particles if fast, gridded methods are used. (e.g. FFT). Special boundary conditions may increase this fraction.

Numerical Work	Method
Small fraction particle moving	FFT with periodic B.C.
	FFT with Capacity Matrix
	SOR
↓	⋮
Dominates particle moving	Green's Function

3D Simulations

Fieldsolve typically is comparable to or dominates the work involved in moving particles even if fast, gridded methods are employed.

- Fieldsolve efficiency of critical importance in 3D to optimize run time.
- You could teach whole classes just on 3D electrostatic field solve methods for Poisson's equation.

3) General Code Overhead

Some system memory is also used for:

- Scratch arrays for various numerical methods in fieldsolvers, movers, etc.
- Diagnostic routines, graphics packages, etc.

$$\boxed{\text{Memory} = M_{\text{overhead}} \sim \text{characteristic of given code} \quad \text{Typical } 1\text{MB} - 20\text{MB}}$$

Some modern graphics packages are large!

Total memory usuage (Electrostatic PIC)

$$\boxed{\text{Memory} = 8 \times N_p \times D + 16\, N_m + M_{overhead} \quad \text{Bytes}}$$

Machine RAM capacity should not be exceeded.

Speed of Computational Cycle:

The speed of the computational advance cycle is also extremely important and dictates the time that runs will take.

- Parts of the code that more time is spent in should be more carefully optimized to minimize operations
  - Particle mover
  - Field solver
  - Frequent diagnostics such as moments.

- Infrequent diagnostics, loaders, problem setup routines ect. often can be coded with less care since they are only executed infrequently. However, diagnostics often take much wall clock time to code!

It is often useful to benchmark the code in terms of

$$\boxed{t_{step} = \text{Time for a computational cycle in an "ordinary" run step.}}$$

Machine Architectures.

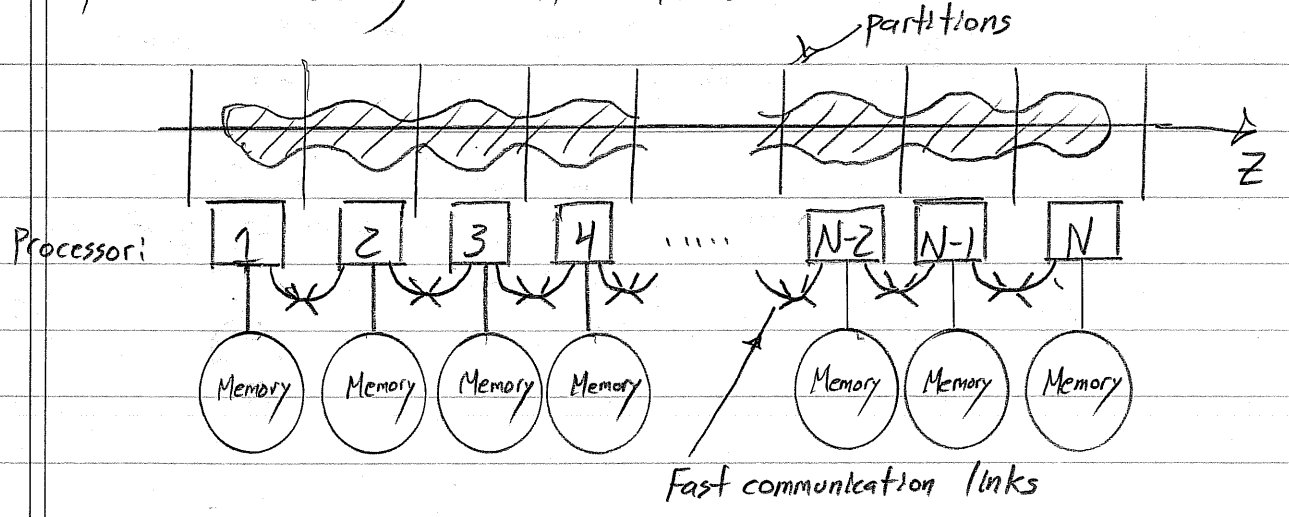
Problems may be simulated on:

1) Serial machines
   - sometimes small clusters of processors without specific need for parallel programing.

2) Parallel machines
   - multi-processors, possibly shared memory but more often linked memory via fast message processing.

Parallel machines have been recently improved with libraries that allow "natural" problem formulation with less effort and they promise to enable much larger simulations (resolution + statistics)
   - Several 100 Million particle simulations possible.

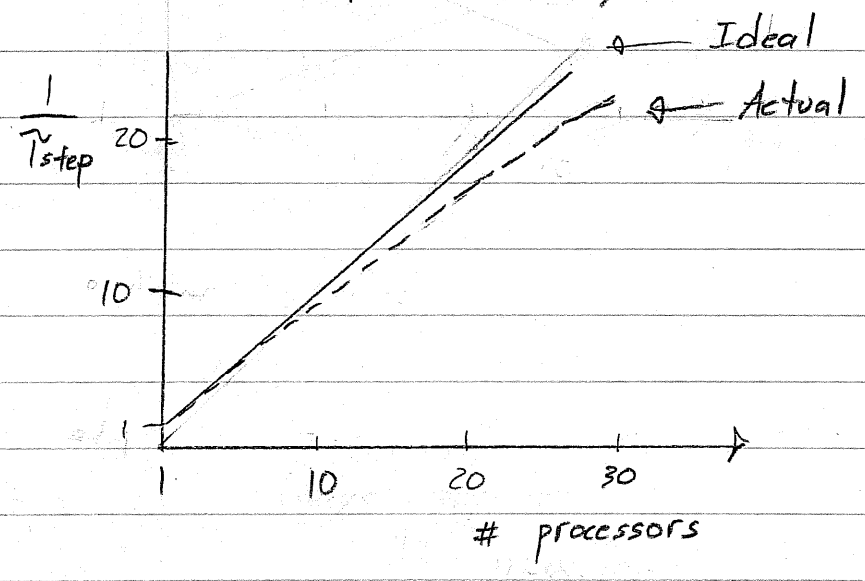
Beams problems may often be conveinently partitioned among processors using axial slices:



- Sharing of data at boundaries is needed for fieldsolves
- Some problems will require the partitions to be adjusted (particle sorting) to maintain the load balance.
  - processors should ideally share load equally since the slowest will limit the advance.

Ideal parallelization will result in a linear speedup with processor number.

τ_{step} = time per "ordinary" step of computational cycle.



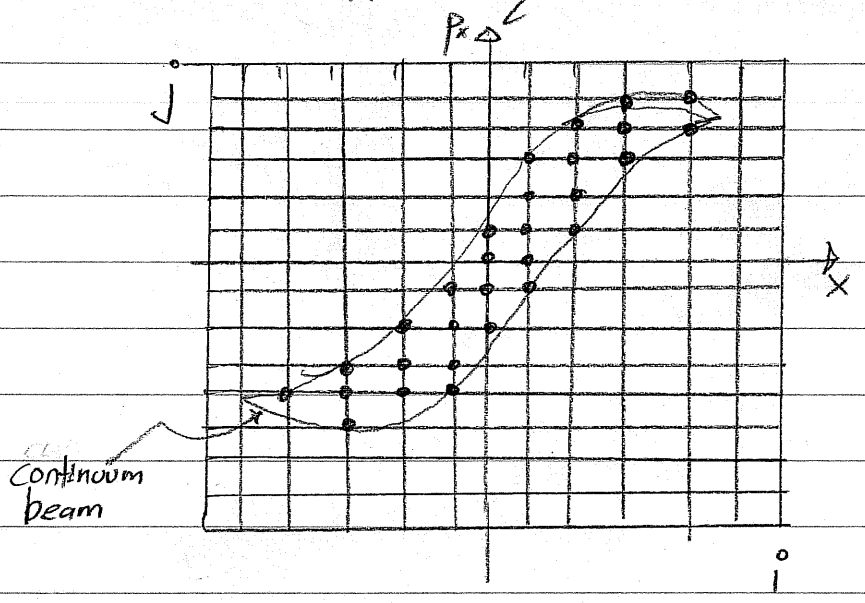
- Actual speedup will be less than ideal due to:
  - overhead in data transfers
  - less than ideal load balance among the processors, resorting of particles, etc.

Even with the significant advances in problem size and speed promised by parallel computers, the solution of realistic 3D beam problems with direct (not gridded) fields will remain far too large a problem to simulate in the forseeable future. Thus we will always try to exploit computer resources to the maximum extent possible

- Better numerical methods
- Parallelization .....

Distribution Methods – Direct Gridded Solution of Vlasov's Equation.

Consider the Vlasov equation as an example.



$$f(x_i, p_j, t) = f_{ij}(t)$$

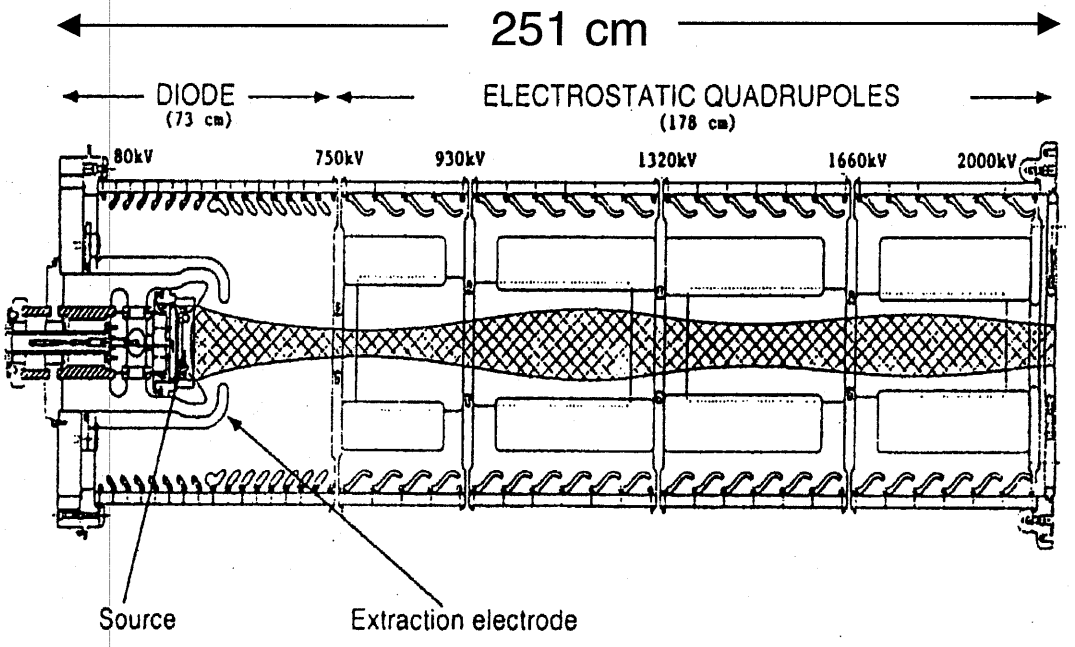
The distribution is advanced at discrete grid points in time.

- Fields are typically solved using a discrete spatial mesh as for the particle methods described before.
  - Deposition on mesh is typically straightforward in Vlasov case (sum over momentum variables.

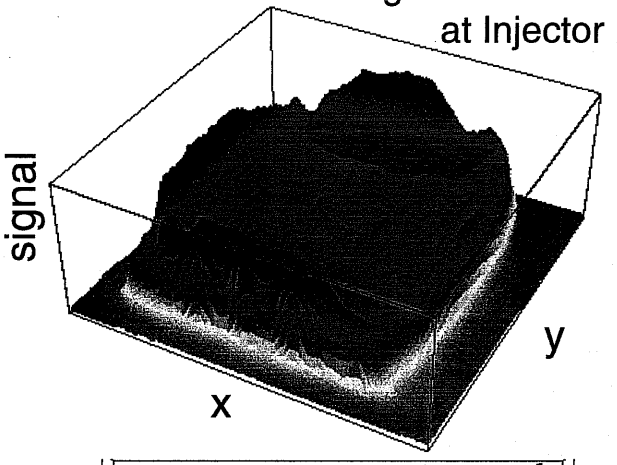
- The distribution advance cycle is different than for particle methods.
  - Numerical stability is key.
  - Characteristics and "semi-Lagrangian" methods can be employed.
  - Methods for solving for characteristics are familiar from dynamics/plasma physics.

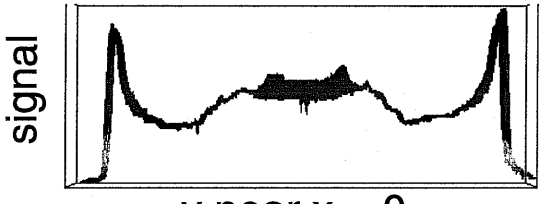
3D PIC simulations are being used to guide retrofits of an existing ESQ injector at LBL

1st principles, mid-pulse 3D simulations have been carried out to guide injector retrofits aimed at decreasing beam aberrations



Multi-Shot Film Images of Beam at Injector Exit





Parameters expected at exit of retrofitted injector:

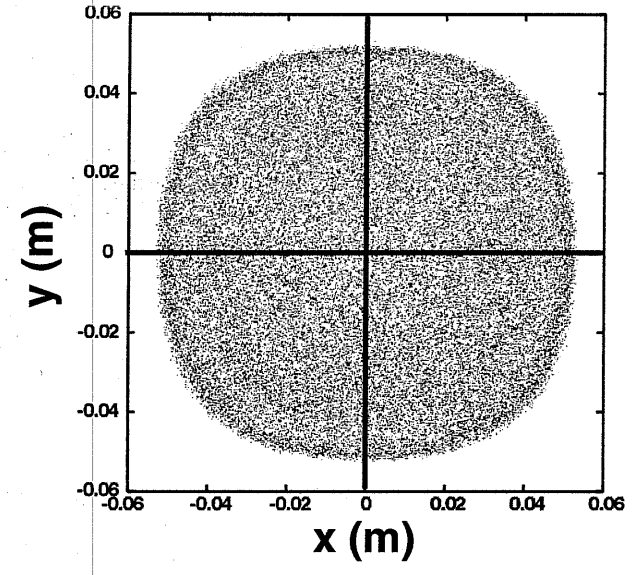
Energy:	**E = 1.71 MeV**	
Current:	**I = 692 mA, K**	
Emittance:	**ε_n = 1.1 π mm-mrad**	**rms edge measure**
	~ 0.7 π mm-mrad	**measure eliminating empty entrained space**
Envelope:	**r_x = 56.3 mm**	**r_x' = 53.8 mrad**
	r_y = 55.7 mm	**r_y' = -44.8 mrad**

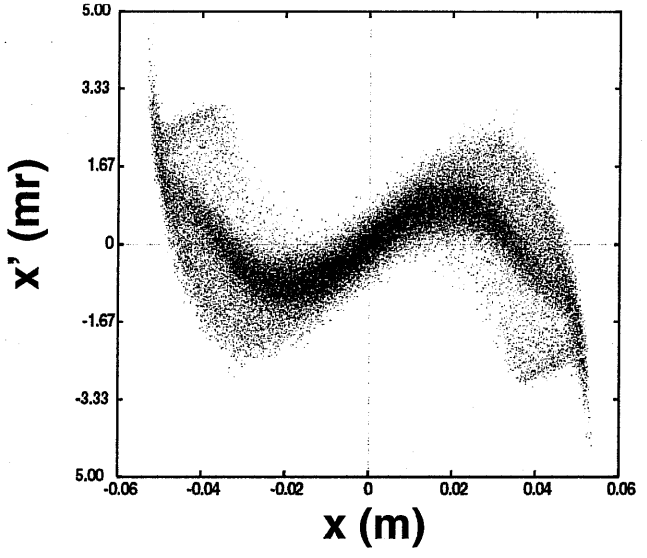
Example PIC Simulations

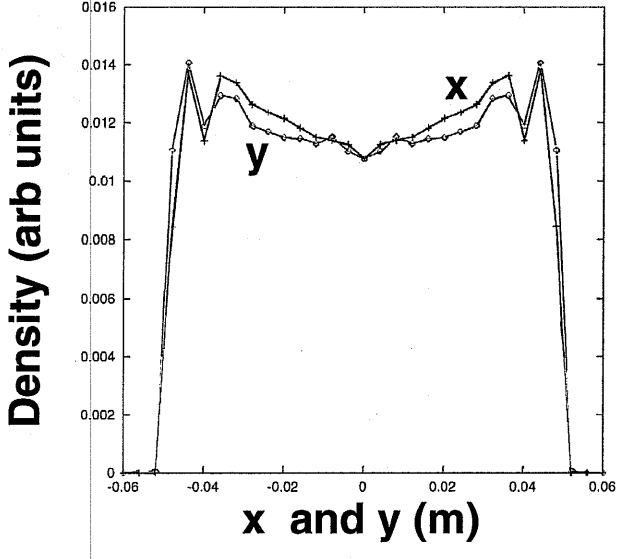
S.M. Lund 54/

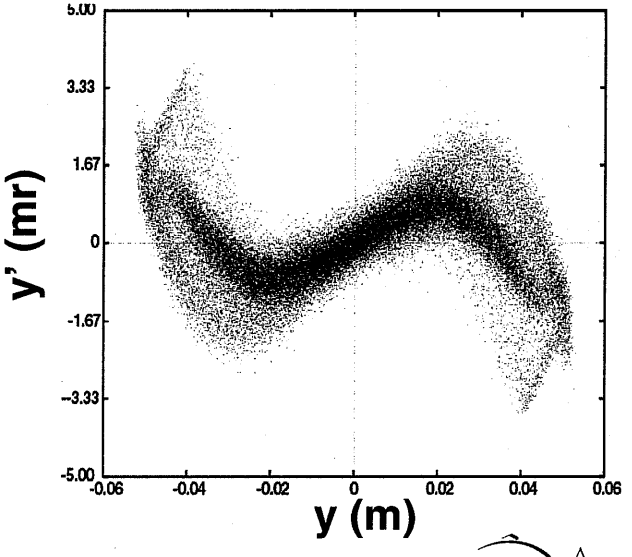
HCX Injector: Simulations show distribution distortions in the retrofitted injector should be more modest

Mid-pulse distribution projections at exit plane of injector









Overall convergence and divergence angles removed to illustrate distortions

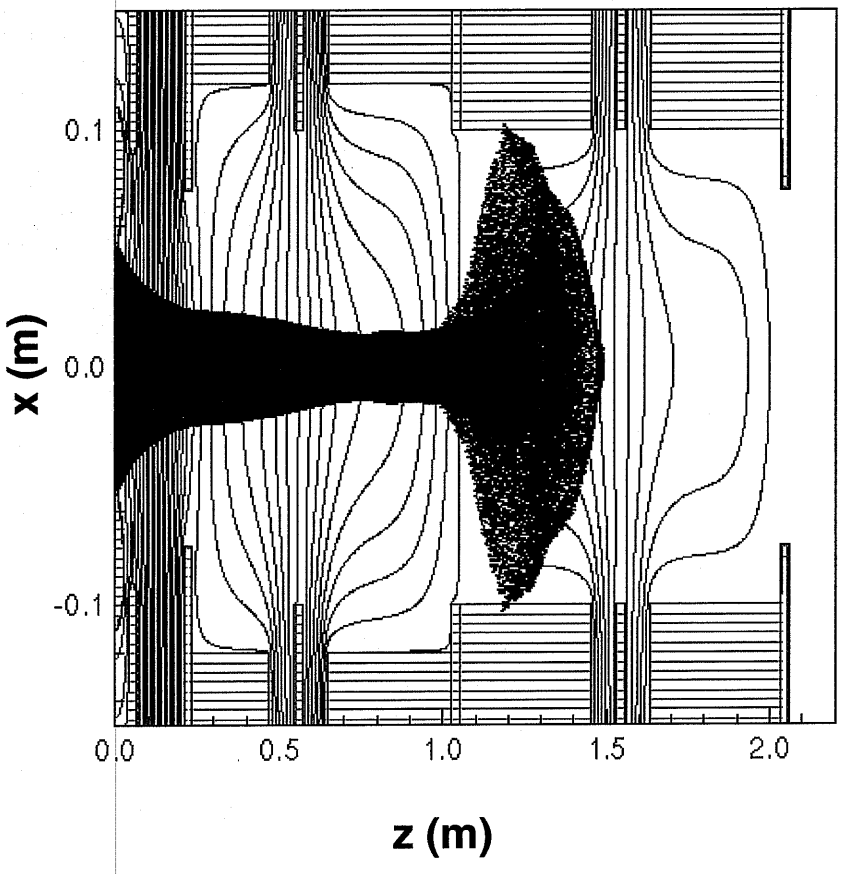
SM, Lund 55/

Simulations show how waveform rise-time determines beam head mismatch

Tuned 1d diode voltage waveform rise-time is 400 ns -- deviations from this lead to significant mismatch effects in the beam head and particle loss with resultant worries about breakdown, electron effects, etc.

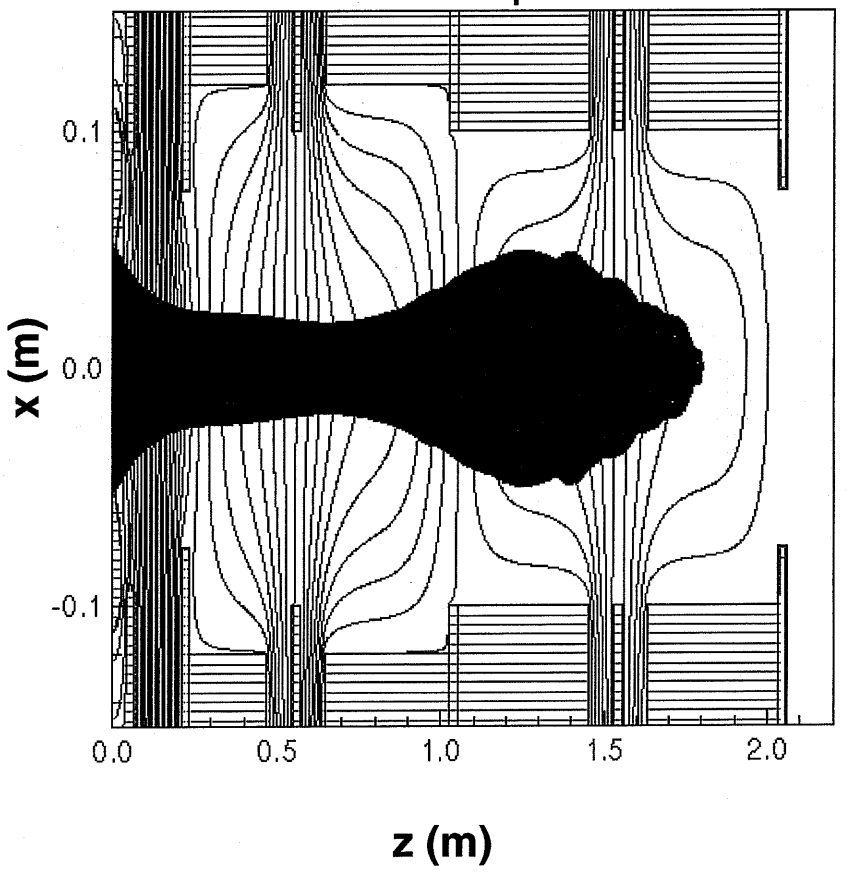
Rise-time τ = 800 ns

beam head particle loss < 0.1%



Rise-time τ = 400 ns

zero beam head particle loss

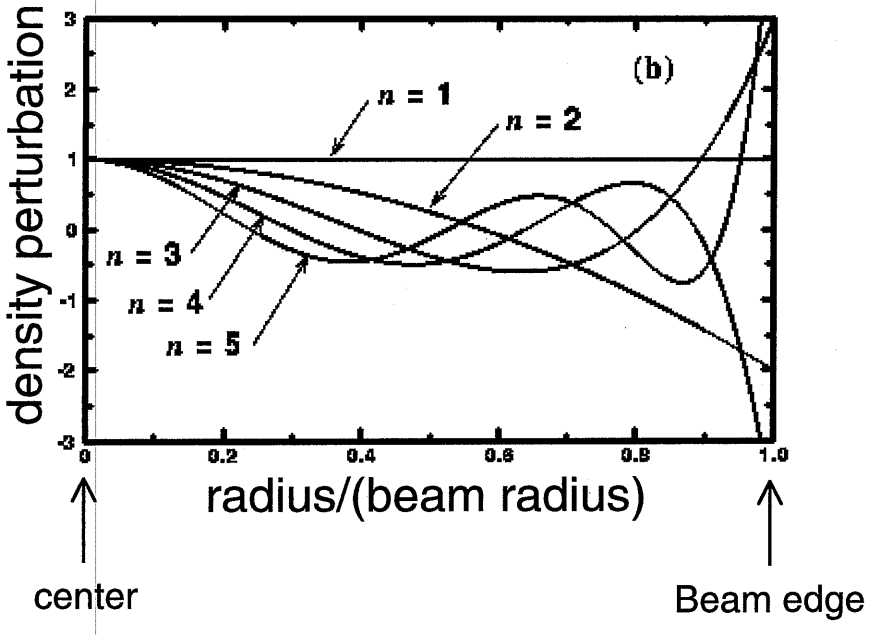


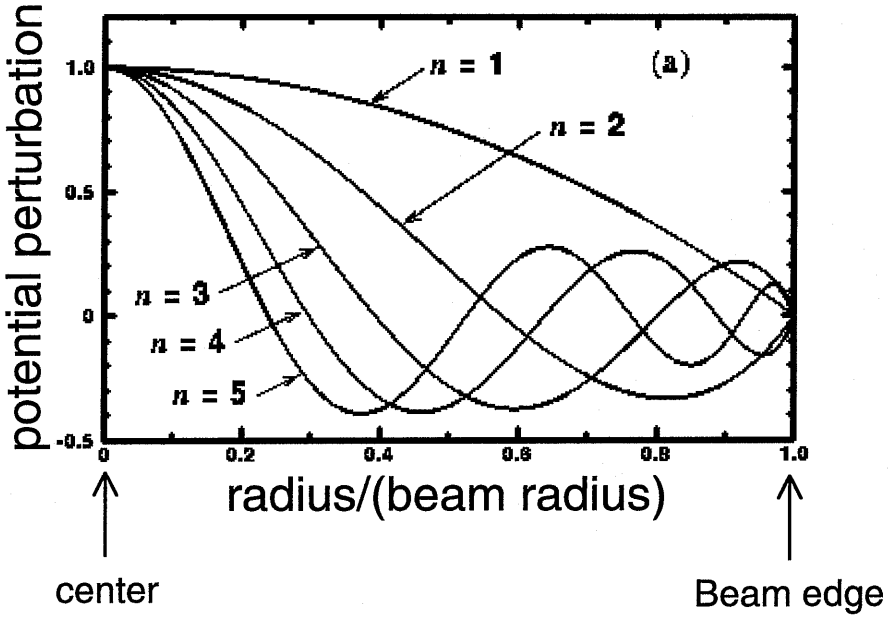
S.M. Lund 56

Initial distribution distortions will launch a spectrum of collective mode perturbations that evolve

Kinetic and fluid theories have been employed to analyze perturbations on a uniform density intense-beam equilibrium [Lund and Davidson, Phys. Plasmas, 5 3028 (1998)]

Small Amplitude Perturbations (arbitrary units, kinetic and fluid theory)





Mode Dispersion Relation (fast branch, from fluid theory)

$$\frac{\sigma_n}{\sigma_0} = \sqrt{2 + 2\left(\sigma/\sigma_0\right)^2 (2n^2 - 1)}$$

σ_n = mode phase advance

n = 1,2,3, ……

Example:

$\sigma_0 = 80^o$, $\sigma/\sigma_0 = 0.2$

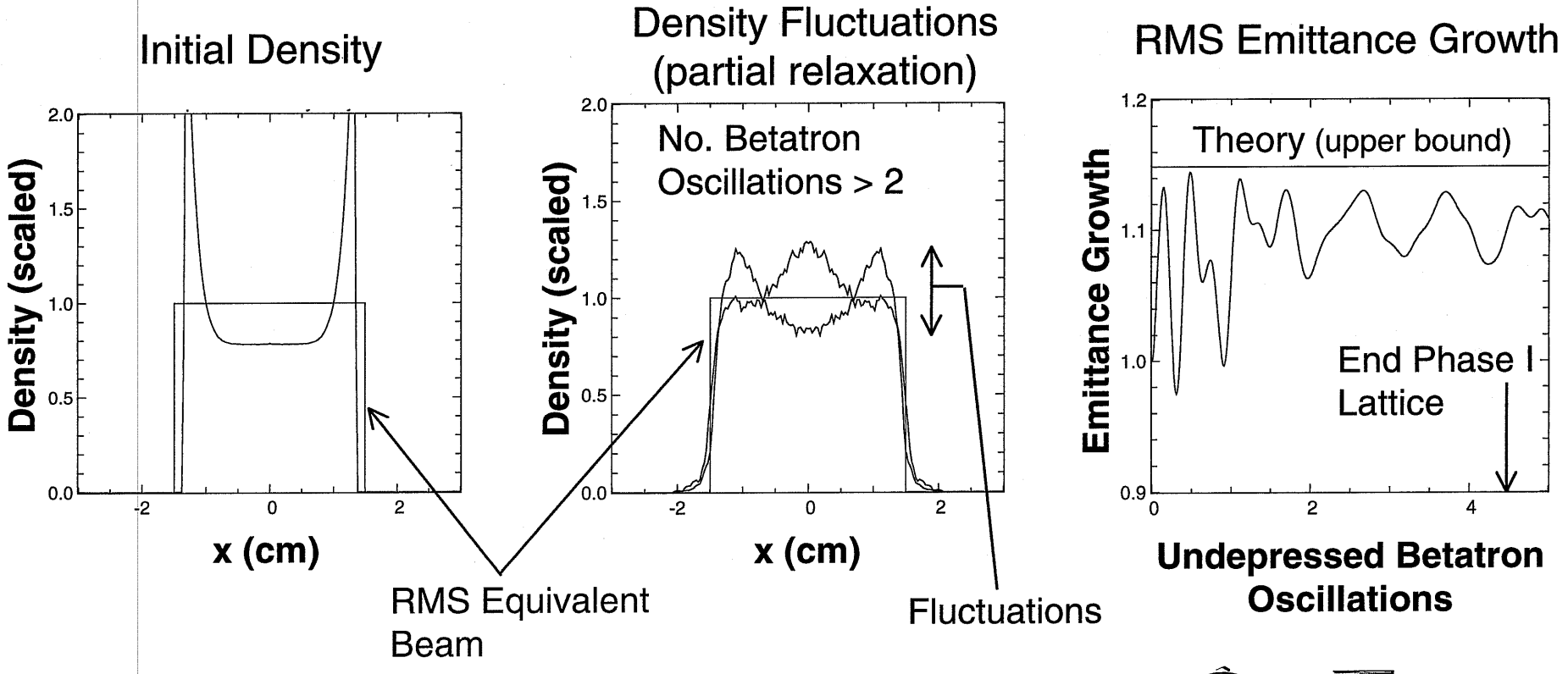
$\sigma_1 = 115^o$, $\sigma_5 = 182^o$, ….

S.M. Lund 57

Perturbations launched by initial distribution nonuniformities can phase-mix to a more uniform profile with increased emittance

Mode spectrum launched can undergo a rapid cascade, settling to a smaller amplitude and lower order distortion

- Approximate conservation constraints employed to bound emittance increases resulting from full relaxation to a uniform profile [Lund, Lee, and Barnard, Proc. Linac 2000, pg. 290]
- How will such evolutions influence the range and interpretation of measurements

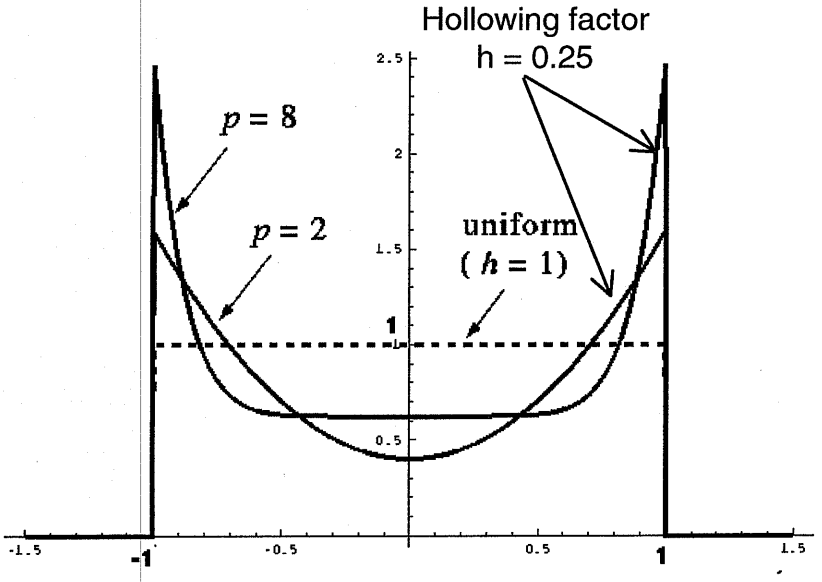


S.M. Lund 58

Analytic theory has been used to parametrically bound emittance growth due to the relaxation of space-charge nonunifomities

Approximate conservation constraints can be employed to estimate maximal emittance increases resulting from the relaxation of an initial nonuniform density profile to a final, uniform profile [Lund, Lee, and Barnard, Proceedings Linac 2000, Monterey, CA, pg. 290]

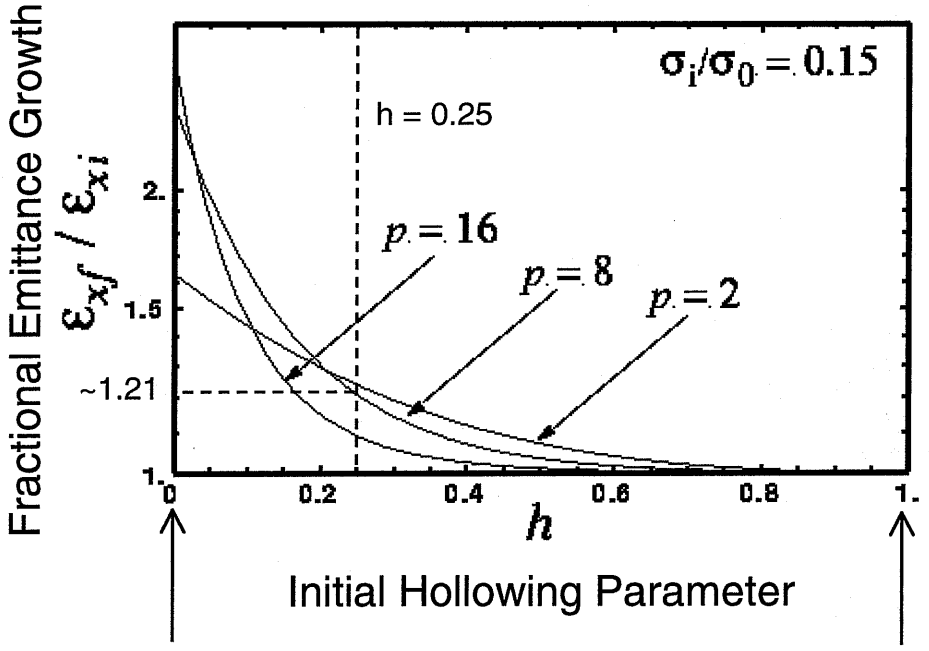
Initial Density



hollowing ~ r^p

h = ratio min to max density

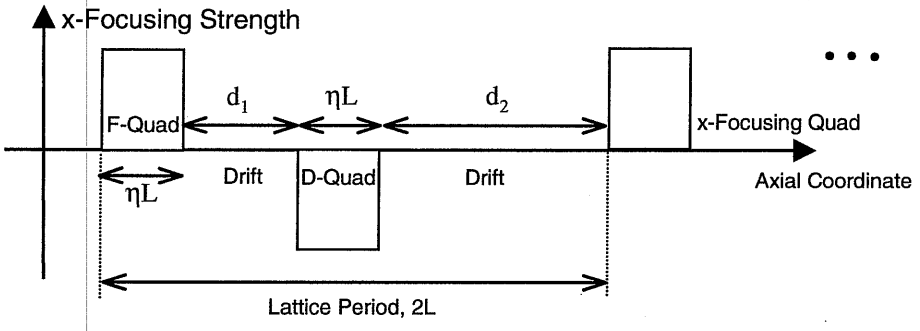
Emittance Growth on Relaxation



S.M. Lund 59/

HCX Phase II: A wide range of parametric simulations have been carried out with realistic magnetic transport lattices

A syncopated magnetic transport lattice of ~ 50 lattice periods has been designed



HCX:

η = Quadrupole Occupancy $(0 < \eta < 1)$ $\eta = 0.453$

α = Syncopation Factor $(0 < \alpha < 1)$ $\alpha = 0.248$

Drifts:

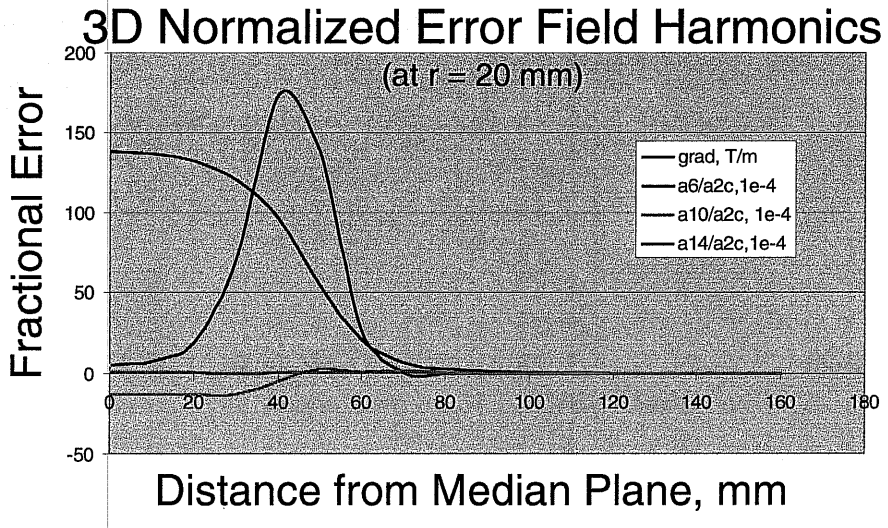
$d_1 = \alpha(1-\eta)\eta L$

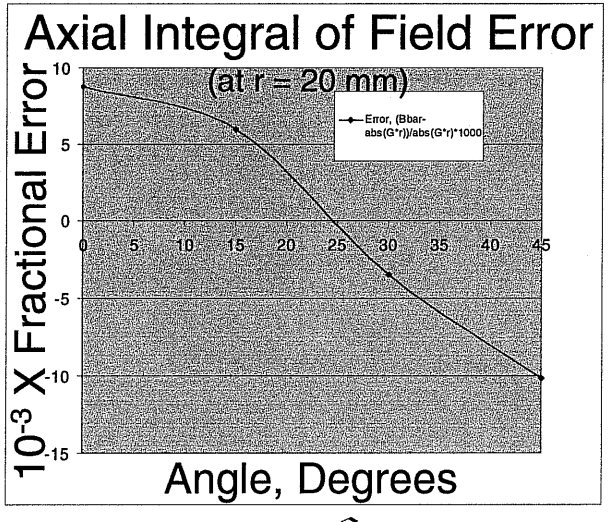
$d_2 = (1-\alpha)(1-\eta)\eta L$

Superconducting magnetic quadrupoles have been designed and prototyped

$r_p = 29.5$ mm $B_q' = 104$ T/m (max) $B_q \sim 6$ Tesla (wire)

$\ell = 136$ mm $\ell_{eff} = 101$ mm Integrated Field Error $< 15 \times 10^{-3}$



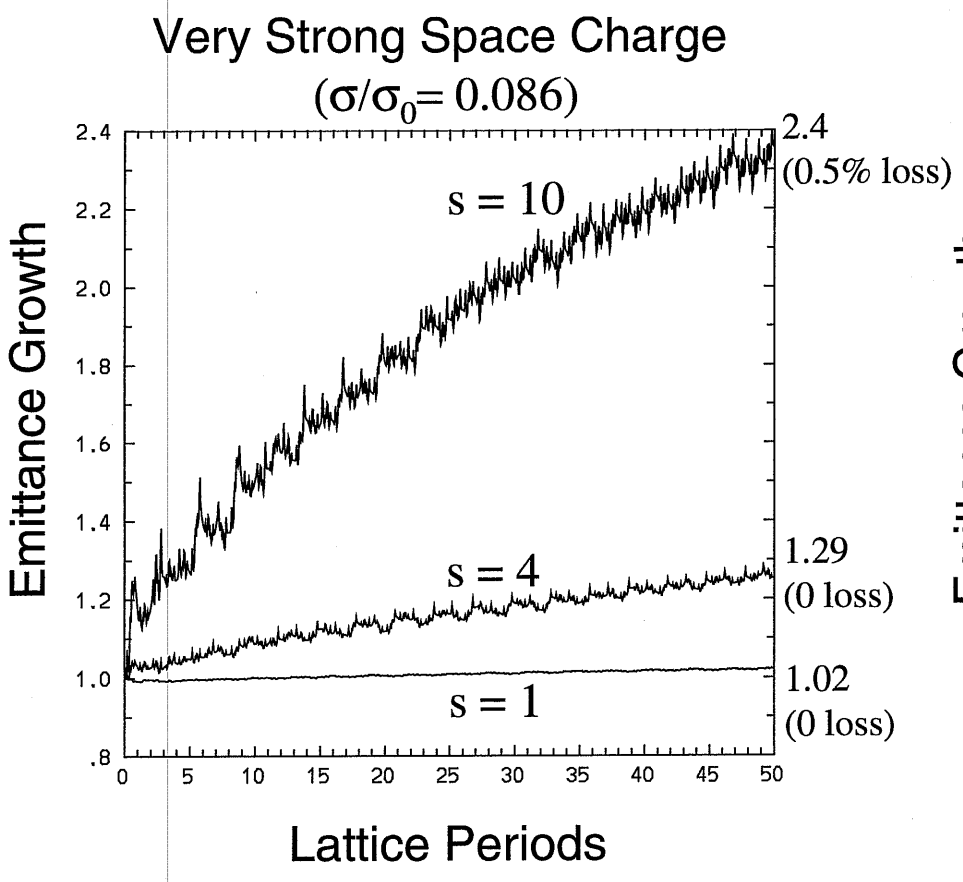


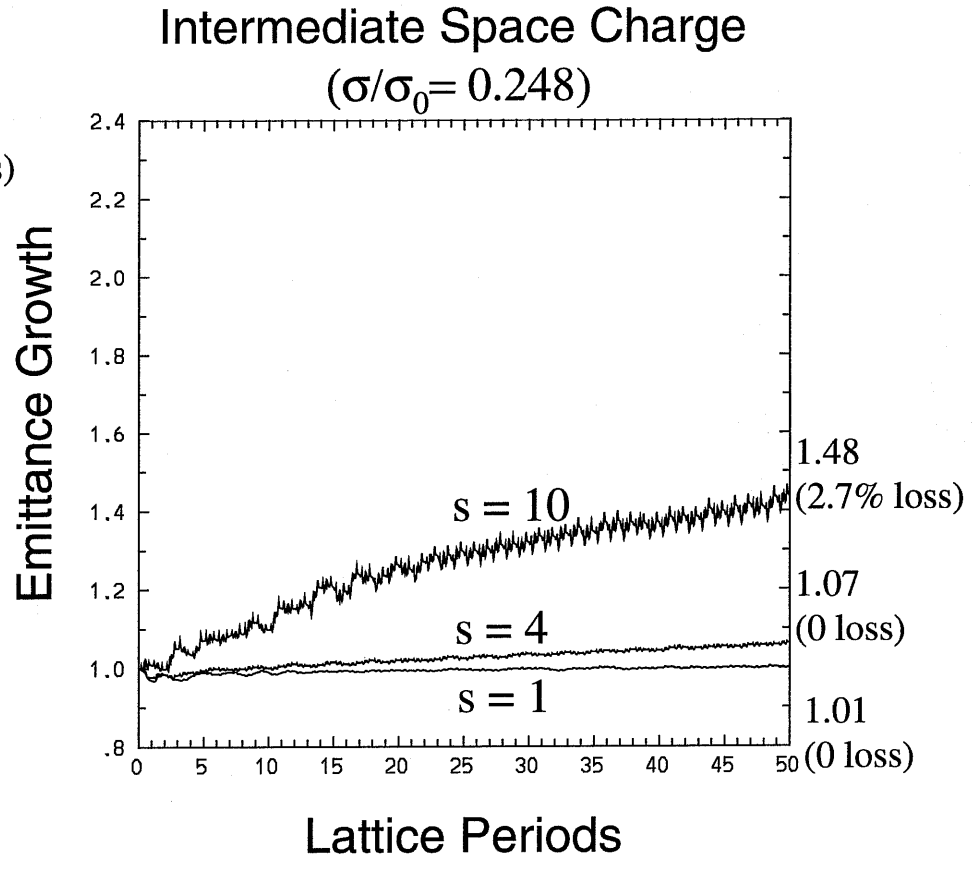
S.M. Lund 60/109

HCX Phase II: Processes influencing beam quality and control have been explored --- Example: Nonlinear applied fields and beam quality

Full 3D magnetic field is resolved as:

$$\vec{B} = \vec{B}_{quad} + \delta\vec{B} \qquad \delta\vec{B} \rightarrow s \cdot \delta\vec{B}$$





S.M. Lund 6/1/

S.M. Lund 62/

Initial KV Distribution

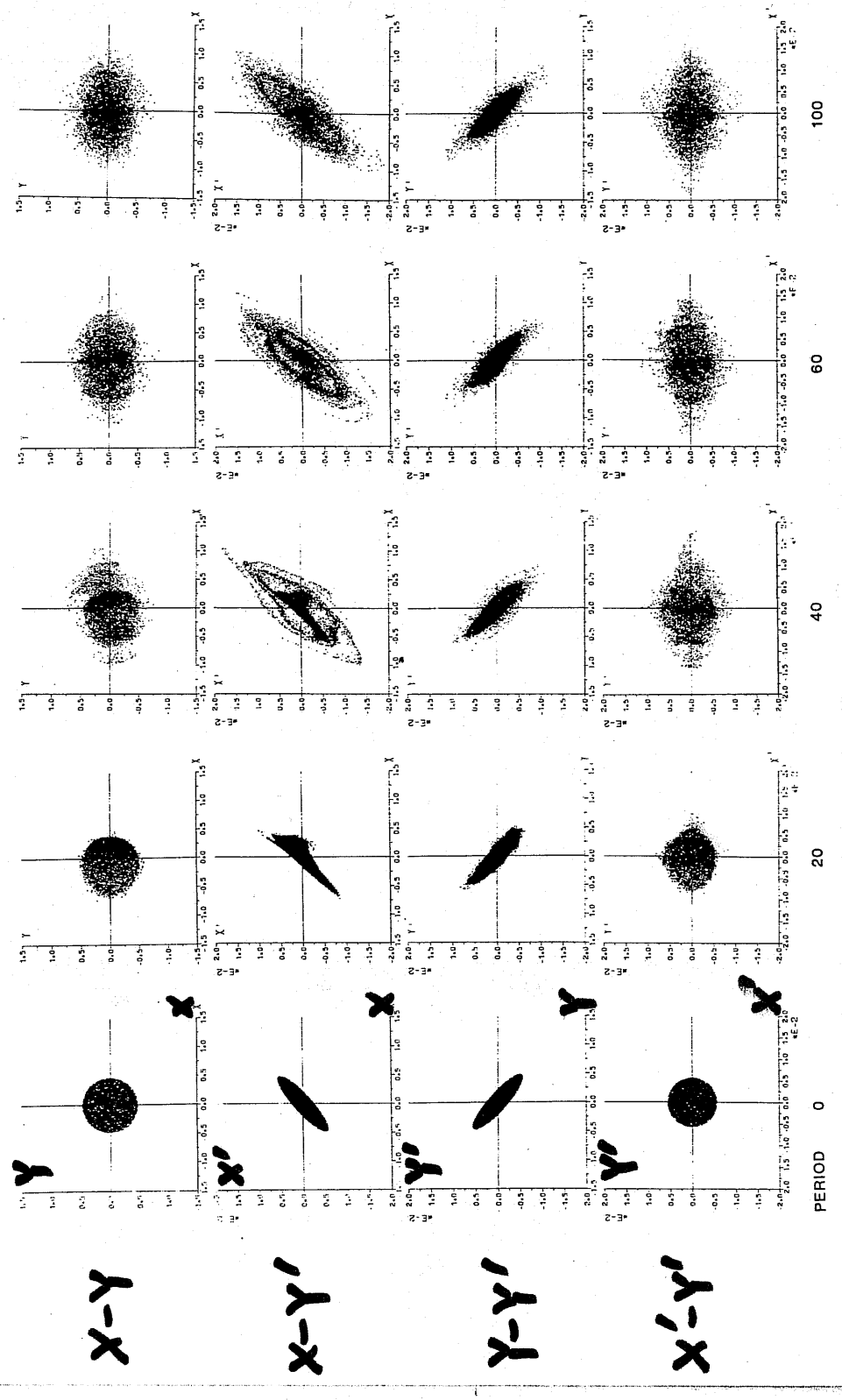


FIGURE 2 Transformation of an initial K-V distribution through the GSI FODO channel at $\sigma_0 = 90°$, $\sigma = 41°$.

J. Struckmeier, J. Klabunde, and M. Reiser, Part. Accel. 15, 47 (1984).

See also Reiser Text, pg 495.

S.M. Lund 63/

Initial Gaussian Distribution

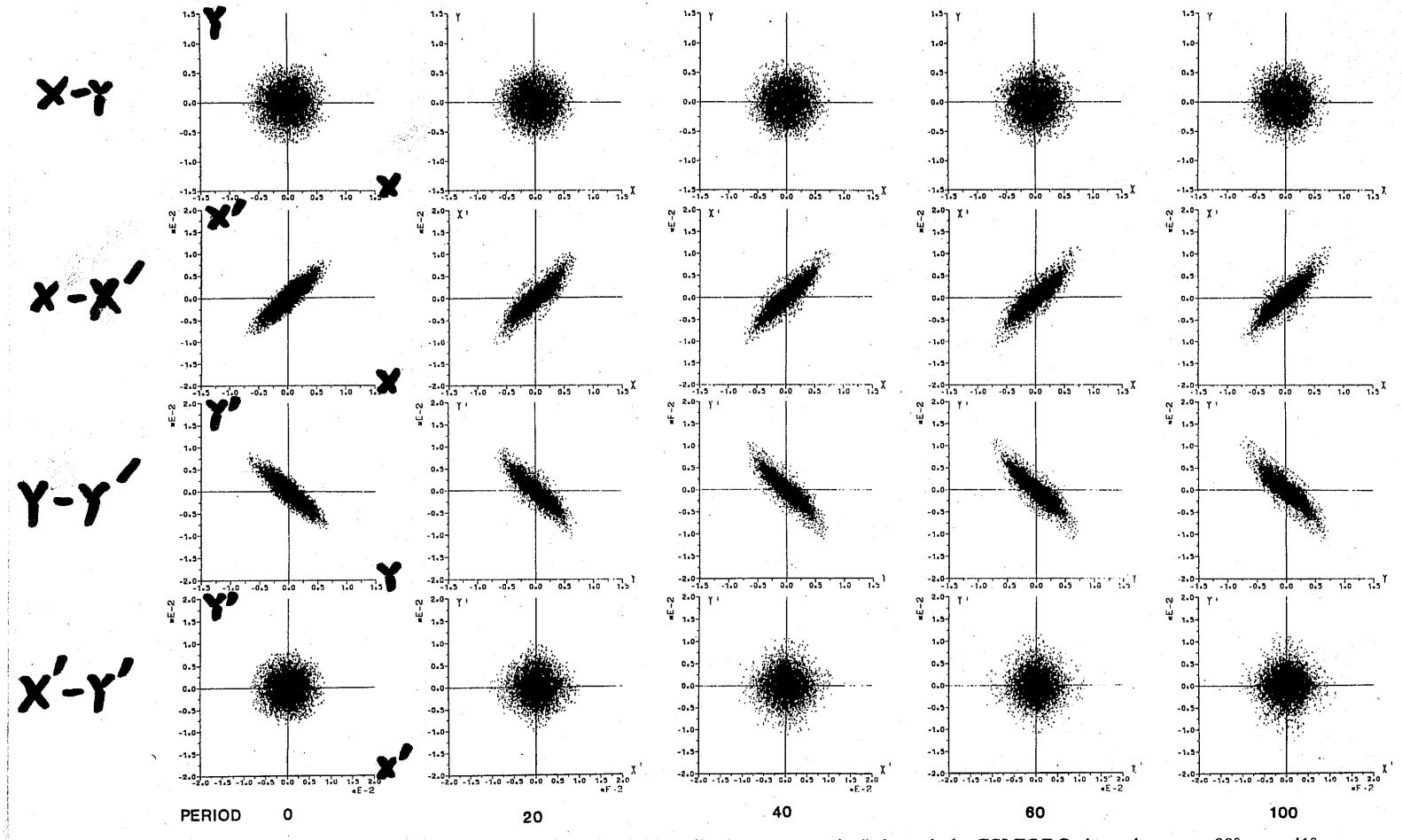


FIGURE 6 Transformation of an initial Gaussian distribution (rms-matched) through the GSI FODO channel at $\sigma_0 = 90°$, $\sigma = 41°$.

J. Struckmeier, J. Kabunde, and M. Reiser, Part. Accel. 15; 47 (1984)

S. M. Lund 64

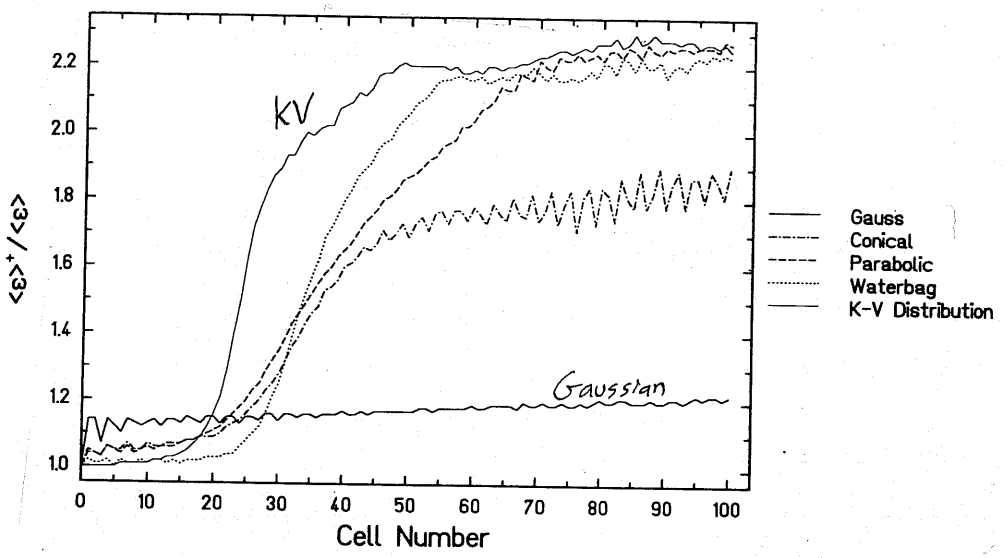


FIGURE 7 Emittance growth factors versus the number of cells obtained from particle simulations for initial K-V, waterbag, parabolic, conical and Gaussian distributions at $\sigma_0 = 90°$, $\sigma = 41°$.

From J. Struckmeier, J. Kabunde, and M. Reiser, Part. Accel. 15 47 (1984).

Direct Vlasov Methods:

"Pros"

Reasons for Vlasov simulations:

- Low noise – only discretization effects without statistical effects
- Allows clear analysis of collective effects and tenuous distribution components.

"Cons"

Reasons why Vlasov simulations are presently employed less than PIC:

- Extreme memory requirements for needed grid resolution in multi-dimensional phase space.
  - Beams often have sharp edges in phase space that move in responce to varying applied focusing forces.
- Numerical stability tends to be more difficult than in particle methods.

No time to illustrate Vlasov and other distribution methods in this introductory course. We hope to cover these methods and other numerical fieldsolve techniques in a later expanded version of this class.

However ..... it is easy to generalize from what we have learned!

WARP Code Overview

Electrostatic Multi-dimensional PIC Code

WARP3d – x, y, z, p_x, p_y, p_z
Moves in t

Many Fieldsolvers!
SOR, Multigrid, FFT,
FFT + Tridiag, FFT + Cap Matrix, ..

WARPxy – x, y, p_x, p_y, p_z
Moves in s

WARPrz – r, z, p_r, p_z, p_θ
Moves in t

WARPenv – envelope solver
used to seed / load PIC
r_x, r_y, r_x', r_y'
Advances in s

Hermes – Fluid II + Gridded
Space Charge Field

- Common diagnostic tools built around gist graphics
- Run with python interpreter

Example Script
"ag-slice.py"

Acknowledgements, References

Numerical Methods

1. Forman S. Acton, "Numerical Methods that Work," Harper and Row Publishers, New York, 1970.
2. Stephen Koonin, Computational Physics," Addison-Wesley Publishing Co., 1986.

Survey of Numerical Methods

3. W. Press, B. Flannery, S. Teukolsky, W. Vetterling, "Numerical Recipes, The Art of Scientific Computing," Cambridge University Press, 1986.

Particle Methods

4. C. K. Birdsall and A. B. Langdon, "Plasma Physics via Computer Simulation," McGraw-Hill Book Company, 1985.
5. R. W. Hockney and J. W. Eastwood, "Computer Simulation using Particles," Institute of Physic Publishing, 1988.

A special thanks to Alex Friedman, Dave Grote, Jean-Luc Vay, and Michiel de Hoon of the Lawrence Livermore and Lawrence Berkeley National Labs for help with these notes and general guidance. Also thanks are due for Irving Haber (NRL), Christine Celata (LBL), and Bill Fawley (LBL) for educating the authors on various simulation methods.