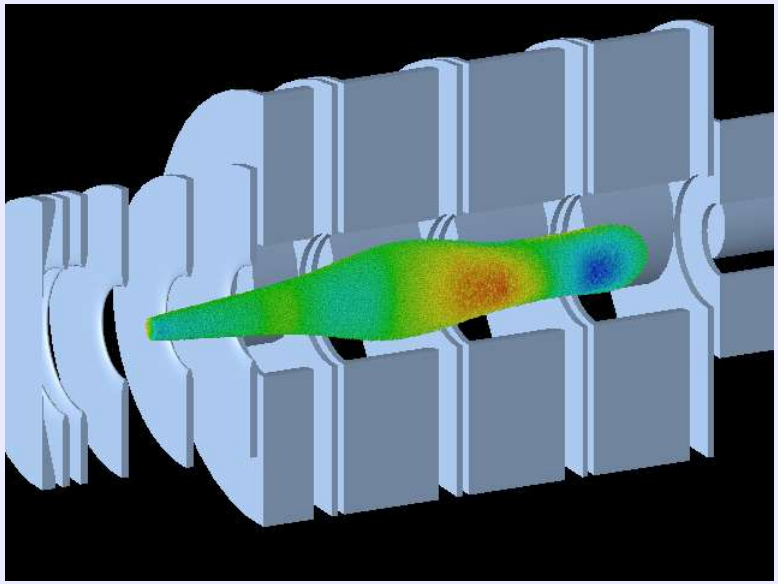
The WARP code: modeling high intensity ion beams



D. P. Grote, A. Friedman, HIF-VNL LLNL/LBNL
J.L. Vay, HIF-VNL LBNL
I. Haber, U. of Maryland

ECRIS'04, September 28, 2004

Warp originally developed for HIF

- Heavy-ion driven inertial confinement fusion requires high-current, high-brightness beams – space-charge dominated
- Warp designed to allow flexible and detailed multi-dimensional modeling of such beams in a wide range of systems
- Warp uses PIC method to model beams (non-neutral plasmas)
- Combined with accelerator lattice description
- Warp is a time-dependent code (as opposed to s-dependent or mapping)

Warp combines many pieces

PIC with self-field (electrostatic for now) in a moving window		
WARP3d	WARPrz	WARPxy
x, y, z, px, py, pz Warped-Cartesian	r, z, pr, pθ, pz Impedance model	x, y, px, py, pz Transverse slice

"Lattice" of focusing, bending, and accelerating elements			
Sharp-edged	Fully-detailed		First principles
Residence corrections	Axially varying multipole moments	Data on a 3-D grid	Electrostatic elements with subgrid resolved conductors
Envelope equation solver			
Envelope/fluid code (Circe/Hermes)			

Python interpreter, extensive diagnostics, ...

Space-charge domination best handled with time-dependent methods

- Very high space-charge (self-forces comparable to applied forces)
- Very extended beams (mapping beam to single z breaks down)
- Several modes of operation:
  - Purely time-dependent mode
  - Quasi-time-dependent mode ($\Delta t_{fieldsolve} > \Delta t_{particles}$)
  - Iterative/steady-state mode

Relativistic mover, but electrostatic self fields (no self-magnetic field yet)

PIC particles are flexibly handled

Leap-frog advance - 2nd order accuracy (either standard or isochronous)

Cloud in cell interpolation to grid

Multiple species

- Each species has its own mass, charge, weight

Distributions

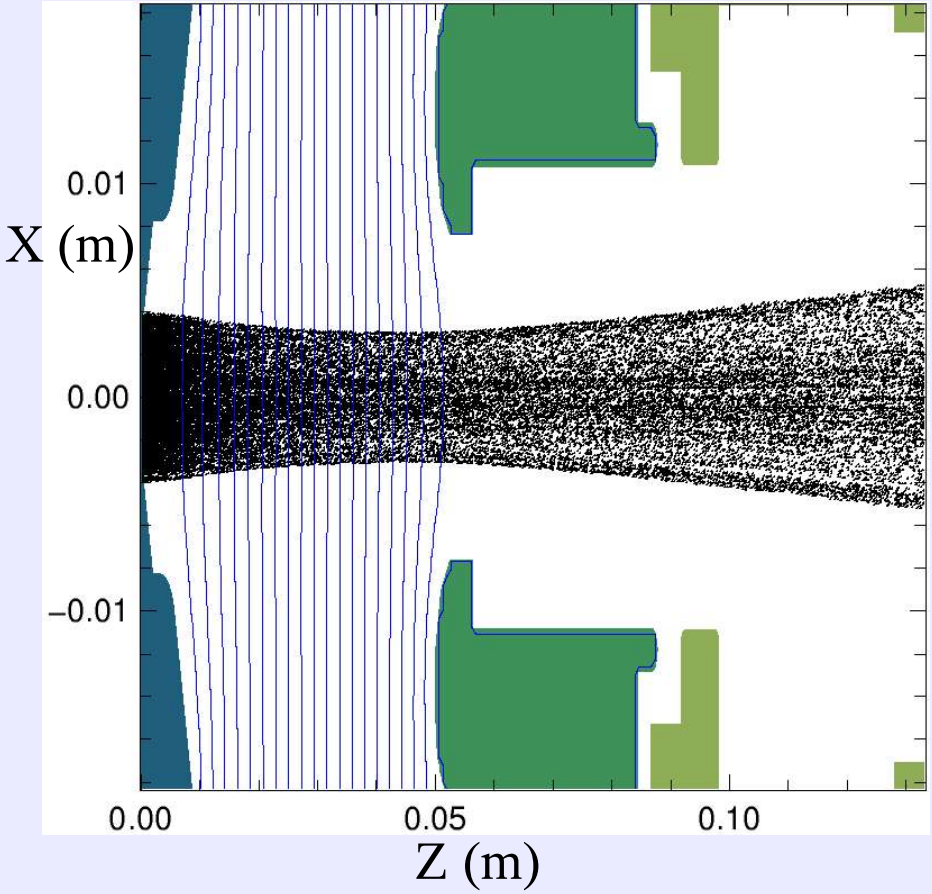
- K-V, Semi-Gaussian, hollow or peaked, user defined f(r)
- Water-bag, parabolic, Gaussian (in development)
- Injection from source
- Secondaries from wall losses (in development)
- User created, arbitrary

Injection

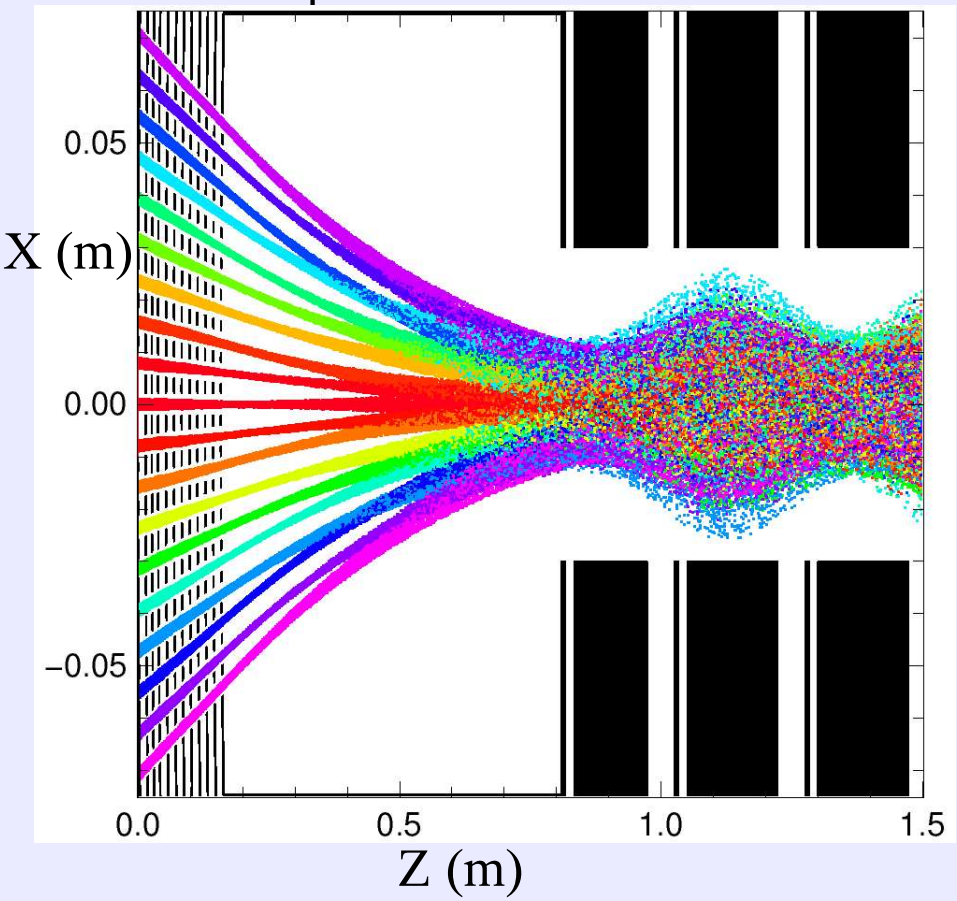
- Implemented in both 3-D and R-Z
- Flat or curved emitter surfaces (CL includes curvature correction terms)
- Injection algorithms:
  - Specified current (can vary in time)
  - Space-charge limited
    - Child-langmuir $N_{particles} = \frac{4}{9}\epsilon_0 \left(\frac{2q}{m}\right)^{\frac{1}{2}} \frac{\phi^{\frac{3}{2}}}{(\delta \Delta z)^2} \frac{\Delta x \Delta y \Delta t}{qw\cos(\alpha)}$
    - Gauss pill-box $N_{particles} = \left(\epsilon_0 \frac{\phi}{\Delta z \delta} - \rho \Delta z\right) \frac{\Delta x \Delta y}{qwcos(\alpha)}$
  - Plasma source (in development – see poster #58, Tuesday, Jin-Soo Kim)
- Grid refinement for improved time-dependence

Injection examples

VENUS source, H^+, 5 mA
Flat emitter surface
Steady-state mode

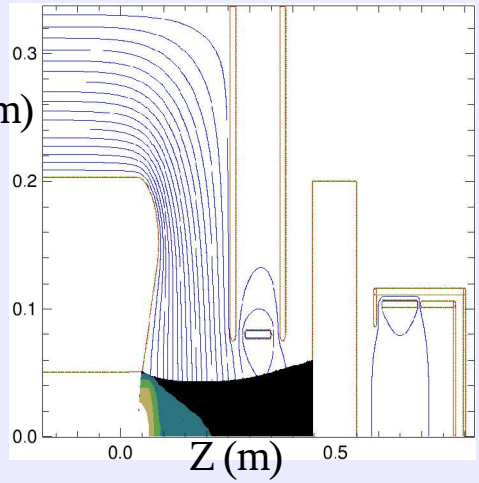


Beamlet merging injector
Flat emitter surfaces
Multiple beamlets



Simulation results compare well to experiment

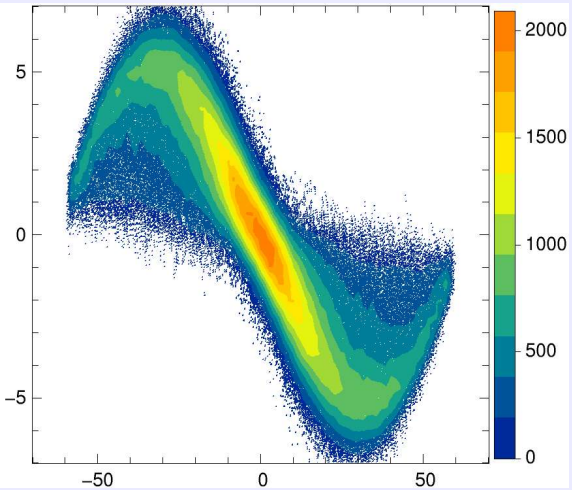
STS500 diode experiment



Phase Space at End of Diode

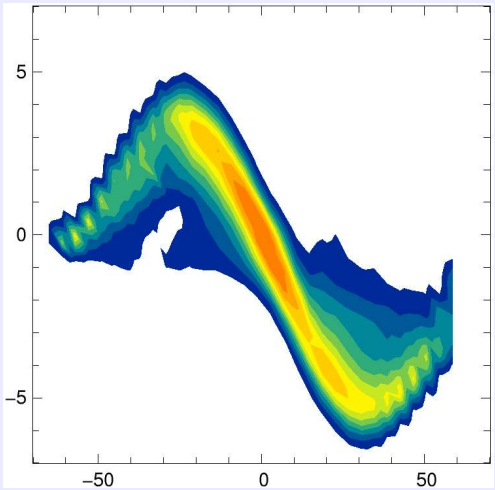
Warp simulations

x vs. x′



Experimental results

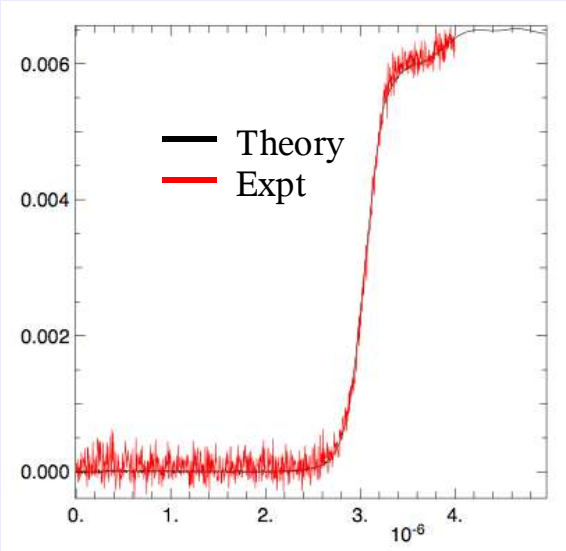
x vs. x′



10-cm diameter K+
Alumino-silicate source

Risetime

Current at Faraday cup



This result depends critically on new computer technique (Adaptive Mesh Refinement), that has moved the state-of-the-art.

Field Solvers

	FFT	FFT/Tridiag	SOR	Multigrid	AMR Chombo
3-D	Sine-sine-periodic	Sine-sine-tridiag	Red-black ordering	SOR rectangular cells	
Boundary conditions	Dirichlet transverse Periodic longitudinal	Dirichlet	Dirichlet, Neumann, Periodic		Dirichlet
Internal conductors	Simple via capacity matrix		Arbitrary with subgrid resolution		

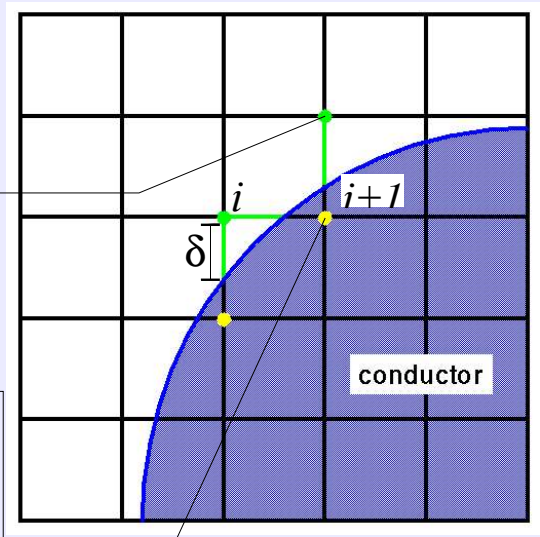
	FFT slice	Multigrid r-z with AMR
	Sine-sine	SOR
Boundary conditions	Dirichlet transverse	Dirichlet, Neumann, Periodic
Internal conductors	Simple via capacity matrix	Arbitrary with subgrid resolution

Subgrid resolution give 2nd order accuracy on internal boundaries

- Also known as cut-cell, Shortly-Weller

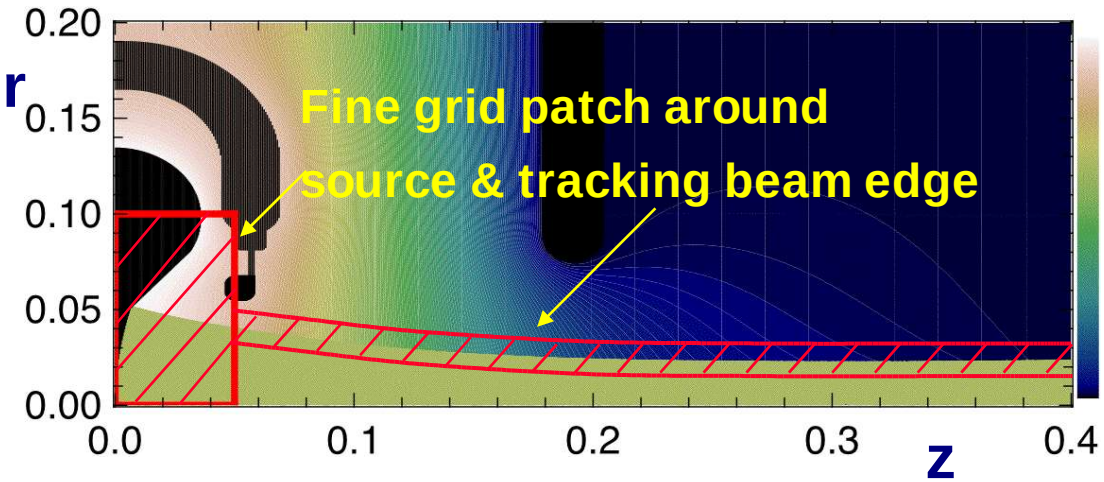
The finite difference form of Poisson's equation for the points near the surface is changed to explicitly include the surface.

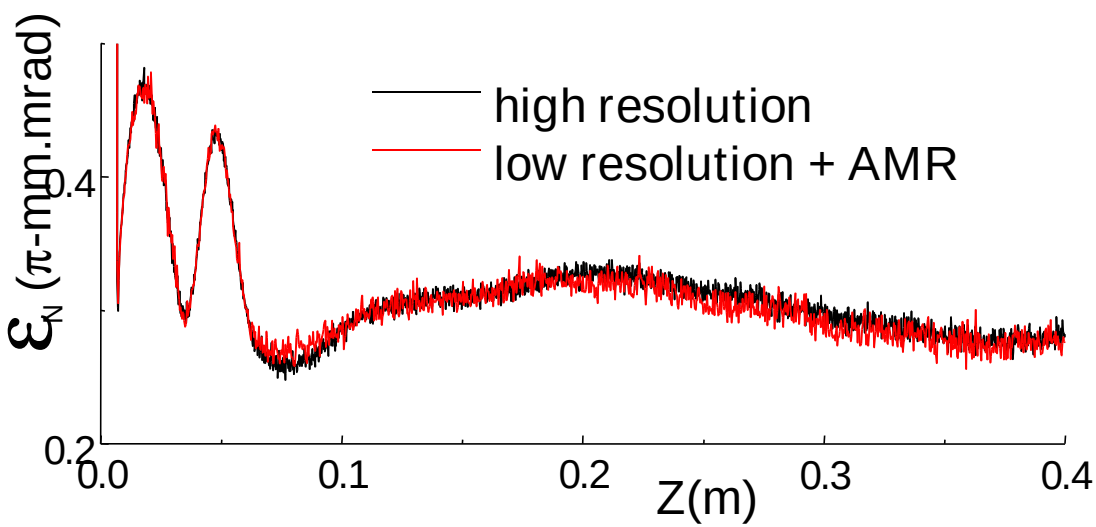
The change is made by replacing the potential of the points inside the conductor with a value extrapolated from the surface.



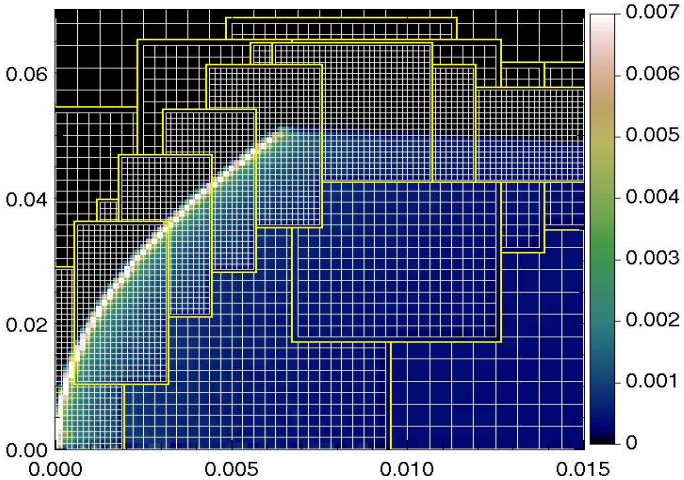
$$\frac{(\phi_{conductor}-(1-\delta)\phi_i)/\delta-2\phi_i+\phi_{i+1}}{\Delta^2}=-\frac{\rho_i}{\epsilon_0}$$

Prototype adaptive mesh refinement implemented in WARPrz, being developed in 3-D





Automatic generation of mesh refined patches



In this case: speedup ~ 4, other cases more

Warp's lattice description is designed for flexibilty

Fields applied directly to particles (no maps)

Elements can overlap

Can use MAD like, hierarchical, lattice description

Several levels of description of fields:

- Hard edged, axially uniform
  - Simplest model
  - Residence corrections to achieve 2nd order accuracy. On entrance and exit, field scaled by fraction of timestep inside element.

Lattice (continued)

- Axially varying moments
  - For each component, magnitude, $e_{n\nu}$ or $b_{n\nu}$, and angle, ψ_n saved in table
  - Linear interpolation between data points

$$E_r = -\sum_{\nu=0}^{\infty} \frac{1}{2(\nu+1)} \frac{de_{o\nu}(z)}{dz} r^{2\nu+1} + \sum_{n=1}^{\infty}\sum_{\nu=0}^{\infty} e_{n\nu}(z)[1+\frac{2\nu}{n}] r^{n-1+2\nu} \cos[n\theta+\psi_n(z)]$$

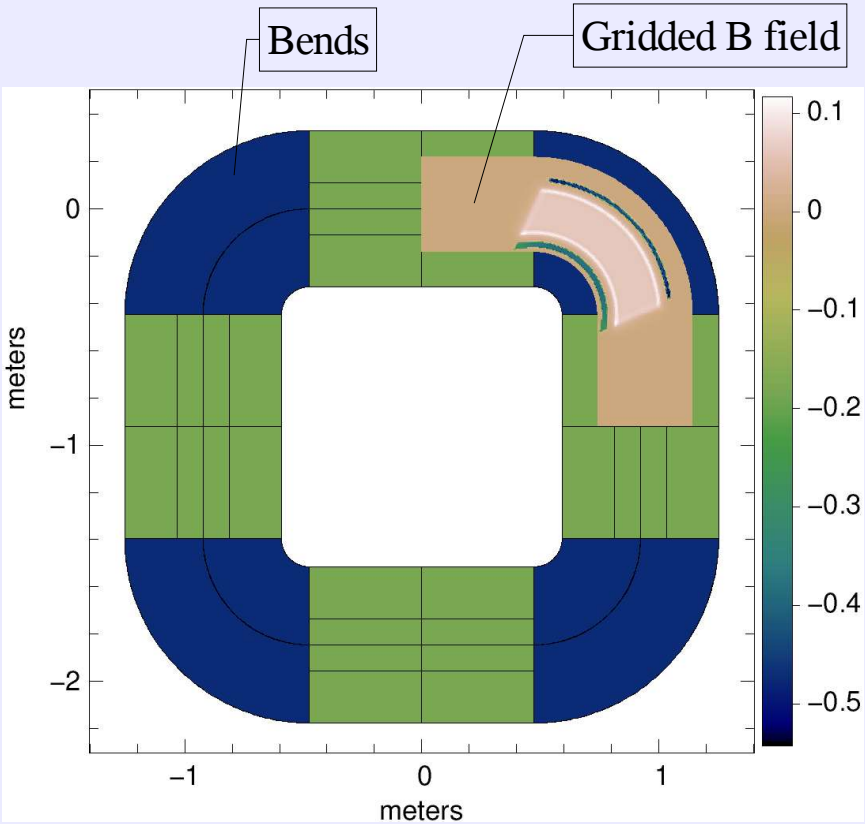
$$E_\theta = -\sum_{n=1}^{\infty}\sum_{\nu=0}^{\infty} e_{n\nu}(z) r^{n-1+2\nu} \sin[n\theta+\psi_n(z)]$$

$$E_z = +\sum_{\nu=0}^{\infty} e_{0\nu}(z) r^{2\nu} + \sum_{n=1}^{\infty}\sum_{\nu=0}^{\infty} \frac{1}{n}\frac{de_{n\nu}(z)}{dz} \cos[n\theta+\psi_n(z)] - \sum_{n=1}^{\infty}\sum_{\nu=0}^{\infty} \frac{1}{n}\frac{d\psi_n(z)}{dz} e_{n\nu}(z) r^{n+2\nu} \sin[n\theta+\psi_n(z)]$$

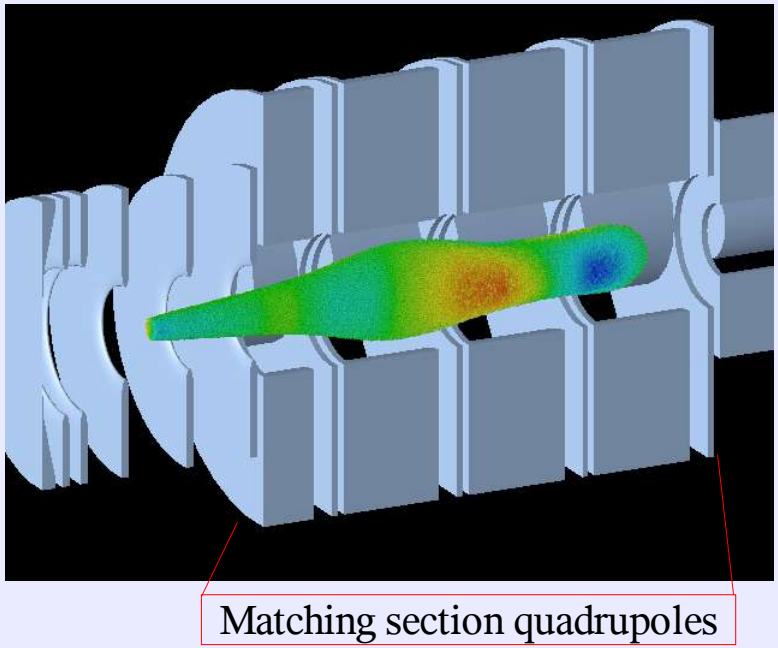
- Gridded elements
  - B field saved in 3 3-D grids
  - Potential in 1 3-D grid (E field calculated using finite difference)
- First principles: element included as boundary condition in field solution
  - Interdigitated electric quadrupoles
  - Round or elliptical pipes for magnetic quadrupoles and other elements

Lattice examples

Small ring at MSU
(Thanks to Felix Marti)



IBX simulation
Low energy, high current, short pulse accelerator



Python is the user interface

- Scripting – full language support for user-programmable code control
- Interactivity – code control and steering
- Visualization – in serial and parallel environment
- Remote steering and visualization – from a remote machine that can be attached to/detached from an already running job

Input deck becomes the “main” program

Interface between python and Fortran automatically generated based on variable description files

User interface is both complex and simple

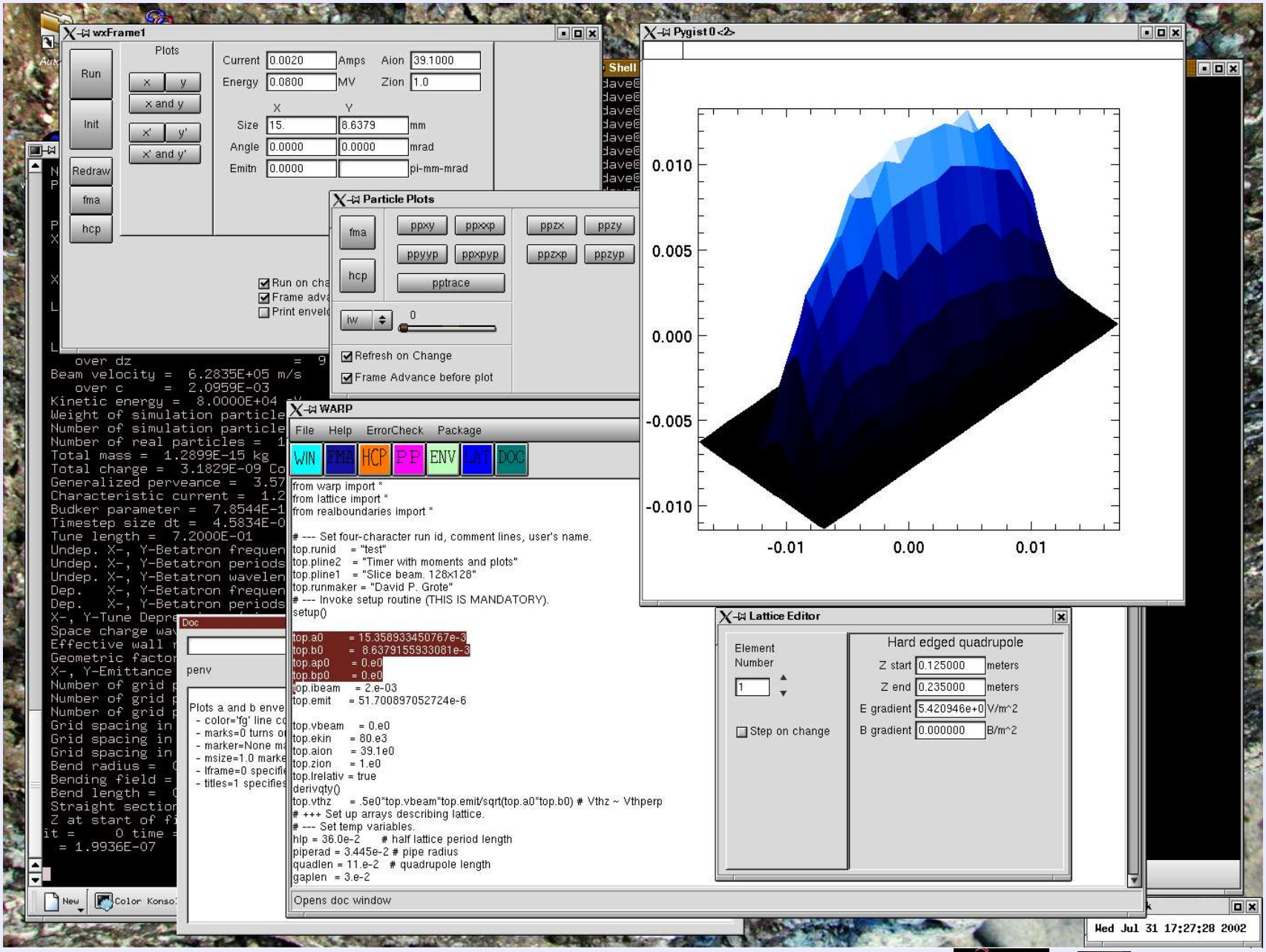
- “guts” of the code directly accessible from python (raw database and routines)
- Python code that provides high level interface, hiding the details of the “guts”

Can take advantage of other Python modules:

- Mathmatical libraries, Web server, GUI

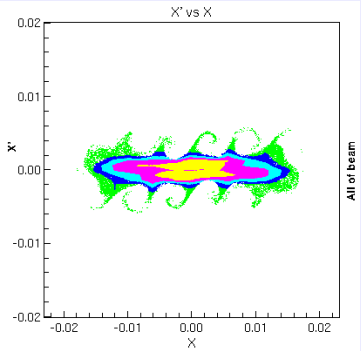
See www.python.com for more info

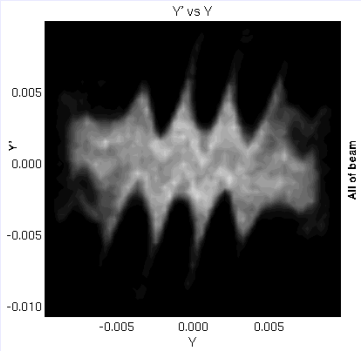
GUI under development



Endless diagnostics available

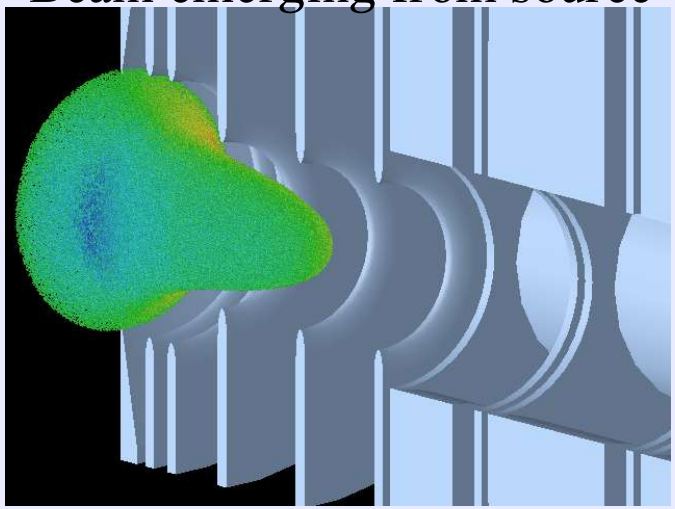
- All 2nd order moments, as function of z
- Time history of any of the moments
- Plots of all phase space projections, in various formats
- Field contours (plus internal boundaries)
- Slit-scan emulation
- 3-d visualization of boundaries, fields, particles
- Anything (almost) can be, queried, plotted, or written to file for postprocessing



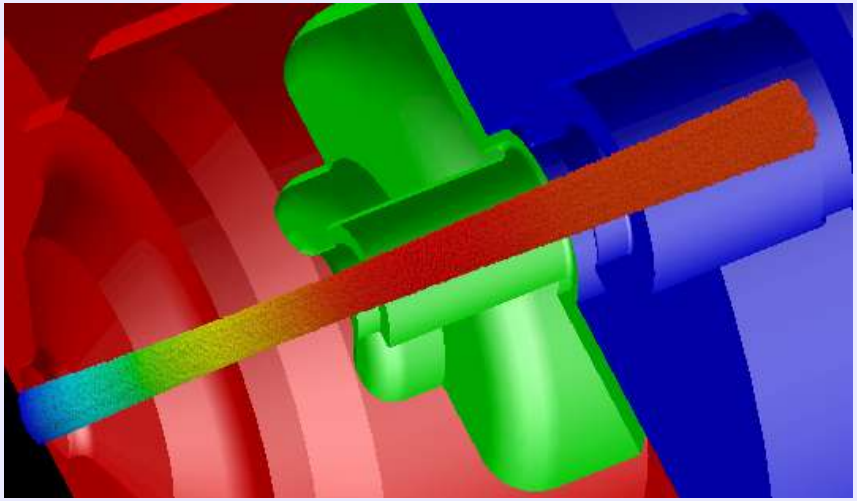


Visualization tools in development

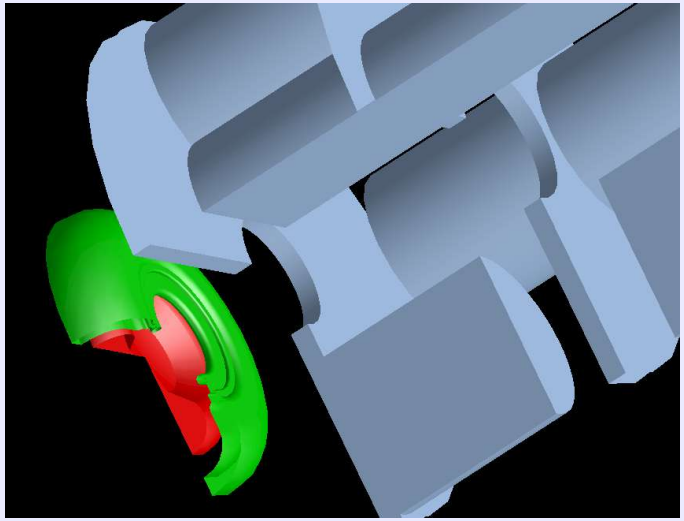
IBX simulation
Beam emerging from source

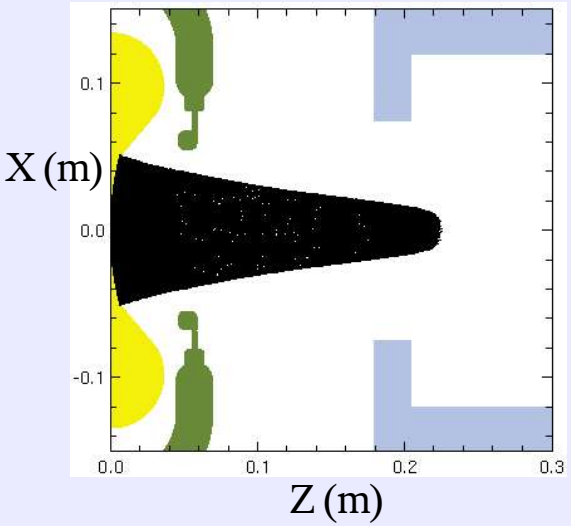


Venus source



ESQ injector





Parallel Processing

Basic features:

- Uses MPI
- Domain decomposition along z-axis
- Field domains and particle domains do not need to overlay
- Serial and parallel source identical except for parallel data handlers
- Maintains same python interface as serial code

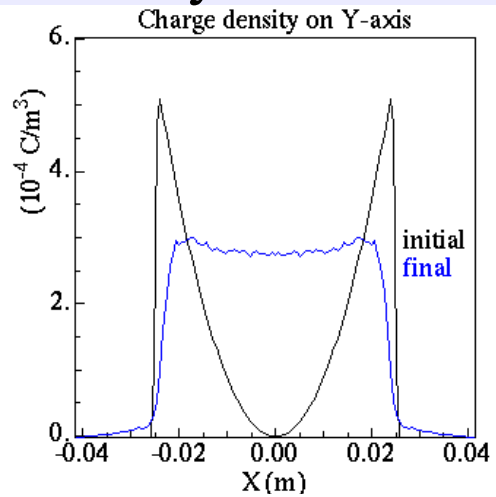
Nearly linear performance scale-up found for up to 256 processors

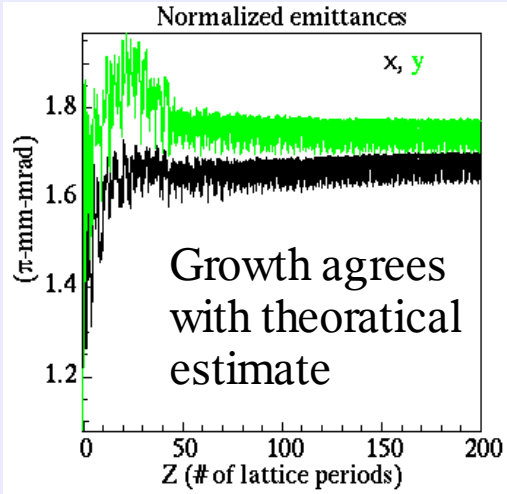
Expect good scaling for larger numbers for largest HIF applications

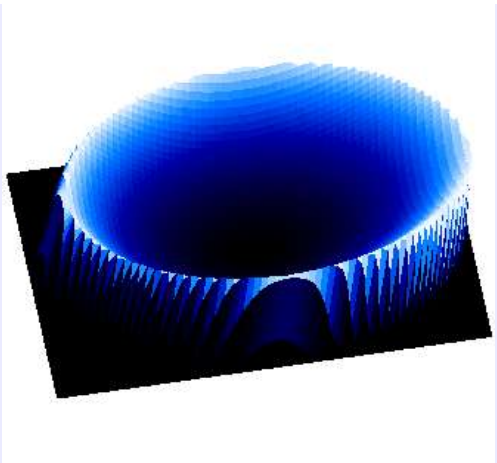
One last example: hollow beam with WARPxy

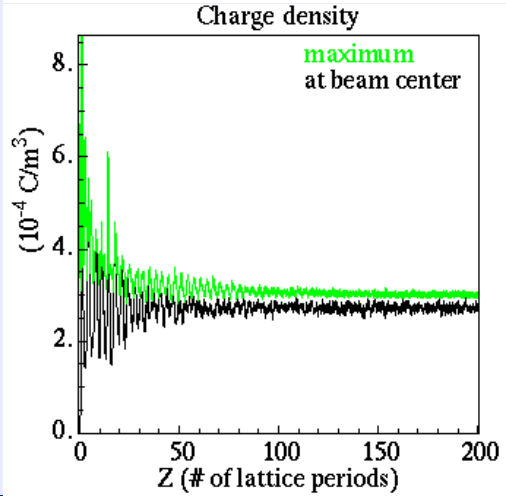
Transport a initially hollow beam in a FODO lattice.

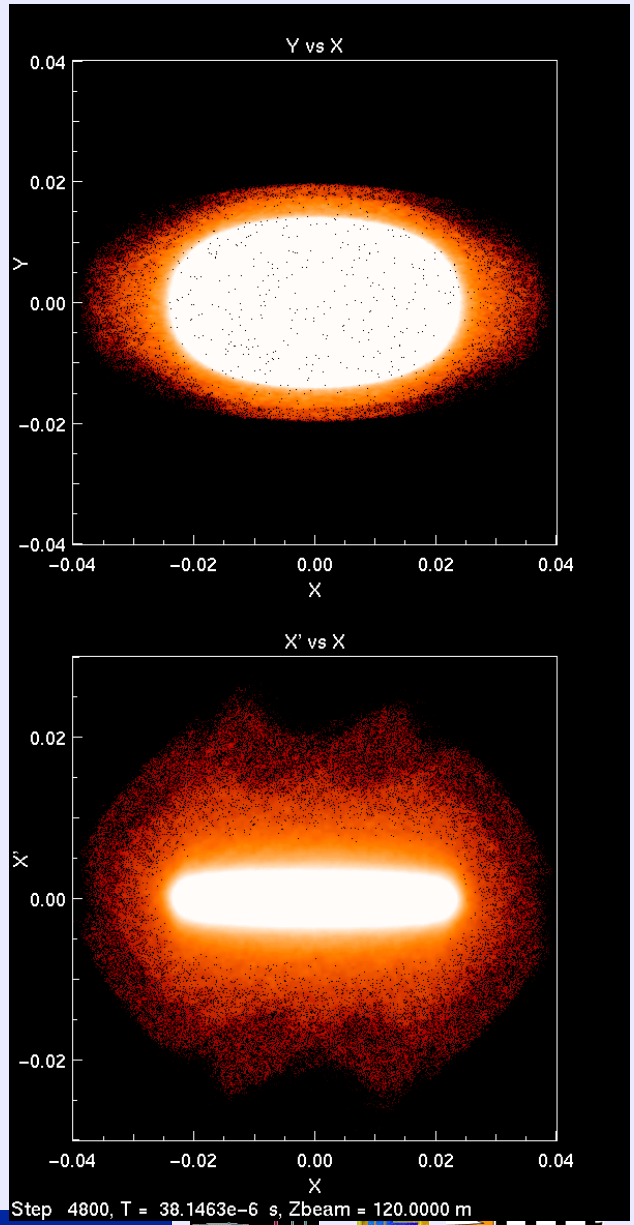
- Beam parameters: 2 MeV, 1 Amp
- Tune depression 70°- 10°
- Density vanishes at beam center









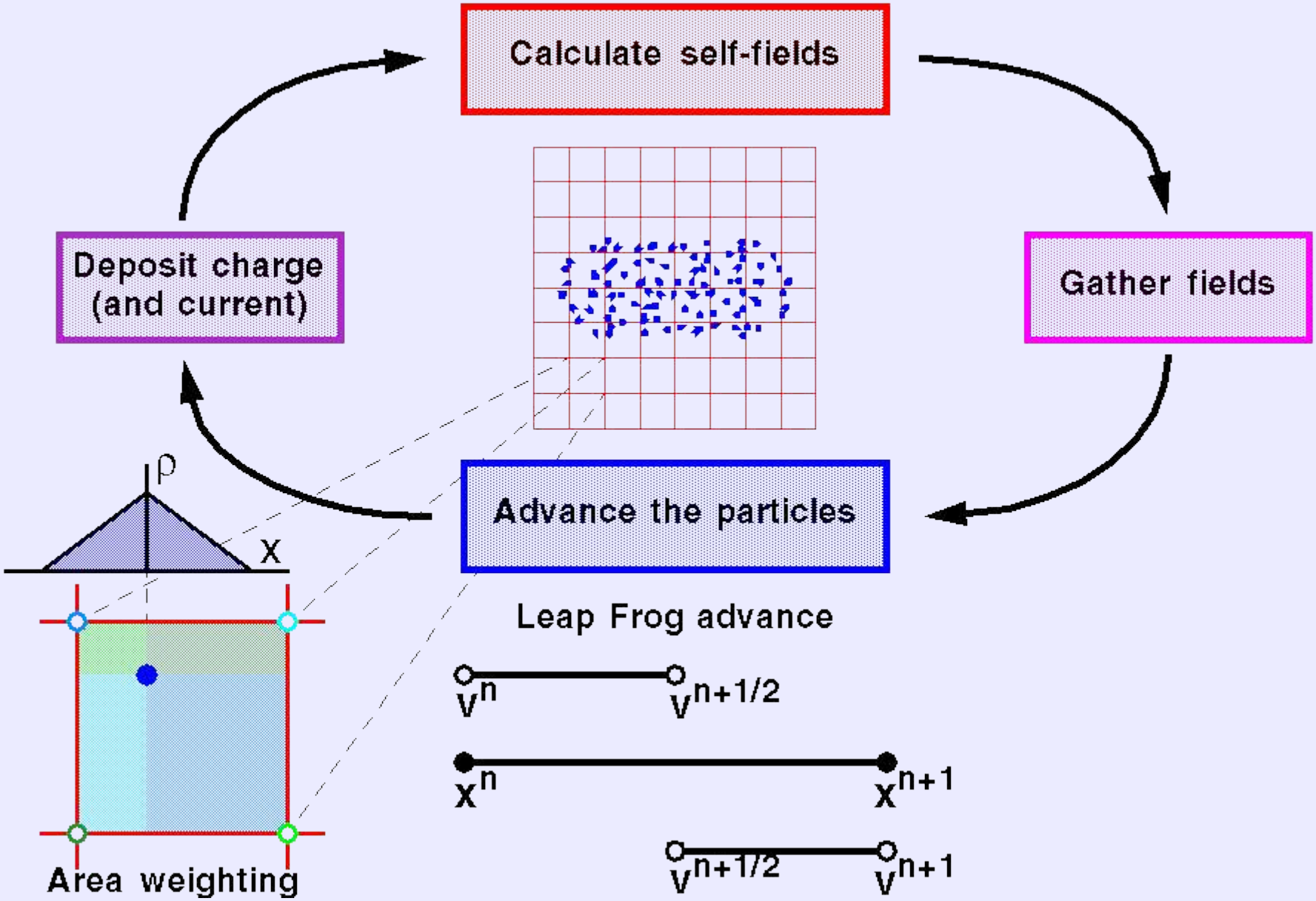


Conclusion

- Warp was originally developed for HIF
  - High current, space-charge dominated beams
  - Time dependence
  - Complex boundary conditions
- Should work well for beam transport for ECR sources
  - Can handle multiple species
  - Applied magnetic fields, bends
  - Plasma source model in development
- Contacts: DPGrote@lbl.gov
  or Alex Friedman AFriedman@lbl.gov

extra slides

PIC method


Calculate self-fields
Gather fields
Advance the particles
Deposit charge (and current)
ρ
x
Area weighting
Leap Frog advance
v^n
$v^{n+1/2}$
x^n
x^{n+1}
$v^{n+1/2}$
v^{n+1}