An Arbitrary Curvilinear Coordinate Particle-in-Cell Code

Chris Fichtl and John Finn
LANL

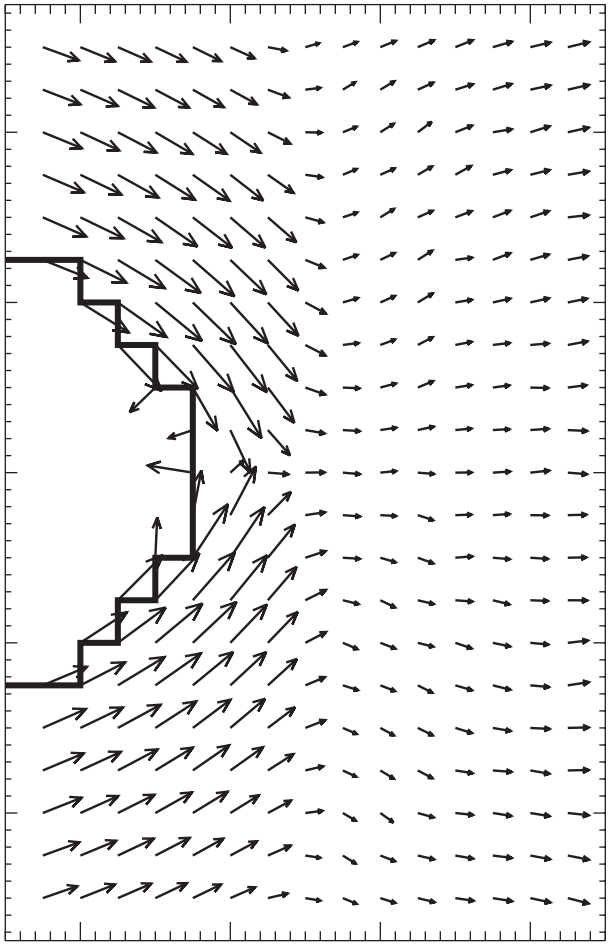
Keith Cartwright
SNL

Outline

- Motivation
- Development of ACC-PIC Components
  - PDE-Based Grid Generation
  - Curvilinear Coordinate Poisson Solver
  - Weighting and Interpolation
  - Logical Domain Particle Equations of Motion
  - Modified Leapfrog Integrator
- 2d ACC-PIC Code
- Conclusions and Future Work

Motivation

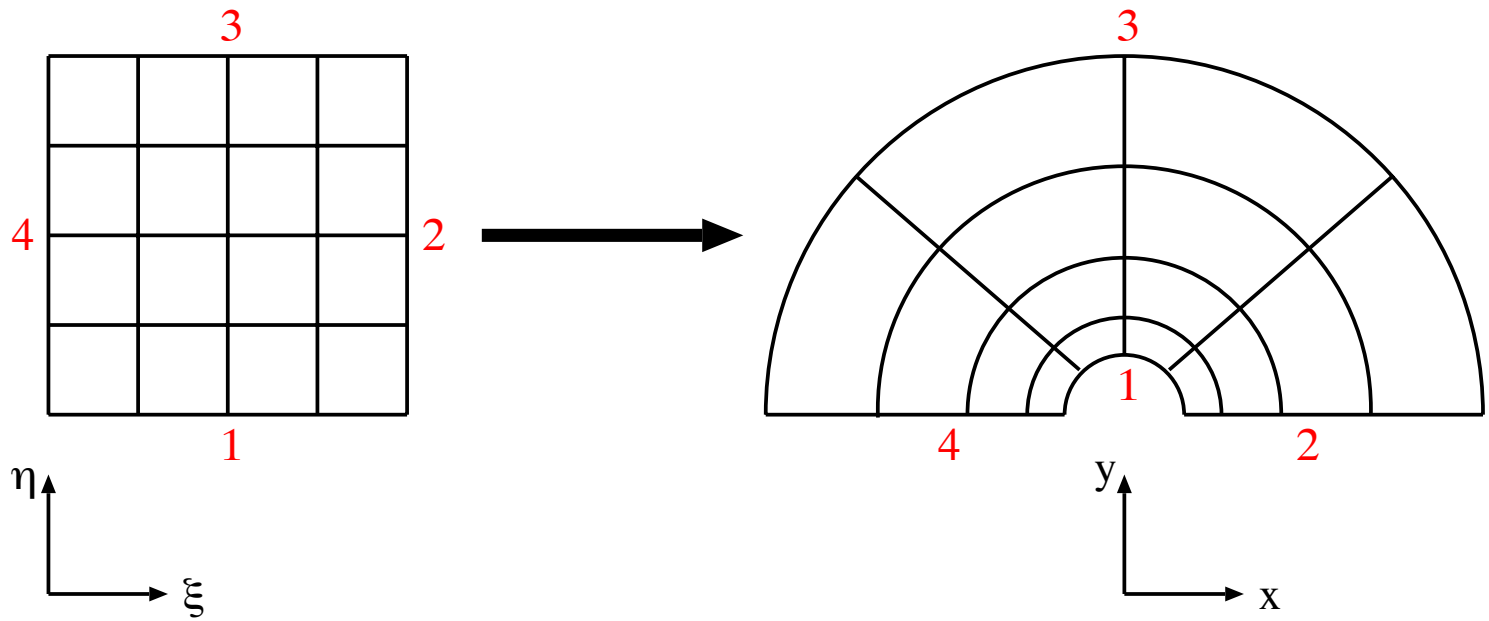
- Conventional PIC codes rely on uniform grid cells (polar, spherical)
- Specific coordinate system
- Multiple grid resolutions often required within domain
  - PIC processes are very complex on nonuniform domain (location/ weighting/interpolation)
  - Dynamically evolving domain/mesh
- Current solution:
  - More gridpoints…
  - More particles…
  - Bigger computers…



Miloch, et al., Phys. Rev. E, 2008

SOMETIMES A BIGGER HAMMER ISN'T THE RIGHT SOLUTION!

Mapping



- Mapping should be:
  - Smooth (at least continuous and differentiable)
  - Orthogonal (preferable but not necessary)
  - Well-adapted to the physical system

Differential Geometry Review

- Jacobian and its inverse:

$$j_{\alpha\beta}(\vec{\xi}) \equiv \frac{\partial x^\alpha}{\partial \xi^\beta} \qquad k^{\alpha\beta}(\vec{x}) \equiv \frac{\partial \xi^\alpha}{\partial x^\beta} \qquad j_{\alpha\beta} k^{\beta\gamma} \equiv \delta^\gamma_\alpha$$

- Metric Tensors:

$$g_{\alpha\beta}(\vec{\xi}) \equiv \mathbf{j}^T \mathbf{j} = \frac{\partial x^\gamma}{\partial \xi^\alpha} \frac{\partial x^\gamma}{\partial \xi^\beta}$$

$$g^{\alpha\beta}(\vec{x}) \equiv \mathbf{k}\mathbf{k}^T = \frac{\partial \xi^\alpha}{\partial x^\gamma} \frac{\partial \xi^\beta}{\partial x^\gamma}$$

- Can easily convert between covariant and contravariant in 2d:

$$g^{\alpha\beta} = (-1)^{\alpha+\beta} \frac{g_{3-\alpha,3-\beta}}{g_{\mathrm{cov}}}$$

$$g_{\alpha\beta} = (-1)^{\alpha+\beta} g_{\mathrm{cov}} \, g^{3-\alpha,3-\beta}$$

Elliptic Grid Generation

- Techniques using systems of PDE's to derive coordinate transforms are very popular for structured grid generation
  - Elliptic, parabolic, hyperbolic, etc
- Advantages:
  - Extremum principle
  - Smoothness of solution
  - 'one-to-one' mapping
- Disadvantages:
  - System of PDE's must be solved numerically (nonlinear solver)

Winslow's Method (JCP 1967)

- Uncoupled Laplace equation

$$\nabla_x^2 \xi^\alpha = \frac{\partial}{\partial x^\beta} \frac{\partial \xi^\alpha}{\partial x^\beta} = 0$$

- Transform to logical grid

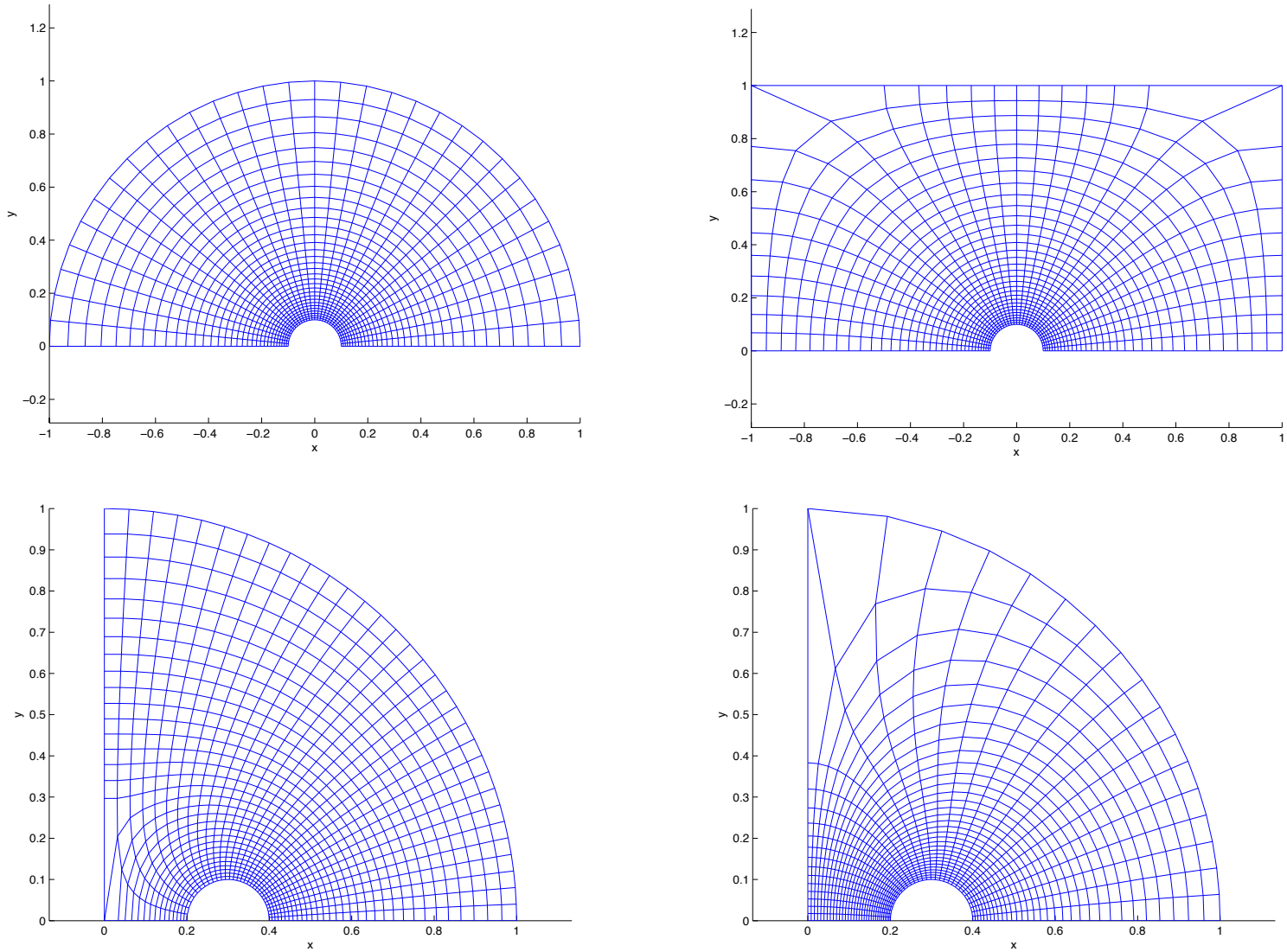
$$g^{\alpha\gamma}(\vec{x}\,) \frac{\partial^2 x^\mu}{\partial \xi^\alpha \xi^\gamma} = 0$$

- Invert metric tensor

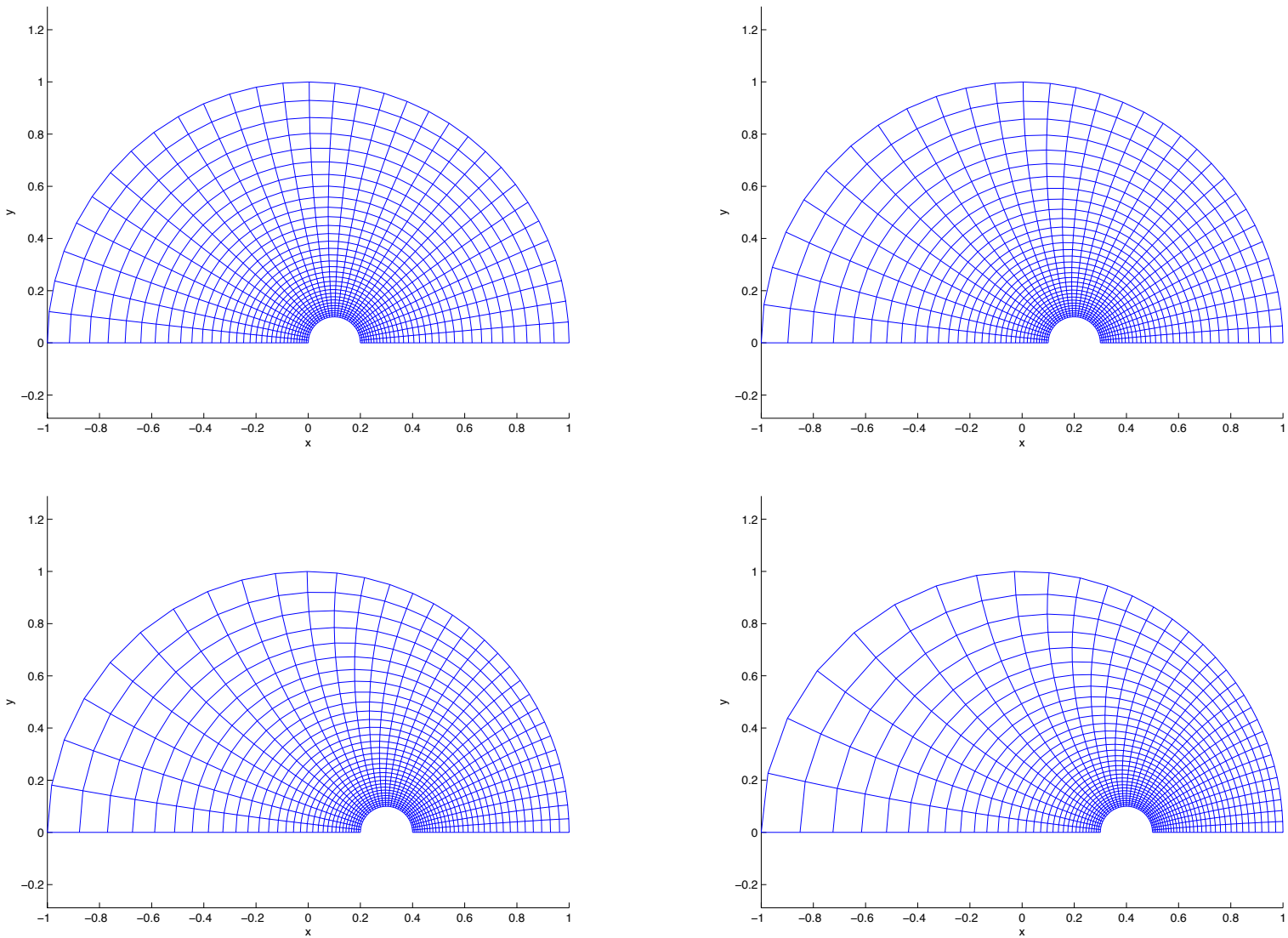
$$g_{22}\, x_{\xi\xi} - 2 g_{12}\, x_{\xi\eta} + g_{11}\, x_{\eta\eta} = 0$$
$$g_{22}\, y_{\xi\xi} - 2 g_{12}\, y_{\xi\eta} + g_{11}\, y_{\eta\eta} = 0$$

System of two nonlinear coupled equations –
Requires Newton-Krylov solver

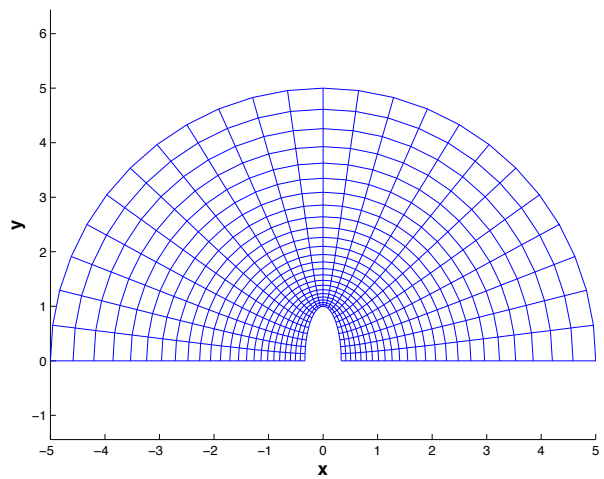
Example Grids

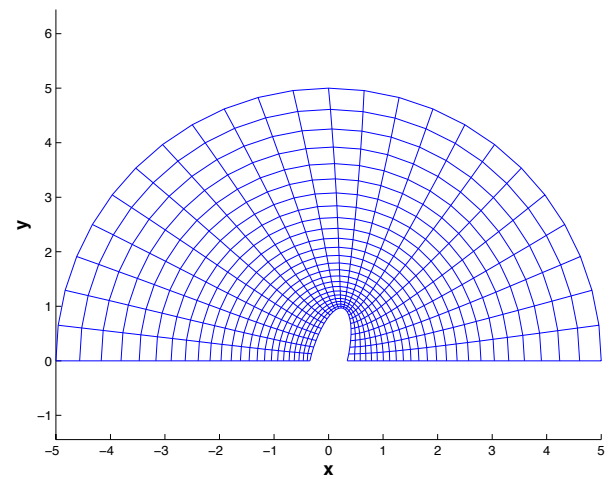


Example Grids



Example Grids





Curvilinear Coordinate Poisson Solver

- Transform $\nabla_x^2 \Phi = \rho(x)$ to logical grid:

$$\frac{\partial}{\partial \xi^\alpha}\left(f J g^{\alpha\beta} \frac{\partial \Phi}{\partial \xi^\beta} \right) = -f J \rho(x) = f \rho(\xi)$$

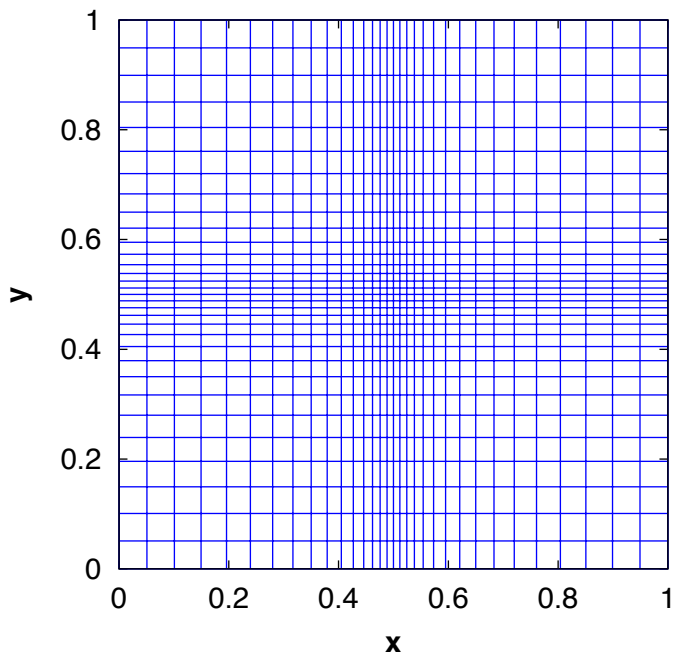
- $f = (1, r)$ is a 'geometry factor'
  - Selects azimuthal or axial symmetry

- Variational approach to discretization of Poisson equation
  - Symmetric
  - Positive definite
    - Solvability condition: $\int \rho d V = 0$
  - Solved using Jacobi-preconditioned CG

Physical potential is now mapped onto the logical grid

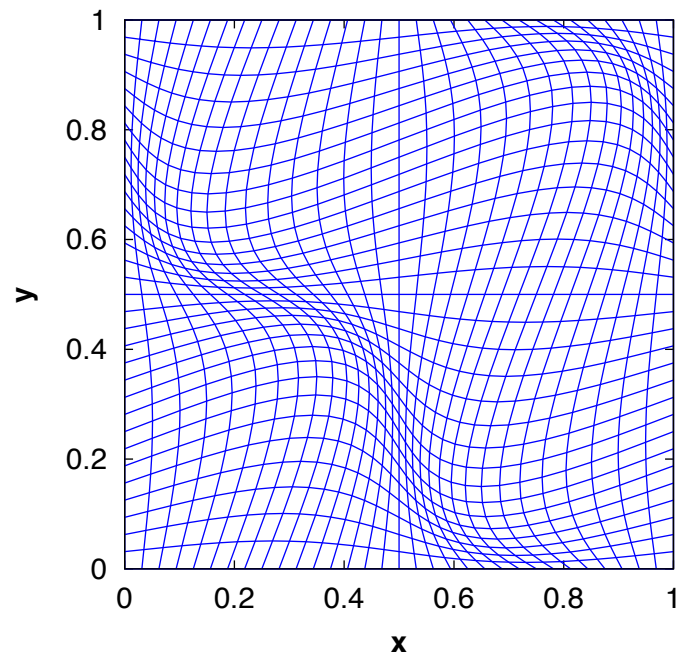
Grids for Validation

Orthogonal



$$x = \xi + \epsilon_g \sin(2\pi\xi)$$
$$y = \eta + \epsilon_g \sin(2\pi\eta)$$

Nonorthogonal



$$x = \xi + \epsilon_g \sin(2\pi\xi)\sin(2\pi\eta)$$
$$y = \eta + \epsilon_g \sin(2\pi\xi)\sin(2\pi\eta)$$

To avoid grid folding, $\varepsilon_g < \dfrac{1}{2\pi}$

Macroparticle Equations of Motion

- On physical grid, particles obey Newton-Lorentz eqns:

$$M\frac{d\vec{v}}{dt} = \vec{F} = Q\vec{E}$$

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

- No existing standard for PIC with nonuniform gridding
- Require iterative cell-search algorithm for physical space mover
- Logical choice: Can transform NL eqns directly to logical space…BUT…

Transformed NL eqns do not conserve phase-space area

Hamiltonian-Based Approach

- Apply contact transformation using an $F_2(\vec{x}, \vec{P}, t)$ generating function:

$$p^\alpha = \frac{\partial F_2}{\partial x^\alpha}$$

$$\xi^\alpha = \frac{\partial F_2}{\partial P^\alpha}$$

$$K(\vec{\xi}, \vec{P}, t) = H\left(\vec{x}(\vec{\xi}, \vec{P}, t), \vec{p}(\vec{\xi}, \vec{P})\right) + \frac{\partial F_2}{\partial t}$$

- Assume time-independent grid
  - Specialize to contact transformation $\vec{\xi} = \vec{\xi}(\vec{x})$

$$K = \frac{1}{2M}\left(g^{\alpha\beta} P^\alpha P^\beta\right) + V(\vec{\xi})$$

Hamiltonian-Based Approach

- Apply contact transformation using an $F_2(\vec{x}, \vec{P}, t)$ generating function:

$$p^\alpha = \frac{\partial F_2}{\partial x^\alpha}$$

$$\xi^\alpha = \frac{\partial F_2}{\partial P^\alpha}$$

$$K(\vec{\xi}, \vec{P}, t) = H\left(\vec{x}(\vec{\xi}, \vec{P}, t), \vec{p}(\vec{\xi}, \vec{P})\right) + \frac{\partial F_2}{\partial t}$$

- Assume time-independent grid
  - Specialize to contact transformation $\vec{\xi} = \vec{\xi}(\vec{x})$

$$K = \frac{1}{2M}\left(g^{\alpha\beta} P^\alpha P^\beta\right) + V(\vec{\xi}) = \underline{T(\vec{\xi}, \vec{P})} + V(\vec{\xi})$$

Non-separable!!!

Cannot use 'naïve' Leapfrog integration method

Equations of Motion

- Apply Hamilton's eqns:

$$\dot{\xi}^\alpha = \frac{\partial K}{\partial P^\alpha} \qquad \dot{P}^\alpha = -\frac{\partial K}{\partial \xi^\alpha}$$

- Resulting equations of motion:

$$\dot{\xi} = \frac{1}{M}\left(g^{11}P_\xi + g^{12}P_\eta\right)$$

$$\dot{\eta} = \frac{1}{M}\left(g^{12}P_\xi + g^{22}P_\eta\right)$$

$$\dot{P}_\xi = -\frac{1}{2M}\left(P_\xi^2\frac{\partial g^{11}}{\partial \xi} + 2P_\xi P_\eta\frac{\partial g^{12}}{\partial \xi} + P_\eta^2\frac{\partial g^{22}}{\partial \xi}\right) - \frac{\partial V(\vec{\xi})}{\partial \xi}$$

$$\dot{P}_\eta = -\frac{1}{2M}\left(P_\xi^2\frac{\partial g^{11}}{\partial \eta} + 2P_\xi P_\eta\frac{\partial g^{12}}{\partial \eta} + P_\eta^2\frac{\partial g^{22}}{\partial \eta}\right) - \frac{\partial V(\vec{\xi})}{\partial \eta}$$

Field *and* grid quantities must be interpolated

Modified Leapfrog Integrator

- Rewrite equations of motion as:

$$\dot{\vec{\xi}} = \vec{U} \text{ and } \dot{\vec{P}} = \vec{V}$$

where

$$\frac{\partial U^\alpha}{\partial \xi^\alpha} + \frac{\partial V^\alpha}{\partial P^\alpha} = 0$$

- ML can be written $\vec{M}_{\Delta t} = \vec{P}^e_{\Delta t} \circ \vec{\xi}^i_{\Delta t}$, where

$$\xi^i_{\Delta t} : \begin{cases} \vec{\xi}_1 = \vec{\xi} + \Delta t \vec{U}(\vec{\xi}_1, \vec{P}) \\ \vec{P}_1 = \vec{P} \end{cases}, \quad P^e_{\Delta t} : \begin{cases} \vec{\xi}' = \vec{\xi}_1 \\ \vec{P}' = \vec{P}_1 + \Delta t \vec{V}(\vec{\xi}_1, \vec{P}_1) \end{cases}$$

Combining,

$$\vec{\xi}' = \vec{\xi} + \Delta t\, \vec{U}(\vec{\xi}', \vec{P}), \quad \vec{P}' = \vec{P} + \Delta t\, \vec{V}(\vec{\xi}', \vec{P})$$

Modified Leapfrog Features

- Can be symmetrized for 2nd-order accuracy in Δt

$$\vec{\xi}^{e}_{\Delta t/2} \circ \vec{P}^{i}_{\Delta t/2} \circ \vec{P}^{e}_{\Delta t/2} \circ \vec{\xi}^{i}_{\Delta t/2}$$

- Semi-implicit →Newton/Picard iterations required for $\vec{\xi}^{\,i}, \vec{P}^{\,i}$

- Proven symplectic for *N* degrees of freedom

- Charge accumulation/field solve once per timestep
  - CN and Newmark-type methods require charge accumulation/field solve for *every* implicit iteration for *every* particle to retain accuracy!

Logical Grid Weighting/Interpolation

- Particle shape is uniform and symmetric on logical grid
- Utilizes adjoint operations
  - Weighting: $\rho(\xi_i) = \sum_p Q_p S(x_i - x_p)$
  - Interpolation: $E(x_p) = \sum_i E_i S(x_p - x_i)$
- Define $S_{ip} \equiv S(x_i - x_p)$ and $S_{pi} \equiv S(x_p - x_i) \quad \rightarrow \quad S_{pi} = S_{ip}^T$
- Implemented schemes:
  - Bilinear (S_1)
    - 4 node stencil
  - Biquadratic (S_2)
    - 9 node stencil

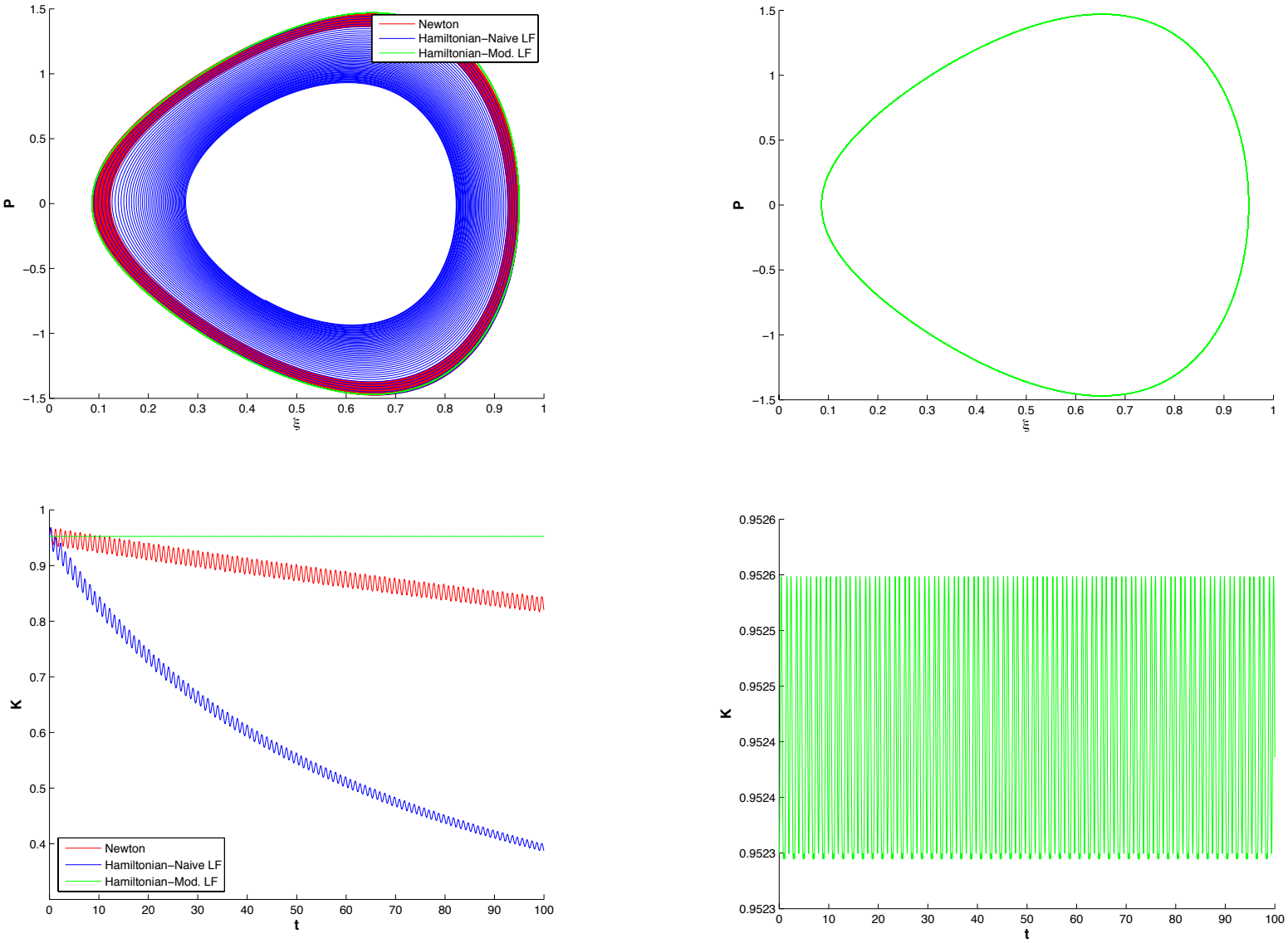
Self Forces

- Equations of motion require $E^{\xi}(\xi_p)$
  - Ideally: Interpolate E^{ξ}
  - Reality: This DOES NOT produce zero self-fields at particle!

- Our method:
  - Interpolate E^{x}
  - Interpolate J
  - Multiply $\rightarrow E^{\xi}(\xi_p) = E^{x}(\xi_p)J(\xi_p)$

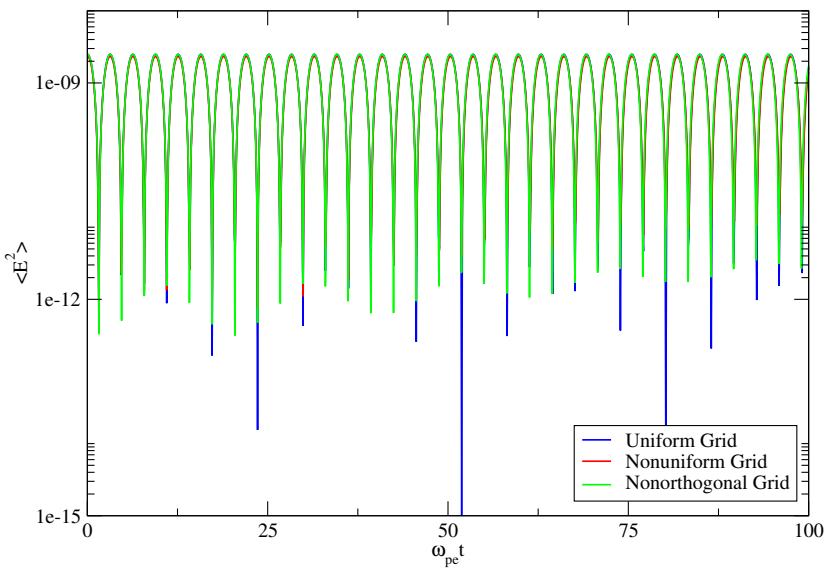
- This method works exactly in 1d, but does not exactly cancel self-forces in 2d.

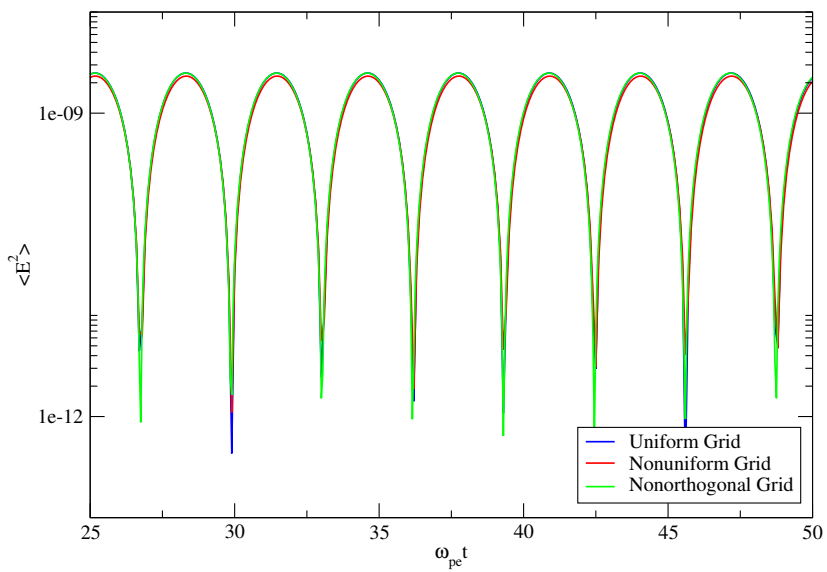
Grid-free ML Test



Sample Data

Cold plasma oscillations on a square domain

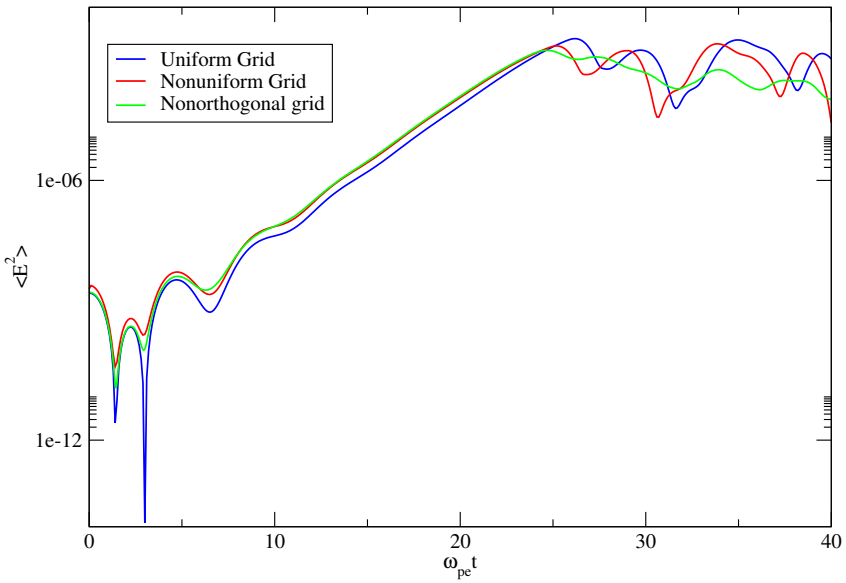


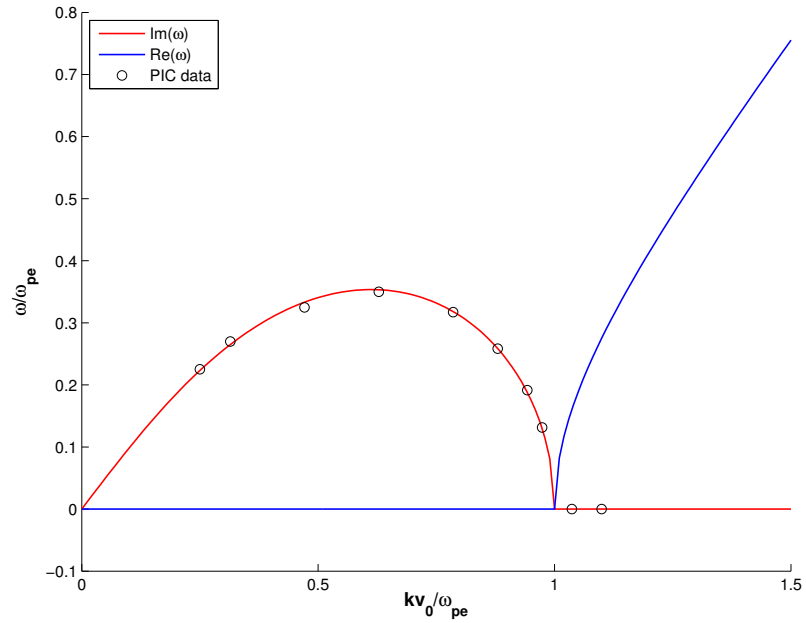


<u>Initial Conditions</u>: Quadratic (S2) particle shape functions with $N_\xi = N_\eta = 128$, $\langle N_{ppc} \rangle = 225$, $\Delta t = 0.025$, and $\varepsilon_{grid} = 0.1$.

Sample Data

Two-stream instability on a square domain

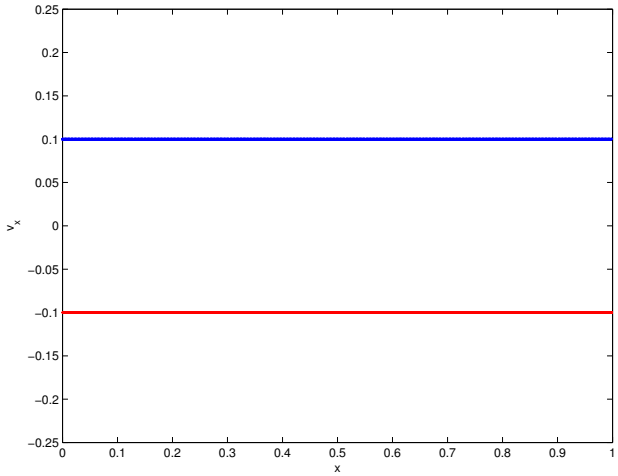




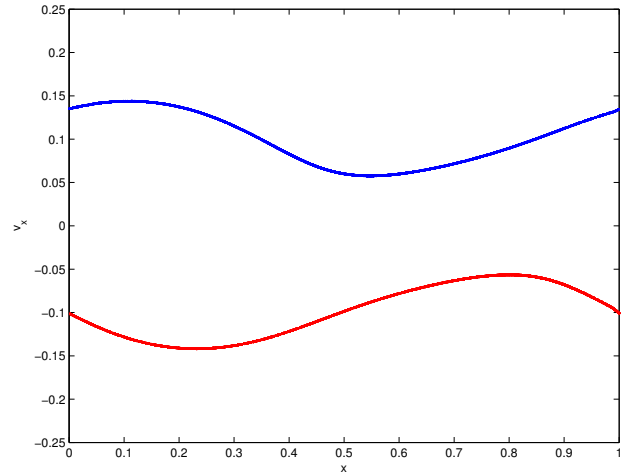
<u>Initial Conditions</u>: Quadratic (S2) particle shape functions with $N_\xi = N_\eta = 128$, $\langle N_{ppc} \rangle = 225$, $\Delta t = 0.025$, $\varepsilon_{grid} = 0.1$, and $v = [0.314, 0.314]$.

Sample Data

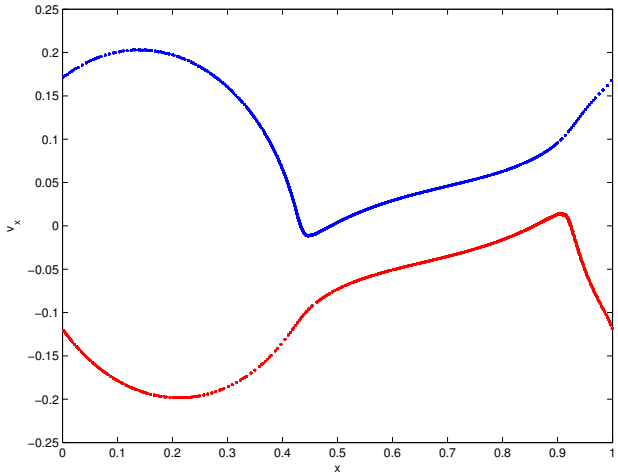
Two-stream instability on a square domain



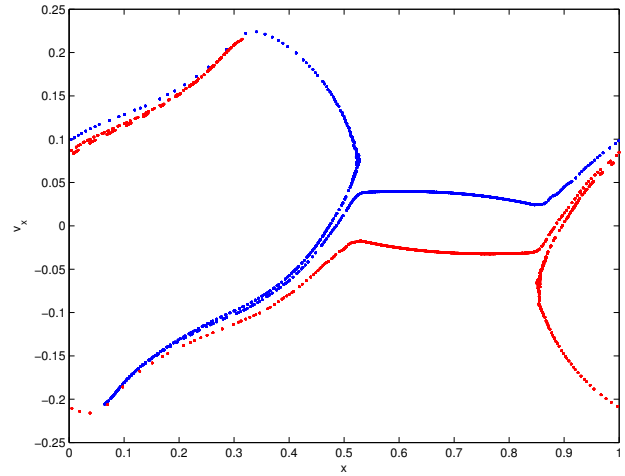
$\omega_{pe}t = 0$



$\omega_{pe}t = 22$



$\omega_{pe}t = 25$

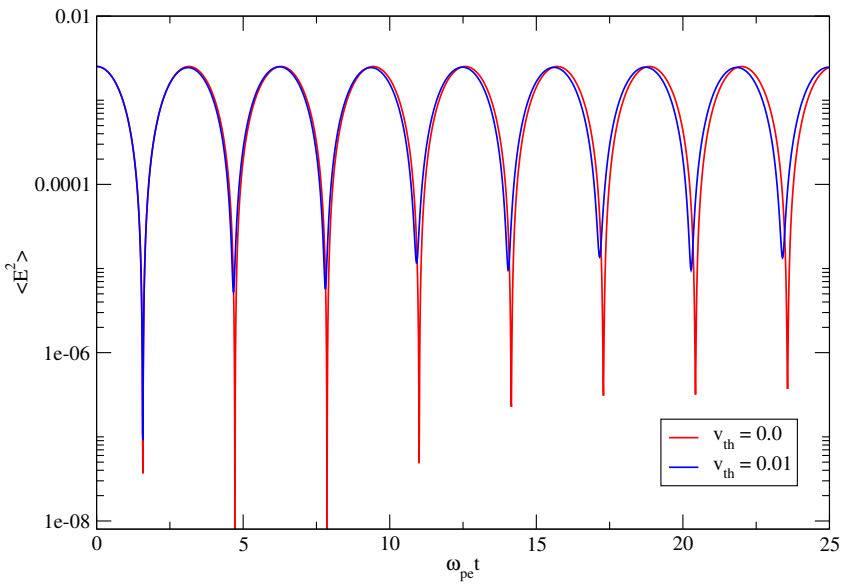


$\omega_{pe}t = 28$

Sample Data

Langmuir waves on a square domain

$$\omega^2 = \omega_{pe}^2 + 3k^2 v_{th}^2$$



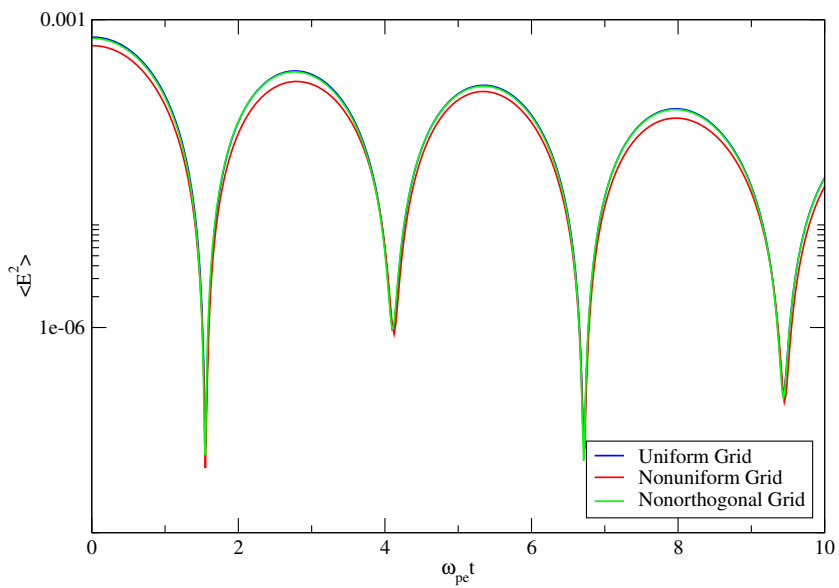
v_{th}	ω_{PIC}	ω_{theory}	% error
0	1.00081	1	0.081
0.005	1.00081	1.00148	0.067
0.01	1.00644	1.0059	0.053
0.015	1.0166	1.01324	0.332
0.02	1.02601	1.02341	0.254
0.025	1.02887	1.03635	0.722

<u>Initial Conditions</u>: Quadratic (S2) particle shape functions with $N_\xi = N_\eta = 128$, $\langle N_{ppc} \rangle = 225$, $\Delta t = 0.01$, $\varepsilon_{grid} = 0.15$

Sample Data

Landau damping on a square domain

$$\omega_i = -\sqrt{\frac{\pi}{8}}\frac{\omega_{pe}}{|k^3\lambda_{De}^3|}\exp\left[-\left(\frac{1}{2k^2\lambda_{De}^2}+\frac{3}{2}\right)\right]$$

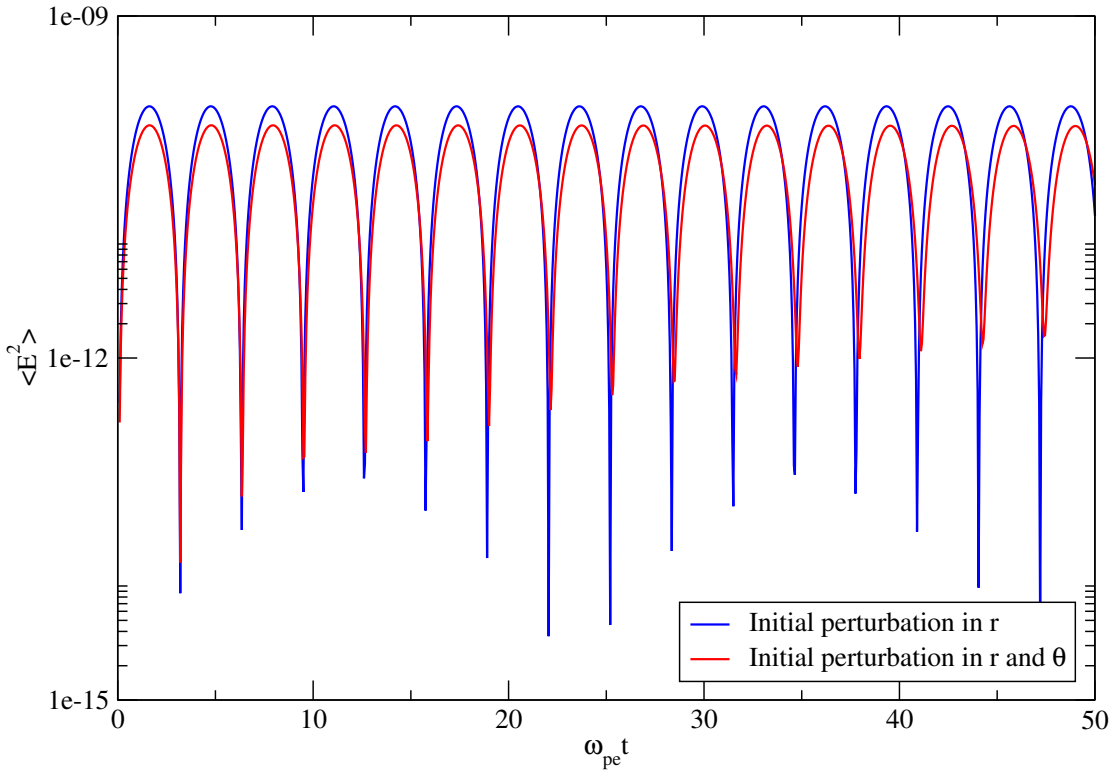


v_{th}	γ_{PIC}	γ_{theory}	% error
0.06	-0.0732	-0.0774	5.426
0.08	-0.1453	-0.1521	4.471
0.10	-0.1504	-0.1589	5.349
0.12	-0.1311	-0.1354	3.176

<u>Initial Conditions</u>: Quadratic (S2) particle shape functions with $N_\xi = N_\eta = 128$, $\langle N_{ppc}\rangle = 400$, $\Delta t = 0.025$, $\varepsilon_{grid} = 0.15$, $v_{th} = 0.07$

Sample Data

Cold plasma oscillations on an annulus

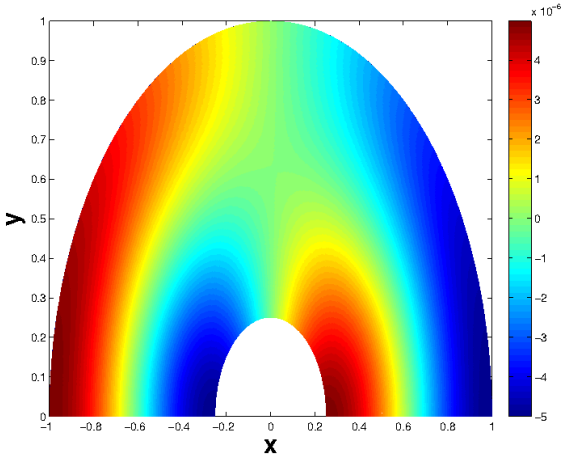


<u>Initial Conditions</u>: Quadratic (S2) particle shape functions with

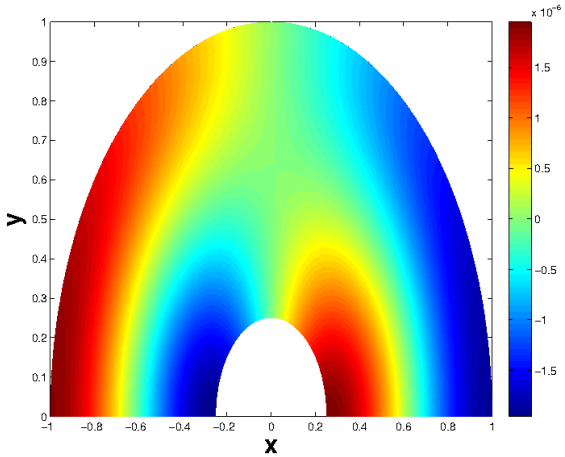
$N_\xi = N_\eta = 64$, $\langle N_{ppc} \rangle = 400$, $\Delta t = 0.05$, $r_2/r_1 = 4$

Sample Data

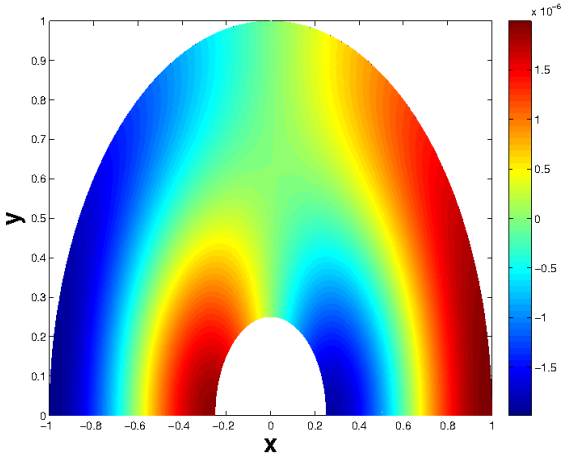
Cold ES plasma oscillation on annulus



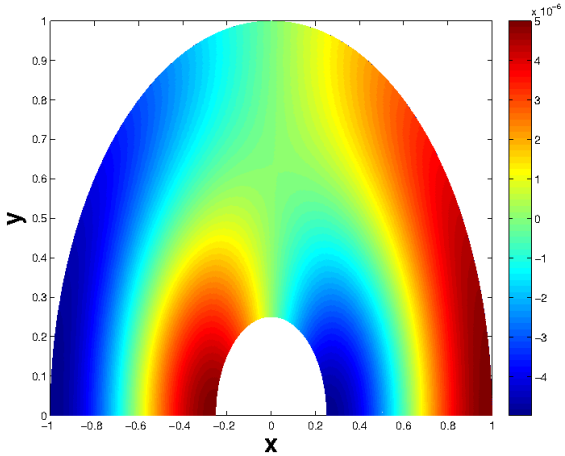
$\omega_{pe}t = 1.6$



$\omega_{pe}t = 2.8$



$\omega_{pe}t = 3.6$



$\omega_{pe}t = 4.8$

Conclusions

- ACC-PIC method was developed to operate entirely on logical domain
  - Winslow's method of grid generation
  - Curvilinear coordinate formulation of Poisson equation
  - Logical grid equations of motion
  - Modified Leapfrog integrator
  - Weighting/Interpolation issues

- Implemented in 1d and 2d

- Future work:
  - Add time dependent grid
  - Add particle management strategy
  - Extend to 3d
  - Include full EM physics