Discovery, characterization, and mitigation of a numerical instability associated with gyrokinetic δf-PIC algorithms

G. Wilkie , W. Dorland

University of Maryland

21 March 2013

6th Gyrokinetics Working Group Meeting
Vienna, 2013

Motivation

Why Particle-In-Cell?

- Physically intuitive
- Scales well to at least thousands of processors
- Extending the algorithm to include additional physics (high energy particles, rotation, etc.) is relatively simple

Outline

The Gyrokinetic Equation

$$\frac{\partial g}{\partial t} + \left(v_{\parallel}\mathbf{b} + \mathbf{v_d} + \frac{c}{B_0}\mathbf{b} \times \nabla \langle \phi \rangle_{\mathbf{R}} \right) \cdot \nabla g - C_{GK}\left[g\right]$$
$$= -\frac{c}{B_0}\left(\mathbf{b} \times \nabla \langle \phi \rangle_{\mathbf{R}}\right) \cdot \nabla F_0 - \frac{q}{T}\mathbf{v_d} \cdot \nabla \left(\langle \phi \rangle_{\mathbf{R}} F_0 \right) - \frac{q}{T} F_0 v_{\parallel} \mathbf{b} \cdot \nabla \langle \phi \rangle_{\mathbf{R}}$$

Frieman and Chen, 1982

Where:

- $g \equiv \langle \delta f \rangle_{\mathbf{R}} \qquad \left(w = \frac{g}{F_0} \right)$
- $F_0 = \frac{n_0}{(2\pi)^{3/2} v_t^3} \exp\left[-\frac{v^2}{2v_t^2} \right]$
- $\mathbf{v_d} = \frac{c}{qB}\frac{m v_{\parallel}^2}{R_c}\hat{\mathbf{r}} \times \mathbf{B} - \frac{c}{qB}\mu \nabla \mathbf{B} \times \mathbf{B}$
- $\langle \rangle_{\mathbf{R}}$ is the gyroaverage at constant gyrocenter $\mathbf{R}$
- $C_{GK}\left[g\right]$ is the gyroaveraged collision operator
- Velocity-space coordinates are $E = \frac{1}{2}mv^2$ and $\mu = \frac{m v_{\perp}^2}{2B_0}$

Use the method of characteristics to solve GK equation

Electrostatic, slab, collisionless:

$$\frac{Dg}{Dt} = \frac{\partial g}{\partial t} + \frac{dz}{dt}\frac{\partial g}{\partial z} + \frac{d\mathbf{R}_\perp}{dt} \cdot \frac{\partial g}{\partial \mathbf{R}_\perp}$$
$$= -\frac{c}{B_0}\mathbf{b} \times \nabla \langle \phi \rangle_{\mathbf{R}} \cdot \nabla F_0 - \frac{q}{T}\mathbf{v_d} \cdot \nabla \left(\langle \phi \rangle_{\mathbf{R}} F_0 \right) - \frac{q}{T} F_0 v_\parallel \mathbf{b} \cdot \nabla \langle \phi \rangle_{\mathbf{R}}$$

$\frac{D}{Dt}$ is the convective derivative, so a 5D partial differential equation becomes an ordinary differential equation *along characteristic curves* defined by:

- $\frac{d\mathbf{R}_\perp}{dt} = \mathbf{v_d} + \frac{c}{B_0}\mathbf{b} \times \nabla \langle \phi \rangle_{\mathbf{R}}$
- $\frac{dz}{dt} = v_\parallel = \pm\sqrt{\frac{2}{m}}\sqrt{E - \mu B}$
- $\frac{dE}{dt} = 0$
- $\frac{d\mu}{dt} = 0$

Nonlinearity in the gyrokinetic equation prevents us from knowing characteristic curves a priori

Fields must be calculated as the simulation progresses. Two possible approaches:

- **Particle-to-Particle** (P2P)
  - For each particle: sum over contribution from all others
  - Requires $\mathcal{O}\left(N_p^2\right)$ operations
- **Particle-in-Cell** (PIC)
  - Fields are calculated locally on a grid, then reinterpolated back to the particles
  - Requires $\mathcal{O}\left(N_p\right) + \mathcal{O}\left(N_g \ln N_g\right)$ operations

Full-f PIC versus δf PIC

- **Full-f PIC**
  - Models the full distribution function as a relative density of total number of particles
  - Usually, the RHS of the kinetic equation is zero (collision operator notwithstanding), so no particle "weight" is necessary.
  - Requires a large number of particles to resolve small perturbations in distribution function F_0
- δf **PIC**
  - Includes a source term along characteristics
  - Each particle carries a "weight": its single sample of the perturbed distribution δf (in our case, $w \equiv \frac{\langle \delta f \rangle_{\mathbf{R}}}{F_0}$)
  - Particles only need to resolve the perturbation: the background distribution is solved exactly and does not contribute to the error

ROMA Overview

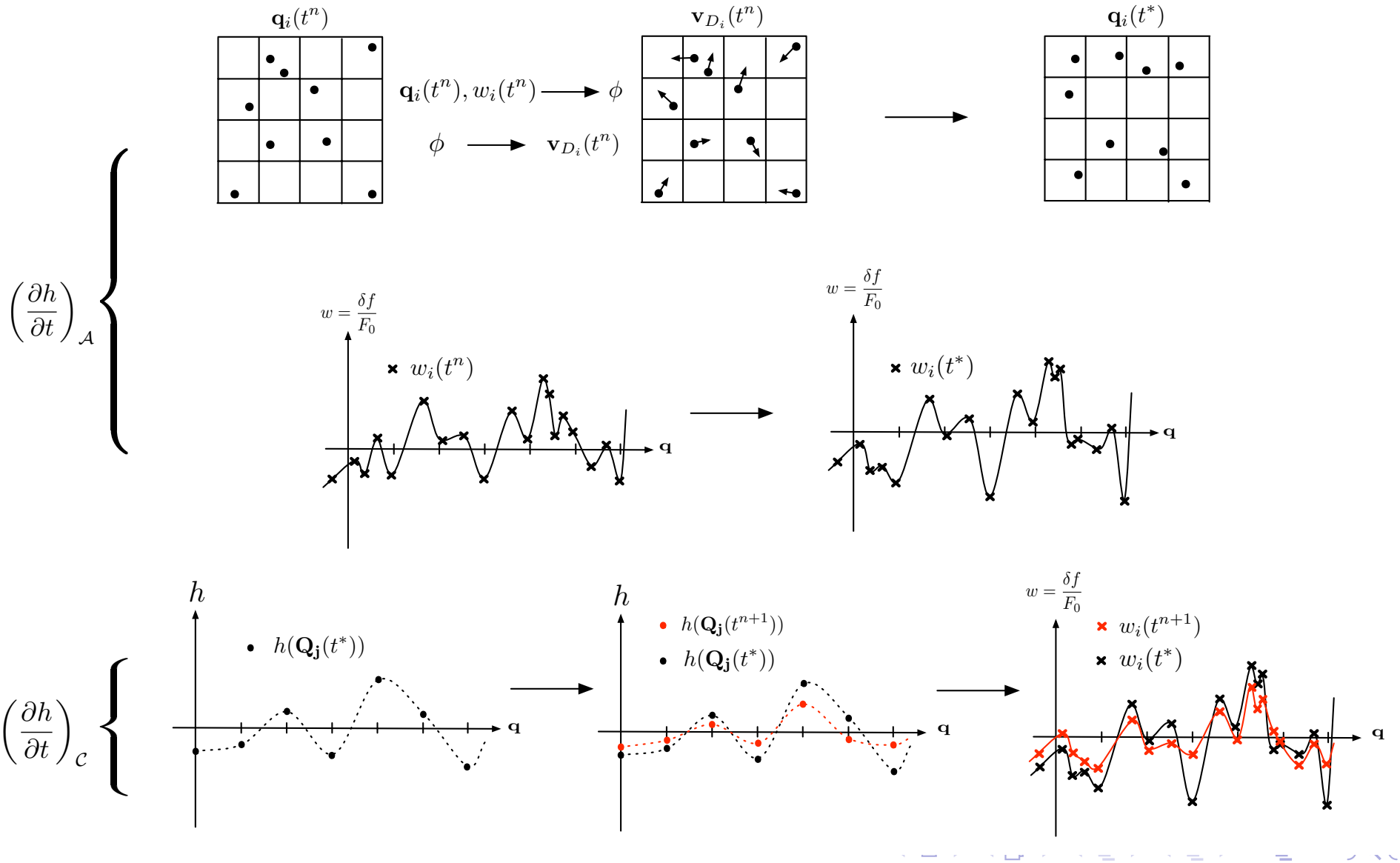
- A δf particle-in-cell code for solving the nonlinear gyrokinetic equation
- Evolved from GSP, written by Ingmar Broemstrup in 2008

Capabilities:

- Electrostatic turbulence
- Three spatial dimensions
- Particle trapping
- Parallelized
- Collision operator
- (Electromagnetic Fluctuations)

Broemstrup, 2008

Diagram of the Algorithm



Outline

Outline

Algorithm admits an undocumented, nonphysical instability when run with nonzero k_z

- Growth rate can be very large for high k_z and/or low $k_\perp$
- Convergent on spatial resolution and time step
- Existence confirmed by at least one other research group

Growth rate is mitigated and/or eliminated with:

- Piecewise-constant fields
- Coarse-graining "collision operator"

The fundamental nature of the instability is not yet fully understood

Simplify the problem to the simplest case where the instability exists

- No FLR, μ-averaged
- Electrostatic
- Linear
- Uniform B_0 and F_0

$$\frac{\partial g_i}{\partial t} + v\frac{\partial g_i}{\partial z} = -vF_0\frac{\partial \phi}{\partial z}$$

$$\sqrt{\frac{m_i}{m_e}}\frac{\partial g_e}{\partial t} + v\frac{\partial g_e}{\partial z} = vF_0\frac{\partial \phi}{\partial z}$$

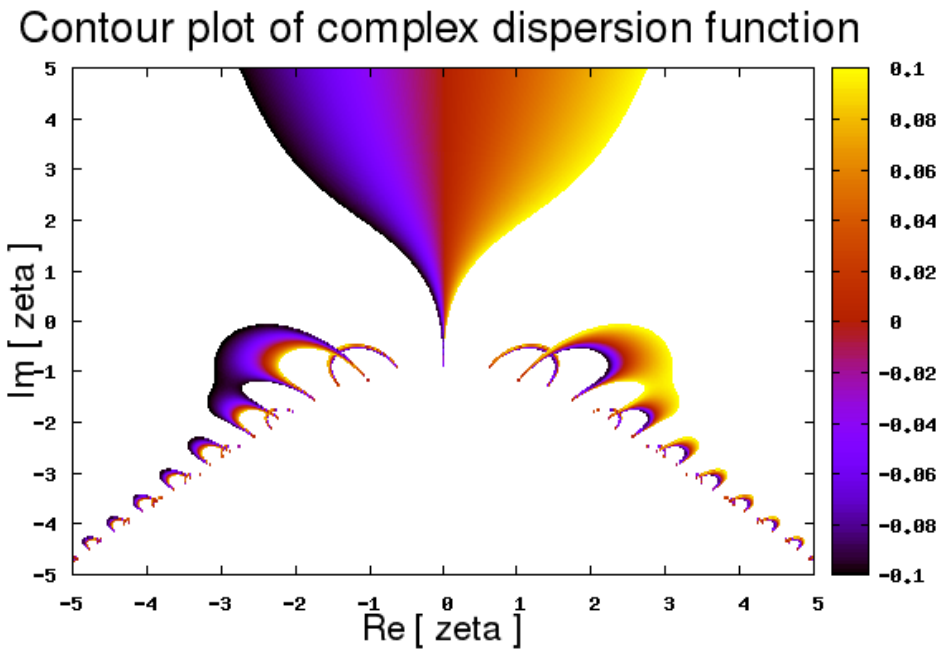
$$-\nabla_\perp^2\phi = \int g_i dv - \int g_e dv$$

This set of equations yields a nontrivial dispersion relation

Electrostatic Alfvén Wave

(or: Ω_H *mode, Inertial Alfvén Wave*)

$$2k_\perp^2\rho_i^2 - Z'\left(\frac{\omega}{\sqrt{2}k_\parallel v_{t,i}}\right) - Z'\left(\frac{\omega}{\sqrt{2}k_\parallel v_{t,e}}\right) = 0$$



Excplicit schemes must resolve the phase velocity of these fast waves

In the high-frequency limit, the wave is stable with:

$$\omega \approx \omega_0 - i\sqrt{\frac{\pi}{2}}\frac{\delta^3\omega_0^3}{2k_\parallel^2}e^{-\frac{\delta^2\omega^2}{2k_\parallel^2}}$$

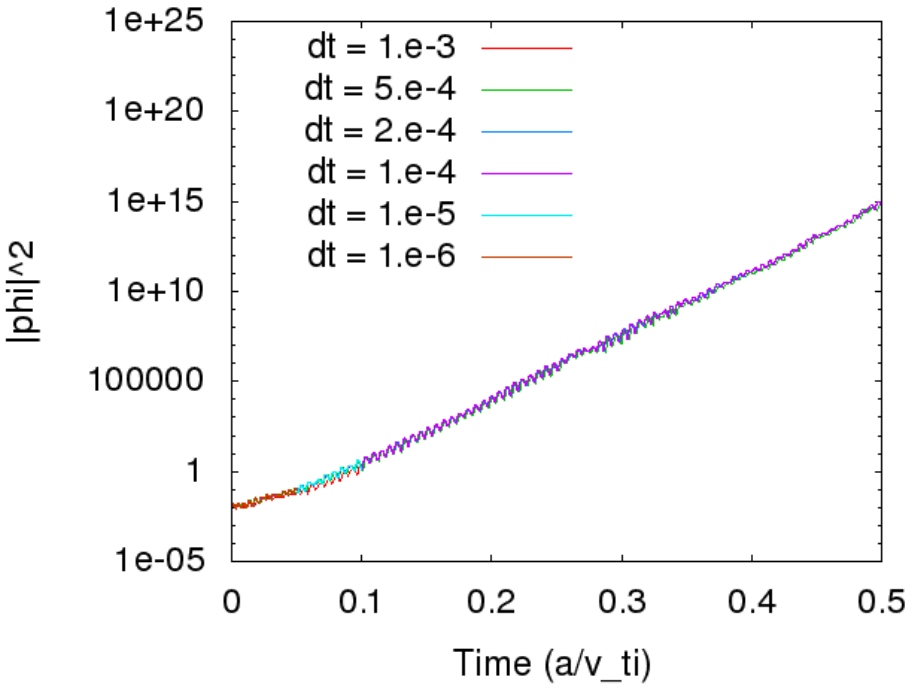
Where $\delta \equiv \sqrt{\frac{m_e}{m_i}}$ and:

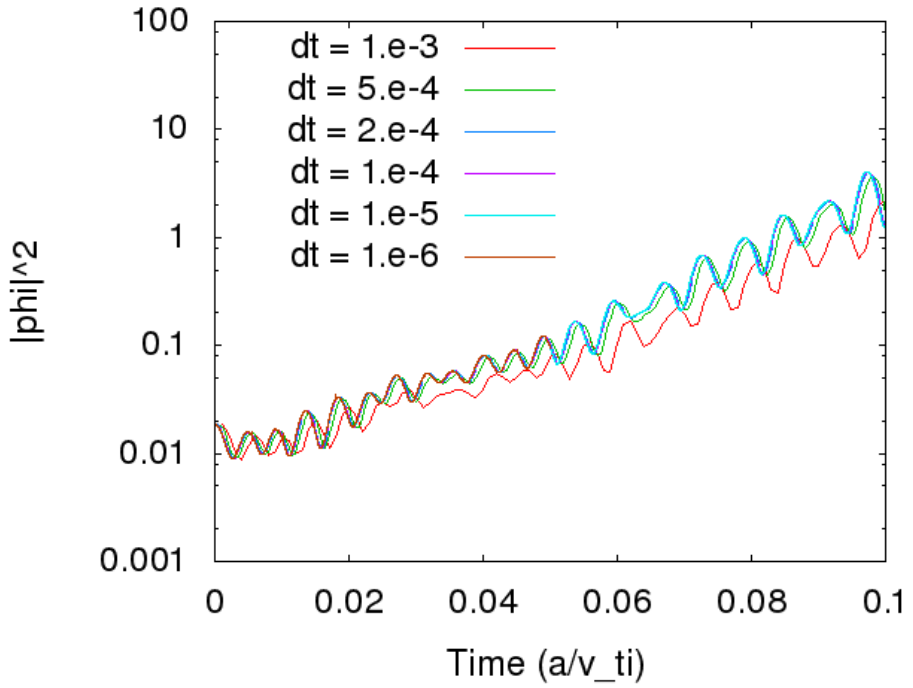
$$\omega_0 = \frac{k_\parallel}{\delta k_\perp}$$

Lee, 1987

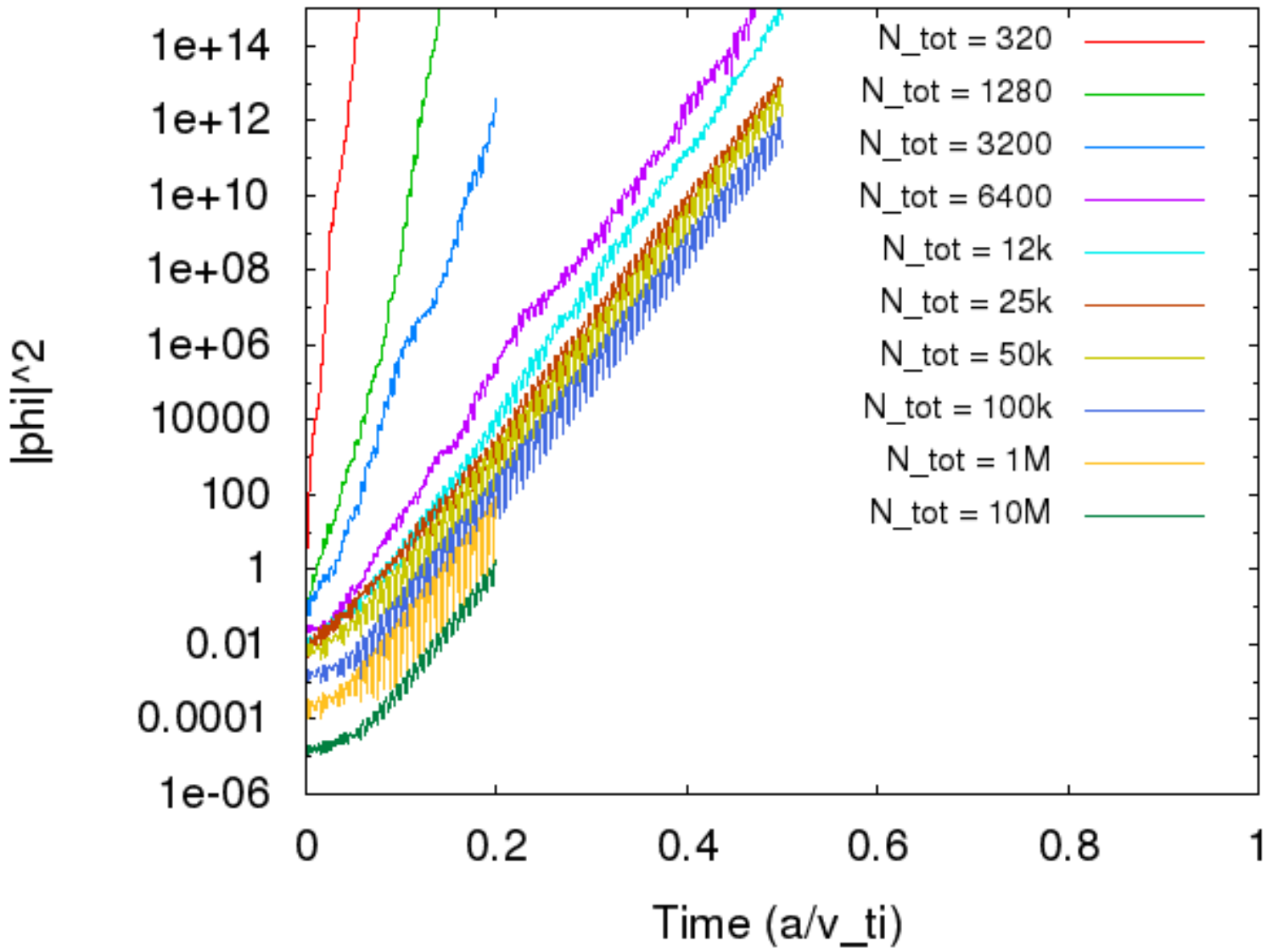
- EAW imposes a severe time constraint if $k_\perp$ is small. Lack of sufficient time resolution results in a CFL-type instability.
- This constraint is relaxed for finite β, where this becomes the lower-frequency shear Alfvén wave.

Instability is converged on timestep



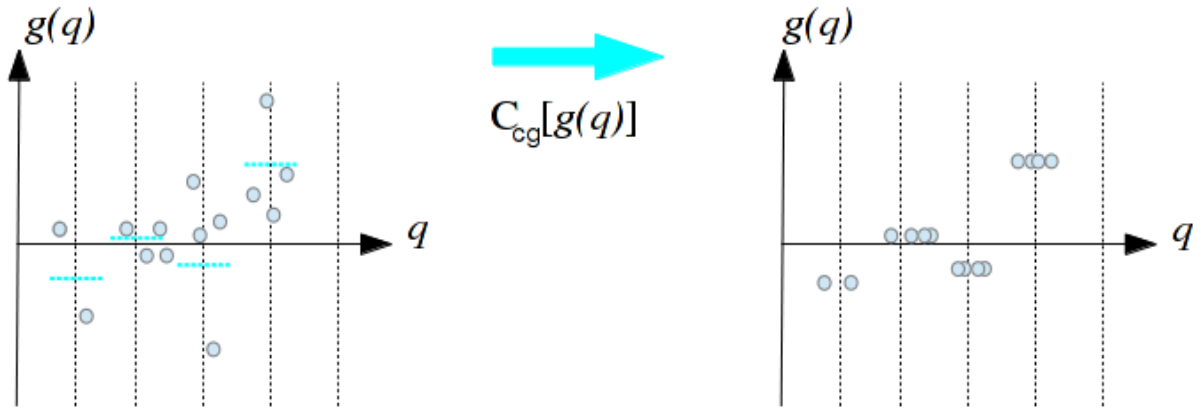


Growth rate is converged on number of particles



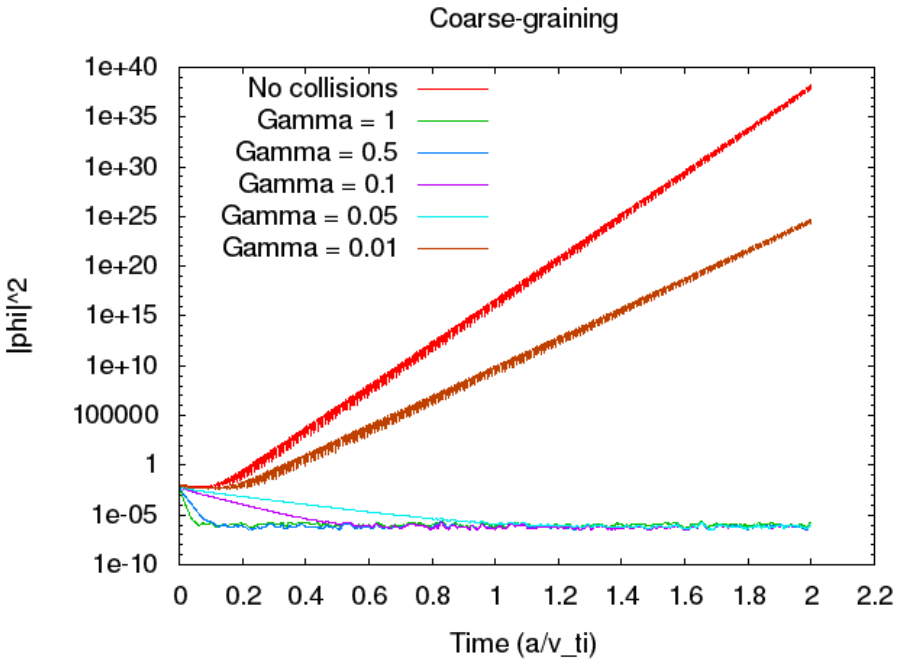
Outline

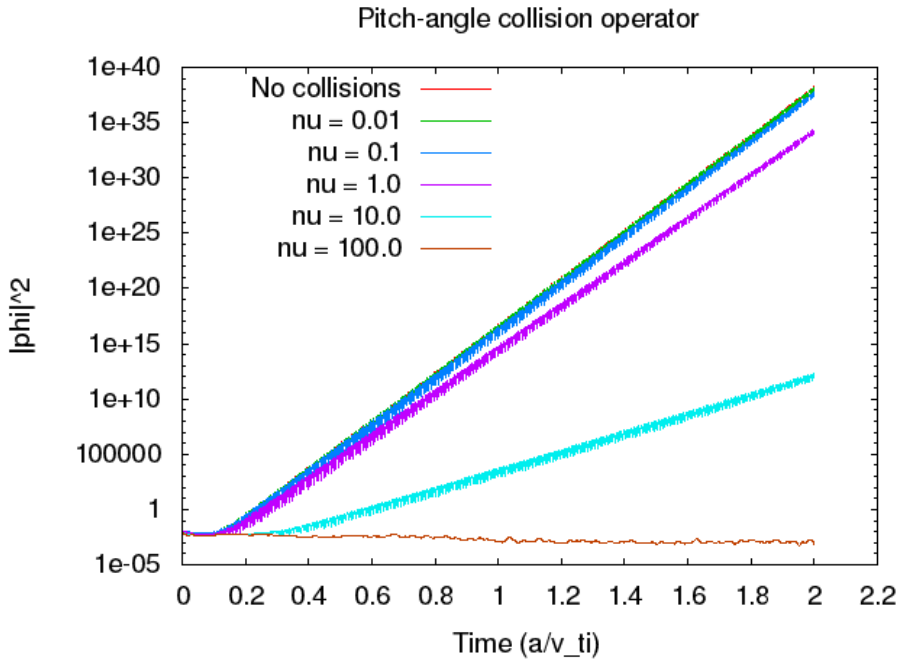
Coarse-graining collision operator



- The coarse-graining operator controls what structure in phase space is deemed physically-relevant
- Relatively moderate application controls the instability, but corresponds to an unacceptably high collision frequency
- We can choose to coarse-grain only once every Γ^{-1} time steps

High collision frequency can damp instability





Recent modification to algorithm: piecewise-constant fields

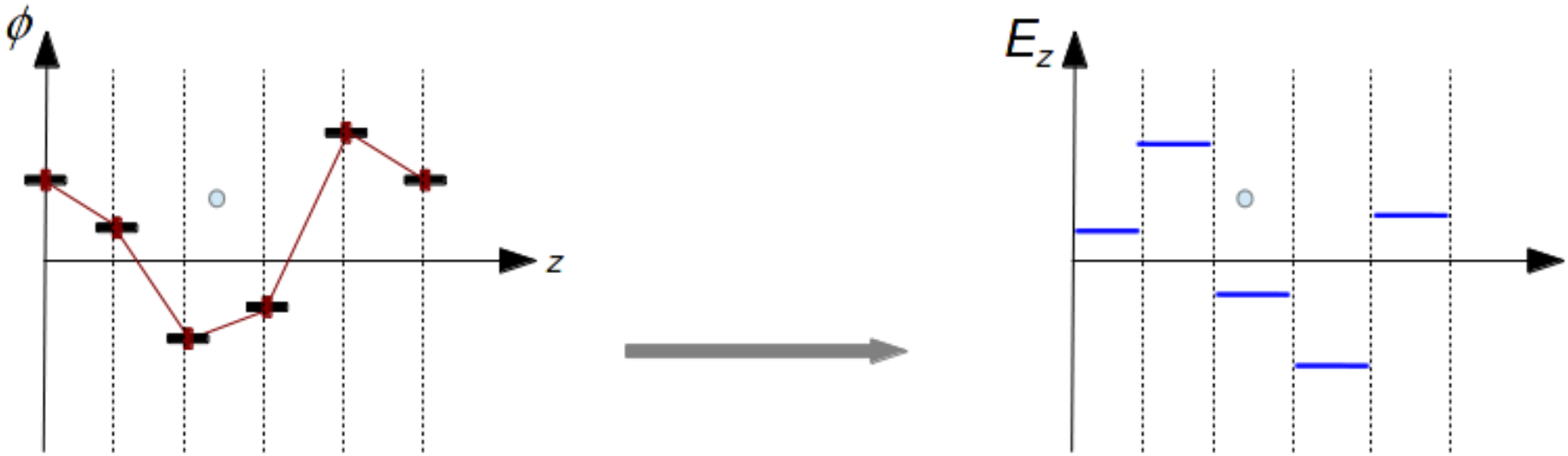
To advance the particles along characteristics, it is necessary to calculate the elecric field. But how?

- Original algorithm calculated the field on the grid, then interpolated to the particle's position

Recent modification to algorithm: piecewise-constant fields

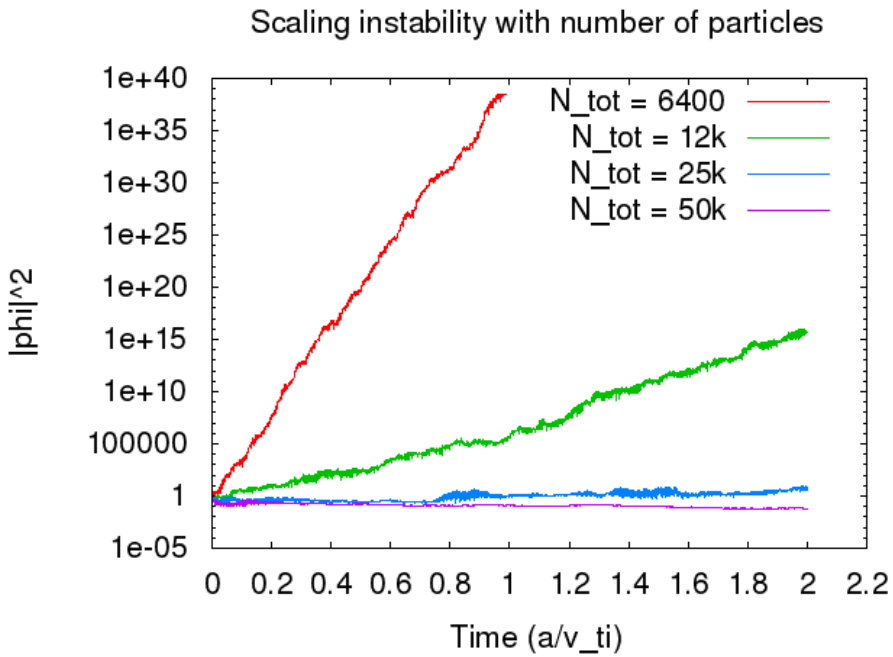
To advance the particles along characteristics, it is necessary to calculate the elecric field. But how?

- Modified algrithm uses the local slope in ϕ to get the field



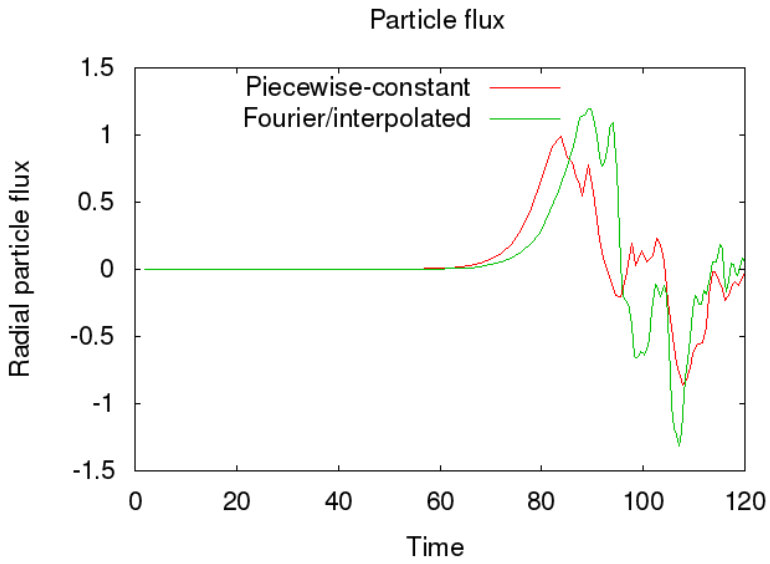
Piecewise-constant fields result in a far more stable algorithm

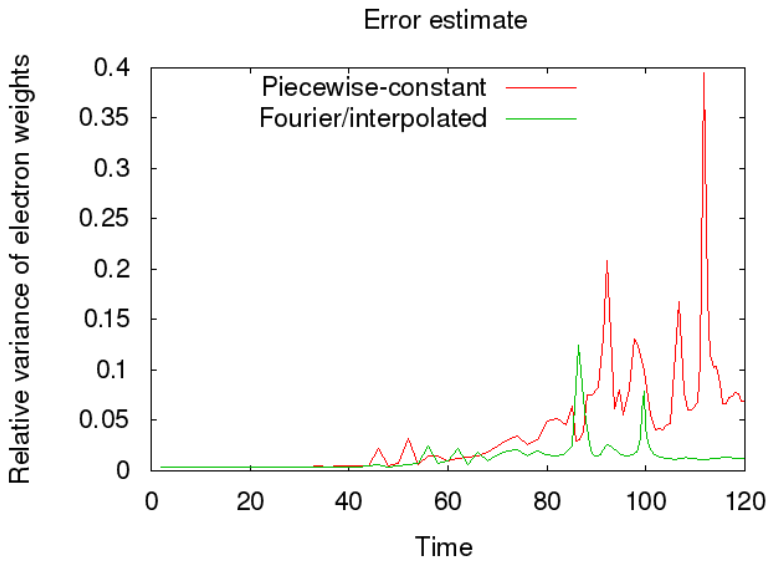
- A tentative explanation for the success of this scheme is dissipation introduced analagously to using constant basis functions in finite-volume methods
- There exists a "residual" instability that disappears for sufficient number of particles



Modification performs similarly, but nonlinear error increases

Nonlinear Z-pinch. $\frac{R}{L_n} = 0.6$.





Outline

How to analyze the δf PIC algorithm in a mathematically rigorous way?

Birdsall and Langdon approach:

1. Obtain the equations for the field a particle experiences due to the presence of many others (P2P)
2. Generalize to the case of finite-size particles
3. Approximate the field on a discrete grid
4. Introduce a discrete timestep

Langdon and Birdsall, 1970
Dawson, 1983

Several numerical instabalities in traditional PIC can be found this way.
How to handle the evolution of the weights?

Particle "weights" are the heart of the algorithm; we need to analyze their evolution

Recall the equation for the weights along characteristics:

$$\frac{Dw_\alpha}{Dt} \equiv \frac{D}{Dt}\left(\frac{g}{F_0}\right)_\alpha = q_\alpha v_{t,\alpha} v_{\parallel,\alpha} E_{\parallel,\alpha}$$

$$\tilde{E}_\parallel = -ik_\parallel \frac{\tilde{\rho}}{k_\perp^2}$$

To find $\rho \equiv n_i - n_e$, we integrate over velocity space, using Monte Carlo integration:

$$n_i = \int g_i dv_\parallel = \int w_i F_{0,i} dv_\parallel \approx \frac{1}{M} \sum_{\beta:i} w_\beta e^{-\frac{1}{2}v_{\parallel,\beta}^2}$$

Where M is the count of the number of particles (or fractions thereof) that contributed to the grid point.

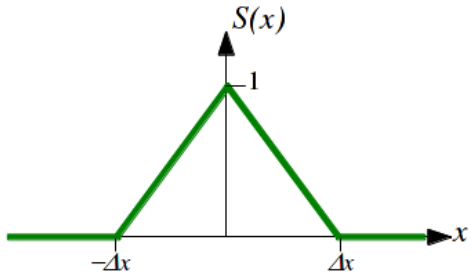
"Shape function" determines interpolation scheme between particles and grid points

First find ρ on a grid defined by $\mathbf{r_i}$:

$$\rho_{\mathbf{r_i}} = \frac{\sum_\beta q_\beta w_\beta e^{-\frac{1}{2}v_{\parallel,\beta}^2} S(\mathbf{r}_\beta - \mathbf{r_i})}{\sum_\gamma S(\mathbf{r}_\gamma - \mathbf{r_i})}$$

Where, for linear interpolation:

$$S(x) = 1 - \frac{|x|}{\Delta x} \quad \text{if} |x| \leq \Delta x$$
$$= 0 \qquad \text{otherwise}$$



After finding the $E(\mathbf{r}_i)$ on the grid, we use the same interpolant to find the field at the particle's position $E_{\parallel,\alpha}$

ODE along characteristics is ultimately linear, so the algorithm can be expressed in matrix form

$$\dot{w}_\alpha = q_\alpha v_{t,\alpha} v_{\parallel,\alpha} E_z(y_\alpha, z_\alpha) = \sum_\beta A_{\alpha\beta}(t) w_\beta$$

Where the matrix A includes all the other PIC operations:

- Interpolating particle β's weight onto a grid to calculate $\rho(y_m, z_n)$
- Fourier transforming the grid to k_y, k_z
- Calculating the fields on the grid
- Inverse transforming to the grid y_i, z_j
- Interpolating to particle α's position

Algorithm is therefore a linear system of first-order ODEs with quadratic (in time) coefficients.

The weight matrix

$$A_{\alpha\beta} = \mathrm{Re}\frac{q_\alpha q_\beta v_{t,\alpha} v_{\parallel,\alpha}}{N_x N_y N_z} \sum_i S\left(\mathbf{r}_\alpha - \mathbf{r}_i\right) \sum_{\mathbf{k}} \frac{-ik_z}{k_\perp^2} e^{i\mathbf{k}\cdot\mathbf{r_i}} \ldots$$
$$\times \sum_n \frac{1}{M_n} e^{-i\mathbf{k}\cdot\mathbf{r_n}} S\left(\mathbf{r}_\beta - \mathbf{r}_n\right) e^{-\frac{1}{2}v_{\parallel,\beta}^2}$$

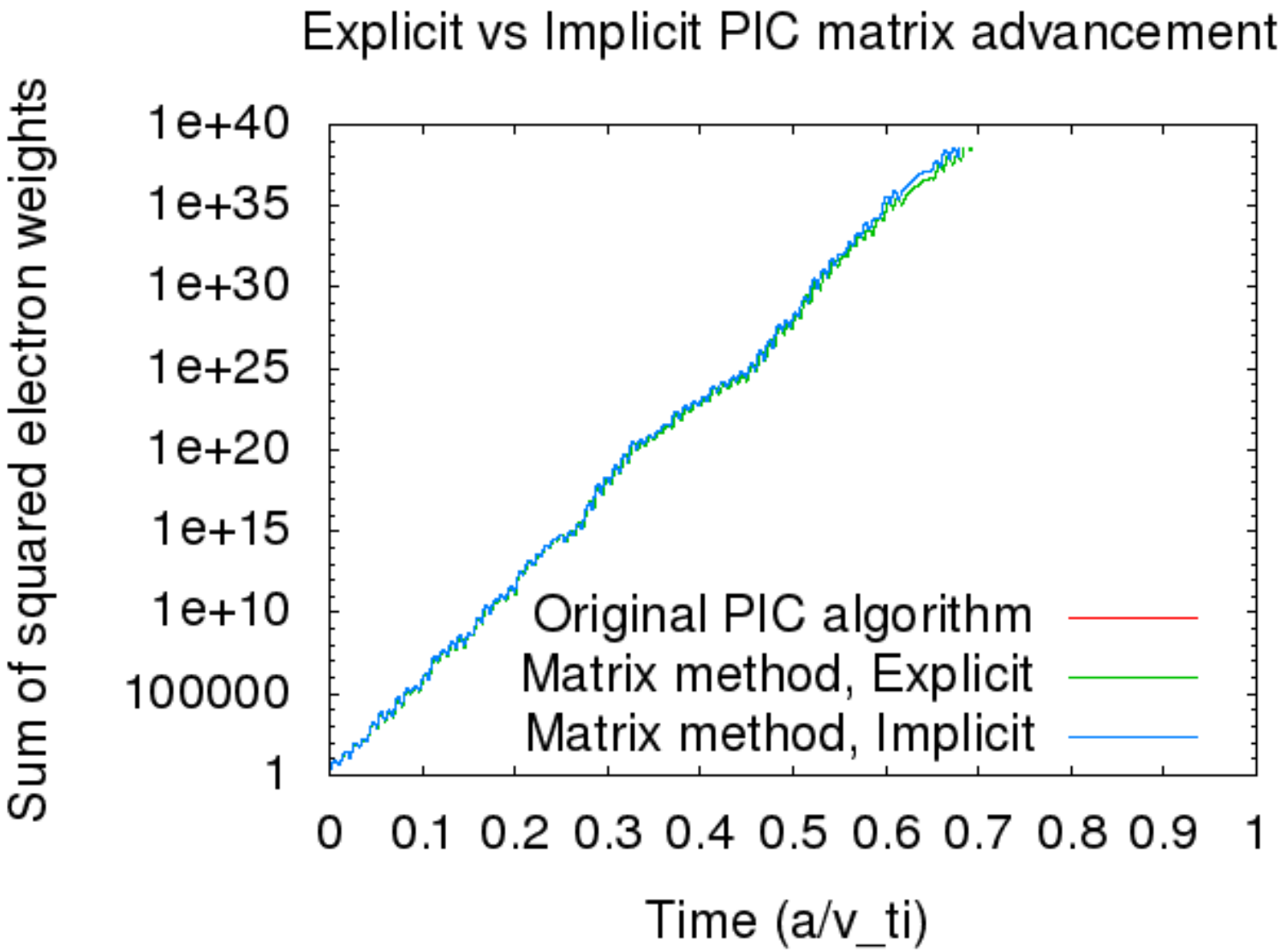
We can approximate $M_n \approx N_{cell}$:

$$A_{\alpha\beta} = \mathrm{Re}\frac{q_\alpha q_\beta v_{t,\alpha} v_{\parallel,\alpha} e^{-\frac{1}{2}v_{\parallel,\beta}^2}}{N_{tot}} \sum_{\mathbf{r}_i, \mathbf{r}_n, \mathbf{k}} \frac{-ik_z}{k_\perp^2} S\left(\mathbf{r}_\alpha - \mathbf{r}_i\right) S\left(\mathbf{r}_\beta - \mathbf{r}_n\right) e^{i\mathbf{k}\cdot(\mathbf{r_i}-\mathbf{r_n})}$$

Matrix formulism has proven to be a useful tool

- **Self-field**
  - A particle's charge could concievably contribute to its own field - a physical absurdity
  - Corresponds to diagonal elements of our matrix $A_{\alpha,\alpha}$, which vanish for this algorithm
- **Eigenfunctions**
  - We can artificially make the matrix constant by turning off z-streaming
  - Can obtain spectrum, find the eigenvector corresponding to the largest eigenvalue (instability growth rate)
  - Unfortunately, no clear relationship has been gleaned
- **PIC as an approximation of P2P**
  - Since we have decomposed the effect of every particle on every other, we can treat it as a particle-to-particle algorithm
- **Time-stepping ruled out**
  - In matrix form, we can advance the particle weights *implicitly*
  - We can confidently assert that even perfect time-stepping would result in the same instability

Truely implicit-PIC sees the same instability



Outline

Take the continuum limit of PIC - a bona-fide P2P scheme

$$A_{\alpha\beta} \propto \sum_{\mathbf{r}_i, \mathbf{r}_n, \mathbf{k}} \frac{-ik_z}{k_\perp^2} S\left(\mathbf{r}_\alpha - \mathbf{r}_i\right) S\left(\mathbf{r}_\beta - \mathbf{r}_n\right) e^{i\mathbf{k}\cdot(\mathbf{r_i}-\mathbf{r_n})}$$

$$= -\int d^3\mathbf{r}_i \int d^3\mathbf{k} \frac{1}{k_\perp^2} \int d^3\mathbf{r}_n e^{i\mathbf{k}_\perp\cdot(\mathbf{r}_{\perp i}-\mathbf{r}_{\perp n})} \ldots$$
$$\times \frac{\partial}{\partial z_i}\left(e^{ik_z(z_i-z_j)}\right) S(\mathbf{r}_\alpha - \mathbf{r}_i) S(\mathbf{r}_\beta - \mathbf{r}_n)$$

Note that $G_\perp(\mathbf{r}_{\perp i} - \mathbf{r}_{\perp n}) \equiv \int d^2\mathbf{k}_\perp \frac{1}{k_\perp^2} e^{i\mathbf{k}_\perp\cdot(\mathbf{r}_{\perp i}-\mathbf{r}_{\perp n})}$ is exactly the Green's function that solves:

$$-\nabla_\perp^2 G_\perp = \delta^2(\mathbf{r}_{\perp i} - \mathbf{r}_{\perp n})$$

$G_\perp = -\frac{1}{2\pi} \ln\left(|\mathbf{r}_{\perp i} - \mathbf{r}_{\perp n}|^2\right)$ in free space

Complication due to the z-direction

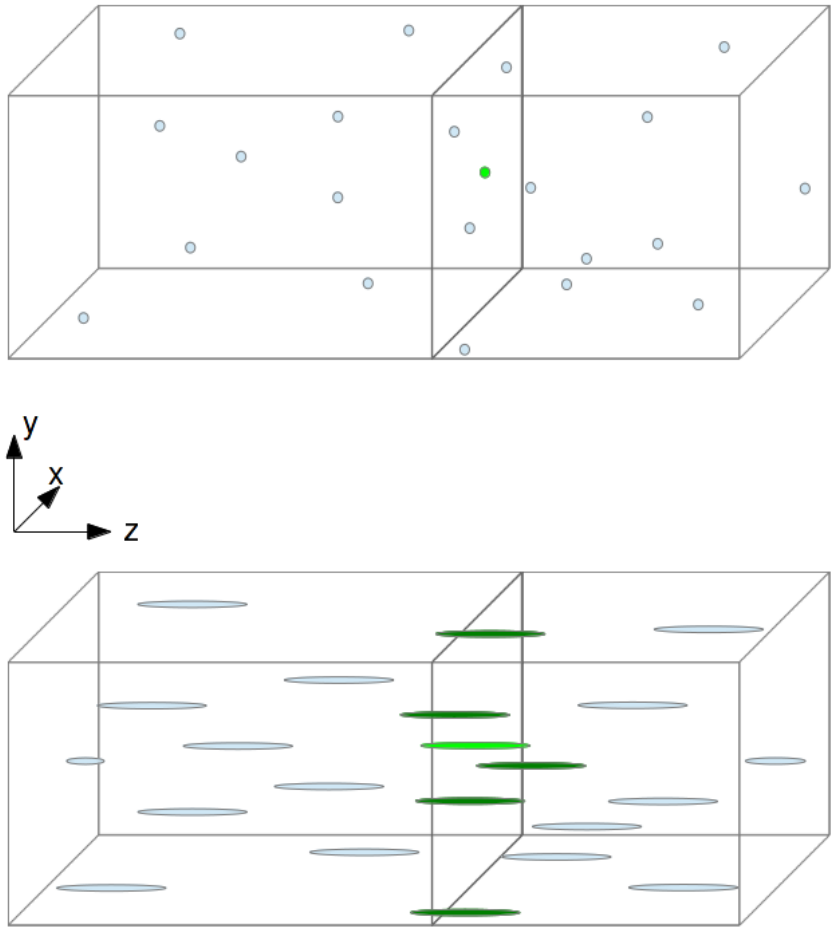
However, the Greens function that we need to find is one that satisfies:

$$-\nabla_\perp^2 G_{3D} = \delta(z_i - z_n)\delta^2(\mathbf{r}_{\perp i} - \mathbf{r}_{\perp n})$$

- Assuming seperability: $G_{3D} = \delta(z_i - z_n)G_\perp(\mathbf{r}_{\perp i} - \mathbf{r}_{\perp n})$.
- This is a manifestation of $\rho^* \to 0$

Solution: keep particles finite-size in z:

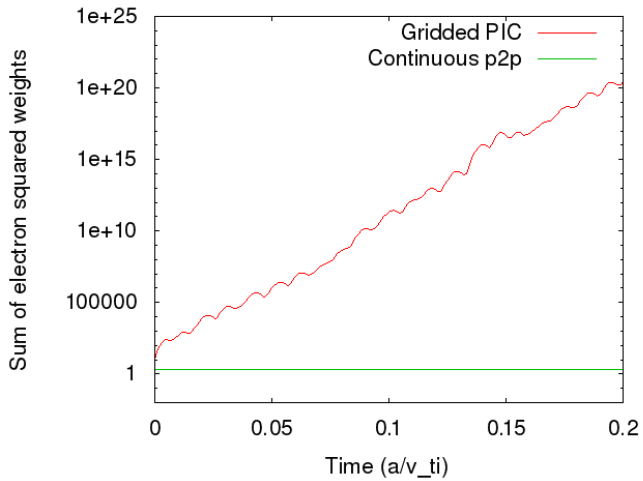
Gyrokinetic ordering prevents particles at different z from influencing eachother



P2P approach eliminates the instability!

Disclaimer: we are erroneously using the free-space 2D Poisson Green's function at the moment. Will need the one for periodic conditions.

$$A_{\alpha,\beta} = \frac{1}{N_{cell}} q_\alpha q_\beta v_\alpha v_{t,\alpha} e^{-\frac{1}{2}v_\beta^2} G_\perp(\mathbf{r}_{\perp\beta} - \mathbf{r}_{\perp\alpha}) \int dz_2 \, S(z_\beta - z_2) \frac{\partial}{\partial z_2} S(z_\alpha - z_2)$$



This tentative success suggests that the instability is likely a manifestation of the Birdsall and Langdon finite-grid instability.

Future work: Improve ROMA functionality

- Revisit Ampere's Law. Implement Kotchenreuther Green's function scheme and/or the split-weight scheme to avoid cancellation problem
- Additional validation against known physics
- Add high-energy particle capability
- Implement toroidal flux coordinates and shear, validate
- Complete conservative collision operator
- Couple code to Trinity

References

- Aydemir, "A unified Monte Carlo interpretation of particle simulatinos and applications to non-neutral plasmas." *Physics of Plasmas*, **1**:822 (1994)
- Langdon and Birdsall, "Theory of Plasma Simulation Using Finite-Size Particles." *Physics of Fluids*, **13**:2115 (1970)
- Broemstrup, "Advanced Lagrangian Simulations Algorithms for Magnetized Plasma Turbulence." PhD Thesis, University of Maryland Department of Physics (2008)
- Chen and Parker, "Gyrokinetic turbulence simulations with kinetic electrons," *Physics of Plasmas*, **8**:2095 (2001)
- Dawson, "Particle simulation of plasmas." *Rev. Mod. Phys*, **55**:403 (1983)
- Frieman and Chen, "Nonlinear gyrokinetic equations for low-frequency electromagnetic waves in general plasma equilibria." *Physics of Fluids*, **25**:502 (1982)

- Howes, et al. "Astrophysical gyrokinetics: Basic equations and linear theory." *Astrophysical Journal*, **651**:590 (2006)
- Lee, "Gyrokinetic particle simulation model." *Journal of Computational Physics*, **72**:243 (1987)
- Manuilksiy and Lee, "The split-weight particle simulation scheme for plasmas," *Physics of Plasmas*, **7**:1381 (2000)
- Press, et al. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*, Cambridge University Press. (2007)
- Ricci, et al, "Gyrokinetic linear theory of the entropy mode in a Z pinch." *Physics of Plasmas*, **13**: 062102 (2006)
- Ricci, Rogers, Dorland, "Small-Scale Turbulence in a Closed-Field-Line Geometry" *Physical Review Letters*, **97**:245001 (2006)

Thank you!

Questions?