LA-UR-15-26051
Approved for public release; distribution is unlimited.

Title: Multirate Particle-in-Cell Time Integration Techniques of Vlasov-Maxwell Equations for Collisionless Kinetic Plasma Simulations

Author(s): Chen, Guangye
Chacon, Luis
Knoll, Dana Alan
Barnes, Daniel C

Intended for: 13th US National Congress on Computational Mechanics, 2015-07-26/2015-07-30 (San Diego, California, United States)

Issued: 2015-07-31

Multirate Particle-in-Cell Time Integration Techniques of Vlasov-Maxwell Equations for Collisionless Kinetic Plasma Simulations

Guangye Chen

Applied Math. and Plasma Physics Group
Theoretical Division
Los Alamos National Laboratory
P.O. Box 1663, Los Alamos, NM 87544

Collaborators:

L. Chacón, D. A. Knoll (LANL)
D. C. Barnes (Coronado Consulting)

USNCCM13
Advances in IMEX time integration of Multiphysics systems
July 29th, 2015

Work funded by the LANL LDRD program and DOE-ASCR

Los Alamos NATIONAL LABORATORY EST.1943

gchen@lanl.gov

Outline

- Vlasov-Maxwell Particle-in-cell (PIC) methods for plasmas
- Explicit, semi-implicit, and implicit time integrations
- Implicit PIC formulation
  - JFNK with nonlinear elimination allows different treatments of disparate scales
  - Discrete conservation properties (energy, charge, canonical momentum, etc.)
- Some numerical examples
- Summary

gchen@lanl.gov

Introduction

gchen@lanl.gov

Vlasov-Maxwell equations for collisionless plasmas

Vlasov Equation

$$\partial_t f + \mathbf{v}\cdot\nabla f + \frac{q}{m}(\mathbf{E}+\mathbf{v}\times\mathbf{B})\cdot\nabla_v f = 0 \tag{1}$$

Maxwell Equations

$$\begin{aligned}
\partial_t \mathbf{B} + \nabla\times\mathbf{E} &= 0 && (2)\\
-\mu_0\epsilon_0\partial_t\mathbf{E} + \nabla\times\mathbf{B} &= \mu_0\mathbf{j} && (3)\\
\nabla\cdot\mathbf{B} &= 0 && (4)\\
\nabla\cdot\mathbf{E} &= \frac{\rho}{\epsilon_0} && (5)
\end{aligned}$$

where

$$\mathbf{j} = \int q\mathbf{v} f d\mathbf{v} \;\; ; \;\; \rho = \int q f d\mathbf{v}$$

- Various approximations are possible: Darwin model, electrostatic model, etc.

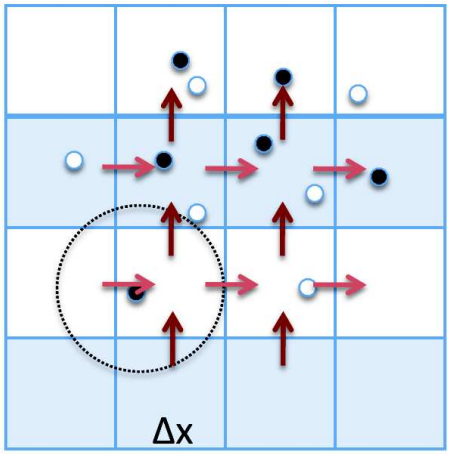
gchen@lanl.gov

Particle-in-cell (PIC) methods for kinetic plasma simulation

- Lagrangian solution by the **method of characteristics**:

$$f(\mathbf{x},\mathbf{v},t) = f_0\left(\mathbf{x} - \int_0^t dt\mathbf{v}, \mathbf{v} - \frac{q}{m}\int_0^t dt(\mathbf{E}+\mathbf{v}\times\mathbf{B})\right) \; ; \; \mathbf{x}(t=0) = \mathbf{x}_0 \; ; \; \mathbf{v}(t=0) = \mathbf{v}_0$$

- PIC approach follows characteristics employing **macroparticles** (volumes in phase space)



$$f(\mathbf{x},\mathbf{v},t) = \textstyle\sum_p \delta(\mathbf{x}-\mathbf{x}_p)\delta(\mathbf{v}-\mathbf{v}_p)$$

$$\begin{aligned} \dot{\mathbf{x}}_p &= \mathbf{v}_p \\ \dot{\mathbf{v}}_p &= \frac{q_p}{m_p}(\mathbf{E}+\mathbf{v}\times\mathbf{B}) \end{aligned}$$

$$\begin{aligned} \partial_t\mathbf{B} + \nabla\times\mathbf{E} &= 0 \\ -\mu_0\epsilon_0\partial_t\mathbf{E} + \nabla\times\mathbf{B} &= \mu_0\mathbf{j} \\ \nabla\cdot\mathbf{B} &= 0 \\ \nabla\cdot\mathbf{E} &= \frac{\rho}{\epsilon_0} \end{aligned}$$

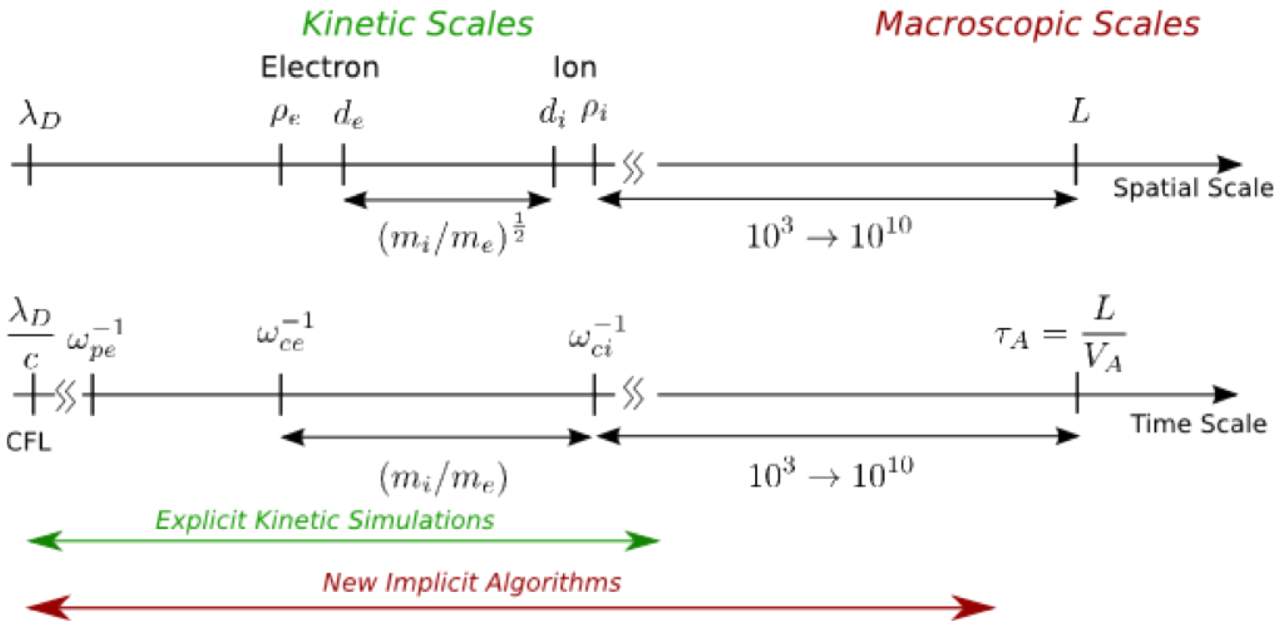
$$\delta(\mathbf{x}-\mathbf{x}_p) \longrightarrow S(\mathbf{x}-\mathbf{x}_p) \; ; \; E_p = \sum_i E_i S(x_i - x_p) \; ; \; j_i = \sum_p j_p S(x_i - x_p)$$

Los Alamos NATIONAL LABORATORY EST. 1943

gchen@lanl.gov

Kinetic Plasma Simulation

- A fully ionized collisionless plasma: ions, electrons, and electromagnetic fields
- **Challenge**: to integrate electron-ion-field equations on a system length scale and an ion time-scale, while retaining electron kinetic effects accurately.



(We are developing a new implicit algorithm for long-term, system-scale simulations.)

- Problem features a **hierarchical description**:
  - How to design a multi-scale algorithm? (e.g. IMEX, exponential integrators...)
  - How to respect conservation laws, and constraints?

Los Alamos NATIONAL LABORATORY EST.1943

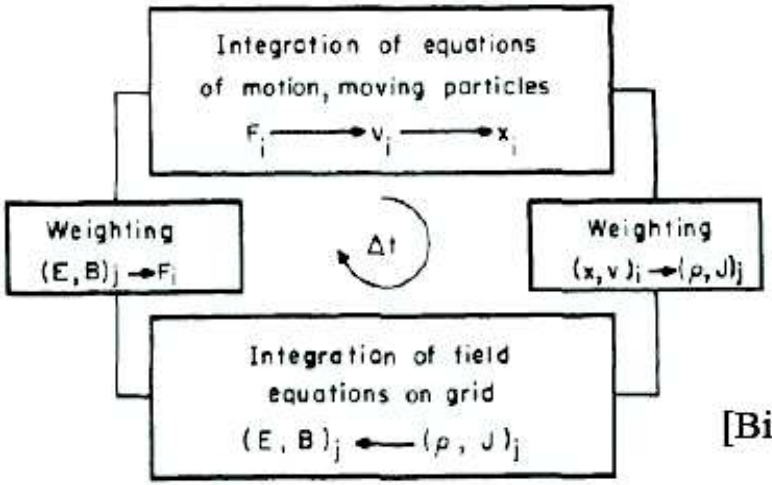
gchen@lanl.gov

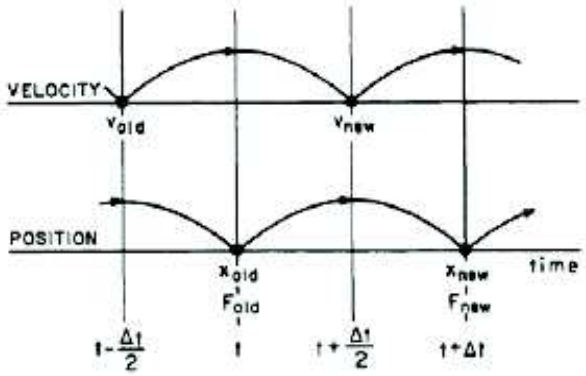
Algorithmic Considerations

gchen@lanl.gov

State-of-the-art *classical* PIC algorithm is explicit

- Particle positions and velocities "leap-frogs", field-solve at position update in lock-step:





[Birdsall and Langdon, Plasma physics via computer simulation]

- *Implementation is straightforward*, but...
- Efficiency limitations:
  - CFL-type instability: $\min(\omega_{pe}\Delta t < 1, c\Delta t < \Delta x)$. **Minimum temporal resolution**
  - Finite-grid instability:$\Delta x < \lambda_{Debye}$. **Minimum spatial resolution**
  - Memory bound: challenging for efficient use of modern computer architectures.
- Accuracy limitations:
  - **Lack of discrete energy conservation**, problematic for long-time-scale simulations
- It is well-known that implicit methods can remove the stability constraints of explicit methods.

gchen@lanl.gov

Fully implicit PIC formulation (at first glance)

- A fully implicit formulation couples particles and fields non-trivially (integro-differential PDE):

$$\frac{f^{n+1}-f^n}{\Delta t}+\mathbf{v}\cdot\nabla\frac{f^{n+1}+f^n}{2} \quad - \quad \frac{q}{m}\nabla\frac{\Phi^{n+1}+\Phi^n}{2}\cdot\nabla_{\mathbf{v}}\frac{f^{n+1}+f^n}{2}=0$$

$$\nabla^2\Phi^{n+1} \quad = \quad \int d\mathbf{v}\, f^{n+1}(\mathbf{x},\mathbf{v},t)$$

(electrostatic system, as an example)

- In PIC, too many degrees of freedom:
  - There are N_p particles, each particle requiring 6 equations (6 dimensions),
  - Field requires N_g equations, one per grid point.
- If implemented fully implicitly, an impractically large algebraic system of equations results:

$$\boxed{\mathbf{G}(\{\mathbf{x},\mathbf{v}\}_p^{n+1},\{\Phi^{n+1}\}_g)=0} \;\rightarrow\; \dim(\mathbf{G})=2dN_p+N_g$$

  - How to evaluate the Jacobian?
  - Unaffordable memory requirements.
  - Algorithmic issues are showstoppers (e.g., how to precondition it?)
  - Likely to advance particle-fields in lock-step.

gchen@lanl.gov

Semi-implicit PIC formulation (at second glance)

➤ Basic idea[1] [2]: *To advance fields,* ρ *and* $\mathbf{j}$ *can be obtained from moment equations.*

$$\partial_t n_\alpha = -\nabla \cdot \mathbf{\Gamma}_\alpha$$

$$\partial_t \mathbf{\Gamma}_\alpha = \frac{q_\alpha n_\alpha}{m_\alpha}(\mathbf{E}^*) - \nabla \cdot \mathbf{\Pi}_\alpha$$

where

$$n_\alpha = \int f_\alpha d\mathbf{v} \quad , \quad \mathbf{\Gamma}_\alpha = \int \mathbf{v} f_\alpha d\mathbf{v}$$

$$\mathbf{\Pi}_\alpha = \int \mathbf{v}\mathbf{v} f_\alpha d\mathbf{v} \quad , \quad \mathbf{E}^* = \theta \mathbf{E}^n + (1-\theta)\mathbf{E}^{n+1}$$

➤ The moment equations are *linearized* to provide an estimate for ρ and $\mathbf{j}$, integrated with a *semi-implicit* scheme.

- ➪ Decoupling of solving field and particle equations ✓
- ➪ Field and particle equations are usually advanced in lock-step X (but can be relaxed)
- ➪ Noise issues with light wave dispersion for large timesteps X
- ➪ No discrete energy conservation X

---

[1] Mason, R. J. (1981)

[2] Brackbill, J. U., and Forslund, D. W. (1982)

Particle enslavement (nonlinear elimination)

- Full residual $\mathbf{G}(\{x,v\}_p,\{\Phi\}_g)=0$ is impractical to implement
- Semi-implicit schemes have undesirable features
- Alternative: nonlinearly eliminate particle quantities from dependent variables:
  - Formally, particle equations of motion (EOM) are functionals of the electrostatic potential:

$$x_p^{n+1}=x_p[\Phi^{n+1}]\ ;\ v_p^{n+1}=v_p[\Phi^{n+1}]$$

$$\mathbf{G}(\mathbf{x}_p{}^{n+1},\mathbf{v}_p{}^{n+1},\mathbf{\Phi}^{n+1})=\mathbf{G}(\mathbf{x}[\mathbf{\Phi}^{n+1}],\mathbf{v}[\mathbf{\Phi}^{n+1}],\mathbf{\Phi}^{n+1})=\tilde{\mathbf{G}}(\mathbf{\Phi}^{n+1})$$

Nonlinear residual can be *unambiguously* formulated in terms of electrostatic potential only!

- Storage requirements are dramatically decreased, making it tractable:
  - Nonlinear solver storage requirements $\propto N_g$, **comparable to a fluid simulation**
  - Particle quantities $\Rightarrow$ auxiliary variables: only a **single copy of particle population** needs to be maintained in memory throughout the nonlinear iteration
  - Considerable freedom is gained for pushing particles (EOMs can be solved "explicitly" with "known" fields)

gchen@lanl.gov

Jacobian-Free Newton-Krylov (JFNK) Method

- After spatial and temporal discretization $\Rightarrow$ a large set of nonlinear equations: $\boxed{\vec{G}(\vec{x}^{n+1}) = \vec{0}}$
- Converging nonlinear couplings requires iteration: Newton-Raphson method:

$$\left.\frac{\partial \vec{G}}{\partial \vec{x}}\right|_k \delta\vec{x}_k = -\vec{G}(\vec{x}_k)$$

- Jacobian linear systems result, which require a linear solver $\Rightarrow$ Krylov subspace methods (GMRES)
  - Only require matrix-vector products to proceed.
  - Jacobian-vector product can be computed Jacobian-free (**CRITICAL**: no need to form Jacobian matrix):

$$\left(\frac{\partial \vec{G}}{\partial \vec{x}}\right)_k \vec{y} = J_k\vec{y} = \lim_{\epsilon\to 0}\frac{\vec{G}(\vec{x}_k + \epsilon\vec{y}) - \vec{G}(\vec{x}_k)}{\epsilon}$$

  - Krylov methods can be "easily" preconditioned: $P_k^{-1} \sim J_k^{-1}$

$$J_k P_k^{-1} P_k \delta\vec{x} = -\vec{G}_k$$

gchen@lanl.gov

JFNK preconditioning via implicit moment equations

- We use 1D electrostatic equations as an example :

$$\partial_t n_e = -\partial_x \Gamma_{ex} \tag{6}$$

$$\partial_t \Gamma_{ex} = \frac{q_e}{m_e} n_e E_x + \partial_x \mathbf{\Pi}_e \tag{7}$$

$$\epsilon_0 \partial_t E_x = q_e \Gamma_{ex} \tag{8}$$

- Linearize (e.g. $n_e = n_{e0} + \delta n_e$) and discretize:

$$\frac{\delta n_e}{\Delta t} = -\partial_x \delta \Gamma_{ex} \tag{9}$$

$$\frac{\delta \Gamma_{ex}}{\Delta t} \approx \frac{q_e}{m_e} \left[\delta n_e E_x + (n_e)_p \delta E_x \right] + \partial_x \left[\left(\frac{\mathbf{\Pi}_e}{n_e} \right)_p \delta n_e \right] \tag{10}$$

$$\epsilon_0 \delta E_x = \Delta t \left[q_e \delta \Gamma_{ex} - G(E_x) \right] \tag{11}$$

  - Eq.(4) (5) can be combined, and inverted by a tri-diagonal solver;
  - δE can be obtained from Eq.(6).

gchen@lanl.gov

Algorithmic implementation details

- The nonlinear residual formulation $\mathbf{G}(\mathbf{E}^{n+1})$ based on Vlasov-Maxwell formulation is as follows:

1. Input $\mathbf{E}$ (given by JFNK iterative method)
2. Push particles (i.e., find $\mathbf{x}_p[\mathbf{E}]$, $\mathbf{v}_p[\mathbf{E}]$ by solving equations of motion)
3. Compute moments (current density)
4. Form the residual of Maxwell equations
5. repeat $1 \rightarrow 4$ until convergence

gchen@lanl.gov

Algorithmic implementation details

➤ The nonlinear residual formulation $\mathbf{G}(\mathbf{E}^{n+1})$ based on Vlasov-Maxwell formulation is as follows:

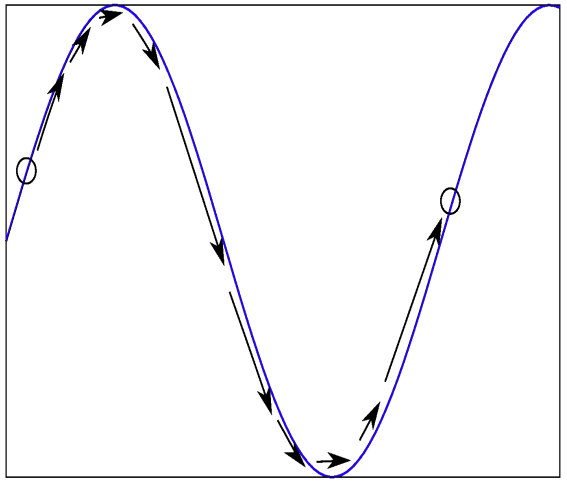
1. Input $\mathbf{E}$ (given by JFNK iterative method)

2. Push particles (i.e., find $\mathbf{x}_p[\mathbf{E}]$, $\mathbf{v}_p[\mathbf{E}]$ by solving equations of motion): Multi-rate integration
   (a) Explicit timesteps (particle orbits are fully resolved)
   (b) Adaptive sub-stepping (sub-cycling)
   (c) Orbit-averaging

3. Compute moments (current density)

4. Form the residual of Maxwell equations

5. repeat $1 \rightarrow 4$ until convergence

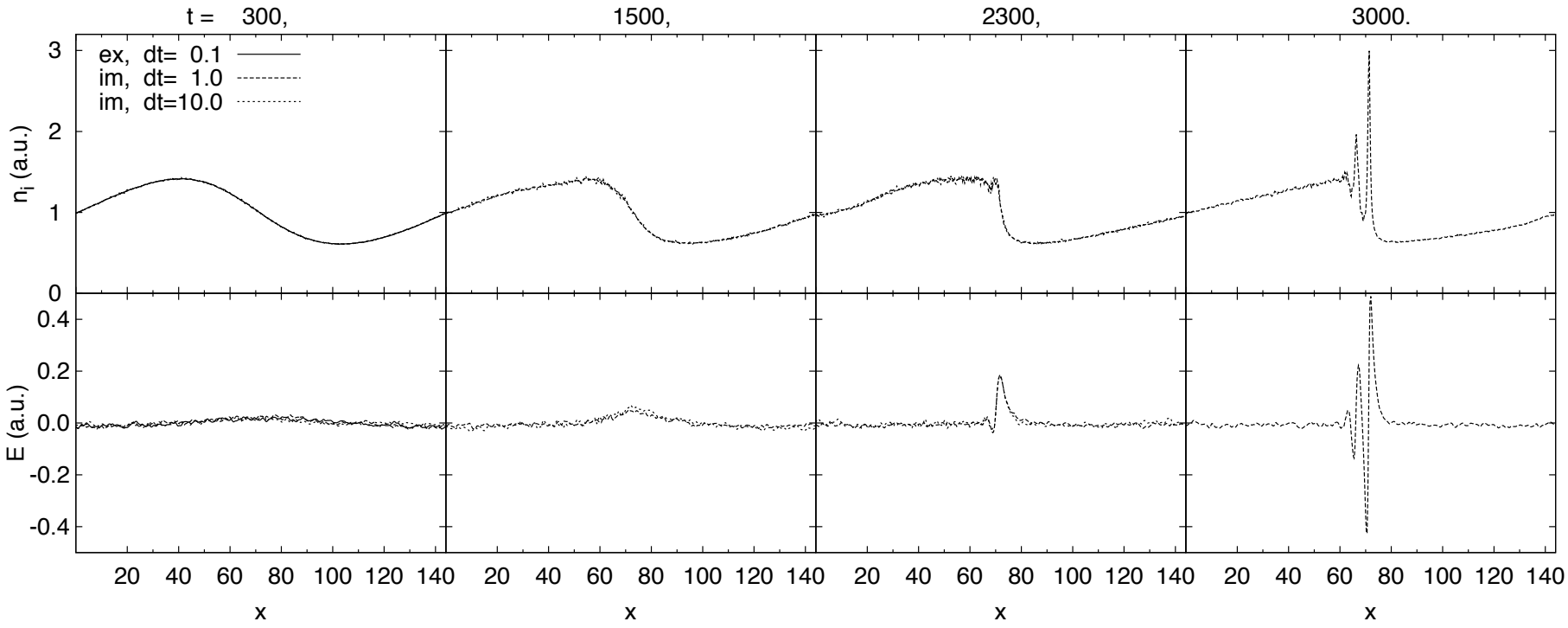


gchen@lanl.gov

Some Numerical Examples

gchen@lanl.gov

Ion acoustic shock wave



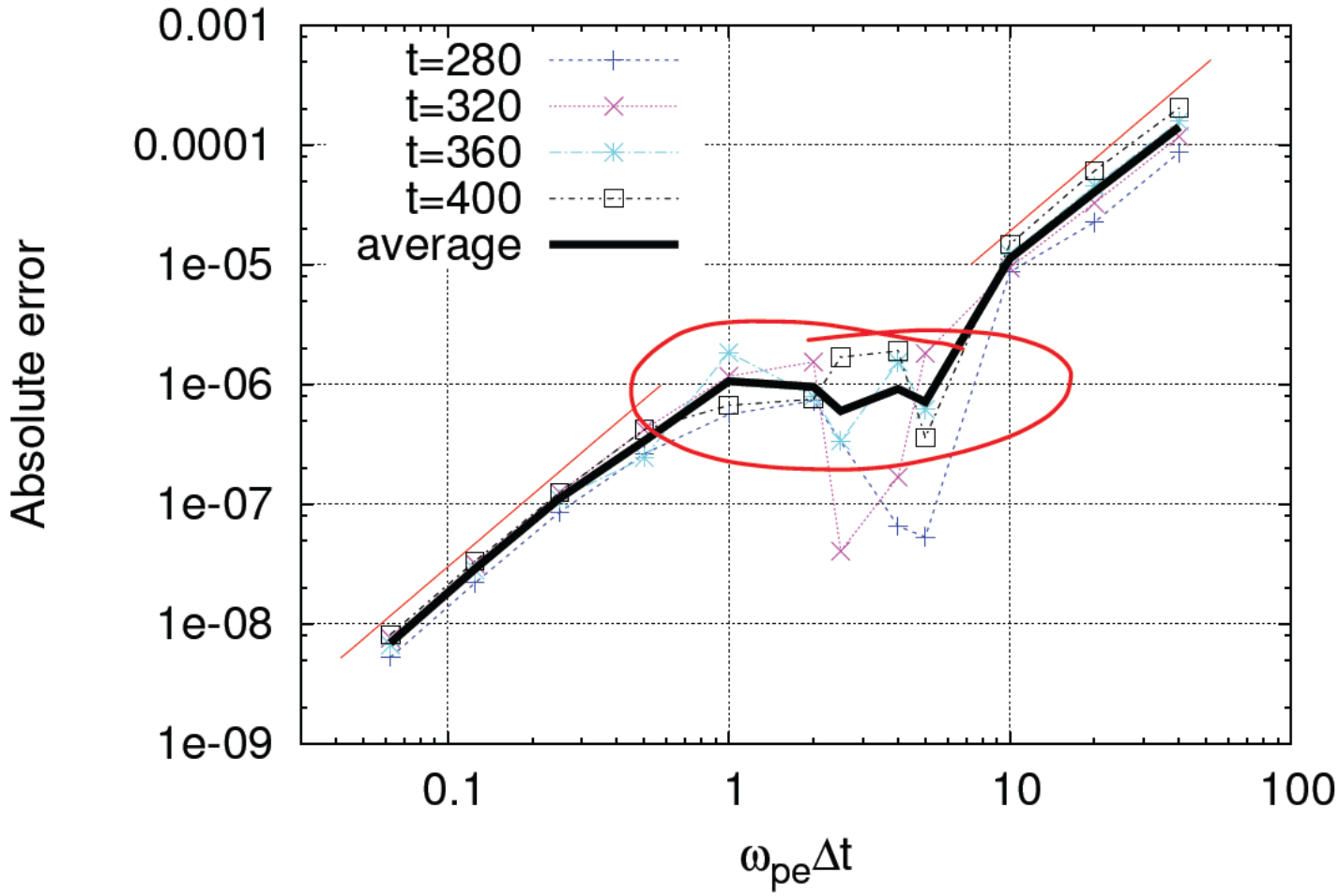
- Propagating IAW with perturbation level $\epsilon = 0.4$, with 4000 particles/cell.
- Realistic mass ratio ($m_i/m_e = 2000$).
- Shock wave length scale$\sim$Debye length.

Los Alamos NATIONAL LABORATORY EST.1943

gchen@lanl.gov

Impact of Multi-rate Integrator on Temporal Rate of Convergence

- Weibel instability
- Numerical demonstration of 2nd order accuracy in time (Δt for field equations)



Los Alamos NATIONAL LABORATORY EST.1943

gchen@lanl.gov

2D Electron Weibel instability: preconditioner performance

$$L_x \times L_y = 22 \times 22\ (d_e^2),\ N_{pc} = 200,\ \Delta t = 0.1\omega_{pi}^{-1}$$

$$N_x \times N_y = 128 \times 128$$

m_i/m_e	no preconditioner		with preconditioner	
	Newton	GMRES	Newton	GMRES
25	5.8	192.5	3	0
100	5.7	188.8	3	0
1836	7.7	237.8	4	2.8

$$m_i/m_e = 1836$$

$N_x \times N_y$	no preconditioner		with preconditioner	
	Newton	GMRES	Newton	GMRES
16×16	3.7	20	3	0.9
32×32	4	38.5	3	0.9
64×64	4.3	79.9	3	0.2

gchen@lanl.gov

Summary and conclusions

- We have developed a multi-rate PIC formulation that employs
  - large timesteps for slow field evolution, and
  - small (adaptive) timesteps for particle orbit integrations
- Implementation is based on a JFNK solver with
  - Nonlinear elimination
  - Moment preconditioning
- The approach is free of numerical instabilities: $\omega_{pe}\Delta t \gg 1$, and $\Delta x \gg \lambda_D$
  - Requires many fewer dofs (vs. explicit PIC) for comparable accuracy in challenging problems
  - Significant gains (vs. conventional explicit PIC) may be possible for large scale simulations.

gchen@lanl.gov