BERKELEY LAB Lawrence Berkeley National Laboratory

U.S. DEPARTMENT OF ENERGY

A high-order accurate Particle-In-Cell method for Vlasov-Poisson problems over long time integrations

Andrew Myers
atmyers2@gmail.com
Applied Numerical Algorithms Group, Computational Research Division

with Phillip Colella, Brian Van Straalen, Bei Wang, Boris Lo, Victor Minden

Advanced Modeling Seminar Series
NASA Ames Research Center
November10th, 2016

https://arxiv.org/abs/1602.00747

Talk Outline

- Background and motivation
- Description of our numerical method
- Implementation using the Chombo framework
- Numerical results
- Summary and future research directions

Talk Outline

- Background and motivation
- Description of our numerical method
- Implementation using the Chombo framework
- Numerical results
- Summary and future research directions

The Vlasov-Poisson Equations

$$\frac{\partial f}{\partial t} + \boldsymbol{v} \cdot \frac{\partial f}{\partial \boldsymbol{x}} - \boldsymbol{E} \cdot \frac{\partial f}{\partial \boldsymbol{v}} = 0$$

$$\nabla^2 \phi = -\rho \qquad\qquad \boldsymbol{E} = -\nabla \phi$$

- The Vlasov-Poisson equations describe the evolution of a charged, collisionless fluid in phase space.
- The electric field is obtained by solving Poisson's equation.
- Applicable as a simplified model to plasma physics (space plasmas, accelerator modeling, controlled fusion) and to cosmology.

Eulerian Methods

- Nonlinear advection equation in a high-dimensional space, can be solved by Eulerian techniques.
- Advantages:
  - Strong body of theory on finite volume methods
  - Do not suffer from “noise”
  - High order methods exist (Vogman + 2014)
- Disadvantages:
  - High cost of grids in high-dimensional (4D, 5D, or 6D) spaces

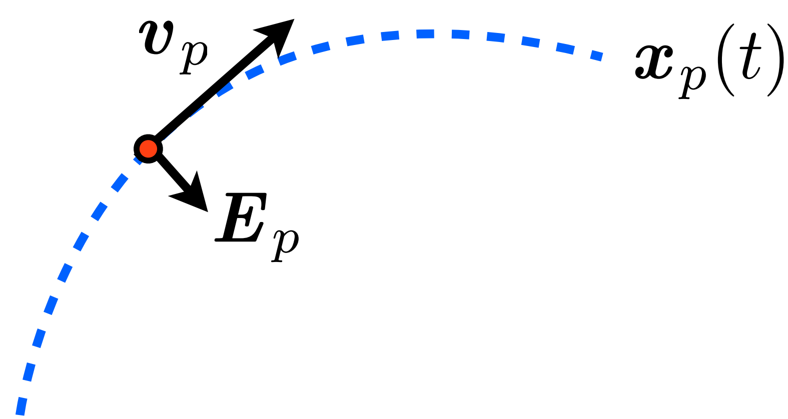
Particle Methods

- Discretize system with set of Lagrangian interpolating points, $\mathbb{P}$

$$f(\boldsymbol{x}, \boldsymbol{v}, t_{\text{ini}}) \approx \sum_{p \in \mathbb{P}} q_p \delta\left(\boldsymbol{x} - \boldsymbol{x}_p^i\right) \delta\left(\boldsymbol{v} - \boldsymbol{v}_p^i\right)$$

- Reduces problem to system of ODEs for particle trajectories:

$$\frac{dq_p}{dt} = 0$$
$$\frac{d\boldsymbol{x}_p}{dt} = \boldsymbol{v}_p$$
$$\frac{d\boldsymbol{v}_p}{dt} = -\boldsymbol{E}_p$$



- Can reconstruct distribution at later times from $(\boldsymbol{x}_p(t), \boldsymbol{v}_p(t))$

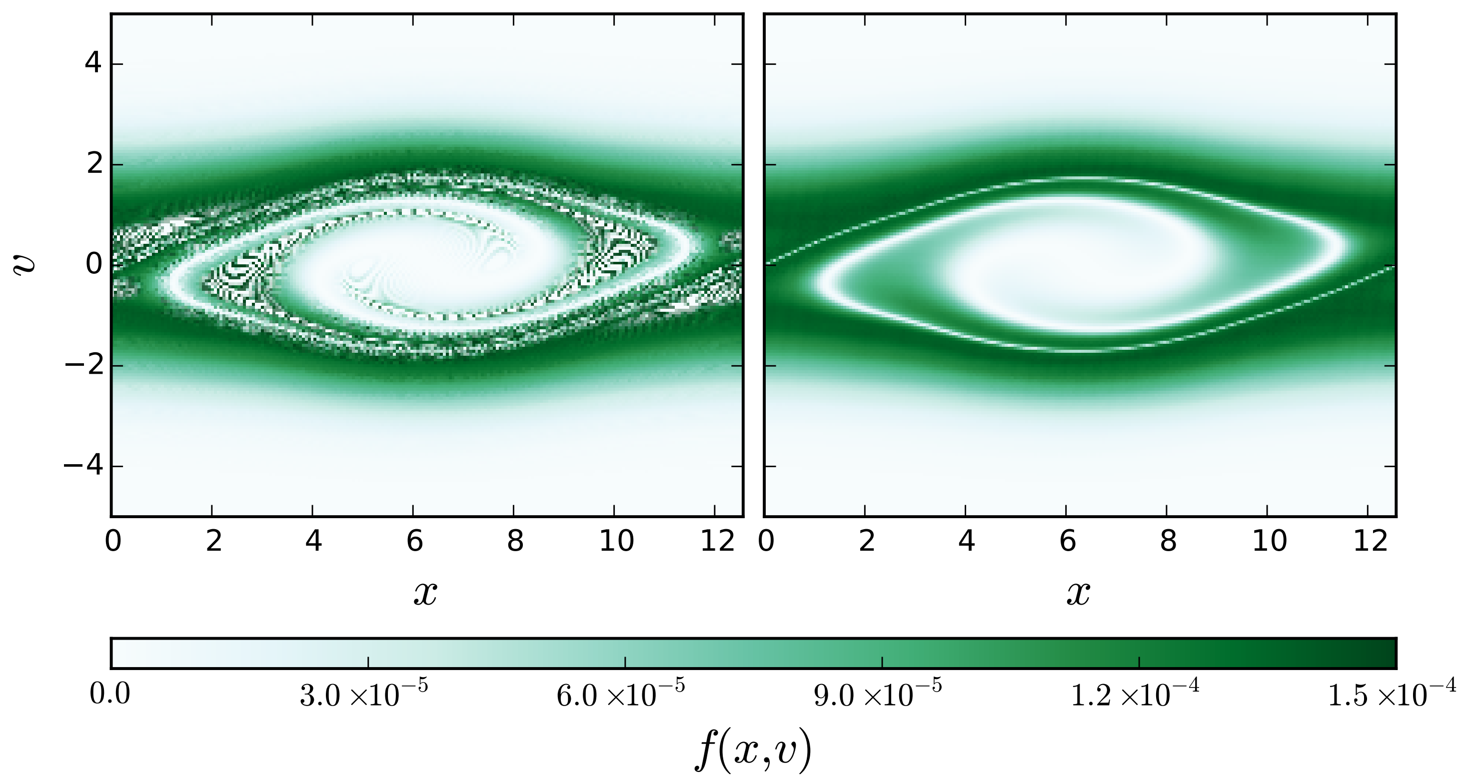
Particle Methods

- Variety of methods - mainly differ in how the force is computed
- PIC methods, in which the force solve is performed on an intermediate grid, are particularly popular.
- Advantages:
  - Mathematically simpler. Reduces a PDE to a set of coupled ODEs.
  - Naturally adaptive - the particles go where more resolution is needed.
  - Usually lower computational cost

Disadvantages of Particle Methods

- Less mathematical guidance
- Particle noise prevents convergence
- Usually limited to 2nd order accuracy

Wang+2011



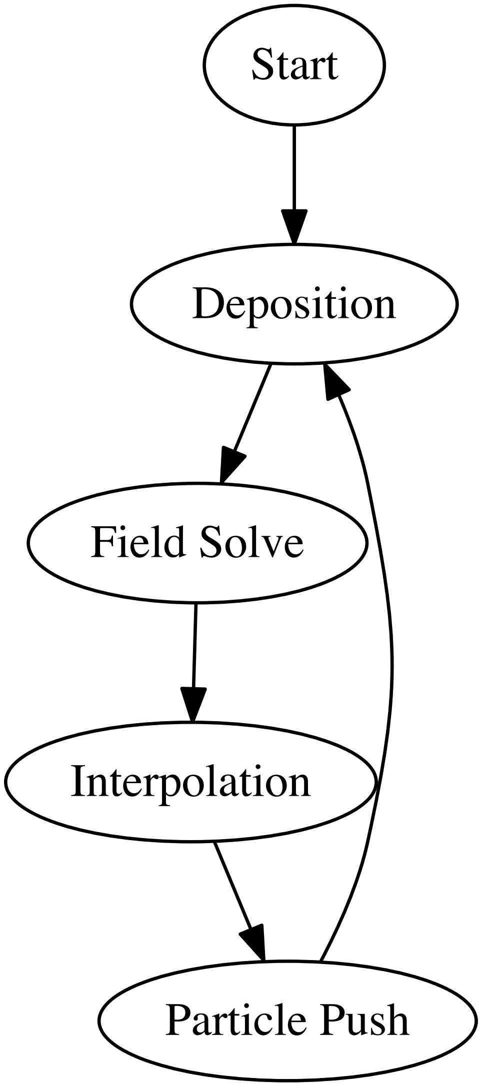
The goal of this talk

- Describe a particle method (PIC) that attempts to address these downsides.
- Show how to all the PIC stages at 4th-order
  - Heart of this is really the interpolation kernels.
- Describe a high-order remapping procedure to ameliorate particle noise while maintaining 4th order accuracy

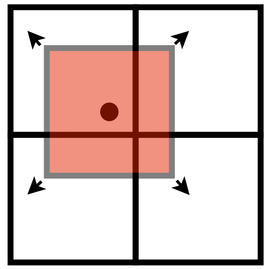
Talk Outline

- Background and motivation
- Description of our numerical method
- Implementation using the Chombo framework
- Numerical results
- Summary and future research directions

Overview of a Particle-in-Cell time step



- Deposition: Particle charges are deposited onto mesh:



$$\rho_{\boldsymbol{i}}^{n+1} = \sum_p \left(\frac{q_p}{\Delta x}\right) \boldsymbol{W} \left(\frac{\boldsymbol{x}_i - \boldsymbol{x}_p^{n+1}}{\Delta x}\right)$$

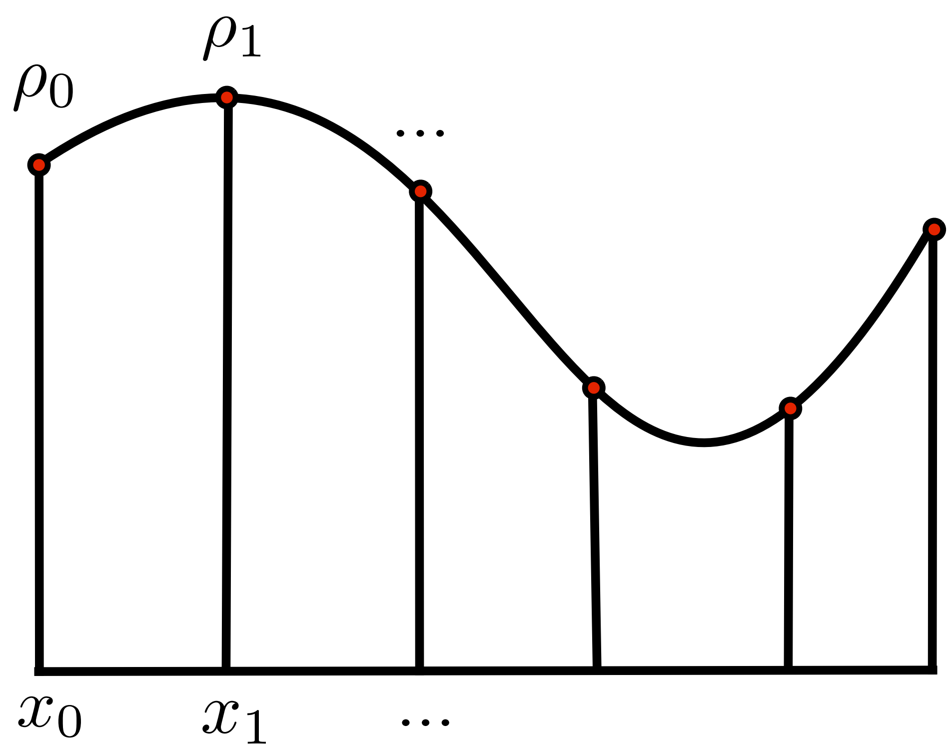
2nd order: Piecewise linear, Cloud-in-Cell interpolation

- Field solve: force is computed on the mesh by e.g. solving Poisson's Equation w/ 2nd order finite differences.
- Interpolation: Force is interpolated back to particle positions using same kernel.
- Particle Push: Particle positions and velocities are updated. 2nd-order leapfrog.

High-order in space (interpolation kernels)

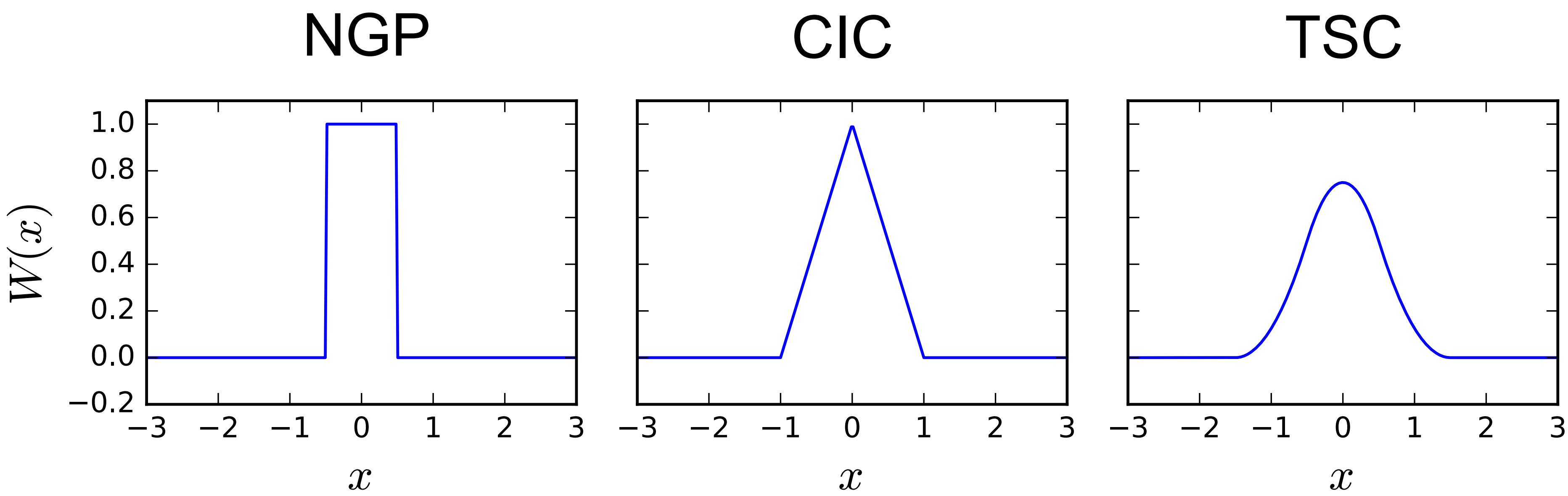
- Given solution at a set of points, reconstruct in between
- Functions W(x) need to be even, normalized, have compact support

$$\rho(x) \approx \sum_j W\left(\frac{x_j - x}{\Delta x}\right)\rho_j$$



High-order in space (interpolation kernels)

- Commonly taken to be one of the B-spline functions. Limited to 2nd Order.



$$M_1(x) = \begin{cases} 1, & 0 \leq |x| \leq 1/2, \\ 0 & \text{otherwise.} \end{cases}$$

$$M_2(x) = \begin{cases} 1 - |x|, & 0 \leq |x| \leq 1, \\ 0 & \text{otherwise.} \end{cases}$$

$$M_3(x) = \begin{cases} \frac{3}{4} - |x|^2, & 0 \leq |x| \leq 1/2, \\ \frac{1}{2}\left(\frac{3}{2} - |x|\right)^2, & 1/2 \leq |x| \leq 3/2, \\ 0 & \text{otherwise.} \end{cases}$$

COMPUTATIONAL RESEARCH DIVISION

BERKELEY LAB

High-order in space (interpolation kernels)

- Error analysis involves looking at the Fourier transform of $W(x)$, $\tilde{W}(k)$
- Can show that $\tilde{W}(k) - 1$ must have a zero of order n at $k = 0$ AND $\tilde{W}(k)$ must have zeros of order n at $k = 2\pi, 4\pi, \ldots$ to be O(n).
- For the Basis spline of order n, $\tilde{W}(k) = \left(\frac{\sin(k/2)}{k/2}\right)^n$
- Fulfills the second requirement, but zero at k = 0 is only order 2.
- Hence, all the basis splines interpolate with at best 2nd order accuracy
- They do, however, have increasing degrees of smoothness which may be desirable when the particles are highly disordered.
- See Schoenberg 1975, Monaghan 1985 for more details.

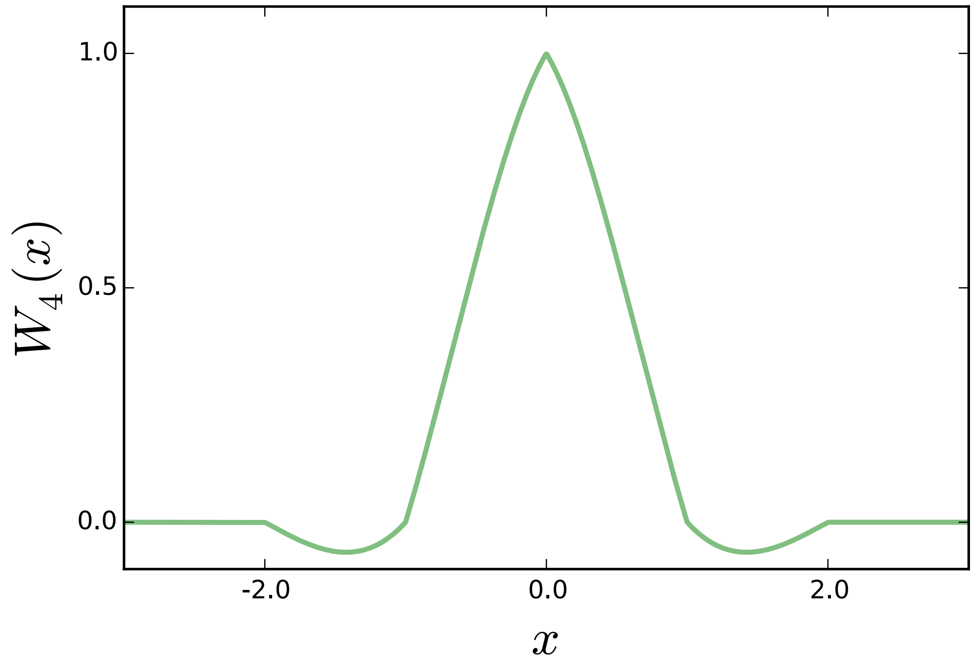
High-order in space (interpolation kernels)

- However, it is possible to design a polynomial with the appropriate zeros in Fourier space and transform back into real space to obtain a $W(x)$ with the desired properties.
- For details, see Lo, Minden, and Colella 2016.

High-order in space (interpolation kernels)

$$W_4(x) = \begin{cases} \frac{|x|^3}{2} - |x|^2 - \frac{|x|}{2} + 1, & |x| \in [0,1], \\ -\frac{|x|^3}{6} + |x|^2 - \frac{11|x|}{6} + 1, & |x| \in [1,2], \\ 0, & \text{else.} \end{cases}$$

- 4-point stencil
- Reproduces cubic functions exactly
- Continuous with discontinuous first derivative



High-order in space (field solve)

- Replace 2nd-order finite difference approximation with 4th-order, centered differences:

$$\nabla^2 \phi = -\rho \quad \rightarrow \quad -\sum_{d=1}^{D} \frac{-\phi_{i+2e^d} + 16\phi_{i+e^d} - 30\phi_i + 16\phi_{i-e^d} - \phi_{i+2e^d}}{12\Delta x^2} = \rho_i$$

$$\boldsymbol{E} = -\nabla\phi \quad \rightarrow \quad \boldsymbol{E}_i^d = -\frac{-\phi_{i+2e^d} + 8\phi_{i+e^d} - 8\phi_{i-e^d} + \phi_{i+2e^d}}{12\Delta x}$$

- Resulting system can be solved with a variety of methods (we used geometric multigrid).

High-order in time (RK4)

- Replace leapfrog with 4th-Order Runge-Kutta method

$$
\begin{aligned}
k_1 &= F\left(t^n, x^n\right) \\
k_2 &= F\left(t^n + \frac{1}{2}\Delta t, x^n + \frac{1}{2}v^n\Delta t + \frac{1}{8}k_1\Delta t^2\right) \\
k_3 &= F\left(t^n + \Delta t, x^n + v^n\Delta t + \frac{1}{2}k_2\Delta t^2\right). \\
x^{n+1} &= x^n + v^n\Delta t + \frac{1}{6}\left(k_1 + 2k_2\right)\Delta t^2 \\
v^{n+1} &= v^n + \frac{1}{6}\left(k_1^n + 4k_2 + k_3\right)\Delta t,
\end{aligned}
$$

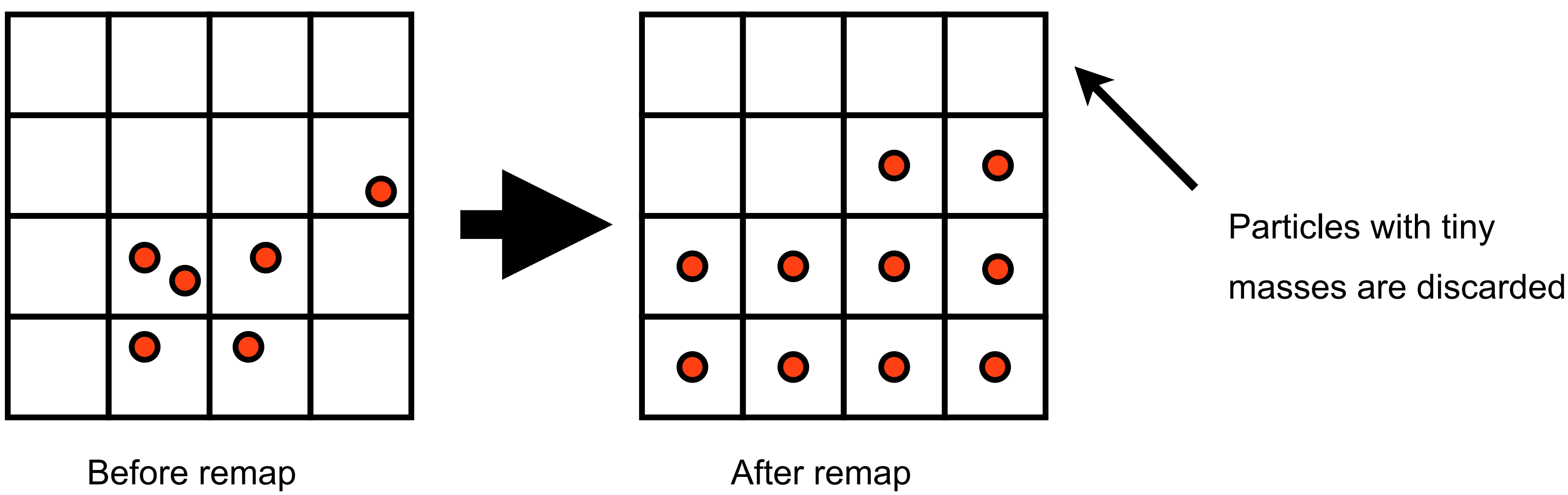
- Requires 3 force solves per time step (velocity independent force), instead of 1
- Self-starting. Gives up on symplectic property.

Particle Remapping

- In PIC convergence theory, stability error for field contains exponential term:

$$e^E(\boldsymbol{x}, t) \propto \exp\left(at\right) \qquad \text{(Wang+2011)}$$

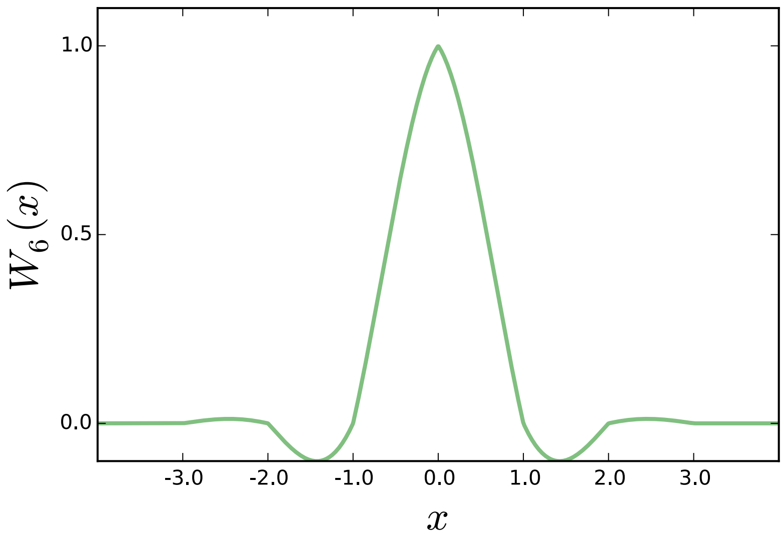
- Periodically restart problem with new particles with interpolated weights



Particle Remapping

$$W_6(x) = \begin{cases} -\frac{|x|^5}{12} + \frac{|x|^4}{4} + \frac{5|x|^3}{12} - \frac{5|x|^2}{4} - \frac{|x|}{3} + 1, & |x| \in [0,1], \\ \frac{|x|^5}{24} - \frac{3|x|^4}{8} + \frac{25|x|^3}{24} - \frac{5|x|^2}{8} - \frac{13|x|}{12} + 1, & |x| \in [1,2], \\ -\frac{|x|^5}{120} + \frac{|x|^4}{8} - \frac{17|x|^3}{24} + \frac{15|x|^2}{8} - \frac{137|x|}{60} + 1, & |x| \in [2,3], \\ 0, & \text{else.} \end{cases}$$

- One order of accuracy is lost during the remap step
- High-order deposition
- 6-point, 6th-order stencil



Particle Remapping

- Particles are laid on at the cell centers of a (possibly AMR) Cartesian grid in phase space. The new weights are then computed as:

$$q_p^* = \sum_p q_p \boldsymbol{W}_6 \left(\frac{\boldsymbol{x}_i - \boldsymbol{x}_p}{h_x} \right) \boldsymbol{W}_6 \left(\frac{\boldsymbol{v}_j - \boldsymbol{v}_p}{h_v} \right)$$

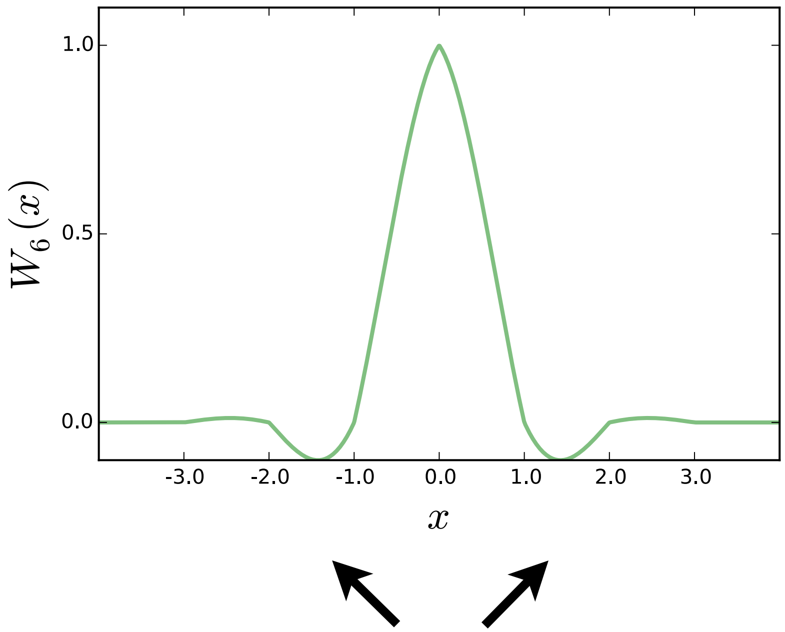
- Particles with small weights are discarded.
- Can be applied every few time steps, or just a few times per calculation
- Can be thought of as Semi-Lagrangian
- Compare the Forward Semi-Lagrangian technique (Crouseilles+2008)

Particle Remapping

- Higher order interpolation functions are not positivity preserving

$$\int_{\mathbb{R}^D} (x-y)^{\ell} W(x-y)\,dy$$

- Can show that the 2nd moment of the interpolating function must vanish for the kernel to have better than 2nd order accuracy.



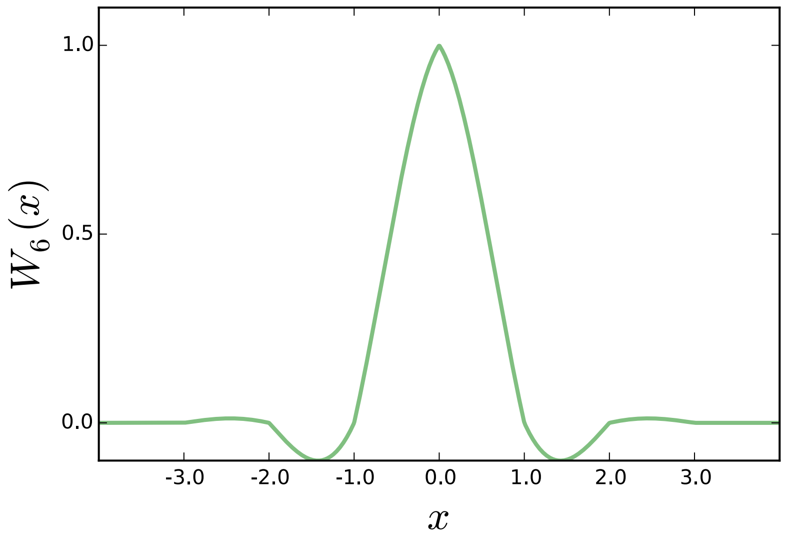
Positivity not guaranteed

Particle Remapping

- Higher order interpolation functions are not positivity preserving

$$\int_{\mathbb{R}^D} (x-y)^{\ell} W(x-y)dy$$

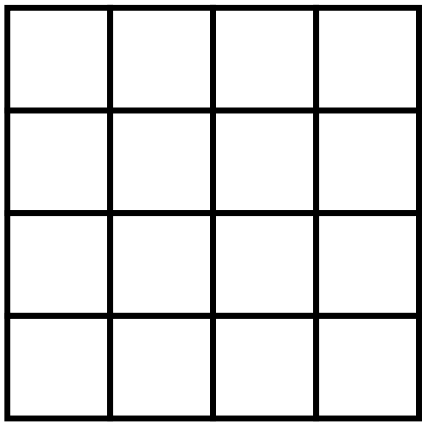
- Use a redistribution procedure to maintain positivity of the distribution function



Positivity not guaranteed

Redistribution procedure

l neighboring cells



$$\delta f_i = \min(0, f_i^n)$$

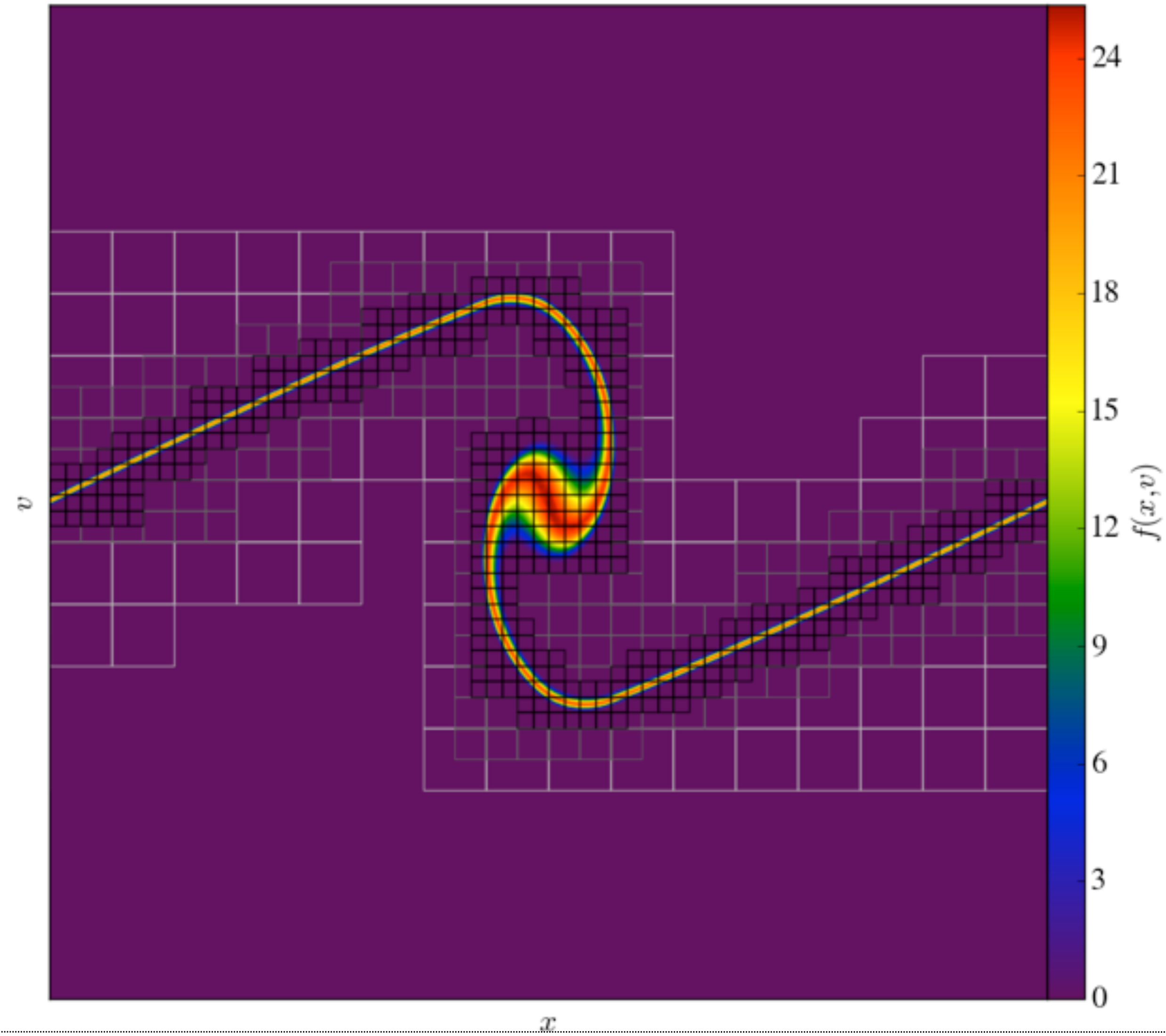
$$\xi_{i+l} = \max(0, f_{i+l}^n)$$

$$f_{i+l}^{n+1} = f_{i+l}^n + \frac{\xi_{i+l}}{\sum_{k \neq 0}^{\text{neighbors}} \xi_{i+k}} \delta f_i$$

- 2 or 3 iterations is usually sufficient

Particle Remapping AMR example

- Example distribution function from a remapping stage, using 4 AMR levels
- Cosmological dark matter test problem (Myers+2015)

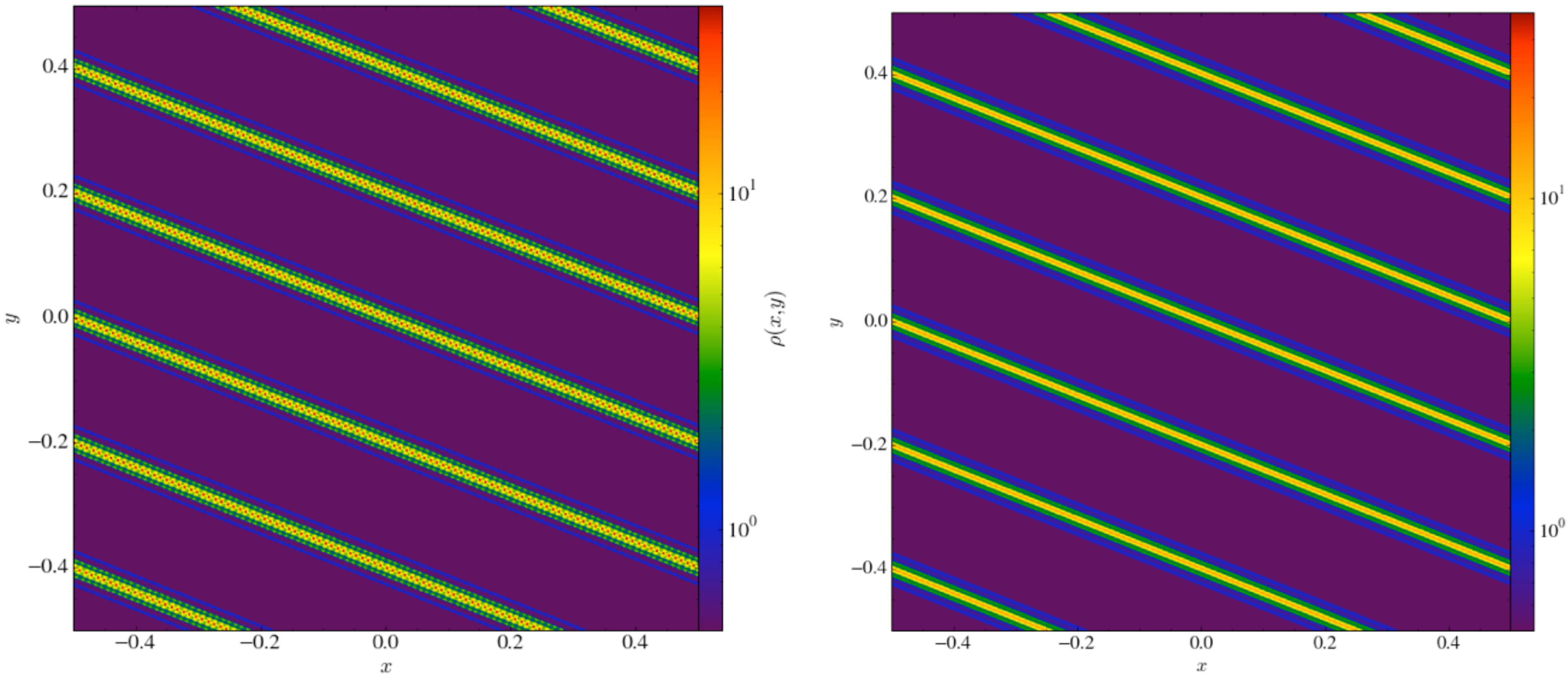


COMPUTATIONAL RESEARCH DIVISION

BERKELEY LAB

Remapping on Cosmology Calculation

Not remapped | Remapped



http://adsabs.harvard.edu/abs/2016ApJ...816...56M

COMPUTATIONAL RESEARCH DIVISION

BERKELEY LAB

Talk Outline

- Background and motivation
- Description of our numerical method
- Implementation using the Chombo framework
- Numerical results
- Summary and future research directions

Implementation

- Chombo - A set of tools for numerically solving PDEs
- Especially useful for AMR applications
- Mixture of C++ and FORTRAN
- Uses a Box-based, SPMD approach to parallelism
- Provides distributed data containers, including (in the next release) for particle data
- Our method was implemented using these tools, running NERSC's Edison machine + my laptop.
- Code, tests, and analysis scripts are all here:

https://bitbucket.org/atmyers/4thorderpic

Talk Outline

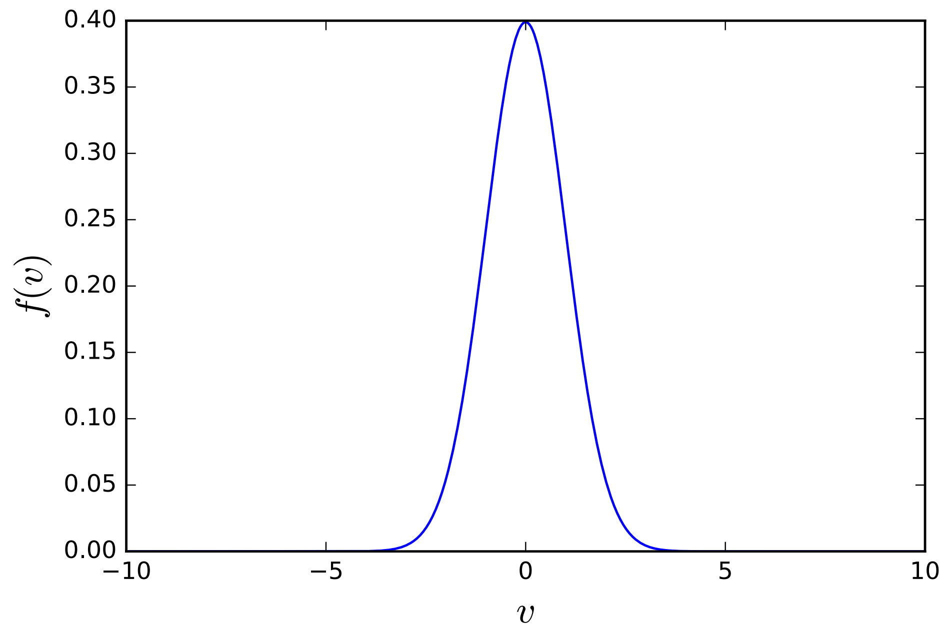
- Background and motivation
- Description of our numerical method
- Implementation using the Chombo framework
- Numerical results
- Summary and future research directions

Linear Landau Damping

- Exponentially damped oscillation of a space charge wave
- Wave-particle interactions cause the electric field wave to lose energy to the particles, damping its amplitude
- Very commonly used test problem for PIC codes
  - Analytic solution for the decay rate
  - Long-time evolution is challenging for classical PIC methods

Linear Landau Damping (initial conditions)

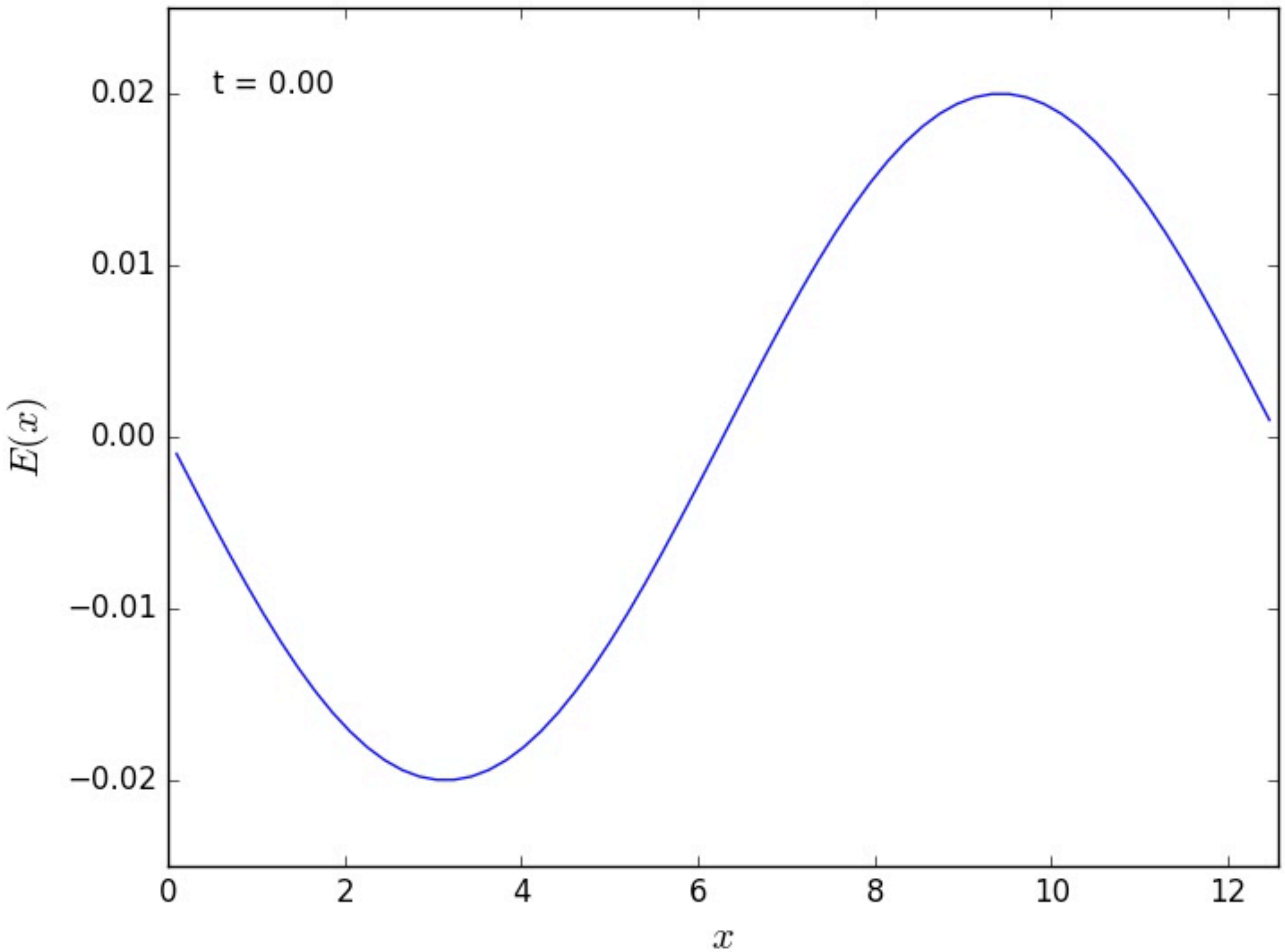
- Gaussian velocity distribution
- Small (linear) perturbation to the charge density (physical space)
- Track the electric field as a function of time



$$f(x, v, t=0) = \frac{1}{\sqrt{2\pi}} \exp\left(-v^2/2\right)\left(1 + \alpha \cos(kx)\right)$$
$$(x, v) = [0, L = 2\pi/k] \times [-v_{\max}, v_{\max}]$$

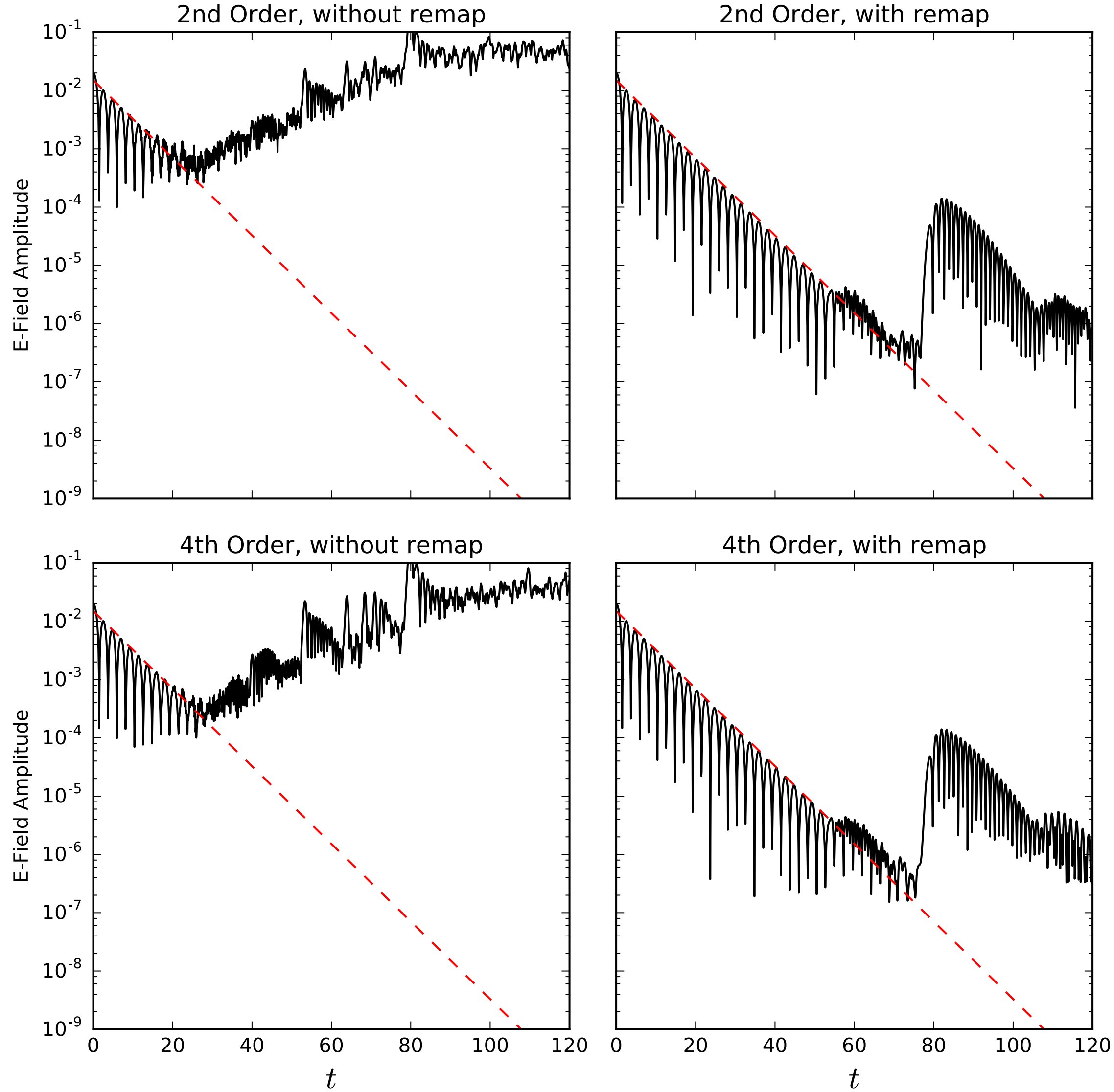
$$k = 0.5$$
$$\alpha = 0.01$$

Linear Landau Damping (movie)



Linear Landau Damping (results)

- Black shows simulation result, red shows damping rate expected from linear theory.
- Remapping is needed to track the analytical damping rate for late times.
- Recurrence effect seen in all simulations past t = 80, a consequence of finite cell spacing in the velocity dimension

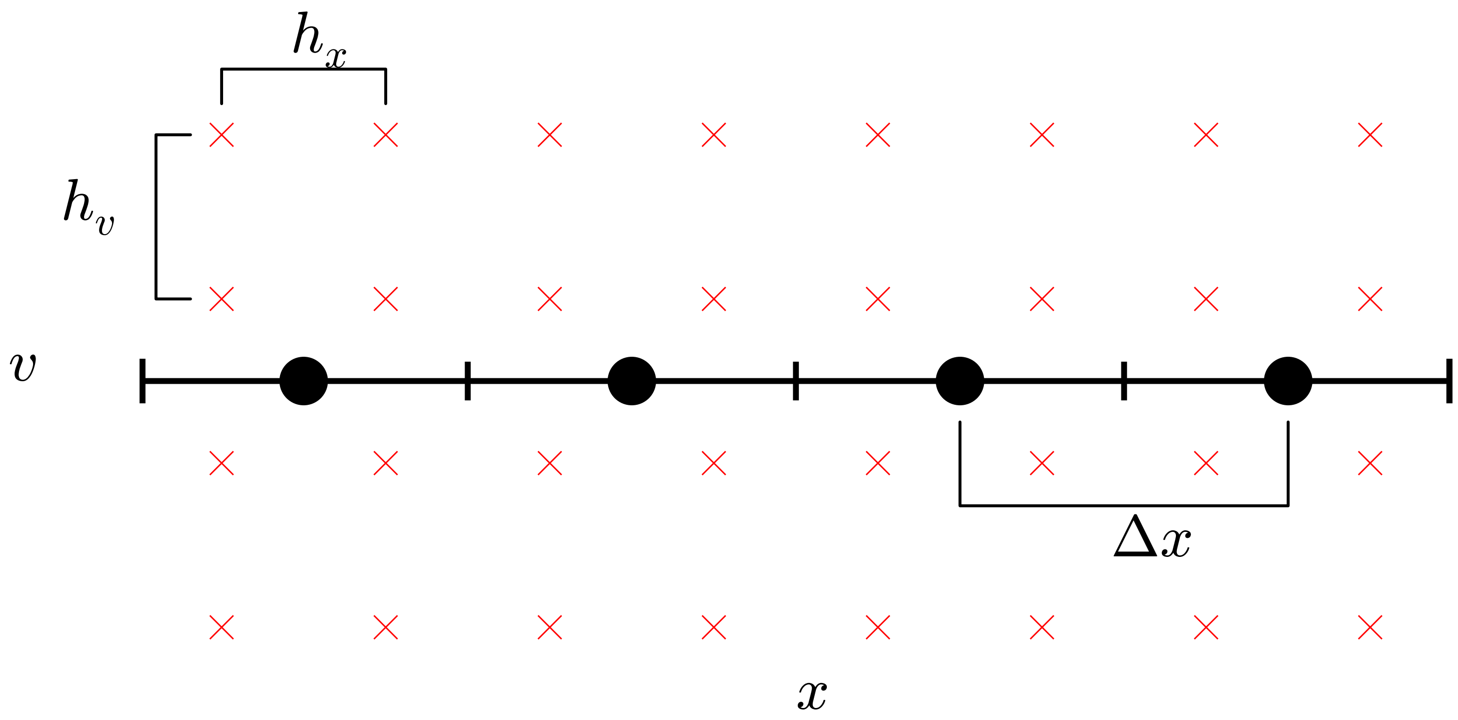


COMPUTATIONAL RESEARCH DIVISION

BERKELEY LAB

Convergence studies for PIC simulations

- All discrete elements of the problem, including Δt, reduced by factor of 2.
- Ratio of $\Delta x / h_x$ remains fixed.



Convergence studies for PIC simulations

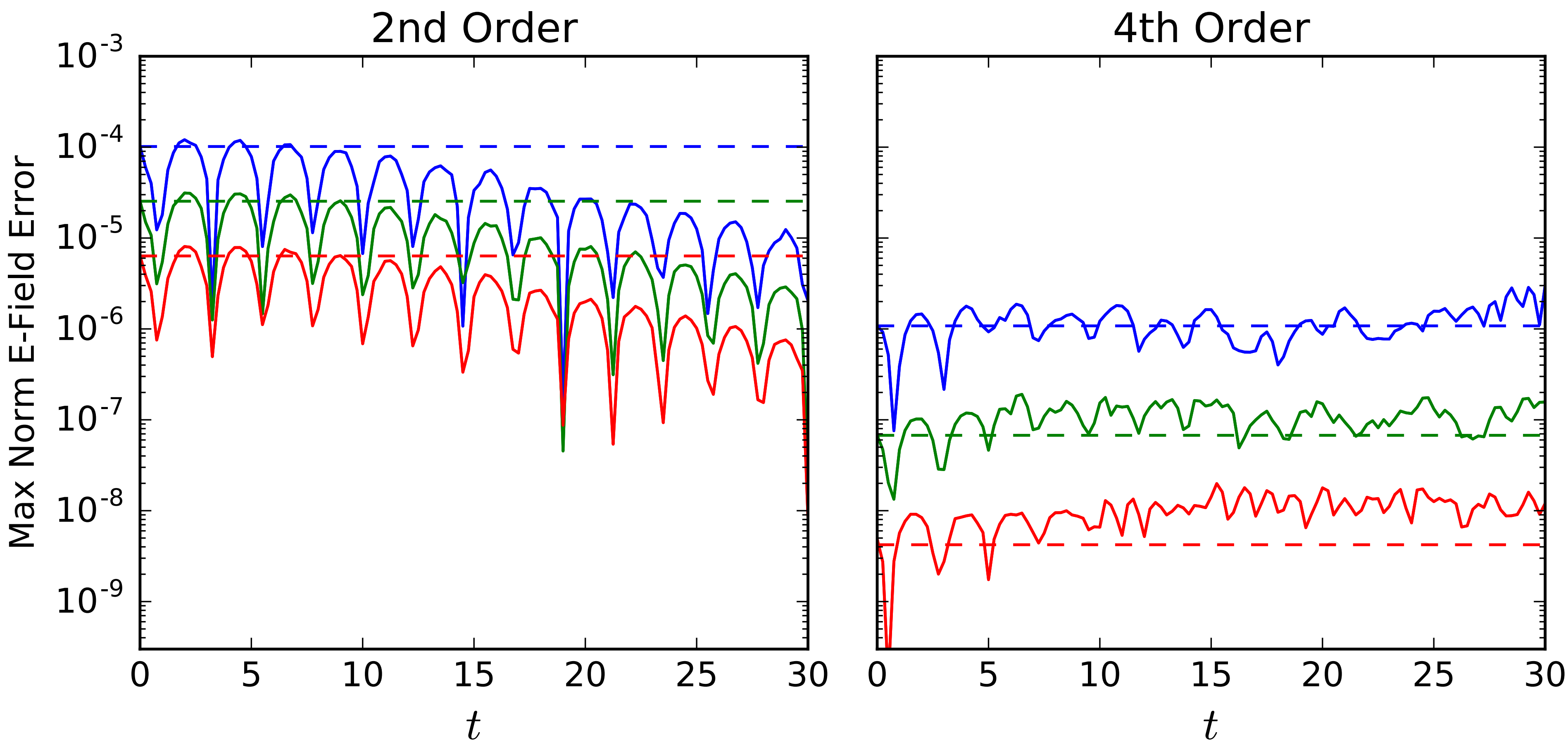
- We then use Richardson extrapolation to compute the error.
- The error in the electric field is then defined as:

$$e^h = |\boldsymbol{E}^h - \boldsymbol{E}^{2h}|$$

- And the order is computed as:

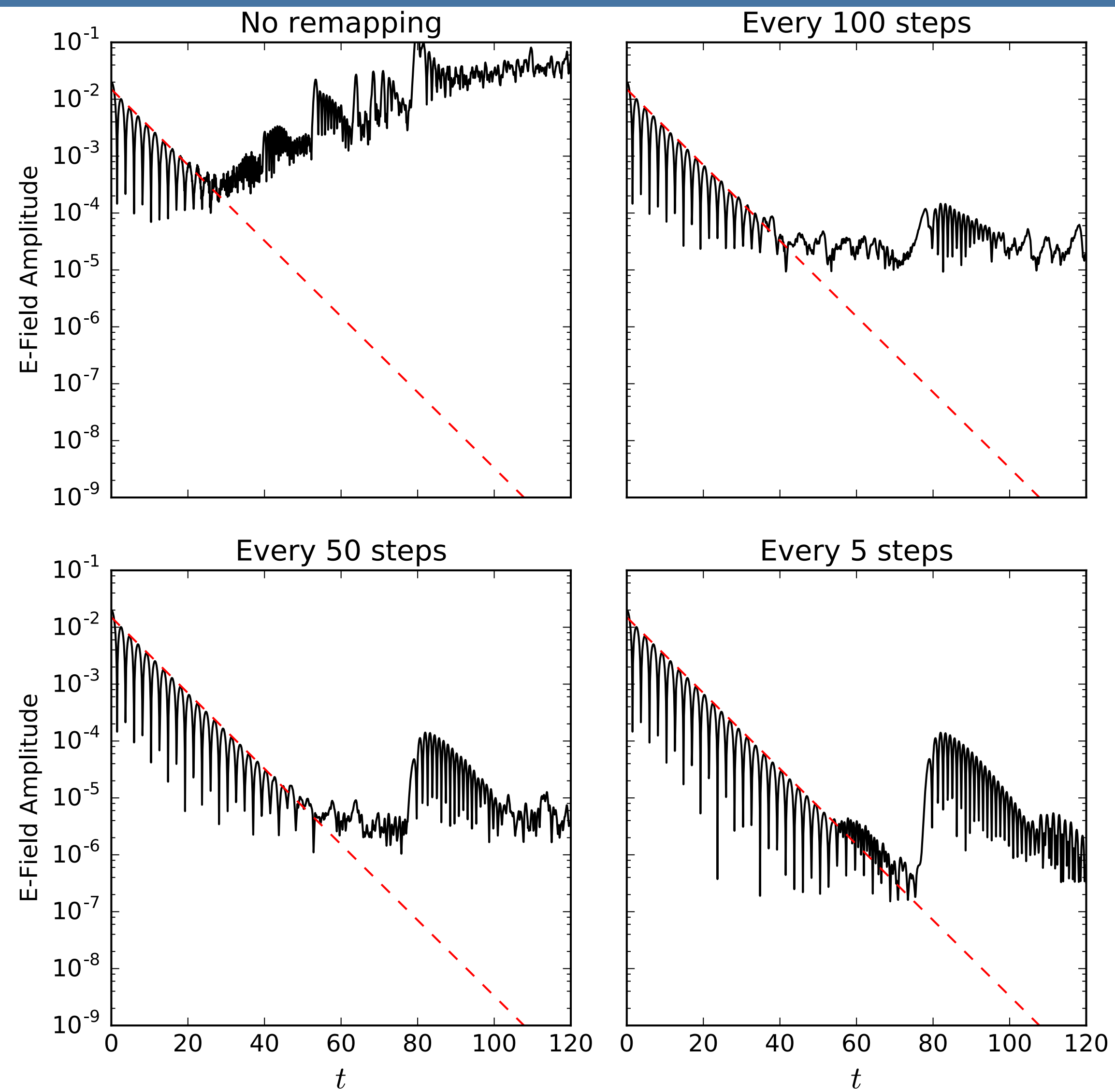
$$q = \log_2\left(\frac{||e^{2h}||}{||e^h||}\right)$$

Linear Landau Damping (convergence results)



Linear Landau Damping (results)

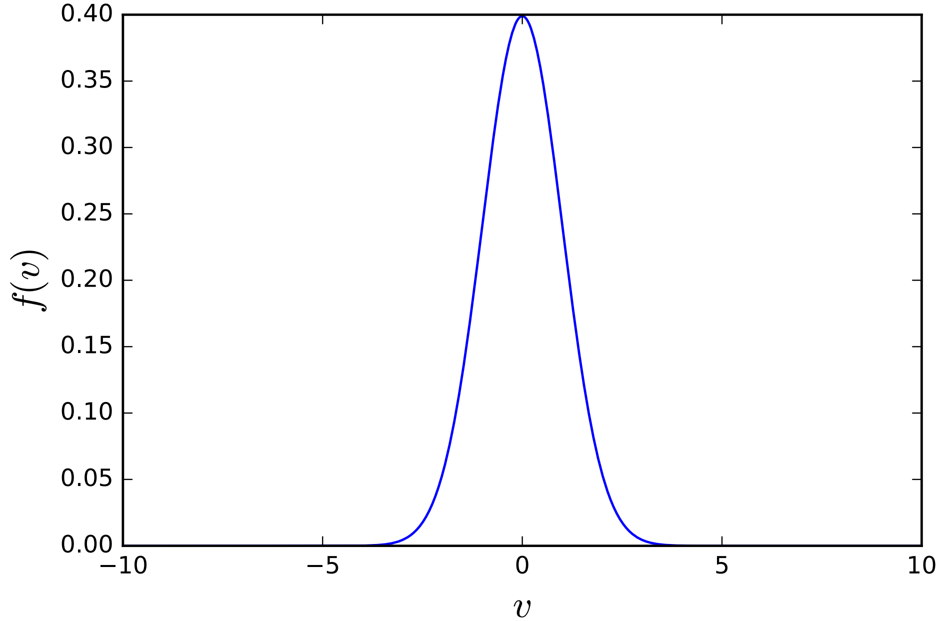
Remapping every 5 steps gives the best results, but even every 100 helps



Nonlinear Landau Damping (initial conditions)

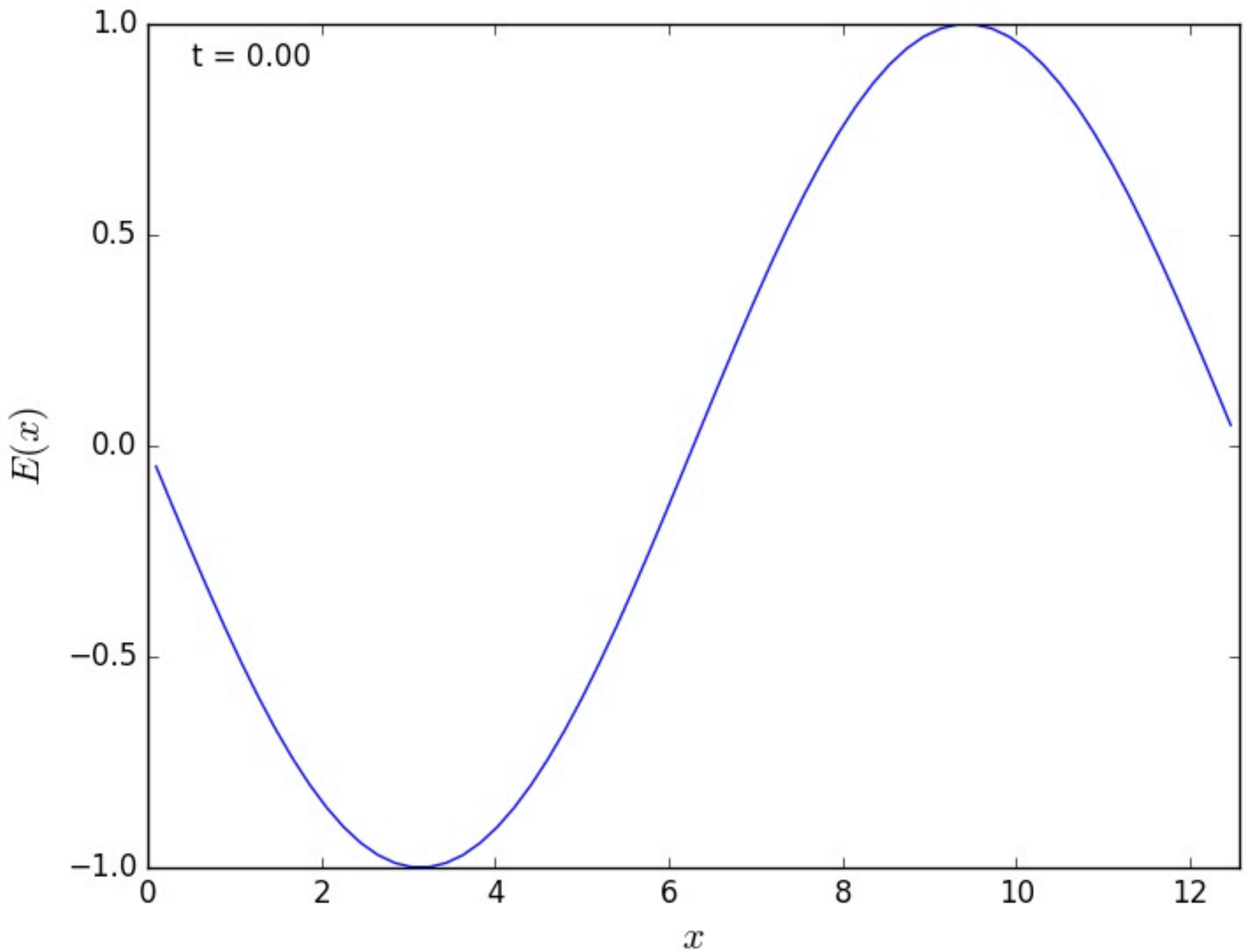
$$f(x, v, t=0) = \frac{1}{\sqrt{2\pi}} \exp\left(-v^2/2\right)\left(1 + \alpha \cos\left(kx\right)\right)$$

$$(x, v) = [0, L = 2\pi/k] \times [-v_{\max}, v_{\max}]$$



$$k = 0.5$$
$$\alpha = 0.5$$

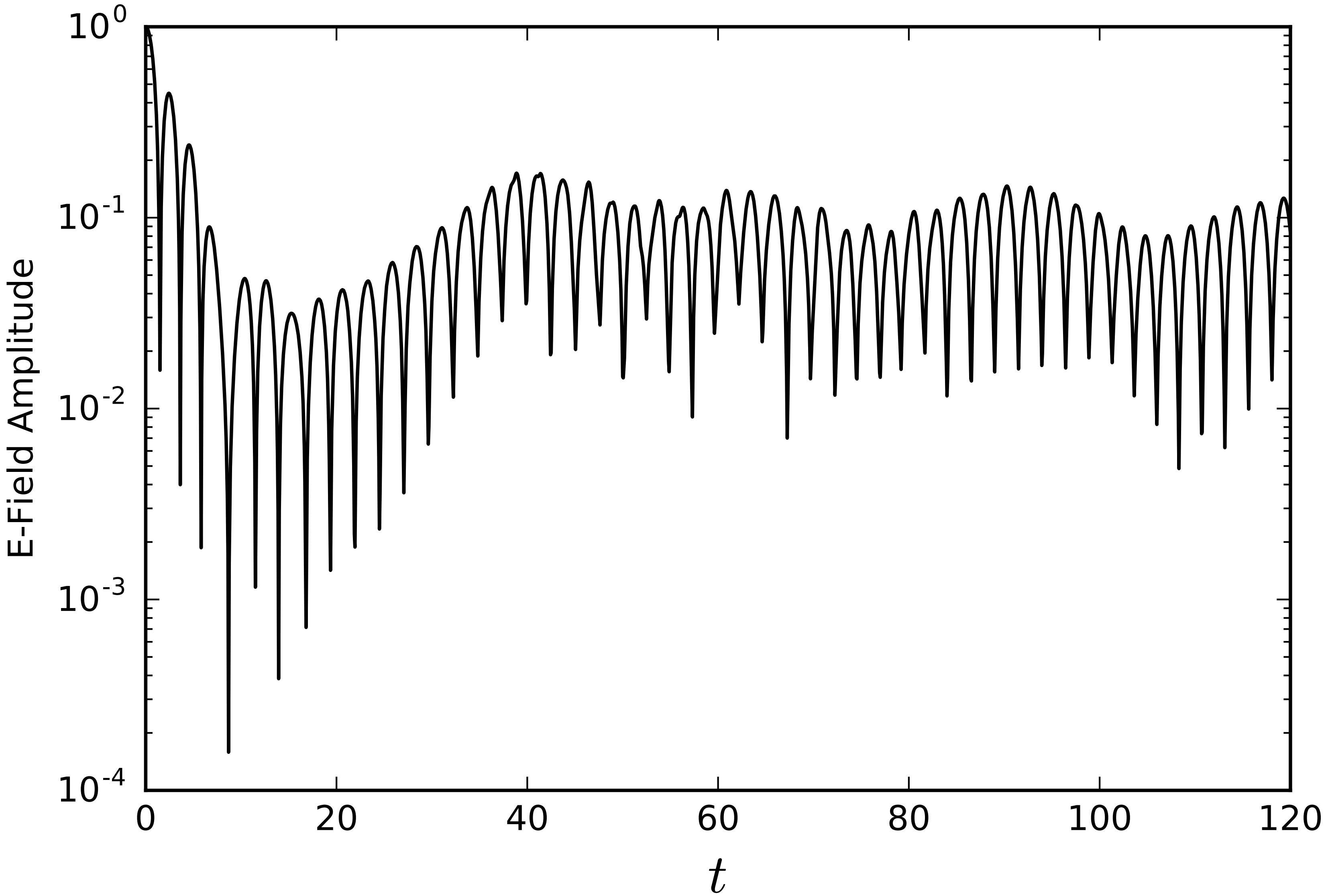
Nonlinear Landau Damping (movie)



COMPUTATIONAL RESEARCH DIVISION

BERKELEY LAB

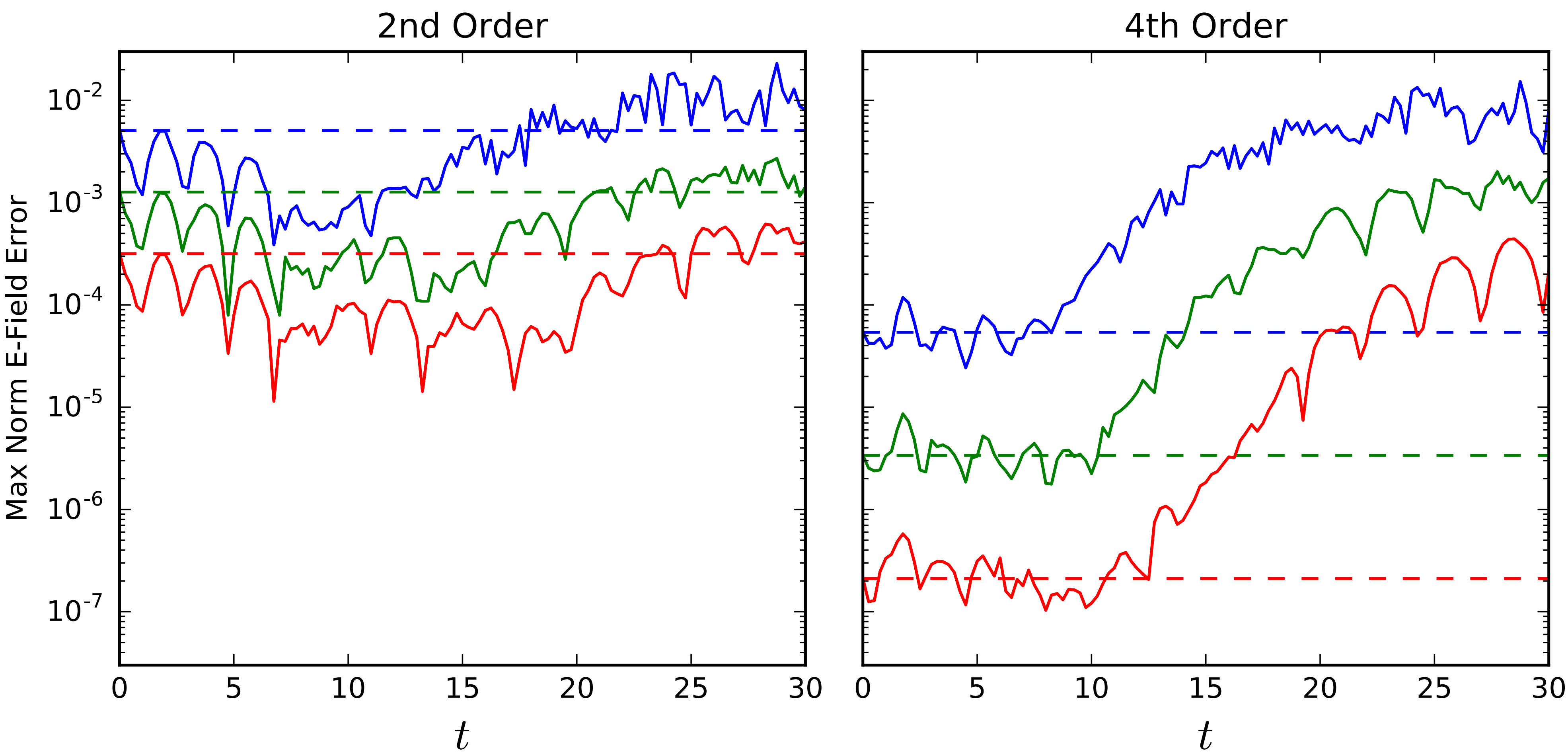
Nonlinear Landau Damping (results)



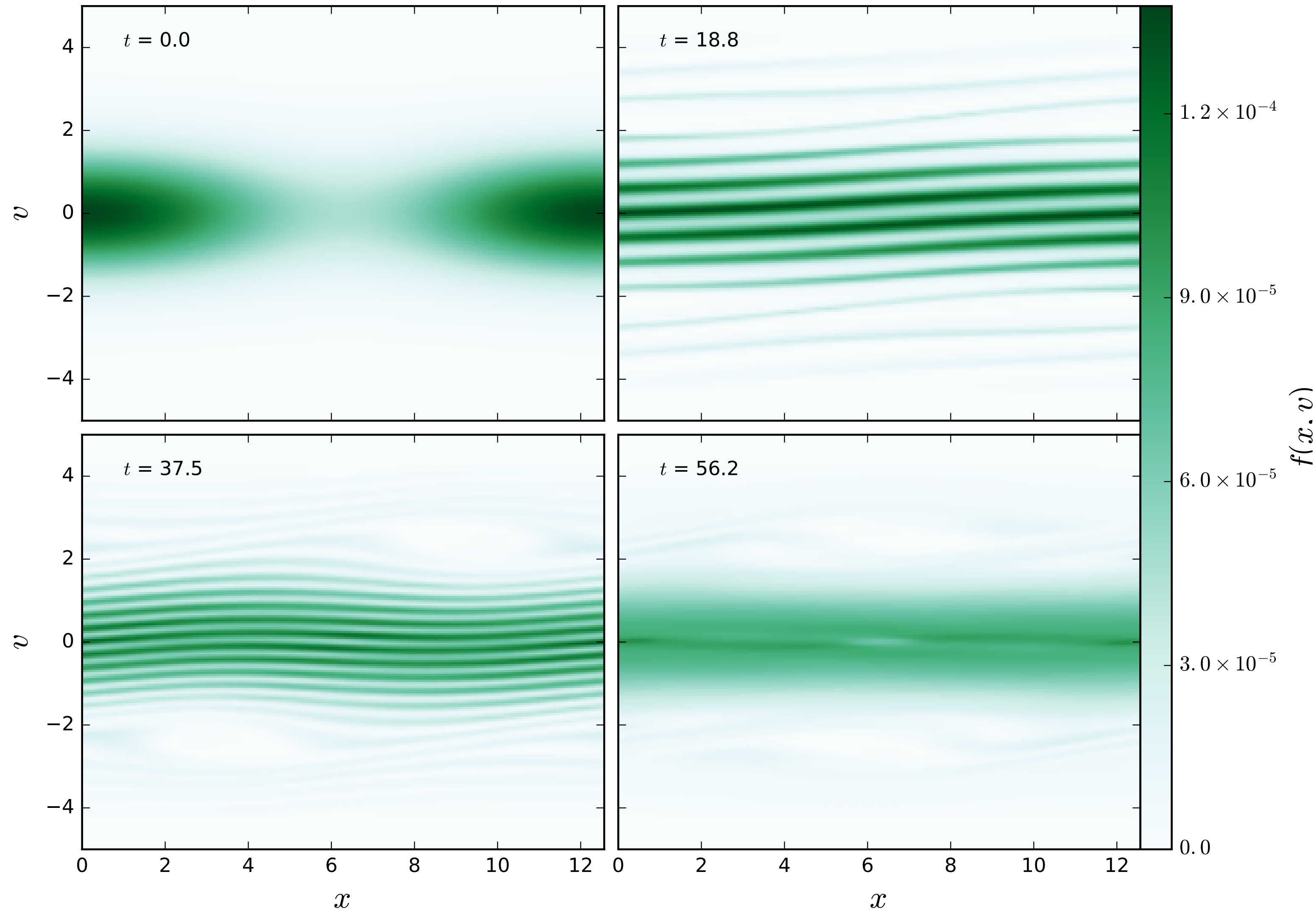
COMPUTATIONAL RESEARCH DIVISION

BERKELEY LAB

Nonlinear Landau Damping (convergence)



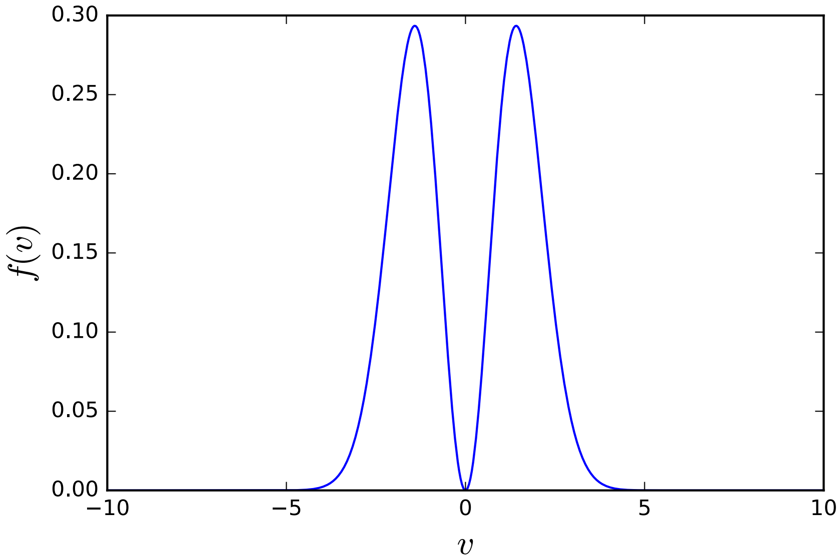
Nonlinear Landau Damping (Phase Diagrams)



Two-Stream Instability

- Initial distribution function corresponds to two particle streams with opposing velocities:

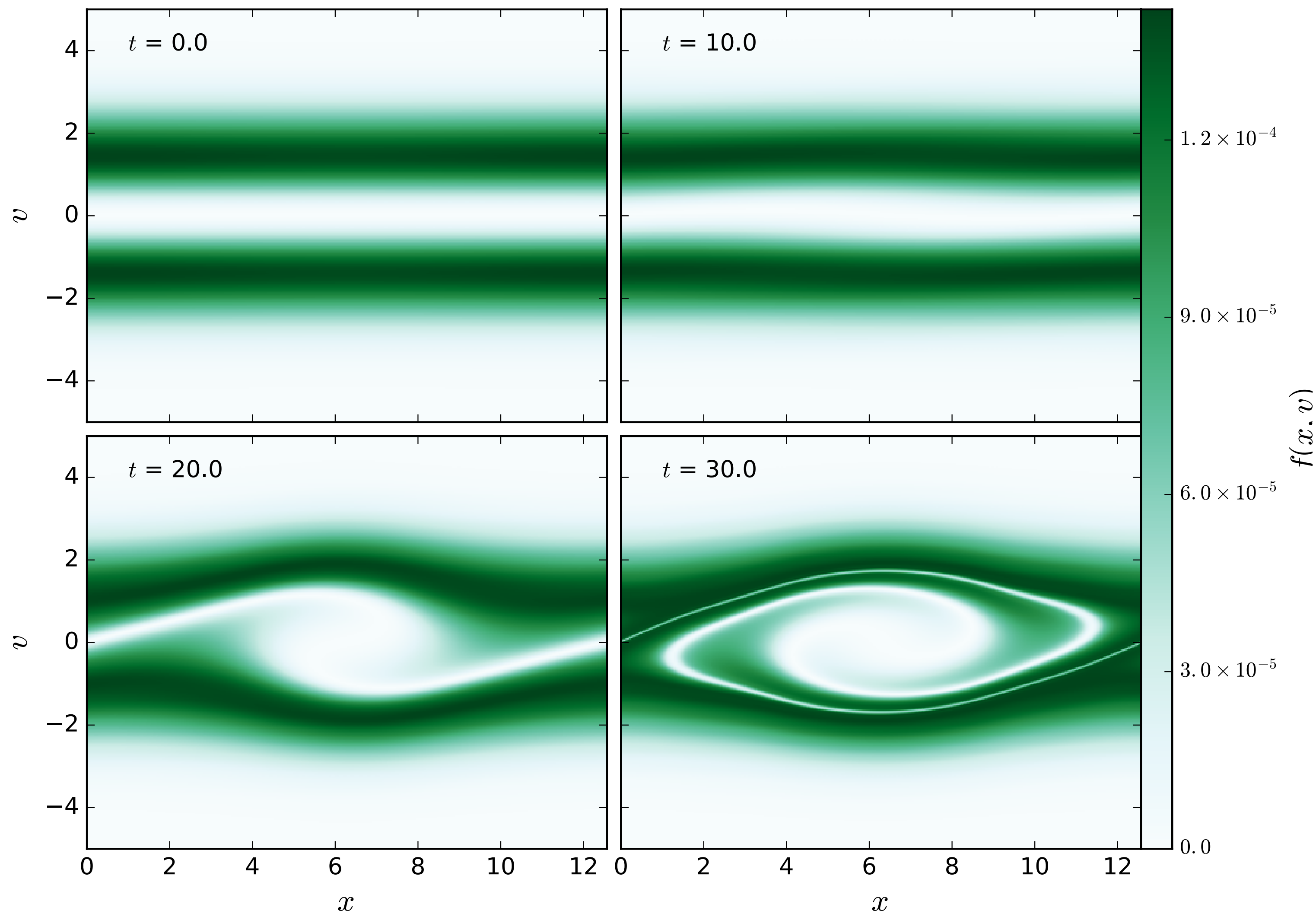
$$f(x,v,t=0) = \frac{1}{\sqrt{2\pi}} v^2 \exp\left(-v^2/2\right)\left(1+\alpha\cos\left(kx\right)\right)$$
$$(x,v) = [0, L=2\pi/k] \times [-v_{\max}, v_{\max}]$$



$$k = 0.5$$
$$\alpha = 0.01$$

"Inverse" of Landau Damping

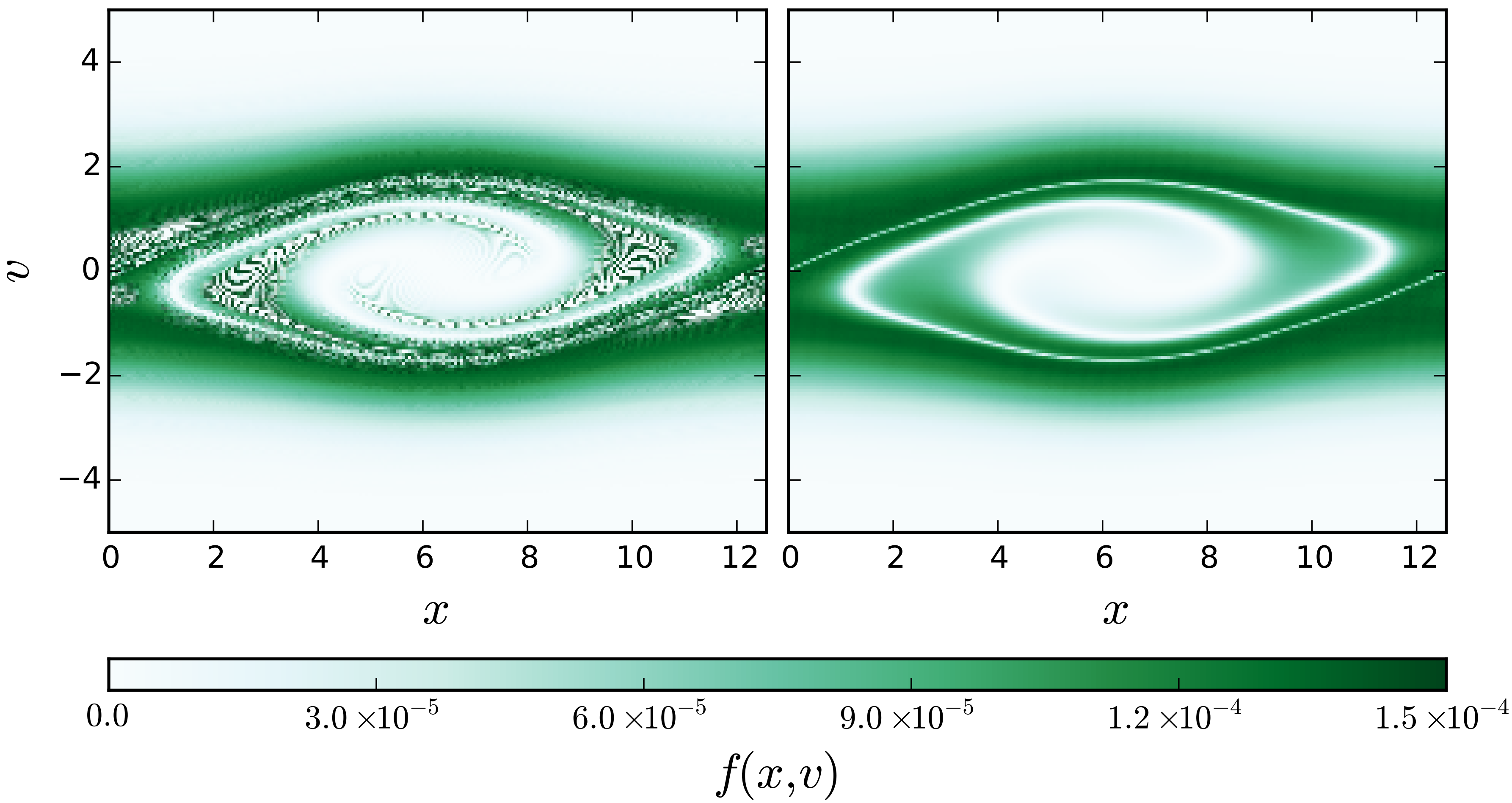
Two-Stream Instability



COMPUTATIONAL RESEARCH DIVISION

BERKELEY LAB

Two-Stream Instability

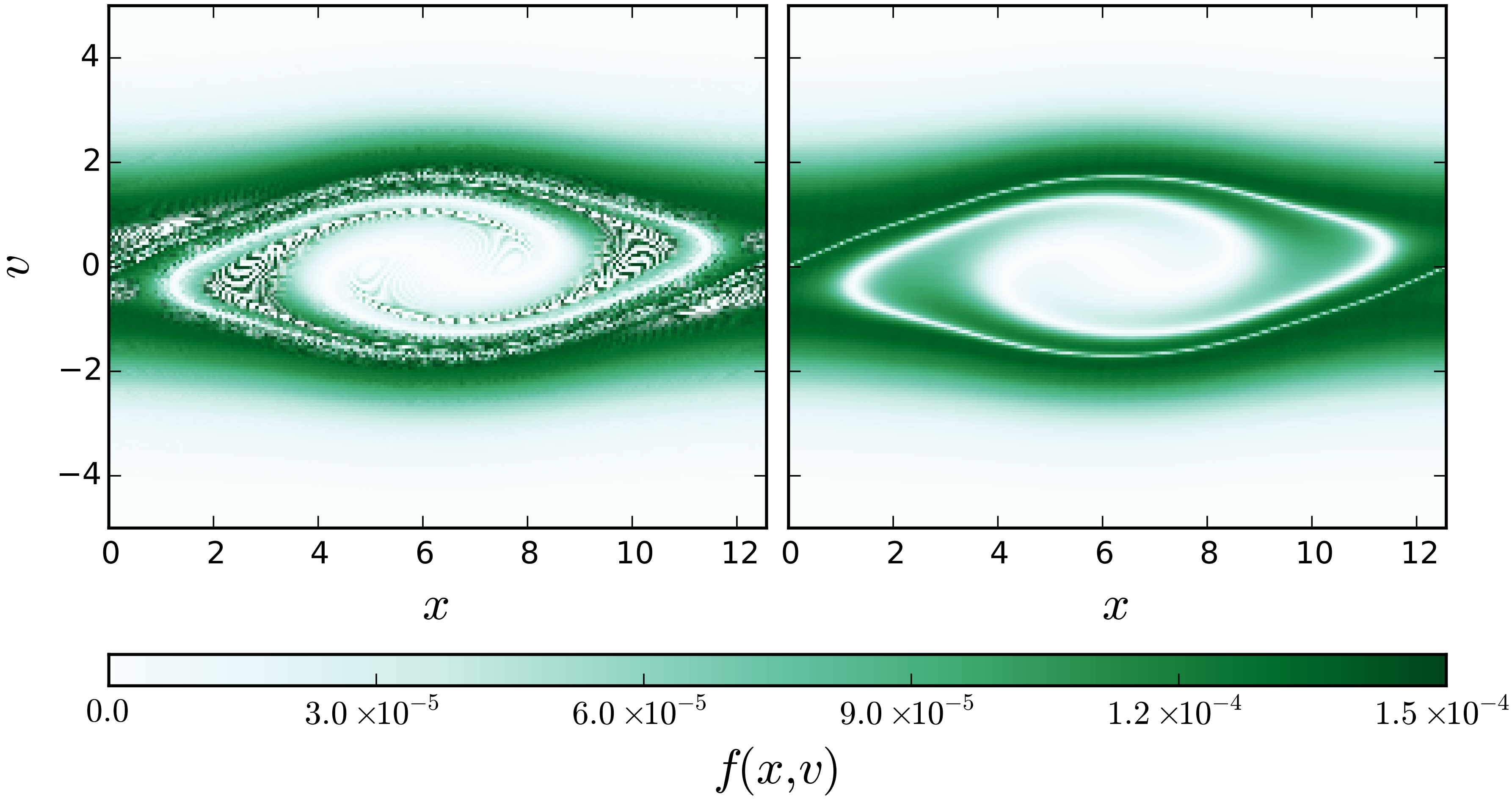


COMPUTATIONAL RESEARCH DIVISION

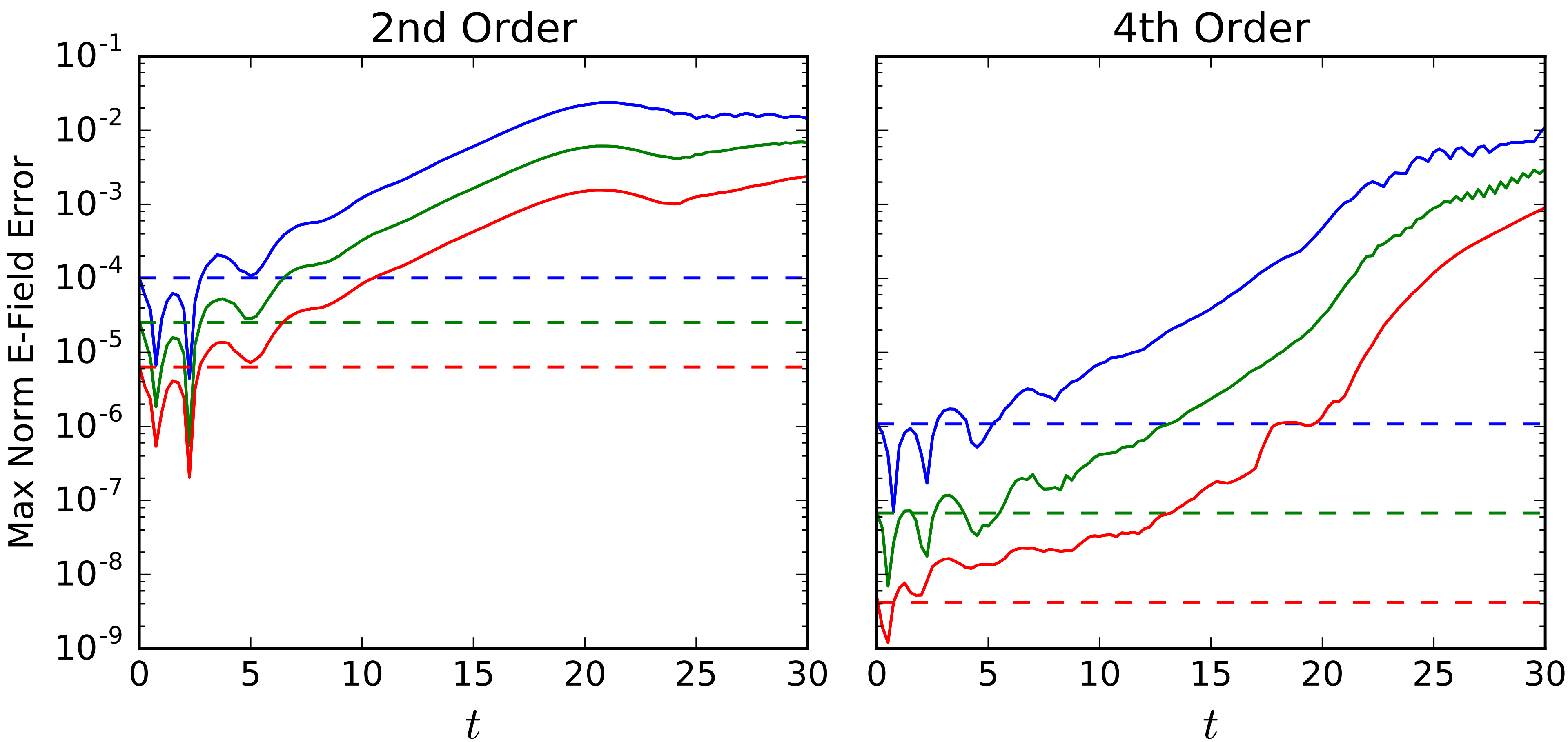
BERKELEY LAB

Two-Stream Instability

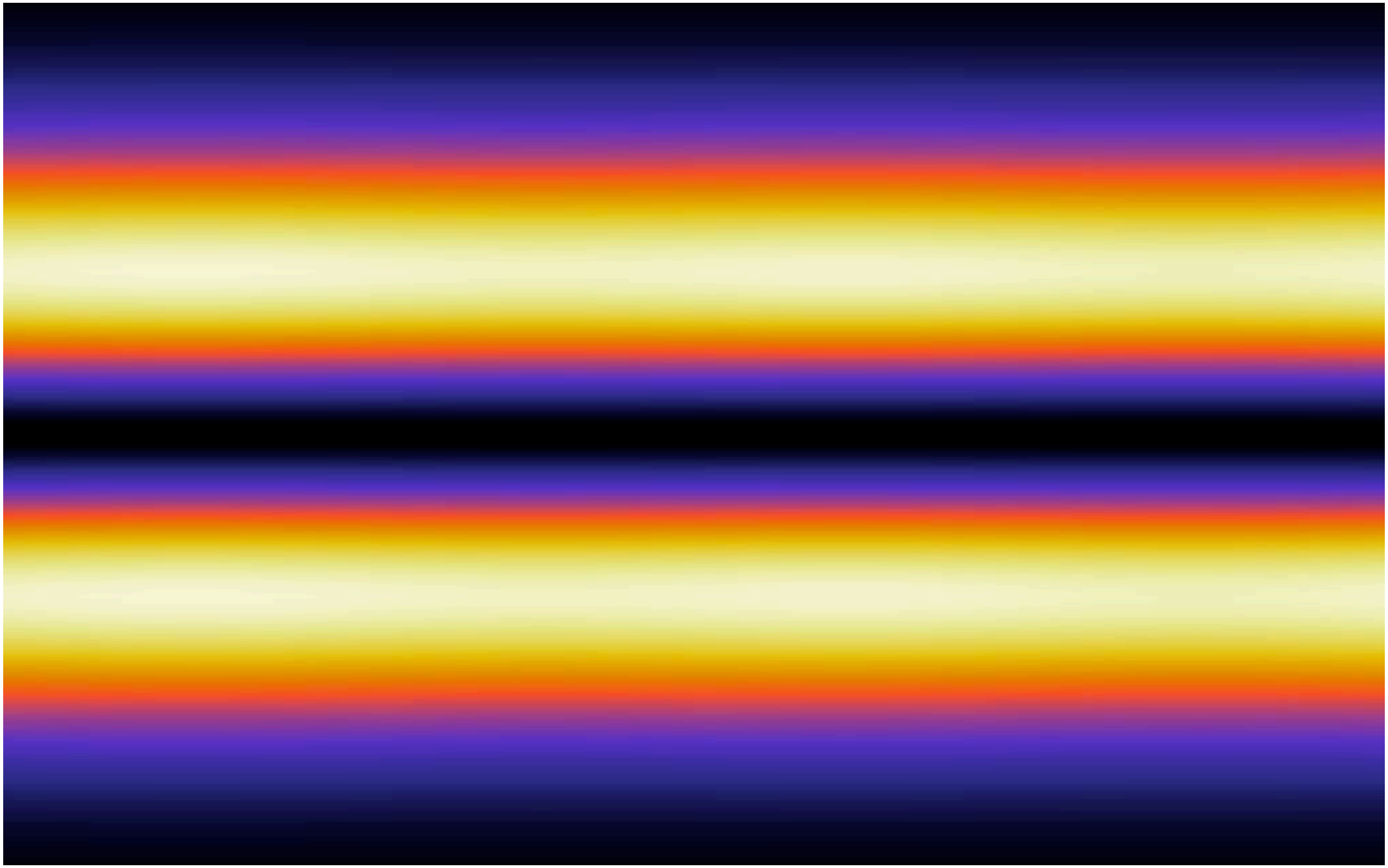
Remapping 10 times (out of ~1000)



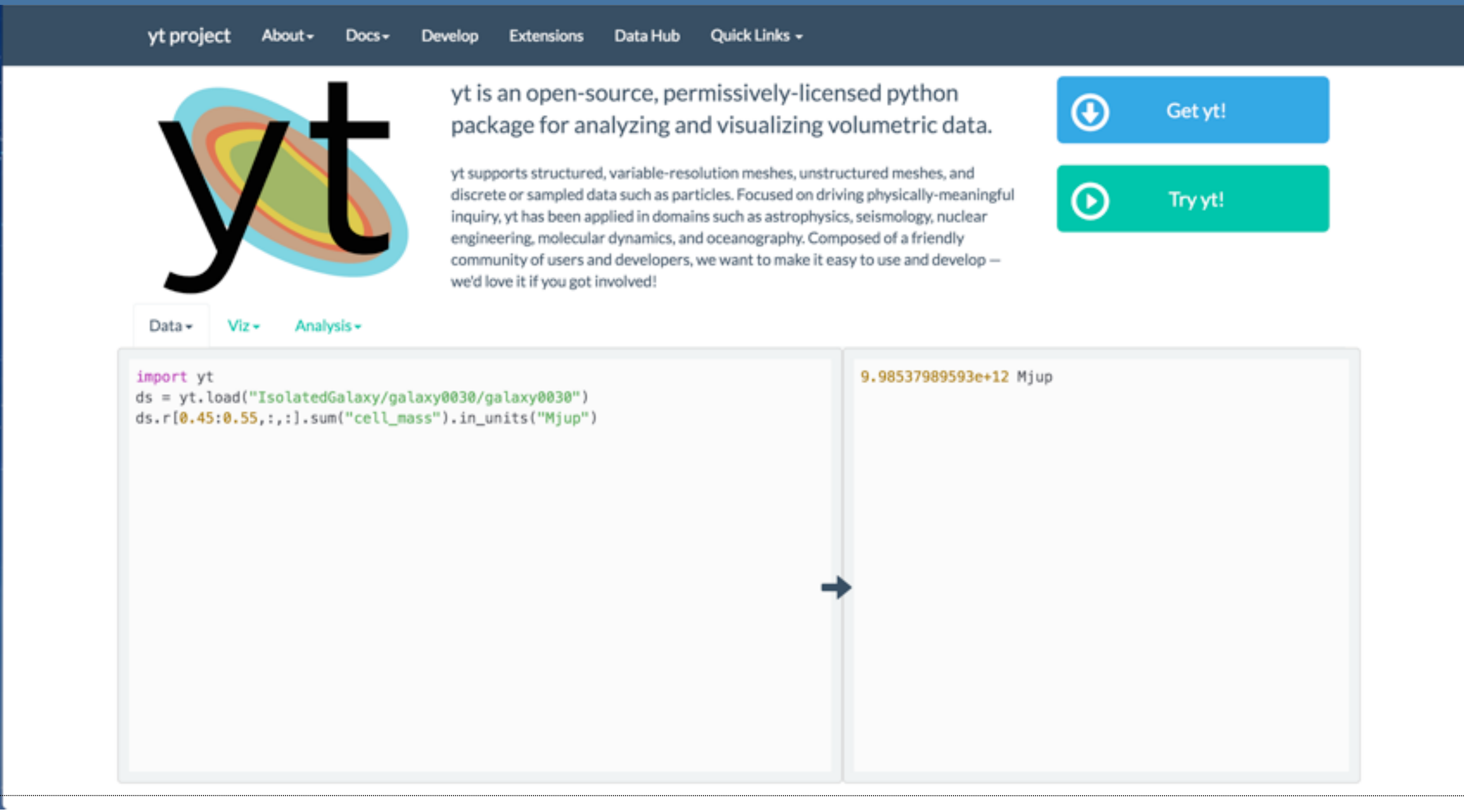
Two-Stream Instability (convergence results)



Two-Stream Instability



http://yt-project.org/



Talk Outline

- Background and motivation
- Description of our numerical method
- Implementation using the Chombo framework
- Numerical results
- Summary and future research directions

Summary and Future Research

- High-order method is more accurate in regions where the flow is smooth
- Not more necessarily more accurate for thin or unresolved phase-space features in the nonlinear regime.
- Remapping is necessary to control particle noise over long time evolutions
- Involves some degree of retreat from pure Lagrangian. AMR helps to an extent.Trade-offs need to be explored.
- Future research involves performing the remap selectively rather than globally.
- Coupled with high-order Maxwell solver for electromagnetic PIC (Sven Chilton)

BERKELEY LAB Lawrence Berkeley National Laboratory

U.S. DEPARTMENT OF ENERGY

Thanks for listening!

Andrew Myers
atmyers2@gmail.com
Applied Numerical Algorithms Group, Computational Research Division

with Phillip Colella, Brian Van Straalen, Bei Wang, Boris Lo, Victor Minden

Advanced Modeling Seminar Series
NASA Ames Research Center
November 10th, 2016

https://arxiv.org/abs/1602.00747