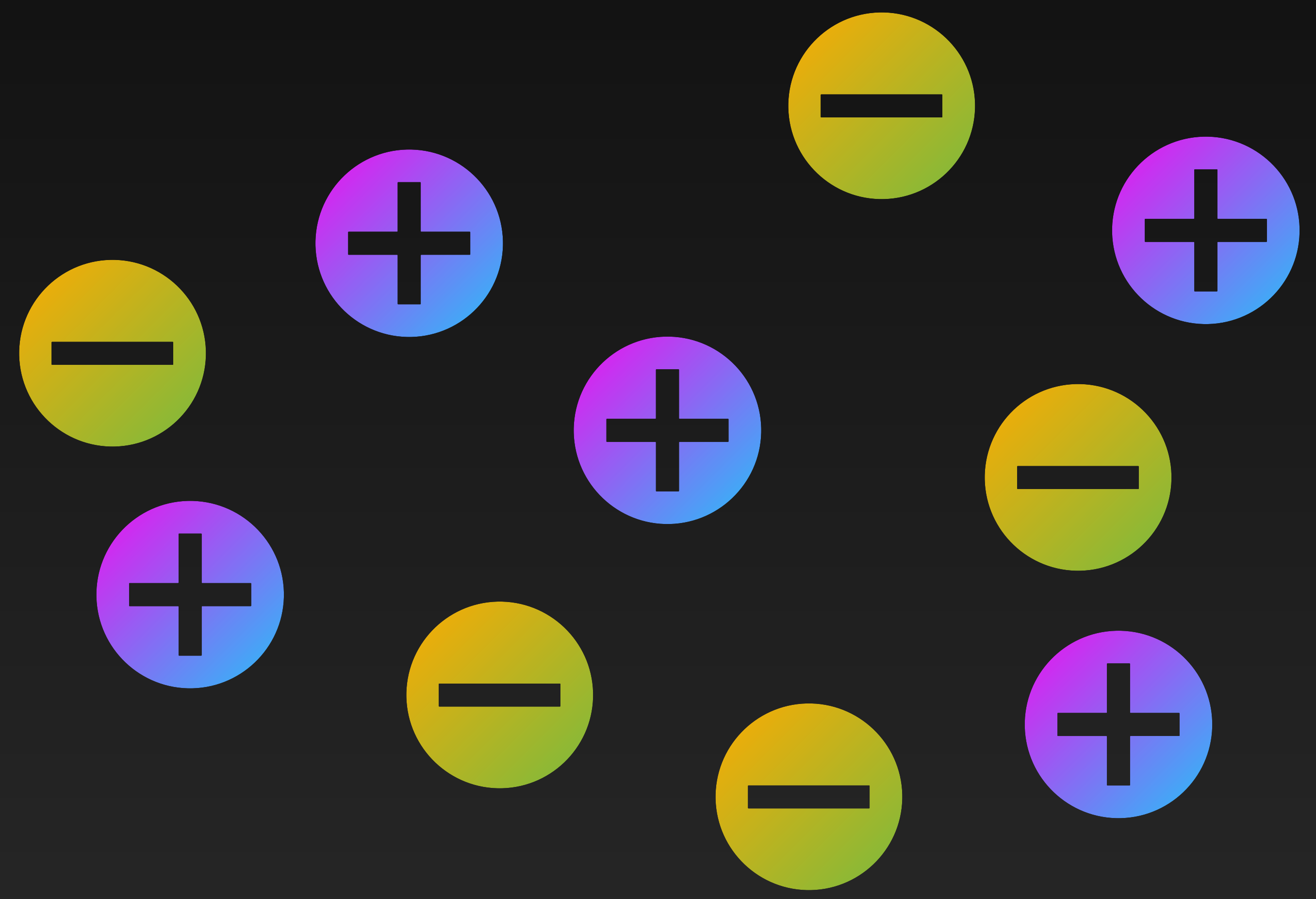
Plasma Simulation Via Particle-In-Cell Method

Nigel Shen
Advisor: Professor Antoine Cerfon
AM-SURE 2022

Introduction to Plasma

- Composed of free, charged particles (ions, electrons), similar to fluids
- Brutal way to simulate: $\mathbf{F} = m\mathbf{a}$ for each particle



Problem:

1. Too many particles ($N \sim 10^{20} - 10^{22}$)
2. Electromagnetic force is long-range $\Rightarrow$ Forces between any pair of particles

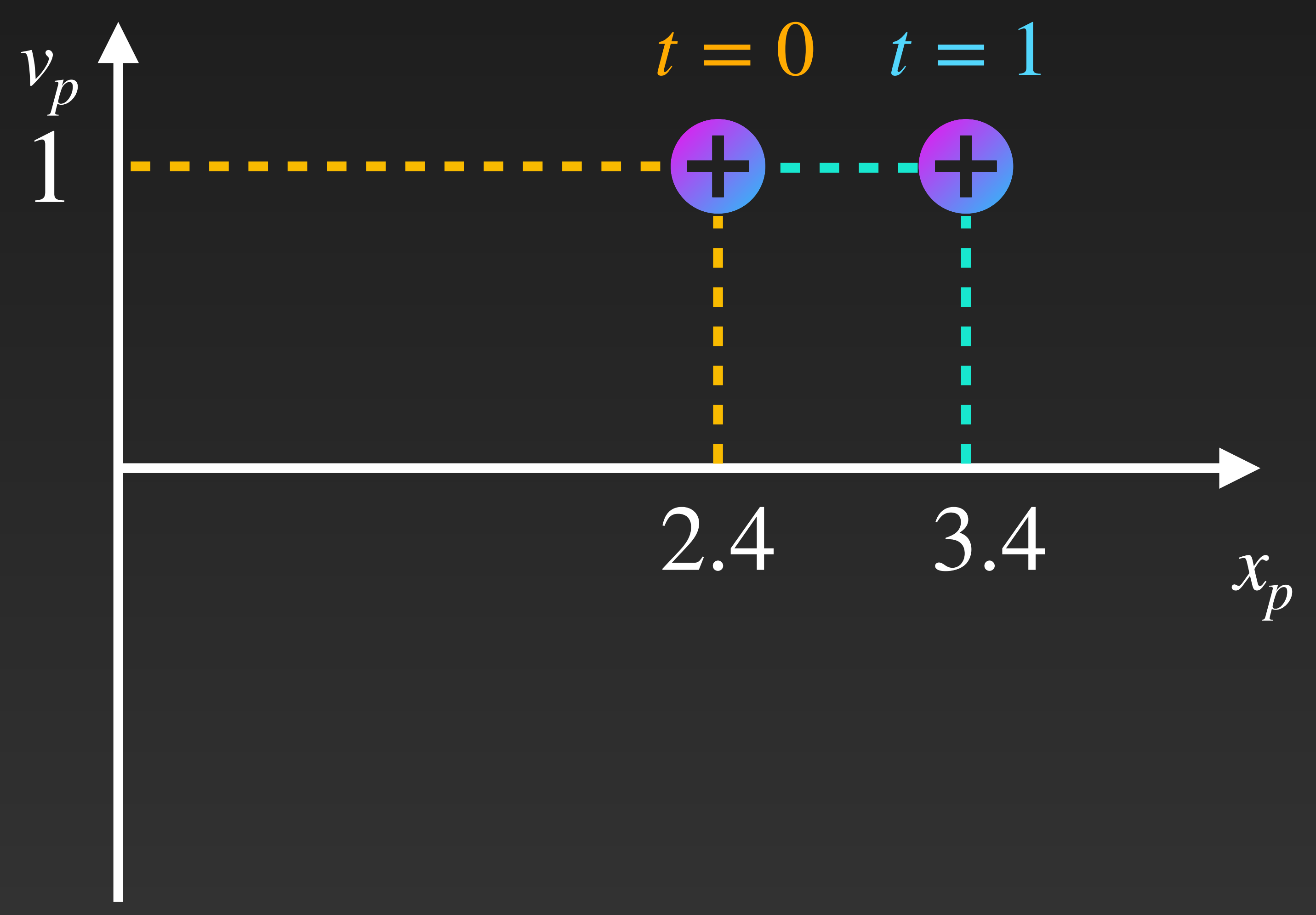
Instead: Want to find a system of PDEs to describe plasma

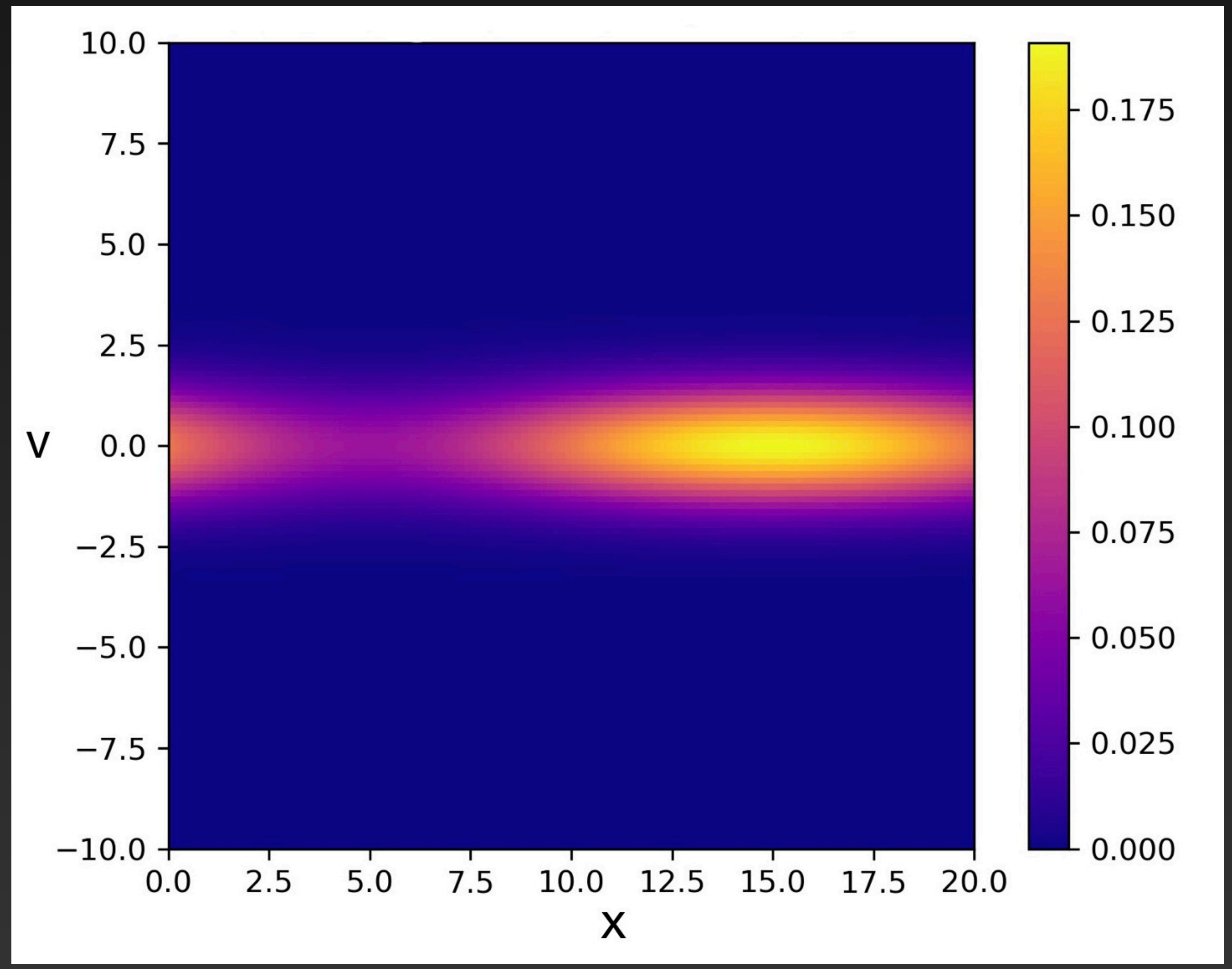
Phase Space & Distribution Function

- The phase space is the space of both particle position and particle velocity

- The distribution function $f(\mathbf{x}, \mathbf{v}, t)$ is defined in the phase space s.t. for all subdomain Ω:

$$\iint_{\Omega} f(\mathbf{x}, \mathbf{v}, t) d\mathbf{v} d\mathbf{x} = \text{\# of particles in } \Omega$$





$$f(x, v) = \frac{1}{\sqrt{2\pi}} e^{-v^2/2} \left(1 - 0.5 \sin\left(\frac{\pi x}{10}\right)\right)$$

The Vlasov-Poisson System

- The evolution of plasma can then be described by the Vlasov's Equation and the Maxwell's Equation:

$$\frac{df}{dt} = \frac{\partial f}{\partial t} + \frac{\partial f}{\partial \mathbf{x}}\frac{d\mathbf{x}}{dt} + \frac{\partial f}{\partial \mathbf{v}}\frac{d\mathbf{v}}{dt}$$

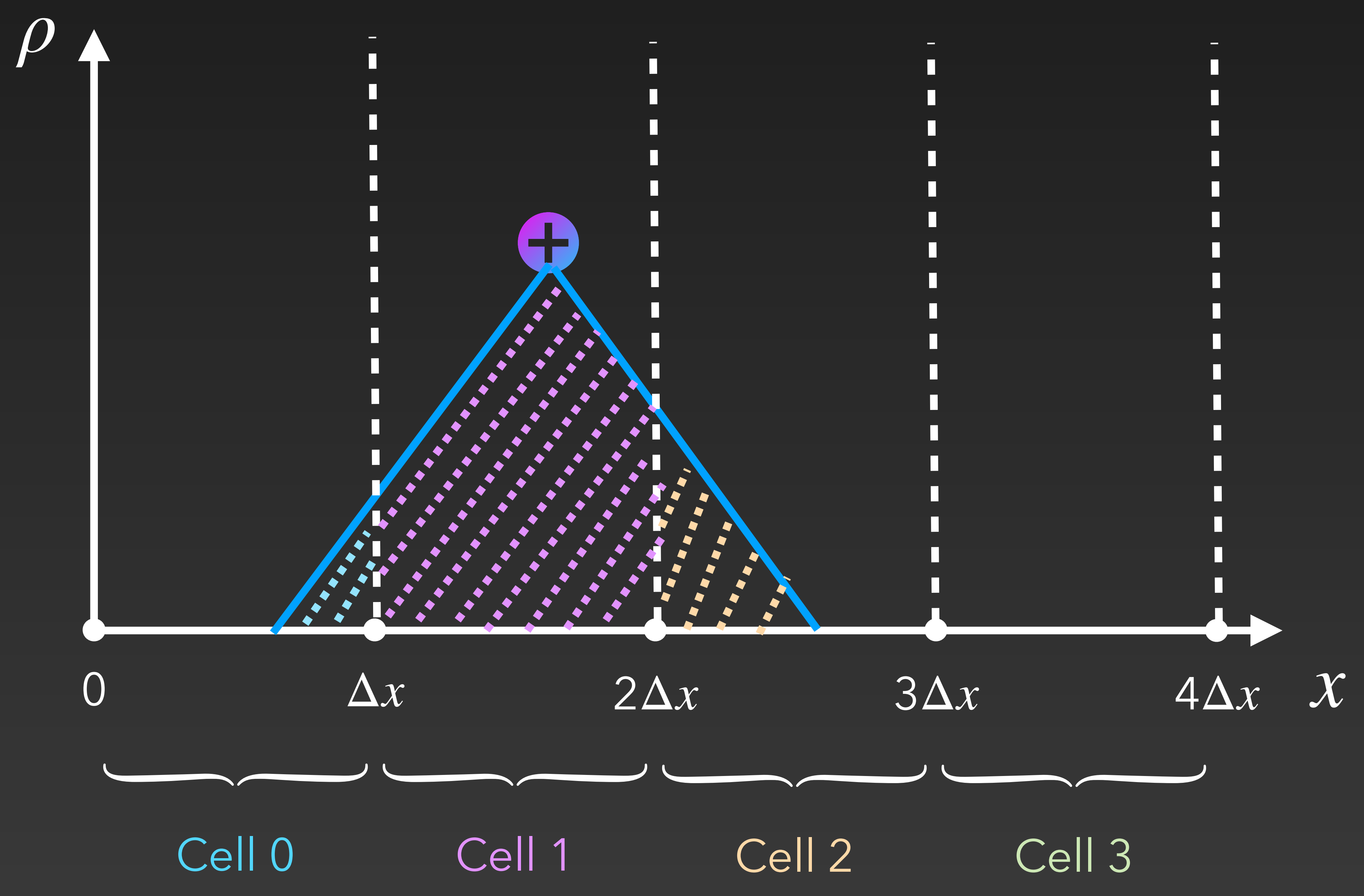
$$= \frac{\partial f}{\partial t} + \mathbf{v}\frac{\partial f}{\partial \mathbf{x}} + \frac{q\mathbf{E}}{m}\frac{\partial f}{\partial \mathbf{v}} = 0$$

$$\Delta\Phi = \rho = q\int_{\mathbb{R}^n} f(\mathbf{x}, \mathbf{v}, t)d\mathbf{v}; \qquad \mathbf{E} = \nabla\Phi$$

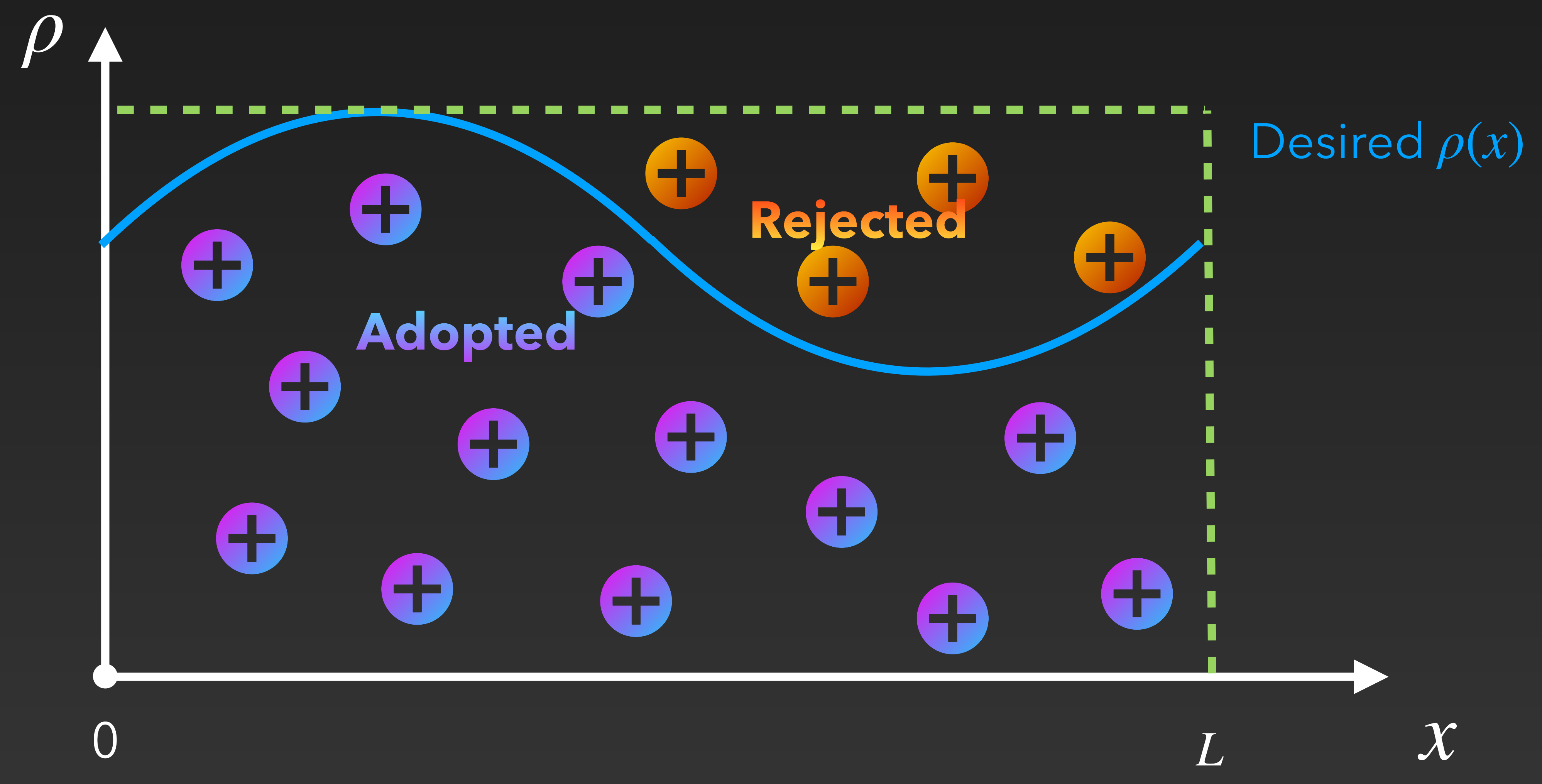
Particle-In-Cell

- Vlasov's Equation is up to 6+1Dimensions, still computationally expensive
- One choice is to use a finite element method called Particle-In-Cell (PIC)
- Basic Idea: Consider clusters of particles, and particle-field interactions
- Easy to sample the phase space via Monte Carlo method

Particle Interpolation Using 2nd Order b-spline



Particle Sampling Using Monte Carlo



PIC Algorithm

1. Initialize particle positions $\mathbf{X}_i^0$ & velocities $\mathbf{V}_i^0$ using Monte Carlo
2. Calculate ρ over the spatial grid points (Here $\mathbf{x}$ are discrete spatial points):
   $$\rho(\mathbf{x}, t) = \sum_i qS(\mathbf{X}_i^t - \mathbf{x})$$
3. Solve for the field $\Phi, \mathbf{E}$ using finite difference / spectral method
4. Calculate accelerations: $\mathbf{a}_i^t = \dfrac{q}{m} \sum_{\mathbf{x}} \mathbf{E}(\mathbf{x}) S(\mathbf{X}_i^t - \mathbf{x})$
5. Push the particle using Euler's Method / leapfrog
6. Repeat step 2-5.

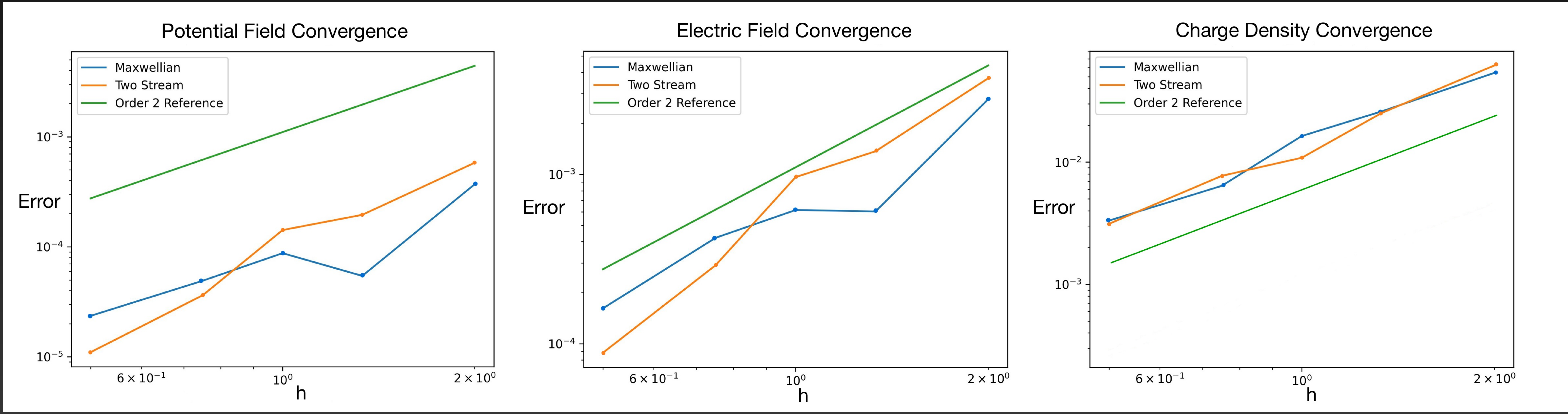
Performance of PIC: Manufactured Solution

$$f(\mathbf{x}, \mathbf{v}, t) = \frac{1}{\sqrt{2\pi}} e^{-\frac{\mathbf{v}^2}{2}} \left(1 - \frac{1}{2}\sin\left(\frac{\pi t}{4}\right)\sin\left(\frac{2\pi x}{L}\right)\right)$$

$$\frac{\partial f}{\partial t} + \mathbf{v}\frac{\partial f}{\partial \mathbf{x}} + \frac{q\mathbf{E}}{m}\frac{\partial f}{\partial \mathbf{v}} = S_f$$

Analytic solutions lead to an exact measurement of $\mathbf{E}, \Phi$, and ρ, at the cost of a source term S_f

Generalized Parameter: $h = \left(\frac{\Delta x}{\Delta x_0}\right)^2 = \left(\frac{\Delta t}{\Delta t_0}\right)^2 = \left(\frac{N}{N_0}\right)^{1/2}$



* "Maxwellian" and "Two Stream" are two different manufactured solutions. Both converges at an order of 2.

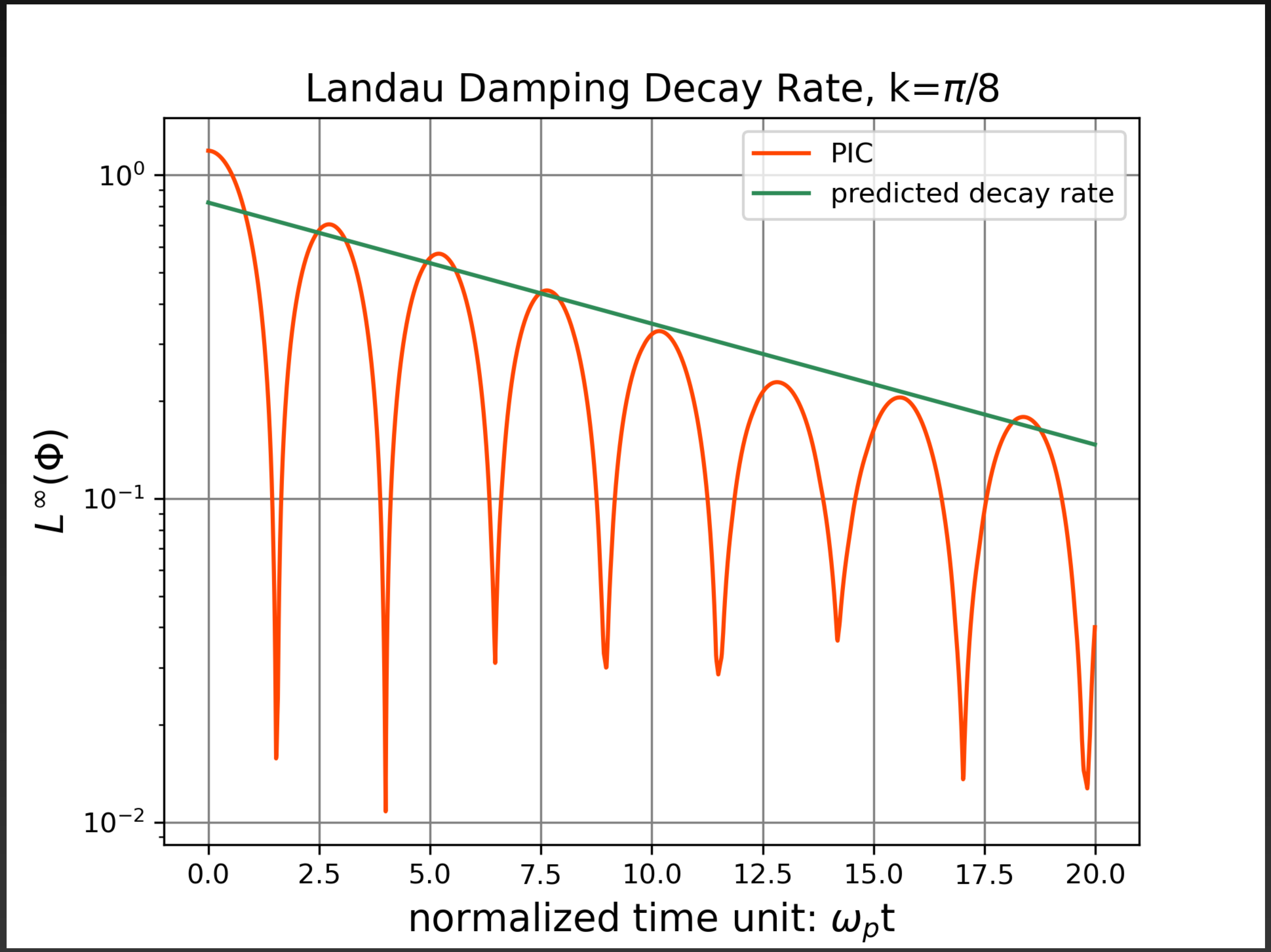
Performance of PIC: Landau Damping

$$f(\mathbf{x},\mathbf{v},0) = \frac{1}{\sqrt{2\pi}} e^{-\frac{v^2}{2}} \left(1 + A_0 \sin\left(\frac{2\pi x}{L}\right)\right)$$

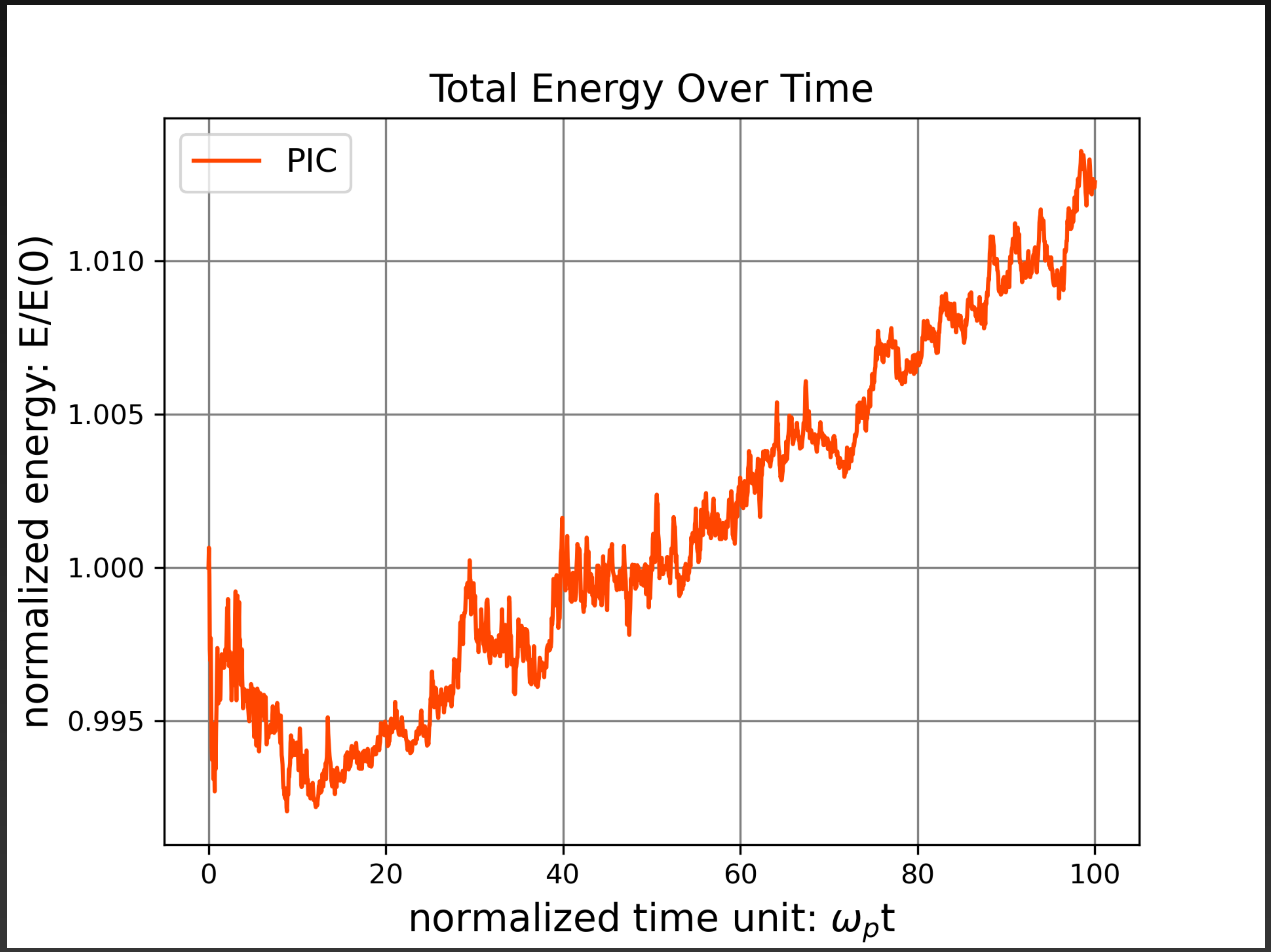
$$\Phi(\mathbf{x}, t) = \Phi_0 e^{i(kx-\omega t)}$$

Landau Damping: A conventional test on plasma simulation codes

The imaginary part of ω is the decay rate



100000 particles, 32 grid points



Grid Heating Observed

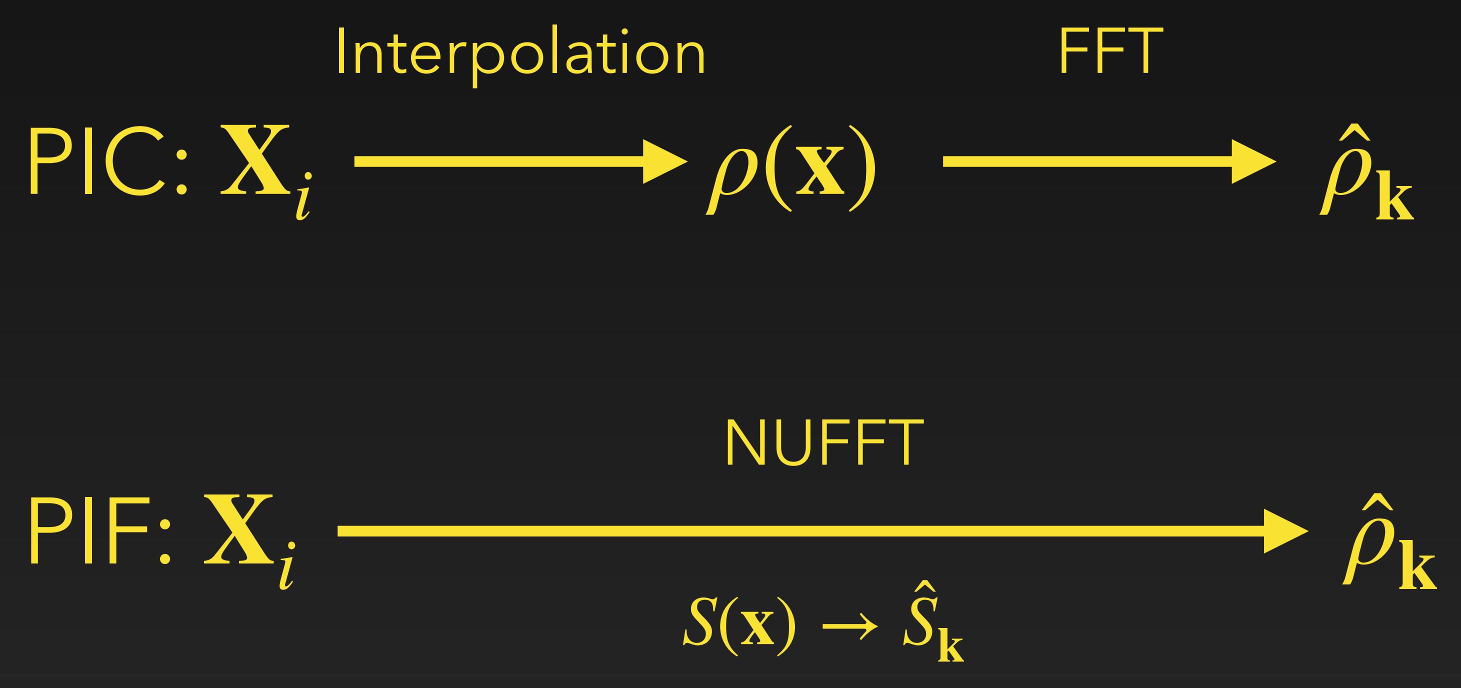
Particle in Fourier

- Calculate $\hat{\rho}$ directly from particle position via NUFFT:

$$\hat{\rho}^t_{\mathbf{k}} = \int_0^L e^{-i\mathbf{k}\cdot\mathbf{x}} \rho^t(\mathbf{x}) d\mathbf{x}$$

$$= \int_0^L e^{-i\mathbf{k}\cdot\mathbf{x}} \sum_{i=1}^{N_p} qS(\mathbf{x} - \mathbf{X}_i) d\mathbf{x}$$

$$= q\hat{S}_{\mathbf{k}} \sum_{i=1}^{N_p} e^{-i\mathbf{k}\cdot\mathbf{X}_i}$$

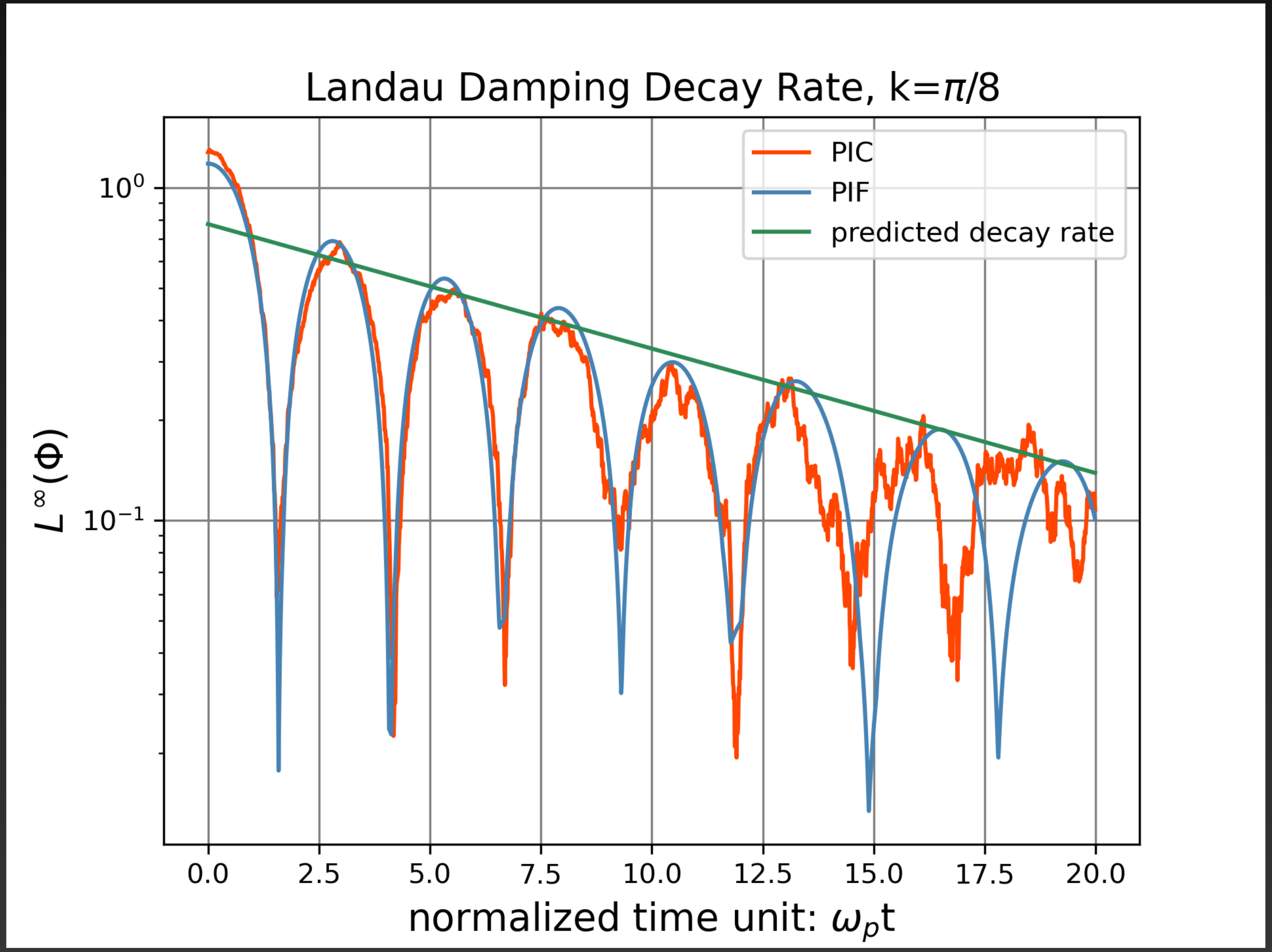


- Calculate accelerations via INUFFT: $\mathbf{a}^t_i = = \dfrac{q}{Lm} \sum_{\mathbf{k}} e^{i\mathbf{k}\cdot\mathbf{X}^t_i} \hat{\mathbf{E}}_{\mathbf{k}} \overline{\hat{S}_{\mathbf{k}}}$

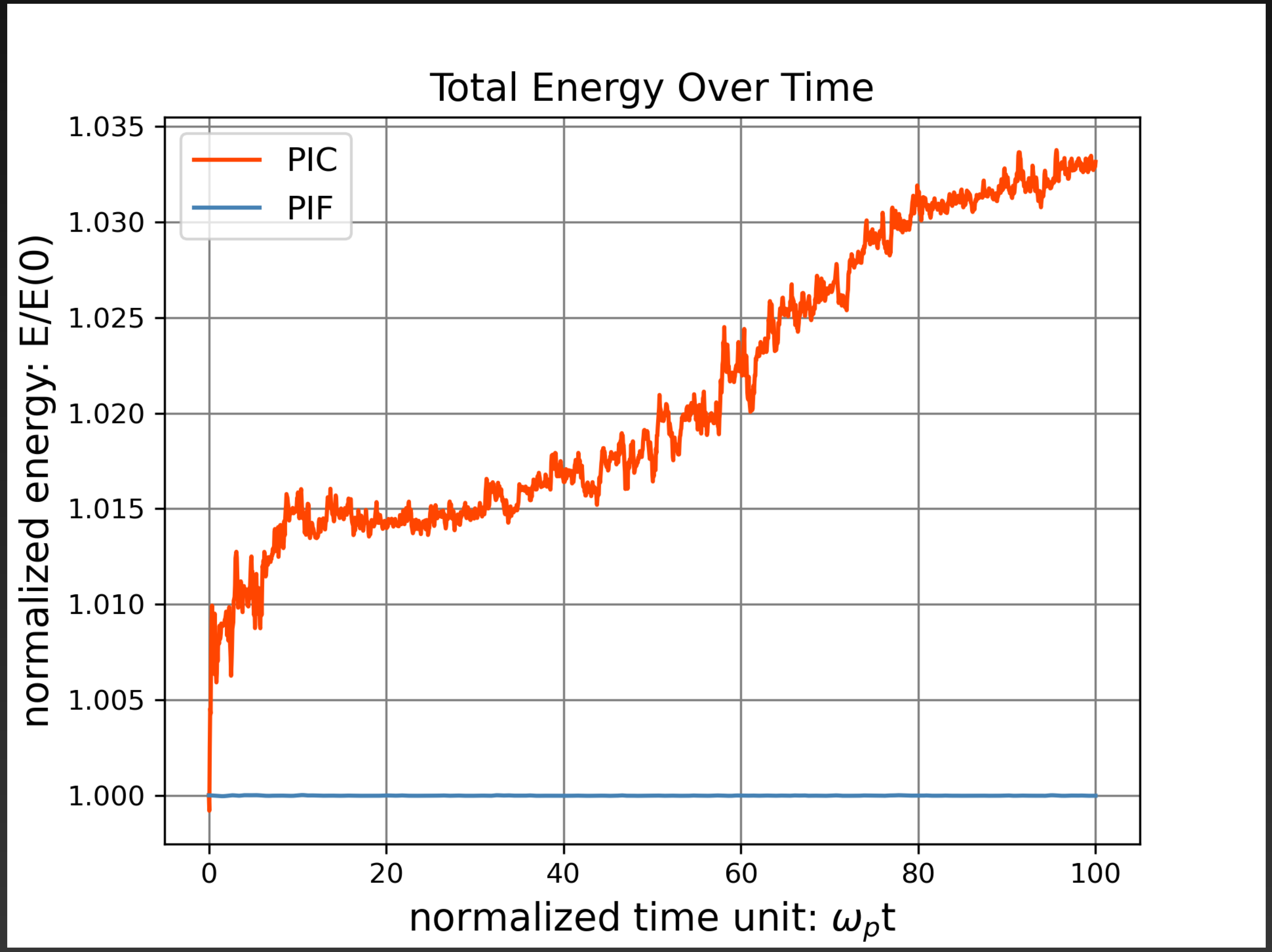
**NUFFT: Non-Uniform Fast Fourier Transform; We use fiNUFFT package from Flatiron Institute*

Performance of PIF

- An exact energy conservation in continuous time scheme
- Compared to PIC: No Grid Heating, Less Noise, Arbitrary Shape Functions, Needs More Computational Power



20000 particles, 16 modes/grid points



20000 particles, 16 modes/grid points

Future Work:

We want to apply the manufactured solution to the PIF algorithm

Probably a modification to the manufactured solution to test PIF properly

Select Works Cited:

1. Giovanni Lapenta. Particle in cell methods. In With Application to Simulations in Space. Weather. Citeseer, 2016.
2. Matthew S Mitchell, Matthew T Miecnikowski, Gregory Beylkin, and Scott E Parker. Efficient fourier basis particle simulation. *Journal of Computational Physics*, 396:837–847, 2019.
3. Paul Tranquilli, Lee Ricketson, and Luis Chaco´n. A deterministic verification strategy for electrostatic particle-in-cell algorithms in arbitrary spatial dimensions using the method of manufactured solutions. *Journal of Computational Physics*, 448:110751, 2022.

Thank You!

Questions?