

# Particle-in-Fourier: A Promising Paradigm for Extreme-Scale Plasma Simulations and Beyond

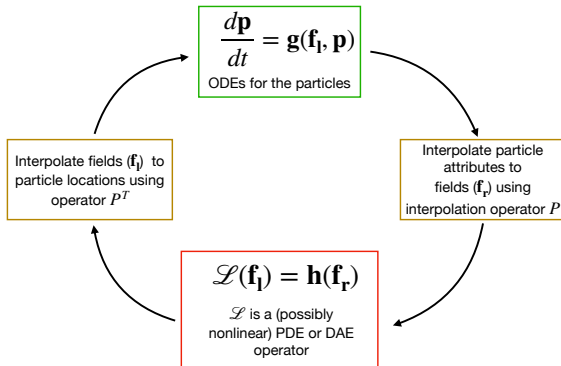
## Go20 2026 conference, Gozo, Malta

18 May 2026 | Sriramkrishnan Muralikrishnan | Jülich Supercomputing Centre, Germany

# Unified Particle-Field Framework for Dynamical Systems

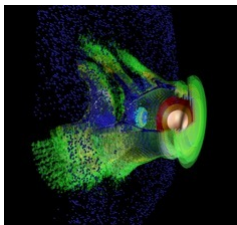
## Unified Particle-Field Representation

General structure: particles evolve via ODEs; fields evolve via PDEs; coupled through interpolation

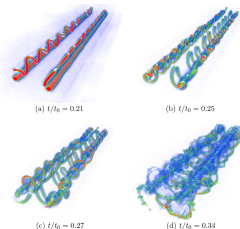


- ▶ Convection/Deformation operators: Treated in a Lagrangian way by tracking ODEs for particles
- ▶ Other operators: Treated in an Eulerian way by solving PDEs or DAEs for fields
- ▶ Two-way coupled through interpolation operators

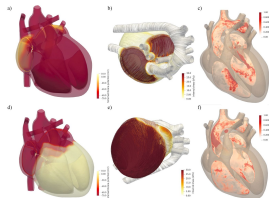
# Examples of Particle-Field Systems



Laser Wakefield acceleration simulation using PIC schemes (OSIRIS code)



Aircraft wake simulation using VIC schemes (Chatelain et. al., 2007)

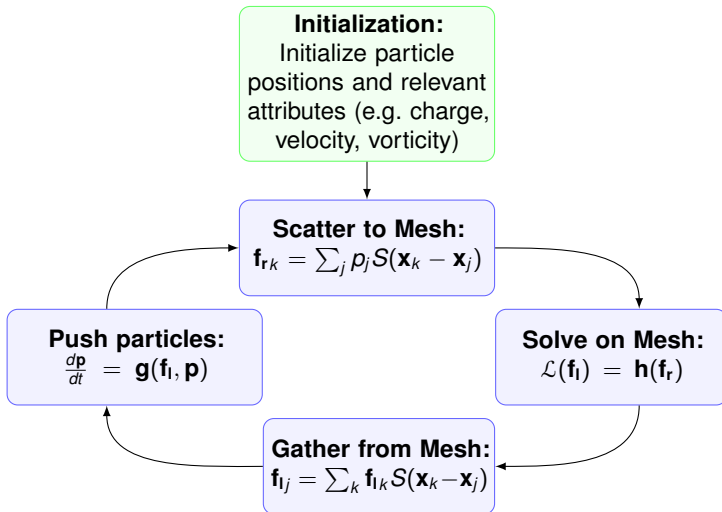


Heart simulations using Immersed Boundary Method (Viola et. al., 2023)

System	$\mathbf{p}$	$\mathbf{f}_l$	$\mathbf{f}_r$	ODE: $\frac{d\mathbf{p}}{dt} = \mathbf{g}(\mathbf{f}_l, \mathbf{p})$	PDE/DAE: $\mathcal{L}(\mathbf{f}_l) = \mathbf{h}(\mathbf{f}_r)$
Vlasov–Poisson	$(\mathbf{x}, \mathbf{u})$	$\mathbf{E}$	$\rho$	$d\mathbf{x}/dt = \mathbf{u}, d\mathbf{u}/dt = q/m(\mathbf{E} + \mathbf{u} \times \mathbf{B}_{ext})$	$-\Delta\phi = \rho(x), \mathbf{E} = -\nabla\phi$
Vlasov–Maxwell	$(\mathbf{x}, \mathbf{u})$	$(\mathbf{E}, \mathbf{B})$	$(\mathbf{J}, \rho)$	$d\mathbf{x}/dt = \mathbf{u}, d\mathbf{u}/dt = q/m(\mathbf{E} + \mathbf{u} \times \mathbf{B})$	Maxwell's eqn. with source $\rho, \mathbf{J}$
Fluid-structure	$\mathbf{x}$	$\mathbf{u}$	$\mathbf{f}$ (force from structure)	$d\mathbf{x}/dt = \mathbf{u}$	Navier–Stokes with forcing $\mathbf{f}$
Vorticity-velocity	$(\mathbf{x}, \omega)$	$\mathbf{u}$	$\omega$	$d\mathbf{x}/dt = \mathbf{u}, \omega$ constant or evolved	$\Delta\mathbf{u} = -\nabla \times \omega$
Navier-Stokes	$\mathbf{x}$	$\mathbf{u}$	$\mathbf{u}^*$ (from semi-Lagrangian)	$d\mathbf{x}/dt = \mathbf{u}$ (backward/forward tracking)	$\rho \frac{d\mathbf{u}}{dt} + \nabla p = \mu \Delta\mathbf{u}, \nabla \cdot \mathbf{u} = 0$

Instantiations of the unified particle–field framework across common plasma and fluid systems. The fluid-structure system uses immersed boundary formulation, whereas the Navier-Stokes in primitive variables uses semi-Lagrangian formulation.

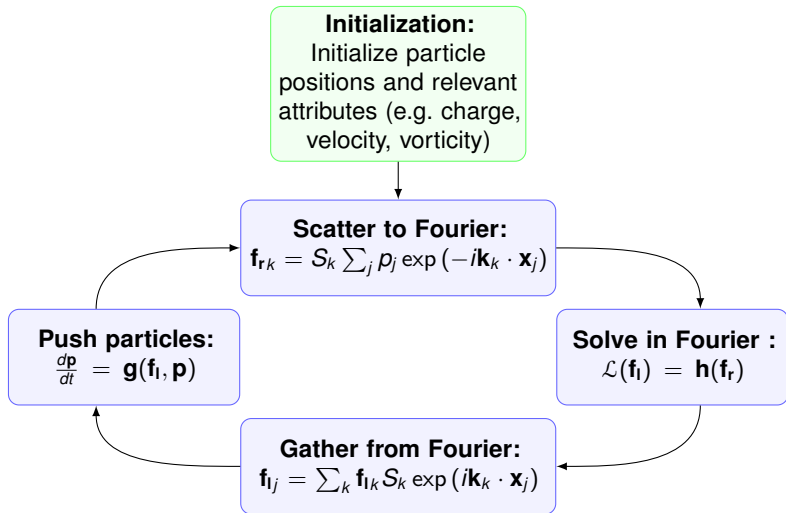
# Particle-Mesh Methods for Particle-Field Systems



# Particle-Mesh Methods for Particle-Field Systems

- ▶ **Particle-in-Cell (PIC)** for kinetic plasma simulations, **Vortex-in-Cell (VIC)** for vortex dynamics, **Immersed Boundary Methods (IBM)** for fluid-structure interaction
- ▶ **Semi-Lagrangian schemes:** Special case of this framework where the (virtual) particle and field grids coincide leading to only one interpolation
- ▶ **Advantages:**
  1. Treatment of convection operators in a Lagrangian way get rid of mesh-based CFL constraints
  2. Can take advantage of fast and scalable mesh-based PDE solvers
  3. Interpolation to a grid with compact stencil is local (good for parallel computing)
- ▶ **Disadvantages:**
  1. Interpolation to a grid can cause aliasing which leads to loss of conservation in invariants (e.g. energy)
  2. Order limited by the shape function used for interpolation

# Particle-Fourier Methods for Particle-Field Systems



## ► Advantages:

- No aliasing, excellent conservation of invariants
- Spectral accuracy

## ► Disadvantages:

- High computational cost  $\mathcal{O}(N_p N_m)$  and global nature
- Periodic boundary conditions

# History of Particle-Fourier Methods

- ▶ **As ideal schemes to analyze particle-mesh methods:** Langdon and Birdsall 1970; Huang et al. 2016; Peskin 2002; Cottet and P. D. Koumoutsakos 2000
- ▶ Mitchell et al. 2019: Use of Nonuniform FFTs (NUFFT) for scatter and gather to reduce the cost of PIF schemes to  $\mathcal{O}(N_p + N_m \log N_m)$  for electrostatic kinetic plasma simulations
- ▶ Chen and Peskin 2023: Use of similar idea in the context of immersed boundary methods

- ▶ NUFFTs bring the complexity of PIF schemes to the same order as PIC schemes (although with a higher constant)
- ▶ What about **boundary conditions, scalability and time to solution?** (focus of this talk)
- ▶ From now onwards we will consider the **electrostatic kinetic plasma (Vlasov-Poisson) system**

$$\frac{\partial f}{\partial t} + \mathbf{v} \cdot \nabla_x f + \frac{q}{m} (\mathbf{E} + \mathbf{v} \times \mathbf{b}) \cdot \nabla_v f = 0,$$

$$-\nabla^2 \phi = \frac{\rho}{\epsilon_0}, \quad \mathbf{E} = -\nabla \phi.$$

- ▶ Nonlinear, high-dimensional, coupled system with multiple spatial and temporal scales

# PIF for open boundary conditions

## Free space Poisson problem

- ▶ Poisson's equation

$$\Delta\varphi = -\rho,$$

in  $\mathbb{R}^d$  with open (free space) boundary conditions.

- ▶ Integral solution to Poisson's equation

$$\varphi(\mathbf{x}) = \int_{\Omega} g(\mathbf{x} - \mathbf{y})\rho(\mathbf{y})d\mathbf{y},$$

where  $g(\mathbf{r}) = \frac{1}{4\pi r}$  for  $d = 3$  and  $g(\mathbf{r}) = \frac{-1}{2\pi} \log(r)$  for  $d = 2$ .

- ▶ Green's function has a slow decay and singular at origin
- ▶ FFT-based free space Poisson solvers:
  - ▶ Convolution in real space becomes point-wise product of Fourier transforms in Fourier space
  - ▶ Zero pad  $\rho$  to perform aperiodic convolution
  - ▶ Regularize Green's function at origin

# Spectral grid-based free space Poisson solver

Vico-Greengard-Ferrando scheme (Vico et. al., 2016)

- ▶ **Key idea:** We are interested in  $\varphi$  only inside  $\Omega$ . Substitute  $g$  with truncated Green's function  $g^L$

$$g^L(\mathbf{r}) = \begin{cases} \frac{1}{4\pi r} \text{rect}\left(\frac{r}{2L}\right) & \text{if } d = 3 \\ \frac{-1}{2\pi} \log(r) \text{rect}\left(\frac{r}{2L}\right) & \text{if } d = 2 \end{cases},$$

where  $\text{rect}(x)$  is the characteristic function on the unit interval:

$$\text{rect}(x) = \begin{cases} 1 & \text{for } |x| < 1/2 \\ 0 & \text{otherwise} \end{cases}.$$

If we set  $L > \sqrt{d}$ , then the solution  $\varphi$  in  $\Omega$  is exactly

$$\varphi(\mathbf{x}) = \int_{\Omega} g^L(\mathbf{x} - \mathbf{y}) \rho(\mathbf{y}) d\mathbf{y}$$

# Spectral grid-based free space Poisson solver

## Vico-Greengard-Ferrando scheme

- **Advantages of  $g^L$ :** Removes singularity at origin and its Fourier transforms are known analytically

$$\mathcal{F}(g^L) = \hat{g}^L = \begin{cases} 2 \left( \frac{\sin(Ls/2)}{s} \right)^2 & \text{if } d = 3 \\ \frac{1 - J_0(Ls)}{s^2} - \frac{L \log(L) J_1(Ls)}{s} & \text{if } d = 2 \end{cases}.$$

- These are smooth functions but have some oscillatory nature
- To resolve them properly without aliasing need an extended grid of size  $(4N_g)^d$  whereas the minimum zero padding needed is  $(2N_g)^d$ .  $N_g$  is the number of grid points in each dimension in the original domain we want to solve

# PIF + spectral free space Poisson solver

## Algorithm pseudo-code

**Require:** Construct convolutional kernels  $g^L * S$  and  $\nabla(g^L * S * S)$  in Fourier space

**Require:** Initialize particle positions, velocities and charges;

1: **for**  $n = 0$  to  $N_t - 1$  **do**

2:   Type 1 NUFFT to transform particle positions into an upscaled  $4N_g$  Fourier grid

$$\hat{\mathbf{x}}(\mathbf{k}) = \sum_{j=1}^{N_p} \exp(-i\mathbf{k} \cdot \mathbf{x}_j);$$

3:   Find the Fourier modes of the free space potential  $\varphi$  and acceleration  $\mathbf{a}$ :

$$\begin{aligned}\hat{\varphi}(\mathbf{k}) &= q\hat{g}^L(\mathbf{k})\hat{S}(\mathbf{k})\hat{\mathbf{x}}(\mathbf{k}), \\ \hat{\mathbf{a}}(\mathbf{k}) &= -\frac{i\mathbf{k}}{m}q^2\hat{g}^L(\mathbf{k})\hat{S}^2(\mathbf{k})\hat{\mathbf{x}}(\mathbf{k});\end{aligned}$$

4:   Find the free space acceleration of each particle using Type 2 NUFFT:

$$\mathbf{a}_j = \sum_{\mathbf{k}} \hat{\mathbf{a}}(\mathbf{k}) \exp(i\mathbf{k} \cdot \mathbf{x}_j) + \frac{q}{m} \mathbf{v}_j \times \mathbf{B}_0.$$

5:   Push the particles exactly as in standard PIC / PIF

6: **end for**

# Energy, Momentum and Charge conservation

## Theorem

*The free space PIF scheme with a second-order leap frog time integrator satisfies the following energy, momentum and charge conservation properties:*

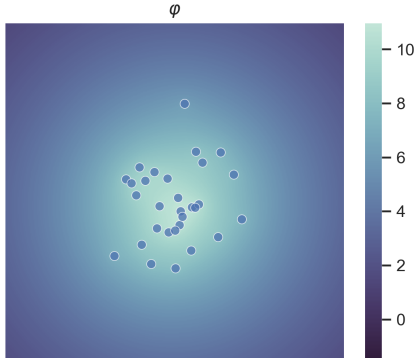
$$\begin{aligned}U_K^{n+1} + U_E^{n+1} &= U_K^n + U_E^n + \mathcal{O}(\Delta t^2), \\ \mathbf{p}^{n+1/2} - \mathbf{p}^{n-1/2} &= \mathbf{0}, \\ \|Q^n - Q\| &\leq \varepsilon,\end{aligned}$$

*where  $U_K$  and  $U_E$  are the kinetic and potential energies,  $\mathbf{p}$  denotes the momentum,  $Q$  the (exact) total charge and  $n$  the time step index.  $\varepsilon$  is the tolerance selected for the NUFFT.*

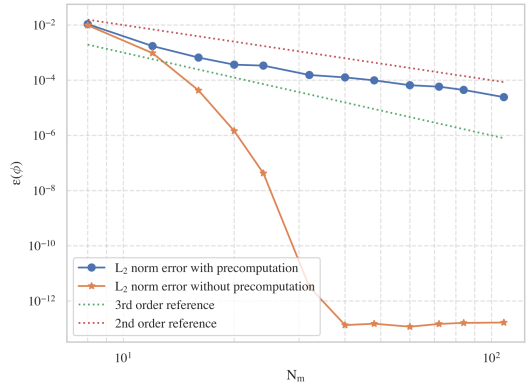
- ▶ Momentum: exactly satisfied
  - ▶ Energy: Satisfied upto the time step size and order of the time integrator
  - ▶ Charge: Satisfied upto the chosen tolerance of the NUFFT
- In contrast energy conservation in the standard PIC scheme also depends on the total number of particles and grid points because of aliasing

# Numerical results: Convergence

Free space Poisson problem with analytic solution

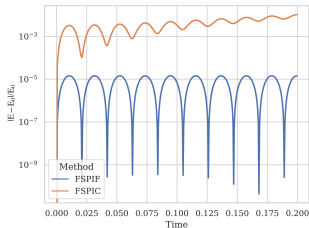
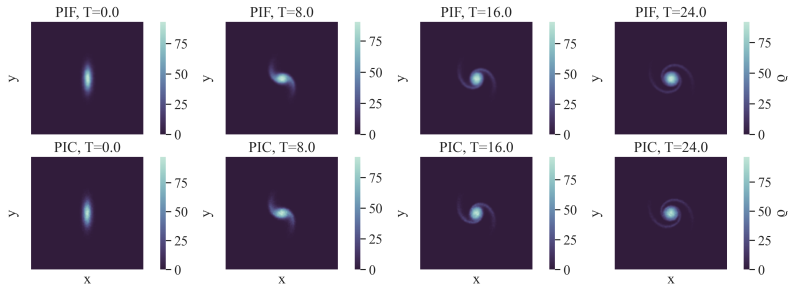


Analytical solution

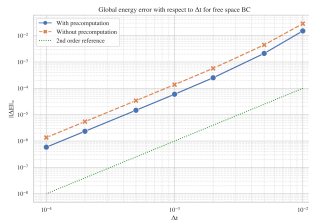


Convergence of gridless Poisson solver

# 2D cyclotron test case



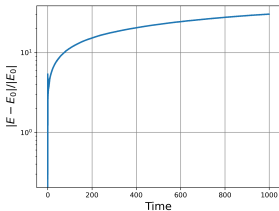
Energy error PIF Vs PIC



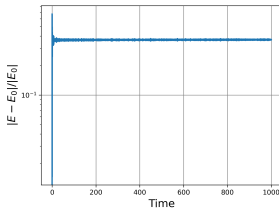
Convergence of energy conservation error

# Long time integration: 2D cyclotron

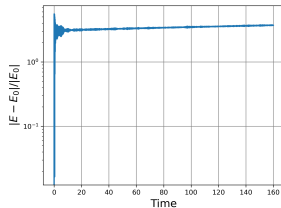
16000 time steps, PIF scheme 12× faster than PIC scheme for comparable accuracy on 1 CPU



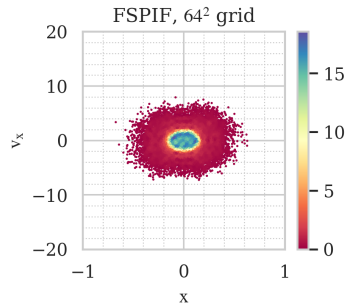
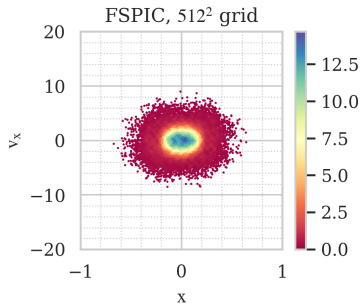
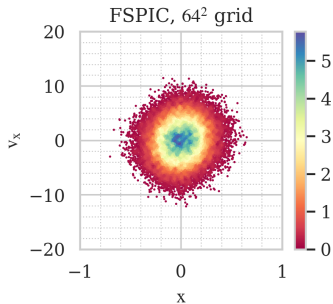
PIC,  $64^2$  grid



PIF,  $64^2$  modes

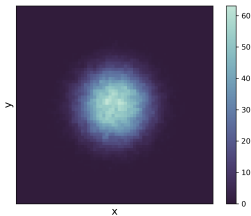


PIC,  $512^2$  grid

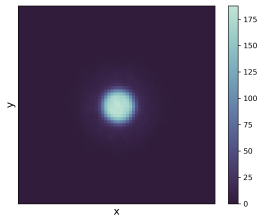


# Long time integration: 2D cyclotron

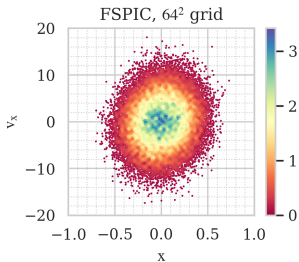
100000 time steps, Beam size and phase space much larger in PIC due to grid heating. Important for particle accelerator applications



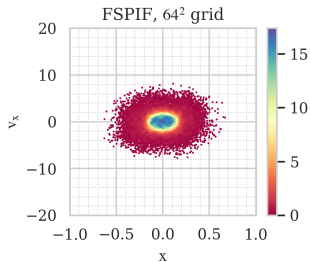
PIC,  $64^2$  grid



PIF,  $64^2$  modes

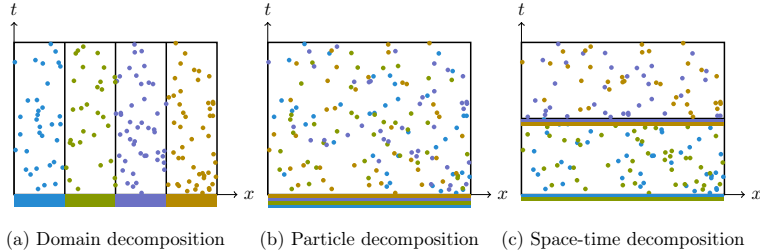


FSPIC,  $64^2$  grid



FSPIF,  $64^2$  grid

# Distributed parallelization strategies for PIF schemes



**Fig. 1** Schematic of the three parallelization strategies for particle-in-Fourier schemes. Colors denote space-time MPI ranks, which is  $4 \times 1$  for domain and particle decompositions and  $2 \times 2$  for space-time decomposition. Horizontal cuts indicate time decomposition; vertical cuts indicate spatial decomposition. The colored bars denote Fourier modes which are distributed for domain decomposition and duplicated for particle and space-time decompositions.

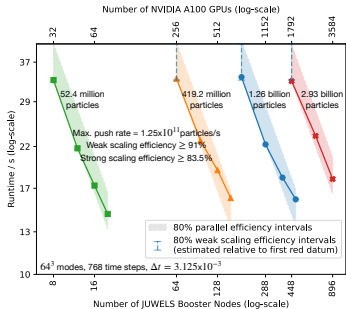
# Communication patterns and suitable parameter regimes

“Collective” refers to global reductions or Alltoall operations, while “P2P” denotes point-to-point communication between neighboring ranks.

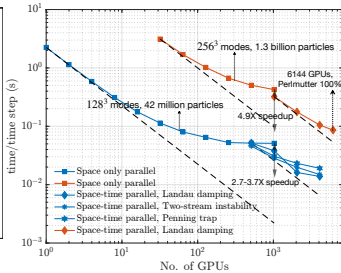
Strategy	Particles		Fields (Fourier modes)	
	Space	Time	Space	Time
Domain decomposition	P2P	none	P2P, Collective (Alltoall)	none
Particle decomposition	none	none	Collective (Allreduce)	none
Space–time decomposition	none	P2P	Collective (Allreduce)	none

- ▶ **Domain decomposition:** Only option when both high number of Fourier modes and particles are required. Needs load balancing strategies for clustered distributions
- ▶ **Particle decomposition:** Load balanced by design. Good strategy when  $N_p \gg N_m$ . Limited by the memory of a single rank to hold all the Fourier modes
- ▶ **Space-time decomposition:** Works best when a lot of time steps are needed. Speedup depends on the effectiveness of the coarse propagator for the problem under consideration

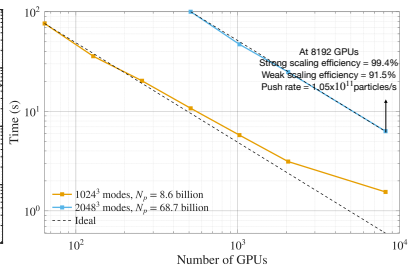
# Strong scaling study: In the best parameter regimes



Particle decomposition, JUWELS Booster



Space-time decomposition, Perlmutter

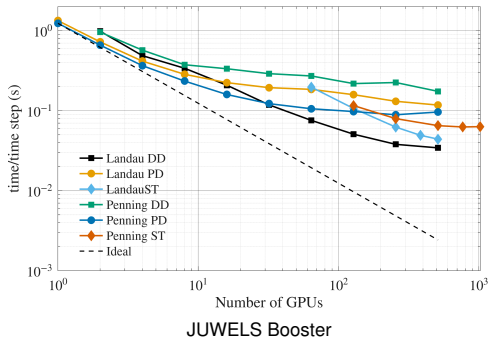
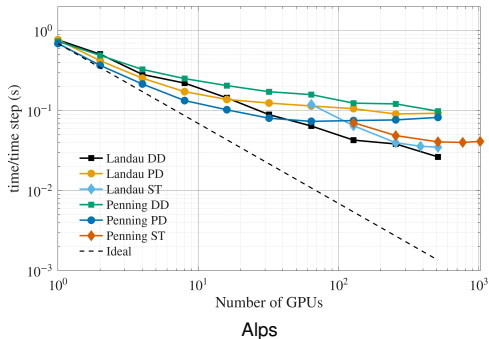


Domain decomposition, LUMI

- ▶ All the parallel implementations are done in our open-source, performance portable (through Kokkos), C++ library Independent Parallel Particle Layer (IPPL) <https://github.com/IPPL-framework/ippil>

# Strong scaling study: comparison

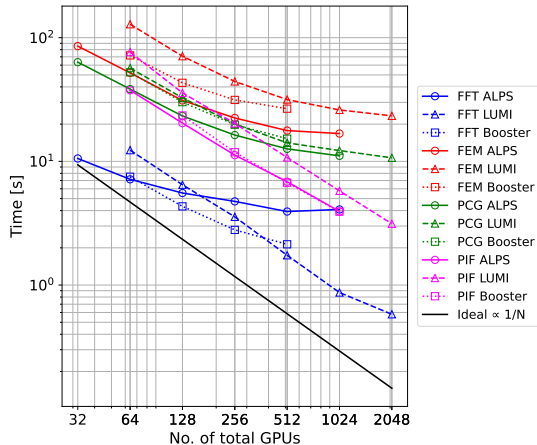
256<sup>3</sup> Fourier modes, 168 million particles



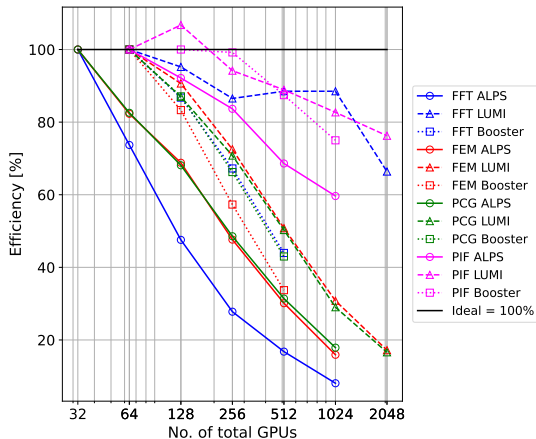
- ▶ **Landau damping:** Homogeneous distribution of particles. Domain decomposition best strategy. Space-time decomposition competitive at 512 GPUs.
- ▶ **Penning trap:** Clustered distribution of particles. Domain decomposition suffers from load balancing. Space-time decomposition offers 2.5× and 2× speedups compared to domain decomposition and particle decomposition at 512 GPUs.

# Comparison with PIC schemes: Strong scaling

Landau damping:  $1024^3$  grid/Fourier modes, 8 billion particles



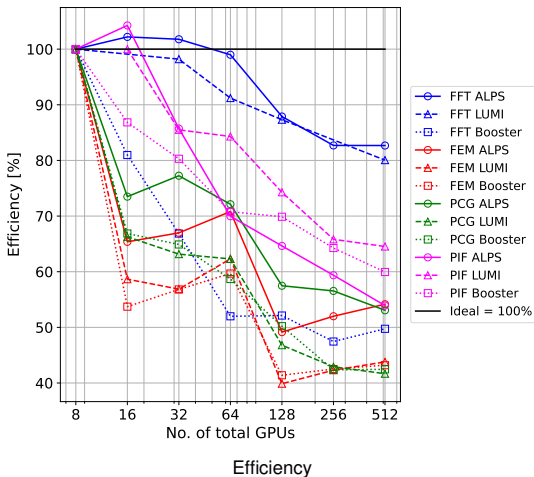
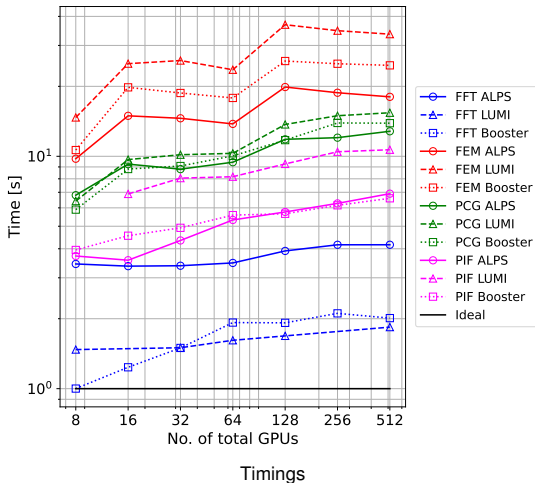
Timings



Efficiencies

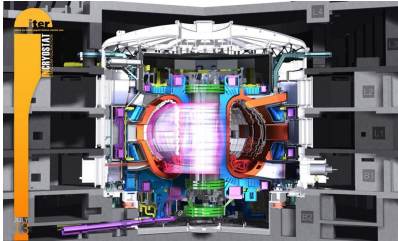
# Comparison with PIC schemes: Weak scaling

Landau damping:  $256^3$  grid/Fourier modes, 134 million particles to  $1024^3$  grid/Fourier modes, 8 billion particles (8 particles per grid/mode)

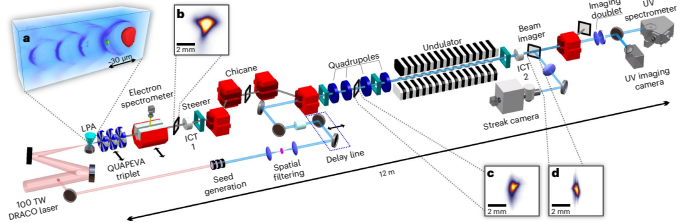


# Surrogate models for particle-field methods

- ▶ Kinetic plasma simulations: Nuclear fusion, Particle accelerators



Source: iter.org

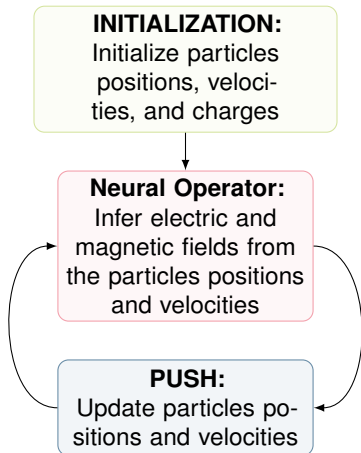


Source: M. Labat et al., Nature Photonics, 2023

## ▶ Issues:

- ▶ Computational cost: large number of grid points/Fourier modes ( $\mathcal{O}(10^{11})$ ), particles ( $\mathcal{O}(10^{12})$ ) and time steps ( $\mathcal{O}(10^5)$ ) for high-fidelity simulations
- ▶ Many-query scenarios: Still unreachable with exascale supercomputers
- ▶ Need cheap surrogate models which still capture essential physics

# NEOPIC: A Neural Operator Framework for particle-based methods



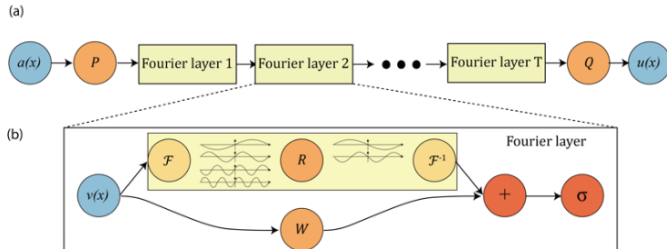
## Steps:

- ▶ **Particle pusher:** Same as in particle-based methods
- ▶ **Electric and magnetic fields:** Obtain from a *neural operator* instead of mesh-based or tree-based (mesh-free) field solvers by approximating the map  $P^T \mathcal{L}^{-1} P : (\mathbf{x}, \mathbf{v}) \rightarrow (\mathbf{E}, \mathbf{B})$

## Advantages:

- ▶ **Not constrained by Debye length and Courant-Friedrichs-Lewy time stepping restrictions**
- ▶ Can use data from a variety of particle-based methods (**discretization invariance**)
- ▶ Can be trained and tested with different numbers of particles (**resolution invariance**)
- ▶ **Can be interfaced with any particle-based production/community code**
- ▶ Can use **implicit time stepping with Jacobians from autograd**
- ▶ Utilizes the strengths of both particle methods and AI

# A Nonuniform Fourier Neural Operator



**(a) The full architecture of neural operator:** start from input  $a$ . 1. Lift to a higher dimension channel space by a neural network  $P$ . 2. Apply four layers of integral operators and activation functions. 3. Project back to the target dimension by a neural network  $Q$ . Output  $u$ . **(b) Fourier layers:** Start from input  $v$ . On top: apply the Fourier transform  $\mathcal{F}$ ; a linear transform  $R$  on the lower Fourier modes and filters out the higher modes; then apply the inverse Fourier transform  $\mathcal{F}^{-1}$ . On the bottom: apply a local linear transform  $W$ .

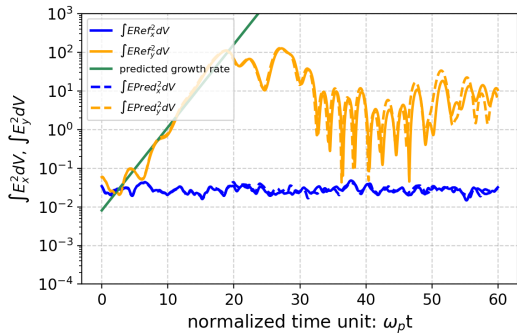
Figure 2: **top:** The architecture of the neural operators; **bottom:** Fourier layer.

- ▶ Nonuniform Discrete Fourier Transform (NUDFT) or NUFFT is used for the forward and inverse Fourier transforms
- ▶ Each Fourier block (without the skip connection) resembles a **learnable data-driven PIF scheme**
- ▶ But this operates on the lifted latent space

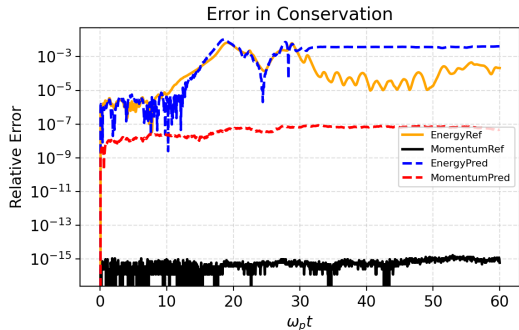
# Training Data Generation

- ▶ Using PIF or PIC scheme
- ▶ Miniapps:
  1. Weak and strong Landau: collisionless damping of plasma waves
  2. Two-stream instability: two interpenetrating electron beams
  3. Bump-on-tail instability: small amount of high-velocity particles
  4. Cyclotron: external magnetic field
- ▶ 600 time steps with 500,000 particles for each mini-app
- ▶ Input: [batchsize or timesteps, dimension, particle positions]
- ▶ Output: [batchsize or timesteps, dimension, electricfield at particle positions]

# Inference: Two-stream instability



Field energy

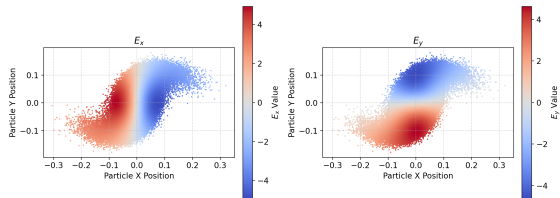


Conservation

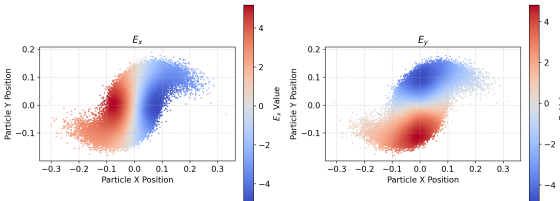
Inference still does not deteriorate much after the training horizon of  $T = 30$

# Inference: Cyclotron

Spatial E-Field Comparison for Timestep 1000

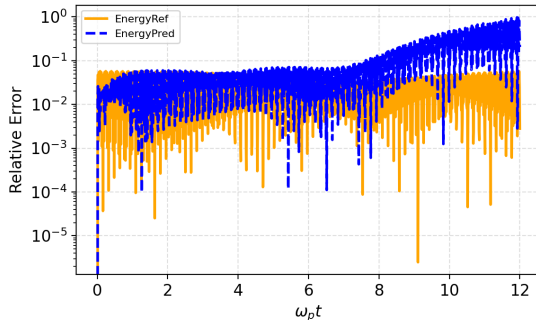


PIF, T=10



NEOPIC, T=10

Error in Conservation



- ▶ Energy conservation error steadily increases after the training time horizon ( $T = 6$ )

Results are still preliminary and work is ongoing

# Summary, Ongoing and Future works

## Summary

- ▶ Particle-in-Fourier are a class of spectral particle methods and offers benefits from the Lagrangian nature of particles and spectral accuracy of Fourier space
- ▶ Presented our works to address their concerns with respect to **boundary conditions, scalability and time to solution**
- ▶ PIF is available through our open-source, performance portable, C++ library IPPL  
<https://github.com/IPPL-framework/ippl>

## Ongoing work

- ▶ Particle-in-Fourier for fluid/vortex dynamics (Spectral Semi-Lagrangian and Vortex methods) Y. Mostafa Master's thesis
- ▶ Multigrid Reduction in Time (MGRIT) for time parallelization of PIF with J. Schroder
- ▶ Foundational model for fields in particle-based methods C. John PhD thesis

## Future work

- ▶ **Time integration for PIF:** So far only second order explicit leap frog/Boris is only used. Implicit time integration with PIF (preferably higher order to match its spectral nature) for stiff plasma systems.

# Acknowledgements

- ▶ **Collaborators:** C.N. Shen (MIT, USA), A. Cerfon (Type One Energy Group, Canada), A. Adelman (PSI, Switzerland), R. Speck (JSC, Germany), S. Mayani (ETHZ, Switzerland), P. Fischill (ETHZ, Switzerland), C. John (JSC, Germany)

The project 'NEOPIC: A Neural Operator Framework for Particle-based Kinetic Plasma Simulations' (INF funding code: ZT-I-PF-5-242) is funded by the Initiative and Networking Fund of the Helmholtz Association in the framework of the Helmholtz AI project call.

# References

- ▶ Shen, C.N., Cerfon, A. and Muralikrishnan, S., 2024. A particle-in-Fourier method with semi-discrete energy conservation for non-periodic boundary conditions. *Journal of computational physics*, 519, p.113390.
- ▶ Muralikrishnan, S., Fischill, P., Adelman, A. and Speck, R., 2026. On Distributed Parallelization Strategies for Particle-in-Fourier Schemes. *arXiv preprint arXiv:2605.10729*.
- ▶ Mayani, S., Fischill, P., Muralikrishnan, S. and Adelman, A., 2026. A Comparison of Massively Parallel Performance Portable Particle-in-Cell schemes for electrostatic kinetic plasma simulations. *arXiv preprint arXiv:2605.05469*.
- ▶ Muralikrishnan, S. and Speck, R., 2025. Error Analysis and Parallel Scaling Study of a Parareal Parallel-in-Time Integration Algorithm for Particle-in-Fourier Schemes. *SIAM Journal on Scientific Computing*, pp.S311-S336.
- ▶ Fischill, P., Adelman, A. and Muralikrishnan, S., 2026. A Performance-Portable, Massively Parallel Distributed Nonuniform FFT. *arXiv preprint arXiv:2605.10678*.