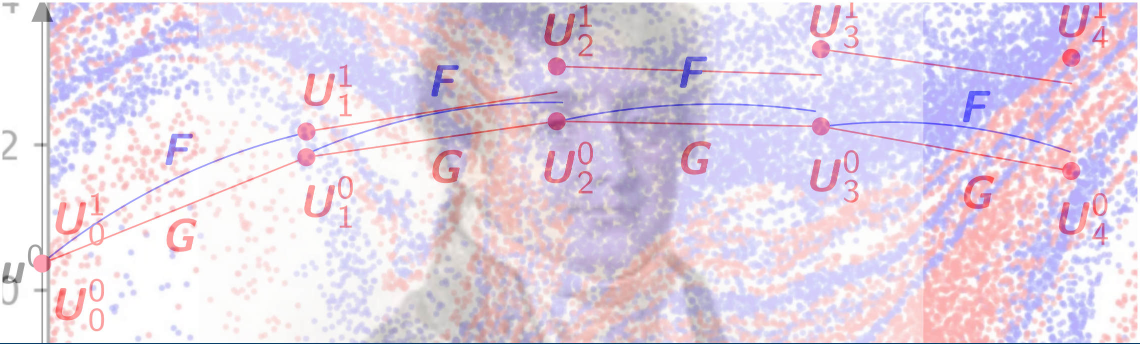


A Massive Space-Time Parallel Particle-In-Fourier Framework for Kinetic Plasma Simulations

Go20 conference, Gozo Malta

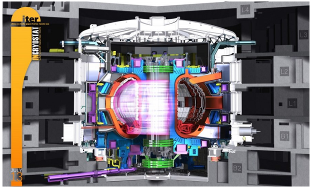
22 May 2025 | *Robert Speck* and Sriramkrishnan Muralikrishnan | Jülich Supercomputing Centre, Germany

Member of the Helmholtz Association

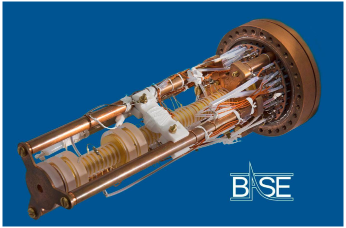
JÜLICH Forschungszentrum

Scientific Motivation

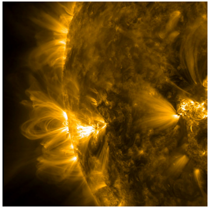
- Kinetic simulations of plasma play an important role in modelling nuclear fusion, particle accelerators, particle physics, astrophysical phenomena such as solar flares etc.



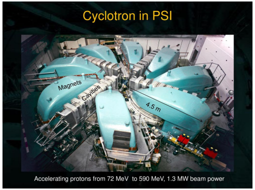
Source: iter.org



Source: CERN



Source: NASA

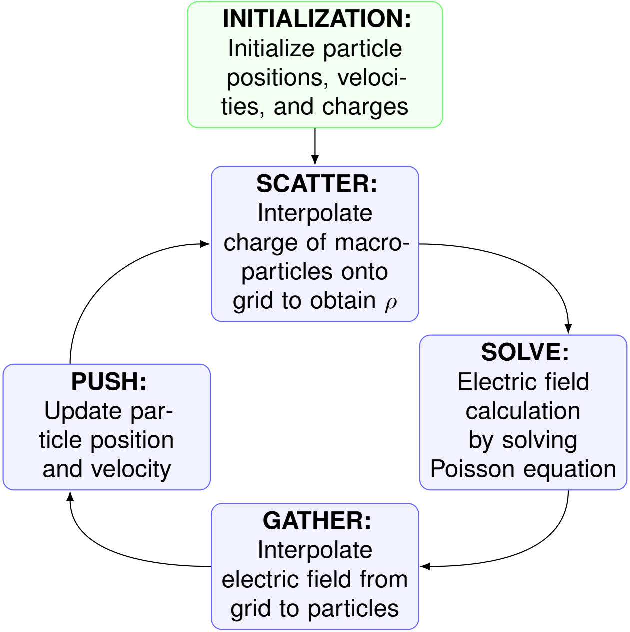


Source: PSI

- Needs novel algorithms with less approximations which can take advantage of the massive computing power of GPUs to improve our understanding of physical phenomena

Particle-In-Cell (PIC) Scheme

In the Electrostatic approximation



Vlasov-Poisson system

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

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

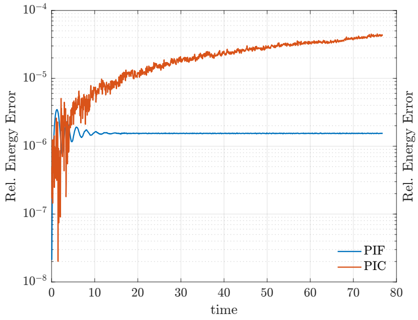
- Commonly used method for the simulation of kinetic plasmas
- Simple and highly parallelizable
- Computationally efficient compared to both continuum kinetic codes and pure particle codes

Why Particle-in-Fourier (PIF) Schemes?

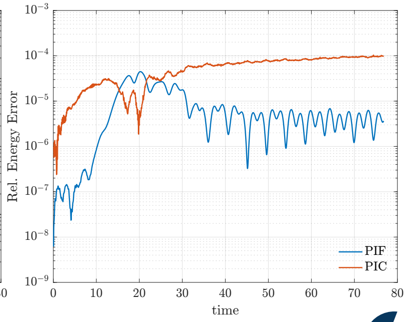
PIC

- Aliasing, Grid-heating, loss of energy conservation
- Only second-order with typical B-spline shape functions

Particle-in-Fourier: Removes the above issues by scatter and gather directly in the Fourier space without the use of a grid. Charge, momentum and energy (up to time discretization error) conservation



Weak Landau damping



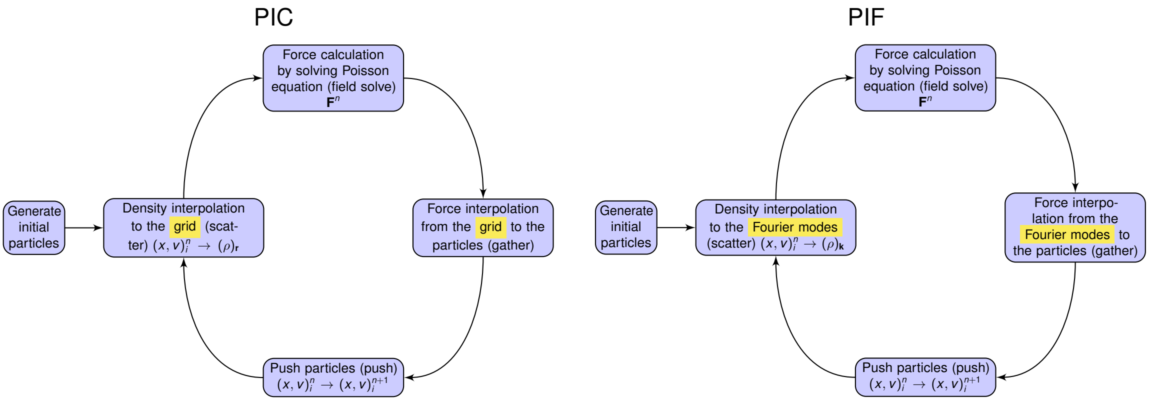
Two-stream instability

JÜLICH Forschungszentrum

PIF and PIC algorithms

In the Electrostatic approximation

The discrete force field is **F**. A particle i is denoted by $(\mathbf{x}, \mathbf{v})_i \in \Re^3 \times \Re^3$ and pushed from time step n to $n+1$:



PIF

- Non-uniform FFT (NUFFT) with a prescribed tolerance is used for scatter and gather[1]

[1] M. S. Mitchell, M. T. Miecnikowski, G. Beylkin, and S. E. Parker, Efficient Fourier basis particle simulation, JCP, 2019.

Parallelization of PIF

Spatial parallelization

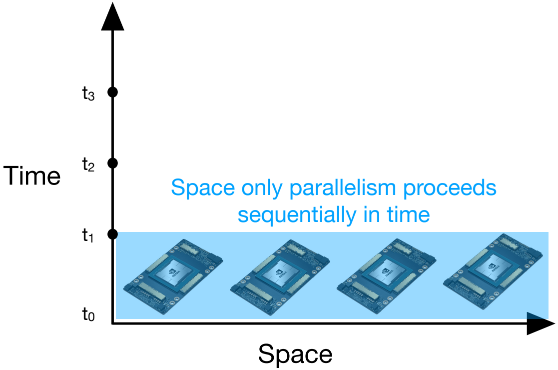
- Scatter and gather in PIF: Global operations due to lack of a grid in the real space
- Spatial parallelization: Particle decomposition (every core holds all the Fourier modes and only particles are split) as opposed to domain decomposition in PIC
- NUFFT complexity $\mathcal{O}(N_p + N_m log N_m)$: For high no. of modes NUFFT as well as All reduce in the scatter limits the scalability of PIF

Space-Time parallelization of PIF

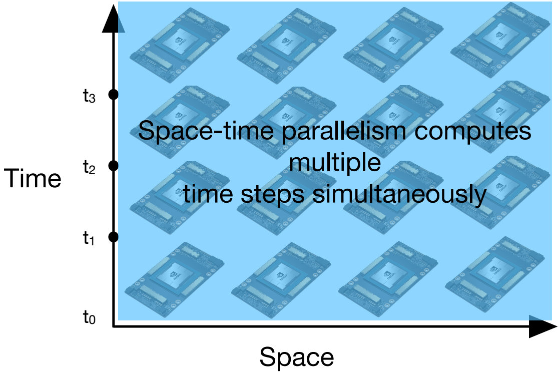
- Can we exploit parallelism in the time direction also to improve scalability of PIF?
- Goal: A Space-Time Parallelization framework for PIF towards Exascale simulations

JÜLICH Forschungszentrum

Space-Time Parallelization of PIF



Space-only parallelization



Space-Time parallelization

- Here: Parareal [2] for parallel-in-time integration
- Parareal: Predictor-corrector method where a cheap predictor provides the initial guesses for each of the time slices

[2] J. L. Lions, Y. Maday and G. Turinici, Résolution d'EDP par un schéma en temps «pararéel», 2001

Parallel-in-time integration with Parareal

Solving ODEs parallel-in-time with multiple shooting

We want to solve

$$\partial_t \boldsymbol{u}(t) = \boldsymbol{f}(t, \boldsymbol{u}(t)) \qquad t \in (0, T],$$
$$\boldsymbol{u}(0) = \boldsymbol{u}^0.$$

Using two propagation operators:

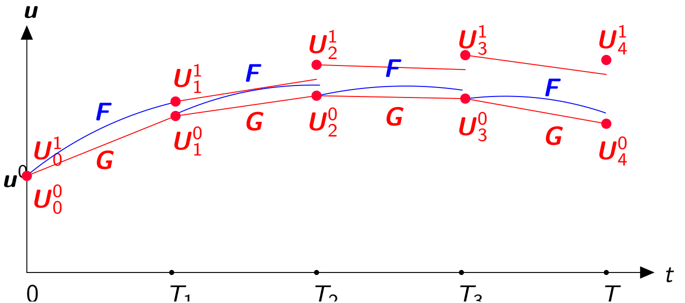
1. $\boldsymbol{F}(t_2, t_1, \boldsymbol{u}_1)$ as a **fine** approximation of solution $\boldsymbol{u}(t_2)$
2. $\boldsymbol{G}(t_2, t_1, \boldsymbol{u}_1)$ as a **coarse** approximation to the solution $\boldsymbol{u}(t_2)$

both with initial condition $\boldsymbol{u}(t_1) = \boldsymbol{u}_1$. Partition $(0, T]$ into subintervals $(T_{n-1}, T_n]$. Starting with initial coarse approximation $\boldsymbol{U}_n^0$ at $T_0, T_1, \ldots, T_N$ and then compute for $k = 0, 1, \ldots$

$$\boldsymbol{U}_0^{k+1} := \boldsymbol{u}^0,$$
$$\boldsymbol{U}_{n+1}^{k+1} := \boldsymbol{F}(T_{n+1}, T_n, \boldsymbol{U}_n^k) + \boldsymbol{G}(T_{n+1}, T_n, \boldsymbol{U}_n^{k+1}) - \boldsymbol{G}(T_{n+1}, T_n, \boldsymbol{U}_n^k)$$

JÜLICH Forschungszentrum

Parareal parallel-in-time integration



M. Gander, PinT Summer School, Michigan 2022.

For Parareal to give speedup

- **Cost of coarse propagator $\ll$ fine propagator** (as the coarse propagator runs serially in each iteration)
- **No. of iterations $\ll$ No. of time subdomains**
- Challenge: Finding a coarse propagator with both of the above properties

Parareal for PIF

Properties

- PIC as coarse propagator (predictor) and PIF as fine propagator (corrector) in Parareal
- NUFFT is typically $>$ 10X more costly than FFT which makes PIC cheap compared to PIF
- Other choice is to use PIF with lenient NUFFT tolerance as coarse propagator
- Same number of total particles are used both in coarse and fine propagators to avoid the need for interpolation in Parareal correction
- Our novel coarse propagators are different from the standard spatial coarsening which is not effective or possible with PIF schemes
- We can also coarsen the time step size in the coarse propagator depending on the test case

Caveat

Parareal applied on the entire time domain: Either too slow or has convergence issues for long time domains. To improve convergence it is applied in multiple blocks/cycles

Convergence theorem with PIC as a coarse propagator

Theorem

At iteration k of the parareal algorithm, we have the following error bound:

$$\mathbf{e}_{n+1}^{k} \leq \bar{C}^{n-k} \frac{\left(C_{grid} h^{\min(m+1,2)} + C_{noise} P_c^{-0.5} + C_{time} \Delta T \Delta t_g^{p}\right)^{k}}{k!} \prod_{j=1}^{k} (n+1-j)\delta.$$

- $\delta = \max_{n=1,\ldots,N} \mathbf{e}_n^0$, $\mathbf{e}_n^0$: initial error
- h: mesh size
- m: order of shape function
- P_c: number of particles per cell
- Δt_g: coarse time step size
- ΔT: length of n^{th} time subdomain

- p: order of time integrator
- C_{pic}: Lipschitz constant of the PIC scheme
- $\bar{C} = \max(1, C_{pic})$
- C_{grid}, C_{noise}: constants related to the density distribution and the shape function in the PIC scheme
- C_{time}: constant related to the time integrator and the smoothness of the distribution with respect to time

JÜLICH Forschungszentrum

Convergence theorem with PIF as a coarse propagator

Theorem

At iteration k of the parareal algorithm, we have the following error bound:

$$\mathbf{e}_{n+1}^{k} \leq \bar{C}^{n-k} \frac{\left(C_{nufft}\varepsilon + C_{time}\Delta T \Delta t_g^p\right)^k}{k!} \prod_{j=1}^{k}(n+1-j)\delta$$

- $\delta = \max_{n=1,\ldots,N} \mathbf{e}_n^0$, $\mathbf{e}_n^0$: initial error
- ε: Tolerance of NUFFT
- Δt_g: coarse time step size
- ΔT: length of n^{th} time subdomain
- p: order of time integrator

- C_{pif}: Lipschitz constant of the PIF scheme
- $\bar{C} = \max(1, C_{pif})$
- C_{nufft}: constant related to the NUFFT scheme
- C_{time}: constant related to the time integrator and the smoothness of the distribution with respect to time

JÜLICH Forschungszentrum

Implementation

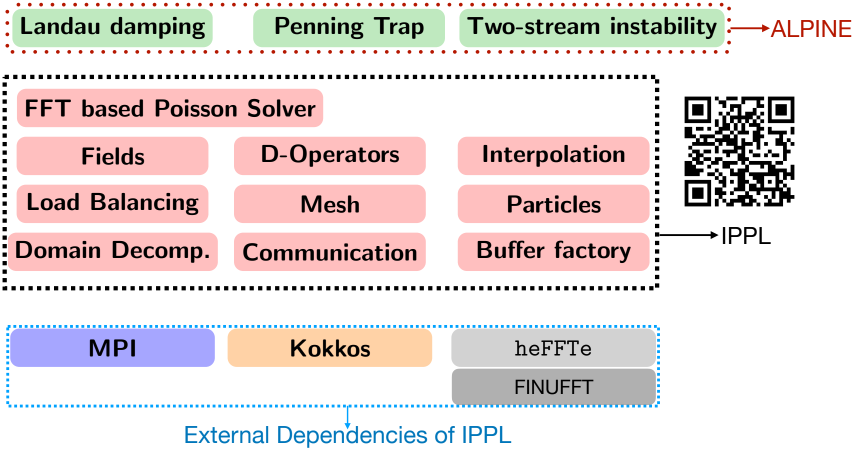
IPPL

- The space-time parallel PIF is implemented in Independent Parallel Particle Layer (IPPL) (Muralikrishnan et al. arXiv:2205.11052)
- IPPL is an open source performance portable particle-mesh framework in modern C++ (C++20) targeting extreme scale heterogeneous computing architectures
- Used in the particle accelerator library OPAL. Supports novel algorithms development for exascale architectures by providing a set of mini-apps (ALPINE)

Key features of IPPL

- Performance portable
- Scalability to thousands of GPUs and CPU cores
- Dimension independent (1 to 6 dimensions)
- Mixed-precision and mixed-execution spaces
- In-situ visualization and analysis

ALPINE (A set of performance portable pLasma physics Particle-in-cell mINi-apps for Exascale computing)[3]



[3] S. Muralikrishnan, M. Frey, A. Vinciguerra, M. Ligotino, A. J. Cerfon, M. Stoyanov, R. Gayatri and A. Adelmann, Scaling and performance portability of the particle-in-cell scheme for plasma physics applications through mini-apps targeting exascale architectures, In Proceedings of the 2024 SIAM Conference on Parallel Processing for Scientific Computing (PP) (pp. 26-38). Society for Industrial and Applied Mathematics.

Penning trap and two-stream instability

Penning trap

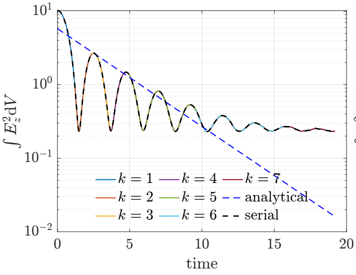
Two-stream instability

Numerical Results

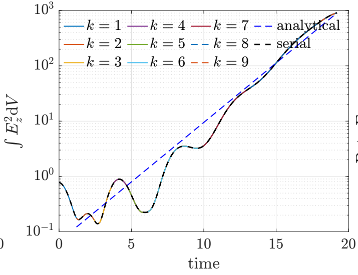
Setup

- 64^3 Fourier modes
- Total number of particles N_p = 2,621,440
- 3D-3V
- Leap Frog/Boris time integrator

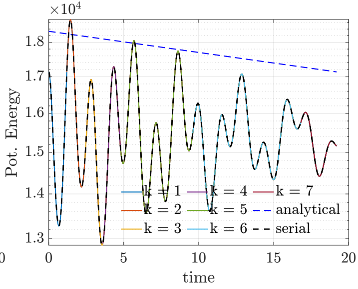
- $\Delta t = 0.003125$, 6144 time steps
- Tolerance for Parareal convergence: 10^{-8}
- Fine propagator: PIF with NUFFT tolerance 10^{-7}
- Coarse propagator: PIC
- 4 or 16 GPUs in space and 16 GPUs in time



Landau damping

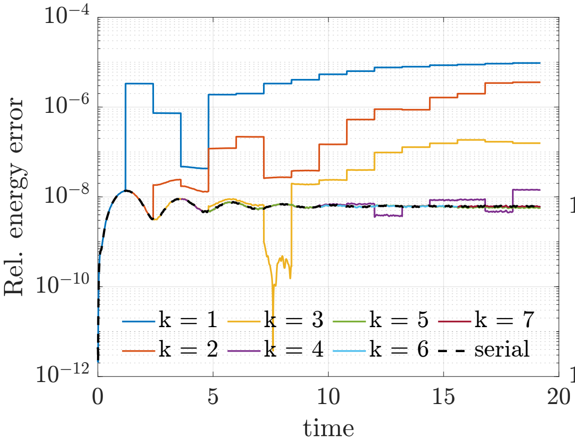


Two-stream instability

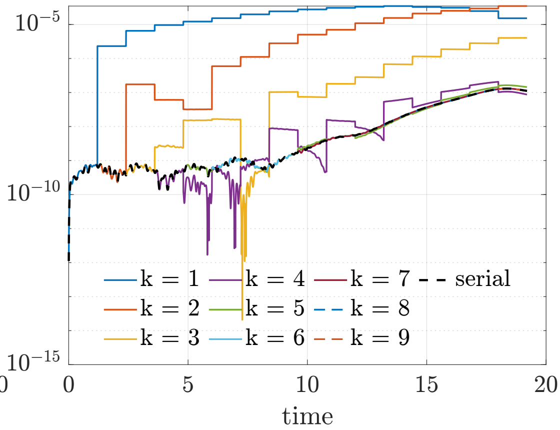


Penning trap

Energy conservation



Landau damping



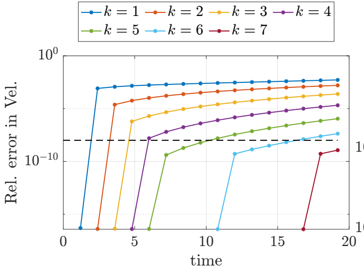
Two-stream instability

(Charge and momentum show expected conservation properties, too)

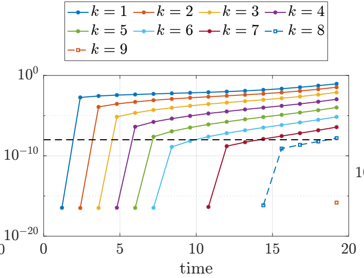
JÜLICH Forschungszentrum

Parareal convergence

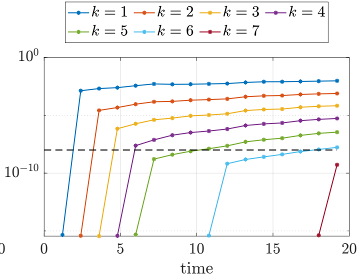
Error vs. number of iterations



Landau damping

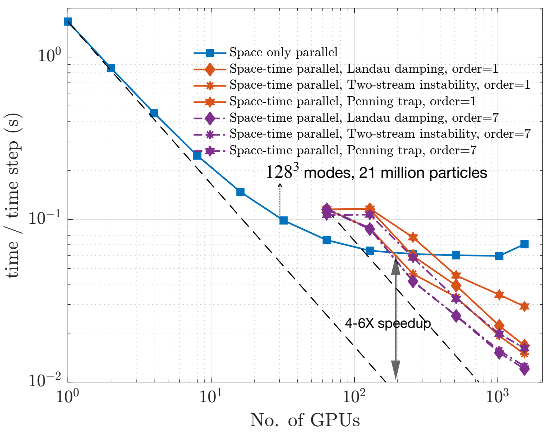


Two-stream instability

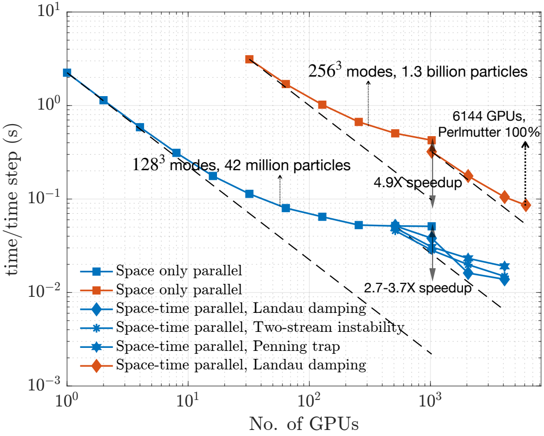


Penning trap

Scaling results on NVIDIA A100s



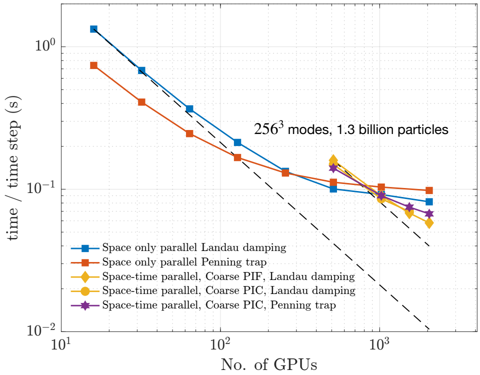
JUWELS Booster at JSC

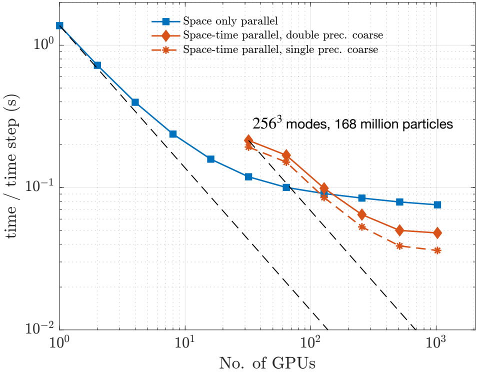


Perlmutter at NERSC

JÜLICH Forschungszentrum

Scaling results on NVIDIA GH200s at Alps (CSCS)





- Improvement in hardware (i.e. from A100 to GH200) as well as the NUFFT library improved the absolute time to solution
- Now the scaling of time parallelization needs improvement. For example by utilizing multilevel PinT

JÜLICH Forschungszentrum

Conclusions and Future Work

Conclusions

- Parareal with PIF as fine propagator and PIC or PIF with lenient NUFFT tolerance as coarse propagator converges
- Convergence proved theoretically and verified numerically
- Massive space-time parallel results shown on JUWELS Booster, Perlmutter and Alps with time parallelism giving additional speedup compared to space-only parallelization
- Space-Time parallel PIF shows promise for exascale kinetic plasma simulations

Ongoing/Future work

- Investigate other PinT methods (e.g. MGRIT, PFASST) for PIF schemes
- Apply the scheme to more realistic applications where high accuracy and conservation are important
- See if PIF/PIC Parareal can be applied to other domains such as fluid dynamics

Conclusions and Future Work

Conclusions

- Parareal with PIF as fine propagator and PIC or PIF with lenient NUFFT tolerance as coarse propagator converges
- Convergence proved theoretically and verified numerically
- Massive space-time parallel results shown on JUWELS Booster, Perlmutter and Alps with time parallelism giving additional speedup compared to space-only parallelization
- Space-Time parallel PIF shows promise for exascale kinetic plasma simulations

Ongoing/Future work

- Investigate other PinT methods (e.g. MGRIT, PFASST) for PIF schemes
- Apply the scheme to more realistic applications where high accuracy and conservation are important
- See if PIF/PIC Parareal can be applied to other domains such as fluid dynamics

Thank you for your attention!