Some recent improvements of parallel-in-time algorithms

TUHH
Technische Universität Hamburg

7.11.2023

https://time-x.eu/

EuroHPC Joint Undertaking

TIME-X

SPONSORED BY THE Federal Ministry of Education and Research

Gefördert vom Bundesministerium für Bildung und Forschung (BMBF) unter Förderkennzeichen 16HPC048.

This project has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 955701. The JU receives support from the European Union's Horizon 2020 research and innovation programme and Belgium, France, Germany, and Switzerland.

Daniel Ruprecht

Results by Abdul Ibrahim, Thibaut Lunet, Thomas Baumann

The problem with parallelizing only in space

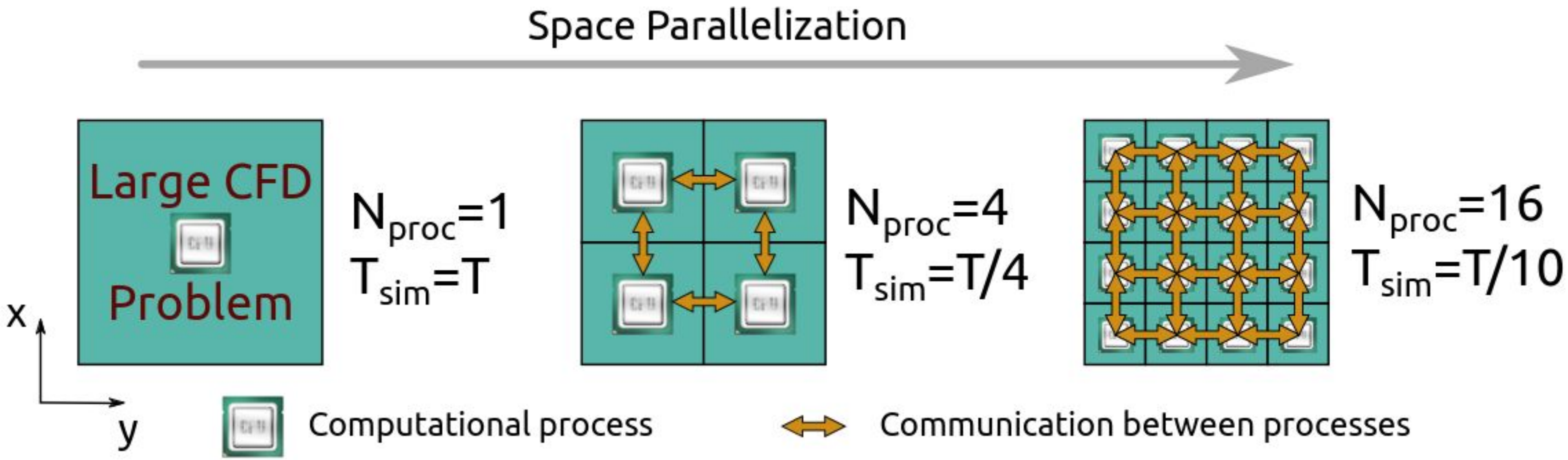
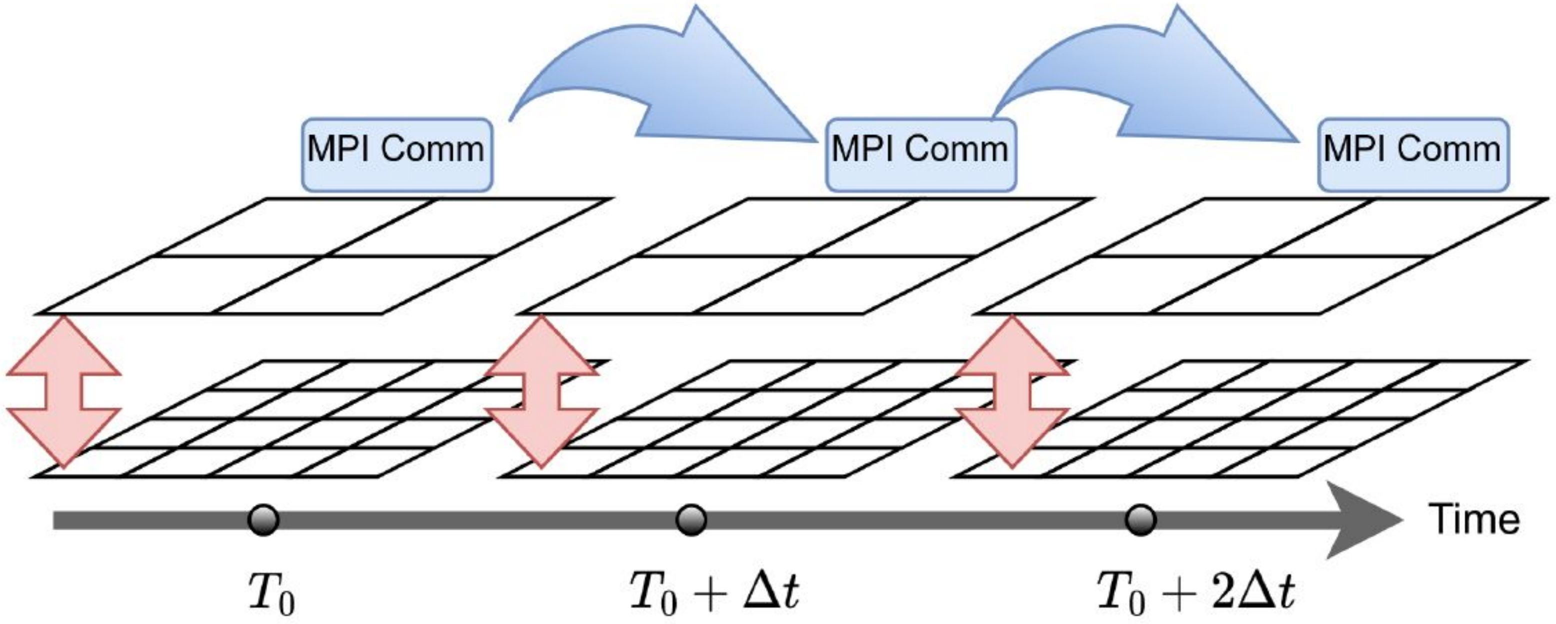


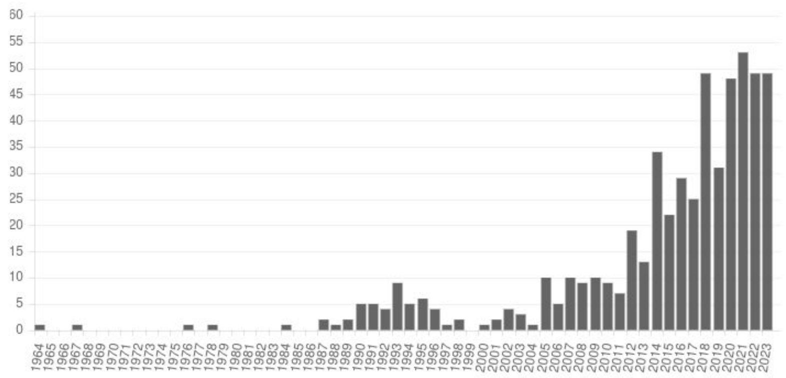
Figure courtesy T. Lunet

- Spatial strong scaling eventually saturates
- Even with perfect weak scaling in space, increased time resolution still increases solution times

Parallel-across-the-steps: the basic idea



- First idea published in 1964
- Small surge of interest in 1990's
- rapid growth of the field since 2001



We cannot completely avoid serial dependency in time .... but we can *relax* it.

https://parallel-in-time.org/references/index.html

What PinT promises to deliver

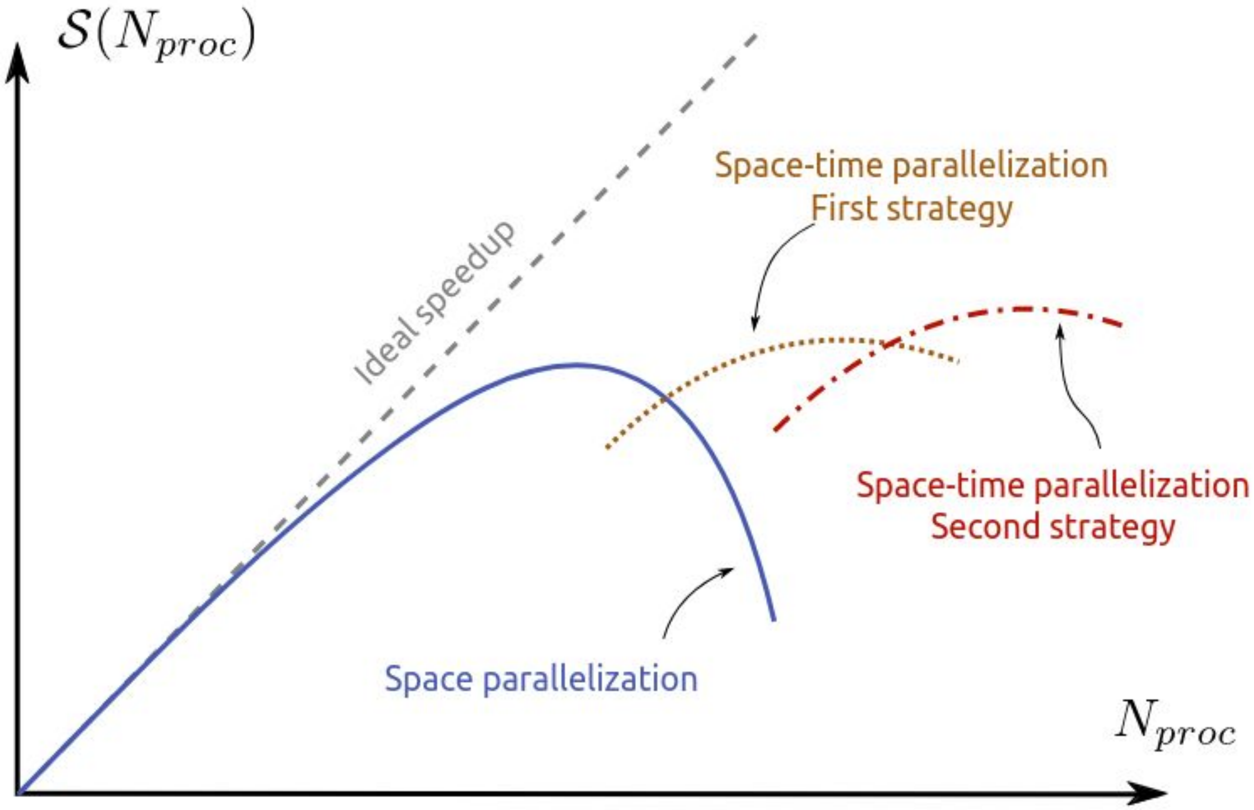
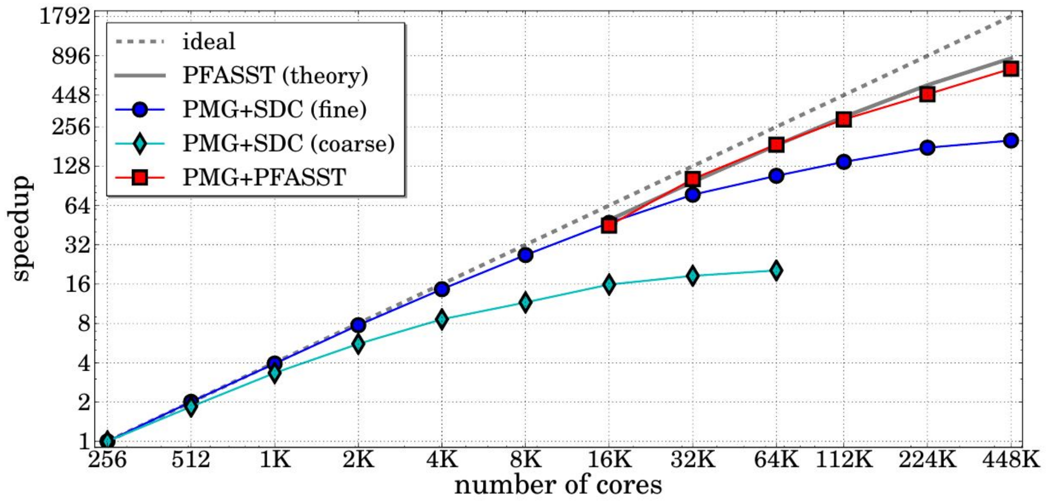


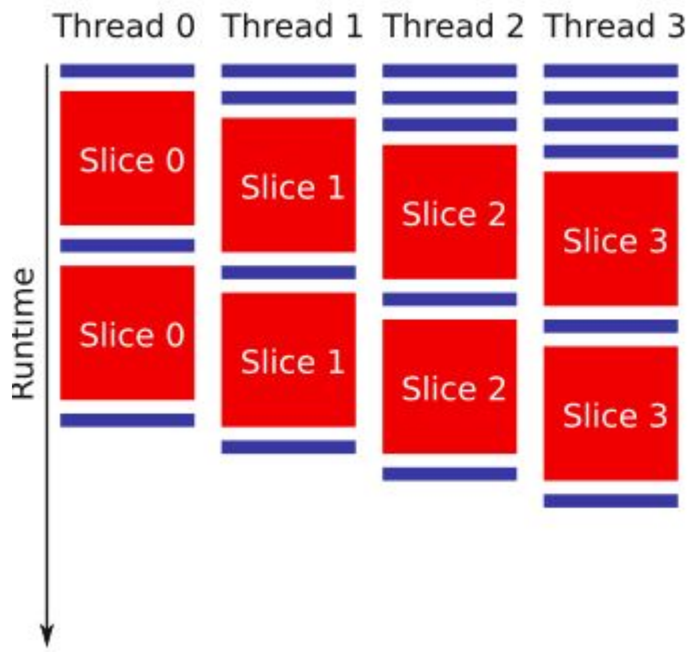
Figure courtesy T. Lunet

We can see this in practice! Sometimes.



* D. Ruprecht, R. Speck, M. Emmett, M. Bolten, and R. Krause, "*Poster: Extreme-scale space-time parallelism*," in Proceedings of the 2013 Conference on High Performance Computing Networking, Storage and Analysis Companion, ser. SC '13 Companion, Denver, Colorado, USA, 2013.

PinT: The challenge



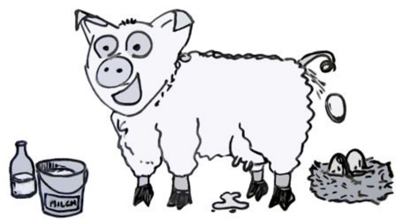
The coarse propagator is a serial bottleneck:

$$s(N_p) \leq \min\left(\frac{N_p}{N_{it}}, \frac{\text{runtime fine}}{\text{runtime coarse}}\right)$$

This is for *Parareal*, but similar bounds hold for other algorithms like *MGRIT* or *PFASST*

Thus, coarse propagator needs to be both fast and reasonably accurate.

Proverbial "*Eier legende Wollmilchsau*" … animal that lays eggs, gives milk and provides wool.



Using ML to build coarse propagators

PDE Residual (used in loss function of PINN)

risk-free interest rate

$$f(V) = \frac{\partial V}{\partial t}(S,t) + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2}(S,t) + rS\frac{\partial V}{\partial S}(S,t) - rV(S,t) = 0.$$

volatility

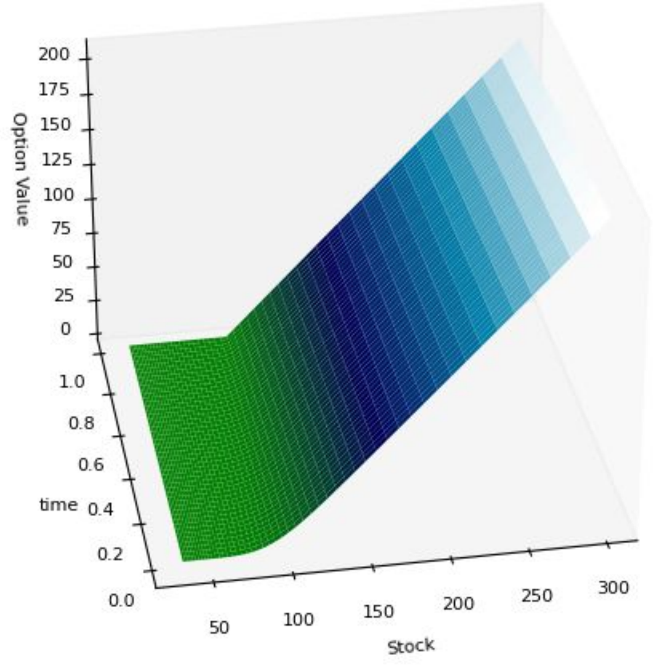
- Boundary conditions

$$V(t,S) \sim 0 \text{ as } S \to \infty, \text{ for fixed } t.$$

$$V(t,0) = 0 \text{ for all } t.$$

- Expiration / end time condition

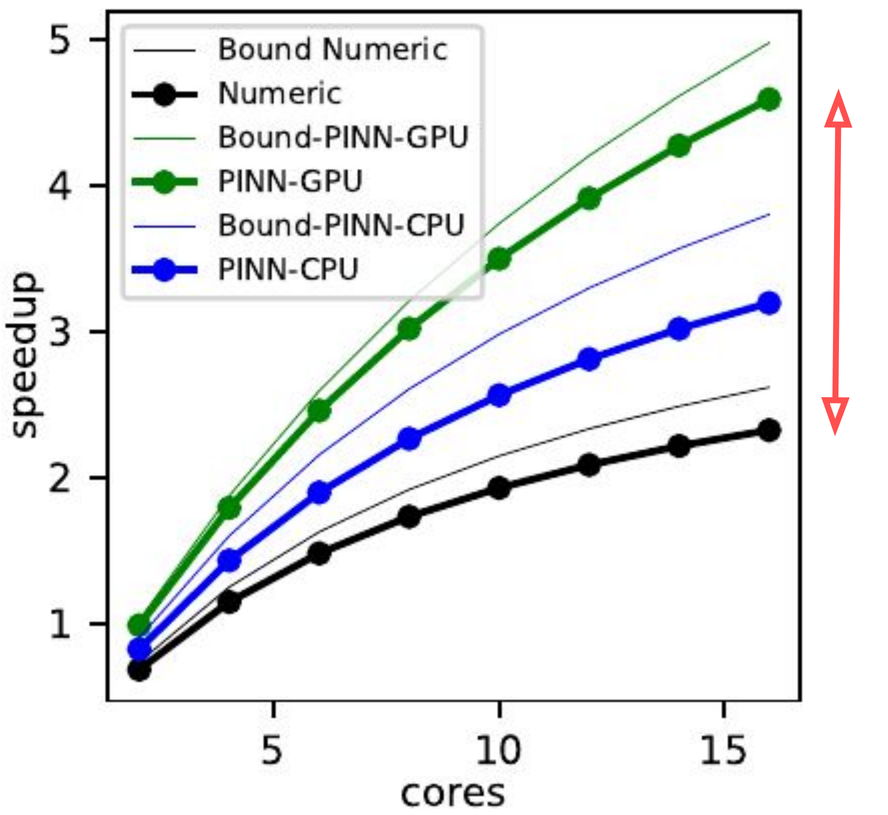
$$V(T,S) = \max(S-K,0) \text{ for all } S$$

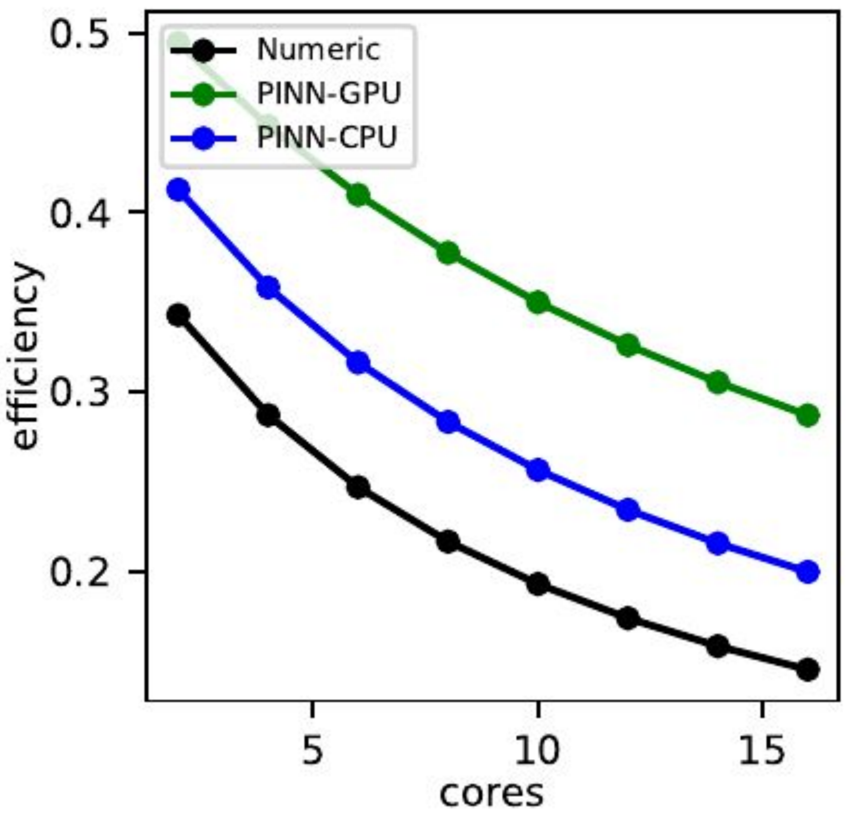


Work by: Abdul Qadir Ibrahim, M. Sc.

A. Q. Ibrahim, S. Götschel, and D. Ruprecht, “*Parareal with a physics-informed neural network as coarse propagator*,” in **Euro-Par 2023: Parallel Processing**, Springer Nature Switzerland, 2023, pp. 649–663.

Using ML to build coarse propagators





Work by: Abdul Qadir Ibrahim, M. Sc.

A. Q. Ibrahim, S. Götschel, and D. Ruprecht, “*Parareal with a physics-informed neural network as coarse propagator*,” in **Euro-Par 2023: Parallel Processing**, Springer Nature Switzerland, 2023, pp. 649–663.

PINN coarse propagator on GPU yields more than twice the Parareal speedup than a mesh-based coarse propagator.

Performance modelling and comparison

TUHH

$$u_{n+1}^{k+1} = B_0^0 u_n^k + B_1^0 u_{n+1}^k + B_0^1 u_n^{k+1} + \dots$$

Algorithm	$\mathbf{B}_1^0\ (\boldsymbol{u}_{n+1}^k)$	$\mathbf{B}_0^0\ (\boldsymbol{u}_n^k)$	$\mathbf{B}_0^1\ (\boldsymbol{u}_n^{k+1})$
damped Block Jacobi	$\mathbf{I} - \omega\mathbf{I}$	$\omega\boldsymbol{\phi}^{-1}\boldsymbol{\chi}$	–
ABJ	$\mathbf{I} - \tilde{\boldsymbol{\phi}}^{-1}\boldsymbol{\phi}$	$\tilde{\boldsymbol{\phi}}^{-1}\boldsymbol{\chi}$	–
ABGS	$\mathbf{I} - \tilde{\boldsymbol{\phi}}^{-1}\boldsymbol{\phi}$	–	$\tilde{\boldsymbol{\phi}}^{-1}\boldsymbol{\chi}$
Parareal	–	$(\boldsymbol{\phi}^{-1} - \tilde{\boldsymbol{\phi}}^{-1})\boldsymbol{\chi}$	$\tilde{\boldsymbol{\phi}}^{-1}\boldsymbol{\chi}$
TMG	$(1-\omega)(\mathbf{I} - \mathbf{T}_C^F\boldsymbol{\phi}_C^{-1}\mathbf{T}_F^C\boldsymbol{\phi})$	$\omega(\boldsymbol{\phi}^{-1} - \mathbf{T}_C^F\boldsymbol{\phi}_C^{-1}\mathbf{T}_F^C)\boldsymbol{\chi}$	$\mathbf{T}_C^F\boldsymbol{\phi}_C^{-1}\mathbf{T}_F^C\boldsymbol{\chi}$
TMG$_c$	–	$(\boldsymbol{\phi}^{-1} - \mathbf{T}_C^F\tilde{\boldsymbol{\phi}}_C^{-1}\mathbf{T}_F^C)\boldsymbol{\chi}$	$\mathbf{T}_C^F\tilde{\boldsymbol{\phi}}_C^{-1}\mathbf{T}_F^C\boldsymbol{\chi}$
TMG$_f$	$(\mathbf{I} - \mathbf{T}_C^F\boldsymbol{\phi}_C^{-1}\mathbf{T}_F^C\boldsymbol{\phi})(\mathbf{I} - \tilde{\boldsymbol{\phi}}^{-1}\boldsymbol{\phi})$	$(\tilde{\boldsymbol{\phi}}^{-1} - \mathbf{T}_C^F\boldsymbol{\phi}_C^{-1}\mathbf{T}_F^C\boldsymbol{\phi}\tilde{\boldsymbol{\phi}}^{-1})\boldsymbol{\chi}$	$\mathbf{T}_C^F\boldsymbol{\phi}_C^{-1}\mathbf{T}_F^C\boldsymbol{\chi}$
PFASST	$(\mathbf{I} - \mathbf{T}_C^F\tilde{\boldsymbol{\phi}}_C^{-1}\mathbf{T}_F^C\boldsymbol{\phi})(\mathbf{I} - \tilde{\boldsymbol{\phi}}^{-1}\boldsymbol{\phi})$	$(\tilde{\boldsymbol{\phi}}^{-1} - \mathbf{T}_C^F\tilde{\boldsymbol{\phi}}_C^{-1}\mathbf{T}_F^C\boldsymbol{\phi}\tilde{\boldsymbol{\phi}}^{-1})\boldsymbol{\chi}$	$\mathbf{T}_C^F\tilde{\boldsymbol{\phi}}_C^{-1}\mathbf{T}_F^C\boldsymbol{\chi}$

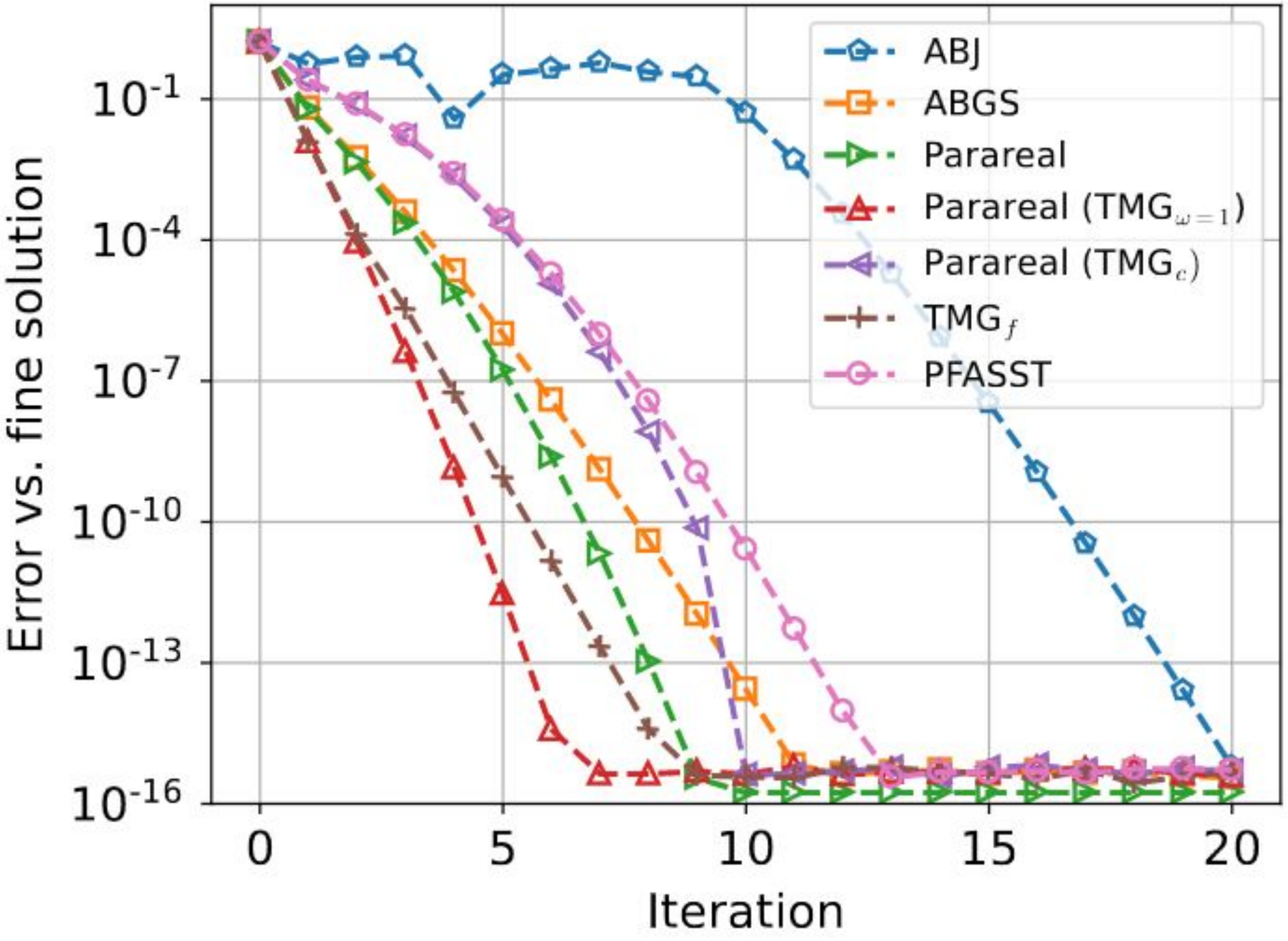
Work by: Dr Thibaut Lunet.

M. J. Gander, T. Lunet, D. Ruprecht, and R. Speck, "*A unified analysis framework for iterative parallel-in-time algorithms*," **SIAM Journal on Scientific Computing**, vol. 45, no. 5, 2275–A2303, 2023.

Can write and analyse different iterative PinT algorithms in a common framework (for linear problems)

Performance modelling and comparison

Can predict how different methods will converge



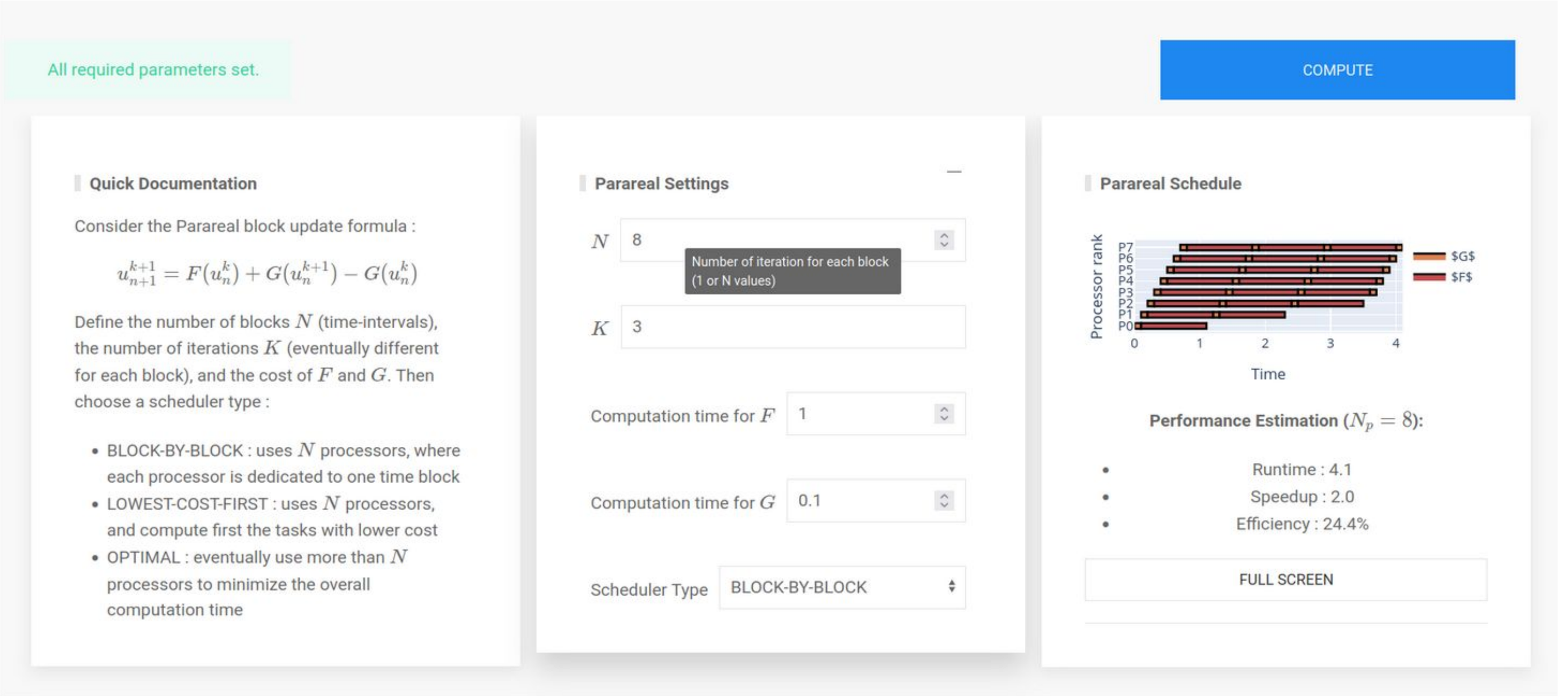
Work by: Dr Thibaut Lunet.

M. J. Gander, T. Lunet, D. Ruprecht, and R. Speck, "*A unified analysis framework for iterative parallel-in-time algorithms*," **SIAM Journal on Scientific Computing**, vol. 45, no. 5, 2275–A2303, 2023.

Performance modelling and comparison

TUHH

https://time-x.eu/educational-website-for-first-performance-analysis-of-pint-algorithms/



Combined with a unified performance model by Bolten et al. 2023 from BU Wuppertal, we can predict performance of PinT methods on realistic HPC systems.

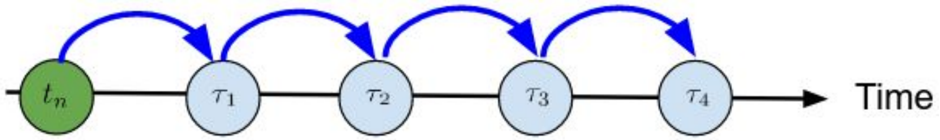
Work by: Dr Thibaut Lunet.

M. J. Gander, T. Lunet, D. Ruprecht, and R. Speck, "*A unified analysis framework for iterative parallel-in-time algorithms*," **SIAM Journal on Scientific Computing**, vol. 45, no. 5, 2275–A2303, 2023.

M. Bolten, S. Friedhoff, and J. Hahne, "Task graph-based performance analysis of parallel-in-time methods," *Parallel Computing*, vol. 118, p. 103050, 2023

Adaptivity and soft-fault resilience

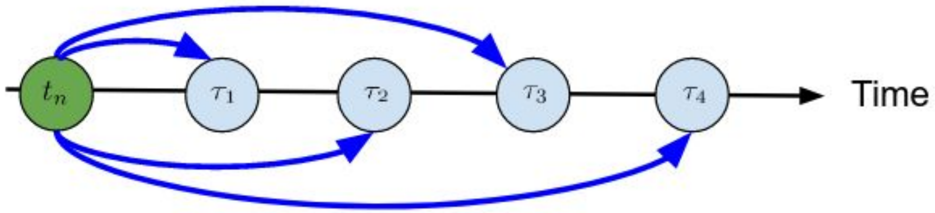
Serial SDC



$$u_m^{k+1} = u_n + \Delta t \sum_{j=1}^{M} q_{m,j} f(u_j^k) + \Delta t \sum_{j=1}^{\mathbf{m}} \Delta\tau_j [f(\tau_j, u_j^{k+1}) - f(\tau_j, u_j^k)]$$

- quadrature terms
- correction terms

Parallel SDC



$$u_m^{k+1} = u_n + \Delta t \sum_{j=1}^{M} q_{m,j} f(u_j^k) + \Delta t \alpha_m [f(\tau_m, u_m^{k+1}) - f(\tau_m, u_m^k)]$$

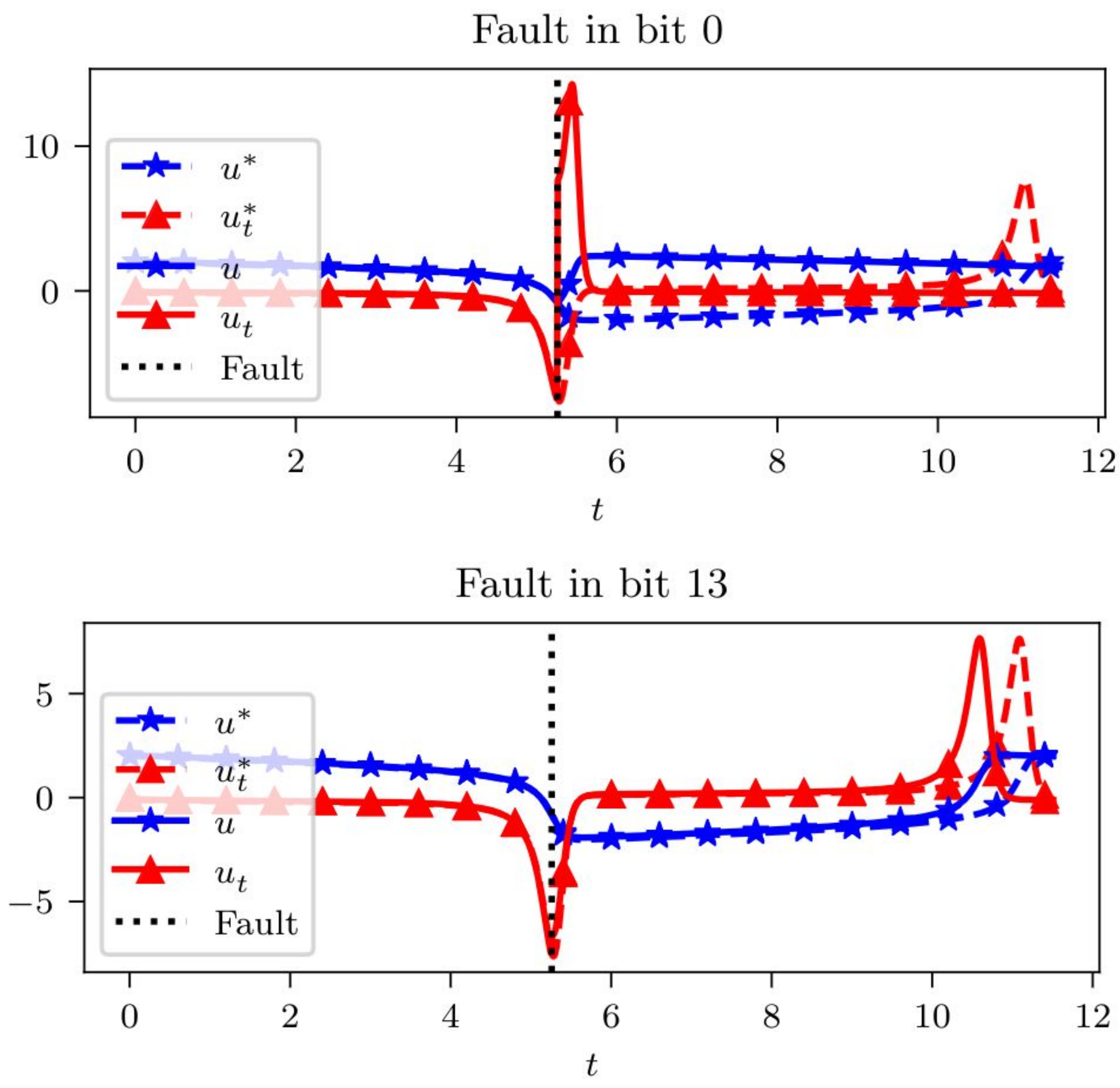
- quadrature terms
- correction terms

Work by: Thomas Baumann.

T. Baumann, S. Götschel, T. Lunet, D. Ruprecht, and R. Speck, "*Adaptive time step selection for spectral deferred corrections*," in preparation, 2024

Adaptivity and soft-fault resilience

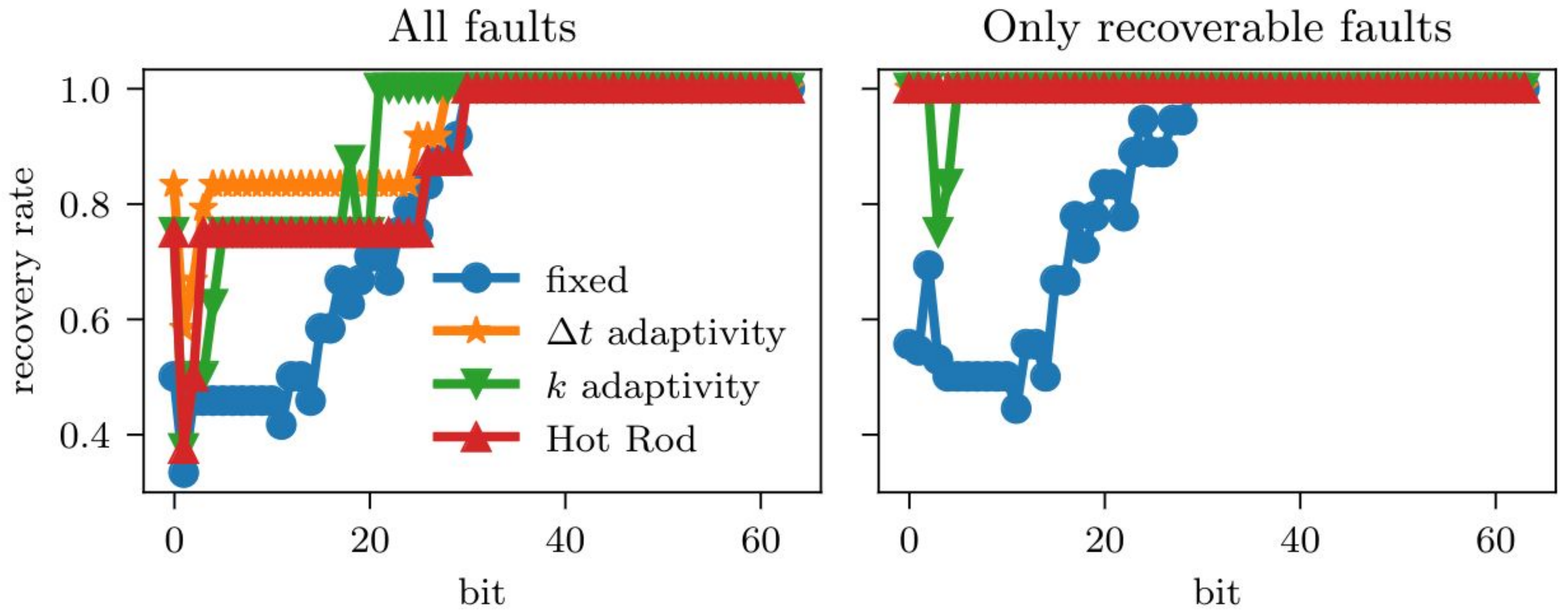
- Iterative nature of algorithm can protect against “bit flips”
- Flips show up in residual.
- Can mitigate, restart or continue to iterate and hope for the best!



Work by: Thomas Baumann.

T. Baumann, S. Götschel, T. Lunet, D. Ruprecht, and R. Speck, “*Adaptive time step selection for spectral deferred corrections*,” in preparation, 2024

Adaptivity and soft-fault resilience



Work by: Thomas Baumann.

T. Baumann, S. Götschel, T. Lunet, D. Ruprecht, and R. Speck, “*Adaptive time step selection for spectral deferred corrections*,” in preparation, 2024

Suitable mitigation strategies can catch almost all faults in later bits and fix all recoverable faults.

Thanks.

Hamburg University of Technology (TUHH)
Chair for Computational Mathematics
Institute of Mathematics
Am Schwarzenberg-Campus 1
21073 Hamburg
Tel.: +49 40 42878-3330
www.tuhh.de

tuhh.de

TUHH
Technische
Universität
Hamburg