INSTITUTE OF MATHEMATICS

Generic Tool for Preliminary Performance Analysis of Iterative Parallel-in-Time Methods

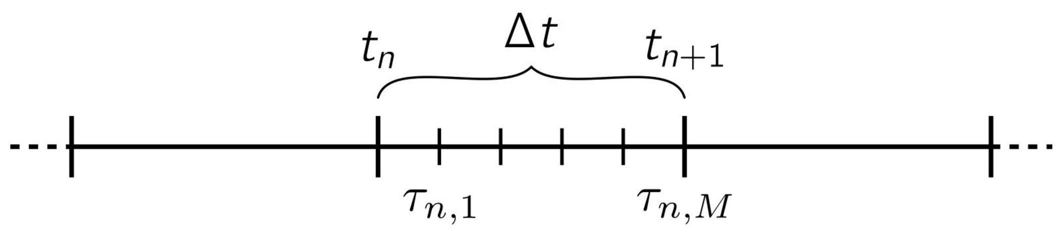
T. Lunet (TUHH), J. Hahne (University of Wuppertal)

Toward a common formalism ...

Focus on an elementary ODE :

$$\frac{du}{dt} = \lambda u, \quad \lambda \in \mathbb{C}, \quad t \in [0, T]$$

1) Decompose time domain in N blocks :



Block Variable : $\boldsymbol{u}_n = [u_{n,1}, u_{n,2}, ..., u_{n,M}]^T$

2) Define the **Block Operators**

$$\phi(\boldsymbol{u}_{n+1}) = \chi(\boldsymbol{u}_n)$$

ϕ is a bijective operator
- Runge-Kutta type
- Multistep
- Collocation based, SDC, …

χ builds the initial solution for next block

3) Build the **Block Problem**

$$\begin{pmatrix} \phi & & & \\ -\chi & \phi & & \\ & \ddots & \ddots & \\ & & -\chi & \phi \end{pmatrix} \begin{bmatrix} \boldsymbol{u}_1 \\ \boldsymbol{u}_2 \\ \vdots \\ \boldsymbol{u}_N \end{bmatrix} = \begin{bmatrix} \chi(u_0 \mathbb{1}) \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$\Leftrightarrow \quad \mathbf{A}\boldsymbol{u} = \boldsymbol{f}$$

Iterative Parallel-in-Time algorithms simply solve a Block Problem iteratively

Main Idea : for a preconditioned iteration

$$\boldsymbol{u}^{k+1} = \boldsymbol{u}^k + \mathbf{P}^{-1}(\boldsymbol{f} - \mathbf{A}\boldsymbol{u}^k)$$

→write the block update component wise : **Block Iteration**

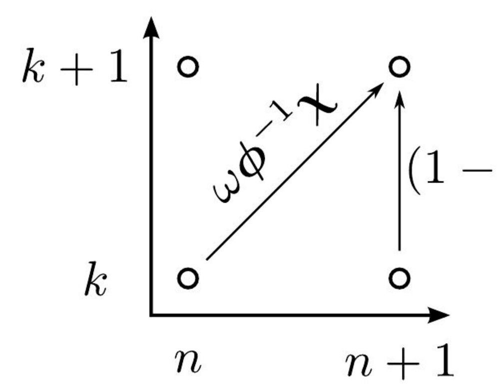
$$\boldsymbol{u}_{n+1}^{k+1} = \mathbf{B}_0^0 \boldsymbol{u}_n^k + \mathbf{B}_1^0 \boldsymbol{u}_{n+1}^k + \mathbf{B}_0^1 \boldsymbol{u}_n^{k+1} + \dots$$

→same approach for time multigrid with coarse grid correction

Example : Block Jacobi Smoothing

$$\mathbf{P}_{BJ} = \frac{1}{\omega} \begin{pmatrix} \phi & & \\ & \ddots & \\ & & \phi \end{pmatrix}, \quad \omega > 0$$

KN-graph :



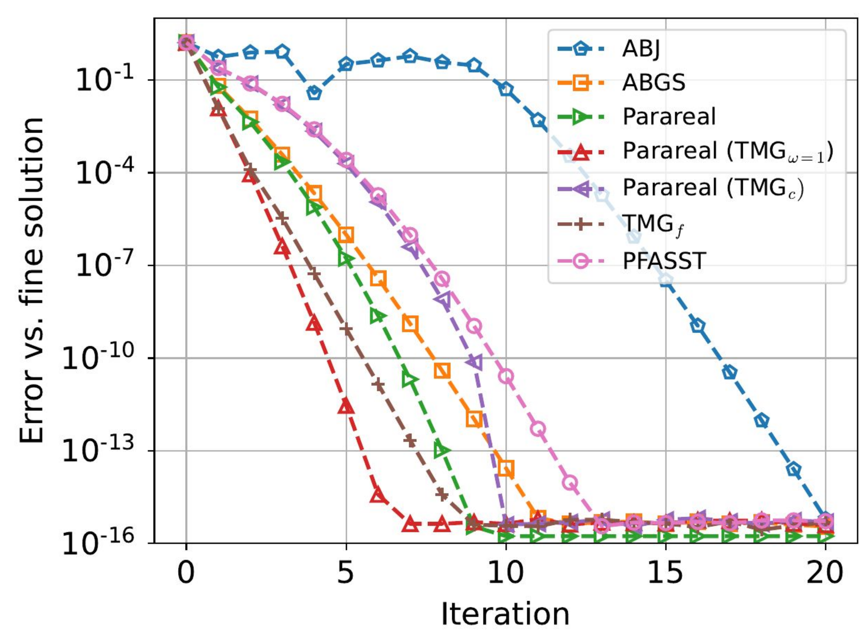
⇒ Block Iteration formula :

$$\boldsymbol{u}_{n+1}^{k+1} = (1-\omega)\boldsymbol{u}_{n+1}^k + \omega\phi^{-1}\chi\boldsymbol{u}_n^k$$

Possibility to write most iterative methods as Block Iterations (Parareal, MGRIT, PFASST, Time Multigrid, …)

Use of the block representation to analyze Parallel-in-Time error and estimate speedup

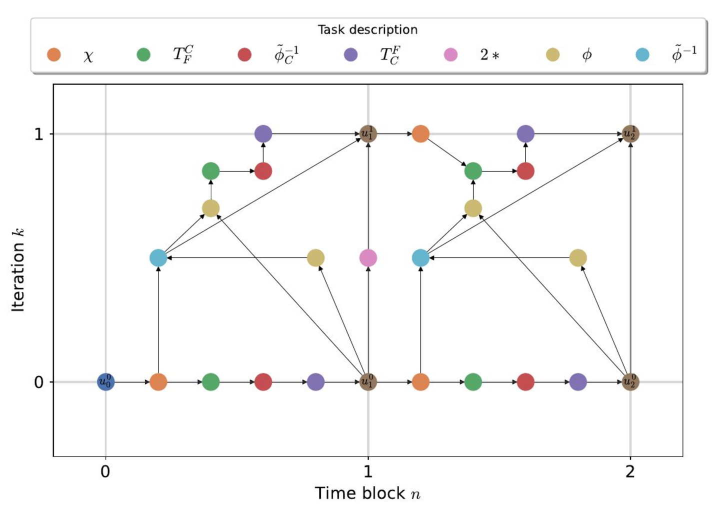
Error Analysis



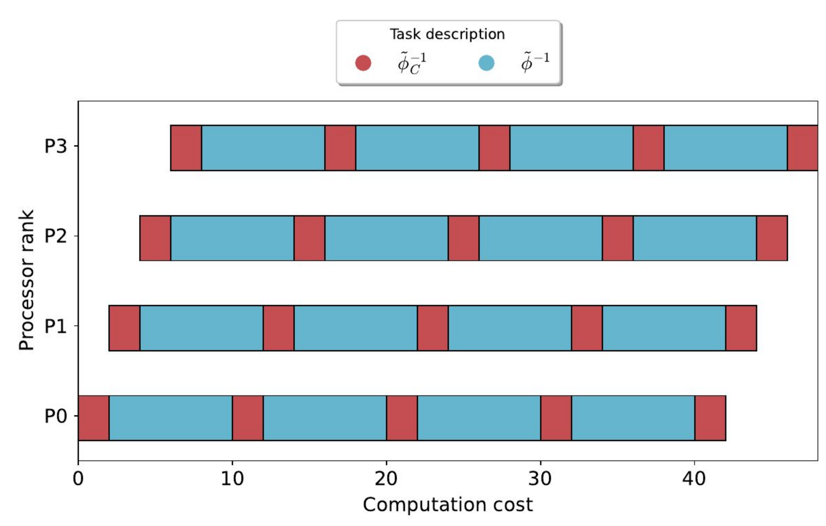
Example : convergence comparison of different algorithms with
- same time integration methods
- same problem parameters

In addition : generic error bounds for more in-depth analysis …

Task Graph and Scheduling

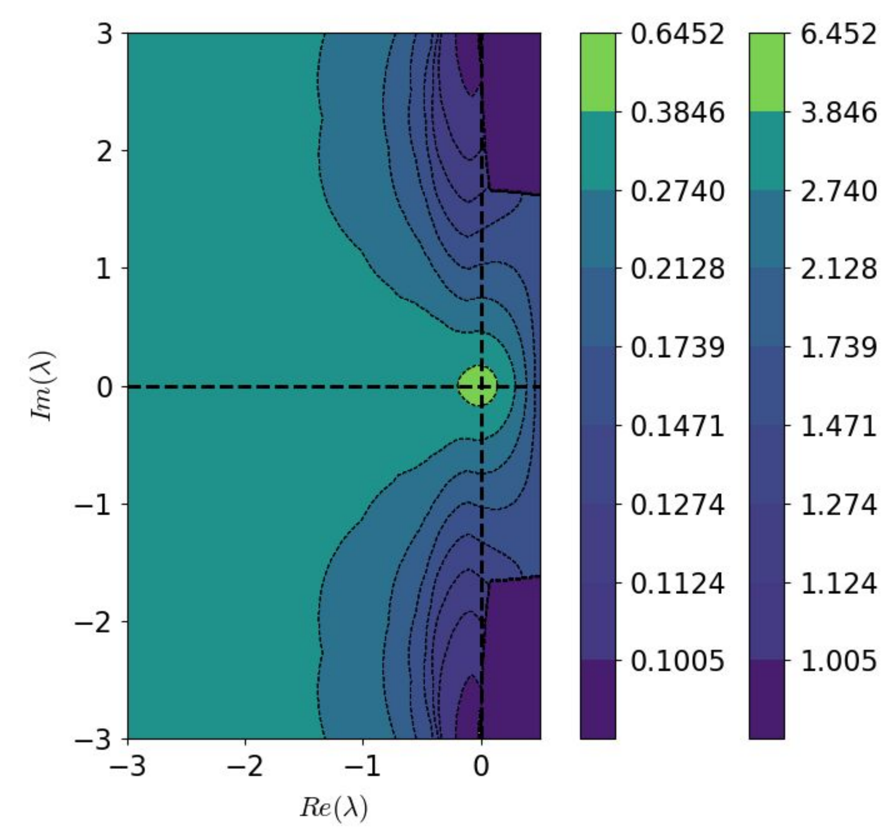


Task graph for one iteration of PFASST on 2 blocks



Schedule graph for 4 iterations of PFASST on 4 blocks

Speedup Estimation



Parallel efficiency and speedup for Parareal in the complex plane.
Use of Backward Euler with a 1/20 coarsening

Open Source Code & Web Application
https://github.com/Parallel-in-Time/time4apint

TUHH Hamburg University of Technology

Technische Universität Hamburg
Institut für Mathematik
Lehrstuhl Computational Mathematics
Am Schwarzenberg-Campus 3,
Gebäude E
21073 Hamburg

The Time-X 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.

This project also received funding from the German Federal Ministry of Education and Research (BMBF) grant 16HPC048.

EuroHPC Joint Undertaking

SPONSORED BY THE Federal Ministry of Education and Research