

# A multigrid perspective on PFASST

November 28, 2016 | Dieter Moser, Robert Speck, Matthias Bolten  
Jülich Supercomputing Centre, Forschungszentrum Jülich GmbH

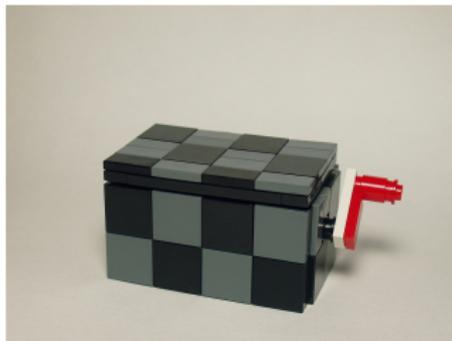
# Motivation

<b>50 Years of Time Parallel Time Integration</b> .....	<b>1</b>
Martin J. Gander	
1 Introduction .....	4
2 Shooting Type Time Parallel Methods .....	5
2.1 Nievergelt 1964 .....	5
2.2 Bellen and Zennaro 1989 .....	7
2.3 Chartier and Philippe 1993 .....	9
2.4 Saha, Stadel and Tremaine 1996 .....	10
2.5 Lions, Maday and Turinici 2001 .....	12
3 Domain Decomposition Methods in Space-Time .....	16
3.1 Picard and Lindelöf 1893/1894 .....	16
3.2 Lelarasme, Ruehli and Sangiovanni-Vincentelli 1982 .....	17
3.3 Gander 1996 .....	20
3.4 Gander, Halpern and Nataf 1999 .....	21
3.5 Recent Developments .....	23
4 Multigrid Methods in Space-Time .....	23
4.1 Hackbusch 1984 .....	23
4.2 Lubich and Ostermann 1987 .....	25
4.3 Horten and Vandevale 1995 .....	26
4.4 Emmett and Minion 2012 .....	27
4.5 Neumüller 2014 .....	29
5 Direct Solvers in Space-Time .....	30
5.1 Miranker and Liniger 1967 .....	30
5.2 Axelson and Verwer 1985 .....	31
5.3 Womble 1990 .....	33
5.4 Maday and Ronquist 2008 .....	34
5.5 Christlieb, Macdonald and Ong 2010 .....	35
5.6 Güttel 2012 .....	37
6 Conclusions .....	39
References .....	40

# Motivation

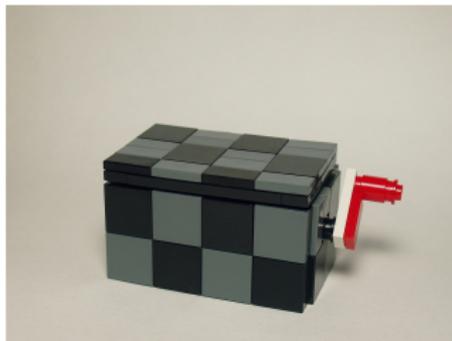
<b>50 Years of Time Parallel Time Integration</b> .....	1
Martin J. Gander	
1 Introduction .....	4
2 Shooting Type Time Parallel Methods .....	5
2.1 Nievergelt 1964 .....	5
2.2 Bellen and Zennaro 1989 .....	7
2.3 Chartier and Philippe 1993 .....	9
2.4 Saha, Stadel and Tremaine 1996 .....	10
2.5 Lions, Maday and Turinici 2001 .....	12
3 Domain Decomposition Methods in Space-Time .....	16
3.1 Picard and Lindelöf 1893/1894 .....	16
3.2 Lelarasme, Ruehli and Sangiovanni-Vincentelli 1982 .....	17
3.3 Gander 1996 .....	20
3.4 Gander, Halpern and Nataf 1999 .....	21
3.5 Recent Developments .....	23
4 Multigrid Methods in Space-Time .....	23
4.1 Hackbusch 1984 .....	23
4.2 Lubich and Ostermann 1987 .....	25
4.3 Horten and Vandevale 1995 .....	26
4.4 Emmett and Minion 2012 .....	27
4.5 Neumüller 2014 .....	29
5 Direct Solvers in Space-Time .....	30
5.1 Miranker and Liniger 1967 .....	30
5.2 Axelson and Verwer 1985 .....	31
5.3 Womble 1990 .....	33
5.4 Maday and Ronquist 2008 .....	34
5.5 Christlieb, Macdonald and Ong 2010 .....	35
5.6 Güttel 2012 .....	37
6 Conclusions .....	39
References .....	40

## Embedding PFASST into multigrid theory



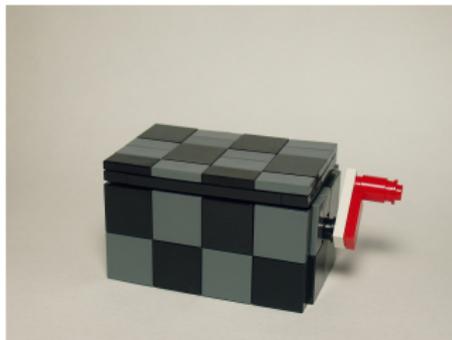
- PFASST looks complicated
- PFASST shows similarities to multigrid
- multigrid is extensively studied

## Embedding PFASST into multigrid theory



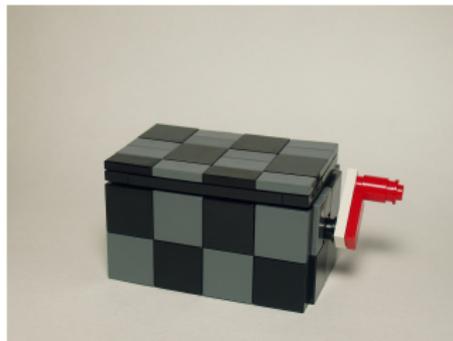
- PFASST looks complicated
- PFASST shows similarities to multigrid
- multigrid is extensively studied

## Embedding PFASST into multigrid theory



- PFASST looks complicated
- PFASST shows similarities to multigrid
- multigrid is extensively studied

## Embedding PFASST into multigrid theory



- PFASST looks complicated
- PFASST shows similarities to multigrid
- multigrid is extensively studied

Now let's show that PFASST **actually is** a multigrid algorithm, under certain assumptions and use this to analyze the parallel performance.

## Collocation formulation on a single time-step

Consider the Picard form of an initial value problem on  $[T_l, T_{l+1}]$

$$u(t) = u_l + \int_{T_l}^t \mathbf{A} \cdot u(s) ds,$$

discretized using spectral quadrature rules with nodes  $\tau_m$ :

$$(\mathbf{I} - \Delta t \mathbf{Q} \otimes \mathbf{A})(\mathbf{u}) = \mathbf{u}_l$$

This corresponds to a fully implicit Runge-Kutta method on  $[T_l, T_{l+1}]$ , which we solve iteratively.



## Collocation formulation on a single time-step

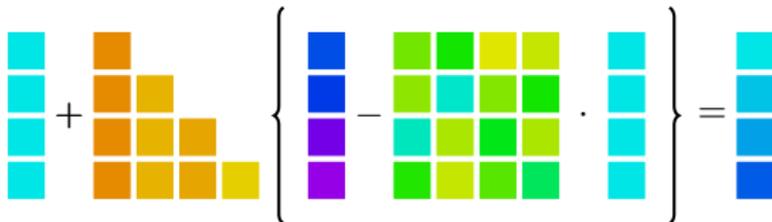
Consider the Picard form of an initial value problem on  $[T_l, T_{l+1}]$

$$u(t) = u_l + \int_{T_l}^t \mathbf{A} \cdot u(s) ds,$$

discretized using spectral quadrature rules with nodes  $\tau_m$ :

$$(\mathbf{I} - \Delta t \mathbf{Q} \otimes \mathbf{A})(\mathbf{u}) = \mathbf{u}_l$$

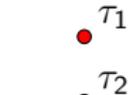
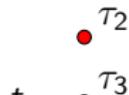
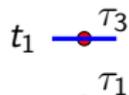
This corresponds to a fully implicit Runge-Kutta method on  $[T_l, T_{l+1}]$ , which we solve iteratively.





## Linked collocation problem

$t_0$  — We now link  $L$  time-steps together, using  $\mathbf{N}$  to transfer information from step  $l$  to step  $l + 1$ . We get:



$$\mathbf{M}_{lcp} \mathbf{U} = \mathbf{U}_0$$

## Linked collocation problem

$t_0$  — We now link  $L$  time-steps together, using  $\mathbf{N}$  to transfer information from step  $l$  to step  $l + 1$ . We get:

$\tau_1$

$\tau_2$

$t_1$  —  $\tau_3$

$\tau_1$

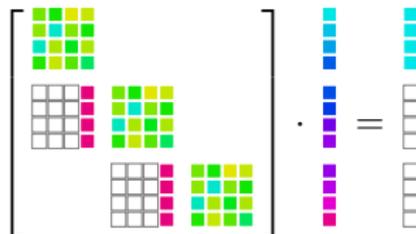
$\tau_2$

$t_2$  —  $\tau_3$

$\tau_1$

$\tau_2$

$T$  —  $\tau_3$



## Linked collocation problem

$t_0$  — We now link  $L$  time-steps together, using  $\mathbf{N}$  to transfer information from step  $l$  to step  $l + 1$ . We get:

$\tau_1$



$\tau_2$



$t_1$  —  $\tau_3$



$\tau_1$



$\tau_2$



$\tau_2$



$t_2$  —  $\tau_3$



$\tau_1$

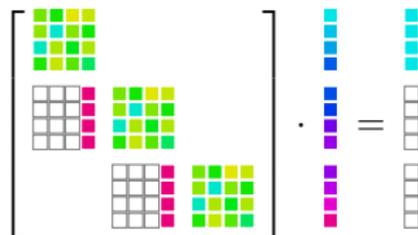


$\tau_2$



$T$  —  $\tau_3$





- use (linear/FAS) multigrid to solve this system iteratively
- exploit cheapest coarse level to quickly propagate information forward in time
- smoother: block Jacobi + block Gauß-Seidel

# Approximative Block-Gauß-Seidel

$t_0$  —

# Approximative Block-Gauß-Seidel

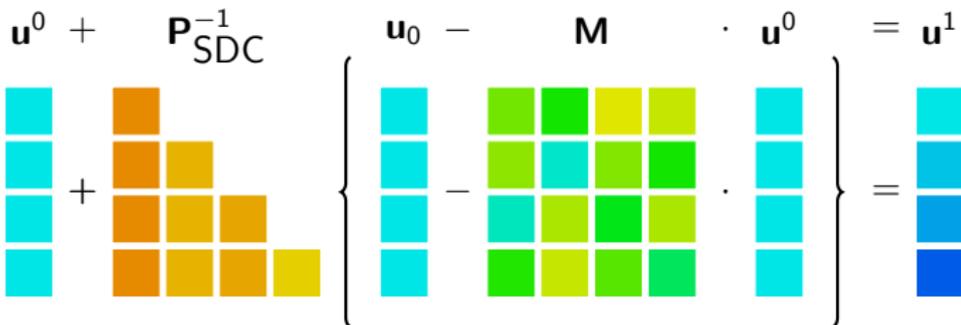
... on the first subinterval

$t_0$  —

•  $\tau_1$

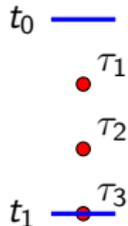
•  $\tau_2$

•  $\tau_3$

$$\mathbf{u}^0 + \mathbf{P}_{\text{SDC}}^{-1} \left\{ \mathbf{u}_0 - \mathbf{M} \cdot \mathbf{u}^0 \right\} = \mathbf{u}^1$$


# Approximative Block-Gauß-Seidel

... passing end value to the next subinterval

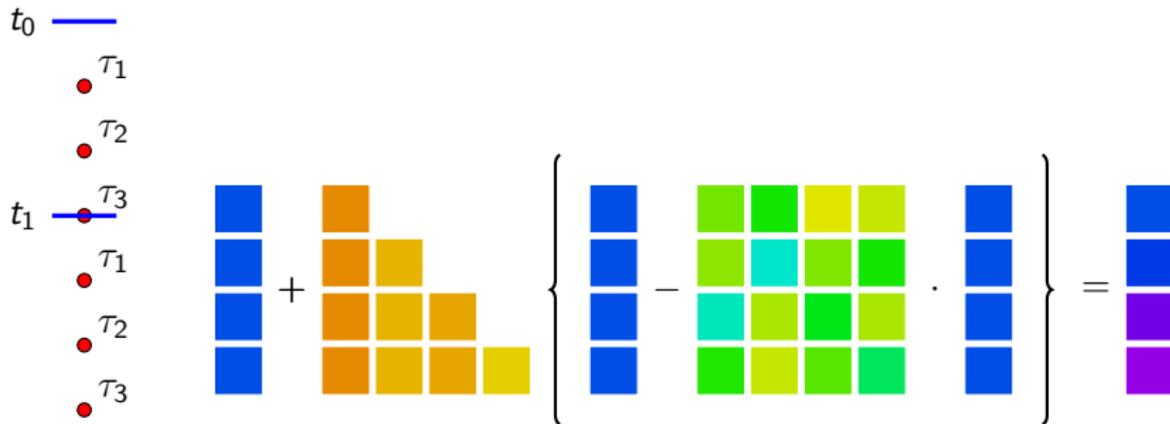


$$\begin{array}{c}
 \mathbf{N} \\
 \begin{array}{|c|c|c|c|}
 \hline
 \square & \square & \square & \color{red}{\square} \\
 \hline
 \square & \square & \square & \color{red}{\square} \\
 \hline
 \square & \square & \square & \color{red}{\square} \\
 \hline
 \square & \square & \square & \color{red}{\square} \\
 \hline
 \end{array}
 \cdot
 \begin{array}{|c|}
 \hline
 \color{cyan}{\square} \\
 \hline
 \color{cyan}{\square} \\
 \hline
 \color{cyan}{\square} \\
 \hline
 \color{blue}{\square} \\
 \hline
 \end{array}
 =
 \begin{array}{|c|}
 \hline
 \color{blue}{\square} \\
 \hline
 \color{blue}{\square} \\
 \hline
 \color{blue}{\square} \\
 \hline
 \color{blue}{\square} \\
 \hline
 \end{array}
 \end{array}$$

The diagram shows a matrix multiplication. The matrix  $\mathbf{N}$  is a 4x4 grid where the first three columns are white squares and the fourth column is red squares. This is multiplied by a column vector  $\mathbf{u}^1$  with four elements: the top three are cyan squares and the bottom one is a blue square. The result is a column vector  $\mathbf{u}_0^2$  with four blue squares.

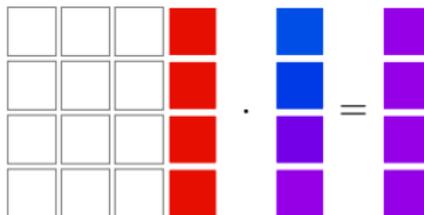
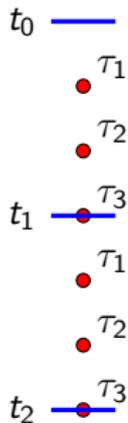
# Approximative Block-Gauß-Seidel

... on the second subinterval



# Approximative Block-Gauß-Seidel

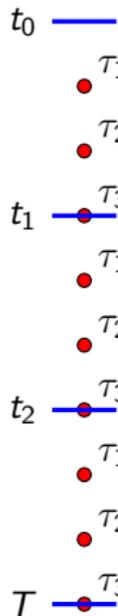
... passing end value to the next subinterval





# Approximative Block-Gauß-Seidel

... all in one



$$\begin{bmatrix} \text{cyan} \\ \text{cyan} \\ \text{cyan} \\ \text{cyan} \end{bmatrix} + \left[ \begin{array}{c|c|c} \begin{matrix} \text{magenta} & & \\ \text{magenta} & \text{magenta} & \\ \text{magenta} & \text{magenta} & \text{magenta} \end{matrix} & & \\ \hline \begin{matrix} \text{red} & & \\ \text{red} & \text{red} & \\ \text{red} & \text{red} & \text{red} \end{matrix} & \begin{matrix} \text{magenta} & & \\ \text{magenta} & \text{magenta} & \\ \text{magenta} & \text{magenta} & \text{magenta} \end{matrix} & \\ \hline & \begin{matrix} \text{red} & & \\ \text{red} & \text{red} & \\ \text{red} & \text{red} & \text{red} \end{matrix} & \begin{matrix} \text{magenta} & & \\ \text{magenta} & \text{magenta} & \\ \text{magenta} & \text{magenta} & \text{magenta} \end{matrix} \end{array} \right]^{-1} \left\{ \begin{array}{c} \begin{matrix} \text{cyan} \\ \text{cyan} \\ \text{cyan} \\ \text{cyan} \end{matrix} - \begin{matrix} \text{green} & \text{yellow} \\ \text{cyan} & \text{green} & \text{yellow} \\ \text{green} & \text{yellow} & \text{green} \end{matrix} \\ \begin{matrix} \text{red} \\ \text{red} \\ \text{red} \end{matrix} - \begin{matrix} \text{red} & \text{magenta} \\ \text{red} & \text{red} & \text{magenta} \\ \text{red} & \text{red} & \text{red} \end{matrix} \\ \begin{matrix} \text{cyan} \\ \text{cyan} \\ \text{cyan} \\ \text{cyan} \end{matrix} \cdot \begin{matrix} \text{green} & \text{yellow} \\ \text{cyan} & \text{green} & \text{yellow} \\ \text{green} & \text{yellow} & \text{green} \end{matrix} \end{array} \right\} = \begin{matrix} \text{cyan} \\ \text{blue} \\ \text{purple} \\ \text{magenta} \end{matrix}$$

# Approximative Block-Gauß-Seidel

$t_0$  —

•  $\tau_1$

•  $\tau_2$

$t_1$  — •  $\tau_3$

•  $\tau_1$

•  $\tau_2$

$t_2$  — •  $\tau_3$

•  $\tau_1$

•  $\tau_2$

$T$  — •  $\tau_3$

$\mathbf{U}^0 +$

$\mathbf{P}_{\text{aGS}}^{-1}$

$\mathbf{U}_0 -$

$\mathbf{M}_{\text{lcp}}$

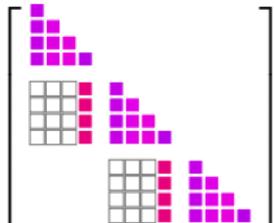
$\cdot \mathbf{U}^0 =$

$\mathbf{U}^1$



+



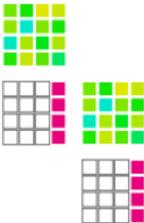
]

<sup>-1</sup>

{



-



-



-



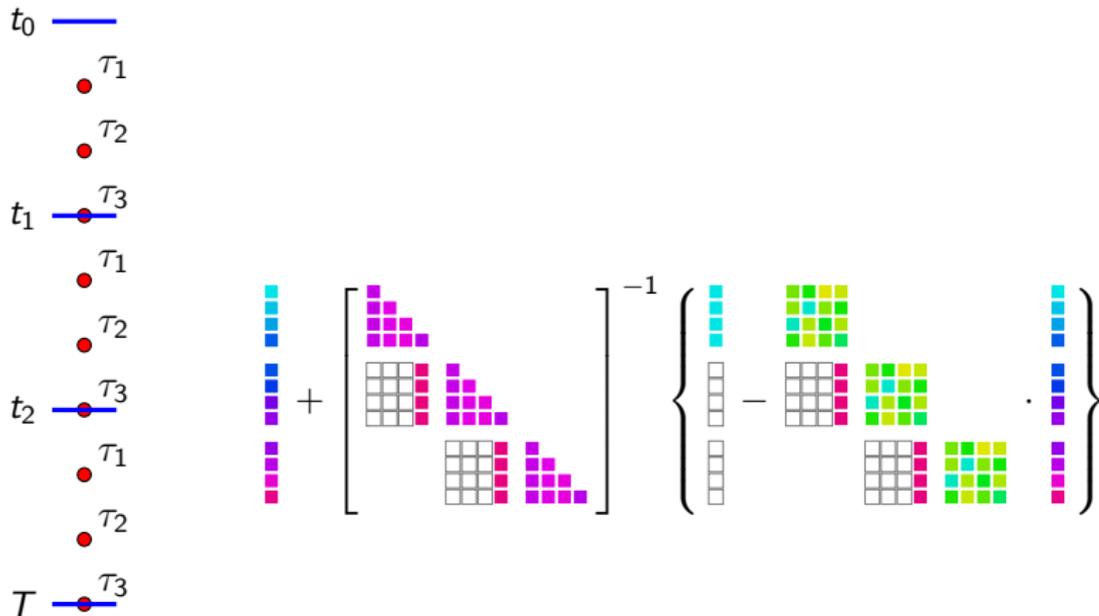
}

=



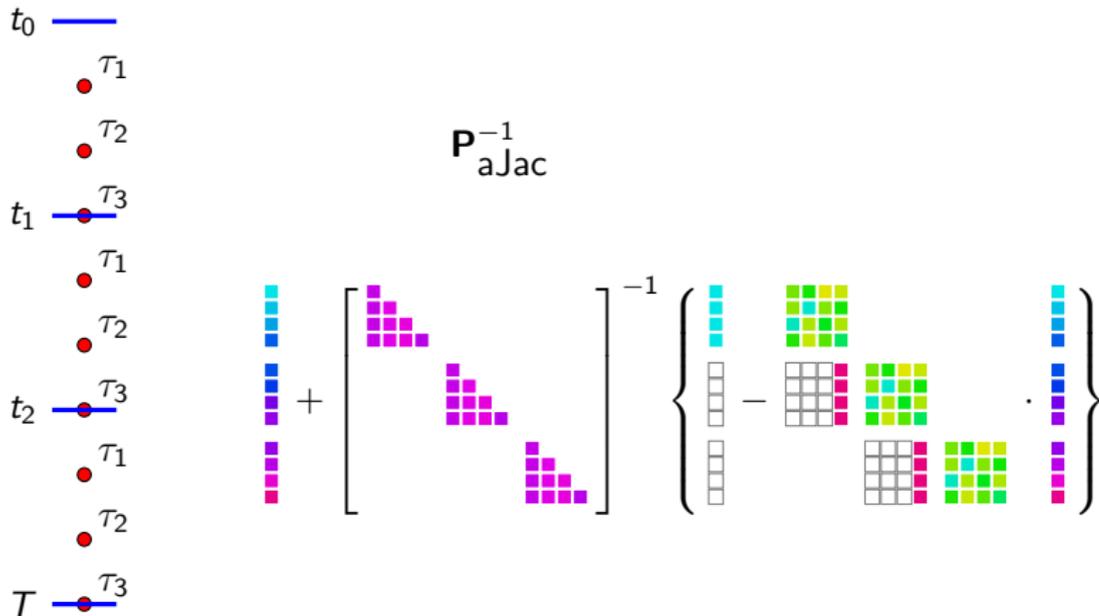
# Approximative Block-Jacobi

... starting from the approximative Gauß-Seidel



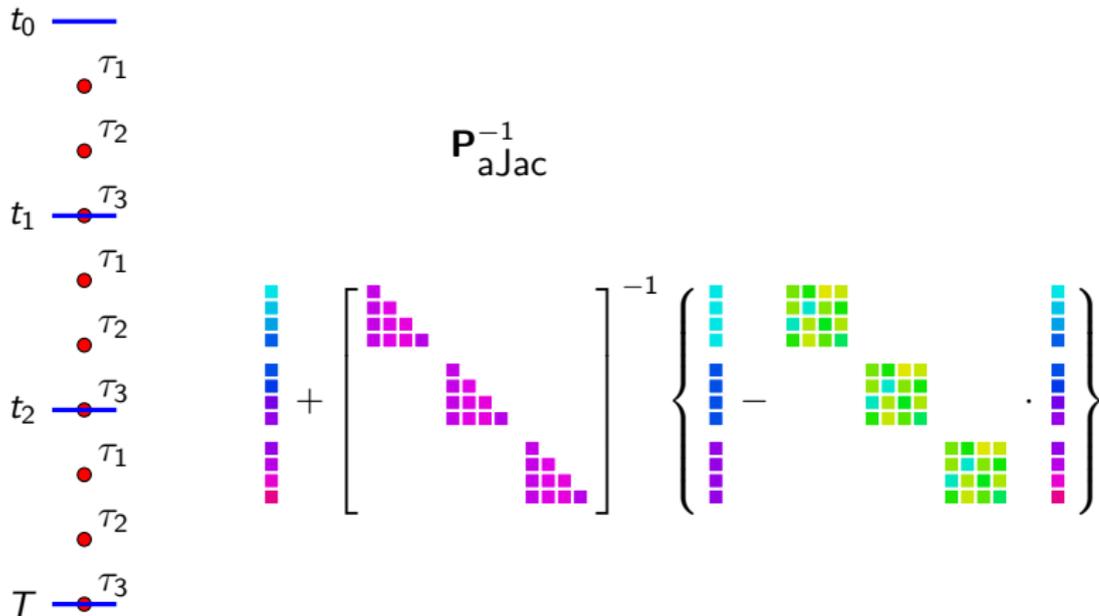
# Approximative Block-Jacobi

... one little adjustment



# Approximative Block-Jacobi

... another little manipulation



# Coarse Grid Correction



## Coarse Grid Correction



Do a **block Jacobi** step

## Coarse Grid Correction



Do a **block Jacobi** step

## Coarse Grid Correction



Do a **block Jacobi** step

$$\text{Compute } \tau_k = \tilde{\mathbf{M}}_{|cp} \mathbf{I}_h^{2h} \mathbf{U}^k - \mathbf{I}_h^{2h} \mathbf{M}_{|cp} \mathbf{U}^k$$

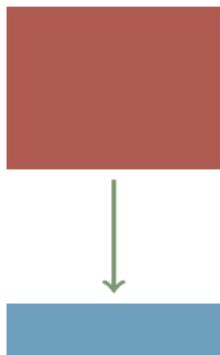
## Coarse Grid Correction



Do a **block Jacobi** step

$$\text{Compute } \tau_k = \tilde{\mathbf{M}}_{\text{lcp}} \mathbf{I}_h^{2h} \mathbf{U}^k - \mathbf{I}_h^{2h} \mathbf{M}_{\text{lcp}} \mathbf{U}^k$$

## Coarse Grid Correction



Do a **block Jacobi step**

$$\text{Compute } \tau_k = \tilde{\mathbf{M}}_{\text{lcp}} \mathbf{I}_h^{2h} \mathbf{U}^k - \mathbf{I}_h^{2h} \mathbf{M}_{\text{lcp}} \mathbf{U}^k$$

Do a **block Gauß-Seidel step** with  $\tilde{\mathbf{U}}_0^k + \tau_k$

## Coarse Grid Correction



Do a **block Jacobi** step

$$\text{Compute } \tau_k = \tilde{\mathbf{M}}_{\text{lcp}} \mathbf{I}_h^{2h} \mathbf{U}^k - \mathbf{I}_h^{2h} \mathbf{M}_{\text{lcp}} \mathbf{U}^k$$

Do a **block Gauß-Seidel** step with  $\tilde{\mathbf{U}}_0^k + \tau_k$

## Coarse Grid Correction



Do a **block Jacobi step**

$$\text{Compute } \tau_k = \tilde{\mathbf{M}}_{\text{lcp}} \mathbf{I}_h^{2h} \mathbf{U}^k - \mathbf{I}_h^{2h} \mathbf{M}_{\text{lcp}} \mathbf{U}^k$$

Do a **block Gauß-Seidel step** with  $\tilde{\mathbf{U}}_0^k + \tau_k$

$$\text{Correct } \mathbf{U}^{k+1} = \mathbf{U}^k + \mathbf{I}_{2h}^h (\tilde{\mathbf{U}}^{k+1/2} - \mathbf{I}_h^{2h} \mathbf{U}^k)$$

## Coarse Grid Correction



Do a **block Jacobi step**

$$\text{Compute } \tau_k = \tilde{\mathbf{M}}_{\text{lcp}} \mathbf{I}_h^{2h} \mathbf{U}^k - \mathbf{I}_h^{2h} \mathbf{M}_{\text{lcp}} \mathbf{U}^k$$

Do a **block Gauß-Seidel step** with  $\tilde{\mathbf{U}}_0^k + \tau_k$

$$\text{Correct } \mathbf{U}^{k+1} = \mathbf{U}^k + \mathbf{I}_{2h}^h (\tilde{\mathbf{U}}^{k+1/2} - \mathbf{I}_h^{2h} \mathbf{U}^k)$$

## Coarse Grid Correction



Do a **block Jacobi step**

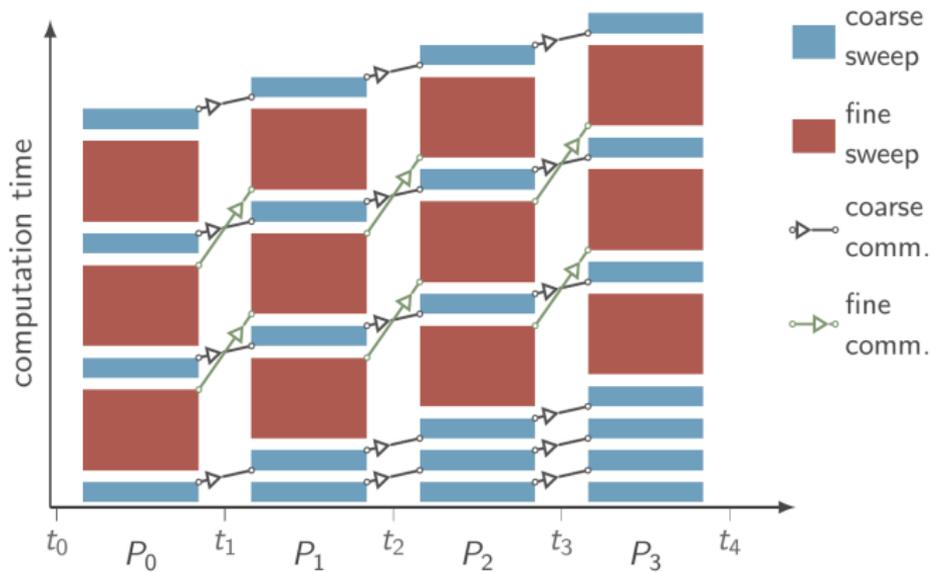
$$\text{Compute } \tau_k = \tilde{\mathbf{M}}_{\text{lcp}} \mathbf{I}_h^{2h} \mathbf{U}^k - \mathbf{I}_h^{2h} \mathbf{M}_{\text{lcp}} \mathbf{U}^k$$

Do a **block Gauß-Seidel step** with  $\tilde{\mathbf{U}}_0^k + \tau_k$

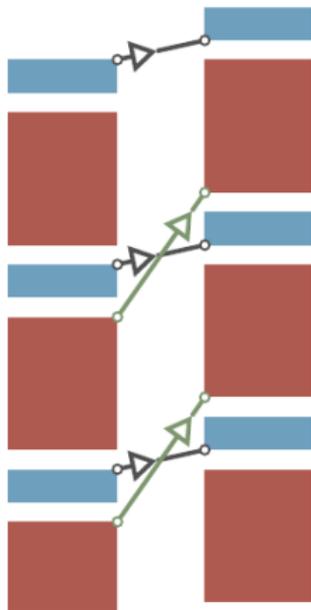
$$\text{Correct } \mathbf{U}^{k+1} = \mathbf{U}^k + \mathbf{I}_{2h}^h (\tilde{\mathbf{U}}^{k+1/2} - \mathbf{I}_h^{2h} \mathbf{U}^k)$$

Do next **block Jacobi step**

# PFASST overview



## Putting the pieces together



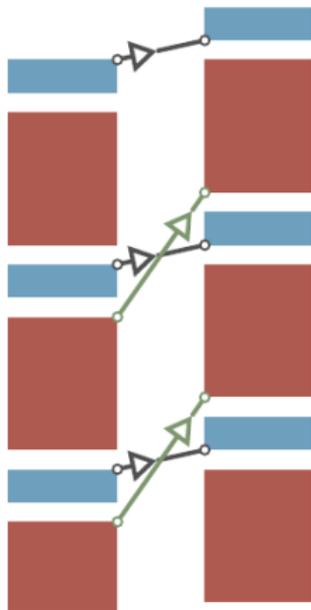
This can easily be written as

$$\mathbf{U}^{k+\frac{1}{2}} = \mathbf{U}^k + \mathbf{I}_{2h}^h \tilde{\mathbf{P}}_{\text{aGS}}^{-1} \mathbf{I}_h^{2h} \left( \mathbf{U}_0 - \mathbf{M}_{\text{lcp}} \mathbf{U}^k \right)$$

$$\mathbf{U}^{k+1} = \mathbf{U}^{k+\frac{1}{2}} + \mathbf{P}_{\text{aJac}}^{-1} \left( \mathbf{U}_0 - \mathbf{M}_{\text{lcp}} \mathbf{U}^{k+\frac{1}{2}} \right),$$

which is a two-level multigrid scheme, with an approximative **Block-Gauß-Seidel** on the coarse level and an approximative **Block-Jacobi** on the fine level.

## Putting the pieces together



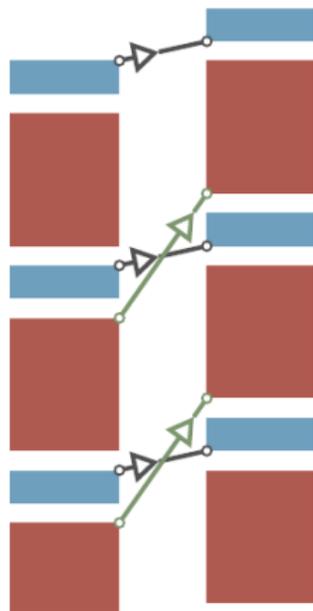
This can easily be written as

$$\mathbf{U}^{k+\frac{1}{2}} = \mathbf{U}^k + \mathbf{I}_{2h}^h \tilde{\mathbf{P}}_{\text{aGS}}^{-1} \mathbf{I}_h^{2h} \left( \mathbf{U}_0 - \mathbf{M}_{\text{lcp}} \mathbf{U}^k \right)$$

$$\mathbf{U}^{k+1} = \mathbf{U}^{k+\frac{1}{2}} + \mathbf{P}_{\text{aJac}}^{-1} \left( \mathbf{U}_0 - \mathbf{M}_{\text{lcp}} \mathbf{U}^{k+\frac{1}{2}} \right),$$

which is a two-level multigrid scheme, with an approximative **Block-Gauß-Seidel** on the coarse level and an approximative **Block-Jacobi** on the fine level.

## Putting the pieces together



This can easily be written as

$$\mathbf{U}^{k+\frac{1}{2}} = \mathbf{U}^k + \mathbf{I}_{2h}^h \tilde{\mathbf{P}}_{\text{aGS}}^{-1} \mathbf{I}_h^{2h} \left( \mathbf{U}_0 - \mathbf{M}_{\text{lcp}} \mathbf{U}^k \right)$$

$$\mathbf{U}^{k+1} = \mathbf{U}^{k+\frac{1}{2}} + \mathbf{P}_{\text{aJac}}^{-1} \left( \mathbf{U}_0 - \mathbf{M}_{\text{lcp}} \mathbf{U}^{k+\frac{1}{2}} \right),$$

which is a two-level multigrid scheme, with an approximative **Block-Gauß-Seidel** on the coarse level and an approximative **Block-Jacobi** on the fine level.

## Analysis of PFASST - a modest try

The center of attention is the **iteration matrix of PFASST**

$$\mathbf{T}_{\text{PFASST}} = \mathbf{I} - \left( \mathbf{I}_{2h}^h \tilde{\mathbf{P}}_{\text{aGS}}^{-1} \mathbf{I}_h^{2h} + \mathbf{P}_{\text{aJac}}^{-1} - \mathbf{P}_{\text{aJac}}^{-1} \mathbf{M}_{\text{lcp}} \mathbf{I}_{2h}^h \tilde{\mathbf{P}}_{\text{aJac}}^{-1} \mathbf{I}_h^{2h} \right) \mathbf{M}_{\text{lcp}}$$

## Analysis of PFASST - a modest try

The center of attention is the **iteration matrix of PFASST**

$$\begin{aligned}
 \mathbf{T}_{\text{PFASST}} &= \mathbf{I} - \left( \mathbf{I}_{2h}^h \tilde{\mathbf{P}}_{\text{aGS}}^{-1} \mathbf{I}_h^{2h} + \mathbf{P}_{\text{aJac}}^{-1} - \mathbf{P}_{\text{aJac}}^{-1} \mathbf{M}_{\text{lcp}} \mathbf{I}_{2h}^h \tilde{\mathbf{P}}_{\text{aJac}}^{-1} \mathbf{I}_h^{2h} \right) \mathbf{M}_{\text{lcp}} \\
 &= \underbrace{\left( \mathbf{I} - \mathbf{P}_{\text{aJac}}^{-1} \mathbf{M}_{\text{lcp}} \right)}_{\text{Post-Smoothen}} \underbrace{\left( \mathbf{I} - \mathbf{I}_{2h}^h \tilde{\mathbf{P}}_{\text{aGS}}^{-1} \mathbf{I}_h^{2h} \mathbf{M}_{\text{lcp}} \right)}_{\approx \text{CG-Correction}} \underbrace{\mathbf{I}}_{\text{Pre-Smoothen}},
 \end{aligned}$$

## Analysis of PFASST - a modest try

The center of attention is the **iteration matrix of PFASST**

$$\begin{aligned}
 \mathbf{T}_{\text{PFASST}} &= \mathbf{I} - \left( \mathbf{I}_{2h}^h \tilde{\mathbf{P}}_{\text{aGS}}^{-1} \mathbf{I}_h^{2h} + \mathbf{P}_{\text{aJac}}^{-1} - \mathbf{P}_{\text{aJac}}^{-1} \mathbf{M}_{\text{lcp}} \mathbf{I}_{2h}^h \tilde{\mathbf{P}}_{\text{aJac}}^{-1} \mathbf{I}_h^{2h} \right) \mathbf{M}_{\text{lcp}} \\
 &= \underbrace{\left( \mathbf{I} - \mathbf{P}_{\text{aJac}}^{-1} \mathbf{M}_{\text{lcp}} \right)}_{\text{Post-Smoother}} \underbrace{\left( \mathbf{I} - \mathbf{I}_{2h}^h \tilde{\mathbf{P}}_{\text{aGS}}^{-1} \mathbf{I}_h^{2h} \mathbf{M}_{\text{lcp}} \right)}_{\approx \text{CG-Correction}} \underbrace{\mathbf{I}}_{\text{Pre-Smoother}},
 \end{aligned}$$

which is decomposable into 3 layers.

## Analysis of PFASST - a modest try

The center of attention is the **iteration matrix of PFASST**

$$\begin{aligned}
 \mathbf{T}_{\text{PFASST}} &= \mathbf{I} - \left( \mathbf{I}_{2h}^h \tilde{\mathbf{P}}_{\text{aGS}}^{-1} \mathbf{I}_h^{2h} + \mathbf{P}_{\text{aJac}}^{-1} - \mathbf{P}_{\text{aJac}}^{-1} \mathbf{M}_{\text{lcp}} \mathbf{I}_{2h}^h \tilde{\mathbf{P}}_{\text{aJac}}^{-1} \mathbf{I}_h^{2h} \right) \mathbf{M}_{\text{lcp}} \\
 &= \underbrace{\left( \mathbf{I} - \mathbf{P}_{\text{aJac}}^{-1} \mathbf{M}_{\text{lcp}} \right)}_{\text{Post-Smoothen}} \underbrace{\left( \mathbf{I} - \mathbf{I}_{2h}^h \tilde{\mathbf{P}}_{\text{aGS}}^{-1} \mathbf{I}_h^{2h} \mathbf{M}_{\text{lcp}} \right)}_{\approx \text{CG-Correction}} \underbrace{\mathbf{I}}_{\text{Pre-Smoothen}},
 \end{aligned}$$

which is decomposable into 3 layers.

dof e.g.      over 9000      10      5

$$\mathbf{T}_{\text{PFASST}} \simeq \mathbf{T}_{\text{space}} \otimes \mathbf{T}_{\text{time}} \otimes \mathbf{T}_{\text{colloc}}$$

## Analysis of PFASST - a modest try

The center of attention is the **iteration matrix of PFASST**

$$\begin{aligned}
 \mathbf{T}_{\text{PFASST}} &= \mathbf{I} - \left( \mathbf{I}_{2h}^h \tilde{\mathbf{P}}_{\text{aGS}}^{-1} \mathbf{I}_h^{2h} + \mathbf{P}_{\text{aJac}}^{-1} - \mathbf{P}_{\text{aJac}}^{-1} \mathbf{M}_{\text{lcp}} \mathbf{I}_{2h}^h \tilde{\mathbf{P}}_{\text{aJac}}^{-1} \mathbf{I}_h^{2h} \right) \mathbf{M}_{\text{lcp}} \\
 &= \underbrace{\left( \mathbf{I} - \mathbf{P}_{\text{aJac}}^{-1} \mathbf{M}_{\text{lcp}} \right)}_{\text{Post-Smoothen}} \underbrace{\left( \mathbf{I} - \mathbf{I}_{2h}^h \tilde{\mathbf{P}}_{\text{aGS}}^{-1} \mathbf{I}_h^{2h} \mathbf{M}_{\text{lcp}} \right)}_{\approx \text{CG-Correction}} \underbrace{\mathbf{I}}_{\text{Pre-Smoothen}},
 \end{aligned}$$

which is decomposable into 3 layers.

dof e.g.      over 9000      10      5

$$\mathbf{T}_{\text{PFASST}} \simeq \mathbf{T}_{\text{space}} \otimes \mathbf{T}_{\text{time}} \otimes \mathbf{T}_{\text{colloc}}$$



# Local Fourier Analysis from a matrix point of view

## just transformation

$$\mathcal{F}^{-1} \mathbf{T}_{\text{PFASST}} \mathcal{F} \simeq \psi^{-1} \mathbf{T}_{\text{space}} \psi \otimes \mathbf{T}_{\text{time}} \otimes \mathbf{T}_{\text{colloc}}$$


Now we have e.g. 4500 "time collocation" blocks  $\mathcal{B}_k$  of size  $2 \cdot 10 \cdot 5$  instead of **one** matrix of size  $4.5 \cdot 10^5$ .

# Local Fourier Analysis from a matrix point of view

## just transformation

$$\mathcal{F}^{-1} \mathbf{T}_{\text{PFASST}} \mathcal{F} \simeq \psi^{-1} \mathbf{T}_{\text{space}} \psi \otimes \mathbf{T}_{\text{time}} \otimes \mathbf{T}_{\text{colloc}}$$



Now we have e.g. 4500 “time collocation” blocks  $\mathcal{B}_k$  of size  $2 \cdot 10 \cdot 5$  instead of **one** matrix of size  $4.5 \cdot 10^5$ .

# Local Fourier Analysis from a matrix point of view

## just transformation

$$\mathcal{F}^{-1} \mathbf{T}_{\text{PFASST}} \mathcal{F} \simeq \psi^{-1} \mathbf{T}_{\text{space}} \psi \otimes \mathbf{T}_{\text{time}} \otimes \mathbf{T}_{\text{colloc}}$$

$$= \begin{bmatrix} \begin{matrix} \color{magenta} \blacksquare & \color{magenta} \blacksquare & \color{magenta} \blacksquare & \color{magenta} \blacksquare \\ \color{magenta} \blacksquare & \color{magenta} \blacksquare & \color{magenta} \blacksquare & \color{magenta} \blacksquare \\ \color{red} \blacksquare & \color{magenta} \blacksquare & \color{magenta} \blacksquare & \color{magenta} \blacksquare \end{matrix} & & & \\ & \begin{matrix} \color{purple} \blacksquare & \color{purple} \blacksquare & \color{purple} \blacksquare & \color{purple} \blacksquare \\ \color{purple} \blacksquare & \color{purple} \blacksquare & \color{purple} \blacksquare & \color{purple} \blacksquare \end{matrix} & & \\ & & \dots & \\ & & & \begin{matrix} \color{red} \blacksquare & \color{red} \blacksquare & \color{red} \blacksquare & \color{red} \blacksquare \\ \color{red} \blacksquare & \color{red} \blacksquare & \color{red} \blacksquare & \color{red} \blacksquare \\ \color{magenta} \blacksquare & \color{magenta} \blacksquare & \color{magenta} \blacksquare & \color{magenta} \blacksquare \end{matrix} \end{bmatrix}$$

Now we have e.g. 4500 “time collocation” blocks  $\mathcal{B}_k$  of size  $2 \cdot 10 \cdot 5$  instead of **one** matrix of size  $4.5 \cdot 10^5$ .

## Local Fourier Analysis from a matrix point of view

### just transformation

$$\mathcal{F}^{-1} \mathbf{T}_{\text{PFASST}} \mathcal{F} \simeq \psi^{-1} \mathbf{T}_{\text{space}} \psi \otimes \mathbf{T}_{\text{time}} \otimes \mathbf{T}_{\text{colloc}}$$

$$= \begin{bmatrix} \begin{matrix} \color{magenta} \square & \color{magenta} \square & \color{magenta} \square \\ \color{magenta} \square & \color{magenta} \square & \color{magenta} \square \\ \color{red} \square & \color{magenta} \square & \color{magenta} \square \end{matrix} & & & & \\ & \begin{matrix} \color{purple} \square & \color{purple} \square & \color{purple} \square \\ \color{purple} \square & \color{purple} \square & \color{purple} \square \end{matrix} & & & & \\ & & \dots & & & & \\ & & & & & \begin{matrix} \color{red} \square & \color{red} \square & \color{red} \square \\ \color{red} \square & \color{red} \square & \color{red} \square \\ \color{magenta} \square & \color{red} \square & \color{red} \square \end{matrix} & & \end{bmatrix}$$

Now we have e.g. 4500 “time collocation” blocks  $\mathcal{B}_k$  of size  $2 \cdot 10 \cdot 5$  instead of **one** matrix of size  $4.5 \cdot 10^5$ .

## The convenience of blocks

spectral radii

$$\rho(\mathbf{T}) = \max_l \rho(B_l)$$

norms

$$\|\mathbf{T}\|_2 = \max_l \|B_l\|_2$$

power

$$\mathbf{T}^k = \mathcal{F} \operatorname{diag}(B_1^k, B_2^k, \dots, B_N^k) \mathcal{F}^{-1}$$

## The convenience of blocks

spectral radii

$$\rho(\mathbf{T}) = \max_l \rho(B_l)$$

norms

$$\|\mathbf{T}\|_2 = \max_l \|B_l\|_2$$

power

$$\mathbf{T}^k = \mathcal{F} \operatorname{diag}(B_1^k, B_2^k, \dots, B_N^k) \mathcal{F}^{-1}$$

## The convenience of blocks

spectral radii

$$\rho(\mathbf{T}) = \max_l \rho(\mathcal{B}_l)$$

norms

$$\|\mathbf{T}\|_2 = \max_l \|\mathcal{B}_l\|_2$$

power

$$\mathbf{T}^k = \mathcal{F} \operatorname{diag}(\mathcal{B}_1^k, \mathcal{B}_2^k, \dots, \mathcal{B}_N^k) \mathcal{F}^{-1}$$

## A model problem

Use second order difference method to discretize the heat equation

$$\mathbf{u}_t(t) = \mathbf{A}\mathbf{u}(t)$$

$$\mathbf{A} = \frac{\mu}{(\Delta x)^2} \begin{pmatrix} 2 & -1 & 0 & \cdots & -1 \\ -1 & 2 & -1 & & \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & & & -1 & 2 & -1 \\ -1 & 0 & \cdots & -1 & 2 \end{pmatrix}$$

$$\nu = \mu \Delta t / (\Delta x)^2$$

Space problem is decomposable into the modes

$$\mathbf{m}_k = \left[ \exp\left(i \cdot \frac{kn}{N}\right) \right]_{n=1, \dots, N}$$

## A model problem

Use second order difference method to discretize the heat equation

$$\mathbf{u}_t(t) = \mathbf{A}\mathbf{u}(t)$$

$$\mathbf{A} = \frac{\mu}{(\Delta x)^2} \begin{pmatrix} 2 & -1 & 0 & \cdots & -1 \\ -1 & 2 & -1 & & \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & & & -1 & 2 & -1 \\ -1 & 0 & \cdots & -1 & 2 \end{pmatrix}$$

$$\nu = \mu \Delta t / (\Delta x)^2$$

Space problem is decomposable into the modes

$$\mathbf{m}_k = \left[ \exp\left(i \cdot \frac{kn}{N}\right) \right]_{n=1, \dots, N}$$

## A model problem

Use second order difference method to discretize the heat equation

$$\mathbf{u}_t(t) = \mathbf{A}\mathbf{u}(t)$$

$$\mathbf{A} = \frac{\mu}{(\Delta x)^2} \begin{pmatrix} 2 & -1 & 0 & \cdots & -1 \\ -1 & 2 & -1 & & \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & & -1 & 2 & -1 \\ -1 & 0 & \cdots & -1 & 2 \end{pmatrix}$$

$$\nu = \mu \Delta t / (\Delta x)^2$$

Space problem is decomposable into the modes

$$\mathbf{m}_k = \left[ \exp \left( i \cdot \frac{kn}{N} \right) \right]_{n=1, \dots, N}$$

## A model problem

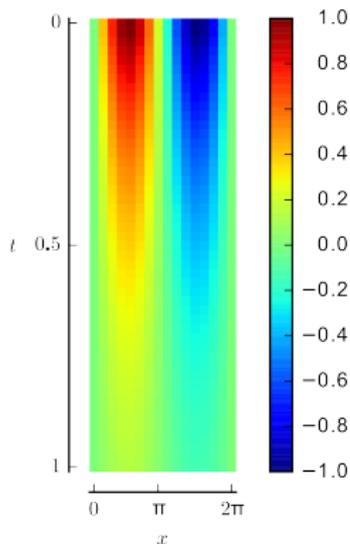


Figure: Numerical solution for the initial value  $u_0 = \sin(x)$ .

Use second order difference method to discretize the heat equation

$$\mathbf{u}_t(t) = \mathbf{A}\mathbf{u}(t)$$

$$\mathbf{A} = \frac{\mu}{(\Delta x)^2} \begin{pmatrix} 2 & -1 & 0 & \cdots & -1 \\ -1 & 2 & -1 & & \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & & -1 & 2 & -1 \\ -1 & 0 & \cdots & -1 & 2 \end{pmatrix}$$

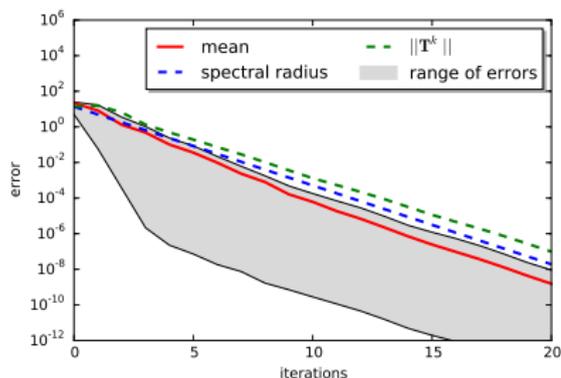
$$\nu = \mu \Delta t / (\Delta x)^2$$

Space problem is decomposable into the modes

$$\mathbf{m}_k = \left[ \exp\left(i \cdot \frac{kn}{N}\right) \right]_{n=1, \dots, N}$$

# First convergence tests

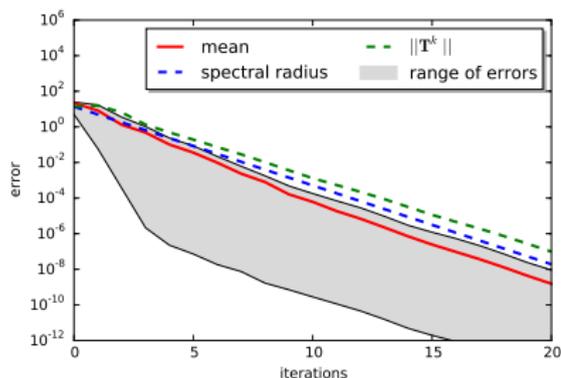
8 time steps



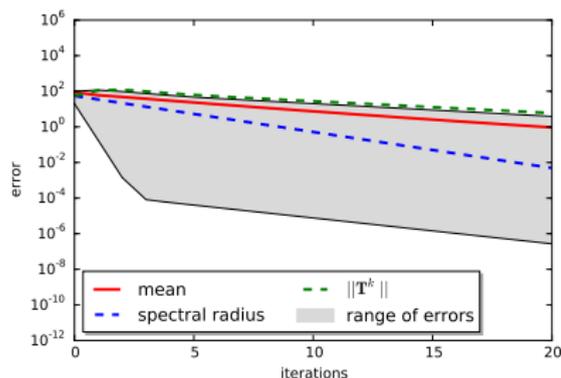
32 spatial nodes, 5 quadrature nodes and  $\mu = 0.01$ .

# First convergence tests

8 time steps



128 time steps



32 spatial nodes, 5 quadrature nodes and  $\mu = 0.01$ .

## Estimating iterations

### Use the spectral radius

- Works great with a few time steps.
- Is awfully wrong for many time steps
- Only a worst case estimation

## Estimating iterations

### Use the spectral radius

- Works great with a few time steps.
- Is awfully wrong for many time steps
- Only a worst case estimation

## Estimating iterations

### Use the spectral radius

- Works great with a few time steps.
- Is awfully wrong for many time steps
- Only a worst case estimation

## Estimating iterations

### Use the spectral radius

- Works great with a few time steps.
- Is awfully wrong for many time steps
- Only a worst case estimation

**Not ideal, so what about**  $\|\mathbf{T}\|_2 = \max_l \|\mathcal{B}_l\|_2$ ?

- Matrix matrix multiplication for each iteration.
- Like the spectral radius, only a worst case estimation.

## Estimating iterations

### Use the spectral radius

- Works great with a few time steps.
- Is awfully wrong for many time steps
- Only a worst case estimation

**Not ideal, so what about**  $\|\mathbf{T}\|_2 = \max_l \|\mathcal{B}_l\|_2$ ?

- Matrix matrix multiplication for each iteration.
- Like the spectral radius, only a worst case estimation.

## Estimating iterations

### Use the spectral radius

- Works great with a few time steps.
- Is awfully wrong for many time steps
- Only a worst case estimation

**Not ideal, so what about**  $\|\mathbf{T}\|_2 = \max_l \|\mathcal{B}_l\|_2$ ?

- Matrix matrix multiplication for each iteration.
- Like the spectral radius, only a worst case estimation.

⇒ **back to the roots, back to counting!**

## Block structure and space modes

... how to count

- 1** Decompose spatial problem into modes  $\mathbf{m}_j$
- 2** Spread  $j$ -th mode across all collocation points and time steps to get initial error mode:

$$\mathbf{e}_j^0 = \mathbf{m}_j \otimes \mathbf{1}_L \otimes \mathbf{1}_M$$

- 3** Use block Fourier transformation to track  $j$ -th error mode over iterations:

$$\|\mathcal{F}\mathbf{e}_j^k\| = \|\mathcal{F}\mathbf{T}^k\mathbf{e}_j^0\| = \|\text{diag}(\mathcal{B}_i^k)\mathcal{F}\mathbf{e}_j^0\| = \|\mathcal{B}_j^k\mathbf{1}_{LM}\|.$$

- 4** Estimate number of iterations  $K_{\text{PFASST}}$  to achieve a certain error reduction for this mode

## Block structure and space modes

... how to count

- 1 Decompose spatial problem into modes  $\mathbf{m}_j$
- 2 Spread  $j$ -th mode across all collocation points and time steps to get initial error mode:

$$\mathbf{e}_j^0 = \mathbf{m}_j \otimes \mathbf{1}_L \otimes \mathbf{1}_M$$

- 3 Use block Fourier transformation to track  $j$ -th error mode over iterations:

$$\|\mathcal{F}\mathbf{e}_j^k\| = \|\mathcal{F}\mathbf{T}^k\mathbf{e}_j^0\| = \|\text{diag}(\mathcal{B}_j^k)\mathcal{F}\mathbf{e}_j^0\| = \|\mathcal{B}_j^k\mathbf{1}_{LM}\|.$$

- 4 Estimate number of iterations  $K_{\text{PFASST}}$  to achieve a certain error reduction for this mode

## Block structure and space modes

... how to count

- 1 Decompose spatial problem into modes  $\mathbf{m}_j$
- 2 Spread  $j$ -th mode across all collocation points and time steps to get initial error mode:

$$\mathbf{e}_j^0 = \mathbf{m}_j \otimes \mathbf{1}_L \otimes \mathbf{1}_M$$

- 3 Use block Fourier transformation to track  $j$ -th error mode over iterations:

$$\|\mathcal{F}\mathbf{e}_j^k\| = \|\mathcal{F}\mathbf{T}^k\mathbf{e}_j^0\| = \|\text{diag}(\mathcal{B}_i^k)\mathcal{F}\mathbf{e}_j^0\| = \|\mathcal{B}_j^k\mathbf{1}_{LM}\|.$$

- 4 Estimate number of iterations  $K_{\text{PFASST}}$  to achieve a certain error reduction for this mode

## Block structure and space modes

...how to count

- 1 Decompose spatial problem into modes  $\mathbf{m}_j$
- 2 Spread  $j$ -th mode across all collocation points and time steps to get initial error mode:

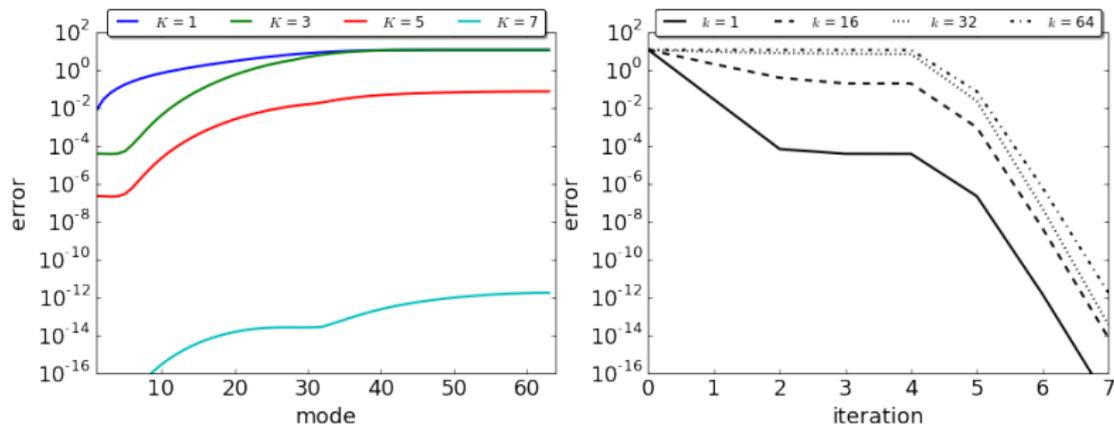
$$\mathbf{e}_j^0 = \mathbf{m}_j \otimes \mathbf{1}_L \otimes \mathbf{1}_M$$

- 3 Use block Fourier transformation to track  $j$ -th error mode over iterations:

$$\|\mathcal{F}\mathbf{e}_j^k\| = \|\mathcal{F}\mathbf{T}^k\mathbf{e}_j^0\| = \|\text{diag}(\mathcal{B}_i^k)\mathcal{F}\mathbf{e}_j^0\| = \|\mathcal{B}_j^k\mathbf{1}_{LM}\|.$$

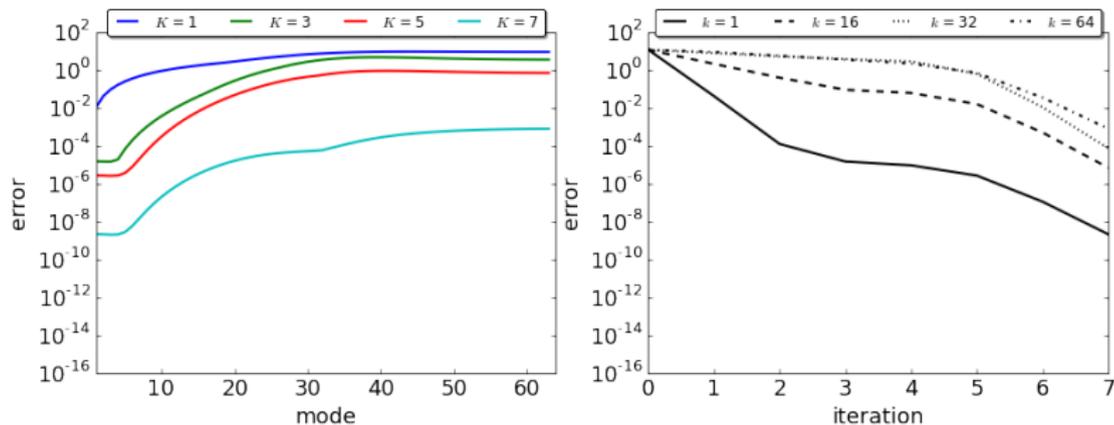
- 4 Estimate number of iterations  $K_{\text{PFASST}}$  to achieve a certain error reduction for this mode

## Convergence of PFASST for another setup



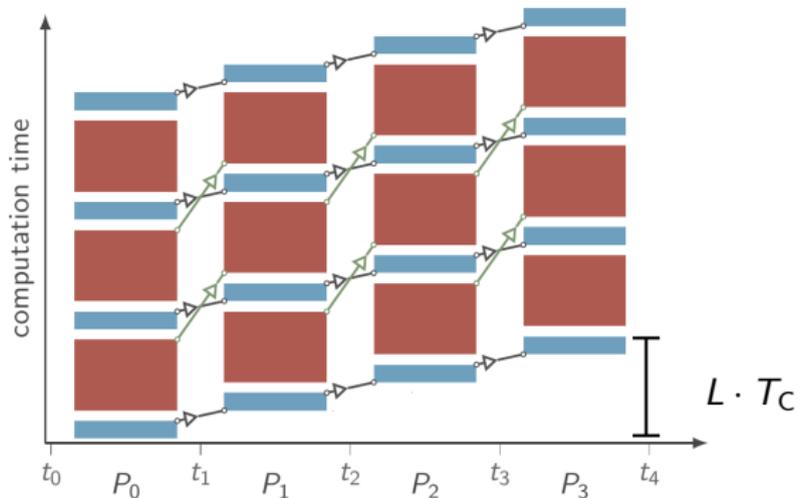
128 spatial nodes, 5 quadrature nodes, 10 time steps and  $\nu = 0.01$

## Convergence of PFASST for another setup

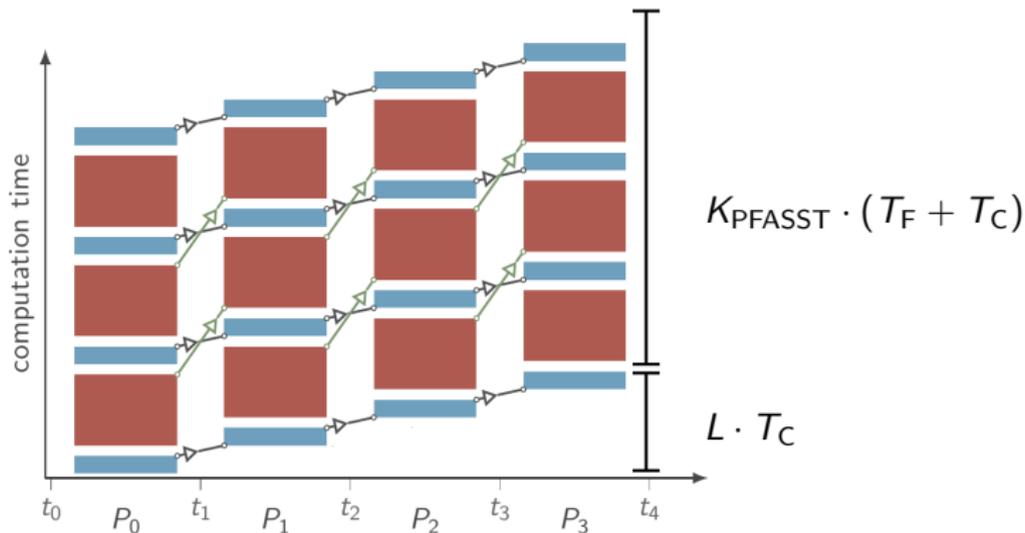


128 spatial nodes, 5 quadrature nodes, 10 time steps and  $\nu = 1.0$

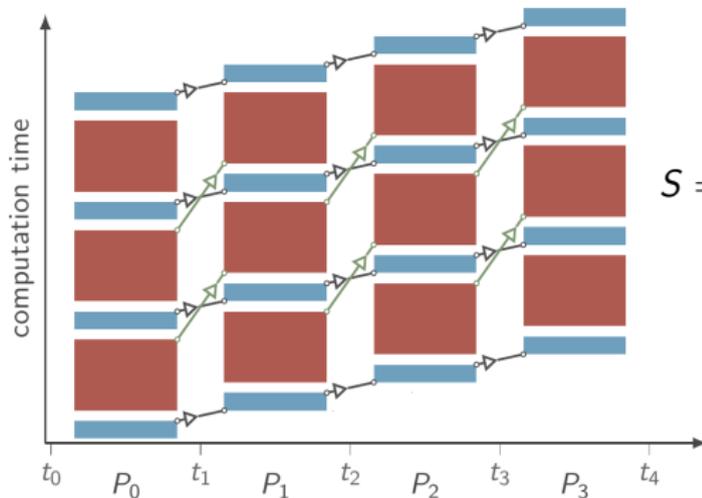
# How to estimate the speedup



# How to estimate the speedup

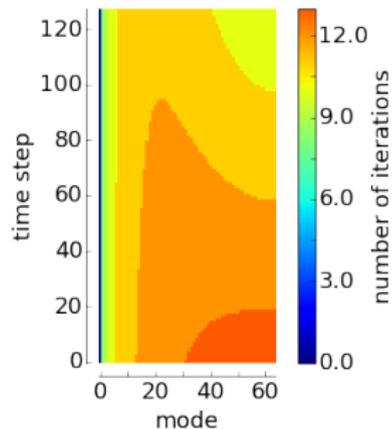
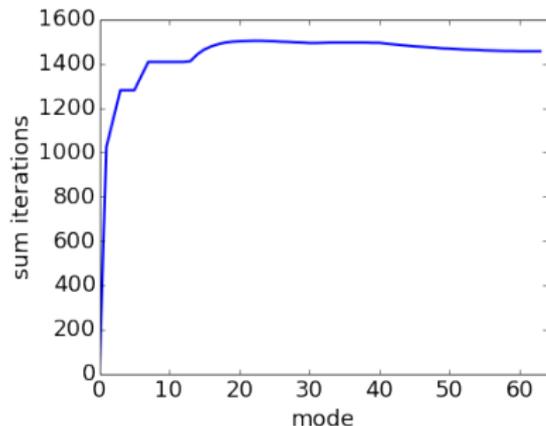


# How to estimate the speedup



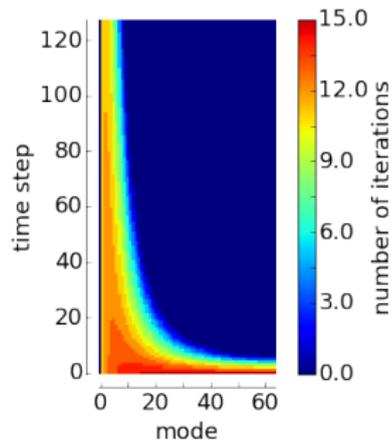
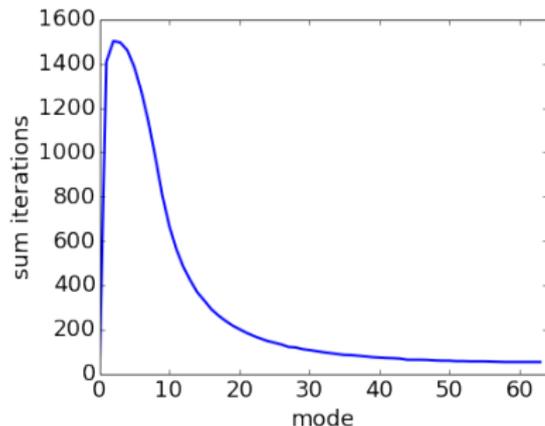
$$S = \frac{\sum_{l=1}^L K_{\text{SDC},l} \cdot T_F}{L \cdot T_C + K_{\text{PFASST}} \cdot (T_F + T_C)}$$

## How SDC performs



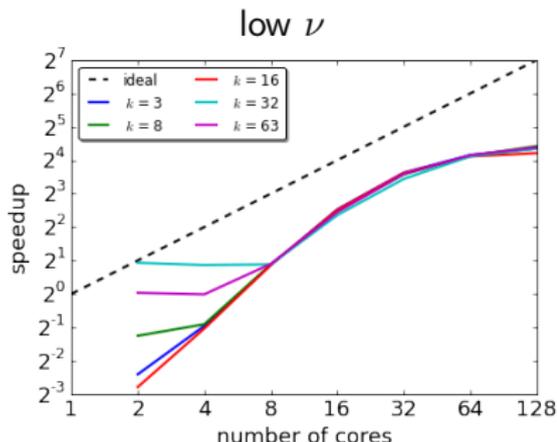
128 spatial nodes, 5 quadrature nodes, 128 time steps and  $\nu = 0.01$ .

## How SDC performs



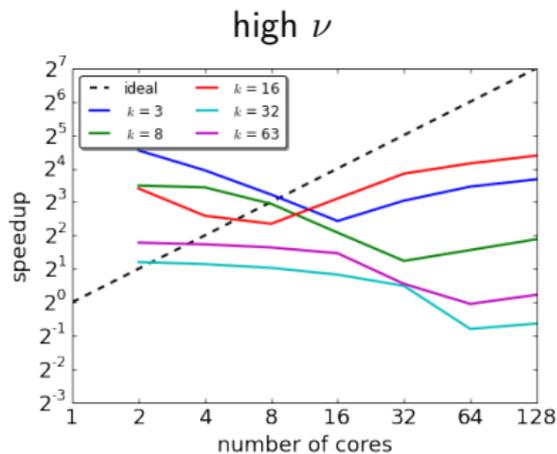
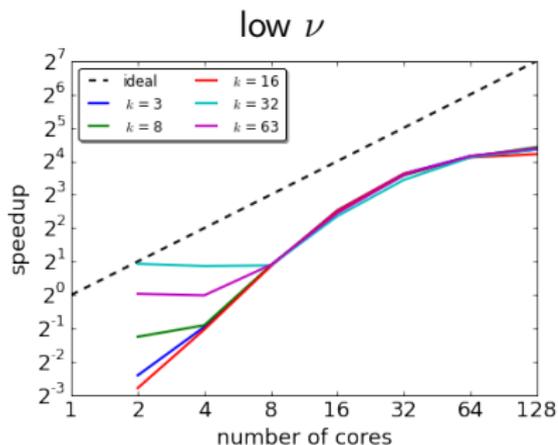
128 spatial nodes, 5 quadrature nodes, 128 time steps and  $\nu = 1.0$ .

# Estimated speedup



128 spatial nodes, 5 quadrature nodes, 128 time steps and  $\nu = 0.01$ .

# Estimated speedup



128 spatial nodes, 5 quadrature nodes, 128 time steps and  $\nu = 1.0$ .

## What's next?

### Achievements until now

- A multigrid view on PFASST
- Iteration matrix in a nice form
- Plug&Play framework
- First insights in the parallel performance

## What's next?

DONE!

### Achievements until now

- A multigrid view on PFASST
- Iteration matrix in a nice form
- Plug&Play framework
- First insights in the parallel performance

## What's next?

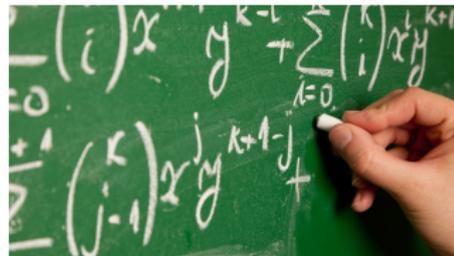
DONE!

### Achievements until now

- A multigrid view on PFASST
- Iteration matrix in a nice form
- Plug&Play framework
- First insights in the parallel performance

### Upcoming challenges

- Local Fourier analysis
- Time coarsening
- Compare to other space time MGs
- Writing the PhD thesis



**Thank you for your attention!**