What is new in Domain Decomposition?

Martin J. Gander
martin.gander@unige.ch

University of Geneva

Rouen, November 8, 2024

Rencontres Normandes sur les aspects théoriques et numériques des EDP

What is not new?

Toselli and Widlund (2005): Domain Decomposition Methods - Algorithms and Theory

Domain decomposition generally refers to the splitting of a partial differential equation, or an approximation thereof, into coupled problems on smaller subdomains forming a partition of the original domain. This decomposition may enter at the continuous level, where different physical models may be used in different regions, or at the discretization level, where it may be convenient to employ different approximation methods in different regions, or in the solution of the algebraic systems arising from the approximation of the partial differential equation. These three aspects are very often interconnected in practice.

This monograph is entirely devoted to the third aspect of domain decomposition.

Definition 1.2 (Optimality). *An iterative method for the solution of a linear system is said to be optimal, if its rate of convergence to the exact solution is independent of the size of the system.*

Classical Results in Domain Decomposition

Two Level Schwarz Methods:

Theorem 3.13 *In case exact solvers are employed on all subspaces, the condition number of the additive Schwarz operator satisfies*

$$\kappa(P_{ad}) \leq C\left(1+\frac{H}{\delta}\right),$$

where C depends on N^c, but is otherwise independent of h, H, and δ.

Balancing Neumann-Neumann Methods:

Theorem 6.4 *The hybrid Schwarz method defined by the operator (6.10) and the spaces and bilinear forms of this section satisfies*

$$s(u,u) \leq s(P_{hy1}u,u) \;\leq\; C(1+\log(H/h))^2 s(u,u),$$

where C is independent not only of the mesh size and the number of substructures, but also of the values ρ_i of the coefficient of (4.3).

Two Level FETI-DP Methods:

Theorem 6.35 (Algorithm B) *The preconditioner M_B satisfies*

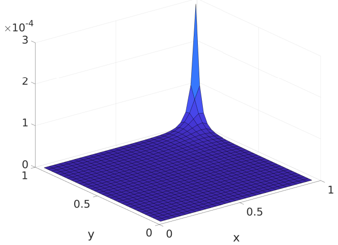
$$\langle M_B\lambda,\lambda\rangle \leq \langle F_B\lambda,\lambda\rangle \leq C(1+\log(H/h))^2\langle M_B\lambda,\lambda\rangle, \quad \lambda \in V. \qquad (6.77)$$

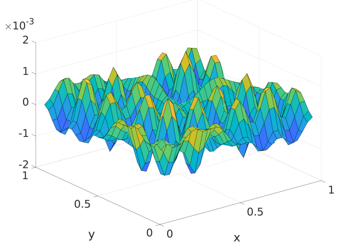
Here C is independent of h, H, γ, and the values of the ρ_i.

New 1: What about Helmholtz Problems?

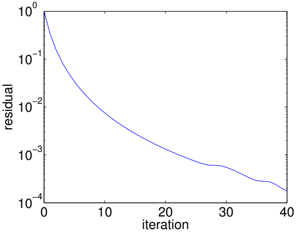
All Classical DD Results are for Laplace-type Problems!

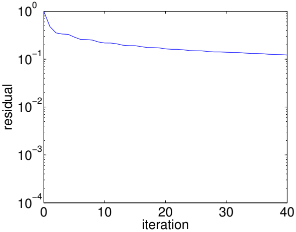
- Non-locality of solutions to $(\Delta + \omega^2)u = f$





- Difficulty of Krylov Methods: GMRES





- Algebraic preconditioners with QMR:

ω	5	10	15	20
QMR	197	737	1775	> 2000
ILU('0')	60	370	> 2000	—
ILU(1e-2)	22	80	220	> 2000

- Multigrid method (see also Brandt, Lifshitz 1997!)

ω	Smoothing steps	2.5π	5π	10π	20π
Iterative	$\nu = 2$	7	div	div	div
Preconditioner	$\nu = 2$	6	12	41	127
Iterative	$\nu = 5$	7	stag	div	div
Preconditioner	$\nu = 5$	5	13	41	223
Iterative	$\nu = 10$	8	div	div	div
Preconditioner	$\nu = 10$	5	10	14	87

- Schwarz methods (see also Després 1991!)

ω	Overlap	10π	20π	40π	80π	160π
Iterative	h	div	div	div	div	div
Preconditioner	h	20	33	45	69	110
Iterative	fixed	div	div	div	div	div
Preconditioner	fixed	16	23	43	86	155

G, Ernst (2012): Why it is difficult to solve Helmholtz problems with classical iterative methods

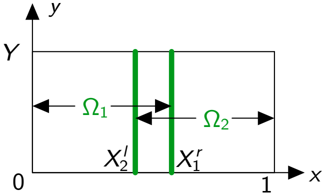
New sweeping type DD preconditioners

- **Engquist and Ying (2011):** Sweeping preconditioner for the Helmholtz equation: moving perfectly matched layers
- **Chen, Xiang (2013):** A source transfer domain decomposition method for Helmholtz equations in unbounded domain I and II
- **Stolk (2013):** A rapidly converging domain decomposition method for the Helmholtz equation
- **Zepeda-Núñez and Demanet (2018):** Nested domain decomposition with polarized traces for the 2D Helmholtz equation
- **Graham, Spence, Zou (2020):** Domain decomposition with local impedance conditions for the Helmholtz equation with absorption

All these are Optimized Schwarz Methods

G, Zhang (2019): A class of iterative solvers for the Helmholtz equation: Factorizations, sweeping preconditioners, source transfer, single layer potentials, polarized traces, and optimized Schwarz methods.

$$(\Delta + \omega^2)u = f \quad \text{in } \Omega := (0,1)\times(0,Y)$$



Subdomains: $\Omega_1 := (0, X_1^r)\times(0,Y)$, $\Omega_2 := (X_2^l, 1)\times(0,Y)$

$$\begin{aligned}(\Delta+\omega^2)u_1^n &= f && \text{in } \Omega_1, & (\Delta+\omega^2)u_2^n &= f && \text{in } \Omega_2\\ \mathcal{B}_1^r(u_1^n) &= \mathcal{B}_1^r(u_2^{n-1}) && \text{at } X_1^r, & \mathcal{B}_2^l(u_2^n) &= \mathcal{B}_2^l(u_1^n) && \text{at } X_2^l\end{aligned}$$

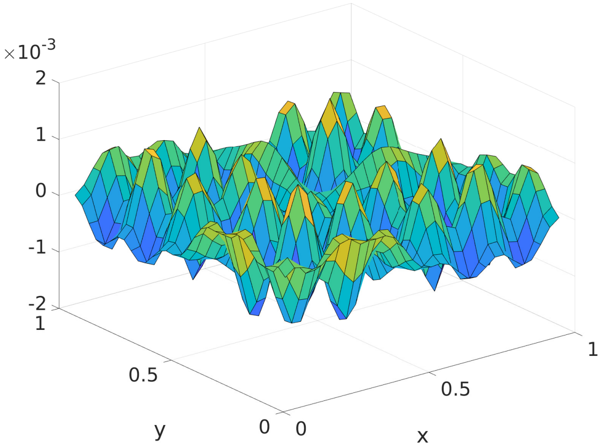
General transmission conditions of the form

$$\mathcal{B}_1^r(u) := \partial_{n_1}u + \mathcal{S}_1^r(u), \quad \mathcal{B}_2^l(u) := \partial_{n_2}u + \mathcal{S}_2^l(u).$$

However their performance also deteriorates!

G, Magoules, Nataf (2002): Optimized Schwarz Methods without Overlap for the Helmholtz Equation

Convergence factor is $1 - O(\omega^{-\frac{1}{4}})$

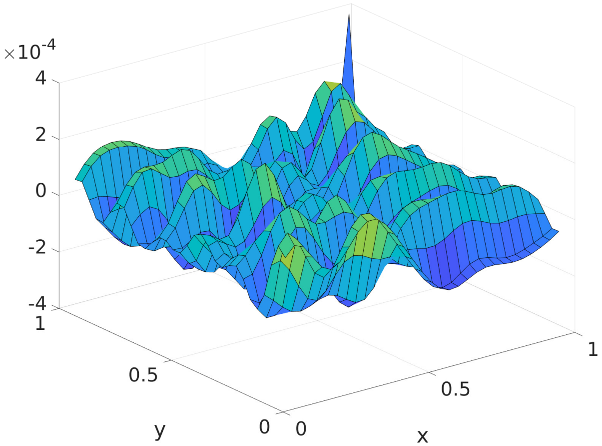


Helmholtz solution in a cavity

However their performance also deteriorates?

G, Magoules, Nataf (2002): Optimized Schwarz Methods without Overlap for the Helmholtz Equation

Convergence factor is $1 - O(\omega^{-\frac{1}{4}})$?

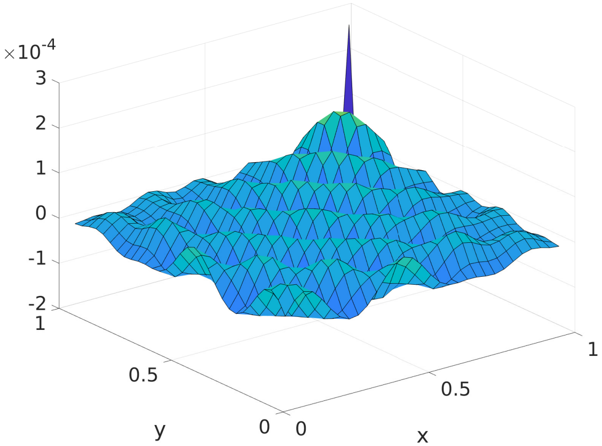


Helmholtz solution in a wave guide

However their performance also deteriorates?

G, Magoules, Nataf (2002): Optimized Schwarz Methods without Overlap for the Helmholtz Equation

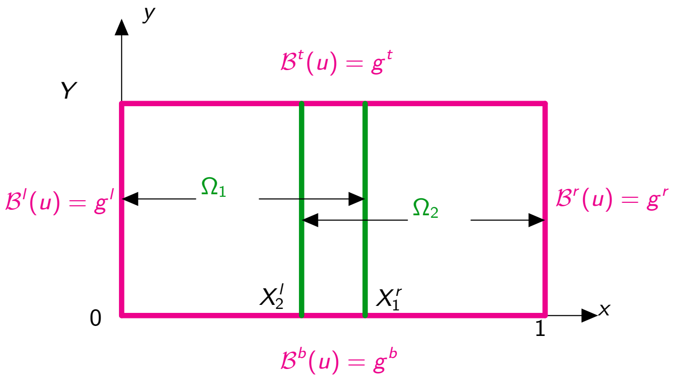
Convergence factor is $1 - O(\omega^{-\frac{1}{4}})$???



Helmholtz solution in free space

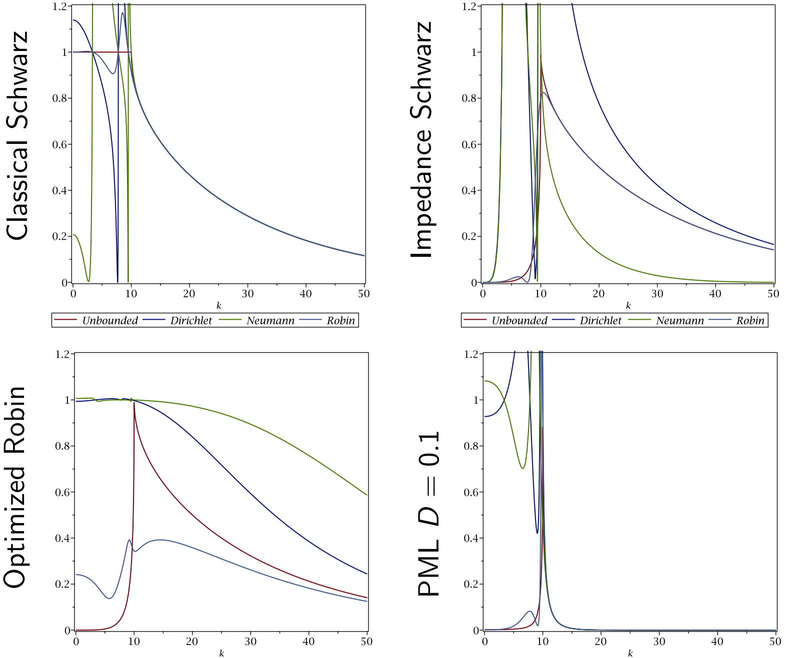
Analysis of these new DD methods

G, Zhang (2022): Schwarz Methods by Domain Truncation, Acta Numerica.

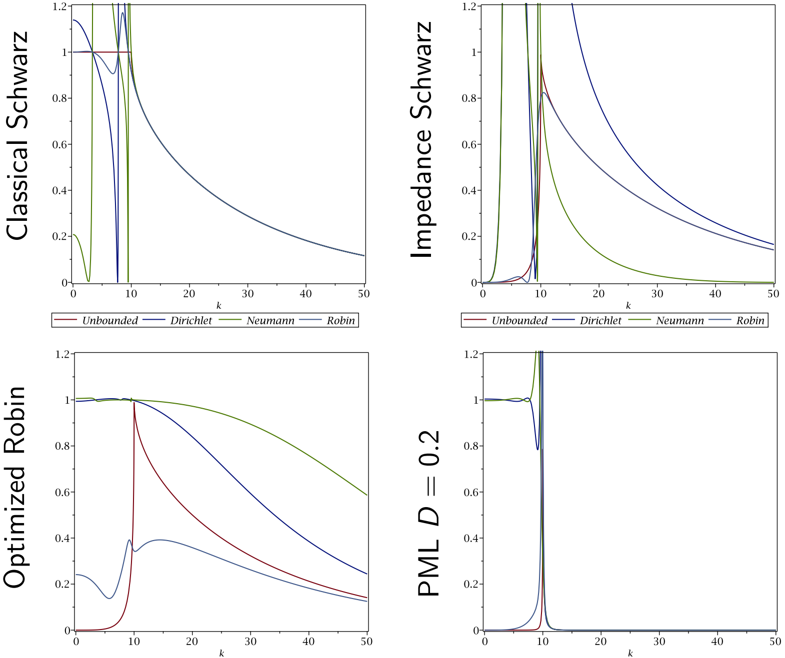


$$\mathcal{B}^\ell(u) := \partial_n u + p^\ell u = g^\ell, \quad \ell \in \{l, r, t, b\}$$

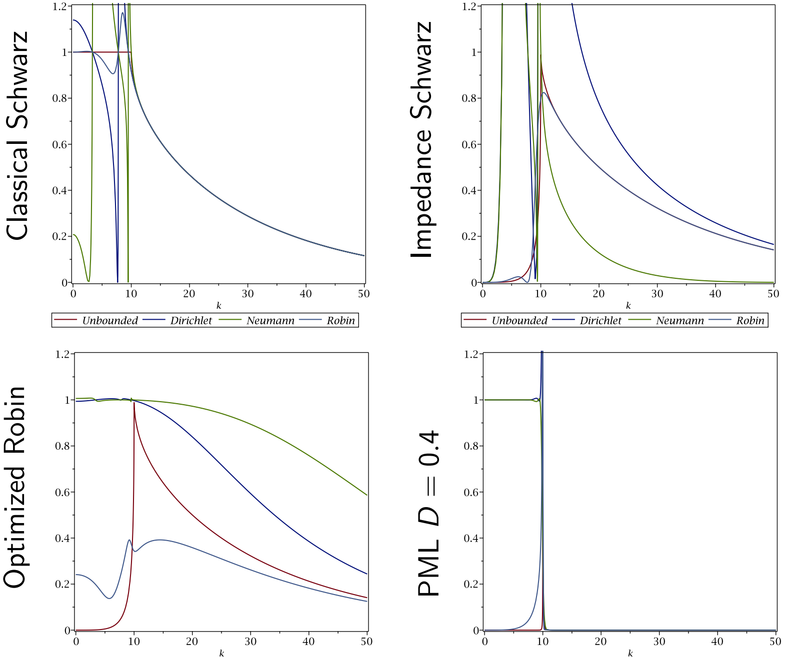
Dirichlet Top/Bottom, Vary Left/Right



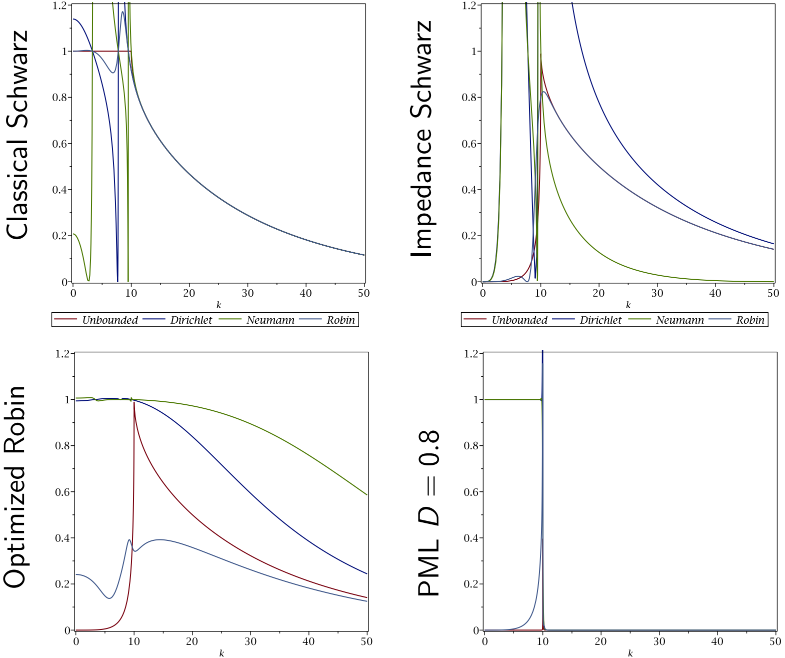
Dirichlet Top/Bottom, Vary Left/Right



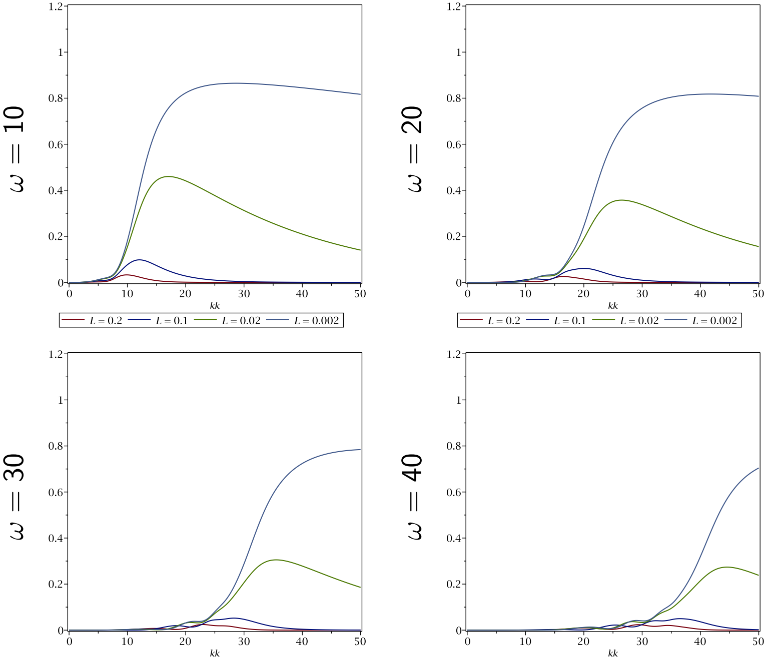
Dirichlet Top/Bottom, Vary Left/Right



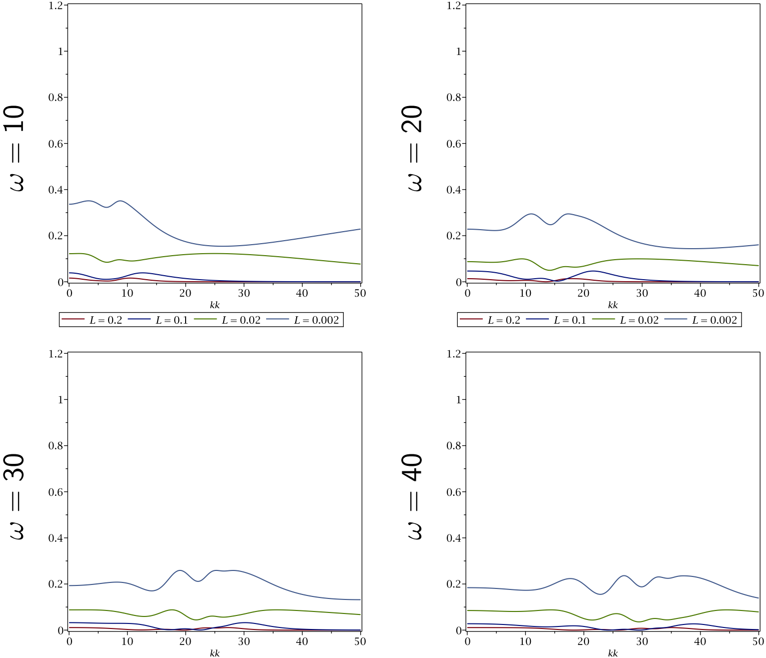
Dirichlet Top/Bottom, Vary Left/Right



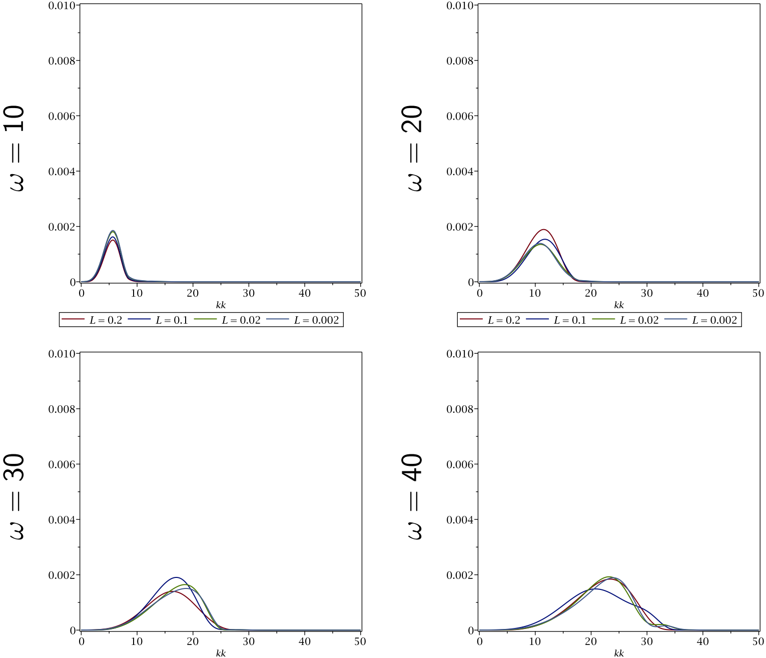
Helmholtz free Space $(\partial_n + i\omega)u = 0$ also TC



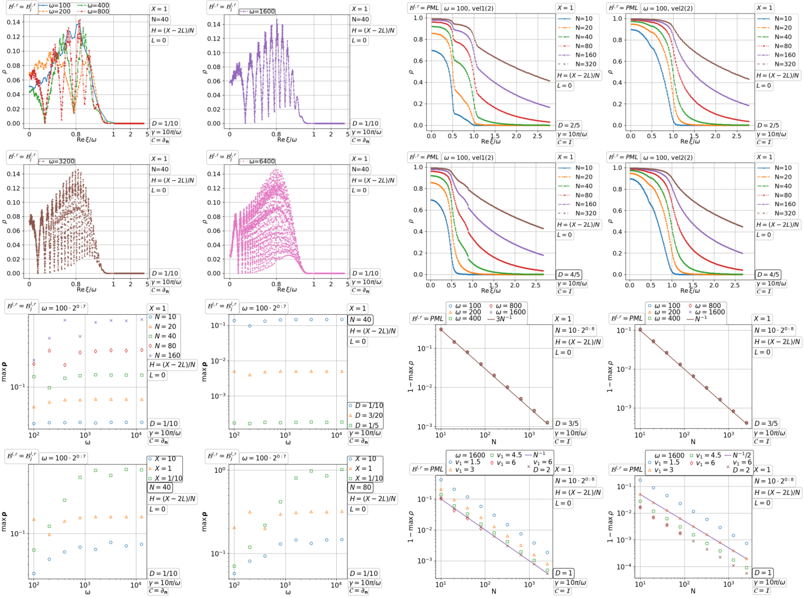
Helmholtz free Space $(\partial_n + i\omega)u = 0$ OSM



Helmholtz free Space $(\partial_n + i\omega)u = 0$ PML



Many many results in Acta Numerica 2022



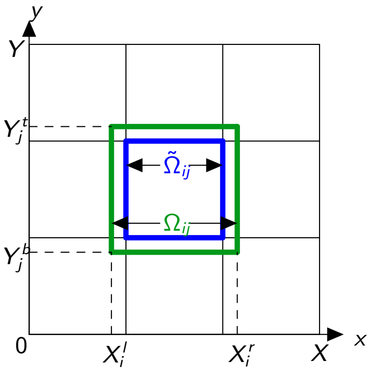
G, Tonnoir 2024: Schwarz for Convected Helmholtz Equations

New 2: Scalability and Coarse Spaces

Toselli and Widlund (2005): Classical view in DD

Therefore an iterative method for the solution of the resulting linear system in which information is only exchanged between neighboring subregions must necessarily, for certain initial errors, require a number of steps which is at least equal to the diameter of the dual graph corresponding to the subdomain partition.

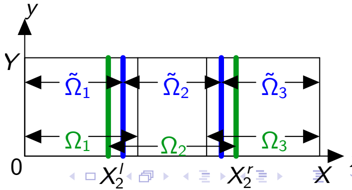
In Sect. 5.4, we then consider the problems of devising efficient coarse solvers, which are the key and a quite delicate part of any successful preconditioners for three-dimensional problems.



$\tilde{\Omega}_{ij}$ non-overlapping, and

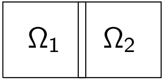
Ω_{ij} enlarged overlapping:

$$\begin{aligned} \Delta u_{ij}^n &= f && \text{in } \Omega_{ij} \\ u_{ij}^n &= u_{kl}^{n-1} && \text{on } \partial\Omega_{ij} \cap \tilde{\Omega}_{kl} \end{aligned}$$



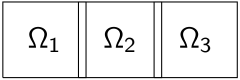
One level DD methods can be scalable!

One level Schwarz methods for $\Delta u = f$ are scalable with Dirichlet conditions when adding subdomains like



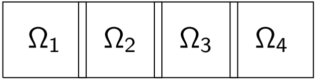
One level DD methods can be scalable!

One level Schwarz methods for $\Delta u = f$ are scalable with Dirichlet conditions when adding subdomains like



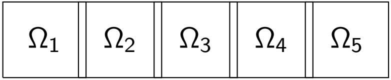
One level DD methods can be scalable!

One level Schwarz methods for $\Delta u = f$ are scalable with Dirichlet conditions when adding subdomains like



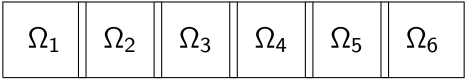
One level DD methods can be scalable!

One level Schwarz methods for $\Delta u = f$ are scalable with Dirichlet conditions when adding subdomains like



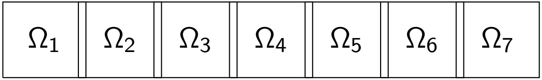
One level DD methods can be scalable!

One level Schwarz methods for $\Delta u = f$ are scalable with Dirichlet conditions when adding subdomains like



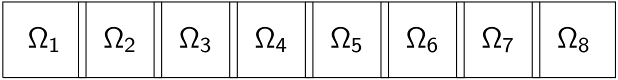
One level DD methods can be scalable!

One level Schwarz methods for $\Delta u = f$ are scalable with Dirichlet conditions when adding subdomains like



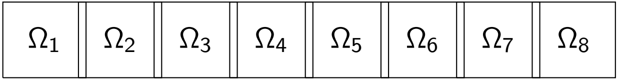
One level DD methods can be scalable!

One level Schwarz methods for $\Delta u = f$ are scalable with Dirichlet conditions when adding subdomains like

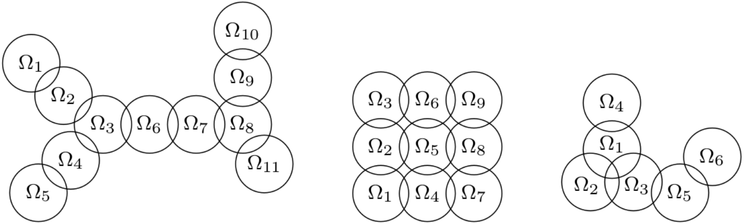


One level DD methods can be scalable!

One level Schwarz methods for $\Delta u = f$ are scalable with Dirichlet conditions when adding subdomains like

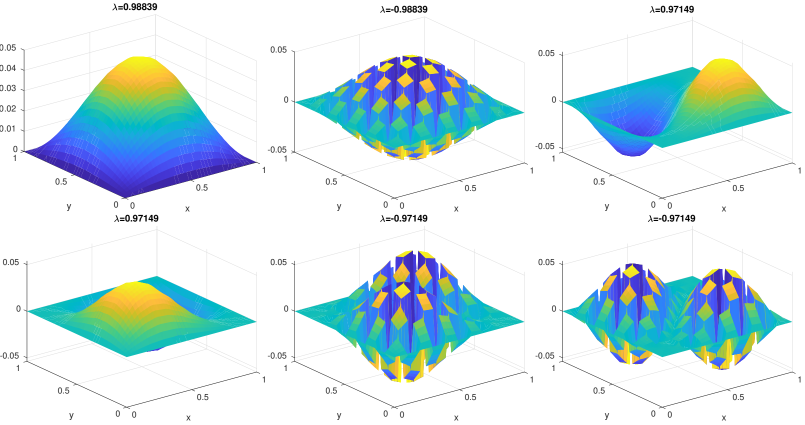


Ciaramella and G. (2017): Analysis of the parallel Schwarz method for the solution of chains of particles, Part I-III
Three different proofs: Fourier in L^2, maximum principle in L^∞ and alternating projection interpretation in H^1



Chaouqui, Ciaramella, G, Vanzan (2018): On the scalability of classical one-level domain-decomposition

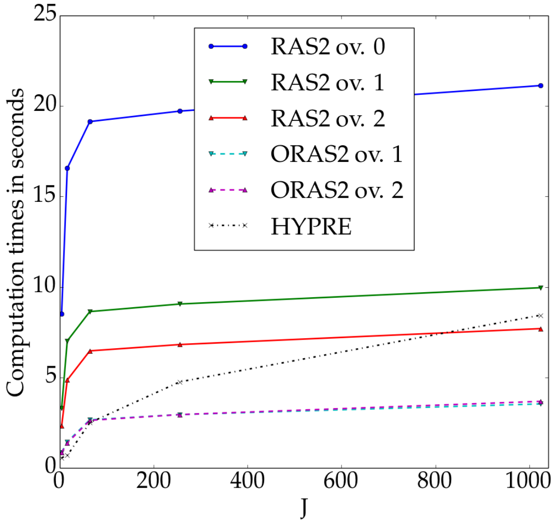
Eigenmodes of the Schwarz iteration operator



$\Longrightarrow$ Should use *discontinuous Q1 coarse space aligned with subdomains* or relaxation parameter 2/3 in two level Schwarz

Hackbusch (1985): Multigrid methods and applications
Xu (1992): Iterative methods by space decomposition and subspace correction

PETSc comparison with HPYRE/BoomerAMG



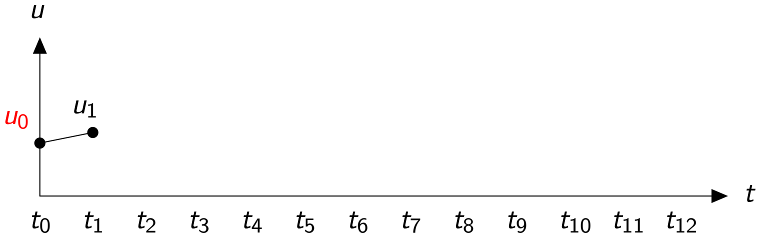
G, Van Criekingen (2020): New coarse corrections for ORAS using PETSc

New 3: Time Parallelization (PinT)

The time direction is special for parallelization, because of the *causality principle*: the solution later in time is determined by the solution earlier in time, and never the other way round.

Example: $\frac{du}{dt} = f(u)$, $u(t_0) = u_0$, Euler: $\frac{du}{dt} \approx \frac{u(t_{n+1})-u(t_n)}{\Delta t}$

$$u_1 = u_0 + \Delta t f(u_0)$$

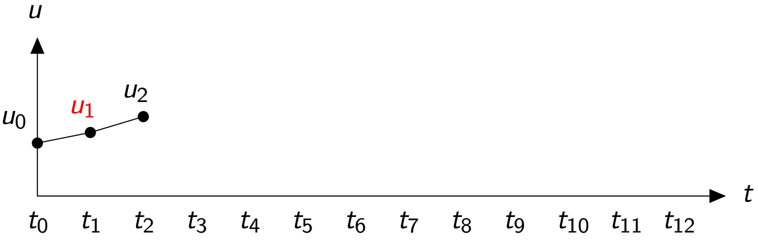


New 3: Time Parallelization (PinT)

The time direction is special for parallelization, because of the *causality principle*: the solution later in time is determined by the solution earlier in time, and never the other way round.

Example: $\frac{du}{dt} = f(u)$, $u(t_0) = u_0$, Euler: $\frac{du}{dt} \approx \frac{u(t_{n+1})-u(t_n)}{\Delta t}$

$$u_2 = u_1 + \Delta t f(u_1)$$

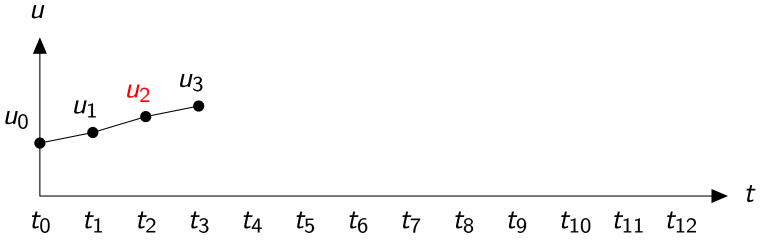


New 3: Time Parallelization (PinT)

The time direction is special for parallelization, because of the *causality principle*: the solution later in time is determined by the solution earlier in time, and never the other way round.

Example: $\frac{du}{dt} = f(u)$, $u(t_0) = u_0$, Euler: $\frac{du}{dt} \approx \frac{u(t_{n+1})-u(t_n)}{\Delta t}$

$$u_3 = u_2 + \Delta t f(u_2)$$

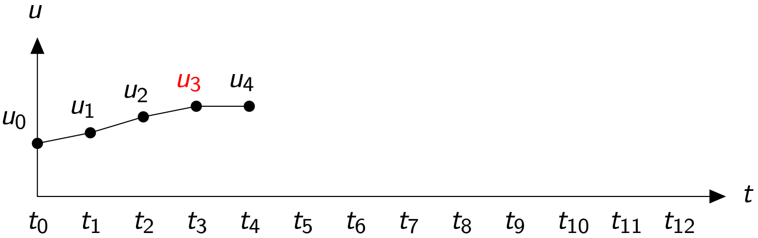


New 3: Time Parallelization (PinT)

The time direction is special for parallelization, because of the *causality principle*: the solution later in time is determined by the solution earlier in time, and never the other way round.

Example: $\frac{du}{dt} = f(u)$, $u(t_0) = u_0$, Euler: $\frac{du}{dt} \approx \frac{u(t_{n+1}) - u(t_n)}{\Delta t}$

$$u_4 = u_3 + \Delta t f(u_3)$$

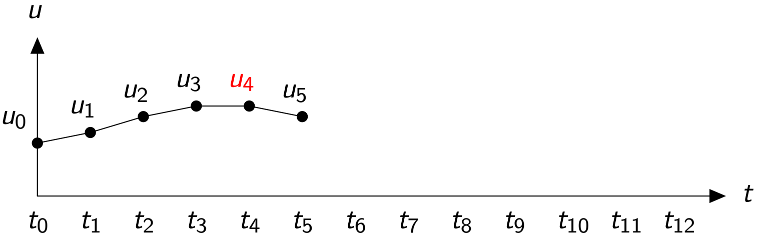


New 3: Time Parallelization (PinT)

The time direction is special for parallelization, because of the *causality principle*: the solution later in time is determined by the solution earlier in time, and never the other way round.

Example: $\frac{du}{dt} = f(u)$, $u(t_0) = u_0$, Euler: $\frac{du}{dt} \approx \frac{u(t_{n+1})-u(t_n)}{\Delta t}$

$$u_5 = u_4 + \Delta t f(u_4)$$

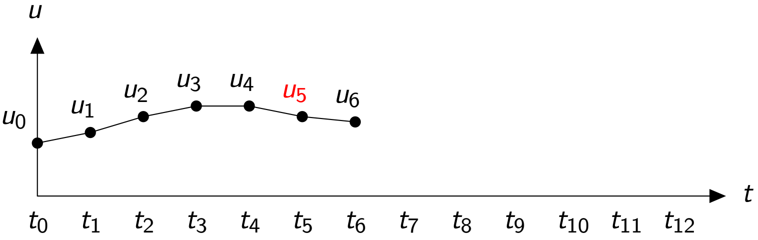


New 3: Time Parallelization (PinT)

The time direction is special for parallelization, because of the *causality principle*: the solution later in time is determined by the solution earlier in time, and never the other way round.

Example: $\frac{du}{dt} = f(u)$, $u(t_0) = u_0$, Euler: $\frac{du}{dt} \approx \frac{u(t_{n+1}) - u(t_n)}{\Delta t}$

$$u_6 = u_5 + \Delta t f(u_5)$$

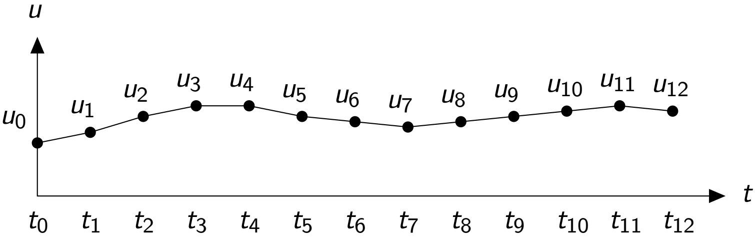


New 3: Time Parallelization (PinT)

The time direction is special for parallelization, because of the *causality principle*: the solution later in time is determined by the solution earlier in time, and never the other way round.

Example: $\frac{du}{dt} = f(u)$, $u(t_0) = u_0$, Euler: $\frac{du}{dt} \approx \frac{u(t_{n+1}) - u(t_n)}{\Delta t}$

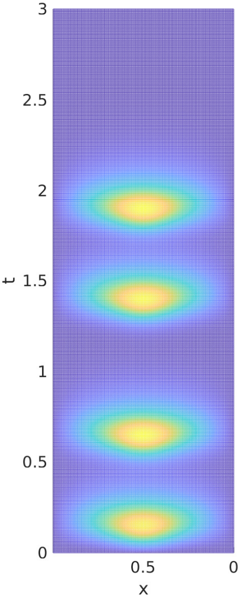
$$u_{n+1} = u_n + \Delta t f(u_n)$$

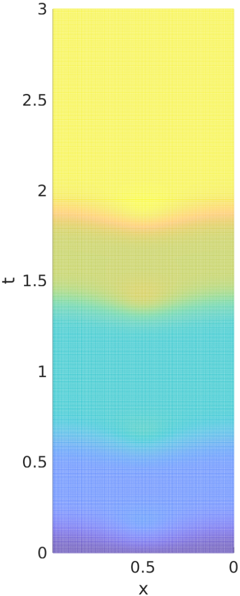


Domain decomposition in time ?

Heat Equation: Dirichlet and Neumann Conditions

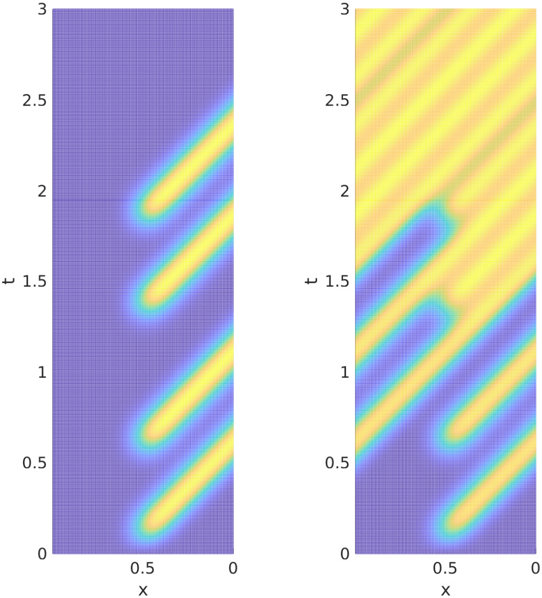
$$u_t = u_{xx} + f,\ u(0,t) = u(1,t) = 0 \text{ and } u_x(0,t) = u_x(1,t) = 0$$





Transport Equation: Dirichlet and Periodic

$u_t + u_x = f$, $u(0,t) = 0$ and $u(0,t) = u(1,t)$



Parabolic PinT: the Parareal Algorithm

For solving the evolution problem

$$\begin{aligned} \partial_t \boldsymbol{u}(t) &= \boldsymbol{f}(t, \boldsymbol{u}(t)) \quad t \in (0, T], \\ \boldsymbol{u}(0) &= \boldsymbol{u}^0, \end{aligned}$$

The time domain $(0, T]$ is partitioned into subdomains $(T_{n-1}, T_n]$, and Parareal needs two propagation operators:

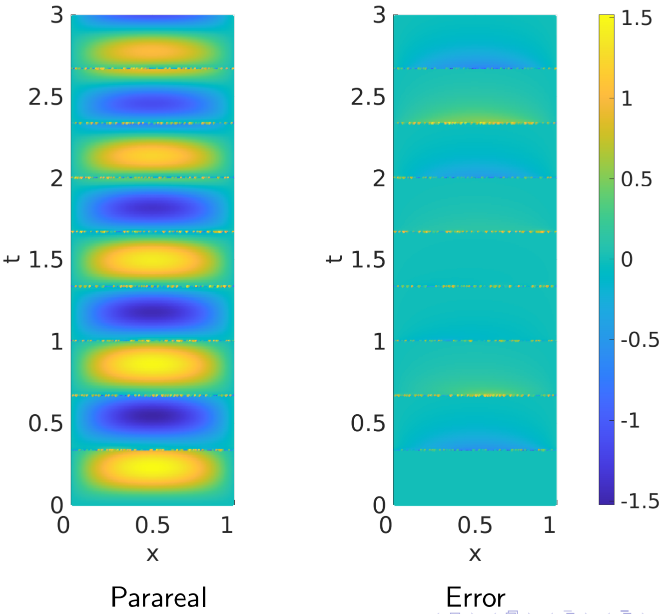
1. $\boldsymbol{G}(t_2, t_1, \boldsymbol{u}_1)$ is a coarse approximation to the solution $\boldsymbol{u}(t_2)$ with initial condition $\boldsymbol{u}(t_1) = \boldsymbol{u}_1$,
2. $\boldsymbol{F}(t_2, t_1, \boldsymbol{u}_1)$ is a more accurate approximation of the solution $\boldsymbol{u}(t_2)$ with initial condition $\boldsymbol{u}(t_1) = \boldsymbol{u}_1$.

Parareal then starts with an initial coarse approximation $\boldsymbol{U}_n^0$ at $T_0, T_1, \ldots, T_N$, and computes

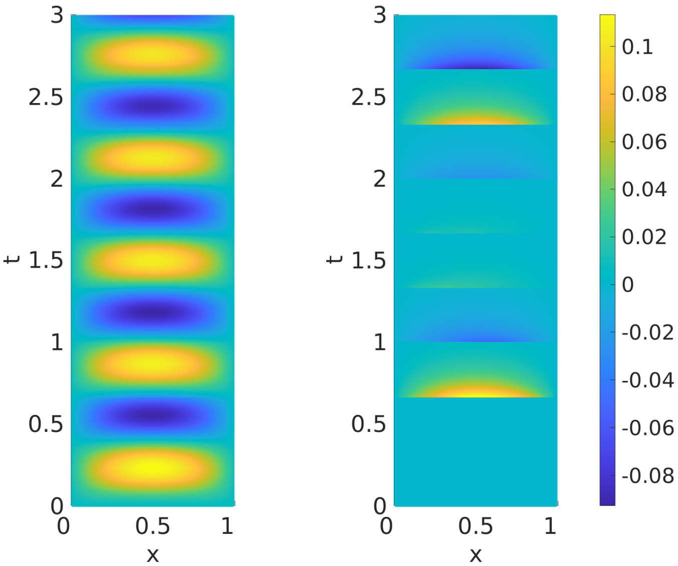
$$\begin{aligned} &\boldsymbol{U}_0^{k+1} := \boldsymbol{u}^0, \\ &\boldsymbol{U}_{n+1}^{k+1} := \boldsymbol{F}(T_{n+1}, T_n, \boldsymbol{U}_n^k) + \boldsymbol{G}(T_{n+1}, T_n, \boldsymbol{U}_n^{k+1}) - \boldsymbol{G}(T_{n+1}, T_n, \boldsymbol{U}_n^k) \end{aligned}$$

Lions, Maday, Turinici (2001): Résolution d'EDP par un schéma en temps "pararéel"

Heat example: Iteration 1

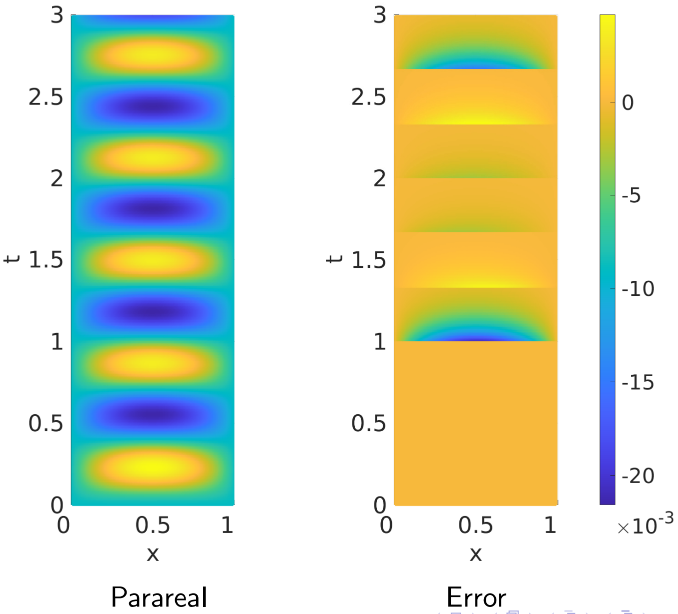


Heat example: Iteration 2

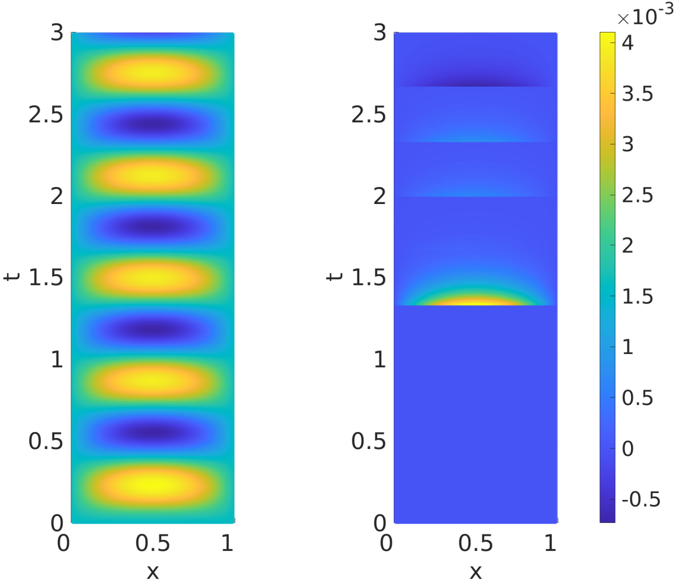


Parareal Error

Heat example: Iteration 3



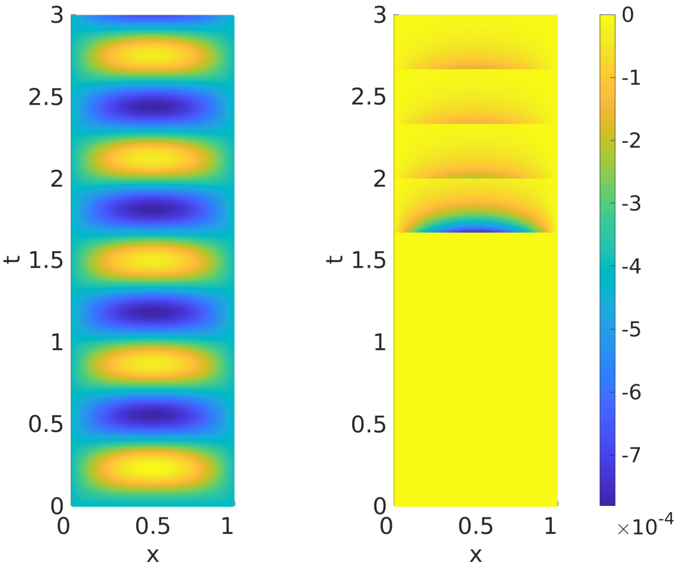
Heat example: Iteration 4



Parareal

Error

Heat example: Iteration 5



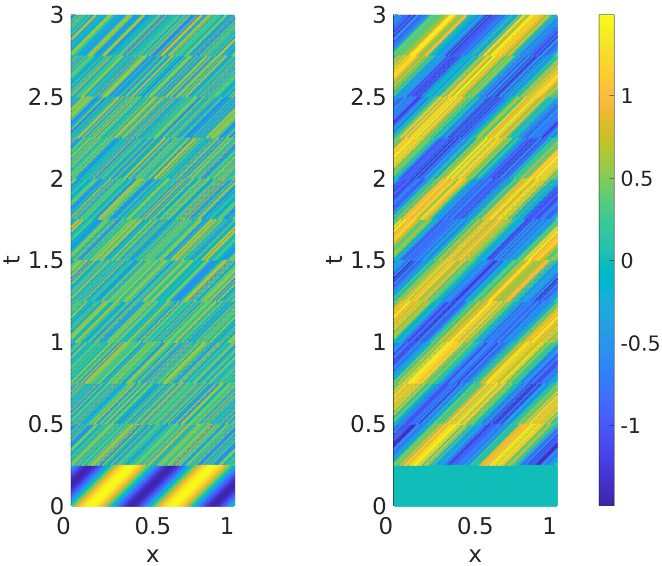
Parareal Error

Much better than Parareal: Heat eq. in 3D

G, Neumüller (2016): Space-Time Parallel Multigrid

cores	time steps	dof	iter	time	fwd. sub.
1	2	59 768	7	28.8	19.0
2	4	119 536	7	29.8	37.9
4	8	239 072	7	29.8	75.9
8	16	478 144	7	29.9	152.2
16	32	956 288	7	29.9	305.4
32	64	1 912 576	7	29.9	613.6
64	128	3 825 152	7	29.9	1 220.7
128	256	7 650 304	7	29.9	2 448.4
256	512	15 300 608	7	30.0	4 882.4
512	1 024	30 601 216	7	29.9	9 744.2
1 024	2 048	61 202 432	7	30.0	19 636.9
2 048	4 096	122 404 864	7	29.9	38 993.1
4 096	8 192	244 809 728	7	30.0	81 219.6
8 192	16 384	489 619 456	7	30.0	162 551.0
16 384	32 768	979 238 912	7	30.0	313 122.0
32 768	65 536	1 958 477 824	7	30.0	625 686.0
65 536	131 072	3 916 955 648	7	30.0	1 250 210.0
131 072	262 144	7 833 911 296	7	30.0	2 500 350.0
262 144	524 288	15 667 822 592	7	30.0	4 988 060.0

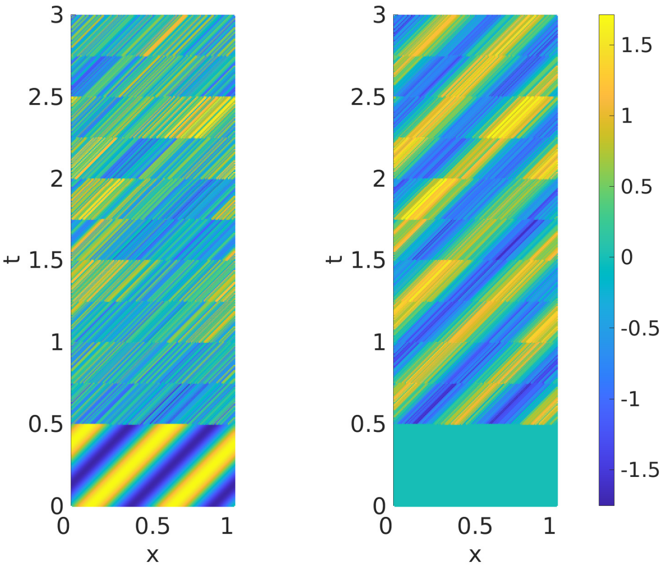
Parareal advection example: Iteration 1



Parareal

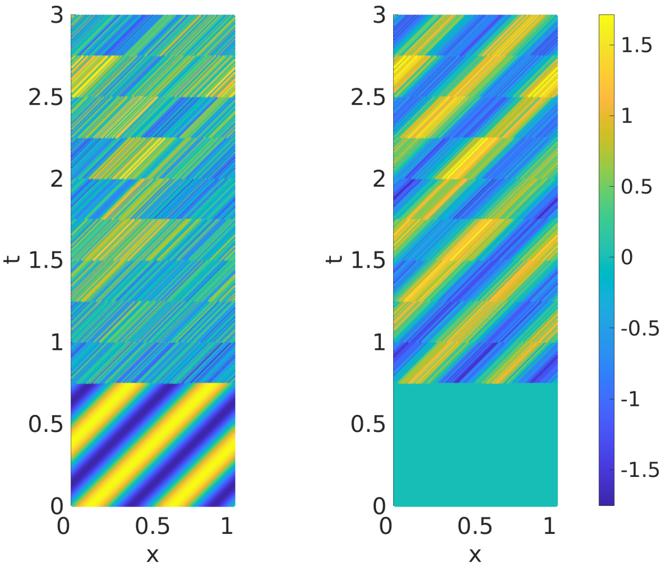
Error

Parareal advection example: Iteration 2



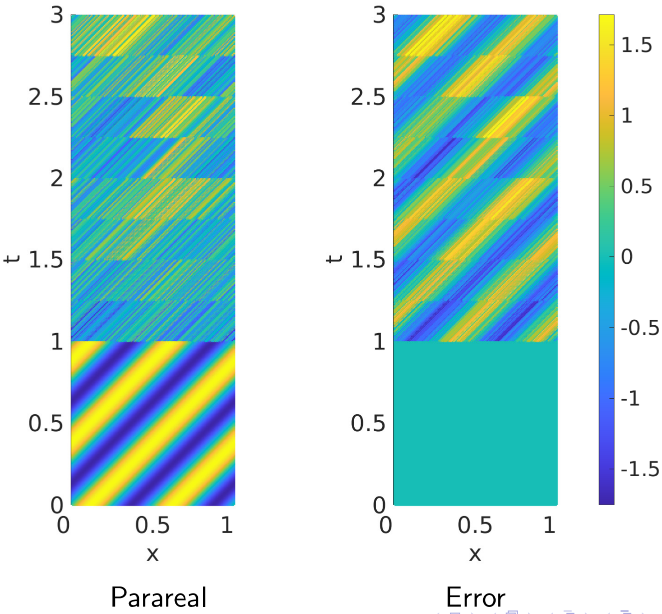
Parareal
Error

Parareal advection example: Iteration 3

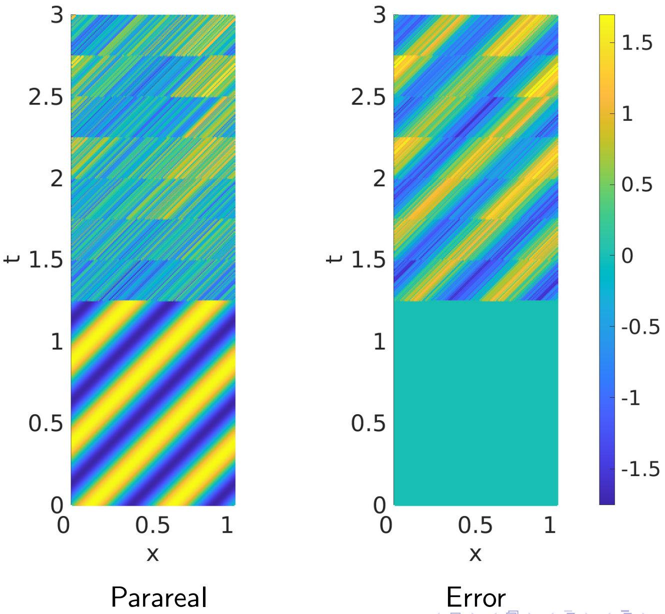


Parareal Error

Parareal advection example: Iteration 4



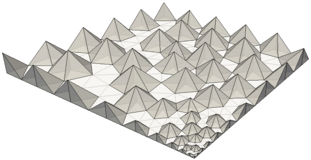
Parareal advection example: Iteration 5

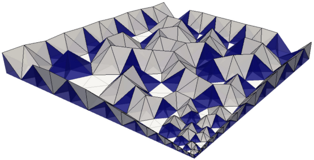


Hyperbolic PinT: Mapped Tent Pitching (MTP)

Gopalakrishnan, Schöberl, Wintersteiger (2017): Mapped Tent Pitching Schemes for Hyperbolic Systems

"This paper explores a technique by which standard discretizations, including explicit time stepping, can be used within tent-shaped spacetime domains. The technique transforms the equations within a spacetime tent to a domain where space and time are separable."

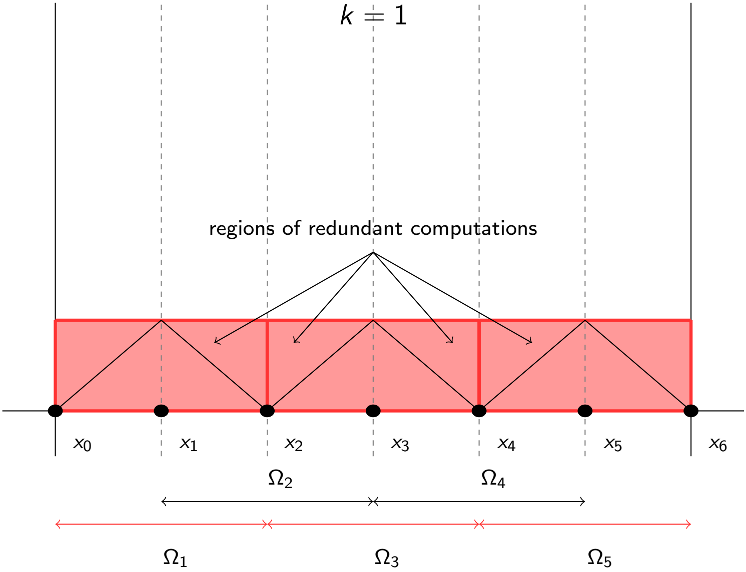




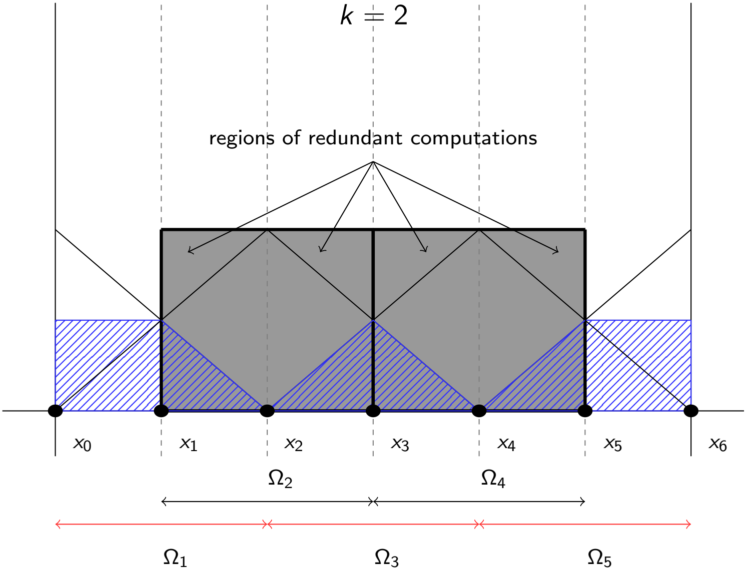
Gopalakrishnan, Hochteger, Schöberl, Wintersteiger (2020): An Explicit Mapped Tent Pitching Scheme for Maxwell Equations

Probably the best PinT Maxwell solver currently available!

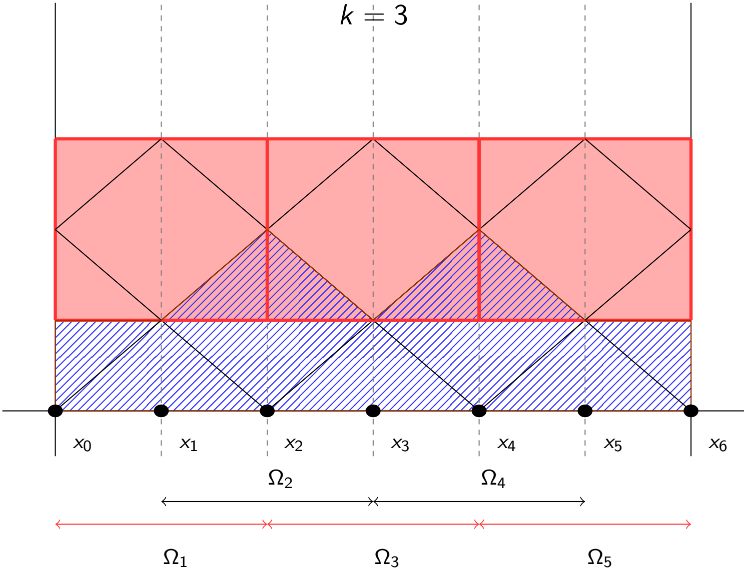
Red Black Schwarz Waveform Relaxation



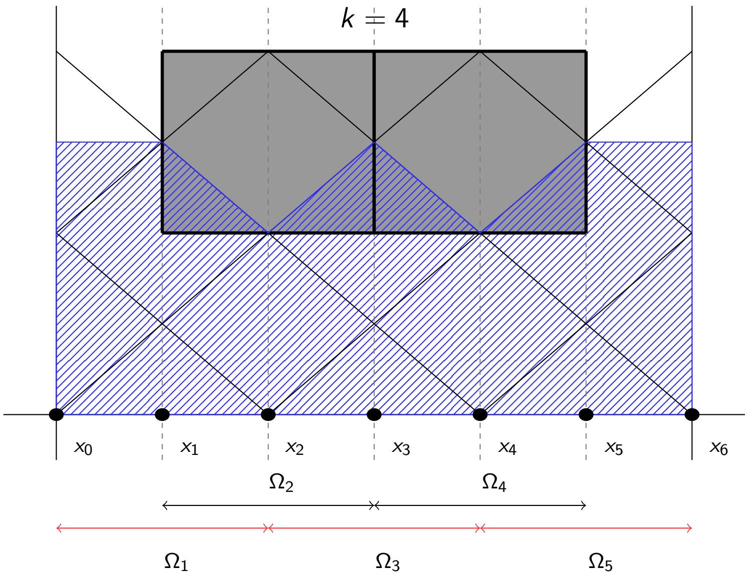
Second Iteration of RBSWR



Third Iteration of RBSWR

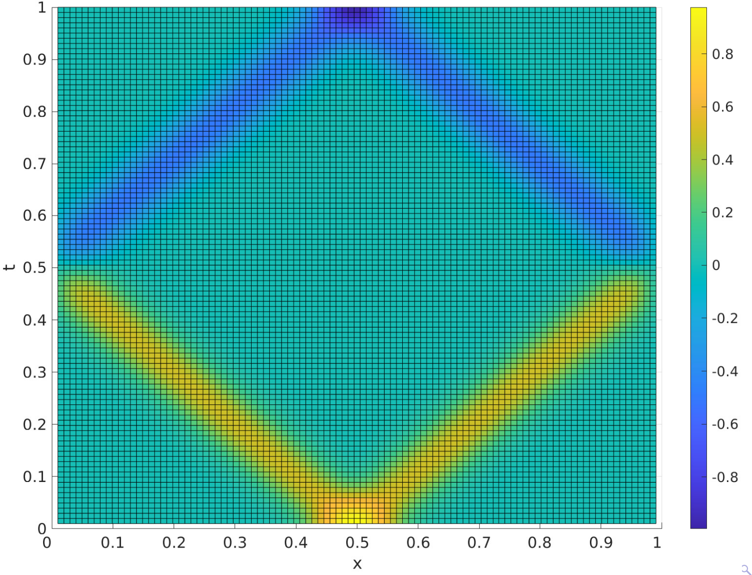


Fourth Iteration of RBSWR



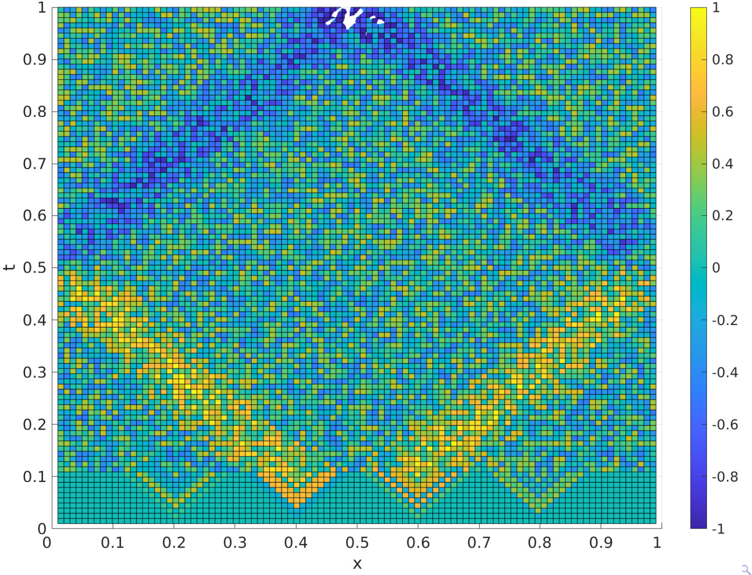
Red Black Schwarz Waveform Relaxation

Ciaramella, G, Mazzieri (2023): Unmapped tent pitching schemes by waveform relaxation



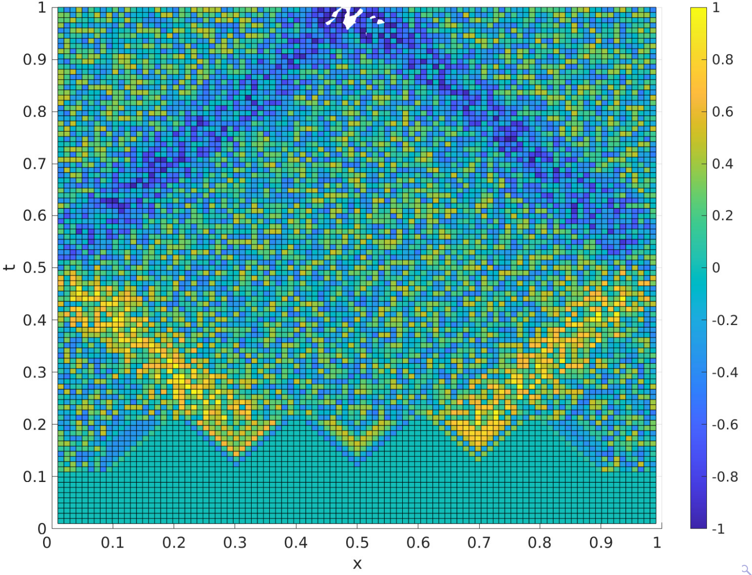
Red iteration 1 error

Ciaramella, G, Mazzieri (2023): Unmapped tent pitching schemes by waveform relaxation



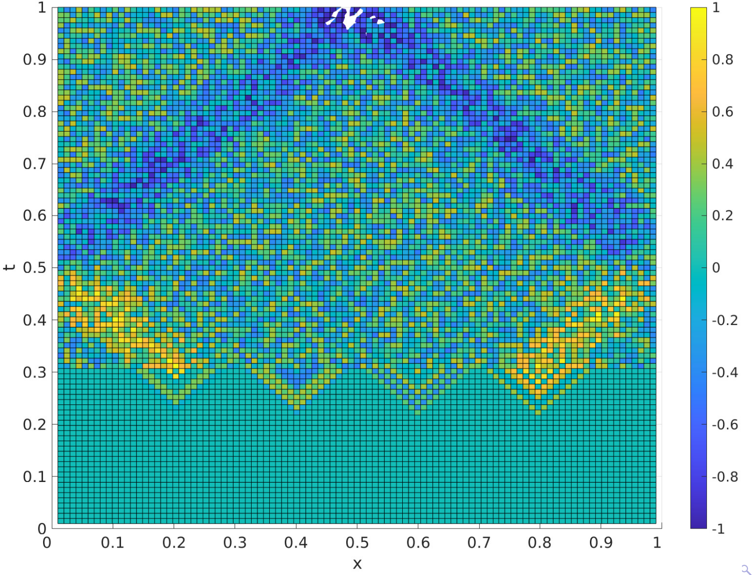
Black iteration 1 error

Ciaramella, G, Mazzieri (2023): Unmapped tent pitching schemes by waveform relaxation



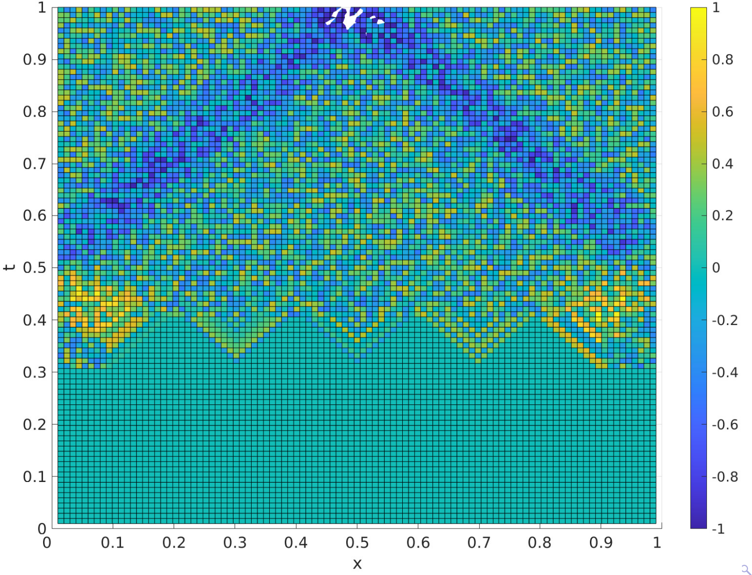
Red iteration 2 error

Ciaramella, G, Mazzieri (2023): Unmapped tent pitching schemes by waveform relaxation



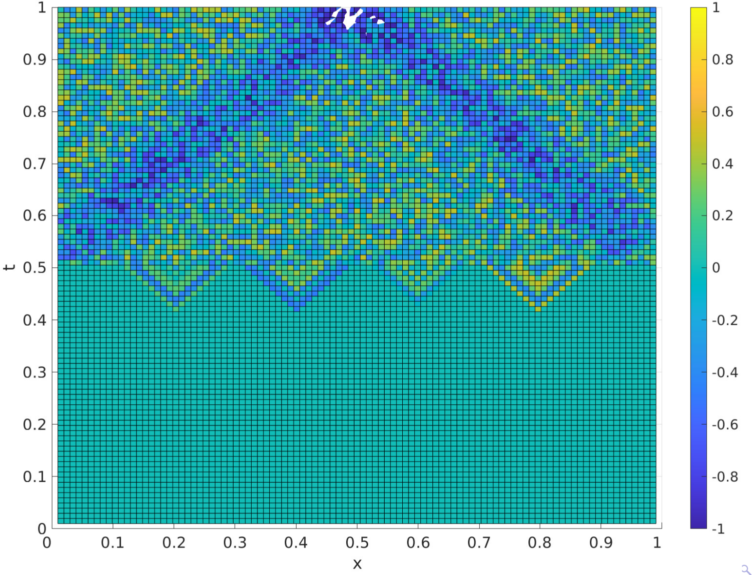
Black iteration 2 error

Ciaramella, G, Mazzieri (2023): Unmapped tent pitching schemes by waveform relaxation



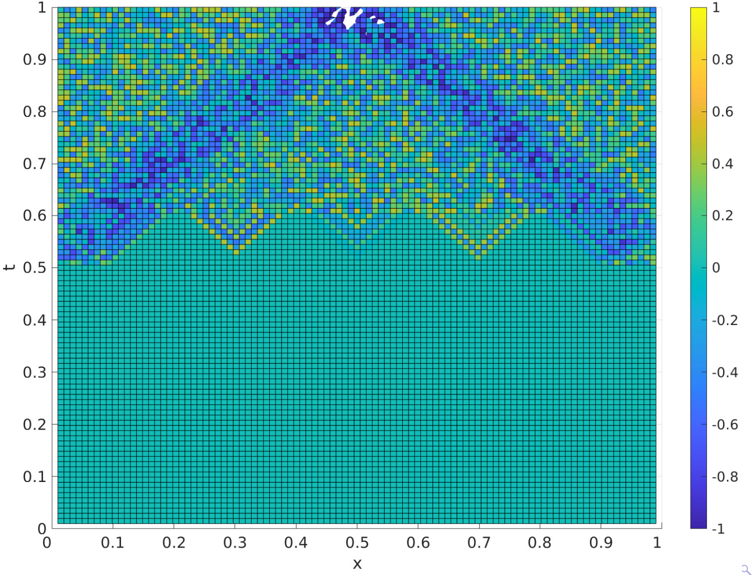
Red iteration 3 error

Ciaramella, G, Mazzieri (2023): Unmapped tent pitching schemes by waveform relaxation



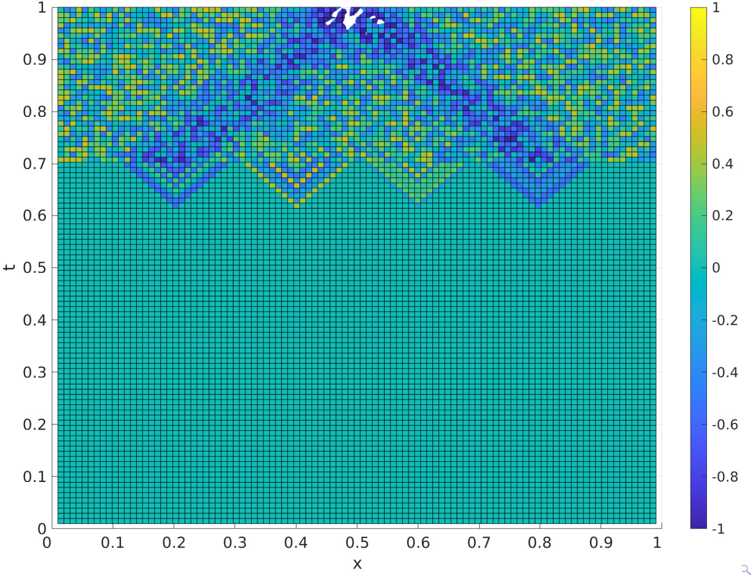
Black iteration 3 error

Ciaramella, G, Mazzieri (2023): Unmapped tent pitching schemes by waveform relaxation



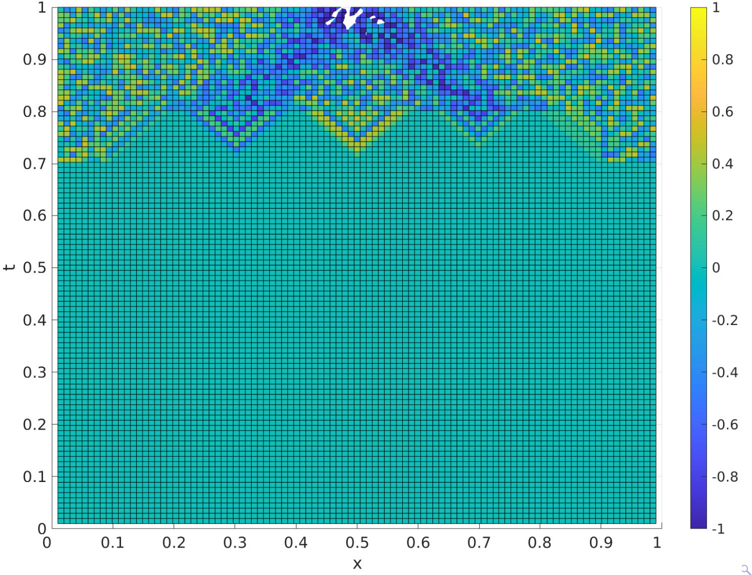
Red iteration 4 error

Ciaramella, G, Mazzieri (2023): Unmapped tent pitching schemes by waveform relaxation



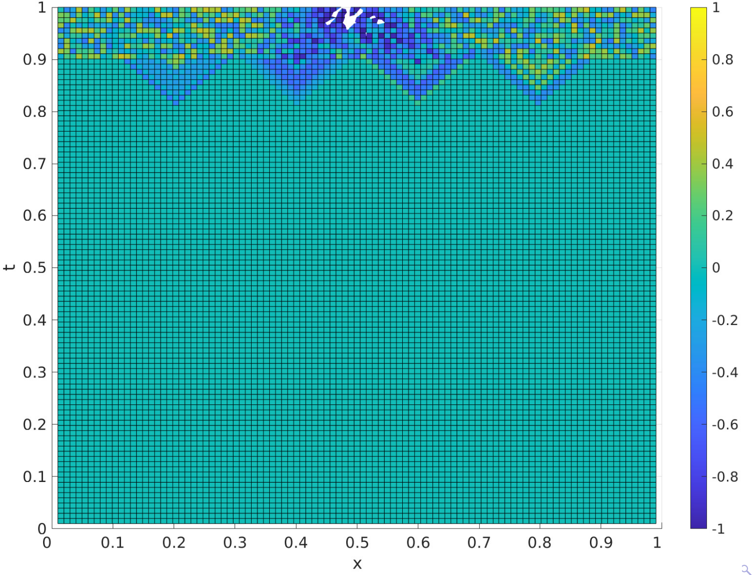
Black iteration 4 error

Ciaramella, G, Mazzieri (2023): Unmapped tent pitching schemes by waveform relaxation



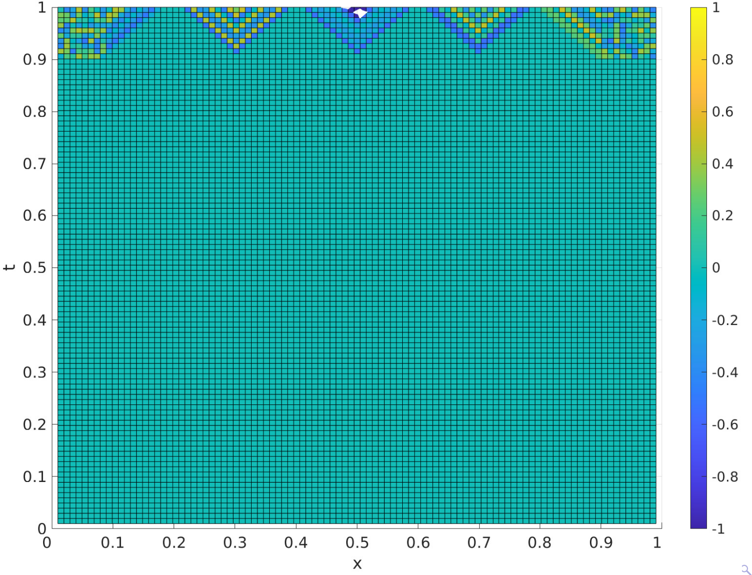
Red iteration 5 error

Ciaramella, G, Mazzieri (2023): Unmapped tent pitching schemes by waveform relaxation



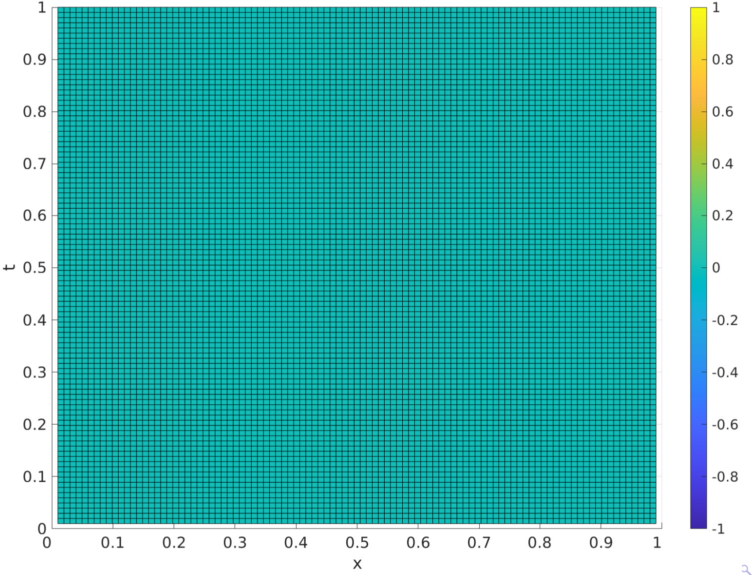
Black iteration 5 error

Ciaramella, G, Mazzieri (2023): Unmapped tent pitching schemes by waveform relaxation



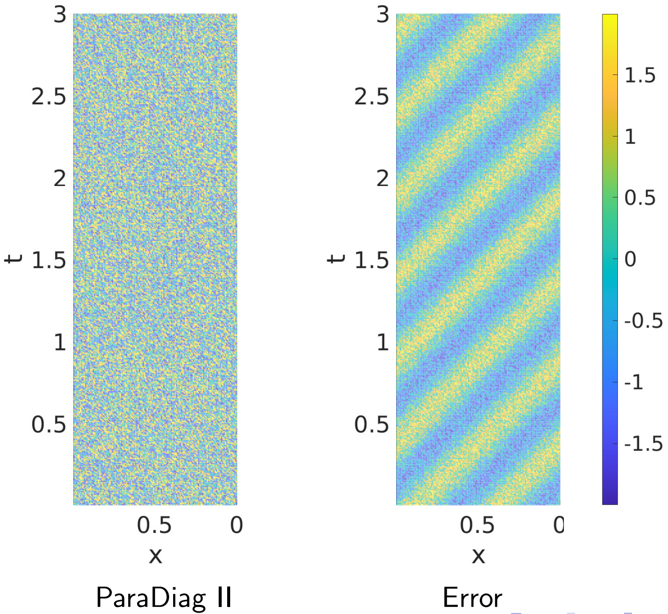
Red iteration 6 error

Ciaramella, G, Mazzieri (2023): Unmapped tent pitching schemes by waveform relaxation

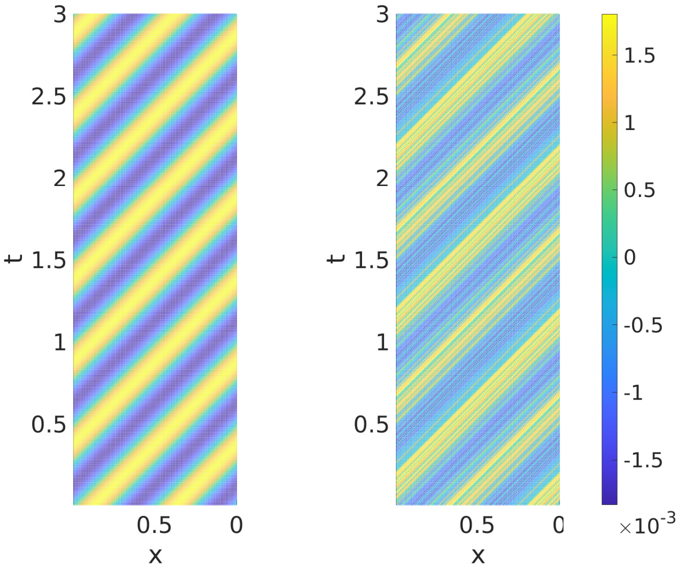


```
[A,f,ue]=AllAtOnceSystem(n,m);
[Rr,Rb,mtr,mtb]=RedBlackSubdomains(n,m,nx);
u=rand(m*n,1)-1/2;
for jr=1:mtr
  for ir=1:nx
    u=u+Rr{ir,jr}'*((Rr{ir,jr}*A*Rr{ir,jr}')...
      \(Rr{ir,jr}*(f-A*u)));
  end;
  U=reshape(ue-u,n,m);
  surf(t,x,U); xlabel('t');ylabel('x'); pause
  if jr<=mtb
    for ib=1:nx-1
      u=u+Rb{ib,jr}'*((Rb{ib,jr}*A*Rb{ib,jr}')...
        \(Rb{ib,jr}*(f-A*u)));
    end;
    U=reshape(ue-u,n,m);
    surf(t,x,U); xlabel('t');ylabel('x'); pause
  end
end;
```

ParaDiag II on advection: Initial Guess

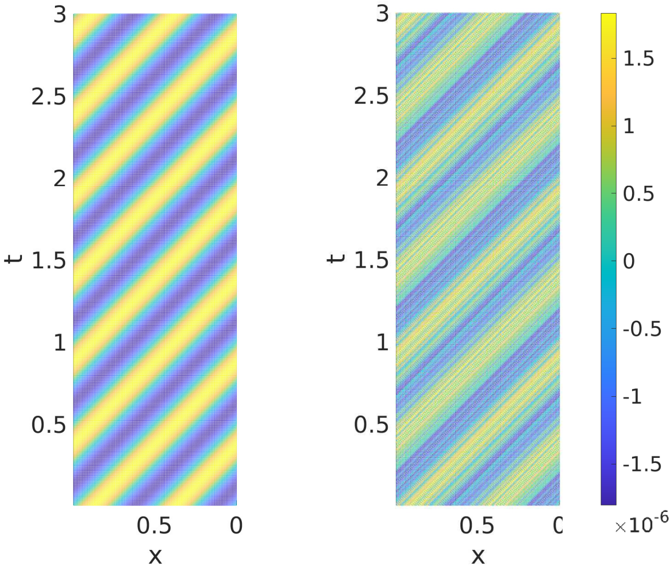


ParaDiag II on advection: Iteration 1



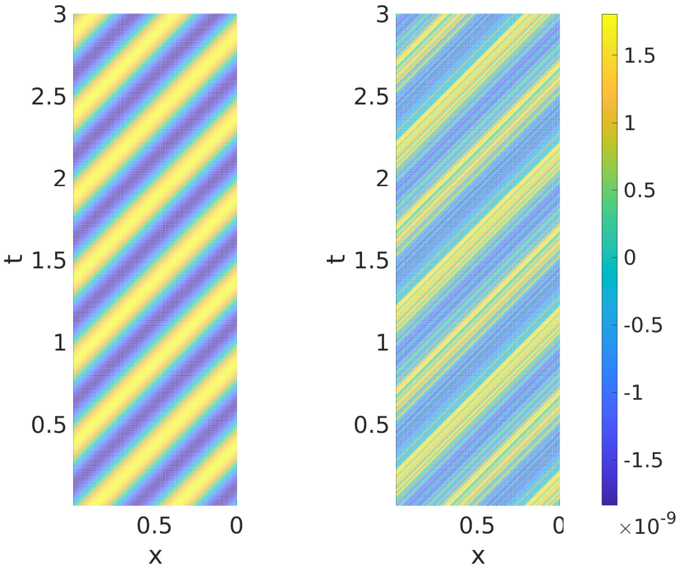
ParaDiag II Error

ParaDiag II on advection: Iteration 2

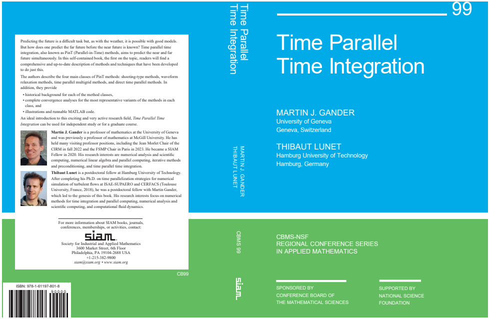


ParaDiag II   Error

ParaDiag II on advection: Iteration 3



ParaDiag II Error



New book (CIRM 2025)

Optimized Schwarz Methods for Anisotropic Diffusion with DDFV Discetizations