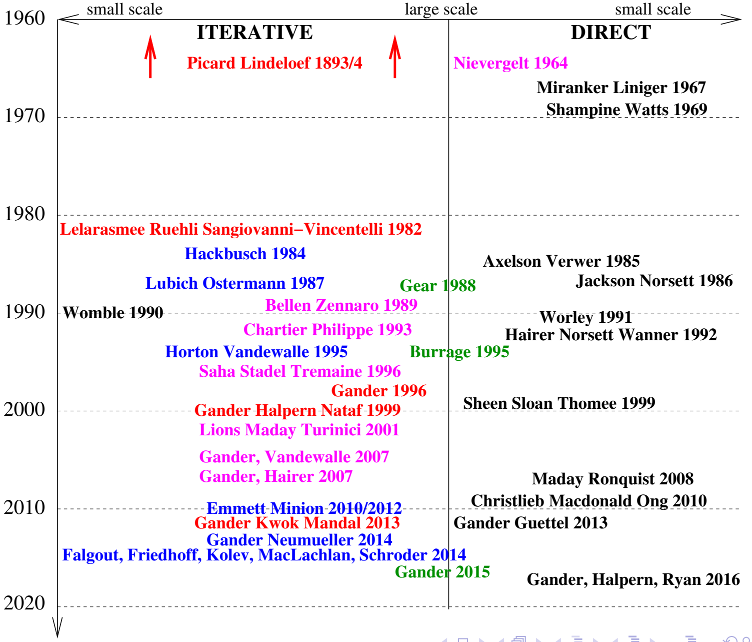
Time Parallel Time Integration
Chapter 4: Space-Time Multigrid Methods

Martin J. Gander
martin.gander@unige.ch

University of Geneva

Michigan, August 4th, 2022

Space-Time Multigrid Methods



Parabolic Multigrid

Wolfgang Hackbusch (1984): Parabolic Multi-Grid Methods

> *"A multi-grid iteration for solving parabolic partial differential equations is presented. It is characterized by the simultaneous computation of several time steps in one step to the computational process."*

One dimensional heat equation as the model problem:

$$\begin{aligned} \partial_t u(x,t) &= \partial_{xx} u(x,t) + f(x,t) && \text{in } \Omega \times (0,T],\ \Omega := (0,L),\\ u(x,0) &= u_0(x) && \text{in } \Omega,\\ u(0,t) &= g_0(t) && \text{in } (0,T],\\ u(L,t) &= g_L(t) && \text{in } (0,T]. \end{aligned}$$

Centered differences in space and Backward Euler in time:

$$\frac{\boldsymbol{u}_{n+1} - \boldsymbol{u}_n}{\Delta t} = L\boldsymbol{u}_{n+1} + \boldsymbol{f}_{n+1},$$

where $L := \frac{1}{\Delta x^2}[1 \quad -2 \quad 1] \in \mathbb{R}^{J \times J}$.

Hackbusch's Idea for a Smoother

"The conventional approach is to solve time step by time step; $\boldsymbol{u}_{n+1}$ is computed from $\boldsymbol{u}_n$, then $\boldsymbol{u}_{n+2}$ from $\boldsymbol{u}_{n+1}$ etc. The following process will be different. Assume that $\boldsymbol{u}_n$ is already computed or given as an initial state. Simultaneously, we shall solve for $\boldsymbol{u}_{n+1}, \boldsymbol{u}_{n+2}, \ldots, \boldsymbol{u}_{n+k}$ in one step of the algorithm."

$$\underbrace{(I - \Delta t L)}_{A}\boldsymbol{u}_{n+1} = \underbrace{\boldsymbol{u}_n + \Delta t \boldsymbol{f}_{n+1}}_{\boldsymbol{b}}.$$

$A = L + D + U$, $D := \mathrm{diag}(A)$, damped Jacobi for $k = 0, 1, \ldots, \nu$

$$\begin{aligned}
\boldsymbol{u}_{n+1}^{k+1} &= \boldsymbol{u}_{n+1}^{k} + \alpha D^{-1}(\boldsymbol{b} - A\boldsymbol{u}_{n+1}^{k}) \\
&= \boldsymbol{u}_{n+1}^{k} + \frac{\alpha}{1 + \frac{2\Delta t}{\Delta x^2}}(\boldsymbol{u}_n^{\nu} + \Delta t \boldsymbol{f}_{n+1} - (I - \Delta t L)\boldsymbol{u}_{n+1}^{k})
\end{aligned}$$

Hackbusch's Idea for Coarsening

Now use this sequential Jacobi procedure in time as a smoother, and use a coarse correction in space:

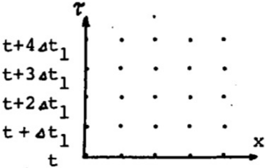


Fig 2.1a: Grid at level l

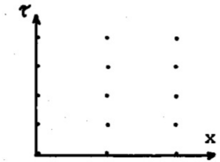


Fig 2.1b: Grid at level l-1

Hackbusch observes:

- Very fast multigrid convergence when coarsening in space
- Much less good convergence when coarsening in time as well

Hackbusch's Idea for Coarsening

Now use this sequential Jacobi procedure in time as a smoother, and use a coarse correction in space:

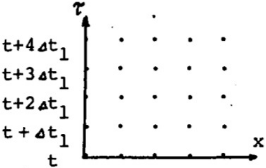


Fig 2.1a: Grid at level l

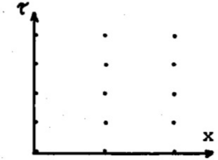


Fig 2.1b: Grid at level l-1

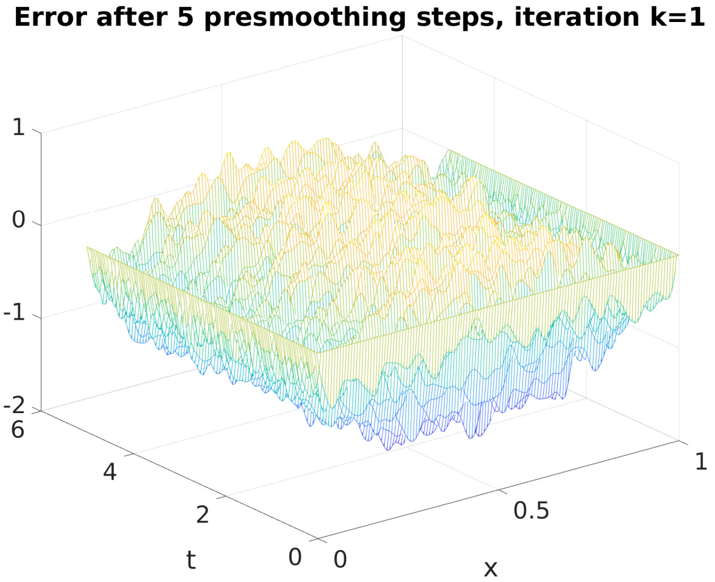
Hackbusch observes:

- Very fast multigrid convergence when coarsening in space
- Much less good convergence when coarsening in time as well

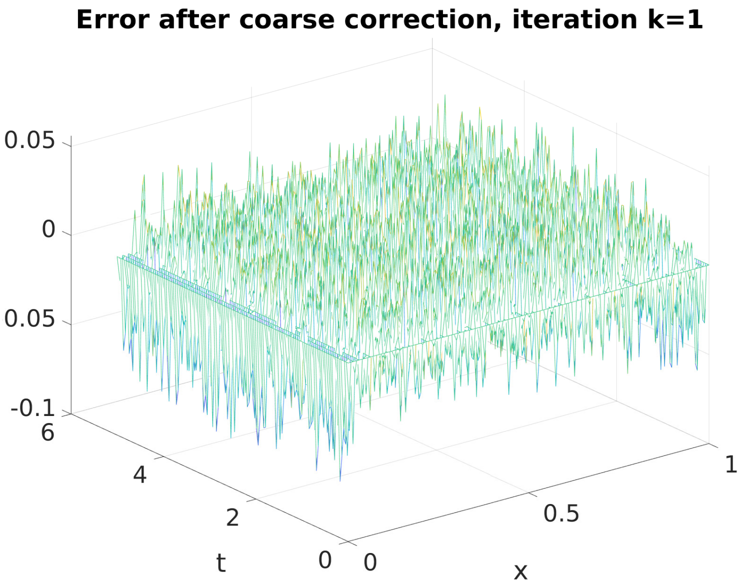
Lubich and Ostermann (1987): Multigrid WR

"For the case when the same step-size is used at all the nodes of a level, we regain the method proposed by Hackbusch to which our results will apply in particular."

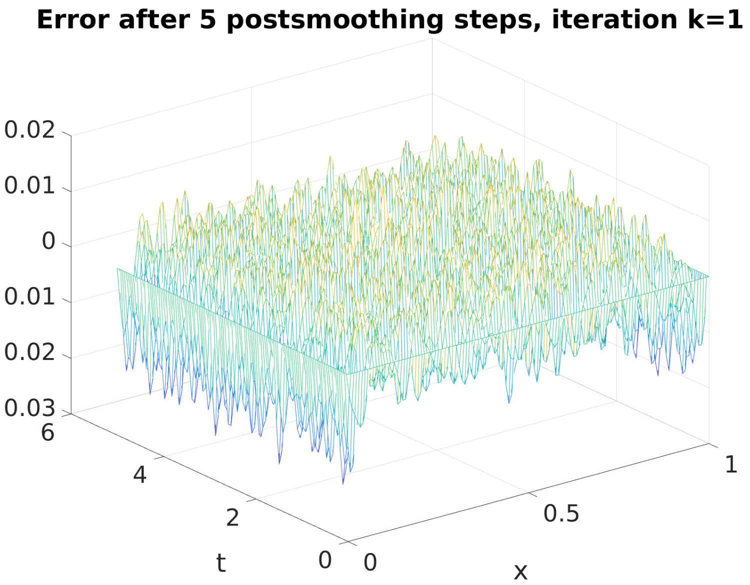
Parabolic Multigrid with Space Coarsening



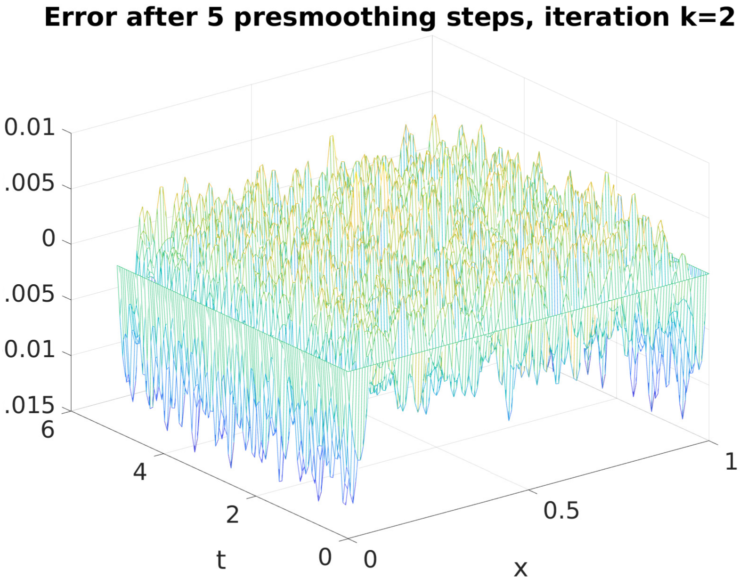
Parabolic Multigrid with Space Coarsening



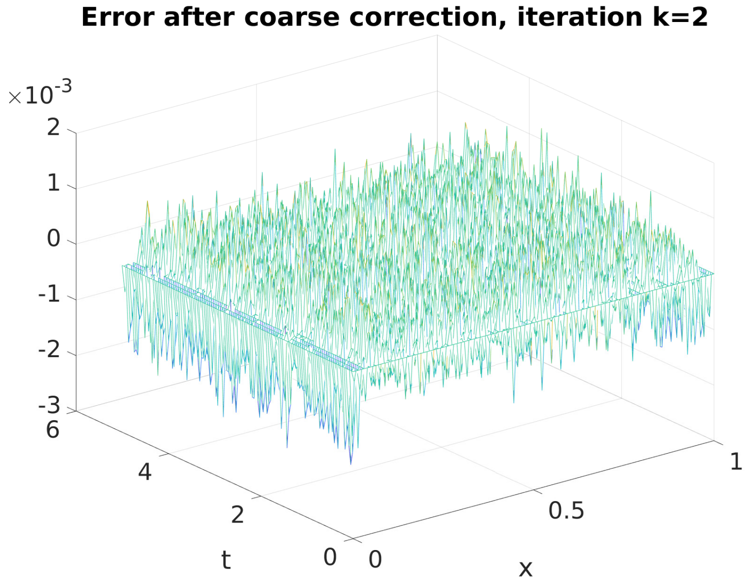
Parabolic Multigrid with Space Coarsening



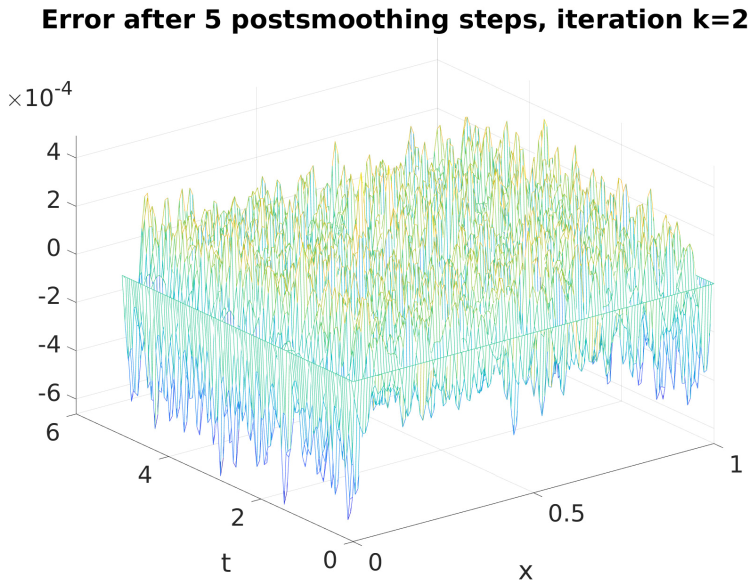
Parabolic Multigrid with Space Coarsening



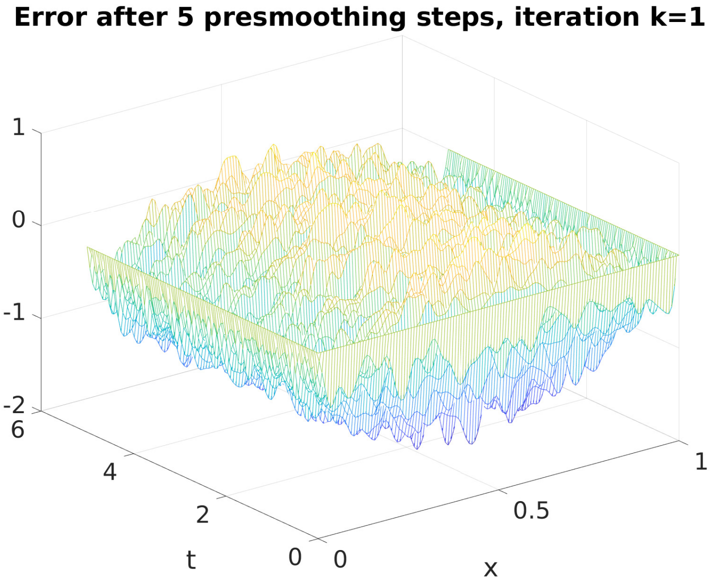
Parabolic Multigrid with Space Coarsening



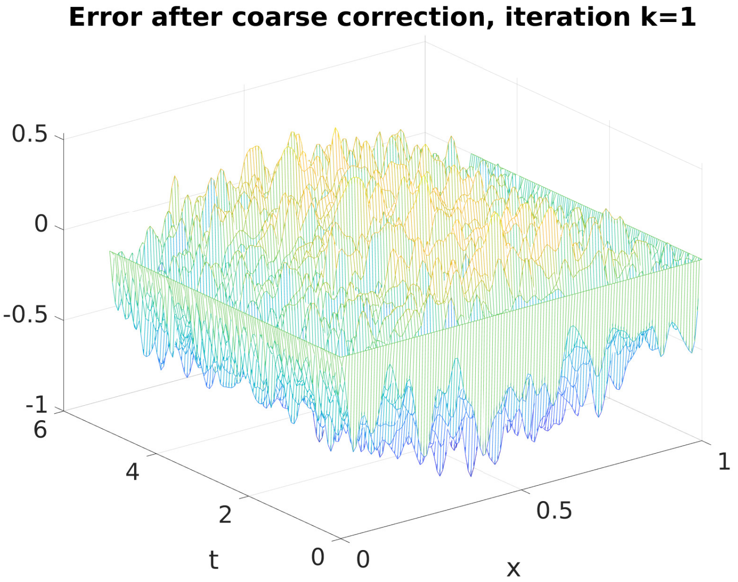
Parabolic Multigrid with Space Coarsening



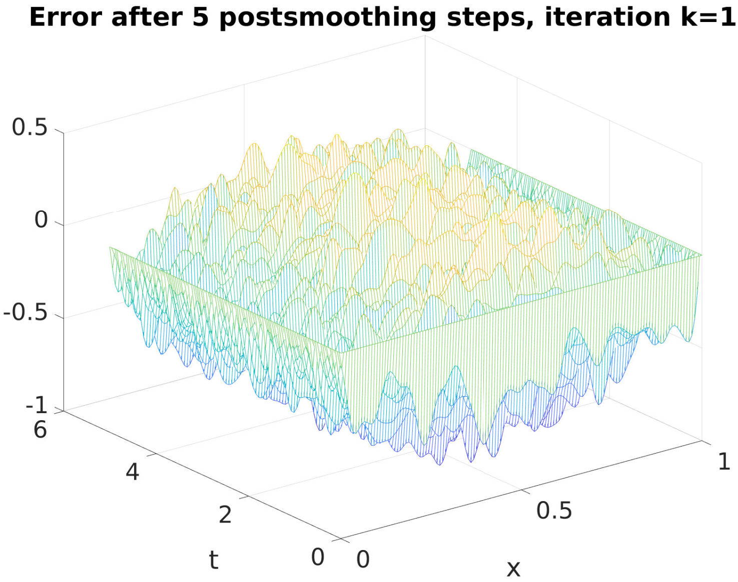
Parabolic Multigrid with Space-Time Coarsening



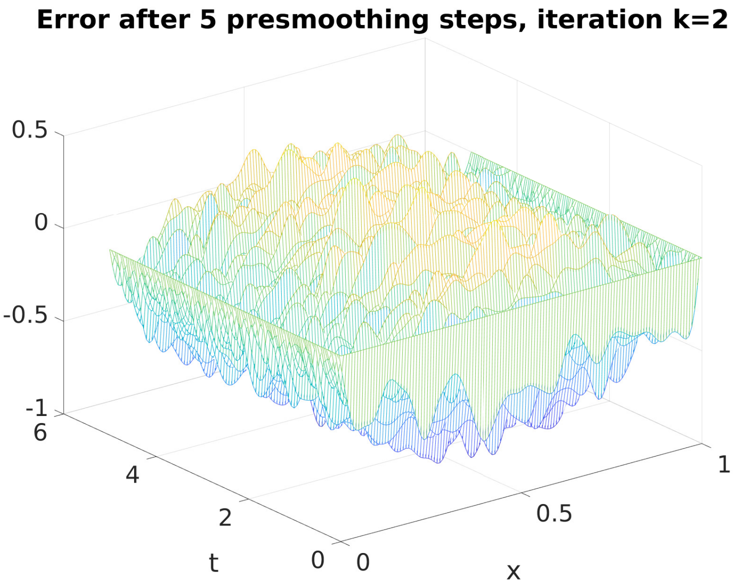
Parabolic Multigrid with Space-Time Coarsening



Parabolic Multigrid with Space-Time Coarsening

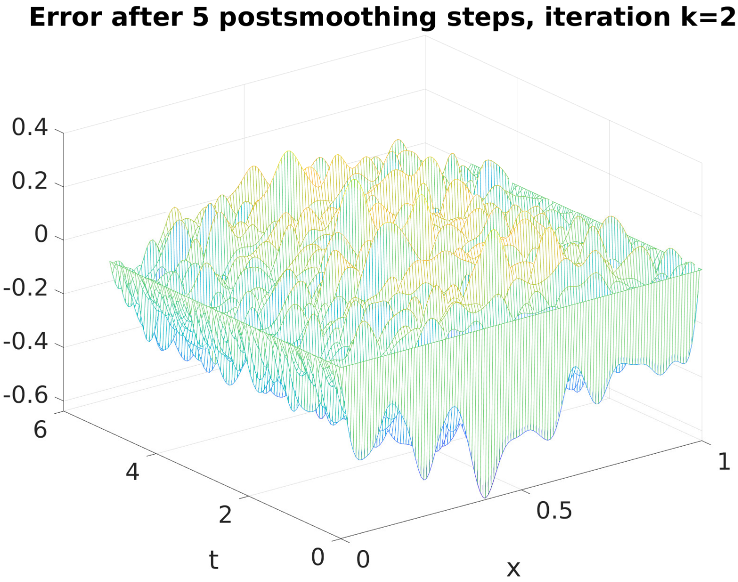


Parabolic Multigrid with Space-Time Coarsening

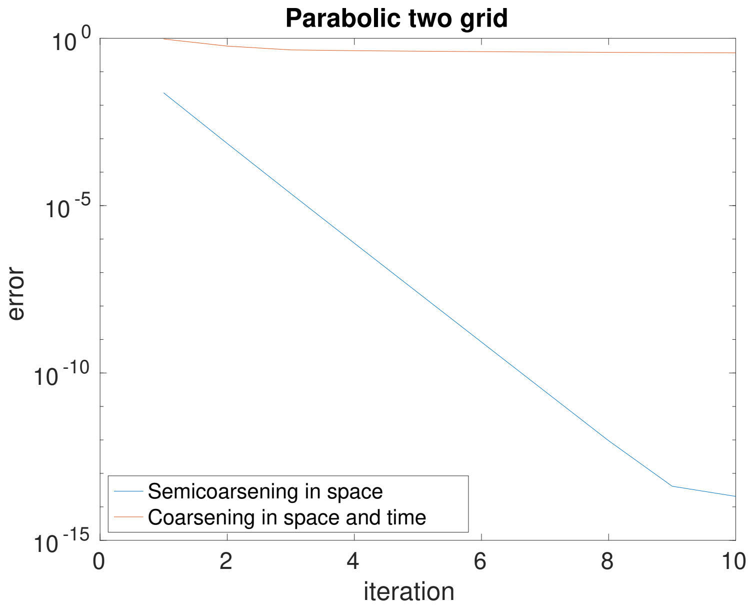


Parabolic Multigrid with Space-Time Coarsening

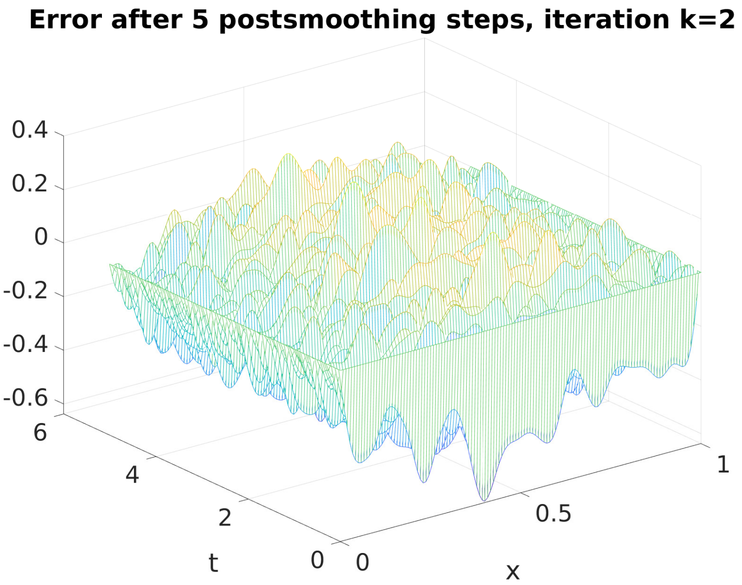
Parabolic Multigrid with Space-Time Coarsening



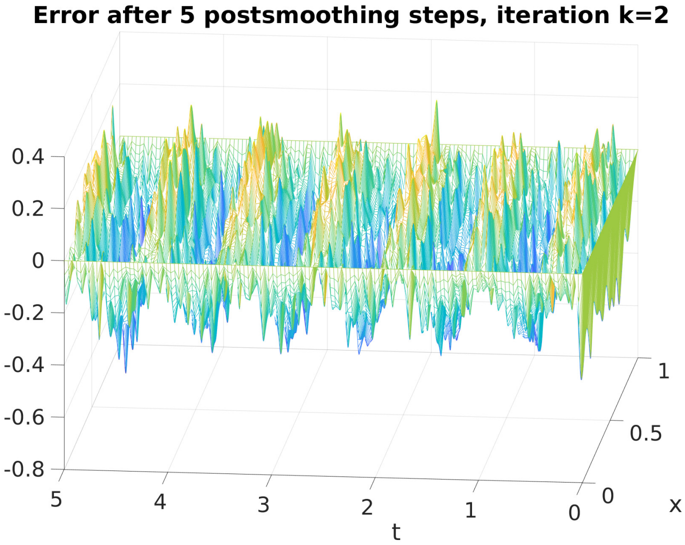
Convergence Comparison



What goes wrong with space-time coarsening?



There is no smoothing in time!



Early Remedies

Horton and Vandewalle (1995): A Space-Time Multigrid Method for Parabolic Partial Differential Equations

> *"The fully discrete PDE is a strongly anisotropic problem. Pointwise smoothing combined with standard coarsening is a notoriously slow procedure for such problems."*

Proposed Remedies:

1. Adaptive semi-coarsening in space or time depending on the anisotropy
2. Prolongation operators only forward in time

"Numerical results [...] for the one- and two-dimensional heat equations for both first- and second-order discretizations of the time derivative [...] proved to converge quickly, although at present the F-cycle seems to be necessary to achieve grid-independent rates."

Time Multigrid for Dahlquist's Equation

$$\partial_t u = \lambda u, \quad u(0) = 0, \quad \lambda \in \mathbb{C}$$

Applying Backward Euler in time, we obtain

$$\frac{u_{n+1} - u_n}{\Delta t} = \lambda u_{n+1} \quad \Longleftrightarrow \quad (1 - \lambda \Delta t) u_{n+1} - u_n = 0.$$

Writing these equations simultaneously for many time steps leads to the linear system

$$\underbrace{\begin{bmatrix} (1-\lambda\Delta t) & & & \\ -1 & (1-\lambda\Delta t) & & \\ & -1 & \ddots & \\ & & \ddots & \ddots \end{bmatrix}}_{A} \underbrace{\begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \end{pmatrix}}_{\boldsymbol{u}} = \underbrace{\begin{pmatrix} f_1 \\ f_2 \\ f_3 \\ \vdots \end{pmatrix}}_{\boldsymbol{f}}.$$

Using a Jacobi smoother for this linear system $A\boldsymbol{u} = \boldsymbol{f}$ with damping parameter α gives

$$\boldsymbol{u}^{k+1} = \boldsymbol{u}^k + \alpha D^{-1}(\boldsymbol{f} - A\boldsymbol{u}^k) = \boldsymbol{u}^k - \frac{\alpha}{1 - \lambda \Delta t} A \boldsymbol{u}^k.$$

Local Fouriermode Analysis (LFA)

Insert a Fourier mode in time,

$$u_n^k := C_\omega^k e^{i\omega n\Delta t},$$

into the Jacobi smoother

$$\boldsymbol{u}_n^{k+1} = \boldsymbol{u}_n^k - \frac{\alpha}{1-\lambda\Delta t}((1-\lambda\Delta t)\boldsymbol{u}_n^k - \boldsymbol{u}_{n-1}^k),$$

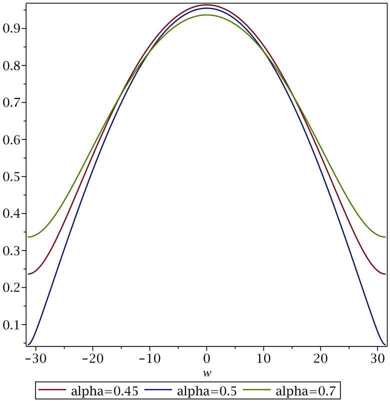
on one line (LFA does not see initial conditions)

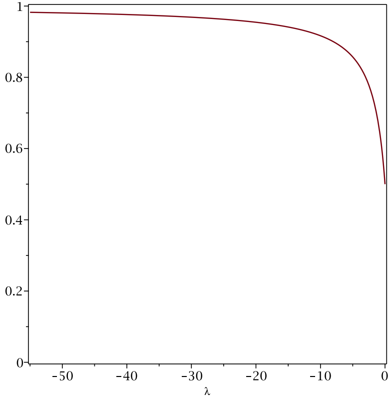
$$\begin{aligned} C_\omega^{k+1} &= C_\omega^k - \tfrac{\alpha}{1-\lambda\Delta t}\left((1-\lambda\Delta t)C_\omega^k - C_\omega^k e^{-i\omega\Delta t}\right) \\ &= \left(1-\alpha+\tfrac{\alpha e^{-i\omega\Delta t}}{1-\lambda\Delta t}\right) C_\omega^k. \end{aligned}$$

The convergence factor is thus

$$\rho(\omega,\alpha) = \left(1-\alpha+\frac{\alpha e^{-i\omega\Delta t}}{1-\lambda\Delta t}\right), \quad \omega\Delta t \in (-\pi,\pi).$$

Smoothing Properties of Jacobi in time



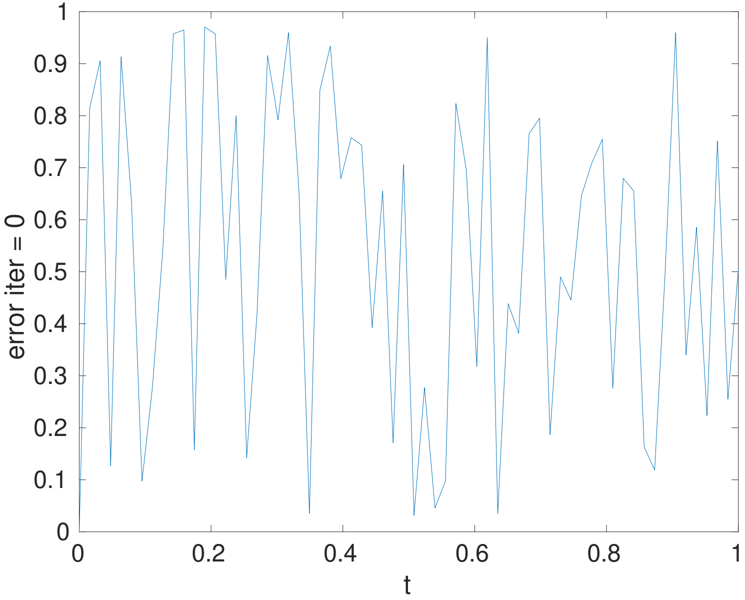


Left: Smoothing properties for the Jacobi smoother applied to the Dahlquist equation.

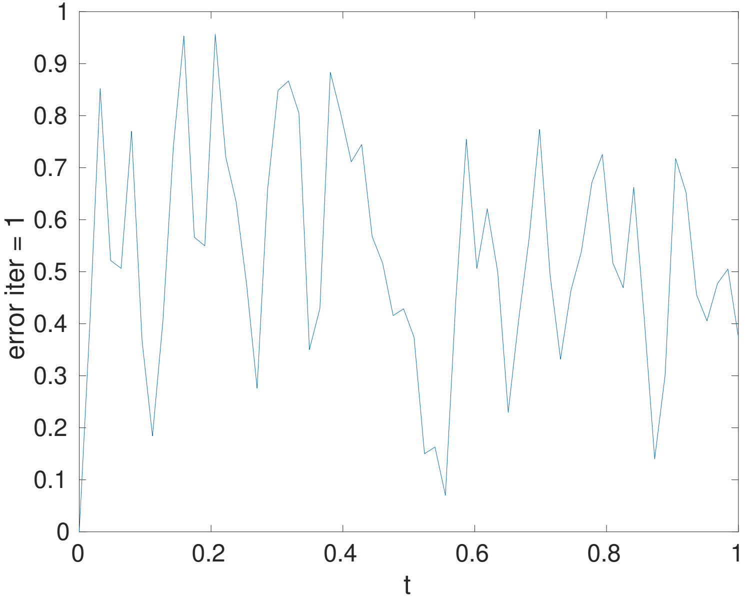
Right: $\alpha^* = \frac{\Delta t^2\lambda^2 - 3\Delta t\lambda + 2}{\Delta t^2\lambda^2 - 4\Delta t\lambda + 4}$ for best smoothing properties.

Damped Jacobi is a good smoother in time !?

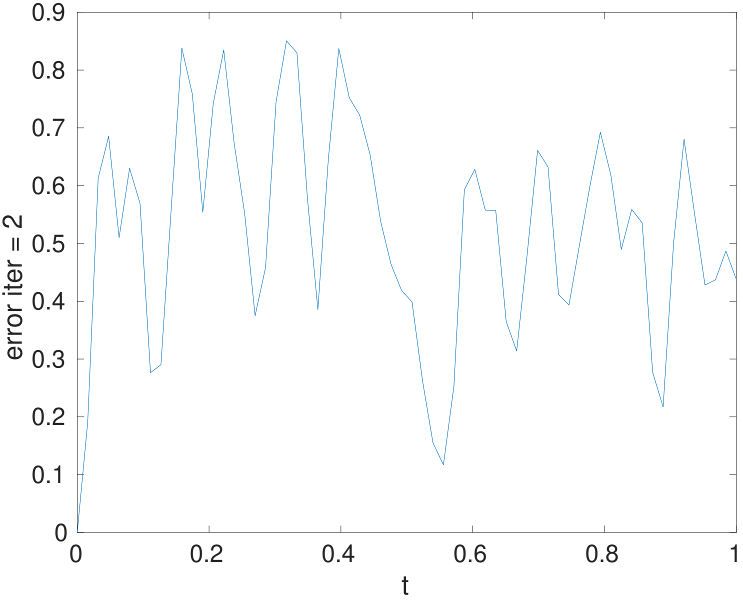
Jacobi Smoother for Dahlquist's Equation: $k = 0$



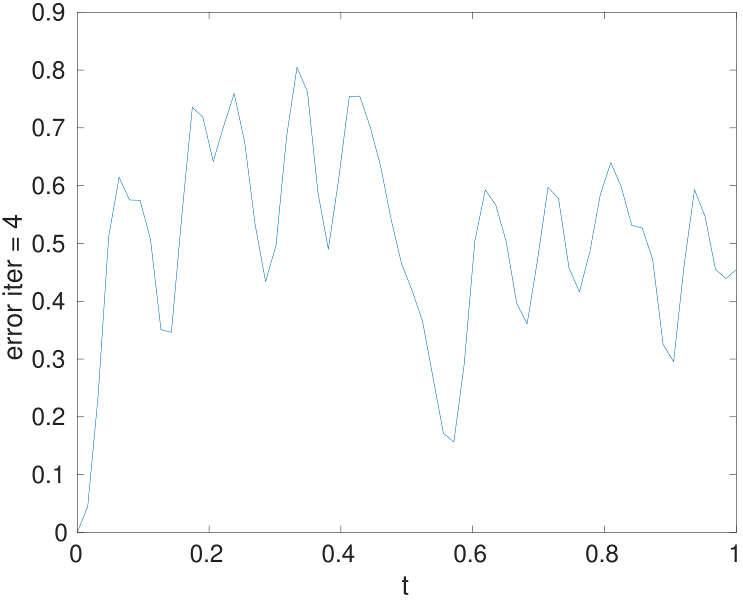
Jacobi Smoother for Dahlquist's Equation: $k = 1$



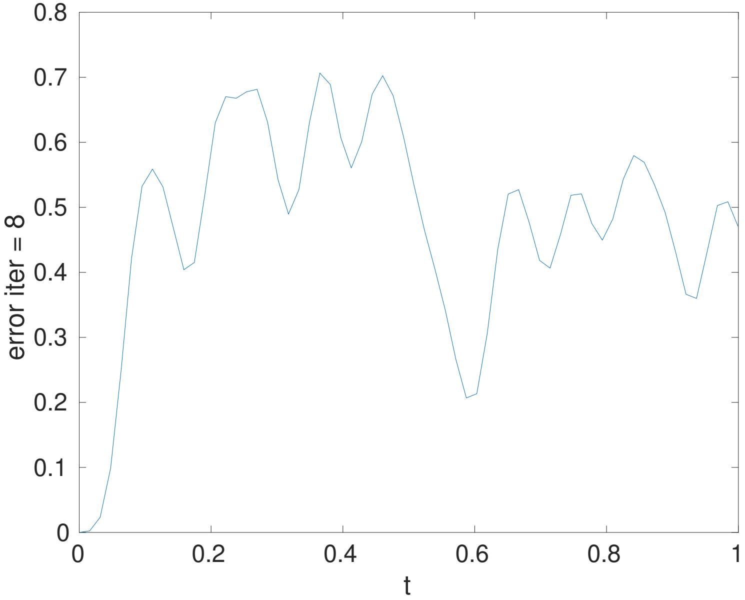
Jacobi Smoother for Dahlquist's Equation: $k = 2$



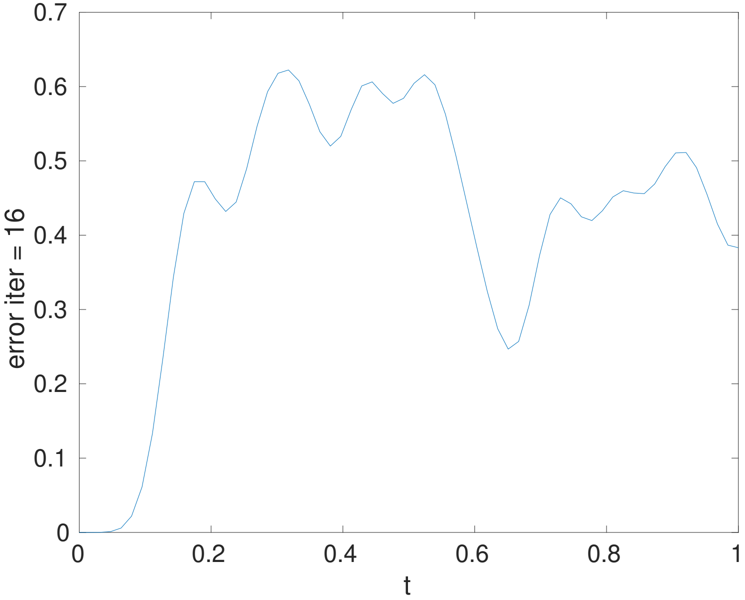
Jacobi Smoother for Dahlquist's Equation: $k = 4$



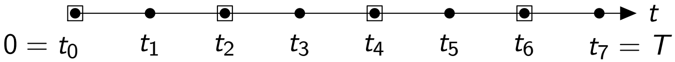
Jacobi Smoother for Dahlquist's Equation: $k = 8$



Jacobi Smoother for Dahlquist's Equation: $k = 16$



Two Grid Method in Time



$$
\begin{aligned}
\boldsymbol{u}^{k+\frac{1}{3}} &= S(\boldsymbol{f},\boldsymbol{u}^k,\nu_1)\,; && \text{\% presmoothing}\\
\boldsymbol{u}^{k+\frac{2}{3}} &= \boldsymbol{u}^{k+\frac{1}{3}} + PA_c^{-1}R(\boldsymbol{f}-A\boldsymbol{u}^{k+\frac{1}{3}}) && \text{\% coarse correction}\\
\boldsymbol{u}^{k+1} &= S(\boldsymbol{f},\boldsymbol{u}^{k+\frac{2}{3}},\nu_2)\,; && \text{\% postsmoothing}
\end{aligned}
$$

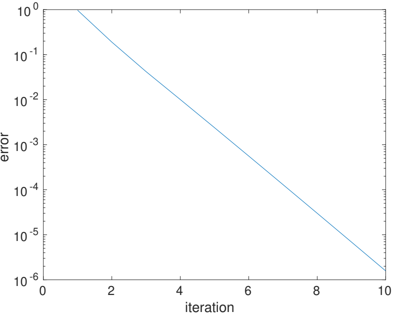
$$
P = \begin{bmatrix} \frac{1}{2} & & \\ 1 & & \\ \frac{1}{2} & \frac{1}{2} & \\ & 1 & \\ & \frac{1}{2} & \frac{1}{2} \\ & & 1 \\ & & \frac{1}{2} \end{bmatrix}
$$

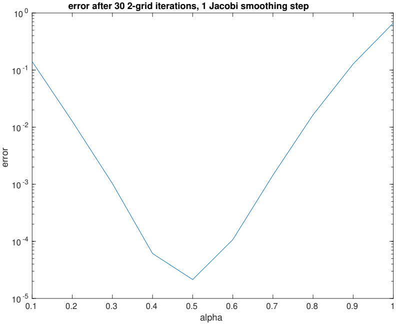
$R = \frac{1}{2}P^T$ (full weighting) or injection

$$
R = \begin{bmatrix} 0 & 1 & 0 & & & & \\ & & 0 & 1 & 0 & & \\ & & & & 0 & 1 & 0 \end{bmatrix}
$$

$$
A_c = \begin{bmatrix} 1-\lambda 2\Delta t & & \\ -1 & 1-\lambda 2\Delta t & \\ & -1 & 1-\lambda 2\Delta t \end{bmatrix} \text{ or } A_c = RAP
$$

Convergence, and best choice of α





Left: error decay for the two grid method applied to the Dahlquist equation.

Right: dependence on the choice of the relaxation parameter α of the error after $k = 30$ iterations

Space-Time Multigrid

All at once system for the heat equation in space-time:

$$\underbrace{\begin{bmatrix} (I - L\Delta t) & & & \\ -I & (I - L\Delta t) & & \\ & -I & \ddots & \\ & & \ddots & \ddots \end{bmatrix}}_{A} \underbrace{\begin{pmatrix} \boldsymbol{u}_1 \\ \boldsymbol{u}_2 \\ \boldsymbol{u}_3 \\ \vdots \end{pmatrix}}_{\boldsymbol{u}} = \underbrace{\begin{pmatrix} \boldsymbol{f}_1 \\ \boldsymbol{f}_2 \\ \boldsymbol{f}_3 \\ \vdots \end{pmatrix}}_{\boldsymbol{f}},$$

Key idea: need to divide by the diagonal block $I - L\Delta t$, as we divided by $1 - \lambda\Delta t$ in Jacobi for the Dahlquist equation, i.e. use a block Jacobi smoother.

Insert a Fourier mode in space and time

$$u_{n,j}^k := C_{\omega,\xi}^k e^{i\omega n\Delta t} e^{i\xi j\Delta x}$$

into the block Jacobi smoother

$$\begin{aligned} \boldsymbol{u}_n^{k+1} &= \boldsymbol{u}_n^k - \alpha(I - L\Delta t)^{-1}((I - \lambda L\Delta t)\boldsymbol{u}_n^k - \boldsymbol{u}_{n-1}^k) \\ &= (1-\alpha)\boldsymbol{u}_n^k + \alpha(I - L\Delta t)^{-1}\boldsymbol{u}_{n-1}^k \end{aligned}$$

Smoothing Analysis

For the term Lu^k_{n-1} we get with $u^k_{n,j} := C^k_{\omega,\xi} e^{i\omega n\Delta t} e^{i\xi j\Delta x}$

$$
\begin{aligned}
& \frac{1}{\Delta x^2}(u_{n-1,j+1} - 2u_{n-1,j} + u_{n-1,j-1}) \\
= \quad & e^{-i\omega\Delta t} \frac{1}{\Delta x^2} \left(e^{i\xi\Delta x} - 2 + e^{-i\xi\Delta x} \right) C^k_{\omega,\xi} e^{i\omega n\Delta t} e^{i\xi j\Delta x} \\
= \quad & e^{-i\omega\Delta t} \frac{2(\cos\xi\Delta x - 1)}{\Delta x^2} C^k_{\omega,\xi} e^{i\omega n\Delta t} e^{i\xi j\Delta x}.
\end{aligned}
$$

The symbol of the block Jacobi smoother

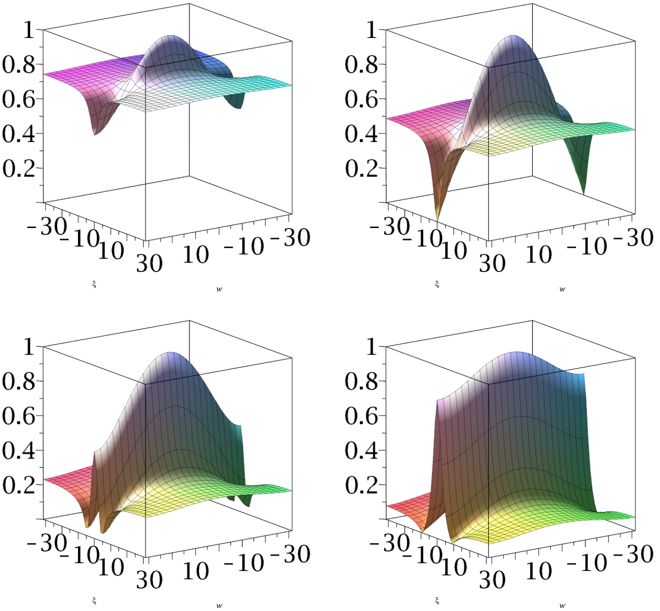
$$(1-\alpha)\boldsymbol{u}^k_n + \alpha(I - L\Delta t)^{-1}\boldsymbol{u}^k_{n-1}$$

is thus for $\omega\Delta t \in (-\pi, \pi)$, $\xi\Delta x \in (-\pi, \pi)$

$$\rho(\omega, \xi, \alpha) = 1 - \alpha \left(1 - \frac{e^{i\omega\Delta t}}{1 + 2\frac{\Delta t}{\Delta x^2}(1 - \cos\xi\Delta x)} \right)$$

G, Neumüller (2016): Analysis of a New Space-Time Parallel Multigrid Algorithm for Parabolic Problems

Fourier Local Mode Analysis Results



$|\rho|$ for $\Delta t = \Delta x = 0.1$ and $\alpha = 0.25, 0.5, 0.75, 0.9$

Time and Space Smoothing Lemmas

Lemma (Optimal parameter for smoothing in time)

The best choice for α to obtain smoothing in time is

$$\alpha^* = \frac{1}{2}.$$

Then all high frequencies in time, $\omega \in \pm(\frac{\pi}{2\Delta t}, \frac{\pi}{\Delta t})$ are multiplied by at least the factor $\frac{1}{\sqrt{2}}$.

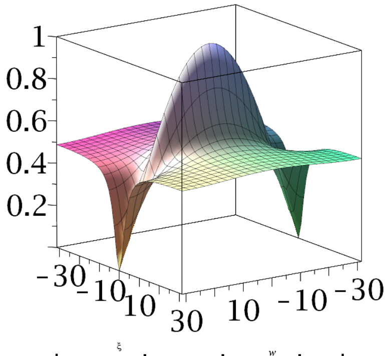
Proof. The derivative w.r.t ω,

$$\partial_\omega |\rho(\omega, \xi, \alpha)|^2 = -2\alpha(1-\alpha)\frac{\Delta x^2 \Delta t \sin(\omega \Delta t)}{2\Delta t(1-\cos(\xi\Delta x)) + \Delta x^2}$$

is negative for positive ω, and positive for negative ω. Thus the maximum for $\omega \in (\frac{\pi}{2\Delta t}, \frac{\pi}{\Delta t})$ is attained at $\omega = \frac{\pi}{2\Delta t}$, and similarly for negative ω at $\omega = -\frac{\pi}{2\Delta t}$.
The derivative with respect to ξ at $\omega = \frac{\pi}{2\Delta t}$ is

$$\partial_\xi |\rho(\frac{\pi}{2\Delta t}, \xi, \alpha)|^2 = -\frac{4\alpha^2 \Delta x^5 \Delta t \sin(\xi \Delta x)}{(2\Delta t(1-\cos(\xi\Delta x)) + \Delta x^2)^3},$$

Proof continued



which shows that the maximum is attained at $\xi = 0$. The worst smoothing in time is thus at $(\omega, \xi) = (\pm\frac{\pi}{2\Delta t}, 0)$, and the convergence factor value in modulus at this location is

$$|\rho(\pm\frac{\pi}{2\Delta t}, 0, \alpha)|^2 = (1-\alpha)^2\alpha^2,$$

and this value is minimized for $\alpha = \alpha^* = \frac{1}{2}$, for which $|\rho(\pm\frac{\pi}{2\Delta t}, 0, \frac{1}{2})| = \frac{1}{\sqrt{2}}$.

Lemma (Condition for smoothing in space)

With $\alpha = \alpha^* = \frac{1}{2}$*, high frequencies in space* $\xi \in \pm(\frac{\pi}{2\Delta x}, \frac{\pi}{\Delta x})$ *are at least damped by the factor* $\frac{1}{\sqrt{2}}$ *if*

$$\mu := \frac{\Delta t}{\Delta x^2} \geq \frac{1}{\sqrt{2}}.$$

Proof.

By the derivatives from the previous Lemma, the least damping is at $\omega = 0$ and $\xi = \pm\frac{\pi}{2\Delta x}$, namely

$$|\rho(0, \frac{\pi}{2\Delta x}, \alpha^*)| = \frac{\mu + 1}{2\mu + 1}$$

and

$$\frac{\mu+1}{2\mu+1} \leq \frac{1}{\sqrt{2}} \iff \sqrt{2}(\mu+1) \leq 2\mu + 1$$

$$\iff \sqrt{2} - 1 \leq (2 - \sqrt{2})\mu \iff \mu \geq \frac{1}{\sqrt{2}}$$

Theorem (Space-time multigrid coarsening)

In the Space-Time Multi-Grid (STMG) method with block Jacobi smoother applied to the all at once space time system of the one dimensional heat equation discretized by Backward Euler in time and centered finite differences in space, and the best choice of the relaxation parameter for time smoothing $\alpha^* = \frac{1}{2}$, one can always perform coarsening in time, and in space one can also use coarsening provided the condition $\frac{\Delta t}{\Delta x^2} \geq \frac{1}{\sqrt{2}}$ holds on the current level.

Proof.

For the two level method, this is a direct consequence of the two Lemmas, and the extension to the multilevel case follows by the fact that for the multigrid method, the two grid correction is simply applied recursively, with the bounds not depending on the levels. □

Multigrid Iterations 3D Heat Equation

One V-cycle in space to invert the diagonal blocks

space levels	time levels 1	2	3	4	5	5	7	8	9	10
0	7	8	8	8	7	7	7	8	8	8
1	7	8	8	8	7	7	7	8	8	8
2	8	8	8	8	8	7	8	8	8	8
3	8	9	8	8	8	8	8	8	8	8
4	10	9	9	9	8	8	8	8	8	8
5	10	10	10	9	9	8	8	8	8	8

Solution times in seconds:

dof	forward substitution	multigrid
2 304	3.30	0.06
23 296	3.69	1.02
218 880	9.80	13.19
1 912 576	95.27	136.99
16 015 104	1031.43	1155.12
131 120 896	9970.89	10416.90

3D Heat Equation Weak Scaling Results

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

Vulcan BlueGene/Q Supercomputer in Livermore (by M. Neumüller)

3D Heat Equation Strong Scaling Results

cores	time steps	dof	iter	time
1	512	15 300 608	7	7 635.2
2	512	15 300 608	7	3 821.7
4	512	15 300 608	7	1 909.9
8	512	15 300 608	7	954.2
16	512	15 300 608	7	477.2
32	512	15 300 608	7	238.9
64	512	15 300 608	7	119.5
128	512	15 300 608	7	59.7
256	512	15 300 608	7	30.0
512	524 288	15 667 822 592	7	15 205.9
1 024	524 288	15 667 822 592	7	7 651.5
2 048	524 288	15 667 822 592	7	3 825.3
4 096	524 288	15 667 822 592	7	1 913.4
8 192	524 288	15 667 822 592	7	956.6
16 384	524 288	15 667 822 592	7	478.1
32 768	524 288	15 667 822 592	7	239.3
65 536	524 288	15 667 822 592	7	119.6
131 072	524 288	15 667 822 592	7	59.8
262 144	524 288	15 667 822 592	7	30.0

Vulcan BlueGene/Q Supercomputer in Livermore (by M. Neumüller)

Parareal as a Geometric Multigrid Method

Consider the Dahlquist equation

$$\frac{du}{dt} = \lambda u \quad \text{in } (0,T), \quad u(0) = u_0$$

N coarse time intervals and M fine time steps in every coarse time interval, $0 = t_0 < t_1 < t_2 < \ldots < t_{MN} = T$, $t_m - t_{m-1} = \Delta t$, forward Euler

$$A\boldsymbol{u} := \begin{pmatrix} 1 & & & \\ -\phi & 1 & & \\ & \ddots & \ddots & \\ & & -\phi & 1 \end{pmatrix} \begin{pmatrix} u_0 \\ u_1 \\ \vdots \\ u_{MN} \end{pmatrix} = \begin{pmatrix} u_0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} =: \boldsymbol{f},$$

where $\phi := 1 + \lambda\Delta t$. Eliminate every second unknown

$$\begin{pmatrix} 1 & & & \\ -\phi^2 & 1 & & \\ & \ddots & \ddots & \\ & & -\phi^2 & 1 \end{pmatrix} \begin{pmatrix} u_0 \\ u_2 \\ \vdots \\ u_{MN} \end{pmatrix} = \begin{pmatrix} u_0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$

Eliminate more unknowns, keeping only every M-th one,

$$\tilde{A}\tilde{\boldsymbol{U}} := \begin{pmatrix} 1 & & & \\ -\tilde{F} & 1 & & \\ & \ddots & \ddots & \\ & & -\tilde{F} & 1 \end{pmatrix} \begin{pmatrix} \tilde{U}_0 \\ \tilde{U}_1 \\ \vdots \\ \tilde{U}_N \end{pmatrix} = \begin{pmatrix} u_0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} =: \tilde{\boldsymbol{f}},$$

where $\tilde{F} := (1 + \lambda\Delta t)^M$. Note that at the coarse nodes $T_n := nM\Delta t$ we have $\tilde{U}_n = u_{nM}$. Now approximate

$$\tilde{A} := \begin{pmatrix} 1 & & & \\ -\tilde{F} & 1 & & \\ & \ddots & \ddots & \\ & & -\tilde{F} & 1 \end{pmatrix} \approx \begin{pmatrix} 1 & & & \\ -\tilde{G} & 1 & & \\ & \ddots & \ddots & \\ & & -\tilde{G} & 1 \end{pmatrix} =: \tilde{M}$$

with $\tilde{G} := 1 + \lambda\Delta T$ one forward Euler step, and consider the preconditioned stationary iteration

$$\tilde{\boldsymbol{U}}^{k+1} = \tilde{\boldsymbol{U}}^k + \tilde{M}^{-1}(\boldsymbol{f} - \tilde{A}\tilde{\boldsymbol{U}}^k).$$

Theorem

The stationary iteration is equal to Parareal, $\tilde{U}_n^k = U_n^k$ for $k = 1, 2, \ldots$ and $n = 0, 1, \ldots, N$, provided initially we have $\tilde{U}_n^0 = U_n^0$ for $n = 0, 1, \ldots, N$.

Proof. The preconditioned stationary iteration computes

$$\begin{pmatrix} 1 & & & \\ -\tilde{G} & 1 & & \\ & \ddots & \ddots & \\ & & -\tilde{G} & 1 \end{pmatrix} \left(\begin{pmatrix} \tilde{U}_0^{k+1} \\ \tilde{U}_1^{k+1} \\ \vdots \\ \tilde{U}_N^{k+1} \end{pmatrix} - \begin{pmatrix} \tilde{U}_0^{k} \\ \tilde{U}_1^{k} \\ \vdots \\ \tilde{U}_N^{k} \end{pmatrix} \right) = \begin{pmatrix} u_0 - \tilde{U}_0^k \\ \tilde{F}\tilde{U}_0^k - \tilde{U}_1^k \\ \vdots \\ \tilde{F}\tilde{U}_{N-1}^k - \tilde{U}_N^k \end{pmatrix}.$$

The n-th line in this iteration reads

Proof continued

$$-\tilde{G}\tilde{U}_{n-1}^{k+1} + \tilde{G}\tilde{U}_{n-1}^{k} + \tilde{U}_n^{k+1} - \tilde{U}_n^k = \tilde{F}\tilde{U}_{n-1}^k - \tilde{U}_n^k,$$

and we obtain after simplification

$$\tilde{U}_n^{k+1} = \tilde{F}\tilde{U}_{n-1}^k + \tilde{G}\tilde{U}_{n-1}^{k+1} - \tilde{G}\tilde{U}_{n-1}^k.$$

Applying parareal to the Dahlquist problem using forward Euler, we get for the parareal fine integrator

$$F(T_{n+1}, T_n, v) := (1 + \lambda\Delta t)^M v \equiv \tilde{F}v$$

and for the coarse integrator

$$G(T_{n+1}, T_n, v) := (1 + \lambda\Delta T)v \equiv \tilde{G}v,$$

and thus the updating formula coincides with Parareal.

Remark: This result also holds for any other integrator since we never used the precise form of Forward Euler, e.g. for Backward Euler $\phi = (1 - \lambda\Delta t)^{-1}$, $\tilde{F} = (1 - \lambda\Delta t)^{-M}$, $\tilde{G} = (1 - \lambda\Delta T)^{-1}$.

Parareal as a geometric multigrid method

For solving approximately the linear system

$$A\boldsymbol{u} = \boldsymbol{f},$$

a geometric two grid method would, starting with the initial guess $\boldsymbol{u}^0$, compute for $k = 0, 1, 2, \ldots$

$$\begin{aligned} \tilde{\boldsymbol{u}}^k &:= \text{Smooth}(A, \boldsymbol{f}, \boldsymbol{u}^k); \\ \boldsymbol{e} &:= A_c^{-1} R(\boldsymbol{f} - A\tilde{\boldsymbol{u}}^k); \\ \boldsymbol{u}^{k+1} &:= \tilde{\boldsymbol{u}}^k + P\boldsymbol{e}; \end{aligned}$$

To identify parareal with geometric multigrid, we need a block Jacobi splitting

$$A = \tilde{M}_J - \tilde{N}_J,$$

where $\tilde{M}_J$ is a block diagonal matrix with diagonal blocks of size $M \times M$, except for the first block which is one bigger because of the initial condition of the problem.

Theorem (Parareal=Multigrid+Agressive Coarsening)

Using one presmoothing step with the modified block Jacobi smoother

$$\boldsymbol{u}^{\ell} = \boldsymbol{u}^{\ell-1} + E\tilde{M}_J^{-1}(\boldsymbol{f} - A\boldsymbol{u}^{\ell-1}),$$

where E is the identity, except with zeros at the coarse nodes, injection for P and $R := P^T$, and $A_c = \tilde{M}$, then the two grid algorithm produces the parareal iterates at the coarse nodes, $U_n^k = u_{nM}^k$, provided one starts with an initial approximation $\boldsymbol{u}^0$ that satisfies $U_n^0 = u_{nM}^0$.

Proof. By induction on k: statement holds trivially for $k = 0$ by assumption.

We thus assume that for k, $u_{nM}^k = U_n^k$ for $n = 0, 1, \ldots, N$, and prove this for $k+1$. After one step with the modified block Jacobi smoother, $\tilde{\boldsymbol{u}}^k$ contains the fine solutions starting at each initial condition U_n^k,

$$\tilde{u}_{nM+j}^k = (1 + \lambda\Delta t)^j U_n^k, \quad j = 0, 1, \ldots, M-1,$$

except for the coarse variables which have not changed because of E, we still have $\tilde{u}^k_{nM} = U^k_n$ for $n = 0, 1, \ldots, N$.

Now with the coarse operator $A_c = \tilde{M}$ we need to solve

$$\tilde{M}\boldsymbol{e} = R(\boldsymbol{f} - A\tilde{\boldsymbol{u}}^k).$$

Looking at any line $n > 0$, with R the transpose of injection, the definition of A and $\tilde{M}$, and that $\boldsymbol{f}$ is zero except in the first component

$$\boldsymbol{e}_n - G(\boldsymbol{e}_{n-1}) = -U^k_n + (1 + \lambda\Delta t)(\tilde{u}^k_{nM-1}).$$

Now from the Jacobi smoother, we have

$$\tilde{u}^k_{nM-1} = (1 + \lambda\Delta t)^{M-1} u^k_{(n-1)M} = (1 + \lambda\Delta t)^{M-1} U^k_{n-1},$$

and thus with the definition of F

$$\boldsymbol{e}_n = -U^k_n + F(U^k_{n-1}) + G(\boldsymbol{e}_{n-1}),$$

Now performing the last of the 3 multigrid steps, with P injection, and $\tilde{u}^k_{nM} = U^k_n$, we get on the lines nM

$$u^{k+1}_{nM} = F(U^k_{n-1}) + G(\boldsymbol{e}_{n-1}). \qquad (*)$$

We next note that at step $n-1$ can be written as

$$\boldsymbol{e}_{n-1} = -U_{n-1}^{k} + F(U_{n-2}^{k}) + G(\boldsymbol{e}_{n-2}) = -U_{n-1}^{k} + u_{(n-1)M}^{k+1},$$

where we used (*) for the last step, and inserting this into (*) and using linearity gives

$$u_{nM}^{k+1} = F(U_{n-1}^{k}) + G(u_{(n-1)M}^{k+1}) - G(U_{n-1}^{k}),$$

which concludes the proof by induction, since this is the recurrence formula for the parareal algorithm, and $u_0^{k+1} = U_0^{k+1} = u_0$.

Remarks:

- This Theorem also holds in the non-linear context [G, Vandewalle 2007]
- The special block Jacobi smoother is only modifying the fine nodes, and is thus not convergent, and the second iteration will not produce any modification.
- Multilevel Parareal by recursion on $\tilde{M}$

Algebraic Multigrid (AMG)

Ruge, Stüben (1987): Algebraic multigrid

> *"Thus, the set of variables on level h can be split into two disjoint subsets: the first one contains the variables also represented in the coarser level (C-variables), and the second one is just the complementary set (F-variables)."*

Reordering the system matrix accordingly yields

$$A\boldsymbol{u} = \begin{bmatrix} A_{\mathrm{ff}} & A_{\mathrm{fc}} \\ A_{\mathrm{cf}} & A_{\mathrm{cc}} \end{bmatrix} \begin{pmatrix} \boldsymbol{u}_f \\ \boldsymbol{u}_c \end{pmatrix} = \begin{pmatrix} \boldsymbol{f}_f \\ \boldsymbol{f}_c \end{pmatrix} = \boldsymbol{f}.$$

A block-LU factorization is given by

$$\begin{bmatrix} A_{\mathrm{ff}} & A_{\mathrm{fc}} \\ A_{\mathrm{cf}} & A_{\mathrm{cc}} \end{bmatrix} = \begin{bmatrix} I & \\ A_{\mathrm{cf}} A_{\mathrm{ff}}^{-1} & I \end{bmatrix} \begin{bmatrix} A_{\mathrm{ff}} & \\ & S_{\mathrm{cc}} \end{bmatrix} \begin{bmatrix} I & A_{\mathrm{ff}}^{-1} A_{\mathrm{fc}} \\ & I \end{bmatrix},$$

with the Schur complement $S_{\mathrm{cc}} := A_{\mathrm{cc}} - A_{\mathrm{cf}} A_{\mathrm{ff}}^{-1} A_{\mathrm{fc}}$.

One can express then the inverse of A explicitly,

$$A^{-1} = \begin{bmatrix} I & -A_{\mathrm{ff}}^{-1} A_{\mathrm{fc}} \\ & I \end{bmatrix} \begin{bmatrix} A_{\mathrm{ff}}^{-1} & \\ & S_{\mathrm{cc}}^{-1} \end{bmatrix} \begin{bmatrix} I & \\ -A_{\mathrm{cf}} A_{\mathrm{ff}}^{-1} & I \end{bmatrix}.$$

Defining the coarse restriction and extension matrices by

$$R_c := [-A_{\mathrm{cf}} A_{\mathrm{ff}}^{-1} \quad I], \quad P_c := \begin{bmatrix} -A_{\mathrm{ff}}^{-1} A_{\mathrm{fc}} \\ I \end{bmatrix},$$

and the more simple fine restriction and extension matrices by

$$R_f := [I \quad 0], \quad P_f := R_f^T,$$

one can obtain the following surprising result

Lemma

The inverse A^{-1} of the reordered system matrix A can be expressed as a sum of an inverse just acting on the fine variables, and a complementary inverse,

$$A^{-1} = P_c (R_c A P_c)^{-1} R_c + P_f (R_f A P_f)^{-1} R_f.$$

Proof.

By a direct calculation: we first compute for the coarse nodes

$$
\begin{aligned}
R_c A P_c &= [-A_{\mathrm{cf}} A_{\mathrm{ff}}^{-1} \;\; I] \begin{bmatrix} A_{\mathrm{ff}} & A_{\mathrm{fc}} \\ A_{\mathrm{cf}} & A_{\mathrm{cc}} \end{bmatrix} \begin{bmatrix} -A_{\mathrm{ff}}^{-1} A_{\mathrm{fc}} \\ I \end{bmatrix} \\
&= [-A_{\mathrm{cf}} A_{\mathrm{ff}}^{-1} \;\; I] \begin{bmatrix} 0 \\ A_{\mathrm{cc}} - A_{\mathrm{cf}} A_{\mathrm{ff}}^{-1} A_{\mathrm{fc}} \end{bmatrix} = S_{\mathrm{cc}}.
\end{aligned}
$$

For the fine nodes, we get with the simple P_f and R_f

$$
R_f A P_f = A_{\mathrm{ff}}.
$$

We thus get

$$
\begin{aligned}
& P_c (R_c A P_c)^{-1} R_c + P_f (R_f A P_f)^{-1} R_f \\
&= \begin{bmatrix} -A_{\mathrm{ff}}^{-1} A_{\mathrm{fc}} \\ I \end{bmatrix} S_{\mathrm{cc}}^{-1} [-A_{\mathrm{cf}} A_{\mathrm{ff}}^{-1} \;\; I] + \begin{bmatrix} I \\ 0 \end{bmatrix} A_{\mathrm{ff}}^{-1} [I \quad 0] \\
&= \begin{bmatrix} I & -A_{\mathrm{ff}}^{-1} A_{\mathrm{fc}} \\ & I \end{bmatrix} \begin{bmatrix} A_{\mathrm{ff}}^{-1} & \\ & S_{\mathrm{cc}}^{-1} \end{bmatrix} \begin{bmatrix} I & \\ -A_{\mathrm{cf}} A_{\mathrm{ff}}^{-1} & I \end{bmatrix}.
\end{aligned}
$$

This result is interesting when we look at a classical stationary iterative method with preconditioner $M \approx A$,

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

The error $\boldsymbol{e}^k := \boldsymbol{u} - \boldsymbol{u}^k$ satisfies

$$\boldsymbol{e}^{k+1} = (I - M^{-1}A)\boldsymbol{e}^k.$$

Using for $M^{-1} = A^{-1}$, the error propagator $(I - M^{-1}A)$ vanishes identically, but writing it down explicitly gives

$$\begin{aligned} 0 &= (I - A^{-1}A) \\ &= I - P_c(R_cAP_c)^{-1}R_cA - P_f(R_fAP_f)^{-1}R_fA, \end{aligned}$$

which is an optimal additive correction scheme between fine and coarse nodes, it converges in one iteration (nilpotent). For a multiplicative correction scheme, we compute

$$\begin{aligned} &\left(I - P_c(R_cAP_c)^{-1}R_cA\right)\left(I - P_f(R_fAP_f)^{-1}R_fA\right) \\ &= I - P_c(R_cAP_c)^{-1}R_cA - P_f(R_fAP_f)^{-1}R_fA \\ &+P_c(R_cAP_c)^{-1}R_cAP_f(R_fAP_f)^{-1}R_fA, \end{aligned}$$

and the last term cancels, because the middle term

$$R_c A P_f = [-A_{\mathrm{cf}} A_{\mathrm{ff}}^{-1} \ \ I] \begin{bmatrix} A_{\mathrm{ff}} & A_{\mathrm{fc}} \\ A_{\mathrm{cf}} & A_{\mathrm{cc}} \end{bmatrix} \begin{bmatrix} I \\ 0 \end{bmatrix}$$
$$= [-A_{\mathrm{cf}} A_{\mathrm{ff}}^{-1} \ \ I] \begin{bmatrix} A_{\mathrm{ff}} \\ A_{\mathrm{cf}} \end{bmatrix} = 0.$$

Therefore, the multiplicative correction scheme in this exact setting coincides with the additive one.

AMG idea:

- approximate the operators

$$R_c = [-A_{\mathrm{cf}} A_{\mathrm{ff}}^{-1} \ \ I], \quad P_c := \begin{bmatrix} -A_{\mathrm{ff}}^{-1} A_{\mathrm{fc}} \\ I \end{bmatrix}$$

  in these exact correction schemes, i.e. A_{ff}^{-1}
- very different from geometric multigrid based on smoothing and coarse correction.

MGRIT (Multigrid Reduction in Time)

Friedhoff, Falgout, Kolev, MacLachlan, Schroder (2013): A multigrid-in-time algorithm for solving evolution equations in parallel

> *"Our algorithm is based on interpreting the parareal time integration method as a two level reduction scheme, and developing a multilevel algorithm from this viewpoint"*

Theorem

In the parareal algorithm, the error propagation operator is

$$(I - P_c\tilde{M}^{-1}R_cA)(I - P_f(R_fAP_f)^{-1}R_fA),$$

where $\tilde{M}$ is the coarse time stepping matrix.

The only approximation is therefore $\tilde{M} \approx R_cAP_c$ in the AMG setting of Parareal.

Proof.

The error propagation operator of Parareal on the coarse variables is

$$(I - \tilde{M}^{-1}\tilde{A}).$$

To write this for all variables, we need $R := [0 \;\; I]$ selecting the coarse nodes. Noting that

$$P_c = \begin{bmatrix} -A_{\mathrm{ff}}^{-1} A_{\mathrm{fc}} \\ I \end{bmatrix},$$

leaves coarse nodes invariant, and extends to fine by fine solves the error propagation operator of Parareal for all variables is

$$P_c(I - \tilde{M}^{-1}\tilde{A})R.$$

Now $\tilde{A} = R_c A P_c = S_{cc}$ is the Schur complement in the proof of the Lemma, since $\tilde{A}$ was obtained by elimination of the fine unknowns. Thus the error propagation operator becomes

$$\begin{aligned} P_c(I - \tilde{M}^{-1}\tilde{A})R &= P_c(I - \tilde{M}^{-1}R_c A P_c)R \\ &= (I - P_c\tilde{M}^{-1}R_c A)P_c R, \end{aligned}$$

Proof continued

and since

$$P_c R = \begin{bmatrix} -A_{\text{ff}}^{-1}A_{\text{fc}} \\ I \end{bmatrix} [0 \;\; I] = \begin{bmatrix} 0 & -A_{\text{ff}}^{-1}A_{\text{fc}} \\ 0 & I \end{bmatrix}$$

is identical to

$$\begin{aligned} I - P_f(R_f A P_f)^{-1} R_f A &= I - \begin{bmatrix} A_{\text{ff}}^{-1} \\ 0 \end{bmatrix} \begin{bmatrix} A_{\text{ff}} & A_{\text{fc}} \end{bmatrix} \\ &= \begin{bmatrix} 0 & -A_{\text{ff}}^{-1}A_{\text{fc}} \\ 0 & I \end{bmatrix}, \end{aligned}$$

the error propagation operator of parareal is indeed

$$(I - P_c \tilde{M}^{-1} R_c A)(I - P_f(R_f A P_f)^{-1} R_f A),$$

which concludes the proof.

Remark: Multilevel Parareal by recursion on $\tilde{M}$.

MGRIT and FCF relaxation

Idea of MGRIT: replace the F-relaxation, the second term in the error propagation operator, by

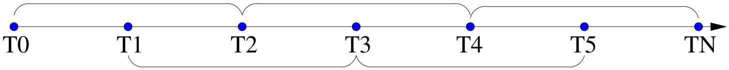
$$(I-P_f(R_fAP_f)^{-1}R_fA)(I-R^T(RAR^T)^{-1}RA)(I-P_f(R_fAP_f)^{-1}R_fA)$$

The C-relaxation term $(I-R^T(RAR^T)^{-1}RA)$ closes precisely the gap left by the F-relaxation to make a second F-relaxation useful (cf the E matrix in the geometric setting)

Theorem

The two level MGRIT algorithm with FCF-smoother computes the same iterations as the parareal algorithm with overlap of one coarse time interval,

$$\begin{aligned}
U_0^{k+1} &= u_0, \quad U_1^{k+1} = \tilde{F}u_0, \\
U_n^{k+1} &= \tilde{F}\tilde{F}U_{n-2}^k + \tilde{G}U_{n-1}^{k+1} - \tilde{G}\tilde{F}U_{n-2}^k.
\end{aligned}$$



Proof.

Need an interpretation of the added CF-relaxation in MGRIT in terms of the parareal algorithm. For the C-part, we obtain

$$I - R^T(RAR^T)^{-1}RA = I - \begin{bmatrix} 0 & 0 \\ 0 & A_{\rm cc}^{-1} \end{bmatrix} \begin{bmatrix} A_{\rm ff} & A_{\rm fc} \\ A_{\rm cf} & A_{\rm cc} \end{bmatrix}$$

$$= \begin{bmatrix} I & 0 \\ -A_{\rm cc}^{-1}A_{\rm cf} & 0 \end{bmatrix},$$

and multiplying with the F-part leads to

$$\begin{bmatrix} I & 0 \\ -A_{\rm cc}^{-1}A_{\rm cf} & 0 \end{bmatrix} \begin{bmatrix} 0 & -A_{\rm ff}^{-1}A_{\rm fc} \\ 0 & I \end{bmatrix} = \begin{bmatrix} 0 & -A_{\rm ff}^{-1}A_{\rm fc} \\ 0 & A_{\rm cc}^{-1}A_{\rm cf}A_{\rm ff}^{-1}A_{\rm fc} \end{bmatrix}.$$

Now this is multiplied from the left by the parareal error propagation operator $P_c(I - \tilde{M}^{-1}\tilde{A})R$, and we get

$$P_c(I - \tilde{M}^{-1}\tilde{A})R \begin{bmatrix} 0 & -A_{\rm ff}^{-1}A_{\rm fc} \\ 0 & A_{\rm cc}^{-1}A_{\rm cf}A_{\rm ff}^{-1}A_{\rm fc} \end{bmatrix}$$

$$= P_c(I - \tilde{M}^{-1}\tilde{A})[0 \quad A_{\rm cc}^{-1}A_{\rm cf}A_{\rm ff}^{-1}A_{\rm fc}].$$

Proof continued

Now parareal is only operating on the coarse variables, so we obtain for them the error propagation operator of MGRIT to be

$$(I-\tilde{M}^{-1}\tilde{A})A_{\mathrm{cc}}^{-1}A_{\mathrm{cf}}A_{\mathrm{ff}}^{-1}A_{\mathrm{fc}}. \qquad (*)$$

Now recall the Schur complement

$$S_{\mathrm{cc}} = A_{\mathrm{cc}} - A_{\mathrm{cf}}A_{\mathrm{ff}}^{-1}A_{\mathrm{fc}},$$

which equals $\tilde{A}$, and since in parareal $A_{\mathrm{cc}} = I$, because the original matrix only contains ones on the diagonal, we get

$$I-\tilde{A} = I-(A_{\mathrm{cc}} - A_{\mathrm{cf}}A_{\mathrm{ff}}^{-1}A_{\mathrm{fc}}) = A_{\mathrm{cc}}^{-1}A_{\mathrm{cf}}A_{\mathrm{ff}}^{-1}A_{\mathrm{fc}},$$

and inserting this into (*) we see that the error propagation operator of MGRIT on the coarse variables is simply

$$(I-\tilde{M}^{-1}\tilde{A})(I-\tilde{A}).$$

This corresponds to the two step iterative procedure

$$\begin{aligned} \boldsymbol{Y}^k &= \boldsymbol{U}^k + \boldsymbol{f} - \tilde{A}\boldsymbol{U}^k, \\ \tilde{M}\boldsymbol{U}^{k+1} &= \tilde{M}\boldsymbol{Y}^k + \boldsymbol{f} - \tilde{A}\boldsymbol{Y}^k. \end{aligned}$$

Proof continued

Writing this componentwise, we obtain

$$
\begin{aligned}
Y_0^k &= u_0, \quad Y_n^k = \tilde{F} U_{n-1}^k \\
U_0^k &= u_0, \quad U_n^{k+1} = \tilde{F} Y_{n-1}^k + \tilde{G} U_{n-1}^{k+1} - \tilde{G} Y_{n-1}^k.
\end{aligned}
$$

Substituting the values of Y_n^k into the equation for U_n^{k+1} then yields the result.

Corollary

The two level MGRIT algorithm with the $F(CF)^\nu$-smoother computes the same iterations as the parareal algorithm using $\nu \Delta T$ overlap.

PFASST

PFASST stands for Parallel Full Approximation Scheme in Space-Time, and there are several steps in the development PFASST:

Minion (2010): A hybrid parareal spectral deferred corrections method

> *"This paper investigates a variant of the parareal algorithm first outlined by Minion and Williams in 2008 that utilizes* **a deferred correction strategy within the parareal iterations**."

Deferred correction: consider the initial value problem

$$u' = f(u), \quad u(0) = u_0.$$

We can rewrite this problem in integral form

$$u(t) = u(0) + \int_0^t f(u(\tau))d\tau.$$

Integral Deferred Correction

Let $\tilde{u}(t)$ be an approximation with error $e(t) := u(t) - \tilde{u}(t)$. Inserting $u(t) = \tilde{u}(t) + e(t)$ into the integral form, we get

$$\tilde{u}(t) + e(t) = u(0) + \int_0^t f(\tilde{u}(\tau) + e(\tau))d\tau. \quad (*)$$

Let $F(u) := u(0) + \int_0^t f(u(\tau))d\tau - u(t)$ from the integral form, the residual $r(t)$ of the approximate solution $\tilde{u}(t)$ is

$$r(t) := F(\tilde{u}) = \tilde{u}(0) + \int_0^t f(\tilde{u}(\tau))d\tau - \tilde{u}(t),$$

and thus from (*) the error satisfies the equation

$$\begin{aligned} e(t) &= u(0) + \int_0^t f(\tilde{u}(\tau) + e(\tau))d\tau - \tilde{u}(t) \\ &= r(t) + \int_0^t f(\tilde{u}(\tau) + e(\tau)) - f(\tilde{u}(\tau))d\tau, \end{aligned}$$

or written as a differential equation

$$e'(t) = r'(t) + f(\tilde{u}(t) + e(t)) - f(\tilde{u}(t)).$$

Integral Deferred Correction

Idea of integral deferred correction:

1. use e.g. Forward Euler to get a first approximate solution of the ODE,
$$\tilde{u}_{m+1} = \tilde{u}_m + \Delta t f(\tilde{u}_m), \quad \text{for } m = 0, 1, \ldots, M-1.$$
2. With these values, compute the residual at the points t_m, $m = 0, 1, \ldots, M$ using a high order quadrature formula.
3. Solve the error equation in differential form again with Forward Euler,
$$e_{m+1} = e_m + r_{m+1} - r_m + \Delta t(f(\tilde{u}_m + e_m) - f(\tilde{u}_m)).$$
4. Add this correction to obtain a new approximation
$$\tilde{u}_m + e_m,$$

One can show that the order has increased by one, and one can continue this process to increase the order up to the order of the quadrature used.

Integral Deferred Correction as Iteration

This is an iterative method to compute the Runge-Kutta method corresponding to the quadrature rule used to approximate the integral: with $\boldsymbol{u}^0$ obtained by forward Euler, we have the non-linear fixed point iteration

$$\boldsymbol{u}^k = F(\boldsymbol{u}^{k-1}, u_0).$$

Classical Use of Integral Deferred Correction:
partition the time interval $[0, T]$ into subintervals $[T_{j-1}, T_j]$ $j = 1, 2, \ldots, J$, and then perform K iterations on each:

$u_{0,M}^K = u_0$;
for $j = 1 : J$
  compute $\boldsymbol{u}_j^0$ as Euler approximation on $[T_{j-1}, T_j]$;
  for $k = 1 : K$
    $\boldsymbol{u}_j^k = F(\boldsymbol{u}_j^{k-1}, u_{j-1,M}^K)$;
  end;
end;

This is purely sequential, like a time stepping scheme

Idea Proposed by Minion (2010)

Replace the inner updating formula by (see Womble later)

$$\boldsymbol{u}_j^k = F(\boldsymbol{u}_j^{k-1}, u_{j-1,M}^k), \quad \text{(note the lower case } k \text{ !).}$$

Can now perform spectral deferred corrections in parallel.

Minion (2010) combines this with a coarse correction from parareal, thus using a more and more accurate fine integrator.

PFASST Emmett, Minion (2012) uses this as a smoother in a FAS scheme in space-time:

> *"The method is iterative with each iteration consisting of* **deferred correction sweeps performed alternately on fine and coarse space-time discretizations**. *The coarse grid problems are formulated using a space-time analog of the* **full approximation scheme popular in multigrid methods for nonlinear equations**.*"*