CISM COURSE COMPUTATIONAL ACOUSTICS

Solvers

Part 5: Multigrid II

Ulrich Langer and **Martin Neumüller**
Institute of Computational Mathematics
Johannes Kepler University Linz
Udine, May 23-27, 2016

JOHANNES KEPLER UNIVERSITY LINZ

Institut für NuMa Numerische Mathematik

motus CISM est vita

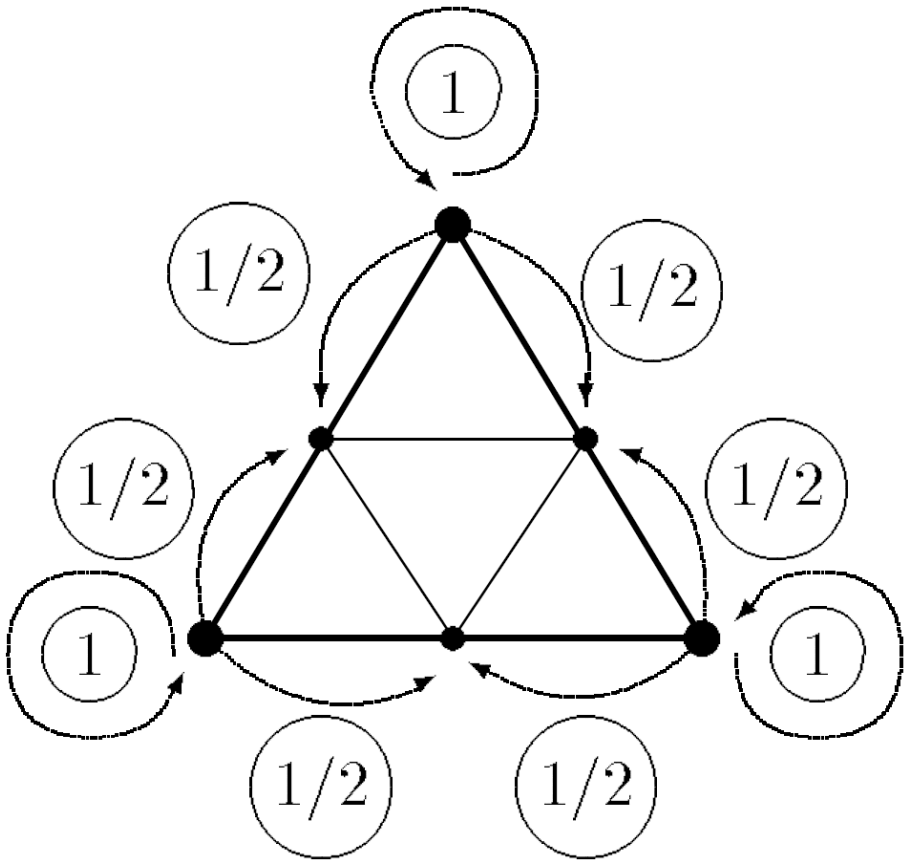
Outline

Outline

Higher dimensions

Changes:

- **Smoother:** all introduced smoothers possible ✓
  $\rightarrow$ only other optimal damping parameter
- **Restriction/ prolongation:**



Outline

Algebraic multigrid method

Some aspects:

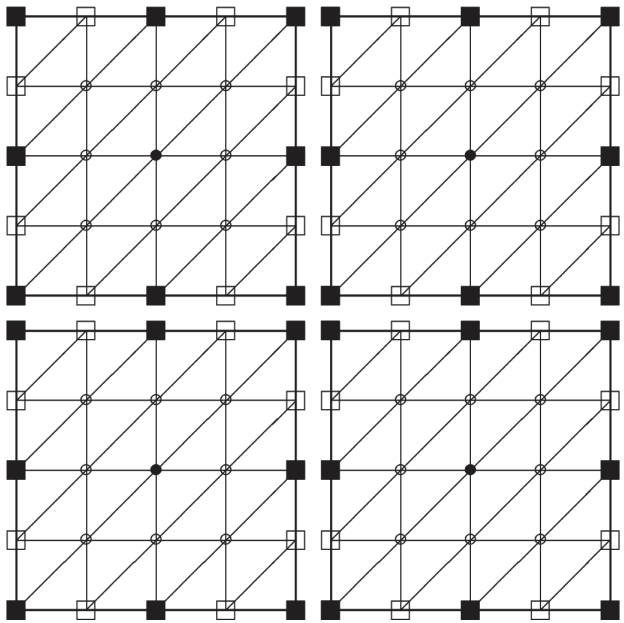
- Construct coarse problems by using only matrix information
  - Construction of coarse grid $\rightarrow$ find coarse and fine nodes (e.g. Ruge/Stüben coarsening)
  - Compute weights for restriction and prolongation
  - Generate coarse system matrix
- Many possible algorithms and options
- After the construction of the coarse levels $\rightarrow$ standard multigrid procedure
- Parallel implementation possible (e.g. software package hypre)

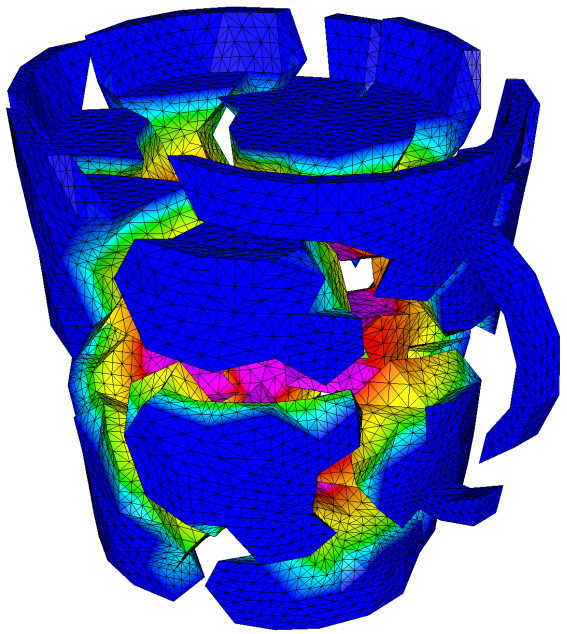
Outline

Parallelization

Some remarks:

- Partition mesh over the processors $\rightarrow$ METIS
- Distributed assembling of matrices and vectors
- Communication over the interfaces
- Coarse grid solver: parallel or sequentiell on one processor?





Outline

Time-multigrid methods

Consider:

$$\partial_t u(t) + u(t) = f(t) \quad \forall t \in (0,T), \qquad u(0) = u_0.$$

Implicit Euler:

$$(1+\tau)u_{k+1} = \tau\, f(t_{k+1}) + u_k \qquad k = 0, 1, 2, \ldots$$

Big linear system:

$$\begin{pmatrix} 1+\tau & & & & \\ -1 & 1+\tau & & & \\ & \ddots & \ddots & & \\ & & -1 & 1+\tau & \\ & & & -1 & 1+\tau \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_{m-1} \\ u_m \end{pmatrix} = \begin{pmatrix} \tau f(t_1) + u_0 \\ \tau f(t_2) \\ \vdots \\ \tau f(t_{m-1}) \\ \tau f(t_m) \end{pmatrix}.$$

$\rightarrow$ apply multigird idea

Time-multigrid methods

Test smoothing iteration:

- Use rhs $= 0$ and $\underline{u}^{(0)} = [\mathrm{rand}(0,1)]_{j=1}^{m}$.
- Apply Jacobi method for $\alpha = 0.5$:

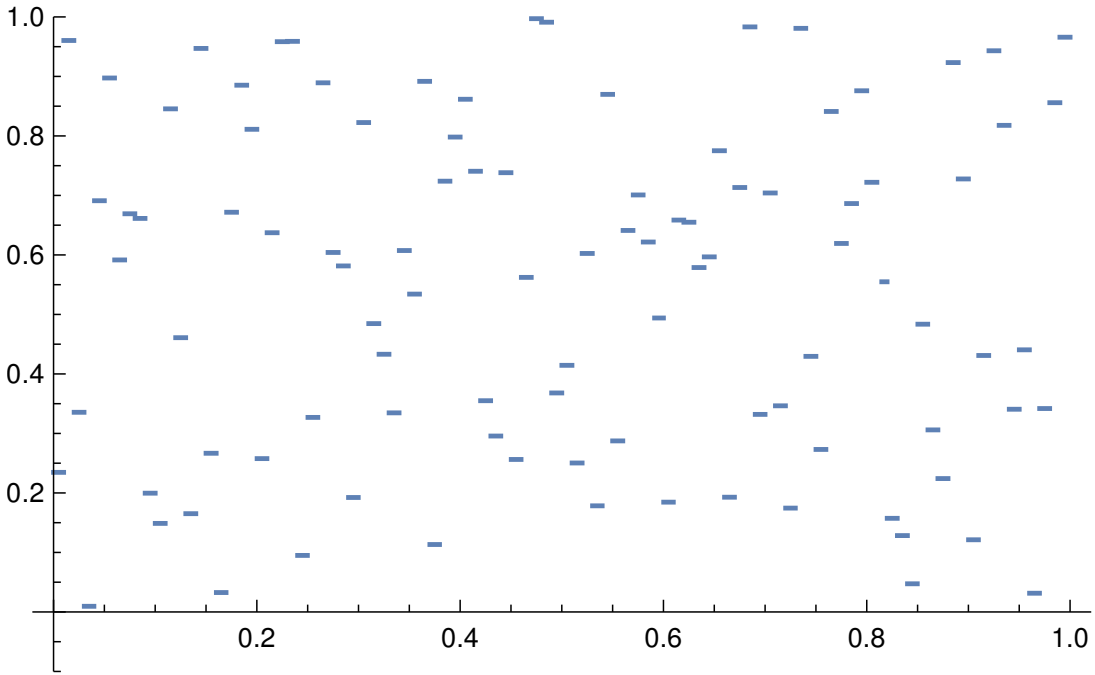


Figure: $k = 0$

- Parallel smoother $\rightarrow$ parallel multigrid method
- Full parallel methods for parabolic problems

Time-multigrid methods

Test smoothing iteration:

- Use rhs $= 0$ and $\underline{u}^{(0)} = [\mathrm{rand}(0,1)]_{j=1}^{m}$.
- Apply Jacobi method for $\alpha = 0.5$:

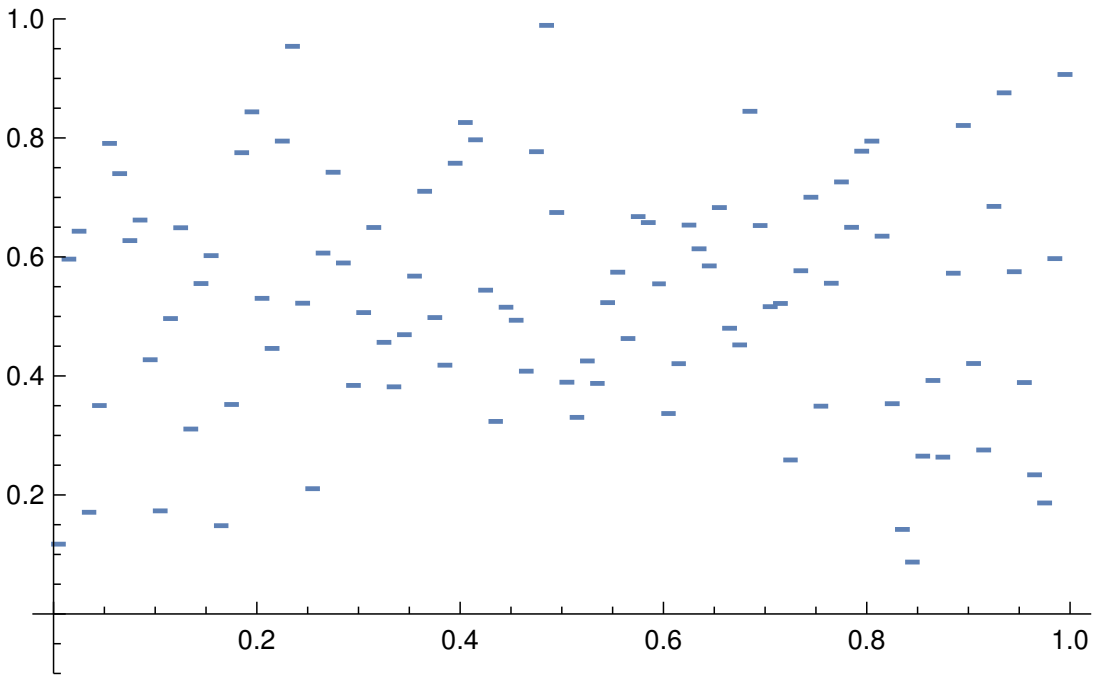


Figure: $k = 1$

- Parallel smoother $\rightarrow$ parallel multigrid method
- Full parallel methods for parabolic problems

Time-multigrid methods

Test smoothing iteration:

- Use rhs $= 0$ and $\underline{u}^{(0)} = [\mathrm{rand}(0,1)]_{j=1}^{m}$.
- Apply Jacobi method for $\alpha = 0.5$:

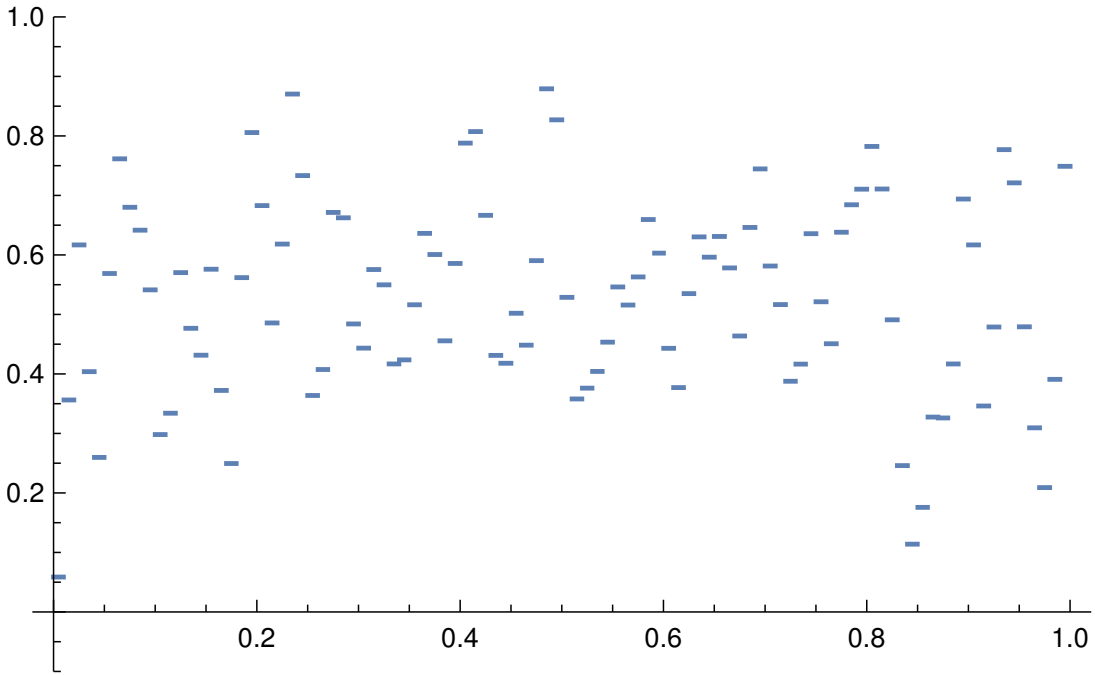


Figure: $k = 2$

- Parallel smoother $\rightarrow$ parallel multigrid method
- Full parallel methods for parabolic problems

Time-multigrid methods

Test smoothing iteration:

- Use rhs $= 0$ and $\underline{u}^{(0)} = [\mathrm{rand}(0,1)]_{j=1}^{m}$.
- Apply Jacobi method for $\alpha = 0.5$:

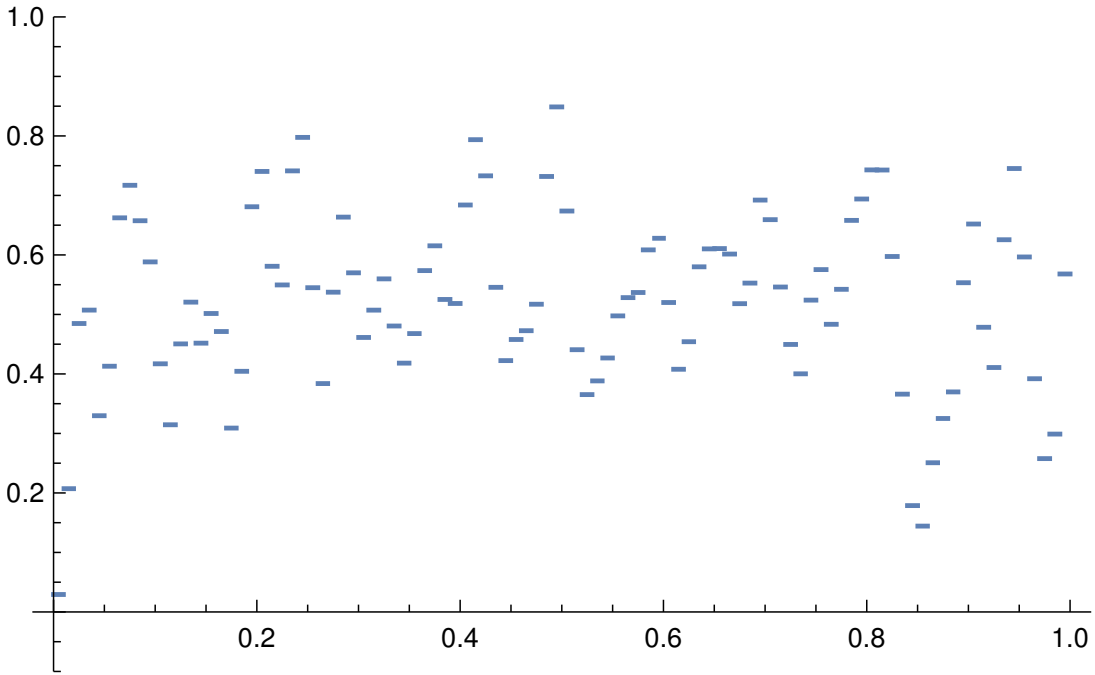


Figure: $k = 3$

- Parallel smoother $\rightarrow$ parallel multigrid method
- Full parallel methods for parabolic problems

Time-multigrid methods

Test smoothing iteration:

- Use rhs $= 0$ and $\underline{u}^{(0)} = [\mathrm{rand}(0,1)]_{j=1}^{m}$.
- Apply Jacobi method for $\alpha = 0.5$:

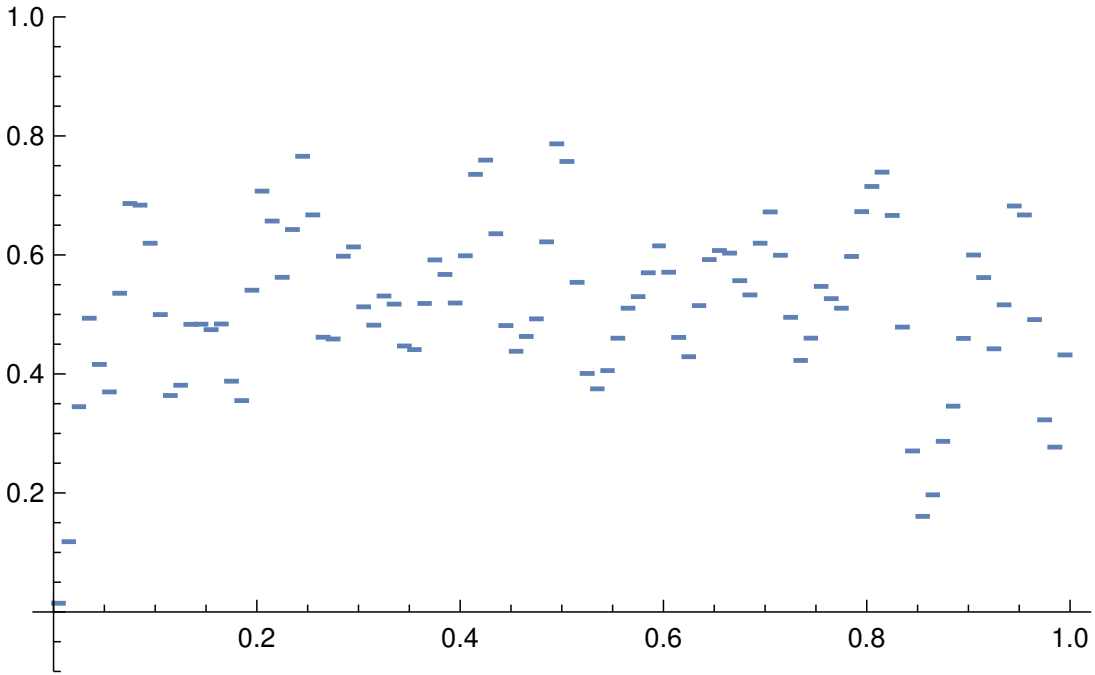


Figure: $k = 4$

- Parallel smoother $\rightarrow$ parallel multigrid method
- Full parallel methods for parabolic problems

Time-multigrid methods

Test smoothing iteration:

- Use rhs $= 0$ and $\underline{u}^{(0)} = [\mathrm{rand}(0,1)]_{j=1}^{m}$.
- Apply Jacobi method for $\alpha = 0.5$:

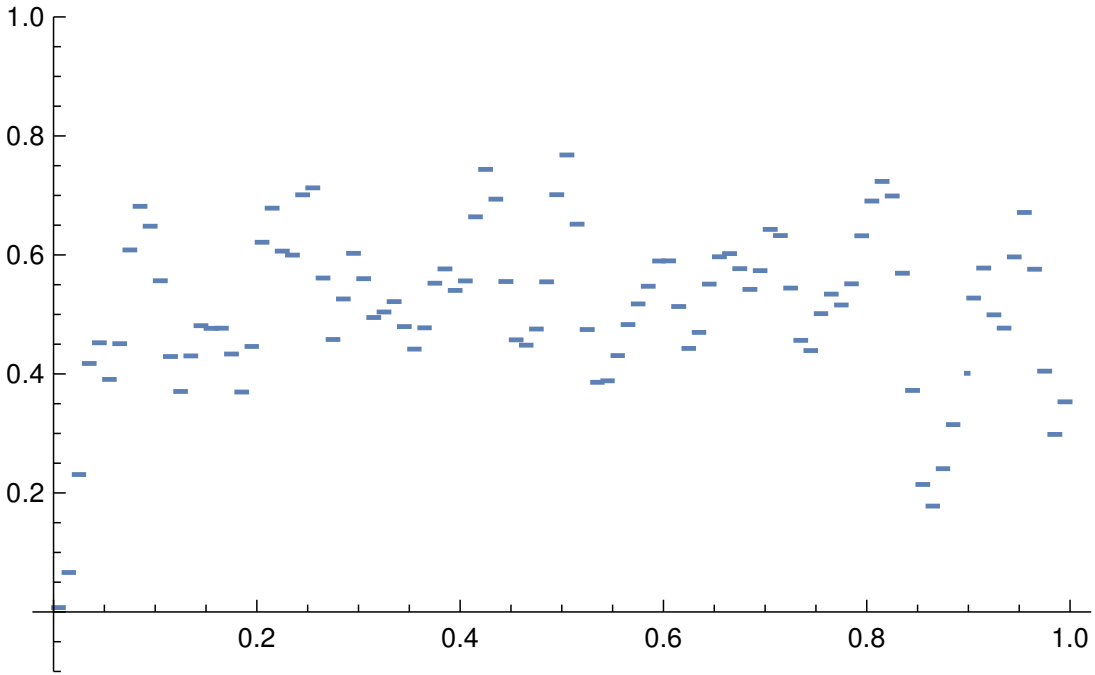


Figure: $k = 5$

- Parallel smoother $\rightarrow$ parallel multigrid method
- Full parallel methods for parabolic problems

Time-multigrid methods

Test smoothing iteration:

- Use rhs $= 0$ and $\underline{u}^{(0)} = [\mathrm{rand}(0,1)]_{j=1}^{m}$.
- Apply Jacobi method for $\alpha = 0.5$:

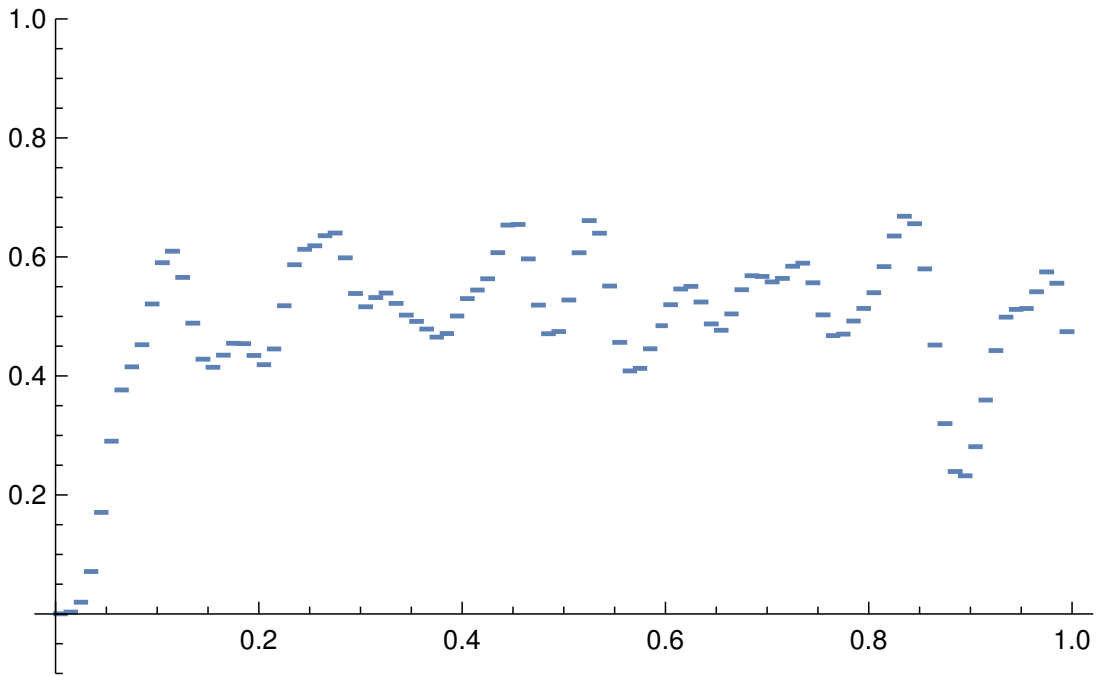


Figure: $k = 10$

- Parallel smoother $\rightarrow$ parallel multigrid method
- Full parallel methods for parabolic problems

Time-multigrid methods

Test smoothing iteration:

- Use rhs $= 0$ and $\underline{u}^{(0)} = [\mathrm{rand}(0,1)]_{j=1}^{m}$.
- Apply Jacobi method for $\alpha = 0.5$:

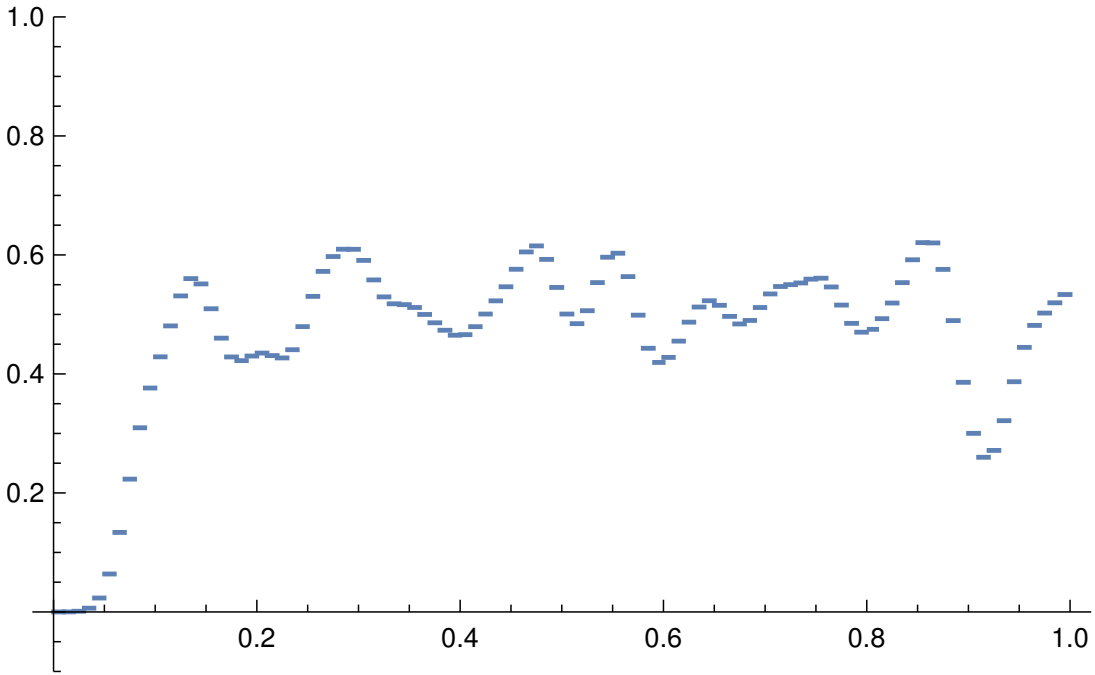


Figure: $k = 15$

- Parallel smoother $\rightarrow$ parallel multigrid method
- Full parallel methods for parabolic problems

Time-multigrid methods

Test smoothing iteration:

- Use rhs $= 0$ and $\underline{u}^{(0)} = [\mathrm{rand}(0,1)]_{j=1}^{m}$.
- Apply Jacobi method for $\alpha = 0.5$:

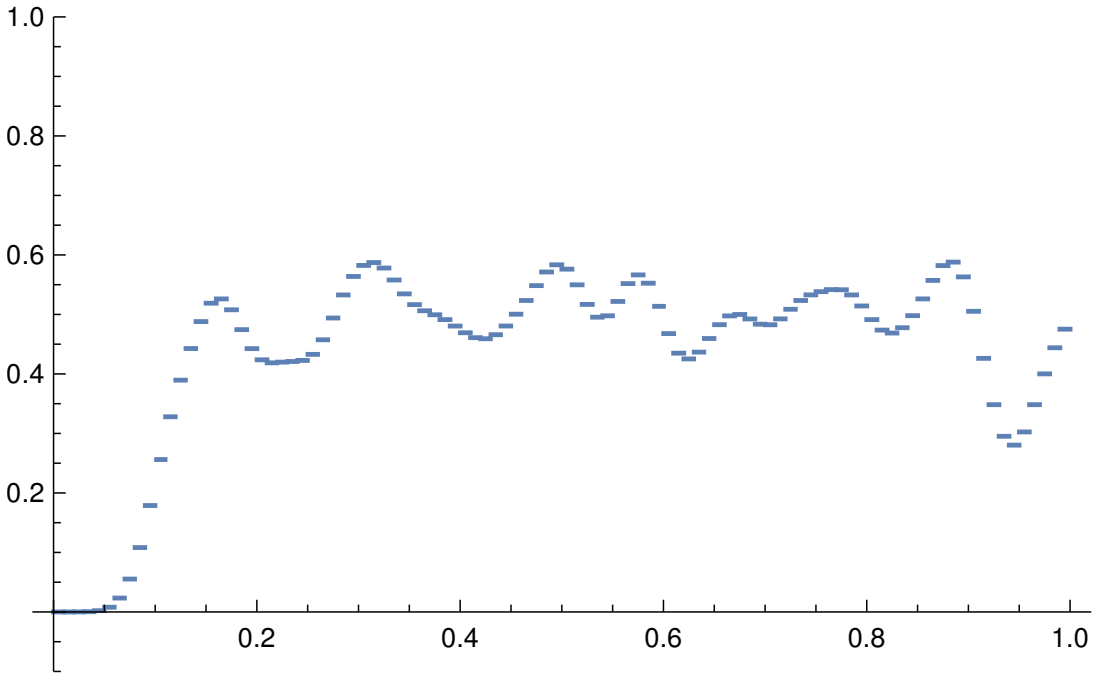


Figure: $k = 20$

- Parallel smoother $\rightarrow$ parallel multigrid method
- Full parallel methods for parabolic problems

Time-multigrid methods

Test smoothing iteration:

- Use rhs $= 0$ and $\underline{u}^{(0)} = [\mathrm{rand}(0,1)]_{j=1}^{m}$.
- Apply Jacobi method for $\alpha = 0.5$:

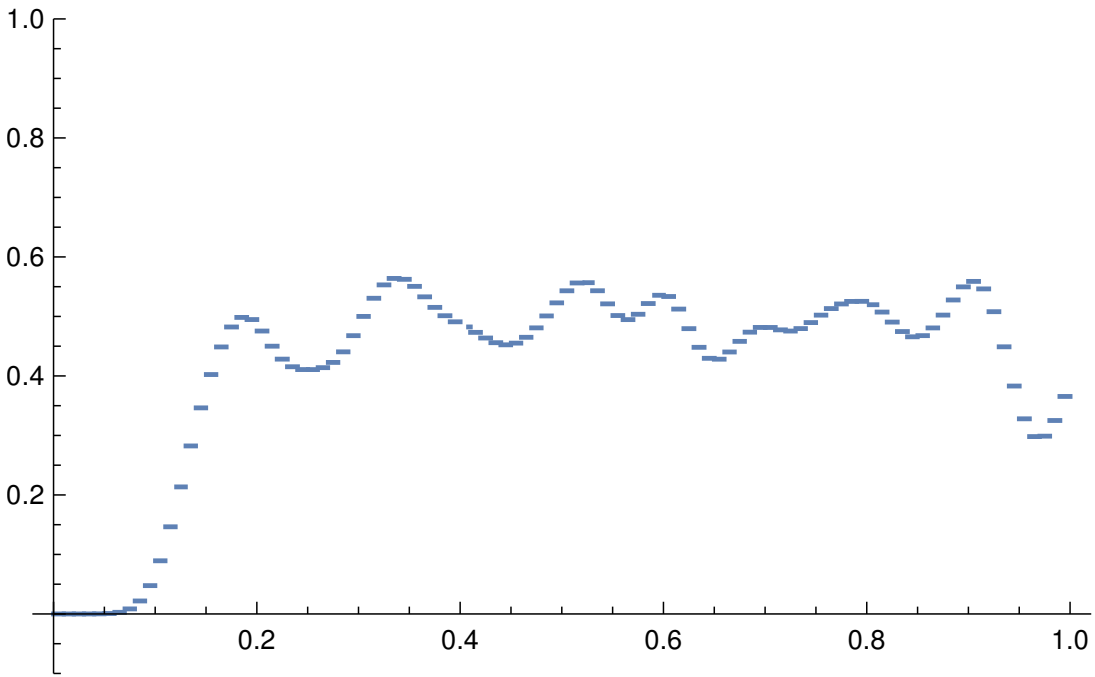


Figure: $k = 25$

- Parallel smoother $\rightarrow$ parallel multigrid method
- Full parallel methods for parabolic problems

Time-multigrid methods

Test smoothing iteration:

- Use rhs $= 0$ and $\underline{u}^{(0)} = [\mathrm{rand}(0,1)]_{j=1}^{m}$.
- Apply Jacobi method for $\alpha = 0.5$:

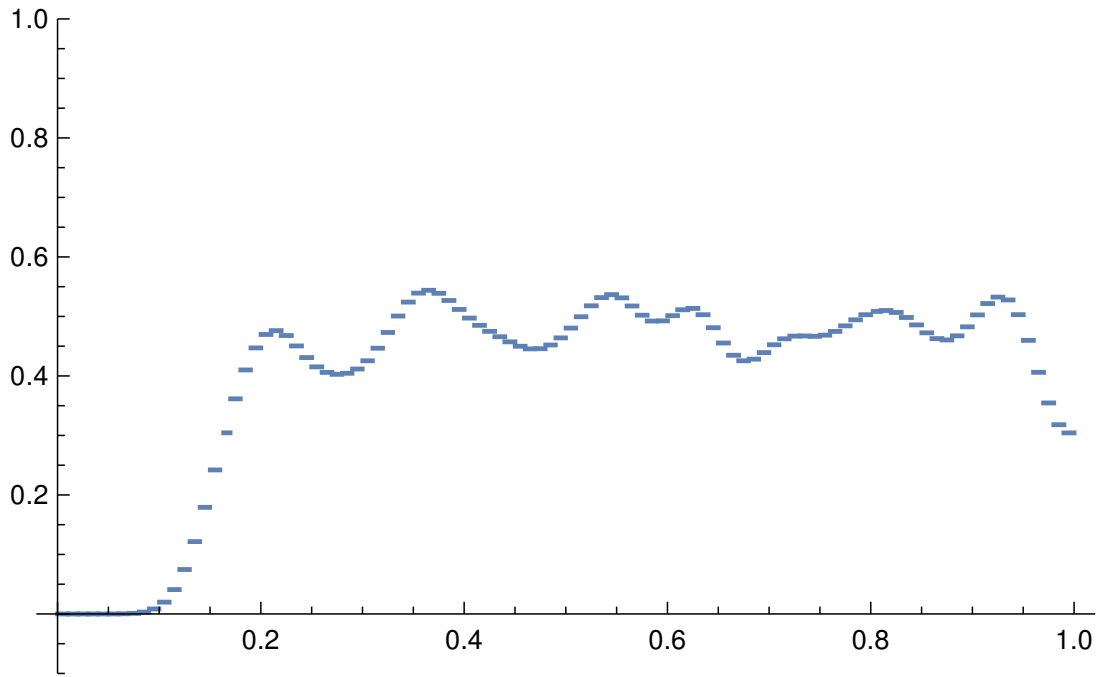


Figure: $k = 30$

- Parallel smoother $\rightarrow$ parallel multigrid method
- Full parallel methods for parabolic problems

Time-multigrid methods

Test smoothing iteration:

- Use rhs $= 0$ and $\underline{u}^{(0)} = [\mathrm{rand}(0,1)]_{j=1}^{m}$.
- Apply Jacobi method for $\alpha = 0.5$:

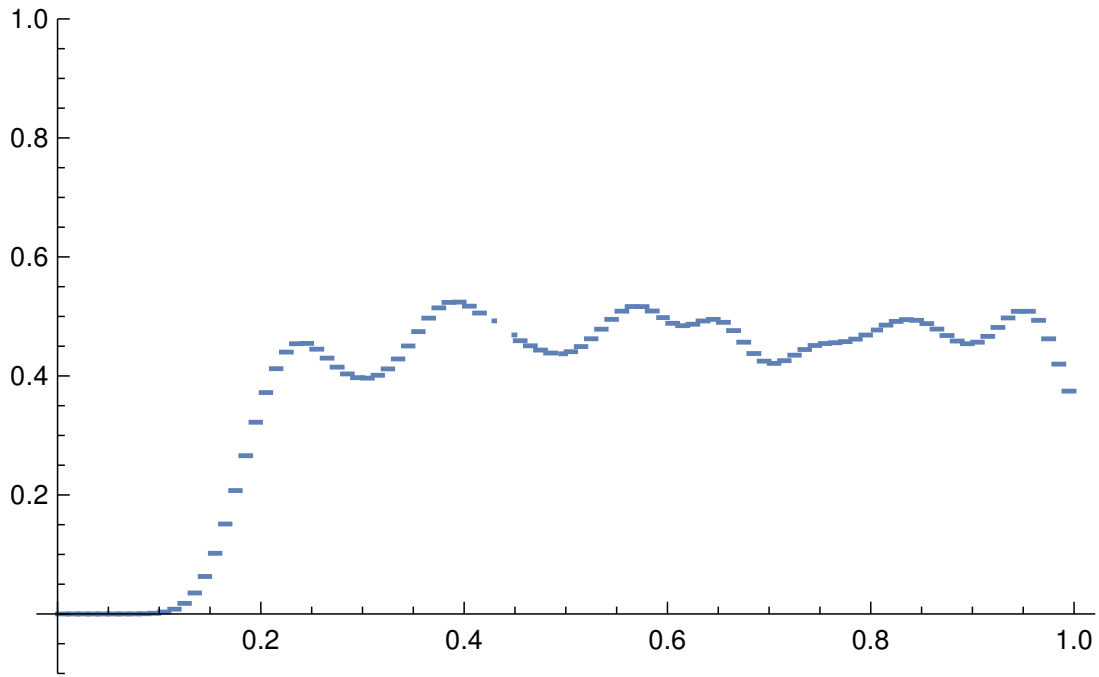


Figure: $k = 35$

- Parallel smoother $\rightarrow$ parallel multigrid method
- Full parallel methods for parabolic problems

Time-multigrid methods

Test smoothing iteration:

- Use rhs $= 0$ and $\underline{u}^{(0)} = [\mathrm{rand}(0,1)]_{j=1}^{m}$.
- Apply Jacobi method for $\alpha = 0.5$:

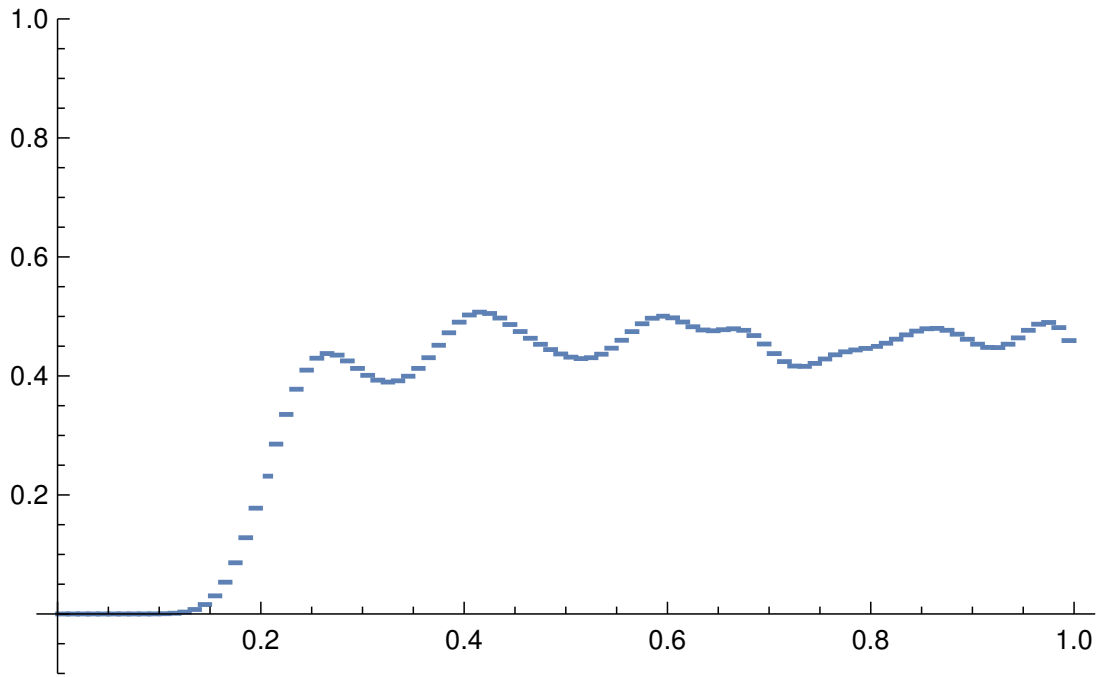


Figure: $k = 40$

- Parallel smoother $\rightarrow$ parallel multigrid method
- Full parallel methods for parabolic problems

Time-multigrid methods

Test smoothing iteration:

- Use rhs $= 0$ and $\underline{u}^{(0)} = [\mathrm{rand}(0,1)]_{j=1}^{m}$.
- Apply Jacobi method for $\alpha = 0.5$:

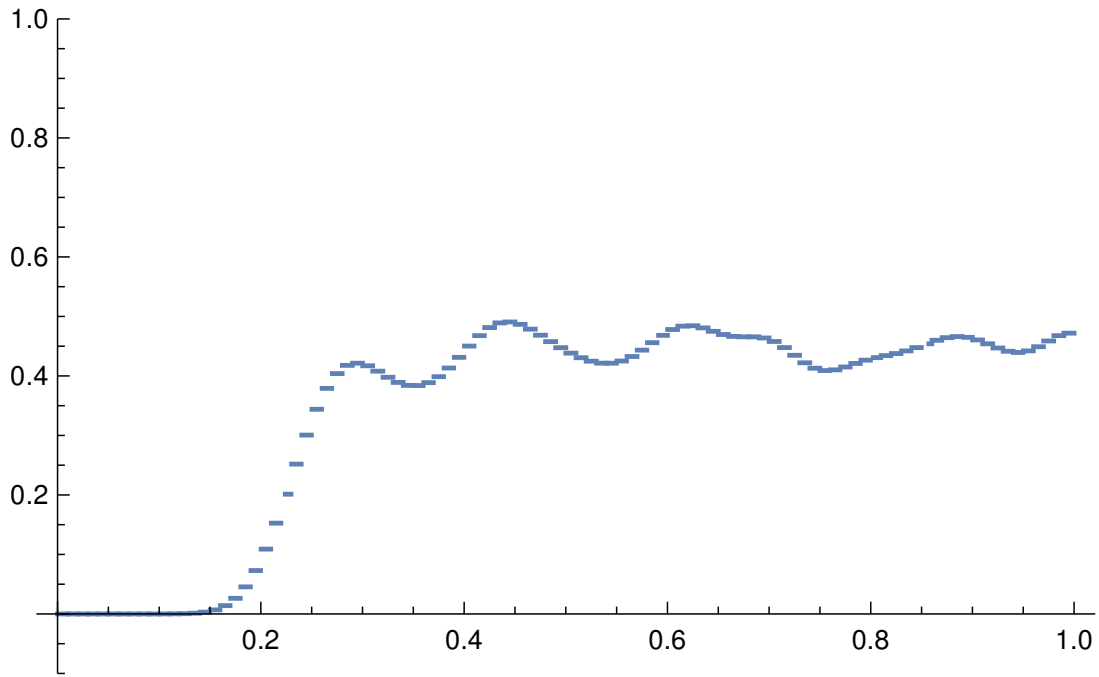


Figure: $k = 45$

- Parallel smoother $\rightarrow$ parallel multigrid method
- Full parallel methods for parabolic problems

Time-multigrid methods

Test smoothing iteration:

- Use rhs $= 0$ and $\underline{u}^{(0)} = [\mathrm{rand}(0,1)]_{j=1}^{m}$.
- Apply Jacobi method for $\alpha = 0.5$:

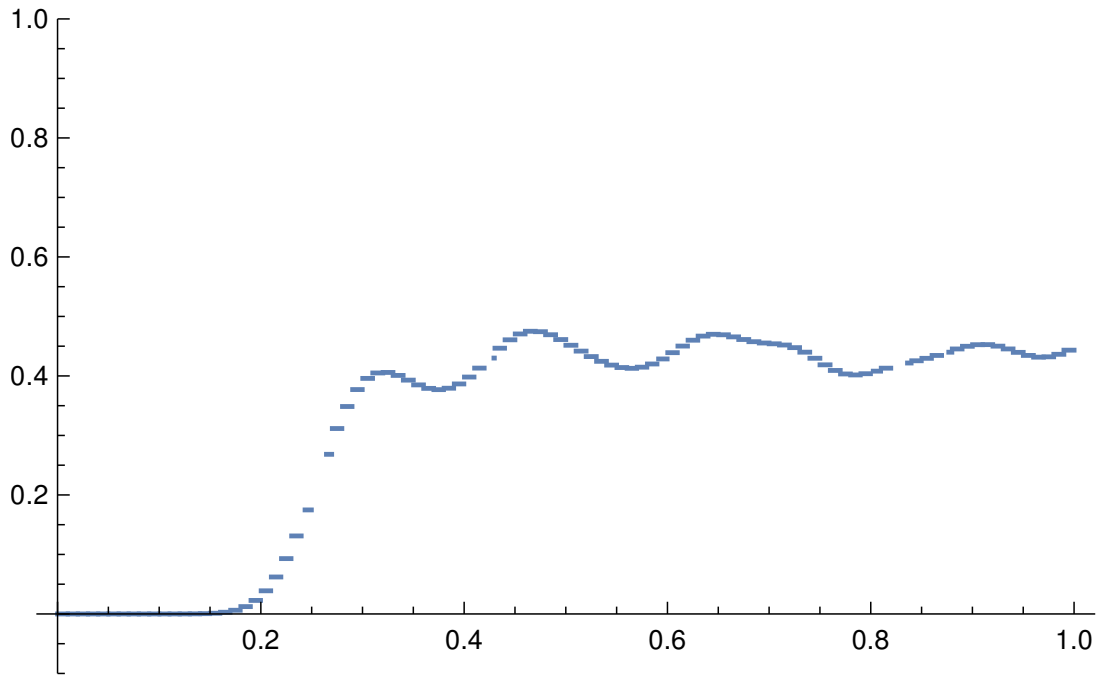


Figure: $k = 50$

- Parallel smoother $\rightarrow$ parallel multigrid method
- Full parallel methods for parabolic problems

Time-multigrid methods

Test smoothing iteration:

- Use rhs $= 0$ and $\underline{u}^{(0)} = [\mathrm{rand}(0,1)]_{j=1}^{m}$.
- Apply Jacobi method for $\alpha = 0.5$:

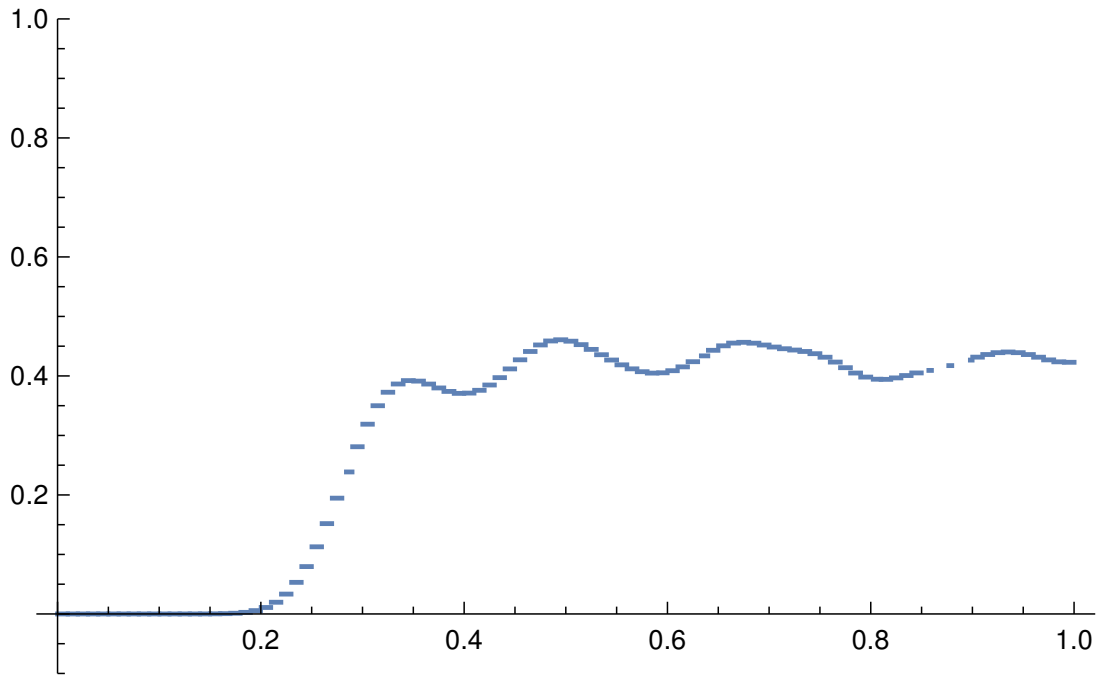


Figure: $k = 55$

- Parallel smoother $\rightarrow$ parallel multigrid method
- Full parallel methods for parabolic problems

Time-multigrid methods

Test smoothing iteration:

- Use rhs $= 0$ and $\underline{u}^{(0)} = [\mathrm{rand}(0,1)]_{j=1}^{m}$.
- Apply Jacobi method for $\alpha = 0.5$:

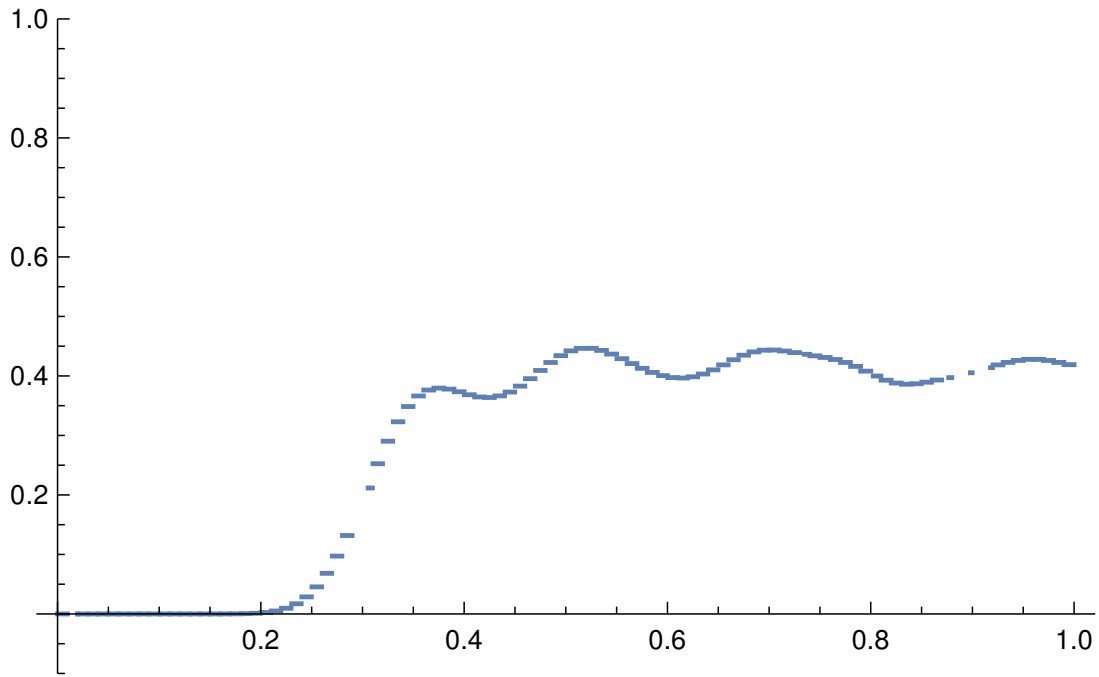


Figure: $k = 60$

- Parallel smoother $\rightarrow$ parallel multigrid method
- Full parallel methods for parabolic problems

Parabolic multigrid

- $\Omega = (0,1)^3$, $\tau = 10^{-1}$, $p_t = 3$, space level 4 ($49\,152$ elements)
- computed on VULCAN (IBM Blue Gene/Q), LLNL, USA

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

Table: Weak scaling.

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

Table: Strong scaling.

Outline

Summary

- Multigrd method in higher dimensions
- Algebraic multigird method
- Parallelization
- Time-multigrid methods