Multigrid methods on high performance computers

Matthias Bolten

ICERM Topical Workshop “Asynchronous Methods for Numerical Linear Algebra”

May 6, 2026

MATHEMATICAL MODELLING,
ANALYSIS AND
COMPUTATIONAL MATHEMATICS

BERGISCHE
UNIVERSITÄT
WUPPERTAL

Acknowledgements

- Lisa Claus (Lawrence Berkeley National Laboratory, Berkeley, USA)
- Marco Donatelli (Università dell'Insubria, Como, Italy)
- Paola Ferrari (now in industry)
- Isabella Furci (Università Telematica San Raffaele Roma, Rome, Italy)
- Simon Hoffmann (Bergische Universität Wuppertal, Wuppertal, Germany)
- Thomas Huckle (TU München, Munich, Germany)
- Oliver Letterer (now in industry)

Outline

Outline

Multigrid
- Multigrid
- Parallel multigrid

Multgrid for structured matrices
- Multigrid for Toeplitz and circulant matrices
- Multigrid for symmetric positive definite block matrices
- Aggregation-based multigrid for Toeplitz matrices
- Block smoothers in multigrid methods for Toeplitz matrices

Advanced smoothing techniques
- Block smoothers
- Smoothers for systems of PDEs

Asynchrony in multigrid methods

Conclusions

Two-grid method

We solve

$$A_n x_n = b_n$$

using a combination of pre-smoothing (denoted by $\mathcal{V}_{n,\mathrm{pre}}$), coarse grid correction and post-smoothing (denoted by $\mathcal{V}_{n,\mathrm{post}}$).

TGM$(A_n, \mathcal{V}_{n,\mathrm{pre}}^{\nu_{\mathrm{pre}}}, \mathcal{V}_{n,\mathrm{post}}^{\nu_{\mathrm{post}}}, P_{n,k}, b_n, x_n^{(j)})$

0. $\tilde{x}_n = \mathcal{V}_{n,\mathrm{pre}}^{\nu_{\mathrm{pre}}}(A_n, b_n, x_n^{(j)})$
1. $r_n = b_n - A_n \tilde{x}_n$
2. $r_k = P_{n,k}^H r_n$
3. $A_k = P_{n,k}^H A_n P_{n,k}$
4. Solve $A_k y_k = r_k$
5. $\hat{x}_n = \tilde{x}_n + P_{n,k} y_k$
6. $x_n^{(j+1)} = \mathcal{V}_{n,\mathrm{post}}^{\nu_{\mathrm{post}}}(A_n, b_n, \hat{x}_n)$

Multigrid method

TGM is a stationary method with iteration matrix

$$\mathrm{TGM}(A_n, V_{n,\mathrm{pre}}^{\nu_{\mathrm{pre}}}, V_{n,\mathrm{post}}^{\nu_{\mathrm{post}}}, P_{n,k}) = \\ V_{n,\mathrm{post}}^{\nu_{\mathrm{post}}} \left[I_n - P_{n,k} \left(P_{n,k}^H A_n P_{n,k} \right)^{-1} P_{n,k}^H A_n \right] V_{n,\mathrm{pre}}^{\nu_{\mathrm{pre}}}.$$

Replacing the inversion of $P_{n,k}^H A_n P_{n,k}$ by recursive application of the TGM yields the multigrid method, given by

$$\mathrm{MGM}_{\ell_{\min}}(A_{n_{\ell_{\min}}}, V_{n_{\ell_{\min}}}^{\nu}, V_{n_{\ell_{\min}}}^{\nu}, P_{n_{\ell_{\min}}, k_{\ell_{\min}}}) = O_{n_{\ell_{\min}}, n_{\ell_{\min}}},$$

$$\mathrm{MGM}_{\ell}(A_{n_\ell}, V_{n_\ell}^{\nu}, V_{n_\ell}^{\nu}, P_{n_\ell, k_\ell}) = \\ V_{n_\ell}^{\nu} \left[I_{n_\ell} - P_{n_\ell, k_\ell} \left(I_{n_\ell + 1} - \mathrm{MGM}_{\ell+1} \right) \left(P_{n_\ell, k_\ell}^H A_{n_\ell} P_{n_\ell, k_\ell} \right)^{-1} P_{n_\ell, k_\ell}^H A_{n_\ell} \right] V_{n_\ell}^{\nu}.$$

Convergence of TGM

Theorem (Ruge, Stüben [12])

Let A_n be a positive definite matrix of size n and let $V_{n,\mathrm{post}}$, $V_{n,\mathrm{pre}}$ be defined as in the TGM algorithm. Assume

(a) $\exists \alpha_{\mathrm{pre}} > 0 \, : \, \|V_{n,\mathrm{pre}} x_n\|_{A_n}^2 \leq \|x_n\|_{A_n}^2 - \alpha_{\mathrm{pre}} \|V_{n,\mathrm{pre}} x_n\|_{A_n^2}^2, \qquad \forall x_n \in \mathbb{C}^n,$

(b) $\exists \alpha_{\mathrm{post}} > 0 \, : \, \|V_{n,\mathrm{post}} x_n\|_{A_n}^2 \leq \|x_n\|_{A_n}^2 - \alpha_{\mathrm{post}} \|x_n\|_{A_n^2}^2, \qquad \forall x_n \in \mathbb{C}^n,$

(c) $\exists \gamma > 0 \, : \, \min_{y \in \mathbb{C}^k} \|x_n - P_{n,k} y\|_2^2 \leq \gamma \|x_n\|_{A_n}^2, \qquad \forall x_n \in \mathbb{C}^n.$

Then $\gamma \geq \alpha_{\mathrm{post}}$ and

$$\|\mathrm{TGM}(A_n, V_{n,\mathrm{pre}}, V_{n,\mathrm{post}}, P_{n,k})\|_{A_n} \leq \sqrt{\frac{1 - \alpha_{\mathrm{post}}/\gamma}{1 + \alpha_{\mathrm{pre}}/\gamma}} < 1.$$

Conditions (a) and (b) are known as *smooothing properties* and (c) is known as *approximation property*.

Convergence of MGM

Napov and Notay have shown that

$$\rho\left(\mathrm{MGM}_0(A_{n_0}, V_{n_0}^{\nu}, V_{n_0}^{\nu}, P_{n_0,k_0})\right) \leq$$

$$1 - \min_{\ell} \frac{1 - \rho\left(\mathrm{TGM}(A_{n_0}, V_{n_0}^{\nu}, V_{n_0}^{\nu}, P_{n_0,k_0})\right)}{\|\pi_{A_{n_\ell}}\|^2_{A_{n_\ell}(I_{n_\ell} - V_{n_\ell}^{2\nu})^{-1}}}.$$

We make use of the following theorem to show uniform boundedness:

Lemma (Napov, Notay [11])

Assume that there exists a positive C independent of n such that

$$\Lambda(A_{n_\ell}) \subseteq (0, C],$$

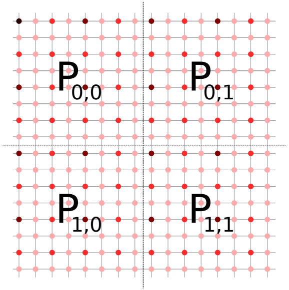
where $\Lambda(A_{n_\ell})$ denotes the spectrum of the matrix A_{n_ℓ}. Suppose that one iteration of the Richardson method with the damping parameter $\omega \in (0, 2/C)$ both as pre-smoother and post-smoother is applied. Then, the boundness of $\|\pi_{A_{n_\ell}}\|_2^2$ implies that $\|\pi_{A_{n_\ell}}\|^2_{A_{n_\ell}(I_{n_\ell} - V_{n_\ell}^{2\nu})^{-1}}$ is bounded as well.

Data distribution

- variables distributed to processors
- usually domain splitting approach
- variables are staying on the processors they have been assigned to initially $\rightsquigarrow$ unused processors on coarser levels
- parallel execution time:

$$T_{\mathcal{MG}}(N, P) = \mathcal{O}(N/P + \log P)$$

- result of inherently necessary information exchange that is necessary to solve the problem
- much better than `all-to-all`



Parallel computer architectures

GPUs

Usage for scientific computations rising in the past years (supercomputers as well as smaller clusters), but the programming model is more complicated.

Xeon Phi Cluster Oakforest-PACS

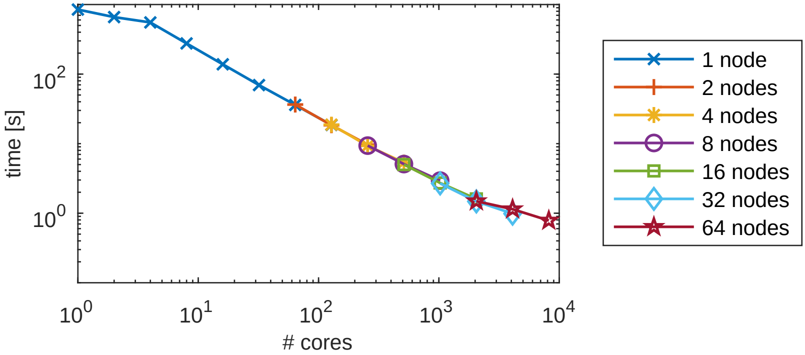
Massivley parallel system of the JSCAHPC (U Tokyo und U Tsukuba) with Intel x86 CPUs with 68 cores and fat-tree network with full bisection bandwidth (#16 TOP500 Jun. 2019).

Multigrid for Toeplitz matrices on Oakforest-PACS

- 3-level Toeplitz matrix of dimension $511 \times 511 \times 511$ with symbol

$$f(x, y, z) = 6 - 2\cos(x) - 2\cos(y) - 2\cos(z)$$

- Corresponds to 3D Laplace with 2nd order FD/FEM



Multigrid on multiple GPUs

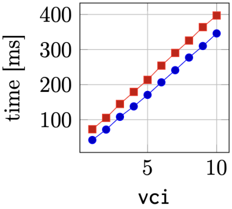
We consider three problems:

p1: $-\Delta u = f$ for $x \in \Omega = (0,1)^3$, periodic boundary conditions, given f

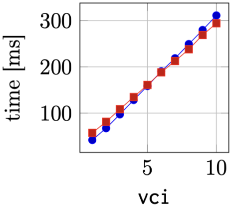
p2: $-\Delta u = f$ for $x \in \Omega = (0,1)^3$, $u = 0$ for $x \in \partial\Omega$, same f

p3: $-\epsilon u_{xx} - u_{yy} - u_{zz} = f$ for $x \in \Omega = (0,1)^3$, $u = 0$ for $x \in \partial\Omega$,
$\epsilon = \frac{1}{2}(2 + \sin(2\pi x) + \sin(2\pi y) + \sin(2\pi z))$, modified f

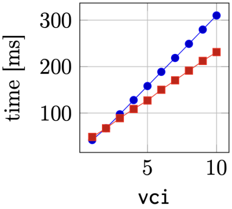
Comparison of our OpenCL solver `mgcl` with hypre on 8 NVIDIA A100 GPUs of Pleiades at University of Wuppertal, solving on 512^3 global grid:



(a) p1



(b) p2



(c) p3

Outline

Toeplitz and circulant matrices

- *Toeplitz matrices* $\{T_n\}_{n=0}^{\infty} \in \mathbb{C}^{n\times n}$ given by

$$T_n = \begin{pmatrix} t_0 & t_{-1} & t_{-2} & \cdots & t_{-n+1} \\ t_1 & t_0 & t_{-1} & \ddots & \vdots \\ t_2 & t_1 & t_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ t_{n-1} & t_{n-2} & t_{n-3} & \cdots & t_0 \end{pmatrix}$$

- Diagonal entries given by Fourier coefficients of *generating symbol* f.
- Classical result from Szegö:

Theorem

If $f \in L_\infty$ is real-valued, then the eigenvalues of the Hermitian Toeplitz matrices A_n are distributed as $f(x)$.

- *Circulant matrix* $\mathcal{C}_n \in \mathbb{C}^{n\times n}$ for trigonometric polynomial f of degree lower than n given by

$$\mathcal{C}_n(f) = F_n \mathrm{diag}(f(\theta_i^{(n)}))F_n^H,$$

$i = 0, \ldots, n-1$, with $\theta_i^{(n)} = \frac{2\pi i}{n}$ and

$$F_n = \frac{1}{\sqrt{n}} \left[\mathrm{e}^{-\imath j \theta_i^{(n)}} \right]_{i,j=0}^{n-1}$$

- Circulant matrices form an algebra of normal matrices
- Tensor products of Toeplitz and circulant matrices lead to *multilevel* matrices

Multigrid for circulant matrices

Assume the following:

- $A_i = \mathcal{A}_{\mathbf{n}_i}(f_i)$ is circulant
- f_i has single zero x^0 of even order β

Multigrid method constructed as follows:

- Define cut matrix $K_{\mathbf{n}_i,k} \in \mathbb{C}^{\mathbf{n}_{i+1}\times\mathbf{n}_i}$ as

$$K_{n_i,k} = \begin{bmatrix} 1 & 0 & \cdots & 0 & & & & \\ & & & & \ddots & & & \\ & & & & 1 & 0 & \cdots & 0 \end{bmatrix}$$

- Define sets of *corners* and *mirror points*

$$\Omega_g(x) = \{y : y_j \in \{x_j + 2k\pi/g, k = 0, \ldots, g-1\}\},$$
$$\mathcal{M}_g(x) = \Omega_g(x) \backslash \{x\}$$

- For $i = 0, \ldots, l_{\text{max}}$ choose p_i s.t.

$$\limsup_{x\to x^0} \left|\frac{p_i(y)}{f_i(x)}\right| < +\infty,\ y \in \mathcal{M}_g(x), \quad (1)$$
$$0 < \sum_{y\in\Omega_g(x)} p_i^2(y),\ \text{for all } x \in [-\pi,\pi) \quad (2)$$

- Set

$$P_i = \mathcal{A}_{\mathbf{n}_i}(p_i)K^H_{\mathbf{n}_i,k},$$
$$A_{i+1} = P_i^H A_i P_i$$

- Similar for Toeplitz matrices with modification of $K_{n_i,k}$

Block Toeplitz matrix

Definition

Let the Fourier coefficients of a function $\mathbf{f} : [-\pi, \pi] \to \mathbb{R}^{d \times d}$ with $f_i \in L^p([-\pi, \pi])$ be

$$\hat{\mathbf{f}}_j := \frac{1}{2\pi} \int_Q \mathbf{f}(\theta) \mathrm{e}^{-\mathrm{i}j\theta} d\theta \in \mathbb{R}^{d \times d}, \qquad j \in \mathbb{Z}.$$

Then, the block-Toeplitz matrix associated with **f** is the matrix with d blocks of size n and hence it has order $d \cdot n$ given by

$$T_n(\mathbf{f}) = \sum_{|j|<n} J_n^{(j)} \otimes \hat{\mathbf{f}}_j,$$

where $\otimes$ denotes the (Kronecker) tensor product of matrices. The term $J_n^{(j)}$ is the matrix of order n whose (i, k) entry equals 1 if $i - k = j$ and zero otherwise.

Multigrid for symmetric positive definite block circulant matrices

Assume the following:

- $A_i = \mathcal{A}_{n_i}(\mathbf{f}_i)$ is circulant
- There exists a unique $\theta_0 \in [0, 2\pi)$ and $\bar{j} \in \{1, \ldots, d\}$ s.t.

$$\begin{cases} \lambda_j(\mathbf{f}_i(\theta)) = 0, & \text{for } \theta = \theta_0 \text{ and } j = \bar{j}, \\ \lambda_j(\mathbf{f}_i(\theta)) > 0, & \text{otherwise} \end{cases} \tag{3}$$

- The order of this zero θ_0 is even

Multigrid method constructed as follows:

- Choose $\mathbf{p}_i$ as appropriate matrix-valued trigonometric polynomial
- Set

$$P_i = \mathcal{A}_{n_i}(\mathbf{p}_i)(K_{n_i,k}^H \otimes I_d) \quad \text{and} \quad A_{i+1} = P_i^H A_i P_i \tag{4}$$

- Similar for Toeplitz matrices with modification of $K_{n_i,k}$.

Requirements from $\mathbf{p}_i$

Sufficient conditions for the choice of $\mathbf{p}_i$ to imply convergence are

(i)

$$\mathbf{p}(\theta)^H\mathbf{p}(\theta) + \mathbf{p}(\theta+\pi)^H\mathbf{p}(\theta+\pi) > 0 \quad \forall\theta \in [0, 2\pi),$$

which implies that the trigonometric function

$$\mathbf{s}(\theta) = \mathbf{p}(\theta)\left(\mathbf{p}(\theta)^H\mathbf{p}(\theta) + \mathbf{p}(\theta+\pi)^H\mathbf{p}(\theta+\pi)\right)^{-1}\mathbf{p}(\theta)^H$$

is well-defined for all $\theta \in [0, 2\pi)$,

(ii)

$$\mathbf{s}(\theta_0)q_{\bar{\jmath}}(\theta_0) = q_{\bar{\jmath}}(\theta_0),$$

(iii)

$$\lim_{\theta\to\theta_0} \lambda_{\bar{\jmath}}(\mathbf{f}(\theta))^{-1}(1-\lambda_{\bar{\jmath}}(\mathbf{s}(\theta))) = c, \quad c \in \mathbb{R}.$$

Convergence of TGM for block circulant matrices

Theorem (B., Donatelli, Ferrari, Furci [2])

Consider the matrix $A_N := \mathcal{A}_n(\mathbf{f})$, with n even and $\mathbf{f} \in \mathcal{M}_d$ matrix-valued trigonometric polynomial, $\mathbf{f} \geq 0$, such that condition (3) is satisfied. Let P_i be the projecting operator defined as in equation (4) with $\mathbf{p} \in \mathcal{M}_d$ trigonometric polynomial satisfying conditions $(i) - (iii)$. Then, there exists a positive value γ independent of n such that

$$\exists \gamma > 0 \, : \, \min_{y \in \mathbb{C}^k} \|x_n - P_{n,k} y\|_2^2 \leq \gamma \|x_n\|_{A_n}^2, \qquad \forall x_n \in \mathbb{C}^n,$$

i.e. (c) in the Theorem mentioned above is fulfilled.

To check wether $\mathbf{p}_i$ fulfills the required properties we also derived a few technical lemmas that guarantee the fulfillment while being easier to check.

Convergence of MGM for block circulant matrices

Lemma (B., Donatelli, Ferrari, Furci [2])

Let $q_{\bar{\jmath}}(\theta_0)$ be the eigenvector associated with the ill-conditioned subspace of $\mathbf{f}(\theta_0)$. In addition, assume that the eigenvector $q_{\bar{\jmath}}(\theta_0)$ is s.t.:

(a) $q_{\bar{\jmath}}(\theta_0)$ is an eigenvector of $\mathbf{p}(\theta_0)$, associated to $\lambda^{(1)} \neq 0$ that is $\mathbf{p}(\theta_0)q_{\bar{\jmath}}(\theta_0) = \lambda^{(1)}q_{\bar{\jmath}}(\theta_0)$;

(b) $q_{\bar{\jmath}}(\theta_0)$ is an eigenvector of $\mathbf{p}(\theta_0 + \pi)$ associated with the zero eigenvalue, that is $\mathbf{p}(\theta_0 + \pi)q_{\bar{\jmath}}(\theta_0) = 0q_{\bar{\jmath}}(\theta_0)$.

If

$$\lim_{\theta \to \theta_0} \frac{|\lambda_{\bar{\jmath}}(\mathbf{p}(\theta + \pi))|}{\lambda_{\bar{\jmath}}(\mathbf{f}(\theta))} = c,$$

then there exists a δ independent from i s.t.

$$\|\pi_{A_i}\|_2 \leq \delta,$$

i.e. the V-cycle converges optimally.

Example: $\mathbb{Q}_r$ finite elements

- Consider the problem:

$$\begin{cases} -(a(x)u(x)')' = \psi(x), \text{ on } (0,1) \\ u(0) = u(1) = 0 \end{cases}$$

- Stiffness matrix for $\mathbb{Q}_r$ finite elements fulfills requirements
- Different generating symbols for the projector can be shown to fulfill the requirements on $\mathbf{p}_i$:
  - Linear interpolation for all grid points
  - Projector using the finite element basis functions
- For $r = 2$ we obtain:

$$\mathbf{f}_{\mathbb{Q}_2}(\theta) = \frac{1}{3}\begin{bmatrix} 16 & -8(1+\mathrm{e}^{\mathrm{i}\theta}) \\ -8(1+\mathrm{e}^{-\mathrm{i}\theta}) & 14+\mathrm{e}^{\mathrm{i}\theta}+\mathrm{e}^{-\mathrm{i}\theta} \end{bmatrix},$$

$$\mathbf{p}_{L_2}(\theta) = \begin{bmatrix} 1+\mathrm{e}^{-\mathrm{i}\theta} & \mathrm{e}^{\mathrm{i}\theta}+1 \\ 2\mathrm{e}^{-\mathrm{i}\theta} & 2 \end{bmatrix}$$

Aggregation-based multigrid for Toeplitz matrices

Aggregation-based multigrid methods are a common variant of algebraic multigrid methods.

- Widely used for non-structured problems
- Not used for Toeplitz matrices very often

In aggregation-based multigrid the restriction is given by

$$
\begin{aligned}
R^d_{n,k} = \left(P^d_{n,k}\right)^T &= \begin{pmatrix} 1 & \cdots & 1 & & & & & & \\ & & & 1 & \cdots & 1 & & & \\ & & & & & & \ddots & & \\ & & & & & & 1 & \cdots & 1 \end{pmatrix} \\
&= I_n \otimes \begin{pmatrix} 1 \cdots 1 \end{pmatrix}.
\end{aligned}
$$

This corresponds to a prolongation with the symbol $p(x) = \sum_{\ell=0}^{d} \mathrm{e}^{\mathrm{i}\ell x}$.

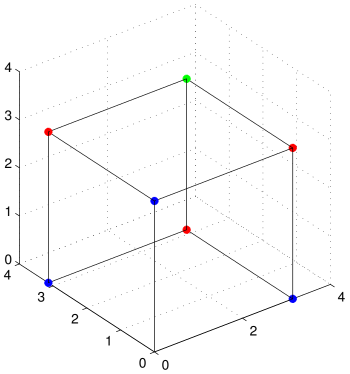
Analysis of smoothed aggregation

- only for some points in $\mathcal{M}(0)$ requirement (1) is not fulfilled
- more precisely we have

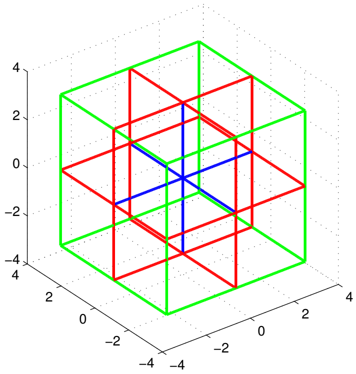
$$\limsup_{x \to \mathbf{0}} \frac{|p_i(y)|}{\sum_{j=1}^{d} x_j^r} = c, \; 0 < c < +\infty, \qquad y \in \mathcal{M}_g(x),$$

where $r = d - \# \{y_j \,|\, y_j = 0, \; j = 1, \dots, d\}$ is the number of directions along which $p(x)$ is zero

Order of the zeros in the 3-level case



$[-\pi, \pi] \times [-\pi, \pi]$



$[0, \pi] \times [0, \pi]$

Analysis of smoothed aggregation

- only for some points in $\mathcal{M}(0)$ requirement (1) is not fulfilled
- more precisely we have

$$\limsup_{x\to\mathbf{0}} \frac{|p_i(y)|}{\sum_{j=1}^{d} x_j^r} = c,\ 0 < c < +\infty, \qquad y \in \mathcal{M}_g(x),$$

  where $r = d - \#\{y_j \,|\, y_j = 0,\ j = 1, \ldots, d\}$ is the number of directions along which $p(x)$ is zero

- order of the zeros of order 1 must be improved by smoothing
- ω-Richardson smoother is used for this purpose

$$\begin{aligned} s_{i,\omega} : [-\pi, \pi)^d &\to \mathbb{C} \\ x \to s_{i,\omega}(x) &= 1 - \omega f_i(x) \end{aligned}$$

- e.g., for 1-level case with $g = 2$ we choose $\omega = 1/f(\pi)$

Convergence result

Lemma (B., Donatelli, Huckle [4])

Assume that $f_i \geq 0$ has a single isolated zero of order 2 at the origin and that f_i obtains the maximum only at all $y \in \mathcal{M}(0)$ lying on the axes and let $\tilde{y}$ be one of these points. Further, let

$$a_d(x) = \prod_{j=1}^{d} (1 + \mathrm{e}^{-\mathrm{i}x_j}), \qquad x \in [-\pi, \pi)^d.$$

Then the symbol of the smoothed prolongation given by

$$p_i(x) = s_{i,1/f(\tilde{y})}(x)\, a_d(x)$$

fulfills (1) and (2).

Multigrid methods with block-sized aggregation

To not deal with block Toeplitz matrices on the coarser levels, block-sized aggregation can be used on the finest level. For that purpose we define

$$P^d_{n,k} = I_n \otimes q_{\bar{j}}(\theta_0). \tag{5}$$

Under the same assumptions as before we can show the approximation property:

Theorem (B., Donatelli, Ferrari, Furci [3])

Consider the matrix $A_N := \mathcal{C}_n(\mathbf{f})$, where $\mathbf{f}$ is a $d \times d$ matrix-valued trigonometric polynomial that satisfies (3). Let $P^d_{n,k}$ be the projection operator defined in (5). Then

$$\exists \gamma > 0 \; : \; \min_{y \in \mathbb{C}^k} \|x_N - P^d_{n,k} y\|_2^2 \leq \gamma \|x_N\|^2_{A_N}, \qquad \forall x_N \in \mathbb{C}^N.$$

From the second level onward the matrix is scalar and multigrid convergence follows using the well-established theory.

Smoothing property for block Jacobi

For block Toeplitz matrices it is natural to consider block Jacobi as a smoother. Considering $A_N := \mathcal{C}_n(\mathbf{f})$ the iteration matrix of relaxed block Jacobi is given by

$$V_{n,\text{post}} = I_N - \omega D_B^{-1} A_N, \tag{6}$$

where D_B is block diagonal with the same block diagonal as A_N, i.e.

$$D_B = I_N \otimes \hat{\mathbf{f}}_0 = \mathcal{C}_n(\hat{\mathbf{f}}_0).$$

Theorem (B., Donatelli, Ferrari, Furci [3])

Let $A_N := \mathcal{C}_n(\mathbf{f})$, where $\mathbf{f}$ is a $d \times d$ matrix-valued trigonometric polynomial, $\mathbf{f} \geq 0$. and let $V_{n,\text{post}}$ given by (6). If

$$0 < \omega < \frac{2}{\|\hat{\mathbf{f}}_0^{-\frac{1}{2}} \mathbf{f} \hat{\mathbf{f}}_0^{-\frac{1}{2}}\|_\infty},$$

then there exists an $\alpha_\text{post} > 0$ s.t. $\|V_{n,\mathrm{post}} x_N\|_{A_N}^2 \leq \|x_N\|_{A_N}^2 - \alpha_\mathrm{post} \|x_N\|_{A_N^2}^2$ for all $x_N \in \mathbb{C}^N$.

Performance of aggregation plus block Jacobi for $C_n(\mathbf{f}_{\mathbb{Q}_d})$

- Varying block size d
- 1 iteration of damped block Jacobi as post smoother using $\omega = \frac{1}{2}$
- Matrix size given by $N = d \cdot 2^t$, $t = 15, \ldots, 20$

t	TGM			V-cycle		
	$d=2$	$d=4$	$d=8$	$d=2$	$d=4$	$d=8$
15	37	64	121	48	84	155
16	37	64	121	48	84	155
17	37	64	121	48	84	155
18	37	64	121	48	84	155
19	37	64	121	48	84	155
20	37	64	121	48	84	155

Outline

Multigrid

- Multigrid
- Parallel multigrid

Multgrid for structured matrices

- Multigrid for Toeplitz and circulant matrices
- Multigrid for symmetric positive definite block matrices
- Aggregation-based multigrid for Toeplitz matrices
- Block smoothers in multigrid methods for Toeplitz matrices

Advanced smoothing techniques

- Block smoothers
- Smoothers for systems of PDEs

Asynchrony in multigrid methods

Conclusions

Improving parallel scalability

- Reducing $\log$-term by aggressive coarsening
- Aggressive coarsening leads to smaller coarse space
  $\rightsquigarrow$ better smoothers necessary

Improving parallel scalability

- Reducing $\log$-term by aggressive coarsening
- Aggressive coarsening leads to smaller coarse space
  $\rightsquigarrow$ better smoothers necessary

Feasible better smoothers:

- Polynomial smoothers
- Block smoothers

Improving parallel scalability

- Reducing $\log$-term by aggressive coarsening
- Aggressive coarsening leads to smaller coarse space
  $\rightsquigarrow$ better smoothers necessary

Feasible better smoothers:

- Polynomial smoothers
- Block smoothers

Polynomial smoothers

- Use polynomials $p(A)$ as approximate inverse
- Are often used when aggressive coarsening is used
- Are easy to implement
- Need $\mathrm{degree}(p)$ communication steps
  (can be reduced to 1 if more involved implementation is used)

Improving parallel scalability

- Reducing $\log$-term by aggressive coarsening
- Aggressive coarsening leads to smaller coarse space
  $\rightsquigarrow$ better smoothers necessary

Feasible better smoothers:

- Polynomial smoothers
- Block smoothers

Block smoothers

- Consider only a small subset of unknowns
- Relax residuals by solving reduced systems
- Subset is determined with the help of geometric information
- Are easy to implement
- Need only one communication step

Block smoothers on GPUs

B. , Letterer [5]

- 3d 7-point Laplace with Dirichlet boundary
- Multicolor block Gauss-Seidel implemented using OpenCL
- Local blocks solved using 10 steps of Gauss-Seidel
- NVIDIA Tesla M2050 of JUDGE at Jülich Supercomputing Centre

n	block size	time/iter.	#iterations	performance
$2^5 - 1$	$2 \times 2 \times 2$	0.156 s	3	4.76 GFLOPS
$2^6 - 1$	$2 \times 2 \times 2$	0.322 s	3	18.50 GFLOPS
$2^7 - 1$	$2 \times 2 \times 2$	1.129 s	3	42.25 GFLOPS
$3^5 - 1$	$3 \times 3 \times 3$	4.222 s	3	70.28 GFLOPS
$4^3 - 1$	$4 \times 4 \times 4$	0.211 s	4	25.11 GFLOPS
$4^4 - 1$	$4 \times 4 \times 4$	4.765 s	4	71.22 GFLOPS
$6^3 - 1$	$6 \times 6 \times 6$	2.673 s	5	75.43 GFLOPS

ExaStencils

SPPEXA

supported by

DFG

Block smoothers on GPUs

B. , Letterer [5]

- 3d 7-point Laplace with Dirichlet boundary
- Multicolor block Gauss-Seidel implemented using OpenCL
- Local blocks solved using 10 steps of Gauss-Seidel
- NVIDIA Tesla M2050 of JUDGE at Jülich Supercomputing Centre

n	block size	time/iter.	#iterations	performance
$2^5 - 1$	$2 \times 2 \times 2$	0.156 s	3	4.76 GFLOPS
$2^6 - 1$	$2 \times 2 \times 2$	0.322 s	3	18.50 GFLOPS
$2^7 - 1$	$2 \times 2 \times 2$	1.129 s	3	42.25 GFLOPS
$3^5 - 1$	$3 \times 3 \times 3$	4.222 s	3	70.28 GFLOPS
$4^3 - 1$	$4 \times 4 \times 4$	0.211 s	4	25.11 GFLOPS
$4^4 - 1$	$4 \times 4 \times 4$	4.765 s	4	71.22 GFLOPS
$6^3 - 1$	$6 \times 6 \times 6$	2.673 s	5	75.43 GFLOPS

ExaStencils

SPPEXA

supported by

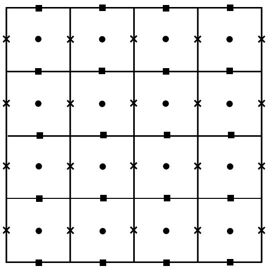
DFG

Smoothers for systems of PDEs

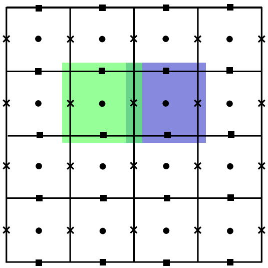
- Systems of PDEs require simultaneous relaxation of unknowns
- Consider the Stokes equations

$$\begin{aligned} \Delta u + \nabla p &= f, && \text{(momentum equation)} \\ \nabla \cdot u &= 0 && \text{(continuity equation)} \end{aligned}$$

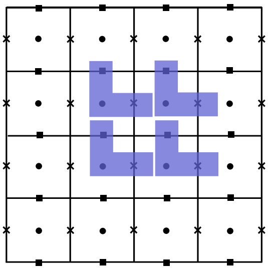
- Often overlapping smoothers are used
- We consider a non-overlapping one



Staggered discretization



Vanka smoother



Triad smoother

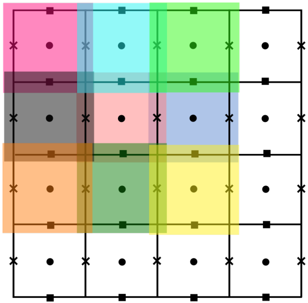
Parallel Implementation

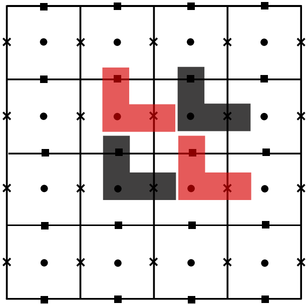
Vanka

- SOR method
- 9 coloring

Triad

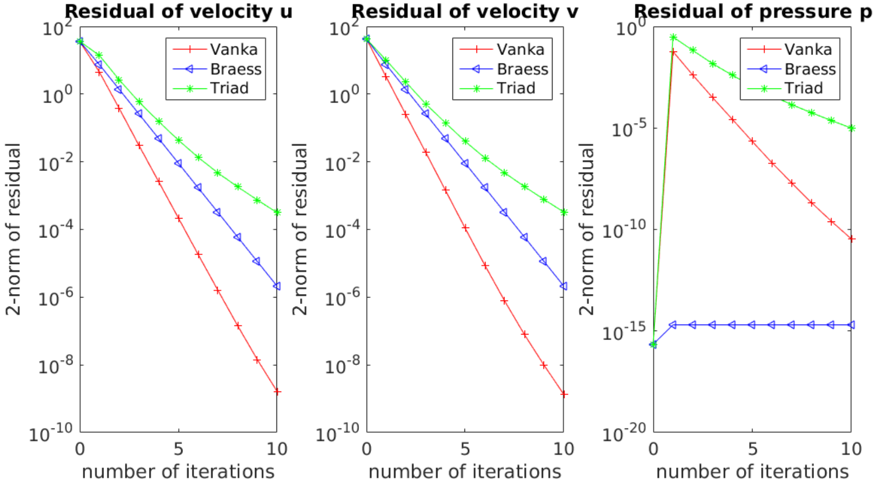
- SOR method
- Red-Black coloring





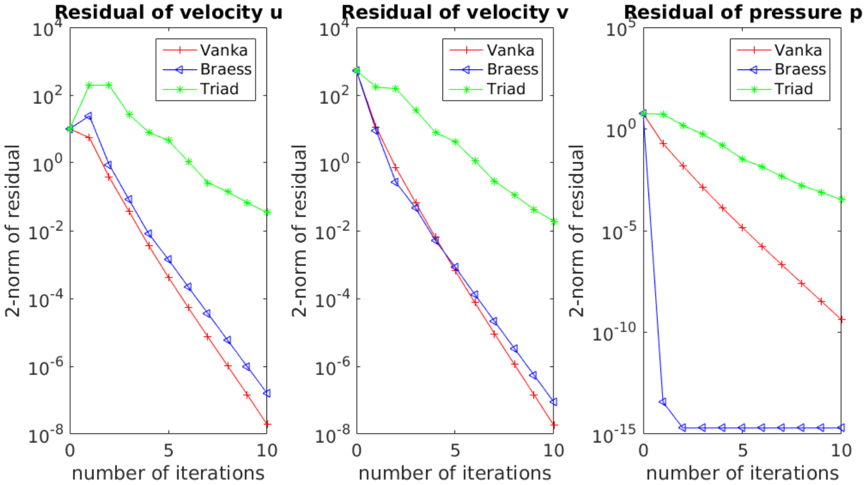
Convergence Behavior

Stokes problem with **periodic** boundary conditions



Convergence Behavior

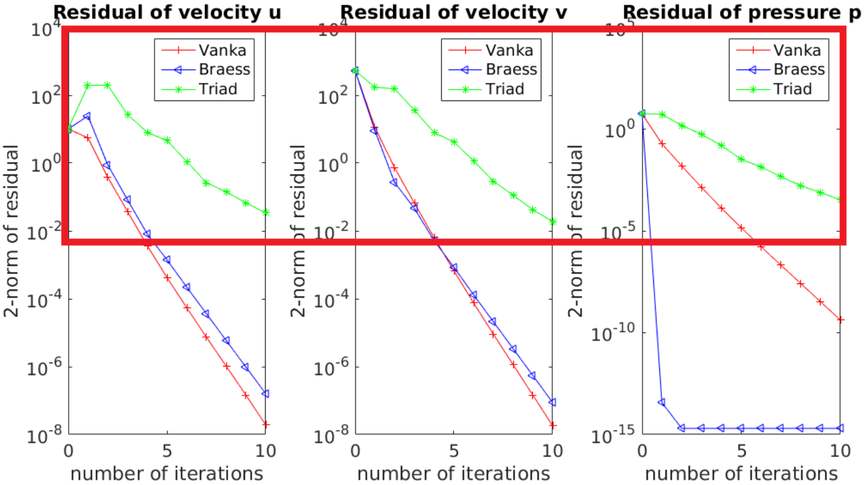
Stokes problem with **Dirichlet** boundary conditions



Convergence Behavior

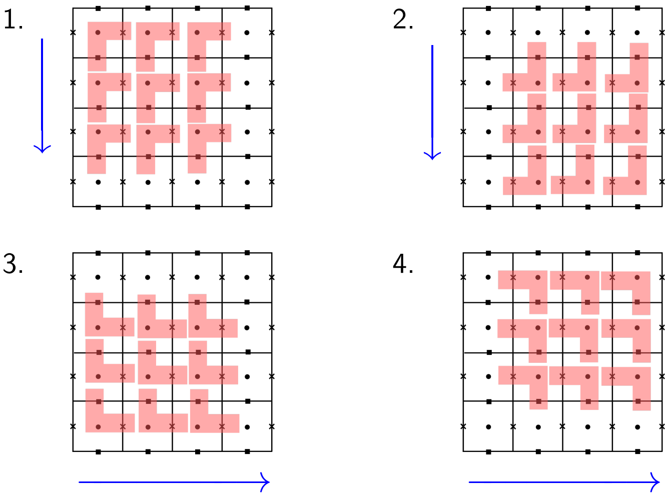
Stokes problem with **Dirichlet** boundary conditions

Triad: inefficient, but cheap



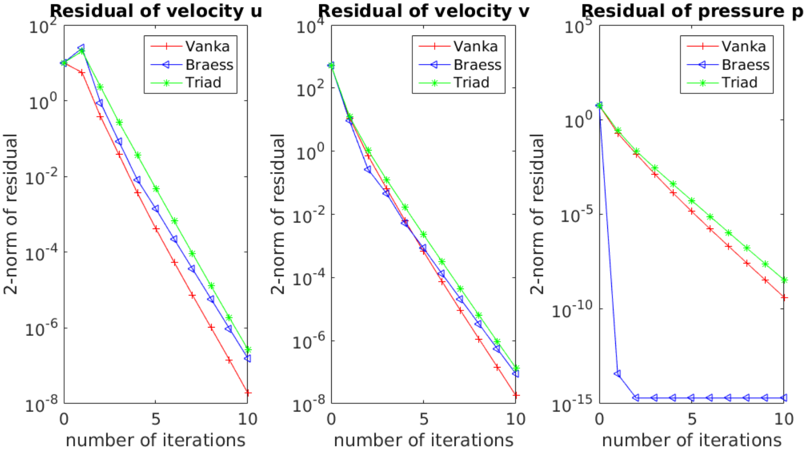
Improvement of Triad smoother

B., Claus [6]



Convergence Behavior

- Stokes problem with **Dirichlet** boundary conditions
- Triad smoother with alternating directions



Outline

Asynchrony in multigrid methods

Asynchrony can be used in multigrid methods in two ways:

1. Using asynchronous methods as smoothers, only
2. Extending the ideas of asynchronous methods to the full multilevel method

Asynchrony in multigrid methods

Asynchrony can be used in multigrid methods in two ways:

1. Using asynchronous methods as smoothers, only
2. Extending the ideas of asynchronous methods to the full multilevel method

The first is usually straightforward and even when employed only on the finest level parallel multigrid methods can benefit, as a vast majority (and synchronization...) of the work is carried out on this level.

Asynchrony in multigrid methods

Asynchrony can be used in multigrid methods in two ways:

1. Using asynchronous methods as smoothers, only
2. Extending the ideas of asynchronous methods to the full multilevel method

The first is usually straightforward and even when employed only on the finest level parallel multigrid methods can benefit, as a vast majority (and synchronization...) of the work is carried out on this level.

The latter usually is based on additive variants of multigrid, that come with an increased number of iterations.

Asynchronous smoothers multigrid methods

- A lot of the asynchronous methods (to be?) discussed here can be used as smoothers in multigrid methods
- A GPU implementation of Jacobi is presented by Tsai et al. in [13]
- Wolfson-Pou and Chow in [15] consider asynchronous semi-iterative methods and the asynchronous Chebyshev method, also in conjunction with multigrid
- ...

Fully asynchronous multigrid methods

- Already in 1989 Hart and McCormick [8] and McCormick and Quinlan [10] propose an asynchronous variant of FAC, the *fast adaptive composite grid method*
- Vassilevski and Yang in [14] propose an additive communication-minimizing variant of AMG that was implemented by AlOnazi et al. [1] where they report speedup despite almost doubled iteration counts due to going from multiplicative to additive
- Chaotic multigrid by Hawkes et al. [9] uses *chaotic cylces* based on *sawtooth cylces*, employing chaotic relaxation methods; authors report a 13.3x speedup
- Hahne et al. in [7] propose overlapping grids that are solved asynchronously specifically for the parallel-in-time method *multigrid-reduction-in-time*
- ...

Outline

Conclusions

- Parallelization of multigrid is possible
- Performance of (parallel) multigrid can be improved on modern architectures by use of block methods
- Analysis of multigrid for structured matrixes, as they arise from the discretization of PDEs on structured meshes, provides a lot of insight
- Results for scalar matrices can be transfered to the block case, as necessary for higher order discretizations and systems of PDEs
- Asynchrony can be used on the level of smoothing or on the whole hierarchy, being beneficial in both cases

References I

[1] Amani AlOnazi, George S. Markomanolis, and David Keyes. "Asynchronous Task-Based Parallelization of Algebraic Multigrid". In: *Proceedings of the Platform for Advanced Scientific Computing Conference*. PASC '17. Lugano, Switzerland: Association for Computing Machinery, 2017. DOI: `10.1145/3093172.3093230`.

[2] M. Bolten, M. Donatelli, P. Ferrari, and I. Furci. "A symbol based analysis for multigrid methods for block-circulant and block-Toeplitz Systems". In: *SIAM J. Matrix Anal. Appl.* 43.1 (2022), pp. 405–438. DOI: `10.1137/21M1390554`.

[3] M. Bolten, M. Donatelli, P. Ferrari, and I. Furci. "Analysis on aggregation and block smoothers in multigrid methods for block Toeplitz linear systems". In: *SIAM J. Matrix Anal. Appl.* 46.2 (2025), pp. 1416–1443. DOI: `10.1137/24M1655147`.

[4] M. Bolten, M. Donatelli, and T. Huckle. "Analysis of smoothed aggregation multigrid methods based on Toeplitz matrices". In: *Electron. Trans. Numer. Anal.* 44 (2015), pp. 25–52. URL: `http://etna.mcs.kent.edu/vol.44.2015/pp25-52.dir/pp25-52.pdf`.

[5] M. Bolten and O. Letterer. "Increasing arithmetic intensity in multigrid methods on GPUs using block smoothers". In: *PPAM 2015, Part I*. Ed. by R. Wyrzykowski, E. Deelman, J. Dongarra, K. Karczewski, J. Kitowski, and K. Wiatr. Vol. 9573. LNCS. Springer, 2016, pp. 515–525. DOI: `10.1007/978-3-319-32149-3_48`.

[6] L. Claus and M. Bolten. "Non-overlapping block smoothers for the Stokes equations". In: *Num. Lin. Alg. Appl.* 28.6 (2021), e2389. DOI: `10.1002/nla.2389`.

[7] Jens Hahne, Ben S. Southworth, and Stephanie Friedhoff. "Asynchronous truncated multigrid-reduction-in-time". In: *SIAM J. Sci. Comput.* 45.3 (2023), s281–s306. DOI: `10.1137/21M1433149`.

References II

[8] L. Hart and S. McCormick. “Asynchronous multilevel adaptive methods for solving partial differential equations on multiprocessors: Basic ideas”. In: *Parallel Comput.* 12.2 (1989), pp. 131–144. DOI: `10.1016/0167-8191(89)90048-3`.

[9] J. Hawkes, G. Vaz, A.B. Phillips, C.M. Klaij, S.J. Cox, and S.R. Turnock. “Chaotic multigrid methods for the solution of elliptic equations”. In: *Computer Physics Communications* 237 (2019), pp. 26–36. DOI: `10.1016/j.cpc.2018.10.031`.

[10] S. McCormick and D. Quinlan. “Asynchronous multilevel adaptive methods for solving partial differential equations on multiprocessors: Performance results”. In: *Parallel Comput.* 12.2 (1989), pp. 145–156. DOI: `10.1016/0167-8191(89)90049-5`.

[11] A. Napov and Y. Notay. “When does two-grid optimality carry over to the V-cycle”. In: *Num. Lin. Alg. Appl.* 17.2–3 (2010), pp. 273–290.

[12] J. W. Ruge and K. Stüben. “Algebraic multigrid”. In: *Multigrid methods*. Ed. by S. F. McCormick. Vol. 3. Frontiers Appl. Math. Philadelphia: SIAM, 1987, pp. 73–130.

[13] Yu-Hsiang Mike Tsai, Pratik Nayak, Edmond Chow, and Hartwig Anzt. “Implementing Asynchronous Jacobi Iteration on GPUs”. In: *2022 IEEE/ACM Workshop on Latest Advances in Scalable Algorithms for Large-Scale Heterogeneous Systems (ScalAH)*. 2022, pp. 1–9. DOI: `10.1109/ScalAH56622.2022.00006`.

[14] Panayot S. Vassilevski and Ulrike Meier Yang. “Reducing communication in algebraic multigrid using additive variants”. In: *Numerical Linear Algebra with Applications* 21.2 (2014), pp. 275–296. DOI: `10.1002/nla.1928`.

References III

[15] Jordi Wolfson-Pou and Edmond Chow. “Asynchronous Semi-iterative Methods and the Asynchronous Chebyshev Method with Multigrid Preconditioning”. In: *SIAM Journal on Scientific Computing* (), S23–S49. DOI: `10.1137/24M1669669`.