Robust multigrid methods for large-scale matrix equations

Workshop *Matrix Equations*, Chemnitz

Bart Vandereycken, Stefan Vandewalle

Department of Computer Science
Katholieke Universiteit Leuven, Belgium

June 15, 2007

Model problem

Consider an elliptic second order PDE

$$\mathcal{L}u := -\sum_{i,j=1}^{d} \frac{\partial}{\partial x_j}\left(a_{ij}(x)\frac{\partial u}{\partial x_i}\right) + \sum_{i=1}^{d} b_i(x)\frac{\partial u}{\partial x_i} + c(x)u = f(x), \quad x \in \Omega$$

discretized by **A**.
We want to solve the corresponding Lyapunov equation

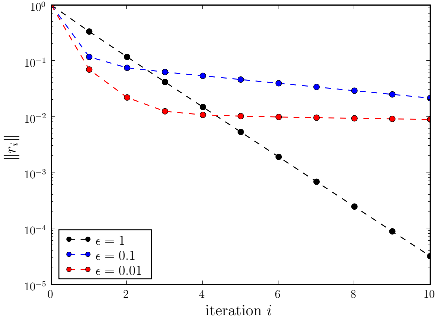
$$\mathbf{AX} + \mathbf{XA}^T + \mathbf{F} = 0, \quad \text{low rank } \mathbf{F}$$

Model problem for $\mathcal{L}$: anisotropic 2d diffusion

$$-\frac{\partial^2 u}{\partial x_1^2} - \epsilon\frac{\partial^2 u}{\partial x_2^2} = f(x_1, x_2), \qquad \epsilon \to 0$$

Observation

(LR-)MG with Jacobi does not work for $-u_1 - \epsilon u_2$ if $\epsilon \to 0$.



Approx. rank of solution and κ stay the same
ADI, KPIK have no problem

Solution

Well known problem for MG

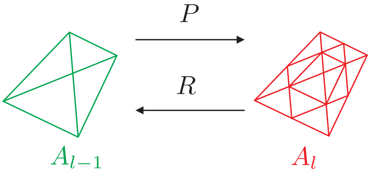
- Fix grids
  Choose grids so that anisotropy is eliminated and Jacobi works again
  $\rightsquigarrow$ Semi-coarsening, MG-S, AMG
- Fix smoothers
  Keep grids but change smoothers
  $\rightsquigarrow$ line(block) smoothers, GS

Better smoothers (RB-GS) also give you a faster MG.

Multigrid

Solve $\mathbf{A}\boldsymbol{x} = \boldsymbol{f}$

Cycling between
fine and coarse grids



$\mathbf{A} : \mathbf{A}_l, \mathbf{A}_{l-1}, \mathbf{P}, \mathbf{R}$, smoother

$Px_{l-1} = x_l$ $\quad Rx_l = x_{l-1}$

- Smoothing error $\boldsymbol{e}_l$ on fine grid
  gives high frequency components
  cheap solve of $\mathbf{A}_l\boldsymbol{x}_l = \boldsymbol{f}_l \rightsquigarrow$ Richardson, Jacobi, ...

- Error correction on coarse grid
  gives low frequency components
  cheap solve of $\mathbf{A}_{l-1}\boldsymbol{e}_{l-1} = \boldsymbol{d}_{l-1} \rightsquigarrow \gamma$ recursive MG

Tensor product MG

$$\mathbf{A}_l \boldsymbol{x} + \boldsymbol{f} = 0$$

1. $\boldsymbol{x}_1 \leftarrow \mathtt{smooth}^{\nu}(\boldsymbol{x}_0, \boldsymbol{f})$
2. $\boldsymbol{d} \leftarrow \mathbf{A}_l \boldsymbol{x}_1 + \boldsymbol{f}$
3. $\bar{\boldsymbol{f}} \leftarrow \mathbf{R}(\boldsymbol{d})$
4. $\bar{\boldsymbol{x}} \leftarrow \mathbf{A}_{l-1}^{-1} \bar{\boldsymbol{f}}$
5. $\boldsymbol{x}_2 \leftarrow \boldsymbol{x}_1 + \mathbf{P}\bar{\boldsymbol{x}}$

Tensor product MG

$$\mathbf{A}_l \boldsymbol{x} + \boldsymbol{f} = 0$$

1. $\boldsymbol{x}_1 \leftarrow \mathtt{smooth}^{\nu}(\boldsymbol{x}_0, \boldsymbol{f})$
2. $\boldsymbol{d} \leftarrow \mathbf{A}_l \boldsymbol{x}_1 + \boldsymbol{f}$
3. $\bar{\boldsymbol{f}} \leftarrow \mathbf{R}(\boldsymbol{d})$
4. $\bar{\boldsymbol{x}} \leftarrow \mathbf{A}_{l-1}^{-1} \bar{\boldsymbol{f}}$
5. $\boldsymbol{x}_2 \leftarrow \boldsymbol{x}_1 + \mathbf{P}\bar{\boldsymbol{x}}$

$$\mathbf{A}_l \mathbf{X} + \mathbf{X}\mathbf{A}_l^T + \mathbf{F} = 0$$

1. $\mathbf{X}_1 \leftarrow \mathtt{smooth}^{\nu}(\mathbf{X}_0, \mathbf{F})$
2. $\mathbf{D} \leftarrow \mathbf{A}_l \mathbf{X}_1 + \mathbf{X}_1 \mathbf{A}_l^T + \mathbf{F}$
3. $\bar{\mathbf{F}} \leftarrow \mathcal{R}(\mathbf{D})$
4. $\bar{\mathbf{X}} \leftarrow (\mathbf{A}_{l-1} \bullet + \bullet\, \mathbf{A}_{l-1}^T + \bar{\mathbf{F}})$
5. $\mathbf{X}_2 \leftarrow \mathbf{X}_1 + \mathcal{P}\bar{\mathbf{X}}$

Framework Hackbusch

Smoothing property

$$\lim_{\nu\to\infty} \sup_{l} \frac{\|\mathbf{A}_l \mathbf{S}_l^{\nu}\|}{\|\mathbf{A}_l\|} = 0$$

Approximation property

$$\sup_{l} \|\mathbf{A}_l^{-1} - \mathbf{P}\mathbf{A}_{l-1}^{-1}\mathbf{R}\| \|\mathbf{A}_l\| = C_A$$

$\rightsquigarrow$ TG and W-cycle mesh-independent convergence

Anisotropy: $C_A \sim 1/\epsilon$
point-smoothers: $\sup_l \frac{\|\mathbf{A}_l \mathbf{S}_l^{\nu}\|}{\|\mathbf{A}_l\|} \not\sim \epsilon$
block-smoothers: $\sup_l \frac{\|\mathbf{A}_l \mathbf{S}_l^{\nu}\|}{\|\mathbf{A}_l\|} \sim \epsilon$

Local Fourier Analysis

$\mathbf{A}_h$ is an operator on an infinite grid

$$\boldsymbol{G}_h = \{\boldsymbol{x} = (x_1, \ldots, x_d) = \boldsymbol{\kappa}\boldsymbol{h} = (\kappa_1 h_1, \ldots, \kappa_d h_d), \boldsymbol{\kappa} \in \mathbb{Z}^d\}.$$

with constant coefficients.
The formal eigenfunctions or Fourier modes

$$\varphi(\boldsymbol{\theta}, \boldsymbol{x}) = e^{\mathrm{i}\boldsymbol{\theta}\boldsymbol{x}/\boldsymbol{h}} = e^{\mathrm{i}\theta_1 x_1/h_1} \cdots e^{\mathrm{i}\theta_1 x_d/h_d} \qquad \text{for } \boldsymbol{x} \in \boldsymbol{G}_h$$

with formal eigenvalue or symbol

$$\mathbf{A}_h \varphi(\boldsymbol{\theta}) = \widetilde{A}_h(\boldsymbol{\theta}) \varphi(\boldsymbol{\theta})$$

Heuristic but very good predictions of MG convergence.

Tensor LFA

Tensor structure of $\mathbf{A} \oplus \mathbf{A}$ also in LFA

- Frequency components

$$\varphi(\theta, x) \to \varphi((\theta_x, \theta_y), (x, y)), \quad x \in G_h \to (x, y) \in G_h \otimes G_h$$

- $2h$-harmonics: $\theta^{(0,0)}\theta^{(1,0)}$
- Symbols

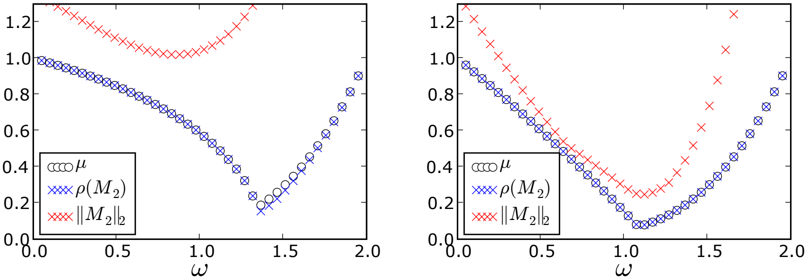
$$\widetilde{R}_h(\theta^\alpha) \to \widetilde{R}_h(\theta_x^{\alpha_x})\widetilde{R}_h(\theta_y^{\alpha_y})$$

- Block-diagonal

$$\cdots \begin{bmatrix} \times & & & \\ & \times & & \\ & & \times & \\ & & & \times \end{bmatrix} - \begin{bmatrix} \times \\ \times \\ \times \\ \times \end{bmatrix} [\times] \cdots \to \cdots \begin{bmatrix} \times & & & \\ & \times & & \\ & & \times & \\ & & & \times \end{bmatrix} \otimes \begin{bmatrix} \times & & & \\ & \times & & \\ & & \times & \\ & & & \times \end{bmatrix} - \begin{bmatrix} \times \\ \times \\ \times \\ \times \end{bmatrix} \otimes \begin{bmatrix} \times \\ \times \\ \times \\ \times \end{bmatrix} [\times] \oplus [\times] \cdots$$

LFA results

Anisotropic Poisson $-u_{x_1x_1} - \epsilon u_{y_1y_1} - u_{x_2x_2} - \epsilon u_{y_2y_2}$, $\epsilon = 0.3$



std.coarsening

x_1x_2-coarsening
(x_1x_2-hyperplane smoothing)

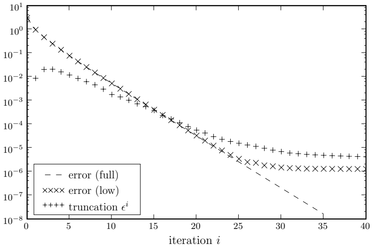
RB G-S, FW,Linear interp., 1 pre, 1 post

Perturbed MG

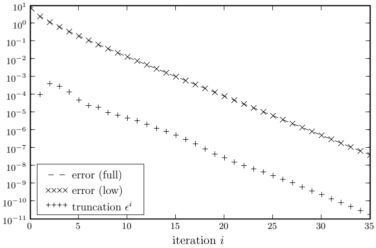
1. $\widetilde{\mathbf{X}_1} \leftarrow \mathcal{T}_k(\mathtt{smooth}^{\nu_1}(\widetilde{\mathbf{X}_0}, \mathbf{F}))$ $\rightsquigarrow \epsilon_s$
2. $\widetilde{\mathbf{D}} \leftarrow \mathcal{T}_k(\mathbf{A}_l\widetilde{\mathbf{X}_1} + \widetilde{\mathbf{X}_1}\mathbf{A}_l^T + \mathbf{F})$ $\rightsquigarrow \epsilon_d$
3. $\bar{\mathbf{F}} \leftarrow \mathcal{R}\widetilde{\mathbf{D}}$
4. $\widetilde{\bar{\mathbf{X}}} \leftarrow \mathcal{T}_k(\mathtt{cgc}(\mathbf{A}_{l-1}, \bar{\mathbf{F}}))$ $\rightsquigarrow \epsilon_c$
5. $\widetilde{\mathbf{X}_2} \leftarrow \mathcal{T}_k(\mathbf{X}_1 + \mathcal{P}\widetilde{\bar{\mathbf{X}}})$ $\rightsquigarrow \epsilon_e$

$$
\begin{aligned}
\|e_2\| &\leq \|\mathrm{TG}\|\|e_0\| \\
&\quad + \|\mathcal{I} - \mathcal{P}\bar{\mathcal{S}}^{-1}\mathcal{R}\mathcal{S}\|\epsilon_s + \|\mathcal{P}\bar{\mathcal{S}}^{-1}\mathcal{R}\|\epsilon_d + \|\mathcal{P}\|\epsilon_c + \epsilon_e \\
&\leq \|\mathrm{TG}\|\|e_0\| + \epsilon_{\mathrm{TG}}
\end{aligned}
$$

Truncation error due to low rank iterations for isotropic Poisson (1,1) Jacobi smoother.



rank 4



rank 10

Conclusion

You can

- predict and proof mesh-independent convergence of MG (mature literature).
  $\rightsquigarrow$ complexity of $\mathcal{O}(n^2)$
- control the errors in LR-MG and keep convergence of MG.
  $\rightsquigarrow$ complexity of $\mathcal{O}(n \log^c n)$

Implementation of LR-MG

How can we use MG components for the tensor problem and apply them in a *low rank* setting?

- Intergrid transfers $\mathbf{P}, \mathbf{R}$:
  we can do $\mathbf{P} \otimes \mathbf{P}$ to a LR matrix $\mathbf{YZ}^T$ as

$$(\mathbf{P} \otimes \mathbf{P})\mathbf{YZ}^T = (\mathbf{PY})(\mathbf{PZ})^T.$$

- Coarse grid operator $\mathbf{A}_{i-1}$:
  tensorised application of the FDM of $\mathbf{A}_{i-1}$ gives

$$\mathbf{A}_{i-1} \otimes \mathbf{I} + \mathbf{I} \otimes \mathbf{A}_{i-1}.$$

- Smoothing:
  more *complicated*.

LR smoother

Richardson iteration

$$\boldsymbol{x}_{i+1} = \boldsymbol{x}_i - \theta(\mathbf{A}\boldsymbol{x}_i + \boldsymbol{f})$$
$$\mathbf{X}_{i+1} = \mathcal{T}_k\left(\mathbf{X}_i - \theta(\mathbf{A}\mathbf{X}_i + \mathbf{X}_i\mathbf{A}^T + \mathbf{F})\right)$$

how to do Gauss-Seidel, block-smoothers, ...

Typical MG smoothers for $\mathbf{A}$ are based on a splitting:

$$\mathbf{A} = \mathbf{W} - \mathbf{R}$$

gives the iteration

$$\mathbf{W}\boldsymbol{u}_{i+1} = \mathbf{R}\boldsymbol{u}_i + \boldsymbol{f} \qquad \mathbf{S} = \mathbf{W}^{-1}\mathbf{R}$$

We propose a similar splitting for $\mathbf{AX} + \mathbf{XA}^T$

$$\mathbf{A} \otimes \mathbf{I} + \mathbf{I} \otimes \mathbf{A} = (\mathbf{W} \otimes \mathbf{I} + \mathbf{I} \otimes \mathbf{W}) - (\mathbf{R} \otimes \mathbf{I} + \mathbf{I} \otimes \mathbf{R})$$

gives

$$\mathbf{W}\mathbf{U}_{i+1} + \mathbf{U}_{i+1}\mathbf{W}^T = \mathbf{R}\mathbf{U}_i + \mathbf{U}_i\mathbf{R}^T + \mathbf{F}$$

The Lyapunov equation of the smoother

$$\mathbf{W}\mathbf{U}_{i+1} + \mathbf{U}_{i+1}\mathbf{W}^T = \mathbf{D}$$

is *supposed* to be easy with a well chosen splitting!

- solve $\mathbf{U}_{i+1}$ as a low rank matrix

The Lyapunov equation of the smoother

$$\mathbf{W}\mathbf{U}_{i+1} + \mathbf{U}_{i+1}\mathbf{W}^T = \mathbf{D}$$

is *supposed* to be easy with a well chosen splitting!

- solve $\mathbf{U}_{i+1}$ as a low rank matrix

$\rightsquigarrow$ use existing LR Lyapunov solvers to *solve the smoother*, e.g. CF-ADI

- a lot better than solving $\mathbf{AX} + \mathbf{XA}^T$ directly
- only *one, two* ADI iterations

Smoothers solved by ADI

Solve

$$\mathbf{W}\mathbf{U}_{i+1} + \mathbf{U}_{i+1}\mathbf{W}^T = \mathbf{D}$$

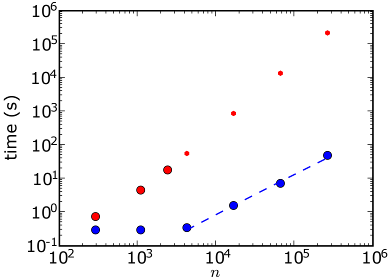
by CF-ADI:

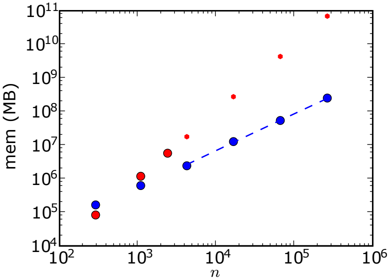
smoother	**W**	spectrum	shift-inv.	nb. ADI
Jacobi	**D**	+	+	+
Lex.GS	**L**	+ (non-normal)	+	?
RB Jac/GS	2 **D/L**	+	+	+/?
ILU	2 **L**	?	-	+
Line-Jac	tri-diag.	symmetric: +	+	+/-

In the end: one or two very cheap ADI iterations

Nested iteration

FMG on $2d \times 2d$ Poisson: up to *discretisation* error





mesh of 513×513 in 50s
$\mathcal{O}(n \log^4(n))$

mesh of 513×513 in 250MB
$\mathcal{O}(n \log^2(n))$

RB G-S, FW,Linear interp., 1 pre, 1 post
3 sweeps / nested iteration

LR-MG vs. K-PIK

Anisotropic diffusion

- LR-MG with line Jac.: nb of its constant with n
- K-PIK: nb of its grows with n