Introduction

Large sparse discrete PDE systems \( A u = b \).

- Iterative methods accelerated by preconditioning: easily invertible \( B \) such that \( B \approx A \).
- Multigrid & multilevel methods: often very efficient.
- Basic principle (two-grid): obtain fast the convergence by solving a smaller problem, on a coarser grid.
- Recursive use: the coarse grid problem is solved using the same two-grid preconditioner.
- This seminar: emphasis on \textit{algebraic} methods (that work using only the information in \( A \)).

Outline

1. An introductory example.
3. The different schemes and their algebraic properties.
4. Algebraic interpolation.
5. Algebraic coarsening: standard from AMG and aggregation.
6. Checking & correcting the coarsening.
7. From two- to multi-level: cycling strategies.
8. Some numerical illustrations

Algebraic methods: field of application

- Robust for scalar elliptic PDEs with standard discretization.
- Emphasis on (theory for) symmetric problems (self-adjoint PDEs), but work in unsymmetric cases as well (e.g. convection diffusion problems).
- Ongoing research for systems of PDEs (efficient preconditioning of each diagonal block).
- Does not work well for indefinite problems (some eigenvalues with negative real part); e.g.: Helmholtz.

Remark: \textit{AMG} is the generic name of a family of methods, but also the specific name of Ruge & Stüben method.
An example

PDE: \(-\Delta u = 20 e^{-10((x-0.5)^2+(y-0.5)^2)}\) in \(\Omega = (0, 1) \times (0, 1)\)
\(u = 0\) on \(\partial\Omega\)

Uniform grid with mesh size \(h\), five-point finite difference.

Solution with \(h^{-1} = 50\)
Solution with \(h^{-1} = 25\)

An idea

Fine grid (system to solve):
\[ Au = b. \]

Coarse grid (auxiliary system):
\[ A_C u_C = b_C. \]

\(u_C\) may be computed and prolonged (by interpolation) on the fine grid:
\[ u^{(1)} = p u_C \]

\(u^{(1)}\) may serve as initial approximation, i.e., one solves
\[ A (u^{(1)} + x) = b \quad \text{or} \quad A x = b - A p A_C^{-1} b_C. \]

How it works

Error on the fine grid after interpolation

\[ \frac{|u - u^{(1)}|}{|u|} = 0.0019 \]

Let us repeat

\[ A (u^{(1)} + x) = b \quad \text{or} \quad A x = b - A p A_C^{-1} b_C = r^{(1)}. \]

(1) Restrict on the coarse grid:
\[ r_C = r r^{(1)}. \]

(2) Solve on the coarse grid:
\[ x_C^{(2)} = A_C^{-1} r_C. \]

(3) Prolongate:
\[ x^{(2)} = p x_C^{(2)}, \quad u^{(2)} = u^{(1)} + x^{(2)}. \]
Still working?

Error on the fine grid after interpolation

Repeating the process...

\[ \frac{\| u - u^{(1)} \|}{\| u \|} = 0.0019 \]

\[ \frac{\| u - u^{(2)} \|}{\| u \|} = 0.0018 \]

Explaination (for simplicity) that \( b_C = r b \).

One has

\[ u - u^{(1)} = u - p A_C^{-1} r b \]
\[ = (I - p A_C^{-1} r A) u, \]
\[ u - u^{(2)} = (I - p A_C^{-1} r A)^2 u, \]

etc. Similarly

\[ r^{(1)} = b - A p A_C^{-1} r b \]
\[ = (I - A p A_C^{-1} r) r^{(0)}. \]

\[ p A_C^{-1} r \) has rank \( n_C \)
\[ \rightarrow \]

\[ \rho(I - A p A_C^{-1} r) = \rho(I - p A_C^{-1} r A) \geq 1. \]

Smoothing enters the scene

\( u - u^{(1)} \) and \( r^{(1)} \) very oscillatory
\[ \rightarrow \]

improve \( u^{(1)} \) with a simple iterative method, efficient in smoothing the error & residual.

Example: symmetric Gauss-Seidel (SGS)

\[ L u^{(1+1/2)} = b - (A - L) u^{(1)}, \quad (L = \text{low}(A)) \]
\[ U u^{(2)} = b - (A - U) u^{(1+1/2)}, \quad (U = \text{upp}(A)) \]

Same as

\[ u^{(2)} = u^{(1)} + M^{-1} r^{(1)}, \quad M = L D^{-1} U \quad (D = \text{diag}(A)) \]

Thus:

\[ u - u^{(2)} = (I - M^{-1} A) (u - u^{(1)}) \]
\[ r^{(2)} = (I - A M^{-1}) r^{(1)} \]

One may repeat:

\[ r^{(m+1)} = (I - A M^{-1})^m r^{(1)}. \]
**Smoothing effect**

Residual after CG correction

Adding 1 SGS step

Adding 3 SGS steps

Adding 8 SGS steps

**What we learned**

For each coarse grid correction:

\[ \mathbf{u} - \mathbf{u}^{(m+1)} = (I - p \ A_C^{-1} r \ A) \ (\mathbf{u} - \mathbf{u}^{(m)}) . \]

Cannot work alone because \( \rho (I - p \ A_C^{-1} r \ A) \geq 1 \).

For each smoothing step

\[ \mathbf{u} - \mathbf{u}^{(m+1)} = (I - M^{-1} A) \ (\mathbf{u} - \mathbf{u}^{(m)}) . \]

Not efficient alone because \( \rho (I - M^{-1} A) \approx 1 \).

However

\[ \rho \left( (I - M^{-1} A) \ (I - p \ A_C^{-1} r \ A) \ (I - M^{-1} A) \right) \ll 1 \]

Rmk: if \( A = A^T \), we assume \( M = M^T \).

**Smoothing + coarse grid correction**

Adding now a CG correction . . . and again 1 SGS step

Adding 3 SGS steps

Adding 8 SGS steps

**How it works**

Initial residual

1 multigrid step

2 multigrid steps

4 multigrid steps

**Rmk:** if \( A = A^T \), we assume \( M = M^T \).
**Some remarks**

**Geometric multigrid**
- Simple in its principles.
- Complicate to analyze.
- Not robust: simple ideas not always lead to efficient schemes.
- There is a lot of research works on multigrid applications.

**Algebraic multigrid**
- More user friendly (“black box”).
- More robust.
- ... sacrificing somewhat on efficiency.

---

**Algebraic multigrid: ingredients**

- Coarsening: $F/C$ partitioning of the unknowns.
- Interpolation $J_{FC}$ and prolongation $p = \begin{pmatrix} J_{FC} \\ I \end{pmatrix}$ satisfying $p e_C = e$.
- For the restriction, one often takes $r = \beta \begin{pmatrix} J_{FC}^T \\ I \end{pmatrix} = \beta p^T$ with $\beta$ such that $r e = e_C$.
- For $A_C$ one may rely on the Galerkin approximation:

$$A_C = r A p \quad \text{or} \quad \hat{A}_C = p^T A p$$

with coarse grid correction given by $p \hat{A}_C^{-1} p^T$.

- $I - p \hat{A}_C^{-1} p^T A = \left( I - p \hat{A}_C^{-1} p^T A \right)^2$ (projector)

---

**Two-grid AMG as a preconditioner**

**AMG as preconditioner**

$v = B_{AMG}^{-1} r$ computed as

1. $t = M^{-1} r; \ w = r - A t$
2. $y_C = w_C + J_{FC}^T w_F$
3. Solve $\hat{A}_C z_C = y_C$
4. $z_F = J_{FC} z_C$
5. $v = t + z + M^{-1} (w - A z)$

$$I - B_{AMG}^{-1} A = \left( I - M^{-1} A \right) \left( I - p \hat{A}_C^{-1} p^T A \right) \left( I - M^{-1} A \right).$$

---

**Multilevel is not multigrid!**

**Coarse grid correction:**

$$p \hat{A}_C^{-1} p^T \quad \text{with} \quad p = \begin{pmatrix} J_{FC} \\ I \end{pmatrix}.$$  

Let’s try an additive complement

$$q Q_{FF}^{-1} q^T \quad \text{with} \quad q = \begin{pmatrix} I \\ 0 \end{pmatrix}.$$  

($Q_{FF} \approx q^T A q = A_{FF}$)

**Corresponding preconditioner:**

$$B_{HBBD}^{-1} = \begin{pmatrix} I & J_{FC} \\ J_{FC}^T & I \end{pmatrix} \begin{pmatrix} Q_{FF}^{-1} & \hat{A}_C^{-1} \\ \hat{A}_C & I \end{pmatrix} \begin{pmatrix} I \\ J_{FC} \end{pmatrix}.$$
Hierarchical finite element bases

Finite element matrices are better conditioned whenever expressed in the hierarchical basis.

\begin{align*}
J = \begin{pmatrix} I & J_{\text{FC}} \\ J_{\text{FC}}^T & I \end{pmatrix}
\end{align*}

performs the basis transformation \((\text{hb}_\text{tl}) \rightarrow (\text{nb})\) \((\text{hb}_\text{tl}): \text{coarse nodal basis} (2h) + \text{compl. functions} (h)\).

Matrix in this basis:

\begin{align*}
\widehat{A} = J^T AJ = \begin{pmatrix} I \\ J_{\text{FC}}^T \end{pmatrix} \begin{pmatrix} A_{\text{FF}} & A_{\text{FC}} \\ A_{\text{CF}} & A_{\text{CC}} \end{pmatrix} \begin{pmatrix} I \\ J_{\text{FC}} \end{pmatrix} \\
= \begin{pmatrix} A_{\text{FF}} \\ A_{\text{CF}} + J_{\text{FC}}^T A_{\text{FF}} \end{pmatrix} A_{\text{FC}} + A_{\text{FF}} J_{\text{FC}} A_{\text{CF}} A_{\text{CC}} A_{\text{FF}} \end{align*}

Additive two-level

Matrix in \((\text{hb}_\text{tl})\):

\[ \widehat{A} = J^T AJ. \]

Two-grid with additive complement:

\[ B_{\text{HBBBD}}^{-1} = \begin{pmatrix} I \\ J_{\text{FC}} \end{pmatrix} \begin{pmatrix} Q_{\text{FF}}^{-1} \\ A_{\text{CF}} \end{pmatrix} \begin{pmatrix} I \\ J_{\text{FC}}^T \end{pmatrix} = J B_{\text{HBBBD}}^{-1} J^T, \]

where

\[ B_{\text{HBBBD}} = \begin{pmatrix} Q_{\text{FF}} \\ A_{\text{CF}} \end{pmatrix} \approx \begin{pmatrix} A_{\text{FF}} \\ A_{\text{CF}} \end{pmatrix}, \]

which is the block diagonal part of \(\widehat{A}\). Further,

\[ B_{\text{HBBBD}}^{-1} A = \begin{pmatrix} J B_{\text{HBBBD}}^{-1} J^T \end{pmatrix} \begin{pmatrix} I \\ J_{\text{FC}} \end{pmatrix} J^{-1} = J B_{\text{HBBBD}}^{-1} \widehat{A} J^{-1}. \]

The strengthened C.B.S. constant

We assume \(A\) symmetric and positive definite.

Definition

\[ \widehat{\gamma} = \max_{v=(v_F)_0 \ne 0, w=(w_C)_0 \ne 0} \frac{|v^T \widehat{A} w|}{(v^T \widehat{A} v)^{1/2} (w^T \widehat{A} w)^{1/2}}. \]

Property. If \(\widehat{A} = \sum_{\ell} \widehat{A}_\ell\) and if, \(\forall \ell, \widehat{\gamma}_\ell\) is such that, for all \(v=(v_F)_0, w=(w_C)_0\)

\[ |v^T \widehat{A}_\ell w| \le \widehat{\gamma}_\ell (v^T \widehat{A}_\ell v)^{1/2} (w^T \widehat{A}_\ell w)^{1/2}, \]

then:

\[ \widehat{\gamma} \le \max_{\ell} \widehat{\gamma}_\ell, \]

\rightarrow \widehat{\gamma} \text{ may often be bounded away from } 1. \]
Two-grid HBBD preconditioning

Preconditioning by HBBD
\[ \mathbf{v} = B_{\text{HBBD}}^{-1} \mathbf{r} \] computed as
1. \( \mathbf{y}_F = Q_{FF}^{-1} \mathbf{r}_F \)
2. \( \mathbf{y}_C = \mathbf{r}_C + J_{FC}^T \mathbf{r}_F \)
3. Solve \( \hat{\mathbf{A}}_C \mathbf{v}_C = \mathbf{y}_C \)
4. \( \mathbf{z}_F = J_{FC} \mathbf{v}_C \)
5. \( \mathbf{v}_F = \mathbf{z}_F + \mathbf{y}_F \)

Let \( \mathbf{S}_A = \mathbf{A}_{CC} - \mathbf{A}_{CF} \mathbf{A}_{FF}^{-1} \mathbf{A}_{FC} \)
One has
\[ \lambda_{\min} \left( \hat{\mathbf{A}}_C^{-1} \mathbf{S}_A \right) = 1 - \delta^2, \]
\[ \lambda_{\max} \left( \hat{\mathbf{A}}_C^{-1} \mathbf{S}_A \right) \leq 1. \]

Two-level block factorization

\[ \hat{\mathbf{A}} = \begin{pmatrix} \mathbf{A}_{FF} & \hat{\mathbf{A}}_{FC} \\ \hat{\mathbf{A}}_{CF} & \hat{\mathbf{A}}_C \end{pmatrix}, \quad \hat{\mathbf{D}} = \begin{pmatrix} \mathbf{A}_{FF} & \hat{\mathbf{A}}_C \\ \hat{\mathbf{A}}_{CF} & \hat{\mathbf{A}}_C \end{pmatrix}, \]

One has
\[ \kappa \left( \hat{\mathbf{D}}^{-1} \hat{\mathbf{A}} \right) = \frac{\lambda_{\max}(\hat{\mathbf{D}}^{-1} \hat{\mathbf{A}})}{\lambda_{\min}(\hat{\mathbf{D}}^{-1} \hat{\mathbf{A}})} = \frac{1 + \gamma}{1 - \gamma}. \]

Two-level block factorization (cont.)

Preconditioning by HBBF
\[ \mathbf{v} = B_{\text{HBBF}}^{-1} \mathbf{r} \] computed as
1. \( \mathbf{y}_F = Q_{FF}^{-1} \mathbf{r}_F \)
2. \( \mathbf{y}_C = \mathbf{r}_C - \mathbf{A}_{CF} \mathbf{y}_F + J_{FC}^T (\mathbf{r}_F - \mathbf{A}_{FF} \mathbf{y}_F) \)
3. Solve \( \hat{\mathbf{A}}_C \mathbf{v}_C = \mathbf{y}_C \)
4. \( \mathbf{z}_F = J_{FC} \mathbf{v}_C \)
5. \( \mathbf{v}_F = \mathbf{z}_F + Q_{FF}^{-1} (\mathbf{r}_F - \mathbf{A}_{FC} \mathbf{v}_C - \mathbf{A}_{FF} \mathbf{z}_F) \)

Let \( \mathbf{B}^{-1}_{\text{HBBF}} \)
\[ \mathbf{B}^{-1}_{\text{HBBF}} = \begin{pmatrix} \mathbf{I} & J^T \\ \mathbf{I} & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{Q}_{FF}^{-1} & \hat{\mathbf{A}}_C^{-1} \\ \mathbf{Q}_{FF}^{-1} \hat{\mathbf{A}}_C^{-1} & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{I} & -Q_{FF}^{-1} \hat{\mathbf{A}}_C \\ \mathbf{I} & \mathbf{I} \end{pmatrix}. \]
Two-level hierarchical basis multigrid

Preconditioning by HBMG

\( \mathbf{v} = B_{\text{HBMG}}^{-1} \mathbf{r} \) computed as

1. \( \mathbf{y}_F = Q_{FF}^{-1} \mathbf{r}_F \)
2. \( \mathbf{y}_C = \mathbf{r}_C - \mathbf{A}_{CF} \mathbf{y}_F + J_{FC}^T ( \mathbf{r}_F - \mathbf{A}_{FF} \mathbf{y}_F ) \)
3. Solve \( \hat{\mathbf{A}}_C \mathbf{v}_C = \mathbf{y}_C \)
4. \( \mathbf{z}_F = J_{FC} \mathbf{v}_C + \mathbf{y}_F \)
5. \( \mathbf{v}_F = \mathbf{z}_F + Q_{FF}^{-1} ( \mathbf{r}_F - \mathbf{A}_{FC} \mathbf{v}_C - \mathbf{A}_{FF} \mathbf{z}_F ) \)

\[
B_{\text{HBMG}}^{-1} = J \begin{pmatrix} I & -Q_{FF}^{-1} \hat{\mathbf{A}}_{FC} \end{pmatrix} \begin{pmatrix} 2Q_{FF}^{-1} - Q_{FF}^{-1} \mathbf{A}_{FF} Q_{FF}^{-1} \hat{\mathbf{A}}_C^{-1} \end{pmatrix} \ldots
\]

Elementary algebra yields

\[
I - B_{\text{HBMG}}^{-1} \mathbf{A} = (I - \mathbf{R} \mathbf{A})(I - p \hat{\mathbf{A}}_C^{-1} p^T \mathbf{A})(I - \mathbf{R} \mathbf{A})
\]

with

\[
\mathbf{R} = \begin{pmatrix} Q_{FF}^{-1} & 0 \\ 0 & 0 \end{pmatrix}.
\]

Reminder:

\[
I - B_{\text{AMG}}^{-1} \mathbf{A} = (I - \mathbf{M}^{-1} \mathbf{A})(I - p \hat{\mathbf{A}}_C^{-1} p^T \mathbf{A})(I - \mathbf{M}^{-1} \mathbf{A})
\]

Block factorization without h.b.

\[
\mathbf{A} = \begin{pmatrix} \mathbf{A}_{FF} & \mathbf{A}_{FC} \\ \mathbf{A}_{CF} & \mathbf{A}_{CC} \end{pmatrix}
\]

\[
\approx \begin{pmatrix} I & \mathbf{A}_{FF}^{-1} \mathbf{A}_{FC} \\ \mathbf{A}_{CF} & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{I} & \mathbf{S}_A \end{pmatrix} \begin{pmatrix} \mathbf{I} & \mathbf{A}_{FF}^{-1} \mathbf{A}_{FC} \\ \mathbf{I} & \mathbf{I} \end{pmatrix}
\]

\[
B_{\text{MBF}}^{-1} = \begin{pmatrix} I & P_{FF}^{-1} \mathbf{A}_{FC} \\ P_{FF} & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{I} & \mathbf{S}_A \end{pmatrix} \begin{pmatrix} \mathbf{I} & \mathbf{A}_{FF}^{-1} \mathbf{A}_{FC} \\ \mathbf{I} & \mathbf{I} \end{pmatrix}
\]

Preconditioning by MBF

\( \mathbf{v} = B_{\text{MBF}}^{-1} \mathbf{r} \) computed as

1. \( \mathbf{y}_F = P_{FF}^{-1} \mathbf{r}_F \)
2. \( \mathbf{y}_C = \mathbf{r}_C - \mathbf{A}_{CF} \mathbf{y}_F \)
3. Solve \( \hat{\mathbf{A}}_C \mathbf{v}_C = \mathbf{y}_C \)
4. \( \mathbf{v}_F = P_{FF}^{-1} ( \mathbf{r}_F - \mathbf{A}_{FC} \mathbf{v}_C ) \)

\[
B_{\text{MBF}}^{-1} = \begin{pmatrix} I & -P_{FF}^{-1} \mathbf{A}_{FC} \\ P_{FF} & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{I} & \mathbf{S}_A \\ \mathbf{A}_{CF} & \mathbf{I} \end{pmatrix}
\]

Possibly unstable!
Block factorization without h.b. (cont.)

\[ B_{HBB}^{-1} = q Q_{FF}^T q + p \hat{A}_C^{-1} p^T , \]

\[ B_{HBFF}^{-1} = q Q_{FF}^T p + \tilde{p} \hat{A}_C^{-1} \tilde{p}^T , \]

\[ B_{HBMM}^{-1} = q (Q_{FF}^T + Q_{FF}^T Q_{FF}^{-1} A_{FF} Q_{FF}^{-T}) q^T + \tilde{p} \hat{A}_C^{-1} \tilde{p}^T , \]

\[ \tilde{p} = \begin{pmatrix} -Q_{FF}^{-1} A_{FC} + (I - Q_{FF}^{-1} A_{FF}) J_{FC} \\ I \end{pmatrix} . \]

\[ B_{MBF}^{-1} = q P_{FF}^{-1} q^T + \bar{p} \hat{A}_C^{-1} \bar{p}^T , \]

\[ \bar{p} = \begin{pmatrix} -P_{FF}^{-1} A_{FC} \\ I \end{pmatrix} . \]

\[ p \] has to define a “correct” interpolation.

Correct interpolation

Essential requirement:

\[ \bar{p} = \begin{pmatrix} -P_{FF}^{-1} A_{FC} \\ I \end{pmatrix} \] good for low energy modes,

i.e. vectors \( v \) such that \( A v \approx 0 \).

Scalar elliptic PDEs: one such vector:

\[ e = \begin{pmatrix} 1 & \ldots & 1 \end{pmatrix}^T . \]

If one satisfies the row-sum criterion

\[ P_{FF} e_F = A_{FF} e_F , \]

then

\[ A_{FF} e_F + A_{FC} e_C \approx 0 \quad \Rightarrow \quad P_{FF}^{-1} A_{FC} e_C \approx e_F . \]

Correct interpolation (cont.)

If \( A \) is a symmetric M-matrix with nonnegative row-sum (SPD with nonpositive offdiagonal entries), several results available.

For instance:

\[ \bar{A} = J^T A J \]

with

\[ J = \begin{pmatrix} I & -P_{FF}^{-1} A_{FC} \\ \bar{I} \end{pmatrix} \]

satisfies

\[ \bar{\gamma} \leq \sqrt{1 - \frac{1}{\kappa(P_{FF}^{-1} A_{FF})}} . \]
Algebraic analysis of AMG

HBBF, HBMG: $\kappa \approx \frac{1}{1 - \gamma^2}$; MBF: $\kappa \approx \frac{1}{1 - \hat{\gamma}^2}$ (!).

AMG

Assumption: $2 M - A$ SPD or, equivalently, $\rho(I - M^{-1}A) < 1$. One has

$$\kappa(B_{AMG}^{-1}A) \leq \mu$$

where

$$\mu = \max_{z \neq 0} \frac{(z_F - J_{FC}z_C)^T X_{FF} (z_F - J_{FC}z_C)}{z^T A z}.$$

with $X_{FF}$ being the top left block of $X = M (2 M - A)^{-1} M$.

Algebraic analysis of AMG (cont.)

Further,

$$\frac{1}{1 - \hat{\gamma}^2} \leq \mu \leq \frac{1}{\lambda_{\min}(X_{FF}^{-1}A_{FF})} \frac{1}{1 - \gamma^2} \leq \frac{1}{\lambda_{\min}(M_{FF}^{-1}A_{FF}) (2 - \lambda_{\max}(M^{-1}A))} \frac{1}{1 - \hat{\gamma}^2}.$$

Example: SGS smoothing: $\lambda_{\max}(M^{-1}A) = 1$.

$$\frac{1}{1 - \hat{\gamma}^2} \leq \mu \leq \frac{1}{\lambda_{\min}(M_{FF}^{-1}A_{FF})} \frac{1}{1 - \hat{\gamma}^2}.$$

What we learned

- All methods work or fail together.
- They are relatively equivalent with respect to algebraic analysis (except “additive” HBBD).
- However they mimic “geometric” methods that behave differently in a multigrid or multilevel context.
- The $F/C$ partitioning has to be such that $A_{FF}$ is well conditioned.
- The interpolation $J_{FC}$ has to be such that $\hat{\gamma}$ is away from 1.
- MBF needs special care; it does not require explicitly $J_{FC}$, but $A_C$ needs to be provided.

Quality of the interpolation measured with

$$\tau = \max_{z \neq 0} \frac{(z_F - J_{FC}z_C)^T D_{FF} (z_F - J_{FC}z_C)}{z^T A z}$$

($D_{FF} = \text{diag}(A_{FF})$).

There holds

$$\tau \leq \frac{1}{\lambda_{\min}(D_{FF}^{-1}A_{FF})} \frac{1}{1 - \gamma^2}$$

$$\tau \geq \max \left( \frac{1}{\lambda_{\max}(D_{FF}^{-1}A_{FF})} \frac{1}{1 - \gamma^2}, \frac{1}{\lambda_{\min}(D_{FF}^{-1}A_{FF})} \right).$$
Consider
\[
\begin{pmatrix}
I & 0 & 0 \\
0 & A_{FF} & A_{FC} \\
0 & A_{CF} & A_{CC}
\end{pmatrix}
\begin{pmatrix}
I & -A_{FF}^{-1}A_{FC} \\
-A_{CF}A_{FF}^{-1} & I \\
0 & A_{CF} & I
\end{pmatrix}
= \begin{pmatrix}
A_{FF} \\
\gamma \\
A_{CF} \end{pmatrix}
\]
Block diagonal $\rightarrow \gamma = 0$.
$\rightarrow -A_{FF}^{-1}A_{FC}$ is the ideal algebraic interpolation.
However: $\hat{A}_C = \left( J_{FC}^T \ I \right) A \left( J_{FC} \ I \right)$
$\rightarrow J_{FC}$ has to remain sparse.

**Algebraic interpolation (cont.)**

- Possible improvement: take also into account “indirect” couplings ($J_{FC}$ less sparse).
- Essentially positive-type matrices with nonnegative row-sum: split $A = A_M + A_P$ where $\text{offdiag}(A_P) = \max(O, \text{offdiag}(A))$ and $A_P e = 0$; apply previous scheme to $A_M$; the bound on $\tau$ depends now on $\kappa(A_M^{-1} A)$.
- General case: no obvious solution so far if $A$ is not (weakly) diagonally dominant.

**Direct interpolation in AMG for $M$-matrices with nonnegative row-sum:**
\[
(J_{FC})_{ij} = \begin{cases} 
-\sum_{j \neq i} |(A)_{ij}| & \text{if } a_{ij} \text{ “strong”} \\
\frac{(A_{FC})_{ij}}{(A_{FF})_{ii}} \sum_{j \in C} |(A_{FC})_{ij}| & \text{if } a_{ij} \text{ “weak”}
\end{cases}
\]
Property:
\[
\tau \leq \max_i \sum_{j \in C} |(A)_{ij}| \sum_{j \in C} \frac{|(A_{FC})_{ij}|}{|a_{ij}| \text{ “strong”}}
\]
(Reminder: $\tau \approx \frac{1}{1-\gamma^2}$).

**Algebraic coarsening**

**Standard coarsening in AMG**
First classify the negative couplings in strong and weak, according some given threshold.
Next, repeat, till all nodes are marked either coarse or fine:
1. select an unmarked node as next coarse grid node, according to some priority rule (designed so as to favor a regular covering of the matrix graph);
2. select as fine grid nodes all nodes strongly negative coupled to this new coarse grid node.
AMG coarsening: how it works

**Five-point stencil**

Each $F$ node is strongly negative coupled to at least 1 $C$ node → standard interpolation works.

- Slow coarsening in case of low connectivity, anisotropy or strong asymmetry. (Too fast coarsening in case of high connectivity).
- May be cured with aggressive coarsening. Requires specialized interpolation.
- The number of nonzero entries per row tends to grow from level to level.
- May be sensitive to the Strong/Weak coupling threshold.
- All in all, works reasonably in many cases.

**Nine-point stencil**

AMG coarsening: pros and cons

- Group nodes into aggregates $G_i$ (partitioning of $[1, n]$).
- (Possible) prolongation $p$:
  \[(p)_{ij} = \begin{cases} 1 & \text{if } i \in G_j \\ 0 & \text{otherwise} \end{cases} \]
- Coarse grid matrix: $\tilde{A}_C = p^T A p$ given by
  \[(\tilde{A}_C)_{ij} = \sum_{k \in G_i} \sum_{t \in G_j} a_{kt}.\]
- Optionally select a $C$ node in each aggregates; other nodes are then $F$ nodes. Associated interpolation:
  \[\forall i \in F, j \in C : (J_{FC})_{ij} = \begin{cases} 1 & \text{if } i \in G_j \\ 0 & \text{otherwise} \end{cases}.\]

Aggregation

**Example: pairwise aggregation**

**Definition:** $S_i = \{ j \neq i \mid a_{ij} < -\beta \max_{a_{ik} < 0} |a_{ik}| \}$

**Initialization:** $F = \emptyset ; C = \emptyset ; U = [1, n]$;
  For all $i : m_i = |\{ j \in U \mid i \in S_j \}|$.

**Algorithm:** While $U \neq \emptyset$
1. select $i \in U$ with minimal $m_i$
2. select $j \in U$ such that $a_{ij} = \min_{k \in U} a_{ik}$
3. if $j \in S_i$:
   3a. $C = C \cup \{ j \}, F = F \cup \{ i \}, G_j = \{ i, j \}, U = U \setminus \{ i, j \}$
   3b. update: $m_k = m_k - 1$ for $k \in S_i$ and $k \in S_j$
   otherwise:
   3a'. $C = C \cup \{ i \}, G_i = \{ i \}, U = U \setminus \{ i \}$
   3b'. update: $m_k = m_k - 1$ for $k \in S_i$

**Aggregation**

- Group nodes into aggregates $G_i$ (partitioning of $[1, n]$).
- (Possible) prolongation $p$:
  \[(p)_{ij} = \begin{cases} 1 & \text{if } i \in G_j \\ 0 & \text{otherwise} \end{cases} \]
- Coarse grid matrix: $\tilde{A}_C = p^T A p$ given by
  \[(\tilde{A}_C)_{ij} = \sum_{k \in G_i} \sum_{t \in G_j} a_{kt}.\]
- Optionally select a $C$ node in each aggregates; other nodes are then $F$ nodes. Associated interpolation:
  \[\forall i \in F, j \in C : (J_{FC})_{ij} = \begin{cases} 1 & \text{if } i \in G_j \\ 0 & \text{otherwise} \end{cases}.\]
Double pairwise aggregation

Algorithm:
1. Apply simple pairwise aggregation to $A$.
   Output: $(F_1, C_1)$, and $G^{(1)}_i$, $i \in C_1$.
2. Compute the auxiliary matrix $A_1 = \left(a_{ij}^{(1)}\right)$, $i, j \in C_1$ with
   
   $a_{ij}^{(1)} = \sum_{k \in G_i^{(1)}} \sum_{\ell \in G_j^{(i)}} a_{k\ell}$.

3. Apply simple pairwise aggregation to $A_1$.
   Output: $(F_2, C_2)$, and $G^{(2)}_i$, $i \in C_2$.

4. $C = C_2$, $F = F_1 \cup F_2$, $G_i = \bigcup_{j \in G^{(1)}_i} G_j^{(1)}$, $i \in C$.

Example ($d = 100$)

Double pairwise aggregation
- First coarse grid
- Second coarse grid

Some remarks

Geometric multigrid does not benefit from semi-coarsening
→ $A_{FF}$ may be badly conditioned
→ has to be compensated by specialized smoothers.

Geometric schemes fix the coarsening and the interpolation; the smoother (the approximation to $A_{FF}$) is adapted to the problem.

Algebraic schemes fix the smoother (the approximation to $A_{FF}$); the coarsening is adapted to the problem.

With algebraic schemes, the adaptation is automatic.

Example

2D problem with anisotropy & discontinuity

Five-point finite difference approx. (uniform mesh) of

$$-a_x \frac{\partial^2 u}{\partial x^2} - a_y \frac{\partial^2 u}{\partial y^2} = f \quad \text{in} \quad \Omega = (0, 1) \times (0, 1)$$

$$\begin{cases} 
 u = 0 & \text{on} \quad y = 1, \ 0 \leq x \leq 1 \\
 \frac{\partial u}{\partial n} = 0 & \text{elsewhere on} \quad \partial \Omega 
\end{cases}$$

$$\begin{cases} 
 a_x = d, \ a_y = 1, \ f = 0 & \text{in} \quad (0.65, 0.95) \times (0.05, 0.65) \\
 a_x = 1, \ a_y = d, \ f = 0 & \text{in} \quad (0.25, 0.45) \times (0.25, 0.45) \\
 a_x = d, \ a_y = d, \ f = 1 & \text{in} \quad (0.05, 0.25) \times (0.65, 0.95) \\
 a_x = 1, \ a_y = 1, \ f = 0 & \text{elsewhere} 
\end{cases}$$

where $d$ is a parameter.
Aggregation: pros and cons

- Control of the coarsening speed.
- Insensitive to the Strong/Weak coupling threshold.
- Maintain the sparsity in coarse grid matrices, that are nevertheless “reasonable”, up to some scaling factor.
- The interpolation that is naturally associated with aggregation is bad (not an issue for MBF-based methods).
- Smoothed aggregation:
  optionally sparsify $A$ into $\tilde{A}$, in such a way that $Ae = \tilde{A}e$; then:

$$ p_{\text{sm agg}} = \left( I - \omega \tilde{D}^{-1} \tilde{A} \right) p_{\text{agg}} $$

where $\tilde{D} = \text{diag}(\tilde{A})$.

Checking the $F/C$ partitioning

$A_{FF}$ has to be well conditioned. This may be a posteriori checked.

Compatible relaxation (AMG)

Perform smoothing on a random r.h.s while freezing the values at $C$ variables. If the error at $F$ variables does not decay quickly, adapt the partitioning by moving to $C$ some of the slowly convergent $F$ variables.

Remark

Amounts to check the conditioning of $M_{FF}^{-1}A_{FF}$.

Remember that

$$ \kappa_{\text{AMG}} \sim \left( \lambda_{\text{min}}(M_{FF}^{-1}A_{FF})(2 - \lambda_{\text{max}}(M^{-1}A)) \right)^{-1}. $$

Checking the $F/C$ partitioning (cont.)

Dynamic MILU

The size of the pivots in a modified ILU ($P_{FF}e_F = A_{FF}e_F$) factorization is a good indication of the conditioning.

For instance, in some cases, letting $P_{FF} = L_{FF} Q_{FF}^{-1} U_{FF}$ with $\text{diag}(L_{FF}) = \text{diag}(U_{FF}) = Q_{FF}$, if $Q_{FF} \geq \xi \text{diag}(A_{FF})$ for some $\xi > \frac{1}{2}$, then

$$ \kappa(P_{FF}^{-1}A_{FF}) \leq \frac{1}{2 - \xi^{-1}}. $$
Dynamic MILU: algorithm

1. **Repeat**: False.
2. (re)initialize:
   \[ Q_{FF} = \text{diag}(A_{FF}), \ L_{FF} = \text{lower}(A_{FF}), \ U_{FF} = \text{upper}(A_{FF}). \]
3. for \( k = 1, \ldots, n, \ k \in F \):
   - if \( q_{kk} \geq \gamma a_{kk} \):
     - eliminate row & column \( k \) in \( A_{FF} \) according to the MILU algorithm
   - otherwise: \( F = F \setminus \{ k \} \), \( C = C \cup \{ k \} \)
   - **Repeat**: True.
4. If (**Repeat**), GoTo 1, possibly decreasing the value of \( \gamma \).

Dynamic MILU: example

Double pairwise aggregation, second coarse grid

Without dynamic MILU

With dynamic MILU

<table>
<thead>
<tr>
<th>n_c</th>
<th>n / n_c</th>
<th>n_z / n_c</th>
</tr>
</thead>
<tbody>
<tr>
<td>1025</td>
<td>14.2</td>
<td>6.02</td>
</tr>
<tr>
<td>1124</td>
<td>12.9</td>
<td>5.94</td>
</tr>
</tbody>
</table>

From two- to multi-level

- Exploit recursively the same ideas.
- Succession of grids (levels), each with its own \( F/C \) partitioning and interpolation \( J_{FC} \), and also with its “ideal” preconditioner in which the matrix at the coarser level is inverted exactly.
- At some point the coarse grid matrix in indeed small enough to be factorized exactly.
- At every other level, the “ideal” preconditioner is adapted, exchanging the exact solution to \( \hat{A}_C v_C = y_C \) for an approximate solution.
- Approximate \( \hat{A}_C v_C = y_C \) with 1 application of the preconditioner: V cycle. inner iterations: W cycle.

W cycles may be based on fixed point iterations, but Krylov (CG, GMRES) is more robust. Then:

- Except at the coarsest level, the so defined preconditioner is slightly variable from step to step → Flexible Krylov subspace methods (FCG, FGMRES).
- Inner iterations are exited when the relative residual error is less than \( 0.35 \), or when the number of iterations reaches \( \text{int}[n_z(A)/n_z(\hat{A}_C)] \).
From two- to multi-level (cont.)

**AMG:** often efficient with V cycle

→ simplicity, consistency with slow coarsening.

The use of V cycle is based on experiment and mimicry of geometric schemes

→ it may be not robust to rely on V cycle.

**Block factorization** methods: require W cycle (geometric schemes do require it too)

→ need coarsening fast enough.

---

**Numerical results (previous problem)**

MBF with aggregation & dynamic MILU

\[ \begin{align*}
\text{Cost of resolution} & \approx 28 \\
\text{Cost of 1 unprec. CG iter.} & \\
\end{align*} \]

<table>
<thead>
<tr>
<th>( h^{-1} = 600 )</th>
<th>( h^{-1} = 1200 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \frac{n}{n_c} )</td>
<td>inner</td>
</tr>
<tr>
<td>1</td>
<td>3.99</td>
</tr>
<tr>
<td>2</td>
<td>3.97</td>
</tr>
<tr>
<td>4</td>
<td>3.96</td>
</tr>
<tr>
<td>10</td>
<td>3.95</td>
</tr>
<tr>
<td>10^2</td>
<td>3.95</td>
</tr>
<tr>
<td>10^4</td>
<td>3.95</td>
</tr>
<tr>
<td>10^6</td>
<td>3.95</td>
</tr>
</tbody>
</table>

---

**A non self-adjoint 3D problem**

Seven-point FD approx. (upwind scheme) of

\[-\nu \Delta u + \nu \nabla \nu = 0 \quad \text{in} \quad \Omega = (0,1) \times (0,1) \times (0,1)\]

\[
\begin{align*}
  u &= 1 \quad \text{on} \quad z = 1, \ 0 \leq x, \ y \leq 1 \\
  u &= 0 \quad \text{elsewhere on} \quad \partial \Omega
\end{align*}
\]

\[ \nu(x, y, z) = \begin{pmatrix}
  2x(1-x)(2y-1)z \\
  -(2x-1)y(1-y) \\
  -(2x-1)(2y-1)z(1-z)
\end{pmatrix}; \]

\( \nu = \infty \) corresponds to the Laplace equation.

**Uniform mesh** with constant mesh size \( h \).

**Stretched mesh:** refined in such a way that the ratio of maximum mesh size to minimum mesh size is equal to 200, the ratio of subsequent mesh sizes being constant.

---

**3D problem: numerical results**

\( \nu \) | \( 101 \times 101 \times 101 \) grid | \( 201 \times 201 \times 201 \) grid |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>( \frac{n}{n_c} )</td>
<td>inner</td>
<td>iter.</td>
</tr>
<tr>
<td>( \infty )</td>
<td>4.00</td>
<td>2.00</td>
</tr>
<tr>
<td>1</td>
<td>3.76</td>
<td>2.00</td>
</tr>
<tr>
<td>( 10^{-2} )</td>
<td>3.75</td>
<td>2.00</td>
</tr>
<tr>
<td>( 10^{-4} )</td>
<td>3.93</td>
<td>2.00</td>
</tr>
<tr>
<td>( 10^{-6} )</td>
<td>3.93</td>
<td>2.00</td>
</tr>
</tbody>
</table>

**Uniform mesh**

**Stretched mesh**
Some references

Many textbooks on multigrid, but few address algebraic schemes.


is recommended for a general introduction to multigrid; it contains in appendix the best available review on AMG:


Other results in research papers. Let mention mine!

- Algebraic multigrid and algebraic multilevel methods: a theoretical comparison
- Aggregation-based algebraic multilevel preconditioning (see homepage for details and download)

PhD Fellowship

Area: numerical nuclear reactor simulation

Collaboration between ULB and Framatome ANP

Location: Framatome ANP GmbH in Erlangen, Germany (main European research center of the group) with periodical stays in Brussels.

Task: adaptation of advanced preconditioned iterative techniques to nuclear reactor simulation.

Please contact me for further information.
ynotay@ulb.ac.be