Algebraic multilevel preconditioning and aggregation

Yvan Notay*

Université Libre de Bruxelles
Service de Métrologie Nucléaire

September 10, 2004, Bulgaria

* Supported by the “Fonds National de la Recherche Scientifique”, Maître de recherches
http://homepages.ulb.ac.be/~ynotay
Algebraic multilevel preconditioning:

- multilevel ILU: combines ideas from ILU and (algebraic) multigrid;
Introduction

Algebraic multilevel preconditioning:

- **multilevel ILU**: combines ideas from ILU and (algebraic) multigrid;
- well established for regular grids & geometric coarsening;
Algebraic multilevel preconditioning:

- multilevel ILU: combines ideas from ILU and (algebraic) multigrid;
- well established for regular grids & geometric coarsening;
- here: towards a truly algebraic method;
Introduction

Algebraic multilevel preconditioning:

- **multilevel ILU**: combines ideas from ILU and (algebraic) multigrid;
- well established for regular grids & geometric coarsening;
- **here**: towards a truly algebraic method;
- using coarsening by aggregation (control of setup cost & memory requirements).
System to solve:

\[ Au = b \]

(1) Select a \( F/C \) partitioning of the unknowns; partition \( A \) accordingly:

\[
A = \begin{pmatrix}
A_{FF} & A_{FC} \\
A_{CF} & A_{CC}
\end{pmatrix}
\]
Basic principle (two-level scheme) (2)

(2) The exact block factorization:

\[
A = \begin{pmatrix}
A_{FF} & S_A \\
A_{CF} & 0
\end{pmatrix}
\begin{pmatrix}
I & A_{FF}^{-1}A_{FC} \\
0 & I
\end{pmatrix}
\]

where

\[
S_A = A_{CC} - A_{CF} A_{FF}^{-1} A_{FC}
\]

is approximated by

\[
B = \begin{pmatrix}
P_{FF} & \\
P_{CF} & S
\end{pmatrix}
\begin{pmatrix}
I & P_{FF}^{-1}A_{FC} \\
0 & I
\end{pmatrix}
\]

(Schur complement)
(2) The exact block factorization:

\[
A = \begin{pmatrix}
A_{FF} & S_A \\
A_{CF} & A_{CC}
\end{pmatrix}
\begin{pmatrix}
I & A_{FF}^{-1}A_{FC} \\
A_{FC} & I
\end{pmatrix}
\]

where

\[
S_A = A_{CC} - A_{CF} A_{FF}^{-1} A_{FC}
\]

is approximated by

\[
B = \begin{pmatrix}
P_{FF} & S \\
A_{CF} & A_{CC}
\end{pmatrix}
\begin{pmatrix}
I & P_{FF}^{-1}A_{FC} \\
A_{FC} & I
\end{pmatrix}
\]

Needed ingredients: \(P_{FF} \approx A_{FF}, \ S \approx S_A\)
Coarsening by aggregation (1)

Main idea:

- group together nodes having strong negative connections between them (partitioning of $[1, n]$);
- in each group select one node $i$ to be a $C$ node; the other ones are $F$ nodes;
- noting $G_i$ the group of nodes from which the $C$ node $i$ has been extracted (set of nodes aggregated with $i$), the coarse grid matrix (aggregated matrix) is obtained by summing entries:

$$S_{ij} = \sum_{i \in G_i} \sum_{j \in G_j} a_{ij}.$$

Tends to reproduce the stencil from the fine grid.
Coarsening by aggregation (2)

\[ S = \begin{pmatrix} J_{FC}^T & I \end{pmatrix} \begin{pmatrix} A_{FF} & A_{FC} \\ A_{CF} & A_{CC} \end{pmatrix} \begin{pmatrix} J_{FC} \\ I \end{pmatrix} \]

where

\[ \forall i \in F, j \in C : (J_{FC})_{ij} = \begin{cases} 1 & \text{if } i \in G_j \\ 0 & \text{otherwise} \end{cases}. \]
Coarsening by aggregation (2)

\[
S = \begin{pmatrix} J_{FC}^T & I \end{pmatrix} \begin{pmatrix} A_{FF} & A_{FC} \\ A_{CF} & A_{CC} \end{pmatrix} \begin{pmatrix} J_{FC} \\ I \end{pmatrix}
\]

where

\[
\forall i \in F, j \in C: (J_{FC})_{ij} = \begin{cases} 1 & \text{if } i \in G_j \\ 0 & \text{otherwise} \end{cases}
\]

\(J_{FC}\) corresponds to a cheap but crude interpolation: each \(F\) node takes its value from exactly one \(C\) node.
Coarsening by aggregation (2)

\[
S = \begin{pmatrix} J_{FC}^T & I \end{pmatrix} \begin{pmatrix} A_{FF} & A_{FC} \\ A_{CF} & A_{CC} \end{pmatrix} \begin{pmatrix} J_{FC} \\ I \end{pmatrix}
\]

where

\[
\forall i \in F, j \in C : (J_{FC})_{ij} = \begin{cases} 1 & \text{if } i \in G_j \\ 0 & \text{otherwise} \end{cases}
\]

\(J_{FC}\) corresponds to a cheap but crude interpolation: each \(F\) node takes its value from exactly one \(C\) node.

**Bad interpolation: does not work well with multigrid.**
Coarsening by aggregation (2)

\[
S = \begin{pmatrix} J_{FC}^T & I \end{pmatrix} \begin{pmatrix} A_{FF} & A_{FC} \\ A_{CF} & A_{CC} \end{pmatrix} \begin{pmatrix} J_{FC} \\ I \end{pmatrix}
\]

where

\[
\forall i \in F, j \in C : (J_{FC})_{ij} = \begin{cases} 1 & \text{if } i \in G_j \\ 0 & \text{otherwise} \end{cases}
\]

\(J_{FC}\) corresponds to a cheap but crude interpolation: each \(F\) node takes its value from exactly one \(C\) node.

Bad interpolation: does not work well with multigrid.

But produces reasonable coarse grid matrices \(S\) (up to a scaling factor).
Coarsening by aggregation (3)

Relative solution cost – vs – Scaling factor

Model problem

Non model problem

Problem 3: $d = 1$

- Agg–AMG, $h^{-1} = 500$
- Agg–AMG, $h^{-1} = 1000$
- Agg–MBF, $h^{-1} = 500$
- Agg–MBF, $h^{-1} = 1000$

Problem 3: $d = 100$

(same legend as for the left figure)
Double pairwise aggregation (1)

Algorithm outline:

- group the nodes pairwise following the strongest (negative) coupling – some singletons left;
- form the corresponding aggregated matrix;
- repeat the process: group pairwise the groups (pairs and singletons) obtained from the first pass, following the strongest (negative) coupling in the aggregated matrix;
- in each resulting group (mostly quadruplets), select one node to be a $C$ node; the other ones are $F$ nodes.
Double pairwise aggregation (2)

Features:

\[
\frac{n_C}{n} \approx \frac{1}{4}
\]

- independently of the problem

→ control of setup cost & memory requirements

(if semi-coarsening: mesh size multiplied by 4 in the direction of coarsening);
Double pairwise aggregation (2)

Features:

- \( \frac{n_C}{n} \approx \frac{1}{4} \) independently of the problem
  - control of setup cost & memory requirements
  - (if semi-coarsening: mesh size multiplied by 4 in the direction of coarsening);

- not very sensitive to Strong/Weak coupling threshold;
Double pairwise aggregation (2)

Features:

- \( \frac{n_C}{n} \approx \frac{1}{4} \) independently of the problem

  \[ \rightarrow \text{control of setup cost & memory requirements} \]

- (if semi-coarsening: mesh size multiplied by 4 in the direction of coarsening);

- not very sensitive to Strong/Weak coupling threshold;

- easy to parallelize.
Approximation $P_{FF}$ to $A_{FF}$ (1)

\[
B^{-1} = \begin{pmatrix} I & -P_{FF}^{-1}A_{FC} \\ I & I \end{pmatrix} \begin{pmatrix} P_{FF}^{-1} \\ S \end{pmatrix} \begin{pmatrix} I & 0 \\ -A_{CF}P_{FF}^{-1} & I \end{pmatrix}
= \begin{pmatrix} I \\ 0 \end{pmatrix} P_{FF}^{-1} \begin{pmatrix} I & 0 \end{pmatrix} + \begin{pmatrix} -P_{FF}^{-1}A_{FC} \\ I \end{pmatrix} S^{-1} \begin{pmatrix} -A_{CF}P_{FF}^{-1} & I \end{pmatrix}
\]

Additive two-level scheme with prolongation

\[
\tilde{\rho} = \begin{pmatrix} -P_{FF}^{-1}A_{FC} \\ I \end{pmatrix}
\]

$\rightarrow P_{FF}$ such that $-P_{FF}^{-1}A_{FC}$ is a "correct" interpolation.
Approximation $P_{FF}$ to $A_{FF}$ (2)

MILU factorization
Cheap approximation that satisfies

$$P_{FF} e_F = A_{FF} e_F,$$

where $e = \begin{pmatrix} 1 & \ldots & 1 \end{pmatrix}^T$.

Hence, $A_{FF} e_F + A_{FC} e_C \approx 0$ entails

$$-P_{FF}^{-1} A_{FC} e_C \approx e_F.$$

For scalar PDEs, this is enough to ensure that the interpolation is “correct”.
How to ensure that

- the MILU factorization is well defined (no break down);
- it is an accurate approximation to $A_{FF}$?
Approximation $P_{FF}$ to $A_{FF}$ (3)

How to ensure that
- the MILU factorization is well defined (no break down);
- it is an accurate approximation to $A_{FF}$?

In many cases, no problem arises because $A_{FF}$ is well conditioned (diagonally dominant).
Approximation $P_{FF}$ to $A_{FF}$ (3)

How to ensure that

- the MILU factorization is well defined (no break down);
- it is an accurate approximation to $A_{FF}$?

In many cases, no problem arises because $A_{FF}$ is well conditioned (diagonally dominant).

For better robustness: dynamic MILU factorization: $F$ nodes for which the pivot would not be at least $\frac{3}{5}$ of the corresponding diagonal element are rejected to the $C$ set.

Guarantees existence and good conditioning properties of the MILU factorization.
From two- to multi-level (1)

Preconditioner:

\[
B = \begin{pmatrix}
P_{FF} & S \\
A_{CF} & S
\end{pmatrix}
\begin{pmatrix}
I & P_{FF}^{-1}A_{FC} \\
I & I
\end{pmatrix}.
\]

Requires solving a system with \( S \).

Recursivity: \( S \) has a structure similar to that of \( A \)

\( \rightarrow \) the same technique is applied to precondition \( S \).

The system with \( S \) is then solved by preconditioned iterations (flexible conjugate gradient of FGMRES).
**From two- to multi-level (2)**

**Stability:** consider

\[
B = \begin{pmatrix} P_{FF} \\ A_{21} \\ S \end{pmatrix} \begin{pmatrix} I & P_{FF}^{-1}A_{12} \\ & I \end{pmatrix}.
\]

The error in the solution to

\[
Bw = r
\]  

comes from the error made in solving

\[
Sx_C = y_C \quad (y_C = r_C - A_{CF}P_{FF}^{-1}r_F).
\]
From two- to multi-level (2)

Stability: consider

\[
B = \begin{pmatrix}
P_{FF} & S \\ A_{21} & I \\
\end{pmatrix}
\begin{pmatrix}
I & P_{FF}^{-1} A_{12} \\
& I \\
\end{pmatrix}.
\]

The error in the solution to

\[B w = r\] (1)

comes from the error made in solving

\[S x_C = y_C \quad (y_C = r_C - A_{CF} P_{FF}^{-1} r_F)\] (2)

Good news:

- the relative error in energy norm in (1) = that in (2);
- the residual in (1) is the one in (2) padded with zeros.
Some technical details:

- stopping criterion for inner iterations: relative residual error $\leq 0.35$;

- at most a few iterations allowed (to bound the cost of the application of the preconditioner);

- the coarse grid matrix $S$ is factorized exactly as soon as the cost of this factorization represents less than a fraction of an unpreconditioned CG iteration for the whole system.
Numerical results: Problem 1

Model 2D anisotropic problem, 5 point FD

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

\[u = 0 \quad \text{on} \quad x = 1, \quad 0 \leq y \leq 1\]

\[\frac{\partial u}{\partial n} = 0 \quad \text{elsewhere on} \quad \partial \Omega\]
Numerical results: Problem 1

Model 2D anisotropic problem, 5 point FD

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

\[u = 0 \quad \text{on} \quad x = 1, \ 0 \leq y \leq 1\]

\[\frac{\partial u}{\partial n} = 0 \quad \text{elsewhere on} \quad \partial \Omega\]

Geometric multigrid:

V cycle, damped Jacobi smoothing \((\omega = 0.5)\),

1 pre- and 1 post-smoothing step, preconditioner for CG.
Numerical results: Problem 1

Model 2D anisotropic problem, 5 point FD

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

\[u = 0 \quad \text{on} \quad x = 1, \ 0 \leq y \leq 1\]

\[\frac{\partial u}{\partial n} = 0 \quad \text{elsewhere on} \quad \partial \Omega\]

Geometric multigrid:

V cycle, damped Jacobi smoothing \((\omega = 0.5)\),

1 pre- and 1 post-smoothing step, preconditioner for CG.

Solution cost

\[
\frac{\text{Cost of 1 unprec. CG iter.}}{\text{Solution cost}}
\]
Numerical results: Problem 1

<table>
<thead>
<tr>
<th>$a_x$</th>
<th>$a_y$</th>
<th>$h^{-1} = 600$</th>
<th></th>
<th>$h^{-1} = 1200$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$\frac{n}{n_c}$</td>
<td>inner iter.</td>
<td>sol.</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>3.99</td>
<td>1.00</td>
<td>9</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>3.98</td>
<td>1.95</td>
<td>19</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>3.97</td>
<td>2.10</td>
<td>20</td>
</tr>
<tr>
<td>1</td>
<td>10</td>
<td>3.95</td>
<td>2.05</td>
<td>22</td>
</tr>
<tr>
<td>1</td>
<td>$10^2$</td>
<td>3.95</td>
<td>2.00</td>
<td>22</td>
</tr>
<tr>
<td>1</td>
<td>$10^4$</td>
<td>3.95</td>
<td>1.94</td>
<td>18</td>
</tr>
</tbody>
</table>

Geometric multigrid

Aggregation-based Algebraic multilevel
Numerical results: Problem 2

Model 2D anisotropic problem, bilinear finite elements

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

\[u = 0 \quad \text{everywhere on} \quad \partial \Omega\]

Stencil:

\[
\begin{align*}
    a_x = a_y &= 1 & a_x = 1, \ a_y &= 2 & a_x = 1, \ a_y &= 100 \\
    -2 & 16 & -2 & 24 & & & +196 & 808 & +196 \\
\end{align*}
\]
## Numerical results: Problem 2

<table>
<thead>
<tr>
<th>$a_x$</th>
<th>$a_y$</th>
<th>$h^{-1} = 600$</th>
<th>$h^{-1} = 1200$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$\frac{n}{n_c}$</td>
<td>inner</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>3.99</td>
<td>1.94</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>3.99</td>
<td>2.20</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>3.99</td>
<td>2.11</td>
</tr>
<tr>
<td>1</td>
<td>10</td>
<td>3.99</td>
<td>2.00</td>
</tr>
<tr>
<td>1</td>
<td>$10^2$</td>
<td>3.99</td>
<td>1.95</td>
</tr>
<tr>
<td>1</td>
<td>$10^4$</td>
<td>3.99</td>
<td>1.75</td>
</tr>
</tbody>
</table>
Numerical results: Problem 5

3D problem with discontinuity, 7 point FD

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

\[u = 0 \quad \text{on} \quad z = 1, 0 \leq x, y \leq 1\]

\[\frac{\partial u}{\partial n} = 0 \quad \text{elsewhere on} \quad \partial \Omega\]

\[a_x = a_y = a_z = \begin{cases} d & \text{in} \quad (\frac{1}{4}, \frac{3}{4}) \times (\frac{1}{4}, \frac{3}{4}) \times (\frac{1}{4}, \frac{3}{4}) \\ 1 & \text{elsewhere} \end{cases}\]

\[d : 1 \rightarrow 10^6\]
### Numerical results: Problem 5

<table>
<thead>
<tr>
<th>$d$</th>
<th>Uniform mesh</th>
<th>201 × 201 × 201 grid</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\frac{n}{n_c}$</td>
<td>inner iter.</td>
</tr>
<tr>
<td>1</td>
<td>4.00</td>
<td>2.05</td>
</tr>
<tr>
<td>10</td>
<td>4.00</td>
<td>2.05</td>
</tr>
<tr>
<td>$10^2$</td>
<td>4.00</td>
<td>2.05</td>
</tr>
<tr>
<td>$10^6$</td>
<td>4.00</td>
<td>2.09</td>
</tr>
</tbody>
</table>

### Uniform mesh

### Non uniform mesh

<table>
<thead>
<tr>
<th>$d$</th>
<th>Uniform mesh</th>
<th>201 × 201 × 201 grid</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\frac{n}{n_c}$</td>
<td>inner iter.</td>
</tr>
<tr>
<td>1</td>
<td>3.93</td>
<td>2.05</td>
</tr>
<tr>
<td>10</td>
<td>3.89</td>
<td>2.05</td>
</tr>
<tr>
<td>$10^2$</td>
<td>3.89</td>
<td>2.04</td>
</tr>
<tr>
<td>$10^6$</td>
<td>3.89</td>
<td>2.04</td>
</tr>
</tbody>
</table>
Numerical results: Problem 7

3D convection diffusion problem, 7 point FD (upwind)

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

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

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

\[
\nu = \infty \rightarrow \quad \text{Laplace equation (no convection).}
\]
### Numerical results: Problem 7

<table>
<thead>
<tr>
<th>$\nu$</th>
<th>$\frac{n}{n_c}$</th>
<th>inner iter.</th>
<th>sol.</th>
<th>$\frac{n}{n_c}$</th>
<th>inner iter.</th>
<th>sol.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\infty$</td>
<td>4.00</td>
<td>2.00</td>
<td>15</td>
<td>79.6</td>
<td>4.00</td>
<td>2.00</td>
</tr>
<tr>
<td>$10^{-1}$</td>
<td>3.80</td>
<td>2.00</td>
<td>16</td>
<td>99.4</td>
<td>3.81</td>
<td>2.00</td>
</tr>
<tr>
<td>$10^{-2}$</td>
<td>3.75</td>
<td>2.00</td>
<td>18</td>
<td>119.7</td>
<td>3.84</td>
<td>1.94</td>
</tr>
<tr>
<td>$10^{-4}$</td>
<td>3.93</td>
<td>2.00</td>
<td>21</td>
<td>136.9</td>
<td>3.93</td>
<td>2.00</td>
</tr>
</tbody>
</table>

#### Uniform mesh

#### Stretched mesh

<table>
<thead>
<tr>
<th>$\nu$</th>
<th>$\frac{n}{n_c}$</th>
<th>inner iter.</th>
<th>sol.</th>
<th>$\frac{n}{n_c}$</th>
<th>inner iter.</th>
<th>sol.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\infty$</td>
<td>3.91</td>
<td>1.94</td>
<td>16</td>
<td>79.3</td>
<td>3.94</td>
<td>1.88</td>
</tr>
<tr>
<td>$10^{-1}$</td>
<td>3.91</td>
<td>2.00</td>
<td>16</td>
<td>79.2</td>
<td>3.94</td>
<td>1.88</td>
</tr>
<tr>
<td>$10^{-2}$</td>
<td>3.92</td>
<td>1.65</td>
<td>20</td>
<td>95.5</td>
<td>3.95</td>
<td>1.94</td>
</tr>
<tr>
<td>$10^{-4}$</td>
<td>3.46</td>
<td>1.81</td>
<td>21</td>
<td>117.1</td>
<td>3.64</td>
<td>1.87</td>
</tr>
</tbody>
</table>
The resulting method offers a good compromise between robustness and efficiency.
Conclusions

- The resulting method offers a good compromise between robustness and efficiency.

- Grey box method: applicable to any sparse matrix without user handling; but not advisable to use it as in all cases, e.g. use specialized $P_{FF}$ for systems of PDEs (exact for all rigid body modes).
The resulting method offers a good compromise between robustness and efficiency.

**Grey box method:**
applicable to any sparse matrix without user handling;
but
not advisable to use it as in all cases, e.g. use specialized $P_{FF}$ for systems of PDEs (exact for all rigid body modes).

Further testing on unstructured grid problems is welcome.
Conclusions

- The resulting method offers a good compromise between robustness and efficiency.

- **Grey box method:** applicable to any sparse matrix without user handling; but not advisable to use it as in all cases, e.g. use specialized $P_{FF}$ for systems of PDEs (exact for all rigid body modes).

- Further testing on unstructured grid problems is welcome.

- The method seems not too difficult to parallelize.
The resulting method offers a good compromise between robustness and efficiency.

Grey box method:
applicable to any sparse matrix without user handling;
but
not advisable to use it as in all cases, e.g. use specialized $P_{FF}$ for systems of PDEs (exact for all rigid body modes).

Further testing on unstructured grid problems is welcome.

The method seems not too difficult to parallelize.

Thank you for your attention!