Multilevel Preconditioning Based on Element Agglomeration

Johannes Kraus
University of Leoben
Department of Applied Mathematics

ELEVENTH COPPER MOUNTAIN CONFERENCE ON MULTIGRID METHODS
Copper Mountain, April 1, 2003
Contents

- Type of problem and method
- 3 test problems
- Element agglomeration
- The Schur Complement Approximation
- ILU based on element agglomeration
- Factorization of the pivot matrix
- Nonlinear Algebraic Multilevel Iteration
- Outlook
Type of problem and method

Type of Problem:
- elliptic PDEs (scalar equations, or systems)
- finite element problems
- (general) SPD matrices

Type of Method:
- preconditioning based on element agglomeration
- hierarchical incomplete factorization
- nonlinear Algebraic Multilevel Iteration
- inner and outer iterations of CG or GCG typ

The main emphasis is on non-M matrices.
3 test problems

**Problem 1**

\[-(\Delta u + 2\alpha \frac{\partial^2 u}{\partial x \partial y}) = f \quad \text{on} \quad \Omega = [0, 1] \times [0, 1] \]

\[u = 0 \quad \text{on} \quad \Gamma, \quad |\alpha| < 1\]

**Grid:** orthogonal and uniform.

**Discretization by the finite difference stencil:**

\[
\frac{1}{h^2} \cdot \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} + \frac{\alpha}{h^2} \cdot \begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix}
\]

or, equivalently, the element matrix:

\[
A_e = \frac{1}{h^2} \cdot \begin{pmatrix} 1 & -\frac{1+\alpha}{2} & -\frac{1+\alpha}{2} & \alpha \\ -\frac{1+\alpha}{2} & 1 + \alpha & 0 & -\frac{1+\alpha}{2} \\ -\frac{1+\alpha}{2} & 0 & 1 + \alpha & -\frac{1+\alpha}{2} \\ \alpha & -\frac{1+\alpha}{2} & -\frac{1+\alpha}{2} & 1 \end{pmatrix}
\]

For \(\alpha > 0\) the stiffness matrix will be a non-M matrix!
Problem 2

\[ a(u, v) = (f, v) \quad \forall \quad v \in H^1(\Omega) \]
\[ a(u, v) = \int_{\Omega} (\nabla v)^T D(x) \nabla u \, dx + \int_{\Omega} \sigma uv \, dx \]
\[ \frac{\partial u}{\partial n} = 0 \quad \text{on} \quad \Gamma \]
\[ D(x) = \begin{pmatrix} 1 & 0 \\ 0 & \epsilon \end{pmatrix}, \quad 0 < \epsilon \leq 1 \]

Domain: \( \Omega \) is the unit square.

Discretization: quadratic finite elements with bilinear Ansatz functions.

Element matrix:

\[ A_e = \frac{1}{\epsilon h^2} \cdot \begin{pmatrix} 2 + 2\epsilon^2 & 1 - 2\epsilon^2 & -2 + \epsilon^2 & -1 - \epsilon^2 \\ 1 - 2\epsilon^2 & 2 + 2\epsilon^2 & -1 - \epsilon^2 & -2 + \epsilon^2 \\ -2 + \epsilon^2 & -1 - \epsilon^2 & 2 + 2\epsilon^2 & 1 - 2\epsilon^2 \\ -1 - \epsilon^2 & -2 + \epsilon^2 & 1 - 2\epsilon^2 & 2 + 2\epsilon^2 \end{pmatrix} \]

For \( |\epsilon| < \sqrt{2}/2 \) the stiffness matrix will be a non-M matrix!
Problem 3

\[
\frac{\partial^2 u}{\partial x^2} + \frac{1 - \mu}{2} \cdot \frac{\partial^2 u}{\partial y^2} + \frac{1 + \mu}{2} \cdot \frac{\partial^2 v}{\partial x \partial y} = f \quad \text{on} \quad \Omega
\]
\[
\frac{1 + \mu}{2} \cdot \frac{\partial^2 u}{\partial x \partial y} + \frac{1 - \mu}{2} \cdot \frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} = g \quad \text{on} \quad \Omega
\]
\[
u = v = 0 \quad \text{on} \quad \Gamma
\]

Domain: $\Omega$ is the unit square.

Discretization: quadratic finite elements.

Element matrix $A_e = \frac{1}{3\gamma_1\gamma_2 h^2} \begin{pmatrix} B_e & -C_e \\ -C_e^T & B_e \end{pmatrix}$:

\[
B_e : \begin{pmatrix} 4(1 + \gamma_1) & 3\gamma_2 & 2(1 - 2\gamma_1) & \gamma_3 \\ 3\gamma_2 & 4(1 + \gamma_1) & -\gamma_3 & -2(2 - \gamma_1) \\ 2(1 - 2\gamma_1) & -\gamma_3 & 4(1 + \gamma_1) & -3\gamma_2 \\ \gamma_3 & -2(2 - \gamma_1) & -3\gamma_2 & 4(1 + \gamma_1) \end{pmatrix}
\]

\[
C_e : \begin{pmatrix} 2(1 + \gamma_1) & 3\gamma_2 & 2(2 - \gamma_1) & \gamma_3 \\ 3\gamma_2 & 2(1 + \gamma_1) & -\gamma_3 & -2(1 - 2\gamma_1) \\ 2(2 - \gamma_1) & -\gamma_3 & 2(1 + \gamma_1) & -3\gamma_2 \\ \gamma_3 & -2(1 - 2\gamma_1) & -3\gamma_2 & 2(1 + \gamma_1) \end{pmatrix}
\]

$\gamma_1 = (1 - \mu)/2$, $\gamma_2 = (1 + \mu)/2$ and $\gamma_3 = 3(1 - 3\mu)/2$. 

5
Element agglomeration: principles

We consider a simple element agglomeration of the following type:

- dofs are associated with vertices of elements, that is, no dofs on edges (or faces) of elements

- elements are agglomerated along edges in 2D (and faces in 3D)

- agglomerates build coarse grid elements at each level, where coarse dofs are associated with vertices of coarse-grid elements
Element agglomeration: 2D example—Scalar equation

Fine-grid elements:

22 = 5 = 18 = 6 = 23
||   ||   ||   ||
7  –  1  –  8  –  2  –  9
||   ||   ||   ||
19 = 10 = 17 = 11 = 20
||   ||   ||   ||
12 –  3  – 13  –  4  – 14
||   ||   ||   ||
24 = 15 = 21 = 16 = 25

Coarse-grid elements:

22 = 18 = 23
||   ||   ||
19 = 17 = 20
||   ||   ||
24 = 21 = 25

Local ordering of dofs within an agglomerate:

6  –  2  –  7
|   |   |
3  –  1  –  4
|   |   |
8  –  5  –  9
Element agglomeration: 2D example—System of 2 PDEs

Fine-grid elements:

\[
\begin{align*}
43,44 & = 9,10 & = 35,36 & = 11,12 & = 45,46 \\
13,14 & - 1,2 & - 15,16 & - 3,4 & - 17,18 \\
37,38 & = 19,20 & = 33,34 & = 21,22 & = 39,40 \\
23,24 & - 5,6 & - 25,26 & - 7,8 & - 27,28 \\
47,48 & = 29,30 & = 41,42 & = 31,32 & = 49,50
\end{align*}
\]

Coarse-grid elements:

\[
\begin{align*}
43,44 & = 35,36 & = 45,46 \\
37,38 & = 33,34 & = 39,40 \\
47,48 & = 41,42 & = 49,50
\end{align*}
\]

Local ordering of dofs within an agglomerate:

\[
\begin{align*}
11,12 & - 3,4 & - 13,14 \\
5,6 & - 1,2 & - 7,8 \\
15,16 & - 9,10 & - 17,18
\end{align*}
\]
With respect to the ordering of dofs within an agglomerate, the agglomerate matrices take the $2 \times 2$ block form

$$A_a = \begin{pmatrix} A_{a,11} & A_{a,12} \\ A_{a,21} & A_{a,22} \end{pmatrix},$$

where $A_{a,11}$ and $A_{a,22}$ are the blocks corresponding to fine dofs and coarse dofs (coarse-element dofs), respectively.

This allows us to compute the local Schur complements

$$S_a = A_{a,22} - A_{a,21}(A_{a,11})^{-1}A_{a,12}$$

that serve as element matrices on the next coarser level, i.e.,

$$A^{(k+1)}_e := S^{(k)}_a$$

for all agglomerates $a$ at level $k$. 
As a global Schur complement approximation we now take the globally assembled local Schur complements (coarse-element matrices), i.e.,

\[ \tilde{S}^{(k)} = \sum_a S^{(k)}_a = \sum_e A^{(k+1)}_e. \]

That is, the approximation matrix \( \tilde{S}^{(k)} \) acts as coarse-grid operator, i.e., \( A^{(k+1)} := \tilde{S}^{(k)} \).

We will use the assembled Schur complement approximation later on.

Our next goal will be to assemble an incomplete factorization of the pivot-matrix (that yields the exact Schur complement).

Thus, all components of an algebraic multi-level iteration will be computed locally from agglomerate matrices!
A straight-forward approach can be used to define a new type of incomplete factorization method:

At levels $0 \leq k < l$, let

$$A^{(k)}_a = \begin{pmatrix} A^{(k)}_{a,11} & A^{(k)}_{a,12} \\ A^{(k)}_{a,21} & A^{(k)}_{a,22} \end{pmatrix},$$

and $A^{(k)}_{a,11} = L^{(k)}_a U^{(k)}_a$, where $\text{diag}(L^{(k)}_a) = I$.

Further on, let $A^{(l)} = \tilde{S}^{(l-1)} = L^{(l)} U^{(l)}$, $\text{diag}(L^{(l)}) = I$.

Then, since

$$A^{(k)}_a = \begin{pmatrix} L^{(k)}_a \\ A^{(k)}_{a,21} \left(U^{(k)}_a\right)^{-1} \end{pmatrix} \cdot \begin{pmatrix} U^{(k)}_a & (L^{(k)}_a)^{-1} A^{(k)}_{a,12} \\ 0 & I \end{pmatrix},$$

we define

$$\tilde{U}^{(l)} := \sum_{k=0}^{l-1} \sum_a \left[ U^{(k)}_a, (L^{(k)}_a)^{-1} A^{(k)}_{a,12} \right]$$

and, finally,

$$U := \tilde{U}^{(l)} + U^{(l)}, \quad \text{and} \quad L = U^T (\text{diag}(U))^{-1}.$$
Factorization of the pivot matrix: assembly of L and U

Now, let us use this idea in order to compute an incomplete LU factorization of the pivot matrix $A_{11}^{(k)}$ at every level $k$, $0 \leq k < l$:

The pivot block $A_{11}^{(k)}$ of

$$A^{(k)} = \begin{pmatrix} A_{11}^{(k)} & A_{12}^{(k)} \\ A_{21}^{(k)} & A_{22}^{(k)} \end{pmatrix}$$

is given by the assembly of the local pivot blocks, i.e.,

$$A_{11}^{(k)} = \sum_a A_{a,11}^{(k)}.$$

Thus,

$$\tilde{U}^{(k)} := \sum_a U_a^{(k)}$$

gives an approximation of $U^{(k)}$, where

$$A_{a,11}^{(k)} = L_a^{(k)} U_a^{(k)}, \quad A_{11}^{(k)} = L^{(k)} U^{(k)},$$

and

$$\tilde{L}^{(k)} = (\tilde{U}^{(k)})^T (\text{diag}(\tilde{U}^{(k)}))^{-1}.$$
Factorization of the pivot matrix: modification

As numerical experiments show, a simple modification helps to improve the approximations $\tilde{U}(k)$ and $\tilde{L}(k)$.

That is, we adjust the diagonal entries of $\tilde{U}(k)$ such that

$$\text{diag}(\tilde{A}_{11}^{(k)}) = \text{diag}(A_{11}^{(k)})$$

where

$$A_{11}^{(k)} = L^{(k)} U^{(k)}$$

and

$$\tilde{A}_{11}^{(k)} = \tilde{L}^{(k)} \tilde{U}^{(k)}.$$  

This is done by replacing the main diagonal of $\tilde{U}(k)$ by $(\tilde{u}_{ii})_{i=1}^{n}$, where

$$\tilde{u}_{ii} = a_{ii} - \sum_{j=1}^{i-1} \frac{\tilde{u}_{ji}^2}{\tilde{u}_{jj}}$$

$$\tilde{u}_{ij} = \tilde{u}_{ij} \quad \text{for} \quad j > i.$$
Factorization of the pivot matrix: numerical analysis

\[ \lambda_{\text{max}}((\tilde{A}_{11}^{(0)})^{-1}A_{11}^{(0)})/\lambda_{\text{min}}((\tilde{A}_{11}^{(0)})^{-1}A_{11}^{(0)}) : \]

**Problem 1**

<table>
<thead>
<tr>
<th>( \alpha )</th>
<th>4 aggs</th>
<th>16 aggs</th>
<th>64 aggs</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1.06955</td>
<td>1.07786</td>
<td>1.08069</td>
</tr>
<tr>
<td>0.5</td>
<td>1.07408</td>
<td>1.07447</td>
<td>1.07428</td>
</tr>
<tr>
<td>0.9</td>
<td>1.10782</td>
<td>1.11126</td>
<td>1.11164</td>
</tr>
<tr>
<td>0.99</td>
<td>1.11729</td>
<td>1.12165</td>
<td>1.12206</td>
</tr>
</tbody>
</table>

**Problem 2**

<table>
<thead>
<tr>
<th>( \epsilon )</th>
<th>4 aggs</th>
<th>16 aggs</th>
<th>64 aggs</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>1.16665</td>
<td>1.19919</td>
<td>1.21136</td>
</tr>
<tr>
<td>0.25</td>
<td>1.10414</td>
<td>1.13528</td>
<td>1.14592</td>
</tr>
<tr>
<td>0.1</td>
<td>1.02519</td>
<td>1.03400</td>
<td>1.04152</td>
</tr>
<tr>
<td>0.01</td>
<td>1.00215</td>
<td>1.00225</td>
<td>1.00228</td>
</tr>
</tbody>
</table>

**Problem 3**

<table>
<thead>
<tr>
<th>( \mu )</th>
<th>4 aggs</th>
<th>16 aggs</th>
<th>64 aggs</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>1.43393</td>
<td>1.46719</td>
<td>1.47636</td>
</tr>
<tr>
<td>0.25</td>
<td>1.53208</td>
<td>1.55038</td>
<td>1.55821</td>
</tr>
<tr>
<td>0.3</td>
<td>1.59849</td>
<td>1.60504</td>
<td>1.61168</td>
</tr>
<tr>
<td>0.5</td>
<td>1.76591</td>
<td>1.74664</td>
<td>1.75159</td>
</tr>
</tbody>
</table>
Linear System: \( A^{(0)}x^{(0)} = b \)

**initialize:**
- for \( k = 0 \) to \( l \) set \( x^{(k)} := 0 \)
- \( k := 0 \)
- \( d^{(0)} := b \)

**forward:**
- \( \sigma_k := \sigma_k + 1 \)
- if \((\sigma_k = 1 \land \land k \neq 0)\)
  - \( x^{(k)} := 0 \)
  - \( r^{(k)} := d^{(k)} \)
- end

**pcgsolve:** \( p_{1(k)}^{(\sigma_k)} \approx (A^{(k)})^{-1}r_{1(k)} \)

- \( d^{(k+1)} := r_{2(k)} - A_{2,1}^{(k)}p_{1(k)}^{(\sigma_k)} \)
- \( k := k + 1 \)
- if \( k < l \) goto forward

**solve:** \( L^{(l)}U^{(l)}x^{(l)} = d^{(l)} \)

**backward:**
- \( \sigma_k = 0; \quad k := k - 1 \)
- \( p_{2(k)}^{(\sigma_k)} := x^{(k+1)} \)

**solve:** \( \bar{L}^{(k)}\bar{U}^{(k)}y^{(k)} = A_{1,2}^{(k)}p_{2(k)}^{(\sigma_k)} \)

- \( p_{1(k)}^{(\sigma_k)} := p_{1(k)}^{(\sigma_k)} - y^{(k)} \)
- if \( \nu_k = 0 \)
  - \( x^{(k)} := p_{(\sigma_k)}^{(k)} \); goto backward
- end
Nonlinear Algebraic Multilevel Iteration: algorithm (cont.)

\[
q_{(\sigma_k)}^{(k)} := A^{(k)}p_{(\sigma_k)}^{(k)}
\]

for \( j = 1 \) to \( \sigma_k - 1 \)
\[
\beta = (q_{(\sigma_k)}^{(k)}, p_{(\sigma_k)}^{(k)})/\gamma_{(j)}^{(k)}
\]
\[
p_{(\sigma_k)}^{(k)} := p_{(\sigma_k)}^{(k)} - \beta p_{(j)}^{(k)}
\]
\[
q_{(\sigma_k)}^{(k)} := q_{(\sigma_k)}^{(k)} - \beta q_{(j)}^{(k)}
\]
end
\[
\gamma_{(\sigma_k)}^{(k)} = (q_{(\sigma_k)}^{(k)}, p_{(\sigma_k)}^{(k)})
\]
\[
\alpha = (r_{(\sigma_k)}^{(k)}, p_{(\sigma_k)}^{(k)})/\gamma_{(\sigma_k)}^{(k)}
\]
\[
x_{(k)} := x_{(k)} + \alpha p_{(\sigma_k)}^{(k)}
\]
\[
r_{(k)} := r_{(k)} - \alpha q_{(\sigma_k)}^{(k)}
\]
if \( (k = 0) \)
\[
\text{if (termination) goto finish}
\]
\[
\text{if (\( \sigma_0 = \nu_0 \)) \( \sigma_0 = 0 \)}
\]
end
if \( (\sigma_k < \nu_k) \) goto forward

finish:

goto backward
The above algorithm performs a nonlinear AMLI with inner GCG and PCG iterations, where $\nu_k$ is the number of inner GCG iterations at level $k$; restart at the fine-grid level after every $\nu_0$ iterations.

The number of dofs at level $k + 1$ is approximately one fourth of the number of dofs at level $k$. Hence, a visit at level $k$ must induce less than four visits at level $k + 1$ for optimal order of computational complexity of each outer iteration.

The PCG-solve in the forward loop is not critical in this respect if a fixed number of iterations is not exceeded, but it may increase the work per outer iteration significantly.

In our tests we used 3 inner PCG iterations in the forward loop and 2 inner GCG iterations at levels 1, 3, 5, .... The tables summarize the number of outer iterations that reduced the $L_2$-norm of the initial residual by a factor $10^{-6}$. 

17
Nonlinear Algebraic Multilevel Iteration: numerical results

Number of outer iterations for 3 to 8 levels (64 to 65536 elts):

Problem 1

<table>
<thead>
<tr>
<th>α \ l</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>0.5</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>0.9</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>6</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>0.99</td>
<td>5</td>
<td>5</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>7</td>
</tr>
</tbody>
</table>

Problem 2

<table>
<thead>
<tr>
<th>ε \ l</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>7</td>
<td>8</td>
<td>8</td>
<td>9</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>0.25</td>
<td>6</td>
<td>9</td>
<td>9</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>0.1</td>
<td>5</td>
<td>7</td>
<td>9</td>
<td>9</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>0.01</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>

Problem 3

<table>
<thead>
<tr>
<th>μ \ l</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>0.25</td>
<td>5</td>
<td>6</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>0.3</td>
<td>5</td>
<td>7</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>0.5</td>
<td>5</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>9</td>
<td>10</td>
</tr>
</tbody>
</table>
Outlook

Future investigations will deal with:

- numerical tests for 3D Problems and unstructured FE meshes
- more sophisticated agglomeration techniques (and the optimal size of agglomerates)
- different types of elements and problems
- a generalization of the method to unsymmetric problems
- a theoretical convergence analysis (if possible)