Multigrid Methods and their application in CFD

Michael Wurst
TU München
Multigrid Methods – Definition

Multigrid (MG) methods in numerical analysis are a group of algorithms for solving differential equations.

They are among the fastest solution techniques known today.
Outline

1. Typical design of CFD solvers

2. Methods for Solving Linear Systems of Equations

3. Geometric Multigrid

4. Algebraic Multigrid

5. Examples
Different CFD solvers

Typical design of CFD solver

CFD solver

- Coupled solver
- Segregated solver
• segregated, sequential solution of decoupled transport equations

• pressure correction equation: a tight tolerance for guaranteeing mass conservation

→ Multigrid methods
Coupled Solution Algorithm
Typical design of CFD solver

- momentum equations and pressure correction equation are such discretized that one gets a big coupled block equation system
- this equation system becomes very large – fast solver necessary

→ Multigrid methods
Coupled Solution Algorithm
Typical design of CFD solver

- Big coefficient matrix consisting of the momentum matrixes, the pressure correction matrix and coupling matrixes
- The solution vector contains velocity componentes and pressure

\[
\begin{pmatrix}
A_{uu} & 0 & 0 & A_{uP} \\
0 & A_{vv} & 0 & A_{vP} \\
0 & 0 & A_{ww} & A_{wP} \\
A_{Pu} & A_{Pv} & A_{Pw} & A_{PP}
\end{pmatrix}
\begin{pmatrix}
\vec{u} \\
\vec{v} \\
\vec{w} \\
\vec{p}
\end{pmatrix}
=
\begin{pmatrix}
\vec{q}_u \\
\vec{q}_v \\
\vec{q}_w \\
\vec{q}_p
\end{pmatrix}
\]
Basic Definitions
Methods for Solving Linear Systems of Equations

• Linear System of Equation:

\[ Au = f \]
\[ \sum a_{ij} u_j = f_i \]

A: sparse matrix of size \( n \times n \), symmetric, pos. diagonal elements, non-positive off diagonal elements (M-Matrix)

\( u \): exact solution
\( v \): approximation to the exact solution

• Two measures of \( v \) as an approximation to \( u \):

(Absolute) error: \( e = u - v \)
Residual: \( r = f - Av \)

• Measured by norms:

\( L_\infty \) - norm: \( \| e \|_\infty = \max_{1 \leq j \leq n} |e_j| \)
\( L_2 \) - norm: \( \| e \|_2 = \left( \sum_{j=1}^{n} e_j^2 \right)^{1/2} \)
Direct vs. Iterative Methods
Methods for Solving Linear Systems of Equations

• Direct methods
  – i.g. Gauss elimination / LU decomposition
  – solve the problem to the computational accuracy
  – high computational power

• Iterative methods / Relaxation methods
  – Gauss-Seidel / Jacobi relaxation
  – Solve the problem only by an approximation
  – could be sufficient and so be less time consuming
Iterative methods
Methods for Solving Linear Systems of Equations

\[ \sum a_{ij} u_j = f_i \]

- **Jacobi relaxation:**
  \[ u_i^{(n+1)} = \frac{1}{a_{ii}} \left( f_i - \sum_{j \neq i} a_{ij} u_j^{(n)} \right) \]

- **Gauss-Seidel relaxation:**
  \[ u_i^{(n+1)} = \frac{1}{a_{ii}} \left( f_i - \sum_{j<i} a_{ij} u_j^{(n+1)} - \sum_{j>i} a_{ij} u_j^{(n)} \right) \]
Properties of Iterative methods
Methods for Solving Linear Systems of Equations

• Example: Poisson equation
  \[-u'' = 0\]
  \[u(0) = u(n) = 0\]

• Discretisation:
  \[-u_{j-1} + 2u_j - u_{j+1} = 0 \quad \text{for} \quad 1 \leq j \leq n+1\]
  \[u_0 = u_n = 0\]

• Exact solution:
  \[u = 0\]
  error \[e = u - v = -v\]
Properties of Iterative methods

Methods for Solving Linear Systems of Equations

- Different starting values:

\[ v_j = \sin\left(\frac{jk\pi}{n}\right) \]

Fourier modes
Properties of Iterative methods
Methods for Solving Linear Systems of Equations

- Error vs. Number of iteration
Properties of Iterative methods
Methods for Solving Linear Systems of Equations

- Realistic starting value: \( v_j = \frac{1}{3} \left[ \sin \left( \frac{j\pi}{n} \right) + \sin \left( \frac{6j\pi}{n} \right) + \sin \left( \frac{32j\pi}{n} \right) \right] \)

\( k = 1: \)
- “low frequency wave”

\( k = 6: \)
- “medium frequency wave”

\( k = 32: \)
- “high frequency wave”

![Graph showing the error over iterations.](image)
Properties of Iterative methods

Methods for Solving Linear Systems of Equations

- Error: written in eigenvectors of $A$:
  \[ e^{(0)} = \sum_{k=1}^{n-1} c_k w_k \]

- Eigenvectors correspond to the modes of the problem

- Our problem:
  \[ w_{k,j} = \sin \left( \frac{jk\Pi}{n} \right) \quad 1 \leq k \leq n - 1 \]
  \[ 1 \leq k \leq \frac{n}{2} \quad \frac{n}{2} \leq k \leq n - 1 \]

  "Low frequency modes"  "High frequency modes"
  "Do not dissipate"  "Disappear"

Smother
Improvements of iterative solvers

Geometric Multigrid

• Idea: Have a good initial guess
  → How? Do some preliminary iterations on a coarse grid (grid with less points)
  Good, because iterations need less computational time

• How does an error look like on a coarse grid?
  It looks more oscillatory!
Improvements of iterative solvers
Geometric Multigrid

How does an error look like on a coarse grid?

→ If error is smooth on fine grid, maybe good to move to coarse grid.
Possible schemes for improvement
Geometric Multigrid

• Nested iteration:
  – Relax on $A u = f$ on a very coarse grid
to obtain an initial guess for the next finer grid
  
  – Relax on $A u = f$ on $\Omega^{4h}$ to obtain an initial guess for $\Omega^{2h}$
  – Relax on $A u = f$ on $\Omega^{2h}$ to obtain an initial guess for $\Omega^{h}$
  – Relax on $A u = f$ on $\Omega^{h}$ to obtain a final approximation to the solution.

• Problems: Relax on $A u = f$ on $\Omega^{2h}$?
  Last iteration: Error still smooth?
• 2nd possibility: Use of the residual equation

\[ Au = f \]

\[ Au - Av = f - Av \]

\[ Ae = r \]
Possible schemes for improvement

Geometric Multigrid

- **Correction scheme:**
  - Relax on $Au = f$ on $\Omega^h$ to obtain an approximation $v^h$
  - Compute the residual $r = f - A v^h$
    Relax on the residual equation $Ae = r$ on $\Omega^{2h}$ to obtain an approximation to the error $e^{2h}$
  - Correct the approximation obtained on $\Omega^h$ with the error estimate obtained on $\Omega^{2h}$: $v^h \leftarrow v^h + e^{2h}$

- **Problems:** Relax on $Ae = r$ on $\Omega^{2h}$? Transfer from $\Omega^{2h}$ to $\Omega^h$?
Transfer operators
Geometric Multigrid

- Transfer from coarse to fine grids: **Interpolation / Prolongation**
  \[
  \Omega^{2h} \rightarrow \Omega^h
  \]

- Transfer from fine to coarse grids: **Restriction**
  \[
  \Omega^h \rightarrow \Omega^{2h}
  \]
Transfer operators – Interpolation / Prolongation
Geometric Multigrid

• Interpolation / Prolongation: from coarse to fine grid

• Points on fine and on coarse grid: $v_{2j}^h = v_{j}^{2h}$

• Points only on the fine grid: $v_{2j+1}^h = \frac{1}{2}(v_{j}^{2h} + v_{j+1}^{2h})$
Transfer operators – Restriction

Geometric Multigrid

- Restriction: from fine to coarse grid

\[ v_{2j}^{2h} = \frac{1}{4} \left( v_{2j-1}^{h} + 2v_{2j}^{h} + v_{2j+1}^{h} \right) \]
### Properties of transfer operators

#### Geometric Multigrid

#### Interpolation / Prolongation

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

\[ I_{2h}^h = \frac{1}{2} \]

#### Restriction

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[ I_{h}^{2h} = \frac{1}{4} \]

#### Variational property:

\[ I_{2h}^h = c(I_{h}^{2h})^T \]
Properties of transfer operators

Geometric Multigrid

- Transfer of vectors: ✓

- Transfer of matrix $A$: $A^h \rightarrow A^{2h}$
  
  - Geometric answer: $A^{2h}$ is discretisation of the problem on the coarse grid
  
  - Algebraic answer: $A^{2h} = I^{2h}_h A^h I^{2h}_h$ (Galerkin condition)
• Iterative methods can effectively reduce high-oscillating errors until only a smooth error remains
• Smooth errors look less smooth on coarse grids
• Transfer of vectors and matrices from coarse to fine grids possible with two conditions:
  
  \[ A^{2h} = l_h^{2h} A^h l_{2h}^h \]

  \[ l_{2h}^h = c \left( l_h^{2h} \right)^T \]

  How can we put this in a good solution algorithm?
V-Cycle
Geometric Multigrid

- Relax on $A^h u^h = f^h$ $\nu_1$ times with initial guess $v^h$
- Compute $f^{2h} = l^2_h r^h$
  - Relax on $A^{2h} u^{2h} = f^{2h} \nu_1$ times with initial guess $v^{2h}$
  - Compute $f^{4h} = l^{4h}_2 r^{2h}$
    - Relax on $A^{4h} u^{4h} = f^{4h} \nu_1$ times with initial guess $v^{4h}$
    - Compute $f^{8h} = l^{8h}_4 r^{4h}$
      ...
- Solve $A^{Lh} u^{Lh} = f^{Lh}$
  ...
- Correct $v^{4h} \leftarrow v^{4h} + l^{4h}_8 v^{8h}$
  - Relax $A^{4h} u^{4h} = f^{4h} \nu_2$ times with initial guess $v^{4h}$
  - Correct $v^{2h} \leftarrow v^{2h} + l^{2h}_4 v^{4h}$
  - Relax $A^{2h} u^{2h} = f^{2h} \nu_2$ times with initial guess $v^{2h}$
  - Correct $v^{h} \leftarrow v^{h} + l^{h}_2 v^{2h}$
  - Relax $A^h u^h = f^h \nu_2$ times with initial guess $v^{h}$
V-Cycle
Geometric Multigrid

Relax

Restriction

Prolongation

h
2h
4h
8h
16h
Other cycles – W Cycle
Geometric Multigrid
Other cycles – Full Multigrid Cycle (FMG)

Geometric Multigrid
• Geometric Multigrid: structured meshes
• Problem: unstructured meshes, no mesh at all

• → Algebraic Multigrid (AMG)

Questions:
1) What is meant by grid now?
2) How to define coarse grids?
3) Can we use the same smoothers (Jacobi, Gauss-Seidel)
4) When is an error on a grid smooth?
5) How can we transfer data from fine grids to coarse grids or vice versa?
• GMG: known locations of grid points
  well-defined subset of the grid points define coarse grid

• AMG: subset of solution variables form coarse grid

\[ Au = f \]

\[ u = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix} \]
Smooth error

Algebraic Multigrid

- Defined as an error which is not effectively reduced by an iterative method
- Jacobi method: \( e^{i+1} = (I - D^{-1}A)e^i \)

- Measurement of the error with the A-inner product: \( \|e\|_A = (Ae, e)^{1/2} \)

- Smooth error: 
  \[
  \| (I - D^{-1}A)e \|_A \approx \|e\|_A \\
  \| e - D^{-1}Ae \|_A \approx \|e\|_A \\
  \rightarrow \| D^{-1}Ae \|_A \approx \|e\|_A
  \]
• Smooth error:

\[ \left\| D^{-1}Ae \right\|_A \triangleq \left\| e \right\|_A \]

\[ (D^{-1}Ae, Ae) \triangleq (e, Ae) \]

\[ (D^{-1}r, r) \triangleq (e, r) \]

\[ \sum_{i=1}^{n} \frac{r_i^2}{a_{ii}} \triangleq \sum_{i=1}^{n} r_i e_i \]

\[ \rightarrow |r_i| \triangleq a_{ii} |e_i| \]

\[ Ae \approx 0 \]
Implications of smooth error

Algebraic Multigrid

\[
A e \approx 0
\]

\[
a_{ii} e_i + \sum_{j \neq i} a_{ij} e_j \approx 0
\]

\[
a_{ii} e_i \approx -\sum_{j \neq i} a_{ij} e_j
\]
Selecting the coarse grid - requirements

Algebraic Multigrid

- Smooth error can be approximated accurately.

- Good interpolation to the fine grid.

- Should have substantially fewer points, so the problem on coarse grid can be solved with little expense.
• Definition 1:
  Given a threshold value $0 < \theta \leq 1$, the variable (point) $u_i$ strongly depends on the variable (point) $u_j$ if:

$$-a_{ij} \geq \theta \max_{k \neq i} \{-a_{ik}\}$$

• Definition 2:
  If the variable $u_i$ strongly depends on the variable $u_j$, then the variable $u_j$ strongly influences the variable $u_i$. 
Selecting the coarse grid – definitions

Algebraic Multigrid

- Two important sets:

\[ S_i : \text{ set of points that strongly influence } i, \]
that is the points on which the point \( i \) strongly depends.

\[ S_i = \left\{ j : -a_{ij} \geq \theta \max_{k \neq i} \{-a_{ik}\} \right\} \]

\[ S_i^T : \text{ set of points that strongly depend on the point } i. \]

\[ S_i^T = \left\{ j : i \in S_j \right\} \]
• Poisson equation: \(-\Delta u = 0\)

\[
-\frac{u_{i-1} + 2u_i - u_{i+1}}{h^2} + \frac{-u_{j-1} + 2u_j - u_{j+1}}{h^2} = 0
\]

\[
\frac{1}{h^2}(-u_{i-1} - u_{i+1} + 4u_i - u_{j-1} - u_{j+1}) = 0
\]
Discretisation on 5x5 grid:

For example, Point 12:

\[
\frac{1}{h^2} (-u_7 - u_{11} + 4u_{12} - u_{13} - u_{17}) = 0
\]

\[a_{12,7} = -1\]

\[a_{12,11} = -1\]

\[a_{12,13} = -1\]

\[a_{12,17} = -1\]

\[a_{12,12} = 4\]

\[S_i = \{ j : a_{ij} \geq \max_{k \neq i} \{-a_{ik}\} \}\]

\[S_{12} = \{7, 11, 13, 17\}\]

\[S_i^T = \{ j : i \in S_j \}\]

\[S_{12}^T = \{7, 11, 13, 17\}\]
1) Define a measure to each point of its potential quality as a coarse (C) point: amount $\lambda_i$ of members of $S_i^T$.
2) Assign point with maximum $\lambda_i$ to C-point
3) All points in $S_i^T$ become fine (F) points
4) For each new F point $j$: increase the measure $\lambda_k$ for all each unassigned point $k$ that strongly influence $j$: $k \in S_j$

5) Do 2)-4) until all points are assigned
Selecting the coarse grid - Example
Algebraic Multigrid
Interpolation: from coarse to fine grids

\[
(l_{2h}^h e)_i = \begin{cases} 
  e_i & \text{if } i \in C \\
  \sum_{j \in C_i} \omega_{ij} e_j & \text{if } i \in F 
\end{cases}
\]

Each fine grid point \( i \) can have three different types of neighboring points:
The neighboring coarse grid points that strongly influence \( i \)
The neighboring fine grid points that strongly influence \( i \)
Points that do not strongly influence \( i \), can be fine and coarse grid points

\[ \rightarrow \text{This information is contained in } \omega_{ij} \]
Example: 

\[-au_{xx} - cu_{yy} + bu_{xy} = 0\]

- Discretised with a two-dimensional mesh, divided into 4 parts:
Example
Algebraic Multigrid

Grid 2h

![Grid 2h Image](image-url)
Example
Algebraic Multigrid

Grid 4h

[Diagram of a grid with labeled axes 'X' and 'Y', showing a pattern of grid lines]
Example
Algebraic Multigrid

Grid 8h

![Graph of Grid 8h with coordinates and grid lines](image-url)
Advantages & Disadvantages of AMG
Algebraic Multigrid

Number of Iterations

Normalized Residual

-12
-10
-8
-6
-4
-2
0

SIP (Iterations: 2410)
V-Cycle (3 Grid-Levels)
W-Cycle (3 Grid-Level)
FMG-Cycle (3 Grid-Levels)
Advantages & Disadvantages of AMG

Algebraic Multigrid

Advantages

• Fast and robust
• Good for segregated solvers (SIMPLE)

Disadvantages

• The Galerkin Operation is a very expensive step
• Difficult to parallelize
• High setup-phase
• High storage requirements
• Not for coupled solvers

A cure are the aggregation based AMGs
• In the simplest case strongly connected coefficients are simply summed up

• Example:

<table>
<thead>
<tr>
<th></th>
<th>18</th>
<th>19</th>
<th>20</th>
<th>21</th>
<th>22</th>
<th>23</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>17</td>
<td>12</td>
</tr>
<tr>
<td>II</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
</tbody>
</table>

• cell 7 influences strongly cell 1
• cell 2 influences strongly cell 1
• build a new cell I from cell 1,2,7
• do the same to get the new cell II
### Aggregation based AMG

Algebraic Multigrid

<table>
<thead>
<tr>
<th></th>
<th>I</th>
<th>II</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>13</td>
<td>14</td>
<td>15</td>
</tr>
<tr>
<td>18</td>
<td>19</td>
<td>20</td>
</tr>
<tr>
<td>21</td>
<td>22</td>
<td>23</td>
</tr>
</tbody>
</table>

- To get the coefficients of the new coarse linear equation system sum up
- \( A_{I,II} = A_{7,13} + A_{7,8} + A_{2,8} \)
- \( A_{I,II} = A_{13,7} + A_{8,7} + A_{8,2} \)
- \( A_{I,I} = A_{1,1} + A_{2,2} + A_{7,7} \)
- \( A_{II,II} = A_{8,8} + A_{9,9} + A_{13,13} + A_{14,14} + A_{15,15} + A_{18,18} + A_{19,19} + A_{20,20} \)
Advantages

- The Galerkin operation becomes a simple summation of coefficients.
- The setup-phase becomes very fast.
- The procedure is easy to parallelize.
- Through giving maximum and minimum size of cells on coarser grids, one can pre-estimate memory effort.
- In a finite volume method, the coefficients are representing flux sizes from one cell to another, through summation on keeps the conservativness of the discretized system over all coarser levels.

Disadvantages

- The convergence rate becomes small compared to original AMG, but in the case of solution of the non-linear Navier-Stokes equation the reduction of the residual within one outer iteration has not to be very tight, reducing of about one to two orders of magnitude suffices.

The Agglomeration AMG is ideally applicable to the coupled solution of Navier-Stokes Equation System.
Thank you!

Discussion