Multigrid Methods and Applications in CFD

Michael Wurst

10 May 2009

Contents

1 Introduction

Multigrid Methods are a group of algorithms in numerical analysis for solving linear systems of equations. They are especially eligible for elliptical problems, for example the Poisson equation. This type of equation arises among others from the discretisation of the pressure correction equation in the solution of the Navier-Stokes equation. The advantage here is the linear dependence of the solution time from the number of grid points N which could be stated as optimal. In this work, two different multigrid methods (geometric multigrid and algebraic multigrid) are presented and the differences between them are shown. Additionally their application to computational fluid dynamics is demonstrated with an example.

2 Typical design of CFD solvers

There are basically two approaches for the solution of the Navier-Stokes equation, the coupled solver and the segregated solver. The Navier-Stokes equation is a non-linear system of equation. It contains 4 unknowns, three velocity components and the pressure, but consists only of three equations. Therefore the conservation of mass is used as an additional equation. In the segregated solver one uses an iterative scheme for the solution . First the momentum equation is solved with the pressure distribution from the last time step while in a second step the fluid velocity is corrected based on the pressure from the pressure-correction equation. This pressure-correction equation is a Poisson equation whose coefficient matrix is ill conditioned. Nevertheless the solution of the pressure-correction equation is needed to a tight tolerance for guaranteeing mass conservation. This makes a fast, robust linear solver necessary which can be achieved by multigrid methods. The coupled solver discretises the momentum and the pressure correction equation in such a way that one gets a big coupled block equation system which solves the pressure and the three velocity components in one matrix system. This system can become quite large which explains the need for a fast linear solver based on multigrid methods.

3 Basic methods and their properties for solving linear systems of equations

In the coupled as well as in the segregated approach one gets a linear system of equations given by

$$A\boldsymbol{u} = \boldsymbol{f}$$

or in components

$$\sum a_{ij} u_j = f_i$$

In this context ***u*** is the exact solution, A is a sparse (symmetric) matrix of size n×n and ***v*** denotes an approximation to the exact solution ***u***. There are now two important measures of ***v*** as an approximation to ***u***:

- Absolute error ***e*** given by

  $$\boldsymbol{e} = \boldsymbol{u} - \boldsymbol{v} \text{ .}$$

  The problem is that the absolute error is inaccessable as the exact solution itself.
- Residual ***r***; is the amount by which the approximation ***v*** fails to satisfy the original problem *A**u***=***f***:

  $$\boldsymbol{r} = \boldsymbol{f} - A\boldsymbol{v} \text{ .}$$

Both measurements are vectors and can be measured by any vector norm, like the L_2 or the L_∞ norm:

$$\|\boldsymbol{e}\|_\infty = \max_{1\le j\le n} |e_j| \qquad \text{and} \qquad \|\boldsymbol{e}\|_2 = \left\{ \sum_{j=1}^{n} e_j^2 \right\}^{1/2} .$$

In general, there are now two possibilities for the solution of the linear system of equations, direct methods and iterative or relaxation methods. Direct methods like Gauss elimination solve the problem to the computational accuracy but have high computational costs. Iterative methods instead solve the problem only by an approximation but this could be sometimes sufficient and therefore less time consuming. Two iterative methods are the Jacobi relaxation and the Gauss-Seidel relaxation. The idea of the Jacobi method is to solve the ith equation for the ith unknown using the approximations from the last iteration for the other unknowns. It can be written by

$$u_i^{(n+1)} = \frac{1}{a_{ii}}\left(f_i - \sum_{j \neq i} a_{ij} u_j^{(n)}\right).$$

The Gauss-Seidel method incorporates a simple change: components of the new approximation are used as soon as they are computed: This means that components of the approximation vector ***v*** are used as soon as they are updated:

$$u_i^{(n+1)} = \frac{1}{a_{ii}}\left(f_i - \sum_{j<i} a_{ij} u_j^{(n+1)} - \sum_{j>i} a_{ij} u_j^{(n)}\right).$$

Applying these iterative methods one can observe that the error decreases each iteration. But after some iterations the error does not reduce anymore. One can explain this behaviour assuming the error being a vector sum from the eigenvectors of the matrix *A*:

$$\boldsymbol{e}^{(0)} = \sum_{k=1}^{n-1} c_k \boldsymbol{w}_k \ .$$

These eigenvectors are connected closely to the eigen modes of the problem. If we now apply an iterative method, it is found out that the error components from high-frequent eigenvectors (or modes) disappear soon and the error components from low-frequent eigenvectors (or modes) do not disappear. Therefore these iterative methods are also called smoothers.

4 Geometric Multgrid

The open question is now how to improve these iterative methods in order to reduce all error components. The idea behind the multigrid methods is now to have at least two different grids of different size, one coarse grid and one fine grid. One can show that the error on the coarse grid looks more oscillatory than on the fine grid or in other words the error on a fine grid looks less smoth than on a coarse grid. This can be used now for the following procedure. First oen applies an iterative scheme on a fine grid until the error only consists of smooth, low frequent components. Then this error is transformed to the coarse grid where it looks more oscillatory. If we now relax on this grid it should effectively eliminate additional error modes. This could be continued with more than one coarse grid but it leaves the problem if the error is still smooth on the fine grid. Another possibility is the use of the residual equation and applying the following scheme, called the correction scheme:

- Relax on $A\boldsymbol{u} = \boldsymbol{f}$ on Ω^h to obtain an approximation $\boldsymbol{v}^h$
- Compute the residual $\boldsymbol{r} = \boldsymbol{f}\text{-}A\boldsymbol{v}^h$ and relax on the residual equation $A\boldsymbol{e} = \boldsymbol{r}$ on Ω^{2h} to obtain an approximation to the error $\boldsymbol{e}^{2h}$
- Correct the approximation obtained on Ω^h with the error estimate obtained on Ω^{2h}: $\boldsymbol{v}^h \leftarrow \boldsymbol{v}^h + \boldsymbol{e}^{2h}$

Here Ω^h with superscript *h* denotes the fine grid while Ω^{2h} with superscript *2h* denotes the coarse grid. This basic scheme could also be used for more than one coarse grid which will be shown later. Based on this scheme the following formal steps have to be defined:

- Selection of the coarse grid(s)
- Definition of transformation operators for vectors and the matrix *A* from coarse to fine grids and vice versa

In geometrid multigrid the selection of coarse grids is based on geometric relations, for example in a one dimensional problem every second fine grid point is defined as a coarse grid point. With this relation it is possible to design as many coarse grids as desired.

The definition of transformation operators is splitted into two parts. The transformation from coarse to fine grids is denoted as interpolation / prolongation and the transformation from fine to coarse grid is called restriction. The prolongation is defined as a linear interpolation. In the one-dimensional case this is given by

$$v_{2j}^{h} = v_{j}^{2h}$$

if the point is both on the fine and on the coarse grid and by

$$v_{2j+1}^{h} = \frac{1}{2}(v_{j}^{2h} + v_{j+1}^{2h})$$

with

$$0 \le j \le \frac{n}{2} - 1$$

if the point is only on the fine grid. This relation can be seen in Figure 1.

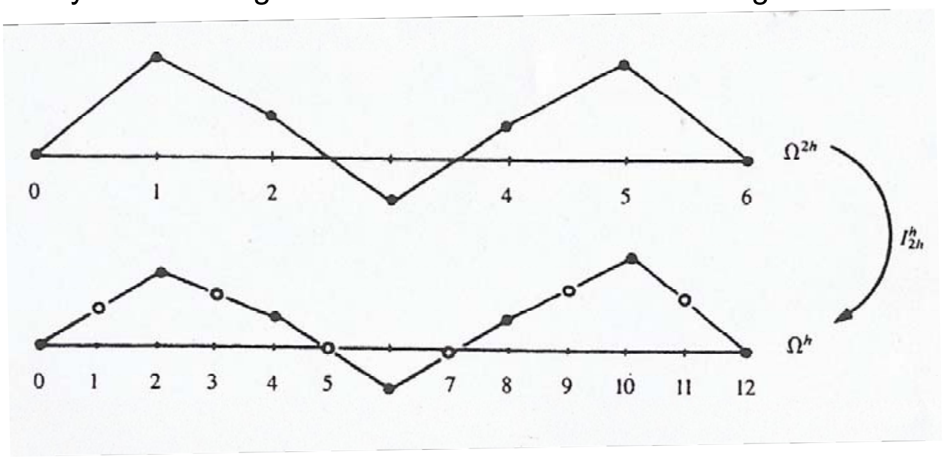

Figure 1: Interpolation / Prolongation in the one-dimensional case

The restriction is normally calculated by full weightening which means that the coarse grid point is defined by an weighted average of its neighbours. For the one-dimensional case it is given by (compare Figure 2)

$$v_j^{2h} = \frac{1}{4}\left(v_{2j-1}^h + 2v_{2j}^h + v_{2j+1}^h\right) \qquad 0 \le j \le \frac{n}{2} - 1.$$

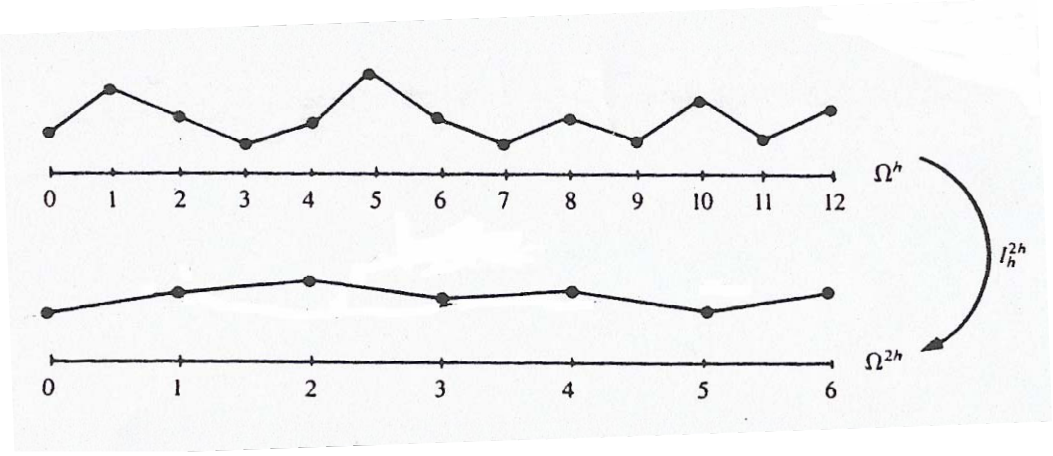

Figure 2: Restriction in the one-dimensional case

For the one-dimensional case the restriction operator and the prolongation operator can also be written in matrix form:

- Restriction operator: $I_h^{2h} = \frac{1}{4}\begin{bmatrix} 1 & 2 & 1 & & & & & \\ & & 1 & 2 & 1 & & & \\ & & & & 1 & 2 & 1 & \\ & & & & & & \cdots & \\ & & & & & & 1 & 2 & 1 \end{bmatrix}$

- Prolongation operator: $I_{2h}^{h} = \frac{1}{2}\begin{bmatrix} 1 & & & & \\ 2 & & & & \\ 1 & 1 & & & \\ & 2 & & & \\ & 1 & 1 & & \\ & & 2 & & \\ & & 1 & & \\ & & & \dots & 1 \\ & & & & 2 \\ & & & & 1 \\ & & & & \end{bmatrix}$

It can be seen that the restriction and the prolongation operator are transpose to each other except for a constant:

$$I_{2h}^{h} = c\left(I_{h}^{2h}\right)^{T} .$$

This property is called variational property. Another important concept is the calculation of the matrix A^{2h} on the coarse grid. This can either be done by discretizing the problem on the coarse problem or in general by multiplying the fine grid matrix A^{h} with the prolongation and restriction operator (called Galerkin condition):

$$A^{2h} = I_{h}^{2h} A^{h} I_{2h}^{h} .$$

Now its possible to apply any multigrid scheme, for example the V-cycle which is based on the correction scheme. It is build up by arbitrary coarse grids and is called V-cycle because it goes from the finest grid down to the coarsest grid and the up again (compare Figure 3):

```
● Relax on $A^{h}\boldsymbol{u}^{h} = \boldsymbol{f}^{h}$ $\nu_1$ times with initial guess $\boldsymbol{v}^{h}$
● Contents $\boldsymbol{f}^{2h} = I_{h}^{2h}\boldsymbol{r}^{h}$
    ● Relax on $A^{2h}\boldsymbol{u}^{2h} = \boldsymbol{f}^{2h}$ $\nu_1$ times with initial guess $\boldsymbol{v}^{2h}$
    ● Compute $\boldsymbol{f}^{4h} = I_{2h}^{4h}\boldsymbol{r}^{2h}$
        ● Relax on $A^{4h}\boldsymbol{u}^{4h} = \boldsymbol{f}^{4h}$ $\nu_1$ with initial guess$\boldsymbol{v}^{4h}$
        ● Compute $\boldsymbol{f}^{8h} = I_{4h}^{8h}\boldsymbol{r}^{4h}$
            .
            ● Solve$A^{Lh}\boldsymbol{u}^{Lh} = \boldsymbol{f}^{Lh}$ (maybe possible with direct solver)
            .
```

        - Correct $\boldsymbol{v}^{4h} \leftarrow \boldsymbol{v}^{4h} + I_{8h}^{4h}\boldsymbol{v}^{8h}$
        - Relax $A^{4h}\boldsymbol{u}^{4h} = \boldsymbol{f}^{4h}$ ν_2 times with initial guess $\boldsymbol{v}^{4h}$
    - Correct $\boldsymbol{v}^{2h} \leftarrow \boldsymbol{v}^{2h} + I_{4h}^{2h}\boldsymbol{v}^{4h}$
    - Relax $A^{2h}\boldsymbol{u}^{2h} = \boldsymbol{f}^{2h}$ ν_2 times with initial guess $\boldsymbol{v}^{2h}$
- Correct $\boldsymbol{v}^{h} \leftarrow \boldsymbol{v}^{h} + I_{2h}^{h}\boldsymbol{v}^{2h}$
- Relax on $A^{h}\boldsymbol{u}^{h} = \boldsymbol{f}^{h}$ ν_2 times with initial guess $\boldsymbol{v}^{h}$

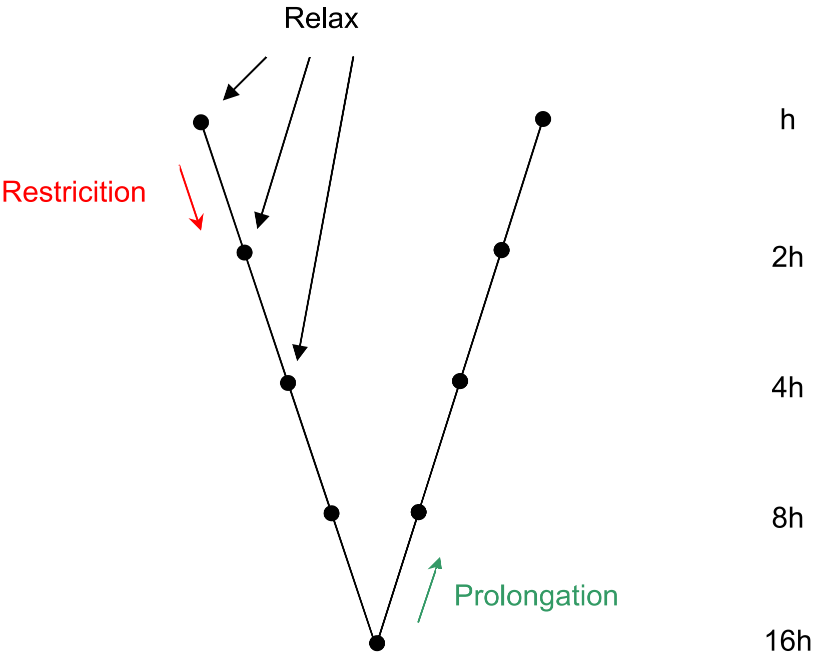

Figure 3: V-cycle

5 Algebraic Multigrid

Geometric multigrid is very effective if a problem is solved on a structured mesh but for many problems an unstructured mesh is necessary where it is difficult to define coarse grid based on geometric relations. Instead one wants to define coarse grids in an algebraic sense. But for transforming a multigrid scheme like the V-cycle from geometric to algebraic multigrid one also needs to define the following things in an algebraic sense:

- Smoothness of an error
- Transfer of vectors and the matrix *A* from coarse to fine grids and vice versa
- Applicability of smoothers (Gauss-Seidel, Jacobi)

In algebraic multigrid an error is defined as a smooth error if

$$A\boldsymbol{e} \approx 0$$

or in components:

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

$$a_{ii}e_i \approx -\sum_{j \neq i} a_{ij}e_j$$

That means a smooth error can be approximated by a weighted average of “neighbouring“ variables.

An important concept for the selection of the coarse grid is the definition of strong influence and dependence. Both definitions take into account that for diagonal dominant matrix the ith row is associated with ith unknown. Of course, it usually takes all of the equation variables to determine any given variable precisely but maybe it is possible to have a preselection of variables that are more important than others

Definition 1:

Given a threshold value $0 < \theta \leq 1$, the variable (point) u_i *strongly depends* on the variable (point) u_j if

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

This says that the variable i strongly depends on grid point j if the coefficient a_{ij} is comparable in magnitude to the largest off-diagonal coefficient in the ith equation. We can state this definition from another perspective.

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.

Two sets can be derived from these two definitions.

S_i : set of points that strongly influence i,

that is the points on which the point i strongly depends.

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

S_i^T : set of points that strongly depend on the point i.

$$S_i^T = \left\{ j : i \in S_j \right\}$$

These sets are important for the selection scheme of the coarse grid. The coarse grid should fulfil the following conditions:

- Smooth error can be approximated accurately
- Good interpolation to the fine grid
- Should have substantially fewer points, so the problem on the coarse grid can be solved with little expense

The selection process is carried out like this:

1) Define a measure to each point of its potential quality as a coarse (C) point: amount λ_i of members of S_i^T

2) Assign point with maximum λ_i to C-point

3) All points in S_i^T become fine (F) points

4) For each new F point j: increase the measeure λ_k for all each unassigned point k that strongly influence j: $k \in S_j$

5) Do 2)-4) until all points are assigned

If the coarse grids are selected, transformation operators must be defined between the grids. In general the component $\left(I_{2h}^h e\right)_i$ of a vector on the fine grid can be determined like this:

$$\left(I_{2h}^h e\right)_i = \begin{cases} e_i & if\ i \in C \\ \sum\limits_{j \in C_i} \omega_{ij} e_j & if\ i \in F \end{cases}$$

The interpolation weights ω_{ij} have information about the neighbouring, off-diagonal elements which can be divided into the neighboring coarse grid points that strongly influence I, the neighboring fine grid points that strongly influence I and into points that do not strongly influence I (fine and coarse grid points).

6 Examples

For testing the algebraic multigrid a test case (compare [3]) was set up. The Poisson equation is solved in four different variants at four different domains:

The coefficients a, b and c are different on each domain:

a=1000
c=1
b=0
	a=1
	c=1
	b=2

a=1
c=1
b=0
	a=1000
	c=1
	b=0

The equation is discretized using the finite volume method and solved on a 100 x 100 mesh. Figure 4 till Figure 6 show the three first coarse levels with red points being the coarse variables on the next grid.

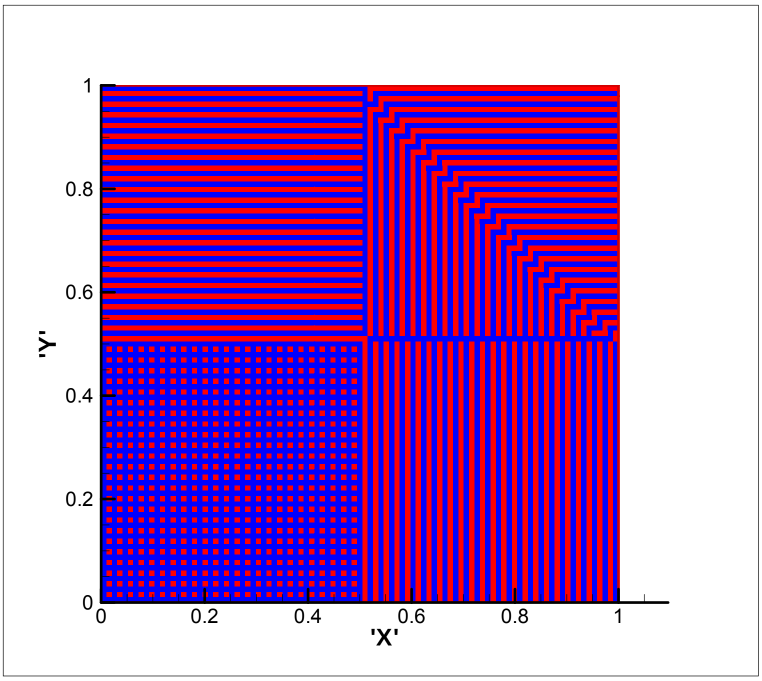

Figure 4: Grid 2h

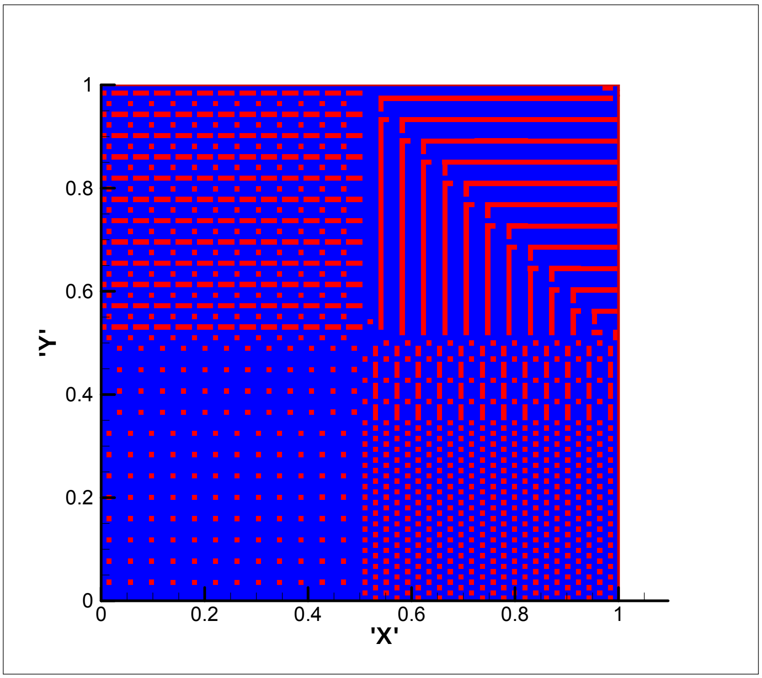

Figure 5: Grid 4h

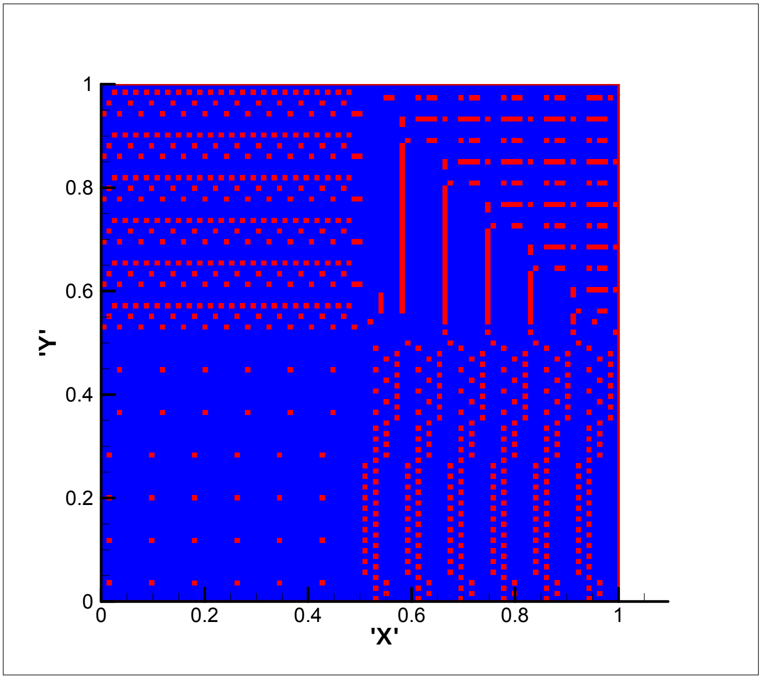

Figure 6: Grid 8h

7. Advantages and Disadvantages of Algebraic Multigrid

Advantages of algebraic multgrid are that the solution is fast and robust and is very effective for segregated solvers.

Disadvantages are that the triple matrix operation at the Galerkin step is a very expensive step and hard to parallise. The definition of coarse grids leads to a high setup-phase and high storage requirements. Additionally this algorithm is not suitable for coupled solvers where the aggregation based algebraic multigrid is a cure against it [1].

Bibliography

[1] M. Raw, *Robustness of coupled Algebraic Multigrid for the Navier-Stokes equations*, AIAA Paper 96-0297 (1996)

[2] W. Briggs, V. Henson und S. MacCormick:, *A multigrid tutorial*; 2. edition;
SIAM Philadelphia (2000);

[3] K. Stüben, *Algebraic Multigrid (AMG): An Introduction with Applications*; GMD Report 70 (1999);