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

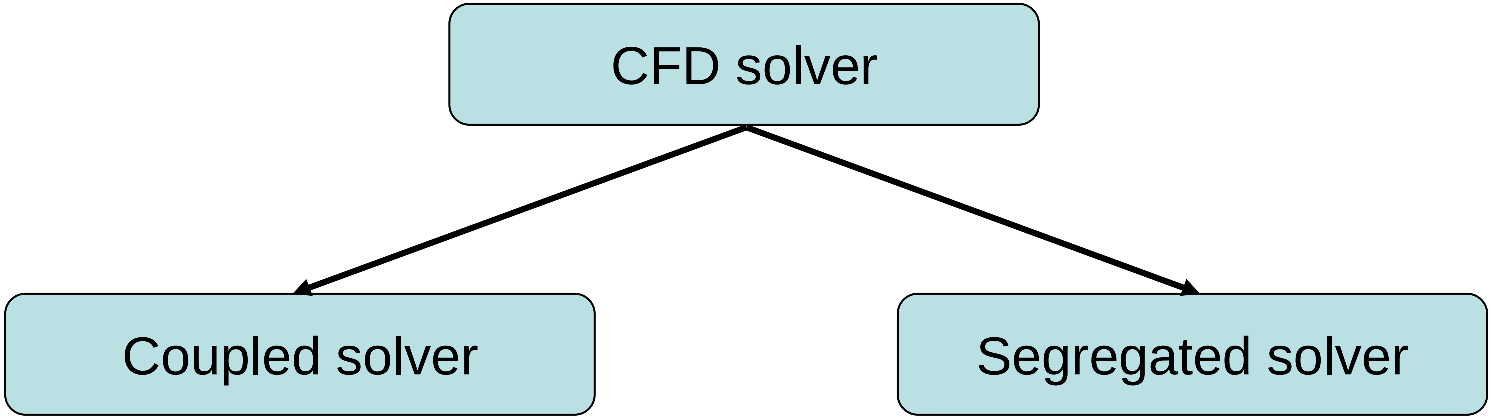
They are among the fastest solution techniques known today

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

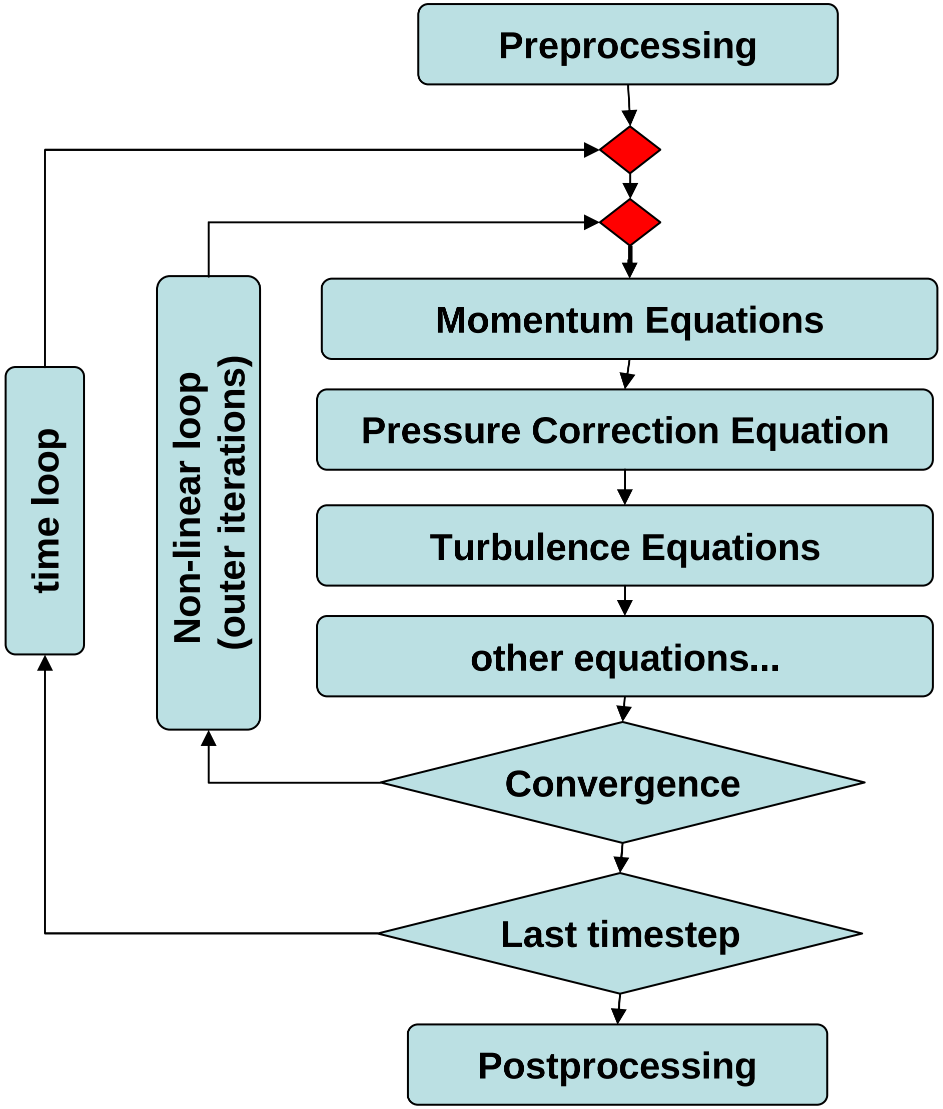
Segregated solver

SIMPLE Algorithm

Typical design of CFD solver

- segregated, sequential solution of decoupled transport equations
- pressure correction equation: a tight tolerance for guaranteeing mass conservation

→ Multigrid methods



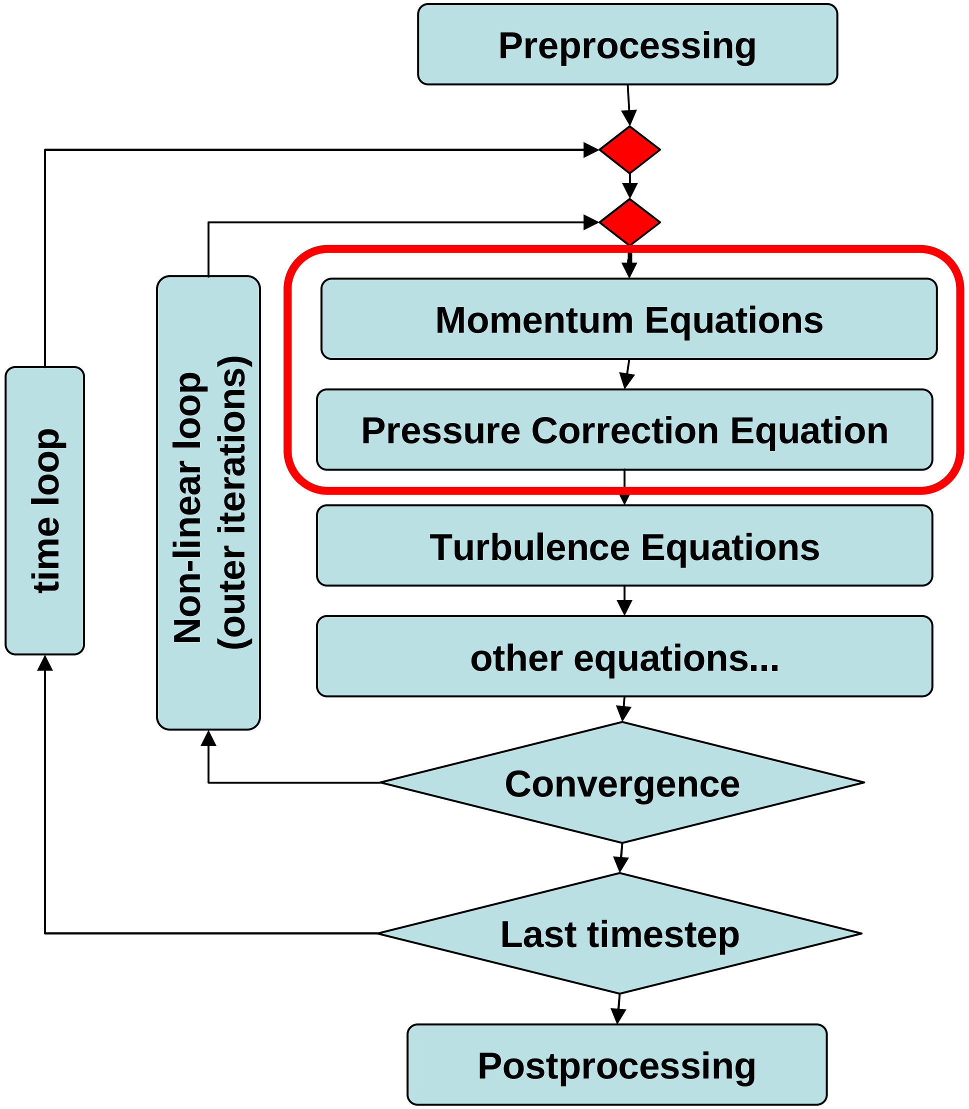
Coupled Solution Algorithm

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

$$\begin{pmatrix} \underline{\underline{A}}_{uu} & \underline{\underline{0}} & \underline{\underline{0}} & \underline{\underline{A}}_{uP} \\ \underline{\underline{0}} & \underline{\underline{A}}_{vv} & \underline{\underline{0}} & \underline{\underline{A}}_{vP} \\ \underline{\underline{0}} & \underline{\underline{0}} & A_{ww} & \underline{\underline{A}}_{wP} \\ \underline{\underline{A}}_{Pu} & \underline{\underline{A}}_{Pv} & \underline{\underline{A}}_{Pw} & \underline{\underline{A}}_{PP} \end{pmatrix} \begin{bmatrix} \vec{u} \\ \vec{v} \\ \vec{w} \\ \vec{p} \end{bmatrix} = \begin{bmatrix} \vec{q}_u \\ \vec{q}_v \\ \vec{q}_w \\ \vec{q}_p \end{bmatrix}$$

Basic Definitions

Methods for Solving Linear Systems of Equations

- Linear System of Equation:

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

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

A: sparse matrix of size n×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*** – *A**v***

- Measured by norms:

L$_\infty$ – norm: $\|\boldsymbol{e}\|_\infty = \max\limits_{1\le j\le n} |e_j|$ L$_2$ - norm: $\|\boldsymbol{e}\|_2 = \left\{ \sum_{j=1}^{n} e_j^2 \right\}^{1/2}$

Direct vs. Iterative Methods

- 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:

$$\frac{-u_{j-1} + 2u_j - u_{j+1}}{h^2} = 0$$

$$-u_{j-1} + 2u_j - u_{j+1} = 0 \qquad 1 \leq j \leq n+1$$

$$u_0 = u_n = 0$$

- Exact solution: $\boldsymbol{u} = 0$

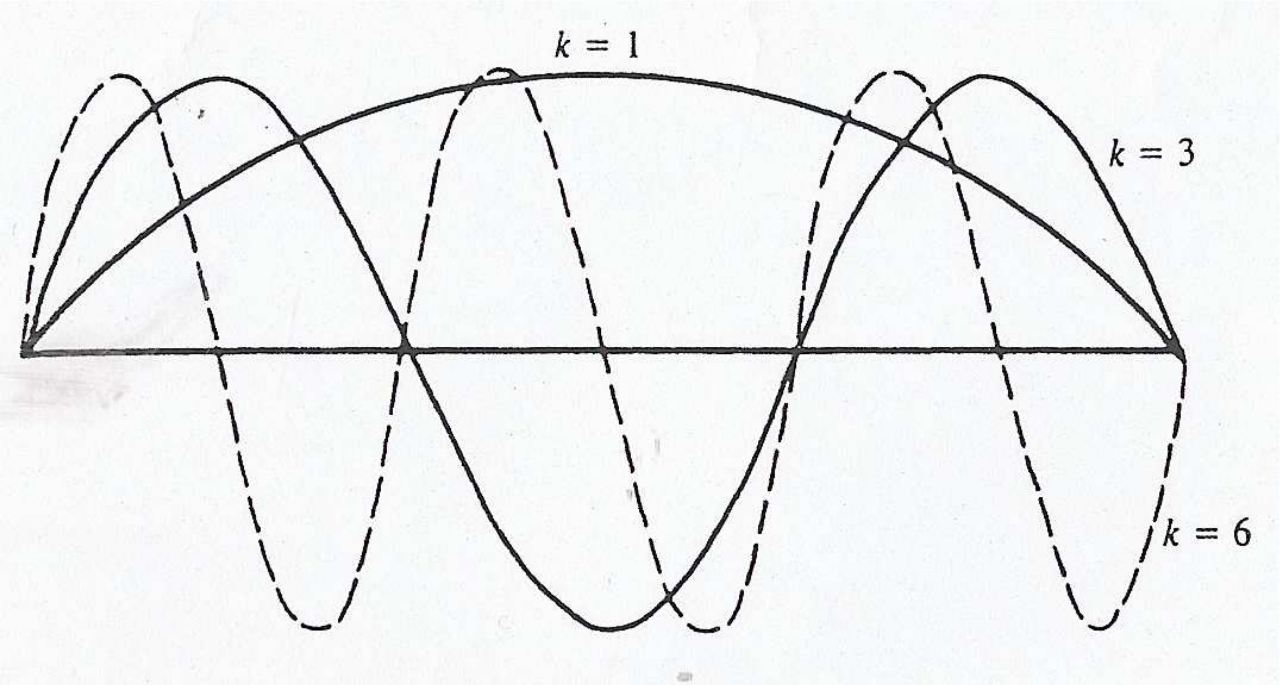
error $\boldsymbol{e} = \boldsymbol{u} – \boldsymbol{v} = -\boldsymbol{v}$

Properties of Iterative methods

Methods for Solving Linear Systems of Equations

- Different starting values:

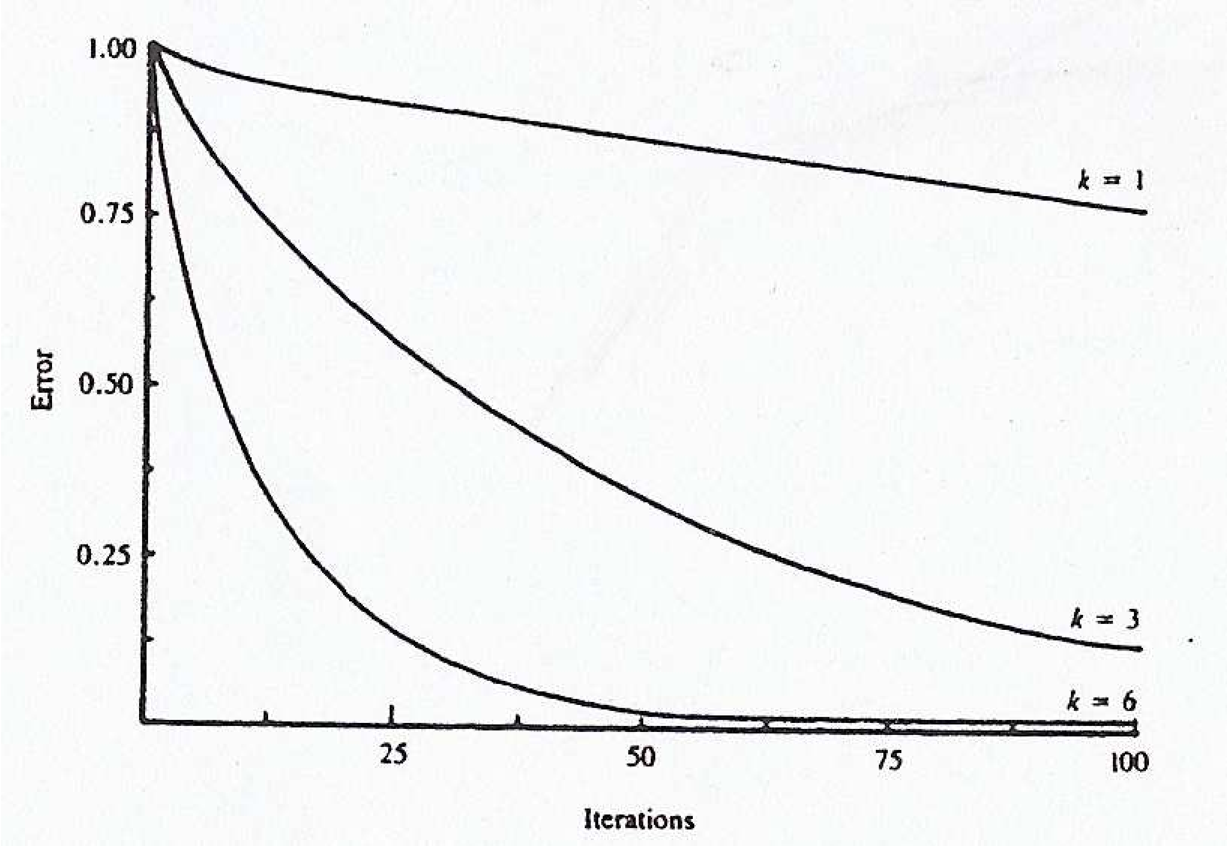
Sinus waves: $$v_j = sin\left(\frac{jk\Pi}{n}\right)$$ Fourier modes



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:

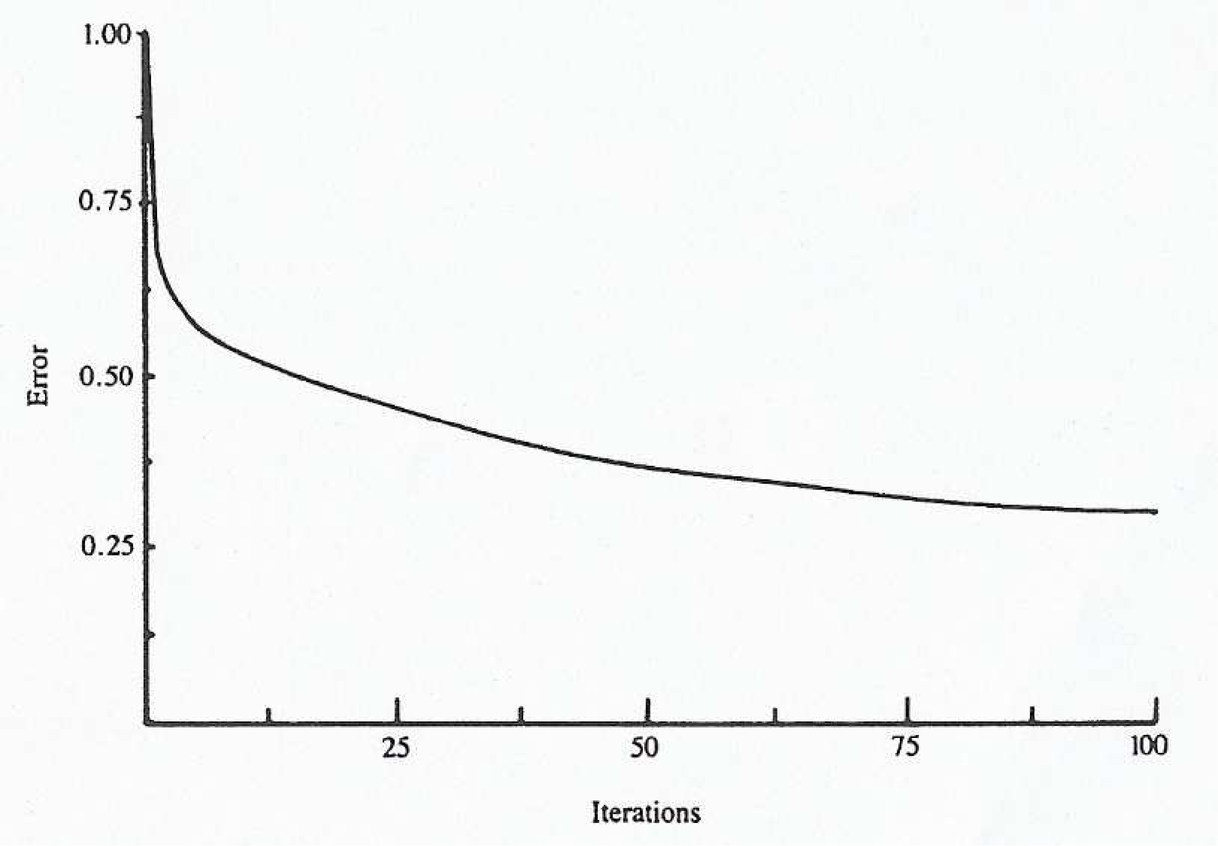
“low frequency wave”

k = 6:

“medium frequency wave”

k = 32:

“high frequency wave”



Properties of Iterative methods

Methods for Solving Linear Systems of Equations

- Error: written in eigenvectors of *A:*

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

- Eigenvectors correspond to the modes of the problem

- Our problem: $w_{k,j} = sin\left(\frac{jk\Pi}{n}\right)$ $\quad 1 \le k \le n-1$

$$1 \le k \le \frac{n}{2} \qquad \frac{n}{2} \le k \le n-1$$

Low frequency modes

„Do not dissappear“

High frequency modes

„Disappear“

Smoother

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

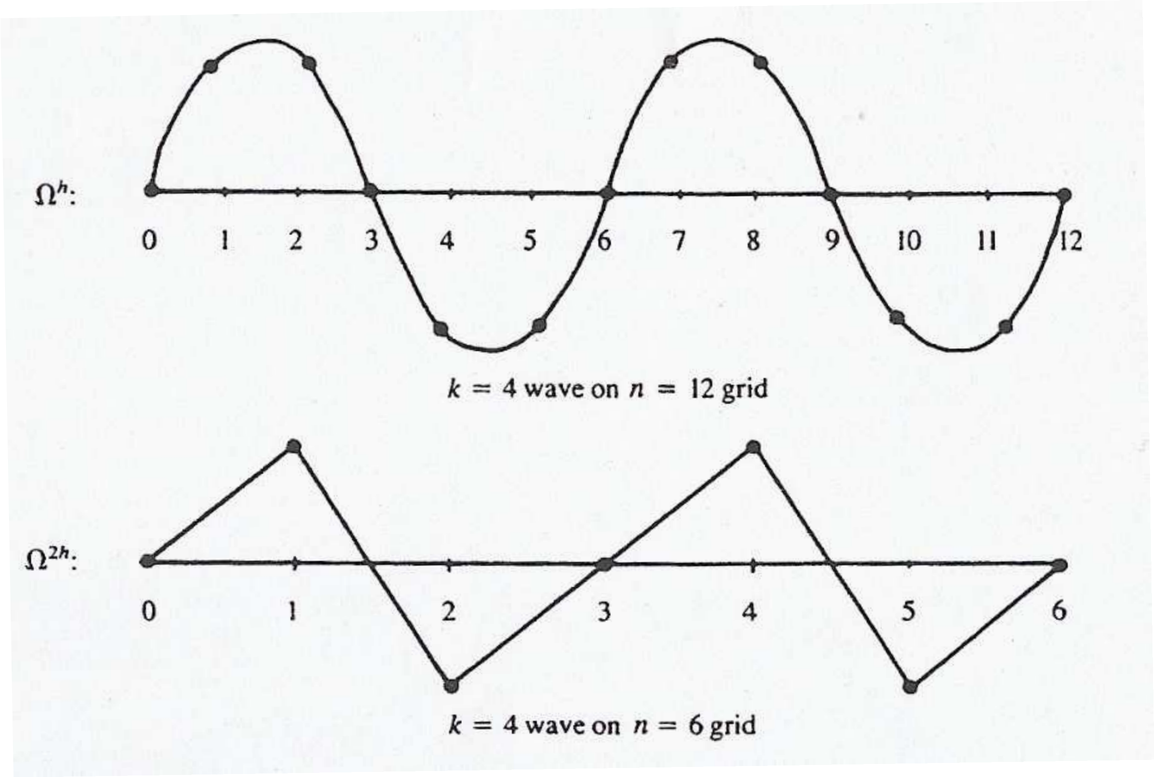
- How does an error look like on a coarse grid?

  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$\boldsymbol{u}$ = $\boldsymbol{f}$ on a very coarse grid

    to obtain an initial guess for the next finer grid

    .

    .

    .
  - Relax on A$\boldsymbol{u}$ = $\boldsymbol{f}$ on Ω^{4h} to obtain an initial guess for Ω^{2h}
  - Relax on A$\boldsymbol{u}$ = $\boldsymbol{f}$ on Ω^{2h} to obtain an initial guess for Ω^{h}
  - Relax on A$\boldsymbol{u}$ = $\boldsymbol{f}$ on Ω^{h} to obtain a final approximation to the solution.

- Problems: Relax on A$\boldsymbol{u}$ = $\boldsymbol{f}$ on Ω^{2h}?

  Last iteration: Error still smooth?

- 2nd possibility: Use of the residual equation

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

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

$$A\boldsymbol{e} = \boldsymbol{r}$$

Possible schemes for improvement

Geometric Multigrid

- **Correction scheme:**

  - Relax on A***u*** = ***f*** on Ω^h to obtain an approximation ***v****ʰ*
  - Compute the residual ***r*** = ***f*** - *A* ***v****ʰ*

    Relax on the residual equation A***e*** = ***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}: $\boldsymbol{v}^h \leftarrow \boldsymbol{v}^h + \boldsymbol{e}^{2h}$

- Problems: Relax on A***e*** = ***r*** on Ω^{2h}?

  Transfer from Ω^{2h} to Ω^h?

Transfer operators

Geometric Multigrid

- Transfer from coarse to fine grids: **Interpolation / Prolongation**

$$\Omega^{2h} \rightarrow \Omega^{h}$$

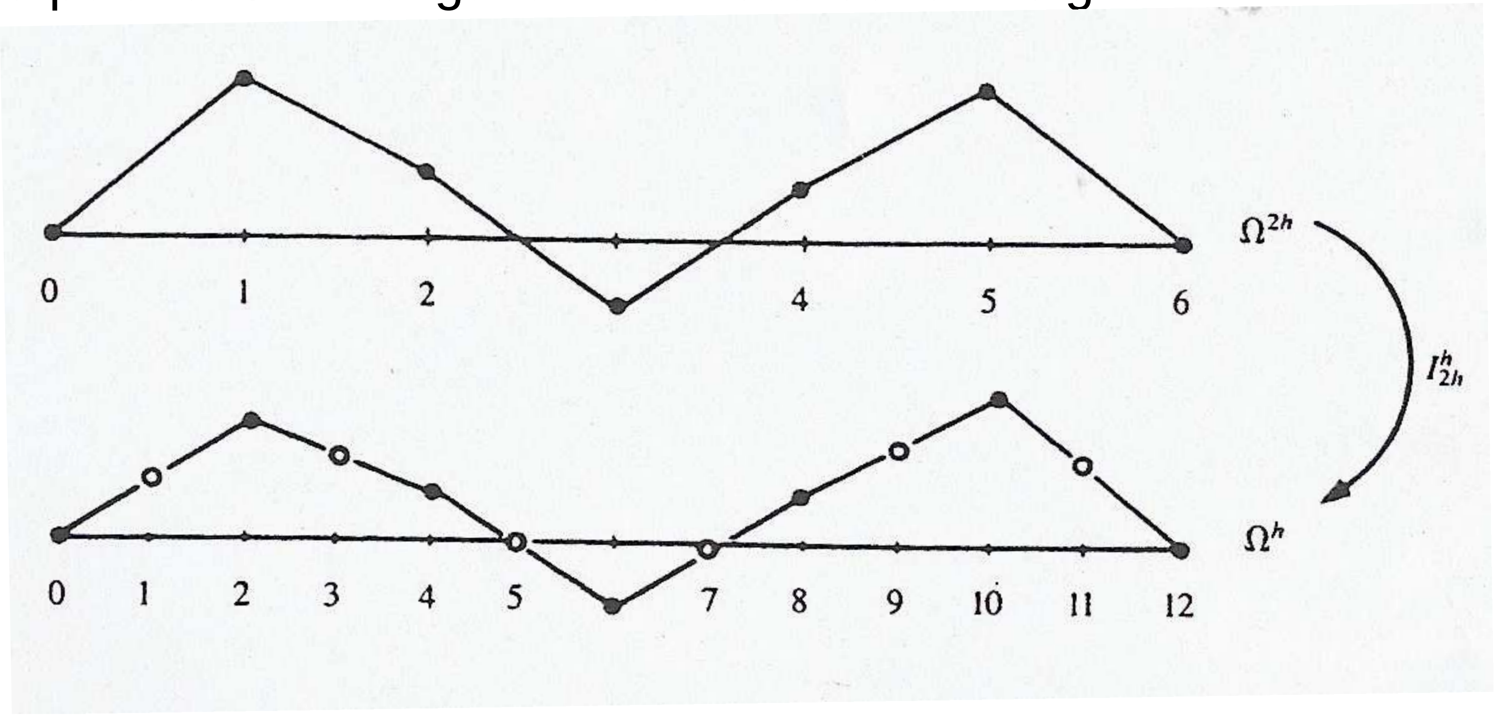
- Transfer from fine to coarse grids: **Restriction**

$$\Omega^{h} \rightarrow \Omega^{2h}$$

Transfer operators – Interpolation / Prolongation

Geometric Multigrid

- Interpolation / Prolongation: from coarse to fine grid



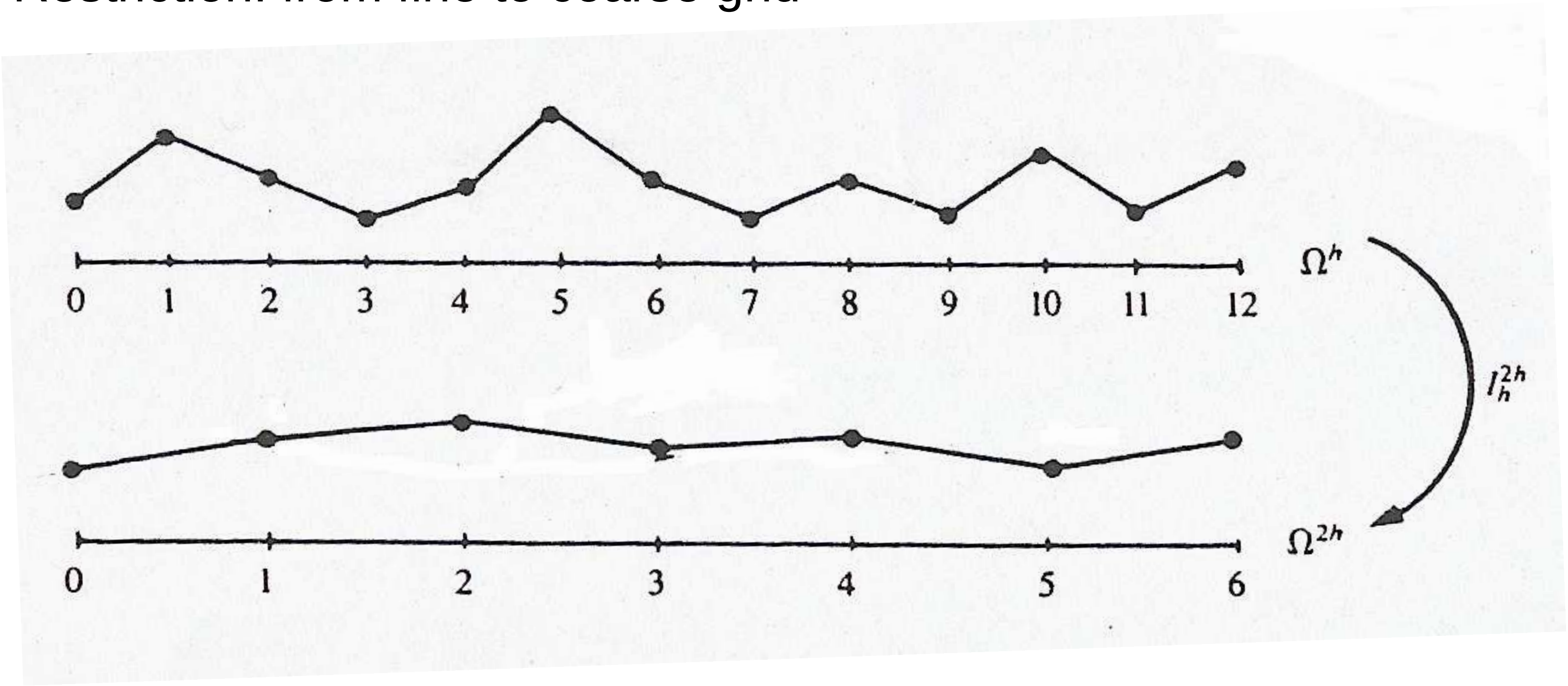
- Points on fine and on coarse grid: $v_{2j}^{h} = v_{j}^{2h}$

- 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



- Full weigthening: $v_j^{2h} = \frac{1}{4}\left(v_{2j-1}^h + 2v_{2j}^h + v_{2j+1}^h\right)$

Properties of transfer operators

Geometric Multigrid

Interpolation / Prolongation

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

Restriction

$$I_{h}^{2h} = \frac{1}{4}\begin{bmatrix} 1 & 2 & 1 & & & & & & \\ & & 1 & 2 & 1 & & & & \\ & & & & 1 & 2 & 1 & & \\ & & & & & & \dots & & \\ & & & & & & 1 & 2 & 1 \end{bmatrix}$$

Variational property: $I_{2h}^{h} = c\left(I_{h}^{2h}\right)^{T}$

Properties of transfer operators

- Transfer of vectors: ✓
- Transfer of matrix *A*: $A^h \to A^{2h}$
  - Geometric answer: A^{2h} is discretisation of the problem on the coarse grid
  - Algebraic answer: $A^{2h} = I_h^{2h} A^h I_{2h}^h$ (**Galerkin condition**)

Recapitulation

Geometric Multigrid

- 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:

Galerkin condition $A^{2h} = I_h^{2h} A^h I_{2h}^h$

Variational property $I_{2h}^h = c\left(I_h^{2h}\right)^T$

How can we put this in a good solution algorithm?

V-Cycle

Geometric Multigrid

- Relax on $A^h \boldsymbol{u}^h = \boldsymbol{f}^h$ ν_1 times with initial guess $\boldsymbol{v}^h$
- Compute $\boldsymbol{f}^{2h} = I_h^{2h} \boldsymbol{r}^h$
  - Relax on $A^{2h} \boldsymbol{u}^{2h} = \boldsymbol{f}^{2h}$ ν_1 times with initial guess $\boldsymbol{v}^{2h}$
  - Compute $\boldsymbol{f}^{4h} = I_{2h}^{4h} \boldsymbol{r}^{2h}$
    - Relax on $A^{4h} \boldsymbol{u}^{4h} = \boldsymbol{f}^{4h}$ ν_1 times with initial guess $\boldsymbol{v}^{4h}$
    - Compute $\boldsymbol{f}^{8h} = I_{4h}^{8h} \boldsymbol{r}^{4h}$

      …

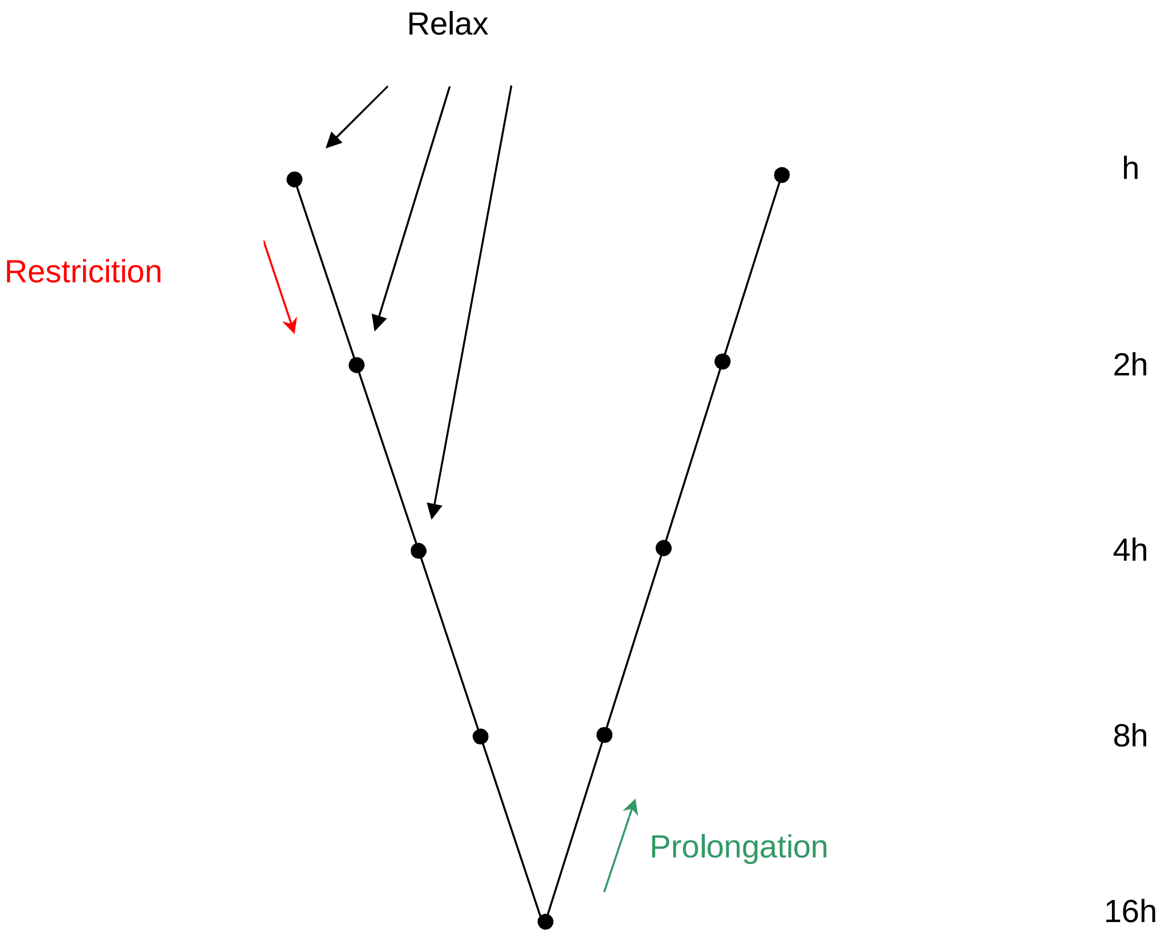
      - Solve $A^{Lh} \boldsymbol{u}^{Lh} = \boldsymbol{f}^{Lh}$

      …

    - 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 $A^h \boldsymbol{u}^h = \boldsymbol{f}^h$ ν_2 times with initial guess $\boldsymbol{v}^h$

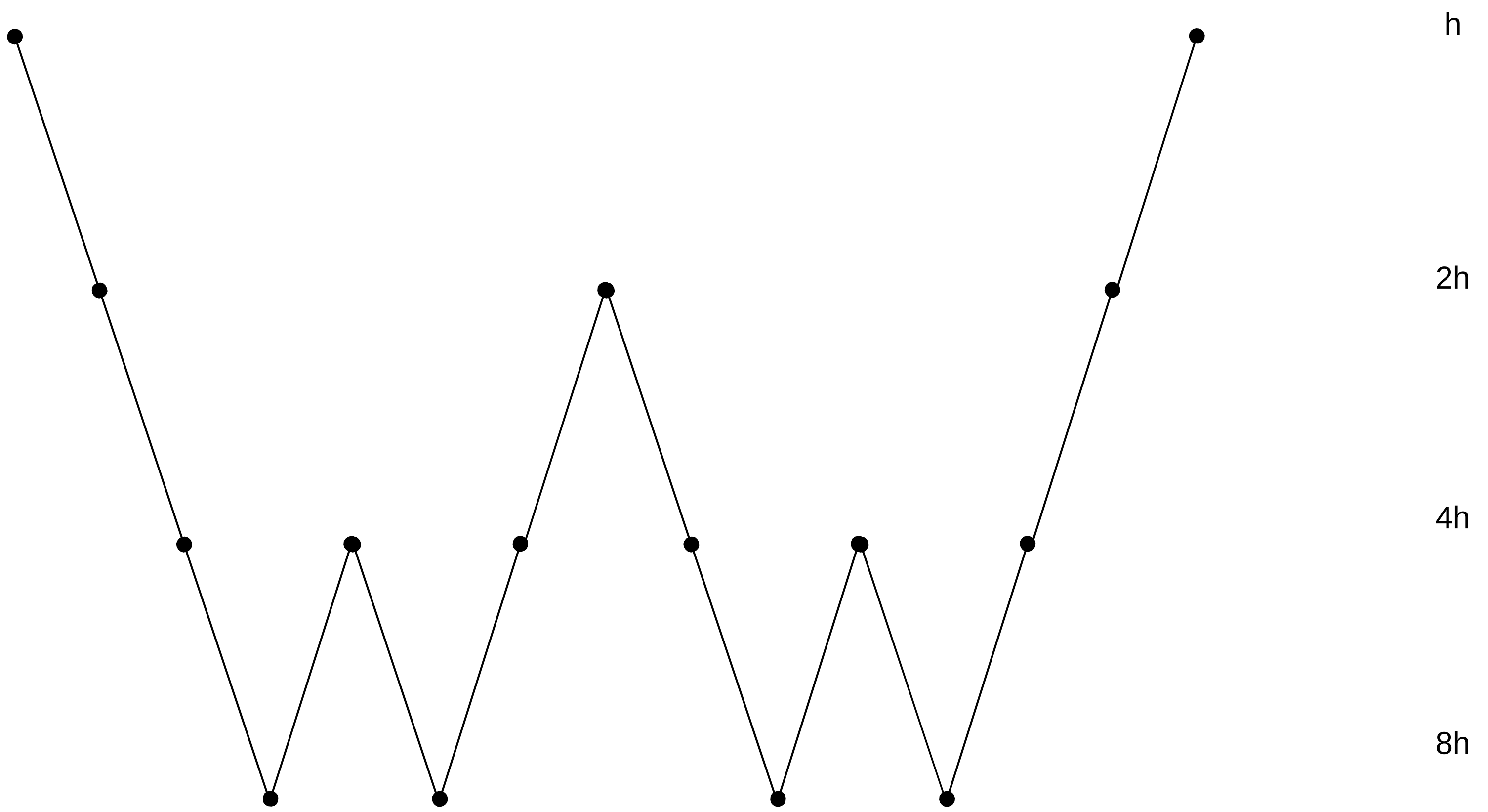
V-Cycle

Geometric Multigrid



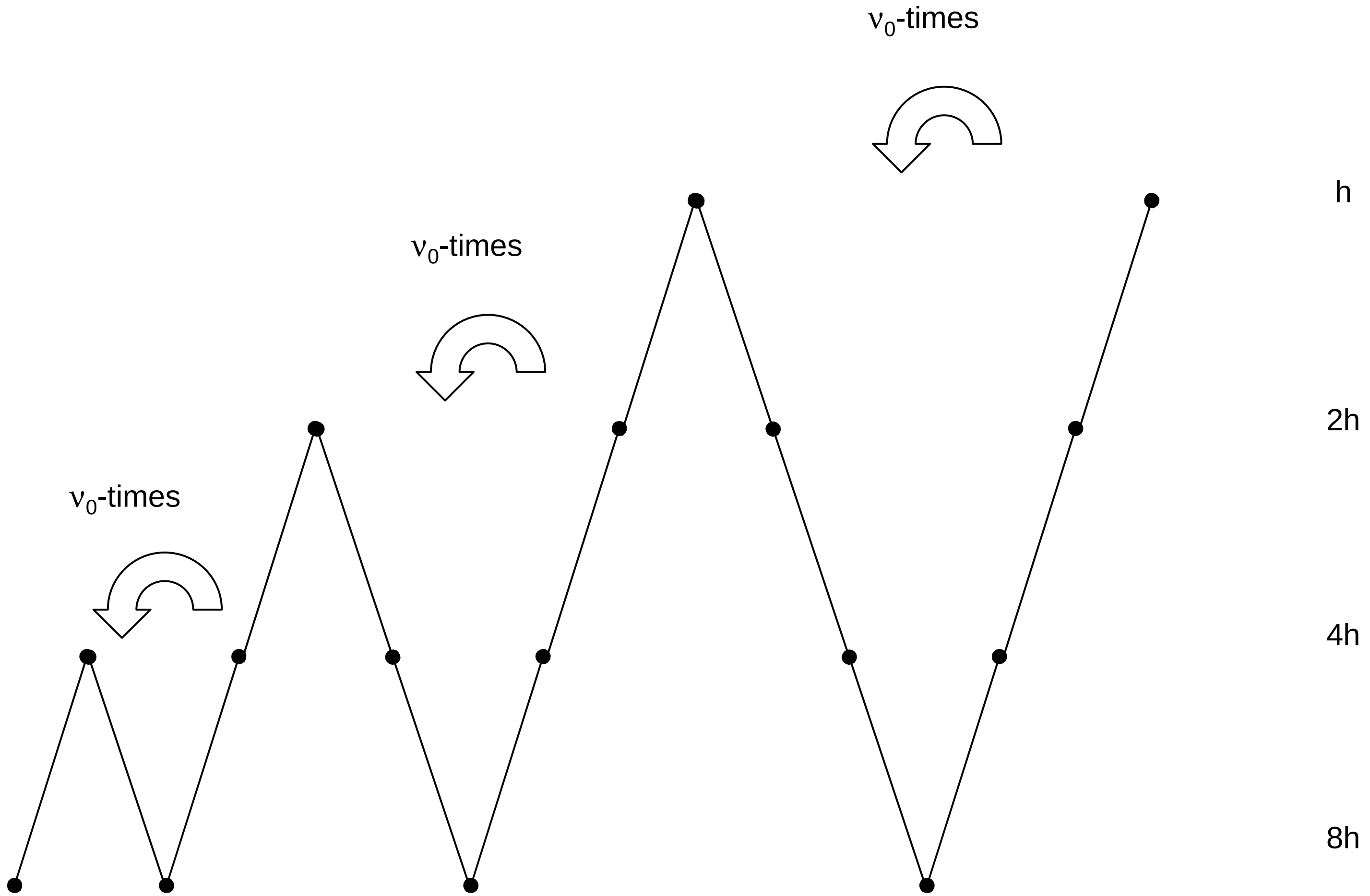
Other cycles – W Cycle

Geometric Multigrid



Other cycles – Full Multigrid Cycle (FMG)

Geometric Multigrid



Geometric vs. Algebraic 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

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

$$\boldsymbol{u} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix}$$

Smooth error

Algebraic Multigrid

- Defined as an error which is not effectively reduced by an iterative method
- Jacobi method: $\boldsymbol{e}^{i+1} = (I - D^{-1}A)\boldsymbol{e}^{i}$
- Measurement of the error with the A-inner product: $\|\boldsymbol{e}\|_A = (A\boldsymbol{e},\boldsymbol{e})^{1/2}$
- Smooth error: $\|(I - D^{-1}A)\boldsymbol{e}\|_A \approx \|\boldsymbol{e}\|_A$

$$\|\boldsymbol{e} - D^{-1}A\boldsymbol{e}\|_A \approx \|\boldsymbol{e}\|_A$$

$$\boxed{\rightarrow \|D^{-1}A\boldsymbol{e}\|_A \ll \|\boldsymbol{e}\|_A}$$

Smooth error

Algebraic Multigrid

- Smooth error: $\left\| D^{-1} A\mathbf{e} \right\|_A \ll \left\| \mathbf{e} \right\|_A$

$$\left(D^{-1} A\mathbf{e}, A\mathbf{e} \right) \ll \left(\mathbf{e}, A\mathbf{e} \right)$$

$$\left(D^{-1} \mathbf{r}, \mathbf{r} \right) \ll \left(\mathbf{e}, \mathbf{r} \right)$$

$$\sum_{i=1}^{n} \frac{r_i^2}{a_{ii}} \ll \sum_{i=1}^{n} r_i e_i$$

$$\rightarrow \left| r_i \right| \ll a_{ii} \left| e_i \right|$$

$$\boxed{A\mathbf{e} \approx 0}$$

Implications of smooth error

Algebraic Multigrid

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

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

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

Selecting the coarse grid - requirements

- 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.

Selecting the coarse grid – Influence and Dependence

- Definition 1:

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

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

- 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

- Two important sets:

S_i: set of points that strongly influence i,
that is the points on which the point i strongly depends.

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

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

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

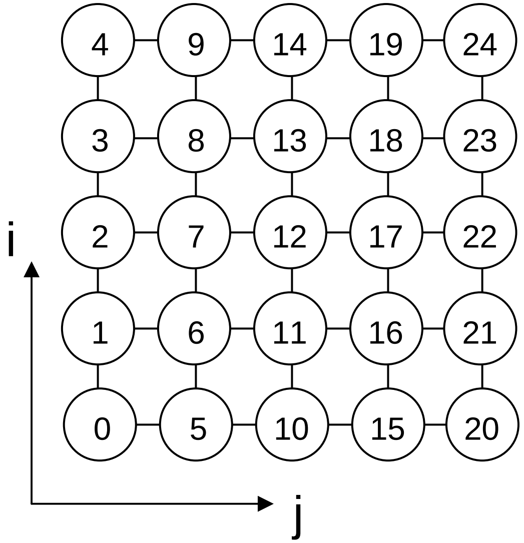
Selecting the coarse grid - Example

Algebraic Multigrid

- Poisson equation: $-\Delta u = 0$

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

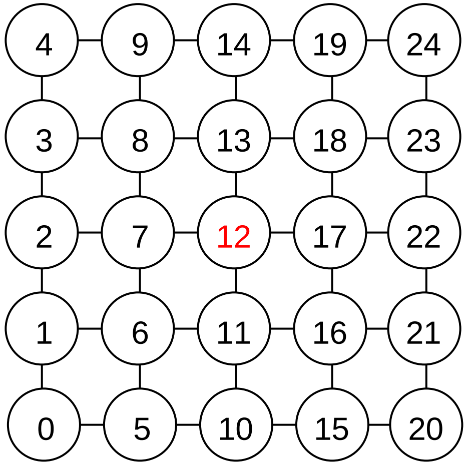
$$\frac{1}{h^2}\left(-u_{i-1}-u_{i+1}+4u_i-u_{j-1}-u_{j+1}\right)=0$$



Selecting the coarse grid - Example

Algebraic Multigrid

Discretisation on 5x5 grid:



For example, Point 12:

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

$$a_{12,7} = -1$$

$$a_{12,11} = -1$$

$$a_{12,13} = -1$$

$$a_{12,17} = -1$$

$$a_{12,12} = 4$$

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

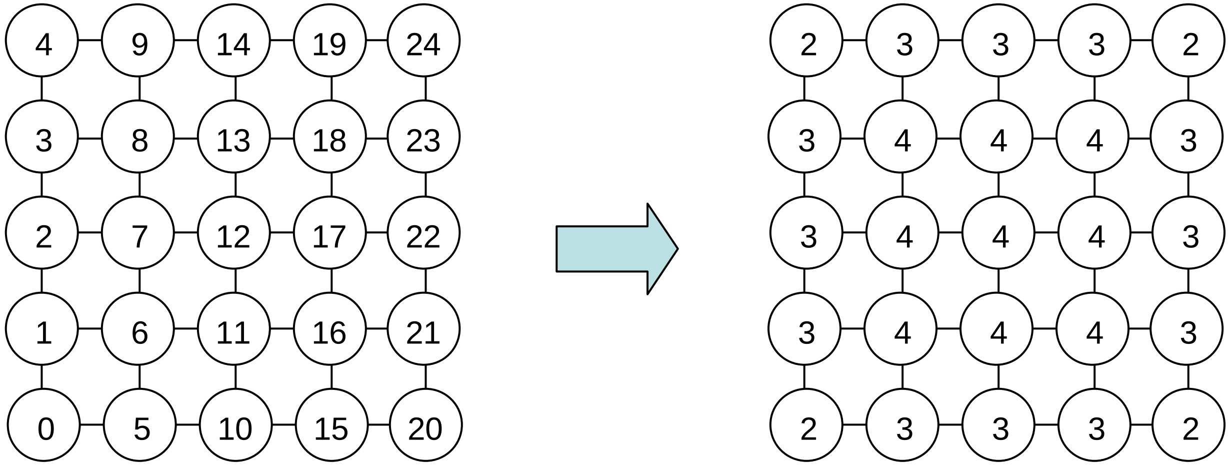
$$S_i^T = \left\{ j : i \in S_j \right\}$$

$$S_{12} = \left\{ 7, 11, 13, 17 \right\}$$

$$S_{12}^T = \left\{ 7, 11, 13, 17 \right\}$$

Selecting the coarse grid - Example

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



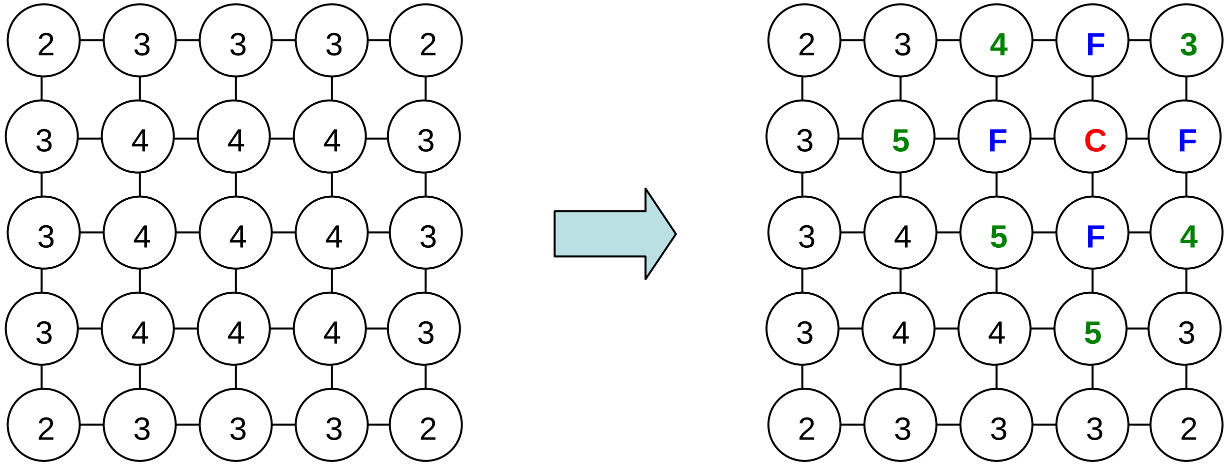
4	9	14	19	24
3	8	13	18	23
2	7	12	17	22
1	6	11	16	21
0	5	10	15	20

2	3	3	3	2
3	4	4	4	3
3	4	4	4	3
3	4	4	4	3
2	3	3	3	2

Selecting the coarse grid - Example

Algebraic Multigrid

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: $k \in S_j$



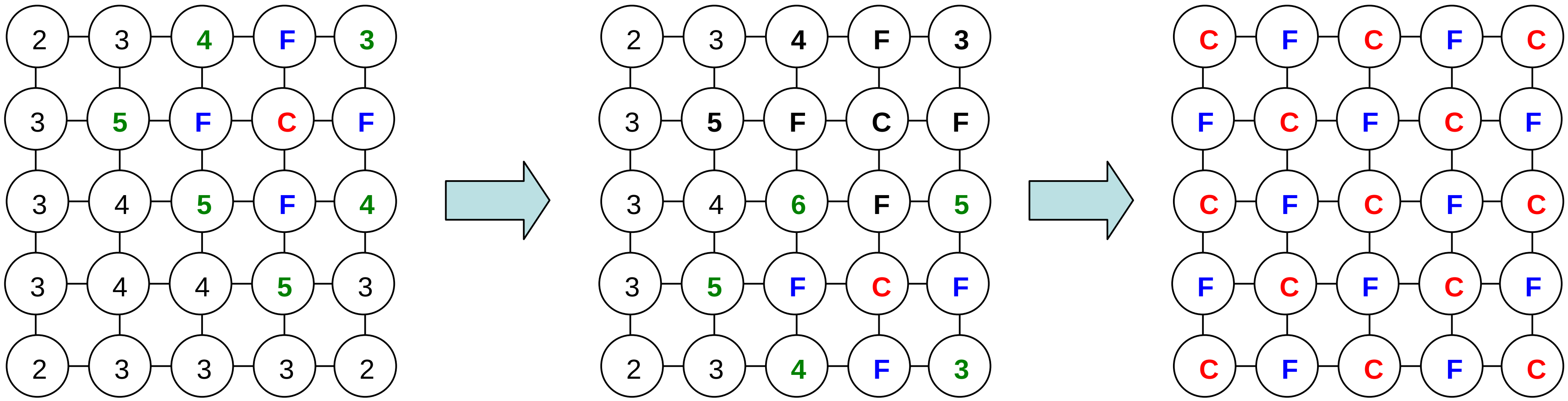
2	3	3	3	2
3	4	4	4	3
3	4	4	4	3
3	4	4	4	3
2	3	3	3	2

2	3	4	F	3
3	5	F	C	F
3	4	5	F	4
3	4	4	5	3
2	3	3	3	2

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

Selecting the coarse grid - Example

Algebraic Multigrid



2	3	4	F	3
3	5	F	C	F
3	4	5	F	4
3	4	4	5	3
2	3	3	3	2

2	3	4	F	3
3	5	F	C	F
3	4	6	F	5
3	5	F	C	F
2	3	4	F	3

C	F	C	F	C
F	C	F	C	F
C	F	C	F	C
F	C	F	C	F
C	F	C	F	C

- Interpolation: from coarse to fine grids

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

- 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

→ This information is contained in ω_{ij}

Example

Algebraic Multigrid

- Example: $-au_{xx} - cu_{yy} + bu_{xy} = 0$
- Discretised with a two-dimensional mesh, divided into 4 parts;

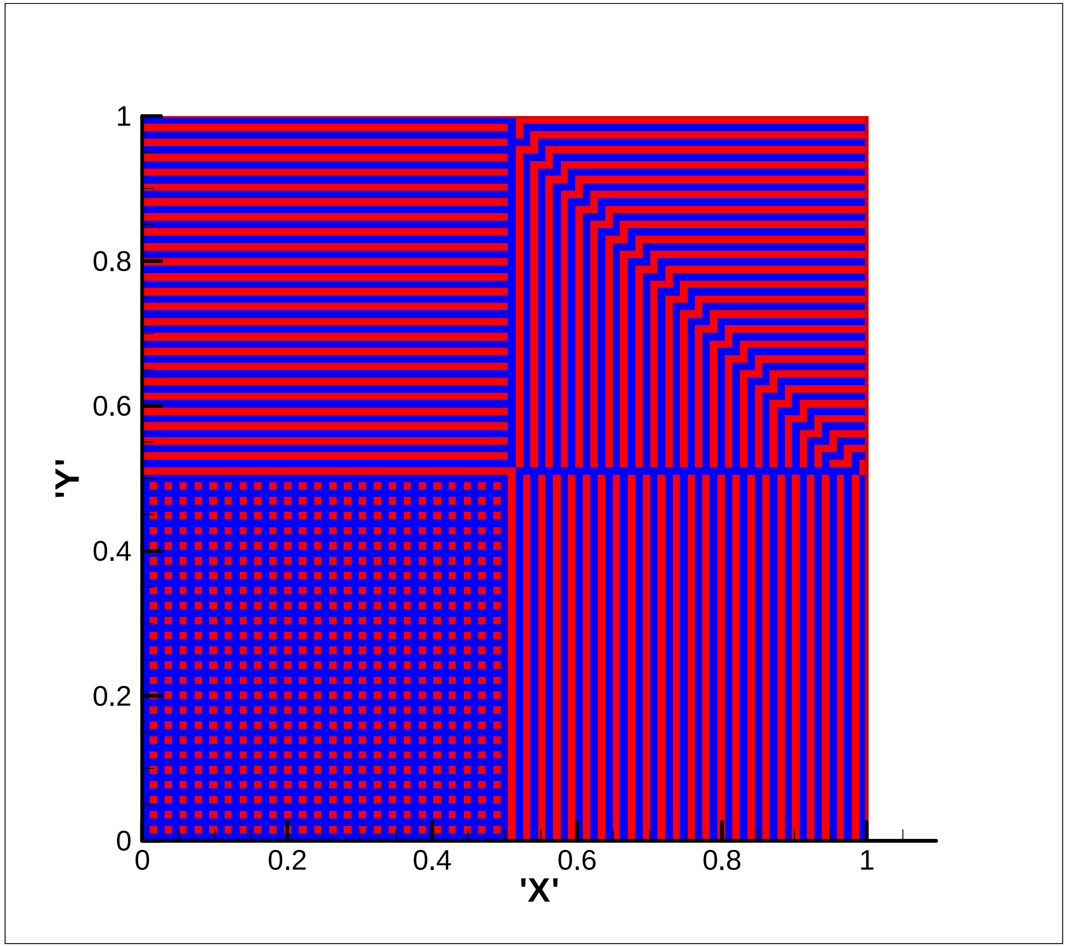
a=1000
c=1
b=0
	a=1
	c=1
	b=2

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

Example

Algebraic Multigrid

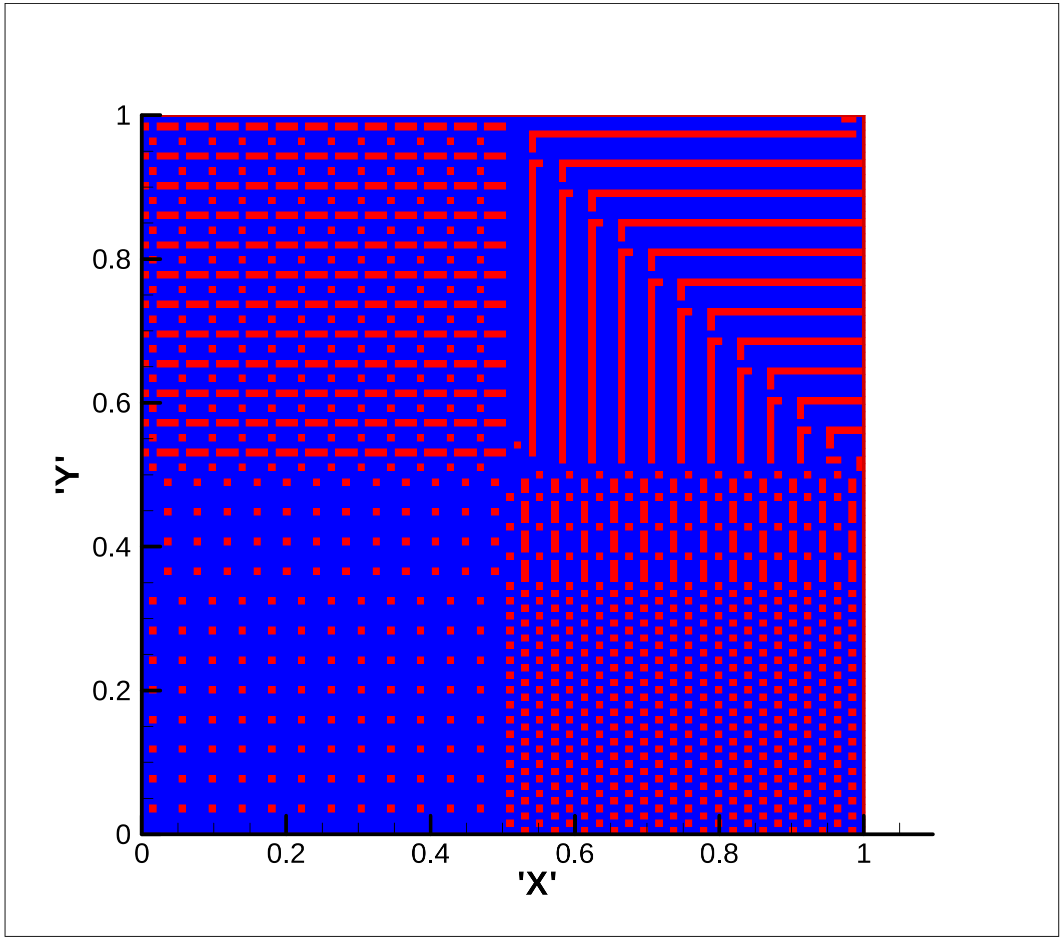
Grid 2h



Example

Algebraic Multigrid

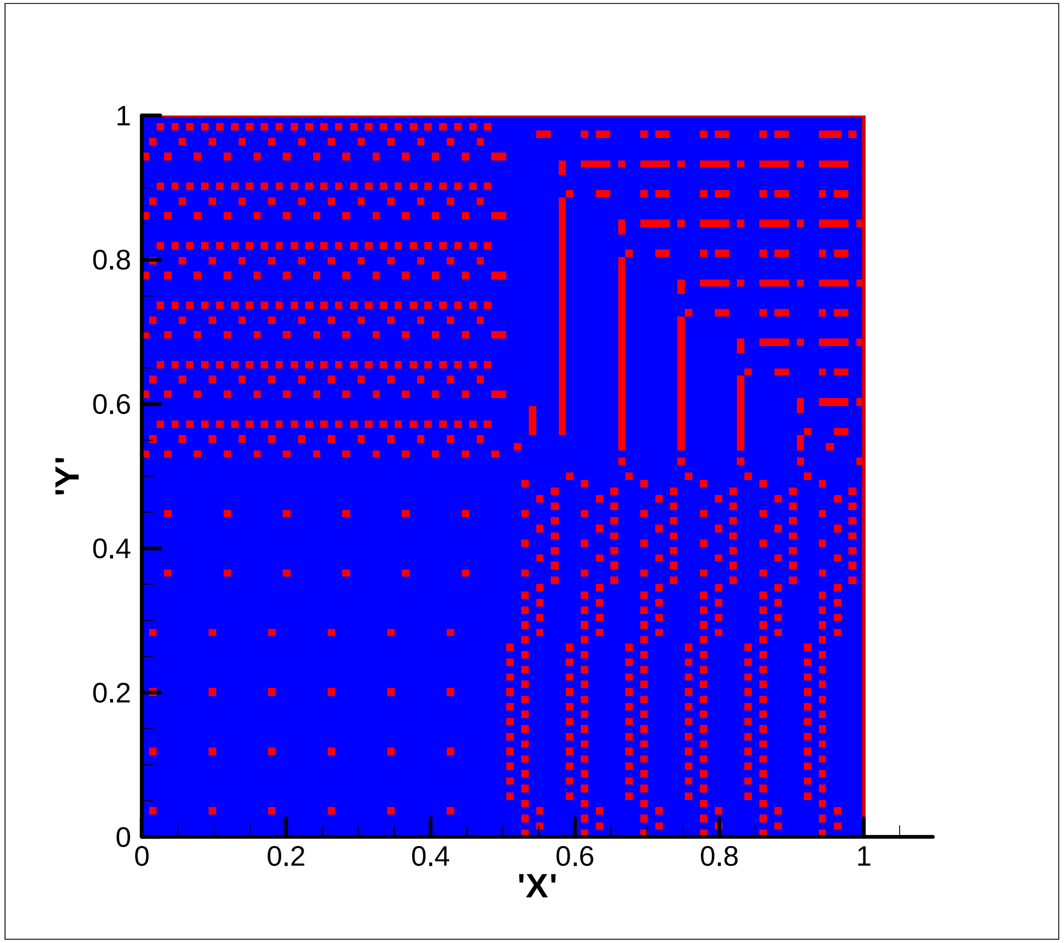
Grid 4h



Example

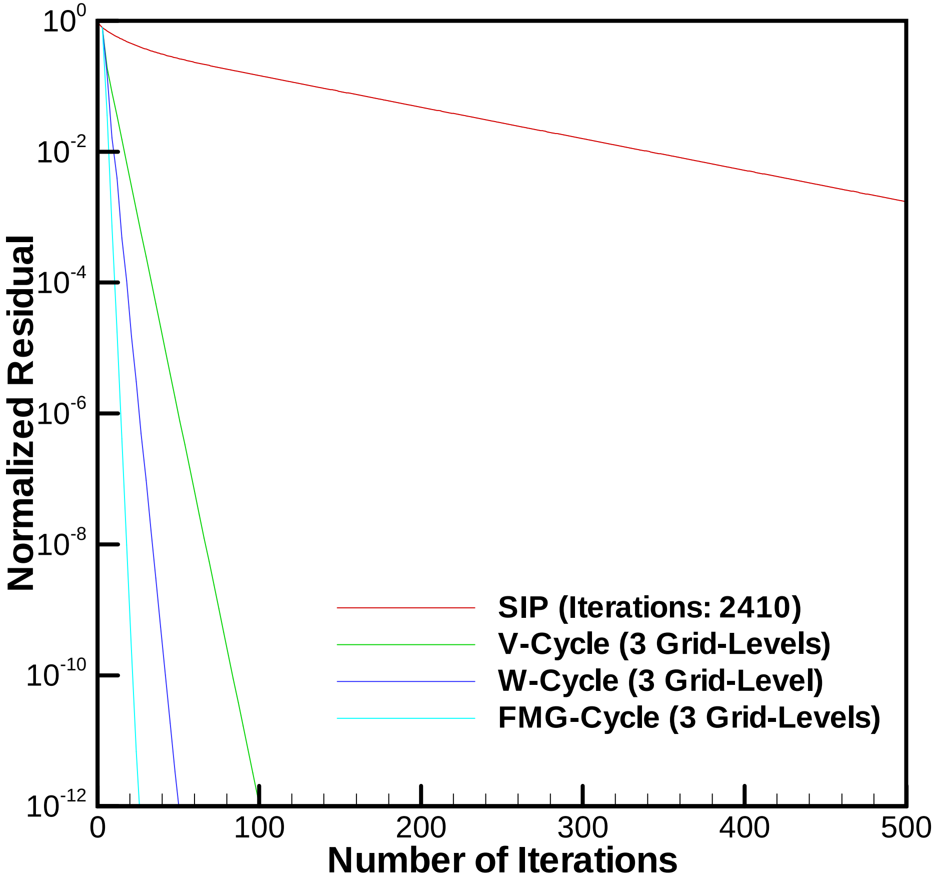
Algebraic Multigrid

Grid 8h



Advantages & Disadvantages of AMG

Algebraic Multigrid



Advantages & Disadvantages of AMG

Algebraic Multigrid

Advantages

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

Disadvantages

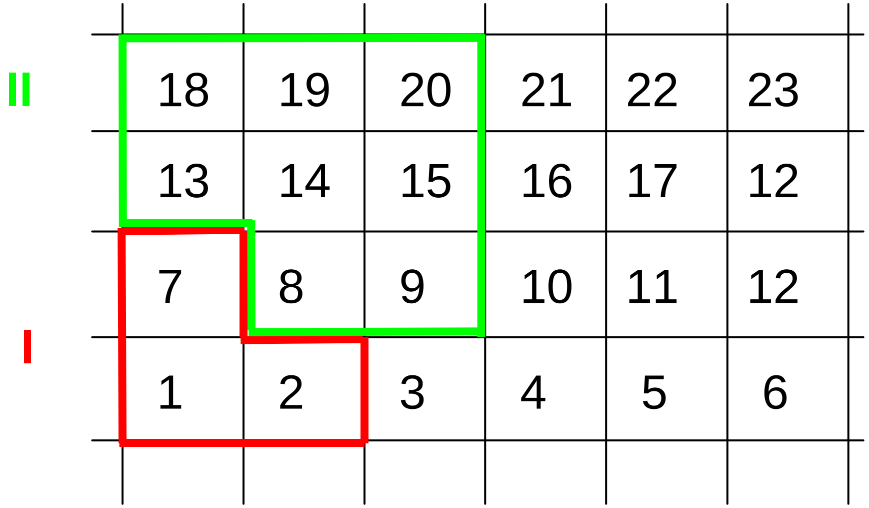
- The Galerkin Operation is a very expensive step
- Diffucult to parallelize
- High setup-phase
- High storage requirements
- Not for coupled solvers

→ A cure are the **aggregation based AMGs**

Aggregation based AMG

Algebraic Multigrid

- In the simplest case strongly connected coefficient are simply summed up
- Example:

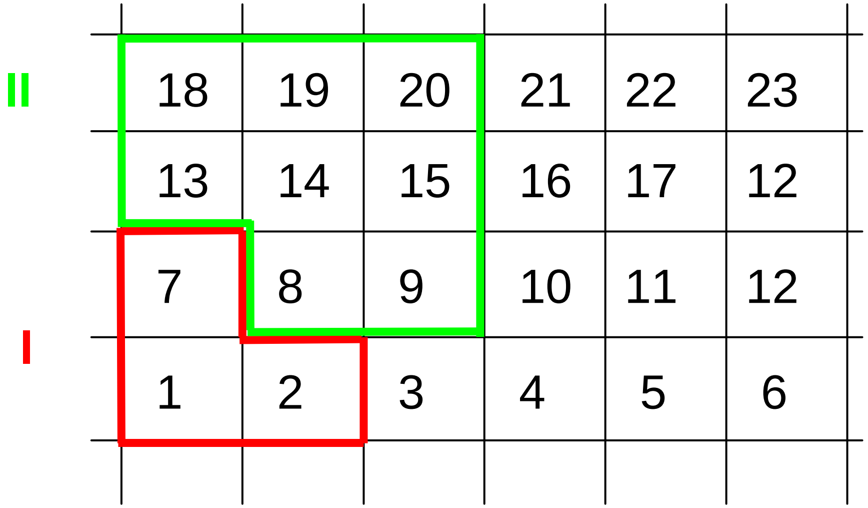


18	19	20	21	22	23
13	14	15	16	17	12
7	8	9	10	11	12
1	2	3	4	5	6

- 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



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

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

Aggregation based AMG

Algebraic Multigrid

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