# The Multigrid Method

Nicolas Alt · n.alt@mytum.de
University of Technology, Munich
JASS 2006

- **Model Problems**
- Relaxation Methods
- Error convergence
- Multiple grids
- Performance
- Theoretical Considerations

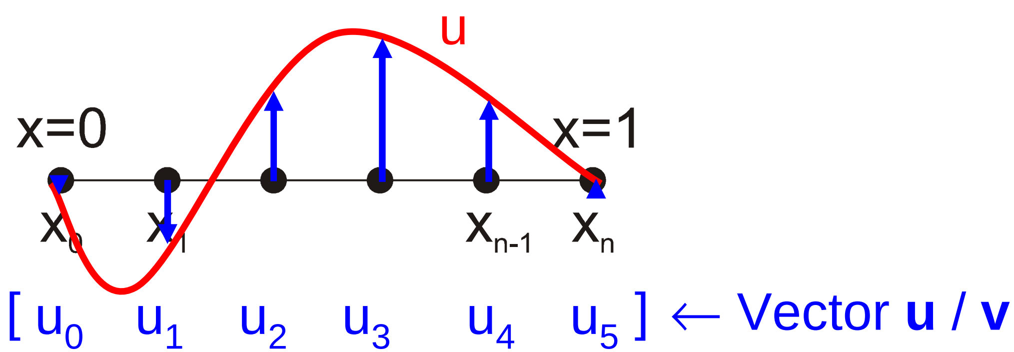
- Differential Equation in 1D

$$-u''(x) + au(x) = f(x)$$

for $0 < x < 1$ , $a > 0$

Boundary: $u(0) = u(1) = 0$

- Partition continuous problem into n subintervals by "sampling" it at the grid points $x_j = jh$, with $h = 1/n$
- Grid $\Omega^h$:



- Second order finite difference approximation

$$\frac{-v_{j-1}+2v_j-v_{j+1}}{h^2}+av_j = f(x_j) \text{ with } 1 \le j \le n-1$$

$v_0 = v_n = 0$; with **v** being the approximate solution to **u**

- Written in Matrix-Vector form

$$\frac{1}{h^2}\begin{bmatrix} 2+ah^2 & -1 & & \\ -1 & 2+ah^2 & -1 & \\ & & \ddots & \\ & & -1 & 2+ah^2 \end{bmatrix} \cdot \begin{bmatrix} v_1 \\ \vdots \\ v_{n-1} \end{bmatrix} = \begin{bmatrix} f(x_1) \\ \vdots \\ f(x_{n-1}) \end{bmatrix} = \begin{bmatrix} f_1 \\ \vdots \\ f_{n-1} \end{bmatrix}$$

- Written compactly: $A\mathbf{v} = \mathbf{f}$

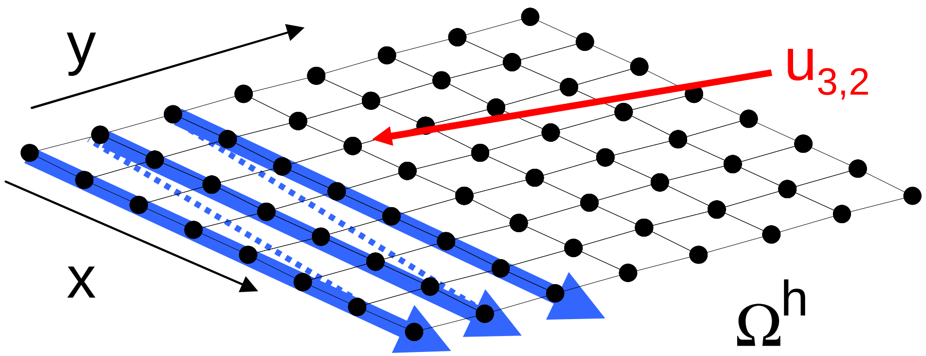
- (Elliptic) Partial Differential Equation

$$-u_{xx} - u_{yy} + au = f(x,y)$$

for 0 < x,y < 1 , a > 0

Boundary: “Frame = 0”

- Sampled with a two-dimensional grid (n-1,m-1 interior grid points)



- Sampling results in difference approximation

$$\frac{-v_{i-1,j}+2v_{ij}-v_{i+1,j}}{h_x^2}+\frac{-v_{i,j-1}+2v_{ij}-v_{i,j+1}}{h_y^2}+av_{ij}=f_{ij}\quad \text{with } 1\le j\le n-1$$

$$v_{i0}=v_{in}=v_{0j}=v_{mj}=0,\qquad 1\le i,j\le m-1$$

- Written in Matrix-Vector form

$$v_i=\begin{pmatrix}v_{i1} & \cdots & \cdots & v_{i,n-1}\end{pmatrix}^T$$

$$f_i=\begin{pmatrix}f_{i1} & \cdots & \cdots & f_{i,n-1}\end{pmatrix}^T$$

$$\begin{bmatrix} B & -\frac{1}{h^2}I & & \\ -\frac{1}{h^2}I & B & -\frac{1}{h^2}I & \\ & & \ddots & \\ & & -\frac{1}{h^2}I & B \end{bmatrix}\cdot\begin{bmatrix} v_1 \\ \vdots \\ v_{m-1}\end{bmatrix}=\begin{bmatrix} f_1 \\ \vdots \\ f_{m-1}\end{bmatrix}$$

B looks almost like matrix from 1D - Problem - dimension is $(n-1)\times(n-1)$

I is a $(n-1)\times(n-1)$ identity matrix

$\Rightarrow$ Dimension of matrix is $(m-1)\cdot(n-1)\times(m-1)\cdot(n-1)$

- Example: System for a=0, n=4, h=1

$$
\begin{bmatrix}
0 & -1 & 0 & -1 & & & & & \\
-1 & 4 & -1 & & -1 & & & & \\
0 & -1 & 0 & & & -1 & & & \\
-1 & & & 0 & -1 & 0 & -1 & & \\
& -1 & & -1 & 4 & -1 & & -1 & \\
& & -1 & 0 & -1 & 0 & & & -1 \\
& & & -1 & & & 0 & -1 & 0 \\
& & & & -1 & & -1 & 4 & -1 \\
& & & & & -1 & 0 & -1 & 0
\end{bmatrix}
\cdot
\begin{bmatrix} v_{1,1} \\ v_{1,2} \\ v_{1,3} \\ v_{2,1} \\ v_{2,2} \\ v_{2,3} \\ v_{3,1} \\ v_{3,2} \\ v_{3,3} \end{bmatrix}
=
\begin{bmatrix} f_{1,1} \\ f_{1,2} \\ f_{1,3} \\ f_{2,1} \\ f_{2,2} \\ f_{2,3} \\ f_{3,1} \\ f_{3,2} \\ f_{3,3} \end{bmatrix}
$$

$I$ $I$ $B$

- Again, written compactly: $A\mathbf{v} = \mathbf{f}$

# Overview

- Model Problems
- **Relaxation Methods**
- Error convergence
- Multiple grids
- Performance
- Theoretical Considerations

- To solve the PDE, $\mathbf{u} = A^{-1}\mathbf{f}$ is too complicated
- Based on an estimated solution $\mathbf{v}^{(0)} \rightarrow$ find better solution $\mathbf{v}^{(1)}$ in next step
- Reduces norm of the error $\mathbf{e} = \mathbf{u} - \mathbf{v}$
- Use residual $\mathbf{r} = \mathbf{f} - A\mathbf{v}$ as a measure
  Relationship error / residual: $A\mathbf{e} = \mathbf{r}$
  For exact solution $\mathbf{v} = \mathbf{u} \Rightarrow \mathbf{r} = \mathbf{0}$
- For the following, split matrix A = D – L – U
  D: diagonal of A; L/U: lower/upper triangular part of A

- General approximation: $\mathbf{v}^{(1)} = \mathbf{v}^{(0)} + B\mathbf{r}^{0}$
  - Try to find a B "close" to $A^{-1}$, as **u** – **v** = $A^{-1}$**r**
- **Jacobi scheme** / Simultaneous displacement
  - $j^{th}$ component of **v** is calculated using the two neighbours from previous step

$$v_j^{(1)} = \tfrac{1}{2}(v_{j-1}^{(0)} + v_{j+1}^{(0)} + h^2 f_j), \qquad 1 \le j \le n-1$$

$$\text{Jacobi iteration matrix}: \mathrm{R_J} = D^{-1}(L+U)$$

$$v^{(1)} = R_J v^{(0)} + D^{-1} f$$

  - Solves the PDE locally
    (compare original problem: $-u_{j-1} + 2u_j - u_{j+1} = h^2 f_j$)

- **Weighted** or damped **Jacobi method**
  - Weighting factor 0 < ω < 1

$$v_j^{(1)} = (1-\omega)v_j^{(0)} + \omega v_j^*, \qquad 1 \le j \le n-1$$
$$\text{with } v_j^* = \tfrac{1}{2}(v_{j-1}^{(0)} + v_{j+1}^{(0)} + h^2 f_j) \quad \text{(like above)}$$
$$\text{Weighted Jacobi iteration matrix: } \mathrm{R}_\omega = (1-\omega)I + \omega R_J$$
$$v^{(1)} = R_\omega v^{(0)} + \omega D^{-1} f$$

- Gauss-Seidel
  - Like Jacobi, but components updated immediately
  - Reduces storage requirements

$$v_j \leftarrow \tfrac{1}{2}(v_{j-1} + v_{j+1} + h^2 f_j), \qquad \leftarrow \text{meaning "overwrite"}$$
$$\text{formally}: v_j^{(1)} = \tfrac{1}{2}(v_{j-1}^{(1)} + v_{j+1}^{(0)} + h^2 f_j)$$

- Model Problems
- Relaxation Methods
- **Error convergence**
- Multiple grids
- Performance
- Theoretical Considerations

- Simplified problem: A**u** = **0**
  $\Rightarrow$ **v** should converge to **0**, and **e** = **v**
- In what way does weighted Jacobi decrease the error?
  $\Rightarrow$ Analyse eigenvectors of iteration matrix
- Eigenvectors $w_k$ of matrices A and $R_\omega$

$$w_{k,j} = \sin\left(\frac{jk\pi}{n}\right), \qquad \text{with } 1 \le k \le n-1,\ 0 \le j \le n$$

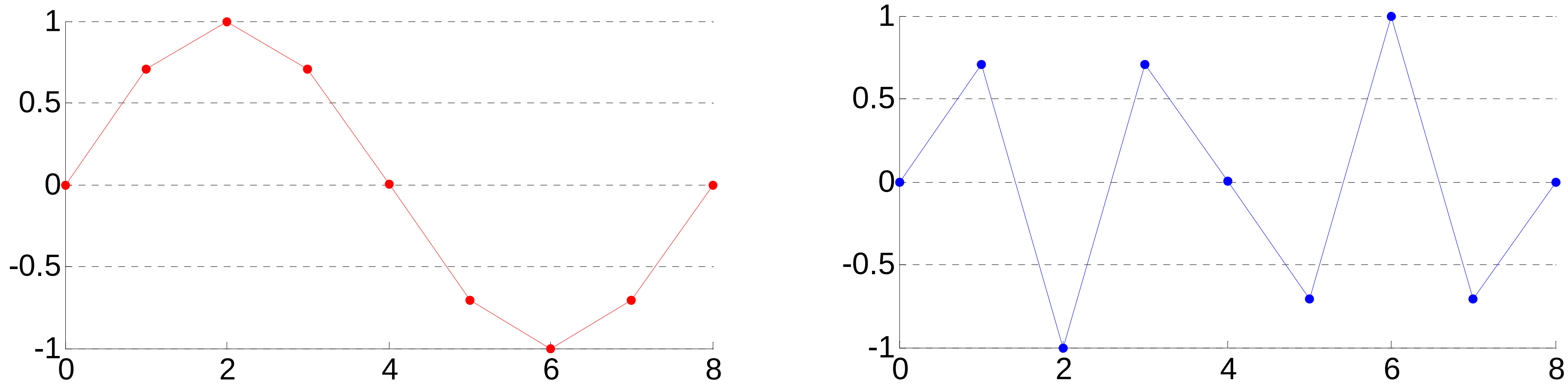
  - Vector $w_k$ is also the $k^{th}$ Fourier mode
- Eigen values $\lambda_k$ of matrix $R_\omega$ (generally: $R_\omega w_k = \lambda_k w_k$)

$$\lambda_k(R_\omega) = 1 - 2\omega \sin^2\left(\frac{k\pi}{2n}\right), \qquad \text{with } 1 \le k \le n-1$$

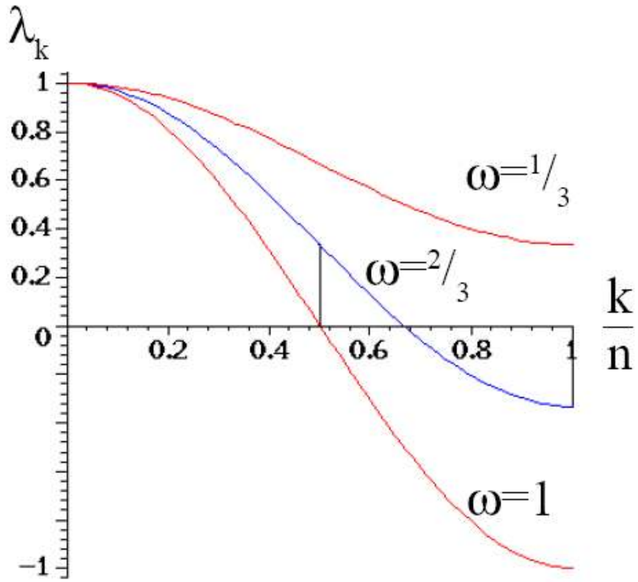
  - For $0 \le \omega < 1 \Rightarrow |\lambda_k| < 1$, iteration converges

# Error convergence

- Eigen values $\lambda_k(R_\omega) = 1 - 2\omega \sin^2\left(\frac{k\pi}{2n}\right), \quad \text{with } 1 \le k \le n-1$
- Smooth, low-frequency Fourier modes of **e**: $1 \le k \le \frac{1}{2}n$
  - $|\lambda_k|$ is close to 1 $\Rightarrow$ no satisfactory damping
- Oscillatory, high-frequency modes: $\frac{1}{2}n \le k \le n-1$
  - For the right $\omega$, $|\lambda_k|$ is close to 0 $\Rightarrow$ good damping
  - Optimal damping for $\omega = \frac{2}{3}$



- Damping diagram for the weighted Jacobi method



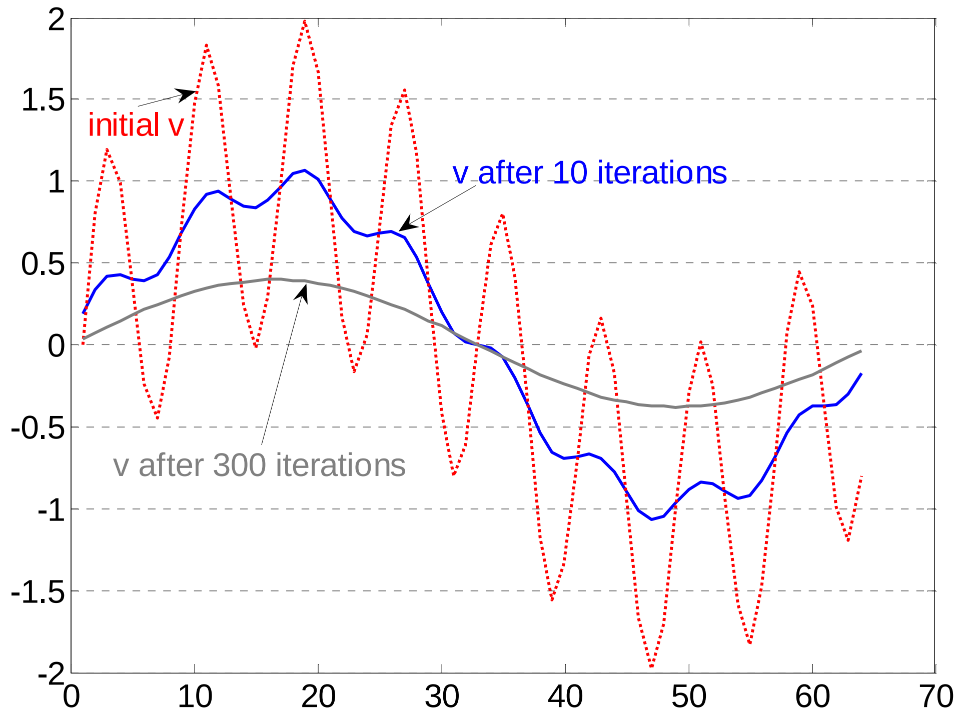

$$\lambda_k(R_\omega) = 1 - 2\omega \sin^2\left(\frac{k\pi}{2n}\right), \qquad \text{with } 1 \le k \le n-1$$

- Oscillatory modes of the error are removed quite well
- Smooth modes are hardly damped.

# Error convergence

- Example code in MATLAB
  - Grid n = 64
  - Initial error modes 2 and 16
  - Solves –u''(x) = 0

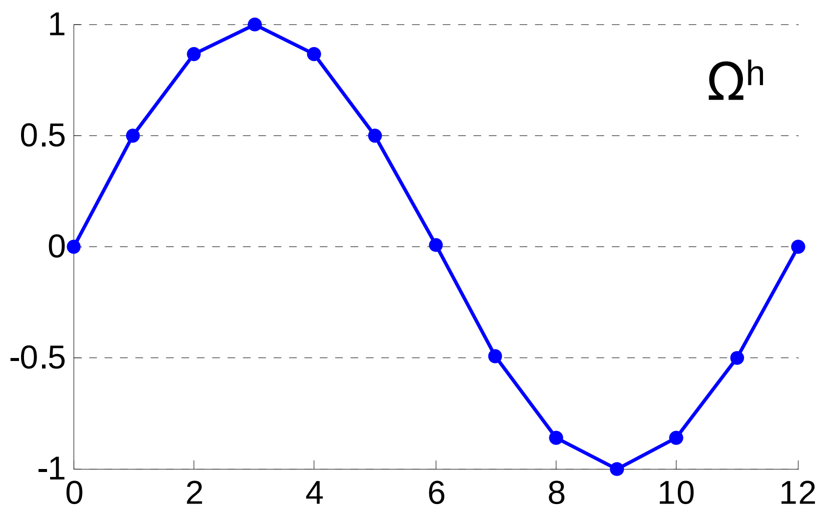
```
n = 64;
% components of A
D = 2 * diag(ones(n-1,1),0); U = d
% iteration matrices
w=2/3;  RJ = inv(D) * (L+U);  RW =
% init f=0, v with modes 2 and 16
f = zeros(n-1,1);
v = transpose(sin((1:n-1) * 2 * pi
plot(v); hold on
% do 10 iterations
for i = 1:10
    v = RW*v + 0; end
plot(v);
```

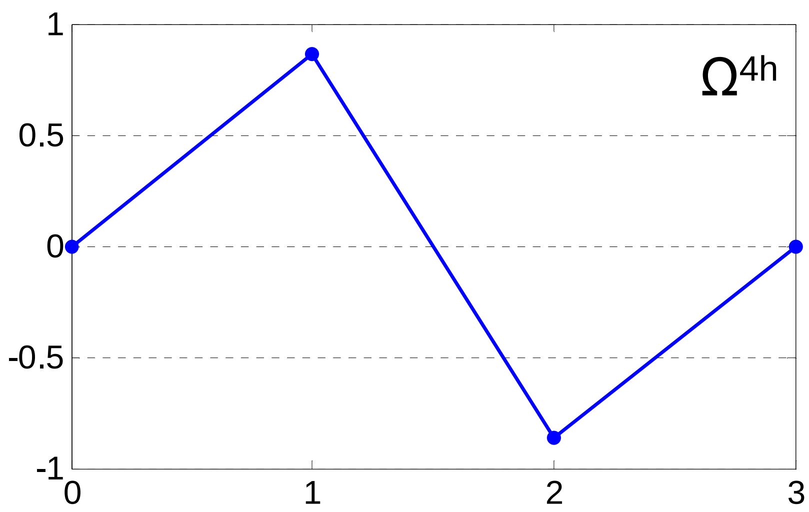


# Overview

- Model Problems
- Relaxation Methods
- Error convergence
- **Multiple grids**
- Performance
- Theoretical Considerations

- Fundamental idea of multigrid
  - Make smooth modes look oscillatory!
  - Smooth mode on $\Omega^h$ looks oscillatory on grid $\Omega^{nh}$
  - A “hierarchy of discretizations” is used to solve the problem of small damping for smooth modes



- Intergrid Transfer coarse → fine: **Interpolation**
  - $\Omega^{2h} \rightarrow \Omega^{h}$, “Upsampling”
  - Linear interpolation is effective

$$
\begin{aligned}
v_{2j}^{h} &= v_{j}^{2h} \\
v_{2j+1}^{h} &= \tfrac{1}{2}\left(v_{j}^{2h} + v_{j+1}^{2h}\right), \qquad 0 \leq j \leq \tfrac{n}{2} - 1
\end{aligned}
$$

$$
I_{2h}^{h}\mathbf{v}^{2h} = \frac{1}{2}\begin{bmatrix} 1 & & \\ 2 & & \\ 1 & 1 & \\ & 2 & \\ & 1 & 1 \\ & & 2 \\ & & 1 \end{bmatrix} \cdot \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}_{2h} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \end{bmatrix}_{h} = \mathbf{v}^{h}
$$

- Intergrid Transfer fine → coarse: **Restriction**
  - $\Omega^h \to \Omega^{2h}$, “Downsampling”
  - Simplest method: Injection
  - Better: Full weighting

$$v_j^{2h} = \tfrac{1}{4}(v_{2j-1}^h + 2v_{2j}^h + v_{2j+1}^h), \qquad 1 \le j \le \tfrac{n}{2} - 1$$

  - Restriction operator:

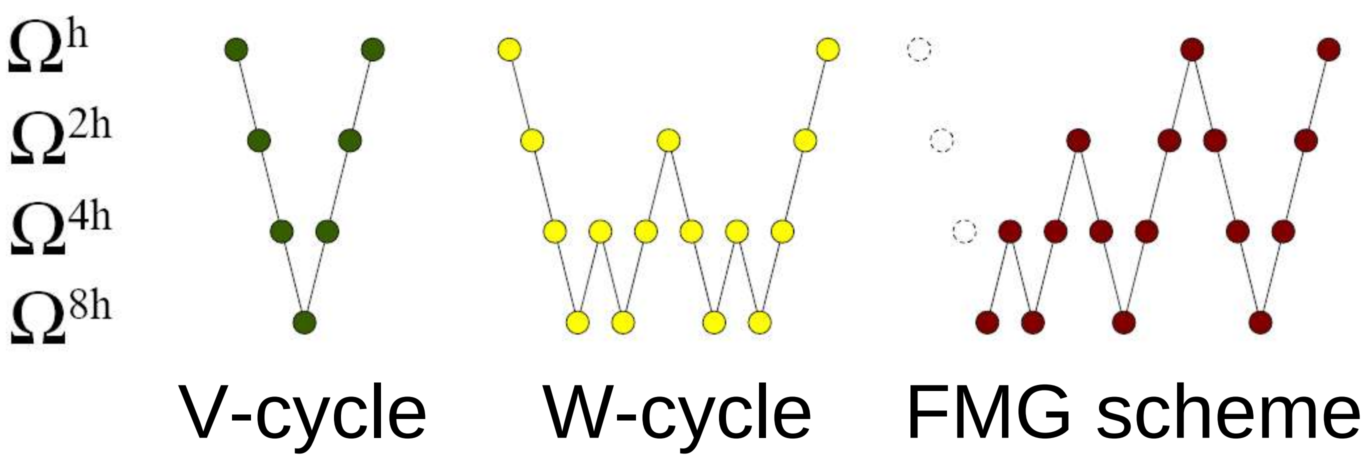
$$I_h^{2h} = \frac{1}{4}\begin{bmatrix} 1 & 2 & 1 & & & & \\ & & 1 & 2 & 1 & & \\ & & & & 1 & 2 & 1 \end{bmatrix}$$

$$\mathbf{v}^{2h} = I_h^{2h}\mathbf{v}^h$$

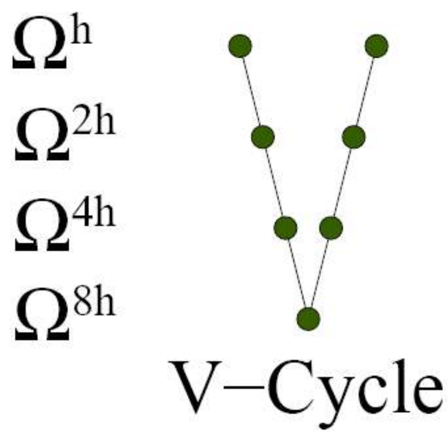
  - Transfer Operations $\Omega^h \leftrightarrow \Omega^{2h}$ sufficient

- Aliasing: Oscillatory modes on $\Omega^h$ will be represented as smooth modes on $\Omega^{2h}$
- A basic two-grid correction scheme
  - On grid $\Omega^h$, relax $\upsilon_1$ times on $A^h\mathbf{v}^h = \mathbf{0}$ with initial guess $\mathbf{v}^{(0)h}$
    - Restrict fine-grid residual $\mathbf{r}^h$ to the coarse grid
    - On grid $\Omega^{2h}$, relax $\upsilon_2$ times on $A^{2h}\mathbf{e}^{2h} = \mathbf{r}^{2h}$ with initial guess $\mathbf{e}^{(0)h} = \mathbf{0}$
    - Interpolate the coarse-grid error
  - Correct the fine-grid approximation: $\mathbf{v}^h \leftarrow \mathbf{v}^h + \mathbf{e}^h$
  - On grid $\Omega^h$, relax $\upsilon_1$ times on $A^h\mathbf{v}^h = \mathbf{0}$ with initial guess $\mathbf{v}^h$

- Multigrid strategies
  - Nested iteration: Use coarse grids to generate improved initial guesses
  - Coarse grid correction: Approximate the error by relaxing on the residual equation on a course grid

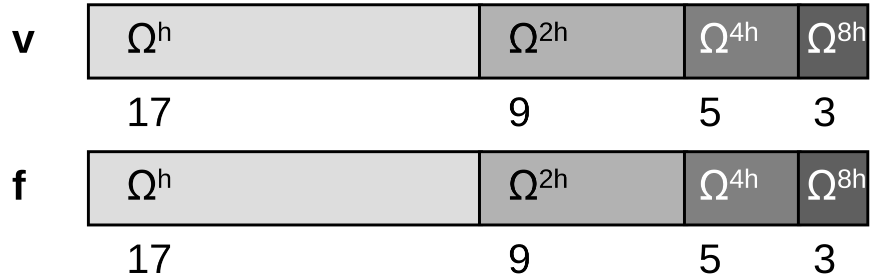


- The V-Cycle Scheme (Coarse Grid Correction)
  - V-Cycle($\mathbf{v}^h$, $\mathbf{f}^h$)
    - Relax $\upsilon_1$ times on $A^h\mathbf{v}^h = \mathbf{0}$ with initial guess $\mathbf{v}^h$
    - If (current grid = coarsest grid) goto last point
      - Else: $\mathbf{f}^{2h}$ = Restrict($\mathbf{f}^h - A^h\mathbf{v}^h$)
      - $\mathbf{v}^{2h} = \mathbf{0}$
      - Call $\mathbf{v}^{2h}$ = V-Cycle($\mathbf{v}^{2h}$, $\mathbf{f}^{2h}$)
    - Correct $\mathbf{v}^h$ += Interpolate($\mathbf{v}^{2h}$)
    - Relax $\upsilon_2$ times on $A^h\mathbf{v}^h = \mathbf{0}$ with initial guess $\mathbf{v}^h$
  - Recursive algorithm



- Model Problems
- Relaxation Methods
- Error convergence
- Multiple grids
- **Performance**
- Theoretical Considerations

- Storage requirements
  - Vectors **v** and **f** for n = 16 with boundary values




$$StorageSpace < \frac{2n^d}{1-2^{-d}}, \qquad \text{with d : Dimension of problem}$$

  - For d = 1, memory requirement is less then twice that of the fine-grid problem alone

- Computational costs
  - 1 work unit (*WU*): one relaxation sweep on $\Omega^h$
  - $O$(WU) = $O$(N), with N: Total number of grid points
  - Intergrid transfer is neglected
  - One relaxation sweep per level ($\upsilon_i = 1$)

$$Cost_{\text{V-Cycle}} < \frac{2}{1-2^{-d}} WU$$

$$Cost_{\text{FMG Cycle}} < \frac{2}{(1-2^{-d})^2} WU$$

  - 1D problem: Single V-Cycle costs ~4WU, Complete FMG cycle ~8WU

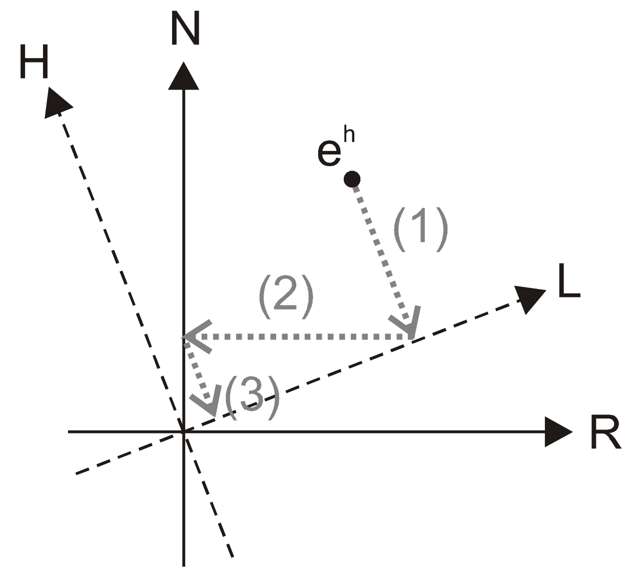
- Diagnostic Tools
  - Help to debug your implementation
  - *Methodical Plan* for testing modules
  - *Starting Simply* with small, simple problems
  - *Exposing Trouble* – difficulties might be hidden
  - *Fixed Point Property* – relaxation may not change exact solution
  - *Homogenous Problem*: norm of error and residual should decrease to zero

- Model Problems
- Relaxation Methods
- Error convergence
- Multiple grids
- Performance
- **Theoretical Considerations**

# The Multigrid Method

- The Impact of *Intergrid Transfer* and the *iterative method* may be expressed and proven in a formal way
- Two-grid correction TG consists of matrices for Interpolation, Restriction and Relaxation
- *Spectral* picture of multigrid
  - Relaxation damps oscillator modes
  - Interpolation & Restriction damp smooth modes
- *Algebraic* picture of multigrid
  - Decompose Space of the error: $\Omega^h = R \oplus N$
  - $R = Range(I_{2h}^{h}) = Ker(TG) \quad N = N(I_h^{2h} A^h) = Ker(I_h^{2h} A^h)$
  - L similar to R, H similar to N

- Operations of multigrid, visualized
  - Plane represents $\Omega^h$
  - Error $\mathbf{e}^h$ is successively projected on one of the axes
    - Relaxations on the fine grid (1)
    - Two-grid correction (2)
    - Again, relaxation on the fine grid (3)



# End of Presentation

- Thanks for your attention
- Any questions?