NUMERICAL LINEAR ALGEBRA

Prof. Paola Antonietti

MOX - Dipartimento di Matematica

Politecnico di Milano

https://antonietti.faculty.polimi.it

TA: Dr. Michele Botti

MODELLISTICA E CALCOLO SCIENTIFICO
MOX
MODELING AND SCIENTIFIC COMPUTING

POLITECNICO MILANO 1863 | DEPARTMENT OF MATHEMATICS

P5: Multigrid methods

Reference text: William L. Briggs, Steve F McCormick, Van Emden Henson, A Multigrid Tutorial, 2nd edition, 2000.

Note: most of the pictures are taken from the Reference text.

IDEA of MG Methods

- A multigrid (MG) method is an iterative algorithm of the form $\mathbf{x}^{(k+1)} = MG(\mathbf{x}^{(k)})$, $k \geq 0$, for solving the (typically) sparse linear systems of equations stemming from the numerical discretisation of differential equations
- MG methods are based on a hierarchy of levels (associated with a hierarchy of discretisations).
- MG-cycle reduces all error components by a fixed amount (bounded well below one) independent of the dimension n of the system.
- The main idea of multigrid is to accelerate the convergence of a basic iterative method by a global correction of the fine grid solution approximation accomplished by solving a coarse problem.
- The coarse-level problem should be "similar" to the fine grid problem. The cost of (direct) solution of the coarse problem should be negligible compared to the cost of one relaxation sweep on the fine grid.

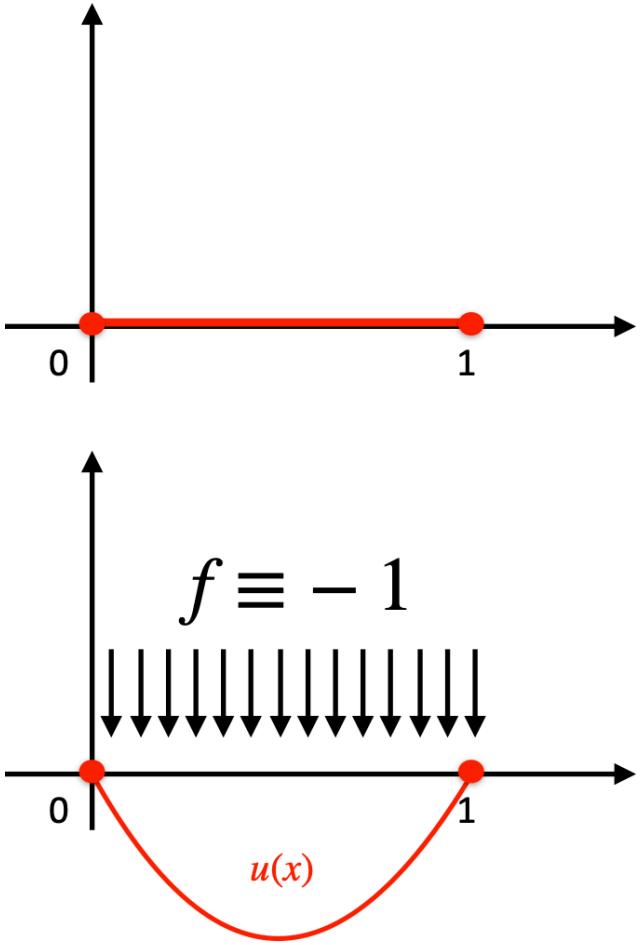
A simple 1d example

One-dimensional boundary value problem describing the displacement $u(x)$ of a uniform rod subject to an external load $f(x)$ and fixed at the extrema.

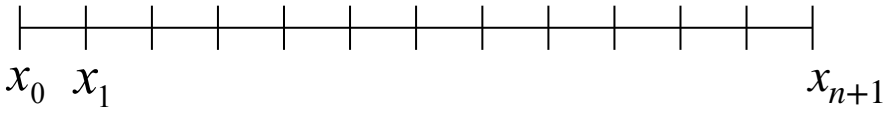
$$-u''(x) = f(x) \qquad x \in (0,1)$$

$$u(0) = 0, \qquad u(1) = 0$$

Grid-size: $h = \dfrac{1}{n+1}, n \geq 1$



Grid: $x_j = jn,\ j = 0,\ldots,n+1$



Approximation with the finite element method

- Let u_j be the approximate solution of $u(x_j)$ $j = 1,\ldots,n$ on a uniform grid of mesh size h (u_0 and u_{n+1} are known).

Then,

$$-u_{j-1} + 2u_j - u_{j+1} = h\,f(x_j) \qquad j = 1,\ldots,n$$

$$u_0 = 0 \qquad u_{n+1} = 0$$

Matrix form

Defining $\mathbf{x}_h = (u_1, u_2, \ldots, u_n)^T$ and $\mathbf{b}_h = (b_1, b_2, \ldots, b_n)^T$ we the (sparse) linear system of equations

$$A_h \mathbf{x}_h = \mathbf{b}_h$$

where

$$\begin{pmatrix} 2 & -1 & \ldots & \ldots & \ldots 0 \\ -1 & 2 & -1 & \ldots & 0 \\ 0 & \ldots & \ldots & \ldots & -1 \\ 0 & \ldots & \ldots & -1 & 2 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \cdots \\ u_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \cdots \\ b_n \end{pmatrix}$$

Remark: A is SPD (prove it by exercise).

Some recap

- Notation for model problem is $A_h \mathbf{x}_h = \mathbf{b}_h$
- Given the approximate solution $\mathbf{x}_h^{(k)}$, the error is $\mathbf{e}_h^{(k)} = \mathbf{x}_h - \mathbf{x}_h^{(k)}$
- For $\mathbf{y}_h \in \mathbb{R}^n$, the residual is given by $\mathbf{r}_h = \mathbf{b}_h - A_h \mathbf{y}_h$
- Given the residual $\mathbf{r}_h$, the error equation is $A_h \mathbf{e}_h = \mathbf{r}_h$

Some recap

- From linear iterative methods of the form

$$\mathbf{x}_h^{(k+1)} = B\mathbf{x}_h^{(k)} + \mathbf{g}_h$$

we know that $\lim_{k\to\infty} B^k = 0 \iff \rho(B) < 1$ where

$$\rho(B) = \max\left\{ |\lambda_1|, |\lambda_2|, \ldots, |\lambda_n| \right\}$$

is the spectral radius or asymptotic convergence factor, and $\lambda_i, i = 1,\ldots,n$, are the eigenvalues of the iteration matrix B.

- For any initial guess $\mathbf{x}_h^{(0)}$,

$$\lim_{k\to\infty} \mathbf{e}_h^{(k)} = \mathbf{0} \iff \rho(B) < 1$$

Example - weighted Jacobi iteration

- We have established the importance of the spectral radius of the iteration matrix in analysing the convergence properties of linear iterative methods.
- For example, we consider the weighted Jacobi iteration applied to the one-dimensional model problem. Recalling that $B_\omega = (1-\omega)I + \omega B_J$, where B_J is the iteration matrix of the Jacobi method, we have

$$B_\omega = I - \frac{\omega}{2}\begin{pmatrix} 2 & -1 & \dots & \dots & \dots 0 \\ -1 & 2 & -1 & \dots & 0 \\ 0 & \dots & \dots & \dots & 0 \\ 0 & \dots & \dots & -1 & 2 \end{pmatrix}$$

- Therefore, the eigenvalues of B_ω satisfy $\lambda(B_\omega) = 1 - \frac{\omega}{2}\lambda(A_h)$

On the eigenpairs of the matrix A_h

The eigenvalues of A_h are

$$\lambda_j(A_h) = 4\sin^2\left(\frac{j\pi}{2(n+1)}\right) \qquad j = 1,\ldots,n$$

and therefore

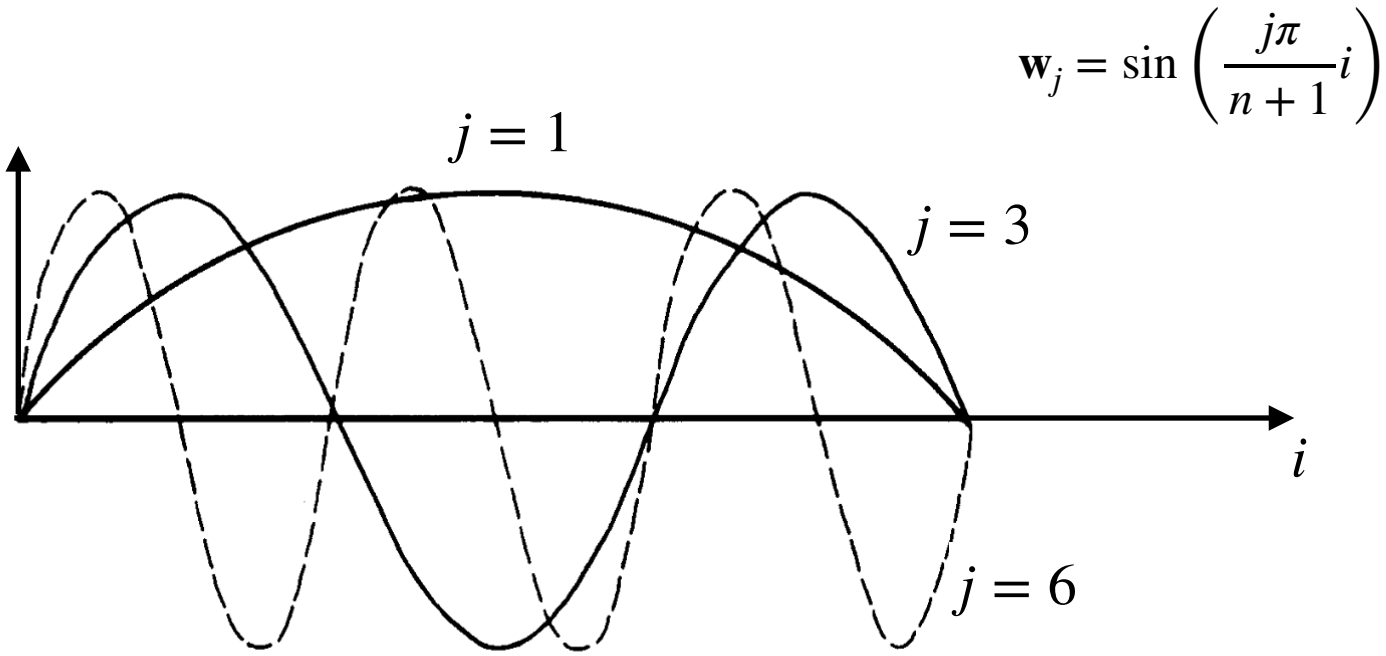
$$\lambda_j(B_\omega) = 1 - \frac{\omega}{2}4\sin^2\left(\frac{j\pi}{2(n+1)}\right) \qquad j = 1,\ldots,n$$

The corresponding eigenvector $\mathbf{w}_j(A_h)$ of A_h (and of B_ω, prove it by exercise) has components

$$(\mathbf{w}_j(A))_i = \sin\left(\frac{ij\pi}{n+1}\right)$$

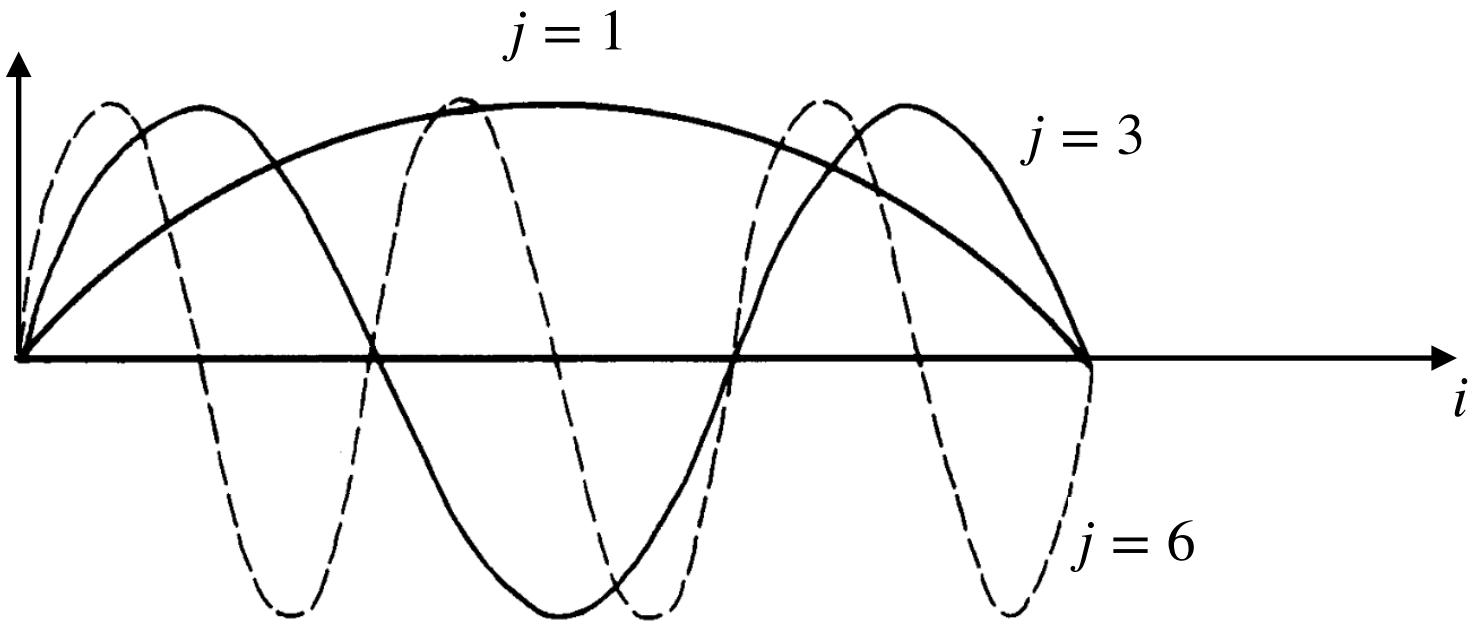
On the eigenpairs of the matrix A_h

Note that if $0 < \omega \leq 1$, then $|\lambda_j(B_\omega)| < 1 \quad \forall j$ and the weighted Jacobi iteration converges

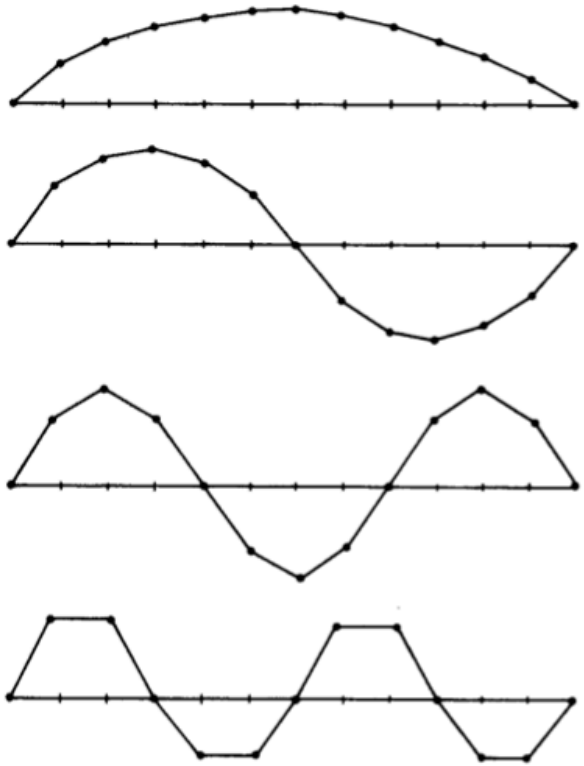


Fourier modes

$$\mathbf{w}_j(x) = \sin\left(\frac{j\pi}{n+1}x\right) \quad x \in \mathbb{R}$$



Fourier modes of A_h on a grid with $h = 1/12$



$$j = 1, \quad (\mathbf{w}_1)_i = \sin\left(\frac{i\pi}{12}\right)$$

$$j = 2, \quad (\mathbf{w}_2)_i = \sin\left(\frac{2i\pi}{12}\right) = \sin\left(\frac{i\pi}{6}\right)$$

$$j = 3, \quad (\mathbf{w}_3)_i = \sin\left(\frac{i\pi}{4}\right)$$

$$j = 4, \quad (\mathbf{w}_4)_i = \sin\left(\frac{i\pi}{3}\right)$$

Fourier modes of A_h on a grid with $h = 1/12$. Modes with wavenumbers $j = 1,2,3,4$ are shown.

Effect of k iterations of weighted Jacobi

- Let $\mathbf{x}_h^{(0)}$ be an initial guess, we have $\mathbf{e}_h^{(0)} = \mathbf{x}_h - \mathbf{x}_h^{(0)} = \mathbf{x}_h^{(0)}$, we represent $\mathbf{e}_h^{(0)}$ using the eigenvectors of A in the form

$$\mathbf{e}_h^{(0)} = \sum_{j=1}^{n} c_j \mathbf{w}_j, \text{ for suitable coefficients } c_j, j = 1, \ldots, j$$

representing the "amount" of each mode in the error.

-It can be shown that after k steps of the iteration, the error is given by

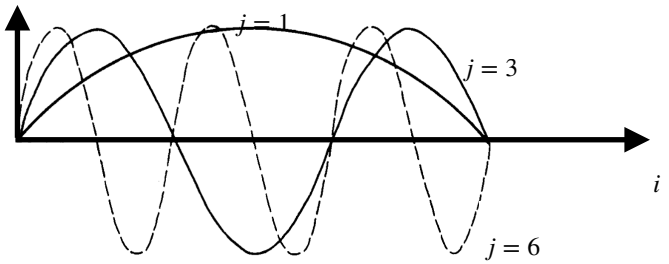
$$\mathbf{e}_h^{(k)} = B_\omega^k \mathbf{e}_h^{(0)} = \sum_{j=1}^{n} c_j B_\omega^k \mathbf{w}_j = \sum_{j=1}^{n} c_j \lambda_j^k(B_\omega) \mathbf{w}_j$$

That is, after k iterations, the jth mode of the initial error has been reduced by a factor of $\lambda_j^k(B_\omega)$

To summarise

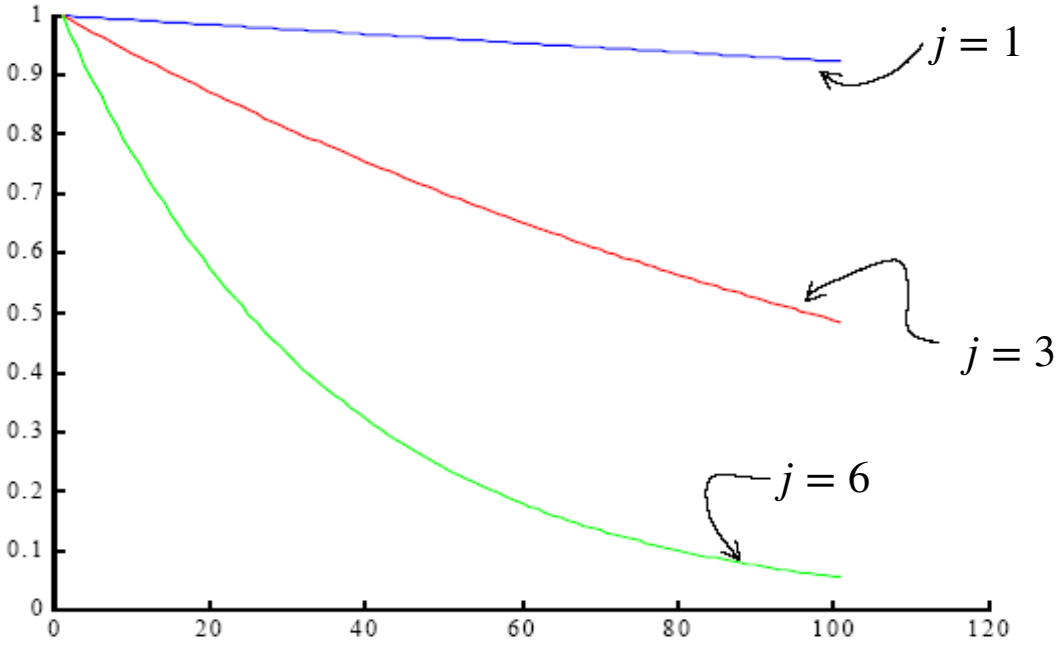
- We work with the homogenous equation $A_h \mathbf{x}_h = \mathbf{0}$
- We can obtain some valuable insight by applying various iterations to the above system with an initial guess consisting of the vector $\mathbf{w}_j$ (or Fourier modes) with wavenumber j
- Small values of j correspond to long, smooth waves, while large values of j correspond to highly oscillatory waves.

$$\mathbf{w}_j = \sin\left(\frac{j\pi}{n+1} i\right)$$



A numerical example

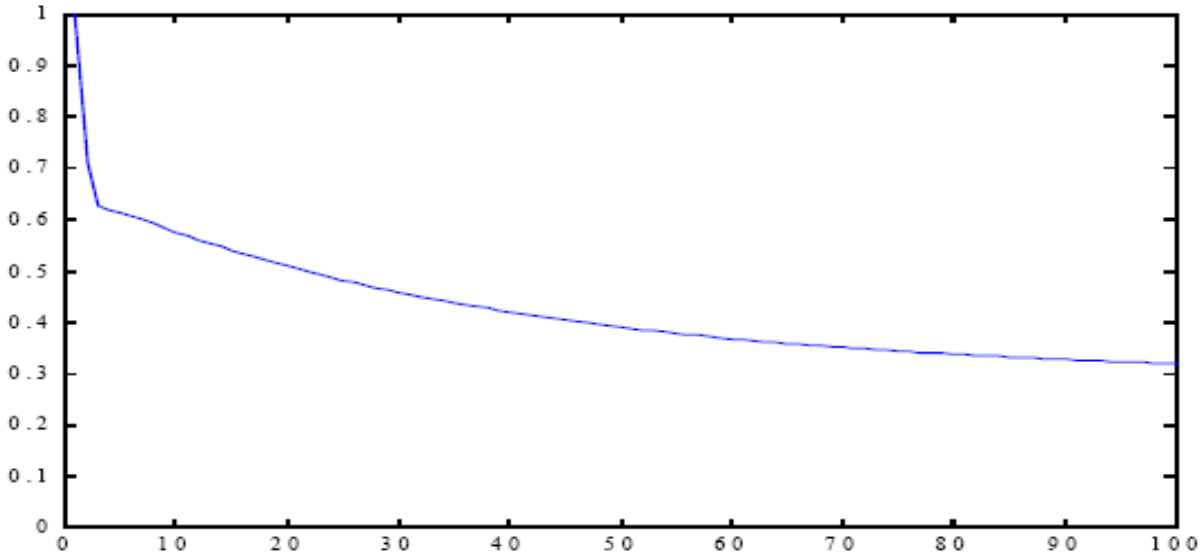
The error in the weighted Jacobi iteration ($\omega = 2/3$) for $k = 100$ iterazioni and with initial condition $\mathbf{x}_\mathbf{h}^{(\mathbf{0})} = \mathbf{w}_\mathbf{j}$, $j = 1,3,6$



Smooth modes are damped slowly, while the oscillatory modes show rapid decay.

A numerical example (2)

The error in the weighted Jacobi iteration ($\omega = 2/3$) for $k = 100$ iterations and with the initial condition $\mathbf{x_h^{(0)}} = (\mathbf{w_1} + \mathbf{w_6} + \mathbf{w_{32}})/\mathbf{3}$.



Fast converge as long as the error has high-frequency components. Slow elimination of the low-frequency components of the error (degradation of the performance)

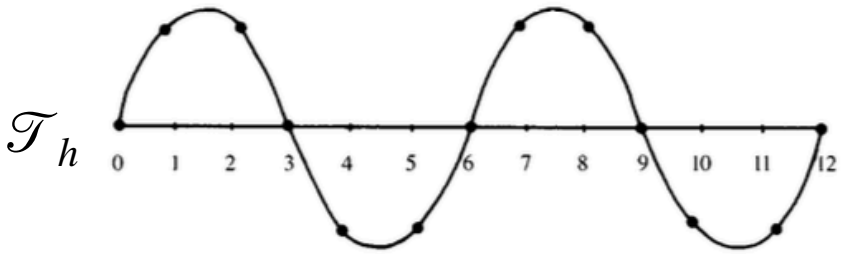
To wrap up

- The eigenvalue associated with the smoothest mode will always be close to 1 (especially for h small)
- The smooth components of the error cannot be reduced even choosing ω in a "smart" way

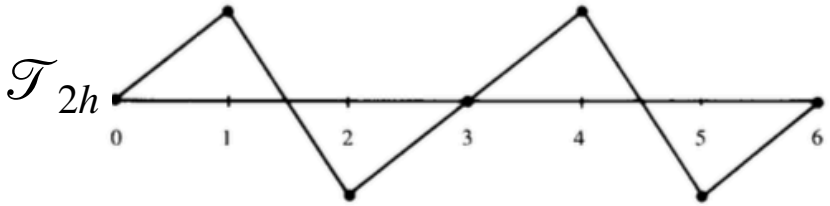
The key observation

- Basic relaxation schemes suffer in the presence of smooth components of the error.
- Assume that a particular relaxation scheme has been applied until only smooth error components remain.
- We now ask what these smooth components look like on a coarser grid

The key observation



$\mathbf{w}_j, j = 4$ on a grid with $1/h = 12$



$\mathbf{w}_j, j = 4$ on a grid with $1/h = 6$

On this coarse grid, the original wave still has a wavenumber of $j = 4$. We see that a smooth wave on $\mathcal{T}_h$ looks more oscillatory on $\mathcal{T}_{2h}$

Elements of MG methods

1. Coarse Grids
2. Correction
3. Interpolation Operator
4. Restriction Operator
5. Two-Grid Scheme
6. V-Cycle Scheme
7. Full Multigrid (FMG) V-Cycle Scheme

Coarse Grids

- The important consequence is that in passing from the fine grid to the coarse grid, a mode becomes more oscillatory. This is true provided that $1 \leq j < n/2$.
- It should be verified that the $j = n/2$ mode on $\mathcal{T}_h$ becomes the zero vector on $\mathcal{T}_{2h}$.
- The important point is that smooth modes on a fine grid look less smooth on a coarse grid.
- This suggests that when relaxation begins to stall, signaling the predominance of smooth error modes, it is a good idea to move to a coarser grid; there, the smooth error modes appear more oscillatory and relaxation will be more effective.

Coarse Grids

The question is: how do we move to a coarser grid and relax on the more oscillatory error modes?

Important observation

- Relaxation on the original equation $A_h \mathbf{x}_h = \mathbf{b}_h$ with an arbitrary initial guess $\mathbf{x}^{(0)}$ is equivalent to relaxing on the residual equation $A_h \mathbf{e}_h = \mathbf{r}_h^{(k)}$ with the specific initial guess $\mathbf{e}_h^{(0)} = \mathbf{0}$. Recall that $\mathbf{r}_h^{(k)} = \mathbf{b}_h - A_h \mathbf{x}_h^{(k)}$, $\mathbf{e}_h^{(k)} = \mathbf{x}_h - \mathbf{x}_h^{(k)}$

- This connection between the original and the residual equations motivates the use of the residual equation

Correction Scheme $\mathbf{x}_h^{(k)} \longrightarrow \mathbf{x}_h^{(k+1)}$

Idea: use the residual equation to relax on the error

1. Relax ν_1 times on $A_h\mathbf{x}_h = \mathbf{b}_h$ to obtain an approximation $\mathbf{x}_h^{(k+\nu_1)}$.
2. Compute the residual $\mathbf{r}_h^{(k+\nu_1)} = \mathbf{b}_h - A_h\mathbf{x}_h^{(k+\nu_1)}$.
3. Move the residual $\mathbf{r}_h^{(k+\nu_1)}$ from $\mathcal{T}_h$ to $\mathcal{T}_{2h}$ to obtain $\mathbf{r}_{2h}^{(k+\nu_1)}$
4. Solve the residual equations $A_{2h}\mathbf{e}_{2h} = \mathbf{r}_{2h}^{(k+\nu_1)}$ on $\mathcal{T}_{2h}$
5. Move the error $\mathbf{e}_{2h}$ from $\mathcal{T}_{2h}$ to $\mathcal{T}_h$ to obtain $\mathbf{e}_h$
6. Correct the approximation obtained on $\mathcal{T}_h$ with the error estimate obtained on $\mathcal{T}_{2h}$, i.e., $\mathbf{x}_h^{(k+1)} = \mathbf{x}_h^{(k+\nu_1)} + \mathbf{e}_h$

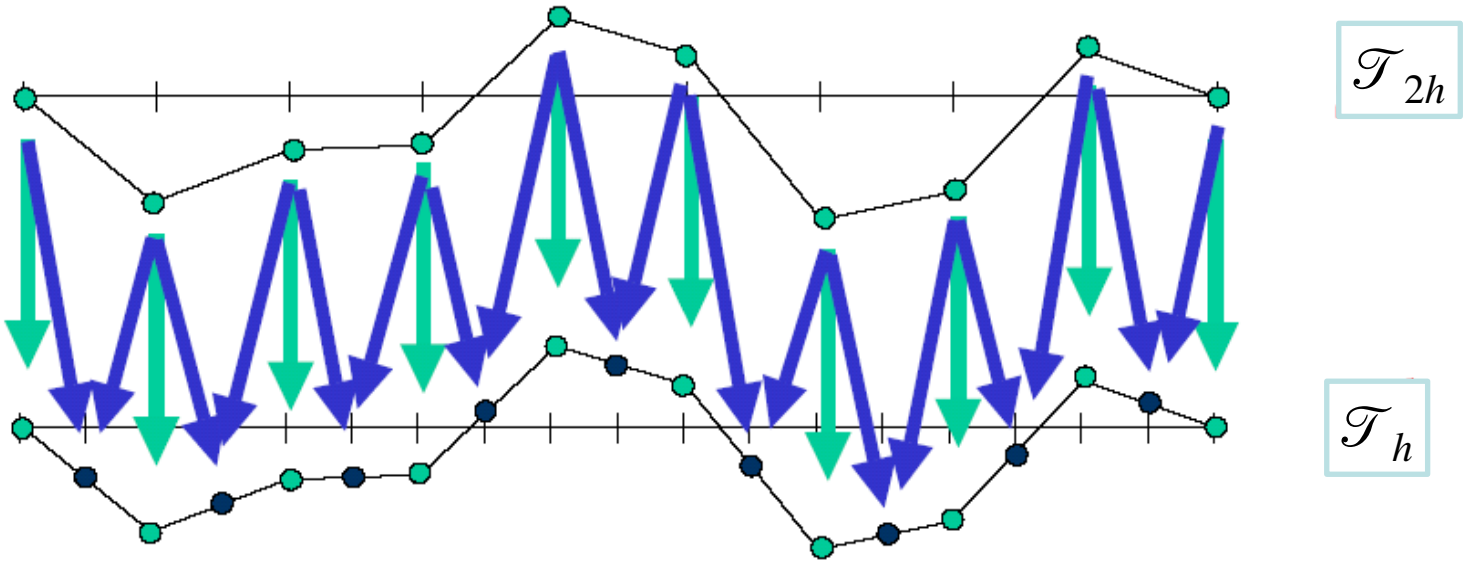
Interpolation Operator (coarse-to-fine)

Mapping $I_{2h}^{h} : \mathcal{T}_{2h} \longrightarrow \mathcal{T}_{h}$, i.e.

$$I_{2h}^{h}\mathbf{v}_{2h} = \mathbf{v}_{h}$$

where

$$\mathbf{v}_{h,i} = \begin{cases} \mathbf{v}_{2h,i} & \text{if the node } i \text{ is common node of both } \mathcal{T}_h \text{ and } \mathcal{T}_{2h} \\ \dfrac{\mathbf{v}^{+}_{2h,i} + \mathbf{v}^{-}_{2h,i}}{2} & \text{if the node } i \text{ in } \mathcal{T}_h \text{is not a node in } \mathcal{T}_{2h} \end{cases}$$



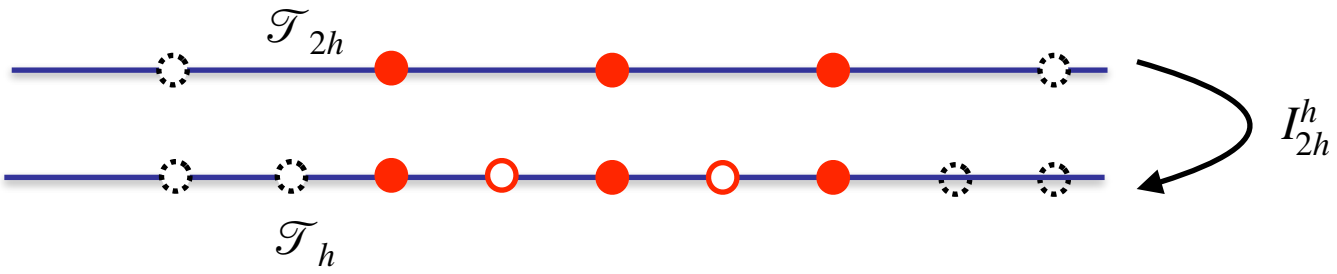
Interpolation Operator - remarks

- If the exact error on $\mathcal{T}_h$ is smooth, an interpolant of the coarse-grid error $\mathbf{e}_{2h}$ should give a good representation of the exact error
- If the exact error on $\mathcal{T}_h$ is oscillatory, an interpolant of the coarse-grid error $\mathbf{e}_{2h}$ may give a poor representation of the exact error.

Interpolation Operator (1d case)

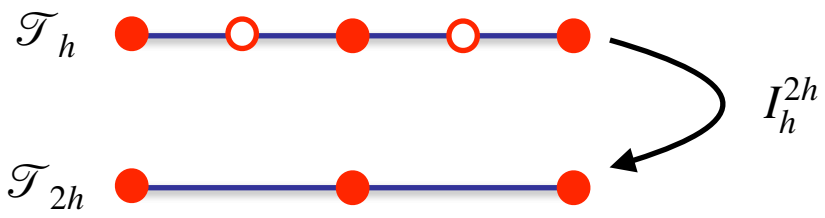
$I_{2h}^{h} : \mathcal{T}_{2h} \longrightarrow \mathcal{T}_{h}$ is a linear operator from $\mathbb{R}^n \longrightarrow \mathbb{R}^m$.

Example



$$I_{2h}^{h}\mathbf{v}_{2h} = \begin{pmatrix} 1 & & \\ 1/2 & 1/2 & \\ & 1 & \\ & 1/2 & 1/2 \\ & & 1 \end{pmatrix} \begin{pmatrix} v_1^{2h} \\ v_2^{2h} \\ v_3^{2h} \end{pmatrix} = \begin{pmatrix} v_1^{2h} \\ (v_1^{2h} + v_2^{2h})/2 \\ v_2^{2h} \\ (v_2^{2h} + v_3^{2h})/2 \\ v_3^{2h} \end{pmatrix} = \begin{pmatrix} v_1^{h} \\ v_2^{h} \\ v_3^{h} \\ v_4^{h} \\ v_5^{h} \end{pmatrix}$$

Restriction Operator (1d case)



- The second class of intergrid transfer operations involves moving vectors from a fine grid to a coarse grid.
- They are generally called restriction operators and are denoted by I_h^{2h}, $I_h^{2h} : \mathcal{T}_h \longrightarrow \mathcal{T}_{2h}$.
- The most obvious restriction operator is injection, defined by $I_h^{2h}\mathbf{w}_h = \mathbf{w}_{2h}$

Building A_{2h} and the Galerkin condition

- Galerkin condition $A_{2h} = I_h^{2h} A_h I_{2h}^h$
- $I_h^{2h} = c(I_{2h}^h)^T \qquad c \in \mathbb{R}$

One iteration of the Two-Grid Correction Scheme: $\mathbf{x}_h^{(k+1)} = MG(\mathbf{x}_h^{(k)}, \mathbf{b}_h, \nu_1, \nu_2)$

1. Do ν_1 iterations (of your favourite method) on $A_h\mathbf{x}_h = \mathbf{b}_h$ with initial guess $\mathbf{x}_h^{(k)}$. Denote by $\mathbf{y}_h^{(\nu_1)}$ the obtained solution.
2. Compute the fine grid residual $\mathbf{r}_h^{(\nu_1)} = \mathbf{b}_h - A_h\mathbf{y}_h^{(\nu_1)}$
3. Restrict the fine residual $\mathbf{r}_h^{(\nu_1)}$ to the coarse grid, i.e. $\mathbf{r}_{2h}^{(\nu_1)} = I_h^{2h}\mathbf{r}_h^{(\nu_1)}$
4. Solve $A_{2h}\mathbf{e}_{2h} = \mathbf{r}_{2h}^{(\nu_1)}$ (e.g., with a direct solver)
5. Interpolate $\mathbf{e}_{2h}$ from the coarse to the fine grid, i.e. $\mathbf{e}_h = I_{2h}^h\mathbf{e}_{2h}$
6. Update $\mathbf{y}_h^{(\nu_1+1)} = \mathbf{y}_h^{(\nu_1)} + \mathbf{e}_h$
7. Do ν_2 iterations (your favourite smoother) on $A_h\mathbf{x}_h = \mathbf{b}_h$ with initial guess $\mathbf{y}_h^{(\nu_1+1)}$. Denote by $\mathbf{y}_h^{(\nu_1+1+\nu_2)}$ the obtained solution.
8. Return $\mathbf{x}_h^{(k+1)} \longleftarrow \mathbf{y}_h^{(\nu_1+1+\nu_2)}$

In practice, $\nu_1 = \nu_2$ is often 1, 2, or 3

One iteration of the Two-Grid Correction Scheme: $\mathbf{x}_h^{(k+1)} = MG(\mathbf{x}_h^{(k)}, \mathbf{b}_h, \nu_1, \nu_2)$

1. $\mathbf{y}_{\mathbf{h}}^{(\nu_1)} \leftarrow$ Relax ν_1 times on $A_h\mathbf{x}_h = \mathbf{b}_h$ with i.g. $\mathbf{x}_h^{(k)}$.
2. $\mathbf{r}_h^{(\nu_1)} \leftarrow \mathbf{b}_h - A_h\mathbf{y}_h^{(\nu_1)}$

$\mathbf{r}_{2h}^{(\nu_1)} = I_h^{2h}\mathbf{r}_h^{(\nu_1)}$

Solve $A_{2h}\mathbf{e}_{2h} = \mathbf{r}_{2h}^{(\nu_1)}$

$\mathbf{e}_h = I_{2h}^h\mathbf{e}_{2h}$

1. $\mathbf{y}_h^{\nu_1+1+\nu_2} \leftarrow$ Relax ν_2 times on $A_h\mathbf{x}_h = \mathbf{b}_h$ with i.g. $\mathbf{y}_h^{(\nu_1+1)}$
2. $\mathbf{y}_h^{(\nu_1+1)} \leftarrow \mathbf{y}_h^{(\nu_1)} + \mathbf{e}_h$

Two-Grid Scheme

GIVEN $\mathbf{x}_h^{(0)}$

While (STOPPING CRITERIA)

$$\mathbf{x}_h^{(k+1)} = MG(\mathbf{x}_h^{(k)}, \mathbf{b}_h, \nu_1, \nu_2)$$

END

One iteration of V-Cycle - Recursive- $\mathbf{x}_h^{(k+1)} = MG_V(\mathbf{x}_h^{(k)}, \mathbf{b}_h, \nu_1, \nu_2, j)$

1. Do ν_1 iterations (your favourite smoother) on $A_h\mathbf{x}_h = \mathbf{b}_h$ with initial guess $\mathbf{x}_h^{(k)}$. Denote by $\mathbf{y}_h^{(\nu_1)}$ the obtained solution.
2. Compute the fine grid residual $\mathbf{r}_h^{(\nu_1)} = \mathbf{b}_h - A_h\mathbf{y}_h^{(\nu_1)}$
3. Restrict the fine residual $\mathbf{r}_h^{(\nu_1)}$ to the coarse grid, i.e. $\mathbf{r}_{2h}^{(\nu_1)} = I_h^{2h}\mathbf{r}_h^{(\nu_1)}$
4. IF j=coarsest level
   - Solve $A_{2h}\mathbf{e}_{2h} = \mathbf{r}_{2h}^{(\nu_1)}$ (e.g., with a direct solver)

   ELSE
   - $\mathbf{e}_{2h} = MG_V(\mathbf{0}, \mathbf{r}_{2h}^{(\nu_1)}, \nu_1, \nu_2, j-1)$

   END
5. Interpolate $\mathbf{e}_{2h}$ from the coarse to the fine grid, i.e. $\mathbf{e}_h = I_{2h}^h\mathbf{e}_{2h}$
6. Update $\mathbf{y}_h^{(\nu_1+1)} = \mathbf{y}_h^{(\nu_1)} + \mathbf{e}_h$
7. Do ν_2 iterations (your favourite smoother) on $A_h\mathbf{x}_h = \mathbf{b}_h$ with initial guess $\mathbf{y}_h^{(\nu_1+1)}$. Denote by $\mathbf{y}_h^{(\nu_1+1+\nu_2)}$ the obtained solution.
8. Return $\mathbf{x}_h^{(k+1)} \longleftarrow \mathbf{y}_h^{(\nu_1+1+\nu_2)}$

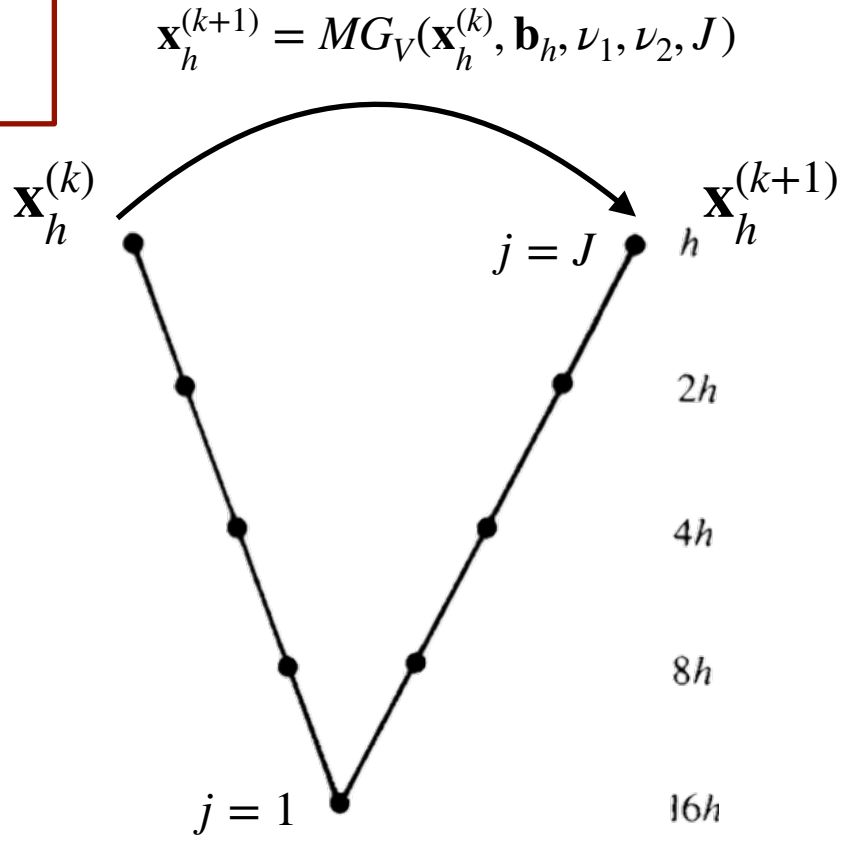
Vcycle - MG - Scheme

GIVEN $\mathbf{x}_h^{(0)}$

While (STOPPING CRITERIA)

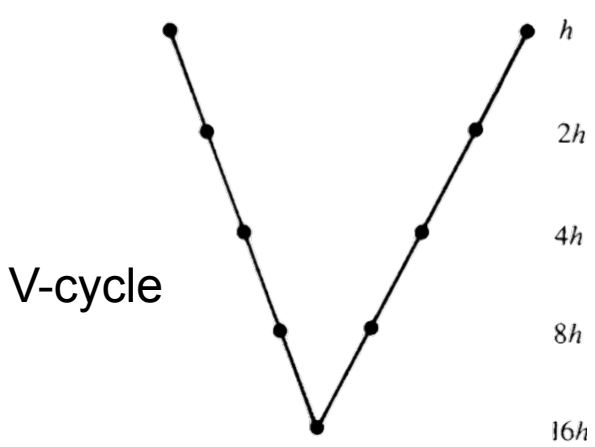
$$\mathbf{x}_h^{(k+1)} = MG_V(\mathbf{x}_h^{(k)}, \mathbf{b}_h, \nu_1, \nu_2, J)$$

END

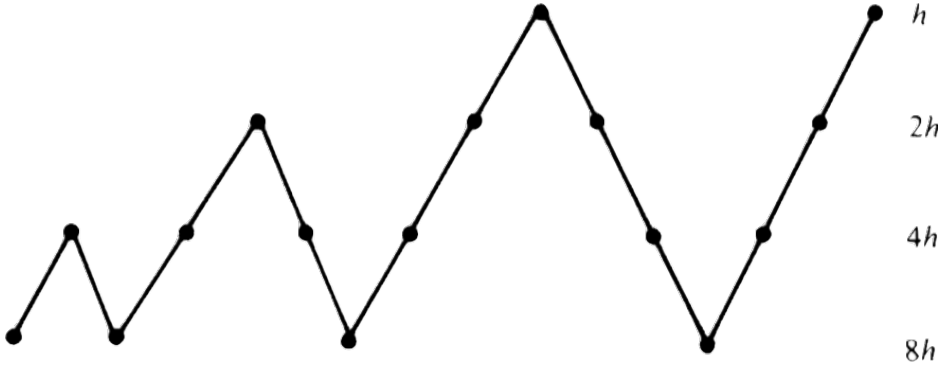


μ-cycle MG - generalizations

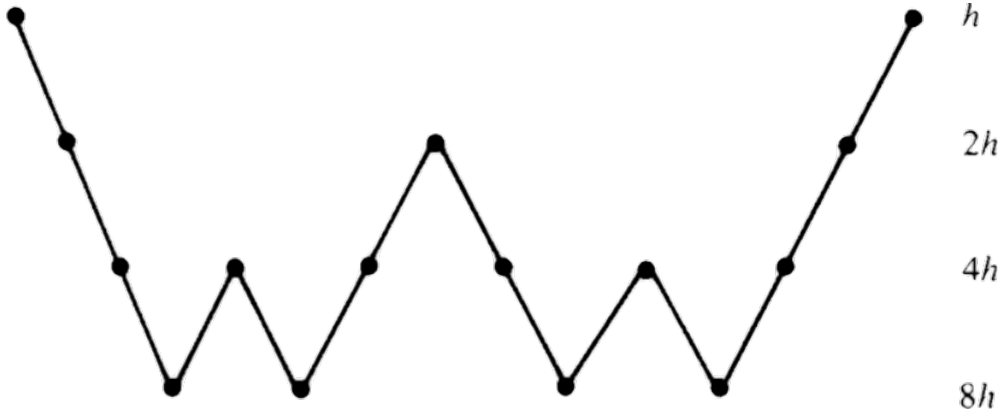
The V-cycle is just one of a family of multigrid cycling schemes.
The entire family is called the μ-cycle method.



V-cycle



F-cycle



W-cycle

Storage Costs

- $\mathbf{x}_h^{(k)}$ and $\mathbf{b}_h$ must be stored at each level j
- In $d-$dimensions, the coarse grid has $\approx 2^{-d}$ the number of points as the finer grid.
- Storage costs $\approx \dfrac{2n^d}{1-2^{-d}}$

For $d = 1$, the storage requirement is less than twice that of the fine-grid problem alone.

For $d \geq 2$, the requirement drops to less than 4/3 of the fine-grid problem alone. Thus, the storage costs of multigrid algorithms decrease relatively as the dimension of the problem increases

Computational Costs

- We measure the costs in terms of Work Unit (WU), which is the cost of performing one relaxation sweep on the finest grid. We neglect the cost of intergrid transfer operations, which typically amounts to 10–20% of the cost of the entire cycle
- Computational cost of V-cycle with $\nu_1 = \nu_2 = 1$

$$\approx \frac{2}{1 - 2^{-d}} WU$$

- A single V-cycle costs about 4 WUs for a one-dimensional (d = 1) problem, about 8 WUs for d=2, and 16 WUs for d=3.

Convergence Analysis (hints)

- The V-cycle scheme has a convergence < 1, independent of h (i.e, independent of n)
- The error should be reduced from $O(1)$ to $O(h^p) = O(n^{-p})$ (that is the approximation error, with $h = n^{-1}$)
- Costs $\approx O(n^d \log(n))$