Méthodes Itératives

Carola Kruse, Alena Kopanicakova et Ronan Guivarch

Cours 2, 11/03/2025
Méthodes Multigrilles

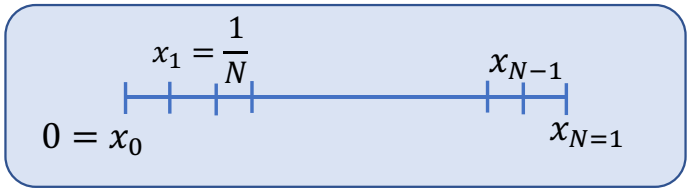
Elements of multigrid

Relaxation schemes and smoothing properties

Model problem: 1D Poisson equation

- The 1D Poisson equation:

$$-u'' = f$$
$$u(0) = u(1) = 0$$



- Grid points

$$h = \frac{1}{N}, \quad x_i = ih, \quad i = 0, .., N$$

- Finite difference scheme

$$u_i'' \approx \frac{-u_{i-1} + 2u_i - u_{i+1}}{h^2} = f_i, \qquad i = 1, .., N-1$$

$$u_0 = u_N = 0$$

- We obtain a linear system

$$A\mathbf{u} = \frac{1}{h^2} \begin{pmatrix} 2 & -1 & 0 & \dots & 0 \\ -1 & 2 & -1 & \ddots & \vdots \\ 0 & -1 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & 2 & -1 \\ 0 & \dots & 0 & -1 & 2 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_{N-1} \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_{N-1} \end{pmatrix}$$

Application to our example: Jacobi method

- Remember the discretization at point i (this gives one matrix row):

$$-u_{i-1} + 2u_i - u_{i+1} = h^2 f_i$$

$$A = \frac{1}{h^2}\begin{pmatrix} 2 & -1 & 0 & \dots & 0 \\ -1 & 2 & -1 & \ddots & \vdots \\ 0 & -1 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & 2 & -1 \\ 0 & \dots & 0 & -1 & 2 \end{pmatrix}$$

- Solving for u_i:

$$u_i = \frac{1}{2}(u_{i-1} + u_{i+1} + h^2 f_i)$$

- We can write this iteratively as

$$u_i^{(m+1)} = \frac{1}{2}(u_{i-1}^{(m)} + u_{i+1}^{(m)} + h^2 f_i)$$

- This can be expressed in matrix form as

$$\begin{aligned} \mathbf{u}^{(m+1)} &= D^{-1}(L+U)\mathbf{u}^{(m)} + D^{-1}h^2\mathbf{f} \\ &= \underbrace{(I - D^{-1}A)}_{:=R}\mathbf{u}^{(m)} + D^{-1}h^2\mathbf{f} \end{aligned}$$

$$R = \frac{1}{2}\begin{pmatrix} 0 & 1 & & \\ 1 & 0 & \ddots & \\ & \ddots & \ddots & 1 \\ & & 1 & 0 \end{pmatrix}$$

Weighted Jacobi method

To increase convergence properties, we next look at the weighted Jacobi method.

We compute

$$u_i^\star = \frac{1}{2}(u_{i-1} + u_{i+1} + h^2 f_i)$$

This is now only an intermediate step, and we define the new weighted iterate by

$$u_j^{(m+1)} = (1-\omega)u_j^{(m)} + \omega u_i^\star, \qquad i \le j \le N-1$$

In matrix notation, we can express it by

$$\begin{aligned}\mathbf{u}^{(m+1)} &= \left[(1-\omega)\mathbf{I} + \omega S_J\right]\mathbf{u}^{(m)} + \omega D^{-1}\mathbf{f}^{(m)} \\ &= (1-\omega D^{-1})A\mathbf{u}^m + \omega D^{-1}\mathbf{f}\end{aligned}$$

Define
$R_\omega := (I - \omega D^{-1}A)$

Eigenvectors and Eigenvalues

- The matrix is spd and sparse (not more than 3 non-zero entries per row and column).

$$A = \frac{1}{h^2}\begin{pmatrix} 2 & -1 & 0 & \dots & 0 \\ -1 & 2 & -1 & \ddots & \vdots \\ 0 & -1 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & 2 & -1 \\ 0 & \dots & 0 & -1 & 2 \end{pmatrix}$$

- It has the eigenvalues $(\lambda_k \mathbf{w}_k = A\mathbf{w}_k)$

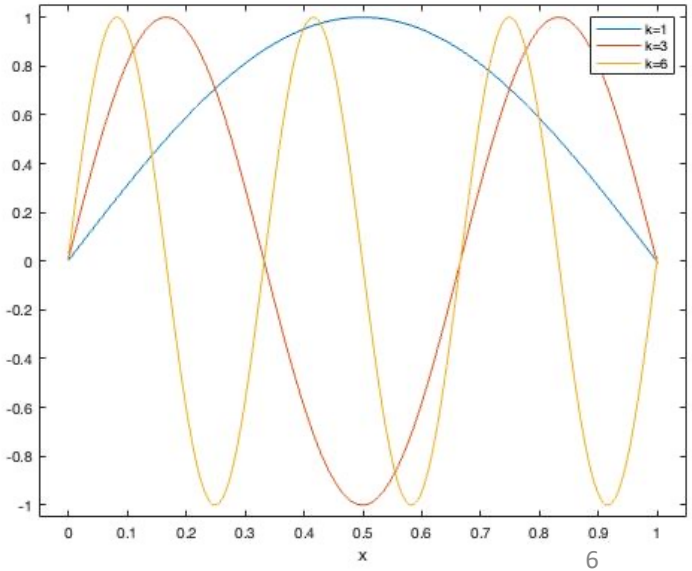
$$\lambda_k = \frac{4}{h^2}\sin^2\left(\frac{k\pi}{2N}\right)$$

- And the eigenvectors

$$(w_k)_j = \sin\left(\frac{jk\pi}{N}\right)$$

- This denotes the *j*-th component of the *k*-th eigenvector.

- The eigenvectors are Fourier modes.



Definition

Modes in the upper half of the spectrum, i.e.

$$\frac{N}{2} \leq k < N - 1$$

are called high-frequency or oscillatory modes**.**

Modes in the lower half of the spectrum, i.e.

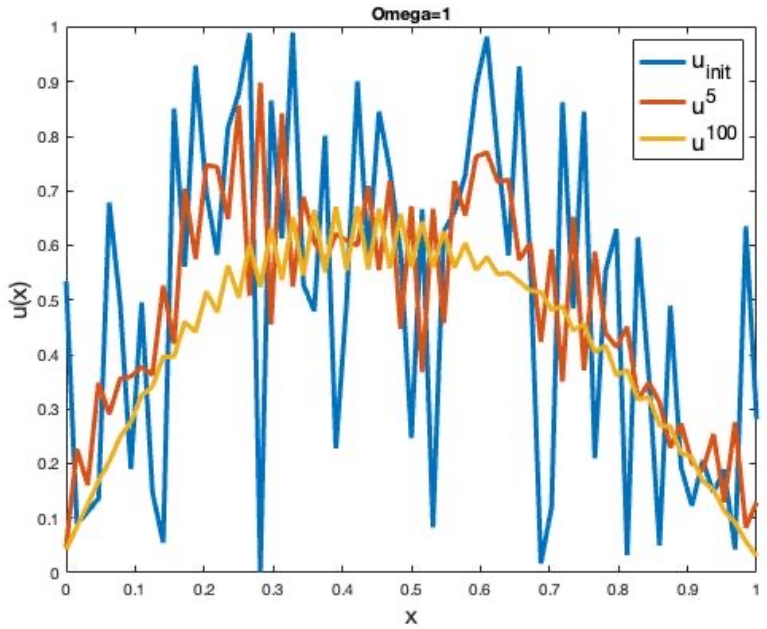
$$1 \leq k < \frac{N}{2}$$

are called low-frequency or smooth modes.

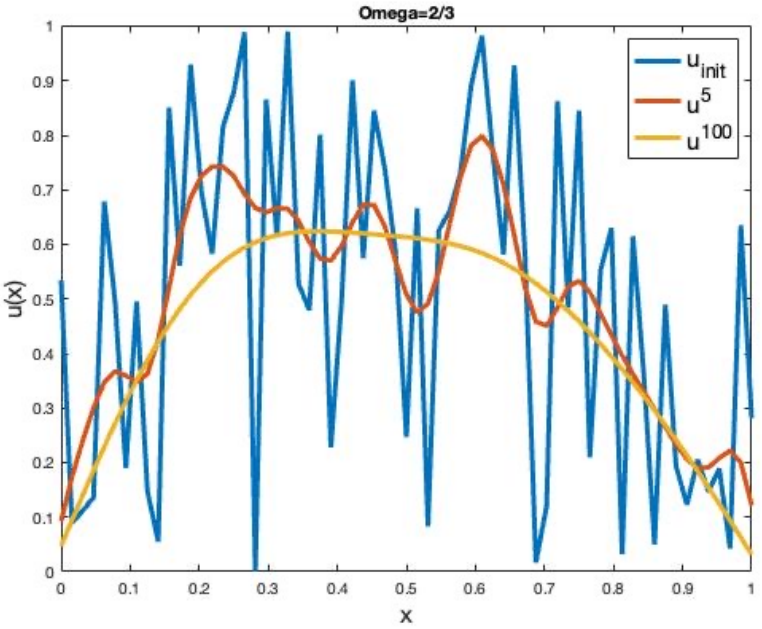
Note that the classification of smooth or oscillatory wave number depends on the total number N of grid points. A fixed wave number *k* might thus be smooth on one, but oscillatory on another grid.

Smoothing with Jacobi and weighted Jacobi

- Let $f = 0$ (thus solution $u = 0$). Then random initial guess → random error



Jacobi $R = I - D^{-1}A$

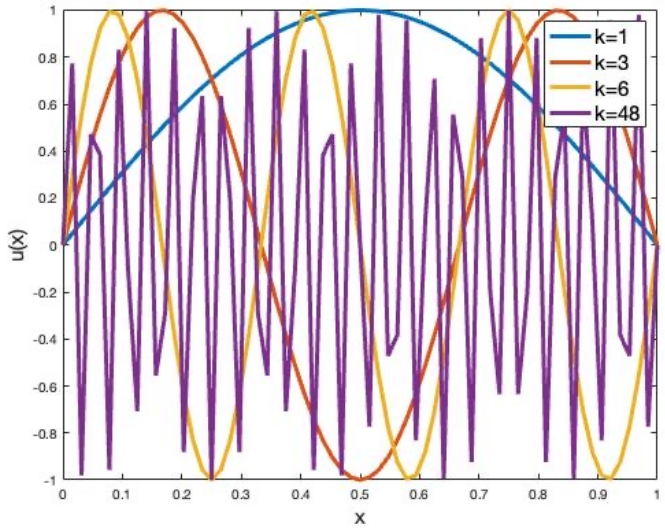


Weighted Jacobi $R_\omega = I - \frac{2}{3}D^{-1}A$

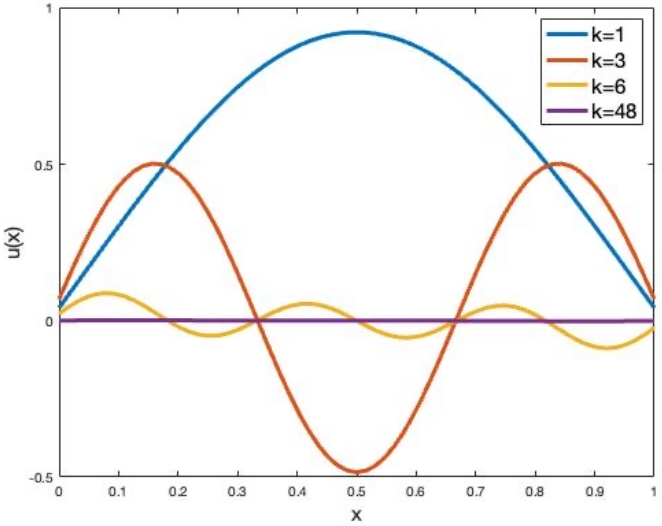
Weighted Jacobi

- Let us use the four modes: k = 1, 3, 6, 48, as initial guess.
- Then smooth modes will dampen less quickly than higher ones.

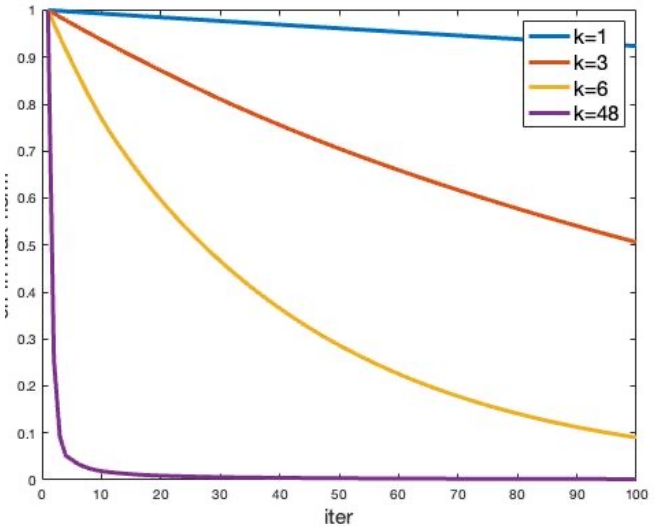
Initial error



Error after 100 iterations



Error decrease



WHY?

A worked example: 1D Poisson

We now use the weighted Jacobi iteration matrix in the version

$$R_\omega = I - \omega D^{-1} A$$

For the 1D Poisson example, we obtain the iteration matrix

$$R_\omega = I - \frac{\omega}{2} \begin{pmatrix} 2 & -1 & & \\ -1 & 2 & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 2 \end{pmatrix}$$

It follows the relationship of the Eigenvalues

$$\lambda(R_\omega) = I - \frac{\omega h^2}{2} \lambda(A)$$

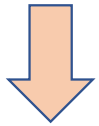
Eigenvalues and Eigenvectors

The Eigenvalues of *A* are given as

$$\lambda_k(A) = \frac{4}{h^2}\sin^2\left(\frac{k\pi}{2N}\right)$$

and the Eigenvectors as

$$(w_k)_j = \sin\left(\frac{jk\pi}{N}\right), \qquad 1 \le k \le N-1, \;\; 0 \le j \le N$$



It follows for R_ω:

$$\lambda_k(R_\omega) = 1 - 2\omega\sin^2\left(\frac{k\pi}{2N}\right)$$

The Eigenvectors are the same as for *A*, thus $(w_k)_j = \sin\left(\frac{jk\pi}{N}\right), \qquad 1 \le k \le N-1, \;\; 0 \le j \le N$

Eigenvector expansion of error

We can represent the error $\boldsymbol{e}^{(0)}$ using the Eigenvectors of A.

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

Where the coefficients c_k give the 'amount' of each mode in the error. We know furthermore that

$$\mathbf{e}^{(m)} = R_\omega^m \mathbf{e}^{(0)}.$$

Now using the eigenvector expansion for $\boldsymbol{e}^{(0)}$, we get

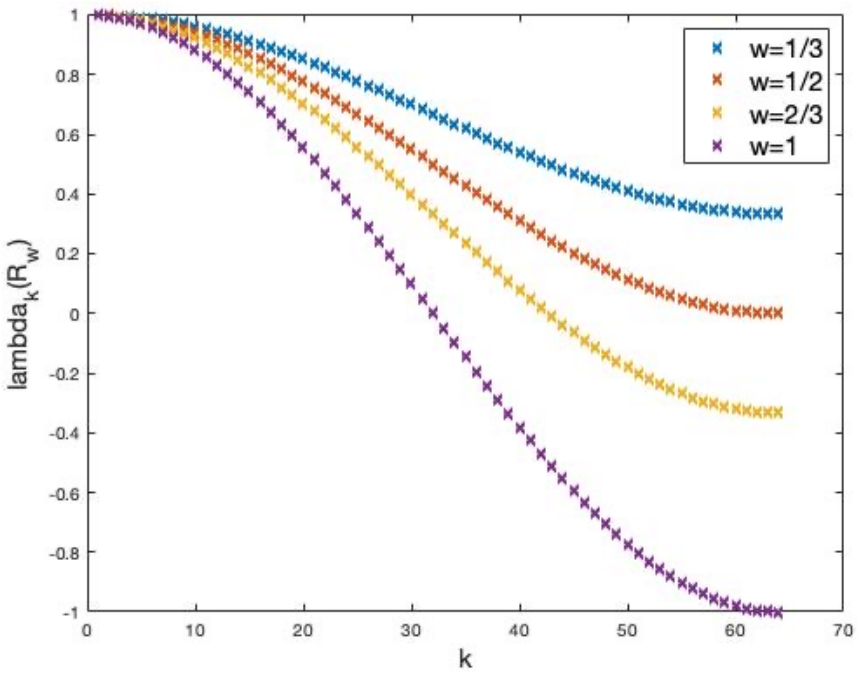
$$\mathbf{e}^{(m)} = R_\omega^m \mathbf{e}^{(0)} = \sum_{k=1}^{N-1} c_k R_\omega^m \mathbf{w}_k = \sum_{k=1}^{N-1} c_k \lambda_k^m (R_\omega) \mathbf{w}_k.$$

After m iterations, the kth mode of the initial error has been reduced by a factor of $\lambda_k^m(R_\omega)$.

If $|\lambda_k|$ is close to 1, then the reduction will be slow. If $|\lambda_k|$ is close to 0, then it will be fast.

What choice of ω gives the best iterative scheme in our example?

Choice of $\omega = 1$

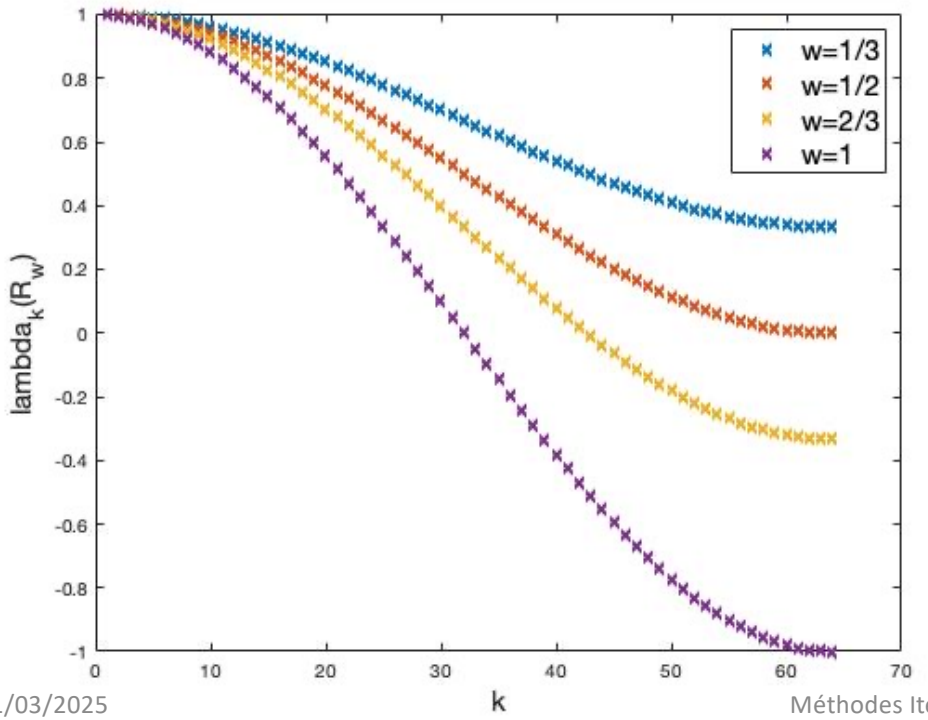


Case $\omega = 1$

- We obtain the non-weighted Jacobi method.
- Here, we see that for small *k*, but also for larger *k* close to *N-1* in the oscillatory spectrum, the eigenvalues are close to 1.
- Convergence is thus also slow in these cases.
- This illustrates why the weighted Jacobi method is advantageous.

Choice of ω

- Recall: for $0 < \omega \leq 1$, we have $|\lambda_k(R_\omega)| < 1$.
- Find the value ω that makes $|\lambda_k(R_\omega)|$ as small as possible for all k



For all values of ω, the eigenvalues associated to the 'smoothest' modes are close to 1.

We have

$$\lambda_1 = 1 - 2\omega \, sin^2\left(\frac{\pi h}{2}\right) \approx 1 - \frac{\omega \, \pi^2 h^2}{2}$$

Thus λ_1 will always be close to one, no matter which ω. It's getting even worse, the smaller h!

Optimal choice of ω

To find the optimal value of ω, we search for the smallest interval $\left[-\bar{\lambda}, \bar{\lambda}\right]$ with $\lambda_k(R_\omega) \in \left[-\bar{\lambda}, \bar{\lambda}\right]$ for $\frac{N}{2} \leq k \leq N-1$. This can be done for example by

$$-\lambda_{N/2}(R_\omega) = \lambda_{N-1}(R_\omega)$$

and we obtain

$$\omega = \frac{2}{3}$$

Definition: The multigrid smoothing factor

- The smoothing factor of a relaxation method R is the maximum magnitude of the upper half of the spectrum

$$\max_{k \in [\frac{N}{2}, N]} |\lambda_k(R_\omega)|$$

- In the example above, we have

$$\max_{k \geq N/2} |\lambda_k(R_{2/3})| = \max_{k \geq \frac{N}{2}} \left|1 - \frac{4}{3} sin^2\left(\frac{k\pi}{2N}\right)\right| \leq \frac{1}{3}$$

$\Longrightarrow$ The oscillatory components are reduced at least **by a factor of 3** at each relaxation.

- We thus see furthermore, that the bound is independent of the mesh size $h = \frac{1}{N}$.

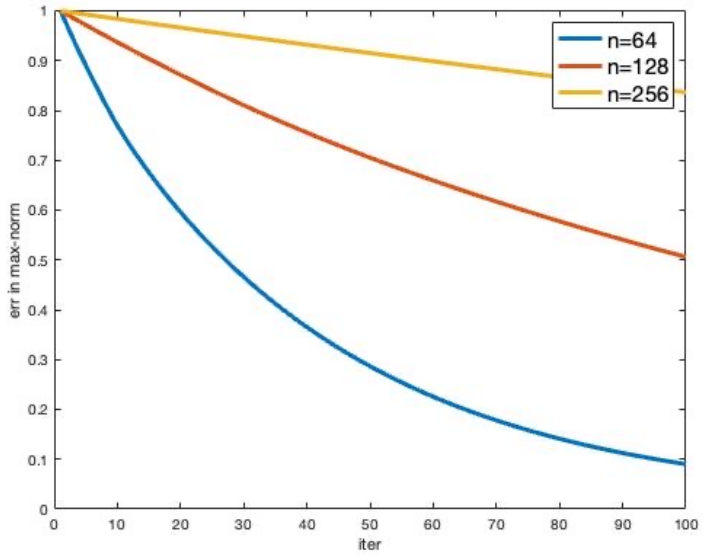
A common feature:

Oscillatory modes are quick to converge

Smooth modes are slow to converge

Fixed wave number on different grids

- Let us still see what happens for the fixed wave number *k=6* on different grids.
- We use weighted Jacobi relaxation with $\omega = \frac{2}{3}$ and show the error for the first 100 iterations



Observation: For a fixed wave number, the error is reduced better on a coarser than on a finer grid.

Summary: Classification of modes

- Consider a fixed Fourier mode $\sin(k\pi x)$ and its discrete representation $\sin\left(\frac{jk\pi}{N}\right), j = 1, .., N-1.$

Then the classification of this mode depends on the fineness of the grid.

- If the grid is sufficiently fine, i.e. $k < \frac{N}{2}$, it is a smooth mode.
- If the grid is sufficiently coarse, i.e. $\frac{\mathrm{N}}{2} \leq k < N-1$, then it is an oscillatory mode.

The damping property depends on the smoothness of the mode.

- If it is a smooth mode, then the weigthed Jacobi method will damp it only slowly.
- If it is an oscillatory mode, then the weigthed Jacobi method may damp it quickly.

Relaxation schemes

Gauss-Seidel method

- Calculate an entry u_k , with $u = (u_1, .., u_n)$, of the new iteration and use it in the computation of $u_i, i = k + 1, .., n$.

$$\boldsymbol{u}^{(m+1)} = (D - L)^{-1} U \, \boldsymbol{u}^{(m)} + (D - L)^{-1} \boldsymbol{f}$$

Define
$S_{GS} := (D - L)^{-1} U$

SOR method

$$\boldsymbol{u}^{(m+1)} = -(\mathrm{D} - \omega L)^{-1}(-\omega U + (\omega - 1)D)\boldsymbol{u}^{(m)} + \omega(\mathrm{D} - \omega L)^{-1}\mathbf{f}$$

$S_{GS\omega} := (D - \omega L)^{-1}(-\omega U + (\omega - 1)D)$

- If $\omega = 1$, then the Gauss-Seidel method is recovered.

Gauss-Seidel method

- The eigenvalues of $S_{GS} = (D - L)^{-1}U$ are given by

$$\lambda_k(S_{GS}) = cos^2\left(\frac{k\,\pi}{N}\right)$$

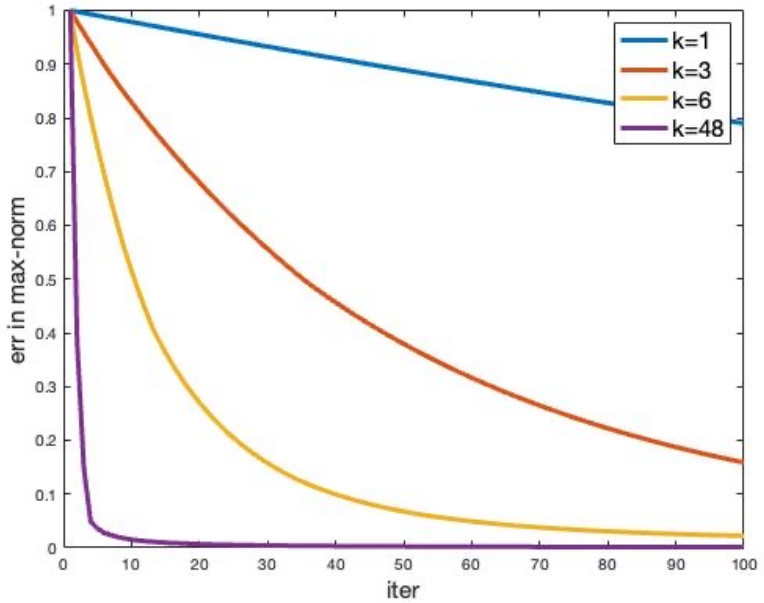
- The eigenvectors of S_{GS} are given by

$$w_{k,j}(S_{GS}) = \left[cos\left(\frac{k\,\pi}{N}\right)\right]^2 \sin\left(\frac{jk\,\pi}{N}\right)$$

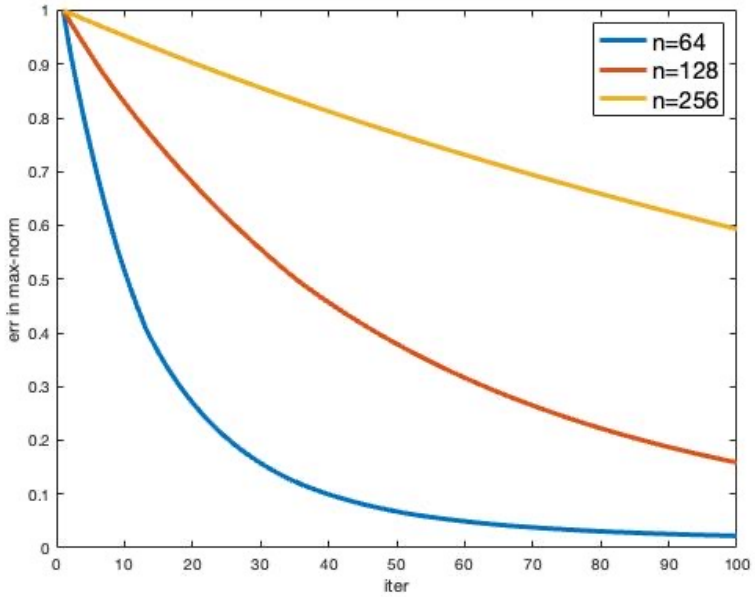
- These are not the same as for *A*. The convergence analysis has thus to be done carefully. We do not want to get into further details here.

Gauss-Seidel method

- With the eigenvectors of *A* as initial guess, the damping behaviour of the Gauss-Seidel method is qualitatively the same as for the Jacobi method.



Fixed number of nodes *n=64*



Fixed wave number *k*=6

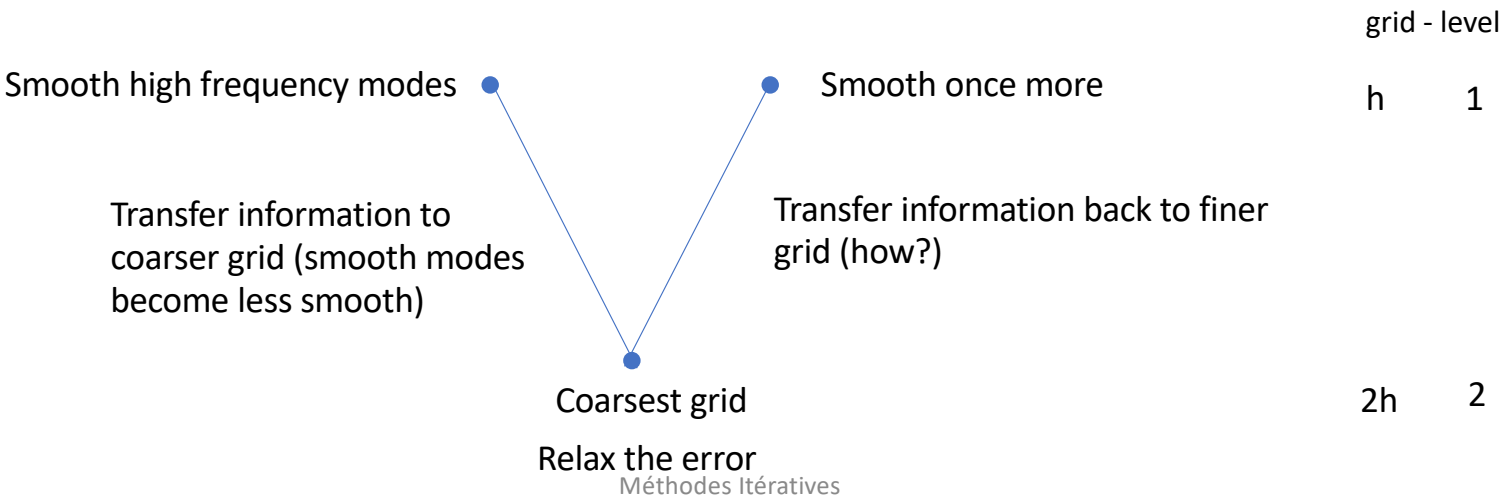
Summary: Relaxation schemes and damping

- Classical iterative schemes might damp highly oscillating discrete error modes very quickly.
- There is only a slow damping of the smooth discrete error modes.
- The smoothness of an error mode depends on the grid. A smooth error mode on a given grid is generally on a coarser grid less smooth.

Improvements possible for all error components?

- We have seen that with a good initial guess, the relaxation scheme converges faster (e.g. $k = 48$ against $k = 1$). Some iterations on a coarse grid could help.
- Could we use this coarse grid idea otherwise?

A two-grid scheme?

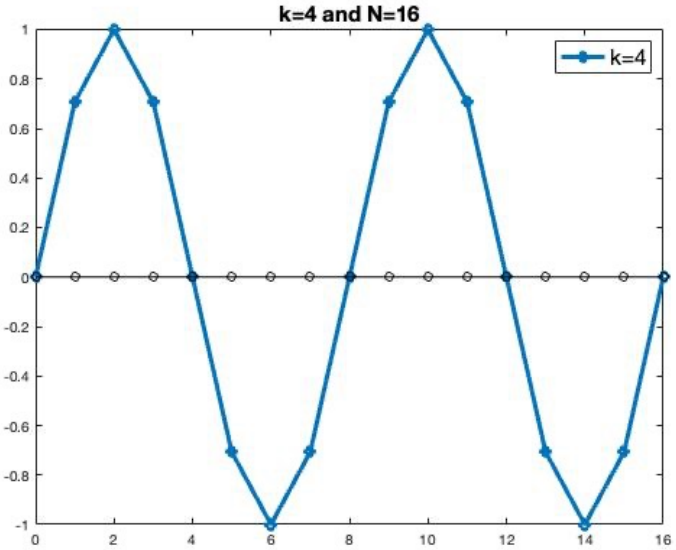


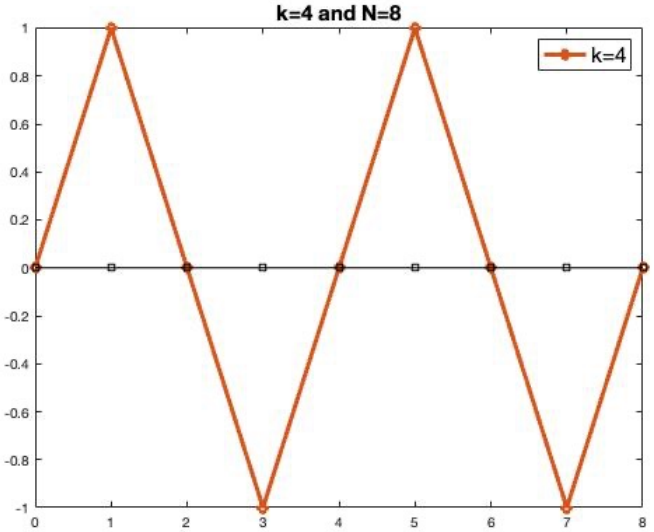
Grid transfer

Coarse grids

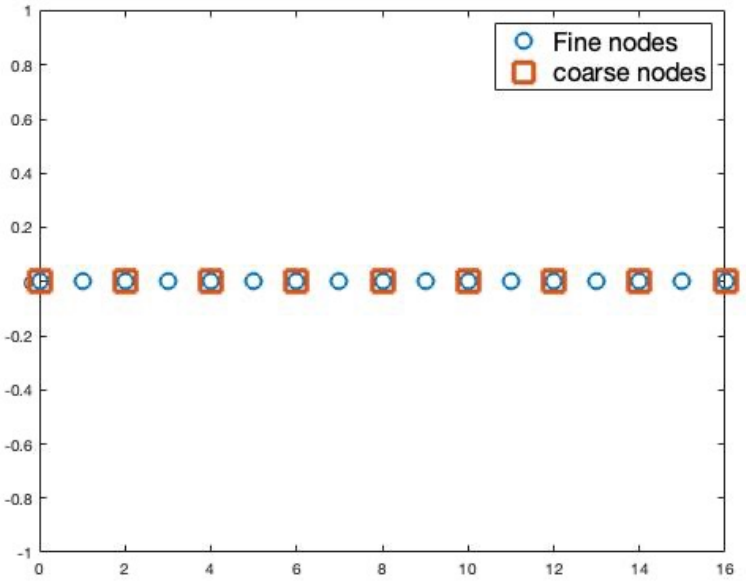
- Imagine all oscillatory modes are **damped out** and we are left only with **smooth modes**.
- Smooth modes look like **oscillatory modes** when sampled on a **coarse grid**.

4-mode of 15 versus 4-mode of 7





Coarse modes – Low frequency



We look at the *k*-th mode on the fine grid evaluated at the even-numbered grid points.

If $1 \leq k \leq \frac{N}{2}$, i.e. **lower part** of **spectrum**, then the components may be rewritten as

$$w^h_{k,2j} = \sin\left(\frac{2jk\pi}{N}\right) = \sin\left(\frac{jk\pi}{\frac{N}{2}}\right) = w^{2h}_{k,j}, \qquad 1 \leq j \leq \frac{N}{2}$$

- The *k*-th mode on Ω^h becomes the *k*-th mode on Ω^{2h}
- Passing from fine to coarse grid, a smooth mode becomes more oscillatory

Coarse modes – High frequency

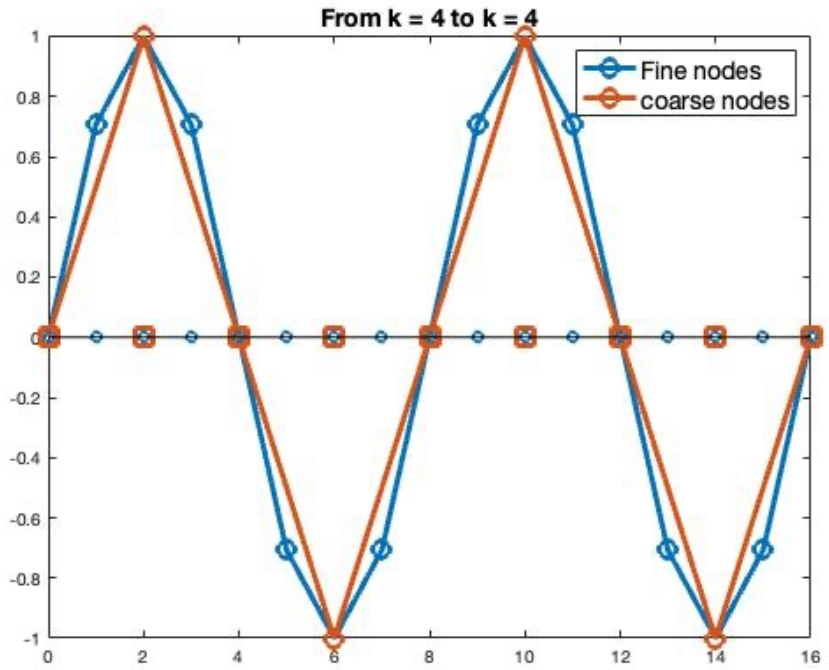
For the **upper part** of the **spectrum**, i.e. $\frac{N}{2} < k \leq N-1$, compute:

$$
\begin{aligned}
-w^{2h}_{N-k,j} &= -\sin\left(\frac{2j(N-k)\pi}{N}\right) \\
&= -\sin\left(\frac{2jN\pi}{N} - \frac{2jk\pi}{N}\right) \\
&= -\sin(2j\pi)\cos\left(\frac{2jk\pi}{N}\right) + \cos(2j\pi)\sin\left(\frac{2jk\pi}{N}\right) \\
&= \sin\left(\frac{2jk\pi}{N}\right) = w^{h}_{k,2j}
\end{aligned}
$$

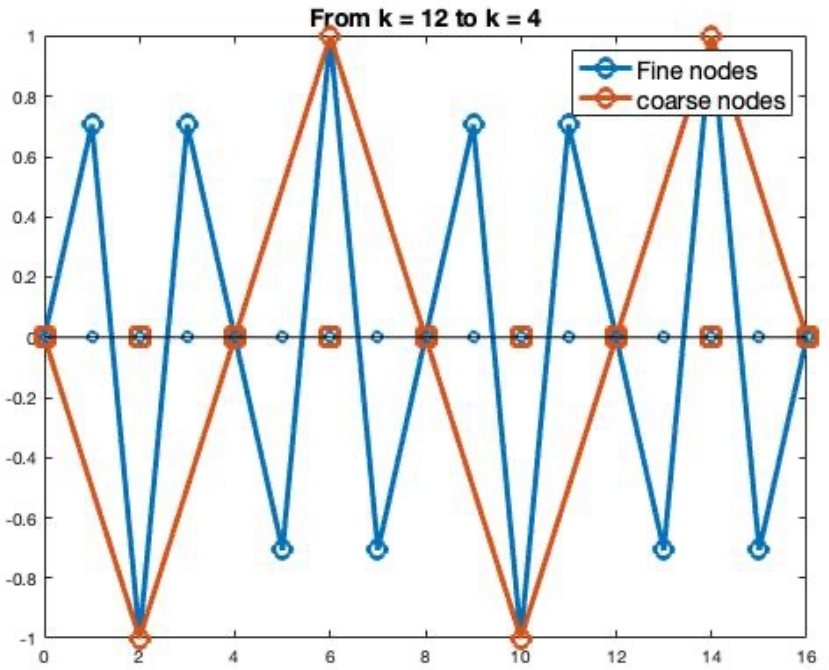
- The *k*-th mode on Ω^h becomes the negative of the *(N-k)*-th mode on Ω^{2h}.
- The oscillatory modes on the fine grid appear relatively smooth on the coarse grid.

- It is advisable to **damp out** the oscillatory modes **before** passing to the coarser grid. Otherwise we obtain additional smooth modes on the coarser grid.

Coarse modes



Low modes -> k-modes are *preserved*



High modes -> k-modes are *aliased*

Some key observations so far

1. Relaxation may be extremely efficient for smoothing the error relative to the grid.

2. A smooth error can be approximated well on a coarser grid.

3. A coarser grid implies less variables, hence less computation.

4. On the coarser grid, the error is no longer as smooth relative to the grid, so relaxation may once again be efficient.

First strategy: Nested iterations

- Use coarse grids to obtain an initial guess for the next finer grid.

- Relax on $A\mathbf{u}^{h_0} = \mathbf{f}^{h_0}$ on a very coarse grid to obtain an initial guess for the next finer grid

⋮

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

Questions:

- What does it mean to relax on the coarser grid (i.e., how to define the equations to be solved thereon?)
- What if some smooth components remain? –> The algorithm will stall on the fine grid.

The residual equation

- An iterative method for $A\boldsymbol{u} = \boldsymbol{f}$ can either be applied to this equation directly, or to an equation formulated for the error. The next iterate will then be corrected by the defect

$$\mathbf{u}^{(m+1)} = \mathbf{u}^{(m)} + \mathbf{e}^{(m)}$$

- Let $\boldsymbol{u}^{(m)}$ be the approximation of $\boldsymbol{u}$ at the *m*-th iteration. The error is then given by $\mathbf{e}^{(m)} = \mathbf{u} - \mathbf{u}^{(m)}$.
- The error satisfies the equation

$$A\mathbf{e}^{(m)} = A\mathbf{u} - A\mathbf{u}^{(m)} = \mathbf{f} - A\mathbf{u}^{(m)} =: \mathbf{r}^{(m)}.$$

- This equation is called the residual equation.

Second strategy: Coarse grid correction (Two-level method)

- We now use the residual equation and relax on the actual error.
- The new iterate is then an update of the previous iterate corrected with the new residual

Smooth $A^h \boldsymbol{u}^h = \boldsymbol{f}^h$ on Ω^h, call solution $\boldsymbol{v}^h$. → We obtain an approximation of the solution, that is to be updated.

Compute the residual $\boldsymbol{r}^h = \boldsymbol{f}^h - A^h \boldsymbol{v}^h$.

Project (restrict) the residual to Ω^{2h}, called $R(\boldsymbol{r}^h)$. → Information is transferred to the coarse grid.

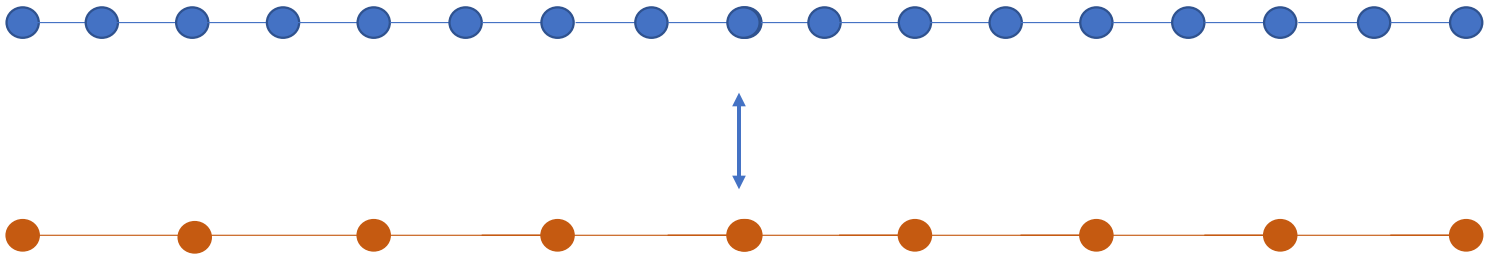
Solve $A^{2h} \boldsymbol{e}^{2h} = R(\boldsymbol{r}^h)$ on Ω^{2h} → We obtain an approximation $\boldsymbol{e}^{2h}$ of the error

Project (prolongate) $\boldsymbol{e}^{2h}$ to Ω^h, denoted $P(\boldsymbol{e}^{2h})$. → Information is transferred back to fine grid

Update the approximate solution on Ω^h by $\boldsymbol{v}^h = \boldsymbol{v}^h + P(\boldsymbol{e}^{2h})$

- How to define the system on the coarse grid?
- How to restrict and prolongate from one grid to the other?

Questions



- How to transfer between fine and coarse grid?
- Which equations do we want to solve on the coarse grid?
- We use superscripts *h* on the variable, indicating the grid they are defined on, e.g., $\boldsymbol{u}^h$ is defined on Ω^h and $\boldsymbol{u}^{2h}$ is defined on Ω^{2h}.

Grid transfer

As we have seen, we need to smooth the error on the fine grid first and only then solve the coarse-grid problem.

Hence, we need two types of intergrid transfer operators:

- A restriction operator (fine-to-coarse): I_h^{2h}
- A prolongation operator (coarse-to-fine): I_{2h}^h

Let us start with the prolongation operator first.

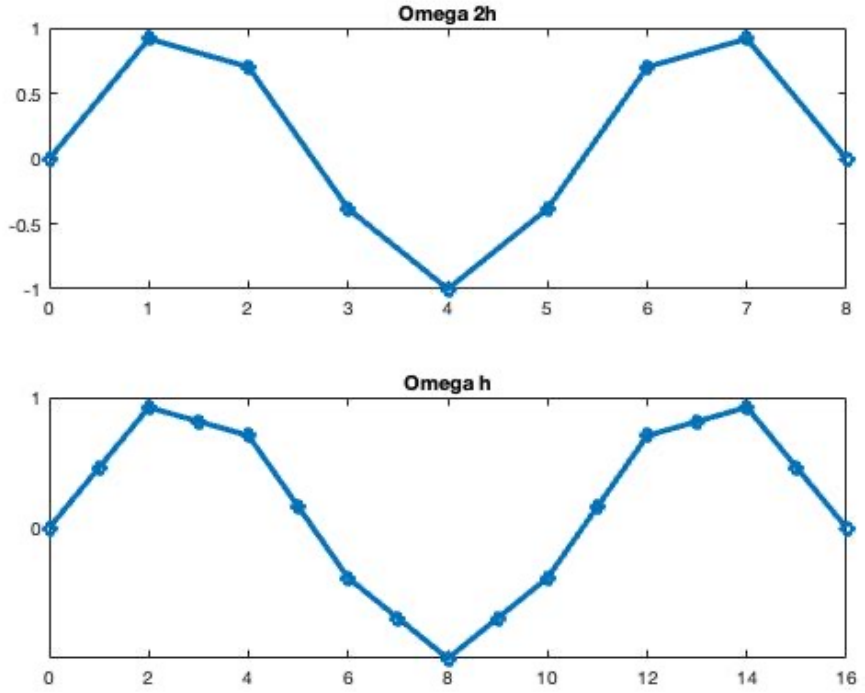
Prolongation: Linear interpolation

- Find a linear interpolation operator $I_{2h}^{h} : \mathbb{R}^{\frac{N}{2}-1} \rightarrow \mathbb{R}^{N-1}$.
- I_{2h}^{h}: coarse grid -> fine grid, with

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

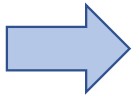
- It can be written in matrix form (here for the case *N=8*):

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



How well does interpolation work?

- Assume that the real error is smooth on the fine grid.
- Assume also that a coarse-grid approximation is given on Ω^{2h} and that it is exact on the coarse nodes.
- When this coarse grid correction is interpolated to the fine grid, the interpolant is also smooth.

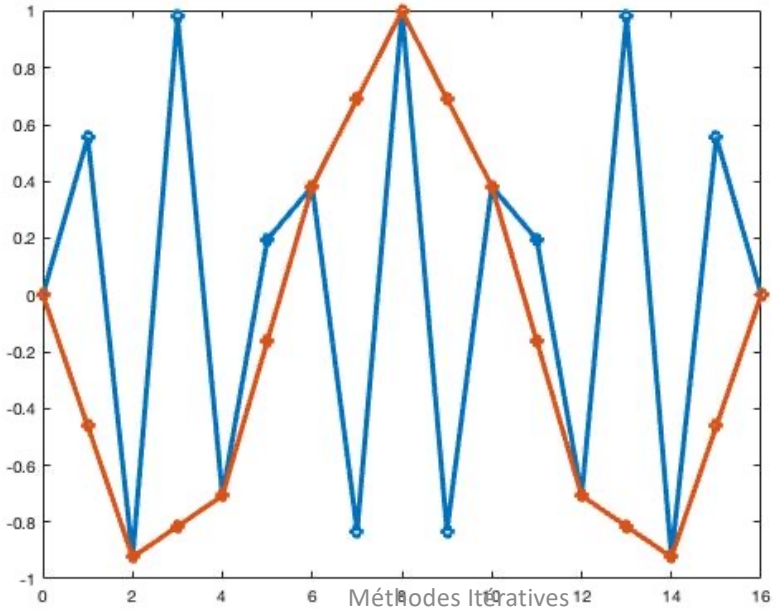


We expect a relatively good approximation to the fine grid error.

Oscillatory real error

- Assume that the real error is oscillatory

Even a very good coarse-grid approximation may produce an interpolant that is not very accurate.



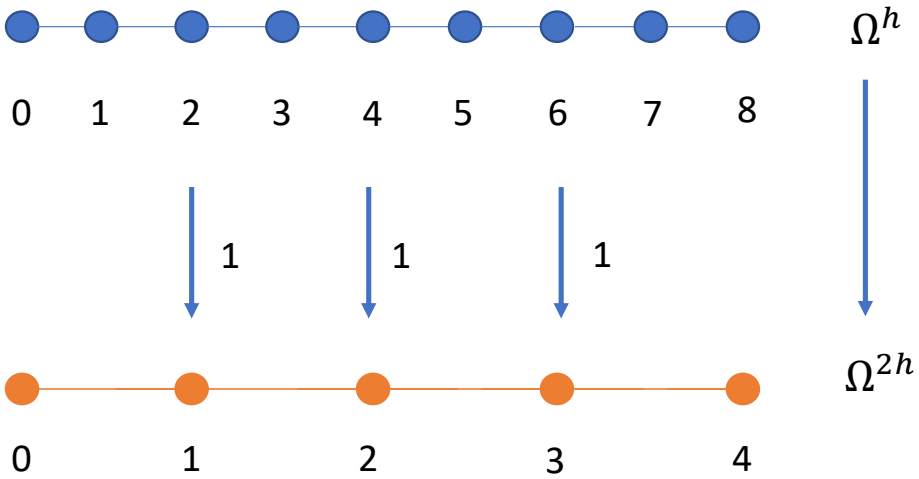
Summary prolongation

- All in all, interpolation gives the best result, if the error on the fine grid is smooth.
- The prolongation is an appropriate complement to the smoother, that works most efficiently if the error is oscillating.

Restriction operator - Injection

- Want to find a good restriction operator $I_h^{2h} : \mathbb{R}^{N-1} \to \mathbb{R}^{\frac{N}{2}-1}$.
- The most obvious restriction operator I_h^{2h} is injection, where

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

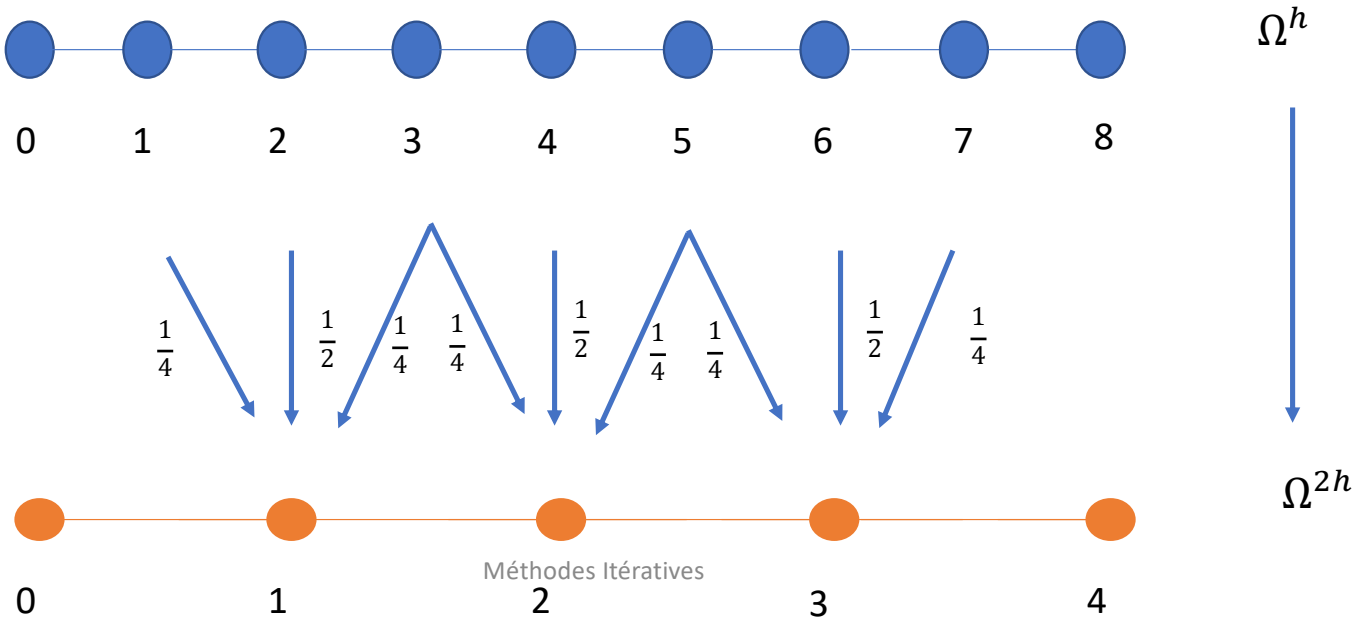


- However, the injection operator often does not lead to an efficient method.

Restriction operator – full weighting

- The full weighting restriction operator $I_h^{2h} : \mathbb{R}^{N-1} \to \mathbb{R}^{\frac{N}{2}-1}$. takes all grid points into account.

$$v_j^{2h} = \frac{1}{4}(v_{2j-1}^h + 2v_{2j}^h + v_{2j+1}^h)$$



Full weighting – matrix representation

For the case *N = 8,* we have

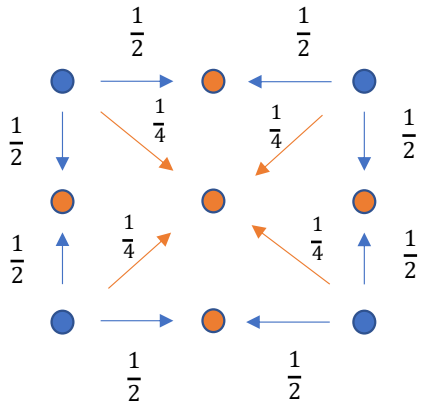
$$I_h^{2h} v^h = \frac{1}{4} \begin{bmatrix} 1 & 2 & 1 & & & & & \\ & & 1 & 2 & 1 & & & \\ & & & & 1 & 2 & 1 & \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \end{bmatrix}_h = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}_{2h} = v^{2h}$$

Note that $I_{2h}^h = c(I_h^{2h})^T$ for $c \in \mathbb{R}$, i.e.

- the full weighting operator is the transpose of the linear interpolation operator
- It is a linear operator and has a rank of $\frac{N}{2} - 1$ and a null space of dimension $\frac{N}{2}$

Prolongation operator for 2D problem

- The interpolation operator may be defined in a similar way.
- Let $I_{2h}^{h}\boldsymbol{v}^{2h} = \boldsymbol{v}^{h}$. The components of $\boldsymbol{v}^{h}$ are given for $0 \leq i, j \leq \frac{N}{2} - 1$ by



$$v_{2i,2j}^{h} = v_{ij}^{2h}$$

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

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

$$v_{2i+1,2j+1}^{h} = \frac{1}{4}(v_{ij}^{2h} + v_{i+1,j}^{2h} + v_{i,j+1}^{2h} + v_{i+1,j+1}^{2h})$$

Full weighting operator in 2D

- Let $I_h^{2h}\mathbf{v}^h = \mathbf{v}^{2h}$. The components of $\boldsymbol{v^h}$ are given for $1 \leq i, j, \leq \frac{N}{2} - 1$ by

$$v_{i,j}^{2h} = \frac{1}{16}\left[v_{2i-1,2j-1}^h + v_{2i-1,2j+1}^h + v_{2i+1,2j-1}^h + v_{2i+1,2j+1}^h \right.$$
$$\left. + 2(v_{2i,2j-1}^h + v_{2i,2j+1}^h + v_{2i-1,2j}^h + v_{2i+1,2j}^h) + 4v_{i,j}^h\right]$$

The two level method

- ✅ Smooth $A^h \boldsymbol{u}^h = \boldsymbol{f}^h$ on Ω^h.
- ✅ Compute the residual $\boldsymbol{r}^h = \boldsymbol{f}^h - A^h \boldsymbol{u}^h$.
- ✅ Project (restrict) the residual to Ω^{2h}, called $R(\boldsymbol{r}^h)$.
- Solve $A^{2h} \boldsymbol{e}^{2h} = R(\boldsymbol{r}^h)$ on Ω^{2h}
- ✅ Project (prolongate) $\boldsymbol{e}^{2h}$ to Ω^h, called $P(\boldsymbol{e}^{2h})$
- ✅ Update the approximation of the solution on Ω^h by $\mathbf{v}^h = \boldsymbol{v}^h + P(\boldsymbol{e}^{2h})$

How to construct the coarse grid matrix A^{2h} ?

Coarse grid matrix

Straightforward approach:

- Define A^{2h} by applying a discretization method for the differential operator on Ω^{2h}.

Second approach:

- Use the intergrid transfer operators to define the coarse grid matrix A^{2h}.

Galerkin projection

- Here we use the prolongation and restriction operators to define the coarse grid matrix.
- Starting point is the residual equation

$$A^h \mathbf{e}^h = \mathbf{r}^h$$

- Let $\boldsymbol{e}^h$ be in the range of the prolongation operator I_{2h}^h, i.e.

$$\mathbf{e}^h = I_{2h}^h(\mathbf{e}^{2h})$$

- We substitute the latter equation into the first equation and obtain

$$A^h I_{2h}^h(\mathbf{e}^{2h}) = \mathbf{r}^h$$

Galerkin projection

Applying on both sides the restriction operator, we obtain

$$I_h^{2h} A^h I_{2h}^h(\mathbf{e}^{2h}) = I_h^{2h}(\mathbf{r}^h)$$

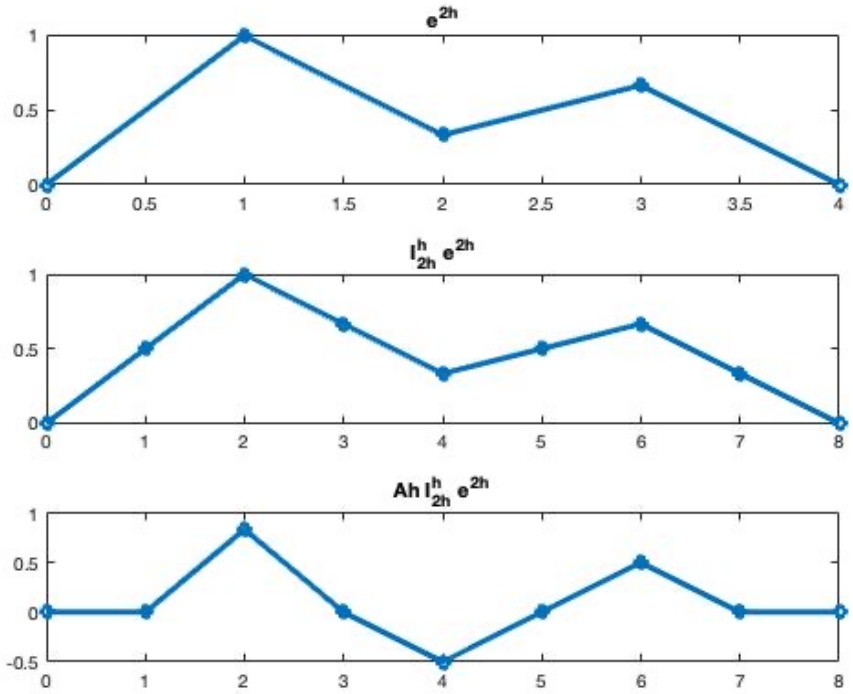
Remember that the coarse grid problem is given as

$$A^{2h}\mathbf{e}^{2h} = I_h^{2h}(\mathbf{r}^h)$$

We can thus conclude that

$$A^{2h} := I_h^{2h} A^h I_{2h}^h$$

Galerkin operator graphically – Poisson 1D



$$A = \frac{1}{h^2}\begin{pmatrix} 2 & -1 & 0 & \dots & 0 \\ -1 & 2 & -1 & \ddots & \vdots \\ 0 & -1 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & 2 & -1 \\ 0 & \dots & 0 & -1 & 2 \end{pmatrix}$$

- Assume that $\boldsymbol{e}^h = I_{2h}^h \boldsymbol{e}^{2h}$

Apply A^h to obtain $A^h I_{2h}^h \boldsymbol{e}^{2h}$, then:

- The odd rows are 0
- The even rows correspond to the coarse grid nodes

Coarse grid matrix – 1D Poisson Finite Difference

$$A = \frac{1}{h^2}\begin{pmatrix} 2 & -1 & 0 & \dots & 0 \\ -1 & 2 & -1 & \ddots & \vdots \\ 0 & -1 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & 2 & -1 \\ 0 & \dots & 0 & -1 & 2 \end{pmatrix}$$

- To obtain the coarse grid matrix, we apply the Galerkin operator to the *j*-th unit vector $\hat{\boldsymbol{e}}_j^{2h}$

	J-1		j		J+1
$\hat{\boldsymbol{e}}_j^{2h}$	0		1		0
$I_{2h}^h \hat{\boldsymbol{e}}_j^{2h}$	0	$\frac{1}{2}$	1	$\frac{1}{2}$	0
$A^h I_{2h}^h \hat{\boldsymbol{e}}_j^{2h}$	$-\frac{1}{2h^2}$	0	$\frac{1}{h^2}$	0	$-\frac{1}{2h^2}$
$I_h^{2h} A^h I_{2h}^h \hat{\boldsymbol{e}}_j^{2h}$	$-\frac{1}{4h^2}$		$\frac{1}{2h^2}$		$-\frac{1}{4h^2}$

- This is the indeed the finite difference stiffness matrix A^{2h} on Ω^{2h}.

Galerkin projection

- The preceding argument is based on the assumption that $\boldsymbol{e}^h$ lies in the range of interpolation.
- In general, this is not the case.
- However, it gives a sensible definition for A^{2h}.

- Assume that e^h does lie in the range of iterpolation
- Then doing the two-grid correction would give the exact (direct) solution.

Exercise: try this out for 1D Poisson.

About the convergence of the two-level method

Two-level method

- We use the two-level method only with pre-smoothing (no post-smoothing).

- Remember that a stationary linear iteration scheme may be expressed in the form

$$\boldsymbol{v}^{(m)} = (1 - BA)^m \boldsymbol{v}^{(0)} + Bf = S^m \ \boldsymbol{v}^{(0)} + C(\boldsymbol{f})$$

- We want to find the iteration matrix S_{2lev} of the two-level method

Let

- S_{sm} be the iteration matrix of the smoother
- $\boldsymbol{v}^{(m)}$ is the approximation before the pre-smoothing and $\boldsymbol{v}^{(m+1)}$ the result after the update
- $\boldsymbol{v}_{\nu}^{(m)}$ is the approximation at step *m* after ν pre-smoothing steps

Iteration matrix

We know that for the error in the pre-smoothing iterations holds

$$\mathbf{e}^{(\nu)} = S_{sm}^{\nu}\mathbf{e}^{(0)}$$

with

$$\mathbf{e}^{(0)} = \mathbf{u} - \mathbf{v}^{(m)}, \quad \mathbf{e}^{(\nu)} = \mathbf{u} - \mathbf{v}_{\nu}^{(m)}$$

It follows that

$$\mathbf{v}_{\nu}^{(m)} = \mathbf{u} - S_{sm}^{\nu}(\mathbf{u} - \mathbf{v}^{(m)})$$

By substitution, we obtain

$$\mathbf{r} = \mathbf{f} - A^h\mathbf{v}_{\nu}^{(m)} = \mathbf{f} - A^h\mathbf{u} + A^h S_{sm}^{\nu}(\mathbf{u} - \mathbf{v}^{(m)}) = A^h S_{sm}^{\nu}(\mathbf{u} - \mathbf{v}^{(m)}).$$

Iteration matrix

- We develop the matrix starting from the update step

- $\boldsymbol{v}^{(m+1)} = \boldsymbol{v}_{\nu}^{(m)} + I_{2h}^{h}(e^{2h})$

$= \boldsymbol{u} - S_{sm}^{\nu}\left(\boldsymbol{u} - \boldsymbol{v}^{(m)}\right) + I_{2h}^{h}\left(A^{2h}\right)^{-1} I_{h}^{2h}\boldsymbol{r}$

$= S_{sm}^{\nu}\boldsymbol{v}^{(m)} + (I - S_{sm}^{\nu})\left(A^{h}\right)^{-1}\boldsymbol{f}$

$+ I_{2h}^{h}\left(A^{2h}\right)^{-1} I_{h}^{2h} A^{h} S_{sm}^{\nu}(\left(A^{h}\right)^{-1}\boldsymbol{f} - \boldsymbol{v}^{(m)})$

$= (I - I_{2h}^{h}\left(A^{2h}\right)^{-1} I_{h}^{2h} A^{h}) S_{sm}^{\nu}\boldsymbol{v}^{(m)}$

$+((I - S_{sm}^{\nu}) + I_{2h}^{h}\left(A^{2h}\right)^{-1} I_{h}^{2h} A^{h} S_{sm}^{\nu})\left(A^{h}\right)^{-1}\boldsymbol{f}$

Def. $v_{\nu}^{(m)}$ on previous slide and coarse grid eqn $A^{2h}\boldsymbol{e}^{2h} = I_{h}^{2h}\boldsymbol{r}$

$A^{h}\mathbf{u} = \boldsymbol{f}$

Def. $\boldsymbol{r}$ on previous slide

Reordering.

Iteration matrix

- The two-level iteration matrix is thus given by

$$S_{2lev} = (I - I_{2h}^{h}(A^{2h})^{-1} I_{h}^{2h} A^{h}) S_{s}^{\nu}$$

- To prove convergence, one could do a rigorous Fourier analysis (see Trottenberg, Osterley et al., Multigrid).
- We will follow an approach of Hackbusch (Multi-grid methods and applications) and show mesh-independent convergence.

Goal of the convergence analysis

- From yesterday, we know that convergence of the two level method is given, if and only if

$$\rho(S_{2lev}) < 1$$

- This is in general hard to prove.

- However, for some induced matrix norm

$$\rho(S_{2lev}) \leq \|S_{2lev}\|$$

- We thus aim to show

$$\|S_{2lev}\| \leq \rho < 1$$

Convergence analysis

- The analysis is based on a splitting of S_{2lev} as

$$S_{2lev} = \left(\left(A^h\right)^{-1} - I_{2h}^h \left(A^{2h}\right)^{-1} I_h^{2h} \right) A^h S_{sm}^{\nu}$$

- Hence,

$$\|S_{2lev}\| \leq \|\left(A^h\right)^{-1} - I_{2h}^h \left(A^{2h}\right)^{-1} I_h^{2h}\| \cdot \|A^h S_{sm}^{\nu}\|$$

Approximation property Smoothing property

- Approximation property: This is the effect of the coarse grid approximation.
- Smoothing property: The efficiency of the smoothing step.

Smoothing and approximation properties

Definition 1: The matrix S_{sm} is said to possess the smoothing property, if there exist functions $\eta(\nu)$ and $\bar{\nu}(t)$ independent of h, such that

$$\left\| A^h S_{sm}^{\nu} \right\| \leq \eta(\nu) h^{-\alpha}$$

for some number $\alpha > 0$ and for all $1 \leq \nu \leq \bar{\nu}(h)$, with $\eta(\nu) \to 0$ as $\nu \to \infty$

Definition 2: The approximation property holds if there is a constant C_α, which is independent of h, such that

$$\left\| \left(A^h\right)^{-1} - I_{2h}^{h} \left(A^{2h}\right)^{-1} I_{h}^{2h} \right\| \leq C_\alpha h^\alpha$$

with the same α as in the smoothing property.

Convergence of the two-level method

Theorem

Suppose that the smoothing and the approximation property hold. Let $\rho > 0$ be a fixed number. Then there exists a number ν^*, such that

$$\|S_{2lev}\| \leq C_\alpha \eta(\nu) \leq \rho,$$

whenever $\nu \geq \nu^*$.

Proof

Since we required that α is the same for the approximation as well as the smoothing property, we get

$$\|S_{2lev}\| \leq C_\alpha h^\alpha \eta(\nu) h^{-\alpha} = C_\alpha \eta(\nu)$$

Since $\eta(\nu) \to 0$ as $\nu \to \infty$, the right hand side can be smaller than any given ρ, namely as ν is sufficiently large.

Convergence of the two-level method

- Note that the constant $C_\alpha \eta(\nu)$ is independant of the mesh size *h*
- It converges, if sufficiently many smoothing steps are applied
- In practice only a few pre-smoothing steps (1 to 3) are often sufficient

- The estimate above is however often not very tight.

V(2,1)-cycle for 2D Poisson

	$n = 16$				$n = 32$				$n = 64$				$n = 128$			
V-cycle	$\|\mathbf{r}^h\|_h$	ratio	$\|\mathbf{e}\|_h$	ratio	$\|\mathbf{r}^h\|_h$	ratio	$\|\mathbf{e}\|_h$	ratio	$\|\mathbf{r}^h\|_h$	ratio	$\|\mathbf{e}\|_h$	ratio	$\|\mathbf{r}^h\|_h$	ratio	$\|\mathbf{e}\|_h$	ratio
0	6.75e+02		5.45e−01		2.60e+03		5.61e−01		1.06e+04		5.72e−01		4.16e+04		5.74e−01	
1	4.01e+00	0.01	1.05e−02	0.02	1.97e+01	0.01	1.38e−02	0.02	7.56e+01	0.01	1.39e−02	0.02	2.97e+02	0.01	1.39e−02	0.02
2	1.11e−01	0.03	4.10e−04	0.04	5.32e−01	0.03	6.32e−04	0.05	2.07e+00	0.03	6.87e−04	0.05	8.25e+00	0.03	6.92e−04	0.05
3	3.96e−03	0.04	1.05e−04	0.26	2.06e−02	0.04	4.41e−05	0.07	8.30e−02	0.04	4.21e−05	0.06	3.37e−01	0.04	4.22e−05	0.06
4	1.63e−04	0.04	1.03e−04	0.98*	9.79e−04	0.05	2.59e−05	0.59	4.10e−03	0.05	7.05e−06	0.17	1.65e−02	0.05	3.28e−06	0.08
5	7.45e−06	0.05	1.03e−04	1.00*	5.20e−05	0.05	2.58e−05	1.00*	2.29e−04	0.06	6.45e−06	0.91*	8.99e−04	0.05	1.63e−06	0.50
6	3.75e−07	0.05	1.03e−04	1.00*	2.96e−06	0.06	2.58e−05	1.00*	1.39e−05	0.06	6.44e−06	1.00*	5.29e−05	0.06	1.61e−06	0.99*
7	2.08e−08	0.06	1.03e−04	1.00*	1.77e−07	0.06	2.58e−05	1.00*	8.92e−07	0.06	6.44e−06	1.00*	3.29e−06	0.06	1.61e−06	1.00*
8	1.24e−09	0.06	1.03e−04	1.00*	1.10e−08	0.06	2.58e−05	1.00*	5.97e−08	0.07	6.44e−06	1.00*	2.14e−07	0.06	1.61e−06	1.00*
9	7.74e−11	0.06	1.03e−04	1.00*	7.16e−10	0.06	2.58e−05	1.00*	4.10e−09	0.07	6.44e−06	1.00*	1.43e−08	0.07	1.61e−06	1.00*
10	4.99e−12	0.06	1.03e−04	1.00*	4.79e−11	0.07	2.58e−05	1.00*	2.87e−10	0.07	6.44e−06	1.00*	9.82e−10	0.07	1.61e−06	1.00*
11	3.27e−13	0.07	1.03e−04	1.00*	3.29e−12	0.07	2.58e−05	1.00*	2.04e−11	0.07	6.44e−06	1.00*	6.84e−11	0.07	1.61e−06	1.00*
12	2.18e−14	0.07	1.03e−04	1.00*	2.31e−13	0.07	2.58e−05	1.00*	1.46e−12	0.07	6.44e−06	1.00*	4.83e−12	0.07	1.61e−06	1.00*
13	2.33e−15	0.11	1.03e−04	1.00*	1.80e−14	0.08	2.58e−05	1.00*	1.08e−13	0.07	6.44e−06	1.00*	3.64e−13	0.08	1.61e−06	1.00*
14	1.04e−15	0.45	1.03e−04	1.00*	6.47e−15	0.36	2.58e−05	1.00*	2.60e−14	0.24	6.44e−06	1.00*	1.03e−13	0.28	1.61e−06	1.00*
15	6.61e−16	0.63	1.03e−04	1.00*	5.11e−15	0.79	2.58e−05	1.00*	2.30e−14	0.88	6.44e−06	1.00*	9.19e−14	0.89	1.61e−06	1.00*

- 2D Poisson on unit square and homogeneous Dirichlet conditions
- Results taken from Briggs et al, A multigrid tutorial, Table 4.1