

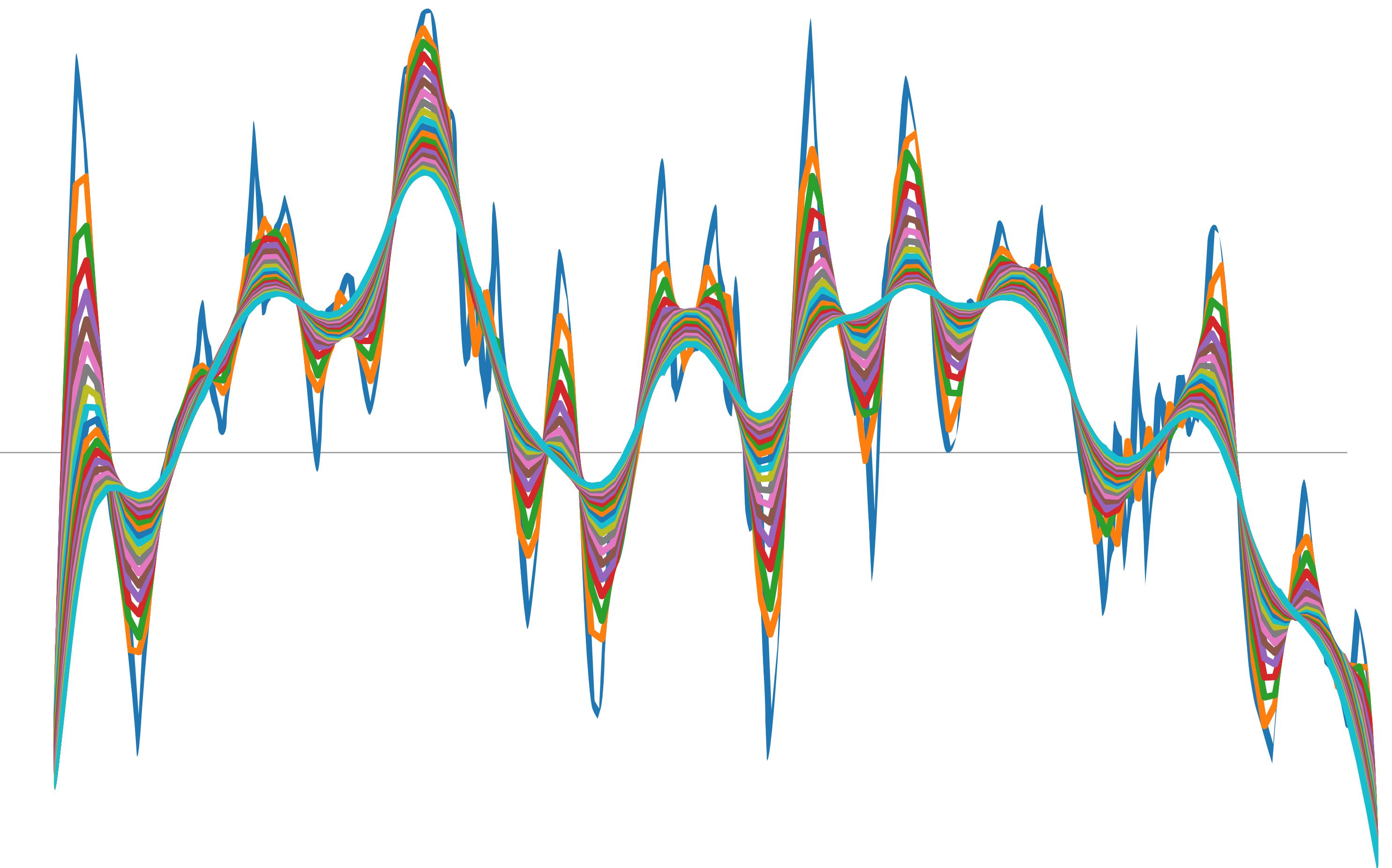
Multigrid Methods

The Basics

2023 Copper Multigrid Conference
April 16, 2023

Hans De Sterck
Department of Applied Mathematics,
University of Waterloo

(slides by Luke Olson, Department of Computer Science, University of Illinois at Urbana-Champaign)



What are we trying to do...

- Solve problems of the form

$$Ax = b$$

- Solve this problem **iteratively**:

$$x_1 \leftarrow x_0 + v$$

- Solve this problem **inexpensively**:

- The update should be “good”
- Finding the update should be “cheap”

Objectives – high level

- Construct a multigrid method for **your** problem
- Interpret the effectiveness of a multigrid method
- Identify *why* a method works — or — *why* a method does not
- Recognize different forms of multigrid, their pitfalls and their uses

Objectives – this session

- **Create** a two level multigrid method
- **Illustrate** the main components of a multigrid method
- **Calculate** the effectiveness of a multigrid method
- **Highlight** some limitations of multigrid
- **Outline** some approaches to fixing multigrid (and why)
- **Identify** key pieces in moving to unstructured problems

A cautionary example

Guess and look for
and update

$$x_1 = x_0 + \text{update}$$

Updating with the
error would be **ideal**

Or in another form

$$x_1 = x_0 + e_0$$

$$x_1 = x_0 + A^{-1}r_0$$

Not practical so...

$$x_1 = x_0 + D^{-1}r_0$$

1 iteration

x^* solution to $Ax = b$

$e_0 = x^* - x_0$ error

$r_0 = b - Ax_0$ residual
 $= Ae_0$

$Ae_0 = r_0$ error equation

Jacobi ~ 1 SpMV

A reminder – projection methods

- Take a guess

$$x_0$$

- Look for an update that is the “best”:

$$x_1 \leftarrow x_0 + u$$

- Minimize over a smaller space

$$\min_{u \in \text{span}\{V\}} \|x^* - x_1\|$$

- Then $u = Vy$

$$V^T V y = V^T e_0$$

- So the update looks like

$$x_1 = x_0 + V(V^T V)^{-1} V^T e_0$$

A reminder – projection methods

- Instead, look at the A-norm (A s.p.d.):

$$\min_{u \in \text{span}\{V\}} \|x^* - x_1\|_A$$

- Then $u = V\bar{y}$

$$V^T A V \bar{y} = V^T A e_0$$

$$V^T A V \bar{y} = V^T r_0$$

- So that

$$x_1 = x_0 + V(V^T A V)^{-1} V^T r_0$$

A reminder – projection methods

$$x_1 = x_0 + u$$

$$x_1 = x_0 + V(V^T A V)^{-1} V^T r_0$$

- What about the error

$$x^* - x_1 = x^* - x_0 - V(V^T A V)^{-1} V^T r_0$$

$$e_1 = e_0 - V(V^T A V)^{-1} V^T A e_0$$

$$= \left(I - V(V^T A V)^{-1} V^T A \right) e_0$$

A-orthogonal
projection onto the
range of V

Model Problem

- A model problem

$$\begin{aligned}-u_{xx} &= f \\ u(0) &= u(1) = 0\end{aligned}$$

- Finite differences

$$\frac{-u_{i-1} + 2u_i - u_{i+1}}{h^2} = f_i \quad i = 1, \dots, n \quad u_0 = u_{n+1} = 0$$

- A model matrix problem

$$A = \frac{1}{h^2} \begin{bmatrix} 2 & -1 & & \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & \ddots \end{bmatrix}$$

Model Problem

- A special matrix:

$$A = \begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & -1 & 2 & -1 & \\ & & \ddots & \ddots & \\ & & & -1 & 2 \end{bmatrix}$$

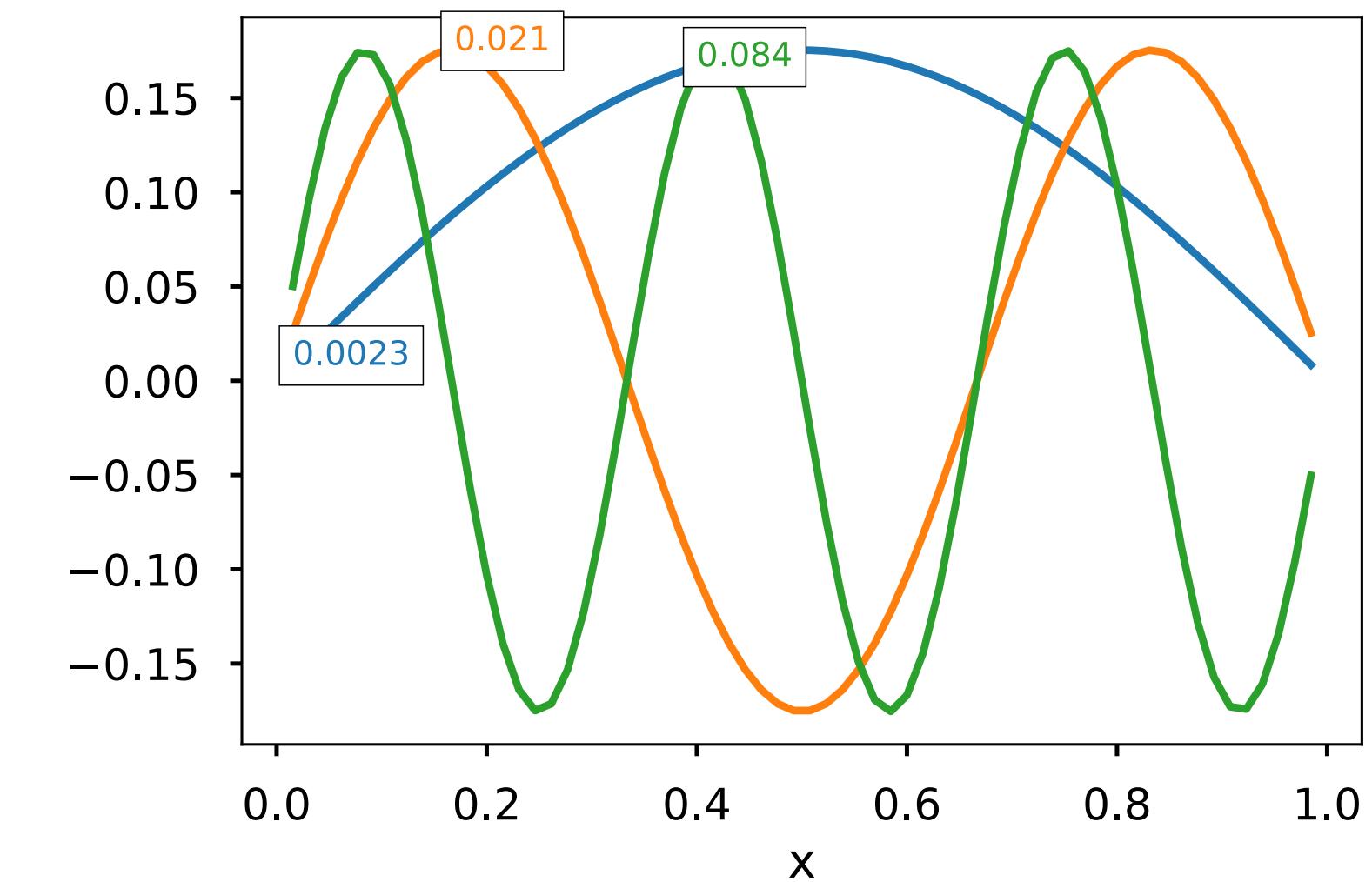
The eigenvalues

range from (0,4]

(or from $c h^2$ to ~ 4)

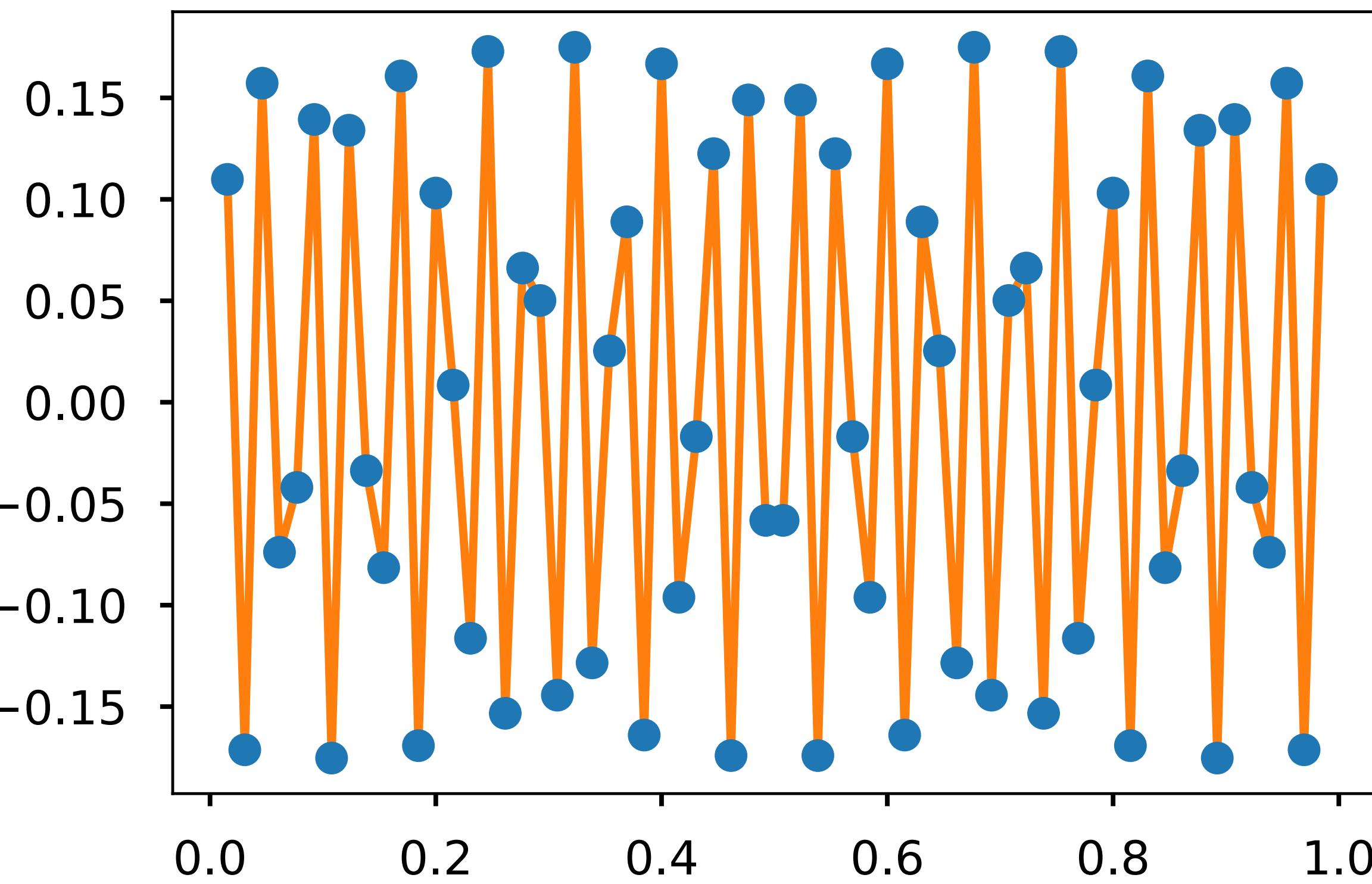
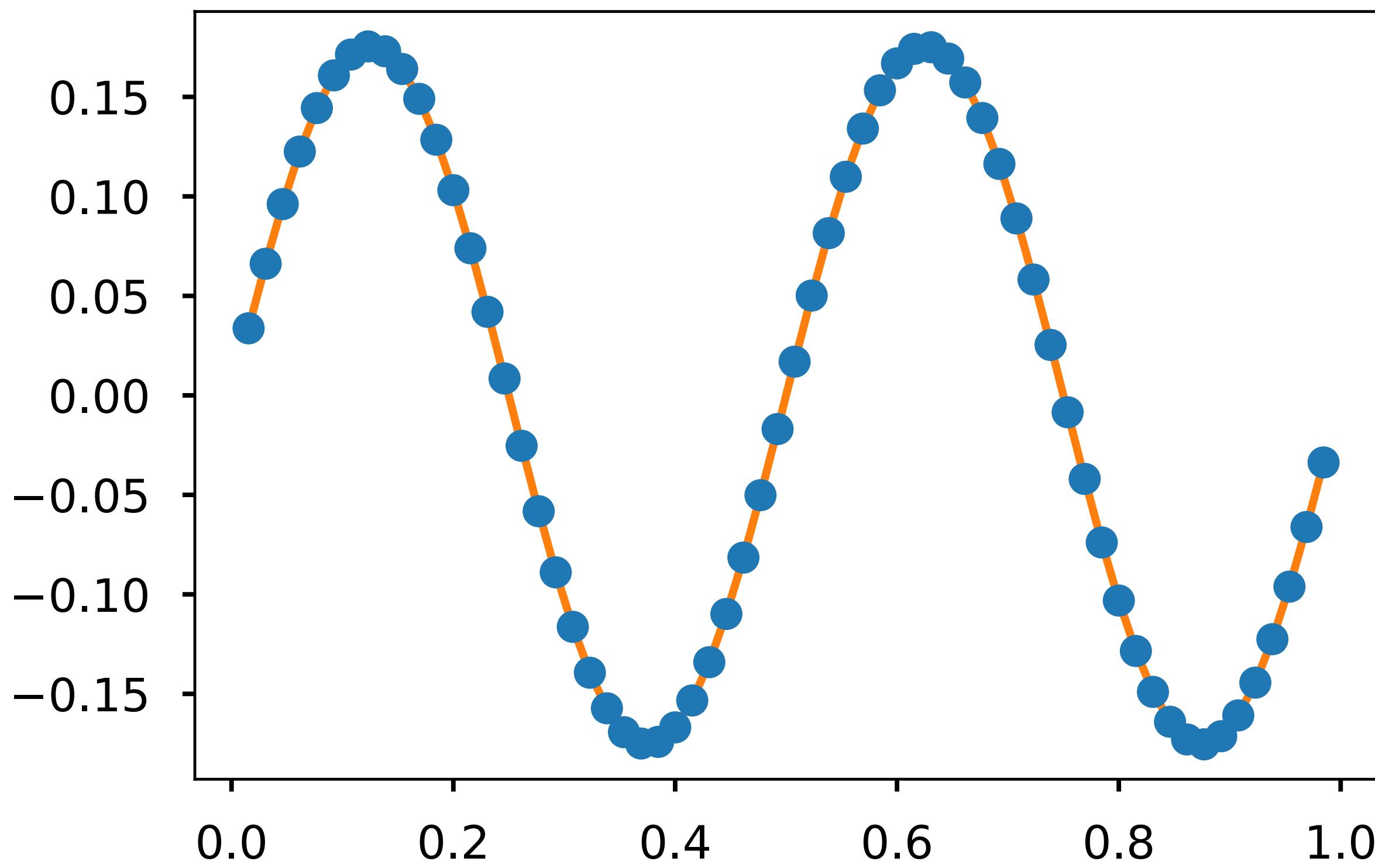
The eigenvectors are Fourier modes:

$$(v_k)_j = \sin\left(\frac{(j+1)*k\pi}{n+1}\right)$$



Smooth Mode ~ low Fourier Modes

- We will talk a lot about “smoothness” – how much variation there is **per** grid point.



First relaxation scheme: Jacobi

- The discretization at point i :

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

- Solving for the variable at this point (eliminating the residual):

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

- In matrix form:

$$u \leftarrow (I - D^{-1} A)u + h^2 D^{-1} f$$

$$u \leftarrow u + D^{-1} r$$

- And the error:

$$e \leftarrow Ge$$

$$G = I - D^{-1} A$$

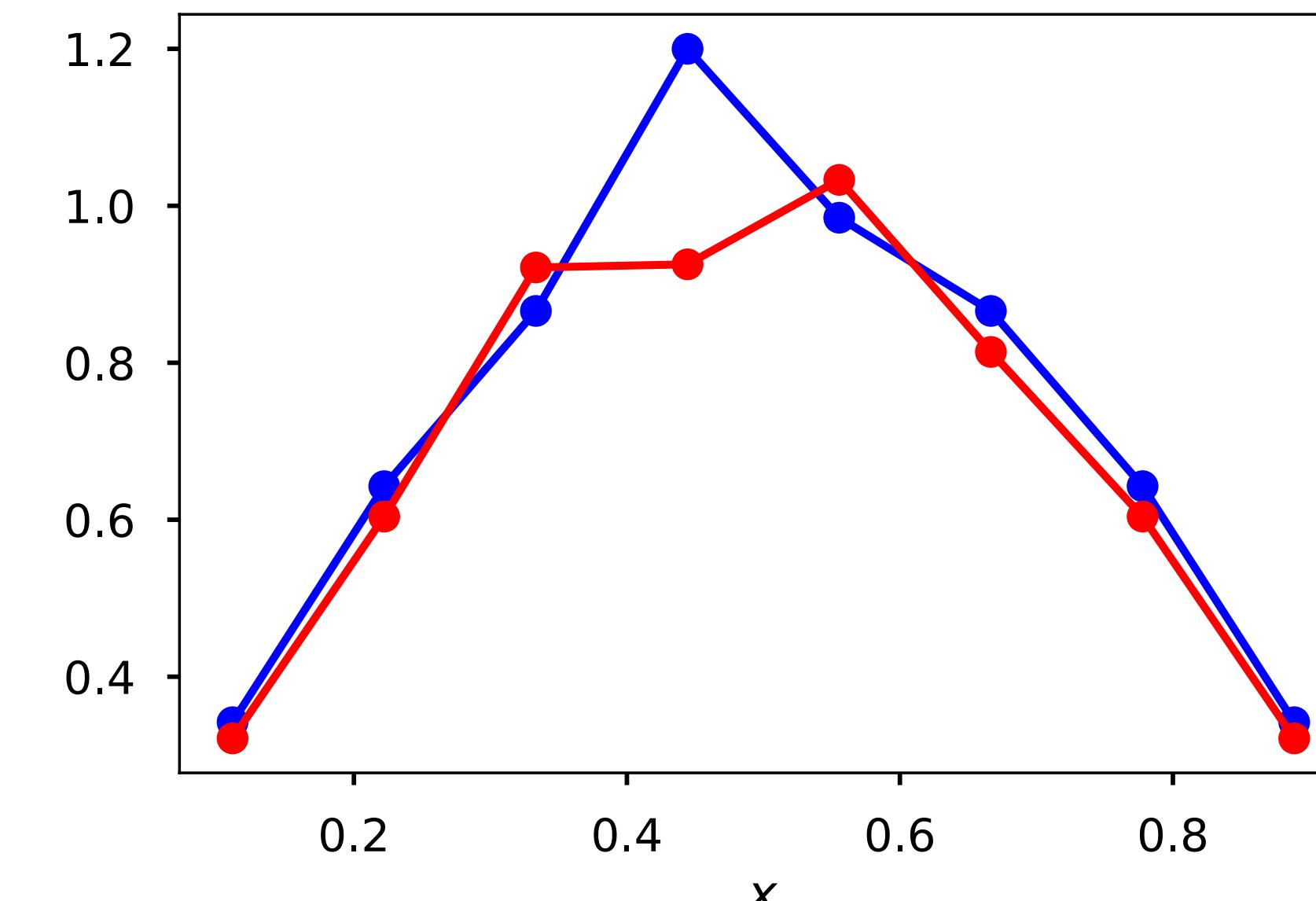
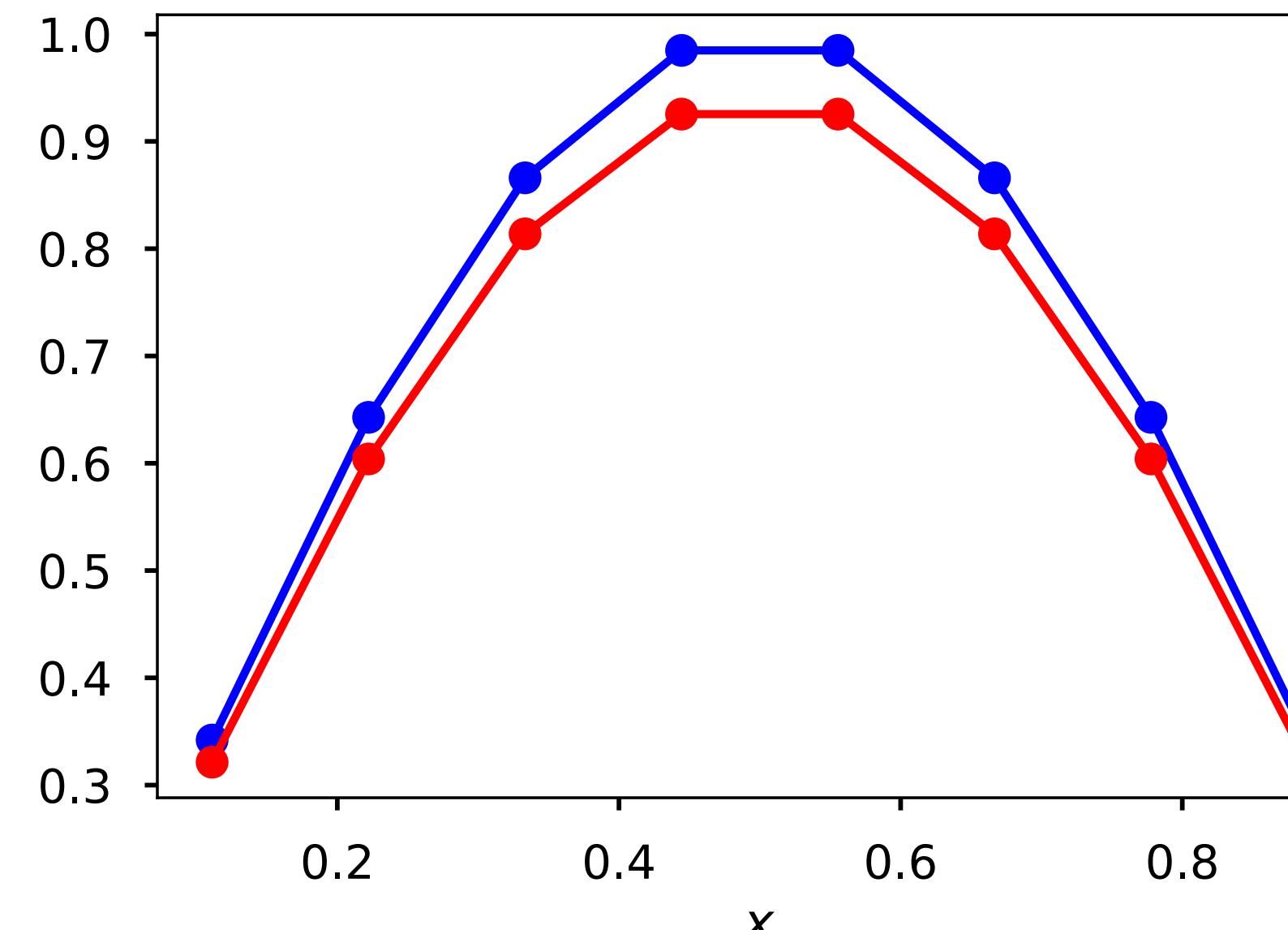
The “error propagation operator”

What does Jacobi do to error?

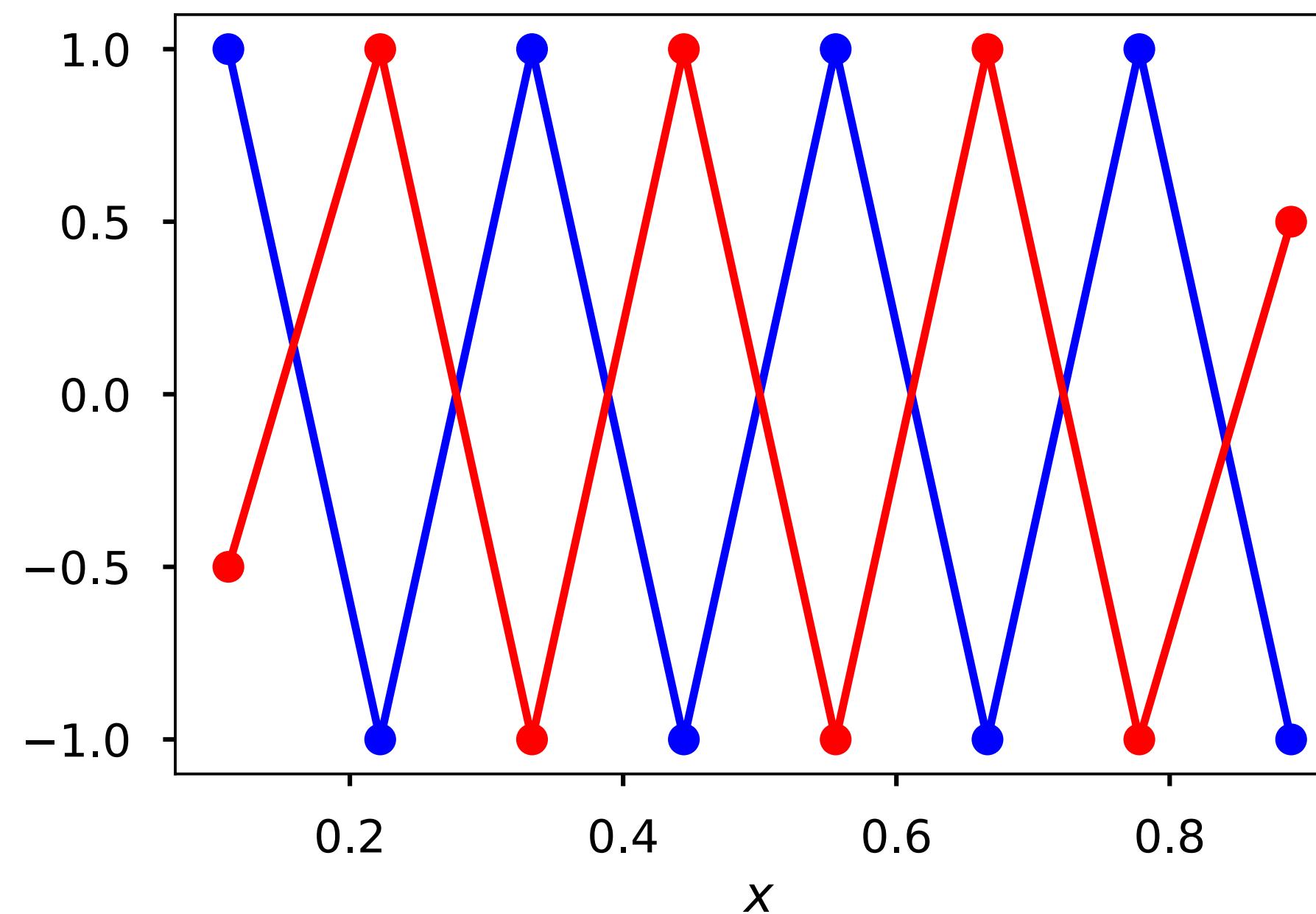
$$e^{new} \leftarrow (I - D^{-1}A)e^{old}$$

$$e_i^{new} \leftarrow \frac{1}{2} (e_{i-1}^{old} + e_{i+1}^{old})$$

- It averages...
- Consider smooth and oscillatory error:



But...

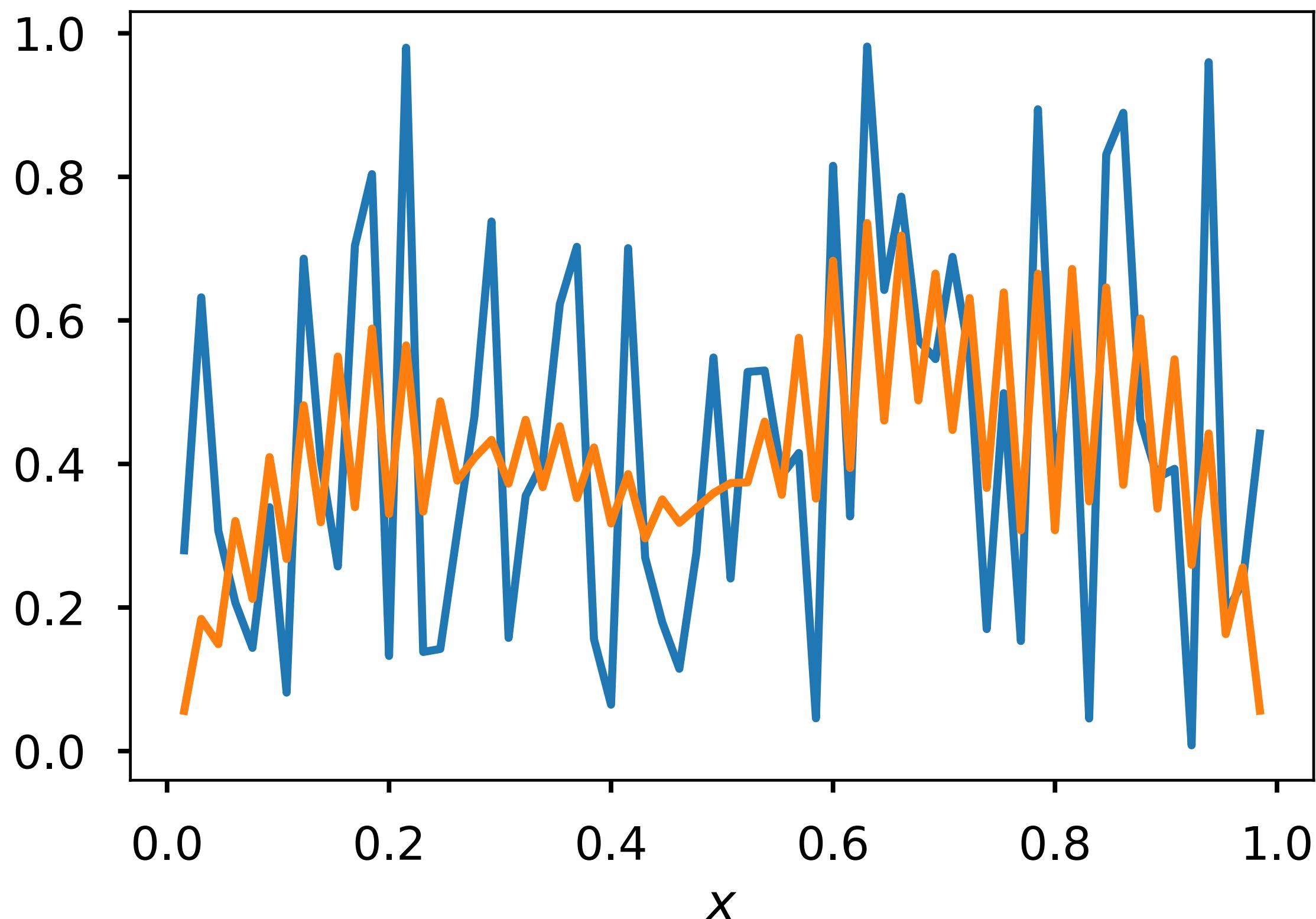


- Jacobi “averages out” certain error very quickly.
- But stagnates on very smooth error or very oscillatory error

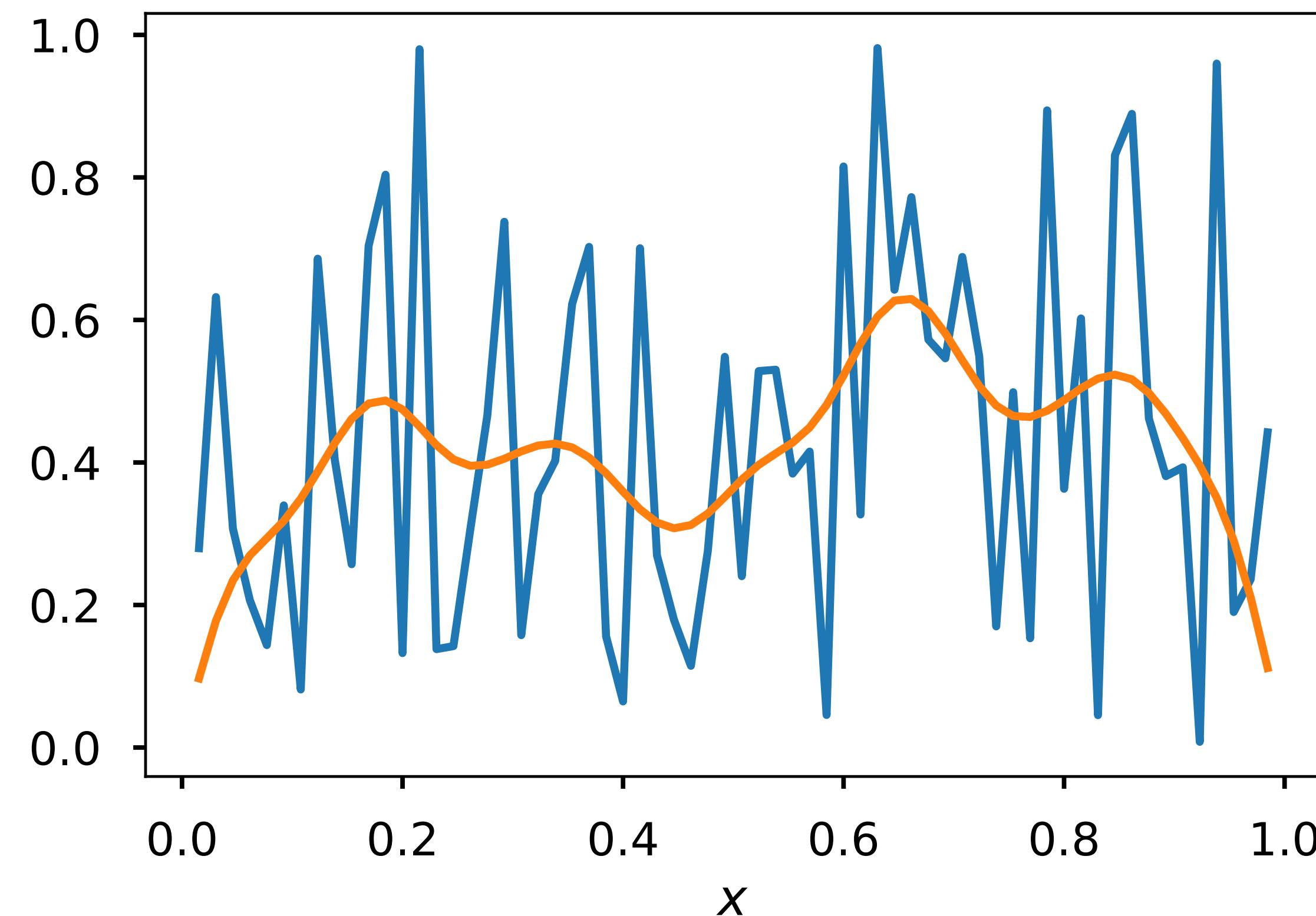
From Jacobi to weighted-Jacobi

- Random initial guess (random error):

$$u \leftarrow u + D^{-1}r$$



$$u \leftarrow u + (2/3)D^{-1}r$$



Weighted Jacobi

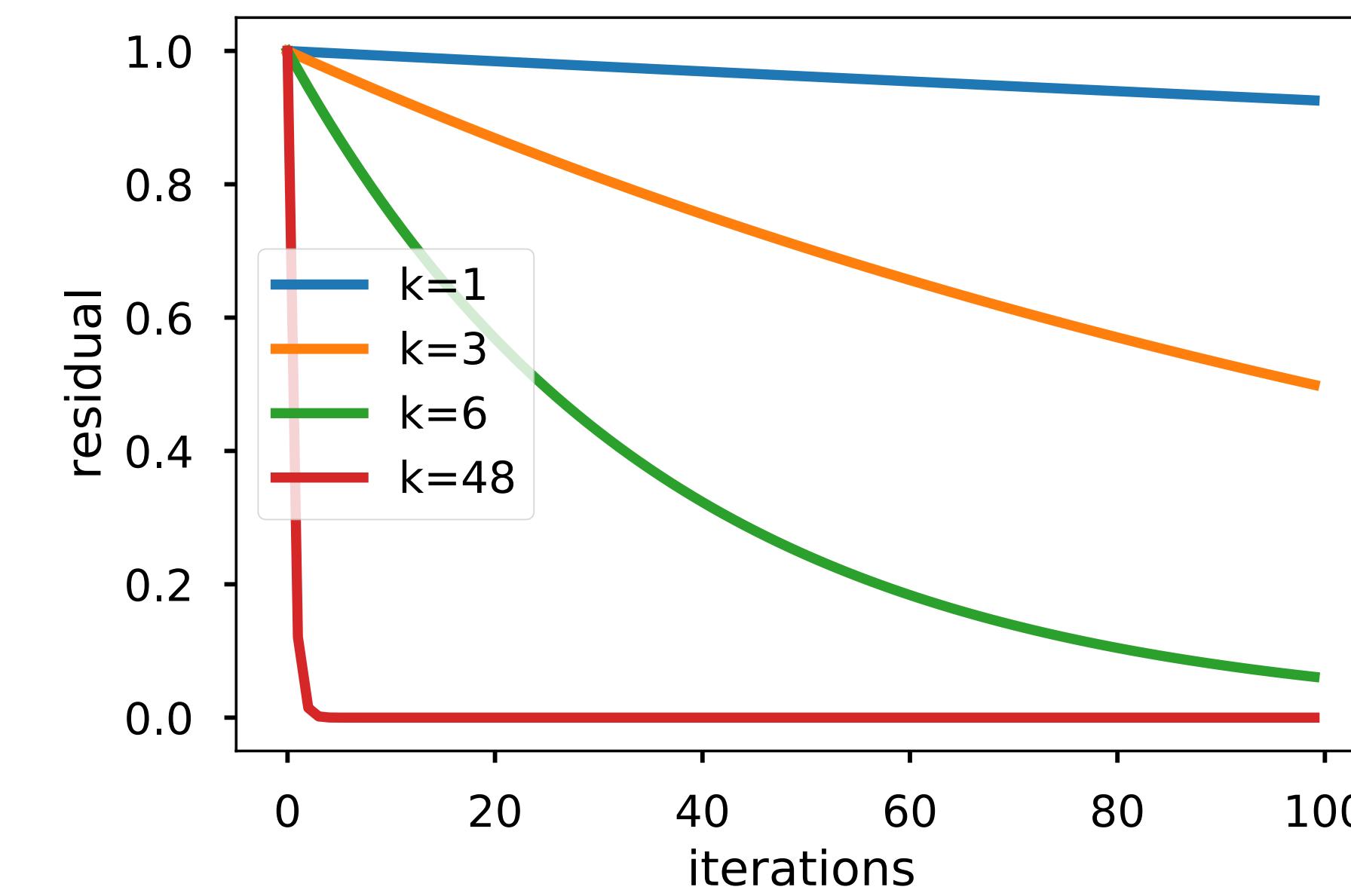
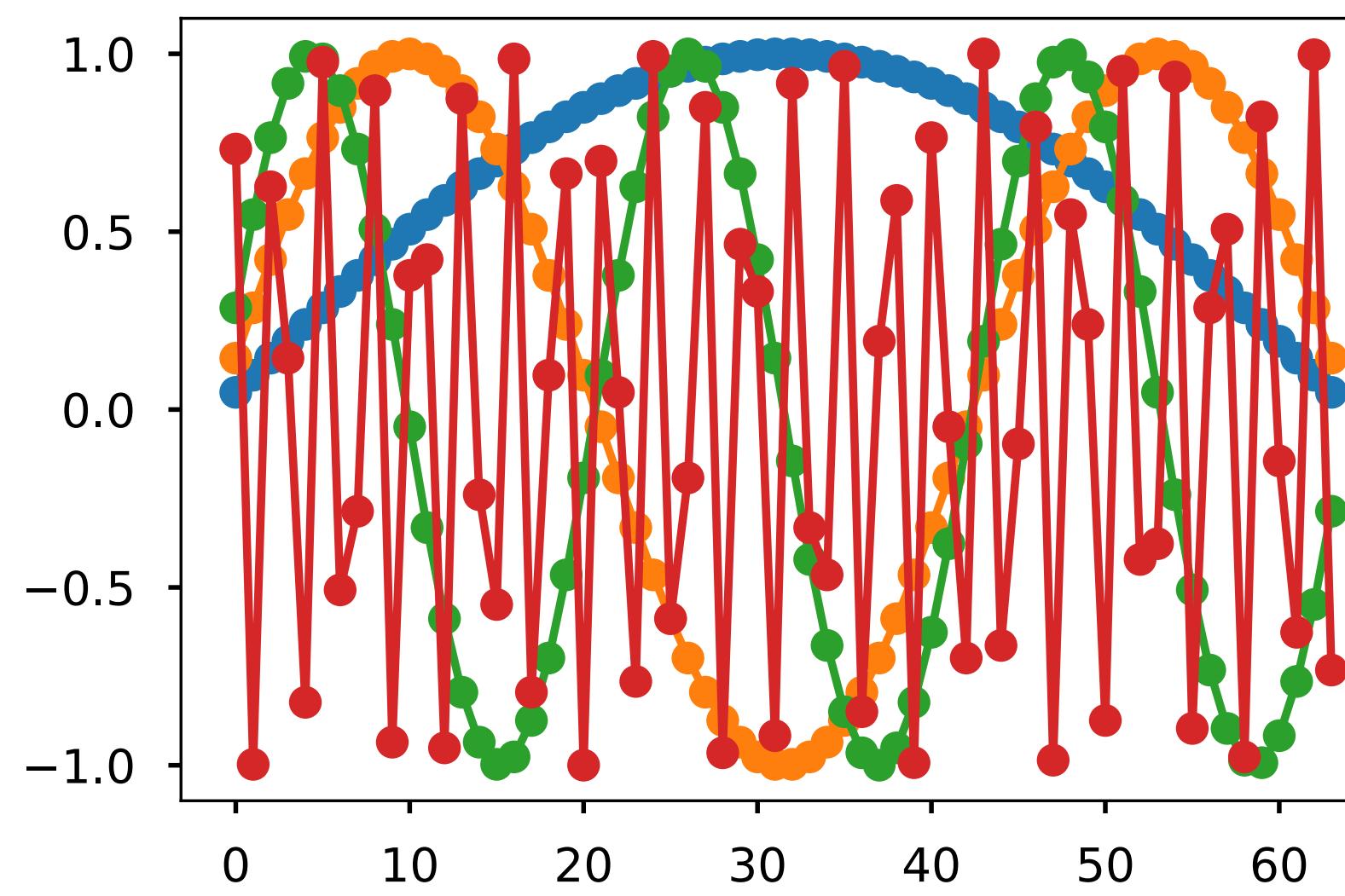
- Weighted Jacobi

$$u \leftarrow u + (2/3)D^{-1}r$$

- What does this do to other modes?
 - Are smooth modes damped slowly?
 - What about oscillatory modes?
- Why did we pick 2/3 – that seemed like a lucky guess?!

Weighted Jacobi

- If we picked 4 modes, 1, 3, 6, and 48 (out of 64)
- Then smooth modes still dampen less quickly than higher ones



Error Propagation

- Let's consider an initial error

$$e_0 = \sum_{k=1}^n c_k v_k$$

- And the weighted Jacobi iteration matrix

$$e \leftarrow (I - \omega D^{-1} A)e = Ge$$

- From ν iterations we have

$$\begin{aligned} G^\nu e_0 &= \sum_{k=1}^n c_k G^\nu v_k \\ &= \sum_{k=1}^n c_k \lambda_k^\nu v_k \end{aligned}$$

- As a result, mode **k** is reduced by the magnitude of λ_k in every pass

Fundamental Theorem of Iteration

$$G = I - M^{-1}A$$

- Convergent ($G^n \rightarrow 0$) if and only if $\rho(G) \leq 1$

$$\frac{\|e_n\|}{\|e_0\|} \leq \|G^n\| \leq \|G\|^n \approx 10^{-d}$$

- How many iterations do we need to guarantee the reduction of the error by d digits?

$$n \approx -\frac{d}{\log_{10} \rho(G)}$$

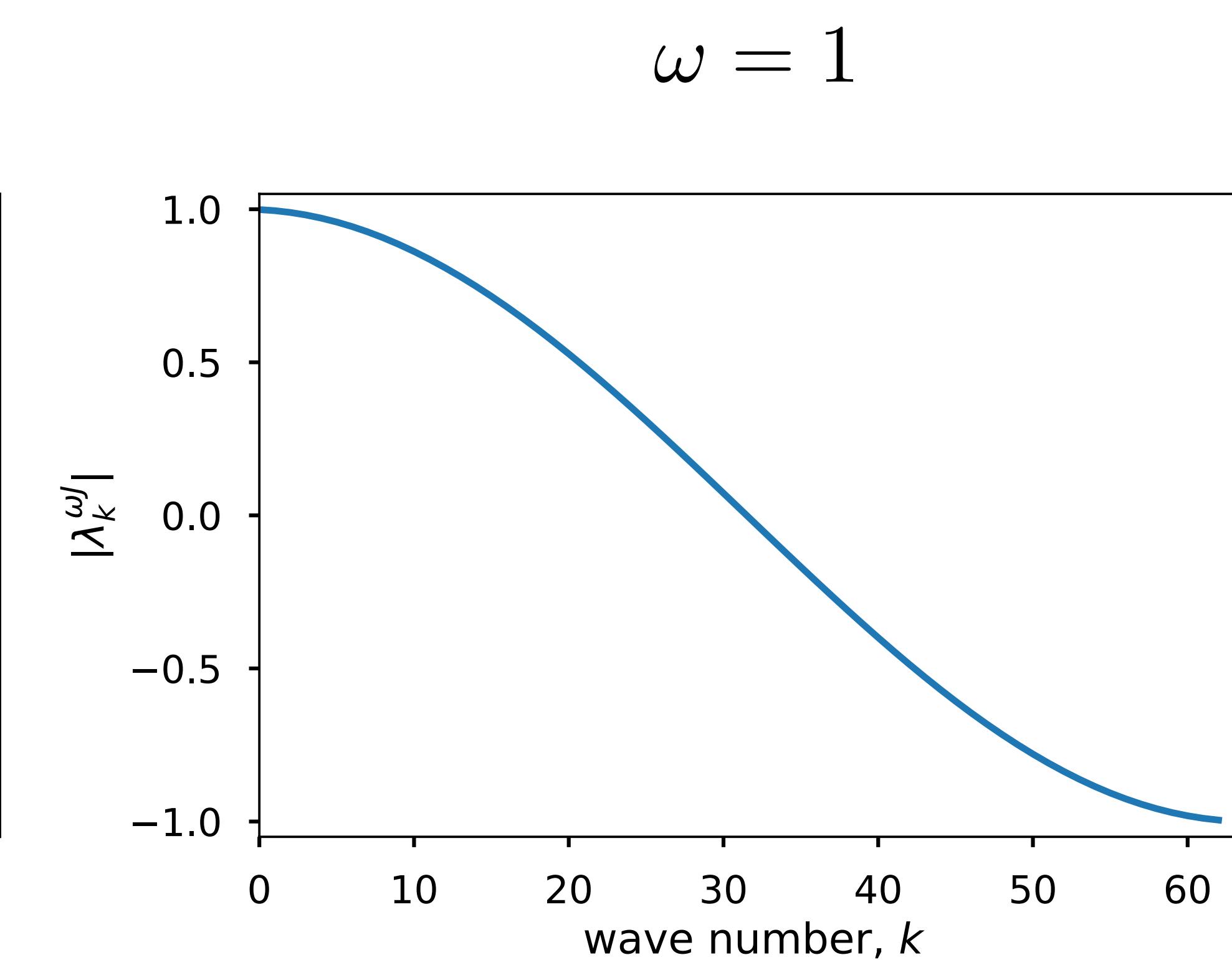
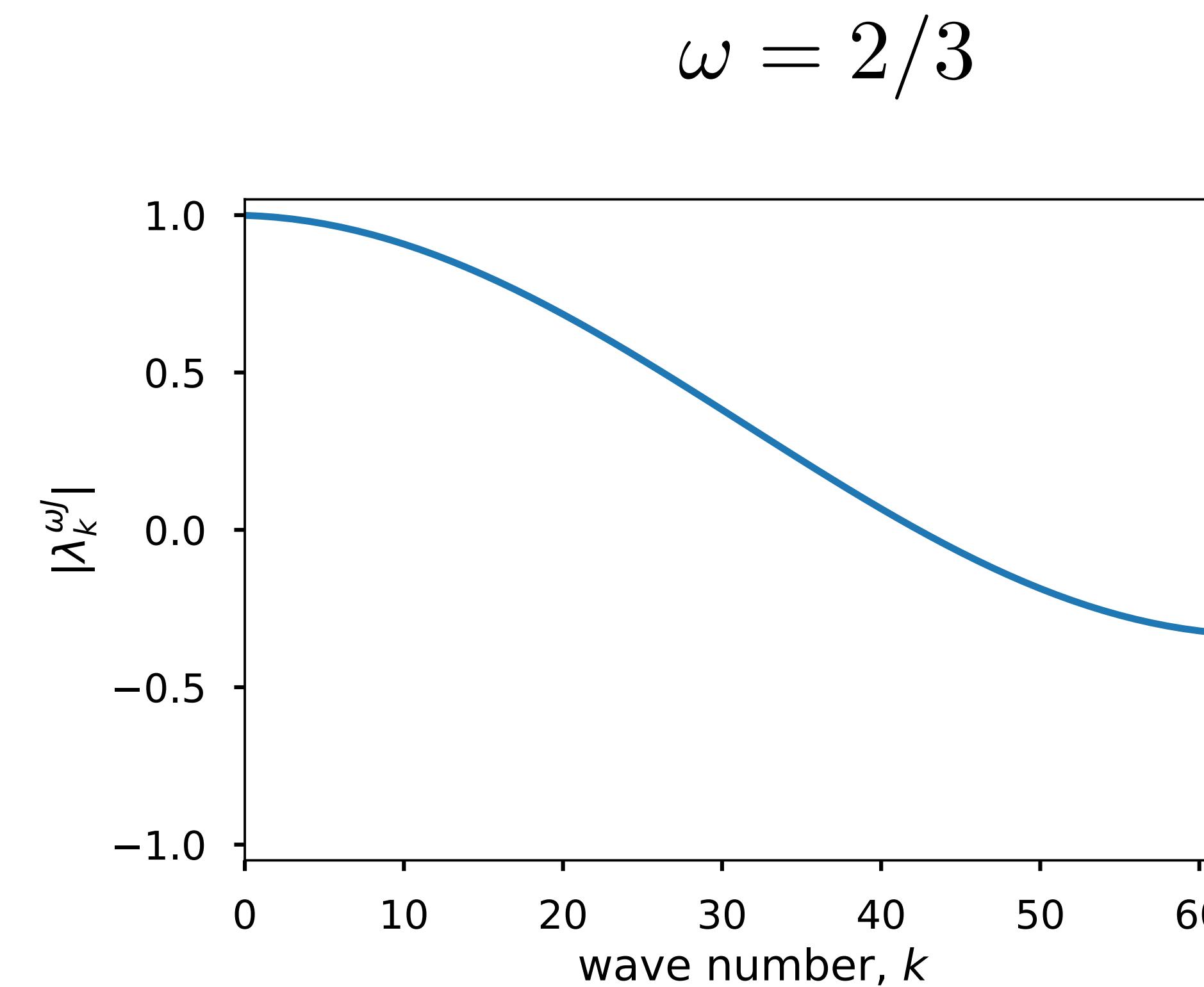
Relaxation

- Convergence **factor** $\|G\|$ or $\rho(G)$
- Convergence **rate** $-\log_{10} \rho(G)$
- For Jacobi:
$$G = I - D^{-1}A \longrightarrow \lambda_k = 1 - \frac{1}{2} \cdot 4 \cdot \sin^2 \left(\frac{k\pi}{2(n+1)} \right)$$
$$G = I - (2/3)D^{-1}A \longrightarrow \lambda_k = 1 - \frac{2}{3} \cdot \frac{1}{2} \cdot 4 \cdot \sin^2 \left(\frac{k\pi}{2(n+1)} \right)$$

The spectral radius is almost the same, but...

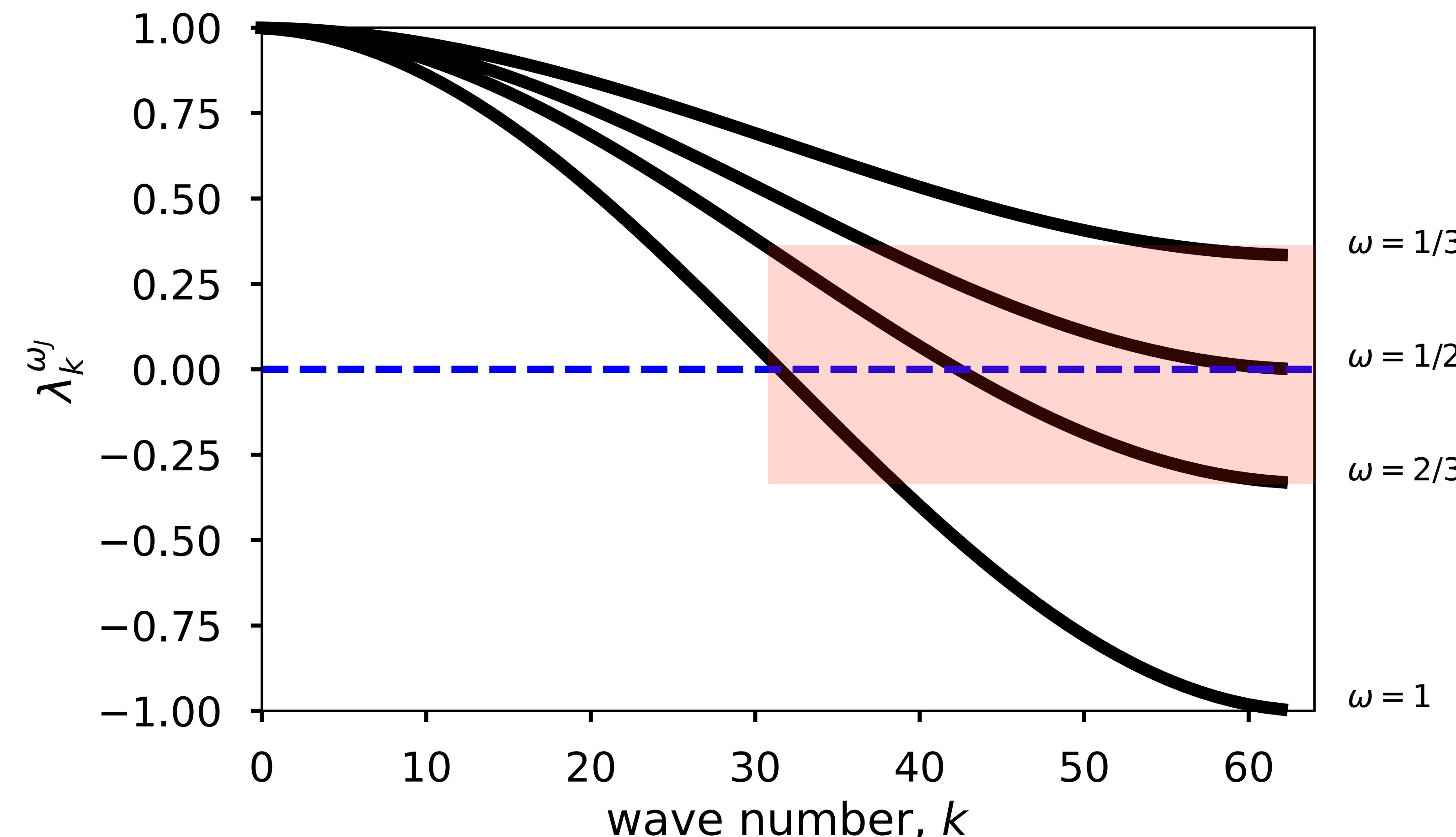
Weighted Jacobi

- If we look at the spectrum:
 - Weighted Jacobi dampens modes that are highly oscillatory



Weighted Jacobi

- Selecting 2/3 balances the reduction in error in the **high** modes



The multigrid smoothing factor

- The **smoothing factor** of relaxation method G is the maximum magnitude of the upper half of the spectrum:

$$\max_{k \in [n/2, n]} |\lambda_k^G|$$

- A desired feature for relaxation:

Oscillatory modes are quick to converge
Smooth modes are slow to converge

Multigrid Step #1: pick a smoother

- For $\omega = 2/3$

$$|\lambda_{n/2}| = |\lambda_n| = 1/3$$

- For $\omega = 1$

$$|\lambda_{n/2}| = |\lambda_n| = 1$$

- Jacobi is not a smoother (weighted *is*)

An important observation on “smoothness”

- So far, we've mainly looked at

$$Au = 0$$

- In general we need to consider

$$Au = f$$

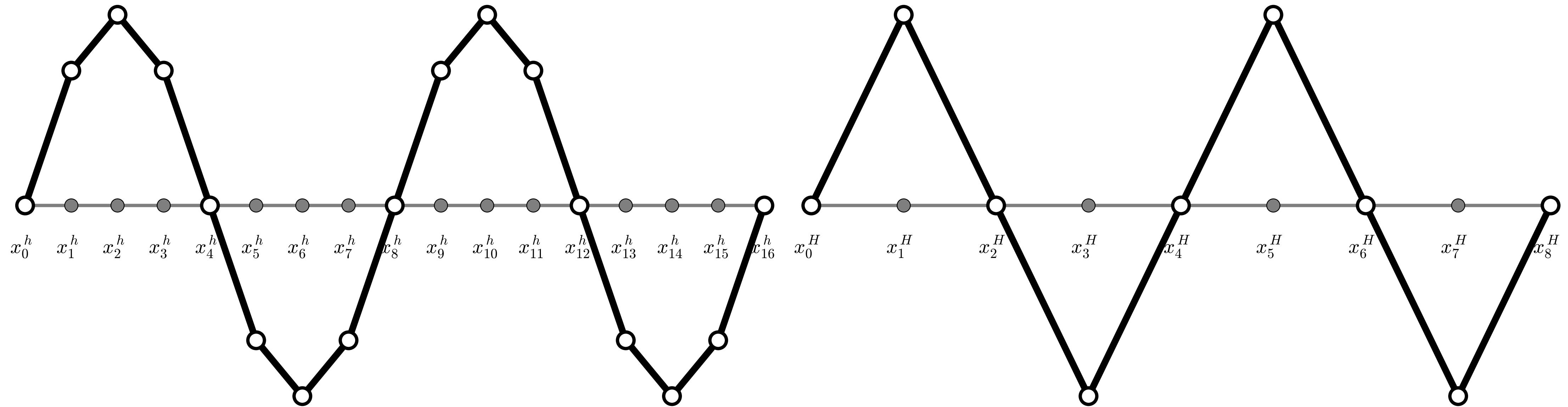
- If we **smooth** with

$$u \leftarrow u + \omega D^{-1}r$$

then the **error** is smooth, not (necessarily the solution).

Coarse Grids

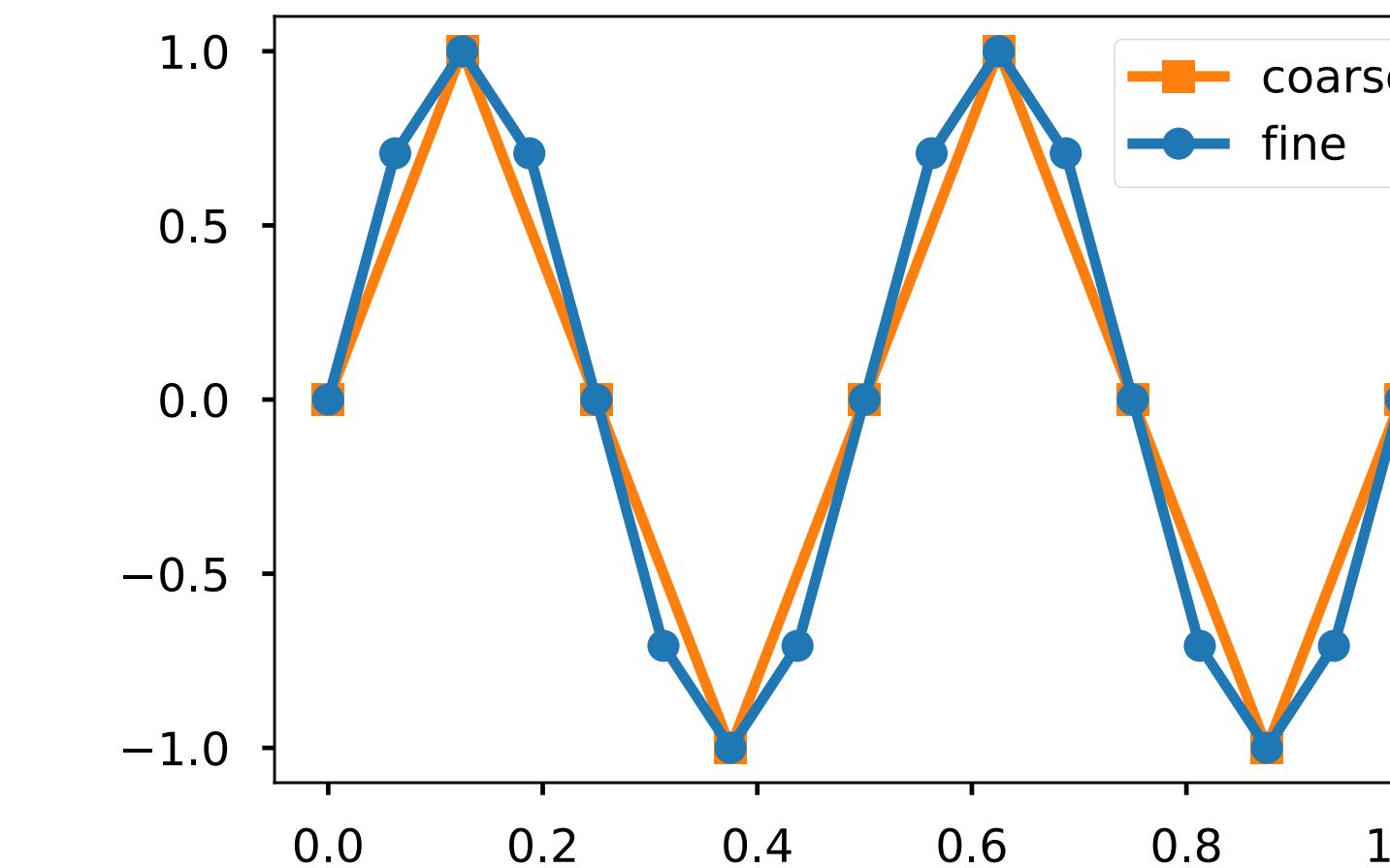
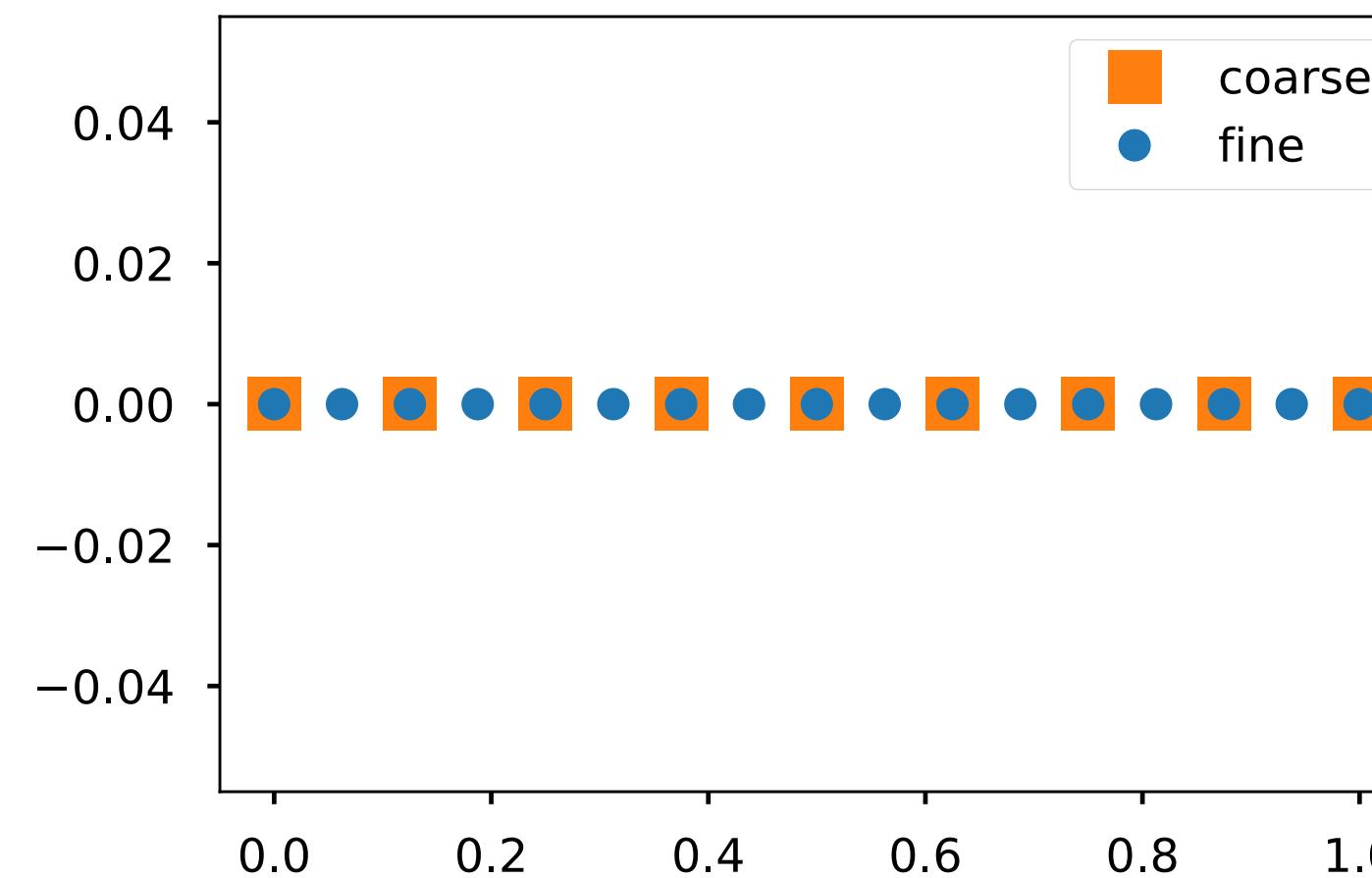
- Smooth modes look like oscillatory modes when sampled on a coarse grid
- 4-mode of 15 **versus** 4-mode of 7



This looks like a “smooth” mode

This looks like an “oscillatory” mode

Coarse modes



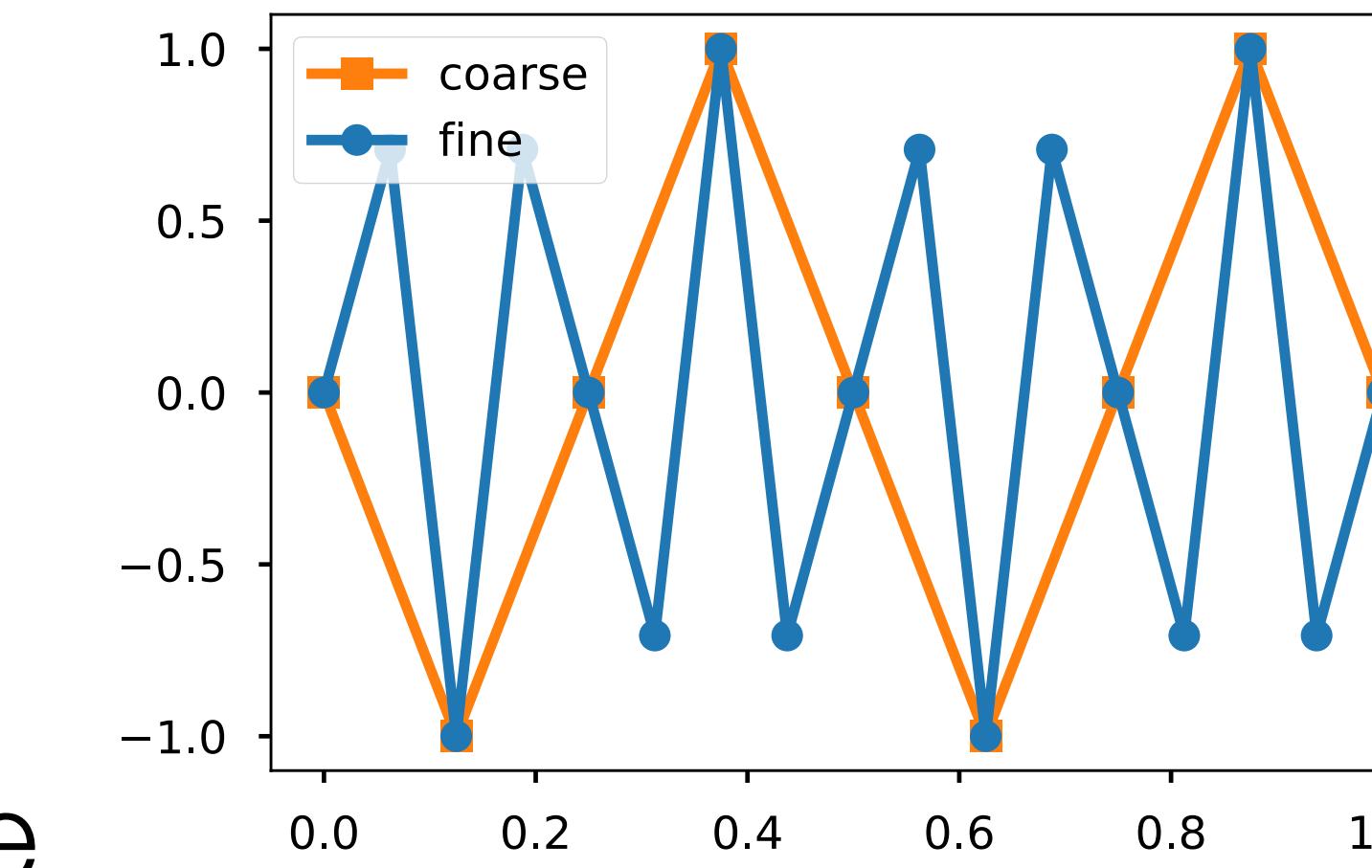
mode 4 of 15

$$\begin{aligned}(v_k)_j &= \sin \frac{jk\pi}{n+1} \\(v_k)_{2j} &= \sin \frac{2jk\pi}{n+1} \\&= \sin \frac{jk\pi}{(n+1)/2} \\&= (\hat{v}_k)_j\end{aligned}$$

Fine mode

Fine mode
(every other)

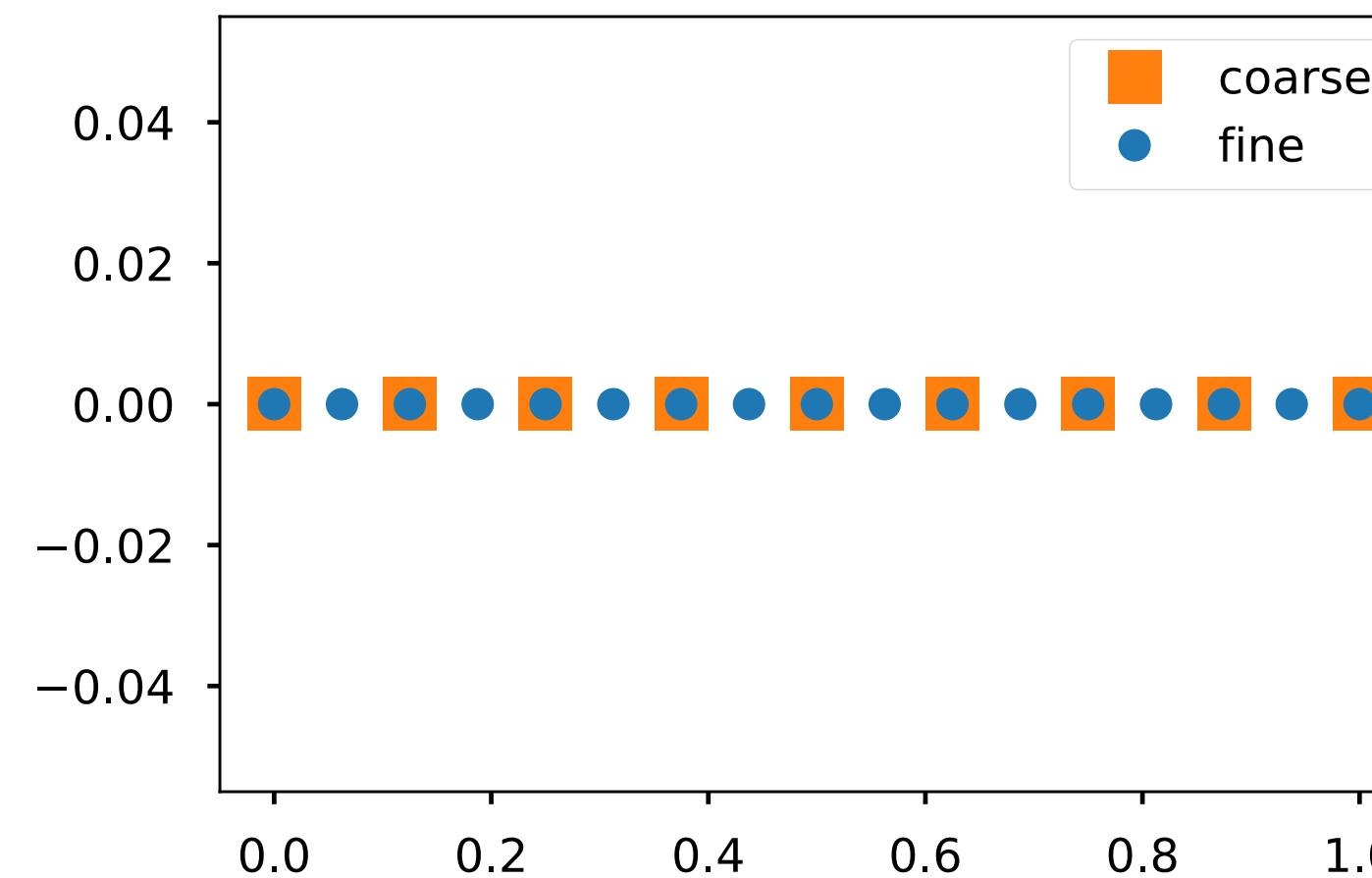
Coarse mode



mode 12 of 15

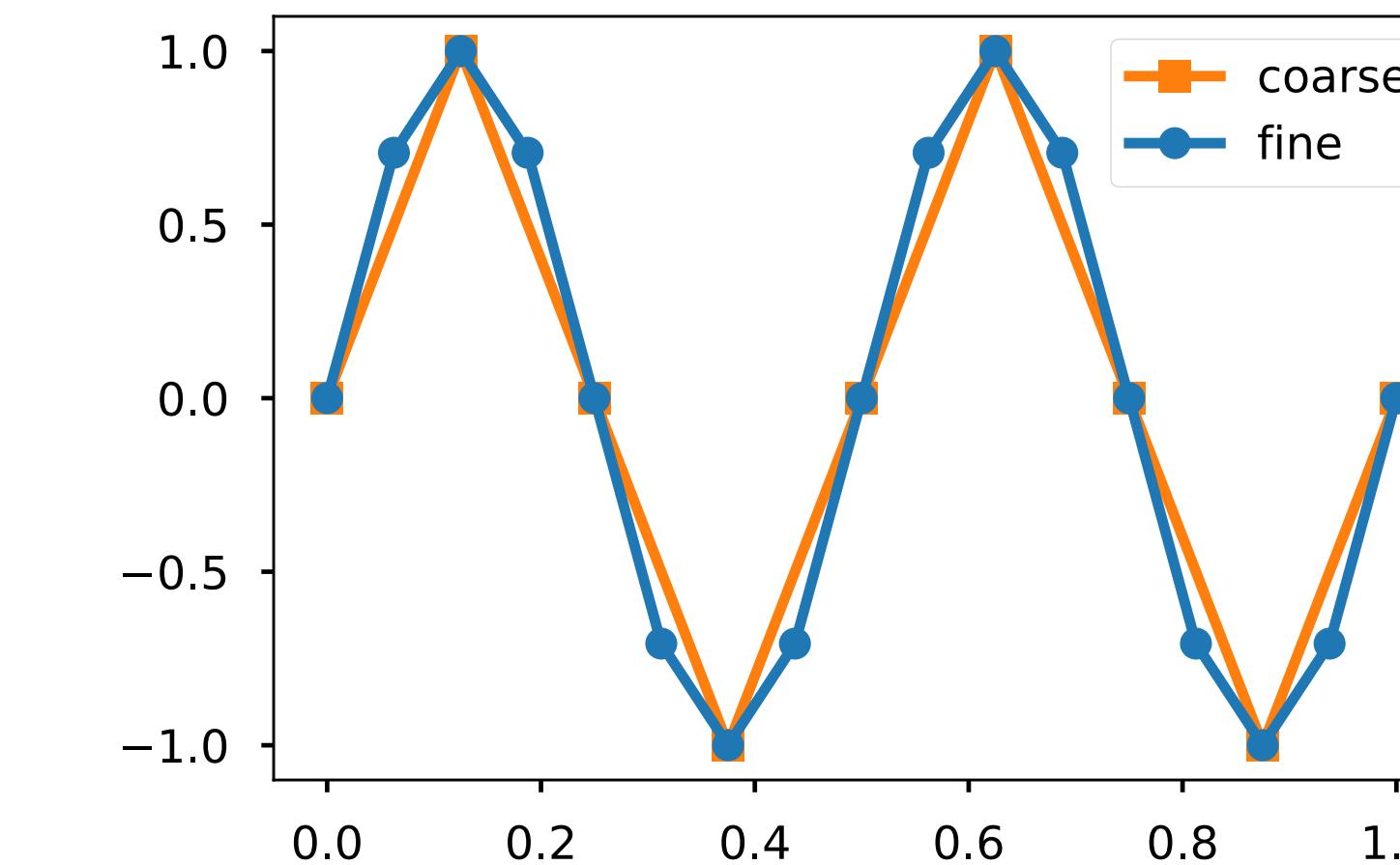
For low modes, k-modes are preserved

Coarse modes

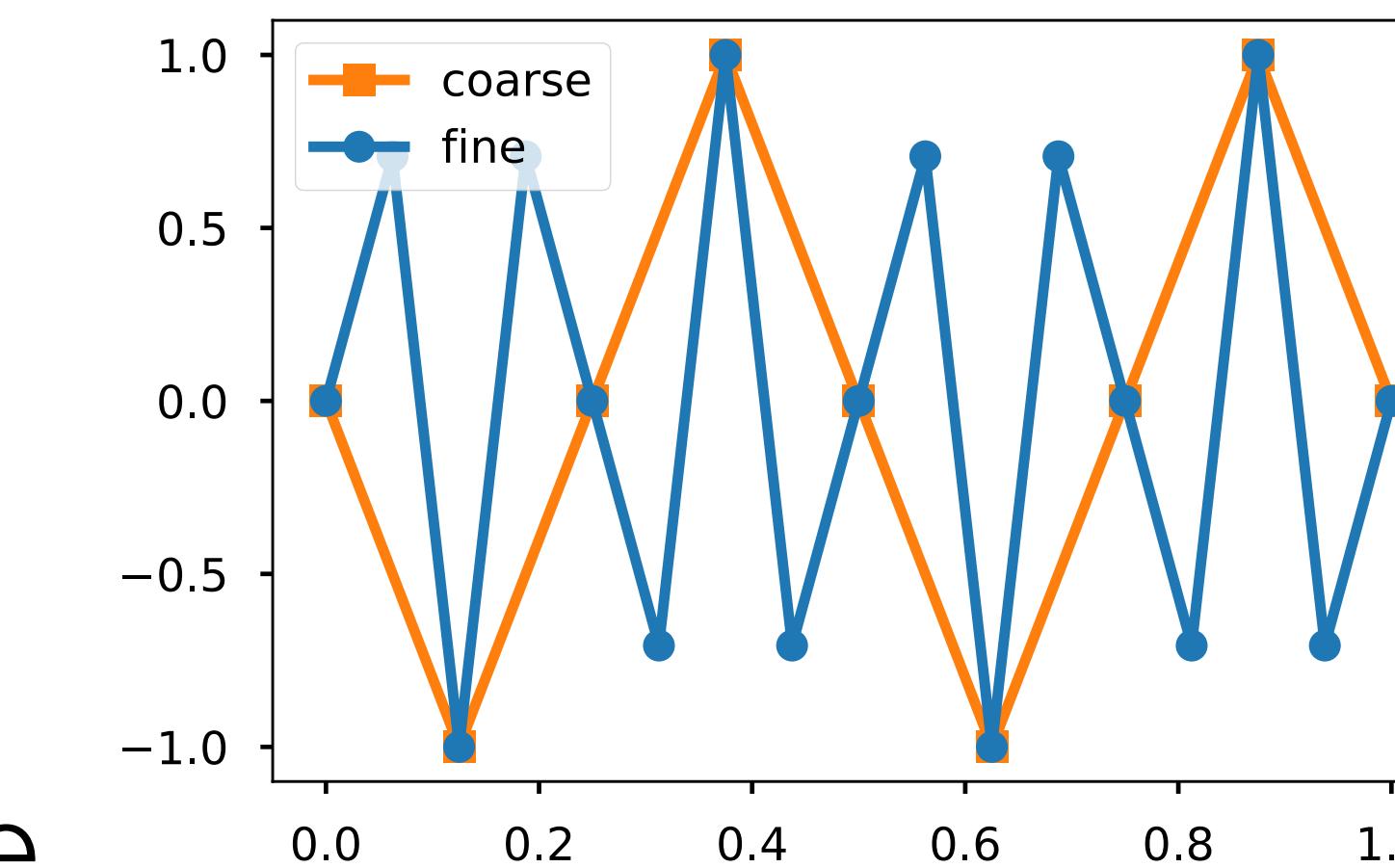


$$\begin{aligned}
 (v_k)_{2j} &= \sin \frac{2jk\pi}{n+1} \\
 &= -\sin \frac{2j(n-k)\pi}{n+1} \quad \text{Fine mode} \\
 &= -\sin \frac{j(n-k)\pi}{(n+1)/2} \quad \text{(every other)} \\
 &= -(\hat{v}_{n-k})_j
 \end{aligned}$$

Fine mode
(every other)
Coarse mode



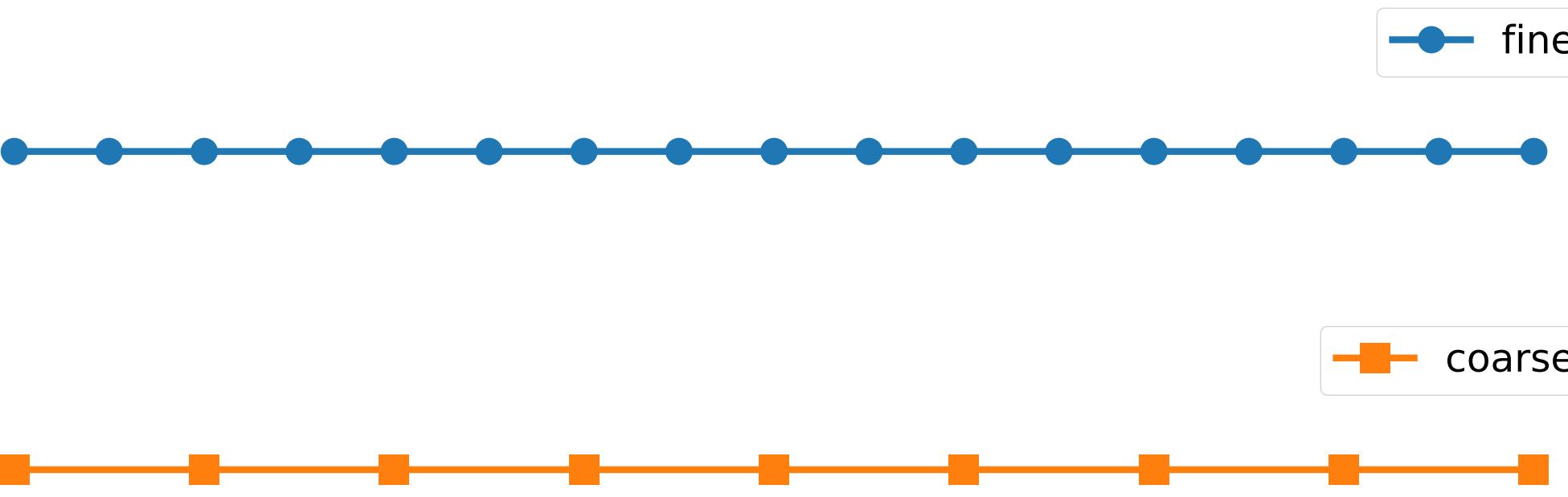
mode 4 of 15



mode 12 of 15

For high modes, k-modes are aliased

Questions to resolve...



- How to transfer between **fine** and **coarse**?
- What do we do to “solve” on a coarse grid?

What to use to transfer...

- Return to our projection problem:

$$x_1 = x_0 + V(V^T A V)^{-1} V^T r_0$$

- If we have a smoothing property, then smoothing a few times will leave smooth error.
- coarse-grid correction: project e_0 onto range of V , and subtract the projection
- Let's construct V from say continuous piecewise linears

Interpolation P: from coarse to fine

- Consider coarse grid

$$\Omega^{2h}$$

- and fine grid

$$\Omega^h$$

- Construct an operator

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

- Such that

$$Pv^{2h}$$

is continuous and piecewise linear

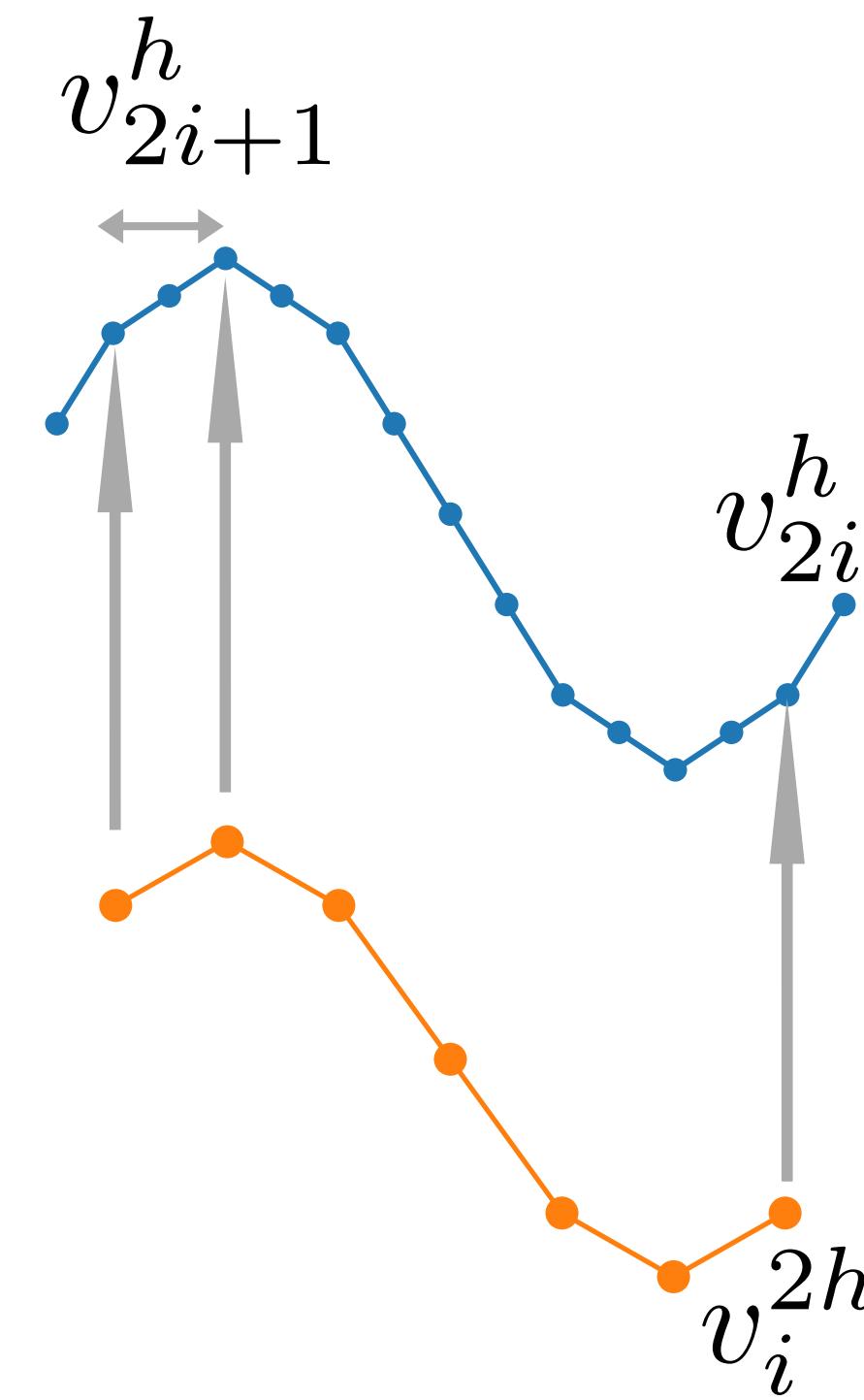
Interpolation

$$v_{2i}^h = v_i^{2h}$$

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

Injection

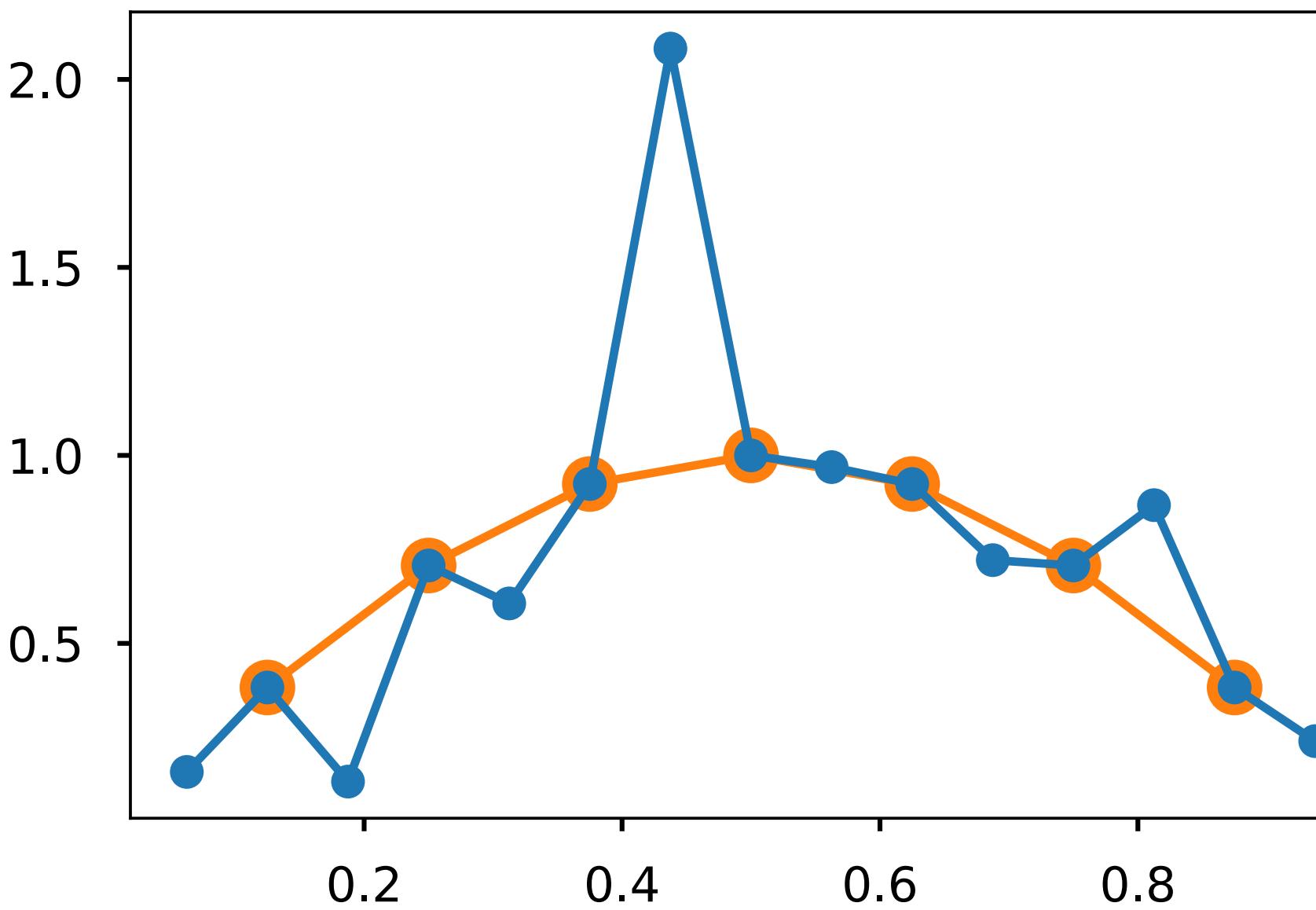
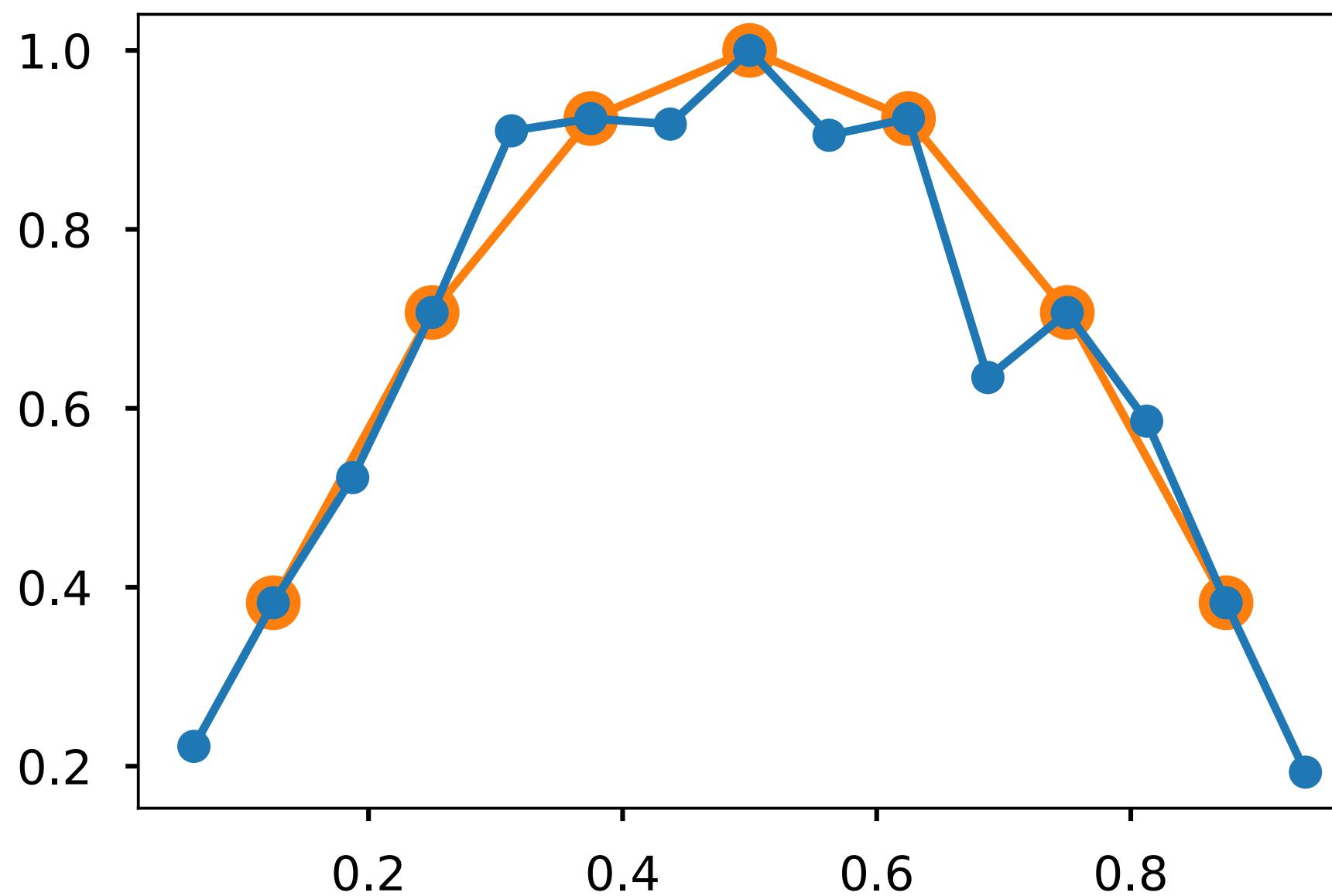
Average
(linear interp)



- Values at points common to both grids are reused (injected)

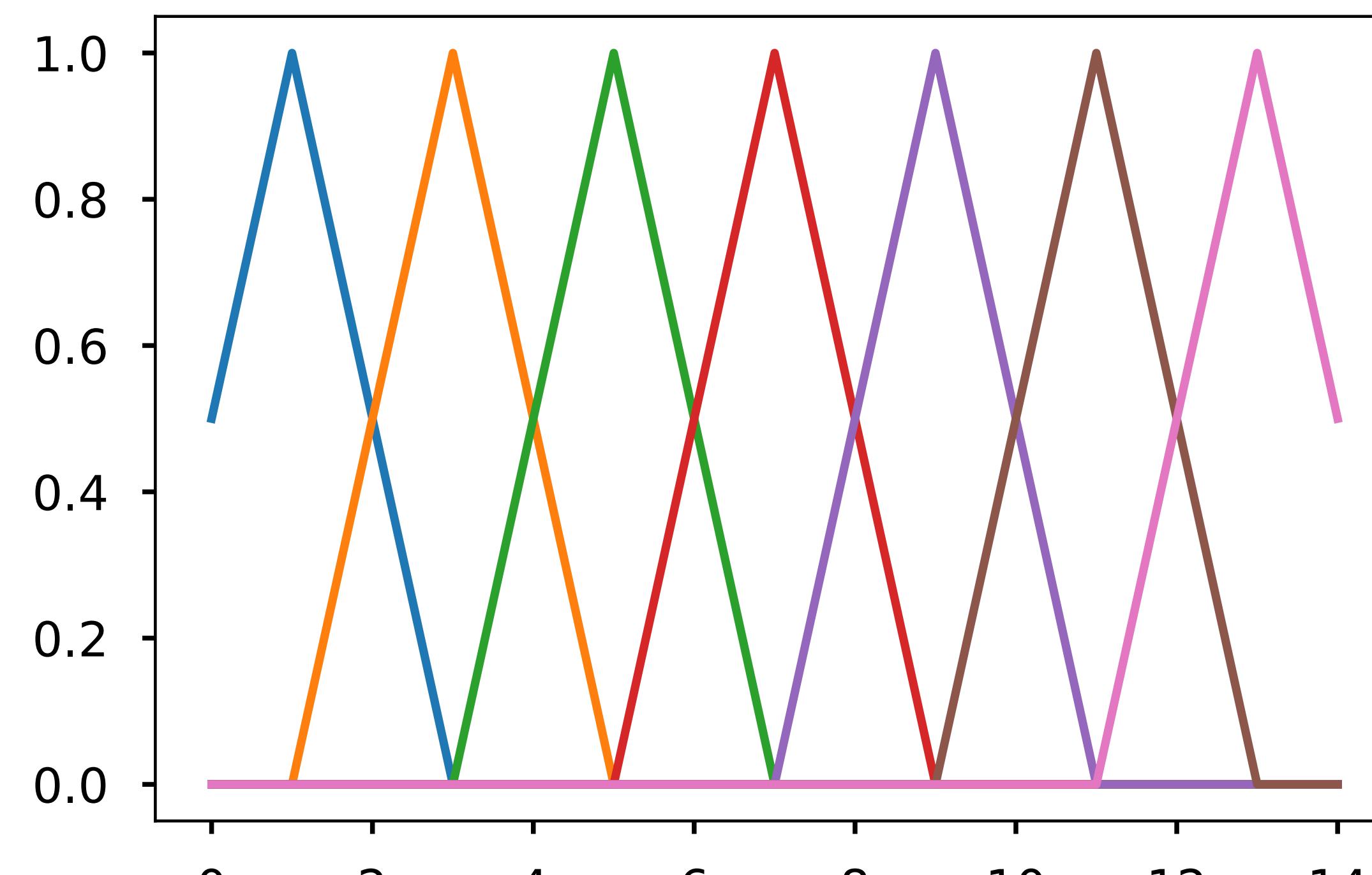
Interpolation

- interpolation gives smooth result



In matrix form

$$P = \frac{1}{2} \begin{bmatrix} 1. & & & & & \\ & 1. & & & & \\ & & 1. & 1. & & \\ & & & 2. & & \\ & & & & 1. & 1. \\ & & & & & 2. \\ & & & & & & 1. \\ & & & & & & & 2. \\ & & & & & & & & 1. \\ & & & & & & & & & 2. \\ & & & & & & & & & & 1. \\ & & & & & & & & & & & 1. \\ & & & & & & & & & & & & 2. \\ & & & & & & & & & & & & & 1. \\ & & & & & & & & & & & & & & 1. \\ & & & & & & & & & & & & & & & 2. \\ & & & & & & & & & & & & & & & & 1. \end{bmatrix}$$



- The columns of P are basis functions (right)
- The fine grid vectors, Pv, are linear combinations of these functions
- Notice: P is full rank!

The coarse grid operator

$$x_1 = x_0 + V(V^T A V)^{-1} V^T r_0$$

(sub)space defined by P

- If V is defined by $\text{span}\{P\}$ – or just P ,
- Then V^T defines **restriction** as P^T
- And the **coarse level operator** is defined by $P^T A P$

Two level method

$$x_1 = x_0 + V(V^T A V)^{-1} V^T r_0$$

↓

$$x_1 = x_0 + P(P^T A P)^{-1} P^T r_0$$

- Given
- Smooth a few times
- Form residual
- Restrict the residual
- Solve the coarse problem
- Interpolate the approx error
- Correct the initial guess

$$\begin{aligned} &x_0 \\ &x_0 \leftarrow x_0 + \omega D^{-1} A r_0 \\ &r_0 = b - A x_0 \\ &P^T r_0 \\ &P^T A P \hat{e}_c = P^T r_0 \\ &\hat{e}_c \\ &x_0 + P \hat{e}_0 \end{aligned}$$

$$A_c \hat{e}_c = r_c$$

- What should this be?
- How many times
- What does this mean?
- What are we interpolating?
- Are we done?

Notes on variants

- An alternative to restriction, is injection
- Or a weighted transpose of linear interpolation (to restrict constants exactly)
- An alternative to $A_c = P^T A_h P$ is to rediscretize $A_c = A_{2h}$

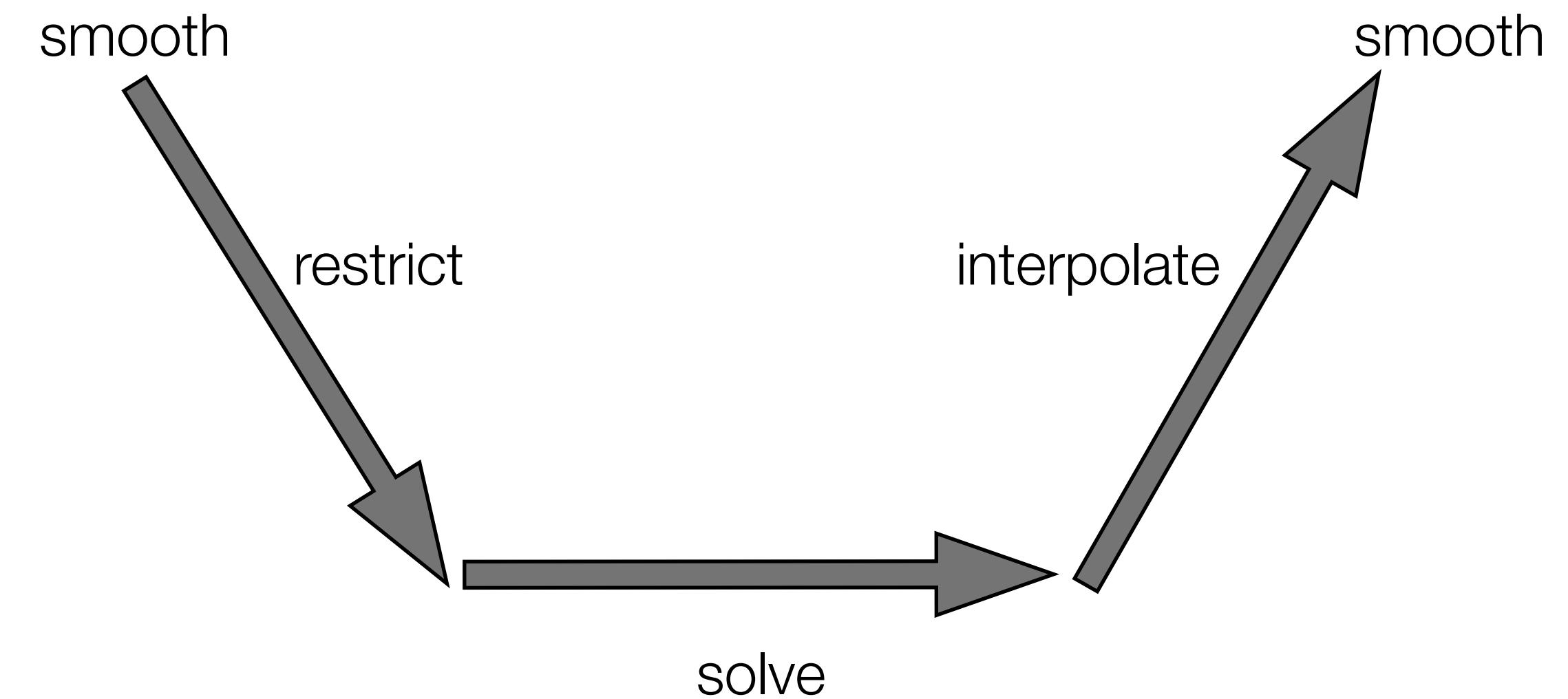
$$\begin{bmatrix} 0 & 1 & 0 & & & \\ & 0 & 1 & 0 & & \\ & & 0 & 1 & 0 & \\ & & & 0 & 1 & 0 \\ & & & & 0 & 1 & 0 \\ & & & & & 0 & 1 & 0 \\ & & & & & & 0 & 1 & 0 \\ & & & & & & & 0 & 1 \\ \end{bmatrix}$$
$$\frac{1}{4} \begin{bmatrix} 1 & 2 & 1 & & & \\ & 1 & 2 & 1 & & \\ & & 1 & 2 & 1 & \\ & & & 1 & 2 & 1 \\ & & & & 1 & 2 & 1 \\ & & & & & 1 & 2 & 1 \\ & & & & & & 1 & 2 & 1 \\ & & & & & & & 1 & 2 \\ \end{bmatrix}$$

$$x_1 \leftarrow x_0 + P A_{2h}^{-1} R r_0$$

Algorithm: two-level multigrid

Input: initial guess

1. Smooth ν_{pre} times on $Au = f$
2. Compute $r = f - Au$
3. Compute $r_c = Rr$
4. Solve $A_c e_c = r_c$
5. Interpolate $\hat{e} = Pe_c$
6. Correct $u \leftarrow u + \hat{e}$
7. Smooth ν_{post} times on $Au = f$



A two-level “V” cycle

How Accurate is Multigrid?

- Consider the exact solution to the PDE u^*

$$-u'' = f$$

- The exact solution to the **discrete** problem u_h^*

$$Au = b$$

- The approximate **discrete** solution $u_h \approx u_h^*$

- Define

$$u^* - u_h^* \quad \text{Discretization error}$$

$$u_h^* - u_h \quad \text{Algebraic error}$$

How Accurate should Multigrid aim to be?

- Would like the total error bounded

$$\begin{aligned}\|u^* - u_h\| &\leq \|u^* - u_h^*\| + \|u_h^* - u^h\| \\ &\leq \varepsilon\end{aligned}$$

- To achieve this,
 - force the discretization (grid space) so that
$$\|u^* - u_h^*\| \leq ch^2 \leq \frac{\varepsilon}{2}$$
 - and the algebraic error (convergence) up to the discretization error
$$\|u_h^* - u^h\| \leq \frac{\varepsilon}{2}$$

Multigrid convergence

- Convergence factor of a cycle – the factor by which the error (residual) is reduced (in some norm) in each iteration

$$\gamma$$

...assume this is independent of n

- Wish to have m cycles such that

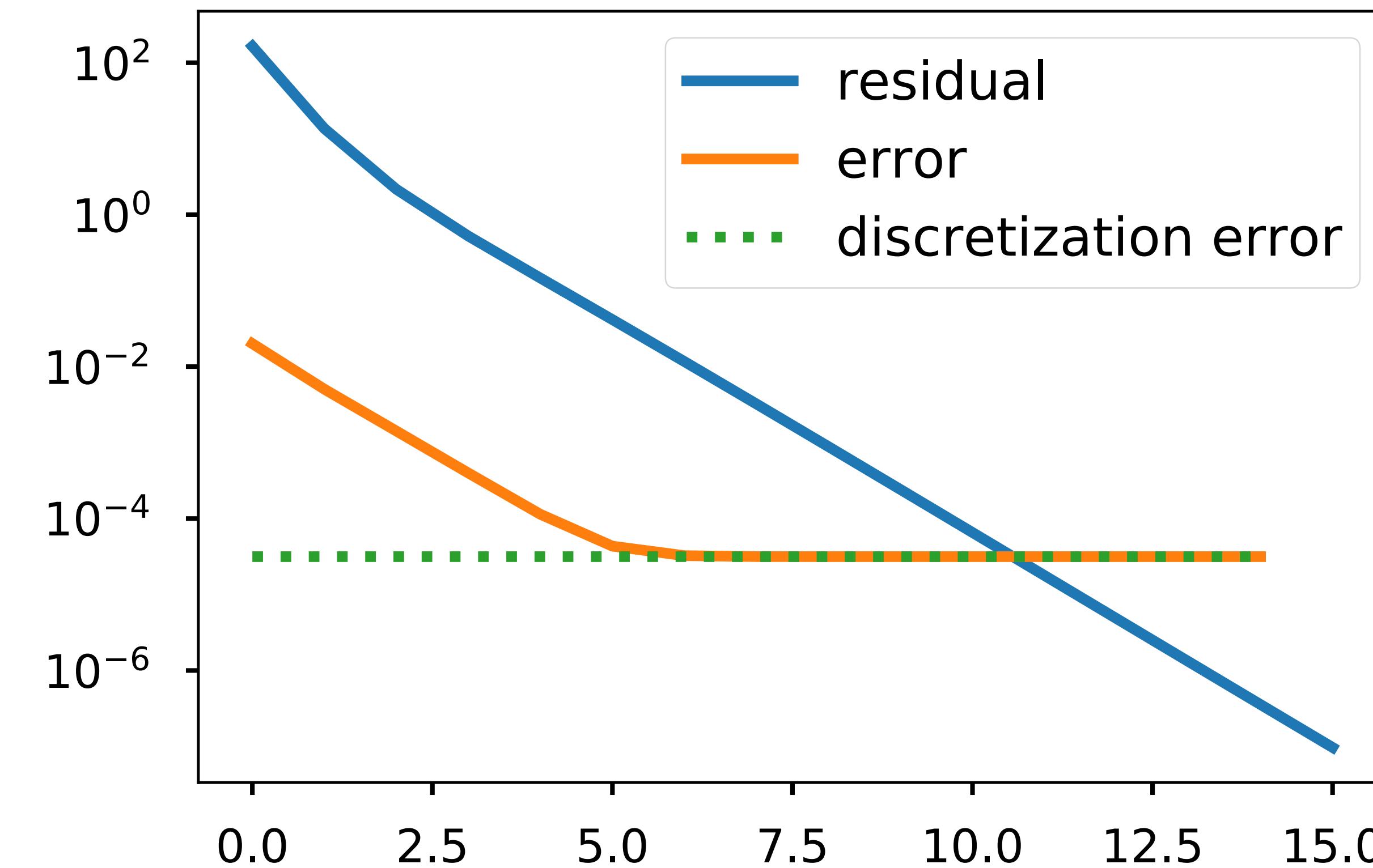
$$\gamma^m \sim \mathcal{O}(n^{-2}) \leq \frac{\varepsilon}{2}$$

- Then we need

$$m \sim \mathcal{O}(\log n)$$

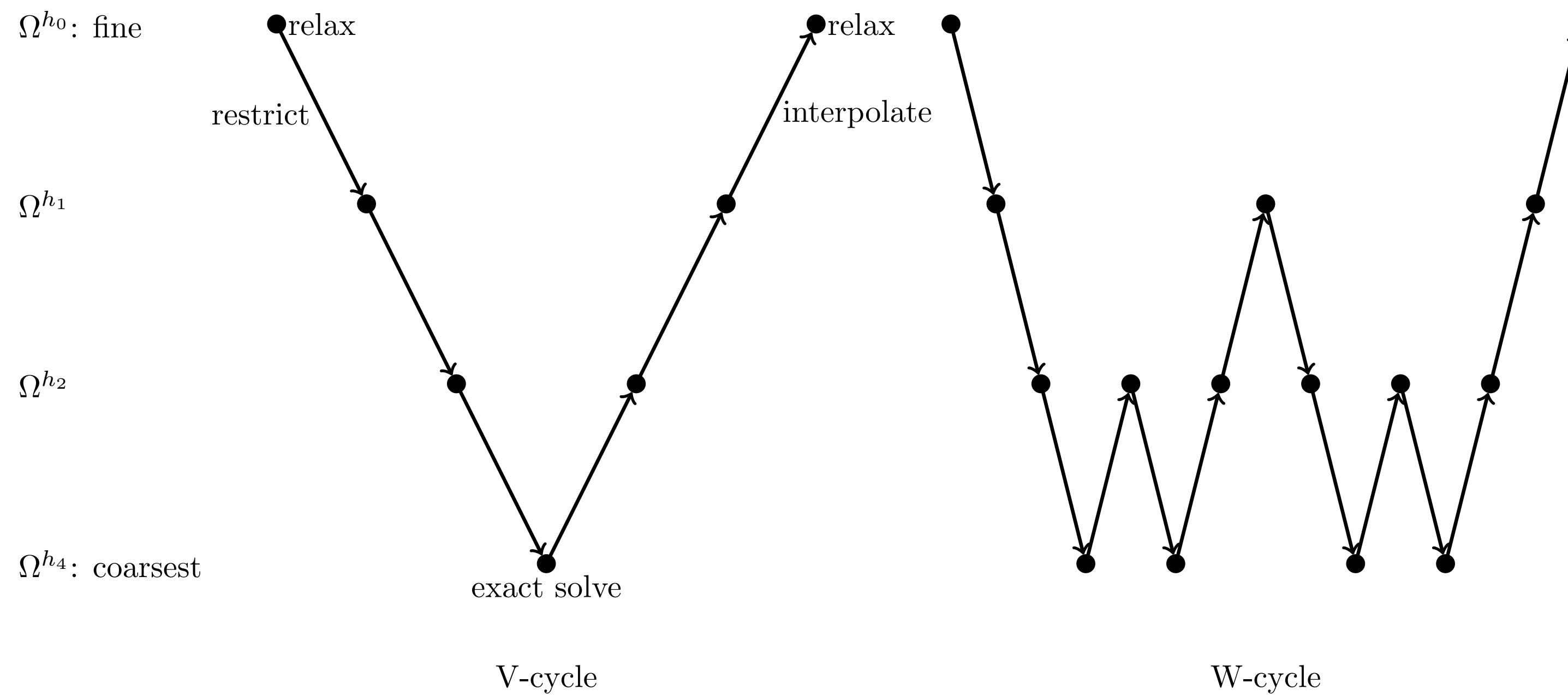
cycles

Algebraic error



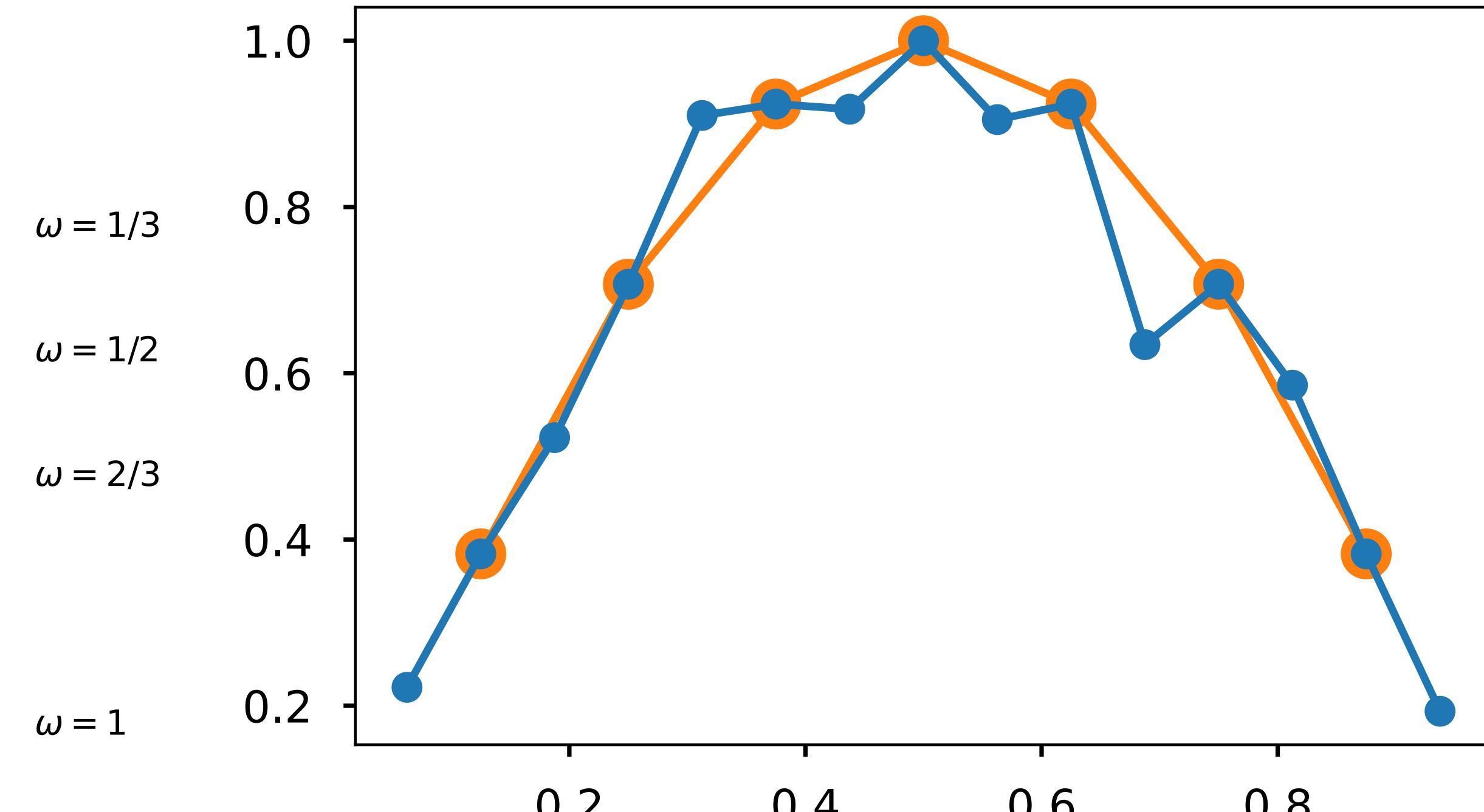
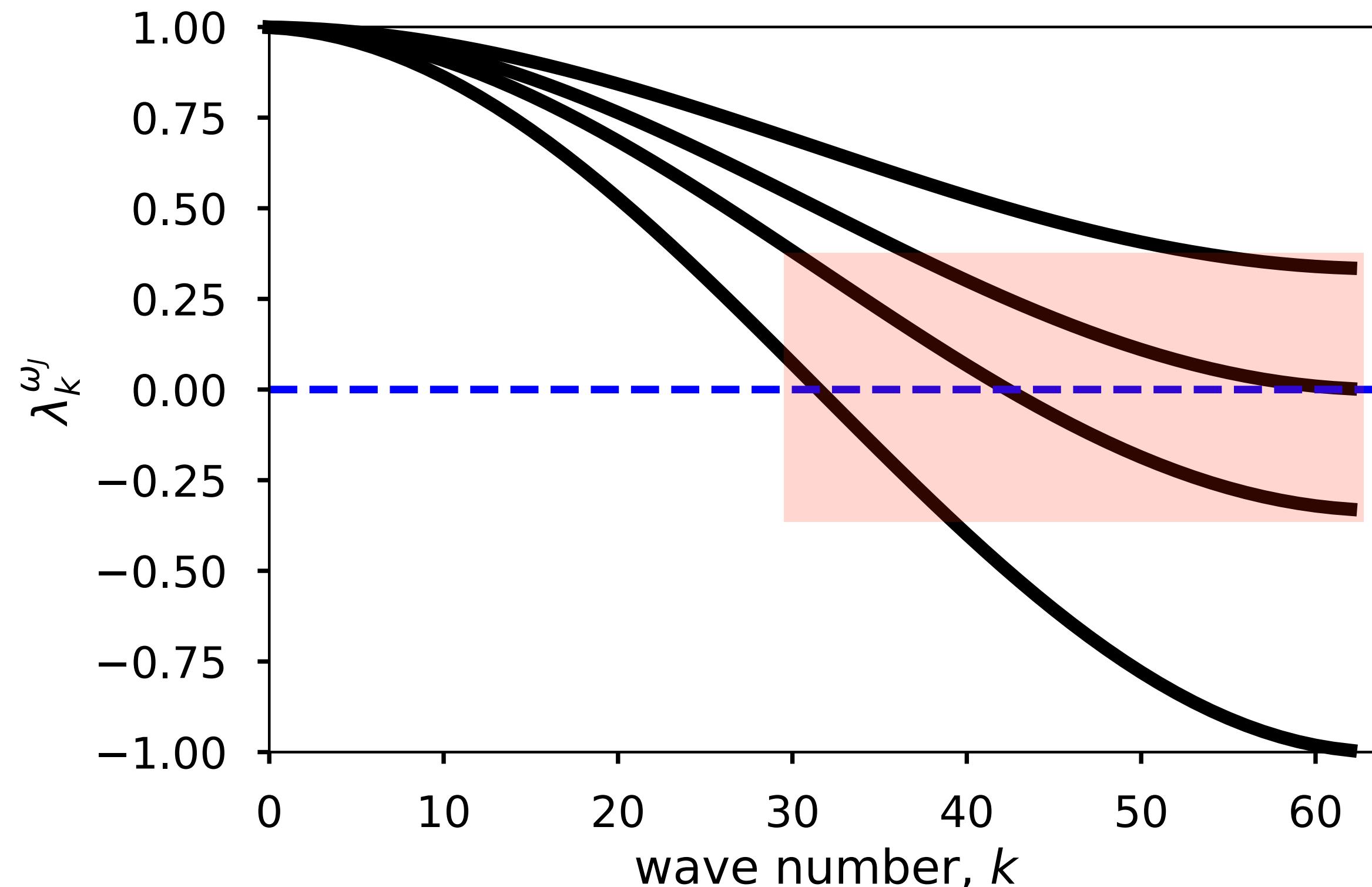
- The total error is limited by the discretization error

The Multigrid V-Cycle and W-Cycle



- W-Cycle can account for inadequate coarse level solves

Recap...

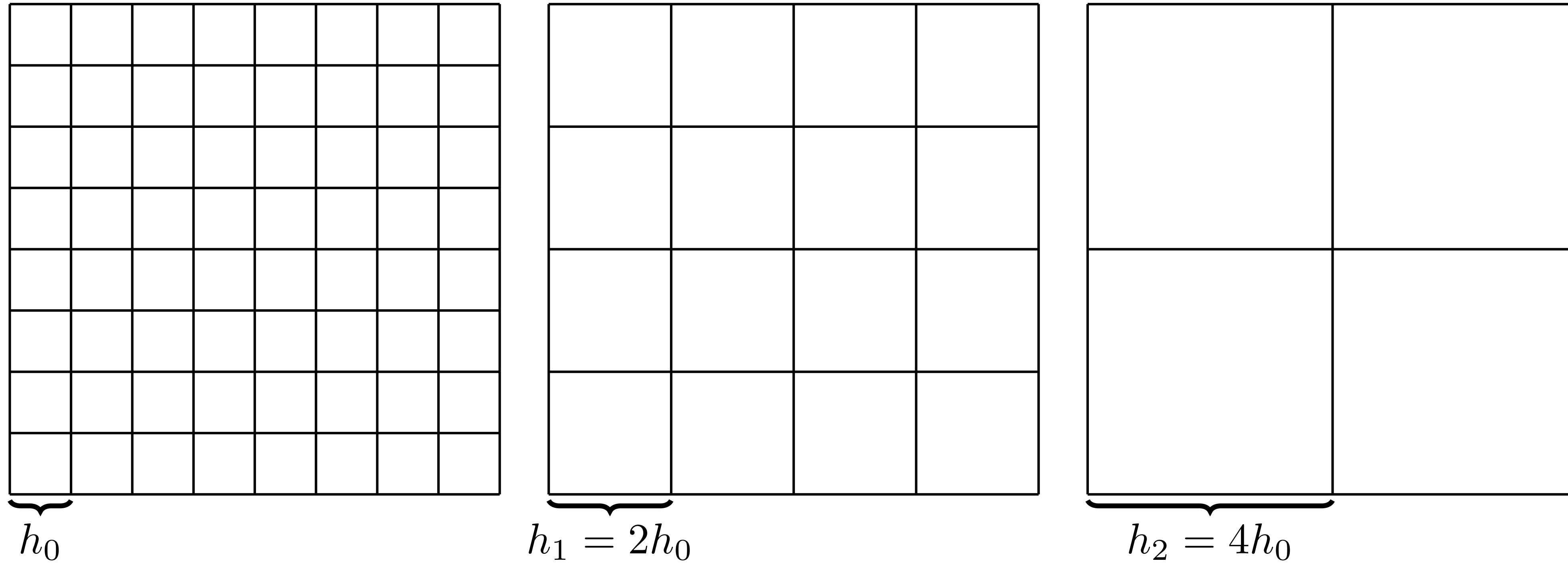


- Smoothing:
Reduce high frequency error

$$e_1 = e_0 - P(P^T A P)^{-1} P^T A G e$$

- Coarse-grid correction:
Reduce smooth-ish things in the range of interpolation

Multigrid in 2D



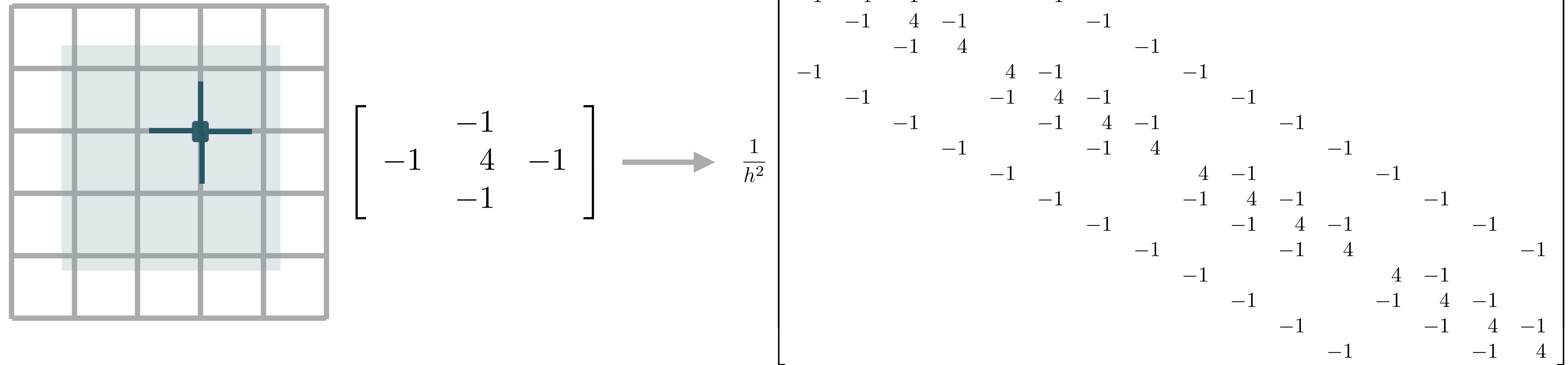
- Again, assume we have a sequence of uniform grids
- Relaxation remains the same (what is ω ?)

Multigrid in 2D

- Model problem

$$\begin{aligned} -u_{xx} - u_{yy} &= f \\ u &= 0 \quad \text{on boundary} \end{aligned}$$

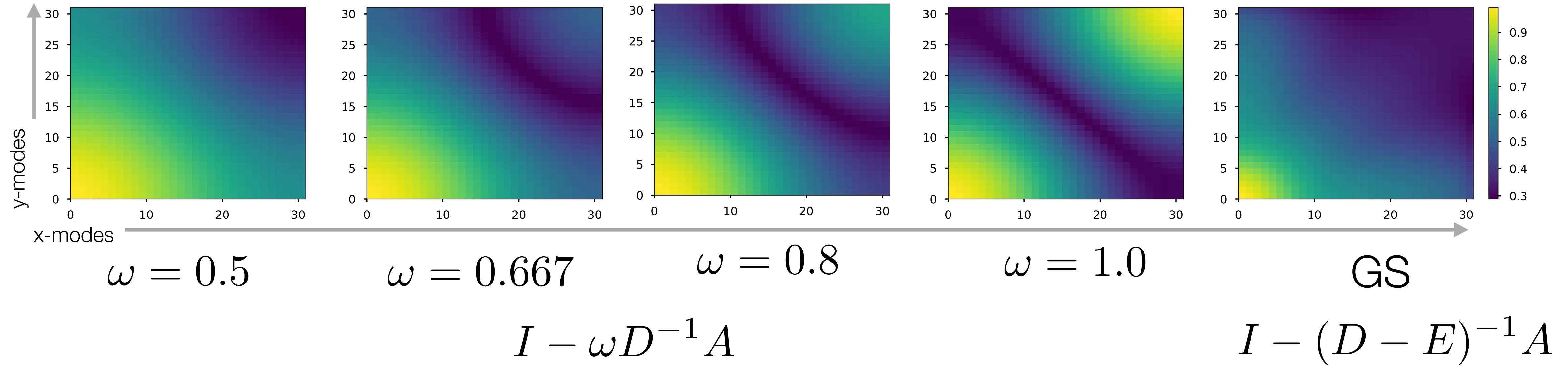
- Results in the stencil / matrix



Relaxation in 2D

$$\sin\left(\frac{k_x i \pi}{n+1}\right) \sin\left(\frac{k_y j \pi}{n+1}\right)$$

Convergence factor
over 10 sweeps



- weighted Jacobi: Same issue – need to select a parameter
- Gauss-Seidel improved
- Red-Black Gauss-Seidel and other schemes even more effective

Interpolation in 2D

- Bilinear interpolation, tensor product of 1D interpolation

$$\begin{bmatrix} 1 & 2 & 1 \end{bmatrix}$$



$$\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

- Example $3 \times 3 \rightarrow 7 \times 7$

$$\begin{bmatrix} 0.5 \\ 1 \\ 0.5 & 0.5 \\ 1 \\ 0.5 & 0.5 \\ 1 \\ 0.5 \end{bmatrix} \otimes \begin{bmatrix} 0.5 & 1 \\ 1 & 0.5 \\ 0.5 & 0.5 \\ 1 & 0.5 \\ 0.5 & 0.5 \\ 1 \\ 0.5 \end{bmatrix}$$



$$\begin{bmatrix} 1 & 2 & 1 & 2 & 1 & 2 & 1 \\ 2 & 4 & 2 & 4 & 2 & 4 & 2 \\ 1 & 2 & 1 & 2 & 1 & 2 & 1 \\ 2 & 4 & 2 & 4 & 2 & 4 & 2 \\ 1 & 2 & 1 & 2 & 1 & 2 & 1 \\ 2 & 4 & 2 & 4 & 2 & 4 & 2 \\ 1 & 2 & 1 & 2 & 1 & 2 & 1 \end{bmatrix}$$

A few observations so far: **One**

- Let's consider a V(1,1) cycle – weighted Jacobi, etc.
- The **error** propagation for this looks like

$$e_1 = \frac{G(I - P(P^T A P)^{-1} P^T A) G e_0}{M}$$

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

$$M e_k \leftarrow 0?$$

- One thing we can do, is consider bounds on each operation.

A few observations so far: **One**

- Take the operator

$$M = G(I - P(P^T A P)^{-1} P^T A)G$$

- And makes some bounds

$$\|I - P(P^T A P)^{-1} P^T A\| \|G\|^2$$

The diagram shows the expression $\|I - P(P^T A P)^{-1} P^T A\| \|G\|^2$ with a horizontal line underneath it. This line is divided into two segments by a vertical tick mark. The left segment is highlighted in blue and labeled "approximation property". The right segment is highlighted in green and labeled "smoothing property".

- General $\|G\| \leq 1$

1D over $[n/2, n]$:

$$\|G\| \leq \frac{1}{3}$$

2D over $[n/2, n]$:

$$\|G\| \leq \frac{3}{5}$$

A few observations so far: **One**

$$\|I - P(P^T A P)^{-1} P^T A\| \|G\|^2$$

approximation property smoothing property

- General $\|G\| \leq 1$

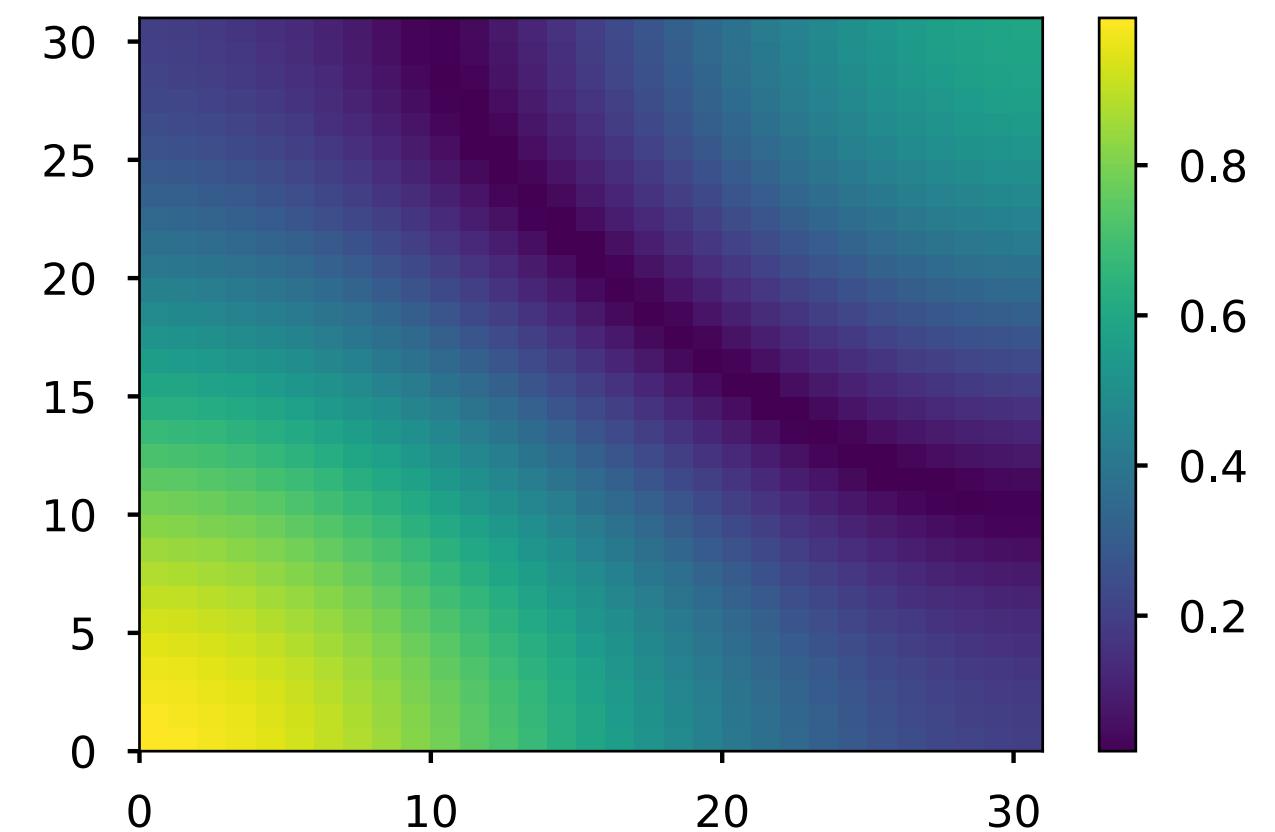
$$1D \text{ over } [n/2, n]: \quad \|G\| \leq \frac{1}{3}$$

$$2D \text{ over } [n/2, n]: \quad \|G\| \leq \frac{3}{5}$$

- Also, if w s.t. $Aw \in \mathcal{N}(P^T)$

then $(I - P(P^T A P)^{-1} P^T A)w = w$

$$\|\cdot\| \geq 1$$



A few observations so far: **Two**

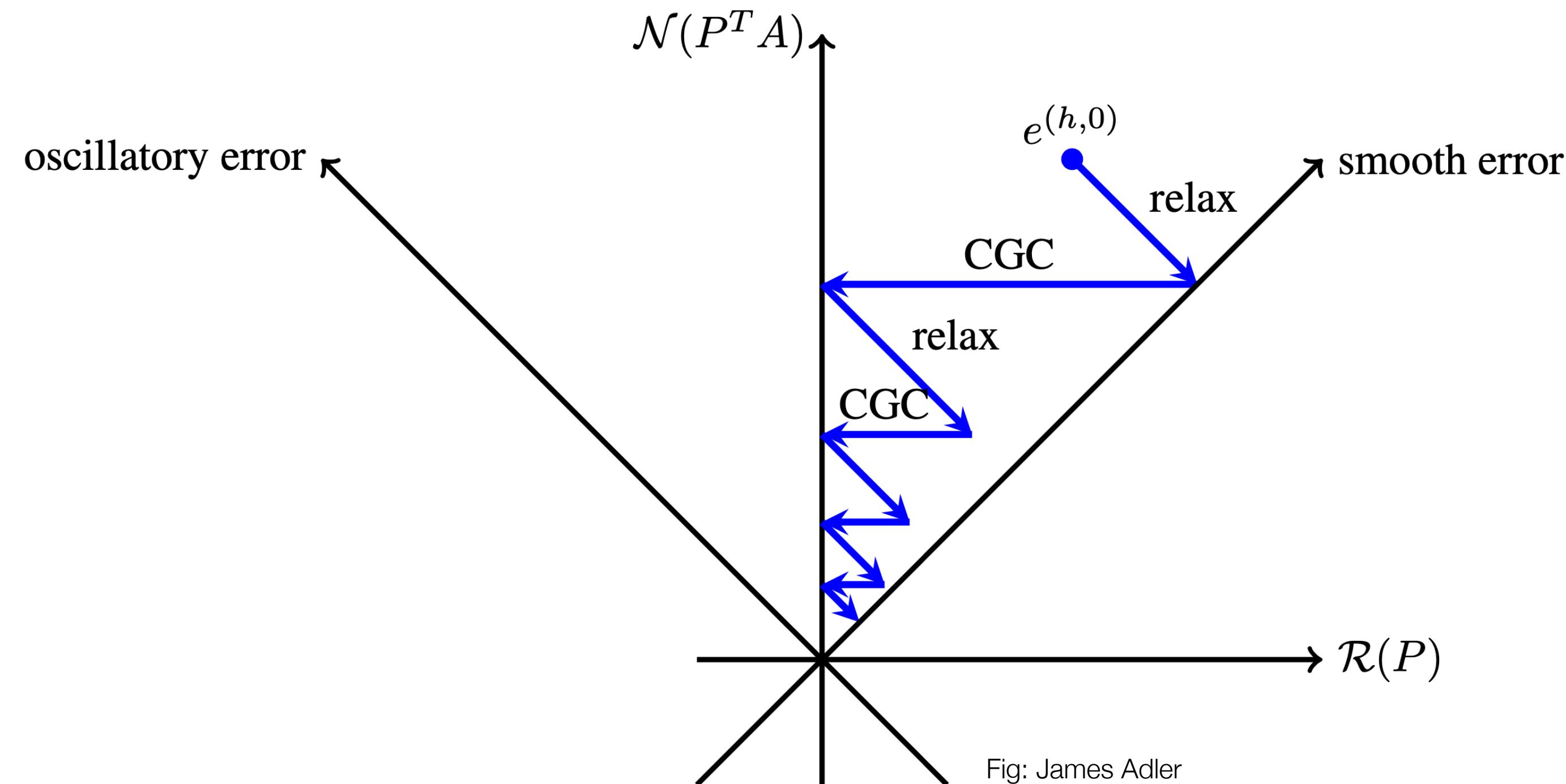
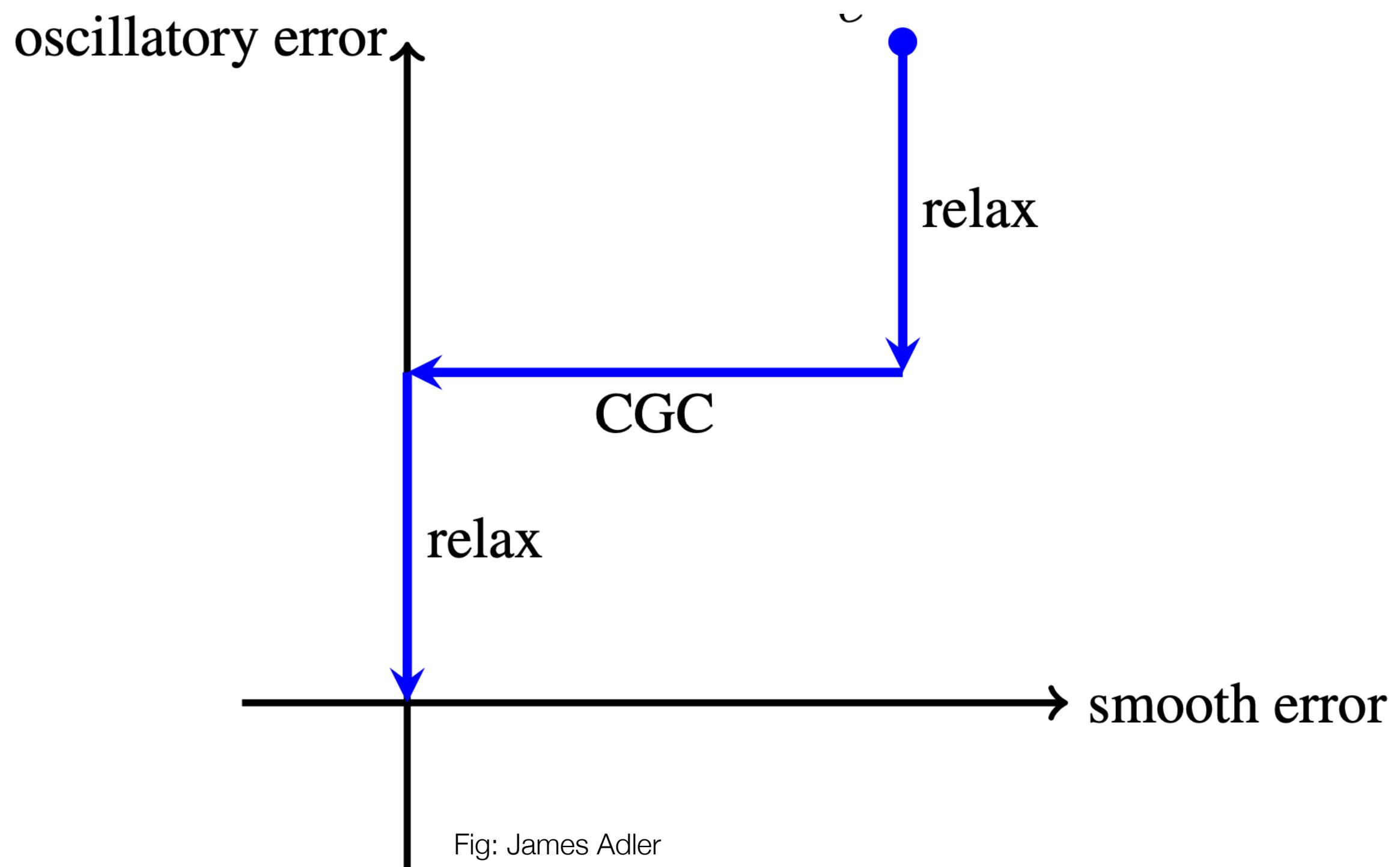


Fig: James Adler

- Complementary processes:
 - **relaxation:** targets (Fourier) smoothing
 - **coarse grid correction:** targets things in the range of interpolation

A few observations so far: **Three**



$$e_1 \leftarrow (I - P(P^T A P)^{-1} P^T A) G e_0$$

$$G e_0 \in \mathcal{R}(P) \Rightarrow e_1 = 0$$

interpolation should capture what relaxation misses

Fig: James Adler

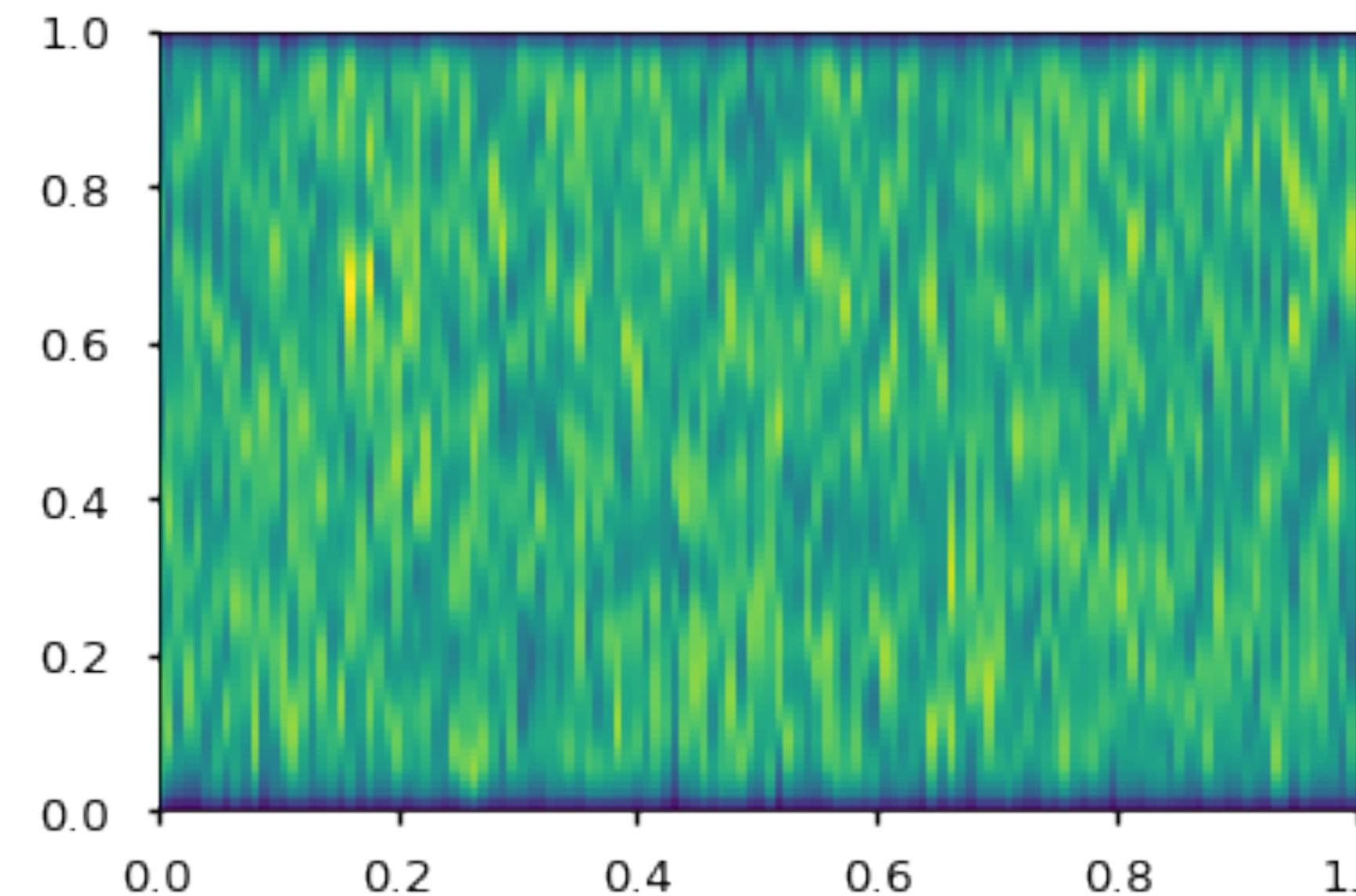
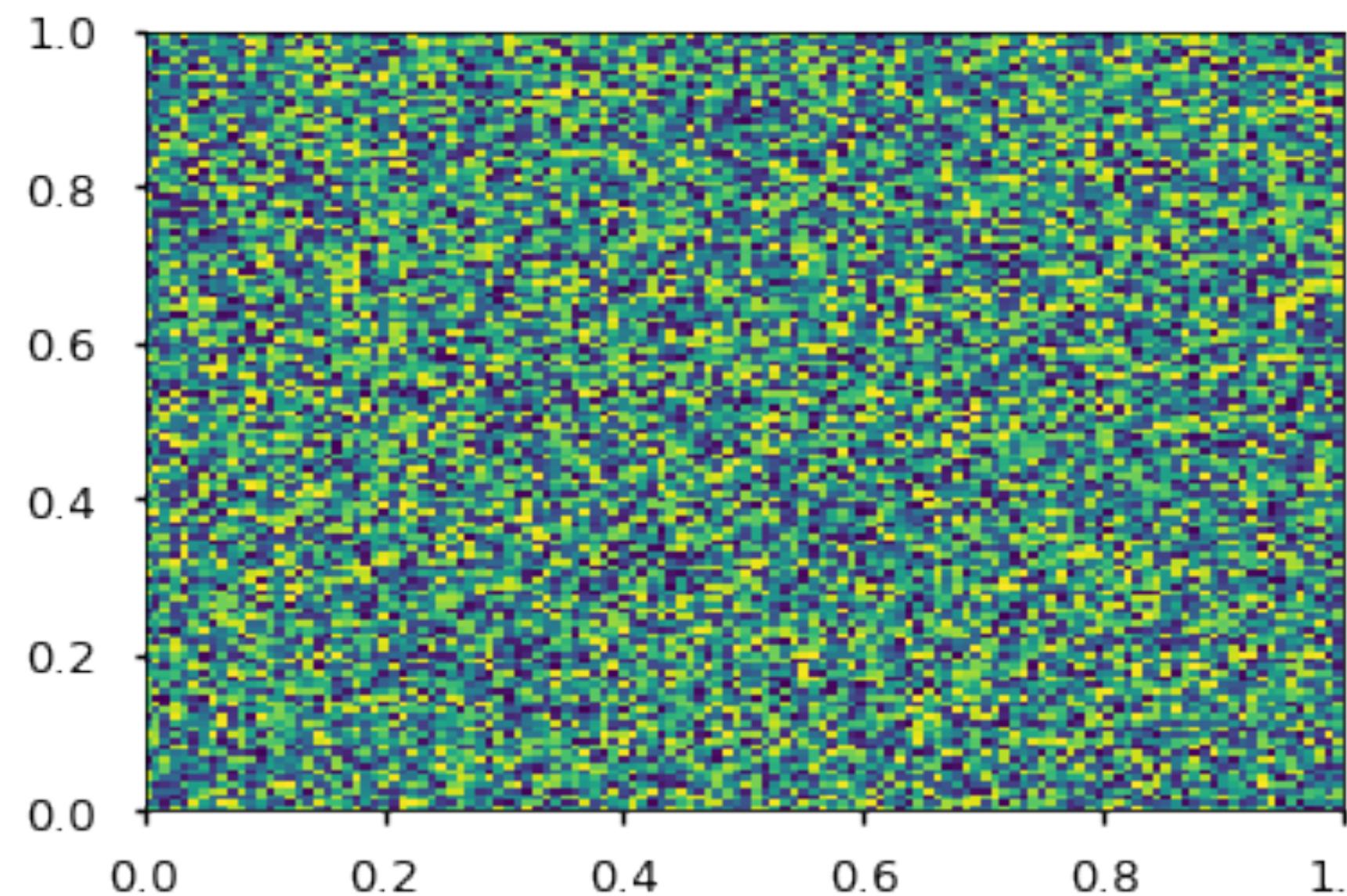
What can go wrong?!

- Consider an *anisotropic problem*

$$-\varepsilon u_{xx} - u_{yy} = f$$

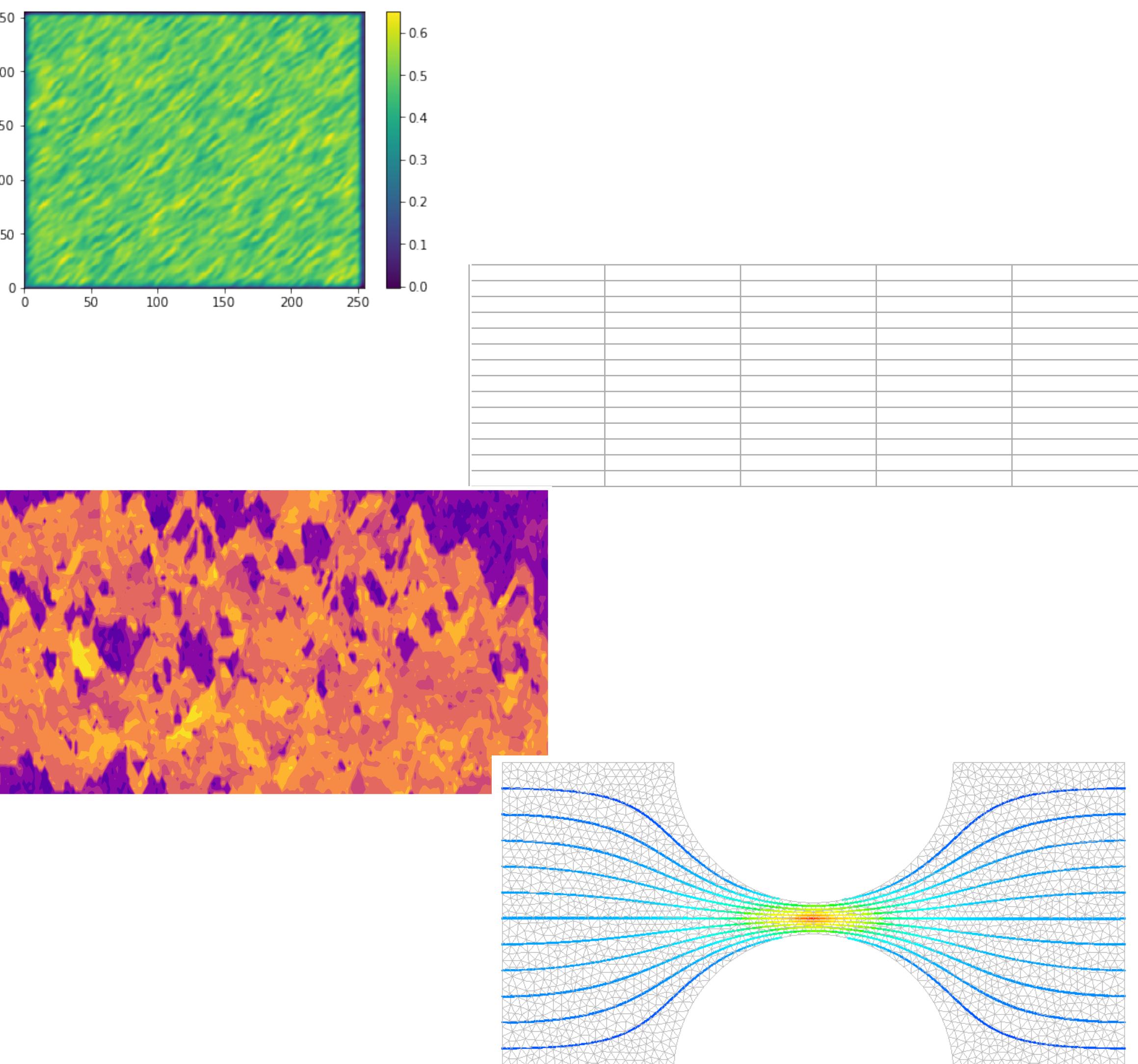
$$u = 0 \quad \text{on boundary}$$

$$\begin{bmatrix} & -1 & \\ -\varepsilon & 2 + 2\varepsilon & -\varepsilon \\ & -1 & \end{bmatrix}$$



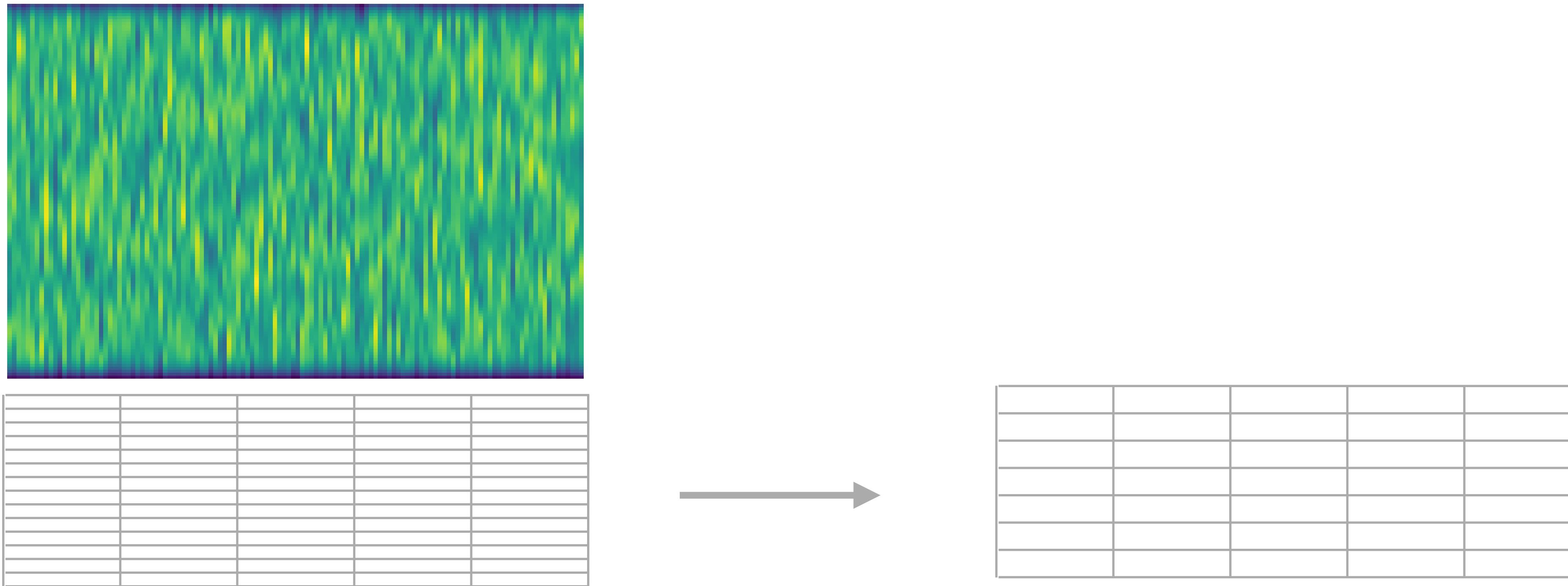
What can go wrong?!

- Anisotropy
- Mesh stretching
- Jumping coefficients
- Non-elliptic



Options for more robust Multigrid

- **Semicoarsening:** Coarsen in the direction of smoothness

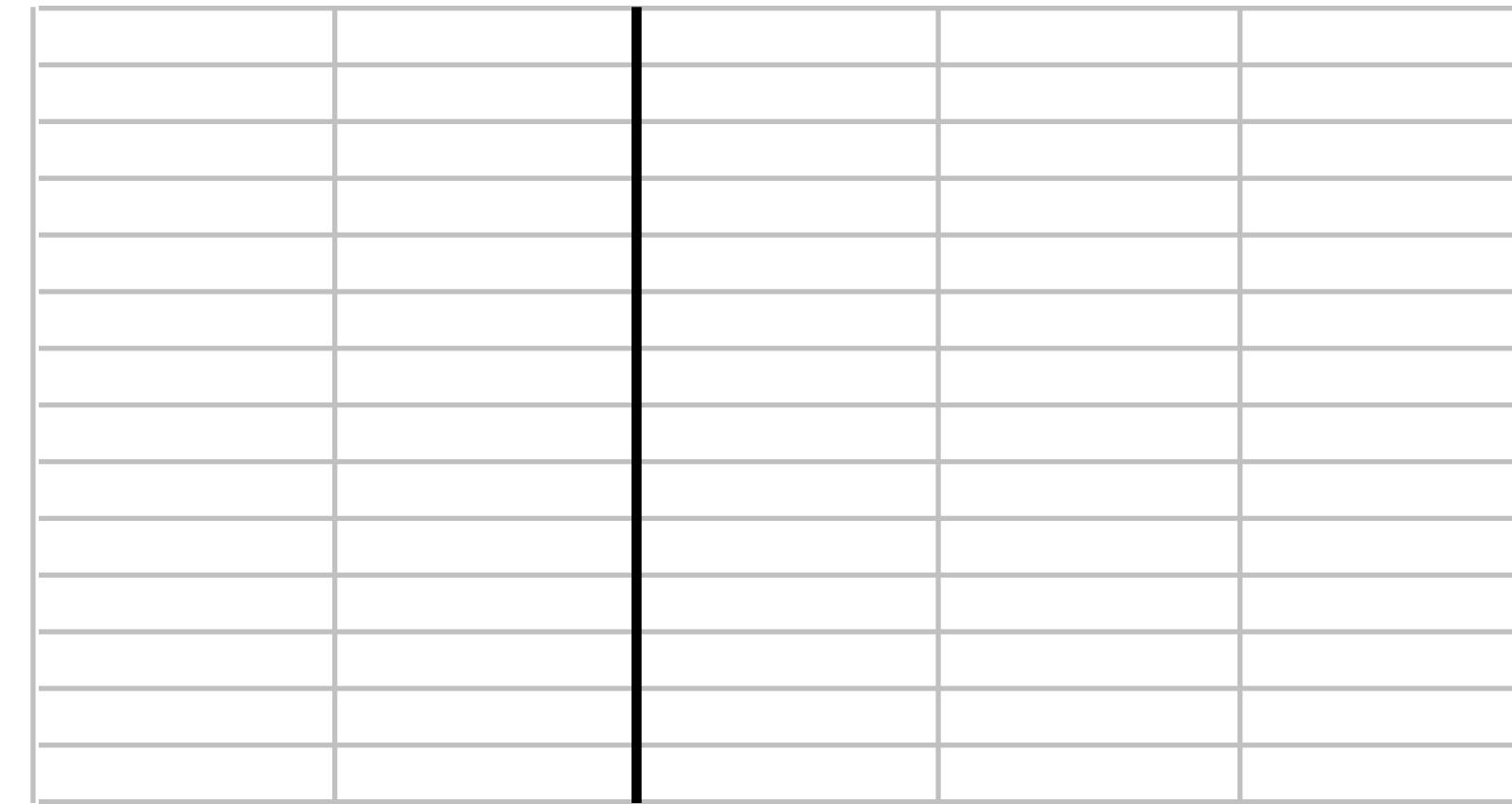
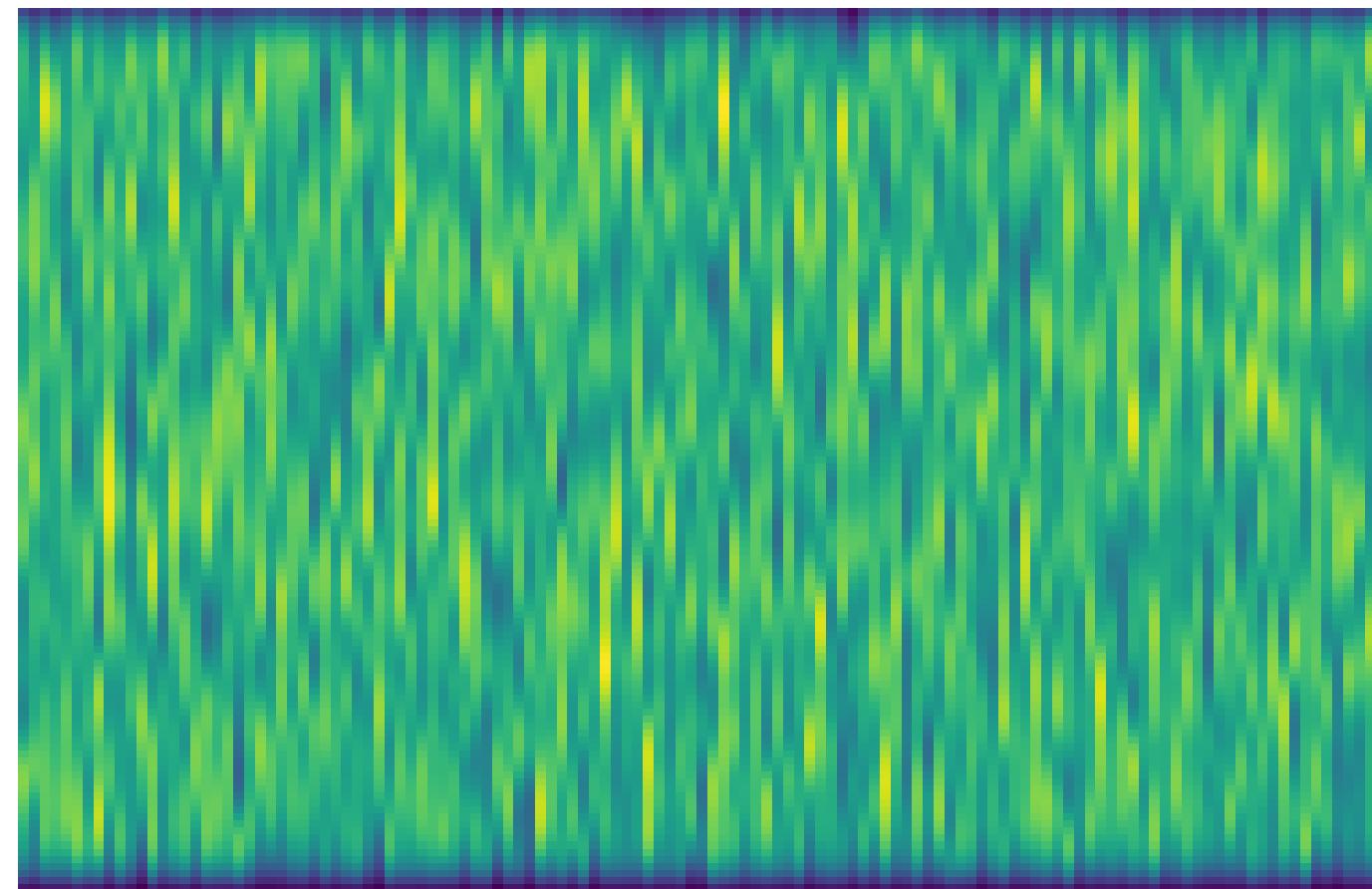


- Downside: if anisotropy varies in another direction, we need a different grid

Options for more robust Multigrid

- **Line/plane relaxation**

Perform relaxation in groups (in a line)

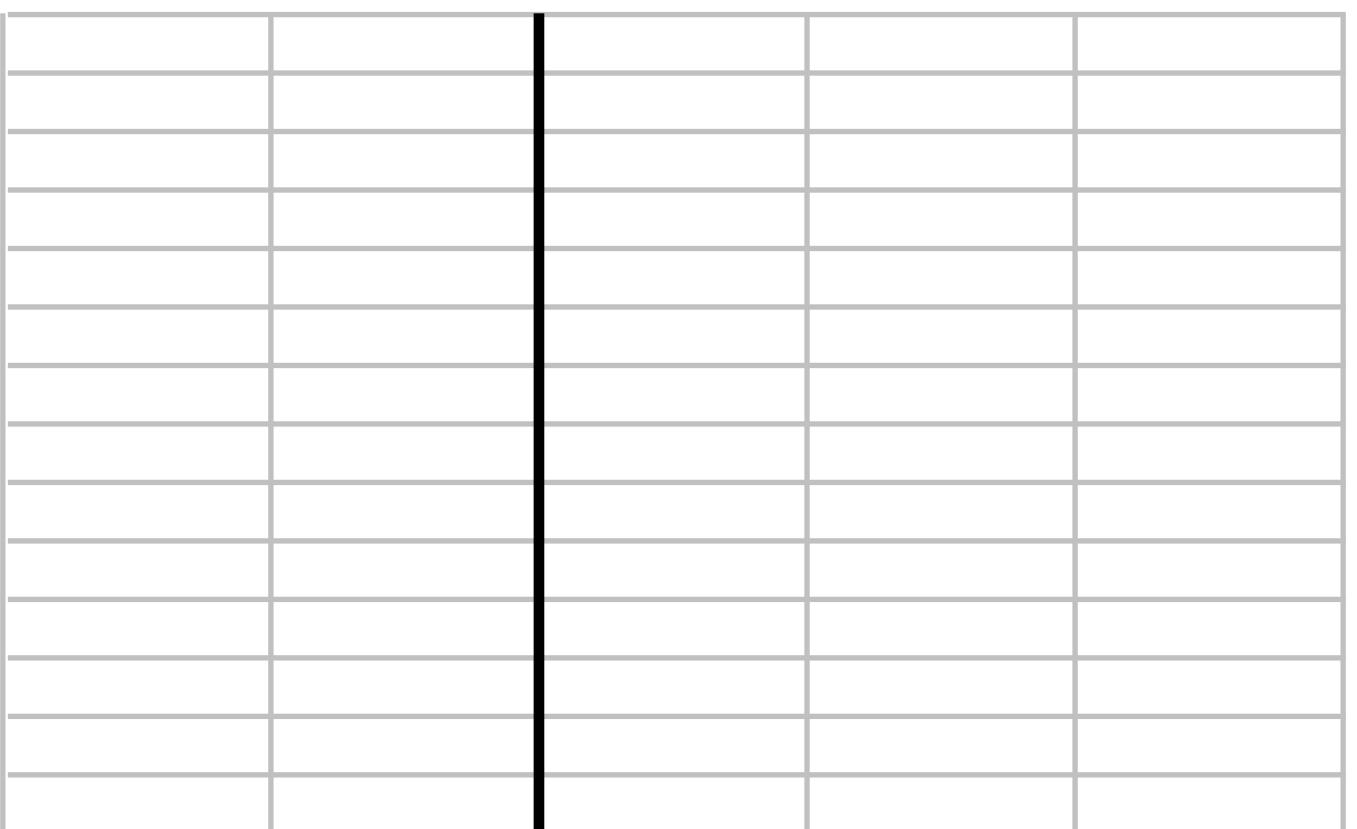
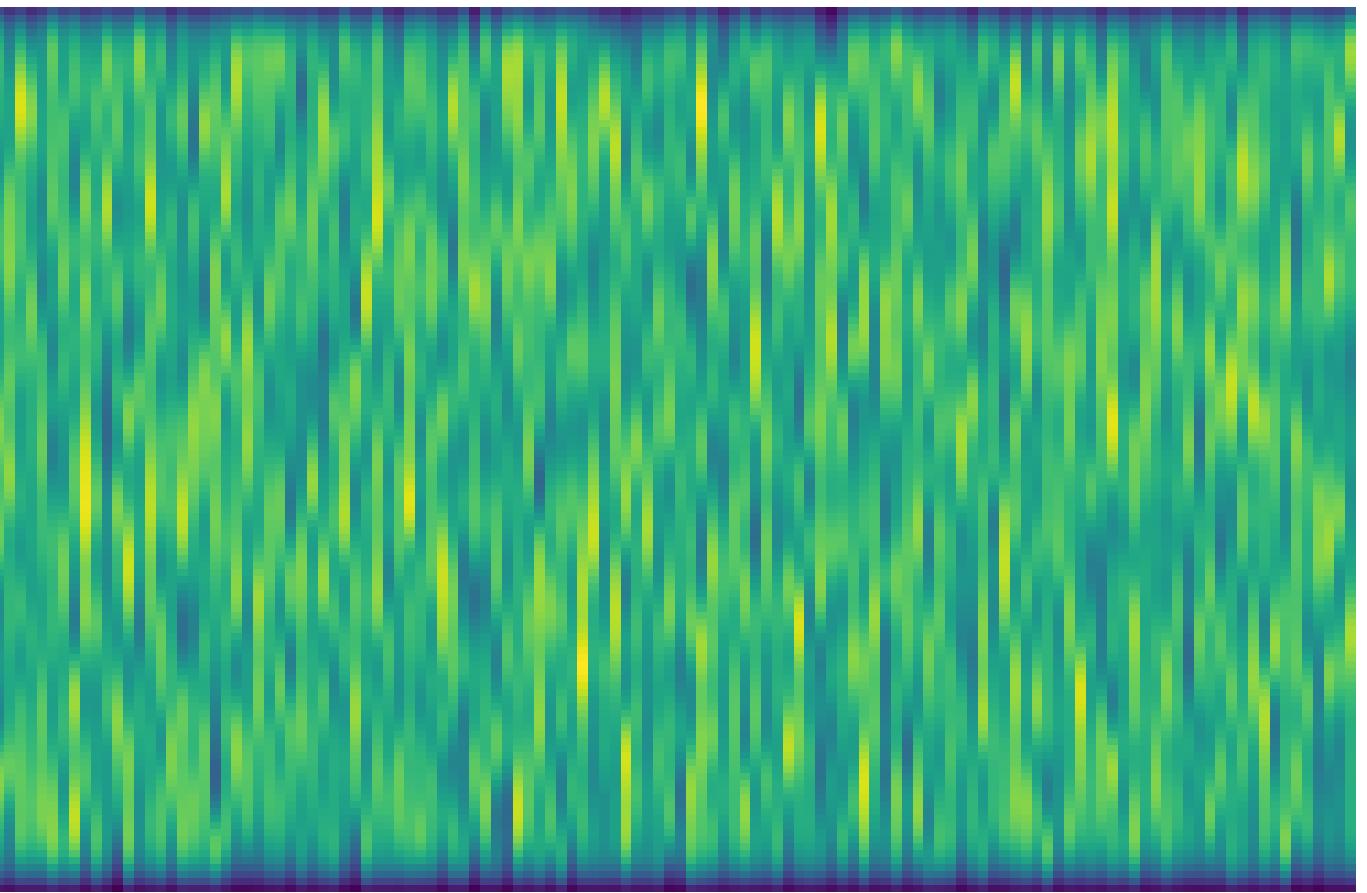


- Example:

- relax error in block in the y-direction (constant x-lines)
- no (block) smoothing in the x-direction (constant y-lines)

Options for more robust Multigrid

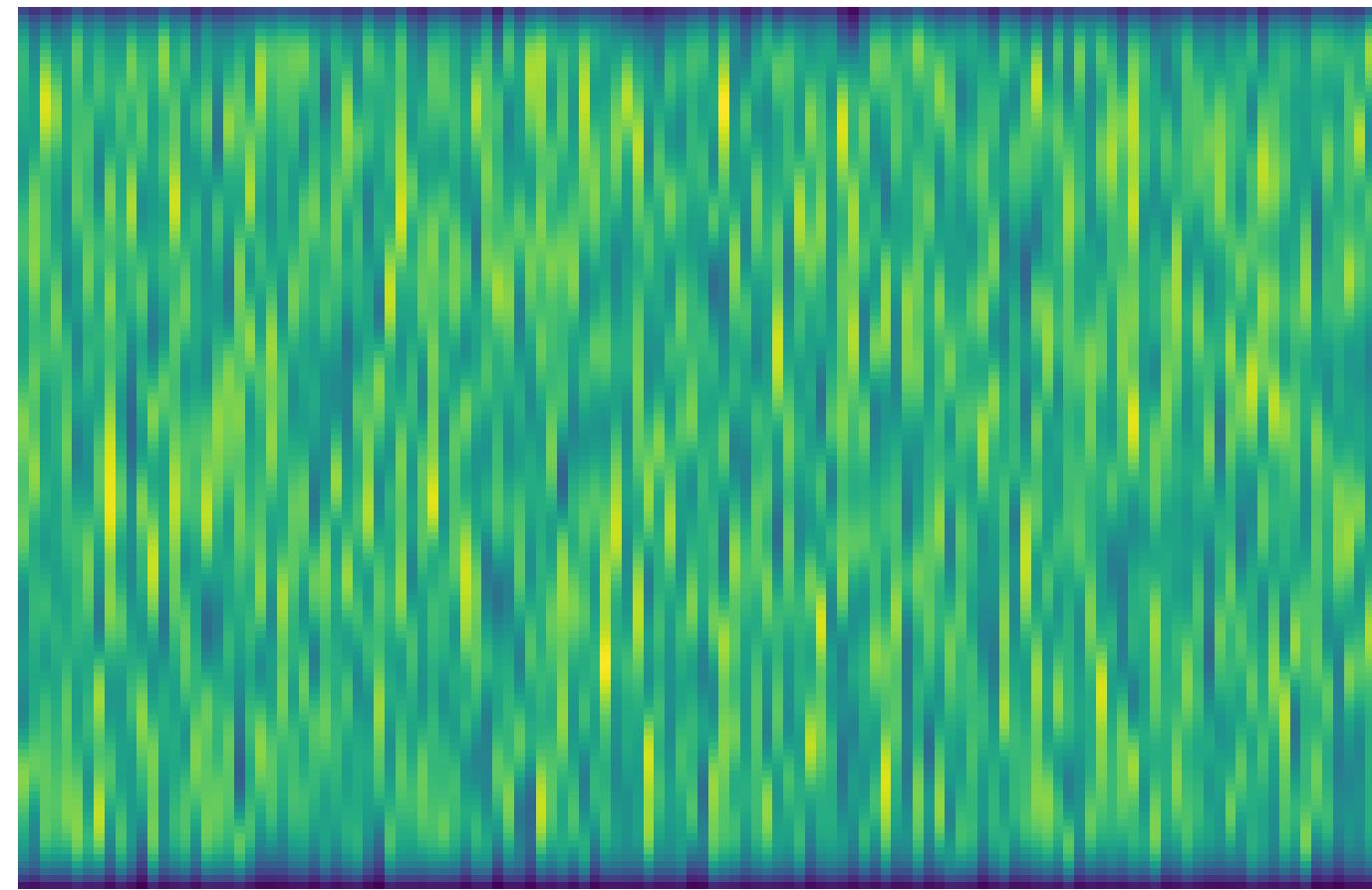
- **Line/plane relaxation**
- For each x-line (lines of strong anisotropy):
 - Eliminate the residual on the entire line
 - (Gauss-Seidel, by lines)



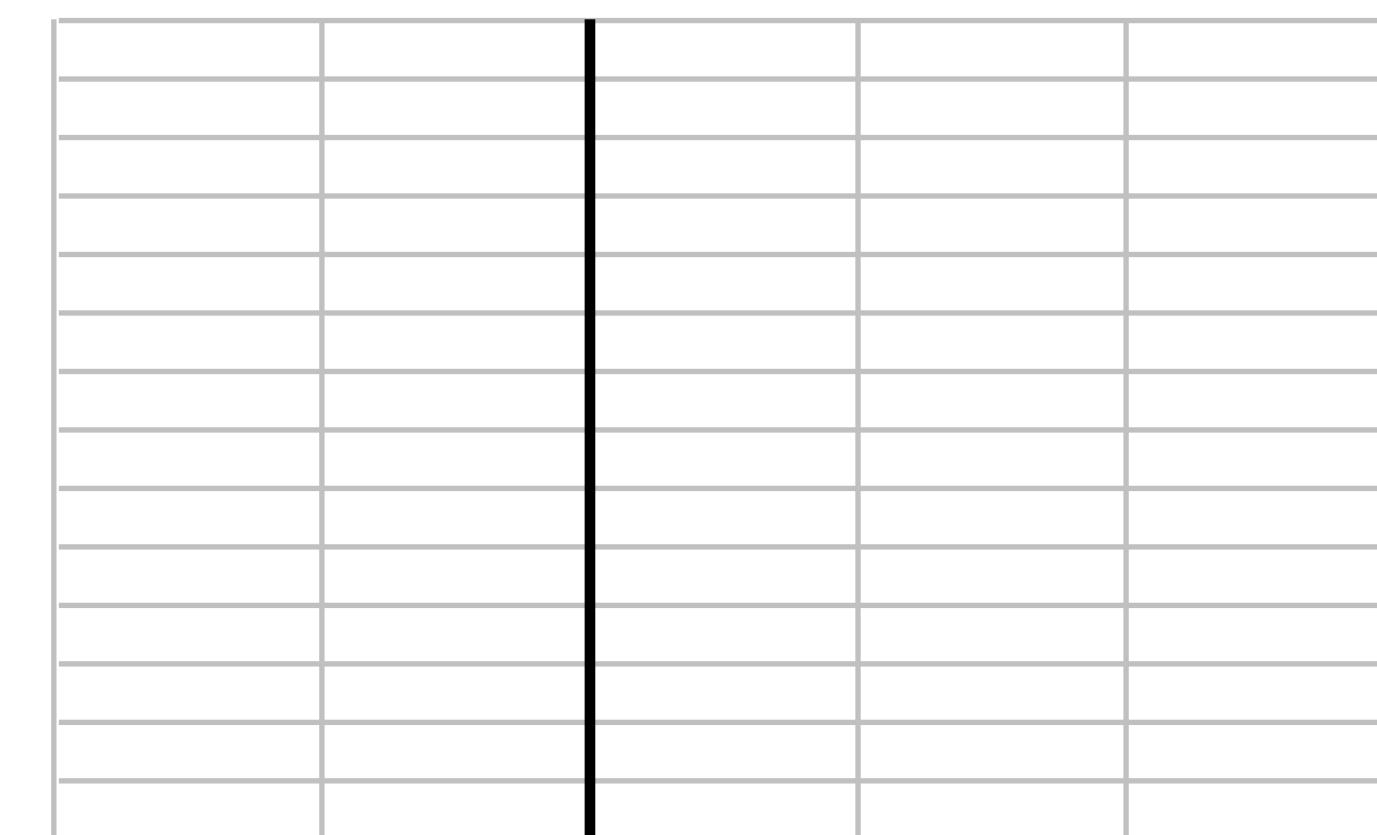
Options for more robust Multigrid

- **Line/plane relaxation**

- $$A = (n + 1)^2 \begin{bmatrix} A_{1D} & -\varepsilon I & & \\ -\varepsilon I & A_{1D} & -\varepsilon I & \\ & -\varepsilon I & A_{1D} & -\varepsilon I \\ & & \ddots & \\ & & & -\varepsilon I & A_{1D} \end{bmatrix}$$



$$A_{1D} = (n + 1)^2 \begin{bmatrix} 2 + 2\varepsilon & -1 & & \\ -1 & 2 + 2\varepsilon & -1 & \\ & \ddots & \ddots & \\ & & -1 & 2 + 2\varepsilon \end{bmatrix}$$



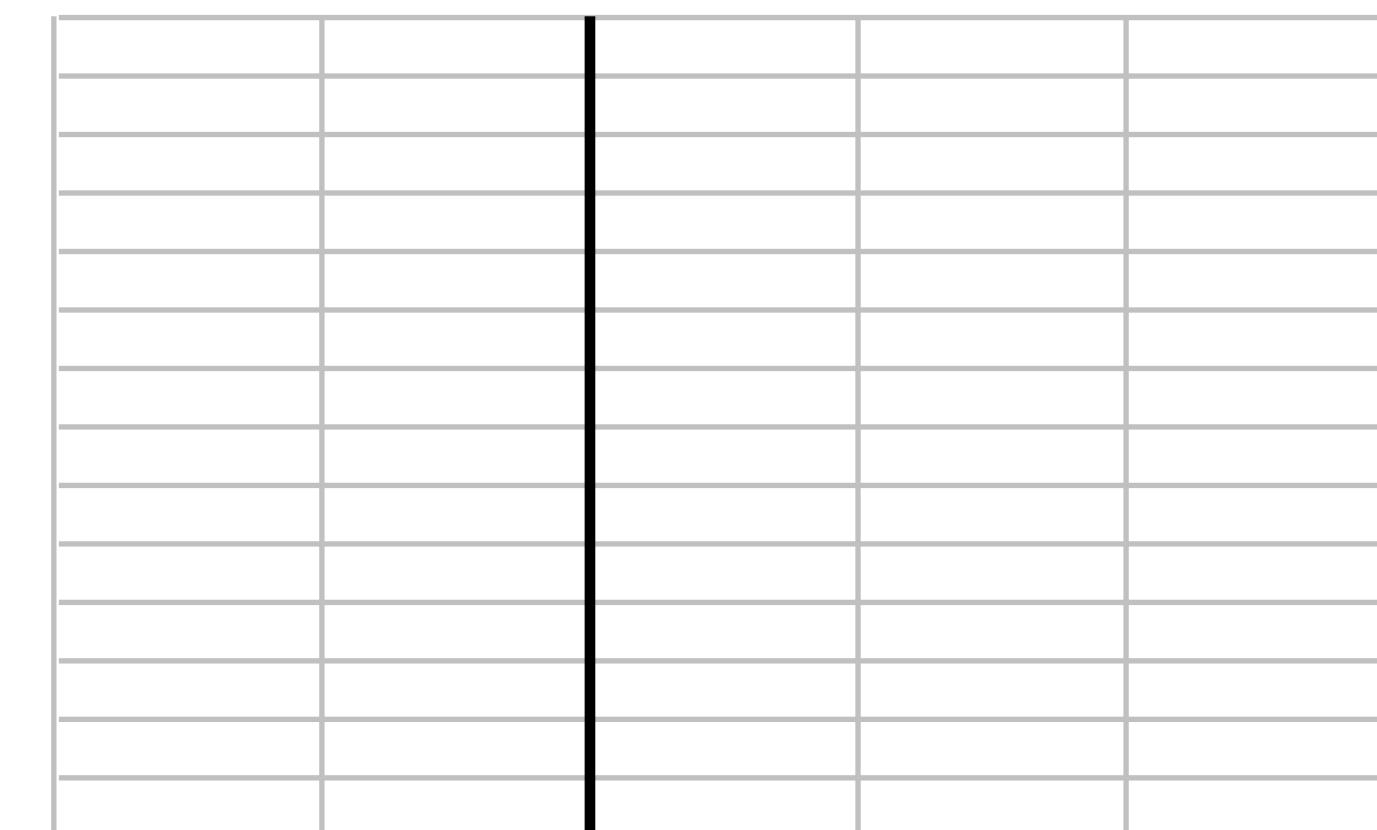
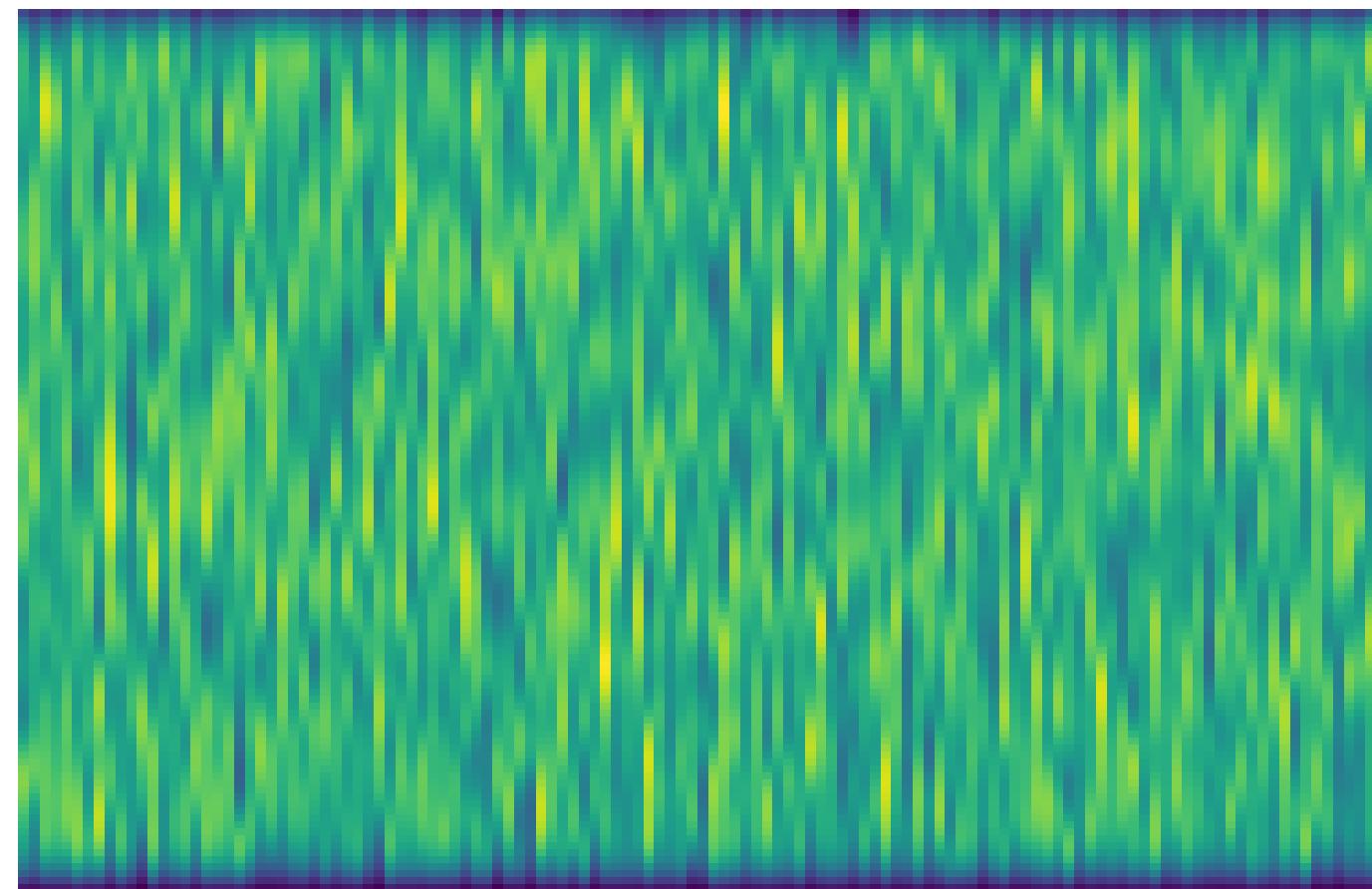
Options for more robust Multigrid

- **Line/plane relaxation**
- For each line, solve

$$A_{1D}v_k = g_k$$

$$v_k = \begin{bmatrix} \vdots \\ v_{k,j-1} \\ v_{k,j} \\ v_{k,j+1} \\ \vdots \end{bmatrix} \quad g_k = \begin{bmatrix} \vdots \\ f_{k,j-1} + \varepsilon(v_{k-1,j-1} + v_{k+1,j-1}) \\ f_{k,j} + \varepsilon(v_{k-1,j} + v_{k+1,j}) \\ f_{k,j+1} + \varepsilon(v_{k-1,j+1} + v_{k+1,j+1}) \\ \vdots \end{bmatrix}$$

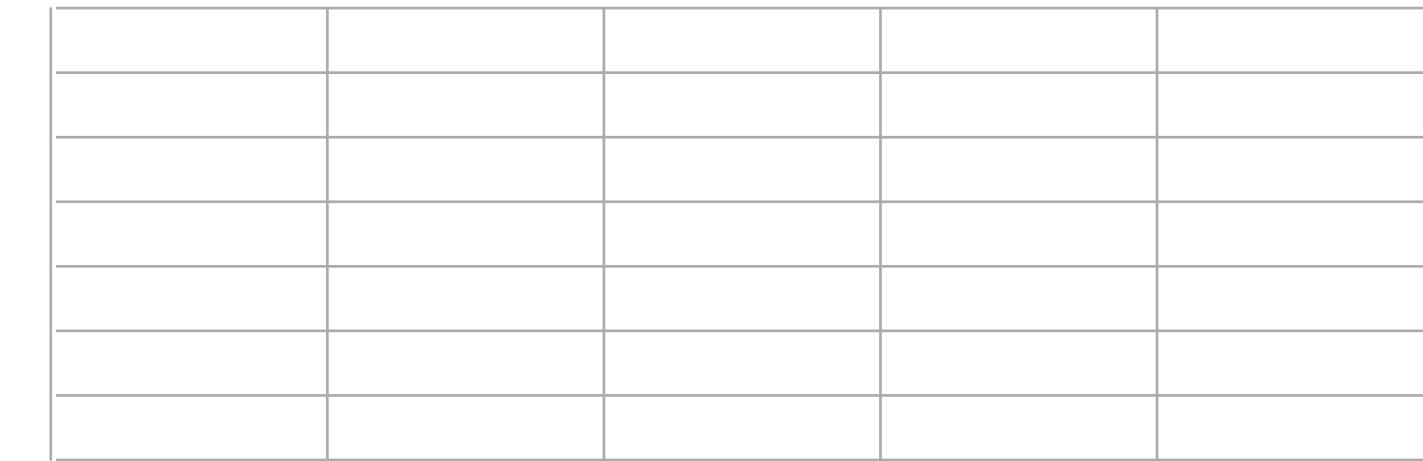
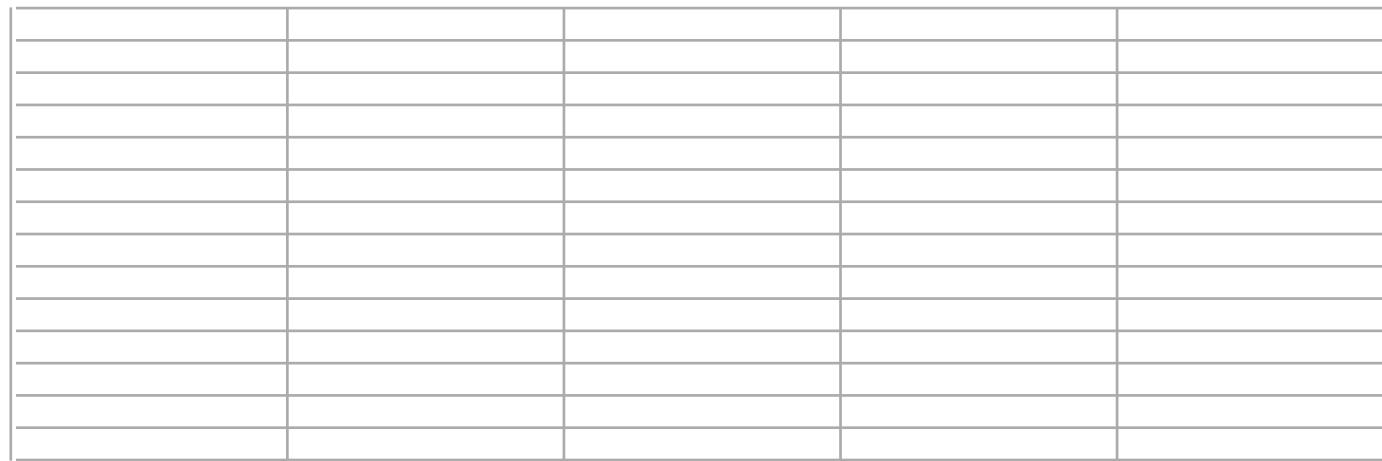
- In 3D, lines become planes...



Options for more robust Multigrid

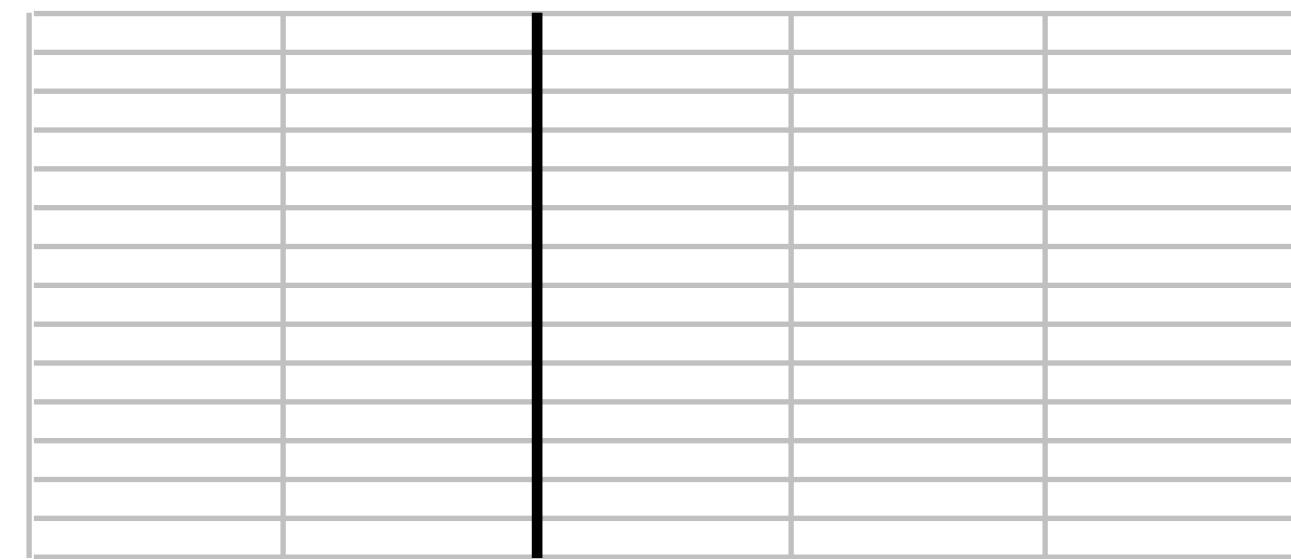
- **Semicoarsening**

Coarsen in the direction of smoothness



- **Line/plane relaxation**

Perform relaxation in groups (in a line)

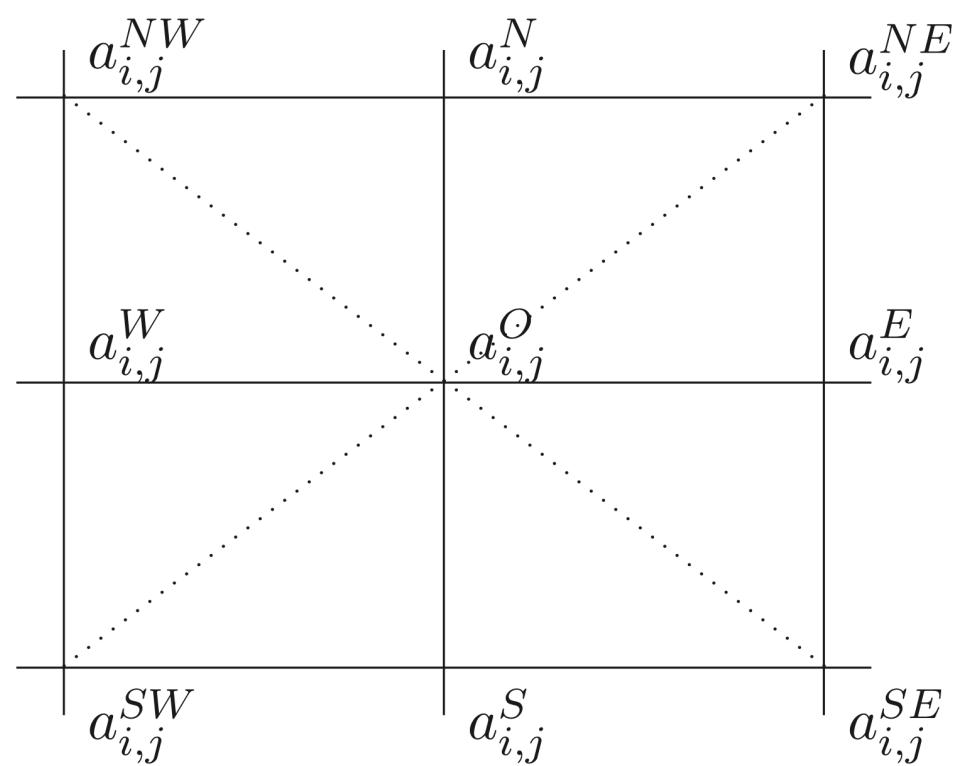


- **Operator based Interpolation** (e.g. BoxMG)
 $Ae = 0$

J. E. Dendy, Black box multigrid, J. Comput. Phys., 1982

J. E. Dendy and J. D. Moulton, Black box multigrid with coarsening by a factor of three, J. Numer. Lin. Alg. App., 2010

- Node (i,j) stencil:

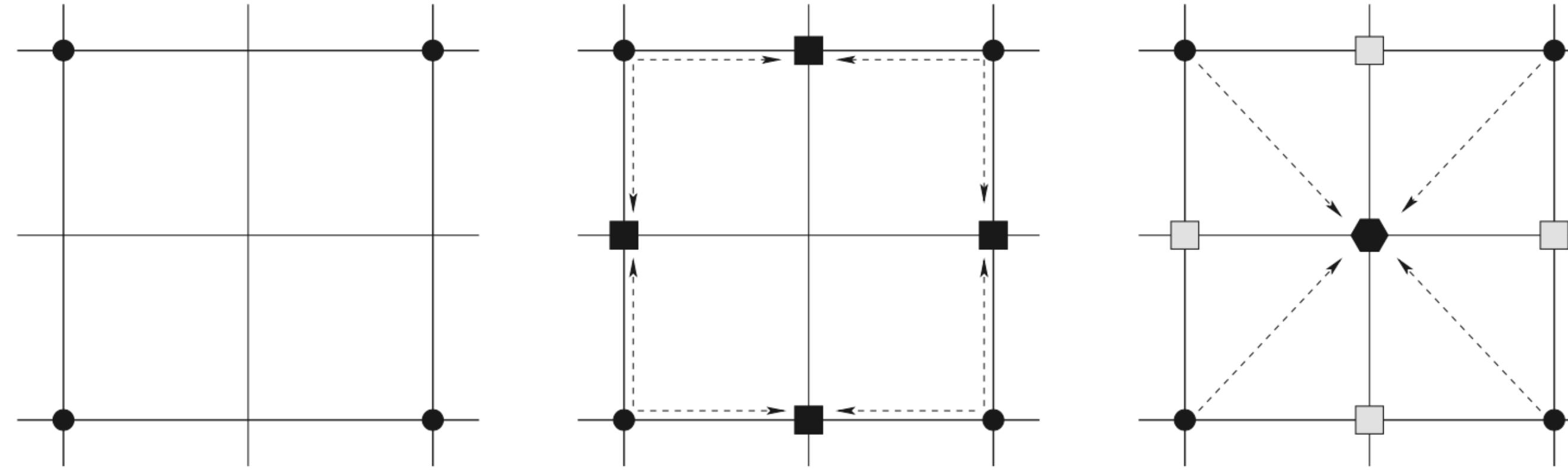


$$Ax = b$$

$$\begin{aligned}
 a_{i,j}^{SW} x_{i-1,j-1} + a_{i,j}^S x_{i,j-1} + a_{i,j}^{SE} x_{i+1,j-1} + a_{i,j}^W x_{i-1,j} + a_{i,j}^O x_{i,j} \\
 + a_{i,j}^E x_{i+1,j} + a_{i,j}^{NW} x_{i-1,j+1} + a_{i,j}^N x_{i,j+1} + a_{i,j}^{NE} x_{i+1,j+1} = b_{i,j}
 \end{aligned}$$

BoxMG

Math and figures from:
[Robust and adaptive multigrid methods: comparing structured and algebraic approaches](#), MacLachlan, Moulton, Chartier



1. Inject coarse points (left)
2. Assume the error is constant along x-lines (and y-lines)
3. Infer interpolation from the edges (right)

Example: Assuming $Ae=0$ and constant y-lines:

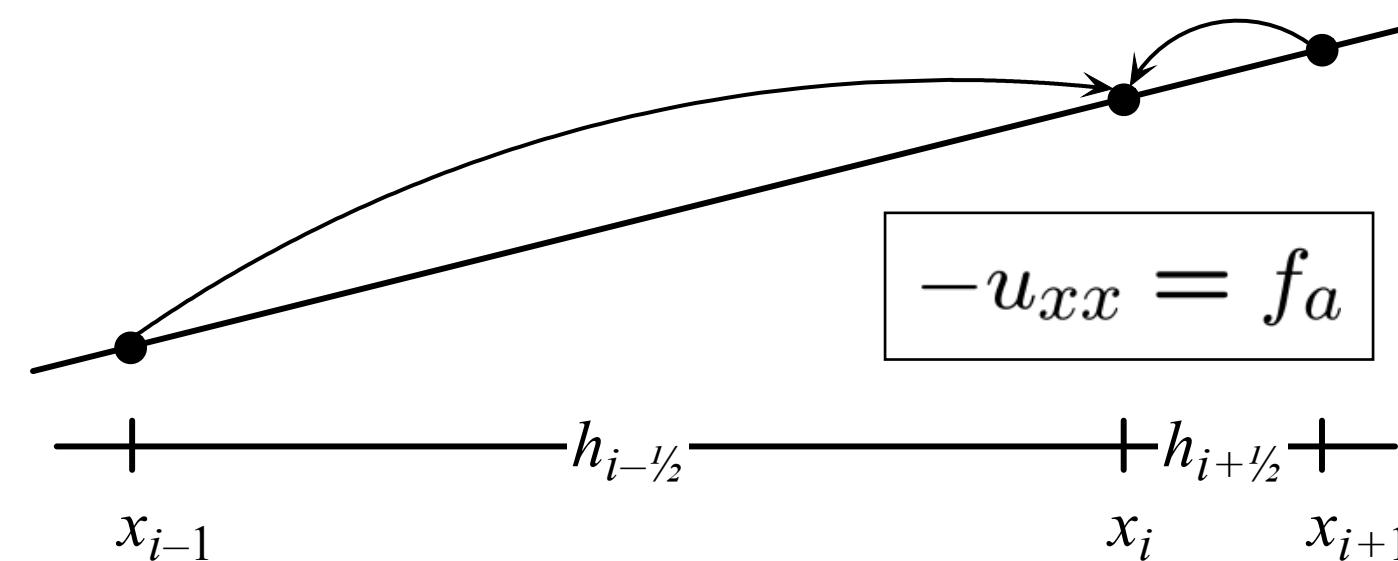
$$(a^S + a^O + a^N)e = -(a^{SW} + a^W + a^{NW})e^c - (a^{SE} + a^E + a^{NW})e^c$$

Interpolation based on entries in A

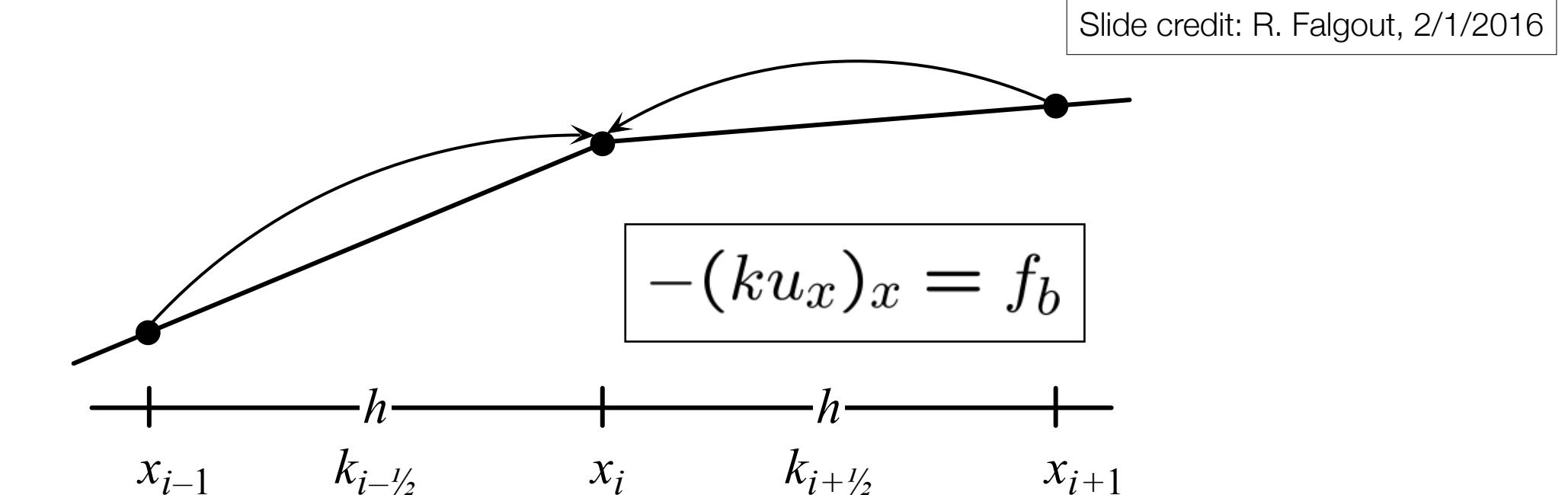
Algebraic Multigrid (AMG) uses matrix coefficients

- Geometric information alone is not sufficient

Linear Interpolation



Operator-Dependent Interpolation

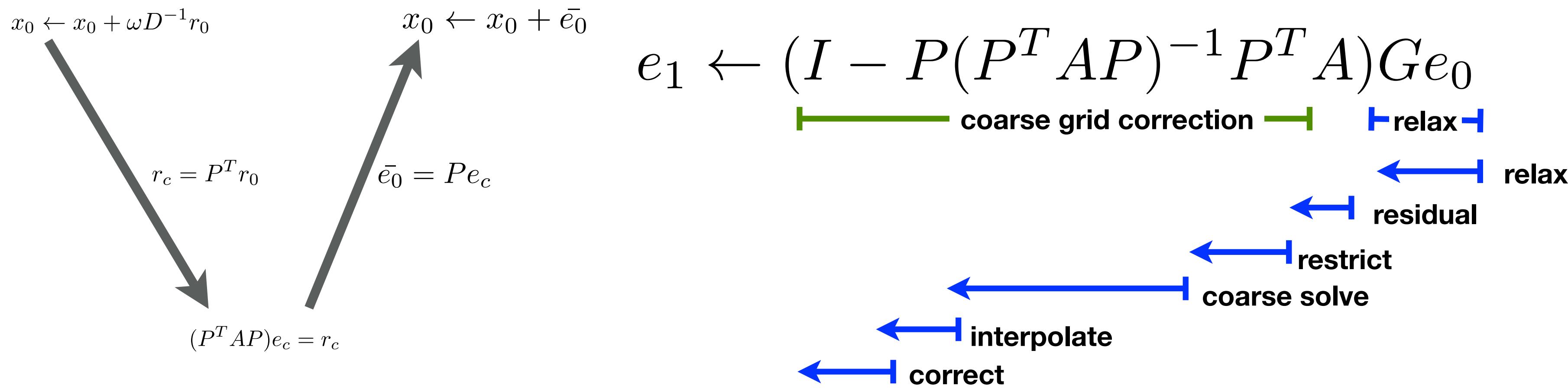


- AMG does not use geometric information, but captures both linear & operator-dependent interpolation

$$(A\mathbf{u})_i = a_{i,i-1}u_{i-1} + a_{i,i}u_i + a_{i,i+1}u_{i+1}$$

$$u_i = \left(-\frac{a_{i,i-1}}{a_{i,i}} \right) u_{i-1} + \left(-\frac{a_{i,i+1}}{a_{i,i}} \right) u_{i+1}$$

Algebraic Observation

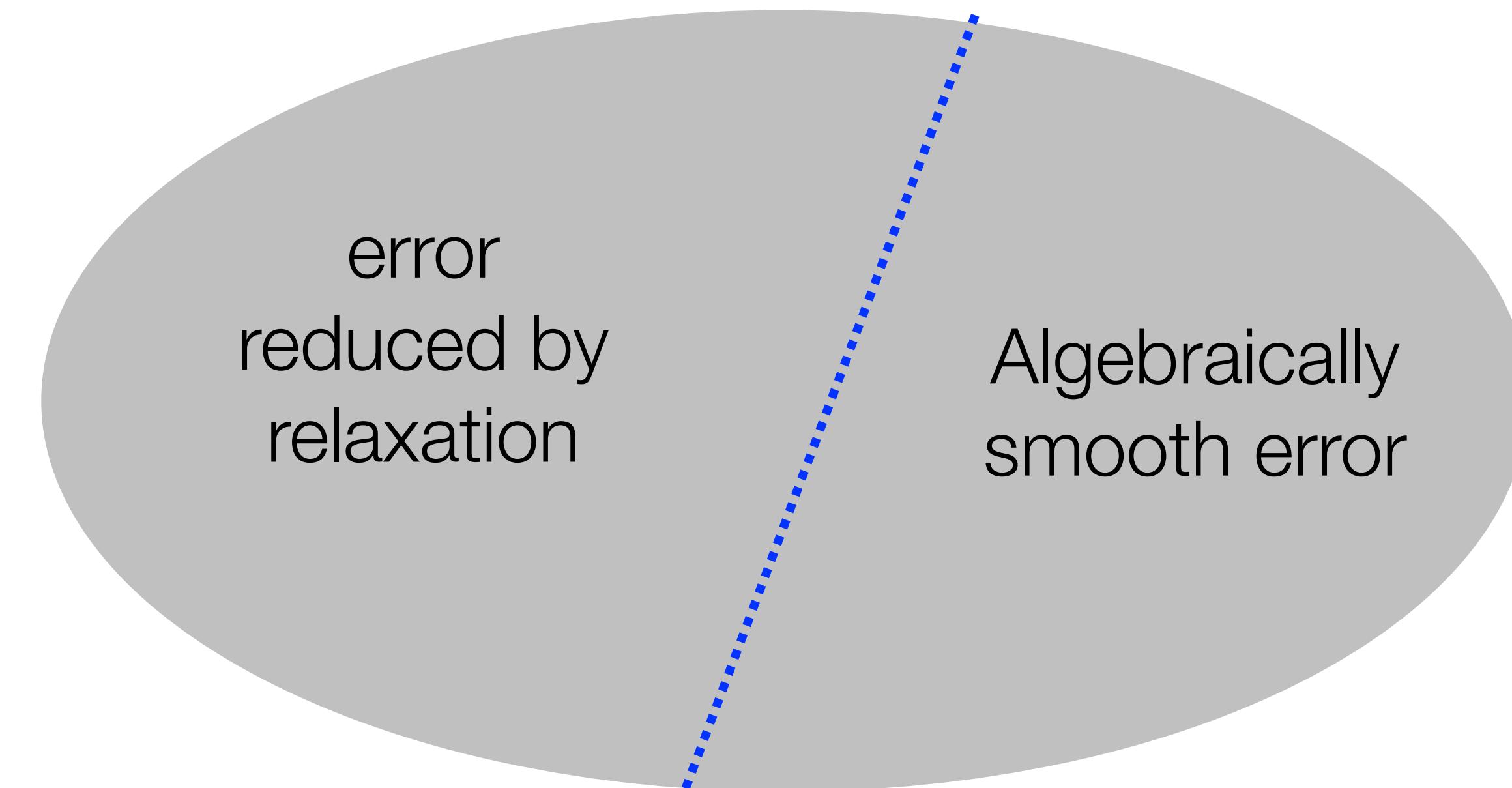


$$G e_0 \in \mathcal{R}(P) \quad \Rightarrow \quad e_1 = 0$$

interpolation should capture what relaxation misses

Algebraically Smooth Error

- “Algebraically smooth” error may not be geometrically smooth

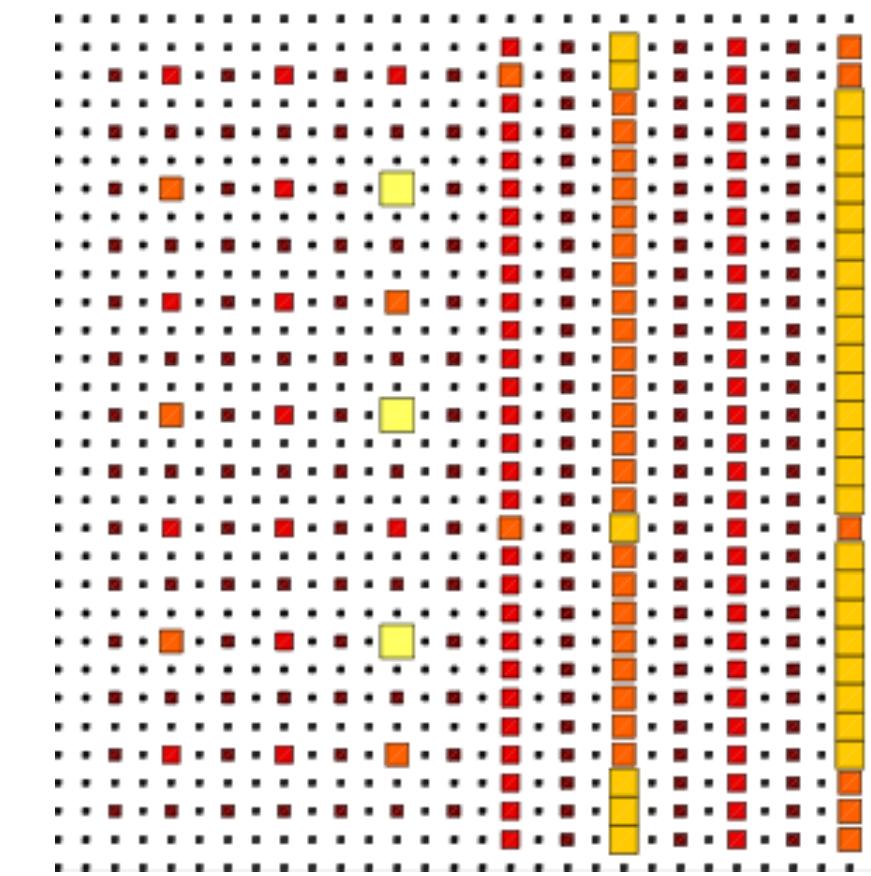
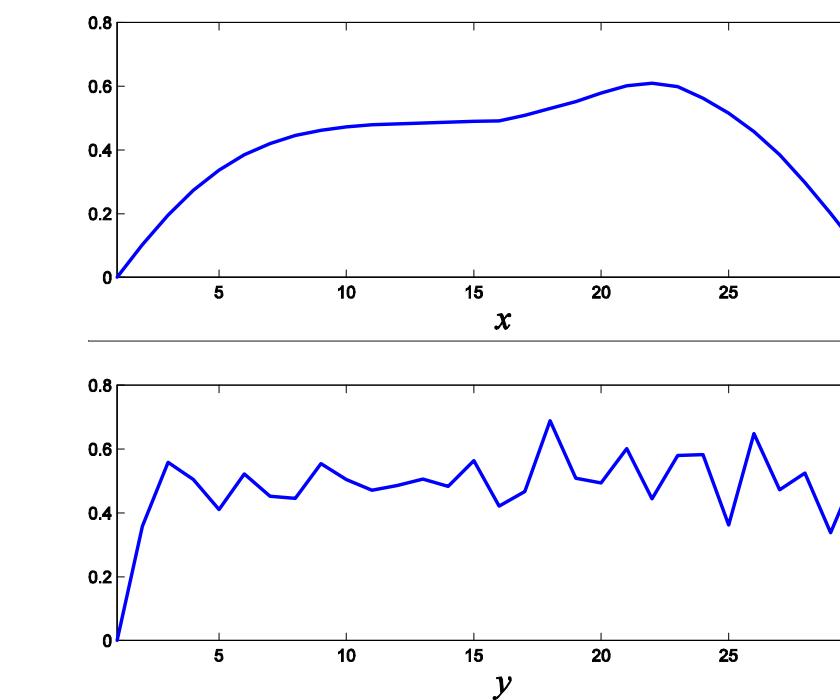
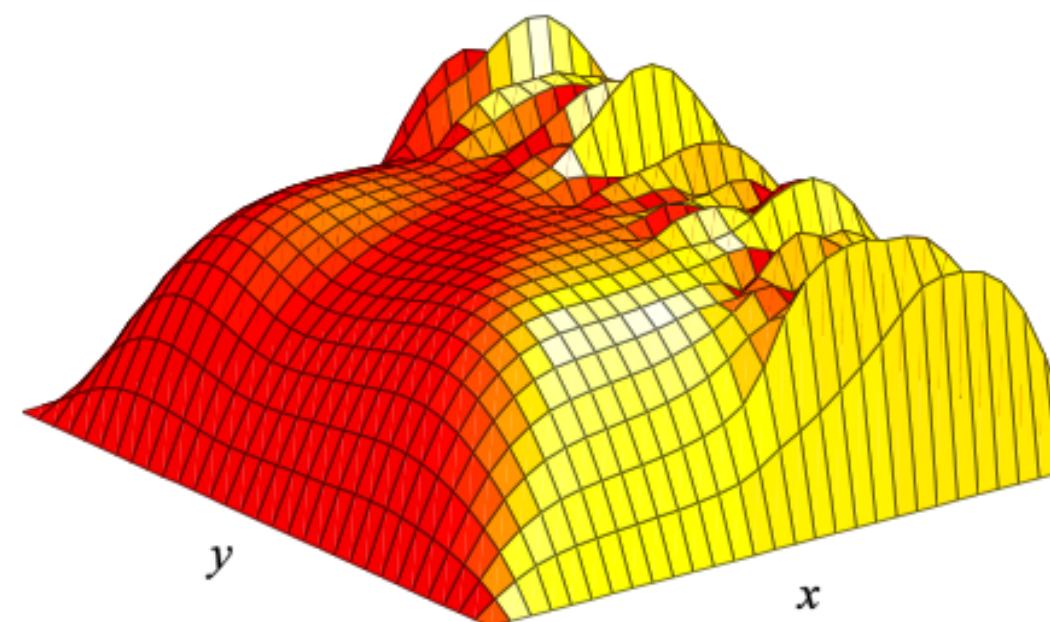


Error left by relaxation can be geometrically oscillatory

- 7 GS sweeps on

$$-au_{xx} - bu_{yy} = f$$

$$\begin{array}{|c|c|} \hline a & b \\ \hline a \gg b & \\ \hline \end{array}$$



Slide credit: R. Falgout, 2/1/2016

- Caution: this example
 - targets **geometric smoothness**
 - uses **pointwise smoothers**

AMG coarsens grids in the direction
of geometric smoothness

Main idea: Algebraically smooth error

- Take a relaxation scheme such as w-Jacobi

$$e \leftarrow (I - M^{-1}A)e$$

- If relaxation stagnates, then the remaining error exhibits poor convergence, so
- Formally (characterized by small eigenvalues)

$$(I - M^{-1}A)e \approx e \Rightarrow M^{-1}Ae \approx 0 \Rightarrow r \approx 0$$

$$\langle Ae, e \rangle \ll 1$$

Main idea: Algebraically smooth error

- We then have

$$\begin{aligned}\langle Ae, e \rangle &= \sum_i e_i (A_{ii}e_i + \sum_{j \neq i} A_{ij}e_j) && \text{assume zero row sum} \\ &= \sum_i e_i \left(\sum_{j \neq i} -A_{ij}(e_i - e_j) \right) \\ &= \sum_{i < j} -A_{ij} \cdot e_i \cdot (e_i - e_j) + \sum_{i > j} -A_{ij} \cdot e_i \cdot (e_i - e_j) && \text{swap } i, j \\ &= \sum_{i < j} -A_{ij} \cdot e_i \cdot (e_i - e_j) - \sum_{i < j} -A_{ij} \cdot e_i \cdot (e_i - e_j) \\ &= \sum_{i < j} -A_{ij} \cdot (e_i - e_j)^2\end{aligned}$$

- Ok, so smooth error varies **slowly** in the direction of large matrix coefficients

Main idea: Algebraically smooth error

- We have assumed **geometric** smoothness to show

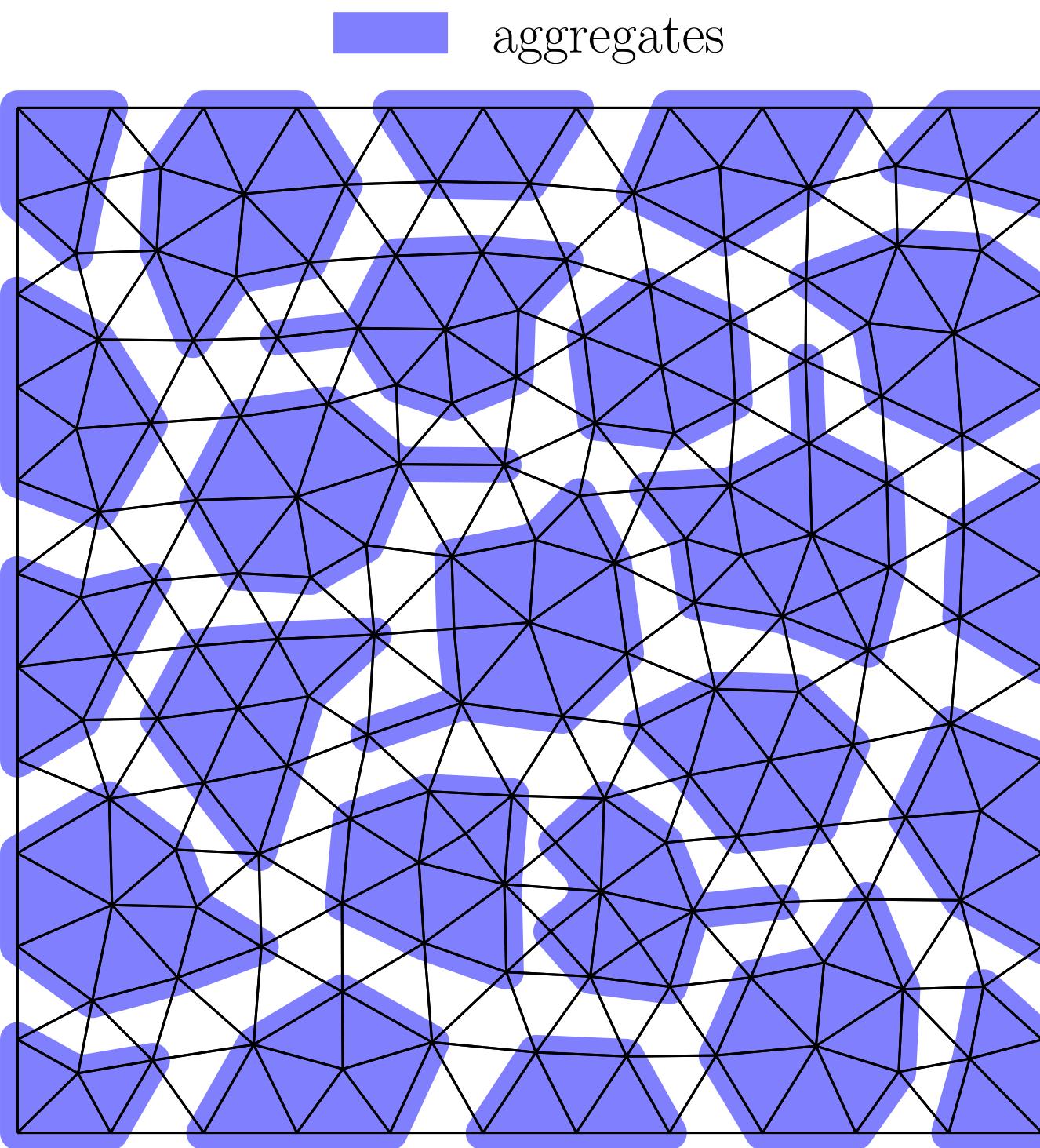
$$\mathbf{e}^T A \mathbf{e} = \sum_{i < j} (-a_{ij})(e_i - e_j)^2 \ll 1$$

- **CF AMG:** Smooth error varies slowly in the direction of “large” matrix coefficients
- **Strength of connection:** Given a threshold $0 < \theta \leq 1$, we say that variable u_i strongly depends on variable u_j if

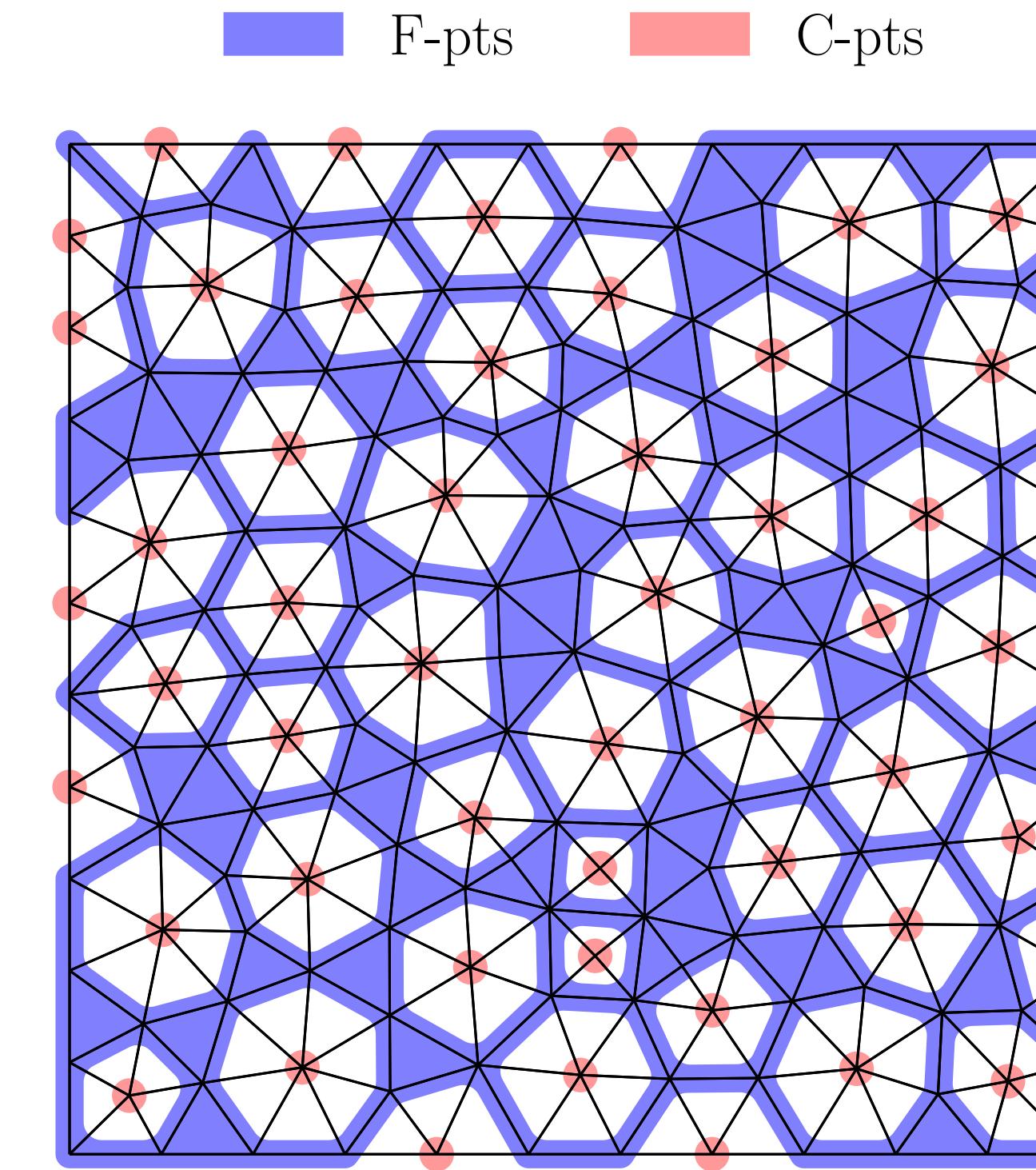
$$-a_{ij} \geq \theta \max_{k \neq i} \{-a_{ik}\}$$

- Often positive off-diagonals are treated as **weak**
- This definition of strength of connection is not symmetric

Two (general) forms of AMG



- Smoothed Aggregation AMG (SA-AMG)
- Interpolation constructed from candidate vectors
- Clear approach to *optimize* interpolation



- Coarse-Fine AMG (CF-AMG) or Ruge-Stüben
- Coarse grid points are a subset of the fine grid points
- Edge-wise construction of interpolation, allowing straightforward control of sparsity
- Incorporating near-nullspace is not straightforward