

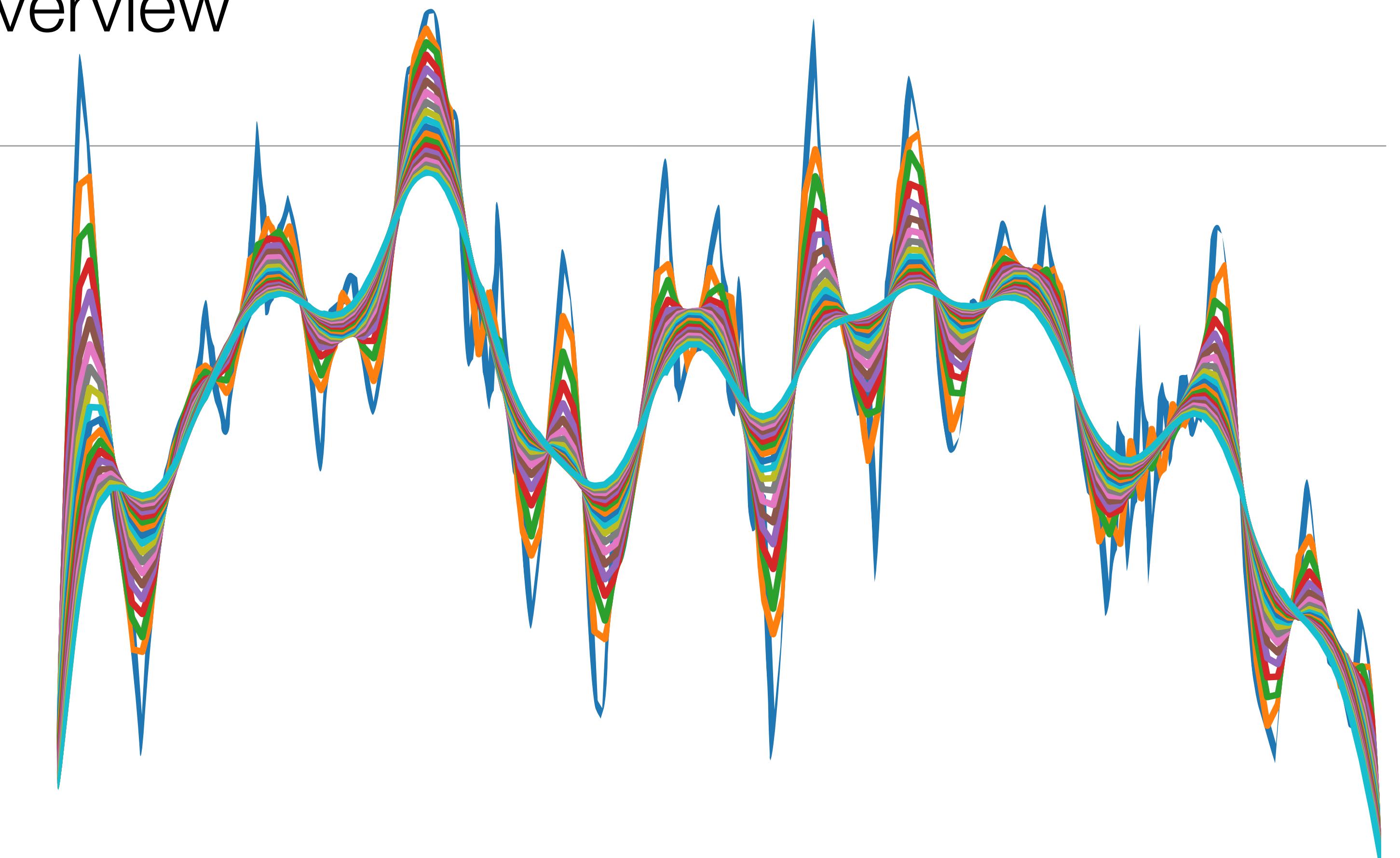
# Multigrid Methods – An Overview

## Lecture 1: Basics

---

Scott MacLachlan  
Department of Mathematics and Statistics  
Memorial University of Newfoundland

*Credit to:*  
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 – high level

---

## **1. Lecture 1 - Basics**

- Basic mechanics of a multigrid method
- 1D, 2D, Poisson

## **2. Lecture 2 - Extensions**

- What can go wrong and how to fix it

## **3. Lecture 3 - Multigrid for Systems**

- How to extend these ideas to coupled systems

## **4. Lecture 4 - Algebraic**

- What to do if we do not have a grid (hierarchy)

# Objectives – this “morning”

---

## **1. Lecture 1 - Basics**

- Create a two-level multigrid method
- Illustrate the main components of a multigrid method
- Calculate the effectiveness of a multigrid method

# Iterative Methods

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

## An alternative – 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

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

## An alternative – projection methods

---

- Instead, look at the A-norm:

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

- 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

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

## An alternative – 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

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

- Model problem matrix

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

# Model Problem

- A special matrix:

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

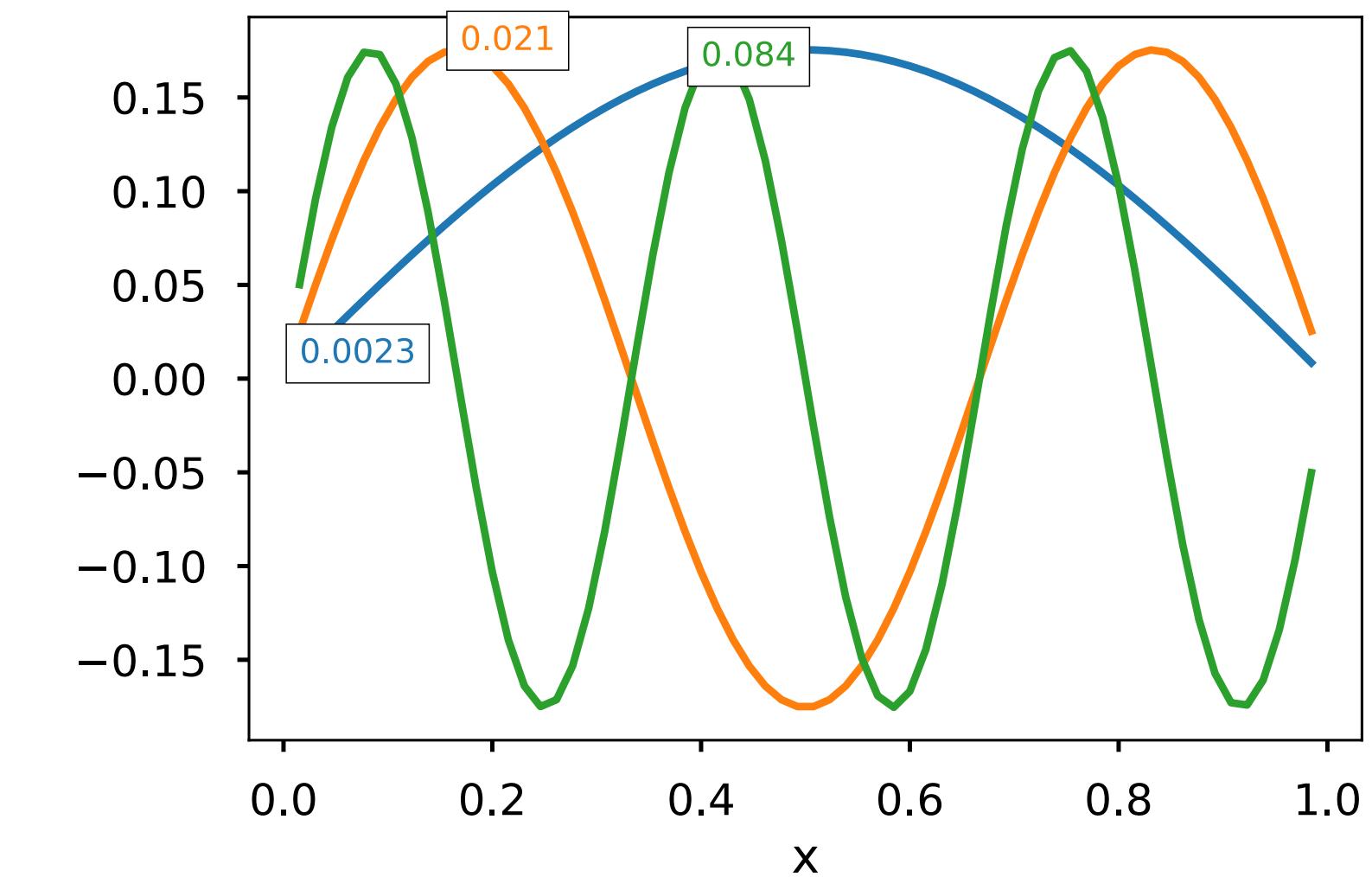
The eigenvalues

range from (0,4]

(or from  $h^2$  to 4)

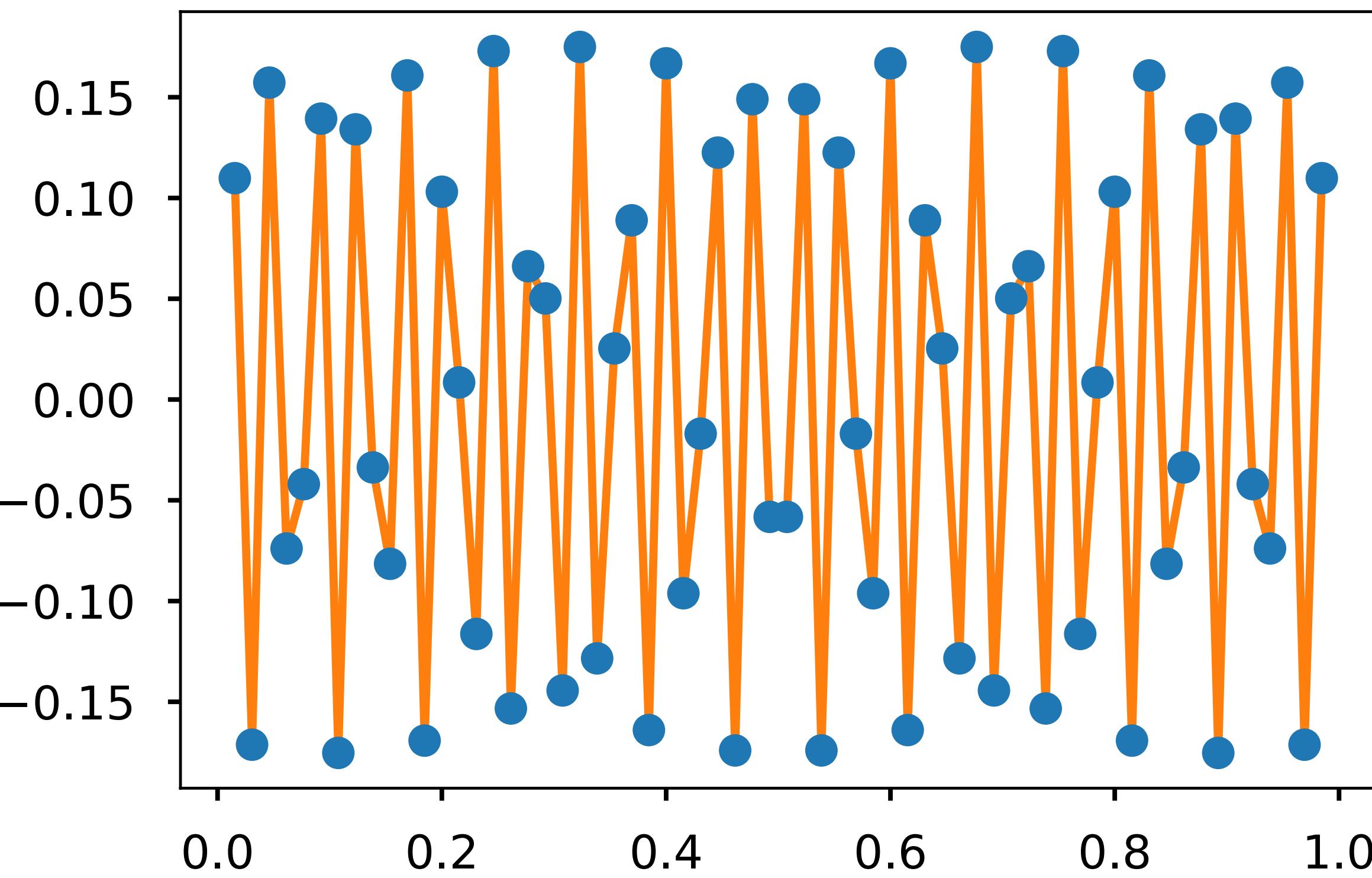
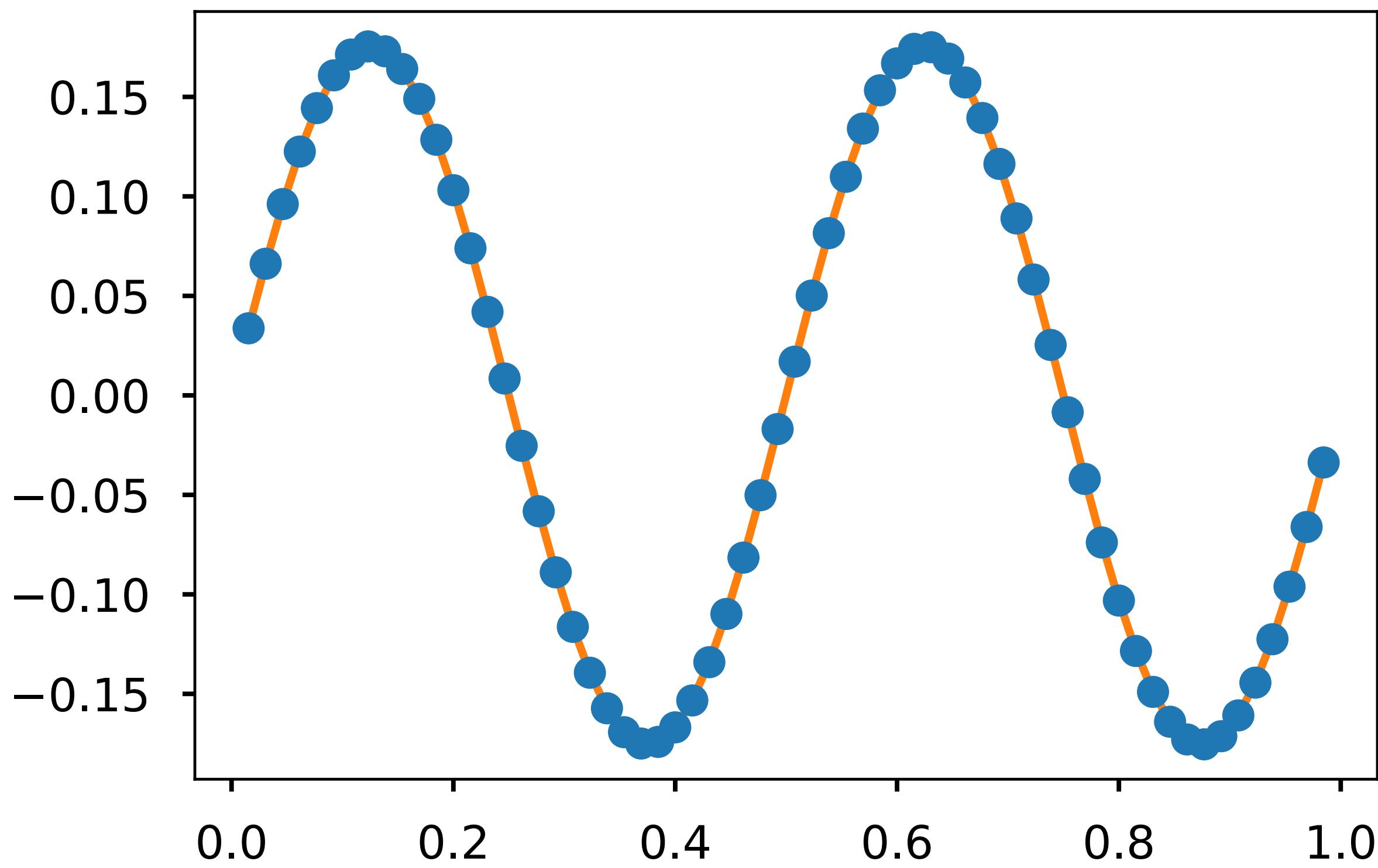
The eigenvectors are Fourier modes:

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



# 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:

$$e \leftarrow Ge$$

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

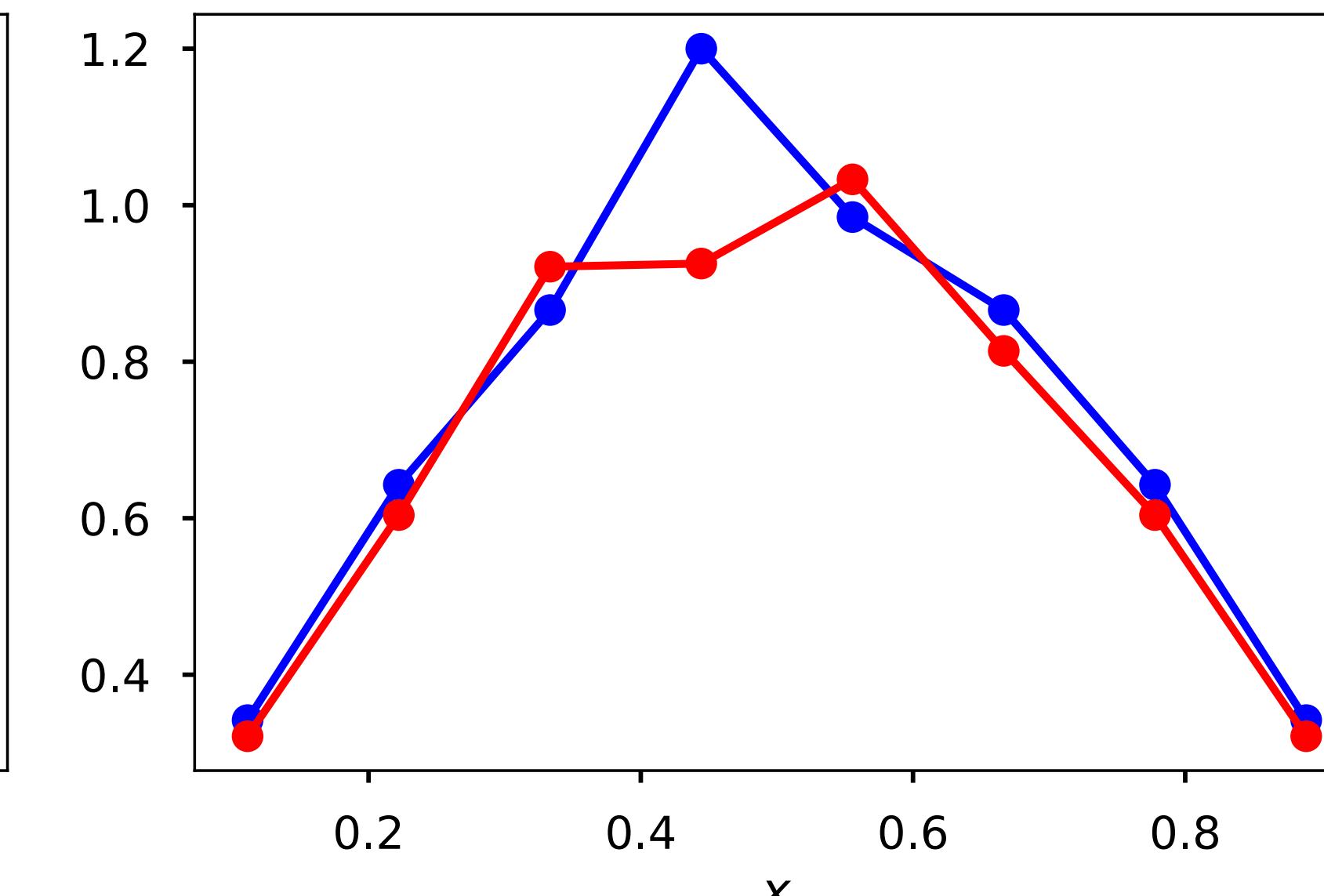
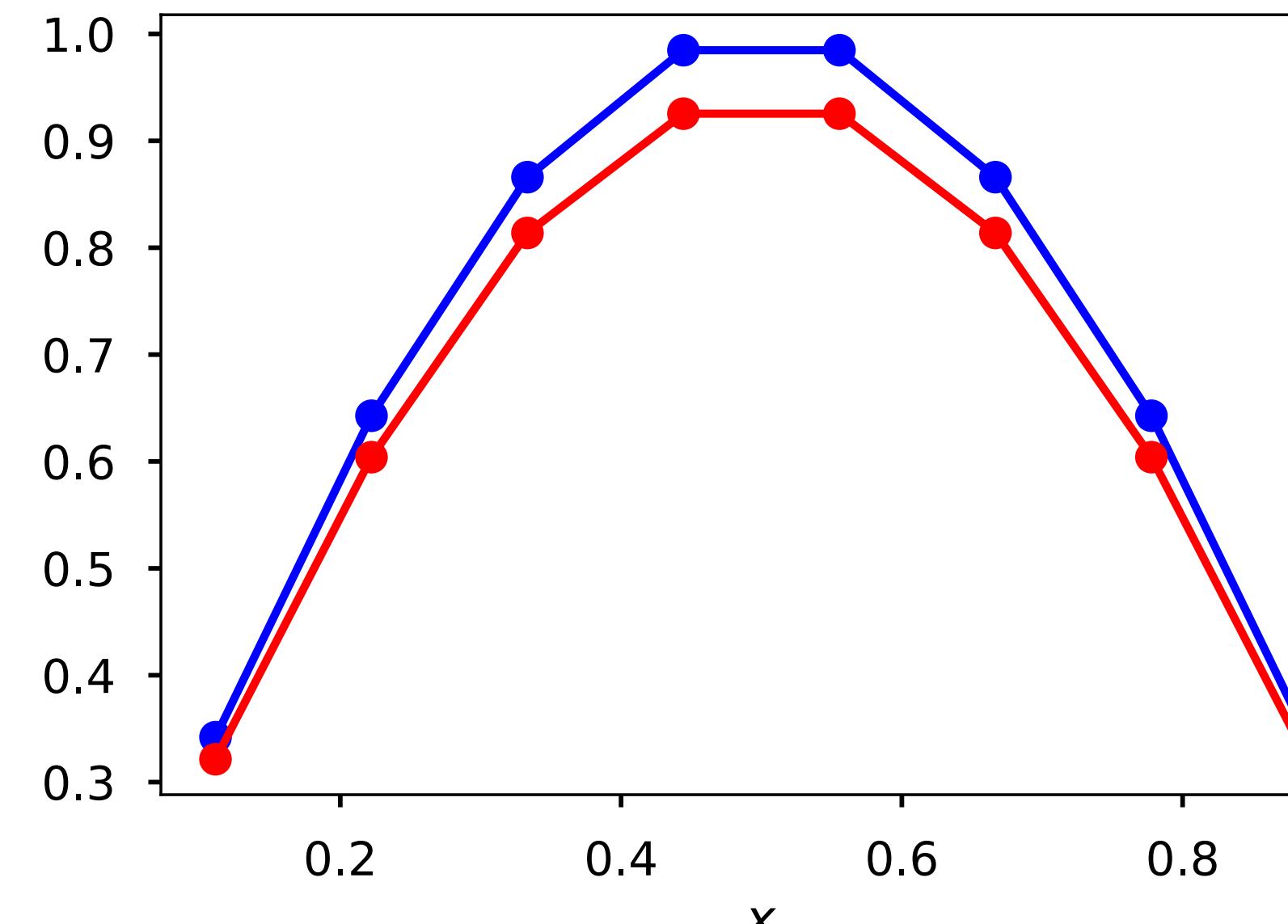
The “error propagation operator”

# What does Jacobi do to error?

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

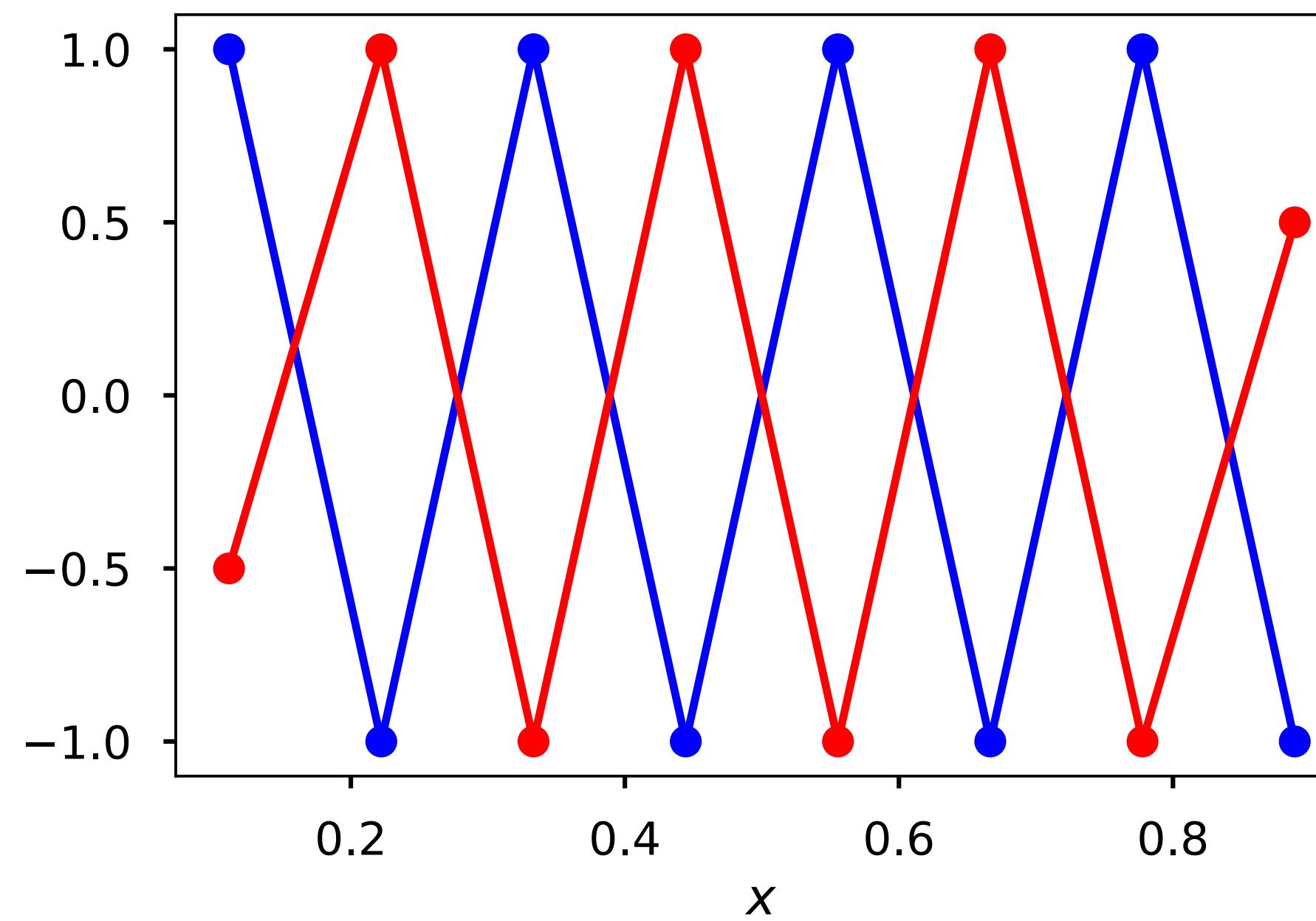
$$e_i^{new} \leftarrow \frac{1}{2} (e_{i-1}^{old} + e_{i+1}^{old})$$

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



But...

---

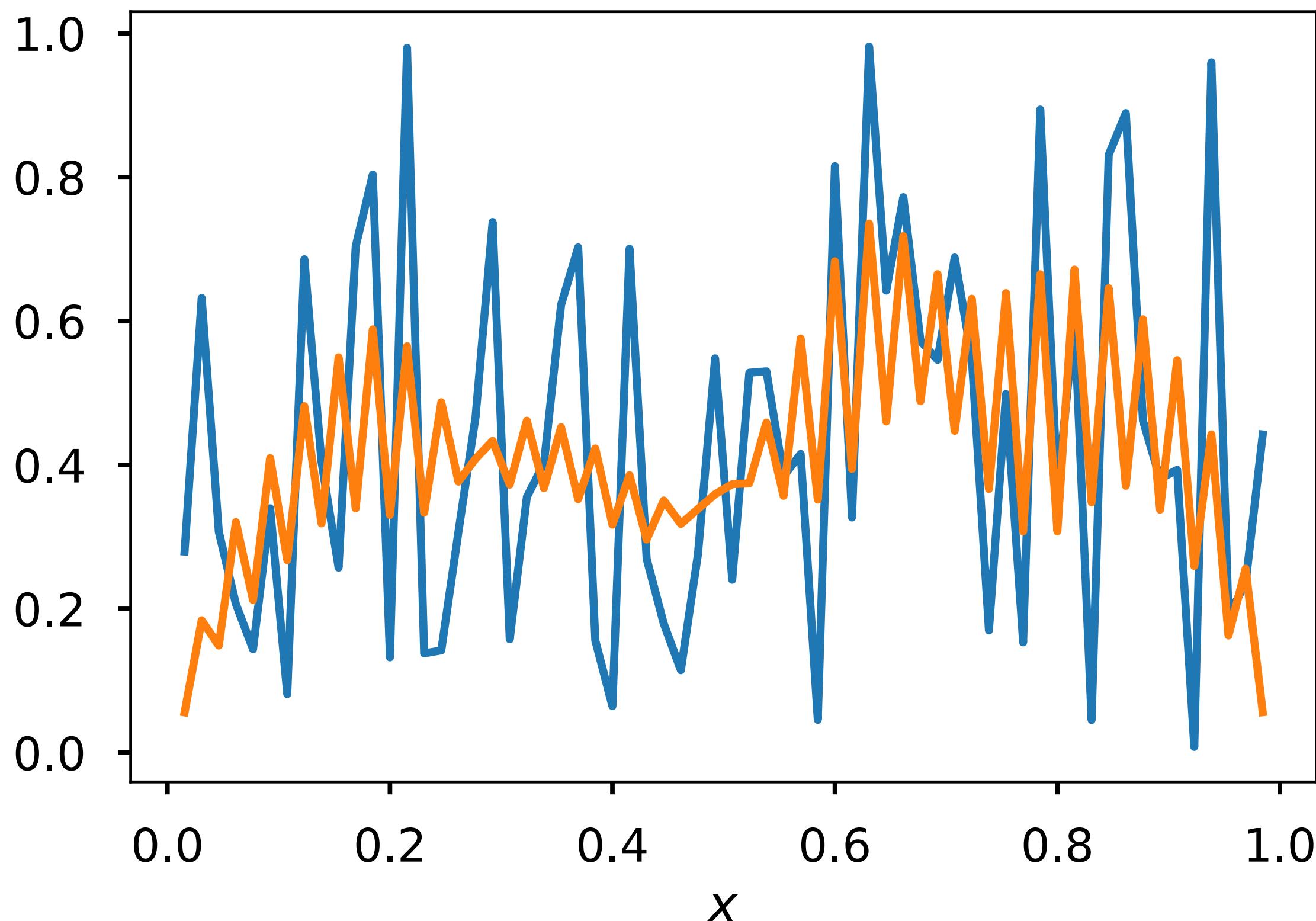


- Jacobi “averages out” certain error very quickly.
- But stagnates on very smooth error or very oscillatory error

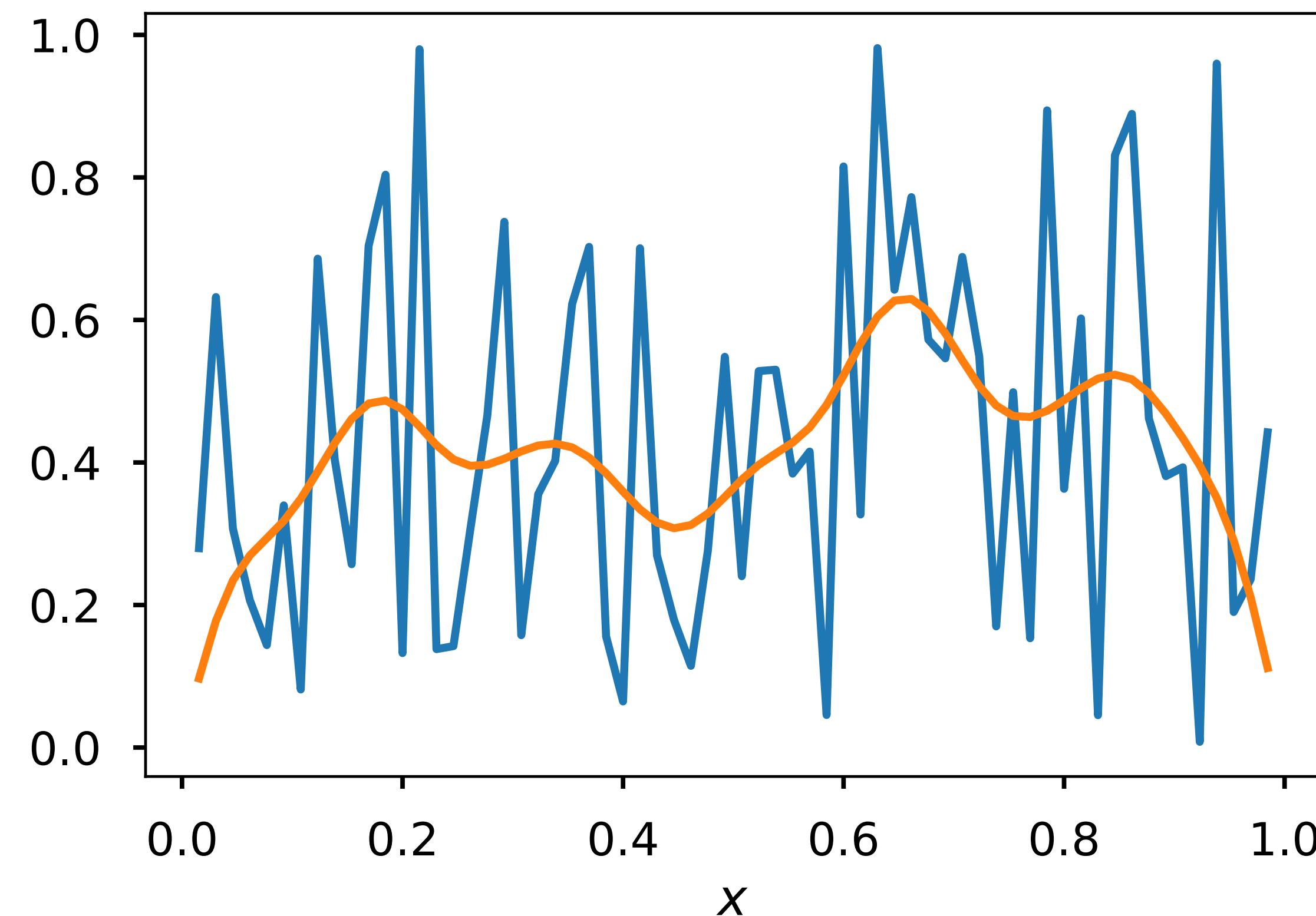
# From Jacobi to weighted-Jacobi

- Random initial guess (random error):

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



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



# Weighted Jacobi

---

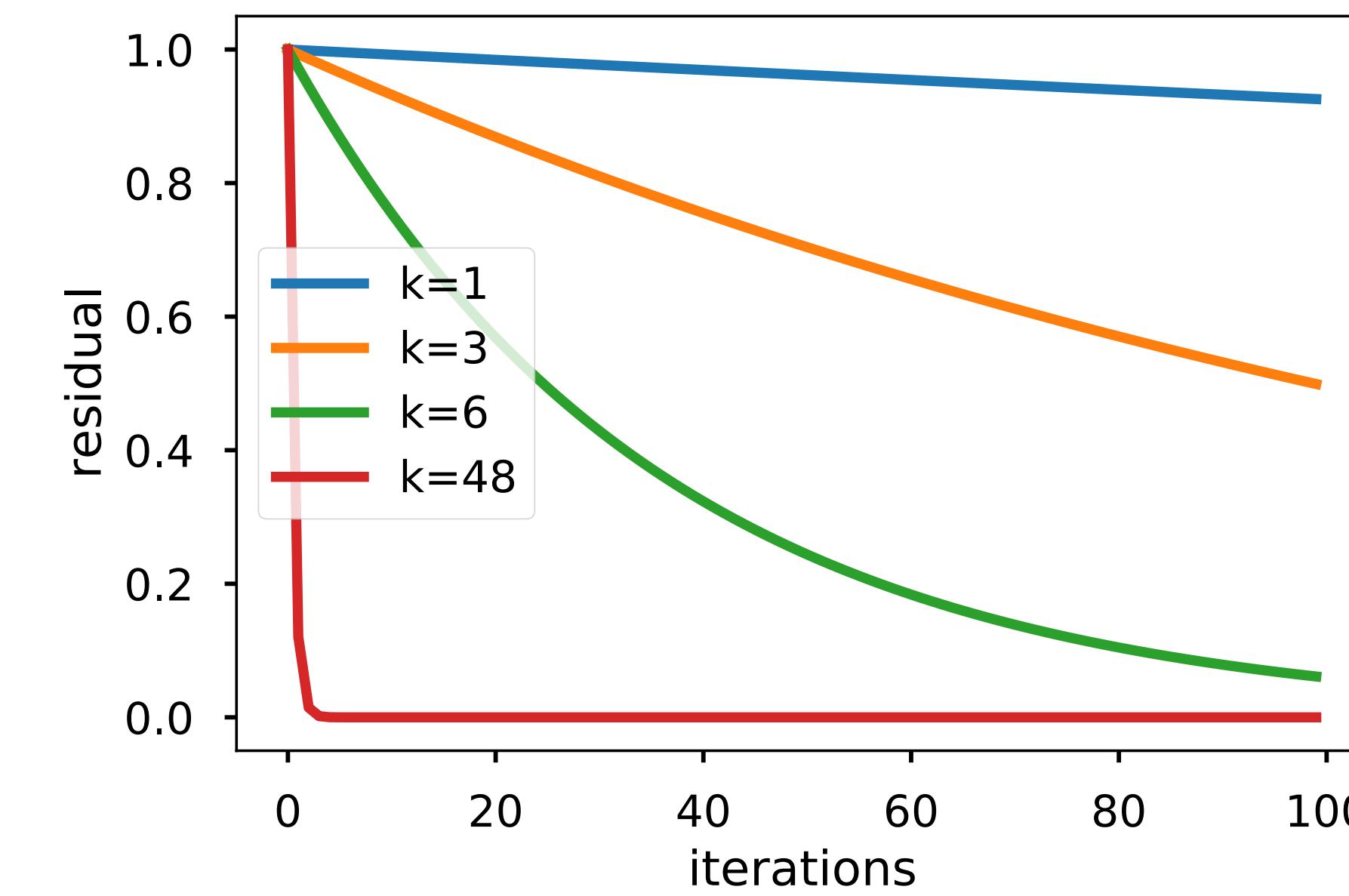
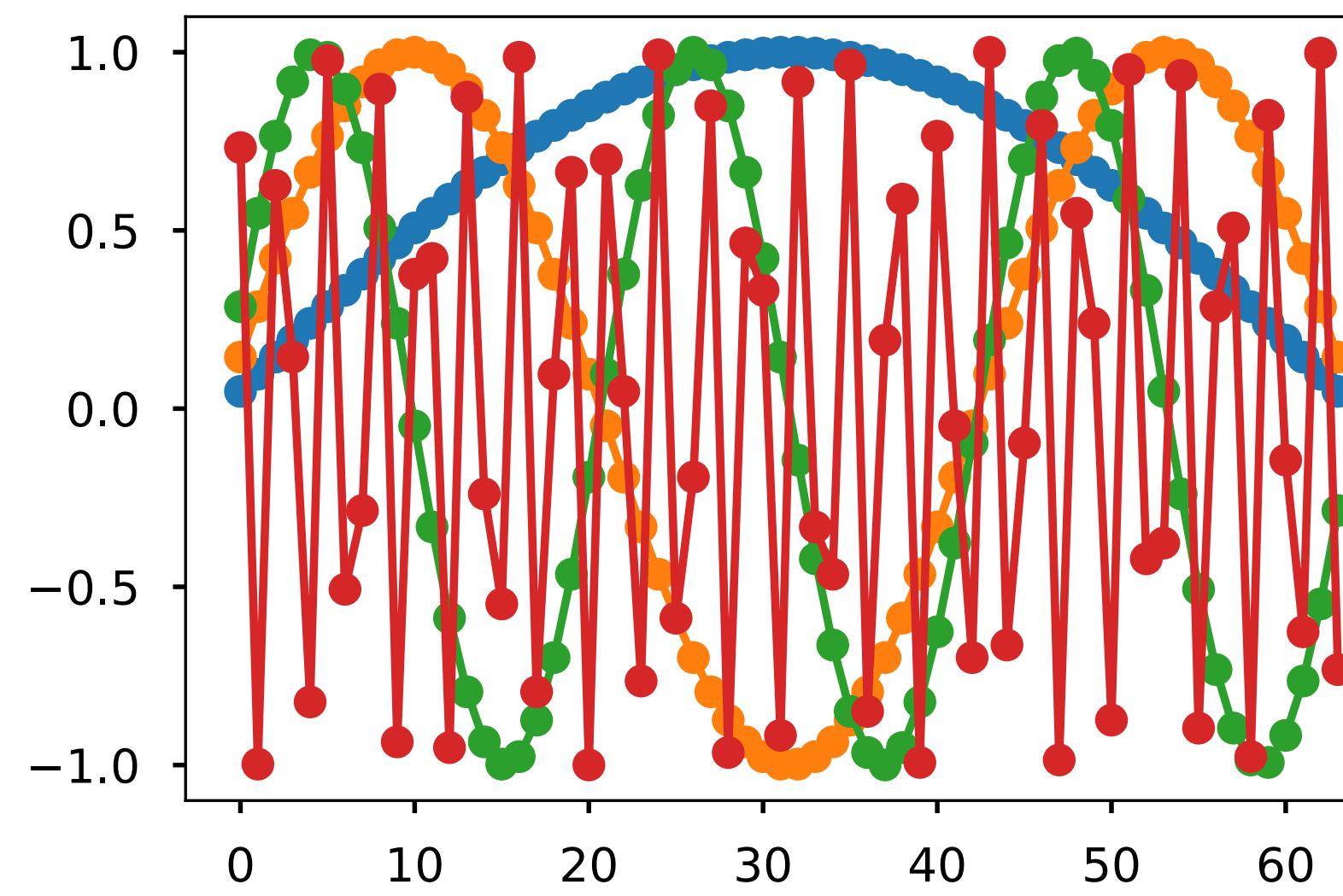
- Weighted Jacobi

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

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

# Weighted Jacobi

- If we pick 4 modes, 1, 3, 6, and 48
- Then smooth modes are still damped 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  $\nu$  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  $\lambda_k$  in every pass

# Fundamental Theorem of Iteration

---

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

- Convergent ( $G^n \rightarrow 0$ ) if and only if  $\rho(G) \leq 1$
- Suppose  $A$  is s.p.d. 
$$\frac{\|e_n\|}{\|e_0\|} \leq \|G^n\| \leq \|G\|^n \approx 10^{-d}$$
- 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

---

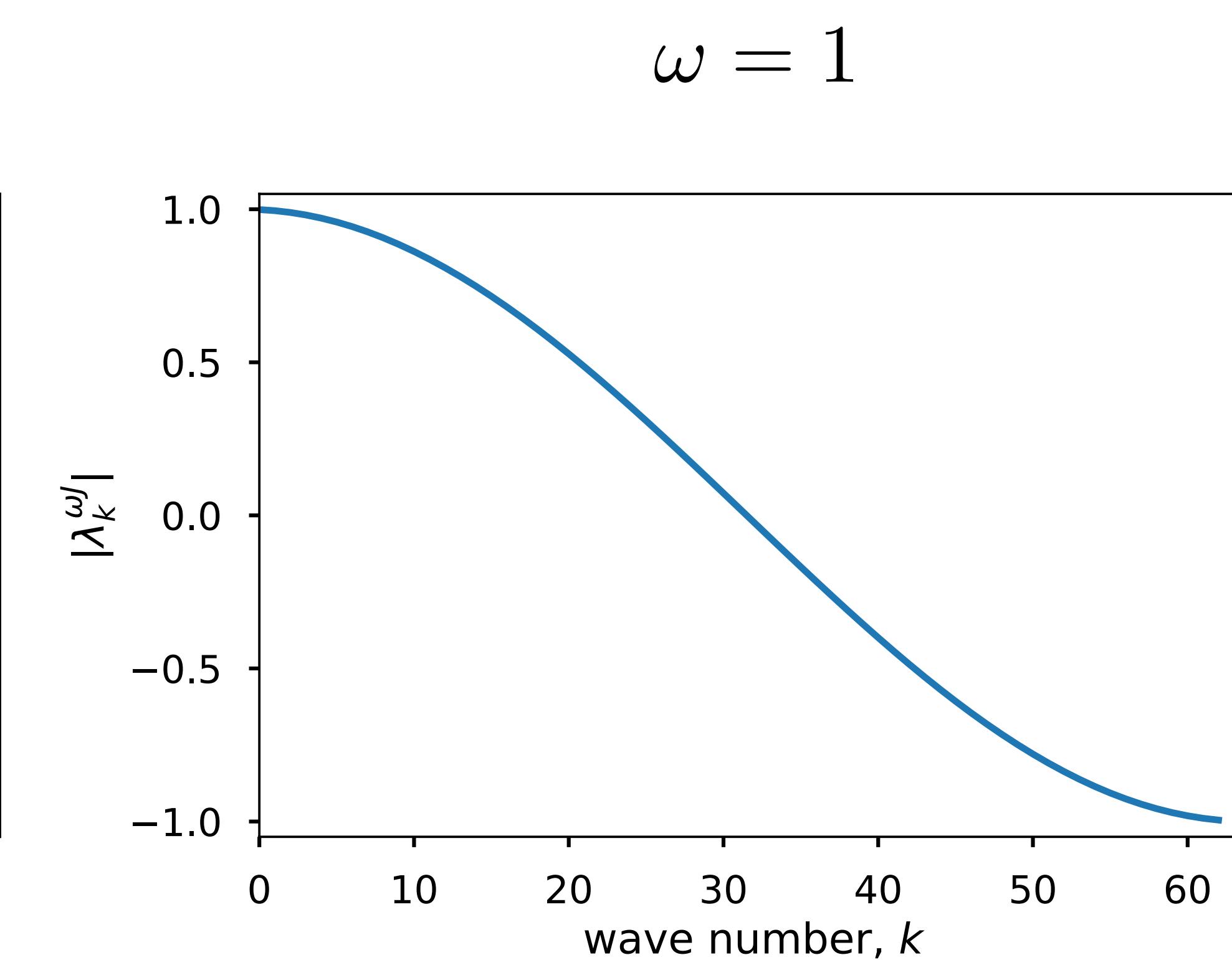
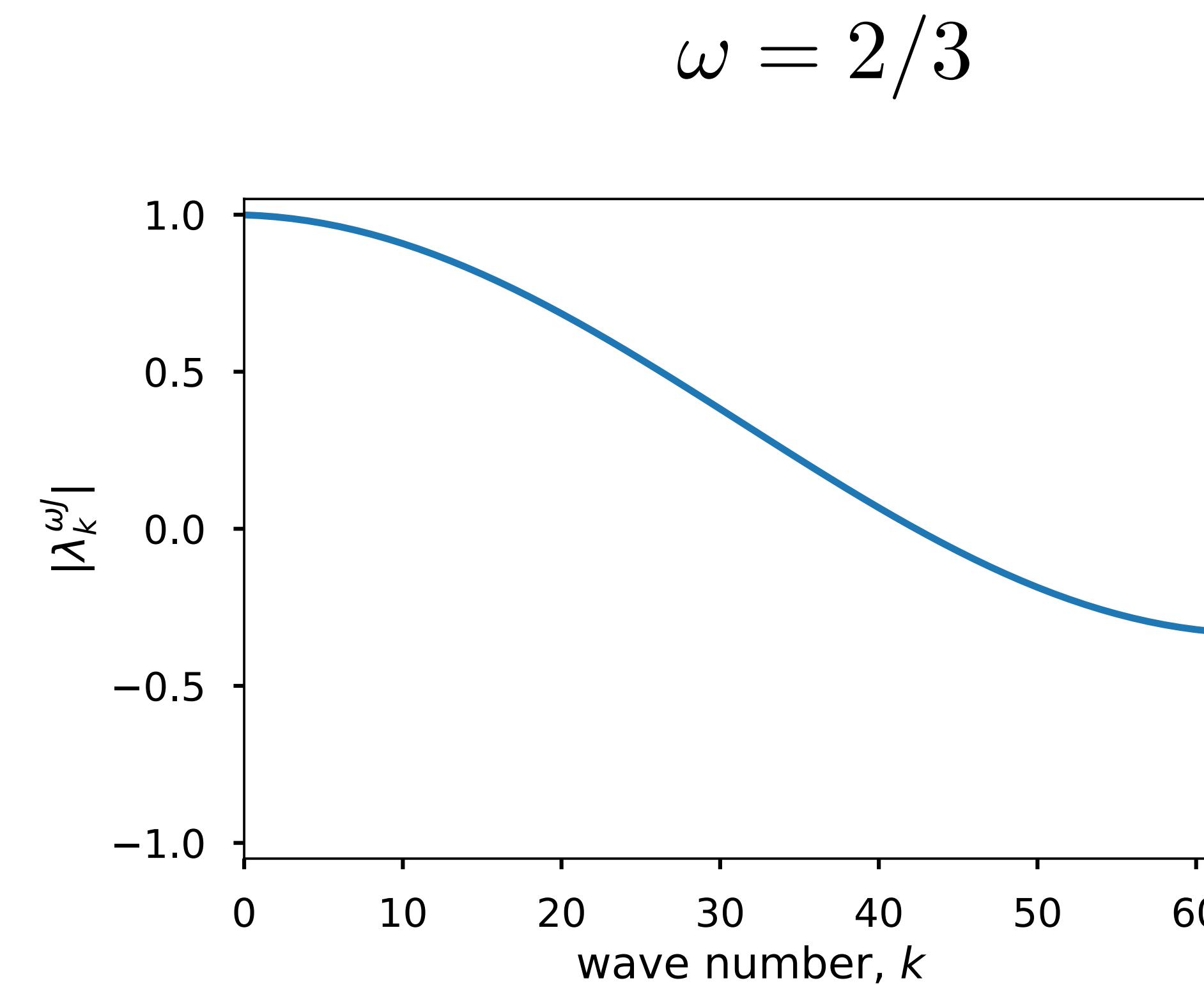
- Convergence **factor**  $\|G\| \text{ or } \rho(G)$
- Convergence **rate**  $-\log_{10} \rho(G)$
- For Jacobi:
$$G = I - D^{-1}A \longrightarrow \lambda_k = 1 - \frac{1}{2} \cdot 4 \cdot \sin^2 \left( \frac{k\pi}{2(n+1)} \right)$$
$$G = I - (2/3)D^{-1}A \longrightarrow \lambda_k = 1 - \frac{2}{3} \cdot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

The spectral radius is the same, but...

# Weighted Jacobi

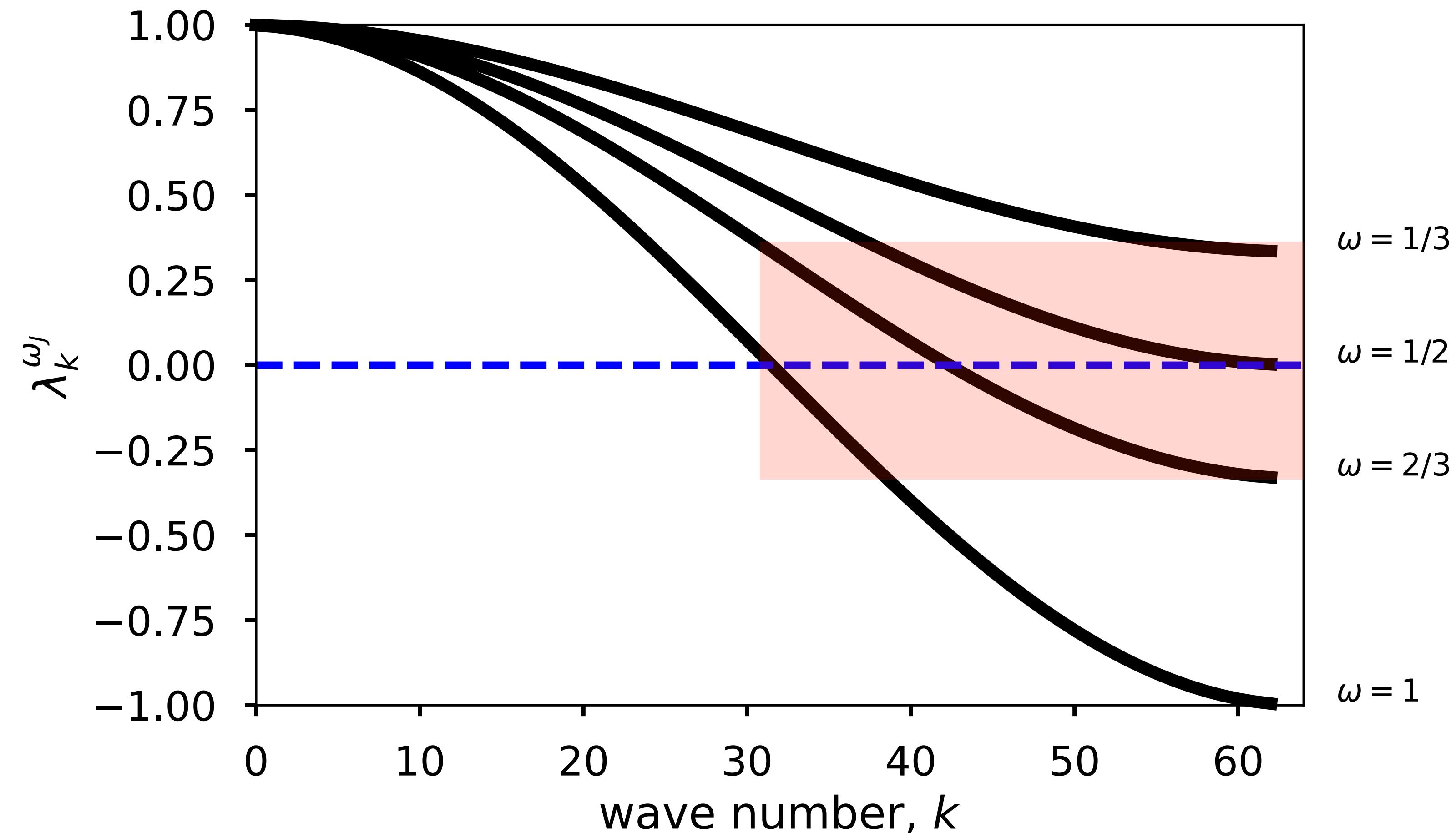
---

- 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 common feature:

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

## Multigrid Step #1: pick a relaxation/smoothen

---

- 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

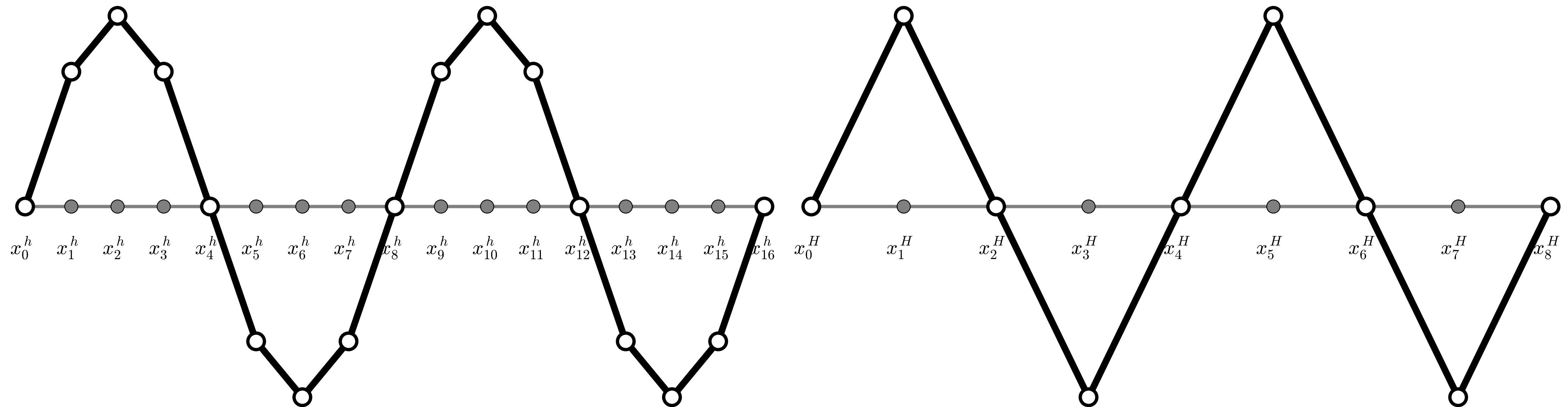
$$u \leftarrow u + \omega D^{-1}r$$

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

# Coarse Grids

---

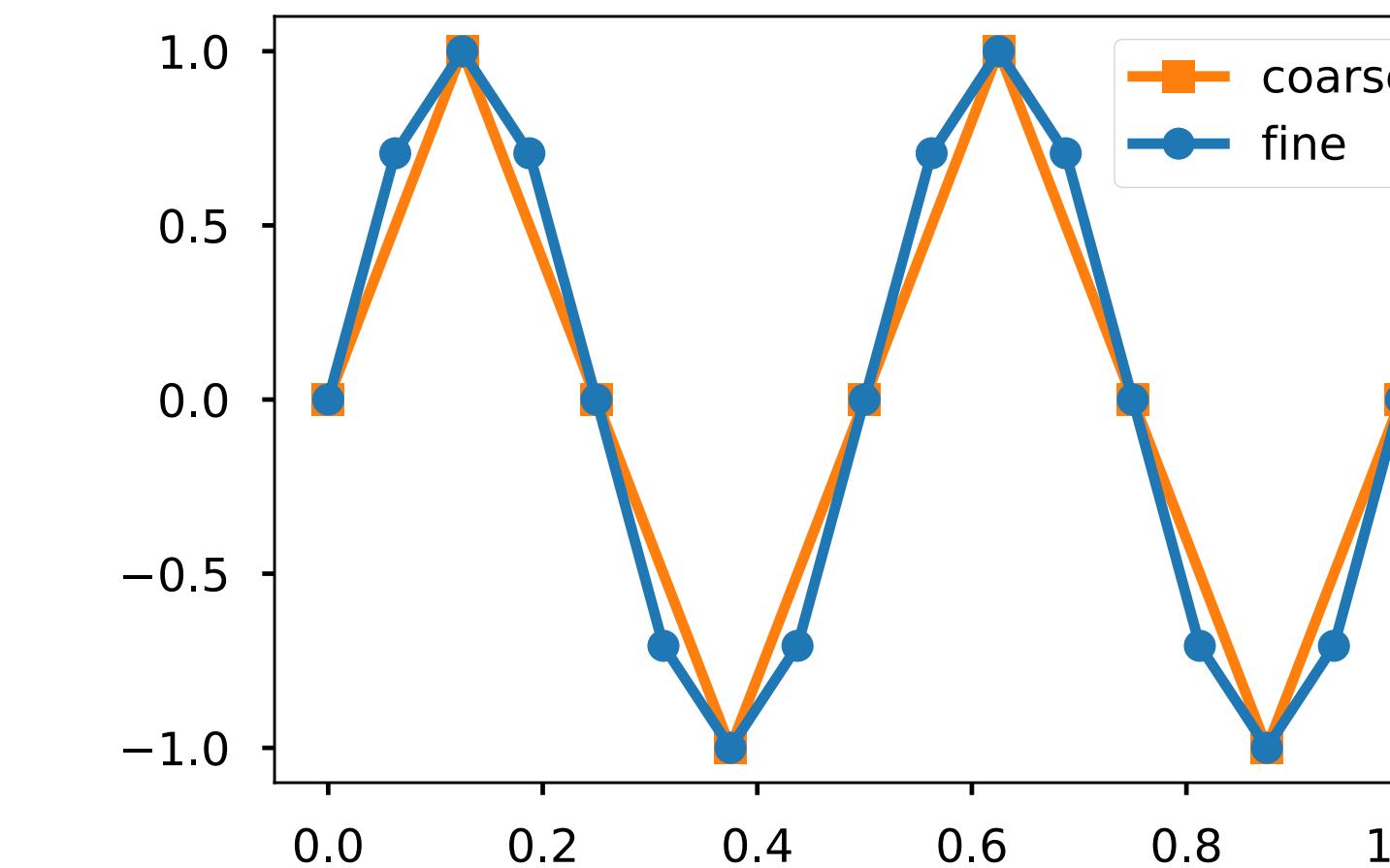
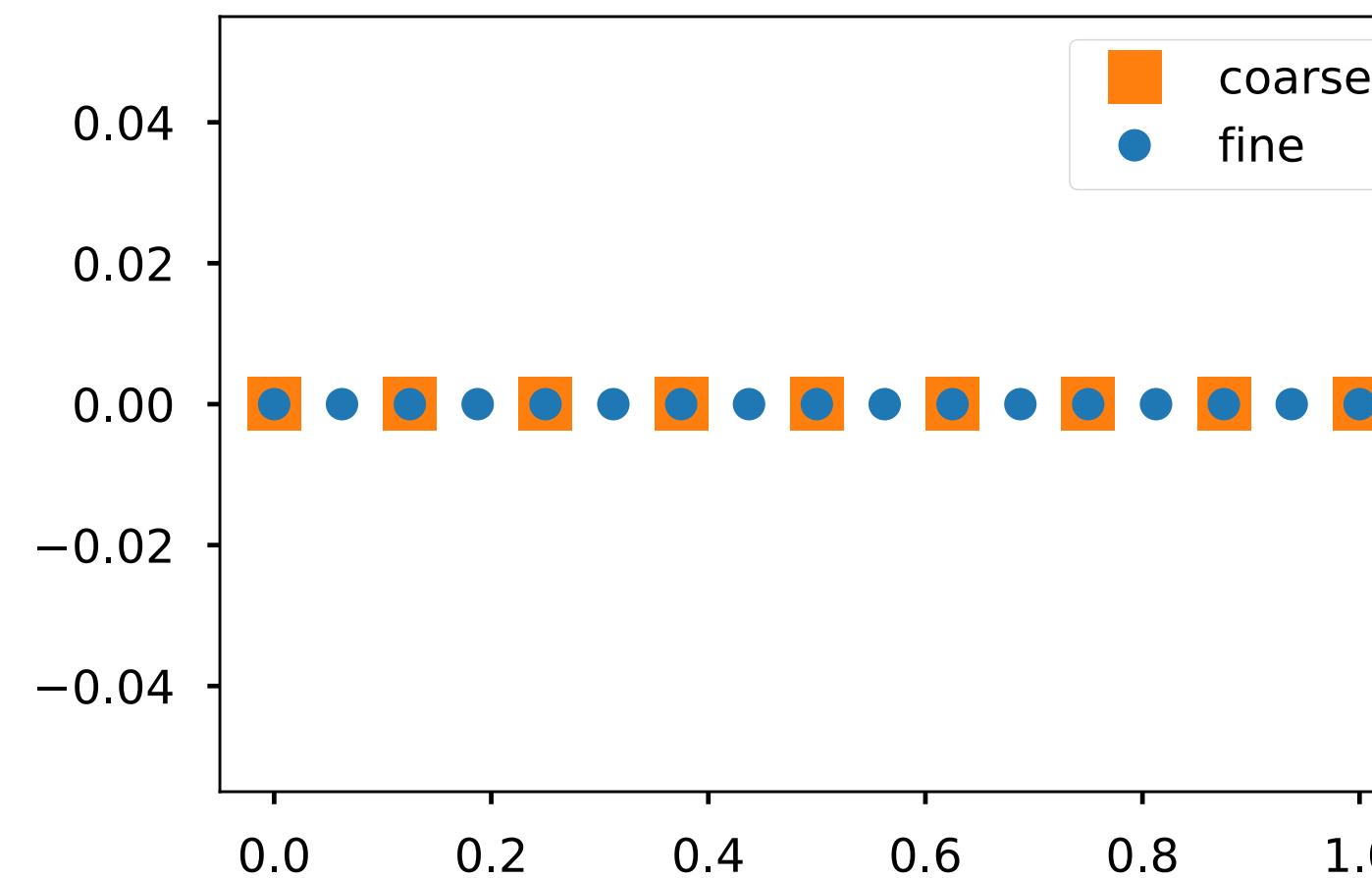
- Smooth modes look like oscillatory modes when sampled on a coarse grid
- 4-mode of 15 **versus** 4-mode of 7



This looks like a “smooth” mode

This looks like an “oscillatory” mode

# Coarse modes



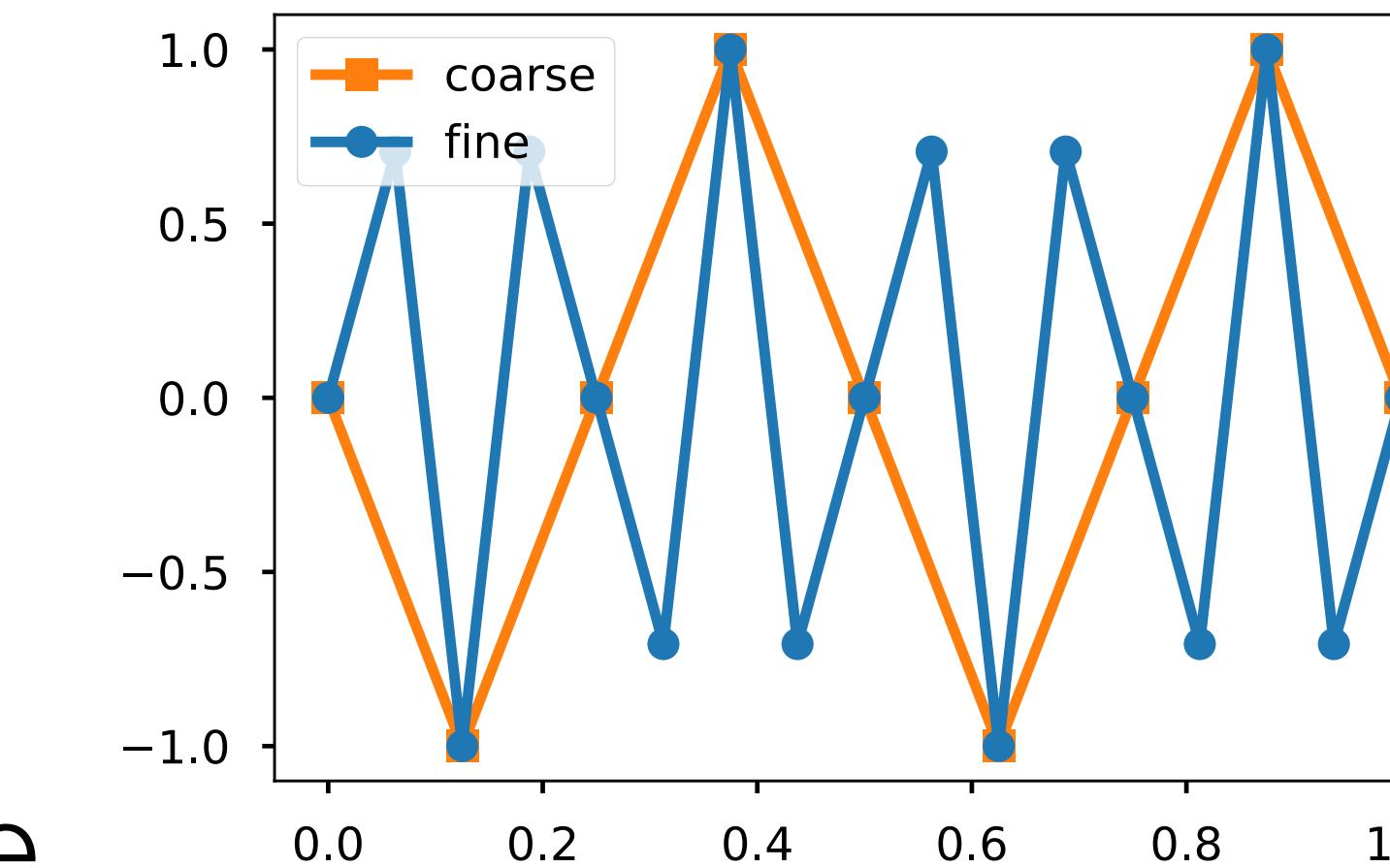
mode 4 of 15

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

Fine mode

Fine mode  
(every other)

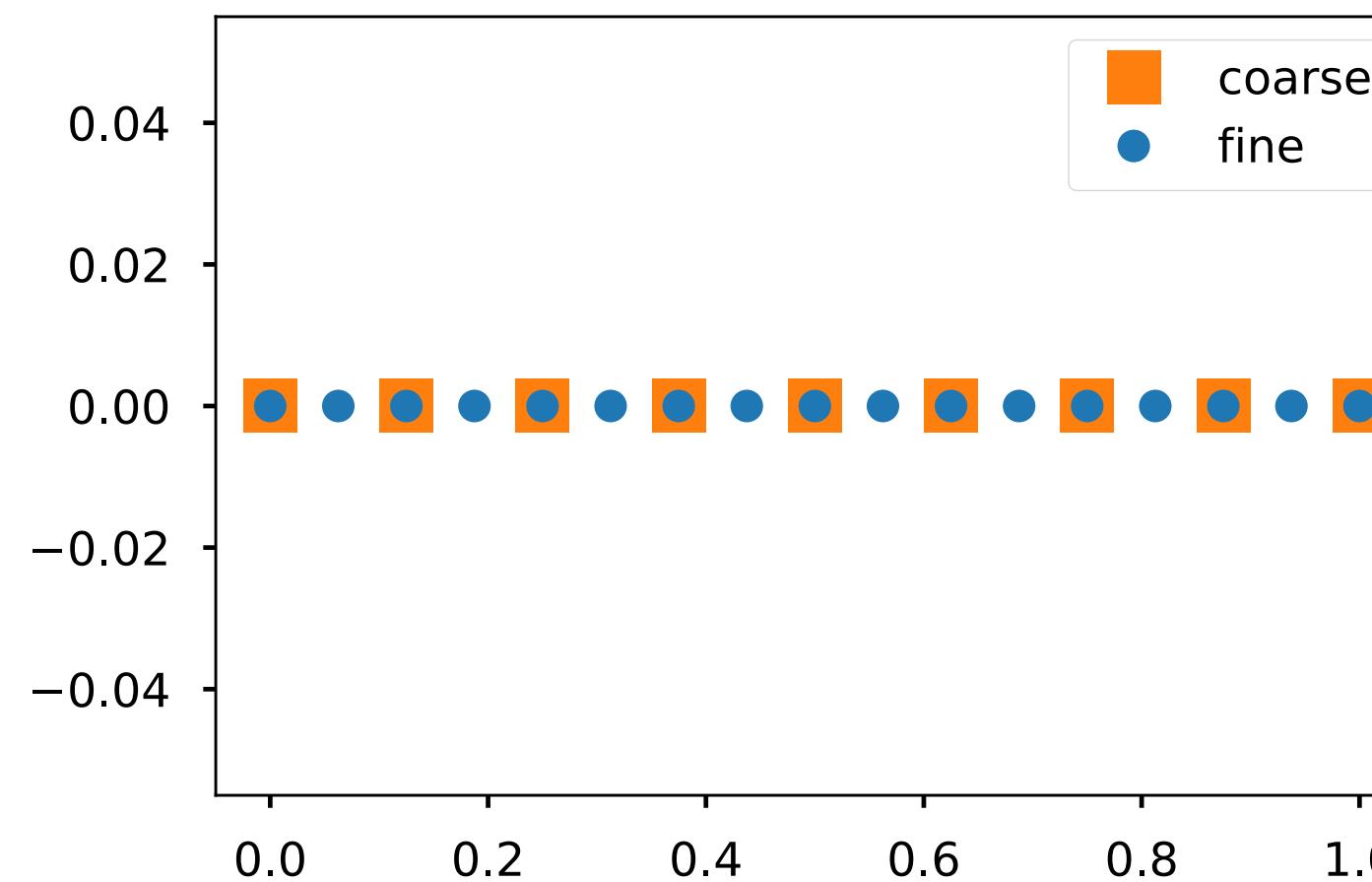
Coarse mode



mode 12 of 15

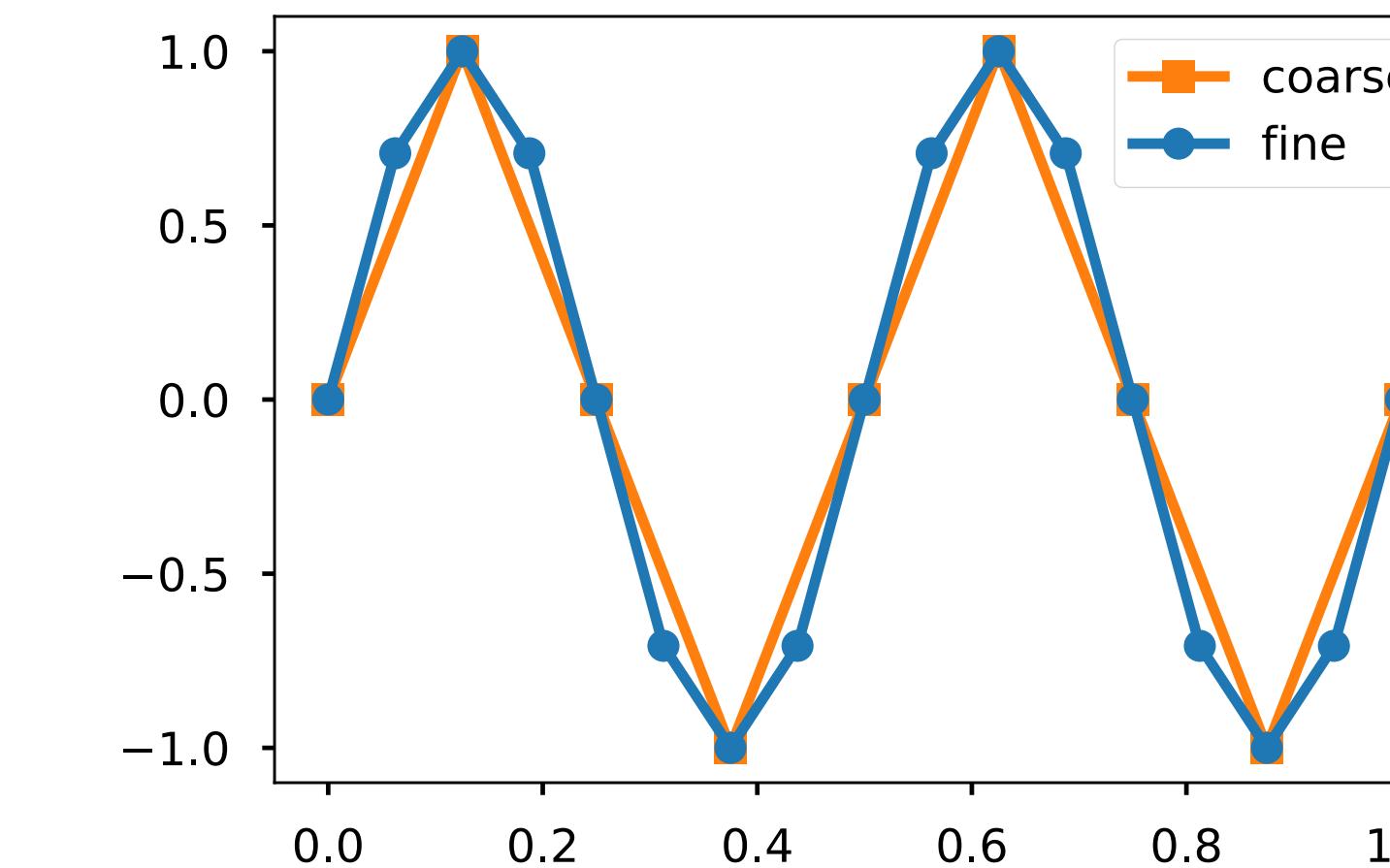
For low modes, k-modes are preserved

# Coarse modes

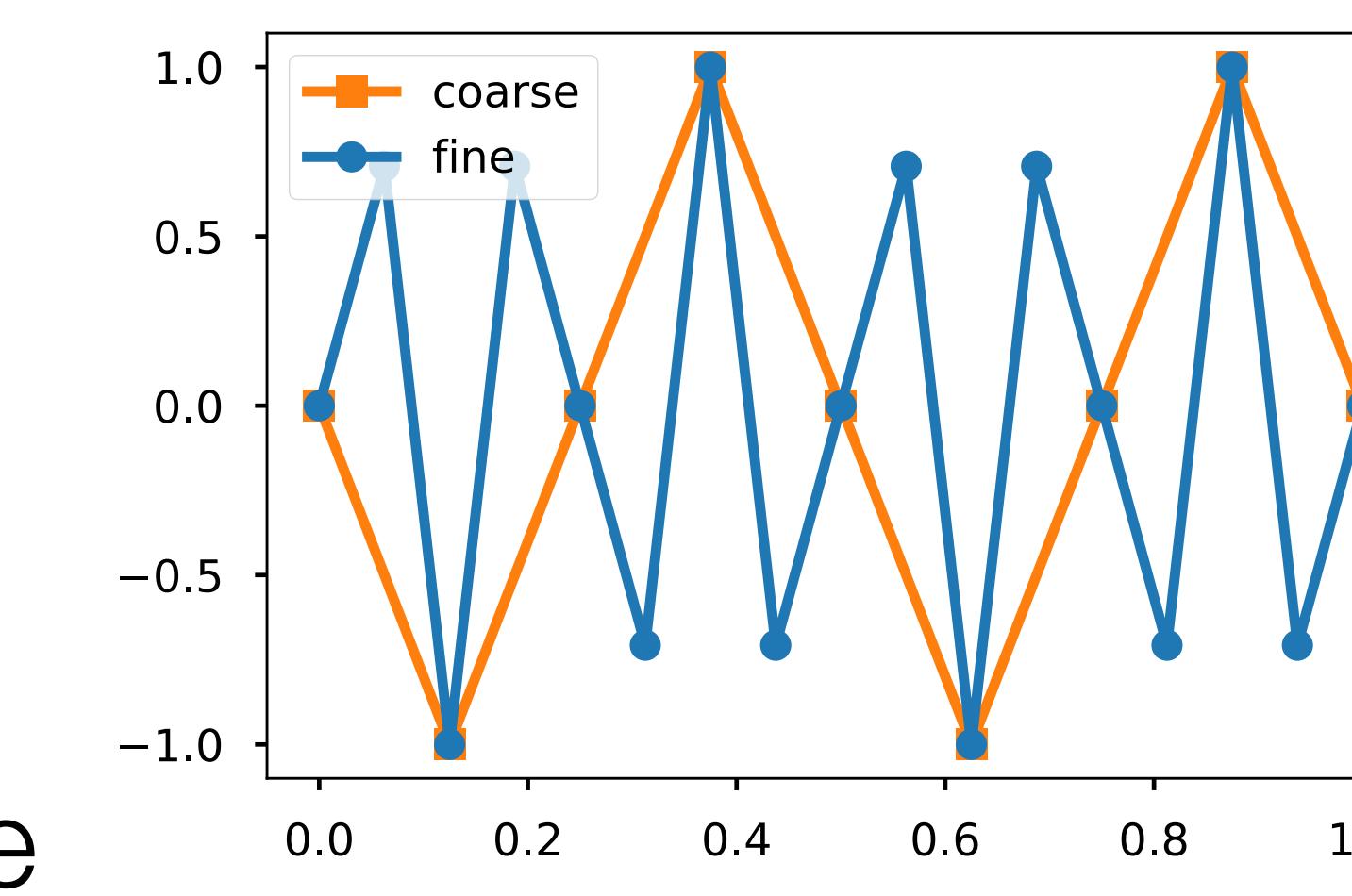


$$\begin{aligned}
 (v_k)_{2j} &= \sin \frac{2jk\pi}{n+1} \\
 &= -\sin \frac{2j(n-k)\pi}{n+1} \quad \text{Fine mode} \\
 &= -\sin \frac{j(n-k)\pi}{(n+1)/2} \quad \text{Fine mode (every other)} \\
 &= -(\hat{v}_{n-k})_j
 \end{aligned}$$

Fine mode  
(every other)  
Coarse mode



mode 4 of 15

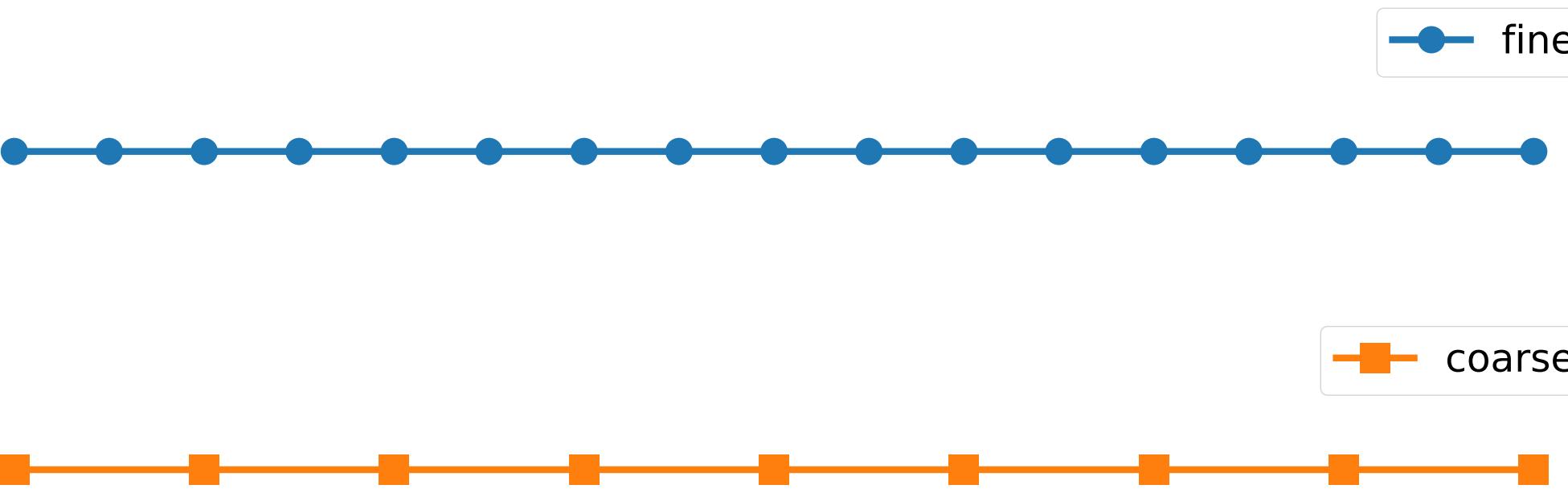


mode 12 of 15

For high modes, k-modes are aliased

# Questions to resolve...

---



- How to transfer between **fine** and **coarse**?
- What do we “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.
- This is  $A$ -orthogonal projection onto  $V$
- Let's construct  $V$  from, say, piecewise cont. linears

# Interpolation

---

- 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

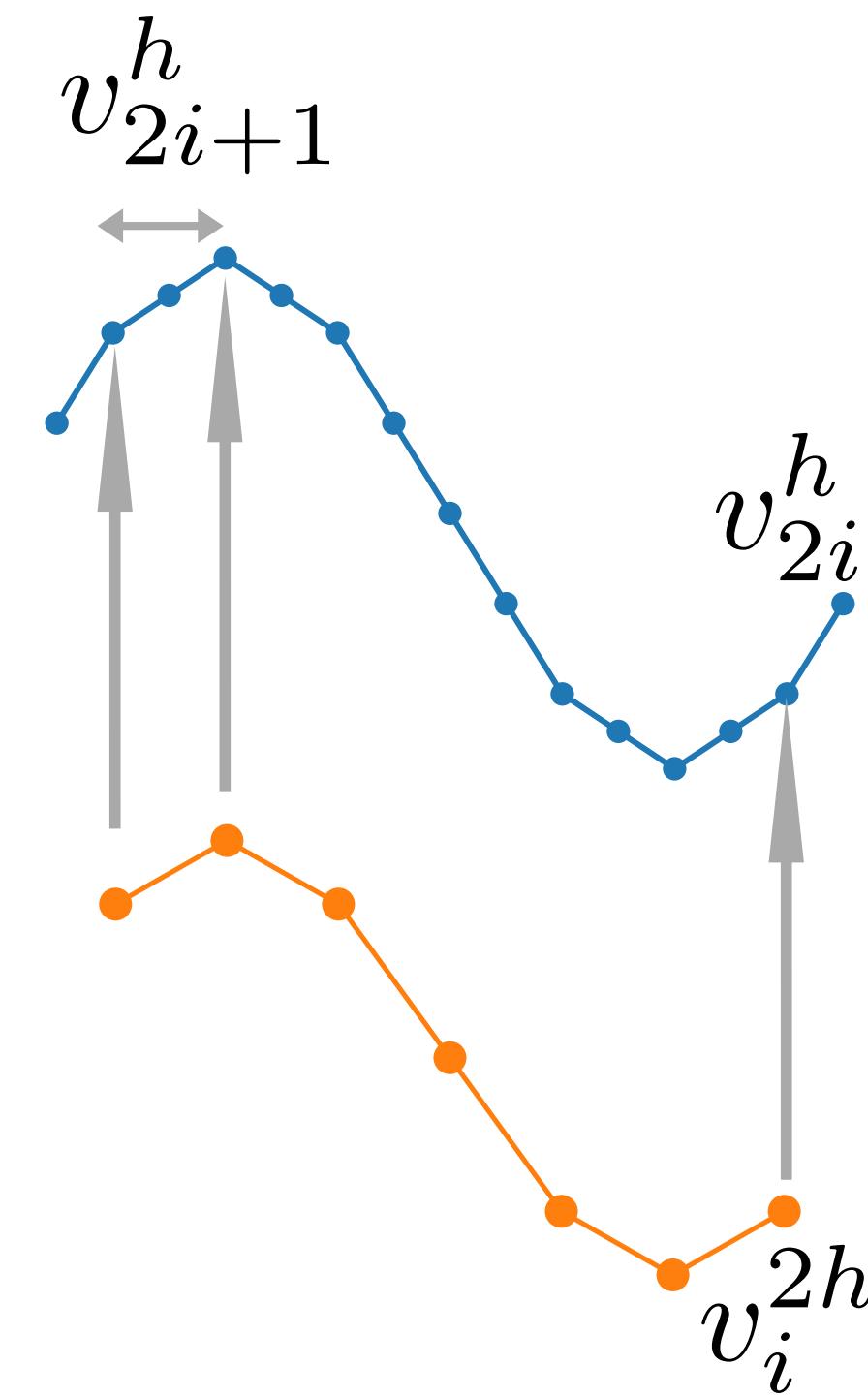
---

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

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

Injection

Average  
(linear interp)

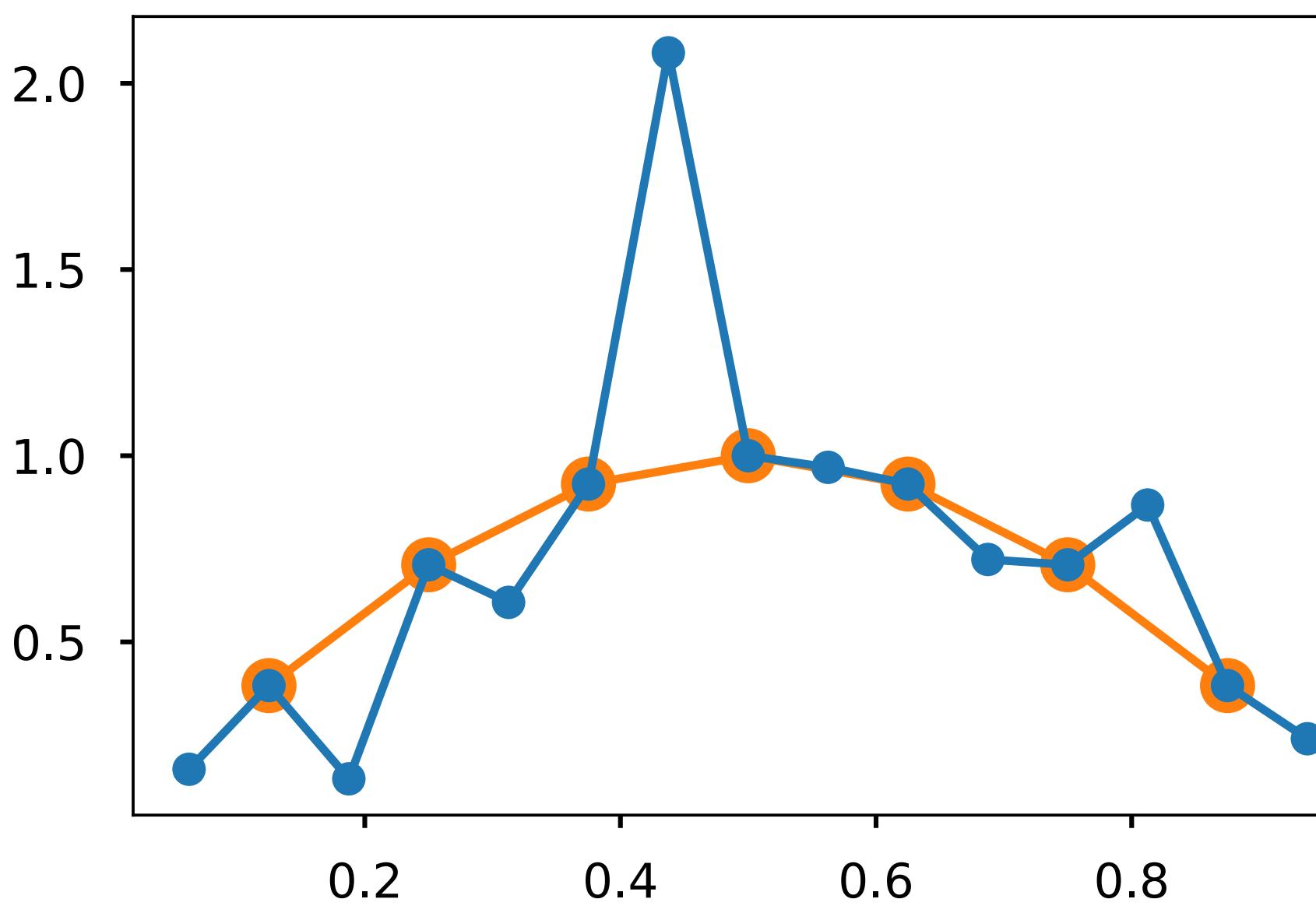
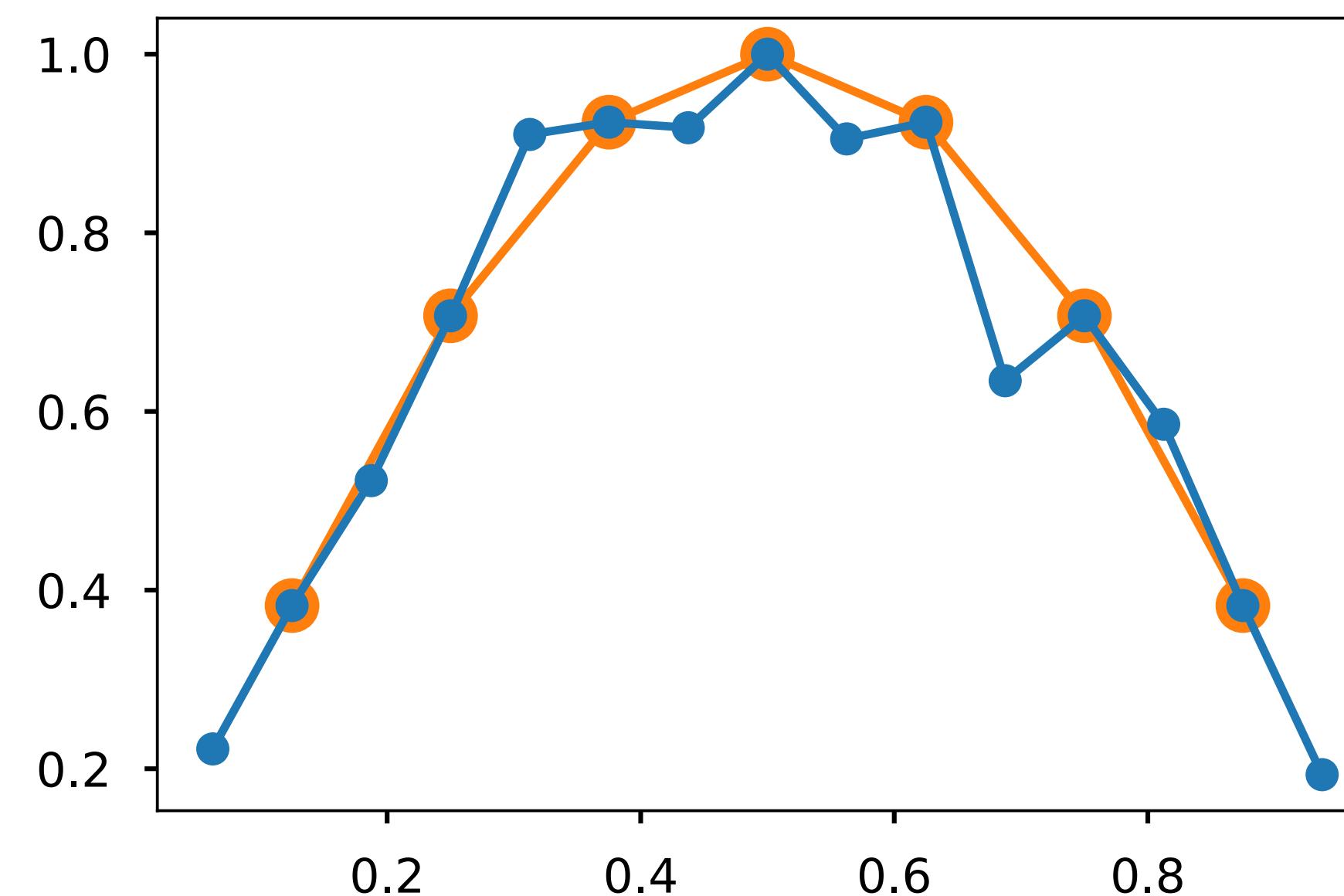


- Values at points common to both grids are reused (injected)

# Interpolation

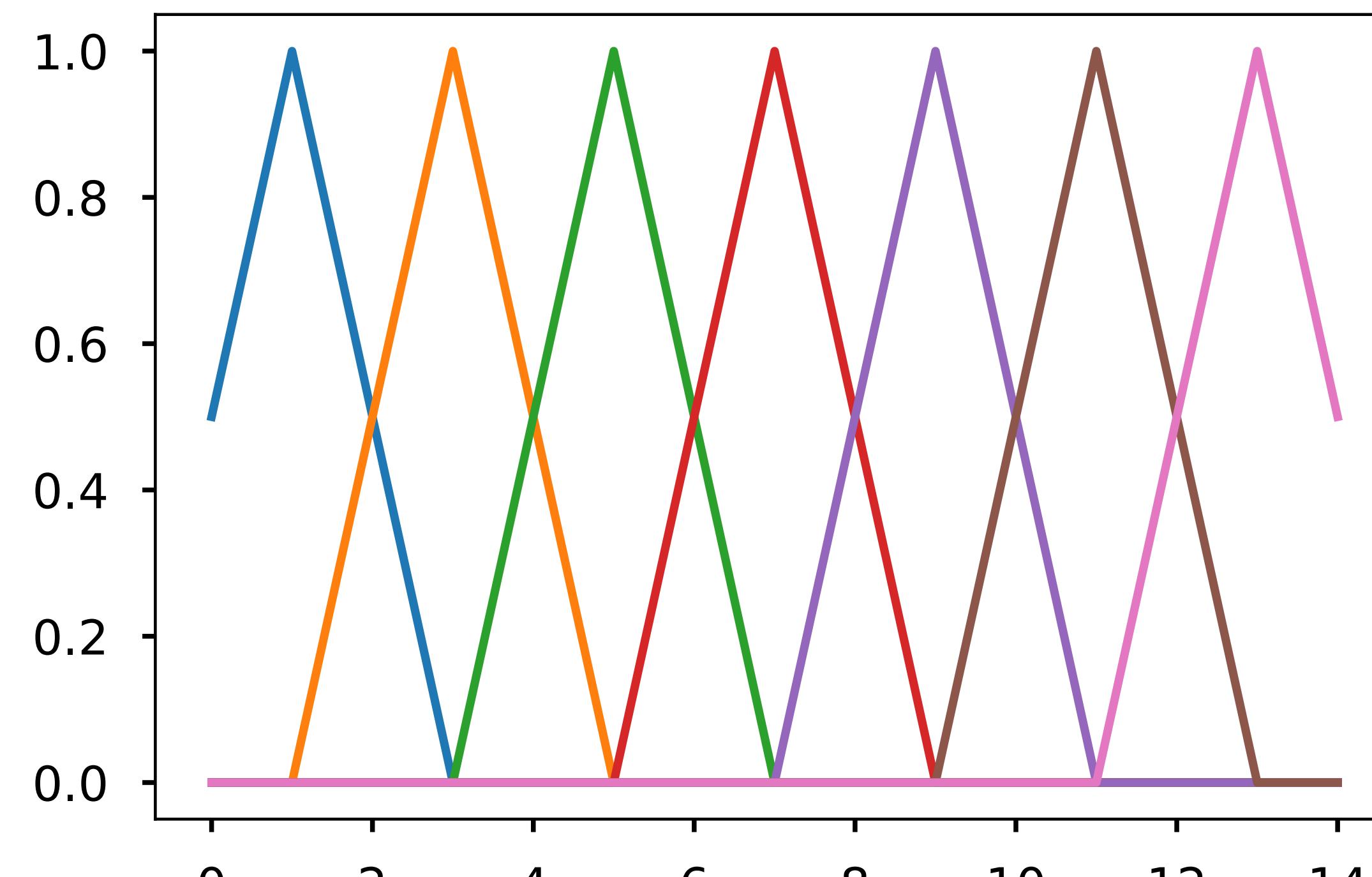
---

- Things that are smooth are interpolated well (smooth error!)
- Things that are not smooth are not interpolated well (un-smooth error!)



## In matrix form

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



- The columns of  $P$  are basis functions (right)
- Fine-grid vectors in  $\text{Range}(P)$ ,  $Pv$ , 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

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

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

## Variations on this theme

---

- An alternative to this 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  $A_c = A_{2h}$

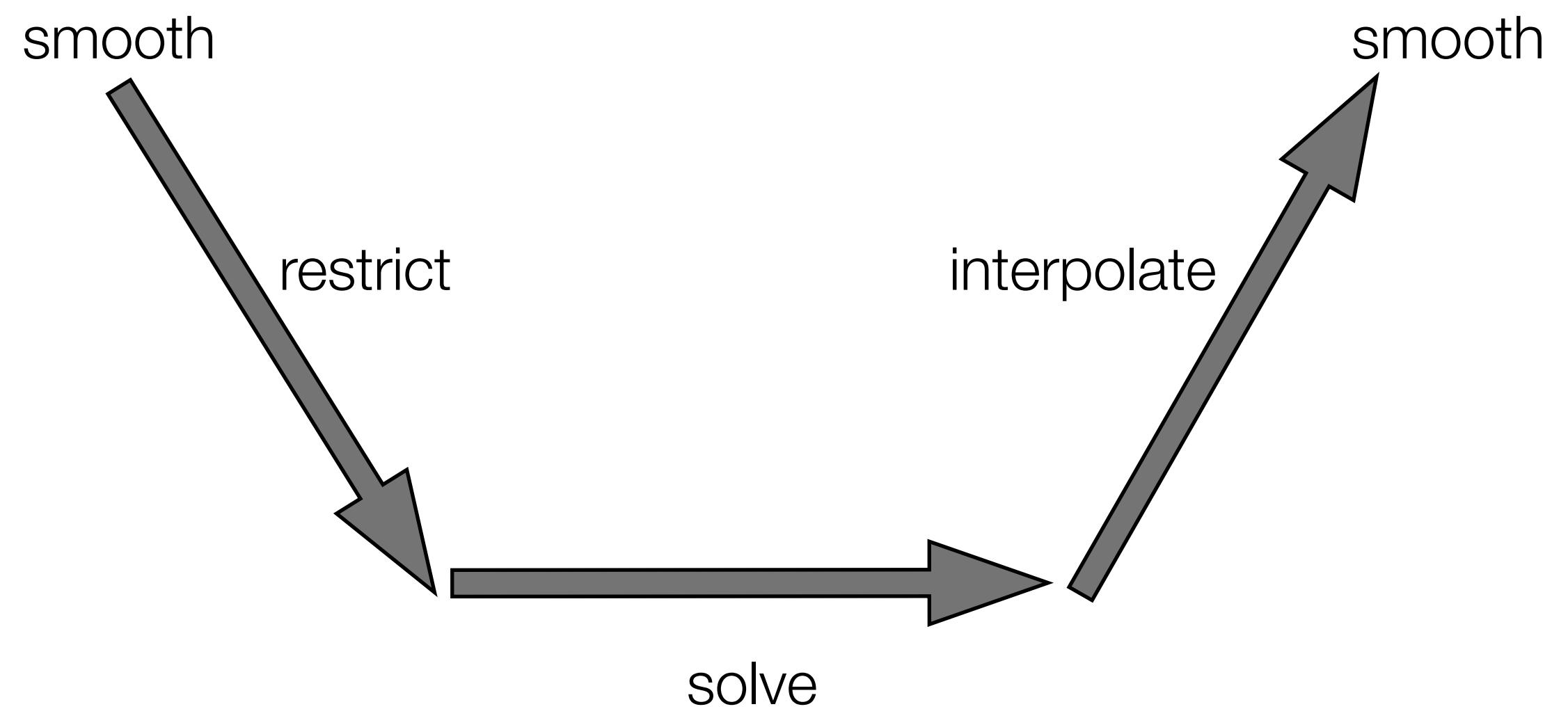
$$\begin{bmatrix} 0 & 1 & 0 & & & & \\ & 0 & 1 & 0 & & & \\ & & 0 & 1 & 0 & & \\ & & & 0 & 1 & 0 & \\ & & & & 0 & 1 & 0 \\ & & & & & 0 & 1 & 0 \\ & & & & & & 0 & 1 & 0 \\ & & & & & & & 0 & 1 \end{bmatrix}$$
$$\frac{1}{4} \begin{bmatrix} 1 & 2 & 1 & & & & & & & & & & & \\ & 1 & 2 & 1 & & & & & & & & &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

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

# Algorithm: two-level multigrid

Input: initial guess

1. Smooth  $\nu_{pre}$  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  $\nu_{post}$  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 is Multigrid?

---

- Would like the 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

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

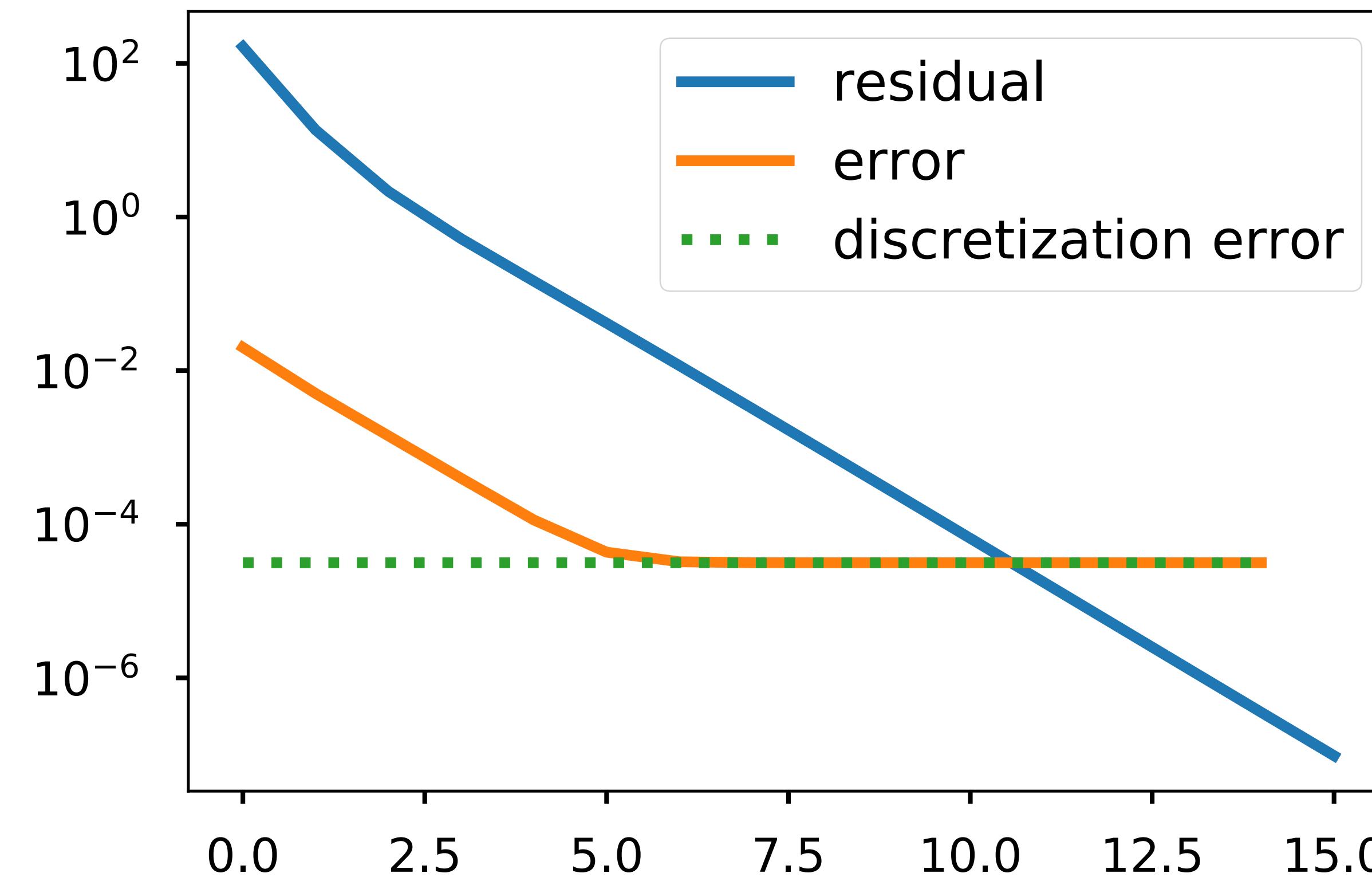
- Then we need

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

cycles

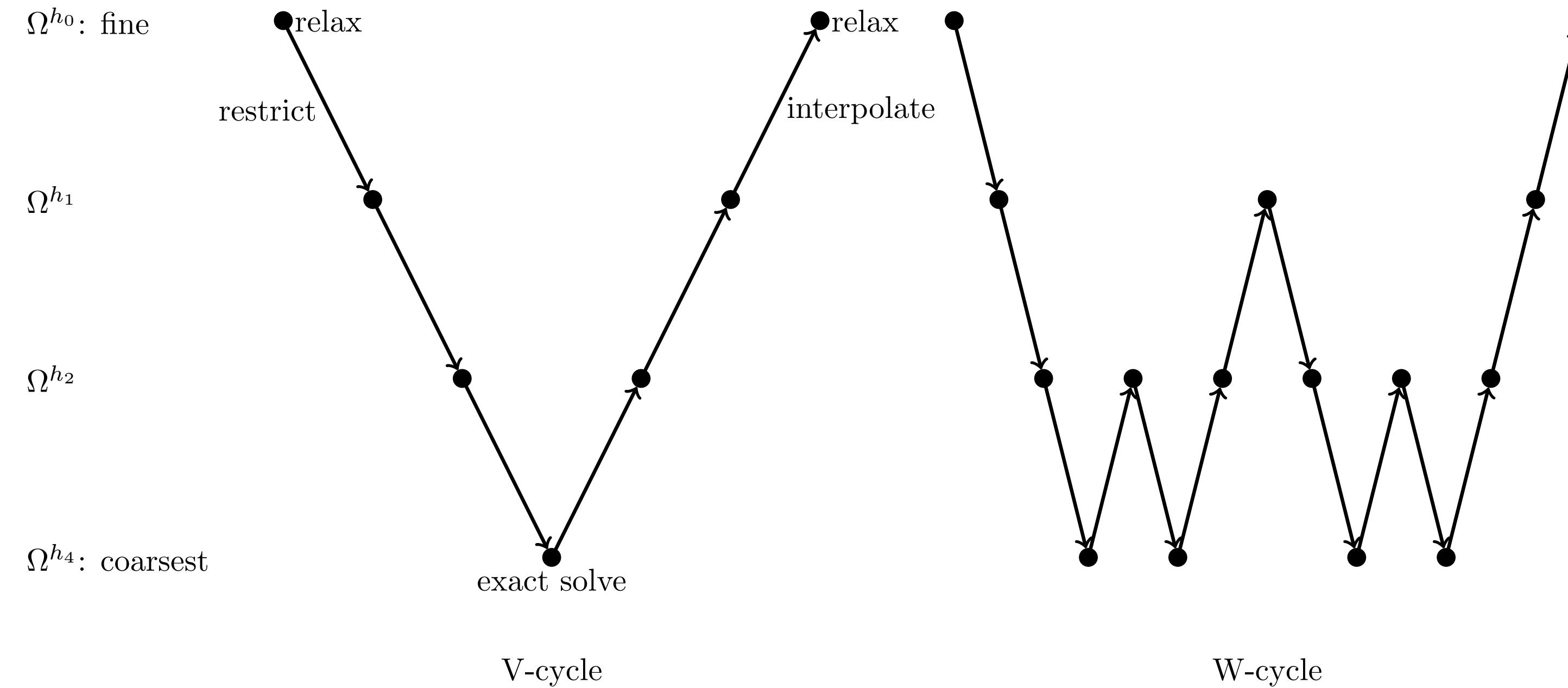
# Algebraic error

---



- The total error is limited by the discretization error

# The Multigrid V-Cycle and W-Cycle



- Two-grid cycle can expose issues with coarser interpolation
- W-Cycle can account for inadequate coarser level solves
- **Exact solve?** Usually a pseudo-inverse

## Up next...

---

- What can go wrong and what to do... (This afternoon)
- What about coupled systems (Tomorrow)
- What happens when we drop the notion of a *grid* ? (Thursday)