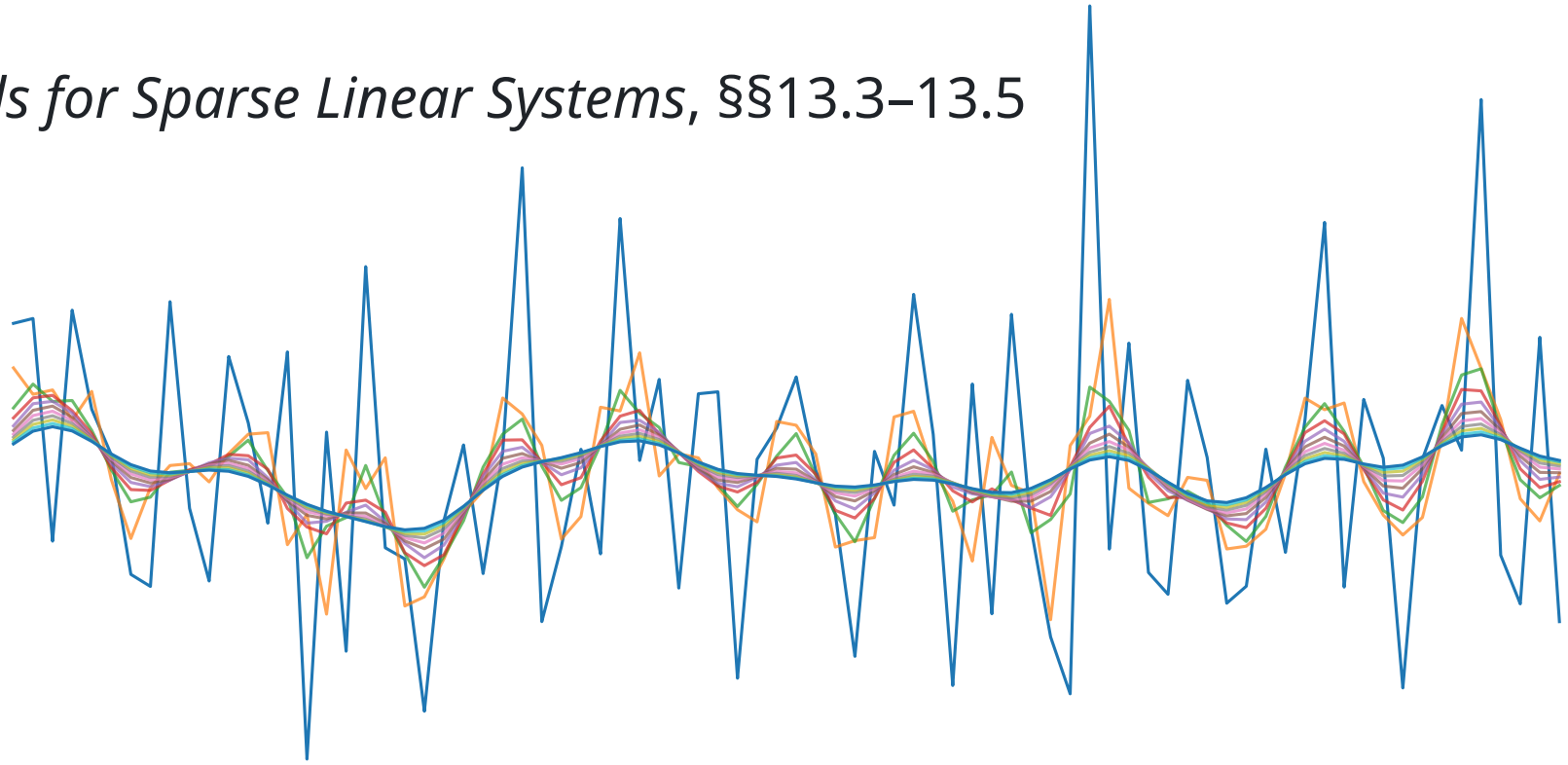
Multigrid Methods

Y. Saad, *Iterative Methods for Sparse Linear Systems*, §§13.3–13.5

Presenter: Jiaze Li



"**Talk is cheap. Show me the code.**" – Linus Torvalds

Code is available on Li-Jesse-Jiaze/multigrid-playbook.

Model Problem

1D model problem

$$-u_{xx} = f$$
$$u(0) = u(1) = 0$$

With finite differences

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

As matrix form

$$\frac{1}{h^2} \begin{bmatrix} 2 & -1 & & \\ -1 & \ddots & \ddots & \\ & \ddots & & \end{bmatrix} \begin{bmatrix} u_1 \\ \vdots \\ u_n \end{bmatrix} = \begin{bmatrix} f_1 \\ \vdots \\ f_n \end{bmatrix}$$

Use the Jacobi Method

Solve the problem

$$Ax = b$$

Iteratively

$$x^{(1)} = x^{(0)} + D^{-1}r^{(0)}$$

> Or consider it as a gradient descent with fixed step size 1 and preconditioner D^{-1}

Exact solution

$$x^*$$

Error

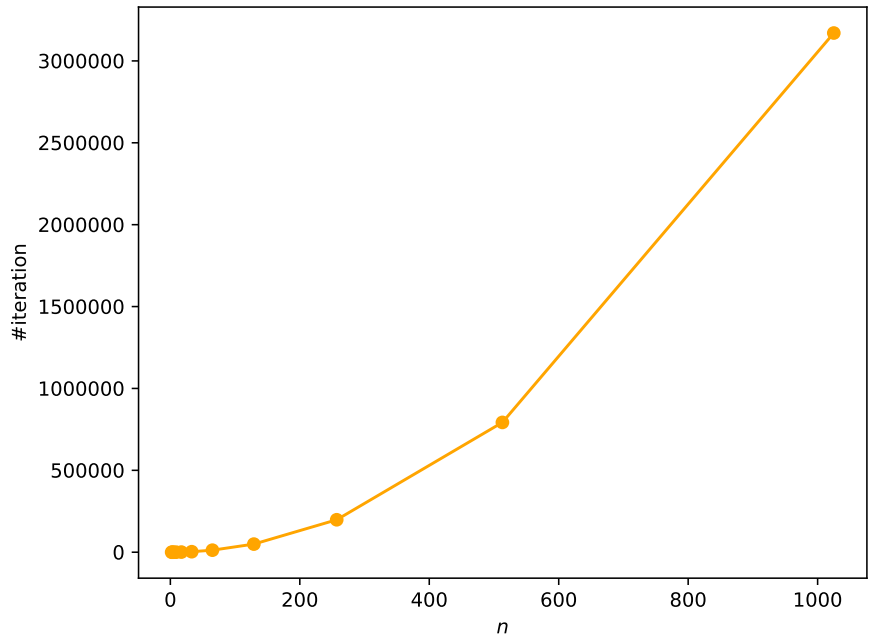
$$e^{(0)} = x^* - x^{(0)}$$

Residual

$$r^{(0)} = b - Ax^{(0)} = Ae^{(0)}$$

Some Convergence Test

The number of iterations required to get the tolerance of 1.0E-10 (RMS residual norm)



Burkardt, J., 2011. Jacobi Iterative Solution of Poisson's Equation in 1D [online]

The computational cost is about $\mathcal{O}(n^3)$. But why is that? Let's look at the details.

Look at the matrix

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

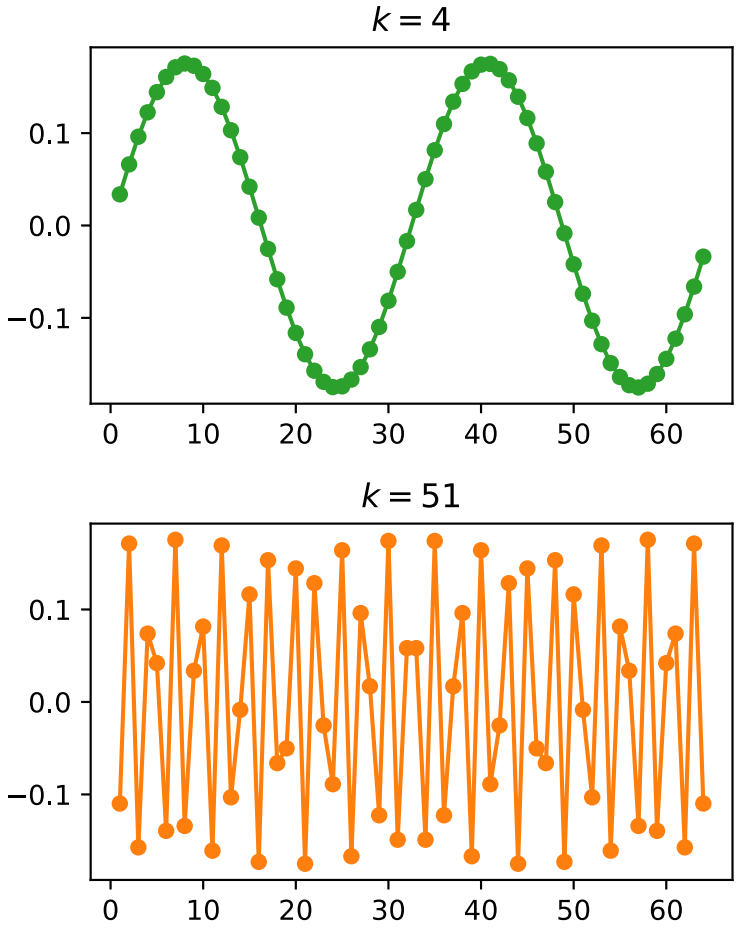
The eigenvalues

$$\lambda_k = 4\sin^2\left(\frac{k\pi}{2(n+1)}\right)$$

The eigenvectors (are Fourier modes)

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

Eigenvectors for $n = 64$



Performs like low and high frequencies

What does Jacobi do to error?

The error propagation

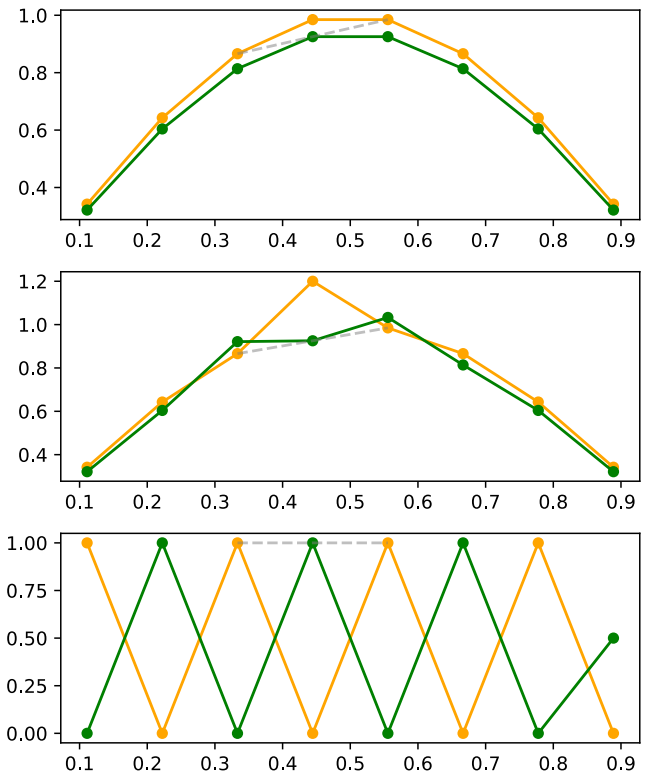
$$e \leftarrow Te \quad T = I - D^{-1}A$$

$$T = \begin{bmatrix} 0 & 1/2 & & & \\ 1/2 & 0 & 1/2 & & \\ & \ddots & \ddots & \ddots & \\ & & 1/2 & 0 & 1/2 \\ & & & 1/2 & 0 \end{bmatrix}$$

It is averaging (like a mean filter)

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

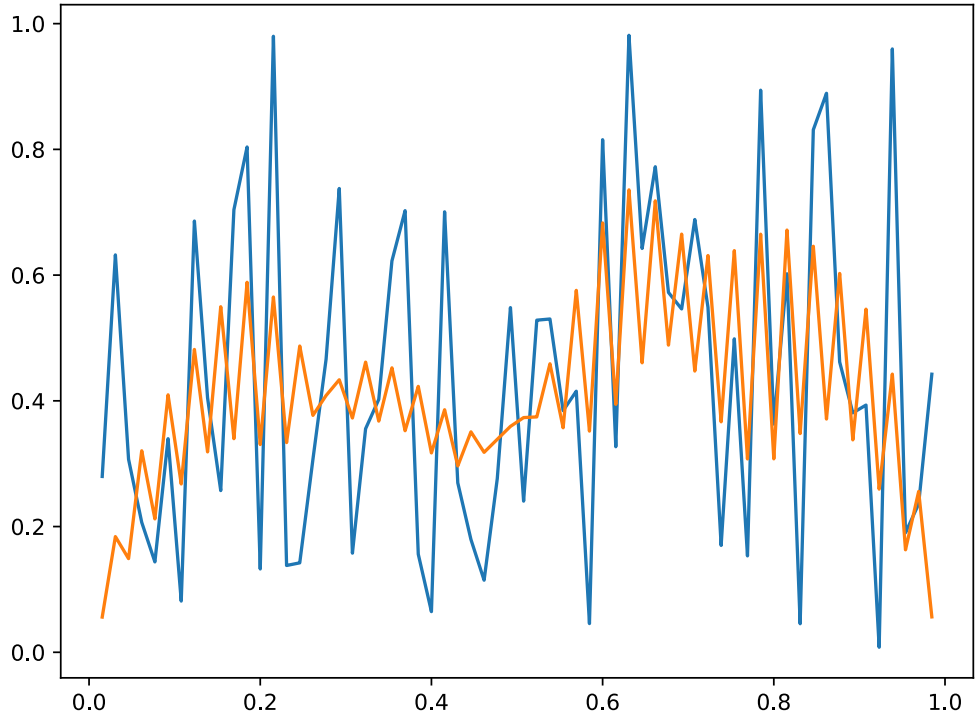
For different types of error



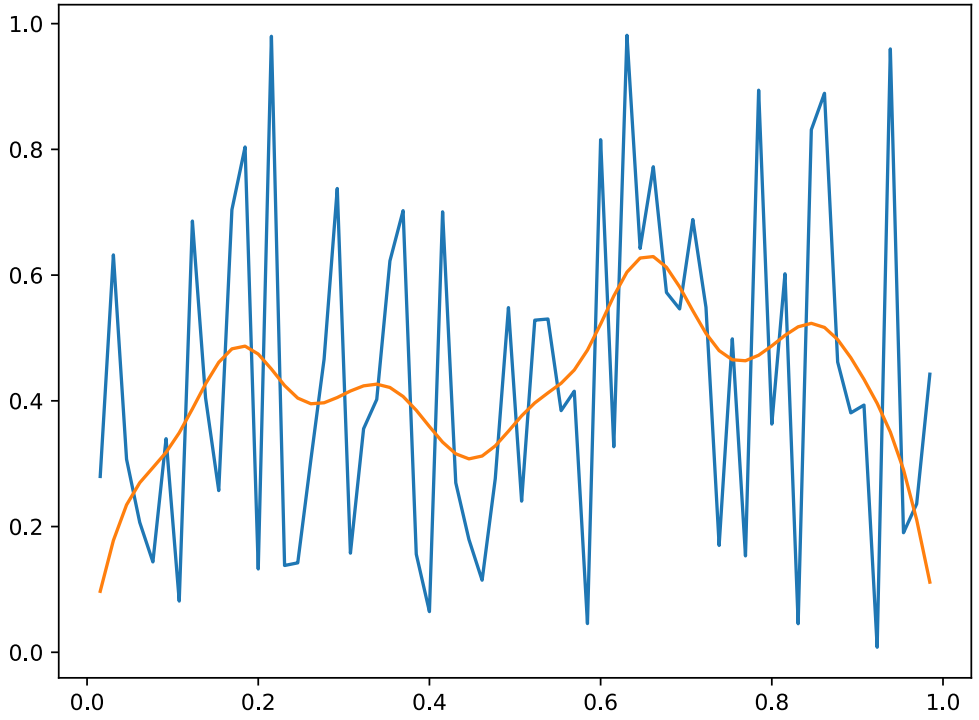
It *averages out* certain frequency quickly

From Jacobi to weighted-Jacobi

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



$$u \leftarrow u + \omega D^{-1}r, \omega = 2/3$$



Like a low-pass filter. Why?

Fourier Analysis

Error propagation T

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

Using the eigenvectors of T as a basis for the error space

$$e^{(0)} = \sum_{k=1}^{n} c_k v_k$$

Then the error transforms like

$$e^{(\sigma)} = Te^{(0)} = \sum_{k=1}^{n} c_k T^{\sigma} v_k = \sum_{k=1}^{n} c_k \lambda_k^{\sigma} v_k$$

Converge with the factor $\|T\|$ or $\rho(T)$

For Jacobi and weighted-Jacobi

$$T = I - D^{-1}A \longrightarrow \lambda_k = 1 - \frac{1}{2} \cdot 4 \cdot \sin^2\left(\frac{k\pi}{2(n+1)}\right)$$

$$T = I - (2/3)D^{-1}A \longrightarrow \lambda_k = 1 - \frac{2}{3} \cdot \frac{1}{2} \cdot 4 \cdot \sin^2\left(\frac{k\pi}{2(n+1)}\right)$$

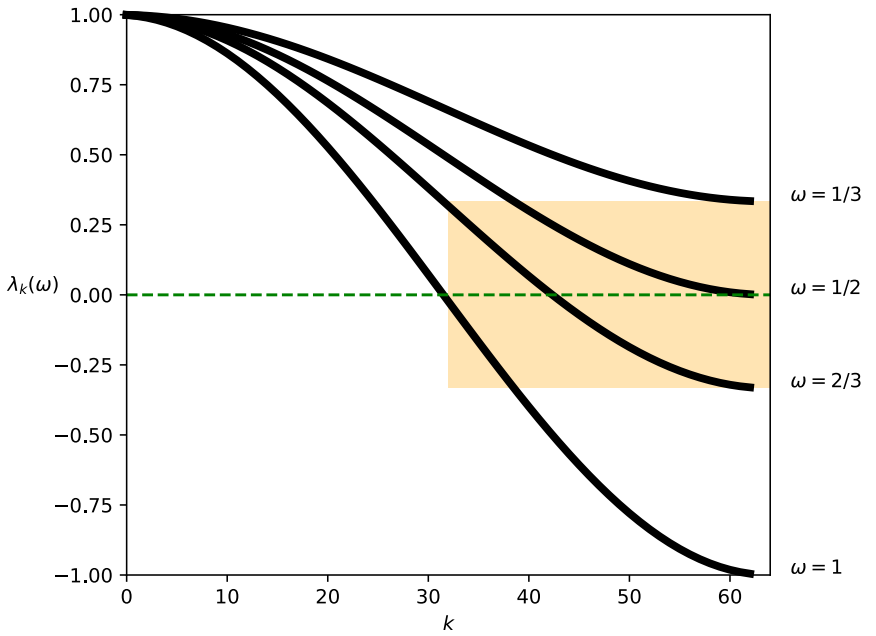
The spectral radius $\rho_n \approx 1 - C/n^2$

But

$$e^{(\sigma)} = \sum_{k=1}^{n} c_k \lambda_k^{\sigma} v_k$$

error on different direction v_k (or frequency k) is reduced by different magnitude of λ_k

λ_k corresponding to different ω

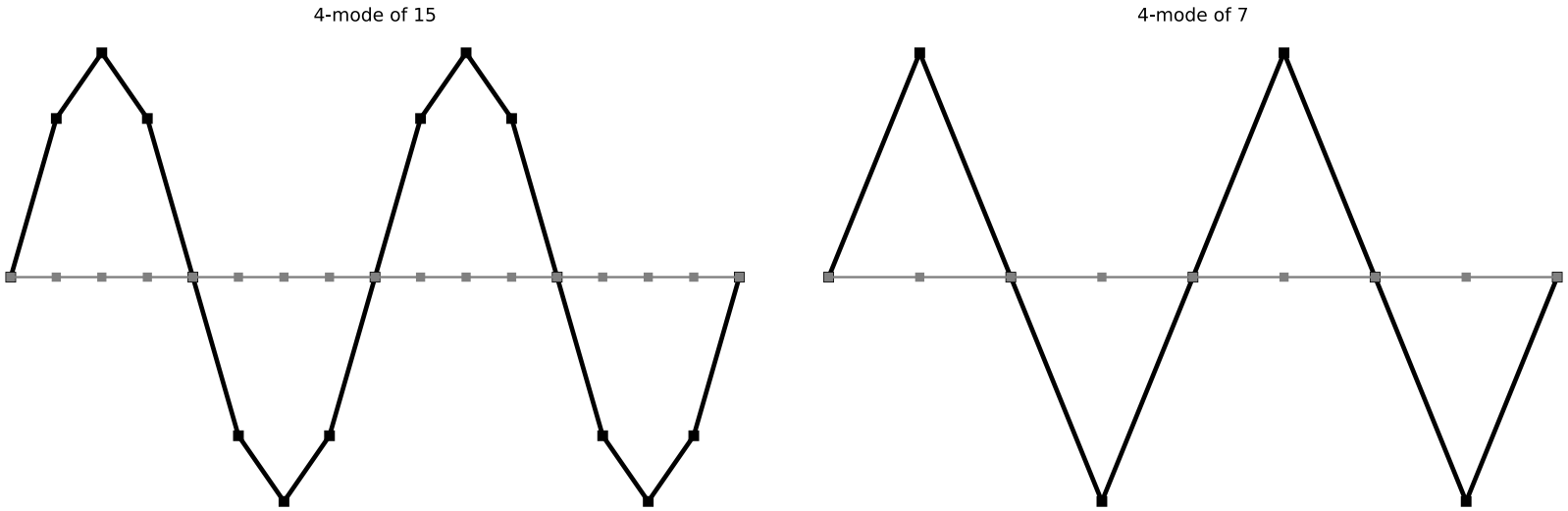


For $\omega = 2/3$, the high frequency part of the error is reduced by (at least) $1/3$

> **Smoothing factor** of T: the maximum magnitude of the upper half spectrum

How can we turn low frequency errors to high frequency?

Sampling on a coarse grid



After this, smooth modes look like oscillatory modes, with little loss of information.

So far we got

Classical iterative methods converge slowly as $n \uparrow$, but have the smoothing property.

Low-frequency information is well approximated on the coarse grid and becomes high-frequency.

Next questions

How to transfer between fine and coarse?

What do we do “solve” on a coarse grid?

Remind the Projection Methods

Look for the "best" update:

$$x^{(1)} \leftarrow x^{(0)} + u$$

Over a smaller space (with A-norm)

$$\min_{u \in \text{ span } \{V\}} \left\| x^* - x^{(1)} \right\|_A$$

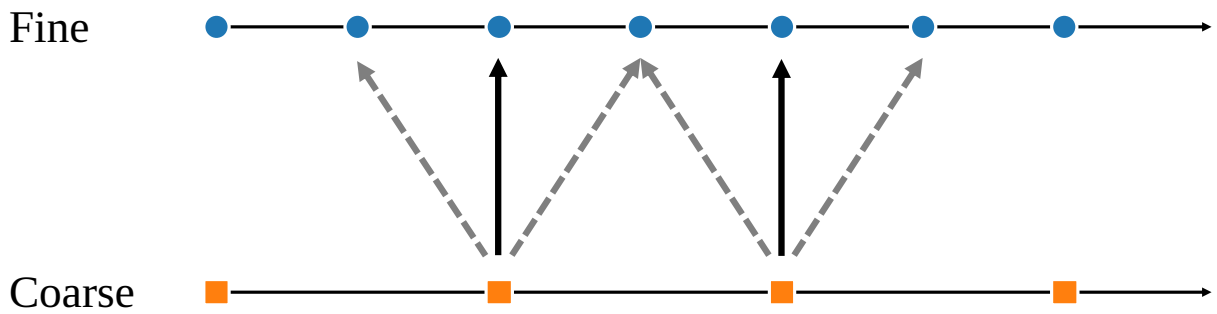
So the update looks like

$$x^{(1)} = x^{(0)} + V \left(V^\top A V \right)^{-1} V^\top r^{(0)}$$

The error

$$e^{(1)} = (I - \underbrace{V \left(V^\top A V \right)^{-1} V^\top A}_{\text{A-orthogonal projection}}) e^{(0)}$$

Prolongation: From Coarse to Fine



Construct an operator

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

where Ω^{2h} is the coarse grid and Ω^{h} is the fine grid

Interpolation

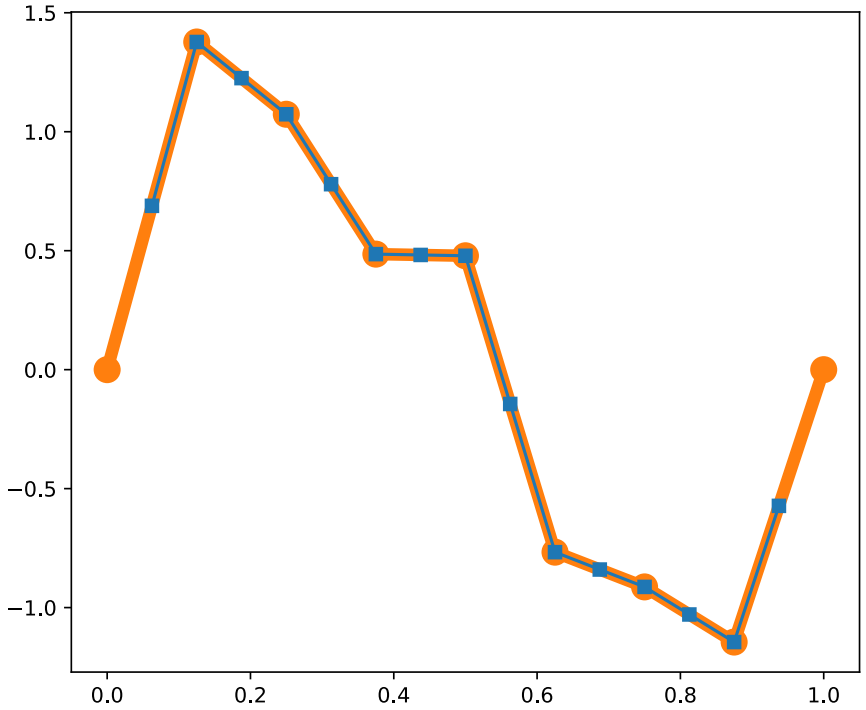
$$v_{2i}^h = v_i^{2h}$$

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

Or in matrix form

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

Looks like



Notice: P is full rank

In projection method

$$x^{(1)} = x^{(0)} + V \left(V^\top A V \right)^{-1} V^\top r^{(0)}$$

If V is $\operatorname{span}\{p\}$, or just P

$$x^{(1)} = x^{(0)} + P \left(P^\top A P \right)^{-1} P^\top r^{(0)}$$

We get the **two-grid method**

Two-grid Method

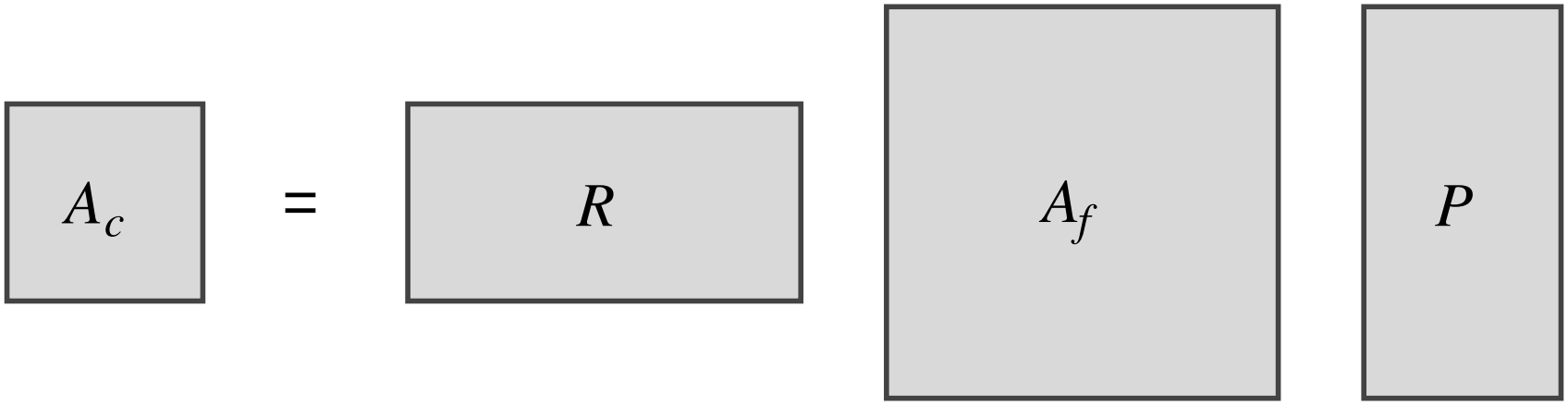
$$x^{(1)} = x^{(0)} + P\left(P^\top AP\right)^{-1} P^\top r^{(0)}$$

1. Given	$x^{(0)}$
2. Smooth a few times	$x^{(0)} \leftarrow x^{(0)} + \omega D^{-1} A r^{(0)}$
3. Form residual	$r^{(0)} = b - Ax^{(0)}$
4. Restrict the residual	$P^\top r^{(0)}$
5. Solve the coarse problem	$P^\top AP\delta = P^\top r^{(0)}$
6. Interpolate the approx error	$P\delta$
7. Correct	$x^{(1)} = x^{(0)} + P\delta$

$R = P^\top$ is the **restriction** here

$A_c = P^\top AP$ is the **coarse level operator** here

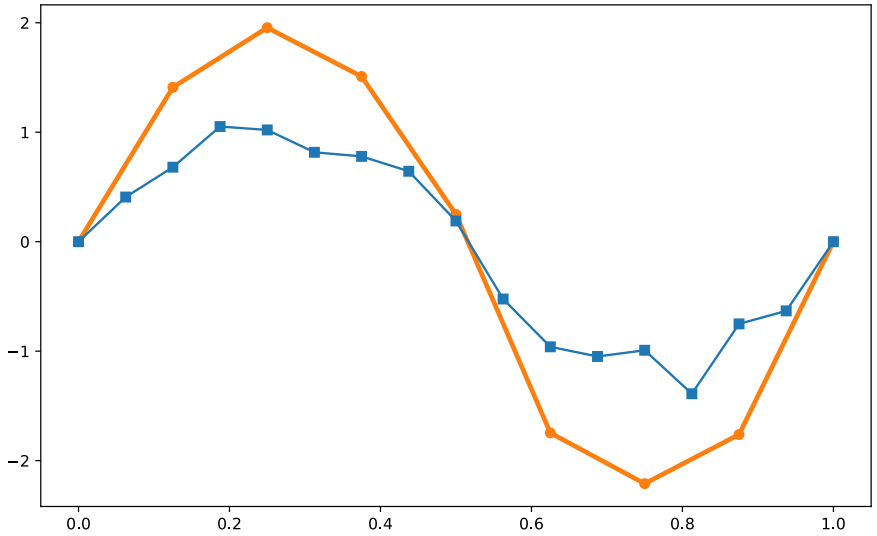
Let's look at the coarse level operator A_c



The restriction R don't have to be $P^\top$

It's called a balance pair if

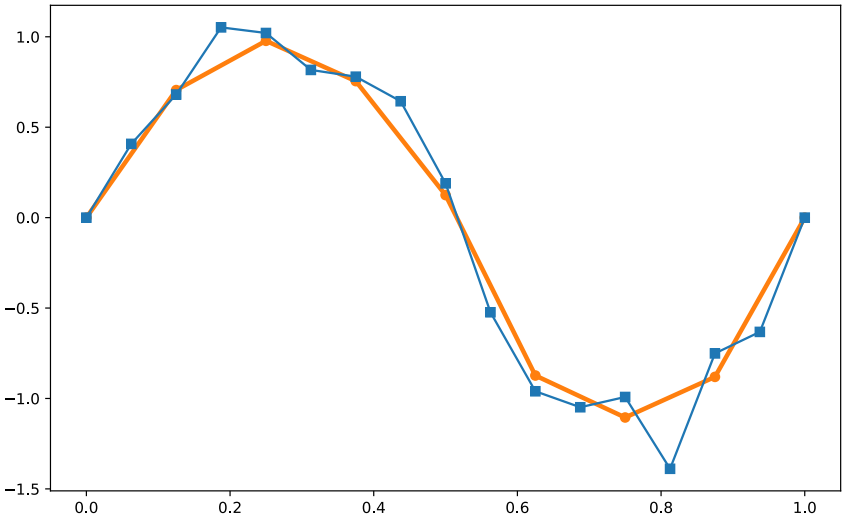
$$R = P^\top = \frac{1}{2}\begin{bmatrix} 1 & 2 & 1 & & & & & & \\ & & 1 & 2 & 1 & & & & \\ & & & & \cdots & \cdots & \cdots & & \\ & & & & & & & 1 & 2 & 1 \end{bmatrix}$$



Other options for restriction

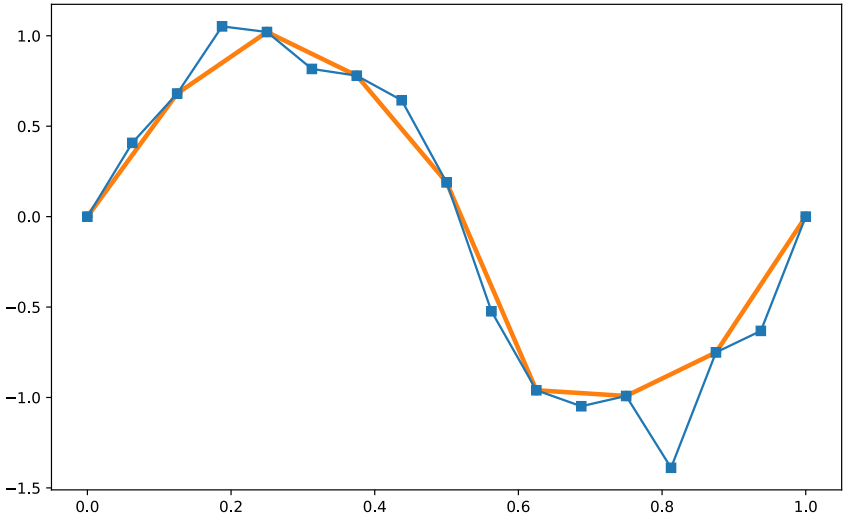
$\frac{1}{2}P^\top$:

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



Injection:

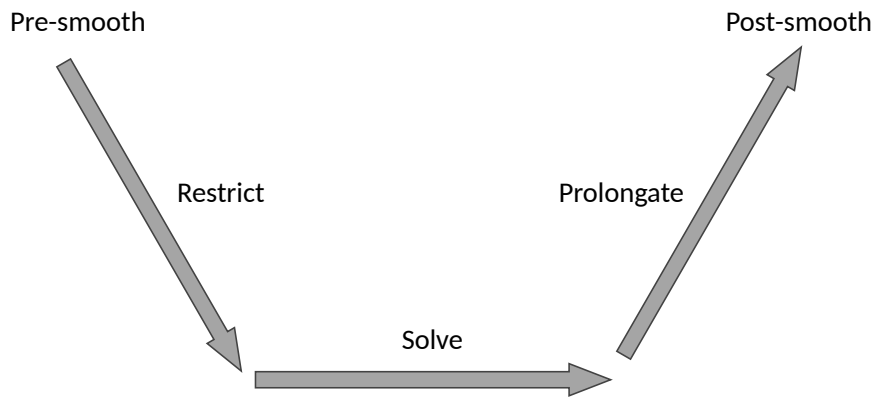
$$\begin{bmatrix} 0 & 1 & 0 & & & & & & \\ & & 0 & 1 & 0 & & & & \\ & & & & \cdots & \cdots & \cdots & & \\ & & & & & & & 0 & 1 & 0 \end{bmatrix}$$



each row of R sums to 1

Algorithm: Two-grid Method

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



A two-level V cycle

Convergence (written)

How Accurate is Multigrid?

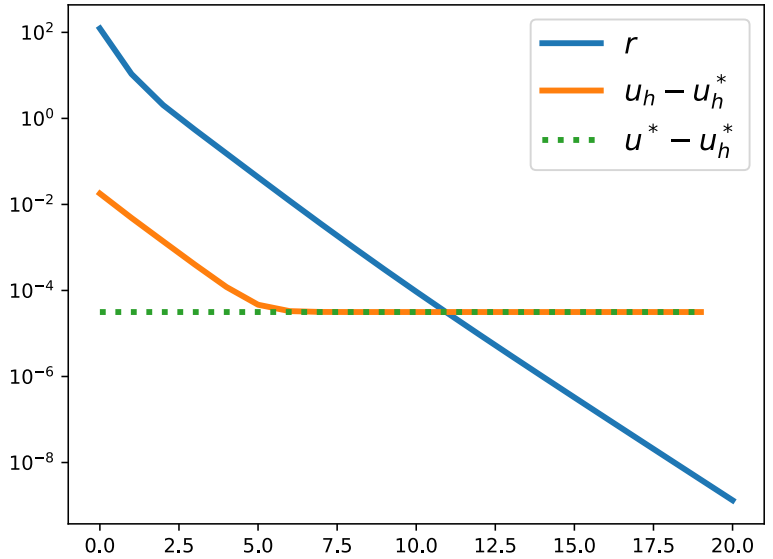
Consider the exact solution to the PDE u^*

$$-u'' = f$$

The exact solution to the linear system u_h^*

$$Au = b$$

The numerical solution $u_h \approx u_h^*$



The total error is limited by the discretization error

W/V cycle

Do two-grid recursively