Lecture 08 – Solving Linear Systems (Part 2)

Prof. Brendan Kochunas

NERS/ENGR 570 - Methods and Practice of Scientific Computing (F22)

COLLEGE OF ENGINEERING
NUCLEAR ENGINEERING & RADIOLOGICAL SCIENCES
UNIVERSITY OF MICHIGAN

Outline

- Recap
- Convergence of Fixed point Methods
- Multigrid Methods
- Hands on Jupyter Notebook
- Examples of how to think about stationary methods
- Questions about HW1 and HW2

Learning Objectives: By the end of Today's Lecture you should be able to

- (*Knowledge*) implement Gauss-Seidel or Jacobi
- (*Knowledge*) determine the rate of convergence for stationary iterative methods
- (*Knowledge*) understand how multigrid exploits stationary methods

Review/Recap

Overview of Solution Methods

Direct $O(n^3)$

Iterative $O(n^2) \times m$ ↳ iterations

SpMV

$O(n)$ - Fast

$Ax = b$

Iterative

L8

L9

non-Stationary

- Conjugate Gradient
- Lanczos
- Arnoldi

Krylov

1950's 1970's 1990's

Stationary

Jacobi
Gauss-Seidel } ~1800's
SOR / SSOR. 1950's
Multigrid – 1970's - 1980's

$$\boxed{x^{(\ell+1)} = F x^{(\ell)} + c}$$

L7

Direct

LU (Gaussian Elimination)
QR
Cholesky
Singular Value Decomposition
Eigendecomposition

Convergence of Stationary Iterative Methods

and SSOR

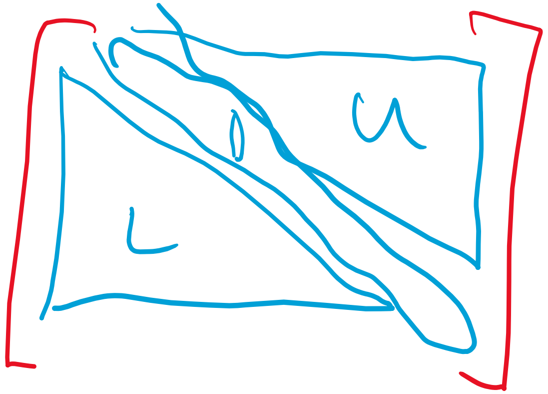
The Jacobi Iterative Method

$$\sum_{\substack{j=1\\ j\neq i}}^{n} a_{i,j} x_j + a_{ii} x_i = b_i$$

Solve for $x_i^{(\ell+1)} = \left[b_i - \sum_{\substack{j=1\\ i\neq j}}^{n} a_{i,j} x_j^{(\ell)} \right] / a_{ii}$

$$x_i^{(\ell+1)} = \left[b_i - \underbrace{\sum_{j=1}^{i-1} a_{i,j} x_j^{(\ell)}}_{L} - \underbrace{\sum_{j=i+1}^{n} a_{i,j} x_j^{(\ell)}}_{U} \right] / \underbrace{a_{ii}}_{D}$$

$$\underline{x}^{(\ell+1)} = \underbrace{-\underline{\underline{D}}^{-1}(\underline{\underline{L}}+\underline{\underline{U}})}_{F_{JI}} \underline{x}^{(\ell)} + \underline{\underline{D}}^{-1}\underline{b}$$

$A =$ 

$A = L + D + U \neq A = LU$

$$\underline{x}^{(\ell+1)} = \underline{\underline{F}}\,\underline{x}^{(\ell)} + \underline{c}$$

The Gauss-Seidel Iterative Method

$$x_i^{(\ell+1)} = \left[b_i - \sum_{j=1}^{i-1} a_{i,j} x_j^{(\ell+1)} - \sum_{j=i+1}^{n} a_{i,j} x_j^{(\ell)} \right] / a_{i,i}$$

Do $i=1,n$

$$x^{(\ell+1)} = -(D+L)^{-1} U x^{(\ell)} + (D+L)^{-1} b$$

$$\boxed{\begin{aligned} F_{JI} &= D^{-1}(L+U) \\ F_{GS} &= (D+L)^{-1} U \end{aligned}}$$

A must be diagonally dominant

$$|a_{i,i}| \geq \sum_{j \neq i} |a_{i,j}|$$

Do they converge?

$x = Fx + c$ ← exact

$Fx + c - x = 0$

- Fixed-point iteration

$$\mathbf{x}^{(\ell+1)} = \mathbf{F}\mathbf{x}^{(\ell)} + \mathbf{c}$$

- Express iterate as combination of exact solution and error $x^{(\ell)} = x + \varepsilon^{(\ell)}$

$$\cancel{x} + \varepsilon^{(\ell+1)} = F(\cancel{x} + \varepsilon^{(\ell)}) \cancel{+ c} \Rightarrow \varepsilon^{(\ell+1)} = F\varepsilon^{(\ell)}$$

- If the method converges then:

$$\lim_{\ell \to \infty} \varepsilon^{(\ell)} = 0$$

Recall: Eigendecomposition of a Matrix

$Av = \lambda v$ λ - eigenvalue
v - eigenvector

A is square and diagonalizable

$$A \left[v_1 \,\middle|\, v_2 \,\middle|\, \cdots \,\middle|\, v_n \right] = \left[v_1 \,\middle|\, v_2 \,\middle|\, \cdots \,\middle|\, v_n \right] \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{bmatrix}$$

$$AQ \quad Q^{-1} = Q \Lambda Q^{-1}$$

$$\boxed{A = Q \Lambda Q^{-1}}$$

Q is orthonormal

$Q^T = Q^{-1}$

$v_i^T v_j = 0 \quad j \neq i$

$\|v_i\| = 1$

Conditions for Convergence

$$\varepsilon^{(\ell+1)} = F\varepsilon^{(\ell)}$$

$$\lim_{\ell\to\infty} \underline{\varepsilon}^{(\ell)} = 0$$

$$\varepsilon^{(1)} = F\varepsilon^{(0)} \qquad \|\varepsilon^{(0)}\| \neq 0$$

$$\varepsilon^{(2)} = F\varepsilon^{(1)} = F(F\varepsilon^{(0)}) = F^2\varepsilon^{(0)}$$

$$\underline{\varepsilon}^{(\ell)} = \underline{\underline{F}}^{\ell}\underline{\varepsilon}^{(0)} \qquad \varepsilon^{(\ell)} \to 0 \quad \text{iff} \quad \lim_{\ell\to\infty} F^{\ell} = 0 \qquad F = Q\Lambda Q^{-1}$$

$$\lim_{\ell\to\infty}(Q\Lambda Q^{-1})^{\ell} \Leftrightarrow \lim_{\ell\to\infty} Q\Lambda^{\ell}Q^{-1} \Rightarrow \lim_{\ell\to\infty}\Lambda^{\ell} = \max_i|\lambda_i| < 1$$

$$(Q\Lambda\cancel{Q^{-1}})(\cancel{Q}\Lambda Q^{-1}) = Q\Lambda^2 Q^{-1} \Rightarrow Q\Lambda^{\ell}Q^{-1}$$

$$\Lambda = \begin{bmatrix} \lambda_1 & & & 0 \\ & \lambda_2 & & \\ & & \ddots & \\ 0 & & & \lambda_n \end{bmatrix}$$

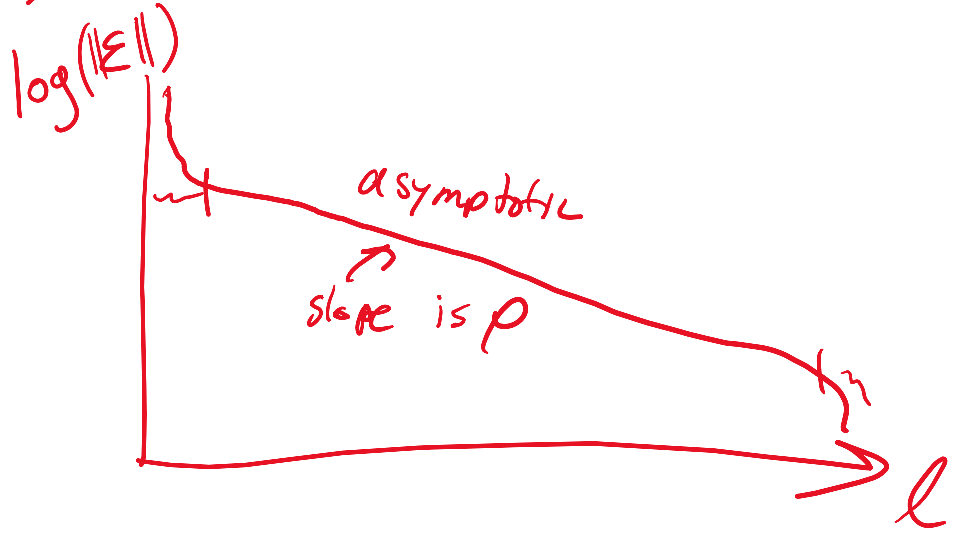
More about the Spectral Radius

$$\rho(A) = \max_i |\lambda_i| < 1$$

For stationary/Fixed point methods, $\rho(A)$, represents the asymptotic rate of convergence

$$?? \; \rho_{GS} = \frac{\rho_{JI}}{2} \; ??$$

$$\rho \approx \frac{\| \varepsilon^{(\ell+1)} \|}{\| \varepsilon^{(\ell)} \|}$$



Number of Iterations and Spectral Radius

$\varepsilon \rightarrow$ true error

$r^{(0)} = \| Ax^{(0)} - b \| \rightarrow O(1)$

$\varepsilon_{crit} = 10^{-5}$

$\frac{r^{(\ell)}}{r^{(0)}}$

$\rho \rightarrow$ this represents the factor by which the error is reduced in a single iteration

$\rho^{\mathcal{L}} = \varepsilon_{crit} = 10^{-5}$

$\rho = 0.99 \rightarrow \mathcal{L} \approx 1{,}100$ iterations

$\rho = 0.3 \rightarrow \mathcal{L} \approx \sim 10$ iterations

$\frac{r^{(\ell)}}{r^{(\ell-1)}} \approx \tilde{\rho}^{(\ell)}$

Successive Over-Relaxation

$$x^{(\ell+1)} = \omega\left[Fx^{(\ell)} + c\right] + (1-\omega)\,x^{(\ell)}$$

$$0 < \omega < 2$$

$$\omega_{opt} \approx \frac{2}{1+\sqrt{1-\rho^2}}$$

$\underbrace{\text{determine } x^{(\ell+1)} \text{ using some linear combination of previous iterates}}$

$\swarrow$ Anderson Acceleration

$\searrow$ Iterative Subspace

Summary of Classical Iteration Schemes

- Implementations are very simple
- Error properties are very well understood
- Generally slowly converging in practical problems

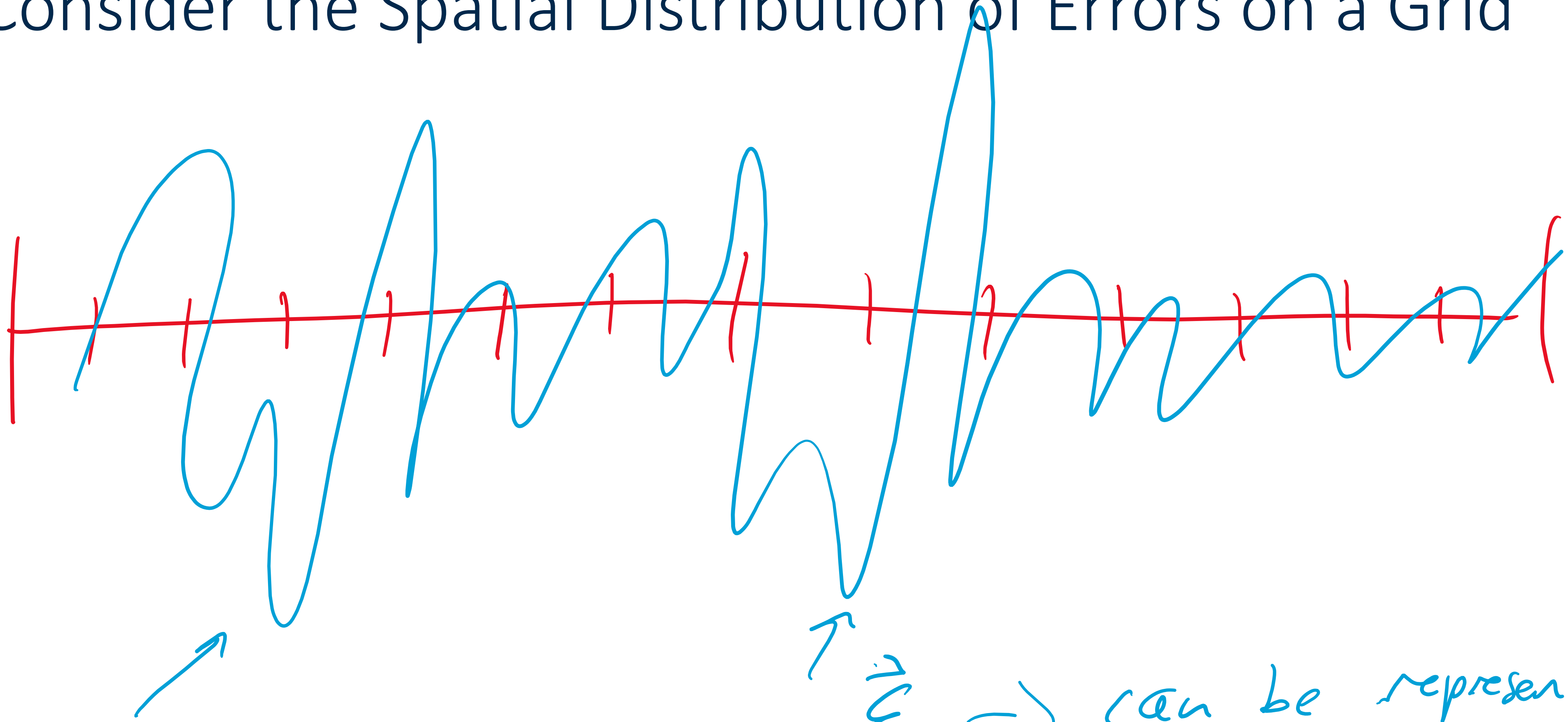
  ☆ several things can be done to improve ☆

- Good for simple problems

Multigrid Methods

Briggs, Henson, and McCormick et al., A Multigrid Tutorial, 2nd Ed., SIAM Press (2000). https://doi.org/10.1137/1.9780898719505

Consider the Spatial Distribution of Errors on a Grid



$\vec{\varepsilon} \Rightarrow$ can be represented by a Fourier series

Multigrid Methods

Images from: Briggs, Henson, and McCormick et al.,
A Multigrid Tutorial, 2nd Ed., SIAM Press (2000).

- Logical extension to classical methods that arises from error analysis.
  - Consider "shape" of error → frequency transform

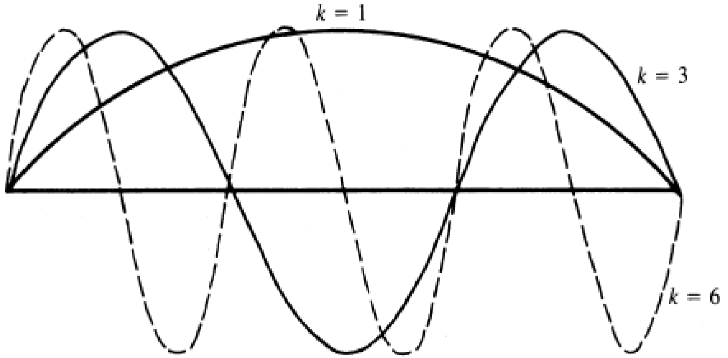


Figure 2.2: *The modes* $v_j = \sin\left(\frac{jk\pi}{n}\right)$, $0 \le j \le n$, *with wavenumbers* $k = 1, 3, 6$. *The kth mode consists of* $\frac{k}{2}$ *full sine waves on the interval.*

Low Frequency

Classical methods very good at "smoothing" high-frequency errors

Real Problem (multiple modes)

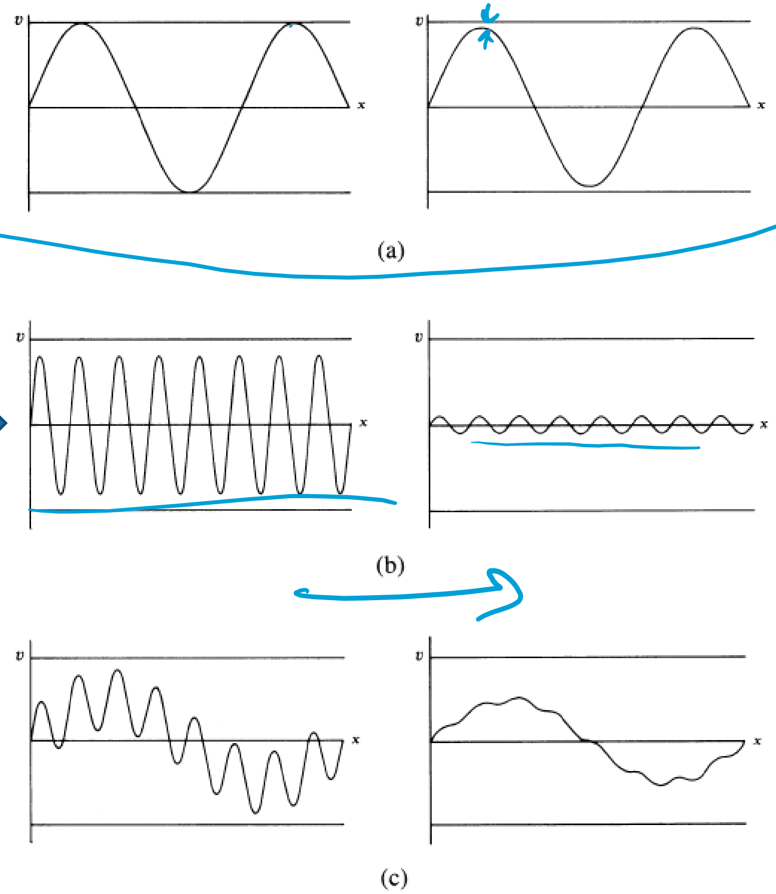


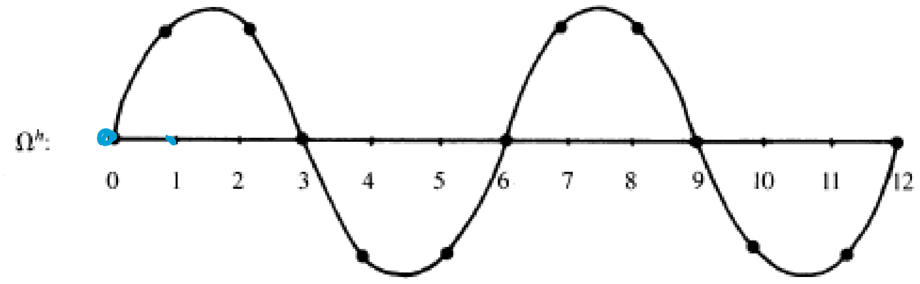
Figure 2.9: *Weighted Jacobi method with* $\omega = \frac{2}{3}$ *applied to the one-dimensional model problem with* $n = 64$ *points and with an initial guess consisting of* (a) $\mathbf{w}_3$, (b) $\mathbf{w}_{16}$, *and* (c) $(\mathbf{w}_2 + \mathbf{w}_{16})/2$. *The figures show the approximation after one iteration (left side) and after 10 iterations (right side).*

smother: GS, JI

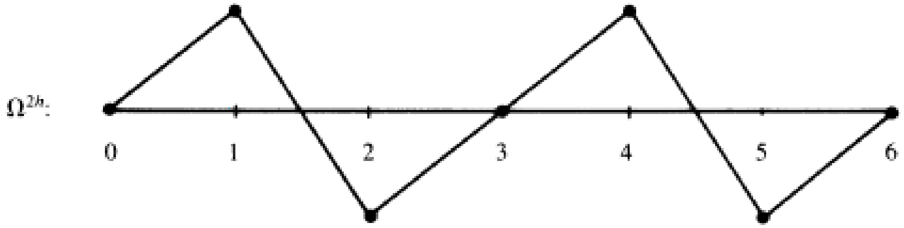
Multigrid Methods (2)

Images from: Briggs, Henson, and McCormick et al., A Multigrid Tutorial, 2nd Ed., SIAM Press (2000).

- Central idea of multigrid is to "map" errors onto coarser grids
  - A low-frequency error on a fine-grid is a high-frequency error on a coarse-grid!
- Recipe for Multigrid includes
  - How to map error from fine-grid to coarse-grid?
    - restriction operator (e.g. bi-linear average)
  - How to smooth error on each grid?
    - classical iteration scheme
  - How to correct error in fine-grid from coarse grid?
    - interpolation operator (e.g. linear interpolate)
  - How to traverse grids?



$k = 4$ wave on $n = 12$ grid



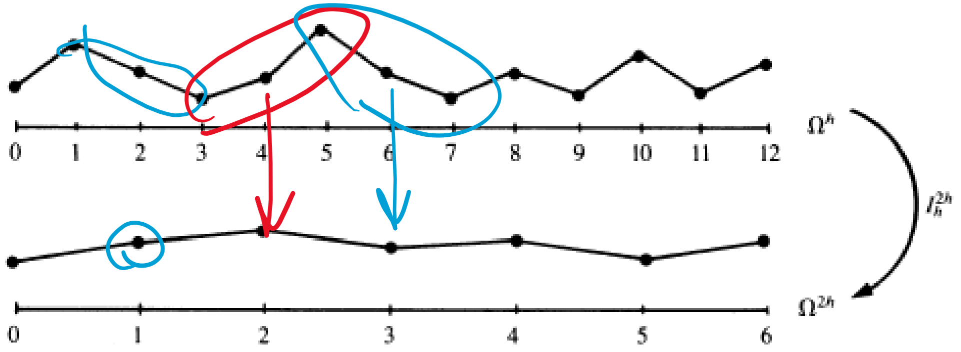
$k = 4$ wave on $n = 6$ grid

Multigrid: Restriction and Interpolation

Images from: Briggs, Henson, and McCormick et al., A Multigrid Tutorial, 2nd Ed., SIAM Press (2000).

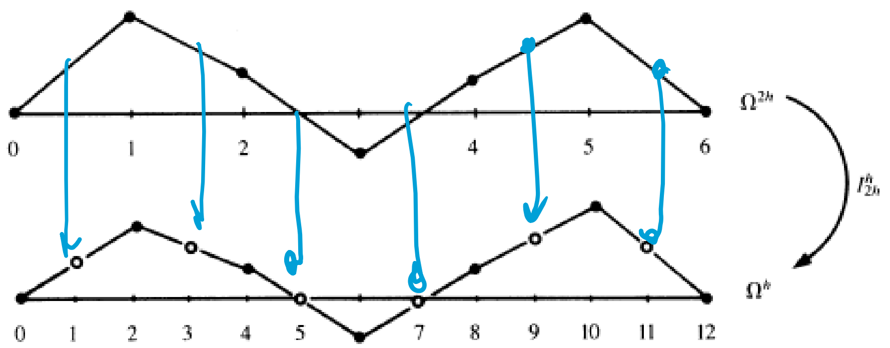
Restriction

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



Interpolation

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

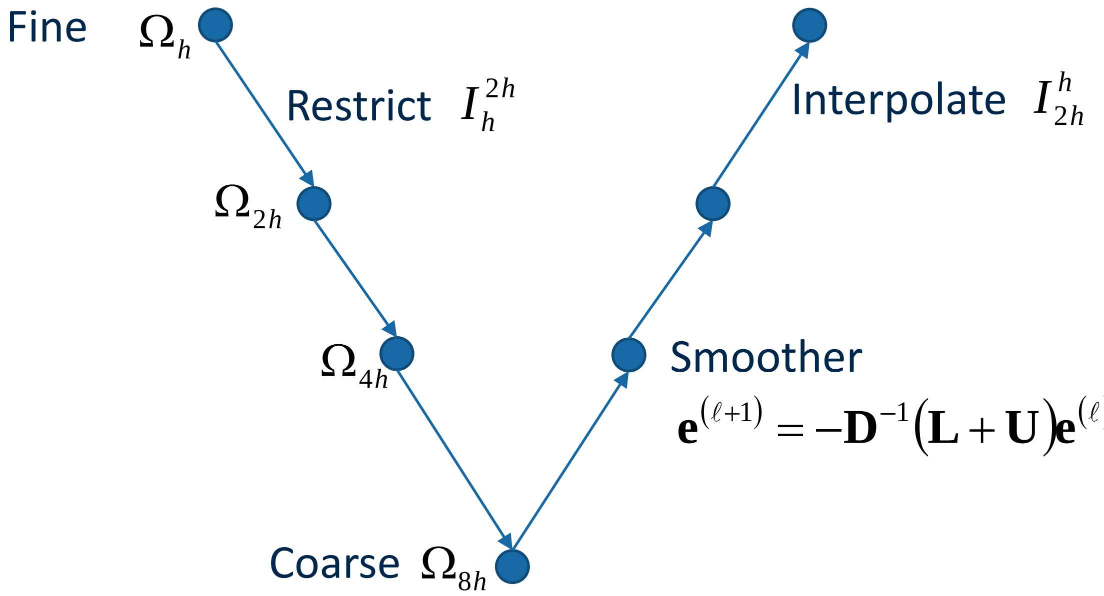


Multigrid: Traversing the Grids

Images from: Briggs, Henson, and McCormick et al., A Multigrid Tutorial, 2nd Ed., SIAM Press (2000).

10,000

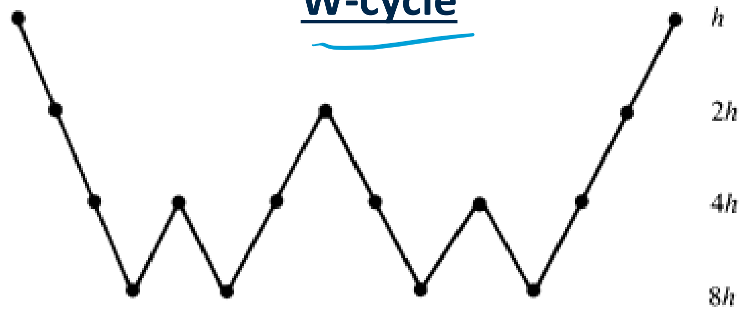
V-cycle



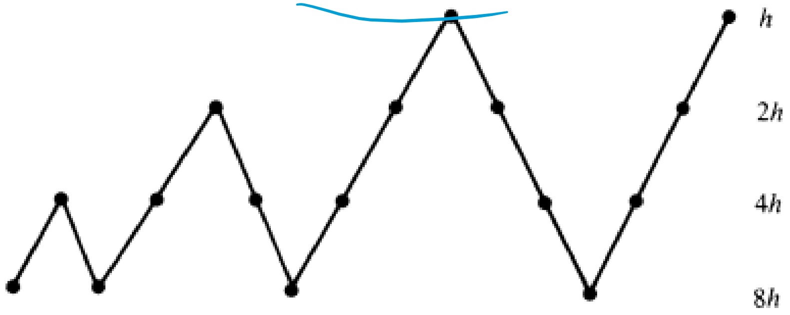
$$\mathbf{e}^{(\ell+1)} = -\mathbf{D}^{-1}(\mathbf{L}+\mathbf{U})\mathbf{e}^{(\ell)} + \mathbf{D}^{-1}\mathbf{r}$$

100 x 100

W-cycle



Full Multigrid



Summary of Multigrid

- Very good for elliptic problems – $O(N)$ work to solve
- A type of fixed point iteration
  - May be analyzed via Fourier/Von Neumann Analysis for asymptotic convergence
- Builds on traditional classical fixed point iterative techniques
  - Uses same elements and adds a few more (interpolation/prolongation)
- Lots of parameters in the iteration that can be "tuned"
- Good for structured grids and finite differenced or finite volume "disc" (e.g. discretized operator is a stencil)
- Can be generalized to algebraic multi-grid (AMG)

Jupyter Notebook Example