

An Introduction to Multigrid

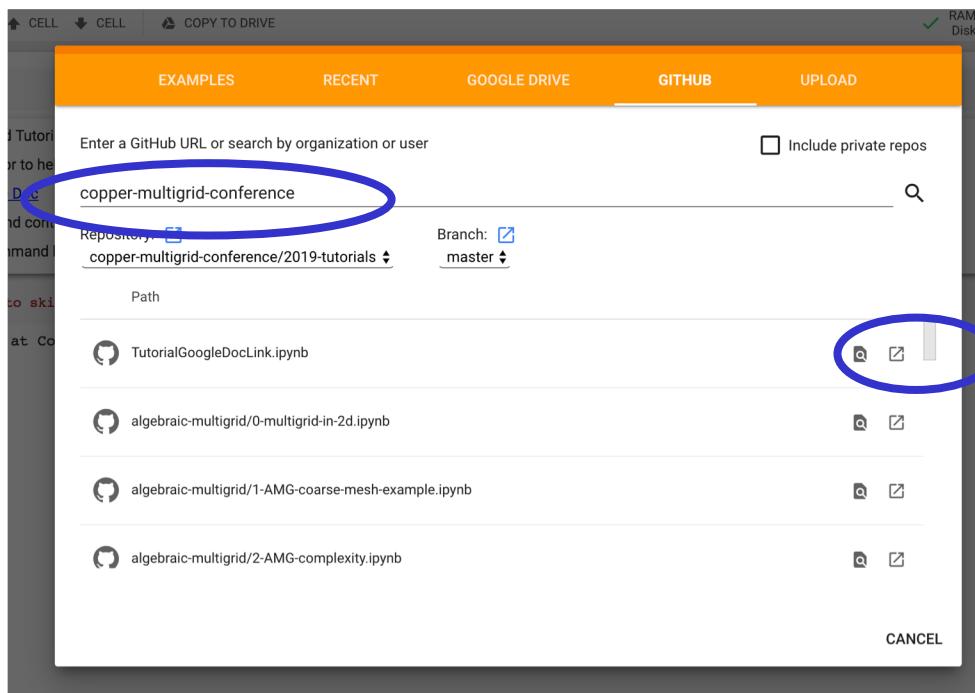
Based on Tutorials from Van Henson, Rob Falgout and Irad Yavneh

J. David Moulton
Applied Mathematics and Plasma Physics
Los Alamos National Laboratory

Google Doc and Jupyter notebooks

Goto Google's colab.research.google.com

- File -> Open new Notebook, then choose GITHUB
- Enter copper-multigrid-conference
- Open the TutorialGoogleDocLink.ipynb in a new tab.



Outline

- Motivation
- A Model Problem
- Basic Iterative Schemes
- Development of Multigrid
 - Complementarity of smoothing and coarse-grid correction
 - Restriction and Interpolation
 - Standard cycles: V, W, FMG
- Anisotropic Problems
- Variable Coefficient and Discontinuous Coefficient Problems
- Nonlinear Problems

Suggested Reading

- Brandt, "Multi-level Adaptive Solutions to Boundary Value Problems," *Math Comp.*, 31, 1977, pp 333-390.
- Brandt, "1984 Guide to Multigrid Development, with applications to computational fluid dynamics."
- Briggs, Henson, and McCormick, "A Multigrid Tutorial, 2nd Edition," SIAM publications, 2000.
- Hackbusch, Multi-Grid Methods and Applications," 1985.
- Hackbusch and Trottenberg, "Multigrid Methods," Springer-Verlag, 1982
- Stüben and Trottenberg, "Multigrid Methods," 1987.
- Trottenberg, Oosterlee, and Schüller, "Multigrid," Academic Press, 2000
- Wesseling, "An Introduction to Multigrid Methods," Wiley, 1992

Multilevel methods have been developed for...

- Elliptic PDEs, CFD, porous media, elasticity, electromagnetics
- Purely algebraic problems, with no physical grid; for example, network and geodetic survey problems.
- Image reconstruction and tomography
- Optimization (e.g., the travelling salesman and long transportation problems)
- Statistical mechanics, Ising spin models.
- Quantum chromodynamics.
- Quadrature and generalized FFTs.
- Integral equations.

Model Problem

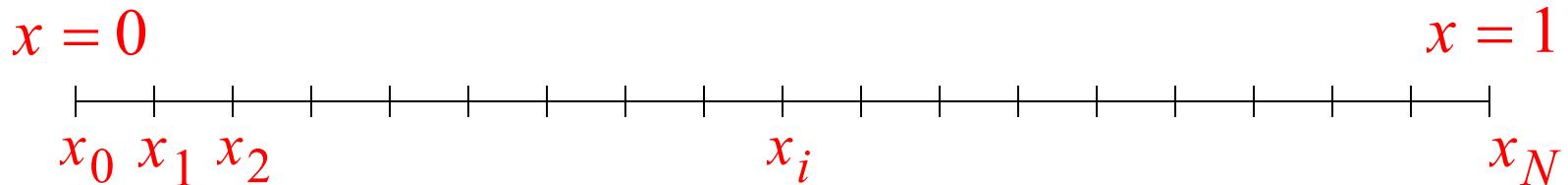
Model Problem Discretization

- One-dimensional boundary value problem:

$$-u''(x) + \sigma u(x) = f(x) \quad 0 < x < 1, \quad \sigma > 0$$

$$u(0) = u(1) = 0$$

- Grid: $h = \frac{1}{N}$, $x_i = ih$, $i = 0, 1, \dots N$



- Let $v_i \approx u(x_i)$ and $f_i \approx f(x_i)$ for $i = 0, 1, \dots N$

This discretizes the variables, but what about the equations?

Approximate $u''(x)$ via Taylor series

- Approximate 2nd derivative using Taylor series:

$$u(x_{i+1}) = u(x_i) + h u'(x_i) + \frac{h^2}{2!} u''(x_i) + \frac{h^3}{3!} u'''(x_i) + O(h^4)$$

+

$$u(x_{i-1}) = u(x_i) - h u'(x_i) + \frac{h^2}{2!} u''(x_i) - \frac{h^3}{3!} u'''(x_i) + O(h^4)$$

- Summing & solving:

$$u''(x_i) = \frac{u(x_{i+1}) - 2u(x_i) + u(x_{i-1})}{h^2} + O(h^2)$$

We approximate the equation with a finite difference scheme

- We approximate the BVP

$$-u''(x) + \sigma u(x) = f(x) \quad 0 < x < 1, \quad \sigma > 0$$

$$u(0) = u(1) = 0$$

with the finite difference scheme:

$$\frac{-v_{i-1} + 2v_i - v_{i+1}}{h^2} + \sigma v_i = f_i \quad i = 1, 2, \dots N-1$$

$$v_0 = v_N = 0$$

The discrete model problem

- Letting $\mathbf{v} = (v_1, v_2, \dots, v_{N-1})^T$ and

$$\mathbf{f} = (f_1, f_2, \dots, f_{N-1})^T$$

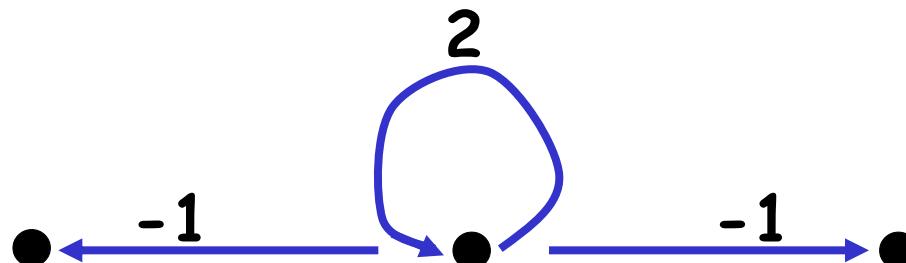
we obtain the matrix equation $A \mathbf{v} = \mathbf{f}$ where A is $(N-1) \times (N-1)$, symmetric, positive definite, and

$$A = \frac{1}{h^2} \begin{pmatrix} 2+\sigma h^2 & -1 & & & \\ -1 & 2+\sigma h^2 & -1 & & \\ & -1 & 2+\sigma h^2 & -1 & \\ & & -1 & 2+\sigma h^2 & -1 \\ & & & -1 & 2+\sigma h^2 \end{pmatrix}, \quad \mathbf{v} = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ \vdots \\ v_{N-2} \\ v_{N-1} \end{pmatrix}, \quad \mathbf{f} = \begin{pmatrix} f_1 \\ f_2 \\ f_3 \\ \vdots \\ f_{N-2} \\ f_{N-1} \end{pmatrix}$$

Stencil Notation

$$A = [-1 \ 2 \ -1]$$

dropping h^{-2} and σ for convenience



Basic Iterative Methods

Basic Solution Methods

- Direct
 - Gaussian elimination
 - Factorization
 - Fast Poisson Solvers (FFT-based, reduction-based, ...)
- Iterative
 - Richardson, Jacobi, Gauss-Seidel, ...
 - Steepest Descent, Conjugate Gradient, ...
 - Incomplete Factorization, ...
- Note: This simple 1-D model problem can be solved very efficiently in several ways. Pretend it can't, and that it is very hard, because it shares many characteristics with some very hard problems.

Iterative Methods for Linear Systems

- Consider $Au = f$ where A is $N \times N$ and let v be an approximation to u .
- Two important measures:
 - The Error: $e = u - v$, with norms

$$\|e\|_\infty = \max |e_i| \quad \|e\|_2 = \sqrt{\sum_{i=1}^N e_i^2}$$

- The Residual: $r = f - Av$ with

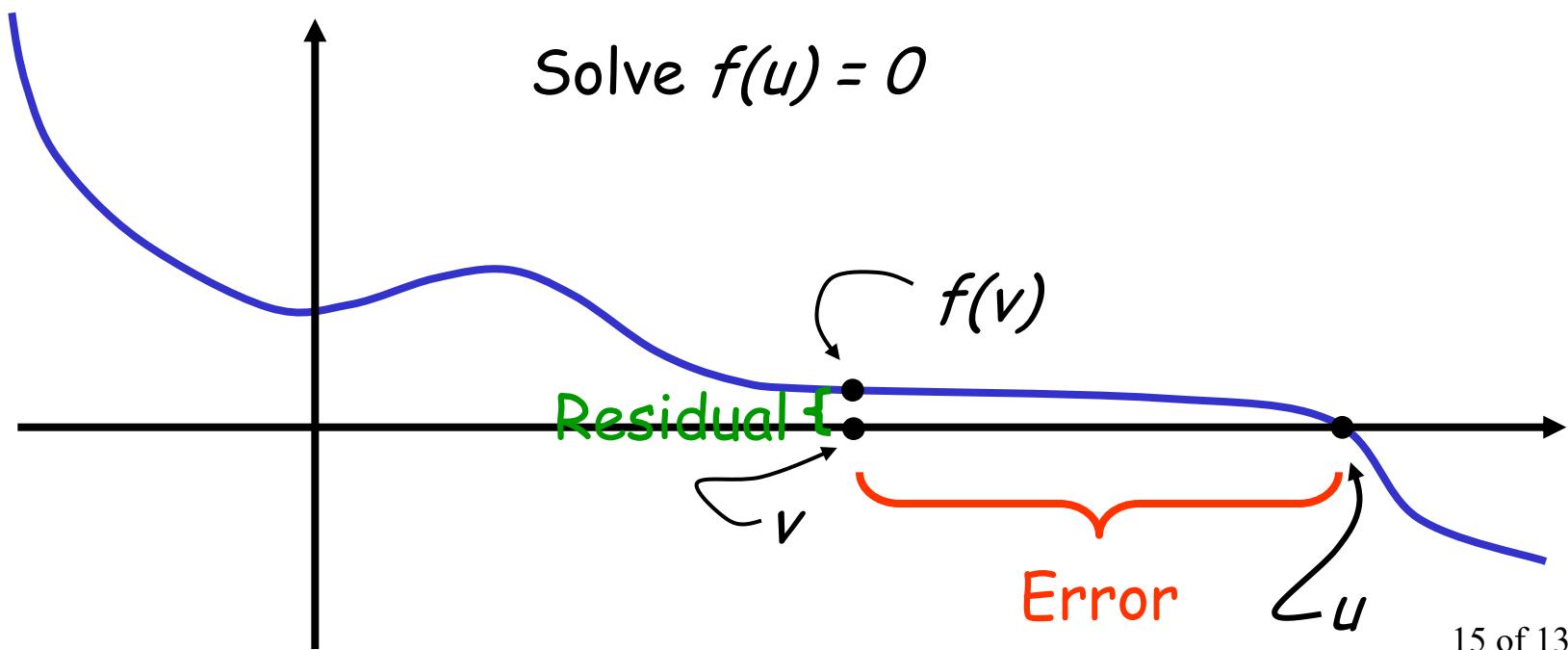
What does
 r measure???

Why have both
 r & e ???

$$\|r\|_\infty \quad \|r\|_2$$

Residual versus error

- Residual measures "by how much does a guess fail to solve the system"
- Error measures "by how much does a guess differ from the solution"



Residual correction

- Since $e = u - v$, we can write $Au = f$ as $A(v + e) = f$

which means that $Ae = f - Av \equiv r$.

$$\begin{aligned}r &= f - Av \\&= Au - Av \\&= A(u - v) \\&= Ae\end{aligned}$$

- Residual Equation:

$$Ae = r$$

- Residual Correction:

$$u = v + e$$

What does this do for us???

Relaxation Schemes

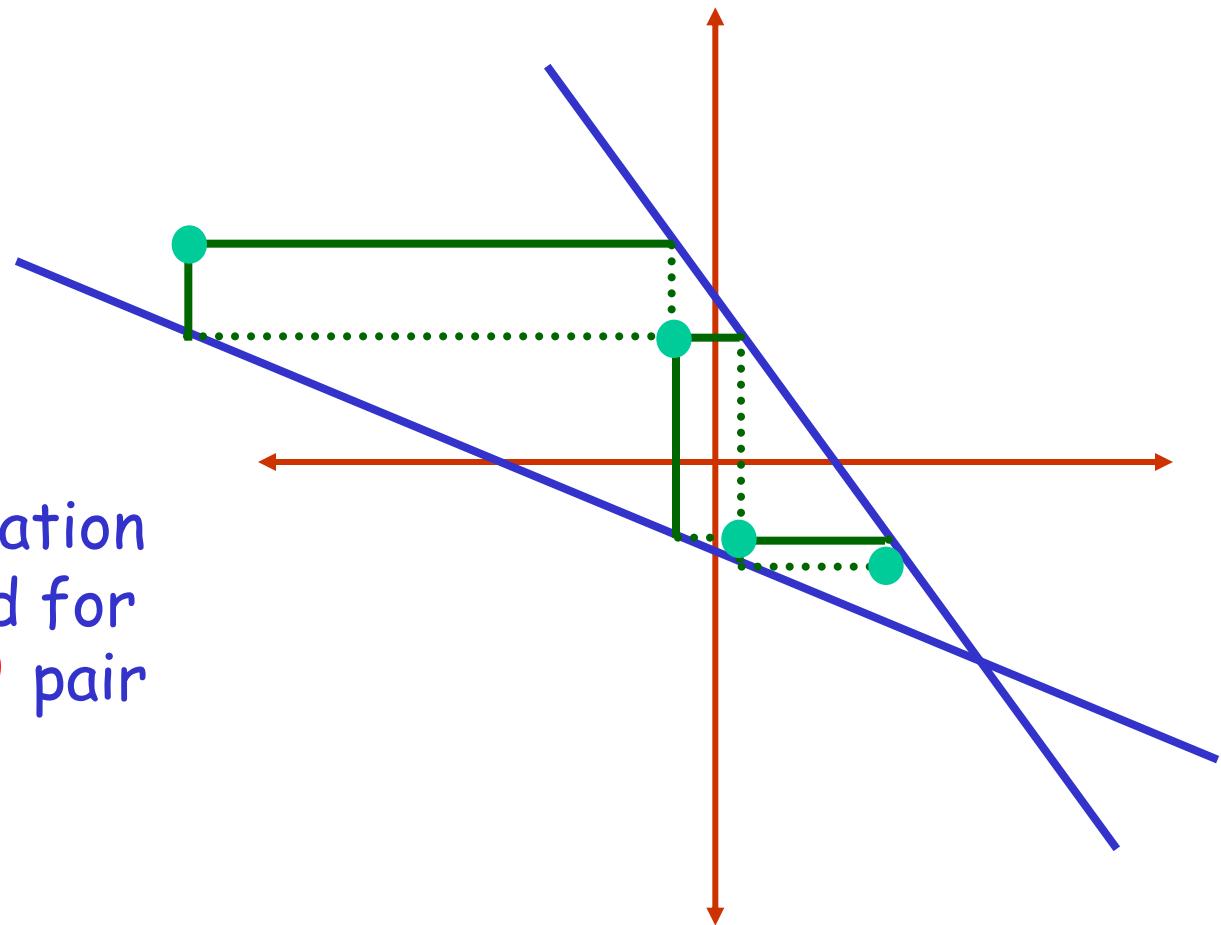
- Consider the 1D model problem

$$-u_{i-1} + 2u_i - u_{i+1} = h^2 f_i \quad 1 \leq i \leq N-1 \quad u_0 = u_N = 0$$

- Jacobi Method (simultaneous displacement): Solve the i^{th} equation for v_i holding other variables fixed:

$$v_i^{(new)} = \frac{1}{2}(v_{i-1}^{(old)} + v_{i+1}^{(old)} + h^2 f_i) \quad 1 \leq i \leq N-1$$

An Example: Jacobi Iteration



Solve the first equation for x , the second for y . The new (x,y) pair forms the next iterate.

In matrix form, the relaxation is

- Let $A = (D - L - U)$ where D is diagonal and L and U are the strictly lower and upper parts of A .
- Then $A u = f$ becomes
$$(D - L - U) u = f$$
$$D u = (L + U) u + f$$
$$u = D^{-1} (L + U) u + D^{-1} f$$
- Let $R_J = D^{-1} (L + U)$, then the iteration is:
$$v^{(new)} = R_J v^{(old)} + D^{-1} f$$

The iteration matrix and the error

- From the derivation,

$$\begin{aligned} u &= D^{-1}(L+U)u + D^{-1}f \\ u &= R_J u + D^{-1}f \end{aligned}$$

- the iteration is

$$v^{(new)} = R_J v^{(old)} + D^{-1}f$$

- subtracting,

$$u - v^{(new)} = R_J u + D^{-1}f - (R_J v^{(old)} + D^{-1}f)$$

- or

$$u - v^{(new)} = R_J u - R_J v^{(old)}$$

- hence

$$e^{(new)} = R_J e^{(old)}$$

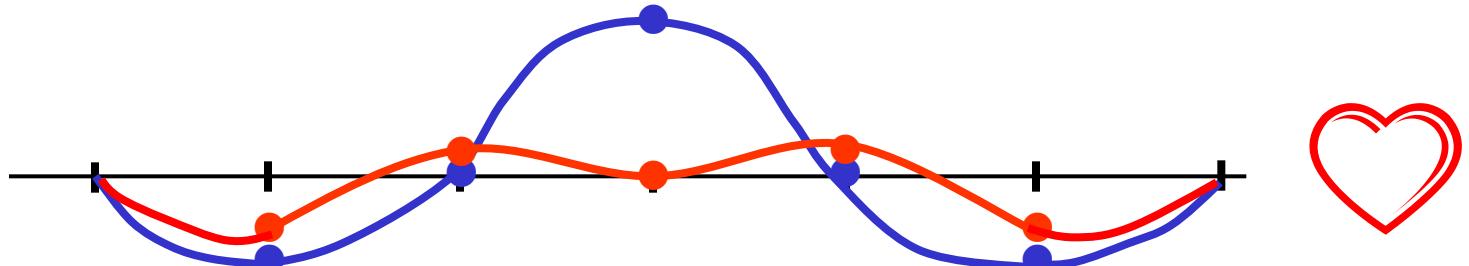
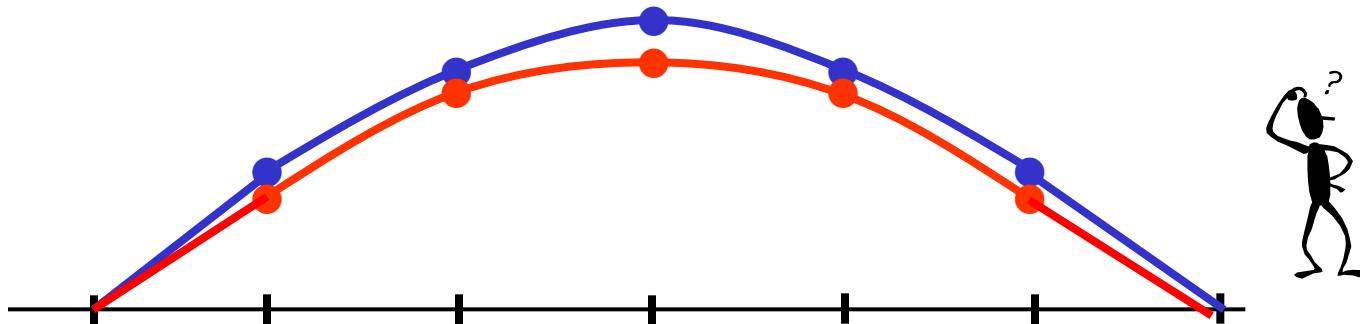
Error propagation!

A Picture

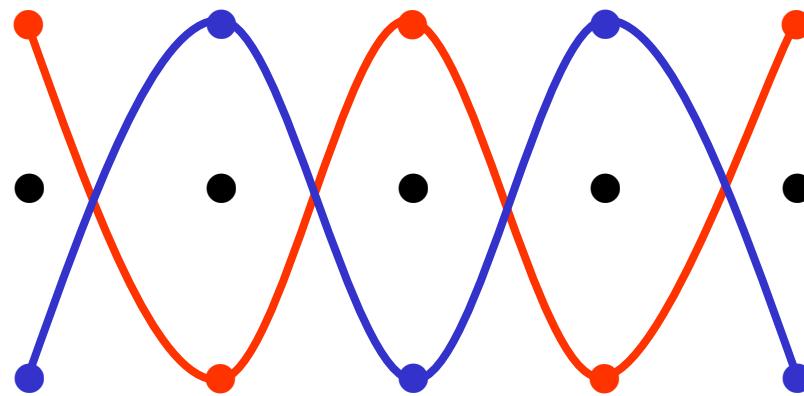
$$R_J = D^{-1} (L + U) = \begin{bmatrix} \frac{1}{2} & 0 & \frac{1}{2} \end{bmatrix}$$

so Jacobi is an **error averaging process**:

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



But...



Weighted Jacobi Relaxation

- Consider the iteration:

$$v_i^{(new)} \leftarrow (1 - \omega) v_i^{(old)} + \frac{\omega}{2} (v_{i-1}^{(old)} + v_{i+1}^{(old)} + h^2 f_i)$$

- Letting $A = D - L - U$, the matrix form is:

$$\begin{aligned} v^{(new)} &= [(1 - \omega)I + \omega D^{-1}(L + U)] v^{(old)} + \omega h^2 D^{-1} f \\ &= R_\omega v^{(old)} + \omega h^2 D^{-1} f \end{aligned} .$$

- Note that

$$R_\omega = [(1 - \omega)I + \omega R_J]$$

- It is easy to see that if $e \equiv u^{(exact)} - u^{(approx)}$, then

$$e^{(new)} = R_\omega e^{(old)}$$

Gauss-Seidel Relaxation (1D)

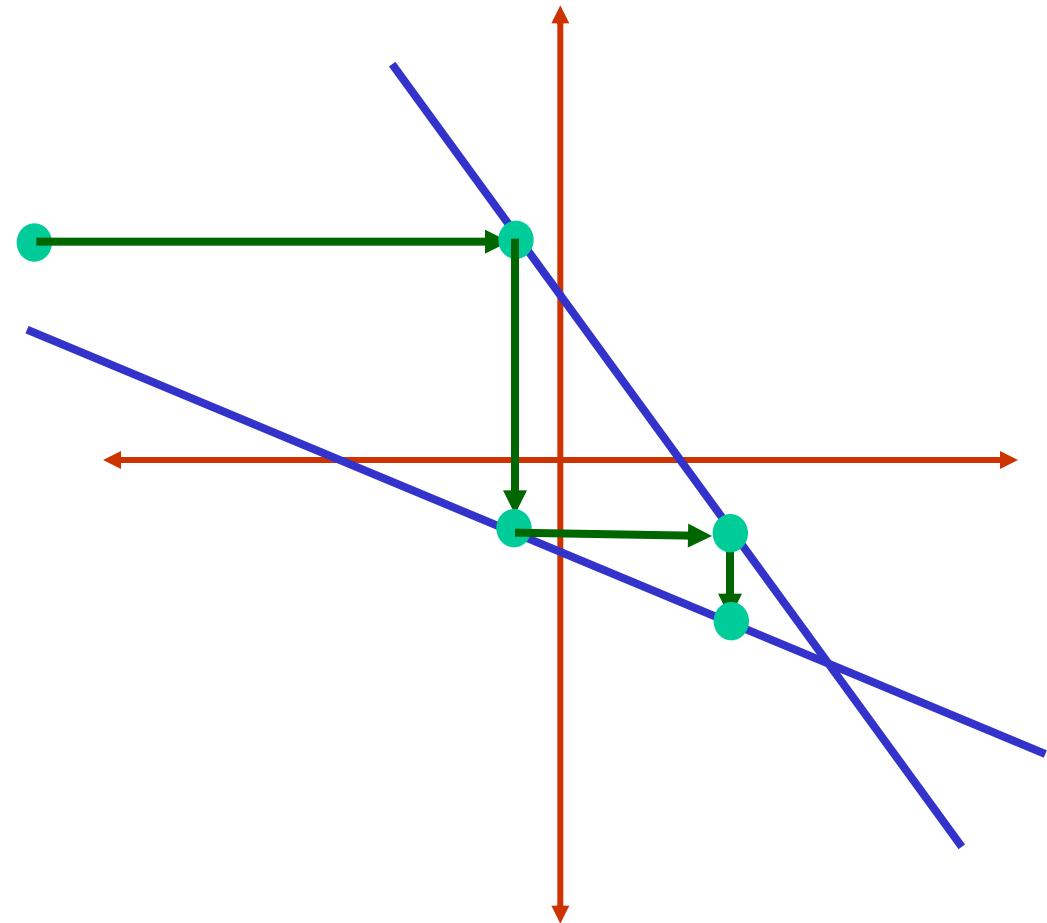
- Solve equation i for u_i and update immediately.
- Equivalently: set each component of r to zero.
- Component form: for $i = 1, 2, \dots, N-1$, set

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

- Matrix form:
$$\begin{aligned} A &= (D - L - U) \\ (D - L) u &= U u + f \\ u &= (D - L)^{-1} U u + (D - L)^{-1} f \end{aligned}$$
- Let $R_G = (D - L)^{-1} U$
- Then iterate $v^{(new)} \leftarrow R_G v^{(old)} + (D - L)^{-1} f$
- Error propagation: $e^{(new)} \leftarrow R_G e^{(old)}$

Gauss-Seidel Iteration

Solve the first equation for x .
Solve the second for y , using the new value for x .
Repeat this process. At all times, the newest available values of the "other" variable are used in solving for the "current" variable.



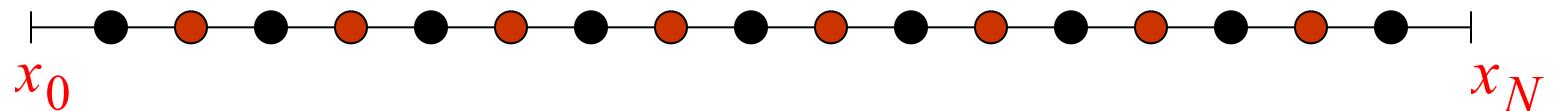
Red-Black Gauss-Seidel

- Update the even (red) points

$$v_{2i} \leftarrow \frac{1}{2}(v_{2i-1} + v_{2i+1} + h^2 f_{2i})$$

- Update the odd (black) points

$$v_{2i+1} \leftarrow \frac{1}{2}(v_{2i} + v_{2i+2} + h^2 f_{2i+1})$$



Test?

$$Au = f$$

Need to know how we're doing!!!

- What f ?

$$Au = 0$$

- What v ?

$$v = \text{rand}$$

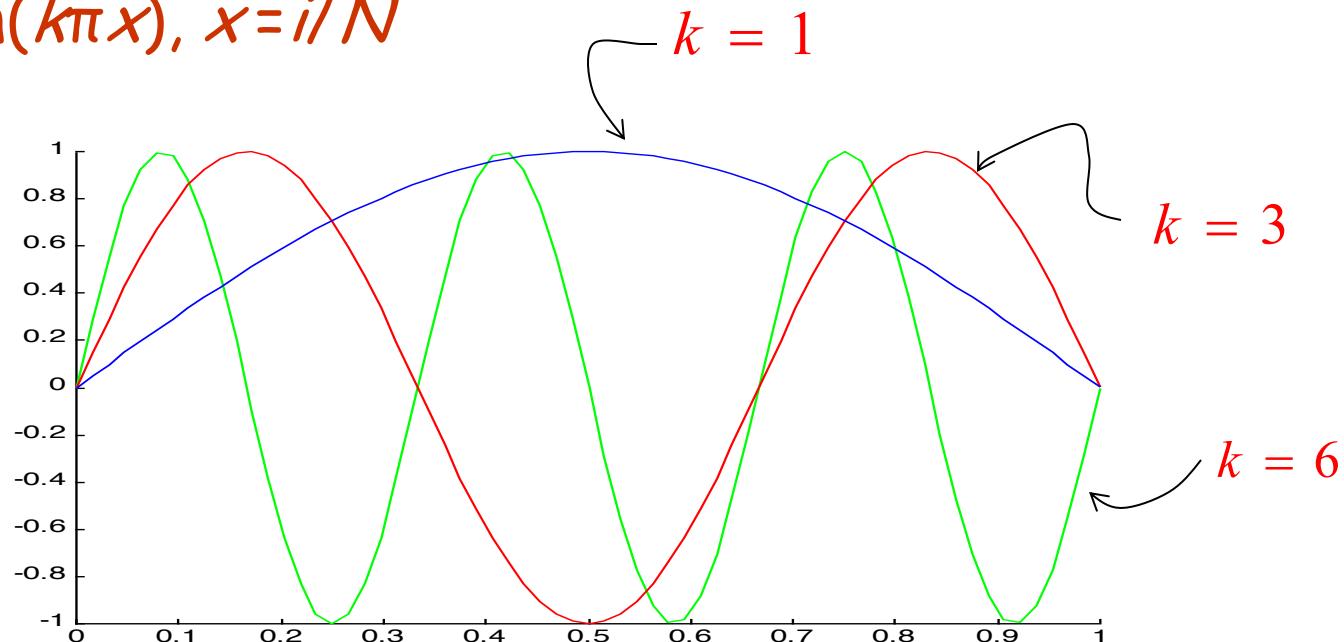
Numerical Experiments

- Solve $Au = 0$, $-u_{i-1} + 2u_i - u_{i+1} = 0$
- Use Fourier modes as initial iterate, with $N=64$:

$$\vec{v}_k = (v_i)_k = \sin\left(\frac{ik\pi}{N}\right) \quad 1 \leq i \leq N-1, \quad 1 \leq k \leq N-1$$

component **mode**

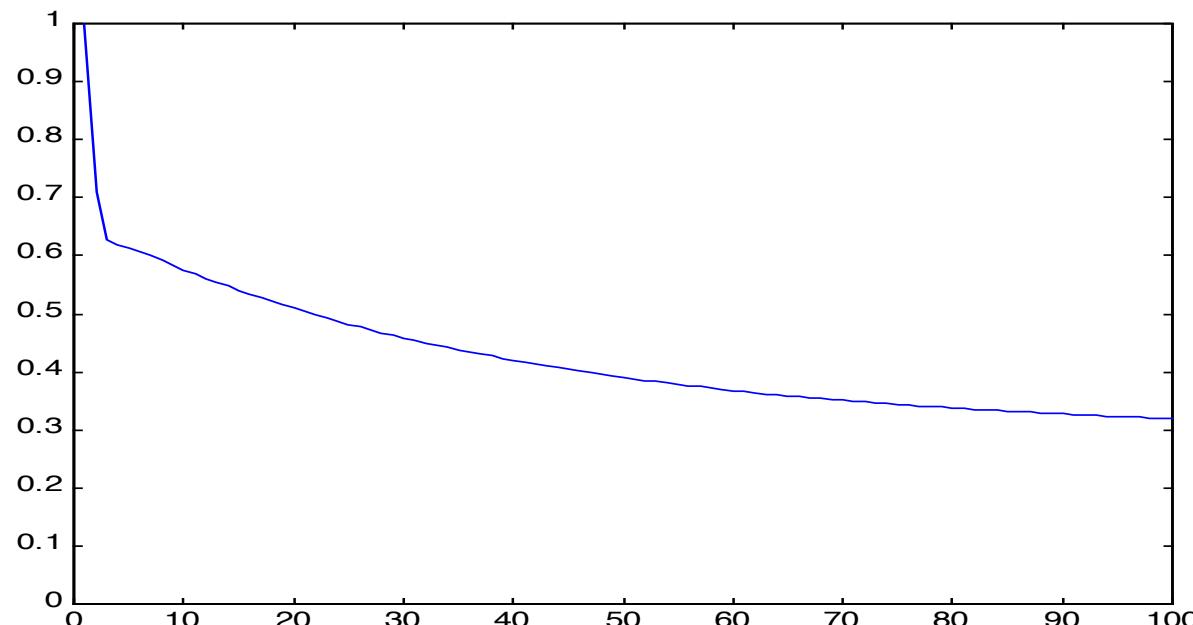
$\sin(k\pi x)$, $x=i/N$



Error reduction stalls

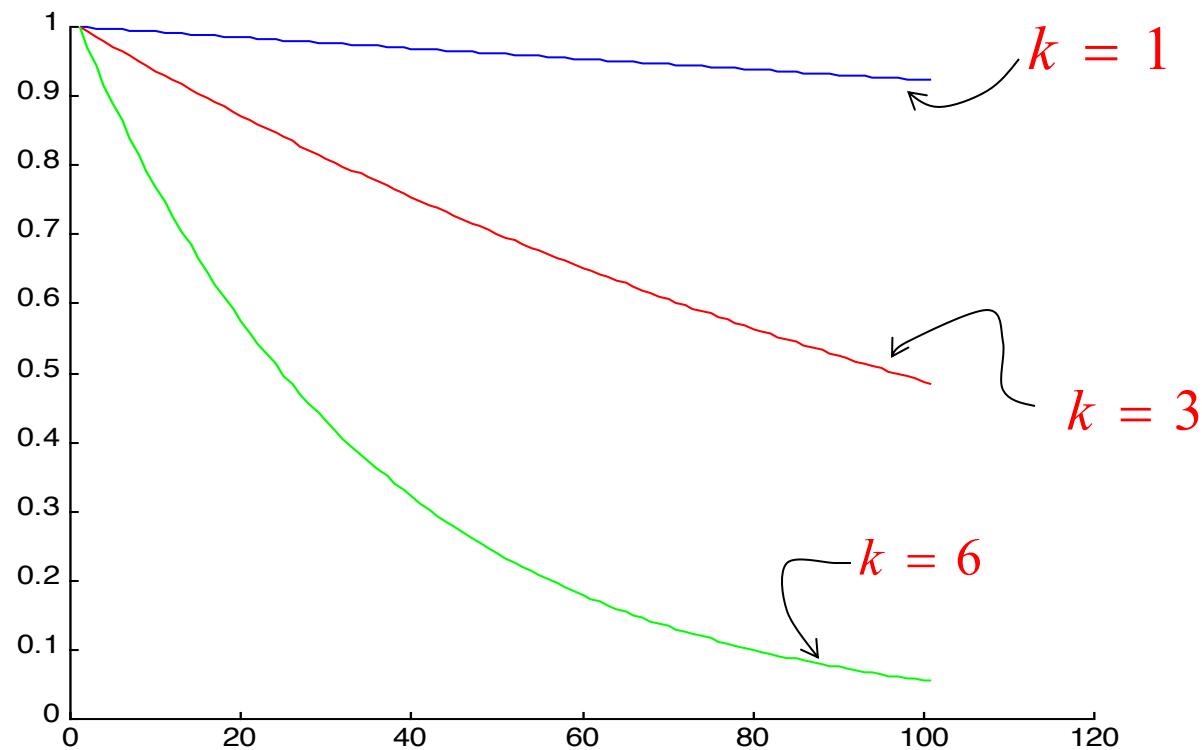
relaxation shoots itself in the foot

- Weighted $\omega = \frac{2}{3}$ Jacobi on 1D problem.
- Initial guess: $v_0 = \frac{1}{3} \left(\sin\left(\frac{j\pi}{N}\right) + \sin\left(\frac{6j\pi}{N}\right) + \sin\left(\frac{32j\pi}{N}\right) \right)$
- Error $\|e\|_\infty$ plotted against iteration number:



Convergence rates differ for different error components

- Error, $\| \mathbf{e} \|_{\infty}$, in weighted Jacobi on $A\mathbf{u} = \mathbf{0}$ for 100 iterations using initial guesses of \mathbf{v}_1 , \mathbf{v}_3 , and \mathbf{v}_6



Analysis of stationary iterations

- Let $v^{(new)} = Rv^{(old)} + g$. The exact solution is unchanged by the iteration, i.e., $u = Ru + g$
- Subtracting, we see that

$$e^{(new)} = Re^{(old)}$$

- Letting e^0 be the initial error and e^i be the error after the i^{th} iteration, we see that after n iterations we have

$$e^{(n)} = R^n e^{(0)}$$

“Fundamental theorem of iteration”

- R is convergent (that is, $R^n \rightarrow 0$ as $n \rightarrow \infty$) if and only if $\rho(R) < 1$, where

$$\rho(R) = \max \{ |\lambda_1|, |\lambda_2|, \dots, |\lambda_N| \}$$

therefore, for any initial vector $v^{(0)}$, we see that $e^{(n)} \rightarrow 0$ as $n \rightarrow \infty$ if and only if $\rho(R) < 1$.

- $\rho(R) < 1$ assures the convergence of the iteration given by R and $\rho(R)$ is called the *convergence factor* for the iteration.

Convergence Rate

- How many iterations are needed to reduce the initial error by 10^{-d} ?

$$\frac{\|e^{(M)}\|}{\|e^{(0)}\|} \leq \|R^M\| \sim (\rho(R))^M \sim 10^{-d}$$

- So, we have $M = \frac{d}{\log_{10}\left(\frac{1}{\rho(R)}\right)}$.
- The *convergence rate* is given:

$$\text{rate} = \log_{10}\left(\frac{1}{\rho(R)}\right) = -\log_{10}(\rho(R)) \frac{\text{digits}}{\text{iteration}}$$

Convergence analysis for weighted Jacobi on 1D model

$$\begin{aligned} R_\omega &= (1-\omega)I + \omega D^{-1}(L + U) \\ &= I - \omega D^{-1}A \end{aligned}$$

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

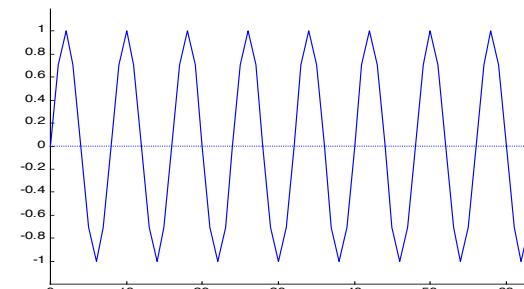
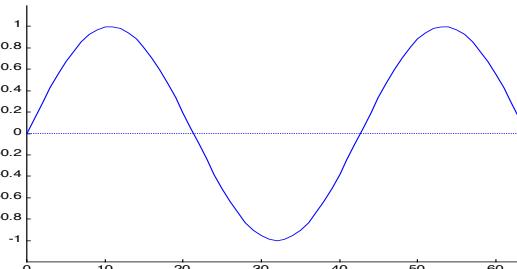
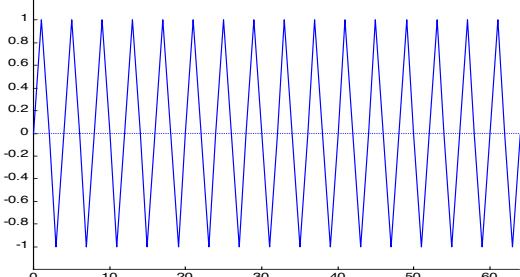
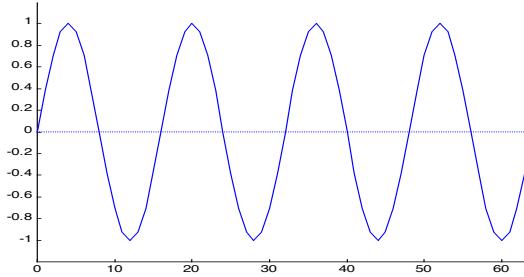
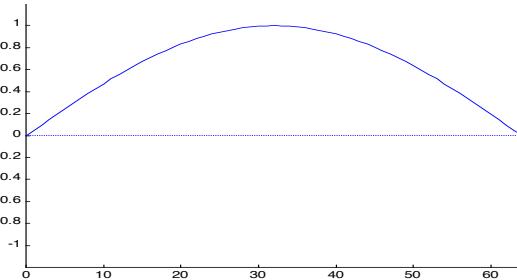
$$\lambda(R_\omega) = 1 - \frac{\omega}{2} \lambda(A)$$

For the 1D model problem, the eigenvectors of the weighted Jacobi iteration and the eigenvectors of the matrix A are the same! The eigenvalues are related as well.

Good exercise: Find the eigenvalues & eigenvectors of A

- Show that the eigenvectors of A are Fourier modes!

$$\lambda_k(A) = 4 \sin^2\left(\frac{k\pi}{2N}\right), \quad w_{k,j} = \sin\left(\frac{jk\pi}{N}\right)$$



Eigenvectors of R_ω and A are the same, the eigenvalues related

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

- Expand the initial error in terms of the eigenvectors:

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

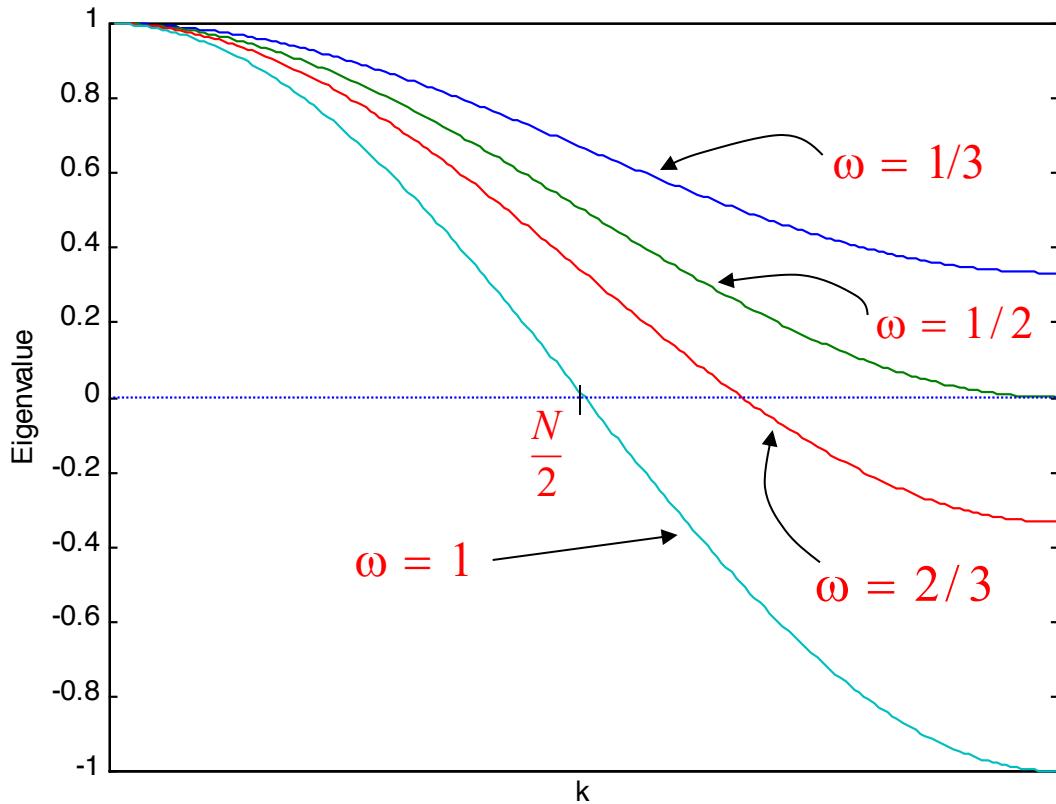
- After M iterations,

$$R^M e^{(0)} = \sum_{k=1}^{N-1} c_k R^M w_k = \sum_{k=1}^{N-1} c_k \lambda_k^M w_k$$

- The k^{th} mode of the error is reduced by λ_k at each iteration

Relaxation suppresses eigenmodes unevenly

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



Note that if $0 \leq \omega \leq 1$
then $|\lambda_k(R_\omega)| < 1$ for
 $k = 1, 2, \dots, N-1$

For $0 \leq \omega \leq 1$,

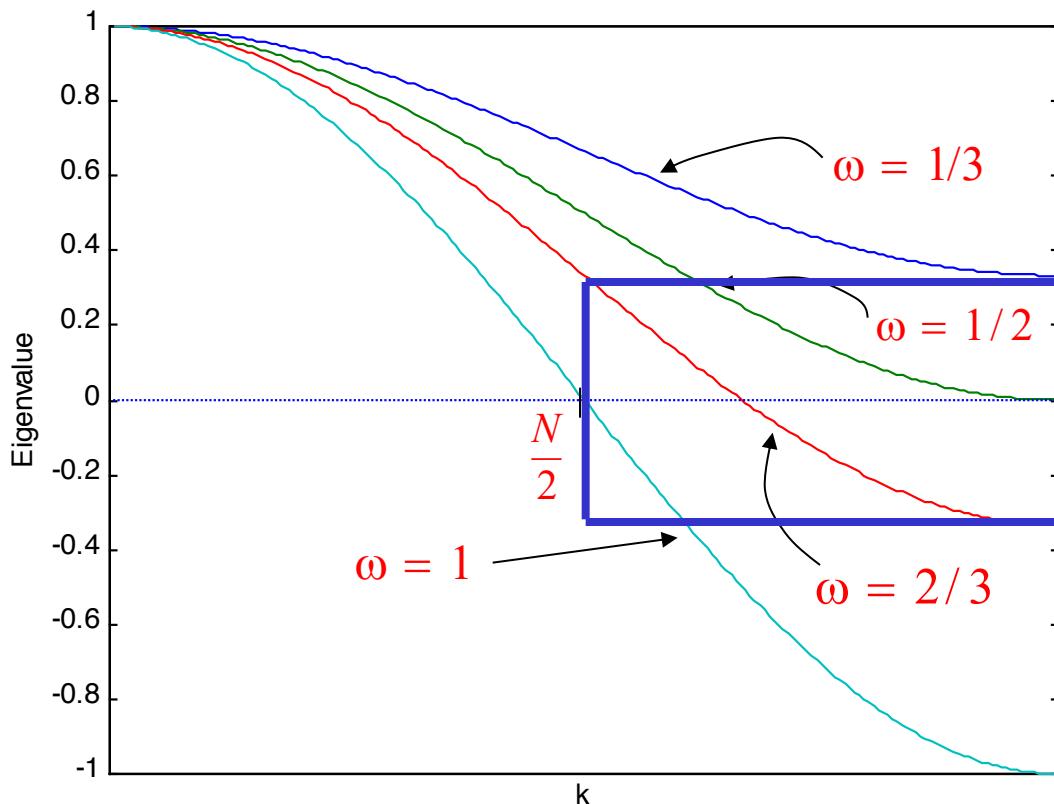
$$\lambda_1 = 1 - 2\omega \sin^2\left(\frac{\pi}{2N}\right)$$

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

$$= 1 - O(h^2) \approx 1$$

Low frequencies are undamped

- Notice that no value of ω will damp out the long (i.e., low frequency) waves.



What value of ω gives the best damping of the short waves ?

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

Choose ω such that

$$\frac{\lambda_N(R_\omega)}{2} = -\lambda_N(R_\omega)$$

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

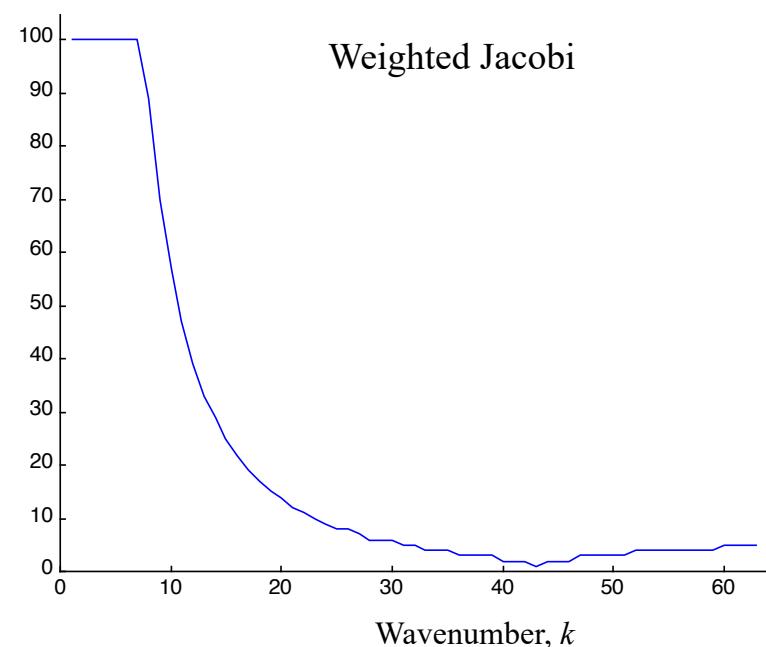
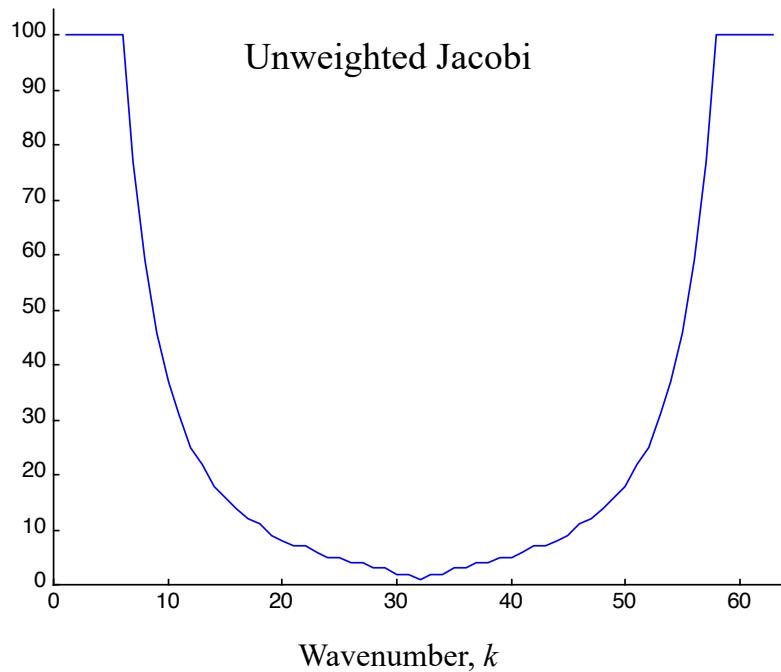
The Smoothing factor

- The smoothing factor is the largest absolute value among the eigenvalues in the upper half of the spectrum of the iteration matrix

$$\text{smoothing factor} = \max \left| \lambda_k(R) \right| \quad \text{for } \frac{N}{2} \leq k \leq N$$

- For R_ω , with $\omega = \frac{2}{3}$, the smoothing factor is $\frac{1}{3}$, since $\left| \frac{\lambda_N}{\frac{N}{2}} \right| = |\lambda_N| = \frac{1}{3}$ and $|\lambda_k| < \frac{1}{3}$ for $\frac{N}{2} < k < N$.
- But, $|\lambda_k| \approx 1 - \frac{2}{3}k^2\pi^2h^2$ for long waves $\left(k \geq \frac{N}{2} \right)$.

Convergence of Jacobi on $Au=0$

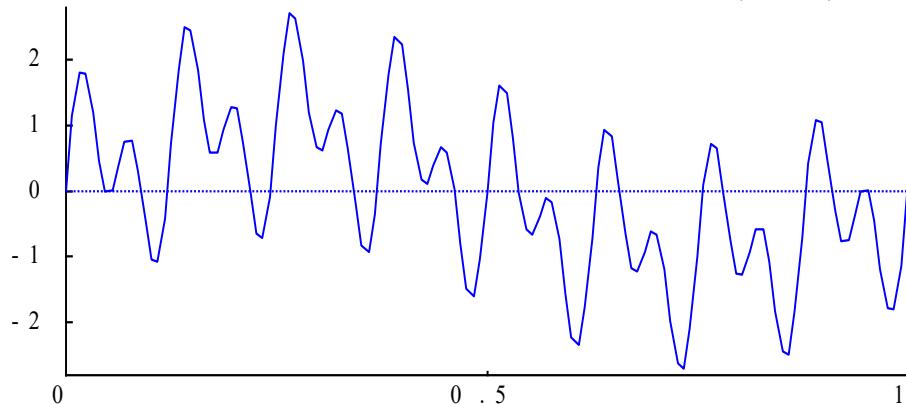


- Jacobi method on $Au=0$ with $N=64$. Number of iterations required to reduce to $\|e\|_\infty < .01$

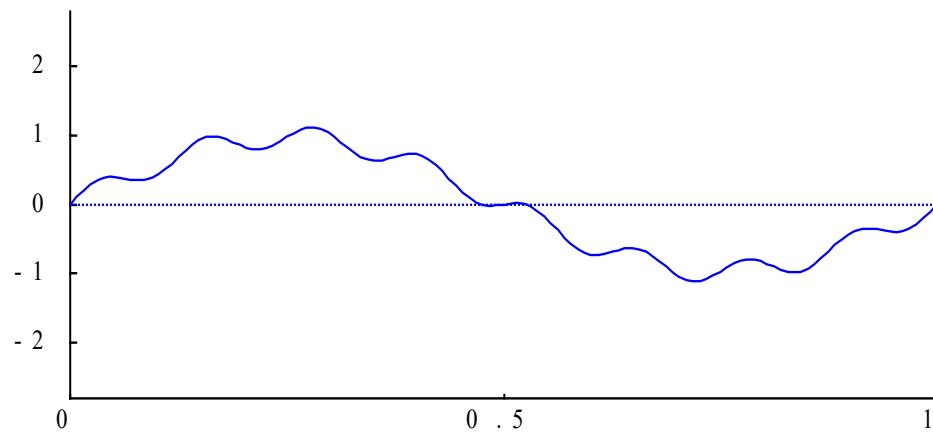
- Initial guess : $v_{kj} = \sin\left(\frac{jk\pi}{N}\right)$

Weighted Jacobi Relaxation Smooths the Error

- Initial error: $v_{kj} = \sin\left(\frac{2j\pi}{N}\right) + \frac{1}{2}\sin\left(\frac{16j\pi}{N}\right) + \frac{1}{2}\sin\left(\frac{32j\pi}{N}\right)$



- Error after 35 iteration sweeps:



Many relaxation schemes have the smoothing property, where oscillatory modes of the error are eliminated effectively, but smooth modes are damped very slowly.

Other relaxations schemes may be analyzed similarly

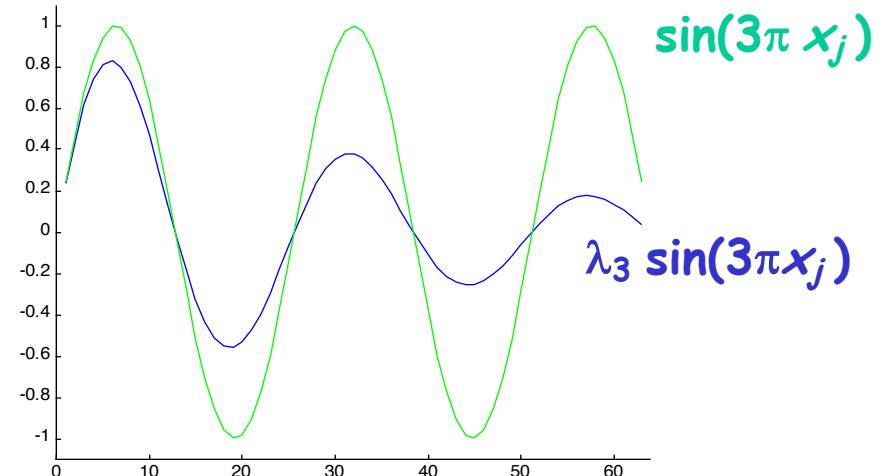
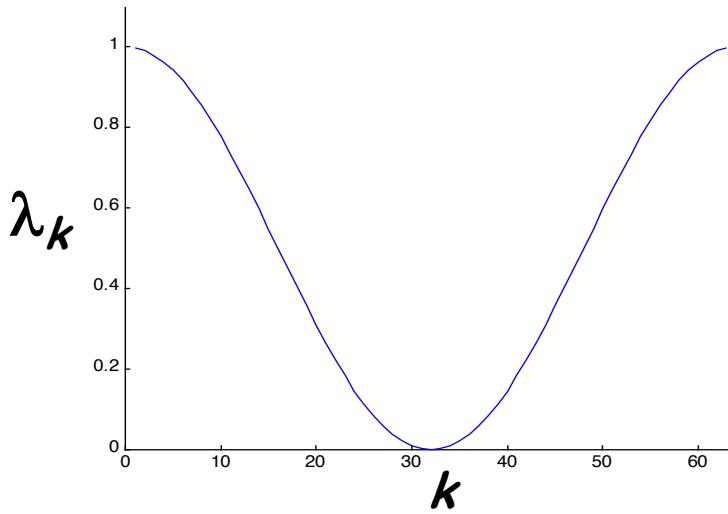
- Gauss-Seidel relaxation applied to the 3-point difference matrix A (1D model problem):

$$R_G = (D - L)^{-1} U$$

- Good exercise: Show that

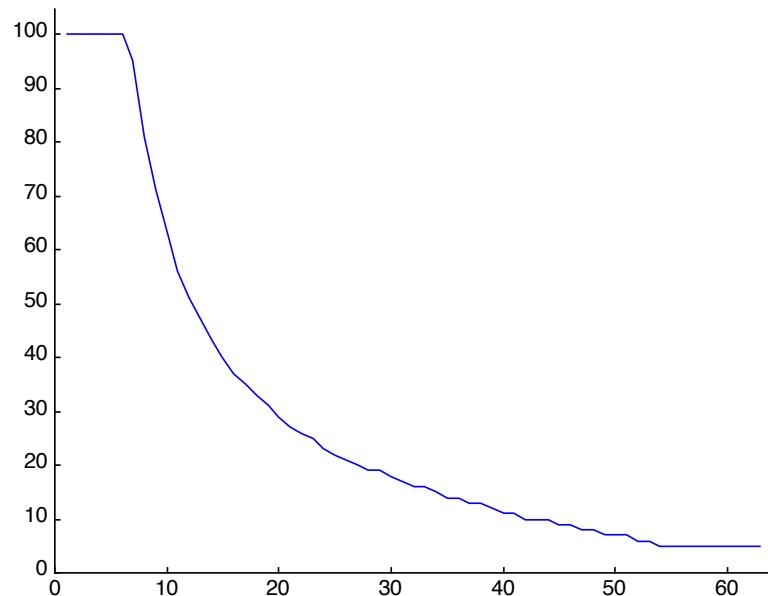
$$\lambda_k(R_G) = \cos^2\left(\frac{k\pi}{N}\right)$$

$$w_{k,j} = (\lambda_k)^{j/2} \sin\left(\frac{jk\pi}{N}\right)$$



Convergence of Gauss-Seidel on $Au=0$

- Eigenvectors of R_G are not the same as those of A .
Gauss-Seidel mixes the modes of A .



Jacobi method on $Au=0$ with $N=64$. Number of iterations required to reduce to $\|e\|_\infty < .01$

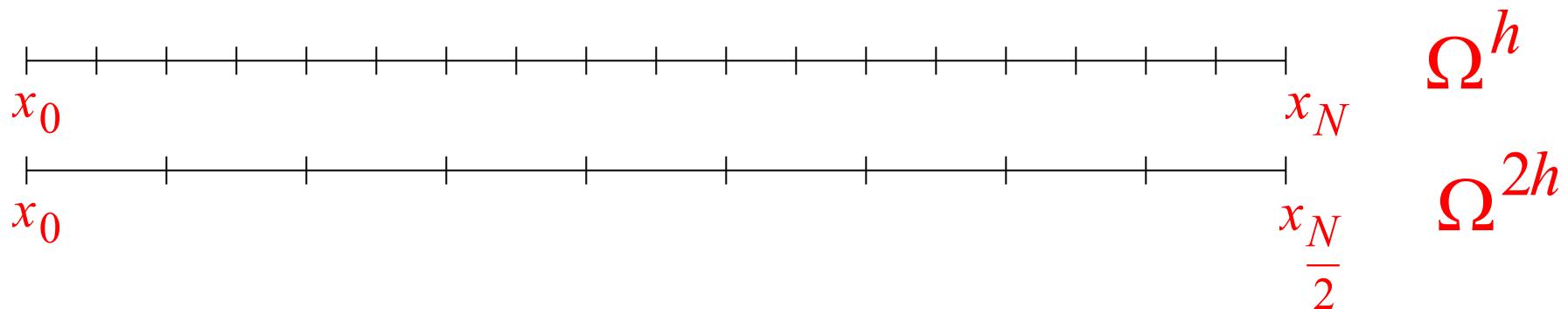
Initial guess : (Modes of A)

$$v_{kj} = \sin\left(\frac{jk\pi}{N}\right)$$

Developing a Multigrid Algorithm

First observation toward multigrid

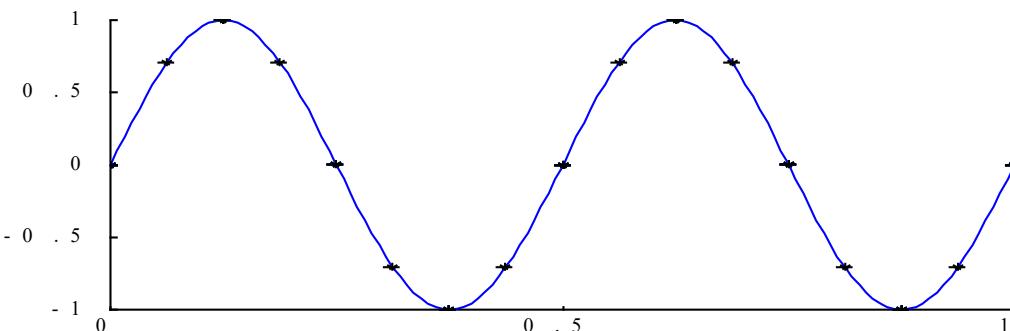
- Many relaxation schemes have the **smoothing property**, where oscillatory modes of the error are eliminated effectively, but smooth modes are damped very slowly.
- This might seem like a limitation, but by using coarse grids we can use the smoothing property to good advantage.



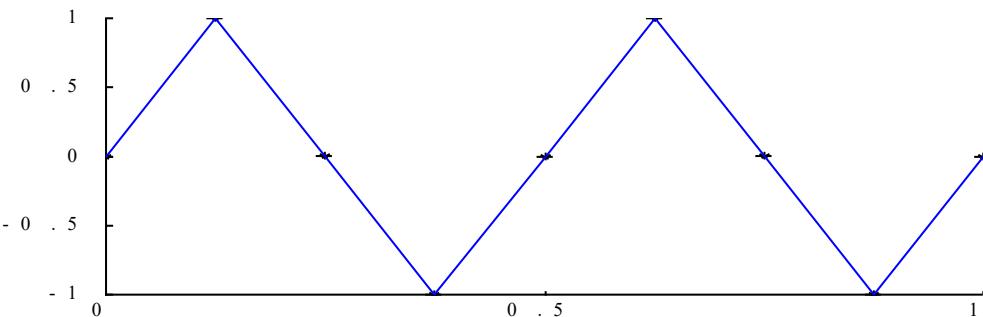
- Why use coarse grids??

Motivation for using a coarse grid: smooth error is (relatively) more oscillatory there!

- A smooth function:



- Can be represented by linear interpolation from a coarser grid:



On the coarse grid, the smooth error appears to be relatively higher in frequency: in the example it is the 4-mode, out of a possible 16, on the fine grid, $1/4$ the way up the spectrum. On the coarse grid, it is the 4-mode out of a possible 8, hence it is $1/2$ the way up the spectrum.

Relaxation will be more effective on this mode if done on the coarser grid!!

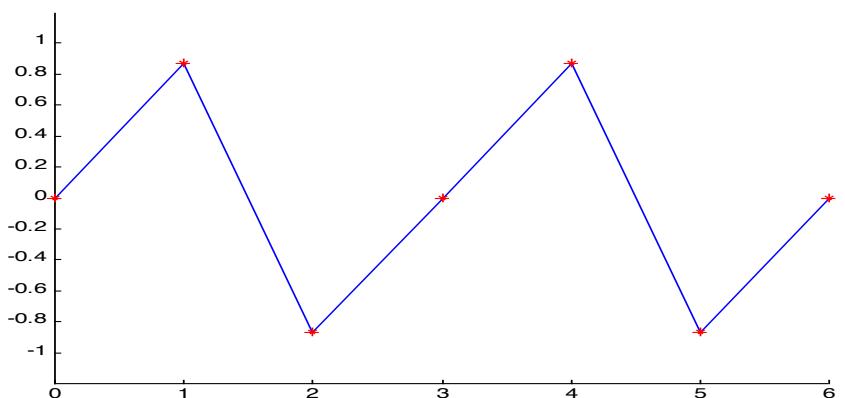
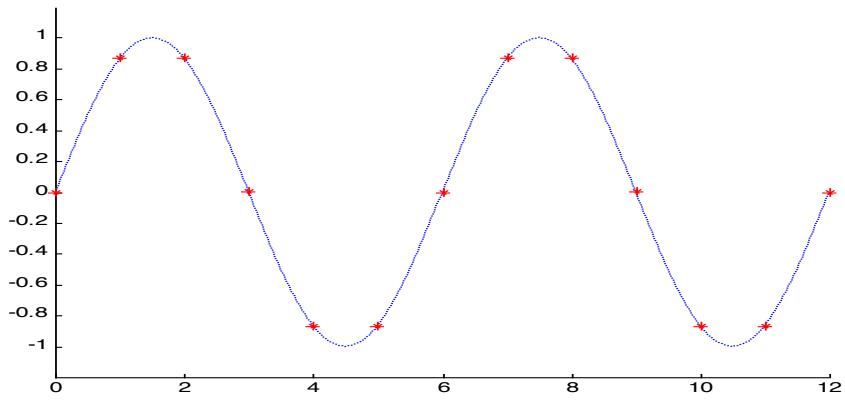
For $k=1, 2, \dots, N/2$, the k^{th} mode is preserved on the coarse grid

$$w_{k,2j}^h = \sin\left(\frac{2jk\pi}{N}\right) = \sin\left(\frac{jk\pi}{N/2}\right) = w_{k,j}^{2h}$$

Also, note that

$$w_{N/2}^h \rightarrow 0$$

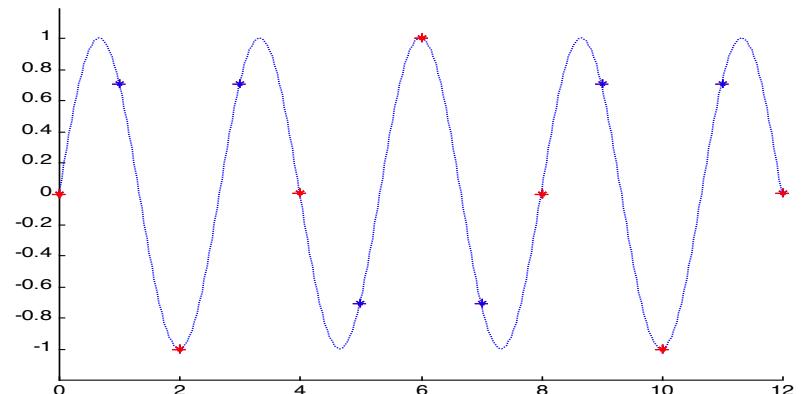
on the coarse grid



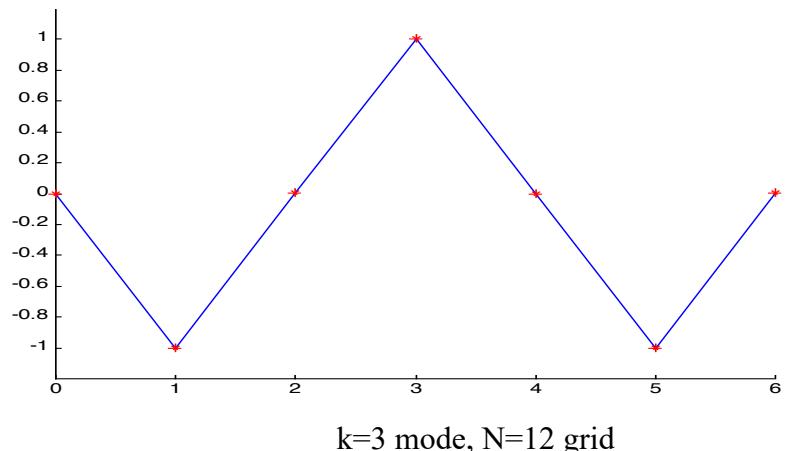
For $k > N/2$, w_k^h is invisible on the coarse grid: aliasing!!

- For $k > N/2$, the k^{th} mode on the fine grid is aliased and appears as the $(N-k)^{\text{th}}$ mode on the coarse grid:

$$\begin{aligned}
 (w_k^h)_{2j} &= \sin\left(\frac{(2j)\pi k}{N}\right) \\
 &= -\sin\left(\frac{2\pi j(N-k)}{N}\right) \\
 &= -\sin\left(\frac{\pi(N-k)j}{N/2}\right) \\
 &= -(w_{N-k}^{2h})
 \end{aligned}$$



$k=9$ mode, $N=12$ grid



$k=3$ mode, $N=12$ grid

Second observation toward multigrid:

- Recall the residual correction idea: Let \mathbf{v} be an approximation to the solution of $\mathbf{A}\mathbf{u}=\mathbf{f}$, where the residual $\mathbf{r}=\mathbf{f}-\mathbf{A}\mathbf{v}$. Then the error $\mathbf{e}=\mathbf{u}-\mathbf{v}$ satisfies $\mathbf{A}\mathbf{e}=\mathbf{r}$.
- After relaxing on $\mathbf{A}\mathbf{u}=\mathbf{f}$ on the fine grid, the error will be smooth. On the coarse grid, however, this error appears more oscillatory, and relaxation will be more effective.
- Therefore we go to a coarse grid and relax on the residual equation $\mathbf{A}\mathbf{e}=\mathbf{r}$, with an initial guess of $\mathbf{e}=0$.



Idea!

Coarse-grid correction

- Relax on $Au=f$ on Ω^h to obtain an approximation v^h
- Compute $r = f - Av^h$.
- Relax on $Ae=r$ on Ω^{2h} to obtain an approximation to the error, e^{2h} .
- Correct the approximation $v^h \leftarrow v^h + e^{2h}$.
- Clearly, we need methods for the mappings
 $\Omega^h \rightarrow \Omega^{2h}$ and $\Omega^{2h} \rightarrow \Omega^h$

1D Interpolation (Prolongation)

- Mapping from the coarse grid to the fine grid:

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

- Let v^h, v^{2h} be defined on Ω^h, Ω^{2h} . Then

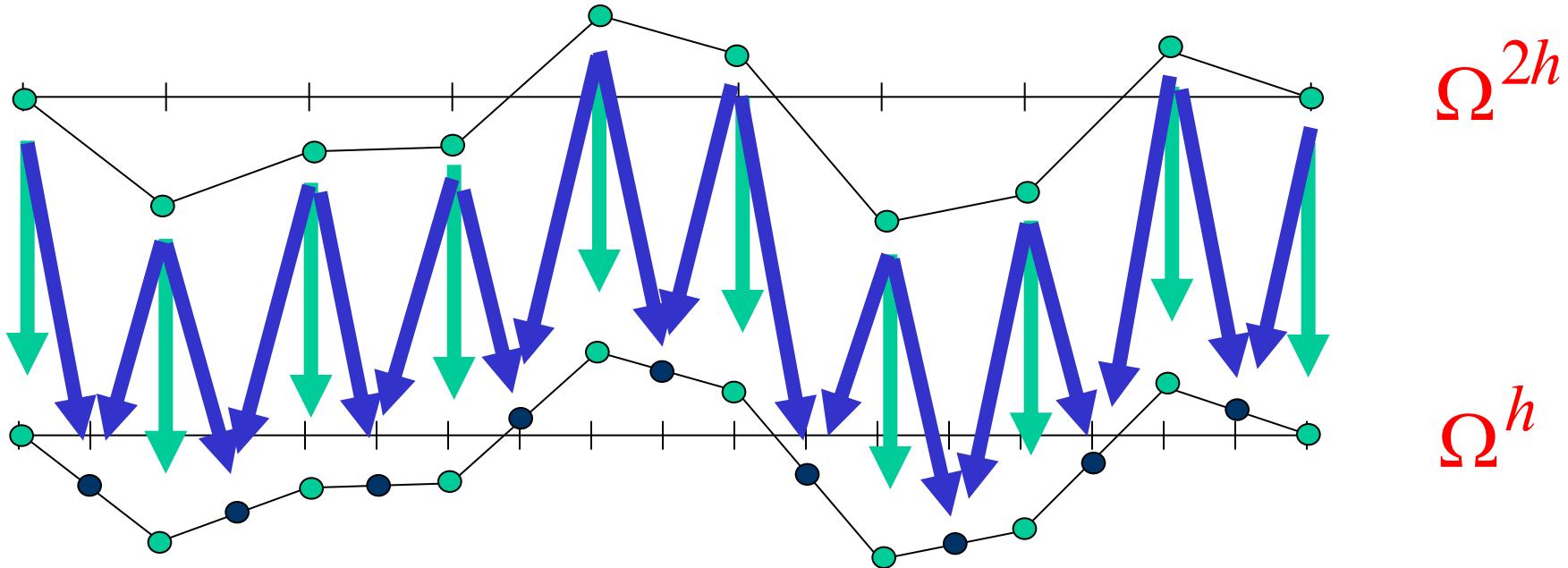
$$I_{2h}^h v^{2h} = v^h$$

where

$$\left. \begin{array}{l} v_{2i}^h = v_i^{2h} \\ v_{2i+1}^h = \frac{1}{2}(v_i^{2h} + v_{i+1}^{2h}) \end{array} \right\} \text{for } 0 \leq i \leq \frac{N}{2} - 1$$

1D Interpolation (Prolongation)

- Values at points on the coarse grid map unchanged to the fine grid
- Values at fine-grid points NOT on the coarse grid are the averages of their coarse-grid neighbors



The prolongation operator (1D)

- We may regard I_{2h}^h as a linear operator from $\mathcal{R}^{N/2-1} \rightarrow \mathcal{R}^{N-1}$

- e.g., for $N=8$,

$$\begin{pmatrix} 1/2 & & \\ 1 & & \\ 1/2 & 1/2 & \\ & 1 & \\ & 1/2 & 1/2 & \\ & & 1 & \\ & & & 1/2 \end{pmatrix}_{7 \times 3} \begin{pmatrix} v_1^{2h} \\ v_2^{2h} \\ v_3^{2h} \end{pmatrix}_{3 \times 1} = \begin{pmatrix} v_1^h \\ v_2^h \\ v_3^h \\ v_4^h \\ v_5^h \\ v_6^h \\ v_7^h \end{pmatrix}_{7 \times 1}$$

- I_{2h}^h has full rank, and thus null space $\{\phi\}$

1D Restriction by injection

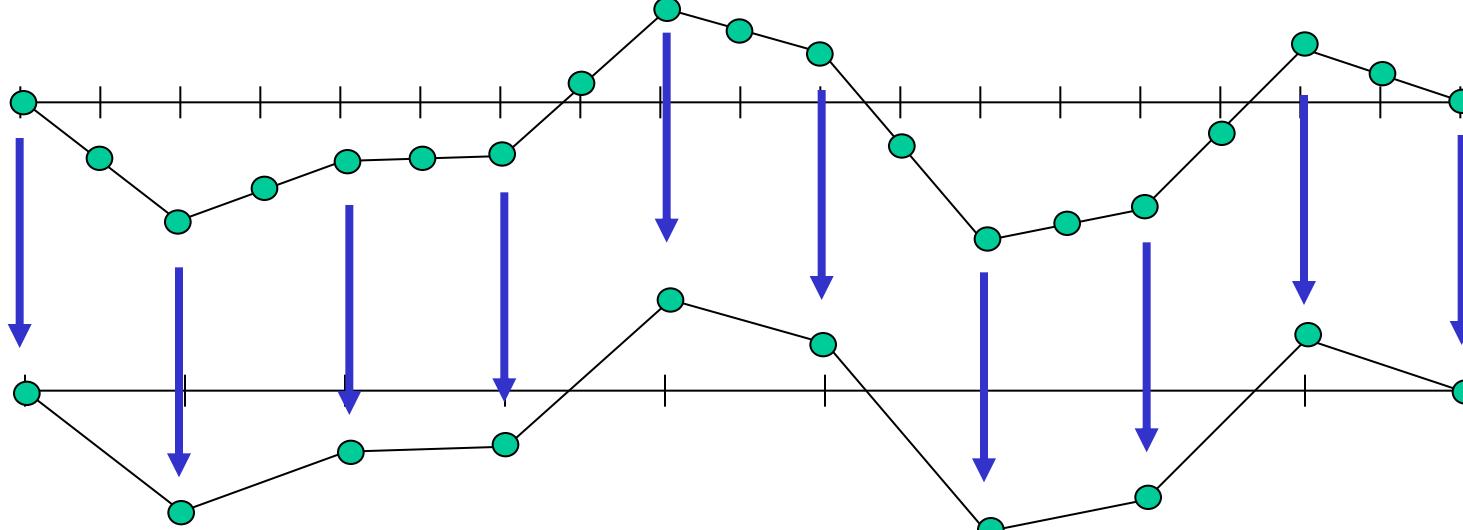
- Mapping from the fine grid to the coarse grid:

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

- Let v^h, v^{2h} be defined on Ω^h, Ω^{2h} . Then

$$I_h^{2h} v^h = v^{2h}$$

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



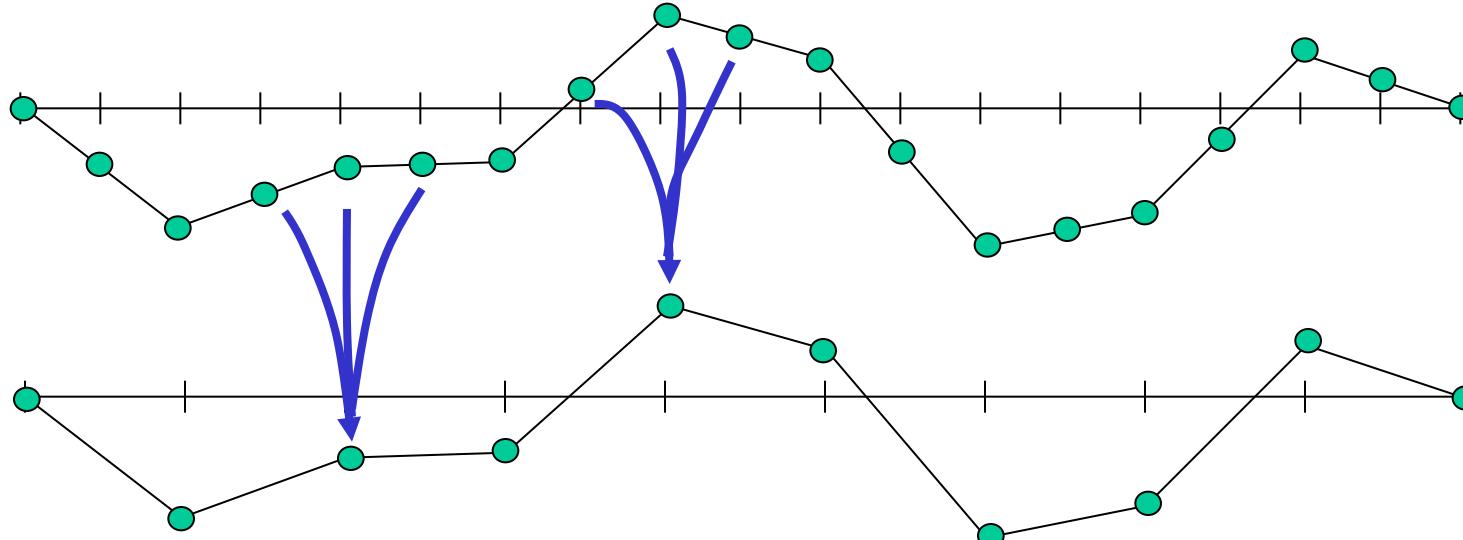
1D Restriction by full-weighting

- Let v^h, v^{2h} be defined on Ω^h, Ω^{2h} . Then

$$I_h^{2h} v^h = v^{2h}$$

where

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



The restriction operator \mathbf{R} (1D)

- We may regard I_h^{2h} as a linear operator from $\mathcal{R}^{N-1} \rightarrow \mathcal{R}^{N/2-1}$
- e.g., for $N=8$,

$$\begin{pmatrix} 1/4 & 1/2 & 1/4 & & & & \\ & 1/4 & 1/2 & 1/4 & & & \\ & & 1/4 & 1/2 & 1/4 & & \\ & & & 1/4 & 1/2 & 1/4 & \end{pmatrix} \begin{pmatrix} v_1^h \\ v_2^h \\ v_3^h \\ v_4^h \\ v_5^h \\ v_6^h \\ v_7^h \end{pmatrix} = \begin{pmatrix} v_1^{2h} \\ v_2^{2h} \\ v_3^{2h} \end{pmatrix}$$

- I_h^{2h} has rank $\sim \frac{N}{2}$, and thus $\dim(\text{NS}(\mathbf{R})) \sim \frac{N}{2}$

Prolongation and restriction are often nicely related

- For the 1D examples, linear interpolation and full-weighting are related by:

$$I_{2h}^h = \frac{1}{2} \begin{pmatrix} 1 \\ 2 \\ 1 & 1 \\ 2 \\ 1 & 1 \\ 2 \\ 1 \end{pmatrix} \quad I_h^{2h} = \frac{1}{4} \begin{pmatrix} 1 & 2 & 1 \\ & 1 & 2 & 1 \\ & & 1 & 2 & 1 \end{pmatrix}$$

- A commonly used, and highly useful, requirement is that

$$I_{2h}^h = c (I_h^{2h})^T \quad \text{for } c \in \mathbb{R}$$

Now, let's put all these ideas together

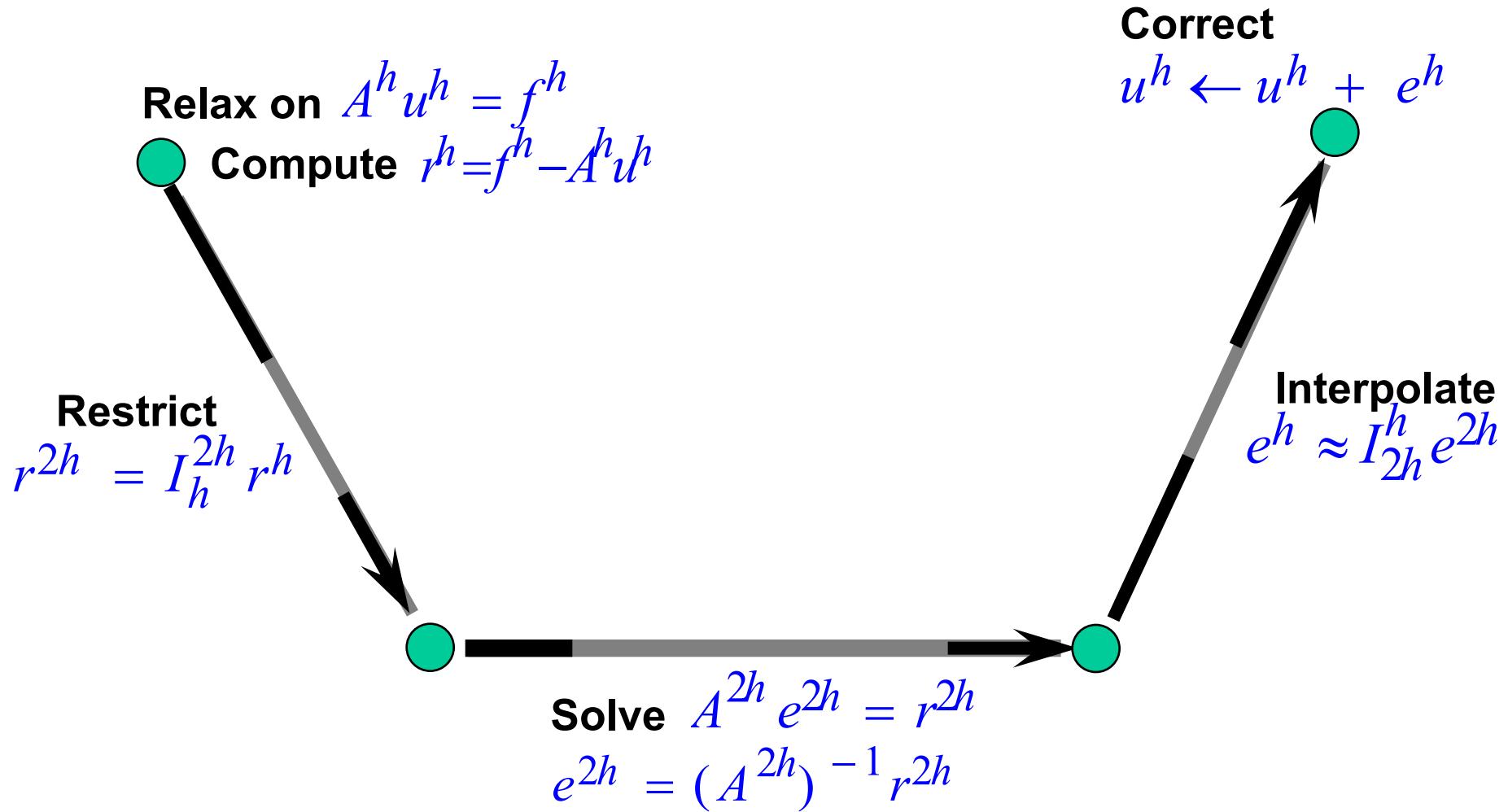
- Relaxation
 - effective on oscillatory error modes
- Residual equation
 - characterizes the error
- Prolongation and Restriction
 - provide pathways between coarse and fine grids

Two-Grid Correction Scheme (CS)

$$v^h \leftarrow CG(v^h, f^h, \alpha_1, \alpha_2)$$

- 1) Relax α_1 times on $A^h u^h = f^h$ on Ω^h with arbitrary initial guess v^h .
- 2) Compute $r^h = f^h - A^h v^h$.
- 3) Compute $r^{2h} = I_h^{2h} r^h$.
- 4) Solve $A^{2h} e^{2h} = r^{2h}$ on Ω^{2h} .
- 5) Correct fine-grid solution $v^h \leftarrow v^h + I_{2h}^h e^{2h}$.
- 6) Relax α_2 times on $A^h u^h = f^h$ on Ω^h with initial guess v^h .

Two-grid Correction Scheme (CS)



Convergence analysis, a little more precisely...

- Continuous problem: $Au = f$, $u_i = u(x_i)$
- Discrete problem: $A^h u^h = f^h$, $v^h \approx u^h$
- Global error: $E_i = u(x_i) - u_i^h$
 $\|E\| \leq Kh^p$ (p=2 for model problem)
- Algebraic error: $e_i = u_i^h - v_i^h$
- Suppose a tolerance ε is specified such that v^h must satisfy $\|u - v^h\| \leq \varepsilon$
- Then this is guaranteed if
$$\|u - u^h\| + \|u^h - v^h\| = \|E\| + \|e\| \leq \varepsilon$$

We can satisfy the requirement by imposing two conditions

1) $\|E\| \leq \frac{\varepsilon}{2}$. We use this requirement to determine a grid spacing h^* from

$$h^* \leq \left(\frac{\varepsilon}{2K} \right)^{1/p}$$

2) $\|e\| \leq \frac{\varepsilon}{2}$, which determines the number of iterations required.

- If we iterate until $\|e\| \leq \frac{\varepsilon}{2} = K(h^*)^p$ on Ω^{h^*} then we have converged to the level of discretization error.

Converging to the level of discretization error

- Use a MV scheme with convergence rate $\gamma < 1$ independent of h (fixed α_1 and α_2).
- Assume a d-dimensional problem on an $N \times N \times \dots \times N$ grid with $h = N^{-\frac{1}{d}}$.
- The V-cycle must reduce the error from $\|e\| \sim O(1)$ to $\|e\| \sim O(h^p) \sim O(N^{-p})$
- We can determine θ , the number of V-cycles required to accomplish this.

Work needed to converge to the level of discretization error

- Since θ V-cycles at convergence rate γ are required, we see that

$$\gamma^\theta \sim O(N^{-p})$$

implying that

$$\theta \sim O(\log N)$$

- Since one V-cycle costs $O(1)$ WU and one WU is $O(N^d)$, we see that the cost of converging to the level of discretization error using the MV method is

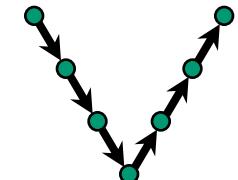
$$O(N^d \log N)$$

which is comparable to fast direct methods (FFT based).

Comments on multigrid cycles

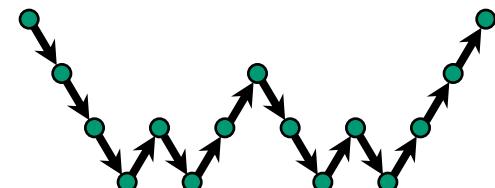
- **V-cycle:**

- Most commonly used cycle
- $O(N)$ work satisfies $\|e\| \leq \varepsilon$ for fixed tolerance ε



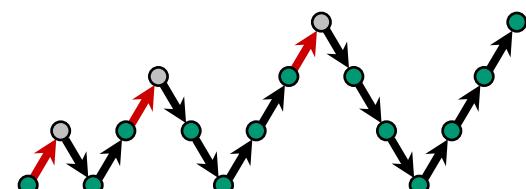
- **W-cycle:**

- More robust than V-cycles
- $O(N)$ work satisfies $\|e\| \leq \varepsilon$
- Not scalable in parallel (discussed later)



- **FMG V-cycle:**

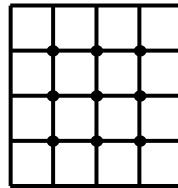
- $O(N)$ work satisfies $\|e\| \leq kh^p$ where h^p is discretization accuracy



2D model problem: Laplace on a square

- Five-point stencil discretization on a uniform grid

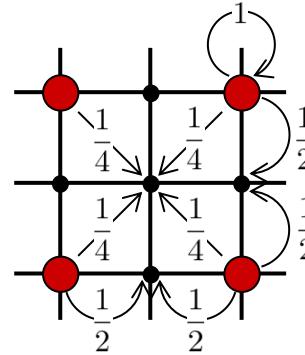
$$-\nabla^2 u = f$$



$$A = \begin{bmatrix} & -1 & \\ -1 & 4 & -1 \\ & -1 & \end{bmatrix}$$

- Smoothers: weighted Jacobi or GS (lexicographical or red/black)
- Full coarsening, bilinear interpolation

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



- Coarse discretization (scaled appropriately) for A_c

2D model problem: Laplace on a square (continued)

- Local Fourier Analysis (LFA) can be used to predict multigrid convergence factors

ν	ρ - LFA (2-grid)	
	Jac ($\omega = 4/5$)	R/B - GS
1	0.600	0.250
2	0.360	0.074
3	0.216	0.053
4	0.137	0.041

N	ρ - V(1,1)	
	Jac ($\omega = 4/5$)	R/B - GS
15×15	0.362	0.105
63×63	0.390	0.134
255×255	0.400	0.140
1023×1023	0.404	0.141

- V-cycle rates are usually worse than 2-grid rates, but 2-grid analysis is still extremely useful
- W-cycle rates are usually very similar to 2-grid rates
- V-cycle costs: relaxation is the dominant cost per cycle, roughly equivalent to $(1+1/4+1/16+\dots)\nu = (4/3)\nu$ fine-grid relaxations

Anisotropic Problems

Anisotropic Problems

- All problems considered thus far have had $-\frac{1}{h^2}$ as the off-diagonal entries.
- We consider two situations when the matrix has two different constant on the off-diagonals.
These situations arise when
 - the (2-d) differential equation has constant, but different, coefficients for the derivatives in the coordinate directions
 - the discretization has constant, but different, mesh spacing in the different coordinate directions

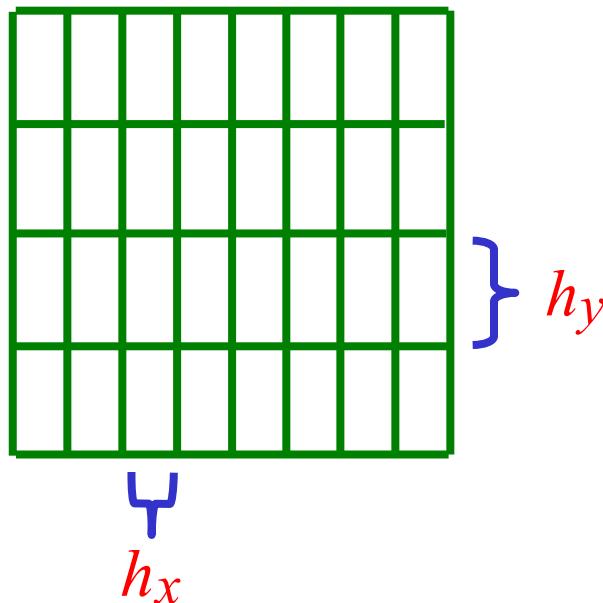
We consider two types of anisotropy

- Different coefficients on the derivatives

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

discretized on a uniform grid in each direction.

- Constant, but different, mesh spacings:



$$h_x = h = \frac{1}{N}$$

$$h_y = \frac{h_x}{\sqrt{\alpha}}$$

Both problems lead to the same stencil

$$\frac{-u_{j-1,k} + 2u_{j,k} - u_{j+1,k}}{h^2} + \alpha \frac{-u_{j,k-1} + 2u_{j,k} - u_{j,k+1}}{h^2}$$


$$A^h = \frac{1}{h^2} \begin{pmatrix} -\alpha & & \\ -1 & 2+2\alpha & -1 \\ & -\alpha & \end{pmatrix}$$


$$\frac{-u_{j-1,k} + 2u_{j,k} - u_{j+1,k}}{h^2} + \frac{-u_{j,k-1} + 2u_{j,k} - u_{j,k+1}}{\left(\frac{h}{\sqrt{\alpha}}\right)^2}$$

Why standard multigrid fails

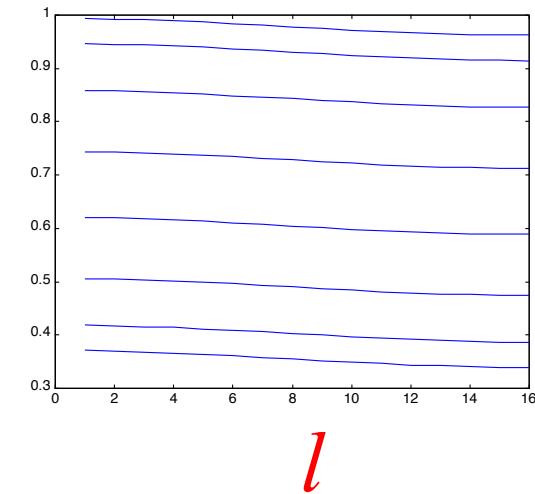
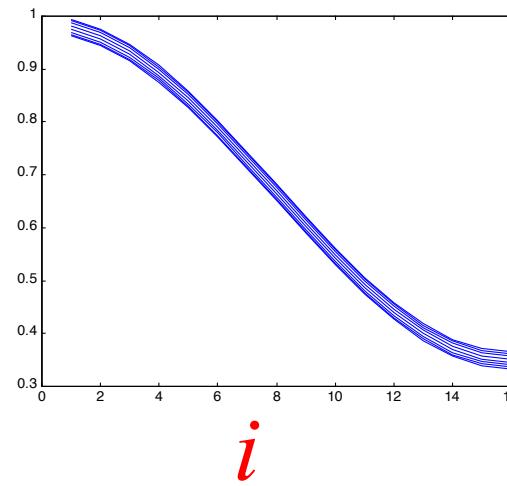
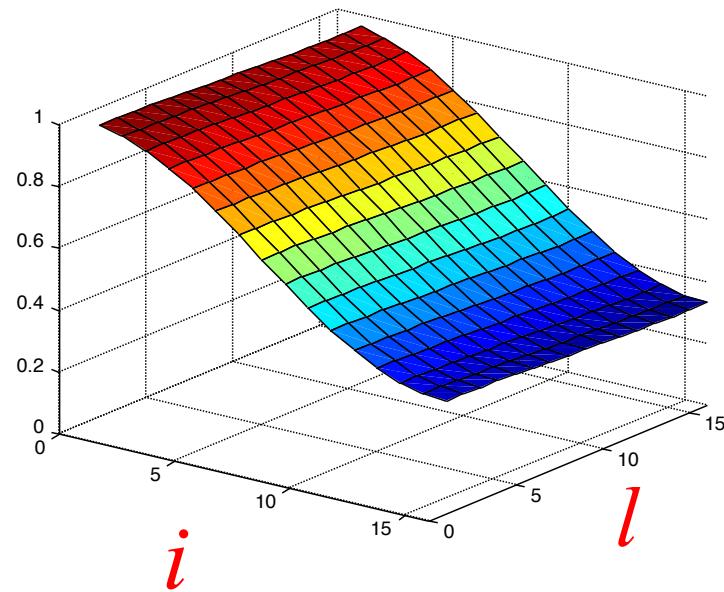
- Note that $A^h = \frac{1}{h^2} \begin{pmatrix} -\alpha & & \\ -1 & 2+2\alpha & -1 \\ & -\alpha & \end{pmatrix}$ has weak connections in the y -direction. MG convergence factors degrade as α gets small. Poor performance at $\alpha = 0.1$.

- Consider $\alpha \Rightarrow 0$. $A^h \Rightarrow \frac{1}{h^2} \begin{pmatrix} 0 & & \\ -1 & 2+2\alpha & -1 \\ 0 & & \end{pmatrix}$
- This is a collection of disconnected 1-d problems!
- Point relaxation will smooth oscillatory errors in the x -direction (strong connections), but with no connections in the y -direction the errors in that direction will generally be random, and no point relaxation will have the smoothing property in the y -direction.

We analyze weighted Jacobi

- The eigenvalues of the weighted Jacobi iteration matrix for this problem are

$$\lambda_{i,l} = 1 - \frac{2\omega}{1+\alpha} \left(\sin^2\left(\frac{i\pi}{2N}\right) + \alpha \sin^2\left(\frac{l\pi}{2N}\right) \right)$$

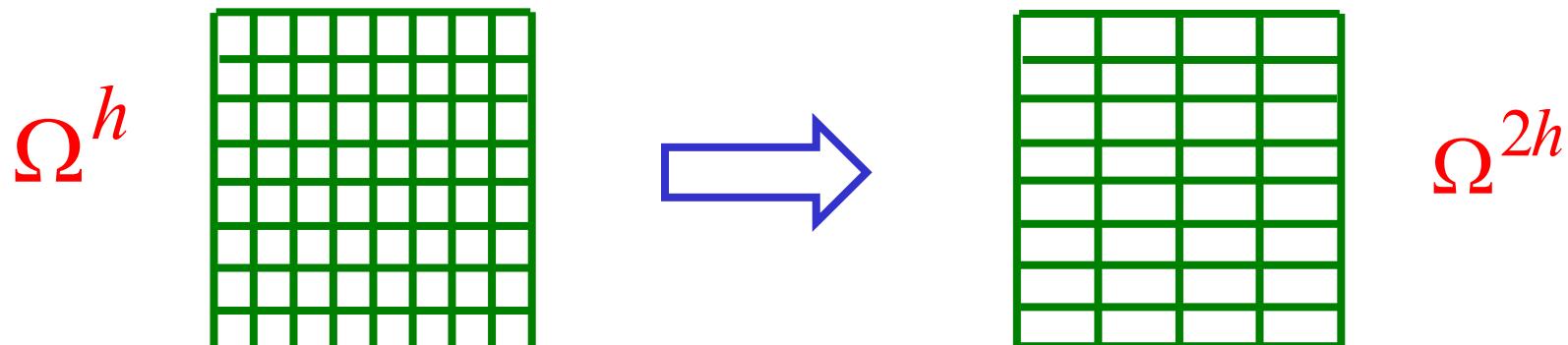


Two strategies for anisotropy

- **Semicoarsening** Because we expect MG-like convergence for the 1-d problems along lines of constant y , we should coarsen the grid in the x -direction, but not in the y -direction.
- **Line relaxation** Because the the equations are strongly coupled in the x -direction it may be advantageous to solve simultaneously for entire lines of unknowns in the x -direction (along lines of constant y)

Semicoarsening with point relaxation

- Point relaxation on $A^h = \frac{1}{h^2} \begin{pmatrix} -\alpha & & \\ -1 & 2+2\alpha & -1 \\ & -\alpha & \end{pmatrix}$ smooths in the x -direction. Coarsen by removing every other x -line.



- We do not coarsen along the y -lines.
- Semicoarsening is not as "fast" as full coarsening. The number of points on Ω^{2h} is about half the number of points on Ω^h , instead of the usual one-fourth.

Interpolation with semicoarsening

- We interpolate in the 1-dimensional way along each line of constant y .
- The coarse-grid correction equations are

$$v_{2j,k}^h = v_{2j,k}^h + v_{j,k}^{2h}$$

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

Line relaxation with full coarsening

- The other approach to this problem is to do full coarsening, but to relax entire lines (constant y) of variables simultaneously.
- Write A^h in block form as

$$A^h = \begin{pmatrix} D & -cI & & \\ -cI & D & -cI & \\ & -cI & D & -cI \\ & & \ddots & \ddots & -cI \\ & & & -cI & D \end{pmatrix}$$

where

$$c = \frac{\alpha}{h^2} \quad \text{and}$$

$$D = \frac{1}{h^2} \begin{pmatrix} 2+2\alpha & -1 & & \\ -1 & 2+2\alpha & -1 & \\ & & \ddots & \\ & & -1 & 2+2\alpha \end{pmatrix}$$

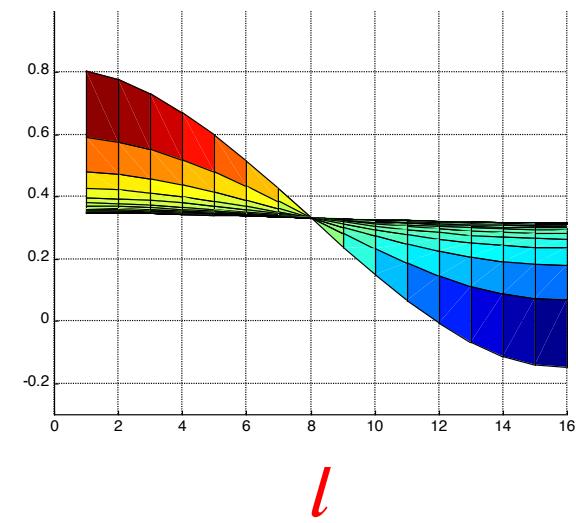
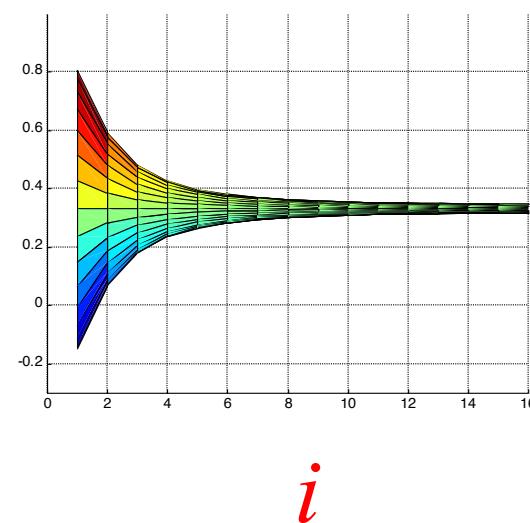
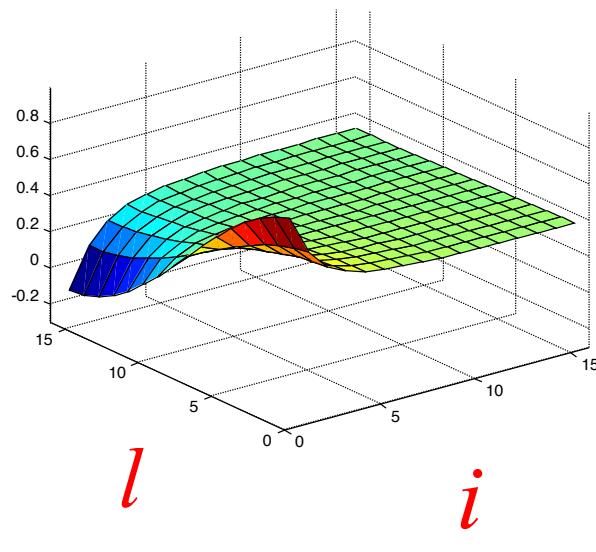
Line relaxation

- One sweep of line relaxation consists of solving a tridiagonal system for each line of constant y .
- The k th such system has the form $Dv_k^h = g_k^h$ where v_k^h is the k th subvector of v^h with entries $(v_k^h)_j = v_{j,k}^h$ and the k th right-hand side subvector is
$$(g_k^h)_j = f_{j,k}^h + \frac{\alpha}{h^2} (v_{j,k-1}^h + v_{j,k+1}^h)$$
- Because D is tridiagonal, the k th system can be solved very efficiently.

Why line relaxation works

- The eigenvalues of the weighted block Jacobi iteration matrix are

$$\lambda_{i,l} = 1 - \frac{2\omega}{2\sin^2\left(\frac{i\pi}{2N}\right) + \alpha} \left(\sin^2\left(\frac{i\pi}{2N}\right) + \alpha \sin^2\left(\frac{l\pi}{2N}\right) \right)$$



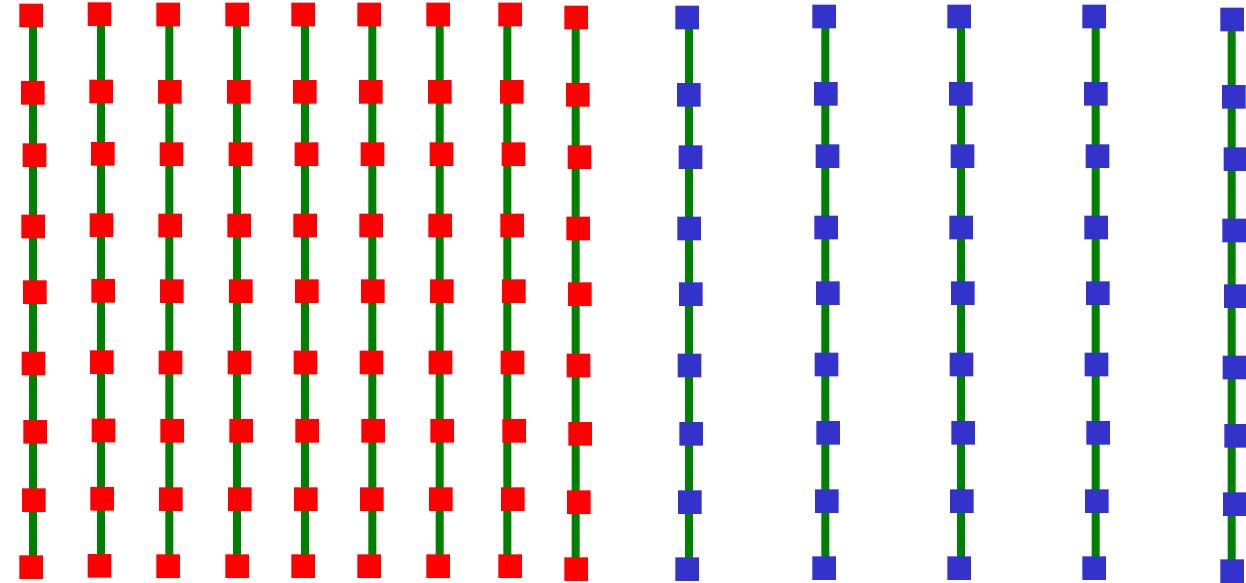
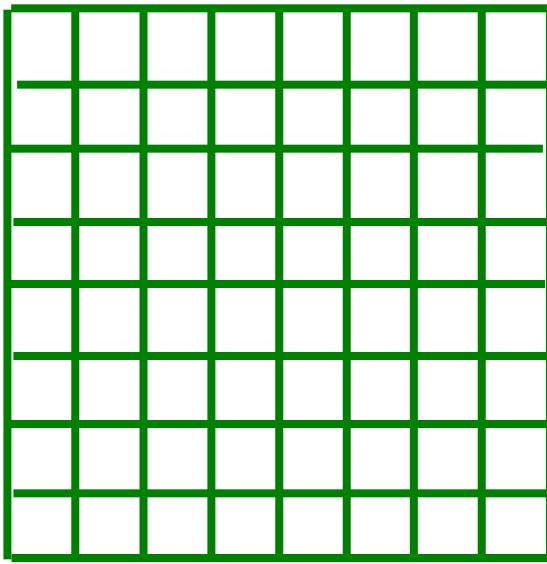
Semicoarsening with line relaxation

- We might not know the direction of weak coupling or it might vary.
- Suppose we want a method that can handle either

$$A_1^h = \frac{1}{h^2} \begin{pmatrix} -\alpha & & \\ -1 & 2+2\alpha & -1 \\ & -\alpha & \end{pmatrix} \quad \text{or} \quad A_2^h = \frac{1}{h^2} \begin{pmatrix} -1 & & \\ -\alpha & 2+2\alpha & -\alpha \\ & -1 & \end{pmatrix}$$

- We could use semicoarsening in the x-direction to handle A_1^h and line relaxation in the y-direction to take care of A_2^h .

Semicoarsening with line relaxation



- The original grid
- Original grid viewed as a stack of "pencils." Line relaxation is used to solve problem along each pencil.
- Coarsening is done by deleting every other pencil

An anisotropic example

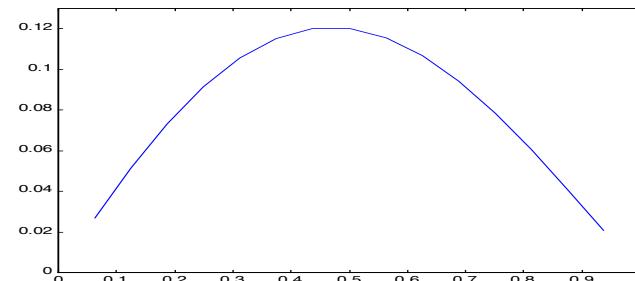
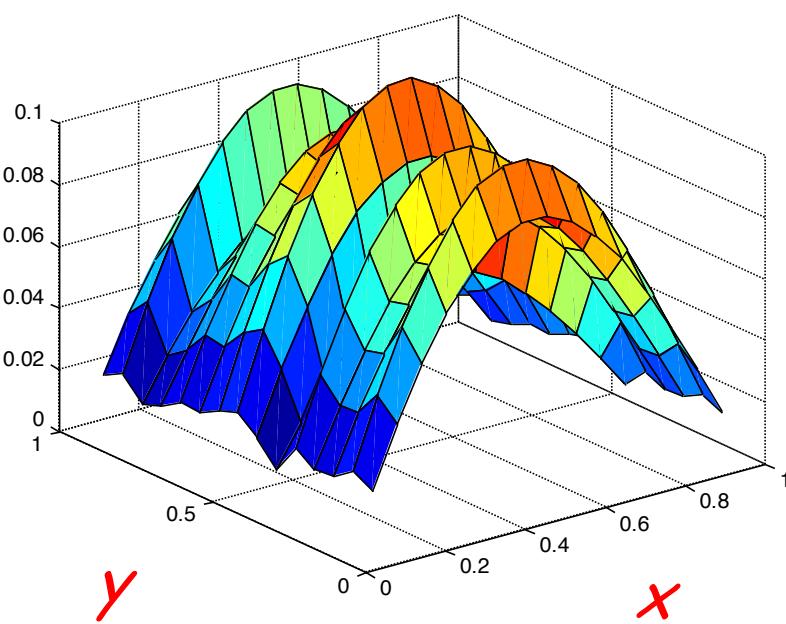
- Consider $-u_{xx} - \alpha u_{yy} = f$ with $u=0$ on the boundaries of the unit square, and stencil given by

$$A^h = \frac{1}{h^2} \begin{pmatrix} & -\alpha \\ -1 & 2+2\alpha & -1 \\ & -\alpha \end{pmatrix}$$

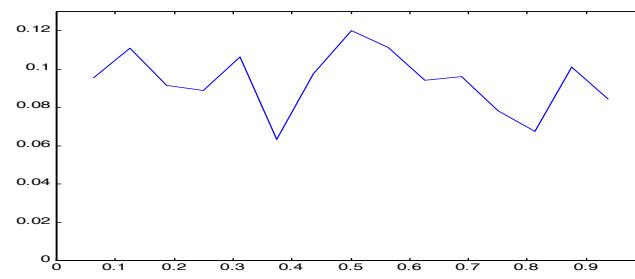
- Suppose $f(x,y) = 2(y-y^2) + 2\alpha(x-x^2)$ so the exact solution is given by $u(x,y) = (y-y^2)(x-x^2)$
- Observe that if α is small, the x -direction dominates while if α is large, the y -direction dominates

What is smooth error?

- Consider $\alpha=0.001$ and suppose point Gauss-Seidel is applied to a random initial guess. The error after 50 sweeps appears as:



Error along line of constant x



Error along line of constant y

We experiment with 3 methods

- Standard V(2,1)-cycling, with point Gauss-Seidel relaxation, full coarsening, and linear interpolation
- Semicoarsening in the x -direction. Coarse and fine grids have the same number of points in the y -direction. 1-d full weighting and linear interpolation are used in the x -direction, there is no y -coupling in the intergrid transfers
- Semicoarsening in the x -direction combined with line relaxation in the y -direction. 1-d full weighting and interpolation.

With semicoarsening, the operator must change

- To account for unequal mesh spacing, the residual and relaxation operators must use a modified stencil

$$A = \begin{pmatrix} -\frac{\alpha}{h_y^2} \\ -\frac{1}{h_x^2} \left(\frac{2}{h_x^2} + \frac{2\alpha}{h_y^2} \right) - \frac{1}{h_x^2} \\ -\frac{1}{h_y^2} \end{pmatrix}$$

- Note that as grids become coarser, h_x grows while h_y remains constant.

How do the 3 methods work for various values of α ?

- Asymptotic convergence factors:

scheme	1000	100	10	1	0.1	0.01	0.001	1E-04
V(2,1)-cycles	0.95	0.94	0.58	0.13	0.58	0.90	0.95	0.95
semicoarsening in x	0.94	0.99	0.98	0.93	0.71	0.28	0.07	0.07
semiC / line relax	0.04	0.08	0.08	0.08	0.07	0.07	0.08	0.08

α



y-direction strong *x*-direction strong

- Note: semicoarsening in x works well for $\alpha < .001$ but degrades noticeably even at $\alpha = .1$

A semicoarsening subtlety

- Suppose α is small, so that semicoarsening in x is used. As we progress to coarser grids, h_x^{-2} gets small but h_y^{-2} remains constant.
- If, on some coarse grid, h_x^{-2} becomes comparable to αh_y^{-2} , the problem effectively becomes recoupled in the y -direction. Continued semicoarsening can produce artificial anisotropy, strong in the y -direction.
- When this occurs, it is best to stop semicoarsening and continue with full coarsening on any further coarse grids.

Variable and Discontinuous Coefficients

Variable coefficient problems

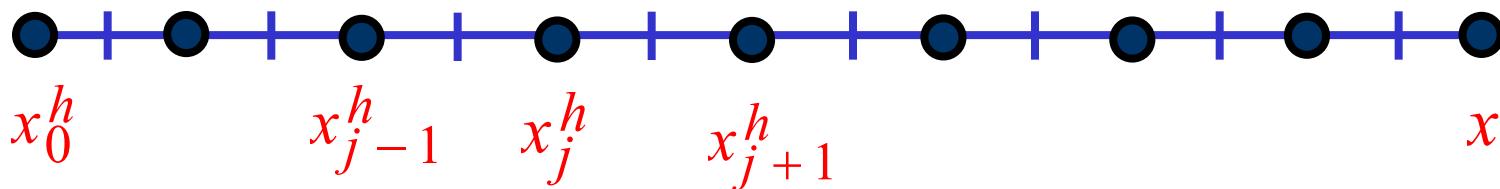
- A common difficulty is the variable coefficient problem, given in 1-d by

$$-(a(x)u'(x))' = f(x) \quad 0 < x < 1$$

$$u(0) = u(1) = 0$$

where $a(x)$ is a positive function on $[0,1]$

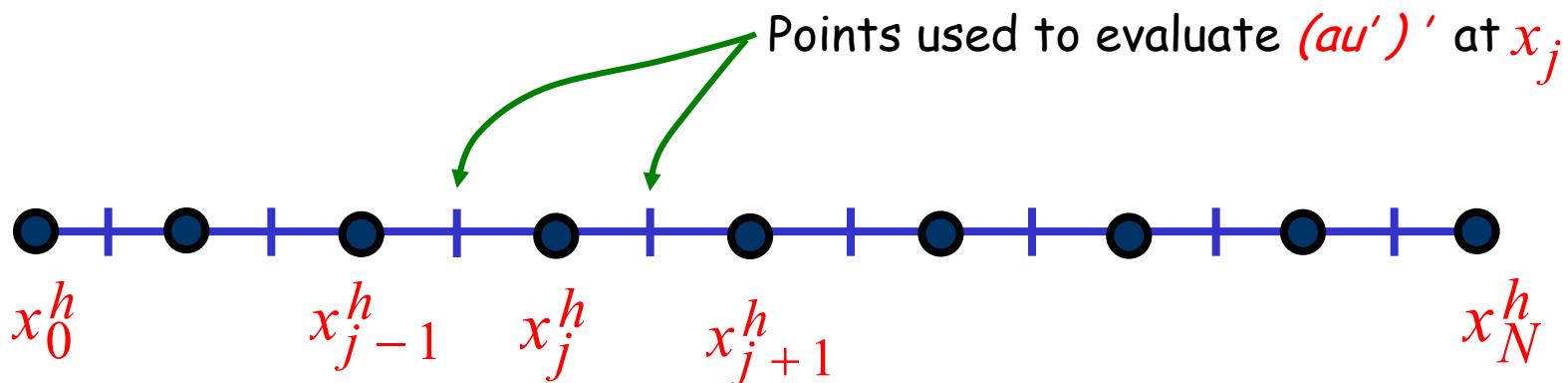
- We seek to develop a **conservative**, or **self-adjoint**, method for discretizing this problem.
- Assume we have available to us the values of $a(x)$ at the midpoints of the grid $a_{j+1/2} \equiv a(x_{j+1/2})$



We discretize using central differences

- We can use second-order differences to approximate the derivatives. To use a grid spacing of h we evaluate $a(x)u'(x)$ at points midway between the gridpoints:

$$(a(x)u'(x))' \Big|_{x_j} \approx \frac{(au')|_{x_{j+1/2}} - (au')|_{x_{j-1/2}}}{h} + O(h^2)$$



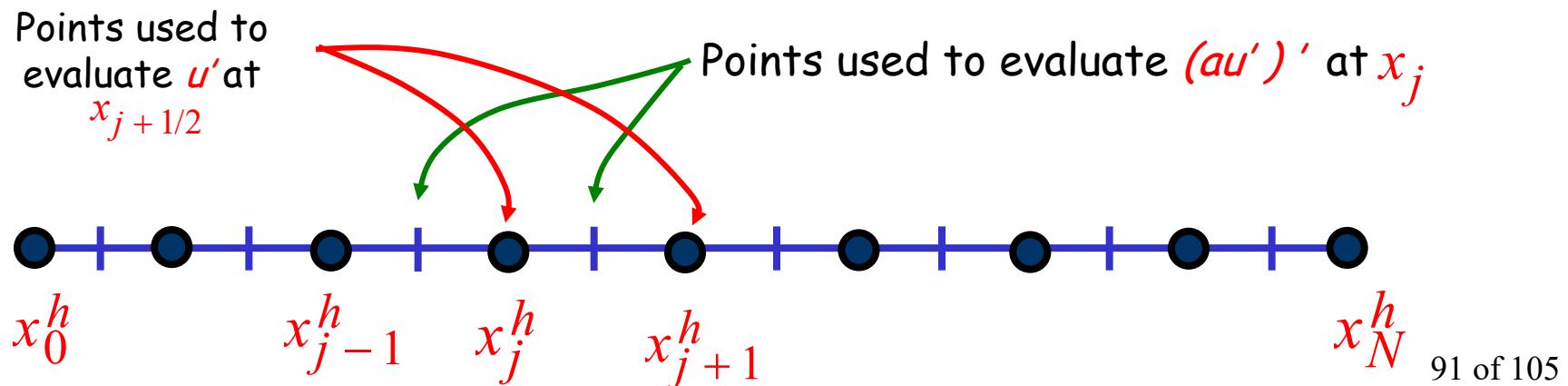
We discretize using central differences

- To evaluate $(au')|_{x_{j+1/2}}$ we must sample $a(x)$ at the point $x_{j+1/2}$ and use second order differences:

$$(au')|_{x_{j+1/2}} \approx a_{j+1/2} \frac{u_{j+1} - u_j}{h} \quad (au')|_{x_{j-1/2}} \approx a_{j-1/2} \frac{u_j - u_{j-1}}{h}$$

where

$$a_{j+1/2} \equiv a(x_{j+1/2})$$



The basic stencil is given

- We combine the differences for u' and for (au') ' to obtain the operator

$$-(a(x_j) u'(x_j))'(x_j) \approx \frac{-a_{j+1/2} \frac{u_{j+1} - u_{j-1}}{h} + a_{j-1/2} \frac{u_j - u_{j-1}}{h}}{h}$$

and the problem becomes, for $1 \leq j \leq N-1$

$$\frac{1}{h^2} (-a_{j-1/2} u_{j-1} + (a_{j-1/2} + a_{j+1/2}) u_j - a_{j+1/2} u_{j+1}) = f_j$$

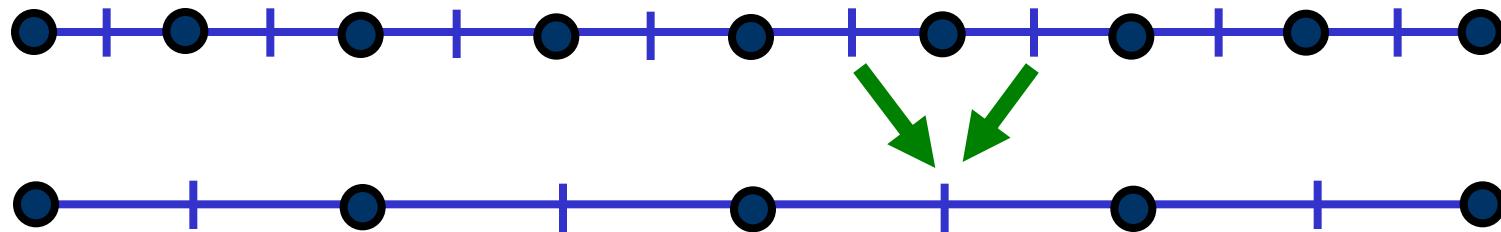
$$u_0 = u_N = 0$$

Coarsening the variable coefficient problem

- A reasonable approach is to use a standard multigrid algorithm with linear interpolation, full weighting, and the stencil

$$A^{2h} = \frac{1}{(2h)^2} \begin{pmatrix} -a_{j-1/2}^{2h} & a_{j-1/2}^{2h} + a_{j+1/2}^{2h} & -a_{j+1/2}^{2h} \end{pmatrix}$$

where $a_{j+1/2}^{2h} = \frac{a_{2j+1/2}^h + a_{2j+3/2}^h}{2}$



- The same stencil can be obtained via the Galerkin relation

A Variable coefficient example

- We use V(2,1) cycle, full weighting, linear interpolation.
- We use $a(x) = 1 + \rho \sin(k\pi x)$ and $a(x) = \rho \text{rand}(k\pi x)$

$$a(x) = 1 + \rho \sin(k\pi x)$$

ρ	k=3	k=25	k=50	k=100	k=200	k=400	
0	0.085	0.085	0.085	0.085	0.085	0.085	0.085
0.25	0.084	0.098	0.098	0.094	0.093	0.083	0.083
0.5	0.093	0.185	0.194	0.196	0.195	0.187	0.173
0.75	0.119	0.374	0.387	0.391	0.39	0.388	0.394
0.85	0.142	0.497	0.511	0.514	0.514	0.526	0.472
0.95	0.191	0.681	0.69	0.694	0.699	0.745	0.672

$$a(x) = \rho \text{rand}(k\pi x)$$



Robust Multigrid

Two key ingredients restore robustness on logically structured problems,

- Use the entries (weights) of the discrete operator to restore the accuracy of the interpolation.
- Use a minimization approach to define an optimal coarse-grid operator.

These techniques set the stage for truly Algebraic Multigrid Methods (AMG).

Optimal Coarse Grid Operator?

Minimize the Error in the Range of the Interpolation

Consider the equivalent variational formulation:

$$u^h = \min_{v \in \mathcal{H}} \left\{ \mathcal{L} [v] \equiv \frac{1}{2} v^T L_h v - v^T f_h \right\}$$

Let \tilde{u}^h be an approximation obtained after smoothing, then the error $e^h \equiv u^h - \tilde{u}^h$ is smooth and is well approximated on a coarser grid.

Substituting the interpolant, $u^h \approx \tilde{u}^h + I_{2h}^h e^{2h}$, into the functional

$$\mathcal{L} [u^h] = (\tilde{u}^h + I_{2h}^h e^{2h})^T L_h (\tilde{u}^h + I_{2h}^h e^{2h}) - (\tilde{u}^h + I_{2h}^h e^{2h})^T f_h$$

$$\frac{\partial \mathcal{L}}{\partial e^{2h}} = 0 \Rightarrow (I_{2h}^h)^T L_h I_{2h}^h e^{2h} = (I_{2h}^h)^T (b_h - L_h \tilde{u}^h)$$

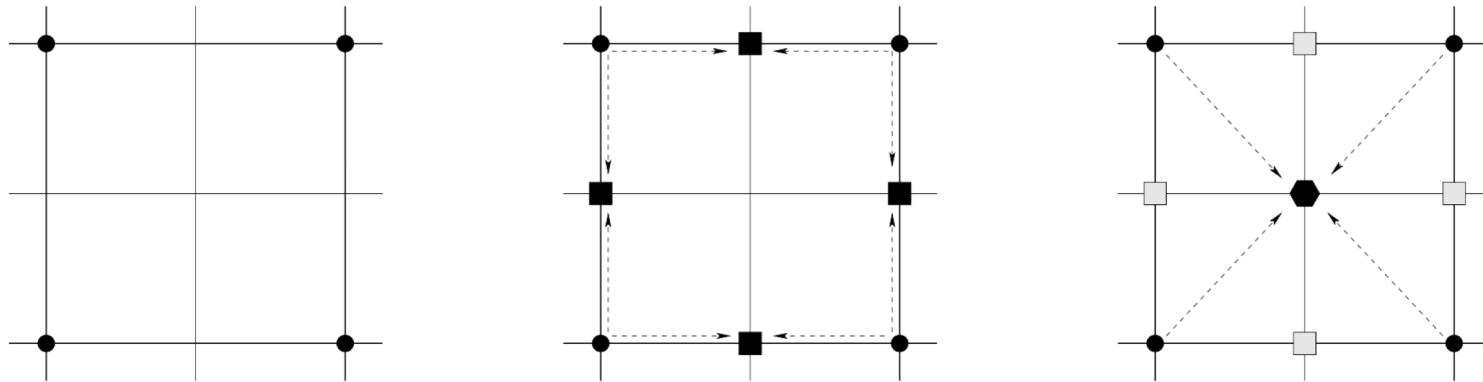
This implies that the coarse-grid operator is given by

$$L_{2h} = (I_{2h}^h)^T L_h I_{2h}^h \quad \text{and} \quad f_{2h} = (I_{2h}^h)^T (f_h - L_h \tilde{u}^h)$$

This Galerkin CG operator is critical to AMG as well.

Brandt, Multigrid Techniques, 1984

Operator Induced Interpolation



- Coarse points are simply injected
- A fine point lying on a coarse-line could use a 1D local solve: need to collapse the operator

$$\begin{bmatrix} -NW & -N & -NE \\ -W & O & -E \\ -SW & -S & -SE \end{bmatrix}_{2i+1,j} \rightarrow [-(W + SW + NW) \quad (O - S - N) \quad -(E + SE + NE)]_{2i+1,j}$$

- Fine-grid points use the fine-grid stencil

Dendy, Black Box Multigrid, 1982

Robust Variational Multigrid on Structured Grids

- Coarsening:
 - selected *a priori*, e.g., standard or semi-coarsening
- Smoothing:
 - selected to complement the coarsening/interpolation
 - standard coarsening: alternating line (2D), plane relaxation (3D)
- Interpolation: I_{k-1}^k
 - constructed from the discrete operator (matrix)
 - approximately preserves the continuity of the flux
- Variational Coarse Grid Operator: $L_{k-1} = (I_{k-1}^k)^* L_k I_{k-1}^k$
 - minimizes the error in the range of the interpolation
- Restriction: $J_k^{k-1} = (I_{k-1}^k)^*$
 - Dictated by variational principle, preserves symmetry of the discrete operator

Nonlinear Problems

Nonlinear Problems

- How should we approach the nonlinear system

$$A(u) = f$$

and can we use multigrid to solve such a system?

- A fundamental relation we've relied on, the residual equation

$$Au - Av = f - Av \Rightarrow Ae = r$$

does not hold, since, if $A(u)$ is a nonlinear operator,

$$A(u) - A(v) \neq A(e)$$

The Nonlinear Residual Equation

- We still base our development around the residual equation, now the **nonlinear** residual equation:

$$A(u) = f$$

$$A(u) - A(v) = f - A(v)$$

$$A(u) - A(v) = r$$

- How can we use this equation as the basis for a solution method?

Let's consider Newton's Method

- The best known and most important method for solving nonlinear equations!
- We wish to solve $F(x) = 0$.
- Expand F in a Taylor series about x :

$$F(x+s) = F(x) + sF'(x) + s^2 F''(\xi)$$

- Dropping higher order terms, if $x+s$ is a solution,
 $0=F(x)+sF'(x)$ thus $s=-F(x)/F'(x)$
- Hence, we develop an iteration

$$x \leftarrow x - \frac{F(x)}{F'(x)}$$

Newton's method for systems

- We wish to solve the system $A(u) = 0$. In vector form this is

$$A(u) = \begin{pmatrix} f_1(u_1, u_2, \dots, u_N) \\ f_2(u_1, u_2, \dots, u_N) \\ \vdots \\ f_N(u_1, u_2, \dots, u_N) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

- Expanding $A(v+e)$ in a Taylor series about v :

$$A(v+e) = A(v) + J(v)e + \text{higher order terms}$$

Newton for systems (cont.)

- Where $J(v)$ is the Jacobian system

$$J(v) = \left[\begin{array}{cccc} \frac{\partial f_1}{\partial u_1} & \frac{\partial f_1}{\partial u_2} & \dots & \frac{\partial f_1}{\partial u_N} \\ \frac{\partial f_2}{\partial u_1} & \frac{\partial f_2}{\partial u_2} & \dots & \frac{\partial f_2}{\partial u_N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_N}{\partial u_1} & \frac{\partial f_N}{\partial u_2} & \dots & \frac{\partial f_N}{\partial u_N} \end{array} \right] \Big|_{u=v}$$

- If $u=v+e$ is a solution, $0 = A(v) + J(v)e$ and

$$e = - [J(v)]^{-1} A(v)$$

- Leading to the iteration

$$v \leftarrow v - [J(v)]^{-1} A(v)$$

Newton's method in terms of the residual equation

- The nonlinear residual equation is

$$A(v + e) - A(v) = r$$

- Expanding $A(v+e)$ in a two-term Taylor series about v :

$$A(v) + J(v) e - A(v) = r$$

$$J(v) e = r$$

- Newton's method is thus:

$$\begin{aligned}r &= f - A(v) \\v &\leftarrow v + [J(v)]^{-1} r\end{aligned}$$

How does multigrid fit in?

- One obvious method is to use multigrid to solve $J(v)e = r$ at each iteration step. This method is called Newton-multigrid and can be very effective.
- However, we would like to use multigrid ideas to treat the nonlinearity directly.
- Hence, we need to specialize the multigrid components (relaxation, grid transfers, coarsening) for the nonlinear case.

What is nonlinear relaxation?

- Several of the common relaxation schemes have nonlinear counterparts. For $A(u)=f$, we describe the nonlinear Gauss-Seidel iteration:
 - For each $j=1, 2, \dots, N$
 - Set the j th component of the residual to zero and solve for v_j . That is, solve $(A(v))_j = f_j$.
- Equivalently,
 - For each $j=1, 2, \dots, N$
 - Find $s \in \mathcal{R}$ such that
$$(A(v + s \varepsilon_j))_j = f_j$$
where ε_j is the canonical j th unit basis vector

How is nonlinear Gauss-Seidel done?

- Each $(A(v))_j = f_j$ is a nonlinear scalar equation for v_j . We use the scalar Newton's method to solve!
- Example: $-u''(x) + u(x) e^{u(x)} = f$, may be discretized so that $(A(v))_j = f_j$ is given by

$$\frac{-v_{j-1} + 2v_j - v_{j+1}}{h^2} + v_j e^{v_j} = f_j \quad 1 \leq j \leq N-1$$

- Newton iteration for v_j is given by

$$v_j \leftarrow v_j - \frac{\frac{-v_{j-1} + 2v_j - v_{j+1}}{h^2} + v_j e^{v_j} - f_j}{\frac{2}{h^2} + e^{v_j} (1 + v_j)}$$

How do we do coarsening for nonlinear multigrid?

- Recall the nonlinear residual equation

$$A(v + e) - A(v) = r$$

- In multigrid, we obtain an approximate solution v^h on the fine grid, then solve the residual equation on the coarse grid.
- The residual equation on Ω^{2h} appears as

$$A^{2h}(v^{2h} + e^{2h}) - A^{2h}(v^{2h}) = r^{2h}$$

Look at the coarse residual equation

- We must evaluate the quantities on Ω^{2h} in

$$A^{2h}(\nu^{2h} + e^{2h}) - A^{2h}(\nu^{2h}) = r^{2h}$$

- Given ν^h , a fine-grid approximation, we restrict the residual to the coarse grid

$$r^{2h} = I_h^{2h}(f^h - A^h(\nu^h))$$

- For ν^{2h} we restrict ν^h by $\nu^{2h} = I_h^{2h}\nu^h$
- Thus,

$$A^{2h}(I_h^{2h}\nu^h + e^{2h}) = A^{2h}(I_h^{2h}\nu^h) + I_h^{2h}(f^h - A^h(\nu^h))$$

We've obtained a coarse-grid equation of the form . $A^{2h}(u^{2h}) = f^{2h}$

- Consider the coarse-grid residual equation:

$$A^{2h} \left(\underbrace{I_h^{2h} v^h + e^{2h}}_{u^{2h}} \right) = A^{2h} \left(I_h^{2h} v^h \right) + \underbrace{I_h^{2h} (f^h - A^h(v^h))}_{f^{2h}}$$

coarse-grid unknown All quantities are known

- We solve $A^{2h}(u^{2h}) = f^{2h}$ for $u^{2h} = I_h^{2h} v^h + e^{2h}$ and obtain

$$e^{2h} = u^{2h} - I_h^{2h} v^h$$

- We then apply the correction:

$$v^h = v^h + I_{2h}^h e^{2h}$$

FAS, the Full Approximation Scheme, two grid form

- Perform nonlinear relaxation on $A^h(u^h) = f^h$ to obtain an approximation v^h .
- Restrict the approximation and its residual

$$v^{2h} = I_h^{2h} v^h \quad r^{2h} = I_h^{2h} (f^h - A(v^h))$$

- Solve the coarse-grid residual problem

$$A^{2h}(u^{2h}) = A^{2h}(v^{2h}) + r^{2h}$$

- Extract the coarse-grid error

$$e^{2h} = u^{2h} - v^{2h}$$

- Interpolate and apply the correction

$$v^h = v^h + I_{2h}^h e^{2h}$$

A few observations about FAS

- If A is a linear operator then FAS reduces directly to the linear two-grid correction scheme.
- A fixed point of FAS is an exact solution to the fine-grid problem and an exact solution to the fine-grid problem is a fixed point of the FAS iteration.

A few observations about FAS, continued

- The FAS coarse-grid equation can be written as

$$A^{2h}(u^{2h}) = I_h^{2h} f^h + \tau_h^{2h}$$

where τ_h^{2h} is the so-called tau correction.

- In general, since $\tau_h^{2h} \neq 0$, the solution u^{2h} to the FAS coarse-grid equation is not the same as the solution to the original coarse-grid problem .

$$A^{2h}(u^{2h}) = I_h^{2h} f^h .$$

- The tau correction may be viewed as a way to alter the coarse-grid equations to enhance their approximation properties.

Still more observations about FAS

- FAS may be viewed as an inner and outer iteration: the outer iteration is the coarse-grid correction, the inner iteration the relaxation method.
- A true multilevel FAS process is recursive, using FAS to solve the nonlinear Ω^{2h} problem using Ω^{4h} . Hence, FAS is generally employed in a V- or W-cycling scheme.

And yet more observations about FAS!

- For linear problems we use FMG to obtain a good initial guess on the fine grid. Convergence of nonlinear iterations depends critically on having a good initial guess.
- When FMG is used for nonlinear problems the interpolant $I_{2h}^h u^{2h}$ is generally accurate enough to be in the basin of attraction of the fine-grid solver.
- Thus, one FMG cycle, whether FAS, Newton, or Newton-multigrid is used on each level, should provide a solution accurate to the level of discretization, unless the nonlinearity is extremely strong.

Intergrid transfers for FAS

- Generally speaking, the standard operators (linear interpolation, full weighting) work effectively in FAS schemes.
- In the case of strongly nonlinear problems, the use of higher-order interpolation (e.g., cubic interpolation) may be beneficial.
- For an FMG scheme, where $I_{2h}^h u^{2h}$ is the interpolation of a coarse-grid solution to become a fine-grid initial guess, higher-order interpolation is always advised.

What is $A^{2h}(u^{2h})$ in FAS?

- As in the linear case, there are two basic possibilities:
- $A^{2h}(u^{2h})$ is determined by discretizing the nonlinear operator $A(u)$ in the same fashion as was employed to obtain $A^h(u^h)$, except that the coarser mesh spacing is used.
- $A^{2h}(u^{2h})$ is determined from the Galerkin condition
$$A^{2h}(u^{2h}) = I_h^{2h} A^h(u^h) I_{2h}^h$$
where the action of the Galerkin product can be captured in an implementable formula.
- The first method is usually easier, and more common.

Nonlinear problems: an example

- Consider

$$-\Delta u(x, y) + \gamma u(x, y) e^{u(x, y)} = f(x, y)$$

on the unit square $[0,1] \times [0,1]$ with homogeneous Dirichlet boundary conditions and a regular Cartesian grid.

- Suppose the exact solution is

$$u(x, y) = (x^2 - x^3) \sin(3\pi y)$$

Discretization of nonlinear example

- The operator can be written (sloppily) as

$$\frac{1}{h^2} \begin{pmatrix} -1 & & \\ -1 & 4 & -1 \\ & -1 & \end{pmatrix} u_{i,j}^h + \gamma u_{i,j}^h e^{u_{i,j}^h} = f_{i,j}$$

$A^h(u^h)$

- The relaxation is given by

$$u_{i,j}^h \leftarrow u_{i,j}^h - \frac{(A^h(u^h))_{i,j} - f_{i,j}}{\frac{4}{h^2} + \gamma(1 + u_{i,j}^h) e^{u_{i,j}^h}}$$

FAS and Newton's method on

$$-\Delta u(x,y) + \gamma u(x,y) e^{u(x,y)} = f(x,y)$$

- FAS

	γ			
	1	10	100	1000
convergence factor	0.135	0.124	0.098	0.072
number of FAS cycles	12	11	11	10

- Newton's Method

	γ			
	1	10	100	1000
convergence factor	4.00E-05	7.00E-05	3.00E-04	2.00E-04
number of Newton iterations	3	3	3	4

Newton, Newton-MG, and FAS on

$$-\Delta u(x,y) + \gamma u(x,y) e^{u(x,y)} = f(x,y)$$

- Newton uses exact solve, Newton-MG is inexact Newton with a fixed number of inner V(2,1)-cycles the Jacobian problem, overall stopping criterion $\|r\|_2 < 10^{-10}$

Method	Outer iterations	Inner iterations	Megaflops
Newton	3		1660.6
Newton-MG	3	20	56.4
Newton-MG	4	10	38.5
Newton-MG	5	5	25.1
Newton-MG	10	2	22.3
Newton-MG	19	1	24.6
FAS	11		27.1

Comparing FMG-FAS and FMG-Newton

$$-\Delta u(x,y) + \gamma u(x,y) e^{u(x,y)} = f(x,y)$$

We will do one FMG cycle using a single FAS V(2,1)-cycle as the “solver” at each new level. We then follow that with sufficiently many FAS V(2,1)-cycles as is necessary to obtain $\|r\| < 10^{-10}$.

Next, we will do one FMG cycle using a Newton-multigrid step at each new level (with a single linear V(2,1)-cycle as the Jacobian “solver.”) We then follow that with sufficiently many Newton-multigrid steps as is necessary to obtain $\|r\| < 10^{-10}$.

Comparing FMG-FAS and FMG-Newton

$$-\Delta u(x,y) + \gamma u(x,y) e^{u(x,y)} = f(x,y)$$

Cycle	$\ r^h\ $	$\ e^h\ $	Mflops	$\ r^h\ $	$\ e^h\ $	Mflops	Cycle
FMG-FAS	1.10E-02	2.00E-05	3.1	1.06E-02	2.50E-05	2.4	FMG-Newton
FAS V	6.80E-04	2.40E-05	5.4	6.70E-04	2.49E-05	4.1	Newton-MG
FAS V	5.00E-05	2.49E-05	7.6	5.10E-05	2.49E-05	5.8	Newton-MG
FAS V	3.90E-06	2.49E-05	9.9	6.30E-06	2.49E-05	7.5	Newton-MG
FAS V	3.20E-07	2.49E-05	12.2	1.70E-06	2.49E-05	9.2	Newton-MG
FAS V	3.00E-08	2.49E-05	14.4	5.30E-07	2.49E-05	10.9	Newton-MG
FAS V	2.90E-09	2.49E-05	16.7	1.70E-07	2.49E-05	12.6	Newton-MG
FAS V	3.00E-10	2.49E-05	18.9	5.40E-08	2.49E-05	14.3	Newton-MG
FAS V	3.20E-11	2.49E-05	21.2	1.70E-08	2.49E-05	16.0	Newton-MG
				5.50E-09	2.49E-05	17.7	Newton-MG
				1.80E-09	2.49E-05	19.4	Newton-MG
				5.60E-10	2.49E-05	21.1	Newton-MG
				1.80E-10	2.49E-05	22.8	Newton-MG
				5.70E-11	2.49E-05	24.5	Newton-MG

Comparing FMG-FAS and FMG-Newton

$$-\Delta u(x,y) + \gamma u(x,y) e^{u(x,y)} = f(x,y)$$

Cycle	$\ r^h\ $	$\ e^h\ $	Mflops	$\ r^h\ $	$\ e^h\ $	Mflops	Cycle
FMG-FAS	1.10E-02	2.00E-05	3.1	1.06E-02	2.50E-05	2.4	FMG-Newton
FAS V	6.80E-04	2.40E-05	5.4	6.70E-04	2.49E-05	4.1	Newton-MG
FAS V	5.00E-05	2.49E-05	7.6	5.10E-05	2.49E-05	5.8	Newton-MG
FAS V	3.90E-06	2.49E-05	9.9	6.30E-06	2.49E-05	7.5	Newton-MG
FAS V	3.20E-07	2.49E-05	12.2	1.70E-06	2.49E-05	9.2	Newton-MG
FAS V	3.00E-08	2.49E-05	14.4	5.30E-07	2.49E-05	10.9	Newton-MG
FAS V	2.90E-09	2.49E-05	16.7	1.70E-07	2.49E-05	12.6	Newton-MG
FAS V	3.00E-10	2.49E-05	18.9	5.40E-08	2.49E-05	14.3	Newton-MG
FAS V	3.20E-11	2.49E-05	21.2	1.70E-08	2.49E-05	16.0	Newton-MG
				5.50E-09	2.49E-05	17.7	Newton-MG
				1.80E-09	2.49E-05	19.4	Newton-MG
				5.60E-10	2.49E-05	21.1	Newton-MG
				1.80E-10	2.49E-05	22.8	Newton-MG
				5.70E-11	2.49E-05	24.5	Newton-MG

Note: In both cases, the single FMG cycle has achieved convergence to the level of the discretization error

Multigrid: increasingly, the right tool!

- Multigrid has been proven on a wide variety of problems, especially elliptic PDEs, but has also found application among parabolic & hyperbolic PDEs, integral equations, evolution problems, geodesic problems, etc.
- *With the right setup, multigrid is frequently an optimal (i.e., $O(N)$) solver.*
- Multigrid is of great interest because it is one of the very few scalable algorithms, and can be parallelized readily and efficiently!