A Multigrid Tutorial part two

William L. Briggs
Department of Mathematics
University of Colorado at Denver

Van Emden Henson
Center for Applied Scientific Computing
Lawrence Livermore National Laboratory

Steve McCormick
Department of Applied Mathematics
University of Colorado at Boulder

This work was performed, in part, under the auspices of the United States Department of Energy by University of California Lawrence Livermore National Laboratory under contract number W-7405-Eng-48. UCRL-VG-136819 Pt 2

Outline

- Nonlinear Problems
- Neumann Boundary Conditions
- Anisotropic Problems
- Variable Mesh Problems
- Variable Coefficient Problems
- Algebraic Multigrid

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:

$$A(u) = f$$

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

$$\boxed{A(u) - A(v) = r}$$

- 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:

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

- Dropping higher order terms, if $x+s$ is a solution,

$$0 = F(s) + sF'(x) \qquad \therefore \qquad s = -F(x) / F'(x)$$

- 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

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

- 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

$$J(v) = \left. \begin{pmatrix} \frac{\partial f_1}{\partial u_1} & \frac{\partial f_1}{\partial u_2} & \cdots & \frac{\partial f_1}{\partial u_N} \\ \frac{\partial f_2}{\partial u_1} & \frac{\partial f_2}{\partial u_2} & \cdots & \frac{\partial f_2}{\partial u_N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_N}{\partial u_1} & \frac{\partial f_N}{\partial u_2} & \cdots & \frac{\partial f_N}{\partial u_N} \end{pmatrix} \right|_{u = v}$$

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

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

- Leading to the iteration

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

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:

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

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

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 us 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, …, N
    - Set the *jth* 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, …, N
    - Find $s \in \Re$ such that

$$(A(v + s\,\varepsilon_j))_j = f_j$$

where ε_j is the canonical *jth* 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

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

- Newton iteration for v_j is given by

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

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

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

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

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

- For v^{2h} we restrict v^h by $v^{2h} = I_h^{2h} v^h$
- Thus,

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

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

- Consider the coarse-grid residual equation:

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

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

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

- 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

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

- 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

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

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 .

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

- 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] × [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

$$\underbrace{\frac{1}{h^2}\begin{pmatrix} & -1 & \\ -1 & 4 & -1 \\ & -1 & \end{pmatrix} u^h_{i,j} + \gamma u^h_{i,j} e^{u^h_{i,j}}}_{A^h(u^h)} = f_{i,j}$$

- The relaxation is given by

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

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	$\lVert r^h \rVert$	$\lVert e^h \rVert$	Mflops	$\lVert r^h \rVert$	$\lVert e^h \rVert$	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

Outline

- ✓ Nonlinear Problems
- Neumann Boundary Conditions
- Anisotropic Problems
- Variable Mesh Problems
- Variable Coefficient Problems
- Algebraic Multigrid

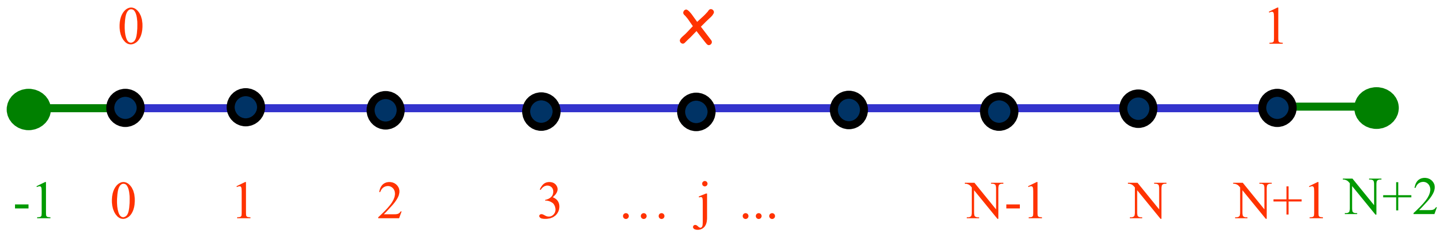
Neumann Boundary Conditions

- Consider the (1-d) problem

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

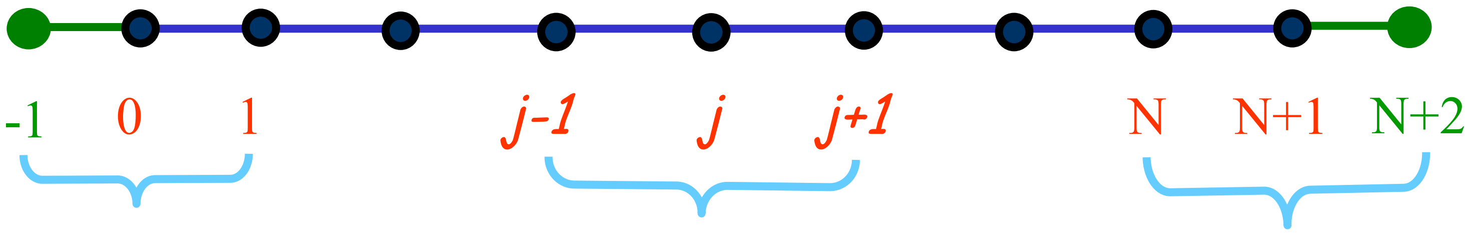
- We discretize this on the interval [0,1] with $h = \dfrac{1}{N+1}$ grid spacing $x_j = jh$ for *j=1,2, …, N+1* .

- We extend the interval with two ghost points



We use central differences

- We approximate the derivatives with differences, using the ghost points:



-1 0 1 *j-1* *j* *j+1* N N+1 N+2

$$u'(0) \approx \frac{u_1 - u_{-1}}{2h} \qquad u''(x_j) \approx \frac{-u_{j-1} + 2u_j - u_{j+1}}{h^2} \qquad u'(1) \approx \frac{u_{N+2} - u_N}{2h}$$

- Giving the system

$$\frac{-u_{j-1} + 2u_j - u_{j+1}}{h^2} = f_j \qquad 0 \le j \le N+1$$

$$\frac{u_1 - u_{-1}}{2h} = 0 \qquad \frac{u_{N+2} - u_N}{2h} = 0$$

Eliminating the ghost points

- Use the boundary conditions to eliminate $u_{-1},\ u_{N+2}$

$$\frac{u_1 - u_{-1}}{2h} = 0 \quad \Rightarrow \quad u_{-1} = u_1 \qquad \frac{u_{N+2} - u_N}{2h} = 0 \quad \Rightarrow \quad u_{N+2} = u_N$$

- Eliminating the ghost points in the j=0 and j=N+1 equations gives the (N+2)x(N+2) system of equations:

$$\frac{-u_{j-1} + 2u_j - u_{j+1}}{h^2} = f_j \qquad 0 \le j \le N+1$$

$$\frac{2u_0 - 2u_1}{h^2} = f_0 \qquad \frac{-2u_N + 2u_{N+1}}{h^2} = f_{N+1}$$

We write the system in matrix form

- We can write $A^h u^h = f^h$, where

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

- Note that A^h is (N+2)x(N+2), nonsymmetric, and the system involves unknowns u_0^h and u_{N+1}^h at the boundaries.

We must consider a compatibility condition

- The problem $-u''(x) = f(x)$, for $0 < x < 1$ and with $u'(0) = u'(1) = 0$ is not well-posed!
- If *u(x)* is a solution, so is *u(x)+c* for any constant *c*.
- We cannot be certain a solution exists. If one does, it must satisfy

$$-\int_0^1 u''(x)\,dx = \int_0^1 f(x)\,dx \quad \Rightarrow \quad -[u'(1) - u'(0)] = \int_0^1 f(x)\,dx$$

$$\boxed{0 = \int_0^1 f(x)\,dx}$$

- This integral compatibility condition is necessary! If *f(x)* doesn't satisfy it, there is no solution!

The well-posed system

- The compatibility condition is necessary for a solution to exist. In general, it is also sufficient, which can be proven that $-\frac{\partial^2}{\partial x^2}$ is a well-behaved operator in the space of functions *u(x)* that have zero mean.
- Thus we may conclude that if *f(x)* satisfies the compatibility condition, this problem is well-posed:

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

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

$$\int_0^1 u(x)\,dx = 0$$

- The last says that of all solutions *u(x)+c* we choose the one with zero mean.

The discrete problem is not well posed

- Since all row sums of A^h are zero, $1^h \in \mathrm{NS}(A^h)$
- Putting A^h into row-echelon form shows that

  $\dim(\mathrm{NS}(A^h)) = 1$ hence $\mathrm{NS}(A^h) = \mathrm{span}(1^h)$
- By the Fundamental Theorem of Linear Algebra, A^h has a solution if and only if $f^h \perp NS((A^h)^T)$
- It is easy to show that $NS((A^h)^T) = c(1/2, 1, 1, \dots, 1, 1/2)^T$
- Thus, $A^h u^h = f^h$ has a solution if and only if

  $f^h \perp c(1/2, 1, 1, \dots, 1, 1/2)^T$
- That is,

$$\boxed{\frac{1}{2} f_0^h + \sum_{j=1}^{N} f_j^h + \frac{1}{2} f_{N+1}^h = 0}$$

We have two issues to consider

- Solvability. A solution exists iff $f^h \perp NS(\,(A^h)^T)$

- Uniqueness. If u^h solves $A^h u^h = f^h$ so does $u^h + c1^h$

- Note that if $A^h = (A^h)^T$ then $\mathrm{NS}(A^h) = \mathrm{NS}((A^h)^T)$ and solvability and uniqueness can be handled together

- This is easily done. Multiply 1st and last equations by 1/2, giving

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

The new system is symmetric

- We have the symmetric system $\hat{A}^h u^h = \hat{f}^h$:

$$\frac{1}{h^2}\begin{pmatrix} 1 & -1 & & & & \\ -1 & 2 & -1 & & & \\ & -1 & 2 & -1 & & \\ & & & \ddots & \ddots & \\ & & & -1 & 2 & -1 \\ & & & & -1 & 1 \end{pmatrix}\begin{pmatrix} u_0^h \\ u_1^h \\ u_2^h \\ \vdots \\ u_N^h \\ u_{N+1}^h \end{pmatrix} = \begin{pmatrix} f_0^h/2 \\ f_1^h \\ f_2^h \\ \vdots \\ f_N^h \\ f_{N+1}^h/2 \end{pmatrix}$$

- Solvability is guaranteed by ensuring that $\hat{f}^h$ is orthogonal to the constant vector 1^h :

$$\boxed{\left\langle \hat{f}^h, 1^h \right\rangle = \sum_{j=0}^{N+1} \hat{f}_j^h = 0}$$

The well-posed discrete system

- The (N+3)x(N+2) system is:

$$\frac{-u_{j-1} + 2u_j - u_{j+1}}{h^2} = f_j \qquad 0 \le j \le N+1$$

$$\frac{u_0 - u_1}{h^2} = \frac{f_0}{2}$$

$$\frac{-u_N + u_{N+1}}{h^2} = \frac{f_{N+1}}{2}$$

$$\sum_{i=0}^{N+1} u_i^h = 0$$

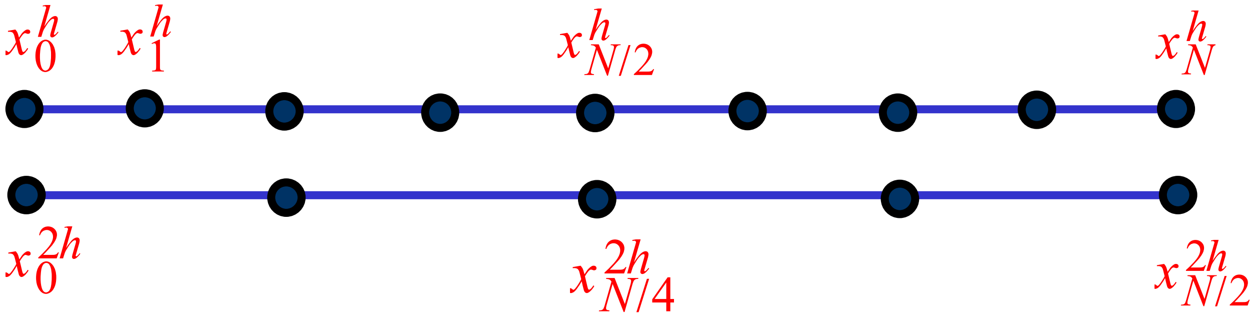
(choose the zero mean solution)

or, more simply

$$\boxed{\begin{array}{c} \widehat{A}^h u^h = \widehat{f}^h \\ \left\langle u^h, 1^h \right\rangle = 0 \end{array}}$$

Multigrid for the Neumann Problem

- We must have the interval endpoints on all grids



- Relaxation is performed at all points, including endpoints:

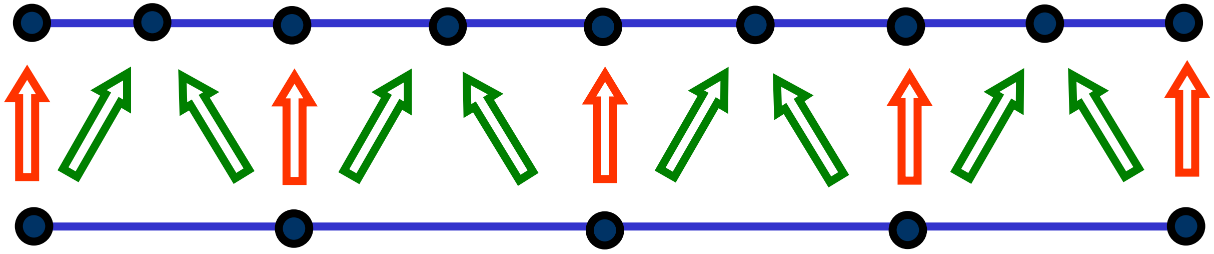
$$v_0^h \leftarrow v_1^h + h^2 \hat{f}_0^h \qquad v_j^h \leftarrow \frac{v_{j-1}^h + + v_{j+1}^h + h^2 \hat{f}_j^h}{2} \qquad v_{N+1}^h \leftarrow v_N^h + h^2 \hat{f}_{N+1}^h$$

- We add a global Gram-Schmidt step after relaxation on each level to enforce the zero-mean condition

$$v^h \leftarrow v^h - \frac{\left\langle v^h, 1^h \right\rangle}{\left\langle 1^h, 1^h \right\rangle} 1^h$$

Interpolation must include the endpoints

- We use linear interpolation:



$$I_{2h}^{h} = \begin{pmatrix} 1 & & & & \\ 1/2 & 1/2 & & & \\ & 1 & & & \\ & 1/2 & 1/2 & & \\ & & 1 & & \\ & & 1/2 & 1/2 & \\ & & & 1 & \\ & & & 1/2 & 1/2 \\ & & & & 1 \end{pmatrix} \begin{array}{c} \\ \Longleftarrow \\ \\ \Longleftarrow \\ \\ \Longleftarrow \\ \\ \Longleftarrow \\ \\ \end{array}$$

Restriction also treats the endpoints

- For restriction, we use $I_h^{2h} = \frac{1}{2}(I_{2h}^h)^T$, yielding the values

$$\hat{f}_0^{2h} = \frac{1}{2}\hat{f}_0^h + \frac{1}{4}\hat{f}_1^h$$

$$\hat{f}_j^{2h} = \frac{1}{4}\hat{f}_{2j-1}^h + \frac{1}{2}\hat{f}_{2j}^h + \frac{1}{4}\hat{f}_{2j+1}^h$$

$$\hat{f}_{N+1}^{2h} = \frac{1}{4}\hat{f}_N^h + \frac{1}{2}\hat{f}_{N+1}^h$$

The coarse-grid operator

- We compute the coarse-grid operator using the Galerkin condition $\widehat{A}^{2h} = I_h^{2h}\,\widehat{A}^{h}\,I_{2h}^{h}$

$j^{2h}=$	0		1
ε_0^{2h}	1		0
$I_{2h}^{h}\varepsilon_0^{2h}$	1	$\frac{1}{2}$	0
$\widehat{A}_h^{2h}\,I_{2h}^{h}\varepsilon_0^{2h}$	$\frac{1}{2h^2}$	0	$-\frac{1}{2h^2}$
$I_h^{2h}\,\widehat{A}^{h}\,I_{2h}^{h}\varepsilon_0^{2h}$	$\frac{1}{4h^2}$		$-\frac{1}{4h^2}$

$$\widehat{A}^{2h} = \frac{1}{4h^2}\begin{pmatrix} 1 & -1 & & & & \\ -1 & 2 & -1 & & & \\ & -1 & 2 & -1 & & \\ & & & \ddots & \ddots & \\ & & & -1 & 2 & -1 \\ & & & & -1 & 1 \end{pmatrix}$$

Coarse-grid solvability

- Assuming $\hat{f}^h$ satisfies $\langle \hat{f}^h, 1^h \rangle = 0$, the solvability condition, we can show that theoretically the coarse-grid problem $\hat{A}^{2h} u^{2h} = I_h^{2h}(\hat{f}^h - \hat{A}^h v^h)$ is also solvable.

- To be certain numerical round-off does not perturb solvability, we incorporate a Gram-Schmidt like step each time a new right-hand side $\hat{f}^{2h}$ is generated for the coarse grid:

$$\hat{f}^{2h} \leftarrow \hat{f}^{2h} - \frac{\langle \hat{f}^{2h}, 1^{2h} \rangle}{\langle 1^{2h}, 1^{2h} \rangle} 1^{2h}$$

Neumann Problem: an example

- Consider the problem $-u''(x) = 2x - 1$,

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

  which has $u(x) = \frac{x^2}{2} - \frac{x^3}{3} + c$ as a solution for any c ($c=-1/12$ gives the zero mean solution).

grid size N	$\|r^h\|$	average conv. factor	$\|e^h\|$	number of cycles
31	6.30E-11	0.079	9.70E-05	9
63	1.90E-11	0.089	2.40E-05	10
127	2.60E-11	0.093	5.90E-06	10
255	3.70E-11	0.096	1.50E-06	10
511	5.70E-11	0.100	3.70E-07	10
1027	8.60E-11	0.104	9.20E-08	10
2047	2.10E-11	0.112	2.30E-08	10
4095	5.20E-11	0.122	5.70E-09	10

Outline

- ✓ Nonlinear Problems
- ✓ Neumann Boundary Conditions
- Anisotropic Problems
- Variable Mesh Problems
- Variable Coefficient Problems
- Algebraic Multigrid

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, mash spacing in the different coordinate directions

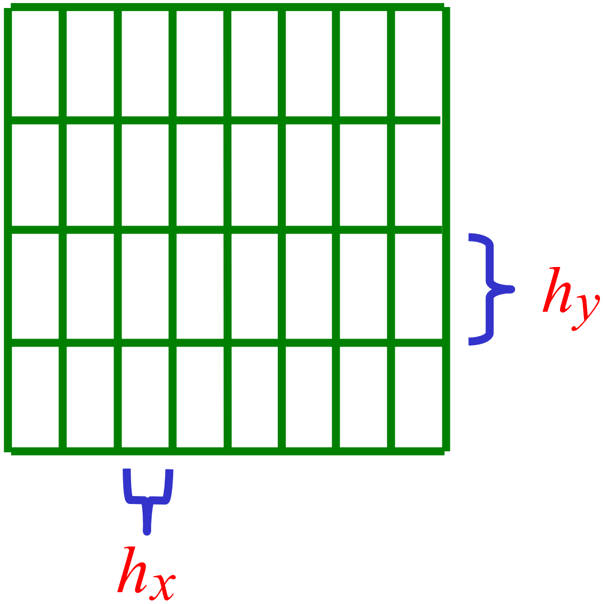
We consider two types of anisotropy

- Different coefficients on the derivatives

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

discretized on a uniform grid with spacing h .

- 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

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

$$\Bigg\Downarrow$$

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

$$\Bigg\Uparrow$$

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

Why standard multigrid fails

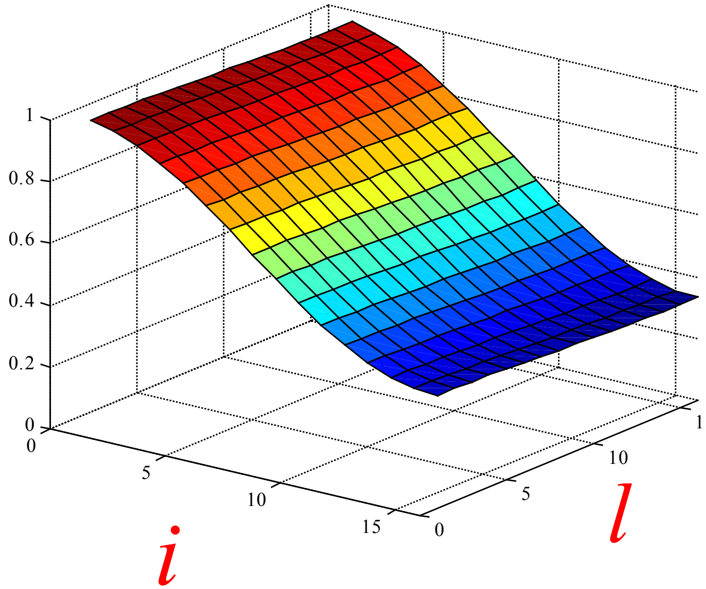
- Note that $A^h = \frac{1}{h^2}\begin{pmatrix} & -\alpha & \\ -1 & 2+2\alpha & -1 \\ & -\alpha & \end{pmatrix}$ has weak connections in the y-direction. MG convergence factors degrade as α gets small. Poor performance at $\alpha = 0.1$.

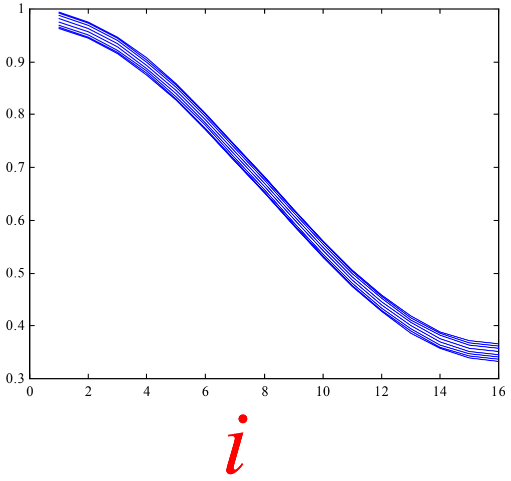
- Consider $\alpha \Rightarrow 0$. $A^h \Rightarrow \frac{1}{h^2}\begin{pmatrix} & 0 & \\ -1 & 2+2\alpha & -1 \\ & 0 & \end{pmatrix}$
- This is a collection of disconnected 1-d problems!
- Point relaxation will smooth oscillatory errors in the x-direction (strong connections), but with no connections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

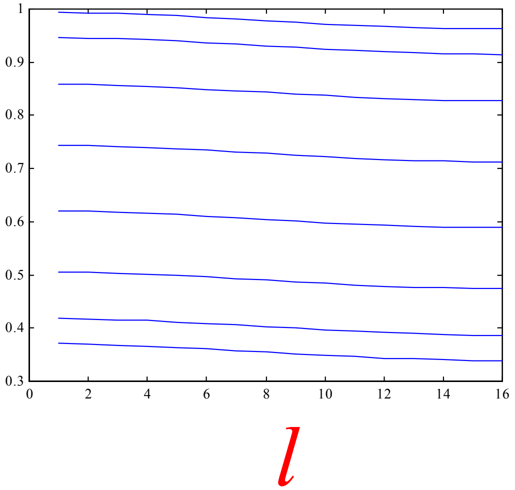
We analyze weighted Jacobi

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

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







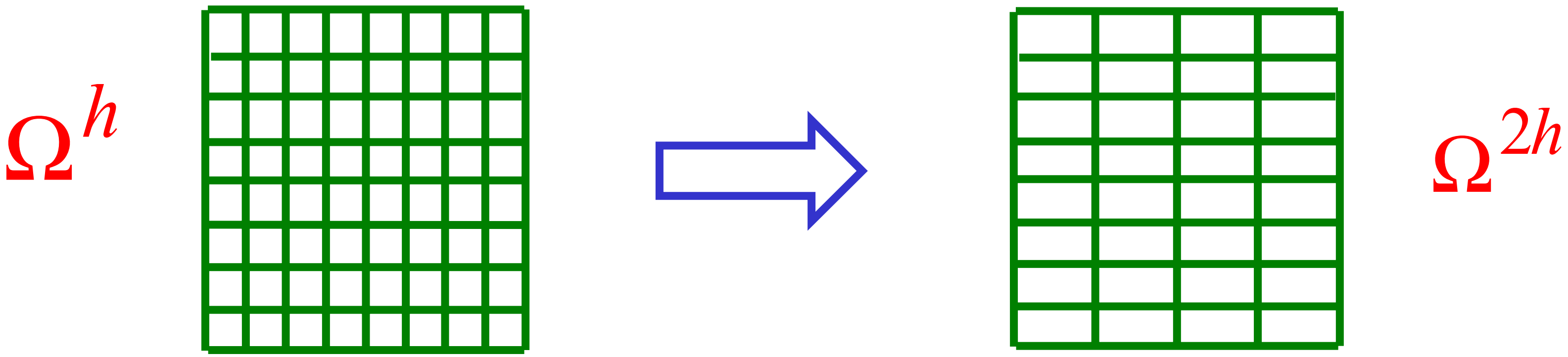
Two strategies for anisotropy

- **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 *y*-line.



- We do not coarsen along the remaining *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

$$v^h_{2j,k} = v^h_{2j,k} + v^{2h}_{j,k}$$

$$v^h_{2j+1,k} = v^h_{2j+1,k} + \frac{v^{2h}_{j,k} + v^{2h}_{j+1,k}}{2}$$

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

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

Line relaxation

- One sweep of line relaxation consists of solving a tridiagonal system for each line of constant *y*.

- The *kth* such system has the form $D v_k^h = g_k^h$ where v_k^h is the *kth* subvector of v^h with entries $(v_k^h)_j = v_{j,k}^h$ and the *kth* right-hand side subvector is

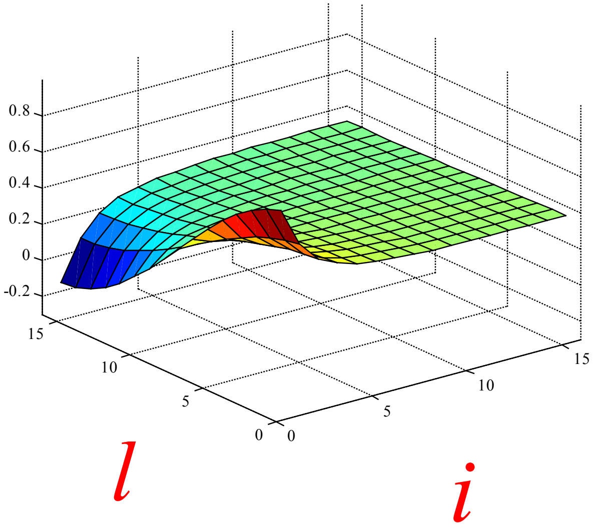
$$(g_k^h)_j = f_{j,k}^h + \frac{\alpha}{h^2}(v_{j,k-1}^h + v_{j,k+1}^h)$$

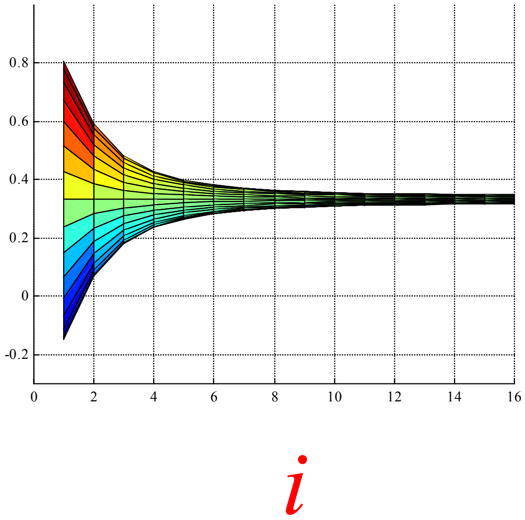
- Because D is tridiagonal, the *kth* system can be solved very efficiently.

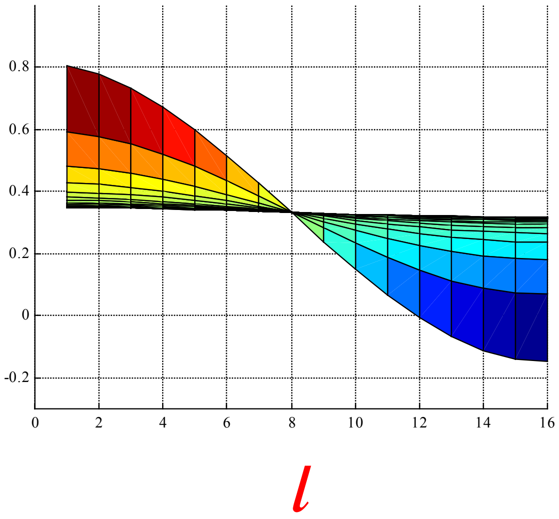
Why line relaxation works

- 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

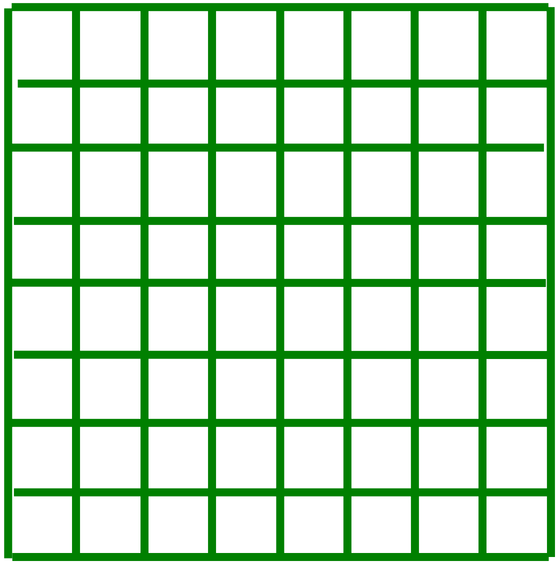
- We might not know the direction of weak coupling or it might vary.

- Suppose we want a method that can handle either

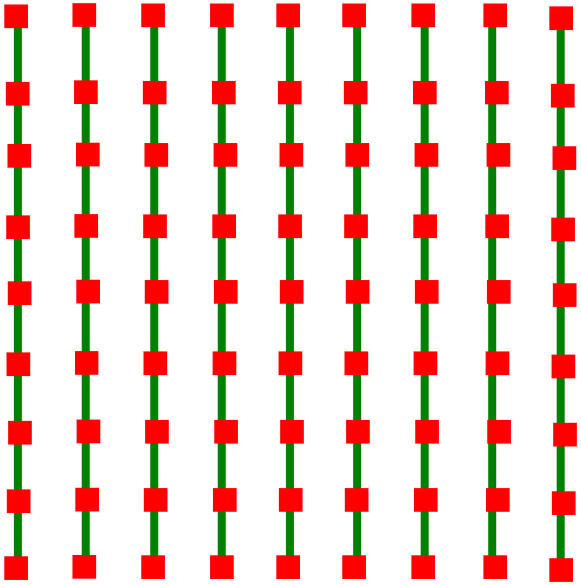
$$A_1{}^h = \frac{1}{h^2}\begin{pmatrix} & -\alpha & \\ -1 & 2+2\alpha & -1 \\ & -\alpha & \end{pmatrix} \quad \text{or} \quad A_2^h = \frac{1}{h^2}\begin{pmatrix} & -1 & \\ -\alpha & 2+2\alpha & -\alpha \\ & -1 & \end{pmatrix}$$

- 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 .

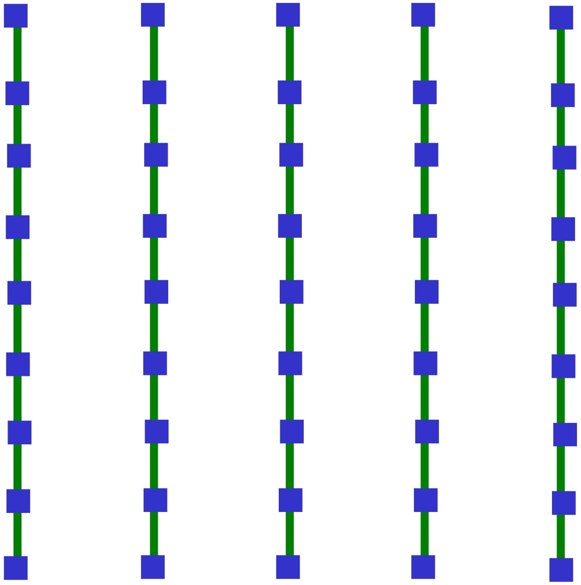
Semicoarsening with line relaxation



- 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

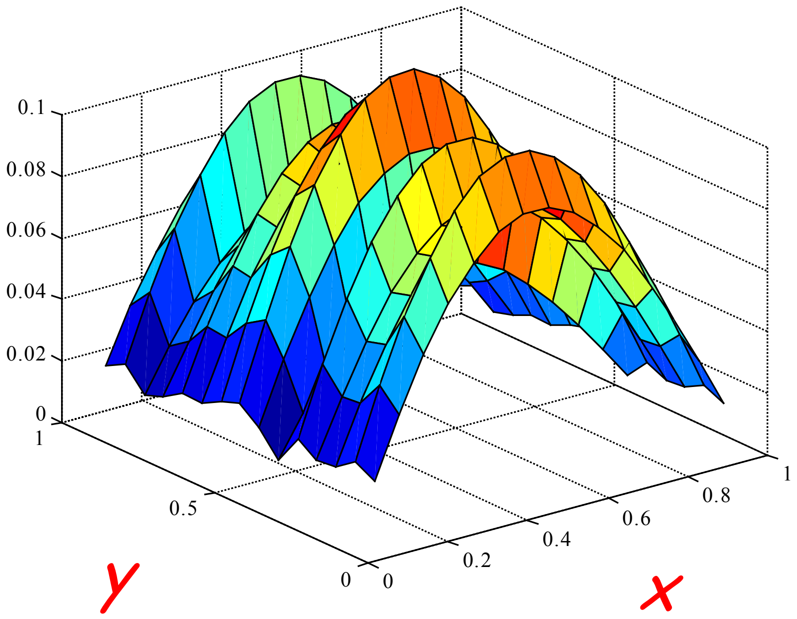
$$A^h = \frac{1}{h^2}\begin{pmatrix} & -\alpha & \\ -1 & 2+2\alpha & -1 \\ & -\alpha & \end{pmatrix}$$

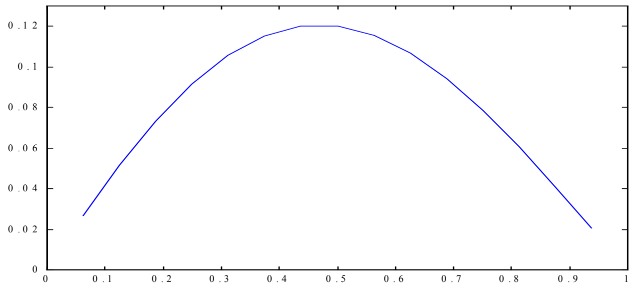
- Suppose $f(x,y) = 2(y-y^2) + 2\alpha(x-x^2)$ so the exact solution is given by $u(x,y) = (y-y^2)(x-x^2)$

- Observe that if α is small, the *x*-direction dominates while if α is large, the *y*-direction dominates

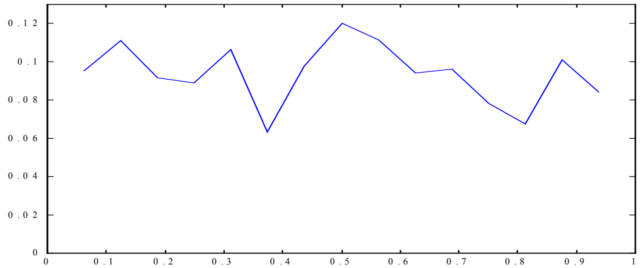
What is smooth error?

- 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:

	α							
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
emicoarsening 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

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.

Outline

- ✓ Nonlinear Problems
- ✓ Neumann Boundary Conditions
- ✓ Anisotropic Problems
- Variable Mesh Problems
- Variable Coefficient Problems
- Algebraic Multigrid

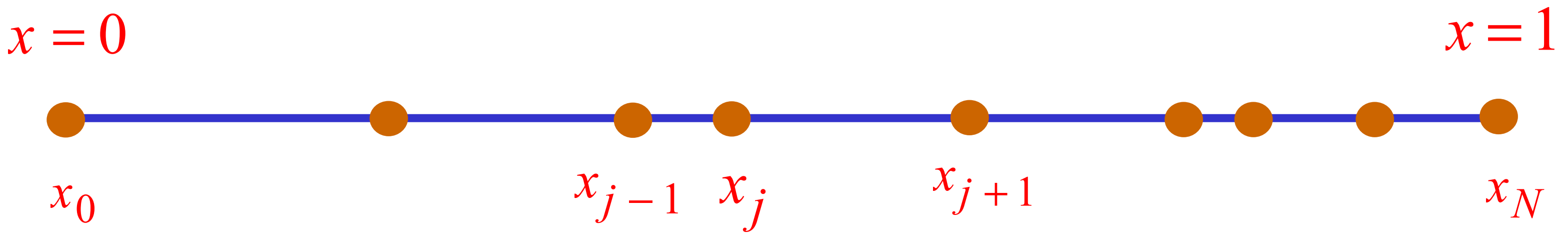
Variable Mesh Problems

- Non-uniform grids are commonly used to accommodate irregularities in problem domains

- Consider how we might approach the 1-d problem

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

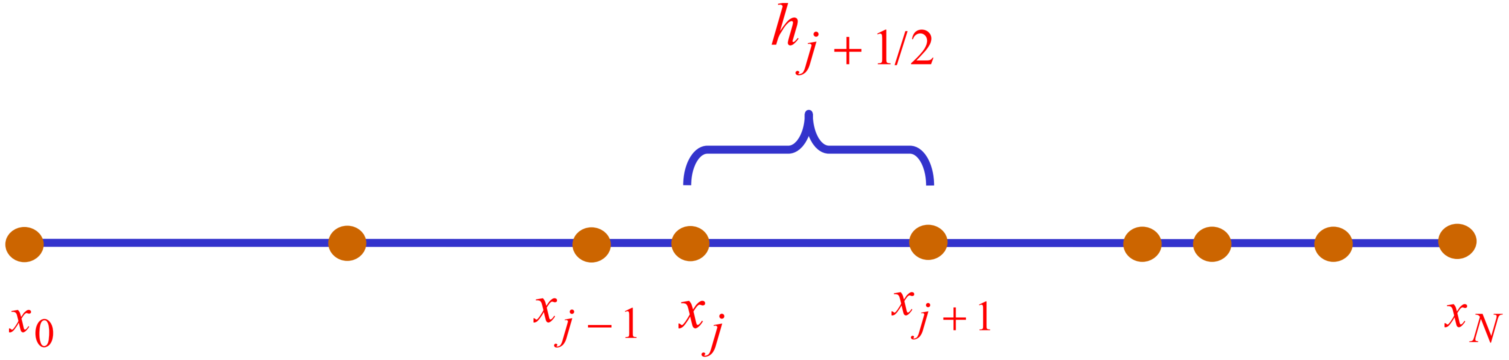
posed on this grid:



We need some notation for the mesh spacing

- Let N be a positive integer. We define the spacing interval between x_j and x_{j+1}:

$$h_{j+1/2} \equiv x_{j+1} - x_j \qquad j = 0, 1, \ldots, N-1$$



We define the discrete differential operator

- Using second order finite differences (and plugging through a mess of algebra!) we obtain this discrete representation for the problem:

$$-\alpha_j^h u_{j-1}^h + (\alpha_j^h + \beta_j^h) u_j^h - \beta_j^h u_{j+1}^h = f_j^h \qquad 1 \le j \le N-1$$

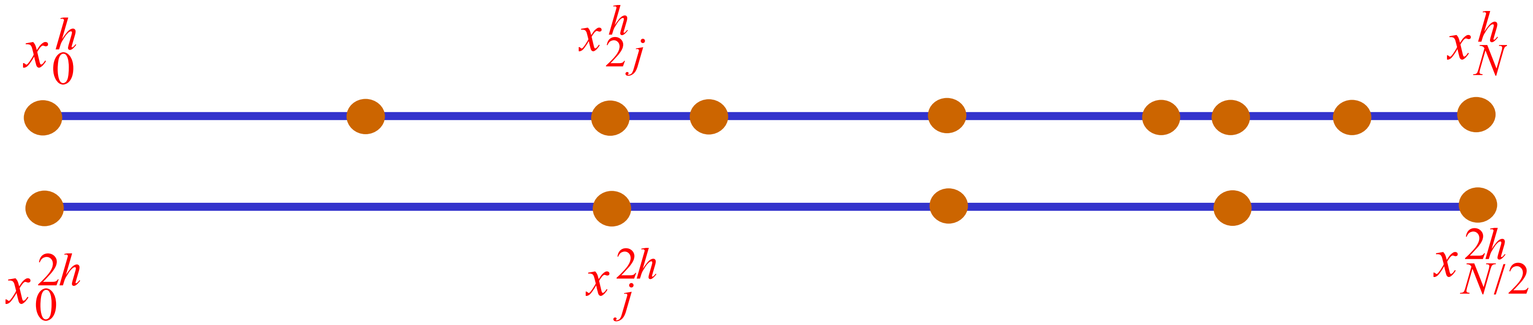
$$u_0^h = u_N^h = 0$$

- where

$$\alpha_j^h = \frac{2}{h_{j-1/2}(h_{j-1/2} + h_{j+1/2})} \qquad \beta_j^h = \frac{2}{h_{j+1/2}(h_{j-1/2} + h_{j+1/2})}$$

We modify standard multigrid to accommodate variable spacing

- We choose every second fine-grid point as a coarse-grid point



- We use linear interpolation, modified for the spacing. If $v^h = I_{2h}^h v^{2h}$, then for $1 \le j \le N/2 - 1$

$$v_{2j}^h = v_j^{2h} \qquad v_{2j+1}^h = \frac{h_{2j+3/2}\, v_j^{2h} + h_{2j+1/2}\, v_{j+1}^{2h}}{h_{2j+1/2} + h_{2j+3/2}}$$

We use the variational properties to derive restriction and A^{2h}.

$$A^{2h} = I_h^{2h} A^h I_{2h}^h \qquad I_h^{2h} = \frac{1}{2}\left(I_{2h}^h \right)^T$$

- This produces a stencil on Ω^{2h} that is similar, but not identical, to the fine-grid stencil. If the resulting system is scaled by $(h_{j-1/2} + h_{j+1/2})$, then the Galerkin product is the same as the fine-grid stencil.

- For 2-d problems this approach can be generalized readily to logically rectangular grids. However, for irregular grids that are not logically rectangular, AMG is a better choice.

Outline

- ✓ Nonlinear Problems
- ✓ Neumann Boundary Conditions
- ✓ Anisotropic Problems
- ✓ Variable Mesh Problems
- Variable Coefficient Problems
- Algebraic Multigrid

Variable coefficient problems

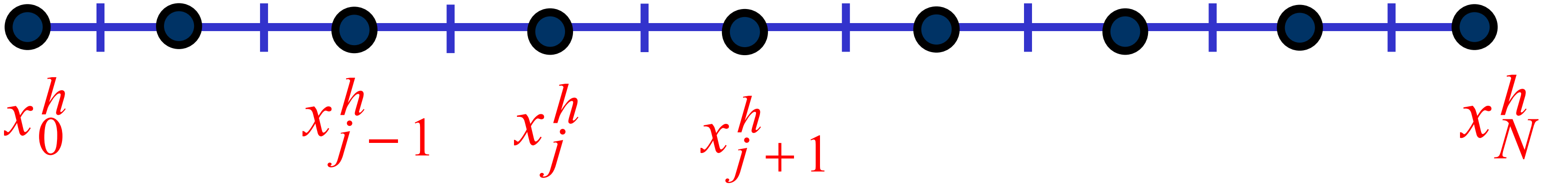
- A common difficulty is the variable coefficient problem, given in 1-d by

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

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

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

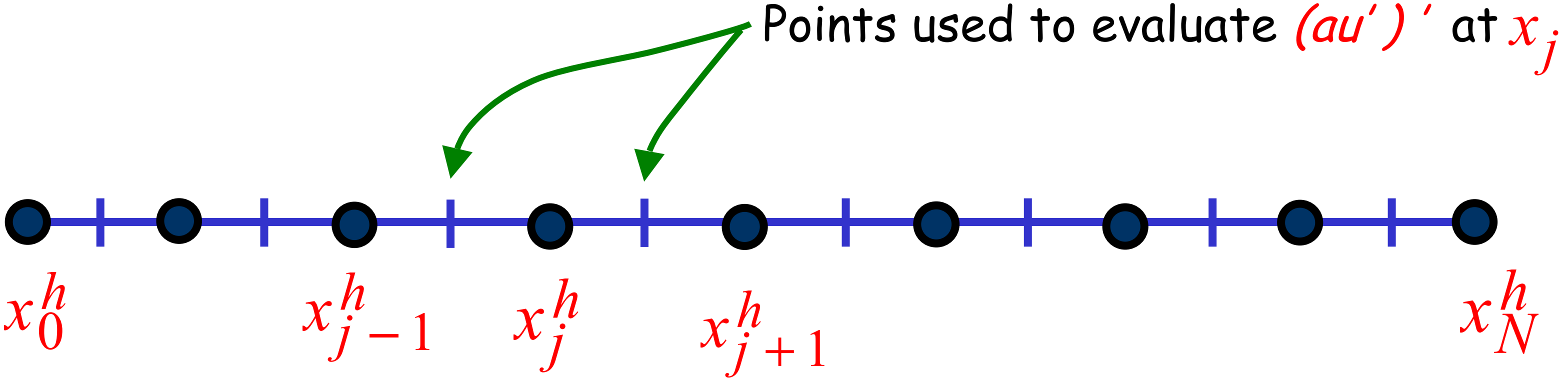
- We seek to develop a conservative, or self-adjoint, method for discretizing this problem.
- Assume we have available to us the values of a(x) at the midpoints of the grid $a_{j+1/2} \equiv a(x_{j+1/2})$



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:

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



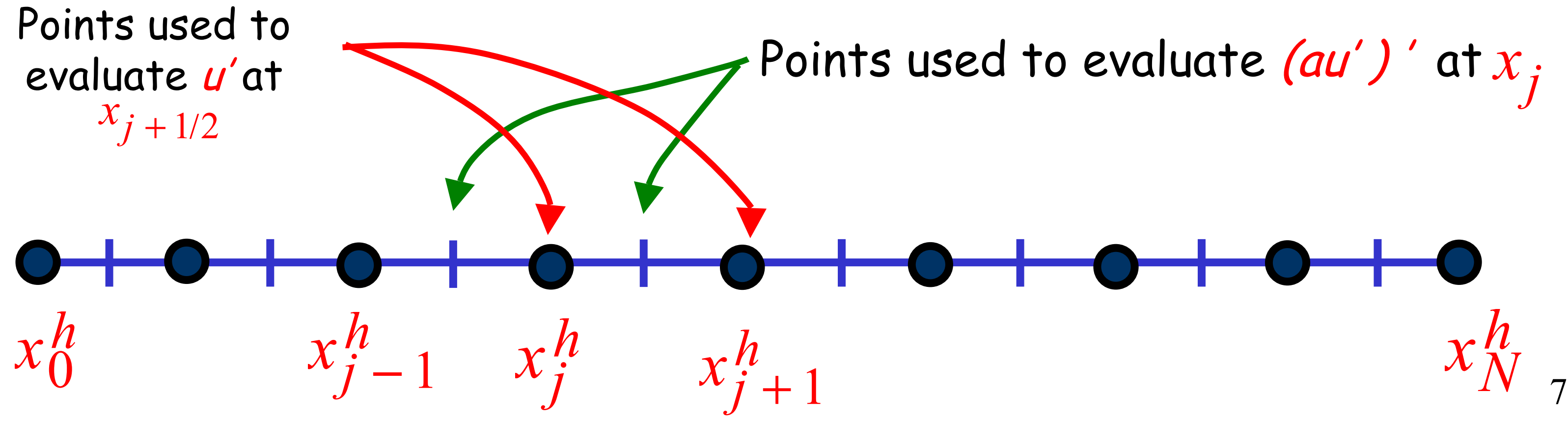
We discretize using central differences

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

$$(a\,u')|_{x_{j+1/2}} \approx a_{j+1/2}\frac{u_{j+1}-u_j}{h} \qquad (a\,u')|_{x_{j-1/2}} \approx a_{j-1/2}\frac{u_j-u_{j-1}}{h}$$

where

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


Points used to evaluate u' at $x_{j+1/2}$
Points used to evaluate $(au')'$ at x_j
x_0^h
x_{j-1}^h
x_j^h
x_{j+1}^h
x_N^h

The basic stencil is given

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

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

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

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

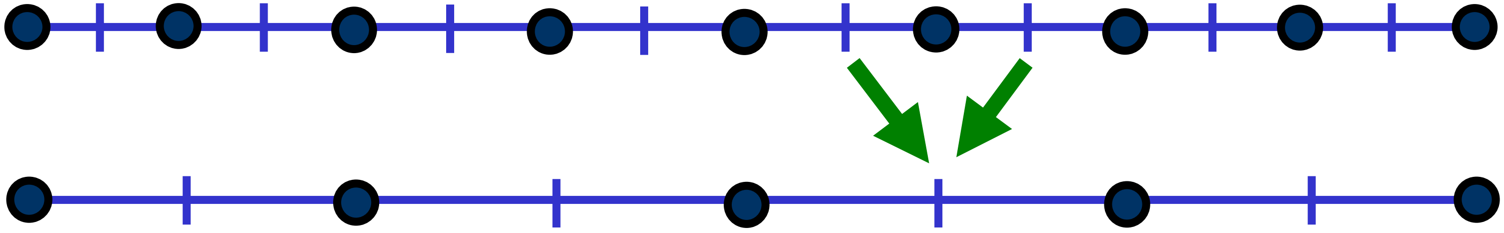
$$u_0 = u_N = 0$$

Coarsening the variable coefficient problem

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

$$A^{2h} = \frac{1}{(2h)^2}\left(-a^{2h}_{j-1/2} \quad a^{2h}_{j-1/2} + a^{2h}_{j+1/2} \quad -a^{2h}_{j+1/2} \right)$$

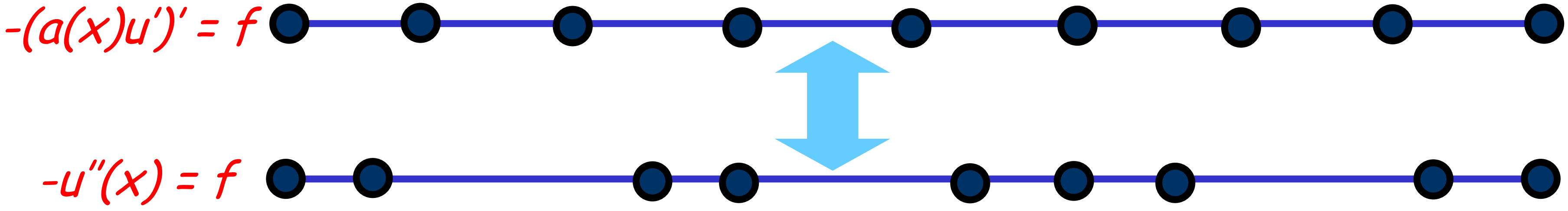
where $a^{2h}_{j+1/2} = \dfrac{a^h_{2j+1/2} + a^h_{2j+3/2}}{2}$



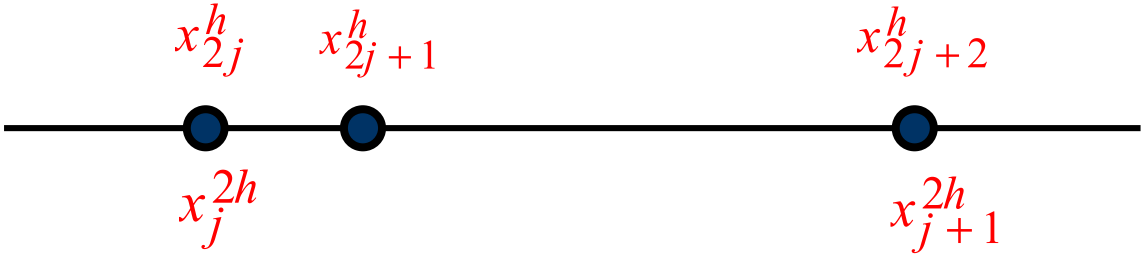
- The same stencil can be obtained via the Galerkin relation

Standard multigrid degrades if *a(x)* is highly variable

- It can be shown that the variable coefficient discretization is equivalent to using standard multigrid with simple averaging on the Poisson problem on a certain variable-mesh spacing.



- But simple averaging won't accurately represent smooth components if x_{2j+1}^h is close to x_{2j}^h but far from x_{2j+2}^h .



One remedy is to apply operator interpolation

- Assume that relaxation does not change smooth error, so the residual is approximately zero. Applying at x^h_{2j+1} yields

$$\frac{-a^h_{2j+1/2}\, e^h_{2j} + (a^h_{2j+1/2} + a^h_{2j+3/2})\, e^h_{2j+1} - a^h_{2j+3/2}\, e^h_{2j+2}}{h^2} = 0$$

- Solving for e^h_{2j+1}

$$\boxed{e^h_{2j+1} = \frac{a^h_{2j+1/2}\, e^{2h}_j + a^h_{2j+3/2}\, e^{2h}_{j+1}}{a^h_{2j+1/2} + a^h_{2j+3/2}}}$$

Thus, the operator induced interpolation is

$$v_{2j}^{h} = v_{j}^{2h}$$

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

- And, as usual, the restriction and coarse-grid operators are defined by the Galerkin relations

$$A^{2h} = I_{h}^{2h} A^{h} I_{2h}^{h} \qquad I_{h}^{2h} = c(I_{2h}^{h})^{T}$$

A Variable coefficient example

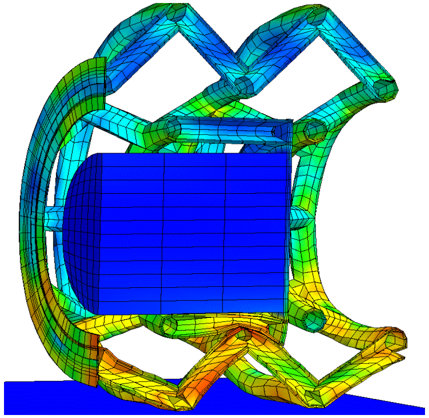
- We use V(2,1) cycle, full weighting, linear interpolation.
- We use $a(x) = \rho\, sin(k \pi x)$ and $a(x) = \rho\ \mathrm{rand}(k\pi x)$

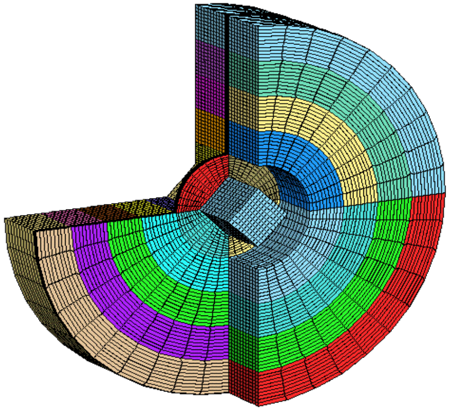
	$a(x) = \rho\, sin(k \pi x)$						$a(x) = \rho\ \mathrm{rand}(k\pi x)$
ρ	k=3	k=25	k=50	k=100	k=200	k=400	
0	0.085	0.085	0.085	0.085	0.085	0.085	0.085
0.25	0.084	0.098	0.098	0.094	0.093	0.083	0.083
0.5	0.093	0.185	0.194	0.196	0.195	0.187	0.173
0.75	0.119	0.374	0.387	0.391	0.39	0.388	0.394
0.85	0.142	0.497	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

Outline

- ✓ Nonlinear Problems
- ✓ Neumann Boundary Conditions
- ✓ Anisotropic Problems
- ✓ Variable Mesh Problems
- ✓ Variable Coefficient Problems
- Algebraic Multigrid

Algebraic multigrid: for unstructured-grids





- Automatically defines coarse "grid"

- AMG has two distinct phases:
  - setup phase: define MG components
  - solution phase: perform MG cycles

- AMG approach is opposite of geometric MG
  - fix relaxation (point Gauss-Seidel)
  - choose coarse "grids" and prolongation, *P*, so that error not reduced by relaxation is in *range(P)*
  - define other MG components so that coarse-grid correction eliminates error in *range(P)* (i.e., use Galerkin principle)

(in contrast, geometric MG fixes coarse grids, then defines suitable operators and smoothers)

AMG has two phases:

- Setup Phase
  - Select Coarse "grids," $\Omega^{m+1}, \; m = 1, 2, \ldots$
  - Define interpolation, $I_{m+1}^{m}, \; m = 1, 2, \ldots$
  - Define restriction and coarse-grid operators

$$I_m^{m+1} = (I_{m+1}^{m})^T \qquad\qquad A^{m+1} = I_m^{m+1} A^m I_{m+1}^{m}$$

- Solve Phase
  - Standard multigrid operations, e.g., V-cycle, W-cycle, FMG, FAS, etc

- Note: Only the selection of coarse grids does not parallelize well using existing techniques!

AMG fundamental concept: Smooth error = "small" residuals

- Consider the iterative method error recurrence

$$e^{k+1} = (I - Q^{-1}A)\, e^k$$

- Error that is slow to converge satisfies

$$(I - Q^{-1}A)\, e \approx e \quad \Rightarrow \quad Q^{-1}A\, e \approx 0$$

$$\Rightarrow \quad r \approx 0$$

- More precisely, it can be shown that smooth error satisfies

$$\| r \|_{D^{-1}} \ll \| e \|_A \qquad (1)$$

AMG uses strong connection to determine MG components

- It is easy to show from (1) that smooth error satisfies

$$\langle Ae, e \rangle \;\text{«}\; \langle De, e \rangle \qquad (2)$$

- Define *i* *is strongly connected to* *j* by

$$-a_{ij} \geq \theta \max_{k \neq i} \{-a_{ik}\}, \qquad 0 < \theta \leq 1$$

- For M-matrices, we have from (2)

$$\frac{1}{2} \sum_{i \neq j} \left(\frac{-a_{ij}}{2a_{ii}} \right) \left(\frac{e_i - e_j}{e_i} \right)^2 \;\text{«}\; 1$$

- implying that smooth error varies slowly in the direction of strong connections

Some useful definitions

- The set of strong connections of a variable u_i , that is, the variables upon whose values the value of u_i depends, is defined as

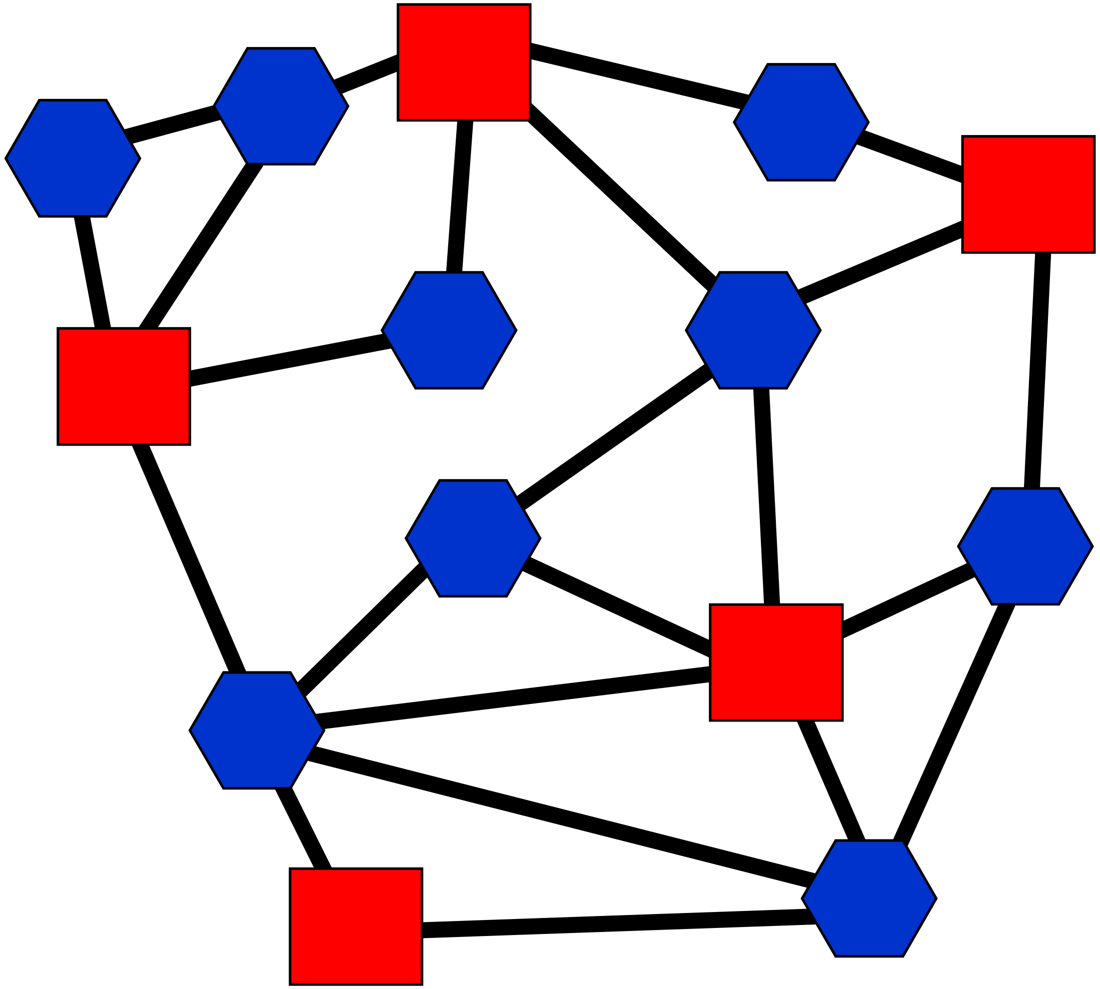
$$\boldsymbol{S_i} = \left\{ j : -a_{ij} > \theta \max_{j \neq i} -a_{ij} \right\}$$

- The set of points strongly connected *to* a variable u_i is denoted: $\boldsymbol{S_i^T} = \{ j : j \in S_i \}$.
- The set of coarse-grid variables is denoted *C*.
- The set of fine-grid variables is denoted *F*.
- The set of coarse-grid variables used to interpolate the value of the fine-grid variable $\boldsymbol{C_i}$ is denoted u_i .

Choosing the Coarse Grid

- Two Criteria
  - (C1) For each $i \in F$, each point $j \in S_i$ should either be in C or should be strongly connected to at least one point in C_i
  - (C2) C should be a maximal subset with the property that no two C-points are strongly connected to each other.
- Satisfying both (C1) and (C2) is sometimes impossible. We use (C2) as a guide while enforcing (C1).

Selecting the coarse-grid points



C-point selected (point with largest “value”)

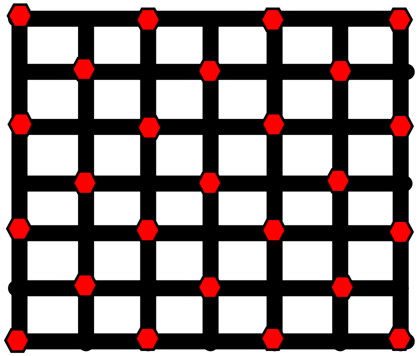
Neighbors of C-point become F-points

Next C-point selected (after updating “values”)

F-points selected, etc.

Examples: Laplacian Operator

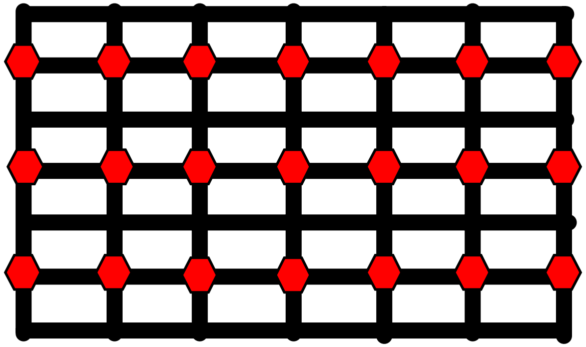
5-pt FD, 9-pt FE (quads), and 9-pt FE (stretched quads)



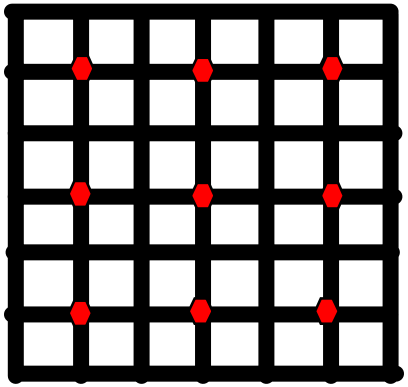
$$\begin{pmatrix} & -1 & \\ -1 & 4 & -1 \\ & -1 & \end{pmatrix}$$

5-pt FD

9-pt FE
(stretched quads)



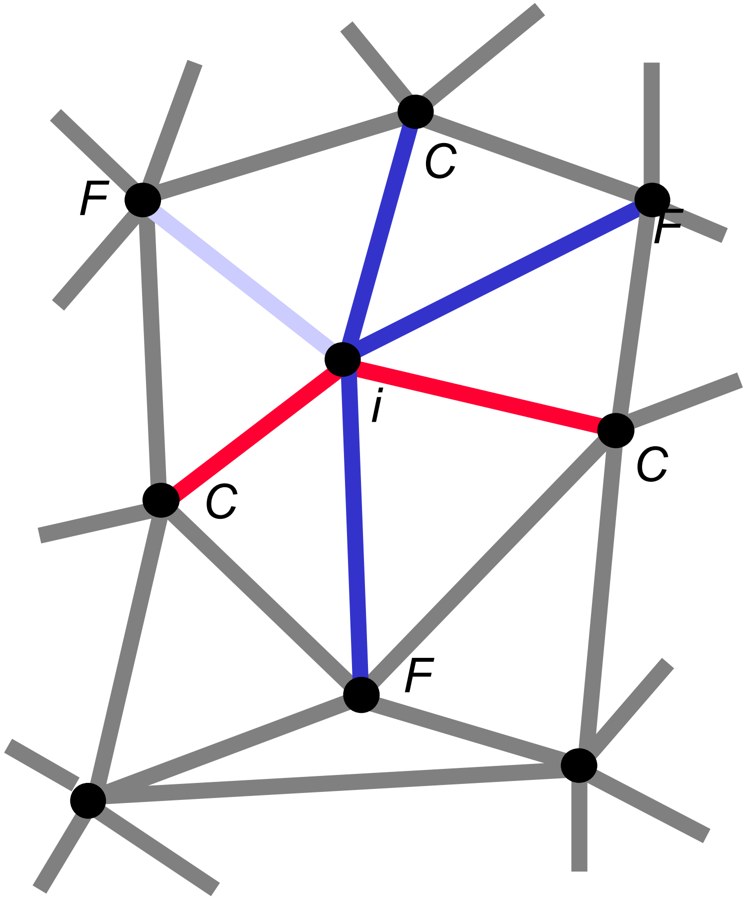
$$\begin{pmatrix} -1 & -4 & -1 \\ 2 & 8 & 2 \\ -1 & -4 & -1 \end{pmatrix}$$



$$\begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix}$$

9-pt FE (quads)

Prolongation is based on smooth error, strong connections (from M-matrices)



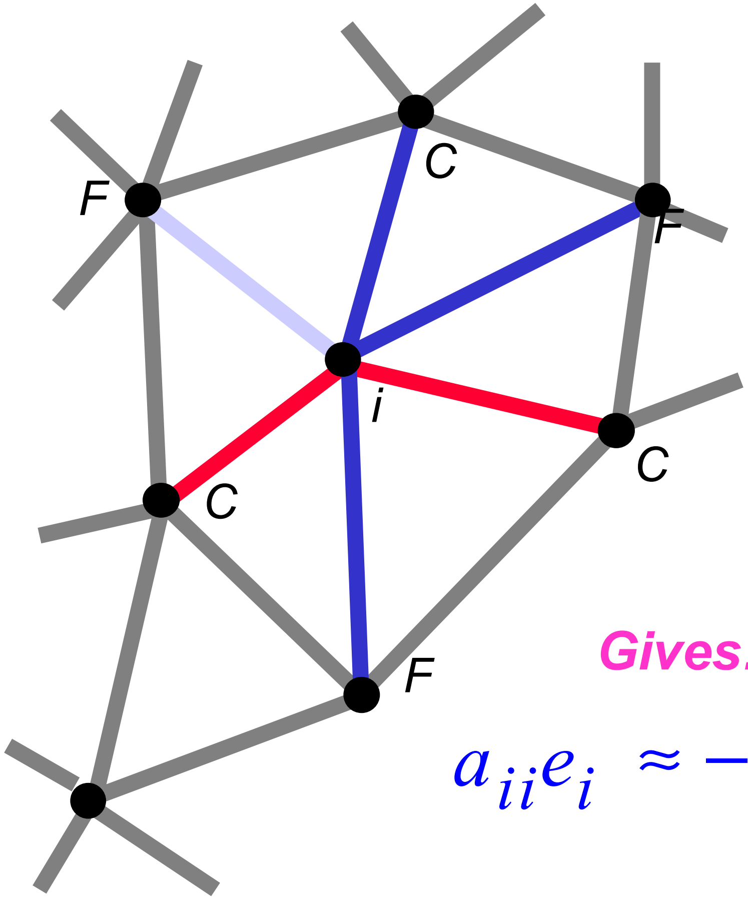
Smooth error is given by:

$$r_i = a_{ii} e_i + \sum_{j \in N_i} a_{ij} e_j \approx 0$$

Prolongation :

$$(Pe)_i = \begin{cases} e_i & , \quad i \in C \\ \sum_{k \in C_i} \omega_{ik} e_k & , \quad i \in F \end{cases}$$

Prolongation is based on smooth error, strong connections (from M-matrices)



Sets:

C_i ▬ ***Strongly connected** C**-pts.***

D_i^s ▬ ***Strongly connected** F**-pts.***

D_i^w ▬ ***Weakly connected points.***

The definition of smooth error,

$$a_{ii} e_i \approx - \sum_{j \neq i} a_{ij} e_j$$

Gives:

$$a_{ii} e_i \approx - \sum_{j \in C_i} a_{ij} e_j - \sum_{j \in D_i^s} a_{ij} e_j - \sum_{j \in D_i^w} a_{ij} e_j$$

Strong C ***Strong F*** ***Weak pts.***

Finally, the prolongation weights are defined

- In the smooth-error relation, use $e_j = e_i$ for weak connections. For the strong *F*-points use :

$$e_j = \left(\sum_{k \in C_i} a_{jk} e_k \right) \Big/ \left(\sum_{k \in C_i} a_{jk} \right)$$

yielding the prolongation weights:

$$w_{ij} = -\frac{a_{ij} + \sum\limits_{j \in D_i^s} \frac{a_{ik} a_{kj}}{\sum\limits_{m \in C_i} a_{km}}}{a_{ii} + \sum\limits_{n \in D_i^w} a_{in}}$$

AMG setup costs: a bad rap

- Many geometric MG methods need to compute prolongation and coarse-grid operators

- The only additional expense in the AMG setup phase is the coarse grid selection algorithm

- AMG setup phase is only 10-25% more expensive than in geometric MG and may be considerably less than that!

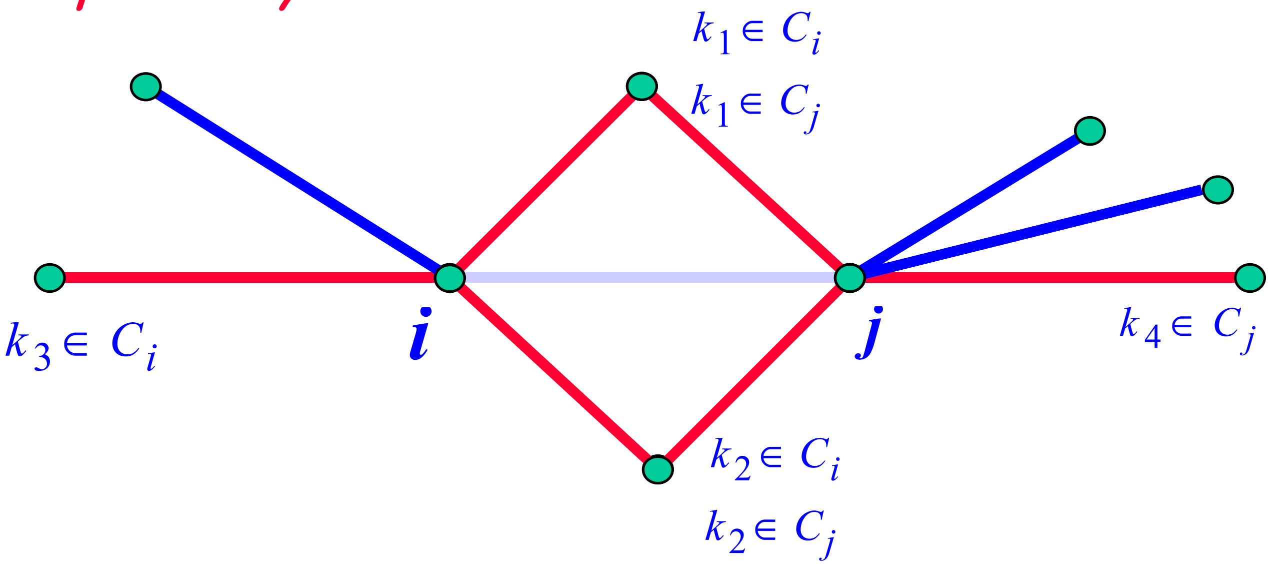
AMG Performance: Sometimes a Success Story

- AMG performs extremely well on the model problem (Poisson's equation, regular grid)- optimal convergence factor (e.g., 0.14) and scalability with increasing problem size.
- AMG appears to be both scalable and efficient on diffusion problems on unstructured grids (e.g., 0.1-0.3).
- AMG handles anisotropic diffusion coefficients on irregular grids reasonably well.
- AMG handles anisotropic operators on structured and unstructured grids relatively well (e.g., 0.35).

So, what could go wrong?

Strong F-F connections: weights are dependent on each other

- For point i the value e_j is interpolated from k_1, k_2, and is needed to make the interpolation weights for approximating e_i
- For point j the value e_i is interpolated from k_1, k_2, and is needed to make the interpolation weights for approximating e_j
- *It's an implicit system!*



Is there a fix?

- A Gauss-Seidel like iterative approach to weight definition is implemented. Usually two passes suffice. But does it work?

- **Frequently, it does:**

Convergence factors for Laplacian, stretched quadrilaterals

	theta	Standard	Iterative
$\Delta x = 10\Delta y$	0.25	0.47	0.14
	0.5	0.24	0.14
$\Delta x = 100\Delta y$	0.25	0.83	0.82
	0.5	0.53	0.23

AMG for systems

- How can we do AMG on systems?

$$\begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix}$$

- Naïve approach: "Block" approach (block Gauss-Seidel, using scalar AMG to "solve" at each cycle)

$$u \leftarrow (A_{11})^{-1}(f - A_{12}v)$$

$$v \leftarrow (A_{22})^{-1}(g - A_{21}u)$$

- **Great Idea! Except that it doesn't work!** **(relaxation does not evenly smooth errors in both unknowns)**

AMG for systems: a solution

- To solve the system problem, allow interaction between the unknowns at all levels:

$$A^k = \begin{pmatrix} A_{11}^k & A_{12}^k \\ A_{21}^k & A_{22}^k \end{pmatrix} \quad \text{and} \quad I_{k+1}^k = \begin{pmatrix} (I_{k+1}^k)_u & 0 \\ 0 & (I_{k+1}^k)_v \end{pmatrix}$$

- This is called the "unknown" approach.
- Results: 2-D elasticity, uniform quadrilateral mesh:

mesh spacing	0.125	0.0625	0.03135	0.015625
Convergence factor	0.22	0.35	0.42	0.44

How's it perform (vol I)?

Regular grids, plain, old, vanilla problems

- The Laplace Operator:

	Convergence		*Time*	*Setup*
Stencil	*per cycle*	*Complexity*	*per Cycle*	*Times*
5-pt	**0.054**	**2.21**	**0.29**	**1.63**
5-pt skew	**0.067**	**2.12**	**0.27**	**1.52**
9-pt (-1,8)	**0.078**	**1.30**	**0.26**	**1.83**
9-pt (-1,-4,20)	**0.109**	**1.30**	**0.26**	**1.83**

- Anisotropic Laplacian: $-\varepsilon U_{xx} - U_{yy}$

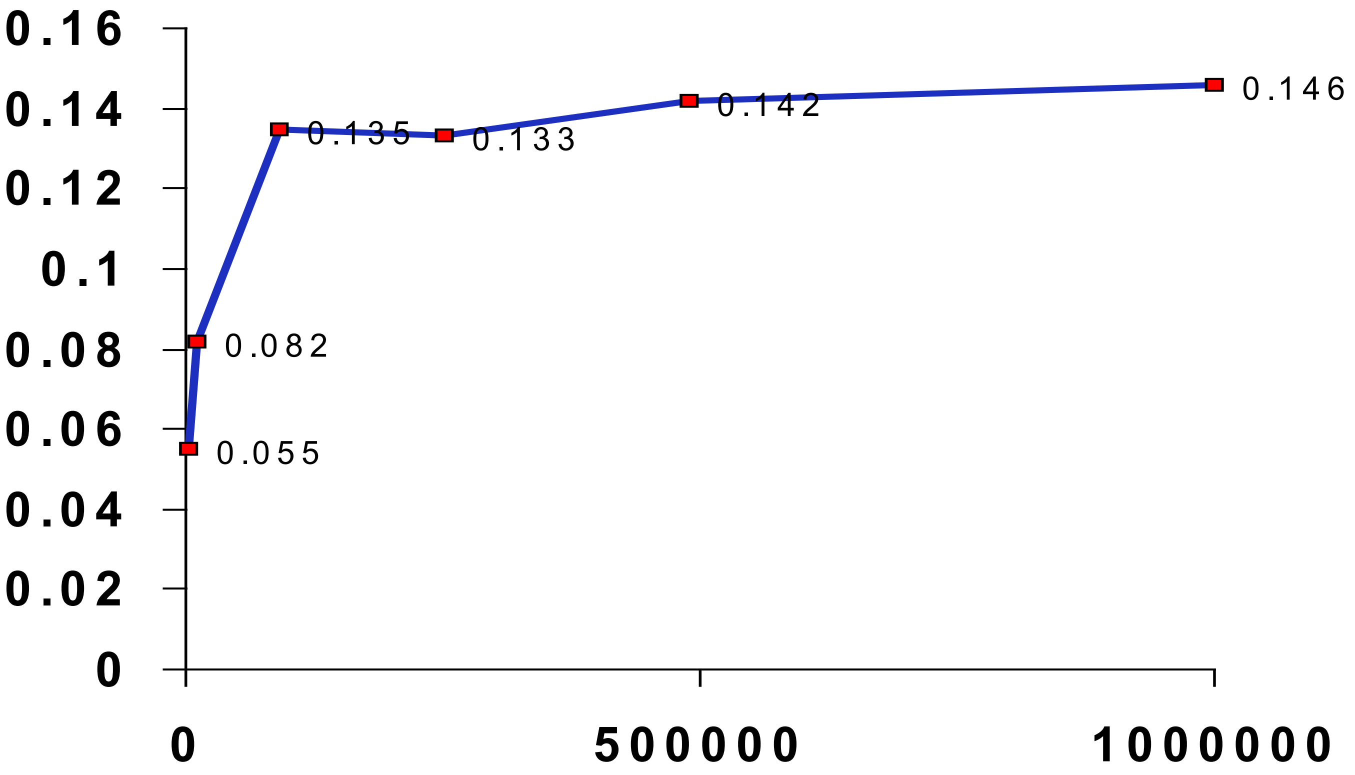
Epsilon	*0.001*	*0.01*	*0.1*	*0.5*	*1*	*2*	*10*	*100*	*1000*
Convergence/cycle	*0.084*	*0.093*	*0.058*	*0.069*	*0.056*	*0.079*	*0.087*	*0.093*	*0.083*

How's it perform (vol II)?

Structured Meshes, Rectangular Domains

- 5-point Laplacian on regular rectangular grids

Convergence factor (y-axis) plotted against number of nodes (x-axis)

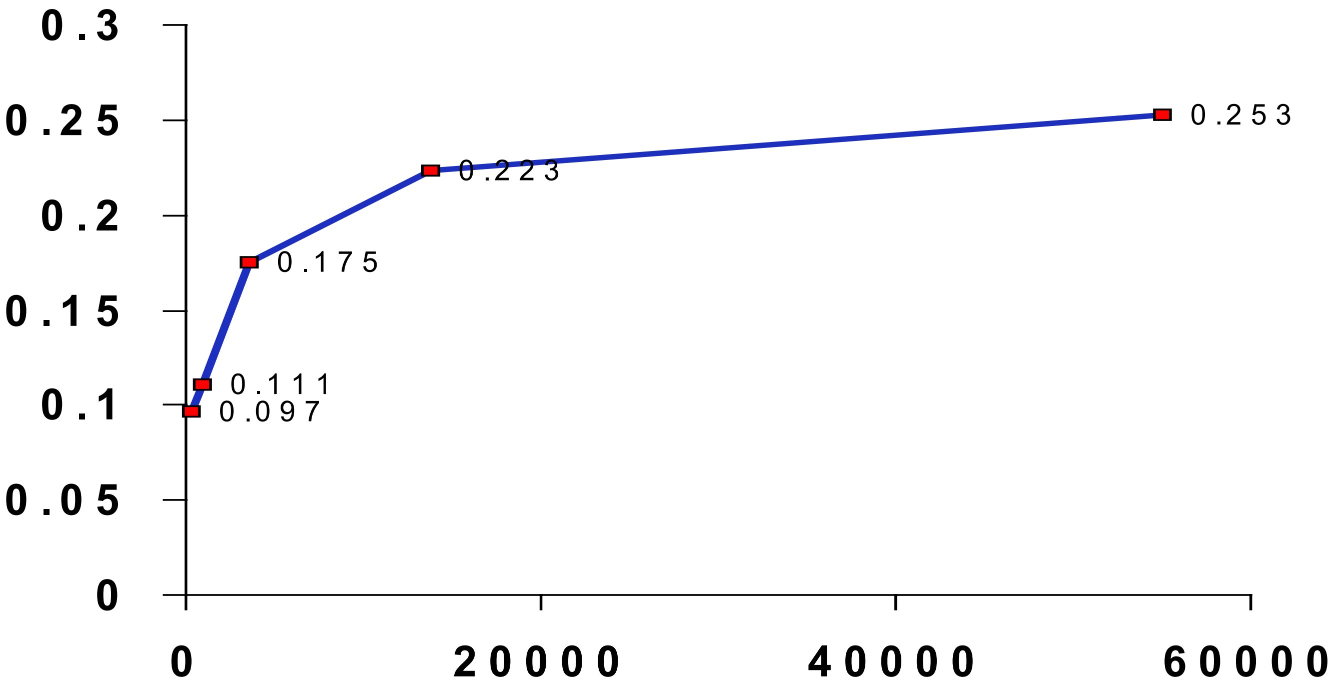


How's it perform (vol III)?

Unstructured Meshes, Rectangular Domains

- Laplacian on random unstructured grids (regular triangulations, 15-20% nodes randomly collapsed into neighboring nodes)

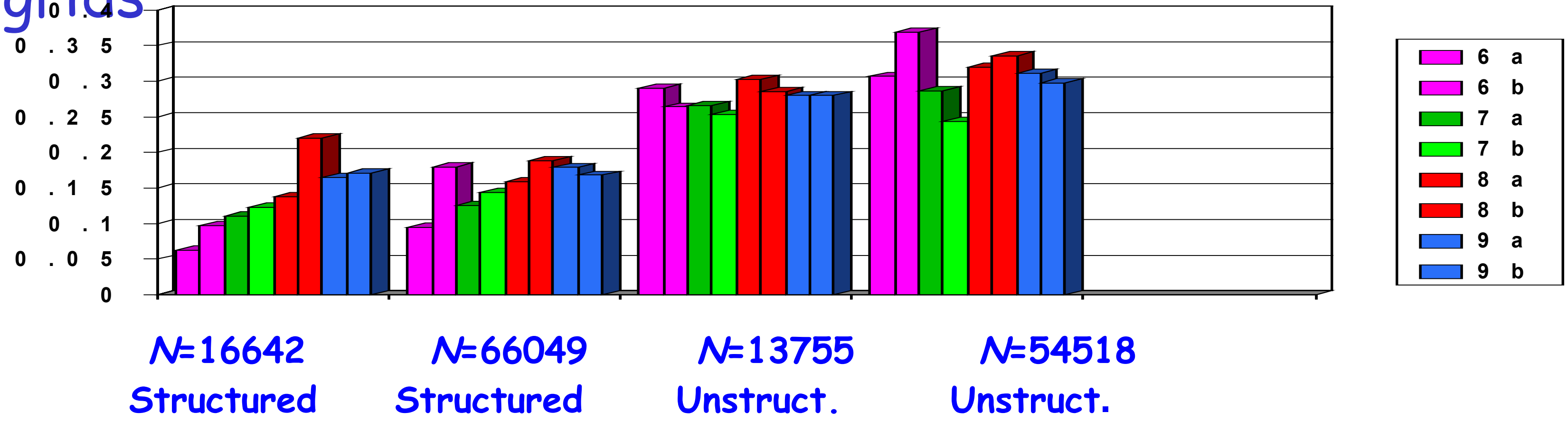
Convergence factor (y-axis) plotted against number of nodes (x-axis)



How's it perform (vol IV)?

Isotropic diffusion, Structured/Unstructured Grids

$\nabla \bullet (\; d(x,y) \; \nabla \; u)$ on structured, unstructured grids



Problems used: “a” means parameter *c=10*, “b” means *c=1000*

6: $d(x,y) = 1.0 + c\,|x-y|$

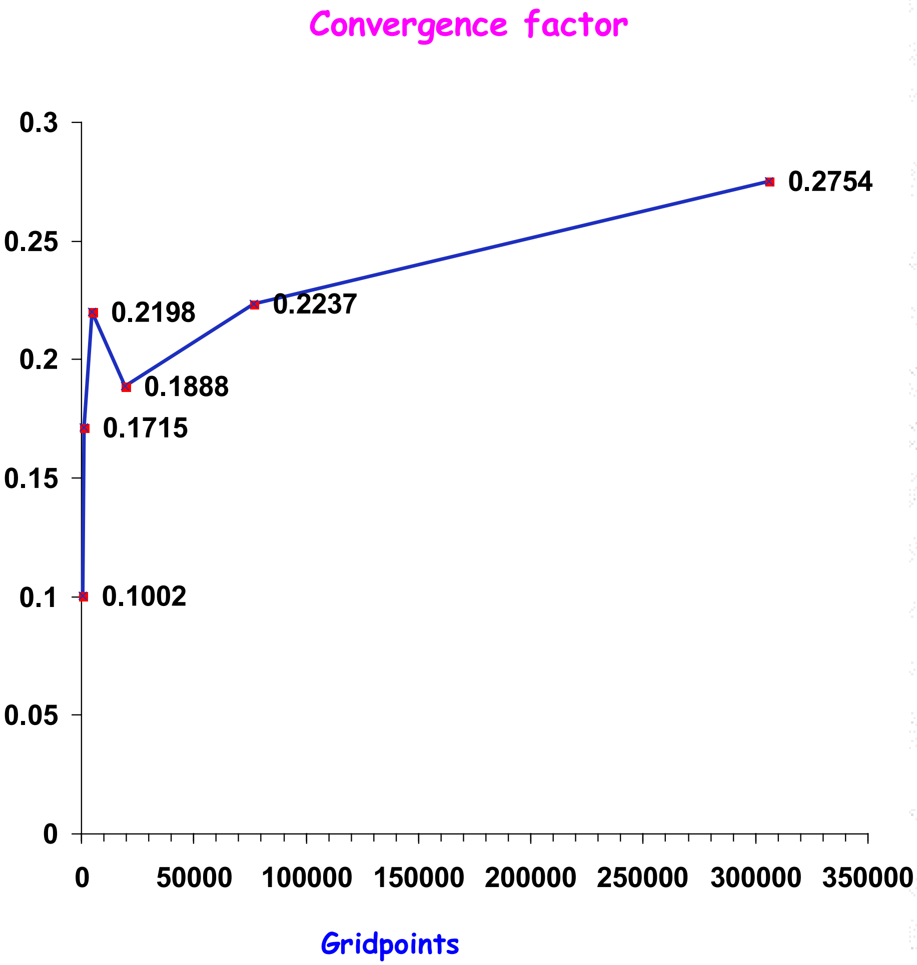
8: $d(x,y) = \begin{cases} 1.0 & 0.125 \le \max\{|x-0.5|, |y-0.5|\} \le 0.25 \\ c & \textbf{otherwise} \end{cases}$

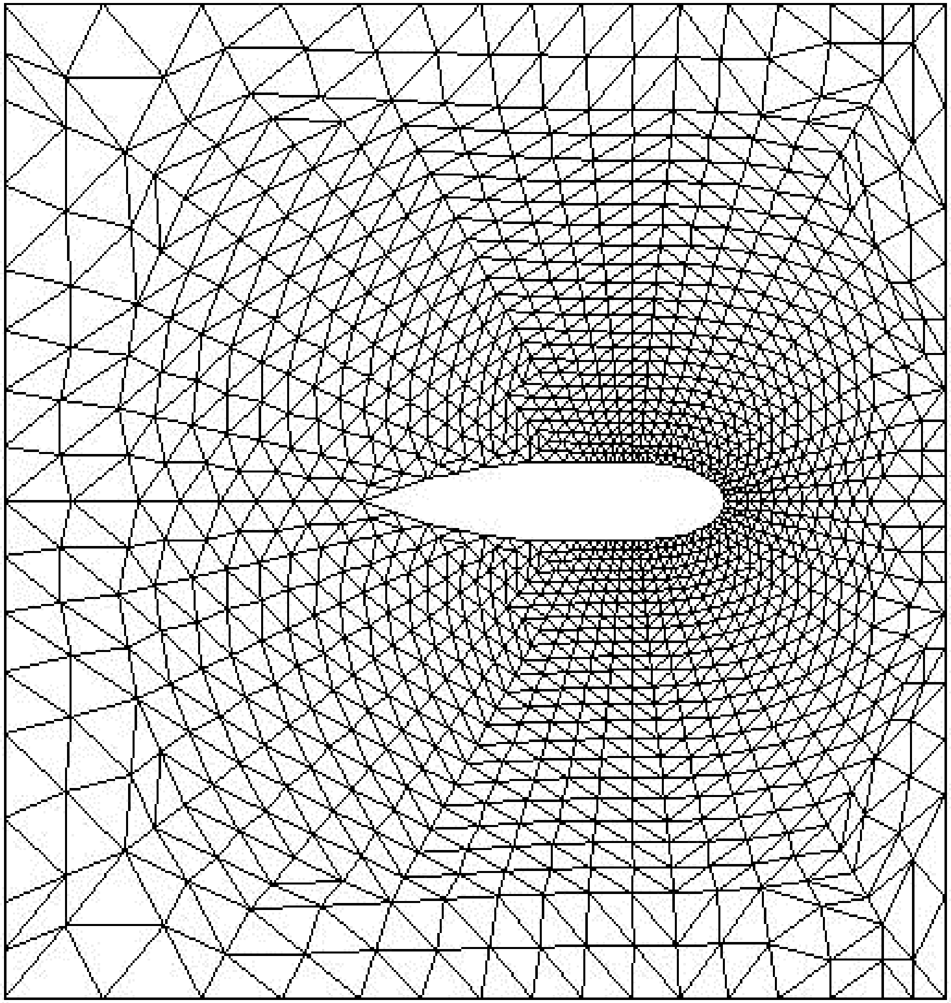
7: $d(x,y) = \begin{cases} 1.0 & x \le 0.5 \\ c & x > 0.5 \end{cases}$

9: $d(x,y) = \begin{cases} 1.0 & 0.125 \le \sqrt{(x-0.5)^2 + (y-0.5)^2} \le 0.25 \\ c & \textbf{otherwise} \end{cases}$

How's it perform (vol V)?

Laplacian operator, unstructured Grids





Outline

- ✓ Nonlinear Problems
- ✓ Neumann Boundary Conditions
- ✓ Anisotropic Problems
- ✓ Variable Mesh Problems
- ✓ Variable Coefficient Problems
- ✓ Algebraic Multigrid

Multigrid Rules!

- We conclude with a few observations:
  - *We have barely scratched the surface of the myriad ways that multigrid has been, and can be, employed.*
  - *With diligence and care, multigrid can be made to handle many types of complications in a robust, efficient manner.*
  - *Further extensions to multigrid methodology are being sought by many people working on many different problems.*