The state of algebraic multigrid research: where did we come from, where are we now, and where are we going?

presentation by

Van Emden Henson

Center for Applied Scientific Computing

Lawrence Livermore National Laboratory

Householder Symposium XV

Peebles, Scotland

June 19, 2002

This work was performed under the auspices of the U.S. Department of Energy by University of California Lawrence Livermore National Laboratory under contract No. W-7405-Eng-48. UCRL-PRES-146466

What this talk is

- What is AMG; who does it?
- Highlights of Multigrid- what makes MG work?
- The basic Pieces of AMG – what are the ingredients?
- The Assumptions- what assumptions (hidden and explicit) are common to most, if not all AMG.
- Creating algorithms: How are the Assumptions used to create algorithms (prolongation, coarse-grid selection)

Who is doing AMG?

- Many algorithms qualify as AMG methods. Some whose approaches are closely related to "classical AMG:"
  - Chang; Griebel, Neunhoeffer, Regler; Huang; Krechel, Stüben; Zaslavsky
- Work close to the original, but using different approaches to coarsening or interpolation:
  - Fuhrmann; Kickinger; Wittum, Wagner, Wieners
- Ideas that are important, novel, historical, or weird:

- Multigraph methods (Bank & Smith)
- Aggregation methods (Braess; Chan & Zikatanov & Xu)
- Smoothed Aggregation methods (Mandel, Brezina, Vanek)
- Black Box Multigrid (Dendy, Dendy & Bandy)
- Algebraic Multilevel Recursive Solver (Saad)
- Element based algebraic multigrid (Chartier; Cleary et al)
- Element-based aggregation AMG (Jones, Vassilevski)
- Element-free element-based methods (Henson, Kraus, Vassilevski)
- MultiCoarse correction with Suboptimal Operators (Sokol)
- Multilevel block ILU methods (Jang & Saad; Bank & Smith & Wagner; Reusken)
- AMG based on Element Agglomeration (Jones & Vassilevski)
- Sparse Approximate Inverse Smoothers (Tang & Wan)
- Algebraic Schur-Complement approaches (Axelsson & Vassilevski)
- Bootstrap AMG; compatible relaxation (Brandt, Yavneh)

Where did we come from?

Multigrid

Highlights of Multigrid:
The 1-d Model Problem

- **Poisson's equation:** $-\Delta u = f$ **in [0,1], with boundary conditions** $u(0) = u(1) = 0$.
- **Discretized as:**

$$\frac{-u_{i-1} + 2u_i - u_{i+1}}{h^2} = f_i \qquad u_0 = u_N = 0$$

- **Leads to the Matrix equation** $Au = f$, **where**

$$A = \frac{1}{h^2}\begin{pmatrix} 2 & -1 & & & & \\ -1 & 2 & -1 & & & \\ & -1 & 2 & -1 & & \\ & & & & & \\ & & & -1 & 2 & -1 \\ & & & & -1 & 2 \end{pmatrix}, \quad u = \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ \\ u_{N-2} \\ u_{N-1} \end{pmatrix}, \quad f = \begin{pmatrix} f_1 \\ f_2 \\ f_3 \\ \\ f_{N-2} \\ f_{N-1} \end{pmatrix}$$

Highlights of Multigrid: Weighted Jacobi Relaxation

- Consider the iteration:

$$u_i^{(new)} \leftarrow (1-\omega)\, u_i^{(old)} + \frac{\omega}{2h^2}\left(u_{i-1}^{(old)} + u_{i+1}^{(old)} + f_i \right)$$

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

$$u^{(new)} = \left[(1-\omega) I - \omega D^{-1}(L+U) \right] u^{(old)} + \omega D^{-1} f$$

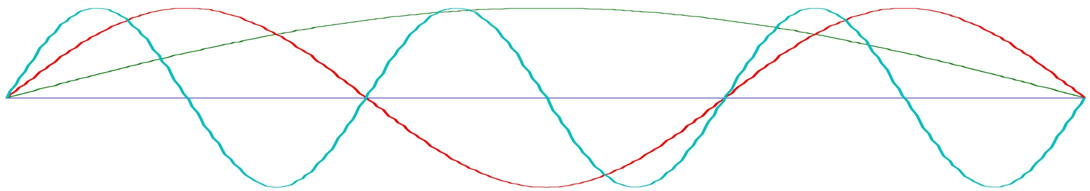
$$= G_\omega u^{(old)} + \omega D^{-1} f \quad .$$

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

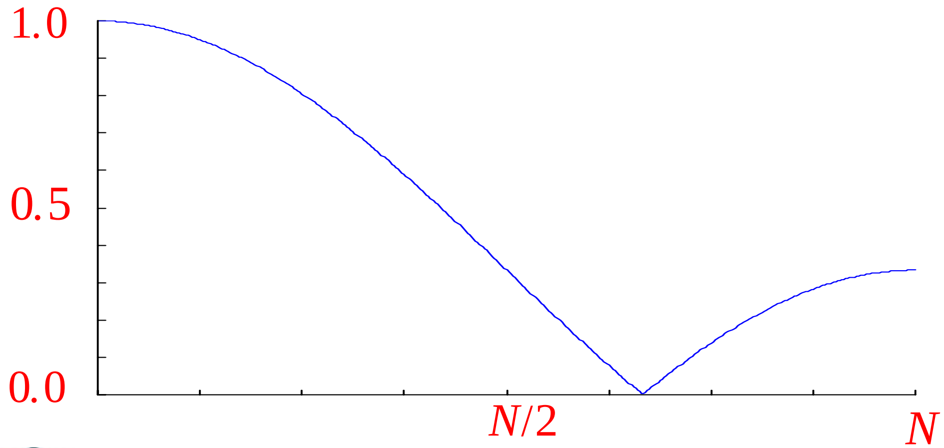
$$e^{(new)} = G_\omega e^{(old)}$$

Highlights of Multigrid:
Relaxation Typically Stalls

- The eigenvectors of G_ω are the same as those of A, the Fourier Modes: $v_i = \sin(ik\pi/N)$, $k = 1, 2, \cdots, N-1$



- The eigenvalues of G_ω are $1 - 2\omega\sin^2(k\pi/2N)$, so the effect of relaxation on the modes is:

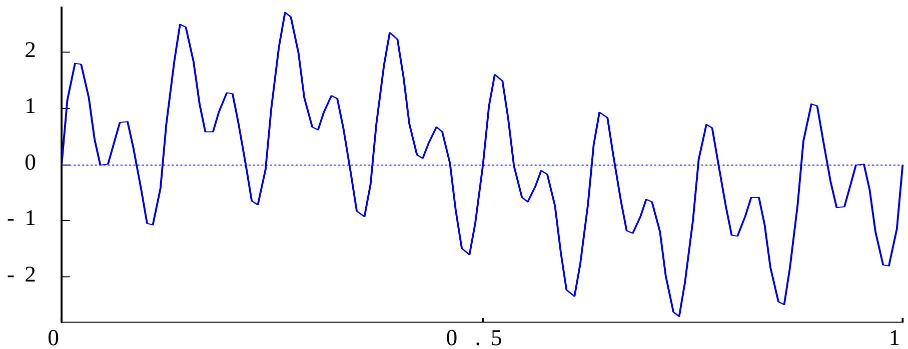


$|\lambda_k|$ for $\omega = \frac{2}{3}$.

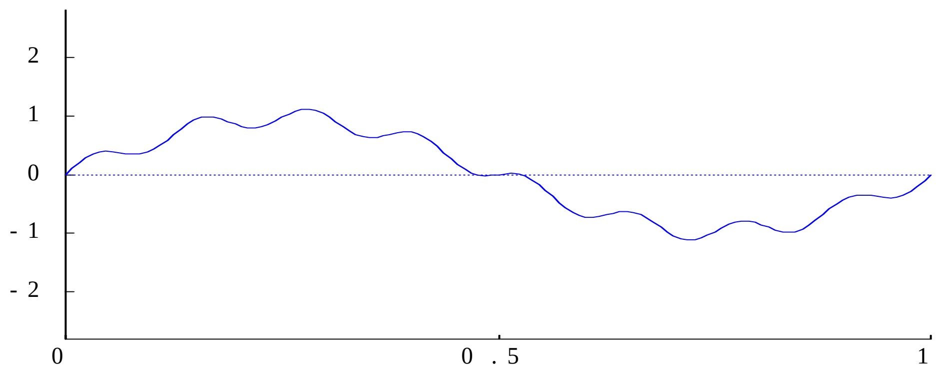
No value of ω attenuates the lowest modes

Highlights of Multigrid:
Relaxation Smooths the Error

- Initial error,



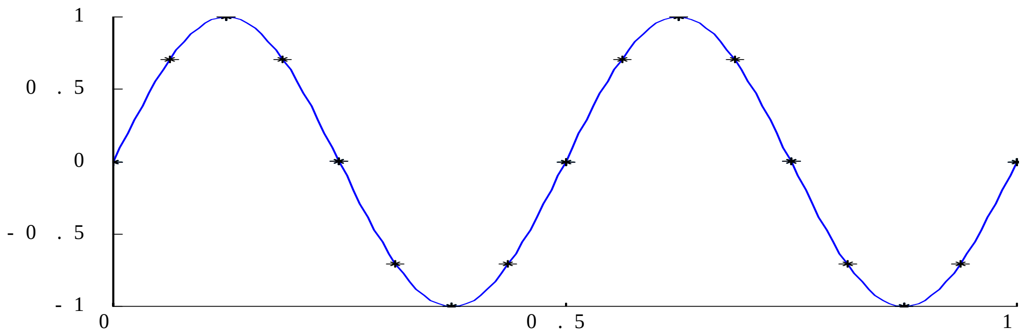
- Error after several iteration sweeps:



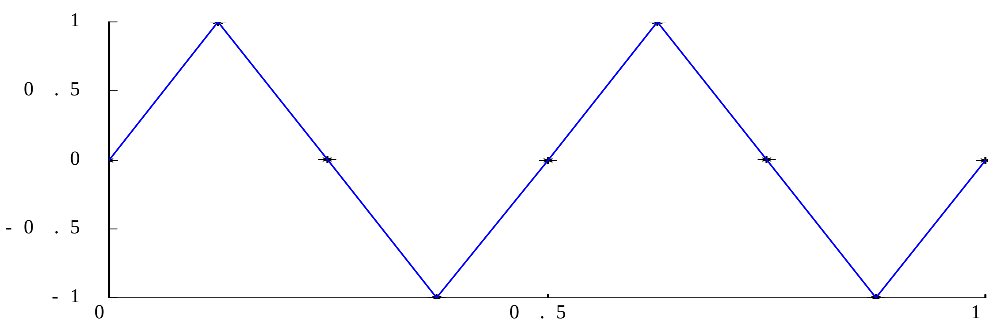
*Many relaxation schemes have the **smoothing property**, where oscillatory modes of the error are eliminated effectively, but smooth modes are damped very slowly.*

Highlights of Multigrid: Smooth error can be represented on a coarse grid

- 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!!

Highlights of Multigrid: What tools are required?

- **Interpolation and restriction operators:**

$$I_{2h}^{h} = \begin{pmatrix} 0.5 & & \\ 1.0 & & \\ 0.5 & 0.5 & \\ & 1.0 & \\ & 0.5 & 0.5 \\ & & 1.0 \\ & & 0.5 \end{pmatrix}, \qquad I_{h}^{2h} = \begin{pmatrix} 0 & 1 & 0 & & & & \\ & & 0 & 1 & 0 & & \\ & & & & 0 & 1 & 0 \end{pmatrix}, \qquad I_{h}^{2h} = \begin{pmatrix} 0.25 & 1.0 & 0.25 & & & & \\ & & 0.25 & 1.0 & 0.25 & & \\ & & & & 0.25 & 1.0 & 0.25 \end{pmatrix}$$

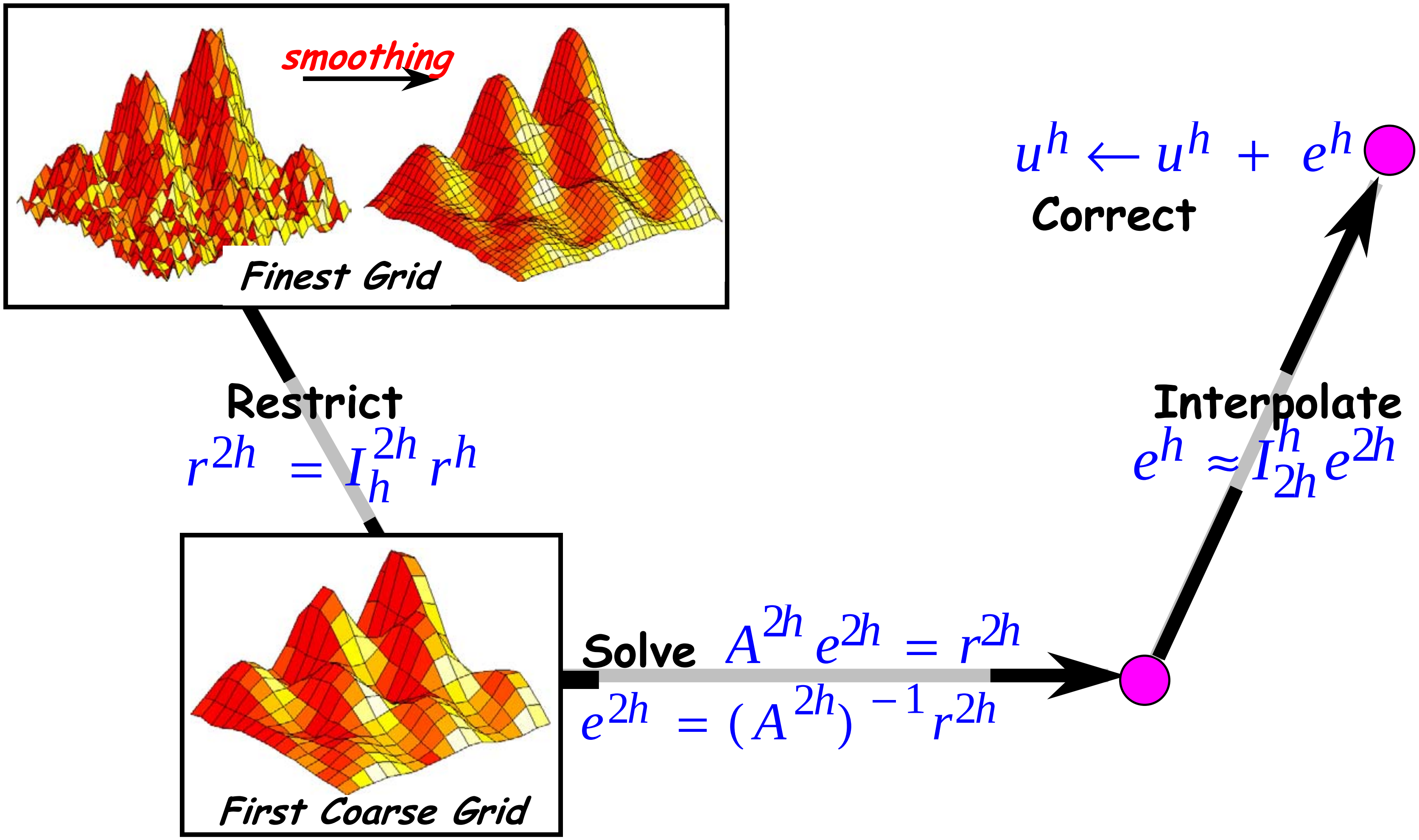
Linear Interp. | Injection | Full-weighting

- **Coarse-grid Operator A^{2h}. Two methods:**
  (1) Discretize equation at larger spacing
  (2) Use Galerkin Formula:

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

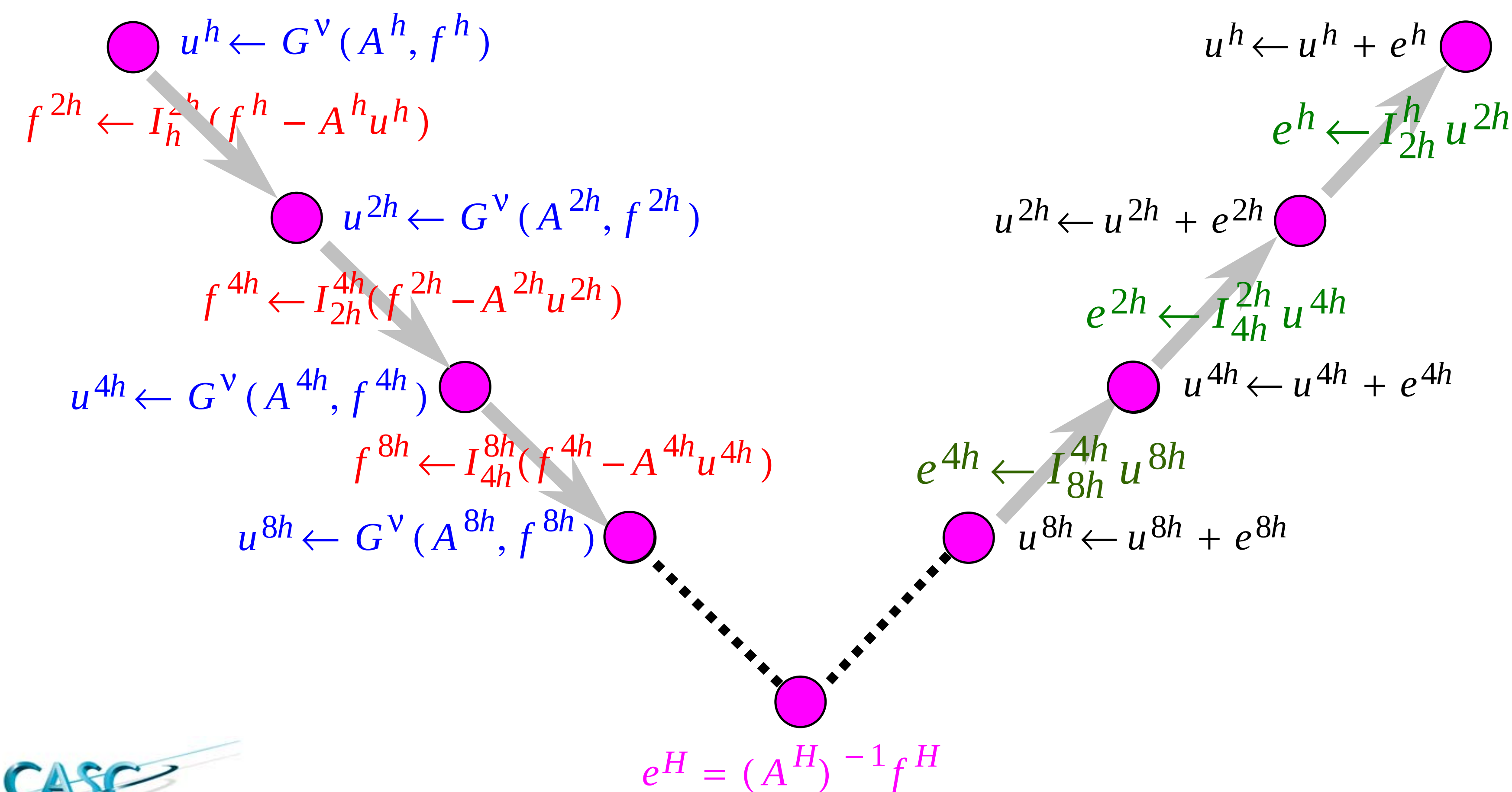
CASC

Highlights of Multigrid:
The coarse-grid correction



Highlights of Multigrid:
Recursion: the (ν,0) V-cycle

- Major question: How do we "solve" the coarse-grid residual equation? *Answer: recursion!*

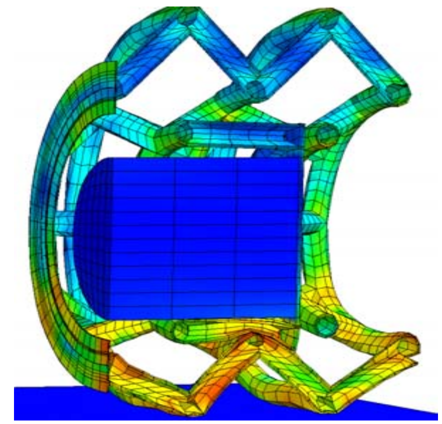


The goals we strive for in AMG

- Use algebraic nature of the problem to define MG components.
- In the most general case, use the matrix only.
- $O(N)$ setup & cycle time.
- "Typical" MG efficiency (for comparable problems).

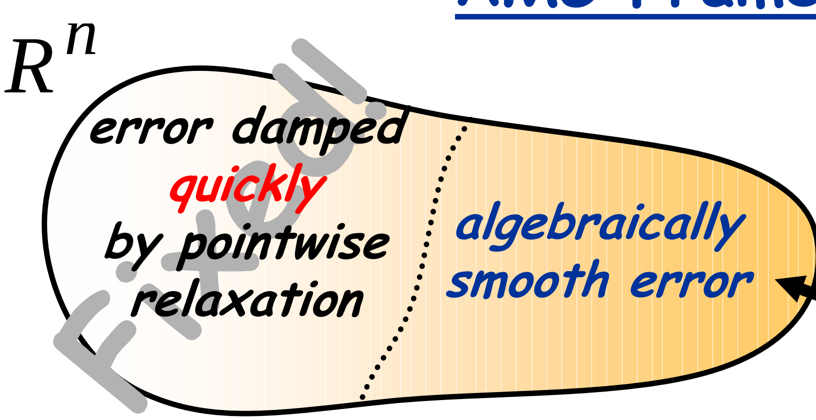
What are the Pieces? The basics of an AMG algorithm

- Standard AMG only uses matrix info
- AMG automatically coarsens "grids"



DYNA3D

AMG Framework



In AMG we DEFINE smooth error:
Smooth error is that error which is slow to converge under relaxation.

Choose coarse grids, transfer operators, etc. to eliminate

Accurate characterization of smooth error is crucial

CASC

There are numerous choices to be made

- Relaxation
  — Jacobi, Gauss-Seidel, block, etc
- Coarse-grid selection (pointwise, aggregation, agglomeration, graph theoretic)
- Interpolation operator P
  — generally depends on concept of "smoothness"
- Restriction operator R
  — most commonly $R = P^T$
- Coarse grid operator A^{k+1}
  — generally Galerkin
- Solution cycle
  — V, W, F, slash, etc

But sometimes, smooth error isn't! (smooth, that is)

- Consider the problem

$$-(a\,u_x)_x - (b\,u_y)_y + c\,u_{xy} = f(x,y)$$

- on the unit square, using a regular Cartesian grid, with finite difference stencils and values for *a*,*b*,and *c*:

a=1 b=1000 c=0	a=1 b=1 c=2
a=1 b=1 c=0	a=1000 b=1 c=0

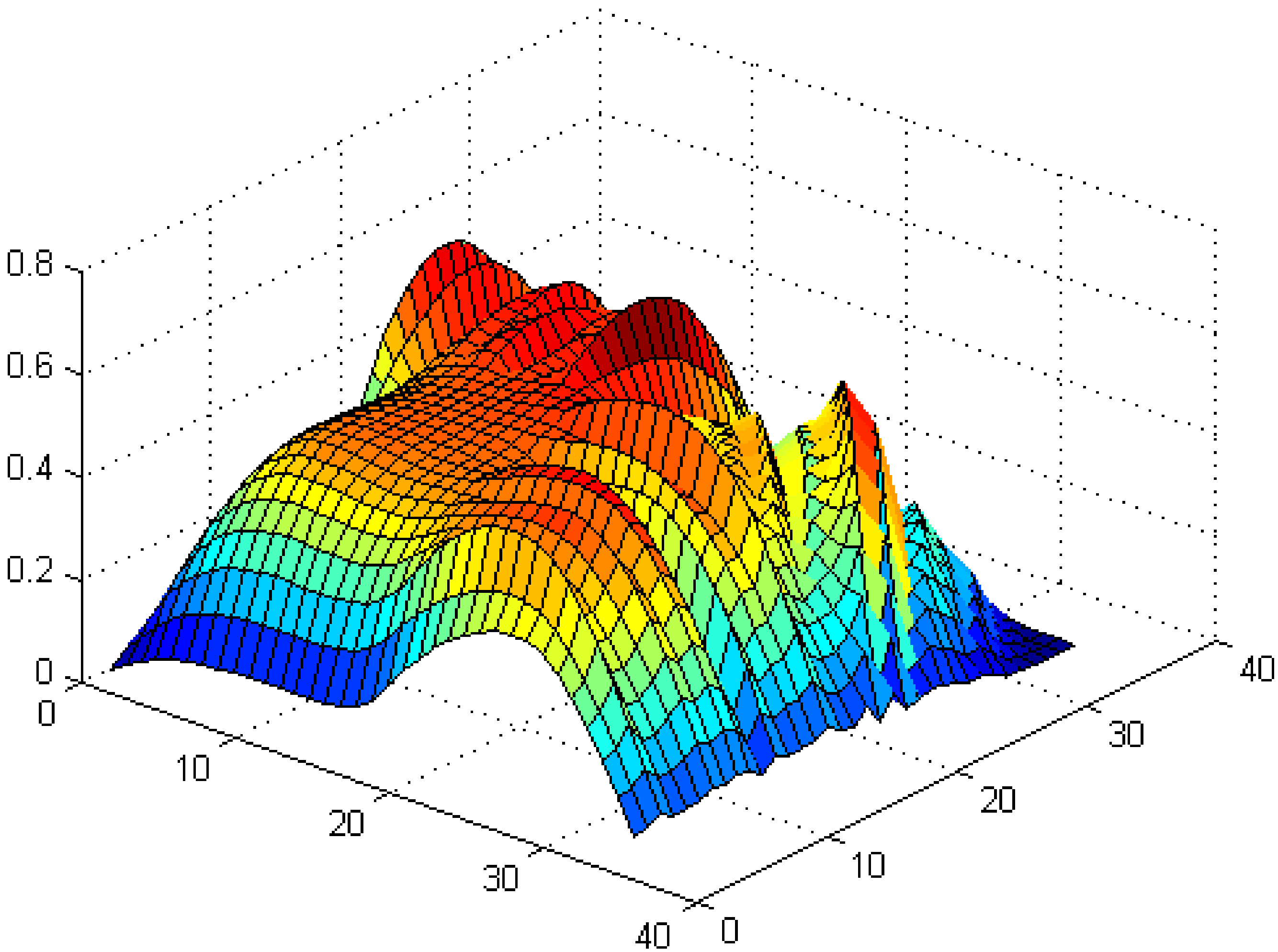
$$u_{xx} = h^{-2}[\,1 \;\; -2 \;\; 1\,]$$

$$u_{yy} = \frac{1}{h^2}\begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}$$

$$u_{xy} = \frac{1}{2h^2}\begin{bmatrix} -1 & 1 & \\ 1 & -2 & 1 \\ & 1 & -1 \end{bmatrix}$$

Smooth error for

$$-(a\,u_x)_x - (b\,u_y)_y + c\,u_{xy} = f(x,y)$$



The Assumptions (often hidden) common to most, if not all, AMG methods

- **In almost any algebraic method, certain assumptions are made regarding nature of "smooth" error.**

- **These assumptions are then used to guide the coarse-grid selection, and to define the prolongation, restriction, and coarse-grid operators**

- The AMG Holy Grail: what is smoothness?

CASC

The Assumptions: characterizing smooth error

- Small residual: $Ae \approx 0$ or $\sum_{i=1}^{N} a_{ij} e_j \approx 0$
- Small energy: $\langle Au - 2f, u \rangle \approx 0$ or $\langle Ae, e \rangle \approx 0$
- Eigenvectors corresponding to small eigenvalues of the operator matrix
- Element-based approaches (low energy modes of local matrices)
- Relaxation-driven: $\sum A_{ij} \vec{e}_j \approx \vec{0}$

The Assumptions: philosophies of prolongation

- The columns of the prolongation operator *P* span the space of "smooth" functions
- The rows of *P* correspond to fine-grid dofs (i.e., what nearby *C*-points contribute, in what proportion, to this *F*-point?)
- The columns of *P* correspond to coarse-grid dofs (i.e., what contribution does this *C*-point make to which F-points?)
- Methods of determining *P* may be either row-based (e.g., Ruge-Stüben, *AMGe*) or column based (e.g., smoothed aggregation, ρ*AMGe*). Which orientation is used depends on the underlying smoothness assumption!

Where did we come from?

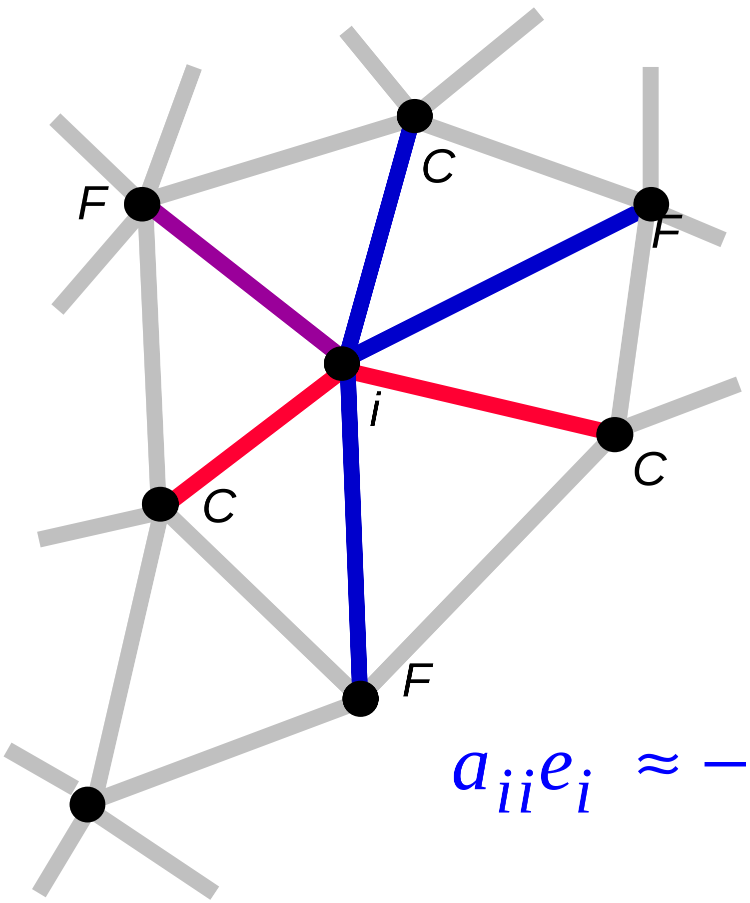
Classical AMG

The Assumptions: characterizing smooth error by $Ae \approx 0$

- **M-Matrices**: Poisson on unstructured grid.
- For most iterations (e.g., Jacobi or Gauss-Seidel) slow convergence holds if $Ae \approx 0$.
- Hence $\sum a_{ij} e_j \approx 0$ implying that $e_i \approx \frac{1}{a_{ii}} \sum_{i \neq j} a_{ij} e_j$
- An implication is that, if e is an error slow to converge, then locally at least, e_i can be well-approximated by an average of its neighbors.
- Another implication is that smooth error varies slowly in the direction of dependence.

CASC

Prolongation based on smooth error, variable inter-dependence



Sets:

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

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

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

$$Ae \approx 0$$

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

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

Prolongation weights defined by collapsing *i*-to-*F* connections

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

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

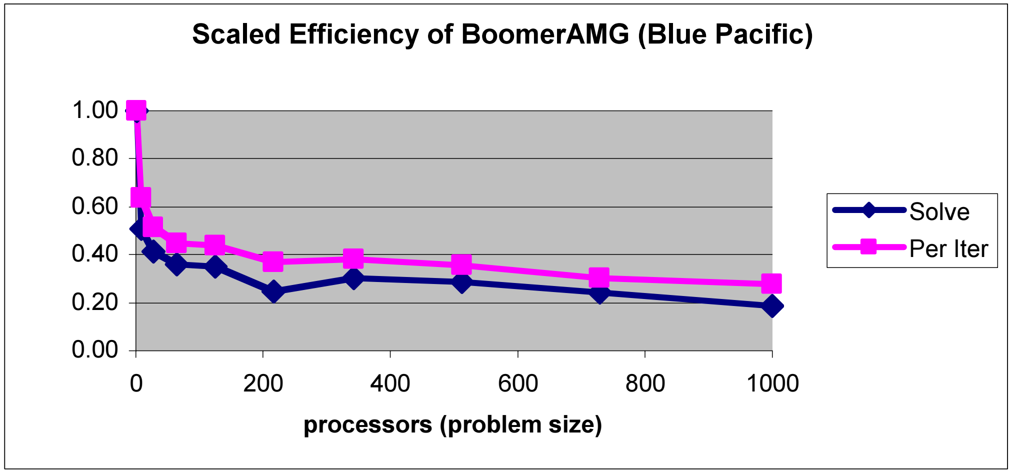
yielding the prolongation weights:

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

Classical AMG algorithm works remarkably well for many problems

- Very effective on scalar problems & some systems.

- Research on parallel coarsening algorithms led to *BoomerAMG*, a parallel AMG code.



Where are we now?

AMGe

Good local characterizations of smooth error is key to robust AMG

- **Traditional AMG uses the following heuristic, based on properties of M-matrices:** smooth error varies slowest in the direction of "large" coefficients.

$$A = \begin{bmatrix} -1 & -4 & -1 \\ 2 & 8 & 2 \\ -1 & -4 & -1 \end{bmatrix}$$

However:

Stretched quad example ($\Delta x \to \infty$):

Direction of strength not apparent.

Worse for systems.

The Assumptions: characterizing smooth error by $\langle Ae, e \rangle \approx 0$

- **Start with a global measure that relates interpolation accuracy and eigenmodes**

- **Fundamental heuristic for AMGe:** for a two grid algorithm, the interpolation operator must be able to reproduce a mode up to the same accuracy as the size of the associated eigenvalue.

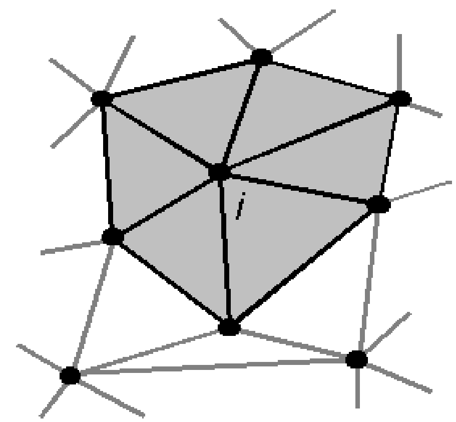
- **That is, the following AMGe measure should be small:**

$$M(Q,e) = \frac{\langle (I-Q)\,e,\,(I-Q)\,e \rangle}{\langle Ae,\, e \rangle};$$

where Q is injection followed by interpolation

AMGe uses elements to localize and approximate modes with error ≈ λ

Use local measure to construct
AMGe components:



$$M_i = \max_{e \neq 0} \frac{\left\langle \varepsilon_i^T (I - Q)\, e,\ \varepsilon_i^T (I - Q)\, e \right\rangle}{\left\langle A_i e, e \right\rangle}; \quad Q = P\begin{bmatrix} 0 & I \end{bmatrix}$$

Use local measure to define interpolation

- Interpolation is defined by the *arg min* of

$$\min_{q_i \in Z_i} M_i(q_i)$$

where we restrict the structure of interpolation to "nearest neighbors" by

$$Z_i = \{v \in R^n : v_j = 0 \text{ for } j \in \Omega \setminus C_i\}$$

- This is easily computed in practice.

Using local measure to define interpolation ⇔ fitting local eigenmodes

- **Assume the eigen-decomposition:**

$$A_i V_i = V_i \Lambda_i; \quad V_i = \left[V_{i0} \;\; V_{i+} \right]; \quad \Lambda_i = \begin{bmatrix} 0 & 0 \\ 0 & \Lambda_{i+} \end{bmatrix};$$

- **Finding the *arg min* is equivalent to solving the following constrained least-squares problem**

$$\min_{q_i} \left\| \Lambda_{i+}^{-1/2} V_{i+}^T (\varepsilon_i - q_i) \right\|^2, \quad \text{subject to} \quad V_{i0}^T (\varepsilon_i - q_i) = 0$$

Computing interpolation in practice

- Partition local matrix by *F* & *C*-pts:

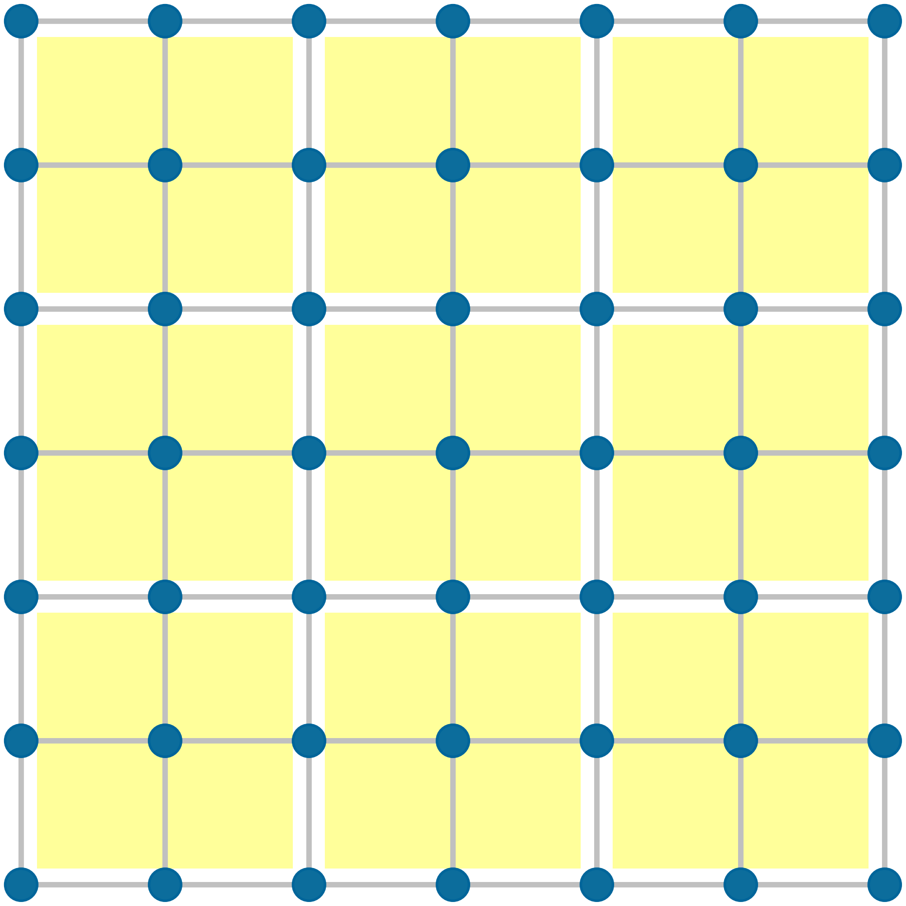
$$A_i = \begin{bmatrix} A_{ff} & A_{fc} \\ A_{cf} & A_{cc} \end{bmatrix}$$

- Interpolation to point *i* is defined by

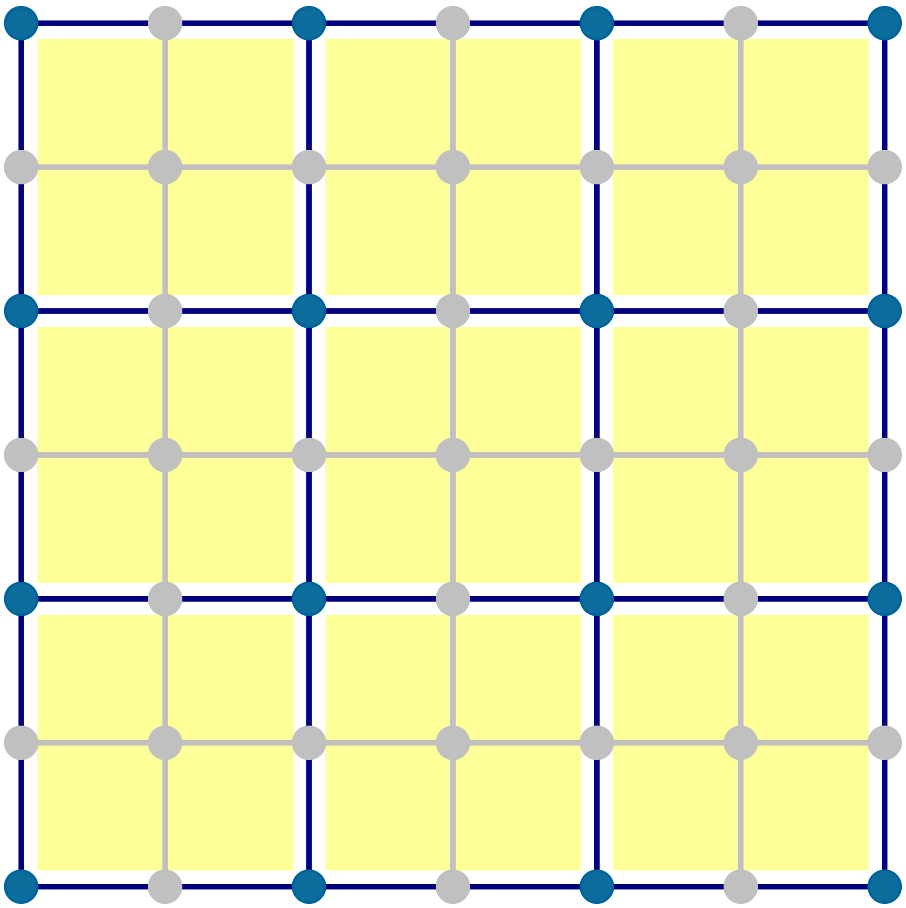
$$q_i = \begin{bmatrix} 0 \\ -A_{cf} A_{ff}^{-1} \varepsilon_i \end{bmatrix}$$

- Perfect interpolation of the local problem.

Agglomeration coarsening

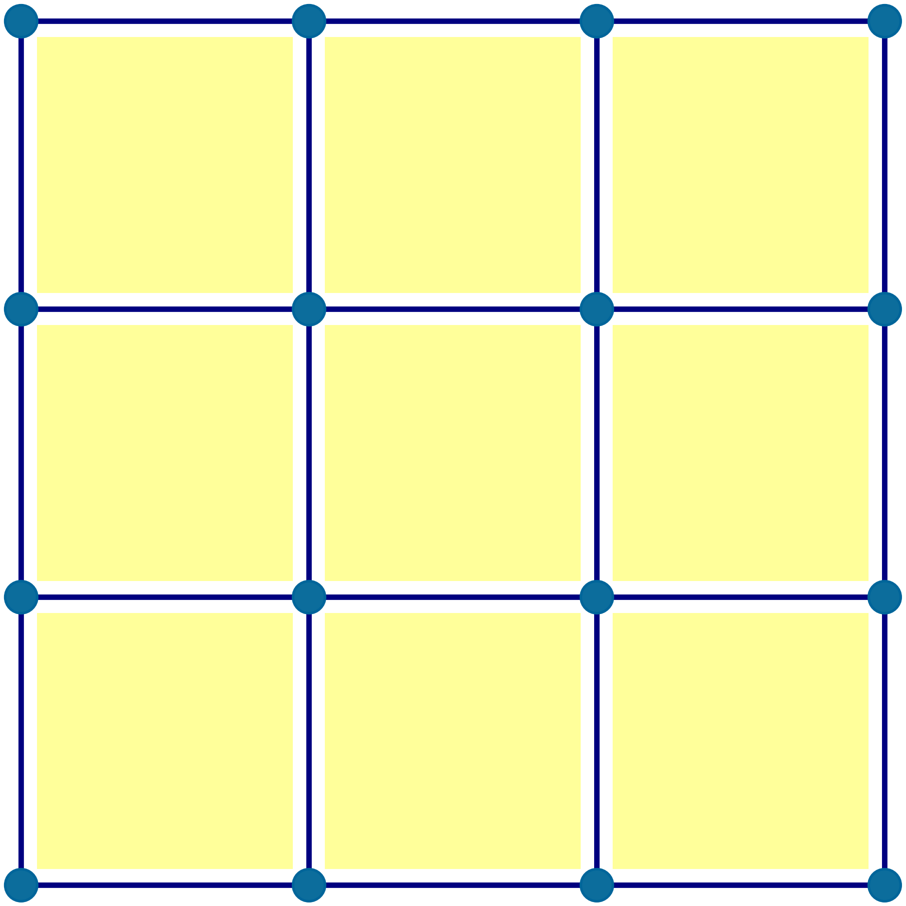


Agglomeration coarsening



- Agglomerate by growing groups of elements using graph & measures.
- Define faces by intersecting elements
- Define vertices by intersecting faces

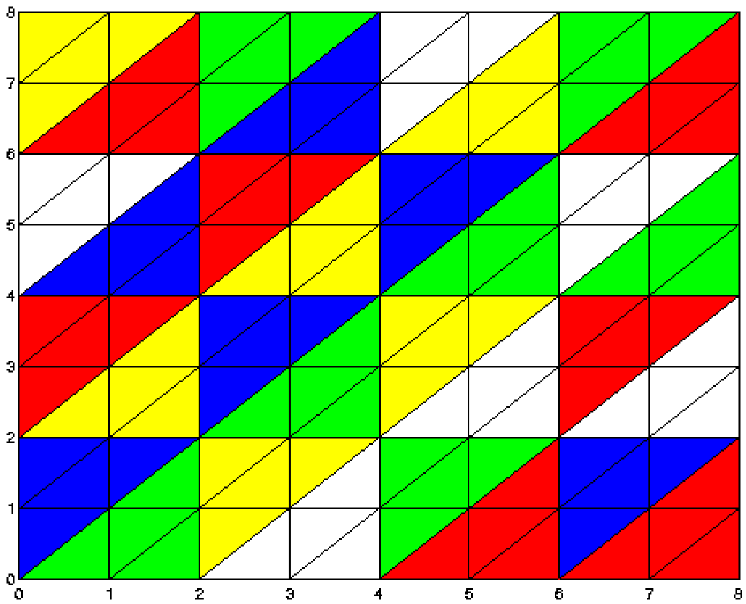
Agglomeration coarsening

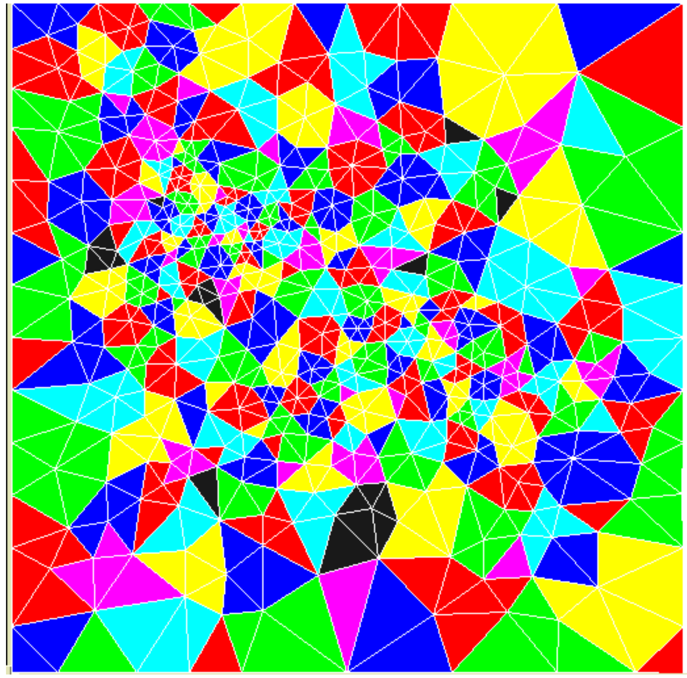


- Let the vertices be the C-points
- Construct coarse elements & stiffness matrices

$$P^T\left(\sum_{\alpha\in E} A_\alpha\right)P$$

Agglomerations for triangular elements, both structured & unstructured



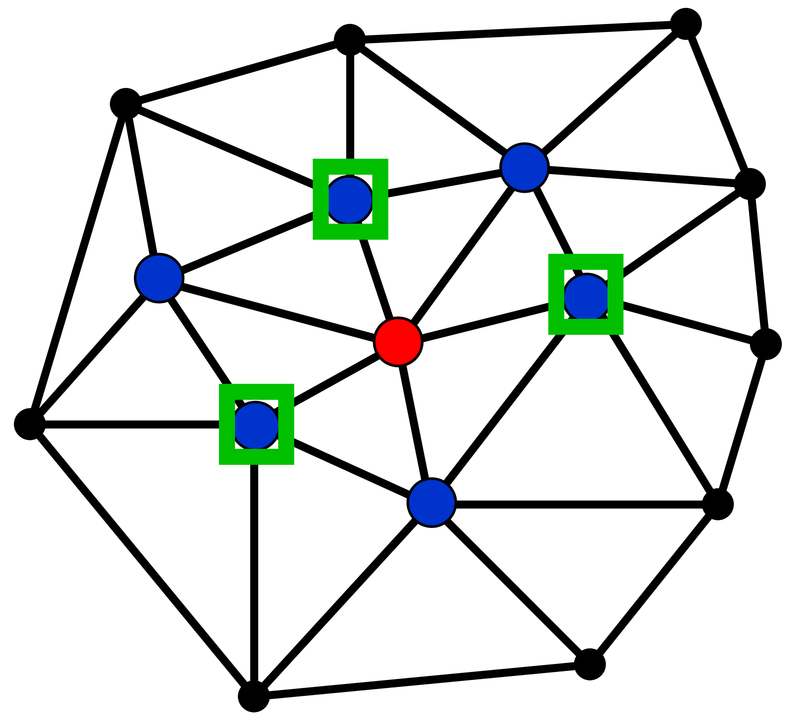


Where are we now?

Element-Free AMGe

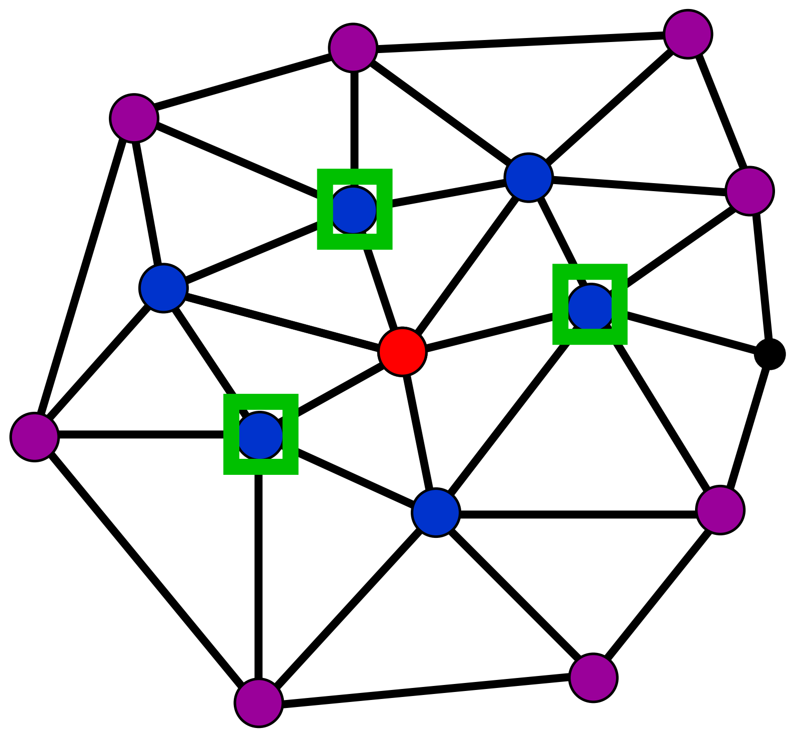
The Assumptions: smooth error given by low energy modes of local matrices

- Let i be the f-point to which we wish to interpolate
- $\Omega(i)$ is the set of points in the neighborhood of i
- $\Omega_C(i)$ is the set of coarse nearest neighbors of i



The Assumptions: smooth error given by low energy modes of local matrices

- Define $\Omega_X(i)$, the set of "exterior" points for the neighborhood of i: the set of points j such that j is connected to a fine point in the neighborhood of i



$$\Omega_X(i) = \{ j \not\subset \Omega(i) : a_{jk} \neq 0, \ j \in \Omega(i) \setminus \Omega_C(i) \}$$

Prolongation in Element-free AMGe: based on extensions

- We use the following window of the matrix *A*

$$\begin{bmatrix} A_{ff} & A_{fc} & A_{fX} & 0 \\ * & * & * & * \\ A_{Xf} & A_{Xc} & A_{XX} & * \\ * & * & * & * \end{bmatrix} \begin{array}{l} \Omega(i) \setminus \Omega_c(i) \\ \Omega_c(i) \\ \Omega_X(i) \\ \text{everything else on grid} \end{array}$$

where we will only be interested in the blocks shown.

Prolongation in Element-free AMGe: based on extensions

- Assume that an extension mapping is available:

$$E = \begin{bmatrix} I & 0 \\ 0 & I \\ E_{Xf} & E_{Xc} \end{bmatrix}$$

i.e., we interpolate the exterior dofs ("X") from the interior dofs f and c, by the rule

$$v_X = E_{Xf} v_f + E_{Xc} v_c$$

The Assumptions: smooth error from low energy modes of local A_i; no elements!

- We construct the prolongation operator on the basis of the modified local matrix

$$\left[\widehat{A}_{ff}\ ,\ \widehat{A}_{fc}\right] = \left[A_{ff}\,, A_{fc}\,, A_{fX}\right]\begin{bmatrix} I & 0 \\ 0 & I \\ E_{Xf} & E_{Xc} \end{bmatrix}$$

- Then the *i*th row of the prolongation matrix *P* is taken as the *i*th row of the matrix:

$$-\left(\widehat{A}_{ff}^{-1}\,\widehat{A}_{fc}\right)$$

Where are we going?

Compatible Relaxation (CR)

CR & AMGe: Measuring coarse-grid quality

- Assume we are given a coarse grid. Then, the following measures the ability of the coarse grid to represent algebraically smooth error:

$$M_C = \min_Q \max_{e \neq 0} M(Q, e)$$

- We have that

$$W = -A_{ff}^{-1} A_{fc}; \qquad M_C = \frac{1}{\lambda_{\min}(A_{ff})}$$

Using CR: How good are the C points?

- Relax on $A_{ff} x_f = 0$. (Compatible relaxation)

- If CR is slow to converge, either increase the coarse-grid size or do more relaxation in the multigrid cycle.

- We have shown that compatible relaxation is fast to converge if and only if the AMGe measure is small.

Using CR: Defining the Coarse Variables

- To check convergence of CR, relax on the equation

$$A_{ff}\, x = 0$$

& monitor pointwise convergence to *0*.

- CR coarsening algorithm:

Initialize $U = \Omega;\ C = \emptyset;\ F = \Omega - C$

While $U \neq \emptyset$

Do ν compatible relaxation sweeps

$U = \{ i : x_i^{\nu} / x_i^{\nu-1} > \theta \}$

$C = C \cup \{\text{independent set of } U\};\ F = \Omega - C$

Using CR: Defining Interpolation

- The *arg min* of the AMGe measure yields

$$P = \begin{bmatrix} W \\ I \end{bmatrix}$$

  where $A_{ff} W = -A_{fc}$.

- If CR is fast to converge, then one might use instead a few sweeps of relaxation with $W_0 = 0$. Yavneh does something similar to this.

- AMGr & Multigraph use $W = -D_1^{-1} A_{fc}$.

Spectral AMGe

Where are we going?

ρAMGe

- **Consider (as before) the measure function**

$$M(Q,e) = \frac{\langle (I-Q)e, (I-Q)e \rangle}{\langle Ae, e \rangle};$$

  & define the new measures

$$M_1 = \min_{Q_1} \max_{e \neq 0} M(Q_1, e); \quad Q_1 = P(P^T P)^{-1} P^T$$

$$M_2 = \min_{Q_2} \max_{e \neq 0} M(Q_2, e); \quad Q_2 = P(P^T A P)^{-1} P^T A$$

- **It is easy to show that** $M_2 = M_1 \leq M_c$ **.**
- **Let** p_i **be the ordered orthonormal eigenvectors of** A **.**
- **Then the *arg min* of both measures is** $P = [p_1, \ldots, p_c]$ **with measure**

$$\frac{1}{\lambda_{c+1}}$$

ρAMGe: Take patched local eigenvectors as the interpolation basis!

- As with AMGe, we use elements to localize the problem of determining & matching smooth error.
- Coarse dofs are no longer subsets of fine dofs: coefficients of local eigenvectors become the coarse-grid dofs.
- Local eigenvectors are "patched" together to form columns of global prolongation operator
- Currently expensive, but potentially very robust.

Adaptive AMGe

Where are we going?

Adaptive AMGe: goals

- We wish to apply AMG to "more difficult" problems (systems, elasticity, slide surfaces, etc.)
- We wish to develop an AMG solver with increased robustness while not sacrificing optimality.
- We wish to develop a solver that defaults to simple algorithms when presented with simple problems.

Adaptive or Bootstrap or Calibration or Prerelaxation or Feedback AMG

- Test your AMG on a problem whose solution you know: $Ax = 0$.
- If it works after a few cycles, stop.
- Else, x is a good bad guy: it's an algebraically smooth error in the sense that AMG cannot quickly reduce it.
- Now adjust the coarse grid and interpolation so that it matches x well. The trick is to do this locally & to continue it on coarser levels.

AMG algorithms can be classified by their characterization of "smooth error"

Small residual	Small energy	Comp. Relaxation	Small eigenvalue
$Ae \approx 0$	$<Ae, e> \approx 0$	on $A_{ff}x_f = 0$.	$Ae = \lambda e;\ \lambda \approx 0$
• Ruge-Stüben • classical AMG • original *BoomerAMG* • mature algorithms	• AMGe • element-free AMGe • recent developments • not yet parallel	• CRAMG • bootstrap AMG • αAMGe • not yet implemented	• spectral AMGe • most recent • implemented in test code
• fast, less memory • low complexity • solves many problems • less robust; fails on difficult, complicated problems	• slower, more memory • higher complexity • solves more problems • more robust; works on more difficult problems, but not all	• slower, more memory • may be high complexity • solves more problems • more robust; should work on more difficult problems, even most • adaptivity can become a very powerful feature	• memory intensive • higher complexity • solves most problems • most robust AMG method known

CASC

AMG Rules!

- Interest in AMG methods is high, and rising, because of the increasing importance of tera-scale simulations on unstructured grids.

- Diverse AMG methods are derived from a very few fundamental assumptions; in particular, assumptions about the nature of smooth error.

- AMG is evolving along a number of disparate lines, each based on some fundamental ideas tailored to address specific difficulties. They run a gamut from "cheap, fast, with limited applicability" to "very robust but expensive."