Lawrence Livermore National Laboratory

An Algebraic Multigrid Tutorial

IMA Tutorial – Fast Solution Techniques
November 28 - 29, 2010

Robert D. Falgout

Center for Applied Scientific Computing

This work performed under the auspices of the U.S. Department of Energy by
Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344

LLNL-PRES-411189

Outline

- Motivation / Background
- Basic Multigrid
- Parallel Multigrid

- Algebraic Multigrid
  - Classical AMG
  - Parallel AMG
  - Smoothed Aggregation
- AMG Theory and Compatible Relaxation
- AMG for Electromagnetic Problems
- Adaptive AMG

- Summary Information

Preliminaries…

- Consider solving the $N \times N$ linear system

$$Au = f$$

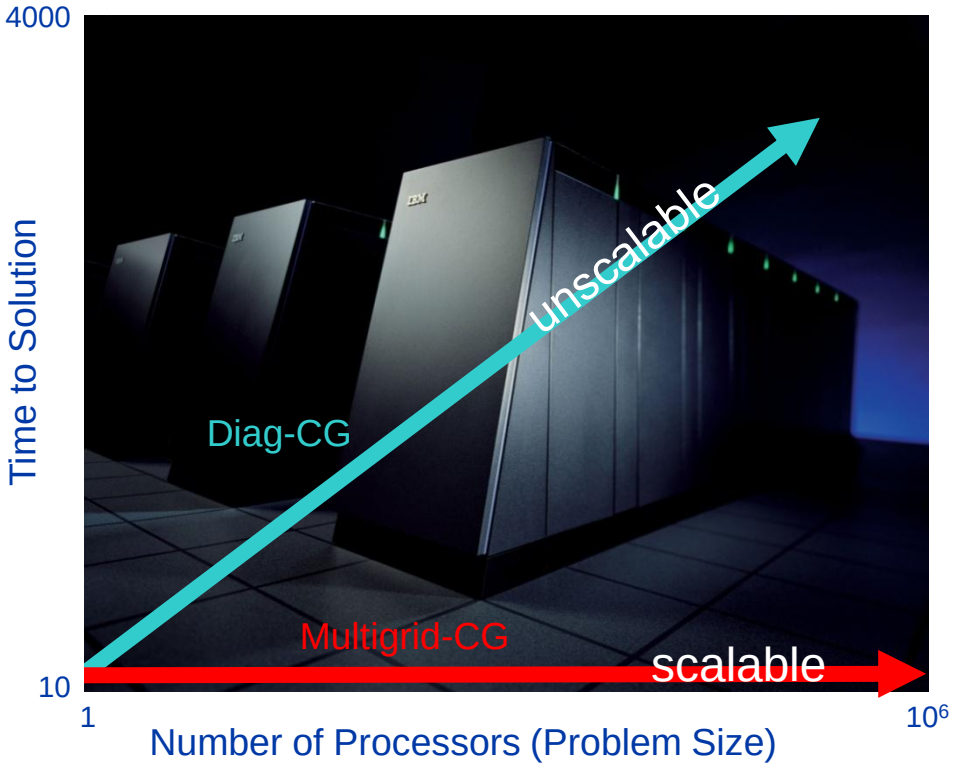
- Most iterative methods have the following form, where $r_k = f - Au_k$ is the residual at iteration k

$$u_{k+1} = u_k + M^{-1} r_k$$

- Let $e_k = u - u_k$ be the error, and note that $r_k = Ae_k$
- The error propagation for the iterative method is

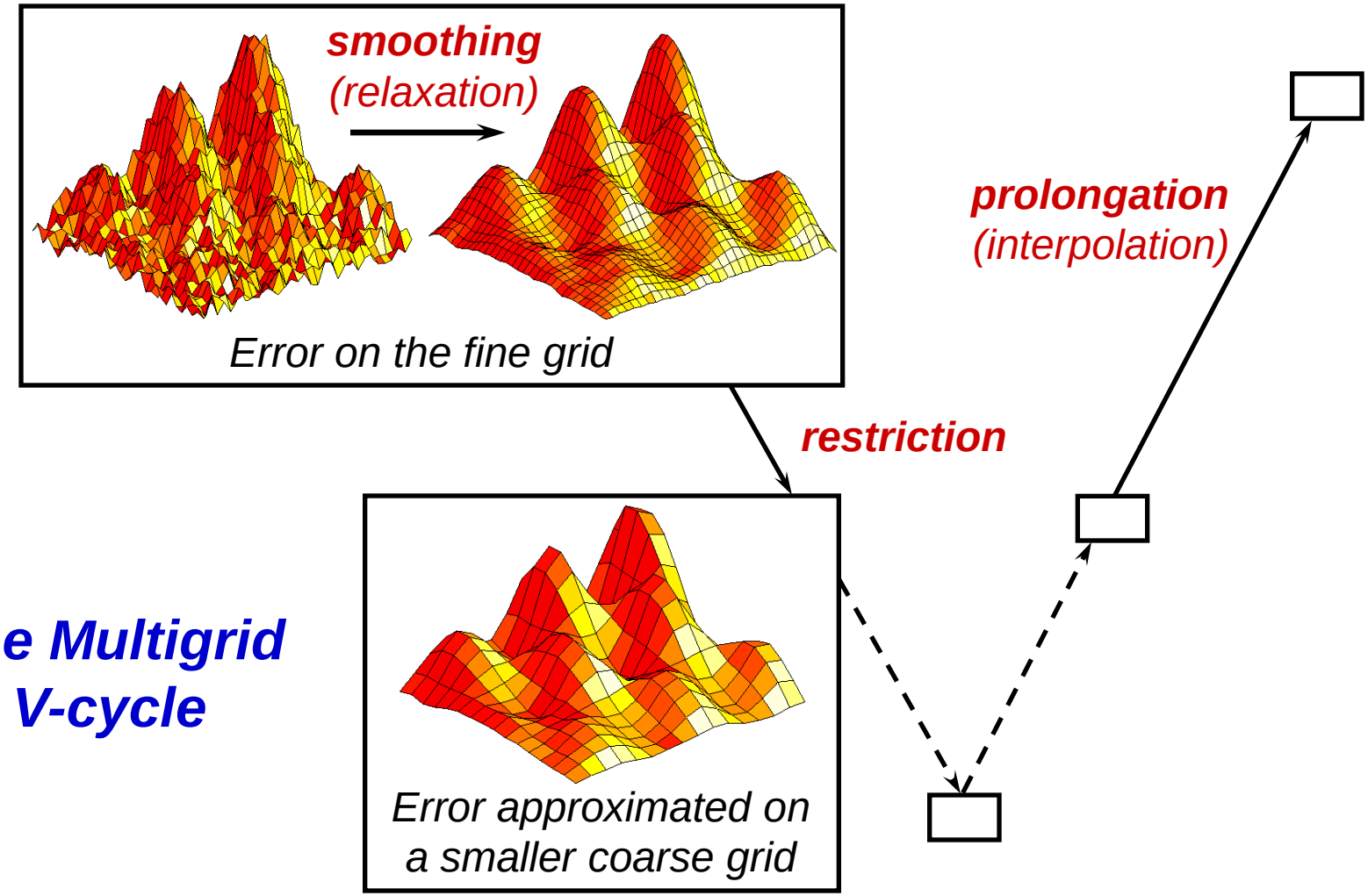
$$e_{k+1} = (I - M^{-1}A)e_k$$

Multigrid linear solvers are optimal ($O(N)$ operations), and hence have good scaling potential



- Weak scaling – want constant solution time as problem size grows in proportion to the number of processors

Multigrid uses a sequence of coarse grids to accelerate the fine grid solution



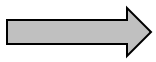
The Multigrid V-cycle

Simple 1D model problem

- 1D Laplace on a uniform grid with spacing h

$$-u_{xx} = f \quad \text{on } \Omega = [0,1]$$
$$u = g \quad \text{on } \Gamma$$

Continuous



$$-u_{i-1} + 2u_i - u_{i+1} = h^2 f_i$$
$$u_0 = u_{N+1} = g$$

Discrete

- Discrete problem is a linear system $Au = f$ with

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

Matrix

or

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

Stencil

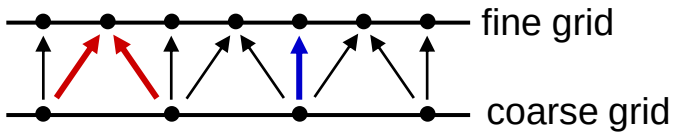
- We will mostly use the stencil form

Multigrid components for 1D model

- Many smoother options, e.g.,
  - Weighted Jacobi, $I - (\omega/2)\, A$, $\omega_{opt} = 2/3$
  - Gauss-Seidel (GS)

- Prolongation is linear interpolation (note brackets)

$$P = \left] \begin{array}{ccc} \frac{1}{2} & 1 & \frac{1}{2} \end{array} \right[$$



- Coarse-grid operator is coarse discretization of the problem (scaled appropriately)

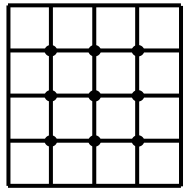
$$A_c = \frac{1}{2} \left[\begin{array}{ccc} -1 & 2 & -1 \end{array} \right]$$

- In practice, a slightly different method (equivalent to cyclic reduction) solves this problem in one *V*-cycle

2D model problem: Laplace on a square (1)

- Five-point stencil discretization on a uniform grid

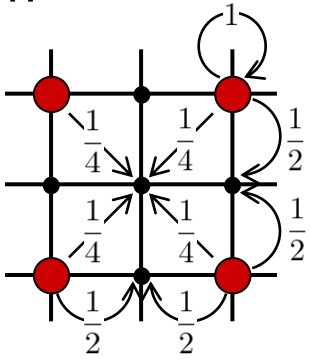
$$-\nabla^2 u = f$$



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

- 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

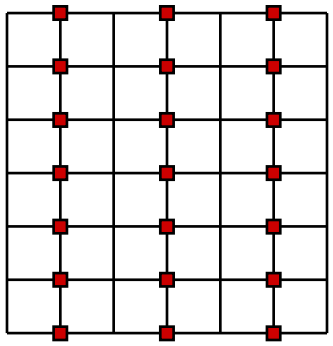
2D anisotropic model problem on a square

- Five-point stencil discretization on a uniform grid

$$-u_{xx} - \epsilon u_{yy} = f, \quad 0 < \epsilon \ll 1 \qquad A = \begin{bmatrix} & -\epsilon & \\ -1 & 2+2\epsilon & -1 \\ & -\epsilon & \end{bmatrix}$$

- Pointwise relaxation smooths only in the x direction!

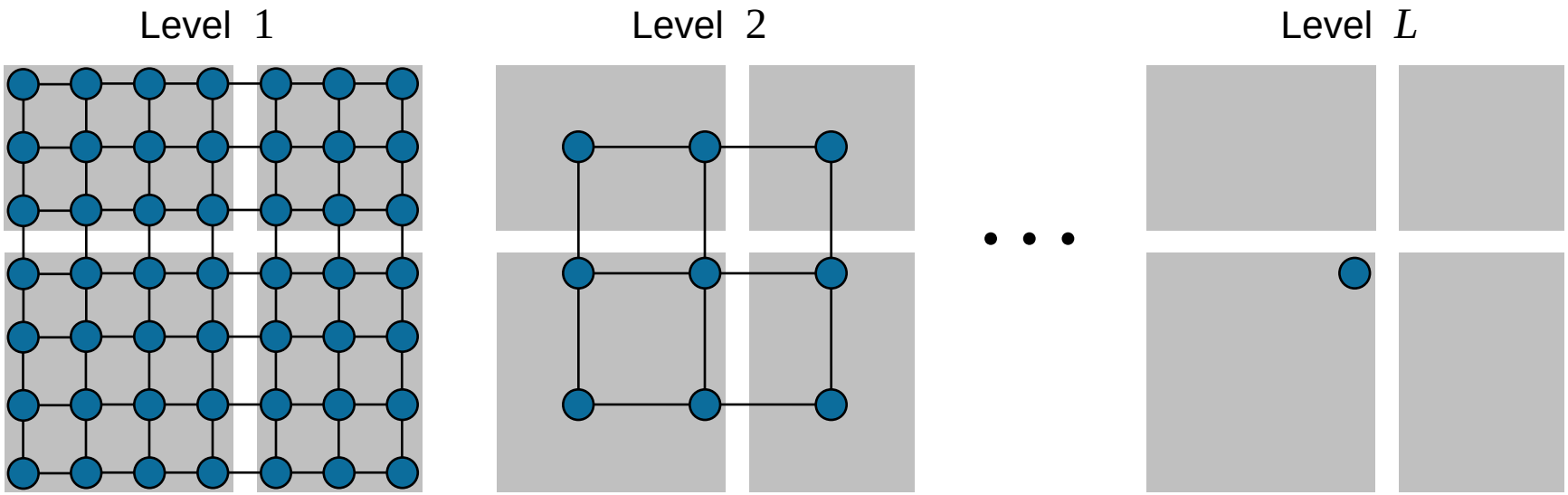
- Two solutions:
  1) Change coarse-grid correction – coarsen only in the direction of smoothness (semicoarsening in x, linear interpolation in x)
  2) Change relaxation – line relaxation with points grouped along y lines



Parallel Multigrid

(see Yang tutorial on Monday)

Approach for parallelizing multigrid is straightforward data decomposition



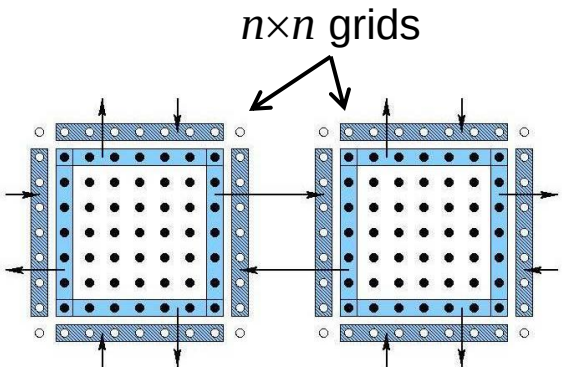
- Basic communication pattern is “nearest neighbor”
  - Relaxation, interpolation, & Galerkin not hard to implement
- Different neighbor processors on coarse grids
- Many idle processors on coarse grids (100K+ on BG/L)
  - Algorithms to take advantage have had limited success

Straightforward parallelization approach is optimal for V-cycles on structured grids (5-pt Laplacian example)

- Standard communication / computation models

$$T_{comm} = \alpha + m\beta \quad \textit{(communicate m doubles)}$$

$$T_{comp} = m\gamma \quad \textit{(compute m flops)}$$



- Time to do relaxation

$$T \approx 4\alpha + 4n\beta + 5n^2\gamma$$

- Time to do relaxation in a V(1,0) multigrid cycle

$$\begin{aligned} T_V &\approx (1+1+\cdots)4\alpha + (1+1/2+\ldots)4n\beta + (1+1/4+\ldots)5n^2\gamma \\ &\approx (\log N)4\alpha + (2)4n\beta + (4/3)5n^2\gamma \end{aligned}$$

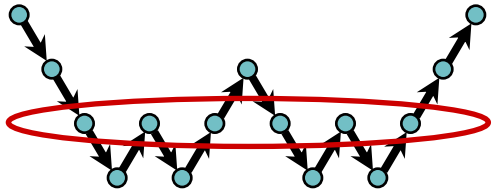
- For achieving optimality in general, the *log* term is unavoidable!

- More precise: $T_{V,better} \approx T_V + (\log P)(4\beta + 5\gamma)$

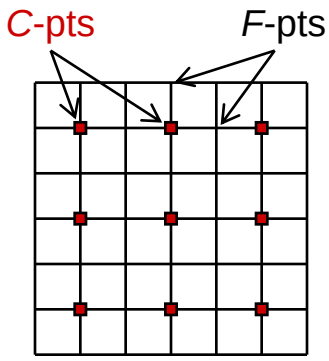
Additional comments on parallel multigrid

- W-cycles scale poorly:

$$T_W \approx (2^{\log N})4\alpha + (\log N)4n\beta + (2)5n^2\gamma$$

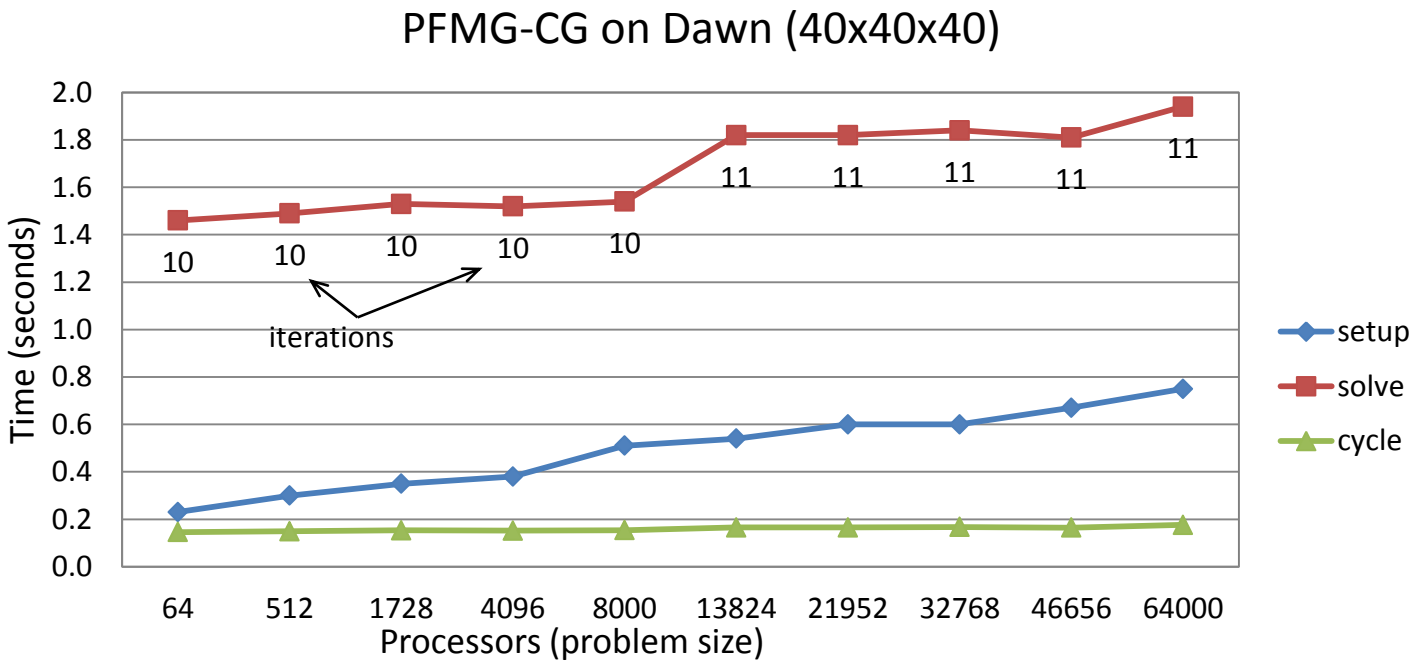


- Lexicographical Gauss-Seidel is too sequential
  - Use red/black or multi-color GS
  - Use weighted Jacobi, hybrid Jacobi/GS, L1
  - Use *C*-*F* relaxation (Jacobi on *C*-pts then *F*-pts)
  - Use Polynomial smoothers
- Parallel smoothers are often less effective



- Recent survey on parallel multigrid:
  - "A Survey of Parallelization Techniques for Multigrid Solvers," Chow, Falgout, Hu, Tuminaro, and Yang, *Parallel Processing For Scientific Computing*, Heroux, Raghavan, and Simon, editors, SIAM, series on Software, Environments, and Tools (2006)
- Recent paper on parallel smoothers:
  - "Multigrid Smoothers for Ultra-Parallel Computing," Baker, Falgout, Kolev, and Yang, *SIAM J. Sci. Comput.*, submitted. LLNL-JRNL-435315

Example weak scaling results on Dawn (an IBM BG/P system at LLNL) in 2010



- Laplacian on a cube; 40^3 = 64K grid points per processor; largest problem had 3 billion unknowns!
- PFMG is a semicoarsening multigrid solver in *hypre*
- Still room to improve setup implementation (these results already employ the assumed partition algorithm described later)

Basic multigrid research challenge

- Optimal $O(N)$ multigrid methods don't exist for some applications, even in serial
- Need to invent methods for these applications

- However …

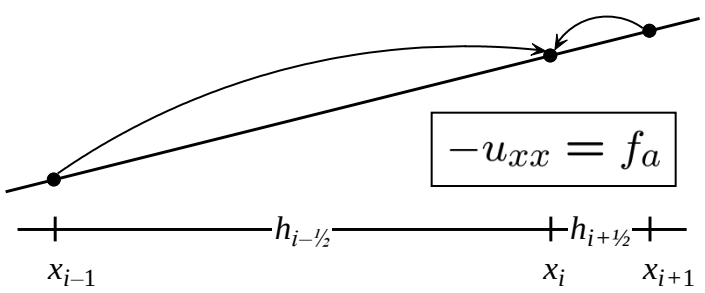
- Some of the classical and most proven techniques used in multigrid methods don't parallelize
  - Gauss-Seidel smoothers are inherently sequential
  - W-cycles have poor parallel scaling
- Parallel computing imposes additional restrictions on multigrid algorithmic development

Algebraic Multigrid (AMG)

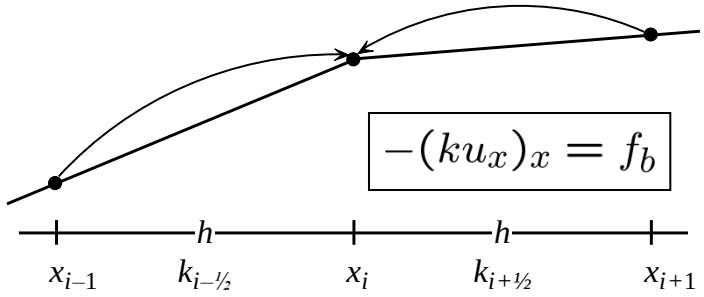
Algebraic Multigrid (AMG) is based on MG principles, but uses matrix coefficients

- For best results, geometry alone is not enough

Linear Interpolation



Operator-Dependent Interpolation



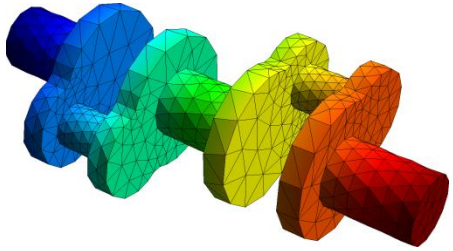
- AMG ignores geometric information altogether, but captures both linear & operator-dep interpolation

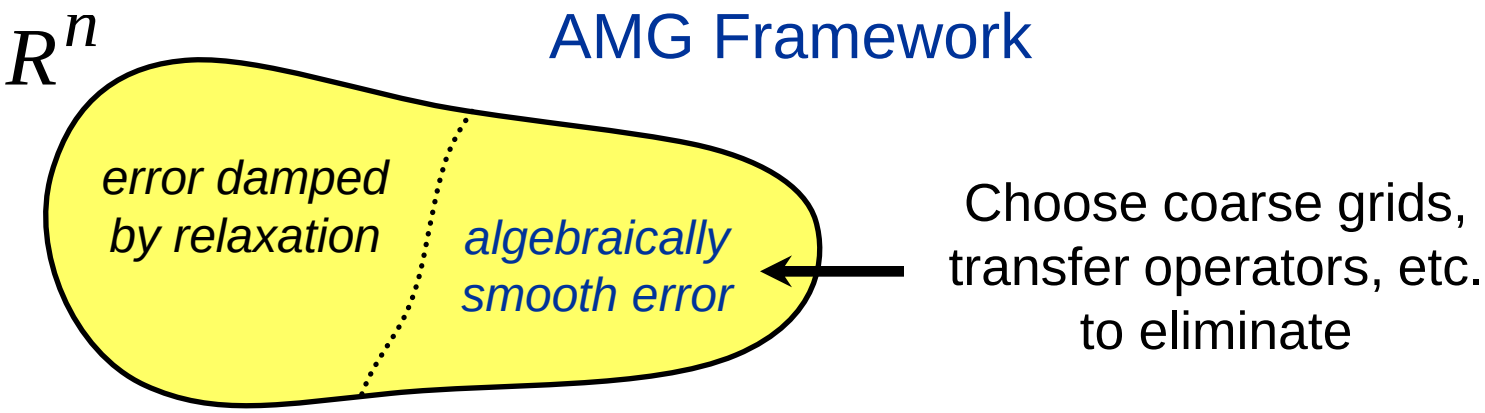
$$(A\mathbf{u})_i = a_{i,i-1}u_{i-1} + a_{i,i}u_i + a_{i,i+1}u_{i+1}$$

$$u_i = \left(-\frac{a_{i,i-1}}{a_{i,i}}\right) u_{i-1} + \left(-\frac{a_{i,i+1}}{a_{i,i}}\right) u_{i+1}$$

AMG is an ideal method for unstructured grid problems

- Many algorithms (AMG alphabet soup)
- Automatically coarsens “grids”





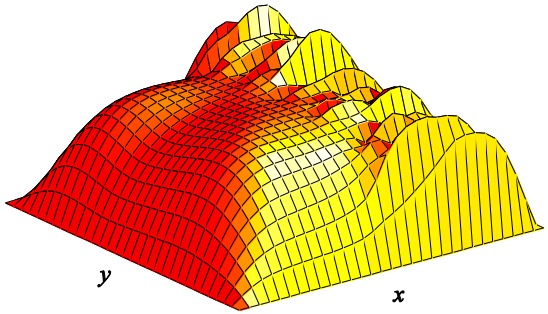
- Algebraically smooth error may not be smooth in a geometric sense

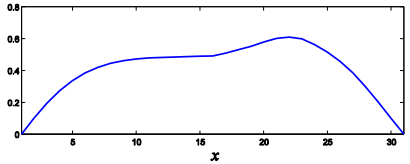
Error left by relaxation can be geometrically oscillatory

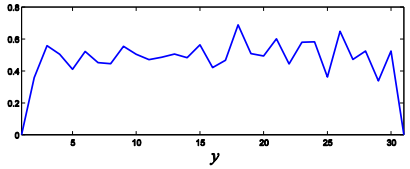
- 7 GS sweeps on

$$-au_{xx} - bu_{yy} = f$$

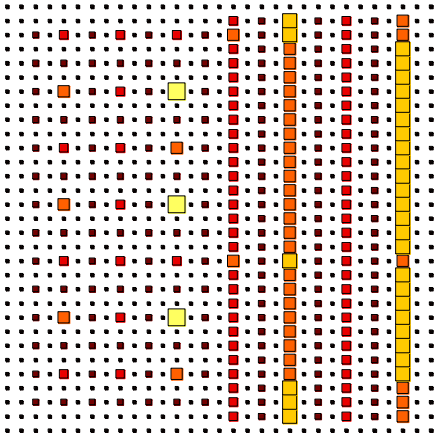
$a = b$	$a \gg b$







- This example…
  - targets geometric smoothness
  - uses pointwise smoothers
- Not sufficient for some problems!



AMG coarsens grids in the direction of geometric smoothness

Preliminaries… the Galerkin coarse-grid operator

- As before, consider solving the $N \times N$ linear system

$$Au = f$$

- Let P be prolongation (interpolation) and P^T restriction

- The coarse-grid operator is defined by the Galerkin procedure, $A_c = P^T A P$
- This gives the "best" coarse-grid correction in the sense that the solution e_c of the coarse system

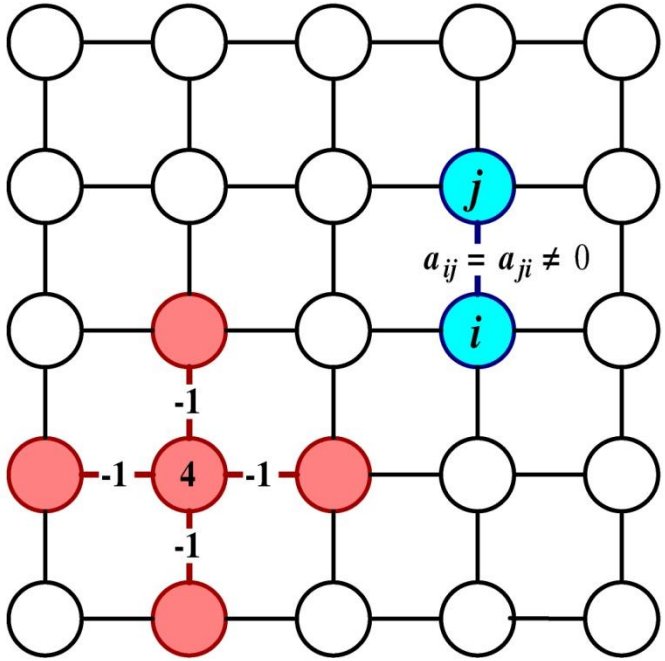
$$A_c e_c = P^T r$$

satisfies

$$e_c = \arg \min \|e - P e_c\|_A$$

Preliminaries… AMG “grids”

- Matrix adjacency graphs play an important role in AMG:
  - grid = set of graph vertices
  - grid point i = vertex i
- As a visual aid, it is highly instructive to relate the matrix equations to an underlying PDE and discretization
- We will often draw the grid points in their geometric locations
- Remember that AMG doesn’t actually use this geometric information!



Classical AMG (C-AMG)

(Brandt, McCormick, Ruge, Stüben)

C-AMG targets geometric smoothness

- From theory (later): smooth error is characterized by small eigenmodes, hence satisfies (A scaled to have norm 1)

$$\mathbf{e}^T A\mathbf{e} = \lambda \ll 1$$

- Constant is geometrically smooth, so assume zero row sum

$$\begin{aligned}
\mathbf{e}^T A\mathbf{e} &= \sum_i e_i \left(a_{ii} e_i + \sum_{j \neq i} a_{ij} e_j \right) \\
&= \sum_i e_i \left(\sum_{j \neq i} (-a_{ij})(e_i - e_j) \right) \qquad a_{ii} = -\sum_{j \neq i} a_{ij} \\
&= \sum_{i<j} (-a_{ij}) e_i (e_i - e_j) + \sum_{i>j} (-a_{ij}) e_i (e_i - e_j) \\
&= \sum_{i<j} (-a_{ij}) e_i (e_i - e_j) - \sum_{i<j} (-a_{ji}) e_j (e_i - e_j) \qquad \text{swap } i,j \\
&= \sum_{i<j} (-a_{ij})(e_i - e_j)^2 \ll 1
\end{aligned}$$

C-AMG targets geometric smoothness through strength-of-connection

- Assuming geometric smoothness, can show

$$\mathbf{e}^T A\mathbf{e} = \sum_{i<j}(-a_{ij})(e_i - e_j)^2 \ll 1$$

- C-AMG Smoothness Heuristic: Smooth error varies slowly in the direction of "large" matrix coefficients

- Strength of connection: Given a threshold $0 < \theta \leq 1$, we say that variable u_i strongly depends on variable u_j if

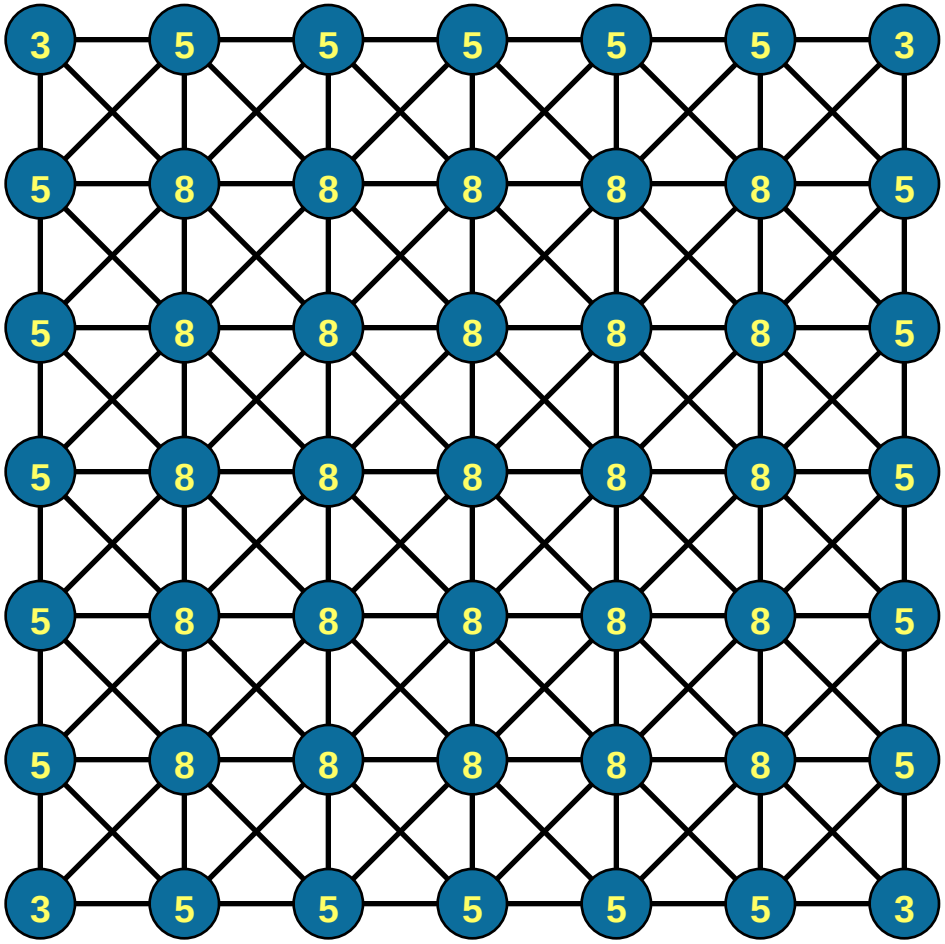
$$-a_{ij} \geq \theta \max_{k \neq i}\{-a_{ik}\}$$

- In practice, positive off-diagonals are weak
- Note that this definition of strength is not symmetric

Choosing the coarse grid

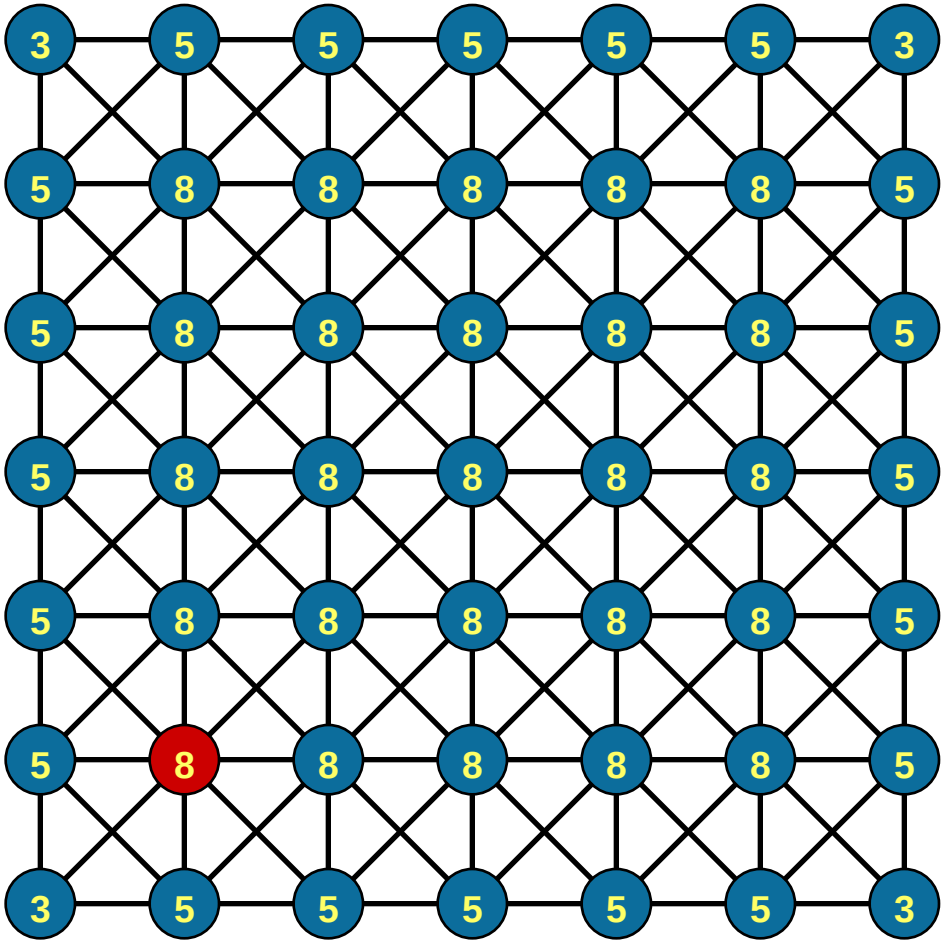
- In C-AMG, the coarse grid is a subset of the fine grid
- The basic coarsening procedure is as follows:
  - Define a strength matrix A_s by deleting weak connections in A
  - First pass: Choose an independent set of fine-grid points based on the graph of A_s
  - Second pass: Choose additional points if needed to satisfy interpolation requirements
- Coarsening partitions the grid into *C*- and *F*-points

C-AMG coarsening



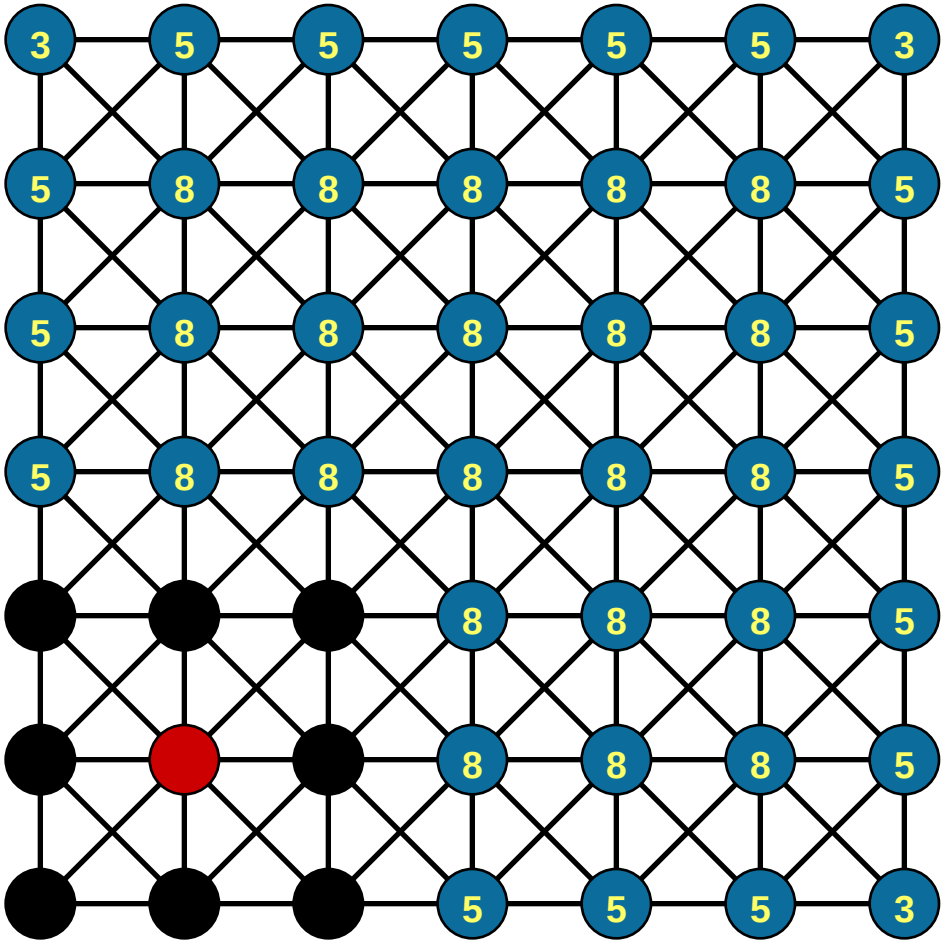
- select C-pt with maximal measure
- select neighbors as F-pts
- update measures of F-pt neighbors

C-AMG coarsening



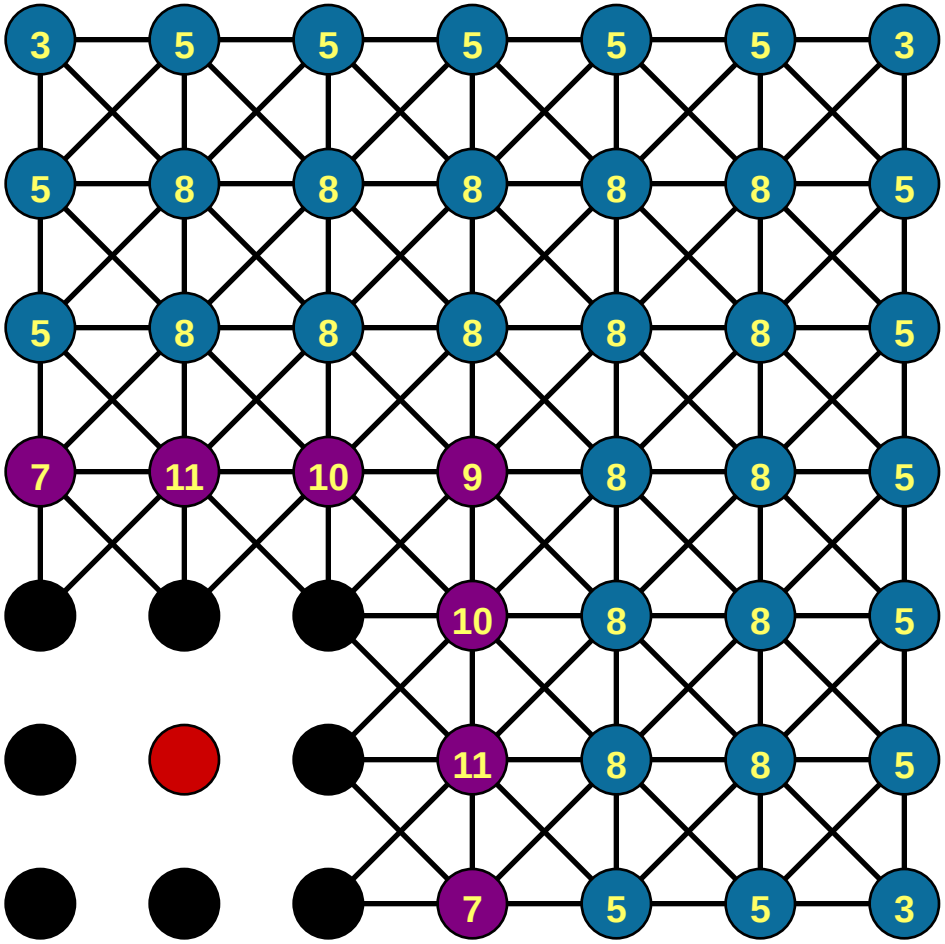
- ➔ **select C-pt with maximal measure**
- ➔ **select neighbors as F-pts**
- ➔ **update measures of F-pt neighbors**

C-AMG coarsening



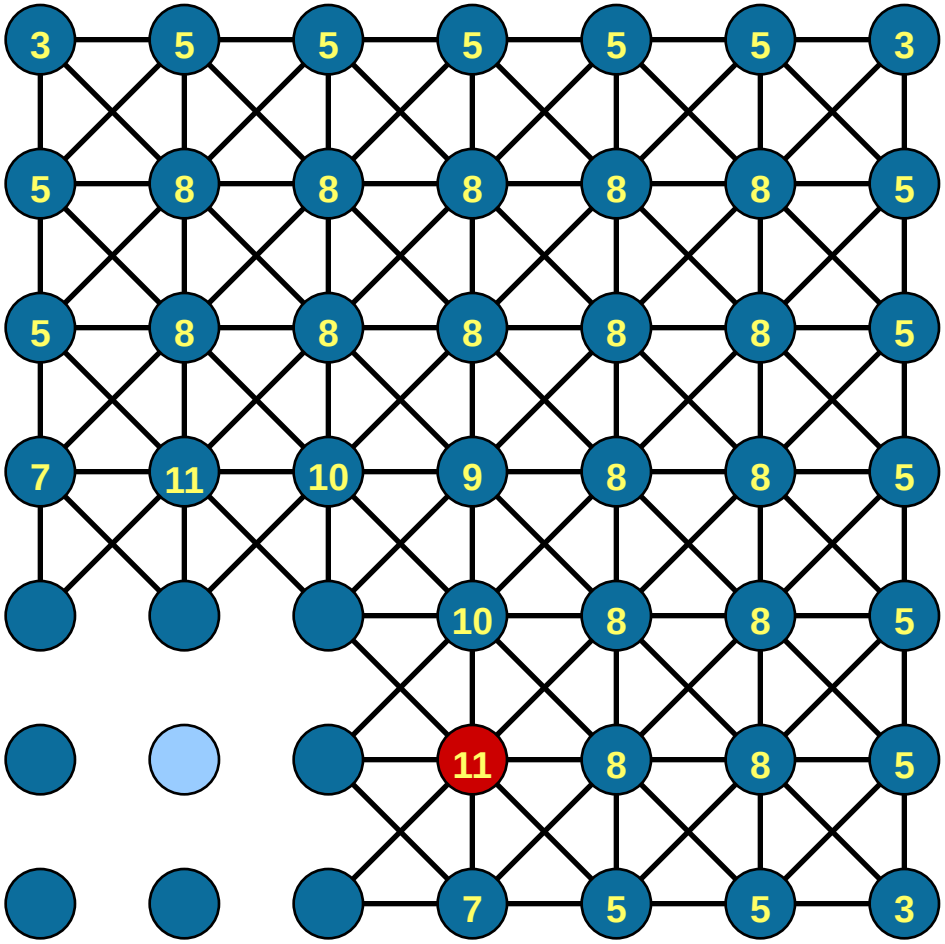
- select C-pt with maximal measure
- select neighbors as F-pts
- update measures of F-pt neighbors

C-AMG coarsening



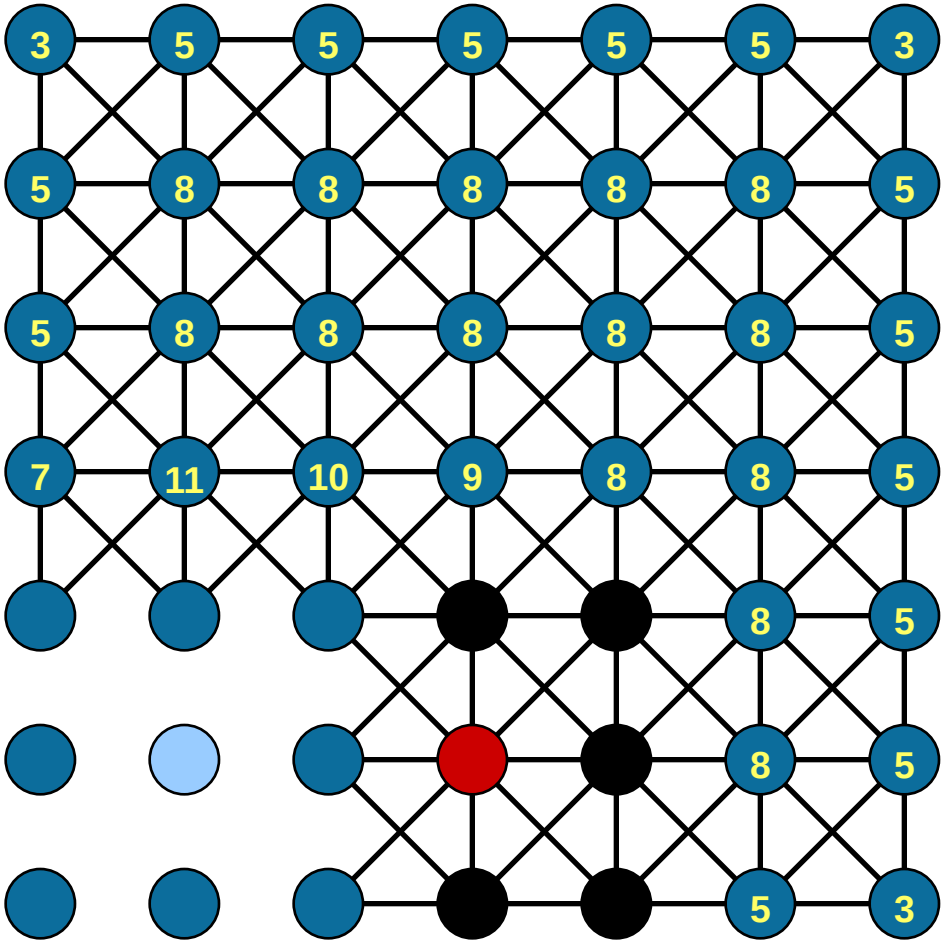
- select C-pt with maximal measure
- select neighbors as F-pts
- update measures of F-pt neighbors

C-AMG coarsening



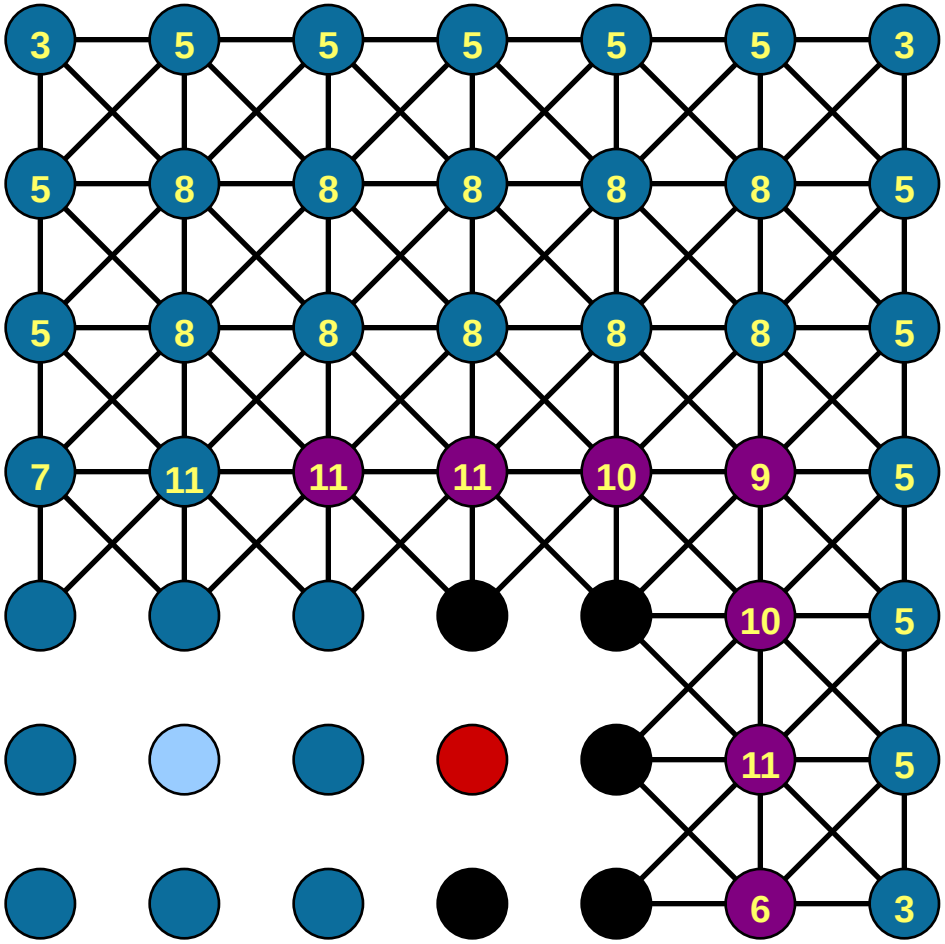
- select C-pt with maximal measure
- select neighbors as F-pts
- update measures of F-pt neighbors

C-AMG coarsening



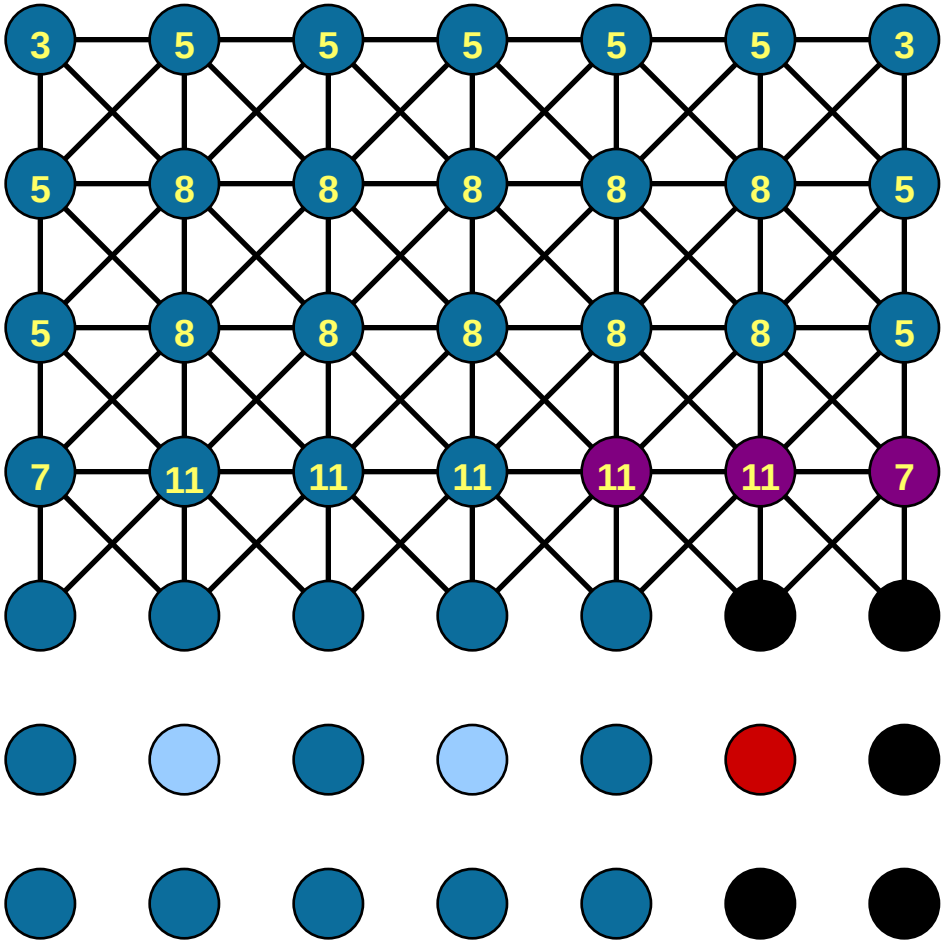
- select C-pt with maximal measure
- select neighbors as F-pts
- update measures of F-pt neighbors

C-AMG coarsening



- select C-pt with maximal measure
- select neighbors as F-pts
- update measures of F-pt neighbors

C-AMG coarsening

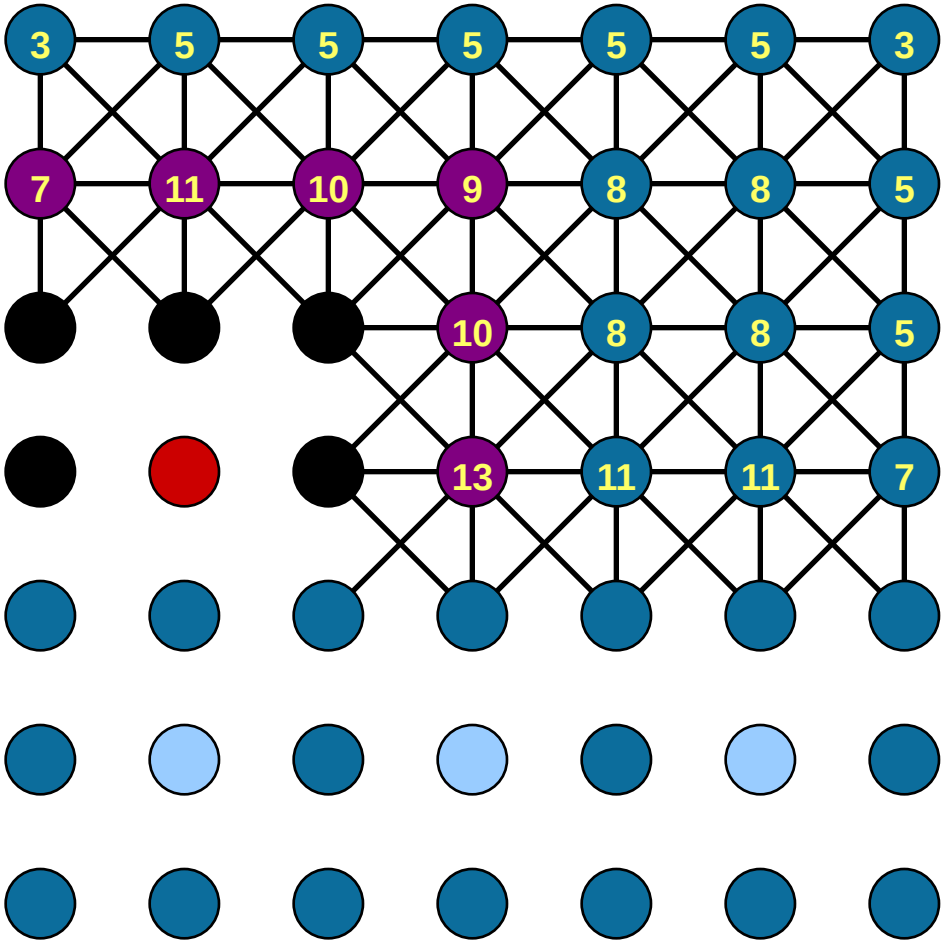


➔ select C-pt with maximal measure

➔ select neighbors as F-pts

➔ update measures of F-pt neighbors

C-AMG coarsening

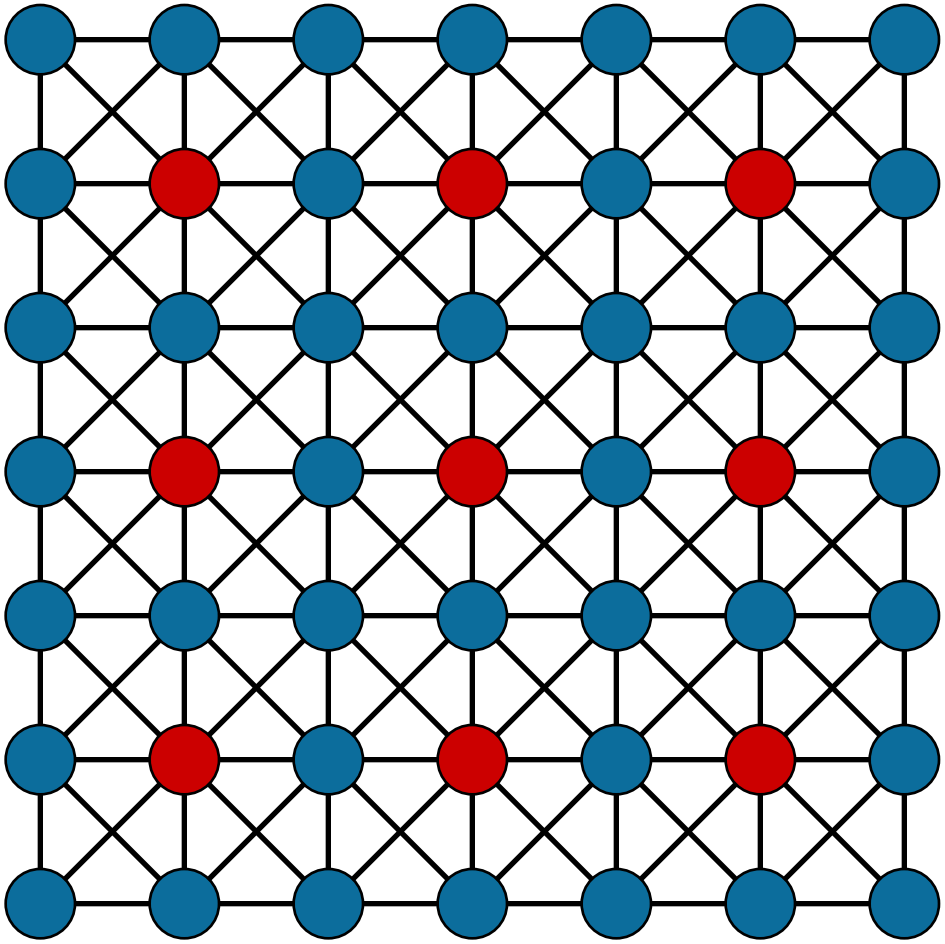


➔ select C-pt with maximal measure

➔ select neighbors as F-pts

➔ update measures of F-pt neighbors

C-AMG coarsening is inherently sequential



- select C-pt with maximal measure
- select neighbors as F-pts
- update measures of F-pt neighbors

C-AMG coarsening – second pass

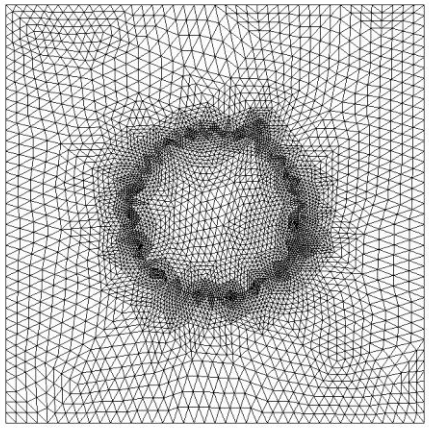
- Recall: Second pass chooses additional points if needed to satisfy interpolation requirements

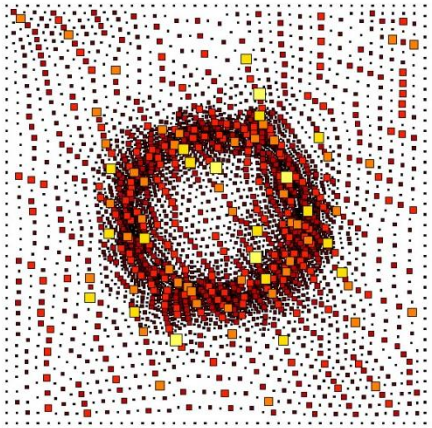
- C-AMG interpolation (discussed next) requires that each pair of strongly connected *F*-points be strongly connected to a common *C*-point
- C-AMG second pass: search for *F*-point pairs that don't satisfy this requirement and changes one to a *C*-point

- Second pass can lead to high complexity
- Idea: eliminate second pass, modify interpolation

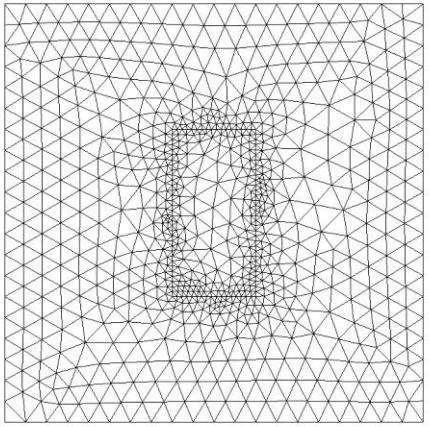
AMG grid hierarchies for several 2D problems

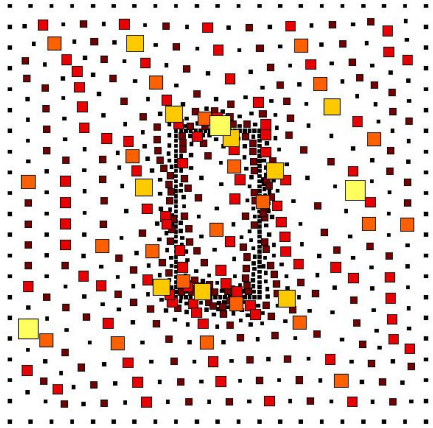
domain1 - 30^o



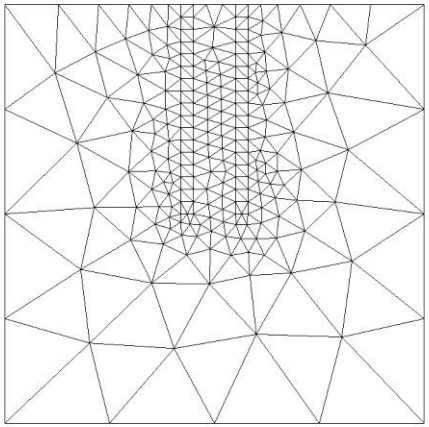


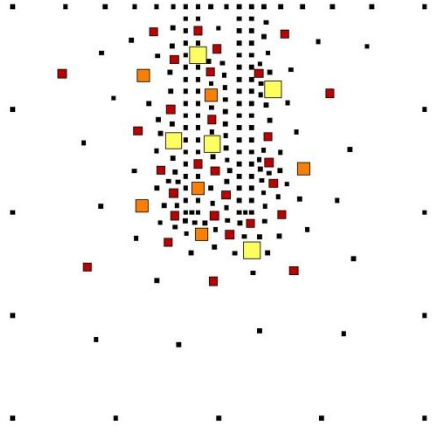
domain2 - 30^o



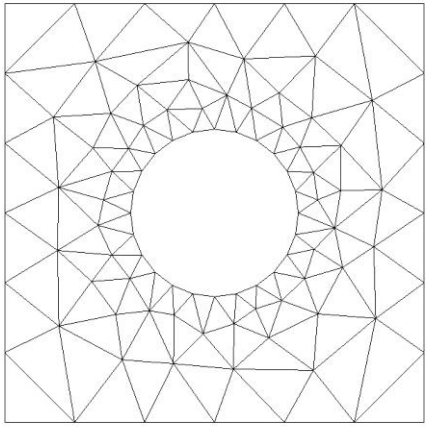


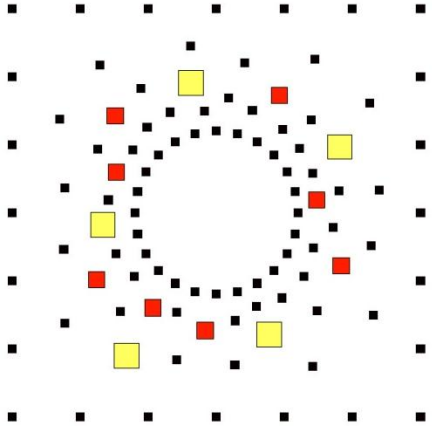
pile





square-hole





C-AMG Interpolation – collapsing the stencil

- Smooth error means “small” residuals

$$r \ll \|A\|$$

- To derive interpolation, assume that

$$r_i = (Ae)_i = 0$$

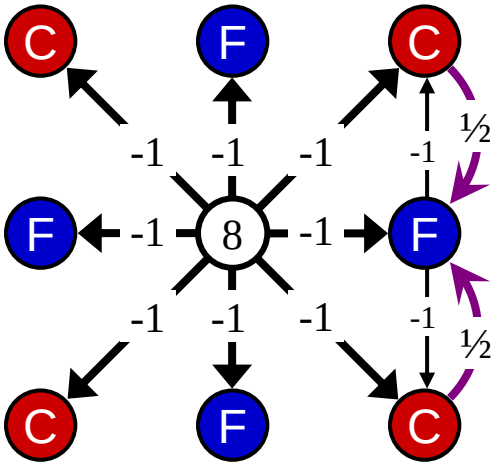
- Hence,

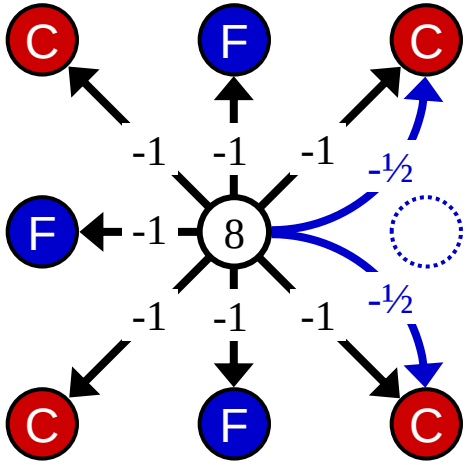
$$a_{ii}e_i = -\sum_{j \in C_i} a_{ij}e_j - \sum_{j \in F_i^s} a_{ij}e_j - \sum_{j \in N_i^w} a_{ij}e_j$$

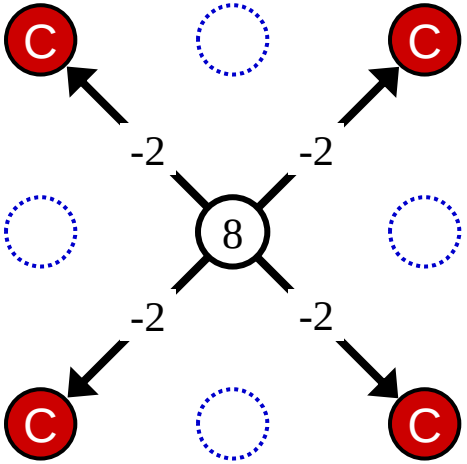
C_i : C-points strongly connected to i
F_i^s : F-points strongly connected to i
N_i^w : all points weakly connected to i.

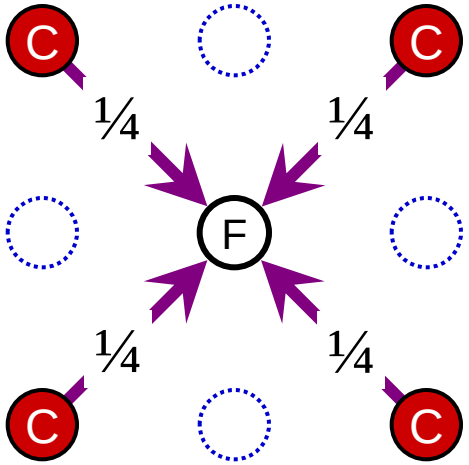
- The trick is to rewrite the e_j in F_i^s and N_i^w in terms of either the interpolatory points in C_i or the F-point i

C-AMG Interpolation – collapsing the stencil…
Isotropic 9-pt Laplacian

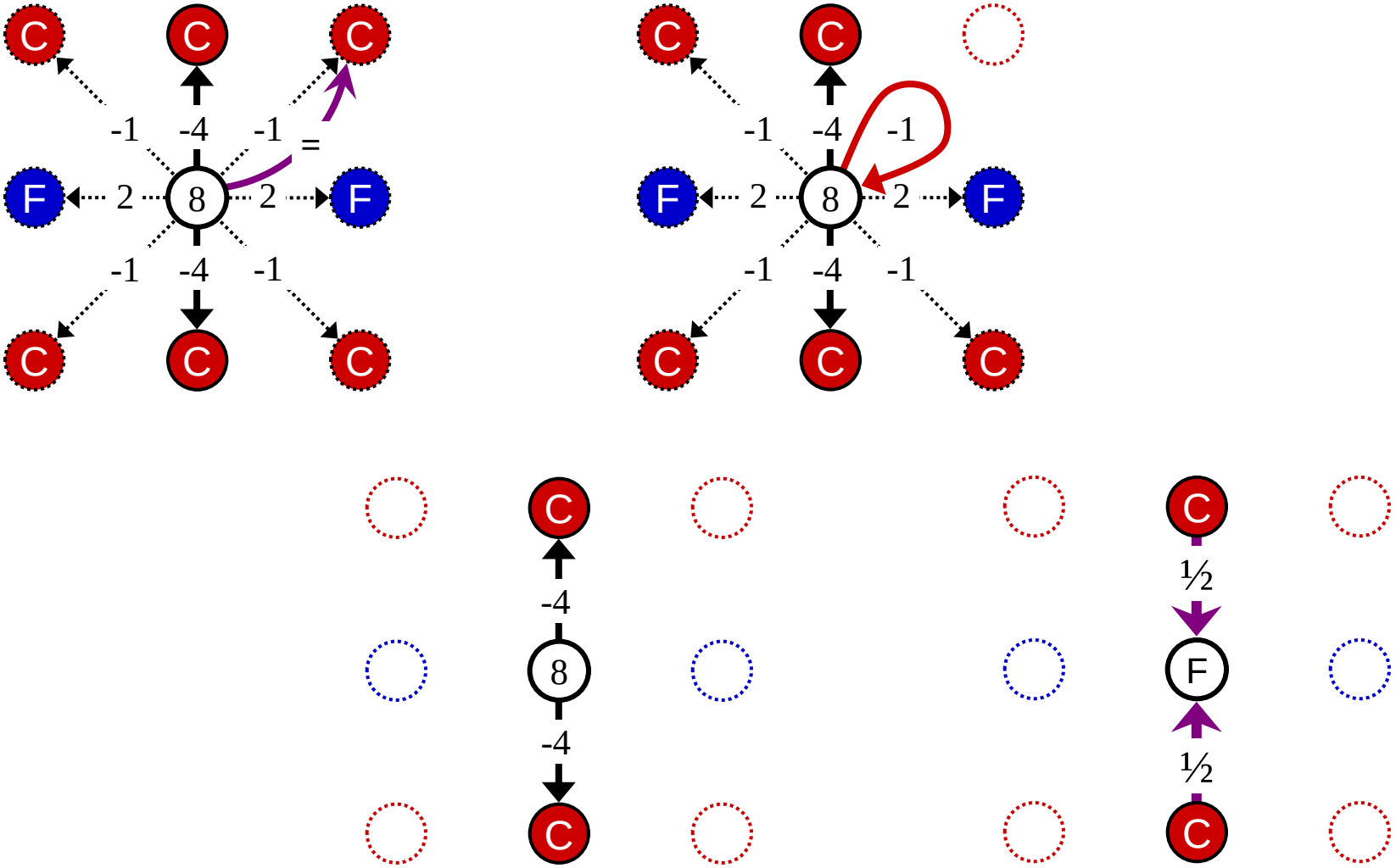








C-AMG Interpolation – collapsing the stencil…
Anisotropic 9-pt Laplacian, $\theta > 0.25$



C-AMG Interpolation – algebraic derivation

$$a_{ii}e_i = -\sum_{j\in C_i} a_{ij}e_j - \sum_{j\in F_i^s} a_{ij}e_j - \sum_{j\in N_i^w} a_{ij}e_j$$

C_i : C-points strongly connected to i
F_i^s : F-points strongly connected to i
N_i^w : all points weakly connected to i.

- Write

$$e_j = \begin{cases} \sum_{k\in C_i} \left(\frac{a_{jk}}{\sum_{l\in C_i} a_{jl}} \right) e_k, & j \in F_i^s \\ e_i, & j \in N_i^w \end{cases}$$

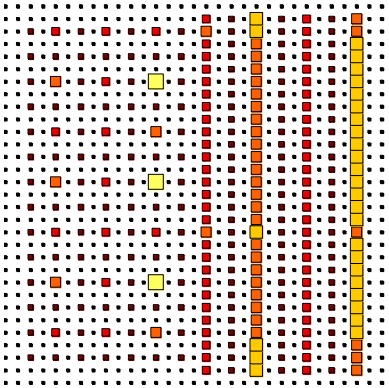
- Then

$$w_{ij} = -\left(a_{ij} + \sum_{k\in F_i^s} \left(\frac{a_{kj}}{\sum_{l\in C_i} a_{kl}} \right) a_{ik} \right) / \left(a_{ii} + \sum_{k\in N_i^w} a_{ik} \right)$$

Example C-AMG results

$$-au_{xx} - bu_{yy} = f$$

$a = b$	$a \gg b$



C-AMG coarse grids

N	Iters	Conv factor	Coarse grids	Grid comp	Oper comp	Setup time	Solve time
61×61	10	0.23	6	1.6	1.6	0.01	0.02
121×121	9	0.23	8	1.6	1.7	0.05	0.07
241×241	9	0.23	9	1.6	1.7	0.25	0.32
481×481	9	0.23	12	1.7	1.7	1.02	1.27
961×961	11	0.29	13	1.7	1.7	4.42	6.28

- Grid complexity – total # of grid points divided by the # of fine grid points
- Operator complexity – total # of nonzeroes in the system operators divided by the # of nonzeroes in the fine grid operator

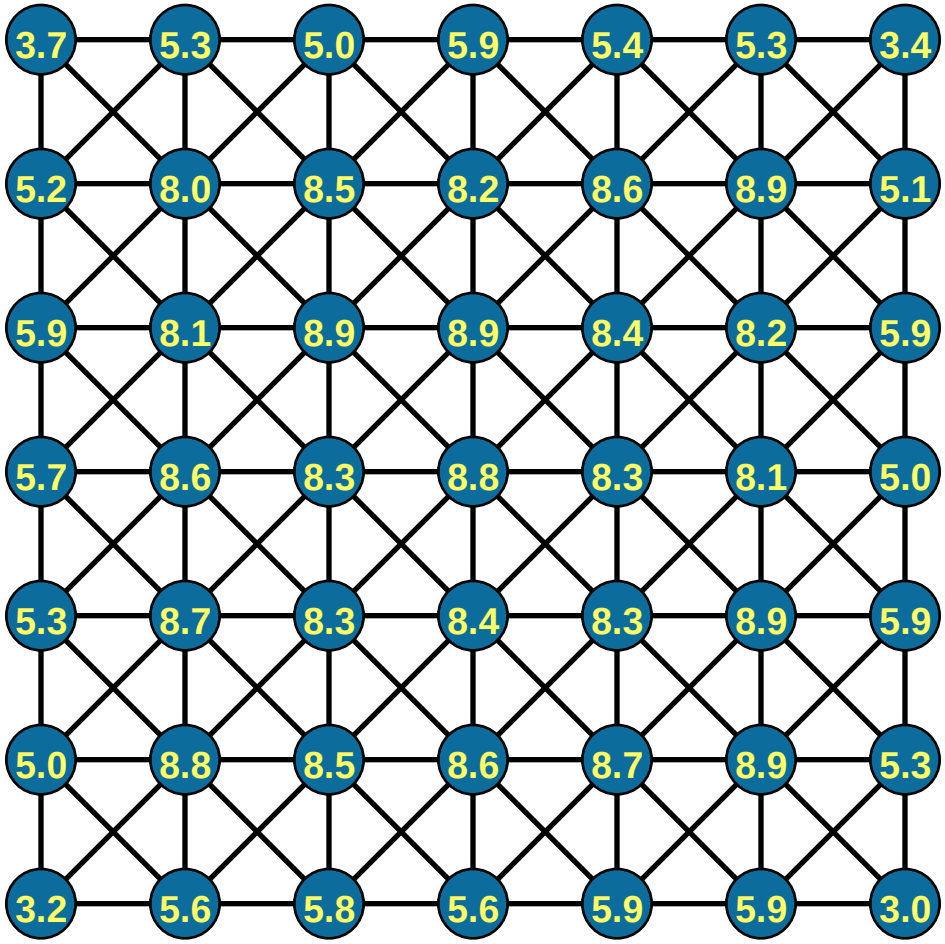
Parallel AMG

Parallel Coarsening Algorithms

- C-AMG coarsening algorithm is inherently sequential

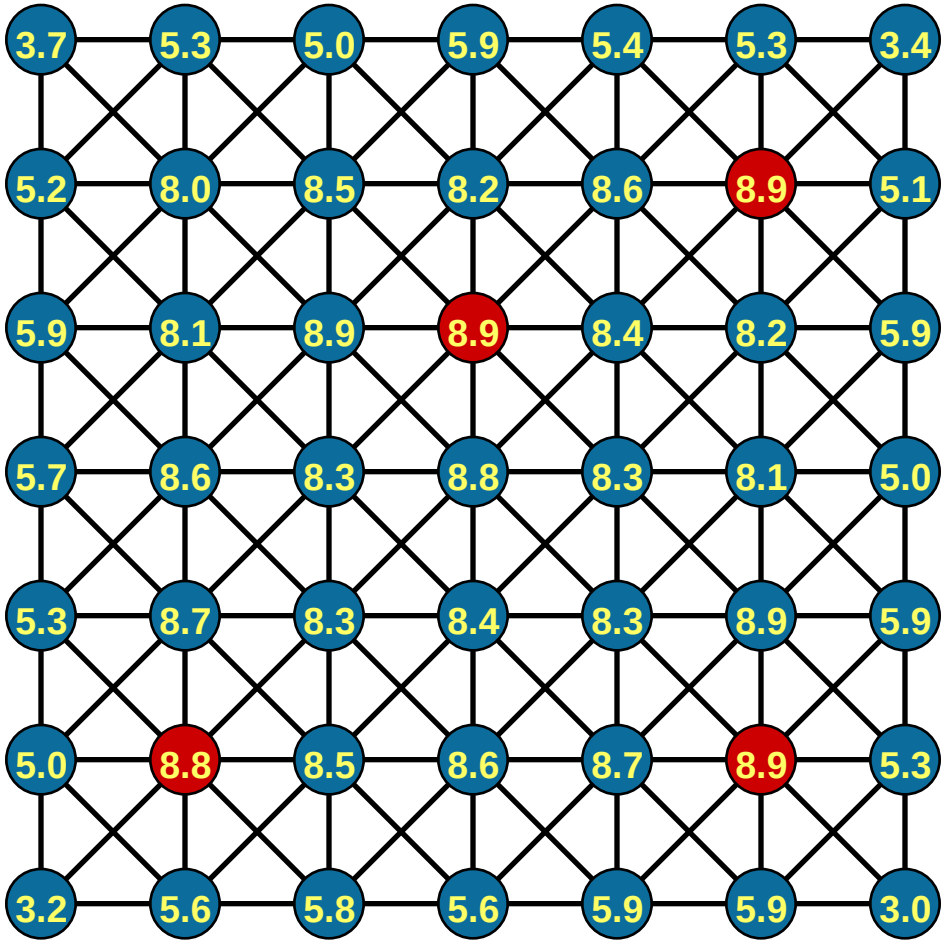
- Several parallel algorithms (in *hypre*):
  - CLJP (Cleary-Luby-Jones-Plassmann) – one-pass approach with random numbers to get concurrency (illustrated next)
  - Falgout – C-AMG on processor interior, then CLJP to finish
  - PMIS – CLJP without the 'C'; parallel version of C-AMG first pass
  - HMIS – C-AMG on processor interior, then PMIS to finish
  - CGC (Griebel, Metsch, Schweitzer) – compute several coarse grids on each processor, then solve a global graph problem to select the grids with the best "fit"
  - …
- Other parallel AMG codes use similar approaches

CLJP coarsening is fully parallel



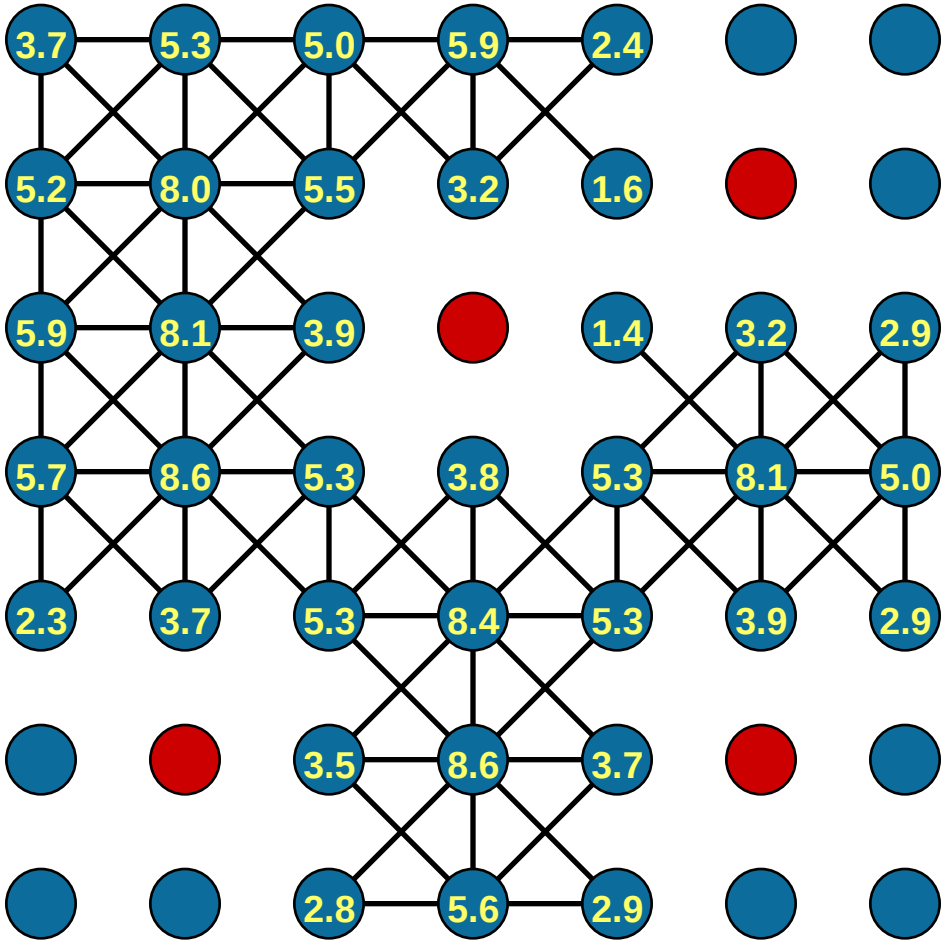
- select C-pts with maximal measure locally
- remove neighbor edges
- update neighbor measures

CLJP coarsening is fully parallel



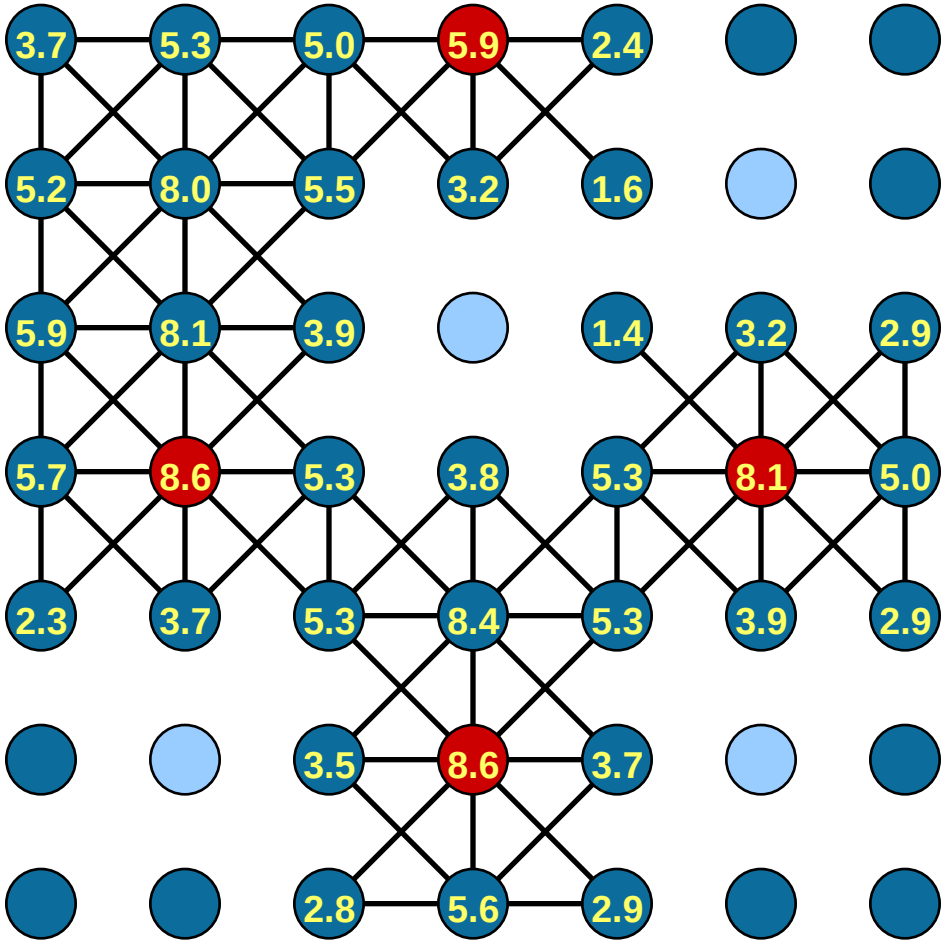
- select C-pts with maximal measure locally
- remove neighbor edges
- update neighbor measures

CLJP coarsening is fully parallel



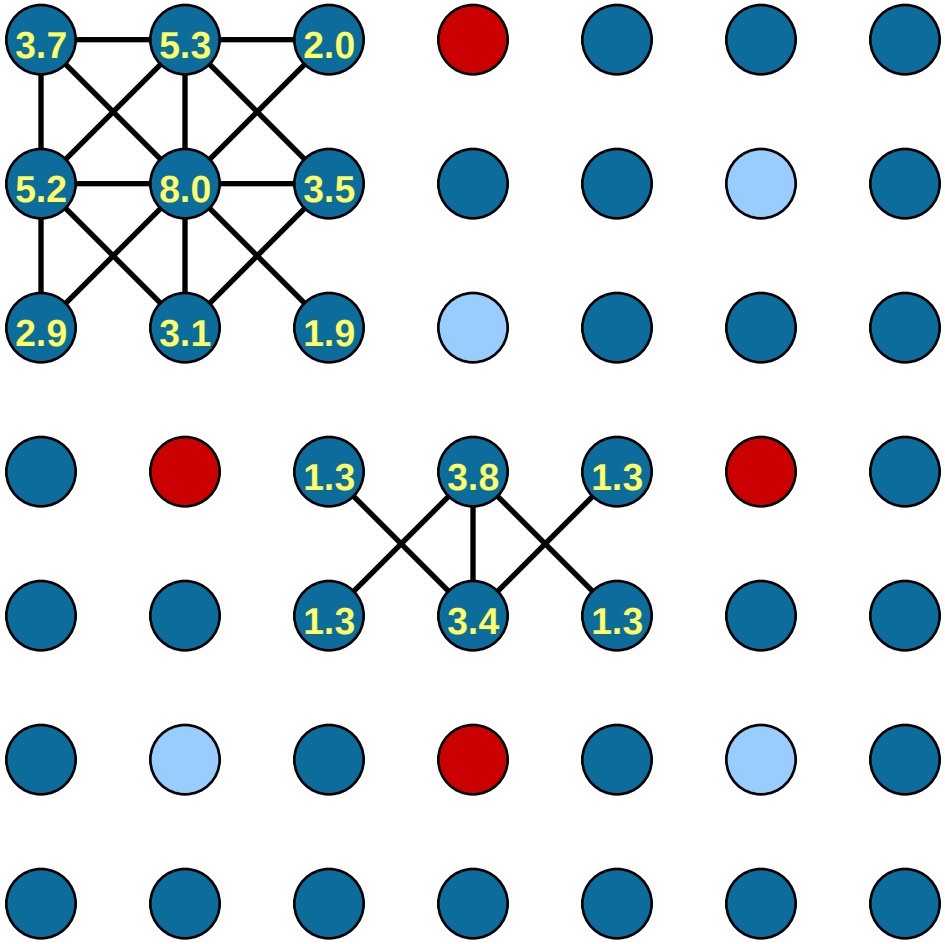
- ➔ **select C-pts with maximal measure locally**
- ➔ **remove neighbor edges**
- ➔ **update neighbor measures**

CLJP coarsening is fully parallel



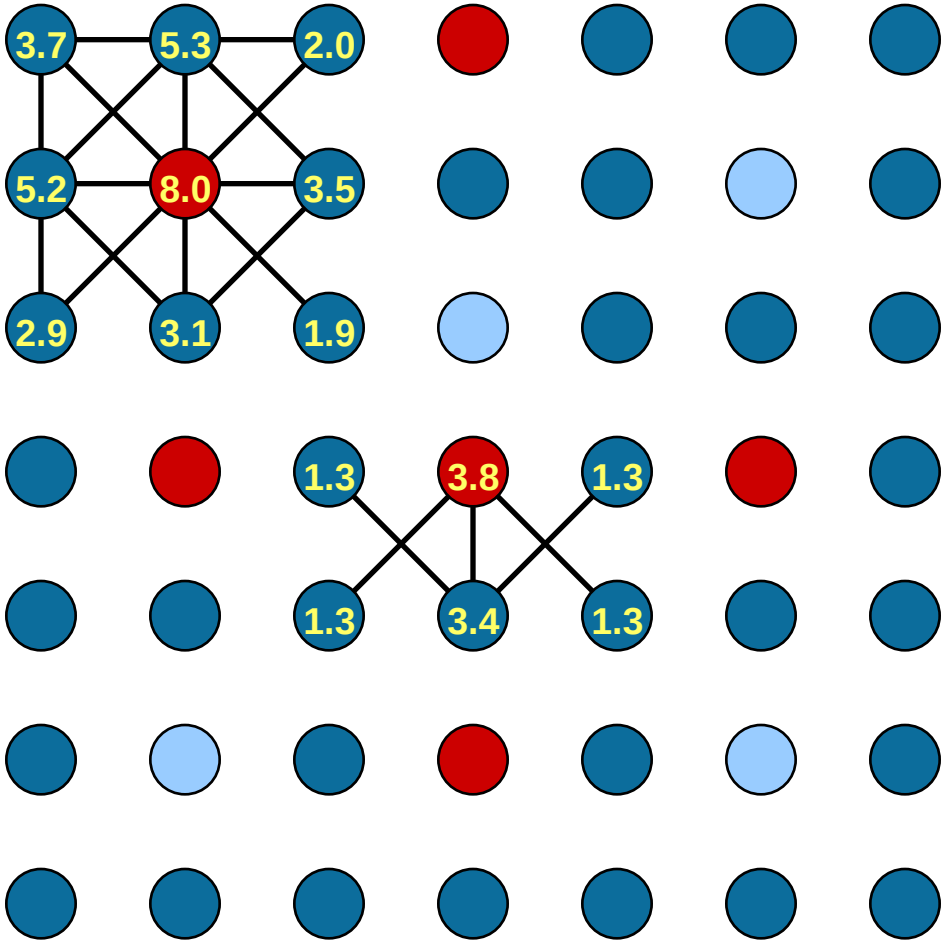
- ➔ **select C-pts with maximal measure locally**
- ➔ **remove neighbor edges**
- ➔ **update neighbor measures**

CLJP coarsening is fully parallel



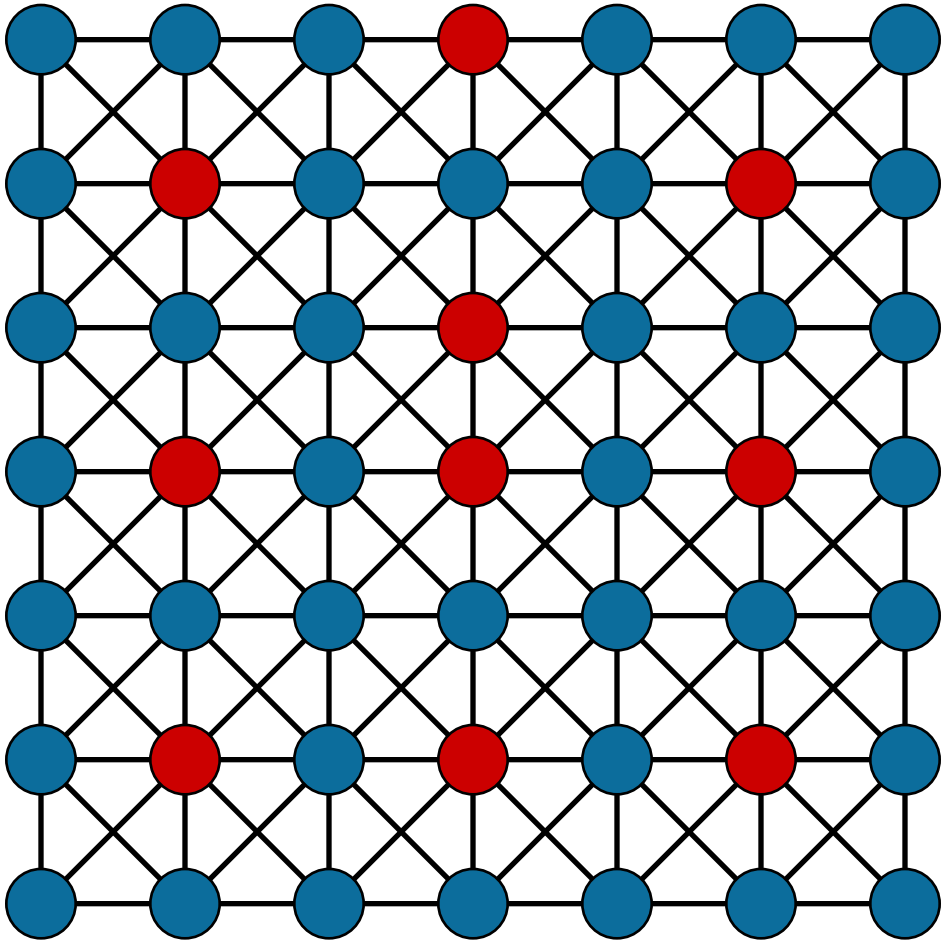
- select C-pts with maximal measure locally
- remove neighbor edges
- update neighbor measures

CLJP coarsening is fully parallel



- select C-pts with maximal measure locally
- remove neighbor edges
- update neighbor measures

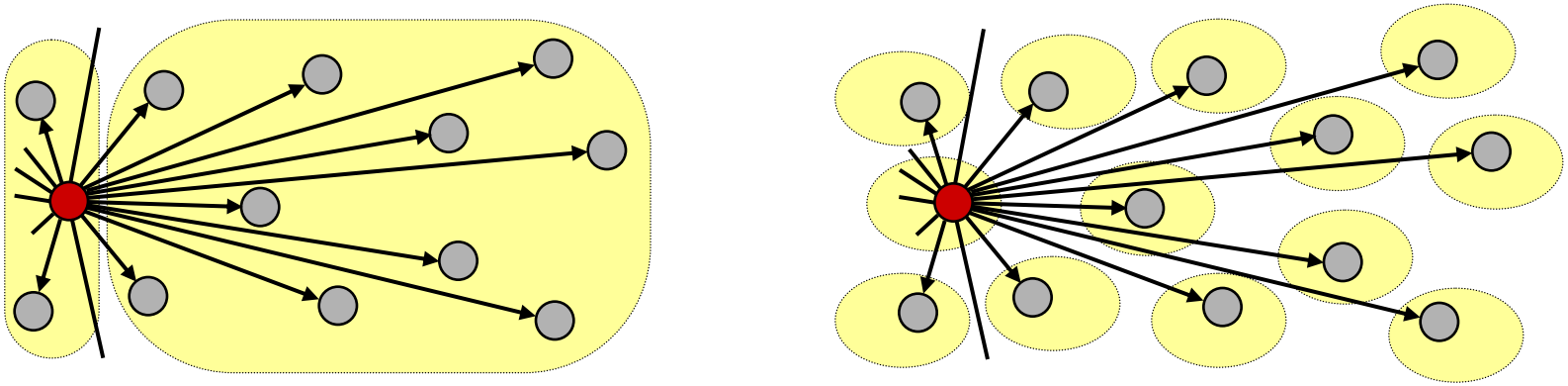
CLJP coarsening is fully parallel



- 10 C-points selected
- Standard AMG selects 9 C-points

Parallel coarse-grid selection in AMG can produce unwanted side effects

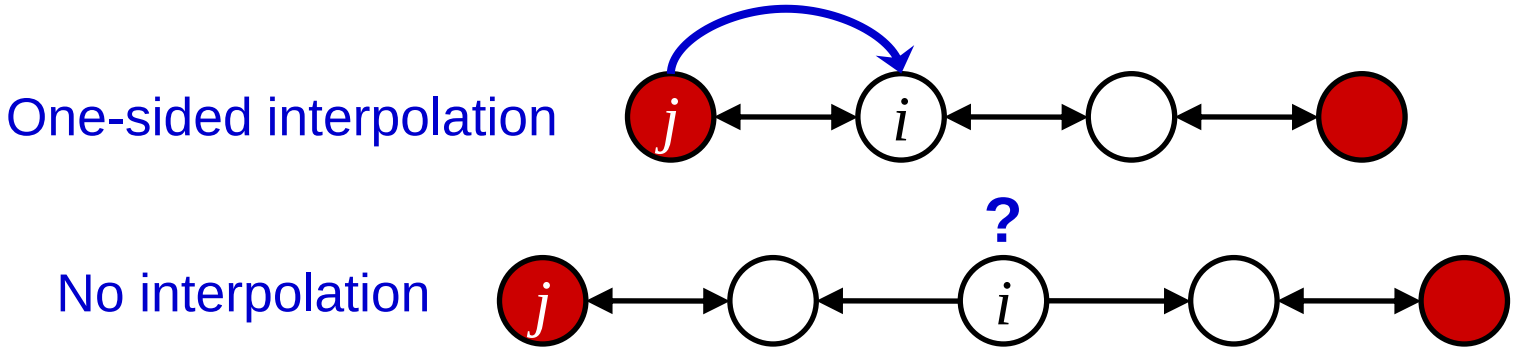
- Non-uniform grids can lead to increased operator complexity and poor convergence
- Operator “stencil growth” reduces parallel efficiency



- Currently no guaranteed ways to control complexity
- Can ameliorate with more aggressive coarsening
- Requires long-range interpolation approaches

C-AMG interpolation is not suitable for more aggressive coarsening

- PMIS is parallel and eliminates the second pass, which can lead to the following scenarios:



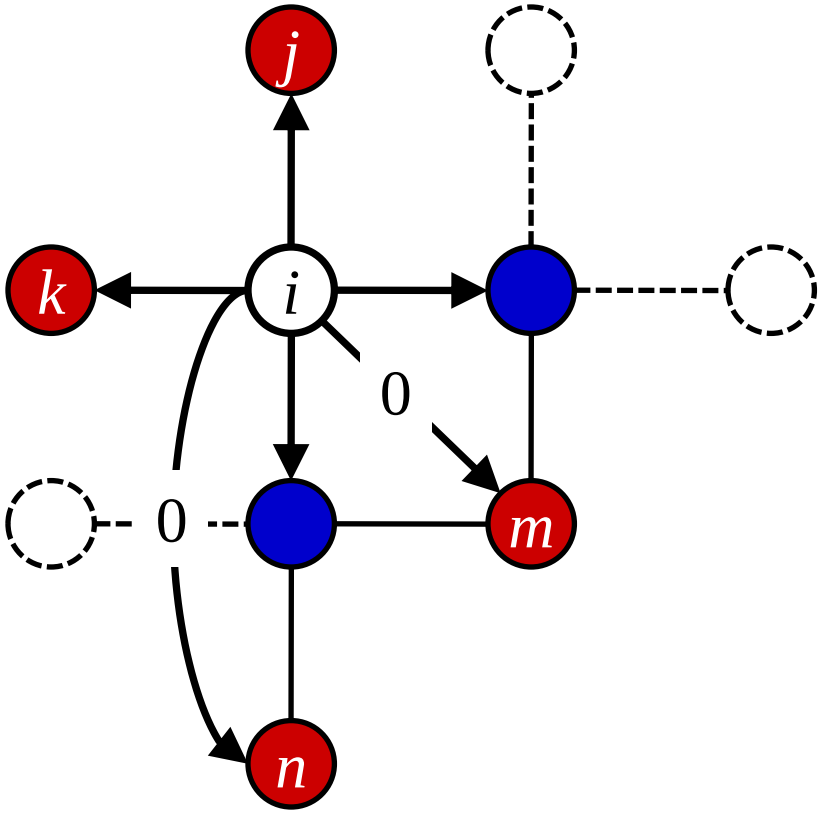
- Want above *i*-points to interpolate from both *C*-points
- Long-range (distance two) interpolation!

One possibility for long-range interpolation is extended interpolation

- C-AMG: $C_i = \{j,k\}$
- Long-range: $C_i = \{j,k,m,n\}$

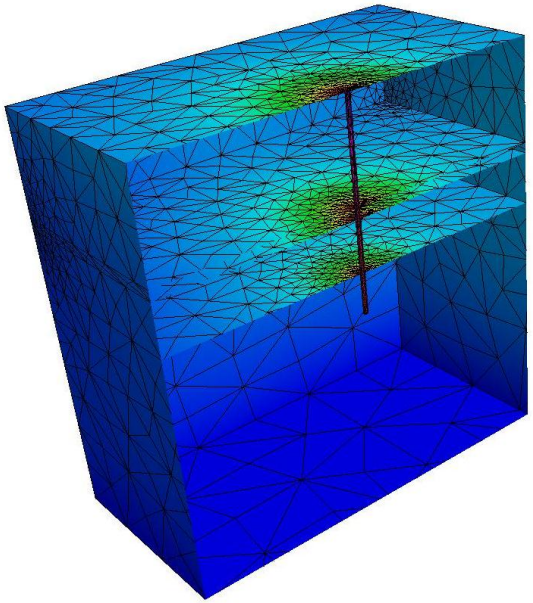
- Extended interpolation – apply C-AMG interpolation to an extended stencil

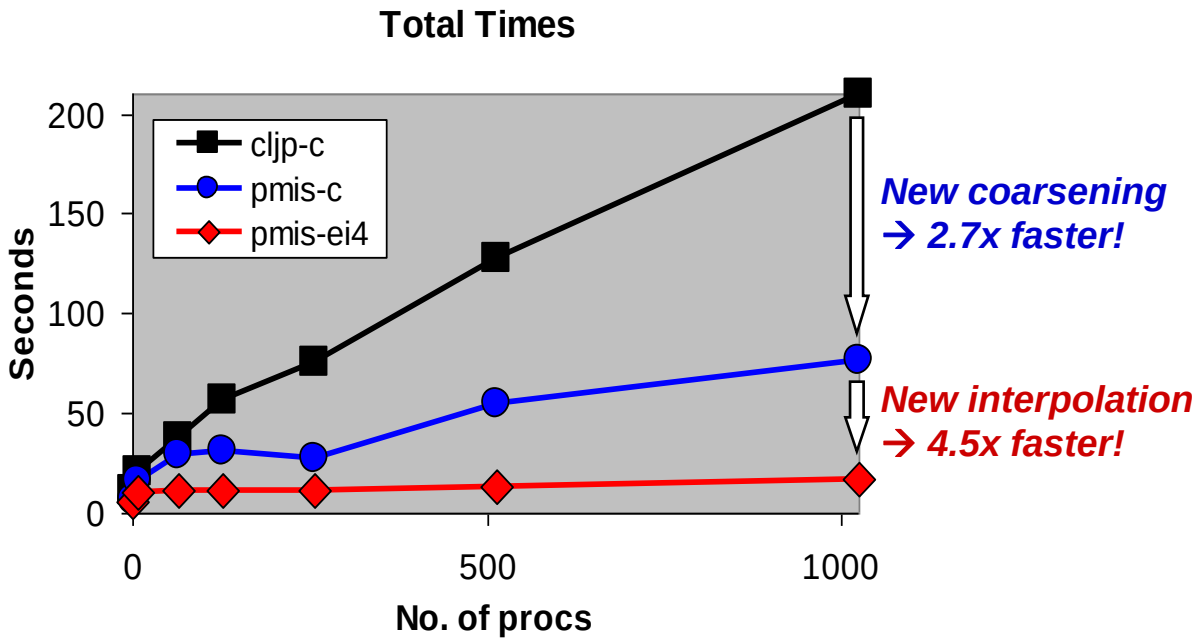
- Extended+i interpolation is the same as extended, but also collapses to point i
- Improves overall quality



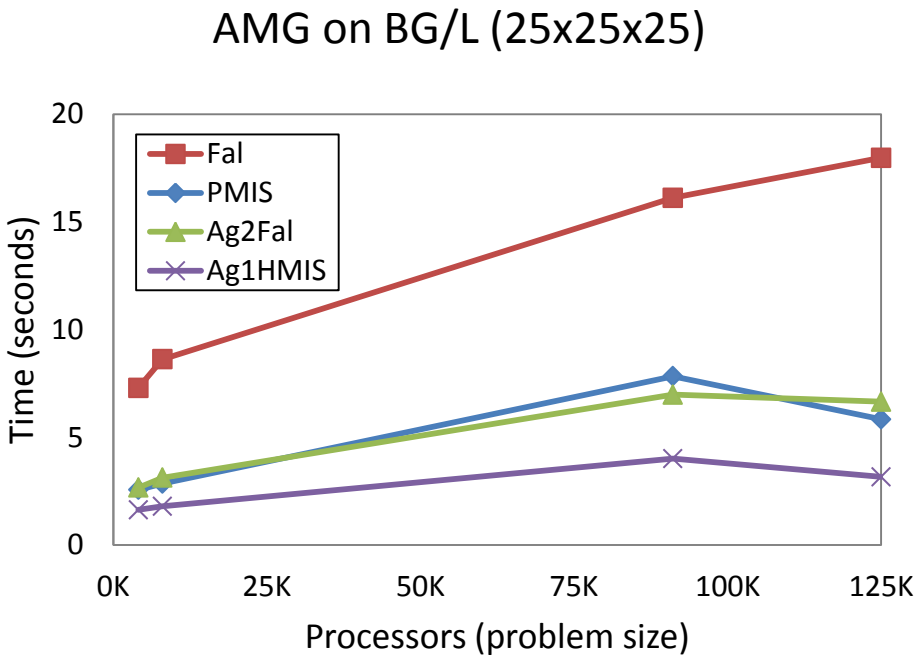
New parallel coarsening and long-range interpolation methods are improving scalability

- Unstructured 3D problem with material discontinuities
- About 90K unknowns per processor on MCR (Linux cluster)
- AMG - GMRES(10)





Parallel AMG in hypre now scales to 130K processors on BG/L … and beyond



- Largest problem above: 2B unknowns
- Largest problem to date: 26B unknowns on 98K processors of BG/L
- Most processors to date: 16B unknowns on 196K cores of Jaguar (Cray XT5 at ORNL)

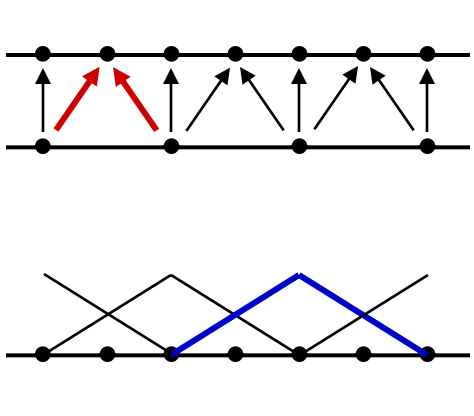
Smoothed Aggregation (SA)

(Vaněk, Mandel, Brezina)

SA views the prolongation operator columnwise, as a set of local basis functions

- 1D Laplacian example:

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



C-AMG – by rows, linear interpolation

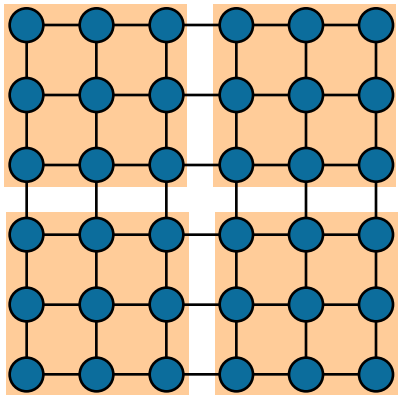
SA – by columns, hat functions

- Range(P) contains the (smooth) constant vector $\mathbf{1}$

- SA approach for building prolongation – decompose near null space into a basis with local support

SA builds interpolation by first chopping up a global basis, then smoothing it

- Tentative interpolation is constructed from “aggregates” (local QR factorization is used to orthonormalize)



$$\hat{P} \quad B_c \quad B$$

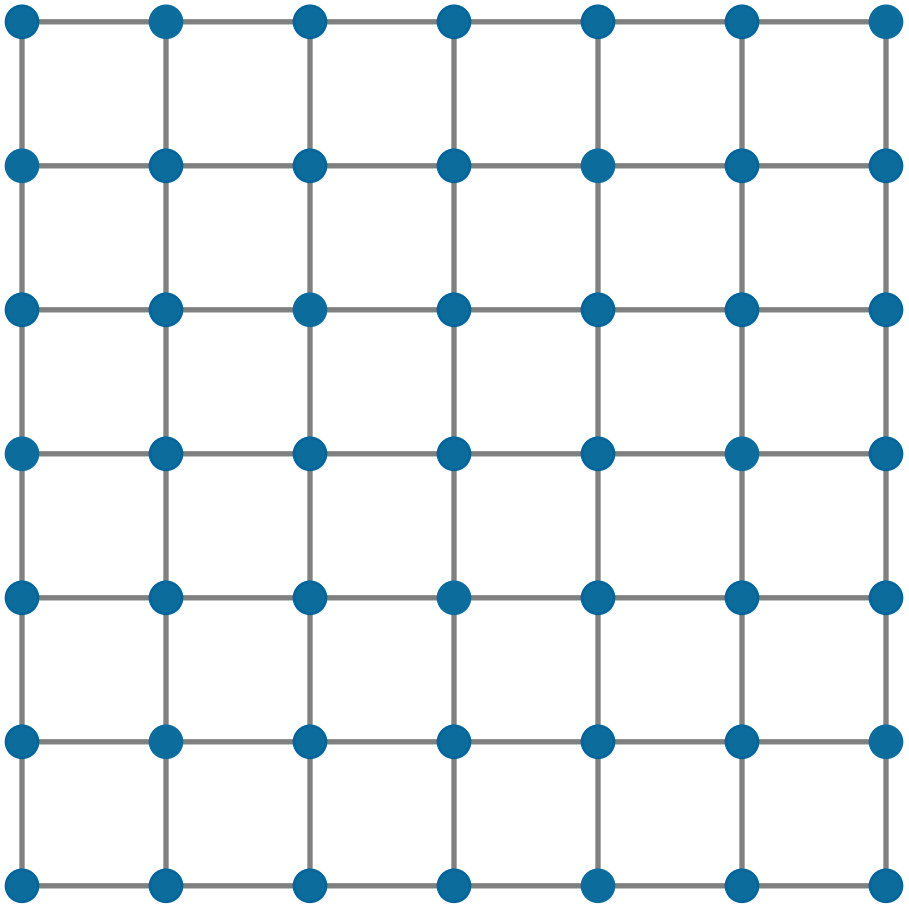
$$\begin{bmatrix} \\ \\ \\ \\ \\ \end{bmatrix} \begin{bmatrix} \\ \\ \\ \end{bmatrix} = \begin{bmatrix} \\ \\ \\ \\ \\ \end{bmatrix}$$

$$; \quad \hat{P}^T \hat{P} = I$$

- Smoothing adds basis overlap and improves approximation property

$$P = S\hat{P}$$

SA coarsening (5-pt Laplacian)



Phase 1:

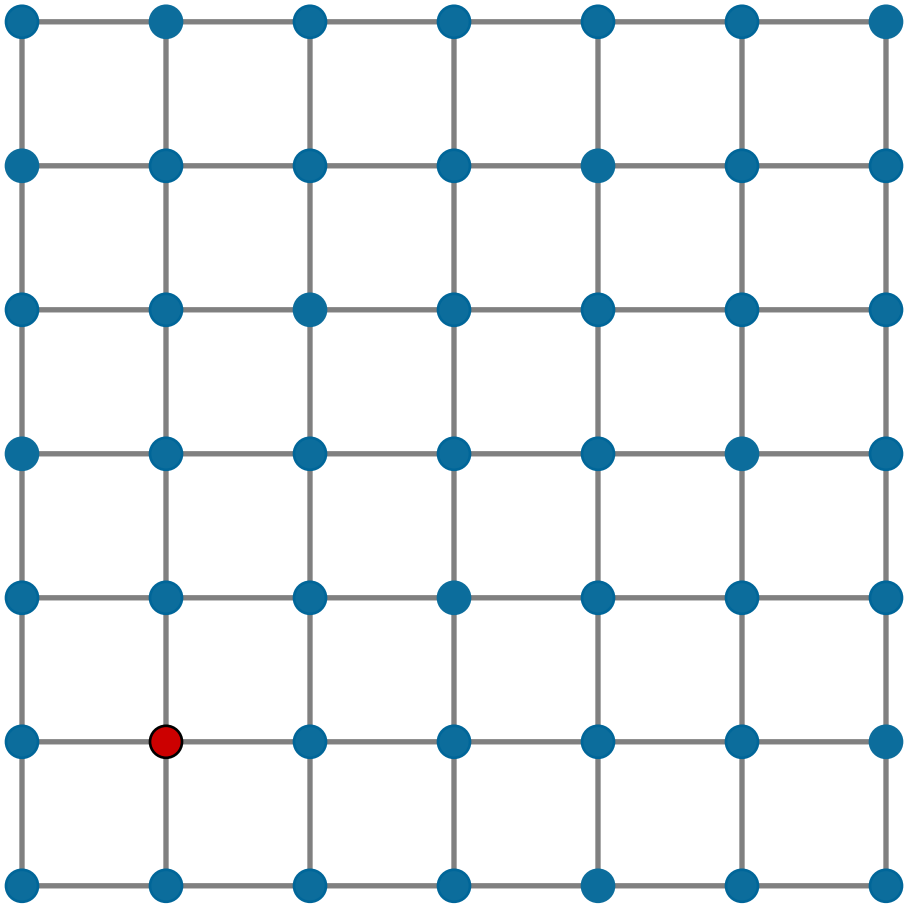
a) Pick root pt not adjacent to agg

b) Aggregate root and neighbors

Phase 2:

Move pts into nearby aggs or new aggs

SA coarsening (5-pt Laplacian)



Phase 1:

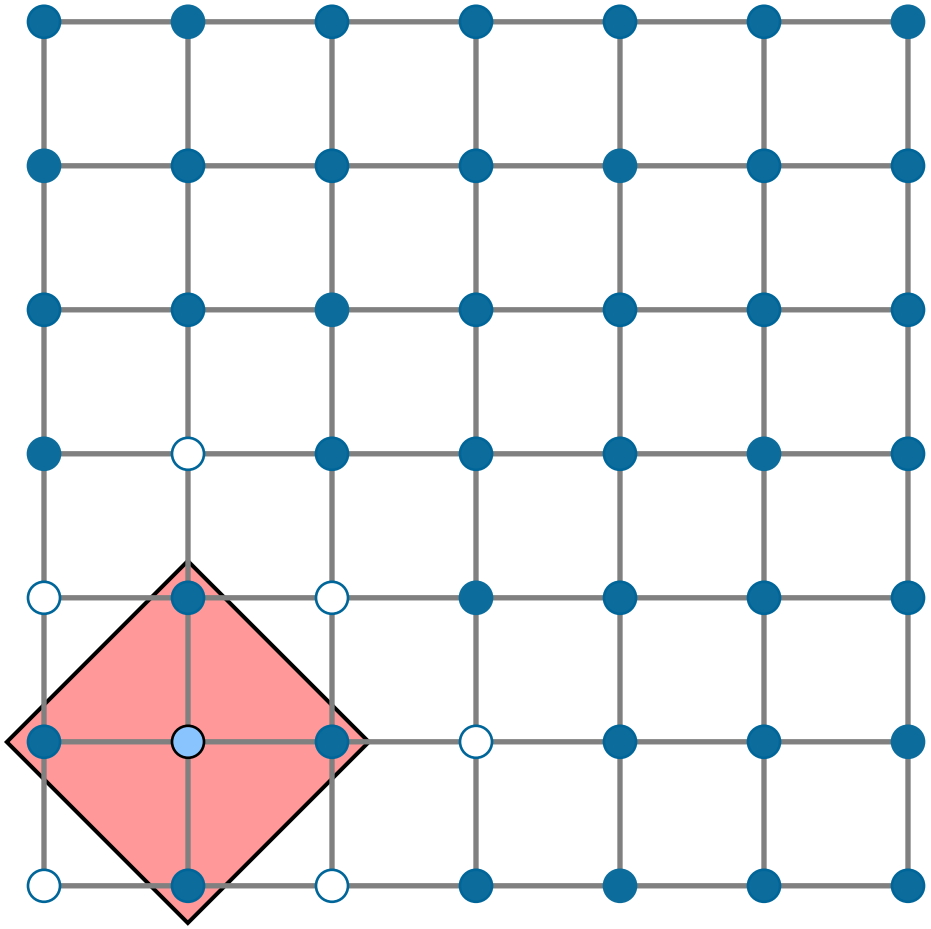
a) Pick root pt not adjacent to agg

b) Aggregate root and neighbors

Phase 2:

Move pts into nearby aggs or new aggs

SA coarsening (5-pt Laplacian)



Phase 1:

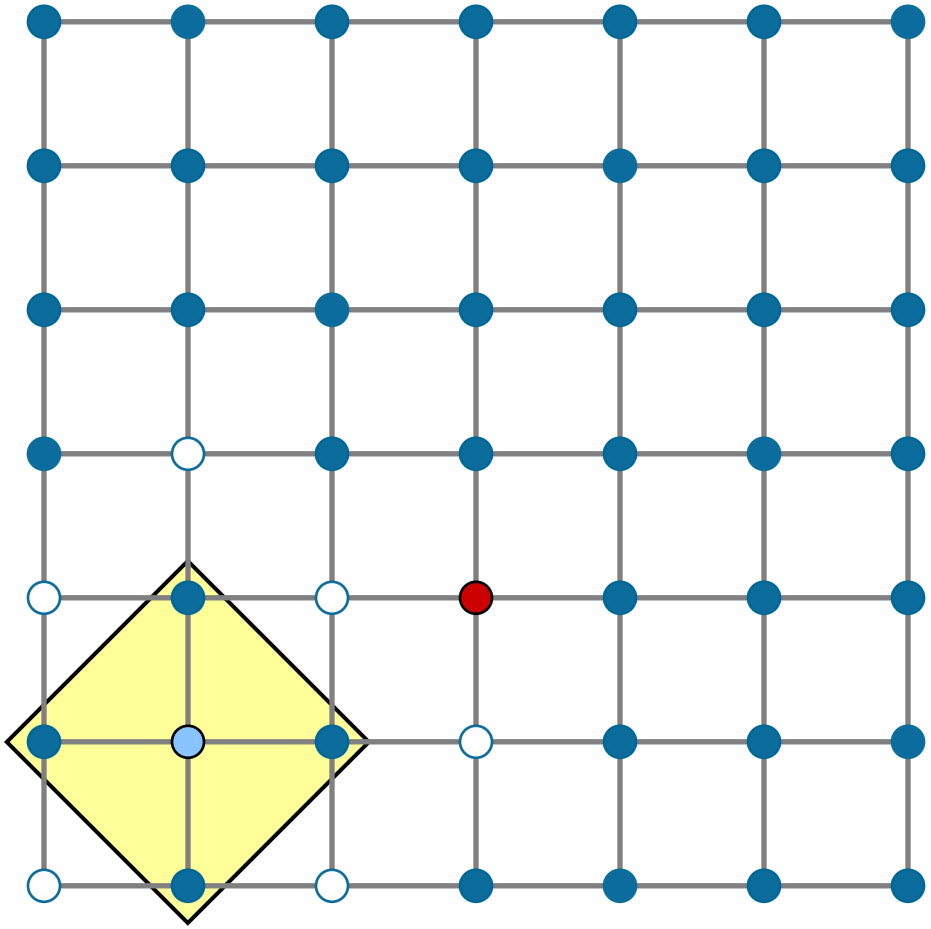
a) Pick root pt not adjacent to agg

b) Aggregate root and neighbors

Phase 2:

Move pts into nearby aggs or new aggs

SA coarsening (5-pt Laplacian)



Phase 1:

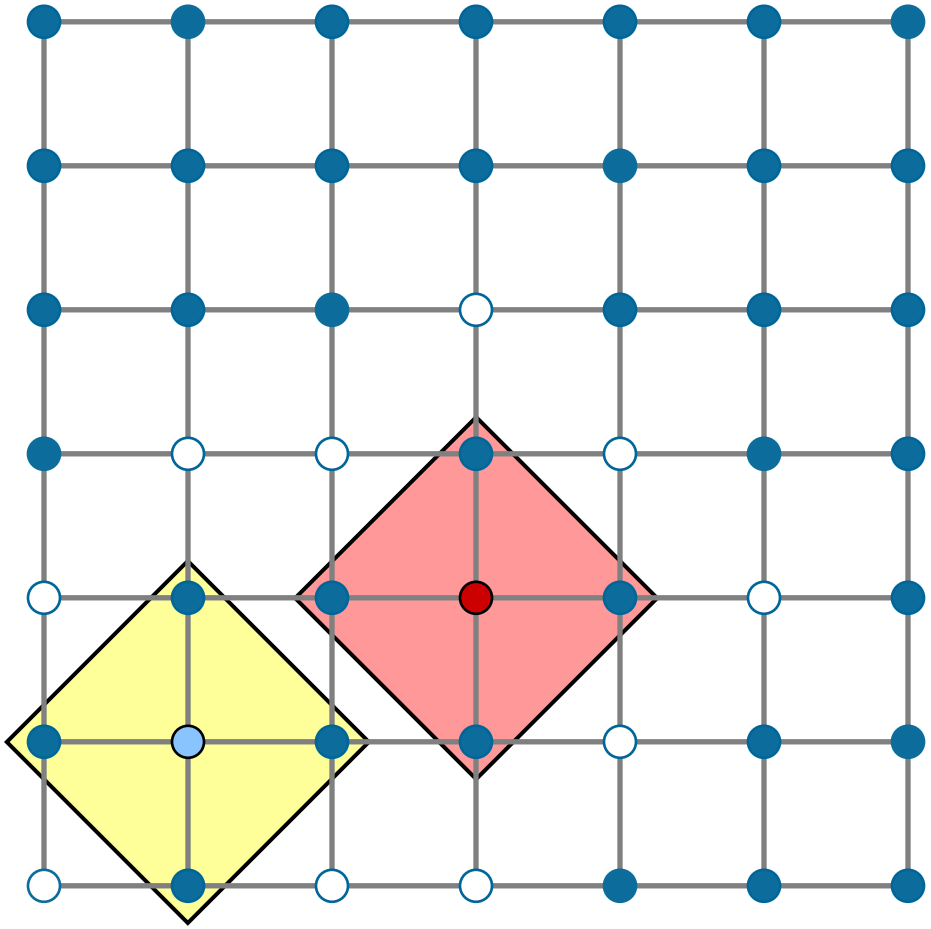
a) Pick root pt not adjacent to agg

b) Aggregate root and neighbors

Phase 2:

Move pts into nearby aggs or new aggs

SA coarsening (5-pt Laplacian)



Phase 1:

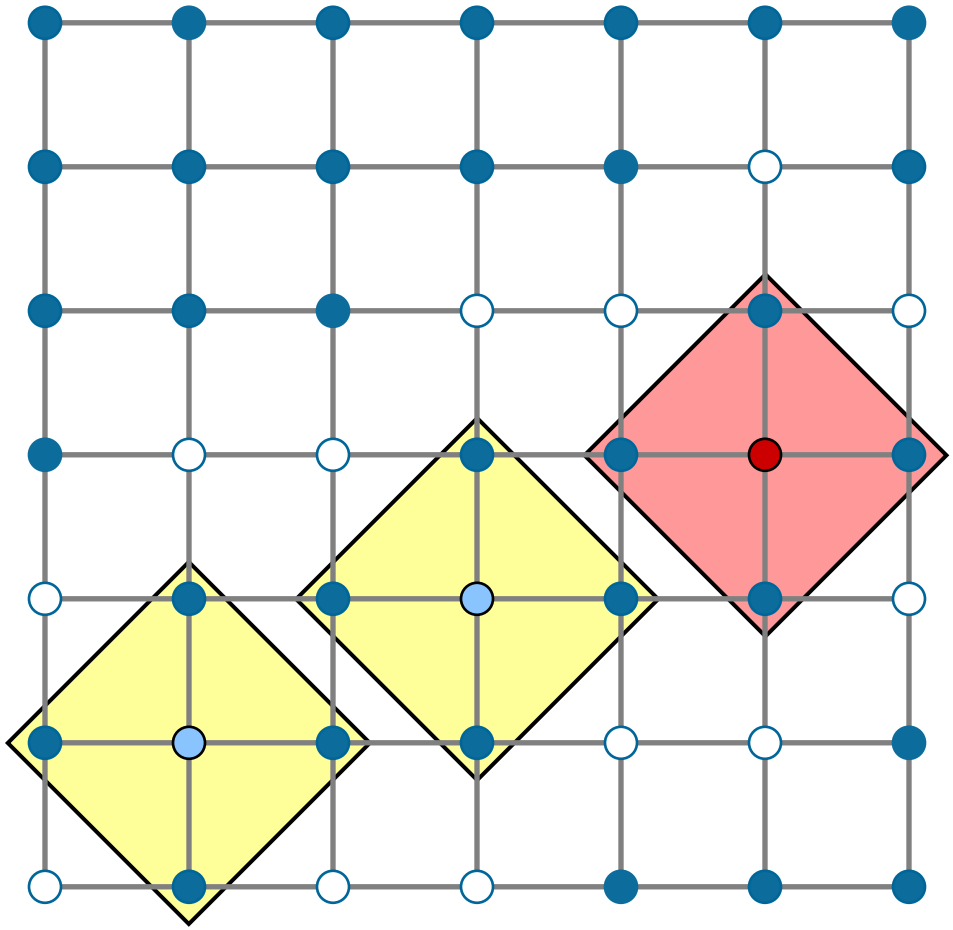
a) Pick root pt not adjacent to agg

b) Aggregate root and neighbors

Phase 2:

Move pts into nearby aggs or new aggs

SA coarsening (5-pt Laplacian)



Phase 1:

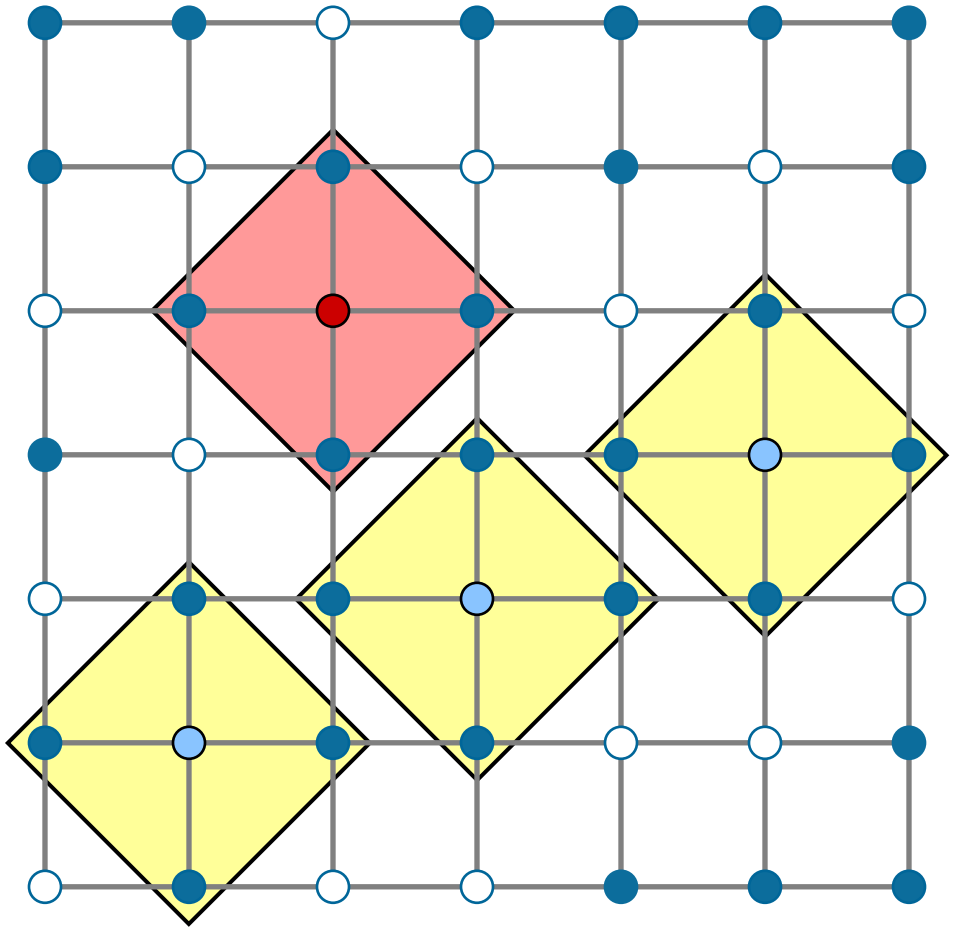
a) Pick root pt not adjacent to agg

b) Aggregate root and neighbors

Phase 2:

Move pts into nearby aggs or new aggs

SA coarsening (5-pt Laplacian)



Phase 1:

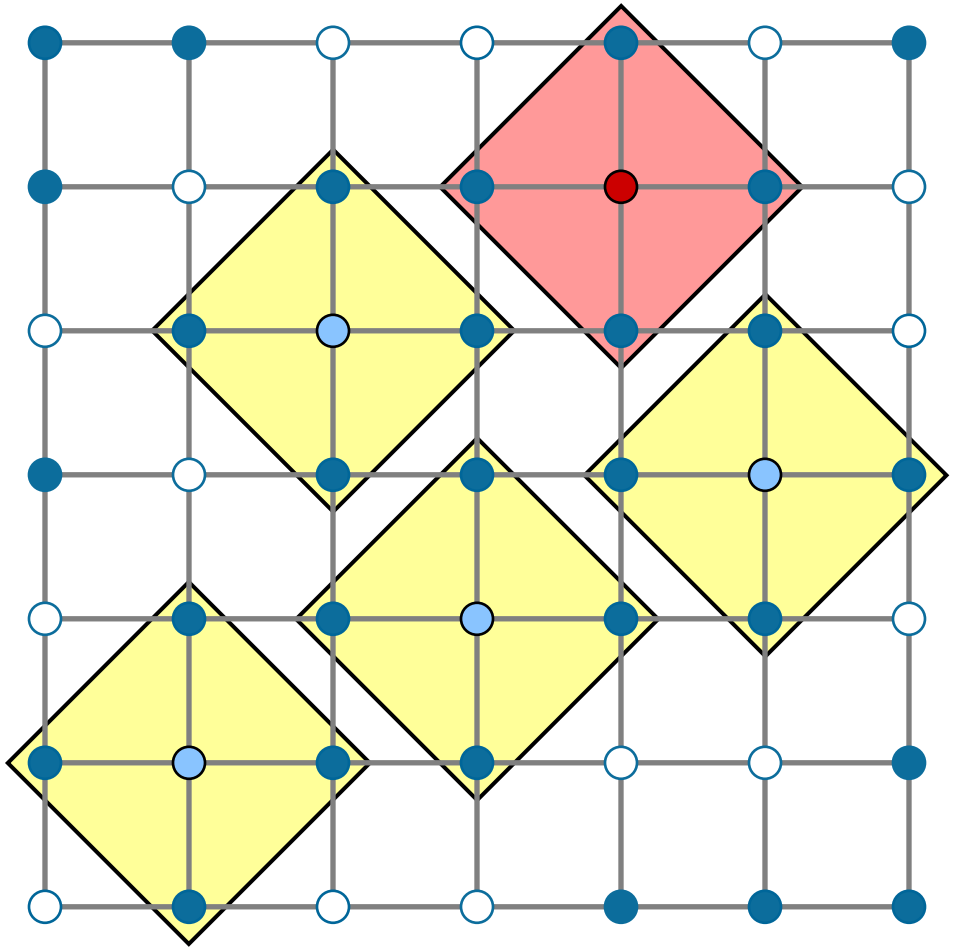
a) Pick root pt not adjacent to agg

b) Aggregate root and neighbors

Phase 2:

Move pts into nearby aggs or new aggs

SA coarsening (5-pt Laplacian)



Phase 1:

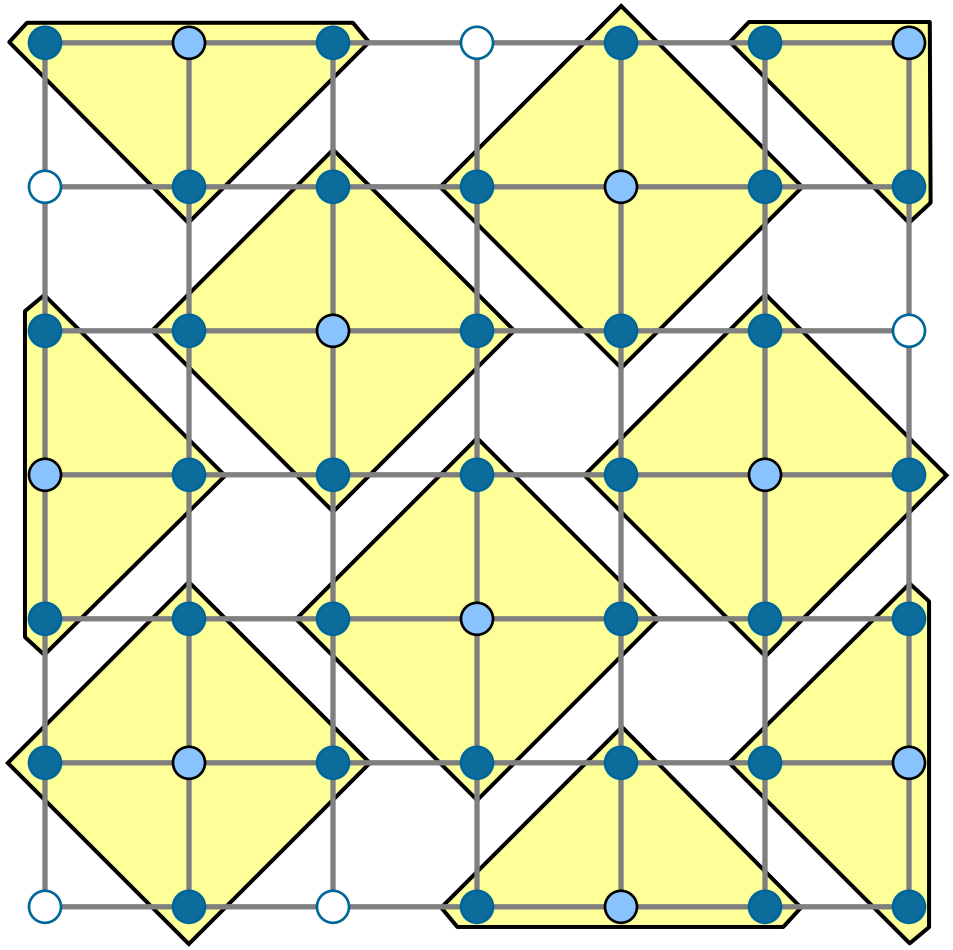
a) Pick root pt not adjacent to agg

b) Aggregate root and neighbors

Phase 2:

Move pts into nearby aggs or new aggs

SA coarsening (5-pt Laplacian)



Phase 1:

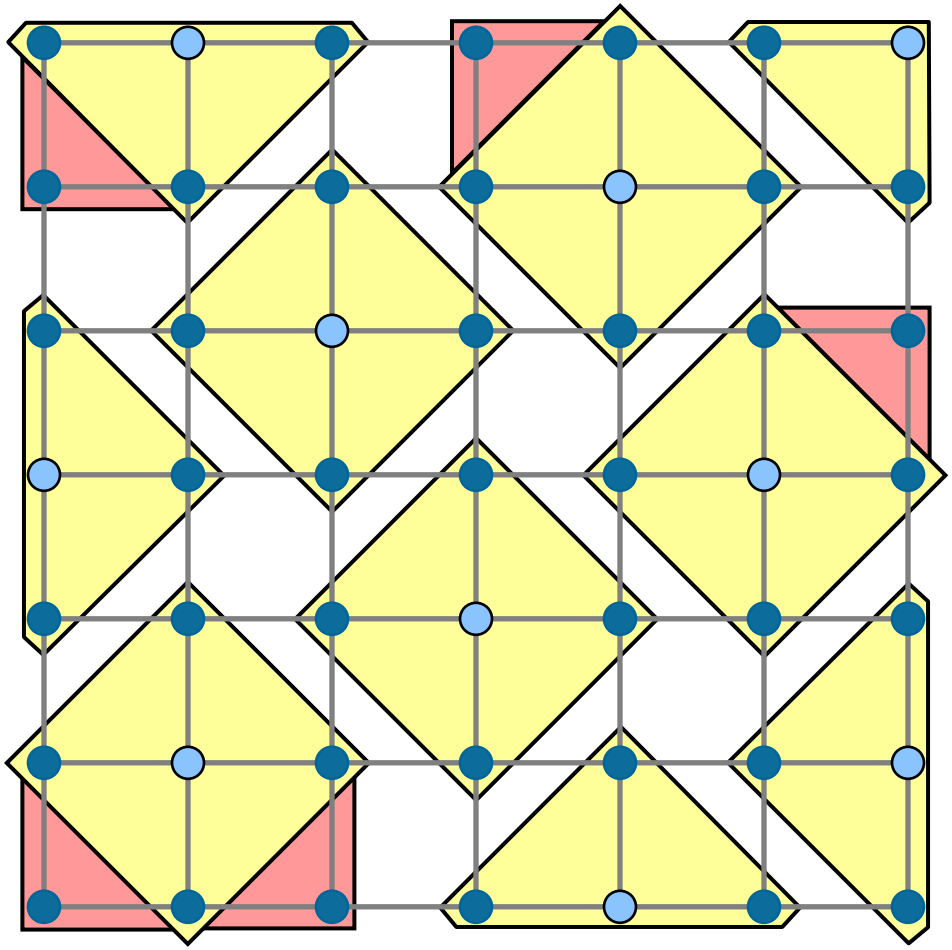
a) Pick root pt not adjacent to agg

b) Aggregate root and neighbors

Phase 2:

Move pts into nearby aggs or new aggs

SA coarsening (5-pt Laplacian)



Phase 1:

a) Pick root pt not adjacent to agg

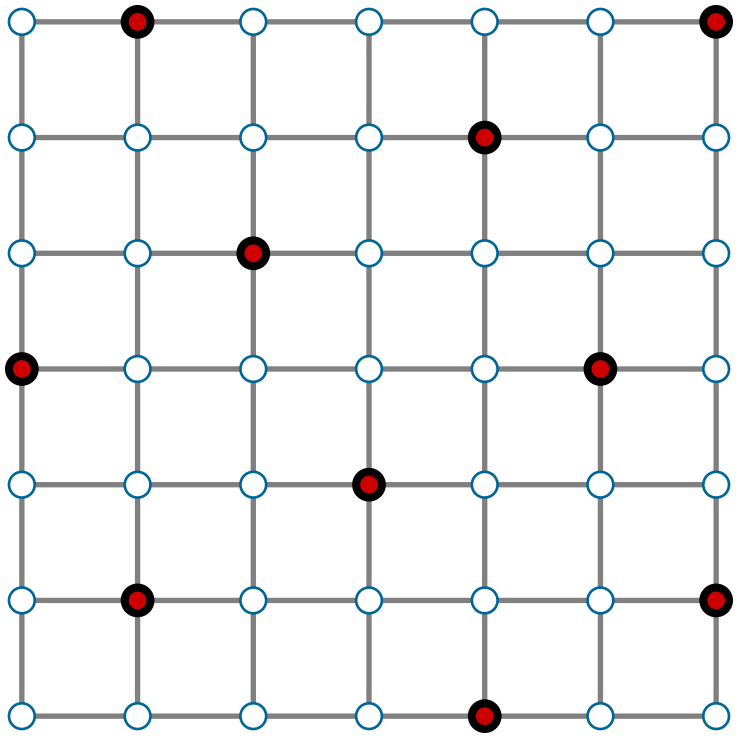
b) Aggregate root and neighbors

Phase 2:

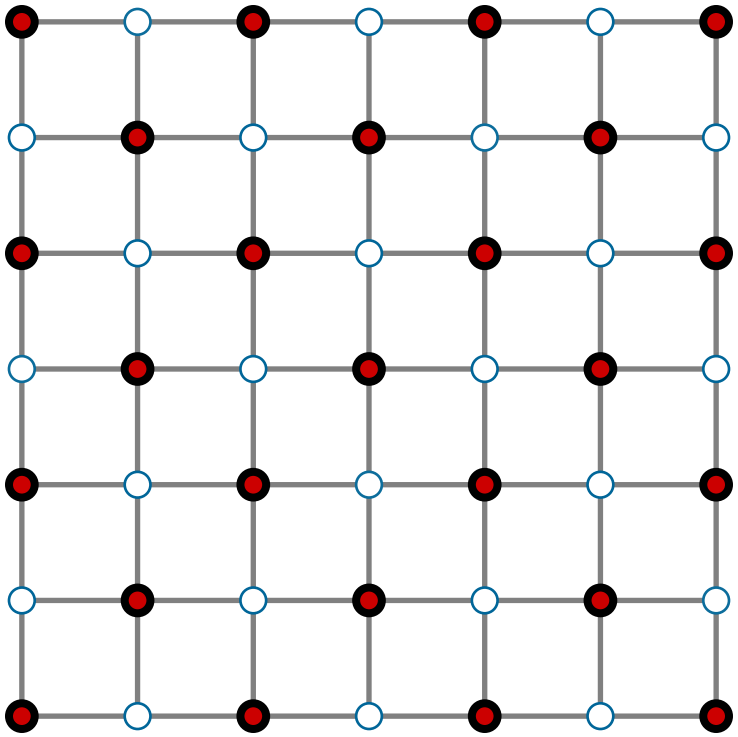
Move pts into nearby aggs or new aggs

SA coarsening is traditionally more aggressive than C-AMG coarsening (5-pt Laplacian example)

SA Seed Points (10)



C-AMG Grid (25)



Operator complexities are usually smaller, too

Additional comments on SA…

- Usual prolongator smoother is damped Jacobi
- Strength of connection is usually defined differently

$$|a_{ij}| > \theta \sqrt{|a_{ii} a_{jj}|}$$

- Special care must be taken for anisotropic problems to keep complexity low
  - Thresholded prolongator smoothing
  - Basis shifting approach
- Parallel SA coarsening has issues similar to C-AMG

AMG Theory
&
Compatible Relaxation

GAMG preliminaries…

- Consider solving $Au = f$, A SPD
- Consider smoothers with error propagation

$$e_{k+1} = (I - M^{-1}A)e_k$$

where we assume that $(M+M^T-A)$ is SPD (necessary and sufficient condition for convergence)

- Note: M may be symmetric or nonsymmetric
- Denote the symmetrized smoother operator by

$$\widetilde{M} = M(M^T + M - A)^{-1}M^T$$

that is, $(I - \widetilde{M}^{-1}A) = (I - M^{-T}A)(I - M^{-1}A)$

GAMG preliminaries continued…

- Let $P : \Re^{n_c} \rightarrow \Re^{n}$ be interpolation (prolongation)
- Let $R : \Re^{n} \rightarrow \Re^{n_c}$ be some "restriction" operator
  - Note that R is not the MG restriction operator
- Define s.t. $RP=I$ and PR is a projection onto $\text{range}(P)$

- For any SPD matrix X and any full-rank matrix B, denote the X-orthogonal projection onto $\text{range}(B)$ by

$$\pi_X(B) = B(B^T X B)^{-1} B^T X$$

- Define the two-grid multigrid error propagator by

$$E_{TG} = (I - M^{-1}A)(I - \pi_A(P))$$

GAMG two-grid theory splits construction of coarse-grid correction into two parts

- Theorem:

$$\|E_{TG}\|_A^2 \leq 1 - \frac{1}{K}; \qquad K = \sup_e \frac{\|(I - PR)e\|_{\widetilde{M}}^2}{\|e\|_A^2}$$

- Now, fix R so that it does not depend on P
  - Defines the coarse-grid variables, $u_c = Ru$
  - Example: R=[$0, I$] (P^T=[W^T, I]^T), i.e., subset of the fine grid
- Theorem:

$$K \leq \eta K_\star; \quad \eta = \|PR\|_A; \quad K_\star = \inf_P \sup_e \frac{\|(I - PR)e\|_{\widetilde{M}}^2}{\|e\|_A^2}$$

  - Small $K_\star$ insures coarse grid quality – use *CR*
  - Small η insures interpolation quality – necessary condition that does not depend on relaxation!

CR is an efficient method for measuring the quality of the set of coarse variables

- *CR* (Brandt, 2000) is a modified relaxation scheme that keeps the coarse-level variables, Ru, invariant
- Theorem: (fast convergence) good coarse grid)

$$K_\star \leq \left(\frac{\Delta^2}{2-\omega}\right)\frac{1}{1-\rho_{cr}}$$

$\Delta \geq 1$ measures the deviation of M from its symmetric part M_σ and $0 < \omega < 2$ is a kind of smoothing parameter

$$\Delta^2 = \left\| M_\sigma^{-1/2}\ M\ M_\sigma^{-1/2} \right\|^2; \quad \omega = \lambda_{\max}(M_\sigma^{-1}A)$$

- Must insure "good" constants
  - in particular, ω « 2

Several general *CR* methods

- Define S such that $\Re^n = \text{range}(S) \oplus \text{range}(R^T)$ and $RS = 0$
  - Example: R=[$0, I$]; S=[$I, 0$]^T; P^T=[W^T, I]^T

- Primary *CR* method – feasible for relaxation based on matrix splittings, where M is explicitly available

$$I - (S^T M S)^{-1}(S^T A S)$$

- Habituated *CR* – not as sharp, but always computable

$$S^T(I - M^{-1}A)S; \quad S^T S = I$$

Sharp Theory insightful for improving *CR* prediction

- GAMG theory

$$\|E_{TG}\|_A^2 \leq 1 - \frac{1}{K}; \qquad K = \sup_e \frac{\|(I - PR)e\|_{\widetilde{M}}^2}{\|e\|_A^2}$$

- Sharp theory

$$\|E_{TG}\|_A^2 = 1 - \frac{1}{K_\sharp}; \qquad K_\sharp = \sup_e \frac{\left\|(I - \pi_{\widetilde{M}}(P))e\right\|_{\widetilde{M}}^2}{\|e\|_A^2}$$

- Differ only in form of the projection
- Careful comparison → optimal R is given by

$$R = P^T \widetilde{M}$$

- But, we don't have P yet (we're trying to build it)

AMG and ILU

Ideal interpolation

- Recall 2-level theory:

$$\|E_{TG}\|_A^2 \leq 1 - \frac{1}{K}; \qquad K = \sup_e \frac{\|(I - PR)e\|_{\widetilde{M}}^2}{\|e\|_A^2}$$

- Consider R=[0, I_c], P^T=[W^T, I_c]^T case
- "Ideal" P is given by

$$P_\star = \arg\min_P \sup_e \frac{\|(I - PR)e\|_{\widetilde{M}}^2}{\|e\|_A^2} = \begin{bmatrix} -A_{ff}^{-1} A_{fc} \\ I_c \end{bmatrix}$$

- Not a practical choice in general

AMG and ILU

- Can factor A as follows

$$A = \begin{bmatrix} A_{ff} & A_{fc} \\ A_{cf} & A_{cc} \end{bmatrix} = \begin{bmatrix} I & 0 \\ (A_{ff}^{-1}A_{fc})^T & I \end{bmatrix} \begin{bmatrix} A_{ff} & 0 \\ 0 & S_{cc} \end{bmatrix} \begin{bmatrix} I & (A_{ff}^{-1}A_{fc}) \\ 0 & I \end{bmatrix}$$

$$\begin{aligned} A^{-1} &= \begin{bmatrix} I & -(A_{ff}^{-1}A_{fc}) \\ 0 & I \end{bmatrix} \begin{bmatrix} A_{ff}^{-1} & 0 \\ 0 & S_{cc}^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ -(A_{ff}^{-1}A_{fc})^T & I \end{bmatrix} \\ &= \underbrace{S(S^TAS)^{-1}S^T}_{A_{ff}} + P_\star \underbrace{(P_\star^TAP_\star)^{-1}}_{S_{cc}} P_\star^T \end{aligned}$$

- Thinking of ILU, write the error propagator

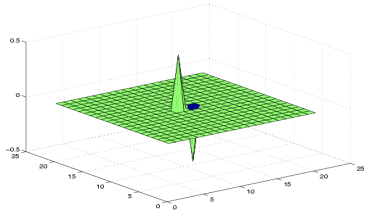
$$(I-(A^{-1})A) = (\underbrace{I-S(S^TAS)^{-1}S^TA}_{\text{F-relaxation}})(\underbrace{I-P_\star(P_\star^TAP_\star)^{-1}P_\star^TA}_{\text{Coarse-grid correction}})$$

AMG for Electromagnetic Problems

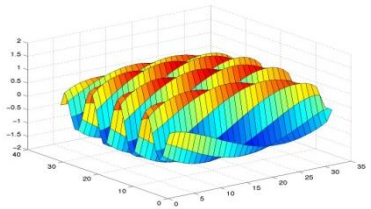
(see Kolev Poster on Monday)

Electromagnetic (EM) problems have huge oscillatory near null spaces

- Definite Maxwell, Indefinite Maxwell, Helmholtz
- Require specialized smoothers and coarse grids



Local: specialized relaxation
(Definite / Indefinite Maxwell)



Global: specialized coarse grids
(Helmholtz, Indefinite Maxwell)

- Definite Maxwell, Nédélec edge FEM discretization

$$\nabla \times \alpha \nabla \times \boldsymbol{E} + \beta \boldsymbol{E} = f \quad \alpha, \beta > 0$$

- Near null-space characterized by gradients

$$\nabla \times (\nabla p_h) = 0$$

Geometric multigrid for definite Maxwell

- Helmholtz decomposition

$$\boldsymbol{E}_h = \boldsymbol{v}_h + \nabla p_h$$

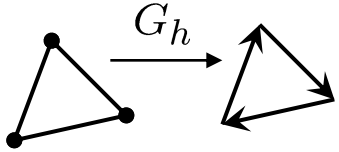
divergence-free curl-free

- Smooth both components (Hiptmair, SINUM 1998)

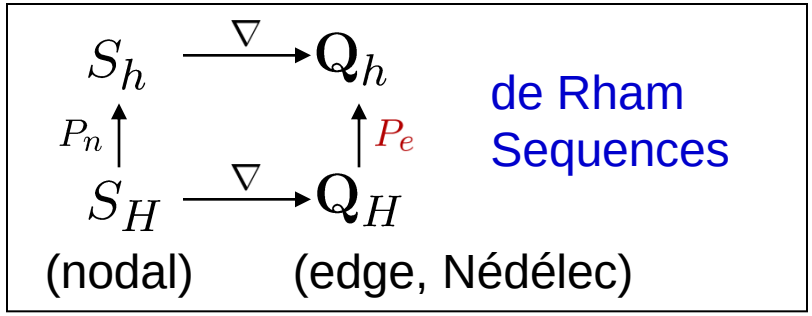
$$R_h = R_{e,h} + G_h R_{v,h} G_h^T$$

Point smoother for A_h Point smoother for $G_h^T A_h G_h$

Discrete Gradient



- Block smoother (Arnold, Falk, Winther, Num. Math. 2000)
- Natural FE interpolation
- Difficulties extending to
  - unstructured meshes
  - variable coefficients



Auxiliary-space Maxwell solver (AMS) utilizes a new decomposition

- Based on Hiptmair, Xu (2006)

$$\boldsymbol{E}_h = \boldsymbol{v}_h + \nabla p_h + \boldsymbol{\Pi}_h \boldsymbol{z}_h$$

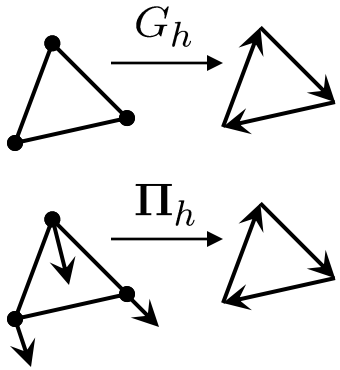
- Define preconditioner based on nodal solvers

$$B_h = R_h + G_h B_{v,h} G_h^T + \boldsymbol{\Pi}_h \boldsymbol{B}_{\boldsymbol{v},h} \boldsymbol{\Pi}_h^T$$

Point smoother for A_h

AMG solver for $G_h^T A_h G_h$

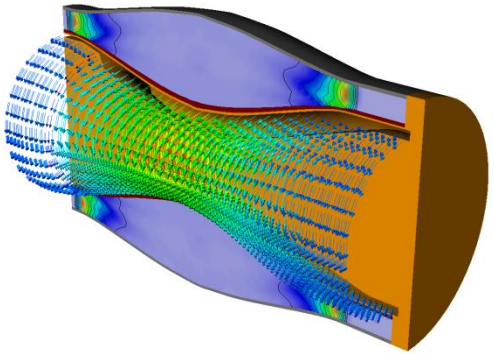
AMG solver for $\boldsymbol{\Pi}_h^T A_h \boldsymbol{\Pi}_h$

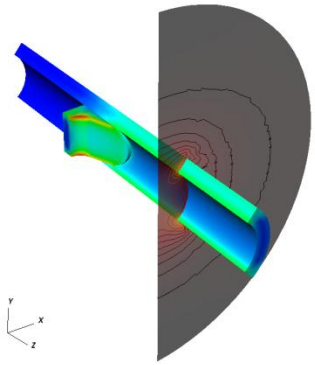


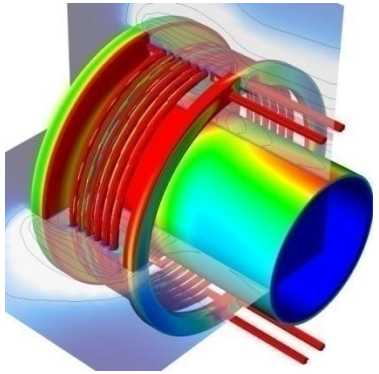
- User provides A, G_h and vertex coordinates
- Fast computation of Π_h (~ 3 mat-vec multiplies)
- AMS is a variational form of Hiptmair-Xu

Auxiliary-space Maxwell Solver (AMS) is improving solve times by up to 25x for some EM problems

- Hiptmair-Xu / AMS are the first provably scalable solvers for EM on unstructured grids
- Employs BoomerAMG
- Highly robust
  - Materials with widely varying electromagnetic properties
  - Unstructured grids
- Example: 1.2B unknowns on 1.9K processors took 355s (23 iterations)



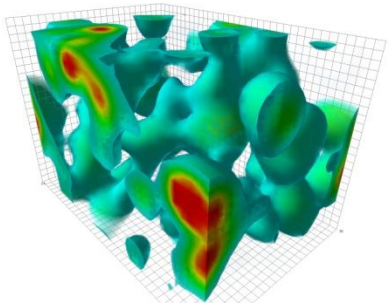




Adaptive AMG

Adaptive *AMG* is well-suited for QCD

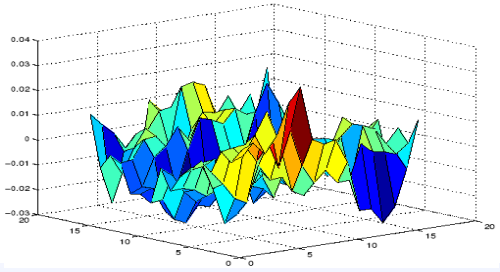
- Quantum Chromodynamics (QCD) is the theory of strong forces in the Standard Model of particle physics



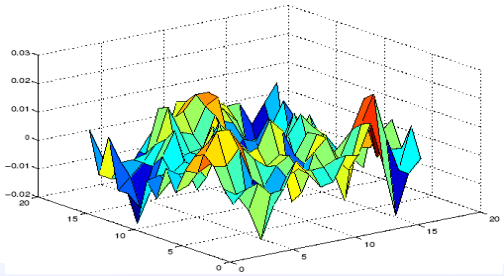
- Scalable solvers for the Dirac equations have been elusive until recently

- Challenges:
  - The system is complex and indefinite
  - The system can be extremely ill-conditioned
  - Near null space is unknown and oscillatory!

Real part



Imaginary part



Adaptive *AMG* idea: use the method to improve the method

- Requires no a-priori knowledge of the near null space
- Idea: uncover *representatives* of slowly-converging error by applying the "current method" to $Ax = 0$, then use these to adapt (improve) the method

- Achi Brandt's *Bootstrap AMG* is an adaptive method
- PCG can be viewed as an adaptive method
  - Not optimal because it uses a global view
  - The key is to view representatives locally

- We developed 2 methods: αAMG and αSA (SISC pubs)

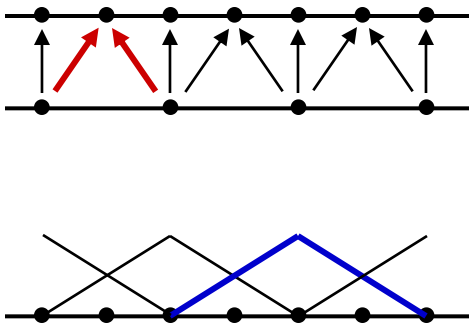
To build effective interpolation, it is important to interpret the near null space in a local way

- (2-level) Coarse-grid correction is a projection

$$(I - P(P^T A P)^{-1} P^T A)e$$

- Better to break up near null space into a local basis

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



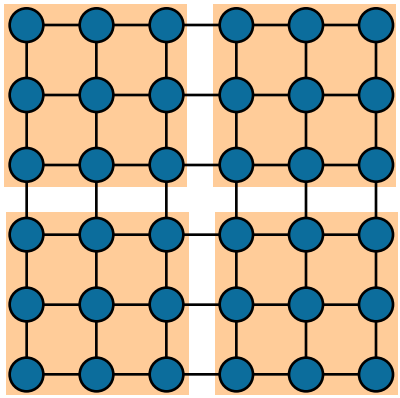
Deflation – not optimal

Multigrid – optimal

- Get full approximation property (low-frequency Fourier modes in this example)

SA builds interpolation by first chopping up a global basis, then smoothing it

- Tentative interpolation is constructed from “aggregates” (local QR factorization is used to orthonormalize)



$$\hat{P} \quad B_c \quad B$$

$$\begin{bmatrix} \\ \\ \\ \\ \\ \end{bmatrix} \begin{bmatrix} \\ \\ \\ \end{bmatrix} = \begin{bmatrix} \\ \\ \\ \\ \\ \end{bmatrix}; \quad \hat{P}^T \hat{P} = I$$

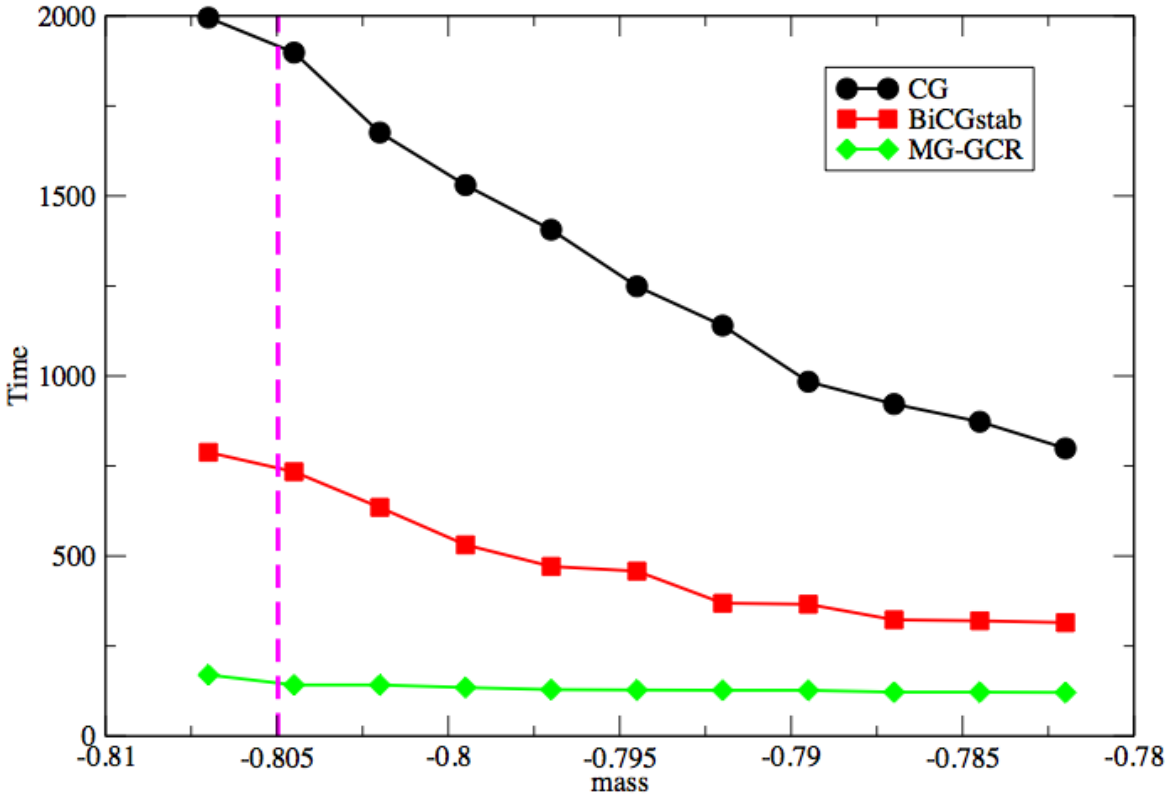
- Smoothing adds basis overlap and improves approximation property

$$P = S\hat{P}$$

Adaptive smoothed aggregation (αSA) automatically builds the global basis for SA

- Generate the basis one vector at a time
  - Start with relaxation on $Au=0$ → u_1 → αSA(u_1)
  - Use αSA(u_1) on $Au=0$ → u_2 → αSA(u_1,u_2)
  - Etc., until we have a good method
- Setup is expensive, but is amortized over many RHS's
- Published in 2004, highlighted in SIAM Review in 2005
  - Brezina, Falgout, MacLachlan, Manteuffel, McCormick, and Ruge, "Adaptive smoothed aggregation (αSA)," *SIAM J. Sci. Comput.* (2004)
- Successfully applied to 2D QED
  - Brannick, Brezina, Keyes, Livne, Livshits, MacLachlan, Manteuffel, McCormick, Ruge, and Zikatanov, "Adaptive smoothed aggregation in lattice QCD," Springer (2006)

4D Wilson-Dirac Results: D-MG shows no critical slowing down (Time)



- Parameters: $N=16^3x32$, $\beta=6.0$, $m_{crit} = -0.8049$
- D-MG Parameters: 4^4x3x2 blocking, 3 levels, W(2,2,4) cycle, $N_v = 20$, setup run at m_{crit}

Summary

- Multigrid methods are optimal and have good scaling potential

- AMG is based primarily on matrix entries
- In practice, some additional properties of the underlying system are assumed (near null space)
- Adaptive AMG uncovers near null space information

- AMG can solve a large class of problems and can scale to BG/L-class machines
- Parallel computing imposes additional restrictions on AMG algorithmic development

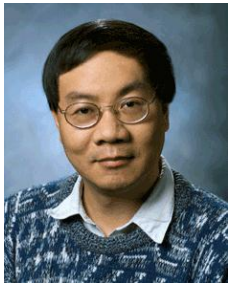
- Still many outstanding research questions

The Scalable Linear Solvers Team

Allison Baker

Rob Falgout

Tzanio Kolev



Charles Tong

Panayot Vassilevski

Ulrike Yang

See http://www.llnl.gov/casc/linear_solvers for publications, presentations, and software (hypre)

Former

- Chuck Baldwin
- Guillermo Castilla
- Edmond Chow
- Andy Cleary
- Noah Elliott
- Van Henson
- Ellen Hill
- David Hysom
- Jim Jones
- Mike Lambert
- Barry Lee
- Jeff Painter
- Tom Treadway
- Deborah Walker

Some of our collaborators

- CU Boulder – Manteuffel, McCormick, Ruge, Brezina
- Penn State – Xu, Zikatanov, Brannick
- Texas A&M – Lazarov, Pasciak
- UCSD – Bank
- UCLA – Brandt
- Ball State – Livshits
- Tufts – MacLachlan
- Technion – Yavneh
- Fraunhofer – Stüben
- …

Some References

- Introductory
  - A Multigrid Tutorial, W.L. Briggs, V.E. Henson, and S.F. McCormick, SIAM (2000)
  - Why Multigrid Methods Are So Efficient, I. Yavneh, *Computing in Science and Engineering*, 8 (2006), pp. 12–22
  - Introduction to Algebraic Multigrid, R.D. Falgout, *Computing in Science and Engineering*, 8 (2006), pp. 24–33
- Comprehensive
  - Multigrid, U. Trottenberg, C. Oosterlee, and A. Schüller, Academic Press (2001)
- Classical
  - Multi-level Adaptive Solutions to Boundary-Value Problems, A. Brandt, *Math. Comput.*, 31 (1977), pp. 333–390
  - Multigrid Methods, W. Hackbusch and U. Trottenberg, Eds., Springer (1982)
  - Multigrid Techniques: 1984 Guide With Applications to Fluid Dynamics, A. Brandt, GMD-Studie 85, Sankt Augustin, West Germany (1984)
  - Multi-Grid Methods and Applications, W. Hackbusch, Springer (1985)

Thank You!

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.