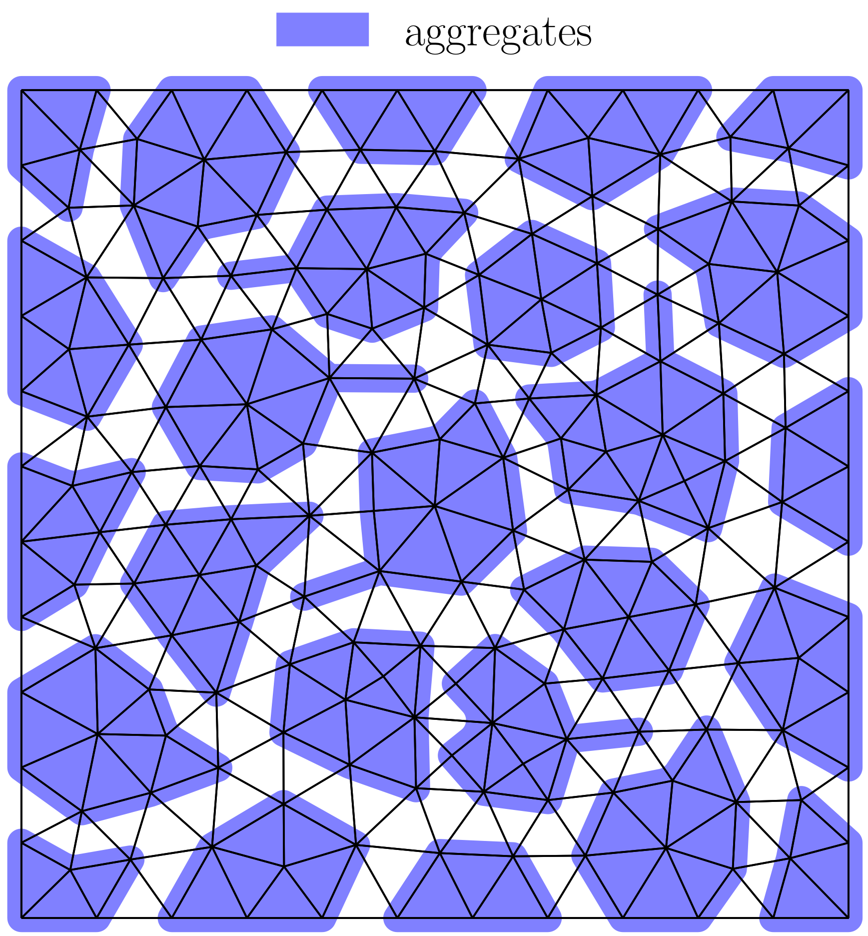
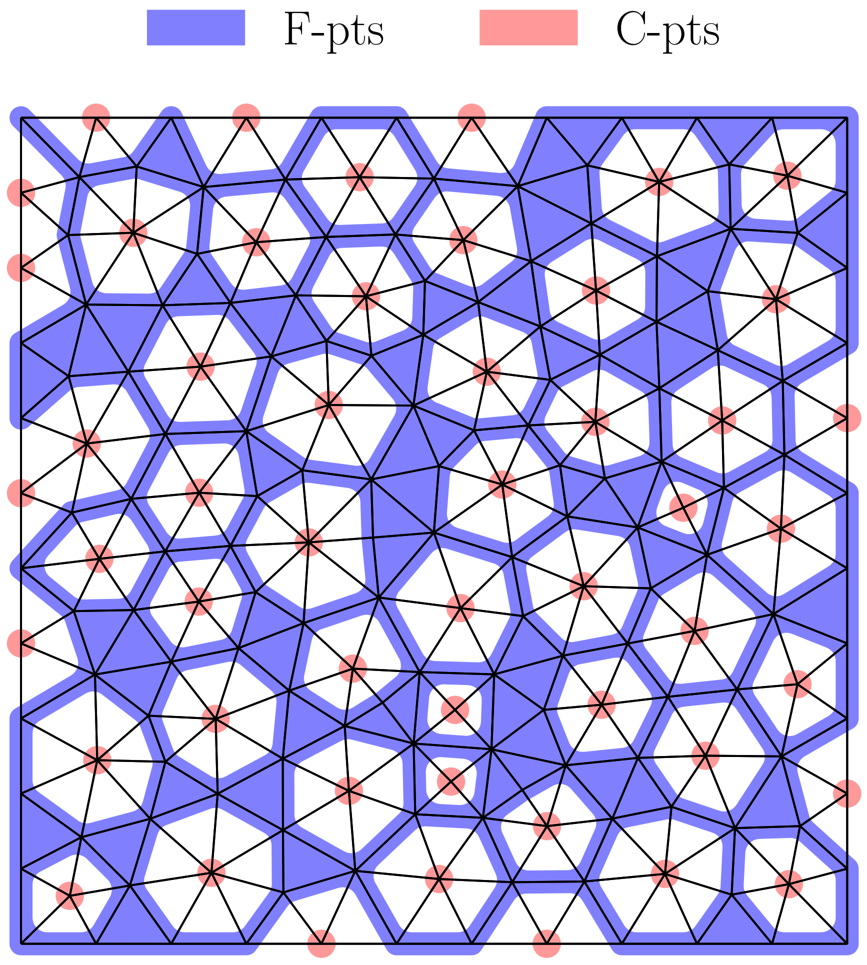




Algebraic Multigrid

[Copper Mountain 2021 Tutorial]()

James Brannick (Penn State) — Algebraic Multigrid

Karsten Kahl (University of Wuppertal) — Generalized Geometric-Algebraic Multigrid

A helpful reminder … projection methods

- **The Problem**

$$Ax = b$$

- Residual

$$r = b - Ax$$

- Subspace

$$V$$

- Finds the "best" approximation

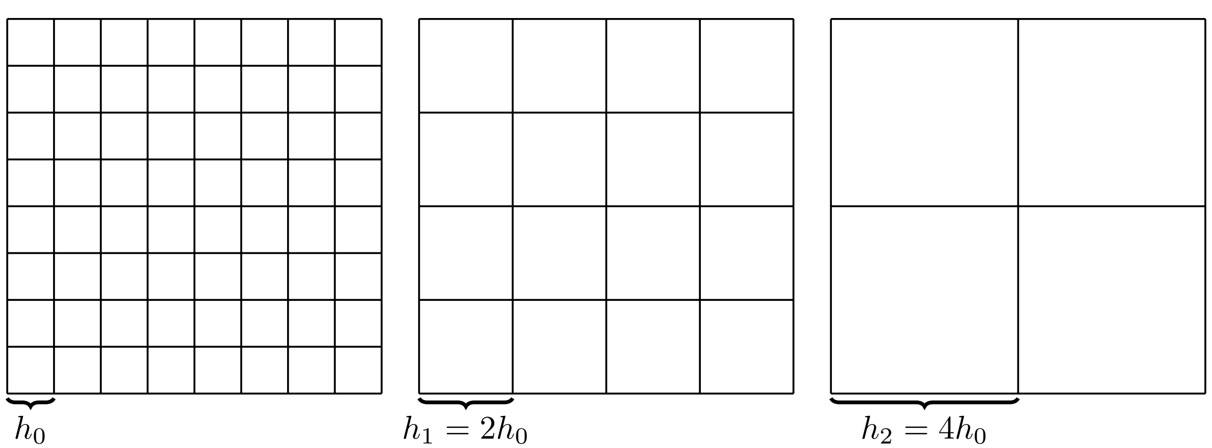
$$x \leftarrow x + V(V^T A V)^{-1} V^T r$$

$\boxed{\text{FEM, CG, Jacobi, Multigrid…}}$

Multigrid Basics

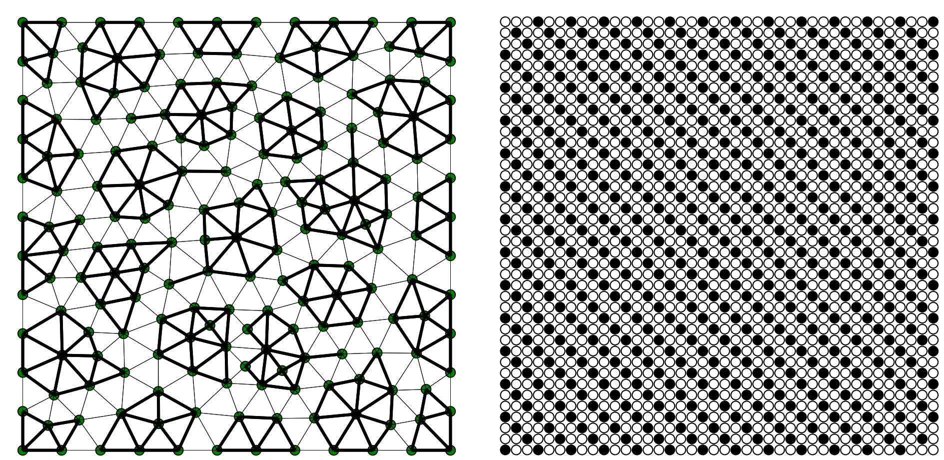
Geometric

- Grids are fixed
- Construct interpolation
- Find the best smoother



Algebraic

- Relaxation method is fixed
- Find coarse grids
- construct interpolation



Two Grid Algorithm

Algorithm 1: CGC

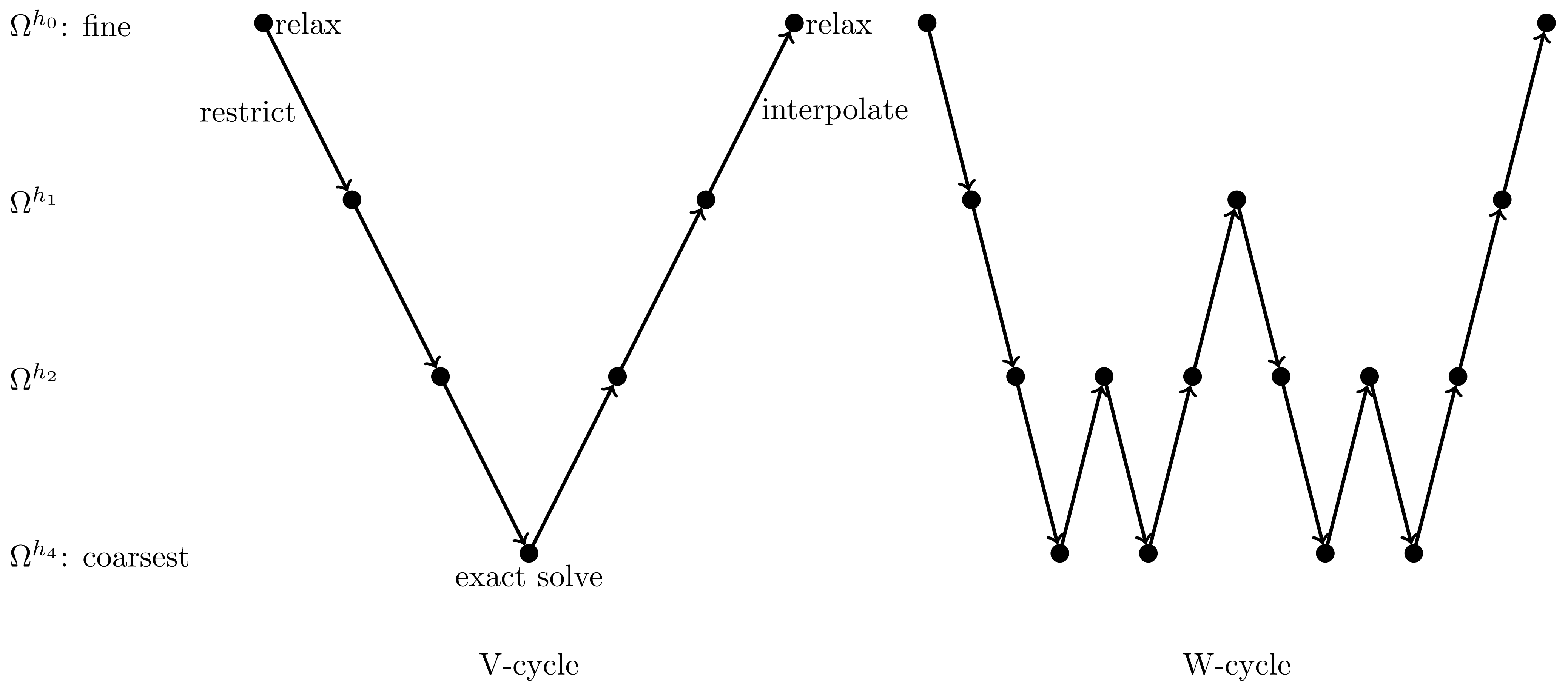
Input: u^h, f^h, $\alpha 1$, and α_2
Return: u^h

ωJacobi(A^h, u^h, f^h, α_1) `{pre-relaxation}`
$r^h = f^h - A^h u^h$ `{fine-grid residual}`
$r^{2h} = I_h^{2h} r^h$ `{restriction}`
solve $A^{2h} e^{2h} = r^{2h}$ `{coarse problem}`
$u^h \leftarrow u^h + I_{2h}^h e^{2h}$ `{interpolation and correction}`
ωJacobi(A^h, u^h, f^h, α_2) `{post-relaxation}`

- Coarse grid operators
    - Rediscritzed
    - Galerkin

$$A_{2h} = RAP = I_h^{2h} A_h I_{2h}^h$$

The Multigrid V-Cycle and W-Cycle



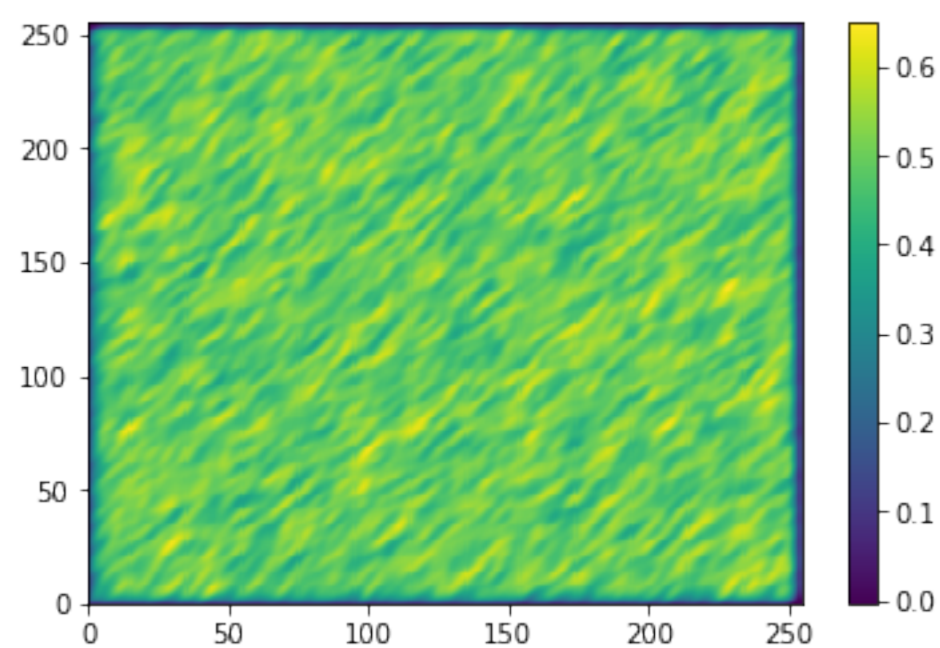
- Two-grid cycle can expose issues with coarser interpolation
- W-Cycle can account for inadequate coarser level solves
- **Exact solve?** Usually a pseudo-inverse

What can go wrong?!

- Consider *anisotropic problems*

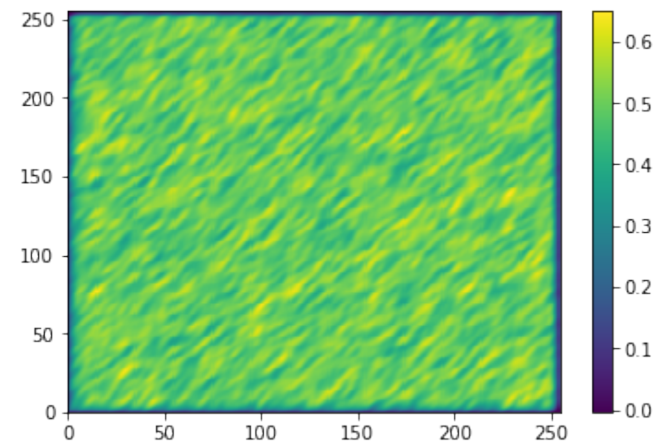
$$-u_{xx} - \epsilon u_{yy} = f$$

$$-\nabla \cdot \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \epsilon \end{bmatrix} \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \nabla u = f$$

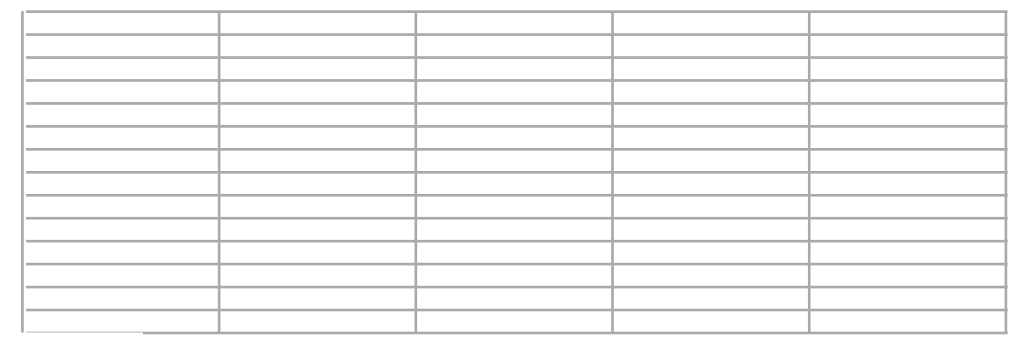


What can go wrong?!

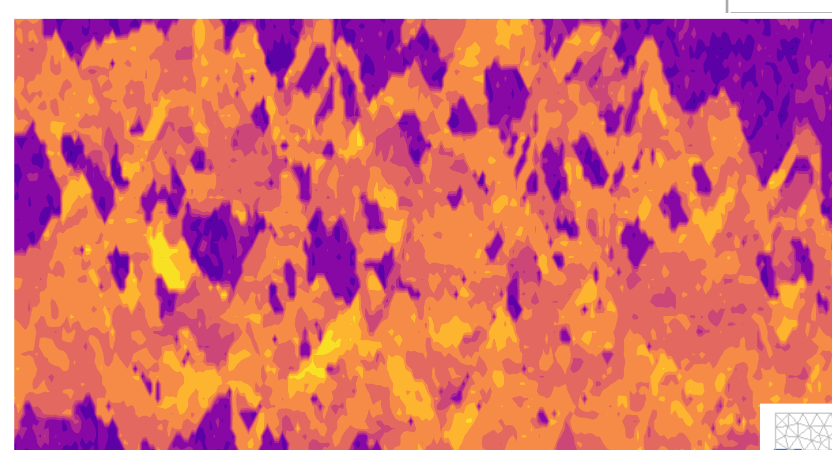
- Anisotropy



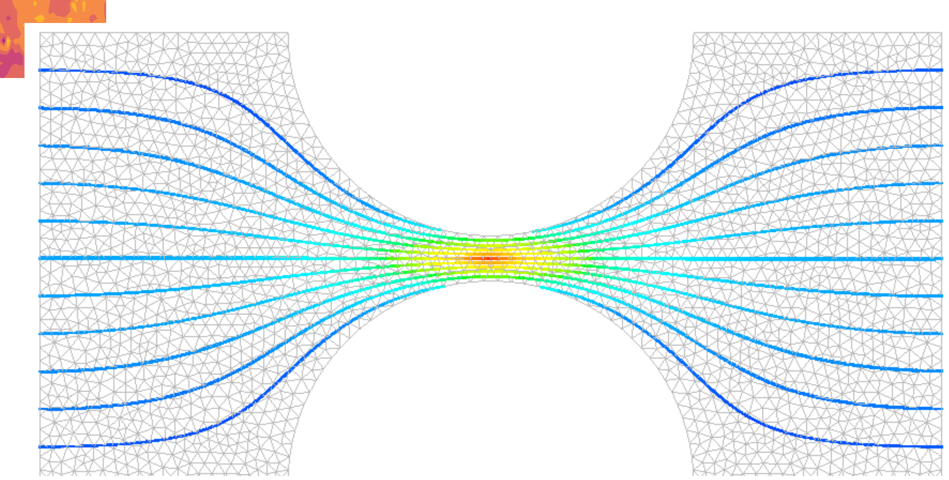
- Mesh stretching



- Jumping coefficients

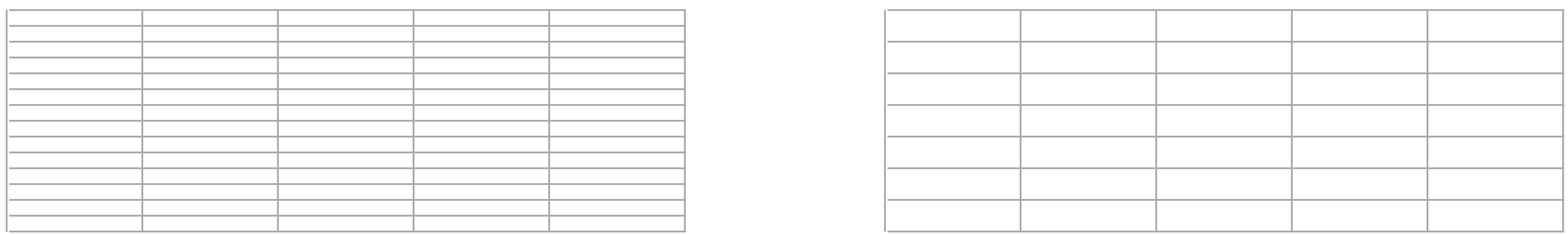


- Non-elliptic

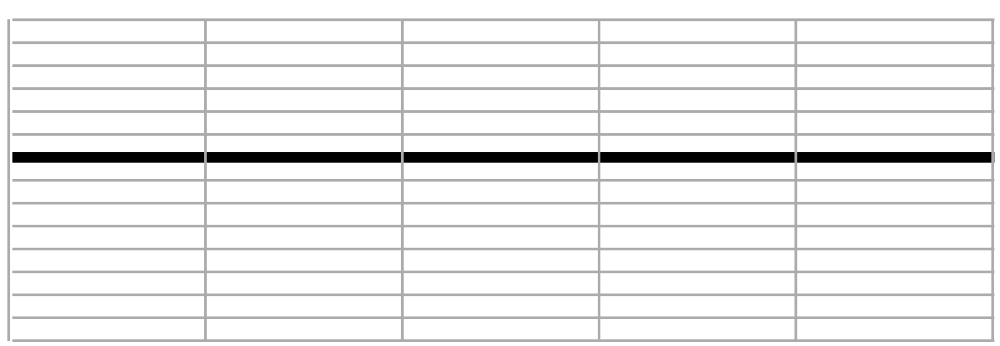


Options for more robust Multigrid

- **Semicoarsening**
  Coarsen in the direction of smoothness



- **Line/plane relaxation**
  Perform relaxation in groups (in a line)



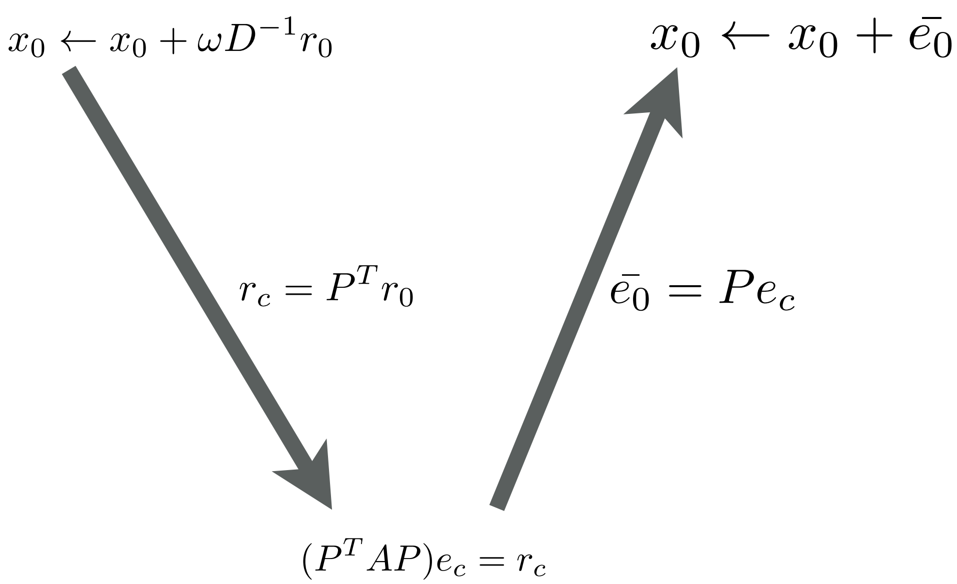
- **Operator based Interpolation** (e.g. BoxMG)

$$Ae = 0$$

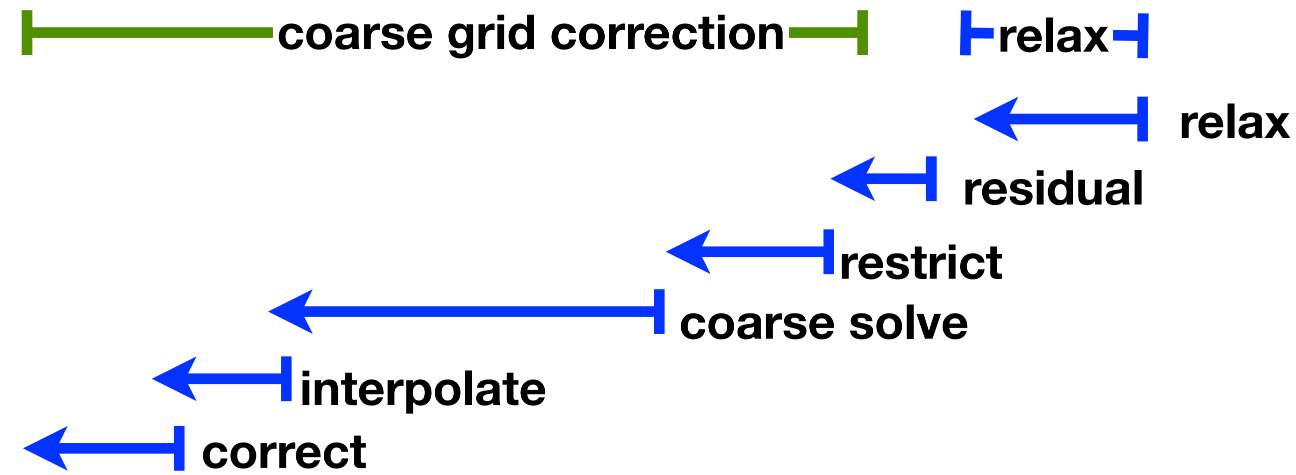
J. E. Dendy, Black box multigrid, J. Comput. Phys., 1982

J. E. Dendy and J. D. Moulton, Black box multigrid with coarsening by a factor of three, J. Numer. Lin. Alg. App., 2010

Algebraic Observation



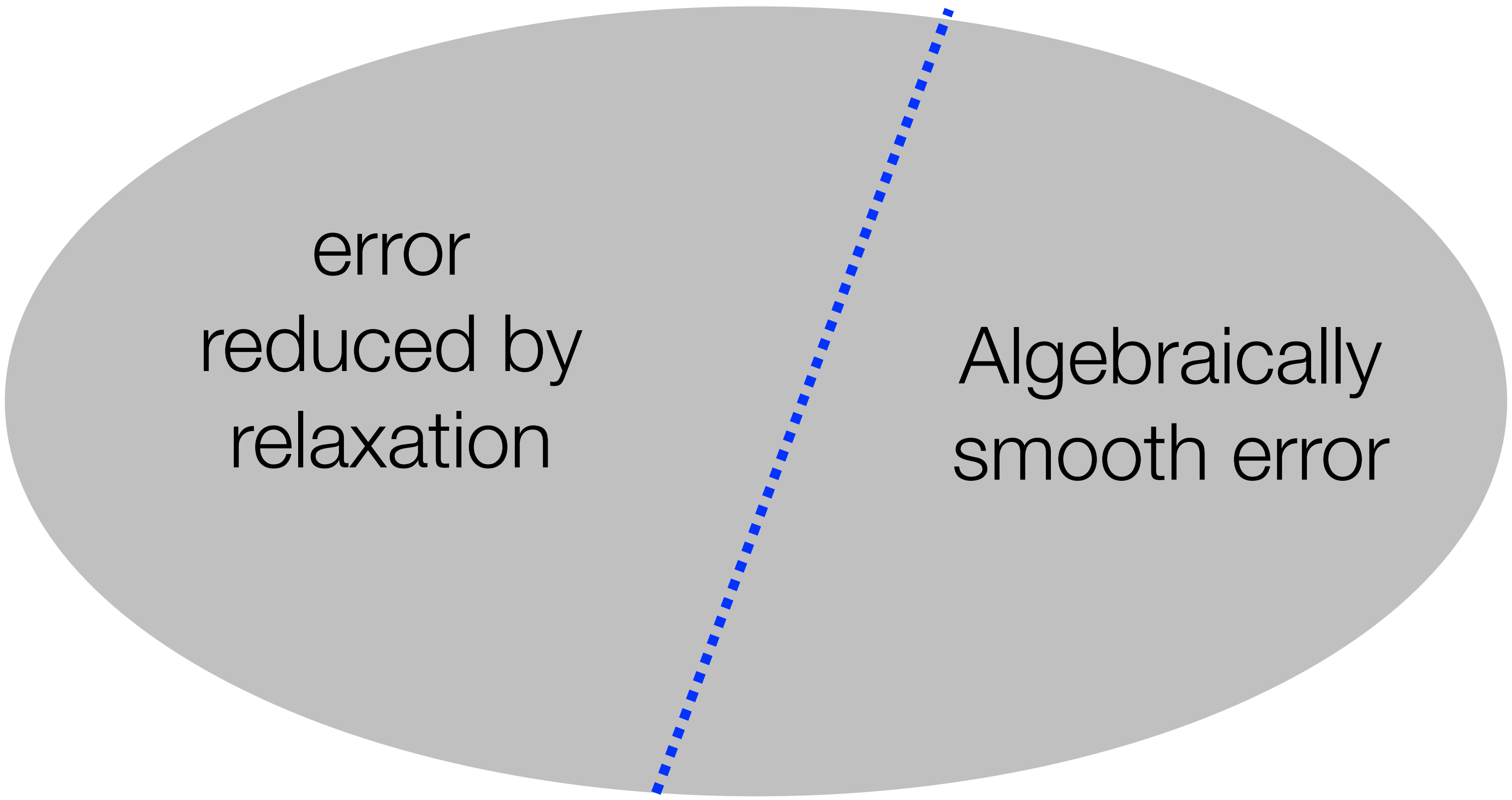
$$e_1 \leftarrow (I - P(P^T AP)^{-1} P^T A) G e_0$$



$$G e_0 \in \mathcal{R}(P) \quad \Rightarrow \quad e_1 = 0$$

interpolation should capture what relaxation misses

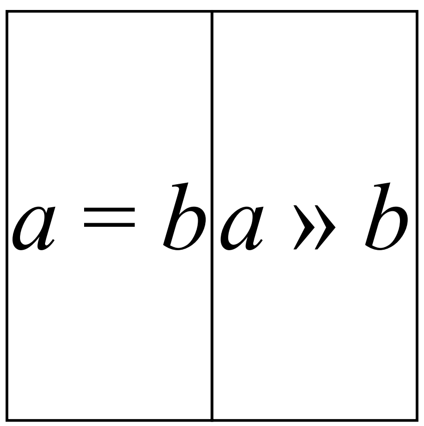
Algebraically Smooth Error

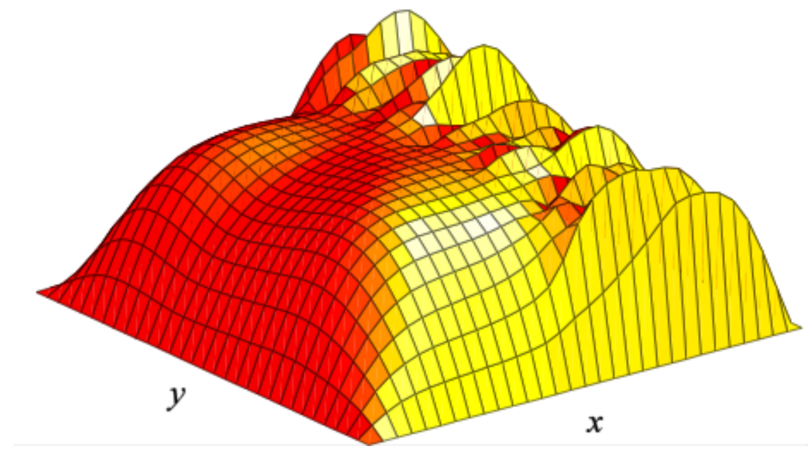


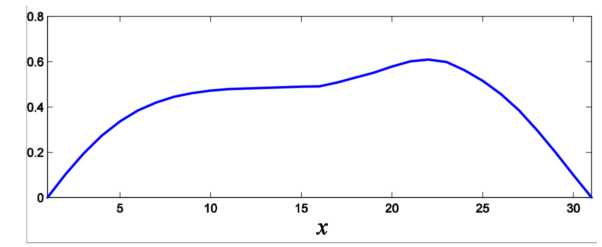
- **"Algebraically smooth" error may not be geometrically smooth**

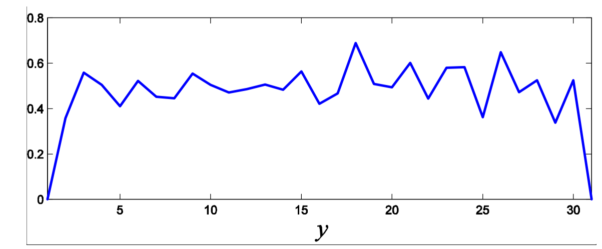
Error left by relaxation can be geometrically oscillatory

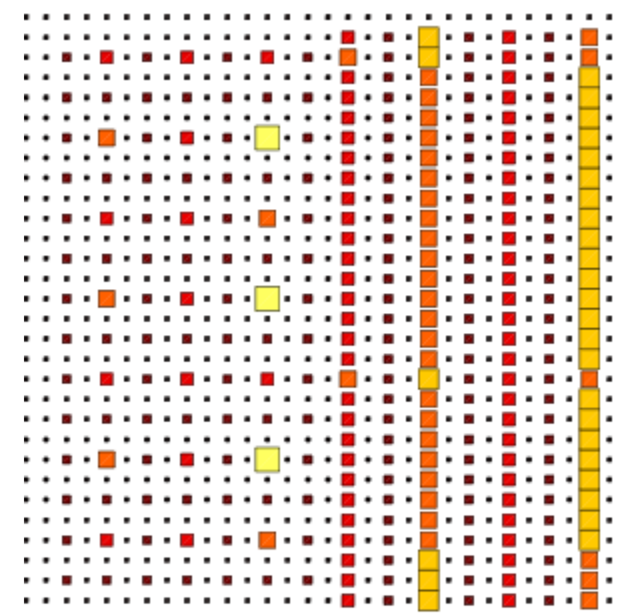
- 7 GS sweeps on
  $$-au_{xx} - bu_{yy} = f$$











AMG coarsens grids in the direction of geometric smoothness

- Caution: this example
  - targets geometric smoothness
  - uses pointwise smoothers

Main idea: Algebraically smooth error

- Take a relaxation scheme such as w-Jacobi

$$e \leftarrow (I - M^{-1}A)e$$

- If relaxation stagnates, then the remaining error exhibits poor convergence, so

$$(I - M^{-1}A)e \approx e \Rightarrow M^{-1}Ae \approx 0 \Rightarrow r \approx 0$$

- Formally (characterized by small eigenvalues)

$$\langle Ae, e \rangle << 1$$

Main idea: Algebraically smooth error

- We then have

$$\langle Ae, e\rangle = \sum_i e_i(A_{ii}e_i + \sum_{j\neq i} A_{ij}e_j)$$

assume zero row sum

$$= \sum_i e_i \left(\sum_{j\neq i} -A_{ij}(e_i - e_j) \right)$$

$$= \sum_{i<j} -A_{ij}\cdot e_i \cdot (e_i - e_j) + \sum_{i>j} -A_{ij}\cdot e_i \cdot (e_i - e_j)$$

swap i, j

$$= \sum_{i<j} -A_{ij}\cdot e_i \cdot (e_i - e_j) - \sum_{i<j} -A_{ij}\cdot e_i \cdot (e_i - e_j)$$

$$= \sum_{i<j} -A_{ij}\cdot (e_i - e_j)^2$$

- Ok, so smooth error varies **slowly** in the direction of large matrix coefficients

Briggs, William L. and Henson, Van Emden and McCormick, Steve F., A Multigrid Tutorial (2Nd Ed., 20000

Main idea: Algebraically smooth error

- We have assumed **geometric** smoothness to show

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

- **CF AMG**: 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

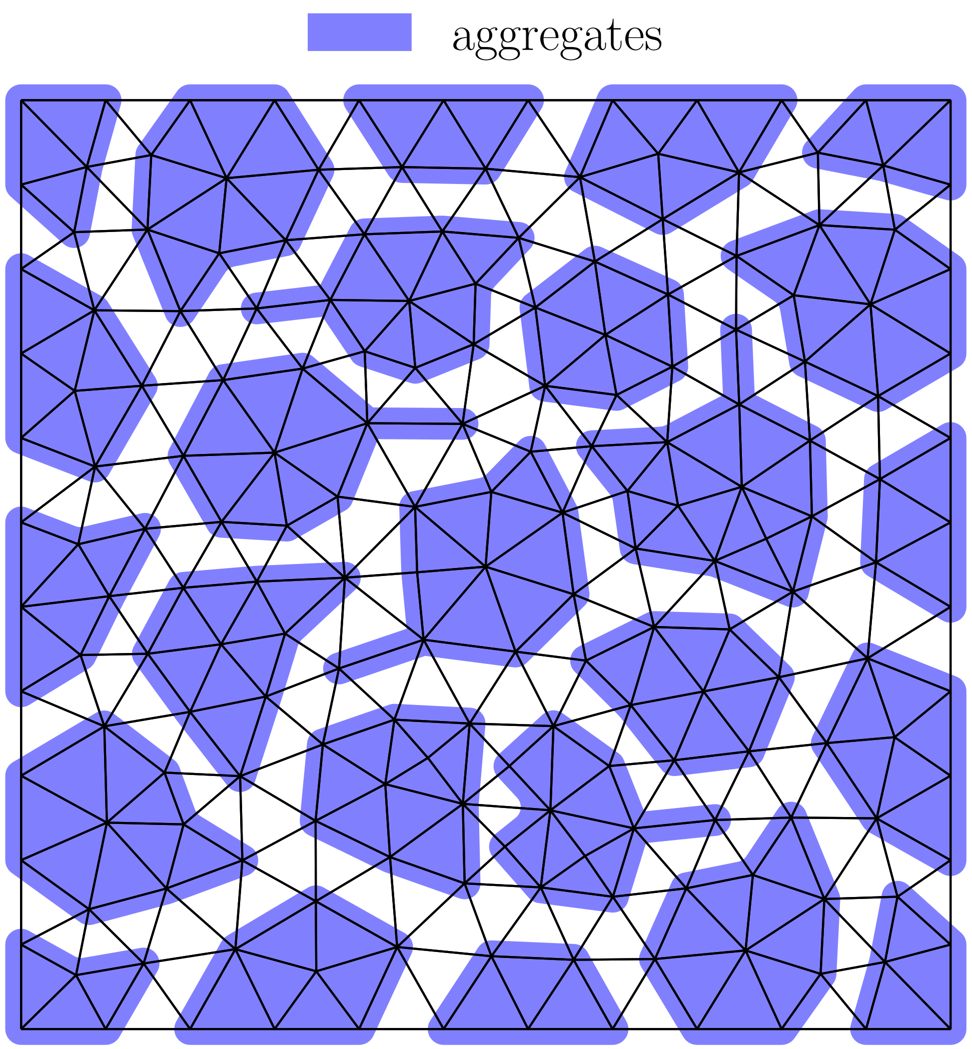
$$-a_{ij} \geq \theta \max_{k \neq i} \{-a_{ik}\}$$

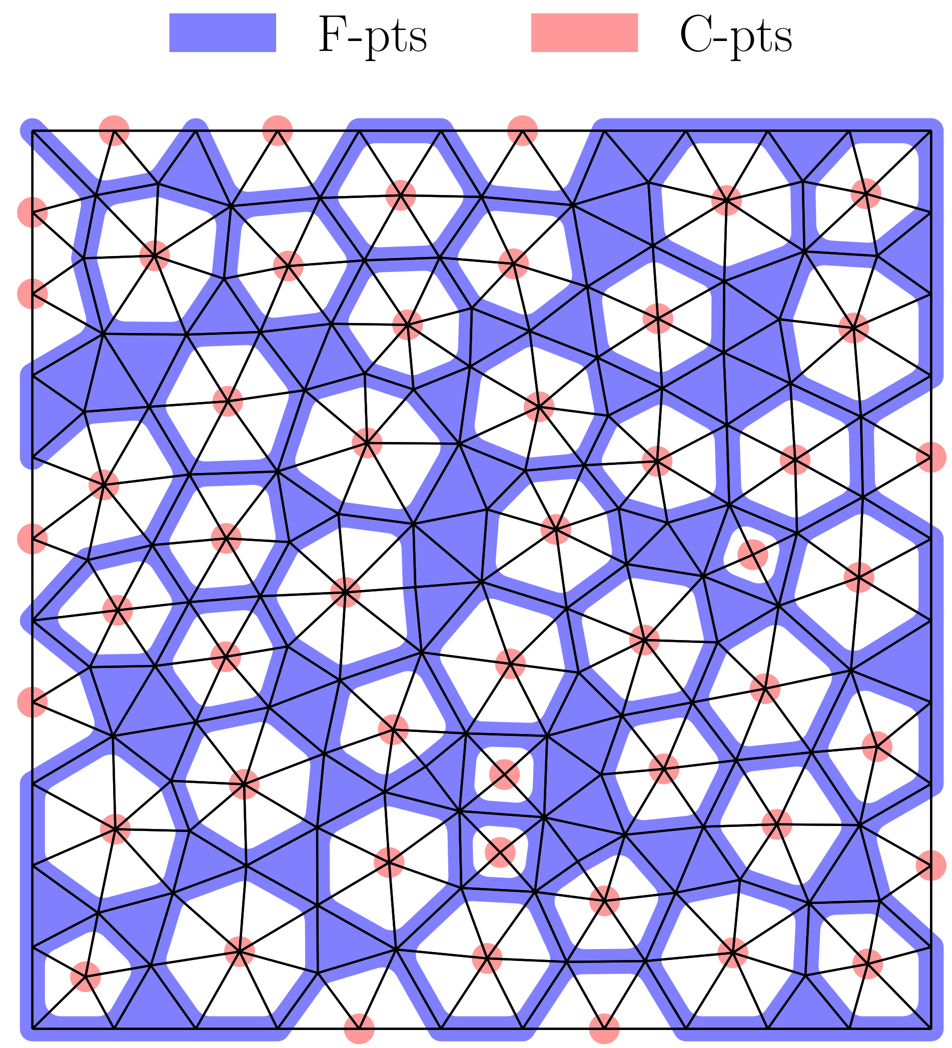
- Often positive off-diagonals are treated as weak

- This definition of strength of connection is not symmetric

Slide credit: R. Falgout, 2/1/2016

Two (general) forms of AMG





- Smoothed Aggregation AMG (SA-AMG)
- Interpolation constructed from candidate vectors
- Clear approach to *optimize* interpolation

- Coarse-Fine AMG (CF-AMG) or Ruge-Stüben
- Coarse grid points are a subset of the fine grid points
- Edge-wise construction of interpolation, allowing straightforward control of sparsity
- Incorporating near-nullspace is not straightforward

CF AMG

- **Goal:** select grid points to form the coarse grid where smooth error is well represented

- **Idea:** the variable at *j* would be a good **C-point** if it strongly influences the variable at *I*

- Strongly depend on…

$$S_i = \{j \, : \, -A_{ij} \geq \theta \max_{k \neq i} -A_{ik}\}$$

- Strongly influence…

$$S_i^T = \{j \, : \, i \in S_i\}$$

J. W. Ruge K. Stüben, Algebraic Multigrid, 1987

CF AMG

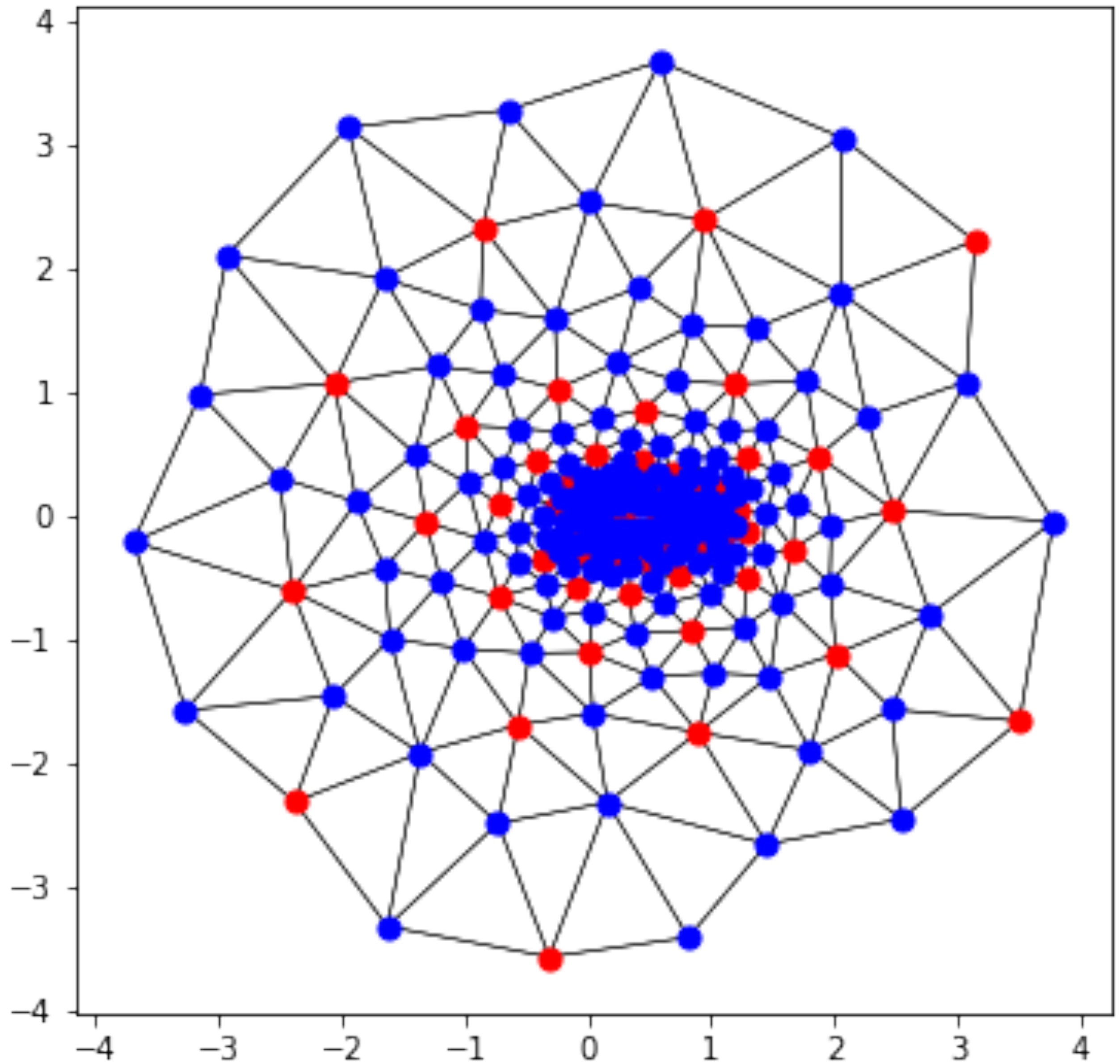
- **C-points:** coarse grid points
- **F-points:** fine grid points
- Either a C-pt or an F-pt

$$\Omega = C \cup F \quad C \cap F = \emptyset$$

- Coarse interpolatory set

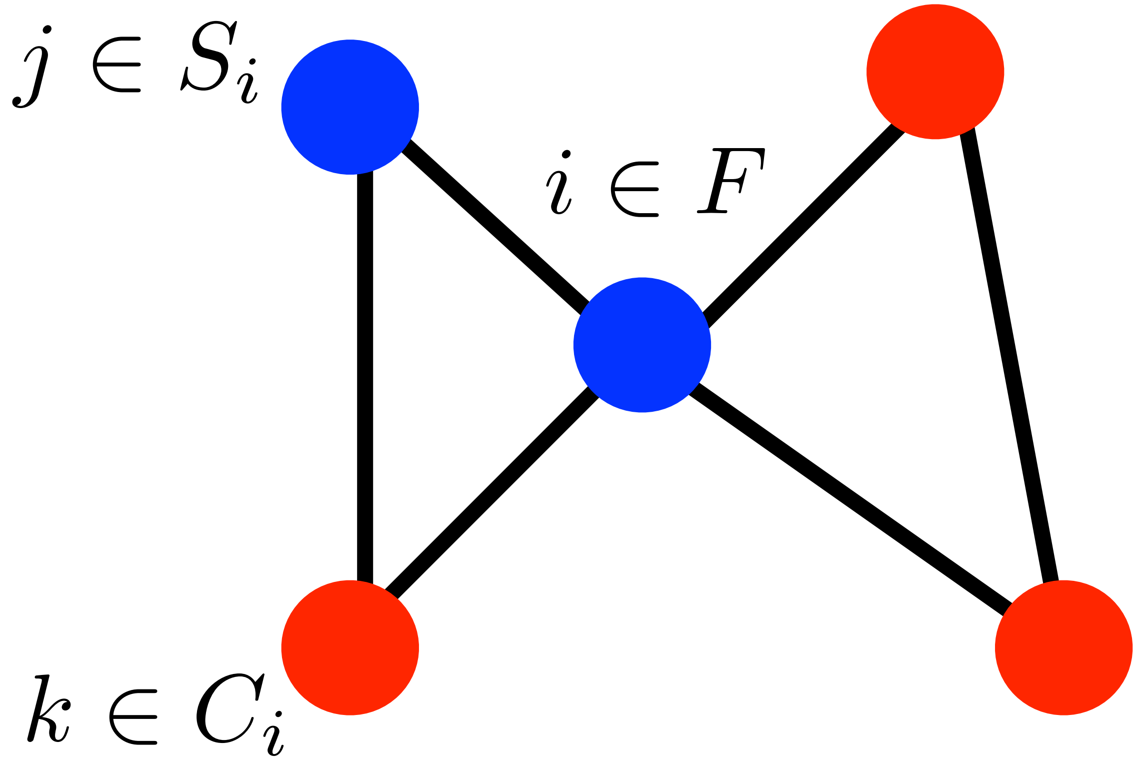
$$C_i$$

C-pts that are used to interpolated F-pt i.



CF AMG

- (C1) Each $i \in F$ should strongly depend on either
  - A point in C
  - A point that strongly depends on a point in C_i



CF AMG

- (C2) The C-points should be *maximal* with no C-point depending on another C-point

- (C1) increases the size of the coarse grid (C-points)
- (C2) puts constraints on the *size* of the coarse grid
- Must satisfy (C1) in order to construct interpolation. Use (C2) to help limit computational complexity

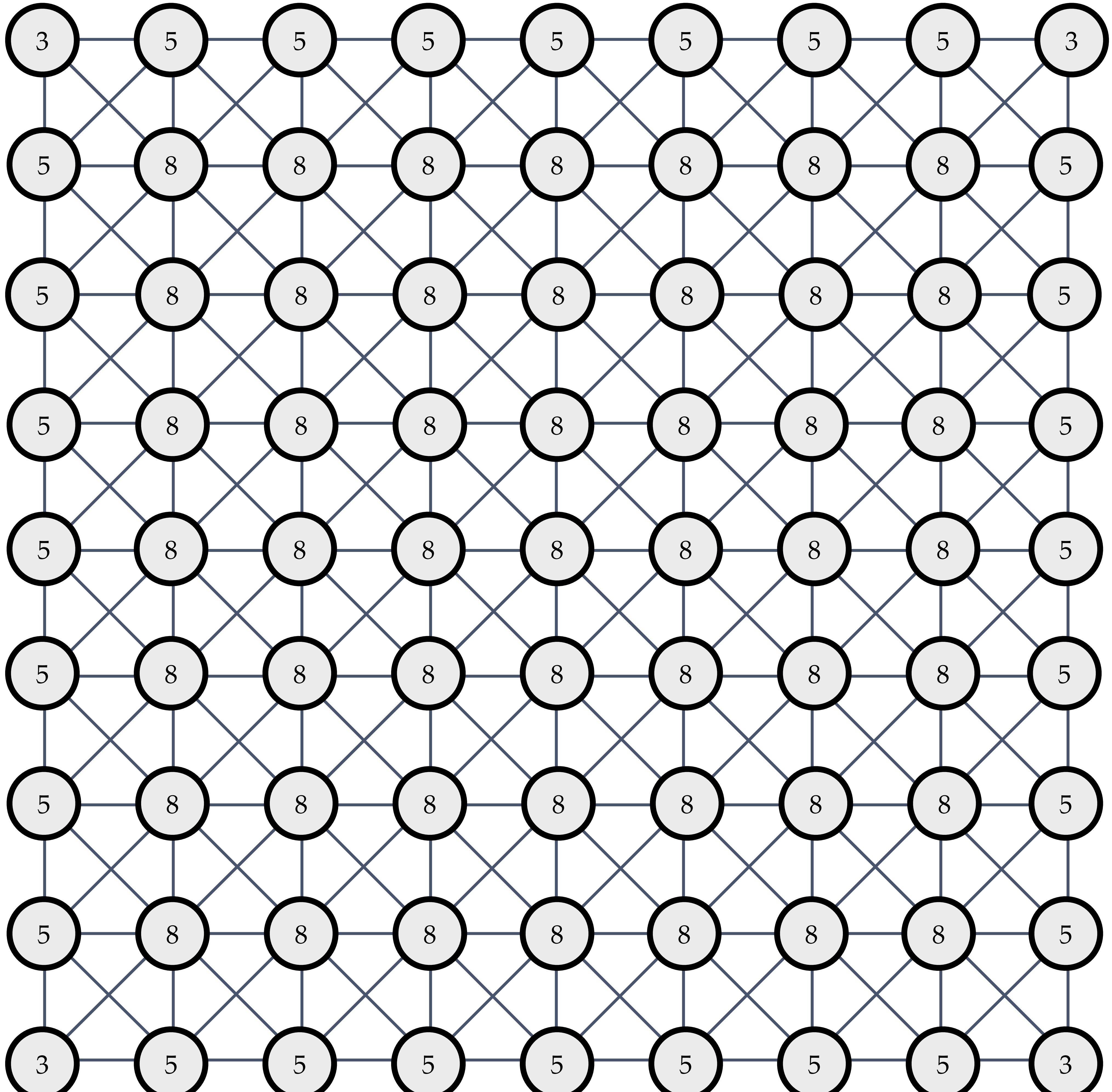
Ruge-Stüben

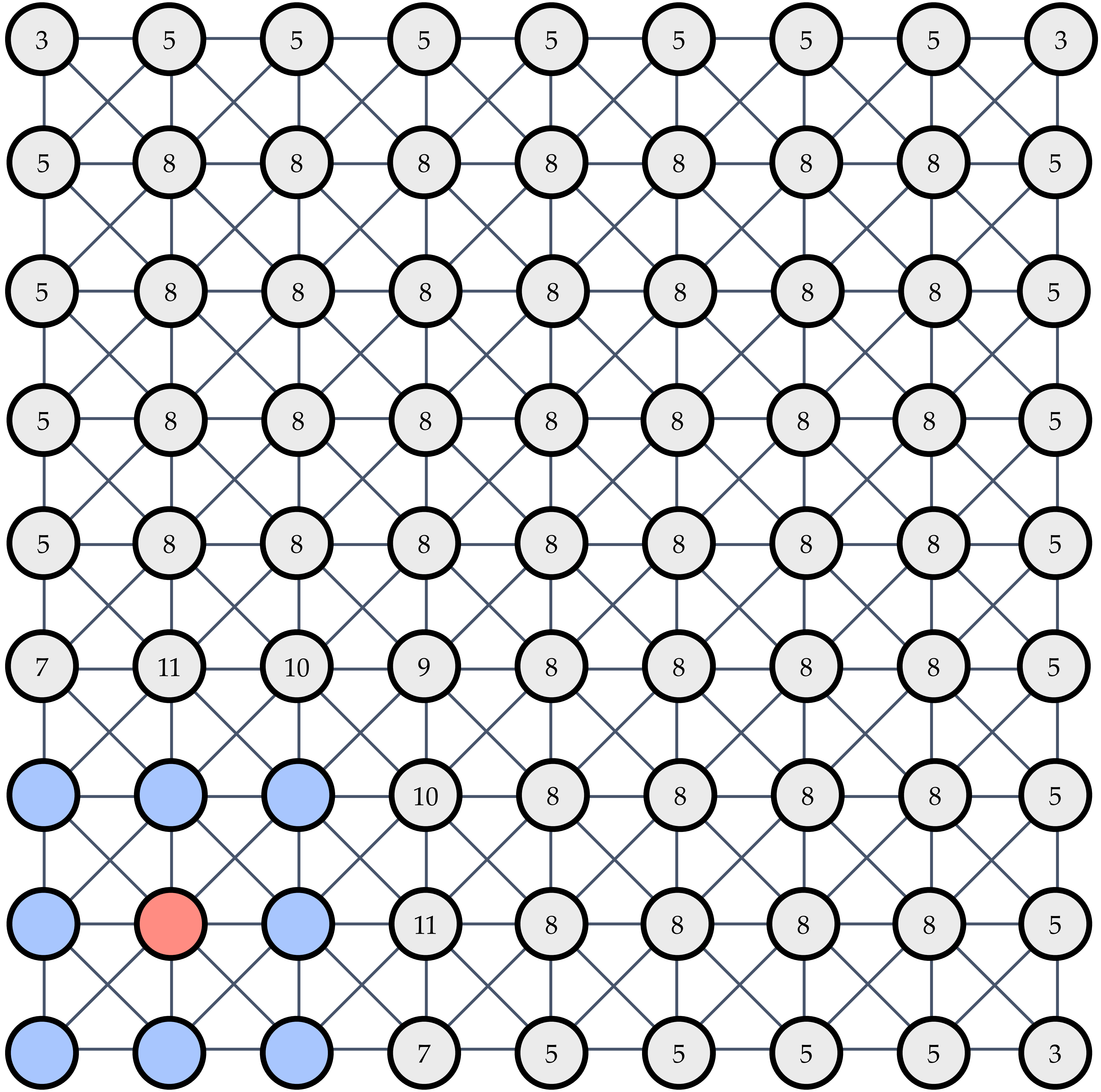
Algorithm 3. RUGE–STÜBEN.

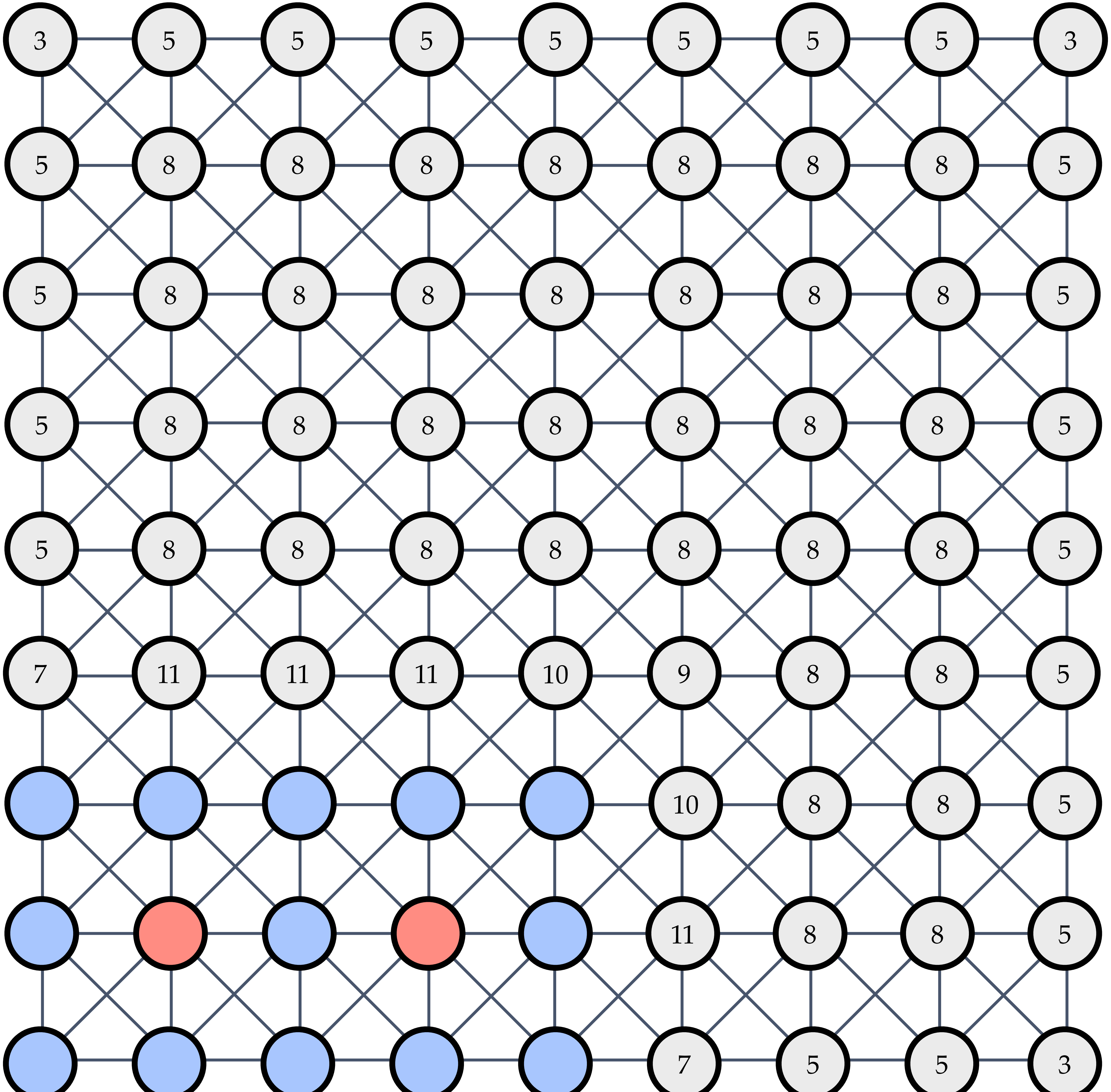
Initialize:

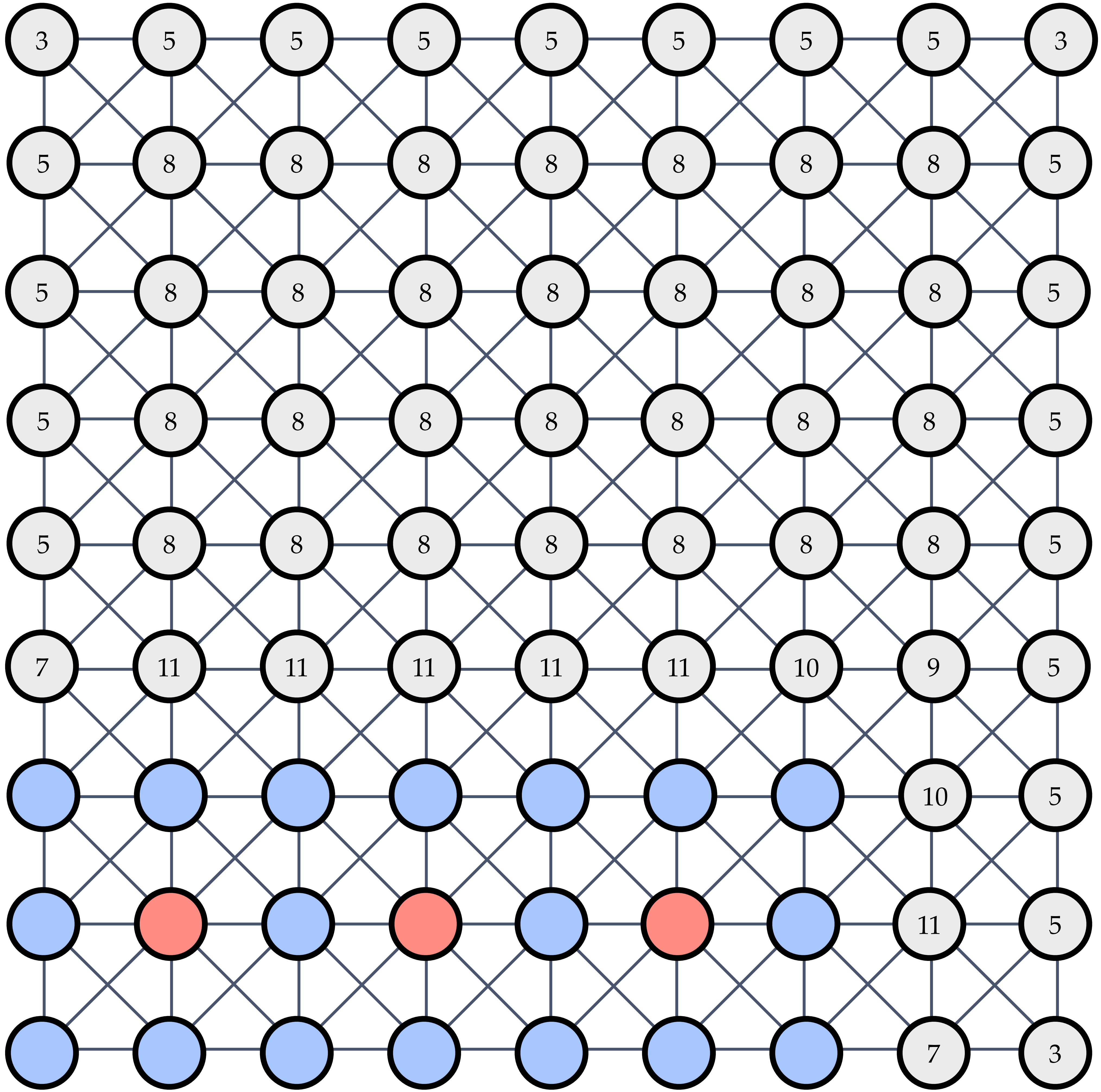
$U = \Omega,\ C = \emptyset,\ F = \emptyset$

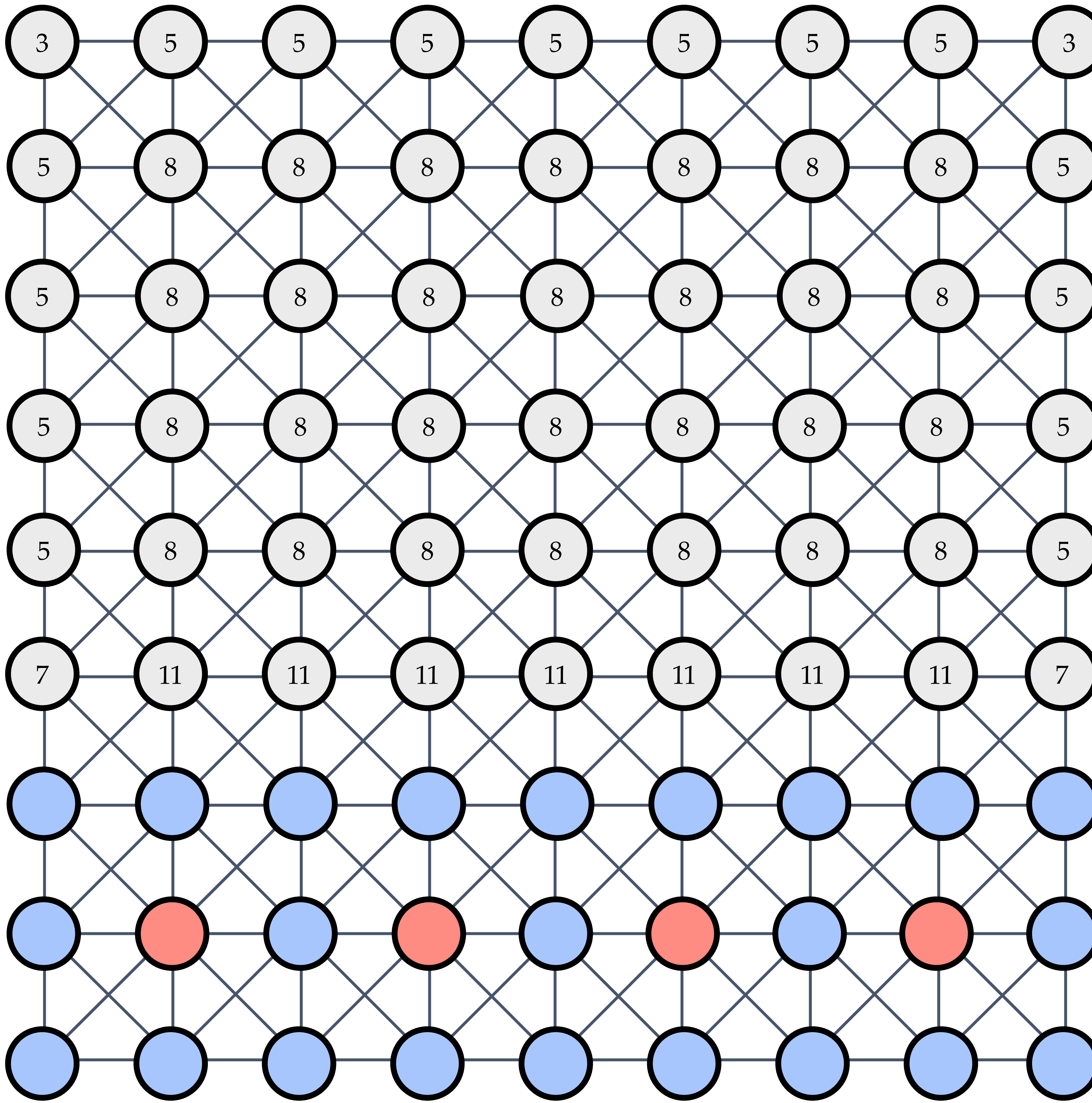
for all $i \in \Omega$ **do**
    $w_i \leftarrow |S_i^{\mathrm{T}}|$
end for
while $|U| > 0$ **do** {First pass}
    select i: $w_i \geqslant w_j,\ \forall j \in U$
    $U \leftarrow U \setminus \{i\}$
    $C \leftarrow C \cup \{i\}$
    for all $j \in S_i^{\mathrm{T}} \cap U$ **do**
        $U \leftarrow U \setminus \{j\}$
        $F = F \cup \{j\}$
        for all $k \in S_j \cap U$ **do**
            $w_k \leftarrow w_k + 1$
        end for
    end for
end while
for all $i \in F$ **do** {Second pass}
    for all $j \in S_i \cap S_i^{\mathrm{T}} \cap F$ **do**
        if $S_i \cap S_j \cap C == \emptyset$ **then**
            make i or j into C-point
        end if
    end for
end for


3 5 5 5 5 5 5 5 3
5 8 8 8 8 8 8 8 5
5 8 8 8 8 8 8 8 5
5 8 8 8 8 8 8 8 5
5 8 8 8 8 8 8 8 5
5 8 8 8 8 8 8 8 5
5 8 8 8 8 8 8 8 5
5 8 8 8 8 8 8 8 5
3 5 5 5 5 5 5 5 3

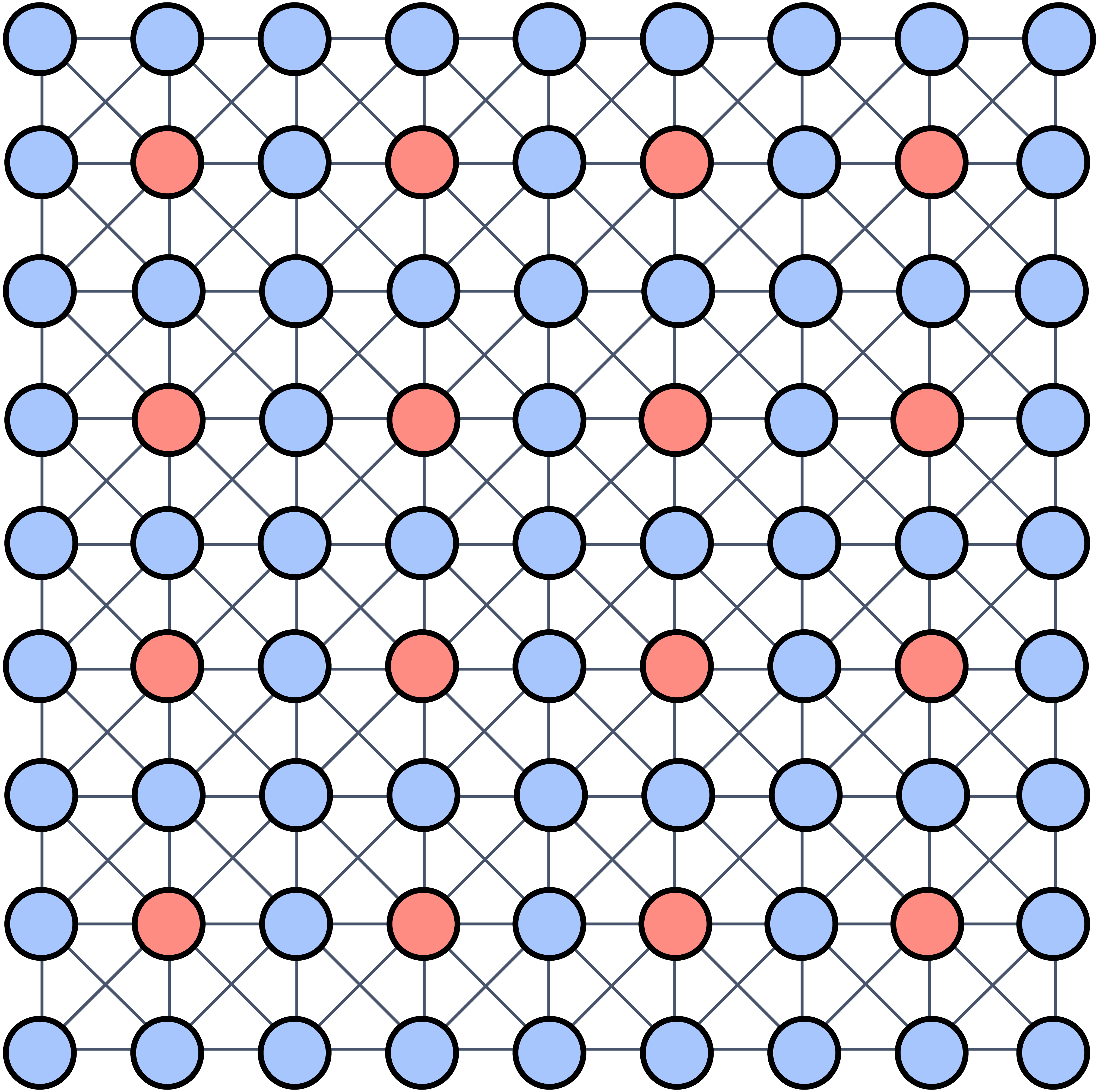

3 5 5 5 5 5 5 5 3
5 8 8 8 8 8 8 8 5
5 8 8 8 8 8 8 8 5
5 8 8 8 8 8 8 8 5
5 8 8 8 8 8 8 8 5
7 11 10 9 8 8 8 8 5
10 8 8 8 8 5
11 8 8 8 8 5
7 5 5 5 5 3


3 5 5 5 5 5 5 5 3
5 8 8 8 8 8 8 8 5
5 8 8 8 8 8 8 8 5
5 8 8 8 8 8 8 8 5
5 8 8 8 8 8 8 8 5
7 11 11 11 10 9 8 8 5
10 8 8 5
11 8 8 5
7 5 5 3


3 5 5 5 5 5 5 5 3
5 8 8 8 8 8 8 8 5
5 8 8 8 8 8 8 8 5
5 8 8 8 8 8 8 8 5
5 8 8 8 8 8 8 8 5
7 11 11 11 11 11 10 9 5
10 5
11 5
7 3



3 5 5 5 5 5 5 5 3
5 8 8 8 8 8 8 8 5
5 8 8 8 8 8 8 8 5
5 8 8 8 8 8 8 8 5
5 8 8 8 8 8 8 8 5
7 11 11 11 11 11 11 11 7



CF AMG

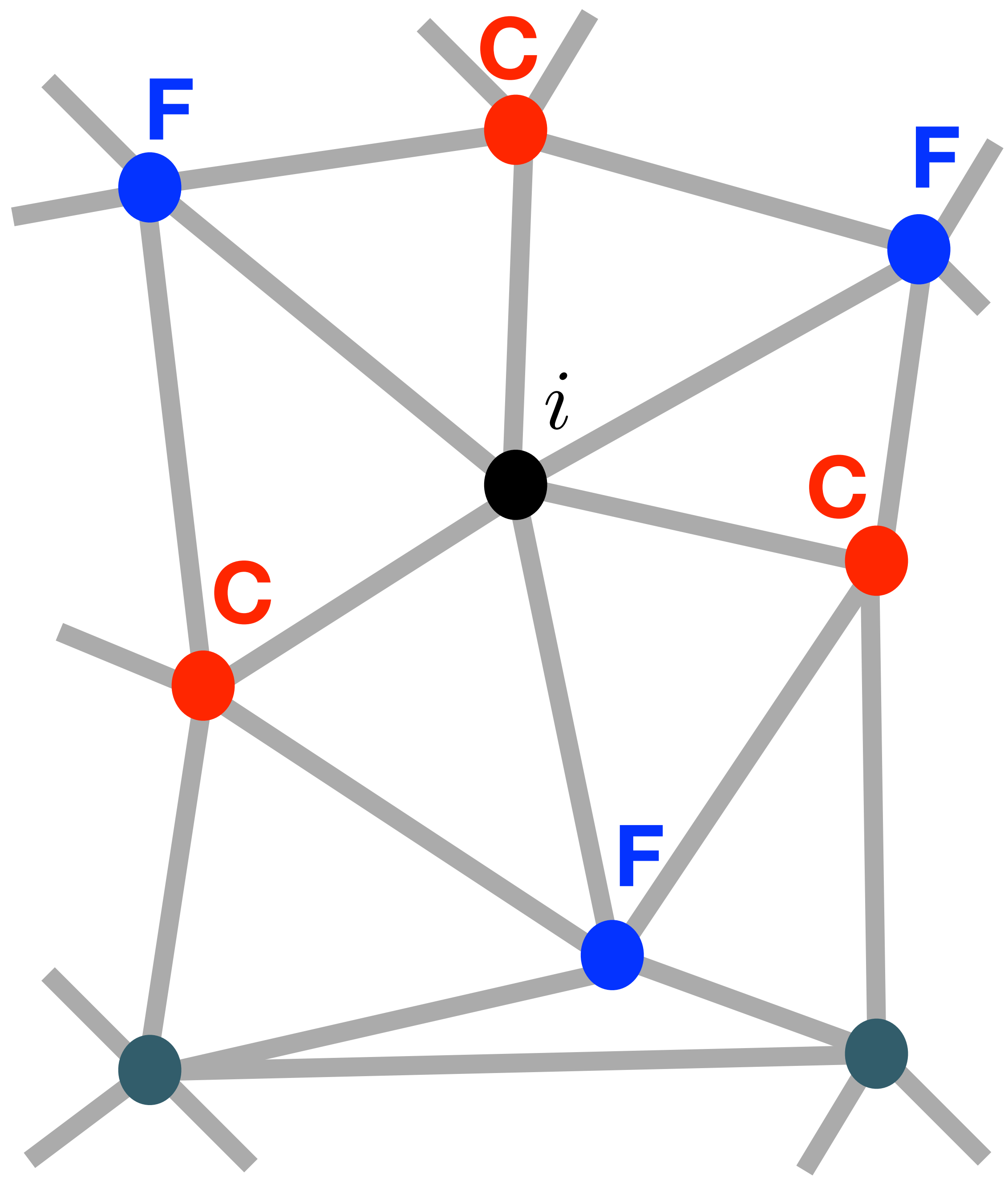
- With a coarse grid (C-points) defined, we can turn to interpolation.

- Write interpolation as a weighted sum of points in the coarse interpolatory set

$$(P\vec{e})_i = \begin{cases} e_i & i \in C \\ \sum_{j \in C_i} \omega_{ij} e_j & i \in F \end{cases}$$

- Or

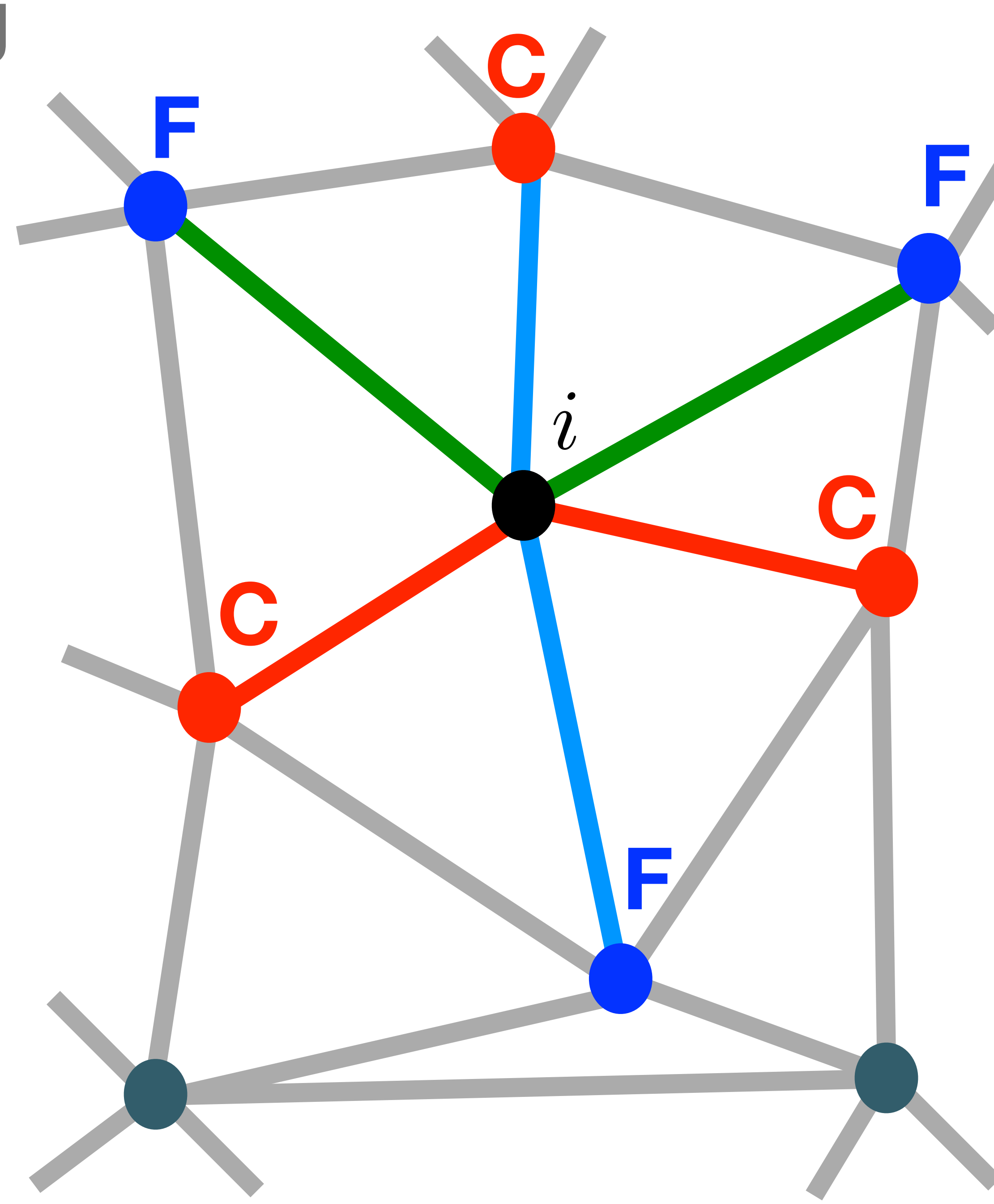
$$P\vec{e} = P \begin{bmatrix} \vec{e}_C \\ \vec{e}_F \end{bmatrix} = \begin{bmatrix} I \\ W \end{bmatrix} \begin{bmatrix} \vec{e}_C \\ \vec{e}_F \end{bmatrix}$$



Example from MG Tutorial

CF AMG

- To interpolate, distinguish **strong** and **weak** connections to interpolate from
- C_i are **strong C-points**
- D_i^s are **strong F-points**
- D_i^w are **weak C/F-points**



CF AMG

- Start with smooth error

$$Ae \approx 0$$

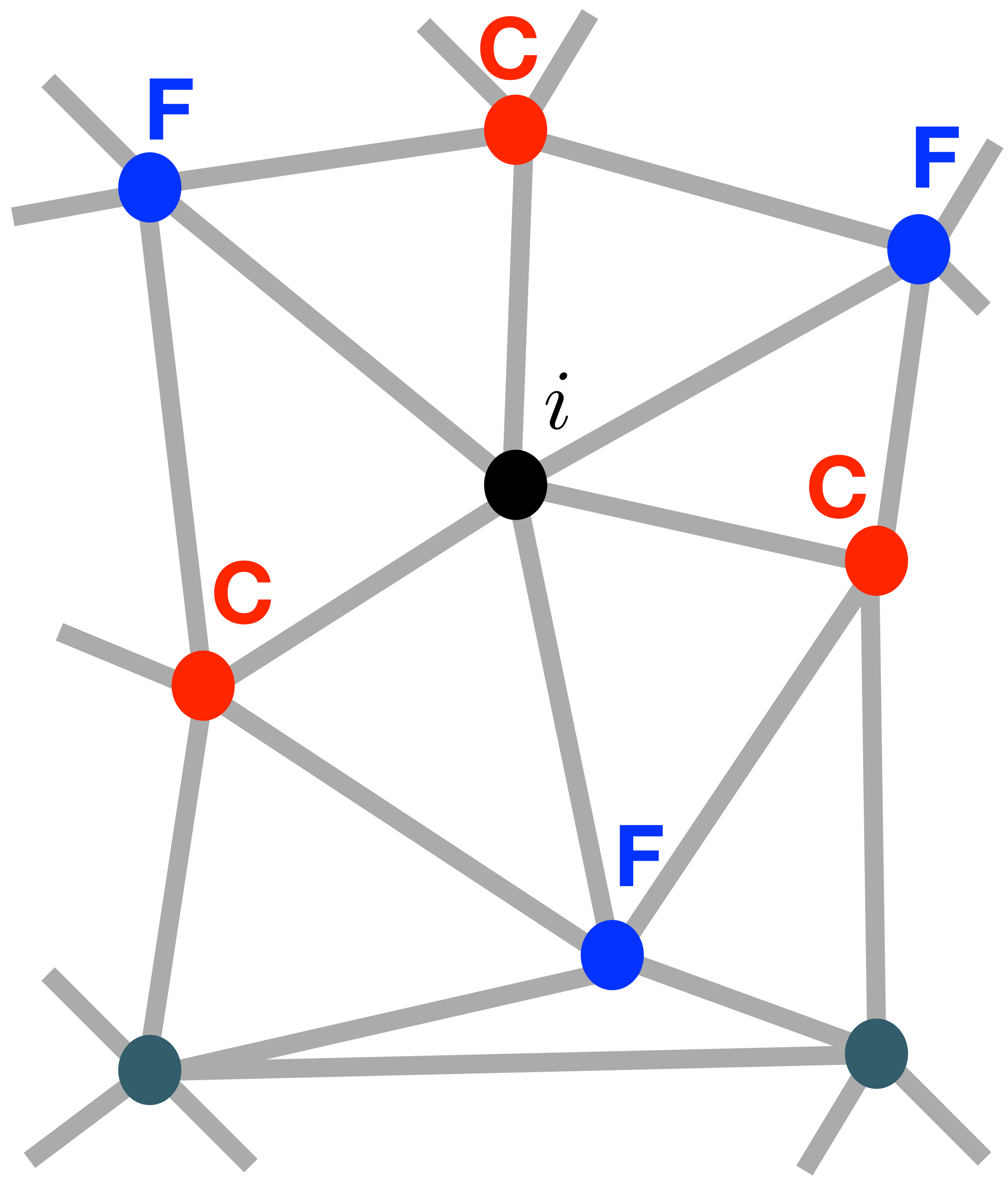
- Then "solve" for i

$$A_{ii}e_i = \sum_{j \neq i} A_{ij}e_j$$

- Then split into types

$$A_{ii}e_i = \sum_{j \in C_i} A_{ij}e_j + \sum_{j \in D_i^s} A_{ij}e_j + \sum_{j \in D_i^w} A_{ij}e_j$$

strong C **strong F** **weak**



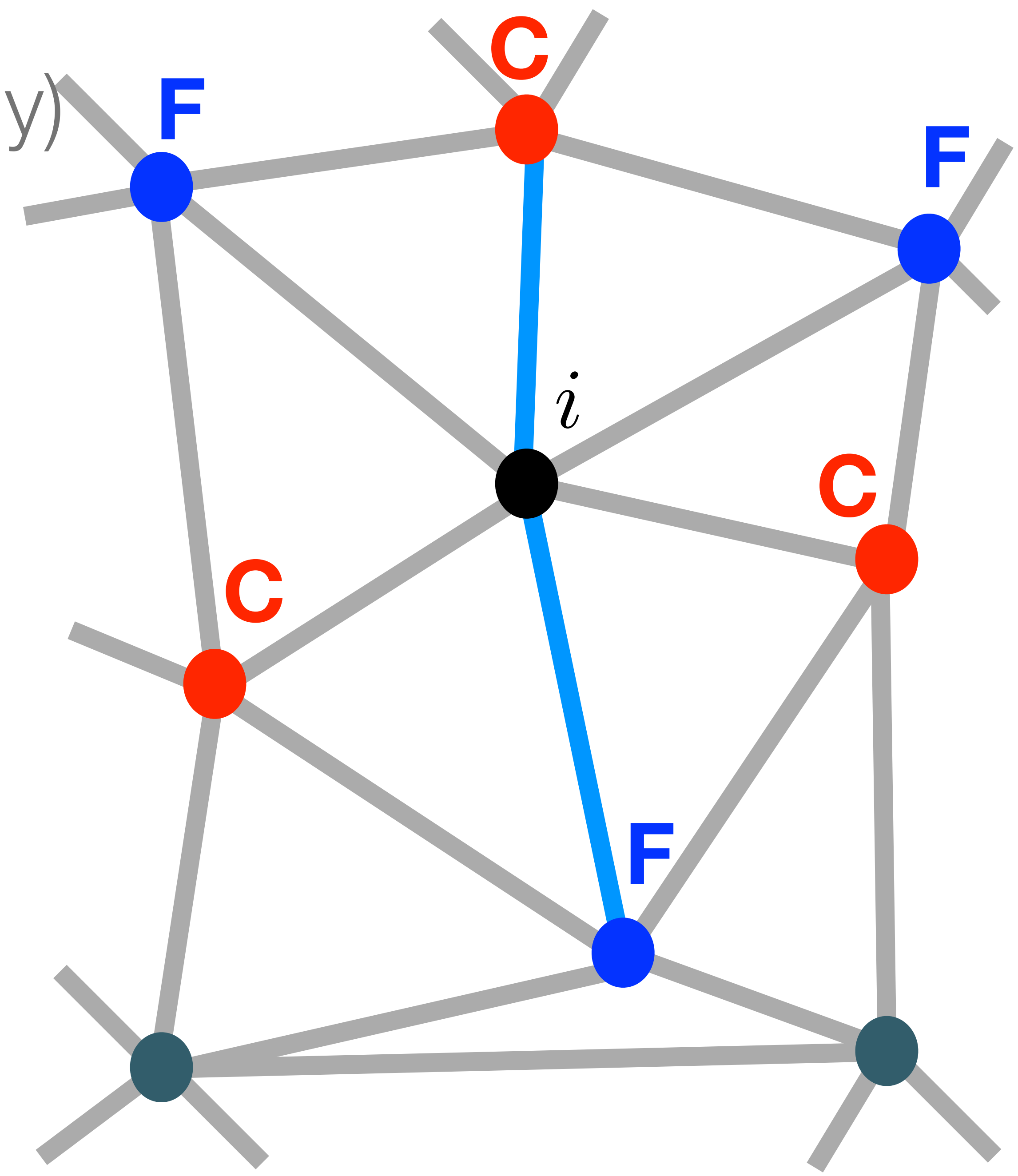
CF AMG

- **Weak:** assume $e_j \approx e_i$ in case there is dependence (=vary slowly)

$$A_{ii} e_i = \sum_{j \in C_i} A_{ij} e_j + \sum_{j \in D_i^s} A_{ij} e_j + \sum_{j \in D_i^w} A_{ij} e_j$$

$$\Downarrow$$

$$\left(A_{ii} + \sum_{j \in D_i^w} A_{ij} \right) e_i = \sum_{j \in C_i} A_{ij} e_j + \sum_{j \in D_i^s} A_{ij} e_j$$



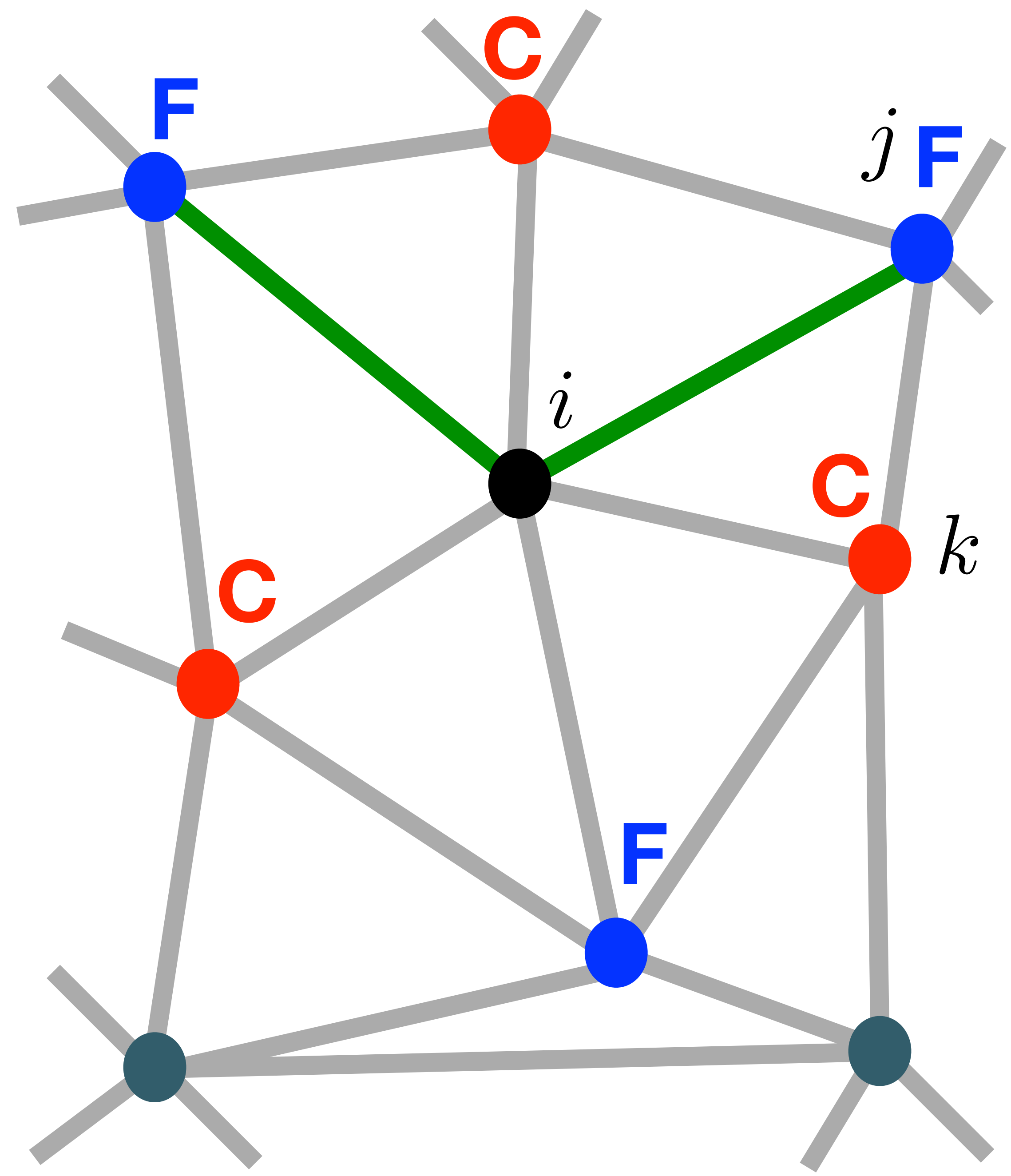
CF AMG

- **Strong F:** approximate e_j by points in $C_i \cap C_j$

$$e_j \approx \frac{\sum_{k \in C_i} A_{jk} e_k}{\sum_{k \in C_i} A_{jk}}$$

- This gives weights

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



CF AMG Setup Algorithm

Algorithm 2: `CF_setup()`

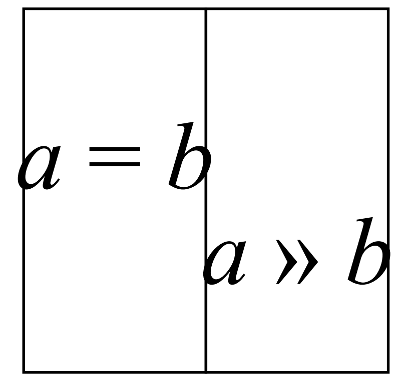
Input: A_0: fine-grid operator
`max_size`: threshold for max size of coarsest problem

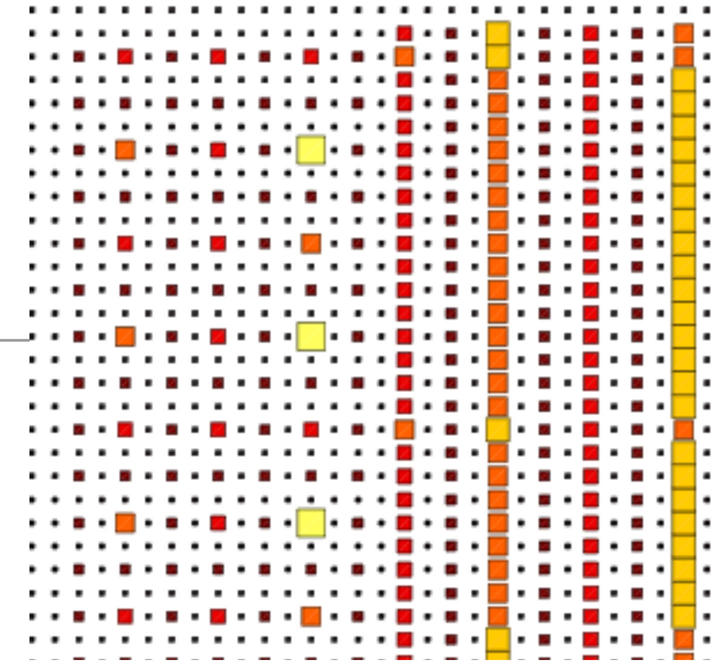
Output: $A_1, \ldots, A_L$,
$P_0, \ldots, P_{L-1}$

$\ell = 0$
while $\texttt{size}(A_\ell) > \texttt{max\_size}$
  $S_\ell = \texttt{strength}(A_\ell)$ {Strength-of-connection}
  $\mathcal{C}_\ell, \mathcal{F}_\ell = \texttt{splitting}(S_\ell)$ {C/F-splitting}
  $W = \texttt{weights}(S_\ell, A_\ell, \mathcal{C}_\ell, \mathcal{F}_\ell)$ {Interpolation weights}
  $P_\ell = \begin{bmatrix} W \\ I \end{bmatrix}$ {Form interpolation}
  $A_{\ell+1} = P_\ell^T A_\ell P_\ell$ {Coarse-grid operator}
  $\ell = \ell + 1$

Example CF-AMG results

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





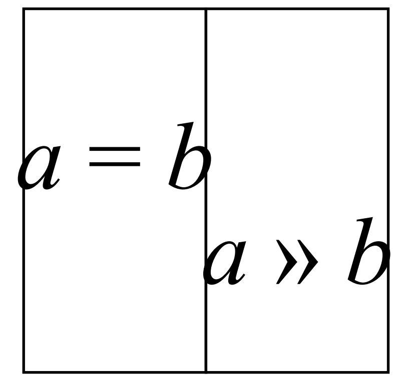
CF-AMG coarse grids

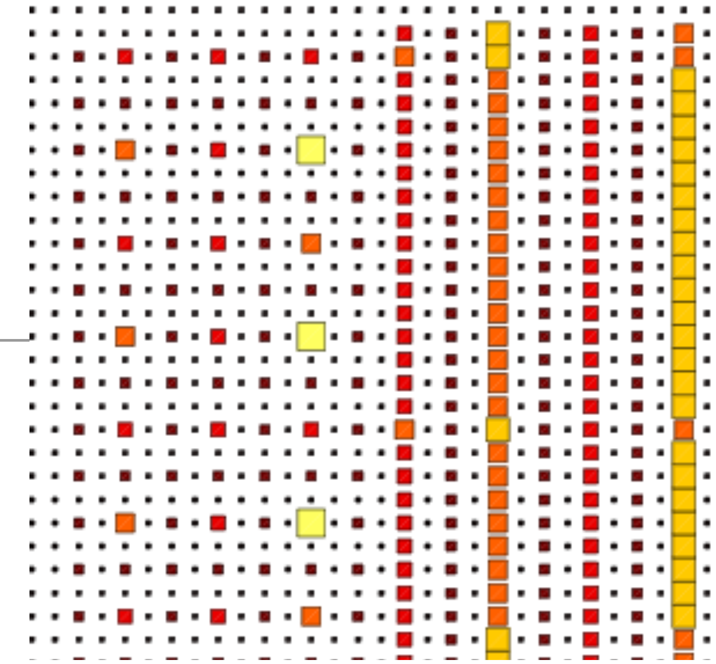
N	Iters	Conv factor	Coarse grids	Grid comp	Oper comp	Setup time	Solve time
61×61	10	0.23	6	1.6	1.6	0.01	0.02
121×121	9	0.23	8	1.6	1.7	0.05	0.07
241×241	9	0.23	9	1.6	1.7	0.25	0.32
481×481	9	0.23	12	1.7	1.7	1.02	1.27
961×961	11	0.29	13	1.7	1.7	4.42	6.28

Slide credit: R. Falgout, 2/1/2016

Example CF-AMG results

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





CF-AMG coarse grids

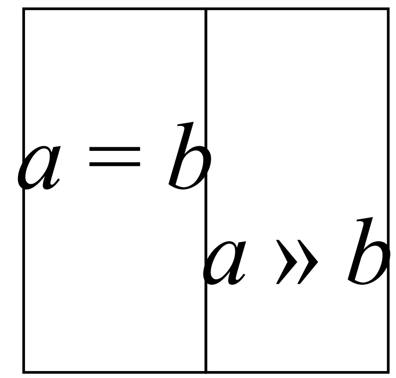
N	Iters	Conv factor	Coarse grids	Grid comp	Oper comp	Setup time	Solve time
61×61	10	0.23	6	1.6	1.6	0.01	0.02
121×121	9	0.23	8	1.6	1.7	0.05	0.07
241×241	9	0.23	9	1.6	1.7	0.25	0.32
481×481	9	0.23	12	1.7	1.7	1.02	1.27
961×961	11	0.29	13	1.7	1.7	4.42	6.28

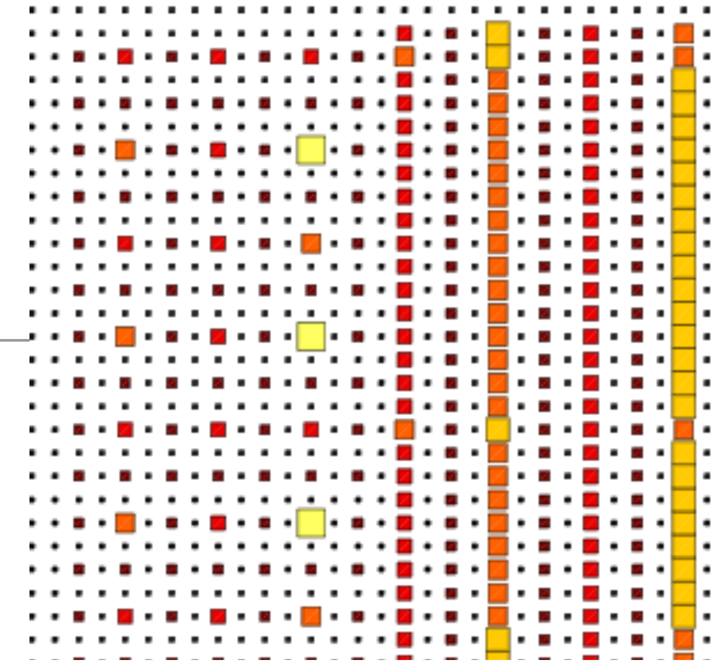
Iterations to a certain tolerance

Slide credit: R. Falgout, 2/1/2016

Example CF-AMG results

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



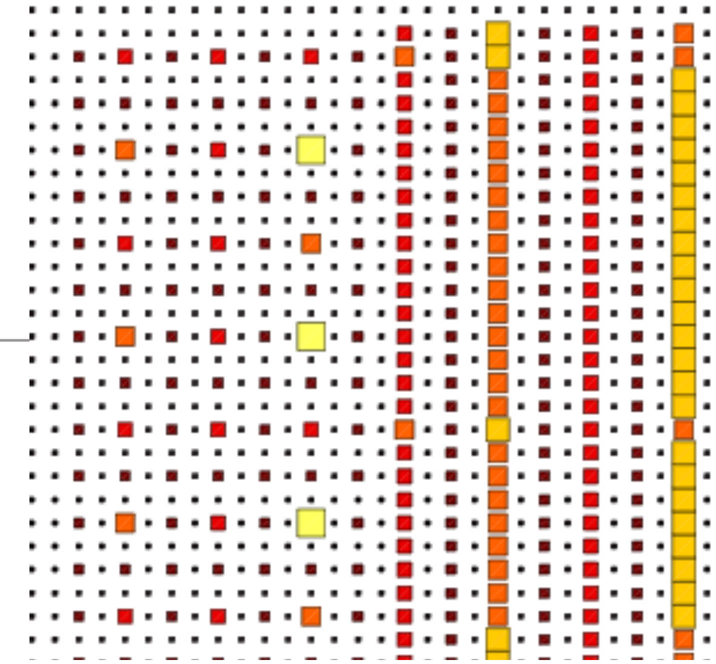


CF-AMG coarse grids

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

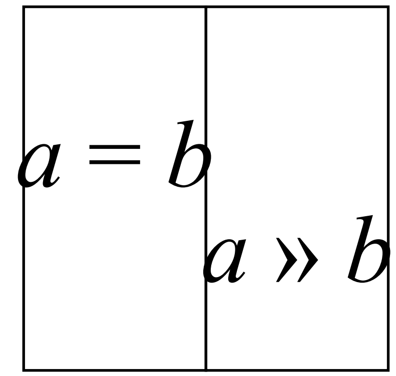
Convergence factor: factor by which the norm of the residual is reduced in each iteration

Example CF-AMG results



CF-AMG coarse grids

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



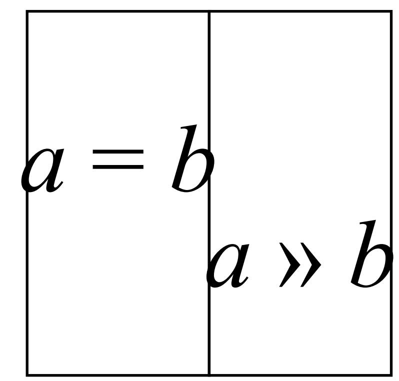
N	Iters	Conv factor	Coarse grids	Grid comp	Oper comp	Setup time	Solve time
61×61	10	0.23	6	1.6	1.6	0.01	0.02
121×121	9	0.23	8	1.6	1.7	0.05	0.07
241×241	9	0.23	9	1.6	1.7	0.25	0.32
481×481	9	0.23	12	1.7	1.7	1.02	1.27
961×961	11	0.29	13	1.7	1.7	4.42	6.28

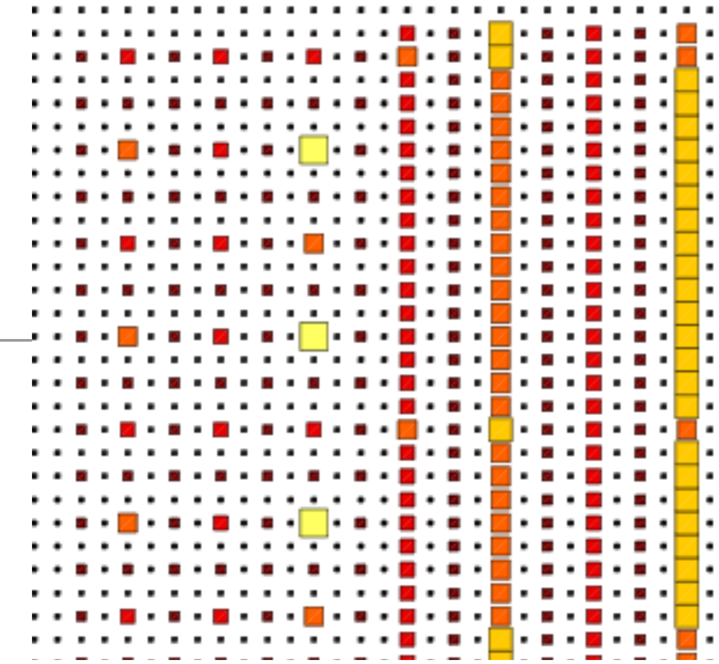
Coarse grids: number of coarse "grids"

Slide credit: R. Falgout, 2/1/2016

Example CF-AMG results

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





CF-AMG coarse grids

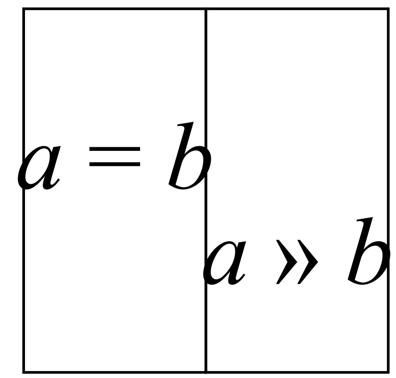
N	Iters	Conv factor	Coarse grids	Grid comp	Oper comp	Setup time	Solve time
61×61	10	0.23	6	1.6	1.6	0.01	0.02
121×121	9	0.23	8	1.6	1.7	0.05	0.07
241×241	9	0.23	9	1.6	1.7	0.25	0.32
481×481	9	0.23	12	1.7	1.7	1.02	1.27
961×961	11	0.29	13	1.7	1.7	4.42	6.28

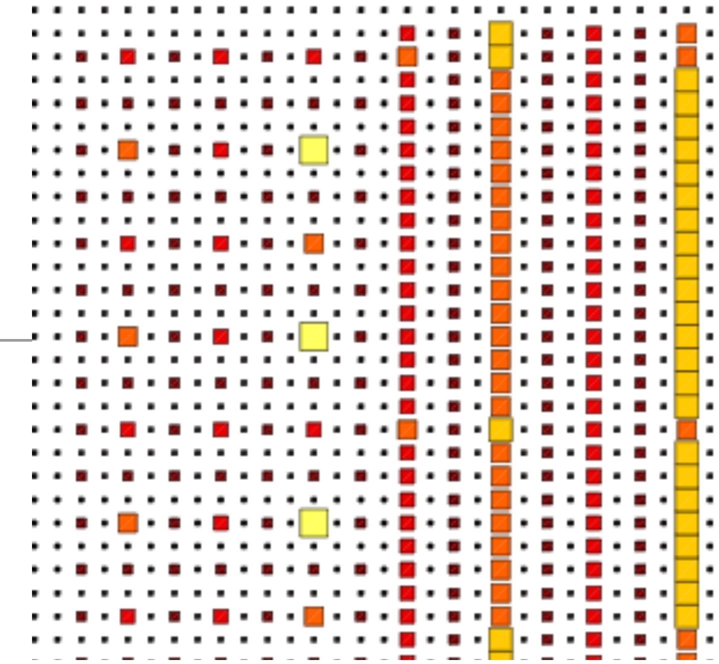
Grid complexity: total number of grid points (system size) on all levels / number of grids points on the fine level

Slide credit: R. Falgout, 2/1/2016

Example CF-AMG results

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





CF-AMG coarse grids

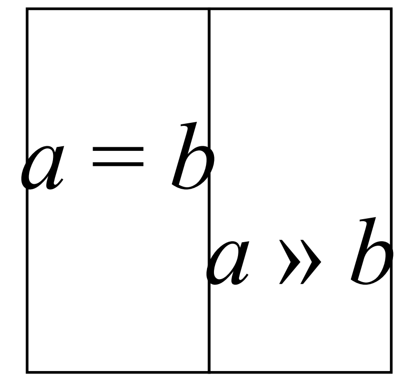
N	Iters	Conv factor	Coarse grids	Grid comp	Oper comp	Setup time	Solve time
61×61	10	0.23	6	1.6	1.6	0.01	0.02
121×121	9	0.23	8	1.6	1.7	0.05	0.07
241×241	9	0.23	9	1.6	1.7	0.25	0.32
481×481	9	0.23	12	1.7	1.7	1.02	1.27
961×961	11	0.29	13	1.7	1.7	4.42	6.28

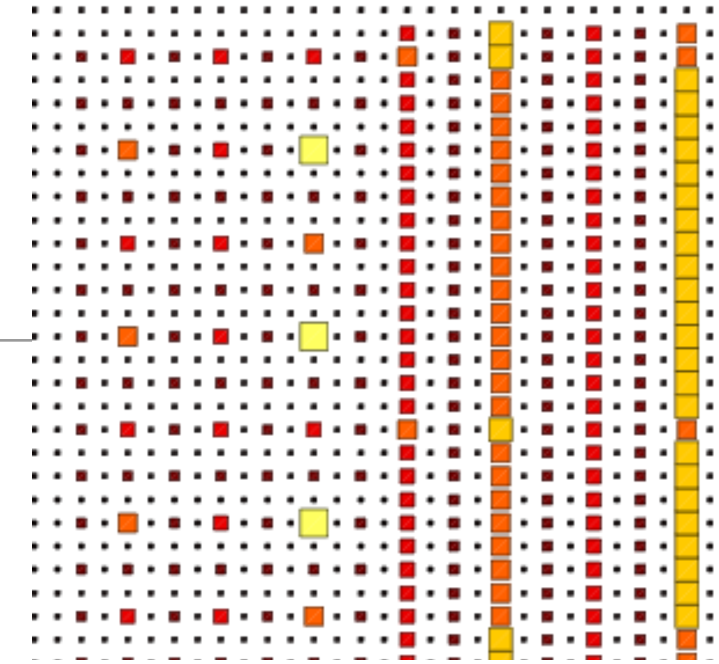
Operator complexity: total number non-zeros on all levels / number of non-zeros on the fine level

Slide credit: R. Falgout, 2/1/2016

Example CF-AMG results

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





CF-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

Setup/solve times: can vary *a lot* depending on the problem, but the setup is significant!

Slide credit: R. Falgout, 2/1/2016

SA AMG

- Smoothed Aggregation based AMG takes a *different* approach

- Still, the same steps:

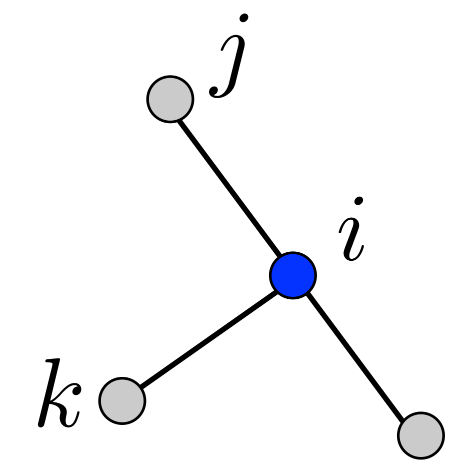
1. strength between points
2. find a coarse grid (this time **aggregates** of points)
3. define interpolation
4. compute the coarse grid operator

P. Vaněk, J. Mandel, M. Brezina, Algebraic multigrid by smoothed aggregation for second and fourth order elliptic problems, Computing, 1996

Symmetric Strength

- i strongly depends on j if

$$-A_{ij} \geq tol * \max_{k \neq i} -A_{ik}$$

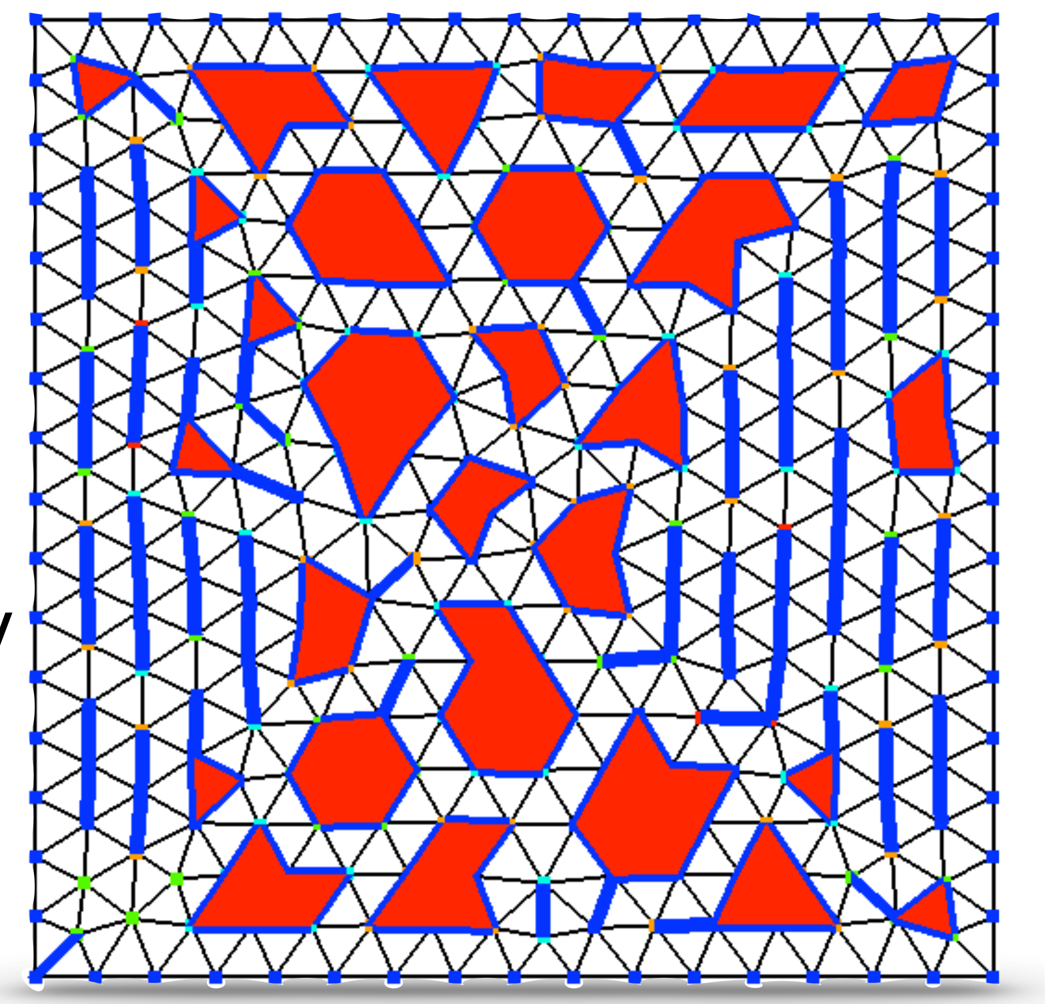


- i strongly depends on j if

$$\frac{|A_{ij}|}{\sqrt{A_{ii}A_{jj}}} \geq tol$$

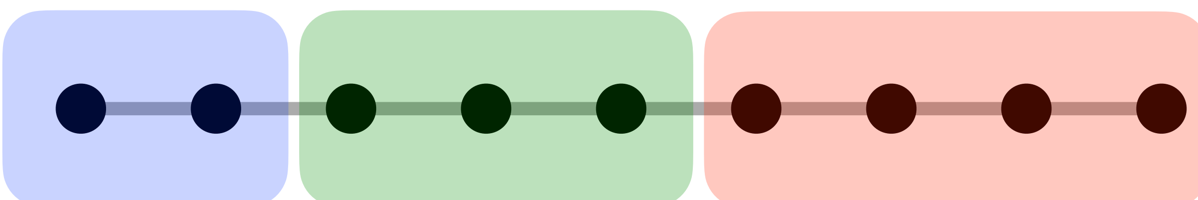
both "think" elliptic

anisotropy



SA AMG Interpolation

- Here we use 1) the aggregation pattern and 2) the candidate vectors



$$AggOpp = \begin{bmatrix} \star & 0 & 0 \\ \star & 0 & 0 \\ 0 & \star & 0 \\ 0 & \star & 0 \\ 0 & \star & 0 \\ 0 & 0 & \star \\ 0 & 0 & \star \\ 0 & 0 & \star \\ 0 & 0 & \star \end{bmatrix}$$

$Q_1 R_1$ $\quad Q_2 R_2$ $\quad Q_3 R_3$

$$B = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \\ B_{31} & B_{32} \\ B_{41} & B_{42} \\ B_{51} & B_{52} \\ B_{61} & B_{62} \\ B_{71} & B_{72} \\ B_{81} & B_{82} \\ B_{91} & B_{92} \end{bmatrix} \rightarrow \hat{T} = \begin{bmatrix} B_{11} & B_{12} & & & & \\ B_{21} & B_{22} & & & & \\ & & B_{31} & B_{32} & & \\ & & B_{41} & B_{42} & & \\ & & B_{51} & B_{52} & & \\ & & & & B_{61} & B_{62} \\ & & & & B_{71} & B_{72} \\ & & & & B_{81} & B_{82} \\ & & & & B_{91} & B_{92} \end{bmatrix} \rightarrow$$

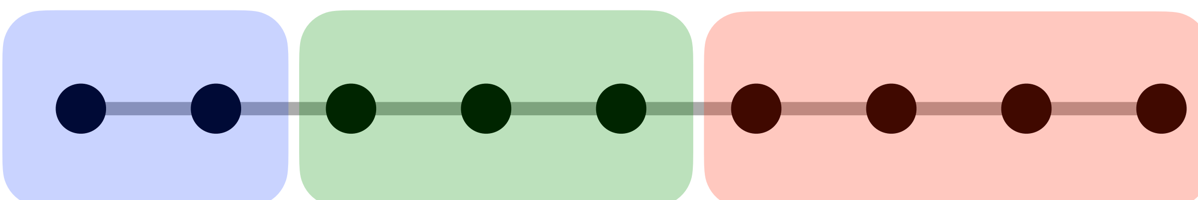
$$T = \begin{bmatrix} Q1 & & \\ & Q2 & \\ & & Q3 \end{bmatrix}$$

$$B_C = \begin{bmatrix} R_1 \\ R_2 \\ R_3 \end{bmatrix}$$

$$\boxed{TB_C = B}$$

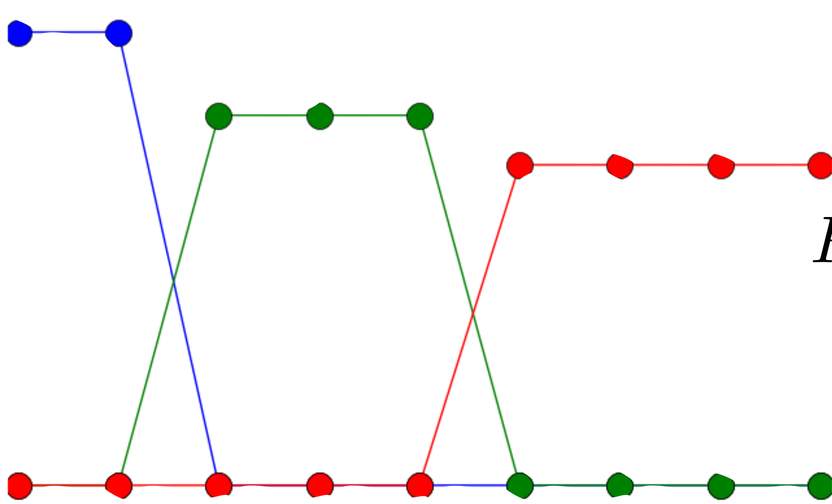
SA AMG Interpolation

- Example

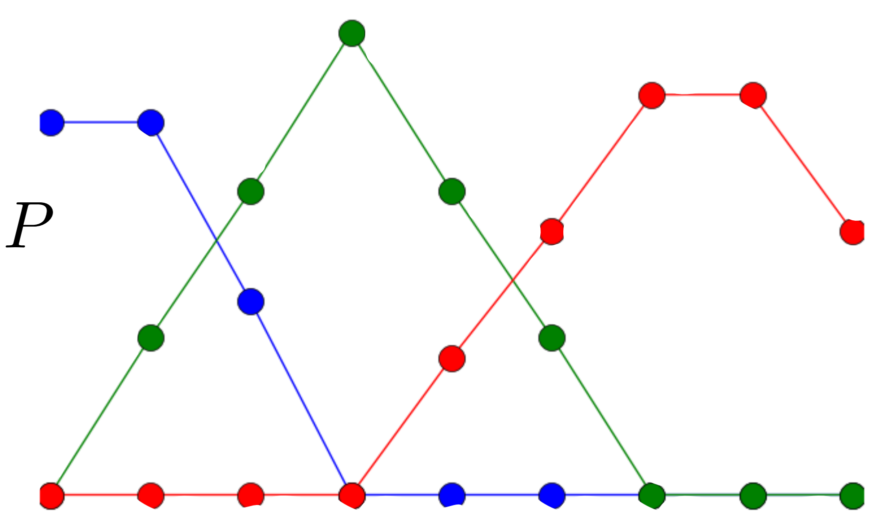


$$\hat{T} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \qquad T = \begin{bmatrix} \frac{\sqrt{2}}{2} & 0 & 0 \\ \frac{\sqrt{2}}{2} & 0 & 0 \\ 0 & \frac{\sqrt{3}}{3} & 0 \\ 0 & \frac{\sqrt{3}}{3} & 0 \\ 0 & \frac{\sqrt{3}}{3} & 0 \\ 0 & 0 & \frac{\sqrt{4}}{4} \\ 0 & 0 & \frac{\sqrt{4}}{4} \\ 0 & 0 & \frac{\sqrt{4}}{4} \\ 0 & 0 & \frac{\sqrt{4}}{4} \end{bmatrix}$$

- Now make interpolation **better**



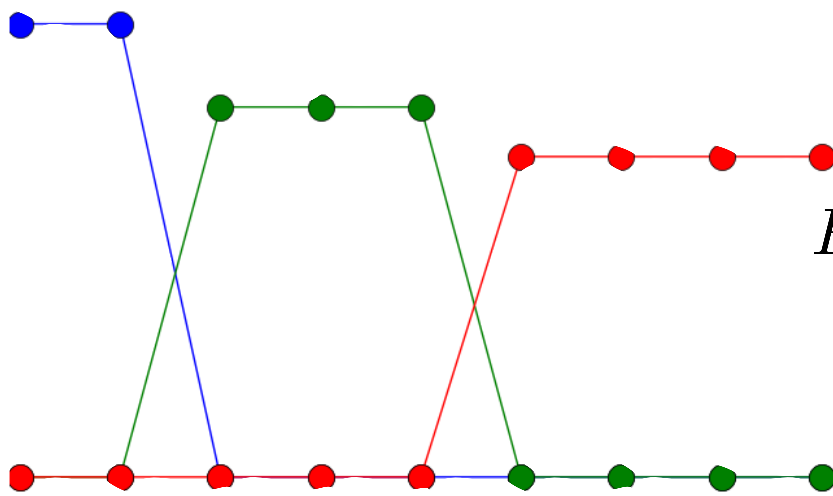
$$P \leftarrow (I - \omega D^{-1} A) P$$



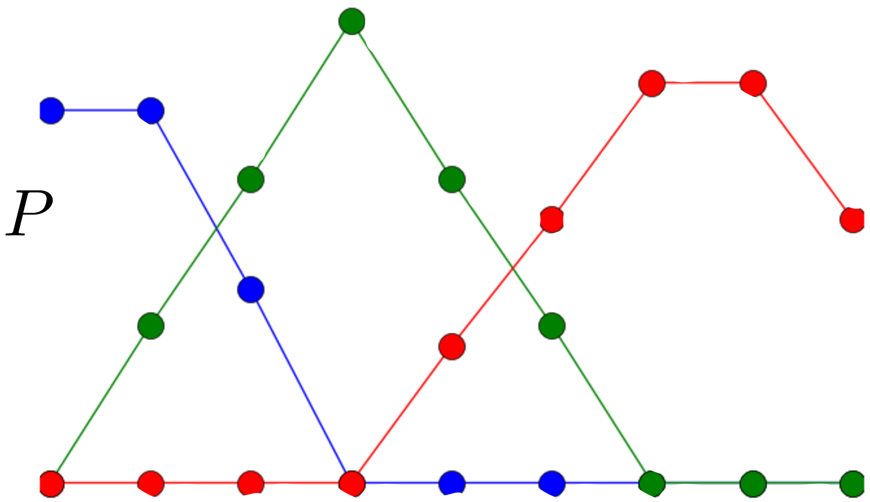
SA AMG Interpolation

- reduce energy
- improve accuracy
- increase complexity

- Improving interpolation



$$P \leftarrow (I - \omega D^{-1} A) P$$



- Makes the columns of P ***smoother***

- Makes the sparsity of P **denser**

Aggregation Based Setup

$\boxed{B}$

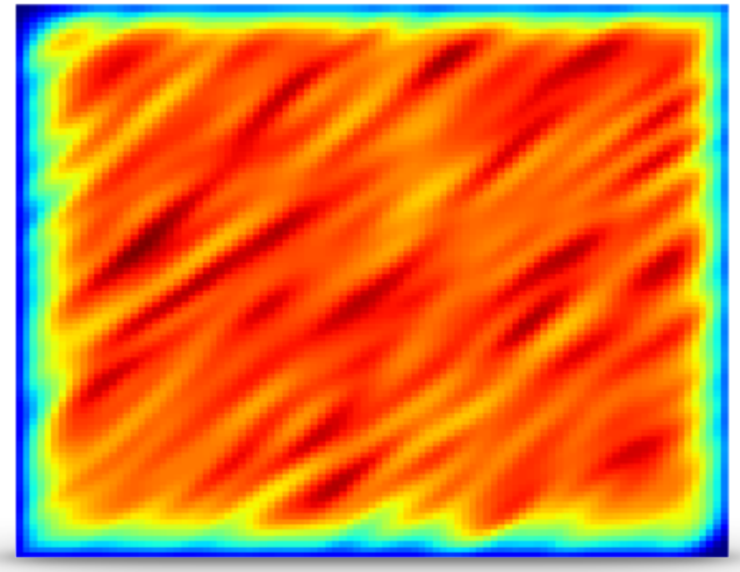
$S \leftarrow \mathrm{strength}(A, B)$

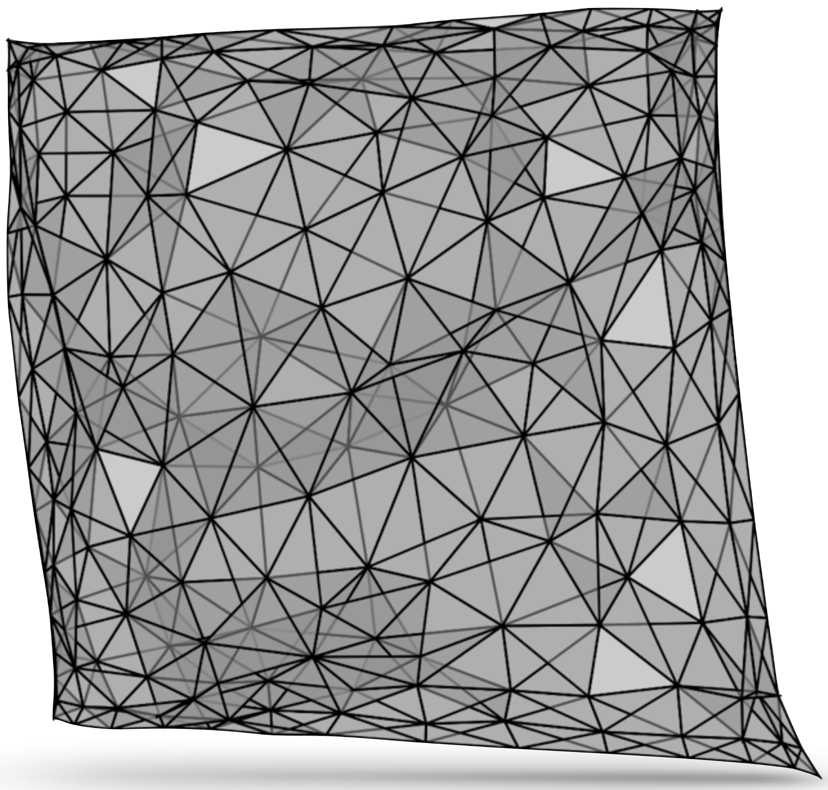
$C \leftarrow \mathrm{aggregate}(S)$

$P^{(0)}, B^C \leftarrow \mathrm{inject}(C, B)$

$P \leftarrow \mathrm{improve}(A, P^{(0)})$

$R = P^* \qquad A^C = RAP$





SA AMG Setup Algorithm

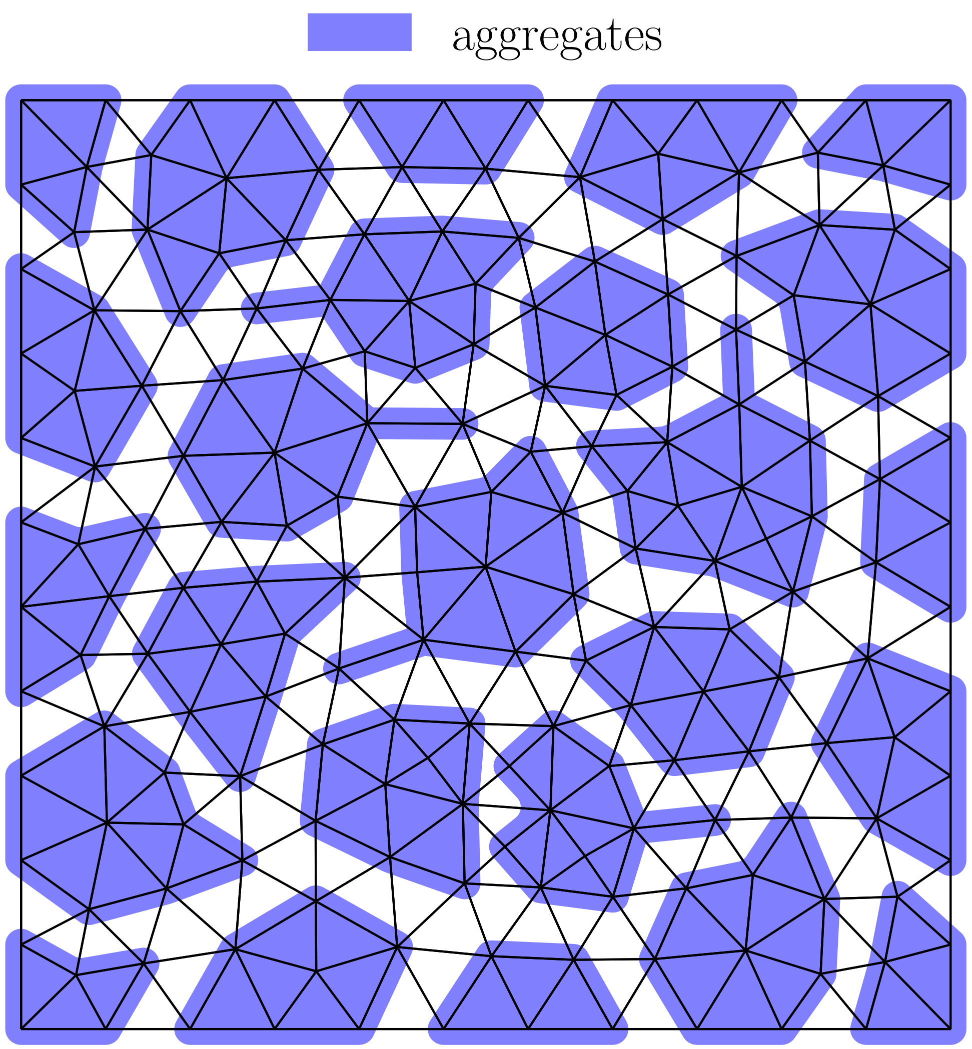
Algorithm 1: `SA_setup()`

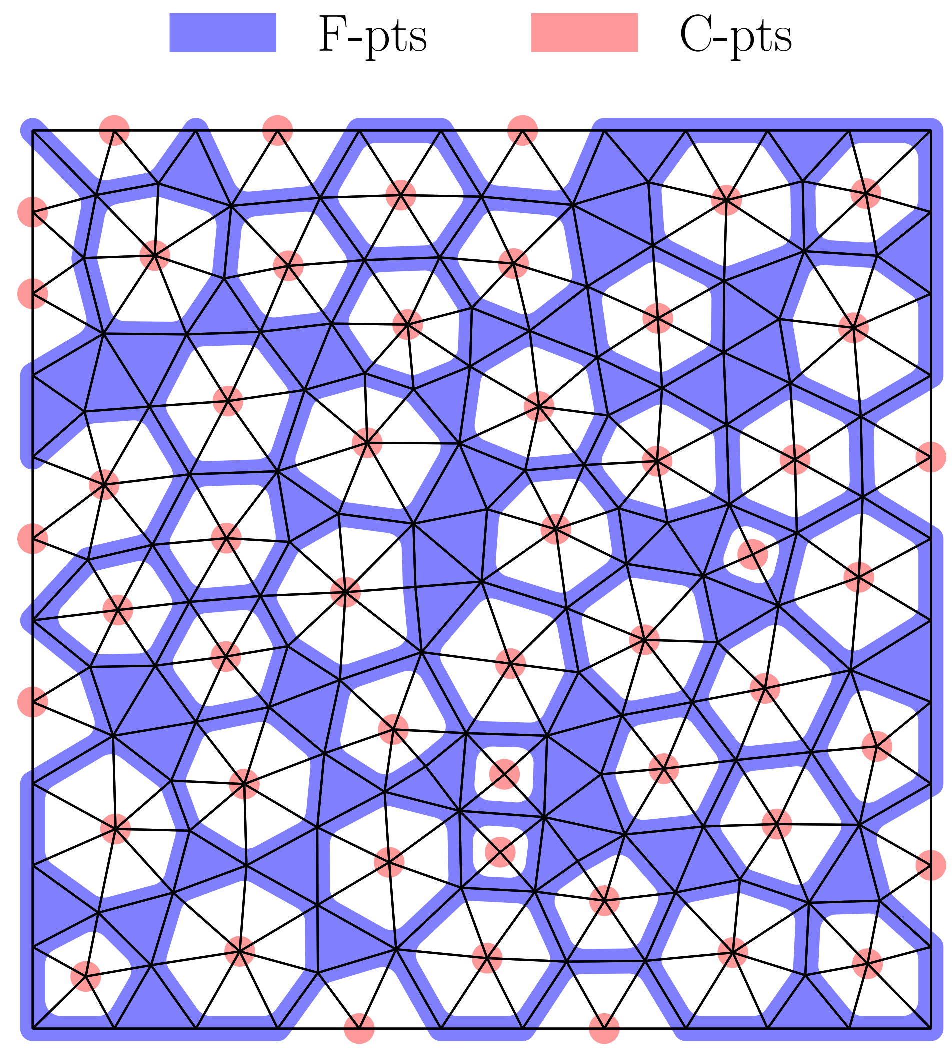
Input: A_0: fine-grid operator
B_0: fine-grid candidate vectors
`max_size`: threshold for max size of coarsest problem

Output: $A_1, \ldots, A_L$,
$P_0, \ldots, P_{L-1}$

$\ell = 0$
while $\texttt{size}(A_\ell) > \texttt{max\_size}$
  $S_\ell = \texttt{strength}(A_\ell)$ {Strength-of-connection}
  $\mathcal{A}_\ell = \texttt{aggregate}(S_\ell)$ {Aggregation}
  $T_\ell,\ B_{\ell+1} = \texttt{inject}(\mathcal{A}_\ell,\ B_\ell)$ {Form tentative interpolation and coarse candidates}
  $P_\ell = \texttt{smooth}(A_\ell,\ T_\ell)$ {Smooth T_ℓ}
  $A_{\ell+1} = P_\ell^T A_\ell P_\ell$ {Coarse-grid operator}
  $\ell = \ell + 1$

AMG





- Smoothed Aggregation AMG (SA-AMG)
- Interpolation constructed from candidate vectors
- Clear approach to *optimize* interpolation

- Coarse-Fine AMG (CF-AMG) or Ruge-Stüben
- Edge-wise construction of interpolation, allowing straightforward control of sparsity
- Incorporating near-nullspace is not straightforward

Theory

Bounds on convergence guide the design of methods.

- strength of connection
- coarse grids
- interpolation
- etc

Look at the Operators

- Smoothing

$$\mathbf{u} \leftarrow G\mathbf{u} + (I - G)A^{-1}\mathbf{f} \qquad \text{or} \quad \mathbf{e} \leftarrow G\mathbf{e}.$$

$$G = I - \omega D^{-1} A$$

- Coarse-grid Correction

$$\mathbf{e} \leftarrow \underbrace{\left(I - P\left(P^T A P\right)^{-1} P^T A\right)}_{T} \mathbf{e}$$

Generalized AMG Theory

- Smoother (symmetric or non-symmetric)

$$e \leftarrow (I - M^{-1}A)$$

- Assume s.p.d. A and $M + M^T - A$

- The *symmetrized* smoother is given as

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

- or

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

Robert D. Falgout and Panayot S. Vassilevski, On Generalizing the Algebraic Multigrid Framework, 2004

Slide credit: R. Falgout, 2/1/2016

Generalized AMG Theory

- **Interpolation**: $$P : \mathbb{R}^{n_c} \rightarrow \mathbb{R}^n$$
- Some **restriction** (not MG restricion):

$$R : \mathbb{R}^n \rightarrow \mathbb{R}^{n_c}$$

- Define such that

$$PR = I$$

Here, RP is a **projection** onto the range of P

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

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

- Define the **two-grid multigrid** error propagator by

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

Slide credit: R. Falgout, 2/1/2016

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

- Theorem: For any projection PR

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

- 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: Pick a coarse grid and interpolation P

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

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

Slide credit: R. Falgout, 2/1/2016

Adaptive and Bootstrap ***AMG***: 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

- ***Bootstrap AMG*** is an adaptive method

Slide credit: R. Falgout, 2/1/2016

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
- Helpful analysis tool

Brannick, Brezina, Keyes, Livne, Livshits, MacLachlan, Manteuffel, McCormick, Ruge, and Zikatanov, "Adaptive smoothed aggregation in lattice QCD," Springer (2006)

Brezina, Falgout, MacLachlan, Manteuffel, McCormick, and Ruge, "Adaptive smoothed aggregation (αSA)," SIAM J. Sci. Comput. (2004)

Slide credit: R. Falgout, 2/1/2016

Other AMG methods

- AMGe
  - Use local stiffness matrices to define smooth error

M. Brezina, A. J. Cleary, R. D. Falgout, V. E. Henson, J. E. Jones, T. A. Manteuffel,S. F. McCormick, and J. W. Ruge, Algebraic multigrid based on element interpolation (AMGe), SIAM J. Sci. Comput., 2000

- Element-Free AMGe
  - Builds "local element matrices" directly from global matrix
  - Requires the definition of an extension map

V. E. Henson and P. S. Vassilevski, Element-free AMGe: General algorithms for computing interpolation weights in AMG, SIAM J. Sci. Comput., 2001

- Spectral AMGe
  - Based on a variant of the AMGe measure
  - Coarse dofs are no longer simple subsets of fine-grid dofs
  - Requires solution of local eigenvalue problems

T. Chartier, R. D. Falgout, V. E. Henson, J. Jones, T. Manteuffel, S. McCormick, J. Ruge, and P. S. Vassilevski, Spectral AMGe (ρAMGe), SIAM J. Sci. Comput., 2003

Slide credit: R. Falgout, 2/1/2016