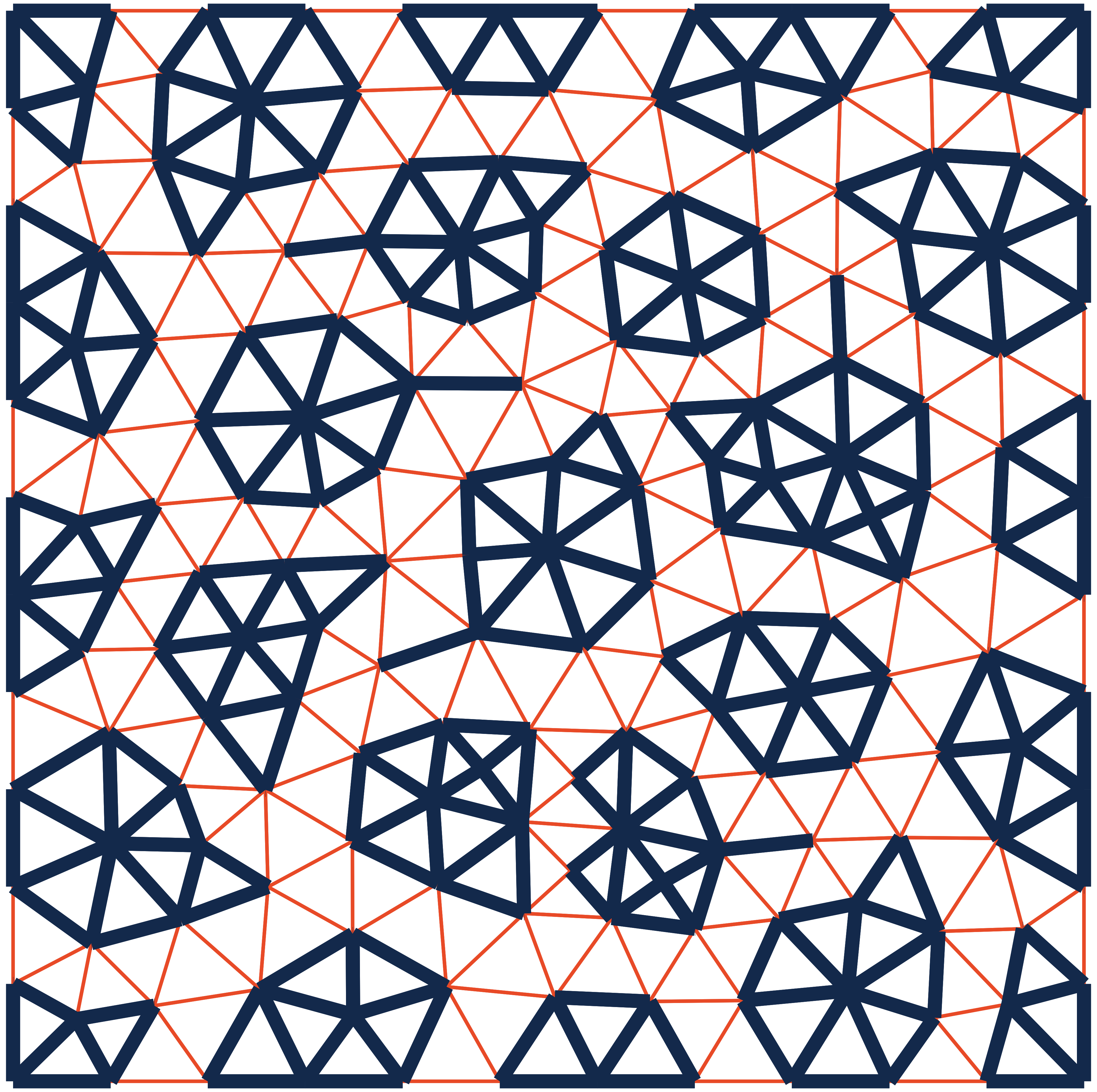
Multigrid Methods — An Overview

Lecture 4: AMG

Scott MacLachlan
Department of Mathematics and Statistics
Memorial University of Newfoundland

Credit to:
Luke Olson
Department of Computer Science
University of Illinois at Urbana-Champaign



Objectives — high level

1. **Lecture 1 - Basics**
   - Basic mechanics of a multigrid method
   - 1D, 2D, Poisson
2. **Lecture 2 - Extensions**
   - What can go wrong and how to fix it
3. **Lecture 3 - Multigrid for Systems**
   - How to extend these ideas to coupled systems
4. **Lecture 4 - Algebraic**
   - What to do if we do not have a grid (hierarchy)

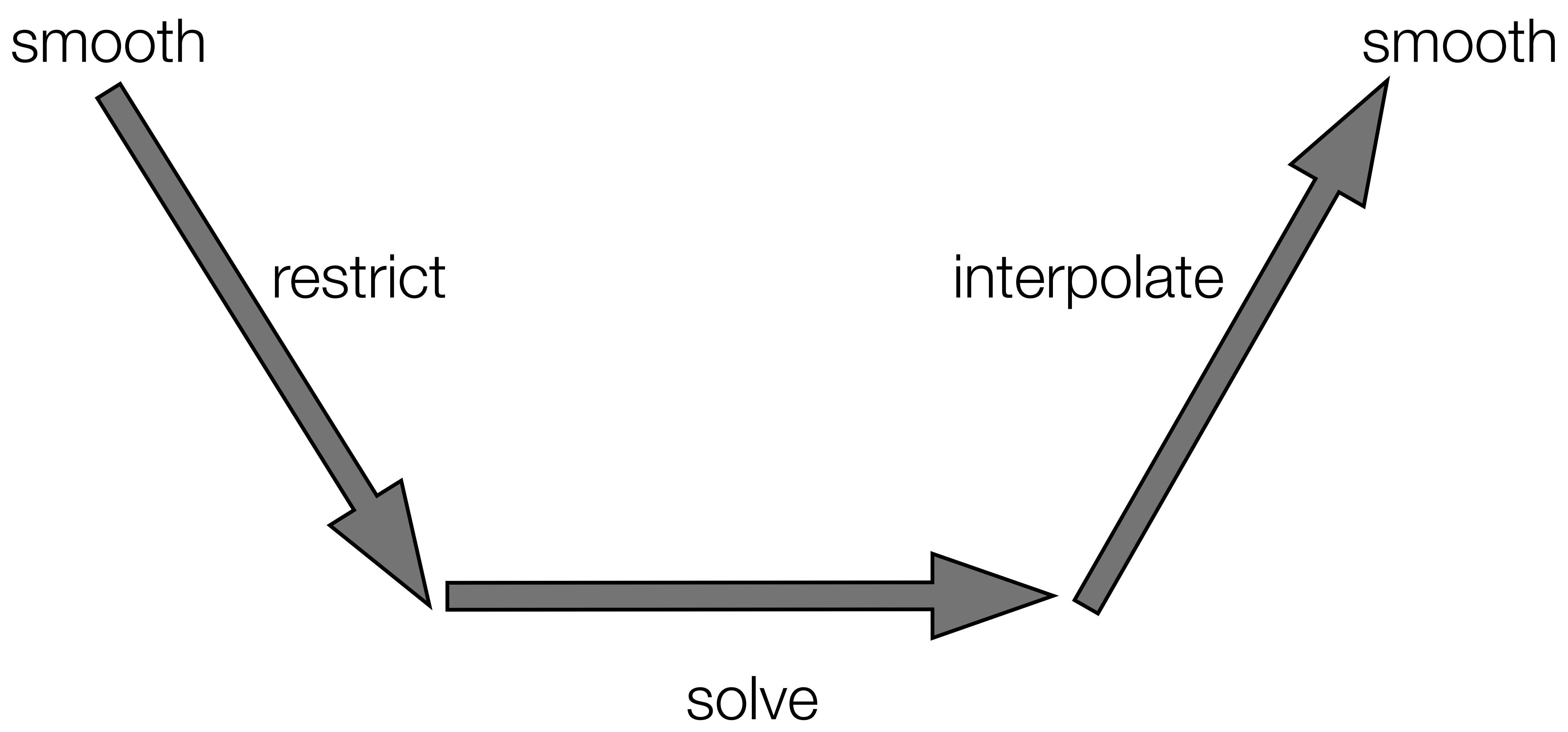
Objectives — today

1. **Lecture 4 - algebraic multigrid**
    - Outline the basic components of an **algebraic** method
    - Compare and contrast different “styles” of AMG
    - Highlight advanced features such as interpolation

Algorithm: two-level multigrid

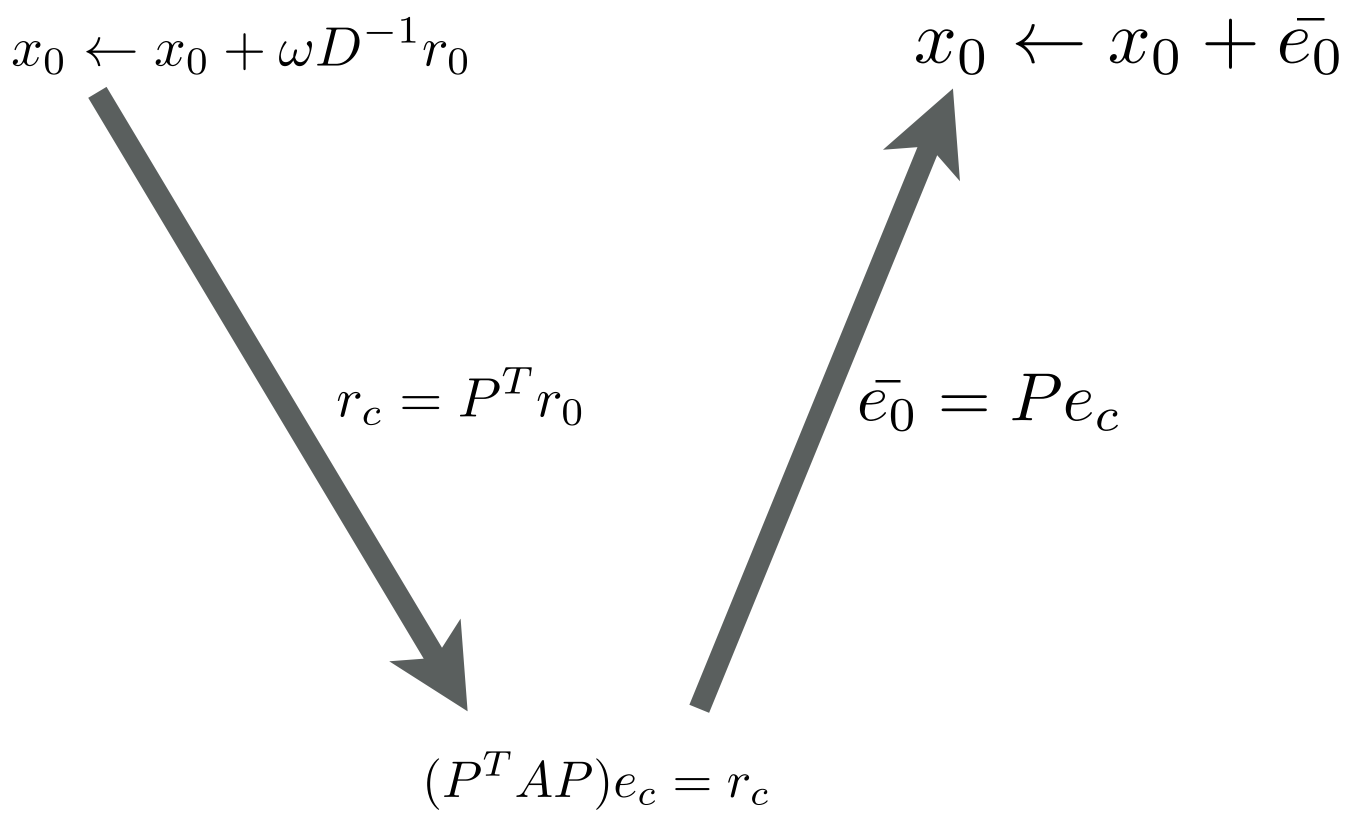
Input: initial guess

1. Smooth ν_{pre} times on $Au = f$
2. Compute $r = f - Au$
3. Compute $r_c = Rr$
4. Solve $A_c e_c = r_c$
5. Interpolate $\hat{e} = P e_c$
6. Correct $u \leftarrow u + \hat{e}$
7. Smooth ν_{post} times on $Au = f$



A two-level “V” cycle

Algebraic Observation



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

coarse grid correction **relax**

relax

residual

restrict

coarse solve

interpolate

correct

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

interpolation should capture what relaxation misses

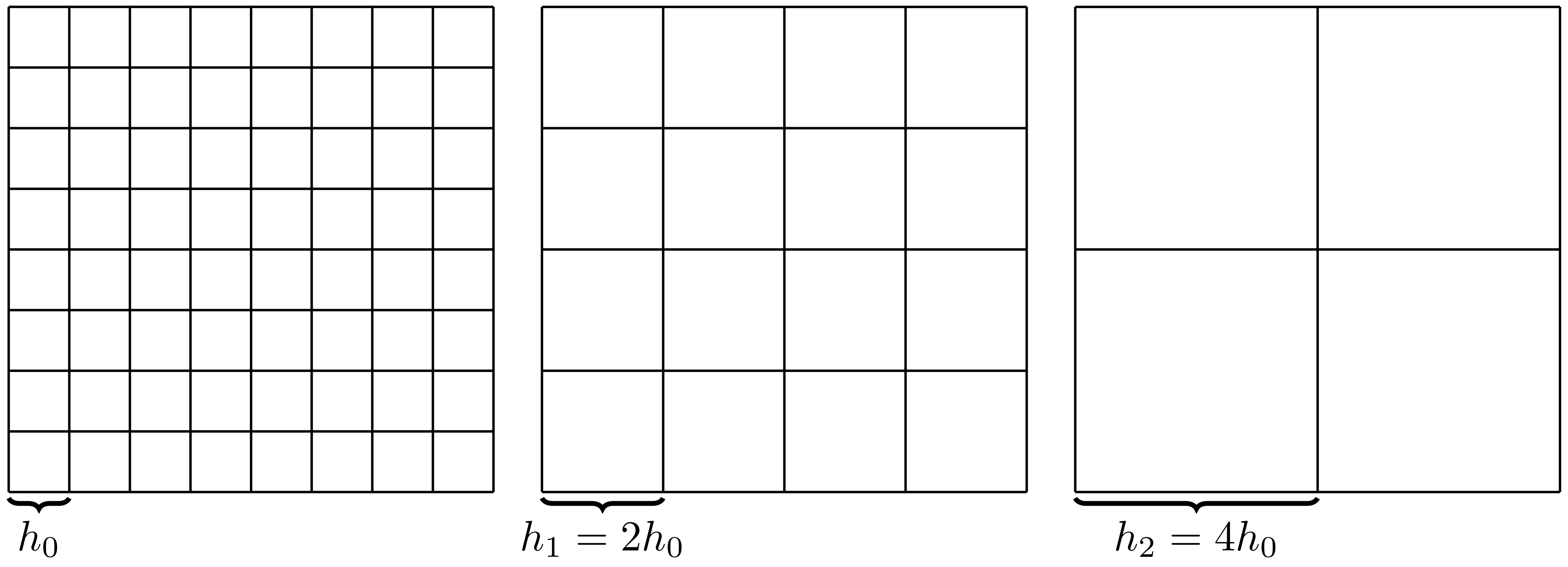
Multigrid Basics

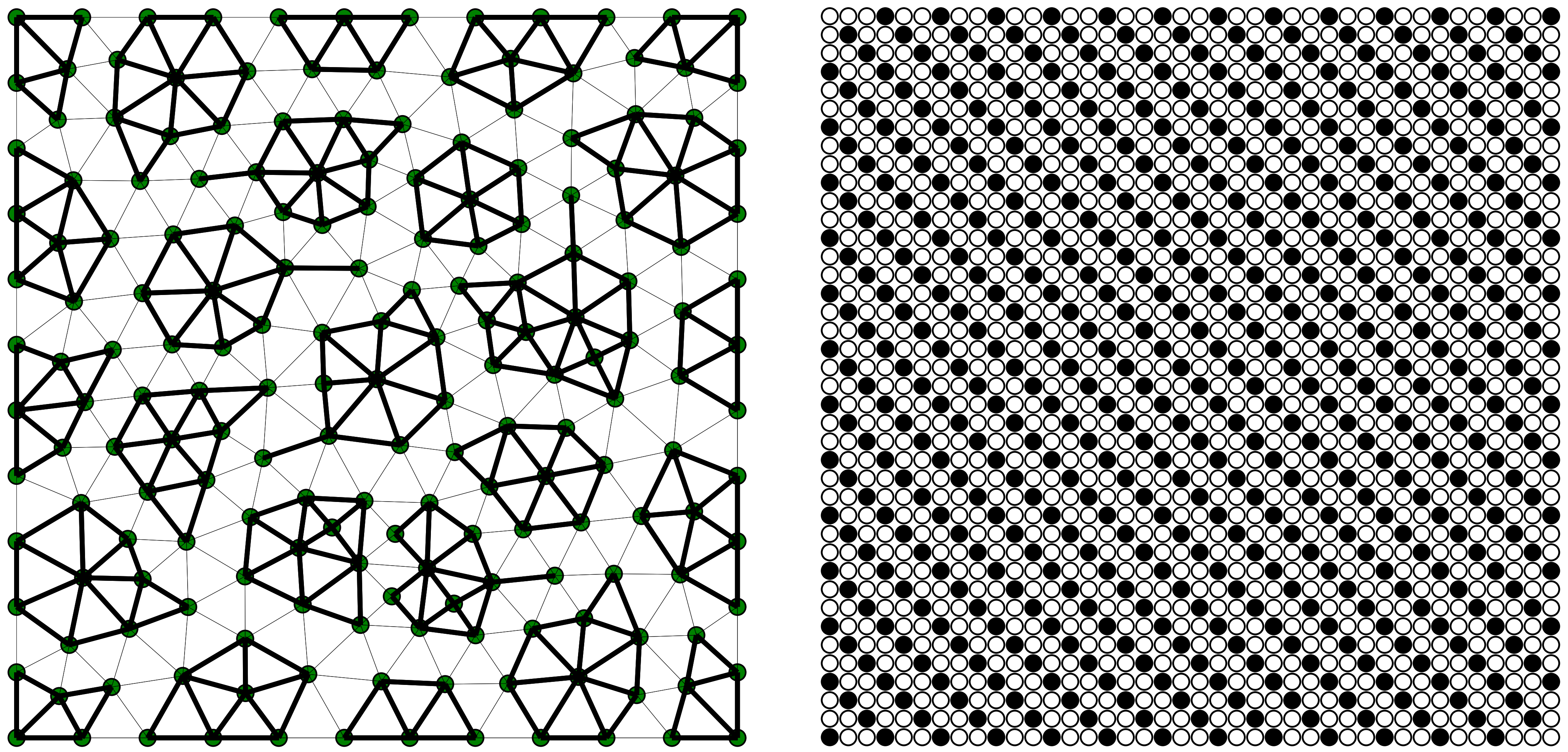
Geometric

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

Algebraic

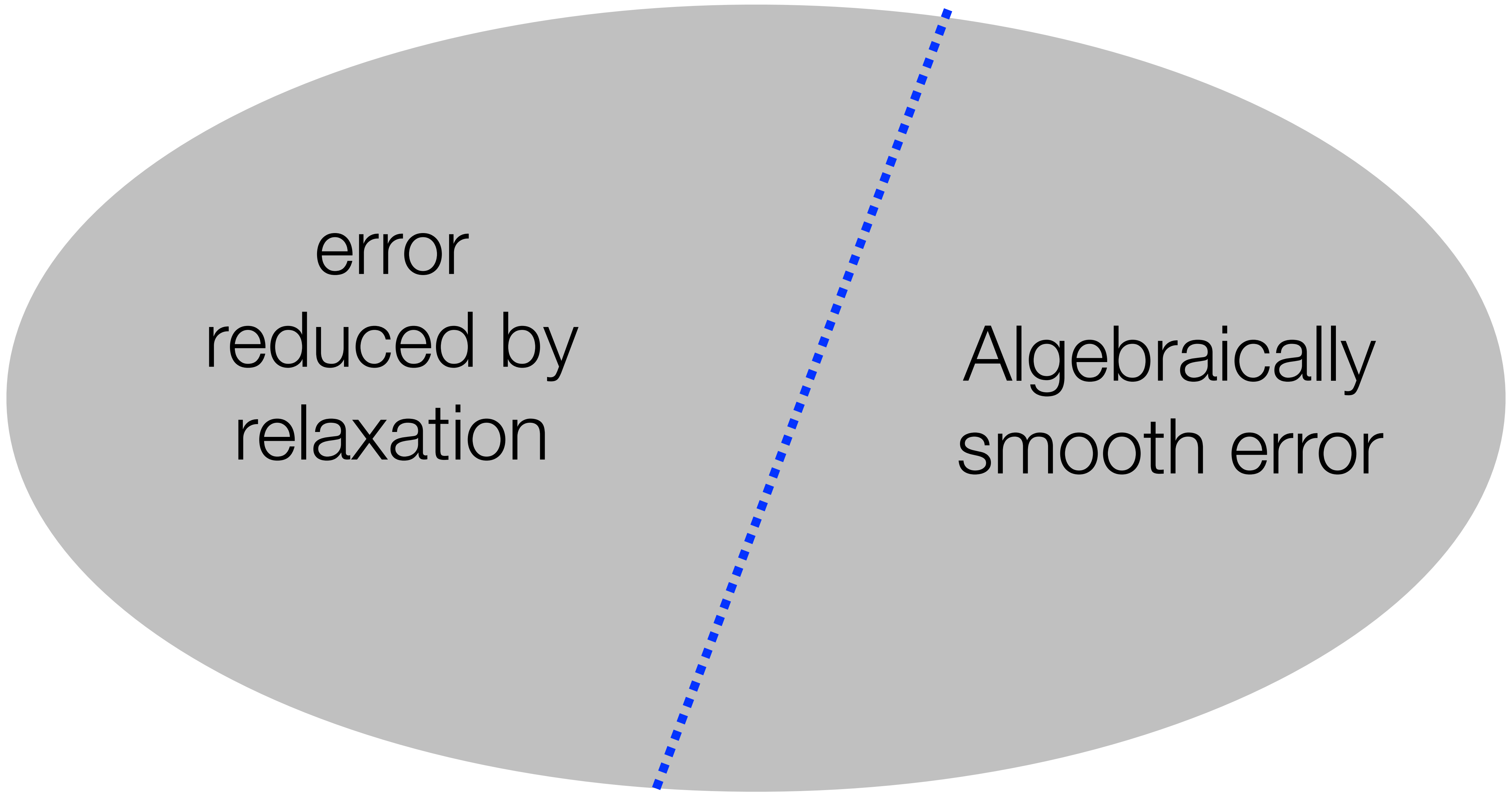
- Relaxation method is fixed
- Find coarse grids
- construct interpolation





Algebraically Smooth Error

- **“Algebraically smooth” error may not be geometrically smooth**



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

$$
\begin{aligned}
\langle Ae, e\rangle &= \sum_i e_i(A_{ii}e_i + \sum_{j\neq i} A_{ij}e_j) \\
&= \sum_i e_i \left(\sum_{j\neq i} -A_{ij}(e_i - e_j) \right) \\
&= \sum_{i<j} -A_{ij}\cdot e_i \cdot (e_i - e_j) + \sum_{i>j} -A_{ij}\cdot e_i \cdot (e_i - e_j) \\
&= \sum_{i<j} -A_{ij}\cdot e_i \cdot (e_i - e_j) - \sum_{i<j} -A_{ij}\cdot e_i \cdot (e_i - e_j) \\
&= \sum_{i<j} -A_{ij}\cdot (e_i - e_j)^2
\end{aligned}
$$

assume zero row sum

swap i, j

- 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

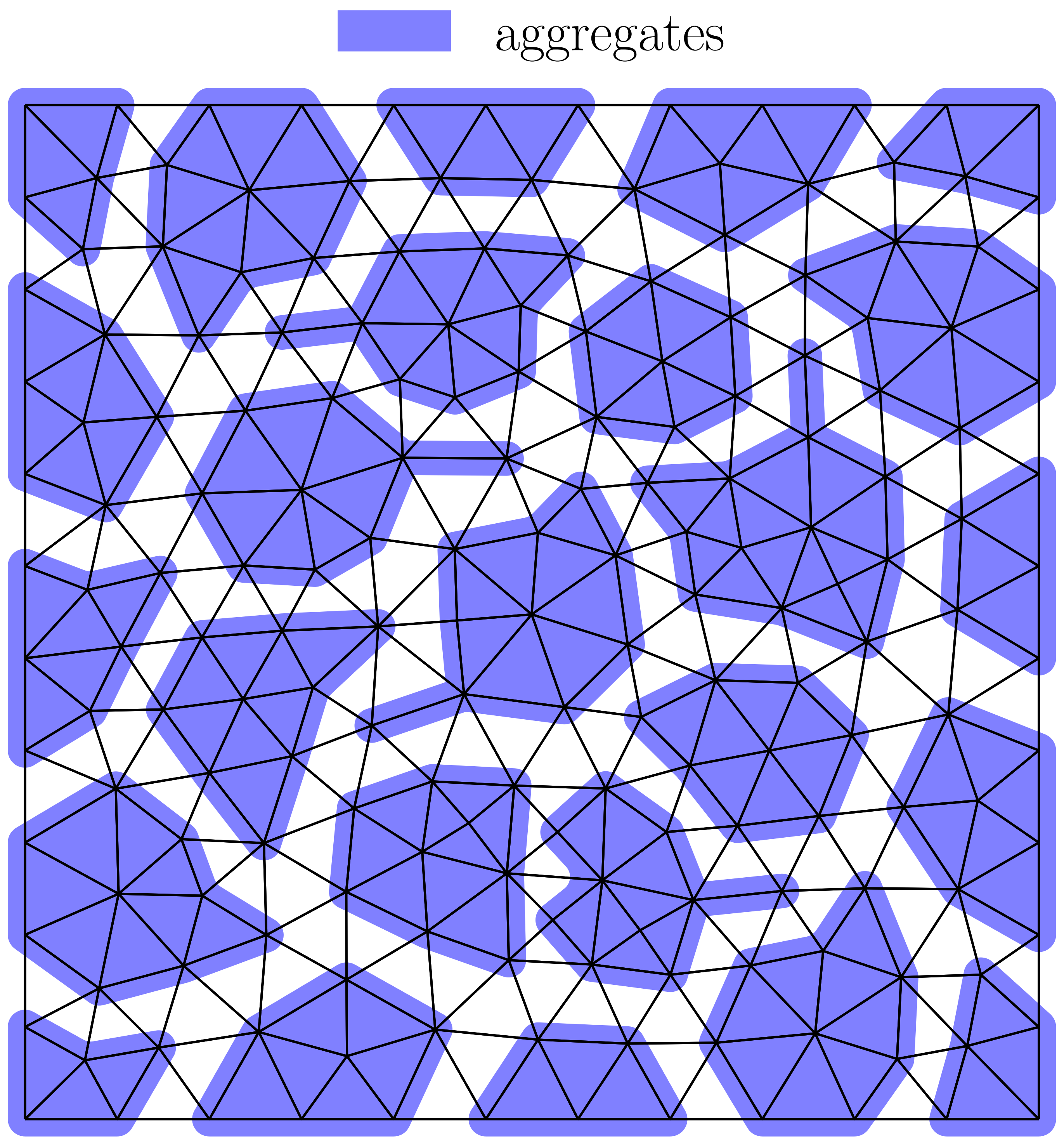
$$-a_{ij} \geq \theta \max_{k \neq i}\{-a_{ik}\}$$

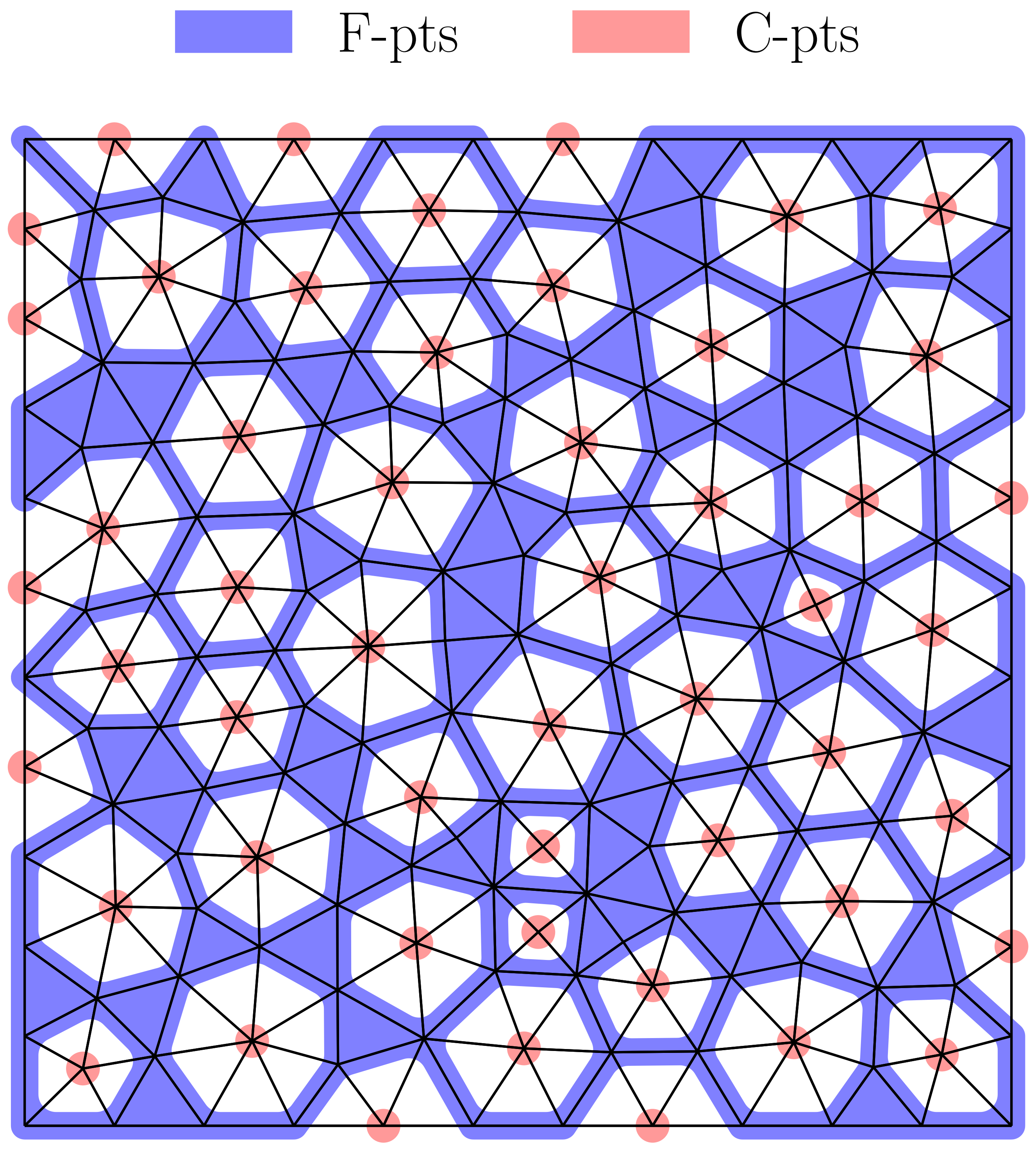
- Often positive off-diagonals are treated as weak
- This definition of strength of connection is not symmetric

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

Demo: `12-AMG-coarse-mesh.ipynb`

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

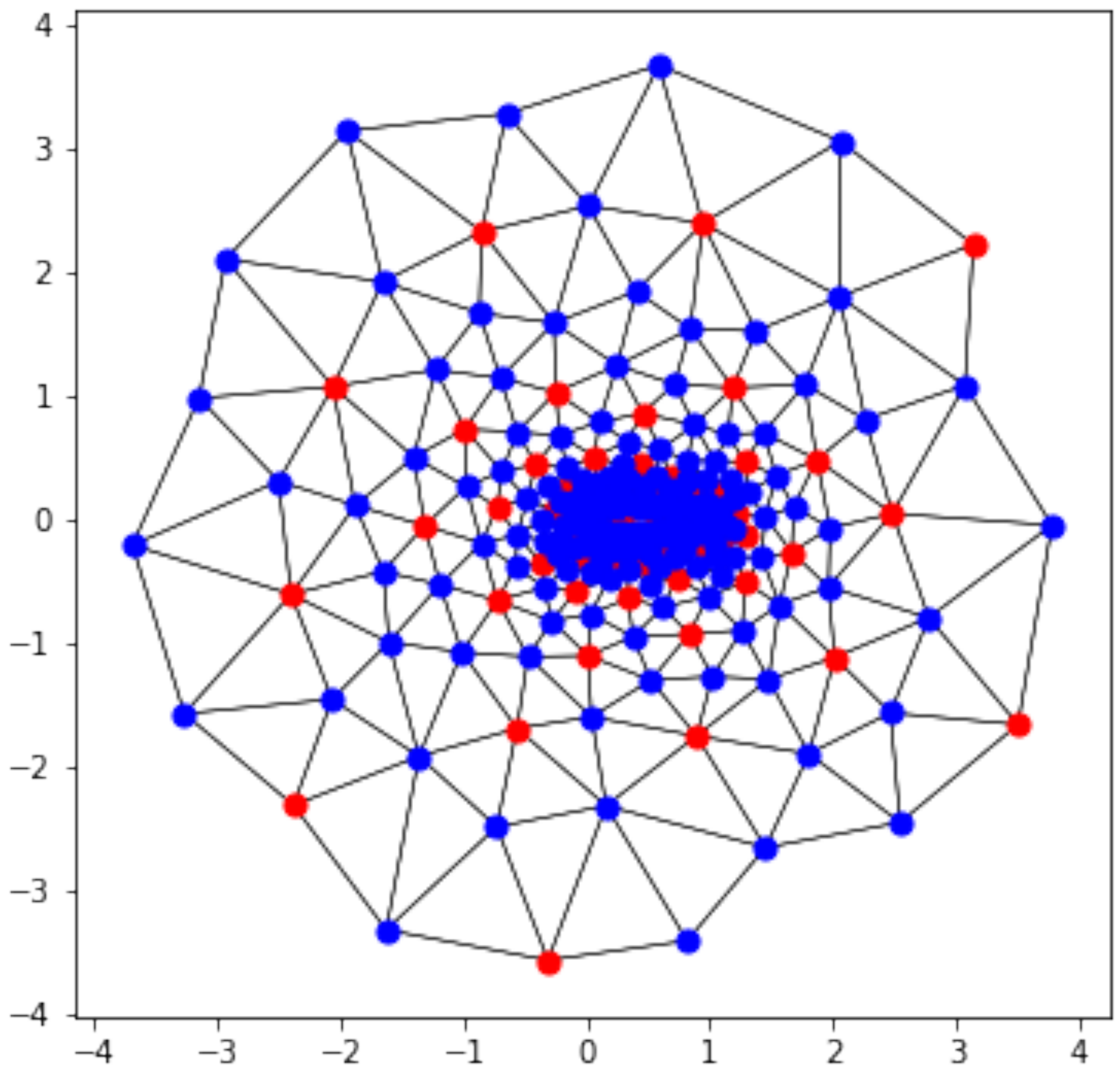
- **C-points:** coarse grid points
- **F-points:** fine grid points
- Either a C-pt or an F-pt

- Coarse interpolatory set

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

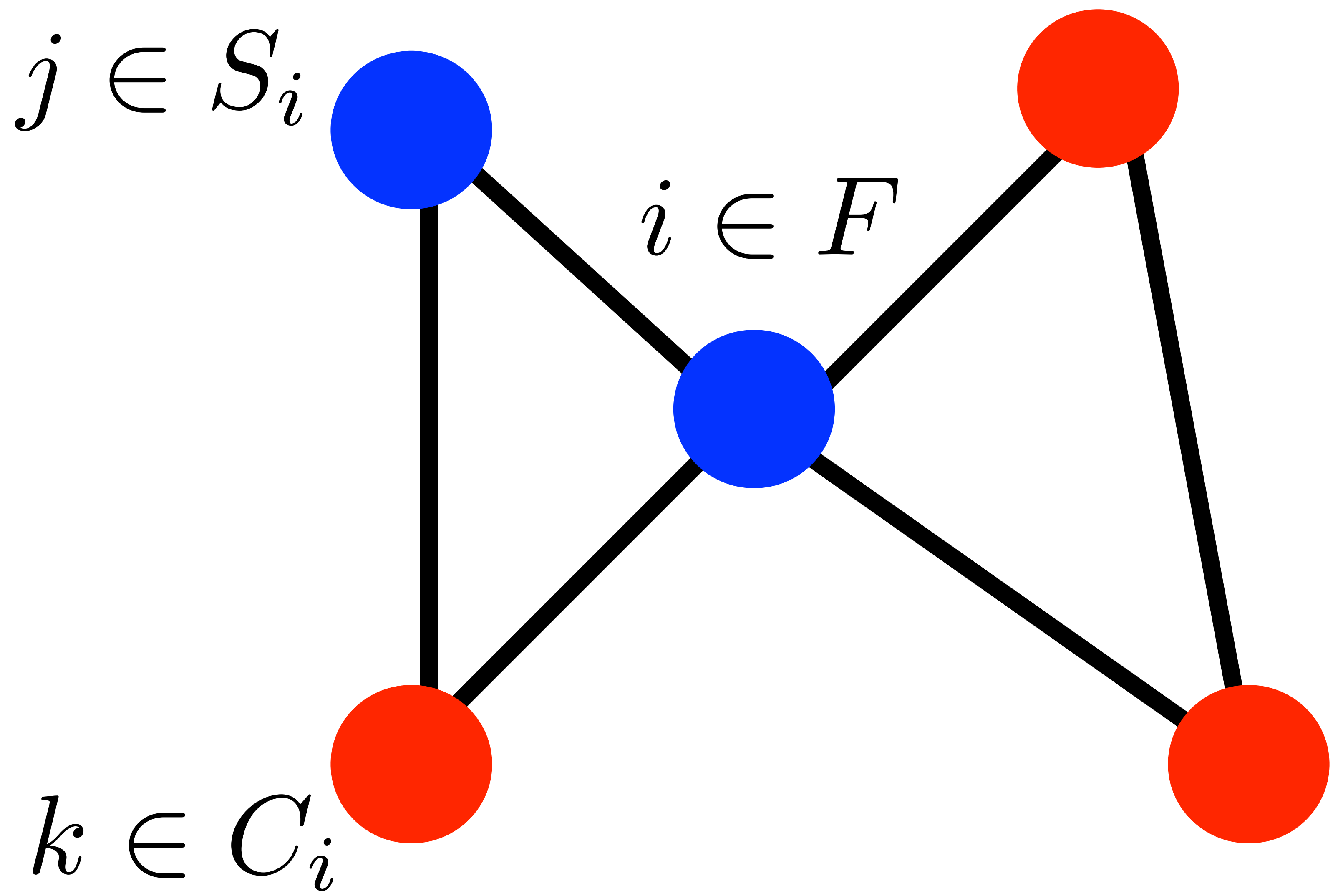
C-pts that are used to interpolated F-pt i.

$$C_i$$



CF AMG

- (C1) Every point that $i \in F$ strongly depends on is 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 strongly 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

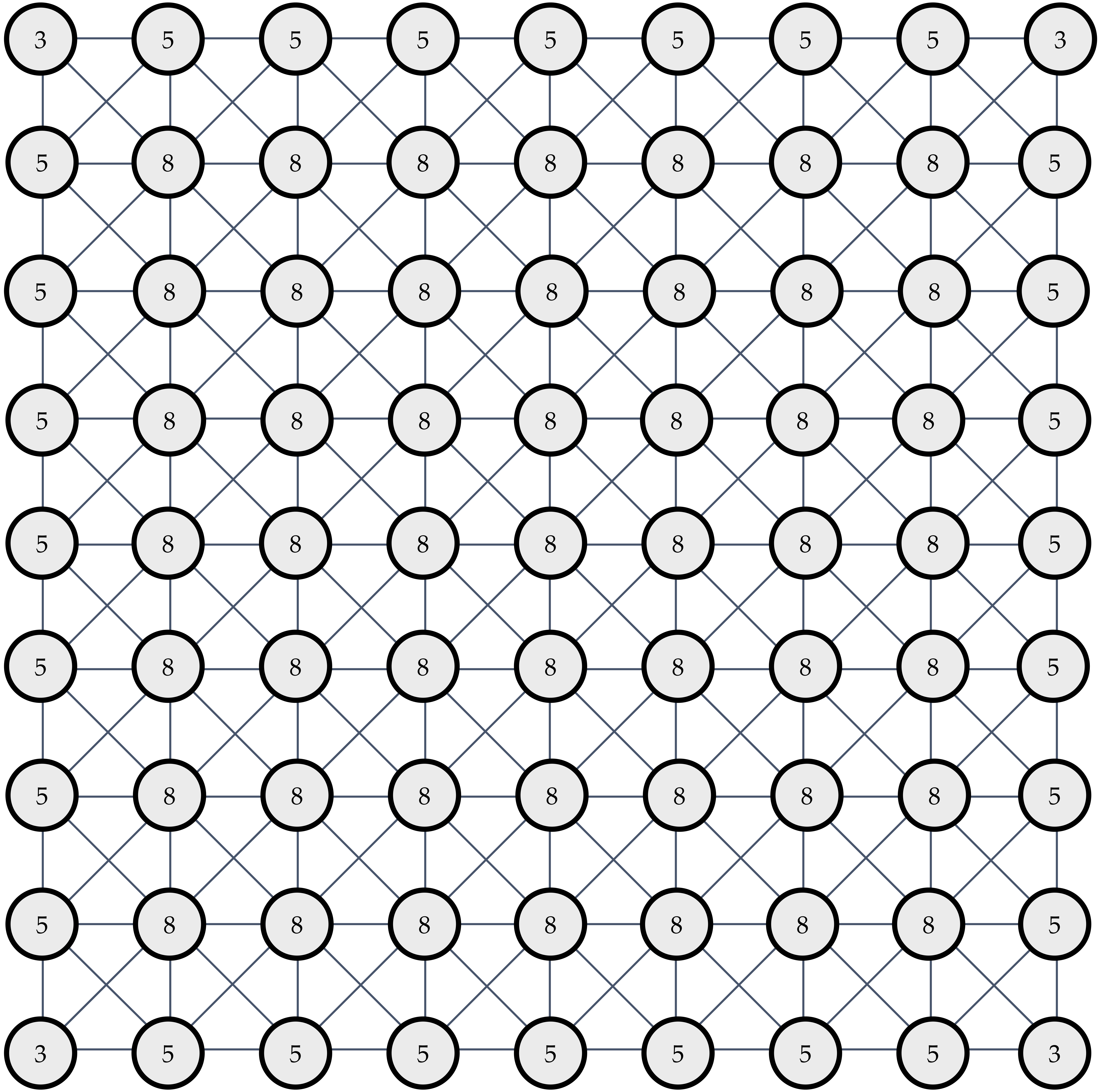
Ruge-Stüben

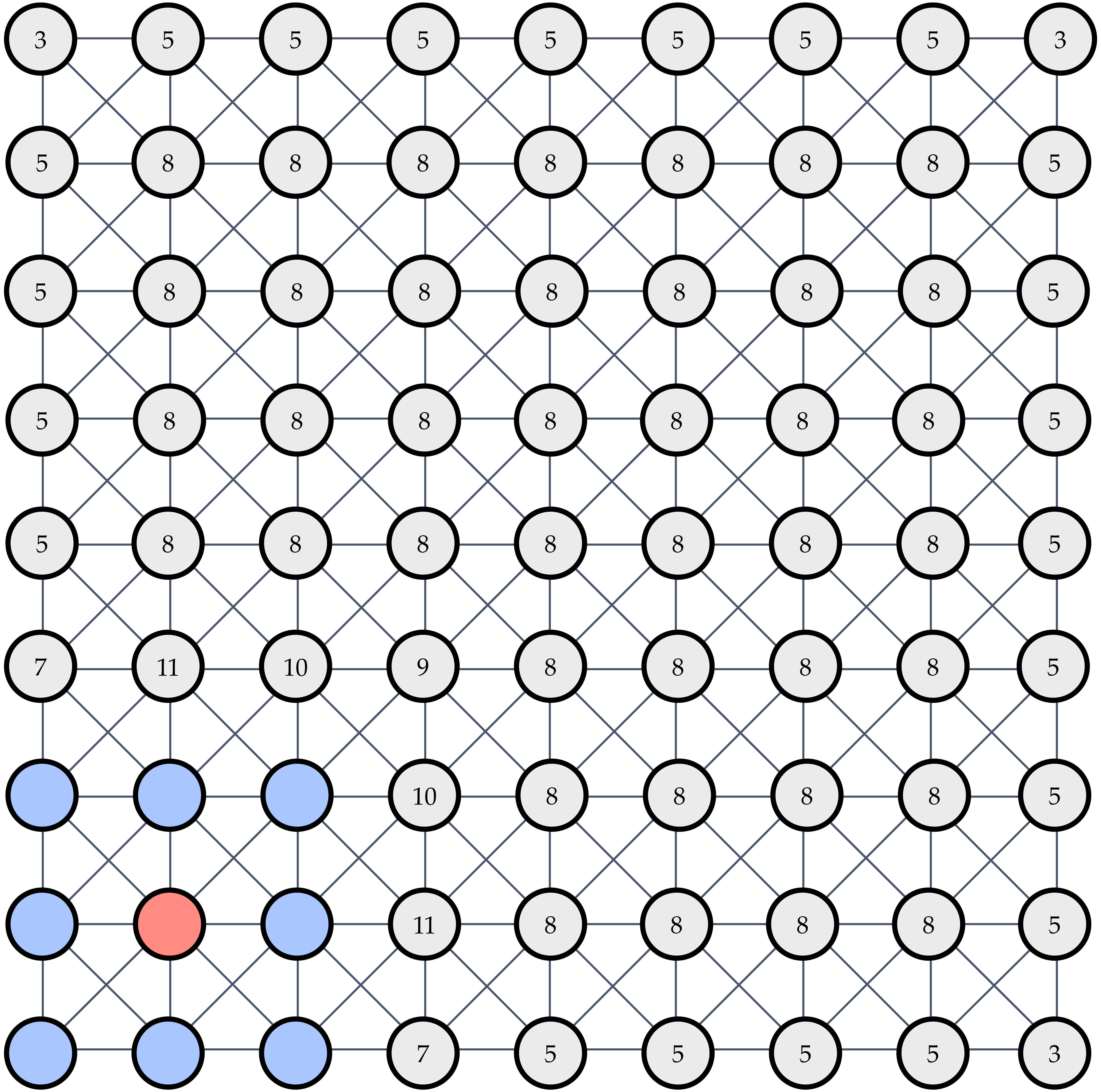
Algorithm 3. RUGE–STÜBEN.

Initialize:

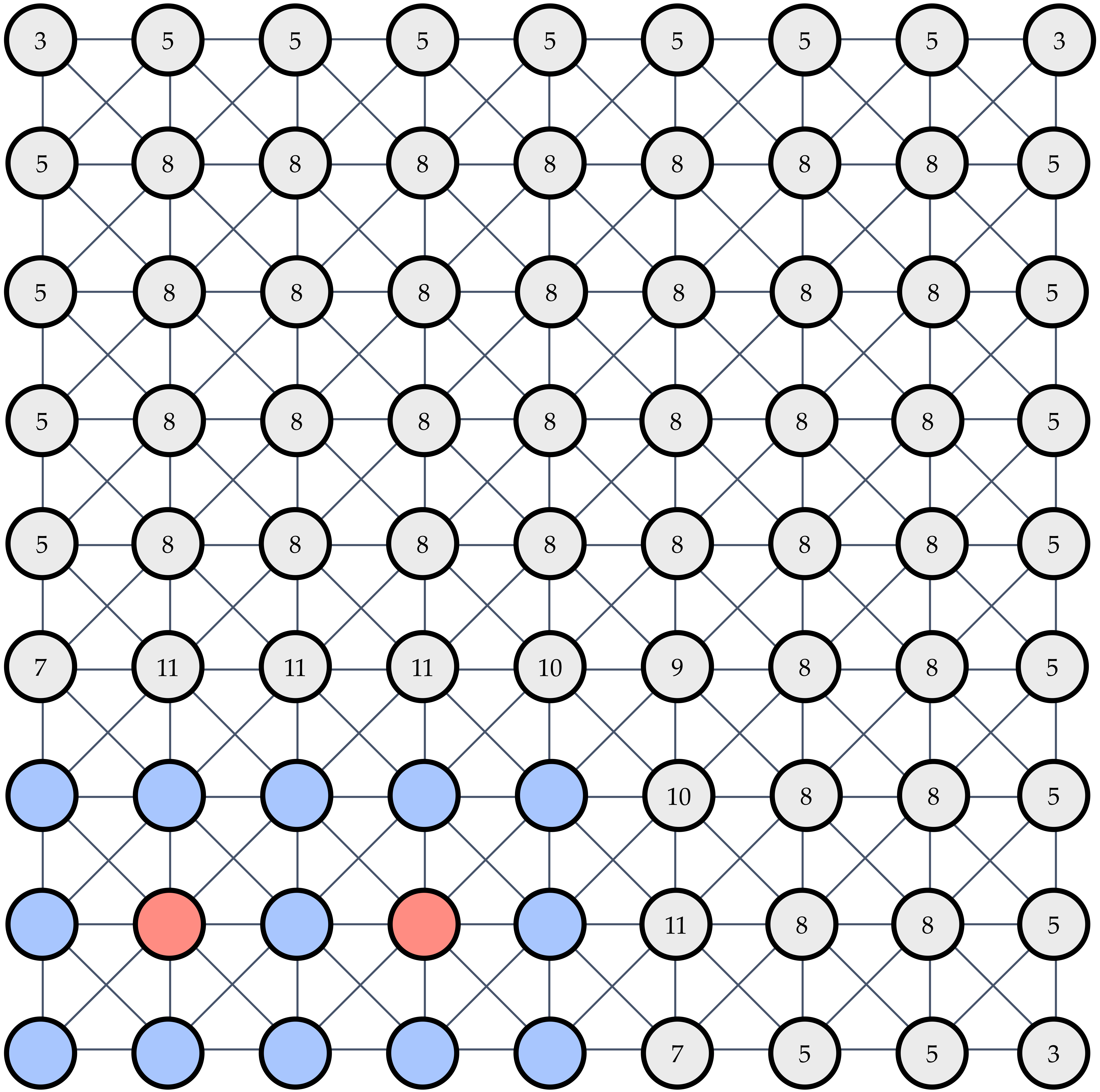
$U = \Omega,\ C = \emptyset,\ F = \emptyset$

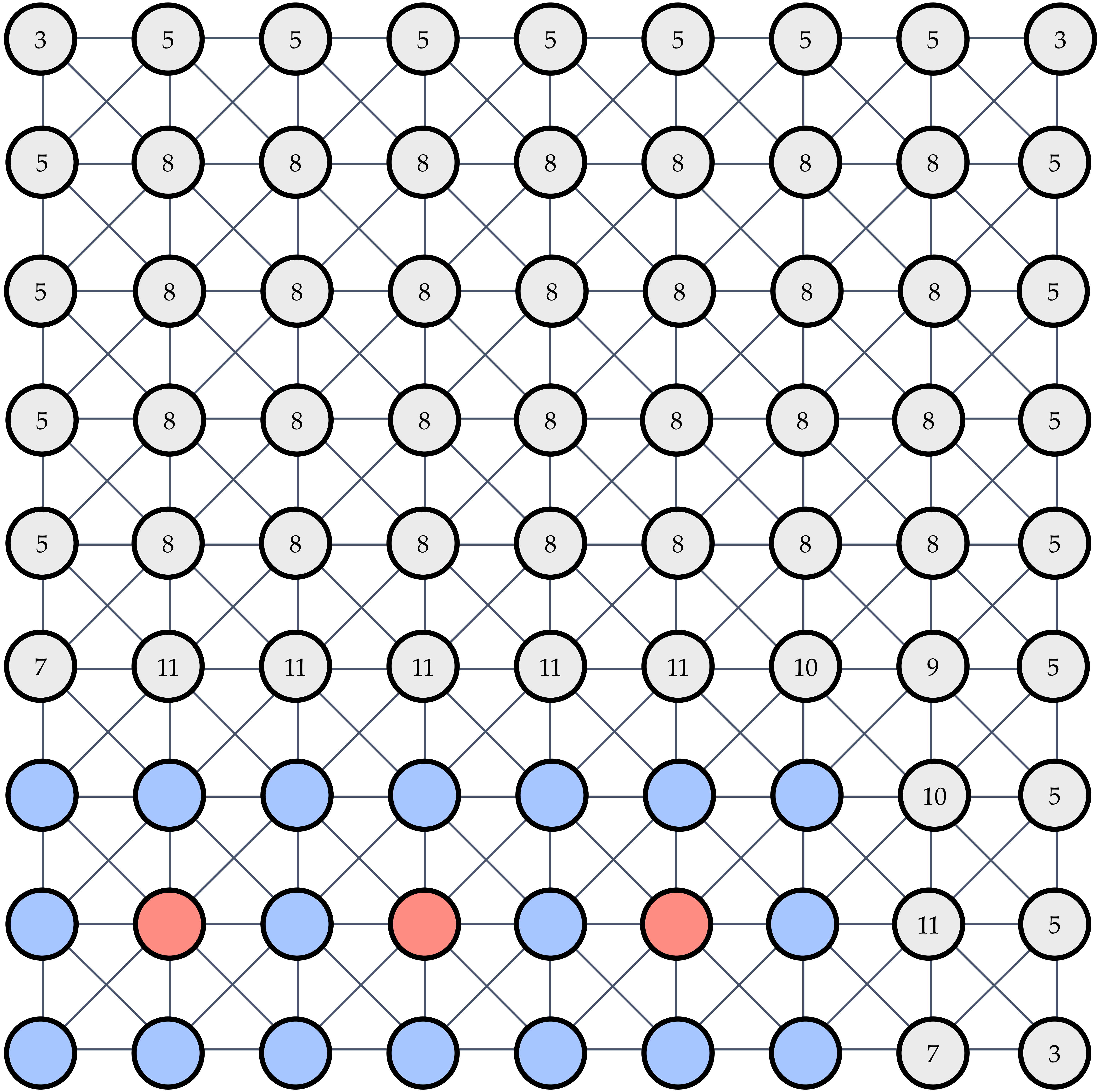
for all $i \in \Omega$ **do**
    $w_i \leftarrow |S_i^{\mathrm{T}}|$
end for
while $|U| > 0$ **do** {First pass}
    select i: $w_i \geqslant w_j,\ \forall j \in U$
    $U \leftarrow U \setminus \{i\}$
    $C \leftarrow C \cup \{i\}$
    for all $j \in S_i^{\mathrm{T}} \cap U$ **do**
        $U \leftarrow U \setminus \{j\}$
        $F = F \cup \{j\}$
        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

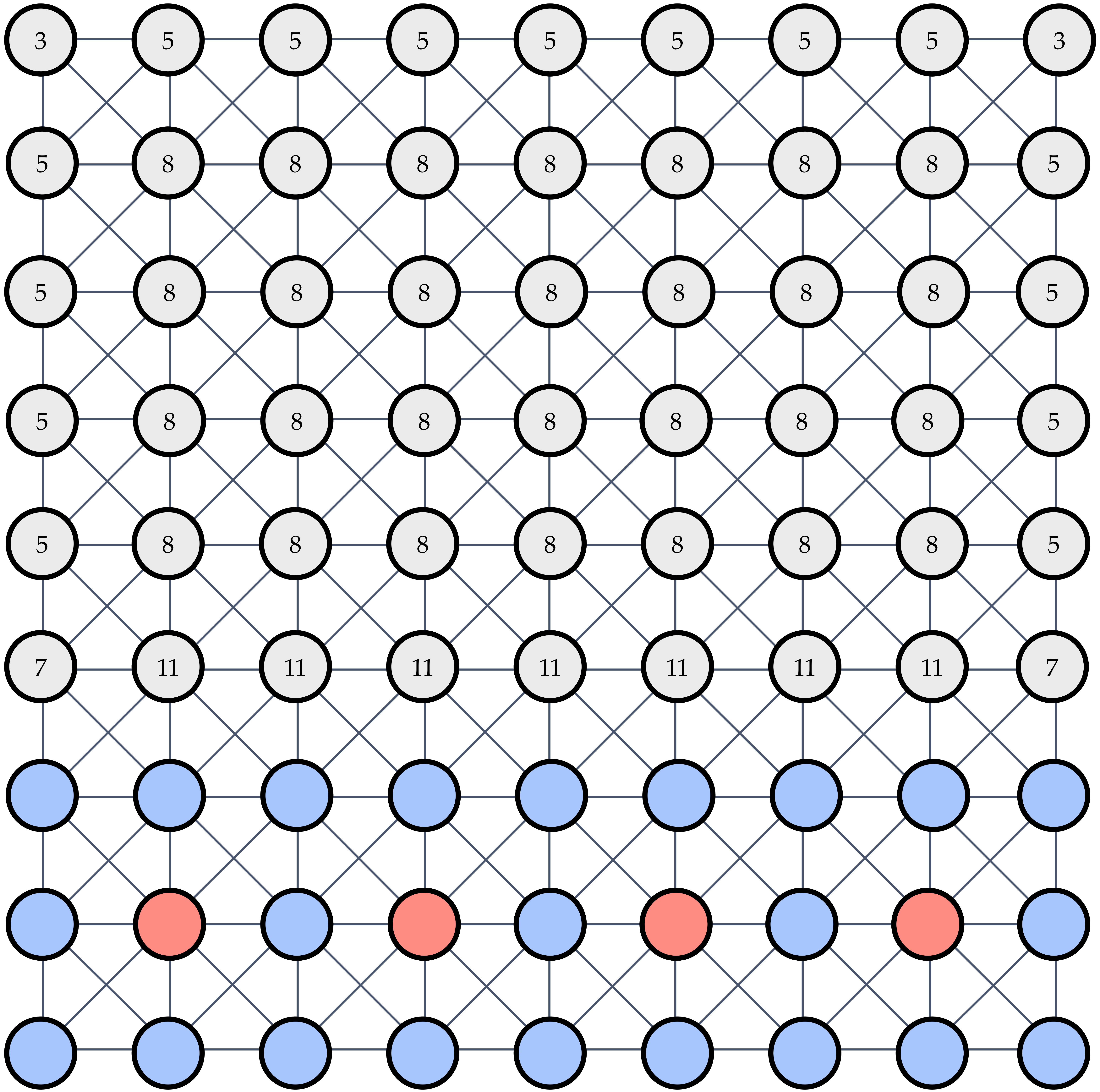

3 5 5 5 5 5 5 5 3
5 8 8 8 8 8 8 8 5
5 8 8 8 8 8 8 8 5
5 8 8 8 8 8 8 8 5
5 8 8 8 8 8 8 8 5
5 8 8 8 8 8 8 8 5
5 8 8 8 8 8 8 8 5
5 8 8 8 8 8 8 8 5
3 5 5 5 5 5 5 5 3

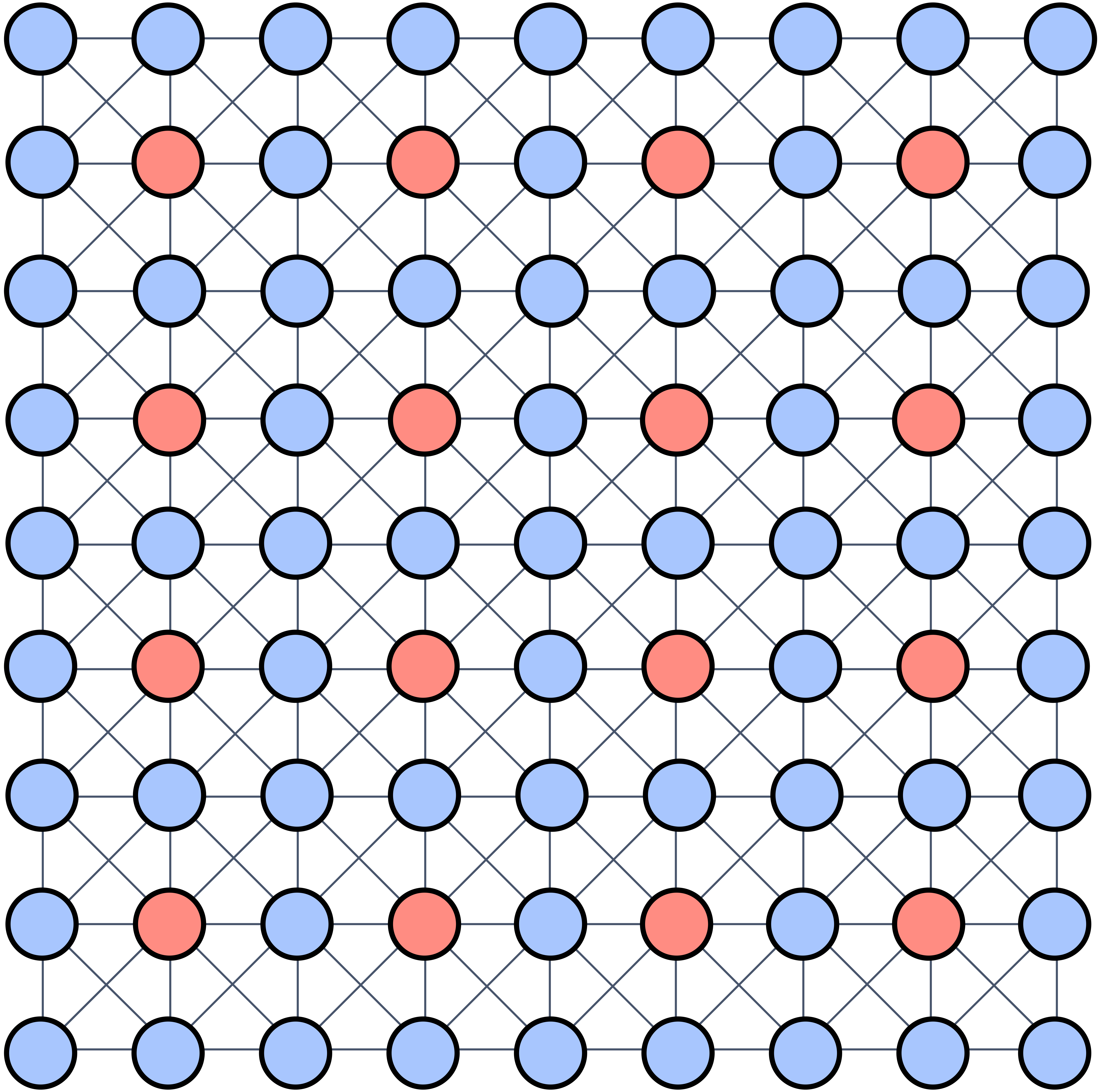


3 5 5 5 5 5 5 5 3
5 8 8 8 8 8 8 8 5
5 8 8 8 8 8 8 8 5
5 8 8 8 8 8 8 8 5
5 8 8 8 8 8 8 8 5
7 11 10 9 8 8 8 8 5
10 8 8 8 8 5
11 8 8 8 8 5
7 5 5 5 5 3


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



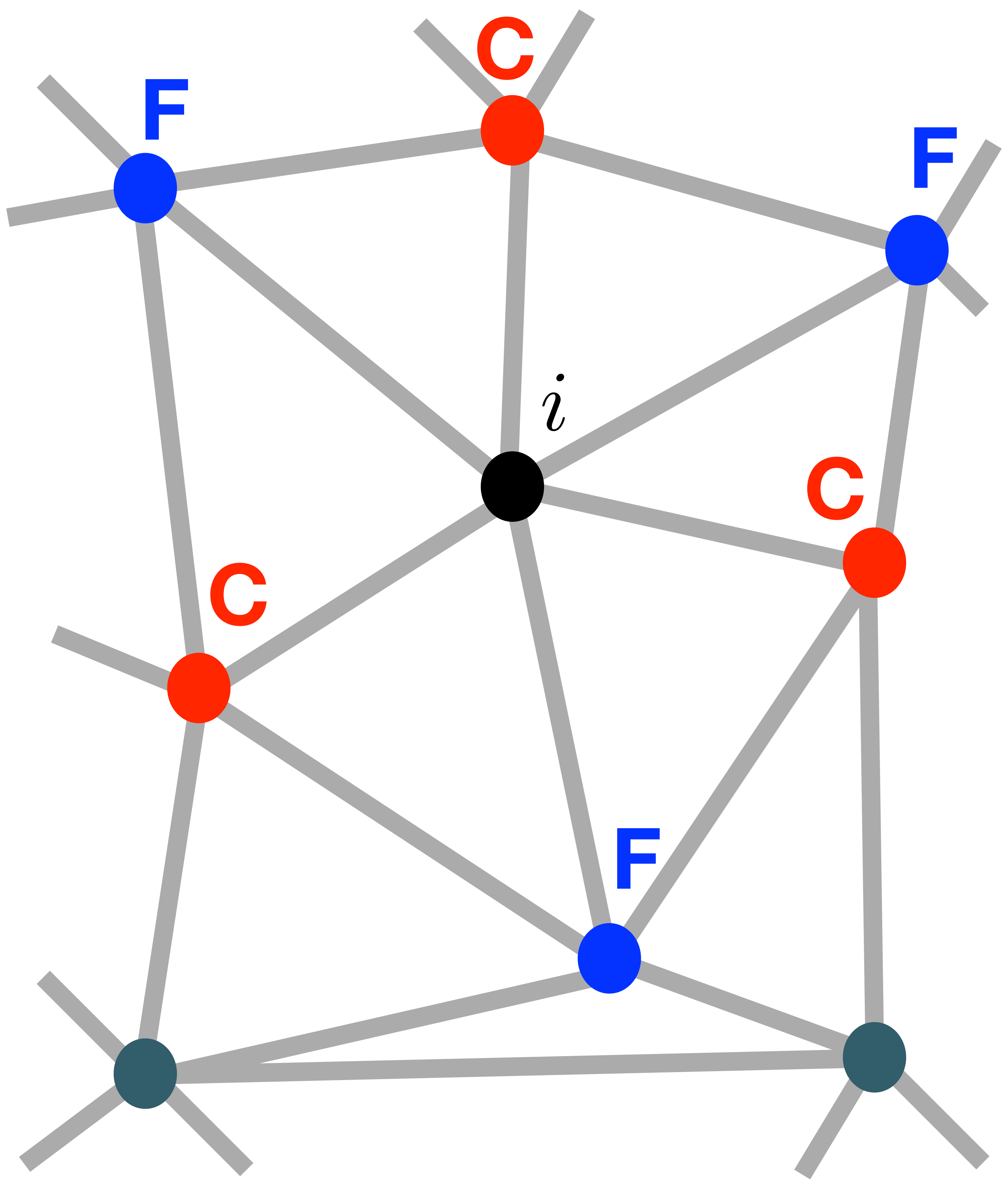
CF AMG

- 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

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

- Or

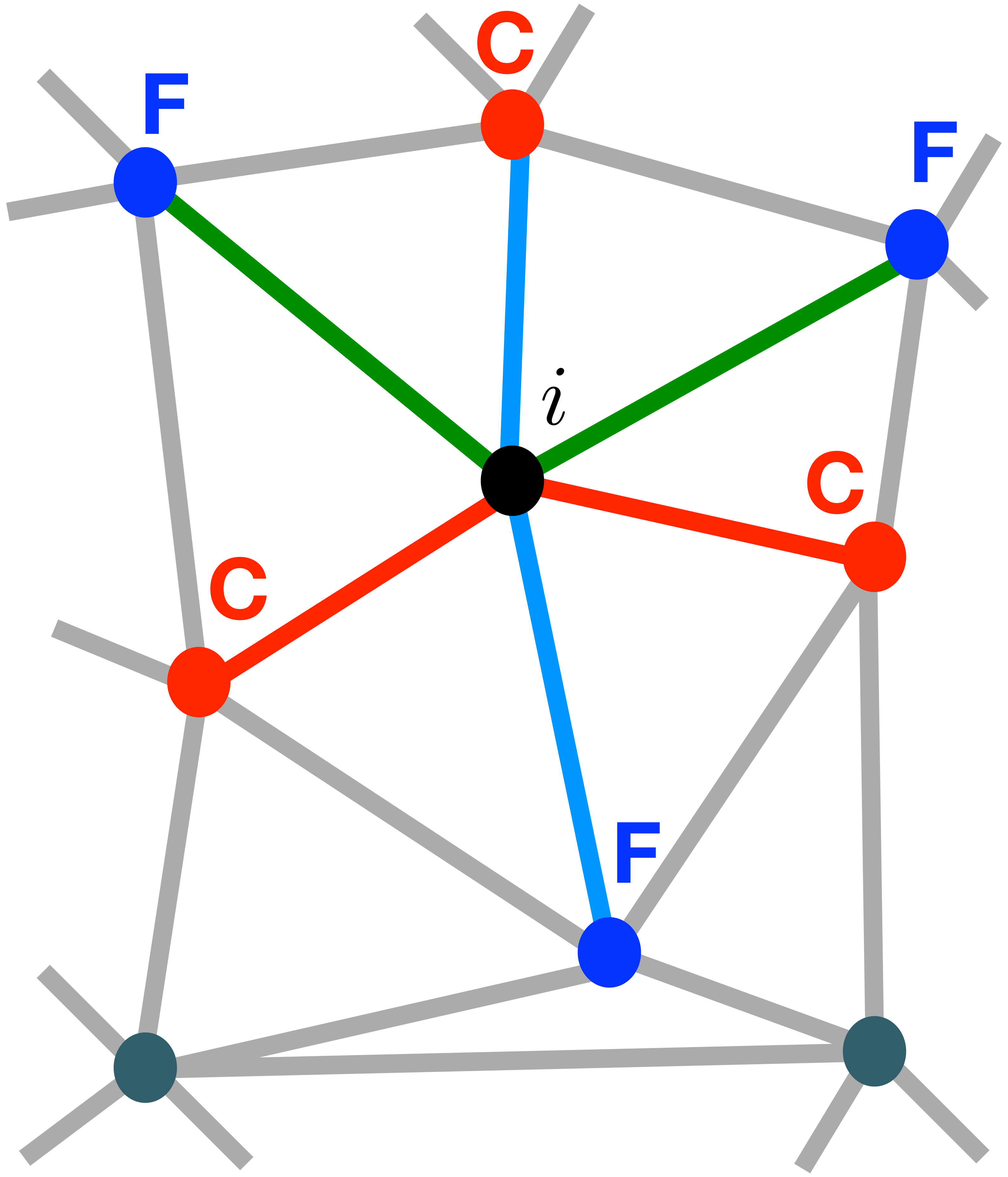
$$P\vec{e} = P \begin{bmatrix} \vec{e}_F \\ \vec{e}_C \end{bmatrix} = \begin{bmatrix} W \\ I \end{bmatrix} \vec{e}_C$$



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

$$Ae \approx 0$$

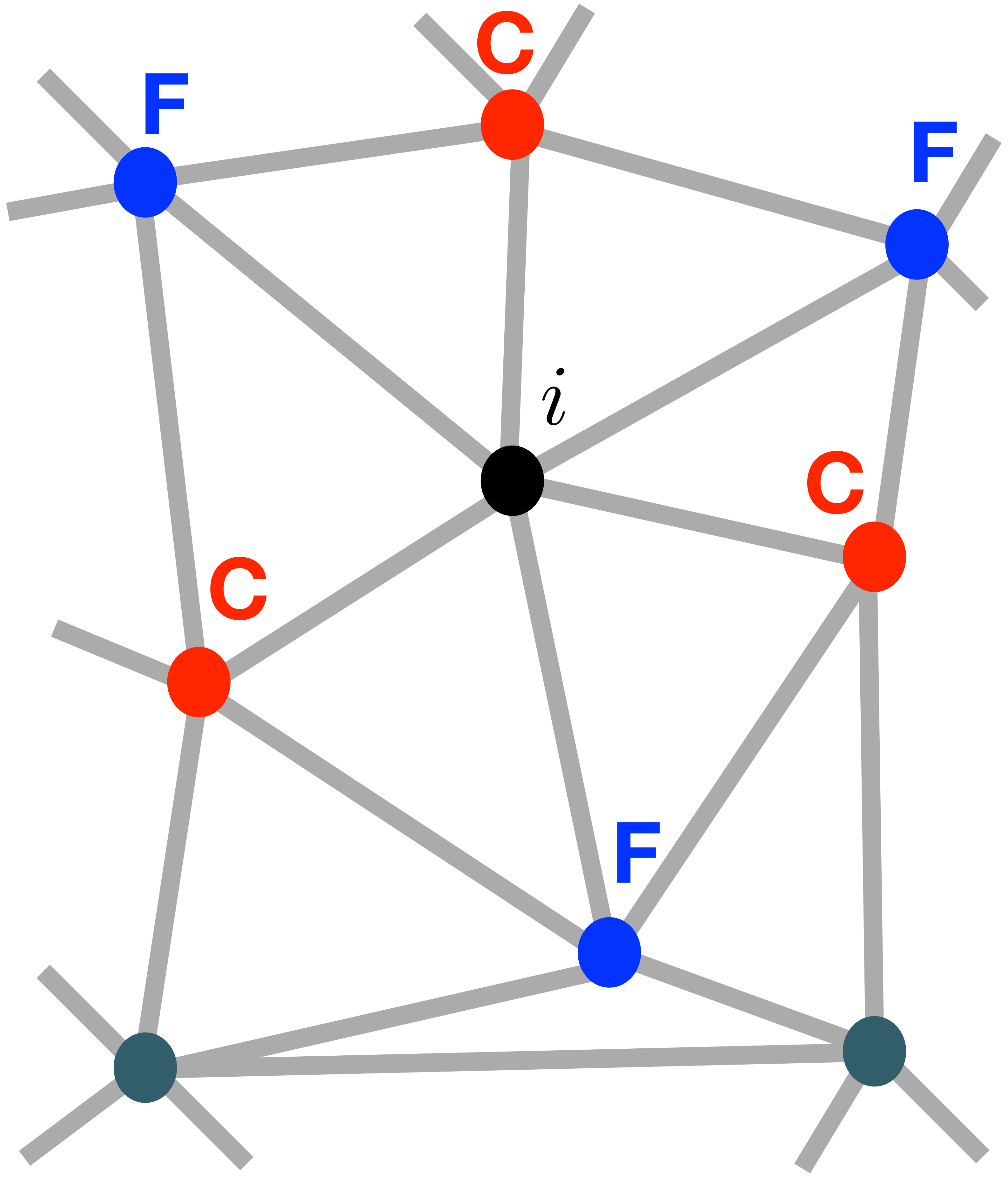
- Then "solve" for i

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

- Then split into types

$$A_{ii}e_i = -\sum_{j \in C_i} A_{ij}e_j - \sum_{k \in D_i^s} A_{ik}e_k - \sum_{m \in D_i^w} A_{im}e_m$$

strong C strong F weak



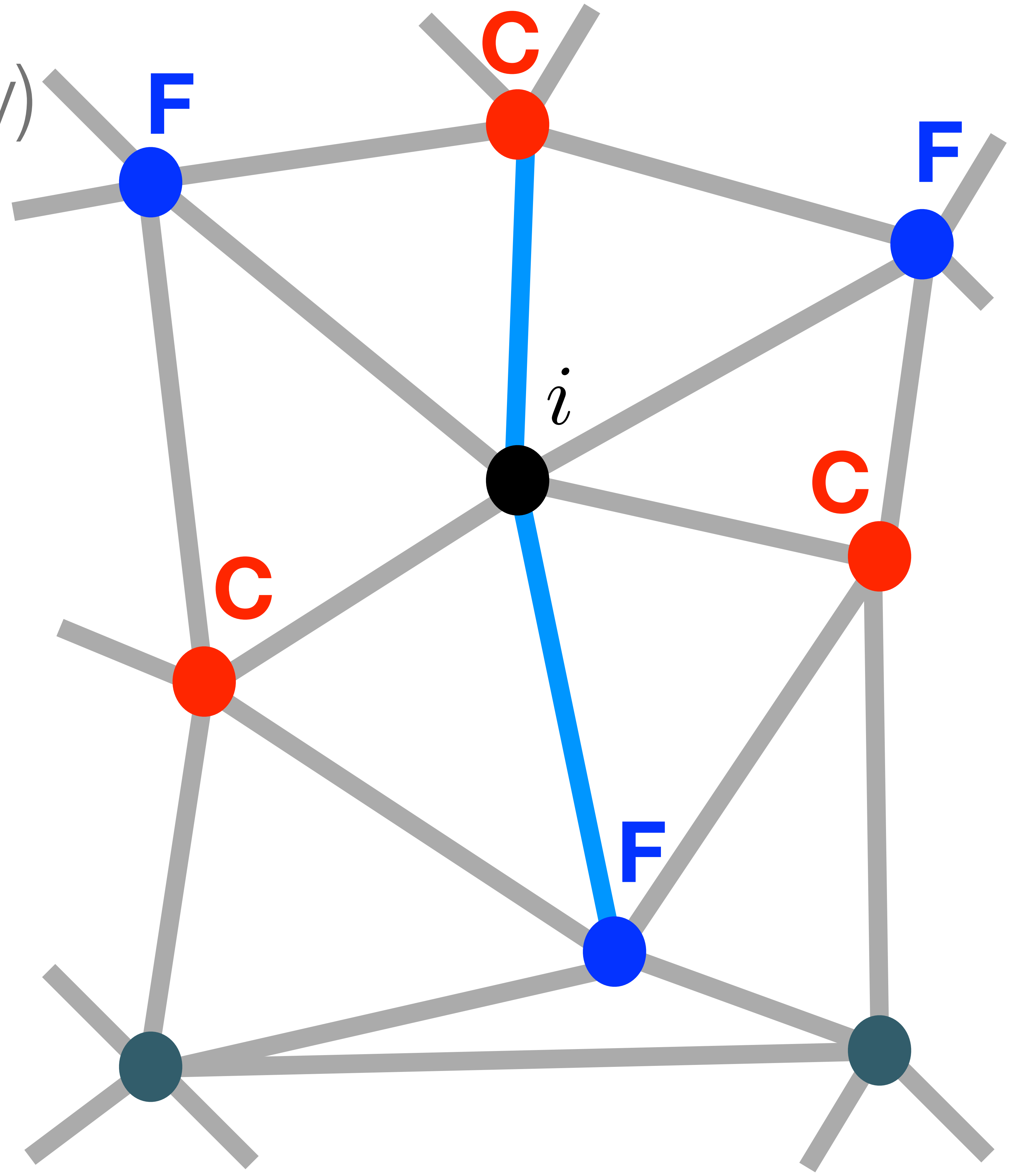
CF AMG

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

$$A_{ii}e_i = -\sum_{j\in C_i} A_{ij}e_j - \sum_{k\in D_i^s} A_{ik}e_k - \sum_{m\in D_i^w} A_{im}e_m$$

$$\Downarrow$$

$$\left(A_{ii} + \sum_{m\in D_i^w} A_{im}\right) e_i = -\sum_{j\in C_i} A_{ij}e_j - \sum_{k\in D_i^s} A_{ik}e_k$$



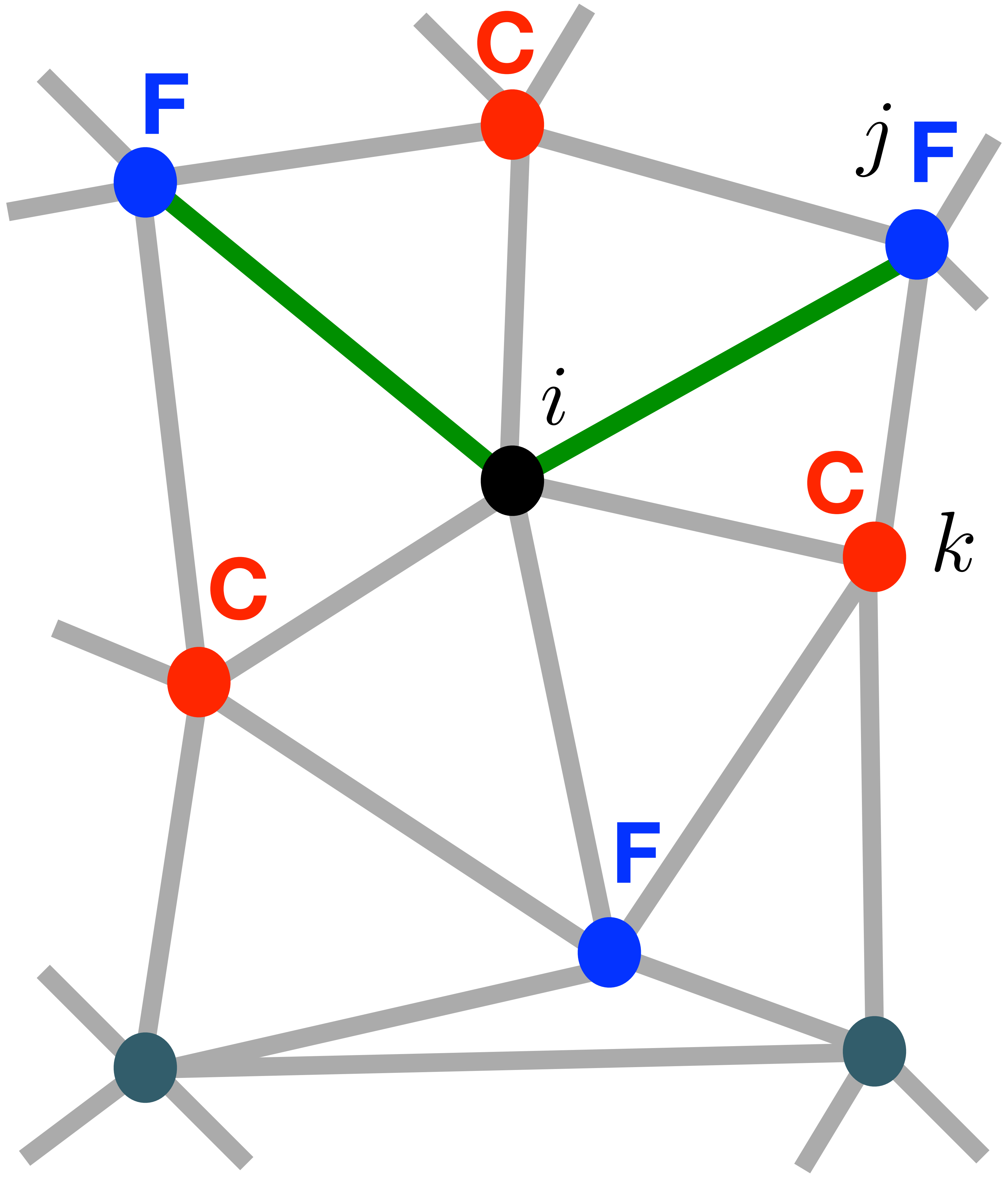
CF AMG

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

$$e_k \approx \frac{\sum_{k \in C_i} A_{kj} e_j}{\sum_{\ell \in C_i} A_{\ell j}}$$

- This gives weights

$$\omega_{ij} = -\frac{A_{ij} + \sum_{k \in D_i^s} \left(\frac{A_{ik} A_{kj}}{\sum_{\ell \in C_i} A_{\ell j}} \right)}{A_{ii} + \sum_{m \in D_i^w} A_{im}}$$



CF AMG Setup Algorithm

Algorithm 2: `CF_setup()`

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

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

```
ℓ = 0
while size(A_ℓ) > max_size
    S_ℓ = strength(A_ℓ)                          {Strength-of-connection}
    C_ℓ, F_ℓ = splitting(S_ℓ)                    {C/F-splitting}
    W = weights(S_ℓ, A_ℓ, C_ℓ, F_ℓ)              {Interpolation weights}
    P_ℓ = [W; I]                                 {Form interpolation}
    A_ℓ+1 = P_ℓ^T A_ℓ P_ℓ                        {Coarse-grid operator}
    ℓ = ℓ + 1
```

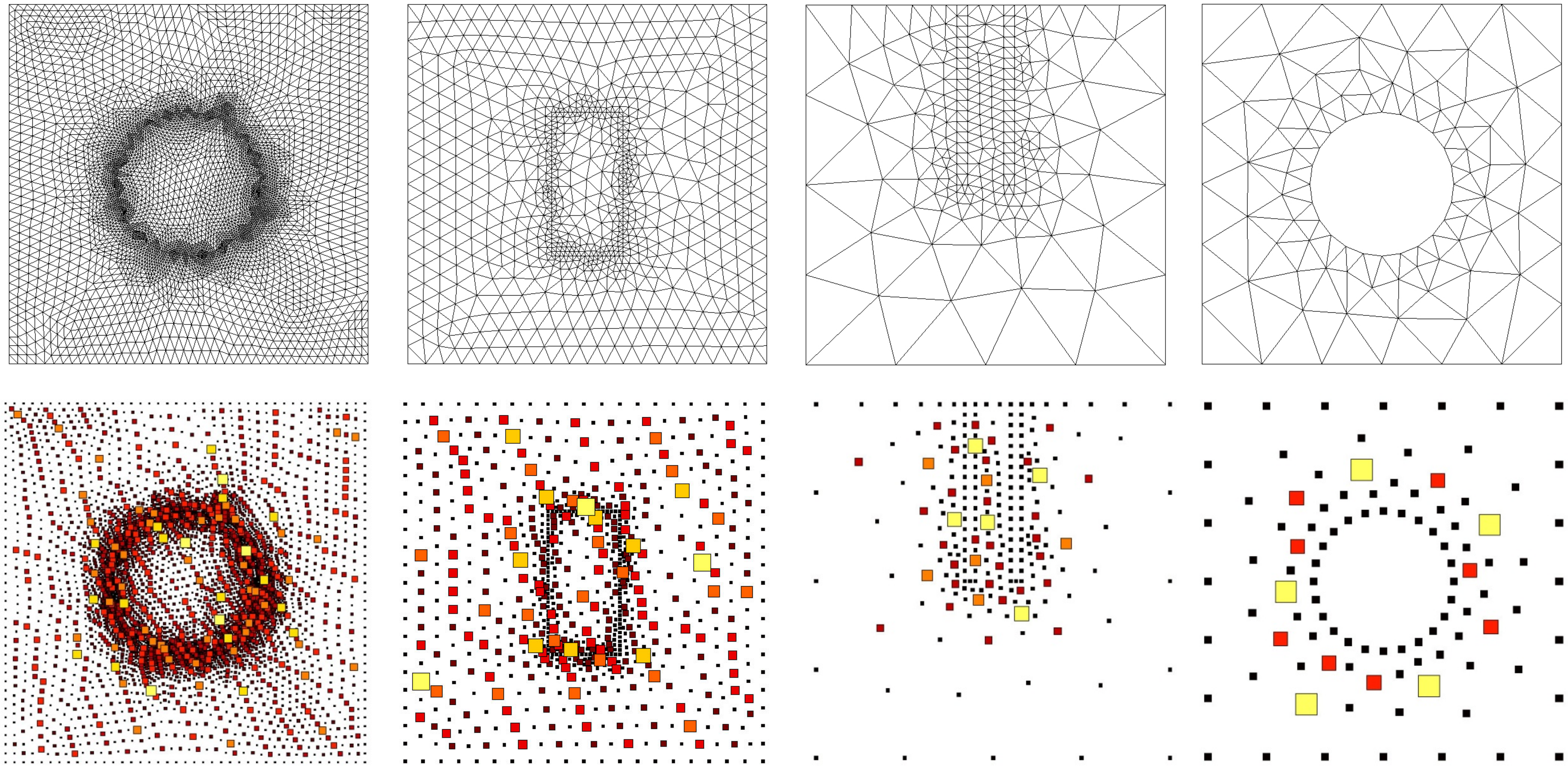
$\ell = 0$

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}
- $\ell = \ell + 1$

AMG grid hierarchies for several 2D problems

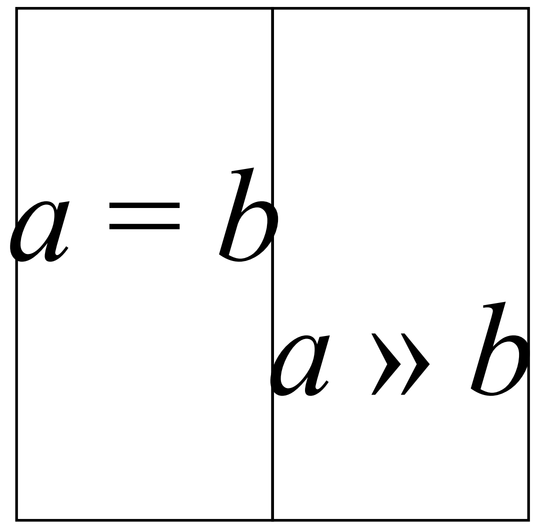
domain1 - 30º domain2 - 30º pile square-hole

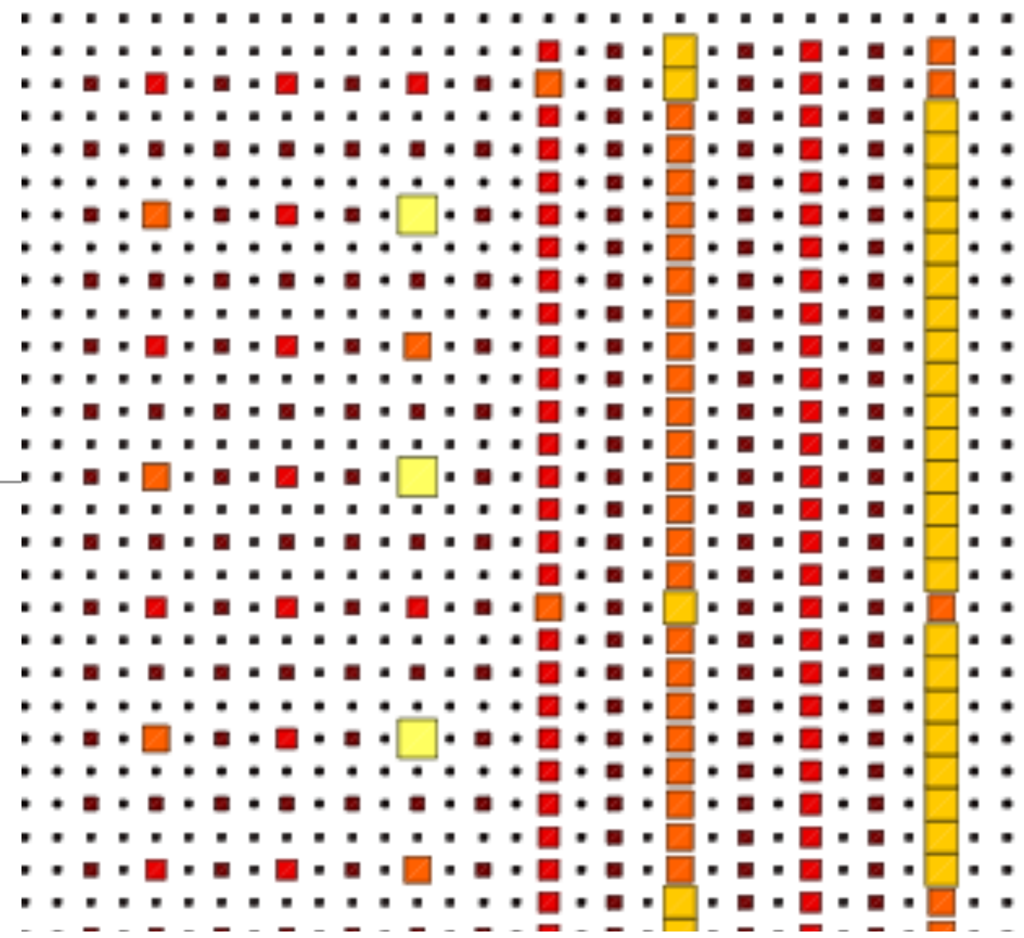


Slide credit: R. Falgout, 2/1/2016

Example CF-AMG results

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





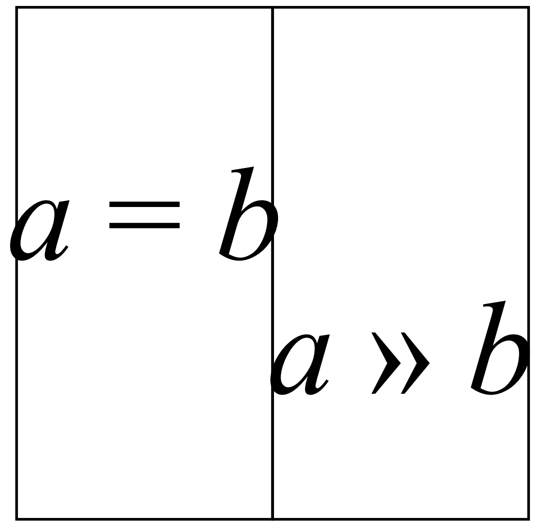
CF-AMG coarse grids

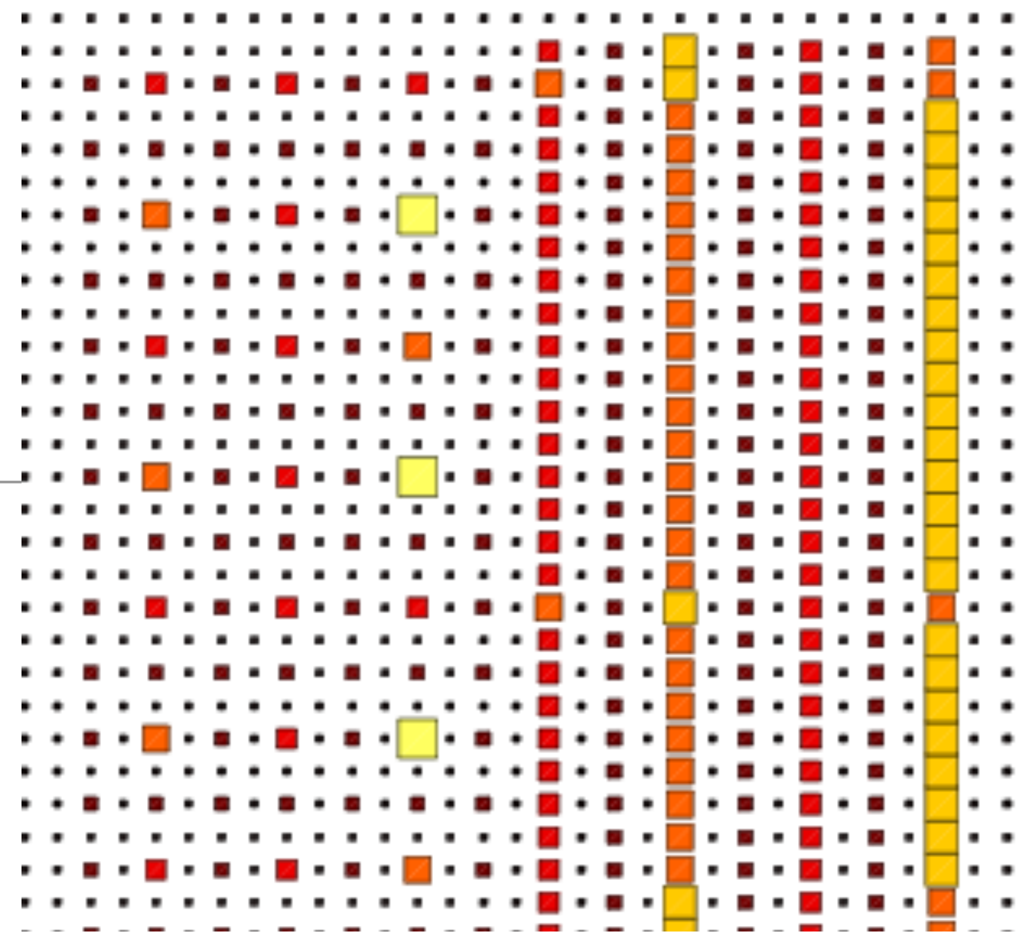
N	Iters	Conv factor	Coarse grids	Grid comp	Oper comp	Setup time	Solve time
61×61	10	0.23	6	1.6	1.6	0.01	0.02
121×121	9	0.23	8	1.6	1.7	0.05	0.07
241×241	9	0.23	9	1.6	1.7	0.25	0.32
481×481	9	0.23	12	1.7	1.7	1.02	1.27
961×961	11	0.29	13	1.7	1.7	4.42	6.28

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

Example CF-AMG results

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





CF-AMG coarse grids

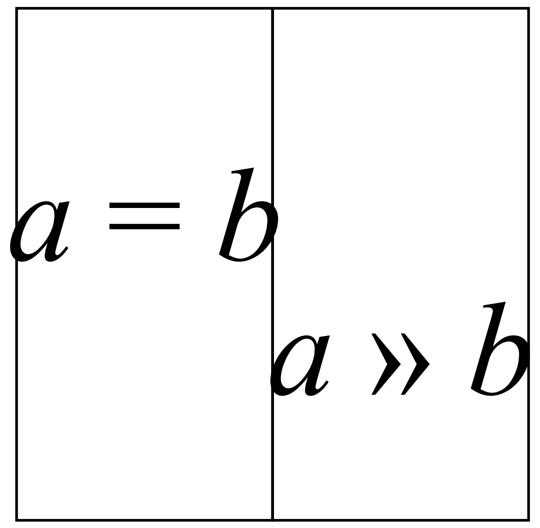
N	Iters	Conv factor	Coarse grids	Grid comp	Oper comp	Setup time	Solve time
61×61	10	0.23	6	1.6	1.6	0.01	0.02
121×121	9	0.23	8	1.6	1.7	0.05	0.07
241×241	9	0.23	9	1.6	1.7	0.25	0.32
481×481	9	0.23	12	1.7	1.7	1.02	1.27
961×961	11	0.29	13	1.7	1.7	4.42	6.28

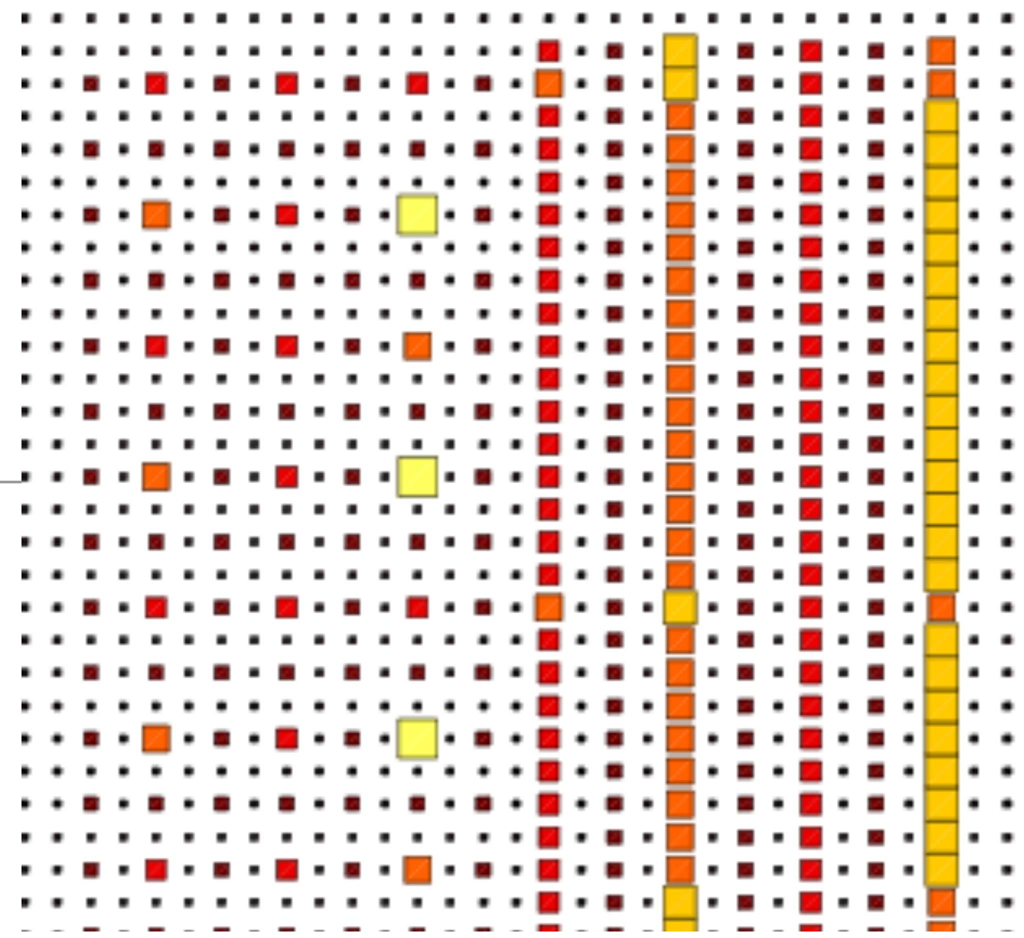
Iterations to a certain tolerance

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

Example CF-AMG results

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





CF-AMG coarse grids

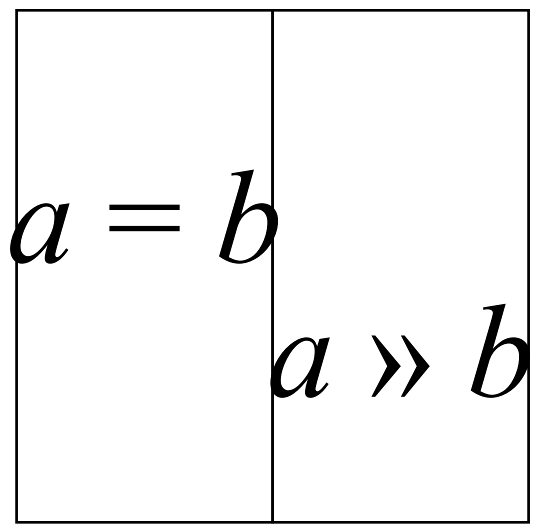
N	Iters	Conv factor	Coarse grids	Grid comp	Oper comp	Setup time	Solve time
61×61	10	0.23	6	1.6	1.6	0.01	0.02
121×121	9	0.23	8	1.6	1.7	0.05	0.07
241×241	9	0.23	9	1.6	1.7	0.25	0.32
481×481	9	0.23	12	1.7	1.7	1.02	1.27
961×961	11	0.29	13	1.7	1.7	4.42	6.28

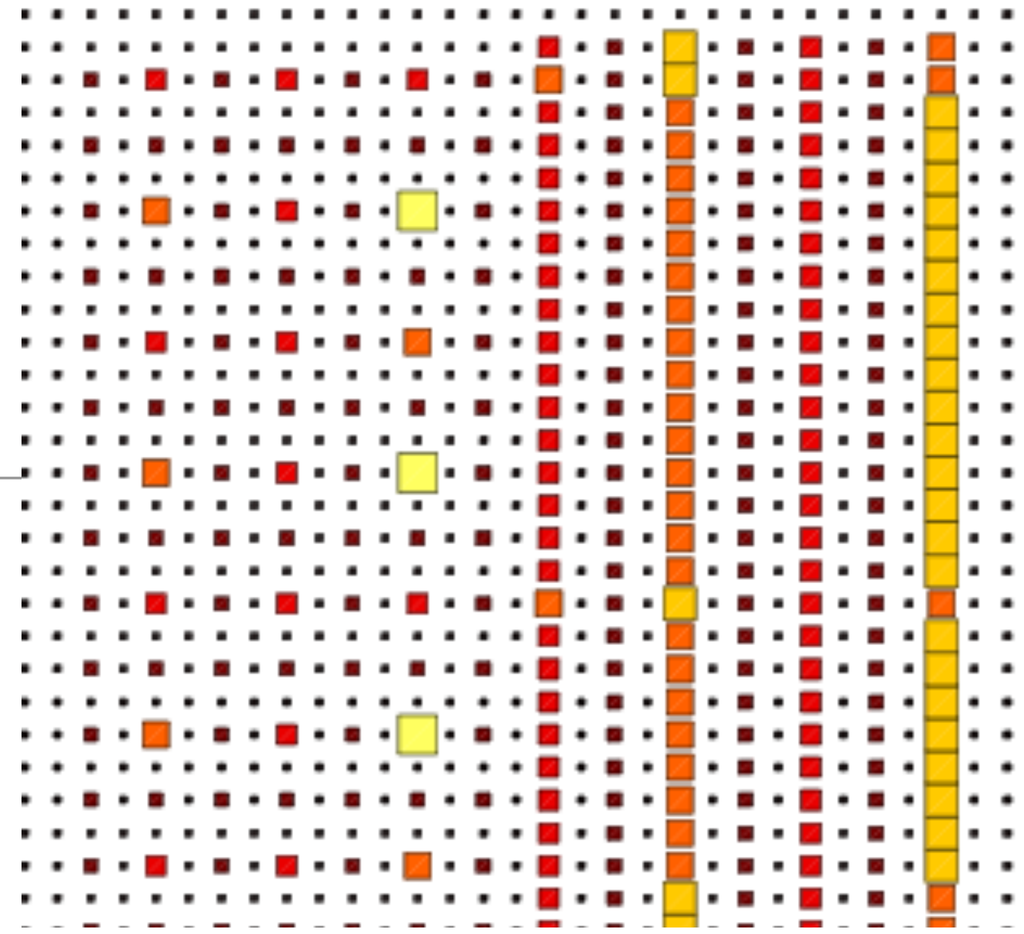
Convergence factor: factor by which the norm of the residual is reduced in each iteration

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

Example CF-AMG results

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





CF-AMG coarse grids

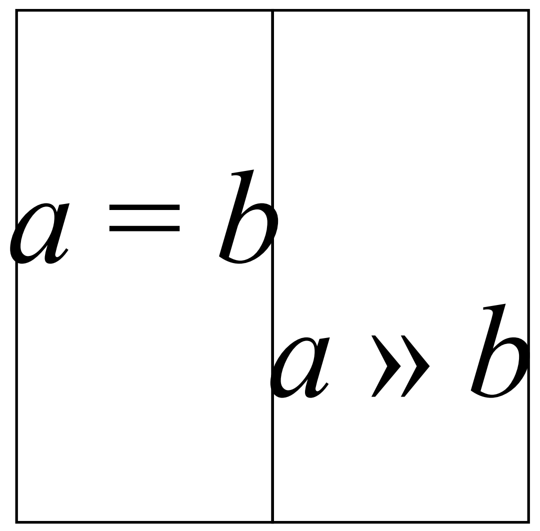
N	Iters	Conv factor	Coarse grids	Grid comp	Oper comp	Setup time	Solve time
61×61	10	0.23	6	1.6	1.6	0.01	0.02
121×121	9	0.23	8	1.6	1.7	0.05	0.07
241×241	9	0.23	9	1.6	1.7	0.25	0.32
481×481	9	0.23	12	1.7	1.7	1.02	1.27
961×961	11	0.29	13	1.7	1.7	4.42	6.28

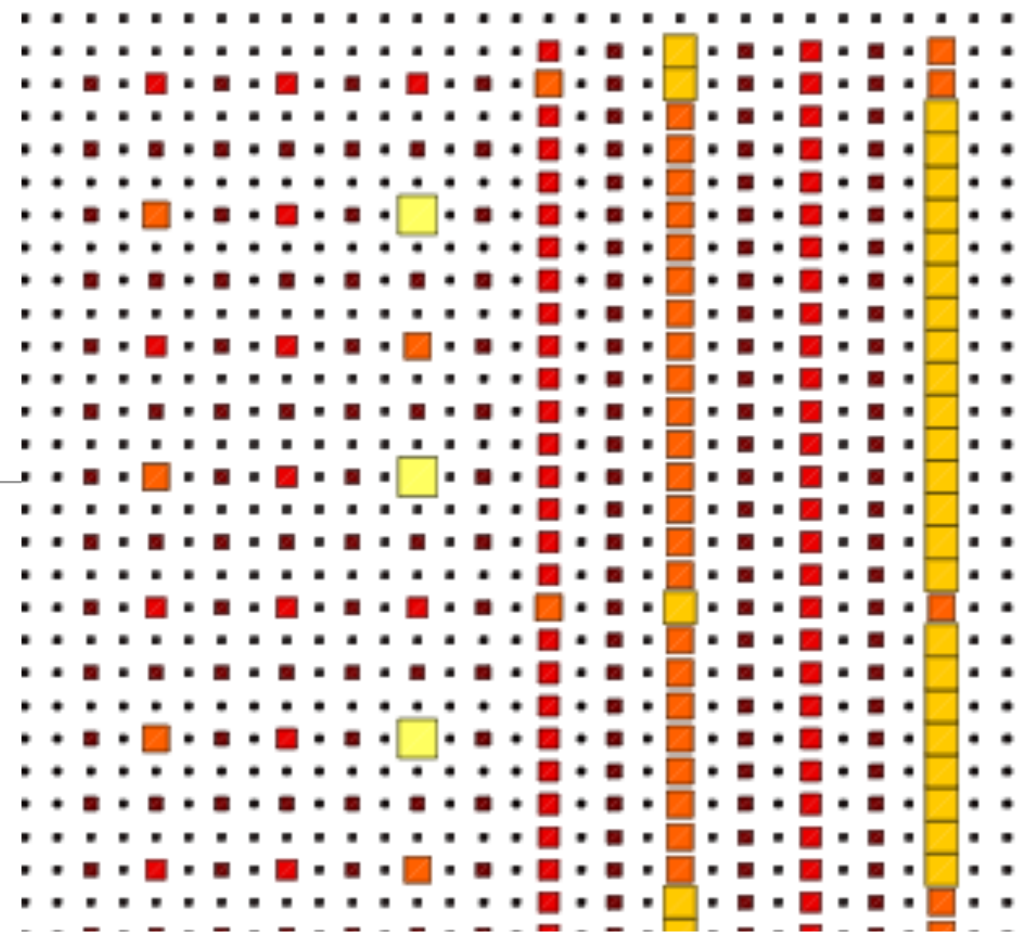
Coarse grids: number of coarse "grids"

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

Example CF-AMG results

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





CF-AMG coarse grids

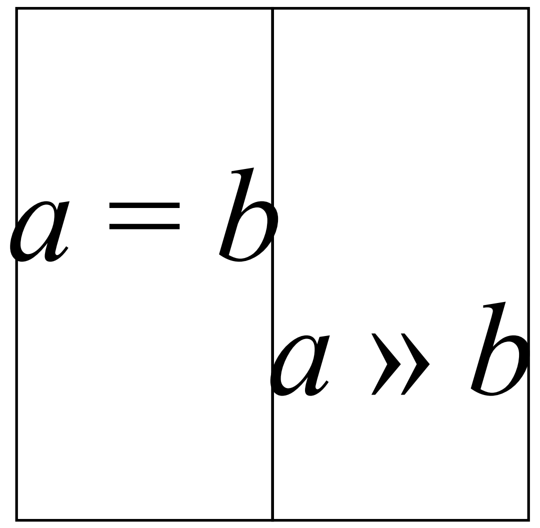
N	Iters	Conv factor	Coarse grids	Grid comp	Oper comp	Setup time	Solve time
61×61	10	0.23	6	1.6	1.6	0.01	0.02
121×121	9	0.23	8	1.6	1.7	0.05	0.07
241×241	9	0.23	9	1.6	1.7	0.25	0.32
481×481	9	0.23	12	1.7	1.7	1.02	1.27
961×961	11	0.29	13	1.7	1.7	4.42	6.28

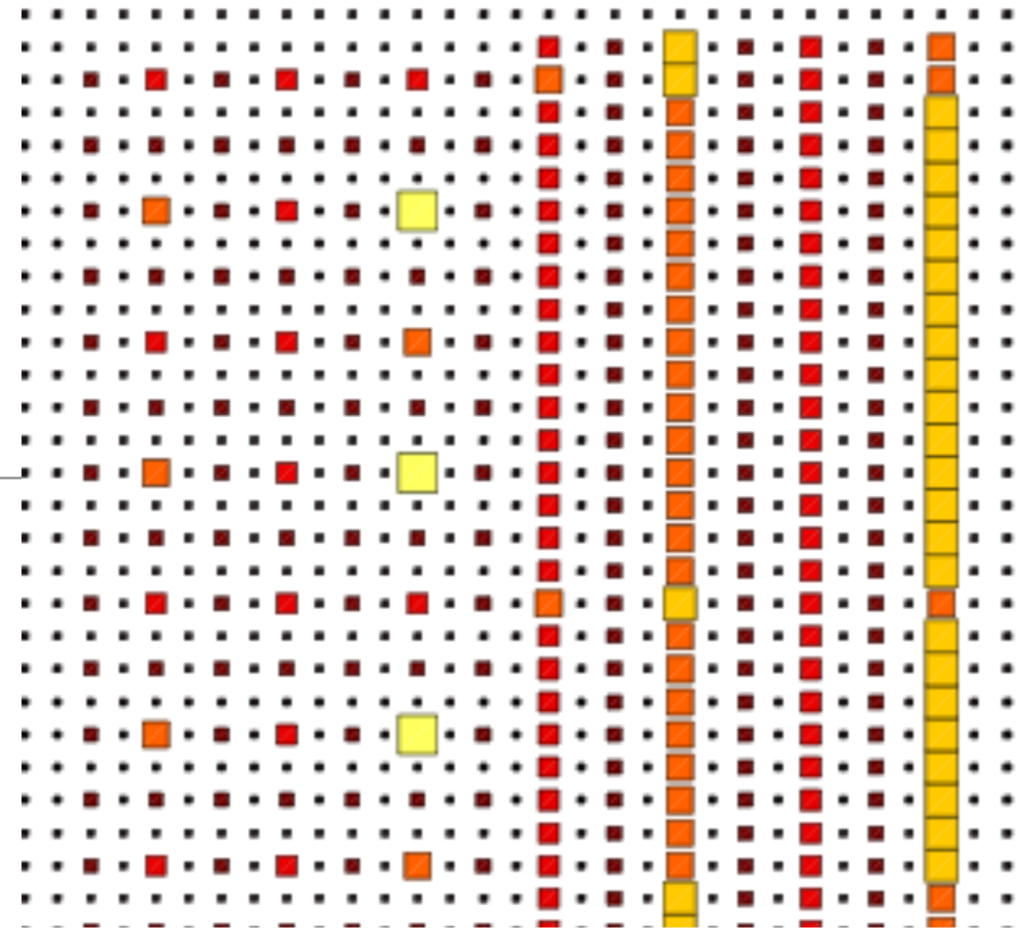
Grid complexity: total number of grid points (system size) on all levels / number of grids points on the fine level

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

Example CF-AMG results

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





CF-AMG coarse grids

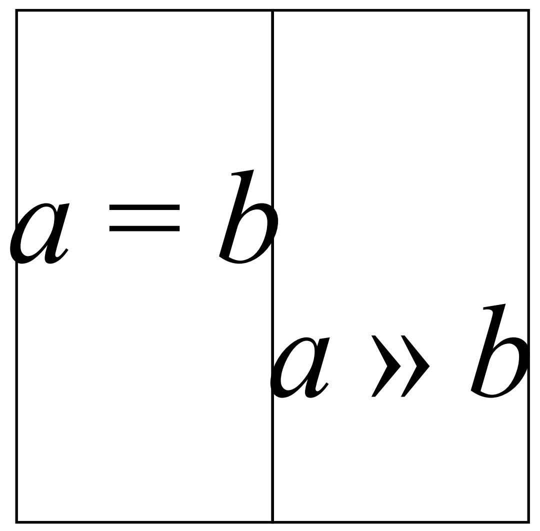
N	Iters	Conv factor	Coarse grids	Grid comp	Oper comp	Setup time	Solve time
61×61	10	0.23	6	1.6	1.6	0.01	0.02
121×121	9	0.23	8	1.6	1.7	0.05	0.07
241×241	9	0.23	9	1.6	1.7	0.25	0.32
481×481	9	0.23	12	1.7	1.7	1.02	1.27
961×961	11	0.29	13	1.7	1.7	4.42	6.28

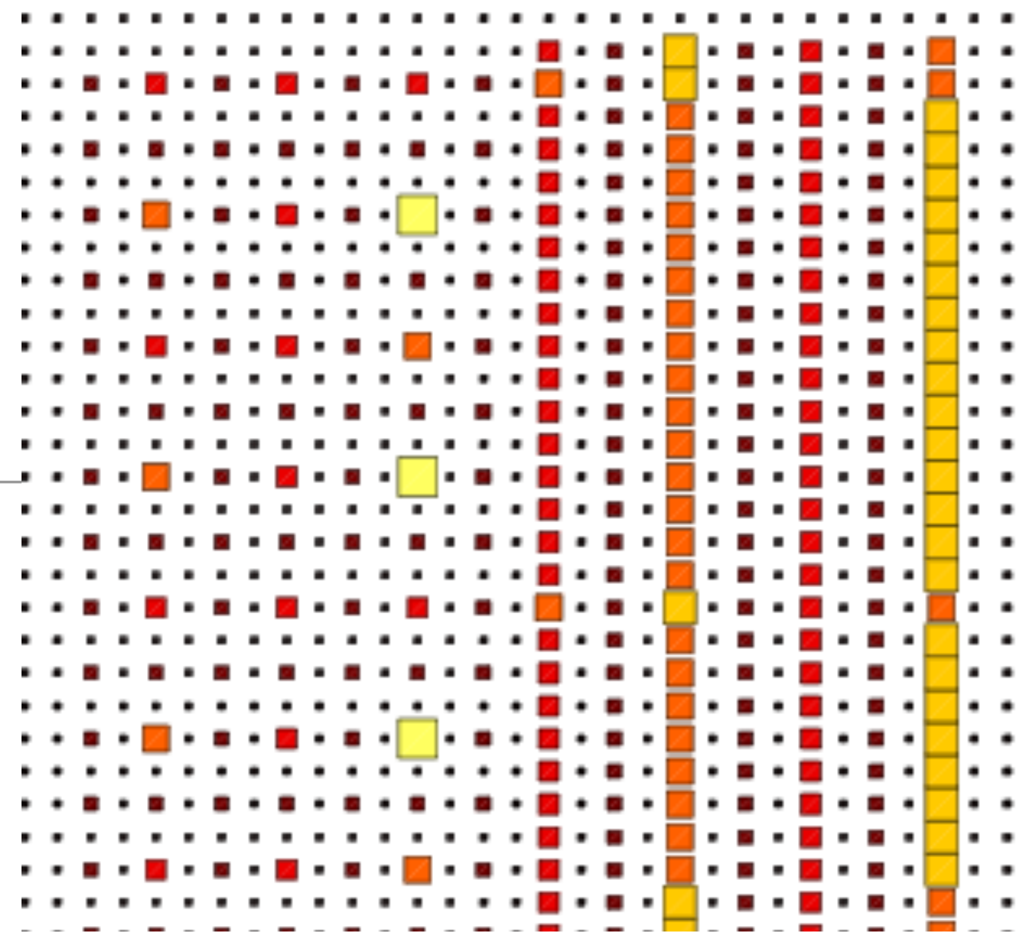
Operator complexity: total number non-zeros on all levels / number of non-zeros on the fine level

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

Demo

- `14-anisotropy-cf-amg.ipynb`

Some theory

- An outline of some (general) theory. *On Generalizing the Algebraic Multigrid Framework,* Falgout, Vassilevski, 2004, SINUM.

- Let *A* be SPD

$$\tilde{M} = M^T \left(M^T + M - A\right)^{-1} M$$

- Consider relaxation of the form

$$u \leftarrow u + M^{-1} r$$

or

$$e \leftarrow e - M^{-1} A e$$

or

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

Some theory

- Let interpolation be P

- Consider some “restriction” operation (not the multigrid operator) such that

$$RP = I$$

- *Note:* PR is a projection onto P

- Example: $R = \begin{bmatrix} 0 & I \end{bmatrix}$

Some theory

- The two-level theory shows

$$\|(I - M^{-1}A)(I - P(P^TAP)^{-1}P^TA)\|_A^2 \leq 1 - \frac{1}{K}$$

- where

$$K = \sup_e \frac{\|(I - PR)e\|_{\tilde{M}}^2}{\|e\|_A^2}$$

an approximation property

- General interpretation: higher quality interpolation (measured with M) leads to improved convergence.

Some theory

- Try to optimize this:

$$K = \sup_e \frac{\|(I - PR)e\|_{\tilde{M}}^2}{\|e\|_A^2}$$

- If $R = \begin{bmatrix} 0 & I \end{bmatrix}$

$$P^T = \begin{bmatrix} W^T & I \end{bmatrix}$$

- Then $P_{ideal} = \text{argmin}_P \sup_e \frac{\|(I - PR)e\|_{\tilde{M}}^2}{\|e\|_A^2}$

$$= \begin{bmatrix} -A_{FF}^{-1} A_{FC} \\ I \end{bmatrix}$$

M

- Try to optimize this:

$$K = \sup_e \frac{\|(I - PR)e\|_{\tilde{M}}^2}{\|e\|_A^2}$$

Make this small
"compatible relaxation" methods

- If $R = \begin{bmatrix} 0 & I \end{bmatrix}$

$$P^T = \begin{bmatrix} W^T & I \end{bmatrix}$$

- Then $P_{ideal} = \mathrm{argmin}_P \sup_e \frac{\|(I - PR)e\|_{\tilde{M}}^2}{\|e\|_A^2}$

$$= \begin{bmatrix} -A_{FF}^{-1} A_{FC} \\ I \end{bmatrix}$$

Impractical, but often a good guide
AMGe-type methods

Demo

- `15-CF-AMG-Interpolation.ipynb`

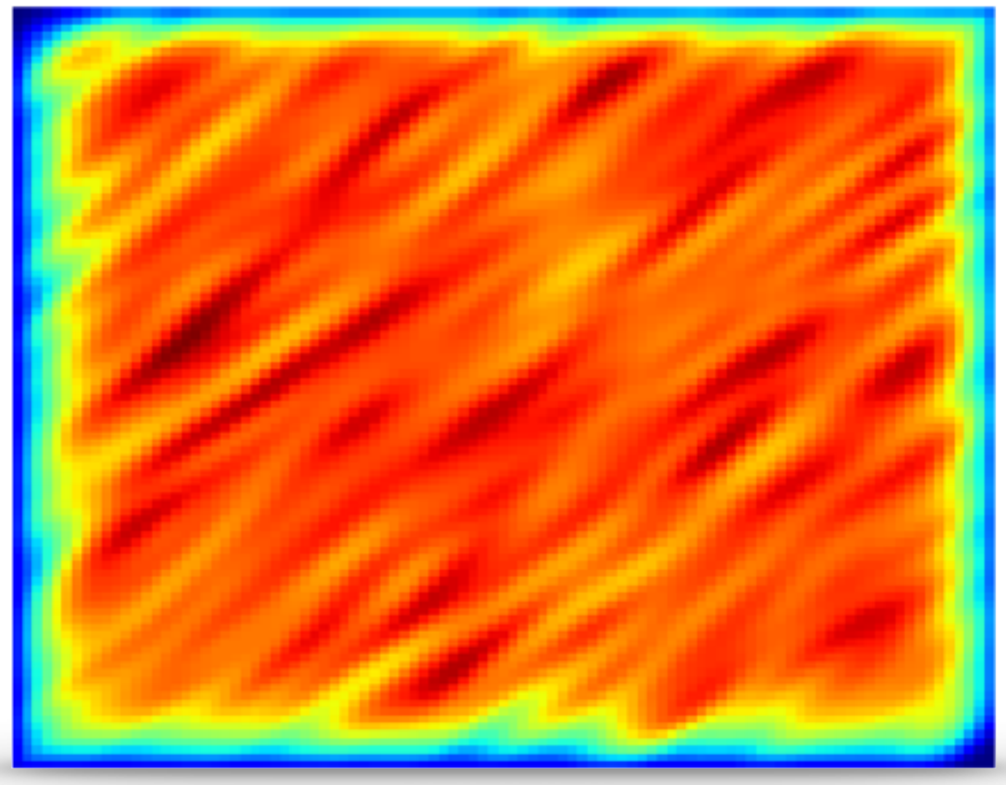
SA AMG

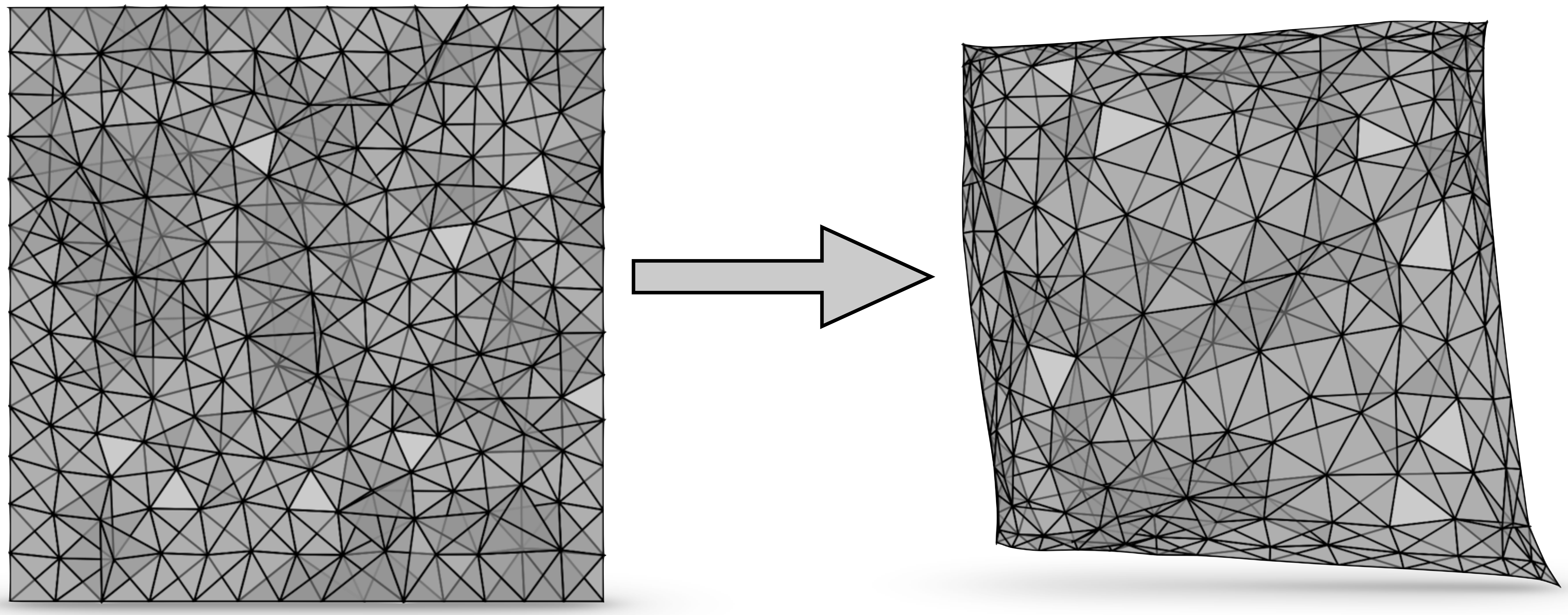
- Smoothed Aggregation based AMG takes a *different* approach
- Still, the same steps:
  1. strength between points
  2. find a coarse grid (this time **aggregates** of points)
  3. define interpolation
  4. compute the coarse-grid operator

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

SA AMG

- **SA AMG relies on "candidate" vectors or the near null space or smooth error**
- vector of ones
- pre-smooth guesses
- adapting a cycle
- *a priori* knowledge
- topological inference





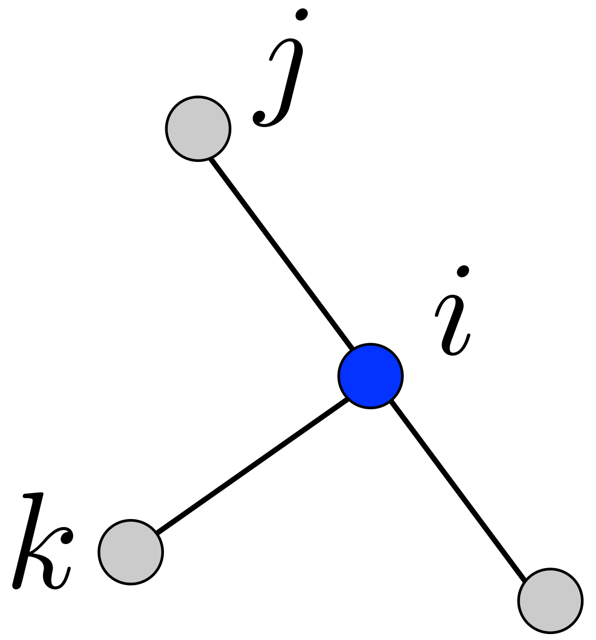
Symmetric Strength

- i strongly depends on j if

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

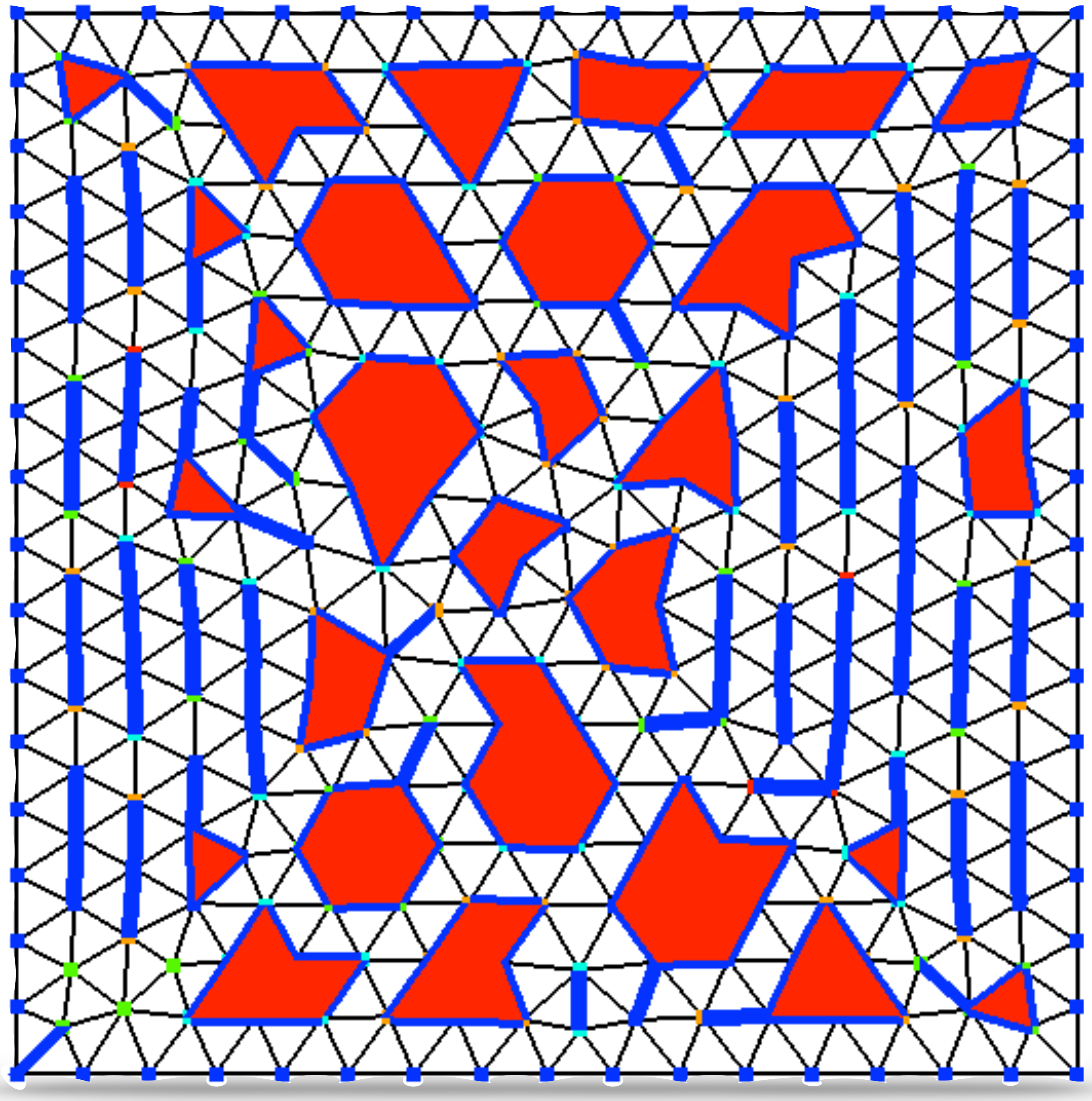
- i strongly depends on j if

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



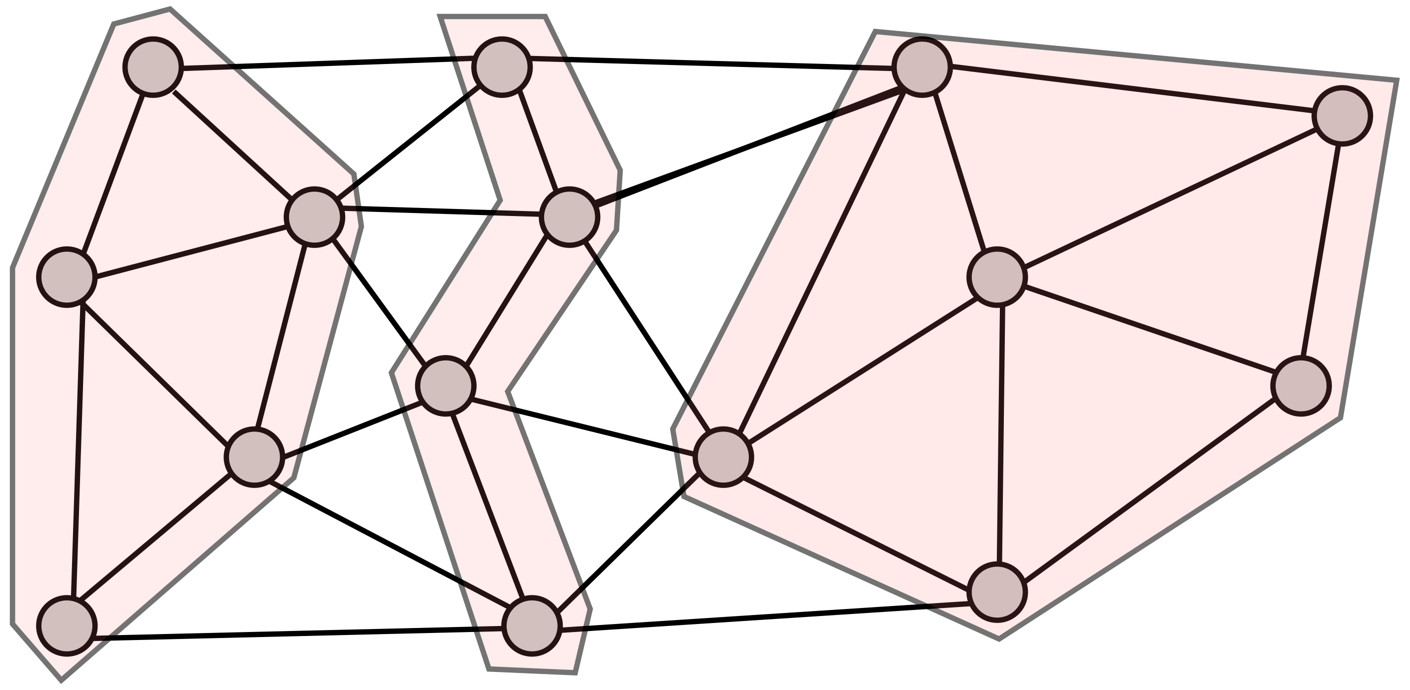
both “think” elliptic

anisotropy



Algebraic Framework

- aggregation: groups of <u>fine</u> nodes form <u>coarse</u> nodes

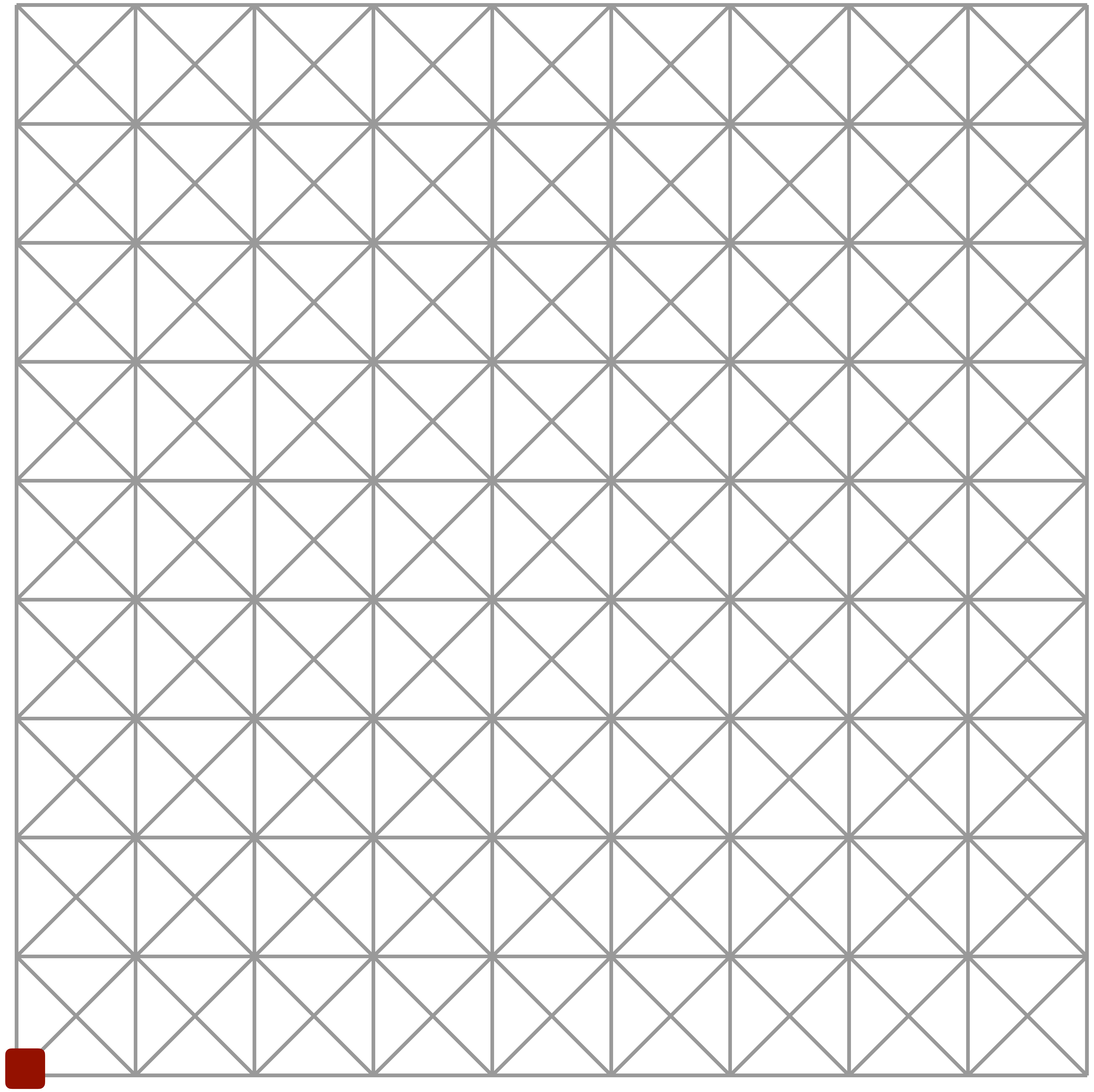


fine: 15

coarse: 3

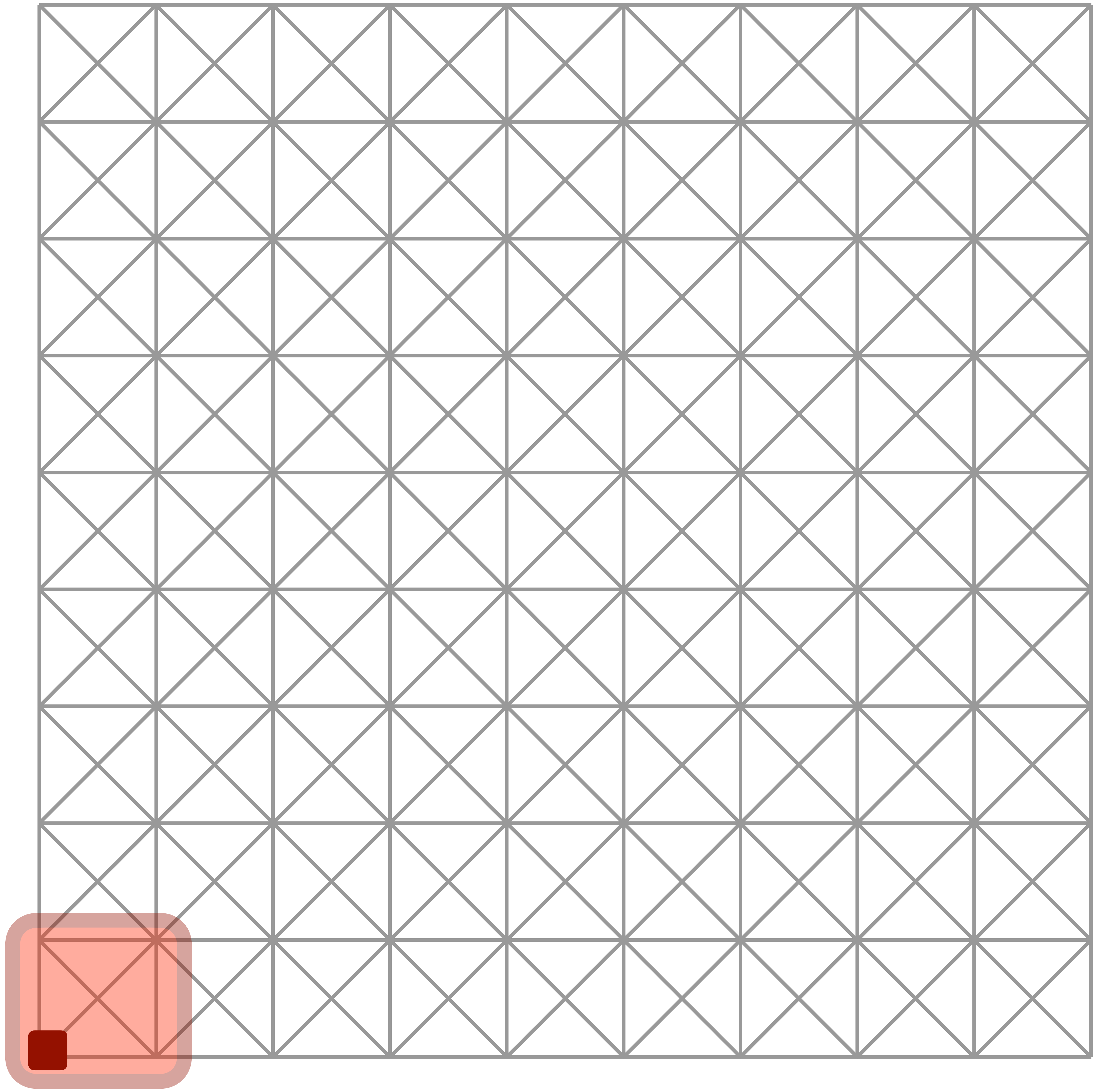
- an initial interpolation pattern
- find an optimal interpolation operator P that contains low energy

Aggregation



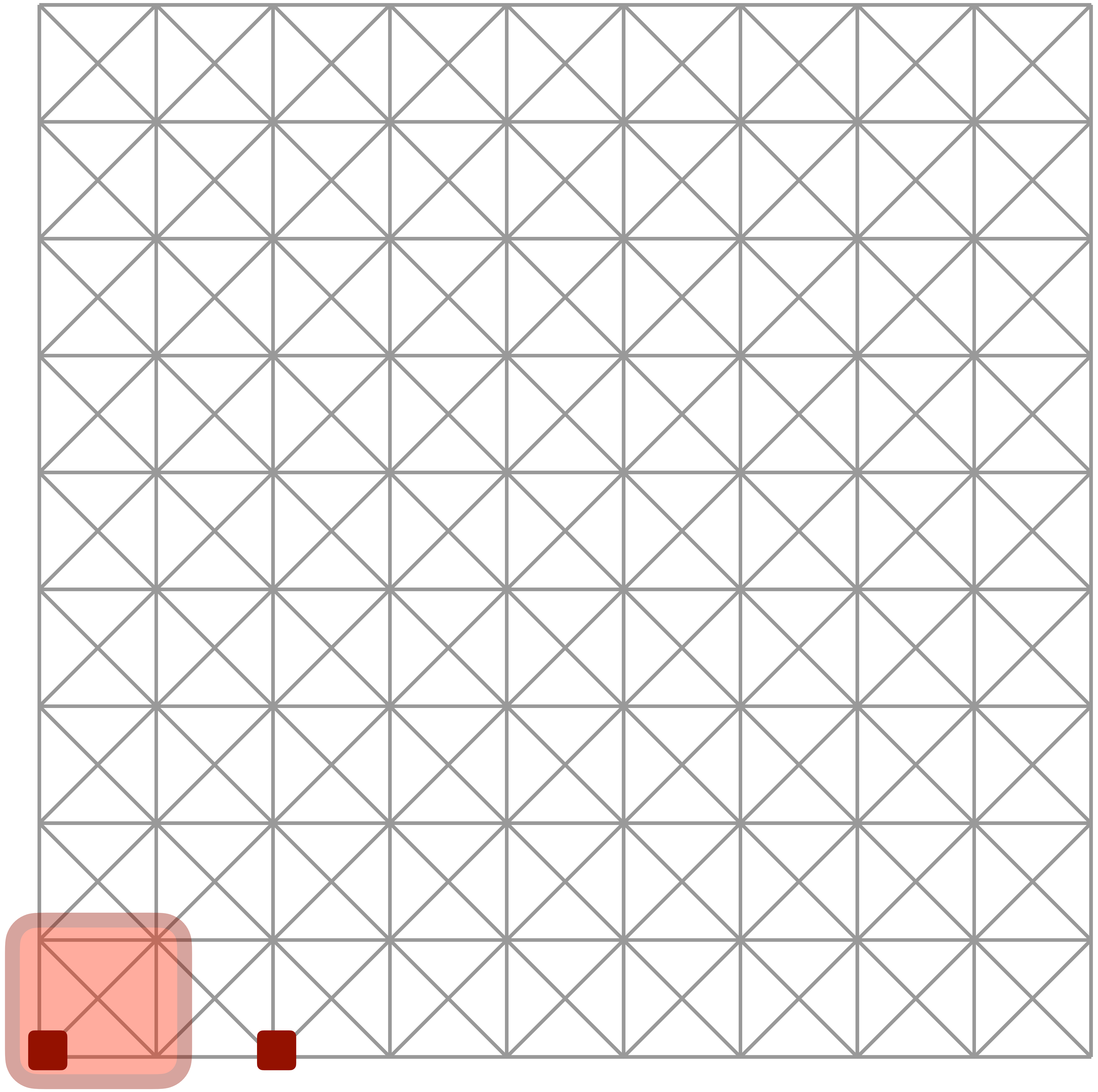
- Select next node
- If all strongly connected neighbors are unaggregated, then aggregate.

Aggregation



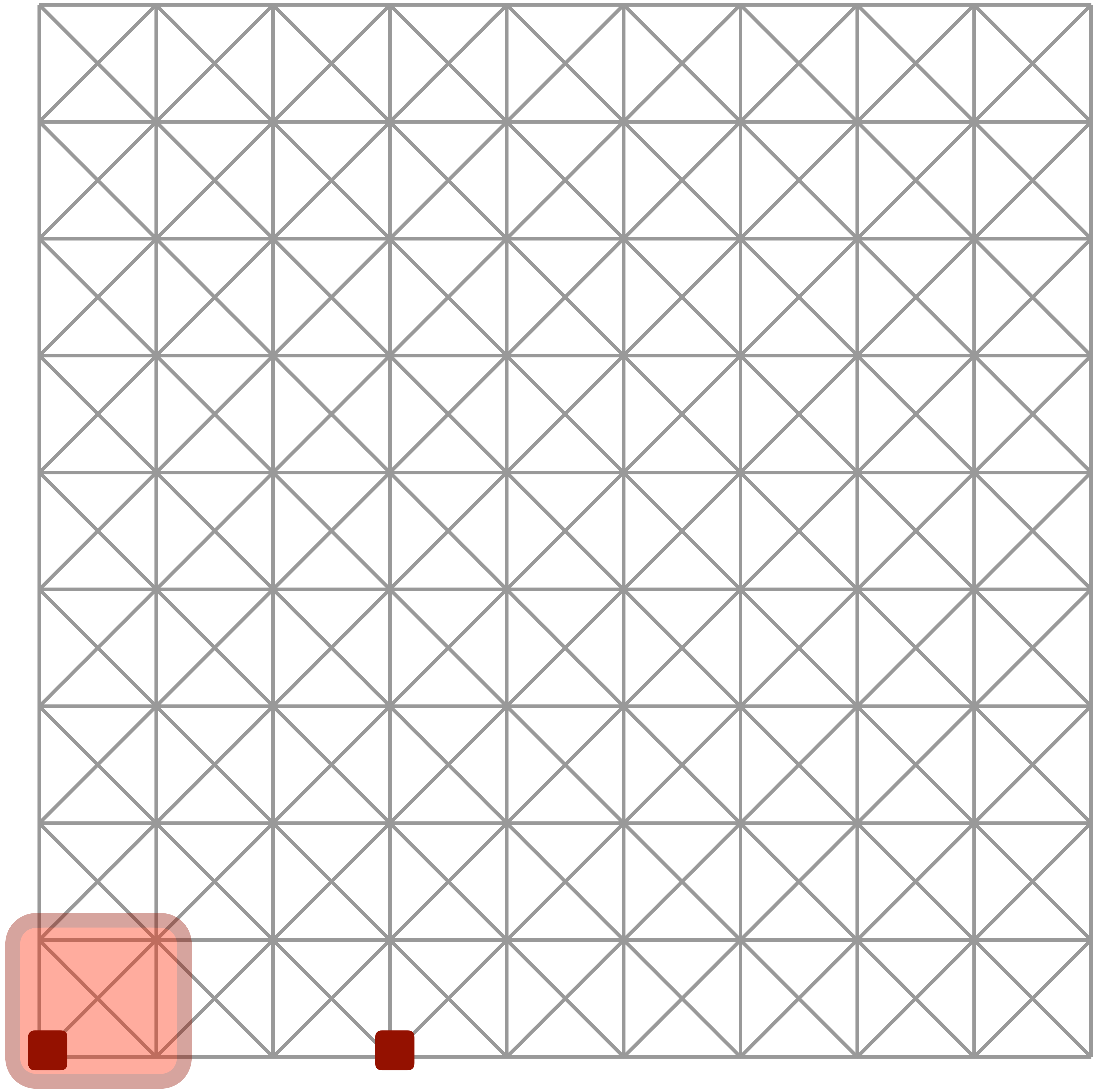
- Select next node
- If all strongly connected neighbors are unaggregated, then aggregate.

Aggregation



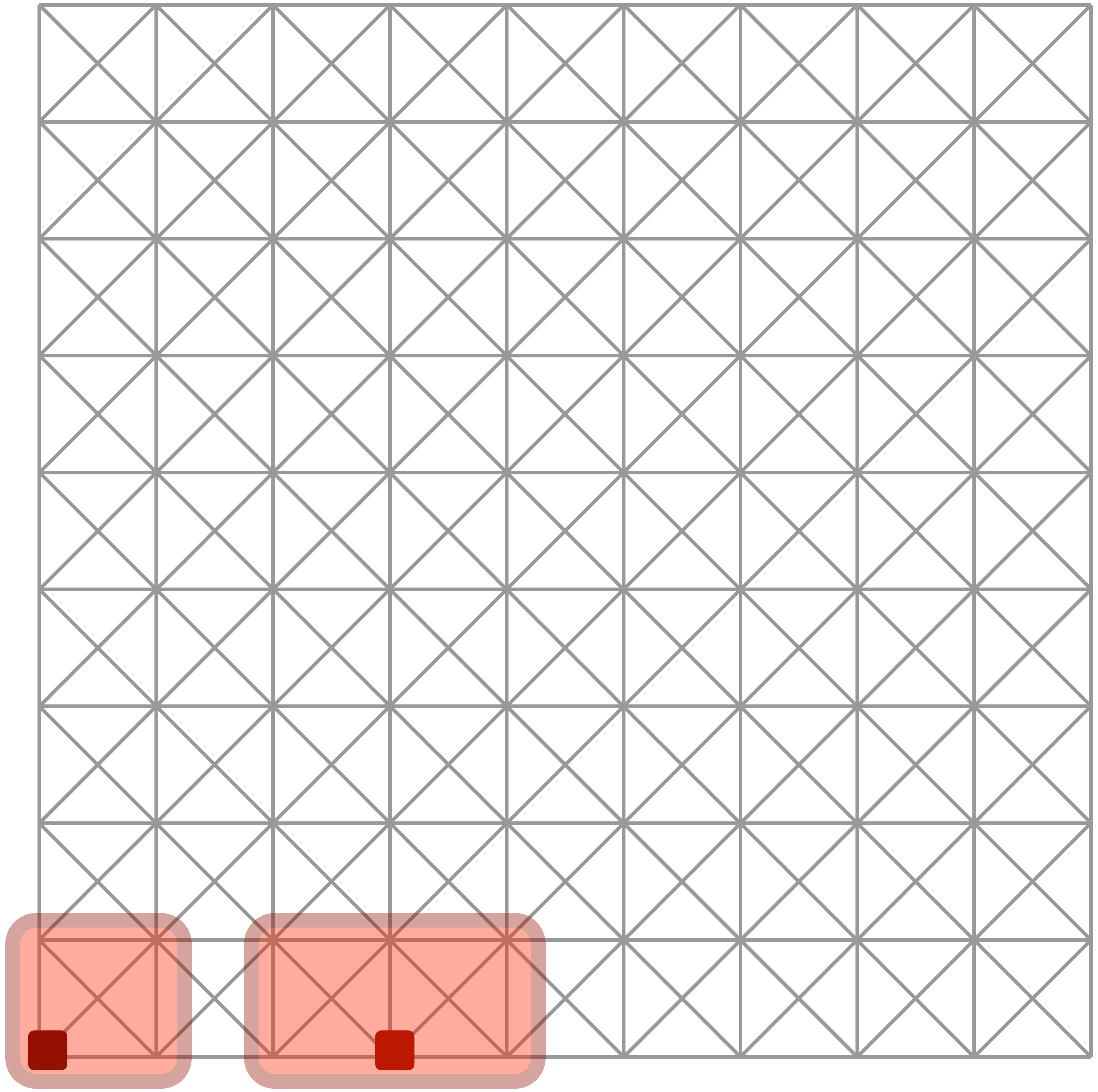
- Select next node
- If all strongly connected neighbors are unaggregated, then aggregate.

Aggregation



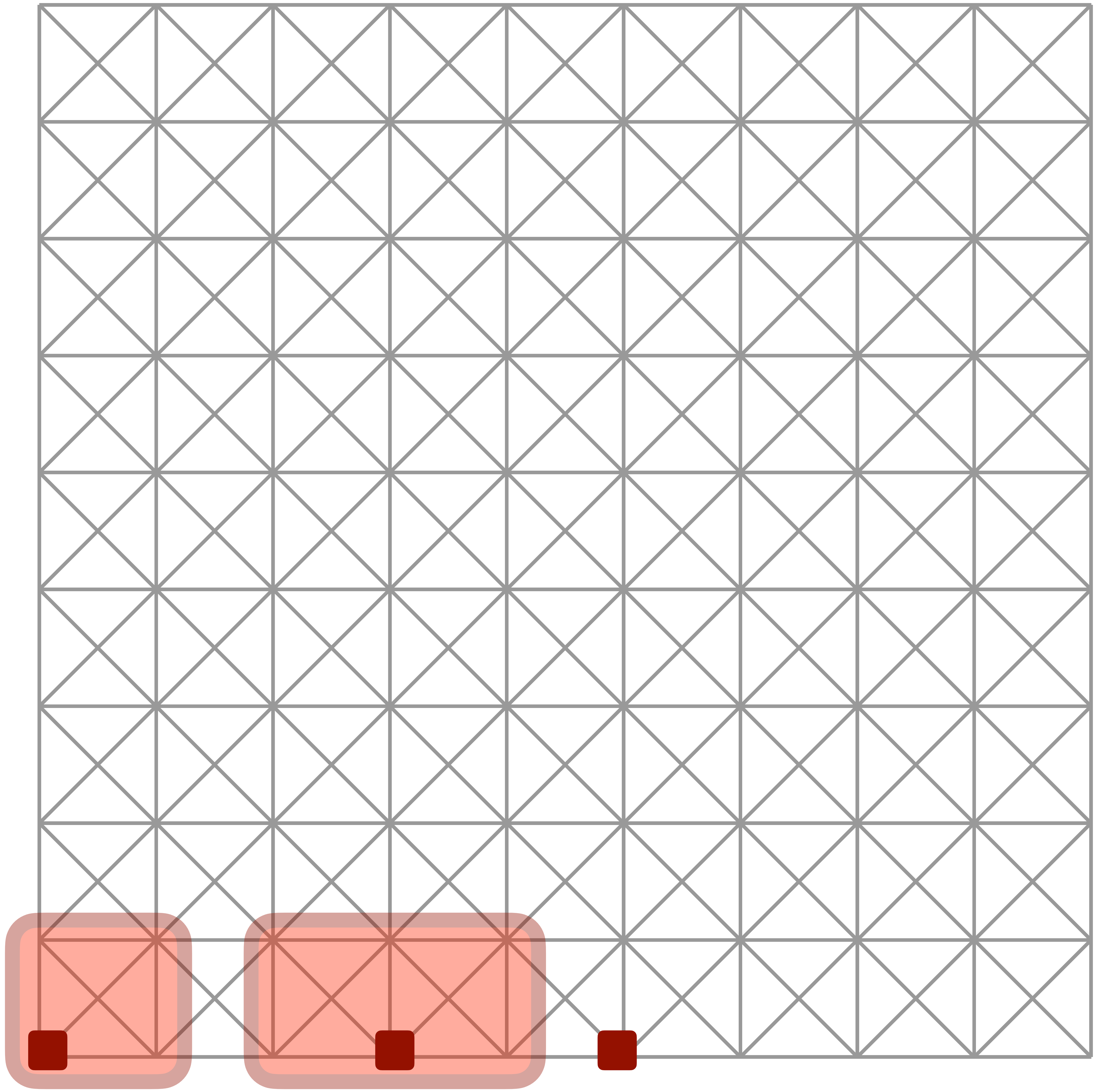
- Select next unaggregated node
- If all strongly connected neighbors are unaggregated,
  then aggregate.

Aggregation



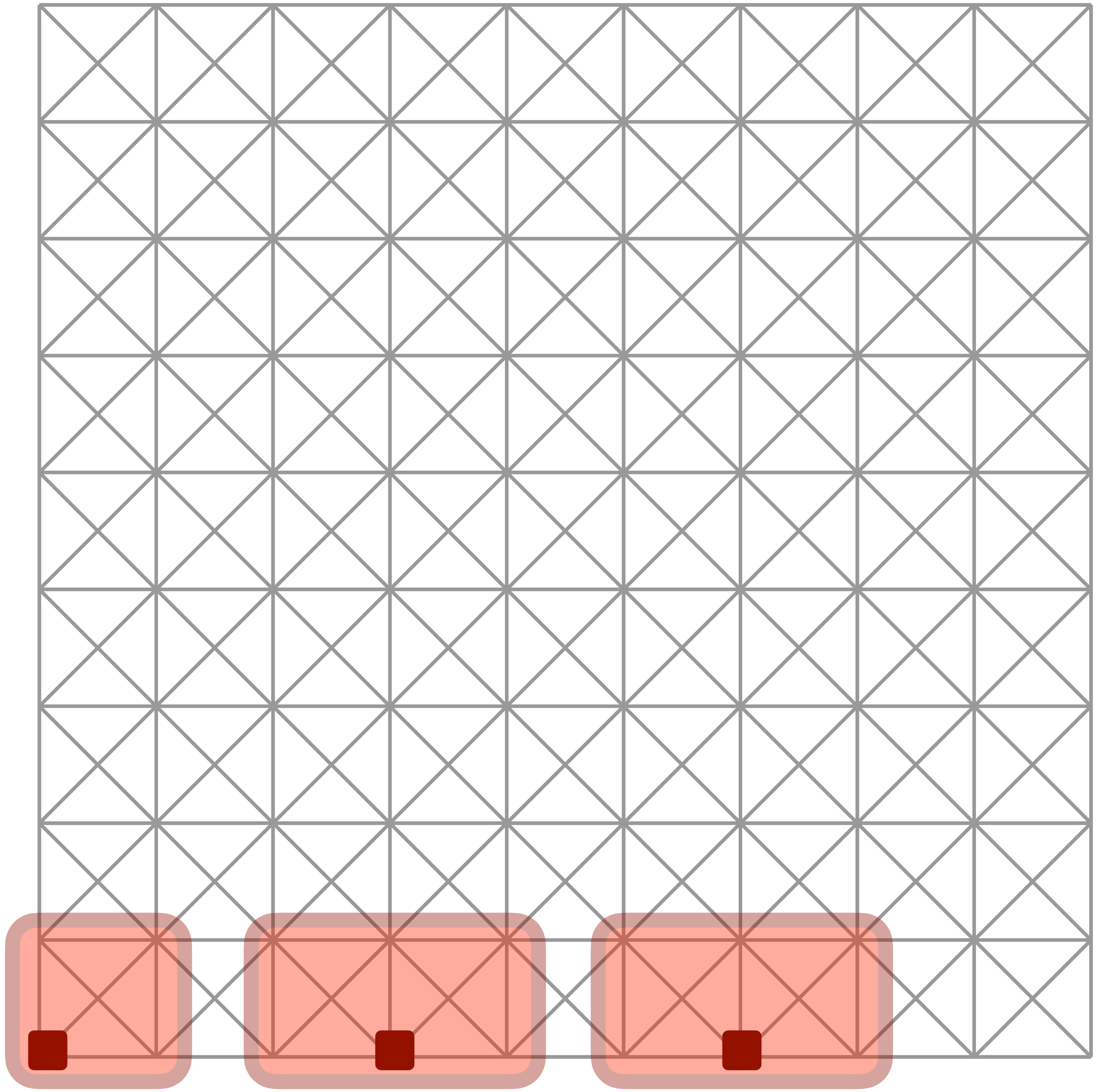
- Select next node
- If all strongly connected neighbors are unaggregated, then aggregate.

Aggregation



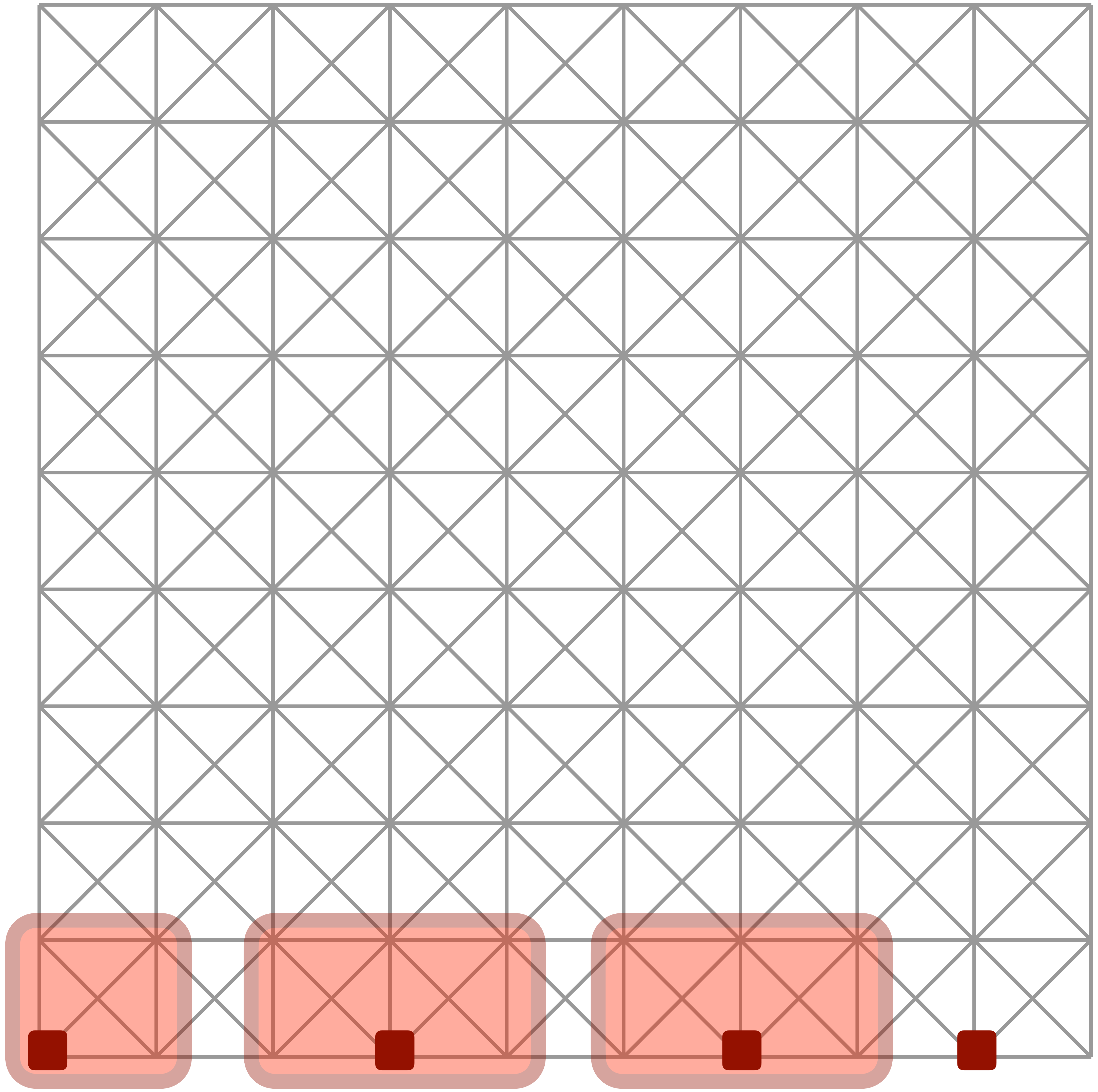
- Select next node
- If all strongly connected neighbors are unaggregated, then aggregate.

Aggregation



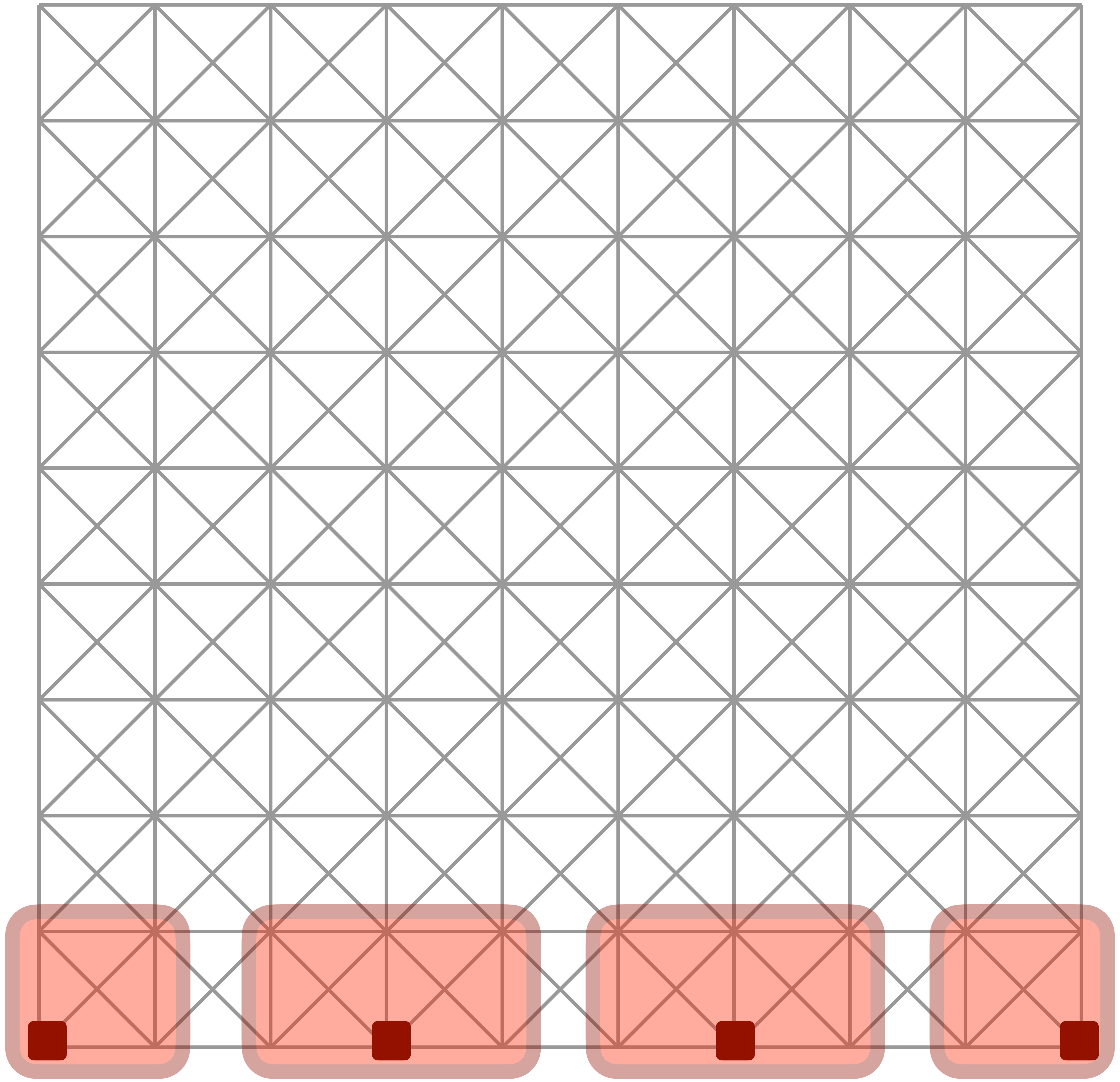
- Select next node
- If all strongly connected neighbors are unaggregated, then aggregate.

Aggregation



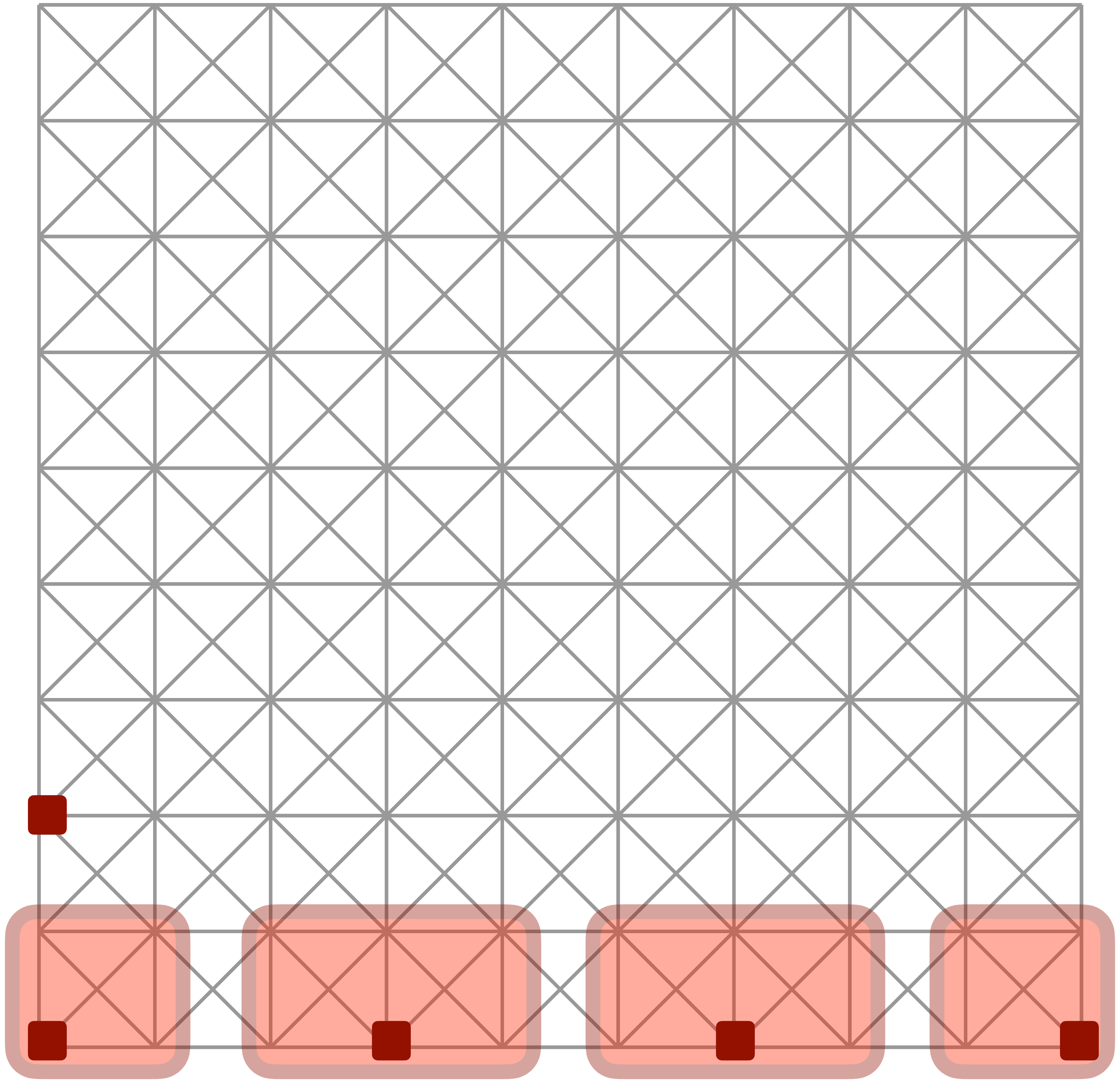
- Select next node
- If all strongly connected neighbors are unaggregated, then aggregate.

Aggregation



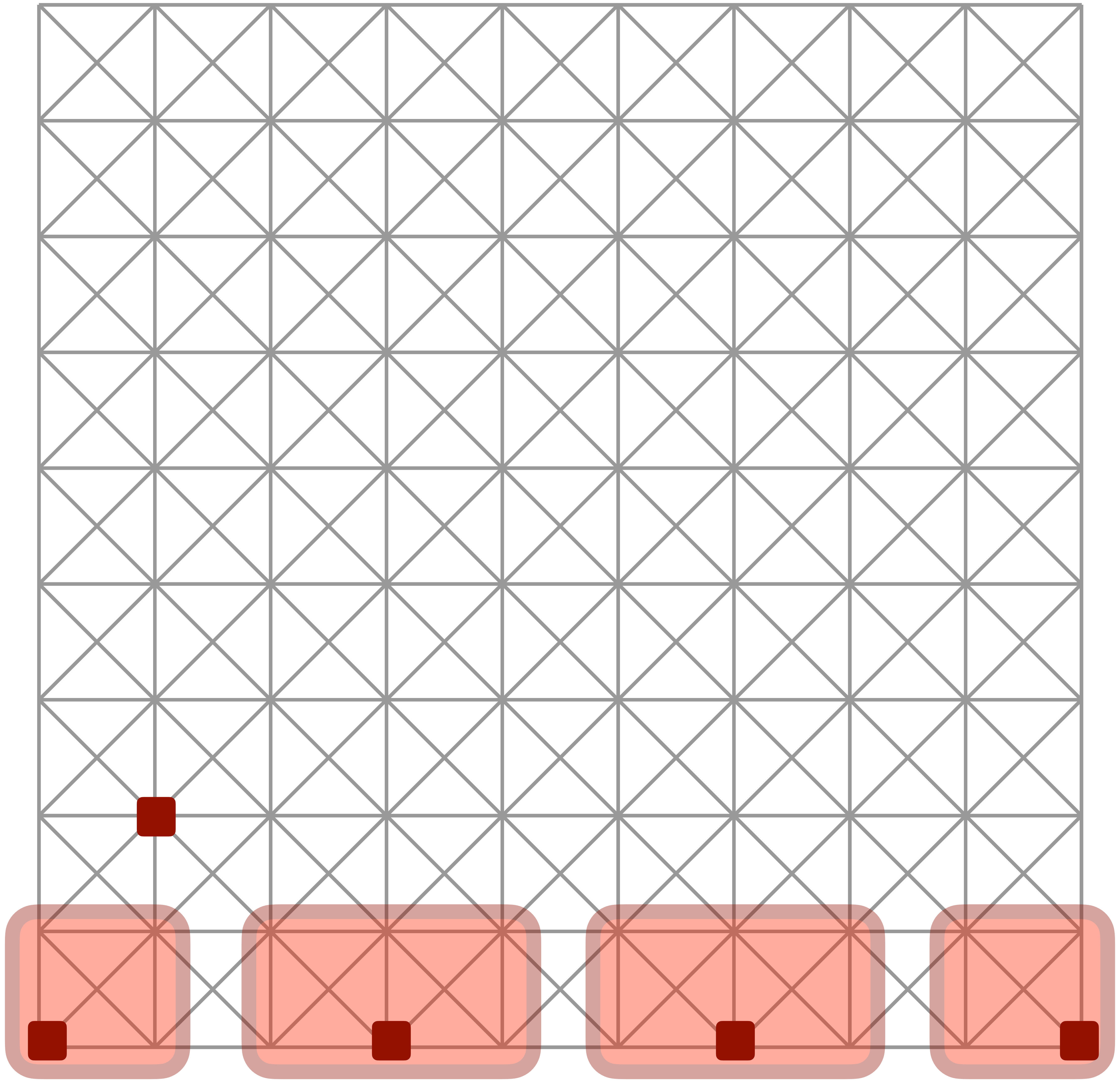
- Select next node
- If all strongly connected neighbors are unaggregated, then aggregate.

Aggregation



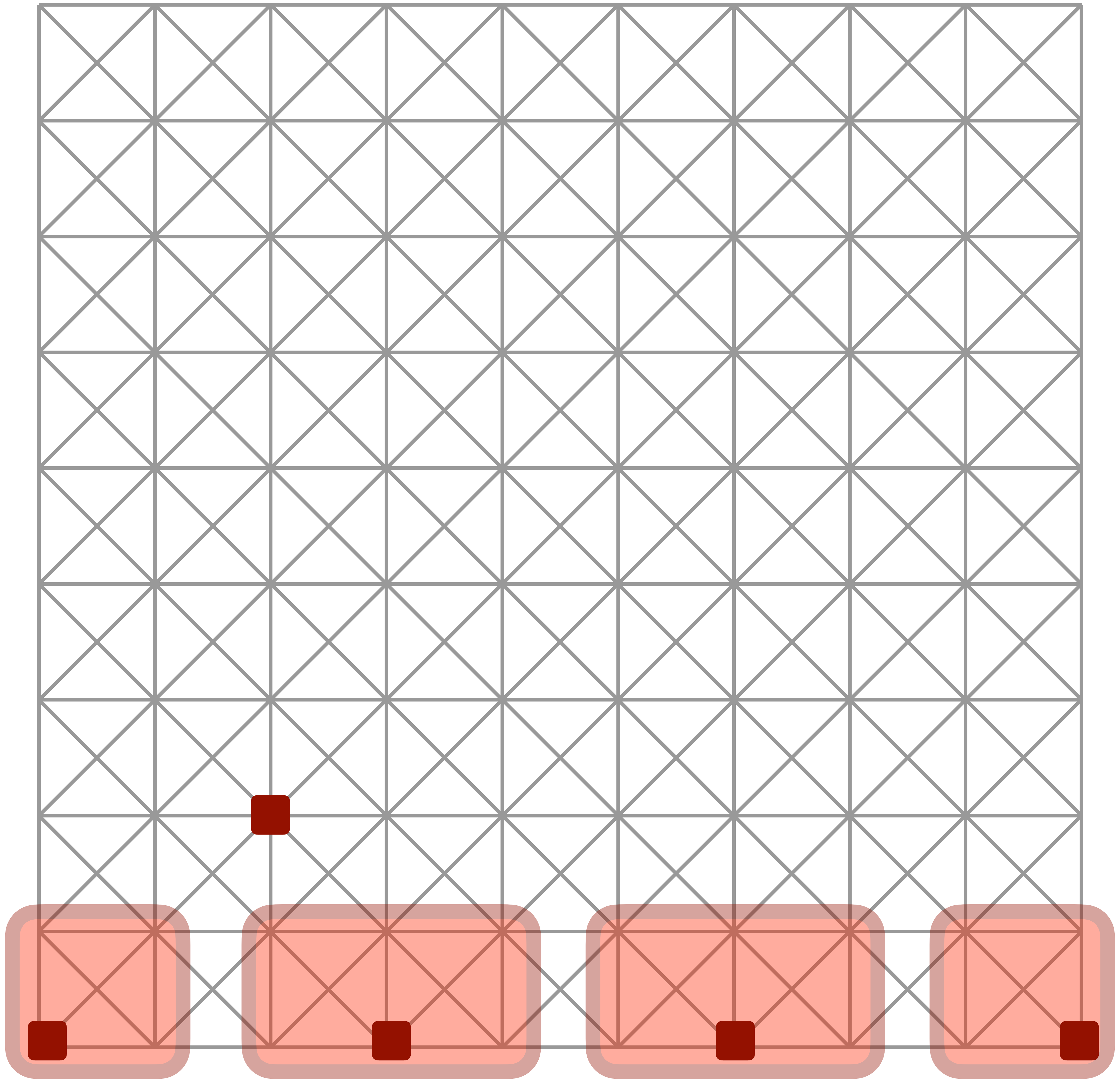
- Select next node
- If all strongly connected neighbors are unaggregated, then aggregate.

Aggregation



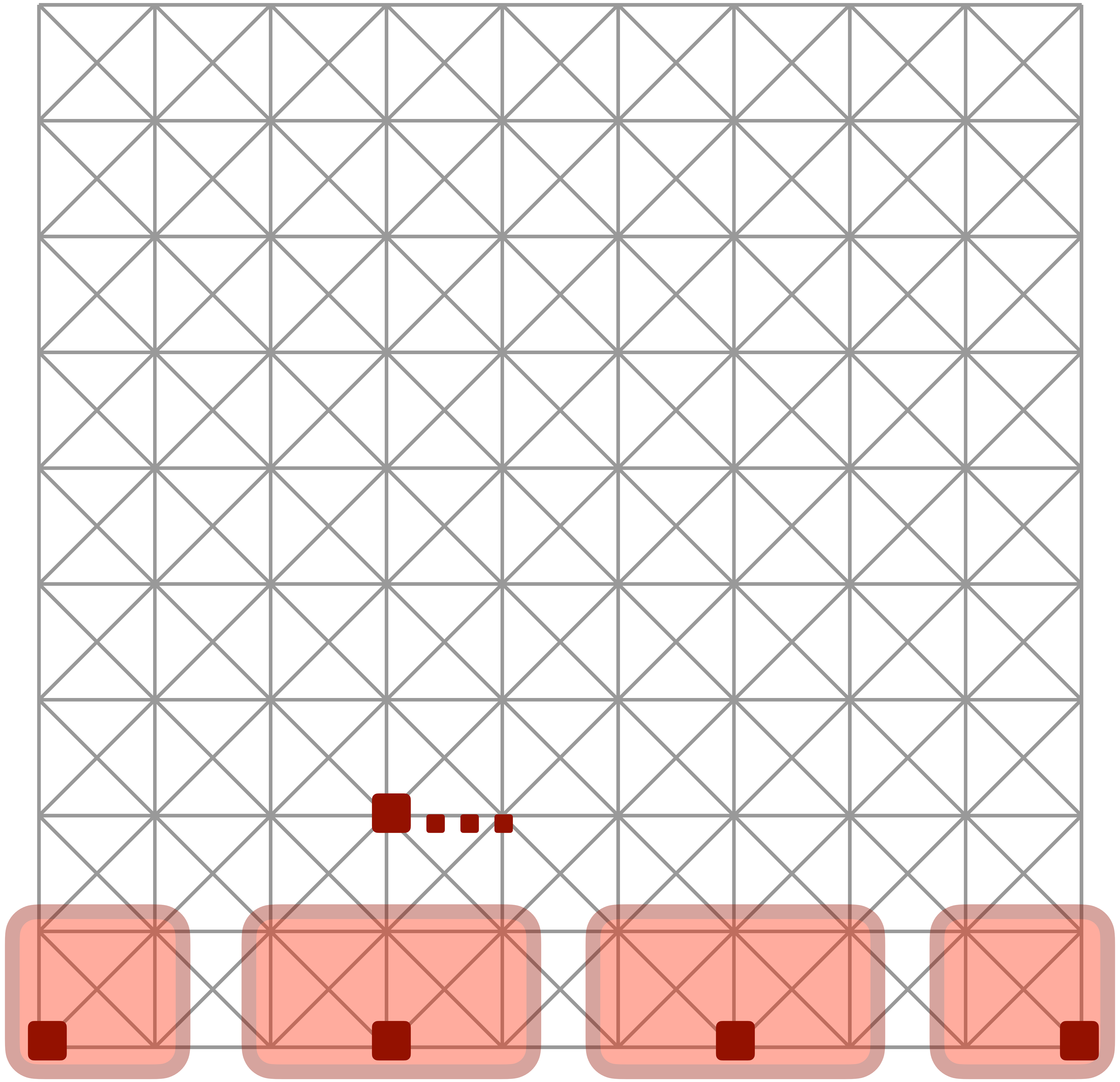
- Select next node
- If all strongly connected neighbors are unaggregated, then aggregate.

Aggregation



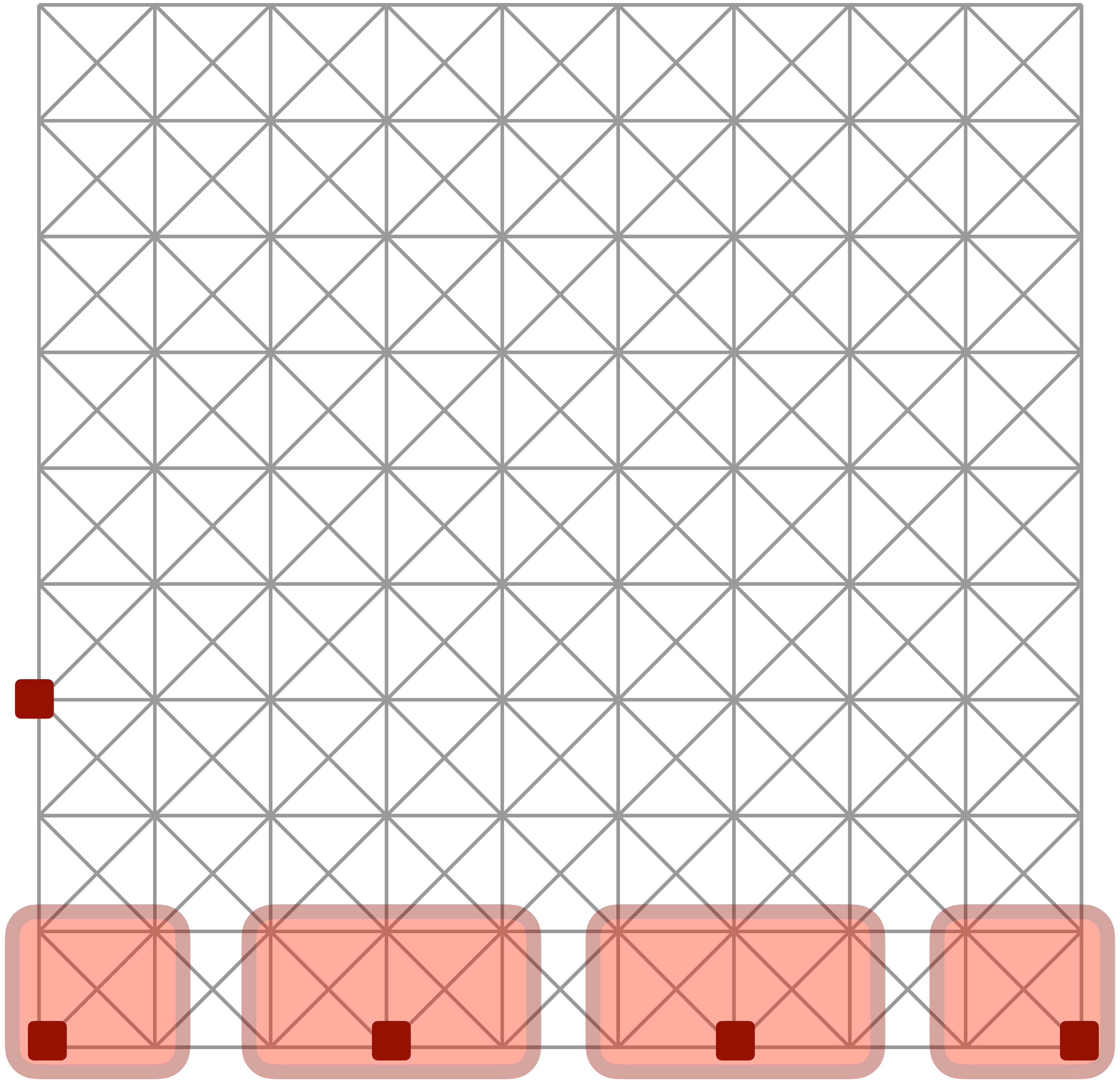
- Select next node
- If all strongly connected neighbors are unaggregated, then aggregate.

Aggregation



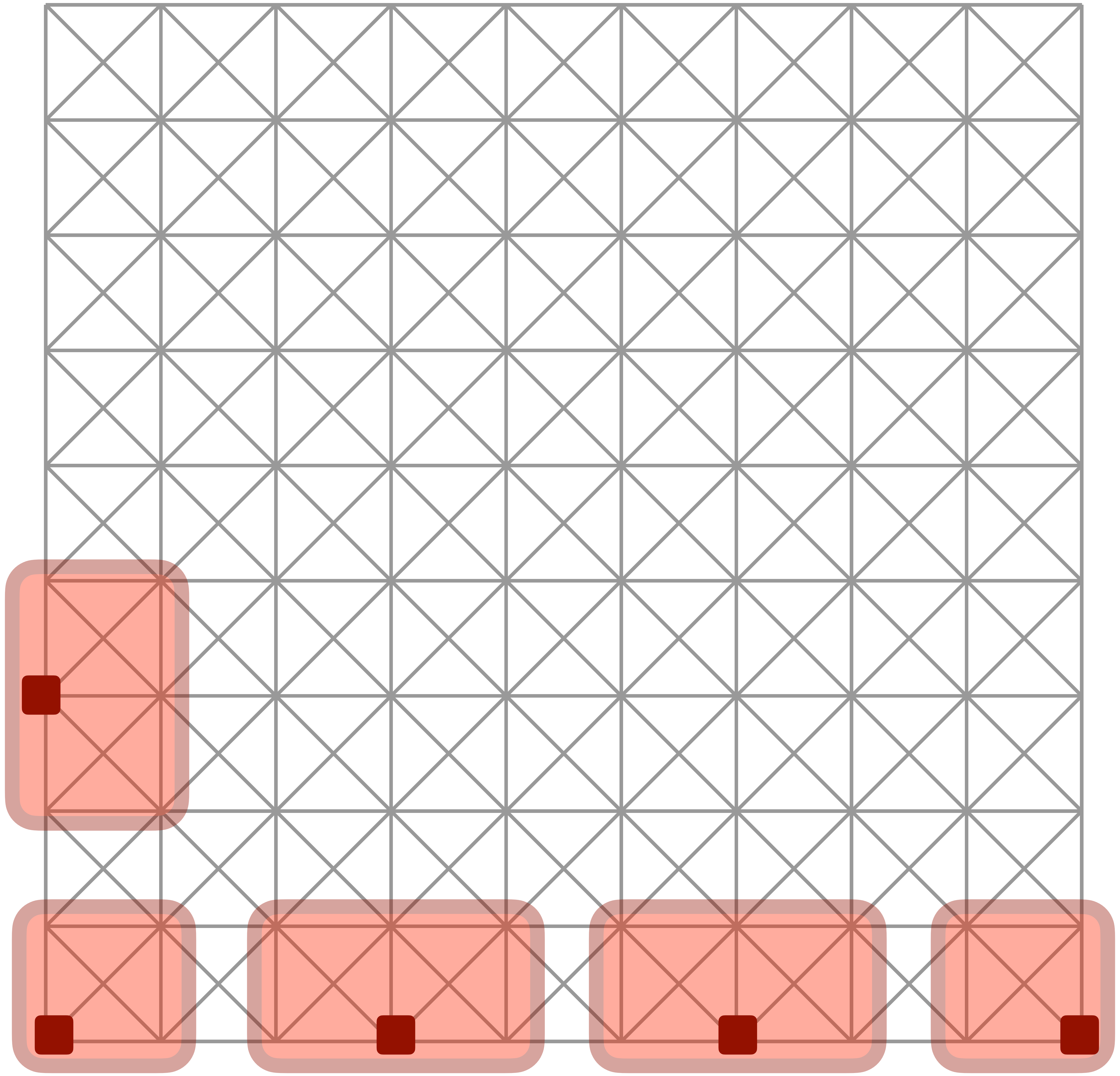
- Select next node
- If all strongly connected neighbors are unaggregated, then aggregate.

Aggregation



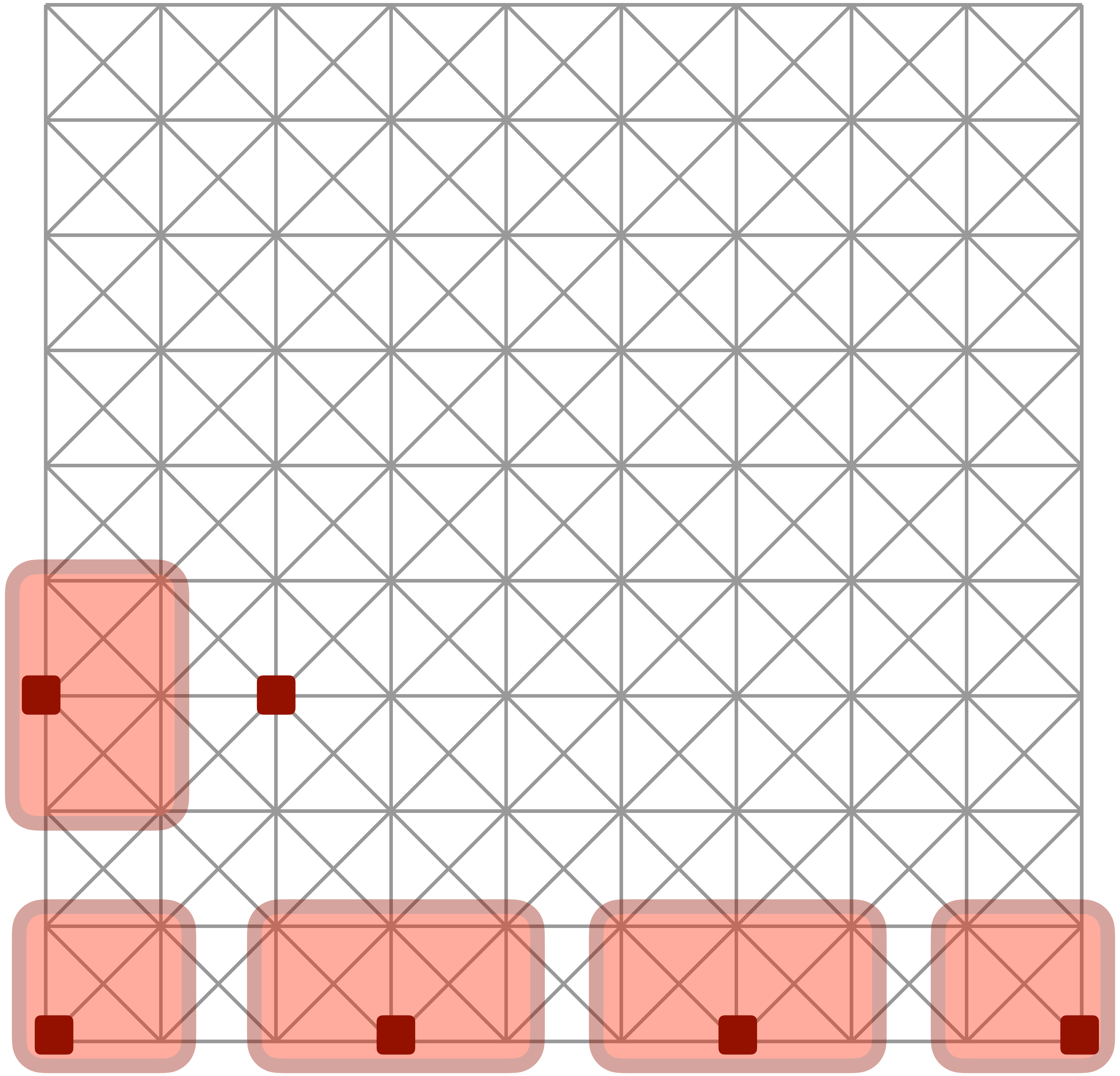
- Select next node
- If all strongly connected neighbors are unaggregated, then aggregate.

Aggregation



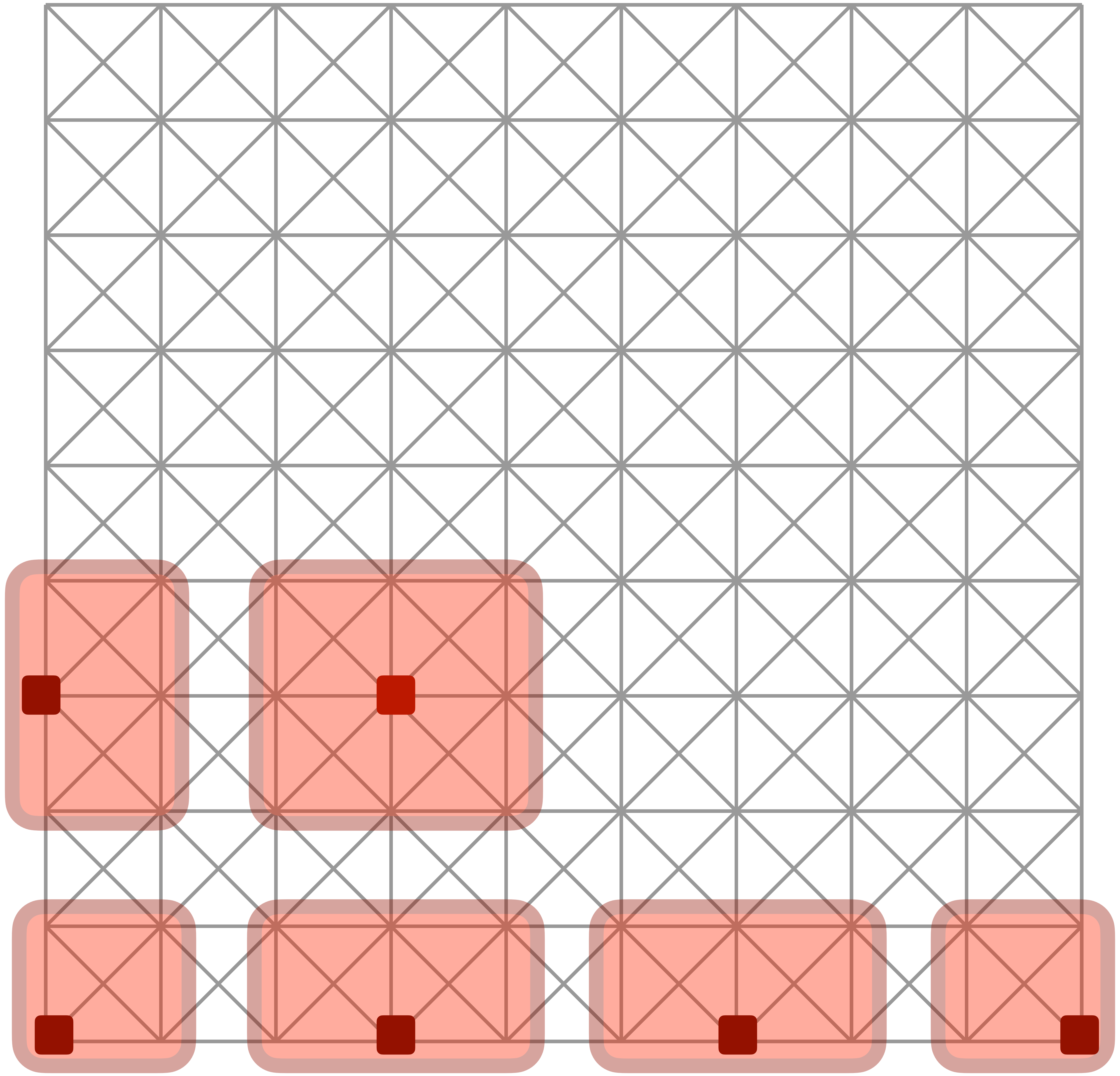
- Select next node
- If all strongly connected neighbors are unaggregated, then aggregate.

Aggregation



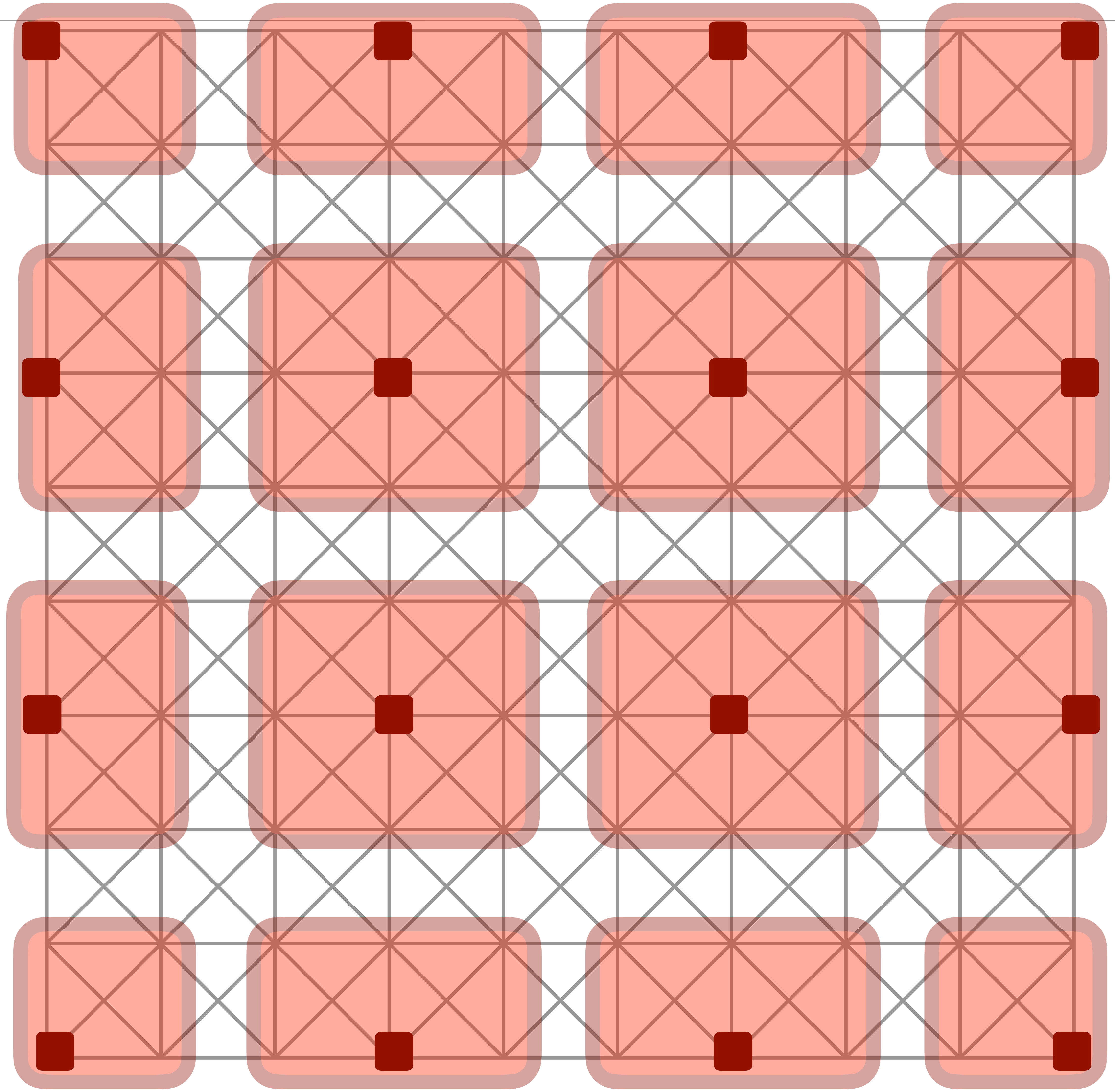
- Select next node
- If all strongly connected neighbors are unaggregated, then aggregate.

Aggregation



- Select next node
- If all strongly connected neighbors are unaggregated, then aggregate.

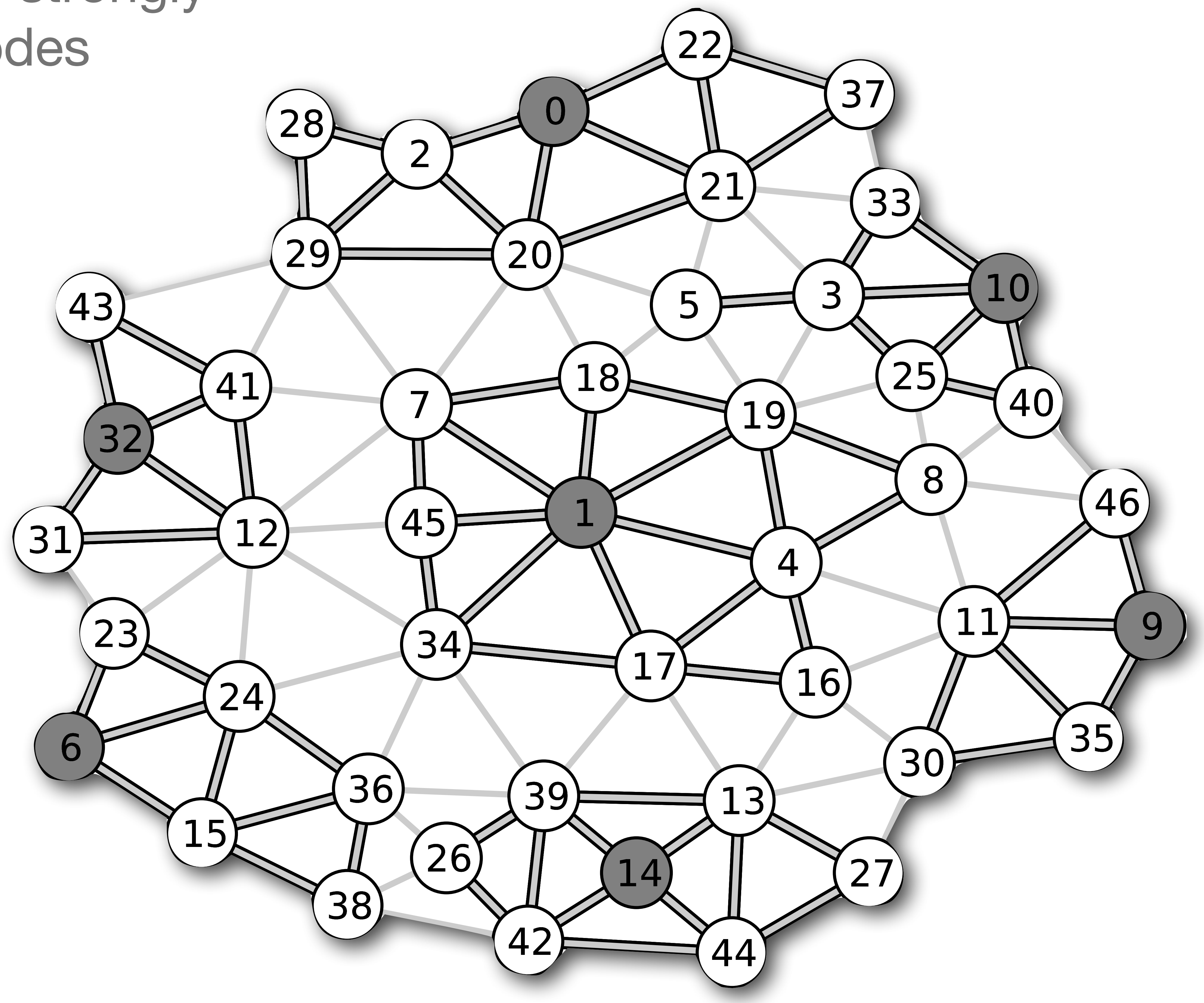
Aggregation



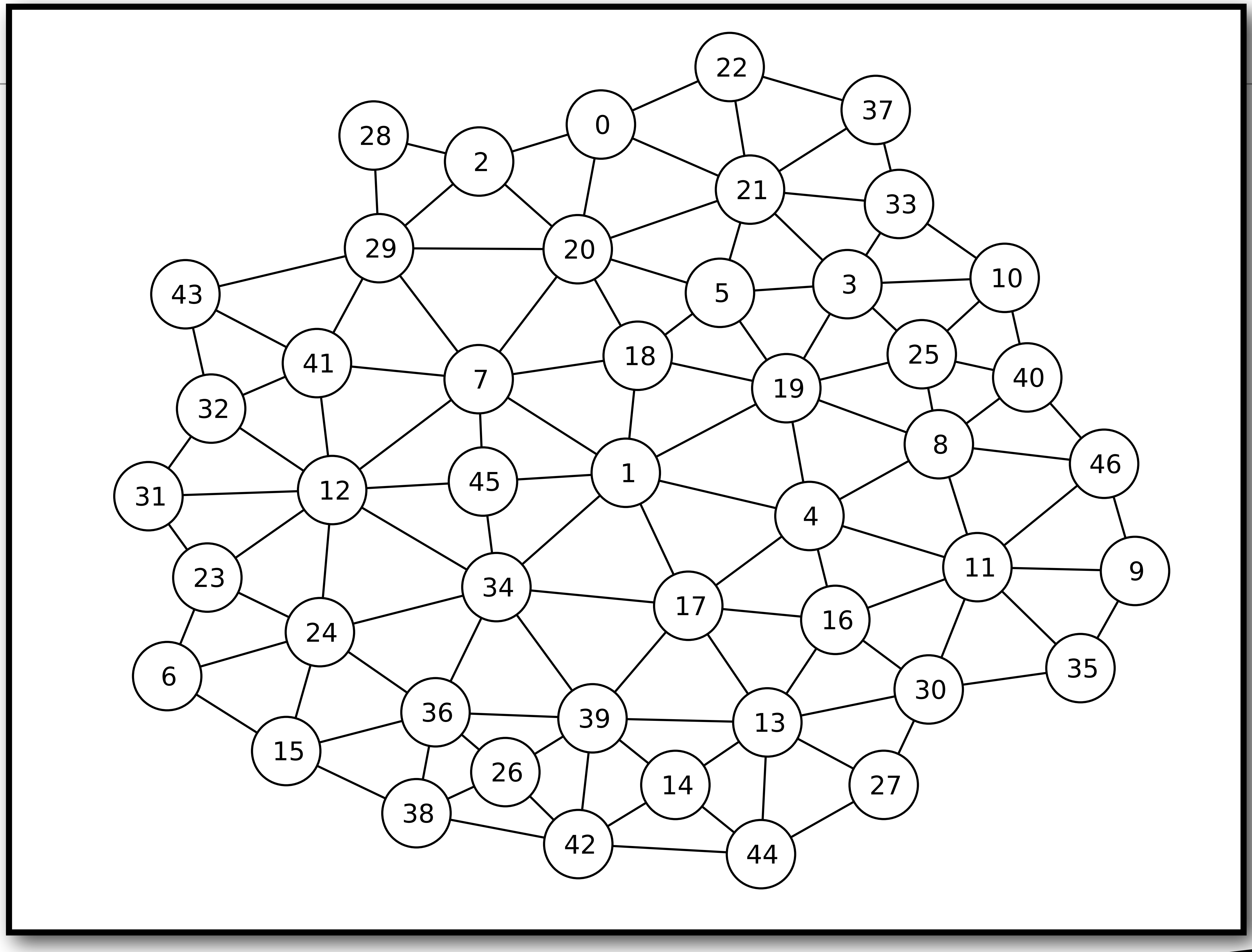
- Select next node
- If all strongly connected neighbors are unaggregated, then aggregate.

MIS-based Graph Aggregation

- Greedy: group collections of strongly connected unaggregated nodes
- Problems:
  1. size fixed
  2. sequential (greedy)



MIS(2)



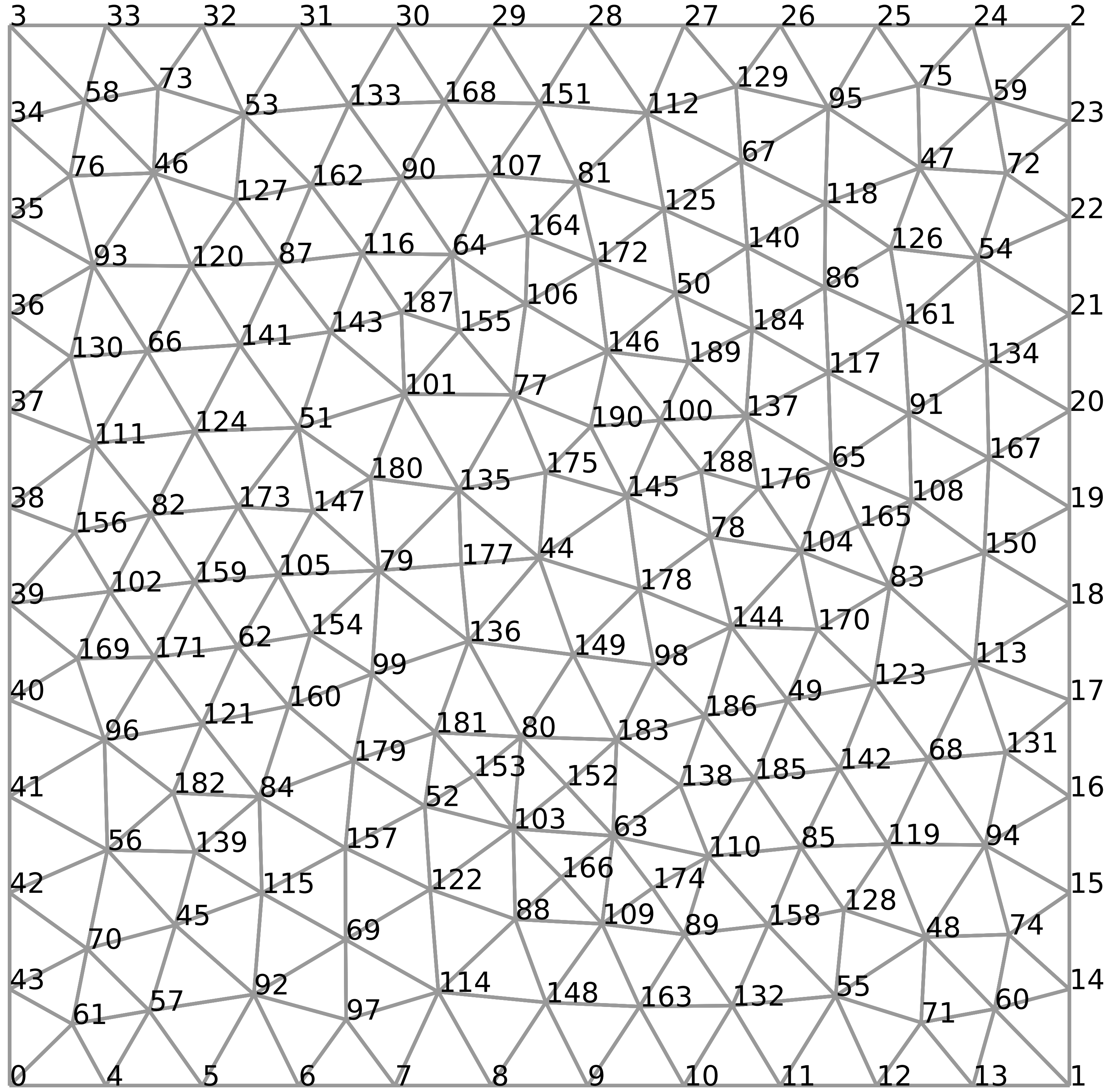
independent

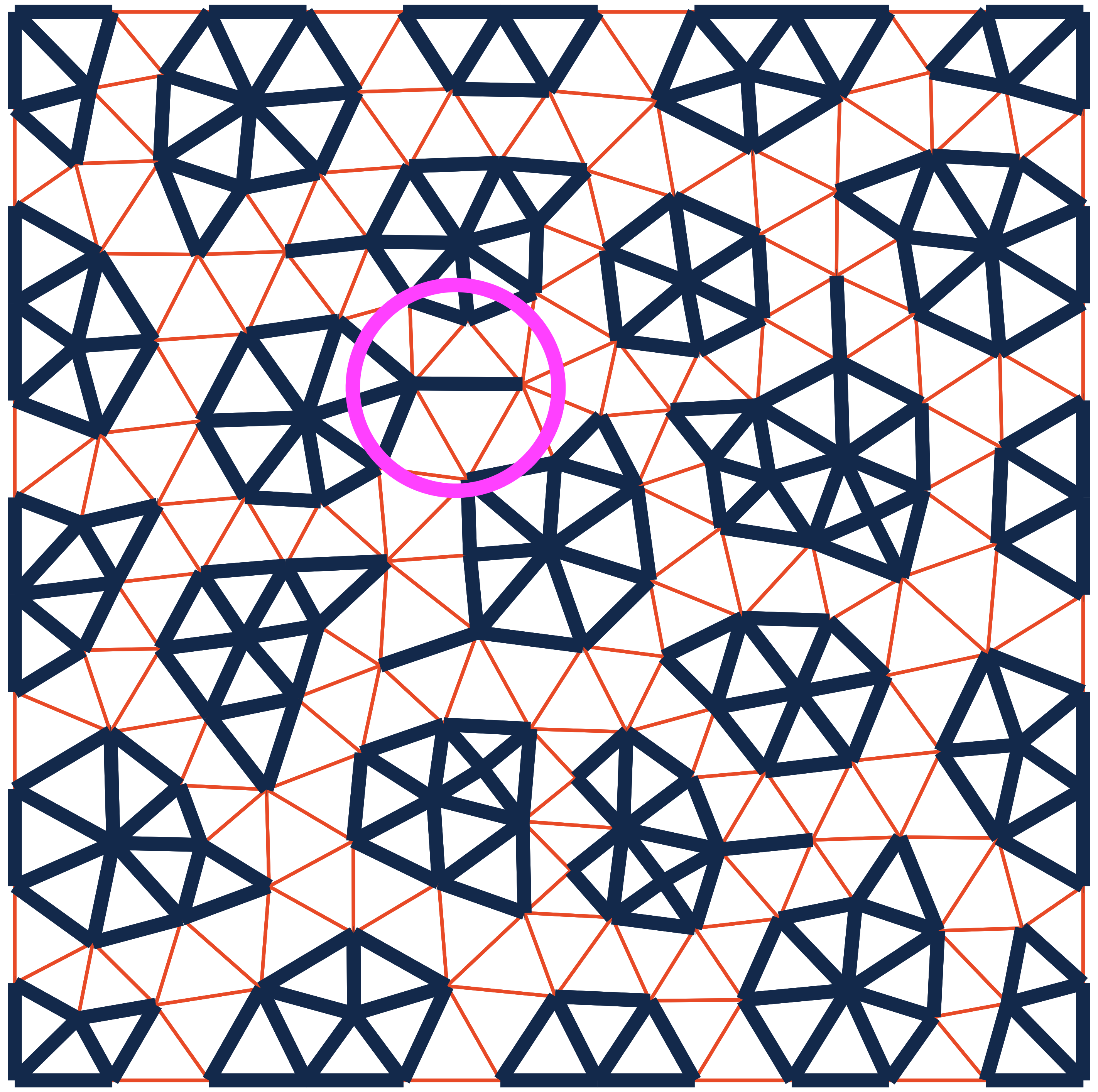
- root nodes more than 2 edges apart (> distance-2)
- an unaggregated node more than 2 edges from a root can become a root

} MIS(2)

maximal

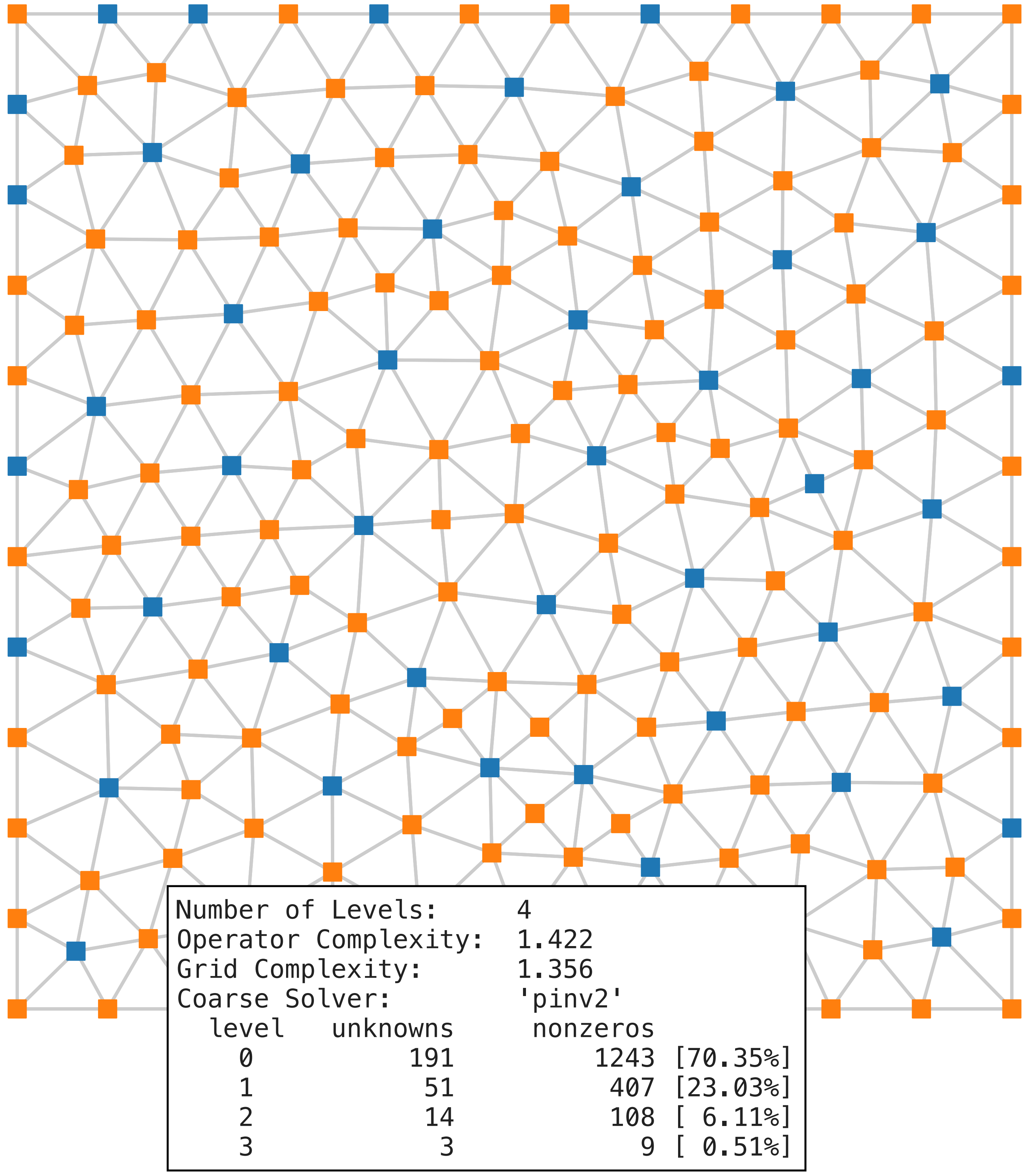
Aggregation

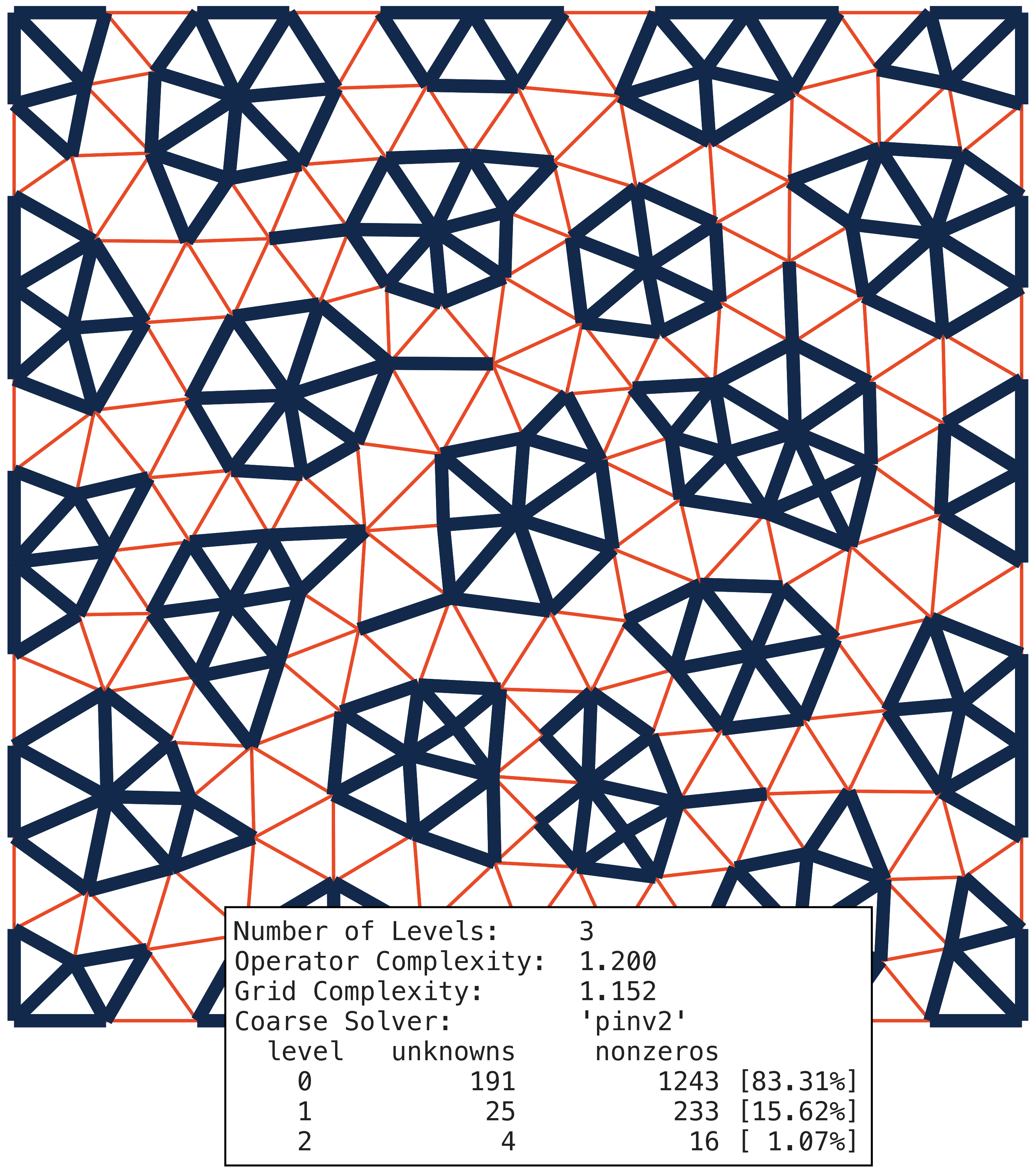




- Second passes (often) needed
- Options:
  - group with largest
  - regroup
  - group with strongest

Aggregation vs splitting

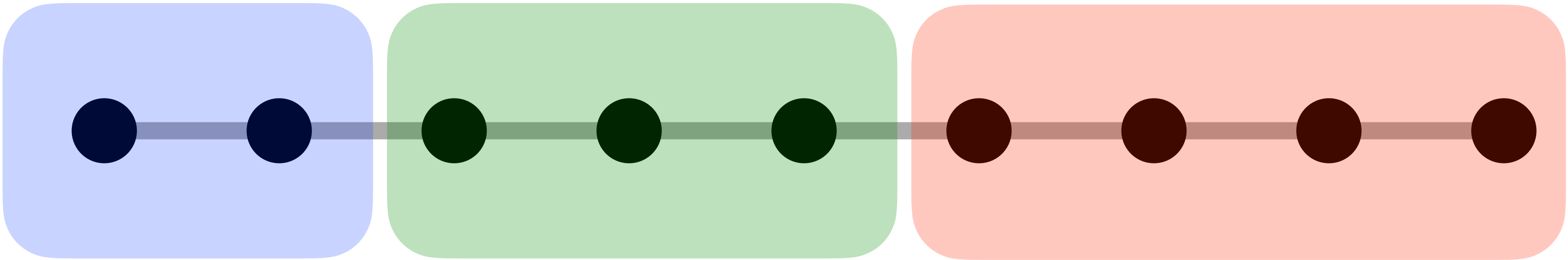




- Aggregation generally more aggressive coarsening

SA AMG Interpolation

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



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

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

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

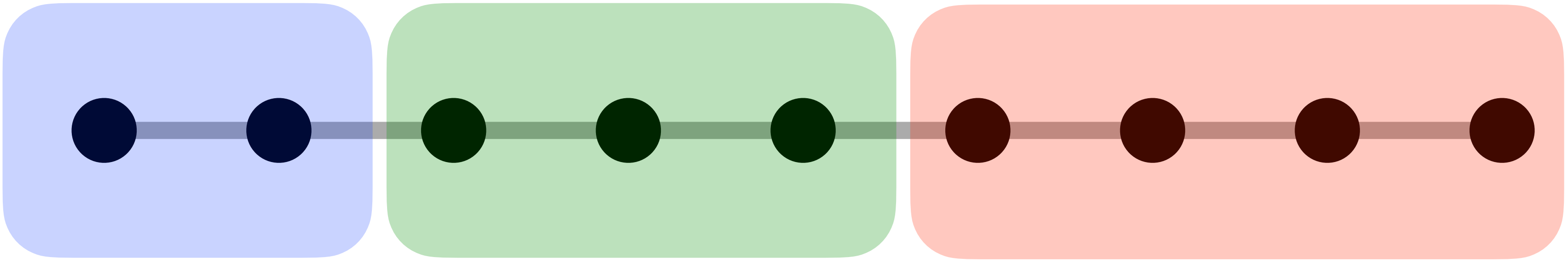
$$T = \begin{bmatrix} Q1 & & \\ & Q2 & \\ & & Q3 \end{bmatrix}$$

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

$$\boxed{TB_C = B}$$

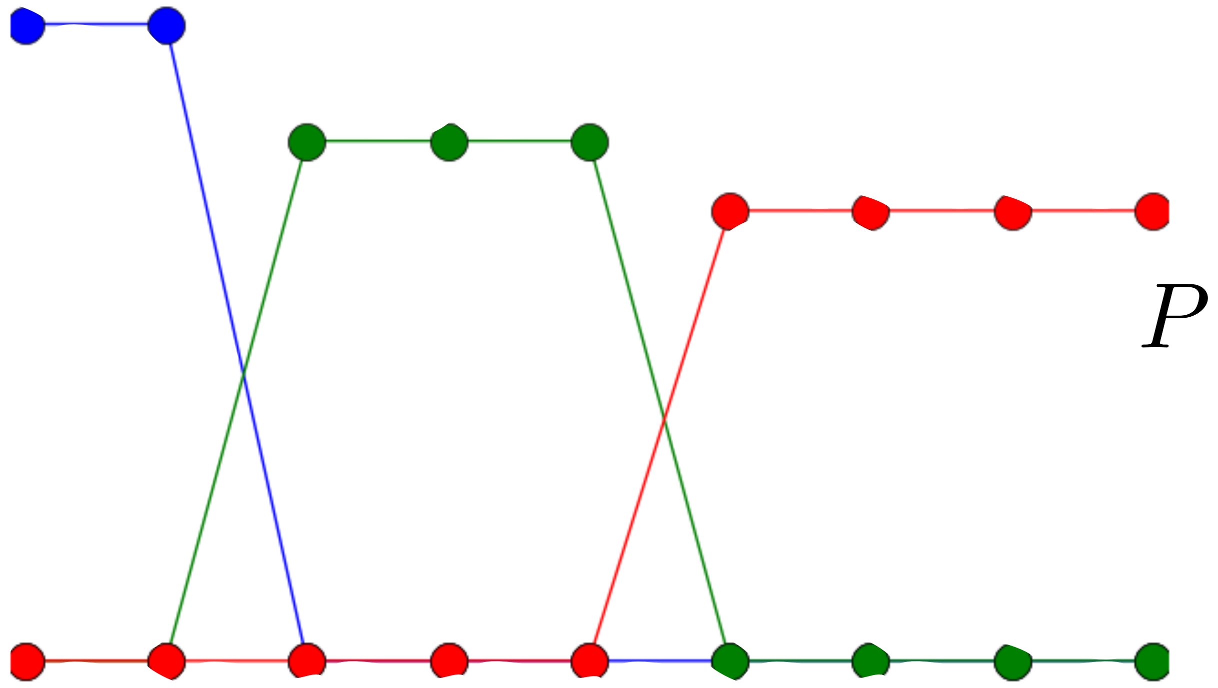
SA AMG Interpolation

- Example

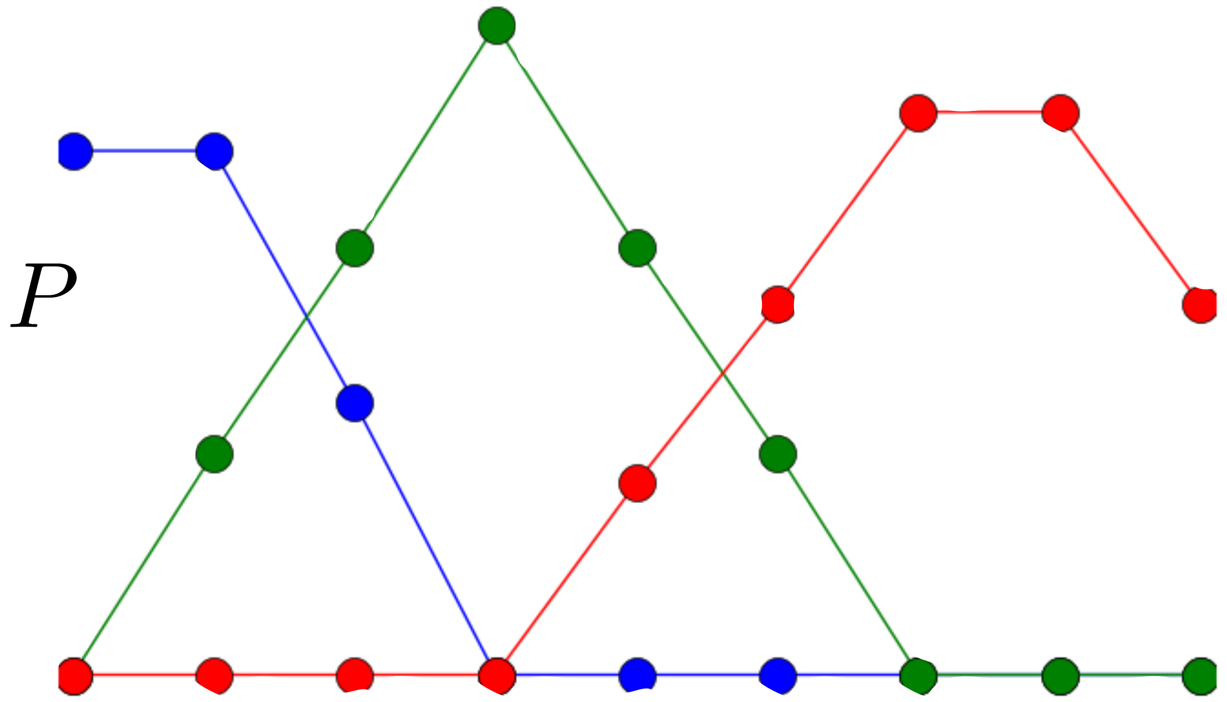


$$\hat{T} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \qquad T = \begin{bmatrix} \frac{\sqrt{2}}{2} & 0 & 0 \\ \frac{\sqrt{2}}{2} & 0 & 0 \\ 0 & \frac{\sqrt{3}}{3} & 0 \\ 0 & \frac{\sqrt{3}}{3} & 0 \\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

- Now make interpolation **better**



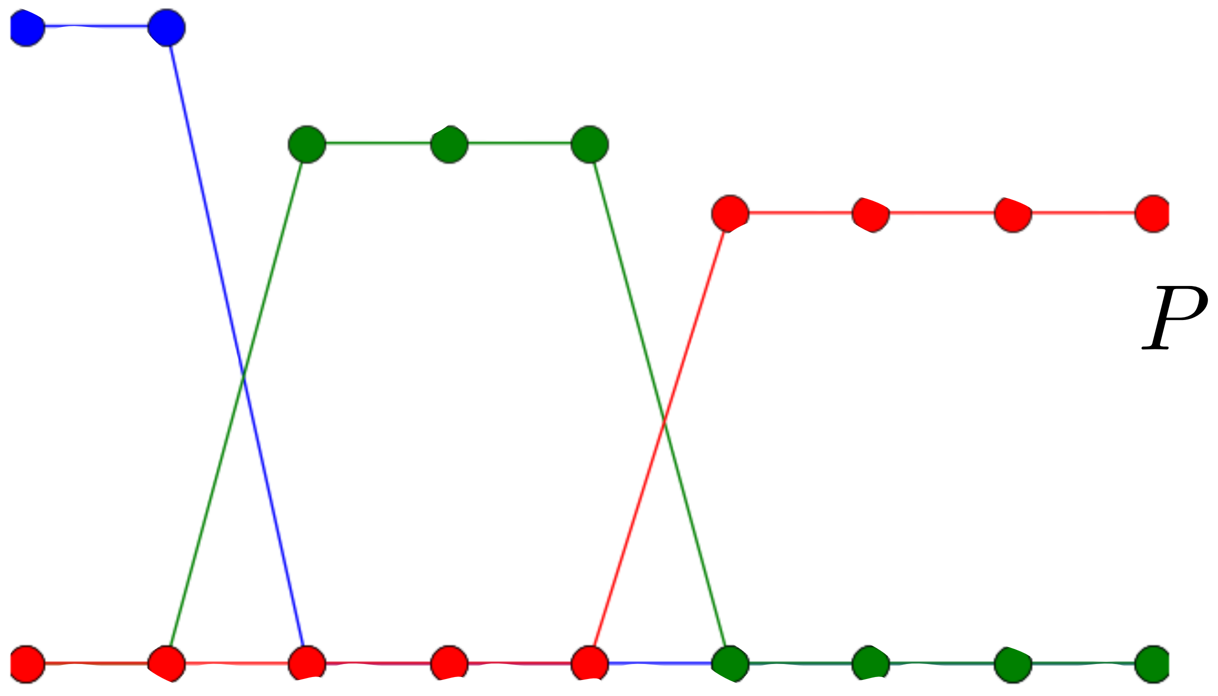
$$P \leftarrow (I - \omega D^{-1} A) P$$



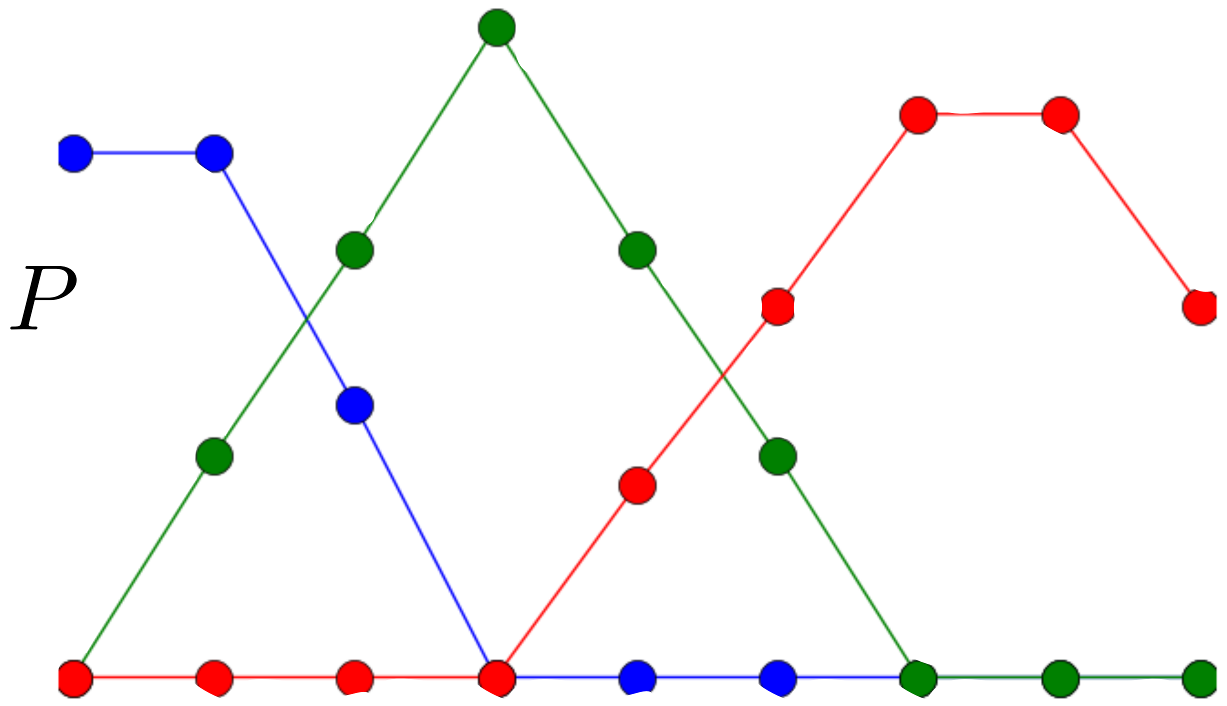
SA AMG Interpolation

- 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**

Demo: 17-smoothed-aggregation-1d.ipynb

Aggregation Based Setup

$$\boxed{B}$$

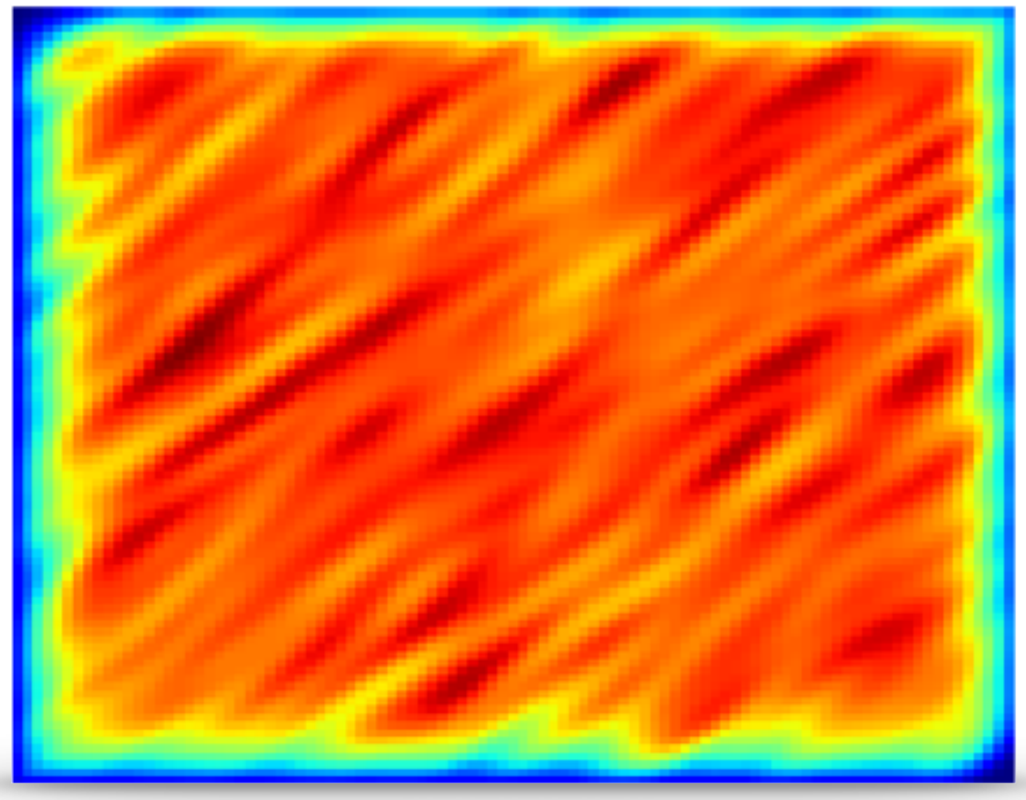
$$S \leftarrow \mathrm{strength}(A, B)$$

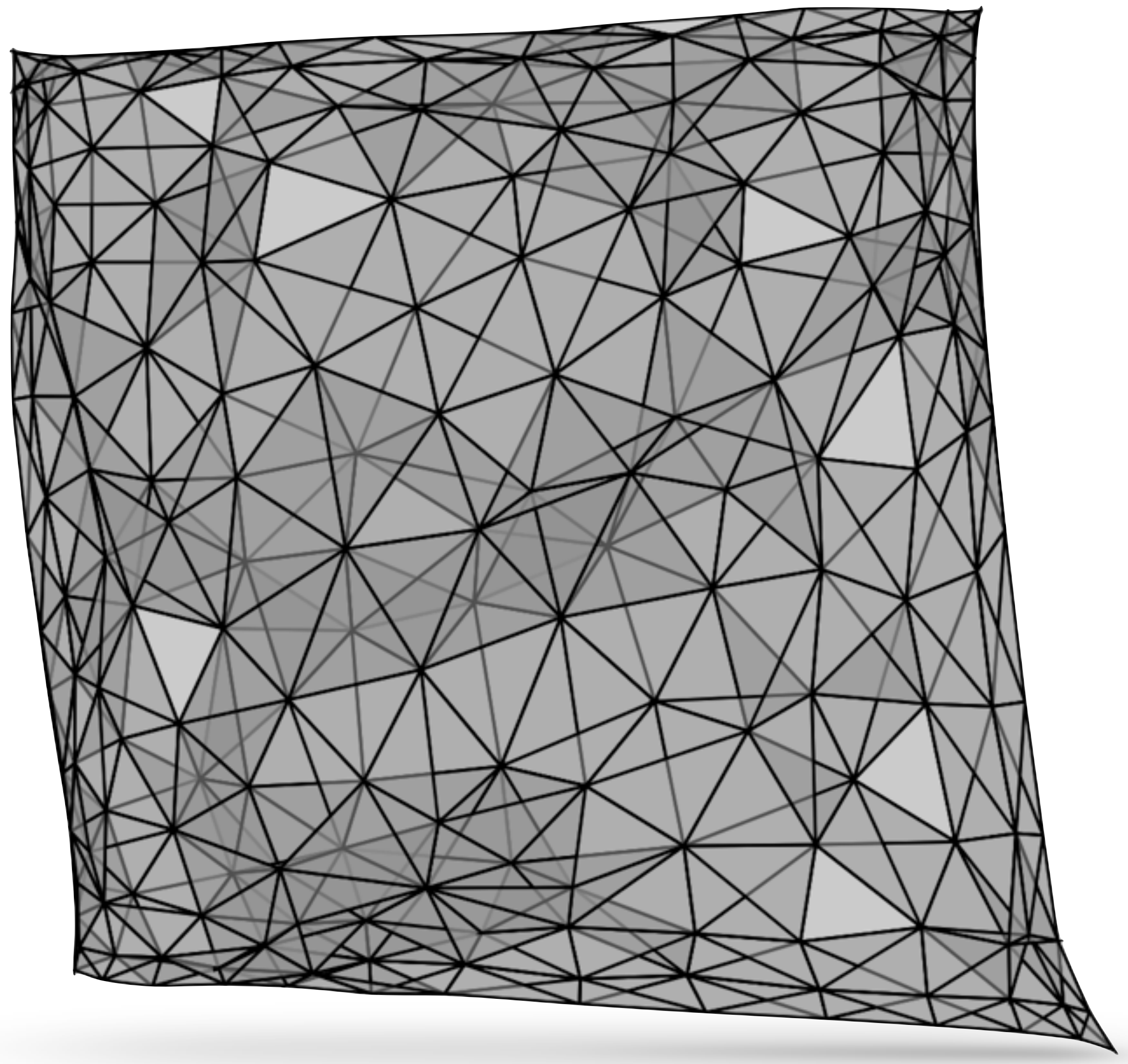
$$C \leftarrow \mathrm{aggregate}(S)$$

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

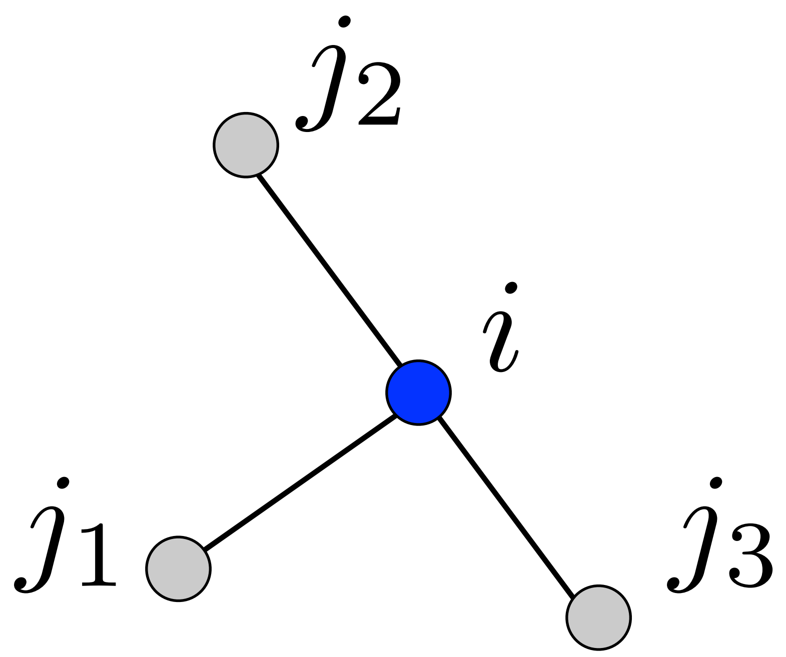
$$P \leftarrow \mathrm{improve}(A, P^{(0)})$$

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





Aggregation Based Setup



$$B$$

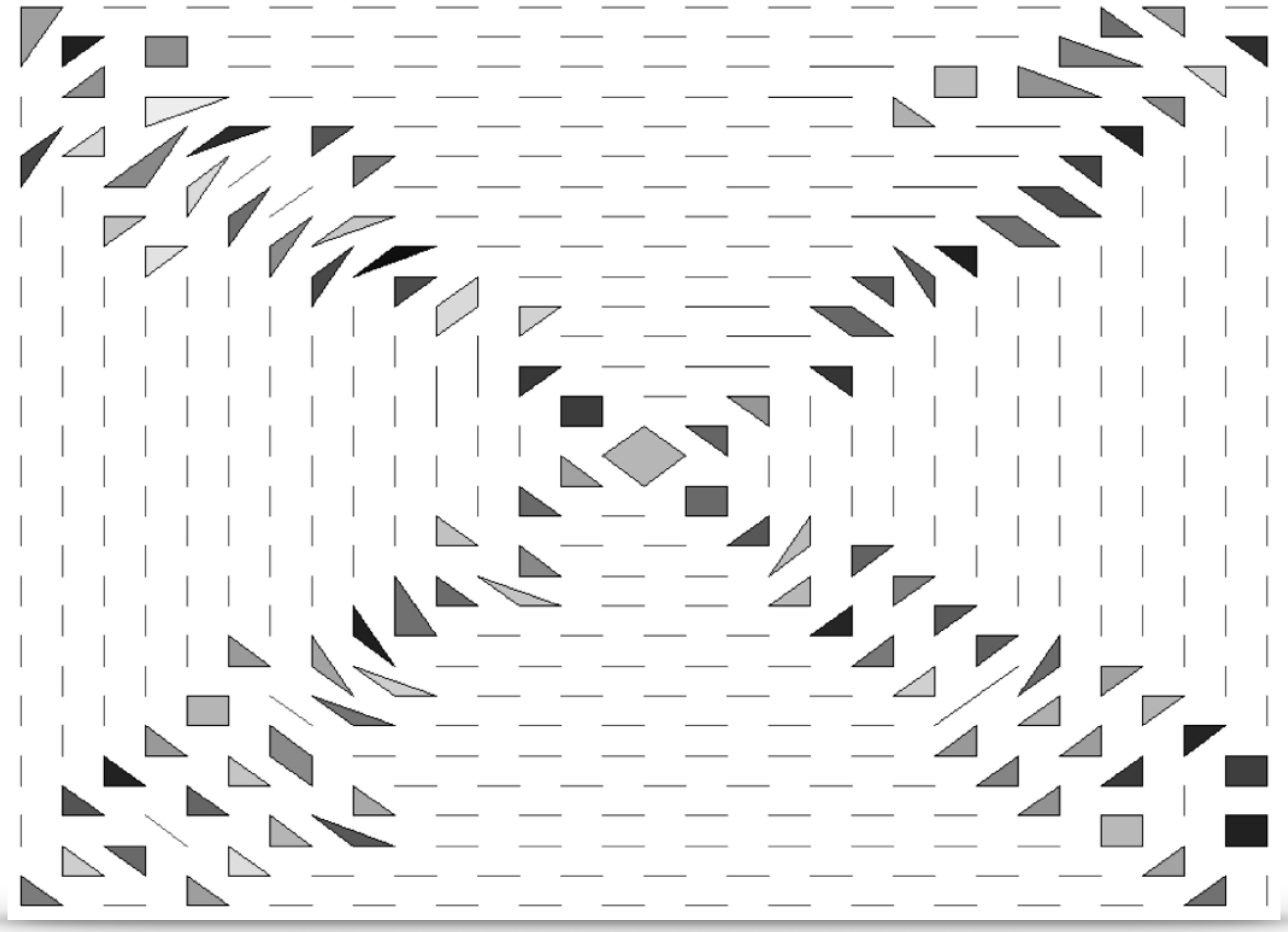
$$\boxed{S \leftarrow \text{strength}(A, B)}$$

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

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

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

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



Aggregation Based Setup

$$B$$

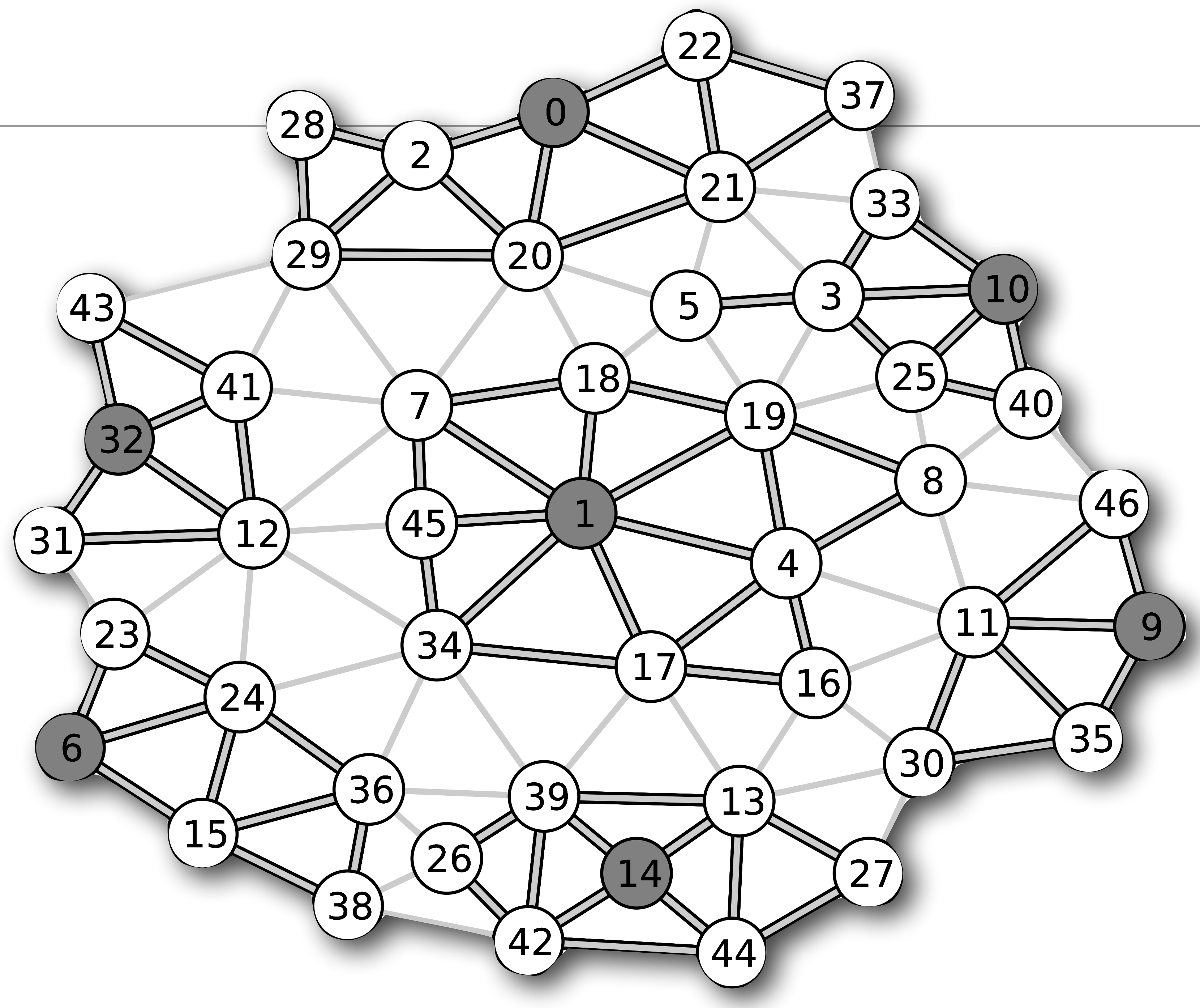
$$S \leftarrow \text{strength}(A, B)$$

$$\boxed{C \leftarrow \text{aggregate}(S)}$$

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

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

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



- Group nodes
- based on strong connections
- variable size

Aggregation Based Setup

$$B$$

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

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

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

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

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

$$B = \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \\ b_{20} & b_{21} \\ b_{30} & b_{31} \\ b_{40} & b_{41} \\ b_{50} & b_{51} \\ b_{60} & b_{61} \\ b_{70} & b_{71} \\ b_{80} & b_{81} \\ b_{90} & b_{91} \end{bmatrix}$$

$$P_{tent} = \begin{bmatrix} b_{00} & b_{01} & & & & & & \\ b_{10} & b_{11} & & & & & & \\ & & b_{20} & b_{21} & & & & \\ & & b_{30} & b_{31} & & & & \\ & & b_{40} & b_{41} & & & & \\ & & & & b_{50} & b_{51} & & \\ & & & & b_{60} & b_{61} & & \\ & & & & b_{70} & b_{71} & & \\ & & & & & & b_{80} & b_{81} \\ & & & & & & b_{90} & b_{91} \end{bmatrix}$$

Aggregation Based Setup

B

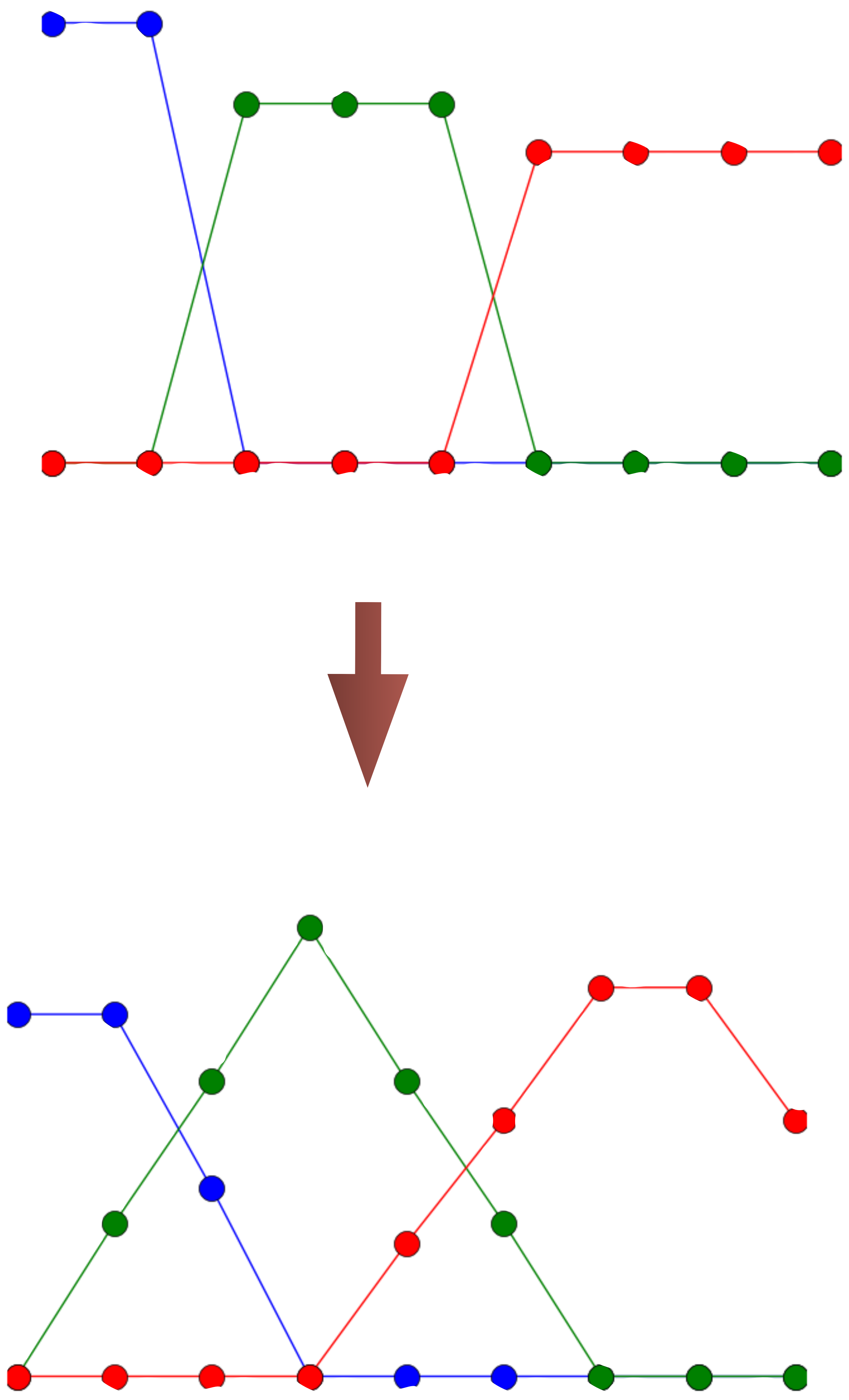
$S \leftarrow \text{strength}(A, B)$

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

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

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

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



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

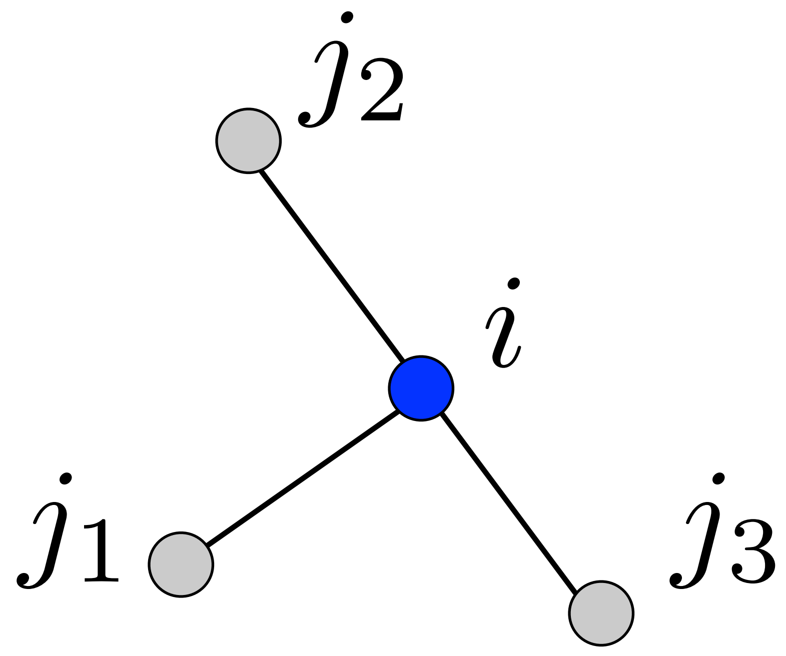
Output: $A_1, \ldots, A_L$,
$P_0, \ldots, P_{L-1}$

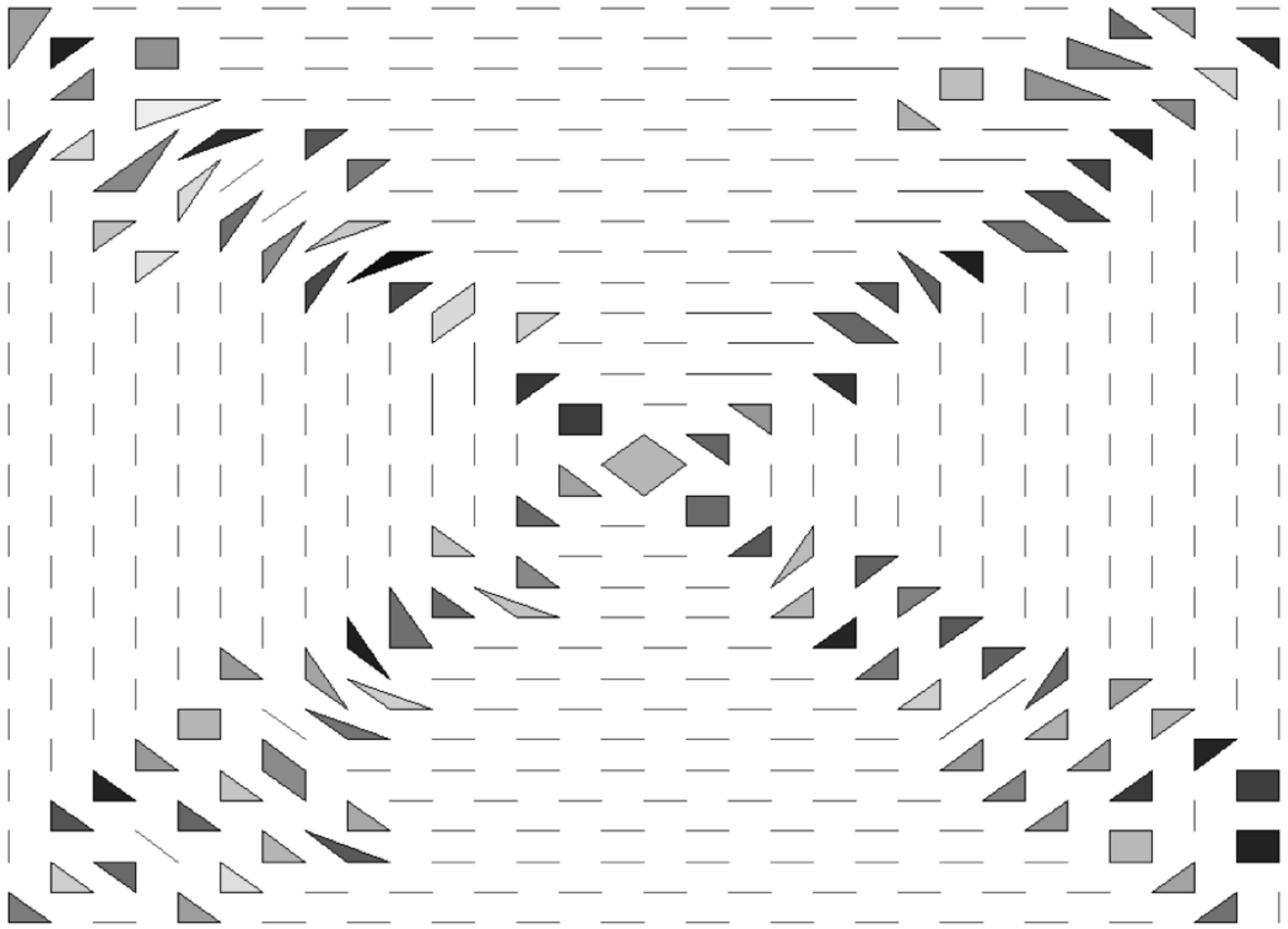
$\ell = 0$
while $\texttt{size}(A_\ell) > \texttt{max\_size}$
  $S_\ell = \texttt{strength}(A_\ell)$ {Strength-of-connection}
  $\mathcal{A}_\ell = \texttt{aggregate}(S_\ell)$ {Aggregation}
  $T_\ell,\ B_{\ell+1} = \texttt{inject}(\mathcal{A}_\ell,\ B_\ell)$ {Form tentative interpolation and coarse candidates}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

Demo: 18-components-of-smoothed-aggregation.ipynb

Advanced feature: evolution measure

1. drop point source at a node
2. evolve point source with A
3. evaluate diffusivity at neighbors in comparison to known low energy



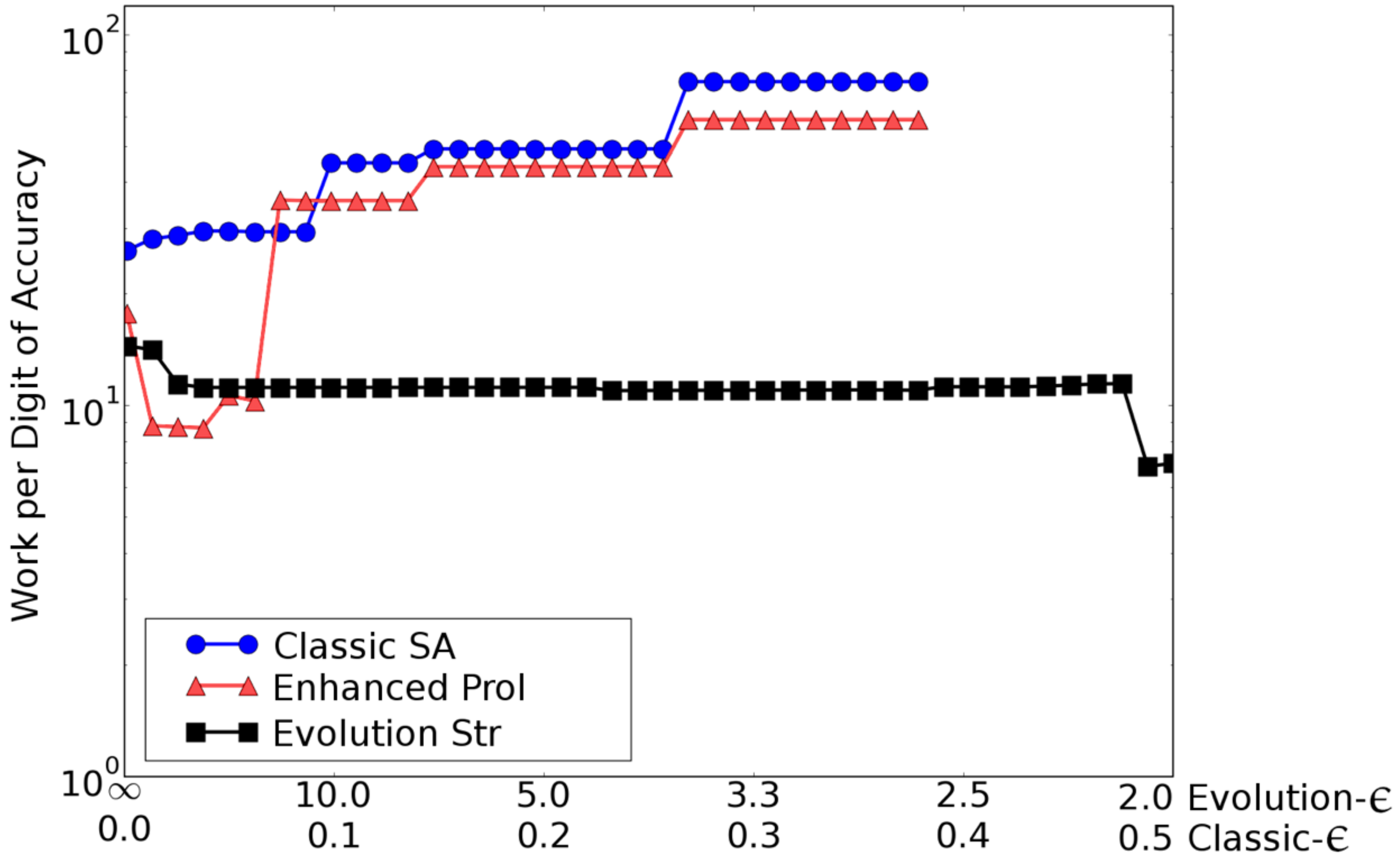


- efficient
- parameter insensitive
- Euler flow
- wave problems
- high-order
- discontinuous elements

Luke N. Olson , Jacob Schroder , Raymond S. Tuminaro, A new perspective on strength measures in algebraic multigrid, NLAA, 2010

Advanced feature: evolution measure

- less sensitive to drop tolerances
- single largest improvement we see in practice
- necessary for any complicated physics or discretization



		Classic SA		with P_{opt}		Evol
ϵ		0	$\frac{1}{4}$	0	$\frac{1}{4}$	4.00
Linear Tets						
$p = 1$	h	18	48	15	27	15
	$h/2$	17	74	14	62	14
	$h/4$	26	130	20	92	17
$p = 2$	h	41	85	24	46	18
Quad Tets						
$p = 1$	h	21	44	26	33	16
	$h/2$	45	88	67	67	14
	$h/4$	63	148	129	129	17
$p = 2$	h	49	86	61	61	17

SA AMG Interpolation

$$e_1 \leftarrow \underbrace{(I - P(P^T AP)^{-1} P^T A)}_{\text{coarse grid correction}} \underbrace{G}_{\text{relax}} e_0$$

$Ge_0 \in \mathcal{R}(P) \quad \Rightarrow \quad e_1 = 0$
interpolation should capture what relaxation misses

- P should have low energy (low A-norm or A^*A-norm)
  1. determine sparsity pattern
  2. minimize energy column-wise (parallel)

SA AMG Interpolation

- Want P so that $u_{low} \in \mathcal{R}(P)$

1. Grow and fix sparsity pattern as $S^k P_{tent}$

2. Minimize residual of

$$AP_j = 0 \quad \text{for each column } j$$

3. Constraint the minimization with

$$PB_c = B$$

SA AMG General Interpolation

- Hermitian (and positive definite): use CG

$$AP_j = 0 \Leftrightarrow \min \|P_j\|_A$$

$$R = P^*$$

- Non-Hermitian: use GMRES

$$AP_j = 0 \Leftrightarrow \min \|P_j\|_{A^*A}$$

$$A^* R_j^* = 0 \Leftrightarrow \min \|R_j^*\|_{AA^*}$$

- Range of interpolation targets “right” low-energy
- Range of restriction* targets “left” low-energy
- Cost is comparable to that of standard smoothing

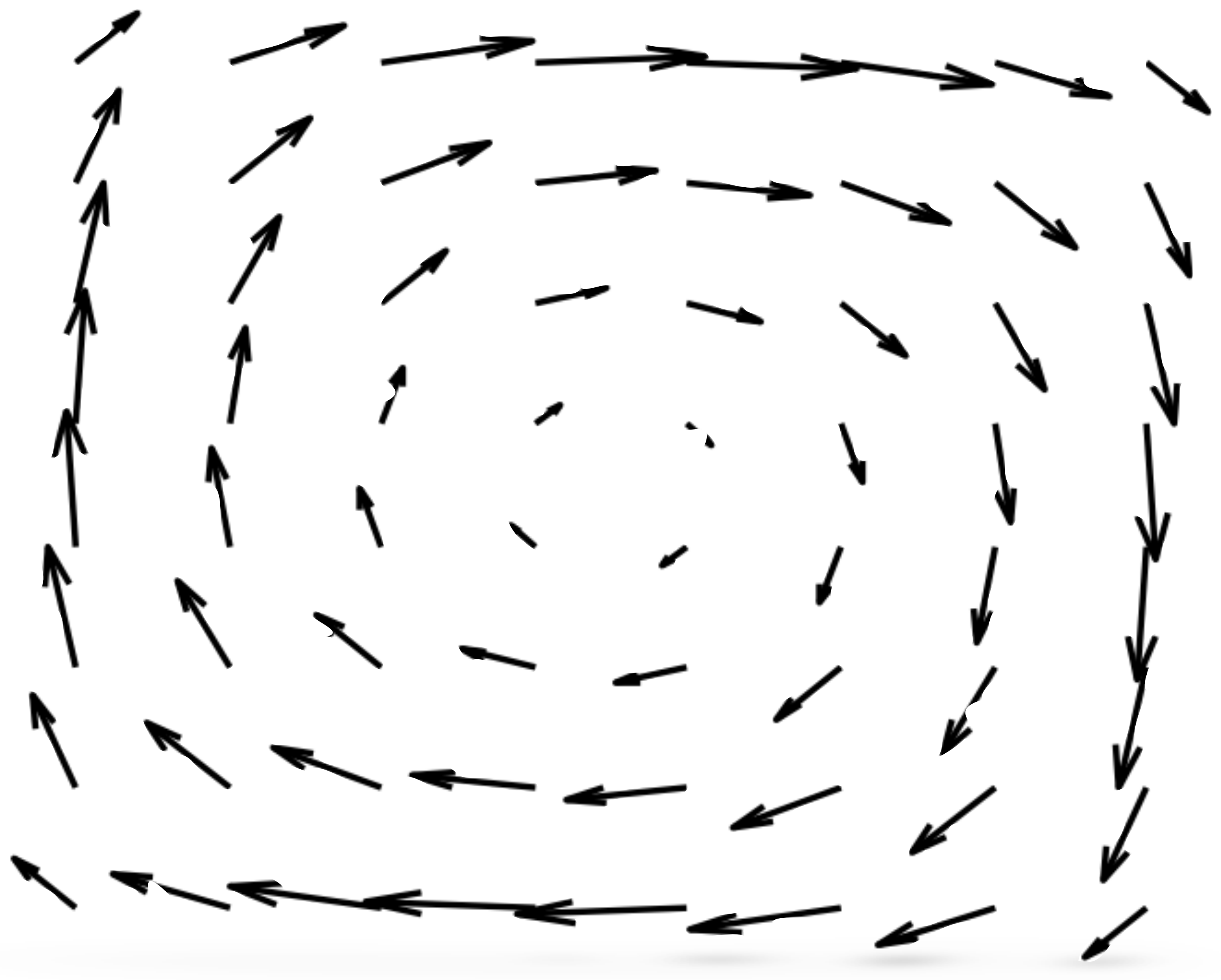
W. L. Wan, T. F. Chan, and B. Smith, An energy-minimizing interpolation for robust multi-grid methods, SIAM J. Sci. Comput., 2000

J. Xu and L. Zikatanov, On an energy minimizing basis for algebraic multigrid methods, Comput. Vis. Sci., 2004

Luke N. Olson , Jacob Schroder , Raymond S. Tuminaro, A General Interpolation Strategy for Algebraic Multigrid Using Energy Minimization, SISC, 2011

SA AMG General Interpolation

- P should have low energy (low A-norm or A^*A-norm)
  1. determine sparsity pattern
  2. minimize energy column-wise (parallel)



h	std.	opt.
1/64	>150	24
1/128	>150	28
1/256	>150	33
1/512	>150	33

Advanced Demos

Demo: `19-AMG-advanced-options-anisotropy.ipynb`

Demo: `20-AMG-advanced-options-nonsymmetric-flow.ipynb`

Demo: `21-AMG-advanced-options-systems-elasticity.ipynb`