Parallel sparse linear algebra (1)

Prof. Richard Vuduc

Georgia Institute of Technology

CSE/CS 8803 PNA, Spring 2008

[L.09] Tuesday, February 5, 2008

Sources for today's material

- Mike Heath (UIUC)
- CS 267 (Yelick & Demmel, UCB)
- John Gilbert (UCSB)
- Xiaoye (Sherry) Li (LBNL)
- *Direct methods for sparse matrices*, by Duff, Erisman, and Reid

Review:
Parallel dense linear algebra

Equivalent formulations of matrix-matrix multiplication

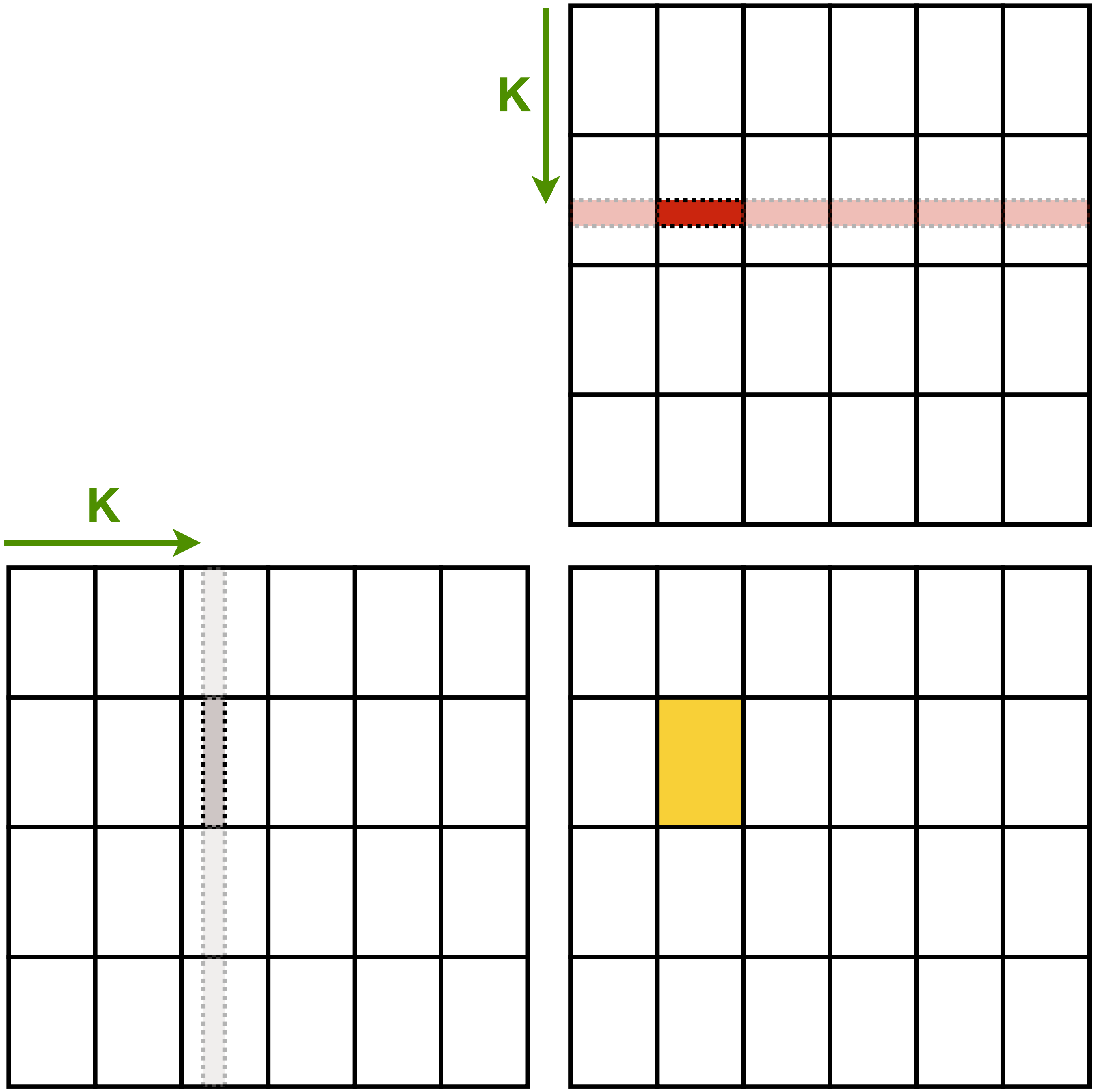
Dot products

$$C = A \cdot B = \begin{pmatrix} \hat{a}_1^T \\ \vdots \\ \hat{a}_m^T \end{pmatrix} \cdot \begin{pmatrix} b_1 & \cdots & b_n \end{pmatrix} = \begin{pmatrix} \hat{a}_1^T \cdot b_1 & \cdots & \hat{a}_1^T \cdot b_n \\ \vdots & & \vdots \\ \hat{a}_m^T \cdot b_1 & \cdots & \hat{a}_m^T \cdot b_n \end{pmatrix}$$

$$\Longrightarrow c_{ij} = \hat{a}_i^T \cdot b_j$$

Outer products

$$C = A \cdot B = \begin{pmatrix} a_1 \cdots a_k \end{pmatrix} \cdot \begin{pmatrix} \hat{b}_1^T \\ \vdots \\ \hat{b}_k^T \end{pmatrix} = a_1 \hat{b}_1^T + \cdots + a_k \hat{b}_k^T = \sum_k a_k \hat{b}_k^T$$

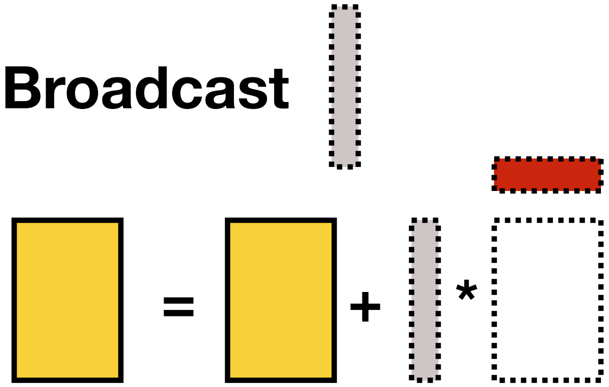


SUMMA:

for-all **K**:

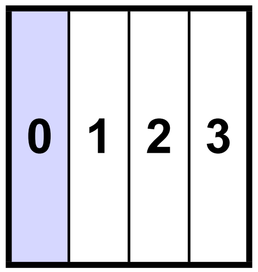
Broadcast

Broadcast



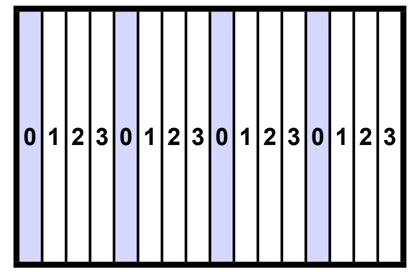
Possible data layouts for dense LU

1-D column blocked



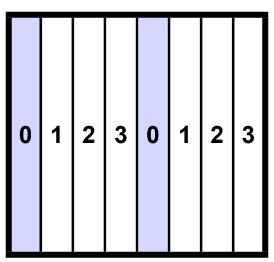
Bad load balance

1-D column cyclic



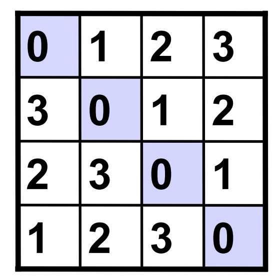
Can't easily use BLAS2/3

1-D column block cyclic



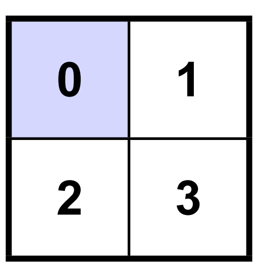
Factorization bottleneck

Block skewed



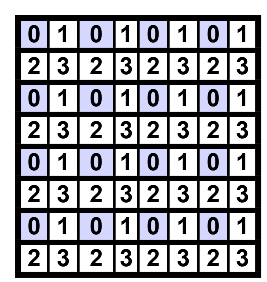
Complicated addressing

2-D row & col blocked

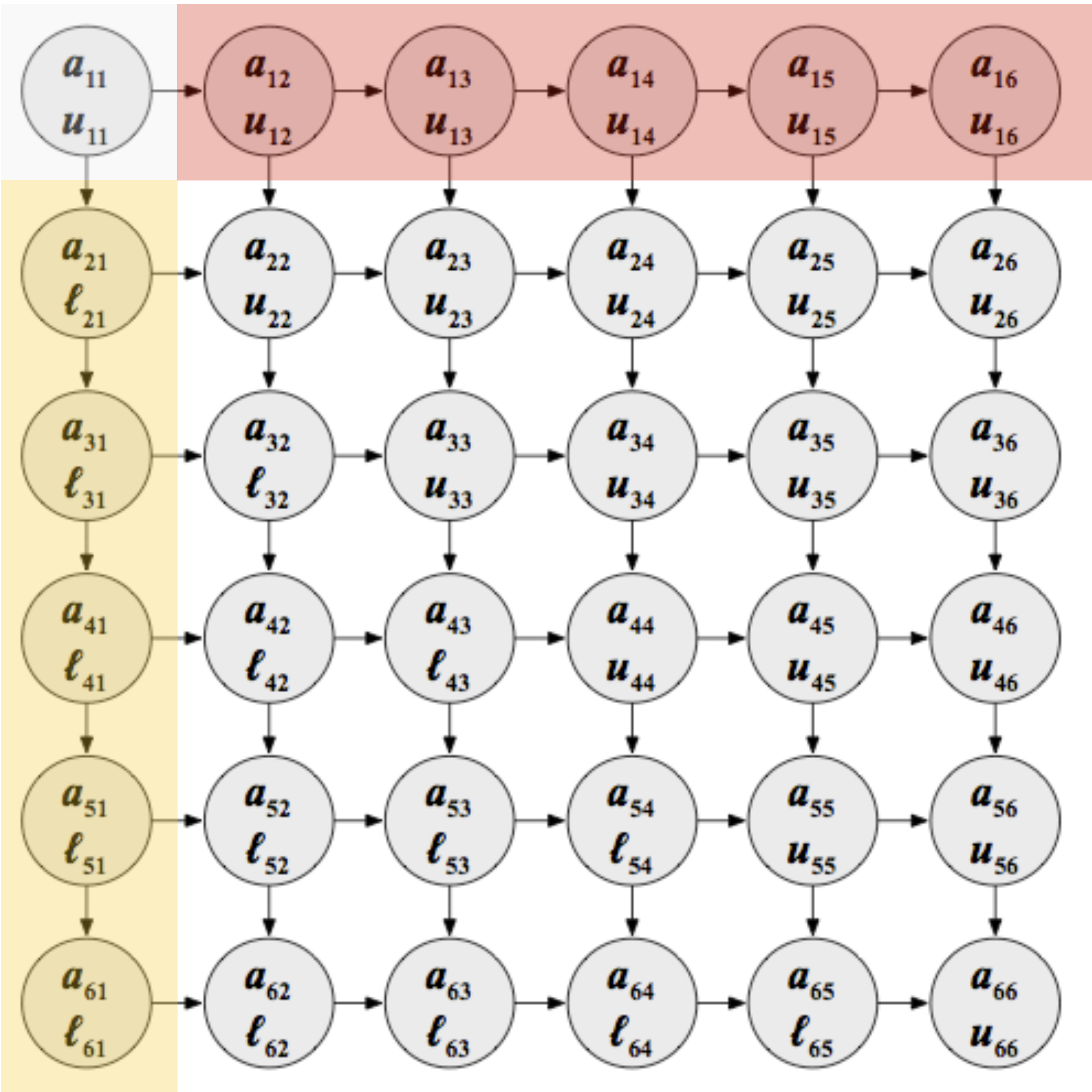


Bad load balance

2-D row & col block cyclic



Best option



$$
\begin{aligned}
l_{11} &= 1 \\
u_{11} &= a_{11} \\
\hat{u}_{12}^T &= \hat{a}_{12}^T \\
l_{21} &= \frac{a_{21}}{a_{11}} \\
L_{22}U_{22} &= A_{22} - l_{21}\hat{u}_{12}^T
\end{aligned}
$$

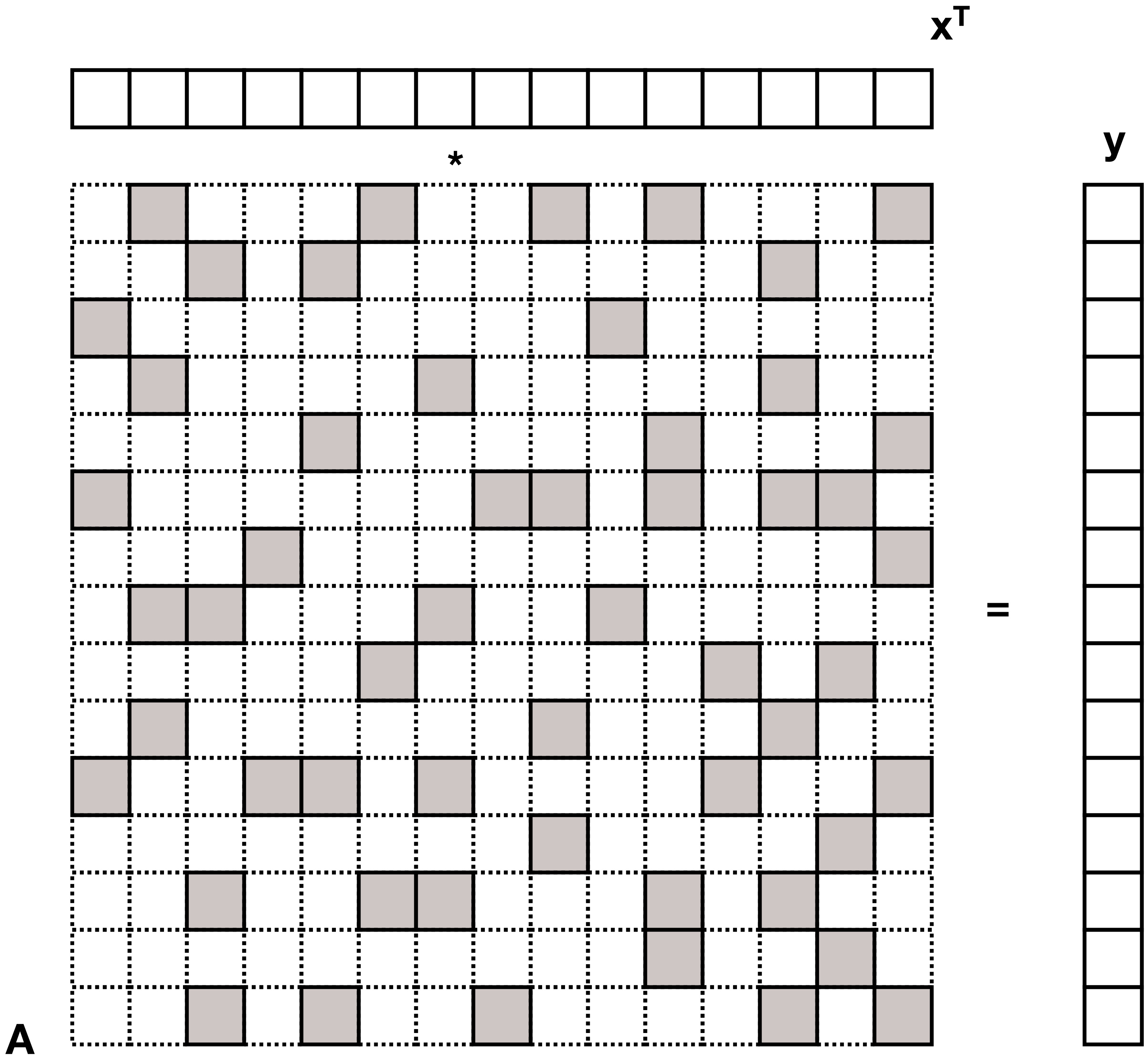
Partial pivoting to improve stability

$$A = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

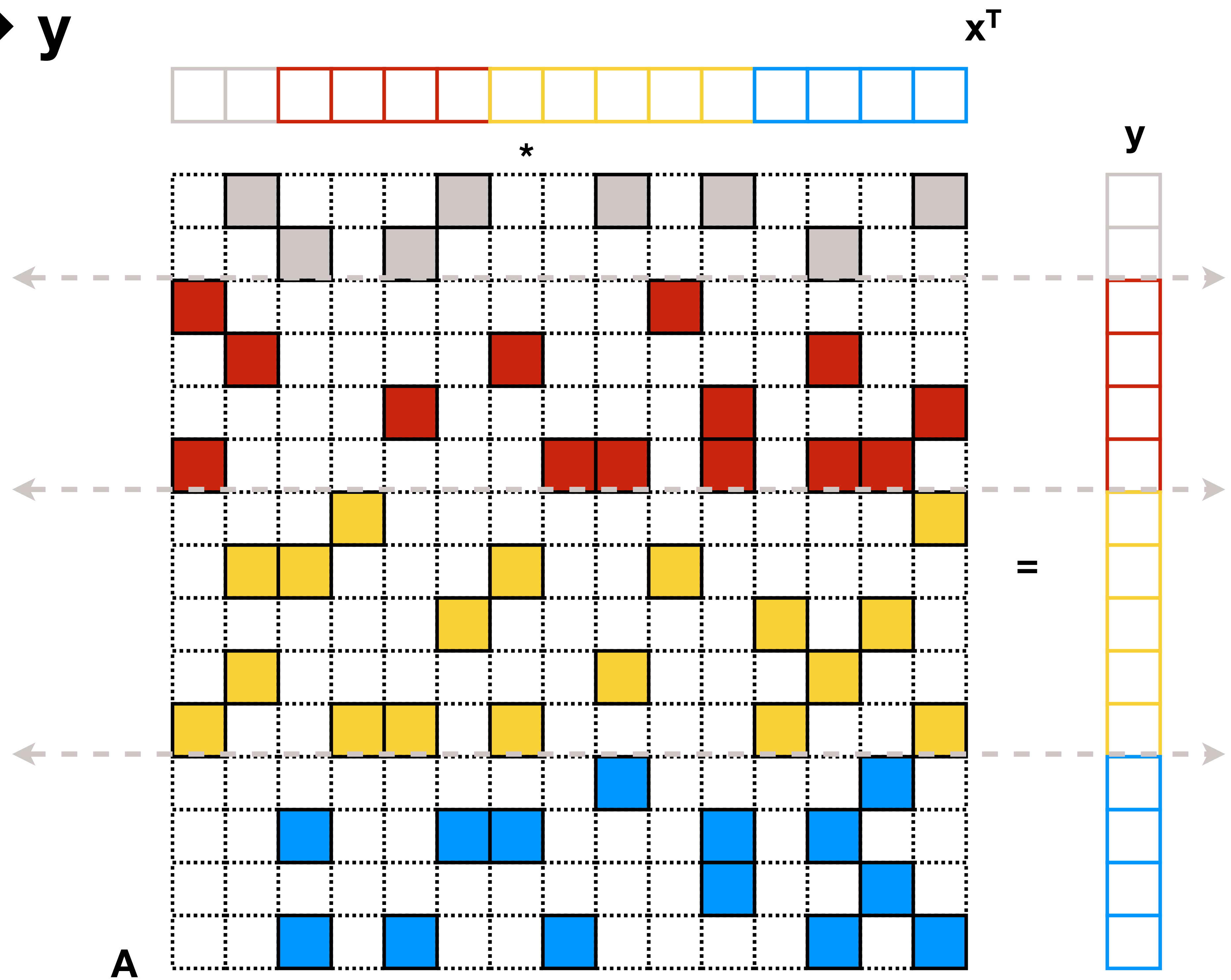
- Observation: GE as described fails on above matrix (divide by 0)
- **Partial pivoting**: At each elimination step, swap "top" row with row that has leading entry with largest magnitude

Sparse matrix-vector multiply

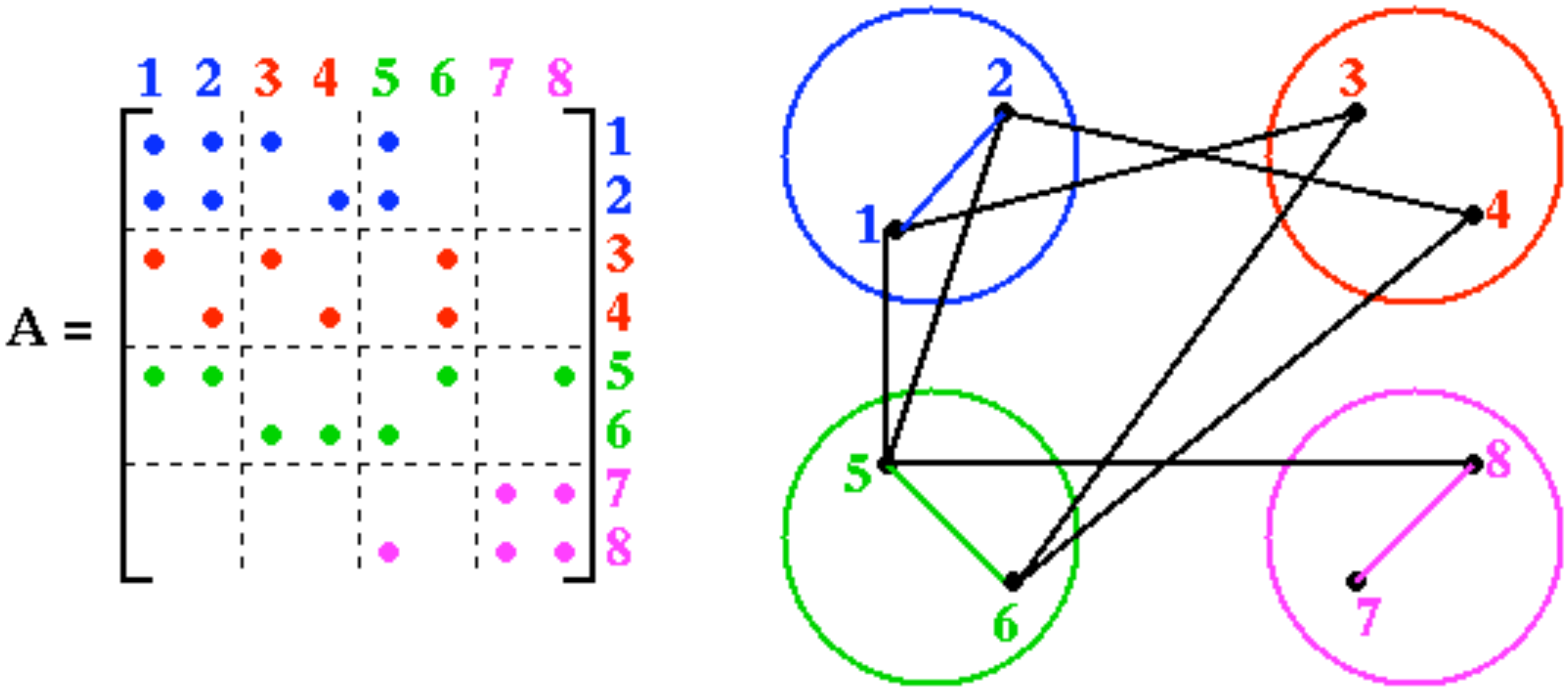
A*x → y



A*x → y



Partitioning a Sparse Symmetric Matrix



Parallelization options

- 1-D row vs. 1-D column vs. 2-D layouts?
    - Broadcasts vs. reductions
- Ideal matrix structure?
- Symmetry?

- Improving communication complexity
    - Distribution/mapping: Formulate as graph **partitioning** problem
    - Reduce communication volume: Row/column **reordering**

Sparse Gaussian elimination

Algorithms for 2-D (3-D) Poisson, $N=n^2$ ($=n^3$)

Algorithm	Serial	PRAM	Memory	# procs
Dense LU	**N^3**	**N**	**N^2**	**N^2**
Band LU	N^2 ($N^{7/3}$)	N	$N^{3/2}$ ($N^{5/3}$)	N ($N^{4/3}$)
Jacobi	**N^2 ($N^{5/3}$)**	**N ($N^{2/3}$)**	**N**	**N**
Explicit inverse	N^2	log N	N^2	N^2
Conj. grad.	**$N^{3/2}$ ($N^{4/3}$)**	**$N^{1/2(1/3)}$ log N**	**N**	**N**
RB SOR	**$N^{3/2}$ ($N^{4/3}$)**	**$N^{1/2}$ ($N^{1/3}$)**	**N**	**N**
Sparse LU	**$N^{3/2}$ (N^2)**	**$N^{1/2}$**	**N log N ($N^{4/3}$)**	**N**
FFT	N log N	log N	N	N
Multigrid	N	$\log^2$ N	N	N
Lower bound	**N**	**log N**	**N**	

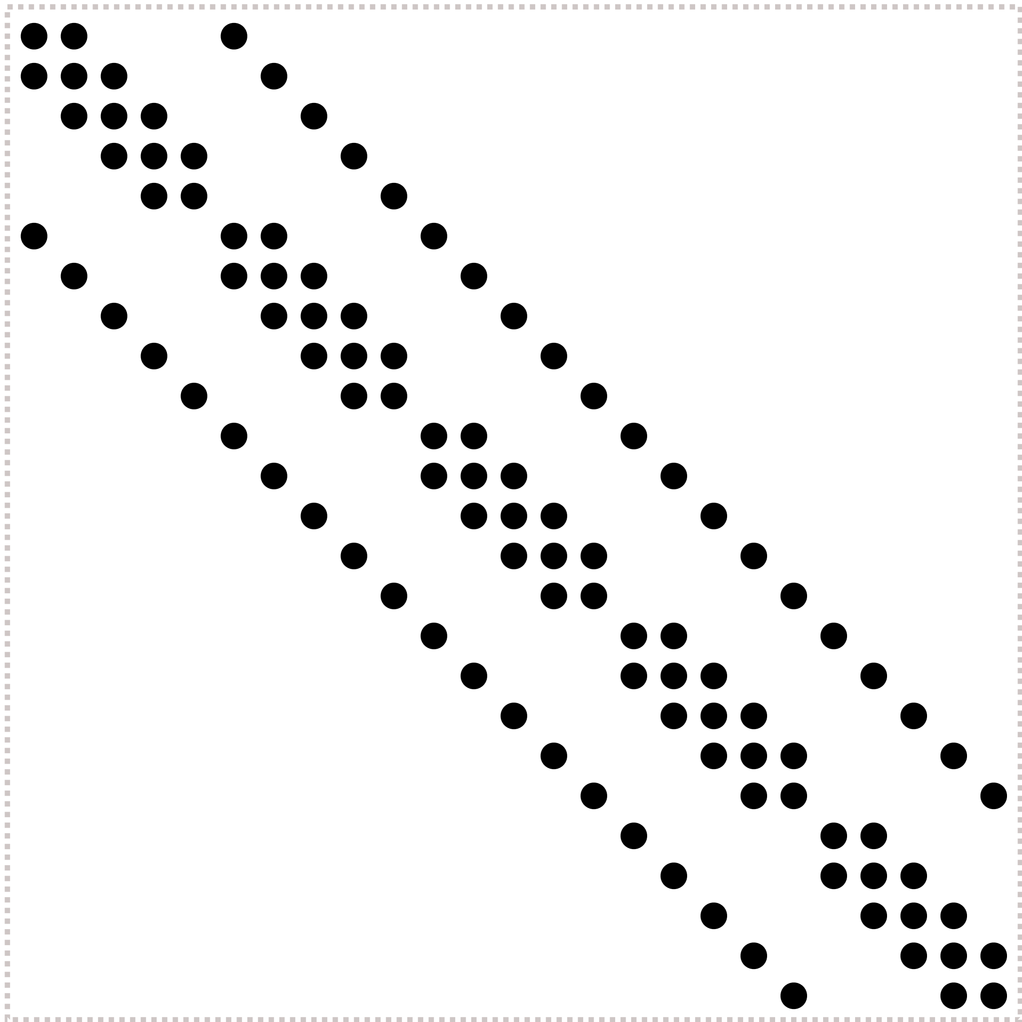
PRAM = idealized parallel model with zero communication cost.
Source: Demmel (1997)

Recall: One step of GE

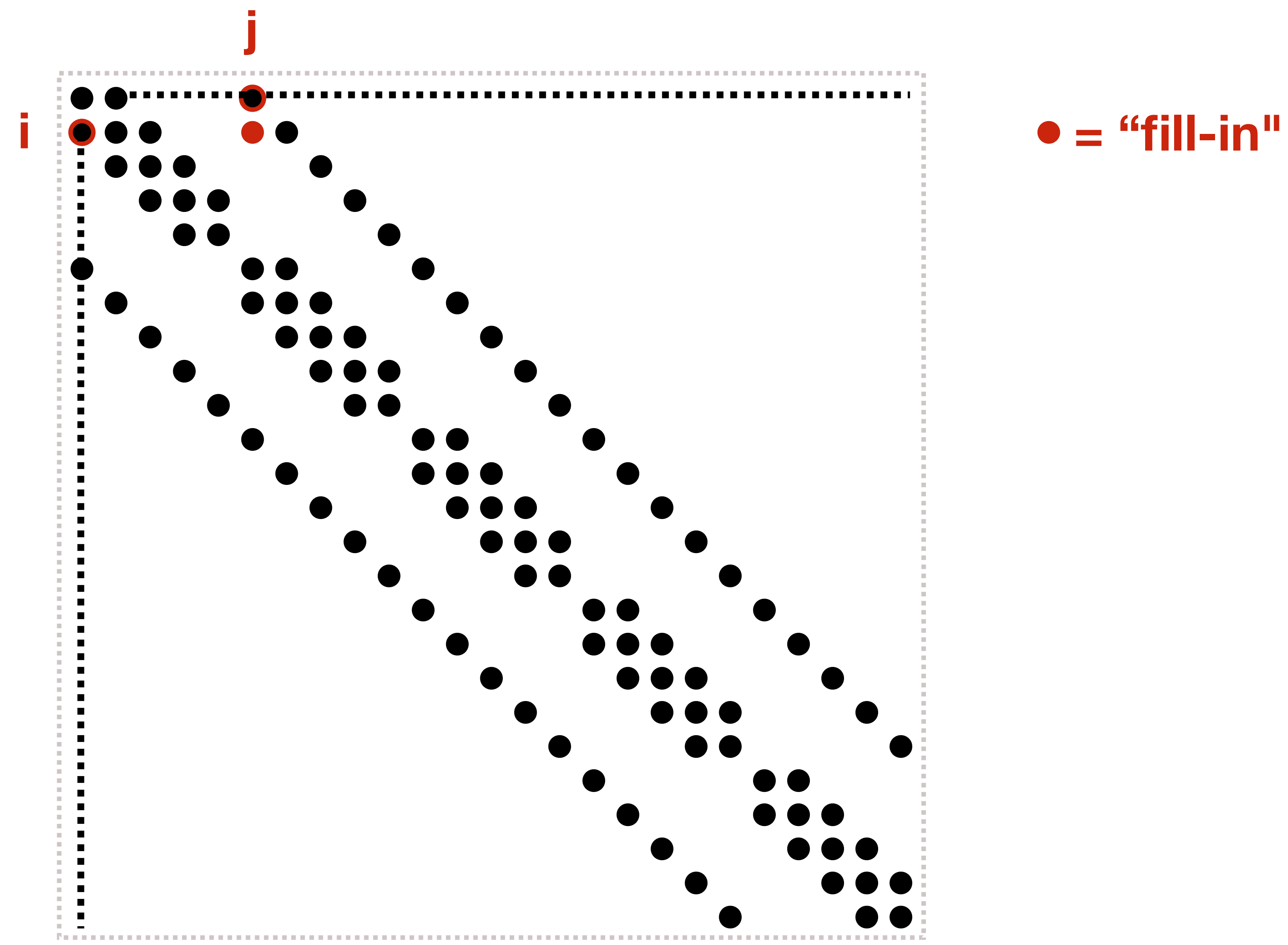
$$A = \begin{pmatrix} \alpha & w^T \\ v & B \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ \frac{v}{\alpha} & I \end{pmatrix} \begin{pmatrix} \alpha & w^T \\ 0 & B - \frac{1}{\alpha} v \cdot w^T \end{pmatrix}$$

- Question: What if A is **sparse**?
  - http://crd.lbl.gov/~xiaoye/SuperLU/SparseDirectSurvey.pdf

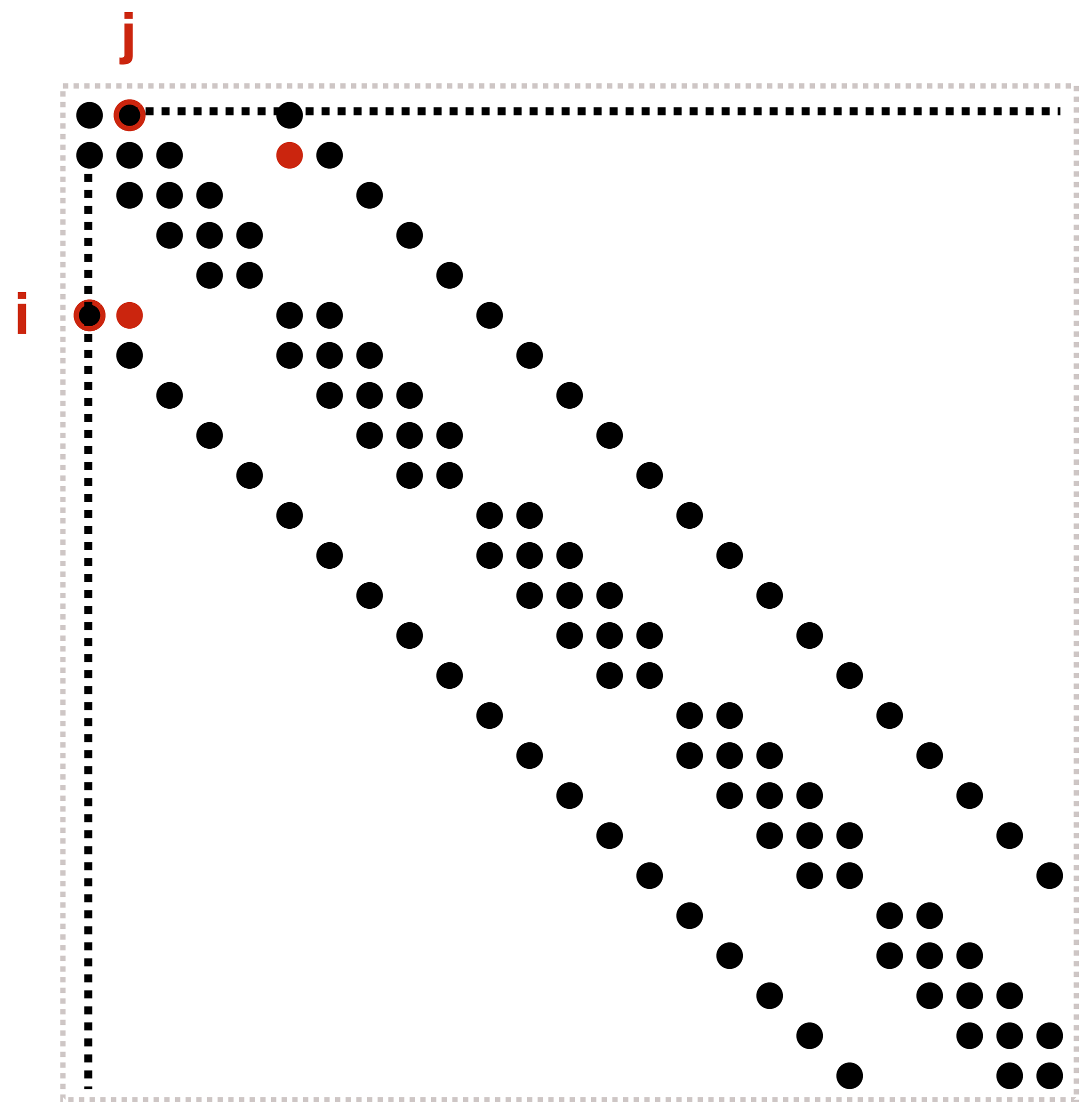
$$a_{ij} = a_{ij} - a_{ik} \cdot a_{kj}$$



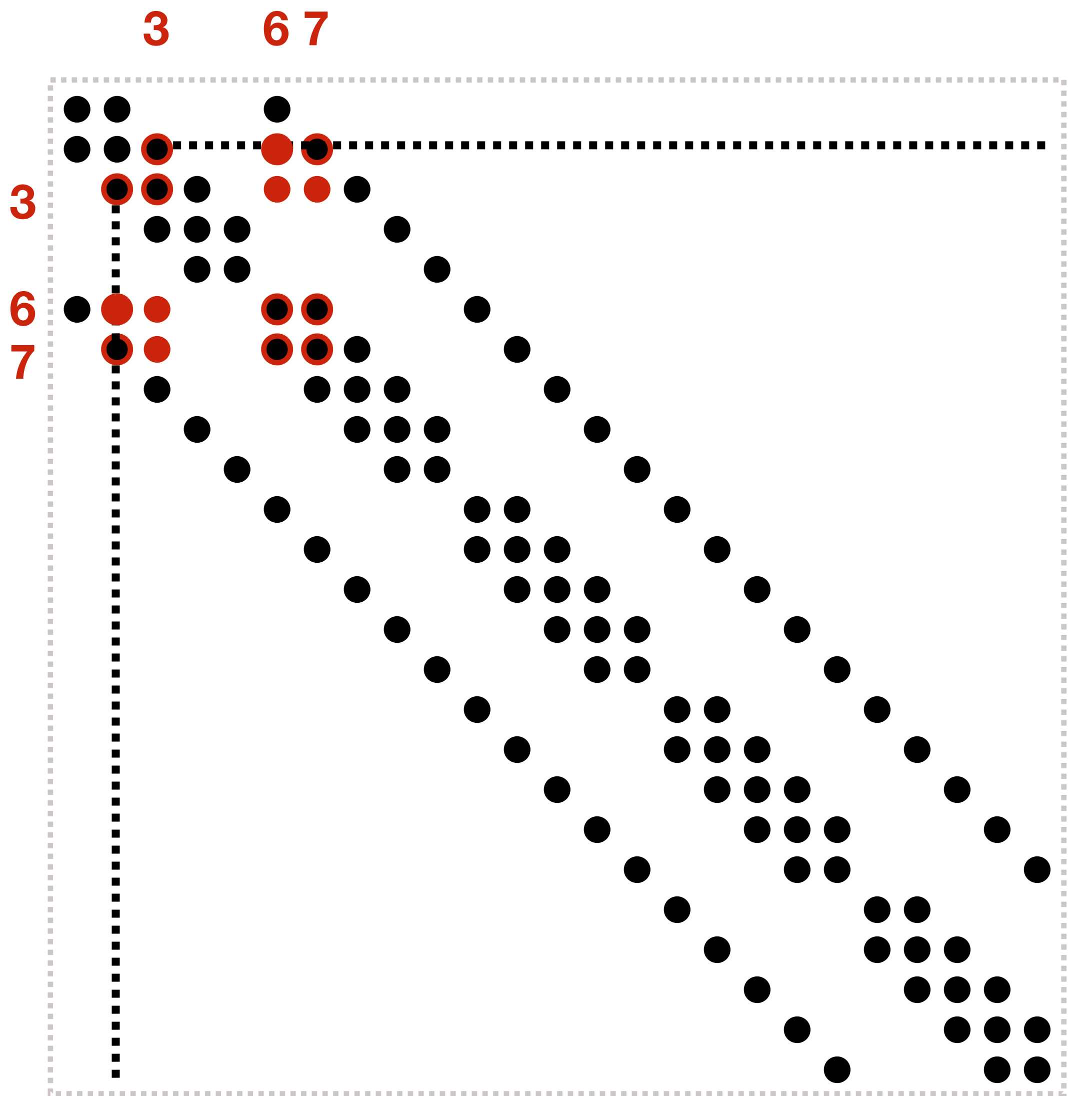
$$a_{ij} = a_{ij} - a_{ik} \cdot a_{kj}$$



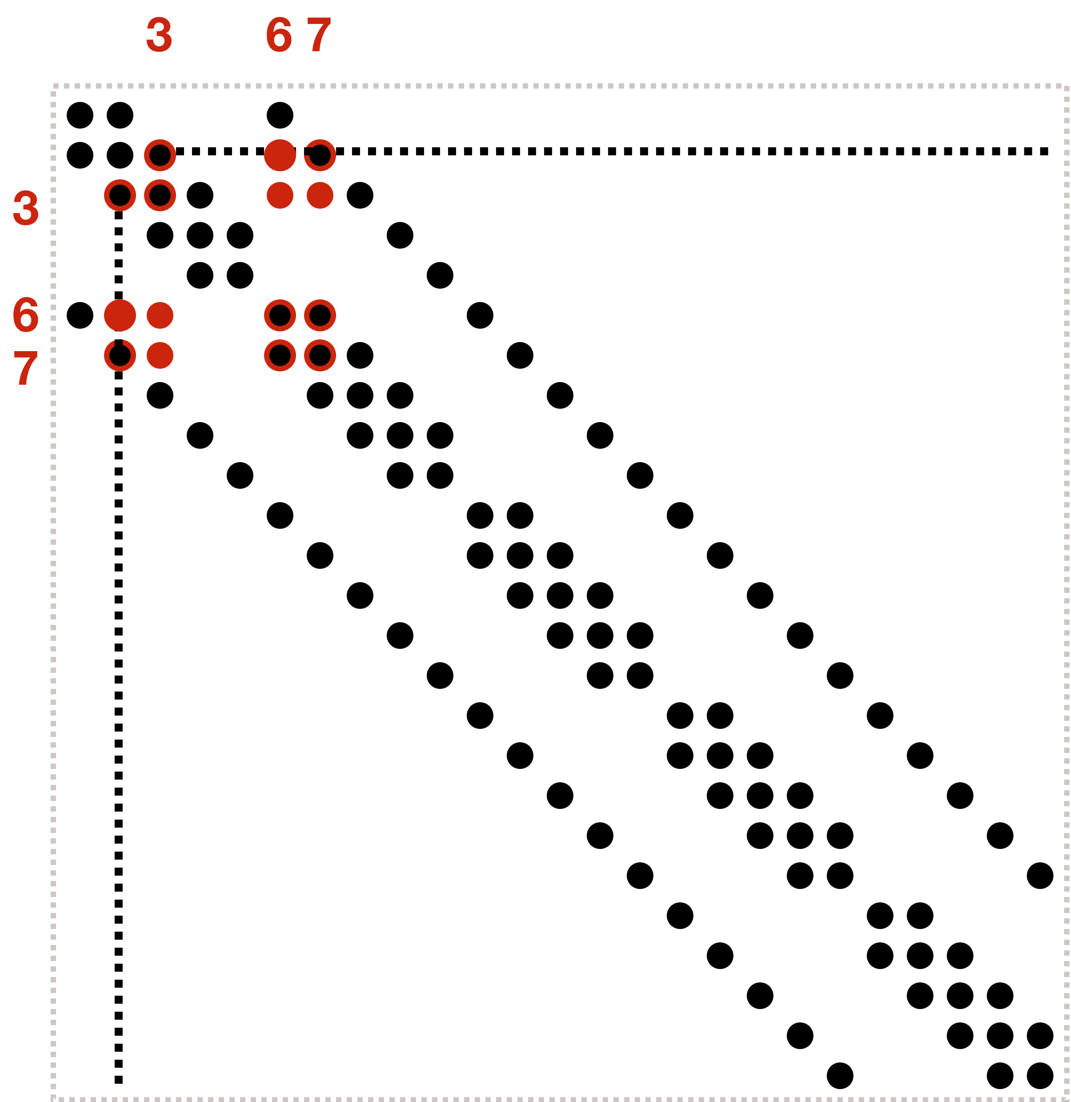
$$a_{ij} = a_{ij} - a_{ik} \cdot a_{kj}$$

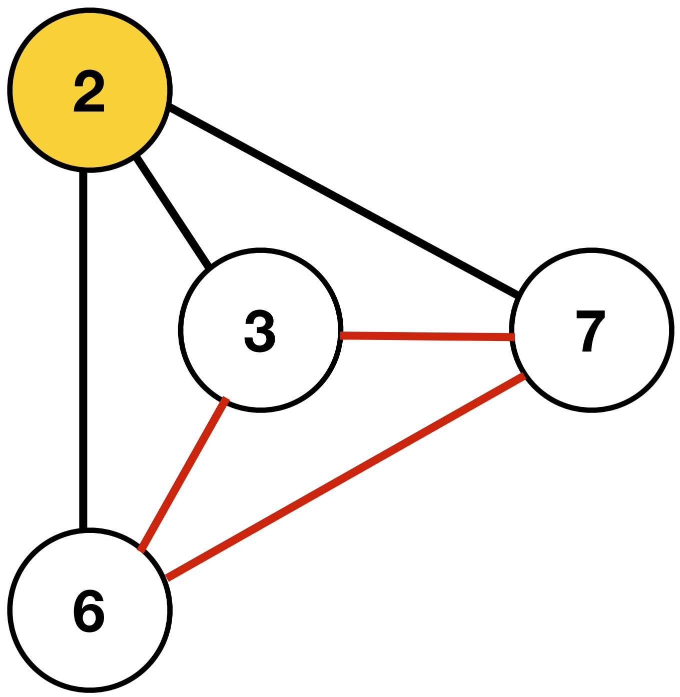


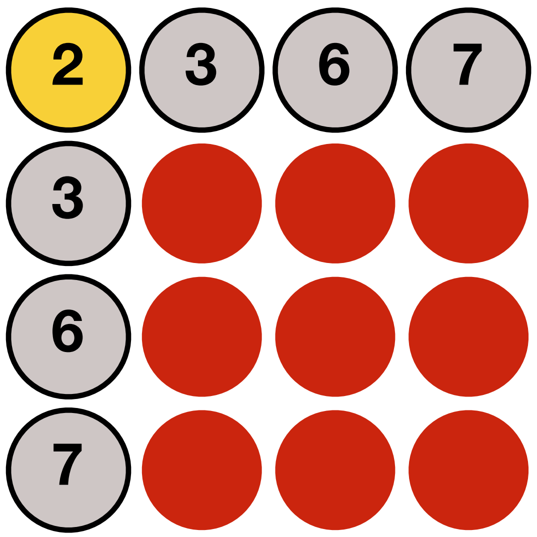
$$a_{ij} = a_{ij} - a_{ik} \cdot a_{kj}$$



$$a_{ij} = a_{ij} - a_{ik} \cdot a_{kj}$$

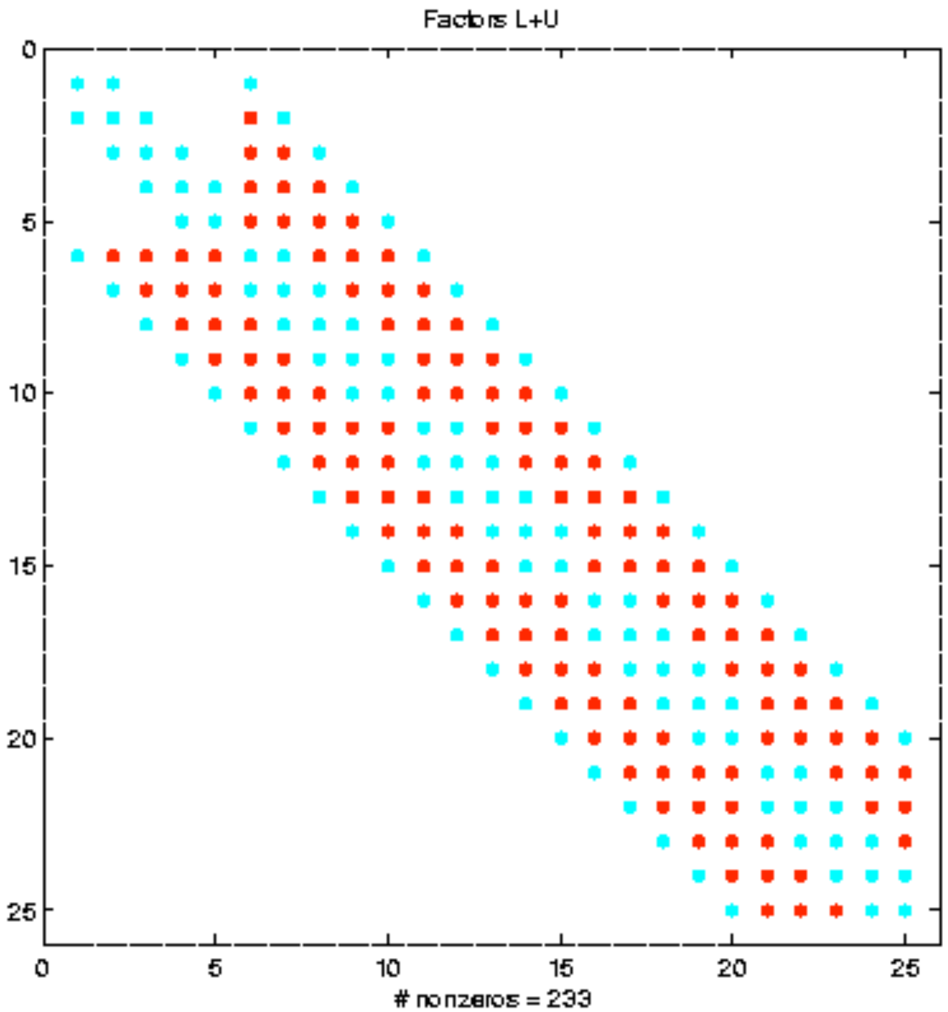




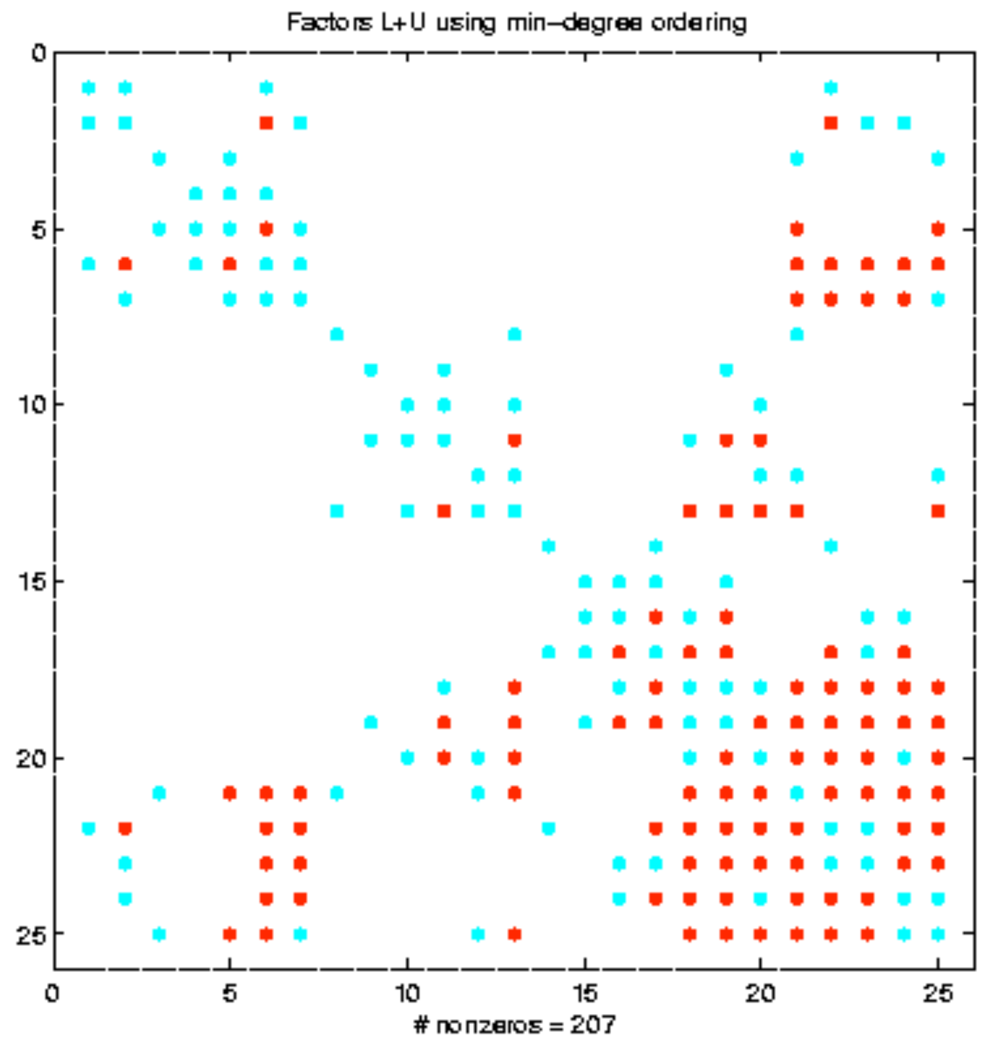


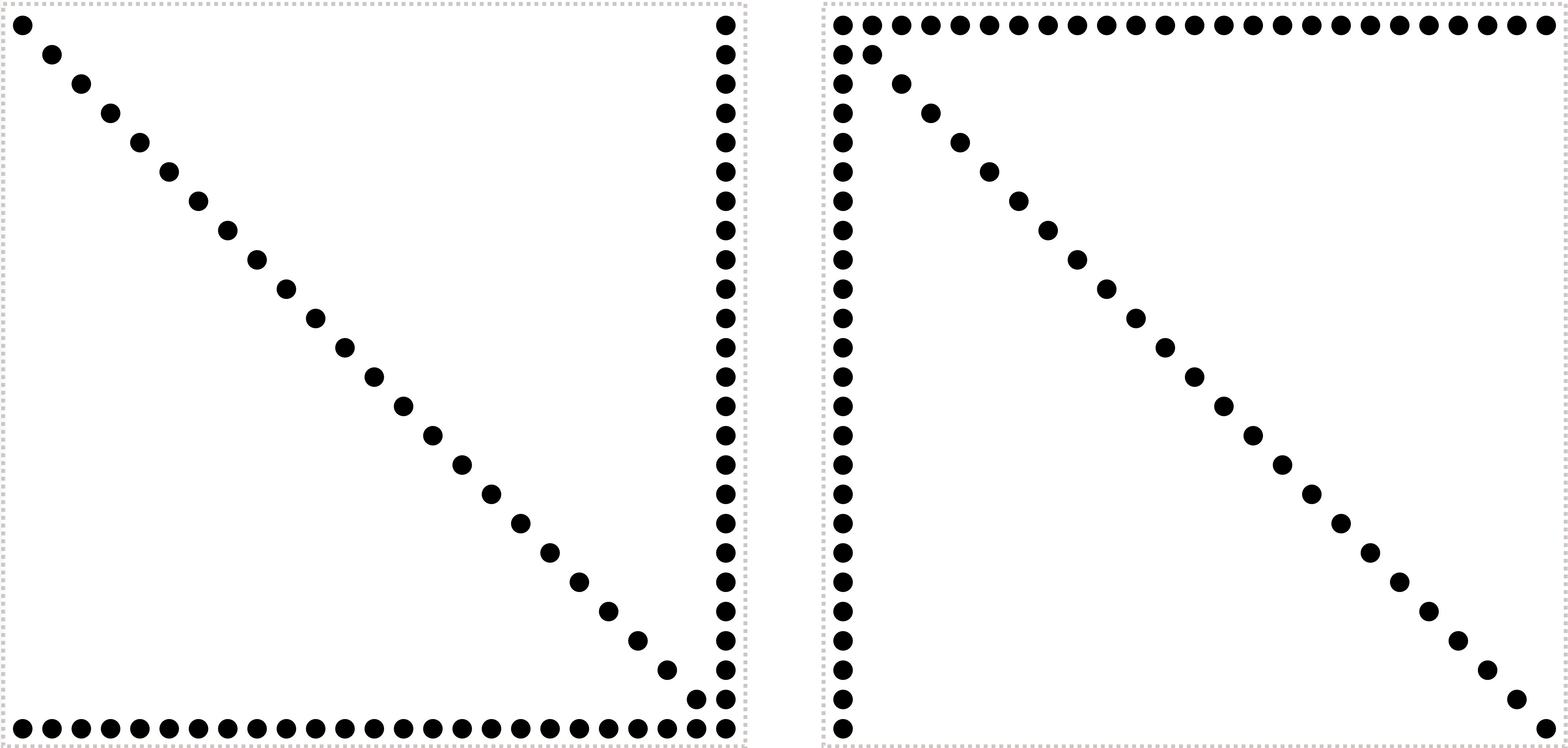
$$a_{ij} = a_{ij} - a_{ik} \cdot a_{kj}$$

Total nnz = 233



Total nnz = 207





Dense vs. sparse LU

$$P_r A P_c = LU$$ **"Complete" pivoting**

- Dense
  - Choose P_r & P_c to **maintain stability**
  - Partial pivoting (P_r only) suffices in most cases
- Sparse
  - Permute to maintain stability and **preserve sparsity**
  - Dynamic pivoting causes dynamic structure updates

Algorithmic issues

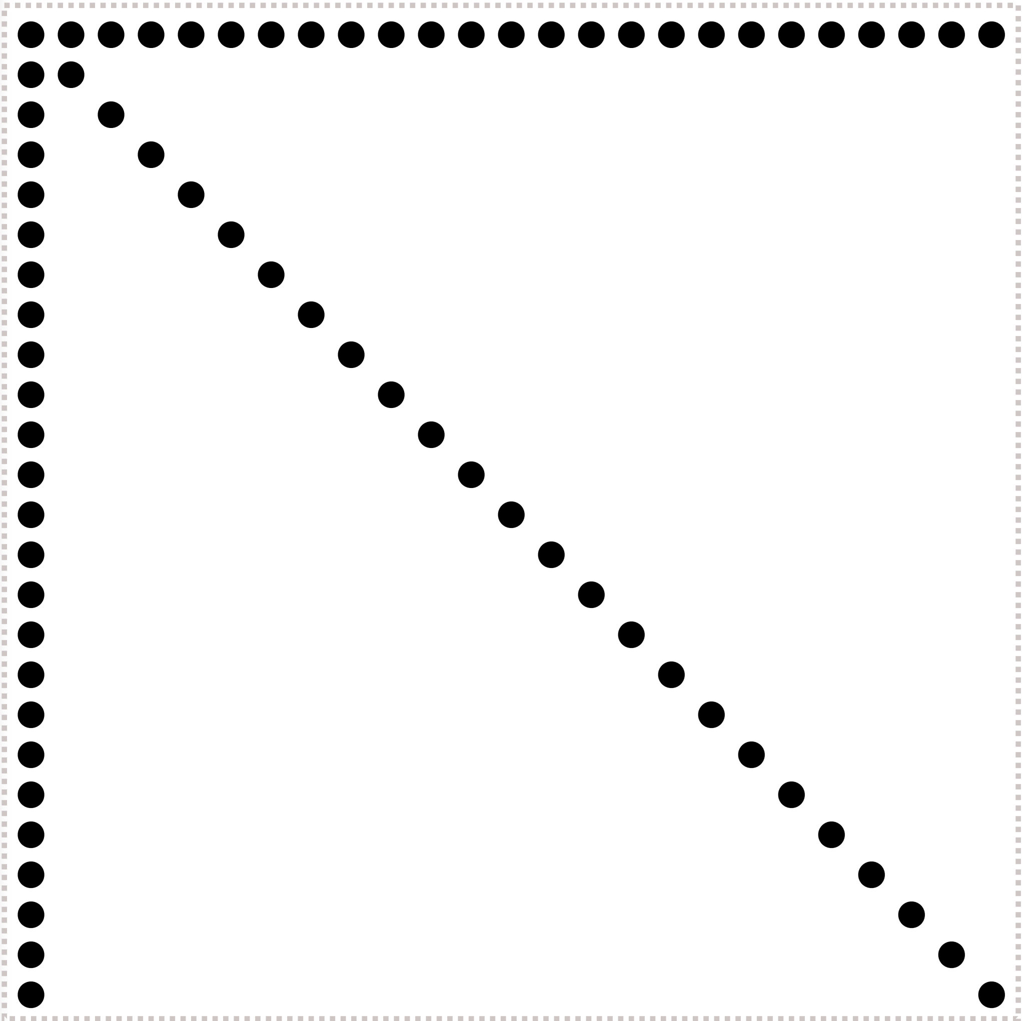
- **Ordering** to minimize fill-in
  - *NP-complete* [Rose & Tarjan, '78; Yannakis '83 for symmetric case]
- **Predict** fill-in: Symbolic factorization step
- **Perform factorization** and triangular solves
  - Numerical factorization step
  - Usually dominates total run-time
- Oh yeah, and do all this stuff in parallel

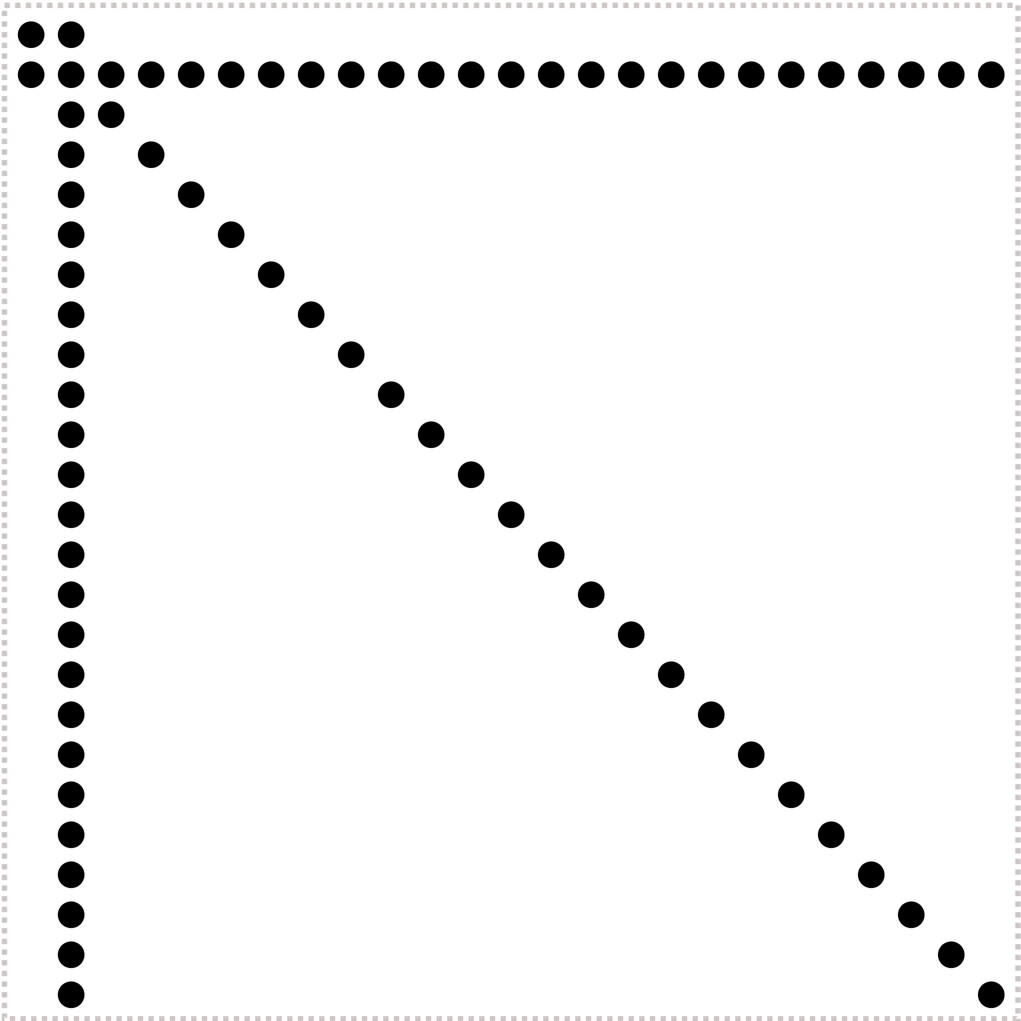
Ordering: Markowitz criterion ('57)

- At elimination stage k:
  - Let $r_i^{(k)}$ = nnz(row i), $c_j^{(k)}$ = nnz(col j), in uneliminated submatrix
  - Choose pivot entry a_{ij} (*i.e.*, swap row & col) that minimizes:

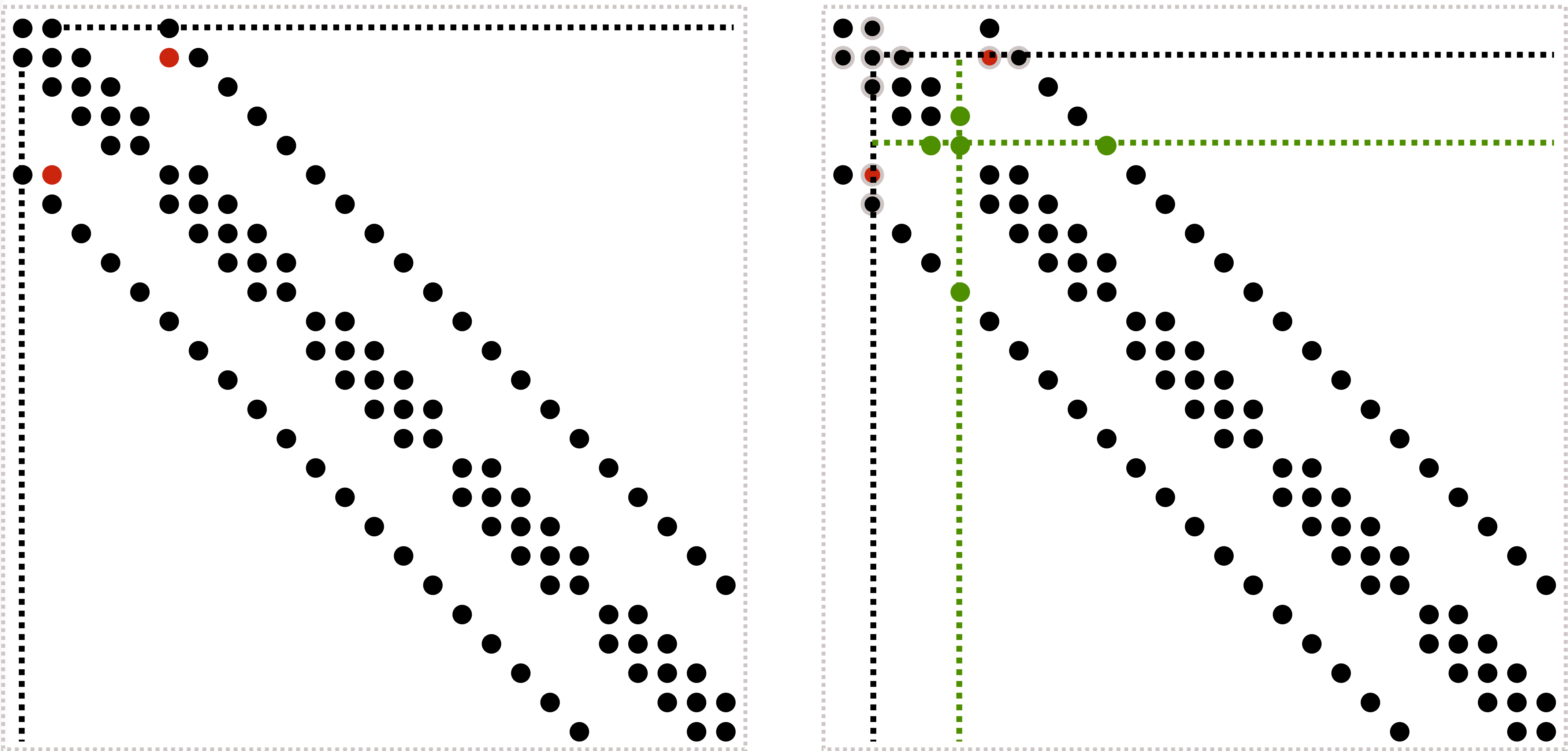
$$(r_i^{(k)} - 1) \times (c_j^{(k)} - 1)$$

- Don't forget about numerical values, too! (*E.g.*, threshold)

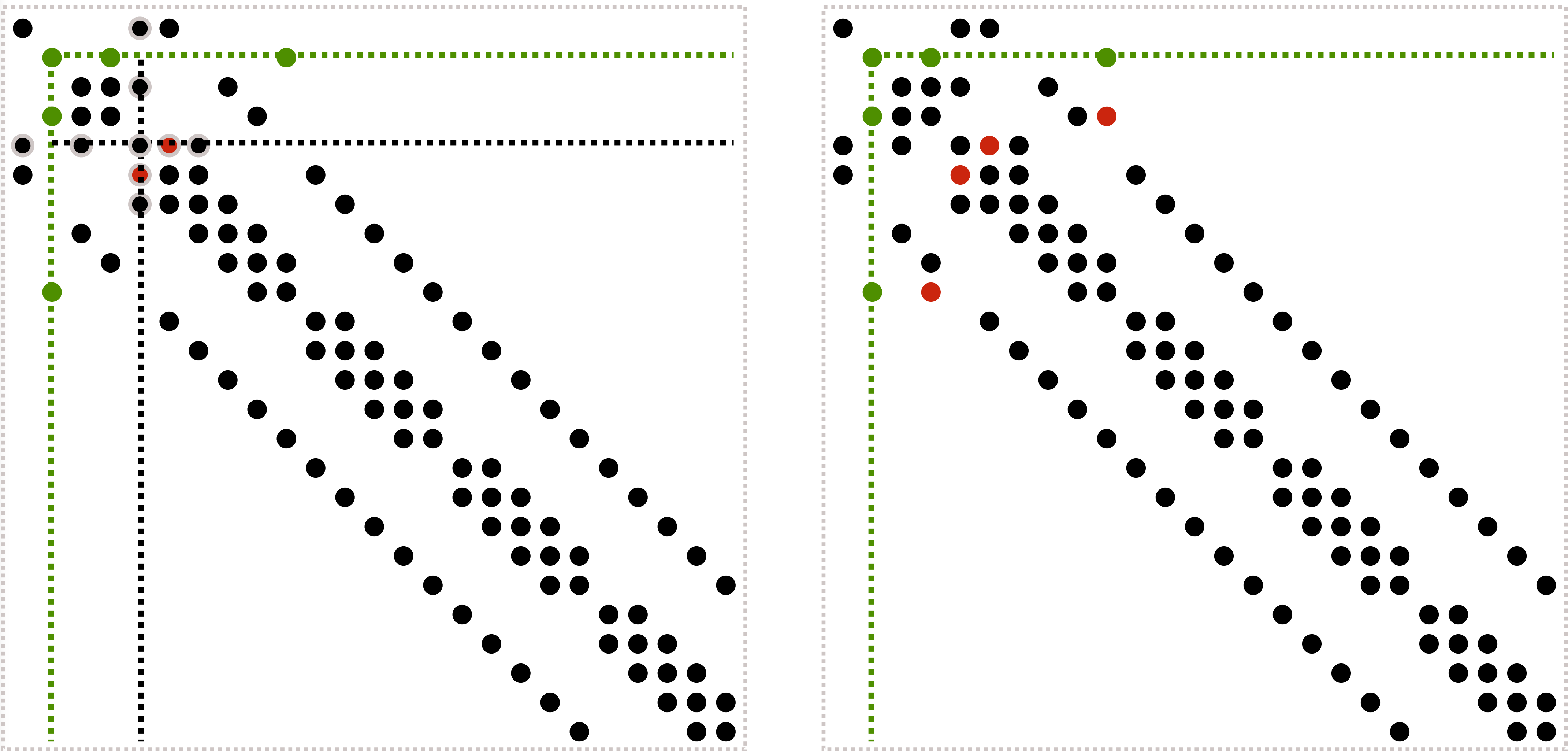




$$a_{ij} = a_{ij} - a_{ik} \cdot a_{kj}$$



$$a_{ij} = a_{ij} - a_{ik} \cdot a_{kj}$$



Markowitz criterion

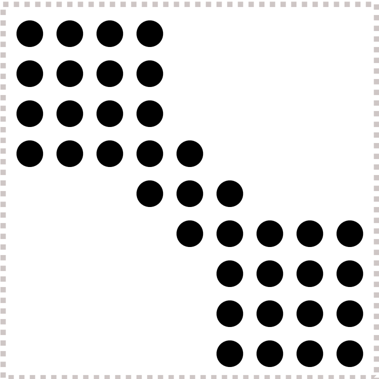
- Generally effective, but has high cost (scan entire uneliminated submatrix)

Minimum degree orderings (Tinney & Walker, ’67)

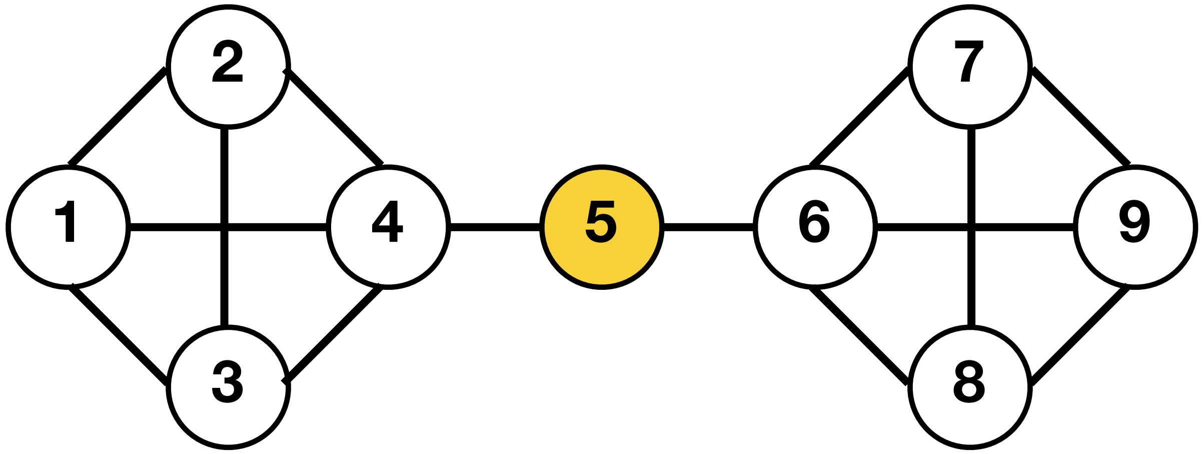
- Suppose A is symmetric positive definite
  - Symmetric: Only need to look at $r_i^{(k)}$
  - Positive definite: Don’t need to pivot

- Greedy algorithm: At each elimination, pick node with fewest edges
- For unsymmetric matrix, apply to pattern(A^TA) or pattern($A+A^T$)

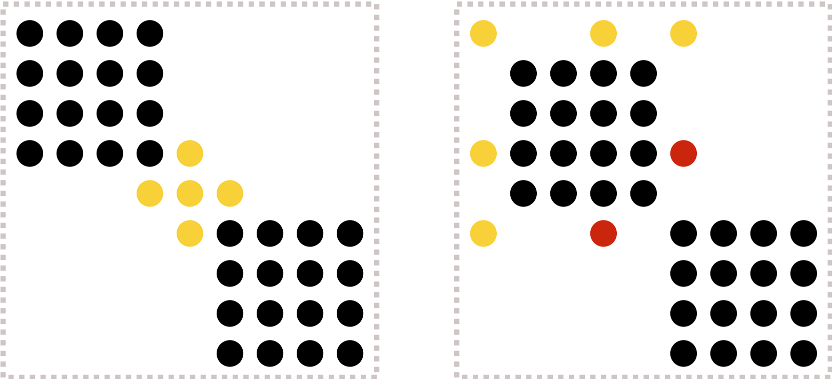
No fill in natural ordering:



Min-degree node: 5

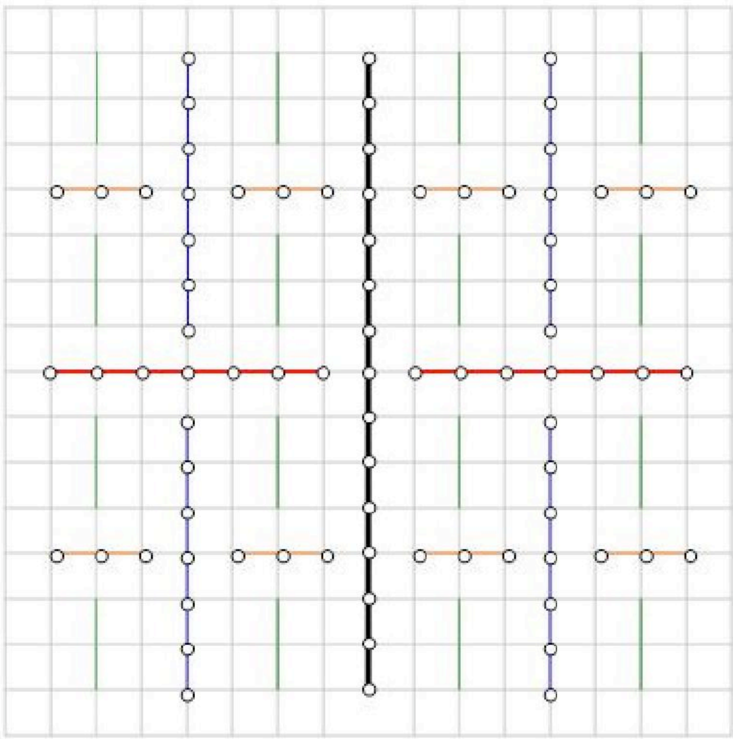


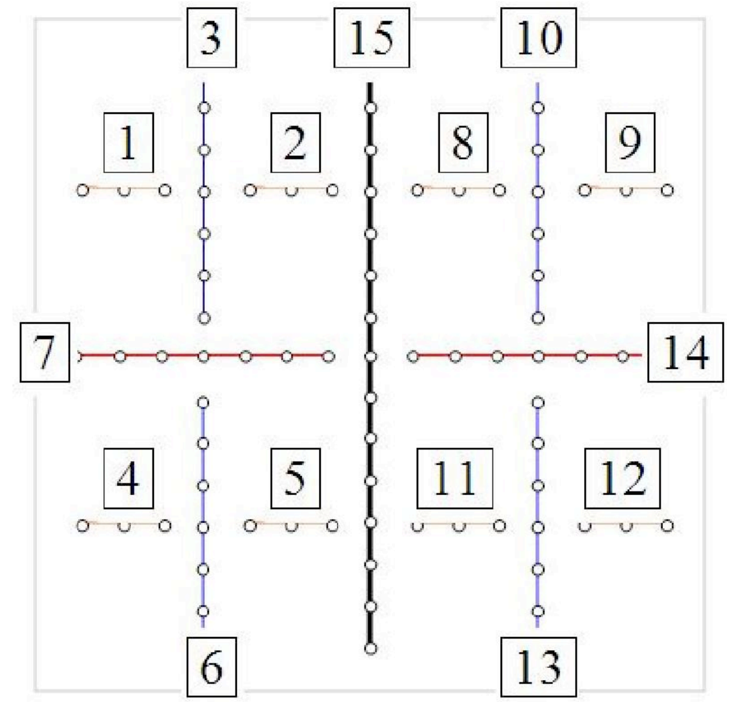
Choosing 5 leads to fill-in:

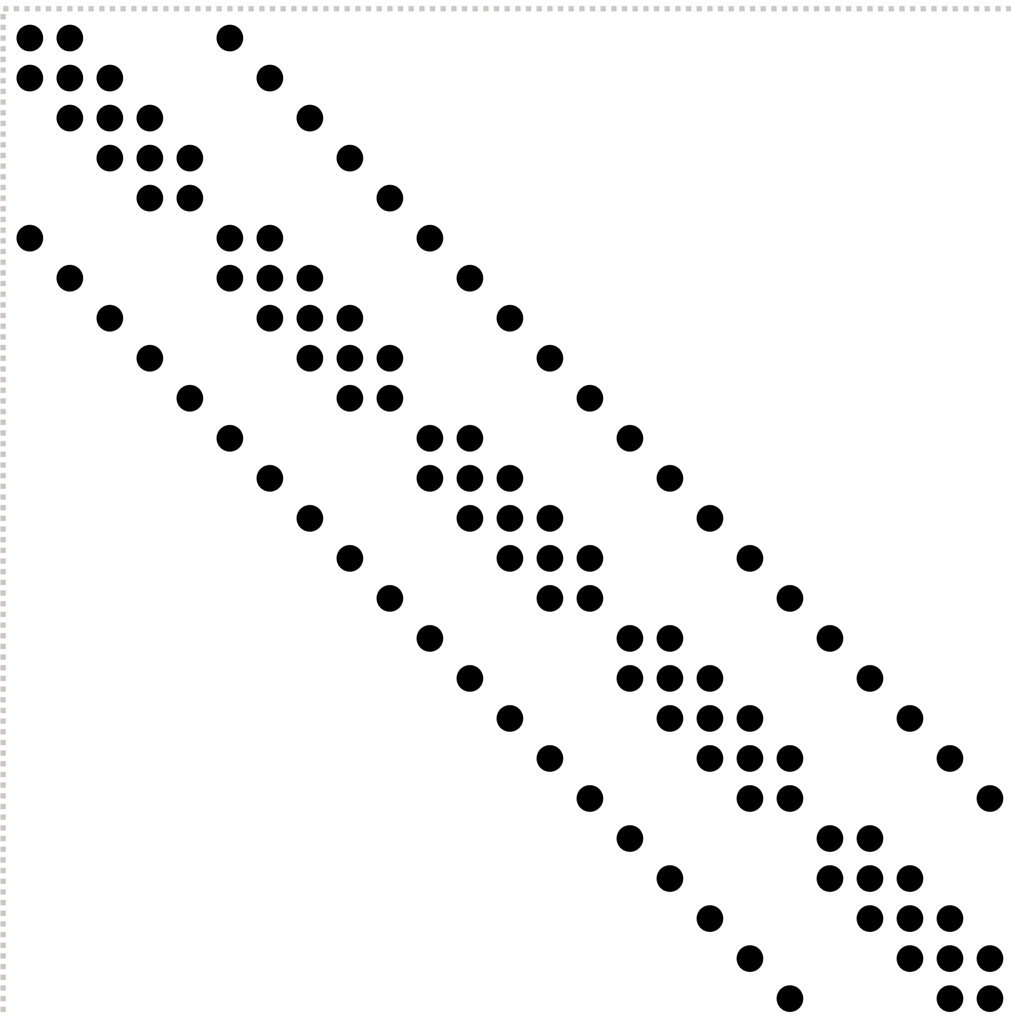


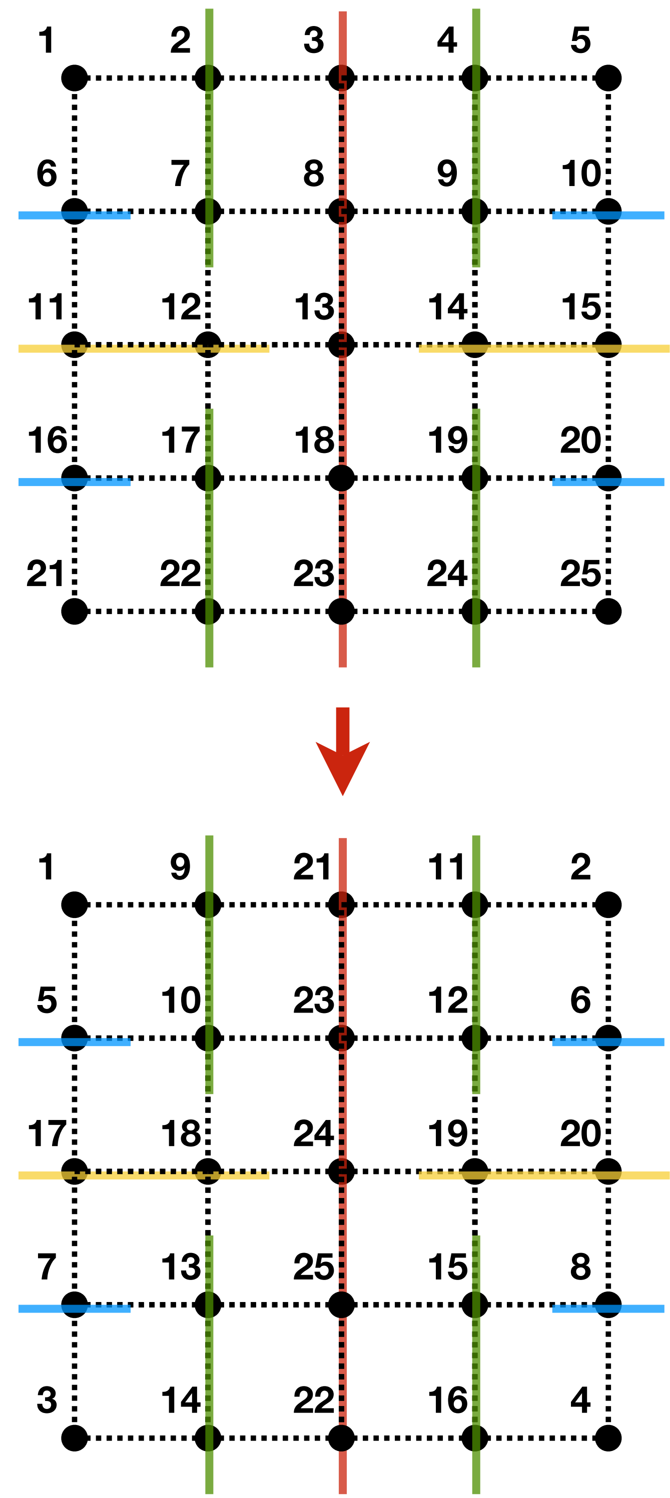
Nested dissection orderings (George ’73)

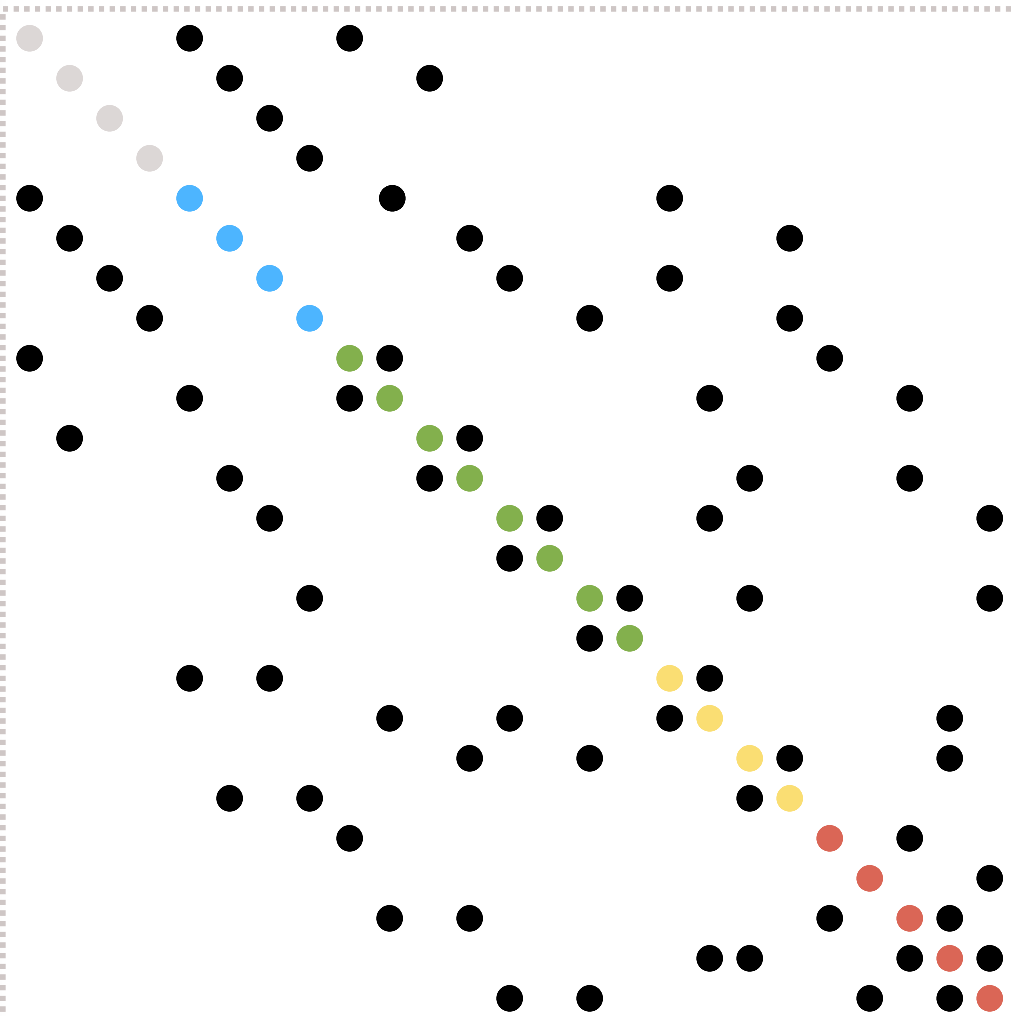
- Originally introduced for the model problem (*e.g.*, 5-point stencil for Poisson)
  - Find a **separator** … **number** it *last* … **recurse**

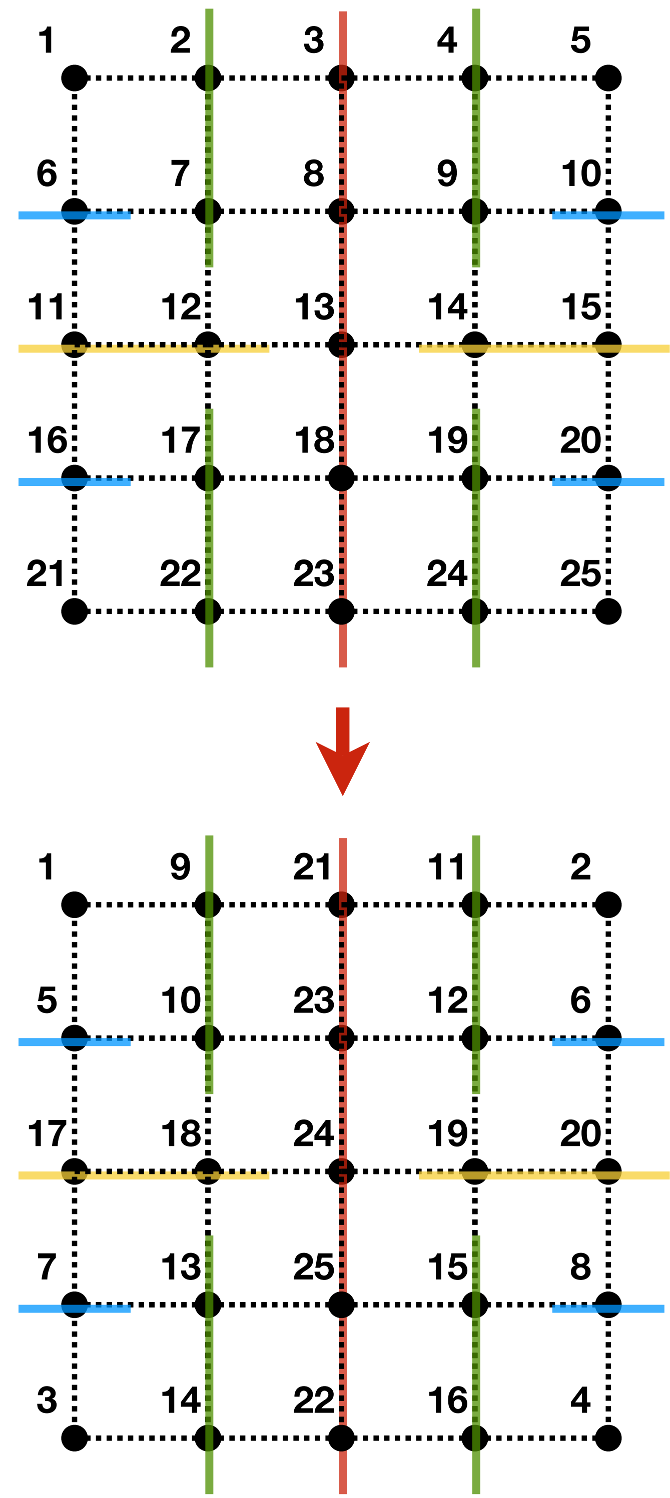







1 2 3 4 5
6 7 8 9 10
11 12 13 14 15
16 17 18 19 20
21 22 23 24 25
1 9 21 11 2
5 10 23 12 6
17 18 24 19 20
7 13 25 15 8
3 14 22 16 4




1 2 3 4 5
6 7 8 9 10
11 12 13 14 15
16 17 18 19 20
21 22 23 24 25
1 9 21 11 2
5 10 23 12 6
17 18 24 19 20
7 13 25 15 8
3 14 22 16 4

Nested dissection orderings

- Given an n x n grid graph, fill-in is $O(n^2 \log n)$ & flops are $O(n^3)$ [George '73]

Finding good separators

- Multilevel schemes
  - Chaco [Hendrickson & Leland '94]
  - Metis [Karypis & Kumar '95]
- Spectral bisection [Simon, *et al*. '90+]
- Geometric and spectral bisection [Chan, Gilbert, & Teng '94]

Administrivia

Administrative stuff

- **New room** (dumpier, but cozier?): College of Computing Building **(CCB) 101**
- **Accounts**: Apparently, you already have them
- Front-end login node: **ccil.cc.gatech.edu** (CoC Unix account)
  - We “own” **warp43—warp56**
  - Some docs (**MPI**): http://www-static.cc.gatech.edu/projects/ihpcl/mpi.html
  - **Sign-up** for mailing list: https://mailman.cc.gatech.edu/mailman/listinfo/ihpc-lab

Homework 1: Parallel conjugate gradients

- Implement a parallel solver for Ax = b (serial C version provided)
  - Evaluate on three matrices: 27-pt stencil, and two application matrices
  - “Simplified:” No preconditioning
  - **Bonus**: Reorder, precondition
- Performance models to understand scalability of your implementation
  - Make measurements
  - Build predictive models
- Collaboration encouraged: Compare programming models or platforms

“In conclusion…”

Backup slides

Landscape of Ax=b solvers

	Iterative $y' = Ay$	Direct $A = LU$
General A	GMRES, BiCGStab, ...	Pivoting LU
$A = A^T$	Conjugate gradient, …	Cholesky

↑ **More general**

↓ **More robust**

← → **Less storage (if sparse)**

Source: Gilbert (UCSB)