

CSE 245: Computer Aided Circuit Simulation and Verification

Fall 2004, Oct 12

Lecture 5:

Matrix Solver I

Outline

- Direct Method (Gaussian Elimination)
 - LU Factorization
 - Ordering and Pivoting
 - KLU
- Iterative Method (next lecture)
 - Stationary Iterative Method (SOR, GS, Jacob)
 - Krylov Method (CG, GMRES)
 - Multigrid Method

Direct vs Iterative

□ Direct methods:

- find the exact solution in a finite number of steps
- Robust
- Usually computational expensive.
- Good for small dense matrix

□ Iterative methods:

- Start with an initial “guess”, successively improve the guess until it is close enough to the exact solution (within error tolerance)
- Accuracy and Convergence Issues
- Good for large sparse matrix

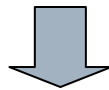
Linear Systems

- Linear Equation: $Ax = b$
 - A is a $n \times n$ square matrix
- For a n by n matrix A
 - $\text{Rank}(A)$: # of independent rows or columns
 - Full rank: $\text{rank}(A) = n$
 - Linear independent rows/columns
 - For $Ax = b$, one and only one solution for any b
 - Singular: $\text{rank}(A) < n$
 - For $Ax = b$, no solution or solution is not unique

Gaussian Elimination Basics

Reminder by 3x3 example

$$\begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$



$$M_{11}x_1 + M_{12}x_2 + M_{13}x_3 = b_1$$

$$M_{21}x_1 + M_{22}x_2 + M_{23}x_3 = b_2$$

$$M_{31}x_1 + M_{32}x_2 + M_{33}x_3 = b_3$$

Gaussian Elimination (1)

Use Eqn 1 to Eliminate x_1 from Eqn 2 and 3

$$M_{11}x_1 + M_{12}x_2 + M_{13}x_3 = b_1$$

$$\left(M_{22} - \frac{M_{21}}{M_{11}} M_{12} \right) x_2 + \left(M_{23} - \frac{M_{21}}{M_{11}} M_{13} \right) x_3 = b_2 - \frac{M_{21}}{M_{11}} b_1$$

$$\left(M_{32} - \frac{M_{31}}{M_{11}} M_{12} \right) x_2 + \left(M_{33} - \frac{M_{31}}{M_{11}} M_{13} \right) x_3 = b_3 - \frac{M_{31}}{M_{11}} b_1$$

Gaussian Elimination (2)

Remove x_1 from eqn 2 and eqn 3

Pivot MULTIPLIERS

$$\begin{bmatrix}
 M_{11} & M_{12} & M_{13} \\
 0 & \left(M_{22} - \left(\frac{M_{21}}{M_{11}} \right) M_{12} \right) & \left(M_{23} - \left(\frac{M_{21}}{M_{11}} \right) M_{13} \right) \\
 0 & \left(M_{32} - \left(\frac{M_{31}}{M_{11}} \right) M_{12} \right) & \left(M_{33} - \left(\frac{M_{31}}{M_{11}} \right) M_{13} \right)
 \end{bmatrix}
 \begin{bmatrix}
 x_1 \\
 x_2 \\
 x_3
 \end{bmatrix}
 =
 \begin{bmatrix}
 b_1 \\
 b_2 - \left(\frac{M_{21}}{M_{11}} \right) b_1 \\
 b_3 - \left(\frac{M_{31}}{M_{11}} \right) b_1
 \end{bmatrix}$$

Gaussian Elimination (3)

Remove x2 from eqn 3

Pivot

$$\begin{array}{ccc}
 M_{11} & M_{12} & M_{13} \\
 0 & \left(M_{22} - \left(\frac{M_{21}}{M_{11}} \right) M_{12} \right) & \left(M_{23} - \left(\frac{M_{21}}{M_{11}} \right) M_{13} \right) \\
 0 & 0 & \left(M_{33} - \left(\frac{M_{31}}{M_{11}} \right) M_{13} - \frac{\left(M_{32} - \left(\frac{M_{31}}{M_{11}} \right) M_{12} \right) \left(M_{23} - \left(\frac{M_{21}}{M_{11}} \right) M_{13} \right)}{\left(M_{22} - \left(\frac{M_{21}}{M_{11}} \right) M_{12} \right)} \right)
 \end{array}$$

Multiplier

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 - \left(\frac{M_{21}}{M_{11}} \right) b_1 \\ b_3 - \left(\frac{M_{31}}{M_{11}} \right) b_1 - \frac{\left(M_{32} - \left(\frac{M_{31}}{M_{11}} \right) M_{12} \right) \left(b_2 - \left(\frac{M_{21}}{M_{11}} \right) b_1 \right)}{\left(M_{22} - \left(\frac{M_{21}}{M_{11}} \right) M_{12} \right)} \end{bmatrix}$$

Simplified Notation (1)

Remove x_1 from eqn 2 and eqn 3

$$\begin{bmatrix} M_{11} & M_{12} & M_{13} \\ 0 & \tilde{M}_{22} & \tilde{M}_{23} \\ 0 & \tilde{M}_{32} & \tilde{M}_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ \tilde{b}_2 \\ \tilde{b}_3 \end{bmatrix}$$

Simplified Notation (2)

Remove x_2 from eqn 3

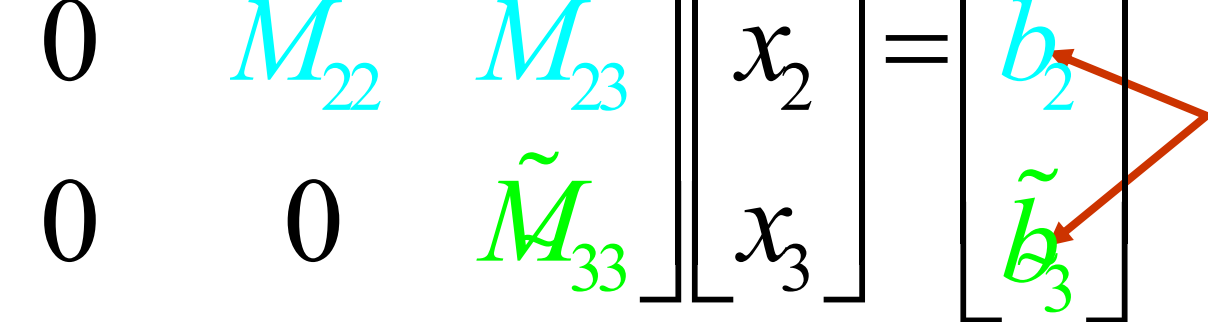
$$\begin{bmatrix}
 M_{11} & M_{12} & M_{13} \\
 0 & \tilde{M}_{22} & \tilde{M}_{23} \\
 0 & 0 & \tilde{M}_{33} - \frac{\tilde{M}_{32} \tilde{M}_{23}}{\tilde{M}_{22}}
 \end{bmatrix}
 \begin{bmatrix}
 x_1 \\
 x_2 \\
 x_3
 \end{bmatrix}
 =
 \begin{bmatrix}
 b_1 \\
 \tilde{b}_2 \\
 \tilde{b}_3 - \frac{\tilde{M}_{32} \tilde{b}_2}{\tilde{M}_{22}}
 \end{bmatrix}$$

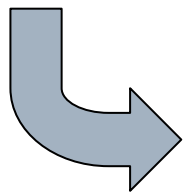
Pivot (points to $\tilde{M}_{22}$)
Multiplier (points to $\frac{\tilde{M}_{32} \tilde{M}_{23}}{\tilde{M}_{22}}$)

Upper Triangle Matrix U

$$\begin{bmatrix} M_{11} & M_{12} & M_{13} \\ 0 & \tilde{M}_{22} & \tilde{M}_{23} \\ 0 & 0 & \tilde{M}_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ \tilde{b}_2 \\ \tilde{b}_3 \end{bmatrix}$$

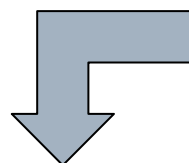
Altered During GE





$$\begin{bmatrix} U_{11} & U_{12} & U_{13} \\ 0 & U_{22} & U_{23} \\ 0 & 0 & U_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$$

Solve: Backward substitution


$$\begin{bmatrix} U_{11} & U_{12} & U_{13} \\ 0 & U_{22} & U_{23} \\ 0 & 0 & U_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$$

$$x_3 = \frac{y_3}{U_{33}}$$

$$x_2 = \frac{y_2 - U_{23}x_3}{U_{22}}$$

$$x_1 = \frac{y_1 - U_{12}x_2 - U_{13}x_3}{U_{11}}$$

Right Hand Side Update

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 - \left(\frac{M_{21}}{M_{11}} \right) b_1 \\ b_3 - \left(\frac{M_{31}}{M_{11}} \right) b_1 - \frac{\tilde{M}_{32}}{\tilde{M}_{22}} \tilde{b}_2 \end{bmatrix} \quad \rightarrow \quad \begin{bmatrix} 1 & 0 & 0 \\ \frac{M_{21}}{M_{11}} & 1 & 0 \\ \frac{M_{31}}{M_{11}} & \frac{\tilde{M}_{32}}{\tilde{M}_{22}} & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ L_{21} & 1 & 0 \\ L_{31} & L_{32} & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

Gauss Elimination Summary

$$(1) \begin{array}{l} M x = b \\ \quad \text{GE} \\ \quad \swarrow \\ U x = y \end{array}$$

Equivalent system
 U : upper trg

(2) Noticed that:

$$L y = b$$

L : unit lower trg

$$(3) \begin{array}{l} U x = y \\ LU x = b \rightarrow M x = b \end{array}$$

$\Rightarrow$ Efficient way of implementing GE: LU factorization

Gauss Elimination Summary

Solve $Mx = b$

Step 1

$$M = LU$$

$$\boxed{} = \begin{array}{|c|} \hline \triangle \\ \hline \end{array} \cdot \begin{array}{|c|} \hline \triangle \\ \hline \end{array}$$

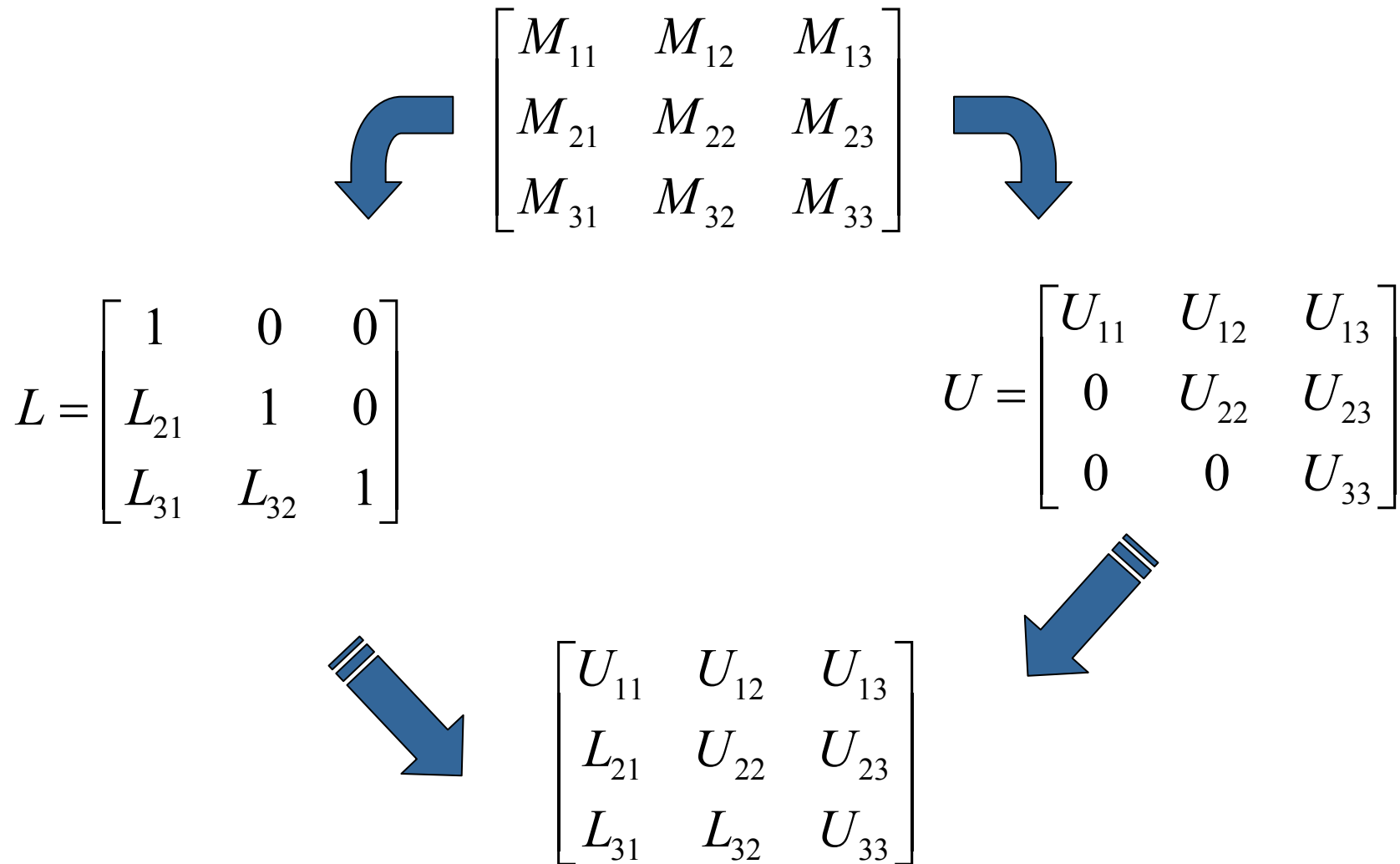
Step 2 Forward Elimination

Solve $Ly = b$

Step 3 Backward Substitution

Solve $Ux = y$

GE Basics – Fitting the pieces together



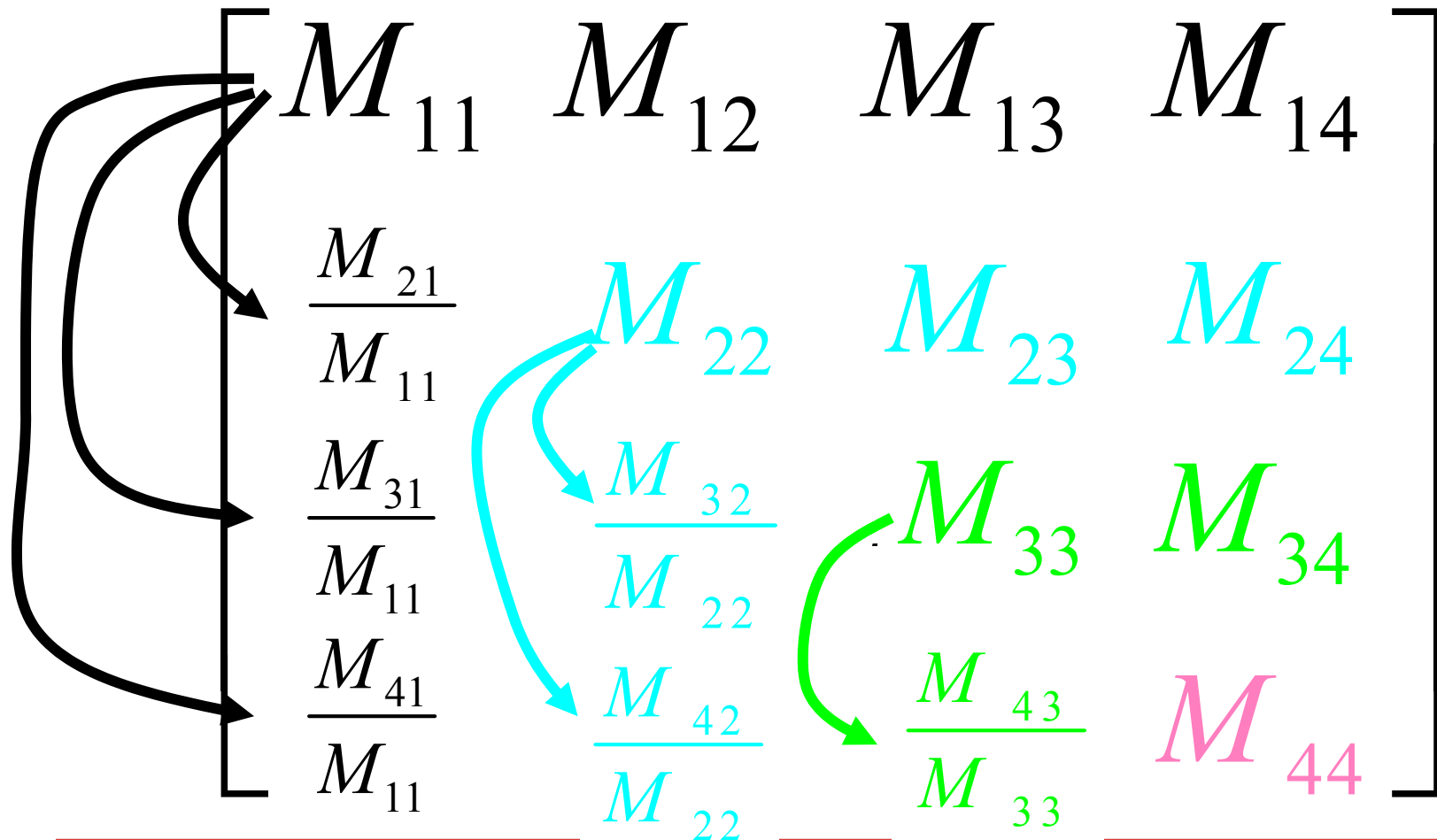
LU Factorization

$$L = \begin{bmatrix} 1 & 0 & 0 \\ L_{21} & 1 & 0 \\ L_{31} & L_{32} & 1 \end{bmatrix} \quad U = \begin{bmatrix} U_{11} & U_{12} & U_{13} \\ 0 & U_{22} & U_{23} \\ 0 & 0 & U_{33} \end{bmatrix}$$

$$LU = \begin{bmatrix} U_{11} & U_{12} & U_{13} \\ L_{21}U_{11} & L_{21}U_{12} + U_{22} & L_{21}U_{13} + U_{23} \\ L_{31}U_{11} & L_{31}U_{12} + L_{32}U_{22} & L_{31}U_{13} + L_{32}U_{23} + U_{33} \end{bmatrix}$$

$$\begin{bmatrix} U_{11} & U_{12} & U_{13} \\ L_{21} & U_{22} & U_{23} \\ L_{31} & L_{32} & U_{33} \end{bmatrix} = \begin{bmatrix} U_{11} & U_{12} & U_{13} \\ L_{21} = M_{21}/U_{11} & M_{22} - L_{21}U_{12} & M_{23} - L_{21}U_{13} \\ L_{31} = M_{31}/U_{11} & L_{32} = (M_{32} - L_{31}U_{12})/U_{22} & M_{33} - L_{31}U_{13} - L_{32}U_{23} \end{bmatrix}$$

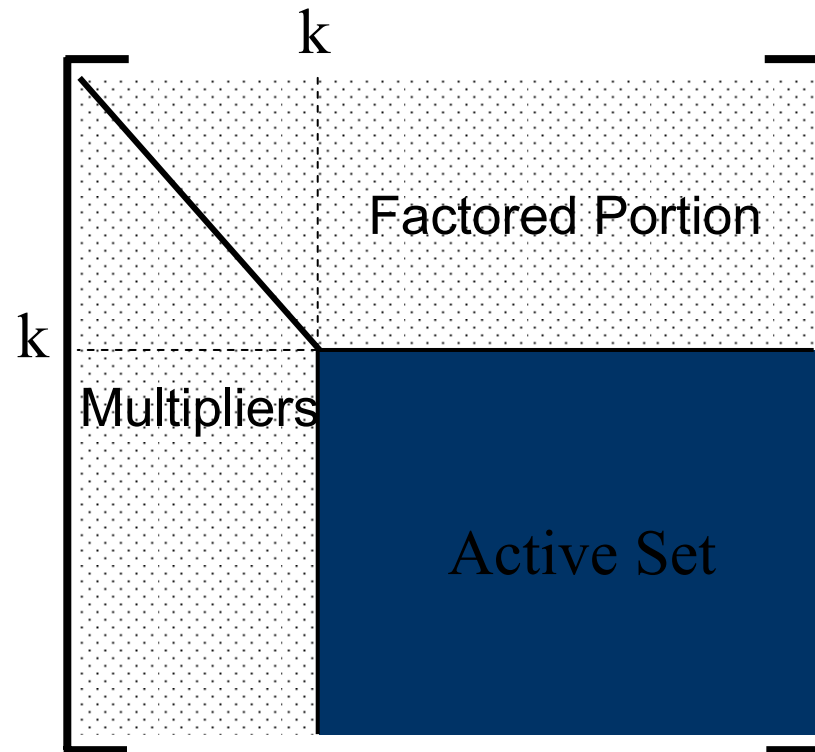
LU Factorization



Algorithm Description

For $i = 1$ to $n-1$ { “For each source row”
 For $j = i+1$ to n { “For each target row below the source”
 $M_{ji} \leftarrow \frac{M_{ji}}{M_{ii}}$ Pivot
 For $k = i+1$ to n { “For each row element beyond Pivot”
 $M_{jk} \leftarrow M_{jk} - M_{ji} M_{ik}$
 Multiplier
 }
 }
}

LU Factorization



Complexity Analysis

For $i = 1$ to $n-1$ { “For each Row”
 For $j = i+1$ to n { “For each target Row below the source”
 $M_{ji} = \frac{M_{ji}}{M_{ii}}$ Pivot
 For $k = i+1$ to n { “For each Row element beyond Pivot”
 $M_{jk} \leftarrow M_{jk} - M_{ji} M_{ik}$
 Multiplier
 }
 }
}

$$\sum_{i=1}^{n-1} (n-i) = \frac{n^2}{2} \text{ multipliers}$$

$$\sum_{i=1}^{n-1} (n-i)^2 \approx \frac{2}{3} n^3$$

Multiply-adds

Numerical Issues - Pivoting

- Zero Pivots

- Small Pivots (Round-off error)

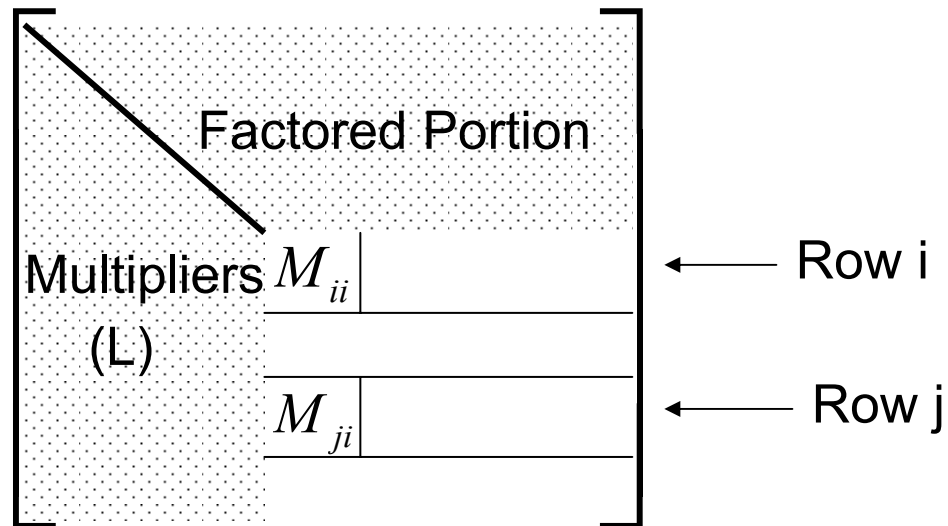
⇒ both can be solved with partial pivoting

Need for Pivoting

- When the diagonal is zero
 - To avoid zero denominator
- When diagonal term is much smaller than off-diagonal term
 - To reduce roundoff error

Zero Pivoting

At Step i



What if $M_{ii} = 0$? Cannot form $\frac{M_{ji}}{M_{ii}}$

Simple Fix (Row exchange)

If $M_{ii} = 0$

Find $M_{ji} \neq 0 \quad j > i$

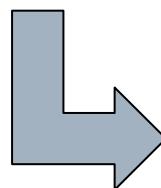
Swap Row j with i

Roundoff Error Example(1)

$$\begin{bmatrix} 1.25 \cdot 10^{-4} & 1.25 \\ 12.5 & 12.5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 6.25 \\ 75 \end{bmatrix}$$

GE


$$\begin{bmatrix} 1.25 \cdot 10^{-4} & 1.25 \\ 0 & 12.5 - 1.25 \cdot 10^5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 6.25 \\ 75 - 6.25 \cdot 10^5 \end{bmatrix}$$


$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_{5 \text{ digits}} = \begin{bmatrix} 1.0001 \\ 4.9999 \end{bmatrix}$$

Roundoff Error Example (2)

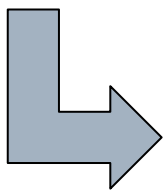
$$\begin{bmatrix} 1.25 \cdot 10^{-4} & 1.25 \\ 12.5 & 12.5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 6.25 \\ 75 \end{bmatrix}$$

GE

$$\begin{bmatrix} 1.25 \cdot 10^{-4} & 1.25 \\ 0 & 12.5 - 1.25 \cdot 10^5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 6.25 \\ 75 - 6.25 \cdot 10^5 \end{bmatrix}$$

Rounded to 3 digits

$$\begin{bmatrix} 1.25 \cdot 10^{-4} & 1.25 \\ 0 & -1.25 \cdot 10^5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 6.25 \\ -6.25 \cdot 10^5 \end{bmatrix}$$



$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_{3\text{digits}} = \begin{bmatrix} 0 \\ 5 \end{bmatrix}$$

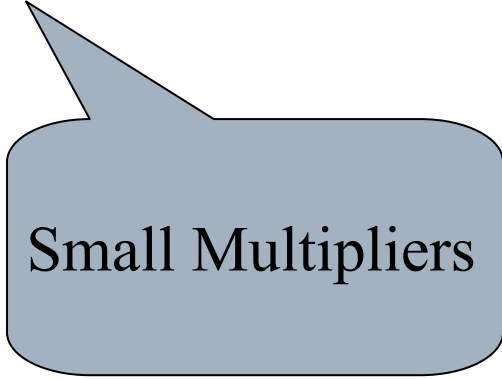
$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_{5\text{digits}} = \begin{bmatrix} 1.0001 \\ 4.9999 \end{bmatrix}$$

Partial Pivoting

Partial Pivoting for Roundoff reduction

If $|M_{ii}| < \max_{j>i} |M_{ji}|$

Swap row i with $\arg(\max_{j>i} |M_{ij}|)$



Small Multipliers

Partial Pivoting

swap

$$\begin{bmatrix} 1.25 \cdot 10^{-4} & 1.25 \\ 12.5 & 12.5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 6.25 \\ 75 \end{bmatrix}$$

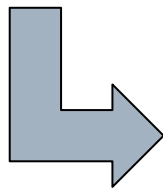
$$\begin{bmatrix} 12.5 & 12.5 \\ 1.25 \cdot 10^{-4} & 1.25 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 75 \\ 6.25 \end{bmatrix}$$

GE

$$\begin{bmatrix} 12.5 & 12.5 \\ 0 & 1.25 - 12.5 \cdot 10^{-5} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 75 \\ 6.25 - 75 \cdot 10^{-5} \end{bmatrix}$$

Rounded to 3 digits

$$\begin{bmatrix} 12.5 & 12.5 \\ 0 & 1.25 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 75 \\ 6.25 \end{bmatrix}$$

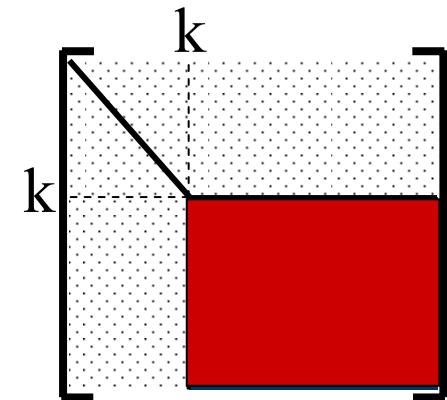
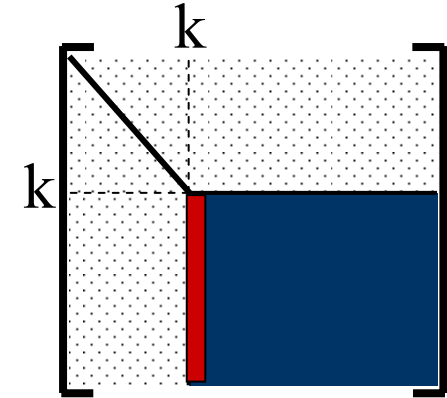


$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_{3\text{digits}} = \begin{bmatrix} 1 \\ 5 \end{bmatrix}$$

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_{5\text{digits}} = \begin{bmatrix} 1.0001 \\ 4.9999 \end{bmatrix}$$

Pivoting strategies

- Partial Pivoting:
 - Only row interchange
- Complete Pivoting
 - Row and Column interchange
- Threshold Pivoting(SPICE)
 - Only if prospective pivot is found to be smaller than a certain threshold



Need for Reordering

- Computational Complexity
 - $O(n^3)$, where $M: n \times n$
- We cannot afford this complexity
- Exploit natural sparsity that occurs in circuits equations
 - Sparsity: many zero elements
- Exploiting sparsity: $O(n^{1.1})$ to $O(n^{1.5})$

Sparse Data Structure

(1) Avoid storing zero entries

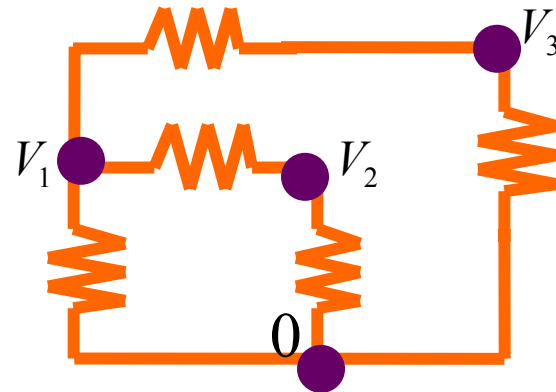
- Memory usage reduction
- Decomposition is faster since you do need to access them (but more complicated data structure)

(2) Avoid trivial operations

- Multiplication by zero
- Addition with zero

(3) Avoid losing sparsity → Less Nonzero fillins

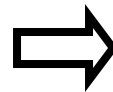
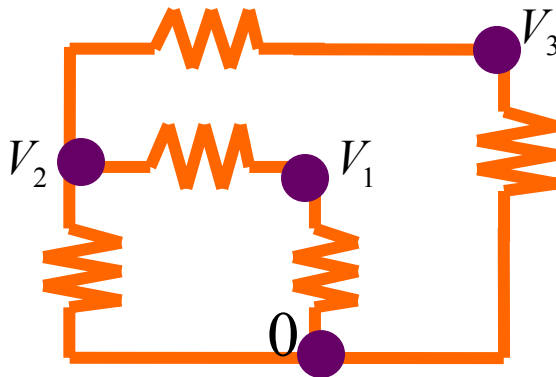
Sparse Matrices – Fill-in&Reordering



$$\begin{bmatrix} \text{X} & \text{X} & \text{X} \\ \text{X} & \text{X} & \text{X} \\ \text{X} & \text{X} & \text{X} \end{bmatrix}$$

Fill-ins

The matrix shows non-zero entries (X) for all pairs of nodes. The off-diagonal entries are circled in green, and arrows point to them from the text "Fill-ins".



$$\begin{bmatrix} \text{X} & \text{X} & 0 \\ \text{X} & \text{X} & \text{X} \\ 0 & \text{X} & \text{X} \end{bmatrix}$$

No Fill-ins

The matrix shows non-zero entries (X) for all pairs of nodes. The entry at (1,3) is 0, indicating no fill-in.

Node Reordering Can Reduce Fill-in

- Preserves Properties (Symmetry, Diagonal Dominance)
- Equivalent to swapping rows and columns

Reordering

- ❑ Objective:
 - Minimum number of fill-ins => less complexity
- ❑ Minimum # fill-ins
 - NP-Complete Problem
- ❑ Heuristics Algorithm
 - Various Minimum Degree Algorithm
 - Markowitz Product Algorithm: Used in Spice
 - ❑ 5% more fill-ins but
 - ❑ Faster
- ❑ Pivoting for accuracy may conflict with pivoting for sparsity

Fill-in Non-Zeros

- During solution of sparse linear system (by GE or LU), row-updates often result in creation of non-zero entries that were originally zero.

X	X	0	X	X	X	0	X
X	X	0	0	X	X	0	F
0	0	X	X	0	0	X	X
X	0	X	X	X	F	X	X

- Row updates using row-1 result in fill-in non-zeros (F).

Effect of reordering on fill-in

- Re-ordering the equations (rows) or unknowns (columns) can result in significant change in the number of fill-in non-zeros, and hence time for matrix factorization.

X	X	X	X		X	X	X	X
X	X	0	0	Fill-in with GE	X	X	F	F
X	0	X	0		X	F	X	F
X	0	0	X		X	F	F	X

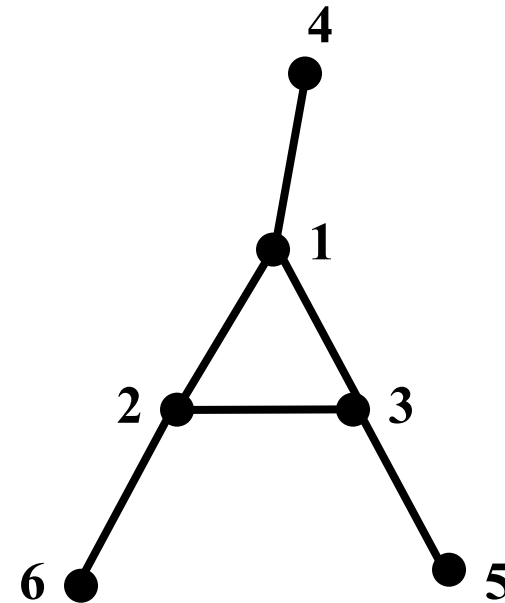
Reorder rows/cols
↓

X	0	0	X		X	0	0	X
0	X	0	X	No fill-in with GE	0	X	0	X
0	0	X	X		0	0	X	X
X	X	X	X		X	X	X	X

Associated graph of matrix

- A graph-based view of matrix's sparsity structure is extremely useful in generating low-fill re-orderings.

	1	2	3	4	5	6
1	X	X	X	X		
2	X	X	X			X
3	X	X	X		X	
4	X			X		
5			X		X	
6		X				X

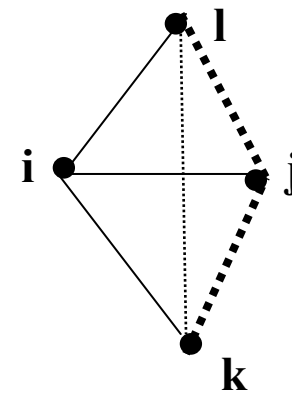
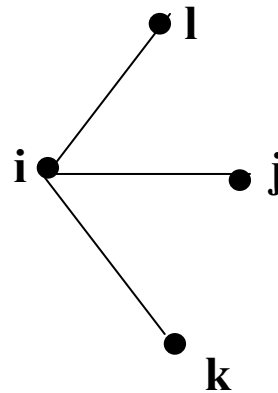


- The associated graph of a symmetric sparse matrix has a vertex corresponding to each row/col. of matrix, and an edge corresponding to each non-zero matrix entry.

Fill-in and graph transformation

- Row- i updates row- j , $j > i$ iff A_{ji} is non-zero; in the associated graph a matrix non-zero corresponds to an edge.
- Row-update($i \rightarrow j$) could cause fill-in non-zero A_{jk} corresponding to all non-zeros A_{ik} .

	i	j	k	l
i	X	X	X	X
j	X	X	F	F
k	X		X	
l	X			X

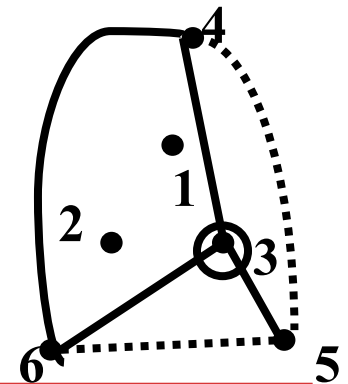
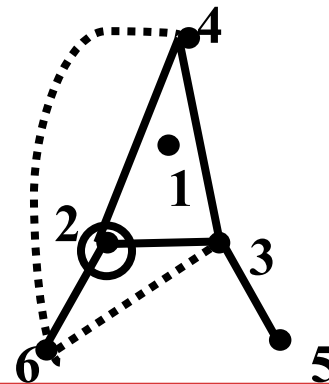
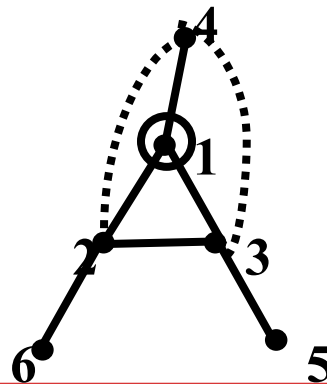
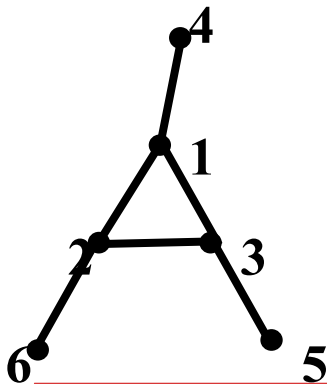


- After all updates from row- i , all neighbors of vertex i in the associated graph form a clique.

Fill-in and graph transformation

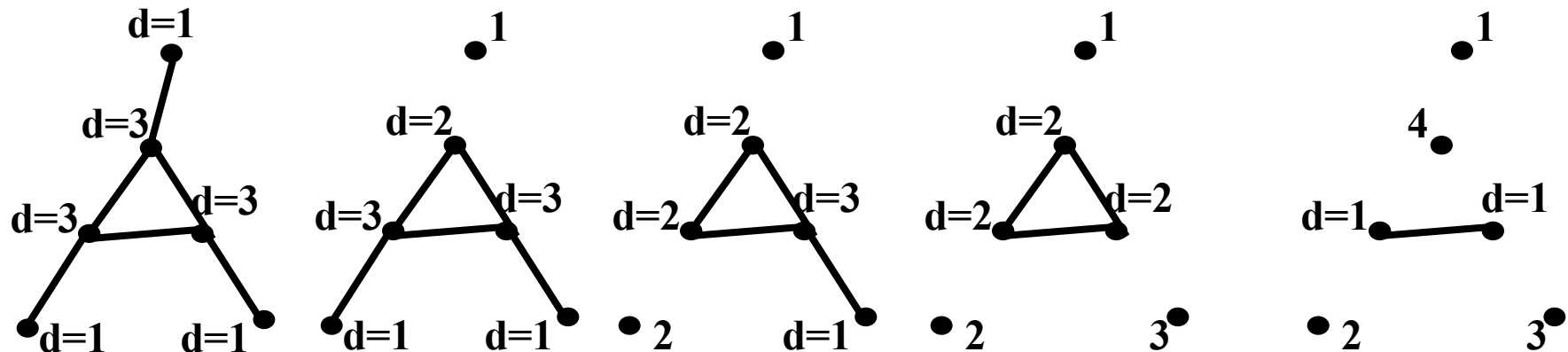
- Each row's effect on fill-in generation is captured by the “clique” transformation on the associated graph.
- The graph view is valuable in suggesting matrix re-ordering approaches.

	1	2	3	4	5	6		1	2	3	4	5	6		1	2	3	4	5	6		1	2	3	4	5	6
1	X	X	X	X			→	1	X	X	X	X			1							1					
2	X	X	X			X		2	X	X	X	F		X	2	X	X	F		X		2					
3	X	X	X		X		→	3	X	X	X	F	X		3	X	X	F	X	F		3			X	F	
4	X			X				4	X	F	F	X			4	F	F	X		F		4			F	F	
5			X		X			5			X	X			5		X		X			5			X	F	
6		X				X		6		X			X		6	X	F	F		X		6			F	F	X



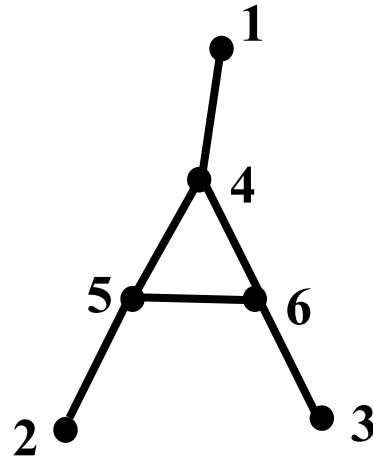
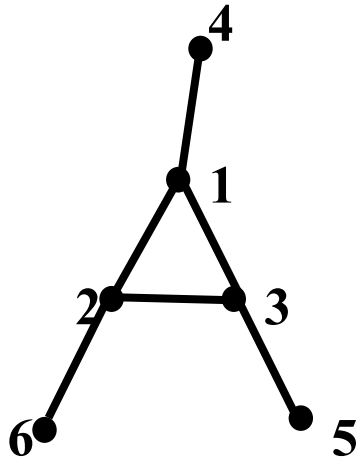
Matrix re-ordering: Minimum Degree

- Graph-based algorithm for generating low-fill re-ordering.
- Matrix permutation is viewed as node-numbering problem in associated graph.
- Low-degree nodes are numbered early - so that they are removed without adding many fill-in edges.



- For example, minimum-degree finds a no-fill ordering.

Re-ordered matrix



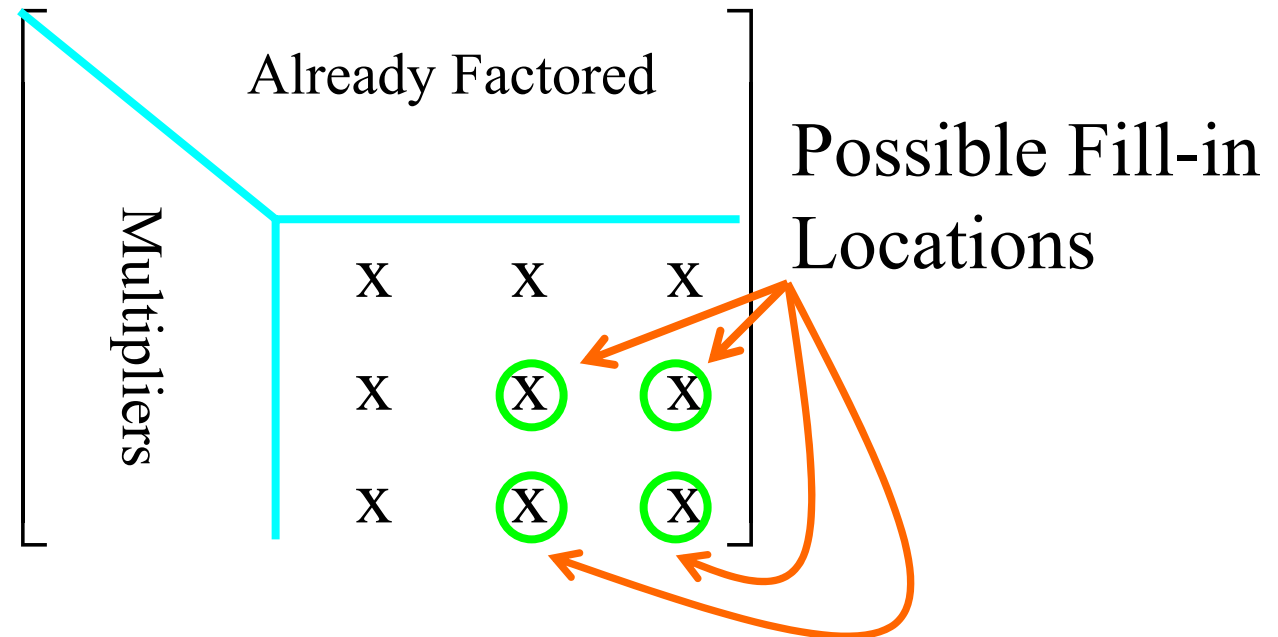
Old #	New #
1	4
2	5
3	6
4	1
5	3
6	2

	1	2	3	4	5	6
1	X	X	X	X		
2	X	X	X	F		X
3	X	X	X	F	X	F
4	X	F	F	X	F	F
5			X	F	X	F
6		X	F	F	F	X

	1	2	3	4	5	6
1	X			X		
2		X			X	
3			X			X
4	X			X	X	X
5		X		X	X	X
6			X	X	X	X

Reordering in Spice

Where can fill-in occur ?



Fill-in Estimate = (Non zeros in unfactored part of Row -1)
• (Non zeros in unfactored part of Col -1)
 $\equiv$ Markowitz product

Reordering-Markowitz Product

Markowitz Reordering (Diagonal Pivoting)

$i = 1$ to n

Find diagonal $j \geq i$ with min Markowitz Product

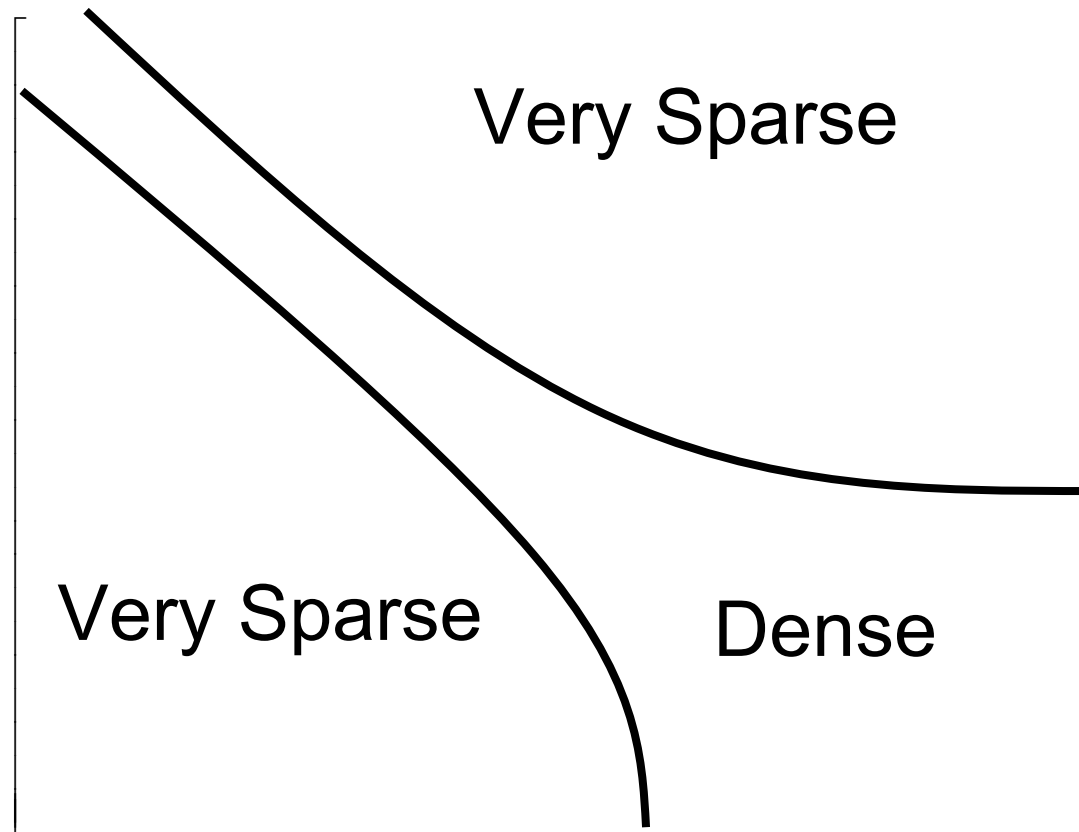
Swap rows $j \neq i$ and columns $j \neq i$

Factor the new row i and determine fill-ins

Greedy Algorithm (but close to optimal) !

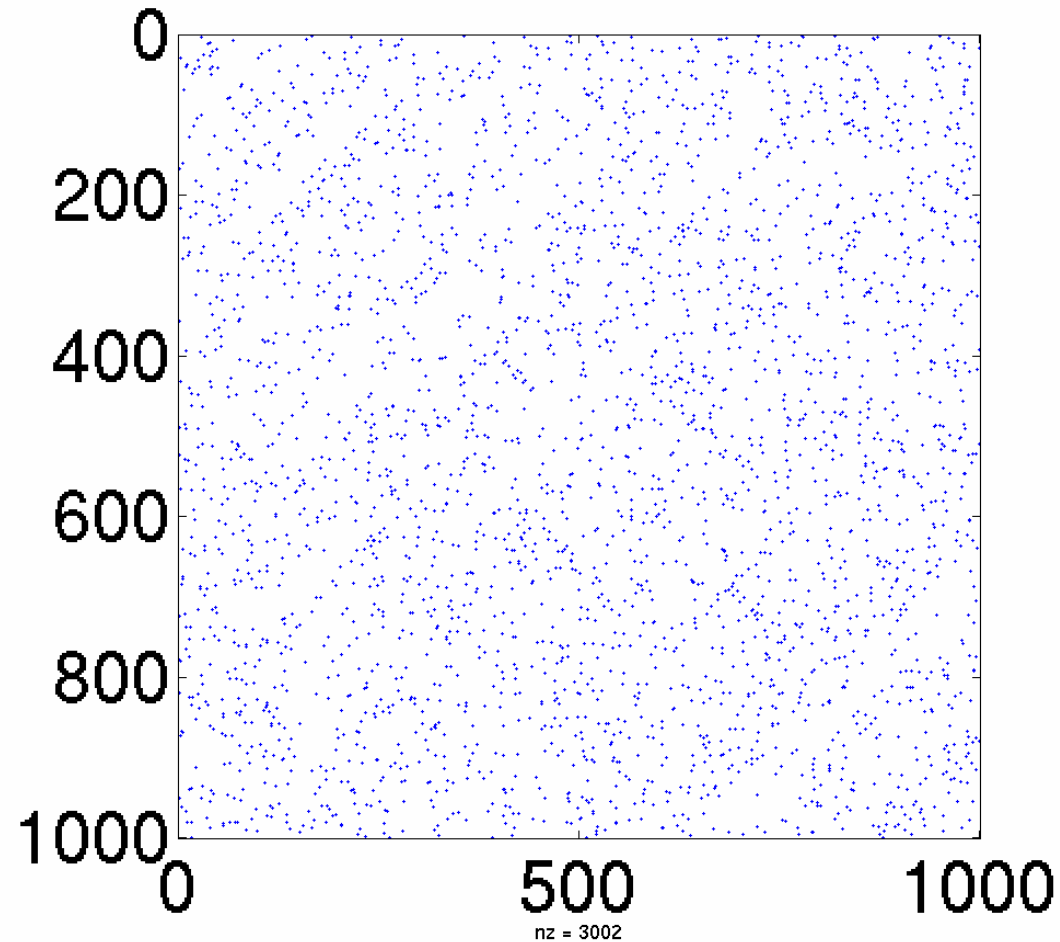
Reordering

Pattern of a Filled-in Matrix



Reordering

Unfactored random Matrix



Reordering

Factored random Matrix

