REORDERINGS FOR FILL-REDUCTION

Band and Envelope methods

- **Permutations and reorderings - graph interpretations**
- **Simple reorderings : Cuthill-Mc Kee, Reverse Cuthill Mc Kee**
- **Profile/envelope methods. Profile reduction.**
- **Multicoloring and independent sets [for iterative methods]**

Reorderings and graphs

➤ Let $\pi = \{i_1, \cdots, i_n\}$ a permutation

➤ $A_{\pi,*} = \{a_{\pi(i),j}\}_{i,j=1,\ldots,n}$ = matrix A with its i-th row replaced by row number $\pi(i)$.

➤ $A_{*,\pi}$ = matrix A with its j-th column replaced by column $\pi(j)$.

➤ Define $P_\pi = I_{\pi,*}$ = "Permutation matrix" – Then:

(1) Each row (column) of P_π consists of zeros and exactly one "1"
(2) $A_{\pi,*} = P_\pi A$
(3) $P_\pi P_\pi^T = I$
(4) $A_{*,\pi} = A P_\pi^T$

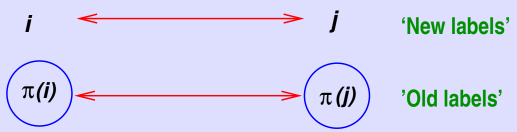
Consider now:

$$A' = A_{\pi,\pi} = P_\pi A P_\pi^T$$

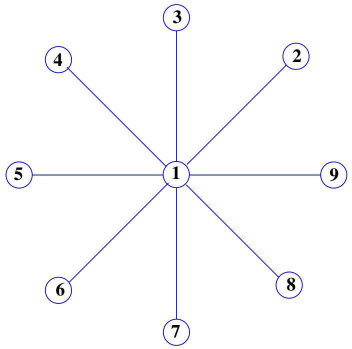
➤ Element in position (i,j) in matrix A' is exactly element in position $(\pi(i), \pi(j))$ in A. $(a'_{ij} = a_{\pi(i),\pi(j)})$

$$(i,j) \in E_{A'} \iff (\pi(i), \pi(j)) \in E_A$$

General picture :



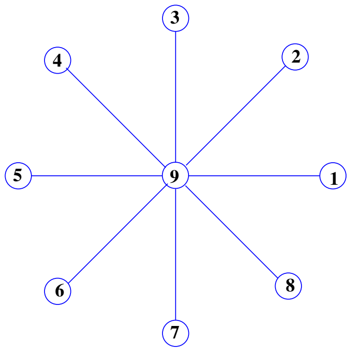
Example: A 9 × 9 'arrow' matrix and its adjacency graph.



$$\begin{bmatrix} * & * & * & * & * & * & * & * & * \\ * & * & & & & & & & \\ * & & * & & & & & & \\ * & & & * & & & & & \\ * & & & & * & & & & \\ * & & & & & * & & & \\ * & & & & & & * & & \\ * & & & & & & & * & \\ * & & & & & & & & * \end{bmatrix}$$

Fill-in?

➤ Graph and matrix after swapping nodes 1 and 9:



$$\begin{bmatrix} * & & & & & & & & * \\ & * & & & & & & & * \\ & & * & & & & & & * \\ & & & * & & & & & * \\ & & & & * & & & & * \\ & & & & & * & & & * \\ & & & & & & * & & * \\ & & & & & & & * & * \\ * & * & * & * & * & * & * & * & * \end{bmatrix}$$

✍ Fill-in?

The Cuthill-McKee and its reverse orderings

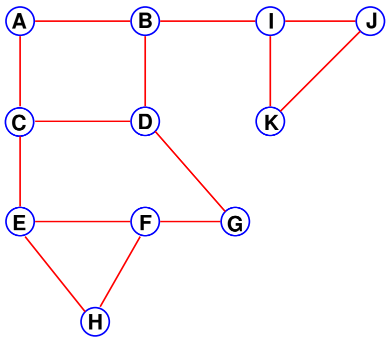
➤ A class of reordering techniques which proceed by levels in the graph.

➤ Related to Breadth First Search (BFS) traversal in graph theory.

➤ Idea of BFS is to visit the nodes by 'levels'. Level 0 = level of starting node.

➤ Start with a node, visit its neighbors, then the (unmarked) neighbors of its neighbors, etc...

Example:



Tree	Queue
A	B, C
A, B	C, I, D
A, B, C	I D, E
A, B, C, I	D, E, J, K
A, B, C, I, D	E, J, K, G
A, B, C, I, D, E	J, K, G, H, F

➤ Final traversal order:

A,	B, C,	I, D, E,	J, K, G, H, F
Level 0	Level 1	Level 2	Level 3

➤ Levels represent distances from the root

➤ Algorithm can be implemented by crossing levels 1,2, ...

➤ More common: Queue implementation

Algorithm $\boldsymbol{BFS(G, v)}$ – Queue implementation

- Initialize: $\boldsymbol{Queue} := \{\boldsymbol{v}\}$; Mark $\boldsymbol{v}$; $\boldsymbol{ptr} = 1$;
- While $\boldsymbol{ptr} < \boldsymbol{length}(\boldsymbol{Queue})$ do
  - $\boldsymbol{head} = \boldsymbol{Queue}(\boldsymbol{ptr})$;
  - ForEach Unmarked $\boldsymbol{w} \in \boldsymbol{Adj}(\boldsymbol{head})$:
    - $*$ Mark $\boldsymbol{w}$;
    - $*$ Add $\boldsymbol{w}$ to Queue: Queue = {Queue, w};
  - $\boldsymbol{ptr}++$;

A few properties of Breadth-First-Search

➤ If G is a connected undirected graph then each vertex will be visited once; each edge will be inspected at least once

➤ Therefore, for a connected undirected graph,

The cost of BFS is $O(|V| + |E|)$

➤ Distance = level number; ➤ For each node v we have:

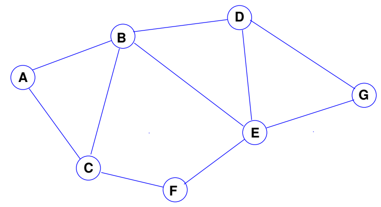
$$min\_dist(s, v) = level\_number(v) = depth_T(v)$$

➤ Several reordering algorithms are based on variants of Breadth-First-Search

Cuthill McKee ordering

Same as BFS except: Adj($head$) always sorted by increasing degree

Example:



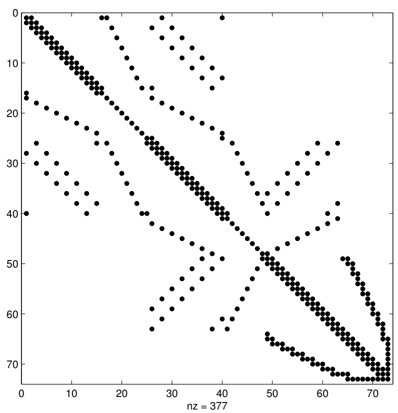
A	C(3) B(4)
A, C	B, F(2)
A, C, B	F, D(3), E(4)
A, C, B, F	D, E
A, C, B, F, D	E, G(2)
A, C, B, F, D, E	G
A, C, B, F, D, E, G	

Rule: when adding nodes to the queue list them in ↑ deg.

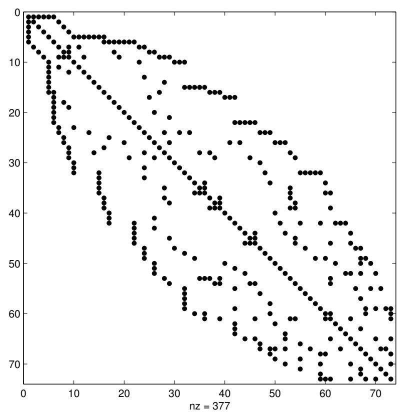
Reverse Cuthill McKee ordering

➤ The Cuthill - Mc Kee ordering has a tendency to create small arrow matrices (going the wrong way):

Origimal matrix

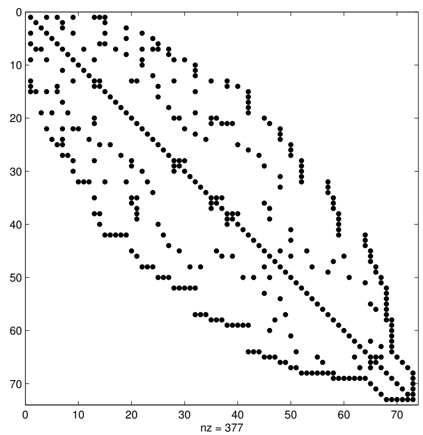


CM ordering



➤ Idea: Take the reverse ordering

RCM ordering



➤ Reverse Cuthill M Kee ordering (RCM).

Envelope/Profile methods

Many terms used for the same methods: Profile, Envelope, Skyline, ...

- Generalizes band methods
- Consider only the symmetric (in fact SPD) case
- Define bandwith of row i. ("i-th bandwidth of A):

$$\beta_i(A) = \max_{j \leq i; a_{ij} \neq 0} |i - j|$$

Definition: Envelope of A is the set of all pairs (i, j) such that $0 < i - j \leq \beta_i(A)$. The quantity $|Env(A)|$ is called profile of A.

Main result The envelope is preserved by GE (no-pivoting)

Theorem: Let $A = LL^T$ the Cholesky factorization of A. Then

$$Env(A) = Env(L + L^T)$$

- An envelope / profile/ Skyline method is a method which treats any entry a_{ij}, with $(i, j) \in Env(A)$ as nonzero.

- Definition. Frontwidth:

$$\omega_i(A) = |\{k > i \,|a_{kl} \neq 0 \text{ for some } l \leq i\}|$$

- $\omega_i(A)$ = number of active rows at $i-th$ step of GE = Number of rows in $Env(A)$ which intersect column i.
- Cost of an envelope method is

$$\sum_{i=1}^{n} \omega_i(A)(\omega_i(A) + 2)$$

Proof: Use earlier result on cost and notice that $\eta_i = \omega_i + 1$

Matlab test: do the following

1. Generate `A = Lap2D(64,64)`
2. Compute `R = chol(A)`
3. show `nnz(R)`
4. Compute RCM permutation (symrcm)
5. Compute `B = A(p,p)`
6. `spy(B)`
7. compute `R1 = chol(B)`
8. Show `nnz(R)`
9. `spy(R1)`

Papers to read:

Main:

- GIBBS, N E., POOLE, W G., JR., AND STOCKMEYER, P.K. An algorithm for reducing the bandwidth and profile of a sparse matrix. SIAM J. Numer. Analyszs 13, 2 (April 1976), 235-251
- John G. Lewis, Implementation of the Gibbs-Poole-Stockmeyer and Gibbs-King Algorithms, ACM Transactions on Mathematical Software (TOMS), v.8 n.2, p.180-189, June 1982

Others:

- KING, I.P. An automatic reordering scheme for simultaneous equations derived from network systems. Int. J. Numer. Methods Engrg. 2 (1970), 523-533.
- Norman E. Gibbs and William G. Poole, Jr. and Paul K. Stockmeyer, A Comparison of Several Bandwidth and Profile Reduction Algorithms, ACM Trans. Math. Softw., vol 2, number 4, (1976), pages 322–330.

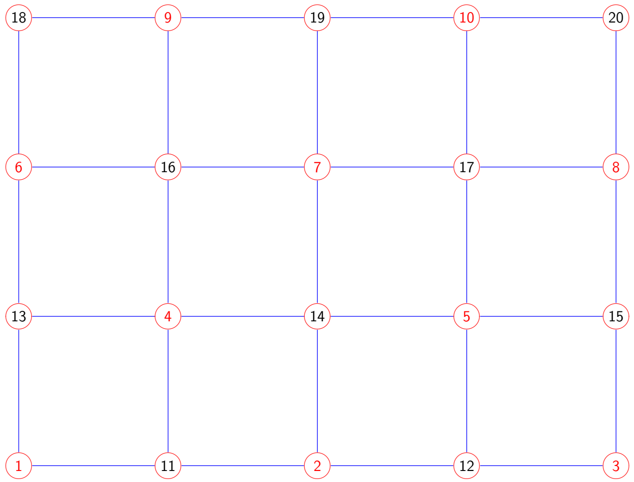
Orderings for iterative methods: Multicoloring

➤ General technique that can be exploited in many different ways to introduce parallelism – generally of order N.

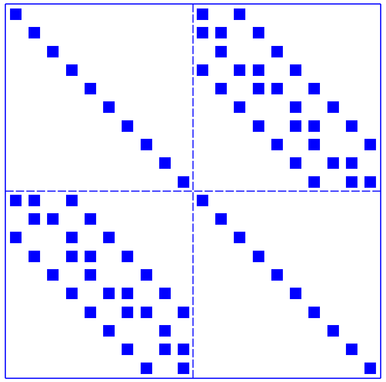
➤ Constitutes one of the most successful techniques for introducing vector computations for iterative methods..

➤ Want: assign colors so that no two adjacent nodes have the same color.

Simple example: Red-Black ordering.



Corresponding matrix



➤ Observe: L-U solves (or SOR sweeps) in Gauss-Seidel will require only diagonal scalings + matrix-vector products with matrices of size $N/2$.

How to generalize Red-Black ordering?

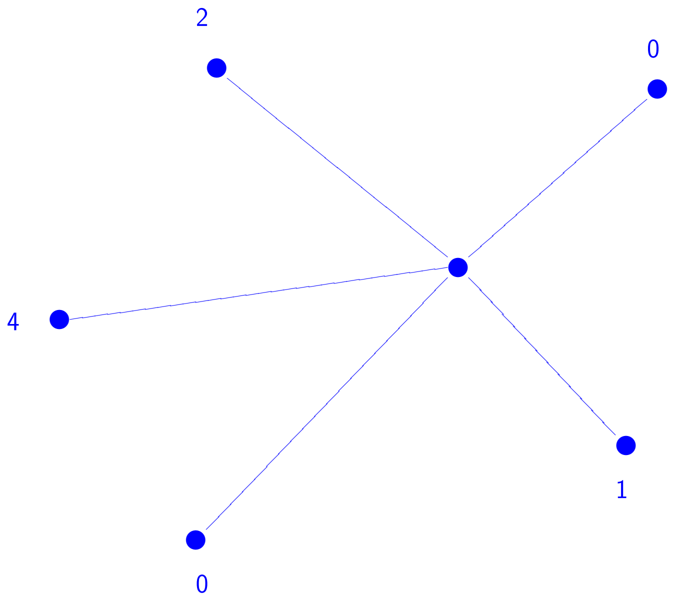
Answer: **Multicoloring** & **independent sets**

A greedy multicoloring technique:

- Initially assign color number zero (uncolored) to every node.
- Choose an order in which to traverse the nodes.
- Scan all nodes in the chosen order and at every node i do

$$Color(i) = \min\{k \neq 0 | k \neq Color(j), \forall\, j \in \text{ Adj (i)}\}$$

Adj(i) = set of nearest neighbors of $i = \{k \mid a_{ik} \neq 0\}$.



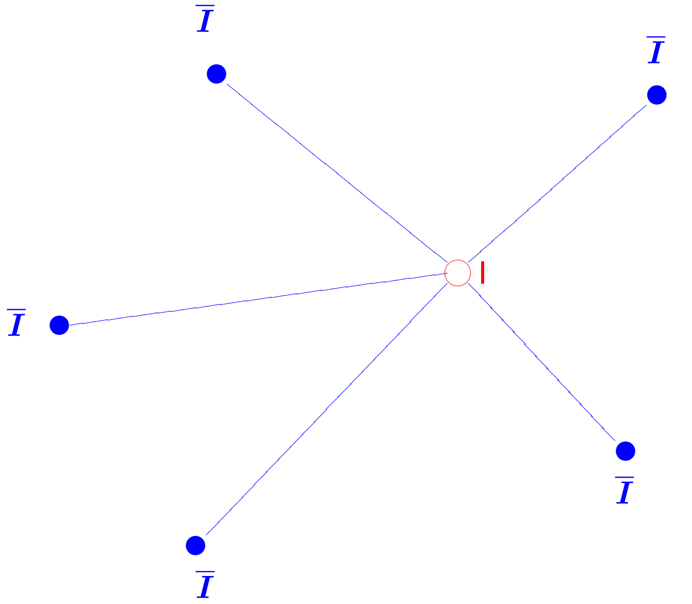
Independent Sets

An independent set (IS) is a set of nodes that are not coupled by an equation. The set is maximal if all other nodes in the graph are coupled to a node of IS. If the unknowns of the IS are labeled first, then the matrix will have the form:

$$\begin{bmatrix} B & F \\ E & C \end{bmatrix}$$

in which B is a diagonal matrix, and E, F, and C are sparse.

Greedy algorithm: Scan all nodes in a certain order and at every node i do: if i is not colored color it Red and color all its neighbors Black. Independent set: set of red nodes. Complexity: $O(|E| + |V|)$.



Show that the size of the independent set I is such that

$$|I| \geq \frac{n}{1 + d_I}$$

where d_I is the maximum degree of each vertex in I (not counting self cycle).

➤ According to the above inequality what is a good (heuristic) order in which to traverse the vertices in the greedy algorithm?

➤ Are there situations when the greedy alorithm for independent sets yield the same sets as the multicoloring algorithm?