REORDERINGS FOR FILL-REDUCTION

Band and Envelope methods

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

Reorderings and graphs

Let \( \pi = \{i_1, \cdots, i_n\} \) a permutation

\[
\begin{align*}
\pi & = \{i_1, \cdots, i_n\} \\
A_{\pi, *} & = \{a_{\pi(i), j}\}_{i,j=1,\ldots,n} = \text{matrix } A \text{ with its } i\text{-th row replaced by row number } \pi(i). \\
A_{*, \pi} & = \text{matrix } A \text{ with its } j\text{-th column replaced by column } \pi(j). \\
P_\pi & = I_{\pi,*} = \text{"Permutation matrix"} \\
\end{align*}
\]

(1) Each row (column) of \( P_\pi \) 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:

![Diagram](image)

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

![Matrix and Graph](image)
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:

<table>
<thead>
<tr>
<th>Tree</th>
<th>Queue</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>A, B, C</td>
</tr>
<tr>
<td>A, B</td>
<td>C, I, D</td>
</tr>
<tr>
<td>A, B, C</td>
<td>I, D, E</td>
</tr>
<tr>
<td>A, B, C, I</td>
<td>D, E, J, K</td>
</tr>
<tr>
<td>A, B, C, I, D</td>
<td>E, J, K, G</td>
</tr>
</tbody>
</table>

Final traversal order:


Levels represent distances from the root

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

More common: Queue implementation

Algorithm $BFS(G, v)$ – Queue implementation

- Initialize: $Queue := \{v\}$; Mark $v$; $ptr = 1$;
- While $ptr < length(\text{Queue})$ do
  - $head = \text{Queue}(ptr)$;
  - ForEach Unmarked $w \in Adj(head)$:
    - Mark $w$;
    - Add $w$ to Queue: $Queue = \{\text{Queue}, w\}$;
  - $ptr += 1;$
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:
  \[ \min_{s} \text{dist}(s, v) = \text{level}_T(v) = \text{depth}_T(v) \]
- 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:

<table>
<thead>
<tr>
<th>A</th>
<th>C(3) B(4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A, C</td>
<td>B, F(2)</td>
</tr>
<tr>
<td>A, C, B</td>
<td>F, D(3), E(4)</td>
</tr>
<tr>
<td>A, C, B, F</td>
<td>D, E</td>
</tr>
<tr>
<td>A, C, B, F, D</td>
<td>E, G(2)</td>
</tr>
<tr>
<td>A, C, B, F, D, E</td>
<td>G</td>
</tr>
</tbody>
</table>

Rule: when adding nodes to the queue list them in \( \uparrow \) deg.

Reverse Cuthill McKee ordering

- The Cuthill-McKee ordering has a tendency to create small arrow matrices (going the wrong way):
  Original matrix
  CM ordering

Idea: Take the reverse ordering

RCM ordering

Reverse Cuthill McKee 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 \(A\):

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

**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.

**Matlab test: do the following**

1. Generate \(A = \text{Lap2D}(64, 64)\)
2. Compute \(R = \text{chol}(A)\)
3. show \(\text{nnz}(R)\)
4. Compute RCM permutation (symrcm)
5. Compute \(B = A(p, p)\)
6. \(\text{spy}(B)\)
7. compute \(R1 = \text{chol}(B)\)
8. Show \(\text{nnz}(R)\)
9. \(\text{spy}(R1)\)
Papers to read:

Main:


Others:


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**

![Matrix Diagram]

- 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

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

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

---

**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 algorithm for independent sets yield the same sets as the multicoloring algorithm?