REORDERINGS FOR FILL-REDUCTION

GENERAL SPARSE MATRICES

- **Minimal degree ordering**
- **Nested Dissection (ND) ordering**
- **Complexity of ND for model problems**

Orderings used in direct solution methods

➤ Two broad types of orderings used:

- Minimal degree ordering + many variations
- Nested dissection ordering + many variations

➤ Minimal degree ordering is easiest to describe:

At each step of GE, select next node to eliminate, as the node v of smallest degree. After eliminating node v, update degrees and repeat.

Minimal Degree Ordering

At any step i of Gaussian elimination define for any candidate pivot row j

$$\boxed{Cost(j) = (nz_c(j) - 1)(nz_r(j) - 1)}$$

where $nz_c(j)$ = number of nonzero elements in column j of 'active' matrix, $nz_r(j)$ = number of nonzero elements in row j of 'active' matrix.

- Heuristic: fill-in at step j is $\leq cost(j)$
- Strategy: select pivot with minimal cost.
- Local, greedy algorithm
- Good results in practice.

Many improvements made over the years

- Alan George and Joseph W-H Liu, THE EVOLUTION OF THE MINIMUM DEGREE ORDERING ALGORITHM, SIAM Review, vol 31 (1989), pp. 1-19.

Min. Deg. Algorithm	Storage (words)	Order. time
Final min. degree	1,181 K	43.90
Above w/o multiple elimn.	1,375 K	57.38
Above w/o elimn. absorption	1,375 K	56.00
Above w/o incompl. deg. update	1,375 K	83.26
Above w/o indistiguishible nodes	1,308 K	183.26
Above w/o mass-elimination	1,308 K	2289.44

➤ Results for a $\mathbf{180} \times \mathbf{180}$ 9-point mesh problem

➤ Since this article, many important developments took place.

➤ In particular the idea of "Approximate Min. Degree" and and "Approximate Min. Fill", see

• E. Rothberg and S. C. Eisenstat, NODE SELECTION STRATEGIES FOR BOTTOM-UP SPARSE MATRIX ORDERING, SIMAX, vol. 19 (1998), pp. 682-695.

• Patrick R. Amestoy, Timothy A. Davis, and Iain S. Duff. AN APPROXIMATE MINIMUM DEGREE ORDERING ALGORITHM. SIAM Journal on Matrix Analysis and Applications, 17 (1996), pp. 886-905.

Practical Minimal degree algorithms

First Idea: Use quotient graphs

* Avoids elimination graphs which are not economical

* Elimination creates cliques

* Represent each clique by a node termed an *element* (recall FEM methods)

* No need to create fill-edges and elimination graph

* Still expensive: updating the degrees

Second idea: Multiple Minimum degree

* Many nodes will have the same degree. Idea: eliminate many of them simultaneously –

* Specifically eliminate independent set of nodes with same degree.

Third idea: Approximate Minimum degree

* Degree updates are expensive –

* Goal: To save time.

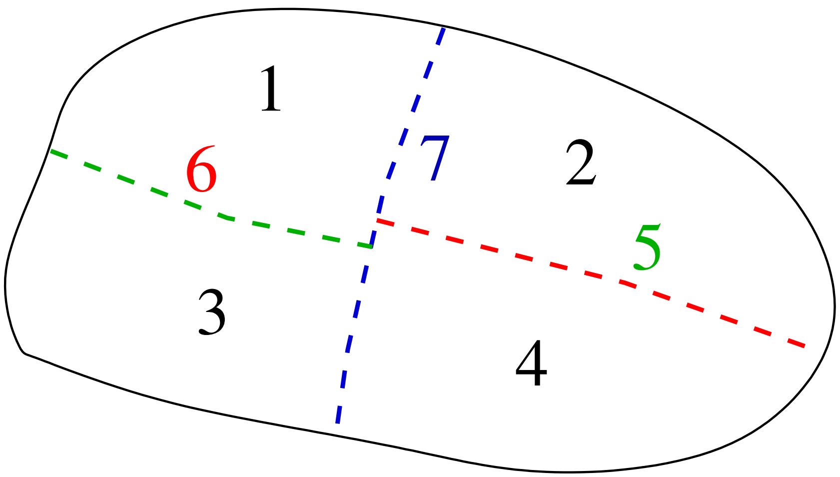
* Approach: only compute an approximation (upper bound) to degrees.

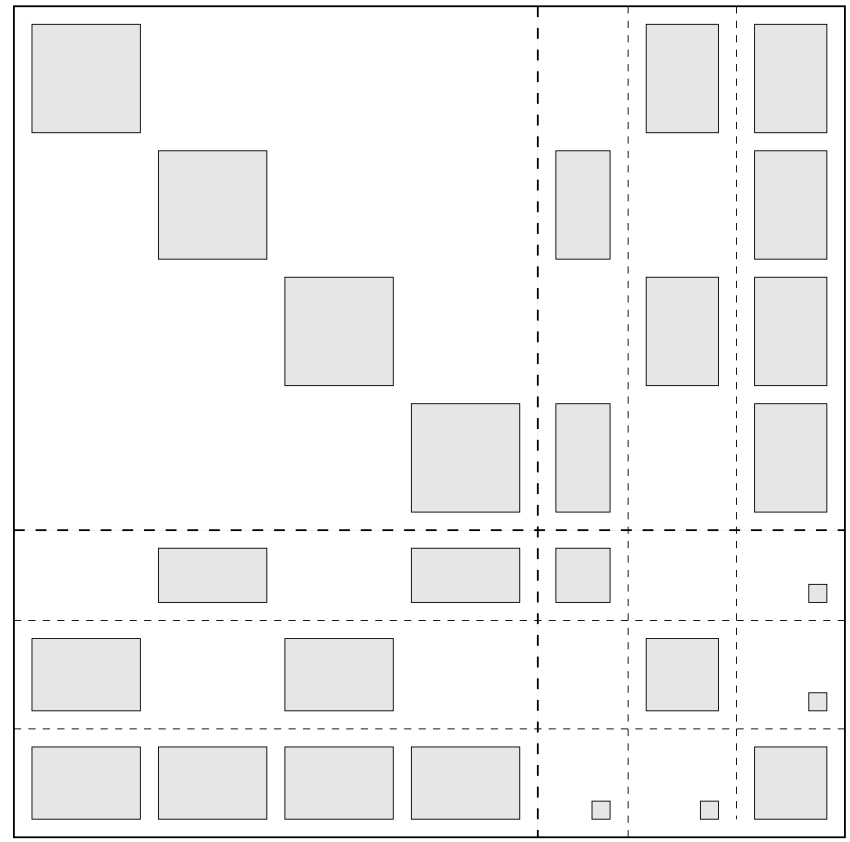
* Details are complicated and can be found in Tim Davis' book

Nested Dissection Reordering (Alan George)

- Computer science 'Divide-and-Conquer' strategy.
- Best illustration: PDE finite difference grid.
- Easily described by using recursivity and by exploiting 'separators': 'separate' the graph in three parts, two of which have no coupling between them. The 3rd set ('the separator') has couplings with vertices from both of the first 2 sets.
- Key idea: dissect the graph; take the subgraphs and dissect them recursively.
- Nodes of separators always labeled last after those of the parents

Nested dissection ordering: illustration



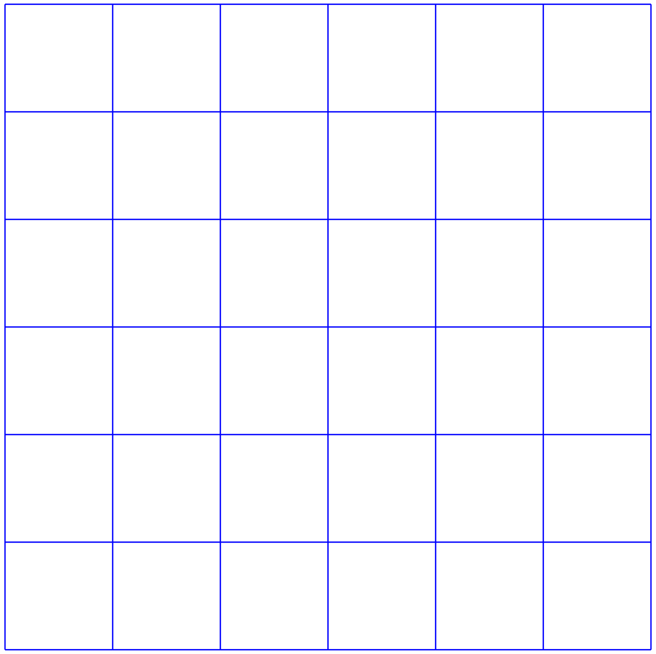


➤ For regular $n \times n$ meshes, can show: fill-in is of order $n^2 \log n$ and computational cost of factorization is $O(n^3)$

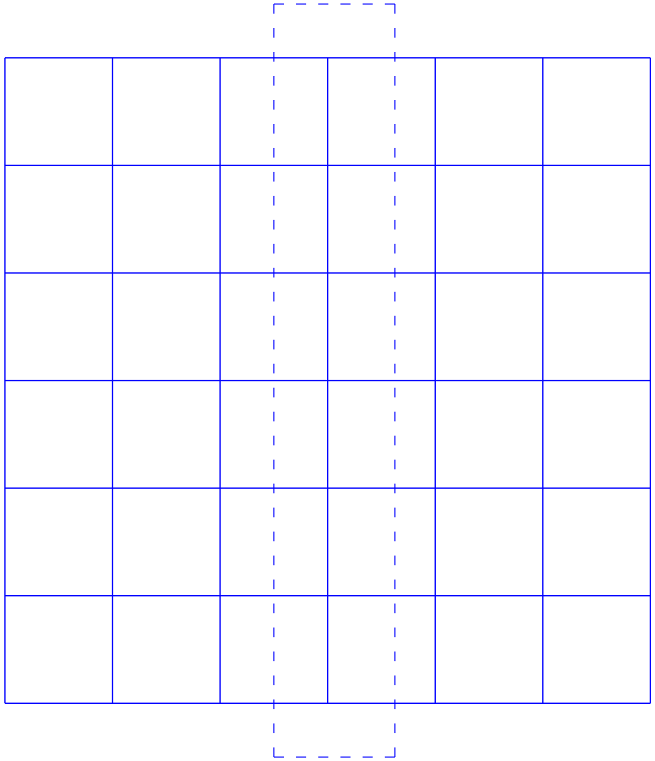
✍ How does this compare with a standard band solver?

Nested dissection for a small mesh

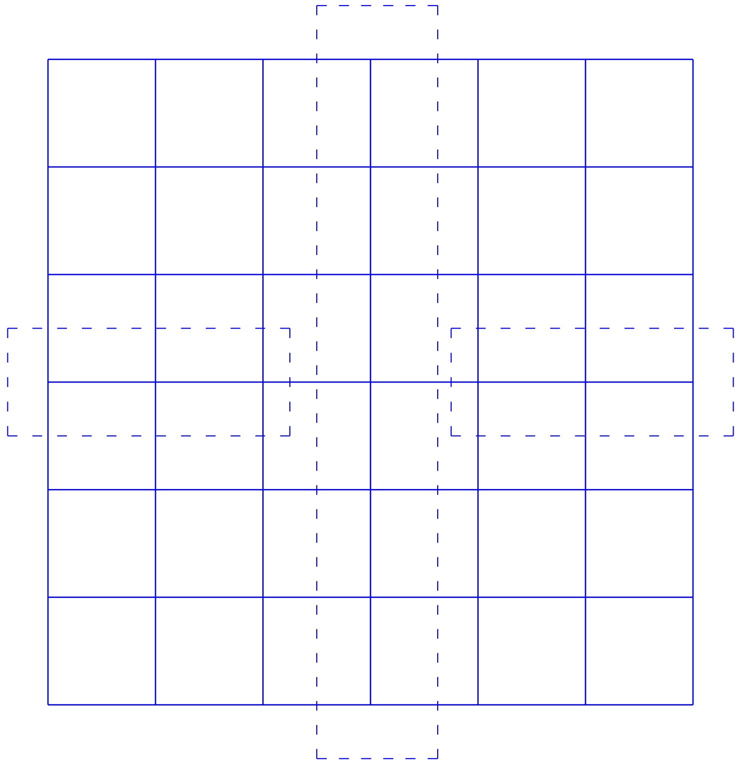
Original Grid



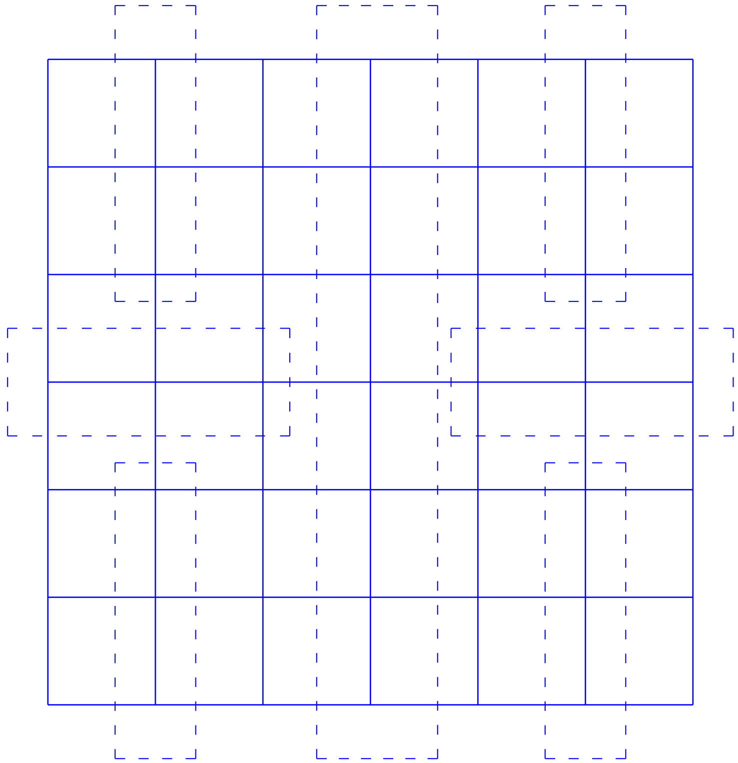
First dissection



Second Dissection



Third Dissection



Nested dissection: cost for a regular mesh

- In 2-D consider an $n \times n$ problem, $N = n^2$
- In 3-D consider an $n \times n \times n$ problem, $N = n^3$

	2-D	3-D
space (fill)	$O(N \log N)$	$O(N^{4/3})$
time (flops)	$O(N^{3/2})$	$O(N^2)$

- Significant difference in complexity between 2-D and 3-D

Nested dissection and separators

➤ Nested dissection methods depend on finding a good graph separator: $V = T_1 \cup U T_2 \cup S$ such that the removal of S leaves T_1 and T_2 disconnected.

➤ Want: S small and T_1 and T_2 of about the same size.

➤ Simplest version of the graph partitioning problem.

A theoretical result:

If G is a planar graph with N vertices, then there is a separator S of size $\leq \sqrt{N}$ such that $|T_1| \leq 2N/3$ and $|T_2| \leq 2N/3$.

In other words "Planar graphs have $O(\sqrt{N})$ separators"

➤ Many techniques for finding separators: Spectral, iterative swapping (K-L), multilevel (Metis), BFS, ...

The 2-D model problem

- 2-D finite difference mesh with N vertices.

Theorem:

With natural ordering, resulting fill-in is $\Theta(N^{3/2})$

Theorem:

With any ordering, resulting fill-in is $\Omega(N \ \log N)$

Theorem:

With nested dissection ordering, resulting fill-in is $O(N \ \log N)$

Ordering techniques in practice

- In practice: Nested dissection ($+$ variants) is preferred for parallel processing

- Good implementations of Min. Degree algorithm work well in practice. Currently AMD and AMF are best known implementations/variants/

- Best practical reordering algorithms usually combine Nested dissection and min. degree algorithms.