Sparse Matrix Ordering

Rob H. Bisseling

Mathematical Institute, Utrecht University

Course Introduction Scientific Computing
February 19, 2018

Outline

Cholesky factorisation

Sparse matrices and graphs

Minimum degree ordering

Nested dissection ordering

Separated Block-Diagonal ordering

Summary

Lower triangular matrices

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 5 & 8 \\ 3 & 8 & 14 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 3 & 2 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 3 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix} = LL^T.$$

- L is lower triangular if $l_{ij} = 0$ for all $i < j$.
- Cholesky factorisation is the factorisation of A into $A = LL^T$, with L lower triangular.
- Cholesky factor L exists and is unique if A is real, symmetric, positive definite.

Deriving an algorithm for Cholesky factorisation

Some simple algebra:

$$A = LL^T \iff a_{ij} = \sum_{r=0}^{n-1} L_{ir}(L^T)_{rj} = \sum_{r=0}^{n-1} l_{ir} l_{jr} \quad \text{for all } i, j.$$

Assume $i \geq j$. Then:

$$\begin{aligned} a_{ij} &= \sum_{r=0}^{n-1} l_{ir} l_{jr} = \sum_{r=0}^{j} l_{ir} l_{jr} \quad (\text{because } l_{jr} = 0 \text{ for } j < r) \\ &= \sum_{r=0}^{j-1} l_{ir} l_{jr} + l_{ij} l_{jj} \\ &\iff \\ l_{ij} &= \frac{1}{l_{jj}} \left(a_{ij} - \sum_{r=0}^{j-1} l_{ir} l_{jr} \right). \end{aligned}$$

Deriving an algorithm (cont'd)

Substituting $i = j$ gives:

$$l_{ii} = \left(a_{ii} - \sum_{r=0}^{i-1} l_{ir}^2 \right)^{\frac{1}{2}}.$$

Dense Cholesky factorisation

input: A: real symmetric positive definite $n \times n$ matrix.
output: L: lower triangular matrix such that $A = LL^T$.

$L := \mathrm{lower}(A)$;
for $k := 0$ **to** $n - 1$ **do**
  $l_{kk} := \sqrt{l_{kk}}$;
  for $i := k + 1$ **to** $n - 1$ **do**
    $l_{ik} := l_{ik} / l_{kk}$;
  for $j := k + 1$ **to** $n - 1$ **do**
    for $i := j$ **to** $n - 1$ **do**
      $l_{ij} := l_{ij} - l_{ik} l_{jk}$;

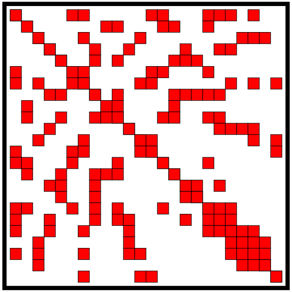
Sparse Cholesky factorisation

input: A: sparse real symmetric positive definite matrix.
output: L: lower triangular matrix such that $A = LL^T$.

$L := \mathrm{lower}(A)$;
for $k := 0$ **to** $n-1$ **do**
    $l_{kk} := \sqrt{l_{kk}}$;
    for $i := k+1$ **to** $n-1$ **do**
        if $l_{ik} \neq 0$ **then**
            $l_{ik} := l_{ik}/l_{kk}$;
    for $j := k+1$ **to** $n-1$ **do**
        if $l_{jk} \neq 0$ **then**
            for $i := j$ **to** $n-1$ **do**
                if $l_{ik} \neq 0$ **then**
                    $l_{ij} := l_{ij} - l_{ik} l_{jk}$;

If-statements are implemented by using a sparse data structure.

Sparse matrix can_24



24 × 24 matrix from Cannes with 160 nonzeros representing a 2D structural problem

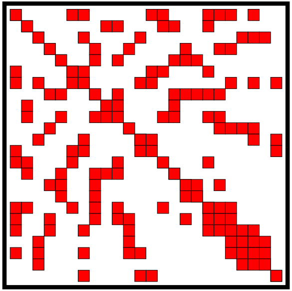
Universiteit Utrecht

Symbolic Cholesky factorisation

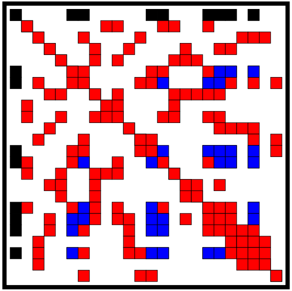
input: A: sparse real symmetric positive definite matrix.
output: L: sparsity pattern of Cholesky factor of A.

```
for j := 0 to n − 1 do
    for i := j to n − 1 do
        if a_ij ≠ 0 then
            l_ij := 1;
        else
            l_ij := 0;
for k := 0 to n − 1 do
    for j := k + 1 to n − 1 do
        if l_jk = 1 then
            for i := j to n − 1 do
                if l_ik = 1 then
                    l_ij := 1;
```

Sparse matrix can_24 at stage $k = 0$

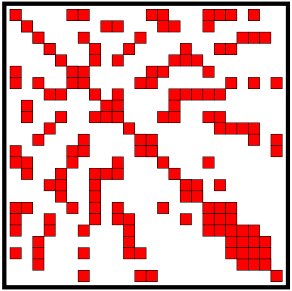


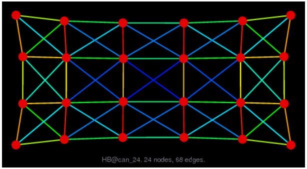
Sparse matrix can_24 at stage $k = 1$



Universiteit Utrecht

Sparse matrix can_24 and its graph





23
11
2
22

- 24×24 matrix: 24 vertices in graph, numbered $0, \ldots, 23$
- 136 off-diagonal nonzeros: 68 edges in graph

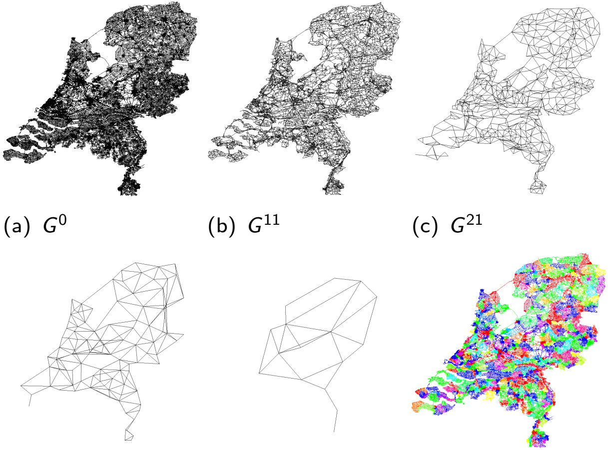
Relation sparse matrix and its graph

- For a symmetric $n \times n$ matrix A with nz off-diagonal nonzeros we can create the corresponding graph $G = (V, E)$.
- The vertex set V has n vertices.
- The edge set E is defined by

$$E = \{(i, j) \in V \times V \ : \ i > j \wedge a_{ij} \neq 0\}.$$

- E has $m = nz/2$ edges.

Clustering of road network of the Netherlands



(a) G^0 (b) G^{11} (c) G^{21}

(d) G^{26} (e) G^{33} (f) Best clustering (G^{21})

Open Street Map graph with 2,216,688 vertices and 2,441,238 edges

Cholesky factorisation in the graph

for $k := 0$ **to** $n-1$ **do**
  for $j := k+1$ **to** $n-1$ **do**
    if $l_{jk} = 1$ **then**
      for $i := j$ **to** $n-1$ **do**
        if $l_{ik} = 1$ **then**
          $l_{ij} := 1;$

- Row/column k of the matrix is eliminated in stage k.
- If $l_{ik} = 1$ and $l_{jk} = 1$, then l_{ij} becomes 1.
- Hence, if i and j are both neighbours in the graph of k, then they become neighbours of each other: $(i, j) \in E$.
- Vertex k and its edges are eliminated.
- Its neighbours become a clique (a dense subgraph).

Elimination process in the graph

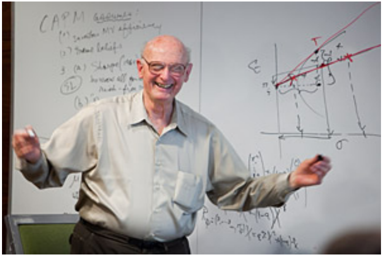
input: $G = (V, E)$: undirect graph.

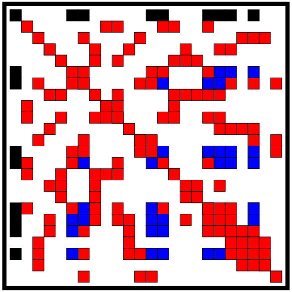
$$
\begin{array}{l}
\textbf{for } k := 0 \textbf{ to } n-1 \textbf{ do} \\
\quad V := V - \{k\}; \\
\quad E := E \cup \{(i,j) \;:\; i,j \in Adj(k) \wedge i > j\}; \\
\quad E := E - \{(i,k) \;:\; i \in Adj(k)\};
\end{array}
$$

- The adjacency set (or neighbour set) of vertex k is defined by

$$Adj(k) = \{i \in V \;:\; (i,k) \in E\}.$$

How to avoid fill-in?





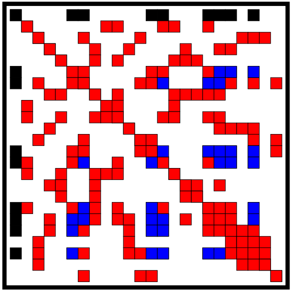
Nobel Laureate Harry Markowitz (b. 1927).
Source: `http://rady.ucsd.edu/rbj/2011/markowitz-interview/index.html`

- Markowitz (1957) criterion for unsymmetric matrices: minimise the upper bound m_{ij} on the fill-in (new nonzeros),

$$m_{ij} = (r_i - 1)(c_j - 1),$$

with r_i the number of nonzeros in row i,
and c_j the number in column j.

Symmetric matrices



- Choose nonzero a_{ii} from the diagonal.
- $r_i = c_i$, so we can simply choose i with minimum r_i.
- This corresponds to choosing a vertex i in the graph with minimum degree $r_i - 1$.

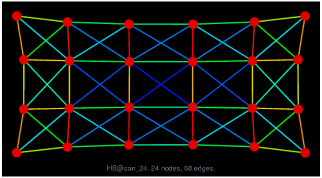
Minimum degree algorithm

input: $G = (V, E)$: undirect graph.
output: π: permutation, a minimum degree ordering of G.

$$
\begin{aligned}
&\textbf{for } r := 0 \textbf{ to } n-1 \textbf{ do}\\
&\quad mindeg(r) := \min\{deg(q) \ : \ q \in V\};\\
&\quad \pi(r) := \operatorname{argmin}\{deg(q) \ : \ q \in V\};\\
&\quad k := \pi(r);\\
&\quad V := V - \{k\};\\
&\quad E := E \cup \{(i,j) \ : \ i,j \in Adj(k) \wedge i > j\};\\
&\quad E := E - \{(i,k) \ : \ i \in Adj(k)\};
\end{aligned}
$$

- The ordering is not unique: ties can occur.
- Interesting connection: the treewidth of a graph G is the minimum possible value of $\max_r mindeg(r)$ for all possible permutations π (not only minimum degree permutations).

First split of nested dissection



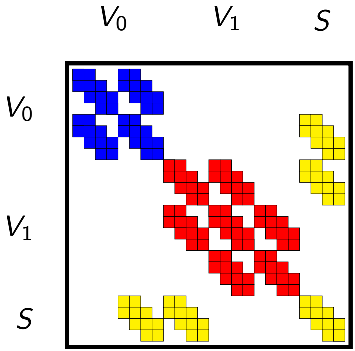
Renumber the grid points:

3	7	23	11	15	19
2	6	22	10	14	18
1	5	21	9	13	17
0	4	20	8	12	16

- $V = V_0 \cup S \cup V_1$, with $V_0 = \{0, \ldots, 7\}$, $V_1 = \{8, \ldots, 19\}$, and $S = \{20, \ldots, 23\}$.
- No edges between V_0 and V_1
- Separator set S is ordered last.

Universiteit Utrecht

Reordered matrix after one split

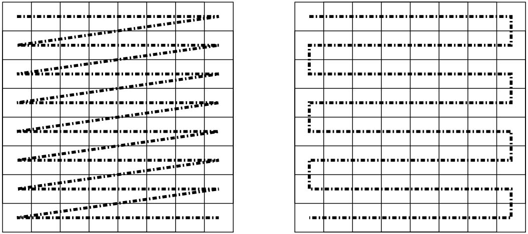


- Fill-in is limited to 2 diagonal blocks and separator rows/columns (yellow).
- Diagonal blocks can be handled in parallel.
- Structure is called Doubly Bordered Block Diagonal (DBBD)

Hybrid ordering

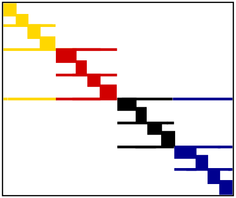
- Aim of hybrid: best of two worlds.
- Use nested dissection ordering for creating top-level blocks..
- Use minimum degree ordering within diagonal blocks.
- Advantage: both a local and a global view.

Ordering to improve cache use for SpMV



- Sparse matrix–vector multiplication (SpMV) $\mathbf{u} := A\mathbf{v}$, the workhorse of iterative linear system solvers.
- Dense input vector $\mathbf{v}$ is stored as a row.
- Left: Compressed Row Storage (CRS) with all nonzeros of the same row stored together, sorted by increasing column index.
- Right: zig-zag CRS, which avoids unnecessary end-of-row jumps in cache, thus improving access to the input vector.

Separated Block-Diagonal (SBD) ordering



- The SBD ordering is obtained by recursively partitioning the columns of a sparse matrix, each time moving the cut (mixed) rows to the middle. Columns are permuted accordingly.
- The cut rows are sparse and serve as a gentle transition between cache accesses to two different vector parts.
- The ordering itself is not symmetric.
- Yzelman and Bisseling, *SIAM Journal on Scientific Computing* **31**, No. 4, pp. 3128–3154, 2009.

Summary

- We examined variants of Cholesky factorisation $A = LL^T$:
  - Dense: nearly all matrix elements are nonzero
  - Sparse: most elements are zero
  - Symbolic: use only the sparsity pattern
- We have seen two basic sparse matrix ordering methods:
  - Minimum degree: greedy, local view,
  - Nested dissection: global view, creates parallelism, provably optimal for regular 2D grid
- A hybrid between minimum degree and nested dissection tries to get the best of both worlds.
- The Separated Block-Diagonal ordering is cache-friendly, for all sizes of cache.