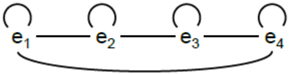
4.2. Sparse Matrices and Graphs

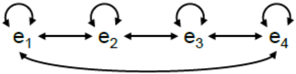
4.2.1. Graph $G(A)$ for symmetric positive definite (SPD) $A = A^T > 0$

$n \times n$-matrix: vertices $e_1, \ldots, e_n$ with edges (e_i, e_k) for $a_{ik} \neq 0$, undirected Graph

$$A = \begin{pmatrix} * & * & 0 & * \\ * & * & * & 0 \\ 0 & * & * & * \\ * & 0 & * & * \end{pmatrix} \rightarrow G(A):$$



G(A) as directed graph:



Adjacency Matrix for $G(A)$ or A

- Can be obtained directly by replacing in A each nonzero by 1.

$$\mathcal{A}(G(A)) = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix}$$

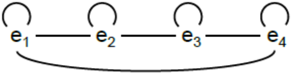
- Symmetric permutations of A in the form PAP^T change the ordering of the rows and columns of A simultaneously.
- Therefore, the graph of PAP^T can be obtained by the graph of A by renumbering the vertices.

Matrix A with graph G(A) (cont.)

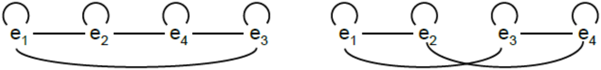
Symmetric permutation PAP^T with graph $G(PAP^T)$.

Example: P permutation that changes $3 \leftrightarrow 4$:

$G(A)$



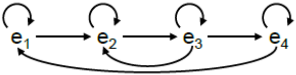
$G(PAP^T)$



4.2.2. Matrix A nonsymmetric, $G(A)$ directed

$$A = \begin{pmatrix} * & * & 0 & 0 \\ 0 & * & * & 0 \\ 0 & * & * & * \\ * & 0 & 0 & * \end{pmatrix} \Rightarrow \mathcal{A}(G(A)) = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \end{pmatrix}$$

G(A):



How can we characterize “good“ sparsity patterns?

”good”: Gaussian elimination can be reduced to smaller subproblems or produces no (or small) fill-in.

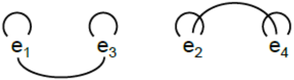
Block Diagonal Pattern

New matrix A!

$$A = \begin{pmatrix} * & 0 & * & 0 \\ 0 & * & 0 & * \\ * & 0 & * & 0 \\ 0 & * & 0 & * \end{pmatrix} \Rightarrow \mathcal{A}(G(A)) = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}$$

G(A):

$2 \leftrightarrow 3$:



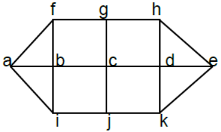
$$PAP^T = \begin{pmatrix} * & * & 0 & 0 \\ * & * & 0 & 0 \\ 0 & 0 & * & * \\ 0 & 0 & * & * \end{pmatrix} = \begin{pmatrix} A_1 & 0 \\ 0 & A_2 \end{pmatrix}$$

By this permutation, A can be transformed into block diagonal form $\rightarrow$ easy to solve!

$$\begin{pmatrix} A_1 & 0 \\ 0 & A_2 \end{pmatrix}^{-1} = \begin{pmatrix} A_1^{-1} & 0 \\ 0 & A_2^{-1} \end{pmatrix}$$

4.3. Reordering

Consider sparse matrix A with graph G(A):

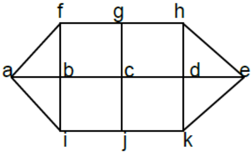


Consider following matrix (in tabular form) with bandwidth 9:

	a	b	c	d	e	f	g	h	i	j	k
a	a	*				*			*		
b	*	b	*			*			*		
c		*	c	*			*			*	
d			*	d	*			*			*
e				*	e			*			*
f	*	*				f	*				
g			*			*	g	*			
h				*	*		*	h			
i	*	*							i	*	
j			*						*	j	*
k				*	*					*	k

4.3.1. Cuthill McKee

Graph G(A):



Optimal ordering that leads to small bandwidth?

Level sets, starting with:

$S_1 := \{a\}$

$S_2 := \{f, b, i\}$, all vertices directly connected with S_1

$S_3 := \{g, c, j\}$, all vertices directly connected with S_2

$S_4 := \{h, d, k\}$, all vertices directly connected with S_3

$S_5 := \{e\}$

Cuthill McKee (cont.)

1. First ordering by level sets
2. Inside level sets order the vertices such that first group of indices in S_i are neighbors of first vertex in S_{i-1}
3. If there is choice left: number indices with small degree first!

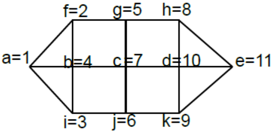
$S_1 := \{a\}$

$S_2 := \{f = 2, i = 3, b = 4\}$, (could also be different!)

$S_3 := \{g = 5, j = 6, c = 7\}$, (as we start with the neighbors of $f = 2$, then b, and then i)

$S_4 := \{h = 8, k = 9, d = 10\}$, (as we start with neighbors of g)

$S_5 := \{e = 11\}$



Cuthill McKee (cont. 2)

- Resulting matrix with small bandwidth 4:

$$
\begin{pmatrix}
1 & * & * & * & & & & & & & \\
* & 2 & 0 & * & * & & & & & & \\
* & 0 & 3 & * & 0 & * & & & & & \\
* & * & * & 4 & 0 & 0 & * & & & & \\
 & * & 0 & 0 & 5 & 0 & * & * & & & \\
 & & * & 0 & 0 & 6 & * & 0 & * & & \\
 & & & * & * & * & 7 & 0 & 0 & * & \\
 & & & & * & 0 & 0 & 8 & 0 & * & * \\
 & & & & & * & 0 & 0 & 9 & * & * \\
 & & & & & & * & * & * & 10 & * \\
 & & & & & & & * & * & * & 11
\end{pmatrix}
$$

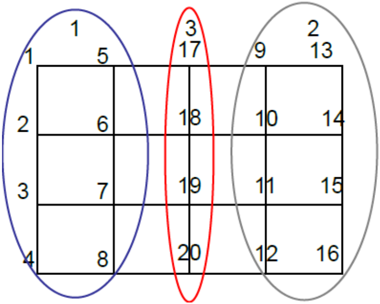
- Often „Reverse Cuthill Mckee“ gives slightly better results: reversing the ordering of Cuthill McKee: $11 \leftrightarrow 1$, $10 \leftrightarrow 2, \ldots$ Choose start as extreme vertex (diameter)!
- This method is far from being optimal and strongly heuristical.

4.3.2. Dissection Reordering

Consider matrix A with graph G(A):

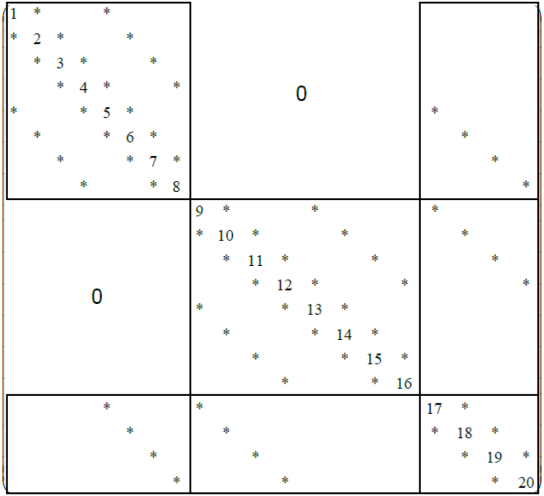
Numbering of unknows in two groups separated by third group.

group:



Dissection Reordering (cont.)

This numbering leads to sparsity pattern:



Dissection Reordering (cont. 2)

- Leads automatically to dissection form:

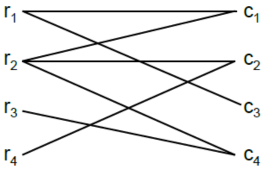
$$\begin{pmatrix} A_1 & 0 & F_1 \\ 0 & A_2 & F_2 \\ G_1 & G_2 & A_3 \end{pmatrix}$$

- Can be solved, e.g. based on Schur complement.
- General idea:
  - Cut $G(A)$ by separator between unconnected subgraphs.
  - Separator is numbered last.
  - Repeat recursively $\rightarrow$ Nested dissection form
- Looking for partitioning of the graph with minimum connections!

4.3.3. Perfect Matching Reordering

- Find row permutation of A such that elements with largest absolute values are located at diagonal.
- Advantage: No pivoting necessary (also in the indefinite case).
- Describe the sparsity pattern of A by bipartite graph $G = (V_r, V_c, E)$ with $V_r = \{r_1, \ldots, r_n\}$ and $V_c = \{c_1, \ldots, c_n\}$.
- Vertices r_i and c_j are connected by edge $\leftrightarrow$ $a_{i,j} \neq 0$:

$$A = \begin{pmatrix} * & 0 & * & 0 \\ * & * & 0 & 0 \\ 0 & 0 & 0 & * \\ 0 & * & 0 & 0 \end{pmatrix}$$

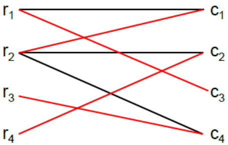


For each column we want to choose a large entry $\rightarrow$ diagonal.

Perfect Matching (cont.)

- Matching is subset of edges such that each row vertex is connected exactly to one column vertex and vice versa. Bijective mapping between V_r and V_c.

  Matching in our example:



- This matching induces row permutation to permute related entries $a_{1,3}$, $a_{2,1}$, $a_{3,4}$, and $a_{4,2}$ on the diagonal positions.

  Permutation:
  $1 \rightarrow 3,\ 2 \rightarrow 1,\ 3 \rightarrow 4,\ 4 \rightarrow 2$

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 1 & 4 & 2 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 3 & 4 & 2 \end{pmatrix}$$

$$A = \begin{pmatrix} * & 0 & * & 0 \\ * & * & 0 & * \\ 0 & 0 & 0 & * \\ 0 & * & 0 & 0 \end{pmatrix}$$

Perfect Matching (cont. 2)

- Add additional condition to matching problem: ”Find matching that maximizes a given function”
- Look for a subset of edges $M \leq E$:
  - where each vertex is incident to exactly one edge e in M and
  - where the matched edges maximize a weight function, e.g.

$$w(M) = \sum_{(i,j)\in M} c_{i,j} \qquad \text{with} \qquad c_{i,j} = \begin{cases} \log(|a_{i,j}|) & \text{for } a_{i,j} \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

- Solution of this perfect matching problem leads to permutation of A such that product of the new diagonal entries is maximized (therefore, all the diagonal entries should be large)
- Exact solution to costly, use heuristic approximate solutions!

Perfect Matching: Example

$$A = \begin{pmatrix} 100 & 0 & -1 & 0 & 0 \\ 1 & -2 & 0 & 0 & 110 \\ -1 & 0 & 0 & -120 & 0 \\ 0 & 1 & -80 & 0 & 0 \\ 2 & 90 & -2 & 1 & 1 \end{pmatrix}$$

$$\begin{pmatrix} 1 & & & & \\ & & & & 1 \\ & & & 1 & \\ & & 1 & & \\ & 1 & & & \end{pmatrix} \begin{pmatrix} 100 & 0 & -1 & 0 & 0 \\ 1 & -2 & 0 & 0 & 110 \\ -1 & 0 & 0 & -120 & 0 \\ 0 & 1 & -80 & 0 & 0 \\ 2 & 90 & -2 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 100 & 0 & -1 & 0 & 0 \\ 2 & 90 & -2 & 1 & 1 \\ 0 & 1 & -80 & 0 & 0 \\ -1 & 0 & 0 & -120 & 0 \\ 1 & -2 & 0 & 0 & 110 \end{pmatrix}$$

$$A = \begin{pmatrix} 100 & 90 & -100 & 80 & 70 \\ 1 & -2 & 0 & 0 & 10 \\ -1 & 0 & 0 & -20 & 0 \\ 0 & 1 & -8 & 0 & 0 \\ 2 & 9 & -2 & 1 & 1 \end{pmatrix} \rightarrow ?$$

Perfect Matching for Symmetric A

- For symmetric A row permutation would destroy symmetry! We need way to move large entries **near** the diagonal and permute rows and columns symmetrically!
- Idea: Solve perfect matching unsymmetrically $\rightarrow$ gives permutation.
- Resulting permutation can be written as sequence of cyclic permutations, e.g., we can rewrite the permutation in the following way

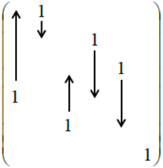
$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 2 & 4 & 5 & 1 & 3 & 6 \end{pmatrix} \Rightarrow (1 \quad 2 \quad 4)(3 \quad 5)(6)$$

Perfect Matching for Symmetric A (cont.)

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 2 & 4 & 5 & 1 & 3 & 6 \end{pmatrix} \Rightarrow (1 \quad 2 \quad 4)(3 \quad 5)(6)$$

Original row permutation gives

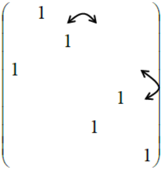
$P :=$



Reordering of columns in view of cyclic representation gives

Reordering such that the indices inside a cycle are ordered sequentially: $3 \leftrightarrow 4$

$P_c :=$



Perfect Matching for Symmetric A: Example

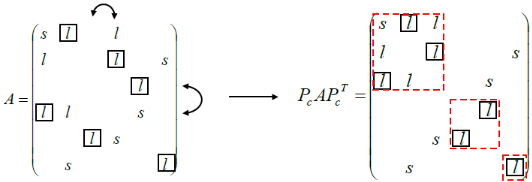
s: small entry, l: large entry, $\boxed{l}$: chosen entry

$$A = \begin{pmatrix} s & \boxed{l} & & l & & \\ l & & & \boxed{l} & & s \\ & & & & \boxed{l} & \\ \boxed{l} & l & & & s & \\ & & \boxed{l} & s & & \\ & s & & & & \boxed{l} \end{pmatrix}$$

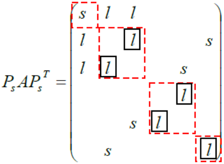
- Nonsymmetric perfect matching $P = (1, 2, 4)(3, 5)(6)$
- To allow symmetric permutation: switch $3 \leftrightarrow 4$.
- Change numbering in A accordingly.
- Then, the nonsymmetric perfect matching is $P = (1, 2, 3)(4, 5)(6)$ $\rightarrow$ block structure maintained under symmetric application of P!

Perfect Matching for Symmetric A: Example (cont.)

P_c from P by changing the order of the columns: $P_c : 3 \leftrightarrow 4$



P_s from P_c by changing the interpretation of the blocks:



PARDISO: fastest parallel direct solver (uses perfect matching).

4.4. Gaussian Elimination for Sparse Matrices

4.4.1. Algebraic Pivoting in GE

- Numerical pivoting: for eliminating elements in column k choose large(st) entry in column/row/block k and permute this element on the diagonal position.
- Disadvantage: may lead to large fill in in the sparsity pattern of A.
- Idea: Choose pivot element according to minimum fill in! Note that for well-conditioned $A = A^T > 0$ no numerical pivoting is necessary.
- Heuristic: Choose pivot element according to the degree in graph $\rightarrow$ minimum degree reordering

Special Case $A = A^T$

- For elimination in the kth column of A:
  - Define $r_m :=$ number of nonzero entries in row m
  - Choose pivot index i by $r_i = \min_m r_m$
  - Do the pivot permutation and the elimination
  - Go to next column k

- r_m is #nonzeros in the mth row = #vertices directly connected with vertex m
  Hence, pivot vertex is vertex with minimum degree in $G(A_k)$

- Heuristics: few entries in mth row/column $\rightarrow$ few fill in because
  - only few elements to eliminate
  - the pivot row used in the elimination is very sparse

  $\rightarrow$ Multiple minimum degree reordering

Generalization to Nonsymmetric Problems: Markowitz

- Define $r_m :=$ nnz in row m; $\quad c_p :=$ nnz in column p
- Choose pivot element with index pair (i, j) such that

$$(r_i - 1)(c_j - 1) = \min_{m,p}(r_m - 1)(c_p - 1)$$

- Heuristics:
  - small c_j leads to few elimination steps
  - small r_i leads to sparse pivot row used in the elimination.
- Special case $r_i = 1$ or $c_j = 1$: no fill in.
- Include numerical pivoting by applying algebraic pivoting only on indices with absolute value that is not to small, e.g.,

$$|a_{i,j}| \geq 0.1 \cdot \max_{r,s} |a_{r,s}|$$