

# MAT 226B Large Scale Matrix Computation

## Homework 1

Ahmed Mahmoud

January, 30th 2020

### Problem 1:

- (a) Let  $A = [a_{j,k}] \in \mathbb{R}^{n \times n} \succ 0$  and  $L = [l_{j,k}]$  be its Cholesky factor. Using MATLAB notation, Algorithm ?? shows Cholesky factorization algorithm

---

**Algorithm 1:** Cholesky Factorization

---

**Input:**  $A = [a_{j,k}] \in \mathbb{R}^{n \times n} \succ 0$   
**Output:**  $L = [l_{j,k}]$  such that  $A = LL^T$   
1  $l_{j,k} = a_{j,k}, \forall j \geq k$  and  $j, k = 1, 2, \dots, n$   
2 **for**  $k = 1, 2, \dots, n$  **do**  
3      $l_{k,k} = \sqrt{l_{k,k}}$   
4      $l_{k+1:n,k} = \frac{1}{l_{k,k}} l_{k+1:n,k}$   
5     **for**  $j = k + 1, k + 2, \dots, n$  **do**  
6          $l_{j:n,j} = l_{j:n,j} - l_{j:n,k} l_{jk}$   
7     **end**  
8 **end**

---

Line 1 in Algorithm ?? is a memory copy and does not include any flops. Line 3 accounts for  $n$  square root operations. On iteration  $k$ , Line 4 will account for  $n - k$  division operations. Since this loop goes from  $k = 1, 2, \dots, n$ , we get  $\sum_{k=1}^n (n - k) = \frac{1}{2}n(n - 1)$  division operation.

Line 6 does two operations; subtraction and multiplication, each on a vector of length  $(n - j + 1)$ . Thus, the total cost of the inner loop is

$$\sum_{k=1}^n \sum_{j=k+1}^n 2(n - j + 1) = \frac{1}{3}n(n^2 - 1)$$

Thus, the total cost of Algorithm ?? is

$$n + \frac{1}{2}n(n - 1) + \frac{1}{3}n(n^2 - 1) \text{ flops}$$

- (b) Let  $A$  be a banded  $n \times n$  matrix with bandwidth  $2p + 1$ , i.e.,  $a_{jk} = 0$  if  $|j - k| > p$ . To show that Cholesky factor  $L$  has lower bandwidth  $p$ , i.e.,  $l_{jk} = 0$  if  $j - k > p$ , we need to show the Cholesky factorization does not introduce any fill-in's. Line 3 and 4 in Algorithm ?? do not introduce any fill-in's.

At step  $k$ , the factor  $l_{jk}$  in Line 6 will be non-zero only for  $k \leq j \leq k + p$ . Thus, the only possible fill-in's for column  $j$  (i.e., at iteration  $j$  of the inner loop) is at the first  $p$  rows below the diagonal which are already non-zero since  $A$  is banded matrix with bandwidth  $2p + 1$  which means there is  $p$  non-zero rows below the diagonal already.

- (c) Cholesky factorization can be re-written more efficiently for banded matrices since it is guaranteed to *not* introduce fill-in such that computation can be skipped for the zero elements. Algorithm ?? shows Cholesky factorization algorithm for banded matrices

---

**Algorithm 2:** Cholesky Factorization for Banded Matrices

---

**Input:**  $A = [a_{j,k}] \in \mathbb{R}^{n \times n} \succ 0$  with bandwidth  $2p + 1$

**Output:**  $L = [l_{j,k}]$  such that  $A = LL^T$

```

1  $l_{j,k} = a_{j,k}, \forall j \geq k$  and  $j, k = 1, 2, \dots, n$ 
2 for  $k = 1, 2, \dots, n$  do
3    $l_{k,k} = \sqrt{l_{k,k}}$ 
4    $l_{k+1:k+p,k} = \frac{1}{l_{k,k}} l_{k+1:k+p,k}$ 
5   for  $j = k + 1, k + 2, \dots, k + p$  do
6      $l_{j:k+p,j} = l_{j:k+p,j} - l_{j:k+p,k} l_{j,k}$ 
7   end
8 end

```

---

In the algorithm above, we note the following

- Line 4 now only operates on the first  $p$  rows below the diagonal.
- Line 5 only goes through the first  $p$  columns after column  $k$  (during iteration  $k$ ) since the factor  $l_{jk}$  (Line 6) will be zero for  $j > k + p$
- Line 6 now only operates up to row  $k + p$  since the rows below  $k + p$  for column  $k$  (i.e., at iteration  $k$ ) will contain zeros.

- (d) Algorithm ?? requires  $n$  square root operations. Line 4 requires only  $p$  division. Since Line 4 runs for all  $k$  values (except for the last  $p$  columns but we ignore this corner case assuming  $p \ll n$ ), then the total number of division done by Line 4 is  $np$ .

Line 6 costs  $p$  subtraction and division (assuming  $p \ll n$ ). Thus the total cost of the whole loop (Line 5-7) is

$$\sum_{k=1}^n \sum_{j=k+1}^{k+p} 2p = 2np^2$$

Thus, the total cost of Algorithm ?? is

$$2np^2 + np + n =$$

## Problem 2:

- (a) Take an example for  $m = 5$  to see the pattern of how the non-zero values arise in the Cholesky factor  $L$

$$T_5 = \begin{bmatrix} 2 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & 2 \end{bmatrix} \Rightarrow L^1 = \begin{bmatrix} d_1 & 0 & 0 & 0 & 0 \\ h_1 & 2 - h_1^2 & 0 & 0 & 0 \\ 0 & -1 & 2 & 0 & 0 \\ 0 & 0 & -1 & 2 & 0 \\ 0 & 0 & 0 & -1 & 2 \end{bmatrix}$$

where  $d_1 = \sqrt{2}$  and  $h_1 = \frac{-1}{d_1}$

$$L^2 = \begin{bmatrix} d_1 & 0 & 0 & 0 & 0 \\ h_1 & d_2 & 0 & 0 & 0 \\ 0 & h_2 & 2 - h_2^2 & 0 & 0 \\ 0 & 0 & -1 & 2 & 0 \\ 0 & 0 & 0 & -1 & 2 \end{bmatrix}$$

where  $d_2 = \sqrt{2 - h_1^2}$  and  $h_2 = \frac{-1}{d_2}$

$$L^3 = \begin{bmatrix} d_1 & 0 & 0 & 0 & 0 \\ h_1 & d_2 & 0 & 0 & 0 \\ 0 & h_2 & d_3 & 0 & 0 \\ 0 & 0 & h_3 & 2 - h_3^2 & 0 \\ 0 & 0 & 0 & -1 & 2 \end{bmatrix}$$

where  $d_3 = \sqrt{2 - h_2^2}$  and  $h_3 = \frac{-1}{d_3}$

Following until the final step, we can see that the diagonal elements of Cholesky factor  $L$  are  $d_i = \sqrt{2 - h_{i-1}^2}$  and the lower diagonal elements  $h_i = \frac{-1}{d_i}$  where  $h_0 = 0$ .

- (b) From the problem description,  $T_{m \times m}$  is a banded matrix with bandwidth of  $2m + 1$ . In addition, each row and column in  $T_{m \times m}$  has at most 5 nonzero entries.

Let  $L$  be the Cholesky factor of  $T_{m \times m}$ . At the first step of Cholesky factorization (i.e.,  $k = 1$  in Algorithm ??) the first column will have the same structure but it will introduce a new fill-in in column 2 at row  $m + 1$  because  $l_{1,m+1}$  contains non-zero (namely -1) while  $l_{2,m+1}$  is zero and the operation  $l_{j,2} = l_{j,2} - l_{j,1} * l_{2,1}$  where  $j = m + 1$  will result in a nonzero. There is no more fill-in's will be introduced with  $k = 1$ .

Following with the same process for subsequent columns (i.e.,  $k > 1$ ), it is clear that there is at least a new fill-in will be introduced at entry  $l_{m+k,k+1}$  (actually there might be more than one new fill-in due to the fill-in's in previous columns). Thus, the sparsity structure of  $L$  is a banded matrix with lower bandwidth of  $m$  and all the entries  $l_{j,k} \neq 0$  when  $k - j \leq m$ .

### Problem 3:

Given the matrix description, it is possible to find a reordering (symmetric permutation matrix  $P$ ) that pushes all the non-zero rows to the bottom and all the non-zero columns to the right (except the diagonal entries). The resultant matrix is an *arrow* matrix with  $m$  non-zero (dense) rows and columns resides at the bottom-right corner. This is a generalization of the arrow matrix of  $m = 1$ .

As an example, we take  $A \in \mathcal{R}^{n \times n}$ ,  $m = 3$ ,  $n = 10$ , and  $\mathcal{I} = \{2, 6, 8\}$ .

$$A = \begin{bmatrix} x & x & 0 & 0 & 0 & x & 0 & x & 0 & 0 \\ x & x & x & x & x & x & x & x & x & x \\ 0 & x & x & 0 & 0 & x & 0 & x & 0 & 0 \\ 0 & x & 0 & x & 0 & x & 0 & x & 0 & 0 \\ 0 & x & 0 & 0 & x & x & 0 & x & 0 & 0 \\ x & x & x & x & x & x & x & x & x & x \\ 0 & x & 0 & 0 & 0 & x & x & x & 0 & 0 \\ x & x & x & x & x & x & x & x & x & x \\ 0 & x & 0 & 0 & 0 & x & 0 & x & x & 0 \\ 0 & x & 0 & 0 & 0 & x & 0 & x & 0 & x \end{bmatrix}$$

We can permute  $A$  with the following permutation matrix

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

And the result will be

$$P^T A P = \begin{bmatrix} x & 0 & 0 & 0 & 0 & 0 & 0 & x & x & x \\ 0 & x & 0 & 0 & 0 & 0 & 0 & x & x & x \\ 0 & 0 & x & 0 & 0 & 0 & 0 & x & x & x \\ 0 & 0 & 0 & x & 0 & 0 & 0 & x & x & x \\ 0 & 0 & 0 & 0 & x & 0 & 0 & x & x & x \\ 0 & 0 & 0 & 0 & 0 & x & 0 & x & x & x \\ 0 & 0 & 0 & 0 & 0 & 0 & x & x & x & x \\ x & x & x & x & x & x & x & x & x & x \\ x & x & x & x & x & x & x & x & x & x \\ x & x & x & x & x & x & x & x & x & x \end{bmatrix}$$

The Cholesky factor  $L$  of  $P^T AP$  will thus have the same sparsity structure as  $P^T AP$  as shown in Figure ?? for arbitrary  $n$  and  $m$ . Thus, the upper bound of the  $nnz(L) = (n-m) + m(n-m) + \frac{1}{2}m^2$ ; where  $n - m$  accounts for the diagonal elements (orange line),  $m(n - m)$  is the blue rectangle, and  $\frac{1}{2}m^2$  is the red triangle.

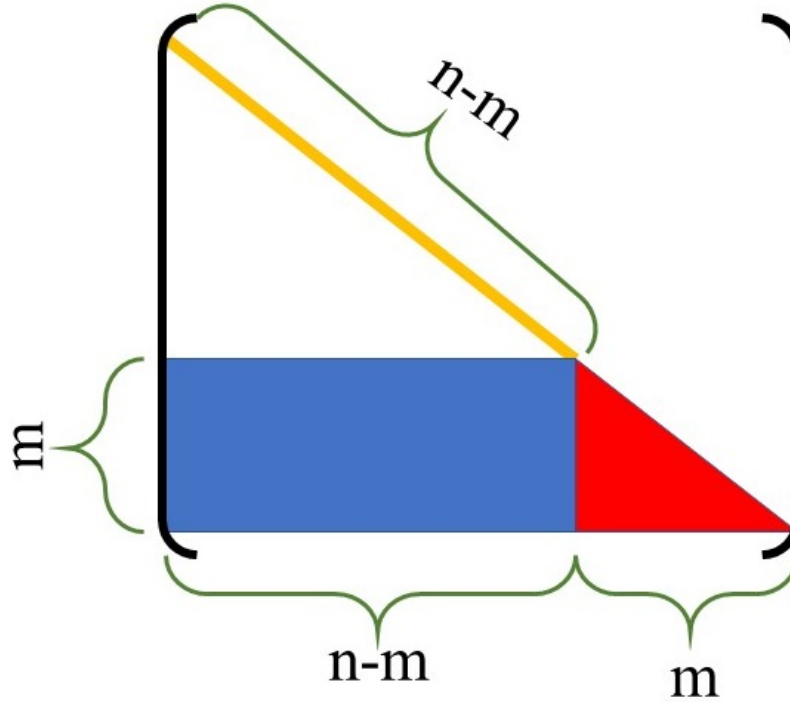


Figure 1: Sparsity structure of Cholesky factor  $L$

It is however possible to eliminate the non-zero in the red rectangle due to cancellation. This will give the lower bound of  $nnz(L)$  to be  $(n - m) + m(n - m)$ .

## Problem 4:

Figure ?? shows the associated graph  $G(A)$  of matrix  $A$  along with the steps of the minimum degree algorithm. From these steps, the reordering of the nodes will be 2, 4, 5, 3, 6, 7, 1, 8, 9

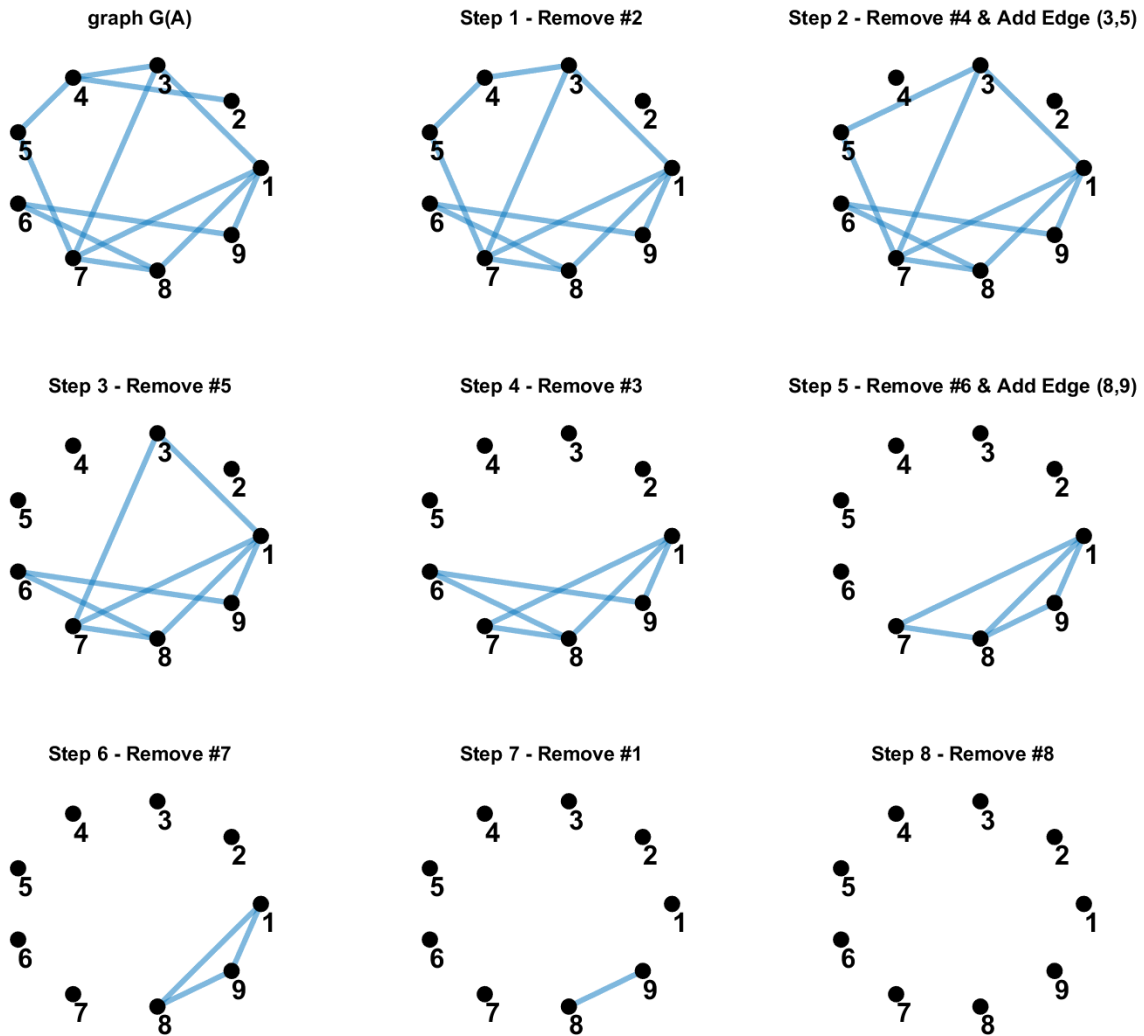


Figure 2: Graph  $G(A)$  along with the 8 steps of the minimum degree algorithm applied on it.

From the reordering above, the permutation matrix can be constructed such that

$$P = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

From which, we can compute  $P^T AP$  to be

$$P^T AP = \begin{bmatrix} * & * & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ * & * & * & * & 0 & 0 & 0 & 0 & 0 \\ 0 & * & * & 0 & 0 & * & 0 & 0 & 0 \\ 0 & * & 0 & * & 0 & * & * & 0 & 0 \\ 0 & 0 & 0 & 0 & * & 0 & 0 & * & * \\ 0 & 0 & * & * & 0 & * & * & * & 0 \\ 0 & 0 & 0 & * & 0 & * & * & * & * \\ 0 & 0 & 0 & 0 & * & * & * & * & 0 \\ 0 & 0 & 0 & 0 & * & 0 & * & 0 & * \end{bmatrix}$$

Applying Cholesky factorization to  $P^T AP$  we get the following lower triangular matrix where the fill-in elements are shown with +

$$L = \begin{bmatrix} * & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ * & * & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & * & * & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & * & + & * & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & * & 0 & 0 & 0 & 0 \\ 0 & 0 & * & * & 0 & * & 0 & 0 & 0 \\ 0 & 0 & 0 & * & 0 & * & * & 0 & 0 \\ 0 & 0 & 0 & 0 & * & * & * & * & 0 \\ 0 & 0 & 0 & 0 & * & 0 & * & + & * \end{bmatrix}$$



## Problem 5:

(a) The number of nonzero entries in  $L = 80$

Nonzero entries in  $L$  at column 4:

$$L(:, 4) = \begin{bmatrix} L_{4,4} = & 2.0000000000000000e+00 \\ L_{5,4} = & -5.0000000000000000e-01 \\ L_{9,4} = & -5.0000000000000000e-01 \\ L_{15,4} = & -5.0000000000000000e-01 \\ L_{16,4} = & -5.0000000000000000e-01 \end{bmatrix}$$

Nonzero entries in  $L$  at column 7:

$$L(:, 7) = \begin{bmatrix} L_{7,7} = & 1.921537845661046e+00 \\ L_{8,7} = & -4.003203845127178e-02 \\ L_{9,7} = & -4.003203845127178e-02 \\ L_{11,7} = & -8.006407690254357e-02 \\ L_{12,7} = & -5.604485383178049e-01 \\ L_{15,7} = & -8.006407690254357e-02 \\ L_{16,7} = & -5.604485383178049e-01 \end{bmatrix}$$

Nonzero entries in  $L$  at column 10:

$$L(:, 10) = \begin{bmatrix} L_{10,10} = & 1.9317955145497680e+00 \\ L_{11,10} = & -5.581787456852771e-01 \\ L_{12,10} = & -5.821358759653915e-03 \\ L_{13,10} = & -2.238984138328429e-04 \\ L_{14,10} = & -1.388170165763626e-01 \\ L_{15,10} = & -4.097340973141025e-02 \\ L_{16,10} = & -6.269155587319600e-03 \end{bmatrix}$$

Nonzero entries in  $L$  at column 13:

$$L(:, 13) = \begin{bmatrix} L_{13,13} = & 1.9314855917443310e+00 \\ L_{14,13} = & -5.180834914653560e-01 \\ L_{15,13} = & -1.515180286251492e-01 \\ L_{16,13} = & -2.197481923194983e-02 \end{bmatrix}$$

Nonzero entries in  $L$  at column 16:

$$L(:, 16) = [L_{16,16} = \quad 1.717283221311613e+00]$$

The relative error is

$$\frac{\|A - LL^T\|_2}{\|L\|_2} = 3.513060e - 16$$

(b) **medium\_ex1.mat** ( $A \in \mathbb{R}^{1600 \times 1600}$ )

**No ordering:**

- $nnz(L) = 64039$
- $flag = 0$
- $k = 1600$
- Ten largest entries in  $l_{j,1000} =$

$$\begin{bmatrix} L_{1000,1000} = & 1.818644\text{e}+00 \\ L_{1040,1000} = & 5.498601\text{e}-01 \\ L_{1039,1000} = & 1.903980\text{e}-01 \\ L_{1038,1000} = & 7.649394\text{e}-02 \\ L_{1037,1000} = & 3.549975\text{e}-02 \\ L_{1036,1000} = & 1.867330\text{e}-02 \\ L_{1035,1000} = & 1.086313\text{e}-02 \\ L_{1034,1000} = & 6.832166\text{e}-03 \\ L_{1033,1000} = & 4.562541\text{e}-03 \\ L_{1032,1000} = & 3.192202\text{e}-03 \end{bmatrix}$$

**Reordering with symamd**

- $nnz(L) = 20296$
- $flag = 0$
- $k = 1600$
- Ten largest entries in  $l_{j,1000} =$

$$\begin{bmatrix} L_{1000,1000} = & 1.657056\text{e}+00 \\ L_{1064,1000} = & 3.344285\text{e}-01 \\ L_{1067,1000} = & 3.344285\text{e}-01 \\ L_{1066,1000} = & 2.225333\text{e}-01 \\ L_{1061,1000} = & 2.225333\text{e}-01 \\ L_{1068,1000} = & 1.508700\text{e}-01 \\ L_{1011,1000} = & 3.897475\text{e}-02 \\ L_{1062,1000} = & 3.897475\text{e}-02 \\ L_{1012,1000} = & 2.514500\text{e}-02 \\ L_{1063,1000} = & 2.514500\text{e}-02 \end{bmatrix}$$

### Reordering with **colamd**

- $nnz(L) = 32451$
- $flag = 0$
- $k = 1600$
- Ten largest entries in  $l_{j,1000} =$

$$\begin{bmatrix} L_{1000,1000} = & 1.793525\text{e}+00 \\ L_{1001,1000} = & 6.537247\text{e}-01 \\ L_{1003,1000} = & 5.575611\text{e}-01 \\ L_{1002,1000} = & 2.364628\text{e}-01 \\ L_{1005,1000} = & 3.913581\text{e}-02 \\ L_{1007,1000} = & 3.178591\text{e}-02 \\ L_{1032,1000} = & 2.481432\text{e}-02 \\ L_{1036,1000} = & 2.367906\text{e}-02 \\ L_{1034,1000} = & 1.416474\text{e}-02 \\ L_{1038,1000} = & 9.514311\text{e}-03 \end{bmatrix}$$

### Reordering with **symrcm**

- $nnz(L) = 45020$
- $flag = 0$
- $k = 1600$
- Ten largest entries in  $l_{j,1000} =$

$$\begin{bmatrix} L_{1000,1000} = & 1.792960\text{e}+00 \\ L_{1034,1000} = & 6.331742\text{e}-01 \\ L_{1035,1000} = & 5.577368\text{e}-01 \\ L_{1001,1000} = & 2.420380\text{e}-01 \\ L_{1033,1000} = & 1.009250\text{e}-01 \\ L_{1002,1000} = & 4.826873\text{e}-02 \\ L_{1032,1000} = & 3.746352\text{e}-02 \\ L_{1003,1000} = & 1.954431\text{e}-02 \\ L_{1031,1000} = & 1.865257\text{e}-02 \\ L_{1030,1000} = & 1.080417\text{e}-02 \end{bmatrix}$$

### Reordering with `colperm`

- $nnz(L) = 169301$
- $flag = 0$
- $k = 1600$
- Ten largest entries in  $l_{j,1000} =$

$$\begin{bmatrix} L_{1000,1000} = & 1.792811\text{e}+00 \\ L_{1001,1000} = & 6.493278\text{e}-01 \\ L_{1038,1000} = & 5.577833\text{e}-01 \\ L_{1037,1000} = & 2.018610\text{e}-01 \\ L_{1036,1000} = & 8.587340\text{e}-02 \\ L_{1035,1000} = & 4.232677\text{e}-02 \\ L_{1002,1000} = & 4.206819\text{e}-02 \\ L_{1034,1000} = & 2.337139\text{e}-02 \\ L_{1003,1000} = & 2.248362\text{e}-02 \\ L_{1033,1000} = & 1.386363\text{e}-02 \end{bmatrix}$$

**medium\_ex2.mat** ( $A \in \mathbb{R}^{1728 \times 1728}$ )

**No ordering:**

- $nnz(L) = 231419$
- $flag = 0$
- $k = 1728$
- Ten largest entries in  $l_{j,1000} =$

$$\begin{bmatrix} L_{1000,1000} = & 2.314829\text{e}+00 \\ L_{1001,1000} = & 4.761016\text{e}-01 \\ L_{1144,1000} = & 4.319974\text{e}-01 \\ L_{1132,1000} = & 9.419033\text{e}-02 \\ L_{1143,1000} = & 8.877675\text{e}-02 \\ L_{1131,1000} = & 4.015705\text{e}-02 \\ L_{1120,1000} = & 2.484508\text{e}-02 \\ L_{1133,1000} = & 2.205327\text{e}-02 \\ L_{1142,1000} = & 2.035099\text{e}-02 \\ L_{1119,1000} = & 1.557268\text{e}-02 \end{bmatrix}$$

**Reordering with symamd**

- $nnz(L) = 76723$
- $flag = 0$
- $k = 1728$
- Ten largest entries in  $l_{j,1000} =$

$$\begin{bmatrix} L_{1000,1000} = & 2.155675\text{e}+00 \\ L_{1002,1000} = & 2.348857\text{e}-01 \\ L_{1005,1000} = & 2.300113\text{e}-01 \\ L_{1485,1000} = & 1.991918\text{e}-01 \\ L_{1001,1000} = & 1.615753\text{e}-01 \\ L_{1490,1000} = & 1.054859\text{e}-01 \\ L_{1533,1000} = & 5.035958\text{e}-02 \\ L_{1003,1000} = & 4.928256\text{e}-02 \\ L_{1532,1000} = & 3.999123\text{e}-02 \\ L_{1006,1000} = & 3.973150\text{e}-02 \end{bmatrix}$$

### Reordering with **colamd**

- $nnz(L) = 145731$
- $flag = 0$
- $k = 1728$
- Ten largest entries in  $l_{j,1000} =$

$L_{1000,1000} =$	$2.414184e+00$
$L_{1117,1000} =$	$4.267708e-01$
$L_{1001,1000} =$	$4.142187e-01$
$L_{1002,1000} =$	$4.142187e-01$
$L_{1165,1000} =$	$4.142187e-01$
$L_{1006,1000} =$	$7.322048e-02$
$L_{1012,1000} =$	$7.322048e-02$
$L_{1004,1000} =$	$7.112846e-02$
$L_{1119,1000} =$	$1.255208e-02$
$L_{1044,1000} =$	$4.184027e-03$

### Reordering with **symrcm**

- $nnz(L) = 143595$
- $flag = 0$
- $k = 1728$
- Ten largest entries in  $l_{j,1000} =$

$L_{1000,1000} =$	$2.371271e+00$
$L_{1098,1000} =$	$4.362329e-01$
$L_{1109,1000} =$	$4.217148e-01$
$L_{1110,1000} =$	$4.217148e-01$
$L_{1001,1000} =$	$9.197598e-02$
$L_{1012,1000} =$	$8.871207e-02$
$L_{1011,1000} =$	$8.117340e-02$
$L_{1099,1000} =$	$1.654299e-02$
$L_{1088,1000} =$	$1.568771e-02$
$L_{1013,1000} =$	$8.948869e-03$

### Reordering with `colperm`

- $nnz(L) = 337092$
- $flag = 1$
- $k = 682$
- Four largest entries (the rest is all zero) in  $l_{j,1000} =$

$$\begin{bmatrix} L_{1000,1000} = & 6.000000\text{e}+00 \\ L_{1001,1000} = & 1.000000\text{e}+00 \\ L_{1010,1000} = & 1.000000\text{e}+00 \\ L_{1100,1000} = & 1.000000\text{e}+00 \end{bmatrix}$$