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 1 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 $j, k = 1, 2, \ldots, n$
2 **for** $k = 1, 2, \ldots, n$ **do**
3 $\quad l_{k,k} = \sqrt{l_{k,k}}$
4 $\quad l_{k+1:n,k} = \frac{1}{l_{k,k}} l_{k+1:n,k}$
5 $\quad$ **for** $j = k+1, k+2, \ldots, n$ **do**
6 $\quad\quad l_{j:n,j} = l_{j:n,j} - l_{j:n,k} l_{jk}$
7 $\quad$ **end**
8 **end**

Line 1 in Algorithm 1 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 $k = 1, 2, \ldots, n$, we get $\sum_{i=1}^{n}(n-k) = \frac{1}{2}n(n-1)$ 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 1 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 1 do not introduce any fill-in's. Line 6 can be re-written (in scalar notation instead of MATLAB's vector notation) as $l_{ij} = l_{ij} - \frac{l_{i,k}}{l_{k,k}}$ and $i = j, j + 1 \ldots, n$

(c)

Problem 2:

Problem 3:

Problem 4:

Figure 1 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 $2, 4, 5, 3, 6, 7, 1, 8, 9$

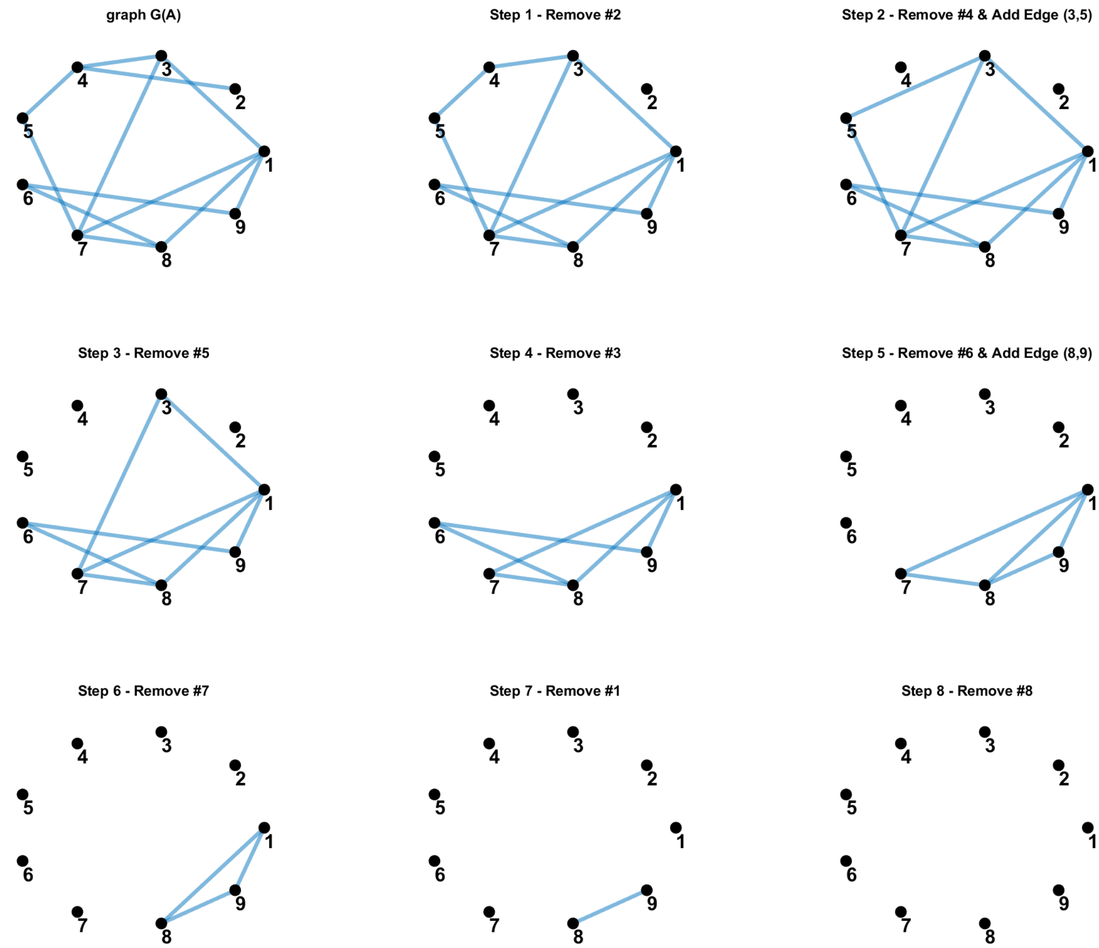

Figure 1: Graph $G(A)$ along with the 8 steps of the minimum degree algorithm applied on it.

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: