Math 170A HW2

Neo Lee

04/14/2023

Problem 1.

a) Let $U\vec{x} = \vec{b}$ be

$$\begin{bmatrix} u_{1,1} & u_{1,2} & \cdots & u_{1,n} \\ 0 & u_{2,2} & \cdots & u_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & u_{n,n} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

Then, we can perform backward substitution, starting from solving x_n with row n. The exact formula would be $x_i = \frac{1}{u_{i,i}} \left(b_i - \sum_{k=i+1}^{n} x_k \cdot u_{i,k} \right)$ for $i \in [1, n]$.

b)

solve_upper_tri.m

```
function x = solve_upper_tri(U,b)

x = b;
n = size(b,1);

x(end) = b(end) / U(end, end);

for i = (n-1):-1:1
    for k = n:-1:(i+1)
        x(i) = x(i) - x(k) * U(i, k);
    end

    x(i) = x(i) / U(i,i);
end

end
```

c) For $i \in [1, n]$, there are $n - i$ multiplications, $n - i$ subtractions, and 1 division. Therefore, there are a total of $\sum_{i=1}^{n} [2(n-i)+1] = 2n^2 + n - 2\sum_{i=1}^{n} i = 2n^2 + n - 2 \times \frac{(1+n)n}{2} = 2n^2 + n - n - n^2 = n^2$ flops.

Problem 2.

a) Let $LA = U$, c be some constant, and

$$L = \begin{bmatrix} 1 & c & \cdots & c \\ m_{2,1} & 1 & \cdots & c \\ \vdots & \vdots & \ddots & \vdots \\ m_{n,1} & m_{n,2} & \cdots & 1 \end{bmatrix}.$$

Then, for each row $i \in [1, n]$ and column $k \in [i, n]$ in U, $u_{i,k} = a_{i,k} + \sum_{j \in [1, i-1]} m_{i,j} \times a_{j,k} + C$, in which $a_{i,k}$ is basically the row i and the summation term is basically the elementary transformation of summing all j rows times the multipliers m.

b)

Problem 3.

a) Let $L_1 \cdot L_2 = L_3$. Then, $l_3^{(i,j)} = \sum_{k=1}^{n} l_1^{(i,k)} \times l_2^{(k,j)}$. Note that for $i < k$, $l_1^{(i,k)} = 0$ and for $k < j$, $l_2^{(k,j)} = 0$. Then, when $i < j$, for all $k \in [1, n]$, $i < k$ or $k < j$ must be true $\Rightarrow l_1^{(i,k)} \times l_2^{(k,j)} = 0 \Rightarrow l_3^{i,j} = 0 \Rightarrow$ all the entries above diagonal are 0 $\Rightarrow L_3$ is lower triangular.

b) Let arbitrary lower triangular matrix L_1, L_2, L_3. $L_1 \cdot L_2 = L_3 \Leftrightarrow (L_1 \cdot L_2)^T = L_3^T \Leftrightarrow L_2^T \cdot L_1^T = L_3^T \Leftrightarrow U_2 \cdot U_1 = U_3$. Hence, we get any arbitrary upper triangular matrix multiplication would get an upper triangular matrix.

Problem 4.

Proposition 1. *LU factorization is unique.*

Proof. Assume to the contrary that there exists $LU = \tilde{L}\tilde{U}$ such that $L \neq \tilde{L}$ and $U \neq \tilde{U}$. Then, $LU = \tilde{L}\tilde{U} \Leftrightarrow L = \tilde{L}\tilde{U}U^{-1} \Leftrightarrow \tilde{L}^{-1}L = \tilde{U}U^{-1}$. Since inverse of a lower triangular matrix is a lower triangular matrix, and the same holds for upper triangular matrix, we can see that the left hand side of the equation is a lower triangular matrix while the right hand side is an upper triangular matrix, which is impossible. Hence, contradiction is reached, and the proposition is proved. $\square$

Problem 5.

```
swap.m
function B = swap(A, i, j)

B = A;
B(i,:) = A(j,:);
B(j,:) = A(i,:);

end
```

swap.m

```
max_entry.m
function index = max_entry(A, i)

n = size(A,1);
maxi = abs(A(i,i));
index = i;
count = i;

for k = A(i:n, i)'
    if (abs(k) > maxi)
        maxi = abs(k);
        index = count;
    end
    count = count + 1;
end

end
```

max_entry.m