Justin Hexem
AMATH 584

Homework #6

21.6.) Proof: Let $A \in \mathbb{C}^{m\times m}$ such that $\sum_{j\neq k} |a_{jk}| < |a_{kk}|$ and let $\hat{A} = L_1 P_1 A$ be the matrix obtained after the first step of the Gaussian elimination with partial pivoting algorithm. Note that $|a_{11}| > |a_{i1}|$ for all $2 \leq i \leq m$, so $P_1 = I$ and $\hat{A} = L_1 A$. Show that if Gaussian elimination with partial pivoting is applied to A, no row interchanges take place. Observe that if we can show the $k{:}m \times k{:}m$ submatrix of $\hat{A}$ is strictly column diagonally dominant, then the largest element in the kth column of this submatrix must always lie at the kth row, kth column, which means no row swaps would be needed. Let this submatrix be called $\tilde{A}$. Then by algorithm 20.1 we see that $\tilde{a}_{j,1:m} = a_{j,1:m} - \left(\frac{a_{j1}}{a_{11}}\right) a_{1,1:m}$, so if $2 \leq k \leq m$ then $\tilde{a}_{jk} = a_{jk} - \frac{a_{j1}}{a_{11}} a_{1k}$. This means

$$\sum_{\substack{j\neq k \\ j>1}} |\tilde{a}_{jk}| = \sum_{\substack{j\neq k \\ j>1}} \left| a_{jk} - \frac{a_{j1}}{a_{11}} a_{1k} \right| \leq \sum_{\substack{j\neq k \\ j>1}} |a_{jk}| + \sum_{\substack{j\neq k \\ j>1}} \left| \frac{a_{1k}}{a_{11}} a_{j1} \right|$$

$$= \sum_{\substack{j\neq k \\ j>1}} |a_{jk}| + \frac{|a_{1k}|}{|a_{11}|} \sum_{\substack{j\neq k \\ j>1}} |a_{j1}| \qquad \left(\text{Note } \sum_{\substack{j\neq k \\ j>1}} |a_{j1}| + |a_{k1}| = \sum_{j\neq 1} |a_{j1}| < |a_{11}|\right)$$

$$< |a_{kk}| - |a_{1k}| + \frac{|a_{1k}|}{|a_{11}|} \left(|a_{11}| - |a_{k1}| \right)$$

$$= |a_{kk}| - |a_{1k}| + |a_{1k}| - |a_{1k}| \frac{|a_{k1}|}{|a_{11}|}$$

$$= |a_{kk}| - |a_{1k}| \frac{|a_{k1}|}{|a_{11}|} = |a_{kk}| - \left| a_{1k} \frac{a_{k1}}{a_{11}} \right|$$

$$\leq \left| a_{kk} - a_{1k} \frac{a_{k1}}{a_{11}} \right| = |\tilde{a}_{kk}|$$

Thus $\tilde{A}$ is also strictly column diagonally dominant after each step of Gaussian elimination is applied, and the top left element $\tilde{a}_{ii}$ satisfies $|\tilde{a}_{ji}| < |\tilde{a}_{ii}|$ for all $j > i$. Therefore no row swaps take place when we apply Gaussian elimination with partial pivoting. ☒

2.) a.) Proof: Let $A \in \mathbb{C}^{m\times m}$ be invertible, $B \in \mathbb{C}^{m\times m}$ be a rank-k matrix such that $B = XCY^T$ with $X \in \mathbb{C}^{m\times k}$, $Y \in \mathbb{C}^{k\times m}$, and $C \in \mathbb{C}^{k\times k}$ where C is invertible. Show $(A+XCY)^{-1} = A^{-1} - A^{-1}X(C^{-1}+YA^{-1}X)^{-1}YA^{-1}$. Observe that we can solve the following system for T in two different ways:

① Substitution:

$$\begin{bmatrix} A & X \\ Y & -C^{-1} \end{bmatrix}\begin{bmatrix} T \\ R \end{bmatrix} = \begin{bmatrix} I \\ 0 \end{bmatrix} \Rightarrow \begin{matrix} AT+XR = I \\ YT - C^{-1}R = 0 \end{matrix}$$

$$YT - C^{-1}R = 0 \Rightarrow YT = C^{-1}R \Rightarrow R = CYT$$

$$\Rightarrow AT + X(CYT) = I \Rightarrow AT + XCYT = I \Rightarrow (A+XCY)T = I$$

$$\Rightarrow T = (A+XCY)^{-1}$$

② Gaussian elimination:

From the lecture 20 notes we see that

$$\begin{bmatrix} A & X \\ Y & -C^{-1} \end{bmatrix}\begin{bmatrix} T \\ R \end{bmatrix} = \begin{bmatrix} I \\ 0 \end{bmatrix} \Rightarrow \begin{bmatrix} I & 0 \\ -YA^{-1} & I \end{bmatrix}\cdot\begin{bmatrix} A & X \\ Y & -C^{-1} \end{bmatrix}\begin{bmatrix} T \\ R \end{bmatrix} = \begin{bmatrix} I & 0 \\ -YA^{-1} & I \end{bmatrix}\cdot\begin{bmatrix} I \\ 0 \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} A & X \\ 0 & -(C^{-1}+YA^{-1}X) \end{bmatrix}\begin{bmatrix} T \\ R \end{bmatrix} = \begin{bmatrix} I \\ -YA^{-1} \end{bmatrix} \Rightarrow \begin{matrix} AT+XR = I \\ -(C^{-1}+YA^{-1}X)R = -YA^{-1} \end{matrix}$$

$$-(C^{-1}+YA^{-1}X)R = -YA^{-1} \Rightarrow R = (C^{-1}+YA^{-1}X)^{-1}YA^{-1}$$

$$\Rightarrow AT + X(C^{-1}+YA^{-1}X)^{-1}YA^{-1}) = I \Rightarrow AT + X(C^{-1}+YA^{-1}X)^{-1}YA^{-1} = I$$

$$\Rightarrow T + A^{-1}X(C^{-1}+YA^{-1}X)^{-1}YA^{-1} = A^{-1}$$

$$\Rightarrow T = A^{-1} - A^{-1}X(C^{-1}+YA^{-1}X)^{-1}YA^{-1}$$

Thus we obtain $(A+XCY)^{-1} = A^{-1} - A^{-1}X(C^{-1}+YA^{-1}X)^{-1}YA^{-1}$. ☒

2.) b.) Pseudocode for $(C+ab^T)x=v$.

1.) Calculate the diagonal entries of Λ^{-1} where Λ^{-1} is the inverse of Λ in $FCF^{-1}=\Lambda$. This is done by calculating Fc where c is the first column of C, and then taking the reciprocals of Fc.

2.) Calculate $C^{-1}v$ and $C^{-1}a$ by calculating $F^{-1}\Lambda^{-1}Fv$ and $F^{-1}\Lambda^{-1}Fa$.

3.) Calculate $x=(C+ab^T)^{-1}v$ using the Sherman-Morrison Woodbury formula with $A=C$, $X=a$, $C=1$, and $Y=b^T$.

4.) Return x.

This requires 1 fft for step 1 and 2 ffts and 2 iffts for step 2, so there are 5 total fft calculations. Step 1 is $\Theta(m\log m)$, step 2 is $\Theta(m\log m)$, step 3 is $\Theta(m)$, and step 4 is $\Theta(1)$, so the overall computational complexity is $\Theta(m\log m)$.

3.) a.) Let $M=A+B$ where $B=\begin{bmatrix} \uparrow & & & \uparrow \\ b_1 & 0 & \cdots & b_2 \\ \downarrow & & & \downarrow \end{bmatrix}$.

We see that if $X=[b_1\ b_2]$ and $Y=[e_1\ e_m]$, then

$$B=X\,I_{2\times 2}\,Y.$$

By inspection, B only has two nonzero columns, so if b_1 & b_2 are linearly independent then $\text{rank}(B)=2$. If b_1 & b_2 are linearly dependent, then $\text{rank}(B)=1$, so the maximum value of $\text{rank}(B)$ is 2.

b.) Pseudocode for $M(t)x(t)=f(t)$.

1.) Compute the LU decomposition of A

2.) Loop through all times t_k

3.) Call stepsolve function in each iteration, where stepsolve does the following:

3.)b.) continued.)

(i.) Compute $A^{-1}f(t_k)$ by using the LU decomp. of A.

(ii.) Form $X(t_k)$ and $Y(t_k)$ where $B(t_k) = X(t_k) I_{2\times 2} Y(t_k)^T$.

(iii.) Compute $A^{-1}X(t_k)$ by using the LU decomp. of A.

(iv.) Use the Sherman-Morrison Woodbury formula to calculate x_k and return x_k

The time complexity of step 1 is $\Theta(m^3)$, step 3 is contained in the loop in step 2, and the loop occurs p times. For each iteration of the loop, step (i.) is $\Theta(m^2)$, step (ii) is $\Theta(m)$, step (iii) is $\Theta(m^2)$, and step (iv) is $\Theta(m)$. This means stepsolve is $\Theta(m^2)$, so since it occurs p times the time complexity of the loop is $\Theta(m^2 p)$. The total time complexity then becomes $\Theta(m^3 + m^2 p)$.