LINEAR ALGEBRA: ANSWER TO HOMEWORK 6

VISHVAS VASUKI

1. 21.5

A is hermetian, not necessarily positive definite.

1.1. a.

A strategy of symmetric pivoting to preserve the hermitian structure while still leading to a unit lower triangular matrix with entries $|l_{i,j}| \leq 1$ is as follows:

Let $A_{i,i}$ represent the submatrix of A with the first i-1 rows and columns removed.

Consider the symmetric matrix $A_{i-1,i-1}$. For our strategy, we will use partial pivoting.

Suppose that no row exchange is necessary. Then, $a_{j,k} = a_{j,k} - \frac{a_{j,i-1}}{a_{i-1,i-1}} a_{i-1,k} = a_{k,j} - \frac{a_{i-1,j} a_{i-1,k}}{a_{i-1,i-1}}$, which is the new $a_{k,j}$. Hence, for elements not involved in a row exchange, symmetry is preserved.

If necessary, we exchange rows a^*_{h+i-1} and a^*_{i-1}. After this step of elimination, the elements in the hth row of $A_{i-1,i-1}$ may no longer match the elements of the hth column of $A_{i-1,i-1}$. For other elements of $A_{i,i}$ are still symmetric, as argued earlier.

Now, we consider restoring the symmetry of the affected elements after the row exchange and the elimination step. The new elements in the exchanged row can be specified in terms of the old elements of $A_{i-1,i-1}$: $a_{h+i-1,k} = a_{i-1,k} - \frac{a_{h+i-1,k} a_{i-1,i-1}}{a_{h+i-1,i-1}}$. Similarly, elements in the affected column are of the form:
$a_{k,h+i-1} = a_{k,h+i-1} - \frac{a_{h+i-1,h+i-1} a_{k,i-1}}{a_{h+i-1,i-1}}$. [**Incomplete**.]

1.2. b.

The form of the matrix factorization computed by this algorithm:
[**Incomplete**.]

1.3. c.

Its asymptotic operation count is : [**Incomplete**.]

2. 21.6

Theorem 2.0.1. *A is strictly comlumn diagonally dominant. So, for each k; $|a_{k,k}| > \sum_{j \neq k} |a_{j,k}|$. If GEPP is applied to A, no row exchanges take place.*

Proof.

Notation. Let $A_{i,i}$ represent the submatrix of A with the first i-1 rows and columns removed. Let $a'_{i,j}$ represent the element at row i and column j just after GEPP has filled the appropriate entries of a_1 with 0.

$a_{1,1}$ is the largest in a_1. So, there is no row exchange in the first step of GEPP.

We now show that, after the first elimination step of GEPP, the matrix $A_{2,2}$ is still column diagonally dominant.

Consider any column $a_k \neq a_1$ just after zeros have been filled in a_1.

Before A has been altered by GEPP, we observe the following relationship:

$$\begin{aligned}
|a_{k,k}| &> \sum_{j \notin \{k,1\}} |a_{j,k}| + |a_{1,k}| \\
&> \sum_{j \notin \{k,1\}} |a_{j,k}| + |a_{1,k}| \frac{\sum_{j \neq 1} |a_{j,1}|}{|a_{1,1}|} \\
&= \sum_{j \notin \{k,1\}} (|a_{j,k}| + |a_{1,k}| \frac{|a_{j,1}|}{|a_{1,1}|}) + |a_{1,k}| \frac{|a_{k,1}|}{|a_{1,1}|} \\
|a_{k,k}| - |a_{1,k} \frac{a_{k,1}}{a_{1,1}}| &> \sum_{j \notin \{k,1\}} (|a_{j,k}| + |a_{1,k} \frac{a_{j,1}}{a_{1,1}}|) \\
|a_{k,k} - a_{1,k} \frac{a_{k,1}}{a_{1,1}}| &> \sum_{j \notin \{k,1\}} (|a_{j,k} - a_{1,k} \frac{a_{j,1}}{a_{1,1}}|) \\
a'_{k,k} &> \sum_{j \notin \{k,1\}} a'_{j,k}
\end{aligned}$$

Thus, $A_{2,2}$ is still column diagonally dominant. Thus, by induction we see that no row exchange is required. □

3

Consider matrices $W_n(x)$ of the form:

$$W_5(x) = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ -1 & 1 & 0 & 0 & 1 \\ -1 & -1 & 1 & 0 & 1 \\ -1 & -1 & -1 & 1 & 1 \\ -1 & -1 & -1 & -1 & 1+x \end{pmatrix}, \quad 0 \leq x \leq 2.$$

Remark 3.0.2. Answers to b and c should use pen and paper only (but feel free to experiment in MATLAB).

3.1. **a.** What is the condition number (use "rcond" in MATLAB) of $W_n(x)$ when $n = 25, 50, 100$ and $x = 0, 1, 2$?

For n = 25 and $x = 0, 1, 2$, we get condition numbers $0.0400, 0.0256, 0.0185$. For n = 50 and $x = 0, 1, 2$, we get condition numbers $0.0200, 0.0131, 0.0096$. For n = 100 and $x = 0, 1, 2$, we get condition numbers $0.0100, 0.0099, 0.0098$.

3.2. **b.** What is the LU decomposition of $W_n(x)$?

L is a lower traingular matrix with 1's on the diagonal and -1's everywhere below the diagonal.

U is an upper triangualr matrix, with $U_{i,i} = 1$ for $i \in [1, m-1]$; $U_{1:m-1,m}$ being $2^0, ..2^{m-2}$, with $U_{m,m} = 2^{m-1} + x$ and with 0's everywhere else.

Suppose that GEPP uses the following tie breaking strategy in finding pivots during step k: Chose the element with the minimal row index as the pivot. Then, Partial pivoting does not make any difference.

3.3. **c.**

Theorem 3.3.1. *Consider single precision arithmetic in which $fl(2^{24}+1) = 2^{24}$. The smallest value of n for which Gaussian Elimination will return the solution of $W_n(0)y = e_1$, when we asked for $W_n(1)y = e_1$ is n=25.*

Proof. Note that $L^{-1}e_1$ always results in a column vector which has 1's everywhere.

Consider the case where n = 24. Then, during all steps of Gaussian elimination, the most disparate two numbers added or subtracted are 1 and 2^{23}. This operation is not guaranteed to result in round off errors.

However, consider the case when n=25. When U is produced, $U_{n,n} = fl(2^{24} + 1) = 2^{24}$. Thus, we get the U we would have got, had we started from $W_n(0)$ rather than $W_n(1)$. Hence, we compute the result we could have computed if we had started with $W_n(0)$. □

Remark 3.3.2. In IEEE double precision arithmetic, we know that $\epsilon_M = 2^{-53}$. Hence, we know that $fl(1+2^{53}) = 2^{53}$. So, here, n=54 is the minimum n for which Gaussian Elimination will return the solution of $W_n(0)y = e_1$, when we asked for $W_n(1)y = e_1$.

3.4. **d.** As described in another answer, GECC is seen to yield negligible errors. Based on the above, the MATLAB command "\" does partial pivoting. I tried that command on $W_{54}(0)y = e_1$ and $W_{54}(1)y = e_1$ to arrive at that conclusion.

4

Upon running the following Matlab functions, we indeed find that GECP gives higher accuracy solution than GEPP.

For example, for $W_{100}(1)$, GEPP yields norm(P*W - L*U) = 7, whereaas GECP yields 8.1389e-15.

```
% Gaussian elimination with partial pivoting.
% Needs square A
% Gives:
% L, U and P

function [L, U, P] = gepp(A)
[m, m] = size(A);
U = A;
L = eye(m);
P = eye(m);
for k=1:m-1
    maxI = k-1 + find(abs(U(k:m, k)) == max(abs(U(k:m, k))));
    i = maxI(1,1);

    tmp = U(k, k:m);
    U(k, k:m) = U(i, k:m);
    U(i, k:m) = tmp;

    tmp = L(k, 1:k-1);
    L(k, 1:k-1) = L(i, 1:k-1);
    L(i, 1:k-1) = tmp;
```

```
    tmp = P(k, :);
    P(k, :) = P(i, :);
    P(i, :) = tmp;

    for j=k+1:m
        L(j,k) = U(j,k)/U(k,k);
        U(j,k:m) = U(j,k:m) - L(j,k)*U(k,k:m);
    end

end
end

% Gaussian elimination with complete pivoting.
% Needs square A
% Gives:
% L, U and P

function [L, U, P, Q] = gecp(A)
[m, m] = size(A);
U = A;
L = eye(m);
P = eye(m);
Q = eye(m);
for k=1:m-1
    [maxRowIndeces, maxColIndeces] = find(abs(U(k:m, k:m)) == max(max(abs(U(k:m, k:m)))));
    maxRowIndex = k-1 + maxRowIndeces(1,1);
    maxColIndex = k-1 + maxColIndeces(1,1);

%  Exchange rows
    tmp = U(k, k:m);
    U(k, k:m) = U(maxRowIndex, k:m);
    U(maxRowIndex, k:m) = tmp;

    tmp = L(k, 1:k-1);
    L(k, 1:k-1) = L(maxRowIndex, 1:k-1);
    L(maxRowIndex, 1:k-1) = tmp;

    tmp = P(k, :);
    P(k, :) = P(maxRowIndex, :);
    P(maxRowIndex, :) = tmp;

%  Exchange cols
    tmp = U(:, k);
    U(:, k) = U(:, maxColIndex);
    U(:, maxColIndex) = tmp;


    tmp = Q(:, k);
```

```
    Q(:, k) = Q(:, maxColIndex);
    Q(:, maxColIndex) = tmp;

    for j=k+1:m
        L(j,k) = U(j,k)/U(k,k);
        U(j,k:m) = U(j,k:m) - L(j,k)*U(k,k:m);
    end

end
end
```