LAB-5

Q1) Dolittle:

L → eye (N, N) , A → original matrix
U → zeros (N, N)

Algorithm:

for k = 1 to N
~~U(k,k) = A(k,k) - L(k, 1:k-1). U(1:k-1, k);~~

for j = k+1 to N
U(k, j) = A(k, j) - L(k, 1:k-1) U(1:k-1, j);
~~L(j, k) = [A(k, j) - L(j, 1:k-1) U(1:k-1, k)] / U(k, k)~~;

end

end

For A:
$$\begin{bmatrix} 60 & 30 & 20 \\ 30 & 20 & 15 \\ 20 & 15 & 12 \end{bmatrix}$$

$$L = \begin{bmatrix} 1 & 0 & 0 \\ 0.5 & 1 & 0 \\ 0.33 & 1 & 1 \end{bmatrix}, \quad U = \begin{bmatrix} 60 & 30 & 20 \\ 0 & 5 & 5 \\ 0 & 0 & 0.33 \end{bmatrix}$$

- Crout's Method

P.T.O

2

L, U, A → same as before:

for K = 1 to N

~~L(K,K) = A(K,K) - U(K, 1:K-1) . L(1:K-1, K);~~

for j = K+1 to N

L(K, j) = A(K, j) + U(K, 1:K-1) . L(1:K-1, j)

~~U(j,K) = [A(K,j) - U(j,1:K-1) . L(1:K-1, K)]~~ / L(K,K);

end

end.

$L = L^T$

$U = U^T$

A → same as before

$$L = \begin{bmatrix} 60 & 0 & 0 \\ 30 & 5 & 0 \\ 20 & 5 & 0.33 \end{bmatrix}, \quad U = \begin{bmatrix} 1 & 0.5 & 0.3 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

- Cholecky Method

- Check if $A = A^T$ (L → zeros (N, N))
- for K = 1 to N

~~L(K,K) = sqrt (A(K,K) - L(K, 1:K) . L(K : 1 : K-1)~~].

for j = K+1 : N

~~L(j,K) = [A(j,K) - L(j, 1:K-1) . L(K, 1:K-1)]~~ / L(K,K);

end

end

A → same as before

$$L = \begin{bmatrix} 7.74 & 0 & 0 \\ 3.87 & 2.23 & 0 \\ 2.58 & 2.23 & 0.57 \end{bmatrix}$$

$U = L^T$

Q2) • <u>Gauss Elimination</u>

<u>Algorithm:</u>

A → Matrix (original) $\quad [a][x] = [b]$

b → ~~A~~ Matrix on RHS

```
e = ones(N,1)
for K = 1 : N-1
    for j = K+1 : N
        temp(j,1) = A(j,K) / A(K,K)
        A(j, K:N) = A(j, K:N) - e(j,1) . A(K, K:N)
        b(j,1) = b(j,1) - e(j,1) - b(K,1);
    end
end
final → backSubstitute(A,b)  (Defined later on)
```

Module 1

$$A = \begin{bmatrix} -1 & 1 & -4 \\ 2 & 2 & 0 \\ 3 & 3 & 2 \end{bmatrix}, \quad final = \begin{bmatrix} 1.416 \\ -0.91 \\ -0.58 \end{bmatrix} \rightarrow \text{Values of } x$$

~~Note Module 1 (Piece of code) is later on used in Other Algorithms~~

Page No.:

Date:

Partial Pivoting (Algorithm)

A, b → same as before

~~temp~~ e = ones(N,1)

```
for k = 1 to N-1
    temp = k;
    for i = k+1 to N
        if (A(temp,k) < A(i,k))
            temp = i
        end
    end
    check = A(k,1:N)
    A(k,1:N) = A(temp,1:N)
    A(temp,1:N) = check
    temp2 = b(temp,1);
    b(temp,1) = b(k,1);
    b(k,1) = temp2
```

→ //implement module given in previous page

- For case A & B, the final ans obtained is the same as the one obtained on the prev page

$$ans = \begin{bmatrix} 1.41 \\ -0.91 \\ -0.58 \end{bmatrix}$$

* backsubstitute func. defined : (Func. Definition)

```
ans = zeros(1,N)
for i = N:-1:1
    c = 0
    for k = 1:N
        c = c + A(i,k)
    end
    ans(i) = (b(i,1) - c) / A(i,i);
end
```

• Complete Pivoting (Algorithm) [i/p Arg : A, N, b, x]

```
e = ones(N,1)
for k = 1:N-1
    id_x = k
    id_y = k;
    for i = k+1 to N
        if A(id_x, k) < A(i,k)
            id_x = i
        end
        if A(k, id_y) < A(k,i)
            id_y = i
        end
    end
if A(id_x, k) > A(k, id_y)
    swap (A(id_x, 1:N), A(k, 1:N))
    swap (b(id_x, 1), b(k,1))
else
    swap (A(1:N, id_y), A(1:N, k))
    swap (x(id_y, 1), x(k,1))
end
```

P.T.O

~~// implement Module 1 Now~~
end → Code ends here

- Ans final = $\begin{bmatrix} 1.41 \\ -0.91 \\ -0.58 \end{bmatrix}$ → With same A, b.

- For Naive Gauss Method, divisions → 6
  flops → 15

Partial & complete pivoting do more comparisons, hence it might take some more time than Basic Gauss Elimination, but they might give us answers in certain corner cases

Q3) Basic Gauss elimination should be used for banded matrices. Use Algorithm present in Q2.

$$A = \begin{bmatrix} 5 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 5 \end{bmatrix}, \quad b = \begin{bmatrix} 4 \\ 3 \\ 4 \end{bmatrix}$$

ans final = $\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$

Computational Cost : • $2 \times (N-1)$ flops in Algorithm (BGE)
• N more steps in backsubstitution