Tushar Mahat
A00429666

Quiz #6

1) To the given program we have to add the following before the "stop" line.

```
! Assign zero to b
  b = 0.0

! Assign new values to b
  b(1,1) = 5.0
  b(2,1) = 6.0
  b(3,1) = 7.0

! Solve Ax=b and return solution in b
call sgetrs(trans, n, nrhs, A, lda, ipiv, b, ldb, info)
write (*, *) 'SGETRS: info =', info
if (info .NE. 0) stop

! Print the results
write (*,*) 'Solution =', b
write (*,*) ' Estimate of condition number =', 1.0/rcond
```

2 a) We should be looking at the following files:

(i) 'OpCountAnalysisForForwardElimination'
(ii) 'OpCountAnalysisForLy=dSolve'
(iii) 'OpCountAnalysisForUx=ySolve'

2 b) $\underline{Y} = A\underline{X}$

$$\begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ \vdots & & & \\ a_{n1} & \cdots & & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \\ \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n \end{bmatrix}$$

Pseudo code

```
for i=1 to n do   // for each row of A, process ith row
    y(i) = 0
    for j=1 to n do,
        y(i) = y(i) + A(i,j) * x(j)
    end
end
```

Operation count for multiplications:

In the inner loop, there is one multiplication operation.

$$\sum_{j=1}^{n} 1$$

Inner loop executes for $i = 1$ to n, so

$$\sum_{i=1}^{n} \left(\sum_{j=1}^{n} 1 \right) = \sum_{i=1}^{n} (n) = n \sum_{i=1}^{n} 1 = n.n = n^2$$

$\therefore$ The number of multiplications performed by this algorithm is n^2

2 c) Consider $Ax_1 = e_1$ for solving x_i

① The cost of $PA = LU$ factorization, multiplications only is

$$\boxed{\frac{n^3}{3} - \frac{n^2}{2} + \frac{n}{6}}$$

② The cost of solving $\underline{L}\underline{y} = \underline{d}$ for multiplications only is:

$$\frac{n^2}{2} - \frac{n}{2}$$

The cost of solving $U\underline{x}_1 = \underline{y}$ for multiplications only is

$$\frac{n^2}{2} - \frac{n}{2}$$

Total cost for $\underline{L}\underline{y} = \underline{d}$ and $U\underline{x}_1 = \underline{y} = 2\left(\frac{n^2}{2} - \frac{n}{2}\right)$

Now, there are total n systems to solve for $Ax_i = e_i$

i.e $Ax_1 = e_1$; $Ax_2 = e_2 \dots x_n = e_n$.

So, for n systems total cost to solve $Ly = d$ and $Ux = y$ is

$$\boxed{2n\left(\frac{n^2}{2} - \frac{n}{2}\right)}$$

③ we have solved for $X = A^{-1}$, entries of X are $X_1, X_2 \dots X_n$.

Now, compute $\underline{x} = A^{-1}\underline{b}$ the cost would be $\boxed{n^2}$ from (2b)

The total cost of this algorithm is

$$\frac{n^3}{3} - \frac{n^2}{2} + \frac{n}{6} + 2n\left(\frac{n^2}{2} - \frac{n}{2}\right) + n^2 = \frac{4n^3}{3} - \frac{n^2}{2} - \frac{n}{3} \quad -①$$

The cost of solving $Ax = b$ as discussed in class is $\left(\frac{n^3}{3} + \frac{n^2}{2} - \frac{5}{6}n\right)$ — (11)

Comparing two costs.

$$\frac{n^3}{3} + \frac{n^2}{2} - \frac{5}{6}n \leq \frac{4n^3}{3} - \frac{n^2}{2} - \frac{n}{3}$$

This approach is 4 times more expensive than the one we discussed in class.

d) Operation count analysis gives the cost of the algorithm by counting the number of arithmetic operations. It is useful to find the cost of an algorithm.

Operation count analysis gives us the cost of operation of different algorithms without depending on the specifications of computer systems.

3 a) The names of the files that we should be looking are :

'ForwardAndBackwardForLinearSystem'
'LA_forwardBackwardError'
'SummaryOFFE+BE+Conditioning'

b) $A = \begin{bmatrix} 9.7 & 6.6 \\ 4.1 & 2.8 \end{bmatrix}$, $\underline{b} = \begin{bmatrix} 9.7 \\ 4.1 \end{bmatrix}$

The exact solution is $\underline{x} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$

Approximation solution $\hat{\underline{x}} = \begin{bmatrix} 0.34 \\ 0.97 \end{bmatrix}$

Forward Error $\|\underline{e}\|_1 = \|x - \hat{x}\|_1$

$= 1.63$

$x - \hat{x}$

$= \begin{bmatrix} 1 \\ 0 \end{bmatrix} - \begin{bmatrix} 0.34 \\ 0.97 \end{bmatrix}$

$= \begin{bmatrix} 0.66 \\ -0.97 \end{bmatrix}$

Relative forward error $= \dfrac{\|\underline{e}\|_1}{\|x\|_1} = \dfrac{1.63}{1} = 1.63$

3c) $\underline{r} = A\hat{\underline{x}} - \underline{b}$

$$= \begin{bmatrix} 9.7 & 6.6 \\ 4.1 & 2.8 \end{bmatrix} \begin{bmatrix} 0.34 \\ 0.97 \end{bmatrix} - \begin{bmatrix} 9.7 \\ 4.1 \end{bmatrix}$$

$$= \begin{bmatrix} 9.7 \\ 4.11 \end{bmatrix} - \begin{bmatrix} 9.7 \\ 4.1 \end{bmatrix}$$

$$= \begin{bmatrix} 0 \\ 0.01 \end{bmatrix}$$

Perturbing original system by r.

$$\underline{b} + \underline{r} = \begin{bmatrix} 9.7 + 0 \\ 4.1 + 0.01 \end{bmatrix} = \begin{bmatrix} 9.7 \\ 4.11 \end{bmatrix}$$

$\therefore$ Perturbed problem : $\begin{bmatrix} 9.7 & 6.6 \\ 4.1 & 2.8 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 9.7 \\ 4.11 \end{bmatrix}$

$\hat{\underline{x}}$ is exact solution for

Backward error $\|\underline{r}\|_1 = 0.01$

Relative backward error $= \dfrac{\|\underline{r}\|}{\|\underline{b}\|}$

$$= \frac{0.01}{13.8}$$

$$= 0.000724637681$$

e) The expression for Cond (A) is $||A||.||A^{-1}||$

$$\text{Cond}(A) = ||A|| \, ||A^{-1}||$$

$$= (13.80)(163.0)$$

$$= 2249.4$$

$$= 2.2494 \times 10^3$$

The condition number is about 10^3, so we loose 3 digits of accuracy.

The relative backward error is very smaller than relative forward error. It means that if we tweak the original problem by relatively small amount, the solution changes by very large amount. Which eventually means the problem is ill-conditioned.

The theoretical relationship between Cond (A), relative backward error and relative forward error is

Relative forward error $\leq$ Cond (A) x Relative Backward Error

$$\frac{||e||}{||x||} \leq (||A||.||A^{-1}||) . \frac{||r||}{||b||}$$