भारतीय प्रौद्योगिकी संस्थान गुवाहाटी
INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI

MA 423: Matrix Computations Lab

Lab 02

AB Satyaprakash (180123062)

17 Aug 2021

Question 1.

Part (a):

```
The factors L and U are:
L =
     1.000000000000000e+00                        0
     9.999999999999998e+19    1.000000000000000e+00

U =
     1.000000000000000e-20    1.000000000000000e+00
                         0    -9.999999999999998e+19

A - L*U =
                         0                        0
     1.110223024625157e-16    1.000000000000000e+00
```

Part (b):

```
Solution using genp
     0
     1

Exact solution of Ax = b
     -1
     1

2-norm of difference between exact solution and genp solution =
1.000000
```

From the above algorithm we understand the following about GENP -

- With small pivot (diagonal) values in the LU decomposition step, we get incorrect results.
- This is due to the precision error, caused by limited floating point precision in commonly used computers.

Question 2.

- **gepp**(A) - a function program to find a unit lower triangular matrix L, an upper triangular matrix U and a column vector p satisfying A(p,:) = LU via Gaussian Elimination with Partial Pivoting (GEPP).

A script file *q2.m* generates matrices of sizes n = 5 to n = 10 and then calls gepp(A) to generate L, U and p. The same are obtained using MATLAB's inbuilt lu() function and the norm of difference of individual parameters are compared.

```
For value of n = 5
[norm(A(p,:) - L*U), norm(L-L1), norm(U-U1), norm(p-p1)]
         1.582501385132391e-16       1.154730383859585e-16       2.288783399261119e-16                           0

For value of n = 6
[norm(A(p,:) - L*U), norm(L-L1), norm(U-U1), norm(p-p1)]
         3.518986883916217e-16       2.386272608446130e-16       2.559841637791462e-16                           0

For value of n = 7
[norm(A(p,:) - L*U), norm(L-L1), norm(U-U1), norm(p-p1)]
         3.899028304946149e-16       2.105485303479983e-16       6.492204558399693e-16                           0

For value of n = 8
[norm(A(p,:) - L*U), norm(L-L1), norm(U-U1), norm(p-p1)]
         9.322019744580797e-16       4.395945110635763e-16       6.152992238442192e-16                           0

For value of n = 9
[norm(A(p,:) - L*U), norm(L-L1), norm(U-U1), norm(p-p1)]
         7.378597068278373e-16       4.382142268147512e-16       1.501560581738502e-15                           0

For value of n = 10
[norm(A(p,:) - L*U), norm(L-L1), norm(U-U1), norm(p-p1)]
         1.048031358120115e-15       4.870465210620754e-16       2.783465172420512e-15                           0
```

Question 3.

- **geppsolve**(A) - a function program to solve a system Ax = b via GEPP calling the program [L,U,p] = gepp(A) and the programs written in Lab 1 for solving upper and lower triangular systems.

A script file *q3.m* generates matrices of sizes n = 5 to n = 10 and then calls geppsolve(A). The solution is compared with x = A/b and displayed. For instance, given n = 5, a sample result is as below.

```
For value of n = 5
Solution using geppsolve(A,b)   Solution using A\b
   4.4030e+00                      4.4030e+00
   3.3596e-01                      3.3596e-01
  -3.7296e+00                     -3.7296e+00
  -2.6119e-01                     -2.6119e-01
   9.2292e-01                      9.2292e-01
```

Question 4.

- **mydet(A)** - a function program that uses an efficient version of LU factorization to compute the determinant of A in O(n^3) flops.

A script file *q4.m* accepts user input n = size of matrices and displays the output for mydet(A) after generating a random matrix A of size n*n. Also we compare the value of mydet(A) with the inbuilt MATLAB function det(A) which generates the same thing. The results have been compared. Here are the results for n = 100.

```
Enter the size of matrix A: 100
The determinant of A using mydet() function
     1.427856594891380e+76

The determinant of A using MATLAB inbuilt function
     1.427856594891261e+76
```

Question 5.

- **mychol(A)** - a function program that executes the inner product form of the Cholesky Decomposition for finding the Cholesky factor of an n × n positive definite matrix A in n^3/3 + O(n^2)

A script file *q5.m* generates random symmetric and positive definite matrices of sizes n = 5 to n = 10 and then calls mychol(A) to generate G - the Cholesky factor. It also generates G1 by calling the MATLAB inbuilt function chol(A). The norm of the difference between them is tabulated below:

```
     Var1      norm_difference
     ____    _____________________

     5     9.65303448446918e-17
     6     4.49851956655561e-16
     7     4.47653232820368e-16
     8     5.08199325736186e-16
     9     4.76163469113899e-16
     10    4.49713425461879e-16
```

N.B. A slightly different way to generate symmetric positive definite matrices has been used which has been well commented in the code. Here is a snippet of the same.

```
% construct a symmetric matrix using either
     A = 0.5*(X+X');
% since A(i,j) < 1 by construction and a symmetric diagonally dominant matrix
% is symmetric positive definite, which can be ensured by adding nI
     A = A + n*eye(n);
```