Lab 3

Steps you should follow :

1. Start a Matlab session by typing

   ```
   $ matlab &
   ```

   at the command prompt.

2. In the Matlab command window, type

   ```
   >> diary lab3_ID
   ```

   where ID stands for your roll number. For example, if your roll number is 12345, then the command will be `diary lab3_12345`; if your roll number is 123456, then the command will be `diary lab3_123456`. This will create a file named `lab3_ID` in the present working directory. PLEASE DO NOT EDIT THIS FILE.

3. Do your lab assignment – create a separate file to write your scripts/functions, if required; once done, at the Matlab's command prompt, type

   ```
   >> diary off
   ```

4. Attach this file (that is, `lab3_ID`) and any other matlab code file that you may have created for the labwork in an **email with subject Lab3-ID** and send it to

   mth308.iitk@gmail.com

   before the end of the lab session, that is, by 3:50 pm. Note that **late submissions will not get any credit**.

Write a program for estimating the condition number of a matrix A. You may use either the 1-norm or the ∞-norm (or try both and compare the results). You will need to compute $\|A\|$, which is easy, and estimate $\|A^{-1}\|$, which is more challenging. From the properties of the norms, we know that if z is the solution to $Az = y$, then

$$\|z\| = \|A^{-1}y\| \leq \|A^{-1}\|\|y\|,$$

so that

$$\frac{\|z\|}{\|y\|} \leq \|A^{-1}\|,$$

and this bound is achieved for some optimally chosen y. Thus, if we choose a y such that the ratio $\|z\|/\|y\|$ is as large as possible, then we will have a reasonable estimate for $\|A^{-1}\|$. Try two different approaches to choosing y:

1. Choose y as the solution to the system $A^T y = c$ where c is a vector each of whose components is ± 1, with the sign of each component chosen by the following heuristic. Using the factorization $A = LU$, the system $A^T y = c$ is solved in two stages, successively solving $U^T v = c$ and $L^T y = v$. At each step of the first triangular solution, choose the corresponding component of c to be 1 or -1 depending on which will make the resulting component of v larger in magnitude. Then solve the second triangular system in the usual way for y.

2. Choose a small number, say five, different vectors y randomly and use the one producing the largest ratio $\|z\|/\|y\|$.

You may use Matlab's `lu` function to obtain the necessary LU factorization of A; type

```
>> help lu
```

at the Matlab's command prompt to learn more about how to use the `lu` finction. Use both of the approaches on each of the following matrices:

$$A_1 = \begin{bmatrix} 10 & -7 & 0 \\ -3 & 2 & 6 \\ 5 & -1 & 5 \end{bmatrix}, \qquad A_2 = \begin{bmatrix} -73 & 78 & 24 \\ 92 & 66 & 25 \\ -80 & 37 & 10 \end{bmatrix}.$$

How do the results using these two methods compare? To check the quality of your estimates, compute A^{-1} explicitly to to determine its true norm (this computation can also make use of the LU factorization already computed). How does your condition numbers (the that uses estimated $\|A^{-1}\|$ as well as the one with exact $\|A^{-1}\|$) compares with the condition number given by Matlab.

Some extra practice homework problems (not to be submitted as part of the lab assignment and will not be graded).

I. The solution to $Ax = b$ may be written $b = A^{-1}x$. This can be a good way to analyze algorithms involving linear systems. But we try to avoid forming A^{-1} explicitly in computations because it is more than twice as expensive as solving the linear equations. A good way to form $B = A^{-1}$ is to solve the matrix equation $AB = I$. Gauss elimination applied to A gives $A = LU$, where the entries of L are the pivots used in elimination.

(a) Show that about $n^3/3$ work reduces $AB = I$ to $UB = L^{-1}$, where the entries of U and L^{-1} are known.

(b) Show that computing the entries of B from $UB = L^{-1}$ takes about $n^3/2$ work. Hint: It takes one floating point operation (flop) per element for each of the n elements of the bottom row of B, then two flops per element of the $n-1$ row of B, and so on to the top. The total is $n(1+2+\cdots+n)$.

(c) Use this to verify the claim that computing A^{-1} is more than twice as expensive as solving $Ax = b$.

II. Consider the linear system

$$\begin{bmatrix} 1 & 1+\epsilon \\ 1-\epsilon & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1+(1+\epsilon)\epsilon \\ 1 \end{bmatrix}$$

where ϵ is a small parameter to be specified. The exact solution is obviously

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 \\ \epsilon \end{bmatrix}$$

for any value of ϵ.

Use Gaussian elimination to solve the linear system. For $\epsilon = 1.0 \times 10^{-n}, n = 0, 2, 4, 6, 8$, compute an estimate of the condition number of the matrix (using one or both methods given in Lab Problem) and the relative error in each component of the solution. How accurately is each component computed? How does the accuracy attained for each component compare with expectations based on the condition number of the matrix. What conclusions can you draw from this experiment?

III. The determinant of a triangular matrix is equal to the product of its diagonal entries. Use this fact to develop a program for computing the determinant of an arbitrary $n \times n$ matrix by using its LU factorization. How can you determine the proper sign for the determinant? To avoid the risk of underflow and overflow, you may wish to consider computing the logarithm of the determinant instead of the actual value of the determinant.