Computational Linear Algebra – Programming Assignment 1

(points: 65, due on January 5)

1 Important points

- Submit crisp codes in a zip folder with folder name '*FIRSTNAME_LASTNAME_pga1*.zip'.
- Add adequate comments in the code to explain the algorithms.
- The code will be checked for correctness and plagiarism.
- Inbuilt commands should not be used unless specified.
- The programs have to be coded in MATLAB.
- Error between two vectors x and $y \in \mathbb{R}^n$ is the euclidean norm of their difference, i.e. square root of $\sum_{i=1}^{n} (x_i - y_i)^2$.
- Error between two matrices A and $B \in \mathbb{R}^{n \times n}$ is the frobenius norm of their difference.

$$\text{Error}(A, B) = \sqrt{\sum_{i,j=1}^{n} (A_{ij} - B_{ij})^2}.$$

2 Questions

2.1 QR decomposition (5+5+3+3)

a) Implement QR decomposition of $A \in \mathbb{R}^{n \times n}$ using a) Gram-Schmidt (GS) procedure (name it *gs.m*). and b) Householder reflections (name it *hr.m*).

b) Note that we can perform a transformation of a $2-$dimensional vector $[a, b]$ to $[c, 0]$ by performing a transformation using the below matrix

$$R = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix}$$

Here θ can be computed using $[a, b]$ and $[c, 0]$. Use this idea to derive and implement an algorithm (name it *prop.m*) to perform QR decomposition. Hint. Recall the strategy used in deriving method for QR decomposition using Householder reflections.

c) For the matrix $A = 0.00001 * eye(n) + hilb(n)$, where $eye(n)$ returns an identity matrix and $hilb(n)$ return a hilbert matrix of size n.
1) Apply QR decomposition for the above matrix using Gram-Schmidt, Householder reflections and method in b).
2) Note the error between $Q^\top Q$ and identity matrix.
4) Print the name of algorithm gives a better decomposition (lesser error between A and QR)?

d) For $n = 3$, follow the below procedure and use all three methods to find the estimate of x. Print the name of algorithm with lesser error between x and estimate of x)?

1 Generate two orthonormal vectors $v1$ and $v2$.
2 Construct matrix $A = 50000 * v1 * v1^\top + 2 * v2 * v2^\top$.
3 Generate $x = randn(n, 1)$ and $b = Ax$.
4 Take this b as input and find the estimate of x.

2.2 LU decomposition (5+5+4)

In this problem, you are permitted to perform computations using real numbers only.

a) Write a program (name it *cmplx.m*) to solve complex system of equations using LU without pivoting. In particular, the program should take matrix $A \in \mathcal{C}^{n\times n}$, $b \in \mathcal{C}^n$ and output $x \in \mathcal{C}^n$ such that $Ax = b$. Also the program should print the statement *LU decomposition does not exist* if LU decomposition does not exist for the input matrix. Further print the flop counts for LU decomposition in the above case.

b) For an invertible square matrix $A \in \mathcal{C}^{n\times n}$, write a program (name it *LUpartial.m*) to implement LU with partial pivoting and return the determinant and inverse of the matrix. Note that the program should not contain any inversion subroutine applied for L or U matrix. Refer the textbook 'Numerical Linear Algebra' for understanding partial pivoting.

c) Repeat b) by first finding QR decomposition of matrix $A \in \mathcal{C}^{n\times n}$. Then given QR decomposition of a square matrix, write a program (name it *qrtolu.m*) to output LU decomposition of the corresponding matrix without explicitly constructing the matrix.

2.3 Jacobi and Gauss Seidel methods (5+6+4+6)

2.3.1 Consider the following system of equations,

$$\begin{aligned} 3x_1 + x_2 + x_3 &= 5 \\ x_1 + 7x_2 + 3x_3 &= 11 \\ 2x_1 + 4x_3 &= 5 \end{aligned}$$

and

$$\begin{aligned} x_1 + 5x_2 + x_3 &= 7 \\ 9x_1 + 3x_2 + 3x_3 &= 15 \\ 2x_1 + x_2 + 4x_3 &= 7 \end{aligned}$$

a) Compute necessary matrices, make norm estimates and find whether Jacobi and Gauss-Seidel methods converge for these problems. If they do, estimate analytically the number of iterations needed to find the solution with relative errors less than 0.0001 in the 2-norm. You can print this explanation in your code.

b) Create a mat file named $jacobi\_gauss\_1.m$. Write and execute programs for the Jacobi iteration and the Gauss Seidel iteration. Use these programs to find the solutions of both problems above (using both techniques) with required precision.

c) Find the required number of iterations required in all cases. Compare it with the analytical estimates. Compare the number of iterations required in different techniques. Comment on the results. You can print this explanation in your code. Use graphics to illustrate the actual rate of convergence for both methods (for instance plot $\|x^{(n)} - x^{(n-1)}\|_2/\|x^{(n)}\|_2$ for $n = 1, 2, ...$).

2.3.2 Consider the system of equations $Ax = b$, where A is the 10×10 symmetric Toeplitz with first column $[2, -1, 0, 0, ...0]^\top$ and $b = [11, 0, 0....0]^\top$. Create a mat file named $jacobi\_gauss\_2.m$. Solve

for x using the Jacobi method and the Gauss Seidel method until the iterates yield a relative error of 0.0001 in the 2-norm. Why can you be sure that both methods will converge in this case? What do you guess is the true solution x? Repeat this problem with 20×20 symmetric Toeplitz with first column $[2, -1, 0, 0, ...0]^\top$ and $b = [21, 0, 0....0]^\top$. The answers to the theoretical part should be printed in your code.

Note: Your programs should be written in general so that they can handle any other examples, not only the particular ones given above.

2.4 Direct v/s Iterative methods (8+6)

Consider the following system of equations,

```
e=ones(n,1)
A=spdiags([-e 2*e -e], -1:1, n, n);
A=full(A);
b=rand(n,1);
```

Note: A is a matrix with 3 bands, 2 on diagonal and -1 on upper and lower bands. Commands 'spdiag' and 'full' helps to create such a matrix.

a) Create a mat file named $jacobi.m$. Record the number of iterations needed to achieve tolerance of 0.1, 0.01, 0.001, 0.0001, 0.00001 for three different values of n (say n = 10, 50, 100) using Jacobi method. Plot your results and comment (should be printed in your code) on how increasing the system size and reducing tolerance affects the algorithm.

b) Create a mat file named $comparison.m$. Compare the time required to solve the system in part a) for a tolerance of 0.001 with different values of n (say n = 10, 50, 100) using Jacobi method and your best performing direct method. What can you conclude? (should be printed in your code).