Computational Linear Algebra – Programming Assignment 2

(points: 20, due on November 17)

1 Important points

- All the commented codes have to be crisp and submitted online with a report. Submit the codes and report in a zip folder with folder name '*STUDENTNAME_pga2*.zip'.
- If the question is just to implement a program and the matrix is not specified, show an example in the report. The code will be checked for correctness and plagiarism.
- Inbuilt commands should not be used unless specified.
- All the programs should take the matrix and size of the matrix as input (n if matrix $A \in \mathbb{R}^{n \times n}$).
- The programs have to be coded in MATLAB or python.
- Error between two vectors x and $y \in \mathbb{R}^n$ is the euclidean norm of their difference.

2 Questions

2.1 Jacobi iterations (5+4+3)

a) Write a program (*jacobi.m* or *jacobi.py*) to implement Jacobi iteration to solve the linear system of equations $Ax = b$, given the input arguments matrix A, vector b and a number tol specifying the desired accuracy. Use a zero starting guess and iterate until the relative residue $||r^{(k)}||/||r^{(0)}||$ is less than tol, where $r^{(k)} = Ax_k - b$ and tol is an user input parameter for stoping the iterations.

b) Compare the number of iterations to converge to a given accuracy for a linear system with

```
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.

Record the number of iterations needed to achieve tolerance of 0.1, 0.01, 0.001, 0.0001, 0.00001 for a few different values of n (say $n = 10, 50, 100$). Plot your results and comment on how increasing the system size and reducing tolerance affects the algorithm.

c) Compare the time required to solve the system in part b) with different values of n using Jacobi method and your best performing direct method. What can you conclude?

2.2 Application of QR decomposition (4+2+2)

1) Write a program (*QRapp1.m* or *QRapp1.py*) completing the below algorithm to find eigenvalues and eigenvectors of any symmetric matrix.
2) Note the error in eigenvalues using GS and MGS algorithms for $A = B^\top B$, where $B = randn(4,4)$, as compared to the eigenvalues obtained using the inbuilt matlab command. Store the eigenvalues in a vector to compute the error.
3) Can the implemented algorithm be used to find eigenvalues and eigenvectors of an arbitrary non-symmetric matrix? If yes, show an example. If no, explain why.

Input: $A \in \mathbb{R}^{n \times n}$;
Parameter: Maximum iterations N;
Output: Eigenvalues and eigenvectors of A;
$A_1 = A$
for $i = 1, \ldots, N$ **do**
    [Q_i,R_i]= QR(A_i);
    $A_{i+1} = R_i Q_i$;
end
/* Complete the algorithm */