CSPP58001
Final homework
Due: finals week

1) Show that the eigenvalues of a positive definite matrix are all
positive.

2) Show that the condition number of unitary matrix is 1.

3) Following the technique we outlined in class lecture,

a. derive the system of linear equations for the 4*(n-1) coefficients
of the cubic spline interpolator.

b. write a matlab code that takes n data values as input and returns
the cubic spline interpolants. Create some sample data and execute this
code, plotting the results.

4) Consider the system of equations x'=Ax, where x is a column vector
of n unknowns, A is an nxn matrix, and x' denotes dx/dt.

a. discretize the system using forward euler and derive a simple condition
for the stability of the solution based on the size of the discrete timestep dt.

b. do the same for backward Euler.

5) Let A a real nxn matrix and v is an eigenvector of A. Then (I - vv')Av = 0.
Is this correct ? Prove.

6)

a. Provide a sufficient condition so that both Jacobi and Gauss-Seidel methods converge when applied for the solution of a system whose matrix is of the form

```
A = [-10     2;
     alpha   5;]
```

b. Construct an example to show that the convergence of the Jacobi method does not necessarily imply that the Gauss-Seidel method will converge for any initial guess.

7) Implement a code for the Jacobi and SSOR relaxation methods to solve a system of equations Ax=b where A(nxn) is a SPD operator. Note that the sparse matrix 'A' is obtained from the attached matlab script 'FD_2D_operator.m' which discretizes the Poisson equation in 2-dimensions using Finite Difference. You can directly call the function in your code (if MATLAB) to get the required data for different values of 'N'.

```
[n, A, b] = FD_2D_operators(N)
```

For users of any other language, the code also outputs 'Operator.m' along with the size of the system. The format of the file is as follows: For each non-zero entry in the matrix, the row of values are given by

```
RowIndex ColumnIndex            Value
```

Compute the spectral radius of the iteration matrix and verify (and tabulate) the convergence rate of the solvers for N=[10 20 40 80 160]. Use a zero starting vector for 'x' and all ones for 'b'.

NOTE: Based on the spectral radius, compute optimal relaxation factor for SSOR.
--
8) Use the Preconditioned-Conjugate Gradient method to solve

$$P^{-1} A x = P^{-1} b,$$

where both A and P are SPD operators. In MATLAB, the command is 'pcg' for preconditioned CG.
For those writing their code in C++, I will post a PCG solver in C.

Provide to the solver only the action of the matrix A on a vector (A*v) and the action of the
preconditioner P on a vector (P^(-1)*v). Utilize the sparsity of the operators and optimize
A*v and P^(-1)*v operations as much as possible. Use the same matrix A as in problem 2
and the Jacobi and SSOR iteration (omega = 1.4) matrix as the preconditioner P. For different
values of 'n', tabulate the number of iterations, CPU time to converge to a relative tolerance
of 1e-10 without any preconditioner and using the two choices for P. Comment on
the most efficient CG method among the three choices overall.
Use a vector of all ones for 'b' and zeros for starting guess of 'x'.