Math 441 (2020) – Homework 5: 20 points

Due: Thursday, October 29, 2020, 11:59pm (HARD Deadline).

1. [10pts] (Power Method) Recall the Power Method algorithm is defined by:

$$\begin{aligned} \mathbf{y}_n &= A\mathbf{x}_n \\ \lambda_n &= \phi(\mathbf{y}_n)/\phi(\mathbf{x}_n) \\ \mathbf{x}_{n+1} &= \mathbf{y}_n / \parallel \mathbf{y}_n \parallel_2 \end{aligned}$$

where here and in <u>both</u> parts a) and b) below

$$\phi(\mathbf{z}) = z_1 + z_2 + \cdots z_n \qquad , \qquad \mathbf{z} = (z_1, z_2, \ldots z_n)$$

For most (but not all) initial guesses $\mathbf{x}_0$, λ_n will approach the dominant eigenvalue λ of A with x_n being an approximation of the associated (unit) eigenvector.

a) Let

$$A = \begin{bmatrix} 7 & 3 \\ 3 & -1 \end{bmatrix} \qquad , \qquad \mathbf{x}_0 = \begin{pmatrix} 1 \\ -3 \end{pmatrix}$$

Using hand calculations (no decimal approximations) only, compute in sequence: $\mathbf{y}_0, \lambda_0, \mathbf{x}_1, \mathbf{y}_1, \lambda_1, \mathbf{x}_2$. You should find $\mathbf{x}_1 = -\mathbf{x}_2$.

Is the method converging to the dominant eigenvalue? Give some reasonable explanation.

b) Let

$$A = \begin{bmatrix} 16 & 2 & 4 \\ 1 & 40 & -3 \\ 0 & 3 & 5 \end{bmatrix} \qquad , \qquad \mathbf{x}_0 = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

Use the posted *Power.m* and *phi.m* matlab scripts to approximate the dominant eigenvalue using $N = 15$ iterates. Include an extra column in the output matrix R as follows:

```
>> R=[x',lambda(k),residual(k),abs(r-edom)]
```

where *edom* is the dominant eigenvalue. You may find this value using the matlab statement

```
>> evals=eig(A)
```

evals is a vector of all the eigenvalues of A.

2. [10pts] (Steepest Descent) Recall that the steepest descent algorithm for approximating the solution of $Ax = b$ (when $A = A^T$ is positive definite) is:

$$\begin{aligned}\mathbf{r}_n &= A\mathbf{x}_n - \mathbf{b} \\ \alpha_n &= \frac{\mathbf{r}_n^T\mathbf{r}_n}{\mathbf{r}_n^T A\mathbf{r}_n} \\ \mathbf{x}_{n+1} &= \mathbf{x}_n - \alpha_n\mathbf{r}_n\end{aligned}$$

Here $\mathbf{r}_n$ is the residual vector and $\mathbf{x}_n$ is the approximation of the solution of $Ax = b$.

a) Let

$$A = \begin{bmatrix} 2 & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{2} \end{bmatrix} \quad , \quad \mathbf{b} = \begin{pmatrix} 1 \\ 2 \end{pmatrix} \quad , \quad \mathbf{x}_0 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

First prove A is positive definite and then use hand calculations (no decimal approximations) to compute in sequence: $\mathbf{r}_0, \alpha_0, \mathbf{x}_1, \mathbf{r}_1, \alpha_1, \mathbf{x}_2$.

b) Use the code *steepest.m* to approximate the solution of $Ax = b$ where $A \in \mathbb{R}^{50\times 50}$ is a tri-diagonal matrix defined by in code:

```
M=50;
a=2;
am=-1;
ap=-1;
A=diag(a*ones(1,M))+diag(ap*ones(1,M-1),1)+diag(am*ones(1,M-1),-1);
xbar=ones(M,1);    % exact solution
b=A*xbar;
xn=zeros(M,1);    % initial guess
```

The iterate output in *steepest.m* is stored in the matrix R.

```
R(n,:)=[n,xn',norm(en,2)];
```

The last column of R is the 2-norm of the absolute error. Plot this absolute error versus iteration number for $N = 3000$ iterates! Use Matlab to do this. (Doesn't converge as fast as conjugate gradient!!)