HW 6

1: Ghost eigenvalues Run a hundred steps of Lanczos (you can use the code from lecture 32) using the matrix A and starting vector q_1 computed by

```
A  = diag(1:100);
q1 = randn(100,1); q1 = q1/norm(q1);
```

Starting from step ten, plot the convergence of the five largest and the five smallest Ritz values of the tridiagonal T_k as a function of the step k. What do you notice?

2: Symmetrizing Gauss-Seidel Suppose A is symmetric, and write $A = L + L^T - D$, where L is the lower triangular part of A and D is the diagonal. A standard Gauss-Seidel sweep starting at the first solution component and going to the nth component looks like

$$x^{(k+1)} = x^{(k)} + L^{-1}(b - Ax^{(k)}).$$

If we do a forward sweep followed by a backward sweep, we have a *symmetrized* Gauss-Seidel step:

$$\begin{aligned} y^{(k)} &= x^{(k)} + L^{-1}(b - Ax^{(k)}) \\ x^{(k+1)} &= y^{(k)} + L^{-T}(b - Ay^{(k)}). \end{aligned}$$

Show that the symmetrized version computes $x^{(k+1)} = x^{(k)} + M(b - Ax^{(k)})$ where M is symmetric. (*Hint*: Try to reduce to the case $x^{(k)} = 0$.)

3: PCG The following MATLAB function files are linked from the course web page:

```
poisson3d(n)      -- 3D Poisson matrix for an n-by-n-by-n grid
poisson3d_rhs(n) -- Generate an example RHS
pc_gs(A)          -- Generate a Gauss-Seidel preconditioner function
pc_2grid(A,n,p)   -- Generate a two-grid multigrid preconditioner,
                     where the coarse grid is about p-by-p-by-p
```

Use preconditioned conjugate gradients (the MATLAB routine `pcg`) so solve the system $Ax = b$ to a relative residual accuracy $\|r\|/\|b\| < 10^{-6}$, where A and b are generated by

```
n = 65;
A = poisson3d(n);
b = poisson3d_rhs(n);
```

Solve the system with no preconditioner, with a symmetric Gauss-Seidel preconditioner, and with the two-grid multigrid preconditioner (I suggest using a coarse grid size of `p=17`). In each case, plot the residual history and comment on the solution cost. What is the time per iteration for each method? What is the time to solution not including the time to set up the preconditioner? What is the total time to solution including the preconditioner? Note that you will wish to use MATLAB rather than Octave for this — there seems to be something strange about the Octave implementaiton of `pcg`.

Hint: You can use the Gauss-Seidel preconditioner like this:

```
Mfun = pc_gs(A);
x = pcg(A,b,tol,maxit,Mfun);
```

The two-grid preconditioner should be used similarly.