CSci 5304, F'07	Homework # 4	Due Date: 12-06-07

1. Let

$$A = \begin{pmatrix} -9 & 11 & -21 & 63 & -252 \\ 70 & -69 & 141 & -421 & 1684 \\ -575 & 575 & -1149 & 3451 & -13801 \\ 3891 & -3891 & 7782 & -23345 & 93365 \\ 1024 & -1024 & 2048 & -6144 & 24572 \end{pmatrix}.$$

The object of this exercise is to get as good an estimate as possible of the smallest eigenvalue of this matrix.

(a) Use the matlab `eig` function to calculate the eigenvalues and eigenvectors of this matrix. (i) Calculate the residuals of the resulting eigenvalue-vector pairs. (ii) Use the calculated eigenvectors to estimate the condition numbers of the eigenvalues (at least the smallest eigenvalue in absolute value). (iii) Use these condition number estimates and the residuals to estimate the error in the estimates of the eigenvalues (absolute or relative error as appropriate).

(b) Compute the complete SVD of this matrix. Use the singular vectors corresponding to the smallest singular value as estimates of the left and right eigenvectors of A. (i) What is the corresponding eigenvalue and its associated residual as an eigenvalue-vector pair, (ii) what is the estimated condition number of this eigenvalue, based on these eigenvectors, and (iii) what is the resulting estimated accuracy (i.e. error) in the eigenvalue estimate?

(c) Compute $A^5 = A{\cdot}A{\cdot}A{\cdot}A{\cdot}A$ in Matlab. Is this computation exact? What does the result say about the exact eigenvalue(s) of A?

2. Consider the set of 6 web pages A, ..., F in which

A points to B, C, D;
B points to C, D, F;
C points to D, F;
D points to C, E.
E points to F.
F points to B.

(a) What is the matrix H representing the Markov chain that models the Page Rank for this case?

(b) Compute the eigenvector that yields the Page Rank, and list the page rank of the 6 web pages.

(c) Show the adjacency matrix used to compute the Hubs and Authorities for this set of web pages, and compute the Hub Weights and Authority weights for the 6 nodes. Show the matrix decomposition(s) used to obtain these weights.

(d) Change the pointers so that F points to A instead of B, and recompute the Page Rank and the Hub/Authority weights. What difference does this make, especially to node A? You don't need to show all the decompositions, just the final computed weights.

3. Consider the following code to generate sparse test matrices.

```
function A = lap2D(nx,ny)
%% function A = lap2D(nx,ny)
%% generate the finite-difference Laplacian Operator for an  nx-by-ny  grid
tx= -spdiags(ones(nx,3),[-1:1],nx,nx) + 3*speye(nx,nx);
ty= -spdiags(ones(ny,3),[-1:1],ny,ny) + 3*speye(ny,ny);
A = kron(speye(ny,ny),tx) + kron(ty,speye(nx,nx));
```

(a) Illustrate what this script does with a small case, e.g. `nx` = 3, `ny` = 2.

(b) Generate the matrix `lap2D(20,20)` and show its sparsity pattern (with the `spy` function).

(c) Use the Gauss-Seidel iteration to solve the linear system $Ax = b = [1, \cdots, 1]^T$ for the matrix from 3b. Your code should work in sparse mode, so make sure all your matrices generated and used in your script are sparse. You can use the `whos` command to see if everything is sparse. Use the `tril`, `triu`, `diag`, functions to extract the lower, upper, and diagonal parts of the matrix. You should stop the iteration when either $\|b - Ax\|_2 < 10^{-5}\|b\|_2$, or when the number of iterations exceeds 200.

Hand in your code, and a diary of the output showing the number of iterations and the final answer, as well as the result of the `whos` command showing all the matrices are sparse (the single vectors need not be sparse).

(d) Repeat the previous question, but this time use SOR with $\omega = 1.740580010738572$. Compare the number of iterations needed here with the number needed for Gauss-Seidel ($\omega = 1$).