CS 621: Matrix Computations
Fall 2001
Problem Set 6

Handed out: Mon., Nov. 19.

Due: Mon., Dec. 3 in lecture.

1. Let T be a real symmetric tridiagonal matrix, λ an eigenvalue of T, and σ a very accurate approximation to λ. Assume that the multiplicity of λ is 1, and that there is no other eigenvalue very close to λ. We would like to apply a single step of the shifted inverse power method (i.e., solve $(T - \sigma I)\mathbf{x} = \mathbf{b}$ for $\mathbf{x}$) to obtain in $O(n)$ steps a good approximation $\mathbf{x}$ to the eigenvector of λ. The issue is how to choose $\mathbf{b}$ so that the computation works correctly in floating point arithmetic.

   Suppose that $\mathbf{v}$ is the true eigenvector corresponding to λ. Show that this algorithm will work with $\mathbf{b}$ chosen to be any column $\mathbf{e}_i$ of the identity matrix, provided that $\mathbf{v}(i)$ is nonzero. An especially good choice would be $\mathbf{b} = \mathbf{e}_i$, where i is chosen to maximize $|S(i,i)|$ over $i = 1 : n$, where $S = (T - \sigma I)^{-1}$. Explain why.

   [Note: This question was analyzed by Wilkinson and continues to play a role today in algorithms for finding eigenvectors of tridiagonal matrices.]

2. In lecture it was claimed that Arnoldi is equivalent to QR factorization of

$$[\mathbf{q}_1, A\mathbf{q}_1, \ldots, A^{l-1}\mathbf{q}_1].$$

   Show how to obtain the R factor of this QR factorization from the results of the Arnoldi algorithm.

3. Explain how to extend the Arnoldi algorithm so that on the kth iteration, it explicitly computes (with a finite algorithm) the polynomial $p(z) \in \mathcal{P}^l$ described in lecture that solves the Arnoldi approximation problem OPT. The polynomial should be represented via its $l + 1$ coefficients.

   [Hint: In lecture it was shown how to write the solution to the approximation problem in the form $A^l\mathbf{q}_1 + Q_l\mathbf{f}$. Suppose for each i that the ith column of Q_l could be written as $p_i(A)\mathbf{q}_1$ for some polynomial p_i; then we could take a linear combination of the p_i's to obtain the desired p. (Why?) The p_l's can in turn be computed recurrently from the Arnoldi iteration.]

4. The Lanczos and Arnoldi algorithms are identical in exact arithmetic if applied to a real symmetric matrix, but in the presence of roundoff they may give different answers. Implement both algorithms, and compare the performance of the two algorithms on symmetric real matrices. Use the same $\mathbf{q}_1$ for both algorithms, which should be chosen randomly. Run them for the same number of iterations. (Large sparse symmetric matrices can be created in with the `sparse` function.) In particular, compute the Ritz

values in both cases and compare them to the true eigenvalues, determining which algorithm returns better approximations to the true eigenvalues. For very large matrices, the true eigenvalues may be too difficult to compute, in which case you can just compare Arnoldi to Lanczos. Try the special case of a very large diagonal matrices with some repeated eigenvalues. (The `sparse` function can create large diagonal matrices if used correctly. The reason for using a diagonal matrix is that the eigenvalues are known in advance.) It is known that Lanczos may sometimes erroneously produce multiple Ritz values corresponding to the same eigenvalue because of roundoff, whereas Arnoldi should not have this problem.

Hand in listings of your m-files and a plot of your results.