MATH 5620 NUMERICAL ANALYSIS II
HOMEWORK 5, DUE APRIL 2 2010

Problem 1 B&F 9.1.2, 9.1.4:
- Please do these eigenvalue calculations by hand
- In at least one case you need to find an orthonormal basis for a subspace. You can do this using Gram-Schmidt.

Problem 2 B&F 9.2.2 a,b and 9.2.8 a,b (power method)
Problem 3 B&F 9.2.6 a,c and 9.2.12 a,c (symmetric power method)
Problem 4 B&F 9.2.6 a,c and 9.2.12 a,c, but use Rayleigh Quotient Iteration (page 63 in the class notes) instead of the symmetric power method.
Problem 5 B&F 9.4.2 a,c and 9.4.4 a,c (QR Algorithm). Please **do not** implement Algorithm 9.6 in B&F. A simpler algorithm is outlined below.

Here are some implementation guidelines:

- Please display at least 7 digits of precision (or use `format long`)
- In **Problems** 2–5 I expect to see a few iterates of the method in question on the specified matrix and then the approximation to the eigenvalues (and eigenvectors in **Problems** 2–4). You can always check that your eigenvalues are correct using Matlab's command `eig` (try `[V,D]=eig(A);`). Of course the eigenvectors you get maybe different from those given by `eig` because of normalization or signs.
- To check convergence in **Problems** 2–4 simply determine if two consecutive iterates (eigenvalues or eigenvectors) are within the desired tolerance (this is not an industrial strength convergence criterion!)
- In **Problem** 5: Please use directly Matlab's `qr` function. The algorithm in p73 (notes) needs some kind of deflation technique to work well. Deflation simply means that when the desired accuracy of an eigenvalue is met the procedure continues with the next eigenvalue. To detect convergence of the entry $T(m,m)$ to an eigenvalue of T its enough to require that $|T(m,m-1)| < TOL$. If $T(m,m)$ is within the desired tolerance, we can set $m = m-1$ and continue with $T(1:m,1:m)$. See next page for a suggested implementation.

Given an $n \times n$ tridiagonal matrix T and a prescribed tolerance ϵ here is a practical shifted QR algorithm:

```
m = n
for k=0,1,...
  Determine shift μ   (* Wilkinson shift recommended *)
  QR = T(1:m,1:m) - μI  (* QR factorization *)
  T(1:m,1:m) = RQ + μI
  (* check convergence of T(m,m) to an eigenvalue of T *)
  if |A(m,m-1)| < ε then
    if m > 2 then
      (* go to next eigenvalue *)
      m = m - 1
    else
      (* all eigenvalues have converged to within prescribed
           tolerance *)
      print 'done in' k 'iterations'
      stop
    end if
  end if
end for
```

- For determining the shift μ please use the Wilkinson shift (the one given in the notes can have problems in pathological cases):

```
d = 1/2 (A(m-1,m-1) - A(m,m))
if d > 0
  μ = A(m,m) + d - sqrt(d^2 + A(m,m-1)^2)
else
  μ = A(m,m) + d + sqrt(d^2 + A(m,m-1)^2)
end if
```