Instructor: Emre Mengi

Math 504 (Fall 2011)

Study Guide for Weeks 11-14
This homework concerns the following topics.

- Basic definitions and facts about eigenvalues and eigenvectors (Trefethen&Bau, Lecture 24)
- Similarity transformations (Trefethen&Bau, Lecture 24)
- Power, inverse and Rayleigh iterations (Trefethen&Bau, Lecture 27)
- QR algorithm with and without shifts (Trefethen&Bau, Lecture 28&29)
- Simultaneous power iteration and its equivalence with the QR algorithm (Trefethen&Bau, Lectures 28&29)
- The implicit QR algorithm
- The Arnoldi Iteration (Trefethen&Bau, Lectures 33&34)
- GMRES (Trefethen&Bau, Lecture 35)

Homework 5 (Assigned on Dec 26th, Mon; Due on Jan 10th, Tue by 11:00)
Please turn in the solutions only to the questions marked with (*). The rest is to practice on your own. Attach Matlab output, m-files, print-outs whenever they are necessary. The Dec 10, 11:00 deadline is tight.

1. (*) Consider the matrices

$$A_1 = \begin{bmatrix} 1 & 7 \\ 3 & 5 \end{bmatrix} \quad \text{and} \quad A_2 = \begin{bmatrix} 2 & 4 & -5 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{bmatrix}$$

(a) Find the eigenvalues of A_1 and the eigenspace associated with each of its eigenvalues.

(b) Find the eigenvalues of A_2 together with their algebraic and geometric multiplicities.

(c) Find a Schur factorization for A_1.

(d) Let v_0 and v_1 be two linearly independent eigenvectors of A_1. Suppose also that $\{q_k\}$ denotes the sequence of vectors generated by the inverse iteration with shift $\sigma = 2$ and starting with an initial vector $q_0 = \alpha_0 v_0 + \alpha_1 v_1 \in \mathbb{C}^2$ where α_0, α_1 are nonzero scalars.

Determine the subspace that $\text{span}\{q_k\}$ is approaching as $k \to \infty$.

2. Consider the matrices

$$B_1 = \begin{bmatrix} 1 & 1 & 1 \\ 0 & -2 & 1 \\ 0 & 0 & -1 \end{bmatrix} \quad \text{and} \quad B_2 = \begin{bmatrix} 2 & -1 \\ -3 & 0 \end{bmatrix}.$$

(a) Write down the characteristic polynomials for B_1, B_2 and calculate their eigenvalues.

(b) Find the eigenspace associated with each eigenvalue of B_1.

(c) Recall that power iteration (generically) converges to the dominant eigenvector associated with the eigenvalue with largest modulus. Suppose that power iteration is applied to B_1 and B_2. For which of B_1 and B_2 would you expect the convergence to the dominant eigenvector to be faster? Explain.

(d) Write down the companion matrix $\mathcal{C}_1$ whose eigenvalues are same as the roots of the polynomial $p_1(z) = z^3 + 2z^2 - z - 2$.

3. Consider the infinite sequence of integers

$$\{f_0, f_1, f_2, \dots\}$$

where $f_0 = -9$, $f_1 = 2$ and for $n \geq 2$

$$f_n = 0.9f_{n-1} + 0.1f_{n-2}.$$

(a) Define $F_n = \begin{bmatrix} f_n \\ f_{n-1} \end{bmatrix}$. Find a 2×2 matrix such that for $n \geq 2$

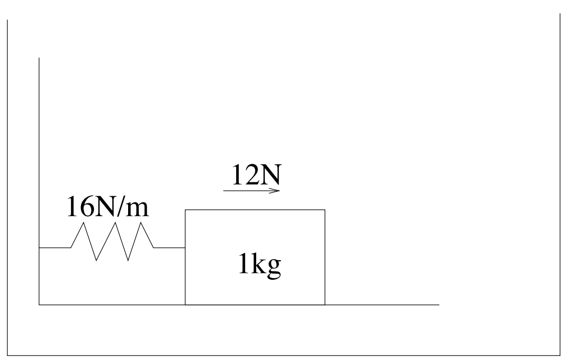
$$F_n = AF_{n-1}.$$

(b) Find the eigenvalues and an eigenvector associated with each of the eigenvalues of A.

(c) Write down $F_1 = \begin{bmatrix} f_1 \\ f_0 \end{bmatrix}$ as a linear combination of the eigenvectors of A.

(d) Using your answers to parts **(a)-(c)** find a general formula for f_n in terms of n.

4. (Watkins 2nd Ed, Exercise 5.1.22, page 302) (*) Consider a mass attached to a wall by a spring as shown below. At time zero the mass is at rest at its equilibrium position $x = 0$.



At that moment a steady force of 12 newtons applied, pushing the cart to the right. Assume that the rolling friction is $-k\dot{x}(t)$ newtons. Do parts **(a)-(c)** by hand.

(a) Set up a system of first-order differential equations of the form $\dot{x} = Ax - b$ for the motion of the mass.

(b) Find the characteristic polynomial of A and solve it by the quadratic formula to obtain an expression (involving k) for the eigenvalues of A.

(c) There is a critical value of k at which the eigenvalues of A change from real to complex. Find this critical value.

(d) Using Matlab solve the initial value problem for the cases **(i)** $k = 2$, **(ii)** $k = 6$, **(iii)** $k = 10$, and $k = 14$. Rather than reporting your solutions, simply plot $x_1(t)$ for $0 \leq t \leq 3$ for each of your solutions on a single set of axes. Comment on your plot. In particular how fast is the rate of decay, does the motion exhibit oscillations or not?

5. A matrix A is called *normal* if it satisfies the property $A^*A = AA^*$. Show that a matrix $A \in \mathbb{C}^{n\times n}$ is *normal* if and only if it has a factorization of the form

$$A = Q\Lambda Q^*$$

where $Q \in \mathbb{C}^{n\times n}$ is unitary and $\Lambda \in \mathbb{C}^{n\times n}$ is diagonal.

6. Suppose $A \in \mathbb{R}^{n\times n}$ has distinct eigenvalues. Denote the eigenvalues of A by $\lambda_1, \lambda_2, \ldots, \lambda_n$ and the associated eigenvectors by $v_1, v_2, \ldots, v_n$. Since A has distinct eigenvalues, $\lambda_j \neq \lambda_k$ for all j, k such that $j \neq k$. For simplicity assume that the eigenvalues and eigenvectors are real, that is $\lambda_j \in \mathbb{R}$, $v_j \in \mathbb{R}^n$ for $j = 1, \ldots, n$.

Show that the set of eigenvectors $\{v_1, v_2, \ldots, v_n\}$ is linearly independent, that is the vector equation

$$c_1v_1 + c_2v_2 + \cdots + c_nv_n = 0$$

with $c_1, c_2, \ldots, c_n \in \mathbb{R}$ holds only for $c_1 = c_2 = \cdots = c_n = 0$. (Hint: Show by induction that all subsets $\{v_1, \ldots, v_j\}$ are linearly independent for $j = 1, \ldots, n$.)

7. Suppose $A \in \mathbb{R}^{n\times n}$ is a symmetric matrix, that is $A^T = A$. Show that A is positive definite (*i.e.* $x^TAx > 0$ for all non-zero $x \in \mathbb{R}^n$) if and only if all of the eigenvalues of A are positive.
(Note: A symmetric matrix is normal; consequently from question 5 it is unitarily diagonalizable. A symmetric matrix also has real eigenvalues.)

8. Consider the sequence of real numbers $\{x_k\}$ defined recursively as

$$x_{k+1} = \frac{x_k}{2} + \frac{5}{2x_k}$$

for $k = 0, 1, 2, ...$ given an x_0. It can be shown that if x_0 is sufficiently close to $\sqrt{5}$, then

$$\lim_{k\to\infty} x_k = \sqrt{5}.$$

Show that the rate of convergence is *q-quadratic* when the sequence converges to $\sqrt{5}$.

9. Suppose that the power iteration is applied to a Hermitian matrix $A \in \mathbb{C}^{n\times n}$ such that $|\lambda_1| = |\lambda_2| > |\lambda_3|$ where $\lambda_1, \lambda_2, \lambda_3$ denote the largest three eigenvalues of A in modulus. Would you expect the power iteration to converge in exact arithmetic? If it converges, which vector does it converge? What is the q-rate of convergence? Explain.

10. (*) Implement a Matlab routine `rayleigh_iter.m` to compute an eigenvalue λ and an associated eigenvector v of an $n \times n$ matrix A by Rayleigh iteration.

Your routine must return two output arguments, the computed eigenvalue λ and the associated eigenvector v. It must take one input argument, the $n \times n$ matrix A for which an eigenvalue and an eigenvector are sought. You can choose a randomly generated vector (by typing the command `randn(n,1)`) as your initial estimate $q_0 \in \mathbb{C}^n$ for the eigenvector. Display the estimate for the eigenvalue at each iteration (by typing `display(q'*A*q)` where `q` is the current estimate for the eigenvector with 2-norm equal to one) so that you can observe the rate of convergence. You should terminate when the eigenvector estimates q_k and q_{k+1} at two consecutive iterations are sufficently close to each other, *e.g.* $\|q_k - q_{k+1}\| \leq 10^{-15}$.

Test your implementations with the following matrices.

$$B_1 = \begin{bmatrix} 1 & 0 & 1/10 \\ 0 & -0.8 & 0 \\ 1/10 & 0 & 1 \end{bmatrix}, \quad B_2 = \begin{bmatrix} 1 & 0 & 1/100 \\ 0 & -0.8 & 0 \\ 1/100 & 0 & 1 \end{bmatrix}$$

Run your routine to compute an eigenvalue and an associated eigenvector for B_1, B_2. Run it several times (*e.g.* six, seven times). Does it always converge to the same eigenvalue? What kind of rate of convergence do you observe in practice?

11. (*) The QR algorithm is one of the standard approaches to compute the eigenvalues of a matrix $A \in \mathbb{R}^{n\times n}$. Below a pseudocode is provided for the explicit QR algorithm. It generates a sequence of matrices $\{A_k\}$ that usually converges to an upper triangular matrix in the limit as $k \to \infty$.

Algorithm 1 Explicit QR Algorithm without Shifts

```
A_0 ← A
for k = 0, 1, ... do
    Compute a QR factorization A_k = Q_{k+1} R_{k+1}
    A_{k+1} ← R_{k+1} Q_{k+1}
end for
```

The QR Algorithm converges only linearly. To speed-up its convergence one can use the following variant given below with shifts. Below in both parts perform the calculations by hand.

(a) Apply one iteration of the QR algorithm to the matrix A provided below. (Note : $\lambda_1 = 5$ and $\lambda_2 = 1$ are the eigenvalues of A.)

$$A = \begin{bmatrix} 3 & 1 \\ 4 & 3 \end{bmatrix}$$

Algorithm 2 The QR Algorithm with Shifts

$A_0 \leftarrow A$
for $k = 0, 1, \ldots$ **do**
  Choose a shift μ_k
  Compute a QR factorization $A_k - \mu_k I = Q_{k+1} R_{k+1}$
  $A_{k+1} \leftarrow R_{k+1} Q_{k+1} + \mu_k I$
end for

(b) Apply one iteration of the QR algorithm to the matrix A given in **(a)** with shift $\mu = 6$. Do you particularly observe (or should you expect) that the QR algorithm with this shift converge faster than the QR algorithm without shift? Why or why not?

12. The explicit QR algorithm without shifts (Algorithm 1 in the previous question) is equivalent to the simultaneous power iteration. Pseudocode for the simultaneous power iteration is given below.

Algorithm 3 Simultaneous Power Iteration

for $k = 1, \ldots, m$ **do**
  Compute a QR factorization $A^k = \hat{Q}_k \hat{R}_k$
  $\hat{\Lambda}_k \leftarrow \hat{Q}_k^T A \hat{Q}_k$
end for

Implement both the explicit QR Algorithm and simultaneous power iteration. Test them on the matrix

$$A = \begin{bmatrix} 3 & -1 & -2 \\ -1 & 3 & -1 \\ -2 & -1 & 3 \end{bmatrix}.$$

In particular verify that $A_3 = \hat{\Lambda}_3$. Repeat this test for a few random square matrices of small size. (Note that the simultaneous power iteration as above is unstable; you should not work on it with large m.)

13. (*) In this question you are expected to shed a light into the relation between the QR algorithm and simultaneous iteration. Pseudocodes are provided in the previous questions for the QR algorithm (Algorithm 1) as well as for the simultaneous iteration (Algorithm 3).

Show that a QR factorization for A^k is given by

$$A^k = \underbrace{Q_1 Q_2 \ldots Q_k}_{\hat{Q}_k} \underbrace{R_k \ldots R_2 R_1}_{\hat{R}_k}.$$

(Hint: First try to express A_k in Algorithm 1 in terms of A and Q_j for $j = 1, \ldots, k$.)

14. (*) The purpose of this question is to establish the equivalence of the simultaneous power iteration to its normalized variant, for which a pseudocode is provided below. Show that $A^k = \tilde{Q}_k R_k$ for some upper triangular matrix R_k.

Algorithm 4 Normalized Simultaneous Power Iteration

$Z_1 \leftarrow A$
for $k = 1, \ldots, m$ **do**
  Compute a QR factorization $Z_k = \tilde{Q}_k \tilde{R}_k$
  $\tilde{\Lambda}_k \leftarrow \tilde{Q}_k^* A \tilde{Q}_k$
  $Z_{k+1} \leftarrow A\tilde{Q}_k$
end for

15. When the QR algorithm is applied to $A \in \mathbb{C}^{n \times n}$, the sequence of matrices generated (generically) converge to

$$\tilde{A} = \begin{bmatrix} A_1 & B \\ 0 & A_2 \end{bmatrix}$$

where $A_1 \in \mathbb{C}^{m \times m}$, $A_2 \in \mathbb{C}^{(n-m)\times(n-m)}$. Then the QR algorithm continues to iterate on A_1 and A_2. (Typically in practice when Wilkinson shifts are used, the matrix A_1 is $(n-1)\times(n-1)$ or $(n-2)\times(n-2)$. Therefore A_2 is either a scalar corresponding to an eigenvalue of A or A_2 is 2×2 corresponding to a conjugate pair of eigenvalues.) This process of repeating the QR algorithm on smaller matrices is called the *deflation*.

Show that λ is an eigenvalue of A if and only if λ is eigenvalue of A_1 or A_2.

16. (*) The aim in this question is to implement the implicit QR algorithm with Wilkinson shifts for the solution of dense eigenvalue problems. Your implementation should at least work on real symmetric matrices.

(a) Implement one iteration of the implicit QR algorithm with a shift on a Hessenberg matrix. In effect for any shift μ and a Hessenberg matrix A_k your routine must perform

$$A_{k+1} = R_{k+1}Q_{k+1} + \mu I \ \text{ where } \ (A_k - \mu I) = Q_{k+1}R_{k+1}.$$

But you must do this implicitly via chase bulging. Make sure that the computational cost is $O(n^2)$. Name your routine as `qrIteration`.

(b) Implement the QR Algorithm with Wilkinson shifts using your routine `qrIteration` from part **(a)**. For initial reduction to Hessenberg form use the built-in Matlab routine `hess`. Your implementation must use deflations, that is if any of the subdiagonal entries is sufficiently close to zero, your routine must start solving smaller eigenvalue problems. I suggest to implement a recursive routine (this would keep your implementation much simpler) such as

```
function eigvals = myqrAlgorithm(A)

A = hess(A);
eigvals = myqrSub(A);

return;
```

```
function eigvals = myqrSub(A)

[n, n1] = size(A);

if (n == 1)
    eigvals = [A];
    return;
end

if (n == 2)
   lambda1 = % one eigenvalue of A using the discriminant formula
   lambda2 = % the other eigenvalue of A
   eigvals = [lambda1; lambda2];
   return;
end

for iter = 1:100 do
    % Calculate the Wilkinson shift and set mu to the Wilkinson shift
    A = qrIteration(A,mu);

    % If any of the subdiagonal entries, say a_(j+1)j, is less than 10^-10
    %    eigvals1 = myqrSub(A(1:j,1:j);
    %    eigvals2 = myqrSub(A(j+1:n,j+1:n);
    %    eigvals = [eigvals1; eigvals2];
    %    return;
end

error('too many iterations: QR algorithm could not converge');
return
```

Test your implementation with various 10×10 random matrices. Compare the eigenvalues returned by your routine with the eigenvalues returned by the built-in solver `eig` in Matlab.

17. (Trefethen&Bau, Exercise 33.1) Let $A \in \mathbb{C}^{m \times m}$ and $b \in \mathbb{C}^m$ be arbitrary. Show that any $x \in \mathcal{K}_n = \text{span}\{b, Ab, \ldots, A^{n-1}b\}$ is equal to $p(A)b$ for some polynomial p of degree $\leq n-1$.

18. (Trefethen&Bau, Exercise 33.2) Suppose that at the nth iteration of the Arnoldi's algorithm the Hessenberg matrix $\tilde{H}_{(n+1)n}$ is such that $h(n+1)n = 0$.

(a) Simplify the Arnoldi iteration

$$AQ_n = Q_{n+1}\tilde{H}_n.$$

What does this imply about the structure of a full $m \times m$ Hessenberg reduction $A = QHQ^*$ of A?

(b) Show that the Krylov subspace $\mathcal{K}_n$ is an invariant subspace of A, i.e., $A\mathcal{K}_n \subseteq \mathcal{K}_n$.

(c) Show that $\mathcal{K}_n = \mathcal{K}_j$ for all $j > n$.

(d) Show that each eigenvalue of H_n is an eigenvalue of A.

(e) Show that if A is nonsingular, then the solution x to the system of equations $Ax = b$ lies in $\mathcal{K}_n$.

The appearance of an entry $h_{(n+1)n} = 0$ is called a *breakdown* of the Arnoldi iteration, but it is a breakdown of a benign sort. For application in computing eigenvalues or solving linear systems (i.e., GMRES), because of **(d)** and **(e)**, a breakdown usually means that convergence has occured and the iteration can be terminated.

19. (Trefethen&Bau, Exercise 34.3) Let A be the $N \times N$ bidiagonal matrix with $a_{k(k+1)} = a_{kk} = 1/\sqrt{k}$, $N = 64$. (In the limit $N \to \infty$, A becomes a non-self-adjoint compact operator.)

(a) Produce a plot showing the spectrum $\Lambda(A)$ (i.e., the set of eigenvalues of A).

(b) Implement the Arnoldi iteration. Starting from a random initial vector, run the Arnoldi iteration and compute Ritz values at steps $n = 1, \ldots, 30$. Plot the Ritz values for $n = 10, 20, 30$ on the same plot as $\Lambda(A)$.

20. (Trefethen&Bau, Exercise 35.3)The recurrence

$$x_{n+1} = x_n + \alpha r_n = x_n + \alpha(b - Ax_n),$$

where α is a constant scalar, is known as *Richardson iteration*. What polynomial $p(A)$ at step n does this correspond to?

21. (Trefethen&Bau, Exercise 35.4) (*) A 2×2 Givens rotation is of the form

$$J = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}.$$

The effect of the transformation $x \to Jx$ on the vector $x \in \mathbb{R}^2$ is a rotation in the clock-wise direction by an angle of θ.

(a) Describe an algorithm with computational complexity $O(n^2)$ based on 2×2 Givens rotations to solve the least squares problem resulting at the nth iteration of GMRES.

(b) Show how the operation count at the nth iteration can be improved to $O(n)$ if the least squares problem at iteration $n-1$ have already been solved by using Givens rotations.

22. (Trefethen&Bau, Exercise 35.5) The standard description of the GMRES algorithm for the solution of $Ax = b$ begins with the initial guess $x_0 = 0$, and the initial residual $r_0 = b$. Describe how the algorithm can be modified for an arbitrary initial guess x_0.

23. Suppose that a matrix $A \in \mathbb{C}^{m\times m}$ has the eigenvalue decomposition

$$A = V \begin{bmatrix} \lambda_1 I & 0 & \dots & 0 \\ 0 & \lambda_2 I & & 0 \\ & & \ddots & \\ 0 & & & \lambda_n I \end{bmatrix} V^{-1}$$

where $n \leq m$.

(a) Show that the Arnoldi iteration would converge to all of the eigenvalues of A after n iterations.

(b) Show that the GMRES algorithm would converge to the exact solution of the linear system $Ax = b$ after n iterations for all $b \in \mathbb{C}^m$.