Math 504, Fall 2018 - Homework 4

December 13, 2018

This homework involves only one computational question that expects an implementation of the QR Algorithm for Hermitian matrices stated in detail below.

Solution of Dense Hermitian Eigenvalue Problems by the QR Algorithm

The QR algorithm is the most-widely employed algorithm to compute all eigenvalues of a dense matrix $A \in \mathbb{C}^{n \times n}$.

The aim in this question is to implement the QR algorithm with Wilkinson shifts for the solution of dense eigenvalue problems. Assume in all parts that the input matrix A is *Hermitian*.

Algorithm 1 The QR Algorithm with Shifts

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

1. Since A is Hermitian, its Hessenberg form $H = Q^*AQ$ for some unitary $Q \in \mathbb{C}^{n\times n}$ is also Hermitian, that is its Hessenberg form H is tridiagonal.

   Design a numerical algorithm to compute the QR factorization of a tridiagonal matrix by 2×2 Householder reflectors. It is essential that your algorithm requires $O(n)$ flops. Your Matlab routine should look like

```
function [U,R] = qrTriD(A)

for k = 1:n-1
     % Form Householder vector v for the kth column
     U(1:2,k) = v;
     % Apply Orthogonal Transformation to A
 end

 R = A;
 return
```

   At termination U must be a matrix of size $2 \times (n-1)$. Do not form the Q factor of $A = QR$ explicitly. Q could be constructed from U, but this is not necessary for the QR algorithm. Test your implementation as follows.

   (i) Create a random matrix : `A5 = randn(5);`

   (ii) Form the Hermitian part of the random matrix : `HA5 = (A5 + A5')/2;`

   (iii) Calculate its tridiagonal (Hessenberg) form : `TD5 = hess(HA5);`

   (iv) Compute the QR factorization : `[U,R5] = qrTriD(TD5);`

   (v) Check the Q factor : `Q = TD5/R5;`

   (vi) See how close it is to orthogonality : `norm(eye(5) - Q'*Q)`

   The norm at the last step must be close to $\epsilon_{\text{mach}} = 2^{-53} \approx 1.11 \cdot 10^{-16}$.

2. Implement one iteration of the QR algorithm with shifts working on tridiagonal matrices. In other words, in Algorithm 1 above, given a tridiagonal matrix A_k, your Matlab routine must produce A_{k+1} (which remains tridiagonal). First you must call `qrTriD` and compute the QR factorization of $A_k - \sigma_k I$. Then, using the returned output parameters U and R, you must perform $RQ + \sigma_k I$. It is essential that your implementation requires $O(n)$ flops for the multiplication RQ. Here you need to exploit the special structures of Q, specifically the fact that it is made of 2×2 Householder reflectors, as well as R which turns out to be bidiagonal. Your Matlab routine should be of the following form.

```
function Anew = qrIteration(A,mu)

[n,n1] = size(A);
[U,R] = qrTriD(A - mu*eye(n));

% Perform Anew = R*Q + mu*eye(n) without forming Q,
% instead by utilizing U

return
```

3. Implement the QR Algorithm with Wilkinson shifts using your routine `qrIteration` from part 2 above. For initial reduction to tridiagonal 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.

   Test your implementation on various Hermitian matrices, for instance on the ones available through the Matlab routine `gallery` or on random matrices. For a matrix A, that is not Hermitian, you could perform the experiments on the Hermitian part $(A + A^*)/2$. Compare the eigenvalues returned by your routine with the eigenvalues returned by the routine `eig` in Matlab. For your convenience, a pseudocode of the overall algorithm is provided on the next page.

% Stage 1
Compute a tridiagonal $H \in \mathbb{C}^{n\times n}$ s.t.

$$H = Q^* A Q$$

for some unitary $Q \in \mathbb{C}^{n\times n}$.
% Stage 2
if H is 1×1 or 2×2 **then**
  $\Lambda \leftarrow$ eigenvalues of H calculated using algebraic formulas
  Return Λ
else
  repeat
    Choose the Wilkinson shift σ
    Compute a QR factorization $H - \sigma I = QR$
    $H \leftarrow RQ + \sigma I$
    if H is of the form $H = \begin{bmatrix} H_1 & 0 \\ 0 & H_2 \end{bmatrix}$
     for some $H_1 \in \mathbb{C}^{k\times k}$, $H_2 \in \mathbb{C}^{(n-k)\times(n-2)}$ with $k \in [1, n-1]$ **then**
      $\Lambda_1 \leftarrow$ Apply Stage 2 on H_1.
      $\Lambda_2 \leftarrow$ Apply Stage 2 on H_2.
      $\Lambda \leftarrow \begin{bmatrix} \Lambda_1 \\ \Lambda_2 \end{bmatrix}$
      Return Λ
    end if
  until
end if

Additional theoretical questions about eigenvalue, singular value computations, as well as Krylov subspace methods are provided below. Solutions to these are not going to be collected or evaluated.

1. Let $v^{(1)}$ and $v^{(2)}$ be two linearly independent eigenvectors of

$$B = \begin{bmatrix} 1 & 7 \\ 3 & 5 \end{bmatrix}.$$

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 $q_0 = \alpha_1 v^{(1)} + \alpha_2 v^{(2)} \in \mathbb{C}^2$ where α_1, α_2 are nonzero scalars.

Determine the subspace that $\mathrm{span}\{q^{(k)}\}$ is approaching as $k \to \infty$.

2. 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 absolute value. Would you expect the power iteration to converge in exact arithmetic? What happens to the vectors in the sequence in the limit, and how quickly? Explain.

3. In class, it was shown that a unit vector $q \in \mathbb{C}^{n\times n}$ that is an estimate for a unit eigenvector v of a given matrix $A \in \mathbb{C}^{n\times n}$ associated with an eigenvalue λ satisfies

$$|\lambda - r(q)| \leq 2\|A\|_2 \|v - q\|_2.$$

Here, $r(q) := q^* A q$ is the Rayleigh quotient.

When A is Hermitian, better estimates can be deduced. This is due to the orthogonality of the eigenvectors of A. Denote the orthonormal set of eigenvectors of A with $\{v^{(1)}, v^{(2)}, \dots, v^{(n)}\}$, and the associated eigenvalues with $\lambda_1, \lambda_2, \dots, \lambda_n$. The question concerns, given an estimate $q \in \mathbb{C}^n$ for the eigenvector $v = v^{(1)}$, how good of an estimate $r(q)$ is for the associated eigenvalue $\lambda = \lambda_1$.

(a) The vector q can be expanded as $q = c_1 v^{(1)} + c_2 v^{(2)} + \cdots + c_n v^{(n)}$ for scalars $c_1, \dots, c_n$. Show that $\sum_{k=1}^{n} |c_k|^2 = \|q\|_2^2 = 1$.

(b) Show that $\sum_{k=2}^{n} |c_k|^2 \leq \|v^{(1)} - q\|_2^2$.

(c) Derive an expression for the Rayleigh quotient $r(q) = q^* A q$ in terms of the coefficients c_j and the eigenvalues λ_j.

(d) Show that $|\lambda_1 - r(q)| \leq \kappa \|v^{(1)} - q\|_2^2$, where $\kappa := \max_{k=2,\ldots,n} |\lambda_1 - \lambda_k|$. (Hint: exploit $\lambda_1 = \sum_{k=1}^{n} \lambda_1 |c_k|^2$.)

4. The QR algorithm is the standard approach to compute the eigenvalues of a dense matrix $A \in \mathbb{C}^{n \times n}$. Below pseudocodes are provided for the QR algorithm without and with shifts.

Algorithm 2 The 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
```

Algorithm 3 The QR Algorithm with Shifts

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

(a) Apply one iteration of the QR algorithm without shifts to the matrix

$$A = \begin{bmatrix} 3 & 2 \\ 4 & 1 \end{bmatrix}.$$

(b) Apply one iteration of the QR algorithm to the matrix A given in **(a)** with the shift $\sigma = 6$.

5. In class, it was shown that the (normalized) simultaneous power iteration is equivalent to the QR algorithm. In this question, you are expected to establish the equivalence of the QR algorithm and unnormalized simultaneous power iteration. A pseudocode for the QR Algorithm without shifts (Algorithm 1) is provided in the previous question. A pseudocode for the unnormalized simultaneous power iteration is given below.

Algorithm 4 Unnormalized Simultaneous Power Iteration

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

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

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

Here, Q_k, R_k are as defined in Algorithm 1.

6. Let $A \in \mathbb{C}^{m \times n}$, and

$$B = \begin{bmatrix} 0 & A \\ A^* & 0 \end{bmatrix}.$$

(a) Show that if σ is a singular value of A, then σ and $-\sigma$ are the eigenvalues of B.

(b) Show that if λ is an eigenvalue of B, then $|\lambda|$ is a singular value of A. Write down also the left and right singular vectors of A corresponding to the singular value $|\lambda|$ in terms of the eigenvector v of B associated with the eigenvalue λ.

7. A pseudocode for the QR Algorithm for singular value computation is given below. If the matrix A is initially bidiagonal, then the sequence $\{A_k\}$ remains bidiagonal, which you are expected to show in this question. This reduces the cost of each iteration of the QR algorithm significantly, indeed linear with respect to m and n.

Algorithm 5 The QR Algorithm for Singular Values

$A_0 \leftarrow A$
for $k = 0, 1, \dots$ **do**
  Choose a shift σ_k
  Compute a QR factorization $A_k^* A_k - \sigma_k I = Q_{k+1} R_{k+1}$
  Compute a QR factorization $A_k A_k^* - \sigma_k I = P_{k+1} S_{k+1}$
  $A_{k+1} \leftarrow P_{k+1}^* A_k Q_{k+1}$
end for

(a) Exploiting the identities

$$(A_k A_k^* - \sigma_k I) A_k = A_k (A_k^* A_k - \sigma_k I) \quad \text{and} \quad (A_k^* A_k - \sigma_k I) A_k^* = A_k^* (A_k A_k^* - \sigma_k I),$$

show that $A_{k+1} = S_{k+1} A_k R_{k+1}^{-1}$ and $A_{k+1}^* = R_{k+1} A_k^* S_{k+1}^{-1}$.

(b) Show that if A_k is upper triangular, then so is A_{k+1}.

(c) Show that if A_k^* is Hessenberg, then so is A_{k+1}^*.

Parts **(b)-(c)** above combined imply that if A_k is bidiagonal, then so is A_{k+1}.

8. Let $A \in \mathbb{C}^{n \times n}$ be a matrix whose eigenvalues are sought. Recall the recurrence

$$A Q_k = Q_{k+1} H_{k+1} \tag{0.1}$$

for the Arnoldi process where

$$Q_k = \left[\begin{array}{cccc} q^{(1)} & q^{(2)} & \dots & q^{(k)} \end{array} \right] \in \mathbb{C}^{n \times k}, \quad n \gg k$$

has orthonormal columns that span the Krylov subspace

$$\mathcal{K}_k := \operatorname{span}\{b, Ab, \dots, A^{k-1} b\}, \tag{0.2}$$

and $H_{k+1} \in \mathbb{C}^{(k+1) \times k}$ is a Hessenberg matrix. The eigenvalues of $\widetilde{H}_k = Q_k^* A Q_k \in \mathbb{C}^{k \times k}$ are supposedly good estimates for the extreme eigenvalues of A, and called the *Ritz values.*

Suppose that at the kth step of the Arnoldi process the Hessenberg matrix H_{k+1} is such that its $(k+1, k)$ entry satisfies $h_{(k+1)k} = 0$.

(a) Simplify the Arnoldi recurrence (0.1).

(b) Show that the Krylov subspace $\mathcal{K}_k$ is an invariant subspace of A, that is show that

$$A\mathcal{K}_k := \{Av \mid v \in \mathcal{K}_k\} \quad \subseteq \quad \mathcal{K}_k.$$

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

(d) Show that each eigenvalue of $\widetilde{H}_k$ is an eigenvalue of A.

9. Let us consider the Krylov subspace $\mathcal{K}_k$ once again defined as in (0.2) for a given matrix $A \in \mathbb{C}^{n\times n}$ and a vector $b \in \mathbb{C}^n$. Throughout this question, assume $\dim(\mathcal{K}_k) = k$. In this case the Arnoldi process generates an orthonormal basis $\mathcal{Q}_k := \{q^{(1)}, \dots, q^{(k)}\}$ for $\mathcal{K}_k$. Especially, letting $Q_k = \begin{bmatrix} q^{(1)} & q^{(2)} & \dots & q^{(k)} \end{bmatrix}$, the eigenvalues of the matrix $\widetilde{H}_k = Q_k^* A Q_k$ capture some of the eigenvalues of A well in certain occasions.

To generate the orthonormal basis $\mathcal{Q}_k$ and the matrix $\widetilde{H}_k$, we rely on the recurrence (0.1), and the fact that $\widetilde{H}_k = H_{k+1}(1:k, 1:k)$.

(a) Suppose $A \in \mathbb{C}^{n\times n}$ is Hermitian. Simplify the recurrence (0.1).

(b) Write down an efficient pseudocode to generate Q_{k+1} and $\widetilde{H}_k$ for an Hermitian matrix $A \in \mathbb{C}^{n\times n}$ based on the simplified recurrence in part **(a)**.