Assignment 5: Eigenproblems

Gregory Smetana
ID 1917370
ACM 106a

November 21, 2013

1 Normal matrices

A matrix $A \in \mathbf{C}^{n\times n}$ is normal if $A^*A = AA^*$

a)

Without loss of generality, assume A is an normal upper triangular matrix. We then have $A^*A = AA^*$, with

$$
\begin{aligned}
AA^* &= \begin{pmatrix} \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & & \times & \times & \times \\ & 0 & & \times & \times \\ & & & & \times \end{pmatrix} \begin{pmatrix} \times & & & & \\ \times & \times & & 0 & \\ \times & \times & \times & & \\ \times & \times & \times & \times & \\ \times & \times & \times & \times & \times \end{pmatrix} \\
A^*A &= \begin{pmatrix} \times & & & & \\ \times & \times & & 0 & \\ \times & \times & \times & & \\ \times & \times & \times & \times & \\ \times & \times & \times & \times & \times \end{pmatrix} \begin{pmatrix} \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & & \times & \times & \times \\ & 0 & & \times & \times \\ & & & & \times \end{pmatrix}
\end{aligned} \tag{1}
$$

$$
(AA^*)_{i,j} = \sum_{m=1}^{n} A_{i,m}\overline{A_{j,m}} = (A^*A)_{i,j} = \sum_{m=1}^{n} \overline{A_{m,i}}A_{m,j} \tag{2}
$$

Now we prove that A must be diagonal using induction. As a base case ($k = 1$) we will show the off-diagonal entries in the first row of A are zero by considering the element in the top left corner:

$$
(A^*A)_{1,1} = \sum_{m=1}^{n} \|A_{m,1}\|^2 \tag{3}
$$

Since A is upper triangular, we have $A_{m,1} = 0$ for $m > 1$

$$\sum_{m=1}^{n} \|A_{m,1}\|^2 = \|A_{1,1}\|^2 \tag{4}$$

Now comparing to the other sum,

$$\|A_{1,1}\|^2 = \sum_{m=1}^{n} \|A_{1,m}\|^2 \tag{5}$$

So we must have

$$\sum_{m=2}^{n} \|A_{1,m}\|^2 = 0 \tag{6}$$

Since each term in the sum is positive, we have concluded that $A_{1,m} = 0$ for $m \neq 1$.

Now, as the inductive step, we must show that if the off-diagonal entries in the first k rows of A are zero, the off-diagonal entries in the $(k+1)^{th}$ row are zero. Using a similar approach,

$$(A^*A)_{k+1,k+1} = \sum_{m=1}^{n} \|A_{m,k+1}\|^2 \tag{7}$$

Due to the triangular structure, $A_{m,k+1} = 0$ for $m > k+1$ and $A_{m,k+1} = 0$ for $m \leq k$ due to zero entries in the first k rows. There is only one non-zero term:

$$\sum_{m=1}^{n} \|A_{m,k+1}\|^2 = \|A_{k+1,k+1}\|^2 \tag{8}$$

Comparing with the other representation,

$$\|A_{k+1,k+1}\|^2 = \sum_{m=1}^{n} \|A_{k+1,m}\|^2 \tag{9}$$

$$\sum_{m=k+2}^{n} \|A_{k+1,m}\|^2 = 0 \tag{10}$$

Since the elements in the sum are positive, the off-diagonal entries in the $(k+1)^{th}$ row must be zero. Therefore, we have proved that a normal triangular matrix must also be diagonal.

b)

The Schur Decomposition is

$$A = QTQ^* \tag{11}$$

where $Q \in \mathbf{R}^{n\times n}$ is unitary and $T \in \mathbf{R}^{n\times n}$ is upper triangular.

$$AA^* = QTQ^*QT^*Q^* = QTT^*Q^* \tag{12}$$

$$A^*A = QT^*Q^*QTQ^* = QT^*TQ^* \tag{13}$$

Suppose that A is a normal matrix

$$QTT^*Q^* = QT^*TQ^* \tag{14}$$

$$Q^*QTT^*Q^*Q = Q^*QT^*TQ^*Q \tag{15}$$

$$TT^* = T^*T \tag{16}$$

So if a matrix A is a normal, its Schur form T is normal.

Now we must show that if T is normal, A is normal. If a triangular matrix is normal, it was shown in part a) that it must be diagonal. In this case,

$$A = QDQ^* \tag{17}$$

$$AA^* = QDQ^*Q\overline{D}Q^* = QD^2Q^* \tag{18}$$

$$A^*A = Q\overline{D}Q^*QTQ^* = QD^2Q^* \tag{19}$$

and A is normal. Therefore, a matrix A is normal *if and only if* its Schur form is normal.

c)

First, suppose that A is normal. From the previous question, the Schur decomposition is

$$A = QDQ^* \tag{20}$$

with $Q \in \mathbf{R}^{n\times n}$ unitary and $D \in \mathbf{R}^{n\times n}$ diagonal. This is the eigendecomposition of A. Consider $AQ = QD$:

$$[Aq_1 ... Aq_n] = [\lambda_1 q_1 ... \lambda_n q_n] \tag{21}$$

where q_i are the columns of Q and $\lambda_i = D_{i,i}$. The columns of Q are orthogonal. Thus, A has n orthogonal eigenvectors.

Now suppose that A has n orthogonal eigenvectors. If Q is a matrix of eigenvectors, then A can be represented

$$A = QDQ^* \tag{22}$$

Using the previous result, since the Schur form is normal, the matrix A is normal.

Therefore, a matrix A is normal *if and only if* it has n orthogonal eigenvectors.

2 Inverse iteration and ill-conditioned systems

the inverse iteration algorithm requires the solution of a linear system of the form

$$(A - \sigma)y_{i+1} = x_i \tag{23}$$

The matrix $A - \sigma I$ is ill-conditioned if the shift is very close to an eigenvalue of A. In the current investigation of this issue, the following simplifications are made:

- $A \in \mathbf{R}^{n\times n}$ is a real symmetric matrix with n real eigenvalues $(\lambda_1, ..., \lambda_n)$ and orthonormal eigenvectors $(q_1, ..., q_n)$
- A is well conditioned so that we may solve linear systems of the form $Ay = x$ backward stably
- The eigenvalue λ_1 is simple, with eigenvector q_1

a)

The shift $\sigma = \lambda_1 + \mu$ for some very small $\mu \in \mathbf{R}$. We solve the linear system

$$(A - \sigma I + \delta A)\tilde{y}_{k+1} = x_k \tag{24}$$

for some $\|\delta A\|_2 = O(\epsilon)$. Taking $x_{k+1} = \tilde{y}_{k+1}/\|\tilde{y}_{k+1}\|_2$,

$$(A + \delta A - (\lambda_1 + \mu)I)x_{k+1} = \frac{x_k}{\|\tilde{y}_{k+1}\|} \tag{25}$$

$$(A - \lambda_1 I)x_{k+1} = (\mu I - \delta A)x_{k+1} + \frac{x_k}{\|\tilde{y}_{k+1}\|_2} \tag{26}$$

Taking the norm of both sides,

$$\|(A - \lambda_1 I)x_{k+1}\|_2 = \|(\mu I - \delta A)x_{k+1}\|_2 + \frac{\|x_k\|_2}{\|\tilde{y}_{k+1}\|_2} \tag{27}$$

Since $\|x_{k+1}\|_2 = \|x_k\|_2 = 1$, and $\|\delta A x_{k+1}\|_2 \leq \|\delta A\|_2 \|x_{k+1}\|_2$ we have

$$\boxed{\|(A - \lambda_1 I)x_{k+1}\|_2 \leq (|\mu| + \|\delta A\|_2) + \frac{1}{\|\tilde{y}_{k+1}\|_2}} \tag{28}$$

b)

A is symmetric, so its set of eigenvectors $\{q_1 ... q_n\}$ forms an orthonormal basis for $\mathbf{R}^n$. Therefore, we may decompose x_k and x_{k+1} as

$$x_k = \sum \alpha_i q_i \tag{29}$$

$$\tilde{y}_{k+1} = \sum \beta_i q_i \tag{30}$$

Plugging this into Equation 24,

$$(A - \lambda_1 I) \sum \beta_i q_i - \mu \sum \beta_i q_i - \delta A \sum \beta_i q_i = \sum \alpha_i q_i \tag{31}$$

Since $Aq_i = \lambda_i q_i$, we have

$$\sum \beta_i (\lambda_i - \lambda_1) q_i - \mu \sum \beta_i q_i - \delta A \sum \beta_i q_i = \sum \alpha_i q_i \tag{32}$$

Multiplying on the left by q_1^T, we have

$$-\mu \beta_1 - q_1^T \delta A \tilde{y}_{k+1} = \alpha_1 \tag{33}$$

Taking norms,

$$|\mu||\beta_1| + \|\delta A\|_2 \|y_{k+1}\|_2 \geq |\alpha_1| \tag{34}$$

Since $\beta_1 \leq \sqrt{\beta_1^2 + ... + \beta_n^2} = \|\tilde{y}_{k+1}\|_2$,

$$\boxed{\|\tilde{y}_{k+1}\|_2 \geq \frac{|\alpha_1|}{|\mu| + \|\delta A\|_2}} \tag{35}$$

3 QR iteration with shifts

a)

To perform QR iteration with shifts, we update A_i by first factorizing $(A_i - \sigma_i I) = Q_i R_i$ and then setting $A_{i+1} = R_i Q_i + \sigma_i I$

$$\begin{aligned} A_{i+1} &= R_i Q_i + \sigma_i I \\ &= Q_i^* Q_i R_i Q_i + \sigma_i Q_i^* Q_i \\ &= Q_i^* (Q_i R_i + \sigma_i I) Q_i \\ &= Q_i^* A_i Q_i \end{aligned} \tag{36}$$

We start with $A_1 = A$, so we may express A_{i+1} with the unitary matrix $\overline{Q}_i = Q_1 Q_2 ... Q_i$ so that

$$\boxed{A_{i+1} = \overline{Q}_i^* A \overline{Q}_i} \tag{37}$$

b)

A matrix H is in upper Hessenberg form if all entries below the first subdiagonal of H are equal to zero, i.e., $H_{i,j} = 0$ for all $i \geq j+2$.

$$(HT)_{i,j} = \sum_{m=1}^{n} H_{i,m} T_{m,j} \tag{38}$$

Considering the structure of the matrices, we have $H_{i,m} = 0$ for $m < i-1$ for upper Hessenberg H and $T_{m,j} = 0$ for $j < m$ for upper triangular T. With these two inequalities, we see that for $i \geq j+2$, each term in the sum is zero. Thus, the product TH is upper Hessenberg.

$$(TH)_{i,j} = \sum_{m=1}^{n} T_{i,m} H_{m,j} \tag{39}$$

Again considering the structure of the matrices, we have $H_{m,j} = 0$ for $m > j+1$ for upper Hessenberg H and $T_{i,m} = 0$ for $i > m$ for upper triangular T. With these two inequalities, we see that for $i \geq j+2$, each term in the sum is zero. Thus, the product HT is upper Hessenberg.

c)

First recall that in the QR decomposition $A = QR$, the first k columns of Q form an orthonormal basis for the span of the first k columns of A. We now consider the factorization

$$A_i - \sigma_i I = Q_i R_i \tag{40}$$

with upper Hessenberg A_i. The the difference $A - \sigma I$ remains upper Hessenberg and the factorization yields an upper Hessenberg Q_i since the j^{th} column of Q is a linear combination of the leading j columns of $A_i - \sigma_i I$.

$$A_{i+1} = R_i Q_i + \sigma_i I \tag{41}$$

If Q_i is upper Hessenberg, the product $R_i Q_i$ is upper Hessenberg using the results of the previous question. Adding $\sigma_i I$ does not change this. Therefore, if A_i is upper Hessenberg, then so is A_{i+1}

4 QR iteration with bad shifts

a)

Explicit QR iteration without shifts (Demmel Algorithm 4.4) was implemented in Matlab in `qr_iteration.m`. Explicit QR iteration with shifts (Demmel Algorithm 4.5) was implemented in `qr_shift.m`. Matlab's built-in "qr" was used to perform the factorizations, and the shifts used $\sigma_i = A_i(4,4)$. The algorithms were said to converge when $\|\text{diag}(A_{i+1} - A_i)\|_\infty$ is less than the tolerance 10^{-6}. The matrix examined was

$$A = \begin{pmatrix} 5 & -1 & 0 & 0 \\ -1 & 5 & -1 & 0 \\ 0 & -1 & 5 & -1 \\ 0 & 0 & -1 & 5 \end{pmatrix} \tag{42}$$

b)

	`qr_iteration.m`	`qr_shift.m`	Matlab `eig`
Eigenvalues	6.6180e+00	5.0000e+00	3.3820e+00
	5.6180e+00	5.0000e+00	4.3820e+00
	4.3820e+00	5.0000e+00	5.6180e+00
	3.3820e+00	5.0000e+00	6.6180e+00
Iterations	43	1	-

Table 1

The results of using the algorithms outlined in part a) are displayed in Table 1. Comparing to those given by the command "eig" in Matlab, we see that QR iteration without shifts gives accurate answers after 43 iterations, but the QR iteration with shifts gives an incorrect answer after 1 iteration.

To understand why the QR iteration with shifts fails to give accurate results, consider convergence of QR iteration on the shifted matrix:

$$A - \sigma = \begin{pmatrix} 0 & -1 & 0 & 0 \\ -1 & 0 & -1 & 0 \\ 0 & -1 & 0 & -1 \\ 0 & 0 & -1 & 0 \end{pmatrix} \tag{43}$$

Using Matlab's `eig` to compute the eigenvalues of this matrix:

$$\begin{aligned} \lambda_{1,2} &= \pm 1.618 \\ \lambda_{3,4} &= \pm 0.618 \end{aligned} \tag{44}$$

As discussed in class, the convergence rate depends on the ratio of the eigenvalues. Since the eigenvalues of $A - \sigma I$ are symmetric about the origin, the algorithm cannot decide which way to go, and it stalls. A algorithm is needed with a different shift that breaks the symmetry.

c)

To correct the issues observed with shifting in Part (b), we may use the Wilkinson shift, described in Demmel §5.3.1. QR iteration with Wilkinson's shift is globally, and at least linearly, convergent. It is asymptotically cubically convergent for almost all matrices. In the Wilkinson shift, we define

$$B = \begin{pmatrix} a_{n-1} & b_{n-1} \\ b_{n-1} & a_n \end{pmatrix} \tag{45}$$

as the lower right 2x2 submatrix of A. We choose σ_i as the eigenvalue B that is closest to a_n. The eigenvalues of B may be expressed

$$\lambda_{1,2} = \frac{1}{2}\left(a_{n-1} + a_n \pm \sqrt{(a_{n-1} - a_n)^2 + 4b_{n-1}^2}\right) \tag{46}$$

Setting $d = (a_{n-1} - a_n)/2$,

$$\lambda_{1,2} = a_n + d \pm \sqrt{b_{n-1}^2 + d^2} \tag{47}$$

So to choose σ_i, we must consider the sign of d:
If $d \geq 0$,

$$\sigma_i = a_n + d - \sqrt{b_{n-1}^2 + d^2} \tag{48}$$

If $d < 0$,

$$\sigma_i = a_n + d + \sqrt{b_{n-1}^2 + d^2} \tag{49}$$

Implementing this in the function `qr_wilkinson.m`, the results are displayed in Table 2. We see that the algorithm converges to the correct result in 16 iterations.

	`qr_wilkinson.m`	Matlab `eig`
Eigenvalues	6.6180e+00	3.3820e+00
	5.6180e+00	4.3820e+00
	3.3820e+00	5.6180e+00
	4.3820e+00	6.6180e+00
Iterations	16	-

Table 2

A qr_iteration.m

```
%Smetana_Gregory_1917370_A5_P4
function [ Anew ] = qr_iteration( A )
%QR_ITERATION
i = 0;
tol = 10^-6;
err = 1;
Aold = A;

while i < 1E6 && err > tol
    [Q,R] = qr(Aold);
    Anew = R*Q;
    i = i+1;
    err = max(diag(Anew - Aold));
    Aold = Anew;
end
i
```

B qr_shift.m

```
%Smetana_Gregory_1917370_A5_P4
function [ Anew ] = qr_shift( A )
%QR_SHIFT
i = 0;
tol = 10^-6;
err = 1;
Aold = A;
n = size(A,1);

while i < 1E6 && err > tol
    sigma = A(n,n);
    [Q,R] = qr(Aold - sigma*eye(n));
    Anew = R*Q + sigma*eye(n);
    i = i+1;
    err = max(diag(Anew - Aold));
    Aold = Anew;
end
i
```

C qr_wilkinson.m

```
function [ Anew ] = qr_wilkinson( A )
%QR_WILKINSON
i = 0;
```

```
tol = 10^-6;
err = 1;
Aold = A;
n = size(A,1);
 while i < 1E6 && err > tol
    B = A(n-1:n, n-1:n);
    d = (B(1,1)-B(2,2))/2;
    if( d >= 0)
        sigma = B(2,2) + d - sqrt(B(1,2)^2 + d^2);
    else
        sigma = B(2,2) + d + sqrt(B(1,2)^2 + d^2);
    end

    [Q,R] = qr(Aold - sigma*eye(n));
    Anew = R*Q + sigma*eye(n);

    i = i+1;
    err = max(diag(Anew - Aold));
    Aold = Anew;
 end
 i
```

D Smetana_Gregory_1917370_A5_P4_DIARY.txt

```
run('Smetana_Gregory_1917370_A5_P4.m');
qr_iteration.m
i =
    43
ans =
   6.6180e+00
   5.6180e+00
   4.3820e+00
   3.3820e+00
qr_shift.m
i =
     1
ans =
     5
     5
     5
     5
eig
ans =
   3.3820e+00
   4.3820e+00
   5.6180e+00
   6.6180e+00
qr_wilkinson.m
```

```
i =
    16
ans =
   6.6180e+00
   5.6180e+00
   3.3820e+00
   4.3820e+00
diary off
```

E Smetana_Gregory_1917370_A5_P4m

```
%Smetana_Gregory_1917370_A5_P4
A = [5, -1, 0, 0;
    -1, 5,  -1,0;
    0, -1, 5, -1;
    0, 0, -1, 5];
format compact;
display('qr_iteration.m')
diag(qr_iteration(A))
display('qr_shift.m')
diag(qr_shift(A))
display('eig')
eig(A)
display('qr_wilkinson.m');
diag(qr_wilkinson(A))
```