Assignment 5 Solutions

Aaron Cahn
University of Wisconsin-Madison
cahn@cs.wisc.edu

April 16, 2015

1 Solutions

1.1 Question 1

1.1.1 Part A

Method 1: LU factor $A - sI$ for $s = 0$ and $s = 1$ and inspecting the number of negative values on the diagonal of U.

Case 1: $s = 0$,

$$A_0 = \begin{pmatrix} -1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 2 \end{pmatrix} \tag{1}$$

$$L_0 = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, U_0 = \begin{bmatrix} -1 & 1 & 0 \\ 0 & 2 & 1 \\ 0 & 0 & 1.5 \end{bmatrix} \tag{2}$$

$$L_1 = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0.5 & 1 \end{bmatrix}, U_1 = \begin{bmatrix} -1 & 1 & 0 \\ 0 & 2 & 1 \\ 0 & 0 & 1.5 \end{bmatrix} \tag{3}$$

Therefore, when we inspect the diagonal of U_1 we see that there is one negative value. This means that there is one eigenvalue in A that is less than zero.

Case 2: $s = 1$,

$$A_1 = \begin{pmatrix} -2 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix} \tag{4}$$

$$L_0 = \begin{bmatrix} 1 & 0 & 0 \\ -0.5 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, U_0 = \begin{bmatrix} -2 & 1 & 0 \\ 0 & 0.5 & 1 \\ 0 & 1 & 1 \end{bmatrix} \tag{5}$$

$$L_1 = \begin{bmatrix} 1 & 0 & 0 \\ -0.5 & 1 & 0 \\ 0 & 2 & 1 \end{bmatrix}, U_1 = \begin{bmatrix} -2 & 1 & 0 \\ 0 & 0.5 & 1 \\ 0 & 0 & -1 \end{bmatrix} \tag{6}$$

Therefore, when we inspect the diagonal of U_1 we see that there are two negative values. This means that there are two eigenvalues in A that are less than one. Utilizing the knowledge from both cases we have found that there is one eigenvalue in the range $[0, 1]$.

Method 2: Compute the number of sign changes in the sequence of main principal minors of $A - sI$ for $s = 0$ and $s = 1$.

Listing 1: Matlab Commands

```
function [d] = mpm(A)
    n = size(A);
    d = []; d(1) = 1;
    d(2) = A(1,1);
    for i=2:n
        d(i+1)=A(i,i)*d(i)-A(i,i-1)^2*d(i-1);
    end
end
```

Listing 2: Matlab Commands

```
% case 1: A-0*I
mpm(A)
ans =
    [1]    [-1]    [-2]    [-3]

% case 2: A-1*I
mpm(A-eye(3))
ans =
    [1]    [-2]    [-1]    [1]
```

We see that in case 1 there is one sign change in the sequence of main principal minors and that in case 2 there are two sign changes in the sequence of main principal minors. Therefore, as with method 1 we have shown there is a single eigenvalue in the range $[0, 1]$.

1.1.2 Part B

The best estimate count of arithmetic operations for the two methods was 116 flops for method 1 and 36 flops for method 2. Therefore, it would seem that method 2 is more efficient for tridiagonal matrices and it makes intuitive sense that this gap will become larger as the matrices grow in size (*i.e.,* method 2 will look better and better relative to method 1).

1.1.3 Part C

The code below generates 100 symmetric tridiagonal matrices of size 50x50 and runs both Method 1 and Method 2 from part A.

Listing 3: Matlab Commands

```
lu_times = []; mpm_times = []; m = 50;
for i=1:100
    % create the tridiagonal matrix
    B=rand(m); A=B*B';
    A=diag(diag(A,-1),-1)+diag(diag(A))+diag(diag(A,1),1);

    % Method 1: LU factorization
    tic;
    % case 1: A-0*I
    [L,U] = lu_sym(A);
    pevals1 = length(find(diag(U)>0));
    % case 2: A-1*I
    [L,U] = lu_sym(A-eye(m));
    pevals2 = length(find(diag(U)>0));
    % eigen values in the range
    pevals = pevals1-pevals2;
    lu_times(i) = toc;

    % Method 2: MPM sequences
    tic;
    % case 1: A-0*I
    [d] = mpm(A);
    % count the sign changes
    pevals1 = sign_changes(d);
    % case 2: A-1*I
    [d] = mpm(A-eye(m));
    % count the sign changes
    pevals2 = sign_changes(d);
    % eigen values in the range
    pevals = pevals2-pevals1;
    mpm_times(i) = toc;

end

avg_lu=sum(lu_times)/m
avg_mpm=sum(mpm_times)/m
```

Listing 4: Matlab Commands

```
function [pevals] = sign_changes(d)
    signs = sign(d);
    s = signs(1);
    pevals = 0;
    for i=2:length(signs)
        if signs(i) ~= s
            pevals = pevals + 1;
        end
        s = signs(i);
    end
end
```

The results of the code snippet above are presented below. It is clear as stated in part B that Method 2 dominates Method 1 in terms of running speed. This contrast grows as the size of the symmetric tridiagonal matrix grows.

Listing 5: Matlab Commands

```
q1_partC

avg_lu =

    0.1649


avg_mpm =

   7.5590e-04
```

1.2 Question 2

1.2.1 Part A

Since the Gerschgorin's disks are pairwise disjoint we know that each disk must contain one eigenvalue. Therefore the cardinality of $\sigma(A) = n$. This means that the algebraic multiplicity m_a of each eigenvalue is $m_a = 1$. We know that the geometric multiplicity m_g follows the rule $m_g \leq m_a$ for all eigenvalues of A. Therefore, since $m_a = 1$ we can say that $m_g = 1$.

1.2.2 Part B

Leon is correct because as stated above $m_g = m_a$ for eigenvalues of A.

1.2.3 Part C

Nina is correct because there are n eigenvalues; one per disk. If there was a complex eigenvalue it's complex conjugate would be an eigenvalue as well. This would require two eigenvalues to be located in a single disk. However this would contradict what we have shown in (a). Therefore all eigenvalues of A are real.

1.2.4 Part D

Elvis is not correct. The problem is that there can be two distinct eigenvalues in the real that have the same magnitude. For example, suppose a matrix B as eigenvalues $\lambda_1 = 6, \lambda_2 = -6$. It is clear, $|\lambda_1| = |\lambda_2|$. In this case the power method will not converge because there is no dominant eigenvalue.

1.3 Question 3

Listing 6: Matlab Commands

```
m = length(A);
[P,D] = eig(A);
D= diag(D);

% Step 1: Power Method
% (find the first two dominant eigenvalues)

% most dominant eigenvalue
[eval1,evec1,err1] = power_method(A,rand(m,1),10^-2);
fprintf('1st dom from pm: %f\n',eval1)
fprintf('true 1st dom: %f\n\n',D(1));

% second most dominant eigenvalue (deflate eval1 from A)
e = [1 zeros(1,m-1)]'; w = e-evec1; w = w/norm(w);
H = eye(m)-2*w*w';
A1 = H*A*H; tilA = A1(2:end,2:end);
[eval2,evec2,err2] = power_method(tilA,rand(m-1,1),10^-2);
fprintf('2nd dom from pm: %f\n',eval2)
fprintf('true 2nd dom: %f\n\n',D(2));

% Step 2: Inverse Power Method
% (find the first two least dominant evals)

% least dominant eigenvalue
[eval1,evec1,err1] = inv_power_method(A,rand(m,1),10^-2);
fprintf('1st least dom from pm: %f\n',eval1)
fprintf('true 1st least dom: %f\n\n',D(m));

% second least dominant eigenvalue (deflate eval1 from A)
e = [1 zeros(1,m-1)]'; w = e-evec1; w = w/norm(w);
H = eye(m)-2*w*w';
A1 = H*A*H; tilA = A1(2:end,2:end);
[eval2,evec2,err2] = inv_power_method(tilA,rand(m-1,1),10^-2);
fprintf('2nd least dom from pm: %f\n',eval2)
fprintf('true 2nd least dom: %f\n\n',D(m-1));
```

Listing 7: Matlab Commands

```
1st dom from pm: 6.112115
true 1st dom: 6.116471

2nd dom from pm: -1.288738
true 2nd dom: -1.286597

1st least dom from pm: -0.115228
true 1st least dom: -0.115301

2nd least dom from pm: -0.342898
true 2nd least dom: -0.344068
```

There are two main issues or problems that may arise during this procedure. The first is that the method simply will not converge (or at least converge in a reasonable amount of time) if there are eigenvalues that are extremely close to each other in magnitude. The second is that this method finds approximations to eigenvalues and eigenvectors. However, after deflating the matrix and subsequent re-inflation of the a found eigenvector is routine done using an approximate result. Therefore, even if the error is very small this error is compounded during the procedure for finding subsequent eigenvalues and eigenvectors. This can already be seen when finding the eigenvector of the 2nd most (least) dominant eigenvalues.

A possible way to remedy these issues is to use Gerschgorin's theorem to find first an approximation of an eigenvalue. Then we can use the shifted inverse power method. This should hopefully push other eigenvalues (perhaps complex pairs) far away so that convergence will occur quickly. There will be less compounding of error the faster the method converges. Therefore, the approximations of the eigen-pairs will be more accurate.

1.4 Question 4

1.4.1 Part A

From the question we know that the matrix A_{mxm} is symmetric tridiagonal. Assume A as follows:

$$A = \begin{bmatrix} & | & & | & \\ & | & & | & \\ a_1 & | & \dots & | & a_m \\ & | & & | & \\ & | & & | & \end{bmatrix} \tag{7}$$

Now we will show the construction of Q and R in general which will demonstrate the general structure Q and R will end in. Note, we show the kth iteration, where $k = 1 : m$. First, we extract $x_k = a_k$, where $x_k = [0\dots0x_{k-1}x_kx_{k+1}0\dots0]'$. Then we create y_k based on x_k, producing $y_k = [0\dots0x_{k-1}\alpha0\dots0]'$. Note, α is not relevant. Next we create $w_k = (x_k - y_k)/(||x_k - y_k||)$, producing $w_k = [0\dots0\beta x_{k+1}0\dots0]'$. Again, β is not relevant. Finally, we create $H_k = I - 2 * w * w'$ which will have the general structure describe below.

$$H_k = \begin{bmatrix} 1 & 0 & \dots & & & & & 0 \\ 0 & \ddots & 0 & \dots & & & & 0 \\ 0 & 0 & 1 & 0 & \dots & & & 0 \\ 0 & \dots & 0 & h_kk & h_kk+1 & 0 & \dots & 0 \\ 0 & \dots & 0 & h_k+1k & h_k+1k+1 & 0 & \dots & 0 \\ 0 & \dots & & & 0 & 1 & 0 & 0 \\ 0 & \dots & & & & 0 & \ddots & 0 \\ 0 & \dots & & & & & 0 & 1 \end{bmatrix} \tag{8}$$

From H_k it is clear to see small sub-matrix (call it h') moves down the diagonal during each iteration and each h' adds an element right below the kth diagonal. Therefore, when QR factorization is complete Q will have Hessenberg structure. This means Q will have zeros everywhere below the second sub-diagonal. Since A began as tridiagonal, after applying the final Q to A, $A = R$ will shift to having elements on the diagonal and the 1$^{\text{st}}$ and 2$^{\text{nd}}$ super-diagonals. The rest will be zero.

1.4.2 Part B

Here we know that A is symmetric, $A = QR$, $B = RQ$, and since Q was Hessenberg this means B will be as well. If we can show B is symmetric then we know B is upper and lower Hessenberg and therefore tridiagonal. The derivation is as follows:

$$\begin{aligned} B &= RQ && (9) \\ &= IRQ && (10) \\ &= Q'QRQ && (11) \\ &= Q'AQ && (12) \\ & && (13) \end{aligned}$$

Then to show B is symmetric,

$$\begin{aligned} B' &= (Q'AQ)' && (14) \\ &= Q'A'Q'' && (15) \\ &= Q'AQ && (16) \\ &= B && (17) \\ & && (18) \end{aligned}$$

This concludes the proof.