Q2 (1) $(A-\lambda_2 I)(A-\lambda_3 I)\ldots(A-\lambda_n I)\left(\sum_{i=1}^{n}\alpha_i v_i\right)=0$

show

$\alpha_1(\lambda_1-\lambda_2)(\lambda_1-\lambda_3)\ldots(\lambda_1-\lambda_n)v_1=0$

case $n=2$:

$(A-\lambda_2 I)(\alpha_1 v_1+\alpha_2 v_2)=0$

$\alpha_1(A-\lambda_2 I)v_1+\alpha_2(A-\lambda_2 I)v_2=0$

Since v_1 & v_2 are eigenvectors $\rightarrow$ $(A-\lambda_1 I)v_1=0$

$(A-\lambda_2 I)v_2=0$

So $\alpha_1(\lambda_1-\lambda_2)v_1=0$

for general case n:

expanding (1) : $\sum_{i=1}^{n}\alpha_i(A-\lambda_2 I)(A-\lambda_3 I)\ldots(A-\lambda_n I)v_i=0$

Since v_i are eigenvectors, $(A-\lambda_i I)v_i=0$.
Thus each term in the sum becomes zero for $i\neq 1$
for $i=1$ the eqn. becomes:

$\alpha_1(\lambda_1-\lambda_2)(\lambda_1-\lambda_3)\ldots(\lambda_1-\lambda_n)v_1=0$

Q5. A & $\tilde{A}$ are similar (there exists some invertible matrix M s.t. $\tilde{A} = M^{-1}AM$)

a) The eigenvalues of $\tilde{A}$ are the same as the eigenvalues of A.

Let λ be an eigenvalue of A w/ corresponding eigenvector x.

$$Ax = \lambda x$$

$$M^{-1}AMM^{-1}x = M^{-1}\lambda MX$$

$$\tilde{A}(M^{-1}x) = \lambda(M^{-1}x)$$

This shows λ is also an eigenvalue of $\tilde{A}$ w/ corr. eigenvector $M^{-1}x$. Hence the eigenvalues of A & $\tilde{A}$ are the same.

b) The eigenvectors of $\tilde{A}$ are not necessarily the same as the eigenvectors of A.

From part a we know that eigenvalues are the same. This shows that eigenvectors of A & $\tilde{A}$ are related but not necessarily the same.

Q6. Let $p(t)$ be a polynomial. & $A: V \to V$ be a lin. map. Suppose λ is an eigenvalue of A w/ eigenvector x, that is, $Ax = \lambda x$. Show that $p(\lambda)$ is an eigenvalue of the linear map $p(A)$ w/ eigenvalue $p(\lambda)$.

Let $p(t) = a_0 + a_1 t + \ldots + a_n t^n$

we know that $Ax = \lambda x$

Consider the linear map $p(A)$ acting on x:

$$p(A)x = (a_0 I + a_1 A + a_2 A^2 + \ldots + a_n A^n)x$$

$$= a_0 x + a_1 \lambda x + a_2 \lambda^2 x + \ldots + a_n \lambda^n x$$

$$= (a_0 + a_1 \lambda + a_2 \lambda + \ldots + a_n \lambda^n)x$$

$$= p(\lambda)x$$

$\rightarrow$ since $p(A)x = p(\lambda)x$ $p(\lambda)$ is an eigenvalue of $p(A)$ w/ the same eigenvector x.

Q8. Let $A = \begin{bmatrix} \lambda & 1 & 0 \\ 0 & \lambda & 1 \\ 0 & 0 & \lambda \end{bmatrix}$. Use the series definition of the exponential to show that $e^{At} = \begin{bmatrix} e^{\lambda t} & te^{\lambda t} & \frac{t^2}{2} e^{\lambda t} \\ 0 & e^{\lambda t} & te^{\lambda t} \\ 0 & 0 & e^{\lambda t} \end{bmatrix}$

(1) the series def: $e^{At} = I + At + \frac{(At)^2}{2!} + \dots$

$$I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$At = \begin{bmatrix} \lambda t & t & 0 \\ 0 & \lambda t & t \\ 0 & 0 & \lambda t \end{bmatrix}$$

$$\frac{(At)^2}{2!} = \frac{1}{2!} \begin{bmatrix} \lambda^2 t^2 & t^2 & 0 \\ 0 & \lambda^2 t^2 & t^2 \\ 0 & 0 & \lambda^2 t^2 \end{bmatrix}$$

$$\frac{(At)^3}{3!} = \frac{1}{3!} \begin{bmatrix} \lambda^3 t^3 & t^3 & 0 \\ 0 & \lambda^3 t^3 & t^3 \\ 0 & 0 & \lambda^3 t^3 \end{bmatrix}$$

$\vdots$

$$\text{ÿ①} \to e^{At} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} \lambda t & t & 0 \\ 0 & \lambda t & t \\ 0 & 0 & \lambda t \end{bmatrix} + \frac{1}{2!} \begin{bmatrix} \lambda^2 t^2 & t^2 & 0 \\ 0 & \lambda^2 t^2 & t^2 \\ 0 & 0 & \lambda^2 t^2 \end{bmatrix} + \cdots$$

$$e^{At}(3,3) = e^{At}(2,2) = e^{At}(1,1) = 1 + \lambda t + \frac{\lambda^2 t^2}{2!} + \frac{\lambda^3 t^3}{3!} + \cdots \overset{(1)}{=} e^{\lambda t}$$

$$e^{At}(3,2) = e^{At}(1,2) = 0 + t + \frac{t^2}{2!} + \frac{t^3}{3!} + \cdots = t.e^{\lambda t}$$

$$e^{At} = \begin{bmatrix} e^{\lambda t} & t.e^{\lambda t} & 0 \\ 0 & e^{\lambda t} & t.e^{\lambda t} \\ 0 & 0 & e^{\lambda t} \end{bmatrix}$$

Q9. $\cos(A)$, $A=\begin{bmatrix}2 & 1\\ -1 & 2\end{bmatrix}$

$\cos(A) = P\cos(\Lambda)P^{-1}$

P: the matrix whose col.s are the eigenvectors of A

Λ: diagonal matrix of eigenvalues

$$\det(A-\lambda I) = \det\begin{pmatrix}2-\lambda & 1\\ -1 & 2-\lambda\end{pmatrix} = (2-\lambda)^2+1=0$$

$\lambda = 2 \mp i$

eigenvector

for $\lambda = 2+i \longrightarrow v=\begin{pmatrix}1\\ -i\end{pmatrix}$

$\lambda = 2-i \longrightarrow v=\begin{pmatrix}1\\ i\end{pmatrix}$

$P=\begin{pmatrix}1 & 1\\ -i & i\end{pmatrix}$, $\quad P^{-1}=\frac{1}{2i}\begin{bmatrix}i & -1\\ i & 1\end{bmatrix}$

$\Lambda=\begin{pmatrix}2+i & 0\\ 0 & 2-i\end{pmatrix}$

$$\cos\Lambda = \begin{pmatrix}\cos(2)\cos(i)-\sin(2)\sin(i) & 0\\ 0 & \cos(2)\cos(-i)-\sin(2)\sin(-i)\end{pmatrix}$$

So

$$\cos(A) = P\cos(\Lambda)P^{-1}$$

$$= \begin{bmatrix} 1 & 1 \\ -i & i \end{bmatrix} \begin{bmatrix} \cos 2\cos i - \sin 2\sin i & 0 \\ 0 & \cos 2\cos(-i) - \sin 2\sin(-i) \end{bmatrix} \begin{bmatrix} 1/2 & -1/2 i \\ 1/2 & 1/2 i \end{bmatrix}$$

$$= \begin{bmatrix} 0.36 + 0.0000i & -1.07 + 0.0000i \\ 1.07 + 0.0000i & -1.64 + 0.000i \end{bmatrix}$$ (used Matlab)

Q10

$$A = \begin{bmatrix} 2 & 1 & 0 & 0 & 0 \\ 0 & 2 & 1 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 2 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

[Jordan]

a) A is in blockdiagonal form.

The 1st block : 3×3 Jordan block, eigenvalue 2
The 2nd block : 1×1 " , " 2
The 3rd block : 1×1 " , " 1

b) Algebraic multiplicity of an eigenvalue is the # of times it appears on the diagonal of its Jordan form.

eigenvalue 2 appears 4 times & 1 appears once

So

algebr. multip. of eigenvalue 2 is 4
" " 1 is 1

c) Geometric multiplicity → # of Jordan blocks corr. to that eigenvalue.

eigenvalue 2 → 2 Jordan blocks (3×3 & 1×1)
eigenvalue 1 → 1 " " (1×1)

So

geo. multip. of eigenvalue 2 is 2
" " 1 is 1

d) The index → size of the largest Jordan block associated w/ that eigenvalue

so eigenvalue 2 has a largest Jordan block of size 3x3
" 1 has a " " of size 1x1

e) $\phi(s) = (s-\lambda_1)^{m_1} \dots (s-\lambda_k)^{m_k}$

for matrix

$A \rightarrow \phi(s) = (s-2)^4 (s-1)$

→ alg. mult. of 4

→ alg. mult of 1.

→ eigenvalues

f) $\psi(A)=0$ → In terms of Jordan form the min. poly. can be determined from the size of the largest Jordan block for each eigenvalue.

$\psi(s) = (s-2)^3 (s-1)$

largest block sizes

eigenvalues

g) The block for eigenvalue 2 : , eigenvalue 1 :

$$e^{2t}\begin{bmatrix} 1 & t & t^2/2 \\ 0 & 1 & t \\ 0 & 0 & 1 \end{bmatrix} \qquad e^t$$

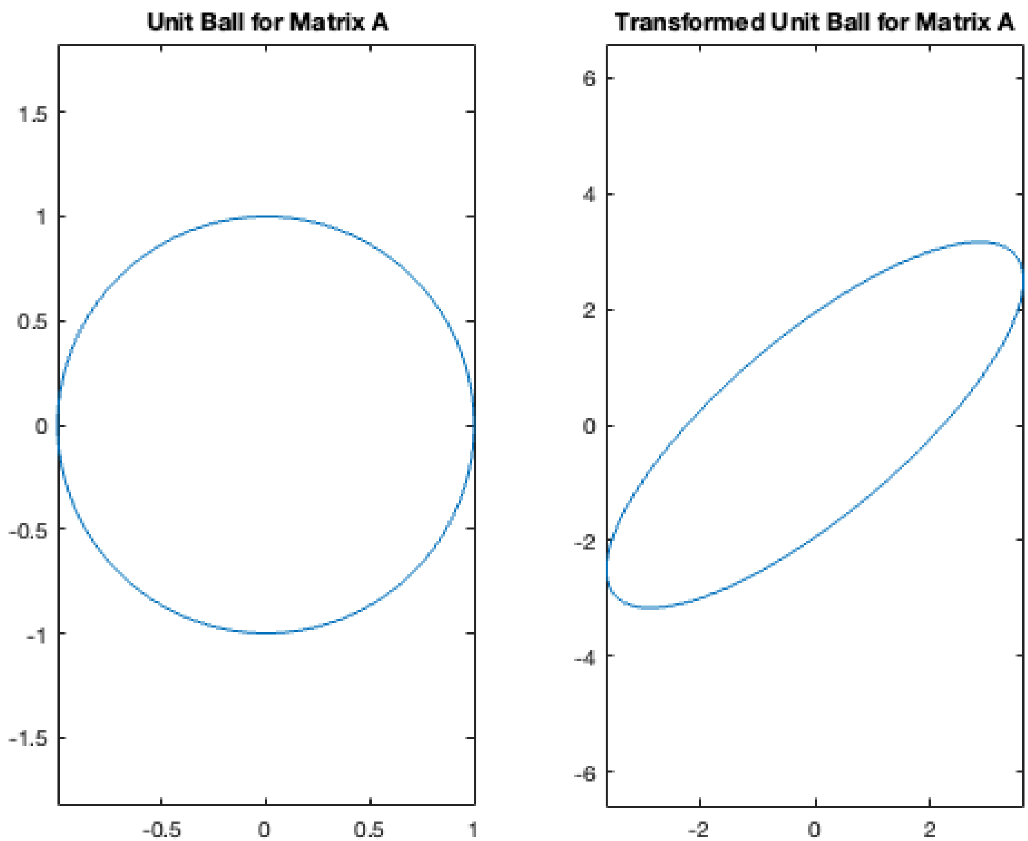
$$e^{Jt} = \begin{bmatrix} e^{2t} & te^{2t} & \frac{t^2}{2}e^{2t} & 0 & 0 \\ 0 & e^{2t} & te^{2t} & 0 & 0 \\ 0 & 0 & e^{2t} & 0 & 0 \\ 0 & 0 & 0 & e^t & 0 \\ 0 & 0 & 0 & 0 & e^t \end{bmatrix}$$

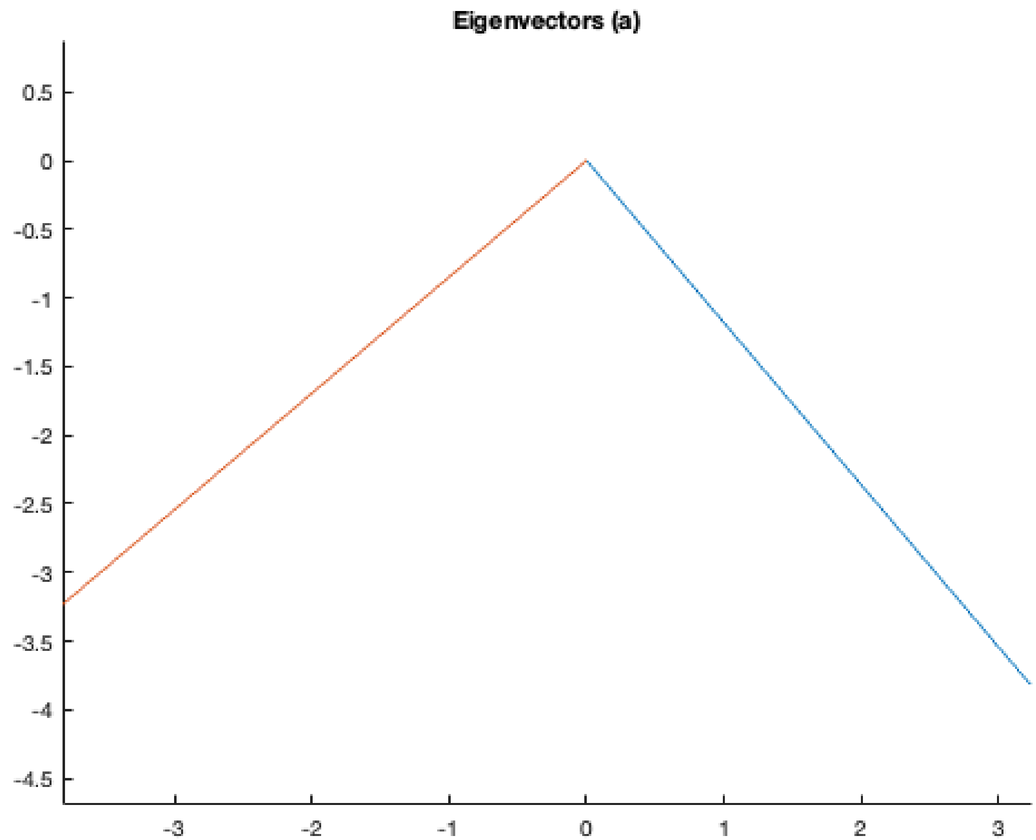
```
% ME564_HW5
% Q1

% a
% Part i: Find Eigenvalues and Eigenvectors
A = [2, 3; 3, 1];
[V, D] = eig(A);

% Part ii: Plot the Unit Ball and Its Transformation
theta = linspace(0, 2*pi, 100);
x = cos(theta);
y = sin(theta);
unitBall = [x; y];

transformedBall = A * unitBall;

figure;
subplot(1,2,1);
plot(x, y);
title('Unit Ball for Matrix A');
axis equal;

subplot(1,2,2);
plot(transformedBall(1,:), transformedBall(2,:));
title('Transformed Unit Ball for Matrix A');
axis equal;

% Part iii: Plot the Eigenvectors
maxLength = 5;
length = linspace(0, maxLength, 100);

figure;
hold on;
for i = 1:size(V, 2)
    v = V(:, i);
    plot(length * v(1), length * v(2));
end
title('Eigenvectors (a)');
axis equal;
hold off;

% Part iv: Find the Value of maxLength
eigenvalues = diag(D);
maxLengthValues = 1 ./ eigenvalues;

disp('Values of maxLength for each eigenvector (a):');
disp(maxLengthValues);

% b

% answer to d: The reason eigenvectors were not plotted for part (b) is
% likely because the matrix A in that part is a rotation matrix & for a
```

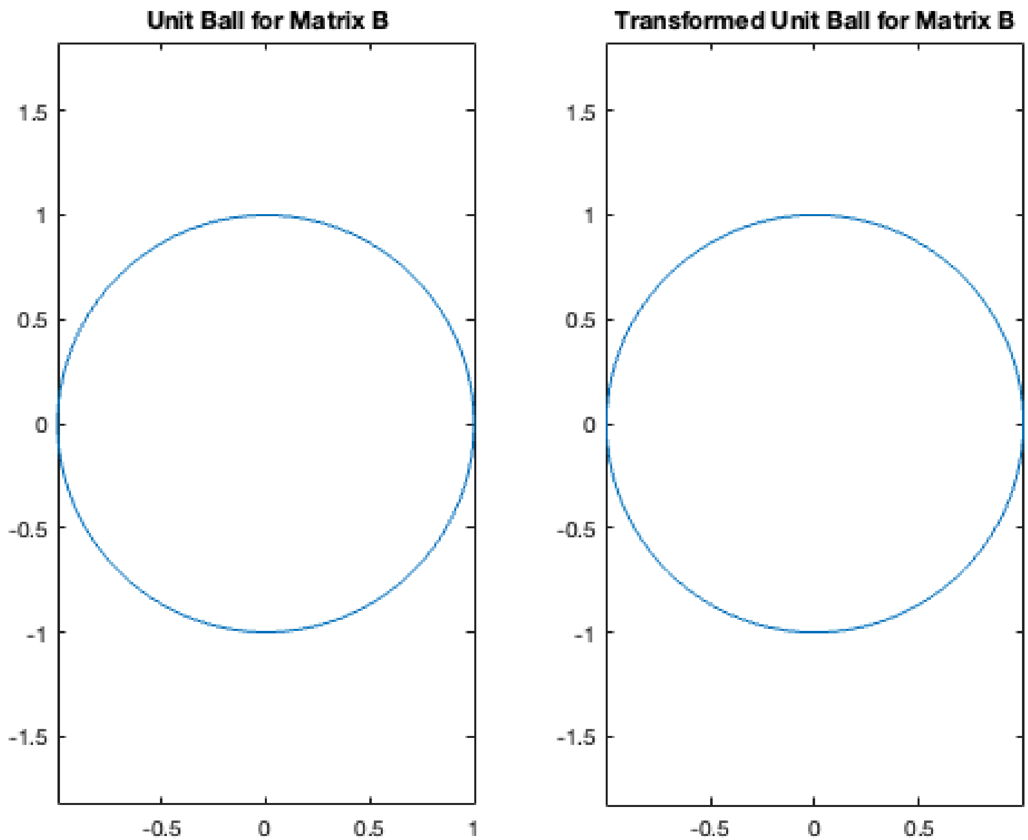
```
% 2D rotation matrix, the eigenvalues & eigenvectors are complex.
% So the reason might be: Plotting complex eigenvectors in the same
% 2D space as the unit ball and its transformation would not be meaningful,
% as the eigenvectors would not lie in the same real plane.

% Part i: Find Eigenvalues and Eigenvectors for Matrix B
A = [cos(pi/5), -sin(pi/5); sin(pi/5), cos(pi/5)];
[V, D] = eig(A);

% Display Eigenvalues and Eigenvectors
disp('Eigenvalues (b):');
disp(diag(D));
disp('Eigenvectors (b):');
disp(V);

% Part ii: Plot the Unit Ball and Its Transformation for Matrix B
theta = linspace(0, 2*pi, 100);
x = cos(theta);
y = sin(theta);
unitBall = [x; y];

transformedBall = A * unitBall;

figure;
subplot(1,2,1);
plot(x, y);
title('Unit Ball for Matrix B');
axis equal;

subplot(1,2,2);
plot(transformedBall(1,:), transformedBall(2,:));
title('Transformed Unit Ball for Matrix B');
axis equal;

% c
% Part i: Find Eigenvalues and Eigenvectors for Matrix C
A = [7/8, -1/4; -1/8, 1];
[V, D] = eig(A);

% Display Eigenvalues and Eigenvectors
disp('Eigenvalues (c):');
disp(diag(D));
disp('Eigenvectors (c):');
disp(V);

% Part ii: Plot the Unit Ball and Its Transformation for Matrix C
theta = linspace(0, 2*pi, 100);
x = cos(theta);
y = sin(theta);
unitBall = [x; y];

transformedBall = A * unitBall;

figure;
```

```
subplot(1,2,1);
plot(x, y);
title('Unit Ball for Matrix C');
axis equal;

subplot(1,2,2);
plot(transformedBall(1,:), transformedBall(2,:));
title('Transformed Unit Ball for Matrix C');
axis equal;

% Part iii: Plot the Eigenvectors for Matrix C
maxLength = 5;
length = linspace(0, maxLength, 100);

figure;
hold on;
for i = 1:size(V, 2)
    v = V(:, i);
    plot(length * v(1), length * v(2));
end
title('Eigenvectors for Matrix C');
axis equal;
hold off;

% Part iv: Find the Value of maxLength for Matrix C
eigenvalues = diag(D);
maxLengthValues = 1 ./ eigenvalues;

disp('Values of maxLength for each eigenvector for Matrix C:');
disp(maxLengthValues);
```

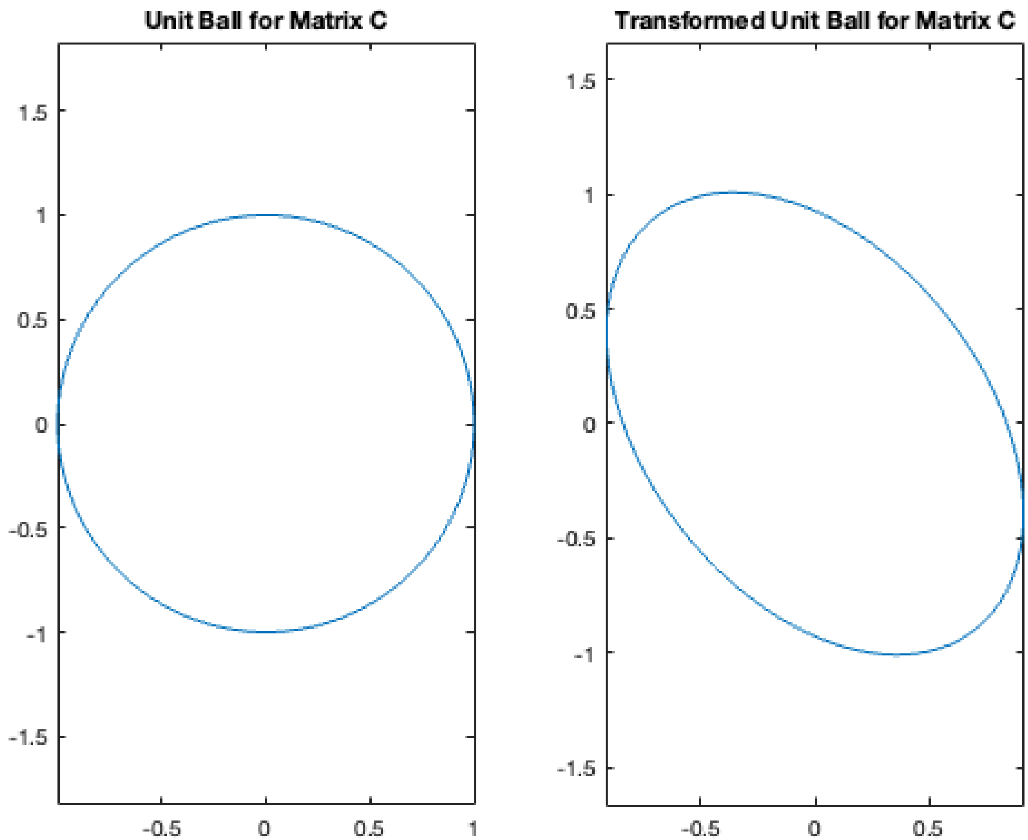
```
Values of maxLength for each eigenvector (a):
   -0.6488
    0.2202

Eigenvalues (b):
   0.8090 + 0.5878i
   0.8090 - 0.5878i

Eigenvectors (b):
   0.7071 + 0.0000i   0.7071 + 0.0000i
   0.0000 - 0.7071i   0.0000 + 0.7071i

Eigenvalues (c):
    0.7500
    1.1250

Eigenvectors (c):
   -0.8944    0.7071
   -0.4472   -0.7071

Values of maxLength for each eigenvector for Matrix C:
    1.3333
    0.8889
```


Unit Ball for Matrix A
1.5
1
0.5
0
-0.5
-1
-1.5
-0.5
0
0.5
1
Transformed Unit Ball for Matrix A
6
4
2
0
-2
-4
-6
-2
0
2

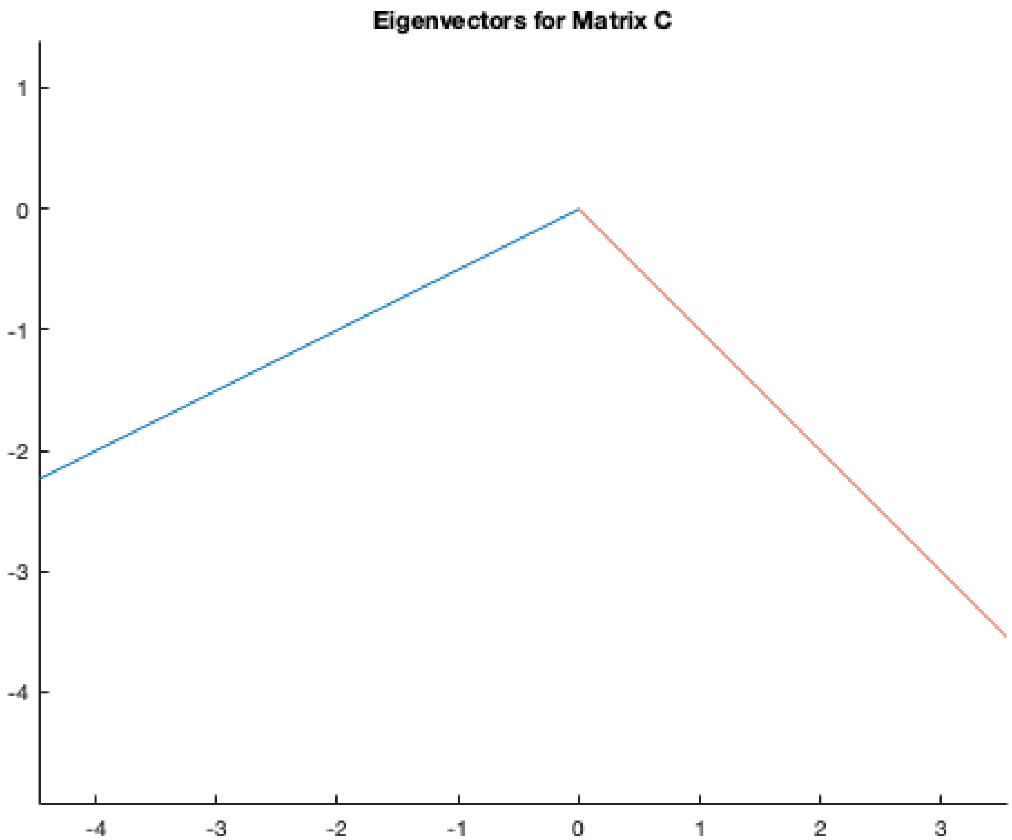

Eigenvectors (a)
0.5
0
-0.5
-1
-1.5
-2
-2.5
-3
-3.5
-4
-4.5
-3
-2
-1
0
1
2
3



Unit Ball for Matrix B
Transformed Unit Ball for Matrix B
1.5
1
0.5
0
-0.5
-1
-1.5
-0.5
0
0.5
1



Unit Ball for Matrix C
1.5
1
0.5
0
-0.5
-1
-1.5
-0.5
0
0.5
1
Transformed Unit Ball for Matrix C
1.5
1
0.5
0
-0.5
-1
-1.5
-0.5
0
0.5



Published with MATLAB® R2022b

```
% ME564_HW5
% Q3

% Part a
disp('Part a:');

% i. Find the eigenvalues
B = [8, -8, -2; 4, -3, -2; 3, -4, 1];
eigenvalues = eig(B);
disp('Eigenvalues:');
disp(eigenvalues);

% ii. Find eigenvectors and/or generalized eigenvectors
[V, J] = jordan(B);
disp('Eigenvectors/Generalized Eigenvectors (columns of P):');
disp(V);

% iii. Compute the Jordan form
J_computed = inv(V) * B * V;
disp('Jordan Form:');
disp(J_computed);

% Double-check with MATLAB's jordan function
[J_check, P_check] = jordan(B);
disp('Jordan Form (MATLAB check):');
disp(J_check);

% Part b
disp('Part b:');

% i. Find the eigenvalues
B = [2, 1, 1; 0, 3, 1; 0, -1, 1];
eigenvalues = eig(B);
disp('Eigenvalues:');
disp(eigenvalues);

% ii. Find eigenvectors and/or generalized eigenvectors
[V, J] = jordan(B);
disp('Eigenvectors/Generalized Eigenvectors (columns of P):');
disp(V);

% iii. Compute the Jordan form
J_computed = inv(V) * B * V;
disp('Jordan Form:');
disp(J_computed);

% Double-check with MATLAB's jordan function
[J_check, P_check] = jordan(B);
disp('Jordan Form (MATLAB check):');
disp(J_check);
```

Part a:

```
Eigenvalues:
    1.0000
    3.0000
    2.0000

Eigenvectors/Generalized Eigenvectors (columns of P):
    2.0000    2.0000    3.0000
    1.0000    1.5000    2.0000
    1.0000    1.0000    1.0000

Jordan Form:
     3     0     0
     0     1     0
     0     0     2

Jordan Form (MATLAB check):
    2.0000    2.0000    3.0000
    1.0000    1.5000    2.0000
    1.0000    1.0000    1.0000

Part b:
Eigenvalues:
    2.0000
    2.0000
    2.0000

Eigenvectors/Generalized Eigenvectors (columns of P):
     1     0     0
     1     1    -1
    -1     0     1

Jordan Form:
     2     1     0
     0     2     0
     0     0     2

Jordan Form (MATLAB check):
     1     0     0
     1     1    -1
    -1     0     1
```

Published with MATLAB® R2022b

```matlab
% ME564_HW5
% Q4

% Define System Matrices
% The system is defined by the equation x(t+1) = Ax(t) + Bu(t) and y(t) =
 Cx(t)
% A is the state transition matrix, B is the input matrix, and C is the output
 matrix
A = [3, 0, -2; 0, 2, 5; 4, 3, -1];
B = [2, 0; 0, 0; 0, 1];
C = [1, 0, 1];

% Proposed Solution for Controllability
% The controllability of the system is checked using the controllability
 matrix.
% The controllability matrix is formed by [B, AB, A^2B, ..., A^(n-1)B]
% If the controllability matrix is of full rank, then the system is
 controllable.

% Calculate the controllability matrix
n = size(A, 1);  % Number of states
ControllabilityMatrix = [];
for i = 0:n-1
    ControllabilityMatrix = [ControllabilityMatrix, A^i * B];
end

% Check if the system is controllable
rank_C = rank(ControllabilityMatrix);
if rank_C == n
    disp('The system is controllable.');
else
    disp('The system is not controllable.');
end

% Proposed Solution for Observability
% The observability of the system is checked using the observability matrix.
% The observability matrix is formed by [C; CA; CA^2; ...; CA^(n-1)]
% If the observability matrix is of full rank, then the system is observable.

% Calculate the observability matrix
ObservabilityMatrix = [];
for i = 0:n-1
    ObservabilityMatrix = [ObservabilityMatrix; C * A^i];
end

% Check if the system is observable
rank_O = rank(ObservabilityMatrix);
if rank_O == n
    disp('The system is observable.');
else
    disp('The system is not observable.');
end
```

```
The system is controllable.
The system is observable.
```

Published with MATLAB® R2022b

```
% ME564_HW5
% Q7

% a
% Define the matrix A
A = [2, 1; 1, -8];

% Initialize the approximation for e^A as the identity matrix
approx_eA = eye(size(A));

% Compute the approximation using the first five terms of the series
 definition
terms = 5;
for k = 1:terms
    approx_eA = approx_eA + (A^k) / factorial(k);
end

disp('Approximation for e^A using the first five terms:');
disp(approx_eA);

% b
% Initialize the approximation for e^-A as the identity matrix
approx_e_neg_A = eye(size(A));

% Compute the approximation using the first five terms of the series
 definition
for k = 1:terms
    approx_e_neg_A = approx_e_neg_A + ((-1)^k) * (A^k) / factorial(k);
end

% Compute the approximation for e^A as the inverse of approx_e_neg_A
approx_eA = inv(approx_e_neg_A);

disp('Approximation for e^A using the first five terms and inverse method:');
disp(approx_eA);

% c
% Compute the eigenvalues and eigenvectors
[V, D] = eig(A);

disp('Eigenvalues of A:');
disp(diag(D));
disp('Eigenvectors of A (columns):');
disp(V);

% d
% Compute e^D
eD = exp(diag(D));

% Compute e^A using the formula e^A = P e^D P^{-1}
eA_exact = V * diag(eD) / V;
```

```
% Compute e^A using MATLAB's expm function
eA_matlab = expm(A);

% Display the results
disp('Exact value of e^A:');
disp(eA_exact);
disp('Value of e^A using MATLAB''s expm function:');
disp(eA_matlab);

% e
% Construct matrix P using the eigenvectors
P = V;

% Construct diagonal matrix # using the eigenvalues
Lambda = D;

% Compute e^#
eLambda = exp(diag(Lambda));

% Compute e^A using the formula e^A = P * (e^#) * P^(-1)
eA_diagonalization = P * diag(eLambda) / P;

% Display the results
disp('Value of e^A using diagonalization:');
disp(eA_diagonalization);
disp('Value of e^A using MATLAB''s expm function for comparison:');
disp(expm(A));
```

```
Approximation for e^A using the first five terms:
    6.2250   17.8417
   17.8417 -172.1917

Approximation for e^A using the first five terms and inverse method:
   31.0825    3.0776
    3.0776    0.3064

Eigenvalues of A:
   -8.0990
    2.0990

Eigenvectors of A (columns):
   -0.0985   -0.9951
    0.9951   -0.0985

Exact value of e^A:
    8.0790    0.7999
    0.7999    0.0795

Value of e^A using MATLAB's expm function:
    8.0790    0.7999
    0.7999    0.0795

Value of e^A using diagonalization:
    8.0790    0.7999
```

```
    0.7999    0.0795

Value of e^A using MATLAB's expm function for comparison:
    8.0790    0.7999
    0.7999    0.0795
```

Published with MATLAB® R2022b