MAS3114 MATLAB Assignment 4

Name: Cole Rottenberg

UF ID: 11062528

WARNING: You will recieve a zero on this assignment if you fail to enter your name, UF ID, and the seed in the random number generator.

```
clear
% Initialize the random number generator by typing the command
rng(11062528,'twister')
% where ######## must be your 8-digit UF ID number
```

Exercise 1 -- Eigenvalues & Eigenvectors

1A

```
B = randi([-8,8],4,4)
```

```
B = 4×4
    -5     3    -5    -4
     3     2    -8    -7
     6    -1     7    -5
     1     3     6    -4
```

```
% The command tril returns the lower triangular portion of matrix A.
A1 = tril(B)
```

```
A1 = 4×4
    -5     0     0     0
     3     2     0     0
     6    -1     7     0
     1     3     6    -4
```

The eigenvalues of A1 are ... beacuse (state the theorem here)

```
% Find eigenvalues and eigenvectors using eig:
% P is (give a description here)
% D is (give a description here)
[P,D] = eig(A1)
```

```
P = 4×4
         0         0         0    0.2700
         0         0    0.8085   -0.1157
         0    0.8779    0.1617   -0.1446
    1.0000    0.4789    0.5659    0.9449
D = 4×4
    -4     0     0     0
     0     7     0     0
     0     0     2     0
     0     0     0    -5
```

```
% check if A1 = P*D*inv(P) YES!
rev_A = P*D*inv(P)
```

```
rev_A = 4×4
   -5.0000         0         0         0
    3.0000    2.0000         0         0
    6.0000   -1.0000    7.0000         0
    1.0000    3.0000    6.0000   -4.0000
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Check whether P*D*inv(P) = A1.  If not, A1 is not diagonalizable. Keep your
% previous work, re-generate another matrix B and repeat the process until
% A1 is diagonalizable.
B = randi([-8,8],4,4)
```

```
B = 4×4
     0     7     4     5
     7    -2     7     3
    -1    -1    -2    -8
     2     6    -4    -3
```

```
A1 = tril(B)
```

```
A1 = 4×4
     0     0     0     0
     7    -2     0     0
    -1    -1    -2     0
     2     6    -4    -3
```

```
[P,D] = eig(A1)
```

```
P = 4×4
         0         0         0    0.0870
         0         0    0.0000    0.3045
         0    0.2425    0.2425   -0.1958
    1.0000   -0.9701   -0.9701    0.9281
D = 4×4
    -3     0     0     0
     0    -2     0     0
     0     0    -2     0
     0     0     0     0
```

```
P*D*inv(P)
```

```
Warning: Matrix is close to singular or badly scaled. Results may be inaccurate. RCOND =  1.015355e-17.
ans = 4×4
         0         0         0         0
    7.0000   -2.0000         0         0
   -8.0000         0   -2.0000         0
   16.0000    6.0000   -4.0000   -3.0000
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Find eigenvalues and eigenvectors using eigvec:
[P,D] = eigvec(A1)
```

```
P = 4×3
    0.0938         0         0
    0.3281         0         0
   -0.2109   -0.2500         0
    1.0000    1.0000    1.0000
D = 3×3
     0     0     0
     0    -2     0
```

```
     0     0    -3
```

```
% check if A1 = P*D*inv(P)? NO
% P*D*inv(P)
```

Conclusion: A1 is diagonalizable since $A_1 = PDP^{-1}$.

Note that P and D are not unique. Decompose A1 using two different sets of P and D.

A1 = $\begin{bmatrix} \quad & \quad & \quad \\ & & \\ & & \\ & & \end{bmatrix}\begin{bmatrix} \quad & \quad & \quad \\ & & \\ & & \\ & & \end{bmatrix}$*inv(P)

A1 = $\begin{bmatrix} \quad & \quad & \quad \\ & & \\ & & \\ & & \end{bmatrix}\begin{bmatrix} \quad & \quad & \quad \\ & & \\ & & \\ & & \end{bmatrix}$*inv(P)

1B

```
A2 = [5 0 0 0; 1 5 0 0; 0 1 5 0; 0 0 1 5]
```

```
A2 = 4×4
     5     0     0     0
     1     5     0     0
     0     1     5     0
     0     0     1     5
```

```
[P,D] = eig(A2)
```

```
P = 4×4
         0         0         0    0.0000
         0         0    0.0000   -0.0000
         0    0.0000   -0.0000    0.0000
    1.0000   -1.0000    1.0000   -1.0000
D = 4×4
     5     0     0     0
     0     5     0     0
     0     0     5     0
     0     0     0     5
```

```
% check if A2*P = P*D
A2 * P
```

```
ans = 4×4
         0         0         0    0.0000
         0         0    0.0000   -0.0000
         0    0.0000   -0.0000    0.0000
    5.0000   -5.0000    5.0000   -5.0000
```

```
P * D
```

```
ans = 4×4
```

```
         0         0         0    0.0000
         0         0    0.0000   -0.0000
         0    0.0000   -0.0000    0.0000
    5.0000   -5.0000    5.0000   -5.0000
```

```
% Yes, they are equal!
% check if A2 = P*D*inv(P)
P*D*inv(P)
```

```
Warning: Matrix is close to singular or badly scaled. Results may be inaccurate. RCOND =  6.842278e-46.
ans = 4×4
     5     0     0     0
     0     5     0     0
     0     0     5     0
     0     0     0     5
```

```
disp("This is not equal.")
```

```
This is not equal.
```

```
% write a single command for a basis of the eigenspace of A2 corresponding
% to lambda = 5 using NulBasis
NulBasis(A2 - 5 * eye(4))
```

```
ans = 4×1
     0
     0
     0
     1
```

Conclusion: A2 is NOT diagonalizable because

- (using P*D*inv(P))
- (using the dimension of the eigenspace)

1C

```
A = [6 3; 3 6]
```

```
A = 2×2
     6     3
     3     6
```

```
[P,D] = eigvec(A)
```

```
P = 2×2
     1    -1
     1     1
D = 2×2
     9     0
     0     3
```

```
dot(P(:,2),P(:,1))
```

```
ans = 0
```

Let v1 and v2 be two linearly indepedent eigenvectors of A. v1 and v2 are _______ because their dot product is zero.

The solution to the system $\begin{aligned} x_1' &= 6x_1 + 3x_2 \\ x_2' &= 3x_1 + 6x_2 \end{aligned}$ is $\mathbf{x}(t) = c_1\, e^{9t}\begin{bmatrix}1\\1\end{bmatrix} + c_2\, e^{?t}\begin{bmatrix}-1\\1\end{bmatrix}$.

1D

```
A = [4 0 2; 2 3 4; 0 0 -2]
```

```
A = 3×3
     4     0     2
     2     3     4
     0     0    -2
```

```
[P,D] = eigvec(A)
```

```
P = 3×3
    0.5000         0   -0.3333
    1.0000    1.0000   -0.6667
         0         0    1.0000
D = 3×3
     4     0     0
     0     3     0
     0     0    -2
```

A is diagonalizable because (provide a reason using matrix P)

The solution to the system $\begin{aligned} x_1' &= 4x_1 + 2x_3 \\ x_2' &= 2x_1 + 3x_2 + 4x_3 \\ x_3' &= -2x_3 \end{aligned}$ is $\mathbf{x}(t) = c_1\, e^{4t}\begin{bmatrix}\frac{1}{2}\\1\\0\end{bmatrix} + c_2\, e^{3t}\begin{bmatrix}0\\1\\0\end{bmatrix} + c_3\, e^{-2t}\begin{bmatrix}-\frac{1}{3}\\-\frac{2}{3}\\1\end{bmatrix}$.

Exercise 2 -- The Markov Chain

2A

```
A = [0.7 0.2 0.1; 0.2 0.7 0.2; 0.1 0.1 0.7]
```

```
A = 3×3
    0.7000    0.2000    0.1000
    0.2000    0.7000    0.2000
    0.1000    0.1000    0.7000
```

```
x = [0.7; 0.15; 0.15]
```

```
x = 3×1
    0.7000
    0.1500
    0.1500
```

```
x = A*x %x1 (probability vector for right now)
```

```
x = 3×1
    0.5350
    0.2750
    0.1900
```

```
x = A*x %x2 (probability vector for two years later)
```

```
x = 3×1
    0.4485
    0.3375
    0.2140
```

Now, 53.5% of those surveyed drive cars, 27.5% minivans, and 19% suv.

Two years later, 44.9% of those surveyed will drive cars, 33.8% minivans, and 21.4% suv.

2B see Sol_DiffEq.m

2C

```
% Method 1
x = [0.7; 0.15; 0.15] %x0 (initial probability vector)
```

```
x = 3×1
    0.7000
    0.1500
    0.1500
```

```
Sol_DiffEq(A,x)
```

```
    0.3500
    0.4000
    0.2500
ans = 3×1
    0.3500
    0.4000
    0.2500
```

```
%%%%%
% Method 2
[P,D] = eigvec(A)
```

```
P = 3×3
    1.4000   -1.0000   -1.0000
    1.6000    0.0000    1.0000
    1.0000    1.0000         0
D = 3×3
    1.0000         0         0
         0    0.6000         0
         0         0    0.5000
```

```
x = [0.7; 0.15; 0.15] %x0 (initial probability vector)
```

```
x = 3×1
    0.7000
    0.1500
    0.1500
```

```
% write a single command to solve for Pc = x0, where P = [v1 v2 v3], v_i is
% an eigenvector of A
```

```
P \ x
```

```
ans = 3×1
    0.2500
   -0.1000
   -0.2500
```

Express x0 as a linear combination of eigenvectors of A:

$$\mathbf{x}_0 = c_1\begin{bmatrix}1\\1\\1\end{bmatrix} + c_2\begin{bmatrix}-\frac{1}{3}\\-\frac{1}{3}\\1\end{bmatrix} + c_3\begin{bmatrix}1\\-\frac{3}{2}\\1\end{bmatrix}$$

(you must replace c_i with the solution that you get from the previous command)

Therefore, a general solution to the difference equation is $\mathbf{x}_k =$

Finally, $\lim_{k\longrightarrow\infty} \mathbf{x}_k = \begin{bmatrix}0.35\\0.40\\0.25\end{bmatrix}$

Do you have the same solution from both methods?

Conclusion: In the long run, we expect 35% of those surveyed will drive cars, 40% minivans, and 25% suv.

The steady-state vector for matrix A is $\begin{bmatrix}0.35\\0.4\\0.25\end{bmatrix}$.

2D

```
x = [0.9; 0.05; 0.05] %x0 (initial probability vector)
```

```
x = 3×1
    0.9000
    0.0500
    0.0500
```

```
Sol_DiffEq(A,x)
```

```
    0.3500
    0.4000
    0.2500
ans = 3×1
    0.3500
    0.4000
    0.2500
```

```
x = [0.1; 0.45; 0.45] %x0 (initial probability vector)
```

```
x = 3×1
    0.1000
    0.4500
    0.4500
```

```
Sol_DiffEq(A,x)
```

```
    0.3500
    0.4000
    0.2500
ans = 3×1
    0.3500
    0.4000
    0.2500
```

The steady-state vector is $\begin{bmatrix} 0.35 \\ 0.4 \\ 0.25 \end{bmatrix}$ for any initial vector x0 because (state the conditions of theorem here)

Exercise 3 -- Inner Products & Orthogonal Projections

3A

```
u1 = [2 1 3 -2].'
```

```
u1 = 4×1
     2
     1
     3
    -2
```

```
u2 = [1 2 3 4].'
```

```
u2 = 4×1
     1
     2
     3
     4
```

```
v1 = [1 -2 3 -4].'
```

```
v1 = 4×1
     1
    -2
     3
    -4
```

```
v2 = [2 4 6 8].'
```

```
v2 = 4×1
     2
     4
     6
     8
```

```
v3 = [-3 -6 -9 -12].'
```

```
v3 = 4×1
    -3
    -6
    -9
   -12
```

```
%do the calcuations for dot product and norms
%Verify the Cauchy-Schwarz inequality for u1 and v1
abs(dot(u1,v1))
```

```
ans = 17
```

```
norm(u1)*norm(v1)
```

```
ans = 23.2379
```

```
%repeat the process for u2, v2, and u2, v3
abs(dot(u2,v2))
```

```
ans = 60
```

```
norm(u2)*norm(v2)
```

```
ans = 60
```

```
abs(dot(u2,v3))
```

```
ans = 90
```

```
norm(u2)*norm(v3)
```

```
ans = 90
```

```

```

The Cauchy-Schwarz inequality:

- For u1 and v1: $|\mathbf{u_1} \cdot \mathbf{v}_1| < \|\mathbf{u}_1\| \, \|\mathbf{v}_1\|$
- For u2 and v2: $|\mathbf{u_2} \cdot \mathbf{v}_2| = \|\mathbf{u}_2\| \, \|\mathbf{v}_2\|$
- For u2 and v3: $|\mathbf{u_2} \cdot \mathbf{v}_3| = \|\mathbf{u}_2\| \, \|\mathbf{v}_3\|$
- "=" when u and v are _______.

3B see projection.m

3C

```
[y_hat,z] = projection(v1,u1)
```

```
y_hat = 4×1
    1.8889
    0.9444
    2.8333
   -1.8889
z = 4×1
   -0.8889
   -2.9444
    0.1667
   -2.1111
```

```
y_hat+z
```

```
ans = 4×1
```

```
      1
     -2
      3
     -4
```

```
dot(z,u1)
```

```
ans = 8.8818e-16
```

Verify if v1 = y_hat + z and if z is orthogonal to u1

```
[y_hat,z] = projection(v2,u2)
```

```
y_hat = 4×1
     2
     4
     6
     8
z = 4×1
     0
     0
     0
     0
```

```
y_hat+z
```

```
ans = 4×1
     2
     4
     6
     8
```

Verify if v2 = y_hat + z and if z is orthogonal to u2

z is the zero vector because ...