EGEE-2110 Engineering Analysis
Homework 1

Kaitlyn Wiseman

05 March 2021

1

Problem 1 was to create the following four matrices:

$$\mathbf{A} = \begin{bmatrix} -4 & -3 & 2 \\ 4 & 3 & -1 \\ -3 & -2 & 1 \end{bmatrix} \qquad \mathbf{B} = \begin{bmatrix} 1 & -1 & -3 \\ -1 & 2 & 4 \\ 1 & 1 & 1 \end{bmatrix} \qquad \mathbf{C} = \begin{bmatrix} 2 & 1 & 3 \\ -1 & 0 & 2 \\ 3 & 2 & 1 \end{bmatrix} \qquad \mathbf{D} = \begin{bmatrix} 0 & 1 & 2 \\ 1 & 2 & 0 \\ 0 & 2 & 1 \end{bmatrix}$$

The code I used to create the four matrices is

```
% Create 4 matrices:
A = [-4 -3  2;
      4  3 -1;
     -3 -2  1];
B = [ 1 -1 -3;
     -1  2  4;
      1  1  1];
C = [ 2  1  3;
     -1  0  2;
      3  2  1];
D = [ 0  1  2;
      1  2  0;
      0  2  1];
```

The code successfully created the four matrices in memory.

2

Problem 2 was to show by multiplication that $\mathbf{B}$ is not the inverse of $\mathbf{A}$.

If $\mathbf{B}$ is the inverse of $\mathbf{A}$, $\mathbf{AB}$ should equal the identity matrix. I used the following code to test whether this was true:

```
% Show by multiplication that B is not the inverse of A.
disp(' If B is the inverse of A, AB should equal I:')
A * B
disp(' =\= I.')
```

The result of that multiplication is

$$\text{ans} = \begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & -1 \\ 0 & 0 & 2 \end{bmatrix}$$

which is not the identity matrix. Hence, $\mathbf{B}$ is not the inverse of $\mathbf{A}$.

3

Problem 3 was to find the actual inverse of $\mathbf{A}$ and prove it is the inverse. If $\mathbf{A}^{-1}$ truly is the inverse of $\mathbf{A}$, then $\mathbf{A}^{-1}\mathbf{A}$ will be the identity matrix.

To find $\mathbf{A}^{-1}$, I used Matlab's inv() function. To prove $\mathbf{A}^{-1}$ was indeed the inverse of $\mathbf{A}$, I performed matrix multiplication of the two, and found the result to be the identity matrix. I used the following code:

```
% Find the actual inverse of A and prove it is the inverse.
invA = inv(A)
A * invA
disp(' = I.')
```

The inverse of A was

$$\mathbf{A}^{-1} = \begin{bmatrix} 1 & -1 & -3 \\ -1 & 2 & 4 \\ 1 & 1 & 0 \end{bmatrix}$$

and the result of the calculation was the identity matrix, hence proving that $\mathbf{A}^{-1}$ is indeed the inverse of $\mathbf{A}$.

4

Problem 4 was to compute the point-by-point multiplication of C and D. In Matlab, to compute point-by-point multiplication (as opposed to matrix multiplication) of the entries of a matrix, you have to precede the operator with a period, i.e. '.*'. To compute this multiplication, I used the following code:

```
% Compute the point-by-point multiplication of C and D.
C .* D
```

The result of this calculation was

$$\begin{bmatrix} 0 & 1 & 6 \\ -1 & 0 & 0 \\ 0 & 4 & 1 \end{bmatrix}.$$

5

Problem 5 was to find the characteristic polynomials for **C** and **D**. In Matlab, to find the coefficients of the characteristic polynomial of a matrix, you can use the poly() function. In order to find these coefficients, I used the following code:

```
% Find the characteristic polynomials for C and D.
polyC = poly(C)
polyD = poly(D)
```

The results of these calculations were

$$\text{polyC} = \begin{bmatrix} 1 & -3 & -10 & 7 \end{bmatrix} \qquad \text{polyD} = \begin{bmatrix} 1 & -3 & 1 & -3 \end{bmatrix}.$$

Hence, from these calculations, the characteristic polynomial of **C** is

$$\lambda^3 - 3\lambda^2 - 10\lambda + 7 = 0,$$

and the characteristic polynomial of **D** is

$$\lambda^3 - 3\lambda^2 + \lambda - 3 = 0$$

6

Problem 6 was to find the roots of the polynomials found in Problem 5. Matlab has a function roots() that accepts a vector containing the coefficients of a polynomial and returns the roots of that polynomial. So, by plugging in the results of Problem 5 (stored in the variables polyC and polyD), I could easily find the roots. I used the following code:

```
% Find the roots of the polynomials.
roots(polyC)
roots(polyD)
```

From these calculations, the roots of the characteristic polynomial for **C** are

$$\lambda = 4.743, -2.3952, 0.6109,$$

and the roots of the characteristic polynomial for **D** are

$$D\lambda = 3, j, -j.$$

7

Problem 7 was to show that the determinant $|\mathbf{CD}|$ is equal to $|\mathbf{C}|\,|\mathbf{D}|$. In order to do this, I computed both quantities and compared them. In Matlab, the function det() computes the determinant of the input matrix. I used the following code to carry out the computations:

```
% Show that the determinant |CD| is equal to |C||D|.
det(C * D)
det(C) * det(D)
```

The result of both calculations was -21.

8

Problem 8 was to first create the following matrix in memory:

$$\mathbf{E} = \begin{bmatrix} 5 & 3 & -4 \\ 6 & -2 & 2 \\ 7 & 7 & -9 \end{bmatrix}.$$

Then, the problem required I find the matrix of eigenvectors of $\mathbf{E}$ and call it $\mathbf{P}$. Finally, I showed that $\mathbf{P}^{-1}\mathbf{E}\mathbf{P}$ is the diagonalized version of $\mathbf{E}$—the similar matrix containing $\mathbf{E}$'s eigenvalues on its diagonal. I used the following code to do so:

```
1  % Create the matrix E.
2  E = [ 5  3 -4;
3        6 -2  2;
4        7  7 -9]
5  %   a) Find the matrix containing the eigenvectors of E (call it P).
6  [P,D] = eigs(E)
7  %   b) Show that inv(P)EP results in a diagonal matrix which contains the
8  %      eigenvalues of E on the main diagonal.
9  P \ E * P   % I would have used inv(P) * E * P, but it was slightly
10             % inaccurate compared to this
```

The result of line 6 was

$$\mathbf{P} = \begin{bmatrix} 0.3112 & -0.4627 & 0.0454 \\ -0.4400 & -0.6543 & 0.7711 \\ 0.8423 & 0.5982 & 0.6351 \end{bmatrix} \qquad \mathbf{D} = \begin{bmatrix} -10.0711 & 0 & 0 \\ 0 & 4.0711 & 0 \\ 0 & 0 & 0.0000 \end{bmatrix}$$

$\mathbf{P}$ is a matrix containing $\mathbf{E}$'s eigenvectors. $\mathbf{D}$ is the diagonal matrix containing $\mathbf{E}$'s eigenvalues on its diagonal, and thus it should match the result of line 9. The result of line 9 was

$$\begin{bmatrix} -10.0711 & 0 & 0 \\ 0 & 4.0711 & 0 \\ 0 & 0 & 0.0000 \end{bmatrix},$$

which indeed is equal to $\mathbf{D}$. Hence, $\mathbf{P}^{-1}\mathbf{E}\mathbf{P}$ does result in a diagonal matrix which contains the eigenvalues of $\mathbf{E}$ on its diagonal.

9

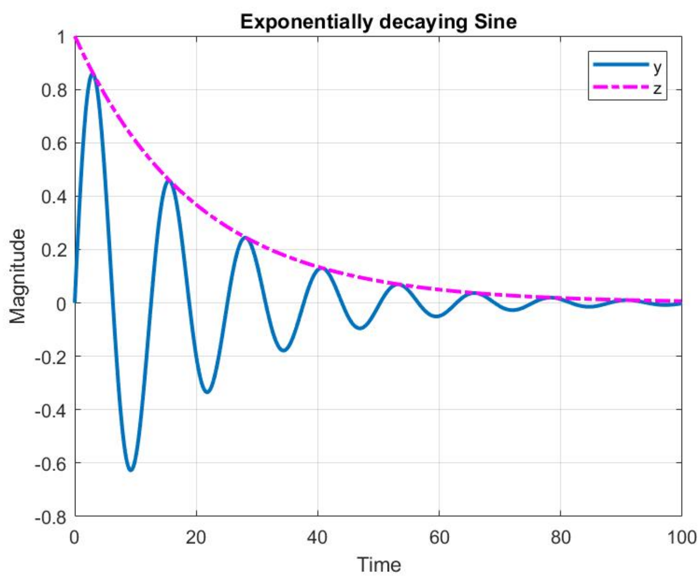
Problem 9 involved plotting the following two functions on the same plot from 0 to 100 in steps of 0.001:

$$y = e^{-0.05t}\sin(t/2), \qquad z = e^{-0.05t}.$$

The x-axis label was "Time", the y-axis label was "Magnitude", the grid was on, and the title was "Exponentially decaying Sine". I used the following code to implement this:

```
%     a) Plot the function y from zero to 100 in steps of 0.001.  Label the
%        x-axis "Time", and the y-axis "Magnitude".  Display the grid.  Give
%        the plot the title "Exponentially decaying Sine".
t = 0:0.001:100;
y = exp(-0.05*t) .* sin(t/2);

figure(1)
plot(t,y,'linewidth',2)
grid on
xlabel('Time')
ylabel('Magnitude')
title('Exponentially decaying Sine')

hold on

%     b) Plot the function z on the same plot in a different color.  Give
%        the figure a legend, and identify each function with a name ("y"
%        and "z" will work).
z = exp(-0.05*t);

plot(t,z,'linewidth',2,'color','m','linestyle','-.')
legend('y','z')

hold off
```

This code produced the following figure:



10

Problem 10 was to compute the RMS value of the y function from Problem 9. To take the RMS value, first square each value, then take the mean of those squares. Finally, take the square root of the mean. I used the following code to compute y_{RMS}:

```
% Compute the RMS value of the signal y from problem 9.  Recall that the
% name "RMS" tells you the algorithm in reverse order.
yRMS = sqrt(mean(y.^2))
```

The result of this calculation was $y_{\text{RMS}} = 0.2225$.