Exercises 1a: MAE Unit 1

Name: Josep Famadas Alsamora

Collaboration Info: I have used some information from mathworks.com.

Exercise 1. Variables and operators.

Explanation:

(1) I created a vector from 2 to n (with a step of 1) and multiplied its terms using ‘prod’.

(2) Starting from a vector previously declared as [1,1], I set the parameter n to the size of this vector ‘v’. After that, I create a new vector ‘r’ which first and last value are ‘1’ and the other ones were fulfilled with a for that summed the numbers of ‘v’ in packs of 2. Once ‘r’ was full, I set the new value of ‘v’ as the current value of ‘r’.

(3) I created the random vector. First, I eliminated the values lower than 30 or equal, then I used the function ‘rem’ dividing the vector by a same-sized vector of 3s, and finally I erased the values different from 0 (what means they had not been multiples of 3). Using the function ‘length’ on the final vector I could know the amount of numbers that satisfy both initial conditions.

Apart from that, I calculate the percentage of numbers and compare it to 23 (which is the expected value).

MATLAB Code:

(1)

```
n = 5;
f = prod(2:1:n)
```

(2)

```
n = length(v);
r(1) = 1;
r(n+1) = 1;
for i = 2:n

    r(i)=v(i-1)+v(i);

end
v=r
```

(3)

```
n = 100000;

v = randi([4,100],1,n);
v2 = v(v>30); %Erase numbers <= 30
v3 = rem(v2,3*ones(1,length(v2))); %Obtain 0 if the number is
multiple of 3
v4 = v3(v3==0); %Erase numbers ~=0 (which means they were not
multiple of 3)
```

```
res = length(v4)
percentage = 100*res/n %The expected value is 23
```

Results:

(1) f = 120

(2) The values on the first 4 iterators are

1- 1 2 1

2- 1 3 3 1

3- 1 4 6 4 1

4- 1 5 10 10 5 1

(3) res = 23807

percentage = 23.8070

Comments:

Exercise 2. Vectors and matrix operations.

Explanation:

(1) First of all, I create a vector of 0s with the same length as columns has the matrix (this step is explained in the comments). I added this vector at the bottom of the matrix and I used 'flip' with the resultant matrix and finally I erased the first row (which is the 0s row).
(2) I generate A using 'ones', B using 'eye' and C using 'zeros'.
(3) I create first the two top rows multiplying 2 * a 2x2 submatrix of B and adding on the right C transposed. The 3 bot rows were created by concatenating -3* A transposed, A transposed and 5* B. Finally I concatenated everything.
(4) I created X and Y using the function 'meshgrid' and then I subtracted X from Y.
(5) First of all, I created a matrix with m times the vector an transposed as columns. Then I created another matrix with n times the numbers from 0 to m-1 as rows. Finally I elevate point by point the first matrix to the second.
(6) I repeated the same steps as before but now setting m = n = length of the vector an. Then I calculated the determinant and the eigens with their corresponding functions. The first comprovation was made with starting with a variable with the value of an(j) and a for iterating the 'j' from 2 to n and in every loop I created a vector of 'i' from 1 to j-1 and multiplied the new an(j)-an(i) by the current value of the variable. I divided the final value of the variable by the determinant expecting to get a 1.
The second comprovation was made by dividing the determinant by the product of the eigen values and expecting to get a 1 again.
(7) The first time I solved the system by multiplying the inverse matrix of coefficients by the vector of results.
The second time I solved the system by dividing the matrix of coefficients by the vector of results using the backslash.

MATLAB Code:

(1)
```
A = [1 2 3 ; 4 5 6]
[r,c]=size(A);
v = zeros(1,c);
B = [A;v];
C = flip(B);
R = C(2:r+1,:)
```

(2)
```
A = ones(1,3);
B = eye(3);
C = zeros(3,2);
```

(3)
```
M1 = [2*B(1:2,1:2) C'];
M2 = [-3*A' A' 5*B];
M = [M1 ; M2]
```

(4)

```
v = [ 1 2 3 4]

[X,Y] = meshgrid(v,v);
M = Y-X
```

(5)

```
m=5;
an = [1 2 3];
n = length(an);

A = (repmat(an',1,m)).^(repmat(0:m-1,n,1))
```

(6)

```
an = [1 2 3 4];
n = length(an);
m = n;

A = (repmat(an',1,m)).^(repmat(0:m-1,n,1))

determ = det(A);
eigen = eig(A);

product = an(1);
for j = 2:n
 i = 1:j-1;
 product = product * prod(an(j)-an(i));
end

comprovation1 = determ/product %This value is 1 if the
verification is true
comprovation2 = determ/prod(eigen) %This value is 1 if the
verification is true
```

(7)

```
A = [2 1 -2 ; 4 2 -3 ; -2 0 1];
b = [4 ; 9 ; -3];

X = inv(A)*b;
x1 = X(1)
x2 = X(2)
x3 = X(3)

Y = A\b;
y1 = Y(1)
y2 = Y(2)
y3 = Y(3)
```

<u>Results:</u>

(1) R = 4   5   6
       1   2   3

(2) No results

(3) M =

$$\begin{matrix} 2 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ -3 & 1 & 5 & 0 & 0 \\ -3 & 1 & 0 & 5 & 0 \\ -3 & 1 & 0 & 0 & 5 \end{matrix}$$

(4) M =

$$\begin{matrix} 0 & -1 & -2 & -3 \\ 1 & 0 & -1 & -2 \\ 2 & 1 & 0 & -1 \\ 3 & 2 & 1 & 0 \end{matrix}$$

(5) A =

$$\begin{matrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 & 16 \\ 1 & 3 & 9 & 27 & 81 \end{matrix}$$

(6) Both comprovations are 1

(7) X = [2 2 1]
Y = [2 2 1]

<u>Comments:</u>

(1) At the begining I didn't used the 0s vector and It worked perfect with everything except with horizontal vectors (they should keep the same but the function flip flipped the row) so the solution was to convert it to a matrix by adding a 0s vector.

Exercise 3. Basic plots.

Explanation:

(1) I generate the vector using the operator ':' and plotted it with the function 'plot'.
(2) I hold the plot to be able to superpose the new function on the same graphic.
(3) First of all, I create the two subplots. Then using the previous code, I put the first graphic on the first subplot, and then the Archimedes spiral on the second. Finally, I gridded both plots and equalized them.

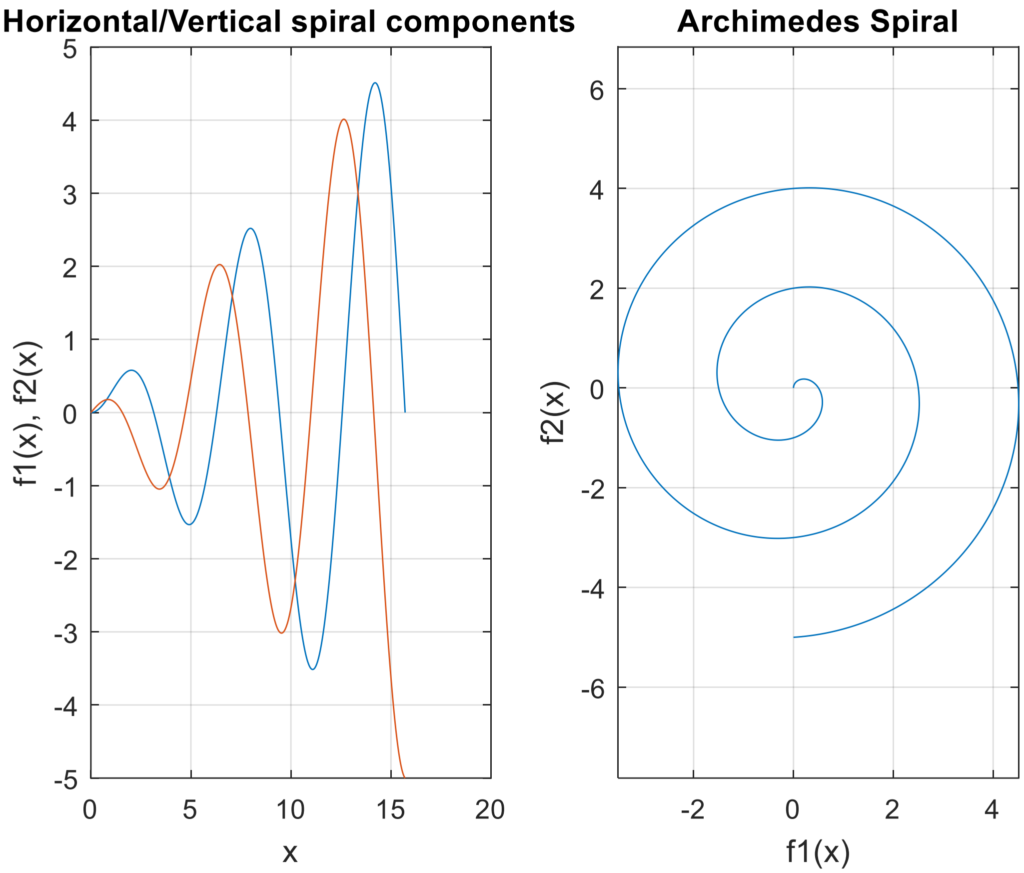
MATLAB Code:

(1) , (2) and (3)

```
x = 0:pi/200:5*pi;
f1 = x/pi.*sin(x);
f2 = x/pi.*cos(x);

subplot(1,2,1);
plot(x,f1);
xlabel ('x');
ylabel ('f1(x), f2(x)');
title ('Horizontal/Vertical spiral components');
hold on
plot(x,f2);
grid on;
hold off
subplot(1,2,2);
plot(f1,f2);
xlabel ('f1(x)');
ylabel ('f2(x)');
title ('Archimedes Spiral');
grid on;
axis equal;
```

Results:

The final result is the following figure.



Comments:

Exercise 4. Computation with polynomials.

Explanation:

(1) I created a vector with the zeros positions and used the function ‘poly’ to find the polynomial coefficients.
(2) I designed the vector w intervals of pi/200 and I found z with the function ‘exp’. Then, I evaluate the vector on the system using the function ‘polyval’ with z and the coefficients vector.
(3) I plotted the magnitude in dB using 10*log10 of the absolute value of the response. In order to see it with more resolution I set the vertical axis from -3 to 3 dB and plotted it again.

MATLAB Code:

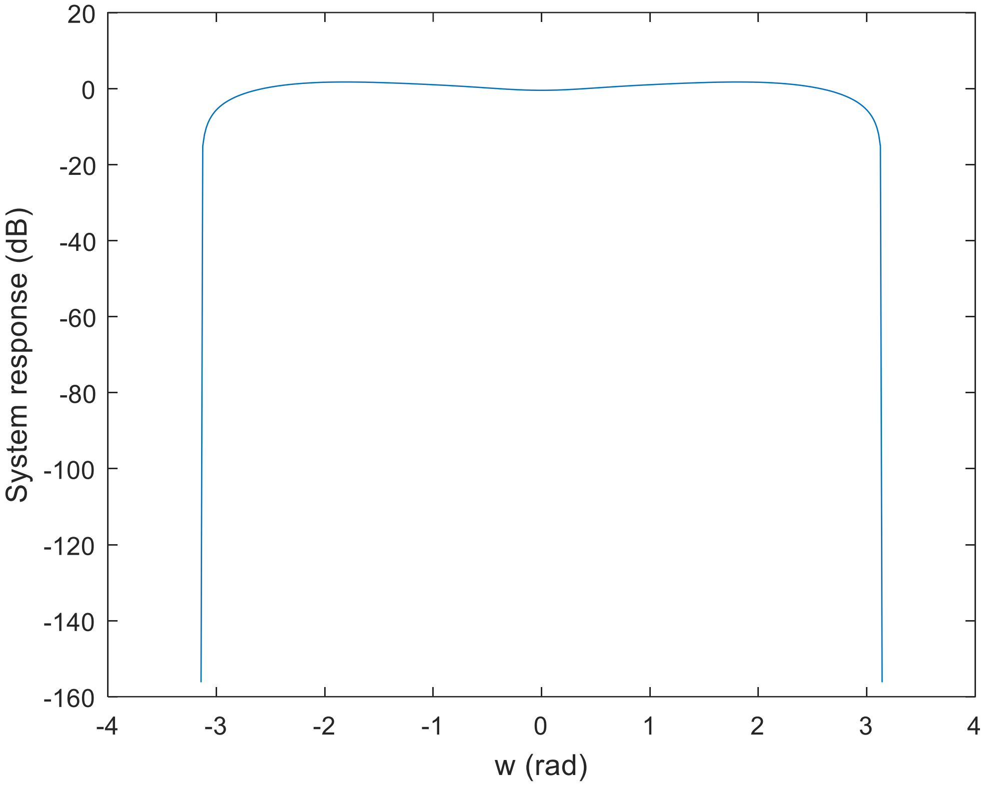
(1) , (2) and (3)

```
v = [0.1+0.3*1i 0.1-0.3*1i 0.5 -1];
coef = poly(v);

w = -pi:pi/200:pi;
z = exp(1i*w);
x = polyval(coef,z);

plot(w,10*log10(abs(x)))
xlabel('w (rad)')
ylabel('System response (dB)')
axis([-pi pi -3 3])
```

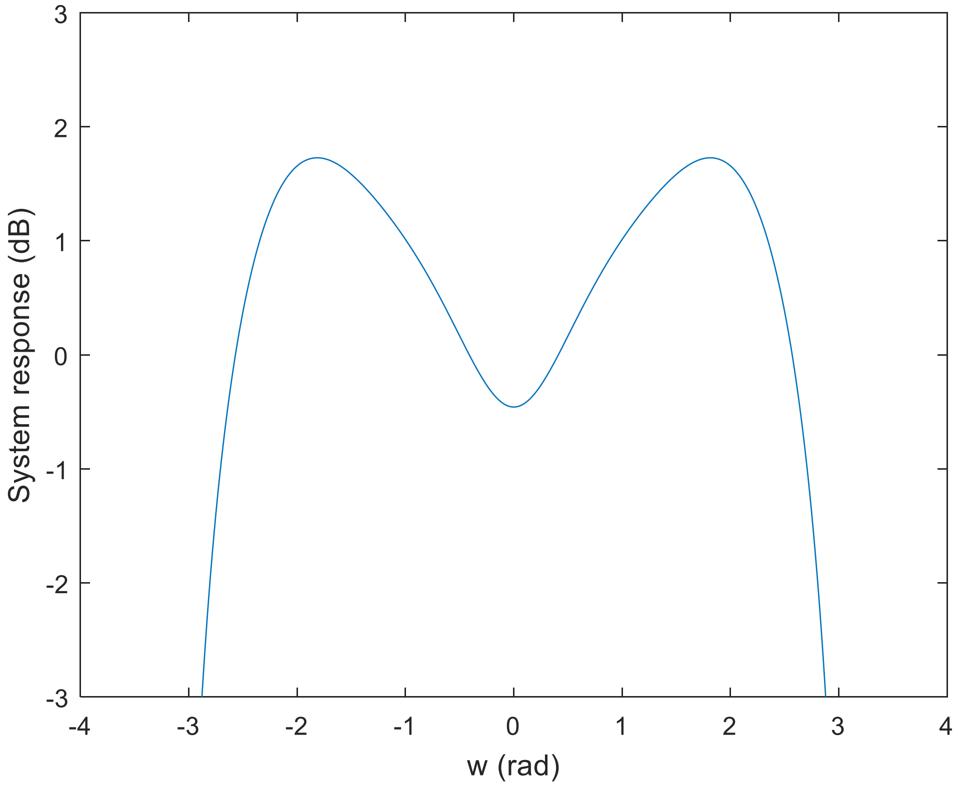
Results:

(Plots in the next page)

This is the first figure:



This is the zoomed figure



Comments: