Table of Contents

Problem 1. 1
Problem 2. 2
Problem 3. 3
Problem 4. 5

Dan Otieno. EE 384 -> Spring '24. Classwork 0 - Intro to MATLAB. Due date: 01/16/24.

Problem 1.

Q1.1. What is the result of: a) a+b, b) a*b, c) a.*b a & b are column vectors.

```
a = [2; 5; 8];
b = [3; 13; 8];
a_result = a + b;
b_result = a .* b;

% Print/Display Results.
fprintf('PROBLEM 1.\n')
fprintf('Q1.1:\n');
fprintf('a) Result of a + b: \n');
disp(a_result);
fprintf('b) a * b is not a valid operaration.\n\n');
fprintf('c) Result of a .* b: \n');
disp(b_result);

% Q1.2.
% Repeat 1.1 but with 'a' as a matrix:
aMatrix = [1 2 -3; 2 1 2; 4 -2 1];
a2_result = aMatrix + b;
b2_result = aMatrix * b;
c2_result = aMatrix .* b;

% Print/Display Results.
fprintf('--------------------------------------------------------------------------------
\n');
fprintf('Q1.2:\n');
fprintf('Repeating Q1.1 with "a" as a Matrix \n');
fprintf('Result of a + b: \n');
disp(a2_result);
fprintf('Result of a * b: \n');
disp(b2_result);
fprintf('Result of a .* b: \n');
disp(c2_result);
fprintf('================================================================================
\n');
```

```
PROBLEM 1.
Q1.1:
```

```
================================================================================
```

```
a) Result of a + b:
     5
    18
    16

b) a * b is not a valid operaration.

c) Result of a .* b:
     6
    65
    64

--------------------------------------------------------------------------------
Q1.2:
Repeating Q1.1 with "a" as a Matrix
Result of a + b:
     4     5     0
    15    14    15
    12     6     9

Result of a * b:
     5
    35
    -6

Result of a .* b:
     3     6    -9
    26    13    26
    32   -16     8

================================================================================
```

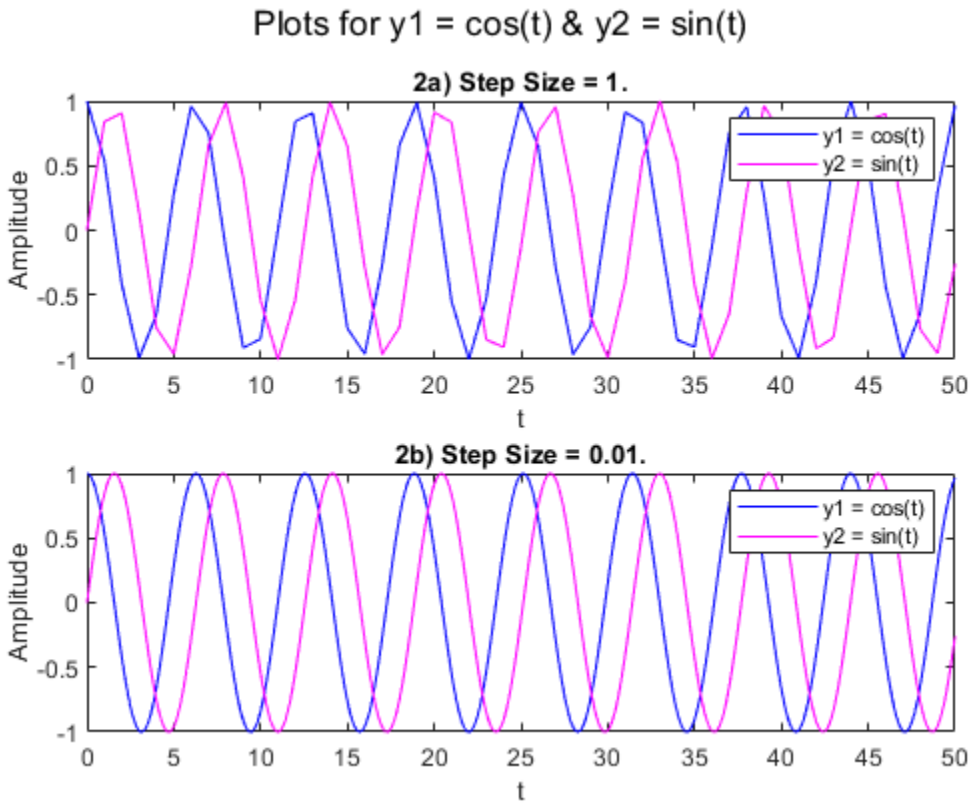
Problem 2.

Plot the functions: y1 = cos(t), y2 = sin(t). 't' is a vector from 0 to 50 with increasing step: a) 1, b) 0.01.

```
% Increasing step = 1.
t1 = 0:1:50;
y1 = cos(t1);
y2 = sin(t1);

% Increasing step = 0.01.
t2 = 0:0.01:50;
y3 = cos(t2);
y4 = sin(t2);

% Plotting the functions.
figure;
sgtitle('Plots for y1 = cos(t) & y2 = sin(t)');
% First subplot.
subplot(2, 1, 1);
plot(t1, y1, 'b-', t1, y2, 'm-');
title('2a) Step Size = 1.');
```

```
legend('y1 = cos(t)', 'y2 = sin(t)');
xlabel('t');
ylabel('Amplitude');

% Second subplot.
subplot(2, 1, 2);
plot(t2, y3, 'b-', t2, y4, 'm-');
title('2b) Step Size = 0.01.');
legend('y1 = cos(t)', 'y2 = sin(t)');
xlabel('t');
ylabel('Amplitude');

fprintf('PROBLEM 2:\n');
fprintf('Yes, signals are smoother when the increasing step is reduced.\n');
fprintf('==========================================================================================
\n');
```

```
PROBLEM 2:
Yes, signals are smoother when the increasing step is reduced.
==========================================================================================
```



Problem 3.

Program to solve system of equations of 3 variables. Matrix inverse method, should include user prompt to input coefficients. Assume users have to give coefficients in the order ‘a’, ‘b’, ‘c’, and ‘d’. Test using 2x+3y+z=3, x+3y-z=6 and 2x+2y=7.

```
fprintf('PROBLEM 3.\n');
fprintf('Program will solve system of equations for variables using matrix
 inverse method.\n');
% To test user input, uncomment the input prompt lines below.
% fprintf('Enter coefficients below in the form ax + by + cz = d:\n');
% Coefficient inputs for the 1st equation.
% a1 = input('a1 = : ');
% b1 = input('b1 = : ');
% c1 = input('c1 = : ');
% d1 = input('d1 = : ');
% fprintf('\n');
% Coefficient inputs for the 2nd equation.
% a2 = input('a2 = : ');
% b2 = input('b2 = : ');
% c2 = input('c2 = : ');
% d2 = input('d2 = : ');
% fprintf('\n');
% Coefficient inputs for the 3rd equation.
% a3 = input('a3 = : ');
% b3 = input('b3 = : ');
% c3 = input('c3 = : ');
% d3 = input('d3 = : ');

% Hard-coded inputs as specified in question, for testing.
a1 = 2;
b1 = 3;
c1 = 1;
d1 = 3;

a2 = 1;
b2 = 3;
c2 = -1;
d2 = 6;

a3 = 2;
b3 = 2;
c3 = 0;
d3 = 7;

A = [a1 b1 c1; a2 b2 c2; a3 b3 c3];
B = [d1; d2; d3];

% Calculate the result.
aInv = inv(A);
result = aInv * B;

% Print/Display result.
fprintf('a1 = 2, b1 = 3, c1 = 1, d1 = 3.\n');
fprintf('a2 = 1, b2 = 3, c2 = -1, d2 = 6.\n');
fprintf('a3 = 2, b3 = 2, c3 = 0, d3 = 7.\n');
fprintf('\nThe solution for this system of equations is:\n');
fprintf('x = %.4f\n', result(1));
fprintf('y = %.4f\n', result(2));
fprintf('z = %.4f\n', result(3));
```

```
fprintf('================================================================================
\n');

PROBLEM 3.
Program will solve system of equations for variables using matrix inverse
 method.
a1 = 2, b1 = 3, c1 = 1, d1 = 3.
a2 = 1, b2 = 3, c2 = -1, d2 = 6.
a3 = 2, b3 = 2, c3 = 0, d3 = 7.

The solution for this system of equations is:
x = 4.0000
y = -0.5000
z = -3.5000
================================================================================
```

Problem 4.

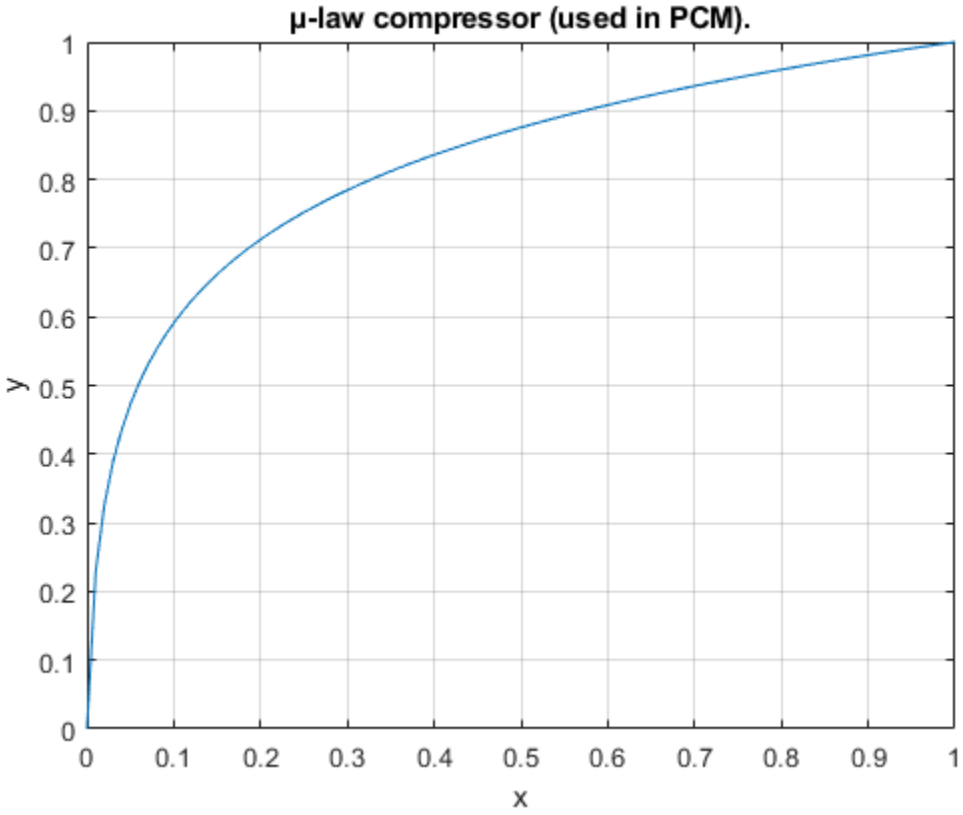
Write program to calculate log(1 + mu * abs(x)) / log(1 + mu) .* sign(x). Program must include a user prompt to input the parameter ‘μ’ and input ‘x’. Test Program by plotting 'y' according to 'x'. μ = 255, x = vector changing from 0 to 1 with inceasing step 0.01.

```
fprintf('PROBLEM 4.\n');
% To test user input, uncomment the input prompt lines below.
% fprintf('Enter values for %μ  and ‘x’:');
% User inputs for μ and x.
% mu = input('μ = : ');
% x = input('x = : ');

% Hard-coded inputs as specified in question, for testing.
mu = 255;
x = 0:0.01:1;

% Calculate 'y' from the function.
y = fnResult(mu, x);

% Plotting the results.
figure;
plot(x, y);
title('μ-law compressor (used in PCM).');
xlabel('x');
ylabel('y');
grid on;

fprintf('See (x, y) plot.\n');
fprintf('================================================================================
\n');

% Function definition.
function y = fnResult(mu, x)
    y = log(1 + mu * abs(x)) / log(1 + mu) .* sign(x);
end

PROBLEM 4.
```

```
================================================================================
```

See (x, y) plot.

```
================================================================================
```



Published with MATLAB® R2023a