CS1MA20 - Spring Term Assignment

Module Title: Mathematics and Computation

Module Code: CS1MA20

Student Number: 30021591

Git Repository: https://csgitlab.reading.ac.uk/il021591/cs1ma20_spring_assignment.git

Lecturer responsible: Professor Xia Hong

Weighting of the Assignment: 35%

Date: 09/03/2022

Actual hrs spent for the assignment: 26 hrs

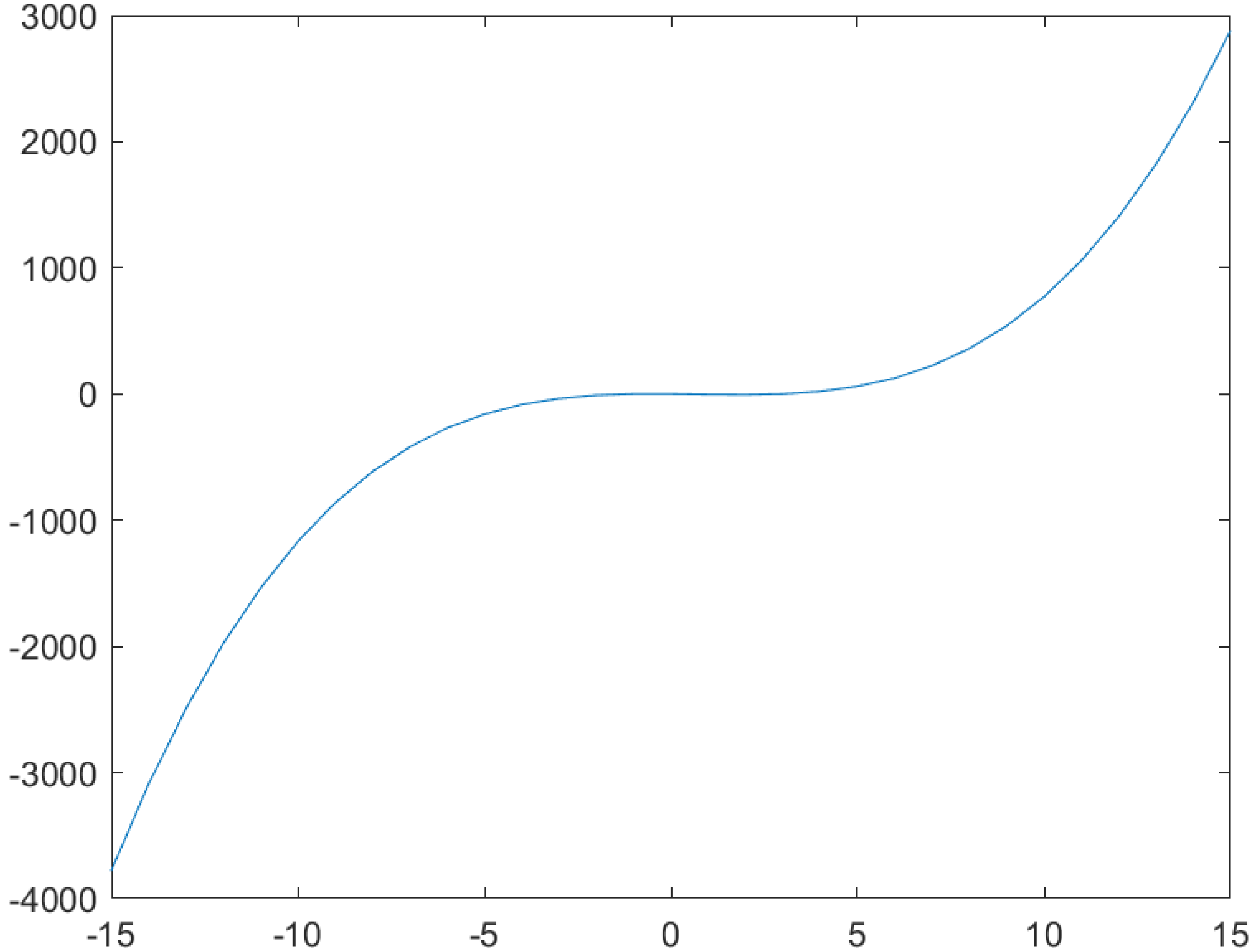
Assignment evaluation : Learnet Calculus and Impoved confidence in matlab

Task 1

$$f(x) = x^3 + (-2)x^2 + (-3)x + (0)$$

1. Plot the function f(x) with input x ranging from -15 to 15

```
x = [-15:15]
y = x.^3+(-2)*x.^2+(-3)*x+(0)
plot(x,y)
```

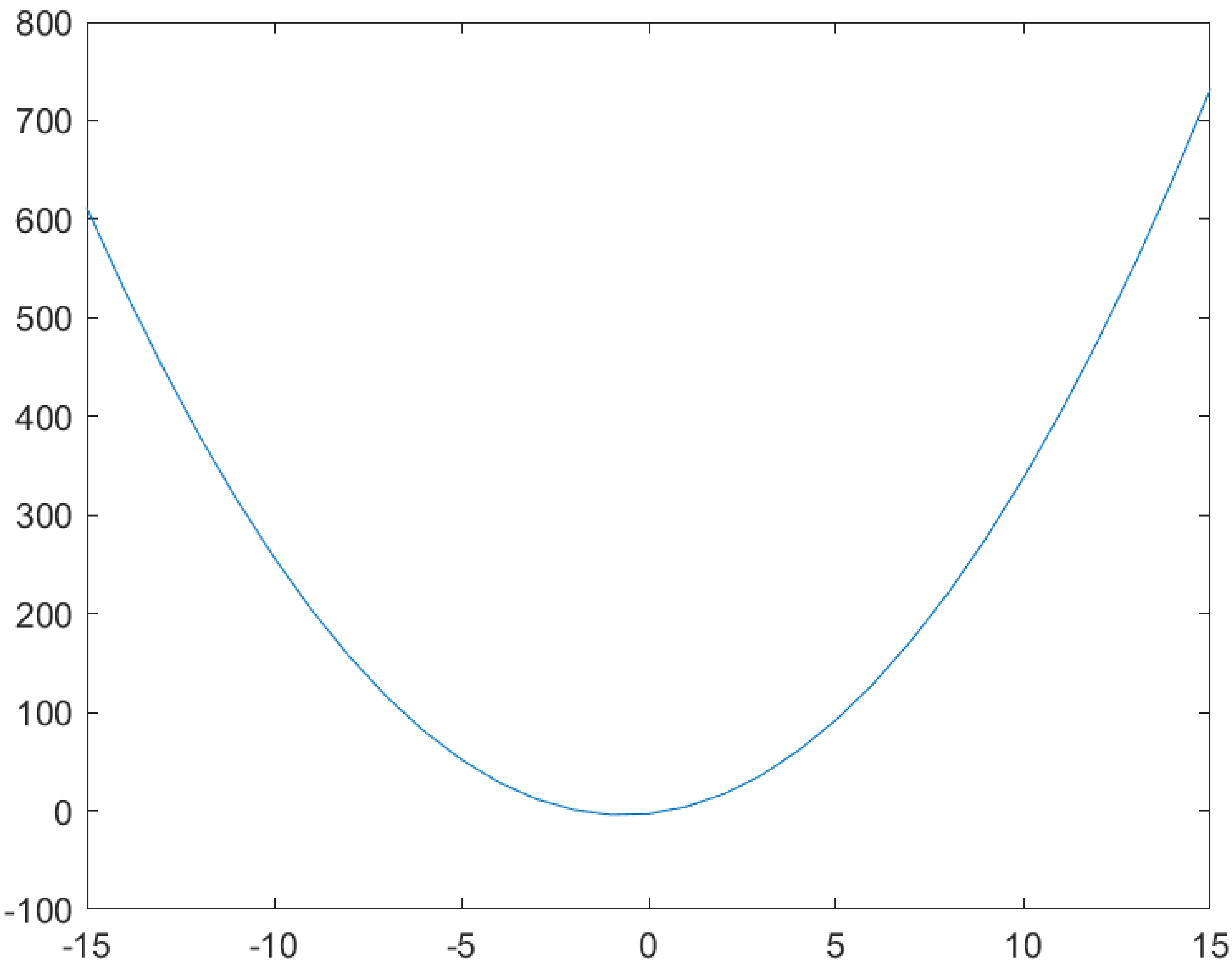


2. Find $\frac{df}{dx}$

$$3x^2 + 4x - 3$$

3. Plot $\frac{df}{dx}$

```
x = [-15:15]
y = 3*x.^2+4*x-3
plot(x,y)
```



4. Write code in Matlab that uses Newton-Raphson method to find all three roots of *f(x)*. You should implement the Newton-Raphson method yourself, rather than relying on an existing library. Show your code. Demonstrate that your code works by showing how you call it from the command line and what output Matlab gives. (If you found repeated roots in your question. Explain which ones are repeated.)

> -1, 0 , 3

```
function x = NewtonRaphsonMethod(x0)
% ^ turns file to a function and lets arg 0 be the estimate
```

```
format long;

x = x0;
f = @(x) x.^3+(-2)*x.^2+(-3)*x+(0);
% ^ function
fd = @(x) 3*x.^2+4*x-3;
% ^ derivite of the function

n = 100;
% ^ ittarations until answer found (approximations)
oldx = 0.1;

for i = 0:n
    x = x - f(x) / fd(x);
    % ^ the Newton Raphson Method
    % v break if answer found
    if oldx == x
        break
    else
    end
end
end
```

```
>> NewtonRaphsonMethod(30)

ans =

   3.000000000000000
```

5. Explain how different initializations are used for each root.

> -1, 0 , 3

> I used randoms numbers to find the closes root of the function. varable `n` was also incresed to improve the approximation of the root. I found three roots for the function -1, 0 , 3 using the arguments -1, 0 and 30. 3 is repiteded.

```
>> NewtonRaphsonMethod(30)

ans =

   3.000000000000000
```

```
NewtonRaphsonMethod(-24)

ans =

     0
```

```
>> NewtonRaphsonMethod(-1)

ans =

    -1
```

Task 2

$$f(x) = (1)x^2 + (6)x + (5)$$

1. Find $\int f(x)dx$ by hand.

$$\int f'(x) = (1)x^2 + (6)x + (5)$$

$$\int f'(x) = 2x + 6$$

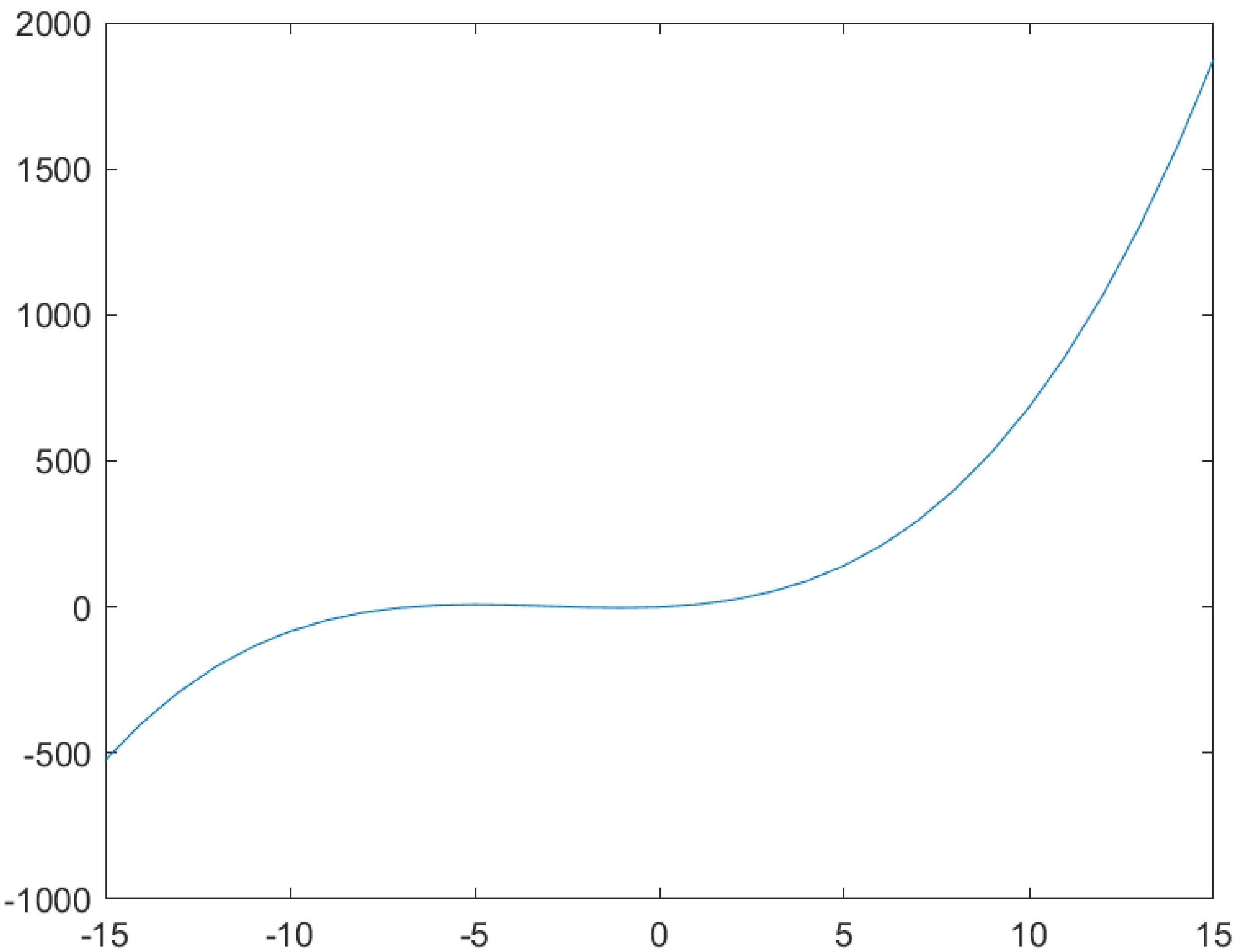
$$f(x) = \int (x^2 + 6x + 5)dx$$

$$\frac{1}{3}x^3 + \frac{6}{2}x^2 + \frac{5}{6}x + c$$

$$\frac{1}{3}x^3 + 3x^2 + 5x + c$$

2. Plot the function $\int f(x)dx$ over input *x* ranging from -15 to 15, by setting $c = 0$.

```
x=[-15:15]
c=0
y=((1/3)*x.^3) + (3*x.^2) + (5*x) + (c)
plot(x,y)
```



3. Find $\int_0^5 f(x)dx$ by hand.

$[1/3x^3 + 3x^2 + 5x]5$

$f'(1) = \frac{1}{3}x^3 + 3x^2 + 5x$

$$f(0)dx = \frac{1}{3}0^3 + 3(0)^2 + 5(0) = 0$$

$$f(5)dx = \frac{1}{3}5^3 + 3(5)^2 + 5(5) = 141.6666$$

$$\int_0^5 f(x)dx = 141.6666$$

4. With integrating points [0 1 2 3 4 5], formulate Matlab built-in function *trapz.m* to find numerical solution of $\int_0^5 f(x)dx$.

```
f=@(x) (1)*x.^2+ (6)*x+(5);
x4 = [0:0.5:5];
Q4 = trapz(x4, f(x4))
```

$$\int_0^5 f(x)dx = 141.8750$$

5. Use Matlab built-in function *integral.m* to find numerical solution of $\int_0^5 f(x)dx$.

```
f=@(x) (1)*x.^2+ (6)*x+(5);

xmin=0;
xmax=5;

Q5 = integral(f,xmin,xmax)
```

$$\int_0^5 f(x)dx. = 141.6667$$

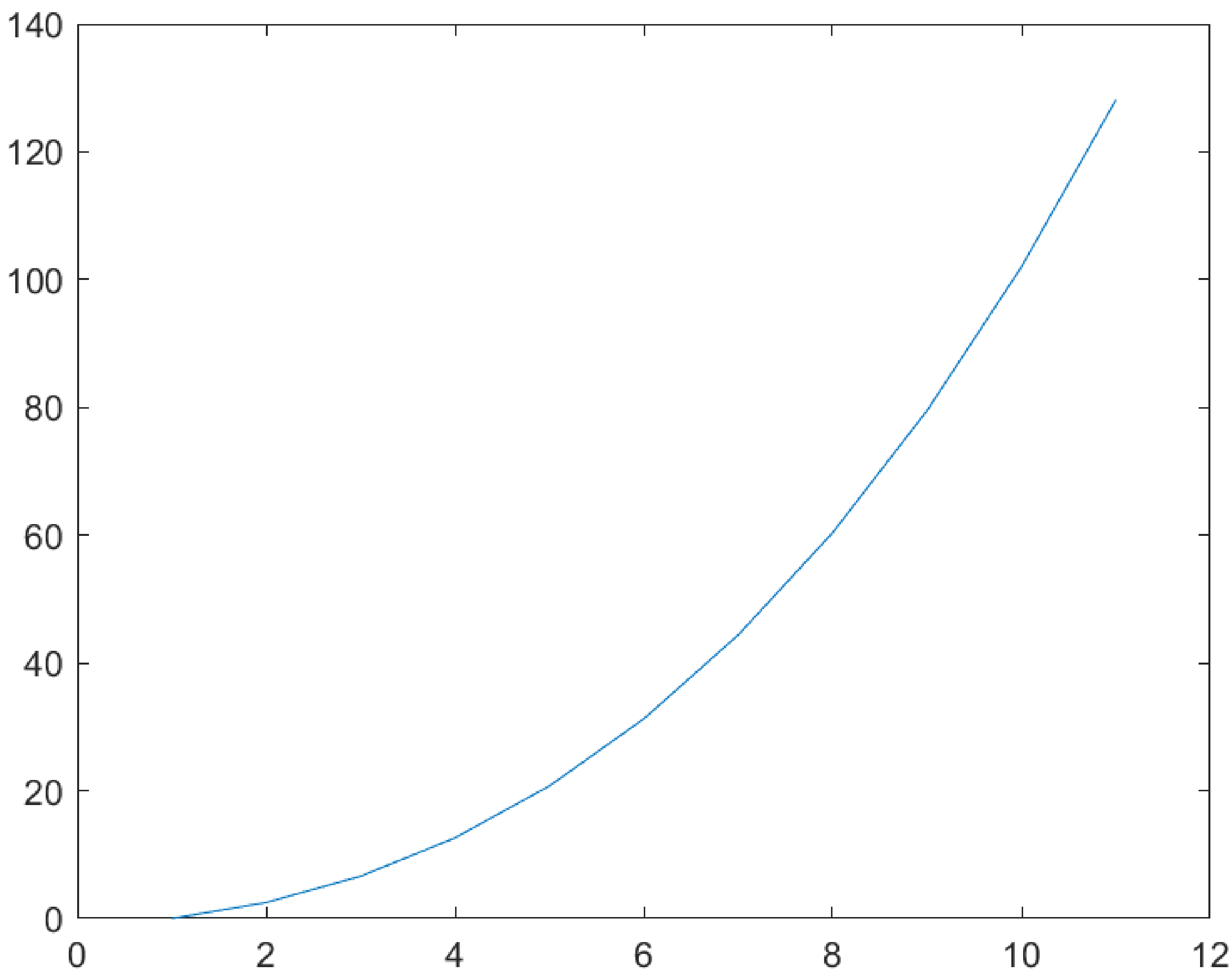
6. Write Euler's method in Matlab to solve $\frac{dy}{dx} = f(x)$, with initial condition *y(0)=0,* step size *h=0.5,* for ten steps.

```
f=@(t) (1)*t.^2+ (6)*t+(5);

h=0.5; /
t0=0;   %starting loop time
eul(1) = 0;

for i = 1:10; % ten steps
     t = t0+(i-1)*h;  % current loop time
     eul(i+1) = eul(i)+h*f(t)   % Euler algorithm
end;

plot(eul)
```



Task 3

$$dy^2/dt^2 + 0.10512 dy/dt + y = 20$$

1. Represent this robotics system in a vector differential equation form (state space model).

$$y = x_1 \qquad \frac{dy}{dt} = x_2 = x_1 \qquad \frac{d^2y}{dt^2} = x_2$$

$$\frac{d^2y}{dt^2} = x_1 - 0.10512\frac{dy}{dt} + 20 \qquad x_2 = -x_1 - 0.10512 \times x_2 + 20$$

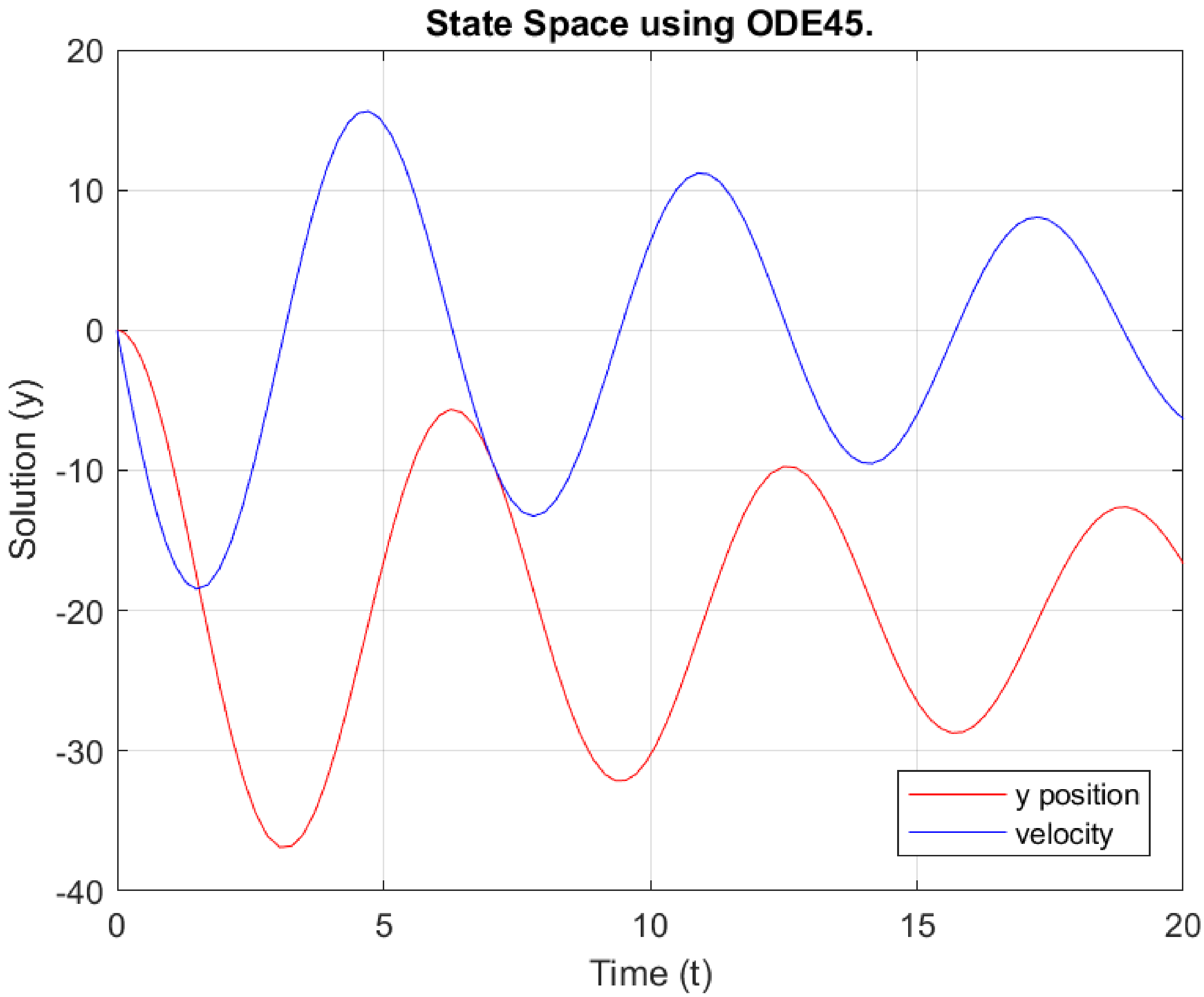
$$\begin{bmatrix} x1 \\ x2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & 0.10512 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 20 \end{bmatrix}$$

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

2. Implement a code to call Matlab built- in function *ode45.m,* with time span [0 20], initial conditions $y(0) = 0, y'(0) = 0$.

```
[t,y] = ode45(@vdp1,[0 20],[0; 0]);
plot(t,y(:,1),'-r',t,y(:,2),'-b')
title('State Space using ODE45.');
xlabel('Time(t)');
ylabel('Solution(y)');
legend('Y Position','Velocity')
grid on
function dydt = vdp1(t,y)
dydt = [y(2); (-20-y(1)-0.10512*y(2))];
end
```

3. Plot the position *y* and velocity $\frac{dy}{dt}$, and briefly discuss the simulated motion.



Task 4

$a = 1 + 11i$ $\qquad\qquad$ $b = 8 + 91i$

1. Add *a* and *b*

$$a + b =$$

$$1 + 11i + 8 + 91i \qquad = 9 + 102i$$

2. Multiply *a* and *b*

$$ab = \qquad (1 + 11i)(8 + 91i)$$

$$= 8 + 19i + 88i + 1001i^2 \qquad = 8 + 179i - 1001 \qquad = -993 + 179i$$

3. Divide *a* by *b*

$a/b =$

$$\frac{(1 + 11i)}{(8 + 91i)} \qquad = \frac{(1 + 11i)(8 - 91i)}{(8 + 91i)(8 + 91i)}$$

$$= \frac{i - 45i = 1001}{8 + 8281} = \frac{1002 - 45i}{8289}$$

4. Represent *a* and *b* in polar form, clearly state the magnitude/angle values.

$$r(\cos\theta + \sin\theta)$$

$$r_a = \sqrt{1^2 + 11^2} \qquad = \sqrt{1 + 212} \qquad = \sqrt{122}$$

$$\theta = \tan^{-1}(\frac{11}{1}) \qquad \theta = 1.48013644 \qquad \theta = 1.48 \text{ radians}$$

$$= \sqrt{122}(\cos 1.48 + i \sin 1.48)$$

$$r_a = \sqrt{8^2 + 91^2} \qquad = \sqrt{64 + 8281} \qquad = \sqrt{8345}$$

$\theta = \tan^{-1}(\frac{91}{1})$ $\theta = 89.37040139$ $\theta = \theta = 89.37 \text{ radians}$

$$= \sqrt{8345}(\cos 89.37 + i \sin 89.37)$$