UNIVERSITY OF ASIA PACIFIC

Department of Computer Science & Engineering

ASSIGNMENT

Course Title – Numerical Methods Lab

Course Code – CSE 314

Submitted by:

Shawan Das.

ID – 19101020

Section – A_1

Submitted To:

Nadeem Ahmed

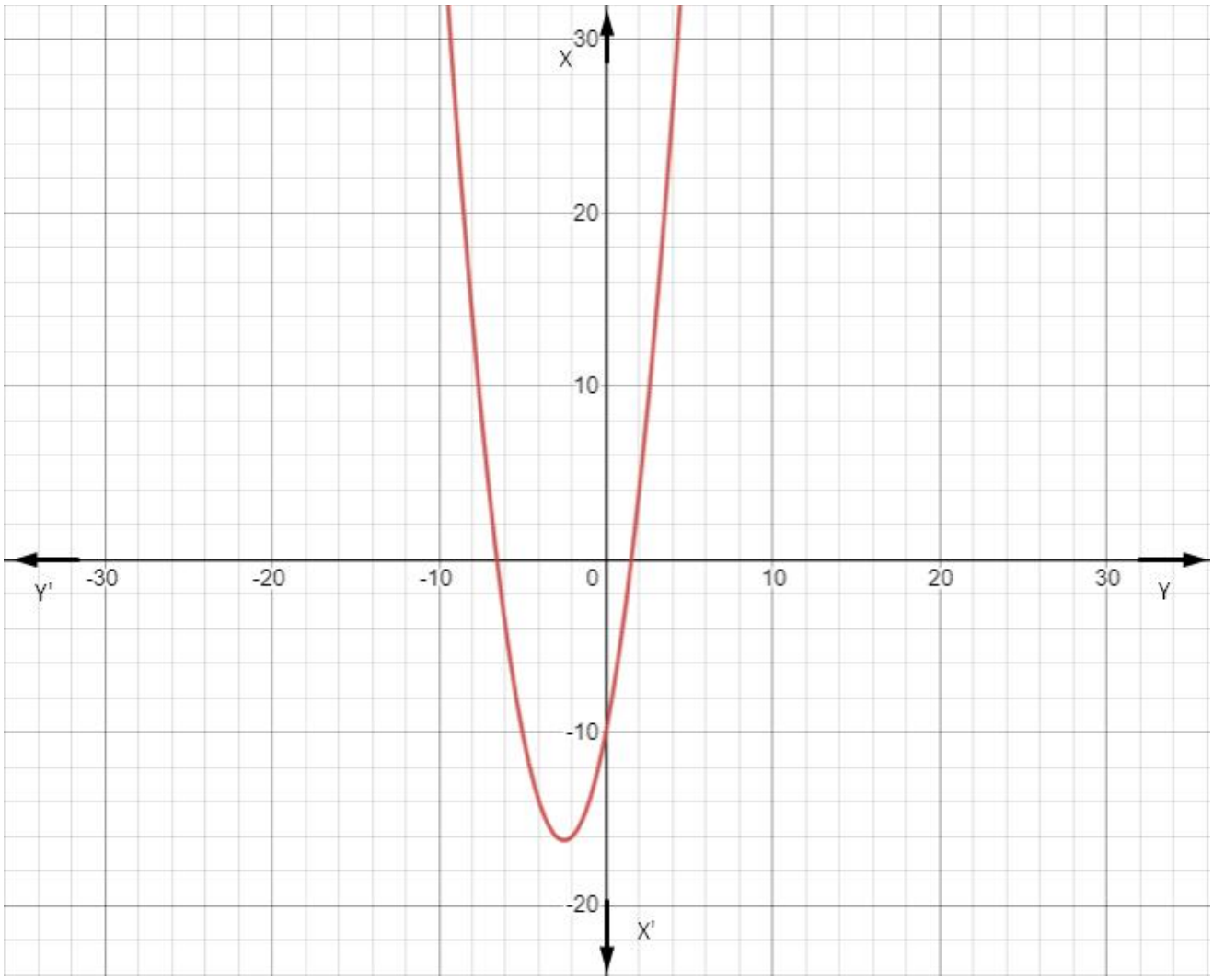
Assistant Professor,

University of Asia Pacific

Date of Submission – 17-04-2022

Bisection Method Algorithm:

Consider, $f(x) = x^2 + 5x - 10$



For the 2nd degree equation f(x) has 2 root value.

For bisection method we need 2 initial guesses.

1) Between -9 and -2.

2) Between 0 and 2.

Considering initial guess xl= -9 and xu= -2

```
clc clear
  f = @(x) x.^2 +5*x -10;
  xl= -9;
  xu= -2;

  if(f(xl)*f(xu)>0)
    disp("Root isn't in range");
  else
    while(1)
     xm=(xl+xu)/2
     if(f(xl)*f(xm)==0)
       disp("Root is in"); disp(xm);
       break;
     end
     if(f(xl)*f(xm)<0)
       xu=xm;
      end
     if(f(xl)*f(xm)>0)
       xl=xm;
     end
    endwhile
end
```

```
xm = -5.5000
xm = -7.2500
xm = -6.3750
xm = -6.8125
xm = -6.5938
xm = -6.4844
xm = -6.5391
xm = -6.5117
xm = -6.5254
xm = -6.5322
xm = -6.5288
xm = -6.5305
xm = -6.5314
xm = -6.5309
xm = -6.5312
xm = -6.5311
xm = -6.5311
xm = -6.5311
xm = -6.5311
xm = -6.5311
xm = -6.5311
xm = -6.5311
xm = -6.5311
xm = -6.5311
xm = -6.5311
xm = -6.5311
xm = -6.5311
xm = -6.5311
xm = -6.5311
xm = -6.5311
xm = -6.5311
xm = -6.5311
xm = -6.5311
xm = -6.5311
xm = -6.5311
xm = -6.5311
```

```
xm = -6.5311
xm = -6.5311
xm = -6.5311
xm = -6.5311
xm = -6.5311
xm = -6.5311
xm = -6.5311
xm = -6.5311
xm = -6.5311
xm = -6.5311
xm = -6.5311
xm = -6.5311
xm = -6.5311
xm = -6.5311
xm = -6.5311
xm = -6.5311
xm = -6.5311
xm = -6.5311
xm = -6.5311
xm = -6.5311
xm = -6.5311
xm = -6.5311
xm = -6.5311
xm = -6.5311
xm = -6.5311
xm = -6.5311
xm = -6.5311
xm = -6.5311
xm = -6.5311
xm = -6.5311
xm = -6.5311
Root is in
-6.5311
>> |
```

The first root found in -6.5311 . It needed around 16 iterations.

Now for the 2nd root consider initial guesses xl=0; xu=2;

And the root was found in 1.5331 after round 14 iteration.

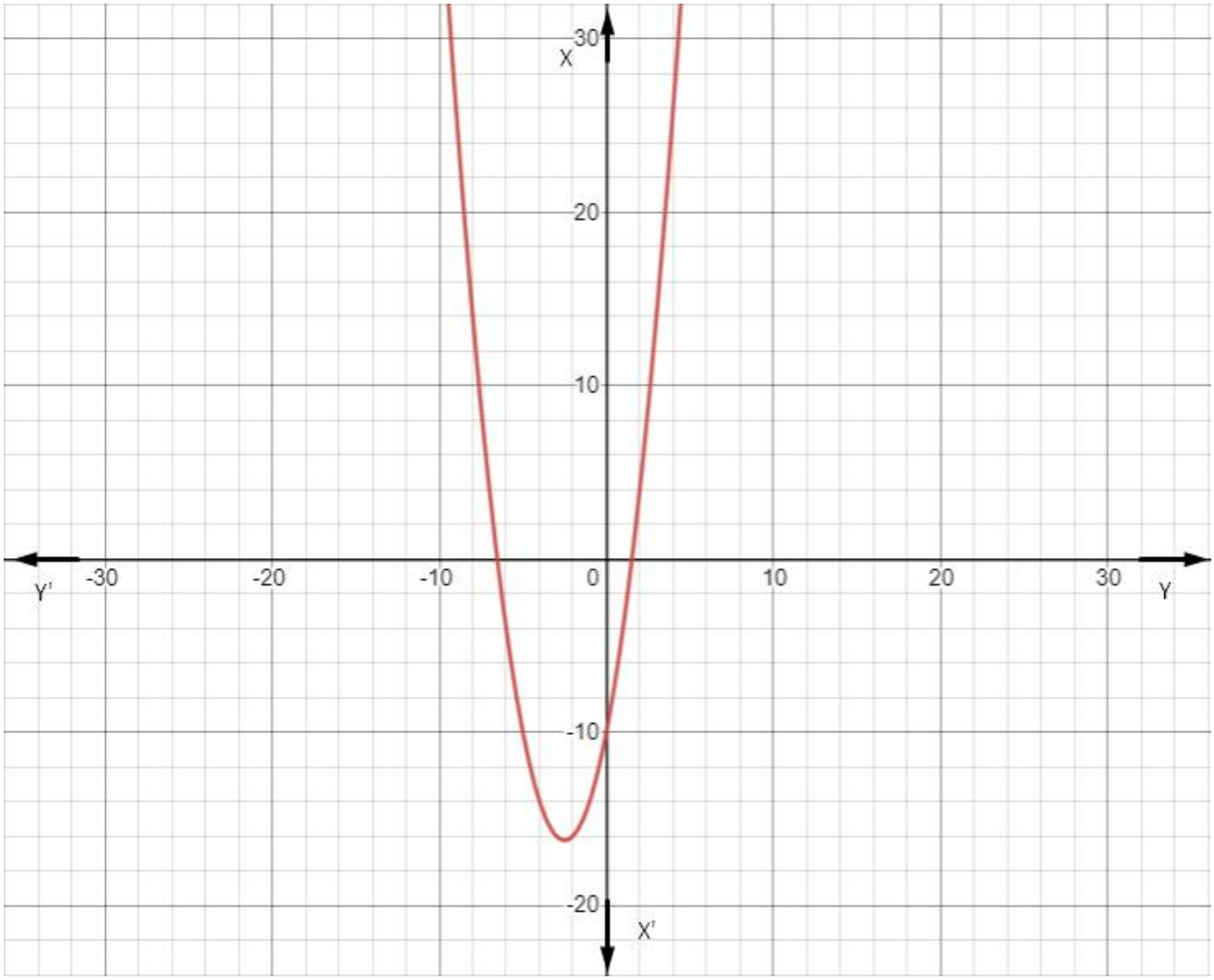
```
clc clear
  f = @(x) x.^2 +5*x -10;
  xl= 0;
  xu= 2;

  if(f(xl)*f(xu)>0)
    disp("Root isn't in range");
  else
    while(1)
      xm=(xl+xu)/2
      if(f(xl)*f(xm)==0)
        disp("Root is in"); disp(xm);
        break;
      end
      if(f(xl)*f(xm)<0)
        xu=xm;
      end
      if(f(xl)*f(xm)>0)
        xl=xm;
      end
    endwhile
end
```

```
xm = 1
xm = 1.5000
xm = 1.7500
xm = 1.6250
xm = 1.5625
xm = 1.5312
xm = 1.5156
xm = 1.5234
xm = 1.5273
xm = 1.5293
xm = 1.5303
xm = 1.5308
xm = 1.5310
xm = 1.5311
xm = 1.5312
xm = 1.5312
xm = 1.5311
xm = 1.5311
xm = 1.5311
xm = 1.5311
xm = 1.5311
xm = 1.5311
xm = 1.5311
xm = 1.5311
xm = 1.5311
xm = 1.5311
xm = 1.5311
xm = 1.5311
xm = 1.5311
xm = 1.5311
xm = 1.5311
xm = 1.5311
xm = 1.5311
xm = 1.5311
xm = 1.5311
xm = 1.5311
```

Root points

Newton Raphson Method Algorithm:

Consider, $f(x) = x^2 + 5x - 10$



In Newton Raphson method, we need only one initial guess close to the root value.
Let's consider the initial guess $x_0 = -8$

Implementing Newton Raphson method and 5 iteration, we get

The root is in -6.5311

```
x0 = -6.7273
x0 = -6.5357
x0 = -6.5311
x0 = -6.5311
x0 = -6.5311
Root is in
-6.5311
>>
```

```
clc
clear
pkg load symbolic
  f = @(x) x.^2 +5*x -10;
  f1 = function_handle (diff(formula (f(sym("x")))));

  iteration=4;
  step=0;
  x0=-8;
  while(step<=iteration)

    if(f1(x0)==0)
      disp("Division By zero");
    end
    x0= x0 - (f(x0)/f1(x0))
    step+=1;
end
    %end
  disp("Root is in "); disp(x0);
```

Initial Guess, x_0 = 4 - we get

Its takes only 4 iterations, to find the root value.

Which is 1.5311 .

```
x0 = 2
x0 = 1.5556
x0 = 1.5312
x0 = 1.5311
x0 = 1.5311
Root is in
1.5311
>>
```

Comparative Analysis:

Bisection Method	Newton Raphson Method
Took much more iteration to find the root in f(x) each time	Performed 5 iteration but found the closer to root value in 4 iteration.
Converges slow	Converge faster than “Bisection Method”.
If we take initial guesses closer to	Any guesses closer to the root point converges fast. And needed only

root value, it converges more faster

```
x0 = 1.5556
x0 = 1.5312
x0 = 1.5311
x0 = 1.5311
Root is in
1.5311
>>
```

one initial guess to perform this method which is much better than the Bisection method because the Bisection method needed 2 initial guesses.

Ans of 2:

Provided code,

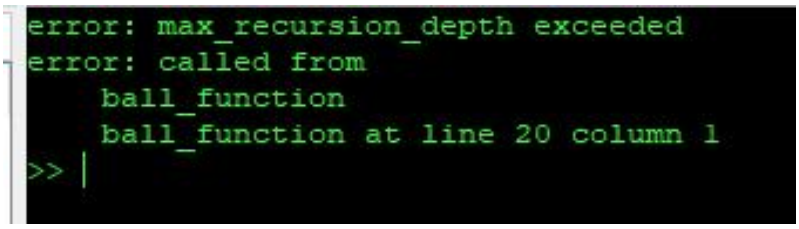
```
clc
clear
pkg load symbolic
function ball_function()
  % This is the main  program
  time = 0.6;
  vertical_position = y(time);
  fprintf("%f \n",vertical_position)
  time =0.9;
  vertical_position = y(time);
  fprintf("%f \n",vertical_position)
end
% The function 'y' is a _local_function in this file

function result = y(t)
  g=9.81;      % Acceleration of  gravity
  v0 =5;       % Initial velocity
  result = v0*t - 0.5*g*t^2;
end
%--
```

```
>> clear
>> ball_function

1.234200
0.526950
>>
```

A) Remove “function” from “function ball_function()” which is the wrong way to call a function.

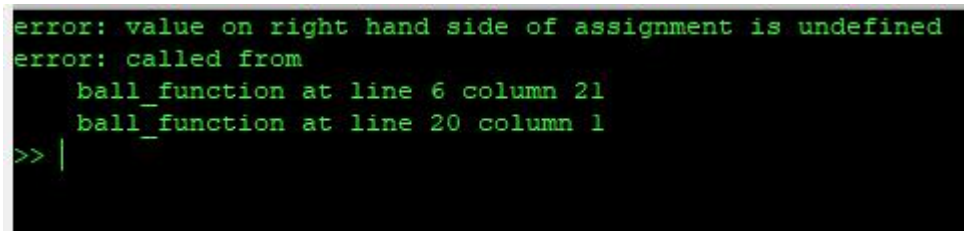
```
error: max_recursion_depth exceeded
error: called from
    ball_function
    ball_function at line 3 column 2
>>
```

B) Change “function ball_function()” to “function ball_func()”.



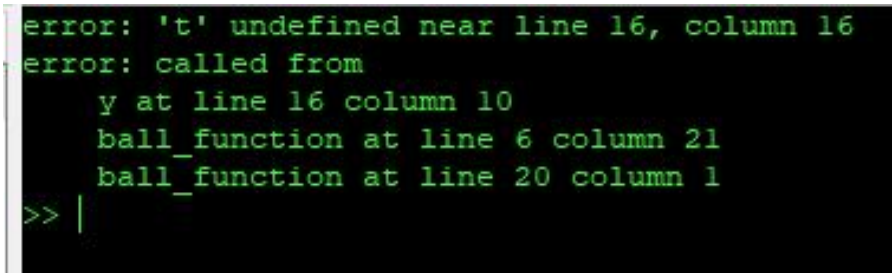
It will give an error in line 20 where I have called the ball_function(). It occurs because there is no function called ball_function().

C) Change “function result = y(t)” to “function y(t)”.



There will be an error in line-6. because here vertical_position = y(time) is expecting a returned value from function y(t). so there must be a return type for function y(t).

D) Change “function result = y(t)” to “function result = y()”. There will be 2 errors in line-6 and line-16. Line-6 will give error because function result=y() have no such formal parameter to pass.



But in line-6 we are calling this function with a parameter y(time). And another error occurs in line-16 because the variable t wasn’t declared in the function.

E) There will be an error in line-6. because function result=y(t) is expecting a parameter, but when we are calling the function y(t), we didn’t pass any parameter. For that it will give an error.