INDIAN INSTITUTE OF TECHNOLOGY GANDHINAGAR

MA 202: MATHEMATICS - IV
Semester–II, Academic Year 2022-23

Tutorial Set -2
Question - 4

By

Kush Patel
[20110131]

➔ In question 4, we are asked to observe the rate of convergence of the given function. In the question, it is given that we can improve the rate of convergence by modifying Newton's method. This method is also known as Modified Newton's Method. Also we are asked to show that function f has a zero of multiplicity 2 at x=0.

We know that

$$f(x) = e^x - x - 1, \qquad f'(x) = e^x - 1, \qquad f''(x) = e^x$$

Thus,

$$f(0) = e^0 - 0 - 1 = 0, \; f'(0) = e^0 - 1 = 0, \quad f''(0) = e^0 = 1$$

Thus, the root p=0 is a zero of multiplicity 2.

➔ Now, the question is asking about the convergence through Newton's method. I have tested it through Newton's method by taking initial guess xi = 1. Approximation after the (i+1)th iteration, Newton's method, as we know, is given by:

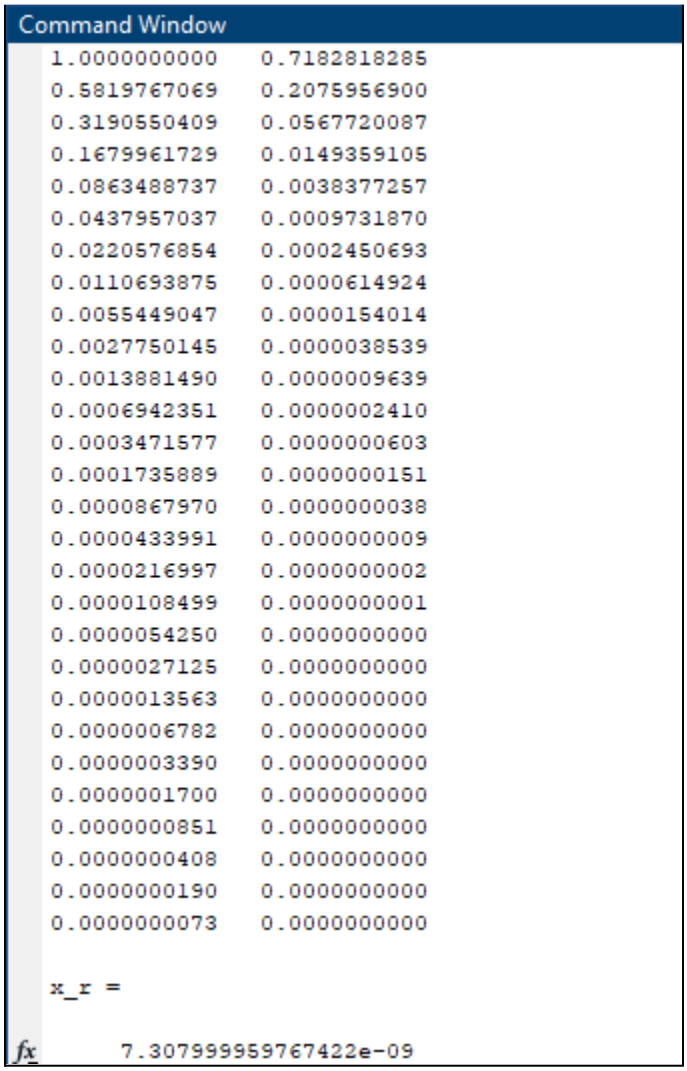
$$x(i+1) = x(i) - \frac{f(x)}{f'(x)}$$

```
% Newton's Method
% Create a fuunction to call the vale of x0 and return the vale of x_r
function [x_r] = T24_20110131(x0)
% to print the value of x_r up to long decimal digits
format("long")
% Initialize the vales of tolerance and error
tol = 0.00001;
err = 100;
% Run the while loop to run the iterations
while err > tol
    [f , df] = Function_3(x0);
    % Print all the values of x(i) and f(x(i))
    fprintf('%8.10f   %8.10f\n', [x0, Function_3(x0)])
    % Check wheather the deriavtive of the function is zero or not
    % otherwise it will retuen infinite value
    if df==0
        x_r = "Please change the initial guess";
        break;
    else
        x_r = x0 - (f/df);
        err = abs((x_r - x0)/(x_r));
        x0  = x_r;
    end
end
% Create the function to find the given function value and it's derivative
function  [val , dval] = Function_3(x)
val = exp(x) - x - 1;
dval = exp(x) - 1;
end
end
```

➔ I have printed all the values of xi and corresponding to that function's value so that at every iteration we can get to know the value of the function and we can observe at which value of xi the function becomes zero. It means function converges. By this method, we can manually count the number of iterations required for a function to converge.

➔ All iterations and its values are shown below. We can easily see the values of xi and f(xi) for each iteration. After all these iterations, I also got the root which is

extremely small (10 power -9)so we can assume it is zero and zero is the actual root. It means, we got the correct output.



➔ By observing above values, we can say that the convergence is much smaller than the quadratic as we expect from Newton. Now, we are asked to improve the rate of convergence by using Modified Newton's Method. Approximation after the (i+1)th iteration, Modified Newton's method, as we know, is given by:

$$x(i+1) = x(i) - \frac{f(x(i))*f(x(i))}{f'(x(i))^2 - f(x(i))*f''(x(i))}$$

```
function [x_r] = T24_20110131(x0)
% to print the value of x_r up to long decimal digits
format("long")
% Initialize the vales of tolerance and error
tol = 0.00001;
err = 100;
% Run the while loop to run the iterations
while err > tol
    [f , df, ddf] = Function_3(x0);
    % Print all the values of x(i) and f(x(i))
    fprintf('%8.10f   %8.10f\n', [x0, Function_3(x0)])
    % Check wheather the deriavtive of the function is zero or not
    % otherwise it will retuen infinite value
    if df==0
        x_r = "Please change the initial guess";
        break;
    else
        x_r = x0 - (f*df)/((df^2)- (f*ddf));
        err = abs((x_r - x0)/(x_r));
        x0  = x_r;
    end
end
% Create the function to find the given function value and it's derivative
function  [val , dval, ddval] = Function_3(x)
val = exp(x) - x - 1;
dval = exp(x) - 1;
ddval = exp(x);
end
```

➔ Here also I have printed all the values of xi and corresponding to that function's value so that at every iteration we can get to know the value of the function and we can observe at which value of xi the function becomes zero. It means function converges. By this method, we can manually count the number of iterations required for a function to converge.

```
>> [x_r] = T24_20110131(1)
1.0000000000   0.7182818285
-0.2342106136   0.0254057755
-0.0084582799   0.0000356706
-0.0000118902   0.0000000001
-0.0000000000   0.0000000000

x_r =

    -4.226406842332091e-11
```

➔ Silimilary, All iterations and its values are shown above. We can easily see the values of xi and f(xi) for each iteration. After all these iterations, I also got the root which is extremely small (10 power -9)so we can assume it is zero and zero is the actual root. It means, we got the correct output.

➔ We can also observe that here we can get the value of the function to be equal to zero after just 3 iterations; this means the rate of convergence improves. Here convergence is much faster than the previous one.The quadratic convergence is recovered. So, it proves that Modified Newton's Method improves the rate of convergence.

Note :- To see the results of newton's method, please comment out the rest of the code and same for modified newton's method also.