The Newton – Raphson Method

Ayush Sharma

[150123046]

ayush.2015@iitg.ac.in

Instructor: Arabin Kumar Dey

Table of Contents

Abstract

The Newton-Raphson method is an iterative process for solving the root of equation $f(x) = 0$. According to the method, starting with an initial guess of x_0, apply the iterative formula $x_{n+1} = x_n + \frac{f(x_n)}{f'(x_n)}$ where f denotes the derivative of the function. The iteration stops until you arrive at an acceptable limit $|x_{n+1} - x_n| < \epsilon$, where ϵ is some pre-specified tolerance value.

Problem Statement

Write a program to approximate the root of equation $3x^2 - e^x = 0$, to within a tolerance of 10^{-5}. Give the steps in your code and the result of executing your code. Give an explanation why your answer is reasonable.

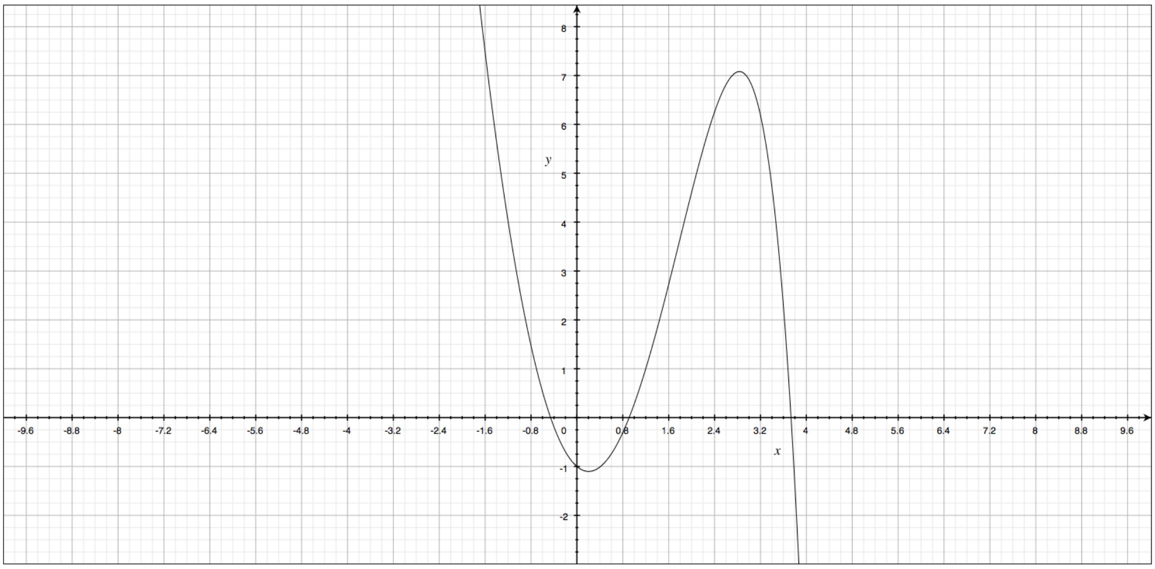
Hint: It may help to graph the function to get a decent initial estimate.

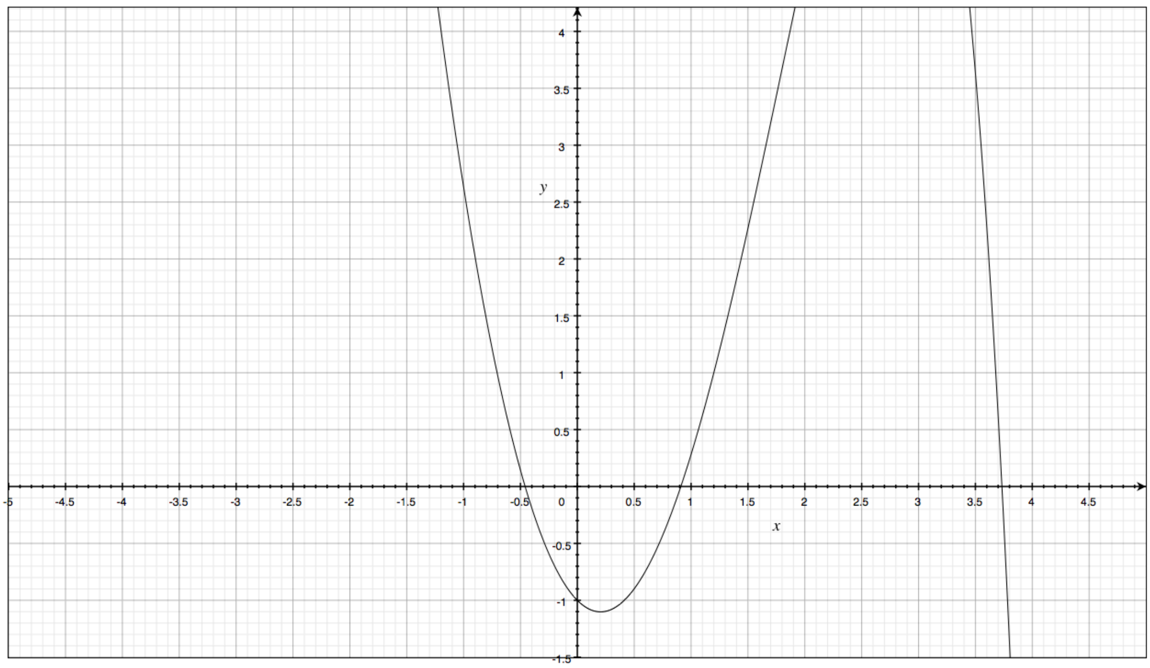
Theory

Take the given equation to be $f(x) = 0$, i.e. $f(x) = 3x^2 - e^x$. Then, its derivative is calculated to be $f'(x) = 6x - e^x$.

The roots of $f(x)$, are approximately (-0.45), (0.9), and (3.75). And, the local extrema of $f(x)$, are approximately (0.2) and (2.8), as can be observed easily from the underlying graphs.

Graph





Program

Code in C

```
#include <stdio.h>
#include <math.h>
long double f(long double x){
	//Given function f(x).
	long double f = ((3*x*x) - expl(x));
	return f;
}
long double f1(long double x){
	//Function f'(x) i.e. differentiation of given function f(x).
	long double f1 = ((6*x) - expl(x));
	return f1;
}
int main(){
	long double x = 0, x1 = 0, temp = 0;
	//Taking input the value of initial guess x0.
	printf ("Enter the value of initial guess x0, to approximate the root of the given equation
'3*x^2 - e^x = 0' by Newton-Raphson method. ");
	scanf ("%Lf",&x);
	x1 = x;
	//Iterations of Newton-Raphson method.
	do
	{
		temp = x1;
		x1 = (temp - (f(temp)/f1(temp)));
	} while (fabsl(temp - x1) > 0.00001);
	//Output of the calculated root.
	printf("Root of the given equation '3*x^2 - e^x = 0' as approximated by Newton-Raphson
method, using the entered x0 = %Lg, is x = %Lf.\n",x,x1);
	return 0;
}
```

Reasonability of the program

In the program, the variables are declared as ‘long doubles’ to increase the efficacy and accuracy.

Further, the program takes the value of initial guess x_0 as input and approximates the root of the equation. The iterations of Newton-Raphson method are defined to stop once the difference in consecutive values of x decreases the given tolerance value, i.e. 10^{-5}.

Hence, the result obtained is said to be reasonable or reasonably close to the real value of root, as the value 10^{-5} or 0.00001 is sufficiently small to accept as error in calculating the root of an equation involving exponential terms, using a computer.

Result

Taking the values of initial guess x_0 as approximated using graphs in the theory section, the value of roots are approximated using program made on the Newton-Raphson method.

- For x_0 = -0.45, the root comes out to be (-0.458962).
- For x_0 = 0.9, the root comes out to be (0.910008).
- For x_0 = 3.75, the root comes out to be (3.733079).

Further, on taking the values of initial guess not equal (or say close to the roots), the values of roots come out to be:

- For x_0 = -1, the root comes out to be (-0.458962).
- For x_0 = 0, the root comes out to be (-0.458962).
- For x_0 = 0.2, the root comes out to be (-0.458962).
- For x_0 = 0.21, the root comes out to be (3.733079).
- For x_0 = 0.25, the root comes out to be (3.733079).
- For x_0 = 0.29, the root comes out to be (3.733079).
- For x_0 = 0.295, the root comes out to be (-0.458962).
- For x_0 = 0.3, the root comes out to be (-0.458962).
- For x_0 = 0.305, the root comes out to be (-0.458962)
- For x_0 = 0.31, the root comes out to be (3.733079).
- For x_0 = 0.315, the root comes out to be (0.910008).
- For x_0 = 0.35, the root comes out to be (0.910008).
- For x_0 = 0.5, the root comes out to be (0.910008).
- For x_0 = 2, the root comes out to be (0.910008).
- For x_0 = 2.4, the root comes out to be (0.910008).
- For x_0 = 2.5, the root comes out to be (-0.458962).
- For x_0 = 2.8, the root comes out to be (-0.458962).
- For x_0 = 2.85, the root comes out to be (3.733079).
- For x_0 = 3, the root comes out to be (3.733079).
- For x_0 = 5, the root comes out to be (3.733079).

Interpretation of the result

As expected, the program performs efficiently to approximate the roots of the equation $3x^2 - e^x = 0$, to a certain tolerance limit.

But on an interesting note, the values of roots approximated at the values of initial guess close to or a little around the points of extrema, highly vary (or say keep shuffling/interchanging), although within the acceptable bounds of the three distinct real roots. This is because the slope of the function is close to 0 (i.e. low magnitude) and changes sign on either side of extremum.