[4MM013]
Computational Mathematics

Student Id: 2431342
Student Name: Swoyam Pokharel
Section: L4CG27
Tutor: Sulav Shrestha
Submitted On: 30-8-2024

Task 1:

In trying to find the roots of an equation, the traditional approach of analytical methods can get complicated. For such scenarios we have two prominent methods for finding the roots of an equation, namely, the **Bisection Method** and the **Newton-Raphson Method.** These methods are foundational for finding approximate solutions to an equation where the traditional approach of direct analytical solutions are impractical.

The Bisection Method:

The bisection method is a simple, straightforward method for approximating the solution of an equation. Given a function f(x), and an interval [a,b] where f(a) . f(b) < 0, the method involves repeatedly dividing the interval between [a,b] into subintervals, selecting the one that contains the root, and narrowing it down until the subinterval is sufficiently small enough. This method is based on the intermediate value theorem that states that if a function is continuous on a closed interval [a,b], and takes on opposite signs at a and b, there exists a point c between a and b where f(c) = 0, i.e where the solution lies. It narrows down by calculating the midpoint m between a and b, and if f(m) has a different sign from f(a) then we know that the root lies between [a,m] or if f(m) has a different sign from f(b) then we know that the root lies between [m,f].

Function:	Interval [a,b]	Iterations	Bisection Method:	SciPy Method:
$y = f(x) = x^2 - x - 1$	[0,1] , [-2,0]	100	1.61803436,-0.61803436	1.61803399, -0.61803399
$y = f(x) = x^3 - x^2 - 2x + 1$	[-3, -1] , [-0.5, 1] , [1, 3]	100	-1.24697971, 0.44504166 ,1.80193758	-1.24697960, 0.44504187 , 1.80193774

The Newton-Raphson Method:

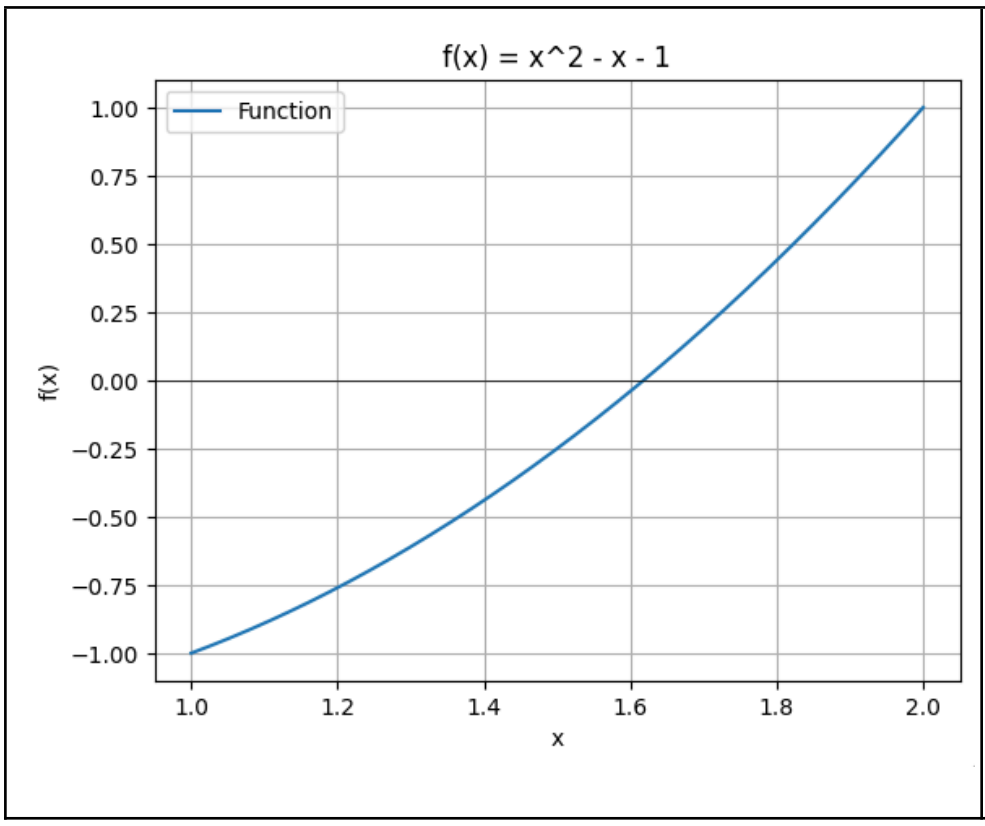
The Newton-Raphson Method, unlike the Bisection Method which narrows down on an interval, uses the function's derivative to quickly converge to a root. The method involves starting off with a guess x_0 for the root of a function f(x), which goes without saying that if the guess is reasonably close to the root, the method will converge faster. Using the formula:

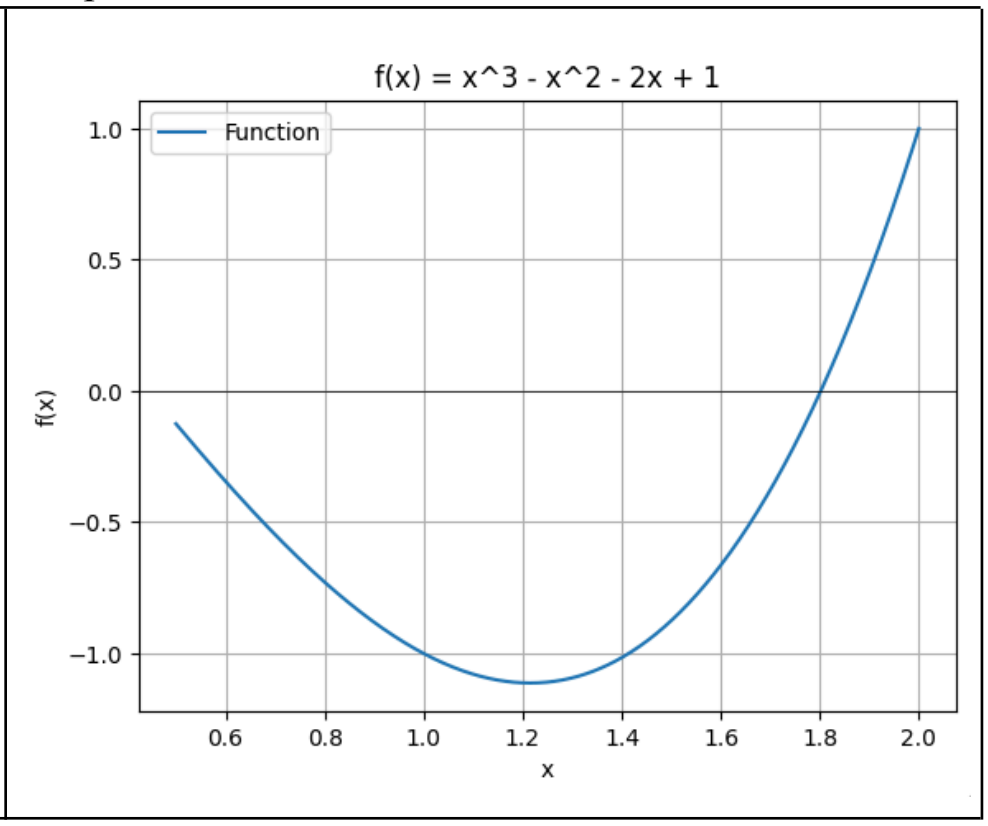
$$X_{n+1} = X_n - \frac{f(x_n)}{f'(x_n)}$$

We calculate the approximation, replacing X_n with X_{n+1} and keep on repeating until the difference between successive approximations < tolerance.

Function:	Initial guess (x0)	Iterations	Newton-Raphson Method:	SciPy Method:
$y = f(x) = x^2 - x - 1$	[1]	100	-0.61803399, 1.61803399	-0.61803399, 1.61803399
$y = f(x) = x^3 - x^2 - 2x + 1$	[0]	100	-1.24697960, 0.44504182, 1.80193774	-1.24697960, 0.44504187, 1.80193774

Function Graphs:





Task 2:

Midpoint Approximation:

The midpoint approximation is a method for approximating the definite integral of a function over a given interval [a,b] by dividing the interval into smaller sub intervals and using the values at the midpoints of these intervals f(m) to estimate the area under the curve. This method splits the interval [a,b] into n equally spaced subintervals, where each interval will have a width of:

$$w = \frac{b-a}{n}$$

Meaning each point dividing the intervals are a, a+2, a+2w, ..., b, and for each sub interval it finds the midpoint m, and calculates the value of f(m). Finally, this method estimates the integral by summing up the areas of rectangles with height f(m) and width w.

$$w \cdot \sum_{i=1}^{n} f(m_i)$$

$y = f(x) = \frac{x}{x^2+1}$ and Limits $[0, 5]$

N	Midpoint Approximation	Analytical	Absolute Error
10	1.640378	1.629048	0.01132983
30	1.630252	1.629048	0.00120420
50	1.629480	1.629048	0.00043219
100	1.629156	1.629048	0.00010791
500	1.629053	1.629048	0.00000431

$y = f(x) = e^x$ and Limits $[0, 5]$

N	Midpoint Approximation	Analytical	Absolute Error
10	145.888729	147.413159	1.52443031
30	147.242680	147.413159	0.17047895
50	147.351755	147.413159	0.06140424
100	147.397805	147.413159	0.01535442
500	147.412545	147.413159	0.00061422

Trapezoidal Rule:

The trapezoidal rule is a numerical method to estimate the integral of a function f(x) on the interval [a,b] similar to the Midpoint Approximation method, but unlike the Midpoint Approximation method, it works by estimating the area under the curve as a series of trapezoids instead of rectangles. This method works by splitting the interval [a,b] into n equally spaced intervals, just like the Midpoint Approximation method.

$$w = \frac{b-a}{n}$$

So the points dividing the integrals will be a (x_0), a+w (x_1), a+2w (x_2), …, (x_n)b, and for each interval, approximate the area under the curve for that interval by using a trapezoid,

$$Area = \frac{w}{2}\left[f(x_i) + f(x_{i+1})\right]$$

and the total integral is approximating the sum of all the areas of the trapezoids.

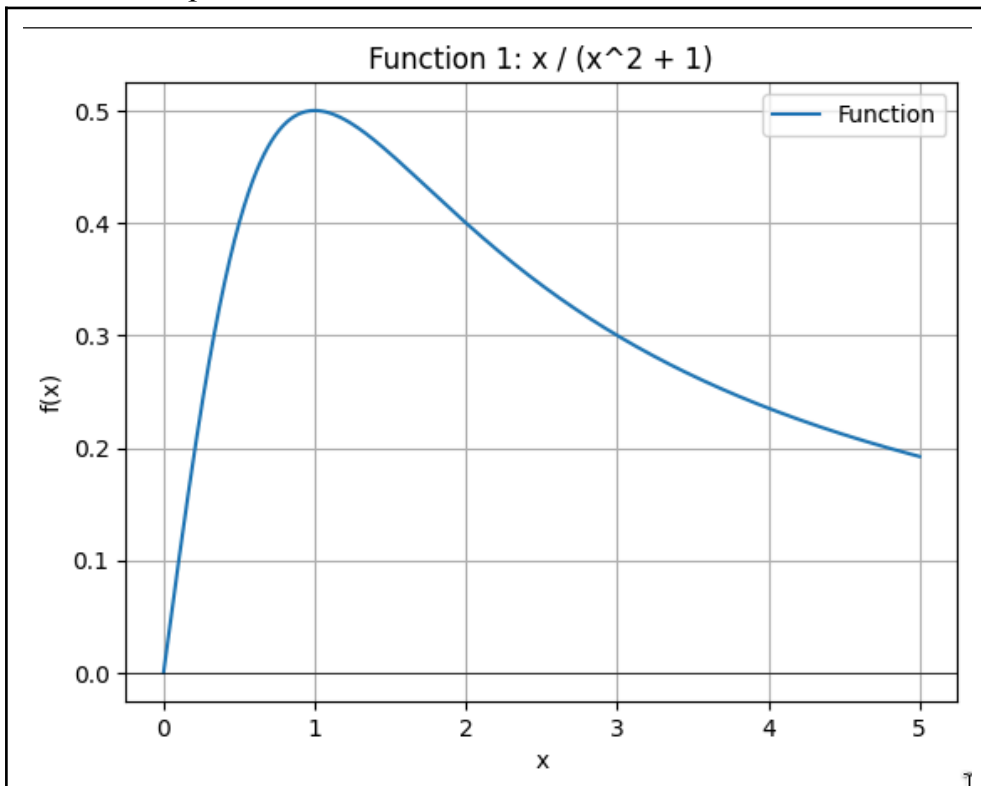
$y = f(x) = \frac{x}{x^2+1}$ and Limits $[0, 5]$

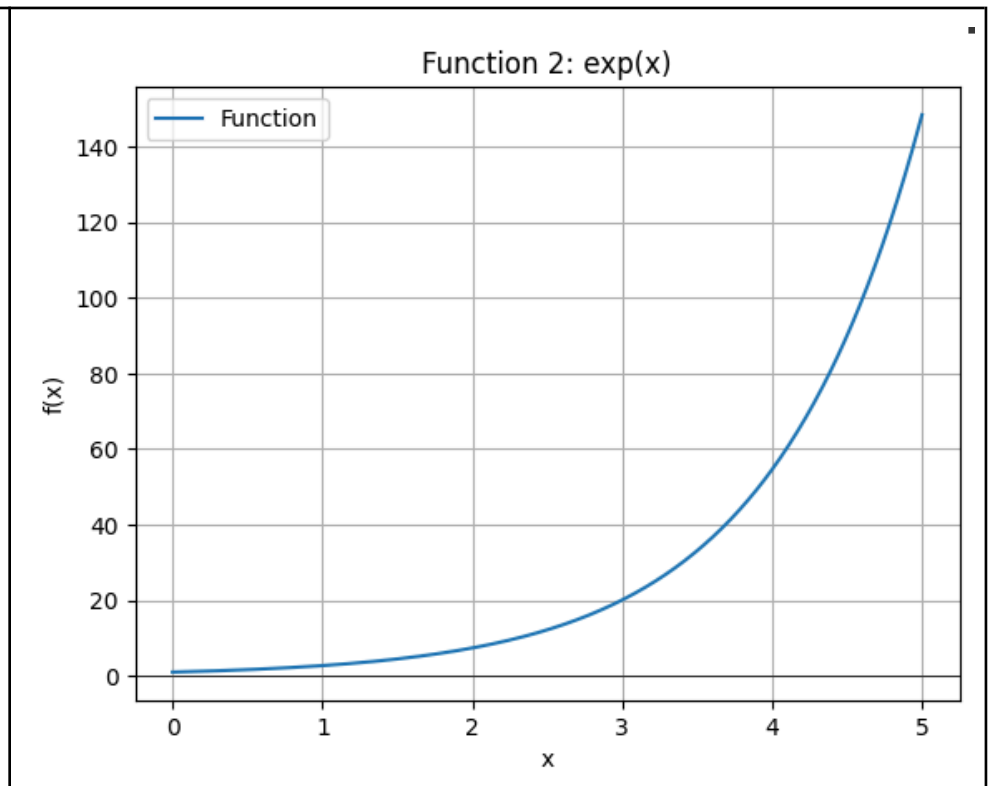
N	Trapezoidal Rule	Analytical	Absolute Error
10	1.606865	1.629048	0.02218344
30	1.626645	1.629048	0.00240351
50	1.628185	1.629048	0.00086376
100	1.628832	1.629048	0.00021578
500	1.629040	1.629048	0.00000863

$y = f(x) = e^x$ and Limits $[0, 5]$

N	Trapezoidal Rule	Analytical	Absolute Error
10	150.471546	147.413159	3.05838690
30	147.754235	147.413159	0.34107629
50	147.535983	147.413159	0.12282383
100	147.443869	147.413159	0.03070980
500	147.414388	147.413159	0.00122844

Function Graphs:





Conclusion:

Comparison between The Bisection Method and the Newton-Raphson Method:

Between the two methods, the Bisection Method is easier to implement, and is very robust. Furthermore, it also guarantees convergence provided that the function is continuous and changes sign over the interval [a,b], but it is slower to converge as it converges linearly, and it needs an interval where the function sign changes.The Newton-Raphson method however, is more complex as it requires the derivative of a function, but is more faster converging as it converges quadratically. Moreover, the Newton-Raphson method can fail to converge totally, if the initial guess is not close to the root or if the function has inflection points where the derivative is zero.

So, for simplicity and reliability, especially dealing with unknown or complex functions, the Bisection Method seems to be a better choice as it guarantees convergence and does not require the derivative. For faster convergence and when a good initial guess is available along with the derivative

of the function, the Newton-Raphson Method is more efficient. A better alternative would be the Secant method which is more efficient compared to the Bisection Method, and does not require the derivative like the Newton-Raphson method, but uses the two previous approximations to estimate the derivative.

Comparison between The Midpoint Approximation and The Trapezoidal Rule:

Between the two methods, Midpoint Approximation is simpler to implement as it only requires the function values at the midpoint of each subinterval, and is reasonably accurate for linear functions. However, more intervals are needed for reasonable accuracy, unlike the trapezoidal rule. The trapezoidal rule though is more complex, it typically results in more accurate results in the same amount of intervals compared to the Midpoint Approximation method. Furthermore, it provides better accuracy with curved functions too, but it may still be inaccurate for functions with sudden changes.

So, the Midpoint Approximation seems to be a better choice when looking for simplicity and reasonable accuracy for linear functions, and the Trapezoidal Rule seems to be better with curved functions and accuracy in fewer intervals. However, the Simpson's Rule is generally considered to be superior to both the above methods in capturing curvature especially for smoother functions, while providing a better approximation in the same amount of intervals compared to both the methods.