Equations Used With Numerical Methods Algorithms

Mohamed Emary Mohamed Abdelfattah Abdelfattah Zakaria

Shrouk Elsayed Dalia Abdallah Sara Reda

November 2, 2023

We have used the same 10 problems with each method and run each method (Bisection, False Position, and Hybrid) 500 times for each problem and then we have calculated the average time. We have also calculated the number of iterations each method have taken for each problem.

We have also used the same accuracy for each problem which is 10^{-10}

These are the problems that we have used for each method:

Table 1: Problem Set

No	Equation	Equation Code	Interval
$P1$	$f(x) = x^3 + 4x^2 - 10 = 0$	`x**3 + 4*x**2 - 10`	`[0, 4]`
$P2$	$f(x) = x^2 - 4$	`x**2 - 4`	`[0, 4]`
$P3$	$f(x) = e^x - 2$	`sympy.exp(x) - 2`	`[0, 2]`
$P4$	$f(x) = \sin(x)$	`sympy.sin(x)`	`[2, 6]`
$P5$	$f(x) = x^3 - 6x^2 + 11x - 6$	`x**3 - 6*x**2 + 11*x - 6`	`[1, 2.5]`
$P6$	$f(x) = x^2 + 3x + 2$	`x**2 + 3*x + 2`	`[-2.5, -1.5]`
$P7$	$f(x) = \cos(x) - x$	`sympy.cos(x) - x`	`[0, 1]`
$P8$	$f(x) = 2^x - 8$	`2**x - 8`	`[2,4]`
$P9$	$f(x) = \tan(x)$	`sympy.tan(x)`	`[-1, 1]`
$P10$	$f(x) = x^4 - 8x^3 + 18x^2 - 8x + 1$	`x**4 - 8*x**3 + 18*x**2 - 8*x + 1`	`[2, 5]`

Table 2: Bisection Table

Problem	Iter	Avg CPU Time	Root
$P1$			
$P2$			
$P3$			
$P4$			

Problem	Iter	Avg CPU Time	Root
$P5$			
$P6$			
$P7$			
$P8$			
$P9$			
$P10$			

Table 3: False Position Table

Problem	Iter	Avg CPU Time	Root
$P1$			
$P2$			
$P3$			
$P4$			
$P5$			
$P6$			
$P7$			
$P8$			
$P9$			
$P10$			

Table 4: Hybrid Method Table

Problem	Iter	Avg CPU Time	Root
$P1$			
$P2$			
$P3$			
$P4$			
$P5$			
$P6$			
$P7$			
$P8$			
$P9$			
$P10$			