axioms

Article

Numerical Analysis of New Hybrid Algorithms for Solving Nonlinear Equations

Miguel Vivas-Cortez [1], Naseem Zulfiqar Ali [2], Awais Gul Khan [2] and Muhammad Uzair Awan [2,*]

1 Escuela de Ciencias Fsicas y Matemticas, Facultad de Ciencias Exactas y Naturales, Pontificia Universidad Catlica del Ecuador, Av. 12 de Octubre 1076, Apartado, Quito 17-01-2184, Ecuador; mjvivas@puce.edu.ec
2 Department of Mathematics, Government College University, Allama Iqbal Road, Faisalabad 38000, Pakistan; naseemansari64jb@gmail.com (N.Z.A.); awaisgulkhan@gmail.com (A.G.K.)
* Correspondence: awan.uzair@gmail.com

Abstract: In this paper, we propose two new hybrid methods for solving nonlinear equations, utilizing the advantages of classical methods (bisection, trisection, and modified false position), i.e., bisection-modified false position (Bi-MFP) and trisection-modified false position (Tri-MFP). We implemented the proposed algorithms for several benchmark problems. We discuss the graphical analysis of these problems with respect to the number of iterations and the average CPU time.

Keywords: modified false position method; classical methods; numerical calculations; hybrid algorithms

MSC: 58C30; 65H05

check for updates

Citation: Vivas-Cortez, M.; Ali, N.Z.; Khan, A.G.; Awan, M.U. Numerical Analysis of New Hybrid Algorithms for Solving Nonlinear Equations. *Axioms* **2023**, *12*, 684. https://doi.org/10.3390/axioms12070684

Academic Editors: Victor Orlov and Michal Feckan

Received: 16 May 2023
Revised: 12 June 2023
Accepted: 19 June 2023
Published: 12 July 2023

1. Introduction

It is widely acknowledged that one of the most challenging problems in the mathematical sciences, particularly in numerical analysis, is solving the nonlinear equation $\Psi(\xi) = 0$. In multiple improvements, numerous researchers have suggested, examined, and developed a variety of numerical approaches, employing various strategies, including Taylor's series, modified homotopy perturbation methods, decomposition methods, variational iterative methods, and quadrature formulae. There is a wealth of literature available that highlight various approaches to solving nonlinear equations; for example, see [1–15]. There are various ways to determine the roots of nonlinear equations, which include bracketing methods (bisection, trisection, false position, and modified false position) and open methods (Newton–Raphson, secant, Steffenson) hybrid methods (Badr et al. [16] and Sabharwal [17]), and metaheuristic algorithms, etc. There are benefits and drawbacks to each of the strategies mentioned above (there is no specific way to solve nonlinear equations effectively). Open approaches are quick but do not converge, whereas closed approaches are known for being slow but close. This research was motivated by the desire to propose fast and convergent methods, as opposed to the conventional methods, because hybrid methods are characterized by combining speed and convergence.

In fact, the idea of developing a hybrid technique by fusing two traditional methodologies is not new but rather has a long history. In 1995, Novak et al. [18] introduced a hybrid secant–bisection method. A novel hybrid bracketing strategy (bisection-false position) was proposed by Sabhrwal [17]. On the other hand, Badr et al. [16] created a hybrid method called Tri-FP that combines two closed algorithms. On fifteen benchmark linear and nonlinear equations, they evaluated the method's performance. Sabhrwal's [17] approach (Bi-FP) was outperformed by the hybrid algorithm (Tri-FP) proposed by Badr et al. [16]. Recently, two new blended algorithms that make use of the advantages of open approaches and bracketing methods were suggested by Badr et al. [19].

The main objective of this research was to combine bracketing methods to propose two hybrid algorithms. It is well known that the modified false position method is better than the classical false position method. The first algorithm is a combination of bisection and modified false position methods (Bi-MFP). The second algorithm combines trisection and modified false position methods (Tri-MFP). We evaluated the proposed algorithms using fifteen benchmark functions against bisection, trisection, and modified false position methods. The numerical results showed that the proposed algorithms outperformed the classical bisection, trisection, and modified false position methods in terms of the number of iterations.

2. Classical Methods

In this section, we discuss four classical methods for finding the roots of non-linear equations. These methods are the bisection method, the trisection method, the false position method, and the modified false position method. This section includes details of these classical methods.

2.1. Bisection Method

The intermediate value theorem (IVT) says that $\Psi(\xi)$ has at least a root or zero of $\Psi(\xi) = 0$ inside the interval $[\omega_1, \omega_2]$ if it is continuous and defined on the interval $[\omega_1, \omega_2]$, fulfilling the relation $\Psi(\omega_1) * \Psi(\omega_2) < 0$ with $\Psi(\omega_1)$ and $\Psi(\omega_2)$ are of opposite signs. Finding two real numbers ω_1 and ω_2 that lie in the interval $[\omega_1, \omega_2]$ and the root in $\omega_1 \leq \xi \leq \omega_2$ with each step's length being half of the interval's initial length is the function of this method.

This process is continued until the interval imparts the desired precision as a result of obtaining the necessary root. The algorithms only work when the multiplicity of a root is odd. Additionally, the root converges very slowly and linearly. However, we can stop iterating when $\Psi(c)$ is very close to zero or is very small. Consequently, the basic formula is

$$c = \frac{\omega_1 + \omega_2}{2}. \tag{1}$$

The size of the interval is reduced by half at each iteration. Therefore, the tolerance (eps) is determined from the following formula:

$$eps = \frac{\omega_2 - \omega_1}{2^{\varpi}}, \tag{2}$$

where ϖ is the number of iterations. From (2), the number of iterations is found by

$$\varpi = \left[\log_2 \left(\frac{\omega_2 - \omega_1}{eps} \right) \right]. \tag{3}$$

As a bracketing approach, the bisection method brackets the root in the range $[\omega_1, \omega_2]$, halving the range's size after each iteration. As a result, this decreases the error between an iteration's approximation root and precise root. However, the bisection method only works quickly if the approximate root is far from the interval's endpoint; otherwise, more iterations are required to reach the root [3].

2.2. Trisection Method

The trisection method [7] is like the bisection method, except that it divides the interval $[\omega_1, \omega_2]$ into three subintervals, while the bisection method divides the interval into two partial periods. This algorithm divides the interval $[\omega_1, \omega_2]$ into three equal subintervals and searches for the root in the subinterval that contains different signs of the function values at the endpoints of these subintervals. If the condition of termination is true, then the iteration has finished its task; otherwise, the algorithm repeats the calculations.

To divide the distance $[\omega_1, \omega_2]$ by ξ_1 into thirds and ξ_2, one needs to know the location of ξ_1 and ξ_2, as follows:

$$\xi_1 - \omega_1 = \omega_2 - \xi_2 \tag{4}$$
$$\xi_2 - \xi_1 = \xi_1 - \omega_1. \tag{5}$$

By solving Equations (4) and (5), we obtain

$$\xi_1 = \frac{2\omega_1 + \omega_2}{3}$$

and

$$\xi_2 = \frac{\omega_1 + 2\omega_2}{3}.$$

The size of the interval $[\omega_1, \omega_2]$ decreases to a third with each repetition. Therefore, the value *eps* is determined from the following formula:

$$eps = \frac{\omega_2 - \omega_1}{3^{\varpi}}, \tag{6}$$

where ϖ is the number of iterations. From (6) the number of iterations is found by

$$\varpi = \left[\log_3\left(\frac{\omega_2 - \omega_1}{eps}\right)\right]. \tag{7}$$

When we compare Equations (2) and (7), we can conclude that the iterations number of the trisection algorithm is less than the iteration number of the bisection algorithm. We might think that the trisection algorithm is better than the bisection algorithm, since it requires fewer iterations. However, it might be the case that one iteration of the trisection algorithm has an execution time longer than the execution time of one iteration of the bisection algorithm. Therefore, we will consider both execution time and the number of iterations to evaluate the different algorithms, see [2,3,5,6,20] and references therein.

2.3. False Position (Regula Falsi) Method

In this technique, one uses results that are known to be false to converge to the true root. This method chooses initial approximations ξ_0 and ξ_1, such that $\Psi(\xi_0) * \Psi(\xi_1) < 0$. The new approximation value is be then obtained using the following relation:

$$\xi_{\varpi} = \xi_{\varpi-1} - \frac{\Psi(\xi_{\varpi-1})(\xi_{\varpi-1} - \xi_{\varpi-2})}{\Psi(\xi_{\varpi-1}) - \Psi(\xi_{\varpi-2})}, \tag{8}$$

for $\varpi \geq 2$.

After that, and in order to decide which secant line to use, the product of $\Psi(\xi_2)$ and $\Psi(\xi_1)$ should be taken. If $\Psi(\xi_1) * \Psi(\xi_2) < 0$ then choose ξ_3 as a line joining $(\xi_1, \Psi(\xi_1))$ and $(\xi_2, \Psi(\xi_2))$, and if $\Psi(\xi_1) * \Psi(\xi_2) > 0$ then choose ξ_3 as a line joining $(\xi_0, \Psi(\xi_0))$ and $(\xi_2, \Psi(\xi_2))$. The process continues until the stopping criteria are satisfied.

2.4. Modified False Position Method

In this method, an improvement over the false position method is obtained by replacing the secant with straight lines with an even smaller slope, until ξ falls to the other side of the zero of $\Psi(\xi)$. To begin, we take the interval $[\omega_1, \omega_2]$ in which the root lies and apply the false position to determine the value of ξ_1, as given by

$$\xi_1 = \omega_2 - \frac{\Psi(\omega_2)(\omega_2 - \omega_1)}{\Psi(\omega_2) - \Psi(\omega_1)}. \tag{9}$$

Then, the products of $\Psi(\xi_1)$, $\Psi(\omega_1)$ and $\Psi(\omega_2)$ decide the root interval. If $\Psi(\xi_1) * \Psi(\omega_1) > 0$, then this implies that root lies in interval $[\xi_1, \omega_2]$ and we take $\omega_1 = \xi_1$ in interval $[\omega_1, \omega_2]$

with a fixed value of ω_2 in a complete iteration. Then, to find the next approximation ξ_2, impose the half function value of ω_2 using the following relation:

$$\xi_2 = \omega_2 - \frac{\frac{\Psi(\omega_2)}{2}(\omega_2 - \omega_1)}{\frac{\Psi(\omega_2)}{2} - \Psi(\omega_1)}. \tag{10}$$

If $\Psi(\xi_1) * \Psi(\omega_2) > 0$ then we take $\omega_2 = \xi_1$ and the fixed value of ω_1 in whole iteration. The next value of approximation ξ_2 can be found by imposing half function value of ω_1, as given below

$$\xi_2 = \omega_2 - \frac{\Psi(\omega_2)(\omega_2 - \omega_1)}{\Psi(\omega_2) - \frac{\Psi(\omega_1)}{2}}. \tag{11}$$

This process continues until the stopping criteria satisfied.

3. Hybrid Algorithms

Instead of the classical methods (bisection, trisection, false position, modified false position), we propose two new hybrid algorithms: a bisection-modified false position method (Bi-MFP) and a trisection-modified false position method (Tri-MFP). These methods have the benefits of bisection, trisection, and modified false position methods. Badr et al. [16,17] proposed blended algorithms containing bisection, trisection, and false position methods. It is shown in this paper that our proposed algorithms performed better than their component algorithms, in the sense of the number of iterations.

3.1. Bisection-Modified False Position Method

This proposed algorithm (bisection-modified false position method) contains the advantages of both the bisection and modified false position methods. It is well known that the modified false position method outperforms the classical false position method, so we took advantage of the superiority of the modified false position method in our proposed algorithm. A flowchart and pseudocode for the Bi-MFP method are given in Figure 1 and Algorithm 1, respectively.

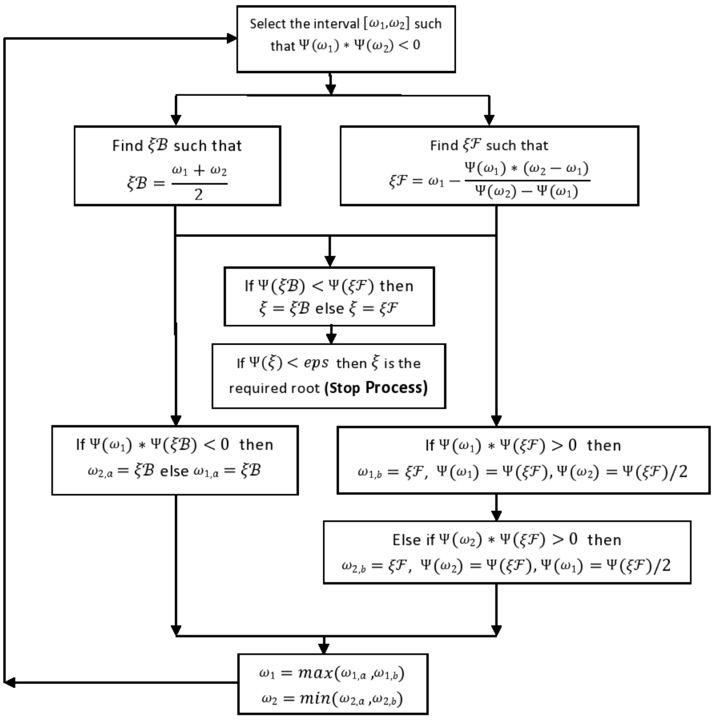

Figure 1. Flow chart of bisection-modified false position.

Algorithm 1: Hybrid Bi-MRF $(\Psi, \omega_1, \omega_2, eps)$.

INPUT: The function $\Psi(\xi)$. The interval $[\omega_1, \omega_2]$ where the root lies in, The absolute error (eps)
OUTPUT: The root (ξ). The value of $\Psi(\xi)$. Number of iterations (ϖ). The interval $[\omega_1, \omega_2]$ where the root lies in.
$\varpi := 0$
$\omega_{1,a} := \omega_1$
$\omega_{1,b} := \omega_1$
$\omega_{2,a} := \omega_2$
$\omega_{2,b} := \omega_2$
while true **do**
  $\varpi := \varpi + 1$
  $\xi\mathcal{B} := (\omega_1 + \omega_2)/2$
  $\xi\mathcal{F} := \omega_1 - (\Psi(\omega_1) * (\omega_2 - \omega_1))/(\Psi(\omega_2) - \Psi(\omega_1))$
  if $|\Psi(\xi\mathcal{B})| < |\Psi(\xi\mathcal{F})|$ **then**
    $\xi := \xi\mathcal{B}$
  else
    $\xi := \xi\mathcal{F}$
  end if
  if $|\Psi(\xi)| <= eps$ **then**
    return $\xi, \Psi(\xi), \varpi, \omega_1, \omega_2$
  end if
  if $\Psi(\omega_1) * \Psi(\xi\mathcal{B}) < 0$ **then**
    $\omega_{2,a} := \xi\mathcal{B}$
  else
    $\omega_{1,a} := \xi\mathcal{B}$
  end if
  if $\Psi(\omega_1) * \Psi(\xi\mathcal{F}) > 0$ **then**
    $\omega_{1,b} := \xi\mathcal{F}$
    $\Psi(\omega_1) := \Psi(\xi\mathcal{F})$
    $\Psi(\omega_2) := \Psi(\omega_2)/2$
  else
    if $\Psi(\omega_2) * \Psi(\xi\mathcal{F}) > 0$ **then**
      $\omega_{2,b} := \xi\mathcal{F}$
      $\Psi(\omega_2) := \Psi(\xi\mathcal{F})$
    end if
  end if
  $\omega_1 := max(\omega_{1,a}, \omega_{1,b})$
  $\omega_2 := min(\omega_{2,a}, \omega_{2,b})$
  end while
end while

3.2. Trisection-Modified False Position Method

Badr et al. [16] proposed a new blended algorithm called the trisection-false position algorithm, which is better than Sabharwal's bisection-false position algorithm [17] in terms of iteration number and average CPU time. We proposed a new method having supremacy over the modified false position method, called the trisection-modified false position method (Tri-MFP). A flowchart and pseudocode of the proposed methods are given below in Figure 2 & Algorithm 2 respectively:

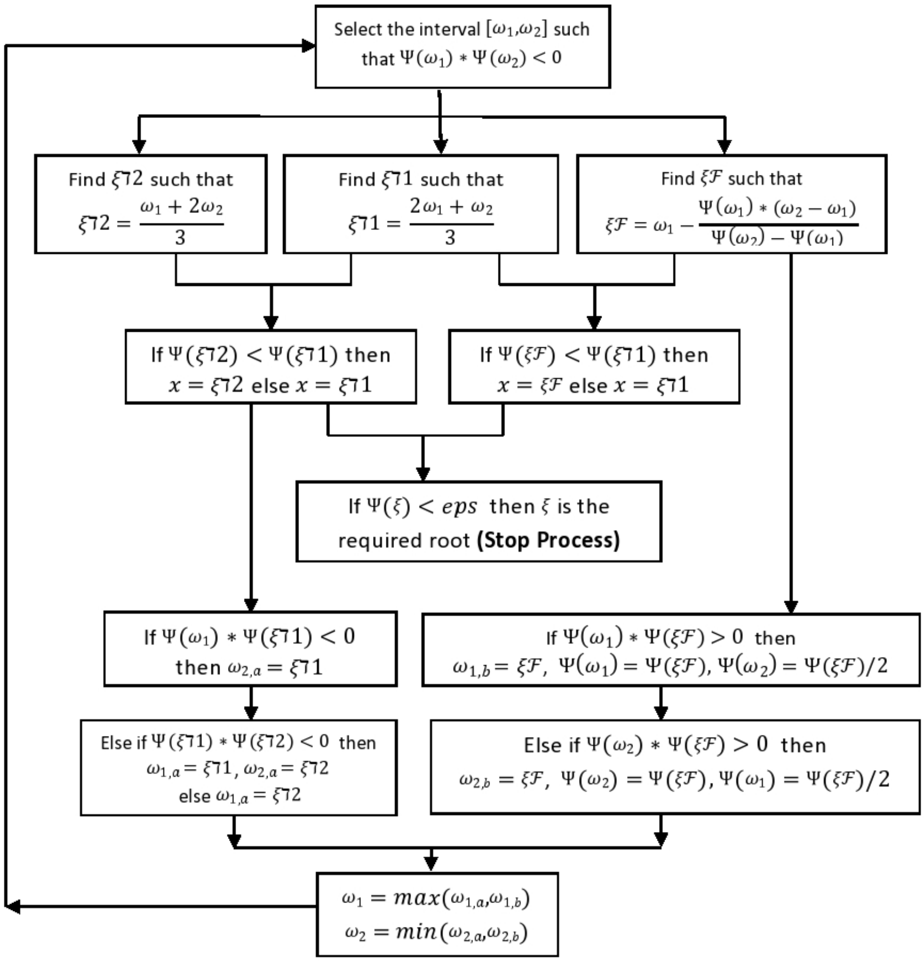

Figure 2. Flow chart of trisection-modified false position.

3.3. Empirical Study

MAPLE version 18.0, 64-bit, windows 7, Core(TM)2, T5600, 1.83 GHz, and 3.00 GB of memory were the elements that made up the software environment. In this paper, we used an absolute error of $(eps = 10^{-10})$ to terminate all the algorithms. The iteration number and CPU time are effective tools for the comparison of algorithms, so we ran every algorithm ten times and calculated the average of the running time to obtain an accurate running time and avoid the problem of operating systems.

We evaluate the suggested technique for numerous benchmark problems as shown in Table 1, since it was not accurate enough to draw conclusions from only one problem. The abbreviations AppRoot, Error, LowerB, and UpperB in Tables 2–6 represent the approximate root, the difference between two consecutive roots, and the lower and upper bounds, respectively.

Figures 1 and 2 shows flow charts of the proposed algorithms Bi-MFP and Tri-MFP. Meanwhile, Figures 3 and 4 display the analysis of the benchmark problems shown in Table 1, according to the number of iterations and average CPU time, respectively.

Algorithm 2: Hybrid Tri-MRF $(\Psi, \omega_1, \omega_2, eps)$.

INPUT: The function $\Psi(\xi)$. The interval $[\omega_1, \omega_2]$ where the root lies in, The absolute error (eps)
OUTPUT: The root (ξ). The value of $\Psi(\xi)$. Number of iterations (ϖ). The interval $[\omega_1, \omega_2]$ where the root lies in.

```
ϖ := 0, ω_{1,a} := ω_1, ω_{1,b} := ω_1, ω_{2,a} := ω_2, ω_{2,b} := ω_2
while true do
  ϖ := ϖ + 1
  ξT1 := (2 * ω_1 + ω_2)/3
  ξT2 := (ω_1 + 2 * ω_2)/3
  ξF := ω_1 − (Ψ(ω_1) * (ω_2 − ω_1))/(Ψ(ω_2) − Ψ(ω_1))
  ξ := ξT1
  Ψξ := Ψ(ξT1)
  if |Ψ(ξT2)| < |Ψ(ξ)| then
    ξ := ξT2
  end if
  if |Ψ(ξF)| < |Ψ(ξ)| then
    ξ := ξF
  end if
  if |Ψ(ξ)| <= eps then
    return ξ, Ψ(ξ), ϖ, ω_1, ω_2
  end if
  if Ψ(ω_1) * Ψ(ξT1) < 0 then
    ω_{2,a} := ξT1
  else
    if Ψ(ξT1) * Ψ(ξT2) < 0 then
      ω_{1,a} := ξT1
      ω_{2,a} := ξT2
    else
      ω_{1,a} := ξT2
    end if
  end if
  if Ψ(ω_1) * Ψ(ξF) > 0 then
    ω_{1,b} := ξF
    Ψ(ω_1) := Ψ(ξF)
    Ψ(ω_2) := Ψ(ω_2)/2
  else
    if Ψ(ω_2) * Ψ(ξF) > 0 then
      ω_{2,b} := ξF
      Ψ(ω_2) := Ψ(ξF)
    end if
  end if
  ω_1 := max(ω_{1,a}, ω_{1,b})
  ω_2 := min(ω_{2,a}, ω_{2,b})
  end while
end while
```

Table 1. Standard benchmark problems.

No.	Problem	Interval	References
P1	$\xi e^{\xi} - 7$	$[0, 2]$	Calihoun [13]
P2	$\xi - \cos(\xi)$	$[0, 1]$	Ehiworio [3]
P3	$\xi \sin(\xi) - 1$	$[0, 2]$	Mathews [18]
P4	$\sin(\xi) \sinh(\xi) + 1$	$[1.5, 4]$	Esfandiari [17]
P5	$e^{\xi} - 3\xi - 2$	$[0, 3]$	Hoffman [14]
P6	$\sin(\xi) - \xi^2$	$[0.5, 1]$	Chapra [11]

Table 2. Solutions of some problems using the bisection method.

Problems	Bisection Method					
	Iter	Average CPU Time	Approx Root	Function Value	Lower Bound	Upper Bound
P1	32	0.1118	1.52434520468	−0.000000002	1.52434520422	1.52434520515
P2	35	0.1152	0.739085134091	0.000000015	0.739085134062	0.739085134119
P3	32	0.1180	1.11415714072	0.000000000	1.11415714026	1.11415714119
P4	33	0.1336	3.22158839905	0.000000000	3.22158839876	3.22158839934
P5	33	0.1308	2.12539119889	0.000000000	2.12539119854	2.12539119923
P6	33	0.1214	0.876726215414	0.000000000	0.876726215355	0.876726215471

Table 3. Solutions of some problems using the trisection method.

Problems	Trisection Method					
	Iter	Average CPU Time	Approx Root	Function Value	Lower Bound	Upper Bound
P1	20	0.1214	1.52434520541	0.000000003	1.52434520484	1.52434520656
P2	22	0.1244	0.739085133325	0.000000000	0.739085133262	0.739085133357
P3	21	0.1120	1.11415714133	0.000000000	1.11415714114	1.11415714171
P4	21	0.1180	3.22158839889	0.000000029	3.22158839841	3.22158839913
P5	21	0.1212	2.12539119879	−0.000000001	2.12539119823	2.12539119908
P6	20	0.1338	0.876726215456	−0.000000000	0.876726215170	0.876726215600

Table 4. Solutions of some problems using the modified regula falsi method.

Problems	Modified Regula Falsi Method					
	Iter	Average CPU Time	Approx Root	Function Value	Lower Bound	Upper Bound
P1	20	0.1304	1.52434520493	−0.0000000062	1.52434520482	2.0000000000
P2	9	0.1370	0.739085133213	0.0000000000	0.739085133171	1.0000000000
P3	6	0.1244	1.11415714087	0.0000000000	1.11415714087	1.11415714304
P4	31	0.1524	3.22158839849	0.0000000081	3.22158839777	4.0000000000
P5	33	0.1244	2.12539119823	−0.0000000031	2.12539119761	3.0000000000
P6	13	0.1306	0.876726215392	0.000000000	0.876726215372	1.0000000000

Table 5. Solutions of some problems using the bisection and modified regula falsi methods.

Problems	Bisection-Modified Regula Falsi Method					
	Iter	Average CPU Time	Approx Root	Function Value	Lower Bound	Upper Bound
P1	9	0.1182	1.52434520539	0.000000000	1.52434520445	1.52772146940
P2	7	0.1118	0.739085133226	0.000000000	0.739085129706	0.745369013289
P3	15	0.1180	1.11415714088	0.000000000	1.11414625500	1.11425614878
P4	10	0.1214	3.22158839955	−0.000000012	3.22158839943	3.22238911820
P5	9	0.1086	2.12539119894	0.000000001	2.12539118521	2.12787084220
P6	7	0.1308	0.876726215441	−0.0000000001	0.876726210685	0.877268445426

Table 6. Solutions of some problems using the trisection and modified regula falsi methods.

Problems	Trisection-Modified Regula Falsi Method					
	Iter	Average CPU Time	Approx Root	Function Value	Lower Bound	Upper Bound
P1	6	0.1180	1.52434520508	0.00000255309	1.52434520413	1.52441127915
P2	6	0.0964	0.739085133236	0.0003113615	0.739085133117	0.739643235290
P3	12	0.1118	1.11415714132	0.000001774	1.11415714046	1.11415841776
P4	8	0.1056	3.22158839889	1.5×10^{-9}	3.22158839878	3.22215305257
P5	7	0.1276	2.12539119914	1.0×10^{-9}	2.12539119840	2.12548466697
P6	5	0.1212	0.876726215473	2.0×10^{-10}	0.876726215114	0.876727040681

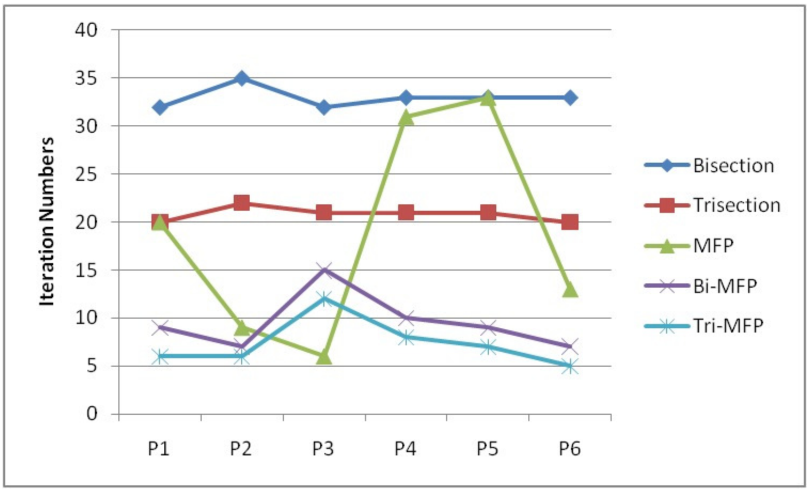

Figure 3. Comparison of six standard problems according to the number of iterations.

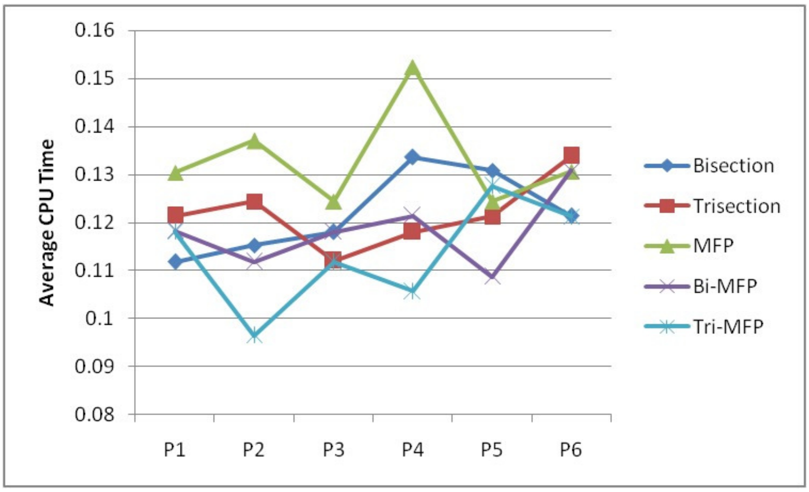

Figure 4. Comparison of six standard problems according to CPU time.

3.4. Some New Numerical Experiments

We now consider some nonlinear polynomial equations from applied biomedical engineering and practical sciences.

Example 1 (Blood rheology and fractional nonlinear equations Model [21])**.** *Blood is modeled as a "Casson Fluid" as it is a non-Newtonian fluid. According to the Casson fluid model, basic fluids such as blood and water flow through a tube in such a way that the velocity gradient near the wall and the center core of the fluids travel as plugs, with minimal deformation. The following* *nonlinear fractional equation, which measures the drop in flow rate, was used to elaborate the plug flow of Casson fluids:*

$$\Psi(\xi) = 1 - \frac{16}{7}\sqrt{\xi} + \frac{4}{3}\xi - \frac{1}{21}\xi^4 - G, \tag{12}$$

or where the reduction in flow rate is measured by $G = 0.40$. *Table 7 and Figures 5 and 6 show numerical and graphical comparisons of the classical and proposed methods.*

Table 7. Results of the above example using Bi, Tri, MFP, Bi-MFP, and Tri-MFP methods.

Method	Blood Rheology and Fractional Nonlinear Equations Model				
	Iter	**Approx Root**	**Function Value**	**Lower Bound**	**Upper Bound**
Bisection	34	0.104698652342	-1.6×10^{-9}	0.104698652284	0.104698652401
Trisection	23	0.104698652104	-1.3×10^{-9}	0.104698652083	0.104698652115
MFP	44	0.104698651667	-2.86×10^{-10}	0.000000000000	0.104698651748
Bi-MFP	19	0.104698651542	0.0000000000	0.104698181152	0.104701995849
Tri-MFP	13	0.104698651485	-7.335×10^{-7}	0.104698357861	0.104700239538

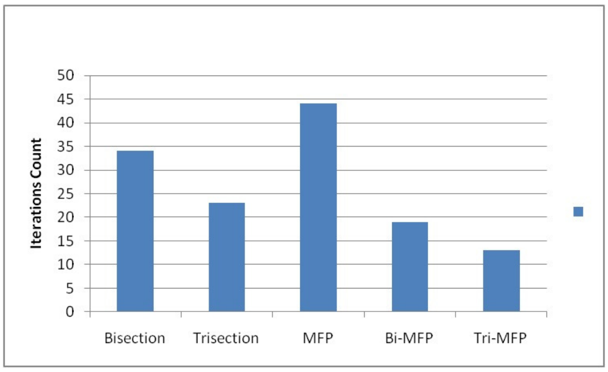

Figure 5. Comparison of methods according to iterations.

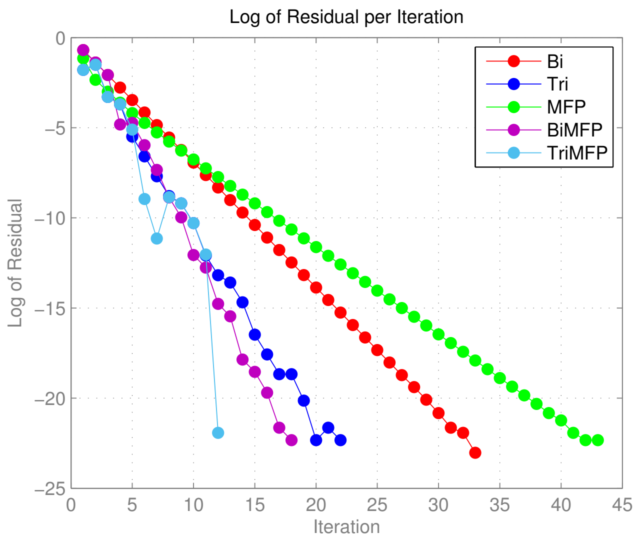

Figure 6. Comparison of log of residual per iteration.

Example 2 (Fluid Permeability in Biogels [21])**.** *The following nonlinear equation comes from the specific hydraulic permeability relationship between the pressure gradient and fluid velocity in porous media (agarose gel or extracellular fiber matrix):*

$$\Psi(\xi) = \Re_e \xi^3 - 20\kappa(1-\xi)^2, \tag{13}$$

Comparison of the proposed methods with the classical methods are numerically and graphically displayed in Table 8 and Figure 7 & Figure 8, respectively.

Table 8. The results of the above example using the Bi, Tri, MFP, Bi-MFP, and Tri-MFP methods

Method	Fluid Permeability in Biogels				
	Iter	Approx Root	Function Value	Lower Bound	Upper Bound
Bisection	30	1.00003698747	7.51×10^{-13}	1.00003698654	1.00003698840
Trisection	20	1.00003698880	-3.31×10^{-13}	1.00003698823	1.00003698909
MFP			Fail		
Bi-MFP	18	1.00003698850	2.09×10^{-13}	1.00003698808	1.00003699241
Tri-MFP	15	1.00003698885	-1.41×10^{-12}	1.00003698848	1.00003699616

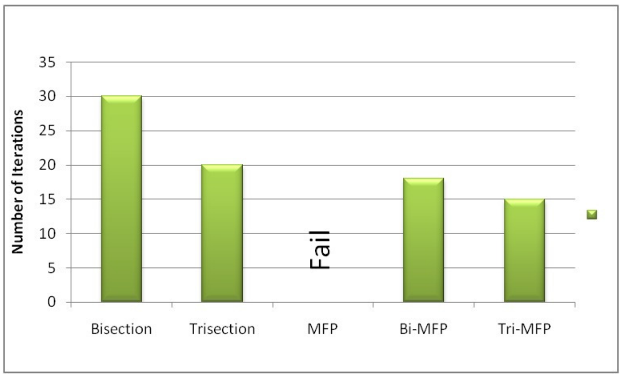

Figure 7. Comparison of the methods according to iterations.

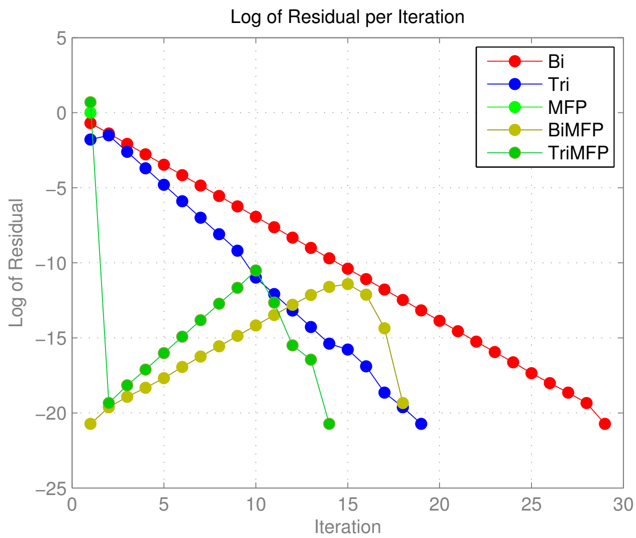

Figure 8. Comparison of the log of residual per iteration.

Example 3 (beam position model [22]). *Consider a mechanical issue with beam positioning that results in the following nonlinear equation:*

$$\Psi(\xi) = \xi^4 + 4\xi^3 - 24\xi^2 + 16\xi + 16. \tag{14}$$

Comparison of the proposed methods with the classical methods are numerically and graphically displayed in Table 9 and Figure 9 & Figure 10, respectively.

Table 9. Results of the above example using the Bi, Tri, MFP, Bi-MFP, and Tri-MFP methods.

Method	Beam Position Model				
	Iter	Approx Root	Function Value	Lower Bound	Upper Bound
Bisection	34	−0.535898384669	1.0×10^{-8}	−0.535898384698	−0.535898384639
Trisection	22	−0.535898384057	3.0×10^{-8}	−0.535898384089	−0.535898384042
MFP	15	−0.535898384849	6.0×10^{-10}	−1.000000000000	−0.535898384808
Bi-MFP	15	−0.535898384738	1.00×10^{-8}	−0.535919189454	−0.535888671875
Tri-MFP	12	−0.535898384898	-1.06×10^{5}	−0.535899563639	−0.535896741125

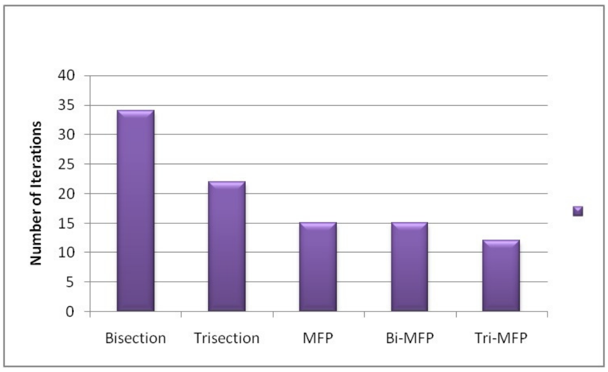

Figure 9. Comparison of the methods according to Iteration.

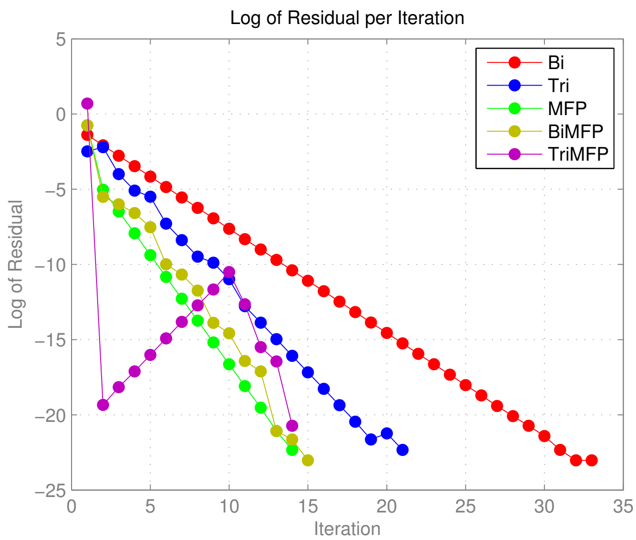

Figure 10. Comparison of the log of residual per iteration.

4. Conclusions

This research study introduced a pair of innovative hybrid algorithms that enhanced the efficiency of bisection, trisection, and modified false position methods. These new hybrid algorithms—bisection-modified false position (Bi-MFP) and trisection-modified false position (Tri-MFP)—outperformed the methods previously proposed by [16,17]. Moreover, we tested some examples and arranged the results in a numerical and graphical manner. Moving forward, in the future, we will focus on some open and bracketing methods for solving non-linear equations.

Author Contributions: Conceptualization, N.Z.A. and A.G.K.; methodology, M.V.-C., N.Z.A., A.G.K. and M.U.A.; software, N.Z.A., A.G.K. and M.U.A.; validation, M.V.-C., N.Z.A., A.G.K. and M.U.A.; formal analysis, N.Z.A. and A.G.K.; investigation, M.V.-C., N.Z.A., A.G.K. and M.U.A.; writing—original draft preparation, M.V.-C., N.Z.A., A.G.K. and M.U.A.; writing—review and editing, A.G.K. and M.U.A.; visualization, M.V.-C., N.Z.A., A.G.K. and M.U.A.; supervision, A.G.K. and M.U.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research is supported by Pontificia Universidad Católica del Ecuador Proyect Título: "Algunos resultados Cualitativos sobre Ecuaciones diferenciales fraccionales y desigualdades integrales" Cod UIO2022.

Acknowledgments: The authors are thankful to the editor and anonymous reviewers for their valuable comments and suggestions.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Burden, R.L.; Faires, J.D. *Numerical Analysis*, 9th ed.; Brooks/Cole, Cengage Learning: Boston, MA, USA, 2011.
2. Chapra, S.C.; Canale, R.P. *Numerical Methods for Engineers*, 7th ed.; McGraw-Hill: Boston, MA, USA, 2015.
3. Ehiwario, J.C.; Aghamie, S.O. Comparative Study of Bisection, Newton-Raphson and Secant Methods of Root-Finding Problems. *IOSR J. Eng.* **2014**, *4*, 1–7.
4. Ernst, T. A method for q-calculus. *J. Nonlinear Math. Phys.* **2003**, *10*, 487–525. [CrossRef]
5. Esfandiari, R.S. *Numerical Methods for Engineers and Scientists Using MATLAB*; CRC Press: Boca Raton, FL, USA, 2013.
6. Harder, D.W.; Khoury, R. *Numerical Methods and Modelling for Engineering*; Springer International Publishing: Berlin/Heidelberg, Germany, 2016.
7. Joe, D.H. *Numerical Methods for Engineers and Scientists*, 2nd ed.; CRC Press: Boca Raton, FL, USA, 2001.
8. Kac, V.; Cheung, P. *Quantum Calculus*; Springer: New York, NY, USA, 2002.
9. Khan, A.G.; Ameen, F.; Awan, M.U.; Nonlaopon, K. Some new numerical schemes for finding the solutions to nonlinear equations. *AIMS Math.* **2022**, *7*, 18616–18631. [CrossRef]
10. Mathews, J.H.; Fink, K.D. *Numerical Methods Using Matlab*, 4th ed.; Prentice-Hall Inc.: Upper Saddle River, NJ, USA, 2004; ISBN 0-13-065248-2.
11. Nonlaopon, K.; Khan, A.G.; Ameen, F.; Awan, M.U.; Cesarano, C. Multi-Step Quantum Numerical Techniques for Finding the Solutions of Nonlinear Equations. *Mathematics* **2022**, *10*, 2595. [CrossRef]
12. Nonlaopon, K.; Khan, A.G.; Ameen, F.; Awan, M.U.; Cesarano, C. Some New Quantum Numerical Techniques for Solving Nonlinear Equations. *Symmetry* **2022**, *14*, 1829. [CrossRef]
13. Noor, M.A. Fifth-order convergent iterative method for solving nonlinear equations using quadrature formula. *J. Math. Control Sci. Appl.* **2018**, *4*, 95–104.
14. Sana, G.; Mohammed, P.O.; Shin, D.Y.; Noor, M.A.; Oudat, M.S. On iterative methods for solving nonlinear equations in quantum calculus. *Fractal Fract.* **2021**, *5*, 60. [CrossRef]
15. Srivastava, R.B.; Srivastava, S. Comparison of numerical rate of convergence of bisection, Newton and secant methods. *J. Chem. Biol. Phys. Sci.* **2011**, *2*, 472–479.
16. Badr, E.; Almotairi, S.; Ghamry, A.E. A Comparative Study among New Hybrid Root Finding Algorithms and Traditional Methods. *Mathematics* **2021**, *9*, 1306. [CrossRef]
17. Sabharwal, C.L. Blended root finding algorithm outperforms bisection and Regula Falsi Algorithms. *Mathematics* **2019**, *7*, 1118. [CrossRef]
18. Novak, E.; Ritter, K.; Wozniakowski, H. Average-case ompitmality of a hybrid secant-bisection method. *Math. Comput.* **1995**, *64*, 1517–1539. [CrossRef]
19. Badr, H.; Attiya, S.; Ghamry, A.E. Novel hybrid algorithms for root determining using advantages of open methods and bracketing methods. *Alex. Eng. J.* **2022**, *61*, 11579–11588. [CrossRef]
20. Hasan, A.; Ahmad, N. Compartive study of a new iterative method with that Newtons Method for solving algebraic and transcesental equations. *Int. J. Comput. Math. Sci.* **2015**, *4*, 32–37.
21. Fournier, R.L. *Basic Transport Phenomena in Biomedical Engineering*; Taylor and Francis: New York, NY, USA, 2007.
22. Zafar, F.; Cordero, A.; Torregrosa, J.R. An efficient family of optimal eighth-order multiple root finders. *Mathematics* **2018**, *6*, 310. [CrossRef]