DATA ANALYTICS AND APPLIED MATHEMATICS
E-ISSN: 2773-4854
VOLUME 7, ISSUE 1, 2026, 17 – 21
DOI: https://doi.org/10.15282/daam.v7i1.13060

RESEARCH ARTICLE

Shooting method with root finding algorithm to solve boundary value problems

Matakim Aryani Rima[1], Nur Idalisa Norddin[1]*, Wan Normaini Wan Mohamad Idris[1], Nurul Hafawati Fadhilah[1], Nur Hanisah Abdul Malek[2]

[1]Department of Mathematics, Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA Cawangan Terengganu Kampus Dungun, Malaysia
[2]Department of Mathematics, Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA Cawangan Terengganu Kampus Bukit Besi, Malaysia

Abstract - One important numerical method for converting Boundary Value Problems (BVPs) into Initial Value Problems (IVPs) is the shooting method, which is mostly used as a root-finding technique. Until the boundary conditions are met, this approach entails making initial-condition predictions and iteratively improving them. Since it guarantees that the initial values yield a solution that satisfies the given boundary conditions, root discovery plays a critical role. Although the shooting approach works well, it can be sensitive to initial estimates, which can sometimes cause convergence problems. Therefore, choosing the right root-finding techniques becomes essential. Thus, this study will assess how well the shooting method performs when paired with various root-finding algorithms to solve BVPs. The effectiveness of these algorithms is compared based on their convergence rate, accuracy, and robustness. The results demonstrate the effectiveness of combining the shooting method with root-finding algorithms to deliver accurate and efficient solutions to BVPs, making it a valuable tool in scientific and engineering applications.

Article history
Received : 30 September 2025
Revised : 18 January 2026
Accepted : 25 February 2026
Published : 31 March 2026

Keywords
Shooting method
Root finding
Boundary value problems
Python

1. Introduction

Boundary value problems (BVPs) are a class of differential equations in which the solution is subject to specified boundary conditions at the endpoints of the domain. BVPs have broad applications in mathematics, physics, fluid mechanics, and engineering. As analytical solutions are often impractical, advanced numerical methods are critical for solving BVPs effectively [1]. BVPs are integral to advancements in applied science and engineering, particularly in the development of essential components such as pivots and bearings [2]. These problems, which involve differential equations with specified derivatives at multiple points, often require numerical methods because deriving analytical solutions is challenging. Among the most effective methods for solving BVPs are the Shooting Method and Finite Difference Methods [3]. The Shooting Method is particularly notable for its ability to convert BVPs into Initial Value Problems, enabling iterative adjustments to initial conditions until the specified boundary conditions are satisfied [2-3]. Recently, Victor et al. [4] proposed a machine learning workflow for tuning numerical settings in the shooting method. On the other hand, by integrating root-finding algorithms, the shooting method achieves greater precision and efficiency by fine-tuning parameters to satisfy boundary constraints [5].

Table 1. Formula for root-finding algorithm

Root finding algorithms	Formulas
Bisection method	$c = \dfrac{a+b}{2}$
Secant method	$a_2 = a_{2-1} - \dfrac{(y(b,a_{2-1})-\beta)(a_{2-1}-a_{2-2})}{y(b,a_{2-1})-y(b,a_{2-2})}$
Brent's method	$x = \dfrac{f(a)f(b)c}{[f(c)-f(a)][f(c)-(b)]} + \dfrac{f(a)f(c)b}{[f(b)-f(a)][f(b)-(c)]} + \dfrac{f(b)f(c)a}{[f(a)-f(b)][f(a)-(c)]}$
Ridder's method	$x_4 = x_3 \pm (x_3 - x_1)\dfrac{f_3}{\sqrt{f_3^2 - f_1 f_2}}$

The effectiveness of root-finding algorithms depends on their unique approaches and efficiencies. In general, bracketing methods and iterative approaches are the two categories of root-finding algorithms. An interval is divided into two halves using the bisection method, a kind of bracketing technique. When a specific precision is met, the root is then found using the intermediate value theorem. However, the Bisection Method is somewhat sluggish despite its reliability [6]. Newton's and Secant Methods are iterative approaches that frequently provide faster convergence; the former uses derivatives, while the latter does not [7]. Furthermore, hybrid approaches that integrate several methodologies to increase accuracy include Brent's and Ridder's methods [8]. Selecting the initial guess for the root-finding process is another

*CORRESPONDING AUTHOR | Nur Idalisa Norddin| ✉ nuridalisa@uitm.edu.my

important consideration. Since the Bisection technique requires an interval, two initial endpoint predictions are needed. Two initial guesses are also necessary for root-finding algorithms such as Secant, Brent, and Ridder. On the other hand, root-finding algorithms like Newton's and Ostrowski's require only one initial guess. The approaches of Bisection, Secant, Brent, and Ridder, which use two initial guesses in the root-finding process, are the main topic of this study. The complete formulas are given in Table 1.

2. Methodology

The focus in this study is to solve two order BVPs using shooting methods combined with root-finding algorithms to set up the IVPs. Then those IVPs will be solved using the Runge-Kutta method of order 4. The following is the algorithm used in this study.

2.1 Algorithm (Shooting method with root finding to solve second-order BVP)

Step 1: Initialization

a. Given a second-order BVP of the form:

$y'' = f(x, y, y'),$ for $a \le x \le b$, with boundary conditions $y(a) = \alpha \; and \; y(b) = \beta,$

where endpoints $a, b,$ and boundary conditions α, β are constants.

b. Set iteration k=0, number of subintervals N, $h = (b - a)/N$, and maximum number of iterations, M.

Step 2: Shooting method iteration

a. Let endpoint$a = x_0$. Introduce new variables $w_{1,i}$ and $w_{2,i}$ for each $i = 0,1, \ldots, N$ where $w_{1,0} = y(x_0)$ and $w_{2,0} = w'_{1,0}$.

b. Set initial guess for $w_{2,0} = s_0$. The system of first-order IVP is obtained as follows:

$IVP1: w_{1,i}' = f_1(x_i, w_1, w_2) = w_2$ with $a \le x \le b,\;\; w_{1,0} = \alpha,$

$IVP2: w_{2,i}' = f_2(x_i, w_1, w_2),$ with $a \le x \le b,\; w_{2,0} = s_0.$

Step 3: Solve IVP1 and IVP2 using the Fourth Order Runge-Kutta (RK4) Method.

Set $x_i = a + ih$.

$$K_{1,1} = hf_1(x_i, w_{1,i}, w_{2,i})$$

$$K_{1,2} = hf_2(x_i, w_{1,i}, w_{2,i})$$

$$K_{2,1} = hf_1\left(x_i + \frac{h}{2},\, w_{1,i} + \frac{1}{2}K_{1,1},\, w_{2,i} + K_{1,2}\right),$$

$$K_{2,2} = hf_2\left(x_i + \frac{h}{2},\, w_{1,i} + \frac{1}{2}K_{1,1},\, w_{2,i} + K_{1,2}\right),$$

$$K_{3,1} = hf_1\left(x_i + \frac{h}{2},\, w_{1,i} + \frac{1}{2}K_{2,1},\, w_{2,i} + \frac{1}{2}K_{2,2}\right)$$

$$K_{3,2} = hf_2\left(x_i + \frac{h}{2},\, w_{1,i} + \frac{1}{2}K_{2,1},\, w_{2,i} + \frac{1}{2}K_{2,2}\right)$$

$$K_{4,1} = hf_1(x_i + h,\, w_{1,i} + K_{3,1},\, w_{2,i} + K_{3,2})$$

$$K_{4,2} = hf_2(x_i + h,\, w_{1,i} + K_{3,1},\, w_{2,i} + K_{3,2})$$

$$w_{1,i+1} = w_{1,i} + \frac{1}{6}(K_{1,1} + 2K_{2,1} + 2K_{3,1} + K_{4,1})$$

$$w_{2,i+1} = w_{2,i} + \frac{1}{6}(K_{1,2} + 2K_{2,2} + 2K_{3,2} + K_{4,2})$$

Step 4: Evaluate the boundary condition.

a. Compute the error function $F(s_0) = w_{1,4} - \beta$.

b. Specify the convergence tolerance, $\varepsilon = 10^{-3}$. If $relative\ error = \left\|\frac{F(s_0)}{\beta}\right\| < \varepsilon$, then stop.

Step 5: Set k=k+1. Compute initial guess of $w_{2,0} = s_k$ by using a root finding algorithm based on Table 1 and go to Step 3. Table 2 lists the test problems for two-order BVPs used in this study. The test problems are used to test the numerical performance of root-finding algorithms comprising Bisection, Secant, Brent, and Ridder in solving BVPs with the shooting method.

Table 2. List of test problems

No.	BVP	Exact Solution	Boundary Conditions
1	$y'' = \frac{3}{2}y^2$	$y(x) = 4/(1+2)^2$	$y(0) = 4, y(1) = 1$
2	$y'' = -e^{-2y}$	$y(x) = \ln x$	$y(1) = 0, y(2) = 0.6931$
3	$y'' = y' \cos x - y \ln y$	$y(x) = e^{\sin x}$	$y(0) = 1, y(1.5708) = 2.7183$
4	$y'' = 2y^3 - 6y - 2y^3$	$y(x) = x + x^{-1}$	$y(1) = 2, y(2) = 2.5$
5	$y'' = \frac{1}{2}(1 - (y')^2 - y \sin x)$	$y(x) = 2 + \sin x$	$y(0) = 2, y(2) = 1$
6	$y'' = y' + 2(y - \ln x)^3 - x^{-1}$	$y(x) = x^{-1} + \ln x$	$y(2) = 0.5, y(3) = 0.3333$
7	$y'' = y^3 - yy'$	$y(x) = (x+1)^{-1}$	$y(1) = 0.5, y(2) = 0.3333$
8	$y'' = \frac{1}{8}(32 + 2x^3 - yy')$	$y(x) = x^2 + \frac{16}{x}$	$y(1) = 17, y(3) = \frac{43}{3}$

Four distinct initial guesses have been used to examine how the method behaves and how the solution varies with different initial guesses. These initial guesses are selected based on trial runs and iteratively refined to find the solution that satisfies the boundary conditions. For consistent evaluation, the same step size and tolerance were used for all methods: 0.01 and 0.001, respectively. This step size and tolerance were chosen for balance accuracy and computational efficiency. All problems in Table 2 are solved using Python.

3. Results and Discussion

The performance of all root-finding methods in shooting methods for solving BVPs is then compared and measured based on the relative error of $\left\|\frac{F(s_0)}{\beta}\right\| < \varepsilon$ and the computation time (CPU time). The performance profile introduced by Dolan and Moré [9] has been used to analyze the efficiency, robustness, and success rates of the compared methods. Figure 1 and Figure 2 show the performance profile of relative error and CPU time on two guess methods.

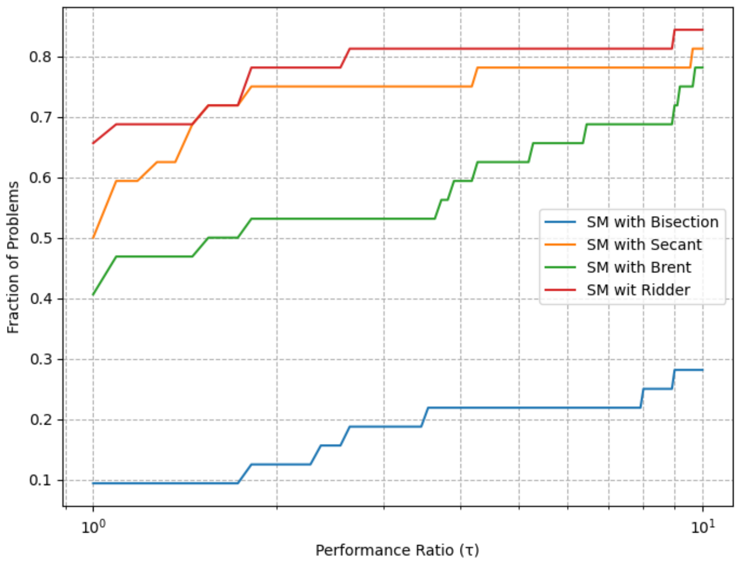

Figure 1. Performance profile of the relative error on the two-guess method

Figure 1 displays the performance profile of relative error for methods that require two initial guesses, such as SM with Bisection, SM with Secant, SM with Brent, and SM with Ridder methods. Based on the graph, SM with the Ridder method shows the highest curve, indicating it is the most accurate and robust method, as it frequently achieves the best relative error. The SM with the Brent method follows closely, offering strong performance but slightly less accuracy and robustness than the SM with the Ridder method. The SM with the Secant method has a moderate curve, indicating it is less accurate and robust but still performs reasonably well. The SM with the Bisection method has the lowest curve, making it the least accurate and robust, as it rarely achieves the best error and often fails to converge. Overall, the SM with the Ridder method is the best method for solving BVPs, while the SM with the Brent method is a strong alternative.

The SM with the Secant method is suitable for simpler problems, and the SM with the Bisection method should only be used when robustness is prioritized over speed and accuracy.

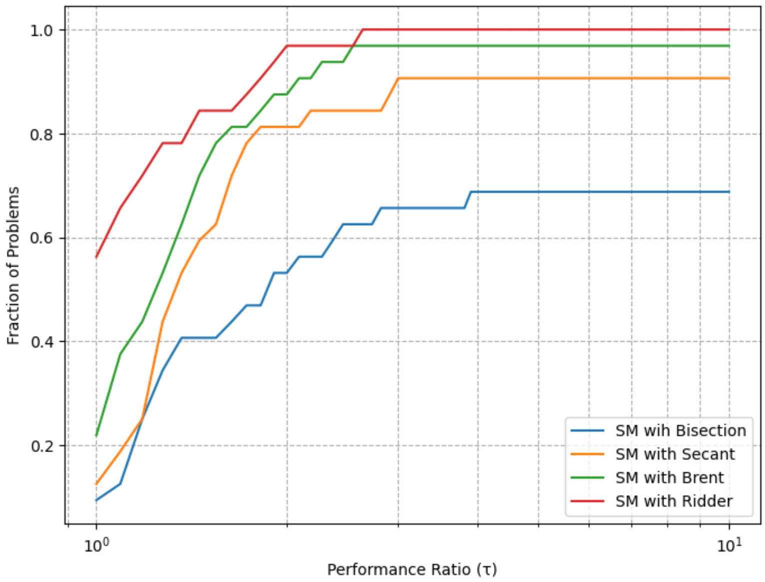

Figure 2. Performance profile of CPU time on two guessing methods

Figure 2 displays the performance profiles of two guessing methods in terms of CPU time. Based on Figure 2, the SM with the Ridder method shows the highest curve, indicating it is the most efficient and robust method. The SM with the Bisection method shown has the lowest curve, reflecting its poor performance and frequent failures. The SM with Secant and Brent methods fall in between, with the SM with Brent method being more robust than the SM with Secant method. Overall, the SM with the Ridder method was the most efficient method for minimizing CPU time, followed by the SM with the Brent method.

4. Conclusions

In conclusion, the SM with the Ridder Method is the best method, as it offers the lowest error and the fastest CPU time, making it well-suited for problems where robustness is critical. It is also observed that a combination of shooting method with a root-finding algorithm can save time by effectively determining accurate initial values for the associated IVPs, compared to manual or uninformed guessing. One potential method for guessing initial points for the shooting method is as proposed by Javeed and Hincal [6]. Future work could explore hybrid approaches that combine the strengths of multiple algorithms or refine existing methods to enhance their robustness and efficiency in solving BVPs.

Acknowledgement

All the authors would like to express their gratitude to Universiti Teknologi MARA for the support/ facilities.

Funding

This study was not supported by any grants from funding bodies in the public, private, or not-for-profit sectors.

Declaration of Competing Interest

The authors declare no conflict of interest.

CRediT Authorship Contribution Statement

Aryani Rima Matakim (Writing - original draft; Data curation; Formal analysis)
Nur Idalisa Norddin (Supervision; Project administration; Validation)
Wan Normaini Wan Mohamad Idris (Conceptualization; Methodology)
Nurul Hafawati Fadhilah (Writing - review & editing; Visualization)
Nur Hanisah Abdul Malek (Writing - review & editing; Software)

Availability of the Data and Materials

The data used to support the findings of this study are included within the article.

Ethical Declarations

No artificial intelligence tools were used in the preparation of this manuscript. All content was developed manually by the authors. This study did not involve human participants or animals. Ethical approval was therefore not required.

Generative Artificial Intelligence Declarations

The authors claim that artificially intelligent-assisted technologies, such as generative AI, were not used to generate content, ideas, or theories. We have just utilised AI to enhance readability and refine the language. This was used with extreme human control and oversight. The authors take full responsibility for reviewing and approving the content.

References

[1] Alzahrani KA, Alzaid NA, Bakodah HO, Almazmumy MH. Computational approach to third-order non-linear boundary value problems via efficient decomposition shooting method. Axioms. 2024;13(4):248.

[2] Arefin MA, Nishu MA, Dhali MN, Uddin MH. Analysis of reliable solutions to the boundary value problems by using shooting method. Mathematical Problems in Engineering. 2022;2022:2895023.

[3] Edun IF, Akinlabi GO. Application of the shooting method for the solution of second-order boundary value problems. IOP Conference Series: Journal of Physics. 2021;1734(1):012020.

[4] Victor VS, Ettmüller M, Schmeißer A, Leitte H, Gramsch S. Machine learning based optimization workflow for tuning numerical settings of differential equation solvers for boundary value problems. arXiv [Preprint]. 2024 [cited 2024 16 April]. Available from: https://arxiv.org/abs/2404.10472

[5] Manyonge AW, Opiyo R, Kweyu D, Maremwa JS. Numerical solution of non-linear boundary value problems of ordinary differential equations using the shooting technique. Journal of Information Technology Education: Research. 2017;4(1):29-36.

[6] Sabharwal CL. An iterative hybrid algorithm for roots of non-linear equations. Engineering. 2021;2(1):80-98.

[7] Nwry AW, Kareem HM, Ibrahim RB, Mohammed M. Comparison between bisection, Newton, and secant methods for determining the root of the non-linear equation using MATLAB. Turkish Journal of Computer and Mathematics Education. 2021;12(14):1115-1122.

[8] Chen XD, Shi J, Ma W. A fast and robust method for computing real roots of non-linear equations. Applied Mathematics Letters. 2017;68:27-32.

[9] Dolan ED, Moré JJ. Benchmarking optimization software with performance profiles. Mathematical Programming. 2002;91(2):201-213.