Impression: Jurnal Teknologi dan Informas, Vol. 4 (2) (2025)
Published by: Lembaga Riset Ilmiah – YMMA Sumut
Impression: Jurnal Teknologi dan Informasi
Journal homepage: https://jurnal.risetilmiah.ac.id/index.php/jti

Monte Carlo Implementation on Bisection and Regula Falsi Methods for Finding Multiple Roots of Polynomial and Exponential Equations

Hery Andi Sitompul[1], Kristian Tarigan[2], Dewi Sholeha[3]
Universitas Darma Agung, Medan. Indonesia
Email:
herystpl@gmail.com, kristiantarigan50@gmail.com, alkhansadewi@gmail.com

ARTICLEINFO

Article history:
Received: 10 April 2025
Revised: 28 Juni 2025
Accepted: 15 Juli 2025

Keywords:
Bisection,
Regula Falsi,
Monte Carlo,
Nonlinear Equations

ABSTRACT

Nonlinear algorithms available in literature and developments carried out by scientists generally can only determine a single root of a nonlinear equation in a single calculation process. The development of the Bisection method and the Brent method can determine multiple roots of polynomial equations. By implementing the Monte Carlo method principle in the Bisection and Regula Falsi methods, it is obtained that multiple roots of a nonlinear equation in polynomial form or an equation in the form of a combination of polynomial and exponential can be done well in a single calculation process with accurate results.

Corresponding Author:
Hery Andy Sitompul
Universitas Darma Agung, Medan
herystpl@gmail.com

PENDAHULUAN

In various mathematical applications, both in science and engineering, we often encounter the equation f(x)=0, which is used to determine the value of a specific parameter, for example, the voltage value of a solar panel (M Rasheed, et al. 2020). The problem faced with this equation is determining its roots. If the equation is a quadratic equation, it is certainly easy to solve. If the equation is a polynomial of degree 3 or an exponential function, a sinusoidal function, or a combination of at least two of these, it will be a more challenging task.

Various methods for determining the solution to this equation are generally numerical and are widely found in the literature. Generally, these methods are divided into two groups: closed and open. Essentially, both methods can only produce a single root in a single calculation.

Scientists have developed methods for determining the roots of a nonlinear equation with more than one root. The Newton-Raphson and Secant methods can determine multiple roots of a nonlinear equation (Chapra, 2010). The modified Bisection method and the modified Brent method can determine multiple roots of a polynomial equation if the initial values selected for both methods clamp one of the roots (Batarius, 2021).

Aitken's Extrapolation Algorithm (AIM) and the Three-Step Method (ITM), both more recently developed methods, can only determine the roots of a single polynomial in a single process (M. Rasheed, et al. 2020). The modified Brent and Bisection methods can determine multiple roots of a polynomial equation, but not necessarily other forms of nonlinear equations (Batarius, 2021). The iterative q method can determine the roots of an equation of the $\boldsymbol{f(x) = xe^{x^2} - sin^2x + 3\cos x + 5}$, but only one root is obtained in a single calculation (Gul Sana, et al., 2021).

The Regula Falsi method generally has a faster convergence rate for determining the roots of third-degree polynomials, and only one root is obtained in a single calculation (Nur Hafidz, et al., 2023).
The Monte Carlo method has been widely used in statistics to predict the value of a probabilistic variable, where the solution to a problem is given based on calculations of the distribution of a particular random variable (Yusmaity, et al., 2019).

The Monte Carlo method approach can determine the exact solution to an integral problem (Halimovich, et al., 2022), and the Monte Carlo method approach can be used effectively to calculate the roots of an algebraic equation (Abadi & Bahnamriri, 2016). Implementation of the Monte Carlo method in the Bisection method can determine several roots of a non-linear equation (Etesami, et al, 2021).
With the development of modern computer technology, calculations using numerical methods that seemed impossible can now be done well (Temencan, et al, 2020). Therefore, in this study, a comparison will be made between the implementation of the Monte Carlo method in the Bisection method and the Regula Falsi method, which are closed methods for determining several roots of nonlinear equations in polynomial and non-polynomial forms using Matlab programming as a calculation tool.

URAIAN TEORI

A nonlinear equation is an equation in the form of a polynomial or exponential, sinusoidal, logarithmic, or a combination of at least two of these forms written in the form f(x)=0.
In numerical methods, the method for determining the roots of the equation is divided into closed methods, namely those that use two initial values (lower and upper limits) in the calculation process, including the Bisection and Regula Falsi methods, and open methods that use one initial value in the calculation process, including the Newton-Rapson and Secant methods (Chapra, 2010)

Bisection Method

Suppose x̅ is a root of a nonlinear equation f(x)=0 located in the interval [a,b] where f(a).f(b)<0.
Algorithm.
Define: $a_0 = a, b_0 = b, x_0 = \frac{1}{2}(a_0 + b_0)$
$n \geq 1$

If $f(a_{n-1}).f(x_{n-1}) < 0$, define $a_n = a_{n-1}, b_n = x_{n-1}, x_n = \frac{1}{2}(a_n + b_n)$.
If $f(a_{n-1}).f(x_{n-1}) > 0$, define $a_n = x_{n-1}, b_n = b_{n-1}, x_n = \frac{1}{2}(a_n + b_n)$.
If $f(a_{n-1}).f(x_{n-1}) = 0$, so $\bar{x} = x_{n-1}$ (Radi&Hami, 2018).

Falsi Regula Method

Suppose x̅ is the root of a nonlinear equation$f(x) = 0$ which lies in the interval [a,b] where $f(a).f(b) < 0$. Algoritma.
Defenisikan :

$$a_0 = a, b_0 = b, x_0 = b_0 - f(b_0)\frac{b_0 - a_0}{f(b_0) - f(a_0)},\ \text{and to } n \geq 1$$

If $f(a_{n-1}).f(x_{n-1}) < 0$, define $a_n = a_{n-1}, b_n = x_{n-1},\ x_n = b_n - f(b_n)\frac{b_n - a_n}{f(b_0) - f(a_n)}$
If $f(a_{n-1}).f(x_{n-1}) > 0$, define $a_n = x_{n-1}, b_n = b_{n-1},\ x_n = b_n - f(b_n)\frac{b_n - a_n}{f(b_0) - f(a_n)}$
If $f(a_{n-1}).f(x_{n-1}) = 0$, so $\bar{x} = x_{n-1}$ (Radi&Hami, 2018).

Monte Carlo Method

Monte Carlo methods are a class of computational algorithms that rely on recursive random sampling to compute related results. They are often used for simulations in computational physics, computational chemistry, and numerical integral calculations. They are well-suited to computer calculations because they perform iterative calculations using random numbers from a specific distribution. They are considered particularly suitable when the answer to a deterministic algorithm is impossible to prove (Abadi & Bahnamriri, 2016).

METODE PENELITIAN

This research was conducted quantitatively through an experimental process, taking two examples of nonlinear equations: a polynomial form with at least three real roots and a combination of at least two with an exponential, sinusoidal, or logarithmic form with at least three real roots.
The following are the steps taken in this research:
Determine the nonlinear equation whose real roots will be found, with at least three different real roots. Determine the same initial values for the Bisection and Regula Falsi methods. Then, apply/implement the Monte Carlo method to both the Bisection and Regula Falsi algorithms.
Obtain the results/roots of the nonlinear equations for both cases mentioned in step one. Then compare each example for the two methods used in terms of accuracy of the results and speed of convergence.

Determining the Roots of a Nonlinear Equation Using the Monte Carlo Method

Suppose f(x)=0 is a nonlinear equation that has at least 3 different real roots. The following are the steps used to determine the roots of the equation

Bisection Method.

- Generate random numbers from a uniform distribution for a given initial value of n.
- Determine the tolerance value to be used.
- Sort the random numbers by their position on the real number line $(x_1, x_2, \cdots, x_{n-1}, x_n)$.
- Take $[x_1, x_2]$ if $f(x_1).f(x_2) < 0$ use *Bisection algorithm* to determine the roots of the equation and continue the process with $[x_2, x_3]$, else continue with $[x_2, x_3]$. And so on until $[x_{n-1}, x_n]$.

Falsi Regula Method

- Generate random numbers from a uniform distribution for a given initial value of n.
- Determine the tolerance value to be used.
- Sort the random numbers by their position on the real number line $(x_1, x_2, \cdots, x_{n-1}, x_n)$.
- Take $[x_1, x_2]$ if $f(x_1).f(x_2) < 0$ use the Regula Falsi algorithm to find the roots of the equation and continue the process with $[x_2, x_3]$, else continue with $[x_2, x_3]$. And so on until $[x_{n-1}, x_n]$.

HASIL PENELITIAN

The equation used in this study is :

1. $x^4 - 2x^3 - 21x^2 + 22x + 40 = 0$
2. $x^2e^{x^2} + \frac{1}{4}xe^{x^2} - \frac{3}{8}e^{x^2} - 3x^2 - \frac{3}{4}x - \frac{9}{8} = 0$

The initial values used in equation 1 are [-10,10] and in equation 2 are [-2,2]. The tolerance value is ε=10^(-6) for both methods.
In the first case, 200 random numbers are generated for the interval [-10,10] with a uniform distribution, representing the Monte Carlo method. In the second case, 100 random numbers are generated from a uniform distribution.

The results obtained are:

Case 1

A visualization of the function graph $f(x) = x^4 - 2x^3 - 21x^2 + 22x + 4$, where the blue circle marks are the intersection points of the curve of the function with the x-axis or the exact roots of the equation in question. While figures 2 and 3 are visualizations of the equation graph along with the roots of the Monte Carlo Method implementation process in the Bisection algorithm marked with red circles and in the Regula Falsi algorithm marked with green circles. While the results of the root calculations from the two methods are presented in table 1 and as a comparison the exact root values are presented in table 2.

.

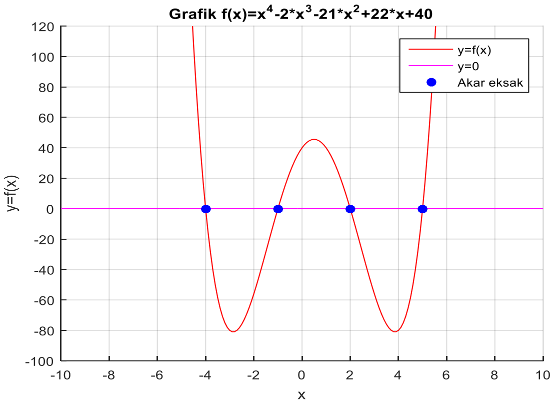

Figure1 . Grafik dari $f(x) = x^4 - 2x^3 - 21x^2 + 22x + 4$ dan akar eksak

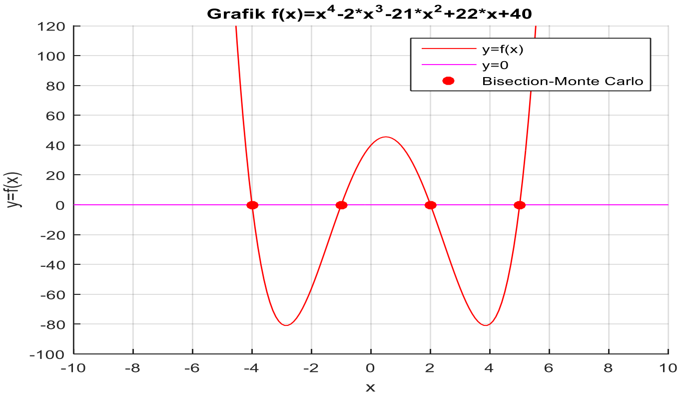

Figure 2 Grafik dari f $f(x) = x^4 - 2x^3 - 21x^2 + 22x + 4$ dan akar *Bisection-Monte Carlo*

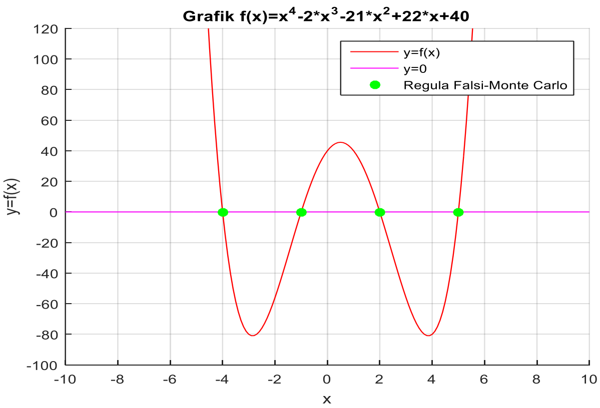

Figure 3. Grafik dari $f(x) = x^4 - 2x^3 - 21x^2 + 22x + 4$

dan akar *Regula Falsi-Monte Carlo*

Table 1. Comparison of the results of the Bisection Method and Regula Falsi by implementing the Monte Carlo Method for the equation $x^4 - 2x^3 - 21x^2 + 22x + 4 = 0$

Bisection			Regula Falsi		
Akar (x_i)	Iterasi	Galat	Akar (x_i)	Iterasi	Galat
-4.000005	15	1.0e-04 *0.053157	-3.999999	4	1.0e-07 *-.613198
-1.000016	18	1.0e-04 *0.110667	-0.999999	4	1.0e-07 *-.114407
1.999989	18	1.0e-04 *0.105010	1.999999	4	1.0e-07 *0.289994
4.999996	18	1.0e-04 *0.05042	4.999999	5	1.0e-07 *-.000126

Sumber : Data Processed Matlab R2015a, 2015

Table 2. Exact roots in case 1

Exact roots (x_i)			
-4	-1	2	5

Sumber : Data Processed Matlab R2015a, 2015

Case 2

A visualization of the function graph $f(x) = x^2e^{x^2} + \frac{1}{4}xe^{x^2} - \frac{3}{8}e^{x^2} - 3x^2 - \frac{3}{4}x - \frac{9}{8}$, where the blue circle marks are the points where the curve of the function intersects the x-axis or the exact roots of the equation in question. Figures 5 and 6 show the graphical visualization of the equation and its roots from the Monte Carlo implementation process in the Bisection algorithm and the Regula Falsi algorithm. The root calculations through the Monte Carlo implementation in both algorithms are given in Table 2

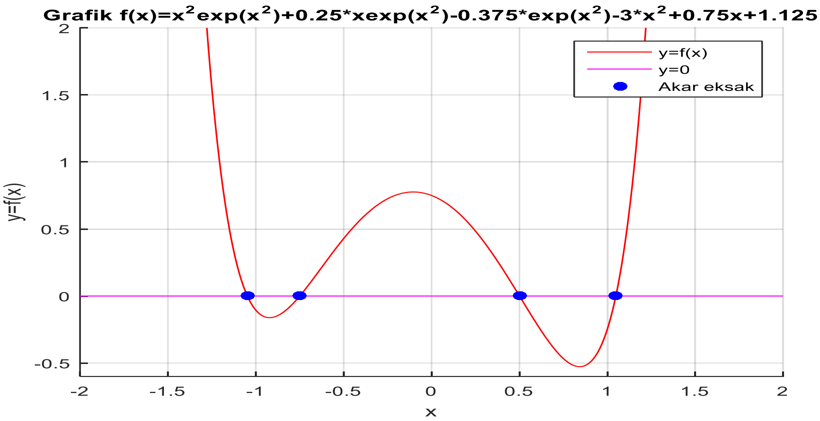

Figure 4. Graph of $f(x) = x^2e^{x^2} + \frac{1}{4}xe^{x^2} - \frac{3}{8}e^{x^2} - 3x^2 - \frac{3}{4}x - \frac{9}{8}$ and exact roots

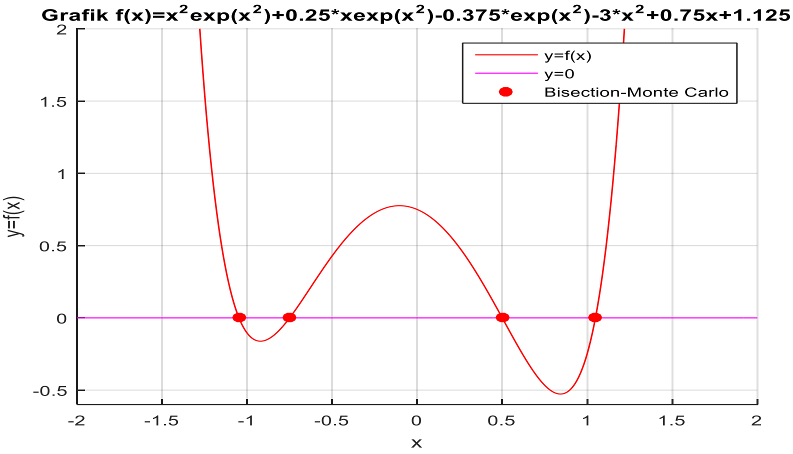

Figure 5. Graph of $f(x) = x^2e^{x^2} + \frac{1}{4}xe^{x^2} - \frac{3}{8}e^{x^2} - 3x^2 - \frac{3}{4}x - \frac{9}{8}$ and the roots of Bisection-Monte Carlo

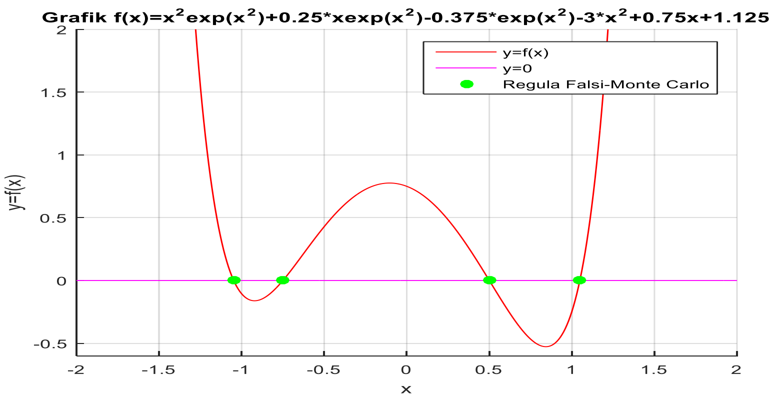

Figure 6 . Graph of $f(x) = x^2e^{x^2} + \frac{1}{4}xe^{x^2} - \frac{3}{8}e^{x^2} - 3x^2 - \frac{3}{4}x - \frac{9}{8}$ and the roots of the Regula Falsi-Monte Carlo

Table 3. Comparison of the results of the Bisection Method and the Regula Falsi by implementing the Monte Carlo Method for the equation x^2$e^{x^2} + \frac{1}{4}xe^{x^2} - \frac{3}{8}e^{x^2} - 3x^2 - \frac{3}{4}x - \frac{9}{8} = 0$

Bisection			*Regula Falsi*		
Akar(x_i)	Iterasi	Galat	Akar (x_i)	Iterasi	Galat
-1.048386	15	1.0e-03 * 0.284095	-1.049814	4	1.0e-04 * 0.466025
-0.75641	16	1.0e-03 * 0.634251	-0.749999	4	1.0e-04 * -0.005376
0.499978	14	1.0e-03 * 0.386096	0.499999	2	1.0e-04* -0.000007
1.048123	16	1.0e-03 * 0.110951	1.048146	5	1.0e-04 * -0.470744

Sumber : Data Processed Matlab R2015a, 2015

Table 4. Exact roots in case 2

Exact roots (x_i)			
-1.0841	-0.75	0.5	1.0481

Sumber : Data Processed Matlab R2015a, 2015

PEMBAHASAN

Based on the results obtained from the calculations using MATLAB above, it can be seen that in cases 1 to 4, the real roots of the equation can be approximated very well by the implementation of the Monte Carlo method on both algorithms used in this study. In the Bisection method, the number of iterations until the solution converges ranges between 15-18 iterations and the error is in a very small range, namely 4×10^{-6} sampai 16×10^{-6}. In the Falsi Regula Method, the number of iterations until the solution converges is 4 and 5 iterations, and the resulting error is in the range 1×10^{-6}.

In cases 2, the 4 real roots of the equation can also be approximated very well by implementing the Monte Carlo method in both algorithms. In the Bisection algorithm, the number of iterations until the solution converges is in the range of 14-16 iterations, and the resulting error is in the values $2{,}1 \times 10^{-5}$ to $6{,}4 \times 10^{-3}$. Meanwhile, in the Regula Falsi algorithm, the number of iterations required until the solution converges is in the range of 2-5 iterations, and the resulting error ranges from the value 1×10^{-6} to $4{,}5 \times 10^{-5}$.

PENUTUP

The Monte Carlo method implemented in the Bisection and Regula Falsi algorithms can accurately determine multiple roots of a nonlinear equation in polynomial form, as well as a combination of exponential and polynomial forms, in a single calculation. This can be concluded from the very small resulting error.

From the two case studies used in this study, it is clear that the convergence speed of the Regula Falsi method is much faster, in the range of 2-5 iterations, compared to the Bisectioan method, which is in the range of 14-18 iterations. Likewise, the error generated by the Monte Carlo method implemented in the Regula Falsi algorithm is much smaller than that of the Monte Carlo method implemented in the Bisection algorithm.

The methods used in this study are closed methods (Bisection and Regula Falsi), which use two initial values to pin down the roots of the equation. It would be beneficial to try them with open methods (Newton-Raphson and Secant), which use a single initial value.

REFERENSI

Bambang Agus Sulistyono, S. Sarmijo, Dian Devi Yohanie. 2024. *Penerapan Metode Regula Falsi untuk Meningkatkan Akurasi Perhitungan Mekanisme Pergeseran Piston,* Prosiding Seminar Nasional Matematika dan Pendidikan Matematika, Yogyakarta.

Bouchaib Radi, Abdelhalak El Hami. 2018. *Advanced Numerical Methods with Matlab*, hlm.4-17. John Wiley & Sons.

Chapra, C.S. & Canale, P.R. 2010. *Numerical Methods for Engineers*, Sixth Edition, hlm. 164-167. McGraw Hill Companies Inc.

Gizem Temelcam, Mustafa Sivri, Inci Albayrak.2020. *A New Iterative Linerazition for Solving Nonlinear Equation Syste,* hlm 47-54. IJOCTA International Journal of Optimization and Control: Theories& Aplication (http://doi.org/10.11121/ijocta.01.2020.00684).

Gul Sana, Pshtiwan Othman Mohammed, Dong Yun Shin, Muhammad Aslam Noor and Mohammad Salem Oudat.2021. *On Iterative Metods for Solving Nonlinear Equations in quantum Calculus*. hlm 1-17. fractal and fractional. (https://doi.org/10.3390/).

Mohammed Rassheed, Suha Shihab, Taha Rashid, Marwa Enneffati.2021. *Some Step Iterative Method for Finding Roots of Non Linear Equations.* hlm 95-102. Journal of Qadisiyah for Computer Science and Mathematics. Vol.13,No.1.

M Rasheed, S Shihab, O Y Mahammed and Aqeel Al-Adili.2020. *Parameters Estimation of Photovoltaic Model Using Nonlinear Algorithms.* ICMLAP Journals of Physics: Conference Series.(doi:10.1088/1742-6596/1795/1/012058).

Nur Hafidz, Moh Rofi, Adhi Pryanto. 2023. *Penggunaan Prosedur Regula Falsi dan Interval Bagi Dua Untuk Pencarian Akar Pada Fungsi Polinomial Berderajat Tiga*. SOSCIED Vol.6,No.1.

Patrisus Batarius. 2021. *Perbandingan Metode Brent dan Bisection dalam Penentuan Akar Ganda Persamaan Berbentuk Polinomial.* Prosiding Seminar Nasional Riset dan Teknologi Terapan (RITEKRA). Bandung.

Reza Etesami, Mohsen Madadi, Mashaallah Mashinchi, Reza Ashraf Ganjoei. 2021. *A New Method for Rooting equation Based On the Bisection Method*, hlm 1-7. MethodsX. (https://doi.org/10.1016/j.mex.2021.101502).

Tajiyev Tahirjon Halimovich, Khusanov Azizjon Abdurashhidovich, Samijonov Azizbek Ismailjon.2022. *Calculation Of The Exact Integral By The Monte Carlo Method.* hlm 38-47. IJSSR Journal, Vol.11, No.12.

Tri Sutrisno. 2023. *Aplikasi Penyelesaian Numerik Pencarian Akar Persamaan Non linier Dan Penerapannya Dalam Menyelesaikan Analisis Break Even Point*. hlm 37-49. Computatio: Journal of Computer Science and Information System, vol7,No.1.

Vahid Mirzaei Mahmoud Abadi, Shila Banari Bahnamriri.2016. *The Aproximate Calculation of The Roots of Algebraic Equation Through Monte Carlo Method.* hlm 64-68. International Journal of Mathematics and Computational science.

Yusmaity, Julius Santony, Yuhandri.2019. *Simulasi Metode Monte Carlo untuk Meprediksi Hasil Ujian Nasional (Studi Kasus di SMK 2 Pekanbaru)* . hlm 1-6, Jurnal Informasi dan Teknologi jidt.Vol.1, No.4.