Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control
Journal homepage: http://kinetik.umm.ac.id
ISSN: 2503-2267
Vol. 11, No. 3, August, Pp. 583-594

Accuracy comparison of multivariate Newton-Raphson, Newton-Kantorovich, and Levenberg–Marquardt methods for solving nonlinear systems using numerical simulation

Syaharuddin*[1], Hendi Hidayah[1], Vera Mandailina[1], Saba Mehmood[2], Wasim Raza[3]
Department of Mathematics Education, Muhammadiyah University of Mataram, Indonesia[1]
Department of Mathematics, University of Management and Technology, Pakistan[2]
Department of Mathematics, Universidade Federal Do Rio de Janeiro, Brazil[3]

Article Info
Keywords:
Numerical Methods, Multivariate Newton Raphson, Newton Kantorovich, Levenberg-Marquardt, Nonlinear Equation Systems

Article history:
Received: December 01, 2025
Accepted: March 10, 2026
Published: August 01, 2026

Cite:
S. Syaharuddin, H. Hidayah, V. Mandailina, S. Mehmood, and W. Raza, "Accuracy Comparison of Multivariate Newton-Raphson, Newton-Kantorovich, and Levenberg–Marquardt Methods for Solving Nonlinear Systems Using Numerical Simulation", *KINETIK*, vol. 11, no. 3, Aug. 2026.
https://doi.org/10.22219/kinetik.v11i3.2603

*Corresponding author.
Syaharuddin
E-mail address:
syaharuddin.ntb@gmail.com

Abstract
Multivariable nonlinear equation systems often appear in engineering, physics, economics, and artificial intelligence modeling, but often do not have closed analytical solutions. Therefore, accurate, efficient, and stable numerical methods are needed. This study aims to comparatively evaluate three iterative methods, namely Multivariate Newton-Raphson, Newton-Kantorovich, and Levenberg–Marquardt, in solving identical high-complexity multivariable nonlinear systems. Simulations were performed using MATLAB with an error tolerance of 0.001 and a maximum iteration limit of 100. The test system consisted of a combination of trigonometric, exponential, and polynomial functions, resulting in nonlinear interactions that were challenging for each method. The simulation results show that Levenberg–Marquardt excelled with only 6 iterations and a final error of 3.246×10^{-10}, indicating high stability and efficiency, followed by Multivariate Newton-Raphson with 13 iterations and an error of 4.606×10^{-9}, while Newton-Kantorovich requires 27 iterations with an error of 5.770×10^{-7}, reflecting slower semi-local corrections. Three-dimensional visualization shows the intersection point of the surface as a solution, providing an intuitive understanding of the iteration trajectory characteristics of each method. The novelty of this research lies in the integrated numerical simulation framework that allows direct quantitative comparison of the three methods on identical systems with the same initial conditions, tolerance, and iteration limits. These findings provide important empirical references for selecting efficient and stable iterative methods for multivariable nonlinear systems, as well as practical guidance for numerical applications in engineering, physics, and scientific computing.

1. Introduction
A nonlinear equation system is a set of equations involving variables with nonlinear relationships, such as power, exponential, logarithmic, and trigonometric functions. Such systems are commonly found in various fields, including applied mathematics, physics, engineering, economics, and artificial intelligence [1]. In various modern applications such as engineering optimization, physical system modeling, and intelligent computing, nonlinear systems are often large-scale and complex, requiring stable and efficient numerical methods. Since they generally do not have closed-form analytical solutions, their solution requires effective numerical approaches to obtain accurate approximate solutions [2]. In recent years, the increasing complexity of computational models in science and engineering has further strengthened the need for iterative methods that are not only accurate but also computationally efficient and stable against initial value variations, especially in high-dimensional and ill-conditioned systems that are sensitive to numerical disturbances [3].

One of the most widely used numerical methods is the multivariate Newton–Raphson method, which is a generalization of the single-variable Newton–Raphson method for solving systems with more than one variable [4]. This method works by linearizing nonlinear functions through first-order Taylor series around the starting point, then solving the linear system iteratively using a Jacobian matrix containing partial derivatives [5]. The speed and accuracy of convergence are greatly influenced by the accuracy of the Jacobian calculation and the selection of initial values that are close enough to the actual roots. Although known to be fast and efficient, this method is sensitive to initial conditions and has the potential to diverge in highly complex or discontinuous systems, so in large-scale computing practice, it often requires additional initialization or modification strategies to maintain convergence stability [6].

In addition to these methods, another approach that is also used is the Newton–Kantorovich method, an extension of the Newton method formulated based on operator theory and functional analysis [7]. Unlike Newton–Raphson, which is generally applied in finite-dimensional spaces, Newton–Kantorovich utilizes a Banach space-based approach and Fréchet operators so that it can be applied to systems in infinite function spaces [8]. This approach provides local convergence guarantees under certain conditions and is suitable for handling more complex nonlinear system structures [9]. However, this method requires explicit operator formulations and in practice often requires a larger number of iterations [10]. Although it has a stronger theoretical foundation, computational implementation challenges remain an important aspect, particularly regarding operator evaluation and the formation of Fréchet derivatives, which are computationally more expensive than the classical Jacobian approach [11][12].

Another widely used method is the Levenberg–Marquardt method, which combines the characteristics of the Gauss–Newton method and the gradient descent method [13]. This method was developed to improve stability while accelerating convergence in solving nonlinear equation systems, especially when the initial iteration value is not close to the root of the solution [14]. By introducing a damping factor, this algorithm is able to adaptively adjust its iterative behavior: acting like gradient descent when far from the solution to maintain stability, and resembling Gauss–Newton when approaching the solution to accelerate convergence [15]. This flexibility makes the Levenberg–Marquardt method more robust in dealing with systems with complex structures and unstable numerical conditions [16], and it is widely applied in optimization and computational modeling due to its ability to balance accuracy and computational efficiency, especially in problems involving least-squares-based nonlinear objective functions [17].

Various experimental and computational studies show that the multivariate Newton–Raphson, Newton–Kantorovich, and Levenberg–Marquardt methods make significant contributions to solving nonlinear equation systems [18][19][20]. The multivariate Newton–Raphson method generally converges quickly, especially for medium to high-dimensional systems, with very small final errors, but it is sensitive to the choice of initial values and Jacobian conditions [21][22]. Newton–Kantorovich is more stable, with a wider semilocal convergence domain of around 35–45%, making it more tolerant to variations in the initial point and complex system structures [23][24]. Meanwhile, Levenberg–Marquardt stands out for its adaptive nature; this method is able to balance stability and convergence speed, reducing errors by 70–85% in the initial iterations, and maintaining good performance in ill-conditioned systems [25][26]. These findings confirm that each method has different theoretical and computational characteristics, so the choice of method is greatly influenced by the system structure, numerical conditions, and initial value configuration. To date, there is no integrated evaluation framework that allows direct quantitative comparison of the three methods under identical computational conditions [27].

However, most previous studies evaluated the Newton–Raphson, Newton–Kantorovich, and Levenberg–Marquardt methods separately or in different experimental configurations, so direct comparisons on identical multivariable nonlinear equation systems are still limited. Differences in test system structure, tolerance parameters, and initial values have the potential to cause interpretation bias, making it difficult to objectively determine the most effective method. Therefore, this study aims to conduct a systematic comparative analysis between Multivariate Newton–Raphson, Newton–Kantorovich, and Levenberg–Marquardt using the same nonlinear equation system, with uniform computational conditions, identical initial values, equivalent error tolerances, and consistent maximum iteration limits. The novelty of this research lies in the application of an integrated numerical simulation framework that enables fair, standardized, and replicable quantitative evaluation under identical computational conditions, which has not been available in previous literature. With this approach, the research is expected to provide a stronger empirical contribution to the numerical computation literature while also serving as a methodological basis for selecting the most appropriate iterative method for solving multivariable nonlinear systems in various scientific and engineering applications.

2. Research Method

This study uses a quantitative approach through numerical simulation to compare the accuracy levels of the multivariate Newton-Raphson, Newton-Kantorovich, and Levenberg–Marquardt methods in solving nonlinear equation systems. The type of research used is a comparative study with a numerical experimental design, in which the three methods are systematically implemented on the same nonlinear equation system [28]. The equation system used consists of three multivariable nonlinear functions, involving trigonometric, exponential, and polynomial elements, to represent the complexity of high-dimensional nonlinear systems. The algorithms were implemented using the of MATLAB software (version 2018), with evaluation metrics including: (1) absolute error value against the exact solution, (2) number of iterations until convergence, and (3) computation time. The results of each method were analyzed descriptively and quantitatively to assess the level of the accuracy and numerical efficiency produced [10]. All simulation and data processing procedures were conducted under controlled computational settings to ensure objective and reproducible comparisons.

2.1 Multivariate Newton-Raphson Method

The Multivariate Newton-Raphson Method is one of the iterative techniques commonly used to solve multivariable nonlinear equation systems, namely equation systems involving more than one variable with nonlinear relationships [29]. This method is based on the linearization of nonlinear functions through first-order Taylor series around the starting point [30]. The iterative process involves the Jacobian matrix $J(x)$, which contains the partial derivatives of each function with respect to each variable, as shown in Equation (1):

$$J(\boldsymbol{x}) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}. \tag{1}$$

The iterative steps of this method can be expressed as Equation 2:

$$\mathrm{x}^{(k+1)} = \mathrm{x}^{(k)} - J^{-1}\left(\mathrm{x}^{(k)}\right) \cdot \mathrm{F}\left(\mathrm{x}^{(k)}\right) \tag{2}$$

In this Equation 1 and 2, $x^{(k)}$is the approximation vector of the solution at iteration k, for example $x^{(k)} = [x^{(k)}, y^{(k)}, z^{(k)}]^T$. $F(x^{(k)}) = [f_1(x^{(k)}), f_2(x^{(k)}), f_3(x^{(k)})]^T$ is the evaluation of the function at point $x^{(k)}$, and $J^{-1}\left(x^{(k)}\right)$ is the inverse of the Jacobian matrix at point $x^{(k)}$, which is used to calculate the direction correction and the size of the iterative step so that the solution approaches the exact root. The difference $x^{(k+1)} - x^{(k)}$ represents the correction vector applied to the current approximation. The advantage of this method lies in its ability to adaptively adjust the direction of the solution search at each iteration, so that it tends to converge quickly if the initial guess is close enough to the solution [31]. However, the computational load is high because it is necessary to calculate and invert the Jacobian matrix at each iteration.

2.2 Newton-Kantorovich Method

The Newton-Kantorovich method is a variation of Newton-Raphson that uses a similar linear approximation but with tighter convergence in theory [32]. Unlike Newton-Raphson, the Jacobian is only calculated once at the starting point $\mathrm{x}^{(0)}$ and is used throughout the iteration, as formulated in Equation 3:

$$\mathrm{x}^{(k+1)} = \mathrm{x}^{(k)} - J^{-1}\left(\mathrm{x}^{(0)}\right) \cdot \mathrm{F}\left(\mathrm{x}^{(k)}\right) \tag{3}$$

Here, $x^{(k)}$ is the approximate solution at iteration $k, J\left(x^{(0)}\right)$ is the Jacobian calculated once at the starting point and not updated during iteration, while $F\left(x^{(k)}\right)$ is still calculated at the current approximation point so that the iteration still adjusts the correction to the estimated solution position. The use of a fixed Jacobian reduces the computational load, but makes this method more sensitive to the level of nonlinearity of the function and the choice of initial values. The iterative step correction may not be as accurate as Newton-Raphson, so the risk of failed convergence is higher, especially if the function is highly nonlinear or the starting point is far from the solution [33].

2.3 Levenberg–Marquardt Method

The Levenberg–Marquardt method is an iterative algorithm that combines the characteristics of the Gauss–Newton method and gradient descent [34]. Its purpose is to improve stability and accelerate convergence, especially when the initial iteration value is not close to the solution. This algorithm is designed to minimize the sum of squares of a nonlinear function, which can be mathematically expressed as shown in Equation 4:

$$\min_x \parallel f(x) \parallel^2 = i = \sum_{i=1}^{m} fi(x)^2 \tag{4}$$

In the context of solving multivariable nonlinear systems of equations, the parameter vector $x \in \mathbb{R}^n$ represents the variables whose values are to be found, while the function vector $f(x) = [f_1(x), f_2(x), \ldots, f_m(x)]^T$ is a set of nonlinear functions that form a system of equations, where each $f_i(x)$ is one equation in the system. The function vector is written in column form to maintain consistency with the mathematical representation in numerical analysis. The Levenberg–Marquardt method combines the convergence speed of the Gauss–Newton method with the stability of gradient descent through the use of a damping factor λ, allowing the algorithm to adaptively adjust its iterative behavior when dealing with complex functions, ill-conditioned conditions, or when the starting point is far from the solution root. This approach results in a more stable and efficient iterative process for obtaining solutions to multivariable nonlinear systems of equations [35]. The Levenberg–Marquardt update incorporates a damping factor to balance the

Gauss-Newton and gradient descent approaches. The solution correction at each iteration is determined by Equation 5:

$$(J^T J + \lambda I)\,\Delta x = -J^T f(x) \tag{5}$$

Where J is the Jacobian matrix containing the partial derivatives of each function $f_i(x)$ with respect to variable $x_j, \lambda > 0$ is the damping factor that regulates the stability of the solution correction a large value of λ makes the method more similar to gradient descent (stable but slow), while a small value of λ makes the method more similar to Gauss–Newton (fast when close to the solution). The identity matrix I is used to stabilize the linear system, and Δx is the correction value of the variable at each iteration. This adaptive approach makes Levenberg–Marquardt very reliable for solving multivariable nonlinear systems with complex function structures or initial conditions that are far from the solution [27].

2.4 Nonlinear Equation System

The simulation in this study involves three multivariable nonlinear equations, which are formulated as shown in Equation 6:

$$Nonlinear\ Equations: \begin{cases} f1(x,y,z) = sin(x) + cos(y) + 0.4z - 1 = 0 \\ f2(x,y,z) = e^{0.3y} + 0.2x - z^2 - 1 = 0 \\ f3(x,y,z) = x^2 - y^2 + 0.5z^2 - 1 = 0 \end{cases} \tag{6}$$

The simulation in this study uses three multivariable nonlinear equations that combine trigonometric, exponential, and polynomial elements, resulting in complex nonlinear interactions between variables. The diversity of these types of functions allows for testing the convergence and accuracy of iterative methods in solving multivariable nonlinear equation systems [36]. The initial values (x_0, y_0, z_0) are determined by mapping the graph of each function to identify areas where the function value approaches zero, so that iteration starts from a point close to the expected solution and increases the likelihood of convergence.

2.5 Research Procedure

This study followed a series of systematically arranged stages to achieve its objectives. Each step in the process was designed to ensure the accuracy and reliability of the results obtained. The complete research procedure is illustrated in Figure 1.

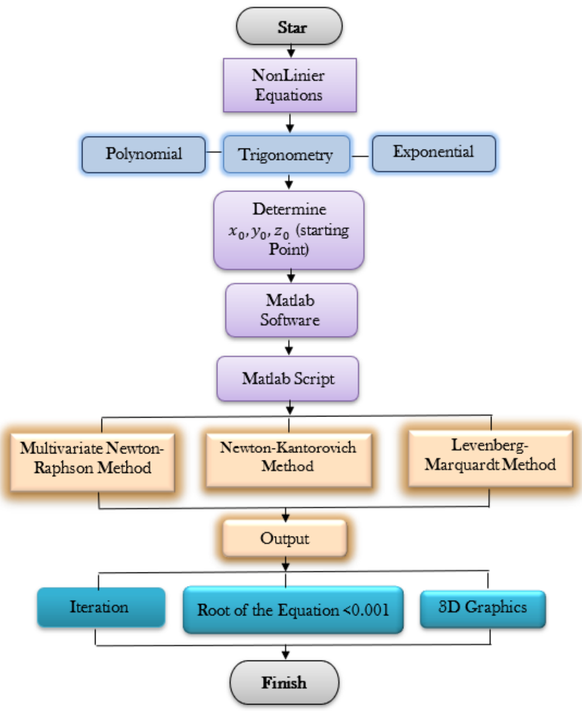

Figure 1. Procedure of Research

Figure 1 shows the systematic stages in the research process aimed at solving nonlinear systems of equations through a MATLAB-based numerical approach. The research began with the development of algorithms from three iterative methods, namely the Multivariate Newton-Raphson Method, the Newton-Kantorovich Method, and the Levenberg–Marquardt Method, based on the results of a literature study, which were then implemented in the form of MATLAB scripts. The nonlinear equation system used as the object of study includes polynomial, trigonometric, and exponential components. The graphs of each function are used to determine the initial values of the variables $x_0 = 1.0$, $y_0 = 0.5$, and $z_0 = 0.2$ as the starting points of the iterative process. The simulation was carried out by directly inputting the three functions into each method, using an error tolerance of 0.001, a maximum iteration limit of 100, and the specified initial values. The iterative process ran until the solution obtained met the convergence criteria. The simulation results were observed based on the number of iterations, the root values found, the final error, and the stability of the numerical results during the iteration process. Next, the results are visualized in the form of a 3D graph as an indicator of the success of the convergence method. All of these stages reflect a strategy for solving numerical problems in nonlinear equation systems efficiently and systematically with the help of MATLAB software. This approach ensures that iterations have a good chance of convergence, numerical errors are controlled, and simulation results can be analyzed objectively to compare the performance of the three iterative methods [37].

3. Results and Discussion

This section presents the results of the implementation and performance analysis of three numerical methods in solving multivariable nonlinear systems of equations using MATLAB, namely Multivariate Newton–Raphson, Newton–Kantorovich, and Levenberg–Marquardt. Simulations were conducted to compare the effectiveness of each method based on the number of iterations, final error value, iterative stability, and convergence pattern towards the solution. Each method was applied to the same nonlinear equation system with uniform conditions to ensure an objective performance comparison. Numerical results are presented in tables, while visualization of the iteration trajectory provides an understanding of the convergence dynamics and iterative behavior characteristics of each method. These findings are then analyzed to highlight quantitative and qualitative performance differences, as well as to compare them with previous research findings.

3.1 Implementation of MATLAB Script Methods

The implementation was carried out by executing MATLAB scripts for each numerical method used, namely Multivariate Newton–Raphson, Newton–Kantorovich, and Levenberg–Marquardt. Each script ran an iterative process in solving the multivariable nonlinear equation system. To maintain objectivity in the comparison, all methods were executed with the same initial conditions, namely $x0$ =1.0, $y0$ = 0.5, and $z0$ = 0.2, an error tolerance of 0.001, and a maximum iteration limit of 100. Each method was simulated three times to ensure consistency of results. From these simulations, the number of iterations, final error values, and numerical solutions achieved were obtained as the basis for evaluating the relative effectiveness of each method. The MATLAB script implementation is presented in Table 1.

Table 1. MATLAB Implementation Scripts for Each Numerical Method

<table>
<tr><th>No</th><th>Method</th><th>Script MATLAB</th></tr>
<tr><td>1.</td><td>Multivariate Newton-Raphson</td><td><pre>for k = 1:max_iter
    F_val = F_func(xk);
    J_val = J_func(xk);

    delta = -J_val \ F_val;
    xk_next = xk + delta;

    err = norm(delta);

    fprintf('%d\t\t%.6f\t%.6f\t%.6f\t%.6e\n', k, xk_next(1),
xk_next(2), xk_next(3), err);

    if err < tol
        break;
    end
    xk = xk_next;
end</pre></td></tr>
<tr><td>2.</td><td>Newton-Kantorovich</td><td><pre>for k = 1:max_iter
    x_val = X(1); y_val = X(2); z_val = X(3);
    F_val = F(x_val,y_val,z_val);
    J_val = J(x_val,y_val,z_val);</pre></td></tr>
</table>

	(continued)

```
    delta = -J_val\F_val;
    X = X + delta;

    err = norm(delta, inf);
    fprintf('%d\t\t%.6f\t%.6f\t%.6f\t%.6e\n', k, X(1), X(2), X(3),
err);

    if any(abs(X) > 1e6) || any(isnan(X)) || any(isinf(X))
        fprintf('Solution diverged or invalid. Iteration
stopped.\n');
        break;
    end

    if err < tol
        fprintf('Converged in %d iterations.\n', k);
        fprintf('Final solution: x=%.6f, y=%.6f, z=%.6f\n', X(1),
X(2), X(3));
        break;
    end
```

Levenberg-Marquardt

```
for k = 1:maxIter
    F_val = F(X);
    J_val = J(X);

    % LM update: (J^T J + ?I) ? = -J^T F
    delta = -(J_val' * J_val + lambda * eye(3)) \ (J_val' * F_val);
    X_new = X + delta;

    % Evaluate new function
    F_new = F(X_new);

    % Adaptive lambda rule
    if norm(F_new) < norm(F_val)
        lambda = lambda / 10;
        X = X_new;
    else
        lambda = lambda * 10;
    end

    % Error
    err = norm(delta, inf);
    fprintf('%d\t%.6f\t%.6f\t%.6f\t%.6e\n', k, X(1), X(2), X(3),
err);

    % Convergence check
    if err < tol
        fprintf('\nConverged in %d iterations.\n', k);
        fprintf('Solution: x=%.6f, y=%.6f, z=%.6f\n', X(1), X(2),
X(3));
        break;
    end
```

The three scripts demonstrate the solution update mechanism through correction vector calculations based on functions and Jacobian matrices. The Multivariate Newton–Raphson and Newton–Kantorovich methods use a system linearization approach through Jacobian to determine the iteration correction direction, while Levenberg–Marquardt adds a damping parameter (λ) to improve stability when Jacobian conditions are unfavorable or approaching singularity [37]. These differences in update strategies not only affect convergence speed, but also the level of numerical stability with respect to initial conditions and the characteristics of the nonlinear system being solved, thus forming the basis for a comparative analysis of the efficiency, robustness, and sensitivity of each method.

3.2 Simulation Results

After conducting three simulations for each method, comparative data was obtained representing the performance of Multivariate Newton–Raphson, Newton–Kantorovich, and Levenberg–Marquardt in solving the tested nonlinear equation system. The simulation results include the number of iterations required to achieve convergence, the final numerical solution, and the final error value obtained for each method. A summary of the computational results is presented in Table 2.

Table 2. Simulation Results

No	Method	Iteration	X	Y	Z	Error
1	Multivariate Newton-Raphson	13	1.452	1.121	-0.854	4.606×10^{-9}
2	Newton-Kantorovich	27	2.654	1.215	-0874	5.770×10^{-7}
3	Levenberg-Marquardt	6	1.362	1.163	-0841	3.246×10^{-10}

Based on Table 2, the Multivariate Newton–Raphson method converged in 13 iterations. The numerical solution obtained was x = 1.452, y = 1.121, and z = -0.854 with a final error value of 4.606×10^{-9}. This very small error value indicates that this method is capable of producing a root approximation of the system with a high degree of precision and good iterative stability. The Newton–Kantorovich method requires 27 iterations to reach convergence. The final solution obtained is were x = 2.654, y = 1.215, and z = -0.874, with a final error of 5.770×10^{-7}. Although the error value is still within the specified tolerance limit, the relatively higher number of iterations indicates that the process of adjusting the correction direction to the nonlinear characteristics of the system is more gradual than other methods. Meanwhile, the Levenberg–Marquardt method showed the most efficient performance with only 6 iterations. The numerical solution obtained was x = 1.362, y = 1.163, and z = -0.841, with a final error value of 3.246×10^{-10}, which was the smallest error among the three methods. These results indicate that the adaptive damping mechanism used is capable of maintaining stability while accelerating the convergence process towards the root of the system.

Overall, based on the number of iterations and the level of solution precision, the Levenberg–Marquardt method showed the best performance, followed by Multivariate Newton–Raphson, while Newton–Kantorovich required a relatively longer iterative process. This difference in performance reflects the influence of the Jacobian update strategy and the correction mechanism on the convergence dynamics of the multivariate nonlinear equation system [38].

3.3 Convergence Visualization

As a continuation of the implementation process of three numerical methods in solving multivariable nonlinear equation systems, a graphical visualization is presented that illustrates the iteration path towards the solution for each method, namely Multivariate Newton–Raphson, Newton–Kantorovich, and Levenberg–Marquardt. This visualization aims to provide a more comprehensive understanding of the dynamics of root search and convergence behavior characteristics in three-dimensional space. The graphical representation displays the intersection points of three function surfaces as the solution to the system, while also showing the differences in the solution search patterns produced by each method. The visualization results can be observed in Figure 2, Figure 3, and Figure 4.

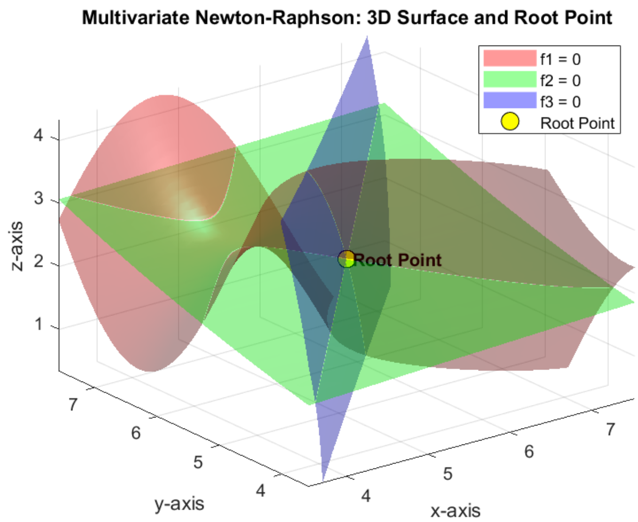

Figure 2. Visualization of the Multivariate Newton-Raphson Method Solution

Figure 2 shows a three-dimensional visualization of the solution to a nonlinear system of equations using the Multivariate Newton–Raphson method. The three surfaces, colored red, green, and blue, represent the sets of points that satisfy the equations $f_1(x,y,z)=0, f_2(x,y,z)=0$, and $f_3(x,y,z)=0$, respectively. The root points of the system are indicated as the intersection points of the three surfaces. The graphics show that the surfaces have uneven contours with significant slope changes in the $-x$ and $-z$ axes, requiring an iterative method capable of dynamically adjusting the direction of the solution search. Multivariate Newton–Raphson, which updates the Jacobian at each step, produces a relatively direct iteration path and precision following the surface geometry. This pattern reflects the method's ability to respond quickly and directionally to changes in local gradients [21], [31].

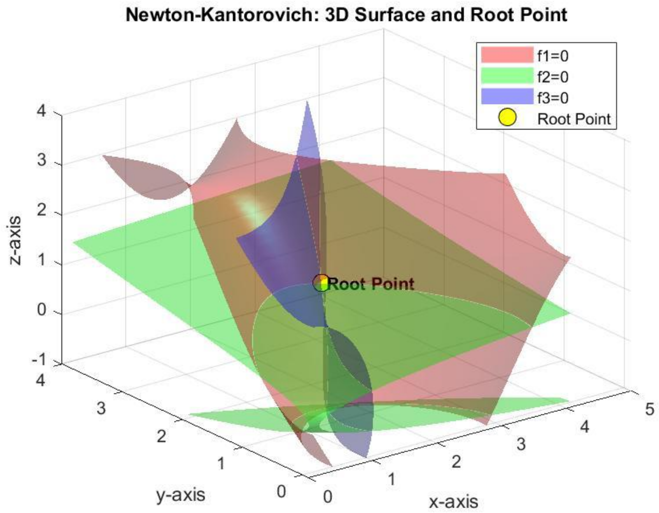

Figure 3. Visualization of the Newton-Kantorovich Method Solution

Figure 3 shows a three-dimensional visualization of the solution to a nonlinear system using the Newton–Kantorovich method, where the red, green, and blue surfaces represent the solution sets of $f_1(x,y,z) = 0, f_2\,(x,y,z) = 0$, and $f_3(x,y,z) = 0$, with their intersection point representing the root of the system. The surfaces appear to have sharp gradient changes, especially around the $-z$ axis area, so that the path to the root requires direction adjustments that are sensitive to contour changes. Unlike Newton–Raphson, the Newton–Kantorovich method uses a Jacobian approach that is fixed or semi-fixed at certain intervals, so that the iteration path does not appear to lead directly to the intersection point. The search path tends to be more winding and undergoes several directional corrections before approaching the solution. This pattern reflects the semi-local convergence characteristics of the Newton–Kantorovich method, which emphasize stability over rapid directional adjustment [33].

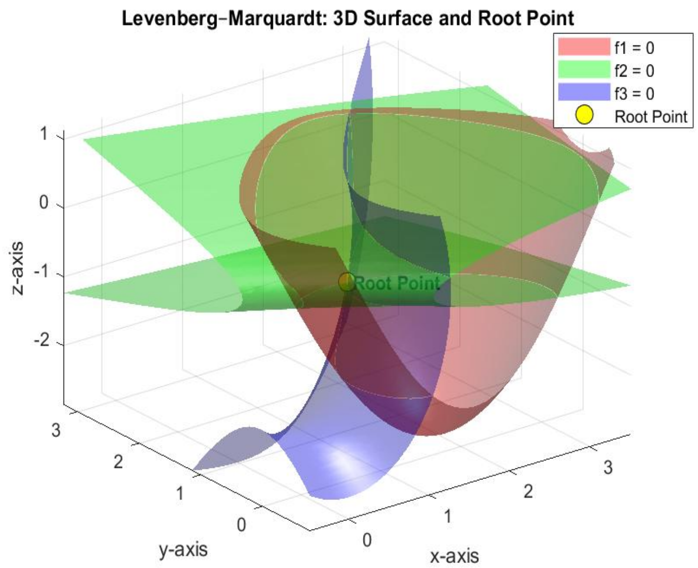

Figure 4. Visualization of the Levenberg–Marquardt Method Solution

Figure 4 shows a three-dimensional visualization of the solution to a nonlinear system of equations using the Levenberg–Marquardt method. The three red, green, and blue surfaces represent the solution sets of $f_1(x,y,z) = 0, f_2(x,y,z) = 0$, and $f_3(x,y,z) = 0$, respectively, with the intersection points as the roots of the system. The curved and sharply changing surface contours, especially along the $-z$ axis, illustrate the complexity of the nonlinear structure that must be traced. Through the adaptive damping mechanism, the iteration direction becomes more stable and controlled in following gradient changes. The search path appears more directed with minimal deviation, indicating that the combination strategy between the Gauss–Newton and gradient descent approaches is able to maintain a balance between stability and convergence speed on surfaces with high curvature [25].

Overall, the visualizations in Figures 2, 3, and 4 reinforce the analytical interpretation of the iteration trajectory characteristics of each method. The more direct trajectory patterns reflect adaptive gradient responses, while the more winding trajectories indicate a gradual correction process. Thus, this graphical analysis not only serves as a visual illustration, but also provides qualitative validation of the convergence dynamics of each method in handling multivariable nonlinear systems.

3.4 Discussion

The results show that the differences in performance between the three methods are mainly determined by the Jacobian update strategy and iterative correction mechanism used. Levenberg–Marquardt was the most superior method with 6 iterations and a final error of 3.246×10^{-10}, which was the best value in all test scenarios. This superiority stems from the adaptive damping mechanism that dynamically controls the transition between the gradient descent and Gauss–Newton approaches [13]. This strategy keeps the method stable in the early stages of iteration and accelerates convergence when approaching the root. These results not only demonstrate higher numerical efficiency but are also consistent with findings [38] and [39] stating that parameter-adaptive Newton modifications can improve convergence rates without sacrificing stability. Compared to the other two methods in an identical system and under the same computational conditions, Levenberg–Marquardt shows the most optimal balance between speed and precision.

Multivariate Newton-Raphson shows competitive performance with 13 iterations and an error of 4.606×10^{-9}. Theoretically, its quadratic convergence provides significant acceleration when the Jacobian is well-defined and the initial value is close enough to the solution. This advantage is in line with reports [24] and [40], which show 62–84% greater efficiency compared to the Secant and Steffensen methods, and is supported by [41], which note an average speed 64% higher than Fixed-Point Iteration at a tolerance of 0.001. However, in the context of nonlinear systems with complex function interactions, the absence of an adaptive damping mechanism makes this method more sensitive to extreme curvature and the potential for singular Jacobian conditions. This explains why the final precision is still below that of Levenberg–Marquardt despite theoretically having a quadratic convergence rate.

Newton–Kantorovich showed the highest number of iterations, namely 27 iterations with an error of 5.770×10^{-7}. Analytically, the semi-local approach used provides broader convergence guarantees, but causes the gradient correction to take place gradually, thereby reducing efficiency. This finding is consistent with [42], [43], [44], which report a need for 30–50% additional iterations compared to Newton-Raphson to achieve the same tolerance. In relatively smooth and moderately dimensional test systems, this theoretical advantage does not result in significant computational gains. Thus, this method is more relevant in cases with highly sensitive function structures or when theoretical stability takes priority over convergence speed.

Overall, the results of this study show that the advantages of Levenberg–Marquardt are not merely minor numerical differences, but rather a direct consequence of its adaptive algorithmic design. The evaluation was conducted within an integrated simulation framework with identical initial conditions, tolerances, and iteration limits, allowing for an objective comparison free from configuration bias. This approach provides significant empirical contributions because most previous studies compared methods in different configurations. Therefore, these findings have practical value in determining the most efficient and stable iterative method for multivariable nonlinear equation systems in modern numerical computation.

4. Conclusion

Based on the results of numerical simulations and computational analysis, Levenberg–Marquardt proved to be the most accurate and efficient method for solving multivariable nonlinear equation systems, as demonstrated by the smallest number of iterations and the smallest final error compared to the other two methods. Newton–Raphson Multivariate showed competitive performance when the Jacobian matrix was consistently updated and the initial value was close enough to the root of the system, while Newton–Kantorovich required more iterations due to the semi-local approach used, despite offering broader theoretical stability guarantees. The main contribution of this research lies in the comparative evaluation under identical initial conditions, error tolerances, and iteration limits, resulting in an objective and controlled comparison of accuracy and efficiency. These findings provide a more robust empirical basis for selecting iterative methods for multivariable nonlinear systems. This research is limited to medium-dimensional systems with relatively smooth functions and initial values close to roots; therefore, further studies are needed on larger-dimensional systems, non-smooth functions, and more extreme initial conditions to test the robustness and scalability of the method more comprehensively.

References

[1] M. Shakeel, S. T. Mohyud-Din, and M. A. Iqbal, "Modified extended exp-function method for a system of nonlinear partial differential equations defined by seismic sea waves," *Pramana - J. Phys.*, vol. 91, no. 2, pp. 1–8, 2018. https://doi.org/10.1007/s12043-018-1601-6

[2] J. Ríos-Ocampo and M. S. Gary, "Using analytical equations to represent nonlinear relationships," *Syst. Dyn. Rev.*, vol. 38, no. 4, pp. 354–370, 2022. https://doi.org/10.1002/sdr.1718

[3] L. Zakaria, A. Eka, and D. Aziz, "Penyelesaian Sistem Persamaan Fully Fuzzy Non Linear Menggunakan Metode Newton Raphson Ganda," *J. Math. Theory Appl.*, vol. 5, no. 2, pp. 67–73, 2023. https://doi.org/10.31605/jomta.v5i2.2876

[4] M. Putri and S. Syaharuddin, "Implementations of Open and Closed Method Numerically: A Non-linear Equations Solution Convergence Test," *IJECA (International J. Educ. Curric. Appl.*, vol. 2, no. 2, p. 1, 2019. https://doi.org/10.31764/ijeca.v2i2.2041

[5] P. Batarius, J. San Juan, and P. Kupang, "Perbandingan Metode Newton-Raphson Modifikasi Dan Metode Secant Modifikasi Dalam Penentuan Akar Persamaan," *Semin. Nas. Ris. dan Teknol. Terap.*, vol. 8, no. 2, pp. 53–63, 2018. https://doi.org/10.21067/pmej.v1i3.2784

[6] M. Adriana, "on the Convergence of the Newton-Raphson Method and Some of Its Generalizations," *Bull. Transilv. Univ. Brasov, Ser. III Math. Comput. Sci.*, vol. 4, no. 2, pp. 215–224, 2024. https://doi.org/10.31926/but.mif.2024.4.66.2.13

[7] Inderjeet and R. Bhardwaj, "A new Iterative Newton Raphson technique for the numerical simulation of Nonlinear Equations," *J. Integr. Sci. Technol.*, vol. 13, no. 4, pp. 1–6, 2025. https://doi.org/10.62110/sciencein.jist.2025.v13.1080

[8] R. Behl, "Highly efficient family of iterative methods for solving nonlinear models," *J. Comput. Appl. Math.*, vol. 346, no. 2, pp. 110–132, 2019. https://doi.org/10.1016/j.cam.2018.06.042

[9] M. Mohammad Ali, "Numerical differential continuation approach for systems of nonlinear equations with singular Jacobian," *AUT J. Math. Comput.*, vol. 3, no. 1, pp. 53–58, 2022. https://doi.org/10.22060/ajmc.2021.20487.1068

[10] A. Hasanudin, "Calculation of Loan Amount When Cooperatives Do Not Make Profits with Non-Linear Equation Method Using Secant and Its Implementation with MATLAB," *J. Penelit. Mat. dan Pendidik. Mat.*, vol. 8, no. 1, pp. 17–22, 2023. https://doi.org/10.26486/jm.v8i1.4020

[11] S. Regmi, I. K. Argyros, S. George, and M. I. Argyros, "Developments on the Convergence Analysis of Newton-Kantorovich Method for Solving Nonlinear Equations," *Eur. J. Math. Anal.*, vol. 3, no. 11, p. 15, 2023. https://doi.org/10.28924/ada/ma.3.15

[12] A. Heuermann, P. Hannebohm, M. Schäfer, and B. Bachmann, "Accelerating the simulation of equation-based models by replacing non-linear algebraic loops with error-controlled machine learning surrogates," *Proc. 15th Int. Model. Conf.*, vol. 204, no. 11, pp. 275–284, 2023. https://doi.org/10.3384/ecp204275

[13] L. Zheng, L. Chen, and Y. Ma, "A variant of the Levenberg-Marquardt method with adaptive parameters for systems of nonlinear equations," *Mathematics*, vol. 7, no. 1, pp. 1241–1256, 2021. https://doi.org/10.3934/math.2022073

[14] J. Ritonga and D. Suryana, "Perbandingan Kecepatan Konvergensi Akar Persamaan Non Linier Metode Titik Tetap dengan Metode Newton Raphson Menggunakan Matlab," *Inf. (Jurnal Inform. dan Sist. Informasi)*, vol. 11, no. 2, pp. 51–64, 2019. https://doi.org/10.37424/informasi.v11i2.17

[15] Y. Han and S. Rui, "SS symmetry A New Adaptive Levenberg – Marquardt Method for Nonlinear Equations and Its Convergence Rate under the Hölderian Local," *symmetry S Artic.*, vol. 16, no. 6, pp. 1–19, 2024. https://doi.org/10.3390/sym16060674

[16] A. K. Venkatesan and S. K. Natarajan, "Stability Enhancement of PV Powered Microgrid using Levenberg-Marquardt Algorithm Based Intelligent Controller Under Grid-connected Mode," *Distrib. Gener. Altern. Energy J.*, vol. 37, no. 2, pp. 361–380, 2021. https://doi.org/10.13052/dgaej2156-3306.37214

[17] H. Mustafidah, "Performance of Levenberg-Marquardt Algorithm in Backpropagation Network Based on the Number of Neurons in Hidden Layers and Learning Rate," *JUITA J. Inform. e-ISSN*, vol. 8, no. 1, pp. 29–35, 2020. https://doi.org/10.30595/juita.v8i1.7150

[18] Yudhi, Devitriani, Mariatul Kiftiah, "Analisis Metode Newton-Raphson Ganda Orde Konvergensi Empat Dalam Menyelesaikan Sistem Persamaan Nonlinear," *Bimaster Bul. Ilm. Mat. Stat. dan Ter.*, vol. 8, no. 2, pp. 213–220, 2019. https://doi.org/10.26418/bbimst.v8i2.31648

[19] E. Sharma, S. Panday, S. K. Mittal, D. M. Joița, L. L. Pruteanu, and L. Jäntschi, "Derivative-Free Families of With- and Without-Memory Iterative Methods for Solving Nonlinear Equations and Their Engineering Applications," *Mathematics*, vol. 11, no. 21, pp. 1–13, 2023. https://doi.org/10.3390/math11214512

[20] O. Paulina Maure and H. Tulan, "Studi Komparasi Beberapa Metode Numerik Dalam Mengaproksimasi Akar-Akar Persamaan Non Linear," *Asimtot J. Kependidikan Mat.*, vol. 6, no. 01, pp. 1–12, 2024. https://doi.org/10.30822/asimtot.v6i01.4008

[21] N. K. Vitanov, "Simple Equations Method (SEsM): An Effective Algorithm for Obtaining Exact Solutions of Nonlinear Differential Equations," *Entropy*, vol. 24, no. 11, pp. 1–55, 2022. https://doi.org/10.3390/e24111653

[22] H. A. H. Abugirda, K. S. Al-Yasiri, and M. K. Abdullah, "Newton-Kantorovich Method for Solving One of the Non-Linear Sturm-Liouville Problems," *Baghdad Sci. J.*, vol. 20, no. 3, pp. 2036–2041, 2023. https://doi.org/10.21123/bsj.2023.7543

[23] S. Regmi, I. K. Argyros, S. George, and J. Warden, "A Unified Kantorovich-type Convergence Analysis of Newton-like Methods for Solving Generalized Equations under the Aubin Property," *Eur. J. Math. Anal.*, vol. 4, no. 3, pp. 1-10. 2024. https://doi.org/10.28924/ada/ma.4.3

[24] A. G. Kamel, E. H. Haraz, and S. N. Hanna, "Numerical Simulation of Channel Flow Over a Skewed Equilateral Cavity," *J. Appl. Math. Comput. Mech.*, vol. 19, no. 3, pp. 29–43, 2020. https://doi.org/10.17512/jamcm.2020.3.03

[25] M. Herlina, F. Tukan, O. P. Maure, K. Trisnawati, and D. Ina, "Analisis keakuratan metode numerik dalam menyelesaikan turunan persamaan nonlinier," *Leibniz J. Mat.*, vol. 04, no. 2, pp. 10–22, 2024. https://doi.org/10.59632/leibniz.v4i02.446

[26] A. Fischer, A. F. Izmailov, and M. V Solodov, "The Levenberg – Marquardt method : An overview of modern convergence theories and more," *Comput. Optim. Appl.*, vol. 89, no. 1, pp. 33–42, 2020. https://doi.org/10.1007/s10589-024-00589-1

[27] N. U. Qadir, "Influence of Principal Component Analysis as a Data Conditioning Approach for Training Multilayer Feedforward Neural Networks with Exact Form of Levenberg-Marquardt Algorithm," *Global Journal of Technology & Optimization.*, vol. 11, no. 1, pp. 1–13, 2020. https://doi.org/10.37421/gjto.2020.11.239

[28] S. Regmi, I. K. Argyros, S. George, and J. Warden, "A Unified Kantorovich-type Convergence Analysis of Newton-like Methods for Solving Generalized Equations under the Aubin Property," *Eur. J. Math. Anal.*, vol. 4, no. 3, pp. 1–10, 2024. https://doi.org/10.28924/ada/ma.4.3

[29] F. Ahmad *et al.*, "A preconditioned iterative method for solving systems of nonlinear equations having unknown multiplicity," *Algorithms*, vol. 10, no. 1, pp. 1–9, 2017. https://doi.org/10.3390/a10010017

[30] M. Farman, A. Akgül, N. Alshaikh, M. Azeem, and J. Asad, "Fractional-Order Newton–Raphson Method for Nonlinear Equation With Convergence and Stability Analyses," *Fractals*, vol. 31, no. 10, pp. 1–12, 2023. https://doi.org/10.1142/S0218348X23400790

[31] E. Sunandar and I. Indrianto, "Perbandingan Metode Newton-Raphson & Metode Secant Untuk Mencari Akar Persamaan Dalam Sistem Persamaan Non-Linier," *J. Pengkaj. dan Penerapan Tek. Inform.*, vol. 13, no. 1, pp. 72–79, 2020. https://doi.org/10.33322/petir.v13i1.893

[32] T. Tuan, "Composition Structure Of Polyconvolution Associated With Index Kontorovich-Lebedev Transform And Fourier Integrals," *Res. Artic.*, vol. 10, no. 2, pp. 1–16, 2025. https://doi.org/10.1080/10652469.2025.2536167

[33] E. Weinan, J. Han, and A. Jentzen, "Deep Learning-Based Numerical Methods for High-Dimensional Parabolic Partial Differential Equations and Backward Stochastic Differential Equations," *Commun. Math. Stat.*, vol. 5, no. 4, pp. 349–380, 2017. https://doi.org/10.1007/s40304-017-0117-6

[34] A. O. Umar, "On damping parameters of Levenberg-Marquardt algorithm for nonlinear least square problems On damping parameters of Levenberg-Marquardt algorithm for nonlinear least square problems," *Electrochem. Soc.*, vol. 20, no. 1, pp. 1–10, 2021. https://doi.org/10.1088/1742-6596/1734/1/012018

[35] M. A. S. Tabbu and A. Makkawaru, "Prediction of Sediment Yield Using the Algoritma Prediction of Sediment Yield Using the Algoritma Lavenberg-Marquardt," *J. Phys. Conf. Ser. Pap.*, vol. 21, no. 1, pp. 1–8, 2021. DOI: 10.1088/1742-6596/2123/1/012037

[36] J. R. Sharma and H. Arora, "Improved Newton-like methods for solving systems of nonlinear equations," *SeMA J.*, vol. 74, no. 2, pp. 147–163, 2017. https://doi.org/10.1515/nleng-2017-0105

[37] F. Ahmad, "Multi-step derivative-free preconditioned Newton method for solving systems of nonlinear equations," *SeMA J.*, vol. 75, no. 1, pp. 45–56, 2018, DOI: 10.1007/s40324-017-0120-6

[38] V. Mandailina, D. Pramita, M. Ibrahim, and H. R. Perwira, "Desimal : Jurnal Matematika Wilkinson Polynomials : Accuracy Analysis Based on Numerical Methods of the Taylor Series Derivative," *Desimal J. Mat.*, vol. 3, no. 2, pp. 155–160, 2020. https://doi.org/10.24042/djm.v3i2.6134

[39] S. J. A. Sumarauw and U. N. Manado, "Algoritma Pelatihan Levenberg-Marquardt Backpropagation Artificial Neural Network Untuk Data Time Series," pp. 213–222, 2019. https://doi.org/10.30595/juita.v7i1.4396

[40] H. Ali, T. Datta, and M. Kamrujjaman, "Efficient Family of Iterative Methods for Solving Nonlinear Simultaneous Equations: A Comparative Study," *J. Appl. Math. Comput.*, vol. 5, no. 4, pp. 331–337, 2021. https://doi.org/10.26855/jamc.2021.12.011

[41] A. F. Kirani, R. Ibnas, and I. Syata, "Solusi Numerik Sistem Persamaan Non Linear Dalam Reaksi Steam Reforming Menggunakan Metode Newton-Rhapson [Numerical Solution of Non-Linear Equation System in Steam Reforming Reaction Using Newthon-Raphson Method]," *Al-Aqlu J. Mat. Tek. dan Sains*, vol. 3, no. 1, pp. 43–55, 2025. https://doi.org/10.59896/aqlu.v3i1.105

[42] Y. Mumtazi, N. Hikmah, and S. Prayitno, "Comparison of Steffensen and Secant Methods in Determining Non- Linear Function Roots Using Hypertext Preprocessor (PHP) Programmer," *J. Pendidikan, Sains, Geol. dan Geofis.*, vol. 5, no. 3, pp. 418–424, 2024. https://doi.org/10.29303/goescienceed.v5i3.393

[43] Rahmatullah, "Perbandingan Metode Newton Raphson dan Metode Steffensen dalam Penentuan Akar Fungsi Non-Linier Menggunakan Pemrograman," *J. Ilm. Pendidik. Dasar*, vol. 10, no. 2, pp. 300–312, 2025. https://doi.org/10.23969/jp.v10i02.26961

[44] H. Bin Jebreen, H. Wang, and Y. Chalco-Cano, "Efficient Newton-Type Solvers Based on for Finding the Solution of Nonlinear Algebraic Problems," *Axioms*, vol. 13, no. 12, p. 880, 2024. https://doi.org/10.3390/axioms13120880