
Faculty of Science

Faculty Publications

A New Application of Gauss Quadrature Method for Solving Systems of Nonlinear Equations

Hari M. Srivastava, Javed Iqbal, Muhammad Arif, Alamgir Khan, Yusif S. Gasimov, & Ronnason Chinram

March 2021

© 2021 Hari M. Srivastava et al. This is an open access article distributed under the terms of the Creative Commons Attribution License. <https://creativecommons.org/licenses/by/4.0/>

This article was originally published at:



<https://doi.org/10.3390/sym13030432>

Citation for this paper:

Srivastava, H. M., Iqbal, J., Arif, M., Khan, A., Gasimov, Y. S., & Chinram, R. (2021). A New Application of Gauss Quadrature Method for Solving Systems of Nonlinear Equations. *Symmetry*, 13(3), 1-12. <https://doi.org/10.3390/sym13030432>.

Article

A New Application of Gauss Quadrature Method for Solving Systems of Nonlinear Equations

Hari M. Srivastava ^{1,2,3} , Javed Iqbal ⁴, Muhammad Arif ⁴ , Alamgir Khan ⁴, Yusif S. Gasimov ³ and Ronnason Chinram ^{5,*}

¹ Department of Mathematics and Statistics, University of Victoria, Victoria, BC V8W 3R4, Canada; harimsri@math.uvic.ca

² Department of Medical Research, China Medical University Hospital, China Medical University, Taichung 40402, Taiwan

³ Department of Mathematics and Informatics, Azerbaijan University, 71 Jeyhun Hajibeyli Street, Baku AZ1007, Azerbaijan; yusif.gasimov@au.edu.az

⁴ Department of Mathematics, Abdul Wali Khan University, Mardan 23200, KPK, Pakistan; javedmath@awkum.edu.pk (J.I.); marifmaths@awkum.edu.pk (M.A.); alamgir.khan@awkum.edu.pk (A.K.)

⁵ Division of Computational Science, Faculty of Science, Prince of Songkla University, Hat Yai, Songkhla 90110, Thailand

* Correspondence: ronnason.c@psu.ac.th

Abstract: In this paper, we introduce a new three-step Newton method for solving a system of nonlinear equations. This new method based on Gauss quadrature rule has sixth order of convergence (with $n = 3$). The proposed method solves nonlinear boundary-value problems and integral equations in few iterations with good accuracy. Numerical comparison shows that the new method is remarkably effective for solving systems of nonlinear equations.

Keywords: nonlinear equations; gauss quadrature formula; ordinary differential equation (ODE); error equations; sixth-order convergence; numerical examples



Citation: Srivastava, H.M.; Iqbal, J.; Arif, M.; Khan, A.; Gasimov, Y.S.; Chinram, R. A New Application of Gauss Quadrature Method for Solving Systems of Nonlinear Equations. *Symmetry* **2021**, *13*, 432. <https://doi.org/10.3390/sym13030432>

Academic Editor: Sun Young Cho

Received: 26 January 2021

Accepted: 1 March 2021

Published: 7 March 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In numerical analysis and other branches of scientific interests, solving a system of nonlinear equations by means of computational methods has always been very well motivated and convincing for researchers. For a system of nonlinear equations:

$$P(S) = (p_1(S), p_2(S), \dots, p_n(S))^T = 0, \quad (1)$$

where $S = (s_1, s_2, \dots, s_n)^T$ and $P : D \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^n$ is a nonlinear system, and $p_i, i = 1, 2, \dots, n : D \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$ is a nonlinear mapping. The solution of the nonlinear system of equations in (1) may be defined as the process of finding a vector $S^* = (s_1^*, s_2^*, \dots, s_n^*)^T$ such that $P(S^*) = 0$. The classical Newton method is one of the most commonly used iterative methods:

$$S^{(k+1)} = S^{(k)} - P'(S^{(k)})^{-1}P(S^{(k)}) \quad (k = 0, 1, 2, \dots),$$

where $P'(S^{(k)})$ is the Jacobian matrix of the nonlinear function $P(S)$ in the k th iteration at the point $S^{(k)}$ (see [1–3]). Newton's method quadratically converges to the solution S^* if the function P is continuous and differentiable. In recent years, several methods have been developed to analyze the solution of systems of nonlinear equations to improve interaction by using the quadrature formulas and fractional iterative method in the literature (see [4–12]).

In particular, Codero and Torregrosa [9] developed the third-order Newton–Simpson method as follows:

$$S^{(k+1)} = S^{(k)} - 6 \left[P'(S^{(k)}) + 4P' \left(\frac{S^{(k)} + Y^{(k)}}{2} \right) + P'(Y^{(k)}) \right]^{-1} P(S^{(k)}), \tag{2}$$

and the Open Newton method:

$$S^{(k+1)} = S^{(k)} - 3 \left[2P' \left(\frac{S^{(k)} + 3Z^{(k)}}{4} \right) - P' \left(\frac{S^{(k)} + Z^{(k)}}{2} \right) + 2P' \left(\frac{3S^{(k)} + Z^{(k)}}{4} \right) \right]^{-1} P(S^{(k)}), \tag{3}$$

where $Z^{(k)}$ represents the Newton approximation. Khirallah and Hafiz [13] suggested a cubically convergent method using the four-point Newton–Cotes formula for solving systems of nonlinear equations as follows:

$$S^{(k+1)} = S^{(k)} - 8 \left[P'(S^{(k)}) + 3P' \left(\frac{2S^{(k)} + Z^{(k)}}{2} \right) + 3P' \left(\frac{S^{(k)} + 2Z^{(k)}}{2} \right) + P'(Z^{(k)}) \right]^{-1} P(S^{(k)}). \tag{4}$$

The quadrature rule is used to approximate the definite integral of a function. The general form of a quadrature rule is given by [14]

$$Q = \int_a^b v(r)s(r)dr \simeq Q_m = \sum_{i=0}^m w_i s(r_i)$$

where $v(r)$ is a weight function, $w_i, i = 0, 1, 2, \dots, m$ are coefficients (weights). r_i are points of the rule and s is a given function integrable on the interval $[a, b]$ with the weight function v .

Motivated and inspired by the research going on in this area, we have introduced a new iterative for solving nonlinear equations. Several numerical examples are considered to show the effectiveness of the proposed method. The new iterative method shows the compatibility of numerical results with the scheme’s theoretical analysis. We have solved nonlinear boundary-value problems by using the proposed method. Our method gives better results than the other methods and converges more rapidly to the solution. Section 5 concludes the paper.

2. Three-Step Newton Method

Let $P : D \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^n$, be s -times Fréchet differentiable function on a convex set $D \subseteq \mathbb{R}^n$. Using the *Mean-Value Theorem* of multi-variable vectors function $P(S^{(k)})$ (see [1]), we have

$$P(S) - P(S^{(k)}) = \int_0^1 P'(S^{(k)} + r(S - S^{(k)}))(S - S^{(k)})dr. \tag{5}$$

Using the left rectangular rule, the right-hand side of (5) can be written as

$$\int_0^1 P'(S^{(k)} + r(S - S^{(k)}))(S - S^{(k)})dr \cong P'(S^{(k)})(S - S^{(k)}). \tag{6}$$

From (5) and (6), we get

$$S = S^{(k)} - P'(S^{(k)})^{-1}P(S^{(k)}) \quad (k = 0, 1, 2, \dots). \tag{7}$$

Replacing S by $S^{(k+1)}$ in (7), we get the Newton method. Using (5) and different numerical integration formulas, one can obtain different iterative methods such as (2), (3),

and (4). To develop the new iterative method, we approximate the integral in (5) by the following three-point Gauss Legendre integration formula:

$$\int_x^y f(t)dt \approx \frac{y-x}{9} \left[4f\left(\frac{y+x}{2}\right) + \frac{5}{2}f\left((y-x)\left(-\sqrt{\frac{3}{5}}\right) + \frac{y+x}{2}\right) + \frac{5}{2}f\left((y-x)\left(\sqrt{\frac{3}{5}}\right) + \frac{y+x}{2}\right) \right], \quad (8)$$

Thus, from (5) and (8), we have

$$\int_0^1 P'(S^{(k)} + r(S - S^{(k)}))(S - S^{(k)})dr \approx \frac{S - S^{(k)}}{9} \left[4P'\left(\frac{S + S^{(k)}}{2}\right) + \frac{5}{2}P'\left((S - S^{(k)})\left(-\sqrt{\frac{3}{5}}\right) + \frac{S + S^{(k)}}{2}\right) + \frac{5}{2}P'\left((S - S^{(k)})\left(\sqrt{\frac{3}{5}}\right) + \frac{S + S^{(k)}}{2}\right) \right]. \quad (9)$$

Moreover, from (1), (5), and (9), we get

$$0 \approx P(S^{(k)}) + \frac{S - S^{(k)}}{9} \left[4P'\left(\frac{S + S^{(k)}}{2}\right) + \frac{5}{2}P'\left((S - S^{(k)})\left(-\sqrt{\frac{3}{5}}\right) + \frac{S + S^{(k)}}{2}\right) + \frac{5}{2}P'\left((S - S^{(k)})\left(\sqrt{\frac{3}{5}}\right) + \frac{S + S^{(k)}}{2}\right) \right]. \quad (10)$$

From (10), the iterative scheme is given by

$$S \approx S^{(k)} - 9 \left[4P'\left(\frac{S + S^{(k)}}{2}\right) + \frac{5}{2}P'\left((S - S^{(k)})\left(-\sqrt{\frac{3}{5}}\right) + \frac{S + S^{(k)}}{2}\right) + \frac{5}{2}P'\left((S - S^{(k)})\left(\sqrt{\frac{3}{5}}\right) + \frac{S + S^{(k)}}{2}\right) \right]^{-1} P(S^{(k)}). \quad (11)$$

Subsequently, we use the k th iteration $T^{(k)}$ and $Z^{(k)}$ of Newton's method to replace S and $S^{(k)}$ respectively on the right-hand side of (11) and obtain a new iterative scheme as follows:

Algorithm 1: Three-Step Newton Method

Step 1: Select an initial guess $S^{(0)} \in \mathbb{R}^n$ and start k from 0.

Step 2: Compute

$$S^{(k+1)} = T^{(k)} - 9 \left[4P'(H^{(k)}) + \frac{5}{2}P'(H^{(k)} + W^{(k)}) + \frac{5}{2}P'(H^{(k)} + J^{(k)}) \right]^{-1} P(T^{(k)}).$$

Step 3: Set for the next step:

$$H^{(k)} = \left(\frac{T^{(k)} + Z^{(k)}}{2} \right),$$

$$J^{(k)} = (T^{(k)} - Z^{(k)}) \left(\sqrt{\frac{3}{5}} \right)$$

and

$$W^{(k)} = (T^{(k)} - Z^{(k)}) \left(-\sqrt{\frac{3}{5}} \right).$$

Step 4: Compute

$$Z^{(k)} = S^{(k)} - P'(S^{(k)})^{-1} P(S^{(k)}) \quad (k = 0, 1, 2, \dots)$$

$$T^{(k)} = Z^{(k)} - P'(Z^{(k)})^{-1} P(Z^{(k)}) \quad (k = 0, 1, 2, \dots).$$

Step 5: If $\|S^{(k+1)} - S^{(k)}\| < \epsilon$, then stop; otherwise. put $k = k + 1$ and go to Step 2.

In the next section, we discuss the convergence of the proposed method.

3. Convergence Analysis

In the following theorem, we prove the convergence of the proposed method.

Theorem 1. Suppose that the function $P : U \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^n$ is sufficiently Fréchet differentiable at each point of an open convex neighborhood U of the solution $S^* \in \mathbb{R}^n$ of (1). Assume also that $P'(S)$ is continuous and nonsingular at $S = S^*$. Then, the sequence $\{S^{(k)}\}$ generated by Algorithm 5 converges to S^* with the sixth order of convergence and the error equation is given by

$$e_{k+1} = 3C_2^5 e_k^6 + O(e_k^7), \quad (12)$$

where

$$e_k = S^{(k)} - S^*.$$

Proof. Let $P : D \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^n$ be s -times Fréchet differentiable in U . Then, by using the usual notation for the m th derivative of P at $v \in \mathbb{R}^n$, the m -linear function $P^{(m)}(v) : \mathbb{R}^n \times \dots \times \mathbb{R}^n \rightarrow \mathbb{R}^n$ is such that $P^{(m)}(v)(u_1, u_2, \dots, u_n) \in \mathbb{R}^n$. Suppose now that $S^* + h \in \mathbb{R}^n$ lies in the neighborhood of S^* . The Taylor polynomial for $P(S^* + h)$ can be of the form:

$$P(S^* + h) = P'(S^*) \left(h + \sum_{m=2}^{f-1} C_m h^m \right) + O(h^f), \quad (13)$$

where

$$C_m = \left(\frac{1}{m!} \right) [P'(S^*)]^{-1} P^{(m)}(S^*) \quad (m \geq 2).$$

We observe that $C_m h^m \in \mathbb{R}^n$, since

$$P^{(m)}(S^*) \in \mathcal{L} \in (\mathbb{R}^n \times \mathbb{R}^n, \dots, \mathbb{R}^n) \quad \text{and} \quad [P'(S^*)]^{-1} \in \mathcal{L} \in \mathbb{R}^n.$$

In addition, we can express P' as follows:

$$P'(S^* + h) = P'(S^*) \left(I + \sum_{m=2}^{f-1} mC_m h^{m-1} \right) + O(h^f), \tag{14}$$

where $I \in \mathbb{R}^{n \times n}$ is the identity matrix. We note that $mC_m h^{m-1} \in \mathcal{L} \in \mathbb{R}^n$. From (13) and (14), we get

$$P(S^{(k)}) = P'(S^*) [e_k + C_2 e_k^2 + C_3 e_k^3 + C_4 e_k^4 + C_5 e_k^5 + C_6 e_k^6 + \dots]. \tag{15}$$

$$P'(S^{(k)}) = P'(S^*) [I + 2C_2 e_k + 3C_3 e_k^2 + 4C_4 e_k^3 + 5C_5 e_k^4 + 6C_5 e_k^5 + \dots], \tag{16}$$

where

$$C_k = \left(\frac{1}{k!} \right) [P'(S^*)]^{-1} P^{(k)}(S^*) \quad (k = 2, 3, 4, \dots)$$

and

$$e_k = S^{(k)} - S^*.$$

From (15), we have

$$P'(S^{(k)})^{-1} = [I - 2C_2 e_k + (4C_2^2 - 3C_3) e_k^2 + (-4C_4 + 6C_2 C_3 + 6C_3 C_2 - 8C_2^3) e_k^3 + (16C_2^4 - 36C_2^2 C_3 + 16C_2 C_4 + 9C_3^2 - 5C_5) e_k^4 + \dots] (P'(S^*))^{-1}. \tag{17}$$

By multiplying $P'(S^{(k)})^{-1}$ and $P(S^{(k)})$, we obtain

$$P'(S^{(k)})^{-1} P(S^{(k)}) = e_k - C_2 e_k^2 + 2(C_2^2 - C_3) e_k^3 + (-4C_2^3 + 4C_2 C_3 + 3C_3 C_2 - 3C_4) e_k^4 + (8C_2^4 - 20C_2^2 C_3 + 6C_3^2 + 10C_2 C_4 - 4C_5) e_k^5 + (-16C_2^5 + 52C_2^3 C_3 - 33C_2 C_3^2 - 28C_2^2 C_4 + 17C_3 C_4 + 13C_2 C_5 - 5C_6) e_k^6 + \dots. \tag{18}$$

Taylor's series expansion of $P(Z^{(k)})$ is given by

$$P(Z^{(k)}) = P'(S^*) [(Z^{(k)} - S^*) + C_2 (Z^{(k)} - S^*)^2 + C_3 (Z^{(k)} - S^*)^3 + C_4 (Z^{(k)} - S^*)^4 - C_5 (Z^{(k)} - S^*)^5 + C_6 (Z^{(k)} - S^*)^6 + \dots], \tag{19}$$

where

$$C_k = \left(\frac{1}{k!} \right) [P'(S^*)]^{-1} P^{(k)}(S^*) \quad (k = 2, 3, 4, \dots).$$

Moreover, $Z^{(k)}$ can be written as follows:

$$Z^{(k)} = S^* + C_2 e_k^2 - 2(C_2^2 - C_3) e_k^3 - (-4C_2^3 + 7C_2 C_3 - 3C_4) e_k^4 + (4C_5 - 12C_2^4 + 24C_2^2 C_3 - 10C_2 C_4 - 6C_3^2) e_k^5 + (16C_2^5 - 52C_2^3 C_3 + 28C_2^2 C_4 + 33C_2 C_3^2 - 13C_2 C_5 - 17C_3 C_4 + 5C_6) e_k^6 + \dots. \tag{20}$$

Taylor's series expansion of $P'(Z^{(k)})$ is given by

$$P'(Z^{(k)}) = P'(S^*) [I + 2C_2 (Z^{(k)} - S^*) + 3C_3 (Z^{(k)} - S^*)^2 + 4C_4 (Z^{(k)} - S^*)^3 + 5C_5 (Z^{(k)} - S^*)^4] + O(Z^{(k)} - S^*)^5. \tag{21}$$

Putting (20) in (21), we have

$$\begin{aligned}
 P'(S^{(k)}) = P'(S^*) [& I + 2C_2^2 e_k^2 - 4C_2(C_2^2 - C_3)e_k^3 - C_2(-8C_2^3 + 11C_2C_3 - 6C_4)e_k^4 \\
 & + (-16C_2^5 + 28C_2^3C_3 - 20C_2^2C_4 + 8C_2C_5)e_k^5 + (32C_2^6 - 68C_2^4C_3 \\
 & + 60C_2^3C_4 - 26C_2^2C_5 - 16C_2C_6 - 3C_4 + 12C_3^3 + 10C_2] C_6 e_k^6 \\
 & + \dots .
 \end{aligned}
 \tag{22}$$

Upon multiplying $P'(Z^{(k)})^{-1}$ by $P(Z^{(k)})$, we have

$$\begin{aligned}
 P'(Z^{(k)})^{-1}P(Z^{(k)}) = & C_2e_k^2 + 2(-C_2^2 + C_3)e_k^3 + (3C_2^3 - 7C_2C_3 + 3C_4)e_k^4 \\
 & + (-4C_2^4 + 16C_2^2C_3 - 10C_2C_4 - 6C_3^2 + 4C_5)e_k^5 \\
 & + (6C_2^5 - 32C_2^3C_3 + 22C_2^2C_4 + 29C_2C_3^2 - 13C_2C_5 \\
 & - 17C_3C_4 + 5C_6)e_k^6 + \dots .
 \end{aligned}
 \tag{23}$$

The expression for $T^{(k)}$ is given below:

$$\begin{aligned}
 T^{(k)} = S^* + & 3C_2^3e_k^4 - (16C_2^4 - 20C_2^2C_3 + 6C_3^2 + 10C_2C_4C_5)e_k^5 \\
 & - (-24C_2^5 + 96C_2^3C_3 - 41C_2C_3^2 - 40C_2^2C_4 - 12C_2^5 + 17C_3C_4 \\
 & + 13C_2C_5 + 8C_2^6 - 5C_6)e_k^6 + \dots .
 \end{aligned}
 \tag{24}$$

Similarly, $P(T^{(k)})$ can be written as follows:

$$\begin{aligned}
 P(T^{(k)}) = P'(S^*) [& 3C_2^3e_k^4 - (16C_2^4 - 20C_2^2C_3 + 6C_3^2 + 10C_2C_4 - 4C_5)e_k^5 \\
 & - (-24C_2^5 + 96C_2^3C_3 - 41C_2C_3^2 - 40C_2^2C_4 - 12C_2^5 \\
 & + 17C_3C_4 + 13C_2C_5 + 8C_2^6 - 5C_6)e_k^6 + \dots].
 \end{aligned}
 \tag{25}$$

Furthermore, we have

$$\begin{aligned}
 H^{(k)} = \left(\frac{1}{2}\right)C_2e_k^2 + & (-C_2^2 + C_3)e_k^3 + \left(-\frac{7}{2}C_2C_3 + \frac{3}{2}C_4 + \frac{5}{2}C_2^3\right)e_k^4 + \left(-6C_2^4 \right. \\
 & + 12C_2^2C_3 - 5C_2C_4 - 3C_3^2 + 2C_5 + \frac{69}{5}C_2^3C_3 - 10C_2^2C_4 \left.)e_k^5 \right. \\
 & + \left(\frac{37}{2}C_2C_3^2 + 17C_2^2C_4 - \frac{13}{2}C_2C_5 - 36C_2^3C_3 + \frac{5}{2}C_6\right)e_k^6 + \dots
 \end{aligned}
 \tag{26}$$

and

$$\begin{aligned}
 J^{(k)} = \left(\frac{1}{2}C_2 + \frac{1}{5}\sqrt{15}C_2\right)e_k^2 + & \left(-C_2^2 + C_3 - \frac{2}{5}\sqrt{15}C_2^2 + \frac{2}{5}\sqrt{15}C_3\right)e_k^3 \\
 & + \left(\frac{3}{2}C_4 + \frac{5}{2}C_2^3 + \frac{3}{5}\sqrt{15}C_2^3 + \frac{3}{5}\sqrt{15}C_4 - \frac{7}{5}\sqrt{15}C_2C_3 \right. \\
 & \left. - \frac{7}{2}C_2C_3\right)e_k^4 + \dots .
 \end{aligned}
 \tag{27}$$

Similarly, we have

$$\begin{aligned}
 W^{(k)} = \left(\frac{1}{2}C_2 + \frac{1}{5}\sqrt{15}C_2\right)e_k^2 + & \left(-C_2^2 + C_3 - \frac{2}{5}\sqrt{15}C_2^2 + \frac{2}{5}\sqrt{15}C_3\right)e_k^3 \\
 & + \left(\frac{3}{2}C_4 + \frac{5}{2}C_2^3 + \frac{3}{5}\sqrt{15}C_2^3 + \frac{3}{5}\sqrt{15}C_4 - \frac{7}{5}\sqrt{15}C_2C_3 \right. \\
 & \left. - \frac{7}{2}C_2C_3\right)e_k^4 + \dots .
 \end{aligned}
 \tag{28}$$

The expression $P'(H^{(k)} + J^{(k)})$ can be written as follows:

$$\begin{aligned}
 P'(H^{(k)} + J^{(k)}) = I + & \left(C_2^2 - \frac{2}{5}\sqrt{15}C_2^2 \right) e_k^2 + \left(2C_2^3 - \frac{4}{5}C_2\sqrt{15}C_3 + 2C_2C_3 \right. \\
 & + \left. \frac{4}{5}\sqrt{15}C_2^3 \right) e_k^3 + \left(5C_2^4 + \frac{11}{5}\sqrt{15}C_2^2C_3 - \frac{6}{5}C_2\sqrt{15}C_4 \right. \\
 & \left. - \frac{89}{20}C_2^2C_3 + 3C_2C_4 - \frac{6}{5}\sqrt{15}C_2^4 \right) e_k^4 + \dots
 \end{aligned} \tag{29}$$

and

$$\begin{aligned}
 P'(H^{(k)} + W^{(k)}) = I + & \left(C_2^2 - \frac{2}{5}\sqrt{15}C_2^2 \right) e_k^2 + \left(-2C_2^3 + \frac{4}{5}C_2\sqrt{15}C_3 + 2C_2C_3 \right. \\
 & \left. - \frac{4}{5}\sqrt{15}C_2^3 \right) e_k^3 + \left(5C_2^4 - \frac{11}{5}\sqrt{15}C_2^2C_3 + \frac{6}{5}C_2\sqrt{15}C_4 \right. \\
 & \left. - \frac{89}{20}C_2^2C_3 + 3C_2C_4 + \frac{6}{5}\sqrt{15}C_2^4 \right) e_k^4 + \dots
 \end{aligned} \tag{30}$$

Furthermore, we have

$$\begin{aligned}
 P'(H^{(k)}) = I + C_2^2 e_k^2 + & \left(-2C_2^3 + 2C_2C_3 \right) e_k^3 + \left(5C_2^4 - \frac{25}{4}C_2^2C_3 + 3C_2C_4 \right) e_k^4 \\
 & + \left(-12C_2^5 + 21C_2^3C_3 - 10C_2^2C_4 - 3C_2C_3^2 + 4C_2C_5 \right) e_k^5 + \dots
 \end{aligned} \tag{31}$$

From (29)–(31), we have

$$\begin{aligned}
 & \left[4P'(H^{(k)}) + \frac{5}{2}P'(H^{(k)} + W^{(k)}) + \frac{5}{2}P'(H^{(k)} + J^{(k)}) \right]^{-1} \\
 & = -\frac{1}{3}C_2^2 e_k^2 + \left(\frac{2}{3}C_2^3 \right. \\
 & \quad \left. - \frac{2}{3}C_2C_3 \right) e_k^3 + \left(-\frac{2}{3}C_2^4 + \frac{7}{4}C_2^2C_3 - C_2C_4 \right) e_k^4 + \left(-\frac{5}{3}C_2^3C_3 + \frac{10}{3}C_2^2C_4 \right. \\
 & \quad \left. - \frac{1}{3}C_2C_3^2 - \frac{4}{3}C_2C_5 \right) e_k^5 + \dots
 \end{aligned} \tag{32}$$

From (18)–(32), we obtain

$$\begin{aligned}
 e_{k+1} = 3C_2^5 e_k^6 + & \left(-38C_2^6 + 78C_2^4C_3 + -32C_2^3C_4 - 28C_2^2C_3^2 \right. \\
 & \left. + 8C_2^2C_5 + 12C_2C_3C_4 \right) e_k^7 + \dots
 \end{aligned} \tag{33}$$

From (33), we conclude that the proposed method yields convergence of order 6. □

4. Numerical Results

In this section, we consider some problems to show the performance and efficiency of the newly developed method. We compare Newton’s method (NM) (see [6]) and methods (4), (5), (23), (25) and (27) in [15] with Algorithm 1. The stopping criterion is

$$\text{Error} = \|S^{(k+1)} - S^{(k)}\|_\infty < 10^{-15},$$

and k denotes the number of iterations. The computational order of convergence q (see [16]) is approximated by

$$q \approx \frac{\ln(\|S^{(k+1)} - S^{(k)}\| / \|S^{(k)} - S^{(k-1)}\|)}{\ln(\|S^{(k)} - S^{(k-1)}\| / \|S^{(k-1)} - S^{(k-2)}\|)}. \quad (34)$$

Consider the following systems of nonlinear equations (see [16]).

Problem 1. $-x^2 + \frac{y^2}{6} + y - 17 = 0,$
 $x^2 - y - 19 = 0,$

Problem 2. $x^2 + y^2 + z^2 - 1 = 0,$
 $2x^2 + y^2 - 4z = 0,$
 $3x^2 - 4y^2 + z^2 = 0.$

Problem 3. $e^{x^2} + 8x \sin(y) = 0,$
 $x + 2y - 1 = 0.$

Problem 4. $\cos(x) - \sin(y) = 0,$
 $z^x + 1/y = 0,$
 $e^x - z^2 = 0.$

Problem 5. $x^2 + y^2 + z^2 - 9 = 0,$
 $xyz = 1,$
 $x + y - z^2 = 0.$

Problem 6. $xy + xz + yz - 1 = 0,$
 $yz + w(y + z) = 0,$
 $xz + w(x + z) = 0,$
 $xy + w(x + y) = 0.$

Numerical results are given in Table 1 below.

Table 1. Numerical results for the Problems 1 to 6.

Method	Initial Guess	k	Approximate Solution	q
Problem 1				
NM		6	(5.000000000000, 6.000000000000) ^T	2.0
(4)		4	(5.000000000000, 6.000000000000) ^T	2.9
(5)	(5.5, 6.8) ^T	4	(5.000000000000, 6.000000000000) ^T	2.9
(23)		4	(5.000000000000, 6.000000000000) ^T	3.9
(25)		4	(5.000000000000, 6.000000000000) ^T	3.9
(27)		4	(5.000000000000, 6.000000000000) ^T	3.9
Algorithm 1		3	(5.000000000000, 6.000000000000) ^T	5.9

Table 1. Cont.

Method	Initial Guess	k	Approximate Solution	q
Problem 2				
NM	$(0.5, 0.5, 0.5)^T$	6	$(0.698288610, 0.628524230, 0.342564189)^T$	2.0
(4)		4	$(0.698288610, 0.628524230, 0.342564189)^T$	3.0
(5)		4	$(0.698288610, 0.628524230, 0.342564189)^T$	3.0
(23)		4	$(0.698288610, 0.628524230, 0.342564189)^T$	4.0
(25)		4	$(0.698288610, 0.628524230, 0.342564189)^T$	4.0
(27)		4	$(0.698288610, 0.628524230, 0.342564189)^T$	4.0
Algorithm 1		3	$(0.698288610, 0.628524230, 0.342564189)^T$	5.9
Problem 3				
NM	$(0.5, 1.0)^T$	7	$(-0.22850805121143, 0.61425402560572)^T$	2.0
(4)		4	$(-0.22850805121143, 0.61425402560572)^T$	3.0
(5)		4	$(-0.22850805121143, 0.61425402560572)^T$	3.0
(23)		4	$(-0.22850805121143, 0.61425402560572)^T$	3.8
(25)		4	$(-0.22850805121143, 0.61425402560572)^T$	4.2
(27)		4	$(-0.22850805121143, 0.61425402560572)^T$	4.2
Algorithm 1		3	$(-0.22850805121143, 0.61425402560572)^T$	6.1
Problem 4				
NM	$(1.0, 0.5, 1.5)^T$	7	$(0.90956949, 0.66122683, 1.57583414)^T$	2.0
(4)		5	$(0.90956949, 0.66122683, 1.57583414)^T$	3.0
(5)		5	$(0.90956949, 0.66122683, 1.57583414)^T$	3.0
(23)		4	$(0.90956949, 0.66122683, 1.57583414)^T$	4.7
(25)		4	$(0.90956949, 0.66122683, 1.57583414)^T$	4.1
(27)		4	$(0.90956949, 0.66122683, 1.57583414)^T$	4.1
Algorithm 1		3	$(0.90956949, 0.66122683, 1.57583414)^T$	6.3
Problem 5				
NM	$(2.5, 0.5, 1.5)^T$	7	$(2.49137571, 0.2427458788, 1.653517941)^T$	2.0
(4)		5	$(2.49137571, 0.2427458788, 1.653517941)^T$	3.0
(5)		5	$(2.49137571, 0.2427458788, 1.653517941)^T$	3.0
(23)		4	$(2.49137571, 0.2427458788, 1.653517941)^T$	4.5
(25)		4	$(2.49137571, 0.2427458788, 1.653517941)^T$	4.5
(27)		4	$(2.49137571, 0.2427458788, 1.653517941)^T$	4.5
Algorithm 1		3	$(2.49137571, 0.2427458788, 1.653517941)^T$	6.1
Problem 6				
NM	$(0.6, 0.6, 0.6, -0.2)^T$	5	$(0.577350, 0.577350, 0.577350, -0.288680)^T$	2.1
(4)		4	$(0.577350, 0.577350, 0.577350, -0.288680)^T$	3.3
(5)		4	$(0.577350, 0.577350, 0.577350, -0.288680)^T$	3.3
(23)		3	$(0.577350, 0.577359, 0.577350, -0.288680)^T$	5.5
(25)		3	$(0.577350, 0.577350, 0.577350, -0.288680)^T$	5.5
(27)		3	$(0.577350, 0.577350, 0.577350, -0.288680)^T$	5.5
Algorithm 1		3	$(0.577350, 0.577350, 0.577350, -0.288680)^T$	6.0

Problem 7 ([15]). Consider a nonlinear boundary-value problem of the following form:

$$\begin{aligned}
 y''(t) + y^{1+b}(t) &= 0 \quad (t \in [0, 1]; b > 0) \\
 y(0) &= 0, y(1) = 1.
 \end{aligned}
 \tag{35}$$

Here we have discretized the above nonlinear ODE (35) by using the finite difference method.

By taking $b = 2.5$ and $n = 10$, we obtain the following system of nonlinear equations.

$$\begin{aligned}
 100y_2 - 200y_1 + y_1^{3.5} &= 0, \\
 100y_3 - 200y_2 + 100y_1 + y_2^{3.5} &= 0, \\
 100y_4 - 200y_3 + 100y_2 + y_3^{3.5} &= 0, \\
 100y_5 - 200y_4 + 100y_3 + y_4^{3.5} &= 0, \\
 100y_6 - 200y_5 + 100y_4 + y_5^{3.5} &= 0, \\
 100y_7 - 200y_6 + 100y_5 + y_6^{3.5} &= 0, \\
 100y_8 - 200y_7 + 100y_6 + y_7^{3.5} &= 0, \\
 100y_9 - 200y_8 + 100y_7 + y_8^{3.5} &= 0, \\
 -200y_9 + 100y_8 + 100 + y_9^{3.5} &= 0,
 \end{aligned}
 \tag{36}$$

where $y^{(0)} = (1, 1, 1, 1, 1, 1, 1, 1, 1)^T$ is the initial guess. We obtain the approximate solution as follows:

$$\begin{aligned}
 y^* = & (0.1039574502033 \dots, 0.2079117381290 \dots, 0.3118302489670 \dots, \\
 & 0.4156008747144 \dots, 0.5189667289214 \dots, 0.6214486996519 \dots, \\
 & 0.7222575415768 \dots, 0.8201966396108 \dots, 0.9135562704267 \dots)^T.
 \end{aligned}
 \tag{37}$$

We compare Algorithm 1 with the Newton–Simpson method (NS-M) and the Open Newton method (ON-M) (see [9]), the four-point method (KH-M) (see [13]), the Newton–Gauss method (NG-M), and the fifth-order scheme (M 14) (see [15]). The numerical results are shown in Table 2 below.

Table 2. Numerical results for Problem 7.

Method	k	1	2	3	4	5
NS-M	$\ S_{k+1} - S_k\ _2$	1.61121	4.7021×10^{-2}	5.7614×10^{-7}	1.2500×10^{-21}	1.3253×10^{-65}
	q	3.20022	2.98531	2.99899	2.99999	3.00000
	$\ F(S_k)\ _2$	4.3531×10^{-3}	5.6034×10^{-8}	1.2677×10^{-22}	1.3581×10^{-66}	1.6345×10^{-198}
ON-M	$\ S_{k+1} - S_k\ _2$	1.61111	46671×10^{-2}	5.6464×10^{-7}	1.17711×10^{-21}	1.10665×10^{-65}
	q	3.19700	2.98555	2.99891	2.99999	3.00000
	$\ F(S_k)\ _2$	4.3121×10^{-3}	5.4934×10^{-8}	1.1944×10^{-22}	1.13333×10^{-66}	9.4989×10^{-199}
KH-M	$\ S_{k+1} - S_k\ _2$	1.61111	4.6911×10^{-2}	5.7277×10^{-7}	1.2288×10^{-21}	1.2577×10^{-65}
	q	3.19921	2.98555	2.99800	2.99999	3.00000
	$\ F(S_k)\ _2$	4.3411×10^{-3}	5.5731×10^{-8}	1.2455×10^{-22}	1.2888×10^{-66}	1.3951×10^{-198}
NG-M	$\ S_{k+1} - S_k\ _2$	6.44555	1.8681×10^{-1}	2.2633×10^{-6}	4.7391×10^{-21}	4.5099×10^{-65}
	q	3.19711	2.98555	2.99899	2.99999	3.00000
	$\ F(S_k)\ _2$	4.3175×10^{-3}	5.5051×10^{-8}	1.2011×10^{-22}	1.1544×10^{-66}	1.0044×10^{-198}
M(14)	$\ S_{k+1} - S_k\ _2$	1.64888	2.0166×10^{-3}	2.8188×10^{-18}	1.6500×10^{-92}	1.1465×10^{-463}
	q	5.10032	4.99732	4.99999	5.00000	5.00000
	$\ F(S_k)\ _2$	1.9300×10^{-4}	2.8741×10^{-19}	1.6941×10^{-93}	1.1788×10^{-464}	1.9077×10^{-2320}
Alg. 1	$\ S_{k+1} - S_k\ _2$	0.1622×10^{-48}	0	-	-	-
	q	6.10040	6	-	-	-
	$\ F(S_k)\ _2$	0.7544×10^{-47}	0	-	-	-

From Table 2, we see that the proposed method converges to the solution in just two iterations. To illustrate the performance of the new method, we plot the approximate solution against the Maple solution in Figure 1.

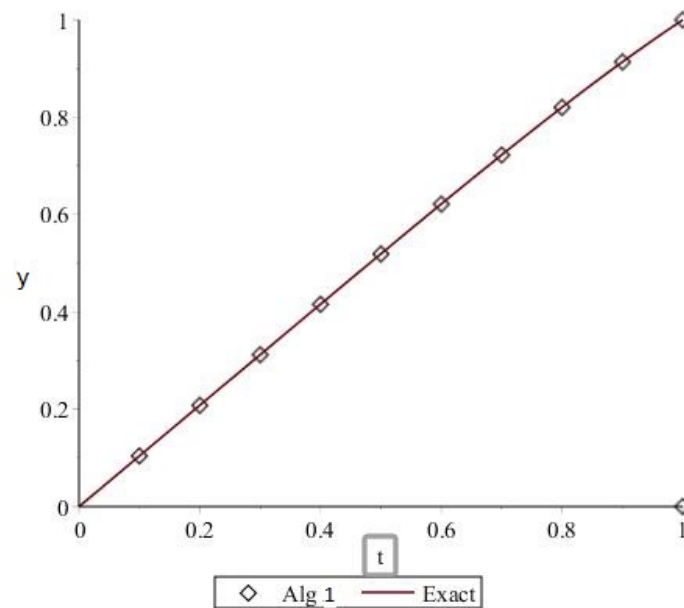


Figure 1. Comparison between the exact solution (Maple solution) and the approximate solution

In the next problem, we compare Algorithm 1 with M6 [17] of order 6.

Problem 8. Consider the following integral equation:

$$y(r) - 1 - \frac{y(r)}{4} \mu y(r) \int_0^1 k(r, u) y(u) du = 0. \quad (38)$$

Solving (38), we have the following system of nonlinear equations:

$$y_i \approx 1 + \frac{1}{8} y_i \sum_{j=1}^8 \frac{u_i \beta_j}{u_i + u_j} y_j, i = 1, 2 \dots 8. \quad (39)$$

For more detail see [17]. We compare Algorithm 1 with M6 [17] in Table 3.

Table 3. Numerical results and comparison for Problem 8.

Method	Number of Iterations	Error
Newton	5	3.1408×10^{-16}
M6	3	2.2204×10^{-16}
Algorithm 1	3	1.0000×10^{-19}

From the last column of Table 3, we conclude that the new method is more accurate than M6 [17].

5. Conclusions

In this article, we have implemented a new three-step Newton method for solving a system of nonlinear equations. The order of convergence of the proposed method is six. To show the effectiveness of the new method, we have provided some numerical tests. The graphical illustration shows the accuracy of the proposed method. Numerical results confirmed that the suggested method converges to the solution in fewer iterations with high accuracy, which justifies the advantage of this method.

Author Contributions: Conceptualization, H.M.S., J.I. and A.K.; methodology, J.I., M.A. and A.K.; formal analysis, Y.S.G., J.I. and R.C.; investigation, review and editing, H.M.S., J.I. and M.A.; writing,

J.I. and A.K.; funding acquisition, R.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work has been supported by the grant provided by Division of Computational Science, Faculty of Science, Prince of Songkla University, Hat Yai, Songkhla 90110, Thailand.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Atkinson, K.E. *An Introduction to Numerical Analysis*, 2nd ed.; John Wiley and Sons: New York, NY, USA, 1987.
2. Abbasbandy, S. Extended Newton's method for a system of nonlinear equations by modified Adomian decomposition method. *Appl. Math. Comput.* **2005**, *170*, 648–656. [[CrossRef](#)]
3. Babajee, D.K.R.; Dauhoo, M.Z. An analysis of the properties of the variants of Newton's method with third order convergence. *Appl. Math. Comput.* **2006**, *183*, 659–684. [[CrossRef](#)]
4. Babajee, D.K.R.; Dauhoo, M.Z.; Darvishi, M.T.; Barati, A. A note on the local convergence of iterative methods based on Adomian decomposition method and 3-node quadrature rule. *Appl. Math. Comput.* **2008**, *200*, 452–458. [[CrossRef](#)]
5. Babolian, E.; Biazar, J.; Vahidi, A.R. Solution of a system of nonlinear equations by Adomian decomposition method. *Appl. Math. Comput.* **2004**, *150*, 847–854. [[CrossRef](#)]
6. Burden, R.L.; Faires, J.D. *Numerical Analysis*, 7th ed.; PWS Publishing Company: Boston, MA, USA, 2001.
7. Candelario, G.; Cordero, A.; Torregrosa, J.R. Multipoint Fractional Iterative Methods with $(2\alpha + 1)$ th-Order of Convergence for Solving Nonlinear Problems. *Mathematics* **2020**, *8*, 452. [[CrossRef](#)]
8. Cordero, A.; Torregrosa, J.R. Variants of Newton's method for functions of several variables. *Appl. Math. Comput.* **2006**, *183*, 199–208. [[CrossRef](#)]
9. Cordero, A.; Torregrosa, J.R. Variants of Newton's method using fifth-order quadrature formulas. *Appl. Math. Comput.* **2007**, *190*, 686–698. [[CrossRef](#)]
10. Darvishi, M.T.; Barati, A. A third-order Newton-type method to solve systems of nonlinear equations. *Appl. Math. Comput.* **2007**, *187*, 630–635. [[CrossRef](#)]
11. Darvishi, M.T.; Barati, A. Super cubic iterative methods to solve systems of nonlinear equations. *Appl. Math. Comput.* **2007**, *188*, 1678–1685. [[CrossRef](#)]
12. Darvishi, M.T.; Barati, A. A fourth-order method from quadrature formulae to solve system of nonlinear equations. *Appl. Math. Comput.* **2007**, *188*, 257–261. [[CrossRef](#)]
13. Khirallah, M.Q.; Hafiz, M.A. Novel three order methods for solving a system of nonlinear equations. *Bull. Math. Sci. Appl.* **2012**, *2*, 1–14. [[CrossRef](#)]
14. Noeiaghdam, S.; Araghi, M.A.F. A novel algorithm to evaluate definite integrals by the Gauss-Legendre integration rule based on the stochastic arithmetic: Application in the model of osmosis system. *Math. Model. Eng. Prob.* **2020**, *7*, 577–586.
15. Su, Q.-F. A unified model for solving a system of nonlinear equations. *Appl. Math. Comput.* **2016**, *290*, 46–55. [[CrossRef](#)]
16. Liu, Z.-L.; Zheng, Q.; Huang, C.-E. Third- and fifth-order Newton-Gauss methods for solving system of nonlinear equations with n variables. *Appl. Math. Comput.* **2016**, *290*, 250–257.
17. Maduh, K. Sixth order Newton-Type method for solving system Of nonlinear equations and its applications. *Appl. Math. E-Notes.* **2017**, *17*, 221–230.