symmetry

MDPI

Article

Fixed-Point Iterative Method with Eighth-Order Constructed by Undetermined Parameter Technique for Solving Nonlinear Systems

Xiaofeng Wang

School of Mathematical Sciences, Bohai University, Jinzhou 121000, China; w200888w@163.com; Tel.: +86-150-4169-9258

Abstract: In this manuscript, by using undetermined parameter method, an efficient iterative method with eighth-order is designed to solve nonlinear systems. The new method requires one matrix inversion per iteration, which means that computational cost of our method is low. The theoretical efficiency of the proposed method is analyzed, which is superior to other methods. Numerical results show that the proposed method can reduce the computational time, remarkably. New method is applied to solve the numerical solution of nonlinear ordinary differential equations (ODEs) and partial differential equations (PDEs). The nonlinear ODEs and PDEs are discretized by finite difference method. The validity of the new method is verified by comparison with analytic solutions.

Keywords: ordinary differential equations; partial differential equations; nonlinear systems; convergence order; computational efficiency; computational cost

MSC: 65H05; 65B99

check for updates

Citation: Wang, X. Fixed-Point Iterative Method with Eighth-Order Constructed by Undetermined Parameter Technique for Solving Nonlinear Systems. *Symmetry* **2021**, *13*, 863. https://doi.org/10.3390/sym13050863

Academic Editors: Wei-Shih Du, Alicia Cordero Barbero, Huaping Huang and Juan Ramón Torregrosa Sánchez

Received: 13 March 2021
Accepted: 20 April 2021
Published: 12 May 2021

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

Copyright: © 2021 by the author. 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

Symmetry is a fundamental topic in many areas of physics and mathematics. Many problems from engineering and mathematics possess symmetry, which can be transformed into the nonlinear systems $F(z) = 0$. Some nonlinear ODEs and PDEs can be discretized into nonlinear systems. Using the iterative method to solve these nonlinear systems, we can find the numerical solution of ODEs and PDEs. Newton's method [1,2] is the oldest and famous iterative method for the numerical solution of nonlinear systems,

$$z^{(j+1)} = z^{(j)} - J_j^{-1} F(z^{(j)}), \tag{1}$$

where $J_j = F'(z^{(j)})$ is the Jacobin matrix. Newton's method is one-point iterative method, which is convergent quadratically. Any one-point iterative method which is constructed by F and its first $r-1$ derivatives cannot get higher order than r. Researchers try to improve the computational efficiency of one-point method by different ways. One effective way is to increase the iteration step of iterative method. This kind of methods are called multistep or multipoint iterative methods. It has better efficiency than one-point iterative method. For example, based on a quadrature formulae, Darvishi et al. [3] suggested an efficient multipoint method with order four that requires two F, two matrix inversions and three $F^{'}$. Grau-Sánchez et al. [4] proposed a variant of Ostrowski's method, which needs two matrix inversions, three F, two $F^{'}$ and one first-order divided difference. Using the pseudocompostion technique, Cordero et al. [5] presented a sixth-order Jarratt-type method that uses three $F^{'}$, two matrix inversions and three F. They [6] also presented a four-step eighth-order method that requires the same computational cost as their sixth-order Jarratt-type method. Using the weight function technique, Sharma and Arora [7] suggested a Jarratt-type method with order six, which requires one matrix inversion, three F and two

F'. Behl and Cordero [8] designed a sixth-order scheme that needs two matrix inversions, two F and one first-order divided difference. Behl and Arora [9] proposed a derivative-free scheme with order seven for solving nonlinear systems, which needs one matrix inversion, five F and two first-order divided differences. Using the interpolation technique, we [10] obtained a seventh-order method that is extendible to solve nonlinear systems. This method requires three matrix inversions, four F and five first-order divided differences. We [11] also obtained another seventh-order fixed-point method that needs one matrix inversion, five F and three first-order divided differences. Sharma and Aroa [12] obtained a seventh-order derivative-free method. This method requires two matrix inversions, four F and five first-order divided differences. Using the undetermined parameter technique, Narang et al. [13] designed a seventh-order method, which needs one matrix inversion, three F and two first-order divided differences per iteration.

Many efficient multipoint iterative methods for solving nonlinear equations have been proposed, see [14–20]. However, not all multipoint iterative methods can be extended to solve nonlinear systems. Therefore, it is an interesting research to construct multipoint iterative method for solving systems of nonlinear equations. Ham and Chun [14] proposed the following fifth-order method for solving nonlinear equations

$$\begin{cases} t_j = z_j - \frac{f(z_j)}{f'(z_j)}, \\ z_{j+1} = t_j - \frac{f'(t_j)+3f'(z_j)}{5f'(t_j)-f'(z_j)} \frac{f(t_j)}{f'(z_j)}, \end{cases} \tag{2}$$

which is called Ham-Chun's method. We generalize Ham-Chun's method to Banach space to solve nonlinear systems and obtain the following iterative scheme

$$\begin{cases} t^{(j)} = z^{(j)} - J_j^{-1} F(z^{(j)}), \\ z^{(j+1)} = t^{(j)} - N_j^{-1}[F'(t^{(j)}) + 3J_j] J_j^{-1} F(t^{(j)}), \end{cases} \tag{3}$$

where $N_j = 5F'(t^{(j)}) - J_j$ and $J_j = F'(z^{(j)})$. Method (3) is called HM5 in this paper, which requires two F, two matrix inversions and two F' per iteration. Method HM5 is not the fixed-point iterative method, so its computational efficiency is low.

In this paper, we propose an eighth-order fixed-point iterative method for the numerical solution of nonlinear systems. First, we prove the order of convergence of method HM5 in Section 2. Inspired by method HM5, we propose an eighth-order fixed-point method by using the undetermined parameter method in Section 3. The proposed method requires one LU decomposition per iteration, which means that this method has low computational cost. The computational efficiency of iterative method is analyzed in Section 4. The proposed method is used to solve the solution of nonlinear systems, nonlinear ODEs and PDEs in Section 5. Section 6 gives a short conclusion.

2. Iterative Method with Order Five for Solving Nonlinear Systems

Theorem 1. *Let $\xi \in \mathbb{R}^m$ be the zero of $F : \mathbf{D} \subset \mathbb{R}^m \to \mathbb{R}^m$,$F(z)$ be sufficiently Fréchet differentiable and $F'(z)$ be continuous and nonsingular at $\mathbf{D}$. Then, method HM5 converges to ξ with order five, if initial guess $z^{(0)}$ close to ξ.*

Proof. Let $B_i = \frac{1}{i!} F'(\xi)^{-1} F^{(i)}(\xi) \in L_i(\mathbb{R}^m, \mathbb{R}^m)$ and $e = z^{(j)} - \xi$. By using the results in [15], we get

$$F(z^{(j)}) = F'(\xi)[e + B_2 e^2 + B_3 e^3 + O(e^4)], \tag{4}$$

$$F'(z^{(j)}) = F'(\xi)[I + 2B_2 e + 3B_3 e^2 + 4B_4 e^3 + O(e^4)], \tag{5}$$

and

$$F'(z^{(j)})^{-1} = [I - 2B_2 e + (4B_2^2 - 3B_3)e^2 - (8B_2^3 - 6B_2 B_3 - 6B_3 B_2 + 4B_4)e^3 + O(e^4)] F'(\xi)^{-1}. \tag{6}$$

Let $E = t^{(j)} - \xi$. From (4)–(6), we get

$$E = t^{(j)} - \xi = e - F'(z^{(j)})^{-1}F(z^{(j)}) = B_2e^2 - 2(B_2^2 - B_3)e^3 + O(e^4). \tag{7}$$

Similar argument to (4), we arrive at

$$F(t^{(j)}) = F'(\xi)[E + B_2E^2 + O(E^3)], \tag{8}$$

$$F'(t^{(j)}) = F'(\xi)[I + 2B_2E + O(E^2)], \tag{9}$$

$$N_j = F'(\xi)[4I - 2B_2e - (10B_2^2 - 3B_3)e^2 + 4(5B_2^3 - 5B_2B_3 + B_4)e^3 + O(e^4)], \tag{10}$$

and

$$N_j^{-1} = [\frac{1}{4}I - \frac{1}{8}B_2e + \frac{1}{16}(-9B_2^2 + 3B_3)e^2 + \frac{1}{32}(21B_2^3 - 34B_2B_3 + 8B_4)e^3 + O(e^4)]F'(\xi)^{-1}, \tag{11}$$

Using (3), (5), (6), (9) and (11), we obtain the error equation:

$$e_{n+1} = (\frac{3}{2}B_2^4 - B_2^2B_3)e^5 + O(e^6) \tag{12}$$

It is easy to see that method HM5 is of fifth-order convergence. □

Based on method HM5, we will give a new fixed-point method with order eight in the following section.

3. Fixed-Point Iterative Method with Order Eight

Inspired by method HM5, we design a fixed-point iterative method by using the undetermined parameter technique as follows:

$$\begin{cases} t^{(j)} = z^{(j)} - J_j^{-1}F(z^{(j)}), \\ w^{(j)} = t^{(j)} - \left[I + (I + g_1M^{(j)})M^{(j)}\right]J_j^{-1}F(t^{(j)}), \\ z^{(j+1)} = w^{(j)} - \left[I + (I + g_2M^{(j)})M^{(j)}\right]J_j^{-1}F(w^{(j)}), \end{cases} \tag{13}$$

where $M^{(j)} = J_j^{-1}(J_j - F'(t^{(j)}))$, I is an identity matrix and $g_i, (i = 1, 2)$ are constant parameters to be determined.

Theorem 2. *Let $\xi \in \mathbb{R}^m$ be the zero of $F(z) : \mathbf{D} \subset \mathbb{R}^m \to \mathbb{R}^m$, $F(z)$ be sufficiently Fréchet differentiable and $F'(z)$ be continuous and nonsingular at $\mathbf{D}$. If $g_1 = \frac{5}{4}$, $g_2 = \frac{3}{2}$ and initial guess $z^{(0)}$ is close to ξ, then method (13) reaches eighth-order convergence.*

Proof. From (5), (6) and (9), we obtain

$$M^{(j)} = 2B_2e - (6B_2^2 - 3B_3)e^2 + 4(4B_2^3 - 4B_2B_3 + B_4)e^3 + O(e^4). \tag{14}$$

Let $\varepsilon = w^{(j)} - \xi$. Using (6), (8) and (14), we arrive at

$$\varepsilon = w^{(j)} - \xi = B_2^3(5 - 4g_1)e^4 + 4B_2^2(B_3(6 - 5g_1) + A_2^2(-9 + 10g_1))e^5 + O(e^6), \tag{15}$$

and

$$F(w^{(j)}) = F'(\alpha)[\varepsilon + B_2\varepsilon^2 + O(\varepsilon^3)]. \tag{16}$$

Using (13)–(16), we can get the error equation of method (13)

$$e_{n+1} = z^{(j+1)} - \xi = 2B_2^5(-5 + 4g_1)(-3 + 2g_2)e^6 - 4(B_2^4(B_3(-56 + g_1(46 - 32g_2) + 39g_2)$$
$$+B_2^2(89 - 76g_2 + 8g_1(-11 + 9g_2))))e^7 + B_2^3(2B_2B_4(157 - 108g_2 + 8g_1($$

$$-16+11g_2))+B_2^4(2483-16g_1^2-2652g_2+24g_1(-123+122g_2))$$
$$+B_3^2(651-481g_2+g_1(-554+408g_2))+B_2^2B_3(-3123$$
$$+2760g_2-4g_1(-789+668g_2)))e^8+O(e^9) \tag{17}$$

Taking $g_1=\frac{5}{4}$ and $g_2=\frac{3}{2}$, we get

$$e_{n+1}=(280B_2^7-48B_2^5B_3+2B_2^3B_3^2)e^8+O(e^9) \tag{18}$$

It is easy see that the order of method (13) is eight. Per iteration, method (13) requires three F, two F' and twice LU decompositions. Compared to method HM5, method (13) only increases one function evaluation $F(z)$.

Thus, we get an iterative method with eighth-order as follows

$$\begin{cases} t^{(j)}=z^{(j)}-J_j^{-1}F(z^{(j)}), \\ w^{(j)}=t^{(j)}-\left[I+(I+\frac{5}{4}M^{(j)})M^{(j)}\right]J_j^{-1}F(t^{(j)}), \\ z^{(j+1)}=w^{(j)}-\left[I+(I+\frac{3}{2}M^{(j)})M^{(j)}\right]J_j^{-1}F(w^{(j)}), \end{cases} \tag{19}$$

where $M^{(j)}=J_j^{-1}(J_j-F'(t^{(j)}))$ and $J_j=F'(z^{(j)})$. Method (19) is called NM8 in this paper. □

4. Computational Efficiency

The computational efficiency indexes CEI is proposed by Grau-Sánchez et al. [4,16,17]:

$$CEI(\nu_0,\nu_1,m)=\rho^{\frac{1}{C(\nu_0,\nu_1,m)}}, \tag{20}$$

where

$$C(\nu_0,\nu_1,m)=S_0(m)\nu_0+S_1(m)\nu_1+S(m,l). \tag{21}$$

In per iteration, $C(\nu_0,\nu_1,m)$ represents the cost of iterative method. Parameter $S_0(m)$ is the number of the scalar functions used in F and $[y,x;F]$. Parameter $S_1(m)$ is the number of scalar functions used in F'. $S(m,l)$ is the number of products . The parameter l in (21) is the ratio between the divisions and products. The parameters ν_0 and ν_1 in (21) are the ratios between products and evaluations that are required to express $C(\nu_0,\nu_1,m)$ in terms of products. ρ represents the convergence order. The LU decomposition is used to solve linear systems in the processing of iteration. The following methods are used to compared the computational efficiency:

Darvishi et al. fourth-order method [3] (DM4)

$$\begin{cases} w^{(j)}=z^{(j)}-J_j^{-1}F(z^{(j)}), \\ t^{(j)}=z^{(j)}-J_j^{-1}(F(z^{(j)})+F(w^{(j)})), \\ z^{(j+1)}=z^{(j)}-6(J_j+4F'(\frac{z^{(j)}+t^{(j)}}{2})+F'(t^{(j)}))^{-1}F(t^{(j)}). \end{cases} \tag{22}$$

Grau-Sánchez et al. sixth-order method [4] (GM6)

$$\begin{cases} w^{(j)}=z^{(j)}-J_j^{-1}F(z^{(j)}), \\ t^{(j)}=w^{(j)}-\mu_1F(w^{(j)}), \\ z^{(j+1)}=t^{(j)}-\mu_1F(t^{(j)}), \end{cases} \tag{23}$$

where $\mu_1=2[z^{(j)},w^{(j)};F]^{-1}-J_j^{-1}$ and $[w^{(j)},z^{(j)};F]$ is the first-order divided difference operator.

Cordero et al. sixth-order method [5] (CM6)

$$\begin{cases} w^{(j)} = z^{(j)} - \frac{1}{2}J_j^{-1}F(z^{(j)}), \\ t^{(j)} = z^{(j)} + \mu_2[3J_j - 4F(w^{(j)})], \\ z^{(j+1)} = t^{(j)} + \mu_2 F(t^{(j)}). \end{cases} \tag{24}$$

where $\mu_2 = [J_j - 2F'(w^{(j)})]^{-1}$.

Cordero et al. eighth-order method [6] (CM8)

$$\begin{cases} w^{(j)} = z^{(j)} - \frac{1}{2}J_j^{-1}F(z^{(j)}), \\ u^{(j)} = \frac{1}{3}(4w^{(j)} - z^{(j)}) + \mu_3 F(z^{(j)}), \\ v^{(j)} = u^{(j)} + 2\mu_3 F(u^{(j)}), \\ z^{(j+1)} = v^{(j)} + 2\mu_3 F(v^{(j)}). \end{cases} \tag{25}$$

where $\mu_3 = [J_j - 3F'(t^{(j)})]^{-1}$.

The C and CEI of different iterative methods are given below:

$$C_{DM4} = 2m\nu_0 + 3m^2\nu_1 + \frac{m}{3}(2m^2 + 9m - 5 + 3(m+2)l) \tag{26}$$

$$CEI_{DM4} = 4^{1/C_{DM4}}. \tag{27}$$

$$C_{GM6} = (m^2 + 2m)\nu_0 + m^2\nu_1 + \frac{2m}{3}(m^2 + 6m - 4 + 3(m+2)l) \tag{28}$$

$$CEI_{GM6} = 6^{1/C_{GM6}}. \tag{29}$$

$$C_{CM6} = 3m\nu_0 + 2m^2\nu_1 + \frac{m}{3}(2m^2 + 9m + 1 + 3(m+2)l) \tag{30}$$

$$CEI_{CM6} = 6^{1/C_{CM6}}. \tag{31}$$

$$C_{CM8} = 3m\nu_0 + 2m^2\nu_1 + \frac{m}{3}(2m^2 + 12m + 4 + 3(m+3)l) \tag{32}$$

$$CEI_{CM8} = 8^{1/C_{CM8}}. \tag{33}$$

$$C_{HM5} = 2m\nu_0 + 2m^2\nu_1 + \frac{m}{3}(2m^2 + 12m - 8 + 3(m+2)l) \tag{34}$$

$$CEI_{HM5} = 5^{1/C_{HM5}}. \tag{35}$$

$$C_{NM8} = 3m\nu_0 + 2m^2\nu_1 + \frac{m}{6}(2m^2 + 63m - 29 + 3(m+13)l) \tag{36}$$

$$CEI_{NM8} = 8^{1/C_{NM8}}. \tag{37}$$

The computational efficiencies of different methods are compared by the ratio $R_{A,B}$ [4,10]

$$R_{A,B} = \frac{\ln CEI_A}{\ln CEI_B} = \frac{\ln(\rho_A)C_B(\nu_0, \nu_1, m)}{\ln(\rho_B)C_A(\nu_0, \nu_1, m)}. \tag{38}$$

If $R_{A,B} > 1$, then method B is less efficient than method A. $R_{A,B} = 1$ is the boundary between the computational efficiencies, which is an equation of ν_0 as a function of m,l and ν_1. Parameters in (38) satisfy $\nu_0 > 0$, $\nu_1 > 0$, $l \geq 1$ and $m \geq 2$, the CEI of different methods is studied in the following result.

Theorem 3.

1. For all $\nu_0, \nu_1 > 0$ and $l \geq 1$, we arrive at:

(a) $CEI_{NM8} > CEI_{DM4}$ for all $m \geq 8$.

(b) $CEI_{NM8} > CEI_{CM6}$ for all $m \geq 2$.

(c) $CEI_{NM8} > CEI_{HM5}$ for all $m \geq 11$.

(d) $CEI_{NM8} > CEI_{CM8}$ for all $m \geq 18$.

2. We have $CEI_{NM8} > CEI_{GM6}$ *for all* $m \geq 2$ *and* $\nu_0 > H$, *where* $H = \frac{6(2r-q)m\nu_1+2(r-5q)m^2+(63r-9q+(3r-33q)l)m+43q-5r+(39r-33q)l}{18(mq+q-r)}$.

Proof. 1. From (26), (27) and (36)–(38), we get the boundary $R_{NM8,DM4} = 1$ that is given by

$$\nu_1 = \frac{-4(m-9)m - 14 + (21-6m)l}{15m}. \tag{39}$$

Figure 1 shows the boundary $R_{NM8,DM4} = 1$, which implies that $CEI_{NM8} > CEI_{DM4}$ on the above and $CEI_{NM8} < CEI_{DM4}$ on the below of boundary plane. Parameter $m \geq 8$, ν_1 is always negative which means that DM4 is less efficient than NM8 for all $m \geq 8$ and $l \geq 1$.

Figure 2 shows the boundary lines in (l, ν_1)-plane for $m = 4, 8$ and 9 which means that $CEI_{NM8} > CEI_{DM4}$ for $m = 8, m = 9$ and $l \geq 1$. For $m < 4$, $CEI_{NM8} > CEI_{DM4}$ on the above of boundary line and $CEI_{NM8} < CEI_{DM4}$ on the below of boundary line.

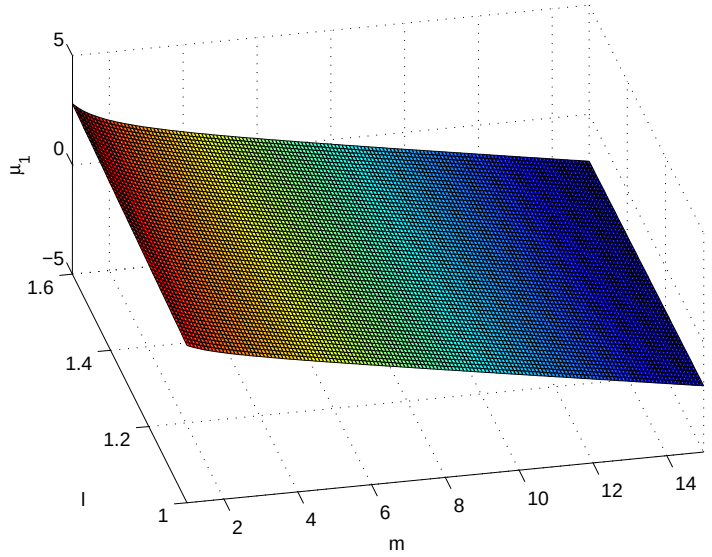


Figure 1. Boundary plane of $R_{NM8,DM4} = 1$.

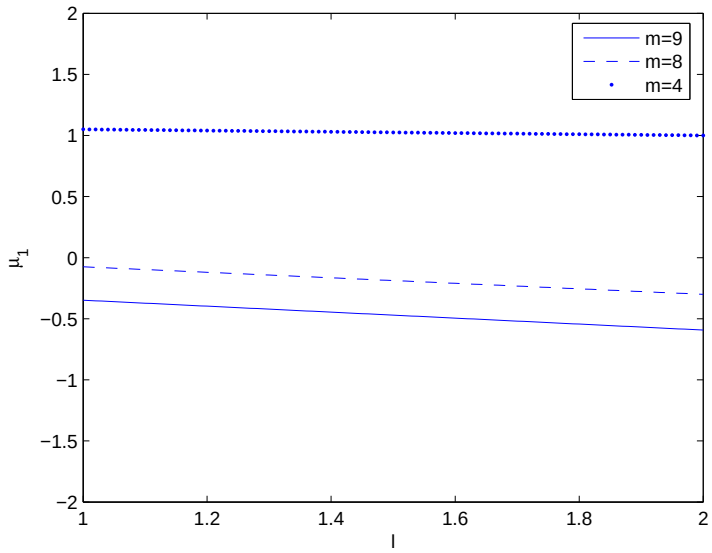


Figure 2. Boundary lines in (l, ν_1)-plane for m = 4, 8 and 9.

From (28), (29) and (36)–(38), we get the boundary $R_{NM8,CM6} = 1$ that is given by

$$\nu_0 = \frac{12(\ln 3 - 2\ln 2)m\nu_1 + 2(\ln 3 - 35\ln 2)m^2 + (63 + 9\ln 2 + (3\ln 3 - 15\ln 2)l)m - 35\ln 2 - 29\ln 3 + (39\ln 3 + 3\ln 2)l}{18(2\ln 2 - \ln 3)}, \quad (40)$$

For $m \geq 2$, ν_0 is always negative. Figure 3 shows the boundary for $l = 1$, which means that CM6 is less efficient than N8 for all $m \geq 2$ and $\nu_1 \geq 0$.

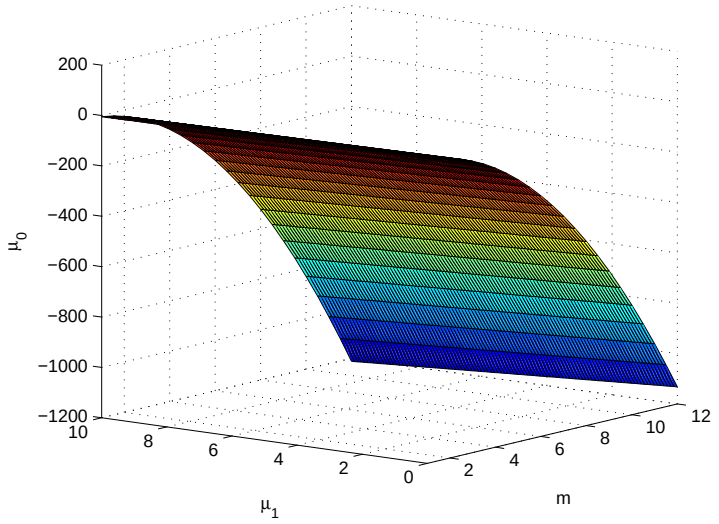


Figure 3. Boundary plane of $R_{NM8,CM6} = 1$.

Using (35)–(38), we obtain the boundary $R_{NM8,HM5} = 1$ that is written by

$$\nu_0 = \frac{2(p - 6q)m^2 + 63mp - 72mq + 48q - 29p + [3(m + 13)p - 18(m + 2)q]l + 12(q - 3p)m\nu_1}{18(2q - p)}, \quad (41)$$

where $p = \ln 5$ and $q = \ln 2$. For $m \geq 11$,ν_0 is always negative. Figure 4 shows the boundary for $l = 1$, which implies that method NM8 is more efficient than method HM5 for all $m \geq 11$,$\nu_1 \geq 0$ and $l = 1$.

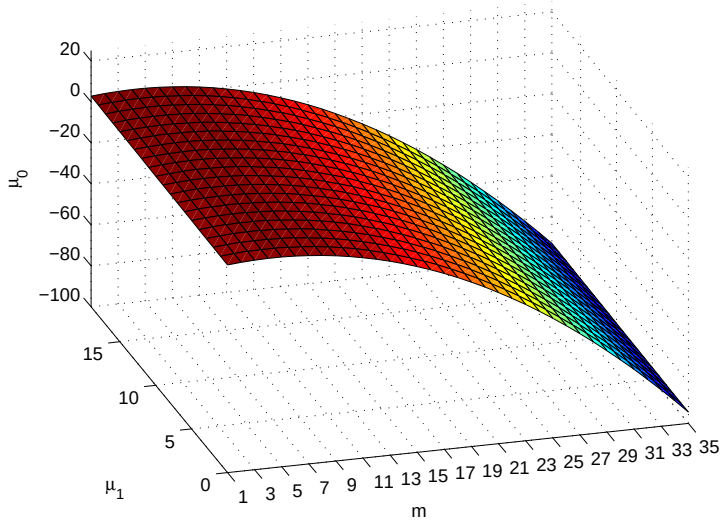


Figure 4. Boundary plane of $R_{NM8,HM5} = 1$.

Using (32), (33) and (36)–(38), we obtain the following relation between NM8 and CM8:

$$R_{NM8,CM8} = \frac{\ln(\rho_{NM8})C_{CM8}}{\ln(\rho_{CM8})C_{NM8}} = \frac{3\nu_0 + 2m\nu_1 + \frac{1}{3}(2m^2 + 12m + 4 + 3(m+3)l)}{3\nu_0 + 2m\nu_1 + \frac{1}{6}(2m^2 + 63m - 29 + 3(m+13)l)}. \quad (42)$$

Using the numerator to subtract the denominator of (42), we get

$$\frac{1}{6}((2m-39)m + 37 + 3(m-7)l). \quad (43)$$

If the expression (43) is more than zero then $R_{NM8,CM8} > 1$. Thus, we obtain that $CEI_{NM8} > CEI_{CM8}$ for all $m \geq 18$ and $l \geq 1$.

2. From (28), (29) and (36)–(38), we get the boundary $R_{NM8,GM6} = 1$ that is given by

$$\nu_0 = \frac{6(2r-q)m\nu_1 + 2(r-5q)m^2 + (63r - 9q + (3r+3q)l)m + 19q - 29r + (39 - 33q)l}{18(mq + q - r)} \quad (44)$$

where $r = \ln 3$ and $q = \ln 2$.

Figure 5 shows the boundary plane for Equation (44). Taking $l = 1$, Figure 6 shows some particular boundaries corresponding to $m = 4, 19$ and 99, wherein $CEI_{NM8} > CEI_{GM6}$ on the above and $CEI_{NM8} < CEI_{GM6}$ on the below of each line. □

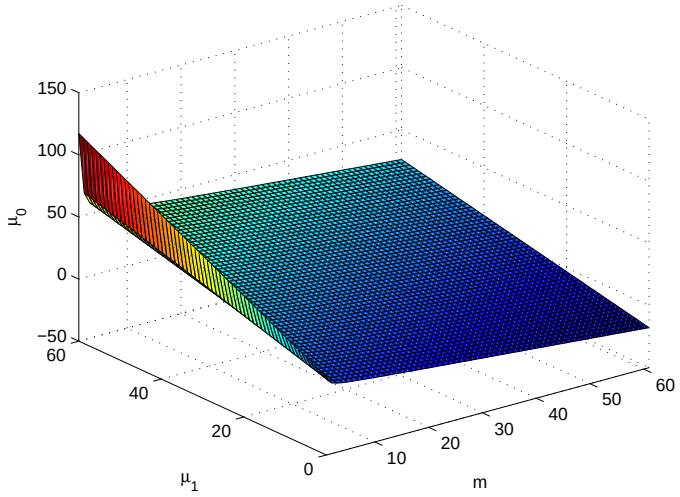


Figure 5. Boundary plane of $R_{NM8,GM6} = 1$.

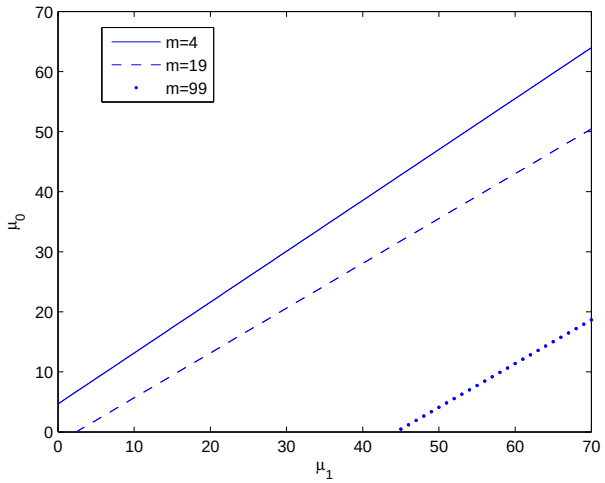


Figure 6. Boundary lines in (ν_1, ν_0)-plane for $m = 4, 19$ and 99.

5. Numerical Results

New methods HM5 and NM8 are compared with methods DM4, GM6, CM6 and CM8 for solving some nonlinear systems. Numerical algorithms are written by the Maple 14. Table 1 shows an estimate of the cost of functions in product units.

Table 1. Estimation of cost of elementary functions, where $z = \sqrt{3} - 1$ and $p = \sqrt{5}$.

Function	$z*p$	z/p	$\sqrt{z}$	$\ln(z)$	$\exp(z)$	$\arctan(z)$	$\sin(z)$	$\cos(z)$
Cost	0.110 ms	1	5	12	53	95.03	112	110

The computational cost of elementary functions based on product is showed in Table 1. Estimation of computing time of elementary functions are computed with Maple 14 in a processor Intel R Core (TM) i3-2350M CPU, 1.79 GHz (32-bit Machine) Microsoft Windows 7 Professional. Table 1 shows that the computing time of one product is 0.110 milliseconds (ms). The computational cost of division with respect to product is $l = 1$ and the computational cost of function $\sqrt{x}$ with respect to product is $l = 5$.

In Tables 2–9, EF represents the errors of function values at the last step, NI represents the number of iterations, CC represents the computational cost, EV represents the errors values of $||z^{(j)} - z^{(j-1)}||$, CEI represents the computational efficiency index, $ACOC$ [17] represents the approximated computational order of convergence, $e - Time$ represents the mean CPU time. We choose the tolerance $||z^{(j)} - z^{(j-1)}|| < 10^{-200}$ for implementing the methods.

Example 1. *Nonlinear system [16]:*

$$z_j - \cos(2z_j - \sum_{i=1}^{m} z_i) = 0, 1 \leq j \leq m.$$

For this problem, we choose the parameter $m = 4$. For $m = 4$, we require four scalar cosine functions in the evaluation of vector functions F and four scalar sine functions in the evaluation of matrix F'. Base on (21) and Table 1, we get parameters $\nu_0 = \frac{110 \times 4}{4} = 110$ and $\nu_1 = \frac{112 \times 4}{16} = 28$. For this example, we can get the computational cost of different methods by using parameters $l = 1, m = 4, \nu_0 = 110$ and $\nu_1 = 28$. The solution $\xi \approx (0.5149, \cdots, 0.5149)^T$ is founded by initial guess $z^{(0)} = (1.0, \cdots 1.0)^T$. The results of comparisons for this example are displayed in Table 2.

Table 2. Numerical results in Example 1.

Methods	NI	EV	EF	$ACOC$	C	CEI	e-$Time$
DM4	5	1.118×10^{-239}	4.630×10^{-957}	4	2560	1.000541668	1.0293
HM5	5	8.158×10^{-518}	7.000×10^{-2048}	5	1896	1.000849220	0.671
GM6	4	1.343×10^{-277}	3.887×10^{-830}	6	3232	1.000554535	1.2756
CM6	5	2.458×10^{-770}	5.000×10^{-2048}	6	2332	1.000768631	1.0525
CM8	4	1.549×10^{-314}	4.000×10^{-2048}	8	2356	1.000883005	0.7890
NM8	4	1.824×10^{-369}	1.500×10^{-2047}	8	2436	1.000853994	0.8794

Example 2. *Nonlinear system [17]:*

$$\sum_{k=1, k \neq j}^{m} z_k - \exp(-z_k) = 0, k = 1, \ldots, m$$

First, we choose the parameter $m = 8$. Using the same calculation method as example 1, we obtain parameters $v_0 = 53$ and $v_1 = 0.625$. For $m = 8$, the solution $\xi \approx (0.125951, \cdots, 0.125951)^T$ is founded by the initial guess $x^{(0)} = (5.0, \cdots 5.0)^T$. Secondly, we choose the parameters $m = 19$ and get $v_0 = 53$ and $v_1 = 0.20789$. For $m = 19$, the solution $\xi \approx (0.0527, \cdots, 0.0527)^T$ is founded by the initial guess is $z^{(0)} = (2.0, \cdots 2.0)^T$. Tables 3-4 display the numerical results.

Table 3. Numerical results in Example 2 ($m = 8$).

Methods	*NI*	*EV*	*EF*	*ACOC*	*C*	*CEI*	*e-Time*
DM4	6	1.884×10^{-525}	1.000×10^{-2048}	4	3944	1.000351556	1.1695
HM5	5	2.008×10^{-566}	4.000×10^{-2048}	5	2352	1.000684519	0.573
GM6	4	7.608×10^{-313}	6.299×10^{-938}	6	5400	1.000331862	1.9465
CM6	5	1.355×10^{-726}	2.000×10^{-2048}	6	2736	1.000655097	0.8530
CM8	4	1.038×10^{-304}	1.000×10^{-2048}	8	2816	1.000738710	0.6985
NM8	4	1.491×10^{-551}	2.000×10^{-2048}	8	3040	1.000684261	0.7189

Table 4. Numerical results in Example 2 ($m = 19$).

Methods	*NI*	*EV*	*EF*	*ACOC*	*C*	*CEI*	*e-Time*
DM4	6	2.331×10^{-770}	2.600×10^{-2048}	4	23,161	1.000059856	5.9182
HM5	5	3.121×10^{-803}	2.000×10^{-2048}	5	10,392	1.000154875	4.1079
GM6	4	2.862×10^{-471}	2.676×10^{-1412}	6	28,918	1.000061962	1.0168
CM6	5	1.191×10^{-1191}	2.300×10^{-2048}	6	11,096	1.000161491	4.2287
CM8	4	1.167×10^{-511}	2.700×10^{-2048}	8	11,495	1.000180916	4.1368
NM8	4	5.143×10^{-671}	1.900×10^{-2048}	8	11,400	1.000182424	3.3269

Example 3. *Nonlinear system [8]:*

$$\begin{cases} z_j z_{j+1} - 1 = 0, j = 1, \ldots, m-1 \\ z_m z_1 - 1 = 0. \end{cases}$$

The initial guess chosen $z^{(0)} = \{5, 5, \cdots, 5\}^t$ is used for finding the solution $\xi = \{1, 1, \cdots, 1\}^t$. We choose the parameters $m = 99, 199$ and 299 in (25), respectively. In this problem, the parameters $v_0 = 1$ and $v_1 = 0$ are not dependent of m. Tables 5 and 6 show the numerical results.

Table 5. Numerical results in Example 3 ($m = 99$).

Methods	*NI*	*EV*	*EF*	*ACOC*	*C*	*CEI*	*e-Time*
DM4	7	3.538×10^{-652}	0.000e+00	4	2,058,804	1.0000006733	10.233
HM5	6	4.138×10^{-507}	4.000×10^{-2048}	5	696,003	1.000002312	9.078
GM6	5	3.033×10^{-271}	1.674×10^{-811}	6	715,803	1.0000025031	15.178
CM6	6	2.054×10^{-1148}	0.000e+00	6	686,598	1.0000026096	7.924
CM8	5	1.071×10^{-555}	0.000e+00	8	696,597	1.0000029851	7.144
NM8	5	1.686×10^{-302}	4.000×10^{-2048}	8	432,102	1.0000048124	6.130

Table 6. The $e-Time$ (in seconds) in Example 3.

Methods	DM4	HM5	GM6	CM6	CM8	NM8
m = 199	62.759	54.562	72.899	52.213	45.287	28.797
m = 299	199.369	192.386	204.688	169.853	144.924	82.898

Example 4. *Boundary-value problem [13]:*

$$u''(z) = u(z)^3 + sin(u'(z)^2), z \in [0,1],$$
$$u(0) = 0, u(1) = 1.$$

The first and second derivatives in this problem are discretized by difference method

$$u''_j = \frac{u_{j+1}-2u_j+u_{j-1}}{h^2}, j = 1,2,3,\cdots,n-1,$$

and

$$u'_j = \frac{u_{j+1}-u_{j-1}}{2h}, j = 1,2,3,\cdots,n-1,$$

The interval $[0,1]$ *is partitioned into n smaller intervals with end points* $0 = z_0 < z_1 < \cdots < z_{n-1} < z_n = 1$. *The partition is regular, this is* $\triangle x_j = 1/n$ *for all j. We get the following nonlinear systems*

$$u_{j-1} - 2u_j + u_{j+1} - h^2u_j^3 - h^2sin((\frac{u_{j-1}-u_{j+1}}{2h})^2) = 0, j = 1,2,3,\ldots,n-1.$$

For $n = 201$, *the solution* $(0.003239, 0.006488, 0.009748, \ldots, 0.9922)^T$ *is founded by the initial guess* $z^{(0)} = (0.5, \cdots 0.5)^T$. *Parameters* $\nu_0 = 120$ *and* $\nu_1 = 80.54$. *Table 7 display the numerical results.*

Table 7. Numerical results in Example 4.

Methods	NI	EV	EF	$ACOC$	C	CEI	$e-Time$
DM4	6	3.318×10^{-249}	2.152×10^{-994}	15,200,200	1.00000009120238	103.849	
HM5	6	8.878×10^{-727}	8.845×10^{-2048}	12,020,400	1.00000013389222	89.560	
GM6	7	1.765×10^{-614}	8.828×10^{-2048}	13,641,200	1.00000013134912	2708.988	
CM6	6	1.843×10^{-1167}	9.282×10^{-2048}	12,005,000	1.00000014925111	100.137	
CM8	5	1.220×10^{-1287}	7.138×10^{-2048}	12,045,400	1.00000017263368	88.889	
NM8	5	1.370×10^{-793}	9.248×10^{-2048}	9,618,200	1.00000021619864	75.036	

Example 5. *Boundary-value problem:*

$$-u''(z) = y(z)^2, z \in [0,1],$$
$$u(0) = 0, u(1) = 1.$$

Using the same discretization method as problem 4, we obtain the nonlinear systems as follows:

$$u_{j-1} - 2u_j + u_{j+1} - h^2u_j^2 = 0, j = 1,2,3,\ldots,n-1.$$

For $n = 101$, *the solution* $(0.01084, 0.02168, 0.03252, \ldots, 0.9854)^T$ *is founded by the initial value* $z^{(0)} = (0.5, \cdots 0.5)^T$. *Parameters* $\nu_0 = 4$ *and* $\nu_1 = 2$. *Table 8 display the numerical results.*

Table 8. Numerical results in Example 5.

Methods	*NI*	*EV*	*EF*	*C*	*CEI*	*e-Time*
DM4	6	4.850×10^{-693}	1.096×10^{-2047}	767,500	1.00000180624836	6.583
HM5	5	9.508×10^{-457}	2.676×10^{-1831}	757,400	1.00000212495329	5.304
GM6	6	3.823×10^{-201}	3.593×10^{-802}	787,600	1.00000227496382	149.324
CM6	4	4.521×10^{-328}	1.381×10^{-1972}	748,100	1.00000239508303	4.009
CM8	4	5.505×10^{-765}	9.052×10^{-2048}	758,300	1.00000274224501	4.399
NM8	4	4.472×10^{-356}	8.581×10^{-2048}	484,700	1.00000429017124	3.728

Example 6. *Nonlinear PDE problem [18]:*

$$u_{zz} = u_t + u_z - u^2 + f(z,t), z \in [0,1], t \geq 0,$$
$$u(0,t) = 0, u(1,t) = 0.$$
$$f(z,t) = e^{(-t)}(-\pi * cos(\pi * z) - (\pi^2 - 2) * sin(\pi * z))$$

This PDE problem is a Heat Conduction Problems. We transform this problem to the nonlinear systems by using finite differences. The intervals $[0,1]$ *and* $[0,T]$ *are partitioned into* N *smaller intervals, and get step size* $h = 1/N$ *and* $k = T/N$ *in* z *and* t *directions. Let* $u = u(z,t)$ *be the exact solution and* $u_{i,j} \approx u(z_i, t_j)$*. Using difference method, we use obtain approximations* $u_z(z,t) \approx \frac{u(z+h,t)-u(z-h,t)}{2h}$, $u_t(z,t) \approx \frac{u(z,t)-u(z,t-k)}{k}$ *and* $u_{zz}(z,t) \approx \frac{u(z+h,t)-2u(z,t)+u(z-h,t)}{h^2}$*. We get a nonlinear system as follows:*

$$(2k + kh)u_{i-1,j} + (-4k - 2h^2)u_{i,j} + (2k - kh)u_{i+1,j} + 2kh^2u_{i,j}^2 + 2h^2u_{i,j-1} - 2kh^2f(z_i, t_j) = 0$$

for $i = 1, 2, \ldots, N-1$ *and* $j = 1, 2, \ldots, N$ *For fixed* j*, we get some nonlinear systems with size* $(N-1) \times (N-1)$*. Choosing different* N *and* T*, we solve this problem by different methods. The results are given in Table 9, where* NI *represents the number of iteration and* $e - Time$ *represents the CUP time. The approximate solution and absolute value of error of this problem are shown in Figures 7 and 8, when* $T = 0.01$ *and* $N = 50$*. Figure 9 shows the exact solutions of this problems.*

Table 9. Numerical results in Example 6.

Methods	DM4	HM5	GM6	CM6	CM8	NM8
NI	5	4	4	4	3	3
$e - Time$	22.5	18.7	19.6	19.8	16.3	15.5

Tables 2–9 show that compared with other methods, method NM8 has lower computational cost and higher computational efficiency. We find that method NM8 costs less computing time than other methods. This advantage is obvious for solving large scale nonlinear systems. Figures 7–9 show that method NM8 can be used to solve nonlinear PDE equations with high accuracy.

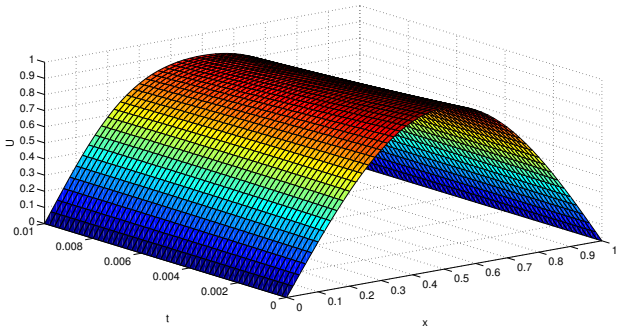


Figure 7. Approximate solutions of Heat conduction equation.

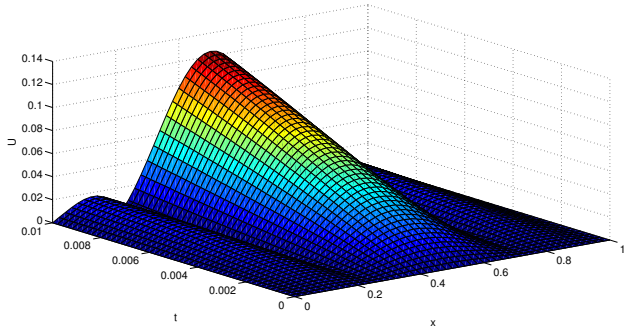


Figure 8. Absolute error of u for $t \in [0, 0.01]$.

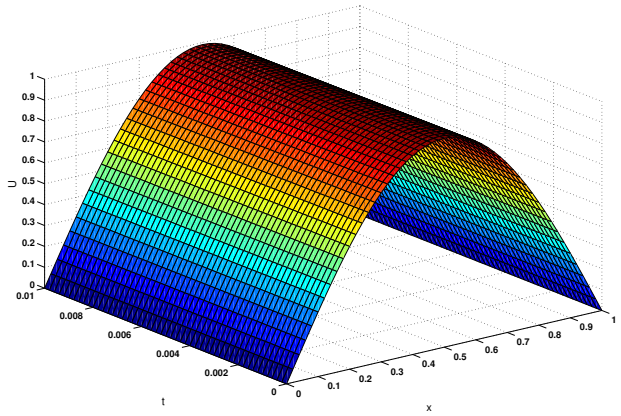


Figure 9. The exact solutions of Heat conduction equation.

6. Conclusions

We have extended the fifth-order Ham-Chun's method to Banach space and developed an efficient method with eighth-order for solving standard nonlinear systems, nonlinear ODEs and PDEs. Our method requires three functions F, two derivatives F' and one matrix inversion F'^{-1} per iteration. Tables 2–9 show that our method NM8 show a highly efficient especially in large scale nonlinear systems. Numerical results verify that our method NM8 is better than other methods in this paper.

Funding: This research was supported by the National Natural Science Foundation of China (No. 61976027), Educational Commission Foundation of Liaoning Province of China (No. LJ2019010) and University-Industry Collaborative Education Program (Nos. 201901077017, 201902014012, 201902184038).

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

References

1. Jelley, C.T. *Solving Nonlinear Equations with Newton's Method*; SIAM: Philadelphia, PA, USA, 2003.
2. Ortega, J.M.; Rheinbolt, W.C. *Iterative Solution of Nonlinear Equations in Several Variables*; Academic Press: New York, NY, USA, 1970. [CrossRef]
3. Darvishi, M.T.; Barati, A. A fourth-order method from quadrature formulae to solve systems of nonlinear equations. *Appl. Math. Comput.* **2007**, *188*, 257–261. [CrossRef]
4. Grau-Sánchez, M.; Peris, M.P.; Gutiérrez, J.M. Accelerated iterative methods for finding solutions of system of nonlinear equations. *J. Comput. Appl. Math.* **2007**, *190*, 1815–1823. [CrossRef]
5. Cordero, A.; Torregrosa, J.R.; Vassileva, M.P. Pseudocomposition: A technique to design predictor-corrector methods for systems of nonlinear equations. *Appl. Math. Comput.* **2012**, *218*, 11496–11504. [CrossRef]
6. Cordero, A.; Jordán, C.; Sanabria-Codesal, E.; Torregrosa, J.R. Highly efficient iterative algorithms for solving nonlinear systems with arbitrary order of convergence $p+3$, $p \geq 5$. *J. Comput. Appl. Math.* **2018**, *330*, 748–758. [CrossRef]
7. Sharma, J.R.; Arora, H. Efficient Jarratt-like methods for solving systems of nonlinear equations. *Calcolo* **2014**, *51*, 193–210. [CrossRef]
8. Behl, R.; Cordero, A.; Torregrosa, J.R. High order family of multivariate iterative methods: Convergence and stability. *J. Comput. Appl. Math.* **2020**, *2020*, 113053. [CrossRef]
9. Behl, R.; Arora, H. CMMSE: A novel scheme having seventh-order convergence for nonlinear systems. *J. Comput. Appl. Math.* **2020**. [CrossRef]
10. Wang, X.; Zhang, T. A family of Steffensen type methods with seventh-order convergence. *Numer. Algorithms* **2013**, *62*, 429–444. [CrossRef]
11. Wang, X.; Zhang, T.; Teng, Q.W.; Teng, M. Seventh-order derivative-free iterative method for solving nonlinear systems. *Numer. Algorithms* **2015**, *70*, 545–558. [CrossRef]
12. Sharma, J.R.; Arora, H. A novel derivative free algorithm with seventh order convergence for solving systems of nonlinear equations. *Numer. Algorithms* **2014**, *4*, 917–933. [CrossRef]
13. Narang, M.; Bhatia, S.; Janwar, V. New efficient derivative free family of seventh-order methods for solving systems of nonlinear equations. *Numer. Algorithms* **2017**, *76*, 283–307. [CrossRef]
14. Ham, Y.; Chun, C. A fifth-order iterative method for solving nonlinear equations. *Appl. Math. Comput.* **2007**, *194*, 287–290.
15. Zhanlav, T.; Otgondorj, K. Higher order Jarratt-like iterations for solving systems of nonlinear equations. *Appl. Math. Comput.* **2021**, *395*, 125849. [CrossRef]
16. Grau-Sánchez, M.; Grau, À.; Noguera, M. On the computational efficiency index and some iterative methods for solving systems of nonlinear equations. *J. Comput. Appl. Math.* **2011**, *236*, 1259–1266. [CrossRef]
17. Behl, R.; Bhalla, S.; Magreñán, Á.A.; Kumar, S. An efficient high order iterative scheme for large nonliear systems with dynamics. *J. Comput. Appl. Math.* **2020**, 113249. [CrossRef]
18. Cordero, A; Gómez, E.; Torregrosa, J.R. Efficient High-order iterative methods for solving nonlinear systems and their appliation on Heat Conduction Problems. *Complexity* **2017**, *2017*, 6457532. [CrossRef]
19. Salimi, M.; Lotfi, T.; Sharifi, S.; Siegmund, S. Optimal Newton-Secant like methods without memory for solving nonlinear equations with its dynamics. *Int. J. Comput. Math.* **2017**, *94*, 1759–1777. [CrossRef]
20. Behl, R.; Salimi, M.; Ferrara, M.; Sharifi, S.; Alharbi, S.K. Some Real-Life Applications of a Newly Constructed Derivative Free Iterative Scheme. *Symmetry* **2019**, *11*, 239.