algorithms

MDPI

Article

A Preconditioned Iterative Method for Solving Systems of Nonlinear Equations Having Unknown Multiplicity

Fayyaz Ahmad [1,2,3,*], Toseef Akhter Bhutta [4], Umar Shoaib [4], Malik Zaka Ullah [1,5], Ali Saleh Alshomrani [5], Shamshad Ahmad [6] and Shahid Ahmad [7]

1 Dipartimento di Scienza e Alta Tecnologia, Universita dell'Insubria, Via Valleggio 11, Como 22100, Italy; malik.zakaullah@uninsubria.it
2 Departament de Física i Enginyeria Nuclear, Universitat Politècnica de Catalunya, Eduard Maristanu 10, Barcelona 08019, Spain
3 UCERD (Pvt) Ltd, Islamabad 44000, Pakistan
4 Department of Computer Science, University of Gujrat, Gujrat 50700, Pakistan; tos_bhutta@hotmail.com (T.A.B.); shoaib.umar@gmail.com (U.S.)
5 Department of Mathematics, King Abdulaziz University, Jeddah 21589, Saudi Arabia; aszalshomrani@kau.edu.sa
6 Heat and Mass Transfer Technological Center, Technical University of Catalonia, Colom 11, Terrassa 08222, Spain; shamshad@cttc.upc.edu
7 Department of Mathematics, Government College University Lahore, Lahore 54000, Pakistan; shahidsms@gmail.com
* Correspondence: fahmad@studenti.uninsubria.it or fayyaz.ahmad@upc.edu; Tel.: +34-63-206-6627

Academic Editor: Alicia Cordero
Received: 13 November 2016; Accepted: 13 January 2016; Published: 18 January 2016

Abstract: A modification to an existing iterative method for computing zeros with unknown multiplicities of nonlinear equations or a system of nonlinear equations is presented. We introduce preconditioners to nonlinear equations or a system of nonlinear equations and their corresponding Jacobians. The inclusion of preconditioners provides numerical stability and accuracy. The different selection of preconditioner offers a family of iterative methods. We modified an existing method in a way that we do not alter its inherited quadratic convergence. Numerical simulations confirm the quadratic convergence of the preconditioned iterative method. The influence of preconditioners is clearly reflected in the numerically achieved accuracy of computed solutions.

Keywords: nonlinear equations; systems of nonlinear equations; singular Jacobian; roots with unknown multiplicity; nonlinear preconditioners; auxiliary function

1. Introduction

The design of an iterative method for solving nonlinear equations and systems of nonlinear equations is an active area of research. Many researchers have proposed iterative methods for solving nonlinear and systems of nonlinear equations for finding simple zeros or zeros with multiplicity greater than one [1–15]. The classical iterative method for solving nonlinear and systems of nonlinear equations to find simple zeros is the Newton method, which offers quadratic convergence [16,17] under certain conditions. When we are talking about the iterative method for solving nonlinear equations or systems of nonlinear equations to find zeros with multiplicities greater than one, the classical Newton method requires a modification. The modified Newton method for finding zeros with multiplicity greater than one for nonlinear equations can be written as

$$\begin{cases} z_0 = \text{initial guess} \\ z_{k+1} = z_k - m\,\frac{\phi(z_k)}{\phi'(z_k)}, \quad k = 0, 1, \cdots, \end{cases} \tag{1}$$

where $\phi(z_k) = 0$ is the nonlinear equation. Jose et al. [18] proposed the multidimensional version of of (1) as

$$\begin{cases} \mathbf{z}_0 = \text{initial guess} \\ \mathbf{z}_{k+1} = \mathbf{z}_k - \mathbf{\Phi}'(\mathbf{z}_k)^{-1}\,\mathrm{diag}(\mathbf{m})\,\mathbf{\Phi}(\mathbf{z}_k), \quad k = 0, 1, \cdots, \end{cases} \tag{2}$$

where $\mathbf{m} = [m_1, m_2, \cdots, m_n]^T$ is a vector of multiplicities for a system of nonlinear equations $\mathbf{\Phi}(\mathbf{z}) = \mathbf{0}$, and $\mathrm{diag}(\cdot)$ represents a diagonal matrix that keeps the vector at its main diagonal. The proof of the quadratic convergence of (2) is provided in [18]. Wu [19] proposed a variant of the Newton method with the help of an auxiliary or a preconditioner exponential function. Suppose we have a system of nonlinear equations $\mathbf{\Phi}(\mathbf{z}) = \mathbf{0}$, and we define a new system of nonlinear equations with a nonlinear preconditioner function that has the same root

$$\mathbf{\Psi}(\mathbf{z}) = e^{\mathbf{v}\odot\mathbf{z}} \odot \mathbf{\Phi}(\mathbf{z}) = \mathbf{0}, \tag{3}$$

where $\odot$ is the element-wise multiplication of two vectors. The application of the Newton method for (3) is

$$\begin{aligned} \mathbf{z}_{k+1} &= \mathbf{z}_k - \mathbf{\Psi}'(\mathbf{z}_k)^{-1}\,\mathbf{\Psi}(\mathbf{z}_k) \\ \mathbf{z}_{k+1} &= \mathbf{z}_k - \left(\mathrm{diag}\left(e^{\mathbf{v}\odot\mathbf{z}}\right)\left(\mathbf{\Phi}'(\mathbf{z}) + \mathrm{diag}(\mathbf{v}\odot\mathbf{\Phi}(\mathbf{z}))\right)\right)^{-1} e^{\mathbf{v}\odot\mathbf{z}} \odot \mathbf{\Phi}(\mathbf{z}) \\ \mathbf{z}_{k+1} &= \mathbf{z}_k - \left(\mathbf{\Phi}'(\mathbf{z}) + \mathrm{diag}(\mathbf{v}\odot\mathbf{\Phi}(\mathbf{z}))\right)^{-1}\,\mathrm{diag}\left(e^{\mathbf{v}\odot\mathbf{z}}\right)^{-1} e^{\mathbf{v}\odot\mathbf{z}} \odot \mathbf{\Phi}(\mathbf{z}) \\ \mathbf{z}_{k+1} &= \mathbf{z}_k - \left(\mathbf{\Phi}'(\mathbf{z}) + \mathrm{diag}(\mathbf{v}\odot\mathbf{\Phi}(\mathbf{z}))\right)^{-1}\,\mathbf{\Phi}(\mathbf{z}). \end{aligned} \tag{4}$$

The rate of convergence of (4) is quadratic. A modification [18] in (1) is proposed by using a exponential preconditioner

$$\mathbf{\Psi}(\mathbf{z}) = e^{\mathbf{v}\odot\mathbf{z}} \odot \mathbf{\Phi}(\mathbf{z})^{1/\mathbf{m}} = \mathbf{0}, \tag{5}$$

where $1/\mathbf{m} = [1/m_1, 1/m_2, \cdots, 1/m_n]^T$ and power of $\mathbf{\Phi}(\mathbf{z})$ is component-wise. The application of the Newton method to (5) gives

$$\mathbf{z}_{k+1} = \mathbf{z}_k - \left(\mathbf{\Phi}'(\mathbf{z}) + \mathrm{diag}(\mathbf{v}\odot\mathbf{\Phi}(\mathbf{z}))\right)^{-1}\,\mathrm{diag}(\mathbf{m})\,\mathbf{\Phi}(\mathbf{z}). \tag{6}$$

The original idea of a nonlinear preconditioner function was proposed in [19]. Noor et al. [20] proposed a Newton method with a general preconditioner. They defined a preconditioned system of nonlinear equations as follows:

$$\mathbf{\Psi}(\mathbf{z}) = \mathbf{\Lambda}(\mathbf{z}) \odot \mathbf{\Phi}(\mathbf{z}) = \mathbf{0}, \tag{7}$$

where $\mathbf{\Lambda}(\mathbf{z}) \neq \mathbf{0}$. Notice that the roots of $\mathbf{\Psi}(\mathbf{z}) = \mathbf{0}$ and $\mathbf{\Phi}(\mathbf{z}) = \mathbf{0}$ are the same because $\mathbf{\Lambda}(\mathbf{z}) \neq \mathbf{0}$ for all $\mathbf{z}$. The first order Fréchet derivative of (7) can be computed as

$$\begin{aligned}
&\Psi_i(\mathbf{z}) = \Phi_i(\mathbf{z})\,\Lambda_i(\mathbf{z}) \\
&\nabla\Psi_i(\mathbf{z})^T = \Phi_i(\mathbf{z})\,\nabla\Lambda_i(\mathbf{z})^T + \Lambda_i(\mathbf{z})\,\nabla\Phi_i(\mathbf{z})^T, \;\; i = 1, 2, \cdots, n \\
&\begin{bmatrix} \nabla\Psi_1(\mathbf{z})^T \\ \nabla\Psi_2(\mathbf{z})^T \\ \nabla\Psi_3(\mathbf{z})^T \\ \vdots \\ \nabla\Psi_n(\mathbf{z})^T \end{bmatrix} = \begin{bmatrix} \Phi_1(\mathbf{z}) & 0 & \cdots & 0 \\ 0 & \Phi_2(\mathbf{z}) & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \Phi_n(\mathbf{z}) \end{bmatrix} \begin{bmatrix} \nabla\Lambda_1(\mathbf{z})^T \\ \nabla\Lambda_2(\mathbf{z})^T \\ \nabla\Lambda_3(\mathbf{z})^T \\ \vdots \\ \nabla\Lambda_n(\mathbf{z})^T \end{bmatrix} + \begin{bmatrix} \Lambda_1(\mathbf{z}) & 0 & \cdots & 0 \\ 0 & \Lambda_2(\mathbf{z}) & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \Lambda_n(\mathbf{z}) \end{bmatrix} \begin{bmatrix} \nabla\Phi_1(\mathbf{z})^T \\ \nabla\Phi_2(\mathbf{z})^T \\ \nabla\Phi_3(\mathbf{z})^T \\ \vdots \\ \nabla\Phi_n(\mathbf{z})^T \end{bmatrix}
\end{aligned} \tag{8}$$

From (8), the Fréchet derivative of $\mathbf{\Phi}(\mathbf{z}) \odot \mathbf{\Lambda}(\mathbf{z})$ is

$$\begin{aligned}
&(\mathbf{\Phi}(\mathbf{z}) \odot \mathbf{\Lambda}(\mathbf{z}))' = \operatorname{diag}(\mathbf{\Phi}(\mathbf{z}))\mathbf{\Lambda}'(\mathbf{z}) + \operatorname{diag}(\mathbf{\Lambda}(\mathbf{z}))\mathbf{\Phi}'(\mathbf{z}) \\
&\mathbf{\Psi}'(\mathbf{z}) = \operatorname{diag}(\mathbf{\Lambda}(\mathbf{z}))\,\mathbf{\Phi}'(\mathbf{z}) + \operatorname{diag}(\mathbf{\Phi}(\mathbf{z}))\,\mathbf{\Lambda}'(\mathbf{z}) \\
&\mathbf{\Psi}'(\mathbf{z}) = \operatorname{diag}(\mathbf{\Lambda}(\mathbf{z}))\left(\mathbf{\Phi}'(\mathbf{z}) + \operatorname{diag}(\mathbf{\Phi}(\mathbf{z}))\operatorname{diag}(\mathbf{\Lambda}(\mathbf{z}))^{-1}\mathbf{\Lambda}'(\mathbf{z})\right).
\end{aligned} \tag{9}$$

If we apply the Newton method to (7), we obtain

$$\begin{aligned}
&\mathbf{z}_{k+1} = \mathbf{z}_k - \left(\mathbf{\Phi}'(\mathbf{z}) + \operatorname{diag}(\mathbf{\Phi}(\mathbf{z}))\operatorname{diag}(\mathbf{\Lambda}(\mathbf{z}))^{-1}\mathbf{\Lambda}'(\mathbf{z})\right)^{-1}\operatorname{diag}(\mathbf{\Lambda}(\mathbf{z}))^{-1}\mathbf{\Lambda}(\mathbf{z}) \odot \mathbf{\Phi}(\mathbf{z}) \\
&\mathbf{z}_{k+1} = \mathbf{z}_k - \left(\mathbf{\Phi}'(\mathbf{z}) + \operatorname{diag}(\mathbf{\Phi}(\mathbf{z}))\operatorname{diag}(\mathbf{\Lambda}(\mathbf{z}))^{-1}\mathbf{\Lambda}'(\mathbf{z})\right)^{-1}\mathbf{\Phi}(\mathbf{z}).
\end{aligned} \tag{10}$$

The convergence order of (10) is two. The iterative method (6) with a general preconditioner can be written as

$$\mathbf{z}_{k+1} = \mathbf{z}_k - \left(\mathbf{\Phi}'(\mathbf{z}) + \operatorname{diag}(\mathbf{\Phi}(\mathbf{z}))\operatorname{diag}(\mathbf{\Lambda}(\mathbf{z}))^{-1}\mathbf{\Lambda}'(\mathbf{z})\right)^{-1}\operatorname{diag}(\mathbf{m})\,\mathbf{\Phi}(\mathbf{z}). \tag{11}$$

The convergence order of (11) is also two. The modified Newton method [17,21,22] for solving nonlinear equations with unknown multiplicity can be developed in this way. We define a new function

$$s(z) = \frac{\phi(z)}{\phi'(z)}. \tag{12}$$

The application of the Newton method to (12) gives

$$\begin{aligned}
z_{k+1} &= z_k - \frac{s(z_k)}{s'(z_k)} \\
z_{k+1} &= z_k - \frac{\phi'(z_k)\phi(z_k)}{\phi'(z_k)^2 - \phi''(z_k)\,\phi(z_k)}.
\end{aligned} \tag{13}$$

The order of convergence of (13) is two. Noor and his co-researchers [23] have constructed a family of iterative methods for solving nonlinear equations with unknown multiplicity by introducing a preconditioner. They defined a new function

$$q(z) = \frac{\phi(z)\,\lambda(z)}{\phi'(z)} \tag{14}$$

and application of the Newton method to (14) gives

$$
\begin{aligned}
z_{k+1} &= z_k - \frac{q(z_k)}{q'(z_k)} \\
z_{k+1} &= z_k - \frac{\phi'(z_k)\,\phi(z_k)\,\lambda(z_k)}{\phi'(z_k)(\phi(z_k)\,\lambda(z_k))' - \phi''(z_k)\,\phi(z_k)\,\lambda(z_k)},
\end{aligned}
\tag{15}
$$

where $\lambda(z)$ is a non-zero function. The order of convergence of (15) is two.

2. Proposed Method

When we observe (14), we can notice that the preconditioner is only introduced for $\phi(z)$, and not for $\phi'(z)$. We will also introduce a preconditioner for $\phi'(z)$, and will show that the convergence order of (14) is still quadratic. We define a new function

$$
q(z) = \frac{\lambda(z)\,\phi(z)}{(\omega(z)\,\phi(z))'},
\tag{16}
$$

and after applying the Newton method, we obtain

$$
\begin{aligned}
z_{k+1} &= z_k - \frac{q(z_k)}{q'(z_k)} \\
z_{k+1} &= z_k - \frac{(\omega(z_k)\,\phi(z_k))'\,\lambda(z_k)\,\phi(z_k)}{(\omega(z_k)\,\phi(z_k))'\,(\lambda(z_k)\,\phi(z_k))' - (\omega(z_k)\,\phi(z_k))''\,\lambda(z_k)\,\phi(z_k)},
\end{aligned}
\tag{17}
$$

where $\omega(z)$ is a non-zero function. For the purpose of generalization of the iterative method (17) to a system of nonlinear equations, we define a new function $\mathbf{Q}(\mathbf{z})$

$$
\mathbf{Q}(\mathbf{z}) = \Big((\mathbf{\Omega}(\mathbf{z}) \odot \mathbf{\Phi}(\mathbf{z}))'\Big)^{-1}\,(\mathbf{\Lambda}(\mathbf{z}) \odot \mathbf{\Phi}(\mathbf{z})) = \mathbf{0}.
\tag{18}
$$

The first order Fréchet derivative of (18) can be written as

$$
\begin{aligned}
\mathbf{Q}'(\mathbf{z}) = &\left(\Big((\mathbf{\Omega}(\mathbf{z}) \odot \mathbf{\Phi}(\mathbf{z}))'\Big)^{-1}\right)^2 \Big((\mathbf{\Omega}(\mathbf{z}) \odot \mathbf{\Phi}(\mathbf{z}))'\,\Big(\mathbf{\Lambda}(\mathbf{z}) \odot \mathbf{\Phi}(\mathbf{z})\Big)' \\
&- (\mathbf{\Omega}(\mathbf{z}) \odot \mathbf{\Phi}(\mathbf{z}))'\,(\mathbf{\Omega}(\mathbf{z}) \odot \mathbf{\Phi}(\mathbf{z}))''\,\Big((\mathbf{\Omega}(\mathbf{z}) \odot \mathbf{\Phi}(\mathbf{z}))'\Big)^{-1}\,(\mathbf{\Lambda}(\mathbf{z}) \odot \mathbf{\Phi}(\mathbf{z}))\Big).
\end{aligned}
\tag{19}
$$

Further simplification of $\mathbf{Q}'(\mathbf{z})^{-1}\,\mathbf{Q}(\mathbf{z})$ gives

$$
\begin{aligned}
\mathbf{Q}'(\mathbf{z})^{-1}\,\mathbf{Q}(\mathbf{z}) = &\Big(\,(\mathbf{\Omega}(\mathbf{z}) \odot \mathbf{\Phi}(\mathbf{z}))'\,(\mathbf{\Lambda}(\mathbf{z}) \odot \mathbf{\Phi}(\mathbf{z}))' - (\mathbf{\Omega}(\mathbf{z}) \odot \mathbf{\Phi}(\mathbf{z}))'\,(\mathbf{\Omega}(\mathbf{z}) \odot \mathbf{\Phi}(\mathbf{z}))'' \\
&\Big((\mathbf{\Omega}(\mathbf{z}) \odot \mathbf{\Phi}(\mathbf{z}))'\Big)^{-1}\,(\mathbf{\Lambda}(\mathbf{z}) \odot \mathbf{\Phi}(\mathbf{z}))\,\Big)(\mathbf{\Omega}(\mathbf{z}) \odot \mathbf{\Phi}(\mathbf{z}))'\,(\mathbf{\Lambda}(\mathbf{z}) \odot \mathbf{\Phi}(\mathbf{z}))\,.
\end{aligned}
\tag{20}
$$

If compare the underlined expressions in (17) and (20), they are different. Generally, it is not possible to commute $(\mathbf{\Omega}(\mathbf{z}) \odot \mathbf{\Phi}(\mathbf{z}))'$ with $(\mathbf{\Omega}(\mathbf{z}) \odot \mathbf{\Phi}(\mathbf{z}))''$. However, we artificially eliminate terms $(\mathbf{\Omega}(\mathbf{z}) \odot \mathbf{\Phi}(\mathbf{z}))'$ and $\Big((\mathbf{\Omega}(\mathbf{z}) \odot \mathbf{\Phi}(\mathbf{z}))'\Big)^{-1}$ from expression $(\mathbf{\Omega}(\mathbf{z}) \odot \mathbf{\Phi}(\mathbf{z}))'\,(\mathbf{\Omega}(\mathbf{z}) \odot \mathbf{\Phi}(\mathbf{z}))''\,\Big((\mathbf{\Omega}(\mathbf{z}) \odot \mathbf{\Phi}(\mathbf{z}))'\Big)^{-1}$, and get the following iterative method.

$$
\begin{aligned}
\mathbf{z}_{k+1} = \mathbf{z}_k - &\Big(\,(\mathbf{\Omega}(\mathbf{z}_k) \odot \mathbf{\Phi}(\mathbf{z}_k))'\,(\mathbf{\Lambda}(\mathbf{z}_k) \odot \mathbf{\Phi}(\mathbf{z}_k))' - (\mathbf{\Omega}(\mathbf{z}_k) \odot \mathbf{\Phi}(\mathbf{z}_k))''\,(\mathbf{\Lambda}(\mathbf{z}_k) \odot \mathbf{\Phi}(\mathbf{z}_k))\,\Big)^{-1} \\
&(\mathbf{\Omega}(\mathbf{z}_k) \odot \mathbf{\Phi}(\mathbf{z}_k))'\,(\mathbf{\Lambda}(\mathbf{z}_k) \odot \mathbf{\Phi}(\mathbf{z}_k))\,.
\end{aligned}
\tag{21}
$$

It can be seen that the iterative method (21) is not the application of the Newton method to (18). The iterative method (17) for solving scalar nonlinear equations with unknown multiplicity and vector version (21) are exactly the same. We will only provide the proof of quadratic convergence for (21), and it is automatically applicable to scalar version (17). An iterative method was proposed in [24] to compute the zeros with multiplicity of system of nonlinear equations that used preconditioners for a system of nonlinear equations, but not for the Jacobian of the system of nonlinear equations. Notice that in this article we are introducing preconditioners for the system of nonlinear equations as well as the Jacobian of the system of nonlinear equations.

3. Convergence

In the following theorem, we established the proof of quadratic convergence of (21).

Theorem 1. *Let* $\mathbf{\Phi} : D \subseteq \mathbb{R}^n \longrightarrow \mathbb{R}^n$ *and* $\boldsymbol{\kappa} = [\kappa_1, \kappa_2, \kappa_3, \cdots, \kappa_n]^T \in D$ *is a root of* $\mathbf{\Phi}(\mathbf{z}) = (\mathbf{z} - \boldsymbol{\kappa})^{\mathbf{m}} \odot \mathbf{P}(\mathbf{z}) = \mathbf{0}$ *with corresponding multiplicities vector* $\mathbf{m} = [m_1, m_2, \cdots, m_n]^T$ *and non-zero function* $\mathbf{P} = [p_1(\mathbf{z}), p_2(\mathbf{z}), \cdots, p_n(\mathbf{z})]^T$ *with* $p_i(\mathbf{z})(\neq \mathbf{0}) \in C^2(D)$*. Then, there exists a subset* $E \subseteq D$ *such that—if we choose* $\mathbf{z}_0 \in E$*—the iterative method (21) has quadratic convergence in* E*.*

Proof. Let $\boldsymbol{\epsilon} = \mathbf{z} - \boldsymbol{\kappa}$ then $\mathbf{\Phi}(\mathbf{z}) = \boldsymbol{\epsilon}^{\mathbf{m}} \odot \mathbf{P}(\mathbf{z})$. Whenever we take vector power of a vector, it is always component-wise. So, $\boldsymbol{\epsilon}^{\mathbf{m}} = \left[\epsilon_1^{m_1}, \epsilon_2^{m_2}, \cdots, \epsilon_n^{m_n}\right]^T$. The first-order Fréchet derivative of $\mathbf{\Phi}(\mathbf{z})$ is

$$\mathbf{\Phi}'(\mathbf{z}) = \mathrm{diag}\Big(\mathbf{m} \odot \boldsymbol{\epsilon}^{\mathbf{m}-\mathbf{1}} \odot \mathbf{P}(\mathbf{z})\Big) + \mathrm{diag}(\boldsymbol{\epsilon}^{\mathbf{m}})\mathbf{P}'(\mathbf{z}). \tag{22}$$

The expressions for terms in (21) are computed as follows.

$$(\mathbf{\Omega}(\mathbf{z}) \odot \mathbf{\Phi}(\mathbf{z}))' = \mathrm{diag}\Big(\mathbf{m} \odot \boldsymbol{\epsilon}^{\mathbf{m}-\mathbf{1}} \odot \mathbf{P}(\mathbf{z}) \odot \mathbf{\Omega}(\mathbf{z})\Big) + \mathrm{diag}(\boldsymbol{\epsilon}^{\mathbf{m}} \odot \mathbf{\Omega}(\mathbf{z}))\mathbf{P}'(\mathbf{z}) + \mathrm{diag}(\boldsymbol{\epsilon}^{\mathbf{m}} \odot \mathbf{P}(\mathbf{z}))\mathbf{\Omega}'(\mathbf{z}) \tag{23}$$

$$(\mathbf{\Lambda}(\mathbf{z}) \odot \mathbf{\Phi}(\mathbf{z}))' = \mathrm{diag}\Big(\mathbf{m} \odot \boldsymbol{\epsilon}^{\mathbf{m}-\mathbf{1}} \odot \mathbf{P}(\mathbf{z}) \odot \mathbf{\Lambda}(\mathbf{z})\Big) + \mathrm{diag}(\boldsymbol{\epsilon}^{\mathbf{m}} \odot \mathbf{\Lambda}(\mathbf{z}))\mathbf{P}'(\mathbf{z}) + \mathrm{diag}(\boldsymbol{\epsilon}^{\mathbf{m}} \odot \mathbf{P}(\mathbf{z}))\mathbf{\Lambda}'(\mathbf{z}) \tag{24}$$

By using (23) and (24), we can write the product of $(\mathbf{\Omega}(\mathbf{z}) \odot \mathbf{\Phi}(\mathbf{z}))'$ and $(\mathbf{\Lambda}(\mathbf{z}) \odot \mathbf{\Phi}(\mathbf{z}))'$ as

$$(\mathbf{\Omega}(\mathbf{z}) \odot \mathbf{\Phi}(\mathbf{z}))' \, (\mathbf{\Lambda}(\mathbf{z}) \odot \mathbf{\Phi}(\mathbf{z}))' = \Big(\mathbf{m}^2 \odot \boldsymbol{\epsilon}^{2\mathbf{m}-\mathbf{2}} \odot \mathbf{P}(\mathbf{z})^2 \odot \mathbf{\Omega}(\mathbf{z}) \odot \mathbf{\Lambda}(\mathbf{z})\Big) + O\Big(\mathrm{diag}\Big(\boldsymbol{\epsilon}^{2\mathbf{m}-\mathbf{1}}\Big)\Big). \tag{25}$$

Next, we compute the second order Fréceht derivative of $(\mathbf{\Omega}(\mathbf{z}) \odot \mathbf{\Phi}(\mathbf{z}))'$. Let $\boldsymbol{\phi}$ be a scalar vector of length n

$$(\mathbf{\Omega}(\mathbf{z}) \odot \mathbf{\Phi}(\mathbf{z}))'\boldsymbol{\phi} = \mathbf{m} \odot \boldsymbol{\epsilon}^{\mathbf{m}-\mathbf{1}} \odot \mathbf{P}(\mathbf{z}) \odot \mathbf{\Omega}(\mathbf{z}) \odot \boldsymbol{\phi} + \mathbf{A}(\boldsymbol{\phi}), \tag{26}$$

where $\mathbf{A}(\boldsymbol{\phi}) = \mathrm{diag}(\boldsymbol{\epsilon}^{\mathbf{m}} \odot \mathbf{\Omega}(\mathbf{z}))\mathbf{P}'(\mathbf{z})\boldsymbol{\phi} + \mathrm{diag}(\boldsymbol{\epsilon}^{\mathbf{m}} \odot \mathbf{P}(\mathbf{z}))\mathbf{\Omega}'(\mathbf{z})\boldsymbol{\phi}$. We again compute the first order Fréchet derivative of (26)

$$\begin{aligned}
(\mathbf{\Omega}(\mathbf{z}) \odot \mathbf{\Phi}(\mathbf{z}))''\boldsymbol{\phi} =& \mathrm{diag}\Big(\mathbf{m} \odot (\mathbf{m}-\mathbf{1}) \odot \boldsymbol{\epsilon}^{\mathbf{m}-\mathbf{1}} \odot \boldsymbol{\phi} \odot \mathbf{P}(\mathbf{z}) \odot \mathbf{\Omega}(\mathbf{z})\Big) \\
& + \mathrm{diag}\Big(\mathbf{m} \odot \boldsymbol{\epsilon}^{\mathbf{m}-\mathbf{1}} \odot \boldsymbol{\phi}\Big)(\mathbf{P}(\mathbf{z}) \odot \mathbf{\Omega}(\mathbf{z}))' + \mathbf{A}'(\boldsymbol{\phi}) \\
(\mathbf{\Omega}(\mathbf{z}) \odot \mathbf{\Phi}(\mathbf{z}))''(\mathbf{\Lambda}(\mathbf{z}) \odot \mathbf{\Phi}(\mathbf{z})) =& \mathrm{diag}\Big(\mathbf{m} \odot (\mathbf{m}-\mathbf{1}) \odot \boldsymbol{\epsilon}^{2\mathbf{m}-\mathbf{2}} \odot \mathbf{P}(\mathbf{z})^2 \odot \mathbf{\Omega}(\mathbf{z})\Big) \\
& + \mathrm{diag}\Big(\mathbf{m} \odot \boldsymbol{\epsilon}^{2\mathbf{m}-\mathbf{1}} \odot \mathbf{P}(\mathbf{z})\Big)(\mathbf{P}(\mathbf{z}) \odot \mathbf{\Omega}(\mathbf{z}))'\mathbf{A}'(\mathbf{\Lambda}(\mathbf{z}) \odot \mathbf{\Phi}(\mathbf{z})) \\
(\mathbf{\Omega}(\mathbf{z}) \odot \mathbf{\Phi}(\mathbf{z}))''(\mathbf{\Lambda}(\mathbf{z}) \odot \mathbf{\Phi}(\mathbf{z})) =& \mathrm{diag}\Big(\mathbf{m} \odot (\mathbf{m}-\mathbf{1}) \odot \boldsymbol{\epsilon}^{2\mathbf{m}-\mathbf{2}} \odot \mathbf{P}(\mathbf{z})^2 \odot \mathbf{\Omega}(\mathbf{z})\Big) + O\Big(\mathrm{diag}\Big(\boldsymbol{\epsilon}^{2\mathbf{m}-\mathbf{1}}\Big)\Big).
\end{aligned} \tag{27}$$

By subtracting (27) from (25), we obtain

$$\begin{aligned}(\boldsymbol{\Omega}(\mathbf{z}) \odot \boldsymbol{\Phi}(\mathbf{z}))' (\boldsymbol{\Lambda}(\mathbf{z}) \odot \boldsymbol{\Phi}(\mathbf{z}))' - (\boldsymbol{\Omega}(\mathbf{z}) \odot \boldsymbol{\Phi}(\mathbf{z}))''(\boldsymbol{\Lambda}(\mathbf{z}) \odot \boldsymbol{\Phi}(\mathbf{z})) = \operatorname{diag}\Big(\mathbf{m} \odot \boldsymbol{\epsilon}^{\mathbf{2m-2}} \odot \mathbf{P}(\mathbf{z})^2 \odot \boldsymbol{\Omega}(\mathbf{z})\Big) \\ (\mathbf{I} + O(\operatorname{diag}(\boldsymbol{\epsilon}))). \\ \Big((\boldsymbol{\Omega}(\mathbf{z}) \odot \boldsymbol{\Phi}(\mathbf{z}))' (\boldsymbol{\Lambda}(\mathbf{z}) \odot \boldsymbol{\Phi}(\mathbf{z}))' - (\boldsymbol{\Omega}(\mathbf{z}) \odot \boldsymbol{\Phi}(\mathbf{z}))''(\boldsymbol{\Lambda}(\mathbf{z}) \odot \boldsymbol{\Phi}(\mathbf{z}))\Big)^{-1} = \\ (\mathbf{I} - O(\operatorname{diag}(\boldsymbol{\epsilon}))) \left(\operatorname{diag}\Big(\mathbf{m} \odot \boldsymbol{\epsilon}^{\mathbf{2m-2}} \odot \mathbf{P}(\mathbf{z})^2 \odot \boldsymbol{\Omega}(\mathbf{z})\Big)\right)^{-1}.\end{aligned} \tag{28}$$

By using (26), the expression for $(\boldsymbol{\Omega}(\mathbf{z}) \odot \boldsymbol{\Phi}(\mathbf{z}))'(\boldsymbol{\Lambda}(\mathbf{z}) \odot \boldsymbol{\Phi}(\mathbf{z}))$ is

$$(\boldsymbol{\Omega}(\mathbf{z}) \odot \boldsymbol{\Phi}(\mathbf{z}))'(\boldsymbol{\Lambda}(\mathbf{z}) \odot \boldsymbol{\Phi}(\mathbf{z})) = \mathbf{m} \odot \boldsymbol{\epsilon}^{\mathbf{2m-1}} \odot \mathbf{P}(\mathbf{z})^2 \odot \boldsymbol{\Omega}(\mathbf{z}) \odot (\mathbf{1} + O(\boldsymbol{\epsilon})). \tag{29}$$

From (28) and (29), we get

$$\begin{aligned}&\Big((\boldsymbol{\Omega}(\mathbf{z}) \odot \boldsymbol{\Phi}(\mathbf{z}))' (\boldsymbol{\Lambda}(\mathbf{z}) \odot \boldsymbol{\Phi}(\mathbf{z}))' - (\boldsymbol{\Omega}(\mathbf{z}) \odot \boldsymbol{\Phi}(\mathbf{z}))''(\boldsymbol{\Lambda}(\mathbf{z}) \odot \boldsymbol{\Phi}(\mathbf{z}))\Big)^{-1} (\boldsymbol{\Omega}(\mathbf{z}) \odot \boldsymbol{\Phi}(\mathbf{z}))'(\boldsymbol{\Lambda}(\mathbf{z}) \odot \boldsymbol{\Phi}(\mathbf{z})) \\ &= (\mathbf{I} - O(\operatorname{diag}(\boldsymbol{\epsilon}))) \left(\operatorname{diag}\Big(\mathbf{m} \odot \boldsymbol{\epsilon}^{\mathbf{2m-2}} \odot \mathbf{P}(\mathbf{z})^2 \odot \boldsymbol{\Omega}(\mathbf{z})\Big)\right)^{-1} \mathbf{m} \odot \boldsymbol{\epsilon}^{\mathbf{2m-1}} \odot \mathbf{P}(\mathbf{z})^2 \odot \boldsymbol{\Omega}(\mathbf{z}) \odot (\mathbf{1} + O(\boldsymbol{\epsilon})) \\ &= (\mathbf{I} - O(\operatorname{diag}(\boldsymbol{\epsilon})))\operatorname{diag}(\boldsymbol{\epsilon})(\mathbf{1} + O(\boldsymbol{\epsilon})) = \boldsymbol{\epsilon} + O(\boldsymbol{\epsilon}^2).\end{aligned} \tag{30}$$

The error equation for (21) can be written as

$$\boldsymbol{\epsilon}_{k+1} = \boldsymbol{\epsilon}_k - \Big(\boldsymbol{\epsilon}_k + O\Big(\boldsymbol{\epsilon}_k^2\Big)\Big) = O\Big(\boldsymbol{\epsilon}_k^2\Big). \tag{31}$$

The error Equation (31) for (21) indicates that the order of convergence for the proposed iterative method is quadratic. □

4. Numerical Testing

The two preconditioners $\omega(z)$ and $\lambda(z)$ produce families of iterative methods. If we define $\omega(z) = exp(\varpi z)$ and $\lambda(z) = exp(\vartheta z)$, we get the following two-parameter family of iterative methods for solving nonlinear equations that have zeros with unknown multiplicity.

$$\text{S1: } z_{k+1} = z_k - \frac{(\varpi\,\phi(z) + \phi'(z))\,\phi(z)}{(\vartheta - \varpi)\,\phi'(z)\,(\varpi + \phi(z))\,\phi'(z)^2 - \phi''(z)\,\phi(z)}.$$

Now we choose $\omega(z) = exp(\varpi\,\phi(z))$ and $\lambda(z) = (\vartheta\,\phi(z))$, and obtain the following method

$$\text{S2: } z_{k+1} = z_k - \frac{\phi'(z)\,(1 + \varpi\,\phi(z))\,\phi(z)}{\phi'(z)^2\,(1 + (\vartheta - \varpi)\,(1 + \varpi\,\phi(z))\,\phi(z)) - \phi''(z)\,\phi(z)\,(1 + \varpi\,\phi(z))}.$$

We only conducted numerical testing for the system of nonlinear equations, and the cases for the nonlinear equations are similar. It is important to test the computational convergence order (CCO) of the proposed iterative methods. In all our simulations, we adopted the following definition of CCO:

$$\text{CCO} = \frac{log(||\boldsymbol{\Phi}(\mathbf{z}_{k+1})||_\infty / ||\boldsymbol{\Phi}(\mathbf{z}_k)||_\infty)}{log(||\boldsymbol{\Phi}(\mathbf{z}_k)||_\infty / ||\boldsymbol{\Phi}(\mathbf{z}_{k-1})||_\infty)} \quad \text{or} \quad \frac{log(||\mathbf{z}_{k+1} - \boldsymbol{\kappa}||_\infty / ||\mathbf{z}_k - \boldsymbol{\kappa}||_\infty)}{log(||\mathbf{z}_k - \boldsymbol{\kappa}||_\infty / ||\mathbf{z}_{k-1} - \boldsymbol{\kappa}||_\infty)}. \tag{32}$$

For numerical simulations, three problems were selected with different multiplicities. The performance of iterative method (11) is not better, comparatively. The various choices for the preconditioners are made in Tables 1–3 for all three problems. In Table 1, we have shown that the selection of preconditioners has an influence on the numerical accuracy of computed zeros with multiplicities. Moreover, the computational cost of performing the different operation is reasonable, because in all cases, we selected preconditioners in a way that their first- and second-order Fréchet derivatives are diagonal matrices. When we selected $\mathbf{\Lambda}(\mathbf{z}) = 6 + cos(\mathbf{z})/10$ and $\mathbf{\Omega}(\mathbf{z}) = 6 + cos(\mathbf{z})/10$ for Problem 1, we achieved the best accuracy in computed zeros with different multiplicities. For the second problem, Table 2 shows that the selection of $\mathbf{\Lambda}(\mathbf{z})$ produces good accuracy. In Table 3, again the selection of both preconditioners provides the best accuracy, comparatively.

$$\text{Problem 1} = \begin{cases} \Phi_1(\mathbf{z}) = (z_1 - 1)^4 \, exp(z_2) = 0 \\ \Phi_2(\mathbf{z}) = (z_2 - 2)^5 \, (z_1 z_2 - 1) = 0 \\ \Phi_3(\mathbf{z}) = (z_3 + 4)^6 = 0 \end{cases} \tag{33}$$

$$\text{Problem 2} = \begin{cases} \Phi_1(\mathbf{z}) = z_1 z_2 = 0 \\ \Phi_2(\mathbf{z}) = z_2 z_3 = 0 \\ \Phi_3(\mathbf{z}) = z_3 z_4 = 0 \\ \Phi_4(\mathbf{z}) = z_4 z_1 = 0 \end{cases} \tag{34}$$

$$\text{Problem 3} = \begin{cases} \Phi_1(\mathbf{z}) = \sqrt{z_1 - 1}\, z_2 z_3 = 0 \\ \Phi_2(\mathbf{z}) = \sqrt{z_2 - 1}\, z_1 z_3 = 0 \\ \Phi_3(\mathbf{z}) = \sqrt{z_3 - 1}\, z_1 z_2 = 0 \end{cases} \tag{35}$$

Table 1. Problem 1: initial guess = $[2, 1, -2]$, $m = [4, 5, 6]$. CCO: computational convergence order.

	$\mathbf{\Lambda}(\mathbf{z})$	$\mathbf{\Omega}(\mathbf{z})$	Iter.	$\lvert\lvert \mathbf{z} - \boldsymbol{\kappa} \rvert\rvert_\infty$	CCO
Iterative method (21)	$\mathbf{1}$	$\mathbf{1}$	6	$O(10^{-43})$	2.0
	$6 + cos(\mathbf{z})/10$	$\mathbf{1}$	6	$O(10^{-51})$	2.05
	$1 + \mathbf{z}^3/1000$	$\mathbf{1}$	6	$O(10^{-42})$	2.0
	$exp(-\mathbf{z}/100)$	$\mathbf{1}$	6	$O(10^{-46})$	2.0
	$\mathbf{1}$	$6 + cos(\mathbf{z})/10$	6	$O(10^{-38})$	2.0
	$\mathbf{1}$	$1 + \mathbf{z}^3/1000$	6	$O(10^{-46})$	2.0
	$\mathbf{1}$	$exp(-\mathbf{z}/100)$	6	$O(10^{-39})$	2.0
	$6 + cos(\mathbf{z})/10$	$6 + cos(\mathbf{z})/10$	6	$O(10^{-41})$	2.0
	$6 + cos(\mathbf{z})/10$	$1 + \mathbf{z}^3/1000$	6	$O(10^{-65})$	2.0
	$6 + cos(\mathbf{z})/10$	$exp(-\mathbf{z}/100)$	6	$O(10^{-43})$	2.0
	$1 + \mathbf{z}^3/1000$	$1 + \mathbf{z}^3/1000$	6	$O(10^{-45})$	2.0
	$1 + \mathbf{z}^3/1000$	$6 + cos(\mathbf{z})/10$	6	$O(10^{-37})$	2.0
	$1 + \mathbf{z}^3/1000$	$exp(-\mathbf{z}/100)$	6	$O(10^{-38})$	2.0
	$exp(-\mathbf{z}/100)$	$exp(-\mathbf{z}/100)$	6	$O(10^{-41})$	2.0
	$exp(-\mathbf{z}/100)$	$exp(\mathbf{z}/100)$	6	$O(10^{-53})$	2.0
	$exp(-\mathbf{z}/100)$	$6 + cos(\mathbf{z})/10$	6	$O(10^{-40})$	2.0
	$exp(-\mathbf{z}/100)$	$1 + \mathbf{z}^3/1000$	6	$O(10^{-53})$	2.0
Iterative method (11)	$\mathbf{1}$	-	6	$O(10^{-30})$	2.0
	$6 + cos(\mathbf{z})/10$	-	6	$O(10^{-30})$	2.0
	$1 + \mathbf{z}^3/1000$	-	6	$O(10^{-30})$	2.0
	$exp(\mathbf{z}/100)$	-	6	$O(10^{-30})$	2.0

Table 2. Problem 2: initial guess = $[1,2,4,3]$, $m = [2,2,2,2]$.

	$\mathbf{\Lambda(z)}$	$\mathbf{\Omega(z)}$	**Iter.**	$\|\|\mathbf{\Phi(z)}\|\|_\infty$	**CCO**
Iterative method (21)	$\mathbf{1}$	$\mathbf{1}$	1	-	-
	$6+cos(\mathbf{z})/10$	$\mathbf{1}$	7	$O\left(10^{-2042}\right)$	3.0
	$1+\mathbf{z}^3/1000$	$\mathbf{1}$	7	$O\left(10^{-8482}\right)$	3.98
	$exp(\mathbf{z}/100)$	$\mathbf{1}$	7	$O\left(10^{-376}\right)$	2.00
Iterative method (11)	$\mathbf{1}$	-	1	-	-
	$6+cos(\mathbf{z})/10$	-	20	$O\left(10^{-23}\right)$	1.0
	$1+\mathbf{z}^3/1000$	-	20	Not converging	-
	$exp(\mathbf{z}/100)$	-	7	$O\left(10^{-443}\right)$	2.0

Table 3. Problem 3: initial guess = $[2,4,3]$, $m = [1/2,1/2,1/2]$.

	$\mathbf{\Lambda(z)}$	$\mathbf{\Omega(z)}$	**Iter.**	$\|\|\mathbf{\Phi(z)}\|\|_\infty$	**CCO**
Iterative method (21)	$\mathbf{1}$	$\mathbf{1}$	12	$O\left(10^{-2011}\right)$	2.00
	$6+cos(\mathbf{z})/10$	$\mathbf{1}$	12	$O\left(10^{-1914}\right)$	2.00
	$1+\mathbf{z}^3/1000$	$\mathbf{1}$	12	$O\left(10^{-1248}\right)$	2.00
	$exp(-\mathbf{z}/10)$	$\mathbf{1}$	12	$O\left(10^{-2767}\right)$	2.00
	$exp(-\mathbf{z}/10)$	$exp(\mathbf{z}/10000)$	12	$O\left(10^{-2110}\right)$	2.00
	$exp(-\mathbf{z}/10)$	$exp(-\mathbf{z}/10000)$	12	$O\left(10^{-2771}\right)$	2.00
Iterative method (11)	$\mathbf{1}$	-	1	-	-
	$6+cos(\mathbf{z})/10$	-	12	$O\left(10^{-56}\right)$	2.00
	$1+\mathbf{z}^3/1000$	-	20	Not converging	-
	$exp(-\mathbf{z}/10)$	-	7	$O\left(10^{-35}\right)$	2.00

5. Conclusions

The inclusion of preconditioners in the existing iterative methods for finding zeros with multiplicities for solving a system of nonlinear equations gives benefits in numerical stability and numerical accuracy. The proposed methodology is equally effective for nonlinear and systems of nonlinear equations. It is assumed in all cases that the preconditioners should be non-zero, because in this way, it does not affect the zeros of nonlinear or systems of nonlinear equations. The different selections of preconditioners provide different families of iterative methods. The claimed order of convergence is also verified by computing the computational order of convergence in all numerical simulations. Study of the dynamics of nonlinear preconditioners for finding zeros with multiplicities of nonlinear equations and systems of nonlinear equations could be an interesting topic for research.

Author Contributions: First author established the idea and all other authors contributed equally in the article.

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

References

1. Ahmad, F.; Tohidi, E.; Carrasco, J.A. A parameterized multi-step Newton method for solving systems of nonlinear equations. *Numer. Algorithms* **2016**, *71*, 631.
2. Ullah, M.Z.; Serra-Capizzano, S.; Ahmad, F. An efficient multi-step iterative method for computing the numerical solution of systems of nonlinear equations associated with ODEs. *Appl. Math. Comput.* **2015**, *250*, 249–259.
3. Ahmad, F.; Tohidi, E.; Ullah, M.Z.; Carrasco, J.A. Higher order multi-step Jarratt-like method for solving systems of nonlinear equations: Application to PDEs and ODEs. *Comput. Math. Appl.* **2015**, *70*, 624–636.
4. Alaidarous, E.S.; Ullah, M.Z.; Ahmad, F.; Al-Fhaid, A.S. An Efficient Higher-Order Quasilinearization Method for Solving Nonlinear BVPs. *J. Appl. Math.* **2013**, *2013*, 259371.
5. Ullah, M.Z.; Soleymani, F.; Al-Fhaid, A.S. Numerical solution of nonlinear systems by a general class of iterative methods with application to nonlinear PDEs. *Numer. Algorithms* **2014**, *67*, 223–242.
6. Montazeri, H.; Soleymani, F.; Shateyi, S.; Motsa, S.S. On a New Method for Computing the Numerical Solution of Systems of Nonlinear Equations. *J. Appl. Math.* **2012**, *2012*, 751975, doi:10.1155/2012/751975.
7. Cordero, A.; Hueso, J.L.; Martinez, E.; Torregrosa, J.R. A modified Newton-Jarratt's composition. *Numer. Algorithms* **2010**, *55*, 87–99.
8. Chun, C. A method for obtaining iterative formulas of order three. *Appl. Math. Lett.* **2007**, *20*, 1103–1109.
9. Chun, C. On the construction of iterative methods with at least cubic convergence. *Appl. Math. Comput.* **2007**, *189*, 1384–1392.
10. Chun, C. Some variant of Chebshev-Halley method free from second derivative. *Appl. Math. Comput.* **2007**, *191*, 1384–1392.
11. Osada, N. Improving the order of convergence of iterative functions. *J. Comput. Appl. Math.* **1998**, *98*, 311–315.
12. Noor, M.A.; Shah, F.A. Variational iteration technique for solving nonlinear equations. *J. Appl. Math. Comput.* **2009**, *31*, 247–254.
13. Noor, M.A.; Shah, F.A.; Noor, K.I.; Al-Said, E. Variational iteration technique for finding multiple roots of nonlinear equations. *Sci. Res. Essays* **2011**, *6*, 1344–1350.
14. Noor, M.A.; Shah, F.A. A family of iterative schemes for finding zeros of nonlinear equations having unknown multiplicity. *Appl. Math. Inf. Sci.* **2014**, *8*, 2367–2373.
15. Shah, F.A.; Noor, M.A.; Batool, M. Derivative-free iterative methods for solving nonlinear equations. *Appl. Math. Inf. Sci.* **2014**, *8*, 2189–2193.
16. Ortega, J.M.; Rheinbodt, W.C. *Iterative Solution of Nonlinear Equations in Several Variables*; Academic Press Limited: London, UK, 1970.
17. Traub, J.F. *Iterative Methods for the Solution of Equations*; Prentice-Hall: Englewood Cliffs, NJ, USA, 1964.
18. Hueso, J.L.; Martinez, E.; Torregrosa, J.R. Modified Newton's method for systems of nonlinear equations with singular Jacobian. *J. Comput. Appl. Math.* **2009**, *224*, 77–83.
19. Wu, X. Note on the improvement of Newton's method for systems of nonlinear equations. *Appl. Math. Comput.* **2007**, *189*, 1476–1479.
20. Noor, M.A.; Waseem, M.; Noor, K.I.; Al-Said, E. Variational iteration technique for solving a system of nonlinear equations. *Optim Lett.* **2013**, *7*, 991–1007, doi:10.1007/s11590-012-0479-3.
21. Burden, R.L.; Faires, J.D. *Numerical Analysis*; PWS Publishing Company: Boston, MA, USA, 2001.
22. McNamee, J.M. *Numerical Methods for Roots of Polynomials, Part I*; Elsevier: Amsterdam, The Netherlands, 2007.
23. Noor, M.A.; Shah, F.A. A Family of Iterative Schemes for Finding Zeros of Nonlinear Equations having Unknown Multiplicity. *Appl. Math. Inf. Sci.* **2014**, *8*, 2367–2373.
24. Ahmad, F.; S-Capizzano, S.; Ullah, M.Z.; Al-Fhaid, A.S. A Family of Iterative Methods for Solving Systems of Nonlinear Equations Having Unknown Multiplicity. *Algorithms* **2016**, *9*, 5.