Hindawi Publishing Corporation
International Journal of Mathematics and Mathematical Sciences
Volume 2013, Article ID 814587, 6 pages
http://dx.doi.org/10.1155/2013/814587

Research Article

An Enhanced Matrix-Free Secant Method via Predictor-Corrector Modified Line Search Strategies for Solving Systems of Nonlinear Equations

M. Y. Waziri[1,2] and Z. A. Majid[1,3]

[1] *Department of Mathematics, Faculty of Science, Universiti Putra Malaysia, 43400 Serdang, Selangor, Malaysia*
[2] *Department of Mathematical Sciences, Faculty of Science, Bayero University Kano, Kano, Nigeria*
[3] *Institute for Mathematical Research, Universiti Putra Malaysia, 43400 Serdang, Selangor, Malaysia*

Correspondence should be addressed to M. Y. Waziri; mywaziri@gmail.com

Received 28 September 2012; Revised 24 December 2012; Accepted 7 January 2013

Academic Editor: Paolo Ricci

Diagonal updating scheme is among the cheapest Newton-like methods for solving system of nonlinear equations. Nevertheless, the method has some shortcomings. In this paper, we proposed an improved matrix-free secant updating scheme via line search strategies, by using the steps of backtracking in the Armijo-type line search as a step length predictor and Wolfe-Like condition as corrector. Our approach aims at improving the overall performance of diagonal secant updating scheme. Under mild assumptions, the global convergence results have been presented. Numerical experiments verify that the proposed approach is very promising.

1. Introduction

Consider the problem

$$F(x) = 0, \tag{1}$$

with $F : R^n \to R^n$.

The mapping F is assumed to satisfy the following standard assumptions:

(1) F is continuously differentiable in an open convex set Φ;

(2) there exists a solution x^* of (1) in Φ such that $F(x^*) = 0$ and $F'(x^*) \neq 0$;

(3) the Jacobian $F'(x)$ is local Lipschitz continuous at x^*.

The famous scheme for finding the solution to (1) is the Newton method. The method is simple to implement, and it produces an iterative sequence $\{x_k\}$ from any given initial guess x_0 in the neighborhood of x^* via

$$x_{k+1} = x_k - \left(F'(x_k)\right)^{-1} F(x_k), \tag{2}$$

where $k = 0, 1, 2 \ldots n$.

The attractive features of this method are that it is easy to implement and converges rapidly [1]. However, the major difficulty of Newton-type method is matrix storage requirements especially when handling large systems of nonlinear equations [1]. To overcome such difficulty, the simple modification on the Newton method is the fixed Newton method. Fixed Newton method for the determination of solution x^* is given by

$$x_{k+1} = x_k - \left(F'(x_0)\right)^{-1} F(x_k), \quad k = 0, 1, 2, \ldots. \tag{3}$$

This method avoids computation and storing the Jacobian in each iterations (except at $k = 0$). However, it still requires to solve systems of n linear equations and consumes more CPU time as the system's dimension increases [2, 3].

Quasi-Newton method is another variant of Newton-type methods. It replaces the Jacobian or its inverse with an approximation which can be updated at each iteration and is given as

$$x_{k+1} = x_k - B_k^{-1} F(x_k), \tag{4}$$

where the matrix B_k is the approximation of the Jacobian at x_k. The main idea behind quasi-Newton's method is to

eliminate the evaluation cost of the Jacobian matrix [4, 5]. However, the most critical part of such quasi-Newton method is storing the full matrix of the approximate Jacobian, which can be a very expensive task as the dimension of systems increases [6]. In contrast, this paper presents an improved variant of quasi-Newton update via the steps of backtracking in the Armijo-type line search and Wolfe–like condition and incorporating restarting strategy whenever the updating matrix is singular or nearly singular. The anticipation has been to improve the overall performance of the diagonal quasi-Newton method.

We organized the rest of this paper as follows. In the next section, we present the proposed method. Section 3 presents convergence results. Numerical experiments are reported in Section 4, and finally, conclusion is given in Section 5.

2. The Improved Secant Diagonal Updating

It is well known that it is not always feasible to compute the full elements of the Jacobian matrix of the given nonlinear function or it may be very expensive. We often have to approximate the Jacobian matrix by some other approach, and the famous method of doing so is quasi-Newton's method [4]. The basic idea underlining this approach has been to reduce the evaluation cost of the Jacobian matrix. This new approach generates a sequence of points $\{x_k\}$ via

$$x_{k+1} = x_k - \alpha_k B_k F\left(x_k\right), \tag{5}$$

where α_k is a step length, and B_k is a diagonal approximation of the inverse Jacobian matrix which can be updated in each iterations. To achieve this, we incorporate some new line search strategies, via the steps of backtracking in the Armijo-type line search as predictor and then improved via Wolfe-like condition as a corrector. The Armijo rule is among the inexact line search methods which guarantees a sufficient degree of accuracy to ensure the algorithm convergence. Nevertheless, the scheme needs high floating points operations and function call. In this paper, we present a simple line search strategy, which is less computational cost, floating points operations and CPU time consumptions respectively compared to classical Armijor line search. The approach is as follows.

Given $\epsilon \in (0, 1)$ and $\sigma < 1$, the proposed approach finds the appropriate α, such that

$$\left\|F\left(x_k + \alpha_k d_k\right)\right\| \leq \sigma \left\|F\left(x_k\right)\right\|. \tag{6}$$

In addition, the new strategy is implemented in an iterative way using a fixed initial value of α as follows.

Algorithm 1 (Armijo-Like).

Step 0. Set $K = 0$, $\alpha_0 > 0$, and $\sigma < 1$.

Step 1. $\|F(x_{k+1})\| \leq \sigma\|F(x_k)\|$. Choose α_k as the step size; stop. Otherwise $\alpha_{k+1} = \alpha_k/2$, $k = k + 1$.

Waziri et al. [7] have set the step length ($\alpha_k = 1$, for all k); this approach is mostly used in many Newton-like methods. Here, we continue in the spirit of diagonal updating, using a new line search strategy to obtain a good step length (α_k) in every iteration, anticipating to produce a more accurate approximation of the Jacobian inverse matrix and then employing restating strategy whenever the updating matrix is undefined. To this end, B_{k+1} would be obtained almost similar to the diagonal updating scheme presented in [7] in which $s_k = -\alpha_k B_k F(x_k)$ instead of $s_k = -B_k F(x_k)$. Now, let the deviation between B_{k+1} and B_k denoted as $\Phi_k = B_{k+1} - B_k$ be minimized under some norms; the optimal solution is given as

$$\Phi_k = \frac{\left(y_k^T s_k - y_k^T B_k y_k\right)}{\operatorname{tr}\left(\Psi_k^2\right)}\Psi_k, \tag{7}$$

where $\Psi_k = \operatorname{diag}((y_k^{(1)})^2, (y_k^{(2)})^2, \ldots, (y_k^{(n)})^2)$, $\sum_{i=1}^{n}(y_k^{(i)})^4 = \operatorname{tr}(\Psi_k^2)$, $y_k = F(x_{k+1}) - F(x_k)$, $s_k = x_{k+1} - x_k$, and tr is the trace operation, respectively. The updated formula for the proposed matrix B_k is given as [7]

$$B_{k+1} = B_k + \frac{\left(y_k^T s_k - y_k^T B_k y_k\right)}{\operatorname{tr}\left(\Psi_k^2\right)}\Psi_k. \tag{8}$$

To safeguard on the possibilities of generating undefined B_{k+1}, we let $B_{k+1} = B_0$, whenever this situation happens

$$B_{k+1} = \begin{cases} B_k + \dfrac{\left(y_k^T s_k - y_k^T B_k y_k\right)}{\operatorname{tr}\left(\Psi_k^2\right)}\Psi_k, & \operatorname{tr}\left(\Psi_k^2\right) \neq 0, \\ B_0, & \text{otherwise.} \end{cases} \tag{9}$$

Now, we can describe the algorithm for our proposed approach as follows.

Algorithm 2 (EMFM).

Step 1. Choose an initial guess x_0, $\sigma \in (0, 1)$, $B_0 = I_n$, $\alpha_0 > 0$ and let $k := 0$.

Step 2. Compute $F(x_k)$, and if $\|F(x_k)\| \leq 10^{-4}$, stop.

Step 3. Compute $d = -F(x_k)B_k$.

Step 4. If $\|F(x_k + \alpha_k d_k)\| \leq \sigma\|F(x_k)\|$, retain α_k and go to 5. Otherwise, set $\alpha_{k+1} = \alpha_k/2$ and repeat 4.

Step 5. If $\|F(x_k + \alpha_k d_k) - F(x_k)\| \geq \|F(x_k + \alpha_k d_k)\| - \|F(x_k)\|$, retain α_k and go to 6. Otherwise, set $\alpha_{k+1} = \alpha_k \times 1.1$ and repeat 5.

Step 6. Let $x_{k+1} = x_k + \alpha_k d_k$.

Step 7. If $\|x_{k+1} - x_k\|_2 + \|F(x_k)\|_2 \leq 10^{-4}$, stop. Otherwise go to Step 8.

Step 8. If $\|\Delta F_k\|_2 \geq \epsilon_1$, where $\epsilon_1 = 10^{-4}$, compute B_{k+1}, and if not, $B_{k+1} = B_0$.

Step 9. Set $k := k + 1$ and go to 2.

3. Convergence Analysis

We present the convergence result of EMFM method by proving the existence of the step length $\alpha_k > 0$. We will make the following assumptions on nonlinear system F.

Assumption 3. (i) F is differentiable in an open convex set E in $\Re^n$.

(ii) There exists $x^* \in E$ such that $(x^*) = 0$, and $F'(x)$ is continuous for all x.

(iii) $F(x)$ satisfies Lipschitz condition of order one; that is, there exists a positive constant μ such that

$$\|F(x) - F(y)\| \leq \mu \|x - y\|, \tag{10}$$

for all $x, y \in \Re^n$.

(iv) There exist constants $c_1 \leq c_2$ such that $c_1\|\omega\|^2 \leq \omega^T F'(x)\omega \leq c_2\|\omega\|^2$ for all $x \in E$ and $\omega \in \Re^n$.

To this end, we proceed by given the following result for the step length generated by the proposed strategies.

Theorem 4. *Assume that F is a strictly convex function. Suppose that the new strategies are employed with $d_k \neq 0$ and positive α_k exist, for all sufficiently large k. Then, the iterates $\{x_k\}$ generated by the line search algorithm have the property that*

$$\lim \|F(x_k + \alpha_k d_k)\| = 0, \quad as\ k \longrightarrow \infty. \tag{11}$$

Proof. From condition (iii) of Assumption 3 and the fact that $s_k = x_{k+1} - x_k$, we have

$$\begin{aligned} \|F(x_k) - F(x_k + \alpha_k d_k)\|_F &\leq \mu \|s_k\|_F \\ &= \mu \|\alpha d_k\|_F \\ &\leq \mu |\alpha| \|d_k\|_F. \end{aligned} \tag{12}$$

Recall that

$$\|F(x) - F(x_k + \alpha_k d_k)\|_F \leq \|F(x)\|_F - \|F(x_k + \alpha_k d_k)\|_F. \tag{13}$$

Equation (13) gives

$$\begin{aligned} \|F(x_k)\|_F - \|F(x_k + \alpha_k d_k)\|_F &\leq \alpha\mu \|d_k\|_F \\ &= \mu \|\alpha d_k\|_F \\ &\leq \mu |\alpha| \|d_k\|_F. \end{aligned} \tag{14}$$

Since $\|d_k\| \neq 0$ and $\|F(x_k + \alpha_k d_k)\| \leq \sigma\|F(x_k)\|$, hence, it follows that

$$\begin{aligned} \|F(x_k)\|_F - \sigma\|F(x_k)\|_F &\leq \alpha\mu \|d_k\|_F, \\ (1 - \sigma)\|F(x_k)\|_F &\leq \alpha\mu \|d_k\|_F. \end{aligned} \tag{15}$$

After little simplifications, we obtain

$$\alpha \geq \frac{(1 - \sigma)\|F(x_k)\|_F}{\mu \|d_k\|_F} > 0. \tag{16}$$

Hence, α_k exist and are positive.

We continue to show the convergence of the iterates $\{x_k\}$ by recalling that

$$\|F(x_k + \alpha_k d_k)\| \leq \sigma \|F(x_k)\|. \tag{17}$$

Then, we have

$$\begin{aligned} \|F(x_k + \alpha_k d_k)\| &\leq \sigma^2 \|F(x_k)\| \\ &\leq \sigma^3 \|F(x_k)\| \end{aligned} \tag{18}$$

for a finite k, and (18) yields

$$\|F(x_k + \alpha_k d_k)\| \leq \sigma^k \|F(x_k)\|. \tag{19}$$

Since the algorithm terminates at $\|F(x_k)\| = 0$, it follows from (19) that

$$\frac{\|F(x_k + \alpha_k d_k)\|}{\|F(x_k)\|} \leq \sigma^k. \tag{20}$$

We have

$$\lim \frac{\|F(x_k + \alpha_k d_k)\|}{\|F(x_k)\|} \leq \lim \sigma^k, \quad \text{as } k \longrightarrow \infty. \tag{21}$$

Therefore, due to $\sigma < 1$, it implies that

$$\lim \sigma^k \longrightarrow 0, \quad \text{as } k \longrightarrow \infty. \tag{22}$$

hence

$$\lim \|F(x_k + \alpha_k d_k)\| = 0, \quad \text{as } k \longrightarrow \infty. \tag{23}$$

□

To show the convergence results of EMFM method, we require to show that the updating matrix B_k is bounded above and below by some positive constants. Hence, we can state the following result on the boundedness of $\{\|\Phi_k\|_F\}$ by assuming that, without loss of generality, the updating matrix (8) is always used; then, we have the following.

Theorem 5. *Let F satisfy Assumption 3, and let $\|y_k\| \neq 0$ for all finite k. Let $\{B_k\}$ be the sequence generated by* (8)*. If the given nonsingular B_0 satisfies*

$$\theta \leq B_0^i \leq \omega, \quad i = 1, 2, ..., n, \tag{24}$$

for some constants θ and ω, then the sequence $\{\|B_k\|_F\}$ is bounded for all finite k.

Proof. Since $B_{k+1} = B_k + \Phi_k$, it follows that

$$\|B_{k+1}\|_F \leq \|B_k\|_F + \|\Phi_k\|_F. \tag{25}$$

For $k = 0$ and assuming $B_0 = I$, we have

$$\begin{aligned} \left|\Phi_0^{(i)}\right| &= \left|\frac{y_0^T s_0 - y_0^T B_0 y_0}{\operatorname{tr}(\Psi_0^2)}\left(y_0^{(i)}\right)^2\right| \\ &\leq \frac{\left|y_0^T s_0 - y_0^T B_0 y_0\right|}{\operatorname{tr}(\Psi_0^2)}\left(y_0^{(\max)}\right)^2, \end{aligned} \tag{26}$$

where $(y_0^{(\max)})^2$ is the largest element among $(y_0^{(i)})^2$, $i = 1, 2, \ldots, n$.

After multiplying (26) by $(y_0^{(\max)})^2/(y_0^{(\max)})^2$ and substituting $\operatorname{tr}(\Psi_0^2) = \sum_{i=1}^{n}(y_0^{(i)})^4$, we have

$$\left|\Phi_0^{(i)}\right| \leq \frac{\left|y_0^T s_0 - y_0^T B_0 y_0\right|}{\left(y_0^{(\max)}\right)^2 \sum_{i=1}^{n}\left(y_0^{(i)}\right)^4}\left(y_0^{(\max)}\right)^4. \tag{27}$$

Since $(y_0^{(\max)})^4 / \sum_{i=1}^{n}(y_0^{(i)})^4 \leq 1$, then (27) turns into

$$\left|\Phi_0^{(i)}\right| \leq \frac{\left|y_0^T F'(x)\ \ y_0 - y_0^T B_0 y_0\right|}{\left(y_0^{(\max)}\right)^2}. \tag{28}$$

From Assumption 3 and $B_0 = I$, (28) becomes

$$\left|\Phi_0^{(i)}\right| \leq \frac{|c-1|\left(y_0^T y_0\right)}{\left(y_0^{(\max)}\right)^2}, \tag{29}$$

where $c = \max\{|c_1|, |c_2|\}$.

Since $(y_0^{(i)})^2 \leq (y_0^{(\max)})^2$ for $i = 1, \ldots, n$, it follows that

$$\left|\Phi_0^{(i)}\right| \leq \frac{n|c-1|\left(y_0^{(\max)}\right)^2}{\left(y_0^{(\max)}\right)^2}. \tag{30}$$

Hence, we obtain

$$\left\|\Phi_0\right\|_F \leq n^{3/2}|c-1|. \tag{31}$$

Suppose that $\eta = n^{3/2}|c-1|$; then,

$$\left\|\Phi_0\right\|_F \leq \eta. \tag{32}$$

From the fact that $\|B_0\|_F = \sqrt{n}$, it follows that

$$\left\|B_1\right\|_F \leq \beta, \tag{33}$$

where $\beta = \sqrt{n} + \eta > 0$. □

4. Numerical Results

In this section, we consider some benchmark problems to illustrate the performance of the method proposed in this paper for solving large-scale systems of nonlinear equations when compared to some Newton-like methods. The computations are performed in MATLAB 7.0 using double precision computer, and the stopping rule used is

$$\left\|s_k\right\| + \left\|F(x_k)\right\| \leq 10^{-4}. \tag{34}$$

The identity matrix has been chosen as an initial approximate Jacobian inverse. We further design the codes to terminate whenever one of the following happens:

(i) the number of iteration is at least 250, but no point of x_k that satisfies (34) is obtained;

(ii) CPU time in seconds reaches 250;

(iii) There is insufficient memory to initiate the run.

The performances of these methods are be compared in terms of number of iterations and CPU time in seconds. In the following, some details on the benchmarks test problems are presented.

Problem 1. System of n nonlinear equations is as follows:

$$f_i(x) = x_i - 3x_i\left(\frac{\sin x_i}{3} - 0.66\right) + 2, \quad i = 1, 2, \ldots, n,\ \ x_0 = (3, 3, \ldots, 3). \tag{35}$$

Problem 2. Extended Trigonometric function of Spedicator [8] is as follows:

$$f_i(x) = n - \sum_{j=1}^{n}\cos x_j + i\left(1 - \cos x_i\right) - \sin x_i + \exp^{(1-\cos x_i)} - \sum_{j=1}^{n}\left(x_j + 1\right), \quad i = 1, \ldots, n,\ \ x_0 = \left(\frac{1}{n}, \frac{1}{n}, \ldots, \frac{1}{n}\right). \tag{36}$$

Problem 3. System of n nonlinear equations is as follows:

$$f_i(x) = x_i - 0.1x_{i+1}^2, \qquad f_n(x) = x_n - 0.1x_1^2, \quad i = 1, \ldots, n-1,\ \ x_0 = (7, 7, \ldots, 7). \tag{37}$$

(A1) Nonadiabatic Stirred Tank Reactors. The first application is the model of [9]. The model deals with two continuous nonadiabatic stirred tank reactors. The reactors are in a series, at steady state having a recycle component, and with an exothermic first-order irreversible reaction. By eliminating certain variables, the model results into two nonlinear systems of equations as follows:

$$f_1 = (1-\lambda)\left[\frac{D}{10\left(1+\beta_1\right)} - x_1\right]\exp\left(\frac{10x_1}{1+\left(10x_1/\gamma\right)}\right) - x_1,$$
$$f_2 = x_1 - \left(1+\beta_2\right)x_2 + (1-\lambda) \times \left[\frac{D}{10} - \beta_1 x_1 - \left(1+\beta_2 x_2\right)\right]\exp\left(\frac{10x_2}{1+\left(10x_2/\gamma\right)}\right). \tag{38}$$

The dimensionless temperatures of the two reactors are represented by x_1 and x_2. The parameters λ, γ, β_1, β_2, and D are given as 1, 1000, 2, 2, and 22, respectively.

(A2) Navigation by Range Measurements. Consider two beacons determining position by measuring the distances $r1$ and $r2$, where $(p1, p2)$ is the position of beacon 1, $(q1, q2)$ is the position of beacon 2, and (u, v) is an unknown point (see Figure 1). This can be modeled into a two nonlinear systems of equations as follows:

$$f_1 = \sqrt{(p_1-u)^2 + (q_1-v)^2} - r_1 = 0,$$
$$f_1 = \sqrt{(p_2-u)^2 + (q_2-v)^2} - r_2 = 0. \tag{39}$$

Table 1: Numerical comparison of NM, FN, BM, I-VDN, and EMFM methods.

Prob	Dim	NM		FN		BM		I-VDN		EMFM	
		NI	CPU	NI	CPU	NI	CPU	NI	CPU	NI	CPU
1	25	4	0.031	11	0.016	8	0.015	8	0.006	6	0.001
2	25	4	0.062	11	0.031	8	0.016	8	0.002	7	0.001
3	25	6	0.031	—	—	10	0.031	10	0.001	7	0.001
1	50	4	0.046	11	0.031	8	0.046	8	0.011	6	0.008
2	50	4	0.109	11	0.062	8	0.034	8	0.010	7	0.005
3	50	6	0.156	—	—	10	0.124	10	0.008	7	0.004
1	100	4	0.064	11	0.032	8	0.048	8	0.014	6	0.010
2	100	4	0.125	11	0.046	8	0.032	8	0.015	7	0.011
3	100	6	0.508	—	—	10	0.187	10	0.031	7	0.015
1	1000	4	1.919	12	0.7332	9	2.122	9	0.033	6	0.018
2	1000	4	2.277	12	0.998	8	1.778	8	0.049	5	0.031
3	1000	6	88.031	—	—	10	13.057	10	0.041	7	0.024

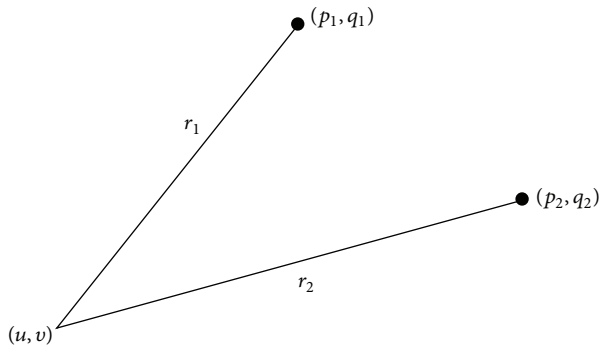

Figure 1

The parameters (p_1, q_1), (p_2, q_2), r_1, and r_2 are given as (10, 10), (10, −10), 14, and 16, respectively.

Table 1 shows the number of iterations and CPU time for these five methods, respectively. In Table 1, the value "Dim" denotes the dimension of the systems, "NI" denotes number of iterations, while "CPU" is the CPU time in seconds respectively. We analyze the performance of each method via execution time, floating points operations, and storage locations. One can observe that EMFM has the smallest number of iterations compared to the classical diagonal updating (I-VDN) proposed by Waziri et al. [7]. This shows that the line search strategies presented in this paper have increased the convergence speed of the classical diagonal updating method.

If we compare the performance of all methods, in terms of CPU time, it is clear that EMFM method consumes less CPU time than the others and still keeping memory requirement and CPU time in seconds to only $O(n)$. All five methods are able to obtain the solution $(x^* = -1.1243, 1.5001)$ of A2, but EMFM method consumes less CPU time in second (0.001) compared to the other 4 methods. Moreover, for A1, still proposed method has shown a promising performance with less storage locations (2 locations) whereas NM, FN, and BM, respectively, required 4 locations for each.

5. Conclusion

From the fact that there is a rapid development in research on solving nonlinear systems, nevertheless, the dimension of the nonlinear system is most of the times so large that it requires a very costly arithmetic operations when using some other Newton-like methods; so we usually employ cheap iterative approach, and the good candidate is classical diagonal updating. Notwithstanding, the classical updating scheme has some lapses. One is that it usually needs high number of iterations. The other is that the method generally has slow convergence behavior. This paper proposes an enhanced diagonal secant updating scheme based on the steps of backtracking in the Armijo-type line search and then improved via Wolfe-like condition. Our approach aims at improving the overall performance of diagonal secant updating scheme. The algorithm only requires to store a row vector while ignoring all the off and low diagonal elements and therefore largely reduces memory locations. In addition, as it uses two line search strategies (predictor and corrector) to obtain a new iterates point, the spectral properties of the diagonal updating scheme is improved, and rapid convergence property is gained.

Computational experiment suggests that it is very vital for diagonal updating scheme to use line search strategy. EMFM method has very good solving speed and the best performance among the Newton-like methods. Finally, it can be concluded that this approach would certainly be quite useful for solving large-scale systems of nonlinear equations.

References

[1] J. E. Dennis, Jr. and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice Hall, Englewood Cliffs, NJ, USA, 1983.

[2] M. Y. Waziri, W. J. Leong, M. A. Hassan, and M. Monsi, "A New Newton method with diagonal Jacobian approximation for

systems of Non-Linear equations," *Journal of Mathematics and Statistics*, vol. 6, no. 3, pp. 246–252, 2010.

[3] M. Y. Waziri, W. J. Leong, and M. Mamat, "A two-step matrix-free secant method for solving large-scale systems of nonlinear equations," *Journal of Applied Mathematics*, vol. 2012, Article ID 348654, 9 pages, 2012.

[4] C. G. Broyden, "A class of methods for solving nonlinear simultaneous equations," *Mathematics of Computation*, vol. 19, pp. 577–593, 1965.

[5] K. Natasa and L. Zorna, "Newton-like method with modification of the right-hand vector," *Mathematics of Computation*, vol. 71, pp. 237–250, 2001.

[6] C. T. Kelley, *Iterative Methods for Linear and Nonlinear Equations*, vol. 16, SIAM, Philadelphia, Pa, USA, 1995.

[7] M. Y. Waziri, W. J. Leong, and M. A. Hassan, "Diagonal Broyden-like method for large-scale systems of nonlinear equations," *Malaysian Journal of Mathematical Sciences*, vol. 6, no. 1, pp. 59–73, 2012.

[8] E. Spedicator, "Computational experience with quas-Newton algorithms for minimization problems of moderately large size," Tech. Rep. CISE-N-175, Segrate, Milano, Italy, 1975.

[9] J. Sinkule, "Multiplicity and stability in a sequence of two nonadiabatic non-isothermal CSTR," *Chemical Engineering Sceinces*, vol. 35, pp. 987–996, 1980.