symmetry

Article

Two Modified Single-Parameter Scaling Broyden–Fletcher–Goldfarb–Shanno Algorithms for Solving Nonlinear System of Symmetric Equations

Jie Guo and Zhong Wan *

School of Mathematics and Statistics, Central South University, Changsha 410083, China; guojie@csu.edu.cn
* Correspondence: wanmath@csu.edu.cn

Abstract: In this paper, we develop two algorithms to solve a nonlinear system of symmetric equations. The first is an algorithm based on modifying two Broyden–Fletcher–Goldfarb–Shanno (BFGS) methods. One of its advantages is that it is more suitable to effectively solve a small-scale system of nonlinear symmetric equations. In contrast, the second algorithm chooses new search directions by incorporating an approximation method of computing the gradients and their difference into the determination of search directions in the first algorithm. In essence, the second one can be viewed as an extension of the conjugate gradient method recently proposed by Lv et al. for solving unconstrained optimization problems. It was proved that these search directions are sufficiently descending for the approximate residual square of the equations, independent of the used line search rules. Global convergence of the two algorithms is established under mild assumptions. To test the algorithms, they are used to solve a number of benchmark test problems. Numerical results indicate that the developed algorithms in this paper outperform the other similar algorithms available in the literature.

Keywords: nonlinear system of equations; computational method; algorithms; conjugate gradient method; global convergence

MSC: 90c25; 90C30

check for updates

Citation: Guo, J.; Wan, Z. Two Modified Single-Parameter Scaling Broyden–Fletcher–Goldfarb–Shanno Algorithms for Solving Nonlinear System of Symmetric Equations. *Symmetry* **2021**, *13*, 970. https://doi.org/10.3390/sym13060970

Academic Editor: Aviv Gibali

Received: 23 April 2021
Accepted: 26 May 2021
Published: 30 May 2021

1. Introduction

In this paper, we study solution methods of the following nonlinear system of symmetric equations:

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

where $F : \mathbb{R}^n \to \mathbb{R}^n$ is a continuously differentiable function, and its Jacobian $J(x) = \nabla F(x)$ is symmetric, i.e., $J(x) = J(x)^T$. Such a problem is closely related with many scientific problems, such as unconstrained optimization problems, equality constrained mathematical programming problems, discretized two-point boundary value problems, and discretized elliptic boundary value problems (see Chapter 1 in [1]). For example, when F is the gradient mapping of an objective function $f : \mathbb{R}^n \to \mathbb{R}$, (1) is just the first order necessary condition for a local minimizer of the following problem:

$$\min\{f(x), x \in \mathbb{R}^n\}. \tag{2}$$

For the equality constrained mathematical programming problems:

$$\begin{array}{ll} \min & f(z) \\ \text{s.t.} & h(z) = 0, \end{array} \tag{3}$$

where $h : \mathbb{R}^n \to \mathbb{R}^m$ is a vector-valued function. The Karush–Kuhn–Tucker (KKT) conditions (see Chapter 8 in [2]) for Problem (3) is also the system (1) with $x = (z, v) \in \mathbb{R}^{n+m}$, and:

$$F(z, v) = \begin{pmatrix} \nabla f(z) + \nabla h(z)v \\ h(z) \end{pmatrix}. \tag{4}$$

Among various methods for solving (1), the Newton method needs to compute the Jacobian matrix $J(x)$, and requires that $J(x)$ is nonsingular at each iterative point. Due to this stringent condition, the Newton method is not applicable in a general case.

Li and Fukushima [3] proposed a Gauss–Newton method to solve the symmetric system of equations, which can ensure that an approximate residual of $\|F(x)\|$ is descent. Gu et al. [4] modified the method in [3] such that the residual $\|F(x)\|$ is descent. As a generalization of the method in [5] for solving smooth unconstrained optimization, Zhou [6] presented an inexact modified BFGS method to solve the symmetric system of equations by approximately computing the gradient of the residual square. In [7], Wang and Zhu proposed an inexact-Newton via GMRES (generalized minimal residual) subspace method without line search technique for solving symmetric nonlinear equations. The iterative direction was obtained by solving the Newton equation of the system of nonlinear equations with the GMRES algorithm. Yuan and Yao [8] also proposed a BFGS method for solving symmetric nonlinear equations and the method possesses a good property that the generated sequence of the quasi-Newton matrix is positive definite. However, since the search direction is generated by solving a system of linear equations in these methods, all of them are not applicable to solving large-scale problems.

For large-scale symmetric nonlinear equations, Li and Wang [9] proposed a modified Fletcher–Reeves derivative-free method, as an extension of the conjugate gradient method [10]. Similarly, as an extension of descent conjugate gradient methods in [11] for unconstrained optimization, Xiao et al. [12] presented a family of derivative-free methods for symmetric equations, and established the global convergence under some appropriate conditions, and showed their effectiveness by numerical experiments. Zhou and Shen [13] presented an efficient iterative method for solving large-scale symmetric equations, as an extension of the three-term PRP conjugate gradient method in [14] for solving unconstrained optimization problems. Liu and Feng [15] proposed a norm descent derivative-free algorithm for solving large-scale nonlinear symmetric equations, as an extension of the three-term conjugate gradient method in [16] for solving unconstrained optimization problems. More details can be seen in [17–20].

Our motivation in this paper is to develop two algorithms to solve Problem (1). Firstly, based on the single-parameter scaling memoryless BFGS method proposed by Lv et al. [21] and the modification of the BFGS method in [22], we intended to develop an efficient algorithm (MSBFGS) which would incorporate the approximation method of computing the gradients in [3] and their difference in [6] such that it can solve the system of nonlinear Equations (1) more efficiently. Secondly, since MSBFGS is involved with computation and the storage of matrices, it is not applicable to solve a large-scale system of nonlinear equations. Therefore, by giving an inverse formula of the update matrix in MSBFGS, we are going to develop another method (MSBFGS2) such that it can solve large-scale systems of nonlinear Equations (1). Additionally, in addition to the establishment of the two algorithms' convergence, we shall also demonstrate their powerful numerical performance as they are applied to solve benchmark test problems in the literature.

The rest of this paper is organized as follows. In Section 2, we first state the idea to propose two methods for solving the nonlinear symmetric equations. Then, two new algorithms are developed. Global convergence of algorithms is established in Section 3. Section 4 is devoted to numerical tests. Some conclusions are drawn in Section 5.

Some words about our notation: throughout the paper, the space $\mathbb{R}^n$ is equipped with the Euclidean norm $\|\cdot\|$, the transpose of any matrix is denoted by $\cdot^T$, and the $F(x_k)$ and $J(x_k)$ are abbreviated as F_k and J_k, respectively.

2. Development of Algorithm

In this section, we first simply recall a single-parameter scaling BFGS method [21] for solving the following unconstrained optimization problem:

$$\min f(x), x \in \mathbb{R}^n, \tag{5}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable such that its gradient is available. This method generates a sequence x_k satisfying:

$$x_{k+1} = x_k + \alpha_k d_k, \tag{6}$$

where $k \geq 0$, x_0 is the initial point, $\alpha_k > 0$ is called a step length obtained by some line search rule, and d_k is a search direction defined as

$$\begin{cases} d_k = -B_k^{-1} \nabla f_k, \\ B_{k+1} = B_k - \dfrac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \gamma_k \dfrac{y_k y_k^T}{y_k^T s_k}. \end{cases} \tag{7}$$

where $s_k = x_{k+1} - x_k$, $y_k = \nabla f_{k+1} - \nabla f_k$.

By minimizing the measure function introduced by Byrd and Nocedal [23]:

$$\psi(B_k) = \mathrm{tr}(B_k) - \ln(\det(B_k)). \tag{8}$$

Lv et al. [21] obtained that:

$$\gamma_k = \frac{y_k^T s_k}{\|y_k\|^2}. \tag{9}$$

In 2006, ref. [22] proposed a modification of the BFGS algorithm for unconstrained nonconvex optimization. The matrix B_k in [22] was updated by the formula:

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_{ks} y_{ks}^T}{y_{ks}^T s_k}. \tag{10}$$

where y_{ks} is the sum of y_k and $t_k \|\nabla f_k\|^r s_k$, and $t_k > 0, r > 0$.

Due to their impressive numerical efficiency, we now attempt to modify the aforementioned methods to solve the symmetric system of nonlinear Equation (1).

If we define f in (5) as

$$f(x) = \frac{1}{2}\|F(x)\|^2, x \in \mathbb{R}^n. \tag{11}$$

Then, for this objective function, any global minimizer of Problem (5) at which f vanishes is a solution of Problem (1). If an algorithm stops at a global minimizer x_k, i.e., $f(x_k) = 0$, then the algorithm finds a solution of (1).

By a symmetry of J, it holds that

$$\nabla f(x) = J(x)^T F(x) = J(x) F(x). \tag{12}$$

In [3], Li and Fukushima suggested that $\nabla f(x)$ is approximately computed by

$$g(x, \alpha) = \frac{F(x + \alpha F(x)) - F(x)}{\alpha}, \tag{13}$$

where $\alpha > 0$, and it can be proved that:

$$\lim_{\|\alpha F(x)\| \to 0} g(x, \alpha) = J(x)^T F(x) = \nabla f(x). \tag{14}$$

In other words, when $\|\alpha F(x)\|$ is sufficiently small, it is true that the vector $g(x, \alpha)$ defined by (13) is a nice approximation to $\nabla f(x)$.

In the actual calculation, [3] computed g_k by

$$g_k = g(x_k, \alpha_{k-1}). \tag{15}$$

where α_{k-1} is the step size at the last iterate point x_{k-1}. In general, the convergence of algorithms can ensure that $\lim_{k\to 0}\inf \|F_k\| = 0$.

Based on the work done by Li and Fukushima [3], Zhou [6] proposed a modified BFGS method to solve (1). The modified BFGS update formula is given by

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{\delta_k \delta_k^T}{\delta_k^T s_k}, \tag{16}$$

where:

$$\delta_k = \bar{\delta_k} + \max\left\{0, -\frac{\bar{\delta_k}^T s_k}{s_k^T s_k}\right\} s_k + \mu \|F_k\| s_k, \mu > 0, \tag{17}$$

and:

$$\bar{\delta_k} = g(x_{k+1}, \alpha_{k-1}) - g(x_k, \alpha_{k-1}). \tag{18}$$

Based on the ideas of [6,21,22], we now attempt to propose a modified single-parameter scaling BFGS method to solve (1). The modified BFGS update formula is given by

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \gamma_k \frac{\delta_k \delta_k^T}{\delta_k^T s_k}, \tag{19}$$

where $\gamma_k > 0$ and:

$$\delta_k = \begin{cases} \bar{\delta_k} + t\|F_k\|^r s_k, , & \text{if } s_k^T \bar{\delta_k} > 0, \\ \bar{\delta_k} - \dfrac{\bar{\delta_k}^T s_k}{s_k^T s_k} s_k + t\|F_k\|^r s_k, & \text{otherwise.} \end{cases} \tag{20}$$

It is clear that γ_k also minimizes (8) where B_k is defined by (19) if we compute γ_k by

$$\gamma_k = \frac{\delta_k^T s_k}{\|\delta_k\|^2}. \tag{21}$$

Moreover, we obtain an approximate quasi-Newton direction:

$$d_k = -B_k^{-1} g_k, \tag{22}$$

where g_k is an approximate gradient defined by (15).

Remark 1. *δ_k in (20) is slightly different from that in (17) where $r = 1$. Since γ_k in (21) can minimize (8) where B_k is defined by (19), its condition number which is the quotient of the maximum eigenvalue and the minimum eigenvalue of B_k is also minimized. Clearly, a smaller condition number of search direction matrices can theoretically ensure the stability of algorithms [21]. Numerical experiments will also show that B_k in (19) with γ_k being defined by (21) is more efficient and robust than that in (16).*

Since nonmonotone line search rules can play a critical role in solving a complicated nonconvex optimization problem, we use the nonmonotone line search in [3,6] to determine a step-size α_k along the direction d_k. Specifically, let $\sigma_1, \sigma_2, \rho, \rho_1, \in (0,1)$ and $\eta > 0$ be five given constants, and let η_k be a given positive sequence such that:

$$\sum_{k=0}^{+\infty} \eta_k \leq \eta < +\infty. \tag{23}$$

We search for a step size α_k satisfying:

$$\alpha_k = \begin{cases} 1, \text{ if } \|F(x_k + d_k)\| \le \rho_1 \|F_k\|, \\ \max\{\rho^i \mid \|F(x_k + \rho^i d_k)\|^2 \le (1+\eta_k)\|F_k\|^2 - \sigma_1 \|\rho^i F_k\|^2 \\ -\sigma_2 \|\rho^i d_k\|^2, i = 1, 2, \ldots, \}, \text{otherwise}. \end{cases} \tag{24}$$

With the above preparation, we are in a position to develop an algorithm to solve Problem (1). We now present its computer procedure as follows.

Remark 2. *In fact, for all $k > 0$, if B_0 is symmetric and positive definite, B_k in (19) is also symmetric and positive definite since*

$$\delta_k^T s_k \ge t \|F_k\|^r \|s_k\|^2 > 0. \tag{25}$$

Therefore, the algorithm is well defined. From the definition of δ_k in (20), we can also obtain:

$$\|\delta_k\|^2 \ge t^2 \|F_k\|^{2r} \|s_k\|^2 > 0. \tag{26}$$

Since Algorithm 1 cannot efficiently solve large-scale nonlinear symmetric equations, based on the work done by [3], we will develop another algorithm that is not involved with matrix operation and inverse operation. When we set $B_k = I$, the inverse matrix of B_{k+1} in (19) can be written as

$$H_{k+1} = I - \frac{\delta_k s_k^T + s_k \delta_k^T}{\delta_k^T s_k} + \left(\frac{1}{\gamma_k} + \frac{\delta_k^T \delta_k}{\delta_k^T s_k} \right) \frac{s_k s_k^T}{\delta_k^T s_k}, \tag{27}$$

where γ_k is the same as (21), and:

$$\begin{cases} \delta_k = F(x_k + \xi_k) - F_k, \\ \xi_k = F_{k+1} - F_k, \end{cases} \tag{28}$$

In fact, δ_k and ξ_k are completely the same as those in [3,15].

Moreover, in order to guarantee that our proposed method generates descent directions and to further increase its computational efficiency and robustness, we can compute the direction by

$$d_k = \begin{cases} -F(x_0) \triangleq -g_0, \quad \text{if } k = 0, \\ -g_k, \quad \text{if } \delta_{k-1}^T s_{k-1} \le 0, \\ -g_k + \beta_k s_{k-1} + \theta_k \delta_{k-1}, \quad \text{otherwise}, \end{cases} \tag{29}$$

where g_k is defined in (15) and:

$$\begin{cases} \beta_k = \dfrac{\delta_{k-1}^T g_k}{\delta_{k-1}^T s_{k-1}} - 2 \dfrac{\|\delta_{k-1}\|^2}{\delta_{k-1}^T s_{k-1}} \dfrac{s_{k-1}^T g_k}{\delta_{k-1}^T s_{k-1}}, \\ \theta_k = \dfrac{s_{k-1}^T g_k}{\delta_{k-1}^T s_{k-1}}. \end{cases} \tag{30}$$

Remark 3. *Note that the nonmonotone line search (31) is a variant of (24) with $\sigma_1 = 0$.*

Remark 4. *Since the search direction of Algorithm 1 at each iteration is an approximate quasi-Newton direction, which is involved with the solution of a linear system of equations, Algorithm 1 can only efficiently solve small–medium-scale Problem (1). Instead, the needed search directions in Algorithm 2 is only associated with evaluating the function F without requirement of computing or storing its Jacobian matrix. Thus, compared with Algorithm 1, Algorithm 2 is more applicable to solving large-scale systems of nonlinear equations. In addition, two different approximation methods are used to compute the difference of gradients (see (18) and (28)).*

Algorithm 1: (Modified Single-Parameter Scaling BFGS Algorithm (MSBFGS))

Step 0. Choose three constants $\sigma_1, \sigma_2, \rho, \rho_1, \varepsilon \in (0,1), r > 0, t > 0, \alpha_{-1} > 0$. Take a sequence η_k satisfying (23). Arbitrarily choose an initial iterate point $x_0 \in \mathbb{R}^n$, a symmetric and positive definite matrix $B_0 \in R^{n\times n}$. Set $k := 0$.
Step 1. If $\|F_k\| \leq \varepsilon$ is satisfied, then the algorithm stops.
Step 2. Compute d_k by (22) and (19).
Step 3. Determine a step length α_k satisfying (24).
Step 4. Set $x_{k+1} = x_k + \alpha_k d_k$.
Step 5. Set $k := k+1$, return to Step 1.

Algorithm 2: (Modified Single-Parameter Scaling BFGS Algorithm 2(MS-BFGS2))

Step 0. Choose three constants $\sigma, \rho, \varepsilon \in (0,1)$. Take a sequence η_k satisfying (23). Arbitrarily choose an initial iterate point $x_0 \in \mathbb{R}^n$. Set $k = 0$.
Step 1. If $\|F_k\| \leq \varepsilon$ is satisfied, then the algorithm stops.
Step 2. Compute d_k by (29).
Step 3. Determine a step length $\alpha_k = \max\{\rho^i \mid i = 0,1,2,\ldots,\}$ satisfying:

$$f(x_k + \alpha_k d_k) - f(x_k) \leq -\sigma\|\alpha_k d_k\|^2 + \eta_k f(x_k). \tag{31}$$

Step 4. Set $x_{k+1} = x_k + \alpha_k d_k$.
Step 5. Set $k := k+1$, return to Step 1.

Remark 5. *By combining the advantages of the three-term conjugate gradient method in [21] and those of the approximation methods for computing the difference of gradients, it is believable that the numerical performance of Algorithm 2 is better than the algorithm in [21]. In the two subsequent sections, apart from establishing the convergence theory of Algorithm 2, we will also test its efficiency in solving large-scale problems.*

Remark 6. *Very recently, Liu et al. [15] developed an algorithm to solve (1), where β_k and θ_k in (29) were replaced by*

$$\begin{cases} \beta_k = \dfrac{g_k^T \delta_{k-1}}{s_{k-1}^T \delta_{k-1}} - \dfrac{\|\delta_{k-1}\|^2 g_k^T s_{k-1}}{(s_{k-1}^T \delta_{k-1})^2}, \\ \theta_k = \dfrac{g_k^T s_{k-1}}{s_{k-1}^T \delta_{k-1}}, \end{cases} \tag{32}$$

respectively. Since (30) and (32) are two similar choices, it is interesting to compare their numerical performance for solving the problem (1).

3. Convergence of Algorithm

In this section, we establish the global convergence of Algorithms 1 and 2. For this purpose, we first define the level set:

$$\Omega = \left\{ x \mid f(x) = \frac{1}{2}\|F(x)\|^2 \leq \mathrm{e}^{\eta} f(x_0) \right\}. \tag{33}$$

Clearly, it follows from Step 3 of Algorithm 1 that:

$$\begin{aligned} f(x_{k+1}) = \tfrac{1}{2}\|F_{k+1}\|^2 &\leq \tfrac{1}{2}(1+\eta_k)\|F_k\|^2 \leq \tfrac{1}{2}\mathrm{e}^{\eta_k}\|F_k\|^2 \\ &\leq \tfrac{1}{2}\mathrm{e}^{\sum_{i=0}^{k}\eta_i}\|F_0\|^2 \leq \tfrac{1}{2}\mathrm{e}^{\eta}\|F_0\|^2 = \mathrm{e}^{\eta} f(x_0). \end{aligned}$$

Thus, any sequence $\{x_k\}$ generated by Algorithm 1 belongs to Ω, i.e., $x_k \in \Omega$ for all k. In other words, there exists a constant $\Delta > 0$, such that:

$$\|F_k\| \leq \Delta. \tag{34}$$

Moreover, since η_k satisfies (23), from Lemma 3.3 in [24], we know that the sequence $\{\|F_k\|\}$ generated by Algorithm 1 converges.

Likewise, from the line search rule of Algorithm 2, we know that the sequence of iterate points $\{x_k\}$ generated by Algorithm 2 also belongs to Ω and $\{\|F_k\|\}$ generated by Algorithm 2 also satisfies (34).

As done in the existing results [13,15,25], we also suppose that F in (1) satisfies the following conditions:

Assumption 1. *The solution set of the problem (1) is nonempty.*

Assumption 2. *The level set Ω is bounded.*

Assumption 3. *F is a continuous differentiable on an open and convex set $V \subseteq \mathbb{R}^n$ containing the level set Ω, and its Jacobian matrix is symmetric and bounded on V, i.e., there exists a positive constant M such that:*

$$\|J(x)\| \leq M, \ \forall x \in V. \tag{35}$$

Assumption 4. *$J(x)$ is uniformly nonsingular on V, i.e., there exists a positive constant m such that:*

$$m\|p\| \leq \|J(x)p\|, \ \forall\, x \in V, p \in \mathbb{R}^n. \tag{36}$$

Clearly, Assumptions 2–4 imply that there exist positive constants $M \geq m > 0$ such that the following statements are true:

(1) For any $x \in V$, $p \in \mathbb{R}^n$,

$$m\|p\| \leq \|J(x)p\| \leq M\|p\|. \tag{37}$$

(2) For any $x, y \in V$,

$$\begin{aligned} &m\|x-y\| \leq \|F(x) - F(y)\| \\ &= \|J(\theta x + (1-\theta)y)(x-y)\| \leq M\|x-y\|, \end{aligned} \tag{38}$$

where $\theta \in (0,1)$.

(3) For any sequence $\{x_k\} \subset V$,

$$m\|F_k\| \leq \|g_k\| = \|J(x_k + \alpha_{k-1} t F_k)F_k\| \leq M\|F_k\|, \tag{39}$$

where $t \in (0,1)$.

Under Assumptions 2–4, we can prove that Algorithm 1 has the following nice properties.

Lemma 1. *Let $\{B_k\}$ be generated by the BFGS formula (19), where B_0 is a symmetric and positive definite and γ_k is defined by (21). If there exists a positive constant $m_0 > 0$, such that:*

$$\|F_k\| \geq m_0, \quad \forall k \geq 0, \tag{40}$$

then for any $p \in (0,1)$ and $k > 1$, there exist positive constants $\beta_i, i = 1,2,3,4$ such that:

$$\begin{cases} \beta_1\|s_j\| \leq \|B_j s_j\| \leq \beta_2\|s_j\|, \\ \beta_3\|s_j\|^2 \leq s_j^T B_j s_j \leq \beta_4\|s_j\|^2 \end{cases} \tag{41}$$

hold for at least $\lceil pk \rceil$ values of $j \in [1,k]$, where $\lceil t \rceil$ is the smallest integer which is larger than or equal to t.

Proof. From (8) and (19), we have:

$$\begin{aligned}
\psi(B_{k+1}) &= \mathrm{tr}(B_{k+1}) - \ln(\det(B_{k+1})) \\
&= \mathrm{tr}(B_k) - \frac{\|B_k s_k\|^2}{s_k^T B_k s_k} + \gamma_k \frac{\|\delta_k\|^2}{\delta_k^T s_k} - \ln\left(\gamma_k \frac{\delta_k^T s_k}{s_k^T B_k s_k} \det(B_k)\right) \\
&= \psi(B_k) - \frac{\|B_k s_k\|^2}{s_k^T B_k s_k} + \gamma_k \frac{\|\delta_k\|^2}{\delta_k^T s_k} - \ln\left(\gamma_k \frac{\delta_k^T s_k}{s_k^T B_k s_k}\right) \\
&= \psi(B_k) - \left[\frac{\|B_k s_k\| \|s_k\|}{s_k^T B_k s_k}\right]^2 \frac{s_k^T B_k s_k}{s_k^T s_k} + \gamma_k \frac{\|\delta_k\|^2}{\delta_k^T s_k} - \ln\left(\gamma_k \frac{\delta_k^T s_k}{s_k^T s_k} \frac{s_k^T s_k}{s_k^T B_k s_k}\right).
\end{aligned} \tag{42}$$

Take $\cos\theta_k = \dfrac{s_k^T B_k s_k}{\|B_k s_k\| \|s_k\|}$ and $q_k = \dfrac{s_k^T B_k s_k}{s_k^T s_k}$, then (42) can be rewritten as

$$\begin{aligned}
\psi(B_{k+1}) &= \psi(B_k) + \gamma_k \frac{\|\delta_k\|^2}{\delta_k^T s_k} - \ln\left(\gamma_k \frac{\delta_k^T s_k}{s_k^T s_k}\right) - \frac{q_k}{\cos^2\theta_k} + \ln q_k \\
&= \psi(B_k) + \gamma_k \frac{\|\delta_k\|^2}{\delta_k^T s_k} - \ln\left(\gamma_k \frac{\delta_k^T s_k}{s_k^T s_k}\right) - 1 + \ln\cos^2\theta_k \\
&\quad + \left[1 - \frac{q_k}{\cos^2\theta_k} + \ln\frac{q_k}{\cos^2\theta_k}\right].
\end{aligned} \tag{43}$$

On the other hand, from (2.11) in [6], we know:

$$\|\delta_k\| \le C_1 \|s_k\|, \tag{44}$$

where $C_1 > 0$ is a constant. Hence, it follows from (25), (26), (34) , (40) and (44) that:

$$\frac{\delta_k^T s_k}{s_k^T s_k} \ge t\|F_k\|^r \ge t m_0^r \triangleq m_1 > 0, \tag{45}$$

and:

$$m_2 \triangleq \frac{t^2 m_0^{2r}}{C_1} \le \frac{1}{\gamma_k} = \frac{\|\delta_k\|^2}{\delta_k^T s_k} \le \frac{C_1^2}{t m_0^r} \triangleq M_2. \tag{46}$$

From (43), (45) and (46), we have:

$$\begin{aligned}
&\psi(B_{k+1}) \le \psi(B_1) + \left(\frac{M_2}{m_2} - 1 - \ln\frac{m_1}{M_2}\right)k \\
&+ \sum_{j=1}^{k} \left(\ln\cos^2\theta_j + 1 - \frac{q_j}{\cos^2\theta_j} + \ln\frac{q_j}{\cos^2\theta_j}\right).
\end{aligned} \tag{47}$$

Take $\eta_j \ge 0$:

$$\eta_j = -\ln\cos^2\theta_j - \left(1 - \frac{q_j}{\cos^2\theta_j} + \ln\frac{q_j}{\cos^2\theta_j}\right). \tag{48}$$

It is clear that $\psi(B_{k+1}) > 0$ since B_{k+1} is symmetric and positive definite. Hence, from (47), we have:

$$\frac{1}{k}\sum_{j=1}^{k} \eta_j < \frac{\psi(B_1)}{k} + \left(\frac{M_2}{m_2} - 1 - \ln\frac{m_1}{M_2}\right). \tag{49}$$

Let us define J_k to be a set consisting of the $\lceil pk \rceil$ indices corresponding to the $\lceil pk \rceil$ smallest values of η_j, for $j \le k$, and let η_{m_k} denote the largest of the η_j for $j \in J_k$. Then:

$$\frac{1}{k}\sum_{j=1}^{k}\eta_j \ge \frac{1}{k}\Big[\eta_{m_k} + \sum_{j=1, j\notin J_k}^{k}\eta_j\Big] = \frac{1}{k}(\eta_{m_k} + (k - \lceil pk \rceil)\eta_{m_k}) \ge \eta_{m_k}(1-p). \tag{50}$$

Thus, from (48)–(50) and the following fact:

$$1 - \frac{q_j}{\cos^2\theta_j} + \ln\frac{q_j}{\cos^2\theta_j} \le 0,$$

we have:

$$-\ln\cos^2\theta_j \le \eta_j \le \tfrac{1}{1-p}\left(\psi(B_1) + \left(\frac{M_2}{m_2} - 1 - \ln\frac{m_1}{M_2}\right)\right) \triangleq \beta_0, \forall j \in J_k. \tag{51}$$

It follows from (51) that:

$$\cos\theta_j \ge \mathrm{e}^{-\beta_0/2}, \forall j \in J_k. \tag{52}$$

On the other hand, since $\ln\cos^2\theta_j \le 0$, we have:

$$1 - \frac{q_j}{\cos^2\theta_j} + \ln\frac{q_j}{\cos^2\theta_j} \ge -\beta_0, \forall j \in J_k. \tag{53}$$

Let $\mu(t) = 1 - t + \ln t$, then by simple analysis, we have:

$$\begin{cases} \mu(t) \le 0, \\ \arg\max\limits_{t>0}\mu(t) = 1, \\ \lim\limits_{t\to 0}\mu(t) = -\infty, \\ \lim\limits_{t\to\infty}\mu(t) = -\infty. \end{cases} \tag{54}$$

Therefore, there exist positive constants $\bar{\beta_3}$ and $\bar{\beta_4}$ such that the following inequalities hold:

$$\bar{\beta_3} \le \frac{q_j}{\cos^2\theta_j} \le \bar{\beta_4}, \forall j \in J_k.$$

Together with (52), we obtain:

$$\beta_3 \triangleq \bar{\beta_3}\mathrm{e}^{-\beta_0} \le q_j = \frac{s_j^T B_j s_j}{s_j^T s_j} = \frac{q_j}{\cos^2\theta_j}\cos^2\theta_j \le \bar{\beta_4} \triangleq \beta_4, \forall j \in J_k. \tag{55}$$

Moreover:

$$\beta_3 \le \frac{\|B_j s_j\|}{\|s_j\|} = \frac{q_j}{\cos\theta_j} \le \frac{\beta_4}{\mathrm{e}^{-\beta_0/2}} \triangleq \beta_2, \forall j \in J_k. \tag{56}$$

Take $\beta_1 = \beta_3$, we obtain the desired result. □

Remark 7. *From the proof of Lemma 1, we know that if γ_k is not defined by (21), Lemma 1 is also true whenever there exist constants $m_3 > 0$ and $M_3 > 0$ such that $m_3 \le \gamma_k \le M_3$ holds.*

By Lemma 1, since the definition of $\bar{\delta_k}$ and the line search rule are completely the same as those in [6], we can obtain the same convergence result as Algorithm 1 without proof.

Theorem 1. *Suppose that Assumptions 1–4 hold. Let $\{x_k\}$ be a sequence generated by Algorithm 1. Then:*

$$\lim_{k\to\infty}\inf\|F_k\| = 0. \tag{57}$$

To establish the global convergence of Algorithm 2, we first prove the following results.

Lemma 2. *Let $\{x_k\}$ be a sequence generated by Algorithm 2. If Assumptions 1–4 hold. Then:*

$$\lim_{k\to\infty} \alpha_k \|d_k\| = 0. \tag{58}$$

Proof. Similar to the proof of Lemma 3.1 in [15], we can prove (58). □

Lemma 2 shows that $\lim_{k\to\infty} s_k = 0$ holds.

Lemma 3. *Let $\{x_k\}$ be a sequence generated by Algorithm 2. If Assumptions 1–4 hold, then:*

$$\|\delta_k\| \leq M^2 \|s_k\|. \tag{59}$$

Additionally, if $s_k \to 0$, then there exists a constant $\bar{m} > 0$ such that for all sufficiently large k:

$$s_k^T \delta_k \geq \bar{m} \|s_k\|^2. \tag{60}$$

Proof. On the one hand, it follows from (28) and (37) that:

$$\begin{aligned} \|\delta_k\| &= \|F(x_k + \xi_k) - F(x_k)\| \\ &= \|J(\theta_1 x_k + (1-\theta_1)(x_k + \xi_k))\| \|\xi_k\| \\ &\leq M\|\xi_k\| \\ &= M\|F(x_{k+1}) - F(x_k)\| \\ &\leq M\|J(\theta_2 x_k + (1-\theta_2)x_{k+1})\| \|x_{k+1} - x_k\| \\ &\leq M^2 \|s_k\|, \end{aligned} \tag{61}$$

where $\theta_1, \theta_2 \in (0,1)$. On the other hand, by the mean-value theorem, we have:

$$\begin{aligned} & s_k^T \delta_k \\ = \ & s_k^T (F(x_k + \xi_k) - F(x_k)) \\ = \ & s_k^T \int_0^1 J(x_k + t\xi_k)\, dt \xi_k \\ = \ & s_k^T \int_0^1 J(x_k + t\xi_k)\, dt \int_0^1 J(x_k + ts_k)\, dt s_k \\ = \ & s_k^T \left(\int_0^1 J(x_k + ts_k)\, dt \right)^2 s_k + s_k^T \int_0^1 (J(x_k + t\xi_k) - J(x_k + ts_k))\, dt \cdot \int_0^1 J(x_k + ts_k)\, dt s_k \\ = \ & \| \int_0^1 J(x_k + ts_k)\, dt s_k \|^2 + s_k^T \int_0^1 (J(x_k + t\xi_k) - J(x_k + ts_k))\, dt \cdot \int_0^1 J(x_k + ts_k)\, dt s_k \\ = \ & \| \int_0^1 J(x_k + ts_k)\, dt s_k \|^2 - s_k^T \int_0^1 (J(x_k + ts_k) - J(x_k + t\xi_k))\, dt \cdot \int_0^1 J(x_k + ts_k)\, dt s_k \\ \geq \ & \| \int_0^1 J(x_k + ts_k)\, dt s_k \|^2 - \| \int_0^1 (J(x_k + ts_k) - J(x_k + t\xi_k))\, dt s_k \| \cdot \| \int_0^1 J(x_k + ts_k)\, dt s_k \| \\ = \ & \|F(x_{k+1}) - F(x_k)\|^2 - \| \int_0^1 (J(x_k + ts_k) - J(x_k + t\xi_k))\, dt s_k \| \cdot \| \int_0^1 J(x_k + ts_k)\, dt s_k \| \\ \geq \ & m^2 \|s_k\|^2 - M\|s_k\|^2 \cdot \int_0^1 \|J(x_k + ts_k) - J(x_k + t\xi_k)\|\, dt \\ = \ & \left(m^2 - M \int_0^1 \|J(x_k + ts_k) - J(x_k + t\xi_k)\|\, dt \right) \cdot \|s_k\|^2. \end{aligned} \tag{62}$$

From Lemma 2, we have $s_k \to 0$, hence $\xi_k = F(x_{k+1}) - F(x_k) \to 0$. By continuity of J, we get (60). □

Lemma 4. *Suppose that Assumptions 1–4 hold. If there exists a constant $r > 0$ such that for all $k \in \mathbb{N}$:*

$$\|F_k\| \geq r, \tag{63}$$

Then, there exists a constant $\hat{M} \geq \hat{m} > 0$ such that:

$$\hat{m} \leq \|d_k\| \leq \hat{M}, \tag{64}$$

hold, where $\hat{m} = rm$.

Proof. Similar to Proposition 3 in [21], we have:

$$g_k^T d_k \leq -\frac{1}{2}\|g_k\|^2. \tag{65}$$

From (39), (63) and (65), it follows that:

$$\|d_k\| \geq \frac{1}{2}\|g_k\| \geq \frac{1}{2}m\|F_k\| \geq \frac{1}{2}mr. \tag{66}$$

Therefore, the left-hand side of (64) holds.

From (60), the definition of β_k in (29) and (30), we have:

$$\begin{aligned} |\beta_k| &= \left| \frac{\delta_{k-1}^T g_k}{\delta_{k-1}^T s_{k-1}} - 2\frac{\|\delta_{k-1}\|^2}{\delta_{k-1}^T s_{k-1}} \frac{s_{k-1}^T g_k}{\delta_{k-1}^T s_{k-1}} \right| \\ &\leq \frac{\|\delta_{k-1}\|\|g_k\|}{\left|\delta_{k-1}^T s_{k-1}\right|} + 2\frac{\|\delta_{k-1}\|^2\|s_{k-1}\|\|g_k\|}{(s_{k-1}^T \delta_{k-1})^2} \\ &\leq \frac{M^2\|s_{k-1}\|\|g_k\|}{\bar{m}\|s_{k-1}\|^2} + 2M^4\frac{\|s_{k-1}\|^3\|g_k\|}{\bar{m}^2\|s_{k-1}\|^4} \\ &\leq \frac{M^2\|g_k\|}{\bar{m}\|s_{k-1}\|} + 2M^4\frac{\|g_k\|}{\bar{m}^2\|s_{k-1}\|}. \end{aligned} \tag{67}$$

On the other hand:

$$\begin{aligned} |\theta_k| &= \frac{\left|s_{k-1}^T g_k\right|}{\left|\delta_{k-1}^T s_{k-1}\right|} \\ &\leq \frac{\|s_{k-1}\|\|g_k\|}{\bar{m}\|s_{k-1}\|^2} \\ &\leq \frac{\|g_k\|}{\bar{m}\|s_{k-1}\|}. \end{aligned} \tag{68}$$

From Assumptions 2 and 3, (34) and (39), we know that the sequence $\{g_k\}$ is bounded, i.e., there exists a positive constant $\hat{\gamma}$ such that for all $k \geq 0$:

$$\|g_k\| \leq \hat{\gamma}. \tag{69}$$

Thus, from (29), (30), (67), (68), (69) and the line search rule, it is easy to obtain that:

$$\begin{aligned} \|d_k\| &= \|-g_k + \beta_k s_{k-1} + \theta_k \delta_{k-1}\| \\ &\leq \|g_k\| + |\beta_k|\|s_{k-1}\| + |\theta_k|\|\delta_{k-1}\| \\ &\leq \hat{\gamma}\left(1 + \frac{M^2}{\bar{m}} + \frac{2M^4}{\bar{m}^2} + \frac{1}{\bar{m}}\right). \end{aligned} \tag{70}$$

The proof is completed. □

Lemma 5. *Suppose that Assumptions 1–4 hold. Then:*

$$\alpha_k \geq \min\left\{1, \frac{\rho(\|g_k\|^2 - 2t_k\|F_k\|\|d_k\|)}{(M^2 + 2\sigma)\|d_k\|^2}\right\}, \tag{71}$$

where:

$$t_k = \int_0^1 \|J(x_k + t\rho^{-1}\alpha_k d_k) - J(x_k + t\alpha_{k-1}F_k)\|\, dt. \tag{72}$$

Proof. If $\alpha_k = 1$, it is easy to see that (71) holds. If $\alpha_k \neq 1$, then $\alpha_k^{'} = \rho^{-1}\alpha_k$ does not satisfy (31), that is to say, $\alpha_k^{'}$ satisfies:

$$f(x_k + \alpha_k' d_k) - f(x_k) > -\sigma \|\alpha_k' d_k\|^2. \tag{73}$$

On the other hand, from (38), it follows that:

$$\begin{aligned} & f(x_k + \alpha_k' d_k) - f(x_k) \\ & = \tfrac{1}{2}\|F(x_k + \alpha_k' d_k)\|^2 - \tfrac{1}{2}\|F(x_k)\|^2 \\ & = F(x_k)^T \Big(F(x_k + \alpha_k' d_k) - F(x_k) \Big) + \tfrac{1}{2}\|F(x_k + \alpha_k' d_k) - F(x_k)\|^2 \\ & \le F(x_k)^T \int_0^1 J(x_k + t\alpha_k' d_k)\alpha_k' d_k \, dt + \tfrac{1}{2} M^2 \|\alpha_k' d_k\|^2. \end{aligned} \tag{74}$$

Combined with (73), we obtain:

$$\begin{aligned} \alpha_k' \ & \ge \frac{-2F_k^T \int_0^1 J(x_k + t\alpha_k' d_k) d_k \, dt}{(M^2 + 2\sigma)\|d_k\|^2} \\ & = \big(-2F_k^T \int_0^1 J(x_k + t\alpha_k' d_k) d_k \, dt + 2g_k^T d_k - 2g_k^T d_k \big) / (M^2 + 2\sigma)\|d_k\|^2 \\ & \ge \big(\|g_k\|^2 - 2F_k^T \int_0^1 J(x_k + t\alpha_k' d_k) d_k \, dt + 2g_k^T d_k \big) / (M^2 + 2\sigma)\|d_k\|^2 \\ & = \Big[\|g_k\|^2 - 2F_k^T \int_0^1 J(x_k + t\alpha_k' d_k) d_k \, dt + 2\big(\int_0^1 J(x_k + t\alpha_{k-1} F_k) F_k \, dt \big)^T d_k \Big] \\ & \ \ / (M^2 + 2\sigma)\|d_k\|^2 \\ & \ge \frac{\|g_k\|^2 - 2t_k \|F_k\| \|d_k\|}{(M^2 + 2\sigma)\|d_k\|^2}, \end{aligned} \tag{75}$$

where the first inequality follows from (65), the second equality follows from (13) and the differentiability of F. The third inequality follows from the Cauchy–Schwartz inequality.

By (75), we obtain the desired result. □

Lemma 6. *Suppose that Assumptions 1–4 hold. Let $\{x_k\}$ and $\{d_k\}$ be two sequences generated by Algorithm 2. Then, the line search rule (31) by Step 3 in Algorithm 2 is well defined.*

Proof. Our aim is to show that the line search rule (31) terminates finitely with a positive step length α_k. In contrast, suppose that for some iterate indexes such as k^*, the condition (31) does not hold. As a result, for all $m \in \mathbb{N}$:

$$f(x_{k^*} + \rho^m d_{k^*}) - f(x_{k^*}) > -\sigma \|\rho^m d_{k^*}\|^2 + \eta_{k^*} f(x_{k^*}). \tag{76}$$

which can be written as

$$\frac{f(x_{k^*} + \rho^m d_{k^*}) - f(x_{k^*})}{\rho^m} > -\sigma \rho^m \|d_{k^*}\|^2 + \frac{\eta_{k^*} f(x_{k^*})}{\rho^m}. \tag{77}$$

By taking the limit as $m \to \infty$ in both sides of (77), we have:

$$\nabla f(x_{k^*})^T d_{k^*} \ge +\infty. \tag{78}$$

However, from Assumption 3, Lemma 4 and (34) and the stop rule of Algorithm 2, we obtain:

$$\|\nabla f(x_{k^*}) d_{k^*}\| = \|J(x_{k^*}) F(x_{k^*}) d_{k^*}\| \le M\hat{M}\Delta < +\infty. \tag{79}$$

Clearly, (79) contradicts (78). That is to say, the line search rule terminates within a finite number of many trials to obtain a positive step length α_k, i.e., Step 3 of Algorithm 2 is well defined. □

With the above preparation, we now state the convergence result of Algorithm 2.

Theorem 2. *Suppose that Assumptions 1–4 hold. Let $\{x_k\}$ be a sequence generated by Algorithm 2. Then:*

$$\lim_{k \to \infty} \inf \|F_k\| = 0. \tag{80}$$

Proof. For the sake of contradiction, we suppose that the conclusion is not true. Then, there exists a constant $\varepsilon_0 > 0$ such that $\|F_k\| \geq \varepsilon_0$ for all $k \in \mathbb{N}$. Hence, (66) holds. Hence, from (58), we have:

$$\lim_{k\to\infty} \alpha_k = 0. \tag{81}$$

It follows from (81) and (72) that:

$$\lim_{k\to\infty} t_k = 0. \tag{82}$$

From (39), Lemmas 4 and 5, we know the following inequality:

$$\alpha_k \geq \min\left\{1, \frac{\rho\left(m^2\varepsilon_0^2 - 2t_k\|F_k\|\|d_k\|\right)}{(M^2+2\sigma)\hat{M}^2}\right\} \tag{83}$$

holds for all sufficiently large k. Therefore, taking the limit as $k \to \infty$ in both sides of (83), it holds that:

$$\lim_{k\to\infty} \alpha_k \geq \min\left\{1, \frac{\rho m^2\varepsilon_0^2}{(M^2+2\sigma)\hat{M}^2}\right\} > 0, \tag{84}$$

which is a contradiction. Thus, the proof of Theorem 2 has been completed. □

4. Numerical Tests

In this section, by numerical tests, we study the effectiveness and robustness of Algorithm 1 when it is used to solve nonlinear systems of symmetric equations.

We first list the benchmark test problems $F(x) = (F_1(x), F_2(x), \ldots, F_n(x))^T = 0$, which includes all the four test problems in [6].

Problem 1. *Strictly convex function 1 ([26], p. 29) Let $F(x)$ be the gradient of $h(x) = \sum_{i=1}^{n}(e^{x_i} - x_i)$, meaning that:*

$$F_i(x) = e^{x_i} - 1,\ i = 1, 2, \ldots, n.$$

Problem 2. *In Reference [22], the elements of $F(x)$ are given by*

$$F_i(x) = 2x_i - \sin x_i,\ i = 1, 2, \ldots, n-1.$$

Problem 3. *The discretized Chandrasekhar's H-Equation [27]:*

$$F_i(x) = x_i - \left(1 - \frac{c}{2n}\sum_{j=1}^{n}\frac{\mu_i x_j}{\mu_i + \mu_j}\right)^{-1},\ i = 1, 2, \ldots, n-1.$$

where $c = 0.9$ and $\mu_i = (i - 1/2)/n$.

Problem 4. *Unconstrained optimization problem:*

$$\min f(x), \quad x \in \mathbb{R}^n,$$

with Engval function [28] $f : \mathbb{R}^n \to \mathbb{R}$ defined by

$$f(x) = \sum_{i=2}^{n}\left((x_{i-1}^2 + x_i^2)^2 - 4x_{i-1} + 3\right).$$

The related symmetric nonlinear equation is:

$$F(x) = \frac{1}{4}\nabla f(x) = 0,$$

where $F(x) = (F_1(x), F_2(x), \ldots, F_n(x))^T$ *is defined by:*

$$\begin{aligned} &F_1(x) = x_1(x_1^2 + x_2^2) - 1, \\ &F_i(x) = x_i(x_{i-1}^2 + 2x_i^2 + x_{i+1}^2) - 1,\ i = 2, 3, \ldots, n-1, \\ &F_n(x) = x_n(x_{n-1}^2 + x_n^2). \end{aligned}$$

Problem 5. *The discretized two-point boundary value problem like the problem in [1]:*

$$F(x) = Ax + \frac{G(x)}{(n+1)^2},$$

$$A = \begin{pmatrix} 8 & -1 & & & \\ -1 & 8 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & -1 \\ & & & -1 & 8 \end{pmatrix}.$$

and $G(x) = (G_1(x), G_2(x), \ldots, G_n(x))^T$ *with* $G_i(x) = \sin x_i - 1$, $i = 1, 2, \ldots, n$.

Problem 6. *In Reference [6], the elements of* $F(x)$ *are given by*

$$\begin{aligned} &F_i(x) = 2x_i - x_{i+1} + \sin(x_i) - 1,\ i = 1, 2, \ldots, n-1, \\ &F_n(x) = 2x_n + \sin(x_n) - 1. \end{aligned}$$

Problem 7. *In Reference [6], the elements of* $F(x)$ *are given by*

$$\begin{aligned} &F_i(x) = x_i - 1,\ i = 1, 2, \ldots, n-2, \\ &F_{n-1}(x) = x_{n-1} \textstyle\sum_{i=1}^{n-2} i(x_i - 1), \\ &F_n(x) = \left(\textstyle\sum_{i=1}^{n-2} i(x_i - 1)\right)^2. \end{aligned}$$

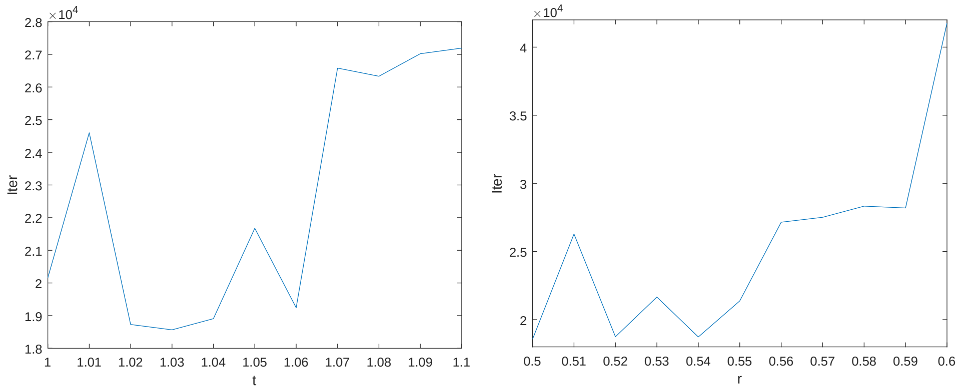
All the algorithms are coded in MATLAB R2021a and run on a desktop (at Peking University) computer with a 3.6 GHZ CPU processor, 16 GB memory and Windows 7 operation system. The relevant parameters are specified by

$$B_0 = I,\ \ \sigma_1 = \sigma_2 = 0.01,\ \ \rho = 0.5,\ \ \rho_1 = 0.95,$$
$$\alpha_{-1} = 0.01,$$

and μ in MBFGS method (Algorithm 2.1 in [6]) is the same as [6], i.e., $\mu = 10^{-4}$. In fact, the above parameters all are same as those in [6]. Similarly to [6], we use the matrix left division command $d = -B\backslash g$ to directly solve the linear subproblem (22). The termination condition of all the algorithms is: $\|F_k\| \le 10^{-6}$, or the number of iterations exceeds 10^4, or the MATLAB R2010b crashes, or the CPU time exceeds 100 s.

In order to choose optimal values for the parameters t and r in Algorithm 1, we first take $r = 0.5$, and choose t from the interval $[1, 1.1]$ with a step size of 0.01. We present the total number of iterations (Iter) in Figure 1a as Algorithm 1 is used to solve all the seven test problems with different sizes n (10, 50, 100 and 500) and different initial guesses. The initial guesses are $x_1 = (0.1, 0.1, \ldots, 0.1)^T$, $x_2 = (-0.1, -0.1, \ldots, -0.1)^T$, $x_3 = (1, 1, \ldots, 1)^T$, $x_4 = (-1, -1, \ldots, -1)^T$, $x_5 = (1/n, 1/n, \ldots, 1/n)^T$, $x_6 = (-1/n, -1/n, \ldots, -1/n)^T$. From Figure 1a, we know that Iter changes little when $t \in [1.02, 1.03]$ and it is the least when $t = 1.03$.

We then take $t = 1.03$, and choose r from the interval $[0.5, 0.6]$ with a step size of 0.01. We present the total number of iterations (Iter) in Figure 1b as Algorithm 1 is used to solve all seven test problems with different sizes n (10, 50, 100 and 500) and different initial guesses (x_1–x_6). Figure 1b shows that Iter changes little when $r \in [0.52, 0.54]$ and Algorithm 1 with $r = 0.5$ performs the best.



(**a**) t changes (**b**) r changes

Figure 1. The total number of iterations with different values of algorithmic parameters.

According to the above research, we take $t = 1.03$ and $r = 0.5$, and compare Algorithm 1 (MSBFGS) with two similar algorithms proposed very recently to see which is more efficient as they are used to solve all seven test problems with different sizes n and different initial guesses. One is the Gauss–Newton-based BFGS method (GNBFGS for short) in [3] and another is MBFGS in [6] since they have been reported to be more efficient than the state-of-the-art ones.

In Table A1, we report the numerical performance of the three algorithms. For the simplification of statement, we use the following notations in Table A1.

P: the problems;

Dim: the dimension of test problems;

CPU: the CPU time in seconds;

Ni: the number of iterations;

Nf: the number of function evaluations;

Norm (F): the norm of F_k at the stopping point;

F: a notation when an algorithm fails to achieve the given iteration tolerance, or in the limited number of iterations exceeds 10^4, or the MATLAB R2010b crashes, or in the limited the CPU time exceeds 100 s.

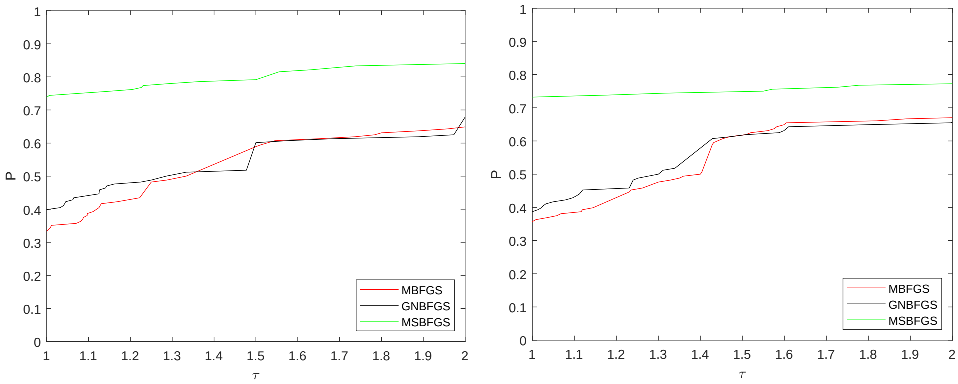
The underlined data in Table A1 indicate the superiority of Algorithm 1 in comparison with the others.

To further show the efficiency of the proposed method, we calculated the number of wins for the three algorithms in terms of the elapsed CPU time (CPU wins), the number of iterations (Iter wins) and the number of function evaluations (Nf wins) and we also calculated the failures (Fails) of the three algorithms. The results are recorded in Table 1.

In addition, we adopted the performance profiles introduced by Dolan and Moré [29] to evaluate the required number of iterations and the number of function evaluations.

It follows from the results in Table 1 and Figure 2 that our algorithm (MSBFGS) performs the best among the three algorithms, either with respect to the number of iterations, or with respect to the elapsed CPU time.

In order to test the efficiency of the Algorithm 2 (MSBFGS2), we compared its performance for solving large-scale nonlinear symmetric equations with Algorithm 2.1 (NDDF) in [15] and Algorithm 2.1 (DFMPRP) in [13]. For the sake of fairness, we chose seven test problems, all from [15], where the relevant parameters of Algorithm 2 are same as those of NDDF in [15]. The values of parameters in DFMPRP are from [13]. The termination condition of all three algorithms is $\|F_k\| \leq 10^{-4}$, or the number of iterations exceeds 10^4, or the MATLAB R2010b crashes, or the CPU time exceeds 100 s.



(**a**) The number of iterations (**b**) The number of function evaluations

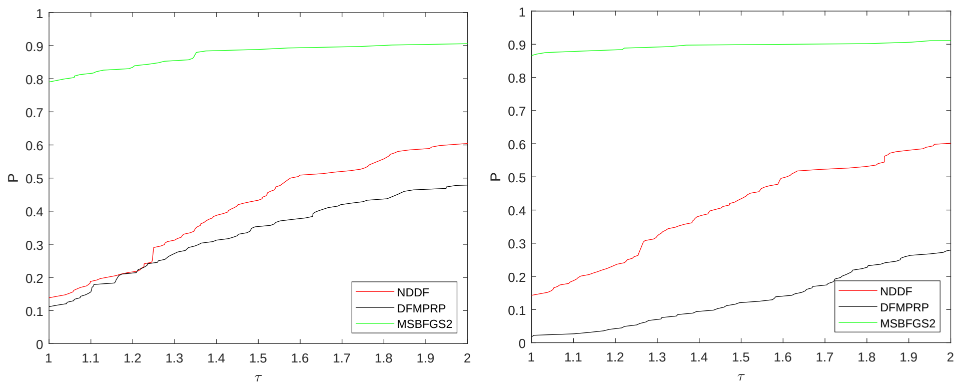
Figure 2. Comparison of numerical performance among three methods.

The numerical performance of all the algorithms is reported in Tables A2 and A3. Table A2 shows the numerical performance of all three algorithms with the fixed initial points x_1–x_6. Table A3 demonstrates the numerical performance of all the three algorithms with initial points x_7 and x_8 randomly generated by Matlab's Code "rand(n,1)" and "-rand(n,1)", respectively. Furthermore, we calculated the "CPU wins" , the "Iter wins", the "Nf wins" and the "Fails" of the three algorithms. The results are recorded in Table 2. We also adopted the performance profiles introduced by Dolan and Moré [29] to evaluate the required number of iterations and the required number of function evaluations of the three algorithms.

All the results of numerical performance in Figure 3 and Tables 2, A2 and A3 demonstrate that our algorithm (MSBFGS2) performs better than the other two algorithms. MSBFGS2 is more efficient and robust than the others since the failures of MSBFGS2 are the least among the three algorithms in the case that different initial guesses are chosen.

Table 1. Total number of wins or failures of algorithms.

Algorithm	CPU Wins	Iter Wins	Nf Wins	Fails
MBFGS	88	56	60	15
GNBFGS	95	67	65	0
MSBFGS	127	124	123	1



(**a**) The number of iterations (**b**) The number of function evaluations

Figure 3. Comparison of numerical performance among three methods.

Table 2. Total number of wins or failures of algorithms.

Algorithm	CPU Wins	Iter Wins	Nf Wins	Fails
NDDF	35	31	32	54
DFMPRP	14	25	4	46
MSBFGS2	180	177	194	16

5. Conclusions and Future Research

In this paper, we presented two derivative-free methods for solving nonlinear symmetric equations. For the first method, the direction is an approximate quasi-Newton direction and it can solve small-scale problems efficiently. For the second method, since it is not involved with the computation or storage of any matrix, it is applicable to solve the large scale system of nonlinear equations.

Global convergence theories of the developed algorithms were established. Compared with the similar algorithms, numerical tests demonstrated that our algorithms outperformed the others by costing less iterations, or less CPU time to find a solution with the same tolerance.

In future research, it would be valuable to deeply study the local convergence of the developed algorithms, in addition to the conducted analysis of global convergence in this paper. Additionally, our algorithms were designed only for the system of equations which is symmetric and satisfied with some relatively restrictive assumptions. Thus, it is interesting to study how to modify our algorithms to solve a more general system of equations.

Author Contributions: Z.W. conceived and designed the research plan and wrote the paper. J.G. performed the mathematical analysis, the development of the algorithm, experiments and wrote the paper. All authors have read and agreed to the published version of the manuscript.

Funding: This research is supported by the National Natural Science Foundation of China (Grant No. 71671190).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used to support the findings of this study are available from the corresponding author upon request. All the computer codes used in this study are available from the corresponding author upon request.

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

Appendix A. Numerical Results

Table A1. Numerical results of Problems 1–7.

P	dim	x0	MBFGS CPU	MBFGS Ni	MBFGS Nf	MBFGS Norm (F)	GNBFGS CPU	GNBFGS Ni	GNBFGS Nf	GNBFGS Norm (F)	MSBFGS CPU	MSBFGS Ni	MSBFGS Nf	MSBFGS Norm (F)
P1	10	x_1	0.0000	5	16	1.96×10^{-10}	0.0000	4	13	1.47×10^{-7}	$\underline{0.0000}$	$\underline{4}$	$\underline{13}$	$\underline{1.86 \times 10^{-12}}$
		x_2	0.0000	4	13	8.37×10^{-7}	0.0156	4	13	8.08×10^{-8}	$\underline{0.0000}$	$\underline{3}$	$\underline{10}$	9.23×10^{-7}
		x_3	0.0000	11	51	1.27×10^{-8}	0.0000	9	40	4.99×10^{-9}	0.0624	120	604	$\underline{2.29 \times 10^{-12}}$
		x_4	0.0000	7	35	8.98×10^{-7}	0.0000	9	31	9.28×10^{-10}	$\underline{0.0000}$	8	$\underline{25}$	$\underline{4.76 \times 10^{-8}}$
		x_5	0.0000	5	16	1.96×10^{-10}	0.0000	4	13	1.47×10^{-7}	$\underline{0.0000}$	$\underline{4}$	$\underline{13}$	$\underline{1.86 \times 10^{-12}}$
		x_6	0.0000	4	13	8.37×10^{-7}	0.0000	4	13	8.08×10^{-8}	$\underline{0.0000}$	$\underline{3}$	$\underline{10}$	9.23×10^{-7}
	50	x_1	0.0936	5	16	4.39×10^{-10}	0.0000	4	13	3.30×10^{-7}	$\underline{0.0000}$	$\underline{4}$	$\underline{13}$	$\underline{4.16 \times 10^{-12}}$
		x_2	0.0000	5	16	2.28×10^{-10}	0.0000	4	13	1.81×10^{-7}	$\underline{0.0000}$	$\underline{4}$	$\underline{13}$	$\underline{1.20 \times 10^{-12}}$
		x_3	0.0780	11	54	2.78×10^{-8}	0.0000	9	40	1.12×10^{-8}	0.0780	120	604	$\underline{5.13 \times 10^{-12}}$
		x_4	0.0780	7	34	4.99×10^{-8}	0.0000	9	31	2.07×10^{-9}	$\underline{0.0000}$	8	$\underline{25}$	1.06×10^{-7}
		x_5	0.0000	3	10	1.59×10^{-7}	0.0000	3	10	5.27×10^{-8}	$\underline{0.0000}$	$\underline{3}$	$\underline{10}$	$\underline{7.90 \times 10^{-12}}$
		x_6	0.0000	3	10	1.50×10^{-7}	0.0000	3	10	4.98×10^{-8}	$\underline{0.0000}$	$\underline{3}$	$\underline{10}$	$\underline{6.97 \times 10^{-12}}$
	100	x_1	0.0312	5	16	6.21×10^{-10}	0.0000	4	13	4.66×10^{-7}	$\underline{0.0000}$	$\underline{4}$	$\underline{13}$	$\underline{5.89 \times 10^{-12}}$
		x_2	0.0936	5	16	3.23×10^{-10}	0.0000	4	13	2.56×10^{-7}	$\underline{0.0000}$	$\underline{4}$	$\underline{13}$	$\underline{1.70 \times 10^{-12}}$
		x_3	0.0000	10	39	2.49×10^{-10}	0.0000	9	40	1.58×10^{-8}	0.2808	120	604	$\underline{7.25 \times 10^{-12}}$
		x_4	0.0000	9	39	2.45×10^{-7}	0.0000	9	31	2.93×10^{-9}	$\underline{0.0000}$	$\underline{8}$	$\underline{25}$	1.50×10^{-7}
		x_5	0.0000	3	10	6.91×10^{-9}	0.0000	3	10	2.30×10^{-9}	$\underline{0.0000}$	$\underline{2}$	$\underline{7}$	4.60×10^{-7}
		x_6	0.0000	3	10	6.72×10^{-9}	0.0000	3	10	2.23×10^{-9}	$\underline{0.0000}$	$\underline{2}$	$\underline{7}$	4.46×10^{-7}
	500	x_1	0.1092	5	16	1.39×10^{-9}	0.1248	5	16	2.71×10^{-11}	$\underline{0.0780}$	$\underline{4}$	$\underline{13}$	$\underline{1.32 \times 10^{-11}}$
		x_2	0.1248	5	16	7.18×10^{-10}	0.0780	4	13	5.71×10^{-7}	$\underline{0.0624}$	$\underline{4}$	$\underline{13}$	$\underline{3.81 \times 10^{-12}}$
		x_3	0.2028	9	40	2.26×10^{-10}	0.2496	9	40	3.53×10^{-8}	3.7128	120	604	$\underline{1.62 \times 10^{-11}}$
		x_4	0.2496	9	38	6.04×10^{-7}	0.2496	9	31	6.56×10^{-9}	0.2496	$\underline{8}$	$\underline{25}$	3.36×10^{-7}
		x_5	0.0624	2	7	4.06×10^{-7}	0.0624	2	7	2.70×10^{-7}	$\underline{0.0000}$	$\underline{2}$	$\underline{7}$	$\underline{1.63 \times 10^{-9}}$
		x_6	0.0000	2	7	4.04×10^{-7}	0.0312	2	7	2.68×10^{-7}	0.0780	$\underline{2}$	$\underline{7}$	$\underline{1.62 \times 10^{-9}}$
P2	10	x_1	0.0312	3	10	1.16×10^{-11}	0.0000	3	10	1.06×10^{-12}	$\underline{0.0000}$	$\underline{2}$	$\underline{7}$	1.28×10^{-9}
		x_2	0.0000	3	10	1.16×10^{-11}	0.0000	3	10	1.06×10^{-12}	$\underline{0.0000}$	$\underline{2}$	$\underline{7}$	1.28×10^{-9}
		x_3	0.0000	5	16	1.09×10^{-7}	0.0000	5	16	3.32×10^{-8}	$\underline{0.0000}$	$\underline{5}$	$\underline{16}$	8.89×10^{-8}
		x_4	0.0000	5	16	1.09×10^{-7}	0.0000	5	16	3.32×10^{-8}	$\underline{0.0000}$	$\underline{5}$	$\underline{16}$	8.89×10^{-8}
		x_5	0.0000	3	10	1.16×10^{-11}	0.0000	3	10	1.06×10^{-12}	$\underline{0.0000}$	$\underline{2}$	$\underline{7}$	1.28×10^{-9}
		x_6	0.0000	3	10	1.16×10^{-11}	0.0000	3	10	1.06×10^{-12}	$\underline{0.0000}$	$\underline{2}$	$\underline{7}$	1.28×10^{-9}
	50	x_1	0.0000	3	10	3.44×10^{-11}	0.0000	3	10	2.37×10^{-12}	$\underline{0.0000}$	$\underline{2}$	$\underline{7}$	2.86×10^{-9}
		x_2	0.0000	3	10	3.44×10^{-11}	0.0000	3	10	2.37×10^{-12}	$\underline{0.0000}$	$\underline{2}$	$\underline{7}$	2.86×10^{-9}

Table A1. *Cont.*

P	dim	x0	MBFGS				GNBFGS				MSBFGS			
			CPU	Ni	Nf	Norm (F)	CPU	Ni	Nf	Norm (F)	CPU	Ni	Nf	Norm (F)
	x_3	0.0000	5	16	2.43×10^{-7}	0.0000	5	16	7.42×10^{-8}	0.0000	5	16	1.99×10^{-7}	
	x_4	0.0000	5	16	2.43×10^{-7}	0.0000	5	16	7.42×10^{-8}	0.0936	5	16	1.99×10^{-7}	
	x_5	0.0000	2	7	1.07×10^{-8}	0.0000	2	7	5.06×10^{-9}	0.0000	2	7	1.46×10^{-15}	
	x_6	0.0000	2	7	1.07×10^{-8}	0.0000	2	7	5.06×10^{-9}	0.0000	2	7	1.46×10^{-15}	
	100	x_1	0.0000	3	10	5.78×10^{-11}	0.0000	3	10	3.35×10^{-12}	0.0000	2	7	4.05×10^{-9}
	x_2	0.0000	3	10	5.78×10^{-11}	0.0000	3	10	3.35×10^{-12}	0.0000	2	7	4.05×10^{-9}	
	x_3	0.0000	5	16	3.43×10^{-7}	0.0000	5	16	1.05×10^{-7}	0.0000	5	16	2.81×10^{-7}	
	x_4	0.0780	5	16	3.43×10^{-7}	0.0000	5	16	1.05×10^{-7}	0.0000	5	16	2.81×10^{-7}	
	x_5	0.0000	2	7	5.18×10^{-10}	0.0000	2	7	2.24×10^{-10}	0.0000	2	7	4.04×10^{-18}	
	x_6	0.0000	2	7	5.18×10^{-10}	0.0000	2	7	2.24×10^{-10}	0.0000	2	7	4.04×10^{-18}	
	500	x_1	0.1248	3	10	2.17×10^{-10}	0.0624	3	10	7.48×10^{-12}	0.0624	2	7	9.05×10^{-9}
	x_2	0.0624	3	10	2.17×10^{-10}	0.2028	3	10	7.48×10^{-12}	0.0936	2	7	9.05×10^{-9}	
	x_3	0.2340	5	16	7.64×10^{-7}	0.1560	5	16	2.35×10^{-7}	0.1248	5	16	6.28×10^{-7}	
	x_4	0.1248	5	16	7.64×10^{-7}	0.0780	5	16	2.35×10^{-7}	0.1248	5	16	6.28×10^{-7}	
	x_5	0.0000	1	4	1.20×10^{-7}	0.0624	1	4	1.20×10^{-7}	0.0000	1	4	1.20×10^{-7}	
	x_6	0.1092	1	4	1.20×10^{-7}	0.0000	1	4	1.20×10^{-7}	0.0000	1	4	1.20×10^{-7}	
P3	10	x_1	0.0624	804	5068	9.94×10^{-7}	0.1248	1182	7633	9.81×10^{-7}	0.0468	78	331	9.86×10^{-7}
	x_2	0.0312	281	1671	9.82×10^{-7}	0.1092	1080	6918	9.84×10^{-7}	0.0000	104	459	7.40×10^{-7}	
	x_3	0.0468	134	624	9.21×10^{-7}	0.0000	157	760	9.59×10^{-7}	0.0000	36	109	8.76×10^{-7}	
	x_4	0.0468	183	879	9.38×10^{-7}	0.0936	739	4591	9.60×10^{-7}	0.0000	65	250	9.01×10^{-7}	
	x_5	0.0936	804	5068	9.94×10^{-7}	0.1872	1182	7633	9.81×10^{-7}	0.0000	78	331	9.86×10^{-7}	
	x_6	0.0624	281	1671	9.82×10^{-7}	0.1716	1080	6918	9.84×10^{-7}	0.0000	104	459	7.40×10^{-7}	
	50	x_1	1.3416	873	5489	9.91×10^{-7}	1.8096	1317	8585	9.87×10^{-7}	0.2028	118	566	9.76×10^{-7}
	x_2	0.0000	11	36	2.32×10^{-7}	1.5288	1213	7782	9.97×10^{-7}	0.2808	160	916	9.95×10^{-7}	
	x_3	0.2028	130	592	9.31×10^{-7}	0.2964	162	762	9.81×10^{-7}	0.0936	37	112	8.65×10^{-7}	
	x_4	0.1560	164	779	9.76×10^{-7}	1.0764	762	4718	9.76×10^{-7}	0.1872	151	736	9.98×10^{-7}	
	x_5	0.0000	12	40	5.29×10^{-8}	1.4820	1268	8228	9.82×10^{-7}	0.1560	136	695	9.69×10^{-7}	
	x_6	0.0000	12	39	3.25×10^{-8}	1.5912	1238	8023	9.98×10^{-7}	0.1872	152	809	1.00×10^{-6}	

Table A1. *Cont.*

P	dim	x0	MBFGS				GNBFGS				MSBFGS			
			CPU	**Ni**	**Nf**	**Norm (F)**	**CPU**	**Ni**	**Nf**	**Norm (F)**	**CPU**	**Ni**	**Nf**	**Norm (F)**
	100	x_1	3.6192	890	5605	9.95×10^{-7}	5.4756	1349	8809	9.99×10^{-7}	0.2808	77	305	9.73×10^{-7}
		x_2	0.0000	11	36	2.07×10^{-7}	4.8048	1237	7932	9.83×10^{-7}	0.3744	109	478	9.75×10^{-7}
		x_3	0.3744	134	610	9.06×10^{-7}	0.5304	162	764	9.70×10^{-7}	0.0624	37	112	8.69×10^{-7}
		x_4	0.5928	171	810	9.99×10^{-7}	3.0576	777	4809	9.80×10^{-7}	0.4212	113	514	9.95×10^{-7}
		x_5	0.0624	12	40	6.16×10^{-8}	4.8516	1302	8446	9.99×10^{-7}	0.1872	81	323	9.80×10^{-7}
		x_6	0.0624	12	40	4.92×10^{-8}	5.1324	1301	8422	9.90×10^{-7}	0.2964	84	346	9.99×10^{-7}
	500	x_1	38.6258	947	5970	9.85×10^{-7}	60.1384	1428	9322	9.90×10^{-7}	3.6816	91	376	9.27×10^{-7}
		x_2	15.2413	374	2253	9.76×10^{-7}	54.5691	1313	8422	9.99×10^{-7}	4.9764	131	615	9.72×10^{-7}
		x_3	5.4444	145	662	9.43×10^{-7}	7.0200	176	827	9.90×10^{-7}	1.2636	39	118	6.18×10^{-7}
		x_4	7.4412	179	856	9.36×10^{-7}	34.4762	828	5128	9.79×10^{-7}	1.2792	40	121	5.78×10^{-7}
		x_5	0.4524	12	40	1.15×10^{-7}	59.2960	1371	8901	9.91×10^{-7}	4.2900	104	454	9.45×10^{-7}
		x_6	0.4524	12	40	1.11×10^{-7}	57.2368	1373	8892	9.97×10^{-7}	4.1028	98	420	8.05×10^{-7}
P4	10	x_1	0.0000	34	157	2.29×10^{-7}	0.0000	22	108	7.80×10^{-8}	0.0000	52	361	9.99×10^{-7}
		x_2	0.0000	26	113	3.21×10^{-7}	0.0000	23	113	8.78×10^{-8}	0.0468	40	274	8.03×10^{-8}
		x_3	0.0000	40	168	5.72×10^{-7}	0.0000	73	479	8.91×10^{-7}	0.0000	37	264	5.66×10^{-7}
		x_4	0.0000	44	178	7.37×10^{-7}	0.0000	46	195	7.51×10^{-7}	0.0000	53	377	1.94×10^{-7}
		x_5	0.0780	34	157	2.29×10^{-7}	0.0000	22	108	7.80×10^{-8}	0.0000	52	361	9.99×10^{-7}
		x_6	0.0000	26	113	3.21×10^{-7}	0.0000	23	113	8.78×10^{-8}	0.0000	40	274	8.03×10^{-8}
	50	x_1	0.0936	68	383	8.93×10^{-7}	0.0624	64	370	3.89×10^{-7}	0.0000	62	435	9.71×10^{-7}
		x_2	0.0000	66	364	8.46×10^{-7}	0.0780	63	367	8.24×10^{-7}	0.0000	40	274	9.21×10^{-7}
		x_3	0.1092	92	453	3.07×10^{-7}	0.1248	123	790	9.82×10^{-7}	0.0000	43	308	6.78×10^{-7}
		x_4	0.0624	94	487	9.36×10^{-7}	0.1092	146	698	8.53×10^{-7}	0.0624	48	347	6.51×10^{-7}
		x_5	0.0000	66	378	5.27×10^{-7}	0.0780	65	383	5.35×10^{-7}	0.0000	44	306	6.88×10^{-7}
		x_6	0.0624	66	378	7.88×10^{-7}	0.0000	64	389	8.35×10^{-7}	0.0468	38	264	8.79×10^{-7}
	100	x_1	0.0624	96	591	9.52×10^{-7}	0.1404	94	574	8.59×10^{-7}	0.0624	41	283	8.09×10^{-7}
		x_2	0.0624	99	591	9.55×10^{-7}	0.1404	91	563	9.85×10^{-7}	0.1092	40	279	6.69×10^{-7}
		x_3	0.1404	141	770	7.32×10^{-7}	0.2808	203	1419	6.97×10^{-7}	0.0000	47	332	8.20×10^{-7}
		x_4	0.2184	176	962	5.94×10^{-7}	0.3120	204	1075	6.36×10^{-7}	0.1092	51	368	7.80×10^{-7}
		x_5	0.1716	110	659	8.91×10^{-7}	0.1092	86	543	8.79×10^{-7}	0.0936	39	272	5.21×10^{-7}
		x_6	0.2340	125	710	6.71×10^{-7}	0.0624	100	633	9.61×10^{-7}	0.0780	41	287	5.46×10^{-7}

Table A1. *Cont.*

P	dim	x0	MBFGS				GNBFGS				MSBFGS			
			CPU	Ni	Nf	Norm (F)	CPU	Ni	Nf	Norm (F)	CPU	Ni	Nf	Norm (F)
	500	x_1	3.2136	109	667	8.71×10^{-7}	4.2432	138	842	9.33×10^{-7}	1.2480	42	294	9.07×10^{-7}
	x_2	3.1044	107	644	8.88×10^{-7}	3.1356	103	633	9.64×10^{-7}	1.1700	44	310	1.00×10^{-6}	
	x_3	7.4724	252	1549	9.08×10^{-7}	12.8389	456	2833	9.15×10^{-7}	1.2480	50	357	9.88×10^{-7}	
	x_4	7.6596	259	1581	9.28×10^{-7}	20.2801	653	3804	6.96×10^{-7}	1.5288	47	348	6.02×10^{-7}	
	x_5	3.1980	105	633	9.94×10^{-7}	4.4772	151	962	9.39×10^{-7}	1.0920	40	283	4.21×10^{-7}	
	x_6	4.2432	162	983	9.32×10^{-7}	3.1980	102	651	9.44×10^{-7}	0.9828	38	265	8.81×10^{-7}	
P5	10	x_1	0.0156	9	59	5.39×10^{-7}	0.0000	9	59	5.40×10^{-7}	0.0000	14	139	4.92×10^{-7}
	x_2	0.0000	9	59	5.53×10^{-7}	0.0000	9	59	5.54×10^{-7}	0.0000	14	139	4.92×10^{-7}	
	x_3	0.0000	11	72	5.10×10^{-7}	0.0000	11	72	8.43×10^{-7}	0.0000	15	149	4.50×10^{-7}	
	x_4	0.0000	10	62	3.20×10^{-7}	0.0000	10	62	9.74×10^{-7}	0.0000	15	149	4.48×10^{-7}	
	x_5	0.0000	9	59	5.39×10^{-7}	0.0000	9	59	5.40×10^{-7}	0.0000	14	139	4.92×10^{-7}	
	x_6	0.0000	9	59	5.53×10^{-7}	0.0000	9	59	5.54×10^{-7}	0.0000	14	139	4.92×10^{-7}	
	50	x_1	0.1248	37	280	6.78×10^{-7}	0.0000	42	317	6.90×10^{-7}	0.0000	18	177	9.53×10^{-7}
	x_2	0.0000	34	267	1.20×10^{-7}	0.1248	34	267	1.21×10^{-7}	0.0000	18	177	9.55×10^{-7}	
	x_3	0.0000	50	380	7.21×10^{-7}	0.1092	36	281	1.65×10^{-7}	0.0000	18	177	6.66×10^{-7}	
	x_4	0.0000	53	403	9.50×10^{-7}	0.0000	37	285	8.69×10^{-7}	0.0000	18	177	6.66×10^{-7}	
	x_5	0.0000	34	267	3.06×10^{-9}	0.0780	34	267	2.90×10^{-9}	0.0000	17	167	7.25×10^{-7}	
	x_6	0.0000	35	268	9.86×10^{-8}	0.0000	35	268	9.01×10^{-8}	0.0936	17	167	7.31×10^{-7}	
	100	x_1	0.0624	79	610	6.68×10^{-7}	0.0624	70	549	8.62×10^{-7}	0.0624	19	187	7.76×10^{-7}
	x_2	0.1092	66	527	5.24×10^{-7}	0.1092	69	539	6.10×10^{-7}	0.0000	19	187	7.76×10^{-7}	
	x_3	0.1716	78	610	8.88×10^{-7}	0.1560	73	579	9.85×10^{-7}	0.0000	19	187	4.02×10^{-7}	
	x_4	0.1716	77	603	8.51×10^{-7}	0.0624	84	654	8.96×10^{-7}	0.0624	19	187	4.02×10^{-7}	
	x_5	0.0936	65	517	4.80×10^{-7}	0.1092	69	541	1.79×10^{-7}	0.0624	15	147	7.65×10^{-7}	
	x_6	0.0624	65	517	9.82×10^{-7}	0.0624	65	517	5.24×10^{-7}	0.0000	15	147	7.58×10^{-7}	
	500	x_1	2.3400	81	662	5.23×10^{-7}	2.3868	80	655	9.55×10^{-7}	0.5304	20	196	4.53×10^{-7}
	x_2	2.2152	77	630	7.75×10^{-7}	2.1216	79	648	9.81×10^{-7}	0.5616	20	196	4.53×10^{-7}	
	x_3	2.9484	98	800	8.37×10^{-7}	2.8392	91	745	7.14×10^{-7}	0.6084	22	216	7.17×10^{-7}	
	x_4	2.8392	98	800	8.21×10^{-7}	2.8236	91	745	6.67×10^{-7}	0.6708	22	216	7.17×10^{-7}	
	x_5	1.8408	59	482	7.82×10^{-7}	2.1996	59	482	7.91×10^{-7}	0.3588	11	107	7.59×10^{-7}	
	x_6	1.8096	59	483	9.54×10^{-7}	1.7940	59	483	9.87×10^{-7}	0.3900	11	107	7.60×10^{-7}	

Table A1. *Cont.*

P	dim	x0	MBFGS				GNBFGS				MSBFGS			
			CPU	Ni	Nf	Norm (F)	CPU	Ni	Nf	Norm (F)	CPU	Ni	Nf	Norm (F)
P6	10	x_1	0.0000	108	484	9.16×10^{-7}	0.0000	107	542	8.70×10^{-7}	0.0468	267	2333	9.73×10^{-7}
	x_2	0.0000	111	488	8.34×10^{-7}	0.0468	125	640	9.45×10^{-7}	0.0000	242	1940	9.41×10^{-7}	
	x_3	0.0000	90	416	9.73×10^{-7}	0.0000	52	220	7.73×10^{-7}	0.0000	50	380	8.72×10^{-7}	
	x_4	F	F	F	F	0.0312	222	1224	9.11×10^{-7}	0.0000	287	2801	8.81×10^{-7}	
	x_5	0.0312	108	484	9.16×10^{-7}	0.0000	107	542	8.70×10^{-7}	0.0000	267	2333	9.73×10^{-7}	
	x_6	0.0156	111	488	8.34×10^{-7}	0.0000	125	640	9.45×10^{-7}	0.0000	242	1940	9.41×10^{-7}	
	50	x_1	0.0624	148	789	8.70×10^{-7}	0.0624	115	625	8.94×10^{-7}	0.1092	141	1111	9.97×10^{-7}
	x_2	0.0624	139	740	8.90×10^{-7}	0.0624	148	823	9.32×10^{-7}	0.3588	569	4552	9.85×10^{-7}	
	x_3	0.0624	113	611	6.27×10^{-7}	0.0624	103	572	9.11×10^{-7}	0.1716	280	2468	9.59×10^{-7}	
	x_4	F	F	F	F	0.1092	195	1113	8.38×10^{-7}	0.8268	1065	10678	9.21×10^{-7}	
	x_5	0.1404	182	1028	9.64×10^{-7}	0.0000	102	565	8.84×10^{-7}	0.4524	684	5784	9.87×10^{-7}	
	x_6	0.0624	99	554	9.81×10^{-7}	0.0624	115	612	8.33×10^{-7}	0.5928	942	7985	9.95×10^{-7}	
	100	x_1	0.1248	187	1118	9.12×10^{-7}	0.1716	160	999	9.25×10^{-7}	0.2184	161	1315	9.34×10^{-7}
	x_2	0.2808	186	1151	9.84×10^{-7}	0.2808	171	1031	9.70×10^{-7}	0.6396	520	4163	9.80×10^{-7}	
	x_3	0.2808	178	1058	9.83×10^{-7}	0.1872	176	1049	9.38×10^{-7}	0.4056	287	2537	9.99×10^{-7}	
	x_4	F	F	F	F	0.4836	321	2078	9.09×10^{-7}	4.8048	3042	35312	8.44×10^{-7}	
	x_5	0.1248	160	974	9.21×10^{-7}	0.2340	170	1022	7.19×10^{-7}	1.4352	940	8407	9.88×10^{-7}	
	x_6	0.1716	163	986	8.04×10^{-7}	0.1716	170	1018	9.93×10^{-7}	1.0296	915	7757	9.89×10^{-7}	
	500	x_1	17.0509	576	4354	9.63×10^{-7}	15.5377	538	3805	9.92×10^{-7}	19.8433	662	5897	9.96×10^{-7}
	x_2	F	F	F	F	17.3317	559	4014	9.49×10^{-7}	15.1945	490	3929	9.86×10^{-7}	
	x_3	17.5345	558	3934	9.70×10^{-7}	18.0961	548	3909	9.26×10^{-7}	5.6472	181	1556	9.39×10^{-7}	
	x_4	F	F	F	F	29.5934	961	8274	8.95×10^{-7}	F	F	F	F	
	x_5	F	F	F	F	16.2865	534	3793	9.47×10^{-7}	14.1649	467	3744	9.95×10^{-7}	
	x_6	F	F	F	F	16.5205	544	3847	9.40×10^{-7}	13.5721	444	3564	9.95×10^{-7}	
P7	10	x_1	0.0000	1	4	3.28×10^{-12}	0.0000	1	4	3.28×10^{-12}	0.0000	1	4	3.28×10^{-12}
	x_2	0.0000	1	4	5.61×10^{-11}	0.0000	1	4	5.61×10^{-11}	0.0000	1	4	5.61×10^{-11}	
	x_3	0.0000	0	1	0.00	0.0000	0	1	0.00	0.0000	0	1	0.00	
	x_4	0.0000	1	4	3.30×10^{-10}	0.0000	1	4	3.30×10^{-10}	0.0000	1	4	3.30×10^{-10}	

Table A1. *Cont.*

P	dim	x0	MBFGS				GNBFGS				MSBFGS			
			CPU	**Ni**	**Nf**	**Norm (F)**	**CPU**	**Ni**	**Nf**	**Norm (F)**	**CPU**	**Ni**	**Nf**	**Norm (F)**
	x_5	0.0000	1	4	3.28×10^{-12}	0.0000	1	4	3.28×10^{-12}	0.0000	1	4	3.28×10^{-12}	
	x_6	0.0000	1	4	5.61×10^{-11}	0.0000	1	4	5.61×10^{-11}	0.0000	1	4	5.61×10^{-11}	
	50	x_1	0.0000	1	4	1.18×10^{-7}	0.0000	1	4	1.18×10^{-7}	0.0000	1	4	1.18×10^{-7}
	x_2	F	F	F	F	0.0000	2	10	1.32×10^{-7}	0.0000	2	7	0.00	
	x_3	0.0000	0	1	0.00	0.0000	0	1	0.00	0.0000	0	1	0.00	
	x_4	F	F	F	F	0.0000	4	24	0.00	0.0000	2	7	0.00	
	x_5	0.0000	1	4	9.80×10^{-8}	0.0000	1	4	9.80×10^{-8}	0.0000	1	4	9.80×10^{-8}	
	x_6	0.0000	1	4	4.55×10^{-8}	0.0000	1	4	4.55×10^{-8}	0.0000	1	4	4.55×10^{-8}	
	100	x_1	F	F	F	F	0.0000	4	18	0.00	0.0000	2	7	0.00
	x_2	F	F	F	F	0.0000	4	17	0.00	0.0000	2	7	0.00	
	x_3	0.0000	0	1	0.00	0.0000	0	1	0.00	0.0000	0	1	0.00	
	x_4	F	F	F	F	0.0000	4	26	0.00	0.0000	2	7	0.00	
	x_5	0.0000	2	7	2.51×10^{-7}	0.0000	2	7	2.51×10^{-7}	0.0000	2	7	0.00	
	x_6	0.0000	1	4	2.61×10^{-7}	0.0000	1	4	2.61×10^{-7}	0.0000	1	4	2.61×10^{-7}	
	500	x_1	F	F	F	F	0.1248	4	23	0.00	0.0000	2	7	0.00
	x_2	F	F	F	F	0.0156	4	23	0.00	0.0468	2	7	0.00	
	x_3	0.0000	0	1	0.00	0.0000	0	1	0.00	0.0000	0	1	0.00	
	x_4	F	F	F	F	0.0936	4	30	0.00	0.0624	2	7	0.00	
	x_5	0.0624	3	10	0.00	0.0780	3	10	0.00	0.0780	2	7	0.00	
	x_6	0.1248	3	10	0.00	0.0156	3	10	0.00	0.0624	2	7	0.00	

Table A2. Numerical results of the 7 problems in [15] with fixed initial points.

P	dim	x0	NDDF				DFMPRP				MSBFGS2			
			CPU	**Ni**	**Nf**	**Norm (F)**	**CPU**	**Ni**	**Nf**	**Norm (F)**	**CPU**	**Ni**	**Nf**	**Norm (F)**
P1	10000	x_1	F	F	F	F	0.1716	11	74	3.55×10^{-5}	0.0624	3	11	5.65×10^{-7}
		x_2	F	F	F	F	0.0000	9	61	6.61×10^{-5}	0.1092	3	11	4.33×10^{-7}
		x_3	F	F	F	F	F	F	F	F	0.0000	8	31	4.52×10^{-8}
		x_4	F	F	F	F	0.0624	10	71	3.98×10^{-5}	0.0000	6	23	8.90×10^{-7}
		x_5	0.0000	1	3	5.00×10^{-7}	0.1092	5	33	5.85×10^{-5}	0.0000	1	3	5.00×10^{-7}
		x_6	0.0000	1	3	5.00×10^{-7}	0.0000	5	33	5.87×10^{-5}	0.0000	1	3	5.00×10^{-7}
	100000	x_1	F	F	F	F	0.3744	12	81	2.80×10^{-5}	0.0000	3	11	1.79×10^{-6}
		x_2	F	F	F	F	0.2496	10	68	5.23×10^{-5}	0.0000	3	11	1.37×10^{-6}
		x_3	F	F	F	F	F	F	F	F	0.1404	8	31	1.43×10^{-7}
		x_4	F	F	F	F	0.3744	11	78	3.15×10^{-5}	0.1248	6	23	2.81×10^{-6}
		x_5	0.0000	1	3	1.58×10^{-8}	0.1872	4	26	7.41×10^{-5}	0.0000	1	3	1.58×10^{-8}
		x_6	0.0000	1	3	1.58×10^{-8}	0.1248	4	26	7.41×10^{-5}	0.0312	1	3	1.58×10^{-8}
	500000	x_1	F	F	F	F	1.6536	12	81	6.27×10^{-5}	0.1872	3	11	4.00×10^{-6}
		x_2	F	F	F	F	1.6224	11	75	2.92×10^{-5}	0.3276	3	11	3.06×10^{-6}
		x_3	F	F	F	F	F	F	F	F	0.6396	8	31	3.20×10^{-7}
		x_4	F	F	F	F	1.4664	11	78	7.04×10^{-5}	0.3744	6	23	6.29×10^{-6}
		x_5	0.0624	1	3	1.41×10^{-9}	0.4524	4	26	3.31×10^{-5}	0.0624	1	3	1.41×10^{-9}
		x_6	0.0468	1	3	1.41×10^{-9}	0.5304	4	26	3.31×10^{-5}	0.0624	1	3	1.41×10^{-9}
	1000000	x_1	F	F	F	F	3.0108	12	81	8.87×10^{-5}	0.4368	3	11	5.65×10^{-6}
		x_2	F	F	F	F	3.1356	11	75	4.13×10^{-5}	0.4836	3	11	4.33×10^{-6}
		x_3	F	F	F	F	F	F	F	F	1.0608	8	31	4.52×10^{-7}
		x_4	F	F	F	F	3.2292	11	78	9.95×10^{-5}	0.8268	6	23	8.90×10^{-6}
		x_5	0.0624	1	3	5.00×10^{-10}	0.7020	3	19	9.37×10^{-5}	0.1404	1	3	5.00×10^{-10}
		x_6	0.1248	1	3	5.00×10^{-10}	0.6864	3	19	9.38×10^{-5}	0.1248	1	3	5.00×10^{-10}
P2	10000	x_1	F	F	F	F	0.1092	10	68	7.81×10^{-5}	0.0000	2	7	6.16×10^{-10}
		x_2	F	F	F	F	0.0624	10	68	7.81×10^{-5}	0.0000	2	7	6.16×10^{-10}
		x_3	F	F	F	F	0.1872	12	83	5.01×10^{-5}	0.0000	3	11	2.01×10^{-5}
		x_4	F	F	F	F	0.1716	12	83	5.01×10^{-5}	0.0000	3	11	2.01×10^{-5}
		x_5	0.0000	1	3	1.67×10^{-11}	0.0000	5	33	5.86×10^{-5}	0.0000	1	3	1.67×10^{-11}
		x_6	0.0000	1	3	1.67×10^{-11}	0.0000	5	33	5.86×10^{-5}	0.0000	1	3	1.67×10^{-11}

Table A2. *Cont.*

P	dim	x0	NDDF				DFMPRP				MSBFGS2			
			CPU	Ni	Nf	Norm (F)	CPU	Ni	Nf	Norm (F)	CPU	Ni	Nf	Norm (F)
	100000	x_1	F	F	F	F	0.3900	11	75	6.17×10^{-5}	0.0156	2	7	1.95×10^{-9}
		x_2	F	F	F	F	0.5460	11	75	6.17×10^{-5}	0.0156	2	7	1.95×10^{-9}
		x_3	F	F	F	F	0.4992	13	90	3.96×10^{-5}	0.0780	3	11	6.34×10^{-5}
		x_4	F	F	F	F	0.4836	13	90	3.96×10^{-5}	0.0936	3	11	6.34×10^{-5}
		x_5	0.0000	1	3	5.27×10^{-14}	0.1560	4	26	7.41×10^{-5}	0.0000	1	3	5.27×10^{-14}
		x_6	0.0624	1	3	5.27×10^{-14}	0.0936	4	26	7.41×10^{-5}	0.0000	1	3	5.27×10^{-14}
	500000	x_1	F	F	F	F	1.8720	12	82	3.45×10^{-5}	0.0936	2	7	4.36×10^{-9}
		x_2	F	F	F	F	1.7784	12	82	3.45×10^{-5}	0.1404	2	7	4.36×10^{-9}
		x_3	F	F	F	F	2.5428	13	90	8.86×10^{-5}	0.3588	4	15	5.98×10^{-17}
		x_4	F	F	F	F	2.9640	13	90	8.86×10^{-5}	0.2496	4	15	5.98×10^{-17}
		x_5	0.0312	1	3	9.43×10^{-16}	0.7176	4	26	3.31×10^{-5}	0.0156	1	3	9.43×10^{-16}
		x_6	0.0312	1	3	9.43×10^{-16}	0.6396	4	26	3.31×10^{-5}	0.0780	1	3	9.43×10^{-16}
	1000000	x_1	F	F	F	F	3.7440	12	82	4.88×10^{-5}	0.2964	2	7	6.16×10^{-9}
		x_2	F	F	F	F	3.9000	12	82	4.88×10^{-5}	0.1560	2	7	6.16×10^{-9}
		x_3	F	F	F	F	4.6956	14	97	3.13×10^{-5}	0.6084	4	15	3.67×10^{-17}
		x_4	F	F	F	F	4.4928	14	97	3.13×10^{-5}	0.7332	4	15	3.67×10^{-17}
		x_5	0.0468	1	3	1.67×10^{-16}	1.0296	3	19	9.38×10^{-5}	0.0468	1	3	1.67×10^{-16}
		x_6	0.0468	1	3	1.67×10^{-16}	0.7176	3	19	9.38×10^{-5}	0.1404	1	3	1.67×10^{-16}
P3	10000	x_1	0.3276	87	625	9.94×10^{-5}	0.1560	48	498	5.88×10^{-5}	0.3432	66	452	6.51×10^{-5}
		x_2	0.2964	74	520	9.73×10^{-5}	0.3900	56	573	6.30×10^{-5}	0.1716	48	326	4.52×10^{-5}
		x_3	F	F	F	F	F	F	F	F	9.0325	2719	14327	9.93×10^{-5}
		x_4	F	F	F	F	F	F	F	F	F	F	F	F
		x_5	0.3276	79	563	9.77×10^{-5}	F	F	F	F	0.1872	45	309	7.00×10^{-5}
		x_6	0.3900	67	474	8.60×10^{-5}	F	F	F	F	0.0468	35	242	9.36×10^{-5}
	100000	x_1	2.2464	80	570	9.38×10^{-5}	2.0748	53	553	5.57×10^{-5}	1.5444	59	410	9.27×10^{-5}
		x_2	2.3556	82	580	9.00×10^{-5}	1.8096	47	485	8.74×10^{-5}	1.3416	56	381	9.20×10^{-5}
		x_3	F	F	F	F	F	F	F	F	56.1448	2779	14670	9.81×10^{-5}
		x_4	F	F	F	F	F	F	F	F	F	F	F	F
		x_5	3.1200	106	761	9.30×10^{-5}	2.5584	69	692	9.27×10^{-5}	1.4352	57	392	9.72×10^{-5}
		x_6	2.2932	95	677	8.81×10^{-5}	1.7628	52	536	8.26×10^{-5}	1.0920	39	267	9.77×10^{-5}

Table A2. *Cont.*

P	dim	x0	NDDF				DFMPRP				MSBFGS2			
			CPU	Ni	Nf	Norm (F)	CPU	Ni	Nf	Norm (F)	CPU	Ni	Nf	Norm (F)
	500000	x_1	10.1401	67	479	6.34×10^{-5}	F	F	F	F	7.4412	55	385	7.61×10^{-5}
		x_2	11.6065	74	520	6.45×10^{-5}	9.3445	44	479	7.44×10^{-5}	7.4880	53	366	9.88×10^{-5}
		x_3	F	F	F	F	F	F	F	F	F	F	F	F
		x_4	F	F	F	F	F	F	F	F	F	F	F	F
		x_5	10.6549	64	448	7.95×10^{-5}	10.4521	52	548	8.44×10^{-5}	3.6348	25	175	5.30×10^{-5}
		x_6	12.1525	79	554	6.70×10^{-5}	11.1073	56	576	7.94×10^{-5}	3.5880	25	175	4.79×10^{-5}
	1000000	x_1	F	F	F	F	22.1989	59	646	7.32×10^{-5}	19.9369	73	505	9.73×10^{-5}
		x_2	22.1053	80	563	7.62×10^{-5}	29.4062	76	791	9.50×10^{-5}	9.8125	39	268	9.48×10^{-5}
		x_3	F	F	F	F	F	F	F	F	F	F	F	F
		x_4	F	F	F	F	F	F	F	F	F	F	F	F
		x_5	21.8089	74	525	8.24×10^{-5}	20.3893	56	568	9.30×10^{-5}	15.1945	58	396	8.79×10^{-5}
		x_6	22.8073	75	530	9.75×10^{-5}	20.9665	57	584	9.36×10^{-5}	18.0025	63	434	7.58×10^{-5}
P4	10000	x_1	1.2324	102	716	7.59×10^{-5}	0.4212	23	249	7.98×10^{-5}	0.5304	31	210	9.43×10^{-5}
		x_2	1.2324	99	716	6.37×10^{-5}	F	F	F	F	0.3432	33	225	7.26×10^{-5}
		x_3	0.6396	60	423	9.11×10^{-5}	F	F	F	F	0.4836	48	337	3.20×10^{-5}
		x_4	1.3728	109	767	5.00×10^{-5}	0.9048	56	566	7.69×10^{-5}	0.2652	23	162	6.99×10^{-5}
		x_5	0.7644	67	489	5.95×10^{-5}	0.9360	56	563	7.74×10^{-5}	0.3120	33	230	9.46×10^{-5}
		x_6	0.9048	69	503	6.38×10^{-5}	0.7488	45	456	9.46×10^{-5}	0.3120	29	201	7.19×10^{-5}
	100000	x_1	6.3336	145	1016	5.55×10^{-5}	2.0280	29	308	8.39×10^{-5}	0.8580	25	171	5.97×10^{-5}
		x_2	4.3056	81	595	5.72×10^{-5}	F	F	F	F	1.2168	26	185	9.51×10^{-5}
		x_3	4.4148	94	672	6.11×10^{-5}	F	F	F	F	F	F	F	F
		x_4	5.7096	133	941	5.40×10^{-5}	F	F	F	F	10.6705	242	1693	9.95×10^{-5}
		x_5	10.8109	236	1669	9.65×10^{-5}	4.5396	76	743	8.82×10^{-5}	1.0140	26	182	8.04×10^{-5}
		x_6	12.4801	278	1963	9.33×10^{-5}	3.8376	61	588	9.79×10^{-5}	1.3884	33	230	6.18×10^{-5}
	500000	x_1	18.7513	80	563	8.60×10^{-5}	11.4037	36	382	7.71×10^{-5}	5.1636	22	164	8.06×10^{-5}
		x_2	27.3314	108	783	4.19×10^{-5}	F	F	F	F	6.8484	34	243	6.07×10^{-5}
		x_3	22.2457	104	727	9.42×10^{-5}	F	F	F	F	F	F	F	F
		x_4	25.4438	108	772	4.53×10^{-5}	F	F	F	F	9.5317	41	290	5.30×10^{-5}
		x_5	16.0057	65	471	6.16×10^{-5}	22.5421	83	787	7.39×10^{-5}	F	F	F	F
		x_6	15.1477	65	471	6.10×10^{-5}	29.6246	106	1001	7.35×10^{-5}	F	F	F	F

Table A2. *Cont.*

P	dim	x0	NDDF				DFMPRP				MSBFGS2			
			CPU	Ni	Nf	Norm (F)	CPU	Ni	Nf	Norm (F)	CPU	Ni	Nf	Norm (F)
	1000000	x_1	26.1146	58	396	9.70×10^{-5}	18.0025	30	318	6.99×10^{-5}	13.2289	30	214	8.53×10^{-5}
		x_2	43.9299	88	641	5.95×10^{-5}	F	F	F	F	35.1938	79	559	9.75×10^{-5}
		x_3	50.0763	107	766	7.27×10^{-5}	F	F	F	F	F	F	F	F
		x_4	41.1687	89	646	6.58×10^{-5}	F	F	F	F	32.0894	76	544	8.00×10^{-5}
		x_5	F	F	F	F	40.9815	71	696	5.67×10^{-5}	15.9277	36	262	7.99×10^{-5}
		x_6	F	F	F	F	46.7847	84	794	8.32×10^{-5}	15.3193	33	243	6.13×10^{-5}
P5	10000	x_1	0.1872	63	393	7.45×10^{-5}	0.2808	57	523	9.51×10^{-5}	0.1872	43	260	9.24×10^{-5}
		x_2	0.1560	58	359	6.55×10^{-5}	0.3276	51	482	9.76×10^{-5}	0.2184	44	263	9.33×10^{-5}
		x_3	0.2184	66	416	7.21×10^{-5}	0.3276	48	462	8.48×10^{-5}	0.1092	46	279	8.87×10^{-5}
		x_4	0.2184	57	356	8.81×10^{-5}	0.1716	56	529	7.64×10^{-5}	0.0624	48	287	8.86×10^{-5}
		x_5	0.3276	59	372	8.83×10^{-5}	0.2184	50	476	9.29×10^{-5}	0.1716	46	277	6.38×10^{-5}
		x_6	0.2184	61	380	5.17×10^{-5}	0.2808	56	528	9.65×10^{-5}	0.2184	57	342	6.78×10^{-5}
	100000	x_1	1.1856	55	342	7.59×10^{-5}	1.7628	54	505	9.86×10^{-5}	1.1388	49	293	9.38×10^{-5}
		x_2	1.3416	64	403	8.71×10^{-5}	2.0124	70	652	8.63×10^{-5}	0.9672	49	298	8.25×10^{-5}
		x_3	1.0764	55	351	8.08×10^{-5}	1.7316	58	540	7.87×10^{-5}	0.9360	40	244	9.46×10^{-5}
		x_4	1.0608	58	364	7.90×10^{-5}	2.2152	72	664	8.05×10^{-5}	1.0296	50	297	9.74×10^{-5}
		x_5	1.1232	54	345	9.11×10^{-5}	1.8252	60	564	6.98×10^{-5}	0.9828	44	266	9.25×10^{-5}
		x_6	1.1700	54	345	9.68×10^{-5}	1.9188	58	546	9.60×10^{-5}	1.1700	53	317	6.33×10^{-5}
	500000	x_1	9.8593	68	436	8.54×10^{-5}	10.7173	58	544	9.43×10^{-5}	6.3492	48	291	9.11×10^{-5}
		x_2	7.3320	54	343	9.05×10^{-5}	9.0325	53	509	8.04×10^{-5}	6.6300	52	314	7.39×10^{-5}
		x_3	9.1573	61	390	9.16×10^{-5}	9.3913	54	524	9.64×10^{-5}	5.8968	49	297	9.96×10^{-5}
		x_4	9.1729	65	409	6.99×10^{-5}	11.5753	59	554	9.60×10^{-5}	6.1932	48	291	9.49×10^{-5}
		x_5	9.7501	70	446	9.58×10^{-5}	10.0465	54	515	8.61×10^{-5}	5.1480	46	274	9.18×10^{-5}
		x_6	10.0465	74	471	8.81×10^{-5}	9.9061	54	510	9.37×10^{-5}	5.3040	42	254	9.39×10^{-5}
	1000000	x_1	12.8545	47	300	9.63×10^{-5}	19.2349	52	495	9.05×10^{-5}	12.2617	47	285	9.79×10^{-5}
		x_2	17.7529	64	404	8.89×10^{-5}	19.4377	57	534	8.71×10^{-5}	12.2773	47	284	9.13×10^{-5}
		x_3	16.5985	57	367	9.40×10^{-5}	18.3145	53	514	8.15×10^{-5}	9.7969	41	252	9.05×10^{-5}
		x_4	17.6437	58	365	9.47×10^{-5}	21.3565	57	547	9.38×10^{-5}	13.3381	53	321	9.97×10^{-5}
		x_5	16.5517	58	374	9.98×10^{-5}	23.0569	61	574	7.28×10^{-5}	10.7797	42	254	8.93×10^{-5}
		x_6	21.1069	73	470	8.30×10^{-5}	17.9089	50	484	9.87×10^{-5}	9.8437	48	290	9.45×10^{-5}

Table A2. *Cont.*

P	dim	x0	NDDF				DFMPRP				MSBFGS2			
			CPU	Ni	Nf	Norm (F)	CPU	Ni	Nf	Norm (F)	CPU	Ni	Nf	Norm (F)
P6	10000	x_1	0.2340	29	356	8.90×10^{-5}	0.3588	44	726	7.57×10^{-5}	0.3120	25	302	4.15×10^{-5}
		x_2	0.2028	18	249	6.26×10^{-5}	0.2496	24	394	9.27×10^{-5}	0.1716	27	331	1.57×10^{-5}
		x_3	F	F	F	F	0.4212	46	771	6.08×10^{-5}	0.2184	31	389	7.49×10^{-6}
		x_4	0.3900	39	496	5.60×10^{-5}	0.2652	30	499	5.28×10^{-5}	1.6380	178	2328	9.74×10^{-5}
		x_5	0.4368	61	714	3.56×10^{-5}	0.3276	31	617	9.65×10^{-5}	0.2808	28	345	3.73×10^{-5}
		x_6	0.3900	32	382	4.98×10^{-5}	0.4368	33	658	8.18×10^{-5}	0.2652	30	365	4.52×10^{-5}
	100000	x_1	2.7612	64	730	7.66×10^{-5}	2.7612	45	745	5.99×10^{-5}	1.4196	27	325	4.56×10^{-5}
		x_2	1.2480	20	271	9.77×10^{-5}	1.6224	26	428	6.80×10^{-5}	1.6380	27	331	4.95×10^{-5}
		x_3	F	F	F	F	3.1356	51	859	7.87×10^{-5}	1.4352	31	389	2.37×10^{-5}
		x_4	1.5132	23	355	8.61×10^{-5}	2.1528	32	531	7.49×10^{-5}	10.3273	200	2614	9.53×10^{-5}
		x_5	1.5600	35	410	5.52×10^{-5}	1.9968	26	516	8.87×10^{-5}	1.5756	35	424	9.13×10^{-5}
		x_6	1.3260	23	297	2.67×10^{-5}	2.3244	29	577	8.33×10^{-5}	1.0920	23	286	8.68×10^{-5}
	500000	x_1	19.7497	69	785	7.69×10^{-5}	17.2537	49	810	3.66×10^{-5}	6.6924	28	337	7.63×10^{-5}
		x_2	6.7392	21	282	7.50×10^{-5}	8.3149	27	444	3.39×10^{-5}	7.7688	28	343	7.28×10^{-5}
		x_3	F	F	F	F	16.5829	49	823	8.70×10^{-5}	7.7532	31	389	5.31×10^{-5}
		x_4	13.7281	37	510	9.35×10^{-5}	10.5613	35	581	7.86×10^{-5}	62.7748	215	2809	9.57×10^{-5}
		x_5	23.5874	74	915	7.18×10^{-5}	11.6221	31	617	7.23×10^{-5}	6.9420	23	287	7.50×10^{-5}
		x_6	F	F	F	F	14.5237	37	738	7.64×10^{-5}	7.0356	24	298	2.68×10^{-5}
	1000000	x_1	20.4517	37	441	8.63×10^{-5}	28.8758	49	812	9.26×10^{-5}	15.4285	29	347	3.20×10^{-5}
		x_2	20.3425	37	436	3.12×10^{-5}	15.5845	27	444	4.69×10^{-5}	14.0089	29	355	4.89×10^{-5}
		x_3	F	F	F	F	32.1830	46	774	8.76×10^{-5}	16.9105	31	389	7.50×10^{-5}
		x_4	F	F	F	F	21.6217	37	613	5.54×10^{-5}	F	F	F	F
		x_5	22.7137	40	463	5.55×10^{-5}	25.9430	34	677	6.85×10^{-5}	14.3365	27	332	7.83×10^{-5}
		x_6	60.4036	104	1197	7.67×10^{-5}	25.8026	37	737	6.91×10^{-5}	17.0353	32	390	2.16×10^{-5}
P7	10000	x_1	0.8580	115	1106	9.91×10^{-5}	1.0452	94	1352	9.98×10^{-5}	0.3276	63	572	7.90×10^{-5}
		x_2	0.6864	115	1100	8.92×10^{-5}	0.7956	78	1095	8.56×10^{-5}	0.3276	51	470	8.36×10^{-5}
		x_3	0.8580	117	1106	9.28×10^{-5}	1.7004	197	2809	8.56×10^{-5}	0.6240	82	776	5.81×10^{-5}
		x_4	0.4992	83	789	8.12×10^{-5}	0.7332	81	1127	8.95×10^{-5}	0.2808	55	497	9.46×10^{-5}
		x_5	0.3276	44	422	8.06×10^{-5}	0.3900	38	534	9.13×10^{-5}	0.2028	28	261	9.80×10^{-5}
		x_6	0.2808	38	365	9.66×10^{-5}	0.3120	35	490	8.47×10^{-5}	0.2028	30	280	8.79×10^{-5}

Table A2. *Cont.*

P	dim	x0	NDDF				DFMPRP				MSBFGS2			
			CPU	Ni	Nf	Norm (F)	CPU	Ni	Nf	Norm (F)	CPU	Ni	Nf	Norm (F)
	100000	x_1	3.5256	91	876	9.83×10^{-5}	5.6472	119	1641	7.16×10^{-5}	2.4960	73	669	8.97×10^{-5}
		x_2	2.7144	70	664	9.68×10^{-5}	3.9624	79	1090	9.90×10^{-5}	1.7472	49	451	8.62×10^{-5}
		x_3	1.9812	54	507	6.34×10^{-5}	10.4521	197	2921	8.97×10^{-5}	2.6208	73	694	7.84×10^{-5}
		x_4	4.8984	129	1214	8.21×10^{-5}	4.8204	101	1375	9.77×10^{-5}	2.0436	54	493	8.52×10^{-5}
		x_5	1.7160	40	384	8.11×10^{-5}	1.9188	34	478	6.05×10^{-5}	0.9828	26	245	6.81×10^{-5}
		x_6	1.6692	40	384	8.13×10^{-5}	1.7472	37	520	6.48×10^{-5}	0.9204	25	235	7.47×10^{-5}
	500000	x_1	27.0818	122	1175	8.79×10^{-5}	28.6418	115	1535	9.80×10^{-5}	11.2321	59	539	6.96×10^{-5}
		x_2	19.1257	88	844	8.49×10^{-5}	29.9210	105	1502	9.54×10^{-5}	11.2321	61	557	6.76×10^{-5}
		x_3	17.0041	78	732	9.30×10^{-5}	67.8604	245	3540	9.88×10^{-5}	13.7437	71	665	8.59×10^{-5}
		x_4	21.4345	93	863	9.32×10^{-5}	35.8178	127	1817	8.58×10^{-5}	12.7609	69	622	6.21×10^{-5}
		x_5	9.9061	41	394	8.66×10^{-5}	13.4005	47	669	7.35×10^{-5}	5.7252	26	245	8.93×10^{-5}
		x_6	10.2025	41	394	8.66×10^{-5}	11.6065	44	623	8.75×10^{-5}	5.4912	27	255	4.76×10^{-5}
	1000000	x_1	41.9331	98	946	9.61×10^{-5}	64.8340	128	1775	9.35×10^{-5}	20.9197	54	498	7.66×10^{-5}
		x_2	50.1543	113	1081	9.85×10^{-5}	38.2670	79	1022	9.06×10^{-5}	22.9477	64	587	9.00×10^{-5}
		x_3	44.3355	107	1006	8.38×10^{-5}	F	F	F	F	30.1238	80	762	9.65×10^{-5}
		x_4	41.8083	101	940	9.97×10^{-5}	64.4284	135	1839	8.57×10^{-5}	19.0321	51	465	9.58×10^{-5}
		x_5	17.8933	43	413	9.12×10^{-5}	19.9369	40	566	6.89×10^{-5}	10.7797	26	245	8.30×10^{-5}
		x_6	18.3613	43	413	9.13×10^{-5}	17.5501	35	504	7.07×10^{-5}	9.8593	25	235	8.16×10^{-5}

Table A3. Numerical results of the 7 test problems in [15] with random initial guesses.

P	dim	x0	NDDF				DFMPRP				MSBFGS2			
			CPU	**Ni**	**Nf**	**Norm (F)**	**CPU**	**Ni**	**Nf**	**Norm (F)**	**CPU**	**Ni**	**Nf**	**Norm (F)**
P1	10000	x_7	0.1092	11	43	5.71×10^{-6}	F	F	F	F	0.0624	6	23	4.08×10^{-5}
		x_8	0.0780	9	35	6.98×10^{-8}	0.1092	11	78	3.52×10^{-5}	0.0000	5	19	9.40×10^{-7}
	100000	x_7	0.2184	11	43	2.04×10^{-5}	F	F	F	F	0.1560	7	27	8.28×10^{-9}
		x_8	0.0936	9	35	2.11×10^{-7}	0.3900	12	85	2.81×10^{-5}	0.0624	5	19	2.85×10^{-6}
	500000	x_7	0.9204	11	43	4.53×10^{-5}	F	F	F	F	0.3744	7	27	1.87×10^{-8}
		x_8	0.6240	9	35	4.80×10^{-7}	1.7784	12	85	6.20×10^{-5}	0.3744	5	19	6.47×10^{-6}
	1000000	x_7	1.8720	11	43	6.28×10^{-5}	F	F	F	F	1.0764	7	27	2.74×10^{-8}
		x_8	1.3728	9	35	6.73×10^{-7}	3.9000	12	85	8.82×10^{-5}	0.7956	5	19	9.12×10^{-6}
P2	10000	x_7	0.0624	5	19	2.41×10^{-7}	0.0624	11	76	7.03×10^{-5}	0.0780	4	15	6.89×10^{-10}
		x_8	0.0000	5	19	2.33×10^{-7}	0.0780	11	76	6.97×10^{-5}	0.0000	4	15	8.04×10^{-10}
	100000	x_7	0.1560	5	19	7.46×10^{-7}	0.4680	12	83	5.53×10^{-5}	0.0624	4	15	2.41×10^{-9}
		x_8	0.1404	5	19	7.43×10^{-7}	0.4368	12	83	5.51×10^{-5}	0.0936	4	15	2.47×10^{-9}
	500000	x_7	0.3120	5	19	1.66×10^{-6}	1.8876	13	90	3.09×10^{-5}	0.4056	4	15	5.52×10^{-9}
		x_8	0.2652	5	19	1.67×10^{-6}	1.9188	13	90	3.09×10^{-5}	0.2808	4	15	5.52×10^{-9}
	1000000	x_7	0.7176	5	19	2.35×10^{-6}	3.3696	13	90	4.36×10^{-5}	0.4836	4	15	7.86×10^{-9}
		x_8	0.7020	5	19	2.37×10^{-6}	3.2604	13	90	4.37×10^{-5}	0.4368	4	15	7.81×10^{-9}
P3	10000	x_7	F	F	F	F	F	F	F	F	F	F	F	F
		x_8	0.3900	84	598	7.73×10^{-5}	F	F	F	F	0.1872	44	305	9.94×10^{-5}
	100000	x_7	F	F	F	F	F	F	F	F	F	F	F	F
		x_8	1.6224	67	484	8.60×10^{-5}	F	F	F	F	1.5132	61	418	9.42×10^{-5}
	500000	x_7	F	F	F	F	F	F	F	F	F	F	F	F
		x_8	8.8765	63	452	6.87×10^{-5}	F	F	F	F	5.5536	42	293	8.80×10^{-5}
	1000000	x_7	F	F	F	F	F	F	F	F	F	F	F	F
		x_8	25.7714	78	562	5.31×10^{-5}	F	F	F	F	13.4629	56	387	4.05×10^{-5}
P4	10000	x_7	0.7176	63	470	9.28×10^{-5}	0.4680	29	319	7.79×10^{-5}	0.2964	27	193	6.50×10^{-5}
		x_8	0.6396	45	325	7.25×10^{-5}	1.1856	70	715	9.12×10^{-5}	0.3120	31	210	7.03×10^{-5}
	100000	x_7	4.4148	102	753	4.81×10^{-5}	2.5584	34	374	9.94×10^{-5}	4.1652	102	713	3.86×10^{-5}
		x_8	2.2464	57	411	4.66×10^{-5}	11.1073	187	1632	9.72×10^{-5}	1.6068	37	273	7.04×10^{-5}
	500000	x_7	36.5198	167	1221	9.33×10^{-5}	9.2821	31	343	7.05×10^{-5}	11.1385	54	385	9.44×10^{-5}
		x_8	10.9669	51	369	9.70×10^{-5}	45.9579	188	1703	8.95×10^{-5}	19.6717	103	720	6.99×10^{-5}
	1000000	x_7	F	F	F	F	22.1677	35	386	3.90×10^{-5}	23.7590	55	391	7.98×10^{-5}
		x_8	38.5322	86	616	3.87×10^{-5}	F	F	F	F	90.5274	215	1527	1.00×10^{-4}

Table A3. *Cont.*

P	dim	x0	NDDF				DFMPRP				MSBFGS2			
			CPU	Ni	Nf	Norm (F)	CPU	Ni	Nf	Norm (F)	CPU	Ni	Nf	Norm (F)
P5	10000	x_7	0.2652	81	495	7.48×10^{-5}	0.3120	66	604	7.55×10^{-5}	0.2808	70	414	9.63×10^{-5}
		x_8	0.3276	83	512	6.64×10^{-5}	0.2808	65	595	8.63×10^{-5}	0.2808	82	487	8.19×10^{-5}
	100000	x_7	2.2152	113	695	7.98×10^{-5}	2.4492	86	762	8.45×10^{-5}	1.6224	78	468	9.51×10^{-5}
		x_8	2.1216	104	638	6.55×10^{-5}	1.9188	67	621	8.75×10^{-5}	1.8876	90	530	9.22×10^{-5}
	500000	x_7	11.4817	94	577	8.91×10^{-5}	10.6237	69	633	8.47×10^{-5}	9.7033	88	521	9.80×10^{-5}
		x_8	10.5301	87	533	6.40×10^{-5}	12.3553	83	755	7.93×10^{-5}	9.4537	88	523	9.64×10^{-5}
	1000000	x_7	20.9041	92	564	8.90×10^{-5}	21.2161	75	696	8.82×10^{-5}	18.4861	90	531	9.86×10^{-5}
		x_8	25.8806	108	668	8.34×10^{-5}	22.9009	77	712	9.18×10^{-5}	17.9869	87	514	8.06×10^{-5}
P6	10000	x_7	1.3104	150	2015	7.23×10^{-5}	0.9672	122	1897	6.02×10^{-5}	0.2184	34	415	3.80×10^{-5}
		x_8	0.2340	36	459	5.85×10^{-5}	0.4368	36	599	8.73×10^{-5}	0.1716	34	430	3.98×10^{-5}
	100000	x_7	9.1105	184	2547	8.10×10^{-5}	11.7625	213	3364	7.34×10^{-5}	1.3728	35	422	9.91×10^{-5}
		x_8	3.9624	92	1151	8.71×10^{-5}	1.8564	33	551	5.52×10^{-5}	1.3104	27	335	4.85×10^{-5}
	500000	x_7	73.7105	255	3550	7.16×10^{-5}	F	F	F	F	8.9389	41	487	4.28×10^{-5}
		x_8	45.0219	153	2119	6.35×10^{-5}	10.2493	35	583	8.12×10^{-5}	8.0809	33	398	2.05×10^{-5}
	1000000	x_7	67.9384	107	1445	8.83×10^{-5}	F	F	F	F	16.4737	37	449	6.45×10^{-5}
		x_8	46.0047	82	1094	7.63×10^{-5}	21.6061	42	695	8.61×10^{-5}	13.9309	34	407	2.22×10^{-5}
P7	10000	x_7	0.5460	107	1006	9.88×10^{-5}	1.8408	189	2772	9.46×10^{-5}	0.4368	70	647	9.52×10^{-5}
		x_8	0.5772	86	837	9.93×10^{-5}	1.4664	173	2473	8.88×10^{-5}	0.4056	80	749	6.61×10^{-5}
	100000	x_7	2.4024	72	681	8.18×10^{-5}	17.5813	324	4944	7.03×10^{-5}	1.9344	59	543	7.70×10^{-5}
		x_8	3.9468	109	1114	6.27×10^{-5}	15.0385	263	4219	8.64×10^{-5}	3.3384	90	863	9.70×10^{-5}
	500000	x_7	16.7857	87	822	9.82×10^{-5}	F	F	F	F	8.2369	48	450	9.59×10^{-5}
		x_8	37.4714	162	1775	8.86×10^{-5}	F	F	F	F	14.1025	73	712	9.46×10^{-5}
	1000000	x_7	41.8083	108	1032	8.80×10^{-5}	F	F	F	F	25.8650	80	746	7.03×10^{-5}
		x_8	66.1444	134	1487	9.60×10^{-5}	F	F	F	F	29.2970	84	827	4.95×10^{-5}

References

1. Ortega, J.M.; Rheinboldt, W.C. *Iterative Solution of Nonlinear Equations in Several Variables*; Siam: Philadelphia, PA, USA, 1970.
2. Sun, W.; Yuan, Y.X. *Optimization Theory and Methods*; Springer: Berlin/Heidelberg, Germany, 2006.
3. Li, D.; Fukushima, M. A globally and superlinearly convergent Gauss-Newton based BFGS method for symmetric equations. *SIAM J. Numer. Anal.* **1999**, doi:10.1137/S0036142998335704. [CrossRef]
4. Gu, G.Z.; Li, D.H.; Qi, L.; Zhou, S.Z. Descent directions of quasi-Newton methods for symmetric nonlinear equations. *SIAM J. Numer. Anal.* **2002**, *40*, 1763–1774. [CrossRef]
5. Li, D.H.; Fukushima, M. A modified BFGS method and its global convergence in nonconvex minimization. *J. Comput. Appl. Math.* **2001**, *129*, 15–35. [CrossRef]
6. Zhou, W. A modified BFGS type quasi-Newton method with line search for symmetric nonlinear equations problems. *J. Comput. Appl. Math.* **2020**, *367*, 112454. [CrossRef]
7. Wang, J.; Zhu, D. The inexact-newton via GMRES subspace method without line search technique for solving symmetric nonlinear equations. *Appl. Numer. Math.* **2016**, *110*, 174–189. [CrossRef]
8. Yuan, G.L.; Yao, S.W. A BFGS algorithm for solving symmetric nonlinear equations. *Optimization* **2013**, *62*, 85–99. [CrossRef]
9. Wang, X.L.; Li, D.H. A modified Fletcher-Reeves-type derivative-free method for symmetric nonlinear equations. *NUmerical Algebr. Control Optim.* **2011**, *1*, 71–82.
10. Zhang, L.; Zhou, W.; Li, D. Global convergence of a modified Fletcher-Reeves conjugate gradient method with Armijo-type line search. *Numer. Math.* **2006**, *104*, 561–572. [CrossRef]
11. Dai, Y.H.; Kou, C.X. A nonlinear conjugate gradient algorithm with an optimal property and an improved Wolfe line search. *SIAM J. Optim.* **2013**, *23*, 296–320. [CrossRef]
12. Xiao, Y.; Wu, C.; Wu, S.Y. Norm descent conjugate gradient methods for solving symmetric nonlinear equations. *J. Glob. Optim.* **2015**, *62*, 751–762. [CrossRef]
13. Zhou, W.; Shen, D. Convergence properties of an iterative method for solving symmetric nonlinear equations. *J. Optim. Theory Appl.* **2015**, *164*, 277–289. [CrossRef]
14. Zhang, L.; Zhou, W.; Li, D.H. A descent modified Polak-Ribière-Polyak conjugate gradient method and its global convergence. *IMA J. Numer. Anal.* **2006**, *26*, 629–640. [CrossRef]
15. Liu, J.K.; Feng, Y.M. A norm descent derivative-free algorithm for solving large-scale nonlinear symmetric equations. *J. Comput. Appl. Math.* **2018**, *344*, 89–99. [CrossRef]
16. Liu, J.K.; Li, S.J. New three-term conjugate gradient method with guaranteed global convergence. *Int. J. Comput. Math.* **2014**, 91, 1744–1754. [CrossRef]
17. Cheng, W.Y.; Chen, Z.X. Nonmonotone spectral method for large-scale symmetric nonlinear equations. *Numer. Algorithms* **2013**, *62*, 149–162. [CrossRef]
18. Yusuf, W.M.; Sabiu, J. A derivative-free conjugate gradient method and its global convergence for solving symmetric nonlinear equations. *Int. J. Math. Math. Sci.* **2015**, *2015*, 1–8.
19. Zhou, W.J.; Chen, X.L. On the convergence of a derivative-free HS type method for symmetric nonlinear equations. *Adv. Model.Optim.* **2012**, *3*, 645–654.
20. Yakubu, U.A.; Mamat, M. A recent modification on dai-liao conjugate gradient method for solving symmetric nonlinear equations. *Far East J. Math. Sci.* **2018**, *103*, 1961–1974. [CrossRef]
21. Lv, J.; Deng, S.; Wan, Z. An efficient single-parameter scaling memoryless Broyden-Fletcher-Goldfarb-Shanno algorithm for solving large scale unconstrained optimization problems. *IEEE Access* **2020**, *8*, 85664–85674. [CrossRef]
22. Zhou, W.J.; Zhang, L. A nonlinear conjugate gradient method based on the MBFGS secant condition. *Optim. Methods Softw.* **2006**, *21*, 707–714. [CrossRef]
23. Byrd, R.H.; Nocedal, J. A tool for the analysis of quasi-Newton methods with application to unconstrained minimization. *SIAM J. Numer. Anal.* **1989**, *26*, 727–739. [CrossRef]
24. Dennis, J.E. A characterization of superlinear convergence and its application to quasi-Newton methods. *Math. Comput.* **1974**, *28*, 549–560. [CrossRef]
25. Li, Q.; Li, D.H. A class of derivative-free methods for large-scale nonlinear monotone equations. *IMA J. Numer. Anal.* **2011**, *31*, 1625–1635. [CrossRef]
26. Raydan, M. The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem. *SIAM J. Optim.* **1997**, *7*, 26–33. [CrossRef]
27. Kelley, C.T. Iterative methods for linear and nonlinear equations. *Front. Appl. Math.* **1995**, *16*, 206–207.
28. Yamakawa, E.; Fukushima, M. Testing parallel variable transformation. *Comput. Optim. Appl.* **1999**, *13*, 253–274. [CrossRef]
29. Dolan, E.D.; Moré, J.J. Benchmarking optimization software with performance profiles. *Math. Program.* **2002**, *91*, 201–213. [CrossRef]