An Inexact Optimal Hybrid Conjugate Gradient Method for Solving Symmetric Nonlinear Equations

Jamilu Sabi'u, Kanikar Muangchoo, Abdullah Shah, Auwal Bala Abubakar and Kazeem Olalekan Aremu

Abstract: This article presents an inexact optimal hybrid conjugate gradient (CG) method for solving symmetric nonlinear systems. The method is a convex combination of the optimal Dai–Liao (DL) and the extended three-term Polak–Ribiere–Polak (PRP) CG methods. However, two different formulas for selecting the convex parameter are derived by using the conjugacy condition and also by combining the proposed direction with the default Newton direction. The proposed method is again derivative-free, therefore the Jacobian information is not required throughout the iteration process. Furthermore, the global convergence of the proposed method is shown using some appropriate assumptions. Finally, the numerical performance of the method is demonstrated by solving some examples of symmetric nonlinear problems and comparing them with some existing symmetric nonlinear equations CG solvers.

Keywords: approximate gradient; convex combination; symmetric system; hybrid CG

MSC: 90C30; 90C26

1. Introduction

Consider the symmetric nonlinear system

\[ F(x) = 0, \quad x \in \mathbb{R}^n, \]  

where \( F : \mathbb{R}^n \to \mathbb{R}^n \) is a continuously differentiable mapping. The symmetry of the function \( F(x) \) means that the Jacobian of \( F(x) \) is symmetric. Such a class of problems could be defined from the gradient mapping of an unconstrained optimization problem, the Karush–Kuhn–Tucker (KKT) of equality constrained optimization problem, the discretized two-point boundary value problem, the saddle point problem, the discretized elliptic boundary value problem, and so on, see in [1–3]. The Newton method and its modifications are among the common methods for solving (1) despite their drawbacks [4–6]. Among the main drawbacks is that if the Jacobian of \( F(x) \) is singular, the Newton method may not be useful for finding the solution of (1). Nonetheless, the solving of linear equation or the storage of Jacobian inverse at each iteration are also known to be among the drawbacks of both Newton and quasi-Newton methods.
The conjugate gradient method is known to be a remedy for all matrix requiring iterative methods for finding the solution of (1), see in [7]. The method performs the iteration

\[ x_0 \in \mathbb{R}^n, \quad x_k = x_{k-1} + \gamma_k d_{k-1}, \quad k = 1, 2, \ldots, \] (2)

where \( \gamma_k \) is the step size to be determined via any suitable line search and \( d_k \) is the CG direction defined by

\[ d_0 = -F(x_0), \quad d_k = -F(x_k) + \beta_k d_{k-1}, \quad k \geq 1. \] (3)

The term \( \beta_k \) in (3) is a scalar known as the CG parameter. This scalar is what distinguished the various CG methods [8–12]. However, the CG parameter that was proposed by Dai–Liao (DL) is considered to be one of the most efficient CG parameters whenever the non-negative constant is appropriately selected [13]. The DL CG parameter is given by

\[ \beta^\text{DL}_k = \frac{F^T(y_{k-1} - ts_{k-1})}{d_{k-1}^T y_{k-1}}, \] (4)

with \( t \geq 0, s_{k-1} = x_k - x_{k-1}, y_{k-1} = F_k - F_{k-1}, \) and \( F_k = F(x_k) \). Moreover, in an attempt to ensure the appropriate and optimal selection of \( t \), Babaie-Kafaki and Ghanbari [14] proposed the following choices:

\[ t^1_k = \frac{\|y_{k-1}\|}{\|s_{k-1}\|}, \] (5)

and

\[ t^2_k = \frac{y_{k-1}^T s_{k-1}}{\|s_{k-1}\|^2} + \frac{\|y_{k-1}\|}{\|s_{k-1}\|}. \] (6)

Notwithstanding, the CG parameter proposed by Polak, Ribíre, and Polyak (PRP) is also numerically efficient and possess a restart feature that avoids jamming [8,9]. The PRP parameter is given by

\[ \beta^\text{PRP}_k = \frac{F^T y_{k-1}}{\|F_{k-1}\|^2}. \] (7)

The following descent modification of the PRP parameter has been proposed by Babaie-Kafaki and Ghanbari [10] based on the Dai–Liao approach [13] as

\[ \beta^\text{DPRP}_k = \frac{F^T y_{k-1}}{\|F_{k-1}\|^2} + \zeta \frac{F_k^T d_{k-1}}{\|F_{k-1}\|^2}, \] (8)

where \( \zeta \) is a real constant. Based on comprehensive singular value analysis, the authors suggested

\[ \zeta^*_k = \frac{d_{k-1}^T y_{k-1}}{\|d_{k-1}\|^2} \] (9)

to be the optimal choice of the real constant \( \zeta \), see in [15]. Moreover, for the symmetric nonlinear equations, Li and Wang [7] proposed an effective algorithm for solving symmetric nonlinear equations by combining the modified Fletcher–Reeves CG method [16] with an inexact gradient relation introduced in [1]. The reported numerical experiments illustrated that their algorithm is promising in handling large-scale symmetric nonlinear equations. This result in further studies on conjugate gradient methods for solving symmetric nonlinear equations were inspired. Zhou and Chen used the modified three-term CG method [17] with the approximate gradient and suggested HS-type CG method for solving symmetric
nonlinear equations [18]. Thereafter, some efficient CG methods for unconstrained optimization problems were incorporated with the approximate gradient relation to solve large scale symmetric nonlinear equations. For example, the inexact PRP method [5], the norm descent CG method [19], the derived CG algorithm for symmetric nonlinear systems [20], and so on, see in [12,21–23] for more details. In this article, motivated by the efficiencies of both the optimal DL CG method [14] and the optimal PRP CG method [15], we will propose an inexact optimal hybrid method for solving symmetric nonlinear equations. The proposed method is derivative-free and matrix-free, and therefore could sufficiently handle large-scale symmetric nonlinear systems efficiently.

The remaining parts of this paper are as follows. The next section is the derivation and details of the proposed method, followed by a convergence analysis, numerical experiment, and conclusion.

2. A Class of Optimal Hybrid CG Method

This section describes the proposed optimal hybrid CG method for solving symmetric nonlinear equations with two different choices for the selection of the convex parameter at every iteration. Recall that Li and Fukushima [1] suggested the following approximation:

$$h_k = \frac{F(x_k + \gamma_k F_k) - F_k}{\gamma_k}, \quad (10)$$

for the gradient function $\nabla f(x)$, where $\gamma_k$ is an arbitrary scalar and the function $f(x)$ is specified by

$$f(x) := \frac{1}{2} \| F(x) \|^2. \quad (11)$$

Now, we proposed the optimal hybrid CG parameter as

$$\beta^H_k = (1 - \omega_k) \beta^{DL *}_k + \omega_k \beta^{PRP *}_k, \quad \omega_k \in [0, 1], \quad (12)$$

where

$$\beta^{DL *}_k = \frac{h_k^T (y_{k-1} - t_k^s s_{k-1})}{d_{k-1}^T y_{k-1}}, \quad (13)$$

$$\beta^{PRP *}_k = \frac{h_k^T y_{k-1}}{\| h_{k-1} \|^2} + \zeta_k \frac{h_k^T d_{k-1}}{\| h_{k-1} \|^2}, \quad (14)$$

$y_{k-1} = h_k - h_{k-1}, s_{k-1} = x_k - x_{k-1}, t_k^s = t_1^1$ or $t_2^1$, and $\gamma_k$ to be determined such that

$$f(x_k + \gamma_k d_k) - f(x_k) \leq -\tau_1 \| \gamma_k F(x_k) \|^2 - \tau_2 \| \gamma_k d_k \|^2 + \xi_k f(x_k), \quad (15)$$

with $\tau_1, \tau_2 \in (0, 1), \xi_k$ is a non-negative sequence defined by

$$\sum_{k=0}^{\infty} \xi_k < \infty. \quad (16)$$

Moreover, we defined our optimal hybrid direction as

$$d_k = -h_k, \quad k = 0 \quad \text{and} \quad d_k = -h_k + \beta^H_k d_{k-1} \quad k \geq 1. \quad (17)$$

However, an optimal selection of $\omega_k \in [0, 1]$ in (12) would yield an optimal $\beta^H_k$. Therefore, in order to have an optimal choice of $\omega_k \in [0, 1]$ at every iteration, we proceed as follows.
2.1. The First Choice

The newton method is known to contain the full information of the Jacobian matrix; therefore, the first hybridization parameter will be derived by combining the proposed direction and the Newton direction. Recall that the default Newton direction is defined by

\[ d_k = -J_k^{-1}F_k, \]  

(18)

where \( J_k^{-1} \) is the Jacobian inverse. Using the approximate gradient (10), the Newton direction can be rewritten as

\[ d_k = -J_k^{-1}h_k, \]  

(19)

Combining (19) with (17), we get

\[ -J_k^{-1}h_k = -h_k + \beta_k^H d_{k-1}. \]  

(20)

Multiply (20) by \(-s_k^T h_k\) to obtain

\[ s_k^T h_k = s_k^T h_k - \beta_k^H s_k^T h_k d_{k-1}. \]  

(21)

It can be observed that (21) has a Jacobian matrix; therefore, to eliminate the Jacobian matrix we employ the following secant equation:

\[ y_{k-1}^T h_k = J_k s_{k-1}. \]  

(22)

Now, using (22) in (21) together with the symmetry of the Jacobian matrix, we have

\[ s_k^T h_k = y_k^T h_k - \beta_k^H y_k^T d_{k-1}. \]  

(23)

this yield

\[ s_k^T h_k = y_{k-1}^T h_k - \left\{ (1 - \omega_k) \beta_k^{DL} + \omega_k \beta_k^{PRP} \right\} y_k^T d_{k-1}. \]  

(24)

Now, solving for \( \omega_k \) in (28), we get

\[ \omega_k = \frac{s_k^T h_k - \beta_k^{DL} y_k^T d_{k-1} - \beta_k^{PRP} y_{k-1}^T d_{k-1}}{(\beta_k^{DL} - \beta_k^{PRP}) y_{k-1}^T d_{k-1}}. \]  

(25)

2.2. The Second Choice

The second choice of the hybridization parameter is obtained by utilizing the conjugacy condition. Recall that the conjugacy condition is given by

\[ y_k^T d_k = 0, \forall k. \]  

(26)

Now, using the direction (17) and Equation (26), we obtain that

\[ -y_{k-1}^T h_k + \beta_k^H y_{k-1}^T d_{k-1} = 0, \]  

(27)

and using the definition of \( \beta_k^H \) in (27), we get

\[ -y_{k-1}^T h_k + \beta_k^{DL} y_{k-1}^T d_{k-1} + \omega_k (\beta_k^{PRP} - \beta_k^{DL}) y_{k-1}^T d_{k-1} = 0. \]  

(28)

Therefore, solving for \( \omega_k \) in (28), we get the second formula for the computation of \( \omega_k \) at every iteration as follows:

\[ \omega_k^2 = \frac{y_{k-1}^T h_k - \beta_k^{DL} y_{k-1}^T d_{k-1}}{(\beta_k^{PRP} - \beta_k^{DL}) y_{k-1}^T d_{k-1}}. \]  

(29)

Furthermore, note that the proposed choices for \( \omega_k \) in (25) and (29) may be outside the interval \([0, 1]\). However, to maintain the convex combination in (12), we choose \( \omega_k \) to be
Symmetry 2021, 13, 1829

1 whenever $\omega_k > 1$, and $\omega_k = 0$ whenever $\omega_k < 0$. Some of the interesting features of the proposed method is that if $\omega_k = 1$, the proposed method reduces to PRP-type method [24] and for $\omega_k = 0$, we have an inexact Dai–Liao CG method for symmetric nonlinear equations which has not yet been presented in the literature. Below is the proposed Algorithm 1:

Algorithm 1 Optimal hybrid CG method (OHCG).

step 0: Select $x_0 \in \mathbb{R}^n$, and initialize the constants $s \in (0, 1)$, $\epsilon, \tau_1, \tau_2 \geq 0$. Set $k = 0$ and choose the positive sequence $\{\xi_k\}$.

step 1: Whenever $\|F_k\| \leq \epsilon$, stop, if not go to Step 2.

step 2: Calculate the search direction via (17) using any of the optimal choice (25) or (29).

step 3: Determine $\gamma_k = \max\{1, s, s^2 \cdots\}$, satisfying

$$f(x_k + \gamma_k d_k) - f(x_k) \leq -\tau_1 \|\gamma_k F(x_k)\|^2 - \tau_2 \|\gamma_k d_k\|^2 + \xi_k f(x_k).$$

(30)

step 4: Compute $x_k$ using (2).

step 5: Set $k = k + 1$ and go to Step 1.

3. Global Convergence

This section will prove the global convergence of Algorithm 1 using the following assumptions: Start by defining the level set

$$\chi = \{x| f(x) \leq \exp^\xi f(x_0)\},$$

(31)

such that $\xi$ fulfills the condition (16).

Assumption 1.

1. The level set (31) is bounded.
2. In a neighborhood $W$ of $\chi$, the Jacobian of $h(x)$ is bounded and symmetric positive definite, i.e., there exists some $q \geq q_2 > 0$ such that

$$\|\nabla f(x)\| \leq q, \quad \forall x \in W,$$

(32)

and

$$q_2 \|g\|^2 \geq g^T \nabla f(x) g, \quad \forall x \in W, g \in \mathbb{R}^n.$$  

(33)

The Assumption 1 above implies that there exists $R_1, R_2, q_1 > 0$ such that

$$\|f(x)\| \leq R_1, \quad \|F(x)\| \leq R_2, \quad \forall x \in W,$$

(34)

and

$$\|\nabla f(x) - \nabla f(z)\| \leq q_1 \|x - z\|, \quad \forall x, z \in W.$$  

(35)

Lemma 1. Let $\{x_k\}$ be generated by Algorithm 1 (OHCG). Then, $\{\|F_k\|\}$ converges and $\{x_k\} \in \chi$.

Proof. From the step length (30) and using the definition (11), it is not difficult to see that

$$\|F_{k+1}\| \leq (1 + \xi_k)\|F_k\|.$$  

(36)
As the sequence $\{\xi_k\}$ is bounded, then we apply the result of the Lemma 3.3 in [25] and hence $\{\|F_k\|\}$ converges. Again, from the result

\[
\|F_{k+1}\| \leq (1 + \xi_k)^{1/2} \|F_k\|
\]

\[
\leq \prod_{i=0}^{k} (1 + \xi_k)^{1/2} \|F_0\|
\]

\[
\leq \|F_0\| \left[ \frac{1}{k+1} \sum_{i=0}^{k} (1 + \xi_i) \right]^{k+1/2}
\]

\[
\leq \|F_0\| \left[ 1 + \frac{1}{k+1} \sum_{i=0}^{k} \xi_i \right]^{k+1/2}
\]

\[
\leq \|F_0\| \left( 1 + \frac{\xi}{k+1} \right)^{k+1}
\]

\[
\leq e^{\xi} \|F_0\|.
\]

This implies $\{x_k\} \in \chi$. \qed

**Lemma 2.** Suppose our assumption hold. Then,

\[
\lim_{k \to \infty} \|\gamma_k d_k\| = \lim_{k \to \infty} ||s_k|| = 0, \quad \lim_{k \to \infty} \|\gamma_k F_k\| = 0.
\] (38)

**Proof.** The proof is directly follow from (16) and (30). \qed

**Theorem 1.** Algorithm 1 (OHCG) converges globally whenever our assumption hold, i.e.,

\[
\lim \inf_{k \to \infty} \|\nabla f(x_k)\| = 0.
\] (39)

**Proof.** Suppose (39) is not true, there exists $\alpha > 0$ such that

\[
\|\nabla f(x_k)\| \geq \alpha, \quad \forall k \geq 0.
\] (40)

As $\nabla f(x_k) = J^T_k F_k$, we can deduce from (40) that there exists $\alpha_1 > 0$ satisfying

\[
\|F_k\| \geq \alpha_1, \quad \forall k \geq 0.
\] (41)

CASE I: $\lim \sup_{k \to \infty} \gamma_k > 0$, then (38) implies that $\lim \sup_{k \to \infty} \|F_k\| = 0$. This fact and that of Lemma 1 implies that $\lim_{k \to \infty} \|F_k\| = 0$. Therefore, this contradicts with (41).

CASE II: $\lim \sup_{k \to \infty} \gamma_k = 0$. Already $\gamma_k \geq 0$, this shows that

\[
\lim_{k \to \infty} \gamma_k = 0.
\] (42)

Let us consider the definition of $h_k$ in (10) and (34), then

\[
\|h_k\| = \left\| \int_0^1 f(x_k + l \gamma_{k-1} F_k) F_k dl \right\| \leq R_1 R_2, \quad \forall k \geq 0.
\] (43)

Now using (33), we have

\[
\|y_k\| = \left\| \int_0^1 \nabla h(x_k + l s_k) dl s_k \right\| \leq g \|s_k\|.
\] (44)
Furthermore, by the mean-value theorem, we have
\[ y_k^T s_k = s_k^T (h_k - h_{k-1}) = s_k^T \nabla h(\theta)s_k \geq q_2 \| s_k \|^2, \]  
where \( \theta = x_{k-1} + v(x_k - x_{k-1}), v \in (0, 1) \). Again, from the definition of \( \beta_k^{DL*} \) and \( \beta_k^{PRP*} \), we get
\[ \| d_k \| = -h_k + \beta_k^{Hs} d_{k-1} = -h_k + \left\{ (1 - \omega_k) \beta_k^{DL*} + \omega_k \beta_k^{PRP*} \right\} d_{k-1} = -h_k + \left\{ (1 - \omega_k) \frac{h_k^T (y_{k-1} - \frac{1}{2} s_{k-1})}{d_{k-1}^T y_{k-1}} + \omega_k \frac{h_k^T y_{k-1}}{\| h_{k-1} \|^2} \right\} d_{k-1} \]
\[ \leq \| h_k \| + \left\{ (1 - \omega_k) \frac{\| h_k \| (\| y_{k-1} \| + \| s_{k-1} \|)}{d_{k-1}^T y_{k-1}} + \omega_k \frac{\| h_k \| \| y_{k-1} \|}{\| h_{k-1} \|^2} \right\} \| d_{k-1} \| \]
\[ \leq R_1 R_2 + (1 - \omega_k) \frac{2 R_1 R_2 q_2}{\omega_2} + 2 \omega_k \frac{R_1 R_2 q_2}{\omega_2} \| s_{k-1} \| \| d_{k-1} \|. \]
where we used Cauchy–Schwarz inequality in the first equality and inequalities (43), (44), and (45) in the last inequality. Now from (38) there exist \( \tilde{\epsilon} \in (0, 1) \) such that \( \| s_{k-1} \| \leq \tilde{\epsilon} \), thus we have
\[ \| d_k \| \leq R_1 R_2 + M \| d_{k-1} \| \]
\[ \leq R_1 R_2 (1 + M^2 + \cdots + M^k) + M^k \| d_0 \| \]
\[ \leq \frac{R_1 R_2}{1 - M} + R_1 R_2 = \frac{R_1 R_2 (2 - M)}{1 - M}, \]
where \( M = (1 - \omega_k) \frac{2 R_1 R_2 q_2}{\omega_2} + 2 \omega_k \frac{R_1 R_2 q_2}{\omega_2} \). This shows that our direction is bounded. As \( \lim_{k \to \infty} \gamma_k = 0 \), then \( \gamma_k^* = \frac{2}{\delta} \) does not satisfy (30), that is to say,
\[ f(x_k + \gamma_k^* d_k) - f(x_k) > -\tau_1 \| \gamma_k^* F(x_k) \|^2 - \tau_2 \| \gamma_k^* d_k \|^2 + \xi_k f(x_k), \]
which means that
\[ \frac{f(x_k + \gamma_k^* h_k) - f(x_k)}{\gamma_k} > -\tau_1 \gamma_k^* \| F(x_k) \|^2 - \tau_2 \gamma_k^* \| d_k \|^2. \]
By mean value theorem, \( \delta \in (0, 1) \) exists in such a way that
\[ \frac{f(x_k + \gamma_k^* d_k) - f(x_k)}{\gamma_k} = \nabla f(x_k + \delta \gamma_k^* d_k)^T d_k. \]
As \( \chi \) is bounded and \( x \in \chi \), then we assume that \( x_k \to x^* \) and have the following result using (3) and (10):
\[ \lim_{k \to \infty} d_k = -\lim_{k \to \infty} h_k + \lim_{k \to \infty} \beta_k^{Hs} d_{k-1} = -\nabla f(x^*). \]
On the other hand,
\[ \lim_{k \to \infty} \nabla f(x_k + \delta \gamma_k^* d_k) = \nabla f(x^*). \]
Therefore, using (48)–(51), we get \(- \nabla f(x^*)^T \nabla f(x^*) \geq 0 \) which indicate that \( \| \nabla f(x^*) \| = 0 \). This yields a contradiction and thus the proof is complete.

4. Numerical Experiment
This section will provide the numerical experiments comparison between the proposed algorithm and some efficient CG solvers for solving symmetric nonlinear equations. However, for the numerical experiment of the proposed algorithm, we chose the optimal
choice $t_k^i$ defined by (5) for the optimal DL method and $\xi_k^i$ defined by (9) for the optimal PRP methods. The algorithms were written in Matlab R2014a and ran on a 1.6 GHz CPU processor with 8GB RAM memory. To justify the numerical comparison, the parameters for the NDAS method [22] and the ICGM method [21] were set the same as in the reference papers. However, we set $s = 0.3, \tau_1 = \tau_2 = 0.0001, \gamma_{k-1} = 0.01,$ and $\epsilon = 10^{-4}$ for our proposed methods. In our experiment, we considered the following problems:

**Problem 1.** [26] The precise description of the $F(x)$ function is described as $F(x_i) = \exp(x_i) - 1$, for $i = 1, 2, 3, \cdots, n$.

**Problem 2.** [27] The precise description of the $F(x)$ function is described as $F_1(x) = x_1 - x_2^2 - 1, \quad F_2(x) = x_1 - x_2^3 - x_3^2 - 1, \quad F_3(x) = x_1 - x_2^4 - x_3^3 - x_4^2 - 1, \quad F_4(x) = x_1 - x_2^5 - x_3^4 - x_4^3 - x_5^2 - 1, \quad F_5(x) = x_1 - x_2^6 - x_3^5 - x_4^4 - x_5^3 - x_6^2 - 1, \quad F_6(x) = x_1 - x_2^7 - x_3^6 - x_4^5 - x_5^4 - x_6^3 - x_7^2 - 1, \quad F_7(x) = x_1 - x_2^8 - x_3^7 - x_4^6 - x_5^5 - x_6^4 - x_7^3 - x_8^2 - 1, \quad F_8(x) = x_1 - x_2^9 - x_3^8 - x_4^7 - x_5^6 - x_6^5 - x_7^4 - x_8^3 - x_9^2 - 1, \quad F_9(x) = x_1 - x_2^{10} - x_3^9 - x_4^8 - x_5^7 - x_6^6 - x_7^5 - x_8^4 - x_9^3 - x_{10}^2 - 1$.

**Problem 3.** [5] The precise description of the $F(x)$ function is described as $F_1(x) = \frac{1}{10}(1 - x_1^2 - e^{-x_1^2}), \quad F_2(x) = \frac{1}{10}(1 - e^{-x_2^2})$.

**Problem 4.** [28] The precise description of the $F(x)$ function is described as $F_1(x) = 3x_1^2 + 2x_2 - 5\sin(x_1 - x_2) - \sin(x_1 + x_2), \quad F_2(x) = x_{i-1} \exp(x_{i-1} - x_i) + x_i(4 + 3x_1^2) + \sin(x_i - x_{i-1}) \sin(x_i + x_{i-1}), \quad i = 2, 3, \cdots, n - 1, \quad F_3(x) = -x_{i-1} \exp(x_{i-1} - x_i) + 4x_i - 3$.

**Problem 5.** [27] The precise description of the $F(x)$ function is described as $F_1(x) = \frac{1}{10}(1 - x_1^2 - e^{-x_1^2})$, $F_2(x) = \frac{1}{10}(1 - e^{-x_2^2})$.

**Problem 6.** [29] The precise description of the $F(x)$ function is described as $F_1(x) = \log(|x_i| + 1) - \frac{2}{n}, \quad i = 1, 2, 3, \cdots, n$.

**Problem 7.** [29] The precise description of the $F(x)$ function is described as $F_1(x) = 2x_i - \sin|x_i|$, for $i = 1, 2, 3, \cdots, n$.

**Problem 8.** [29] The precise description of the $F(x)$ function is described as $F_1(x) = \exp(\cos(hx_1 + x_2))$, $F_2(x) = \exp(\cos(hx_{i-1} + x_i + x_{i+1}))$, $i = 2, 3, \cdots, n - 1, \quad F_3(x) = \exp(\cos(hx_{n-1} + x_n))$.

**Problem 9.** [29] The precise description of the $F(x)$ function is described as $F_1(x) = x_i - \sin|x_i - 1|$, for $i = 1, 2, 3, \cdots, n$.

The numerical efficacy of our algorithm in terms of number of iterations and the CPU time in second is shown in Tables 1–5. The term “ITR” reflects the number of iterations, “TIME” for the CPU time, and “NORM” for the norm value of the function at the stopping point. Iteration ends whenever $\|F_k\| \leq 10^{-4}$ or the number of iterations reaches 1000. During the entire experiment, the following initial points were used, namely, $x_1 = (0.1, 0.1, \cdots, 0.1)$, $x_2 = (0.2, 0.2, \cdots, 0.2)$, $x_3 = (0.3, 0.3, \cdots, 0.3)$, $x_4 = (1/2, 1/2, \cdots, 1/2)$, $x_5 = (1 - 1/2, 1 - 1/2, \cdots, 1 - 1/2)$, $x_6 = (-0.1, -0.1, \cdots, -0.1)$, $x_7 = 0.1(1, 1/2, 1/2, \cdots, 1/2)$, and $x_8 = -(1 - 1/2, 1 - 1/2, \cdots, 1 - 1/2)$. 

Table 1. Numerical efficiency of Algorithm 1 with the choice of $\beta_k^{H^*}(\omega^1_k)$ and $\beta_k^{H^*}(\omega^2_k)$ versus NDAS method [22] and ICGM method [21].

| DIM  | GUESS | ITR | TIME | NORM | ITR | TIME | NORM | ITR | TIME | NORM | ITR | TIME | NORM |
|------|-------|-----|------|------|-----|------|------|-----|------|------|-----|------|------|
| 50,000
| x1   | 4     | 0.051017 | 1.32 \times 10^{-10} | 5   | 0.711194 | 9.38 \times 10^{-11} | 1000 | 5.358113 | 0.000606 | 10 | 0.083581 | 7.98 \times 10^{-5} |
| x2   | 4     | 0.03615 | 1.27 \times 10^{-5} | 6   | 0.127425 | 6.69 \times 10^{-6} | 1000 | 5.769465 | 0.001268 | 12 | 0.08618 | 4.37 \times 10^{-5} |
| x3   | 5     | 0.064414 | 8.12 \times 10^{-7} | 7   | 0.136223 | 4.81 \times 10^{-7} | 1000 | 6.545712 | 0.001127 | 13 | 0.090501 | 5.89 \times 10^{-5} |
| x4   | 37    | 0.320713 | 2.01 \times 10^{-8} | 90  | 1.087917 | 2.48 \times 10^{-8} | 51   | 0.313294 | 6.54 \times 10^{-5} | 45 | 0.312344 | 9.66 \times 10^{-6} |
| x5   | 37    | 0.373899 | 1.33 \times 10^{-8} | 90  | 0.952427 | 2.05 \times 10^{-8} | 51   | 0.385819 | 6.47 \times 10^{-5} | 45 | 0.307783 | 1.04 \times 10^{-5} |
| x6   | 9     | 0.097553 | 6.76 \times 10^{-7} | 13  | 0.160268 | 7.64 \times 10^{-10} | 1000 | 7.526905 | 0.000645 | 18 | 0.117476 | 3.26 \times 10^{-5} |
| x7   | 3     | 0.035575 | 4.37 \times 10^{-5} | 4   | 0.1074 | 5.78 \times 10^{-6} | 1000 | 6.622959 | 0.002336 | 8  | 0.069154 | 6.78 \times 10^{-5} |
| x8   | 7     | 0.073464 | 1.92 \times 10^{-6} | 12  | 0.184662 | 3.25 \times 10^{-10} | 1000 | 6.60593  | 0.000566 | 10 | 0.072791 | 1.17 \times 10^{-5} |
| 100,000
| x1   | 4     | 0.08817 | 1.86 \times 10^{-10} | 5   | 0.216573 | 1.57 \times 10^{-10} | 1000 | 12.85136 | 0.000858 | 11 | 0.163569 | 3.39 \times 10^{-5} |
| x2   | 4     | 0.112781 | 1.8 \times 10^{-5} | 6   | 0.18873 | 1.31 \times 10^{-5} | 1000 | 12.02928 | 0.001794 | 12 | 0.161687 | 6.18 \times 10^{-5} |
| x3   | 5     | 0.133655 | 1.15 \times 10^{-6} | 8   | 0.241027 | 1.19 \times 10^{-7} | 1000 | 12.12703 | 0.001593 | 13 | 0.200714 | 8.33 \times 10^{-5} |
| x4   | 37    | 0.958361 | 2.56 \times 10^{-8} | 90  | 2.439265 | 1.41 \times 10^{-5} | 51   | 0.670645 | 9.21 \times 10^{-5} | 45 | 0.571358 | 1.39 \times 10^{-5} |
| x5   | 37    | 0.881123 | 2.08 \times 10^{-8} | 90  | 2.389912 | 1.31 \times 10^{-5} | 51   | 0.812228 | 9.16 \times 10^{-5} | 45 | 0.573077 | 1.44 \times 10^{-5} |
| x6   | 9     | 0.241214 | 9.56 \times 10^{-7} | 13  | 0.329319 | 2.73 \times 10^{-7} | 1000 | 13.23539 | 0.000913 | 18 | 0.230604 | 4.61 \times 10^{-5} |
| x7   | 3     | 0.080855 | 6.18 \times 10^{-5} | 4   | 0.118452 | 1.32 \times 10^{-5} | 1000 | 12.86233 | 0.003303 | 8  | 0.110558 | 9.59 \times 10^{-5} |
| x8   | 7     | 0.176245 | 2.71 \times 10^{-6} | 12  | 0.339665 | 1.73 \times 10^{-9} | 1000 | 12.07158 | 0.000801 | 10 | 0.160622 | 1.65 \times 10^{-5} |
| Problem 2 | $\beta^H_k(\omega_k^1)$ | $\beta^H_k(\omega_k^2)$ | NDAS | ICGM |
|----------|------------------|------------------|------|------|
|          | DIM   | GUESS | ITR | TIME | NORM | ITR | TIME | NORM | ITR | TIME | NORM | ITR | TIME | NORM |
|          | x₁    |       | 49  | 1.388361 | $9.42 \times 10^{-5}$ | 73  | 2.309985 | $6.85 \times 10^{-5}$ | 7   | 0.247055 | NaN | 537 | 10.94105 | $9.91 \times 10^{-5}$ |
| 50,000   | x₂    |       | 46  | 1.211725 | $9.52 \times 10^{-5}$ | 63  | 2.034998 | $9.98 \times 10^{-5}$ | 133 | 3.125014 | $9.48 \times 10^{-5}$ | 630 | 12.987 | $9.96 \times 10^{-5}$ |
|          | x₃    |       | 68  | 1.560972 | $8.81 \times 10^{-5}$ | 73  | 2.637633 | $9.47 \times 10^{-5}$ | 128 | 3.220816 | $9.87 \times 10^{-5}$ | 543 | 11.74005 | $9.94 \times 10^{-5}$ |
|          | x₄    |       | 129 | 2.932939 | $8.14 \times 10^{-5}$ | 134 | 3.832334 | $9.47 \times 10^{-5}$ | 246 | 5.627216 | $9.44 \times 10^{-5}$ | 373 | 7.827305 | $9.93 \times 10^{-5}$ |
|          | x₅    |       | 117 | 2.406448 | $6.01 \times 10^{-5}$ | 115 | 3.223662 | $8.55 \times 10^{-5}$ | 176 | 3.771749 | $3.71 \times 10^{-5}$ | 681 | 14.5844 | $9.99 \times 10^{-5}$ |
|          | x₆    |       | 72  | 1.532064 | $9.72 \times 10^{-5}$ | 81  | 2.325801 | $8.45 \times 10^{-5}$ | 138 | 3.11174 | $9.54 \times 10^{-5}$ | 620 | 12.9883 | $9.87 \times 10^{-5}$ |
|          | x₇    |       | 137 | 3.046806 | $5.47 \times 10^{-5}$ | 64  | 2.250733 | $9.24 \times 10^{-5}$ | 142 | 3.320312 | $9.39 \times 10^{-5}$ | 175 | 3.640726 | $9.1 \times 10^{-5}$ |
|          | x₈    |       | 153 | 3.159947 | $4.9 \times 10^{-5}$ | 129 | 4.421972 | $6.39 \times 10^{-5}$ | 51  | 1.257545 | $9.36 \times 10^{-5}$ | 490 | 10.21439 | $9.98 \times 10^{-5}$ |
| 100,000  | x₁    |       | 56  | 2.134486 | $9.31 \times 10^{-5}$ | 79  | 6.900653 | $9.86 \times 10^{-5}$ | 7   | 0.58915 | NaN | 573 | 22.36094 | $9.95 \times 10^{-5}$ |
|          | x₂    |       | 75  | 3.026543 | $7.88 \times 10^{-5}$ | 75  | 6.693699 | $9.03 \times 10^{-5}$ | 119 | 6.103506 | $9.78 \times 10^{-5}$ | 653 | 26.02026 | $9.99 \times 10^{-5}$ |
|          | x₃    |       | 68  | 2.695133 | $8.66 \times 10^{-5}$ | 72  | 6.2076 | $9.2 \times 10^{-5}$ | 96  | 4.540488 | $9.76 \times 10^{-5}$ | 376 | 14.5441 | $9.88 \times 10^{-5}$ |
|          | x₄    |       | 134 | 5.237932 | $8.17 \times 10^{-5}$ | 151 | 8.914932 | $8.62 \times 10^{-5}$ | 287 | 11.58659 | $9.12 \times 10^{-5}$ | 622 | 23.51092 | $9.93 \times 10^{-5}$ |
|          | x₅    |       | 127 | 4.708571 | $8.25 \times 10^{-5}$ | 119 | 6.758494 | $7.34 \times 10^{-5}$ | 179 | 7.320956 | $5.78 \times 10^{-5}$ | 645 | 24.98121 | $9.91 \times 10^{-5}$ |
|          | x₆    |       | 85  | 3.38715 | $9.14 \times 10^{-5}$ | 79  | 3.983438 | $9.18 \times 10^{-5}$ | 81  | 3.417891 | $9.45 \times 10^{-5}$ | 501 | 18.89237 | $9.95 \times 10^{-5}$ |
|          | x₇    |       | 170 | 6.837026 | $9.79 \times 10^{-5}$ | 71  | 3.73959 | $9.81 \times 10^{-5}$ | 128 | 5.434381 | $9.72 \times 10^{-5}$ | 505 | 19.19699 | $9.93 \times 10^{-5}$ |
|          | x₈    |       | 138 | 5.192739 | $7.29 \times 10^{-5}$ | 128 | 6.579588 | $9.15 \times 10^{-5}$ | 46  | 1.912222 | $6.6 \times 10^{-5}$ | 487 | 19.58842 | $9.86 \times 10^{-5}$ |
Table 2. Numerical efficiency of Algorithm 1 with the choice of $\beta^H_{k}^* (\omega_1^k)$ and $\beta^H_{k}^* (\omega_2^k)$ versus NDAS method [22] and ICGM method [21].

| Problem 3 | $\beta^H_{k}^* (\omega_1^k)$ | $\beta^H_{k}^* (\omega_2^k)$ | NDAS       | ICGM       |
|-----------|-----------------------------|-----------------------------|------------|------------|
|           | DIM | GUESS | ITR | TIME | NORM | ITR | TIME | NORM | ITR | TIME | NORM | ITR | TIME | NORM | ITR | TIME | NORM |
| x_1       | 50,000 | x_1 | 7 | 0.130459 | 9.01 × 10^{-6} | 7 | 0.326768 | 1.05 × 10^{-5} | 5 | 0.147254 | 9.59 × 10^{-5} | 6 | 0.161476 | 9.33 × 10^{-6} |
| x_2       | | x_2 | 3 | 0.097567 | 4.49 × 10^{-5} | 7 | 0.208194 | 5.53 × 10^{-5} | 3 | 0.100919 | 3.96 × 10^{-5} | 5 | 0.205028 | 1.46 × 10^{-5} |
| x_3       | | x_3 | 4 | 0.137949 | 3.04 × 10^{-5} | 7 | 0.269881 | 1.38 × 10^{-5} | 8 | 0.304496 | 1.99 × 10^{-5} | 10 | 0.297645 | 9.95 × 10^{-8} |
| x_4       | | x_4 | 5 | 0.156842 | 9.3 × 10^{-5} | 8 | 0.316907 | 2.48 × 10^{-5} | 9 | 0.36357 | 2.23 × 10^{-6} | 6 | 0.176784 | 6.58 × 10^{-6} |
| x_5       | | x_5 | 0 | 0.006896 | 0 | 0 | 0.01035 | 0 | 0 | 0.007833 | 0 | 0 | 0.006028 | 0 |
| x_6       | | x_6 | 7 | 0.271075 | 4.55 × 10^{-5} | 3 | 0.175715 | 1.79 × 10^{-5} | 3 | 0.202744 | NaN | 7 | 0.285133 | 4.01 × 10^{-6} |
| x_7       | | x_7 | 7 | 0.190334 | 9.01 × 10^{-6} | 7 | 0.296095 | 1.05 × 10^{-5} | 5 | 0.2407 | 9.59 × 10^{-5} | 6 | 0.157174 | 9.33 × 10^{-6} |
| x_8       | | x_8 | 0 | 0.007218 | 0 | 0 | 0.008271 | 0 | 0 | 0.008564 | 0 | 0 | 0.00537 | 0 |
| x_9       | 100,000 | x_1 | 10 | 0.74008 | 2.32 × 10^{-5} | 13 | 1.602305 | 1.73 × 10^{-6} | 2 | 0.396171 | NaN | 7 | 0.606307 | 3.21 × 10^{-5} |
| x_10      | | x_2 | 6 | 0.376712 | 4.15 × 10^{-8} | 7 | 0.739255 | 3.14 × 10^{-6} | 7 | 0.661026 | 2.1 × 10^{-5} | 4 | 0.336146 | 2.27 × 10^{-5} |
| x_11      | | x_3 | 8 | 0.49952 | 2.45 × 10^{-6} | 7 | 0.64952 | 3.11 × 10^{-5} | 8 | 0.730984 | 4.59 × 10^{-6} | 12 | 0.774873 | 8.18 × 10^{-5} |
| x_12      | | x_4 | 10 | 0.734979 | 2.85 × 10^{-6} | 13 | 1.327325 | 5.86 × 10^{-6} | 6 | 0.423914 | 2.68 × 10^{-8} | 6 | 0.390528 | 4.64 × 10^{-5} |
| x_13      | | x_5 | 0 | 0.015278 | 0 | 0 | 0.01714 | 0 | 0 | 0.017242 | 0 | 0 | 0.011946 | 0 |
| x_14      | | x_6 | 8 | 0.74008 | 2.32 × 10^{-5} | 13 | 1.602305 | 1.73 × 10^{-6} | 2 | 0.396171 | NaN | 7 | 0.606307 | 3.21 × 10^{-5} |
| x_15      | | x_7 | 4 | 0.267311 | 1.8 × 10^{-6} | 4 | 0.363102 | 3.16 × 10^{-7} | 7 | 0.609314 | 4.7 × 10^{-5} | 6 | 0.393849 | 1.72 × 10^{-5} |
| x_16      | | x_8 | 0 | 0.015626 | 0 | 0 | 0.017721 | 0 | 0 | 0.017815 | 0 | 0 | 0.015042 | 0 |
Table 2. Cont.

| Problem 4 | $\beta^H_k(\omega^1_k)$ | $\beta^H_k(\omega^2_k)$ | NDAS | ICGM |
|-----------|-------------------------|-------------------------|------|------|
| DIM       | GUESS | ITR  | TIME  | NORM | ITTR | TIME  | NORM | ITTR | TIME  | NORM | ITTR | TIME  | NORM |
| 50,000    |       |      |       |      |      |       |      |      |       |      |      |       |      |
| $x_1$     | 44    | 2.690291 | $8.41 \times 10^{-5}$ | 43  | 4.617571 | $9.04 \times 10^{-5}$ | 12  | 1.863923 | NaN  | 62  | 4.524161 | $8.63 \times 10^{-5}$ |
| $x_2$     | 45    | 2.663571 | $9.78 \times 10^{-5}$ | 44  | 4.521443 | $9.53 \times 10^{-5}$ | 5   | 3.054371 | NaN  | 89  | 7.671063 | $9.14 \times 10^{-5}$ |
| $x_3$     | 44    | 2.575053 | $8.38 \times 10^{-5}$ | 49  | 4.750722 | $7.64 \times 10^{-5}$ | 3   | 0.230201 | NaN  | 6   | 1.01848 | NaN  |
| $x_4$     | 31    | 1.944946 | $6.56 \times 10^{-5}$ | 36  | 2.844192 | $9.48 \times 10^{-5}$ | 75  | 5.669829 | $6.84 \times 10^{-5}$ | 43  | 3.052905 | $8.39 \times 10^{-5}$ |
| $x_5$     | 46    | 2.77079 | $7.82 \times 10^{-5}$ | 40  | 3.092822 | $7.46 \times 10^{-5}$ | 102 | 6.978015 | $8.05 \times 10^{-5}$ | 47  | 3.349462 | $8.76 \times 10^{-5}$ |
| $x_6$     | 53    | 2.979479 | $8.7 \times 10^{-5}$ | 56  | 4.566533 | $9.46 \times 10^{-5}$ | 4   | 0.600135 | NaN  | 1000 | 178.8212 | 46.85852 |
| $x_7$     | 41    | 2.302363 | $8.68 \times 10^{-5}$ | 46  | 3.301815 | $8.57 \times 10^{-5}$ | 5   | 0.549722 | NaN  | 61  | 4.607346 | $8.1 \times 10^{-5}$ |
| $x_8$     | 47    | 2.698441 | $9.37 \times 10^{-5}$ | 203 | 12.16855 | $7.1 \times 10^{-5}$ | 4   | 0.327506 | NaN  | 1000 | 128.2877 | 3.78474 |
| 100,000   |       |      |       |      |      |       |      |      |       |      |      |       |      |
| $x_1$     | 41    | 4.972517 | $7.93 \times 10^{-5}$ | 49  | 5.813092 | $7.83 \times 10^{-5}$ | 8   | 2.774162 | NaN  | 63  | 10.42985 | $7.91 \times 10^{-5}$ |
| $x_2$     | 45    | 5.210847 | $7.98 \times 10^{-5}$ | 40  | 5.038291 | $4.01 \times 10^{-5}$ | 5   | 5.596234 | NaN  | 70  | 11.08061 | $8.1 \times 10^{-5}$ |
| $x_3$     | 45    | 4.763892 | $8.99 \times 10^{-5}$ | 50  | 6.623488 | $9.66 \times 10^{-5}$ | 3   | 0.353785 | NaN  | 6   | 2.195849 | NaN  |
| $x_4$     | 32    | 3.455663 | $7.19 \times 10^{-5}$ | 37  | 4.694169 | $9.81 \times 10^{-5}$ | 50  | 6.98836 | $5.91 \times 10^{-5}$ | 40  | 6.260133 | $8.2 \times 10^{-5}$ |
| $x_5$     | 48    | 5.294363 | $9.67 \times 10^{-5}$ | 44  | 5.695528 | $7.97 \times 10^{-5}$ | 123 | 14.07205 | $8.1 \times 10^{-5}$ | 47  | 6.353856 | $8.95 \times 10^{-5}$ |
| $x_6$     | 49    | 5.302793 | $6.83 \times 10^{-5}$ | 54  | 6.692191 | $8.51 \times 10^{-5}$ | 4   | 1.395233 | NaN  | 234 | 62.21843 | $7.96 \times 10^{-5}$ |
| $x_7$     | 43    | 4.615465 | $7.44 \times 10^{-5}$ | 48  | 5.789132 | $6.57 \times 10^{-5}$ | 5   | 1.312244 | NaN  | 79  | 11.94492 | $9.2 \times 10^{-5}$ |
| $x_8$     | 45    | 4.827933 | $7.53 \times 10^{-5}$ | 202 | 23.19352 | $6.59 \times 10^{-5}$ | 4   | 0.682058 | NaN  | 240 | 36.76278 | $7.83 \times 10^{-5}$ |
Table 3. Numerical efficiency of Algorithm 1 with the choice of $\beta^{H}(ω_{k}^{1})$ and $\beta^{H}(ω_{k}^{2})$ versus NDAS method [22] and ICGM method [21].

| Problem 5 | $β^{H}(ω_{k}^{1})$ | $β^{H}(ω_{k}^{2})$ | NDAS | ICGM |
|-----------|---------------------|---------------------|-------|-------|
| DIM       | GUESS | ITR | TIME | NORM | ITR | TIME | NORM | ITR | TIME | NORM | ITR | TIME | NORM | ITR | TIME | NORM |
| 50,000    |        |     |      |      |     |      |      |     |      |      |     |      |      |     |      |      |
| $x_{1}$   | 1      | 0.052829 | 8.57×10^{-6} | 1 | 0.028501 | 8.57×10^{-6} | 1 | 0.039928 | 8.57×10^{-6} | 1 | 0.038071 | 8.57×10^{-6} |
| $x_{2}$   | 1      | 0.03692 | 9.07×10^{-6} | 1 | 0.029346 | 9.07×10^{-6} | 1 | 0.035869 | 9.07×10^{-6} | 1 | 0.026299 | 9.07×10^{-6} |
| $x_{3}$   | 1      | 0.036227 | 9.51×10^{-6} | 1 | 0.038833 | 9.51×10^{-6} | 1 | 0.035661 | 9.51×10^{-6} | 1 | 0.027196 | 9.51×10^{-6} |
| $x_{4}$   | 1      | 0.036327 | 9.94×10^{-6} | 1 | 0.039655 | 9.94×10^{-6} | 1 | 0.035812 | 9.94×10^{-6} | 1 | 0.027108 | 9.94×10^{-6} |
| $x_{5}$   | 1      | 0.035525 | 9.94×10^{-6} | 1 | 0.043298 | 9.94×10^{-6} | 1 | 0.036387 | 9.94×10^{-6} | 1 | 0.027746 | 9.94×10^{-6} |
| $x_{6}$   | 1      | 0.036985 | 1.3×10^{-6}  | 1 | 0.042555 | 1.3×10^{-6}  | 1 | 0.036066 | 1.3×10^{-6}  | 1 | 0.027322 | 1.3×10^{-6}  |
| $x_{7}$   | 1      | 0.035547 | 8.29×10^{-6} | 1 | 0.042826 | 8.29×10^{-6} | 1 | 0.03719 | 8.29×10^{-6} | 1 | 0.026749 | 8.29×10^{-6} |
| $x_{8}$   | 1      | 0.036086 | 4.74×10^{-6} | 1 | 0.057875 | 4.74×10^{-6} | 1 | 0.037694 | 4.74×10^{-6} | 1 | 0.029947 | 4.74×10^{-6} |
| 100,000   |        |     |      |      |     |      |      |     |      |      |     |      |      |     |      |      |
| $x_{1}$   | 1      | 0.068163 | 3.03×10^{-6} | 1 | 0.128598 | 3.03×10^{-6} | 1 | 0.072784 | 3.03×10^{-6} | 1 | 0.059565 | 3.03×10^{-6} |
| $x_{2}$   | 1      | 0.063426 | 3.21×10^{-6} | 1 | 0.104958 | 3.21×10^{-6} | 1 | 0.07371 | 3.21×10^{-6} | 1 | 0.067507 | 3.21×10^{-6} |
| $x_{3}$   | 1      | 0.070829 | 3.36×10^{-6} | 1 | 0.103859 | 3.36×10^{-6} | 1 | 0.073861 | 3.36×10^{-6} | 1 | 0.056567 | 3.36×10^{-6} |
| $x_{4}$   | 1      | 0.072002 | 3.52×10^{-6} | 1 | 0.100046 | 3.52×10^{-6} | 1 | 0.072974 | 3.52×10^{-6} | 1 | 0.055419 | 3.52×10^{-6} |
| $x_{5}$   | 1      | 0.071199 | 3.52×10^{-6} | 1 | 0.103975 | 3.52×10^{-6} | 1 | 0.074358 | 3.52×10^{-6} | 1 | 0.050225 | 3.52×10^{-6} |
| $x_{6}$   | 1      | 0.070771 | 4.59×10^{-7} | 1 | 0.10815 | 4.59×10^{-7} | 1 | 0.077524 | 4.59×10^{-7} | 1 | 0.05661 | 4.59×10^{-7} |
| $x_{7}$   | 1      | 0.074755 | 2.93×10^{-6} | 1 | 0.111332 | 2.93×10^{-6} | 1 | 0.077837 | 2.93×10^{-6} | 1 | 0.054052 | 2.93×10^{-6} |
| $x_{8}$   | 1      | 0.074773 | 1.68×10^{-6} | 1 | 0.117776 | 1.68×10^{-6} | 1 | 0.07581 | 1.68×10^{-6} | 1 | 0.050772 | 1.68×10^{-6} |
Table 3. Cont.

| Problem 6 | $\beta_{k}^{H^*}(\omega_{1}^{k})$ | $\beta_{k}^{H^*}(\omega_{2}^{k})$ | NDAS | ICGM |
|-----------|--------------------------------|--------------------------------|-------|-------|
|           | DIM | GUESS | ITR | TIME | NORM | ITR | TIME | NORM | ITR | TIME | NORM | ITR | TIME | NORM | ITR | TIME | NORM |
| 50,000    |     |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |
| $x_1$     | 3   | 0.062624 | 5.81 | $10^{-5}$ | 7   | 0.270178 | 3.31 | $10^{-8}$ | 1000 | 8.22562 | 0.056283 | 10 | 0.121959 | 6.44 | $10^{-5}$ |
| $x_2$     | 4   | 0.048661 | 9.16 | $10^{-7}$ | 7   | 0.200956 | 2.12 | $10^{-5}$ | 1000 | 10.06622 | 0.069926 | 11 | 0.111443 | 8.61 | $10^{-5}$ |
| $x_3$     | 5   | 0.067382 | 5.23 | $10^{-9}$ | 8   | 0.257818 | 5.23 | $10^{-8}$ | 1000 | 9.859871 | 0.071884 | 12 | 0.121548 | 7.02 | $10^{-5}$ |
| $x_4$     | 6   | 0.07856 | 7.36 | $10^{-5}$ | 11  | 0.318332 | 1.49 | $10^{-7}$ | 6     | 0.061333 | 1.18 | $10^{-5}$ | 14 | 0.136175 | 7.66 | $10^{-5}$ |
| $x_5$     | 6   | 0.080187 | 7.36 | $10^{-5}$ | 11  | 0.347429 | 1.49 | $10^{-7}$ | 6     | 0.070442 | 1.17 | $10^{-5}$ | 14 | 0.148883 | 7.66 | $10^{-5}$ |
| $x_6$     | 8   | 0.138834 | 6.23 | $10^{-6}$ | 10  | 0.238758 | 2.98 | $10^{-13}$ | 1000 | 11.3173 | 0.07266 | 15 | 0.146064 | 4.63 | $10^{-5}$ |
| $x_7$     | 3   | 0.053878 | 5.54 | $10^{-5}$ | 5   | 0.137401 | 4.42 | $10^{-5}$ | 5     | 0.051718 | 1.07 | $10^{-7}$ | 8  | 0.080437 | 3.88 | $10^{-5}$ |
| $x_8$     | 6   | 0.09511 | 7.47 | $10^{-8}$ | 10  | 0.24638 | 8.29 | $10^{-5}$ | 9     | 0.109924 | 6.67 | $10^{-6}$ | 10 | 0.10552 | 1.77 | $10^{-5}$ |
| 100,000   |     |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |
| $x_1$     | 3   | 0.108583 | 7.98 | $10^{-5}$ | 7   | 0.470761 | 2.9 | $10^{-8}$ | 1000 | 19.14354 | 0.079596 | 10 | 0.22311 | 9.1 | $10^{-5}$ |
| $x_2$     | 4   | 0.133273 | 1.06 | $10^{-6}$ | 7   | 0.455045 | 3.77 | $10^{-5}$ | 1000 | 18.09004 | 0.098893 | 12 | 0.229377 | 3.65 | $10^{-5}$ |
| $x_3$     | 5   | 0.160068 | 3.72 | $10^{-9}$ | 8   | 0.513187 | 4.32 | $10^{-8}$ | 1000 | 17.21491 | 0.101663 | 12 | 0.242816 | 9.92 | $10^{-5}$ |
| $x_4$     | 7   | 0.22096 | 1.96 | $10^{-9}$ | 11  | 0.568055 | 3.09 | $10^{-7}$ | 6     | 0.113298 | 1.67 | $10^{-5}$ | 14 | 0.271454 | 5.17 | $10^{-5}$ |
| $x_5$     | 7   | 0.250454 | 1.96 | $10^{-9}$ | 11  | 0.568372 | 3.09 | $10^{-7}$ | 6     | 0.148285 | 1.67 | $10^{-5}$ | 14 | 0.284436 | 5.17 | $10^{-5}$ |
| $x_6$     | 8   | 0.329174 | 1.01 | $10^{-5}$ | 10  | 0.540962 | 2.59 | $10^{-5}$ | 1000 | 19.08673 | 0.102761 | 15 | 0.298349 | 6.56 | $10^{-5}$ |
| $x_7$     | 3   | 0.127818 | 7.87 | $10^{-5}$ | 7   | 0.415979 | 2.38 | $10^{-9}$ | 5     | 0.096884 | 1.43 | $10^{-7}$ | 8  | 0.169584 | 5.47 | $10^{-5}$ |
| $x_8$     | 6   | 0.227238 | 1.08 | $10^{-7}$ | 11  | 0.579637 | 1.2 | $10^{-8}$ | 9     | 0.215587 | 9.58 | $10^{-6}$ | 10 | 0.195059 | 2.5 | $10^{-5}$ |
Table 4. Numerical efficiency of Algorithm 1 with the choice of $\beta_h^{H*}(\omega_k^2)$ and $\beta_h^{H*}(\omega_k^2)$ versus NDAS method [22] and ICGM method [21].

| Problem 7 | $\beta_h^{H*}(\omega_k^2)$ | $\beta_h^{H*}(\omega_k^2)$ | NDAS | ICGM |
|-----------|----------------|----------------|--------|-------|
| DIM       | GUESS | ITR | TIME | NORM | ITR | TIME | NORM | ITR | TIME | NORM |
| 50,000    | x_1   | 3   | 0.075366 | 8.32 × 10^{-7} | 3   | 0.08244 | 8.32 × 10^{-7} | 10  | 0.142476 | 4.43 × 10^{-5} | 9   | 0.08795 | 7.07 × 10^{-5} |
|          | x_2   | 5   | 0.092936 | 7.6 × 10^{-16} | 6   | 0.142385 | 4.95 × 10^{-17} | 12  | 0.175039 | 4.6 × 10^{-5} | 11  | 0.101039 | 5.09 × 10^{-5} |
|          | x_3   | 5   | 0.090729 | 4.11 × 10^{-11} | 6   | 0.155189 | 2.87 × 10^{-12} | 13  | 0.212816 | 5.58 × 10^{-5} | 12  | 0.135433 | 5.15 × 10^{-5} |
|          | x_4   | 10  | 0.184006 | 7.24 × 10^{-10} | 10  | 0.231393 | 8.25 × 10^{-13} | 19  | 0.321698 | 6.65 × 10^{-5} | 20  | 0.243073 | 8.12 × 10^{-5} |
|          | x_5   | 10  | 0.178508 | 7.24 × 10^{-10} | 10  | 0.236107 | 8.25 × 10^{-13} | 19  | 0.316433 | 6.65 × 10^{-5} | 20  | 0.211838 | 8.05 × 10^{-5} |
|          | x_6   | 7   | 0.133278 | 8.2 × 10^{-7} | 9   | 0.208276 | 1.58 × 10^{-10} | 17  | 0.304249 | 8.33 × 10^{-5} | 14  | 0.148348 | 4.43 × 10^{-5} |
|          | x_7   | 3   | 0.060296 | 7.62 × 10^{-5} | 4   | 0.108939 | 1.41 × 10^{-6} | 13  | 0.230127 | 6.01 × 10^{-5} | 12  | 0.14197 | 6.93 × 10^{-5} |
|          | x_8   | 9   | 0.163411 | 4.97 × 10^{-5} | 19  | 0.4887 | 9.6 × 10^{-5} | 20  | 0.336258 | 5.31 × 10^{-5} | 18  | 0.249333 | 6.81 × 10^{-5} |
| 100,000   | x_1   | 3   | 0.115519 | 1.18 × 10^{-6} | 3   | 0.15752 | 1.18 × 10^{-6} | 10  | 0.365394 | 6.26 × 10^{-5} | 9   | 0.216016 | 9.99 × 10^{-5} |
|          | x_2   | 5   | 0.186664 | 1.16 × 10^{-15} | 6   | 0.314727 | 1.03 × 10^{-16} | 12  | 0.421284 | 6.5 × 10^{-5} | 11  | 0.245532 | 7.2 × 10^{-5} |
|          | x_3   | 5   | 0.183213 | 5.81 × 10^{-11} | 6   | 0.276542 | 4.06 × 10^{-12} | 13  | 0.449368 | 7.89 × 10^{-5} | 12  | 0.241829 | 7.28 × 10^{-5} |
|          | x_4   | 10  | 0.363477 | 1.02 × 10^{-9} | 10  | 0.48296 | 1.17 × 10^{-12} | 19  | 0.640634 | 9.41 × 10^{-5} | 21  | 0.535285 | 5.45 × 10^{-5} |
|          | x_5   | 10  | 0.360938 | 1.02 × 10^{-9} | 10  | 0.450866 | 1.17 × 10^{-12} | 19  | 0.649932 | 9.41 × 10^{-5} | 21  | 0.519055 | 5.44 × 10^{-5} |
|          | x_6   | 7   | 0.264049 | 1.16 × 10^{-6} | 9   | 0.455213 | 6.79 × 10^{-9} | 18  | 0.634508 | 4.24 × 10^{-5} | 14  | 0.31158 | 6.27 × 10^{-5} |
|          | x_7   | 4   | 0.147267 | 4.94 × 10^{-6} | 4   | 0.185884 | 2.66 × 10^{-7} | 13  | 0.456271 | 8.49 × 10^{-5} | 12  | 0.275253 | 9.8 × 10^{-5} |
|          | x_8   | 9   | 0.319735 | 7.03 × 10^{-5} | 15  | 0.697124 | 8.88 × 10^{-5} | 20  | 0.677796 | 7.71 × 10^{-5} | 18  | 0.477871 | 9.63 × 10^{-5} |
### Table 4. Cont.

| Problem 8 | $\beta_{k}^{H^*}(\omega_{k}^{1})$ | $\beta_{k}^{H^*}(\omega_{k}^{2})$ | NDAS | ICGM |
|-----------|----------------------------------|----------------------------------|------|------|
| DIM GUESS | ITR    | TIME  | NORM | ITR    | TIME  | NORM | ITR    | TIME  | NORM | ITR    | TIME  | NORM |
| 50,000    |        |       |      |        |       |      |        |       |      |        |       |      |
| $x_1$     | 1      | 0.033714 | 8.57×10^{-6} | 1 | 0.055997 | 8.57×10^{-6} | 1 | 0.029796 | 8.57×10^{-6} | 1 | 0.028779 | 8.57×10^{-6} |
| $x_2$     | 1      | 0.028881 | 9.07×10^{-6} | 1 | 0.048449 | 9.07×10^{-6} | 1 | 0.028738 | 9.07×10^{-6} | 1 | 0.027902 | 9.07×10^{-6} |
| $x_3$     | 1      | 0.03261 | 9.51×10^{-6} | 1 | 0.047632 | 9.51×10^{-6} | 1 | 0.02972 | 9.51×10^{-6} | 1 | 0.024409 | 9.51×10^{-6} |
| $x_4$     | 1      | 0.035589 | 9.94×10^{-6} | 1 | 0.045574 | 9.94×10^{-6} | 1 | 0.032915 | 9.94×10^{-6} | 1 | 0.029429 | 9.94×10^{-6} |
| $x_5$     | 1      | 0.035606 | 9.94×10^{-6} | 1 | 0.046116 | 9.94×10^{-6} | 1 | 0.035012 | 9.94×10^{-6} | 1 | 0.031084 | 9.94×10^{-6} |
| $x_6$     | 1      | 0.03751 | 1.3×10^{-6} | 1 | 0.046566 | 1.3×10^{-6} | 1 | 0.034991 | 1.3×10^{-6} | 1 | 0.028503 | 1.3×10^{-6} |
| $x_7$     | 1      | 0.037605 | 8.29×10^{-6} | 1 | 0.045682 | 8.29×10^{-6} | 1 | 0.035925 | 8.29×10^{-6} | 1 | 0.027888 | 8.29×10^{-6} |
| $x_8$     | 1      | 0.03789 | 4.74×10^{-6} | 1 | 0.050778 | 4.74×10^{-6} | 1 | 0.036782 | 4.74×10^{-6} | 1 | 0.02298 | 4.74×10^{-6} |
| 100,000   |        |       |      |        |       |      |        |       |      |        |       |      |
| $x_1$     | 1      | 0.076523 | 3.03×10^{-6} | 1 | 0.110874 | 3.03×10^{-6} | 1 | 0.076747 | 3.03×10^{-6} | 1 | 0.04965 | 3.03×10^{-6} |
| $x_2$     | 1      | 0.075618 | 3.21×10^{-6} | 1 | 0.098932 | 3.21×10^{-6} | 1 | 0.071353 | 3.21×10^{-6} | 1 | 0.046196 | 3.21×10^{-6} |
| $x_3$     | 1      | 0.073683 | 3.36×10^{-6} | 1 | 0.089028 | 3.36×10^{-6} | 1 | 0.071895 | 3.36×10^{-6} | 1 | 0.056912 | 3.36×10^{-6} |
| $x_4$     | 1      | 0.075373 | 3.52×10^{-6} | 1 | 0.097382 | 3.52×10^{-6} | 1 | 0.07301 | 3.52×10^{-6} | 1 | 0.063382 | 3.52×10^{-6} |
| $x_5$     | 1      | 0.075046 | 3.52×10^{-6} | 1 | 0.107567 | 3.52×10^{-6} | 1 | 0.072606 | 3.52×10^{-6} | 1 | 0.055507 | 3.52×10^{-6} |
| $x_6$     | 1      | 0.074018 | 4.59×10^{-7} | 1 | 0.092188 | 4.59×10^{-7} | 1 | 0.072067 | 4.59×10^{-7} | 1 | 0.057577 | 4.59×10^{-7} |
| $x_7$     | 1      | 0.076113 | 2.93×10^{-6} | 1 | 0.097692 | 2.93×10^{-6} | 1 | 0.071651 | 2.93×10^{-6} | 1 | 0.055431 | 2.93×10^{-6} |
| $x_8$     | 1      | 0.076265 | 1.68×10^{-6} | 1 | 0.108356 | 1.68×10^{-6} | 1 | 0.072278 | 1.68×10^{-6} | 1 | 0.049841 | 1.68×10^{-6} |
Table 5. Numerical efficiency of Algorithm 1 with the choice of $\beta^{H^*}(\omega_k^1)$ and $\beta^{H^*}(\omega_k^2)$ versus NDAS method [22] and ICGM method [21].

| Problem 9 | $\beta^{H^*}(\omega_k^1)$ | $\beta^{H^*}(\omega_k^2)$ | NDAS | ICGM |
|-----------|----------------|----------------|------|------|
| DIM       | GUESS | ITR | TIME | NORM | ITR | TIME | NORM | ITR | TIME | NORM | ITR | TIME | NORM | ITR | TIME | NORM |
| 50,000    |       |     |      |      |     |      |      |     |      |      |     |      |      |     |      |      |
| x1        | 5     | 0.065657 | 7  | 4.210^{-5} | 7  | 0.189046 | 2.750^{-5} | 7  | 0.119231 | 8.690^{-5} | 11  | 0.126884 | 2.960^{-5} |
| x2        | 5     | 0.06624 | 5.830^{-5} | 9  | 0.224013 | 8.410^{-5} | 8  | 0.151618 | 1.250^{-5} | 11  | 0.151836 | 2.690^{-5} |
| x3        | 5     | 0.083632 | 1.920^{-5} | 7  | 0.177367 | 1.330^{-5} | 7  | 0.138794 | 8.750^{-5} | 10  | 0.128026 | 8.970^{-5} |
| x4        | 9     | 0.145705 | 3.320^{-5} | 11  | 0.314592 | 5.970^{-5} | 14  | 0.26809 | 1.990^{-5} | 18  | 0.21681 | 4.320^{-5} |
| x5        | 9     | 0.166716 | 2.080^{-5} | 11  | 0.302915 | 3.680^{-5} | 12  | 0.223362 | 4.230^{-5} | 18  | 0.198065 | 4.320^{-5} |
| x6        | 6     | 0.119915 | 8.960^{-5} | 9  | 0.261753 | 8.820^{-5} | 10  | 0.186299 | 7.510^{-5} | 12  | 0.137626 | 4.190^{-5} |
| x7        | 9     | 0.168087 | 7.220^{-5} | 11  | 0.255316 | 7.070^{-5} | 9  | 0.177588 | 7.870^{-5} | 18  | 0.193596 | 4.860^{-5} |
| x8        | 42    | 0.873011 | 8.910^{-6} | 11  | 0.257521 | 8.540^{-5} | 17  | 0.312101 | 4.430^{-5} | 24  | 0.262022 | 5.320^{-5} |
| 100,000   |       |     |      |      |     |      |      |     |      |      |     |      |      |     |      |      |
| x1        | 6     | 0.240266 | 5.410^{-6} | 8  | 0.39591 | 7.430^{-5} | 8  | 0.323746 | 1.50^{-5} | 11  | 0.247176 | 4.190^{-5} |
| x2        | 5     | 0.211107 | 8.240^{-5} | 8  | 0.416573 | 2.590^{-5} | 8  | 0.326042 | 1.760^{-5} | 11  | 0.27141 | 3.810^{-5} |
| x3        | 5     | 0.201544 | 2.720^{-5} | 8  | 0.397299 | 4.020^{-5} | 8  | 0.321037 | 1.510^{-5} | 11  | 0.244046 | 2.880^{-5} |
| x4        | 9     | 0.355143 | 4.390^{-5} | 12  | 0.587486 | 4.350^{-5} | 13  | 0.48467 | 9.190^{-5} | 18  | 0.393596 | 6.110^{-5} |
| x5        | 9     | 0.362649 | 3.530^{-5} | 11  | 0.542996 | 5.210^{-5} | 13  | 0.486746 | 7.210^{-5} | 18  | 0.383299 | 6.110^{-5} |
| x6        | 7     | 0.278369 | 6.540^{-6} | 10  | 0.500734 | 6.470^{-6} | 11  | 0.419632 | 2.050^{-5} | 12  | 0.2268 | 5.930^{-5} |
| x7        | 10    | 0.37336 | 5.270^{-6} | 12  | 0.545659 | 9.290^{-5} | 10  | 0.394767 | 1.360^{-5} | 18  | 0.382106 | 6.870^{-5} |
| x8        | 48    | 1.895847 | 6.760^{-5} | 12  | 0.578936 | 5.080^{-5} | 17  | 0.612865 | 6.270^{-5} | 24  | 0.51844 | 7.520^{-5} |
From Table 1, which includes Problems 1–2, it can be observed that our algorithm wins both the number of iterations and the CPU time, followed by the ICGM and NDAS algorithms. Nevertheless, note that the NDAS method failed to solve Problem 1 for almost six initial points, while the NDAS method has an advantage over the ICGM method for Problem 2 both for the number of iterations and computing time. Similarly, for all remaining problems, our algorithm has the less number of iterations and CPU time, except for Problems 5 and 8. For these two problems, the NDAS and ICGM methods have substantially less computing time and compete with the proposed algorithm for the number of iterations. We provide a picture of the overall performance for both the number of iterations and the CPU time using the Dolan and Moré [30] performance profile. Based on their profile, the best method is the method whose curve is at the top left corner. It could be observed that our algorithm reflected the top left curves in Figures 1 and 2 for the number of iterations and the time of the CPU, respectively. This clearly shows that our algorithm is the best in terms of the number of iteration and CPU time compared to the NDAS and ICGM algorithms.

Figure 1. Performance of Algorithm 1 with the choice of $\beta_k^H(\omega_k^1)$ and $\beta_k^H(\omega_k^2)$ versus NDAS method [22] and ICGM method [21] for number of iterations.

Figure 2. Performance of Algorithm 1 with the choice of $\beta_k^H(\omega_k^1)$ and $\beta_k^H(\omega_k^2)$ versus NDAS method [22] and ICGM method [21] for the CPU time in second.

5. Conclusions

This paper presented an inexact optimal hybrid CG algorithm for solving a system of symmetric nonlinear equations. The hybrid method presented here is the convex combination of the optimal DL method and the optimal three-term PRP CG method using the Li and Fukushima approximate gradient relation. The method is matrix-free and derivative-free, enabling it to handle large-scale system of symmetric nonlinear equations effectively.
Moreover, some mild assumptions are used to prove the global convergence of the method. Nevertheless, the numerical efficiency of the method was also demonstrated using some test problems by comparing the number of iterations and the CPU time compared to the NDAS method [22] and ICGM method [21]. Its overall success has shown that our proposed approach is a better alternative for solving symmetric nonlinear equations in terms of the number of iterations and the CPU time. Generally, the CG methods for the symmetric nonlinear systems can be improved by devising new efficient CG parameters or modifying the existing CG directions using the appropriate techniques. As a possibility for future research, time complexity analysis and comparison with evolutionary optimization algorithms for global optimization problems could be looked into.

**Author Contributions:** Conceptualization, J.S.; methodology, J.S.; software, A.B.A.; validation, K.M. and A.S.; formal analysis, K.M. and K.O.A.; investigation, K.M. and A.B.A.; resources, K.M.; data curation, A.B.A. and A.S.; writing—original draft preparation, J.S.; writing—review and editing, A.B.A.; visualization, K.O.A.; supervision, K.M.; project administration, A.S.; funding acquisition, K.O.A. All authors have read and agreed to the published version of the manuscript.

**Funding:** The first author is grateful to TWAS/CUI for the Award of FR number: 3240299486.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Acknowledgments:** The second author was financially supported by Rajamangala University of Technology Phra Nakhon (RMUTP) Research Scholarship. The last two authors acknowledge with thanks the Department of Mathematics and Applied Mathematics at the Sefako Makgatho Health Sciences University.

**Conflicts of Interest:** The authors declare no conflicts of interest.

**Sample Availability:** Samples of the compounds are available from the authors.

**References**

1. Li, D.; Fukushima, M. A Globally and Superlinearly Convergent Gauss-Newton-Based BFGS Method for Symmetric Nonlinear Equations. *SIAM J. Numer. Anal.* 1999, 37, 152–172. [CrossRef]
2. 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]
3. Waziri, M.Y.; Sabi’u, J. A derivative-free conjugate gradient method and its global convergence for solving symmetric nonlinear equations. *Int. J. Math. Math. Sci.* 2015, 2015, 961487. [CrossRef]
4. Yuan, G.; Lu, X.; Wei, Z. BFGS trust-region method for symmetric nonlinear equations. *J. Comput. Appl. Math.* 2009, 230, 44–58. [CrossRef]
5. Zhou, W.; Shen, D. An inexact PRP conjugate gradient method for symmetric nonlinear equations. *Num. Funct. Anal. Opt.* 2014, 35, 370–388. [CrossRef]
6. Waziri, M.Y.; Sabi’u, J. An alternative conjugate gradient approach for large-scale symmetric nonlinear equations. *J. Math. Comput. Sci.* 2016, 6, 855–874.
7. Li, D.H.; Wang, X.L. A modified Fletcher-Reeves-type derivative-free method for symmetric nonlinear equations. *Numer. Algebra Control Optim.* 2011, 1, 71–82. [CrossRef]
8. Polyak, B.T. The conjugate gradient method in extreme problems. *USSR Comput. Math. Math. Phys.* 1969, 9, 94–112. [CrossRef]
9. Polak, E.; Ribière, G. Note on the convergence of methods of conjugate directions. *RIRO* 1969, 3, 35–43.
10. Babaie-Kafaki, S.; Ghanbari, R. A descent extension of the Polak-Ribiére-Polyak conjugate gradient method. *Comput. Math. Appl.* 2014, 68, 2005–2011. [CrossRef]
11. Yu, G.; Guan, L.; Li, G. Global convergence of modified Polak-Ribiére-Polyak conjugate gradient methods with sufficient descent property. *J. Ind. Manag. Optim.* 2008, 4, 565. [CrossRef]
12. Sabi’u, J. Enhanced derivative-free conjugate gradient method for solving symmetric nonlinear equations. *Int. J. Adv. Appl. Sci.* 2016, 5, 50–57. [CrossRef]
13. Dai, Y.H.; Liao, L.Z. New conjugacy conditions and related nonlinear conjugate gradient methods. *Appl. Math. Opt.* 2001, 43, 87–101. [CrossRef]
14. Babaie-Kafaki, S.; Ghanbari, R. The Dai-Liao nonlinear conjugate gradient method with optimal parameter choices. Eur. J. Opt. Res. 2014, 234, 625–630. [CrossRef]
15. Babaie-Kafaki, S.; Ghanbari, R. An optimal extension of the Polak-Ribièr-Polyak conjugate gradient method. Numer. Func. Anal. Opt. 2017, 38, 1115–1124. [CrossRef]
16. Zhang, L.; Zhou, W.; Li, D.H. Global convergence of a modified Fletcher-Reeves conjugate gradient method with Armijo-type line search. Numer. Math. 2006, 104, 561–572. [CrossRef]
17. Zhang, L.; Zhou, W.; Li, D. Some descent three-term conjugate gradient methods and their global convergence. Optim. Methods Softw. 2007, 22, 697–711. [CrossRef]
18. Zhou, W.; Chen, X. On the convergence of a derivative-free HS type method for symmetric nonlinear equations. Adv. Model. Opt. 2012, 3, 645–654.
19. Xiao, Y.; Wu, C.; Wu, S.Y. Norm descent conjugate gradient methods for solving symmetric nonlinear equations. J. Glob. Opt. 2015, 62, 751–762. [CrossRef]
20. Dauda, M.K.; Mamat, M.; Mohamed, M.A.; Mohamad, F.S.; Waziri, M.Y. Derived Conjugate Gradient Parameter For Solving Symmetric Systems of Nonlinear Equations. Far East J. Math. Sci. FJMS 2017, 102, 2017. [CrossRef]
21. 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]
22. Abubakar, A.B.; Kumam, P.; Awwal, A.M. An inexact conjugate gradient method for symmetric nonlinear equations. Comput. Math. Methods 2019, 1, e1065. [CrossRef]
23. Waziri, M.Y.; Yusuf Kufena, M.; Halliu, A.S. Derivative-Free Three-Term Spectral Conjugate Gradient Method for Symmetric Nonlinear Equations. Thai J. Math. 2020, 18, 1417–1431.
24. Sabi’u, J.; Muangchoo, J.K.; Shah, A.; Abubakar, A.B.; Jolaoso, L.O. A Modified PRP-CG Type Derivative-Free Algorithm with Optimal Choices for Solving Large-Scale Nonlinear Symmetric Equations. Symmetry 2021, 13, 234. [CrossRef]
25. Dennis, J.E.; More, J.J. A characterization of superlinear convergence and its application to quasi-Newton methods. Math. Comp. 1974, 28, 549–560. [CrossRef]
26. Yakubu, U.A.; Mamat, M.; Mohamad, M.A.; Rivaie, M.; Sabi’u, J. A recent modification on Dai-Liao conjugate gradient method for solving symmetric nonlinear equations. Far East J. Math. Sci. 2018, 12, 1961–1974. [CrossRef]
27. Sabi’u, J.; Gadu, A.M. A Projected Hybrid Conjugate Gradient Method for Solving Large-scale System of Nonlinear Equations. Malays. J. Comput. Appl. Math. 2018, 1, 10–20.
28. Cruz, W.L.; Martínez, J.; Raydan, M. Spectral residual method without gradient information for solving large-scale nonlinear systems of equations. Math. Comput. 2006, 75, 1429–1448. [CrossRef]
29. Mohammad, H.; Abubakar, A.B. A descent derivative-free algorithm for nonlinear monotone equations with convex constraints. RAIRO Op. Res. 2020, 54, 489–505. [CrossRef]
30. Dolan, E.D.; Moré, J.J. Benchmarking optimization software with performance profiles. Math. Program. 2002, 91, 201–213. [CrossRef]