Article

A Modified Liu and Storey Conjugate Gradient Method for Large Scale Unconstrained Optimization Problems

Zabidin Salleh ^{1,*} , Ghaliah Alhamzi ², Ibtsam Masmali ³ and Ahmad Alhawarat ¹

- ¹ Department of Mathematics, Faculty of Ocean Engineering Technology and Informatics, Universiti Malaysia Terengganu, Kuala Nerus 21030, Terengganu, Malaysia; ahmadhawarat@tdtu.edu.vn
² Department of Mathematics and Statistics, College of Science, Imam Mohammad Ibn Saud Islamic University (IMSIU), Riyadh 13318, Saudi Arabia; gyalhamzi@imamu.edu.sa
³ Department of Mathematics, College of Science, Jazan University, Jazan 45142, Saudi Arabia; iamasalmi@jazanu.edu.sa
* Correspondence: zabidin@umt.edu.my

Abstract: The conjugate gradient method is one of the most popular methods to solve large-scale unconstrained optimization problems since it does not require the second derivative, such as Newton's method or approximations. Moreover, the conjugate gradient method can be applied in many fields such as neural networks, image restoration, etc. Many complicated methods are proposed to solve these optimization functions in two or three terms. In this paper, we propose a simple, easy, efficient, and robust conjugate gradient method. The new method is constructed based on the Liu and Storey method to overcome the convergence problem and descent property. The new modified method satisfies the convergence properties and the sufficient descent condition under some assumptions. The numerical results show that the new method outperforms famous CG methods such as CG-Descent 5.3, Liu and Storey, and Dai and Liao. The numerical results include the number of iterations and CPU time.

Keywords: conjugate gradient; inexact line search; descent condition

Citation: Salleh, Z.; Alhamzi, G.; Masmali, I.; Alhawarat, A. A Modified Liu and Storey Conjugate Gradient Method for Large Scale Unconstrained Optimization Problems. *Algorithms* **2021**, *14*, 227. <https://doi.org/10.3390/a14080227>

Academic Editors: Ferrante Neri and Frank Werner

Received: 21 June 2021
Accepted: 25 July 2021
Published: 28 July 2021

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

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The optimization problem that we want to solve takes the following form:

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

where $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is a continuous and differentiable function, and its gradient ($\nabla f(x)$) is available. $x_1 \in \mathbb{R}^n$ is an arbitrary initial point for new functions or non-standard functions. The CG method generates a sequence of iterates x_k (vector) as follows:

Where x_k is the current iteration and $\alpha_k > 0$ is a step size obtained from a line search

$$x_{k+1} = x_k + \alpha_k d_k, k = 1, 2, \dots, \quad (1)$$

such as exact or inexact line search. The search direction d_k in the CG method requests only the first derivative of the optimization function, and it is defined as follows:

$$d_k = \begin{cases} -g_k, & \text{if } k = 1, \\ -g_k + \beta_k d_{k-1}, & \text{if } k \geq 2, \end{cases} \quad (2)$$

where $g(x) = \nabla f(x)$, $g_k = g(x_k)$ and β_k is known as the CG formula or CG parameter. To obtain the step size α_k , we can use the following line searches:

Exact line search, which is given by the following equation

$$f(x_k + \alpha_k d_k) = \min f(x_k + \alpha d_k), \alpha \geq 0. \quad (3)$$

However, Equation (3) is computationally expensive if the function has multiple local minima or stationary points.

Inexact line search, normally we use strong Wolfe–Powell (SWP) [1,2], which is defined as follows:

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \delta \alpha_k g_k^T d_k, \quad (4)$$

and

$$\left| g(x_k + \alpha_k d_k)^T d_k \right| \leq \sigma \left| g_k^T d_k \right|, \quad (5)$$

where $0 < \delta < \sigma < 1$.

From Equation (4), we can note that

$$f(x_k + \alpha_k d_k) - f(x_k) \leq \delta \alpha_k g_k^T d_k,$$

i.e.,

$$f(x_{k+1}) - f(x_k) \leq \delta \alpha_k g_k^T d_k.$$

If Equation (7) holds, then the following equation holds

$$f(x_{k+1}) \leq f(x_k).$$

Also, note that Equation (5) can be written as follows

$$\sigma g_k^T d_k \leq g(x_k + \alpha_k d_k)^T d_k \leq -\sigma g_k^T d_k.$$

Another version of Wolfe–Powell (WP) line search is the weak Wolfe–Powell (WWP) line search, which is defined by Equation (4) and the following equation

$$g(x_k + \alpha_k d_k)^T d_k \geq \sigma g_k^T d_k. \quad (6)$$

The most famous classical formulas of the CG methods are Hestenses–Stiefel (HS) [3], Polak–Ribiere–Polyak (PRP) [4], Liu and Storey (LS) [5], Fletcher–Reeves (FR) [6], Fletcher (CD) [7], as well as Dai and Yuan (DY) [8], given as follows:

$$\begin{aligned} \beta_k^{HS} &= \frac{g_k^T y_{k-1}}{d_{k-1}^T y_{k-1}}, \quad \beta_k^{PRP} = \frac{g_k^T y_{k-1}}{\|g_{k-1}\|^2}, \quad \beta_k^{LS} = -\frac{g_k^T y_{k-1}}{d_{k-1}^T g_{k-1}} \\ \beta_k^{FR} &= \frac{\|g_k\|^2}{\|g_{k-1}\|^2}, \quad \beta_k^{CD} = -\frac{\|g_k\|^2}{d_{k-1}^T g_{k-1}}, \quad \beta_k^{DY} = \frac{\|g_k\|^2}{d_{k-1}^T g_{k-1}}, \end{aligned}$$

where $y_{k-1} = g_k - g_{k-1}$.

The first group of classical methods, i.e., (HS), (PRP), and (LS), are efficient. However, Powell [9] proposed a counterexample to show that there exists a non-convex function. PRP, HS, and LS fail to satisfy the convergence properties even when the exact line search is used. Powell suggested the importance to achieve the convergence properties of PRP, HS, and LS methods, where the methods should be non-negative. Gilbert and Nocedal [10] showed that non-negative PRP or HS, i.e., $\beta_k = \max\{\beta_k^{HS}, 0\}$, is convergent under special different line searches. The second group of classical CG methods, i.e., (FR), (PRP), (DY), are robust and converge. However, the group is not efficient.

The descent condition (downhill condition) performs an important ruling in the convergence of the CG method and its robustness, which is defined as follows

$$g_k^T d_k < 0.$$

Al–Baali [11] proposed another version of the descent condition called the sufficient descent condition, which plays a significant role in the convergence of the CG method. The author used Equation (7) to establish the convergence properties of β_k^{FR} , which are defined as follows: There exists a constant $c > 0$ such that

$$g_k^T d_k \leq -c \|g_k\|^2, \forall k \in N. \quad (7)$$

In this case, the function is quadratic, i.e., $f(x) = g^T x + (1/2)x^T Hx$ and the step size obtained by exact line search (3) indicates that the CG method satisfies the conjugacy condition, i.e., $d_i^T H d_j = 0, \forall i \neq j$. Using the mean value theorem and exact line search of Equation (2), we can obtain β_k^{HS} . From the quasi-Newton method, BFGS method, the limited memory (LBFGS) method, and using Equation (2), Dai and Liao [12] proposed the following conjugacy condition.

$$d_k^T y_{k-1} = -t g_k^T s_{k-1}, \quad (8)$$

where $s_{k-1} = x_k - x_{k-1}$, and $t \geq 0$, [12] use $t = 0.1$ in the numerical experiments. In the case of $t = 0$, Equation (8) becomes the classical conjugacy condition. By using Equations (2) and (8), Dai and Liao [12] proposed the following CG formula

$$\beta_k^{DL} = \frac{g_k^T y_{k-1}}{d_{k-1}^T y_{k-1}} - t \frac{g_k^T s_{k-1}}{d_{k-1}^T y_{k-1}}. \quad (9)$$

However, β_k^{DL} face the same problem as β_k^{PRP} and β_k^{HS} , i.e., β_k^{DL} is not non-negative in general. Thus, Dai and Liao [12] replaced Equation (9) by

$$\beta_k^{DL+} = \max\{\beta_k^{HS}, 0\} - t \frac{g_k^T s_{k-1}}{d_{k-1}^T y_{k-1}}. \quad (10)$$

Hager and Zhang [13,14] presented a modified CG parameter that satisfies the descent property for any inexact line search with $g_k^T d_k \leq -(7/8)\|g_k\|^2$. This new version of the CG method is convergent whenever the line search satisfies the WP line search. This formula is given as follows:

$$\beta_k^{HZ} = \max\{\beta_k^N, \eta_k\} \quad (11)$$

where $\beta_k^N = \frac{1}{d_k^T y_k} (y_k - 2d_k \frac{\|y_k\|^2}{d_k^T y_k})^T g_k$, $\eta_k = -\frac{1}{\|d_k\| \min\{\eta, \|g_k\|\}}$, and $\eta > 0$ is a constant.

Note that if $t = 2 \frac{\|y_k\|^2}{s_k^T y_k}$, then $\beta_k^N = \beta_k^{DY}$. Zhang et al. [15] and Zhang and Xu [16] used Taylor's series and presented the following modified secant relation:

$$B_k s_{k-1} = \bar{y}_{k-1} \text{ where } \bar{y}_{k-1} = y_{k-1} + \frac{\theta_{k-1}}{\|s_{k-1}\|^2} s_k \text{ and} \quad (12)$$

$$\theta_{k-1} = 6(f_{k-1} - f_k) + 3(g_{k-1} + g_k)^T s_{k-1}.$$

From (12), Yabe and Takano [17] proposed the following formula:

$$\beta_k^{YT} = \frac{g_k^T (z_{k-1} - t s_{k-1})}{d_{k-1}^T z_{k-1}} \text{ where } z_{k-1} = y_{k-1} + \rho \frac{\theta_{k-1}}{\|s_{k-1}\|^2} s_k \text{ and } \rho \geq 0. \quad (13)$$

Moreover, Razieh Dehghani et al. [18] proposed the following formula:

$$\beta_k^{YT} = \frac{g_k^T (y_{k-1}^* - t s_{k-1})}{d_{k-1}^T y_{k-1}^*}. \quad (14)$$

Jiang et al. [19] proposed the CG method by replacing d_{k-1} by g_{k-1} in β_k^{HS} as follows:

$$\beta_k^{JSL} = \frac{g_k^T y_{k-1}}{g_{k-1}^T y_{k-1}}.$$

Furthermore, Wei et al. [20] proposed a non-negative formula, referred to as the WYL coefficient. It is defined as follows:

$$\beta_k^{WYL} = \frac{\|g_k\|^2 - \frac{\|g_k\|}{\|g_{k-1}\|} g_k^T g_{k-1}}{\|g_{k-1}\|^2}$$

This parameter is similar to the PRP coefficient, which possesses convergence with both an exact line and inexact line searches, providing a sufficient descent condition. Many modifications of the WYL coefficient have been suggested. The reader can refer to [21]. Here we mention some of them.

$$\beta_k^{VHS} = \frac{\|g_k\|^2 - \frac{\|g_k\|}{\|g_{k-1}\|} g_k^T g_{k-1}}{d_{k-1}^T (g_k - g_{k-1})} \text{ and } \beta_k^{NPRP} = \frac{\|g_k\|^2 - \frac{\|g_k\|}{\|g_{k-1}\|} |g_k^T g_{k-1}|}{\|g_{k-1}\|^2},$$

which were presented in [22,23], respectively.

Al-Baali et al. [24] proposed a new CG version called (G3TCG) that offers many selections of CG parameters. The same study [24] found that the G3TCG method is more efficient than β_k^{HZ} in some cases and competitive in some other cases.

The convergence rate for the CG method is linear unless the iterative procedure is occasionally restarted at least every n iterations [25]. Beale [26] suggested the use of the two-term CG method instead of the steepest descent (SD) method ($d_k = -g_k, \forall k \geq 1$) as the restart search direction. He also extended this method to a three-term method as a non-restart direction. Furthermore, Powell [25] suggested restarting the search direction using Beale's method if

$$|g_k^T g_{k-1}| > 0.2 \|g_k\|^2,$$

holds for every n iteration, whichever first occurs. Dai and Yuan [27] extended the Powell restart criteria (7) to the following form

$$|g_k^T g_{k-1}| > \tau \|g_k\|^2, \dots \tau \in (0, 1).$$

Besides, Alhawarat et al. [28] presented the following simple formula

$$\beta_k^{AZPRP} = \begin{cases} \frac{\|g_k\|^2 - \mu_k |g_k^T g_{k-1}|}{\|g_{k-1}\|^2}, & \text{if } \|g_k\|^2 > \mu_k |g_k^T g_{k-1}|, \\ 0, & \text{otherwise,} \end{cases} \quad (15)$$

where $\|\cdot\|$ represents the Euclidean norm and μ_k is defined as follows:

$$\mu_k = \frac{\|x_k - x_{k-1}\|}{\|y_k\|}$$

Lemma 3.1 in [28] shows that if Assumption 1 holds and the Beale–Powell restart condition is violated for non-restart search direction, then $\|g_k\|^2 > \mu_k |g_k^T g_{k-1}|$ holds.

Moreover, Kaelo et al. [29] proposed a non-negative CG formula with convergence properties, and compared the new formula with β_k^{AZPRP} , where the proposed method is given as follows:

$$\beta_k^{PKT} = \begin{cases} \frac{\|g_k\|^2 - g_k^T g_{k-1}}{\max\{d_{k-1}^T y_{k-1}, -g_{k-1}^T d_{k-1}\}}, & \text{If } 0 < g_k^T g_{k-1} < \|g_k\|^2 \\ \frac{\|g_k\|^2}{\max\{d_{k-1}^T y_{k-1}, -g_{k-1}^T d_{k-1}\}}, & \text{otherwise.} \end{cases}$$

To solve non-linear monotone equations with convex constraints, [30] proposed using the Powell symmetrical technique to the Liu–Storey conjugate gradient method. The reader can refer to the references for more about non-linear monotone equations with convex constraints and their application [31–34].

2. The Proposed CG Formula and Its Motivation

The CG method with β_k^{LS} cannot satisfy the decent condition. On the other hand, β_k^{LS} comes from the group of efficient CG method as explained before to inherit the efficiency of β_k^{LS} . To avoid the convergence problem, we use β_k^{AZPRP} in [19] and β_k^{DL} in [12] to propose a new non-negative CG method that satisfies the sufficient descent condition with SWP line search. Moreover, convergence properties are achieved. The new formula is a modification of β_k^{LS} , β_k^{DL} , and β_k^{AZPRP} defined as follows:

$$\beta_k^{LS+} = \begin{cases} -\frac{\|g_k\|^2 - \mu_k |g_k^T g_{k-1}|}{d_{k-1}^T g_{k-1}} & \text{if } \|g_k\|^2 > \mu_k |g_k^T g_{k-1}|, \\ \beta_k^{DL-HS} & \text{otherwise} \end{cases} \quad (16)$$

where $\|\cdot\|$ represents the Euclidean norm and

$$\beta_k^{DL-HS} = -\mu_k \frac{g_k^T s_{k-1}}{d_{k-1}^T y_{k-1}}.$$

The proposed method in Equation (16) has the following attitudes:

1. Satisfies the descent condition.
2. Satisfies the convergence properties.
3. Equation (16) is constructed based on β_k^{LS} with restart condition to be non-negative and avoid the convergence problems.
4. Dai and Liao [12] suggest using $-t \left(g_k^T s_{k-1} / d_{k-1}^T y_{k-1} \right)$ instead of steepest descent as a restart criterion, where Equation (16) is restarted by $-\mu_k \left(g_k^T s_{k-1} / d_{k-1}^T y_{k-1} \right)$, more efficient than the steepest descent.
5. The numerical results demonstrate the efficiency and robustness of Equation (16) compared to the other CG methods, including CG-Descent, LS, and DL+.

3. Convergence Analysis of β_k^{LS+}

In the analysis below, we obtain the stationary point from the convergence mean. In addition, if $g_k = 0$, then a stationary point has been obtained. Thus, we assume $g_k \neq 0$ for all $k \geq 1$ in the subsequent analysis. For more about local and global minimum points, the reader can refer to the following useful references [35,36].

To satisfy the convergence analysis of the modified CG method, the following assumption is required.

Assumption 1. A. The level set $\Phi = \{x | f(x) \leq f(x_1)\}$ is bounded; that is, a positive constant λ exists such that

$$\|x\| \leq \lambda, \forall x \in \Phi.$$

B. In some neighborhood P of Φ , f is continuously differentiable, and its gradient is Lipschitz continuous; that is, for all $x, y \in P$, there exists a constant $L > 0$ such that

$$\|g(x) - g(y)\| \leq L\|x - y\|$$

This assumption implies that there exists a positive constant $\bar{\gamma}$ such that

$$\|g(x)\| \leq \bar{\gamma}, \forall x \in P.$$

The following algorithm (Algorithm 1) classifies the steps of the CG method to obtain the stationary point using Equation (16) and SWP line search.

Algorithm 1. The steps to obtain the optimum point of optimization problem by using the CG method

- Step 1 Provide a starting point x_1 . Set the initial search direction $d_1 = -g_1$. Let $k = 1$.
 - Step 2 If a stopping criteria is satisfied, then stop.
 - Step 3 Compute d_k based on (2) with (16).
 - Step 4 Compute α_k using (4) and (5).
 - Step 5 Update x_{k+1} based on (1).
 - Step 6 Set $k := k + 1$ and go to Step 2.
-

In the following section and subsections, we obtain the descent condition and the convergence property of Equation (16) with CG formula, the proof similar to that presented by [12].

3.1. The Descent Property of CG Method with β_k^{LS+}

Theorem 1. Let the sequences $\{x_k\}$ and $\{d_k\}$ be obtained using (1), (2), and β_k^{LS+} , where α_k is computed using the SWP line search (4) and (5). If $\sigma \in (0, \frac{1}{2}]$, then the descent condition given in (7) holds.

Proof. By multiplying (2) by g_k^T , we obtain

$$\begin{aligned} g_k^T d_k &= g_k^T (-g_k + \beta_k d_{k-1}) = -\|g_k\|^2 + \beta_k g_k^T d_{k-1} \\ &= -\|g_k\|^2 + \left(-\frac{\|g_k\|^2 - \mu_k |g_k^T g_{k-1}|}{d_{k-1}^T g_{k-1}} \right) g_k^T d_{k-1}. \end{aligned} \quad (17)$$

By using the proof by induction technique, for $k = 1$, it is true, i.e., $g_1^T d_1 = -\|g_1\|^2$. We now assume that it is true until $k - 1$, thus we can write Equation (16) as follows:

$$\beta_k^{LS+} = \frac{-\|g_k\|^2 + \mu_k |g_k^T g_{k-1}|}{d_{k-1}^T g_{k-1}} \leq \frac{-\|g_k\|^2}{d_{k-1}^T g_{k-1}} + \frac{\mu_k |g_k^T g_{k-1}|}{d_{k-1}^T g_{k-1}} \leq \frac{-\|g_k\|^2}{d_{k-1}^T g_{k-1}} = \frac{\|g_k\|^2}{\|g_{k-1}\|^2}$$

Divide both sides of Equation (18) by $\|g_k\|^2$ and using (5), we obtain

$$-1 + \sigma \frac{g_{k-1}^T d_{k-1}}{\|g_{k-1}\|^2} \leq \frac{g_k^T d_k}{\|g_k\|^2} \leq -1 - \sigma \frac{g_{k-1}^T d_{k-1}}{\|g_{k-1}\|^2}. \quad (18)$$

Repeating the process for the left-hand side of Equation (18), we obtain

$$-1 + \sum_{j=1}^{k-1} \sigma^j = -\sum_{j=0}^{k-1} \sigma^j.$$

Repeating the process for the right hand side of Equation (18) yields

$$-1 - \sum_{j=1}^{k-1} \sigma^j = -2 + \sum_{j=0}^{k-1} \sigma^j.$$

Thus, Equation (18) can be written as follows:

$$-\sum_{j=0}^{k-1} \sigma^j \leq \frac{g_k^T d_k}{\|g_k\|^2} \leq -2 + \sum_{j=0}^{k-1} \sigma^j$$

Since

$$\sum_{j=0}^{k-1} (\sigma)^j < \frac{1 - (\sigma)^k}{1 - \sigma},$$

hence

$$-\frac{1 - (\sigma)^k}{1 - \sigma} \leq \frac{g_k^T d_k}{\|g_k\|^2} \leq -2 + \frac{1 - (\sigma)^k}{1 - \sigma}.$$

When $\sigma \leq 1/2$, we obtain $\frac{1 - (\sigma)^k}{1 - \sigma} < 2$. Let $c = 2 - \frac{1 - (\sigma)^k}{1 - \sigma}$, then

$$c - 2 \leq \frac{g_k^T d_k}{\|g_k\|^2} \leq -c. \quad (19)$$

The proof is complete. $\square$

3.2. The Convergence Property of the CG Method with β_k^{LS+} to Obtain the Stationary Point

Theorem 2. Let Assumption 1 hold. Consider any form of (1) and (2), with the new formula (10), in which α_k is obtained from the SWP line search (4) and (5) with $\sigma \leq 1/2$. Then,

$$\liminf_{k \rightarrow \infty} \|g_k\| = 0.$$

Proof. We will prove the theorem by contradiction. Assume that the conclusion is not true. Then, a constant $\varepsilon > 0$ exists such that

$$\|g_k\| \geq \varepsilon, \quad \forall k \geq 1. \quad (20)$$

Upon squaring both sides of (2), we obtain

$$\|d_k\|^2 = \|g_k\|^2 - 2\beta_k g_k^T d_{k-1} + \beta_k^2 \|d_{k-1}\|^2. \quad (21)$$

Dividing Equation (21) by $\|g_k\|^4$ yields

$$\frac{\|d_k\|^2}{\|g_k\|^4} = \frac{1}{\|g_k\|^2} - \frac{2\beta_k g_k^T d_{k-1}}{\|g_k\|^4} + \frac{\beta_k^2 \|d_{k-1}\|^2}{\|g_k\|^4}.$$

Using Equations (5) and (16), we now obtain

$$\frac{\|d_k\|^2}{\|g_k\|^4} \leq \frac{\|d_{k-1}\|^2}{\|g_{k-1}\|^4} + \frac{1}{\|g_k\|^2} + \frac{2\sigma |g_{k-1}^T d_{k-1}|}{\|g_{k-1}\|^2 \|g_k\|^2} \leq \frac{\|d_{k-1}\|^2}{\|g_{k-1}\|^4} + \frac{1 + 2\sigma(2-c)}{\|g_k\|^2}. \quad (22)$$

Repeating the process for (22) using the relationship $\frac{1}{\|g_1\|} = \frac{1}{\|d_1\|}$ yields

$$\frac{\|d_k\|^2}{\|g_k\|^4} \leq (1 + 2\sigma(2-c)) \sum_{i=1}^k \frac{1}{\|g_i\|^2}$$

From (20), we obtain

$$\frac{\|g_k\|^4}{\|d_k\|^2} \geq \frac{\varepsilon^2}{k(1 + 2\sigma(2-c))}$$

Therefore,

$$\sum_{k=0}^{\infty} \frac{\|g_k\|^4}{\|d_k\|^2} = \infty.$$

□

Theorem 3. Suppose the sequences $\{x_k\}$ and $\{d_k\}$ are obtained by using (1), (2), and β_k^{DL-HS} , where α_k is computed via the SWP line search (4) and (5). Then, the descent condition holds.

Proof. Let

$$\beta_k^{DL-HS} = -\mu_k \frac{g_k^T s_{k-1}}{d_{k-1}^T y_{k-1}}.$$

By multiplying (2) by g_k^T , and substituting β_k^{DL-HS} , we obtain

$$g_k^T d_k = -\|g_k\|^2 - \mu_k \frac{\alpha_{k-1} g_k^T d_{k-1}}{d_{k-1}^T y_{k-1}} g_k^T d_{k-1}.$$

$$g_k^T d_k = -\|g_k\|^2 - \mu_k \frac{\alpha_{k-1} \|g_k^T d_{k-1}\|^2}{d_{k-1}^T y_{k-1}} < 0.$$

This completes the proof. □

The condition in (23) is called Zoutendijk condition [37], acting as an important role in proving convergence properties of the CG method.

Lemma 1. Assume that Assumption 1 holds. Consider any form of (1) and (2) with step size α_k satisfying the WWP line search, where the search direction d_k is descent. Then, we have

$$\sum_{k=0}^{\infty} \frac{(g_k^T d_k)^2}{\|d_k\|^2} < \infty. \quad (23)$$

Moreover, (23) holds for exact and SWP line searches. Substituting (7) into (18), it follows that

$$\sum_{k=0}^{\infty} \frac{\|g_k\|^4}{\|d_k\|^2} < \infty. \quad (24)$$

Theorem 4. Assume that Assumption 1 holds. Consider the conjugate gradient method in (1) and (2) with β_k^{DL-HS} , where d_k is a descent direction and α_k is obtained by the strong Wolfe line search. Then, the $\liminf_{k \rightarrow \infty} \|g_k\| = 0$.

Proof. We will prove this theorem by contradiction. Suppose Theorem 4 is not true. Then, a constant $\varepsilon > 0$ exists such that

$$\|g_k\| \geq \varepsilon, \forall k \geq 1. \quad (25)$$

By squaring both sides of (2), we obtain

$$\begin{aligned} \|d_k\|^2 &= \|g_k\|^2 - 2\beta_k g_k^T d_{k-1} + \beta_k^2 \|d_{k-1}\|^2 \\ &\leq \|g_k\|^2 + 2|\beta_k| g_k^T d_{k-1} + \beta_k^2 \|d_{k-1}\|^2 \\ &\leq \|g_k\|^2 + \frac{2}{L} \frac{\|g_k\| \|s_k\|}{(1-\sigma) g_{k-1}^T d_{k-1}} (\sigma) g_{k-1}^T d_{k-1} + \frac{1}{L^2} \frac{((\sigma) g_{k-1}^T d_{k-1})^2 \|s_{k-1}\|^2}{(1-\sigma) g_{k-1}^T d_{k-1}} \\ &\leq \|g_k\|^2 + \frac{2}{L} \frac{\|g_k\| \|s_k\|}{(1-\sigma)} \sigma + \frac{1}{L^2} \frac{(\sigma)^2 \|s_{k-1}\|^2}{(1-\sigma)^2} \\ \frac{\|d_k\|^2}{\|g_k\|^4} &\leq \frac{\|g_k\|^2}{\|g_k\|^4} + \frac{2}{L} \frac{\|g_k\| \|s_k\|}{(1-\sigma) \|g_k\|^4} \sigma + \frac{1}{L^2} \frac{\sigma^2 \|s_{k-1}\|^2}{(1-\sigma)^2 \|g_k\|^4} \\ &\leq \frac{1}{\|g_k\|^2} + \frac{2}{L} \frac{\|g_k\| \|s_k\|}{(1-\sigma) \|g_k\|^4} \sigma + \frac{1}{L^2} \frac{\sigma^2 \|s_{k-1}\|^2}{(1-\sigma)^2 \|g_k\|^4} \\ &\leq \frac{1}{\|g_k\|^2} + \frac{2}{L} \frac{\|s_k\|}{(1-\sigma) \|g_k\|^3} \sigma + \frac{1}{L^2} \frac{\sigma^2 \|s_{k-1}\|^2}{(1-\sigma)^2 \|g_k\|^4}. \end{aligned}$$

Let

$$\|g_k\|^m = \min\{\|g_k\|^2, \|g_k\|^3, \|g_k\|^4\}, m \in N$$

then

$$\frac{\|d_k\|^2}{\|g_k\|^4} \leq \frac{1}{\|g_k\|^m} \left(1 + \frac{2}{L} \frac{\lambda}{(1-\sigma)} \sigma + \frac{1}{\lambda^2} \frac{\sigma^2 \lambda^2}{(1-\sigma)^2} \right).$$

Also, let

$$R = \left(1 + \frac{2}{L} \frac{\Gamma}{(1-\sigma)} \sigma + \frac{1}{\lambda^2} \frac{\sigma^2 \Gamma^2}{(1-\sigma)^2} \right)$$

then

$$\frac{\|d_k\|^2}{\|g_k\|^4} \leq \frac{R}{\|g_k\|^m} \leq R \sum_{i=1}^k \frac{1}{\|g_i\|^m} \frac{\|g_k\|^4}{\|d_k\|^2} \geq \frac{\varepsilon^m}{kR}.$$

Therefore,

$$\sum_{k=0}^{\infty} \frac{\|g_k\|^4}{\|d_k\|^2} = \infty.$$

This result contradicts (24). Therefore, $\liminf_{k \rightarrow \infty} \|g_k\| = 0$, completing the proof. $\square$

4. Numerical Results and Discussion

To improve the efficiency of the proposed CG method, we modified Equation (16) as follows:

$$\beta_k^{LS+} = \begin{cases} -\frac{\|g_k\|^2 - |g_k^T g_{k-1}|}{d_{k-1}^T g_{k-1}} & \text{if } \|g_k\|^2 > |g_k^T g_{k-1}| \\ -\frac{\|g_k\|^2 - \mu_k |g_k^T g_{k-1}|}{d_{k-1}^T g_{k-1}} & \text{if } \|g_k\|^2 > \mu_k |g_k^T g_{k-1}| \\ -\mu_k \frac{g_k^T s_{k-1}}{d_{k-1}^T y_{k-1}} & \text{otherwise} \end{cases}$$

To analyze the efficiency of the new formula, we selected several test problems in Appendix A from CUTer [38]. We compared the CG-Descent, LS, DL+ coefficients based on the CPU time and the number of iterations. We employed the modified CG descent 6.8 [39] with the SWP line search with $\delta = 0.01$, $\sigma = 0.1$ and memory = 0 for LS, LS+, DL+ algorithms. To obtain the result of CG-Descent 5.3, we employ the CG-descent 6.3 with memory = 0. The norm of the gradient was employed as the stopping criterion, specifically $\|g_k\| \leq 10^{-6}$ for all algorithms. The host computer is an AMD A4 and 4GB of RAM. The

results are shown in Figures 1 and 2. Note that a performance measure introduced by Dolan and Moré [40] was employed. This performance measure was introduced to compare a set of solvers S on a set of problems ρ . Assuming n_s solvers and n_p problems in S and ρ , respectively, the measure $t_{p,s}$ is defined as the computation time (e.g., the number of iterations or the CPU time) required for the solver s to solve the problem p .

To create a baseline for comparison, the performance of the solver s on the problem p is scaled by the best performance of any solver in S on the problem using the ratio

$$r_{p,s} = \frac{t_{p,s}}{\min\{t_{p,s} : s \in S\}}$$

Let the parameter $r_M \geq r_{p,s}$ for all p, s be selected. A further assumption made is that $r_{p,s} = r_M$ if and only if the solver s does not solve the problem p . As we would like to obtain an overall assessment of the performance of a solver, we defined the following measure

$$P_s(t) = \frac{1}{n_p} \text{size}\{p \in \rho : r_{p,s} \leq t\}$$

Thus, $P_s(t)$ is the probability for a solver $s \in S$ that the performance ratio $r_{p,s}$ is within a factor $t \in \mathbb{R}$ of the best possible ratio. Suppose we define the function p_s as the cumulative distribution function for the performance ratio. In that case, the performance measure $P_s : \mathbb{R} \rightarrow [0, 1]$ for a solver is non-decreasing and piecewise continuous from the right. The value of $p_s(1)$ is the probability that the solver achieves the best performance of all the solvers. In general, a solver with high values of $P_s(t)$, which would appear in the upper right corner of the figure, is preferable.

Figure 1. Performance measure based on the number of iterations.

Figure 2. Performance measure based on the CPU time.

Figure 1 shows that LS+ outperforms CG-Descent 5.3, LS, and DL+ in the number of iterations. It is worth noting that LS fails to obtain the stationary point for some functions. Thus, it solves around 95% of test functions. According to Figure 2, which presents the CPU time, we note that the LS+ also outperforms CG-Descent, LS, and DL+. Also, we can note that CG-Descent outperforms DL+ and LS+ in both figures. Thus, the proposed method is more efficient than DL+ and CG-Descent and more efficient and robust than the original LS. Also, it is worth noting that the proposed method can be extended to a three term CG method by using a modified DL+ CG formula, which is our new paper in near future.

5. Conclusions

In this paper, we proposed a modification of the Liu and Storey CG method that satisfies the following main challenges:

1. The proposed method is a non-negative two-term CG method.
2. The proposed method satisfies the descent condition.
3. The convergences properties are satisfied by obtaining the stationary point/s.
4. The modified method was restarted based on the suggestion presented by [12] instead of using the steepest descent, improving the efficiency of the proposed method.
5. The numerical results show that the new method outperformed CG-Descent, DL+, and LS in terms of the number of iteration and CPU time. Moreover, the modified method is more robust than the original LS.

In future, we will try to improve the line search to reduce the number of function evaluations and the number of gradient evaluations. As an application of the CG method in image restoration, the reader can refer to [41]. In addition, we will focus on some applications of the CG method, such as machine learning, deep learning, and image restoration. The proposed method is of great interest in solving nonlinear coefficient

inverse problems for partial differential equations, for more, the reader can refer to the following references [42–47].

Author Contributions: Conceptualization, Z.S.; Supervision, G.A. and I.M.; methodology, A.A.; Software and original draft preparation. All authors have read and agreed to the published version of the manuscript.

Funding: This study has been partially supported by the universiti Malaysia Terengganu, Center of Research and Innovation Management.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All data generated and analyzed during this study are included within the article.

Acknowledgments: The authors are grateful to the editor and the anonymous reviewers for their valuable comments and suggestions, which improved our paper. We would like to thank William W. Hager for publishing his code for implementing the CG method.

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

Appendix A

Table A1. The set of test functions.

Function	Dimension	No. Iterations	LS+			No. Iterations	CG-Descent		
			No. of Function Evaluations	No. Gradient Evaluations	CPU Time		No. Function Evaluations	No. Gradient Evaluations	CPU Time
AKIVA	2	8	20	15	0.02	10	21	11	0.02
ALLINITU	4	11	28	20	0.02	12	29	18	0.02
ARGLINA	200	1	3	2	0.02	1	3	2	0.02
ARWHEAD	200	7	16	12	0.02	7	15	8	0.02
BARD	5000	14	36	25	0.02	16	33	17	0.02
BDQRTIC	3	140	299	256	0.44	136	273	237	0.58
BEALE	5000	12	30	22	0.02	15	31	16	0.02
BIGGS6	2	23	55	37	0.02	27	57	31	0.02
BOX3	6	10	23	14	0.02	11	24	13	0.02
BOX	3	7	24	20	0.11	8	25	19	0.08
BRKMCC	1000	5	11	6	0.02	5	11	6	0.02
BROWNAL	2	6	19	15	0.02	9	25	18	0.02
BROWNBS	200	12	27	19	0.02	13	26	15	0.02
BROWNDEN	2	16	38	31	0.02	16	31	19	0.02
BROYDN7D	4	66	120	90	0.31	1411	2810	1429	5.47
BRYBND	5000	38	99	69	0.2	85	174	90	0.38
CHAINWOO	5000	257	533	312	0.67	318	619	373	0.866
CHNROSNB	4000	298	596	317	0.02	287	564	299	0.02
CLIFF	50	9	43	36	0.02	18	70	54	0.02
COSINE	2	19	62	50	0.25	11	39	32	0.19
CRAGGLVY	10,000	91	182	142	0.37	103	197	147	0.45
CUBE	5000	17	43	30	0.02	32	77	47	0.02
CURLY10	2	56,628	77,977	92,037	202	47,808	67,294	76,156	173.7
CURLY20	10,000	78,784	101,319	135,082	426.97	66,587	89,245	110,540	383.94
CURLY30	10,000	84,712	111,549	142,856	637	79,030	102,516	134,682	639.63
DECONVU	10,000	181	415	235	0.02	400	801	401	0.02
DENSCHNA	63	6	16	12	0.02	9	19	10	0.02

Table A1. Cont.

Function	Dimension	LS+				CG-Descent			
		No. Iterations	No. of Function Evaluations	No. Gradient Evaluations	CPU Time	No. Iterations	No. Function Evaluations	No. Gradient Evaluations	CPU Time
DENSCHNB	2	6	18	15	0.02	7	15	8	0.02
DENSCHNC	2	11	36	31	0.02	12	26	14	0.02
DENSCHND	2	17	45	34	0.02	47	98	51	0.02
DENSCHNE	3	14	44	36	0.02	18	49	32	0.02
DENSCHNF	3	9	28	23	0.02	8	17	9	0.02
DIXMAANA	2	6	16	13	0.02	7	15	8	0.02
DIXMAANB	3000	8	19	13	0.02	6	13	7	0.02
DIXMAANC	3000	6	14	9	0.02	6	13	7	0.02
DIXMAAND	3000	11	26	17	0.02	7	15	8	0.02
DIXMAANE	3000	275	373	627	0.42	222	239	429	0.33
DIXMAANF	3000	44	140	106	0.06	161	323	162	0.13
DIXMAANG	3000	116	302	274	0.2	157	315	158	0.12
DIXMAANH	3000	446	695	1038	0.78	173	347	174	0.22
DIXMAANI	3000	525	670	1075	0.63	3856	3926	7644	4.25
DIXMAANJ	3000	59	166	117	0.08	327	655	328	0.36
DIXMAANK	3000	86	224	151	0.16	283	567	284	0.28
DIXMAANL	3000	62	175	127	0.08	237	475	238	0.2
DIXON3DQ	3000	333	401	725	0.78	10,000	10,007	19,995	19.12
DJTL	10,000	82	1203	1176	0.02	82	917	880	0.02
DQDRTIC	2	5	11	6	0.02	5	11	6	0.02
DQRTIC	5000	15	32	18	0.02	17	37	21	0.03
EDENSCH	5000	26	56	45	0.03	26	52	38	0.03
EG2	2000	6	13	7	0.02	5	11	6	0.02
EIGENALS	1000	9492	17,167	11,327	166	10,083	18,020	12,244	178.36
EIGENBLS	2550	24,630	49,386	24,810	373	15,301	30,603	15,302	237
EIGENCLS	2550	14,468	28,958	14,501	228.25	10,136	19,292	11,118	174.19
ENGVAL1	2652	22	47	35	0.05	27	50	36	0.06
ENGVAL2	5000	24	64	48	0.02	26	61	37	0.02
ERRINROS	3	3358	6774	3512	0.08	380	773	425	0.02

Table A1. Cont.

Function	Dimension	LS+				CG-Descent			
		No. Iterations	No. of Function Evaluations	No. Gradient Evaluations	CPU Time	No. Iterations	No. Function Evaluations	No. Gradient Evaluations	CPU Time
EXPFIT	50	10	27	21	0.02	13	29	16	0.02
EXTROSNB	2	2113	4570	2607	0.8	3808	7759	3982	1.25
FLETGBV2	1000	1	1	1	0.02	1	1	1	0.02
FLETCHCR	5000	264	517	287	0.09	152	290	176	0.05
FMINSRF2	1000	6	116	112	0.25	346	693	347	1.09
FMINSURF	5625	6	116	112	0.22	473	947	474	1.51
FREUROTH	5625	21	51	40	0.09	25	51	38	0.11
GENHUMPS	5000	2	109	108	0.28	6475	12,964	6493	20.11
GENROSE	5000	1149	2350	1244	0.25	1078	2167	1101	0.17
GROWTHLS	500	113	413	347	0.02	156	456	319	0.02
GULF	3	37	94	65	0.02	37	84	48	0.02
HAIRY	3	8	19	11	0.02	36	99	65	0.02
HATFLDD	2	15	43	35	0.02	20	43	24	0.02
HATFLDE	3	12	32	25	0.02	30	72	45	0.02
HATFLDFL	3	76	223	168	0.02	39	92	55	0.02
HEART6LS	3	391	1096	822	0.02	684	1576	941	0.02
HEART8LS	6	236	570	378	0.02	249	524	278	0.02
HELIX	8	26	59	37	0.02	23	49	27	0.02
HIELOW	3	13	31	21	0.05	14	30	16	0.02
HILBERTA	3	2	5	3	0.02	2	5	3	0.02
HILBERTB	2	4	9	5	0.02	4	9	5	0.02
HIMMELBB	10	4	18	18	0.02	10	28	21	0.02
HIMMELBF	2	22	56	40	0.02	26	60	36	0.02
HIMMELBG	4	8	23	17	0.02	8	20	13	0.02
HIMMELBH	2	5	13	9	0.02	7	16	9	0.02
HUMPS	2	85	371	319	0.02	52	186	144	0.02
JENSMP	2	15	52	46	0.02	15	33	22	0.02
JIMACK	2	8306	16,613	8307	1168.56	8314	16,629	8315	1182.25
KOWOSB	35,449	17	44	32	0.02	17	39	23	0.02

Table A1. Cont.

Function	Dimension	LS+				CG-Descent			
		No. Iterations	No. of Function Evaluations	No. Gradient Evaluations	CPU Time	No. Iterations	No. Function Evaluations	No. Gradient Evaluations	CPU Time
LIARWHD	4	16	49	40	0.05	21	45	25	0.03
LOGHAIRY	5000	43	155	127	0.02	27	81	58	0.02
MANCINO	2	11	23	12	0.08	11	23	12	0.08
MARATOSB	100	779	3264	2857	0.02	1145	3657	2779	0.02
MEXHAT	2	14	54	50	0	20	56	39	0.02
MOREBV	2	161	168	317	0.3	161	168	317	0.41
MSQRTALS	5000	1024	3224	6454	3232	2905	5815	2911	8.64
MSQRTBLS	1024	2414	4705	2548	7.53	2280	4525	2326	6.91
NCB20B	1024	3086	4922	6405	48.9	2035	4694	6006	46.36
NCB20	500	742	1712	1056	9.17	879	1511	1463	11.83
NONCVXU2	5010	-	-	-	-	6610	12,833	6999	15.89
NONDIA	5000	8	21	14	0.02	7	25	20	0.03
NONDQUAR	5000	-	-	-	-	1942	3888	1947	2.45
OSBORNEA	5000	67	167	115	0.02	94	213	124	0.02
OSBORNEB	5	58	132	82	0.02	62	127	65	0.02
OSCIPATH	11	296,950	712,333	463,338	2.09	310,990	670,953	367,325	2.08
PALMER1C	10	12	27	28	0.02	11	26	26	0.02
PALMER1D	8	10	24	23	0.02	11	25	25	0.02
PALMER2C	7	11	21	22	0.02	11	21	21	0.02
PALMER3C	8	11	21	21	0.02	11	20	20	0.02
PALMER4C	8	11	21	21	0.02	11	20	20	0.02
PALMER5C	8	6	13	7	0.02	6	13	7	0.02
PALMER6C	6	11	24	24	0.02	11	24	24	0.02
PALMER7C	8	11	20	20	0.02	11	20	20	0.02
PALMER8C	8	11	19	19	0.02	11	18	17	0.02
PARKCH	8	682	1406	1194	31.17	672	1385	1128	29.45
PENALTY1	15	25	80	64	0	28	69	44	0.02
PENALTY2	1000	192	228	366	0.03	191	221	354	0.05
PENALTY3	200	99	301	240	1.94	99	285	219	1.78

Table A1. Cont.

Function	Dimension	LS+				CG-Descent			
		No. Iterations	No. of Function Evaluations	No. Gradient Evaluations	CPU Time	No. Iterations	No. Function Evaluations	No. Gradient Evaluations	CPU Time
POWELLSG	200	34	80	54	0.05	26	53	27	0.02
POWER	5000	347	757	433	0.59	372	754	384	0.76
QUARTC	10,000	15	32	18	0.05	17	37	21	0.03
ROSENBR	5000	28	77	58	0.02	34	77	44	0.02
S308	2	7	21	17	0	8	19	12	0.02
SCHMVETT	2	201	549	773	2.44	43	73	60	0.23
SENSORS	5000	38	90	59	0.5	21	50	34	0.25
SINEVAL	100	52	145	105	0.02	64	144	88	0.02
SINQUAD	2	15	45	39	0.09	14	40	33	0.09
SISSER	5000	5	19	19	0.02	6	18	14	0.02
SNAIL	2	17	47	33	0.02	100	230	132	0.02
SPARSINE	2	18,177	18,446	36,089	71.8	18,358	18,647	36,431	73
SPARSQUR	5000	85	267	265	0.31	28	61	35	0.31
SPMSRTLS	10,000	209	428	223	0.5	203	412	210	0.59
SROSENBR	4999	9	22	15	0.02	11	23	12	0.02
STRATEC	5000	166	374	239	5.19	462	1043	796	19.98
TESTQUAD	10	1494	1501	2983	1.25	1577	1584	3149	1.52
TOINTGOR	5000	138	237	179	0.02	135	233	174	0.02
TOINTGSS	50	3	7	4	0.02	4	9	5	0.02
TOINTPSP	5000	114	243	176	0.02	143	279	182	0.02
TOINTQOR	50	29	36	53	0.02	29	36	53	0.02
TQUARTIC	50	13	43	36	0.03	14	40	27	0.03
TRIDIA	5000	781	788	1557	0.88	782	789	1559	0.84
VARDIM	5000	11	26	20	0.02	11	25	16	0.02
VAREIGVL	-	24	51	29	0.02	23	47	24	0.02
VIBRBEAM	50	135	337	267	0.02	138	323	199	0.02
WATSON	8	57	129	83	0.02	49	102	54	0.02
WOODS	12	23	55	36	0.02	22	51	30	0.06
YFITU	4000	69	200	157	0.02	84	197	118	0.02
ZANGWIL2	3	1	3	2	0.02	1	3	2	0.02

References

1. Wolfe, P. Convergence Conditions for Ascent Methods. *SIAM Rev.* **1969**, *11*, 226–235. [\[CrossRef\]](#)
2. Wolfe, P. Convergence Conditions for Ascent Methods. II: Some Corrections. *SIAM Rev.* **1971**, *13*, 185–188. [\[CrossRef\]](#)
3. Hestenes, M.; Stiefel, E. Methods of conjugate gradients for solving linear systems. *J. Res. Natl. Inst. Stand. Technol.* **1952**, *49*, 409. [\[CrossRef\]](#)
4. Elijah, P.; Ribiere, G. Note sur la convergence de méthodes de directions conjuguées. *ESAIM: Math. Model. Numer. Anal. Modélisation Mathématique Et Anal. Numérique* **1969**, *3*, R1, 35–43.
5. Liu, Y.; Storey, C. Efficient generalized conjugate gradient algorithms, part 1: Theory. *J. Optim. Theory Appl.* **1991**, *69*, 129–137. [\[CrossRef\]](#)
6. Fletcher, R.; Reeves, C.M. Function minimization by conjugate gradients. *Comput. J.* **1964**, *7*, 149–154. [\[CrossRef\]](#)
7. Fletcher, R. *Practical Methods of Optimization, Unconstrained Optimization*; Wiley: New York, NY, USA, 1987.
8. Dai, Y.H.; Yuan, Y. A non-linear conjugate gradient method with a strong global convergence property. *SIAM J. Optim.* **1999**, *10*, 177–182. [\[CrossRef\]](#)
9. Powell, M.J.D. Non-convex minimization calculations and the conjugate gradient method, Numerical Analysis (Dundee, 1983). In *Lecture Notes in Mathematics*; Springer: Berlin, Germany, 1984; Volume 1066, pp. 122–141.
10. Gilbert, J.C.; Nocedal, J. Global convergence properties of conjugate gradient methods for optimization. *Siam. J. Optim.* **1992**, *2*, 21–42. [\[CrossRef\]](#)
11. Al-Baali, M. Descent property and global convergence of the Fletcher-Reeves method with inexact line search. *IMA J. Numer. Anal.* **1985**, *5*, 121–124. [\[CrossRef\]](#)
12. Dai, Y.H.; Liao, L.Z. New conjugacy conditions and related non-linear conjugate gradient methods. *Appl. Math. Optim.* **2001**, *43*, 87–101. [\[CrossRef\]](#)
13. Hager, W.W.; Zhang, H. A new conjugate gradient method with guaranteed descent and an efficient line search. *Siam. J. Optim.* **2005**, *16*, 170–192. [\[CrossRef\]](#)
14. Hager, W.W.; Zhang, H. The limited memory conjugate gradient method. *Siam. J. Optim.* **2013**, *23*, 2150–2168. [\[CrossRef\]](#)
15. Zhang, J.Z.; Deng, N.Y.; Chen, L.H. New quasi-Newton equation and related methods for unconstrained optimization. *J. Optim. Theory Appl.* **1999**, *102*, 147–167. [\[CrossRef\]](#)
16. Zhang, J.Z.; Xu, C.X. Properties and numerical performance of quasi-Newton methods with modified quasi-Newton equation. *J. Comput. Appl. Math.* **2001**, *137*, 269–278. [\[CrossRef\]](#)
17. Yabe, H.; Takano, M. Global convergence properties of non-linear conjugate gradient methods with modified secant relation. *Comput. Optim. Appl.* **2004**, *28*, 203–225. [\[CrossRef\]](#)
18. Dehghani, R.; Bidabadi, N.; Fahs, H.; Hosseini, M.M. A Conjugate Gradient Method Based on a Modified Secant Relation for Unconstrained Optimization. *Numer. Funct. Anal. Optim.* **2020**, *41*, 621–634. [\[CrossRef\]](#)
19. Jiang, X.; Jian, J.; Song, D.; Liu, P. An improved Polak–Ribière–Polyak conjugate gradient method with an efficient restart direction. *Comput. Appl. Math.* **2021**, *40*, 1–24. [\[CrossRef\]](#)
20. Wei, Z.; Yao, S.; Liu, L. The convergence properties of some new conjugate gradient methods. *Appl. Math. Comput.* **2006**, *183*, 1341–1350. [\[CrossRef\]](#)
21. Qu, A.; Li, M.; Xiao, Y.; Liu, J. A modified Polak–Ribière–Polyak descent method for unconstrained optimization. *Optim. Methods Softw.* **2014**, *29*, 177v188. [\[CrossRef\]](#)
22. Shengwei, Y.; Wei, Z.; Huang, H. A note about WYL’s conjugate gradient method and its applications. *Appl. Math. Comput.* **2007**, *191*, 381–388. [\[CrossRef\]](#)
23. Zhang, L. An improved Wei–Yao–Liu non-linear conjugate gradient method for optimization computation. *Appl. Math. Comput.* **2009**, *215*, 2269–2274.
24. Al-Baali, M.; Narushima, Y.; Yabe, H. A family of three-term conjugate gradient methods with sufficient descent property for unconstrained optimization. *Comput. Optim. Appl.* **2015**, *60*, 89–110. [\[CrossRef\]](#)
25. Powell, M.J.D. Restart procedures for the conjugate gradient method. *Math. Program.* **1977**, *12*, 241–254. [\[CrossRef\]](#)
26. Beale, E.M.L. A derivative of conjugate gradients. In *Numerical Methods for Nonlinear Optimization*; Lootsma, F.A., Ed.; Academic Press: London, UK, 1972; pp. 39–43.
27. Dai, Y.; Yuan, Y. Convergence properties of Beale–Powell restart algorithm. *Sci. China Ser. A Math.* **1998**, *41*, 1142–1150. [\[CrossRef\]](#)
28. Alhawarat, A.; Salleh, Z.; Mamat, M.; Rivaie, M. An efficient modified Polak–Ribière–Polyak conjugate gradient method with global convergence properties. *Optim. Methods Softw.* **2017**, *32*, 1299–1312. [\[CrossRef\]](#)
29. Kaelo, P.; Mtangulwa, P.; Thuto, M.V. A globally convergent hybrid conjugate gradient method with strong Wolfe conditions for unconstrained optimization. *Math. Sci.* **2019**, *14*, 1–9. [\[CrossRef\]](#)
30. Liu, J.K.; Xu, J.L.; Zhang, L.Q. Partially symmetrical derivative-free Liu–Storey projection method for convex constrained equations. *Int. J. Comput. Math.* **2019**, *96*, 1787–1798. [\[CrossRef\]](#)
31. Gao, P.; He, C.; Liu, Y. An adaptive family of projection methods for constrained monotone non-linear equations with applications. *Appl. Math. Comput.* **2019**, *359*, 1–16.

32. Zheng, L.; Yang, L.; Liang, Y. A Modified Spectral Gradient Projection Method for Solving Non-linear Monotone Equations with Convex Constraints and Its Application. *IEEE Access* **2020**, 9, 92677–92686. [\[CrossRef\]](#)
33. Zheng, L.; Yang, L.; Liang, Y. A conjugate gradient projection method for solving equations with convex constraints. *J. Comput. Appl. Math.* **2020**, 375, 112781. [\[CrossRef\]](#)
34. Benrabia, N.; Laskri, Y.; Guebbai, H.; Al-Baali, M. Applying the Powell's Symmetrical Technique to Conjugate Gradient Methods with the Generalized Conjugacy Condition. *Numer. Funct. Anal. Optim.* **2016**, 37, 839–849. [\[CrossRef\]](#)
35. Klivanov, M.V.; Li, J.; Zhang, W. Convexification for an inverse parabolic problem. *Inverse Probl.* **2020**, 36, 085008. [\[CrossRef\]](#)
36. Beilina, L.; Klivanov, M.V. A Globally Convergent Numerical Method for a Coefficient Inverse Problem. *SIAM J. Sci. Comput.* **2008**, 31, 478–509. [\[CrossRef\]](#)
37. Zoutendijk, G. Non-linear programming, computational methods. *Integer Nonlinear Program* **1970**, 143, 37–86.
38. Bongartz, I.; Conn, A.R.; Gould, N.; Toint, P.L. CUTE: Constrained and unconstrained testing environment, *ACM Trans. Math. Softw.* **1995**, 21, 123–160. [\[CrossRef\]](#)
39. Available online: <http://users.clas.ufl.edu/hager/papers/Software/> (accessed on 20 May 2021).
40. Dolan, E.D.; Moré, J.J. Benchmarking optimization software with performance profiles. *Math. Program.* **2002**, 91, 201–213. [\[CrossRef\]](#)
41. Alhawarat, A.; Salleh, Z.; Masmali, I.A. A Convex Combination between Two Different Search Directions of Conjugate Gradient Method and Application in Image Restoration. *Math. Probl. Eng.* **2021**, 2021. [\[CrossRef\]](#)
42. Kaltenbacher, B.; Rundell, W. The inverse problem of reconstructing reaction–diffusion systems. *Inverse Probl.* **2020**, 36, 065011. [\[CrossRef\]](#)
43. Kaltenbacher, B.; Rundell, W. On the identification of a nonlinear term in a reaction–diffusion equation. *Inverse Probl.* **2019**, 35, 115007. [\[CrossRef\]](#)
44. Averós, J.C.; Llorens, J.P.; Uribe-Kaffure, R. Numerical Simulation of Non-Linear Models of Reaction—Diffusion for a DGT Sensor. *Algorithms* **2020**, 13, 98. [\[CrossRef\]](#)
45. Lukyanenko, D.; Yeleskina, T.; Prigorniy, I.; Isaev, T.; Borzunov, A.; Shishlenin, M. Inverse Problem of Recovering the Initial Condition for a Nonlinear Equation of the Reaction–Diffusion–Advection Type by Data Given on the Position of a Reaction Front with a Time Delay. *Mathematics* **2021**, 9, 342. [\[CrossRef\]](#)
46. Lukyanenko, D.V.; Borzunov, A.A.; Shishlenin, M.A. Solving coefficient inverse problems for nonlinear singularly perturbed equations of the reaction-diffusion-advection type with data on the position of a reaction front. *Commun. Nonlinear Sci. Numer. Simul.* **2021**, 99, 105824. [\[CrossRef\]](#)
47. Egger, H.; Engl, H.W.; Klivanov, M.V. Global uniqueness and Hölder stability for recovering a nonlinear source term in a parabolic equation. *Inverse Probl.* **2004**, 21, 271–290. [\[CrossRef\]](#)