Transactions on Soft Computing

www.tsc.reapress.com

Trans. Soft. Comput. Vol. 1, No. 2 (2025) 109–126.

Paper Type: Original Article

A Spectral Hybrid Conjugate Gradient Approach for Large-Scale Unconstrained Optimization with Applications in Image Restoration

Meysam Ranjbar[1,*], Ali Ashrafi[1]

[1]Department of Mathematics, Faculty of Mathematics, Statistics and Computer Science, Semnan University, Semnan, Iran; ranjbar.meysam.73@semnan.ac.ir; a_ashrafi@semnan.ac.ir.

Citation:

Received: 15 March 2024 *Revised: 12 May 2024* *Accepted: 15 June 2024*	Ranjbar, M., & Ashrafi, A. (2025). A spectral hybrid conjugate gradient approach for large-scale unconstrained optimization with applications in image restoration. *Transactions on soft computing*, 1(2), 109-126.

Abstract

This paper introduces a Spectral Hybrid Conjugate Gradient (SHCG) method for large-scale unconstrained optimization. By modifying the Hestenes–Stiefel formula with a correction term and a safeguarded denominator, and embedding it into a spectral framework, the proposed method combines the benefits of hybridization and spectral scaling. The resulting algorithm guarantees sufficient descent independently of line search and ensures global convergence under standard assumptions. Numerical experiments, including applications to image restoration, demonstrate that the SHCG method achieves faster convergence and higher robustness compared with several existing Conjugate Gradient (CG) algorithms.

Keywords: Unconstrained optimization, Spectral conjugate gradient method, Hybrid conjugate gradient method, Global convergence, Image restoration.

1 | Introduction

In this paper, we consider the following unconstrained optimization problem:

$$\min_{x\in\mathbb{R}^n} f(x), \tag{1}$$

in which $f: \mathbb{R}^n \to \mathbb{R}$ is a smooth function.

The Conjugate Gradient (CG) method is one of the most popular iterative algorithms for solving large-scale unconstrained optimization problems due to its simplicity, low memory requirements, and favorable convergence properties. In particular, nonlinear CG methods are designed to ensure global convergence under mild assumptions. Originally proposed by Hestenes and Stiefel for quadratic functions and later

Corresponding Author: ranjbar.meysam.73@semnan.ac.ir

https://doi.org/10.48314/tsc.v1i2.45

Licensee System Analytics. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0).

extended to the nonlinear case by Fletcher and Reeves, CG methods have found wide applications in areas such as signal processing, machine learning, and scientific computing. In general, the CG method generates the next iterate according to

$$x_{k+1} = x_k + \alpha_k d_k, \tag{2}$$

in which x_k is the current iteration point, α_k is a step size obtained using an inexact line search, and d_k is the search direction defined by the following recurrence relation:

$$d_{k+1} = \begin{cases} -g_{k+1}, & \text{if } k = 0, \\ -g_{k+1} + \beta_k d_k, & \text{if } k \geq 1, \end{cases} \tag{3}$$

where g_k denotes $\nabla f(x_k)$ and β_k is the conjugate parameter. One of the most popular techniques for determining the step size α_k is the Wolfe line search conditions:

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \delta \alpha_k g_k^T d_k. \tag{4}$$

$$g(x_k + \alpha_k d_k)^T d_k \geq \sigma g_k^T d_k, \tag{5}$$

in which $0 < \delta < \sigma < 1$. The goal of the strong Wolfe line search is to produce a step size α_k that satisfies *Eq. (4)* and the inequality that follows

$$\left| g(x_k + \alpha_k d_k)^T d_k \right| \leq \sigma \left| g_k^T d_k \right|, \tag{6}$$

where $\delta \in (0, \frac{1}{2})$ and $\sigma \in (\delta, 1)$ [1].

Over the years, numerous variants of CG methods have been developed, each characterized by a distinct choice of the conjugate parameter β_k. Classical examples include the formulas of Fletcher–Reeves (FR) [2], Polak–Ribière–Polyak (PRP) [3], Hestenes–Stiefel (HS) [4], Liu–Storey (LS) [5], Dai–Yuan (DY) [6], and Conjugate Descent (CD) [7]. The following is a list of their respective specialised expressions:

$$\beta_k^{HS} = \frac{g_{k+1}^T y_k}{d_k^T y_k}, \beta_k^{FR} = \frac{\|g_{k+1}\|^2}{\|g_k\|^2}, \beta_k^{PRP} = \frac{g_{k+1}^T y_k}{\|g_k\|^2},$$

$$\beta_k^{DY} = \frac{\|g_{k+1}\|^2}{d_k^T y_k}, \beta_k^{CD} = \frac{\|g_{k+1}\|^2}{-d_k^T g_k}, \beta_k^{LS} = \frac{g_{k+1}^T y_k}{-d_k^T g_k},$$

in which $y_k = g_{k+1} - g_k$ and $\|\cdot\|$ denote the Euclidean norm. although for a strictly convex objective function, all the above methods are equivalent. However, for general objective functions, these methods exhibit distinct convergence behaviors: FR, DY, and CD are known for their global convergence guarantees, while HS, PRP, and LS often demonstrate strong numerical performance but may fail to guarantee descent or convergence without modifications.

To address these drawbacks, numerous modifications and hybrid approaches have been proposed. A significant direction in this line of research is the construction of spectral CG methods, motivated by the quadratic convergence property of Newton's method and the efficiency of spectral step-size strategies introduced by Barzilai and Borwein [8]. Spectral methods aim to enhance the theoretical structure of CG algorithms by introducing a scaling factor that improves descent and convergence properties. Recent studies have shown that spectral CG methods can ensure sufficient descent while maintaining global convergence under the strong Wolfe conditions.

Another active line of research is the development of hybrid CG formulas, where different conjugate parameters are combined to inherit the advantages of multiple methods simultaneously. By carefully blending existing CG rules, hybrid methods often achieve a better balance between robustness and efficiency. For instance, A combination of CG techniques from DL and WYL was proposed by Yao and Qin in [8]. Additionally, two hybrid approaches were proposed by Xu and Kong [9]: In the first, DY and HS CG algorithms are combined linearly, and in the second, FR and PRP are combined. Kaelo and Narayanan, and

[10] suggested a CG method that combines the DY and HS approaches, or their variants, in response to the work of Xu and Kong [9]. Abubakr et al. proposed a hybrid three-term CG method with a combination of CG techniques from HS and LS [11]. Using the convex combination of FR and PRP methods, Djordjevic [12] suggested a hybrid CG method, where the conjugacy criteria are satisfied by the search direction, which has been reported to yield improved performance in both theory and practice (see [13–17]).

Zhang [18] presents a hybrid conjugate parameter β_k^{hybrid} for the traditional two-term Dai–Liao CG method:

$$d_{k+1} = -g_{k+1} + \beta_k^{hybrid} d_k,$$

where

$$\beta_k^{hybrid} = \frac{g_{k+1}^T (y_k - t s_k)}{\max\{y_k^T d_k, \|g_k\|^2\}}. \quad (7)$$

Motivated by these developments, this paper proposes a Spectral Hybrid Conjugate Gradient (SHCG) method in which a new hybrid coefficient replaces the classical HS parameter. The hybrid parameter introduces an additional correction term and a safeguarded denominator to enhance descent properties and avoid instability. By embedding this hybrid update into the spectral CG framework, we establish an algorithm that guarantees sufficient descent independent of the line search and enjoys global convergence under standard assumptions.

The main contributions of this work can be summarized as follows:

I. We introduce a hybrid conjugate parameter that modifies the HS formula with an additional correction term and safeguard.

II. We incorporate this hybrid parameter into a spectral CG structure, thereby combining the advantages of hybridization and spectral scaling.

III. We prove that the resulting method ensures sufficient descent and global convergence.

IV. We provide a structured algorithm that can serve as a basis for further numerical investigation.

In the following, we have an overview of the application of the CG method in image restoration. CGMs play a crucial role in the optimization processes of compressive sensing and image restoration, and their integration with deep learning techniques further enhances the efficiency and quality of signal and image restoration. In order to recover images tainted by salt and pepper noise, Cai et al. [19] utilized CGM to minimize a regularization function that preserves edges. A three-term CGM (TTCGM) was proposed by Chen and Zhou [20] to solve the image restoration problems. Malik et al. [21], Cao and Wu [22], Yin et al. [23], and Liu et al. [24] are some examples of more pertinent publications (see also [25–27]).

The remainder of this paper is organized as follows. Section 2 introduces the proposed SHCG algorithm and establishes that, under suitable conditions, it generates search directions that satisfy the sufficient descent property. Section 3 presents the global convergence analysis of the method. In Section 4, we report numerical experiments, including an application to image restoration and Dolan–Moré performance profiles, to compare the proposed algorithm with several existing CG methods. Finally, Section 5 concludes the paper with a summary of findings and directions for future research.

2 | Algorithm Description and Sufficient Descent Property

In this section, we present the proposed algorithm. We begin by defining the search direction d_{k+1}, following the approach introduced in [11]:

$$d_{k+1} = -\lambda_{k+1} g_{k+1} + \beta_k^{hybrid} d_k, \quad \text{for all } k \geq 0, \tag{8}$$

in which $\beta_k^{hybrid} = \frac{g_{k+1}^T (y_k - t s_k)}{\max\{y_k^T d_k, \|g_k\|^2\}}$, and λ_{k+1} is a spectral parameter, where the parameter λ_{k+1} is determined so that the search direction d_{k+1} fulfills the following condition:

$$d_{k+1} = -\nabla^2 f(x_{k+1})^{-1} g_{k+1}, \tag{9}$$

where $\nabla^2 f(x_{k+1})^{-1} \in \mathbb{R}^{n \times n}$ is the Hessian matrix of f at x_{k+1}. In updating *Eq. (8)*, the Newton method requires the Hessian, which contains second-order derivative details. This leads to fast convergence, with quadratic convergence as its main strength. We assume that $\nabla^2 f(x_{k+1})^{-1}$ is available at each iteration and fulfills an appropriate secant condition, for example:

$$\nabla^2 f(x_{k+1}) s_k = y_k. \tag{10}$$

From *Eqs. (8)* and *(9)*, it can be derived that:

$$-\nabla^2 f(x_{k+1})^{-1} g_{k+1} = -\lambda_{k+1} g_{k+1} + \beta_k^{hybrid} d_k. \tag{11}$$

This relation, combined with *Eq. (10)*, shows that the Spectral Newton Method (SNM) takes the following form:

$$-s_k^T g_{k+1} = -\lambda_{k+1} s_k^T \nabla^2 f(x_{k+1}) g_{k+1} + \beta_k^{hybrid} s_k^T \nabla^2 f(x_{k+1}) d_k,$$

$$-s_k^T g_{k+1} = -\lambda_{k+1} y_k^T g_{k+1} + \beta_k^{hybrid} y_k^T d_k.$$

Since any CG method constructed with the Newton structure exhibits only a local property, we replace *Eq. (9)* with the well-known Dai–Liao conjugacy condition to design an algorithm with a global property. Pre-multiplying *Eq. (8)* with y_k^T and combining with $d_{k+1}^T y_k = -t s_k^T g_{k+1}$, we derive the spectral Dai-Liao (SDL) structure as

$$-t s_k^T g_{k+1} = -\lambda_{k+1} y_k^T g_{k+1} + \beta_k^{hybrid} y_k^T d_k.$$

After some algebraic calculations, we get

$$\lambda_{k+1} = \frac{t s_k^T g_{k+1}}{y_k^T g_{k+1}} + \beta_k^{hybrid} \frac{y_k^T d_k}{y_k^T g_{k+1}}.$$

Therefore, to ensure the sufficient descent property, the optimal SHCG is determined as follows.

$$\lambda_{k+1}{}^* = \max\{1, \lambda_{k+1}\} \tag{12}$$

The selection of the parameter t can be determined as follows [28]:

$$t = \frac{\|y_k\|^2}{s_k^T y_k}. \tag{13}$$

Using β_k and d_{k+1} as defined in *Eq. (7)* and *Eq. (8)* respectively, we now present our algorithm.

Algorithm 1. SHCG algorithm.

Step 1. Given a starting point $x_1 \in \mathbb{R}^n$, parameters $0 < \delta < \sigma < 1$ and $\varepsilon > 0$. Set $d_1 = -g_1$ and $k := 1$.

Step 2. If $\|g_k\|_\infty < \varepsilon$, then stop. Otherwise, proceed to *Step 3*.

Step 3. Compute the step size α_k by the standard Wolfe line search strategy *Eqs. (4)* and *(5)*.

Step 4. Compute the spectral parameter $\lambda_{k+1}{}^*$ using *Eq. (12)* with t from *Eq. (13)* and β_k^{hybrid} from *Eq. (7)*.

Step 5. Obtain the search direction d_{k+1} at x_{k+1} by *Eq. (8)*.

Step 6. Generate the next iterate by *Eq. (2)*.

Step 7. Set $k := k + 1$, then proceed to *Step 2*.

The SHCG method calculates the search direction d_{k+1}, and we will now demonstrate its sufficient descent property.

Lemma 1. Le$\{d_k\}$t and $\{g_k\}$ represent the direction sequences that the Algorithm produced, then it holds that

$$g_{k+1}^T d_{k+1} \leq -\varrho \|g_{k+1}\|^2. \tag{14}$$

Proof: by using induction, the conclusion can be demonstrated. Given that $g_0^T d_0 = -\|g_0\|^2$, the conclusion *Eq. (14)* holds for $k = 0$. Next, we will show that for all $k > 0$, the result *Eq. (10)* holds.

Note that using *Eq. (5)* and *Eq. (6)*, we have

$$y_k^T d_k = g_{k+1}^T d_k - g_k^T d_k \geq -(1-\sigma) g_k^T d_k \geq \frac{1-\sigma}{\sigma} \left| g_{k+1}^T d_k \right|.$$

This implies that

$$\frac{\left| g_{k+1}^T d_k \right|}{\left| y_k^T d_k \right|} \leq \frac{\sigma}{1-\sigma}. \tag{15}$$

The preceding inequality is instrumental in demonstrating that the spectral hybrid method satisfies the sufficient descent condition.

From *Eq. (12)*, it follows that $\lambda_{k+1} \geq 1$. By employing the Wolf line search, we have $y_k^T d_k > 0$. Consequently, the denominator in β_k^{hybrid} is strictly positive. Moreover, it holds that

$$\max\{y_k^T d_k, \|g_k\|^2\} \geq y_k^T d_k.$$

In addition, applying Powell's inequality [29] yields the following inequality:

$$\left| g_{k+1}^T g_k \right| \leq 0.2 \|g_{k+1}\|^2. \tag{16}$$

Since $y_k = g_{k+1} - g_k$, using *Eq. (16)*, we get

$$\left| g_{k+1}^T y_k \right| \leq \|g_{k+1}\|^2 + \left| g_{k+1}^T g_k \right| \leq \|g_{k+1}\|^2 + 0.2\|g_{k+1}\|^2 = 1.2\|g_{k+1}\|^2. \tag{17}$$

Pre-multiplying the *Eq. (8)* by g_{k+1}^T yields

$$\begin{aligned}
g_{k+1}^T d_{k+1} &= -\lambda_{k+1}\|g_{k+1}\|^2 + \beta_k^{\text{hybrid}} g_{k+1}^T d_k \\
&= -\lambda_{k+1}\|g_{k+1}\|^2 + \frac{g_{k+1}^T (y_k - t s_k)}{\max\{y_k^T d_k, \|g_k\|^2\}} g_{k+1}^T d_k \\
&= -\lambda_{k+1}\|g_{k+1}\|^2 + \frac{g_{k+1}^T y_k}{\max\{y_k^T d_k, \|g_k\|^2\}} g_{k+1}^T d_k - t\alpha_k \frac{\left(g_{k+1}^T d_k\right)^2}{\max\{y_k^T d_k, \|g_k\|^2\}} \\
&\leq -\lambda_{k+1}\|g_{k+1}\|^2 + \frac{g_{k+1}^T y_k}{y_k^T d_k} g_{k+1}^T d_k \\
&\leq -\lambda_{k+1}\|g_{k+1}\|^2 + \frac{\left| g_{k+1}^T d_k \right|}{\left| y_k^T d_k \right|} \left| g_{k+1}^T y_k \right|.
\end{aligned}$$

By applying the inequality *Eq. (15)* with *Eq. (17)* in the above inequality, we get

$$\leq -\lambda_{k+1}\|g_{k+1}\|^2 + \frac{1.2\sigma}{1-\sigma}\|g_{k+1}\|^2,$$

$$\leq -\left(\lambda_{k+1} - \frac{1.2\sigma}{1-\sigma}\right)\|g_{k+1}\|^2,$$

$$= -\varrho\|g_{k+1}\|^2.$$

To demonstrate the sufficient descent property, the term $\lambda_{k+1} - \frac{1.2\sigma}{1-\sigma}$ must be positive. By setting $\varrho = \lambda_{k+1} - \frac{1.2\sigma}{1-\sigma}$ with $\sigma \in (0, 0.4]$, thereby motivating the introduction of the SHCG method.

As a result, the proof has been completed.

3 | Global Convergence

This section examines the global convergence of the suggested method under the following assumption.

Assumption 1. The set of levels $\Lambda = \{\{x \in \mathbb{R}^n\} | f(x) \leq f(x_0)\}$ is bounded.

Assumption 2. There is a constant $L > 0$ in a neighborhood N of Λ such that the gradient of the function f is Lipschitz continuous, and the function f is continuously differentiable, namely,

$$\|g(x) - g(y)\| \leq L\|x - y\|, \text{ for all } x, y \in N. \quad \textbf{(18)}$$

Assumption 3.2 implies that there exists a constant m, such that

$$(x - y)^T(g(x) - g(y)) \geq m\|x - y\|^2, \text{for all } x, y \in N, \quad \textbf{(19)}$$

based on these assumptions on f, we have

$$\|g(x)\| \leq \xi, \quad \text{for all } x \in \Lambda, \quad \textbf{(20)}$$

in which $\xi > 0$ is a constant.

Lemma 2. Suppose that the sequences $\{x_k\}$, $\{g_k\}$, $\{d_k\}$ are determined by the SHCG Algorithm and the step size α_k is calculated using the Wolfe line search technique *Eqs. (4)*, *(5)*. If Assumption 1 is correct, then

$$\sum_{k=1}^{+\infty} \frac{\left(g_k^T d_k\right)^2}{\|d_k\|^2} < +\infty,$$

which is known as the Zoutendijk condition.

Using *Eq. (14)*, we can also express it as

$$\sum_{k=1}^{+\infty} \frac{\|g_k\|^4}{\|d_k\|^2} < +\infty.$$

Lemma 3. Assume that *Assumption 2* is true. If the step size α_k satisfies the Wolf condition and d_{k+1} is a descent direction, then

$$\alpha_k \geq \frac{(\sigma - 1)d_k^T g_k}{L\|d_k\|^2}.$$

Proof:

$$(\sigma-1)d_k^T g_k \le d_k^T(g_{k+1}-g_k) \le \|g_{k+1}-g_k\|\|d_k\| = L\alpha_k\|d_k\|^2,$$

Thus,

$$\alpha_k \ge \frac{(\sigma-1)d_k^T g_k}{L\|d_k\|^2}.$$

According to *Lemma 3*, inequalities *Eqs. (6)* and *(10)*, we get that α_k which is calculated in the SHCG method, is not equal to zero, namely, there exists $\varphi > 0$ such that

$$\alpha_k \ge \varphi, \quad \text{for all } k \ge 0 \tag{21}$$

Theorem 1. Suppose *Assumptions 1* and *2* hold. Let the sequences $\{g_k\}$ and $\{d_k\}$ be generated by the SHCG method. Then

$$\liminf_{k\to\infty} \|g_k\| = 0. \tag{22}$$

Proof: Assume that *Eq. (22)* does not hold. Then there exists a constant $\mu > 0$ such that

$$\|g_k\| > \mu,$$

for al k l. By Lipschitz continuity, we have

$$\|y_k\| = \|g_{k+1}-g_k\| \le L\|x_{k+1}-x_k\| \le LD,$$

where $D = \max\{\|x-y\| : x, y \in N\}$ is the diameter of the level set N. From *Eq. (7)* we get

$$\begin{aligned}
\left|\beta_k^{hybrid}\right| &= \left|\frac{g_{k+1}^T(y_k - ts_k)}{\max\{y_k^T d_k, \|g_k\|^2\}}\right| \le \left|\frac{g_{k+1}^T(y_k - ts_k)}{y_k^T d_k}\right|, \\
&\le \left|\frac{\|g_{k+1}\|\left(\|y_k + t\|s_k\|\|\right)}{\|d_k\|\|y_k\|}\right|, \\
&\le \frac{\|g_{k+1}\|(L+t)\|s_k\|}{m\|s_k\|\|d_k\|}, \\
&\le \frac{\xi(L+t)}{m\|d_k\|}.
\end{aligned} \tag{23}$$

From *Eqs. (8)*, *(19)*, *(23)*, we have

$$\|d_{k+1}\| \le \|\lambda_{k+1}g_{k+1}\| + \left|\beta_k^{hybrid}\right|\|d_k\|,$$

$$\le \|g_{k+1}\| + \frac{\xi(L+t)}{m\|d_k\|}\|d_k\|,$$

$$= \|g_{k+1}\| + \frac{\xi(L+t)}{m},$$

$$\le \xi + \frac{\xi(L+t)}{m} = \xi + M = C,$$

where $\frac{\xi(L+t)}{m} = M$, therefore

$$\|d_{k+1}\| \le C. \tag{24}$$

From *Eq. (24)*, we can get

$$\frac{1}{C^2} \leq \frac{1}{\|d_{k+1}\|^2}. \tag{25}$$

Thus, multiplying both sides of the above inequality by $\|g_{k+1}\|^4$, we have the following result

$$\frac{\|g_{k+1}\|^4}{C^2} \leq \frac{\|g_{k+1}\|^4}{\|d_{k+1}\|^2}.$$

Summing up over k, we right away obtain that

$$\frac{1}{C^2}\sum_{k=1}^{+\infty}\|g_{k+1}\|^4 \leq \sum_{k=1}^{+\infty}\frac{\|g_{k+1}\|^4}{\|d_{k+1}\|^2}.$$

This, together with *Lemma 2*, indicates that

$$\frac{1}{C^2}\sum_{k=1}^{+\infty}\|g_{k+1}\|^4 < +\infty.$$

The above inequality implies that

$$\lim_{k\to\infty}\|g_{k+1}\|^4 = 0.$$

Therefore, we obtain that

$$\lim_{k\to\infty}\|g_{k+1}\| = 0,$$

which completes our proof.

4|Numerical Experiments and Results

4.1|Unconstrained Optimization Problem

This section includes numerical experiments that assess the effectiveness of SHCG, SHS [11], CR, and DL+ [30]. Details about the test problems, which include 70 functions from [31], can be found in *Table 1*. All implementations were carried out using MATLAB 7.14.0.739 (R2012a) on a desktop equipped with an AMD Ryzen 7 PRO 5850U with Radeon Graphics 1.90 GHz, 16 GB of RAM, and the CentOS 6.2 Linux operating system.

In our implementations, we chose $\delta = 0.001$ and $\sigma = 0.25$ for the strong Wolfe line search conditions. Furthermore, the algorithms were terminated when either $k > 10000$ or $||g_k|| < 10^{-6}\,(1 + |f_k|)$ met certain termination criteria. Additionally, the algorithm's efficiency was evaluated using the performance profile introduced by Dolan and Moré [32], focusing on the Total Number of Function and Gradient Evaluations (TNFGE), as defined in [33], along with CPU Time (CPUT), in accordance with the nomenclature from [34]. The findings are presented in *Figs. 1* and *2*, which clearly show that SHCG surpasses the three other methods.

Table 1. Numerical results for all methods.

Function	n	SHCG		SHS		CR		DL+	
		TNF	TIME	TNF	TIME	TNF	TIME	TNF	TIME
ARGLINA	200	12	5.35E-02	12	4.07E-02	12	3.59E-02	12	5.07E-02
BDEXP	5000	16	3.46E-02	16	3.66E-02	16	3.44E-02	16	3.51E-02
BIGGSB1	5000	41129	4.88E+00	72911	6.25E+00	60674	5.56E+0054722		5.16E+00
BQPGABIM	50	284	1.70E-02	384	1.70E-02	294	1.70E-02	291	2.52E-02
BQPGASIM	50	284	1.88E-02	384	1.93E-02	294	1.50E-02	291	1.69E-02

Table 1. Continued.

Function	n	SHCG		SHS		CR		DL+	
		TNF	TIME	TNF	TIME	TNF	TIME	TNF	TIME
BRYBND	5000	228	1.10E-01	383	1.48E-01	224	1.07E-01	233	1.11E-01
CHENHARK	5000	864	1.31E-01	1302	1.46E-01	607	8.40E-02	888	1.19E-01
CHNROSNB	50	2292	6.34E-02	3823	9.03E-02	3697	8.71E-02	3087	7.67E-02
CLPLATEB	5041	43531	1.41E+01	74135	1.90E+01	60139	1.64E+01	52965	1.53E+01
COSINE	10000	49	7.09E-02	44	6.88E-02	50	7.12E-02	49	6.97E-02
CURLY10	10000	1328	4.95E-01	1101	3.63E-01	1113	3.58E-01	1236	4.13E-01
CURLY20	10000	2024	1.12E+00	2550	1.11E+00	2335	1.07E+00	1508	7.69E-01
CURLY30	10000	3001	2.21E+00	2797	1.62E+00	2354	1.47E+00	2269	1.52E+00
DECONVU	63	14876	4.10E-01	11196	2.62E-01	19467	4.79E-01	18419	4.47E-01
DIXMAANA	3000	36	2.68E-02	41	2.81E-02	43	2.78E-02	40	2.70E-02
DIXMAANB	3000	32	2.46E-02	32	2.92E-02	41	2.93E-02	32	2.97E-02
DIXMAANC	3000	37	2.56E-02	32	3.23E-02	62	3.11E-02	32	2.81E-02
DIXMAAND	3000	42	2.81E-02	37	2.65E-02	50	3.14E-02	42	2.71E-02
DIXMAANF	3000	1174	1.68E-01	1846	2.12E-01	1166	1.64E-01	1058	1.50E-01
DIXMAANG	3000	1122	1.66E-01	2071	2.36E-01	1204	1.66E-01	1138	1.57E-01
DIXMAANH	3000	1162	1.69E-01	1602	1.87E-01	1270	1.73E-01	1050	1.46E-01
DIXMAANI	3000	12946	1.82E+00	17551	1.84E+00	19067	2.27E+00	18901	2.25E+00
DIXMAANJ	3000	1442	2.08E-01	2005	2.31E-01	1352	1.91E-01	1414	1.97E-01
DIXMAANK	3000	1266	1.85E-01	1867	2.19E-01	1230	1.75E-01	1218	1.70E-01
DIXMAANL	3000	1022	1.52E-01	1722	1.99E-01	1072	1.49E-01	1098	1.57E-01
DIXON3DQ	10000	41045	8.34E+00	73188	1.09E+01	60754	9.57E+00	54480	9.08E+00
DMN15102	66	72935	1.45E+02	84960	1.63E+02	74049	1.46E+02	75357	1.44E+02
DMN15103	99	78348	1.91E+02	117291	2.54E+02	87322	2.07E+02	92030	2.11E+02
DMN37142	66	70349	1.37E+02	77910	1.46E+02	71068	1.38E+02	73592	1.40E+02
DMN37143	99	78556	2.50E+02	166516	3.90E+02	98236	2.63E+02	146633	3.63E+02
DQRTIC	5000	4	2.13E-02	4	2.34E-02	4	2.22E-02	4	1.91E-02
DRCAV2LQ	4489	4	7.63E-02	4	8.02E-02	4	7.21E-02	4	7.88E-02
EDENSCH	2000	143	3.81E-02	154	3.81E-02	179	4.02E-02	151	3.82E-02
EG2	1000	22	1.38E-02	22	1.49E-02	22	1.56E-02	22	1.63E-02
EIGENALS	2550	56779	1.01E+02	72929	1.09E+02	63560	1.03E+02	60057	1.12E+02
EIGENBLS	2550	49263	9.99E+01	71819	1.24E+02	62113	1.22E+02	55531	1.15E+02
EIGENCLS	2652	53645	1.04E+02	71977	1.25E+02	63686	1.15E+02	59174	1.10E+02
ENGVAL1	5000	88	5.66E-02	88	5.41E-02	100	5.69E-02	148	6.33E-02
ERRINROS	50	6325	1.52E-01	89857	1.71E+00	31007	6.24E-01	12209	2.54E-01
FLETCBV2	5000	4	4.48E-02	4	4.41E-02	4	4.41E-02	4	4.38E-02
FLETCBV3	5000	156	1.45E-01	156	1.35E-01	156	1.35E-01	156	1.36E-01
FLETCHBV	5000	148	1.30E-01	148	1.35E-01	148	1.32E-01	148	1.35E-01
FMINSRF2	5625	1730	4.73E-01	2713	6.23E-01	2100	5.34E-01	1863	4.87E-01
FMINSURF	5625	2913	8.16E-01	5092	1.17E+00	4107	1.04E+00	3611	9.35E-01
GENHUMPS	5000	4	3.00E-02	4	3.00E-02	4	2.94E-02	4	3.03E-02
GENHUMPS	5000	4	4.89E-02	4	5.13E-02	4	4.55E-02	4	4.94E-02
LIARWHD	5000	365	8.84E-02	569	1.07E-01	162	5.92E-02	514	1.00E-01
MANCINO	100	89	1.55E-01	84	1.45E-01	89	1.54E-01	89	1.57E-01
MSQRTALS	1024	33374	1.44E+01	40941	1.48E+01	62080	2.33E+01	55728	2.19E+01
MSQRTBLS	1024	18078	7.84E+00	27406	9.90E+00	46743	1.77E+01	34270	1.36E+01

Table 1. Continued.

Function	n	SHCG		SHS		CR		DL+	
		TNF	TIME	TNF	TIME	TNF	TIME	TNF	TIME
NONCVXU2	5000	4	3.36E-02	4	2.74E-02	4	2.94E-02	4	3.14E-02
NONDQUAR	5000	22096	2.48E+00	24406	2.49E+00	27236	3.01E+00	28916	2.95E+00
PENALTY1	1000	356	2.56E-02	401	2.39E-02	310	2.18E-02	269	2.21E-02
PENALTY2	200	4	1.05E-02	4	7.64E-03	4	1.83E-03	4	9.08E-03
POWELLSG	5000	439	8.16E-02	1332	1.56E-01	4870	5.12E-01	947	1.25E-01
POWER	10000	2151	4.62E-01	6092	9.38E-01	3581	6.81E-01	2771	5.03E-01
QUARTC	5000	4	2.05E-02	4	1.88E-02	4	1.91E-02	4	2.10E-02
SCHMVETT	5000	105	9.27E-02	110	9.46E-02	114	9.69E-02	124	9.63E-02
SENSORS	100	134	2.11E-01	143	2.19E-01	137	2.15E-01	139	2.22E-01
SPARSINE	5000	55969	2.86E+01	73537	3.39E+01	64884	3.11E+01	62698	3.02E+01
SPARSQUR	10000	372	2.06E-01	508	2.31E-01	340	1.80E-01	332	1.80E-01
SPMSRTLS	4999	1238	3.72E-01	1797	4.64E-01	1477	4.04E-01	1258	3.63E-01
SROSENBR	5000	128	3.67E-02	316	5.09E-02	112	3.63E-02	215	4.26E-02
TESTQUAD	5000	49539	4.82E+00	73195	5.65E+00	62095	5.02E+00	55795	4.72E+00
TOINTGSS	5000	20	3.42E-02	20	3.51E-02	20	3.33E-02	20	3.35E-02
TOINTPSP	50	710	2.68E-02	832	2.72E-02	735	2.42E-02	725	2.28E-02
TOINTQOR	50	168	1.24E-02	206	1.38E-02	178	1.19E-02	163	1.39E-02
TQUARTIC	5000	190	8.48E-02	409	9.97E-02	178	8.08E-02	254	8.60E-02
TRIDIA	5000	12014	1.35E+00	26172	2.20E+00	31649	2.75E+00	21102	1.95E+00
VARDIM	200	8	1.18E-03	8	7.66E-03	8	4.15E-03	8	1.34E-03

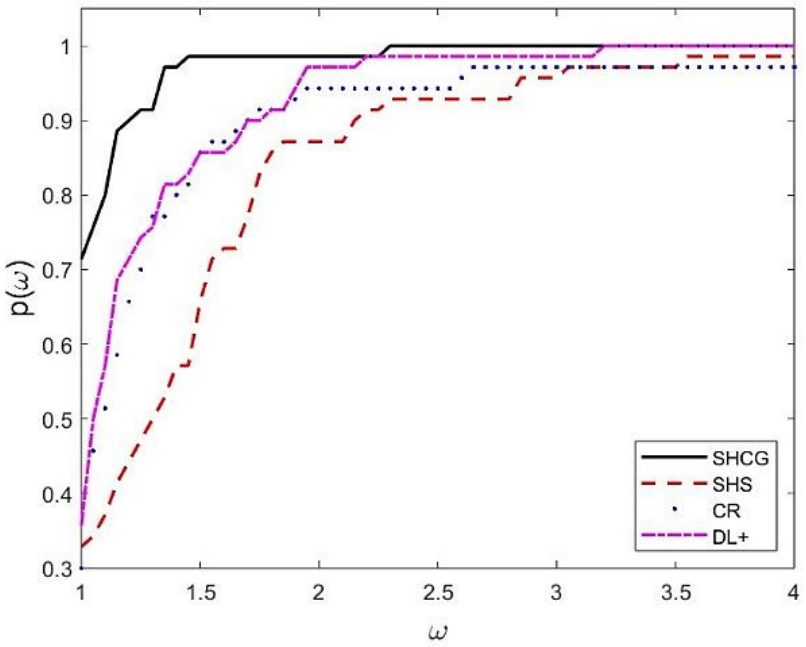


Fig. 1. Comparison based on TNFGE with the strong wolfe line search.

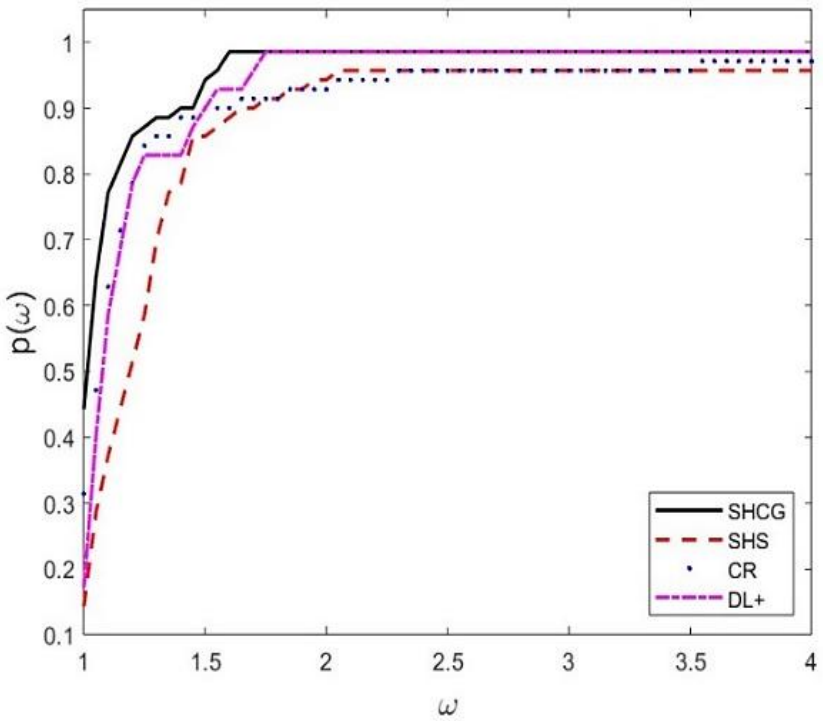


Fig. 2. Comparison based on CPUT with the strong wolfe line search.

4.2 | Application to Image Restoration Problems

In this subsection, we demonstrate applying the suggested techniques to an image reconstruction problem in the sections that follow, offering a practical case study. Images frequently suffer from impulse noise caused by faulty sensors or transmission channels. This type of noise is one of the most prevalent models, where only a subset of the pixels is affected, resulting in a complete loss of information for those specific pixels. To reliably recover the original image, various image-related applications typically require effective noise suppression techniques. These issues are often viewed as complex optimization challenges due to their nonsmooth structures. Recently, researchers have focused on developing nonlinear CG algorithms to tackle such nonsmooth optimization problems; see, for instance [22], [35–38]. In this section, we address the problem of smooth image reconstruction presented in [38]:

$$\min \psi(u),$$

where

$$\psi(u) = \sum_{(i,j)\in\mathcal{N}} \left\{ \sum_{(m,n)\in V_{ij}\setminus\mathcal{N}} \varphi_\alpha(u_{ij} - \zeta_{mn}) + \frac{1}{2} \sum_{(m,n)\in V_{ij}\cap\mathcal{N}} \varphi_\alpha(u_{ij} - u_{mn}) \right\}.$$

In which

$$\mathcal{N} = \{(i,j) \in A \mid \bar{\zeta}_{ij} \neq \zeta_{ij}, \zeta_{ij} = s_{min} \text{ or } s_{max}\}.$$

The index set for the noise candidate is denoted as follows: let X represent the true image composed of $M \times N$ pixels, where X_{ij} indicates the gray level at the pixel location (i,j) for all $(i,j) \in A = \{1, 2, \dots, M\} \times \{1, 2, \dots, N\}$. The neighborhood around the pixel (i,j) is given by $V_{ij} = \{(i, j-1), (i, j+1), (i-1, j), (i+1, j)\}$. The observed image, ζ, reflects the true image X but is affected by salt-and-pepper noise. The image $\bar{\zeta}$ is derived by applying an adaptive median filter to the noisy image ζ. Additionally, s_{min} and s_{max} represent the minimum and maximum values of a noisy pixel, respectively. We utilize the edge-preserving functional φ_α, defined as

$\varphi_\alpha(t) = \sqrt{t^2 + \alpha}$, based on the recommendation in [38]. In our experiments, we set $\alpha = 1$ due to the satisfactory numerical results observed across various choices of α in the set $\{0.5k\}_{k=10}^{10}$.

In this analysis, the images of Lena, Cameraman, and Goldhill, each with dimensions of 256×256 pixels and represented in gray scale, have been used as test cases. Additionally, the Wolfe line search condition and the methods' stopping criteria have been upheld in accordance with those established in the first part of our numerical tests. As illustrated in *Figs. 3-5*, all three methods appear to effectively reconstruct the images. To quantitatively evaluate the results, we analyze the computation time (CPUT in seconds), the Relative Error (RelErr) [39] expressed as a percentage, and the peak signal-to-noise ratio (PSNR) [38] measured in decibels (dB) of the restored image. These metrics are defined as follows.

$$\text{RelErr} = 100\frac{||X^* - X||}{||X||}, \qquad \text{PSNR} = 10\ \log_{10}\frac{255^2}{\frac{1}{M \times N}||X^* - X||_F^2},$$

where X^* represents the recovered image. The results can be found in *Tables 2–4*. As seen in *Tables 3* and *4*, SHCG is better than the other (image reconstruction) methods in the viewpoints of RelErr and PSNR.



Fig. 3. The noisy images corrupted by 25% salt–and–pepper noise; (a)–(d). the restored images via SHCG, (e)–(h). SHS: (i)–(l), CR: (m)–(p), and (q)–(t): DL+.



Fig. 4. The noisy images corrupted by 50% salt–and–pepper noise; (a)–(d). the restored images via SHCG, (e)–(h). SHS, (i)–(l). CR, (m)–(p). (q)–(t). DL+.



Fig. 5. The noisy images corrupted by 75% salt–and–pepper noise: (a)–(d), the restored images via SHCG: (e)–(h), SHS: (i)–(l), CR: (m)–(p), and DL+ : (q)–(t).

Table 2. Image restoration outputs based on CPUT.

Noise	Method	Lena	Cameraman	Goldhill	Bird
25%	SHCG	7.9456	8.1408	6.5357	6.4269
	SHS	6.3199	6.5682	4.9054	4.8875
	CR	6.3494	6.4865	4.8749	4.8641
	DL+	6.3105	6.4849	4.9136	4.8824
50%	SHCG	13.3288	16.1313	13.2541	10.3361
	SHS	128453	15.5334	12.7244	9.8355
	CR	13.1730	15.4926	9.7520	9.8917
	DL+	13.1701	15.5204	9.6922	9.8581

Table 2. Continued.

Noise	Method	Lena	Cameraman	Goldhill	Bird
75%	SHCG	19.2746	23.5000	19.1639	15.1018
	SHS	19.6828	23.4501	19.0108	14.7828
	CR	23.8405	23.5018	18.9533	14.6865
	DL+	23.6793	23.5130	19.0416	14.8335

Table 3. Image restoration outputs based on RelErr.

Noise	Method	Lena	Cameraman	Goldhill	Bird
25%	SHCG	0.7959	1.1330	0.8094	0.2875
	SHS	0.7965	1.1373	0.8097	0.2875
	CR	0.7945	1.1328	0.8090	0.2875
	DL+	0.7945	1.1323	0.8092	0.2875
50%	SHCG	1.3697	1.7262	1.3459	0.6428
	SHS	1.3715	1.7273	1.3478	0.6428
	CR	1.3669	1.7233	1.3187	0.6428
	DL+	1.3667	1.7223	1.3196	0.6428
75%	SHCG	2.1338	2.9029	2.3227	0.8990
	SHS	2.1388	2.9072	2.3297	0.8990
	CR	2.2008	2.8943	2.3067	0.8991
	DL+	2.1996	2.8915	2.3063	0.8991

Table 4. Image restoration outputs based on PSNR.

Noise	Method	Lena	Cameraman	Goldhill	Bird
25%	SHCG	34.4768	31.7427	33.1534	42.6791
	SHS	34.4731	31.7188	33.1545	42.6791
	CR	34.4876	31.7510	33.1613	42.6791
	DL+	34.4878	31.7513	33.1603	42.6787
50%	SHCG	30.4249	27.4956	29.4329	37.1953
	SHS	30.4136	27.4929	29.4216	37.1953
	CR	30.4483	27.5049	29.5888	37.1951
	DL+	30.4479	27.5080	29.5860	37.1941
75%	SHCG	27.0436	24.2538	26.6196	33.8247
	SHS	27.0238	24.2446	26.5953	33.8245
	CR	26.8054	24.2747	26.6792	33.8244
	DL+	26.8093	24.2804	26.6783	33.8242

5 | Conclusion

This work presented a SHCG method for large-scale unconstrained optimization. By integrating a corrected Hestenes–Stiefel scheme with spectral scaling, the method ensures sufficient descent and global convergence, independent of the line search. Extensive numerical experiments, particularly on image restoration, confirmed its superiority in terms of speed, robustness, and solution quality over several state-of-the-art CG variants. We believe that the SHCG framework provides a promising direction for designing next-generation optimization algorithms with broad applicability in science and engineering.

Author Contributions

Conceptualization, Ranjbar M. and Ashrafi A.; Methodology, Ranjbar M.; Software, Ranjbar M.; Validation, Ranjbar M. and Ashrafi A.; Formal analysis, Ranjbar M.; Investigation, Ranjbar M.; Resources, Ashrafi A.; Data maintenance, Ranjbar M.; Writing—creating the initial draft, Ranjbar M.; Writing—reviewing and editing, Ashrafi A.; Visualization, Ranjbar M.; Supervision, Ashrafi A.; Project administration, Ashrafi A. All authors have read and agreed to the published version of the manuscript.

Acknowledgments

The authors would like to thank the anonymous reviewers and the editorial board for their valuable comments, insightful suggestions, and careful evaluation of the manuscript. Their feedback substantially improved the mathematical presentation, technical accuracy, and overall quality of this study.

Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Data Availability

The numerical datasets, computational codes, and image restoration examples used to support the findings of this study are available from the corresponding author upon reasonable request.

References

[1] Wolfe, P. (1969). Convergence conditions for ascent methods. *SIAM review*, *11*(2), 226–235. https://doi.org/10.1137/1011036

[2] Fletcher, R., & Reeves, C. M. (1964). Function minimization by conjugate gradients. *The computer journal*, *7*(2), 149–154. https://doi.org/10.1093/comjnl/7.2.149

[3] Polak, B. T. (1969). The conjugate gradient method in extreme problems. *USSR computational mathematics and mathematical physics*, *9*(4), 94–112. https://doi.org/10.1016/0041-5553(69)90035-4

[4] Hestenes, M. R., Stiefel, E., & others. (1952). Methods of conjugate gradients for solving linear systems. *Journal of research of the national bureau of standards*, *49*(6), 409–436.

[5] Liu, Y., & Storey, C. (1991). Efficient generalized conjugate gradient algorithms, part 1: Theory. *Journal of optimization theory and applications*, *69*(1), 129–137. https://doi.org/10.1007/BF00940464

[6] Dai, Y. H., & Yuan, Y. (1999). A nonlinear conjugate gradient method with a strong global convergence property. *SIAM journal on optimization*, *10*(1), 177–182. https://doi.org/10.1137/S1052623497318992

[7] Chopra, R., & Saxena, R. R. (2019). Linear programming problems with coefficients as trapezoidal intuitionistic Fuzzy numbers. *Advances in fuzzy mathematics. issn 0973-533x*, *14*(1), 41–52. http://www.ripublication.com

[8] Barzilai, J., & Borwein, J. M. (1988). Two-point step size gradient methods. *IMA journal of numerical analysis*, *8*(1), 141–148. https://doi.org/10.1093/imanum/8.1.141

[9] Xu, X., & Kong, F. (2016). New hybrid conjugate gradient methods with the generalized Wolfe line search. *SpringerPlus*, *5*(1), 881. https://doi.org/10.1186/s40064-016-2522-9%0A%0A

[10] Mendoza, H. S. A., Quiroz, E. A. P., & Lengua, M. A. C. (2021). An overview on conjugate gradient methods for optimization, extensions and applications. *2021 IEEE engineering international research conference (EIRCON)* (pp. 1–4). IEEE. https://doi.org/10.1109/EIRCON52903.2021.9613264

[11] Salihu, N., Kumam, P., Sulaiman, I. M., Arzuka, I., & Kumam, W. (2024). An efficient Newton-like conjugate gradient method with restart strategy and its application. *Mathematics and computers in simulation*, *226*, 354–372. https://doi.org/10.1016/j.matcom.2024.07.008

[12] Djordjević, S. S. (2016). New hybrid conjugate gradient method as a convex combination of FR and PRP methods. *Filomat*, *30*(11), 3083–3100. https://www.jstor.org/stable/24899318%0A

[13] Djordjević, S. S. (2017). New hybrid conjugate gradient method as a convex combination of LS and CD methods. *Filomat*, *31*(6), 1813–1825. https://www.jstor.org/stable/24902273%0A

[14] Kaelo, P., Narayanan, S., & Thuto, M. V. (2017). A modified quadratic hybridization of Polak-Ribiere-Polyak and Fletcher-Reeves conjugate gradient method for unconstrained optimization problems. *An international journal of optimization and control: Theories & applications (IJOCTA)*, *7*(2), 177–185. https://pdfs.semanticscholar.org/2883/39f246ac8ca0fc9dd07cbff7a13f6929f313.pdf

[15] Hassan, B. A., Owaid, A. O., & Yasen, Z. T. (2020). A variant of hybrid conjugate gradient methods based on the convex combination for optimization. *Indonesian journal of electrical engineering and computer science*, *20*(2), 1007–1015. https://doi.org/10.11591/ijeecs.v20.i2.pp1007-1015

[16] Alhawarat, A., Salleh, Z., & Masmali, I. A. (2021). A convex combination between two different search directions of conjugate gradient method and application in image restoration. *Mathematical problems in engineering*, *2021*(1), 9941757. https://doi.org/10.1155/2021/9941757

[17] Mohamed, N. S., Mamat, M., Rivaie, M., & Shaharudin, S. M. (2019). A comparison on classical-hybrid conjugate gradient method under exact line search. *International journal of advances in intelligent informatics*, *5*(2), 150–168. http://dx.doi.org/10.26555/ijain.v5i2.356

[18] Zhang, L. (2020). A derivative-free conjugate residual method using secant condition for general large-scale nonlinear equations. *Numerical algorithms*, *83*(4), 1277–1293. https://doi.org/10.1007/s11075-019-00725-7%0A%0A

[19] Cai, J. F., Chan, R., & Morini, B. (2007). Minimization of an edge-preserving regularization functional by conjugate gradient type methods. *Image processing based on partial differential equations: Proceedings of the international conference on PDE-based image processing and related inverse problems, CMA, Oslo, August 8–12, 2005* (pp. 109-122). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-33267-1_7%0A%0A

[20] Chen, X., & Zhou, W. (2010). Smoothing nonlinear conjugate gradient method for image restoration using nonsmooth nonconvex minimization. *SIAM journal on imaging sciences*, *3*(4), 765–790. https://doi.org/10.1137/080740167

[21] Malik, M., Sulaiman, I. M., Abubakar, A. B., Ardaneswari, G., & others. (2023). A new family of hybrid three-term conjugate gradient method for unconstrained optimization with application to image restoration and portfolio selection. *AIMS mathematics*, *8*(1), 1–28. https://doi.org/10.3934/math.2023001

[22] Cao, J., & Wu, J. (2020). A conjugate gradient algorithm and its applications in image restoration. *Applied numerical mathematics*, *152*, 243–252. https://doi.org/10.1016/j.apnum.2019.12.002

[23] Yin, J., Jian, J., & Jiang, X. (2021). A generalized hybrid CGPM-based algorithm for solving large-scale convex constrained equations with applications to image restoration. *Journal of computational and applied mathematics*, *391*, 113423. https://doi.org/10.1016/j.cam.2021.113423

[24] Liu, P., Wu, X., Shao, H., Zhang, Y., & Cao, S. (2023). Three adaptive hybrid derivative-free projection methods for constrained monotone nonlinear equations and their applications. *Numerical linear algebra with applications*, *30*(2), e2471. https://doi.org/10.1002/nla.2471

[25] Yin, J., Jian, J., Jiang, X., Liu, M., & Wang, L. (2021). A hybrid three-term conjugate gradient projection method for constrained nonlinear monotone equations with applications. *Numerical algorithms*, *88*(1), 389–418. https://doi.org/10.1007/s11075-020-01043-z%0A%0A

[26] Liu, Y., Zhu, Z., & Zhang, B. (2022). Two sufficient descent three-term conjugate gradient methods for unconstrained optimization problems with applications in compressive sensing. *Journal of applied mathematics and computing*, *68*(3), 1787–1816. https://doi.org/10.1007/s12190-021-01589-8%0A%0A

[27] Liu, P., Shao, H., Wang, Y., & Wu, X. (2022). A three-term CGPM-based algorithm without Lipschitz continuity for constrained nonlinear monotone equations with applications. *Applied numerical mathematics*, *175*, 98–107. https://doi.org/10.1016/j.apnum.2022.02.001

[28] Aminifard, Z., & Babaie-Kafaki, S. (2022). Dai-Liao extensions of a descent hybrid nonlinear conjugate gradient method with application in signal processing. *Numerical algorithms*, *89*(3), 1369–1387. https://doi.org/10.1007/s11075-021-01157-y%0A%0A

[29] Powell, M. J. (2006). Nonconvex minimization calculations and the conjugate gradient method. *Numerical analysis: Proceedings of the 10th biennial conference held at Dundee, Scotland* (pp. 122-141). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/BFb0099521%0A%0A

[30] Dai, Y. H, & Liao, L. Z. (2001). New conjugacy conditions and related nonlinear conjugate gradient methods. *Applied mathematics & optimization*, *43*(1), 87–101. https://doi.org/10.1007/s002450010019%0A%0A

[31] Gould, N. I. M., Orban, D., & Toint, P. L. (2003). CUTEr and SifDec: A constrained and unconstrained testing environment, revisited. *ACM transactions on mathematical software (toms)*, *29*(4), 373–394. https://doi.org/10.1145/962437.962439

[32] Dolan, E. D., & Moré, J. J. (2002). Benchmarking optimization software with performance profiles. *Mathematical programming*, *91*(2), 201–213. https://doi.org/10.1007/s101070100263%0A%0A

[33] Hager, W. W., & Zhang, H. (2006). Algorithm 851: CG_DESCENT, a conjugate gradient method with guaranteed descent. *ACM transactions on mathematical software (TOMS)*, *32*(1), 113-137. https://doi.org/10.1145/1132973.1132979

[34] Aminifard, Z., & Babaie-Kafaki, S. (2019). An optimal parameter choice for the Dai-Liao family of conjugate gradient methods by avoiding a direction of the maximum magnification by the search direction matrix. *4OR*, *17*(3), 317–330. https://doi.org/10.1007/s10288-018-0387-1%0A%0A

[35] Yuan, G., Li, T., & Hu, W. (2019). A conjugate gradient algorithm and its application in large-scale optimization problems and image restoration. *Journal of inequalities and applications*, *2019*(1), 247. https://doi.org/10.1186/s13660-019-2192-6%0A%0A

[36] Yuan, G., Meng, Z., & Li, Y. (2016). A modified Hestenes and Stiefel conjugate gradient algorithm for large-scale nonsmooth minimizations and nonlinear equations. *Journal of optimization theory and applications*, *168*(1), 129–152. https://doi.org/10.1007/s10957-015-0781-1%0A%0A

[37] Yuan, G., Li, T., & Hu, W. (2020). A conjugate gradient algorithm for large-scale nonlinear equations and image restoration problems. *Applied numerical mathematics*, *147*, 129–141. https://doi.org/10.1016/j.apnum.2019.08.022

[38] Yu, G., Huang, J., & Zhou, Y. (2010). A descent spectral conjugate gradient method for impulse noise removal. *Applied mathematics letters*, *23*(5), 555–560. https://doi.org/10.1016/j.aml.2010.01.010

[39] Esmaeili, H., Shabani, S., & Kimiaei, M. (2019). A new generalized shrinkage conjugate gradient method for sparse recovery. *Calcolo*, *56*(1), 1. https://doi.org/10.1007/s10092-018-0296-x%0A%0A