EUROPEAN JOURNAL OF PURE AND APPLIED MATHEMATICS
2025, Vol. 18, Issue 1, Article Number 5639
ISSN 1307-5543 – ejpam.com
Published by New York Business Global

A Modified Conjugate Gradient Method with Taylor Approximation: Applications in Electric Circuits and Image Restoration

Ahmad Alhawarat[1,*], Sultanah Masmali[2], Ibtisam Masmali[2], Mehiddin Al-Baali[3], Shahrina Ismail[4,5]

[1] *Department of Mathematics, Faculty of Arts and Science, Amman Arab University, Amman 11953, Jordan*
[2] *Department of Mathematics, College of Science, Jazan University, Jazan, Saudi Arabia*
[3] *Department of Mathematics, College of Science, Sultan Qaboos University, Oman*
[4] *Financial Mathematics Program, Faculty of Science and Technology, Universiti Sains Islam Malaysia, Bandar Baru Nilai, 71800, Nilai, Negeri Sembilan, Malaysia*
[5] *Risk and Analytics Research Group, Faculty of Science and Technology, Universiti Sains Islam Malaysia, Bandar Baru Nilai, 71800, Nilai, Negeri Sembilan, Malaysia*

Abstract. In this century, the application of optimization methods is frequently utilized in numerous fields like image restoration, electrical engineering, medical science, machine learning (ML), signal processing as well as many others. In this paper, we choose to improve and maintain one of the most popular, low-memory, and simple algorithms of optimization methods. This method is known as the conjugate gradient method (CGM). Here, we develop a new 3-term CGM with several search directions in the third term suitable for any CGM related to the Fletcher-Reeves method. Apart from that, we propose a new 3-term CGM with mild conditions for any method in relation to the Polak-Ribière-Polyak method. The proposed methods satisfy the descent and convergence properties. Moreover, in the numerical findings section, we perform a comparison of the new method with several renowned methods that have emerged in this century, such as CG-Descent 6.8 and nonnegative Dai-Liao methods utilizing more than 180CUTEst library functions. The numerical findings indicate that the novel approach surpasses recent methodologies. These numerical findings encompass the count of gradient assessments, function assessments, CPU duration as well as iteration count. Additionally, we discussed the implementation of the CG method in image restoration and pi-electric circuits.

2020 Mathematics Subject Classifications: 65K05, 90C30

Key Words and Phrases: Global convergence, Conjugate gradient method, Unconstrained optimization, Pi-electric circuit

---

*Corresponding author.
DOI: https://doi.org/10.29020/nybg.ejpam.v18i1.5639

Email addresses: `a.alhawarat@aau.edu.jo` (A. Alhawarat), `smasmali@jazanu.edu.sa` (S. Masmali), `ibtisam234@hotmail.com` (I. Masmali), `albaali@squ.edu.om` (M. Al-Baali), `shahrinaismail@usim.edu.my` (S. Ismail)

1. Introduction

The conjugate gradient (CG) method is broadly employed to solve unconstrained optimization problems since we can employ the CG method in several fields like image restoration, electrical engineering, machine learning (ML) [24, 25], medical science, signal processing, as well as many others. The general form of optimization problems that we want to solve employing the CG method is expressed as given below:

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

in which the objective function, denoted by$f$, is smooth. The primary iterative formula with regard to the CG method is expressed below:

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

Here, α_k may be gain from inexact or exact line search.

Subsequently, the Strong Wolfe-Powell (SWP) line [22, 23] is an inexact line search commonly used and may be expressed as below:

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \delta \alpha_k \nabla f_k^T d_k, \tag{3}$$

and

$$|\nabla f(x_k + \alpha_k d_k)^T d_k| \leq \sigma |\nabla f_k^T d_k|, \tag{4}$$

On the other hand, the weak Wolfe-Powell (WWP) line search is represented in Eqn. (3) and Eqn. (5) as follows:

$$\nabla f(x_k + \alpha_k d_k)^T d_k \geq \sigma \nabla f_k^T d_k, \tag{5}$$

with $0 < \delta < \frac{1}{2}, \delta < \sigma < 1$.

Here,d_k refers to a search direction often expressed by:

$$d_k = \begin{cases} -\nabla f_k, & \text{if } k = 1, \\ -\nabla f_k + \beta_k d_{k-1}, & \text{if } k \geq 2. \end{cases} \tag{6}$$

in which β_k refers to the CG parameter. Here, the CG parameters are categorized into2 groups. The first group is an effective one. This comprises the methods proposed by Hestenes-Stiefel (HS) [13], Polak-Ribière-Polyak (PRP) [20] as well asLiu and Storey (LS) [17].

$$\beta_k^{HS} = \frac{\nabla f_k^T y_{k-1}}{d_{k-1}^T y_{k-1}}, \quad \beta_k^{PRP} = \frac{\nabla f_k^T y_{k-1}}{\|y_{k-1}\|^2}, \quad \beta_k^{LS} = -\frac{\nabla f_k^T y_{k-1}}{d_{k-1}^T \nabla f_k},$$

in which $y_{k-1} = \nabla f_k - \nabla f_{k-1}$. Nonetheless, this group encounters issues related to convergence, provided that their values become negative [21]. On the other hand, the second group, while less efficient, demonstrates strong global convergence. It is important to note that this category encompasses the Fletcher-Reeves (FR) [10], Fletcher (CD) [9] as well as Dai-Yuan (DY) [7] methods. The equations defining these methods are provided below:

$$\beta_k^{FR} = \frac{\|\nabla f_k\|^2}{\|\nabla f_{k-1}\|^2}, \quad \beta_k^{CD} = -\frac{\|\nabla f_k\|^2}{d_{k-1}^T \nabla f_k}, \quad \beta_k^{DY} = \frac{\|\nabla f_k\|^2}{d_{k-1}^T y_{k-1}}.$$

Dai and Liao [6] recommended the formula of CG and re-expressed it below:

$$\beta_k^{DL} = \frac{\nabla f_k^T y_{k-1}}{d_{k-1}^T y_{k-1}} - \frac{\nabla f_k^T s_{k-1}}{d_{k-1}^T y_{k-1}} = \beta_k^{HS} - \frac{\nabla f_k^T s_{k-1}}{d_{k-1}^T y_{k-1}},$$

where $s_{k-1} = x_k - x_{k-1}$.To avoid the convergence problem mentioned by [21], the authors in [6] use the following restriction:

$$\beta_k^{DL+} = \max\{\beta_k^{HS}, 0\} - t\frac{\nabla f_k^T s_{k-1}}{d_{k-1}^T y_{k-1}},$$

Meanwhile, Li et al. [16] suggested the CG formula as expressed below:

$$\beta_k^{0^*} = \frac{\nabla f_k^T y_{k-1}^*}{d_{k-1}^T y_{k-1}^*} - t\frac{\nabla f_k^T s_{k-1}}{d_{k-1}^T y_{k-1}^*},$$

where

$$y_{k-1}^* = y_{k-1} + \frac{\theta_{k-1}}{\|s_{k-1}\|^2} s_{k-1},$$

$$\theta_{k-1} = 2(f_{k-1} - f_k) + (\nabla f_{k-1} + \nabla f_k)^T s_{k-1}.$$

Hager and Zhang [12] stated the CG formula expressed by:

$$\beta_k^{HZ} = \max\{\beta_k^N, \eta_k\},$$

in which $\beta_k^N = \frac{1}{d_k^T y_k}(y_k - 2d_k \frac{\|y_k\|^2}{d_k^T y_k})^T g_k$, $\eta_k = -\frac{1}{\|d_k\| \min\{\eta, \|\nabla f_k\|\}}$, while $\eta > 0$ denotes a constant.

Alhawarat et al. [3] introduced an effective hybrid CG method that incorporates the SWP line search, as described below:

$$d_k^{PRP^*} = \begin{cases} -\nabla f_k, & k = 1, \\ -\nabla f_k + \left(\frac{\nabla f_k^T \nabla f_k - \nabla f_k^T \nabla f_{k-1}}{\|\nabla f_{k-1}\|^2}\right) d_{k-1}, & \text{if } \|\nabla f_k\|^2 > \nabla f_k^T \nabla f_{k-1},\ k \geq 2, \\ \beta_k^{NPRP}, & \text{elsewhere.} \end{cases}$$

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

To introduce a positive CG method, Alhawarat et al. [4] expressed the positive CG formula having a novel restart property given below:

$$\beta_k^{AZPRP} = \begin{cases} \frac{\|\nabla f_k\|^2 - \mu_k \nabla f_k^T \nabla f_{k-1}}{\|\nabla f_{k-1}\|^2}, & \text{if } \|\nabla f_k\|^2 > \mu_k |\nabla f_k^T \nabla f_{k-1}|, \\ 0, & \text{otherwise.} \end{cases}$$

in which $\|\cdot\|$ resembles the Euclidean norm, while μ_k is expressed as:

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

Jiang et al. [14] recommended the CG method expressed by:

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

To enhance the efficacy of prior methods, they established a restart criterion, which is specified below.

$$d_k = \begin{cases} -g_k, & k = 1, \\ -g_k + \beta_k^{NSL} d_{k-1} + \frac{g_k^T d_{k-1}}{d_{k-1}^T (g_k - g_{k-1})} g_{k-1}, & \text{if } 0 \le g_k^T g_{k-1} \le \|g_k\|^2,\ k \ge 2, \\ -g_k + \frac{g_k^T g_{k-1}}{\|g_{k-1}\|^2} g_{k-1}, & k \ge 2, \text{ otherwise.} \end{cases}$$

Based on Jiang et al. [14] modification, Alhawarat et al. [2] present a 3-term CG method expressed below:

$$d_k = \begin{cases} -g_k, & k = 1, \\ -g_k + \beta_k^{HS} d_{k-1} + \frac{g_k^T d_{k-1}}{d_{k-1}^T (g_k - g_{k-1})} g_{k-1}, & \text{if } \|g_k\|^2 > g_k^T g_{k-1},\ k \ge 2, \\ -\mu_k \frac{g_k^T s_{k-1}}{d_{k-1}^T (g_k - g_{k-1})} d_{k-1}, & \text{otherwise.} \end{cases}$$

Ma et al.[18] presented the 3-term CG method expressed by:

$$d_k = \begin{cases} -\nabla f_k, & k = 1, \\ -\nabla f_k + \beta_k^{LS} d_{k-1} + \xi \frac{\nabla f_k^T q_k}{\|q_{k-1}\|} q_{k-1}, & k \ge 2,\ \beta_k^{LS} \|\nabla f_k^T d_{k-1}\| < \|\nabla f_k\|^2, \\ -\nabla f_k + \xi \frac{\nabla f_k^T q_k}{\|q_{k-1}\|^2} q_{k-1}, & \text{otherwise.} \end{cases}$$

where $0 \le \xi \le 1$, $0 \le \mu \le 1$, while q_{k-1} is any non-zero vector.
Jiang et al. [15] developed the CG method expressed by:

$$d_k = \begin{cases} -\nabla f_k, & k = 1, \\ -\nabla f_k + \beta_k^{TDL} d_{k-1}, & k \ge 2,\ \beta_k^{TDL} \nabla f_k^T d_{k-1} < \xi \|\nabla f_k\|^2, \\ -\nabla f_k + \xi \frac{\nabla f_k^T q_k}{\|q_{k-1}\|^2} q_{k-1}, & \text{otherwise.} \end{cases}$$

in which $\beta_k^{TIDL} = \min\{\beta_k^{IDL}, \kappa \frac{\|\nabla f_k\|}{\|d_{k-1}\|}\}$, $\beta_k^{IDL} = \frac{\nabla f_k^T y_{k-1} - t_k \nabla f_k^T d_{k-1}}{d_{k-1}(\nabla f_k - \nabla f_{k-1})}$, κ is a positive constant, while $0 < t_k < \rho$, where ρ is a positive constant.
Moreover, Alhawarat et al. [2] established the 4-term CGM by employing the directions expressed below:

$$-\nabla f_k,\ d_{k-1},\ y_{k-1},\ \text{and } s_{k-1}$$

$$d_k^{FTCGHS} = -\nabla f_k - t_k \frac{\nabla f_k^T s_{k-1}}{y_{k-1}^T s_{k-1}} d_{k-1} - \left(\frac{\nabla f_k^T d_{k-1}}{y_{k-1}^T d_{k-1}} \right) (y_{k-1} + s_{k-1}).$$

The descent condition (DC), called the downhill condition, is written as

$$\nabla f_k^T d_k < 0, \quad \forall k \geq 1. \tag{7}$$

is crucial in examining CG methods, which is also a crucial part in proving the global convergence analysis. For example, Al-Baali [1] altered Eqn.(7) as follows and employed it in proving the FR method:

$$\nabla f_k^T d_k \leq -c\|\nabla f_k\|^2, \quad \forall k \geq 1, \quad c \in (0, 1). \tag{8}$$

in which $c \in (0,\ 1)$. Here, Eqn. (10) given below resembles the sufficient DC. Furthermore, it has better performance than Eqn.(9) because the quantity of may be regulated by utilising $\|\nabla f_k\|^2$:

2. Proposed CG method related to β_k^{FR}

To maintain or improve the former useful behavior and rectify any formula related to β_k^{FR}, we construct a new 3-term CG method relying on [18] and based on Taylor's expansion as given below:
Let $f(x_{k-1}) \approx f(x_k) - \nabla f_k^T s_{k-1} + \frac{1}{2} s_{k-1}^T G(x_k) s_{k-1}$,
where $G(x_k)$ is a Hessian matrix.

$$\begin{aligned} -(-f(x_{k-1}) + f(x_k)) &= -\nabla f_k^T s_{k-1} + \frac{1}{2} s_{k-1}^T G(x_k) s_{k-1} \\ \frac{2(f(x_k) - f(x_{k-1}))}{\|s_{k-1}\|^2} s_{k-1} &= 2g_k - G(x_k) s_{k-1} \\ \frac{2(f(x_k) - f(x_{k-1}))}{\|s_{k-1}\|^2} s_{k-1} - 2g_k &= -G(x_k) s_{k-1}. \end{aligned}$$

Let

$$\frac{(f(x_k) - f(x_{k-1}))}{\|s_{k-1}\|^2} s_{k-1} = \varepsilon_k.$$

Thus, it is acceptable to add ε_k to the search direction with regards to the CG method given below:

$$d_k = -\nabla f_k + \beta_k d_{k-1} + \varepsilon_k$$

Utilizing Eqn. (3), β_k^{FR}, and Eqn. (6), the new search direction become given by:

$$d_k^{PPR} = \begin{cases} -\nabla f_k, & k = 1, \\ -\nabla f_k + \beta_k \frac{\|\nabla f_k^\top s_{k-1}\|}{\|s_{k-1}\|^2} s_{k-1}, & k \geq 2,\ |\beta_k| \leq \beta_k^{PPR}, \\ -\nabla f_k - \delta \frac{\|\nabla f_k^\top s_{k-1}\|}{\|s_{k-1}\|} s_{k-1}, & \text{otherwise}. \end{cases} \tag{9}$$

In this section, we employ β_k^{FR}as a CG formula. Eqn. (9) can be used for any CG formula related to the FR method, i.e. $|\beta_k| \leq \beta_k^{FR}$. Algorithm 1 illustrates the steps acquired in obtaining the stationary point of the optimization function.

Algorithm 1
Step 1. Establish an initial point x_1, initial direction $d_1 = -g_1$ as well as set$k := 1$
Step 2. Provided that the stopping criterion is met, we may stop.
Step 3. Calculate the search direction d_k according to Eqn. (2) utilizing Eqn. (9).
Step 4. Calculate these step sizes $\alpha_k$utilizing Eqns. (3) and (4).
Step 5. Update x_k relying on Eqn. (2).
Step 6. Set $k := k + 1$ and move forward to Step 2.

3. Global Convergence Properties for Eqn. 11

The objective function is considered to be subject to the following presumption.
Assumption 1
I. The level set $\Psi = \{x \in R^n : f(x) \leq f(x_1)\}$ is bounded, which implies the existence of a nonnegative constant, ρ provided that

$$\|x\| \leq \rho, \forall x \in \Psi.$$

II. In several neighborhoods W of Ψ, f refers to a differentiable as well as a continuous function. Moreover, it has a gradient that is Lipchitz continuous. This implies that, for any $x, y \in W$, a constant $L > 0$ exists given that

$$\|\nabla f(x) - \nabla f(y)\| \leq L \|x - y\|.$$

This assumption suggests the existence of a positive constant η given that

$$\|\nabla f(u)\| \leq \eta, \forall \eta \in W.$$

To establish the CG method's convergence characteristics with multiple line searches, which include SWP as well as WWP line searches, the following Lemma, introduced by Zoutendijk [26], is typically utilized.
Lemma 3.1 Let Assumption 1 to be true. We now take into consideration any form with regard to Eqn. (2),in which α_k fulfills the WWP line search with the DC expressed in Eqn. (8).Therefore, the inequality given below holds.

$$\sum_{k=1}^{\infty} \frac{\left(\nabla f_k^T d_k\right)^2}{\|d_k\|^2} < \infty. \tag{10}$$

The theorem given below demonstrates that the new formula meets the DC (8).Note that the proof is analogous to the one provided in [2].
Theorem 3.1 Assume the sequences $\{x_k\}$ as well as $\{d_k\}$ be generated by the methods (2) as well as (9). We take into consideration the line search method derived from Eqns. (3) as well as (4). Consequently, the sufficient DC specified in Eqn. (8) is met.
Proof.

Case 1 $|\beta_k| \leq \beta_k^{FR}$
Upon multiplying Eqn. (9) with g_k^T gives

$$\nabla f_k^T d_k = -\left\|\nabla f_k\right\|^2 + \frac{\left\|\nabla f_k\right\|^2}{\left\|\nabla f_{k-1}\right\|^2} \nabla f_k^T d_{k-1} + \delta \frac{\nabla f_k^T s_{k-1}}{\left\|s_{k-1}\right\|^2} \nabla f_k^T s_{k-1}.$$

Utilizing a SWP line search, we now have

$$(\delta - 1) + \sigma \frac{\nabla f_{k-1}^T d_{k-1}}{||\nabla f_{k-1}||^2} \leq \frac{\nabla f_k^T d_k}{||\nabla f_k||^2} \leq (\delta - 1) - \sigma \frac{\nabla f_{k-1}^T d_{k-1}}{||\nabla f_{k-1}||^2}$$

$$\delta - \sum_{j=0}^{k-1} (\sigma)^j \leq \frac{\nabla f_k^T d_k}{||\nabla f_k||^2} \leq \delta - 2 + \sum_{j=0}^{k-1} (\sigma)^j$$

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

We can now write

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

When $\sigma \leq \frac{1}{2} - \delta$, we have $\frac{1-(\sigma)^k}{1-\sigma} < 2 - \delta$. Suppose $c = (2 - \delta) - \frac{1-(\sigma)^k}{1-\sigma}$.
Then,

$$c - (2 - \delta) \leq \frac{\nabla f_k^T d_k}{||\nabla f_k||^2} \leq -c$$

$$\nabla f_k^T d_k \leq -c||\nabla f_k||^2.$$

Case 2 $\beta_k > \beta_k^{FR}$

$$\nabla f_k^T d_k = -||\nabla f_k||^2 - \delta \frac{\nabla f_k^T p_{k-1}}{\left\|p_{k-1}\right\|^2} \nabla f_k^T p_{k-1} = -||\nabla f_k||^2 - \delta \frac{\left\|\nabla f_k p_{k-1}\right\|^2}{\left\|p_{k-1}\right\|^2} \leq -c||\nabla f_k||^2.$$

This completes the proof.

Theorem 3.2 Suppose Assumption 1 is met. Additionally, presume that sequences $\{g_k\}$ as well as $\{d_k\}$ are produced through Algorithm 1, in which α_k is gained from Eqns. (3) and (4) with $\sigma \leq \frac{1}{2} - \delta$. We now have that $\lim_{k \to \infty} \inf \|\nabla f_k\| = 0$.

Proof.
Case 1: $|\beta_k| \leq \beta_k^{FR}$.
The theorem is proved via contradiction. Let

$$\|\nabla f_k\| \geq \varepsilon \quad \text{for all } k \geq 0. \tag{11}$$

By taking square with respect to both sides of Eqn. (6), we obtain

$$\|d_k\|^2 = \|\nabla f_k\|^2 - 2\beta_k \nabla f_k^T d_{k-1} + \beta_k^2 \|d_{k-1}\|^2 .$$

Applying the formula in Eqn. (9) yields

$$\|d_k\|^2 \leq \|\nabla f_k\|^2 - 2\frac{\|\nabla f_k\|^2}{\|\nabla f_{k-1}\|^2}\nabla f_k^T d_{k-1} - 2\delta \frac{\nabla f_k^T p_{k-1}}{\|p_{k-1}\|^2}\nabla f_k^T d_{k-1} + \frac{\|\nabla f_k\|^4}{\|\nabla f_{k-1}\|^4}\|d_{k-1}\|^2 + \delta^2 \frac{\left(\nabla f_k^T s_{k-1}\right)^2}{\|s_{k-1}\|^4}\|d_{k-1}\|^2$$
$$+ 2\delta \frac{\|\nabla f_k\|^2}{\|\nabla f_{k-1}\|^2}\frac{\left|\nabla f_k^T s_{k-1}\right|}{\|s_{k-1}\|^2}\|d_{k-1}\|^2 ,$$

We then have the equation given below.

$$\|d_k\|^2 \leq (1+\delta)^2 \|\nabla f_k\|^2 + \frac{\|\nabla f_k\|^4}{\|\nabla f_{k-1}\|^4}\|d_{k-1}\|^2 + (2+2\delta)\frac{\|\nabla f_k\|^2}{\|\nabla f_{k-1}\|^2}\left|\nabla f_k^T d_{k-1}\right| .$$

Dividing both sides of the equation by $\|\nabla f_k\|^4$ and employing Theorem 3.1 yields

$$\frac{\|d_k\|^2}{\|\nabla f_k\|^4} \leq \frac{\|d_{k-1}\|^2}{\|\nabla f_{k-1}\|^4} + \frac{\sigma\ c(2+2\delta)+(1+\delta)^2}{\|\nabla f_k\|^2}$$

Repeating similar process and utilising the relation $\frac{1}{\|\nabla f_0\|} = \frac{1}{\|d_0\|}$ gives

$$\frac{\|d_k\|^2}{\|\nabla f_k\|^4} \leq (\sigma\ \mathrm{c}(2+2\delta)+(1+\delta)^2)\sum_{i=0}^{k}\frac{1}{\|\nabla f_i\|^2}.$$

Now, by using Eqn. (3), we obtain

$$\frac{\|\nabla f_k\|^4}{\|d_k\|^2} \geq \frac{\varepsilon^2}{(\sigma\ c(2+2t)+(1+t)^2)\,k}.$$

Thus,

$$\sum_{k=0}^{\infty}\frac{\|\nabla f_k\|^4}{\|d_k\|^2} = \infty,$$

contradicting Eqn. (10). Therefore, $\lim_{k\to\infty}\ \inf \|\nabla f_k\| = 0$, completing the proof.

Case 2: $\beta_k > \beta_k^{FR}$

Similar to Case 1, let

$$\|\nabla f_k\| \geq \varepsilon \quad \text{for all } k \geq 0.$$

From Eqn. (3), we have

$$\|d_k\| = \left\|-\nabla f_k - \delta \frac{\nabla f_k^T s_{k-1}}{\|s_{k-1}\|^2}s_{k-1}\right\| \leq (1+\delta)\|\nabla f_k\| \leq 1+\eta = \varepsilon_1,$$

where ε_1 is some positive constant, yielding

$$\sum_{k=1}^{\infty} \frac{\|\nabla f_k\|^4}{\|d_k\|^2} \geq \sum_{k=1}^{\infty} \frac{\varepsilon^4}{\varepsilon_1^2} = \infty,$$

which contradicts Eqn. (10). Thus, in all cases, we have

$$\lim_{k \to \infty} \inf \|\nabla f_k\| = 0$$

which completes the proof.

4. Proposed CG method that is related to β_k^{HS} or β_k^{PRP}

Even thoughβ_k^{HS} and $\beta_k^{PRP}$are efficient CG methods while β_k^{HS} inherits the conjugacy condition, both of them can not satisfy the global convergence and the descent properties. Both of these issues are considered open problems. To address these difficulties, we construct the search direction given below relying on the approaches from [18] and [2].

$$d_k^* = \begin{cases} -\nabla f_k, & k = 1, \\ -\nabla f_k + (\beta_k^{HS} or\ \beta_k^{PRP}) d_{k-1} + \delta \frac{\nabla f_k^T s_{k-1}}{\|s_{k-1}\|^2} s_{k-1}, & k \geq 2,\ if\ \|\nabla f_k\|^2 > \left|\nabla f_k^T \nabla f_{k-1}\right| \\ -\nabla f_k - \mu_k \frac{\nabla f_k^T s_{k-1}}{\|s_{k-1}\|^2} s_{k-1}. \text{ elsewhere} \end{cases} \tag{12}$$

As a special case, we employ β_k^{HS} and the condition $\|\nabla f_k\|^2 > \left|\nabla f_k^T \nabla f_{k-1}\right|$ in Eqn. (4) as follows:

$$d_k^{HS+\delta} = \begin{cases} -\nabla f_k, & k = 1, \\ -\nabla f_k + \left(\frac{\nabla f_k^T \nabla f_k - \nabla f_k^T \nabla f_{k-1}}{d_{k-1}^T y_{k-1}}\right) d_{k-1} + \delta \frac{\nabla f_k^T s_{k-1}}{\|s_{k-1}\|^2} s_{k-1}, & k \geq 2, \text{ if } \|\nabla f_k\|^2 > \|\nabla f_k^T \nabla f_k\|, \\ -\nabla f_k - \mu_k \frac{\nabla f_k^T s_{k-1}}{\|s_{k-1}\|^2} s_{k-1}, & \text{elsewhere.} \end{cases} \tag{13}$$

Eqn. (4) possesses the descent property as well as convergence analysis. On the other hand, the numerical results show that Eqn. (4) outperforms DL+ and CG-Descent 6.8.
Theorem 3.3 Let the sequences $\{x_k\}$ as well as $\{d_k\}$ be generated by the Eqns. (2) and (4).We take into consideration the line search method derived utilizing Eqns. (3) as well as (4). Consequently, the sufficient DC (8) is met with$\sigma < \frac{1}{3}$and $\delta \leq \frac{1-3\sigma}{1-\sigma}$.
Proof.
Case 1: $\|\nabla f_k\|^2 > \left|\nabla f_k^T \nabla f_{k-1}\right|$
Upon multiplying Eqn. (4) with ∇f_k^T gives

$$\nabla f_k^T d_k = -\|\nabla f_k\|^2 + \frac{\nabla f_k^T \nabla f_k - \nabla f_k^T \nabla f_{k-1}}{d_{k-1}^T y_{k-1}} \nabla f_k^T d_{k-1} + \delta \frac{\nabla f_k^T s_{k-1}}{\|s_{k-1}\|^2} \nabla f_k^T s_{k-1}.$$

By using the condition in Eqn. (4), we obtain

$$\nabla f_k^T d_k \leq (\delta - 1) \|\nabla f_k\|^2 + \frac{2 \|\nabla f_k\|^2}{d_{k-1}^T y_{k-1}} \nabla f_k^T d_{k-1}.$$

Utilizing the SWP line search, we now have

$$\frac{\nabla f_k^T d_k}{||\nabla f_k||^2} \leq (\delta - 1) - 2\sigma \frac{\nabla f_{k-1}^T d_{k-1}}{(\sigma - 1) \nabla f_{k-1}^T d_{k-1}} = (\delta - 1) + \frac{2\sigma}{(1 - \sigma)}.$$

When $\sigma < \frac{1}{3}$and $\delta \leq \frac{1-3\sigma}{1-\sigma}$, we have

$$\nabla f_k^T d_k < 0.$$

This completes the proof.

Case 2: $\|\nabla f_k\|^2 < \left|\nabla f_k^T \nabla f_{k-1}\right|$

We have

$$d_k = -\nabla f_k - \mu_k \frac{\nabla f_k^T s_{k-1}}{\|s_{k-1}\|^2} s_{k-1}$$

Multiply the equation by

$$\nabla f_k^T,$$

we obtain

$$\nabla f_k^T d_k = - \|\nabla f_k\|^2 - \mu_k \frac{\nabla f_k^T s_{k-1}}{\|s_{k-1}\|^2} \nabla f_k^T s_{k-1} = - \|\nabla f_k\|^2 - \mu_k \frac{- \left\|\nabla f_k^T s_{k-1}\right\|^2}{\|s_{k-1}\|^2} < 0.$$

Thus, in both cases, the DC is satisfied.

Gilbert and Nocedal [11] named a property called Property* to perform a specialized function in research on CG formulas associated with the PRP method given below.

Property*

We take into consideration a method of the form (2) as well as (6) and let

$$0 < \gamma \leq \|g_k\| \leq \bar{\gamma}.$$

The method has Property* provided that ∃ a constant $b > 1$ as well as $\lambda > 0$. Note that ∀ $k \geq 1$, we have $|\beta_k| \leq b$. Moreover, if $\|x_k - x_{k-1}\| \leq \lambda$, this implies that

$$|\beta_k| \leq \frac{1}{2b}.$$

Lemma 3.2 Let Assumption 1 is met. We take into consideration any form of Eqns. (2) as well as (3). Then, β_k^{HS} meets Property* and the proof is similar to the one given in [11].

Proof. Let $b = \frac{2\bar{\gamma}^2}{c((1-\delta)-\sigma)\gamma^2}$ and $\lambda = \frac{c((1-\delta)-\sigma)\gamma^2}{2(L\lambda\bar{\gamma})b}$. Following from here, utilizing β_k^{HS} and SWP line search, we get

$$|\ \beta_k^{HS}| \leq \left| \frac{g_k^T (g_k - g_{k-1})}{d_{k-1}^T y_{k-1}} \right| \leq \frac{\|g_k\|^2 + \left|g_k^T g_{k-1}\right|}{c((1 - \delta) - \sigma) \|g_k\|^2} \leq \frac{2\bar{\gamma}^2}{c((1 - \delta) - \sigma)\gamma^2} = b.$$

Provided that $\|x_{k+1} - x_k\| \leq \lambda$ satisfies Assumption 1, we then have

$$\left|\beta_k^{HS}\right| \leq \left|\frac{\left\|g_k^T\right\|\left(\|g_k - g_{k-1}\|\right)}{d_{k-1}^T y_{k-1}}\right| \leq \frac{(L\lambda\bar{\gamma})}{c((1-\delta)-\sigma)\gamma^2} \leq \frac{1}{2b}.$$

Lemma 3.3 We now have an assumption that Assumption 1 is satisfied, while sequences $\{g_k\}$ as well as $\{d_k\}$ are formed by applying Algorithm 1. Following from here, the step size α_k is created utilizing the SWP line search provided that the sufficient DC is satisfied. Given that $\beta_k \geq 0$, a constant $\gamma > 0$ exists in which $\|g_k\| > \gamma \forall k \geq 1$. Hence, $d_k \neq 0$ and

$$\sum_{k=0}^{\infty} \|u_{k+1} - u_k\|^2 < \infty. \tag{14}$$

in which $u_k = \frac{d_k}{\|d_k\|}$.
Proof. First, provided that $d_k = 0$. Thus, following the sufficient DC, we now obtain $g_k = 0$. Therefore, we assume that $d_k \neq 0$ with

$$\bar{\gamma} \geq \|g_k\| \geq \gamma > 0, \quad \forall k \geq 1. \tag{15}$$

We now define

$$u_k = w_k + \delta_k u_{k-1},$$

in which

$$w_k = \frac{-g_k + \theta_k}{\|d_k\|}, \theta_k = \delta \frac{g_k^T s_{k-1}}{\|s_{k-1}\|^2} s_{k-1}, \delta_k = \beta_k^{HS} \frac{\|d_{k-1}\|}{\|d_k\|}.$$

Provided that u_k expresses a unit vector, we now obtain

$$\|w_k\| = \|u_k - \delta_k u_{k-1}\| = \|\delta_k u_k - u_{k-1}\|.$$

Following the triangular inequality and $\delta_k \geq 0$, we now have

$$\|u_k - u_{k-1}\| \leq (1+\delta_k)\|u_k - u_{k-1}\| = \|u_k - \delta_k u_{k-1} - (u_{k-1} - \delta_k u_k)\|. \tag{16}$$

$$\leq \|u_k - \delta_k u_{k-1}\| + \|u_{k-1} - \delta_k u_k\| = 2\|w_k\|.$$

Next, we express

$$\nu = -g_k + \delta \frac{g_k^T s_{k-1}}{\|s_{k-1}\|^2} s_{k-1}.$$

By utilizing the triangular inequality, we gain:

$$\|\nu\| \leq (1+\delta)\|g_k\| = (1+\delta)\bar{\gamma} = T. \tag{17}$$

Then, $\|\nu\| \leq T$. From Eqn. (16), we have $\|u_k - u_{k-1}\| \leq 2w$.

By Eqns. (16) and (17), we get

$$\sum_{k=0}^{\infty}\|u_{k+1}-u_k\|^2 \leq 4\sum_{k=0}^{\infty}\|w\|^2 \leq 4T^2\sum_{k=0}^{\infty}\frac{1}{\|d_k\|^2} < \infty,$$

completing the proof.
By Lemmas 4.1 and 4.2 in [11], we obtained the results given below.

Theorem 3.4 Suppose sequences $\{x_k\}$ as well as$\{d_k\}$ be generated by Eqns. (2) as well as (3) by utilizing the CG method given in Eqn. (4). Moreover, suppose the step size satisfies (3) as well as (4). By employing Lemmas 3.2, 3.3, as well as Lemmas 4.1 4.2 in [11], we gain the findings such that $\lim_{k\to\infty}\inf\|g_k\| \;= 0$.

5. Numerical Findings and Discussions

To assess the effectiveness of the new search direction as defined in Eqn. (**??**), we selected over 180 test functions from the CUTEr [5] library. We conducted a comprehensive comparison with robust CG coefficients, including the DL+ and FR CG methods. The comparison criteria included CPU time, the number of function evaluations, the number of iterations as well as the number of gradient evaluations. By analyzing these metrics, we aimed to determine how the new search direction performs relative to the benchmark methods. We used the SWP line search setting $\delta = 0.01$ as well as $\sigma = 0.1$ for all methods. The results for the DL+, PFR, and FR methods were obtained by executing a modified version of the CG-Descent code. This modified code is available on the Hager webpage(see here). By adapting the original CG-Descent code, we ensured compatibility with our specific requirements for testing these methods.

The norm of the gradient was used as the stopping criterion for all algorithms, particularly when $\|g_k\| \leq 10^{-6}$. The computations were performed on a host computer running Ubuntu 20.04.2.0 LTS OS, equipped with an 11th Gen Intel(R) Core(TM) i5-1155G7 @ 2.50GHz processor and 8.00 GB RAM. The performance findings are depicted in Figures 1 to 4, utilizing a performance measure established by Dolan and More [8]. Figure 1 illustrates that the PFR method outperforms the DL+ and FR methods in terms of the number of iterations required. As seen in Figure 2, PFR significantly surpasses all other methods in terms of the number of function evaluations. Meanwhile, Figure 3 demonstrates that PFR outperforms DL+ and FR in the context of gradient evaluations. Additionally, Figure 4 indicates that the PFR method not only strongly outperforms the FR CG method with regards to the CPU time but is also competitive with the DL+ CG method.

The numerical findings presented below validate the efficacy of the proposed search direction HS+TA. Detailed results are provided in Table 1. In this study, we utilized over 180 test functions from the CUTEr library to rigorously assess the performance of our

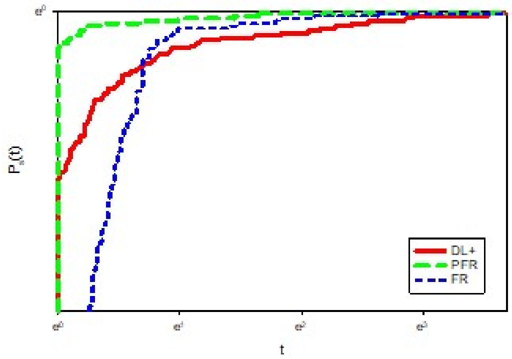


Figure 1: Number of iterations graph

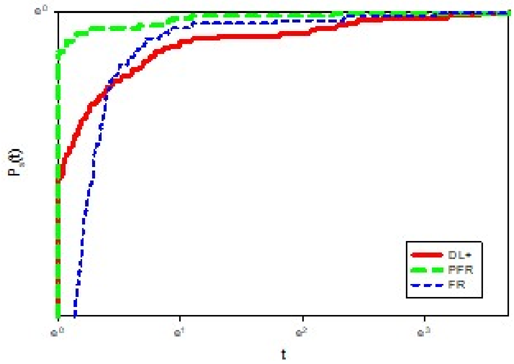


Figure 2: Number of function evaluations graph

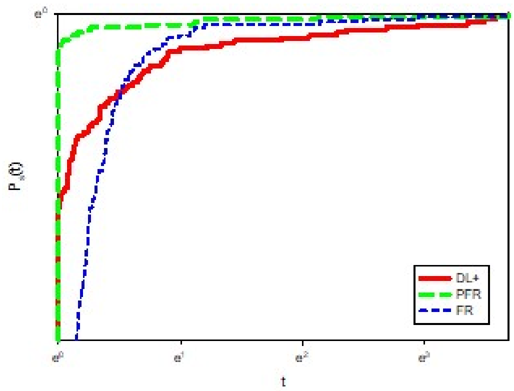


Figure 3: Number of gradient evaluations graph

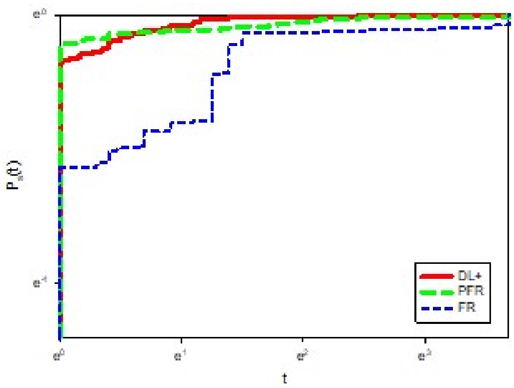


Figure 4: CPU time graph

proposed methods. As an example of using $|\beta_k| \leq \beta_k^{FR}$ in (9), we employ β_k^{AZPRP} as follows:

$$d_k^{TAPRP} = \begin{cases} -\nabla f_k, & k = 1, \\ -\nabla f_k + \beta_k^{AZPRP} d_{k-1} + \delta \dfrac{\nabla f_k^T s_{k-1}}{\|s_{k-1}\|^2} s_{k-1}, & k \geq 2. \end{cases}$$

All computations were performed on a host machine running the Ubuntu 20.04 operating system, equipped with an AMD A4-7210 CPU and 4 GB of RAM. Here, we compared the modified search directions d_k^{HS+TA}, d_k^{TAPRP}, DL+ and CG-Descent 6.8 methods utilizing a SWP line search in obtaining the step length having $\sigma = 0.1$ for d_k^{HS+TA}, d_k^{TAPRP} and DL+ with $\delta = 0.01$. We executed the code using the default parameters specified by the authors for CG-Descent 6.8. The results are illustrated in Figures 5-7, utilizing a performance measure developed by Dolan and More [5]. From Figures 5-7, it can be observed that the new search directions strongly outperformed DL+, CG-Descent 6.8, and d_k^{TAPRP} with regards to the function evaluation, number of iterations as well as CPU time.

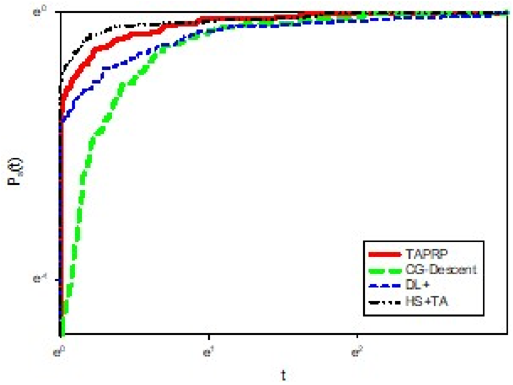


Figure 5: Number of iteration graph

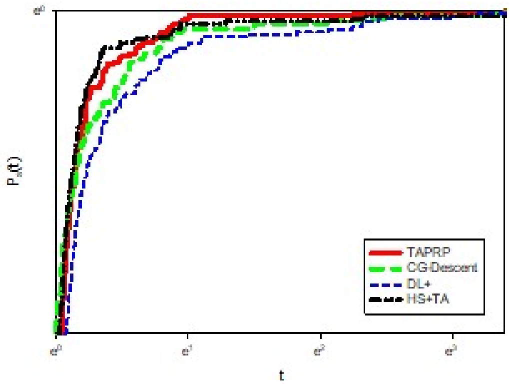


Figure 6: Number of function evaluations graph

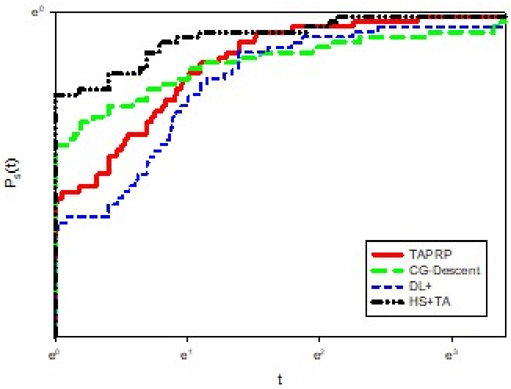


Figure 7: CPU time graph

In Figure 8, we present the function number that is listed in the Table 2 (see in Appendix below) with its gradient for d_k^{HS+TA}. We observe that the number of success functions is more than 99%.

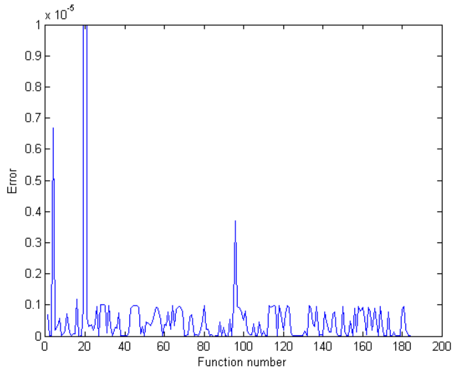


Figure 8: Function number with the gradient for HS+TA method.

6. Application on Pi-Electric circuit

Figure 9 depicts a Pi-Electric circuit, which is utilized as the output network of automatically tuned transmitters [19]. In this circuit, the tuning capacitor C_1 as well as the loading capacitor C_2 are adjusted using semi-independent control loops. Here, the objectivesfor this electric circuit are given below:

1. To achieve a 180° phase shift between $I_1(jw)$ and the voltage across C_1 at a given carrier frequency ω.
2. To attain a specific level of power output to an antenna system.
3. To determine the values of C_1 as well as C_2 that maximize the power delivered to R_2.

Suppose the power output across R_2 be expressed by P_0 given below:

$$P_0 = \frac{|V_0(jw)|^2}{R_2}$$

Let values of $\tau_1 = R_1C_1$ and $\tau_2 = R_2C_2$. Then, from the circuit analysis, we have

$$V_0(s) = \frac{-R_1R_2I_1(s)}{s^3(L\tau_1\tau_1) + s^2L(\tau_1+\tau_2) + s(L + R_1\tau_2 + R_2\tau_1) + R_1 + R_2}$$

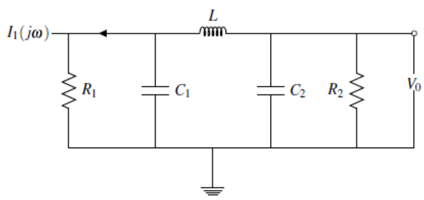


Figure 9: Pi-Electric circuit [8].

$$V_0(jw) = \frac{I_1R_1}{\sqrt{RE^2 + IM^2}}$$

where $I_1 \equiv |I_1(jw)|$

$$RE = 1 + \frac{R_1}{R_2} - \frac{w^2L\tau_1}{R_2} - \frac{w^2L\tau_2}{R_2}$$

$$IM = \frac{wL}{R_2} + \frac{R_1w\tau_2}{R_2} + w\tau_1 - \frac{w^3L\tau_1\tau_2}{R_2}$$

To find P_0, minimize f such that

$$f = [Re(V_0(jw))]^2 + [IM(V_0(jw))]^2$$

having dimensionless parameters stated below:

$$a = \frac{R_1}{R_2}$$

$$b = \frac{wL}{R_2}$$

$$x_1 = w\tau_1$$

$$x_2 = w\tau_2$$

For the case when $a = 10$ and $b = 1$, the objective function that we want to minimize is given by:

$$f(x_1, x_2) = (11 - x_1 - x_2)^2 + (1 + 10x_2 + x_1 - x_1x_2)^2.$$

Its graph is shown in Figure 10.

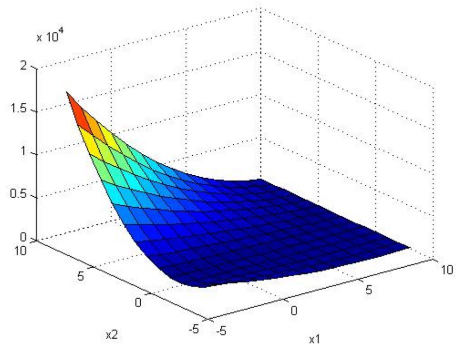


Figure 10: Graph of $f(x_1, x_2)$ by using MATLAB.

By using Algorithm 1, we found that:

$$x_1 = 7,$$
$$x_2 = -2,$$

with function value 40.

7. Application on Image Restoration

Among the important applications of the CG method is restoring damaged images. In this study, we applied Gaussian noise having an SD of 25% to the original images presented in Table 3 and then used Algorithm 1 to restore these images. To illustrate the efficacy of the suggested method (HS+TA), we compared Algorithm 1 with CG-Descent 6.8 and DL+ with regards to the number of iterations, CPU time, as well as RMSE. Moreover, the RMSE between the restored image as well as the original true image was used to evaluate the restoration quality.

$$RMSE = \frac{\|\tau - \tau_k\|_2}{\|\tau\|}.$$

The restored image is expressed by τ_k, while the true image byτ. Moreover, the RMSE is used to assess the restored image's quality, with lower RMSE values indicating higher image quality. The data presented in Table 2 demonstrates that the new search direction surpasses the performance of CG-Descent 6.8 and DL+ in several aspects, including the number of iterations, CPU time, including the RMSE value. The stopping criteria for the process is

$$\frac{\|x_{k+1} - x_k\|_2}{\|x_k\|_2} < \varepsilon.$$

This context yields $\varepsilon = 10^{-3}$. Provided that $\varepsilon = 10^{-4}$ or $\varepsilon = 10^{-6}$, the RMSE remains constant, which indicates that a fixed RMSE value can correspond to a varying number of iterations. This implies that while the RMSE does not change, the number of iterations obligatory to achieve this RMSE may fluctuate.

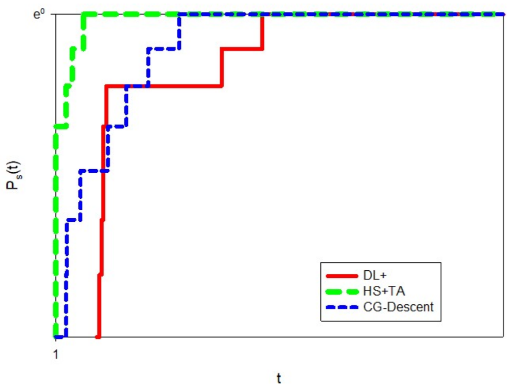


Figure 11: The number of iterations for image restoration represented in Table 2.

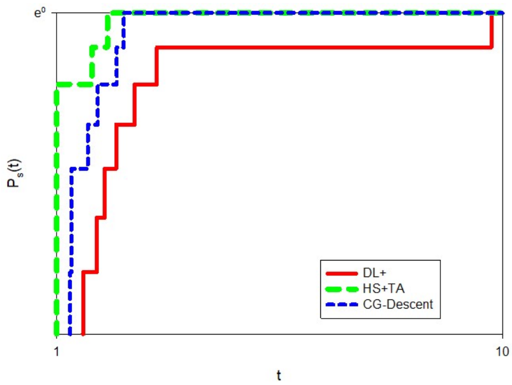


Figure 12: CPU time for image restorationrepresented in Table 2.

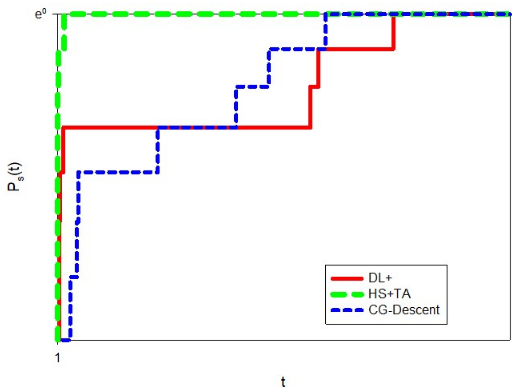
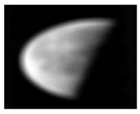
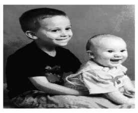
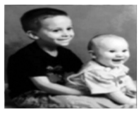
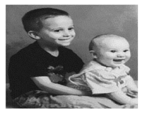
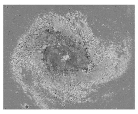
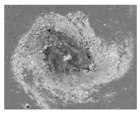
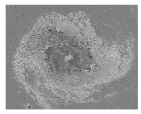


Figure 13: RMSE for image restoration represented in Table 2.

Figures 11, 12, and 13 demonstrate that the latest modification HS+TA outperforms DL+ and CG-Descent in the number of iterations, CPU as well as the RMSE value. On the other hand, we found that CG-Descent outperforms DL+ in all figures. Table 3 below presents the results of restoring corrupted images utilizing HS+TA, highlighting its effectiveness and efficiency. These outcomes suggest that HS+TA is a robust method for image restoration, successfully recovering images to a high standard.

Table 1: Restoration of corrupted images of Moon, Cameraman, Mandi, Coins, Kids, as well as M.83 using HS+TA.

Image	**Original image**	**Image with Gaussian noise**	**Restored image**
Moon (128 pixels)			
Cameraman (128 pixels)			
Mandi (128 pixels)			
Mandi (256 pixels)			
Kids (512 pixels)			
M.83 (1024 pixels)			

8. Conclusion

This research develops a modified CG method based on Taylor expansion. The proposed method satisfies both descent and convergence properties. Moreover, numerical findings demonstrate that the suggested method is robust and effective, outperforming or matching the performance of CG-Descent 6.8 and DL+. In future work, we plan to apply these methods to machine learning applications to explore their potential in this domain.

Acknowledgements

The editors and reviewers are gratefully acknowledged by the authors for any comments and recommendations that might make this work better. We appreciate Dr. William Hager's contribution to the CG method's source code.

Availability of data and material

All data is available in the work itself.

Competing interests

No competing interests are declared by the author.

References

[1] M. Al-Baali. Descent property and global convergence of the fletcher—reeves method with inexact line search. *IMA Journal of Numerical Analysis*, 5(1):121–124, 1985.

[2] A. Alhawarat, G. Alhamzi, I. Masmali, and Z. Salleh. A descent four-term conjugate gradient method with global convergence properties for large-scale unconstrained optimisation problems. *Mathematical Problems in Engineering*, (1):6219062, 2021.

[3] A. Alhawarat, M. Mamat, M. Rivaie, and Z. Salleh. An efficient hybrid conjugate gradient method with the strong wolfe-powell line search. *Mathematical Problems in Engineering*, 2015.

[4] A. Alhawarat, Z. Salleh, M. Mamat, and M. Rivaie. An efficient modified polak–ribière–polyak conjugate gradient method with global convergence properties. *Optimization Methods and Software*, 32(6):1299–1312, 2017.

[5] I. Bongartz, A. R. Conn, N. Gould, and P. L. Toint. Cute: Constrained and unconstrained testing environment. *ACM Transactions on Mathematical Software (TOMS)*, 21(1):123–160, 1995.

[6] Y. H. Dai and L. Z. Liao. New conjugacy conditions and related nonlinear conjugate gradient methods. *Applied Mathematics and Optimization*, 43(1):87–101, 2001.

[7] Y. H. Dai and Y. Yuan. A nonlinear conjugate gradient method with a strong global convergence property. *SIAM Journal on Optimization*, 10(1):177–182, 1999.

[8] E. D. Dolan and J. J. Moré. Benchmarking optimization software with performance profiles. *Mathematical Programming*, 91(2):201–213, 2002.

[9] R. Fletcher. *Practical Methods of Optimization, 2nd Edition.* Chichester, Wiley, 1987.

[10] R. Fletcher and C. M. Reeves. Function minimization by conjugate gradients. *The Computer Journal*, 7(2):149–154, 1964.

[11] J. C. Gilbert and J. Nocedal. Global convergence properties of conjugate gradient methods for optimization. *SIAM Journal on Optimization*, 2(1):21–42, 1992.

[12] W. W. Hager and H. Zhang. The limited memory conjugate gradient method. *SIAM Journal on Optimization*, 23(4):2150–2168, 2013.

[13] M. R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49(6):409–436, 1952.

[14] X. Jiang, J. Jian, D. Song, and P. Liu. An improved polak–ribière–polyak conjugate gradient method with an efficient restart direction. *Computers & Applied Mathematics*, 40:1–24, 2021.

[15] X. Jiang, G. Sun, and J. Jian. A family of dai-liao conjugate gradient methods with strong convergence for image restoration and machine learning. 2024.

[16] G. Li, C. Tang, and Z. Wei. New conjugacy condition and related new conjugate gradient methods for unconstrained optimization. *Journal of Computational and Applied Mathematics*, 202(2):523–539, 2007.

[17] Y. Liu and C. Storey. Efficient generalized conjugate gradient algorithms, part 1: theory. *Journal of Optimization Theory and Applications*, 69(1):129–137, 1991.

[18] G. Ma, L. Liu, J. Jian, and X. Yan. A new hybrid cgpm-based algorithm for constrained nonlinear monotone equations with applications. *Journal of Applied Mathematics and Computing*, 70(1):103–147, 2024.

[19] S. K. Mishra and B. Ram. *Introduction to unconstrained optimization with R*. Springer Nature, 2019.

[20] E. Polak and G. Ribiere. Note sur la convergence de méthodes de directions conjuguées. *ESAIM: Mathematical Modelling and Numerical Analysis-Modélisation Mathématique et Analyse Numérique*, 3(R1):35–43, 1969.

[21] M. J. Powell. Nonconvex minimization calculations and the conjugate gradient method. In *Numerical Analysis*, pages 122–141. Springer, Berlin, Heidelberg, 1984.

[22] P. Wolfe. Convergence conditions for ascent methods. *SIAM Review*, 11(2):226–235, 1969.

[23] P. Wolfe. Convergence conditions for ascent methods. ii: Some corrections. *SIAM Review*, 13(2):185–188, 1971.

[24] K. S. Yow, N. Liao, S. Luo, R. Cheng, C. Ma, and X. Han. A survey on machine learning solutions for graph pattern extraction. *arXiv preprint arXiv:2204.01057*, 2023.

[25] Kai Siong Yow, Ningyi Liao, Siqiang Luo, and Reynold Cheng. Machine learning for subgraph extraction: Methods, applications and challenges. In *Proceedings of the VLDB Endowment*, volume 16, pages 3864–3867, 2023.

[26] G. Zoutendijk. Nonlinear programming, computational methods. In *Integer and Nonlinear Programming*, pages 37–86. 1970.

Table 2: The test function with the number of iterations, function evaluations, gradient evaluations, function value, gradient value, and CPU time for HS+TA.

Function number	Function	DIM	Iteration (HS+TA)	Fun. Evaluations	Grad. Evaluations	Final f	Final g	Time
1	AKIVA	2	8	20	15	6.166042	6.97E-07	0.02
2	ALLINITU	4	9	25	18	5.744385	1.61E-09	0.02
3	ARGLINA	200	1	3	2	200	1.24E-11	0.02
4	ARGLINB	200	1	3	2	99.62547	6.69E-06	0.02
5	ARWHEAD	5000	6	16	12	0	1.84E-07	0.03
6	BARD	3	12	32	22	0.008215	3.28E-07	0.02
7	BDQRTIC	5000	61	162	141	20006.26	5.64E-07	0.02
8	BEALE	2	11	33	26	1.66E-19	2.57E-09	0.02
9	BENNETT5LS	3	14	44	34	0.000539	6.73E-08	0.02
10	BIGGS6	6	24	64	44	0.005656	2.03E-07	0.02
11	BOX	10000	7	25	21	-1864.54	7.17E-07	0.02
12	BOX3	3	10	23	14	6.37E-15	9.22E-08	0.02
13	BOXBODLS	2	12	75	69	1168.009	2.64E-08	0.02
14	BOXPOWER	20000	29	72	47	3.3E-12	7.03E-08	0.03
15	BRKMCC	2	5	11	6	0.169043	6.22E-08	0.02
16	BROWNAL	200	5	64	61	1.47E-09	1.18E-06	0.02
17	BROWNBS	2	10	24	18	0	0	0.02
18	BROWNDEN	4	16	38	31	85822.2	1.53E-10	0.02
19	BROYDN7D	5000	57	104	79	3441.346	9.15E-07	0.02
20	BRYBND	5000	45	105	64	0.538446	0.000967	0.02
21	CHAINWOO	4000	12798	26461	14190	1651.345	5.92E-07	3.06
22	CHWIRUT1LS	3	15	43	34	2384.477	3.18E-07	0.02
23	CHWIRUT2LS	3	15	35	25	513.048	3.58E-07	0.02
24	CLIFF	2	10	46	39	0.199787	1.95E-07	0.02
25	COSINE	10000	10	48	42	-9999	4.37E-07	0.03
26	CRAGGLVY	5000	88	178	149	1688.215	9.4E-07	0.06
27	CUBE	2	17	48	34	1.65E-20	5.82E-09	0.02
28	CURLY10	10000	49278	69394	78474	-1003163	9.98E-07	24.87
29	CURLY20	10000	69850	92869	116727	-1003163	9.99E-07	49.13
30	CURLY30	10000	74607	98792	125185	-1003163	9.87E-07	89.41
31	DANWOODLS	2	8	32	28	0.004317	9.93E-08	0.02
32	DECONVU	63	415	834	420	2.24E-08	9.73E-07	0.02
33	DENSCHNA	2	6	16	12	1.32E-14	3.21E-07	0.02
34	DENSCHNB	2	6	18	15	3.2E-19	1.43E-09	0.02
35	DENSCHNC	2	11	36	31	5.27E-15	2.76E-07	0.02
36	DENSCHND	3	14	46	40	5.24E-12	2.34E-07	0.02
37	DENSCHNE	3	12	43	38	1.46E-13	7.56E-07	0.02
38	DENSCHNF	2	9	31	26	3.18E-22	2.52E-10	0.02
39	DIXMAANA	3000	6	15	11	1	3.94E-13	0.02
40	DIXMAANB	3000	6	16	12	1	2.61E-08	0.02
41	DIXMAANC	3000	6	14	9	1	2.06E-08	0.02
42	DIXMAAND	3000	7	18	13	1	1.12E-07	0.02
43	DIXMAANE	3000	218	245	417	1	9E-07	0.05

Function number	Function	DIM	Iteration (HS+TA)	Fun. Evaluations	Grad. Evaluations	Final f	Final g	Time
44	DIXMAANF	3000	111	227	119	1	9.46E-07	0.02
45	DIXMAANG	3000	171	347	179	1	9.88E-07	0.02
46	DIXMAANH	3000	175	356	185	1	9.89E-07	0.02
47	DIXMAANI	3000	3308	3403	6529	1	9.08E-07	0.56
48	DIXON3DQ	10000	10000	10007	19995	3.16E-13	7.17E-08	2.81
49	DJTL	2	75	1163	1148	-8951.55	3.11E-07	0.02
50	DQDRTIC	5000	5	11	6	3.36E-17	1.14E-08	0.02
51	DQRTIC	5000	16	44	36	1.18E-06	4.47E-07	0.05
52	ECKERLE4LS	3	2	6	4	0.699696	3.81E-07	0.02
53	EDENSCH	2000	24	56	48	12003.28	3.24E-07	0.02
54	EG2	1000	3	8	5	-998.947	4.88E-07	0.02
55	EIGENALS	2550	8775	15326	11017	2.11E-10	7.02E-07	23.2
56	EIGENBLS	2550	14010	28030	14023	1.92E-06	9.28E-07	31.64
57	EIGENCLS	2652	10008	19716	10333	2.18E-11	8.83E-07	21.02
58	ENGVAL1	5000	20	41	35	5548.668	5.18E-07	0.02
59	ENGVAL2	3	26	73	55	3.27E-23	6.59E-10	0.02
60	ENSOLS	9	22	47	27	788.5398	3.55E-07	0.02
61	EXPFIT	2	9	29	22	0.240511	3.34E-07	0.02
62	EXTROSNB	1000	2063	4728	2849	3.81E-07	7.66E-07	0.7
63	FBRAIN2LS	4	79	259	204	0.318972	2.1E-07	0.48
64	FBRAIN3LS	6	1308	3934	3080	0.242722	9.73E-07	1.77
65	FBRAINLS	2	9	27	21	0.416603	3.37E-07	0.03
66	FLETCHCR	1000	209	410	238	7.01E-15	8.69E-07	0.02
67	FMINSRF2	5625	284	578	299	1.000024	9.19E-07	0.13
68	FMINSURF	5625	332	673	345	1	9.45E-07	0.16
69	FREUROTH	5000	29	64	58	608159.2	8.17E-07	0.02
70	GAUSS1LS	8	49	113	74	1315.822	8.44E-09	0.02
71	GAUSS2LS	8	49	124	87	1247.528	4.65E-09	0.02
72	GBRAINLS	2	8	20	13	28.51586	4.33E-08	0.02
73	GENROSE	500	1100	2230	1170	1	6.15E-07	0.03
74	GROWTHLS	3	109	431	369	1.004041	6.87E-07	0.02
75	GULF	3	33	95	72	1.15E-17	1.15E-17	0.02
76	HAHN1LS	7	5	56	53	8522.662	4.03E-08	0.02
77	HAIRY	2	17	82	68	20	7.89E-09	0.02
78	HATFLDD	3	17	49	37	2.55E-07	1.39E-07	0.02
79	HATFLDE	3	13	37	30	2.73E-06	3.38E-07	0.02
80	HATFLDFL	3	21	68	54	6.39E-05	9.84E-07	0.02
81	HEART6LS	6	375	1137	876	8.8E-17	2.23E-07	0.02
82	HEART8LS	8	253	657	440	2.97E-16	2.3E-07	0.02
83	HELIX	3	23	60	42	2.88E-19	1.3E-08	0.02
84	HIELOW	3	13	30	21	874.1654	5.19E-08	0.02
85	HILBERTA	2	2	5	3	5.75E-32	5.75E-32	0.02
86	HILBERTB	10	4	9	5	9.95E-19	2.27E-09	0.02

Function number	Function	DIM	Iteration (HS+TA)	Fun. Evaluations	Grad. Evaluations	Final f	Final g	Time
87	HIMMELBB	2	4	18	18	2.17E-16	4.97E-09	0.02
88	HIMMELBF	4	23	59	46	318.5717	4.39E-07	0.02
89	HIMMELBG	2	7	22	17	3.04E-22	4.92E-11	0.02
90	HIMMELBH	2	5	13	9	-1	2.77E-07	0.02
91	HUMPS	2	45	223	202	9.15E-16	1.35E-08	0.02
92	INDEF	5000	1	46	147	3.184123e-314	3.184008e-314	0.02
93	INTEQNELS	502	6	13	7	1.08E-11	5.53E-07	0.02
94	JENSMP	2	12	47	41	124.3622	1.46E-08	0.02
95	JIMACK	3549	8327	16656	8329	0.866793	9.85E-07	193.23
96	KIRBY2LS	2	54	222	201	3.905074	3.69E-06	0.02
97	KOWOSB	4	16	46	32	0.000308	9.46E-07	0.02
98	LANCZOS1LS	6	61	177	135	2.83E-07	9.05E-07	0.02
99	LANCZOS2LS	6	60	169	125	2.75E-07	7.32E-07	0.02
100	LANCZOS3LS	6	61	164	118	3.41E-07	5.02E-07	0.02
101	LOGHAIRY	2	26	196	179	0.182322	7.9E-07	0.02
102	LSC1LS	3	31	108	89	7.711852	1.73E-07	0.02
103	LSC2LS	3	37	106	86	13.33387	2.98E-08	0.02
104	MANCINO	100	12	30	19	1.95E-21	5.14E-08	0.02
105	MARATOSB	2	589	2885	2585	-1	4.08E-07	0.02
106	MEXHAT	2	14	59	55	-0.04001	1.01E-08	0.02
107	MEYER3	3	19	76	63	9.387247e-323	6.952756e-310	0.02
108	MGH09LS	4	25	82	72	0.001019	4.57E-07	0.02
109	MGH10SLS	19	1082	4052	4968	1.29E+09	6.04E-20	0.02
110	MGH17LS	5	84	323	265	0.024518	1.55E-07	0.02
111	MISRA1ALS	2	33	147	145	0.124551	2.38E-09	0.02
112	MISRA1BLS	2	26	113	101	0.075465	5.97E-09	0.02
113	MOREBV	5000	161	168	317	1.09E-10	9.94E-07	0.3
114	MSQRTALS	1024	2795	5599	2806	6.72E-10	9.2E-07	1.09
115	MSQRTBLS	1024	2235	4476	2247	1.7E-10	9.71E-07	0.95
116	NCB20	5010	906	2105	1348	-82.2049	9.39E-07	0.02
117	NELSONLS	3	1101	5415	7690	25.60412	4.25E-29	0.2
118	NONCVXU2	5000	6476	12572	6855	11584.55	9.83E-07	3.02
119	NONDQUAR	5000	2575	5220	2691	3.09E-06	7.14E-07	0.55
120	OSBORNEA	5	82	230	174	5.46E-05	1.94E-07	0.02
121	OSBORNEB	11	57	134	84	0.040138	7.26E-07	0.02
122	OSCIGRAD	100000	117	191	201	77276.13	9.71E-07	6.48
123	OSCIPATH	500	10	20	14	0.999967	9.3E-07	0.02
124	PALMER1C	8	12	27	28	0.097605	1.28E-07	0.02
125	PALMER1D	7	10	24	23	0.652674	8.27E-11	0.02
126	PALMER2C	8	11	21	22	0.024369	2.2E-09	0.02
127	PALMER3C	8	11	21	21	0.029538	7.92E-10	0.02
128	PALMER4C	8	11	21	21	0.050311	3.55E-09	0.02

Function number	Function	DIM	Iteration (HS+TA)	Fun. Evaluations	Grad. Evaluations	Final f	Final g	Time
129	PALMER5C	6	6	13	7	2.128087	3.75E-12	0.02
130	PALMER6C	8	11	24	24	0.026387	1.88E-08	0.02
131	PALMER7C	8	11	20	20	0.601987	1.52E-07	0.02
132	PALMER8C	8	11	19	19	0.159768	2.75E-09	0.02
133	PARKCH	15	699	1501	1190	1623.743	9.84E-07	1.38
134	PENALTY2	200	191	225	363	4.71E+13	8.84E-07	0.02
135	PENALTY3	200	104	323	256	0.000999	3.82E-07	1.94
136	POWELLBSLS	2	50	211	234	1.14E-07	3.02E-07	0.02
137	POWER	10000	358	730	379	1.74E-09	9.4E-07	0.11
138	POWERSUM	4	4	10	6	37.43145	8.13E-09	0.02
139	PRICE3	2	10	25	17	3.41E-18	3.19E-08	0.02
140	PRICE4	2	9	30	23	3.42E-24	3.13E-10	0.02
141	QING	100	67	134	85	2.62E-14	7.84E-07	0.02
142	QUARTC	5000	16	44	36	1.18E-06	4.47E-07	0.03
143	RAT42LS	3	18	53	44	8.056523	1.36E-07	0.02
144	RAT43LS	4	44	156	122	8786.405	1.17E-07	0.02
145	RECIPELS	3	16	49	38	5.12E-13	7.12E-07	0.02
146	ROSENBR	2	28	84	65	5.87E-16	7.61E-07	0.02
147	ROSENBRTU	2	37	175	153	3.24E-14	1.44E-07	0.02
148	ROSZMAN1LS	4	27	85	66	0.039541	1E-08	0.02
149	S308	2	7	21	17	0.773199	3.3E-09	0.02
150	SCHMVETT	5000	41	71	58	-14994	9.53E-07	0.02
151	SENSORS	100	26	67	47	-2079	1.64E-07	0.06
152	SINEVAL	2	46	181	153	1.49E-23	1.55E-10	0.02
153	SINQUAD	5000	15	44	36	-6757014	1.12E-08	0.02
154	SISSER	2	5	19	19	4.6E-10	4.93E-07	0.02
155	SNAIL	2	61	251	211	7E-17	1.59E-08	0.02
156	SPARSINE	5000	25740	26045	51185	1.15E-10	8.98E-07	14.19
157	SPARSQUR	10000	17	70	70	2.75E-10	1.33E-07	0.02
158	SPMSRTLS	49999	204	413	227	3.49E-11	9.41E-07	0.06
159	SROSENBR	5000	9	23	15	1.3E-12	7.94E-07	0.02
160	SSBRYBND	5000	9465	16566	14479	6.13E-15	9.31E-07	4.56
161	SSI	3	307	1162	990	3.94E-06	5.82E-07	0.02
162	STRATEC	10	170	419	283	2212.262	4.3E-08	1
163	TESTQUAD	5000	1573	1580	3141	1.33E-13	9.19E-07	0.27
164	TOINTGOR	50	122	219	159	1373.905	7.64E-07	0.02
165	TOINTGSS	5000	4	9	5	10.002	1.87E-07	0.02
166	TOINTPSP	50	147	323	241	225.5604	8.83E-07	0.02
167	TOINTQOR	50	29	36	53	1175.472	4.46E-07	0.02
168	TQUARTIC	5000	8	23	18	3.61E-12	2.91E-09	0.02
169	TRIDIA	5000	781	788	1557	4.64E-15	9.37E-07	0.16
170	TRIGON1	10	19	41	22	4.09E-15	3.93E-07	0.02
171	TRIGON2	10	22	57	43	2.966531	2.88E-08	0.02

Function number	Function	DIM	Iteration (HS+TA)	Fun. Evaluations	Grad. Evaluations	Final f	Final g	Time
172	VANDANMSLS	22	2	5	3	9.623043	2.85E-09	0.02
173	VAREIGVL	5000	34	76	45	1.66E-12	7.68E-07	0.16
174	VESUVIALS	8	1262	1954	3155	991.4145	1.6E-18	1.23
175	VESUVIOLS	8	79	198	173	991.41	1.6E-18	0.17
176	VESUVIOULS	8	79	211	173	0.4771138	1.3E-07	0.02
177	VIBRBEAM	8	98	255	174	0.156446	2.83E-08	0.02
178	WAYSEA1	2	11	55	50	6.75E-22	2.99E-09	0.02
179	WAYSEA2	2	9	28	23	1.07E-17	1.37E-09	0.02
180	WATSON	12	60	137	82	1.62E-07	8.55E-07	0.02
181	WOODS	4000	64	184	136	8.34E-15	9.5E-07	0.01
182	YATP1LS	123200	13	34	25	3.88E-16	2.32E-07	0.64
183	YFITU	3	68	208	167	6.67E-13	1.72E-08	0.02
184	ZANGWIL2	2	1	3	2	-18.2	3.79E-15	0.02

Table 3: Numerical results for images with Gaussian noise, incorporating a standard deviation of 25%, via the Dai-Liao CG, AHS, and CG-Descent 6.8 methods.

Image	**Algorithm**	**Number of iterations**	**CPU time (seconds)**	**RMSE**
Mandi 128 pixels	DL+	127	1.724e+000	0.1003
	HS+TA	125	1.563e+000	0.1001
	CG-Descent 6.8	134	1.825e-001	0.1004
Cameraman 128 pixels	DL+	165	1.631e+000	0.1257
	HS+TA	160	1.359e+000	0.1146
	CG-Descent 6.8	163	1.854e+000	0.1148
Coins 128 pixels	DL+	135	1.542e+000	0.0832
	HS+TA	133	1.391e+000	0.0828
	CG-Descent 6.8	133	1.491e+000	0.0831
Mandi 256 pixels	DL+	120	1.856e+001	0.0519
	HS+TA	118	1.745e+001	0.0517
	CG-Descent 6.8	119	1.656e+001	0.0991
Moon 256 pixels	DL+	170	1.678e+001	0.0355
	HS+TA	166	1.234e+001	0.0350
	CG-Descent 6.8	166	1.45e+001	0.0368
Cameraman 256 pixels	DL+	166	1.856e+001	0.0894
	HS+TA	164	1.797e+001	0.0890
	CG-Descent 6.8	162	1.925e+001	0.0892
Coins 256 pixels	DL+	134	1.447e+001	0.0506
	HS+TA	129	1.264e+001	0.0505
	CG-Descent 6.8	130	1.564e+001	0.0508
Mandi 512 pixels	DL+	114	7.981e+001	0.0371
	HS+TA	110	6.955e+001	0.0370
	CG-Descent 6.8	116	7.314e+001	0.0472
kids 512 pixels	DL+	57	6.955e+001	0.0377
	HS+TA	55	5.425e+001	0.0383
	CG-Descent 6.8	55	5.634e+001	0.0395
Coins 512 pixels	DL+	129	7.323e+001	0.0326
	HS+TA	128	5.948e+001	0.0326
	CG-Descent 6.8	127	6.323e+001	0.0503

Cameraman 512 pixels	DL+	148	9.650e+001	0.0533
	HS+TA	146	9.438e+001	0.0533
	CG-Descent 6.8	143	9.727e+001	0.0532
Moon 1024 pixels	DL+	155	1.238e+002	0.0183
	HS+TA	150	1.115e+002	0.0081
	CG-Descent 6.8	150	1.203e+002	0.0082
Cameraman 1024 pixels	DL+	146	3.950e+002	0.0534
	HS+TA	125	2.643e+002	0.0289
	CG-Descent 6.8	130	2.853e+002	0.0298
Coins 1024 pixels	DL+	128	3.441e+002	0.0326
	HS+TA	113	2.049e+002	0.0173
	CG-Descent 6.8	124	2.897e+002	0.0289