

A NEW CONJUGATE GRADIENT METHOD BASED ON A MODIFIED SECANT CONDITION WITH ITS APPLICATIONS IN IMAGE PROCESSING

FAHIMEH ABDOLLAHI AND MASOUD FATEMI*

Abstract. We propose an effective conjugate gradient method belonging to the class of Dai–Liao methods for solving unconstrained optimization problems. We employ a variant of the modified secant condition and introduce a new conjugate gradient parameter by solving an optimization problem. The optimization problem combines the well-known features of the linear conjugate gradient method using some penalty functions. This new parameter takes advantage of function information as well as the gradient information to provide the iterations. Our proposed method is globally convergent under mild assumptions. We examine the ability of the method for solving some real-world problems from image processing field. Numerical results show that the proposed method is efficient in the sense of the PSNR test. We also compare our proposed method with some well-known existing algorithms using a collection of CUTer problems to show its efficiency.

Mathematics Subject Classification. 90C06, 90C26, 65Y20.

Received June 13, 2020. Accepted December 20, 2020.

1. INTRODUCTION

Consider the unconstrained optimization problem

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

where $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is a smooth function. Conjugate gradient algorithm is a powerful iterative method for solving this problem due to its strong local and global convergence properties, and also its low memory requirements. The algorithm generates an iterative sequence

$$x_{k+1} = x_k + \alpha_k d_k, \quad (1.1)$$

where $\alpha_k > 0$ is a step length calculated by some (inexact) line search procedure, like the strong (standard) Wolfe conditions as follows:

$$f(x_k + \alpha_k d_k) \leq f(x_k) + c_1 \alpha_k d_k^T g_k, \text{ and} \quad (1.2)$$

$$|d_k^T g_{k+1}| \leq -c_2 d_k^T g_k \quad (d_k^T g_{k+1} \geq c_2 d_k^T g_k), \quad (1.3)$$

Keywords. Conjugate gradient method, line search, modified secant condition and image processing.

Department of Mathematics, K. N. Toosi University of Technology, Tehran, Iran.

*Corresponding author: smfatemi@kntu.ac.ir

where $0 < c_1 < c_2 < 1$ are arbitrary constants and $g_k := \nabla f(x_k)$. The search direction d_k is computed by

$$d_{k+1} = -g_{k+1} + \beta_k d_k \quad (d_0 = -g_0), \quad (1.4)$$

recursively. The conjugate gradient parameter, denoted by β_k , is one of the most important parameters in this method. This parameter should be defined carefully since different choices of β_k leads to the different conjugate gradient algorithms.

As we know, standard conjugate gradient method for solving strictly convex quadratic problems satisfies both the conjugacy condition

$$d_{k+1}^T y_k = 0, \quad (1.5)$$

where $y_k := g_{k+1} - g_k$, and the orthogonality property

$$d_i^T g_k = 0, \quad i = 0, 1, \dots, k-1. \quad (1.6)$$

Some famous classical choices of β_k can be found in Hestenes and Stiefel (HS) [29], Fletcher and Reeves (FR) [23], Polak and Ribiere and Polyak (PRP) [35], Fletcher (CD) [22], Liu and Storey (LS) [32], and Dai and Yuan (DY) [17]. Some famous of them are

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

where $\|\cdot\|$ denotes the Euclidean norm.

The outcomes of the numerical and theoretical studies, reported by Andrei [2], show that some methods like HS, PRP, and LS (that have the term $g_{k+1}^T y_k$ in the numerator) have a better practical performance than the methods FR, CD, and DY (that have the term $\|g_{k+1}\|^2$ in the numerator). In contrast, the second group has stronger convergence properties. Nowadays, researchers are interested in designing and improving β_k so that the corresponding methods have better numerical performance and good theoretical convergence properties.

Andrei [2] classified conjugate gradient algorithms into 6 groups: classical, hybrid, modified, scaled, parameterized and accelerated; he also introduced 40 different types of these parameters, and thus, 40 different types of conjugate gradient algorithms. See, *e.g.*, [2, 3, 6, 20, 27, 30].

Dai and Liao introduced a class of these parameters in [16] as

$$\beta_k^{\text{DL}} = \frac{g_{k+1}^T y_k - \tau g_{k+1}^T s_k}{y_k^T d_k},$$

where $\tau > 0$ is some constants and $s_k := x_{k+1} - x_k$. They showed that search directions generated by the method satisfy a variant of the conjugacy condition as $d_{k+1}^T y_k = -\tau g_{k+1}^T s_k$, but the sufficient descent condition

$$d_k^T g_k \leq -c \|g_k\|^2, \quad (1.7)$$

where $c > 0$ is a constant, was not necessarily established. It should be noted that an appropriate choices of τ can lead to excellent results. Unfortunately, there are not any theoretically optimal values for this parameter working well for any problem instances. Many efforts have been made to improve this parameter, *e.g.*, Babaie-Kafaki and Ghanbari [9, 10], Andrei [7], Yabe and Takano [39], and Dong *et al.* [20].

One of the most efficient values for τ was presented by Hager and Zhang [25] as $\tau = \frac{2\|y_k\|^2}{s_k^T y_k}$ leading to

$$\beta_k^{\text{HZ}} = \frac{g_{k+1}^T y_k}{d_k^T y_k} - 2 \frac{\|y_k\|^2}{d_k^T y_k} \frac{g_{k+1}^T d_k}{y_k^T d_k}.$$

The search direction generated by β_k^{HZ} satisfies the sufficient descent condition (1.7) with $c = \frac{7}{8}$, independent of the line search procedure. Numerical and theoretical results presented in [2, 6, 25–27] show that it is one of the most efficient methods.

Optimization problems have appeared in many research fields, such as image processing. Image processing is an emerging scientific discipline that has made significant advances in recent years because of its important role in various fields such as photography, medicine, astronomy, industry, and military. Meanwhile, image blur is one of the most common problems in this field. It is caused by different reasons, such as camera and object motion, long exposure times, atmospheric distortions motion, optical aberrations, environmental effects, etc. that, in many cases, is unavoidable [14, 15, 36, 38]. Therefore, the image deblurring problem retrieving an image from a blurred/noisy observation is the intention.

The most commonly used model for the damaged images due to blurring and noise is the following linear model:

$$y = Ax + \eta \quad x \in \mathbb{X}, \quad (1.8)$$

which purpose is to retrieve the original image x from y destroyed by some blurring matrices A and a random noise η , see [14, 15]. The matrix A can be obtained by a certain Point Spread Function (PSF) [14], and there is a PSF corresponding to each kind of blurring [28]. As we know, image deblurring is a large-scale inverse problem that is very ill-condition and hard to solve.

There are various ways to solve this problem [36], for example, SVD approach, Neural network approach, Wiener filtering, Blind deconvolution approach, and iterative algorithms like Richardson–Lucy Algorithm, Van Cittert Algorithm, Landweber Algorithm, and Poisson Map Algorithm.

A well-known approach to model (1.8) is the regularization approach [14]. In this approach, a regularization term is added to the objective function, and the following general model is considered

$$\min_{x \in \mathbb{X}} \frac{1}{2} \|Ax - y\|_2^2 + \lambda \phi(x), \quad (1.9)$$

where ϕ and λ are regularization term and parameter, respectively. The purpose of regularization is to overcome the difficulties that occur in solving (1.8). These difficulties are basically a consequence of the problem's ill-posed nature, and the regularization approach provides a stable solution using some prior knowledge about it. Some examples of regularization terms are

- Tikhonov Regularization: This estimate is defined as the solution of (1.9) with $\phi(x) = \frac{1}{2} \|x\|_2^2$.
- Maximum Entropy Regularization: This estimate is defined as the solution of (1.9) with $\phi(x) = \sum_{i=1}^N x_i \log(x_i)$.
- Total Variation Regularization: This estimate is defined as the solution of (1.9) with $\phi(x) = \|Dx\|_1$, where D is a discrete approximation to the gradient operator.
- Norm p Regularization: This estimate is defined as the solution of (1.9) with $\phi(x) = \|x\|_p$, specially for $0 < p \leq 1$.
- Smooth Regularization: This estimate is defined as the solution of (1.9) with for example $\phi(x) = \sum_{i=1}^N \log(1 + x_i^2)$ or $\phi(x) = \sum_{i=1}^N \frac{x_i^2}{1+x_i^2}$.

Each term has its advantages and disadvantages, and should be used with caution.

In this paper we want to propound a new conjugate gradient method with global convergent and efficient numerical results. The main features of the paper are:

- Our proposed conjugate gradient method, in addition to gradient values, employees the current function information during the iterations. It also takes both advantages of the modified secant condition proposed by Zhang *et al.* [41] and Zhang and Xu [40], and the method presented in [21] to bring out an efficient conjugate gradient parameter. Our numerical results show that the resulting method is more efficient in the sense of Dolan–Moré performance profile. We make extensive numerical comparisons and consider different choices of the main parameters (about 20) and show that our proposed method has a better performance than four recent efficient algorithms in the literature.
- Nowadays, image processing is an active field. Therefore, a slight improvement in the performance of algorithms in this field can have a significant impact on their time efficiency, especially when we are dealing with large dimensions. Thus, to show the efficiency of the method in practice, we examine our algorithm's mathematical performance for solving one of the most appealing image processing problems and report its features and benefits.

The organization of the paper is as follows.

In Section 2, we provide a brief review of the materials required for the other sections. The details of our new method are demonstrated in Section 3. In Sections 4 and 5, the new algorithm is presented, and its global convergence is investigated. In Section 6, we present the numerical behaviour of the method.

2. A BRIEF REVIEW

Nonlinear conjugate gradient methods usually use the secant condition

$$B_{k+1}s_k = y_k, \quad (2.1)$$

where B_{k+1} is an approximation of $G_{k+1} := \nabla^2 f(x_{k+1})$, to benefit from the superlinear convergence properties. Note that, in this condition, the gradient information are only being used. Many efforts have been made by researchers to improve (2.1). Zhang *et al.* [41] and Zhang and Xu [40] introduced the modified secant condition

$$B_{k+1}s_k = \hat{y}_k, \quad \hat{y}_k = y_k + \frac{\theta_k}{s_k^T u_k} u_k, \quad (2.2)$$

where

$$\theta_k = 6(f_k - f_{k+1}) + 3(g_k + g_{k+1})^T s_k, \quad (2.3)$$

and u_k is a suitable vector with $s_k^T u_k \neq 0$. They provided this condition by using the Taylor expansion

$$\begin{aligned} f_k &= f_{k+1} - g_{k+1}^T s_k + \frac{1}{2} s_k^T G_{k+1} s_k - \frac{1}{6} s_k^T (T_{k+1} s_k) s_k + O(\|s_k\|^4), \text{ and} \\ s_k^T g_k &= s_k^T g_{k+1} - s_k^T G_{k+1} s_k + \frac{1}{2} s_k^T (T_{k+1} s_k) s_k + O(\|s_k\|^4), \end{aligned}$$

where $G_k := \nabla^2 f(x_k)$ and $T_{k+1} \in \mathbb{R}^{n \times n \times n}$ is the tensor of f in x_{k+1} . By removing the tensor term in the both expressions, they obtained

$$s_k^T G_{k+1} s_k = (g_{k+1} - g_k)^T s_k + 6(f_k - f_{k+1}) + 3(g_k + g_{k+1})^T s_k + O(\|s_k\|^4).$$

The modified secant condition (2.2) is obtained by considering B_{k+1} as an approximation of G_{k+1} . It is shown in [31] that for a convex quadratic function by using the exact line search, we have $\theta = 0$ in (2.3) and so the modified secant condition is equivalent to the standard secant condition (2.1). Some well-known choices of u_k satisfying $s_k^T u_k \neq 0$ are $u_k = s_k$, $u_k = y_k$ and $u_k = \nabla f(x_k)$.

In the above modified secant condition, in addition to gradient vectors, function values are as well as used. It is proved in [40] that \hat{y}_k is a better approximation of $\nabla^2 f(x_{k+1}) s_k$ than y_k . In fact, if $\|s_k\|$ is sufficiently small,

$$\begin{aligned} s_k^T [G_{k+1} s_k - \hat{y}_k] &= O(\|s_k\|^4), \text{ and} \\ s_k^T [G_{k+1} s_k - y_k] &= O(\|s_k\|^3), \end{aligned}$$

for any u_k with $s_k^T u_k \neq 0$. It is known that using a modified secant condition can lead to designing new methods with better performance in both theoretical and numerical aspects. Many articles, including Andrei [4, 5], Babaie-Kafaki [8], Babaie-Kafaki and Mahdavi-Amiri [11], Babaie-Kafaki *et al.* [12, 13], Yabe and Takano [39], Dong *et al.* [19, 20], Li *et al.* [31], Amini and Gorbani [1], Narushima and Yabe [34], and Wei *et al.* [37], have applied the modified secant equation (2.2) to design and develop new conjugate gradient methods.

Fatemi [21] introduced a new efficient conjugate gradient method which was globally convergent. The method, which was founded on an optimization problem, showed remarkable performance. To provide an efficient conjugate gradient algorithm, the author introduced an optimization problem by combining three conditions (1.5)–(1.7) as follows

$$\min_{\beta_k} \left[g_{k+1}^T d_{k+1} + M \left((g_{k+2}^T s_k)^2 + (d_{k+1}^T y_k)^2 \right) \right], \quad (2.4)$$

where M is a penalty parameter. To solve (2.4), g_{k+2} is replaced by its first order approximation using Taylor series expansion as $g_{k+2} \approx tB_{k+1}d_{k+1} + g_{k+1}$, where $t > 0$ is an estimation of α_{k+1} [21]. By solving this problem and using the secant condition (2.1), the author introduced

$$\beta_k = \frac{-1}{2M(1+t^2)} \frac{g_{k+1}^T d_k}{(y_k^T d_k)^2} + \frac{y_k^T g_{k+1}}{y_k^T d_k} - \frac{t}{(1+t^2)} \frac{s_k^T g_{k+1}}{y_k^T d_k}.$$

This method, with a proper choice of parameters, is globally convergent. Moreover, numerical comparisons reported by the author show that the method is efficient in the sense of Dolan–Moré performance profile [18].

3. A NEW FORMULA FOR β_k

Yabe and Takano in [39] introduced an extension of the modified secant equation (2.2) as

$$z_k = y_k + \rho_k \frac{\theta_k}{s_k^T u_k} u_k, \quad (3.1)$$

where $\rho_k \geq 0$ is an arbitrary scalar. Consider the modified secant condition

$$B_{k+1} s_k = z_k, \quad (3.2)$$

if $\rho_k = 0$ or 1, (3.2) converts to (2.1) and (2.2), respectively.

The following equation was introduced in [21] by solving (2.4)

$$\begin{aligned} \beta_k = & \frac{-g_{k+1}^T d_k + 2Mt^2 (s_k^T B_{k+1} g_{k+1}) (s_k^T B_{k+1} d_k)}{2Mt^2 (s_k^T B_{k+1} d_k)^2 + 2M (y_k^T d_k)^2} \\ & + \frac{-2Mt (s_k^T g_{k+1}) (s_k^T B_{k+1} d_k) + 2M (y_k^T g_{k+1}) (y_k^T d_k)}{2Mt^2 (s_k^T B_{k+1} d_k)^2 + 2M (y_k^T d_k)^2}. \end{aligned} \quad (3.3)$$

It converts to the Dai–Liao parameter, when M goes to infinity.

The good features of the modified secant condition (3.2) motivate us to use it in (3.3) and obtain

$$\beta_k = \frac{-g_{k+1}^T d_k + 2Mt^2 (z_k^T g_{k+1}) (z_k^T d_k)}{2Mt^2 (z_k^T d_k)^2 + 2M (y_k^T d_k)^2} + \frac{-2t (s_k^T g_{k+1}) (z_k^T d_k) + 2 (y_k^T g_{k+1}) (y_k^T d_k)}{2t^2 (z_k^T d_k)^2 + 2 (y_k^T d_k)^2}. \quad (3.4)$$

Now, we face with a big problem which is introducing a suitable penalty parameter M . It should have the following properties.

- The search direction d_{k+1} must satisfy the sufficient descent condition (1.7).
- It should tend to infinity during the iterations.

In the following lemma, we propose such a suitable penalty parameter.

Lemma 3.1. *Consider the conjugate gradient method (1.1) and (1.4) with the strong Wolfe line search (1.2) and (1.3) and β_k in (3.4). Then, for some positive scalars γ_1, γ_2 satisfying $\gamma_1 + \gamma_2 < 1$, and ρ_k selected in such a way that $0 \leq \rho_k < \frac{1-c_2}{3(1+c_2-2c_1)}$, we have*

$$d_{k+1}^T g_{k+1} \leq -(1 - \gamma_1 - \gamma_2) \|g_{k+1}\|^2, \quad (3.5)$$

where

$$(1-t) \leq \frac{\gamma_1 (y_k^T s_k)^2}{(z_k^T s_k) \|s_k\|^2}, \quad (3.6)$$

and

$$M = \frac{2\gamma_2(t^2(z_k^T s_k)^2 + (y_k^T s_k)^2)}{\|t^2(z_k^T s_k)z_k + (y_k^T s_k)y_k - \lambda_k(z_k^T s_k)s_k\|^2}, \quad (3.7)$$

where $\lambda_k \leq 1$ is an arbitrary scalar.

Proof. The proof is similar to Lemma 1 in [21]. We can use the mathematical induction to prove it.

Clearly, we have from (1.4) that

$$d_0^T g_0 = -\|g_0\|^2 \leq -(1 - \gamma_1 - \gamma_2)\|g_0\|^2.$$

Assume

$$d_k^T g_k \leq -(1 - \gamma_1 - \gamma_2)\|g_k\|^2.$$

First, we evaluate the quantity $z_k^T s_k$. We have by (1.2), (1.3), (2.3), and (3.1) that

$$\begin{aligned} z_k^T s_k &= y_k^T s_k + \rho_k \theta_k \\ &= y_k^T s_k + 6\rho_k(f_k - f_{k+1}) + 3\rho_k(g_k + g_{k+1})^T s_k \\ &\geq (1 + 3\rho_k)g_{k+1}^T s_k + (3\rho_k - 1)g_k^T s_k - 6\rho_k c_1 g_k^T s_k \\ &= (1 + 3\rho_k)g_{k+1}^T s_k + (3\rho_k - 6\rho_k c_1 - 1)g_k^T s_k \\ &\geq (1 + 3\rho_k)c_2 g_k^T s_k + (3\rho_k - 6\rho_k c_1 - 1)g_k^T s_k \\ &= (3\rho_k(1 + c_2 - 2c_1) + (c_2 - 1))g_k^T s_k. \end{aligned}$$

Now, using

$$g_k^T s_k = \alpha_k g_k^T d_k < 0,$$

and noting that $0 \leq \rho_k < \frac{1-c_2}{3(1+c_2-2c_1)}$, there exists a constant $m_1 > 0$ such that

$$z_k^T s_k \geq -m_1(g_k^T s_k) > 0, \quad (3.8)$$

and consequently,

$$z_k^T d_k > 0. \quad (3.9)$$

Now, we show that (3.5) holds. Using (1.4) and (3.4), we have

$$\begin{aligned} d_{k+1}^T g_{k+1} &= -\|g_{k+1}\|^2 + \beta_k d_k^T g_{k+1} = -\|g_{k+1}\|^2 - \frac{(s_k^T g_{k+1})^2}{2M \left[t^2(z_k^T s_k)^2 + (y_k^T s_k)^2 \right]} \\ &\quad + \frac{[t^2(z_k^T g_{k+1})(z_k^T s_k) + (y_k^T g_{k+1})(y_k^T s_k)](s_k^T g_{k+1})}{t^2(z_k^T s_k)^2 + (y_k^T s_k)^2} - \frac{t(z_k^T s_k)(s_k^T g_{k+1})^2}{t^2(z_k^T s_k)^2 + (y_k^T s_k)^2}. \end{aligned}$$

Since $\lambda_k \leq 1$, we have that

$$\begin{aligned}
d_{k+1}^T g_{k+1} \pm \frac{(s_k^T g_{k+1})^2 (z_k^T s_k)}{t^2 (z_k^T s_k)^2 + (y_k^T s_k)^2} &\leq -\|g_{k+1}\|^2 - \frac{(s_k^T g_{k+1})^2}{2M [t^2 (z_k^T s_k)^2 + (y_k^T s_k)^2]} \\
&\quad + \frac{[t^2 (z_k^T s_k) (z_k^T g_{k+1}) + (y_k^T s_k) (y_k^T g_{k+1}) - \lambda_k (z_k^T s_k) (s_k^T g_{k+1})] (s_k^T g_{k+1})}{t^2 (z_k^T s_k)^2 + (y_k^T s_k)^2} \\
&\quad + \frac{(1-t) (z_k^T s_k) (s_k^T g_{k+1})^2}{t^2 (z_k^T s_k)^2 + (y_k^T s_k)^2} \\
\Rightarrow d_{k+1}^T g_{k+1} &\leq -\|g_{k+1}\|^2 - \frac{(s_k^T g_{k+1})^2}{2M [t^2 (z_k^T s_k)^2 + (y_k^T s_k)^2]} \\
&\quad + \frac{[t^2 (z_k^T s_k) z_k + (y_k^T s_k) y_k - \lambda_k (z_k^T s_k) s_k]^T g_{k+1} (s_k^T g_{k+1})}{t^2 (z_k^T s_k)^2 + (y_k^T s_k)^2} \\
&\quad + \frac{(1-t) (z_k^T s_k) (s_k^T g_{k+1})^2}{t^2 (z_k^T s_k)^2 + (y_k^T s_k)^2}.
\end{aligned}$$

Now, using the inequality $ab \leq \frac{l}{4}a^2 + \frac{1}{l}b^2$, where a, b and l are positive scalars,

$$\begin{aligned}
d_{k+1}^T g_{k+1} &\leq -\|g_{k+1}\|^2 - \frac{(s_k^T g_{k+1})^2}{2M [t^2 (z_k^T s_k)^2 + (y_k^T s_k)^2]} + \frac{(1-t) (z_k^T s_k) (s_k^T g_{k+1})^2}{t^2 (z_k^T s_k)^2 + (y_k^T s_k)^2} \\
&\quad + \frac{l}{4} \left([t^2 (z_k^T s_k) z_k + (y_k^T s_k) y_k - \lambda_k (z_k^T s_k) s_k]^T g_{k+1} \right)^2 + \frac{1}{l} \left(\frac{(s_k^T g_{k+1})}{t^2 (z_k^T s_k)^2 + (y_k^T s_k)^2} \right)^2,
\end{aligned}$$

where $l = \frac{2M}{t^2 (z_k^T s_k)^2 + (y_k^T s_k)^2}$, and so

$$\begin{aligned}
d_{k+1}^T g_{k+1} &\leq -\|g_{k+1}\|^2 + \frac{(1-t) (z_k^T s_k) (s_k^T g_{k+1})^2}{t^2 (z_k^T s_k)^2 + (y_k^T s_k)^2} \\
&\quad + \frac{M}{2 (t^2 (z_k^T s_k)^2 + (y_k^T s_k)^2)} \left([t^2 (z_k^T s_k) z_k + (y_k^T s_k) y_k - \lambda_k (z_k^T s_k) s_k]^T g_{k+1} \right)^2.
\end{aligned}$$

Finally, using Cauchy–Schwartz inequality implies that

$$d_{k+1}^T g_{k+1} \leq - \left[1 - \frac{(1-t) (z_k^T s_k) \|s_k\|^2}{t^2 (z_k^T s_k)^2 + (y_k^T s_k)^2} - \frac{M}{2 [t^2 (z_k^T s_k)^2 + (y_k^T s_k)^2]} \|t^2 (z_k^T s_k) z_k + (y_k^T s_k) y_k - \lambda_k (z_k^T s_k) s_k\|^2 \right] \|g_{k+1}\|^2.$$

The proof is completed using (3.6), (3.7), and (3.9). \square

In order to accelerate the growth rate of M , that is our second goal, we minimize the denominator of (3.7) and have

$$\lambda_k = \min \left\{ 1, \frac{t^2 (z_k^T s_k)^2 + (y_k^T s_k)^2}{(z_k^T s_k) \|s_k\|^2} \right\}. \quad (3.10)$$

Remark 3.2. Unlike the Newton method which $\alpha_k = 1$ provides a quadratic convergence rate, in conjugate gradient algorithms a suitable step size is not predictable and therefore the following process is suggested.

$$t = \min\{t_1, T\} \quad (3.11)$$

where

$$t_1 = \begin{cases} \alpha_k & \text{if } (1 - \alpha_k) \leq \frac{\gamma_1 (y_k^T s_k)^2}{(z_k^T s_k) \|s_k\|^2} \\ \left| 1 - \frac{\gamma_1 (y_k^T s_k)^2}{(z_k^T s_k) \|s_k\|^2} \right| & \text{o.w.} \end{cases}$$

and $T > 0$ is a large arbitrary constant.

Finally, by replacing (3.7) in (3.4), we have

$$\begin{aligned} \beta_k = & \frac{(y_k^T g_{k+1}) (y_k^T d_k)}{t^2 (z_k^T d_k)^2 + (y_k^T d_k)^2} - \frac{t (s_k^T g_{k+1}) (z_k^T d_k)}{t^2 (z_k^T d_k)^2 + (y_k^T d_k)^2} + \frac{t^2 (z_k^T g_{k+1}) (z_k^T d_k)}{t^2 (z_k^T d_k)^2 + (y_k^T d_k)^2} \\ & - \frac{\|t^2 (z_k^T s_k) z_k + (y_k^T s_k) y_k - \lambda_k (z_k^T s_k) s_k\|^2}{4\gamma_2 \left[t^2 (z_k^T s_k)^2 + (y_k^T s_k)^2 \right]} \frac{g_{k+1}^T d_k}{t^2 (z_k^T d_k)^2 + (y_k^T d_k)^2}. \end{aligned} \quad (3.12)$$

4. GLOBAL CONVERGENCE FOR STRONGLY CONVEX FUNCTIONS

We are now in a position where we can sum up the contents of the previous sections to introduce our new conjugate gradient algorithm.

Algorithm 4.1. Modified Secant Conjugate Gradient Algorithm (MSCG).

- Step 1. Choose a starting point $x_0 \in \mathbb{R}^n$ and a suitable value for the positive parameters $\rho_k, T, 0 < c_1 < c_2 < 1$ and $\gamma_1 + \gamma_2 < 1$. Compute $g_0 = \nabla f(x_0)$, set $d_0 = -g_0$ and $k = 0$.
- Step 2. Check the stopping condition if it is established, then stop; else go to step 3.
- Step 3. Compute the step length α_k using strong Wolfe condition (1.2) and (1.3).
- Step 4. Compute $x_{k+1} = x_k + \alpha_k d_k$, $f_{k+1} = f(x_{k+1})$, $g_{k+1} = \nabla f(x_{k+1})$, $s_k = x_{k+1} - x_k$ and $y_k = g_{k+1} - g_k$.
- Step 5. Select a desirable value of $u_k \in \mathbb{R}^n$ such that $s_k^T u_k \neq 0$, and then compute θ_k and z_k according to (2.3) and (3.1), respectively.
- Step 6. Compute λ_k and t by (3.10) and (3.11).
- Step 7. Compute the conjugate gradient parameter β_k using (3.12).
- Step 8. Compute the search direction $d_{k+1} = -g_{k+1} + \beta_k d_k$.
- Step 9. Set $k = k + 1$ and go to step 2.

Our final task is the investigation of global convergence properties of Algorithm 4.1 (MSCG).

Theorem 4.2. Suppose that f is a strongly convex and bounded below function and the gradient function $g(x)$ is Lipschitz continuous on the level set

$$V = \{x \in \mathbb{R}^n : f(x) \leq f(x_0)\},$$

which is assumed to be bounded. In other words, there exists $L > 0$ such that, for x and y belonging to V ,

$$\|g(x) - g(y)\| \leq L\|x - y\|. \quad (4.1)$$

Furthermore, the generated sequence of iterations, x_k , is bounded.

Consider any descent methods of the form (1.1) where α_k is determined by strong Wolfe line search, then we have that

$$\sum_{k=1}^{\infty} \frac{(g_k^T d_k)^2}{\|d_k\|^2} < \infty.$$

Proof. See [17]. \square

Now using the above theorem and Theorem 3 in [21], we prove the global convergence of the new method for strong convex functions.

Theorem 4.3. *Consider the method presented by Algorithm 4.1, then, under the conditions mentioned in Theorem 4.2 we have that*

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

Proof. Since $f(x)$ is strongly convex, so

$$\mu \|x - y\|^2 \leq (g(x) - g(y))^T (x - y), \quad (4.2)$$

where $\mu > 0$ is a constant. Using (4.2), we have that $\mu \alpha_k \|d_k\|^2 \leq y_k^T d_k$. Moreover, the Cauchy-Schwartz inequality, and the inequalities (5.13) and (5.20) presented in [39] imply that there are $m > 0$ and $L' > 0$ such that:

$$\begin{aligned} \|z_k\| &\leq L' \|s_k\|, \text{ and} \\ z_k^T s_k &\geq m \|s_k\|^2. \end{aligned} \quad (4.3)$$

We show that the search direction d_{k+1} is bounded above. In fact

$$\begin{aligned} |\beta_k| &\leq \frac{\|y_k\|^2 \cdot \|g_{k+1}\| \cdot \|d_k\|}{t^2 (z_k^T d_k)^2 + (y_k^T d_k)^2} + \frac{|t| \cdot \|s_k\| \cdot \|z_k\| \cdot \|g_{k+1}\| \cdot \|d_k\|}{t^2 (z_k^T d_k)^2 + (y_k^T d_k)^2} + \frac{t^2 \cdot \|z_k\|^2 \cdot \|g_{k+1}\| \cdot \|d_k\|}{t^2 (z_k^T d_k)^2 + (y_k^T d_k)^2} \\ &\quad + \frac{\|t^2 (z_k^T s_k) z_k + (y_k^T s_k) y_k - \lambda_k (z_k^T s_k) s_k\|^2}{4\gamma_2 [t^2 (z_k^T s_k)^2 + (y_k^T s_k)^2]} \frac{\|g_{k+1}\| \cdot \|d_k\|}{t^2 (z_k^T d_k)^2 + (y_k^T d_k)^2}. \end{aligned} \quad (4.4)$$

Clearly,

$$t^2 (z_k^T d_k)^2 + (y_k^T d_k)^2 \geq t^2 (z_k^T d_k)^2 \quad (4.5)$$

and

$$t^2 (z_k^T d_k)^2 + (y_k^T d_k)^2 \geq (y_k^T d_k)^2. \quad (4.6)$$

Now from (4.4) to (4.6), and the triangle inequality, we have

$$\begin{aligned} |\beta_k| &\leq \frac{\|y_k\|^2 \cdot \|d_k\| \cdot \|g_{k+1}\|}{(y_k^T d_k)^2} + \frac{|t| \cdot \|z_k\| \cdot \|s_k\| \cdot \|d_k\| \cdot \|g_{k+1}\|}{(y_k^T d_k)^2} + \frac{\|z_k\|^2 \cdot \|d_k\| \cdot \|g_{k+1}\|}{(z_k^T d_k)^2} \\ &\quad + \frac{\|d_k\| \cdot \|g_{k+1}\|}{4\gamma_2 (t^2 (z_k^T d_k)^2 + (y_k^T d_k)^2)} \left[t^2 \|z_k\|^2 + \|y_k\|^2 + \frac{(z_k^T s_k)^2 \|s_k\|^2}{(y_k^T s_k)^2} + \frac{2(y_k^T s_k) \|z_k\| \cdot \|y_k\|}{z_k^T s_k} \right. \\ &\quad \left. + 2\|z_k\| \cdot \|s_k\| + \frac{2(z_k^T s_k) \|y_k\| \cdot \|s_k\|}{y_k^T s_k} \right]. \end{aligned}$$

Therefore,

$$\begin{aligned} \Rightarrow |\beta_k| &\leq \frac{\|y_k\|^2 \cdot \|d_k\| \cdot \|g_{k+1}\|}{(y_k^T d_k)^2} + \frac{|t| \cdot \|z_k\| \cdot \|s_k\| \cdot \|d_k\| \cdot \|g_{k+1}\|}{(y_k^T d_k)^2} + \frac{\|z_k\|^2 \cdot \|d_k\| \cdot \|g_{k+1}\|}{(z_k^T d_k)^2} \\ &\quad + \frac{\|z_k\|^2 \cdot \|d_k\| \cdot \|g_{k+1}\|}{4\gamma_2 (z_k^T d_k)^2} + \frac{\|y_k\|^2 \cdot \|d_k\| \cdot \|g_{k+1}\|}{4\gamma_2 (y_k^T d_k)^2} + \frac{\|z_k\|^2 \cdot \|s_k\|^4 \cdot \|d_k\| \cdot \|g_{k+1}\|}{4\gamma_2 (y_k^T s_k)^2 (y_k^T d_k)^2} \\ &\quad + \frac{\|y_k\|^2 \cdot \|z_k\| \cdot \|s_k\| \cdot \|d_k\| \cdot \|g_{k+1}\|}{2\gamma_2 (z_k^T s_k) (y_k^T d_k)^2} + \frac{\|z_k\| \cdot \|s_k\| \cdot \|d_k\| \cdot \|g_{k+1}\|}{2\gamma_2 (y_k^T d_k)^2} \\ &\quad + \frac{\|z_k\| \cdot \|y_k\| \cdot \|s_k\|^2 \cdot \|d_k\| \cdot \|g_{k+1}\|}{2\gamma_2 (y_k^T s_k) (y_k^T d_k)^2}, \end{aligned}$$

and then, we have from (3.11), (4.1), (4.2) and (4.3) that

$$\begin{aligned} \Rightarrow |\beta_k| &\leq \frac{L^2 \cdot \|s_k\|^2 \cdot \|d_k\| \cdot \|g_{k+1}\|}{\mu^2 \alpha_k^2 \|d_k\|^4} + \frac{T \cdot L' \cdot \|s_k\|^2 \cdot \|d_k\| \cdot \|g_{k+1}\|}{\mu^2 \alpha_k^2 \|d_k\|^4} \\ &\quad + \frac{L'^2 \cdot \|s_k\|^2 \cdot \|d_k\| \cdot \|g_{k+1}\|}{m^2 \alpha_k^2 \|d_k\|^4} + \frac{L'^2 \cdot \|s_k\|^2 \cdot \|d_k\| \cdot \|g_{k+1}\|}{4\gamma_2 m^2 \alpha_k^2 \|d_k\|^4} \\ &\quad + \frac{L^2 \cdot \|s_k\|^2 \cdot \|d_k\| \cdot \|g_{k+1}\|}{4\gamma_2 \mu^2 \alpha_k^2 \|d_k\|^4} + \frac{L'^2 \cdot \|s_k\|^6 \cdot \|d_k\| \cdot \|g_{k+1}\|}{4\gamma_2 \mu^4 \alpha_k^2 \|s_k\|^4 \cdot \|d_k\|^4} \\ &\quad + \frac{L^2 L' \cdot \|s_k\|^4 \cdot \|d_k\| \cdot \|g_{k+1}\|}{2\gamma_2 m \mu^2 \alpha_k^2 \|s_k\|^2 \cdot \|d_k\|^4} + \frac{L' \cdot \|s_k\|^2 \cdot \|d_k\| \cdot \|g_{k+1}\|}{2\gamma_2 \mu^2 \alpha_k^2 \|d_k\|^4} \\ &\quad + \frac{LL' \cdot \|s_k\|^4 \cdot \|d_k\| \cdot \|g_{k+1}\|}{2\gamma_2 \mu^3 \alpha_k^2 \|s_k\|^2 \cdot \|d_k\|^4}. \end{aligned}$$

Finally, we have that

$$\begin{aligned} |\beta_k| &\leq \left(\frac{L^2}{\mu^2} + \frac{TL'}{\mu^2} + \frac{L'^2}{m^2} + \frac{L'^2}{4\gamma_2 m^2} + \frac{L^2}{4\gamma_2 \mu^2} + \frac{L'^2}{4\gamma_2 \mu^4} \right. \\ &\quad \left. + \frac{L^2 L'}{2\gamma_2 m \mu^2} + \frac{L'}{2\gamma_2 \mu^2} + \frac{LL'}{2\gamma_2 \mu^3} \right) \times \frac{\|g_{k+1}\|}{\|d_k\|}. \end{aligned}$$

Now, using the above results, we have

$$\begin{aligned} \|d_{k+1}\| &\leq \|g_{k+1}\| + |\beta_k| \|d_k\| \\ &\leq \left(\frac{L^2}{\mu^2} + \frac{TL'}{\mu^2} + \frac{L'^2}{m^2} + \frac{L'^2}{4\gamma_2 m^2} + \frac{L^2}{4\gamma_2 \mu^2} + \frac{L'^2}{4\gamma_2 \mu^4} \right. \\ &\quad \left. + \frac{L^2 L'}{2\gamma_2 m \mu^2} + \frac{L'}{2\gamma_2 \mu^2} + \frac{LL'}{2\gamma_2 \mu^3} \right) \|g_{k+1}\|, \end{aligned} \tag{4.7}$$

and so the proof is completed, using (3.5), (4.7), and Theorem 4.2. \square

5. GLOBAL CONVERGENCE FOR GENERAL FUNCTIONS

In this section, we use the results obtained in Section 4 of [21], and introduce the following modification of β_k :

$$\beta_k^N = \max(\beta_k, \chi_k) \tag{5.1}$$

where in:

- χ_k is real valued function
- $|\chi_k| \|d_k\|$ is bounded above,
- $\chi_k \leq \frac{\varepsilon \|g_{k+1}\|^2}{g_{k+1}^T d_k}$, for some $\varepsilon < 1$ and $g_{k+1}^T d_k > 0$.

We prove the global convergence of (MSCG) for general functions in follow.

Theorem 5.1. *Assume the method (1.1) and (1.4) with β_k^N defined in (5.1), then for a general function f with the strong Wolfe line search conditions (1.2) and (1.3) and $0 \leq \rho_k < \min \left\{ \frac{1-c_2}{3(1+c_2-2c_1)}, m y_k^T s_k \right\}$, where m is a positive constant, we have*

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

Proof. Following the same steps of the proof of Lemma 4 in [21] and Lemma 1, we have

$$d_{k+1}^T g_{k+1} \leq -\max(1-\varepsilon, 1-\gamma_1-\gamma_2) \|g_{k+1}\|^2. \quad (5.2)$$

If for a subsequence k_j , we have $\beta_{k_j}^N = \chi_{k_j}$, then we reach to the same contradiction as in Theorem 6 in [21] and proof is completed.

If $\beta_k^N = \beta_k$, for sufficiently large k , we have a bound for β_k as follow.

Firstly, from (1.3) and (3.5), we have

$$d_k^T y_k = d_k^T g_{k+1} - d_k^T g_k \geq (c_2 - 1) d_k^T g_k > 0, \quad (5.3)$$

Assuming there exist $\eta_1 > 0$ such that $\|g_k\| > \eta_1$, and using (5.2) and (5.3),

$$d_k^T y_k \geq (1 - c_2) \max(1 - \varepsilon, 1 - \gamma_1 - \gamma_2) \eta_1^2, \quad (5.4)$$

and so

$$\frac{|d_k^T g_{k+1}|}{|d_k^T y_k|} \leq \max\left(\frac{c_2}{1 - c_2}, 1\right), \quad (5.5)$$

see [25]. On the other hand (1.7) and (3.8) imply

$$d_k^T z_k \geq -m_1 (d_k^T g_k) \geq m_1 c \|g_k\|^2 \geq m_1 c \eta_1^2. \quad (5.6)$$

Secondly, we show that there exists a constant $R > 0$ such that

$$|\beta_k| \leq R \|s_k\|. \quad (5.7)$$

Using (3.12), (4.5) and (4.6) we have

$$\begin{aligned} |\beta_k| &\leq \frac{|y_k^T g_{k+1}| |y_k^T d_k|}{(y_k^T d_k)^2} + \frac{|t| |s_k^T g_{k+1}| |z_k^T d_k|}{(y_k^T d_k)^2} + \frac{t^2 |z_k^T g_{k+1}| |z_k^T d_k|}{t^2 (z_k^T d_k)^2} \\ &\quad + \frac{|g_{k+1}^T d_k|}{4\gamma_2 (t^2 (z_k^T d_k)^2 + (y_k^T d_k)^2)} \left(\frac{\|t^2 (z_k^T s_k) z_k\|^2}{t^2 (z_k^T s_k)^2} + \frac{\|(y_k^T s_k) y_k\|^2}{(y_k^T s_k)^2} + \frac{\|(z_k^T s_k) s_k\|^2}{(y_k^T s_k)^2} \right. \\ &\quad \left. + \frac{2\|t^2 (z_k^T s_k) z_k\| \| (y_k^T s_k) y_k\|}{t^2 (z_k^T s_k)^2} + \frac{2\|t^2 (z_k^T s_k) z_k\| \| (z_k^T s_k) s_k\|}{t^2 (z_k^T s_k)^2} + \frac{2\| (y_k^T s_k) y_k\| \| (z_k^T s_k) s_k\|}{(y_k^T s_k)^2} \right) \\ \Rightarrow |\beta_k| &\leq \frac{|y_k^T g_{k+1}|}{y_k^T d_k} + \frac{|t| |s_k^T g_{k+1}| |z_k^T d_k|}{(y_k^T d_k)^2} + \frac{|z_k^T g_{k+1}|}{z_k^T d_k} \\ &\quad + \frac{1}{4\gamma_2} \left(\|z_k\|^2 \times \frac{|g_{k+1}^T d_k|}{(z_k^T d_k)^2} + \|y_k\|^2 \times \frac{|g_{k+1}^T d_k|}{(y_k^T d_k)^2} + \frac{(z_k^T s_k)^2 \|s_k\|^2}{(y_k^T s_k)^2} \times \frac{|g_{k+1}^T d_k|}{(y_k^T d_k)^2} \right. \\ &\quad \left. + \frac{2\|z_k\| \|y_k\|}{(z_k^T d_k)} \times \frac{|g_{k+1}^T d_k|}{(y_k^T d_k)} + 2\|z_k\| \|s_k\| \times \frac{|g_{k+1}^T d_k|}{(y_k^T d_k)^2} + \frac{2(z_k^T s_k) \|y_k\| \|s_k\|}{y_k^T s_k} \times \frac{|g_{k+1}^T d_k|}{(y_k^T d_k)^2} \right) \\ \Rightarrow |\beta_k| &\leq \frac{1}{y_k^T d_k} \left(|y_k^T g_{k+1}| + |t| |s_k^T g_{k+1}| \times \frac{|z_k^T s_k|}{(y_k^T s_k)} + \frac{\|y_k\|^2}{4\gamma_2} \times \frac{|g_{k+1}^T d_k|}{y_k^T d_k} + \frac{\|s_k\|^2}{4\gamma_2} \times \frac{(z_k^T s_k)^2}{(y_k^T s_k)^2} \times \frac{|g_{k+1}^T d_k|}{y_k^T d_k} \right. \\ &\quad \left. + \frac{\|z_k\| \|s_k\|}{2\gamma_2} \times \frac{|g_{k+1}^T d_k|}{y_k^T d_k} + \frac{\|y_k\| \|s_k\|}{2\gamma_2} \times \frac{|z_k^T s_k|}{(y_k^T s_k)} \times \frac{|g_{k+1}^T d_k|}{y_k^T d_k} \right) \\ &\quad + \frac{1}{z_k^T d_k} \left(|z_k^T g_{k+1}| + \frac{\|z_k\|^2}{4\gamma_2} \times \frac{|g_{k+1}^T d_k|}{z_k^T d_k} + \frac{\|z_k\| \|y_k\|}{2\gamma_2} \times \frac{|g_{k+1}^T d_k|}{y_k^T d_k} \right). \end{aligned}$$

Now using mean value theorem

$$\begin{aligned}\theta_k &= 6(f_k - f_{k+1}) + 3(\nabla f_k + \nabla f_{k+1})^T s_k \\ &= 6\nabla f(\eta_k)^T (x_k - x_{k+1}) + 3(\nabla f_k + \nabla f_{k+1})^T s_k \\ &= -6\nabla f(\eta_k)^T s_k + 3(\nabla f_k + \nabla f_{k+1})^T s_k \\ &= 3(\nabla f_k - \nabla f(\eta_k) + \nabla f_{k+1} - \nabla f(\eta_k))^T s_k,\end{aligned}$$

where $\eta_k = \omega x_k + (1 - \omega)x_{k+1}$ and $\omega \in (0, 1)$. Thus, we have using (4.1) that

$$\begin{aligned}|\theta_k| &\leq 3(L\|x_k - \eta_k\| + L\|x_{k+1} - \eta_k\|) \|s_k\| \\ &= 3(L(1 - \omega)\|x_k - x_{k+1}\| + L\omega\|x_{k+1} - x_k\|) \|s_k\| \\ &= 3L\|s_k\|^2.\end{aligned}\tag{5.8}$$

Now using (3.1) and (5.8), we have

$$\left| \frac{z_k^T s_k}{y_k^T s_k} \right| = \left| \frac{y_k^T s_k + \rho_k \frac{\theta_k}{s_k^T u_k} s_k^T u_k}{y_k^T s_k} \right| = \left| 1 + \frac{\rho_k \theta_k}{y_k^T s_k} \right| < 1 + |m\theta_k| < 1 + 3Lm\|s_k\|^2,\tag{5.9}$$

and

$$\|z_k\| = \left\| y_k + \rho_k \frac{\theta_k}{s_k^T u_k} u_k \right\| \leq \|y_k\| + m_1 |\theta_k| \|u_k\| \leq L\|s_k\| + 3Lm\|s_k\|^2 \|u_k\|,$$

where m_1 is an appropriate upper bound obtained by considering different values of u_k as s_k and y_k .

As a consequence, there exist a constant $L' > 0$ such that

$$\|z_k\| \leq L'\|s_k\|. \tag{5.10}$$

Thus, using (3.11), (4.1), (5.9), (5.10), Cauchy–Schwartz inequality and the assumption that η_2 is an upper bound on $\|g_k\|$, we have

$$\begin{aligned}|\beta_k| &\leq \frac{1}{y_k^T d_k} \left(L\eta_2\|s_k\| + T\eta_2(1 + 3Lm\|s_k\|^2)\|s_k\| + \frac{L^2\|s_k\|^2}{4\gamma_2} \times \frac{|g_{k+1}^T d_k|}{y_k^T d_k} \right. \\ &\quad + \frac{\|s_k\|^2}{4\gamma_2} \times (1 + 3Lm\|s_k\|^2)^2 \times \frac{|g_{k+1}^T d_k|}{y_k^T d_k} + \frac{L'\|s_k\|^2}{2\gamma_2} \times \frac{|g_{k+1}^T d_k|}{y_k^T d_k} \\ &\quad \left. + \frac{L\|s_k\|^2}{2\gamma_2} \times (1 + 3Lm\|s_k\|^2) \times \frac{|g_{k+1}^T d_k|}{y_k^T d_k} \right) \\ &\quad + \frac{1}{z_k^T d_k} \left(L'\eta_2\|s_k\| + \frac{L'^2\|s_k\|^2}{4\gamma_2} \times \frac{|g_{k+1}^T d_k|}{y_k^T d_k} + \frac{LL'\|s_k\|^2}{2\gamma_2} \times \frac{|g_{k+1}^T d_k|}{y_k^T d_k} \right).\end{aligned}\tag{5.11}$$

We can see from (5.4)–(5.6), (5.11), and Theorem 4.2 that there exist a constant $R > 0$ such that (5.7) holds.

Finally, following the steps of the proof of Theorem 3.2 in [25], the proof is completed. \square

6. NUMERICAL EXPERIMENT AND COMPARISONS

In this section, we consider the iterative scheme (1.1) and (1.4) with five different values of β_k :

- β_k^{New} : The method presented in Algorithm 4.1 (MSCG) with parameter $\rho = 0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1$.

TABLE 1. Test problems collection.

Function	<i>n</i>	Function	<i>n</i>	Function	<i>n</i>
AKIVA	2	DIXMAANH	3000	QUARTC	5000
ALLINITU	4	DJTL	2	ROSENBR	2
ARGLBLE	200	DQDRTIC	5000	S308	2
ARGLCLE	200	ECKERLE4LS	3	SBRYBND	5000
ARGLINA	200	EG2	1000	SCHMVETT	5000
ARGLINB	200	ENSOLS	9	SENSORS	100
ARGLINC	200	EXTROSNB	1000	SINEVAL	2
ARWHEAD	5000	FMINSURF	5625	SISSER	2
BARD	3	GENROSE	500	SPARSQUR	10 000
BEALE	2	HAIRY	2	SPMSRTLS	4999
BIGGS6	6	HILBERTA	2	SROSENBR	5000
BOX3	3	HILBERTB	10	TOINTGOR	50
BROWNAL	200	HUMPS	2	TOINTGSS	5000
BROYDN3DLS	10	INTEQNELS	12	TOINTQOR	50
BRYBND	500	JENSMP	2	TQUARTIC	5000
CHAINWOO	4000	KOWOSB	4	VARDIM	200
CHNROSNB	50	KOWOSBNE	4	VAREIGVL	50
CHNRSNBM	50	LIARWHD	5000	WOODS	4000
COSINE	10 000	LOGHAIRY	2	YFITU	3
DANIWOODLS	2	LSC1LS	3	ZANGWIL2	2
DANWOODLS	2	MANCINO	100		
DENSCHA	2	MARATOSB	2		
DENSCHB	2	MGH09LS	4		
DIXMAANA	3000	MGH10LS	3		
DIXMAANB	3000	MODBEALE	2000		
DIXMAANC	3000	MOREBV	5000		
DIXMAAND	3000	NONDIA	5000		
DIXMAANE	3000	PENALTY1	1000		
DIXMAANF	3000	POWELLSG	5000		
DIXMAANG	3000	POWER	10 000		

- β_k^F : The method introduced by Fatemi in [21].
- β_k^{YT} : The method introduced by Yabe and Takano in [39] with $(\rho = 0.5, t = 0.3)$ and $(\rho = 0.7, t = 0.5)$.
- β_k^{BG1} : The parameter (3.21) introduced by Babaie-kafaki *et al.* [12] with $t = 0.05005$.
- β_k^{BG2} : The parameter (3.53) introduced by Babaie-kafaki *et al.* [12] with $t = 0.05010$.

It should be noted that all these parameters belong to the Dai–Liao family and all of them use the modified secant condition, except β_k^F .

The following points were taken into account in the implementation of the algorithms.

- We use the CG-DESCENT line search procedure with initial parameters as reported in [26].
- We consider the two different choices of u_k ; namely s_k and y_k .
- The algorithms are terminated if

$$\|\nabla f(x_k)\|_\infty \leq \max \left(10^{-6}, 10^{-12} \|\nabla f(x_1)\|_\infty \right)$$

or the number of function evaluations are exceeded 3000.

- We choose $\gamma_1 = 0.01$ and $\gamma_2 = 0.98$ in our numerical tests.

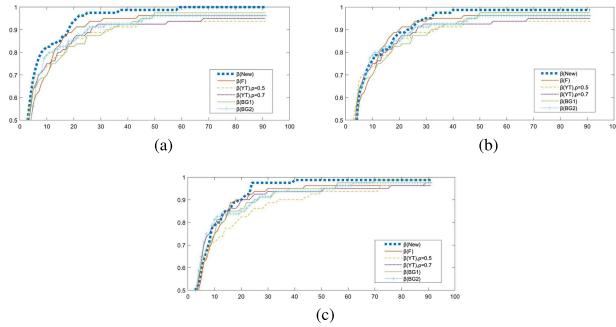


FIGURE 1. Number of function evaluations performance profile. (a) $\rho = 0.7$, $u_k = s_k$. (b) $\rho = 0.8$, $u_k = s_k$. (c) $\rho = 0.8$, $u_k = y_k$.

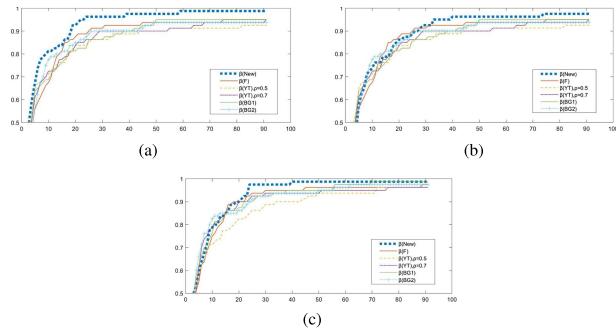


FIGURE 2. Number of gradient evaluations performance profile. (a) $\rho = 0.7$, $u_k = s_k$. (b) $\rho = 0.8$, $u_k = s_k$. (c) $\rho = 0.8$, $u_k = y_k$.

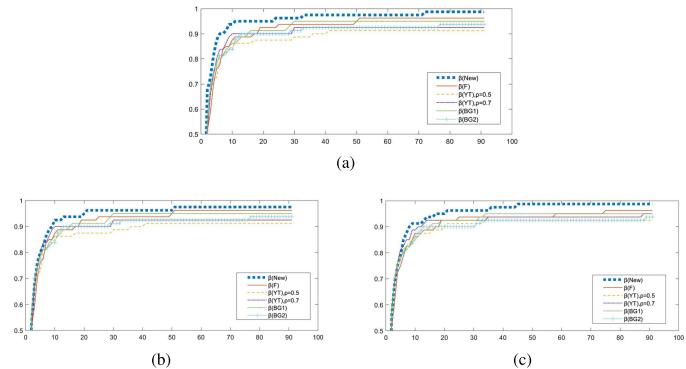


FIGURE 3. CPU time performance profile. (a) $\rho = 0.7$, $u_k = s_k$. (b) $\rho = 0.8$, $u_k = s_k$. (c) $\rho = 0.8$, $u_k = y_k$.

To examine and compare the algorithm's efficiency, we considered 80 problems of CUTEr collection [24] with different dimensions, as presented in Table 1. Since conjugate gradient methods effectively solve large scale problems, we tried to list problems with large dimensions. Moreover, we have also selected some problems with low dimensions, because we found that they were hard to solve for some algorithms due to their ill-condition nature.

The tests were done on a 2.4 Intel Core(TM) i7-5500U processor computer with 8 GB of RAM with Linux operating system on MATLAB R2015a programming environment.

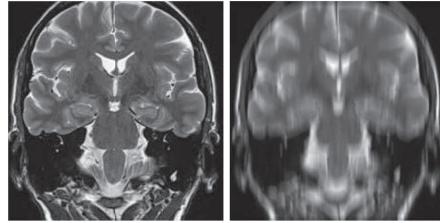
FIGURE 4. The 250×250 original and blurred/noisy Brain image.

TABLE 2. PSNR results of Tikhonov regularization.

Conjugate gradient parameters	Brain's image	Cameraman's image
$\beta_k^{\text{New}}, \rho = 0.7, u = s$	29.7078	24.0688
$\beta_k^{\text{New}}, \rho = 0.8, u = s$	29.7144	24.0658
$\beta_k^{\text{New}}, \rho = 0.8, u = y$	29.7214	24.0377
β_k^F	28.9451	24.031
$\beta_k^{\text{YT}}, \rho = 0.5, u = s$	29.7015	24.1205
$\beta_k^{\text{YT}}, \rho = 0.5, u = y$	29.7466	23.9811
$\beta_k^{\text{YT}}, \rho = 0.7, u = s$	29.6733	24.1912
$\beta_k^{\text{YT}}, \rho = 0.7, u = y$	29.6603	24.1815
β_k^{BG1}	29.6625	24.1733
β_k^{BG2}	29.7295	24.0446
β_k^{HS}	29.6454	23.7595
β_k^{FR}	29.0431	23.1993
β_k^{PRP}	28.9495	23.1458
β_k^{CD}	23.9086	19.4664
β_k^{LS}	29.0302	23.0885
β_k^{DY}	29.5452	23.8327
β_k^{HZ}	29.1291	23.2477

The algorithms equipped with β_k^{New} and β_k^F successfully solved all the test problems, while with the others, we faced with a NaN (Not-a-Number) for DANWOODLS problem. Figures 1–3 show the performance profile for the number of function evaluations, the number of gradient evaluations and CPU time for all algorithms. By examining them, we find that β_k^{New} is more efficient than the other algorithms for many different choices of ρ , especially when $u_k = s_k$. We report $\rho = 0.7$ and 0.8 as the best choices of ρ in our algorithm.

Here, to save space, we decided to post the detail of the results in publication tab of the link "<https://wp.kntu.ac.ir/smfatemi/>" (We investigated the results of each algorithm on every problem for about 20 different parameters).

We also studied image deblurring problem (1.9) in details. We considered some smooth nonlinear regularization terms as well as Tikhonov regularization. We blurred some 250×250 images using Toeplitz matrices. In Figure 4, a medical image has blurred with the linear motion by 15 pixels, and in Figure 8, the cameraman image has blurred with a Gaussian smoothing kernel with 4 deviation. According to [33], the best value of the regularization parameter λ is around 0.001.

Regarding the peak signal-to-noise ratio (PSNR), which usually is defined as the ratio between the maximum possible value of a signal and the power of distorting noise that affects the quality of its representation, it can be seen from Tables 2–4 that on average, the new algorithm outperforms the other algorithms, especially, in the cases of nonlinear regularization terms. The bold values in these tables show the good performance of our algorithm, and possibly some exceptions, in comparison with the other algorithms. Figures 5–7, and 9–11 show the results corresponding to the Tikhonov regularization choice. In other words, β^{New} recovers images in an acceptable quality. Consequently, the method with a proper choice of

TABLE 3. PSNR Results of Brain's image.

Conjugate gradient parameters	$\phi(x) = \sum_{i=1}^n \frac{x_i}{1+x_i^2}$	$\phi(x) = \sum_{i=1}^n \log(1+x_i^2)$
$\beta_k^{\text{New}}, \rho = 0.7, u = s$	20.3239	20.9590
$\beta_k^{\text{New}}, \rho = 0.8, u = s$	20.3148	21.4276
$\beta_k^{\text{New}}, \rho = 0.8, u = y$	20.3131	20.9367
β_k^F	19.2218	20.8551
$\beta_k^{\text{YT}}, \rho = 0.5, u = s$	20.1350	18.3449
$\beta_k^{\text{YT}}, \rho = 0.5, u = y$	20.1188	18.8039
$\beta_k^{\text{YT}}, \rho = 0.7, u = s$	20.1317	19.5504
$\beta_k^{\text{YT}}, \text{rho} = 0.7, u = y$		
β_k^{BG1}	20.3330	20.1586
β_k^{BG2}	9.7171	20.3707
β_k^{HS}	20.4205	20.2007
β_k^{FR}	9.7005	NaN
β_k^{PRP}	20.3105	20.4720
β_k^{CD}	18.8203	18.9510
β_k^{LS}	20.3201	20.4633
β_k^{DY}	19.9363	19.6200
β_k^{HZ}	18.4660	19.8873

TABLE 4. PSNR Results of Cameraman's image.

Conjugate gradient parameters	$\phi(x) = \sum_{i=1}^n \frac{x_i}{1+x_i^2}$	$\phi(x) = \sum_{i=1}^n \log(1+x_i^2)$
$\beta_k^{\text{New}}, \rho = 0.7, u = s$	21.0252	19.3449
$\beta_k^{\text{New}}, \rho = 0.8, u = s$	21.3824	21.7894
$\beta_k^{\text{New}}, \rho = 0.8, u = y$	21.0178	20.8595
β_k^F	20.0459	20.6611
$\beta_k^{\text{YT}}, \rho = 0.5, u = s$	20.9315	14.1041
$\beta_k^{\text{YT}}, \rho = 0.5, u = y$	20.9191	16.3832
$\beta_k^{\text{YT}}, \rho = 0.7, u = s$	20.9462	12.8312
$\beta_k^{\text{YT}}, \text{rho} = 0.7, u = y$	20.9366	12.8099
β_k^{BG1}	20.8959	17.9764
β_k^{BG2}	20.8729	21.7695
β_k^{HS}	21.0225	19.6434
β_k^{FR}	5.5962	NaN
β_k^{PRP}	20.8842	21.7782
β_k^{CD}	21.2427	19.0937
β_k^{LS}	20.8336	21.7804
β_k^{DY}	18.7045	18.6832
β_k^{HZ}	20.1129	17.1293

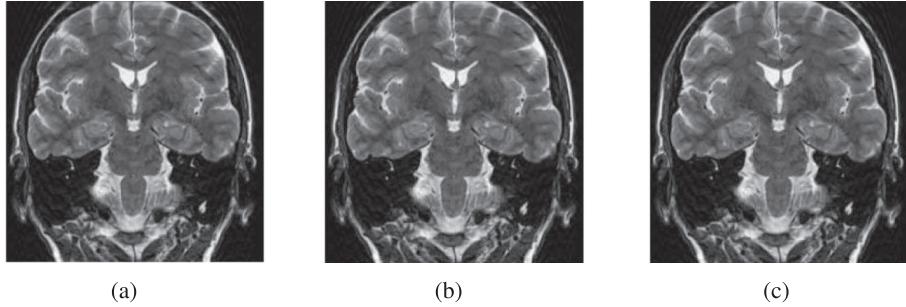


FIGURE 5. Deblurring image with β^{New} . (a) $\rho = 0.7$ and $u_k = s_k$, PSNR = 29.7078. (b) $\rho = 0.8$ and $u_k = s_k$, PSNR = 29.7144. (c) $\rho = 0.8$ and $u_k = y_k$, PSNR = 29.7214.

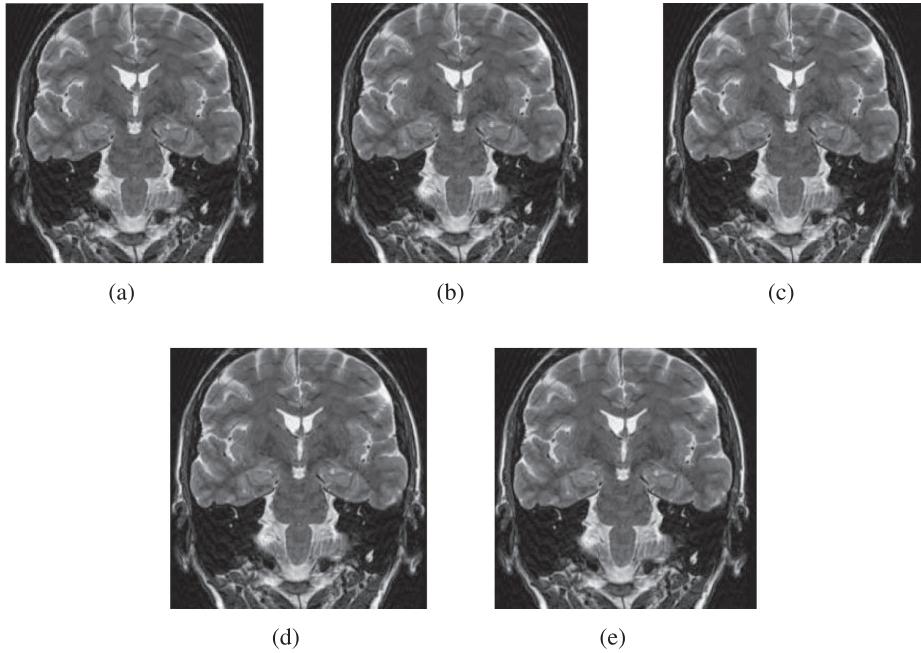


FIGURE 6. Deblurring image with β^F (image (a)), β^{YT} (image (b) and (c)), β^{BG1} (image (d)) and β^{BG2} (image (e)). (a) PSNR = 28.9451. (b) $\rho = 0.7$, $u_k = s_k$, PSNR = 29.6733. (c) $\rho = 0.7$, $u_k = y_k$, PSNR = 29.6603. (d) PSNR = 29.6625. (e) PSNR = 29.7295.

parameters is also effective in solving real-world problems. We also posted the figures for the other regularization terms on the website.

7. CONCLUSION

We presented an effective conjugate gradient method. Our method satisfies the sufficient descent condition and is globally convergent. Numerical results showed that in practice, the method has good performance and outperforms some well-known methods. We also examined the ability of our method to solve some real-world applications. In this way, a common problem from image processing applications was considered. We showed that the image recovered by our method is acceptable in the sense of PSNR test.

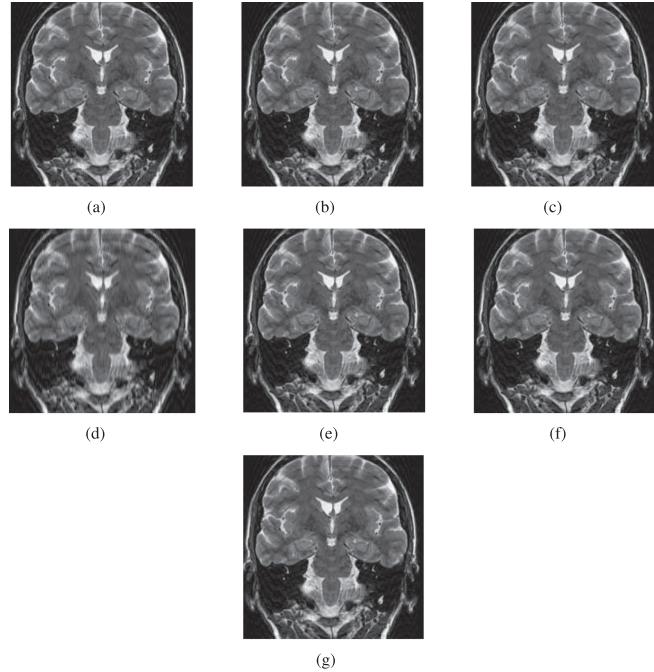


FIGURE 7. Deblurring image with other popular β . (a) β^{HS} , PSNR = 29.6454. (b) β^{FR} , PSNR = 29.0431. (c) β^{PRP} , PSNR = 28.9495. (d) β^{CD} , PSNR = 23.9086. (e) β^{LS} , PSNR = 29.0302. (f) β^{DY} , PSNR = 29.5452. (g) β^{HZ} , PSNR = 29.1291.



FIGURE 8. The 250×250 original and blurred/noisy cameraman image.

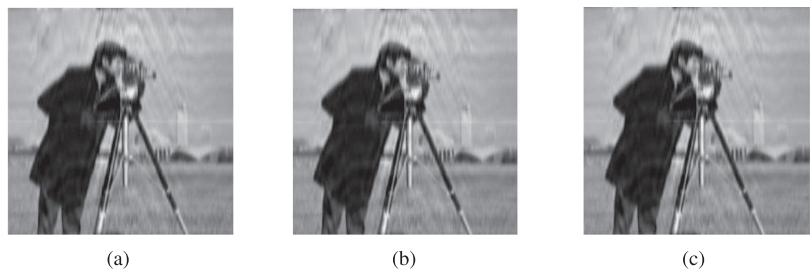


FIGURE 9. Deblurring image with β^{New} . (a) $\rho = 0.7$ and $u_k = s_k$, PSNR = 24.0688. (b) $\rho = 0.8$ and $u_k = s_k$, PSNR = 24.0658. (c) $\rho = 0.8$ and $u_k = y_k$, PSNR = 24.0377.

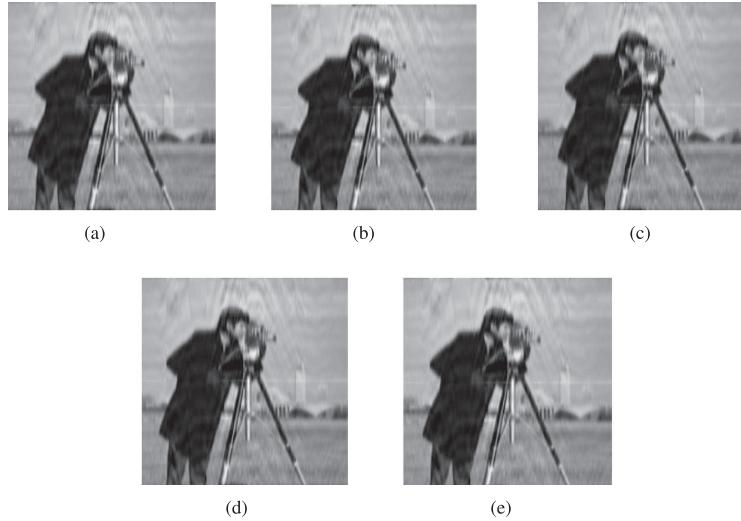


FIGURE 10. Deblurring image with β^F (image (a)), β^{YT} (image (b) and (c)), β^{BG1} (image (d)) and β^{BG2} (image (e)). (a) PSNR = 24.0310. (b) $\rho = 0.5$, $u_k = s_k$, PSNR = 24.1205. (c) $\rho = 0.5$, $u_k = y_k$, PSNR = 23.9811. (d) PSNR = 24.1733. (e) PSNR = 24.0446.

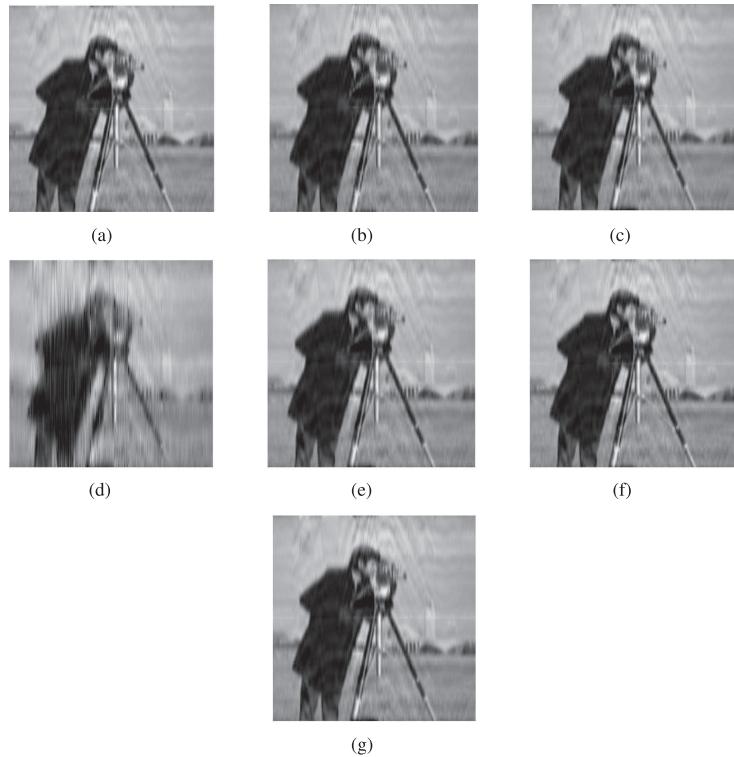


FIGURE 11. Deblurring image with other popular β . (a) β^{HS} , PSNR = 23.7595. (b) β^{FR} , PSNR = 23.1993. (c) β^{PRP} , PSNR = 23.1458. (d) β^{CD} , PSNR = 19.4664. (e) β^{LS} , PSNR = 23.0885. (f) β^{DY} , PSNR = 23.8327. (g) β^{HZ} , PSNR = 23.2477.

Acknowledgements. The author also thank the Research Council of K. N. Toosi University of Technology for supporting this work and sincerely appreciate the helpful comments and suggestions provided by anonymous referees.

REFERENCES

- [1] K. Amini and A.G. Rizi, A new structured quasi-newton algorithm using partial information on hessian. *J. Comput. Appl. Math.* **234** (2010) 805–811.
- [2] N. Andrei, Numerical comparison of conjugate gradient algorithms for unconstrained optimization. *Stud. Inf. Control* **16** (2007) 333–352.
- [3] N. Andrei, Conjugate gradient algorithms for unconstrained optimization. A survey on their definition. Technical report, ICI Technical Report (2008).
- [4] N. Andrei, Accelerated conjugate gradient algorithm with modified secant condition for unconstrained optimization. *Stud. Inf. Control* **18** (2009) 211–232.
- [5] N. Andrei, Accelerated hybrid conjugate gradient algorithm with modified secant condition for unconstrained optimization. *Numer. Algorithms* **54** (2010) 23–46.
- [6] N. Andrei, A numerical study on efficiency and robustness of some conjugate gradient algorithms for large-scale unconstrained optimization. *Stud. Inf. Control* **22** (2013) 259–284.
- [7] N. Andrei, A Dai–Liao conjugate gradient algorithm with clustering of eigenvalues. *Numer. Algorithms* **77** (2018) 1273–1282.
- [8] S. Babaie-Kafaki, Two modified scaled nonlinear conjugate gradient methods. *J. Comput. Appl. Math.* **261** (2014) 172–182.
- [9] S. Babaie-Kafaki and R. Ghanbari, The Dai–Liao nonlinear conjugate gradient method with optimal parameter choices. *Eur. J. Oper. Res.* **234** (2014) 625–630.
- [10] S. Babaie-Kafaki and R. Ghanbari, A descent family of Dai–Liao conjugate gradient methods. *Optim. Methods Softw.* **29** (2014) 583–591.
- [11] S. Babaie-Kafaki and N. Mahdavi-Amiri, Two modified hybrid conjugate gradient methods based on a hybrid secant equation. *Math. Model. Anal.* **18** (2013) 32–52.
- [12] S. Babaie-Kafaki, R. Ghanbari and N. Mahdavi-Amiri, Two new conjugate gradient methods based on modified secant equations. *J. Comput. Appl. Math.* **234** (2010) 1374–1386.
- [13] S. Babaie-Kafaki, M. Fatemi and N. Mahdavi-Amiri, Two effective hybrid conjugate gradient algorithms based on modified bfgs updates. *Numer. Algorithms* **58** (2011) 315–331.
- [14] A.C. Bovik, *Handbook of Image and Video Processing*. Academic Press (2010).
- [15] A.M. Bruckstein, D.L. Donoho and M. Elad, From sparse solutions of systems of equations to sparse modeling of signals and images. *SIAM Rev.* **51** (2009) 34–81.
- [16] Y.-H. Dai and L.-Z. Liao, New conjugacy conditions and related nonlinear conjugate gradient methods. *Appl. Math. Optim.* **43** (2001) 87–101.
- [17] Y.-H. Dai and Y. Yuan, A nonlinear conjugate gradient method with a strong global convergence property. *SIAM J. Optim.* **10** (1999) 177–182.
- [18] E.D. Dolan and J.J. Moré, Benchmarking optimization software with performance profiles. *Math. Program.* **91** (2002) 201–213.
- [19] X.L. Dong, H.W. Liu and Y.B. He, New version of the three-term conjugate gradient method based on spectral scaling conjugacy condition that generates descent search direction. *Appl. Math. Comput.* **269** (2015) 606–617.
- [20] X. Dong, D. Han, Z. Dai, L. Li and J. Zhu, An accelerated three-term conjugate gradient method with sufficient descent condition and conjugacy condition. *J. Optim. Theory Appl.* **179** (2018) 944–961.
- [21] M. Fatemi, A new efficient conjugate gradient method for unconstrained optimization. *J. Comput. Appl. Math.* **300** (2016) 207–216.
- [22] R. Fletcher, *Practical Methods of Optimization*. John Wiley & Sons, New York, NY (2013).
- [23] R. Fletcher and C.M. Reeves, Function minimization by conjugate gradients. *Comput. J.* **7** (1964) 149–154.
- [24] N.I. Gould, D. Orban and P.L. Toint, CUTEr and SifDec: A constrained and unconstrained testing environment, revisited. *ACM Trans. Math. Soft.* **29** (2003) 373–394.
- [25] W.W. Hager and H. Zhang, A new conjugate gradient method with guaranteed descent and an efficient line search. *SIAM J. Optim.* **16** (2005) 170–192.
- [26] W.W. Hager and H. Zhang, Algorithm 851: CG_DESCENT, a conjugate gradient method with guaranteed descent. *ACM Trans. Math. Softw.* **32** (2006) 113–137.
- [27] W.W. Hager and H. Zhang, A survey of nonlinear conjugate gradient methods. *Pac. J. Optim.* **2** (2006) 35–58.
- [28] P.C. Hansen, J.G. Nagy and D.P. O’leary, *Deblurring Images: Matrices, Spectra, and Filtering*. Vol. 3 of: *Fundamentals of Algorithms*. SIAM (2006).
- [29] M.R. Hestenes and E. Stiefel, Methods of Conjugate Gradients for Solving Linear Systems. *J. Res. Nat. Bur. Standards* **49** (1952) 2379.
- [30] X. Jiang and J. Jian, Improved Fletcher–Reeves and Dai–Yuan conjugate gradient methods with the strong wolfe line search. *J. Comput. Appl. Math.* **348** (2019) 525–534.
- [31] G. Li, C. Tang and Z. Wei, New conjugacy condition and related new conjugate gradient methods for unconstrained optimization. *J. Comput. Appl. Math.* **202** (2007) 523–539.
- [32] Y. Liu and C. Storey, Efficient generalized conjugate gradient algorithms, part 1: theory. *J. Optim. Theory Appl.* **69** (1991) 129–137.
- [33] Y. Lu, Out-of-focus Blur: Image De-blurring. Preprint: [arXiv:1710.00620](https://arxiv.org/abs/1710.00620) (2017).
- [34] Y. Narushima and H. Yabe, Conjugate gradient methods based on secant conditions that generate descent search directions for unconstrained optimization. *J. Comput. Appl. Math.* **236** (2012) 4303–4317.
- [35] B.T. Polyak, The conjugate gradient method in extremal problems. *Comput. Math. Math. Phys.* **9** (1969) 94–112.
- [36] F. Vankawala, A. Ganatra and A. Patel, A survey on different image deblurring techniques. *Int. J. Comput. Math.* **116** (2015) 15–18.

- [37] Z. Wei, G. Li and L. Qi, New quasi-newton methods for unconstrained optimization problems. *Appl. Math. Comput.* **175** (2006) 1156–1188.
- [38] Y.-W. Wen, M.K. Ng and W.-K. Ching, Iterative algorithms based on decoupling of deblurring and denoising for image restoration. *SIAM J. Sci. Comput.* **30** (2008) 2655–2674.
- [39] H. Yabe and M. Takano, Global convergence properties of nonlinear conjugate gradient methods with modified secant condition. *Comput. Optim. Appl.* **28** (2004) 203–225.
- [40] J. Zhang and C. Xu, Properties and numerical performance of quasi-Newton methods with modified quasi-Newton equations. *J. Comput. Appl. Math.* **137** (2001) 269–278.
- [41] J. Zhang, N. Deng and L. Chen, New quasi-newton equation and related methods for unconstrained optimization. *J. Optim. Theory Appl.* **102** (1999) 147–167.