A NEW FAMILY OF DAI-LIAO CONJUGATE GRADIENT METHODS WITH MODIFIED SECANT EQUATION FOR UNCONSTRAINED OPTIMIZATION

YUTAO ZHENG*

Abstract. In this paper, a new family of Dai-Liao–type conjugate gradient methods are proposed for unconstrained optimization problem. In the new methods, the modified secant equation used in [H. Yabe and M. Takano, Comput. Optim. Appl. 28 (2004) 203–225] is considered in Dai and Liao’s conjugacy condition. Under some certain assumptions, we show that our methods are globally convergent for general functions with strong Wolfe line search. Numerical results illustrate that our proposed methods can outperform some existing ones.

Mathematics Subject Classification. 65K05, 90C26, 90C30.

1. Introduction

Consider the following unconstrained optimization problem

\[ \min f(x), \ x \in \mathbb{R}^n, \] (1.1)

where the objective function \( f : \mathbb{R}^n \to \mathbb{R} \) is continuously differentiable and its gradient \( g(x) \) is available. The problem (1.1) has a wide range of applications in areas of scientific computing and engineering. Therefore, its efficient and effective numerical solution methods have been intensively studied in the literature, including the spectral gradient methods [5, 15], conjugate gradient methods [4, 13] and memoryless BFGS methods [16]. Among them, conjugate gradient methods are popular and efficient for solving (1.1), especially for large scale problems.

Let \( x_k \) be the \( k \)th iterate point, \( g_k \) the gradient of \( f(x) \) at \( x_k, \) i.e. \( g_k = g(x_k). \) The (nonlinear) conjugate gradient method is given by

\[ x_{k+1} = x_k + \alpha_k d_k, \] (1.2)

where \( \alpha_k \) is the step length computed by carrying out an one-dimension line search and \( d_k \) is the search direction defined by

\[ d_k = \begin{cases} -g_k, & \text{if } k = 0, \\ -g_k + \beta_k d_{k-1}, & \text{if } k \geq 1, \end{cases} \] (1.3)

Keywords. Conjugate gradient method, Dai-Liao–type method, modified secant equation.

School of Mathematics and Information Science, Henan Normal University, Xinxiang 453007, PR China.

* Corresponding author: zhengyutao@htu.edu.cn

© The authors. Published by EDP Sciences, ROADEF, SMAI 2021

This is an Open Access article distributed under the terms of the Creative Commons Attribution License (https://creativecommons.org/licenses/by/4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.
where $\beta_k$ is a scalar.

Since exact line search for searching $\alpha_k$ is usually expensive and impractical, the strong Wolfe inexact line search is often considered in the convergence analysis and implementation of nonlinear conjugate gradient methods. It aims to find a step size $\alpha_k$ satisfying the following two strong Wolfe conditions

\begin{align}
 f(x_k + \alpha_k d_k) &\leq f(x_k) + \rho g_k^T d_k, \quad (1.4) \\
 |g_{k+1}^T d_k| &\leq \sigma |g_k^T d_k|, \quad (1.5)
\end{align}

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

Nonlinear conjugate gradient method for unconstrained optimization problem is generated from the linear conjugate gradient method for a special quadratic minimization problem

\[
 \min \frac{1}{2} x^T Q x + b^T x + c
\]

or its equivalent line system $Q x = b$, where $Q$ is a real symmetric positive definite matrix. Linear conjugate gradient methods generate a search direction such that the conjugacy condition holds, namely,

\[
 d_k^T Q d_j = 0, \forall i \neq j. \quad (1.6)
\]

For general nonlinear functions, it follows from the mean value theorem that there exists some $\tau \in (0, 1)$ such that

\[
 d_k^T y_{k-1} = \alpha_{k-1} d_k^T \nabla^2 f(x_{k-1} + \tau \alpha_{k-1} d_{k-1}) d_{k-1},
\]

where $y_{k-1} = g_k - g_{k-1}$ denotes the gradient change. Therefore, it is reasonable to replace (1.6) by the following conjugacy condition:

\[
 d_k^T y_{k-1} = 0. \quad (1.7)
\]

Let $d_k = -g_k + \beta_k d_{k-1}$ satisfy the above condition, we have the famous Hestenes-Stiefel formula [14]

\[
 \beta_k^{HS} = \frac{g_k^T y_{k-1}}{y_{k-1}^T d_{k-1}}.
\]


\[
 d_k^T y_{k-1} = t g_k^T s_{k-1}, \quad (1.8)
\]

which leads to the following conjugate gradient parameter

\[
 \beta_k^{DL+} = \max \left\{ \frac{g_k^T y_{k-1}}{y_{k-1}^T d_{k-1}}, 0 \right\} - t \frac{g_k^T s_{k-1}}{d_{k-1}^T y_{k-1}}, \quad (1.9)
\]

where $t > 0$ is a scalar, $s_{k-1} = x_k - x_{k-1}$. Note that the first item has been restricted to be nonnegative like [10]. The DL+ method (1.2)–(1.3) with $\beta_k$ in (1.9) is globally convergent for general functions under the sufficient descent condition

\[
 g_k^T d_k \leq -c \|g_k\|^2, \quad c > 0. \quad (1.10)
\]

and some other suitable conditions, where and hereafter $\| \cdot \|$ denotes the Euclidean norm of vectors.

As a special case of Dai-Liao–type conjugate gradient method, the efficient CG_descent method [12] utilizes a particular $t$. The conjugacy parameter of CG_descent method is

\[
 \beta_k^N = \frac{g_k^T y_{k-1}}{y_{k-1}^T d_{k-1}} - 2 \frac{\|y_{k-1}\|^2}{s_{k-1}^T y_{k-1}} \cdot \frac{g_k^T s_{k-1}}{d_{k-1}^T y_{k-1}}.
\]
Two further developments of the Dai and Liao’s method were made by Yabe and Takano [20] and Li et al. [17] based on different modified secant equations. Some more efficient Dai-Liao–type methods were designed and studied in [2, 9, 22, 23] by using different techniques. In this paper, we further give a new family of Dai-Liao–type conjugate gradient methods for unconstrained optimization problems, including their convergence analysis. Numerical experiments show that our methods can outperform the existing ones.

The rest of this paper is organized as follows. In Section 2, we introduce a new Dai-Liao–type method by modifying the conjugate gradient parameter. Based on the strong Wolfe line search rules, the global convergence for uniformly convex and general objective functions is studied in Section 3 and numerical experiments are performed in Section 4. Finally, in Section 5, we give some conclusions to end this paper.

2. New Dai-Liao–Type Methods

We start with the original Dai and Liao’s method in which the quasi-Newton techniques are used. In the quasi-Newton method, an approximation Hessian $B_k$ is updated such that

$$B_k s_{k-1} = y_{k-1}$$

and the search direction $d_k$ is calculated by

$$d_k = -B_k^{-1}g_k. \quad (2.1)$$

Combining the above two equations, we have

$$d_k^T y_{k-1} = -g_k^T s_{k-1}.$$

The above relation implies that (1.7) holds in case of $g_k^T d_{k-1} = 0$, i.e. the line search is exact. However, in practical numerical algorithms, the inexact line search is adopted instead of exact line search. Dai and Liao suggested the following conjugacy condition:

$$d_k^T y_{k-1} = -t g_k^T s_{k-1}, \quad (t \geq 0).$$

In 2004, Yabe and Takano [20] used the modified secant equation

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

where $\rho_k \in [0, 3]$ and $\theta_k = 2(f_k - f_{k+1}) + (g_k + g_{k+1})^T s_k$, $u_k$ is chosen s.t. $u_k^T s_k \neq 0$, to derive a new conjugacy condition through replacing $y_k$ by $z_k = y_k + \rho_k \theta_k \frac{u_k}{u_k^T s_k}$, the modified conjugacy parameter is

$$\beta_{k+1} = \max \left\{ \frac{g_{k+1}^T z_k}{d_k^T z_k}, 0 \right\} - t \frac{g_{k+1}^T s_k}{d_k^T z_k}.$$

In this paper, we will derive a new conjugacy condition from another view of point. Combining (2.2) with (2.1), we have

$$d_{k+1}^T y_k = d_{k+1}^T \left( B_{k+1} s_k - \rho_k \theta_k \frac{u_k}{u_k^T s_k} \right)$$

$$= d_{k+1}^T B_{k+1} s_k - \rho_k \theta_k \frac{u_k^T d_{k+1}}{u_k^T s_k}$$

$$= -g_{k+1}^T s_k - \rho_k \theta_k \frac{u_k^T d_{k+1}}{u_k^T s_k}.$$
Using the Dai-Liao’s conjugacy condition

\[ d_{k+1}^T y_k = - t g_{k+1}^T s_k \]

and \( d_{k+1} = - g_{k+1} + \beta_{k+1} d_k \), where \( t \in [0, 1] \), we have

\[ \rho_k \theta_k \frac{u_k^T (-g_{k+1} + \beta_{k+1} d_k)}{u_k^T s_k} = (t - 1) g_{k+1}^T s_k, \]

which yields a new conjugate gradient parameter

\[ \beta_{k+1}^{\text{new}} = \left[ \frac{(t - 1) g_{k+1}^T s_k \cdot u_k^T s_k}{\rho_k \theta_k} + g_{k+1}^T u_k \right] / d_k^T y_k \]

\[ = \frac{g_{k+1}^T u_k}{d_k^T u_k} - (1 - t) \frac{u_k^T s_k}{\rho_k \theta_k} \cdot \frac{g_{k+1}^T s_k}{d_k^T y_k} \] if \( \rho_k \theta_k \neq 0 \), otherwise, Dai and Liao’s conjugate gradient parameter \( \beta_{k+1}^{\text{DL+}} \) will be used. According to the experience of the quasi-Newton methods with modified secant equations [21], we choose \( u_k = y_k \).

In the case of \( u_k = y_k \), the conjugacy parameter \( \beta_{k+1}^{\text{new}} \) can be written as

\[ \beta_{k+1}^{\text{new}} = \left[ \frac{(t - 1) g_{k+1}^T s_k \cdot y_k^T s_k}{\rho_k \theta_k} + g_{k+1}^T y_k \right] / d_k^T y_k \]

\[ = \frac{g_{k+1}^T y_k}{d_k^T y_k} + (1 - t) \frac{y_k^T s_k}{\rho_k \theta_k} \cdot \frac{g_{k+1}^T s_k}{d_k^T y_k}. \] (2.3)

and we correct it as

\[ \beta_{k+1}^{\text{new+}} = \max \left\{ \frac{g_{k+1}^T y_k}{d_k^T y_k}, 0 \right\} + \frac{(t - 1) y_k^T s_k}{\rho_k \theta_k} \cdot \frac{g_{k+1}^T s_k}{d_k^T y_k}. \] (2.4)

We call the method (1.2) and (1.3) with \( \beta_k \) given in (2.4) NEW+ method. The corresponding algorithm is given as below:

**Algorithm 2.1.** Improved Dai-Liao conjugate gradient method

- **Step 1:** Given \( x_0 \in \mathbb{R}^n, \varepsilon, \eta > 0 \), set \( d_0 = - g_0, k := 0; \) if \( ||g_0|| \leq \varepsilon \), then stop;

- **Step 2:** Compute \( \alpha_k \) such that strong Wolfe line search (1.4) and (1.5) hold;

- **Step 3:** Let \( x_{k+1} = x_k + \alpha_k d_k \), if \( ||g_{k+1}|| \leq \varepsilon \), then stop;

- **Step 4:** Compute \( \beta_{k+1} \) by (2.4) if \( |\theta_k| > \eta \), otherwise, compute \( \beta_{k+1} \) by (1.9); generate \( d_{k+1} \) by (1.3);

- **Step 5:** Set \( k := k + 1 \) and go to Step 2.

In the rest of the paper, we first analyze the convergence properties of the new algorithm, then give some numerical results which show the modified algorithms are robust and efficient.

**3. Convergence Analysis**

Throughout this section, we assume that \( g_k \neq 0 \) for all \( k \geq 0 \), otherwise a stationary point is found. We first give some standard assumptions.
**Assumption 3.1.** The level set $\mathcal{L} = \{x \in \mathbb{R}^n : f(x) \leq f(x_0)\}$ is bounded, where $x_0 \in \mathbb{R}^n$ is an initial point.

**Assumption 3.2.** In some neighborhood $\mathcal{N}$ of $\mathcal{L}$, the function $f$ is continuously differentiable and its gradient $g(x)$ is Lipschitz continuous, i.e. there exists a positive constant $L > 0$ such that $\|g(x) - g(y)\| \leq L\|x - y\|$ for all $x, y \in \mathcal{N}$.

Assumption 3.1 guarantees that there exists some constant $c$ such that $\|s_k\| \leq 2\bar{c}, \forall k > 0$. Assumption 3.2 implies that $\|g\| \leq \bar{\gamma}$ for any $x \in \mathcal{L}$, where $\bar{\gamma} = 2\bar{c}L + \|g_0\|$.

Firstly, we give some estimation on $\theta_k$. We know by mean value theorem that

$$\theta_k = 2(f_k - f_{k+1}) + (g_k + g_{k+1})^T s_k$$

$$= -2\nabla f(\eta_k)^T s_k + (g_k + g_{k+1})^T s_k$$

$$= -[\nabla f(x_k) - \nabla f(\eta_k) + \nabla f(x_{k+1}) - \nabla f(\eta_k)]^T s_k,$$

where $\eta_k = x_k + \tau(x_{k+1} - x_k)$ and $\tau \in (0, 1)$. Hence

$$|\theta_k| \leq (\|\nabla f(x_k) - \nabla f(\eta_k)\| + \|\nabla f(x_{k+1}) - \nabla f(\eta_k)\|)\|s_k\|$$

$$\leq L(\|x_k - \eta_k\| + \|x_{k+1} - \eta_k\|)\|s_k\|$$

$$= L\|s_k\|^2 \leq 4\bar{c}^2.$$

On the other hand, since $\theta_k$ is appeared in the denominator, too small value must be avoided for the numerical stability, we ask $|\theta_k|$ to satisfy $0 < \eta \leq |\theta_k|$ as shown in Algorithm 2.1. Otherwise, $\beta_k^{DL+}$ will be used.

Let $f$ be a uniformly convex function, then there exists some constant $\mu > 0$ such that

$$\left(\nabla f(x) - \nabla f(y)\right)^T (x - y) \geq \mu\|y - x\|^2,$$

which implies

$$\mu\|s_k\|^2 \leq s_k^T y_k \leq L\|s_k\|^2. \tag{3.1}$$

Then we have that

$$\theta_k = 2\left(f_k - f_{k+1}\right) + \left(g_k + g_{k+1}\right)^T s_k$$

$$\geq \left(-g_{k+1}^T s_k + \frac{\mu}{2}\|s_k\|^2\right) + \left(g_k + g_{k+1}\right)^T s_k$$

$$= -s_k^T y_k + \mu\|s_k\|^2$$

$$\geq -\left(1 - \frac{\mu}{L}\right)s_k^T y_k$$

and

$$|\theta_k| \leq L\|s_k\|^2 \leq \frac{L}{\mu}s_k^T y_k.$$

Thus $\theta_k$ locates in the interval $\left(-\left(1 - \mu/L\right)s_k^T y_k, -\eta\right) \cup \left(\eta, L/\mu s_k^T y_k\right)$. Therefore, we assume that the following relationship always holds.

$$\frac{\mu}{L} \leq \left|\frac{s_k^T y_k}{\theta_k}\right| \leq \frac{1}{\bar{c}}.$$

The following theorem states the global convergence property of new method for uniformly convex functions.

**Theorem 3.3.** Let $f$ be a uniformly convex function and Assumptions 3.1 and 3.2 hold. Suppose that $\{x_k\}$ is the sequence generated by Algorithm 2.1 with $\beta_k$ in (2.4), then

$$\lim_{k \rightarrow \infty} \|g_k\| = 0. \tag{3.2}$$
Proof. It follows from \( f \) is uniformly convex function that
\[
d_k^T y_k \geq \mu \alpha_{k-1} \|d_{k-1}\|^2.
\]

By using Triangular and Cauchy-Schwartz inequalities, we have
\[
\|d_k\| \leq \|g_k\| + |\beta_k|^\text{new} \|d_{k-1}\|
\leq \|g_k\| + \frac{(L + (1-t)/\epsilon) \|g_k\| \|s_{k-1}\|}{\mu \alpha_{k-1} \|d_{k-1}\|^2} \|d_{k-1}\|
\leq \mu^{-1}(L + (1-t)/\epsilon + \mu) \|g_k\|,
\]
which means
\[
\sum_{k \geq 1} \|d_k\|^2 = +\infty.
\]

Therefore, from Lemma 3.2 [6] and the fact \( f \) is a uniformly convex function, we have
\[
\lim_{k \to \infty} \|g_k\| = 0.
\]

Q.E.D.

For the general function, we only need to show the modified Dai-Liao method with \( \beta_k \) in (2.4) satisfies the Property(*) depicted by Gilbert and Nocedal [10]. The rest analysis is similar to the original Dai-Liao’s method.

Definition 3.4. Consider a method of the form (1.2)–(1.3), and suppose that
\[
0 < \gamma \leq \|g_k\| \leq \bar{\gamma} \tag{3.3}
\]
for all \( k \geq 1 \). We say that the conjugate gradient method has the Property(*), if for all \( k \), there exist constants \( b > 1, k > 0 \) such that for all \( k \),
\[
|\beta_k| \leq b \text{ and } \|s_{k-1}\| \leq \lambda \Rightarrow |\beta_k| \leq \frac{1}{2b}. \tag{3.4}
\]

By the strong Wolfe condition (1.5), (1.10) and (3.3), we have
\[
d_k^T y_{k-1} \geq (\sigma - 1)g_{k-1}^T d_{k-1} \geq (1 - \sigma)c\gamma^2.
\]

Using this and boundedness of \( \|s_k\| \), we obtain
\[
|\beta_k| \leq \frac{(L + (1-t)/\epsilon) \|g_k\| \|s_{k-1}\|}{(1 - \sigma)c\gamma^2} \leq \frac{2(L + (1-t)/\epsilon)\bar{\gamma}c}{(1 - \sigma)c\gamma^2} =: b.
\]

Note that \( b \) can be defined such that \( b > 1 \). If we set
\[
\lambda := \frac{(1 - \sigma)c\gamma^2}{b(L + (1-t)/\epsilon)\bar{\gamma}}
\]
and \( s_{k-1} \leq \lambda \), then
\[
|\beta_k| \leq \frac{(L + (1-t)/\epsilon)\bar{\gamma}\lambda}{(1 - \sigma)c\gamma^2} = \frac{1}{b}.
\]

Therefore, the NEW+ method has Property (*). Thus, we have the following convergence theorem.

Theorem 3.5. Let Assumptions 3.1 and 3.2 hold. If the sequence \( \{x_k\} \) is generated by the NEW+ method with the strong Wolfe line search for \( \sigma \in (0, 1) \), where \( d_k \) satisfies condition (1.10) with \( c > 0 \). Then we have
\[
\lim_{k \to +\infty} \|g_k\| = 0.
\]
Table 1. Tested conjugate gradient algorithms.

<table>
<thead>
<tr>
<th>( \beta_k )</th>
<th>Name of method</th>
<th>Abbreviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \beta_{FR}^k )</td>
<td>The Fletcher-Reeves method [8]</td>
<td>FR</td>
</tr>
<tr>
<td>( \beta_{PRP+}^k )</td>
<td>The Polak-Ribiere-Polyak [18, 19]</td>
<td>PRP+</td>
</tr>
<tr>
<td>( \beta_{HS}^k )</td>
<td>The Hestenes-Stiefel method [14]</td>
<td>HS</td>
</tr>
<tr>
<td>( \beta_{DY}^k )</td>
<td>The Dai-Yuan method [3]</td>
<td>DY</td>
</tr>
<tr>
<td>( \beta_{HZ}^k )</td>
<td>The Hager and Zhang’s method [12]</td>
<td>HZ</td>
</tr>
<tr>
<td>( \beta_{DL+}^k )</td>
<td>The Dai-Liao method [6]</td>
<td>DL+</td>
</tr>
<tr>
<td>( \beta_{YT+}^k )</td>
<td>Yabe-Takano’s method [20]</td>
<td>YT+</td>
</tr>
<tr>
<td>( \beta_{new+}^k )</td>
<td>Our method</td>
<td>NEW+</td>
</tr>
</tbody>
</table>

Table 2. Numerical results for Trigonometric with \( n = 5000 \).

<table>
<thead>
<tr>
<th>( \beta_k )</th>
<th>Name of method</th>
<th>Abbreviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>0.3</td>
<td>0.5</td>
</tr>
</tbody>
</table>

| \( \rho_k \) | t |
| 0 | 21/100 | 49/136 | 20/92 | 64/156 | 69/158 |
| 0.1 | 21/100 | 82/177 | 20/92 | 61/152 | 63/151 |
| 0.3 | 20/99 | 67/160 | 20/93 | **21/95** | 63/153 |
| 0.5 | 20/94 | **19/95** | 42/134 | 41/127 | 38/127 |
| 0.7 | 22/107 | 21/93 | 67/154 | 69/162 | 23/100 |
| 0.9 | 22/108 | 45/135 | 20/91 | 64/152 | 64/152 |
| 1.0 | 20/95 | 68/166 | 21/99 | 38/127 | 33/1 |
| **20/97** | 47/140 | 20/98 | 67/158 | **19/94** |
| **20/97** | **20/90** | 21/95 | 40/132 | **22/95** |
| **20/97** | 44/128 | **19/89** | 65/153 | **19/95** |

4. Numerical Experiments

In this section, some numerical results are reported on a set of 76 unconstrained optimization problems selected from [1] and CUTEst library [11]. We tested the conjugate gradient algorithms with the conjugacy parameters given in Table 1.

For the algorithms DL+, YT+ and our new method, different scaled parameters \( \rho \) and \( t \) are used. In the case where an ascent direction is generated, we restart the algorithm by setting \( d_k = -g_k \).

All codes were written in Fortran and in double precision arithmetic. (Note that, for the sake of fairness, at the beginning of experiments we do not directly run Hager and Zhang’s CG\_descent codes for the test problems, we just use their conjugate parameter under the same linear search in our test framework). The stopping rule is set as \( \|g_k\|_\infty \leq 10^{-6} \). The iteration is also terminated if the total number of iterations exceeds 10,000. Partial numerical results are summarized in Tables 2–6 and given in the form of (number of iterations/number of function-gradient evaluations), the detailed complete numerical results can be downloaded from the website [https://github.com/piratetwo/ml](https://github.com/piratetwo/ml).
Table 3. Numerical results for ENGVAL1 with $n = 5000$.

<table>
<thead>
<tr>
<th>FR</th>
<th>PRP+</th>
<th>HS</th>
<th>DY</th>
<th>HZ</th>
<th>CG_{descent}</th>
</tr>
</thead>
<tbody>
<tr>
<td>285/8484</td>
<td>195/5333</td>
<td>695/20246</td>
<td>1711/53963</td>
<td>229/6770</td>
<td>24/78</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$\rho_k$</th>
<th>0.1</th>
<th>0.3</th>
<th>0.5</th>
<th>0.7</th>
<th>0.9</th>
</tr>
</thead>
<tbody>
<tr>
<td>DL+</td>
<td>0</td>
<td>117/3068</td>
<td>33/364</td>
<td>24/72</td>
<td>105/2642</td>
</tr>
<tr>
<td>YT+</td>
<td>0.1</td>
<td>101/2570</td>
<td>224/6557</td>
<td>145/3867</td>
<td>105/2774</td>
</tr>
<tr>
<td>NEW+</td>
<td>166/4696</td>
<td>185/5189</td>
<td>184/5306</td>
<td>112/2886</td>
<td>181/5142</td>
</tr>
<tr>
<td>YT+</td>
<td>0.3</td>
<td>228/6821</td>
<td>200/6005</td>
<td>307/9366</td>
<td>137/3683</td>
</tr>
<tr>
<td>NEW+</td>
<td>78/1778</td>
<td>170/4931</td>
<td>157/4448</td>
<td>140/3857</td>
<td>136/3666</td>
</tr>
<tr>
<td>YT+</td>
<td>0.5</td>
<td>58/1225</td>
<td>79/1805</td>
<td>81/1896</td>
<td>217/6311</td>
</tr>
<tr>
<td>NEW+</td>
<td>46/805</td>
<td>164/4715</td>
<td>162/4693</td>
<td>162/4693</td>
<td>128/3529</td>
</tr>
<tr>
<td>YT+</td>
<td>0.7</td>
<td>200/6064</td>
<td>127/3524</td>
<td>60/1233</td>
<td>149/4241</td>
</tr>
<tr>
<td>NEW+</td>
<td>37/431</td>
<td>177/4856</td>
<td>72/1552</td>
<td>234/6807</td>
<td>116/2866</td>
</tr>
</tbody>
</table>

Table 4. Numerical results for Raydan 1 with $n = 5000$.

<table>
<thead>
<tr>
<th>FR</th>
<th>PRP+</th>
<th>HS</th>
<th>DY</th>
<th>HZ</th>
<th>CG_{descent}</th>
</tr>
</thead>
<tbody>
<tr>
<td>–/–</td>
<td>–/–</td>
<td>756/1019</td>
<td>739/819</td>
<td>856/1334</td>
<td>490/1472</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$\rho_k$</th>
<th>0.1</th>
<th>0.3</th>
<th>0.5</th>
<th>0.7</th>
<th>0.9</th>
</tr>
</thead>
<tbody>
<tr>
<td>DL+</td>
<td>816/1077</td>
<td>782/1082</td>
<td>724/972</td>
<td>766/1046</td>
<td>812/1101</td>
</tr>
<tr>
<td>YT+</td>
<td>834/1274</td>
<td>770/1041</td>
<td>766/1077</td>
<td>540/711</td>
<td>767/1001</td>
</tr>
<tr>
<td>NEW+</td>
<td>850/1166</td>
<td>689/933</td>
<td>808/1092</td>
<td>649/853</td>
<td>785/1037</td>
</tr>
<tr>
<td>YT+</td>
<td>809/1109</td>
<td>749/1038</td>
<td>782/1063</td>
<td>810/1122</td>
<td>863/1140</td>
</tr>
<tr>
<td>NEW+</td>
<td>839/1142</td>
<td>628/858</td>
<td>748/1010</td>
<td>791/1063</td>
<td>864/1154</td>
</tr>
<tr>
<td>YT+</td>
<td>752/1032</td>
<td>744/1015</td>
<td>749/1028</td>
<td>724/964</td>
<td>829/1135</td>
</tr>
<tr>
<td>NEW+</td>
<td>764/1015</td>
<td>732/1011</td>
<td>779/1058</td>
<td>710/964</td>
<td>801/1090</td>
</tr>
<tr>
<td>YT+</td>
<td>622/853</td>
<td>679/915</td>
<td>826/1144</td>
<td>842/1114</td>
<td>789/1065</td>
</tr>
<tr>
<td>NEW+</td>
<td>719/970</td>
<td>736/992</td>
<td>704/953</td>
<td>696/940</td>
<td>820/1081</td>
</tr>
<tr>
<td>YT+</td>
<td>712/974</td>
<td>824/1094</td>
<td>882/1224</td>
<td>777/1045</td>
<td>890/1187</td>
</tr>
<tr>
<td>NEW+</td>
<td>671/891</td>
<td>870/115</td>
<td>786/1062</td>
<td>851/1179</td>
<td>746/991</td>
</tr>
<tr>
<td>YT+</td>
<td>708/957</td>
<td>859/1183</td>
<td>669/897</td>
<td>876/1176</td>
<td>815/1103</td>
</tr>
<tr>
<td>NEW+</td>
<td>722/969</td>
<td>715/972</td>
<td>759/1023</td>
<td>706/948</td>
<td>680/905</td>
</tr>
</tbody>
</table>

In the Tables 2–6, the boldface font is used to mark the first and second efficient method which performs better than the other two algorithms for each $\rho_k$ and $t$. The number of the best performance for Algorithm NEW+, YT+, DL+ are 32, 15 and 3, respectively.

In most cases, our new method improves Yabe-Takano’s method. For a special $\rho_k = 3$, which was used in the modified quasi-Newton method, we compare the numerical performance of YT+ and NEW+. We run the codes with different $t = 0.1, 0.2, \ldots, 1$ and compute the medians for each problem. The performance profiles introduced by Dolan and Moré [7] are used to display the behaviours of these two methods. Figure 1 shows that the NEW+ method performs the best result regarding the number of iterations and function-gradient evaluations, which is located at the top curve in Figure 1.
Table 5. Numerical results for SINQUAD with $n = 5000$.

<table>
<thead>
<tr>
<th>FR</th>
<th>PRP+</th>
<th>HS</th>
<th>DY</th>
<th>HZ</th>
<th>CG_descent</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1455/2665</td>
<td>–/-</td>
<td>1891/2579</td>
<td>2021/4044</td>
<td>1071/3813</td>
</tr>
<tr>
<td>$\rho_k$</td>
<td>0.1</td>
<td>0.3</td>
<td>0.5</td>
<td>0.7</td>
<td>0.9</td>
</tr>
<tr>
<td>DL+</td>
<td>481/1014</td>
<td>681/1424</td>
<td>728/1532</td>
<td><strong>475/1018</strong></td>
<td>588/1238</td>
</tr>
<tr>
<td>Y+</td>
<td>602/1270</td>
<td><strong>499/1043</strong></td>
<td>792/1676</td>
<td>703/1515</td>
<td><strong>390/874</strong></td>
</tr>
<tr>
<td>NEW+</td>
<td>550/1172</td>
<td>530/1145</td>
<td><strong>465/1021</strong></td>
<td><strong>451/965</strong></td>
<td>606/1282</td>
</tr>
<tr>
<td>Y+</td>
<td>533/1121</td>
<td>568/1225</td>
<td>1431/2951</td>
<td>477/1013</td>
<td>526/1110</td>
</tr>
<tr>
<td>NEW+</td>
<td><strong>482/1009</strong></td>
<td>721/1507</td>
<td>638/1325</td>
<td>599/1277</td>
<td>597/1249</td>
</tr>
<tr>
<td>Y+</td>
<td>528/1095</td>
<td>579/1272</td>
<td>544/1210</td>
<td>687/1474</td>
<td>553/1185</td>
</tr>
<tr>
<td>NEW+</td>
<td>532/1132</td>
<td><strong>491/1057</strong></td>
<td>804/1682</td>
<td>503/1059</td>
<td>500/1094</td>
</tr>
<tr>
<td>Y+</td>
<td>588/1295</td>
<td>842/1773</td>
<td><strong>496/1048</strong></td>
<td>555/1197</td>
<td>499/1102</td>
</tr>
<tr>
<td>NEW+</td>
<td>533/1105</td>
<td>543/1195</td>
<td>565/1189</td>
<td>452/1158</td>
<td><strong>476/1012</strong></td>
</tr>
<tr>
<td>Y+</td>
<td>546/1212</td>
<td>522/1088</td>
<td>610/1275</td>
<td>598/1287</td>
<td>568/1235</td>
</tr>
<tr>
<td>NEW+</td>
<td>483/1029</td>
<td>594/1280</td>
<td>502/1068</td>
<td>489/1046</td>
<td>644/1358</td>
</tr>
<tr>
<td>Y+</td>
<td><strong>423/873</strong></td>
<td>590/1272</td>
<td>538/1162</td>
<td>498/1048</td>
<td>483/1011</td>
</tr>
<tr>
<td>NEW+</td>
<td>593/1275</td>
<td>526/1119</td>
<td>567/1182</td>
<td>569/1204</td>
<td>506/1064</td>
</tr>
</tbody>
</table>

Table 6. Numerical results for Woods with $n = 5000$.

<table>
<thead>
<tr>
<th>FR</th>
<th>PRP+</th>
<th>HS</th>
<th>DY</th>
<th>HZ</th>
<th>CG_descent</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>112/218</td>
<td>65/133</td>
<td>87/214</td>
<td>95/178</td>
<td>161/309</td>
</tr>
<tr>
<td>$\rho_k$</td>
<td>0.1</td>
<td>0.3</td>
<td>0.5</td>
<td>0.7</td>
<td>0.9</td>
</tr>
<tr>
<td>DL+</td>
<td>39/80</td>
<td>52/99</td>
<td>43/85</td>
<td>45/90</td>
<td>48/94</td>
</tr>
<tr>
<td>Y+</td>
<td><strong>32/68</strong></td>
<td>37/76</td>
<td>63/122</td>
<td>59/113</td>
<td>46/90</td>
</tr>
<tr>
<td>NEW+</td>
<td>38/77</td>
<td><strong>37/75</strong></td>
<td>70/123</td>
<td>43/90</td>
<td>44/89</td>
</tr>
<tr>
<td>Y+</td>
<td>48/93</td>
<td>40/101</td>
<td>53/103</td>
<td>58/116</td>
<td>61/114</td>
</tr>
<tr>
<td>NEW+</td>
<td>39/83</td>
<td>44/86</td>
<td>48/97</td>
<td>46/96</td>
<td><strong>36/74</strong></td>
</tr>
<tr>
<td>Y+</td>
<td>32/69</td>
<td>42/83</td>
<td>41/80</td>
<td>81/161</td>
<td>58/116</td>
</tr>
<tr>
<td>NEW+</td>
<td>62/113</td>
<td><strong>34/72</strong></td>
<td>50/100</td>
<td>36/78</td>
<td>53/103</td>
</tr>
<tr>
<td>Y+</td>
<td>35/74</td>
<td>49/99</td>
<td>40/81</td>
<td>58/114</td>
<td>45/91</td>
</tr>
<tr>
<td>NEW+</td>
<td>35/76</td>
<td>51/103</td>
<td><strong>36/70</strong></td>
<td>69/132</td>
<td>58/109</td>
</tr>
<tr>
<td>Y+</td>
<td><strong>31/66</strong></td>
<td>44/86</td>
<td>47/93</td>
<td>38/79</td>
<td><strong>39/79</strong></td>
</tr>
<tr>
<td>NEW+</td>
<td>40/83</td>
<td>49/97</td>
<td>57/108</td>
<td><strong>36/74</strong></td>
<td>59/99</td>
</tr>
<tr>
<td>Y+</td>
<td>41/83</td>
<td>41/83</td>
<td>36/74</td>
<td>39/79</td>
<td>59/118</td>
</tr>
<tr>
<td>NEW+</td>
<td>48/95</td>
<td>60/113</td>
<td><strong>32/67</strong></td>
<td>46/91</td>
<td>41/84</td>
</tr>
</tbody>
</table>

Finally, we run Hager and Zhang’s method with the approximate Wolfe line search conditions (Hager and Zhang’s CG\_descent Fortran code Version 1.4\textsuperscript{2}). From Figure 2, for about 62% of all problems, CG\_descent needs the least iterations, it has the best performance. However, CG\_descent has the poorer performance than the YT+ and NEW+ regarding the number of function-gradient evaluations which mainly affects the efficiency of the methods.

\textsuperscript{2}https://people.clas.ufl.edu/hager/software/
Figure 1. Performance profiles based on iterations and function-gradient evaluations for YT+ and NEW+.

Figure 2. Performance profiles based on iterations and function-gradient evaluations for YT+, NEW+, and CG\_descent.

5. Conclusions

In this paper, based on the Dai and Liao’s conjugacy condition and the modified secant condition proposed by Zhang and Xu [21], we derived a new family of Dai-Liao-type conjugate gradient methods. Under some certain assumptions, we show that our methods are globally convergent for general functions. Numerical results show that our new methods can outperform some existing ones.

Acknowledgements. This work was supported by the Natural Science Foundation of Henan Province (Grant No. 202300410236) and the Training Program Foundation for Youth of Henan Normal University (Grant No. 2020PL28).

References


---

**Subscribe to Open (S2O)**

A fair and sustainable open access model

This journal is currently published in open access under a Subscribe-to-Open model (S2O). S2O is a transformative model that aims to move subscription journals to open access. Open access is the free, immediate, online availability of research articles combined with the rights to use these articles fully in the digital environment. We are thankful to our subscribers and sponsors for making it possible to publish this journal in open access, free of charge for authors.

Please help to maintain this journal in open access!

Check that your library subscribes to the journal, or make a personal donation to the S2O programme, by contacting subscribers@edpsciences.org

More information, including a list of sponsors and a financial transparency report, available at: https://www.edpsciences.org/en/maths-s2o-programme