Pacific Journal of Optimization
Vol. 15, No. 3, pp. 359 – 377, 2019

A NOTE ON THE OPTIMAL PARAMETER OF BABAIE-KAFAKI'S THREE-TERM CONJUGATE GRADIENT METHOD*

Xiaoliang Dong† and Deren Han

Abstract: Minimizing the condition number is often used in conjugate gradient methods to improve computational efficiency. In this paper, based on an eigenvalue study and a singular value study, respectively, we discuss the condition number of the conjugate gradient method proposed by Babaie–Kafaki. The obtained results improve the method, since the condition number of the corresponding iteration matrix attains its minimum value. Moreover, we propose a modified Hestenes–Stiefel type three–term conjugate gradient method with adaptive strategy, in which the nice properties of the sufficient descent condition and adaptive conjugacy condition can be retained, accelerating the convergence or reducing the condition number of iteration matrix. Under mild conditions, we show that the proposed method converges globally for general objective functions. Numerical experiments indicate that the method is practically promising.

Key words: *three-term conjugate gradient method, sufficient descent condition, conjugacy condition, condition number*

Mathematics Subject Classification: *65K05; 90C53*

---

1 Introduction

Consider the following unconstrained optimization problem:

$$\min_{x\in\mathbb{R}^n} f(x), \tag{1.1}$$

where $f:\mathbb{R}^n\to\mathbb{R}$ is a smooth function. We denote by $g(x)$ the gradient of f at x and abbreviate $g(x_k)$ and $f(x_k)$ by g_k and f_k, respectively. Also, we use $y_{k-1}=g_k-g_{k-1}$ and $\|\cdot\|$ to stand for the Euclidian norm.

Conjugate gradient (CG) methods are a class of efficient tools for solving large–scale unconstrained optimization problems due to their low memory requirement and simple iterative formula. The recursion scheme of CG methods is

$$x_0\in\mathbb{R}^n,\ x_{k+1}=x_k+s_k, s_k=\alpha_k d_k,\ \forall k\geq 0, \tag{1.2}$$

*This work is supported by the National Natural Science Foundation of China (11601012, 11625105,11431002), the Natural Science Basic Research Plan in Shaanxi Province of China (2017JM1046), the Innovation Talent Promotion Plan of Shaanxi Province for Young Sci–Tech New Star (2017KJXX60).

†The work of the first author was done while he was visiting Xi'an Shiyou University

where $\alpha_k > 0$ is a steplength to be computed along the search direction d_k defined by

$$d_1 = -g_1, d_k = -g_k + \beta_k d_{k-1}, k = 2, 3, \cdots \tag{1.3}$$

and β_k is a parameter. Several famous formulas for β_k are called the Fletcher–Reeves [24], Hestenes–Stiefel [29], Polak–Ribière–Polyak [25] and Dai–Yuan [12] formulas, and the corresponding parameters β_k are listed as follows

$$\beta_k^{FR} = \frac{\|g_k\|^2}{\|g_{k-1}\|^2},\ \beta_k^{HS} = \frac{g_k^T y_{k-1}}{d_{k-1}^T y_{k-1}},\ \beta_k^{PRP} = \frac{g_k^T y_{k-1}}{\|g_{k-1}\|^2},\ \beta_k^{DY} = \frac{\|g_k\|^2}{d_{k-1}^T y_{k-1}}. \tag{1.4}$$

We refer to an excellent survey [26] for further details.

Among these formulae, the Hestenes–Stiefel (HS) method has attracted much attention, not only for its important history significance, but also for its excellent computational performance. Numerous modified HS algorithms and variants of the parameter β_k^{HS} have been developed, analyzed and implemented over the past few years. Generally, the conjugacy condition, sufficient descent condition and minimizing condition number are significant factors that play important roles for the efficiency.

The search directions of the HS method satisfy the standard conjugacy condition, independent of the line search used. That is,

$$d_k^T y_{k-1} = 0. \tag{1.5}$$

The modified HS conjugate direction

$$d_k = -\left(I - \frac{s_{k-1} y_{k-1}^T}{s_{k-1}^T y_{k-1}} + \frac{s_{k-1} s_{k-1}^T}{s_{k-1}^T y_{k-1}}\right) g_k = -M_k g_k, \tag{1.6}$$

proposed by Perry [32] satisfied $y_{k-1}^T M_k = s_{k-1}^T$, which is similar, but not identical to the conjugacy condition (1.5). Later, as an extension of (1.5), Dai and Liao [13] proposed the following Dai–Liao (DL) conjugacy condition:

$$d_k^T y_{k-1} = -\xi g_k^T s_{k-1}, \tag{1.7}$$

with a positive constant ξ.

If there exists a constant $c > 0$ such that

$$d_k^T g_k \leq -c\|g_k\|^2,\ \forall k \in \mathbb{N}, \tag{1.8}$$

then the so-called sufficient descent condition holds. Gilbert and Nocedal [25] illustrates its important usefulness for obtaining convergence of the PRP+ CG method. Classical examples included the CG_DESCENT method [27], the CGOPT method [14], the THREECG method [4], the 3HS+ method [31] and the CTTHS and MTTHS methods [38], in which the condition (1.8) holds independent of any line searches used and the convexity of the objective functions. Further development and applications can refer to [1, 2, 5, 6, 8–11, 15, 17–23, 30, 31, 33, 35–37]. Note that although the DL method seldom generates uphill search directions in actual computations, it generally fails to guarantee the sufficient descent condition [3]. In applications, ill–condition usually causes the loss of the orthogonality, which further causes the loss of efficiency. Hence, minimizing the condition number and correcting the orthogonality property are important in implementing CG methods. By combining the orthogonality correction scheme, Hager and Zhang developed an excellent software package L_CG_DESCENT, which is very efficient for solving large–scale test problems [28].

Recently, three–term methods attract more and more attentions. Babaie–Kafaki [7] proposed an extended three–term CG method, in which the search direction d_k was constructed as follows:

$$d_k^{EZZL} = -g_k + \beta_k^{HS} s_{k-1} - \theta_k y_{k-1}, \tag{1.9}$$

where the parameter θ_k is given by

$$\theta_k = \omega_k \frac{g_k^T s_{k-1}}{s_{k-1}^T y_{k-1}}, \tag{1.10}$$

and $\omega_k \in [0,1]$ is a scalar. When $\omega_k = 1$, the above method reduced to the method proposed by Zhang, Zhou and Li [38], and when $\omega_k = 0$, the method reduced to the HS method. Throughout the paper, we use the same notations as in [7] and [38] and call them EZZL and ZZL method for short, respectively.

The convergence analysis of the EZZL method was made if the objective functions were uniformly convex. However, it is unknown that whether this method can be globalized with the Wolfe conditions for the nonconvex functions. On the other hand, note that the search direction (1.9) can be rewritten as $d_k = -Q_k g_k$ with the iteration matrix Q_k

$$Q_k = I - \frac{s_{k-1} y_{k-1}^T}{s_{k-1}^T y_{k-1}} + \omega_k \frac{y_{k-1} s_{k-1}^T}{y_{k-1}^T s_{k-1}}, \tag{1.11}$$

which is neither symmetric nor normal unless $\omega_k = -1$. To study the sufficient descent condition, Babaie–Kafaki introduced the symmetric matrix

$$A_k = \frac{Q_k + Q_k^T}{2} = I + \frac{1}{2}(\omega_k - 1)\frac{s_{k-1} y_{k-1}^T + y_{k-1} s_{k-1}^T}{s_{k-1}^T y_{k-1}}, \tag{1.12}$$

and analyzed its smallest and largest eigenvalues:

$$\lambda_k^{\pm}(\omega_k) = \frac{1}{2}\left\{(1+\omega_k) \pm (1-\omega_k)\frac{\|s_{k-1}\|\|y_{k-1}\|}{s_{k-1}^T y_{k-1}}\right\}. \tag{1.13}$$

From (1.13) we can see that in order to guarantee the positive definiteness of the matrix A_k, we should ensure that $\lambda_k^- = \xi \in (0,1]$. Consequently, the following optimal value of the parameter ω_k is obtained

$$\omega_k = \frac{(2\xi - 1) y_{k-1}^T s_{k-1} + \|s_{k-1}\|\|y_{k-1}\|}{y_{k-1}^T s_{k-1} + \|s_{k-1}\|\|y_{k-1}\|}. \tag{1.14}$$

Though the choice (1.14) can ensure the sufficient descent property, the other two key factors, i.e., conjugacy condition and minimizing condition number, are ignored. In this paper, to keep the intrinsic clustering of the eigenvalues of iteration matrix and self–adjusting conjugacy property, we conduct an eigenvalue study and a singular value study, which help us present a revised version of the parameter. We also prove that the EZZL method is actually the ZZL method where the optimal choice for the parameter λ_k can minimize the condition number of the iteration matrix. The analysis motivates us to consider minimizing the condition number of the iteration matrix in algorithm design.

The paper is organized as follows. In Sect.2, we present the optimal parameter choice from an eigenvalue study and a singular value study, respectively. In Sect. 3, we propose a modified HS type three–term CG method and establish its global convergence with the Wolfe conditions for the nonconvex functions. We report numerical results in Sect. 4. Final conclusions are made in the Sect. 5. Numerical results are listed in the part of Appendix.

$\boxed{2}$ An Optimal Parameter Choice

As mentioned above, the matrix condition number, both due to its role in the error analysis and its role in convergence rate, plays an important role in a numerical algorithm. Hence, for the three–term HS–type method (1.9), we consider finding an optimal choice for the parameter ω_k and minimizing the condition number of iteration matrix (1.11), based on an eigenvalue study and a singular value study, respectively.

- **An eigenvalue study:** We first consider the condition number based on an eigenvalue study. To be specific, we substitute (1.14) in (1.13), and obtain that

$$\lambda_k^+(\omega_k) = \frac{\xi y_{k-1}^T s_{k-1} + (2-\xi)\|s_{k-1}\|\|y_{k-1}\|}{y_{k-1}^T s_{k-1} + \|s_{k-1}\|\|y_{k-1}\|}, \tag{2.1}$$

and the corresponding condition number is computed as

$$Cond_2(A_k) = \frac{\lambda_k^+(\omega_k)}{\lambda_k^-(\omega_k)} = \frac{y_{k-1}^T s_{k-1} + (\frac{2}{\xi}-1)\|s_{k-1}\|\|y_{k-1}\|}{y_{k-1}^T s_{k-1} + \|s_{k-1}\|\|y_{k-1}\|}. \tag{2.2}$$

The choice for $\xi = 1$, which corresponds to $\omega_k = 1$, can minimize the condition number of the iteration matrix A_k.

- **A singular value study:** We now briefly present the definition of the singular value decomposition and spectral condition number as follows.

Definition 2.1 ([34])**.** Let $A \in \mathbb{R}^{n\times m}$ be a nonzero matrix with rank r. Then, $\mathbb{R}^m$ has an orthogonal basis $q_i, i = 1, 2, ..., m$, $\mathbb{R}^n$ has an orthogonal basis $p_i, i = 1, 2, ..., n$, and there exist $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_r > 0$ such that

$$Aq_i = \begin{cases} \sigma_i p_i, & \text{i=1,2,...,r,} \\ 0, & \text{i=r+1,...,m,} \end{cases} \tag{2.3}$$

and

$$A^T p_i = \begin{cases} \sigma_i q_i, & \text{i=1,2,...,r,} \\ 0, & \text{i=r+1,...,n.} \end{cases} \tag{2.4}$$

Furthermore, for a nonsingular matrix $A \in \mathbb{R}^{n\times n}$, its spectral condition number is defined by $\kappa_2(A) = \|A\|\|A^{-1}\|$.

Since $s_k^T y_k = \alpha_k d_k^T y_k > 0$, as guaranteed by the Wolfe line search, we obviously obtain that the vectors s_k and y_k are nonzero vectors. Therefore, there exist a set of mutually unit orthogonal vectors $\{u_k^i\}_{i=1}^{n-2}$ such that

$$s_k^T u_k^i = y_k^T u_k^i = 0, i = 1, 2, ..., n-2, \tag{2.5}$$

which yield

$$Q_k u_k^i = u_k^i, i = 1, 2, ..., n-2. \tag{2.6}$$

It follows that Q_k has the eigenvalue 1 of multiplicity $n-2$, which corresponds to $\{u_k^i\}_{i=1}^{n-2}$. Naturally, we intend to find the two remanning eigenvalues, i.e, $\sigma_k^+(\omega)$ and $\sigma_k^-(\omega)$.

We can easily get the trace of $Q_k^T Q_k$ as follows:

$$\begin{aligned} Tr(Q_k^T Q_k) &= n-2+(1+\omega_k^2)\cdot\frac{\|s_{k-1}\|^2\|y_{k-1}\|^2}{(s_{k-1}^T y_{k-1})^2}, \\ &= \underbrace{1+...+1}_{n-2\quad times}+(\sigma_k^+(\omega_k))^2+(\sigma_k^-(\omega_k))^2, \end{aligned} \tag{2.7}$$

which implies that

$$(\sigma_k^+(\omega))^2+(\sigma_k^-(\omega))^2=(1+\omega_k^2)\Theta_k, \tag{2.8}$$

where

$$\Theta_k=\left(\frac{\|s_{k-1}\|\|y_{k-1}\|}{s_{k-1}^T y_{k-1}}\right)^2\geq 1. \tag{2.9}$$

Notice that Q_k presents rank–two update, we can readily compute its determinant. Specifically, from the relationship that the determinant of the iteration matrix Q_k equals to the product of $\sigma_k^+(\omega)$ and $\sigma_k^-(\omega)$, we get that

$$det(Q_k)=\sigma_k^+(\omega_k)\sigma_k^-(\omega_k)=\omega_k\Theta_k. \tag{2.10}$$

Based on the relationships between the trace and the determinant of a matrix and its eigenvalues, we can construct a quadratic equation as follows:

$$\sigma^2-\sqrt{(1+\omega_k)^2\Theta_k}\sigma+(\omega_k\Theta_k)=0. \tag{2.11}$$

It should be pointed out that the discriminant in quadratic equation (2.11) $\Delta_\sigma=(1-\omega_k)^2\Theta_k\geq 0$ and therefore the existence of two real roots can be ensured.

A simple analysis of Equation (2.11) indicates that

$$\sigma_k^\pm(\omega_k)=\frac{\sqrt{\Theta_k}}{2}\left\{|1+\omega_k|\pm|1-\omega_k|\right\}. \tag{2.12}$$

Subsequently, we list the singular values of Q_k by considering the following two cases.

Case (i) For $0<\omega_k\leq 1$, we can easily obtain that

$$\underbrace{1,1,...,1}_{n-2\quad times},\sigma_k^-=\omega_k\sqrt{\Theta_k},\sigma_k^+=\sqrt{\Theta_k}. \tag{2.13}$$

Case (ii) For $\omega_k>1$, we can readily get that

$$\underbrace{1,1,...,1}_{n-2\quad times},\sigma_k^-=\sqrt{\Theta_k},\sigma_k^+=\omega_k\sqrt{\Theta_k}. \tag{2.14}$$

To minimize the spectral condition number $\kappa_2(Q_k)$, it is equivalent to clustering the singular values of the iteration matrix Q_k as dense as possible. We consider to employ $\omega_k=1$ to achieve this goal.

3 A New Three–Term CG Method

It follows from the last section that the availability of condition number analysis of iteration matrix is essential to practically implement CG method. To elaborate this further, in this section, we find a new hybridization form of search directions of the EZZL method to coordinate with three aforementioned key factors in a suitable way, thereby retaining the properties of HS+ method and ZZL method, and ensuring the global convergence for the general functions. Different from the existent methods, a dynamical adjustment strategy of the search direction is chosen to reduce the condition number of iteration matrix as much as possible or efficiently eliminate round–off error. Since the proposed search directions are obtained from condition number analysis, we call the presented method as CZZL method for short. In Subsection 3.1, a description of a CZZL algorithm is detailed. Convergence analysis of the CZZL method is investigated in Subsection 3.2.

3.1 Properties of the CZZL method

Theoretically, the HS+ method has the conjugacy condition (1.5) and self–restarting mechanism.

In such a case, we combine the obtained optimal parameter and the self–adjusting conjugacy condition [19], and present the following search direction:

$$d_k = \begin{cases} -g_k + \beta_k^{HS} d_{k-1}, & k \in K_1, \\ -g_k + \beta_k^{HS} d_{k-1} - \dfrac{g_k^T d_{k-1}}{d_{k-1}^T y_{k-1}} y_{k-1}, & k \in K_2, \end{cases} \tag{3.1}$$

where the index set K_1 and K_2 are presented by

$$K_1 = \{k \in \mathbb{N} | \;\; g_k^T y_{k-1} > 0, g_k^T d_{k-1} < 0\},$$

$$K_2 = \{k \in \mathbb{N} | \;\; g_k^T y_{k-1} > 0, g_k^T d_{k-1} \geq 0\}.$$

Subsequently, we state the steps of this method as follows.

Algorithm 3.1 (CZZL method)**. Step 1.** Give positive constant ε, and $\rho < \sigma < 1$. Choose an initial point $x_1 \in \mathbb{R}^n$ and set $d_1 = -g_1$ and $k = 1$.

Step 2. Determine a steplength α_k satisfying the Wolfe conditions:

$$f(x_k + \alpha_k d_k) - f(x_k) \leq \rho \alpha_k g_k^T d_k, \tag{3.2}$$

$$g\left(x_k + \alpha_k d_k\right)^T d_k \geq \sigma g_k^T d_k. \tag{3.3}$$

Step 3. Let the new iterate by $x_{k+1} = x_k + \alpha_k d_k$ and calculate g_{k+1}. **If** $\|g_{k+1}\| < \varepsilon$, **then stop**.

Step 4. If $g_{k+1}^T y_k \leq 0$, **then** set $d_{k+1} = -g_{k+1}$ and $k = k+1$, and **goto** Step 2.

Step 5. If $g_{k+1}^T y_k > 0$, **then** compute the search direction d_{k+1} by (3.1), and **goto** Step 2.

The following lemma indicates that the proposed search directions satisfy the sufficient descent condition and adaptive conjugacy condition, the proof is similar to that of the CHS method in [19], so we omit it here.

Lemma 3.2. *Suppose that the steplength α_k satisfies the Wolfe conditions. The search directions $\{d_k\}_{k\geq 0}$ of CZZL method satisfiy the sufficient descent condition* (1.8) *with $c = 1$, that is $d_k^T g_k \leq -\|g_k\|^2$. Furthermore, the adaptive conjugacy condition is also satisfied, that is,*

$$y_k^T d_{k+1} = \begin{cases} -\dfrac{\|y_k\|^2}{s_k^T y_k}\left(g_{k+1}^T s_k\right), & g_{k+1}^T d_k > 0, \\ 0, & g_{k+1}^T d_k \leq 0. \end{cases} \tag{3.4}$$

Remark 3.3. The relaxed form of conjugacy condition above leads to inherit some nice properties of the HS+ method, while maintaining the descent property and known convergence results.

3.2 Global convergence of the CZZL method

We begin this section by making the following regular assumptions, commonly used to establish the global convergence of the CG methods.

Assumption 3.4.

Boundedness Assumption: The level set $\Omega = \{x \in \mathbb{R}^n | f(x) \leq f(x_1)\}$ is bounded.

Lipschitz Assumption: In some neighborhood Ω_0 of Ω, the objective function f is continuously differentiable and its gradient g is Lipschitz continuous, namely, there exists a constant $L > 0$ such that $\|g(x) - g(y)\| \leq L\|x - y\|, \forall x, y \in \Omega_0$.

Note that these Assumptions imply that there exist constants $B > 0$ and $\gamma > 0$ such that $\|x - y\| \leq 2B$ *and* $\|g(x)\| \leq \gamma, \forall x, y \in \Omega$.

Now, we come to establish global convergence of Algorithm 3.1. For general nonlinear functions, similar to [27], we can obtain a weaker global convergence result in the sense that $\liminf_{k\to\infty} \|g_k\| = 0$. In what follows, for the sake of contradiction we assume that there exist a positive constant ε such that

$$\|g_k\| \geq \varepsilon, \ \forall k \in N. \tag{3.5}$$

Similar to analysis of [27], here we need to state some properties of d_k, β_k and s_k.

Lemma 3.5. *Suppose that Assumptions 3.4 hold. Let $\{x_k\}_{k\geq 0}$ be generated by Algorithm 3.1. If* (3.5) *holds, then there exist positive constants C_1 and M such that*

$$|\beta_k^{HS}| \leq C_1\|s_{k-1}\|, \quad \|p_k\| \leq M, \tag{3.6}$$

where

$$p_k = \begin{cases} -g_k, & if \quad g_k^T y_{k-1} \leq 0 \ \ or \ k \in K_1, \\ -g_k + \theta_k^{HS} y_{k-1}, & if \quad k \in K_2. \end{cases} \tag{3.7}$$

Proof. Clearly, it is sufficient to consider the case where $g_k^T d_{k-1} > 0$ and $g_k^T y_{k-1} > 0$, which corresponds to $k \in K_2$. Note that from definition of the parameter θ_k^{HS} as (1.10), where $\omega_k = 1$, it follows

$$\theta_k^{HS} = \frac{g_k^T d_{k-1}}{d_{k-1}^T (g_k - g_{k-1})} \in [0, 1). \tag{3.8}$$

We have from (3.7), (3.8), and the limitation on $\|g(x)\|$ in Ω that

$$\begin{aligned} \|p_k\| &\leq \|g_k\| + |\theta_k^{HS}| \cdot \|y_{k-1}\| \\ &= \|g_k\| + \frac{g_k^T d_{k-1}}{d_{k-1}^T y_{k-1}} \cdot \|g_k - g_{k-1}\| \\ &\leq \|g_k\| + \|g_k - g_{k-1}\| \\ &= 3\gamma. \end{aligned} \tag{3.9}$$

So, setting $M = 3\gamma$, we get $\|p_k\| \leq M$.

Now, we estimate a bound for β_k. Actually,

$$|\beta_k^{HS}| = |\frac{g_k^T y_{k-1}}{d_{k-1}^T y_{k-1}}| \leq \frac{L\|g_k\|\|s_{k-1}\|}{(1-\sigma)\varepsilon^2} \leq \frac{2BL\gamma}{(1-\sigma)\varepsilon^2}. \tag{3.10}$$

Subsequently, we give the convergence result of our presented method. It should be pointed out that the proof is analogous to that of Theorem 3.2 in [27] and we omit it here. □

Theorem 3.6. *Suppose that Assumptions 3.4 hold. Let $\{x_k\}_{k\geq 1}$ be generated by Algorithm 3.1. If (3.5) holds, then $\liminf_{k\to\infty} \|g_k\| = 0$.*

4 Numerical Results

In this section, we report some numerical results on a set of 73 nonlinear unconstrained problems. For each test problem, the dimension n is set to 10000 and the Fortran expression of its function and gradient can be downloaded from Andrei's website: `http://www.ici.ro/camo/neculai/SCALCG/evalfg.for`.

The following CG methods in the form of (1.9) or (1.3), only different in the choice of the CG parameter, are test:

1. The HZ (CG_DESCENT) method [27]: The CG method with the parameter
$\beta_k^{HZ} = \max\left\{\beta_k^{HS} - 2\frac{\|y_{k-1}\|^2}{(d_{k-1}^T y_{k-1})^2} g_k^T d_{k-1}, \frac{-1}{\|d_{k-1}\| \min\{\eta, \|g_{k-1}\|\}}\right\}$, where $\eta = 0.1$.
2. The EZZL method [7]: The three–term CG method with the search directions defined by (1.9) and (1.10) with the parameter ω_k defined by (1.14), where $\xi =\ 0.96$ is an optimal value, please see [7].
3. The ZZL method [38]: The CG method similar to EZZL method with the parameter $\omega_k = 1$.
4. The CZZL method: Algorithm 3.1.

All algorithms use exactly the same implementation of the Wolfe line search conditions with $\rho = 10^{-4}, \sigma = 0.6$. We stop the iterations if the inequality $\|g_k\|_\infty \leq 10^{-6}$ is satisfied.

The detailed numerical results, including the CPU time in seconds and the number of iterations, the total number of function evaluations, and gradient evaluations implementation for each of the tested method, can be found in the part of Appendix.

Further to demonstrate the efficiency of these methods, the use of profiles of Dolan and Moŕe [16] will present a wealth of information including efficiency and robustness. More analytically, the left side of the figure presents the percentage of test problem for which a method performs fastest, the right side gives the percentage of the test problems that are successfully solved. The top curve is the method that solved the most problems in a time that was within a factor ω of the best time.

To some extent, as can be seen from the Figs.1, 2 and 3, the curves of the CZZL method approximately solves 60% of the test problems with the least number of iterations, 55% the number of function and gradient evaluations. Obviously, the three figures above graphically illustrate that the curve of "CZZL" is always the top performer for almost all values of ω.

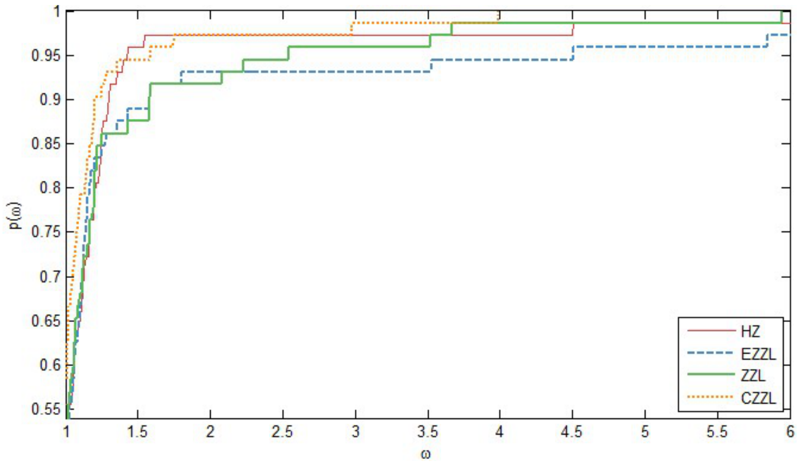

Figure 1: Performance based on iterations

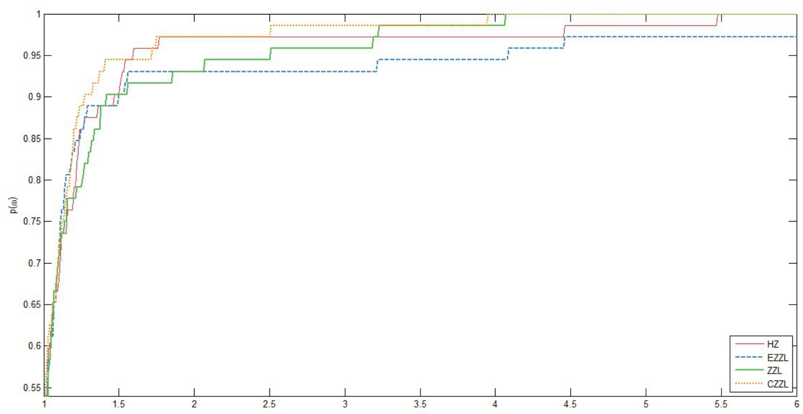

Figure 2: Performance based on function evaluations

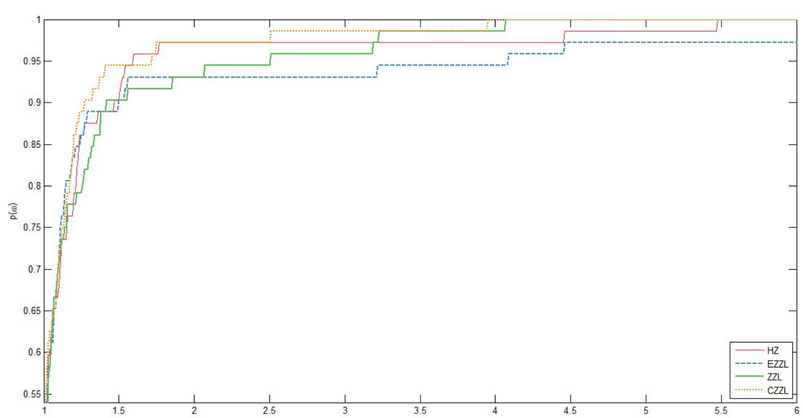

Figure 3: Performance based on gradient evaluations

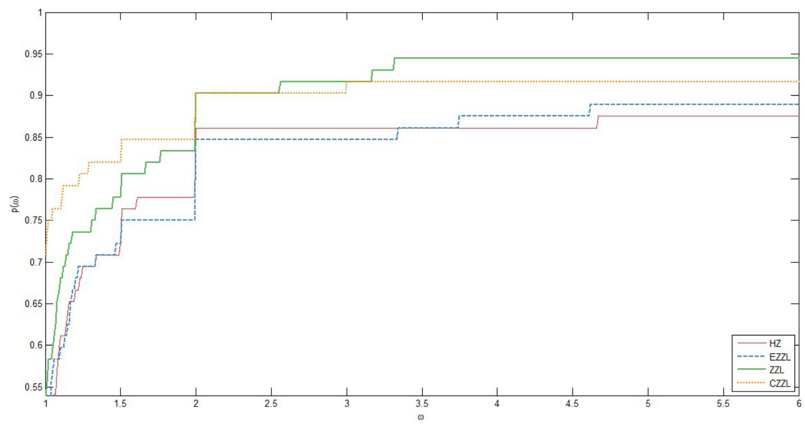

Figure 4: Performance based on CPU time

Meanwhile, as we see in Fig.4, the best performance, relative to the number of the CPU time, is obtained by the "CZZL" method, followed by the "HZ" method. Noted that the optimal parameter is set as $\xi = 1$ in the ZZL method and $\xi = 0.96$ in the EZZL method, which determines the curves of the two algorithms are close. However, as a variant of the EZZL or ZZL methods, the direction of the CZZL method needs less computational cost for inner products than these two methods do. This is maybe the reason why CZZL method is slightly superior to the EZZL and ZZL methods in Fig.4.

Since all methods are implemented with the same condition of line search, we conclude that the CZZL method is efficient for solving large scale test problems.

5 Conclusions

We establish a relationship between the EZZL method and the ZZL one, which adds us to understanding to one of intrinsic advantages of ZZL method. That is, the condition number $Cond_2(Q_k)$ and the spectral condition number $\kappa_2(Q_k)$ attain their minimum values, which is no other than the original ZZL method. Based on this fact, we proposed another three–term HS type CG method, which is essentially designed based on an adaptive switch from the ZZL method to the HS+ one for $g_k^T d_{k-1} \leq 0$. The strategy is justified by the fact that the automatic approximate restart property can effectively avoid jamming, i.e., generating many tiny steps without significant progress to the solution. Under mild conditions, we show that the proposed method converges globally for general objective functions. Computationally, the proposed CZZL method slightly outperforms the HZ, EZZL and ZZL methods.

Acknowledgements

The authors thank associate editor and two anonymous referees for their helpful comments, which have made the paper clearer and more comprehensive than the earlier version.

References

[1] M. Al–Baali, Y. Narushima and H. Yabe, A family of three–term conjugate gradient methods with sufficient descent property for unconstrained optimization, *Comput.Optim. Appl.* 60 (2015) 89–110.

[2] M. Al–Baali, A. Caliciotti, G.Fasano and M.Roma, Exploiting damped techniques for nonlinear conjugate gradient methods, *Math. Methods Oper. Res.* 86 (2017) 501–522.

[3] N.Andrei, Open problems in nonlinear conjugate gradient algorithms for unconstrained optimization, *Bull. Malays. Math. Sci. Soc.* 34 (2011) 319–330.

[4] N.Andrei, A simple three–term conjugate gradient algorithm for unconstrained optimization, *J. Comput. Appl. Math.* 241 (2013) 19–29.

[5] 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.

[6] S. Babaie–Kafaki and R. Ghanbari, A descent family of Dai–Liao conjugate gradient methods. *Optim.Methods Softw.* 243 (2014) 583–591.

[7] S. Babaie–Kafaki and R. Ghanbari, An extended three–term conjugate gradient method with sufficient descent property, *Miskolc Math. Notes* 16 (2015) 45–55.

[8] S. Babaie–Kafaki and R. Ghanbari, A hybridization of the Hestenes–Stiefel and Dai–Yuan conjugate gradient methods based on a least–squares approach, *Optim.Methods Softw.* 30 (2014) 673–681.

[9] S. Babaie–Kafaki and R. Ghanbari, Two modified three–term conjugate gradient method with sufficient descent property. *Optim. Lett.* 8(2014) 2285–2297.

[10] W. Cheng and D. Li, An active set modified Polak-Ribière-Polyak method for large–scale nonlinear bound constrained optimization, J. Optim. Theory Appl. 155 (2012) 1084–1094.

[11] Y. Dai, Conjugate gradient methods with Armijo–type line searches, *Acta Math. Appl. (Sin–E)* 298 (2002) 123–130.

[12] Y.Dai and Y.Yuan, A nonlinear conjugate gradient method with a strong global convergence property, *SIAM J. Optim.* 10 (1999) 177–182.

[13] Y.Dai and L. Liao, New conjugacy conditions and related nonlinear conjugate gradient methods, *Appl. Math. Optim.* 43 (2001) 87–101.

[14] Y. Dai and C. Kou, A nonlinear conjugate gradient algorithm with an optimal property and an improved Wolfe line search, *SIAM J. Optim.* 23 (2013) 296–320.

[15] Y. Dai and C. Kou, A Barzilai–Borwein conjugate gradient method, *Sci. China Math.* 59 (2016) 1511–1524.

[16] E.D.Dolan and J.J. Moré, Benchmarking optimization software with performance profiles, *Math. Program.* 91(2,Ser.A) (2002) 201–213.

[17] Z. Dai and F. Wen, A generalizd apporach to sparse and stable portfolio optimizarion problme, *J. Ind. Manae. Optim.* 14 (2018) 1561–1666.

[18] Z. Dai and F. Wen, Some improved sparse and stable portfolio optimization problems, *Financ Res. Lett.* 27 (2018) 46-52.

[19] X. Dong, H. Liu and Y. He, A self–adjusting conjugate gradient method with sufficient descent condition and conjugacy condition, *J. Optim. Theory Appl.* 165 (2015) 225–241.

[20] X. Dong, D. Han, Z.Dai, X.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] X. Dong, H. Liu, Y.He and X.Yang, A modified Hestenes–Stiefel conjugate gradient method with sufficient descent condition and conjugacy condition, *J. Comput. Appl. Math.* 281 (2015) 239–249.

[22] X. Dong, D. Han, R. Ghanbari, X. Li and Z.Dai, Some new three–term Hestenes–Stiefel conjugate gradient methods with affine combination, *Optimization*, 66 (2017) 1–18.

[23] M. Fatemi, An optimal parameter for Dai–Liao family of conjugate gradient methods, *J. Optim. Theory Appl.* 169 (2016) 587–605.

[24] R.Fletcher and C.Reeves, Function minimization by conjugate gradients, *Comput. J.* 7 (1964) 149–154.

[25] J.C.Gilbert and J.Nocedal, Global convergence properties of conjugate gradient methods for optimization, *SIAM J. Optim.* 2 (1992) 21–42.

[26] W.W.Hager and H.Zhang, A survey of nonlinear conjugate gradient methods, *Pac. J. Optim.* 2 (2006) 35–58.

[27] W. W. Hager and H. Zhang, A new conjugate gradient method with guaranteed and an efficient line search, *SIAM J. Optim.* 16 (2005) 170–192.

[28] W. W. Hager and H. Zhang, The limited memory conjugate gradient method, *SIAM J. Optim.* 23 (2013) 2150–2168.

[29] M.R.Hestenes and E.Stiefel, Methods of conjugate gradients for solving linear systems, *J. Res. Natl.Bur. Stand.* 49 (1952) 409–436.

[30] X. Li, W.Zhang and X.Dong, A class of modified FR conjugate gradient method and applications to non–negative matrix factorization, *Comput. Math. Appl.* 73 (2016) 270–276.

[31] Y. Narushima, H. Yabe and J. A. Ford, A three–term conjugate gradient method with sufficient descent property for unconstrained optimization, *SIAM J. Optim.* 21 (2011) 212–230.

[32] A. Perry, A modified conjugate gradient algorithm, *Oper.Res.* 26 (1976) 1073–1078.

[33] K. Sugiki,Y. Narushima Y and H. Yabe, Globally convergent three–term conjugate gradient methods that use secant conditions and generate descent search directions for unconstrained optimization, *J. Optim. Theory Appl.* 153 (2012) 733–757.

[34] D. S. Watkins, Fundamentals of matrix computations, John Wiley and Sons, New York, 2010.

[35] G. Yu, L. Guan and W.Chen, Spectral conjugate gradient methods with sufficient descent property for large–scale unconstrained optimization, *Optim.Methods Softw.* 23 (2008) 275–293.

[36] L. Zhang, W. Zhou and D. Li, A descent modified Polak–Ribière–Polyak conjugate gradient method and its global convergence, *IMA J. Numer. Anal.* 26 (2006) 629–640.

[37] L. Zhang, W. Zhou and D. Li, Global convergence of a modified Fletcher–Reeves conjugate gradient method with Armijo–type line search, *Numer.Math.* 104 (2006) 561–572.

[38] L. Zhang, W. Zhou and D. Li, Some descent three—term conjugate gradient methods and their global convergence, *Optim. Methods Softw* 22 (2007) 697–711.

Manuscript received 17 April 2017
revised 6 November 2018
accepted for publication 31 December 2018

Xiaoliang Dong
College of Science, Xi'an Shiyou University
Xi'an, 710065, P.R. China
School of Mathematics Science, Nanjing Normal University
Nanjing, P.R. China
E-mail address: dongxl@stu.xidian.edu.cn

Deren Han
School of Mathematics and Systems Science
Beijing Advanced Innovation Center for Big Data and Brain Computing (BDBC)
Beihang University, Beijing,100191, PR China.
E-mail address: handr@buaa.edu.cn

Appendix: Numerical Results

In this appendix, we provide details of the results of our numerical experiments that were summarized in Sect.4. The following table provides the number of iterations, function and gradient evaluation counts and CPU time for the implemented methods HZ, ZZL, EZZL and CZZL.

Table 1: Comparison of efficiency with other algorithms

test function	method	iter	NF	NG	time	$f(x^*)$	$\|g(x^*)\|$
1 Extended Freudenstein and Roth	HZ	14	31	23	0.1560E-01	0.244921E+06	0.414957E-11
	EZZL	14	35	26	0.1560E-01	0.244921E+06	0.372791E-06
	ZZL	12	28	21	0.1560E-01	0.244921E+06	0.616751E-11
	CZZL	19	40	28	0.3120E-01	0.244921E+06	0.173857E-08
2 Extended Trigonometric	HZ	84	169	85	0.4836E+00	0.287021E-07	0.955929E-06
	EZZL	78	158	81	0.4836E+00	0.363714E-07	0.464708E-06
	ZZL	72	146	75	0.4524E+00	0.371338E-07	0.876825E-06
	CZZL	68	138	71	0.4056E+00	0.410899E-07	0.649677E-06
3 Extended Rosenbrock	HZ	35	115	93	0.3120E-01	0.413633E-10	0.883297E-06
	EZZL	34	81	57	0.3120E-01	0.126291E-16	0.194573E-08
	ZZL	29	76	55	0.1560E-01	0.115556E-12	0.170309E-06
	CZZL	30	83	61	0.1560E-01	0.274934E-13	0.350440E-08
4 Extended White and Holst	HZ	39	120	88	0.4680E-01	0.241601E-12	0.111298E-06
	EZZL	31	68	45	0.3120E-01	0.469923E-19	0.246188E-10
	ZZL	28	70	48	0.3120E-01	0.886802E-17	0.137291E-08
	CZZL	30	84	61	0.3120E-01	0.359118E-09	0.157459E-06
5 Extended Beale	HZ	14	29	16	0.1560E-01	0.395971E-09	0.828139E-06
	EZZL	12	26	16	0.1560E-01	0.186754E-13	0.491168E-08
	ZZL	14	29	16	0.0000E+00	0.928397E-12	0.451021E-07
	CZZL	13	27	16	0.1560E-01	0.109417E-13	0.676160E-08
6 Extended Penalty	HZ	39	77	42	0.3120E-01	0.945324E+04	0.432339E-06
	EZZL	38	75	41	0.3120E-01	0.945324E+04	0.457665E-06
	ZZL	39	77	44	0.3120E-01	0.945324E+04	0.873929E-06
	CZZL	37	74	39	0.3120E-01	0.945324E+04	0.566525E-06
7 Perturbed Quadratic	HZ	556	1113	557	0.4836E+00	0.195334E-12	0.971561E-06
	EZZL	556	1113	557	0.4680E+00	0.195334E-12	0.971561E-06
	ZZL	556	1113	557	0.4836E+00	0.195334E-12	0.971561E-06
	CZZL	556	1113	557	0.4524E+00	0.195334E-12	0.971561E-06
8 Raydan 1	HZ	697	969	1124	0.1451E+01	0.500050E+07	0.987437E-06
	EZZL	696	958	1132	0.1466E+01	0.500050E+07	0.981211E-06
	ZZL	617	883	970	0.1264E+01	0.500050E+07	0.974616E-06
	CZZL	700	958	1144	0.1404E+01	0.500050E+07	0.956716E-06
9 Raydan 2	HZ	4	9	5	0.0000E+00	0.100000E+05	0.696901E-06
	EZZL	4	9	5	0.0000E+00	0.100000E+05	0.881940E-06

Table 1(Continued)

No test function	method	iter	NF	NG	time	$f(x^*)$	$\|g(x^*)\|$
	ZZL	4	9	5	0.0000E+00	0.100000E+05	0.874238E-06
	CZZL	4	9	5	0.1560E-01	0.100000E+05	0.876624E-06
10 Diagonal 1	HZ	684	932	1146	0.1638E+01	-0.385558E+09	0.999901E-06
	EZZL	685	937	1156	0.1700E+01	-0.385558E+09	0.996549E-06
	ZZL	664	919	1111	0.1638E+01	-0.385558E+09	0.987551E-06
	CZZL	650	910	1082	0.1513E+01	-0.385558E+09	0.988588E-06
11 Diagonal 2	HZ	532	1072	588	0.1310E+01	0.521304E+02	0.957067E-06
	EZZL	530	1059	597	0.1373E+01	0.521304E+02	0.764140E-06
	ZZL	513	1010	597	0.1310E+01	0.521304E+02	0.976715E-06
	CZZL	513	1010	597	0.1310E+01	0.521304E+02	0.976715E-06
12 Diagonal 3	HZ	752	958	1300	0.2902E+01	-0.499570E+08	0.993599E-06
	EZZL	758	971	1305	0.2964E+01	-0.499570E+08	0.991946E-06
	ZZL	730	945	1247	0.2792E+01	-0.499570E+08	0.983214E-06
	CZZL	683	899	1152	0.2543E+01	-0.499570E+08	0.989931E-06
13 Hager	HZ	84	135	131	0.2340E+00	-0.218141E+07	0.953351E-06
	EZZL	85	135	134	0.2496E+00	-0.218141E+07	0.882455E-06
	ZZL	85	135	134	0.2340E+00	-0.218141E+07	0.903042E-06
	CZZL	78	127	121	0.2184E+00	-0.218141E+07	0.998525E-06
14 Generalized Tridiagonal 1	HZ	22	39	29	0.3120E-01	0.999721E+04	0.972110E-06
	EZZL	22	38	30	0.1560E-01	0.999721E+04	0.983892E-06
	ZZL	22	38	30	0.3120E-01	0.999721E+04	0.999055E-06
	CZZL	22	38	30	0.1560E-01	0.999721E+04	0.998560E-06
15 Extended Tridiagonal 1	HZ	12	25	15	0.1560E-01	0.188017E-05	0.928034E-06
	EZZL	19	39	26	0.1560E-01	0.182143E-05	0.415190E-06
	ZZL	19	39	26	0.1560E-01	0.127890E-05	0.320957E-06
	CZZL	21	43	22	0.1560E-01	0.131771E-06	0.436183E-06
16 Extended Three Expo Terms	HZ	7	16	9	0.3120E-01	0.127963E+05	0.302989E-06
	EZZL	11	24	16	0.6240E-01	0.127963E+05	0.355738E-11
	ZZL	11	22	13	0.4680E-01	0.127963E+05	0.957499E-06
	CZZL	7	16	9	0.3120E-01	0.127963E+05	0.873383E-06
17 Generalized Tridiagonal 2	HZ	45	88	49	0.4680E-01	0.111485E+01	0.884245E-06
	EZZL	48	94	52	0.4680E-01	0.111485E+01	0.850410E-06
	ZZL	48	94	52	0.6240E-01	0.111485E+01	0.909493E-06
	CZZL	48	93	53	0.4680E-01	0.111485E+01	0.846675E-06
18 Diagonal 4	HZ	3	7	5	0.0000E+00	0.233932E-11	0.305896E-07
	EZZL	3	7	5	0.0000E+00	0.129131E-12	0.718695E-08
	ZZL	3	7	5	0.0000E+00	0.635480E-12	0.159434E-07
	CZZL	3	7	5	0.0000E+00	0.648384E-12	0.161045E-07
19 Diagonal 5	HZ	2	6	4	0.1560E-01	0.693147E+04	0.103219E-06
	EZZL	2	6	4	0.3120E-01	0.693147E+04	0.148136E-06
	ZZL	2	6	4	0.1560E-01	0.693147E+04	0.146374E-06
	CZZL	2	6	4	0.1560E-01	0.693147E+04	0.146374E-06
20 Extended Himmelblau	HZ	8	20	13	0.0000E+00	0.694120E-10	0.964836E-06
	EZZL	9	22	14	0.1560E-01	0.614989E-19	0.259490E-10
	ZZL	9	22	14	0.1560E-01	0.148286E-19	0.105267E-10
	CZZL	8	20	13	0.0000E+00	0.874962E-14	0.118941E-07
21 Generalized PSC1	HZ	546	900	1072	0.2122E+01	0.999872E+04	0.709404E-06

Table 1(Continued)

No test function	method	iter	NF	NG	time	$f(x^*)$	$\|g(x^*)\|$
	EZZL	419	754	728	0.1732E+01	0.999872E+04	0.968782E-06
	ZZL	658	1040	1359	0.2496E+01	0.999872E+04	0.999025E-06
	CZZL	568	894	1159	0.2122E+01	0.999872E+04	0.947480E-06
22 Extended PSC1	HZ	12	26	15	0.4680E-01	0.386600E+04	0.756732E-10
	EZZL	10	21	11	0.3120E-01	0.386600E+04	0.224436E-06
	ZZL	10	21	11	0.3120E-01	0.386600E+04	0.444246E-06
	CZZL	12	25	15	0.4680E-01	0.386600E+04	0.578734E-06
23Extended Powell	HZ	52	106	58	0.4680E-01	0.240337E-07	0.664155E-06
	EZZL	683	1368	786	0.5148E+00	0.759875E-07	0.996799E-06
	ZZL	108	219	121	0.7800E-01	0.151117E-07	0.314773E-06
	CZZL	207	418	248	0.1404E+00	0.235726E-06	0.972554E-06
24Extended Block-Diagonal BD1	HZ	19	50	42	0.4680E-01	0.314873E-09	0.316863E-06
	EZZL	19	52	44	0.6240E-01	0.339858E-10	0.106174E-06
	ZZL	17	53	46	0.4680E-01	0.135461E-09	0.865541E-06
	CZZL	14	41	36	0.4680E-01	0.695022E-09	0.673971E-06
25 Extended Maratos	HZ	57	191	155	0.6240E-01	-0.500312E+04	0.266716E-07
	EZZL	52	153	120	0.4680E-01	-0.500312E+04	0.864224E-08
	ZZL	46	127	96	0.4680E-01	-0.500312E+04	0.685762E-08
	CZZL	52	179	147	0.4680E-01	-0.500312E+04	0.612689E-06
26 Extended Cliff	HZ	32	66	35	0.6240E-01	0.998933E+03	0.730857E-06
	EZZL	34	78	48	0.7800E-01	0.998933E+03	0.962513E-10
	ZZL	33	74	44	0.7210E-01	0.998933E+03	0.912443E-10
	HZ	32	66	35	0.6240E-01	0.998933E+03	0.730857E-06
27 Quadratic Diagonal Perturbed	HZ	306	613	376	0.2964E+00	0.468112E-10	0.998910E-06
	EZZL	302	605	367	0.2964E+00	0.468478E-10	0.989436E-06
	ZZL	297	595	361	0.2964E+00	0.526881E-10	0.998999E-06
	CZZL	294	589	351	0.2808E+00	0.487871E-10	0.999760E-06
28 Extended Wood	HZ	164	384	241	0.1404E+00	0.295859E-09	0.733572E-06
	EZZL	374	772	412	0.3120E+00	0.118862E-09	0.914613E-06
	ZZL	373	764	404	0.2964E+00	0.136781E-09	0.980687E-06
	CZZL	106	240	147	0.9360E-01	0.603254E-10	0.181708E-06
29 Extended Hiebert	HZ	74	250	197	0.7800E-01	0.268614E-12	0.341791E-07
	EZZL	59	185	142	0.6240E-01	0.545382E-16	0.564764E-09
	ZZL	66	201	153	0.6240E-01	0.109291E-13	0.977542E-07
	CZZL	60	196	154	0.6240E-01	0.334222E-08	0.302646E-06
30 Quadratic QF1	HZ	558	1117	559	0.3744E+00	-0.500000E-04	0.975083E-06
	EZZL	558	1117	559	0.3744E+00	-0.500000E-04	0.975050E-06
	ZZL	558	1117	559	0.3900E+00	-0.500000E-04	0.975109E-06
	CZZL	558	1117	559	0.3588E+00	-0.500000E-04	0.975208E-06
31 Extended Quadratic Penalty QP1	HZ	17	35	21	0.1560E-01	0.399900E+05	0.225606E-08
	EZZL	17	34	20	0.1560E-01	0.399900E+05	0.428478E-08
	ZZL	17	35	21	0.1560E-01	0.399900E+05	0.104037E-09
	CZZL	17	35	21	0.1560E-01	0.399900E+05	0.550402E-08
32 Extended Quadratic Penalty QP2	HZ	46	139	105	0.2496E+00	0.315808E-19	0.596287E-08
	EZZL	39	95	65	0.1560E+00	0.110587E-15	0.134713E-06
	ZZL	44	119	88	0.2028E+00	0.665493E-19	0.795808E-10

Table 1(Continued)

No test function	method	iter	NF	NG	time	$f(x^*)$	$\|g(x^*)\|$
	CZZL	37	103	74	0.1716E+00	0.579174E-15	0.822382E-06
33 Quadratic QF2	HZ	1105	2019	1305	0.8268E+00	-0.100001E+01	0.982240E-06
	EZZL	1152	2068	1397	0.8580E+00	-0.100001E+01	0.963274E-06
	ZZL	1149	2065	1391	0.8424E+00	-0.100001E+01	0.976485E-06
	CZZL	1153	2068	1400	0.8268E+00	-0.100001E+01	0.999836E-06
34 Extended EP1	HZ	3	7	5	0.1560E-01	0.793176E+05	0.489886E-14
	EZZL	3	7	5	0.0000E+00	0.793176E+05	0.489886E-14
	ZZL	3	7	5	0.0000E+00	0.793176E+05	0.992401E-13
	CZZL	3	7	5	0.0000E+00	0.793176E+05	0.489886E-14
35 Extended Tridiagonal 2	HZ	34	60	52	0.3120E-01	0.389690E+04	0.823902E-06
	EZZL	34	59	53	0.3120E-01	0.389690E+04	0.841719E-06
	ZZL	36	60	56	0.3120E-01	0.389690E+04	0.762596E-06
	CZZL	30	52	44	0.1560E-01	0.389690E+04	0.973055E-06
36 BDQRTIC (CUTE)	HZ	982	2237	2276	0.1591E+01	0.400343E+05	0.836823E-06
	EZZL	7632	13753	18952	0.1296E+02	0.400343E+05	0.979984E-06
	ZZL	3602	7214	8154	0.5975E+01	0.400343E+05	0.937097E-06
	CZZL	2921	5608	8328	0.4898E+01	0.400343E+05	0.892730E-06
37 TRIDIA (CUTE)	HZ	1115	2231	1116	0.9516E+00	0.428965E-14	0.963923E-06
	EZZL	1115	2231	1116	0.1061E+01	0.425900E-14	0.999080E-06
	ZZL	1115	2231	1116	0.1030E+01	0.426354E-14	0.995306E-06
	CZZL	1118	2237	1119	0.8736E+00	0.346980E-14	0.970174E-06
38 ARWHEAD (CUTE)	HZ	10	23	16	0.1560E-01	0.000000E+00	0.334245E-11
	EZZL	10	23	16	0.1560E-01	0.000000E+00	0.477885E-11
	ZZL	7	15	8	0.0000E+00	0.000000E+00	0.962518E-06
	CZZL	8	17	11	0.1560E-01	0.000000E+00	0.198467E-07
39 NONDIA (CUTE)	HZ	11	32	26	0.1560E-01	0.888014E-17	0.987219E-06
	EZZL	10	30	25	0.1560E-01	0.101573E-18	0.145197E-06
	ZZL	12	38	32	0.1560E-01	0.177480E-23	0.267772E-10
	CZZL	10	41	36	0.1560E-01	0.478202E-14	0.899411E-09
40 NONDQUAR (CUTE)	HZ	10005	20031	10277	0.1036E+02	0.264657E-06	0.417689E-06
	EZZL	10009	20033	10144	0.1028E+02	0.345678E-06	0.348193E-06
	ZZL	5627	11261	5669	0.5694E+01	0.773771E-06	0.973591E-06
	CZZL	2223	4498	2803	0.2231E+01	0.468577E-05	0.958940E-06
41 DQDRTIC (CUTE)	HZ	7	15	8	0.1560E-01	0.710648E-13	0.529597E-06
	EZZL	7	15	8	0.0000E+00	0.575440E-14	0.156667E-06
	ZZL	7	15	8	0.0000E+00	0.419615E-14	0.128175E-06
	CZZL	7	15	8	0.1560E-01	0.483353E-14	0.146626E-06
42 EG2 (CUTE)	HZ	921	1621	1643	0.3011E+01	-0.999950E+04	0.835536E-06
	EZZL	183	326	430	0.6864E+00	-0.999950E+04	0.769429E-06
	ZZL	227	549	395	0.8268E+00	-0.999950E+04	0.959097E-06
	CZZL	102	296	253	0.4680E+00	-0.999900E+04	0.515334E-06
43 DIXMAANA (CUTE)	HZ	9	19	10	0.1560E-01	0.100000E+01	0.374357E-07
	EZZL	8	17	9	0.1560E-01	0.100000E+01	0.135182E-06
	ZZL	8	17	9	0.1559E-01	0.100000E+01	0.183715E-06
	CZZL	7	15	8	0.1560E-01	0.100000E+01	0.160362E-09
44 DIXMAANB (CUTE)	HZ	9	19	10	0.1560E-01	0.100000E+01	0.170910E-06
	EZZL	8	17	9	0.1560E-01	0.100000E+01	0.733504E-06

Table 1(Continued)

No test function	method	iter	NF	NG	time	$f(x^*)$	$\|g(x^*)\|$
	ZZL	8	17	9	0.1559E-01	0.100000E+01	0.291293E-06
	CZZL	8	17	9	0.1560E-01	0.100000E+01	0.338649E-07
45 DIXMAANC (CUTE)	HZ	10	21	11	0.3120E-01	0.100000E+01	0.601031E-06
	EZZL	10	21	11	0.1560E-01	0.100000E+01	0.295833E-06
	ZZL	10	21	12	0.1559E-01	0.100000E+01	0.246996E-07
	CZZL	9	19	10	0.1560E-01	0.100000E+01	0.140561E-06
46 DIXMAANE (CUTE)	HZ	376	753	377	0.7176E+00	0.100000E+01	0.994610E-06
	EZZL	380	761	381	0.7332E+00	0.100000E+01	0.970189E-06
	ZZL	380	761	381	0.7488E+00	0.100000E+01	0.971610E-06
	CZZL	381	763	382	0.7020E+00	0.100000E+01	0.991286E-06
47 Partial Perturbed Quadratic PPQ1	HZ	19	39	22	0.8970E+01	0.262885E-13	0.709103E-06
	EZZL	19	39	22	0.8970E+01	0.259904E-13	0.728662E-06
	ZZL	19	39	22	0.9001E+01	0.259925E-13	0.728450E-06
	CZZL	19	39	23	0.9095E+01	0.259297E-13	0.726961E-06
48 Broyden Tridiagonal	HZ	33	67	34	0.3120E-01	0.120744E-12	0.921929E-06
	EZZL	42	86	45	0.4681E-01	0.499226E-13	0.641751E-06
	ZZL	47	95	48	0.4680E-01	0.171076E-12	0.899517E-06
	CZZL	39	79	40	0.4680E-01	0.642722E-13	0.663415E-06
49 Almost Perturbed Quadratic	HZ	558	1117	559	0.4056E+00	0.185646E-12	0.964790E-06
	EZZL	558	1117	559	0.4212E+00	0.185646E-12	0.964791E-06
	ZZL	558	1117	559	0.4056E+00	0.185646E-12	0.964791E-06
	CZZL	558	1117	559	0.3588E+00	0.185646E-12	0.964794E-06
50 Tridiagonal Perturbed Quadratic	HZ	526	1053	527	0.5460E+00	0.124535E-12	0.990364E-06
	EZZL	527	1055	528	0.5460E+00	0.110222E-12	0.932288E-06
	ZZL	527	1055	528	0.5772E+00	0.110221E-12	0.933853E-06
	CZZL	527	1055	528	0.4992E+00	0.110216E-12	0.940543E-06
51 EDENSCH (CUTE)	HZ	25	41	38	0.3120E-01	0.600033E+05	0.724594E-06
	EZZL	24	40	35	0.3120E-01	0.600033E+05	0.667480E-06
	ZZL	24	41	33	0.3120E-01	0.600033E+05	0.937401E-06
	CZZL	25	46	39	0.1560E-01	0.600033E+05	0.699751E-06
52 VARDIM (CUTE)	HZ	8	18	11	0.3120E-01	-0.107966E+04	0.812432E-06
	EZZL	8	18	11	0.1560E-01	-0.107966E+04	0.139305E-06
	ZZL	7	16	10	0.3120E-01	-0.107966E+04	0.235044E-07
	CZZL	8	22	14	0.3120E-01	-0.107966E+04	0.365119E-07
53 STAIRCASE S1	HZ	19653	39307	19654	0.1605E+02	0.520081E-09	0.995424E-06
	EZZL	19922	39845	19924	0.1560E+02	0.124917E-10	0.990152E-06
	ZZL	19854	39709	19855	0.1552E+02	0.549551E-10	0.996042E-06
	CZZL	19919	39839	19920	0.1568E+02	0.157027E-08	0.994485E-06
54 LIARWHD (CUTE)	HZ	20	49	35	0.1559E-01	0.334500E-16	0.473808E-06
	EZZL	19	41	28	0.1559E-01	0.301349E-18	0.212609E-06
	ZZL	23	54	41	0.1561E-01	0.445938E-19	0.270235E-07
	CZZL	20	52	37	0.3120E-01	0.627242E-19	0.100180E-06
55 Diagonal 6	HZ	4	9	5	0.1559E-01	0.238032E-08	0.689838E-06
	EZZL	4	9	5	0.1561E-01	0.397016E-08	0.891065E-06
	ZZL	4	9	5	0.0000E+00	0.389466E-08	0.882629E-06
	CZZL	4	9	5	0.1560E-01	0.389466E-08	0.882607E-06
56 DIXON3DQ (CUTE)	HZ	10000	20001	10002	0.7769E+01	0.999900E-04	0.138933E-10

Table 1(Continued)

No test function	method	iter	NF	NG	time	$f(x^*)$	$\|g(x^*)\|$
	EZZL	10000	20001	10002	0.7550E+01	0.999900E-04	0.251426E-10
	ZZL	10000	20001	10002	0.7566E+01	0.999900E-04	0.246421E-10
	CZZL	10000	20001	10002	0.7613E+01	0.999900E-04	0.542692E-09
57 DIXMAANF (CUTE)	HZ	26	46	37	0.3119E-01	0.110993E+05	0.626820E-06
	EZZL	24	45	33	0.3120E-01	0.110993E+05	0.910804E-06
	ZZL	25	44	36	0.3120E-01	0.110993E+05	0.561396E-06
	CZZL	20	38	28	0.3120E-01	0.110993E+05	0.657467E-06
58 DIXMAANG (CUTE)	HZ	11	23	12	0.3120E-01	0.385249E-09	0.723351E-06
	EZZL	11	23	12	0.1559E-01	0.437943E-12	0.210481E-07
	ZZL	11	23	12	0.1561E-01	0.262043E-12	0.162254E-07
	CZZL	11	23	12	0.1560E-01	0.410242E-18	0.168796E-10
59 DIXMAANH (CUTE)	HZ	12	28	17	0.4680E-01	0.965397E-13	0.111864E-07
	EZZL	15	31	20	0.4680E-01	0.398032E-15	0.582973E-09
	ZZL	14	34	24	0.4680E-01	0.276208E-16	0.495483E-09
	CZZL	15	37	23	0.4680E-01	0.145431E-10	0.216339E-06
60 DIXMAANI (CUTE)	HZ	7	15	8	0.0000E+00	0.125079E-08	0.954422E-06
	EZZL	6	13	7	0.1559E-01	0.443711E-18	0.188267E-10
	ZZL	6	13	7	0.0000E+00	0.661434E-21	0.727418E-12
	CZZL	7	15	8	0.0000E+00	0.301811E-10	0.219250E-06
61 DIXMAANJ (CUTE)	HZ	10	21	11	0.1561E-01	0.425827E-11	0.515915E-06
	EZZL	10	22	12	0.1559E-01	0.314254E-11	0.448012E-06
	ZZL	10	22	12	0.1559E-01	0.278145E-13	0.412097E-07
	CZZL	11	23	12	0.1560E-01	0.542242E-15	0.588938E-08
62 DIXMAANK (CUTE)	HZ	390	980	676	0.2044E+01	0.736996E-04	0.640079E-06
	EZZL	1834	4001	2421	0.7644E+01	0.638282E-08	0.991624E-06
	ZZL	1868	3988	2382	0.6770E+01	0.170386E-12	0.683829E-06
	CZZL	314	1088	906	0.2621E+01	0.575590E-04	0.515243E-06
63 DIXMAANL (CUTE)	HZ	5000	10001	5001	0.3869E+01	0.216929E-13	0.645268E-07
	EZZL	5000	10001	5001	0.3713E+01	0.295826E-12	0.809637E-07
	ZZL	5000	10001	5001	0.3744E+01	0.936852E-12	0.953449E-07
	CZZL	5001	10003	5003	0.3666E+01	0.396277E-13	0.194162E-06
64 ENGVAL1 (CUTE)	HZ	11	23	12	0.1561E-01	0.107462E-10	0.168403E-06
	EZZL	12	25	15	0.3119E-01	0.161700E-11	0.384653E-07
	ZZL	12	25	15	0.3120E-01	0.974852E-11	0.974821E-07
	CZZL	13	27	16	0.3120E-01	0.112284E-10	0.877603E-07
65 FLETCHCR (CUTE)	HZ	10	21	11	0.1559E-01	0.109312E-12	0.607747E-06
	EZZL	10	21	11	0.0000E+00	0.764413E-14	0.125744E-06
	ZZL	10	21	11	0.1561E-01	0.267752E-14	0.797907E-07
	CZZL	10	21	11	0.0000E+00	0.159916E-13	0.245999E-06
66 COSINE (CUTE)	HZ	5	12	7	0.1559E-01	-0.816849E+04	0.427394E-06
	EZZL	5	12	7	0.1559E-01	-0.816849E+04	0.703074E-06
	ZZL	5	12	7	0.1561E-01	-0.816849E+04	0.703459E-06
	CZZL	5	12	7	0.1560E-01	-0.816849E+04	0.753963E-06
67 Extended DENSCHNB (CUTE)	HZ	4	9	5	0.1559E-01	-0.480453E+04	0.650772E-06
	EZZL	4	9	5	0.1561E-01	-0.480453E+04	0.138452E-06
	ZZL	4	9	5	0.1559E-01	-0.480453E+04	0.191262E-06