A Descent Four-Term of Liu and Storey Conjugate Gradient Method for Large Scale Unconstrained Optimization Problems

Ahmad Alhawarat\textsuperscript{1}, Hanan Alolaiyan\textsuperscript{2}, Ibtisam A. Masmali\textsuperscript{3}, Zabidin Salleh\textsuperscript{4}, Shahrina Ismail\textsuperscript{5,}\textsuperscript{*}

\textsuperscript{1} Division of Computational Mathematics and Engineering, Institute for Computational Science, Ton Duc Thang University, Ho Chi Minh City, Vietnam
\textsuperscript{2} Department of Mathematics, College of Science, King Saud University, Riyadh 11451, Saudi Arabia
\textsuperscript{3} Department of Mathematics, College of Science, Jazan University, Jazan, Saudi Arabia
\textsuperscript{4} Department of Mathematics, Faculty of Ocean Engineering Technology and Informatics, Universiti Malaysia Terengganu, Terengganu, Malaysia
\textsuperscript{5} Financial Mathematics Program, Faculty of Science and Technology, Universiti Sains Islam Malaysia, Bandar Baru Nilai, 71800, Nilai, Negeri Sembilan, Malaysia

Abstract. The conjugate gradient (CG) method is a useful tool for obtaining the optimum point for unconstrained optimization problems since it does not require a second derivative or its approximations. Moreover, the conjugate gradient method can be applied in many fields such as machine learning, deep learning, neural network, and many others. This paper constructs a four-term conjugate gradient method that satisfies the descent property and convergence properties to obtain the stationary point. The new modification was constructed based on Liu and Storey’s conjugate gradient method, two-term conjugate gradient method, and three-term conjugate gradient method. To analyze the efficiency and robustness, we used more than 150 optimization functions from the CUTEst library with different dimensions and shapes. The numerical results show that the new modification outperforms the recent conjugate gradient methods such as CG-Descent, Dai and Liao, and others in terms of number of functions evaluations, number of gradient evaluations, number of iterations, and CPU time.

2020 Mathematics Subject Classifications: 65K10, 90C25, 90C26

Key Words and Phrases: Conjugate gradient method, Wolfe Powell line search, Descent condition, Convergence

*Corresponding author.
DOI: https://doi.org/10.29020/nybg.ejpam.v14i4.4128
Email addresses: shahrinaismail@usim.edu.my (S. Ismail)
1. Introduction

To solve unconstrained optimization problems, normally, we use the conjugate gradient (CG) method since it does not require memory storage or the second derivative of the objective function. We consider the following problem:

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

where \( f(x) \) satisfies the following assumption.

**Assumption 1**

A. \( f : \mathbb{R}^n \rightarrow \mathbb{R} \) is a continuous and differentiable function, and the gradient is available.

B. The level set \( \Psi = \{ x | f(x) \leq f(x_1) \} \) is bounded, that is, a positive constant \( \vartheta \) exists such that

\[ \|x\| \leq \vartheta, \forall x \in \Psi. \]

C. In some neighbourhood \( Q \) of \( \Psi \), \( f \) is continuously differentiable, and its gradient is Lipschitz continuous; that is, for all, \( x, y \in Q \), there exists a constant \( L > 0 \) such that

\[ \|g(x) - g(y)\| \leq L \|x - y\|. \]

In addition, from Assumption 1, we can conclude that there exists a positive constant \( B \) such that

\[ \|g(u)\| \leq B, \forall u \in N. \]

The CG method generates a sequence of \( x_k \) starting from the initial point \( x_1 \) by the equation

\[ x_{k+1} = x_k + \alpha_k d_k, \quad k = 1, 2, ..., \]  

where \( x_{k+1} \) is the next iteration.

The search direction \( d_k \) in the CG method is defined by the following equation

\[ d_k = \begin{cases} -g_k, & \text{if } k = 1, \\ -g_k + \beta_k d_{k-1}, & \text{if } k \geq 2, \end{cases} \]  

where \( g_k = g(x_k) = \nabla f \) and \( \beta_k \) is known as the CG formula. Note that for \( k = 1 \), we use the steepest descent method. To obtain the steplength \( (\alpha_k) \), we have the following two-line searches:

A - Exact line search: To find the step size such that the objective function in the search direction is minimized i.e.

\[ f(x_k + \alpha_k d_k) = \min f(x_k + \alpha d_k), \alpha \geq 0. \]

However, exact optimal step size generally can not be found, and it is expensive to find almost exact step size [29]. Thus, we use inexact line search, as discussed in part B.

B - Inexact line search
To avoid expensive computational to obtain the step size by exact line search, we use inexact line search. Here, the most popular inexact line search is Wolfe Powell (WP) line search, which is divided into two parts:

**B1** - The first part is weak Wolfe Powell (WWP) \([30, 31]\) and is given by the following equations:

\[
 f(x_k + \alpha_k d_k) \leq f(x_k) + \delta \alpha_k g_k^T d_k,\]

\[
 g(x_k + \alpha_k d_k)^T d_k \geq \sigma g_k^T d_k. \tag{4}
\]

**B2** - The second part is strong Wolfe Powell (SWP) line search, which is defined by equation (4) and

\[
 |g(x_k + \alpha_k d_k)^T d_k| \leq \sigma |g_k^T d_k|, \tag{5}
\]

where \(0 < \delta < \sigma < 1\).

The most well known classical CG formulas are Hestenes–Stiefel (HS)\([19]\), Polak–Ribiere–Polyak (PRP)\([25]\), Liu and Storey (LS)\([23]\), Fletcher–Reeves (FR)\([14]\), Fletcher (CD)\([13]\), Dai and Yuan (DY)\([11]\), and these formulas are given as follows:

\[
 \beta_{HS}^k = \frac{g_k^T y_{k-1}}{d_k^T y_{k-1}}, \beta_{PRP}^k = \frac{g_k^T y_{k-1}}{\|g_k\|^2}, \beta_{LS}^k = -\frac{g_k^T y_{k-1}}{d_k^T g_{k-1}}, \beta_{FR}^k = \frac{\|g_k\|^2}{d_k^T g_{k-1}}, \beta_{CD}^k = -\frac{\|g_k\|^2}{d_k^T g_{k-1}}, \beta_{DY}^k = \frac{\|g_k\|^2}{d_k^T g_{k-1}},
\]

where \(y_{k-1} = g_k - g_{k-1}\).

The global convergence properties of the FR method were studied by Zoutendijk \([33]\) and Al-Baali \([5]\). The global convergence of the PRP method for convex objective function under exact line search was proved by Polak and Ribiere in \([25]\). Later, Powell \([26]\) gave out a counterexample showing that there exists a non-convex function, where PRP and HS CG methods can cycle infinitely without getting a solution. Therefore, Powell suggested the importance of achieving the global convergence of PRP and HS methods, which should be non-negative. Meanwhile, Gilbert and Nocedal \([15]\) proved that non-negative PRP, i.e. \(\beta_k = \max\{\beta_{PRP}^k, 0\}\), is globally convergent under complicated line searches. Alhawarat et al. \([2]\) also proposed the following non-negative CG formula with new restart property as follows

\[
 \beta_{AZPRP}^k = \begin{cases} 
 \frac{\|g_k\|^2 - \mu_k |g_k^T g_{k-1}|}{\|g_{k-1}\|^2}, & \text{if } \|g_k\|^2 > \mu_k |g_k^T g_{k-1}|, \\
 0, & \text{otherwise}, 
\end{cases}
\]

where \(\|\cdot\|\) represents the Euclidean norm, while \(\mu_k\) is defined as follows:

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

Furthermore, Dai and Liao \([10]\) proposed the following CG formula with a new conjugacy condition as follows:
\begin{align}
\beta_{DL}^k &= \frac{g_k^T y_{k-1}}{d_{k-1}^T y_{k-1}} - t \frac{g_k^T s_{k-1}}{d_{k-1}^T y_{k-1}} = \beta_{HS}^k - t \frac{g_k^T s_{k-1}}{d_{k-1}^T y_{k-1}}. \tag{6}
\end{align}

In addition, Hager and Zhang [17, 18] presented the following CG formula based on equation (6) given by

\begin{align}
\beta_{HZ}^k &= \max \{\beta_{N}^k, \eta_k\}, \tag{7}
\end{align}

where \(\beta_{N}^k = \frac{1}{d_k^T y_k} (y_k - 2d_k \|y_k\|^2) g_k, \eta_k = -\frac{1}{\|d_k\|} \min \{\eta, \|g_k\|\},\) and \(\eta > 0\) is a constant.

Note that if \(t = 2 \frac{\|y_k\|^2}{s_k^T y_k}\), then \(\beta_{N}^k = \beta_{DY}^k\).

Alhawarat et al. [1] also presented a four-term CG method based on equation (6) as follows:

\begin{align}
d_{k}^{FTCGHS} = -g_k + \left(\beta_{HS}^k - t_k \frac{g_k^T s_{k-1}}{y_k^T d_{k-1} y_{k-1}} \right) d_{k-1} - \left(\frac{g_k^T d_{k-1}}{y_k^T d_{k-1} y_{k-1}}\right) (y_{k-1} + s_{k-1}). \tag{8}
\end{align}

Furthermore, Zabidin et al. [32] presented the following CG formula based on [11] as follows

\begin{align}
\beta_{LS}^{k+1} = \begin{cases} 
-\frac{\|y_k\|^2 - \mu_k |g_k^T s_{k-1}|}{d_{k-1}^T y_{k-1}} & \text{if } \|g_k\|^2 > \mu_k |g_k^T s_{k-1}|, \\
\beta_{DL-HS}^k & \text{otherwise},
\end{cases} \tag{9}
\end{align}

where \(\|\cdot\|\) represents the Euclidean norm and

\[\beta_{DL-HS}^k = -\mu_k \frac{g_k^T s_{k-1}}{d_{k-1}^T y_{k-1}}.\]

Moreover, Liu et al. [22] proposed the three-term CG method as follows

\begin{align}
d_k = -g_k + \left(\beta_{LS}^k - \frac{\|g_{k-1}\|^2 g_k^T d_{k-1}}{(d_{k-1}^T g_{k-1})^2} \right) d_{k-1} + \left(\frac{g_k^T d_{k-1}}{d_{k-1}^T y_{k-1}}\right) g_{k-1},
\end{align}

with the following assumption

\[\left(\frac{g_k^T d_{k-1}}{d_{k-1}^T y_{k-1}}\right) > \nu \in (0, 1).\]

Additionally, Yao et al. [27] proposed three terms of CG with a new choice of \(t\) as follows:

\begin{align}
d_{k+1} = -g_{k+1} + \left(\frac{g_k^T y_k - t_k g_k^T s_k}{y_k^T d_k}\right) d_k + \frac{g_{k+1}^T d_k}{y_k^T d_k} y_k.
\end{align}
Based on the SWP line search, Yao et al. [27] selected $t_k$ to satisfy the descent condition as follows:

$$t_k > \frac{\|y_k\|^2}{y_k^T s_k}.$$  

The CG method can be applied in many fields such as medical science, neural network, image restoration, machine learning, finance, economics and many others. For example, Alhawarat et al. [1] presented an application for image restoration using the CG method to restore true images from damaged images with an efficient number of iterations and CPU time. Moreover, we can test the quality using the root-mean-square error (rmse) between the original image and the restored image as follows:

$$\text{rmse} = \frac{\|\varsigma - \iota_k\|_2}{\|\varsigma\|}.$$  

Here, $\varsigma$ and $\iota_k$ are the true image and restored images. For more about the CG method and its applications, the reader can refer to the following references [3, 4, 6, 20].

2. Preliminary

**Definition 2.1** [8]. A sequence of real numbers is a function whose domain is the set of natural numbers $\mathbb{N} = \{1, 2, \ldots\}$ and whose range is contained in $\mathbb{R}$. A sequence $a_n$ is considered increasing if $a_1 < a_2 < a_3 < \ldots < a_k \ldots$ that is, $a_k < a_{k+1}$ for all $k$. Similarly, the decreasing sequences can be defined.

**Definition 2.2** [8]. A sequence $a_n$ is said to converge to the limit $L$ for any given $\varepsilon > 0$. Then, there is a positive integer $N$ such that $|a_n - L| < \varepsilon$ for all $n \geq N$. In this case, we have $\lim_{n \to \infty} a_n = L$. A sequence that does not converge to some finite limit is called to diverge.

**Definition 2.3** [16]. An infinite series is an expression that can be written in the form of

$$\sum_{k=1}^{\infty} u_k = u_1 + u_2 + u_3 + \ldots + u_k + \ldots.$$  

The number $u_1$, $u_2$, $u_3$, $\ldots$ is called the term of the series.

**Definition 2.4** [16]. Let $a_n$ be the sequence of partial sums of the series $u_1 + u_2 + u_3 + \ldots + u_k + \ldots$. If the sequence $a_n$ converges to a limit $A$, then the series is convergent to $A$ and $A$ is called the sum of the series. This is defined by:

$$A = \sum_{k=1}^{\infty} u_k.$$  

If the sequence of partial sums diverges, then the series is diverging.
Theorem 2.1 [16]. A geometric series \( \sum_{k=1}^{\infty} ar^k = a + ar + ar^2 + \ldots + ar^k + \ldots \), where \( a \neq 0 \). Then, it is convergent if \( |r| \leq 1 \) and diverges if \( |r| > 1 \). If the series converges, then the sum is, \( \sum_{k=1}^{\infty} ar^k = \frac{a}{1-r} \).

Definition 2.5 [8]. For \( f: \mathbb{R}^n \to \mathbb{R} \) a function that is continuous and differentiable, then there exists at any point \( x \in \mathbb{R}^n \), a vector of first-order partial derivatives or a gradient vector given by

\[
\nabla f(x) = \begin{bmatrix}
\frac{\partial f(x)}{\partial x_1} \\
\frac{\partial f(x)}{\partial x_2} \\
\vdots \\
\frac{\partial f(x)}{\partial x_n}
\end{bmatrix} = g(x).
\]

Definition 2.6 [28]. Let function \( f: \mathbb{R}^n \to \mathbb{R} \) be twice continuously differentiable. Then, at a point \( x \in \mathbb{R}^n \), there exists a matrix of second-order partial derivatives or a Hessian matrix given by

\[
\nabla^2 f(x) = H(x) = \begin{bmatrix}
\frac{\partial^2 f(x)}{\partial x_1^2} & \frac{\partial^2 f(x)}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f(x)}{\partial x_1 \partial x_n} \\
\frac{\partial^2 f(x)}{\partial x_2 \partial x_1} & \frac{\partial^2 f(x)}{\partial x_2^2} & \cdots & \frac{\partial^2 f(x)}{\partial x_2 \partial x_n} \\
\vdots & \vdots & \ddots & \vdots \\
\frac{\partial^2 f(x)}{\partial x_n \partial x_1} & \frac{\partial^2 f(x)}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f(x)}{\partial x_n^2}
\end{bmatrix},
\]

where \( m, n \in \mathbb{N} \).

Definition 2.7 [8]. Quadratic form of a function \( f: \mathbb{R}^n \to \mathbb{R} \) is denoted by \( f(x) = \frac{1}{2}x^TQx - bx \), where \( x \in \mathbb{R}^n \), \( Q \) is a \( n \times n \) real matrix, and \( b \) is a constant.

Definition 2.8 [7]. A set \( C \) is convex if the line segment between any points in \( C \) lies in \( C \). For any \( x_1, x_2 \in C \) and any \( \theta \) with \( 0 \leq \theta \leq 1 \), the \( \theta x_1 + (1-\theta)x_2 \in C \).

Definition 2.9 [7]. A function \( f: \mathbb{R}^n \to \mathbb{R} \) is convex if the domain of \( f \) is a convex set for all \( x, y \in \text{domain } f \), and

\[
f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta)f(y),
\]

where \( 0 \leq \theta \leq 1 \).

3. The new search direction and its motivation

The CG method has become very rich in recent years. The main goal is to develop a new CG method robust and efficient to solve large scale unconstrained optimization problems. In addition, the CG method can be applied in several fields, as mentioned before. Thus, to overcome the convergence properties of the LS CG method and to improve the efficiency of \( d_k^{FTCGHS} \), we construct the following search direction based on DL and \( d_k^{FTCGHS} \) search directions as follows...
$$d_k^{FTCGLS} = -g_k + \left(-\frac{g_k^T y_{k-1}}{d_{k-1}^T g_{k-1}} - t_k \frac{g_k^T s_{k-1}}{d_{k-1}^T g_{k-1}}\right) d_{k-1} + \left(\frac{g_k^T d_{k-1}}{d_{k-1}^T g_{k-1}}\right) (y_{k-1} - s_{k-1}), \quad (10)$$

where $t_k = \frac{\|s_k\|}{\|y_{k-1}\|}$. In following sections, we assume that:

$$\chi_k = \beta_k^{LS} - t_k \frac{g_k^T s_{k-1}}{d_{k-1}^T g_{k-1}} \quad \text{and} \quad \theta_k = \left(\frac{g_k^T d_{k-1}}{d_{k-1}^T g_{k-1}}\right).$$

If we use exact line search, the equation can be reduced to the original LS. Since

$$g_k^T d_{k-1} = 0,$$

and

$$g_k^T s_{k-1} = \alpha_k g_k^T d_{k-1},$$

we obtain

$$d_k^{FTCGHS} = -g_k + \left(\beta_k^{LS}\right) d_{k-1}.$$

Algorithm 1 describes the steps of CG method to obtain the stationary point using SWP line search and equation (12) with stopping criteria $\|g_k\| \leq 10^{-6}$.

**Algorithm 1**

**Step 1.** Set a starting point $x_1$. This initial point can be arbitrary or standard for scientific functions. The initial search direction is the negative gradient, i.e. $d_1 = -g_1$.

Let $k := 1$.

**Step 2.** If the stopping condition is satisfied, then stop.

**Step 3.** Compute the search direction $d_k$ based on equation (2) using equation (10).

**Step 4.** Compute the step size $\alpha_k$ using equations (4) and (5).

**Step 5.** Update $x_k+1$ based on equation (2).

**Step 6.** Set $k := k + 1$ and go to Step 2.

4. Convergence analysis of the CG method with Algorithm 1

The descent condition (downhill condition) is given by the following equation

$$g_k^T d_k < 0, \forall k \geq 1, \quad (11)$$

which is useful in the study of the CG method and serves important rule in the proof of convergence analysis. Al-baali [5] modified equation (11) to the following form and used it to prove the FR method given by

$$g_k^T d_k \leq -c \|g_k\|^2, \forall k \geq 1, \quad (12)$$

where $c \in (0, 1)$. 
4.1. The descent property of the new search direction

In the next theorem, we show that the search direction in equation (10) ensures that the sufficient descent condition (12) is satisfied with the SWP line search.

**Theorem 4.1** Let the sequences \{x_k\} and \{d_k\} be generated using equations in Algorithm 1, where \(\alpha_k\) is computed using SWP line search. Then, the sufficient descent condition holds.

**Proof.** Multiply (12) by \(g_k^T\), which yields

\[
g_k^T d_k = -\|g_k\|^2 + \left(-\frac{g_k^T g y_{k-1}}{d_{k-1}^T g_{k-1}} - t \frac{g_k^T s_{k-1}}{d_{k-1}^T g_{k-1}}\right) g_k^T d_{k-1} + \left(\frac{g_k^T d_{k-1}}{d_{k-1}^T g_{k-1}}\right) g_k^T (y_{k-1} - s_{k-1}),
\]

\[
= -\|g_k\|^2 + \left(-\frac{g_k^T g y_{k-1}}{d_{k-1}^T g_{k-1}}\right) g_k^T d_{k-1} - t \frac{g_k^T s_{k-1}}{d_{k-1}^T g_{k-1}} g_k^T d_{k-1} + \left(\frac{g_k^T d_{k-1}}{d_{k-1}^T g_{k-1}}\right) g_k^T (y_{k-1} - \frac{g_k^T d_{k-1}}{d_{k-1}^T g_{k-1}}) s_{k-1},
\]

\[
= -\|g_k\|^2 - t \alpha_k \left(\frac{g_k^T d_{k-1}}{d_{k-1}^T g_{k-1}}\right) g_k^T d_{k-1} - \alpha_k \left(\frac{g_k^T d_{k-1}}{d_{k-1}^T g_{k-1}}\right) g_k^T d_{k-1},
\]

\[
= -\|g_k\|^2 - \alpha_k \left(\frac{\|g_k^T d_{k-1}\|^2}{d_{k-1}^T g_{k-1}}\right) (t + 1).
\]

From the SWP line search, we obtain

\[
d_{k-1}^T y_{k-1} > 0.
\]

Thus, \(g_k^T d_k \leq -\|g_k\|^2\). The proof is complete. ■

For example, the Zoutendijk condition [33] presented a useful Lemma for analyzing the convergence property of the CG method. The Lemma is given as follows:

**Lemma 4.1** Let Assumption 1 holds. Consider any CG method in the form (2) and (5), where \(\alpha_k\) satisfies the WWP line search, in which the search direction satisfies the descent condition. Then, the following condition holds:

\[
\sum_{k=0}^{\infty} \frac{(g_k^T d_k)^2}{\|d_k\|^2} < \infty.
\]

### 4.2. Convergence of Algorithm 1 with convex functions

Dai and Liao [9] proposed useful theorems (Theorem 3.2, Theorem 3.3, and Corollary 3.1) for convergence analysis of the CG method as follows:
Theorem 4.2. Suppose Assumption 1 holds. Consider any method in the form (2) and (5), where the search directed is descent with the step length obtained by SWP line search. Then, either

\[ \liminf_{k \to \infty} \| g_k \| = 0, \]

or

\[ \sum_{k=0}^{\infty} \frac{\| g_k \|^4}{\| d_k \|^2} < \infty. \quad (15) \]

Theorem 4.3. Suppose Assumption 1 holds. Consider any method in the form (2) and (5), where the search directed is descent and the step length is obtained by SWP line search. If

\[ \sum_{k=0}^{\infty} \frac{\| g_k \|^t}{\| d_k \|^2} = +\infty. \quad (16) \]

Then, for any \( t \in [0, 4] \), the method converges in the sense that \( \liminf_{k \to \infty} \| g_k \| = 0 \).

**Proof.** Suppose that

\[ \liminf_{k \to \infty} \| g_k \| \neq 0. \]

Then, from Theorem 4.2, we obtain

\[ \sum_{k=0}^{\infty} \frac{\| g_k \|^4}{\| d_k \|^2} < \infty, \quad (17) \]

because \( \| g_k \| \) is bounded away from zero and \( t \in [0, 4] \). It is easy to see that (17) contradicts (16). Thus, the theorem is true and the proof is complete. \( \blacksquare \)

Corollary 4.1. Suppose that Assumption 1 holds. Consider any conjugate gradient method in the form of equations (2) and (5), where \( d_k \) is a descent direction and \( \alpha_k \) is obtained by the strong Wolfe line search. If

\[ \sum_{k \geq 1} \frac{1}{\| d_k \|^2} = \infty, \quad (18) \]

then

\[ \liminf_{k \to \infty} \| g_k \| = 0. \]

**Proof.** From Theorem 4.3 and using \( t = 0 \), we obtain \( \liminf_{k \to \infty} \| g_k \| = 0. \)

The following theorem shows that the new search direction satisfies the convergences analysis with convex functions.
Theorem 4.4. Suppose that Assumption 1 holds. Consider the CG method in the forms of equations (2) and (10), and $d_k$ as a descent direction by using Theorem 3.1, where $\alpha_k$ is obtained using strong Wolfe-Powell line search. It $f(x)$ is a uniformly convex function, then $\liminf_{k \to \infty} \|g_k\| = 0$.

Proof. Because the function $f(x)$ is uniformly convex, there exists a positive constant $\varpi$ such that
$$\varpi \|x - y\|^2 \leq (\nabla f(x) - \nabla f(y))^T (x - y).$$

For all $x, y \in \Psi$, we have
$$d_{k-1}y_{k-1} \geq \varpi \alpha_{k-1} \|d_{k-1}\|^2. \quad (19)$$

Using equation (14) and triangular inequality, we get
$$\|d_k\| \leq \|g_k\| + \frac{(\sigma + 1) \|g_k y_{k-1}\|}{\varpi \alpha_{k-1} \|d_{k-1}\|^2} + t_k \frac{(\sigma + 1) \|g_k s_{k-1}\|}{\varpi \alpha_{k-1} \|d_{k-1}\|^2} \|d_{k-1}\| + \frac{\|g_k^T d_{k-1}\|}{\varpi \alpha_{k-1} \|d_{k-1}\|^2} \left(\|y_{k-1}\| + \|s_{k-1}\|\right).$$

By using equation (18), we have
$$\|d_k\| \leq \|g_k\| + t_k \frac{(\sigma + 1) \|g_k s_{k-1}\|}{\varpi \alpha_{k-1} \|d_{k-1}\|^2} \|d_{k-1}\| + \frac{\|g_k^T d_{k-1}\|}{\varpi \alpha_{k-1} \|d_{k-1}\|^2} \left(\|y_{k-1}\| + \|s_{k-1}\|\right).$$

Also, using triangular inequality and Assumption 1 yields
$$\|d_k\| \leq \|g_k\| + \frac{L \alpha_{k-1}(\sigma+1) \|g_k\| \|d_{k-1}\|^2}{\varpi \alpha_{k-1} \|d_{k-1}\|^2} + t_k \frac{\alpha_{k-1}(\sigma+1) \|g_k\| \|d_{k-1}\|^2}{\varpi \alpha_{k-1} \|d_{k-1}\|^2} + \frac{(\sigma+1) \|g_k\| \|d_{k-1}\|^2}{\varpi \alpha_{k-1} \|d_{k-1}\|^2} \left(L \alpha_{k-1} + \alpha_{k-1}\right),$$

$$\leq \|g_k\| + (\sigma + 1) \left(\frac{L \|g_k\|}{\varpi} + t_k \frac{\|g_k\|}{\varpi}\right) + (\sigma + 1) \left(\frac{\|g_k\|}{\varpi}\right) (L + 1),$$

$$\leq \|g_k\| + (\sigma + 1) \frac{\|g_k\|}{\varpi} (2L + t_k + 1).$$

By using Assumption 1, we obtain
$$\|d_k\| \leq B + \frac{B}{\varpi} (\sigma + 1)(2L + t_k + 1).$$

Let
$$B + \frac{B}{\varpi} (2L + t_k + 1)(\sigma + 1) = M,$$

where $M$ is constant; thus
$$\|d_k\| \leq M,$$
which implies the truth of equation (18). Thus, using Corollary 3.1, we have
\[ \liminf_{k \to \infty} \|g_k\| = 0. \]
The proof is now complete. ■

4.3. Convergence of Algorithm 1 with general nonlinear functions

The following restriction for \( \chi_k \) is essential to establish the convergence analysis for the new search direction. The main importance of this restriction is to avoid the CG method multiplayer being non-negative
\[ \chi_k^+ = \max \left\{ 0, \beta_k \frac{g_k^T s_{k-1}}{d_{k-1}^T y_{k-1}} \right\}. \]
Thus, equation (10) becomes as follows
\[ d_k = -g_k + \chi_k^+ d_{k-1} + \theta_k (y_{k-1} - s_{k-1}). \]

**Lemma 4.2.** Assume that Assumption 1 holds and the sequences \( \{g_k\} \) and \( \{d_k\} \) are generated using Algorithm 1, where the step size \( \alpha_k \) is computed via the SWP line search such that the sufficient descent condition holds. If \( \beta_k \geq 0 \), there exists a constant \( \gamma > 0 \) such that \( \|g_k\| > \gamma \) for all \( k \geq 1 \). Then, \( d_k \neq 0 \) and
\[ \sum_{k=0}^{\infty} \|u_{k+1} - u_k\|^2 < \infty, \tag{20} \]
where \( u_k = \frac{d_k}{\|d_k\|} \).

**Proof.** First, if \( d_k = 0 \), then from the sufficient descent condition, we obtain \( g_k = 0 \). Thus, we suppose that \( d_k \neq 0 \) and
\[ \bar{\gamma} \geq \|g_k\| \geq \gamma > 0, \forall k \geq 1. \tag{21} \]
We now rewrite equation (10) as follows:
\[ d_k^{FTCGLS} = -g_k + \left( -g_k^T y_{k-1} \frac{d_{k-1}^T g_k}{d_{k-1}^T y_{k-1}} \right) d_{k-1} - \left( \frac{g_k^T d_{k-1}}{d_{k-1}^T y_{k-1}} \right) (y_{k-1} - s_{k-1}). \]
We define
\[ u_k = w_k + \delta_k u_{k-1}, \]
where
\[ w_k = -g_k + \theta_k (y_{k-1} - s_{k-1}) \frac{\|d_{k-1}\|}{\|d_k\|}, \delta_k = \chi_k^+ \frac{\|d_{k-1}\|}{\|d_k\|}. \]
Since $u_k$ is a unit vector, then
\[ \| w_k \| = \| u_k - \delta_k u_{k-1} \| = \| \delta_k u_k - u_{k-1} \|. \]

Using the triangular inequality and $\delta_k \geq 0$,
\[ \| u_k - u_{k-1} \| \leq (1 + \delta_k) \| u_k - u_{k-1} \| = \| u_k - \delta_k u_{k-1} - (u_{k-1} - \delta_k u_k) \|, \] (22)
\[ \leq \| u_k - \delta_k u_{k-1} \| + \| u_{k-1} - \delta_k u_k \| = 2 \| w_k \|. \]

We now define
\[ \nu = -g_k - \theta_k (y_k - 1 + s_{k-1}). \]

Using the triangular inequality, we obtain
\[ \| \nu \| \leq \| g_k \| + |\theta_k| \| y_k - 1 + s_{k-1} \|. \]

Moreover, using the equations of SWP line search, we can conclude that
\[ |\theta_k| = \left| \frac{g_k^T d_{k-1}}{d_{k-1}^T g_{k-1}} \right| \leq \sigma. \]

Now, using the triangular inequality and Assumption 1, we obtain
\[ \| y_{k-1} + s_{k-1} \| \leq \| y_{k-1} \| + \| s_{k-1} \| \leq 2B + 2\rho. \]

Thus, the inequality in (4.3) can be written as follows:
\[ \| \nu \| \leq B + \sigma (2B + 2\rho). \]

Let
\[ H = B + \sigma (2B + 2\rho), \]
then
\[ \| \nu \| \leq H. \]

From equation (22), we have
\[ \| u_k - u_{k-1} \| \leq 2w. \]
Thus, the following result is obtained
\[
\sum_{k=0}^{\infty} \|u_{k+1} - u_k\|^2 \leq 4 \sum_{k=0}^{\infty} \|w\|^2 \leq 4H^2 \sum_{k=0}^{\infty} \|d_k\|^2 < \infty.
\]

The proof is now complete. ■

The following property, which is referred to as Property*, was presented by Gilbert and Nocedal in [15].

**Property**
Consider a method of the form (2) and (3). Assume that (21) is satisfied for all \(k \geq 1\). Then, the CG method has Property* if there exist constants \(b > 1\) and \(\lambda > 0\) such that for all \(k \geq 1\),
\[
|\chi_k| \leq b \quad \text{and if } \|s_k\| \leq \lambda,
\]
we obtain
\[
|\eta_k| \leq \frac{1}{2b}.
\]

**Lemma 4.3** Consider the CG method of the form (1) and (2) with \(\chi_k^+\), where the step size satisfies SWP line search (4) and (5). If equation (21) holds true, then \(\chi_k^+\) possesses Property*. Namely, suppose (21) holds, then there exist \(b > 1\) and \(\lambda > 0\) for all \(k \geq 1\) whereby \(|\chi_k^+| \leq b\) and if \(\|s_k\| \leq \lambda\), we obtain
\[
|\chi_k| \leq \frac{1}{2b}.
\]

**Proof.** As a result set \(b = \frac{2(L+t)\bar{\gamma}B}{\gamma^2} \geq 1\), and \(\lambda = \frac{\gamma^2}{2b(L+t)\bar{\gamma}}\).

Using SWP (4) and (5) with equation (21), we obtain
\[
|\chi_k^+| \leq \frac{g_k^T y_{k-1}}{d_k^T g_{k-1}} + t \frac{g_k^T s_{k-1}}{d_k^T g_{k-1}} \leq \frac{L \|g_k\| \|s_{k-1}\| + t \|g_k\| \|s_{k-1}\|}{\gamma^2} \leq \frac{2(L+t)\bar{\gamma}B}{\gamma^2} = b > 1,
\]
and if \(\|s_k\| \leq \lambda\),
\[
|\chi_k^+| \leq \frac{g_k^T y_{k-1}}{d_k^T g_{k-1}} + t \frac{g_k^T s_{k-1}}{d_k^T g_{k-1}} \leq \frac{L \|g_k\| \|s_{k-1}\| + t \|g_k\| \|s_{k-1}\|}{\gamma^2} \leq \frac{(L+t)\bar{\gamma}\lambda}{\gamma^2},
\]
which implies
\[
|\chi_k^+| \leq \frac{1}{2b} \quad (23)
\]

The proof is complete. ■

The following Lemma and theorem are similar to that presented by [15]. Here, we present Lemma 4.4 without its proof, which can be referred to in [15].
**Lemma 4.4.** Assume that Assumption 1 holds. Also, assume that the sequences \( \{g_k\} \) and \( \{d_k\} \) are generated by Algorithm 1, in which \( \alpha_k \) is computed by the WWP line search where the sufficient descent condition holds, assuming that the method has Property\(^*\). Suppose also \( \|g_k\| \geq \gamma \) for some \( \lambda > 0 \). Then, there exists \( \lambda > 0 \) such that for any \( \Delta \in \mathbb{N} \) and any index \( k_0 \), there is an index \( k > k_0 \) exists that satisfies

\[
\kappa_{k,\Delta}^\lambda > \frac{\lambda}{2},
\]

where \( \kappa_{k,\Delta}^\lambda = \{ i \in \mathbb{N} : k \leq i \leq k + \Delta - 1, \|s_i\| > \lambda \} \), \( \mathbb{N} \) denotes the set of positive integers and \( |\kappa_{k,\Delta}^\lambda| \) denotes the number of elements in \( \kappa_{k,\Delta}^\lambda \).

**Theorem 4.4** Suppose that Assumption 1 holds. Assume also that the sequences \( \{g_k\} \) and \( \{d_k\} \) are generated by Algorithm 1 in which \( \alpha_k \) is computed by the WWP line search and the sufficient descent condition holds. Also, suppose the Property\(^*\) holds. Then, we have \( \lim_{k \to \infty} \inf \|g_k\| = 0 \).

**Proof.** Based on Lemma 4.2 and Lemma 4.4, the proof is done by contradiction. We define \( u_i := \frac{d_i}{\|d_i\|} \). For any two indexes \( l, k \) with \( l \geq k \), we have

\[
x_l - x_{k-1} = \sum_{i=k}^{l} \|s_{i-1}\| u_{i-1} = \sum_{i=k}^{l} \|s_{i-1}\| u_{k-1} + \sum_{i=k}^{l} \|s_{i-1}\| (u_{i-1} - u_{k-1}),
\]

where \( s_{i-1} = x_i - x_{i-1} \).

Taking the norms, we obtain

\[
\sum_{i=k}^{l} \|s_{i-1}\| \leq \|x_l\| + \|x_{k-1}\| + \sum_{i=k}^{l} \|s_{i-1}\| \|u_{i-1} - u_{k-1}\|.
\]

Using Assumption 1, we have that the sequence \( \{x_k\} \) is bounded, and there exists a positive constant \( \eta \) such that \( \|x_k\| \leq \eta \), for all \( k \geq 1 \). Thus,

\[
\|x_l\| + \|x_{k-1}\| \leq 2\eta,
\]

which implies that

\[
\sum_{i=k}^{l} \|s_{i-1}\| \leq 2\eta + \sum_{i=k}^{l} \|s_{i-1}\| \|u_{i-1} - u_{k-1}\|. \tag{24}
\]

Assume \( \lambda > 0 \) given in Lemma 4.4, following the notation of this Lemma, we define

\[
\Delta := \left\lceil \frac{8\eta}{\lambda} \right\rceil.
\]

By Lemma 4.2, we can find an index \( k_0 \) such that
\[
\sum_{k \geq k_0} \| u_i - u_{i-1} \|^2 < \frac{1}{4\Delta}. \tag{25}\]

With this \( \Delta \) and \( k_0 \), Lemma 4.4 gives an index \( k \geq k_0 \) such that

\[|\kappa_{k,\Delta}^\lambda| > \frac{\Delta}{2}. \tag{26}\]

Next, by the Cauchy-Schwarz inequality and (25), we have, for any index for \( i \in [k, k + \Delta - 1] \) such that

\[
\| u_{i-1} - u_{k-1} \| \leq \sum_{j=k}^{i-1} \| u_j - u_{j-1} \|,
\]

\[
\leq (i - k)^{1/2} \left( \sum_{j=k}^{i-1} \| u_j - u_{j-1} \|^2 \right)^{1/2},
\]

\[
\leq \Delta^{1/2} \left( \frac{1}{4\Delta} \right)^{1/2} = \frac{1}{2}.
\]

By this relation, (24) and (26), with \( l = k + \Delta - 1 \), we have

\[
2\eta \geq \frac{1}{2} \sum_{i=k}^{k+\Delta-1} \| s_{i-1} \| > \frac{\lambda}{2} |\kappa_{k,\Delta}^\lambda| > \frac{\lambda\Delta}{4}.
\]

Thus, \( \Delta < 8\eta/\lambda \), which contradicts the definition of \( \Delta \). The proof is then complete. \( \blacksquare \)

5. Numerical results

To test the efficiency of the new search direction (12), we selected some test functions in Appendix 1 from CUTEr [21]. A comparison made with strong CG coefficients is such as CG-Descent 5.3 [17], DL+ [10], and the search direction \( d_k^{FTCGHS} \) [1]. Since \( \beta_{DL}^k \) is not non-negative in general, we use \( \beta_{DL}^{k+} \) similar to [24] as follows

\[
\beta_{DL}^{k+} = \max\{\beta_{HS}^k, 0\} - t \frac{g_k^T s_{k-1}}{d_{k-1}^T y_{k-1}}. \tag{27}\]

Moreover, the authors in [10] restate equation (27) using the steepest descent if it does not satisfy the descent condition. The comparison was made based on the CPU time, number of iterations, number of function evaluations, and number of gradient evaluations. We use SWP line search for \( d_k^{FTCGHS} \) and DL+ method with \( \delta = 0.01 \) and \( \sigma = 0.1 \) similar to that used by the authors. For CG-Descent, we also used the Approximate Wolfe-Powell line search similar to that used by [17]. Moreover, we used SWP line search for \( d_k^{FTCGLS} \) similar to that used by [8] as follows:

\[
f(x_k + \alpha_k d_k) \leq f(x_k) + \delta \alpha_k g_k^T d_k,
\]

\[
\sigma_1 g_k^T d_k \leq g(x_k + \alpha_k d_k)^T d_k \leq \sigma_2 g_k^T d_k,
\]
\[ 0 < \delta < \sigma_1 \leq \sigma_2 < 1, \]

where \( \delta = 0.0001, \sigma_1 = 0.1, \) and \( \sigma_2 = 0.4. \)

The results of FTCGHS, FTCGLS, and DL+CG methods are obtained by running the modified code of CG-Descent. The code can be obtained from the Hager webpage:

https://people.clas.ufl.edu/hager/software/

The norm of the gradient was employed as the stopping criterion, specifically \( \|g_k\| \leq 10^{-6} \) for all methods. The host computer is AMD A4-7210 APU Radeon R3 Graphics, where the installed memory is 4 GB with operating system Ubuntu 20.04.2.0 LTS. The results are shown in Figures 1, 2, and 3, in which a performance measure introduced by Dolan and More [12] was employed. This performance measure was introduced to compare a set of solvers \( S \) on a set of problems \( F \). Assuming \( n_s \) solvers and \( n_f \) problems in \( S \) and \( F \), respectively, the measure \( t_{f,s} \) is defined as the number of iterations or the CPU time required to solve the problem \( f \) by the solver \( s \). To create a baseline for comparison, the performance of the solver \( s \) on a problem \( f \) is scaled by the best performance of any solver in \( S \) on the problem using the ratio

\[ r_{f,s} = \frac{t_{f,s}}{\min\{t_{f,s} : s \in S\}}. \]

Suppose that a parameter \( r_M \geq r_{f,s} \) for all \( f, s \) is chosen. Then, \( r_{f,s} = r_M \) if and only if the solver \( s \) does not solve a problem \( f \). Because we would like to obtain an overall assessment of the performance of a solver, we defined the measure as

\[ P_s(t) = \frac{1}{n_f \text{ size}\{f \in F : \log r_{f,s} \leq t\}}. \]

Thus, \( P_s(t) \) is the probability for a solver \( s \in S \) that the performance ratio \( r_{f,s} \) is within a factor \( t \in \mathbb{R} \) of the best possible ratio. Suppose we define the function \( p_s \) as the cumulative distribution function for the performance ratio, then the performance measure \( f_s : \mathbb{R} \to [0, 1] \) for a solver is non-decreasing and piecewise continuous from the right. Thus, the value \( f_s(1) \) is the probability that the solver has the best performance of all the solvers. In general, a solver with high values of \( f(t) \), which would appear in the upper right corner of the figure, is preferable.

5.1. Result analysis

In Appendix 1, the following big notations denote:
A: number of iterations.
B: number of function evaluations.
C: number of gradient evaluations.
D: CPU time.

Figure 1 shows that CG-Descent outperforms DL+ in terms of the number of iterations. On the other hand, FTCGLS and FTCGHS outperform CG-Descent and DL+. Thus,
we can conclude that using the four-term CG method is better than using three-term CG methods. In addition, we can note that FTCGLS slightly outperforms FTCGHS in the number of iteration. From Figure 2, we can note that FTCGLS strongly outperforms all methods in terms of the number of function evaluations. Thus, we conclude that using extended SWP line search is better than using SWP line search. Figure 3 shows that FTCGLS outperform FTCGHS, DL+, and CG-Descent in terms of CPU time. From all figures, we can note that using four-term is better than using three-terms. Moreover, using an extended SWP line search is better than using the original SWP line search since the latter reduces the number of function evaluations. Finally, we can conclude that FTCGLS is better than FTCGHS, CG-Descent, and DL+ in terms of efficiency.

Figure 1: Performance measure based on the number of iterations.
Figure 2: Performance measure based on the function evaluation.

Figure 3: Performance measure based on the CPU time.

Figure 4 presents the Diagonal 4 function in 3D. This function has a long narrow valley with steep walls on both sides. Note that with dimension 2, the minimum is $x^* = (0, 0)$, while the function value is $f(x^*) = 0$. 
6. Conclusion

In this paper, we propose a new four-term CG method based on the LS CG method. The new search direction satisfies the following properties:

(i) Equation (10) satisfies the descent property.

(ii) Equation (10) satisfies the convergence properties, i.e., the stationary point can be obtained by the CG method with equation (10) for any unconstrained optimization function.

(iii) Equation (10) contains a four-term CG method.

(iv) Equation (10) outperforms CG-Descent, DL+, and FTCGHs in the number of iterations, function evaluations, and CPU time.

The future work will focus on the application of the CG methods in many fields such as machine learning, deep learning, and regression problems.

Acknowledgements

The authors would like to thank the reviewers and editor for the comments that highly improved this paper. Also, we would like to thank the University Sains Islam Malaysia (USIM) for the support of this paper via grant P1-18-12220-UNI-RCR-FST. Finally, the authors would like to thank Prof. Dr. William Hager for publishing the code of CG-Descent.

References

[1] I. Masmali A. Alhawarat, G. Alhamzi and Z. Salleh. A Decent Four Term Conjugate Gradient Method with Global Convergence Properties for Large Scale Unconstrained Optimization Problems. Mathematical Problems in Engineering, 2021.

[2] M. Mamat A. Alhawarat, Z. Salleh and M. Rivaie. An efficient modified Polak–Ribière–Polyak conjugate gradient method with global convergence properties. Optimization Methods and Software, 32(6):1299–1312, 2017.
[3] Z. Salleh A. Alhawarat and I. A. Masmali. A Convex Combination between Two Different Search Directions of Conjugate Gradient Method and Application in Image Restoration. *Mathematical Problems in Engineering*, 2021.

[4] Z. Salleh A. Alhawarat and I. A. Masmali. A Convex Combination between Two Different Search Directions of Conjugate Gradient Method and Application in Image Restoration. *Mathematical Problems in Engineering*, 2021.

[5] M. Al-Baali. Descent property and global convergence of the Fletcher-Reeves method with inexact line search. *IMA J. Numer. Anal.*, 5:121–124, 1985.

[6] A. Alhawarat and Z. Salleh. Modification of nonlinear conjugate gradient method with weak Wolfe-Powell line search. *In Abstract and Applied Analysis*, Hindawi, 2017.

[7] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, United Kingdom, 2004.

[8] E. K. P. Chong and S. H. Zak. *An introduction to optimization*. Wiley and Sons, New York, 2nd ed, 2005.

[9] Y. Dai, D. Sun J. Han G. Liu, H. Yin, and Y. X. Yuan. Convergence properties of nonlinear conjugate gradient methods. *SIAM Journal on Optimization*, 10(2):345–358, 2000.

[10] Y. H. Dai and L. Z. Liao. New conjugacy conditions and related nonlinear conjugate gradient methods. *Appl. Math. Optim.*, 43(1):87–101, 2001.

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

[12] E.D. Dolan and J.J. Moré. Benchmarking optimization software with performance profiles. *Math. Program.*, 91:201–213, 2002.

[13] R. Fletcher. *Practical methods of optimization*. John Wiley & Sons, 2013.

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

[15] J. C. Gilbert and J. Nocedal. Global convergence properties of conjugate gradient methods for optimization. *Siam. J. Optim.*, 2:21–42, 1992.

[16] I. C. Bivens H. Anton and S. Davis. *Calculus: Early Transcendentals*. John Wiley, 2010.

[17] W. W Hager and H. Zhang. A new conjugate gradient method with guaranteed descent and an efficient line search. *SIAM Journal on optimization*, 16(1):170–192, 2005.
[18] W. W Hager and H. Zhang. The limited memory conjugate gradient method. *SIAM Journal on Optimization*, 23(4):2150–2168, 2013.

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

[20] Z. Salleh I. A. Masmali and A. Alhawarat. A decent three-term conjugate gradient method with global convergence properties for large scale unconstrained optimization problems. *AIMS Mathematics*, 6(10):10742–10764, 2021.

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

[22] L.Q. Zhang J. K. Liu, J.L. Xu. Partially symmetrical derivative-free Liu-Storey projection method for convex constrained equations. *Int. J. Comput. Math*, 96:1787–1798, 2019.

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

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

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

[26] M. J. Powell. Non-convex minimization calculations and the conjugate gradient method. In *Numerical analysis*, Springer, Berlin, Heidelberg:122–141, 1984.

[27] H. Tu S. Yao, L. Ning and J. Xu. A one-parameter class of three-term conjugate gradient methods with an adaptive parameter choice. *Optimization Methods and Software*, 35(6):1051–1064, 2020.

[28] J. A. Snyman. *Practical mathematical optimization.* Springer Science and Business Media, New York, 2005.

[29] W. Sun and Y. X. Yuan. Optimization theory and methods: nonlinear programming. *Springer Science & Business Media*, 1, 2006.

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

[31] P. Wolfe. Convergence conditions for ascent methods. II: Some corrections. *SIAM review*, 13(2):185–188, 1971.
[32] I. Masmali Z. Salleh, G. Alhamzi and A. Alhawarat. A Modified Liu and Storey Conjugate Gradient Method for Large Scale Unconstrained Optimization Problems. *Algorithms*, 14(8):227, 2021.

[33] G. Zoutendijk. Nonlinear Programming, Computational Methods. *Integer Nonlinear Program*, 143:37–86, 1970.
## Appendix 1

| Function   | FTCGLS A | FTCGLS B | FTCGLS C | FTCGLS D | FTCGHS A | FTCGHS B | FTCGHS C | FTCGHS D | CG-DESCENT A | CG-DESCENT B | CG-DESCENT C | CG-DESCENT D | DL+ A | DL+ B | DL+ C | DL+ D |
|------------|----------|----------|----------|----------|----------|----------|----------|----------|--------------|--------------|--------------|--------------|--------|--------|--------|--------|
| AKIVA      | 9        | 23       | 17       | 0.02     | 8        | 20       | 15       | 0.02     | 10           | 21           | 11           | 0.02         | 8      | 20     | 15     | 0.02   |
| ALLINTU    | 9        | 23       | 16       | 0.02     | 9        | 25       | 18       | 0.02     | 12           | 29           | 18           | 0.02         | 9      | 25     | 18     | 0.03   |
| ARGLINB    | 4        | 71       | 70       | 0.06     | 2        | 101      | 101      | 0.06     | 5            | 13           | 13           | 0.02         | 5      | 73     | 72     | 0.09   |
| ARGLINC    | 3        | 33       | 32       | 0.05     | 2        | 98       | 98       | 0.08     | 11           | 106          | 110          | 0.02         | 5      | 79     | 78     | 0.06   |
| BARD       | 11       | 29       | 21       | 0.02     | 12       | 32       | 22       | 0.02     | 16           | 33           | 17           | 0.02         | 12     | 32     | 22     | 0.02   |
| BDEXP      | 2        | 8        | 8        | 0.02     | 2        | 7        | 7        | 0.02     | 5            | 11           | 6            | 0.02         | 2      | 7      | 7      | 0.02   |
| BDQRTIC    | 113      | 236      | 203      | 0.52     | 154      | 310      | 302      | 0.65     | 136          | 273          | 237          | 0.52         | 168    | 363    | 359    | 0.63   |
| BEALE      | 13       | 34       | 24       | 0.02     | 11       | 33       | 26       | 0.02     | 15           | 31           | 16           | 0.02         | 11     | 33     | 26     | 0.02   |
| BIGGS3     | 80       | 190      | 124      | 0.02     | 79       | 207      | 144      | 0.02     | 110          | 231          | 125          | 0.02         | 79     | 207    | 144    | 0.02   |
| BIGGS5     | 80       | 190      | 124      | 0.02     | 79       | 207      | 144      | 0.02     | 110          | 231          | 125          | 0.02         | 79     | 207    | 144    | 0.02   |
| BIGGS6     | 23       | 57       | 39       | 0.02     | 24       | 64       | 44       | 0.02     | 27           | 57           | 31           | 0.02         | 24     | 64     | 44     | 0.02   |
| BOX2       | 9        | 21       | 13       | 0.02     | 10       | 23       | 14       | 0.02     | 11           | 24           | 13           | 0.02         | 10     | 23     | 14     | 0.02   |
| BOX3       | 9        | 21       | 13       | 0.02     | 10       | 23       | 14       | 0.02     | 11           | 24           | 13           | 0.02         | 10     | 23     | 14     | 0.02   |
| BRKMCC     | 5        | 11       | 6        | 0.02     | 5        | 11       | 6        | 0.02     | 5            | 11           | 6            | 0.02         | 5      | 11     | 6      | 0.02   |
| BROWNAL    | 6        | 17       | 13       | 0.02     | 8        | 19       | 12       | 0.02     | 9            | 25           | 18           | 0.02         | 10     | 29     | 21     | 0.02   |
| BROWNBS    | 11       | 26       | 17       | 0.02     | 10       | 24       | 18       | 0.02     | 13           | 26           | 15           | 0.02         | 10     | 24     | 18     | 0.02   |
| BROWNDEN   | 16       | 36       | 26       | 0.02     | 16       | 38       | 31       | 0.02     | 16           | 31           | 19           | 0.02         | 16     | 38     | 31     | 0.02   |
| BROYDN7D   | 59       | 108      | 81       | 0.37     | 54       | 100      | 76       | 0.26     | 1411         | 2810         | 1429         | 5.22         | 75     | 138    | 112    | 0.36   |
| BRYBND     | 35       | 98       | 72       | 0.22     | 32       | 86       | 62       | 0.15     | 85           | 174          | 90           | 0.28         | 149    | 317    | 174    | 0.55   |
| CAMEL6     | 7        | 27       | 22       | 0.02     | 6        | 22       | 18       | 0.02     | 13           | 34           | 22           | 0.02         | 6      | 22     | 18     | 0.02   |
| CHNRASNB   | 267      | 523      | 310      | 0.02     | 299      | 590      | 343      | 0.02     | 287          | 564          | 299          | 0.02         | 1009   | 1998   | 1180   | 0.01   |
| CLIFF      | 6        | 45       | 37       | 0.02     | 10       | 46       | 39       | 0.02     | 18           | 70           | 54           | 0.02         | 10     | 46     | 39     | 0.01   |
| CUBE       | 15       | 46       | 38       | 0.02     | 17       | 48       | 34       | 0.02     | 32           | 77           | 47           | 0.02         | 17     | 48     | 34     | 0.02   |
| DENSCHNA   | 6        | 16       | 12       | 0.02     | 6        | 16       | 12       | 0.02     | 9            | 19           | 10           | 0.02         | 6      | 16     | 12     | 0.02   |
| DENSCHNB   | 6        | 18       | 15       | 0.02     | 6        | 18       | 15       | 0.02     | 7            | 15           | 8            | 0.02         | 6      | 18     | 15     | 0.02   |
| DENSCHNC   | 15       | 44       | 36       | 0.02     | 11       | 36       | 31       | 0.02     | 12           | 26           | 14           | 0.02         | 11     | 36     | 31     | 0.02   |
| DENSCHND   | 15       | 50       | 44       | 0.02     | 14       | 46       | 40       | 0.02     | 47           | 98           | 51           | 0.02         | 14     | 46     | 40     | 0.02   |
| DENSCHNE   | 13       | 42       | 35       | 0.02     | 12       | 43       | 38       | 0.02     | 18           | 49           | 32           | 0.02         | 12     | 43     | 38     | 0.02   |
|            | FTCGLS | FTCGHS | CG-DESCENT | DL+   |
|------------|--------|--------|------------|-------|
| DENSCNHF   | 10     | 30     | 24         | 0.02  |
| DIXMAANA   | 6      | 16     | 13         | 0.02  |
| DIXMAANB   | 7      | 16     | 10         | 0.02  |
| DIXMAANC   | 6      | 14     | 9          | 0.02  |
| DIXMAAND   | 8      | 20     | 14         | 0.02  |
| DIXMAANE   | 251    | 279    | 482        | 0.28  |
| DIXMAANE   | 10000  | 10007  | 19995      | 20.3  |
| DIXMAANF   | 152    | 311    | 163        | 0.13  |
| DIXMAANH   | 256    | 2917   | 3659       | 3.19  |
| DIXMAANI   | 10000  | 10007  | 19995      | 20.3  |
| DJTL       | 93     | 1469   | 1446       | 0.02  |
| DQDRTIC    | 5      | 11     | 6          | 0.02  |
| ECKERLE4LS| 3      | 7      | 4          | 0.02  |
| EDENSCH    | 26     | 59     | 50         | 0.05  |
| EGGRATCR   | 6      | 15     | 10         | 0.02  |
| EIGENALS   | 7318   | 12590  | 9382       | 0.37  |
| ENGVAL1    | 22     | 48     | 39         | 0.06  |
| ENGVAL2    | 25     | 69     | 52         | 0.02  |
| ENSOLS     | 21     | 45     | 27         | 0.02  |
| EXPFIT      | 10     | 29     | 21         | 0.02  |
| exp2       | 5      | 14     | 8          | 0.02  |
| FBRAINLS   | 10     | 29     | 23         | 0.03  |
| FBRAIN2LS  | 95     | 285    | 220        | 0.58  |
| FMINSRF2   | 304    | 628    | 332        | 1.09  |
| FMINSUF    | 452    | 925    | 477        | 1.59  |
| GENHUMPS   | 9395   | 19074  | 9741       | 0.37  |
| GROWTHLS   | 107    | 382    | 315        | 0.02  |
| GULF       | 27     | 90     | 70         | 0.02  |
| APPENDIX | FTCGLS | FTCGHS | CG-DESCENT | DL+ |
|----------|--------|--------|------------|-----|
| **HAHN1LS** | 4  | 54  | 56  | 53 | 0.02 | 37  | 121  | 86 | 0.02 | 5  | 56  | 53 | 0.02 |
| **HAIRY** | 16 | 63  | 51  | 0.02 | 17  | 82  | 68  | 0.02 | 36  | 99  | 65 | 0.02 | 17  | 82  | 68 | 0.02 |
| **HATFLDD** | 17 | 47  | 38  | 0.02 | 17  | 49  | 37  | 0.02 | 20  | 43  | 24 | 0.02 | 17  | 49  | 37 | 0.02 |
| **HATFLDE** | 12 | 33  | 25  | 0.02 | 13  | 37  | 30  | 0.02 | 30  | 72  | 45 | 0.02 | 13  | 37  | 30 | 0.02 |
| **HATFLDFL** | 39 | 123 | 100 | 0.02 | 21  | 68  | 54  | 0.02 | 39  | 92  | 55 | 0.02 | 21  | 68  | 54 | 0.02 |
| **HATFLDFLS** | 53 | 157 | 124 | 0.02 | 48  | 156 | 125 | 0.02 | 64  | 155 | 97 | 0.02 | 48  | 156 | 125 | 0.02 |
| **HEART6LS** | 376 | 1083 | 814 | 0.02 | 375 | 1137 | 876 | 0.02 | 684 | 1576 | 941 | 0.02 | 375 | 1137 | 876 | 0.02 |
| **HEART8LS** | 240 | 609 | 414 | 0.02 | 253 | 657 | 440 | 0.02 | 240 | 609 | 414 | 0.02 | 253 | 657 | 440 | 0.02 |
| **HELIX** | 37 | 123 | 100 | 0.02 | 21 | 68 | 54 | 0.02 | 39 | 92 | 55 | 0.02 | 21 | 68 | 54 | 0.02 |
| **HIELOW** | 13 | 30 | 22 | 0.05 | 13 | 30 | 21 | 0.03 | 23 | 49 | 27 | 0.02 | 23 | 49 | 27 | 0.02 |
| **HILBERTA** | 2 | 5 | 3 | 0.02 | 2 | 5 | 3 | 0.02 | 2 | 5 | 3 | 0.02 | 2 | 5 | 3 | 0.02 |
| **HILBERTB** | 4 | 9 | 5 | 0.02 | 4 | 9 | 5 | 0.02 | 4 | 9 | 5 | 0.02 | 4 | 9 | 5 | 0.02 |
| **HIMMELBB** | 7 | 22 | 17 | 0.02 | 7 | 22 | 17 | 0.02 | 8 | 20 | 13 | 0.02 | 7 | 22 | 17 | 0.02 |
| **HIMMELBF** | 23 | 56 | 40 | 0.02 | 23 | 59 | 46 | 0.02 | 26 | 60 | 36 | 0.02 | 23 | 59 | 46 | 0.02 |
| **HIMMELBG** | 5 | 13 | 9 | 0.02 | 5 | 13 | 9 | 0.02 | 7 | 16 | 9 | 0.02 | 5 | 13 | 9 | 0.02 |
| **HUMPS** | 45 | 223 | 202 | 0.02 | 52 | 186 | 146 | 0.02 | 45 | 223 | 202 | 0.02 | 45 | 223 | 202 | 0.02 |
| **HYDCAR6LS.SIF** | 70 | 143 | 74 | 0.02 | 120 | 242 | 123 | 0.02 | 1440 | 29028 | 14875 | 0.45 | 1001 | 2027 | 1174 | 0.03 |
| **INTEQNELS.SIF** | 6 | 13 | 7 | 0.02 | 6 | 13 | 7 | 0.02 | 6 | 13 | 7 | 0.02 | 6 | 13 | 7 | 0.02 |
| **JENSMP** | 15 | 54 | 45 | 0.02 | 12 | 47 | 41 | 0.02 | 15 | 33 | 22 | 0.02 | 12 | 47 | 41 | 0.02 |
| **JUDGE** | 9 | 24 | 17 | 0.02 | 9 | 24 | 18 | 0.02 | 10 | 23 | 13 | 0.02 | 9 | 24 | 18 | 0.02 |
| **LANCZOS1LS** | 73 | 184 | 129 | 0.02 | 61 | 177 | 135 | 0.02 | 148 | 325 | 181 | 0.02 | 61 | 177 | 135 | 0.02 |
| **LANCZOS2LS** | 70 | 175 | 119 | 0.02 | 60 | 169 | 125 | 0.02 | 169 | 379 | 215 | 0.02 | 60 | 169 | 125 | 0.02 |
| **LANCZOS3LS** | 70 | 177 | 125 | 0.02 | 61 | 164 | 118 | 0.02 | 179 | 392 | 219 | 0.02 | 61 | 164 | 118 | 0.02 |
| **LOGHAIRY** | 14 | 91 | 79 | 0.02 | 26 | 196 | 179 | 0.02 | 27 | 81 | 58 | 0.02 | 26 | 196 | 179 | 0.02 |
| **LSC1LS** | 34 | 106 | 85 | 0.02 | 31 | 108 | 89 | 0.02 | 36 | 101 | 71 | 0.02 | 31 | 108 | 89 | 0.02 |
| **LSC2LS** | 55 | 173 | 141 | 0.02 | 37 | 106 | 86 | 0.02 | 54 | 119 | 67 | 0.02 | 37 | 106 | 86 | 0.02 |
| **LUKSAN13LS** | 90 | 178 | 132 | 0.02 | 90 | 182 | 168 | 0.02 | 84 | 158 | 121 | 0.02 | 142 | 279 | 243 | 0.02 |
| **LUKSAN14LS** | 156 | 324 | 213 | 0.02 | 188 | 400 | 254 | 0.02 | 98 | 122 | 156 | 0.02 | 157 | 313 | 201 | 0.02 |
| FTCLGS     | FTCGHS    | CG-DESCENT  | DL+     |
|------------|-----------|-------------|---------|
| LUKSAN15LS | 27  61  45 | 28  59  44 | 27  60  45 |
| LUKSAN16LS | 28  56  38 | 31  57  38 | 35  72  53 |
| MANCINO    | 10  21  11 | 11  23  12 | 11  23  12 |
| MEXHAT     | 17  64  60 | 20  56  39 | 14  59  55 |
| MEYER3     | 145 611 534 | 19  76  63 | 19  76  63 |
| MGH09LS    | 49  143 110 | 25  82  72 | 25  82  72 |
| MGH10LS    | 1084 4530 5502 | 1134 4464 535 | 108 4052 4968 |
| MGH10SLS   | 73  470 387 | 146 505 401 | 19  112 102 |
| MGH17LS    | 58  199 150 | 228 564 363 | 84  323 365 |
| MISRA1BLS.SIF | 13  53  62 | 35  139 117 | 26  113 101 |
| MISRA1CLS.SIF | 24  82  94 | 26  110 91  | 26  145 121 |
| MISRA1DLS.SIF | 21  74  74 | 24  74  75  | 22  90  84 |
| MOREBY     | 161 168 317 | 161 168 317 | 117 124 229 |
| MSQRTBLS   | 2201 4410 2211 | 2280 4525 2326 | 5786 11558 5818 |
| NELSONLS   | 1117 3861 5389 | 1118 5692 7331 | 1101 5415 7690 |
| NONDIA     | 7   25  19 | 7   25  19  | 7   25  19 |
| OSBORNEA   | 67  174 122 | 94  213 124 | 82  230 174 |
| OSBORNEB   | 58  140 91 | 62  127 65  | 57  134 84 |
| OSCIPATH   | 292090 7E+05 5E+05 | 295029 534425 | 310999670953673251.91 |
| PALMER1C   | 11  26  26 | 11  26  26  | 12  27  28 |
| PALMER1D   | 10  26  26 | 11  25  25  | 10  24  23 |
| PALMER2C   | 11  20  20 | 11  21  21  | 11  21  22 |
| PALMER3C   | 11  20  20 | 11  20  20  | 11  21  21 |
| PALMER4C   | 11  20  20 | 11  20  20  | 11  21  21 |
| PALMER5C   | 11  20  20 | 11  20  20  | 11  21  21 |
| PALMER6C   | 11  25  26 | 11  29  28  | 11  27  28 |
| PALMER7C   | 11  21  22 | 11  20  20  | 11  20  20 |
| PALMER8C   | 11  21  22 | 11  18  17  | 11  19  19 |
| PENALTY1   | 20  63  53 | 28  69  44  | 14  51  43 |
| PENALTY2   | 185 220 351 | 191 221 354 | 337 480 758 |
| PENALTY3   | 79  247 201 | 99  285 219 | 102 346 290 |
| APPENDIX | 1455 |
| --- | --- |

| FTCGLS | FTCGHS | CG-DESCENT | DL+ |
| --- | --- | --- | --- |
| POWELLBSLS | 38 | 148 | 123 | 0.02 | 50 | 211 | 234 | 0.02 | 61 | 247 | 246 | 0.02 | 50 | 211 | 234 | 0.02 |
| POWELSG | 26 | 63 | 44 | 0.02 | 30 | 83 | 63 | 0.05 | 26 | 53 | 27 | 0.03 | 36 | 92 | 65 | 0.05 |
| 0.4 | 357 | 731 | 382 | 0.7 | 355 | 724 | 376 | 0.59 | 372 | 754 | 384 | 0.58 | 356 | 733 | 391 | 0.58 |
| POWERSUM | 5 | 11 | 6 | 0.02 | 4 | 10 | 6 | 0.59 | 5 | 11 | 6 | 0.02 | 4 | 10 | 6 | 0.02 |
| PRICE3 | 11 | 28 | 18 | 0.02 | 10 | 10 | 17 | 0.02 | 11 | 32 | 23 | 17 | 10 | 25 | 17 | 0.02 |
| PRICE4 | 10 | 28 | 21 | 0.02 | 9 | 30 | 23 | 0.02 | 68 | 135 | 87 | 0.03 | 67 | 134 | 85 | 0.03 |
| QUARTC | 15 | 32 | 18 | 0.02 | 37 | 167 | 155 | 0.08 | 17 | 37 | 21 | 0.02 | 15 | 32 | 18 | 0.02 |
| RAT43LS.SIF | 5 | 55 | 317 | 0.02 | 44 | 156 | 122 | 0.02 | 54 | 145 | 97 | 0.02 | 44 | 156 | 122 | 0.02 |
| RECIPELS.SIF | 14 | 36 | 26 | 0.02 | 16 | 49 | 38 | 0.02 | 27 | 58 | 32 | 0.02 | 16 | 49 | 38 | 0.02 |
| ROSENBR | 24 | 77 | 61 | 0.02 | 28 | 84 | 65 | 0.02 | 34 | 77 | 44 | 0.02 | 28 | 84 | 65 | 0.02 |
| ROSENBRTU.SIF | 39 | 166 | 141 | 0.02 | 37 | 175 | 153 | 0.02 | 49 | 156 | 113 | 0.02 | 37 | 175 | 153 | 0.02 |
| S308 | 7 | 20 | 16 | 0.02 | 7 | 21 | 17 | 0.02 | 8 | 19 | 12 | 0.02 | 7 | 21 | 17 | 0.02 |
| SCHMVTET | 43 | 74 | 59 | 0.02 | 38 | 69 | 52 | 0.17 | 43 | 73 | 60 | 0.02 | 59 | 103 | 88 | 0.28 |
| SENSORS | 21 | 57 | 41 | 0.02 | 25 | 68 | 47 | 0.38 | 21 | 50 | 34 | 0.02 | 24 | 71 | 53 | 0.47 |
| SINEVAL | 41 | 155 | 129 | 0.02 | 46 | 181 | 153 | 0.02 | 64 | 144 | 88 | 0.02 | 46 | 181 | 153 | 0.02 |
| SINOQUAD | 14 | 40 | 32 | 0.08 | 14 | 43 | 34 | 0.08 | 14 | 40 | 33 | 0.08 | 13 | 46 | 38 | 0.09 |
| SISER | 6 | 22 | 21 | 0.02 | 5 | 19 | 19 | 0.08 | 6 | 18 | 14 | 0.02 | 5 | 19 | 19 | 0.02 |
| SNAIL | 15 | 48 | 36 | 0.02 | 61 | 251 | 211 | 0.02 | 100 | 230 | 132 | 0.02 | 61 | 251 | 211 | 0.02 |
| SROSENBR | 9 | 23 | 16 | 0.02 | 9 | 25 | 19 | 1.14 | 11 | 23 | 12 | 0.02 | 9 | 23 | 15 | 0.02 |
| SSI | 276 | 892 | 1063 | 0.02 | 307 | 1162 | 990 | 0.02 | 345 | 948 | 657 | 0.02 | 307 | 1162 | 990 | 0.02 |
| STREG | 58 | 184 | 146 | 0.02 | 60 | 218 | 180 | 0.02 | 96 | 224 | 139 | 0.02 | 60 | 218 | 180 | 0.02 |
| STRATEC | 158 | 353 | 232 | 5.58 | 170 | 419 | 283 | 6.33 | 462 | 1043 | 796 | 19 | 170 | 419 | 283 | 6.3 |
| STRTCHDV.SIF | 11 | 36 | 33 | 0.02 | 12 | 38 | 32 | 0.02 | 16 | 35 | 20 | 0.02 | 12 | 38 | 32 | 0.02 |
| TESTQUAD | 1573 | 1580 | 3141 | 1.2 | 1580 | 1587 | 3155 | 1.52 | 1577 | 1584 | 3149 | 1.28 | 20325 | 20361 | 40674 | 21.61 |
| THURBERLS | 84 | 213 | 146 | 0.02 | 105 | 259 | 216 | 0.02 | 102 | 232 | 175 | 0.02 | 105 | 259 | 216 | 0.02 |
| TOINTGOR | 122 | 216 | 154 | 0.02 | 118 | 216 | 154 | 0.02 | 135 | 233 | 174 | 0.02 | 192 | 348 | 270 | 0.02 |
| TOINTGSS | 4 | 9 | 5 | 0.02 | 4 | 10 | 7 | 0.02 | 4 | 9 | 5 | 0.02 | 4 | 9 | 5 | 0.02 |
| TOINTPS | 162 | 343 | 259 | 0.02 | 151 | 319 | 250 | 0.02 | 143 | 279 | 182 | 0.02 | 145 | 313 | 250 | 0.02 |
| TOINTQDR | 29 | 36 | 53 | 0.02 | 29 | 36 | 53 | 0.02 | 29 | 36 | 53 | 0.02 | 49 | 56 | 93 | 0.02 |
| TQUARTIC | 11 | 39 | 32 | 0.02 | 13 | 45 | 37 | 0.03 | 14 | 40 | 27 | 0.02 | 11 | 41 | 34 | 0.03 |
| TRIDIA | 780 | 787 | 1555 | 0.75 | 780 | 787 | 1555 | 1.03 | 782 | 7889 | 155 | 0.89 | 469 | 4721 | 9408 | 6.38 |
| APPENDIX | 1456 |
|------------------|-------|
| **FTCGLS** | **FTCGHS** | **CG-DESCENT** | **DL+** |
| TRIGON1.SIF | 19 | 40 | 21 | 0.02 | 19 | 41 | 22 | 0.02 | 22 | 45 | 23 | 0.02 | 19 | 41 | 22 | 0.02 |
| TRIGON2.SIF | 25 | 58 | 37 | 0.02 | 22 | 57 | 43 | 0.02 | 26 | 52 | 28 | 0.02 | 22 | 57 | 43 | 0.02 |
| VANDANMSLS.SIF | 8 | 27 | 20 | 0.02 | 8 | 25 | 18 | 0.02 | 5 | 12 | 7 | 0.02 | 5 | 11 | 6 | 0.02 |
| VARDIM | 7 | 20 | 17 | 0.02 | 1 | 4 | 4 | 0.02 | 10 | 21 | 11 | 0.02 | 9 | 20 | 15 | 0.02 |
| VAREIGVL | 24 | 51 | 29 | 0.02 | 27 | 60 | 37 | 0.02 | 23 | 47 | 21 | 0.02 | 28 | 71 | 51 | 0.02 |
| VESUVIALS | 1136 | 1496 | 2821 | 0.02 | 1262 | 1954 | 3155 | 0.02 | 1519 | 2317 | 4111 | 1.56 | 1262 | 1954 | 3155 | 1.22 |
| VESUVIOULS | 72 | 168 | 117 | 0.02 | 79 | 211 | 173 | 0.02 | 80 | 180 | 131 | 0.06 | 79 | 211 | 173 | 0.09 |
| VIBRBEAM | 233 | 552 | 415 | 0.02 | 98 | 255 | 174 | 0.02 | 138 | 323 | 199 | 0.02 | 98 | 255 | 174 | 0.02 |
| WAYSEA1 | 12 | 49 | 41 | 0.02 | 11 | 55 | 50 | 0.02 | 18 | 39 | 22 | 0.02 | 11 | 55 | 50 | 0.02 |
| WAYSEA2 | 9 | 28 | 23 | 0.02 | 9 | 28 | 23 | 0.02 | 31 | 68 | 39 | 0.02 | 9 | 28 | 23 | 0.02 |
| WOODS | 26 | 65 | 43 | 0.05 | 68 | 184 | 129 | 0.02 | 22 | 51 | 30 | 0.03 | 24 | 62 | 41 | 0.03 |
| YATP1CLS | 17 | 48 | 36 | 5.3 | 14 | 41 | 31 | 5.76 | 23 | 53 | 31 | 6.13 | 17 | 48 | 36 | 7.12 |
| YFITU | 66 | 200 | 159 | 0.02 | 68 | 208 | 167 | 0.02 | 84 | 197 | 118 | 0.02 | 68 | 208 | 167 | 0.03 |
| ZANGWIL2 | 1 | 3 | 2 | 0.02 | 1 | 3 | 2 | 0.02 | 1 | 3 | 2 | 0.02 | 1 | 3 | 2 | 0.02 |