J. Math. Computer Sci., 36 (2025), 263–274

Online: ISSN 2008-949X

Journal of Mathematics and Computer Science

Journal Homepage: www.isr-publications.com/jmcs

An improved accelerated 3-term conjugate gradient algorithm with second-order Hessian approximation for nonlinear least-squares optimization

Rabiu Bashir Yunus[a,*], Nooraini Zainuddin[a], Hanita Daud[a], Ramani Kannan[b], Mahmoud Muhammad Yahaya[c], Abdullah Al-Yaari[a]

[a]*Department of Fundamental and Applied Sciences, Faculty of Science and Information Technology, Universiti Teknologi PETRONAS, Bandar Seri Iskandar 32610, Perak Darul Ridzuan, Malaysia.*

[b]*Department of Electrical and Electronics Engineering, Universiti Teknologi PETRONAS, Bandar Seri Iskandar 32610, Perak Darul Ridzuan, Malaysia.*

[c]*Department of Mathematics, Faculty of Science, King Mongkut's University of Technology Thonburi (KMUTT), 126 Pracha-Uthit Road, Bang Mod, Thung Khru, Bangkok 10140, Thailand.*

Abstract

Nonlinear least-squares (NLS) problems find extensive applications across various fields within the applied sciences. Conventional methods for solving NLS problems often face challenges related to computational efficiency and memory requirements, especially when dealing with large-scale systems. In this paper, the solution to the minimization of nonlinear least squares problems has been obtained using a proposed structured accelerated three-term conjugate gradient method, in which from Taylor series approximations of the objective function's Hessian, the structured vector approximation involving a vector's action on a matrix is obtained. This ensures the satisfaction of a quasi-Newton condition. The technique then employs the structured vector approximation to incorporate additional information from the Hessian of the goal function into the standardized search direction. The proposed method's search direction fulfills the necessary descent criterion. Additionally, numerical tests performed on various test problems show that the suggested approach is remarkably efficient, surpassing some existing competitors.

Keywords: Three-term, nonlinear, least-squares, accelerated, conjugate gradient.

2020 MSC: 65K05, 65K10.

1. Introduction

Numerous real-world scientific and engineering problems can be represented as nonlinear least squares (NLS) problems. NLS is highly relevant in many different domains and applications, including motion control [25, 26, 32], parameter estimation [12], image reconstruction, [27] and data fitting [34]. Consider an unconstrained nonlinear least-squares problem of the form:

$$\min_{x\in\mathbb{R}^n} f(x), \quad \text{where} \quad f(x) = \sum_{i=1}^{m} (r_i(x))^2 = \frac{1}{2}\|r(x)\|^2, \; x \in \mathbb{R}^n. \tag{1.1}$$

*Corresponding author

Email address: rabiu_22000788@utp.edu.my (Rabiu Bashir Yunus)

doi: 10.22436/jmcs.036.03.02

Received: 2024-04-19 *Revised:* 2024-06-07 *Accepted:* 2024-06-29

Conventional conjugate gradient (CG) techniques provide an iterative point sequence x_k, which is updated by the following form:

$$x_{k+1} = x_k + s_k, \ s_k = \alpha_k d_k, \ k \geqslant 0, \tag{1.2}$$

where d_k represents the calculated search direction and α_k is the step size. An exact or inexact line search strategy is required to determine the step size. Typically, the search direction is determined using

$$d_0 = -g_0, \quad d_k = -g_k + \beta_k d_{k-1}, \ k \geqslant 1.$$

In this case, the scalar CG coefficient β_k distinguishes the various conjugate gradient techniques [14]. Meanwhile, distinct CG parameters represent several conjugate gradient algorithms, each of which may exhibit notable variations in theoretical characteristics and numerical efficacy [35]. The Hestenes-Stiefel (HS) approach [11], the Polak-Ribière-Polyak (PRP) method [23, 24], and the Liu-Storey (LS) method [18] all share similarities in their numerators, which leads to certain shared characteristics in terms of computational advantage [10]. Moreover, their conjugate parameters are given by:

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

where $g_k = \nabla f(x_k)$ and $y_{k-1} = g_k - g_{k-1}$. Furthermore, additional details regarding additional CG techniques are available in [3, 8].

Due to the computational expense of determining the entire Hessian matrix, some methods were developed that exclusively employ information from the first derivative to solve (1.1) such as the Gauss-Newton (GN) and Levenberg-Marquardt (LM) techniques. However, when dealing with problems involving non-zero residuals, both approaches typically perform less than optimum [1]. As a result of this flaw, Brown and Dennis [7] developed the Structured Quasi-Newton method (SQN), this exploits the structure of the objective function's Hessian (1.1) by utilizing GN's and quasi-Newton's step. Meanwhile, in terms of numerical performance, the SQN algorithm outperforms the GN and LM algorithms [31]. However, the fact that the direction produced by the SQN method might not be a descent direction of f presents a significant challenge to the globalization of these approaches and this hinders their ability to solve large-scale problems. As a result of these drawbacks, structured matrix-free techniques are the better choice for solving (1.1). For example, [15] presented a structured matrix-free approach for solving large-scale NLS problems using conjugate gradient direction. An approximating technique based on the diagonal of the first and second terms of the Hessian matrix was proposed by [20]. To ensure the satisfaction of the sufficient descent criterion, their search direction necessitated a safeguarding technique. Recently, [6] suggested a scaled version of the structured gradient relation (CG) approach. Under some standard assumptions, they demonstrated the algorithm's global convergence. In another attempt, [36] proposed a two-term structured spectral CG method based on the HS formula with an outstanding numerical performance. However, their algorithms require the implementation of several safety strategies, especially when the spectral parameter is non-positive at certain iteration points. Additionally, [33] suggested a different spectral-based structured adaptive technique for nonlinear least squares issues, which are grounded on Barzilai and Borwein (BB) parameters [4]. It was observed that, conjugate gradient methods based on secant conditions may not theoretically guarantee a descent search direction, but they are highly effective in practice for solving large-scale unconstrained optimization problems [28]. Therefore, it is crucial to develop a conjugate gradient method that utilizes second-order information of the objective function to ensure a descent search direction.

In recent years, the three-term CG methods have garnered significant attention, and substantial efforts have been dedicated to developing these approaches. Sugiki et al. [28] applied the approach of Dai and Liao [5] to the three-term conjugate gradient method developed by Narushima et al. [22], and proposed a three-term conjugate gradient method based on secant conditions to solve NLS problems. Based on the least-squares approach, Tang et al. [30] developed novel three-term CG techniques for general nonlinear

function. In another development, two novel algorithms for three-term conjugate gradient (TTCG) coefficients have been devised in [37] to address nonlinear least-squares (NLS) problems by employing the structured secant equation; formulated via the enhanced conjugacy and sufficient descent conditions.

Motivated by the findings mentioned earlier, we formulated a more advanced structured approximation for the Hessian of the NLS problem. In this approach, we approximate the second term of the Hessian up to the second-order Taylor's series, thereby leveraging a significant amount of information from the Hessian. This matrix-vector approximation of the Hessian is incorporated into a novel three-term method. Furthermore, the method is straightforward to implement and does not rely on matrix operations, leading to low computational expenses per iteration.

This study presents a structured accelerated Three-term based method on LS and the modified LS method (mLS+), that incorporates second-order information and guarantees a descent search direction. The following summarizes the main contributions of this study.

1. Using a modified structured secant equation, we proposed a new accelerated three-term CG algorithm.
2. The search direction that has been suggested meets the requirements of sufficient descent property.
3. Under certain assumptions, it has been demonstrated that the proposed method attains global convergence with the help of a non-monotone line search.
4. Numerical experiments were performed to evaluate the efficacy of the proposed method in comparison to other methods reported in the literature.

The following sections of this article are arranged as follows. The suggested approach and related algorithm are described in Section 2. We examine the global convergence properties of the suggested algorithms in Section 3. Section 4 employs numerical tests to assess how well the suggested algorithms perform in comparison to alternative methods that have been documented in the literature.

2. Formulation of the new method and its algorithm

This section outlines the concept of the suggested approach along with its corresponding algorithm.

2.1. Derivation of modified structured secant

Specifically, the NLS problem described in equation (1.1) has a unique structure for its gradient and Hessian matrix, which can be expressed as

$$\begin{aligned} g(x) &:= \sum_{i=1}^{m} r_i(x)\nabla r_i(x) = J(x)^T r(x), \\ H(x) &:= \underbrace{\sum_{i=1}^{m} \nabla r_i(x)\nabla r_i(x)^T}_{\text{1st Term}} + \underbrace{\sum_{i=1}^{m} r_i(x)\nabla^2 r_i(x)}_{\text{2nd Term}} = J(x)^T J(x) + U(x), \end{aligned} \tag{2.1}$$

where the matrix $U(x)$ corresponds to the second term shown in equation (2.1), and $J(x) = R'(x)$ denotes the Jacobian matrix of the residual function $r_i(x)$ for $i = 1, 2, \ldots, m$. At the beginning, we estimate the product of the matrix $U(x)$ and the vector θ, which can be interpreted as approximating the effect of $\nabla^2 r_i(x)$ on a vector, such as $\theta \in \mathbb{R}^n$. This allows for the extraction of significant information from $U(x)$. As a result, we can achieve the necessary approximation by performing a post-multiplication of $U(x) = \sum_{i=1}^{m} r_i(x)\nabla^2 r_i(x)$ in (2.1) by s_{k-1}, that is

$$U(x_k)s_{k-1} = \sum_{i=1}^{m} r_i(x)\nabla^2 r_i(x)s_{k-1}, \tag{2.2}$$

Therefore, to simplify the notations, we define $U_k = U(x_k)$, $r_{k,i} = r_i(x_k)$, $\nabla^2 r_{k,i} = \nabla^2 r_i(x_k) = G_{k,i}$, and gradients of components-wise $r_{k,i}$ by $g_{k,i}$ for $i = 1, 2, \ldots, m$. Our current objective is to estimate $\nabla^2 r_{k,i} s_{k-1}$. We can accomplish this by integrating Taylor's series expansion of $r_{k,i}$ and $g_{k,i}$. This can be acquired in the manner described below:

$$r_{k-1,i} \approx r_{k,i} + g_{k,i}^T s_{k-1} + \frac{1}{2!} s_{k-1}^T G_{k,i} s_{k-1} \tag{2.3}$$

and in a similar manner, we obtain

$$g_{k-1,i} \approx g_{k,i} - G_{k,i} s_{k-1}. \tag{2.4}$$

To attain the intended outcome, we pre-multiply (2.4) by s_{k-1} first,

$$s_{k-1}^T g_{k-1,i} \approx s_{k-1}^T g_{k,i} - s_{k-1}^T G_{k,i} s_{k-1}. \tag{2.5}$$

We may rewrite (2.3) and (2.4) in terms of the term $s_{k-1}^T G_{k,i} s_{k-1}$ since both (2.3) and (2.5) include it in their expression. We can then equal and simplify the result to have as follows:

$$s_{k-1}^T G_{k,i} s_{k-1} \approx 2(r_{k-1,i} - r_{k,i}) + (g_k, i + g_{k-1,i})^T s_{k-1} + s_{k-1}^T (g_{k,i} - g_{k-1,i}).$$

Assuming we use a basic diagonal approximation for $G_{k,i}$, then

$$s_{k-1}^T G_{k,i} s_{k-1} \approx \zeta I \|s_{k-1}\|^2 \approx 2(r_{k-1,i} - r_{k,i}) + (g_k, i + g_{k-1,i})^T s_{k-1} + s_{k-1}^T (g_{k,i} - g_{k-1,i}).$$

As a result, the estimate of $G_{k,i} s_{k-1}$ is

$$G_{k,i} s_{k-1} \approx \frac{[2(r_{k-1,i} - r_{k,i}) + (g_k, i + g_{k-1,i})^T s_{k-1} + s_{k-1}^T (g_{k,i} - g_{k-1,i})]}{\|s_{k-1}\|^2} s_{k-1}. \tag{2.6}$$

Therefore, by putting (2.6) in (2.2) and adding up all of i gives

$$U_k s_{k-1} \approx \frac{[2(r_{k-1} - r_k) + (J_k + J_{k-1})^T s_{k-1} + s_{k-1}^T (J_k - J_{k-1})]}{\|s_{k-1}\|^2} s_{k-1} = z_{k-1},$$

by letting

$$\vartheta_{k-1} = r_k^T \left[2(r_{k-1} - r_k) + (J_k + J_{k-1})^T s_{k-1} + s_{k-1}^T (J_k - J_{k-1})\right].$$

Let us now assume that the Hessian in (2.1) is estimated in a way that makes it satisfy a structured secant condition, which is $B_k s_{k-1} \approx z_{k-1}$. Consequently, we obtain

$$z_{k-1} = J_k^T J_k s_{k-1} + \frac{\vartheta_{k-1}}{\|s_{k-1}\|^2} s_{k-1}.$$

2.2. *The proposed formula*

For the three-term parameters, we proposed a new search direction that is expressed as follows:

$$d_k = -g_k + \beta_k^{(LS)} d_{k-1} - \beta_k^{(mLS+)} z_{k-1}, \ \forall k \geqslant 0, \tag{2.7}$$

where $s_{k-1} = x_k - x_{k-1}$, and where β_k^{LS} is calculated as

$$\beta_k^{LS} = \frac{g_k^T z_{k-1}}{-g_{k-1}^T d_{k-1}}. \tag{2.8}$$

We propose a modified version of the LS method (mLS+) and make adjustments to the LS conjugate parameter in the following manner:

$$\beta_k^{mLS+} = \frac{g_k^T d_{k-1}}{-g_{k-1}^T d_{k-1}}. \tag{2.9}$$

We adopt the acceleration scheme presented in [2] to modify the step in a multiplicative way, improving the numerical performances. Therefore, the standard iteration (1.2) becomes

$$x_{k+1} = \begin{cases} x_k + \omega_k \alpha_k d_k, & \text{if } \gamma_k < 0, \\ x_k + \alpha_k d_k, & \text{otherwise,} \end{cases} \tag{2.10}$$

where ω_k, the parameter, is defined by $\omega_k = \frac{\varphi_k}{\gamma_k}$, and $\varphi_k = \alpha_k g_k^T d_k$ and $\gamma_k = \alpha_k [g_z - g_k]^T d_k$. Furthermore, to get the step length α_k, we use non-monotone line-search technique ([38]) in this study. If d_k exhibits a notable decrease, the step length α_k is determined by applying the non-monotone line-search requirements of the Armijo type, which are described as

$$f(x_k + \alpha_k d_k) \leqslant A_k + \delta \alpha_k g_k^T d_k,$$

where

$$\begin{cases} A_0 = f(x_0), \\ A_{k+1} = \frac{\eta_k B_k A_k + f(x_{k+1})}{B_{k+1}}, \\ B_0 = 1, \\ B_{k+1} = \eta_k B_k + 1. \end{cases} \tag{2.11}$$

Algorithm 1 Structured accelerated 3-term CG (SA-3TCG)

Step 1. Initialization: $x_0 \in \mathbb{R}^n$, termination tolerance $\epsilon > 0$.

Step 2. Evaluate g_k. If $\|g_k\| = 0$, terminate the iteration process.

Step 3. If $k = 0$, set $d_0 := -g_0$, otherwise,

$$d_k = -g_k + \beta_k^{(LS)} d_{k-1} - \beta_k^{(mLS+)} z_{k-1}, \ \forall\, k \geqslant 1, \tag{2.12}$$

where β_k^{LS} and β_k^{mLS+} follow from (2.8) and (2.9), respectively.

Step 4. Determine α_k using non-monotone line search.

Step 5. Calculate the new point via (2.10).

Step 6. Go back to Step 2 with $k := k + 1$.

3. Convergence analysis

The theoretical results of the suggested SA-3TCG approach are covered in this section. To attain these outcomes, we require the following significant presumptions, which are critical in determining the convergence outcomes of the majority of CG algorithms.

Assumptions

1. The level set $\chi = \{x \in \mathbb{R}^n | f(x) \leqslant f(x_0)\}$ of $f(x)$ is bounded; that is, a positive constant b exists such that $\|x\| \leqslant b$, $\forall x \in \chi$.
2. In an open neighborhood N of χ, the residual, $r(x)$, and its derivative, the Jacobian matrix, $J(x)$, are Lipschitz continuous and bounded, that is, $\|J(x) - J(y)\| \leqslant m_1 \|x - y\|$, $\|r(x) - r(y)\| \leqslant m_2 \|x - y\|$, $\|J(x)\| < m_3$, $\|r(x)\| < m_4$, for all $x, y \in \chi$, where m_1, m_2, m_3, m_4 symbolize constants that are positive.

From Assumption 2, it can be inferred that there exist positive constants m_5 and m_6 such that the following inequalities are satisfied:

$$\|g(x) - g(y)\| \leqslant m_5 \|x - y\|.$$

This implies that $\|g(x)\| \leqslant m_6$, $\forall x, y \in \chi$.

Lemma 3.1. *If Assumption 2 is satisfied, then there exists a constant $\kappa > 0$, such that*

$$\|z_{k-1}\| \leqslant \kappa \|s_{k-1}\|, \ \forall k.$$

The proof of the lemma is similar to that in [33] with slight differences.

Lemma 3.2. *Let $\{d_k\}$ be the sequence of direction generated by Algorithm 1, then, for every $k > 0$, there is a constant κ_1 such that $g_k^T d_k \leqslant -\kappa_1 \|g_k\|^2$.*

Proof. If $k = 0$, by the setting $d_0 = -g_0$, it holds that $g_0^T d_0 = -\|g_0\|^2$. For $k \geqslant 1$, by (2.7), we obtain

$$\begin{aligned} g_k^T d_k &= -\|g_k\|^2 + \beta_k^{LS} g_k^T d_{k-1} - \beta_k^{mLS+} g_k^T z_{k-1} \\ &= -\|g_k\|^2 + \frac{g_k^T z_{k-1}}{-g_{k-1}^T d_{k-1}} g_k^T d_{k-1} - \frac{g_k^T d_{k-1}}{-g_{k-1}^T d_{k-1}} g_k^T z_{k-1} \\ &= -\|g_k\|^2 - \frac{g_k^T z_{k-1}}{g_{k-1}^T d_{k-1}} g_k^T d_{k-1} + \frac{g_k^T d_{k-1}}{g_{k-1}^T d_{k-1}} g_k^T z_{k-1} = -\|g_k\|^2. \end{aligned}$$

Therefore, for all $k_1 > 0$, it holds that $g_k^T d_k = -\|g_k\|^2$ with $\kappa_1 = 1$. □

Lemma 3.3. *If Assumptions 1 and 2 are satisfied, then for all $\kappa_2 \geqslant 0$, the search direction d_k produced by Algorithm 1 remains bounded.*

Proof. Following (2.7)'s definition of the search direction, we have

$$\begin{aligned} \|d_k\| &= \| - g_k + \beta_k^{LS} d_{k-1} - \beta_k^{mLS+} z_{k-1}\| \\ &\leqslant \|g_k\| + |\beta_k^{LS}| \|d_{k-1}\| + |\beta_k^{mLS+}| \|z_{k-1}\| \\ &\leqslant \|g_k\| + \frac{\|g_k\| \|z_{k-1}\|}{\|g_{k-1}\| \|d_{k-1}\|} \|d_{k-1}\| + \frac{\|g_k\| \|d_{k-1}\|}{\|g_{k-1}\| \|d_{k-1}\|} \|z_{k-1}\| \\ &\leqslant \left(1 + \frac{\|z_{k-1}\|}{\|g_{k-1}\|} + \frac{\|z_{k-1}\|}{\|g_{k-1}\|}\right) \|g_k\| \leqslant \left(1 + 2\frac{\|z_{k-1}\|}{\|g_{k-1}\|}\right) \|g_k\| = \left(1 + \frac{2\kappa b}{m_6}\right) \|g_k\|. \end{aligned}$$

Therefore, if we let $\kappa_2 := \left(1 + \frac{2\kappa b}{m_6}\right)$, then we have $\|d_k\| \leqslant \kappa_2$. Hence, the search direction is bounded. □

The proof of Theorem 3.7 requires the following lemmas. However, [38] contains their proofs.

Lemma 3.4. *Given that Assumption 1 is satisfied, it can be shown that $f_k \leqslant A_k$ for each value of k if Algorithm 1 generates the iterative sequence $\{x_k\}$.*

Lemma 3.5. *Given that Assumption 2 is satisfied and the iterative sequence $\{x_k\}$ is produced by Algorithm 1, it consequently implies that*

$$\alpha_k \geqslant \left(\frac{2(1-\delta)}{m_5 \zeta}\right) \frac{|g_k^T d_k|}{\|d_k\|^2}.$$

Remark 3.6. Suppose that $A_j \leqslant \psi_j$ for all $0 \leqslant j < k$, given the initial condition $B_0 = 1$, and considering that $\eta_k \in [0, 1]$, then we have

$$B_{j+1} = 1 + \sum_{i=0}^{m} \prod_{m=0}^{i} \eta_{j-m} \leqslant j + 2. \tag{3.1}$$

Theorem 3.7. *Suppose that the function $f(x)$ is defined by equation (1.1) and that Assumptions 1 and 2 hold. The level set ℓ contains the sequence $\{x_k\}$ that is generated by Algorithm 1,*

$$\lim_{k \to \infty} \inf \|g_k\| = 0. \tag{3.2}$$

Moreover, if $\eta_{max} < 1$, then

$$\lim_{k \to \infty} \|g_k\| = 0. \tag{3.3}$$

Proof. To start, we present the fact that

$$f_{k+1} \leqslant A_k - \beta \|g_k\|^2, \tag{3.4}$$

considering the line search in equation (2.9), we have

$$f_{k+1} \leqslant A_k + \delta \alpha_k g_k^T d_k,$$

and by the inequality in equation (3.1), we obtain

$$f_{k+1} \leqslant A_k - \left(\frac{2\delta(1-\delta)}{m_5 \zeta}\right)\left(\frac{|g_k^T d_k|}{\|d_k\|^2}\right).$$

Drawing from the necessary descent and bound properties from Lemmas 3.3 and 3.4, we obtain

$$f_{k+1} \leqslant A_k - \left(\frac{2\delta(1-\delta)\kappa_1^2}{m_5 \zeta \kappa_2^2}\right)\|g_k\|^2,$$

where

$$\beta = \frac{2\delta(1-\delta)\kappa_1^2}{m_5 \zeta \kappa_2^2}.$$

Combining the cost update in equations (2.11) and (3.4), we can obtain

$$\begin{aligned} A_k = \frac{\eta_k B_k A_k + f_{k+1}}{B_{k+1}} &\leqslant \frac{\eta_k B_k A_k + A_k - \beta\|g_k\|^2}{B_{k+1}} \\ &= \frac{A_k(\eta_k B_{k+1} + 1) - \beta\|g_k\|^2}{B_{k+1}} = \frac{A_k B_{k+1} - \beta\|g_k\|^2}{B_{k+1}} = A_k - \frac{\beta\|g_k\|^2}{B_{k+1}}. \end{aligned} \tag{3.5}$$

As f is bounded from below, and for all k, $f_k \leqslant A_k$, we can deduce that A_k is also bounded from below. Therefore, it follows from (3.5) that

$$\sum_{k=1}^{\infty} \frac{\|g_k\|^2}{B_{k+1}} < \infty$$

If $\|g_k\|$ were bounded away from 0, the equation (3.5) would not hold since $B_{k+1} \leqslant k+2$ by (3.1). Hence, if $\eta_{max} < 1$, then by (3.1),

$$B_{k+1} = 1 + \sum_{j=0}^{k}\prod_{i=0}^{j}\eta_{k-i} \leqslant 1 + \sum_{j=0}^{k}\eta_{max}^{j+1} \leqslant \sum_{j=0}^{k}\eta_{max}^{j} = \frac{1}{1-\eta_{max}}.$$

Hence, we can deduce that equation (3.2) leads directly to equation (3.3). Thus, the proof is complete. □

4. Numerical experiments

This section presents numerical tests on several nonlinear least-squares benchmark functions to substantiate the outstanding theoretical properties of the proposed method. We examine 25 benchmark test functions that span five dimensions 3000, 6000, 9000, 12000, and 15,000. These benchmark test functions are detailed in Table 1.

Table 1: List of test functions, starting points, and their sources.

No.	Function name	Starting point	Reference
F1	Penalty function 1	$(3,3,\ldots,3)^T$	[16]
F2	Variably dimensioned	$(1-\frac{1}{n},1-\frac{2}{n},\ldots,0)^T$	[16]
F3	Trigonometric function	$(\frac{1}{n},\frac{1}{n},\ldots)^T$	[21]
F4	Discrete boundary-value	$(\frac{1}{n+1},\frac{1}{n+1}-1,\ldots,\frac{1}{n+1}-1)^T$	[21]
F5	Linear full rank	$(1,1,\ldots,1)^T$	[21]
F6	Problem 202	$(2,2,\ldots,2)^T$	[19]
F7	Problem 206	$(2,2,\ldots,2)^T$	[19]
F8	Problem 212	$(0.5,0.5,\ldots,0.5)^T$	[19]
F9	Raydan 1	$(\frac{1}{n},\frac{2}{n},\ldots,1)^T$	[16]
F10	Raydan 2	$(\frac{1}{10n},\frac{2}{10n},\ldots,\frac{1}{10n})^T$	[16]
F11	Sine function 2	$(1,1,\ldots,1)^T$	[17]
F12	Exponential function 1	$(\frac{n}{n-1},\frac{n}{n-1},\ldots,\frac{n}{n-1})^T$	[16]
F13	Exponential function 2	$(\frac{1}{n^2},\frac{1}{n^2},\ldots,\frac{1}{n^2})^T$	[16]
F14	Singular function 2	$(1,1,\ldots,1)^T$	[16]
F15	Ext. Freudenstein & Roth function	$(6,3,6,3,\ldots,6,3)^T$	[16]
F16	Ext. Powell singular function	$(1.5E-4,\ldots,1.5E-4)^T$	[16]
F17	Function 21	$(1,1,\ldots,1)^T$	[16]
F18	Broyden tridiagonal function	$(-1,-1,\ldots,-1)^T$	[21]
F19	Extended Himmelblau function	$(1,\frac{1}{n},1,\frac{1}{n},\ldots,1,\frac{1}{n})^T$	[13]
F20	Function 27	$(100,\frac{1}{n^2},\ldots,\frac{1}{n^2})^T$	[16]
F21	Triglog function	$(1,1,\ldots,1)^T$	[13]
F22	Zerojacobian function	if $i=1$, $\frac{100(n-100)}{n}$, if $i\geqslant 2$, $\frac{(n-1000)(n-500)}{(60n)^2}$	[16]
F23	Exponential function	$(0.5,0.5,\ldots,0.5)^T$	[16]
F24	Function 18	$(0,0,\ldots,0)^T$	[16]
F25	Brown almost linear function	$(0.5,0.5,\ldots,0.5)^T$	[21]

The efficiency of the new formula is established by comparing it with the results of other existing algorithms with similar characteristics, for the comparison, in the numerical experiments we consider the following methods.

- A structured spectral (SSHS) CG approach by [36] with the following direction and spectral parameter $d_k=-\lambda_k g_k+\beta_k d_{k-1}$, where $\beta_k=\max\left\{0,\frac{g_k^T\bar{\omega}_{k-1}}{d_{k-1}^T\bar{\omega}_{k-1}}\right\}$, and spectral parameter $\lambda_k=\frac{s_{k-1}^T s_{k-1}}{s_{k-1}^T\bar{\omega}_{k-1}}$, and the structured vector defined by $\bar{\omega}_{k-1}=J^TJs_{k-1}+(J-J_{k-1})^Tr_k$.
- A three-term conjugate gradient method (TTRMIL+) by [29] whose direction defines by

$$d_k=-g_k+\beta_k^{RMIL+}d_{k-1}+\theta_k y_{k-1}$$

with $\beta_k^{\text{RMIL+}}=\begin{cases}\frac{g_k^T y_{k-1}}{\|d_{k-1}\|^2}, & \text{if } 0\leqslant g_k^T g_{k-1}\leqslant\|g_k\|^2,\\ 0, & \text{otherwise,}\end{cases}$ and $\theta_k=\tau+\frac{y_{k-1}^T y_{k-1}}{\|d_{k-1}\|^2}$.

- The classical Liu-Storey (LS) conjugate gradient method by [18].

This experiment's algorithms were all developed in MATLAB R2022a, and the computation was carried out on an Intel (R) CORE(TM) i7-3537U processor running at 2.00 GHz and having 8 GB of RAM. For the stopping condition, we command the program to terminate if $\|g_k\|\leqslant 10^{-5}$ or should the following situation arise:

- if there are more than 1000 iterations;

- if the number of function evaluations exceeds 5000.

In assessing the efficacy of the suggested algorithm, we employ the subsequent comparative metrics. (i) (ITER), the number of iterations; (ii) (FEVAL), the number of function evaluations; (iii) (NGRAD), the number of gradient evaluations; (iv) CPU time (CPU-Time); and (vi) residual value (VALUE_F). The following link will take you to Table 2 with the presentation of the numerical experiment results https://acrobat.adobe.com/id/urn:aaid:sc:EU:342dbf9a-4017-48ea-aee0-c0fb4c93eeb0. Subsequently, we employ a method developed by [9] to analyze the numerical efficiency based on the data provided in Table 2. The authors in [9] introduced a methodology to evaluate and compare the efficacy of a solver (s) across a set of problems (p). This method calculates the computational expense (q) associated with each combination of solver s and problem p, considering n_s solvers and n_p problems, as $q_{p,s}$ = the cost of solving problem p by solvers s.

Drawing from the computational cost $q_{p,s}$, [9] additionally devised a metric to facilitate the comparison of efficiency among all solvers. The performance ratio, which is the benchmark, is assessed using

$$r_{p,s} = \frac{q_{p,s}}{\min\{q_{p,s} : s \in S\}}$$

and the distribution function is described as $\rho_s(\tau) = \frac{1}{n_p}\text{size}\{p \in P : \log_2(r_{p,s}) \leqslant \tau\}$. Using the provided data, the described method generates performance profile graphs for each solver $s \in S$. These graphs, known as performance profiles, analyze the proportion $\rho_s(\tau)$ of the considered problems for each solution, where $\tau \geqslant 0$. For a specific τ value, an algorithm is deemed more efficient if it yields a higher $\rho_s(\tau)$ value. Therefore, the algorithm with the highest efficiency is the one whose curve is at the top.

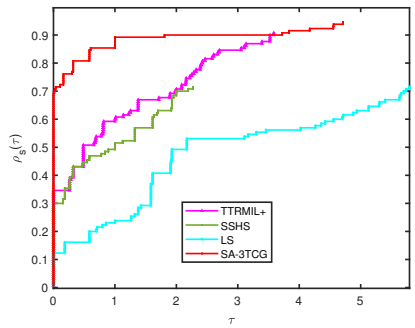

Figure 1: Performance assessment based on the number of iterations.

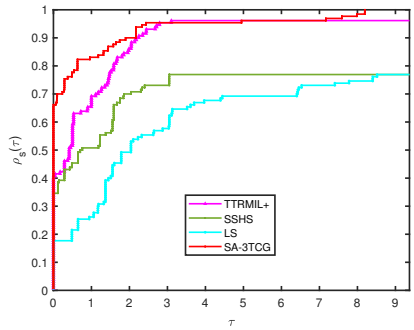

Figure 2: Performance assessment based on functions evaluation.

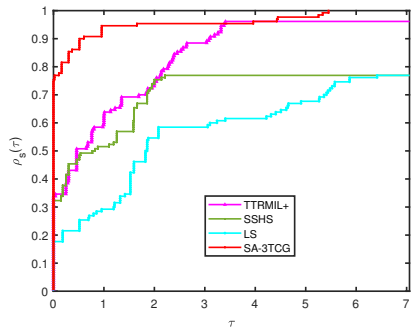

Figure 3: Performance assessment based on the number of gradient evaluations.

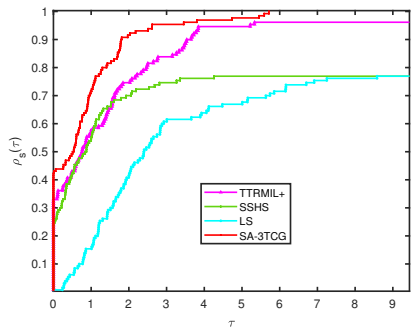

Figure 4: Performance assessment based on CPU time.

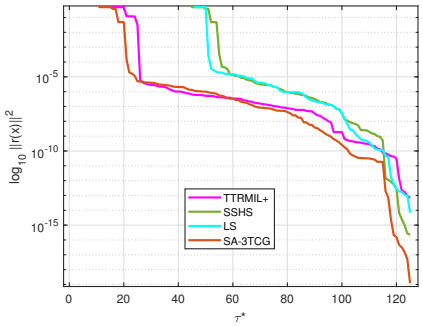

Figure 5: Performance based on the least zero residual error.

In this study, the performance profile plots for (ITER), (FEVAL), (NGrad), (CPU-time), and (VALUE_F) are generated from numerical experiments and are presented in Figures 1, 2, 3, 4, and 5, respectively.

The comparisons were made as shown in Figures 1-5. Upon analyzing Figure 1, for instance, when $\tau \geqslant 1$, SA-3TCG solves about 93% of the test problems with fewer iterations, while TTRMIL+, SSHS, and LS solve nearly 85%, 68%, and 61% of the problems, respectively. In Figure 2, at $\tau \geqslant 2$, SA-3TCG addresses nearly 95% of the test problems with the minimum number of function evaluations, compared to approximately 92%, 73%, and 68% handled by TTRMIL+, SSHS, and LS, respectively. Figure 3 shows that SA-3TCG successfully handles nearly 95% of the test problems with the minimal number of gradient evaluations, while TTRMIL+, SSHS, and LS manage 90%, 75%, and 67% of the problems in terms of gradient evaluations, respectively. Additionally, in terms of CPU-time, Figure 4, particularly at $\tau \geqslant 2$, SA-3TCG demonstrates effectiveness by solving approximately 98% of the test problems in the shortest amount of time, compared to almost 94%, 72%, and 64% addressed by TTRMIL+, SSHS, and LS, respectively. Moreover, SA-3TCG algorithms outperform TTRMIL+, SSHS, and LS algorithms, as indicated in Figure 5, showing that SA-3TCG methods' convergence offers a notably accurate approximation of the solution with significantly fewer errors, where τ^* represents instances of zero residual problems. One noteworthy observation from these figures is that, while all algorithms competed in first, the new technique outscored the others on all metrics as τ increased. This demonstrates how strong and competitive the new formula is for the problems under consideration.

5. Conclusion

In this article, based on the three-term conjugate gradient parameter, we presented a structured accelerated conjugate gradient method for this study. The method that is being provided satisfies the decent criterion regardless of the line search that is employed. We examined its performance on a set of benchmark problems from the literature and established its global convergence under the non-monotone line search conditions. The results show the method's competitiveness versus numerous alternative approaches with similar structures found in the literature. Further study is needed to devise more advanced conjugate gradient algorithms that can be applied to high-dimensional problems arising in science and engineering.

Acknowledgments

This research work was funded by the National Collaborative Research Fund between Universiti Teknologi PETRONAS and Universiti Malaysia Pahang with cost centre: 015MC0-036.

References

[1] M. Al-Baali, R. Fletcher, *Variational methods for non-linear least-squares*, J. Oper. Res. Soc., **36** (1985), 405–421. 1

[2] N. Andrei, *An acceleration of gradient descent algorithm with backtracking for unconstrained optimization*, Numer. Algorithms, **42** (2006), 63–73. 2.2

[3] N. Andrei, *A scaled BFGS preconditioned conjugate gradient algorithm for unconstrained optimization*, Appl. Math. Lett., **20** (2007), 645–650. 1

[4] J. Barzilai, J. M. Borwein, *Two-point step size gradient methods*, IMA J. Numer. Anal., **8** (1988), 141–148. 1

[5] Y.-H. Dai, L.-Z. Liao, *New conjugacy conditions and related nonlinear conjugate gradient methods*, Appl. Math. Optim., **43** (2011) 87–101. 1

[6] R. Dehghani, N. Mahdavi-Amiri, *Scaled nonlinear conjugate gradient methods for nonlinear least squares problems*, Numer. Algorithms, **82** (2019), 1–20. 1

[7] J. E. Dennis, Jr., H. J. Martínez, R. A. Tapia, *Convergence theory for the structured BFGS secant method with an application to nonlinear least squares*, J. Optim. Theory Appl., **61** (1989), 161–178. 1

[8] T. Diphofu, P. Kaelo, A. R. Tufa, *A convergent hybrid three-term conjugate gradient method with sufficient descent property for unconstrained optimization*, Topol. Algebra Appl., **10** (2022), 47–60. 1

[9] E. D. Dolan, J. J. Moré, *Benchmarking optimization software with performance profiles*, Math. Program., **91** (2002), 201–213. 4

[10] W. W. Hager, H. Zhang, *A survey of nonlinear conjugate gradient methods*, Pac. J. Optim., **2** (2006), 35–58. 1

[11] M. R. Hestenes, E. Stiefel, *Methods of conjugate gradients for solving linear systems*, J. Research Nat. Bur. Standards, **49** (1952), 409–436. 1

[12] J. Iwaniec, T. Uhl, *The application of the nonlinear least squares frequency domain method to estimation of the modal model parameters*, Machine Dynamics Problems, **27** (2003), 37–54. 1

[13] M. Jamil, X.-S. Yang, *A literature survey of benchmark functions for global optimisation problems*, Int. J. Math. Model. Numer. Optim., **4** (2013), 150–194. 1

[14] K. Kamfa, S. Ibrahim, S. F. Sufahani, R. B. Yunus, M. Mamat, *A modified bfgs method via new rational approximation model for solving unconstrained optimization problems and its application*, Adv. Math.: Sci. J., **9** (2020), 10771–10786. 1

[15] M. Kobayashi, Y. Narushima, H. Yabe, *Nonlinear conjugate gradient methods with structured secant condition for nonlinear least squares problems*, J. Comput. Appl. Math., **234** (2010), 375–397. 1

[16] W. La Cruz, J. Martínez, M. Raydan, *Spectral residual method without gradient information for solving large-scale nonlinear systems of equations*, Math. Comput., **75** (2006), 1429–1448. 1

[17] J. K. Liu, S. J. Li, *A projection method for convex constrained monotone nonlinear equations with applications*, Comput. Math. Appl., **70** (2015), 2442–2453. 1

[18] Y. Liu, C. Storey, *Efficient generalized conjugate gradient algorithms. I. Theory*, J. Optim. Theory Appl., **69** (1991), 129–137. 1, 4

[19] L. Lukšan, J. Vlcek, *Test problems for unconstrained optimization*, Tech. Rep., Acad. Sci. Czech Republic, Inst. Comput. Sci., (2003). 1

[20] H. Mohammad, S. A. Santos, *A structured diagonal Hessian approximation method with evaluation complexity analysis for nonlinear least squares*, Comput. Appl. Math., **37** (2018), 6619–6653. 1

[21] J. J. Moré, B. S. Garbow, K. E. Hillstrom, *Testing unconstrained optimization software*, ACM Trans. Math. Software, **7** (1981), 17–41. 1

[22] Y. Narushima, H. Yabe, J. A. Ford, *A three-term conjugate gradient method with sufficient descent property for unconstrained optimization*, SIAM J. Optim., **21** (2011), 212–230. 1

[23] E. Polak, G. Ribière, *Note sur la convergence de méthodes de directions conjuguées*, Rev. Française Informat. Recherche Opérationnelle, **3** (1969), 35–43. 1

[24] B. T. Polyak, *The conjugate gradient method in extreme problems*, USSR Comput. Math. Math. Phys., **9** (1969), 94–112. 1

[25] N. Salihu, P. Kumam, A. M. Awwal, I. Arzuka, T. Seangwattana, *A structured Fletcher-Revees spectral conjugate gradient method for unconstrained optimization with application in robotic model*, Oper. Res. Forum, **4** (2023), 25 pages. 1

[26] N. Salihu, P. Kumam, A. M. Awwal, I. M. Sulaiman, T. Seangwattana, *The global convergence of spectral RMIL conjugate gradient method for unconstrained optimization with applications to robotic model and image recovery*, Plos One, **18** (2023), 1–19. 1

[27] N. Salihu, P. Kumam, I. M. Sulaiman, T. Seangwattana, *An efficient spectral minimization of the Dai-Yuan method with application to image reconstruction*, AIMS Math., **8** (2023), 30940–30962. 1

[28] K. Sugiki, Y. Narushima, 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. 1

[29] I. M. Sulaiman, M. Malik, A. M. Awwal, P. Kumam, M. Mamat, S. Al-Ahmad, *On three-term conjugate gradient method for optimization problems with applications on COVID-19 model and robotic motion control*, Adv. Contin. Discrete Models, **2022** (2022), 22 pages. 4

[30] C. Tang, S. Li, Z. Cui, *Least-squares-based three-term conjugate gradient methods*, J. Inequal. Appl., **2020** (2020), 22 pages. 1

[31] F. Wang, D.-H. Li, L. Qi, *Global convergence of Gauss-Newton-MBFGS method for solving the nonlinear least squares problem*, Adv. Model. Optim., **12** (2010), 1–19. 1

[32] M. M. Yahaya, P. Kumam, A. M. Awwal, S. Aji, *A structured quasi-Newton algorithm with nonmonotone search strategy for structured NLS problems and its application in robotic motion control*, J. Comput. Appl. Math., **395** (2021), 17 pages. 1

[33] M. M. Yahaya, P. Kumam, P. Chaipunya, T. Seangwattana, *Structured adaptive spectral-based algorithms for nonlinear least squares problems with robotic arm modelling applications*, Comput. Appl. Math., **42** (2023), 31 pages. 1, 3

[34] R. B. Yunus, K. Kamfa, S. I. Mohammed, M. Mamat, *A Novel Three Term Conjugate Gradient Method for Unconstrained Optimization via Shifted Variable Metric Approach with Application*, In: Intelligent Systems Modeling and Simulation II, Studies in Systems, Decision and Control, Springer, Cham, **444** (2022), 581–596. 1

[35] R. B. Yunus, M. Mamat, A. Abashar, M. Rivaie, Z. Salleh, Z. A. Zakaria, *The convergence properties of a new kind of conjugate gradient method for unconstrained optimization*, Appl. Math. Sci., **9** (2015), 1845–1856. 1

[36] R. B. Yunus, N. Zainuddin, H. Daud, R. Kannan, S. A. A. Karim, M. M. Yahaya, *A modified structured spectral HS method for nonlinear least squares problems and applications in robot arm control*, Mathematics, **11** (2023), 1–17. 1, 4

[37] R. B. Yunus, N. Zainuddin, H. Daud, R. Kannan, M. M. Yahaya, S. A. A. Karim, *New CG-Based Algorithms With Second-Order Curvature Information for NLS Problems and a 4DOF Arm Robot Model*, IEEE Access, **12** (2024), 61086–61103. 1

[38] H. Zhang, W. W. Hager, *A nonmonotone line search technique and its application to unconstrained optimization*, SIAM J. Optim., **14** (2004), 1043–1056. 2.2, 3