PLATFORM - A Journal of Science & Technology

DOI:https://doi.org/10.61762/pjstvol7iss1art27055

Received: 30 May 2024, Accepted: 24 June 2024, Published: 28 June 2024, Publisher: UTP Press, Creative Commons: CC BY 4.0

ON SPECTRAL DAI-YUAN STRUCTURED TECHNIQUE FOR SOLVING NONLINEAR LEAST-SQUARES OPTIMISATION

Rabiu Bashir Yunus[1,2*], Nooraini Zainuddin[1], Hanita Daud[1], Bashir Danladi Garba[2]

[1]Department of Fundamental and Applied Sciences, Universiti Teknologi PETRONAS, Malaysia
[2]Department of Mathematics, Kano University of Science and Technology, Nigeria

*E-mail: rabiu_22000788@utp.edu.my

ABSTRACT

Handling nonlinear least squares (NLS) problems as unconstrained optimisation problems has led to the development and adaptation of numerous numerical approaches. Calculating the Hessian matrix expensively for each iteration is one of the problems with the current numerical approaches for solving NLS problems. Some methods designed to rely solely on first-derivative information while ignoring second-order details tend to underperform when dealing with problems involving non-zero residuals. This article suggests a modification to the Dai-Yuan type (DY) formula by deriving a spectral parameter for the proposed search direction. The sufficient descent results of the new formula are established using the standard assumptions under the strong Wolfe line search. Furthermore, experiments are conducted on some benchmark functions to demonstrate the computational efficiency of the proposed technique. The results highlight the effectiveness of the algorithm compared to existing methods.

Keywords: *conjugate gradient, convergence theory, Hessian matrix, optimisation, Quasi-Newton, spectral, structured secant method*

INTRODUCTION

In the last few decades, a number of numerical techniques have been developed and improved upon to address nonlinear least squares (NLS) issues as unconstrained optimisation problems [1]-[2]. The problem is formulated as follows:

$$\min_{x\in\mathbb{R}^n} f(x) = \frac{1}{2}\|r(x)\|^2 \tag{1}$$

NLS is extremely pertinent across various fields and applications, particularly in motion control [3], data fitting [4], and image reconstruction [5]. Most of the numerical methods employed to tackle NLS problems follow an iterative approach. Beginning with an initial point x_0, each iteration computes a search direction d_k and a step length α_k iteratively, as follows:

$$x_{k+1} = x_k + \alpha_k d_k, \tag{2}$$

The direction d_k is defined recursively as follows:

$$d_k = \begin{cases} -g_k, & for\ k = 0, \\ -g_k + \beta_k d_{k-1}, & for\ k \geq 1. \end{cases} \tag{3}$$

where g_k represents the gradient and β_k is a scalar defining the conjugate gradient (CG) method [6]. The following formulas are applicable to the classical CG methods of Fletcher and Reeves (FR) [7], Conjugate Descent (CD) [8], and Dai and Yuan (DY) [9] :

$$\beta_k^{FR} = \frac{\|g_k\|^2}{\|g_{k-1}\|^2}, \tag{4}$$

$$\beta_k^{CD} = \frac{\|g_k\|^2}{-d_{k-1}^T g_{k-1}}, \tag{5}$$

$$\beta_k^{DY} = \frac{\|g_k\|^2}{d_{k-1}^T y_{k-1}}. \tag{6}$$

In the above formulas, $y_{k-1} = g_k - g_{k-1}$,, and the symbol $\|.\|$ denotes the Euclidean norm of vectors. Moreover, Common features in the numerator of the FR, CD, and DY methods result in similarities in computational performance [10].

The gradient and Hessian of Equation 1 possess a distinct structure [11] described by:

$$\nabla f(x) := J(x)^T r(x), \tag{7}$$

and

$$\nabla^2 f(x) := \underbrace{J(x)^T J(x)}_{1st\ term} + \underbrace{\sum_{i=1}^{n} r_i(x)\nabla^2 r_i(x)}_{2nd\ term}. \tag{8}$$

where the gradient of $r(x)$ is denoted by the Jacobian matrix $J(x)$. Due to the high computational cost of calculating the full Hessian matrix (8), methods that utilise only first derivative information have been developed [12]. For instance, the Gauss-Newton (GN) method and the Levenberg-Marquardt (LM) method approximate $\nabla^2 f(x_k)$ as follows:

$$\nabla^2 f(x_k) \approx B^{GN} = J(x_k)^T r(x_k), \tag{9}$$

and,

$$\nabla^2 f(x_k) \approx B^{LM} = J(x_k)^T r(x_k) + \eta_k I. \tag{10}$$

with η_k being a positive parameter.

Because these methods ignore the second part of the Hessian matrix in (8), they are likely to perform well when the residual $f(x^*)$ is small or when the functions r_i are nearly linear (referred to as small residual problems). But, when the residual $f(x^*)$ is very large, and the functions r_i are quite nonlinear, these methods may perform poorly (referred to as large residual problems). To address this subpar performance, structured quasi-Newton methods approximate the second component of (8) at x_k with A_k, and the overall approximation of the entire Hessian is provided by:

$$\nabla^2 f(x_k) \approx B^{SQN} = J(x_k)^T r(x_k) + A_k. \tag{11}$$

Nevertheless, structured quasi-Newton methods are not as effective as the Gauss-Newton method. This is because a matrix A_k produced by quasi-Newton updates does not converge to the zero matrix. To mitigate this deficiency, Dehghani and Mahdavi-Amiri [13] extended conjugate gradient methods in terms of the structured secant relation to solving nonlinear least squares problems. They showed these methods to be globally convergent. In another attempt, Yunus et al. [14] proposed a structured spectral conjugate gradient based on the Hestenel-Stiefiel formula. However, theoretically, this method does not always yield a descent search direction. To overcome this weakness, in this work, this article proposed another structured spectral formula based on the Dai and Yuan method using the Dai-Liao [15] concept to solve the NLS problems. Moreover, the structured vector equation is defined by Yahaya et. al [16] as:

$$y_{k-1} = J(x_k)^T J(x_k) s_{k-1} + \left(J(x_k) - J(x_{k-1})\right)^T r(x_k). \tag{12}$$

This article is divided into the following sections in the following order: In Section 2, the methodology and associated algorithm are explained. In Section 3, we investigate the global convergence features of the proposed algorithms. Section 4 uses numerical tests to evaluate the performance of the proposed algorithms against other approaches that have been reported in the literature. Lastly, conclusions are drawn in the final section.

METHODOLOGY

Formulation of the Structured Spectral Dai-Yuan (SSDY)

In this subsection, the computation of a new spectral parameter will build upon Mustafa's work [17]. This is usually the orientation that the SSDY-search direction follows:

$$d_k = \begin{cases} -g_k, & if \quad k = 0, \\ -\theta_k^* g_k + \beta_k^{DY} d_{k-1}, & if, \quad k > 0, \end{cases} \tag{13}$$

where θ_k is the spectral parameter to be determined.

$$d_k = -\theta_k g_k + \beta_k^{DY} d_{k-1} = -\theta_k g_k + \frac{\|g_k\|^2}{d_{k-1}^T y_{k-1}} d_{k-1}$$

Multiplying both sides by y_{k-1}^T, will obtain as:

$$d_k^T y_{k-1} = -\theta_k g_k^T y_{k-1} + \frac{\|g_k\|^2}{d_{k-1}^T y_{k-1}} d_{k-1}^T y_{k-1} \tag{14}$$

Considering the Dai-Liao [15] conjugacy condition $d_k^T y_{k-1} = -t g_k^T s_{k-1}$. And substituting $d_k^T y_{k-1}$ in Equation 14, will obtain as:

$$-t g_k^T s_{k-1} = -\theta_k g_k^T y_{k-1} + \|g_k\|^2$$

Dividing both sides of the above equation by $g_k^T y_{k-1}$, will obtain the spectral parameter as follows:

$$\theta_k = \frac{\|g_k\|^2}{g_k^T y_{k-1}} + \frac{t g_k^T s_{k-1}}{g_k^T y_{k-1}}$$

The optimal spectral parameter can be computed as follows:

$$\theta_k^* = \begin{cases} \frac{\|g_k\|^2}{g_k^T y_{k-1}} + \frac{t g_k^T s_{k-1}}{g_k^T y_{k-1}}, & if \ \left|g_k^T y_{k-1}\right| > d_{k-1}^T y_{k-1}, \\ \theta_k, & otherwise. \end{cases} \quad (15)$$

Hence, the proposed SSDY search direction is given by:

$$d_k = -\left(\frac{\|g_k\|^2}{g_k^T y_{k-1}} + \frac{t g_k^T s_{k-1}}{g_k^T y_{k-1}}\right) g_k + \frac{\|g_k\|^2}{d_{k-1}^T y_{k-1}} d_{k-1}, \quad (16)$$

In an inexact line search, the strong Wolfe line-search identifies the value of α_k that minimises the objective function along d_k by using specific conditions.

$$f(x_k + \alpha_k d_k) \le f(x_k) + \delta \alpha_k g_k^T d_k, \quad (17)$$

$$\left|g(x_k + \alpha_k d_k)^T d_k\right| \le \sigma \left|g_k^T d_k\right| \quad (18)$$

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

The Proposed Algorithm: Structured Spectral Dai-Yuan (SSDY)

In this subsection, the explanation of the successive actions required to solve Equation 1 is given by Algorithm 1.

Algorithm 1 Structured Spectral Dai-Yuan CG (SSDY)

Step 1: Initialisation: $x_0 \in \mathbb{R}^n$, Termination tolerance $\varepsilon > 0$.

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

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

$$d_k = -\left(\frac{\|g_k\|^2}{g_k^T y_{k-1}} + \frac{t g_k^T s_{k-1}}{g_k^T y_{k-1}}\right) g_k + \frac{\|g_k\|^2}{d_{k-1}^T y_{k-1}} d_{k-1}$$

Step 4: Determine α_k using strong Wolfe line search in equations (17-18).

Step 5: Calculate the new point via equation (2).

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

CONVERGENCE ANALYSIS

This section presents the convergence analysis of the SSDY approach. Key assumptions are summarised as:

Assumption 1

A1. The set of points defined as $L = \{x: f(x) \le f(x_0)\}$ is bounded, where x_0 represents the starting point of Algorithm 1.

A2. Certain positive constants a_1, a_2, a_3, b_1, b_2, b_3, exist such that

$$\|J(x) - J(y)\| \le a_1 \|x - y\|, \quad \text{for all } x, y \in L \quad (19)$$

$$\|r(x) - r(y)\| \le a_2 \|x - y\|, \quad \text{for all } x, y \in L \quad (20)$$

$$\|\nabla f(x) - \nabla f(y)\| \le a_3 \|x - y\|, \quad \text{for all } x, y \in L \quad (21)$$

and

$$\|r(x)\| \le b_1, \ \ \|J(x)\| \le b_2, \ \ \|g(x)\| \le b_3 \ \ \forall x \in L \quad (22)$$

Lemma 1. Suppose f satisfies Assumption 1 and defines it as y_{k-1}(12). There is a constant ϕ such that

$$\|y_{k-1}\| \le \phi \|s_{k-1}\|, \quad \forall k \ge 0. \quad (23)$$

Proof. From the definition of in (12),

$$\begin{aligned} \|y_{k-1}\| &= \left\|J(x_k)^T J(x_k) s_{k-1} + \left(J(x_k) - J(x_{k-1})\right)^T r(x_k)\right\| \\ &\le \|J(x_k)^T J(x_k) s_{k-1}\| + \left\|\left(J(x_k) - J(x_{k-1})\right)^T r(x_k)\right\| \\ &\le \|J(x_k)\|^2 \|s_{k-1}\| + \|J(x_k) - J(x_{k-1})\| \|r(x_k)\| \end{aligned}$$

By using the property of matrix norm,

$$\begin{aligned} &\le b_2^2 \|s_{k-1}\| + a_1 \|x_k - x_{k-1}\| \|r(x_k)\| \\ &\le b_2^2 \|s_{k-1}\| + a_1 b_1 \|s_{k-1}\| \\ &= (b_2^2 + a_1 b_1) \|s_{k-1}\|. \end{aligned}$$

Hence, by setting $\phi := b_2^2 + a_1 b_1$, the inequality (23) is true.

In the subsequent Lemma, we show that the new algorithm satisfies the sufficient descent condition.

Lemma 2. Suppose that the search direction d_k and the sequence $\{x_k\}$ were generated via Algorithm 1. Let ω be a positive constant. Then, for each $k \ge 0$, the following inequality holds.

$$g_k^T d_k \le -\omega \|g_k\|^2.$$

Proof. Now, in accordance with the strong Wolfe criterion (18), it holds that:

$$\left|g_k^T d_{k-1}\right| \leq -\sigma g_{k-1}^T d_{k-1}, \text{ thus}$$

$$d_{k-1}^T y_{k-1} = d_{k-1}^T g_k - d_{k-1}^T g_{k-1} \geq -(1-\sigma) d_{k-1}^T g_{k-1},$$

Therefore,

$$\frac{\left|g_k^T d_{k-1}\right|}{\left|d_{k-1}^T y_{k-1}\right|} \leq \frac{\sigma}{1-\sigma}.$$

In order to show the inequality holds, if the direction d_k is given by Equation 16,

$$g_k^T d_k = g_k^T\left(-\theta_k g_k + \beta_k^{DY} d_{k-1}\right),$$

$$g_k^T d_k = -\left(\frac{\|g_k\|^2}{g_k^T y_{k-1}} + \frac{t g_k^T s_{k-1}}{g_k^T y_{k-1}}\right)\|g_k\|^2 + \frac{\|g_k\|^2}{d_{k-1}^T y_{k-1}} g_k^T d_{k-1}.$$

$$= -\frac{t g_k^T s_{k-1}}{g_k^T y_{k-1}}\|g_k\|^2 - \frac{\|g_k\|^2}{g_k^T y_{k-1}}\|g_k\|^2 + \frac{\|g_k\|^2}{d_{k-1}^T y_{k-1}} g_k^T d_{k-1}$$

Since, $\|g_k\|^2$ and $g_k{}^T y_{k-1}$ are greater than zero, then

$$\leq -\frac{t g_k^T s_{k-1}}{g_k^T y_{k-1}}\|g_k\|^2 + \frac{\|g_k\|^2}{d_{k-1}^T y_{k-1}} g_k^T d_{k-1}$$

$$\leq -\frac{t\alpha_k \left|g_k^T d_{k-1}\right|}{\left|g_k^T y_{k-1}\right|}\|g_k\|^2 + \frac{\|g_k\|^2}{\left|d_{k-1}^T y_{k-1}\right|}\left|g_k^T d_{k-1}\right|$$

$$\leq -\left(\frac{\sigma t\alpha_k}{1-\sigma} - \frac{\sigma}{1-\sigma}\right)\|g_k\|^2.$$

By letting $\omega = \left(\frac{\sigma t\alpha_k}{1-\sigma} - \frac{\sigma}{1-\sigma}\right)$, therefore, for all $k > 0$, it holds that $g_k^T d_k \leq -\omega\|g_k\|^2$ and the proof is then completed.

Lemma 3. Suppose that the sequence x_k is generated by Equation 2; then, the search direction in (13) fulfils the conjugacy condition given by:

$$d_k^T G_k d_{k-1} = d_k^T y_{k-1} = -t g_k^T s_{k-1}, \quad t > 0. \qquad (24)$$

Proof. By multiplying both sides of equation (16) by $y^T{}_{k-1}$, will obtain as:

$$d_k^T y_{k-1} = -\left(\frac{\|g_k\|^2}{g_k^T y_{k-1}} + \frac{t g_k^T s_{k-1}}{g_k^T y_{k-1}}\right) g_k^T y_{k-1} + \frac{\|g_k\|^2}{d_{k-1}^T y_{k-1}} d_{k-1}^T y_{k-1}$$

$$= -\frac{\|g_k\|^2}{g_k^T y_{k-1}} g_k^T y_{k-1} - \frac{t g_k^T s_{k-1}}{g_k^T y_{k-1}} g_k^T y_{k-1} + \frac{\|g_k\|^2}{d_{k-1}^T y_{k-1}} d_{k-1}^T y_{k-1}$$

$$= -\|g_k\|^2 - t g_k^T s_{k-1} + \|g_k\|^2$$

This implies that,

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

Hence, the proof is completed.

Lemma 4. Given that assumptions A1 and A2 are satisfied, let us consider any conjugate gradient (CG) method in the form of Equation 13, where d_k is a descent direction and α_k adheres to the exact minimisation rule. Under these circumstances, the Zoutendijk condition [17] is fulfilled:

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

Theorem 1. If Assumption *1is* satisfied and the sequence of iterations $\{d_k\}$ is generated by the SSDY algorithm, then:

$$\lim_{k\to\infty} \inf\|g_k\| = 0. \qquad (25)$$

Proof. From (15), given that:

$$|\theta_k| = \left|\frac{\|g_k\|^2}{g_k^T y_{k-1}} + \frac{t g_k^T s_{k-1}}{g_k^T y_{k-1}}\right|$$

$$\leq \frac{\|g_k\|^2}{\|g_k\|\|y_{k-1}\|} \leq \frac{\|g_k\|}{\|y_{k-1}\|} = \bar{\theta}$$

And from (13) and (15), given as:

$$\|d_k\| = \left\|-\theta_k g_k + \beta_k^{DY} d_{k-1}\right\|,$$

$$\leq |\theta_k|\|g_k\| + \left|\beta_k^{DY}\right|\|d_{k-1}\|$$

$$\leq \bar{\theta}\|g_k\| + \frac{\|g_k\|^2}{\left|d_{k-1}^T y_{k-1}\right|}\|d_{k-1}\|$$

$$\leq \left(\bar{\theta} + \frac{\|g_k\|}{\|y_{k-1}\|}\right)\|g_k\|$$

$$= 2\bar{\theta}\|g_k\|$$

Hence, provided by:

$$\frac{1}{\|d_k\|} \geq \frac{1}{\zeta}\|g_k\| \qquad (26)$$

where $\zeta := 2\bar{\theta}$, considering (25) and (26), will obtain:

$$\sum_{k=0}^{\infty}\|g_k\|^2 \leq \zeta^2 \sum_{k=0}^{\infty} \frac{\|g_k\|^4}{\|d_k\|^2} < \infty.$$

Hence, the proof is completed.

NUMERICAL RESULTS

This part presents the findings from the numerical experiment that evaluated the performance of the new proposed method, the SSDY algorithm, which is intended to solve NLS problems. To assess the effectiveness of the proposed scheme, we conducted a comparison against DY [9] and SSHS [14]. The set of test problems for this experiment includes five large-scale benchmark NLS functions, as detailed below. Each problem was considered at five different dimensions: 3000, 6000, 9000, 12000, and 15000. The experiment was carried out using MATLAB R2022a on a PC equipped with an Intel CORE-i7-3537U processor clocked at 2.00 GHz and 8 GB of RAM.

Furthermore, throughout the execution, the termination condition $||g_k|| \leq 10^{-4}$ was utilised. If any of the subsequent conditions arose, denoted by 'F', failure was noted:

a. Exceeding 1000 iterations.
b. Exceeding 5000 function evaluations.

To evaluate the efficacy of the SSDY method, we analyse the following metrics: the number of function evaluations (Feval), the number of iterations (ITER), and the CPU time. Tables 1 to 5 present the experimental results for these metrics.

Problem 1: Penalty function 1 [18].

$$F(x) = \sum_{k=1}^{m} f_k^2(x), \quad f_k(x) = \frac{1}{\sqrt{100000}}(x_k - 1), 1 \leq k \leq n,$$

$$f_k(x) = \sum_{i=1}^{n} x_i^2 - \frac{1}{4}, k = n+1, m = n+1, \bar{x}_L, L \geq 1.$$

Initial Point: $x_0 = (3, 3, ..., 3)^T$.

Problem 2: Variably Dimensionsed Function [19].

$$F(x) = \sum_{k=1}^{m} f_k^2(x), \quad f_k(x) = (x_k - 1), 1 \leq k \leq n,$$

$$f_k(x) = \sum_{i=1}^{n} i(x_i - 1), k = n+1, \qquad f_k(x) = \left(\sum_{i=1}^{n} i(x_i - 1)\right)^2,$$

Initial Point: $x_0 = \left(1 - \frac{1}{n}, 1 - \frac{2}{n}, \ldots, 0\right)^T$

Problem 3: Trigonometric Function [20].

$$F_i(x) = n - \sum_{j=1}^{n} \cos(x_j) + i(\cos(x_i) - \sin(x_i)),$$

Initial Point: $x_0 = \left(\frac{1}{n}, \frac{1}{n}, \ldots, \frac{1}{n}\right)^T$

Problem 4: Discrete Boundary Value Function [21].

$$F_i(x) = 2x_i - x_{i-1} + x_{i+1} + h^2(x_i + t_i - 1)^{\frac{3}{2}},$$

Initial Point: $x_0 = \left(\frac{1}{n+1} - 1, \frac{1}{n+1} - 1, \ldots, \frac{1}{n+1} - 1\right)^T$

Problem 5: Linear Full Rank Function [21]

$$F_i(x) = x_i - \frac{2}{m}\left(\sum_{j=1}^{n} x_j\right) - 1, \qquad 1 \leq i \leq n.$$

Initial Point: $x_0 = (1, 1, \ldots, 1)^T$

Analysing the results presented in Tables 1 to 5 sheds light on the competitiveness of each algorithm. Particularly, Algorithm DY faced challenges in tackling Problems 1 and Problem 2, whereas SSHS encountered difficulties with Problem 2. The newly introduced spectral algorithm SSDY effectively solved all five

Table 1 Numerical results based on Problem 1

Function	Dim.	DY			SSHS			SSDY		
		ITER	Feval	CPU	ITER	Feval	CPU	ITER	Feval	CPU
Problem 1	**3000**	F	F	F	4	8	0.1763	2	5	0.2012
	6000	F	F	F	4	8	0.6742	2	5	0.1239
	9000	F	F	F	4	8	0.4832	2	5	0.1698
	12000	F	F	F	4	8	0.4128	2	5	0.2875
	15000	F	F	F	4	8	0.3442	2	5	0.1995

functions considered in this numerical test. The results, as described in Tables 1 to 5, are visually summarised in Figures 1 to 3, utilising the performance profile developed by Dolan and Moré's, (see [22] for details). When examining ITER, FEVAL, and CPU, these figures demonstrate that the new algorithm SSDY surpasses both DY and SSHS algorithms, respectively. Evidently, the efficacy and efficiency of the SSDY algorithm are notable.

Table 2 Numerical results based on Problem 2

Function	Dim.	DY			SSHS			SSDY		
		ITER	Feval	CPU	ITER	Feval	CPU	ITER	Feval	CPU
Problem 2	**3000**	F	F	F	F	F	F	15	82	0.2485
	6000	F	F	F	F	F	F	16	135	0.7239
	9000	F	F	F	F	F	F	22	139	0.0698
	12000	F	F	F	F	F	F	32	195	0.2875
	15000	F	F	F	F	F	F	70	316	1.6995

Table 3 Numerical results based on Problem 3

Function	Dim.	DY			SSHS			SSDY		
		ITER	Feval	CPU	ITER	Feval	CPU	ITER	Feval	CPU
Problem 3	**3000**	73	1215	1.6598	50	446	0.4861	45	418	0.4324
	6000	94	1591	3.7368	60	578	1.2966	53	524	1.0266
	9000	87	1584	4.9902	63	653	1.8607	61	858	2.1456
	12000	106	2148	6.5381	66	765	4.1457	67	900	2.8439
	15000	108	2177	7.0331	84	1454	4.2474	73	943	2.6230

Table 4 Numerical results based on Problem 4

Function	Dim.	DY			SSHS			SSDY		
		ITER	Feval	CPU	ITER	Feval	CPU	ITER	Feval	CPU
Problem 4	**3000**	15	31	0.06464	5	13	0.02753	4	9	0.0081
	6000	15	31	0.08327	5	13	0.04395	4	9	0.0222
	9000	15	31	0.15834	5	13	0.05096	4	9	0.0261
	12000	15	31	0.20032	5	13	0.12952	4	9	0.0331
	15000	15	31	0.35541	5	13	0.16779	4	9	0.0484

Table 5 Numerical results based on Problem 5

Function	Dim.	DY			SSHS			SSDY		
		ITER	Feval	CPU	ITER	Feval	CPU	ITER	Feval	CPU
Problem 5	**3000**	34	240	0.7437	24	171	0.5712	6	43	0.0174
	6000	54	272	0.5432	29	227	0.1780	7	44	0.0767
	9000	69	315	0.5837	29	230	0.5215	9	49	0.0387
	12000	211	557	1.0062	32	241	0.1780	9	56	0.3549
	15000	212	597	2.7422	34	273	0.1112	11	66	0.3608

This ratio entails comparing the performance of a solver, denoted as "s," on a specific problem "p" with the superior results achieved by any solver for the same problem.

$$r_{p,s} = \frac{\tau_{p,s}}{min\{\tau_{p,s}: s \in S\}} \quad (27)$$

While the performance of solvers on a specific problem is of interest, the primary focus is on obtaining a comprehensive evaluation summary of the solver's overall performance. This evaluation is defined as:

$$\rho_s(\tau) = \frac{1}{n_p} size\{p \in P: r_{p,s} \leq \tau\} \quad (28)$$

Furthermore, the most superior solver demonstrates elevated values of $\rho(\tau)$ and is identifiable in the top right corner of the diagram.

The illustrations in Figures 1 through 3 were utilised to make comparisons. Upon examination of Figure 1, it was observed that when τ is equal to or greater than 2, SSDY resolves approximately 95% of the test functions with fewer iterations. In contrast, DY and SSHS address roughly 61% and 89% of the problems, respectively. In Figure 2, for τ is equal to or greater than 2, SSDY deals with nearly 98% of the test problems with the minimum number of function evaluations, compared to approximately 68%, and 92% handled by DY, and SSHS, respectively. Furthermore, concerning CPU-time, as depicted in Figure 3, especially when τ is equal to or greater than 2, SSDY exhibits efficiency by resolving about 98% of the test problems in the briefest duration, in contrast to nearly 64% and 94% tackled by DY, and SSHS, respectively. A notable observation from these figures is that, while all algorithms initially competed on equal footing, the new technique consistently outperformed the others across all metrics as τ increased. This highlights the robustness and competitiveness of the new spectral technique for the problems under consideration.

CONCLUSION

This research introduces new variants of structured spectral methods that utilise a structured vector equation to approximate the second-order component of the Hessian for solving NLS problems. This strategy not only provides a direct approach but also yields insightful outcomes, unlike the Gauss-Newton method,

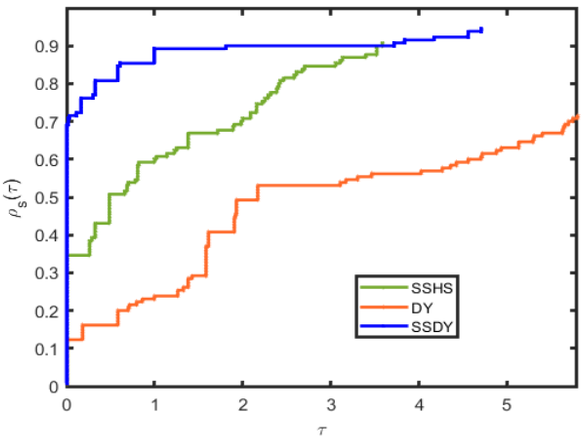

Figure 1 Numerical results based on number of iteration

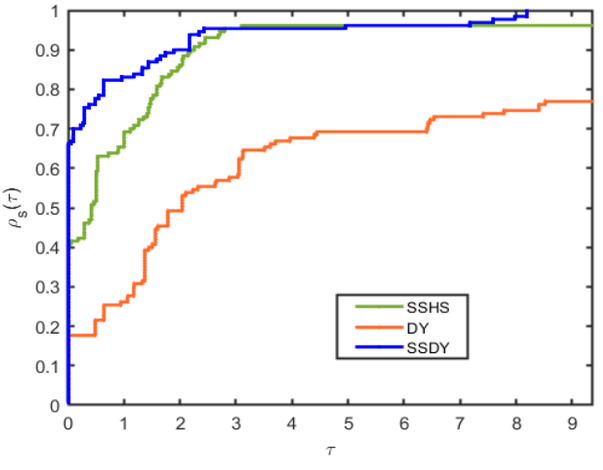

Figure 2 Numerical results based on the number of function evaluation

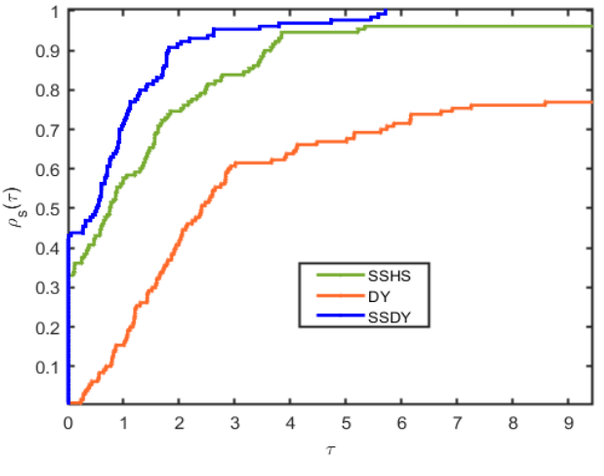

Figure 3 Numerical results based on CPU time

which disregards the second-order term entirely. This article establishes the global convergence of the algorithm under suitable assumptions on the objective function. A sequence of numerical experiments is conducted to illustrate the robustness and efficiency of the proposed methods. Based on the results, the SSDY approach outperforms both the DY and SSHS techniques across all three measures utilised in the numerical comparison. Further research could explore the extension of this method to other structured nonlinear systems of equations.

ACKNOWLEDGMENT

The authors would like to extend their sincere gratitude to Universiti Teknologi PETRONAS for its steadfast support.

REFERENCES

[1] M. M. Yahaya, P. Kumam, A. M. Awwal, P. Chaipunya, S. Aji, and S. Salisu, "A New Generalized Quasi-Newton Algorithm Based on Structured Diagonal Hessian Approximation for Solving Nonlinear Least-Squares Problems with Application to 3DOF Planar Robot Arm Manipulator," *IEEE Access*, vol. 10, pp. 10816–10826, 2022.

[2] R. B. Yunus, N. Zainuddin, H. Daud, R. Kannan, M. Muhammad Yahaya, and S. Ariffin Abdul Karim, "New CG-Based Algorithms With Second-Order Curvature Information for NLS Problems and a 4DOF Arm Robot Model," in *IEEE Access*, vol. 12, pp. 61086-61103, 2024, doi: 10.1109/ACCESS.2024.3393555.

[3] I. M. Sulaiman, M. Malik, A. M. Awwal, P. Kumam, M. Mamat, and S. Al-Ahmad, "On three-term conjugate gradient method for optimisation problems with applications on COVID-19 model and robotic motion control," *Adv. Contin. Discrete Models*, vol. 2022, no. 1, pp. 1–22, 2022.

[4] R. B. Yunus, K. Kamfa, S. I. Mohammed, and 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,* vol. 444, 2022.

[5] N. Salihu, P. Kumam, I. M. Sulaiman, and T. Seangwattana, "An efficient spectral minimisation of the Dai Yuan method with application to image reconstruction," *AIMS Mathematics*, vol. 8, no. 12, pp. 30940–30962, 2023.

[6] R. B. Yunus, M. Mamat, M. Rivaie, Z. Salleh, and Z. A. Zakaria, "The Convergence Properties of a New Kind of Conjugate Gradient Method for Unconstrained Optimization," *Appl. Math. Sciences*, vol. 9, no. 38, pp. 1845-1856, 2015.

[7] R. Fletcher and C. Reeves, "Function minimisation by conjugate gradients," *Comput. J.*, vol. 7, pp. 149–154, 1964.

[8] R. Fletcher, *Practical Methods of Optimisation, vol. 1: Unconstrained Optimisation*, John Wiley & Sons, New York, 1987.

[9] Y. H. Dai and Y. Yuan, "A nonlinear conjugate gradient method with a strong global convergence property," *SIAM J. Optim.*, vol. 10, pp. 177–182, 1999.

[10] W. W. Hager and H. Zhang, "A survey of nonlinear conjugate gradient methods," *Pacific Journal of Optimization*, vol. 2, no. 1, pp. 35-58, 2006.

[11] H. Mohammad, M. Y. Waziri, and S. A. Santos, "A Brief Survey of Methods for Solving Nonlinear Least-Squares Problems," *Numerical Algebra, Control and Optimization*, vol. 9, pp. 1-13, 2019.

[12] M. Kobayashi, Y. Narushima, and H. Yabe, "Nonlinear conjugate gradient methods with structured secant condition for nonlinear least squares problems," *Journal of Computational and Applied Mathematics*, vol. 234, pp. 375–397, 2010.

[13] R. Dehghani and N. Mahdavi-Amiri, "Scaled nonlinear conjugate gradient methods for nonlinear least squares problems," *Numerical Algorithms*, vol. 82, pp. 1-20, 2018.

[14] R. B. Yunus, N. Zainuddin, H. Daud, R. Kannan, S. A. Abdul Karim, and M. M. Yahaya, "A Modified Structured Spectral HS Method for Nonlinear Least Squares Problems and Applications in Robot Arm Control," *Mathematics*, vol. 11, no. 14, p. 3215, 2023.

[15] Y. Dai and L. Liao, "New conjugacy conditions and related nonlinear conjugate gradient methods," *Appl. Math. Optim.*, vol. 43, pp. 87–101, 2001.

[16] M. M. Yahaya, P. Kumam, A. M. Awwal, and S. Aji, "Alternative structured spectral gradient algorithms

for solving nonlinear least-squares problems," *Heliyon*, vol. 7, no. 7, pp. e07456, Jul. 2021.

[17] G. Zoutendijk, "Nonlinear programming computational methods," in *Integer and Nonlinear Programming*, J. Abadie, Ed., North-Holland, Amsterdam, pp. 37–86, 1970.

[18] A. A. Mustafa, "New spectral LS conjugate gradient method for nonlinear unconstrained optimisation," *International Journal of Computer Mathematics*, vol. 100, no. 4, pp. 838-846, 2023.

[19] M. Jamil, M. Momin, and X.-S. Yang, "A literature survey of benchmark functions for global optimisation problems," *Int. J. Math. Model. Numer. Optim.*, vol. 4, no. 2, pp. 150-194, 2013.

[20] L. Lukšan and J. Vlcek, "Test problems for unconstrained optimisation," Academy of Sciences of the Czech Republic, Institute of Computer Science, Technical Report 897, 2003.

[21] J. J. Moré, B. S. Garbow, and K. E. Hillstrom, "Testing unconstrained optimisation software," *ACM Transactions on Mathematical Software (TOMS)*, vol. 7, no. 1, pp. 17–41, 1981.

[22] S. A. Sofi, M. Mamat, S. Z. Mohid, M. A. Ibrahim, and N. Khalid, "Performance profile comparison using MATLAB," in *Proceedings of International Conference on Information Technology & Society 2015*, 2015.