Thai Journal of Mathematics
Volume 19 Number 3 (2021)
Pages 924–941

http://thaijmath.in.cmu.ac.th

ISSN 1686-0209

A New Algorithm with Structured Diagonal Hessian Approximation for Solving Nonlinear Least Squares Problems and Application to Robotic Motion Control

Aliyu Muhammed Awwal[1,2,3,*], Hassan Mohammad[3,4], Mahmoud Muhamad Yahaya[1], Ahmadu Bappah Muhammadu[2] and Adamu Ishaku[2]

[1] *KMUTT-Fixed Point Theory and Applications Research Group, Department of Mathematics, Room SCL 802, Fixed Point Laboratory, Science Laboratory Building, Faculty of Science, King Mongkut's University of Technology Thonburi (KMUTT), 126 Pracha-Uthit Road, Bang Mod, Thung Khru, Bangkok 10140, Thailand*
e-mail : aliyumagsu@gmail.com (A.M. Awwal); mahmoudpd@gmail.com (M.M. Yahaya)

[2] *Department of Mathematics, Faculty of Science, Gombe State University, Gombe, Nigeria*
e-mail : abbappah@yahoo.com (A.B. Muhammadu); adammaths@gmail.com (A. Ishaku)

[3] *Numerical Optimization Research Group, Bayero University, Kano, Nigeria*

[4] *Department of Mathematical Sciences, Faculty of Physical Sciences, Bayero University, Kano, Nigeria*
e-mail : hmuhd.mth@buk.edu.ng (H. Mohammad)

Abstract Iterative methods for solving nonlinear least-squares problems are of great interest because of their unique structures and wide range of applications. This paper exploits the unique structure and proposes a quasi-Newton diagonal-based approximation for solving this class of problems. Under some appropriate conditions, the search direction satisfies the sufficiently descent condition, and the new method is shown to be globally convergent. The numerical efficiency of the new method is tested on some standard benchmark test problems. Subsequently, the new method is implemented to solve 2 dimensional robotic motion control problem.

MSC: 90C30; 90C06; 90C56

Keywords: nonlinear least-squares problems; large-scale problems; Jacobian-free strategy; global convergence

Submission date: 27.04.2021 / Acceptance date: 11.07.2021

1. Introduction

In this paper, we are concerned with iterative methods for solving the following nonlinear least squares (NLS) problem

$$\min_{x\in\mathbb{R}^n} f(x),\ \ f(x) = \frac{1}{2}\|F(x)\|^2, \tag{1.1}$$

*Corresponding author.

where $F(x) = (F_1(x), F_2(x), \dots, F_m(x))^T$ and each residual $F_i : \mathbb{R}^n \to \mathbb{R}$, $i = 1, 2, \dots, m$, is assumed to be twice continuously differentiable. The problem (1.1) is a special class of general unconstrained optimization problems. Let the Jacobian of the vector-valued function $F : \mathbb{R}^n \to \mathbb{R}^m$ $(m \geq n)$ be denoted by $J(x)$ at $x \in \mathbb{R}^n$. The gradient $\nabla f(x)$ and Hessian $\nabla^2 f(x)$ of problem (1.1) are respectively denoted by $g(x)$ and $H(x)$ and are given by

$$g(x) := \sum_{i=1}^{m} F_i(x) \nabla F_i(x) = J(x)^T F(x), \tag{1.2}$$

$$H(x) := U(x) + V(x), \tag{1.3}$$

where $U(x) = J(x)^T J(x)$, $V(x) := \sum_{i=1}^{m} F_i(x) \nabla^2 F_i(x)$, the real-valued function F_i represents the ith component of the vector-valued function F and $\nabla^2 F_i(x)$ is its Hessian.

NLS problems have been of great interest to many researchers due to the special structure of their gradient and Hessian. A number of iterative methods for solving them have been developed and can be categorized into two, namely: (i) general unconstrained optimization methods that includes Newton's method and quasi-Newton methods; and (ii) special methods: these methods utilize the special structure of the problem, example of which includes Gauss-Newton method, Levenberg-Marquardt method and structured quasi-Newton methods (see, [1–5]). Detail of these methods can be found in the following surveys [6, 7]. On the other hand, the study of efficient method for NLS problems is important due to its appearance in many applications such as robotic motion control, data fitting, optimal control, parameter estimation, experimental design, data assimilation, and imaging problems (see, [8–22]). For example, the values for the parameters that best match the model of a given data set can be determined by minimizing the sum of the squares of the residuals.

It is a known fact that the nice structures of the gradient and Hessian of problem (1.1) are usually utilized to come up with efficient methods. For instance, based on a modified secant equation and the formula given in (1.2) and (1.3), Kobayashi et al. [23] proposed a class of conjugate gradient-like structured methods for solving problems in the form of (1.1). Also, Dehghani and Mahdavi-Amiri [24] proposed another conjugate gradient-like method for solving problem (1.1). They incorporated a modified secant equation into the method so as to get as much information of the Hessian of problem (1.1) as possible. For more on recently developed matrix-free structured methods, the interested reader may refer to the following references [25–27].

Motivated by the success of the work presented in [25], Mohammad and Santos [28] proposed a diagonal Hessian approximation method for solving problem (1.1). The entries of the diagonal matrix are calculated using some structured vectors derived in such a way that the secant condition is approximately satisfied. Their method works well and is shown to be numerically efficient. However, the authors commented on the need to further investigate better approximations of the Hessian matrix that utilize the special structure of problem (1.1). In this paper, we present a new method for solving problem (1.1) based on a structured diagonal matrix that approximate the Hessian of the objective function. Furthermore, we incorporate some diagonal correction matrix into the formula for calculating the entries of the diagonal matrix. We highlight some contributions of the propose method below:

- In building the structured diagonal matrix approximation of our method, we incorporated some diagonal correction matrix which will be updated as the iteration process progresses. By this, we expect our diagonal matrix to contain more information of the objective function than the one proposed in [28].
- Our proposed method generates descent directions and is globally convergent.
- We implement the algorithm of the new method without the need to explicitly compute any matrix throughout the iteration process.
- We present numerical experiments to demonstrate the efficiency of the new method.
- We successfully apply the new method to solve 2D robotic motion control problem.

We organized the remainder of this paper as follows. We present the proposed method and its algorithm in the next section and discuss the convergence analysis in Section 3. We give numerical experiments and subsequently apply the algorithm to the robotic motion control problem in Section 4. Throughout this article, we denote the Euclidean norm of vectors and the induced 2-norm of matrices by $\|\cdot\|$.

2. Proposed method

An important concept of a structured quasi-Newton method for nonlinear least squares is the structure principle [29]. Now, let $k = 1,\dots,n$, and suppose that at certain iteration, say $k-1$, the first term of the Hessian, (1.3), is given

$$U(x_{k-1}) = \sum_{i=1}^{m} \nabla F_i(x_{k-1})\nabla F_i(x_{k-1})^T. \tag{2.1}$$

Setting $s_{k-1} = x_k - x_{k-1}$ and applying Taylor's expansion on $F(x_{k-1})$, we have

$$F_i(x_{k-1}) = F_i(x_k) - \nabla F_i(x_k)^T s_{k-1} + o(\|s_{k-1}\|), \tag{2.2}$$

where $F_i(x)$ is the i^{th} component of the vector-valued $F(x)$ and $o : [0,+\infty) \to \mathbb{R}$ such that $\lim\limits_{e\to 0} \frac{o(e)}{e} = 0$.
Truncating the last term of (2.2) and then multiply it by $\nabla F_i(x_k)$ from the left, we have

$$\nabla F_i(x_k)[F_i(x_k) - F_i(x_{k-1})] \approx \nabla F_i(x_k)\nabla F_i(x_k)^T s_{k-1}. \tag{2.3}$$

Summing both sides of (2.3) for $i = 1,\dots,m$, we have

$$\sum_{i=1}^{m} \nabla F_i(x_k)[F_i(x_k) - F_i(x_{k-1})] \approx \sum_{i=1}^{m} \nabla F_i(x_k)\nabla F_i(x_k)^T s_{k-1}. \tag{2.4}$$

This means that the updating matrix $U(x_k)$ of (2.1) satisfies

$$U(x_k)s_{k-1} \approx \widehat{y}_{k-1}, \text{ where } \widehat{y}_{k-1} = J_k^T(F(x_k) - F(x_{k-1})),\ J_k = J(x_k). \tag{2.5}$$

On the other hand, to approximate the second term $V(x)$ of (1.3) at iteration k, we also apply the Taylor expansion on $\nabla F_i(x_k)$ and the result is

$$\nabla^2 F_i(x_k)s_{k-1} = \nabla F_i(x_k) - \nabla F_i(x_{k-1}) + \overline{o}(\|s_{k-1}\|), \tag{2.6}$$

where for each $i = 1,\dots,m$, we have $\overline{o} : [0,+\infty) \to \mathbb{R}^n$ satisfying $\lim\limits_{e\to 0} \frac{\overline{o}^i(e)}{e} = 0$.
Again, truncating the last term of (2.6) and then multiply it with $F_i(x_k)$ on the left yields

$$F_i(x_k)\nabla^2 F_i(x_k)s_{k-1} \approx F_i(x_k)[\nabla F_i(x_k) - \nabla F_i(x_{k-1})]. \tag{2.7}$$

Also, summing both sides of (2.7), for $i = 1, \dots, m$, gives

$$\sum_{i=1}^{m} F_i(x_k)\nabla^2 F_i(x_k)s_{k-1} \approx \sum_{i=1}^{m} F_i(x_k)[\nabla F_i(x_k) - \nabla F_i(x_{k-1})]. \tag{2.8}$$

This gives

$$V(x_k)s_{k-1} \approx \overline{y}_{k-1}, \text{ where } \overline{y}_{k-1} = (J_k - J_{k-1})^T F_k. \tag{2.9}$$

Let $D_k = D(x_k)$, $U_k = U(x_k)$ and $V_k = V(x_k)$. Suppose that the sum of the two square matrices $U(x_k)$ and $V(x_k)$ is approximated with a diagonal matrix $D_k \approx U_k + V_k$, such that the weak secant is satisfied, i.e.,

$$s_{k-1}^T D_k s_{k-1} = s_{k-1}^T y_{k-1}, \tag{2.10}$$

where

$$y_{k-1} = \widehat{y}_{k-1} + \overline{y}_{k-1} = J_k^T(F(x_k) - F(x_{k-1})) + (J_k - J_{k-1})^T F_k. \tag{2.11}$$

Now, let $D_0 = I$, where I is an identity matrix, such that the diagonal matrix D_k can be updated as

$$D_k = D_{k-1} + \Omega_{k-1}, \ k \geq 1, \tag{2.12}$$

where Ω_{k-1} is a diagonal matrix. Observe that Ω_{k-1} is the deviation between D_{k-1} and D_k. The diagonal matrix Ω_{k-1} is a correction matrix of D_{k-1} such that D_k satisfies weak secant equation [30]. The update D_k will be constructed from D_{k-1} in such a way that a constrained minimization problem involving $\|D_k - D_{k-1}\|$ and the trace of $(D_{k-1} + \Omega_{k-1})$ is solved. The reason for integrating the trace of $(D_{k-1} + \Omega_{k-1})$ into the optimization problem is to obtain a correction matrix Ω_{k-1} which clusters the eigenvalues of D_k such that its condition number improves, subject to the constraint condition $s_{k-1}^T D_k s_{k-1} = s_{k-1}^T y_{k-1}$, with y_{k-1} given as in (2.11). The formula for calculating the entries of the diagonal correction matrix Ω_{k-1} is given in the following Lemma. The proof of the Lemma can be obtained in [31].

Lemma 2.1. *Let D_{k-1} be a diagonal matrix, with diagonal entries d_{k-1}^i, $i = 1, 2, \dots, n$. Then the diagonal matrix Ω_{k-1}, with entries ω_{k-1}^i, in the following constrained minimization problem*

$$\min_{\Omega} \frac{1}{2}\|\Omega_{k-1}\|_F^2 + tr(D_{k-1} + \Omega_{k-1}), \tag{2.13}$$

$$s.t \ s_{k-1}^T(D_{k-1} + \Omega_{k-1})s_{k-1} = r_{k-1}, \tag{2.14}$$

satisfies

$$\omega_{k-1}^i = \frac{(s_{k-1}^T s_{k-1} - s_{k-1}^T D_{k-1} s_{k-1} + r_{k-1})(s_{k-1}^i)^2}{\sum_{i=1}^{n}(s_{k-1}^i)^4} - 1, \tag{2.15}$$

where $r_{k-1} = s_{k-1}^T y_{k-1}$ and y_{k-1} is given by (2.11). Also $tr(\cdot)$ and $\|\cdot\|_F$ represent trace of a matrix and Frobenius norm of matrix, respectively.

Let $D_0 = \text{diag}(h_0^1, h_0^2, \dots, h_0^n)$, where for each $i = 1, \dots, n$, $h_0^i = 1$. Using the Lemma 2.1, the entries of the diagonal matrix (2.12) are calculated as follows

$$h_k^i = h_{k-1}^i + \omega_{k-1}^i, \ k \geq 1, \tag{2.16}$$

where ω_{k-1}^i is computed using (2.15).

2.1. Safeguarding strategy

To make sure that the diagonal matrix D_k is positive definite, in what follows, we present a safeguarding strategy to ensure each diagonal entry is strictly positive. This can be done by simply safeguarding the diagonal entries $h_{k-1}^i + \omega_{k-1}^i$ from assuming negative values or zero. Now, let $\ell > 0$ and $0 << u << +\infty$, we redefine the entries of the diagonal matrix D_k as follows

$$h_k^i = \min\left\{\max\{h_{k-1}^i + \omega_{k-1}^i, \ \ell\}, \ u\right\}, \tag{2.17}$$

with ω_{k-1}^i given in (2.15) and $h_0^i = 1$, for each i. The main aim for introducing the parameter u is to prevent the diagonal entries from assuming extremely large values.

2.2. Algorithm of the proposed method

In this subsection, we present the algorithm of the proposed method. Let d_k and g_k be the search direction and the gradient of f at x_k, i.e. $\nabla f(x_k)$, respectively. Let $x_0 \in \mathbb{R}^n$ be an initial guess, then the next point is calculated via

$$x_{k+1} = x_k + \alpha_k d_k, \ k = 0, 1, 2, \dots. \tag{2.18}$$

The search direction, d_k in (2.18) is obtained by solving the linear systems

$$D_k d_k = -g_k,$$

$$D_k = \begin{cases} I, & \text{if } k = 0, \\ \text{diag}(h_k^1, h_k^2, \dots, h_k^n), & \text{if } k \geq 1, \end{cases} \tag{2.19}$$

where h_k^i, $i = 1, 2, \dots, n$, are computed using (2.17). We adopt the non-monotone line search of Zhang and Hager [32] to compute the step length α_k. If the search direction d_k is descent, then the step length $\alpha_k > 0$ in (2.18) should satisfy the following inequality

$$f(x_k + \alpha_k d_k) \leq P_k + \theta \alpha_k g_k^T d_k, \tag{2.20}$$

where

$$\begin{cases} P_0 = f(x_0), \\ P_{k+1} = \frac{\eta_k Q_k P_k + f(x_{k+1})}{Q_{k+1}}, \\ Q_0 = 1, \\ Q_{k+1} = \eta_k Q_k + 1, \end{cases} \tag{2.21}$$

and $\theta \in (0, 1)$, $\eta_k \in [0, 1]$. We quickly note the following:

Remark 2.2.

(i) The scalar P_{k+1} in (2.21) is a convex combination of P_k and $f(x_{k+1})$. Since $P_0 = f(x_0)$, it follows that the sequence $\{P_k\}$ is a convex combination of the function values $f(x_i)$, for $i = 0, 1, 2, \dots, k$.

(ii) The nonnegative parameter η_k commands the degree of monotonicity of the line search. If for each k, $\eta_k = 0$, then the line search is the monotone Armijo-type line search; otherwise, it is nonmonotone.

(iii) If the parameter $\eta_k = 1$, for all k, then $P_k = \psi_k$ with ψ_k given as

$$\psi_k = \frac{1}{k+1} \sum_{i=1}^{k} f(x_i). \tag{2.22}$$

Next, we formally state the algorithm of the new proposed iterative method.

Algorithm 1: New Algorithm with Structured Diagonal Hessian (NASDH)

Input : Given $x_0 \in \mathbb{R}^n$, $\theta \in (0,1)$, $0 \le \eta_{\min} \le \eta_{\max} \le 1$, $\ell > 0$, $0 << u << +\infty$, $\epsilon > 0$, and $k_{\max} \in \mathbb{N}$.

Step 1: Set $k = 0$, $D_k = I$, $Q_k = 1$. Compute F_k, g_k, $f_k = \frac{1}{2}\|F_k\|^2$ and set $P_k = f_k$.

Step 2: If $\|g_k\| \le \epsilon$ and $k \ge k_{\max}$, stop. Else compute the search direction

$$d_k = -D_k^{-1} g_k. \tag{2.23}$$

Step 3: Calculate the step length using nonmonotone line search as follows. Set $\alpha = 1$, if

$$f(x_k + \alpha d_k) \le P_k + \alpha \theta g_k^T d_k, \tag{2.24}$$

then $\alpha_k = \alpha$. Else, set $\alpha = \alpha/2$ and test (2.24) again.

Step 4: Compute the next iterate using $x_{k+1} = x_k + \alpha_k d_k$.

Step 5: Set $s_k = x_{k+1} - x_k$ and compute $y_k = \widehat{y}_k + \overline{y}_k$ using (2.11).

Step 6: Calculate the entries of Ω_k using (2.15) where $r_k = s_k^T y_k$.

Step 7: Update the entries of the diagonal Hessian approximation D_{k+1} using (2.17).

Step 8: Determine $\eta_k \in [\eta_{\min}, \eta_{\max}]$ using any suitable sequence between 0 and 1, then calculate Q_{k+1} and P_{k+1} using (2.21).

Step 9: Set $k = k+1$ and go to **step 2.**

Remark 2.3.

(i) Although the vectors g_k, $\widehat{y}_k$ and $\overline{y}_k$ in Algorithm NASDH are in the form of matrix-vector products, we implemented the algorithm as matrix-free. This implementation is done by writing MATLAB code that computes the matrix-vector products without explicitly forming any matrix throughout the iteration process.

(ii) Due to the fact that the Hessian approximation D_k is a diagonal matrix, its inverse D_k^{-1} is calculated by simply taking the reciprocal of each h_k^i $(i = 1, 2, \dots, n)$. This means that the product $D_k^{-1} g_k$ is calculated as component-wise vector multiplications.

In what follows, we discuss the convergence results of the proposed method. The following assumptions will be useful in our discussion.

Assumption 2.4.

A1. *The level set* $\Theta = \{x \in \mathbb{R}^n \mid f(x) \le f(x_0)\}$ *is bounded.*

A2. *The function $F : \mathbb{R}^n \to \mathbb{R}^m$, and its Jacobian are Lipschitzian. That is, for all $x, y \in \Theta$, there exist some positive constants $L_1 > 0$ and $L_2 >$ for which*

$$\|F(x) - F(y)\| \leq L_1\|x - y\|, \text{ and} \tag{2.25}$$
$$\|J(x) - J(y)\| \leq L_2\|x - y\|. \tag{2.26}$$

Using the inequalities (2.25) and (2.26), we have the following conclusions

$$\|g(x) - g(y)\| \leq l\|x - y\|, \ \|F(x)\| \leq \omega_1, \ \|J(x)\| \leq \omega_2 \text{ and } \|g(x)\| \leq \omega_3,$$

where $l > 0$, $\omega_1 > 0$, $\omega_2 > 0$ and $\omega_3 > 0$ are given constants.

3. Convergence Analysis

In this section, we first show that the diagonal entries of the correction matrix Ω_k given by (2.15) are bounded.

Lemma 3.1. *Let the Assumption 2.4 holds and assume $\|s_{k-1}\| \neq 0$, for all $k \geq 1$. Then the diagonal correction matrix, $\{\Omega_{k-1}\}$, generated by Algorithm NASDH, where its entries are computed by (2.15), is bounded provided the following inequality*

$$|r_{k-1}| \leq v\|s_{k-1}\|^2, \ v > 0, \tag{3.1}$$

holds.

Proof. We only need to show that (3.1) holds and the rest of the proof will follow from Lemma 2.1 in [31]. Let y_{k-1} be defined by (2.11) and $r_{k-1} = s_{k-1}^T y_{k-1}$, then

$$\begin{aligned}
|r_{k-1}| &= \left\|s_{k-1}^T \left(J_k^T(F_k - F_{k-1}) + (J_k - J_{k-1})^T F_k\right)\right\| \\
&\leq \|s_{k-1}\| \left(\left\|J_k^T(F_k - F_{k-1})\right\| + \left\|(J_k - J_{k-1})^T F_k\right\|\right) \\
&\leq \|s_{k-1}\| \left(\|J_k\|\|F_k - F_{k-1}\| + \|J_k - J_{k-1}\|\|F_k\|\right) \\
&\leq \|s_{k-1}\| \left(w_2 L_1\|s_{k-1}\| + w_1 L_2\|s_{k-1}\|\right) \\
&\leq v\|s_{k-1}\|^2,
\end{aligned}$$

where $v = w_2 L_1 + w_1 L_2$. Therefore, the inequality (3.1) holds and the conclusion of the Lemma is true. ∎

Lemma 3.2. *Suppose that the Algorithm NASDH generates the sequence of search directions $\{d_k\}$, then we can find some constants, say $m_1 > 0$, $m_2 > 0$, such that the following inequalities hold*

$$g_k^T d_k \leq -m_1\|g_k\|^2, \tag{3.2}$$
$$\|d_k\| \leq m_2\|g_k\|, \tag{3.3}$$

for all $k \geq 0$.

Proof. Let the diagonal entries of the Hessian approximation D_k be defined by (2.17). From (2.17), it holds that

$$\ell \leq \min\left\{\max\{h_{k-1}^i + \omega_{k-1}^i, \ \ell\}, \ u\right\} \leq u, \tag{3.4}$$

and therefore, by the Equation (2.23) we have

$$
\begin{aligned}
g_k^T d_k &= -g_k^T D_k^{-1} g_k \\
&= -g_k^T h_k^{-1} g_k \\
&= -\sum_{i=1}^{n} (g_k^i)^2 / \min\left\{\max\{h_{k-1}^i + \omega_{k-1}^i,\ \ell\},\ u\right\} \\
&\leq -(1/u) \sum_{i=1}^{n} (g_k^i)^2 \\
&= -m_1 \|g_k\|^2,
\end{aligned}
$$

where $m_1 = 1/u$, and (3.2) holds.
Similarly, by the fact that the diagonal matrix is symmetric, for all k, we have

$$
\begin{aligned}
\|d_k\|^2 &= g_k^T D_k^{-2} g_k \\
&= g_k^T h_k^{-2} g_k \\
&= \sum_{i=1}^{n} \left(g_k^i / \min\left\{\max\{h_{k-1}^i + \omega_{k-1}^i,\ \ell\},\ u\right\}\right)^2 \\
&\leq (1/\ell^2) \sum_{i=1}^{n} (g_k^i)^2 \\
&= m_2 \|g_k\|^2,
\end{aligned}
$$

where $m_2 = 1/\ell^2$ and hence the proof. ∎

Now, by Lemma 3.1 and 3.2 together with the Proposition 1 in [28], we conclude that the new Algorithm NASDH is well-defined. The following result comes from Lemma 1.1 of [32].

Lemma 3.3. *Suppose the sequence $\{x_k\}$ is generated by Algorithm NASDH and P_k and ψ_k are defined by (2.21) and (2.22), respectively, then for all $k = 0, 1, 2 \dots,$ it holds that $f(x_k) \leq P_k \leq \psi_k$.*

Lemma 3.4. *Suppose the sequences $\{x_k\}$ and $\{d_k\}$ are generated by Algorithm NASDH such that the relation (3.2) holds, then $f(x_k) \leq f(x_0)$, for each $k \geq 0$. That is, $\{x_k\} \subset \Theta$, where Θ is the level set.*

Proof. Let P_{k+1} be given as in Equation (2.21), then using $Q_{k+1} = \eta_k Q_k + 1$, together with (2.20) and (3.2) yield

$$\begin{aligned} P_{k+1} &= \frac{\eta_k Q_k P_k + f(x_{k+1})}{Q_{k+1}}, \\ &= \frac{(Q_{k+1} - 1)P_k + f(x_{k+1})}{Q_{k+1}}, \\ &\leq \frac{Q_{k+1}P_k + \alpha_k \theta g_k^T d_k}{Q_{k+1}}, \\ &\leq \frac{Q_{k+1}P_k - \alpha_k \theta m_1 \|g_k\|^2}{Q_{k+1}}, \\ &= P_k - \frac{\alpha_k \theta m_1 \|g_k\|^2}{Q_{k+1}} \\ &\leq P_k, \end{aligned}$$

where the last inequality holds by dropping the negative term in the preceding line. This means the sequence $\{P_{k+1}\}$ is decreasing and since $f_k \leq P_k$, (see, Lemma 3.3), we have

$$f(x_{k+1}) \leq P_{k+1} \leq P_k \leq P_{k-1} \leq \ldots \leq P_0 = f(x_0),$$

and the proof in complete. ■

The following Theorem shows that Algorithm NASDH converges. The proof of the Theorem follows from [32] and so, we omit it here.

Theorem 3.5. *Suppose that x_k is the algorithm generated by Algorithm NASDH. If the Assumption 2.4 holds, then*

$$\liminf_{k\to\infty} \|g_k\| = 0. \tag{3.5}$$

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

$$\lim_{k\to\infty} \|g_k\| = 0. \tag{3.6}$$

4. Numerical Experiments

In this section, numerical experiments are reported in order to assess the efficiency of the proposed Algorithm NASDH in comparison with Algorithm SDHAM proposed in [28]. Both algorithms, that is, NASDH and SDHAM are implemented in such a way that computation of any matrix is completely avoided. In the course of this experiment, seventeen large-scale test problems (see, Table 1) are solved using five different dimensions, i.e. 3000, 6000, 9000, 12000 and 15000. This gives a total of eighty two large-scale test problems solve for this experiment. In Table 1, Pi stands for problem i where $i = 1, 2, \ldots, 17$. The parameters used for the implementation of both algorithms are as follows

- Algorithm NASDH: $\eta_k = \frac{1}{\exp(k+1)^{(k+1)}}$ with $\eta_{\min} = 0.1$, $\eta_{\max} = 0.85$, $\ell = 10^{-30}$, $u = 10^{30}$; and $\theta = 0.00001$.
- Algorithm SDHAM: All parameters are as presented in [28].

The two algorithms are coded in MATLAB R2019b and the device used for the experiments is a PC with intel Core(TM) i5-8250u processor with 4 GB of RAM and CPU 1.60 GHZ. The iteration process is terminated whenever $\|g_k\| \leq 10^{-4}$. On the other hand, failure, denoted by "_", is recorded when the number of iterations is greater than 1,000 and the stopping criterion mentioned above has not been satisfied.

The results of the experiment are presented in Tables 2–3, where NITER, NFVAL, NMVP, TIME and FVALUE denote the number of iterations, the number of function evaluations, the number of matrix-vector product, the CPU time in seconds (TIME), and the value of the objective function f at the solution. From the information presented in Tables 2–3, it can be seen that both algorithms are competitive. Interestingly, while Algorithm SDHAM failed in some instances, the new Algorithm NASDH solves all the test problems considered in this experiment successfully. The results reported in Tables 2–3 are summarized in Figures 1–3 with the aid of the performance profile of Dolan and Moré [33]. These figures show that the new Algorithm NASDH outperforms Algorithm SDHAM in terms of NITER, NFVAL and NMVP. Putting everything together, it can be seen that the new Algorithm NASDH works well and is efficient.

TABLE 1. List of test problems with references and starting points

Problems	Function name	Starting point
Large-scale		
P1	Penalty function I [34]	$(1/3, 1/3, \dots, 1/3)^T$
P2	Trigonometric function [35]	$(1, 1, \dots, 1)^T$
P3	Discrete boundary value [35]	$(\frac{1}{n+1}(\frac{1}{n+1} - 1), \dots, \frac{1}{n+1}(\frac{n}{n+1} - 1))^T$
P4	Linear function full rank [35]	$(1, 1, \dots, 1)^T$
P5	Problem 202 [36]	$(2, 2, \dots, 2)^T$
P6	Sine function [37]	$(1, 1, \dots, 1)^T$
P7	Exponential function I [34]	$(\frac{n}{n-1}, \dots, \frac{n}{n-1})^T$
P8	Exponential function II [34]	$(\frac{1}{n^2}, \frac{1}{n^2}, \dots, \frac{1}{n^2})^T$
P9	Logarithmic function [34]	$(1, 1, \dots, 1)^T$
P10	Trigonometric Exponential System [38]	$(0, 0, \dots, 0)^T$
P11	Extended Freudenstein and Roth [34]	$(6, 3, 6, 3, \dots, 6, 3)^T$
P12	Extended Powell singular [34]	$(-1, -1, \dots, -1)^T$
P13	Broyden tridiagonal function [35]	$(-\frac{5}{4}, -\frac{5}{4}, \dots, -\frac{5}{4})^T$
P14	Extended Himmelblau [39]	$(1, 1/n, 1, 1/n, \dots, 1, 1/n)^T$
P15	Function 27 [34]	$(100, 1/n^2, 1/n^2, \dots, 1/n^2)^T$
P16	Trigonometric logarithmic function [28]	$(1, 1, \dots, 1)^T$
P17	Linear full rank I [35]	$(1, 1, \dots, 1)^T$

4.1. Application of Algorithm NASDH in motion control

In this subsection, we implement the Algorithm NASDH to solve a time-varying nonlinear optimization (TVNO) called the real-time motion control problem of a 2-D planar robotic manipulator. This problem is described as follows:

Initially, we have a discrete-time kinematic formulation of the problem at certain position level, as given in [40], is

$$f(\theta(t_k)) = \psi_k, \tag{4.1}$$

where for notational simplicity, we set $\theta_k = \theta(t_k) \in \mathbb{R}^2$ and it denotes the joint angle vector at time t_k, $k = 0, 1, 2 \dots$. The function f in (4.1) is a kinematic mapping function

Table 2. Numerical results of our NASDH and SDHAM methods for problems 1–9

		NASDH					SDHAM				
Problems	**DIM**	**NITER**	**NFEVAL**	**NMVP**	**TIME**	**FVALUE**	**NITER**	**NFEVAL**	**NMVP**	**TIME**	**FVALUE**
1	**3000**	4	9	13	0.12161	2.07E-08	11	34	34	0.039916	2.24E-05
	6000	3	7	10	0.045944	4.42E-05	17	52	52	0.057568	8.96E-05
	9000	3	7	10	0.02353	2.34E-05	22	67	67	0.11434	0.000115
	12000	3	7	10	0.017739	1.92E-08	26	79	79	0.25034	0.000133
	15000	3	7	10	0.033054	2.73E-05	29	88	88	0.34191	0.000159
2	**3000**	14	29	43	0.97561	4.98E-21	17	85	52	0.1472	7.92E-18
	6000	31	63	94	1.4646	3.42E-23	19	94	58	0.42884	7.96E-19
	9000	24	49	73	2.0649	1.35E-23	37	151	112	0.51592	5.8E-23
	12000	19	39	58	6.911	1.92E-21	41	164	124	0.83117	5.16E-23
	15000	19	39	58	4.8339	4.67E-20	39	160	118	1.4421	6.37E-25
3	**3000**	18	37	55	0.78725	6.79E-09	7	23	22	0.075409	3.46E-09
	6000	7	15	22	0.48987	2.92E-09	3	11	10	0.023303	1E-09
	9000	3	7	10	0.42246	2.11E-09	2	8	7	0.049546	1.95E-09
	12000	5	11	16	1.2967	9.07E-10	1	5	4	0.022423	1.41E-09
	15000	5	11	16	1.0007	7.18E-10	1	5	4	0.027679	9E-10
4	**3000**	2	5	7	0.036097	0.50067	2	7	7	0.010385	0.50067
	6000	2	5	7	0.03346	0.50033	2	7	7	0.005553	0.50033
	9000	2	5	7	0.039963	0.50022	2	7	7	0.011027	0.50022
	12000	2	5	7	0.060487	0.50017	2	7	7	0.013325	0.50017
	15000	2	5	7	0.074288	0.50013	2	7	7	0.020024	0.50013
5	**3000**	5	11	16	0.26367	1.39E-14	5	16	16	0.013016	4.16E-10
	6000	5	11	16	0.022962	2.78E-14	5	16	16	0.017584	8.33E-10
	9000	4	9	13	0.031193	5.4E-10	5	16	16	0.02754	1.25E-09
	12000	5	11	16	0.042908	5.56E-14	5	16	16	0.053497	1.67E-09
	15000	5	11	16	0.15259	8.64E-16	5	16	16	0.044452	2.08E-09
6	**3000**	4	9	13	0.077298	2.49E-15	4	13	13	0.007726	5.92E-10
	6000	4	9	13	0.033328	4.99E-15	4	13	13	0.017933	1.18E-09
	9000	4	9	13	0.036238	6.26E-11	4	13	13	0.043097	1.78E-09
	12000	4	9	13	0.044584	9.98E-15	4	13	13	0.026489	2.37E-09
	15000	4	9	13	0.051073	1.07E-11	4	13	13	0.041235	2.96E-09
7	**3000**	18	37	55	0.35413	4E-09	4	13	13	0.016785	6.03E-08
	6000	16	33	49	1.2676	3.27E-09	4	13	13	0.01899	3.01E-08
	9000	36	73	109	3.9131	3.3E-09	3	10	10	0.028139	8.05E-08
	12000	13	27	40	1.4672	3.13E-09	3	10	10	0.025121	6.03E-08
	15000	1	3	4	0.057628	6.8E-09	3	10	10	0.053128	4.83E-08
8	**3000**	34	69	103	0.29725	2.86E-11	2	20	7	0.011249	1.26E-12
	6000	41	83	124	0.56536	1.05E-11	2	22	7	0.029077	3.15E-13
	9000	31	63	94	0.80417	4.01E-12	2	23	7	0.029875	1.4E-13
	12000	61	123	184	1.868	3.31E-12	2	24	7	0.044134	7.87E-14
	15000	63	127	190	3.2395	2.3E-12	2	24	7	0.093146	5.04E-14
9	**3000**	4	9	13	0.17067	4.9E-13	6	19	19	0.018662	4.59E-12
	6000	4	9	13	0.019709	1.26E-12	6	19	19	0.029738	9.11E-12
	9000	6	13	19	0.03606	7.01E-16	6	19	19	0.033826	1.36E-11
	12000	4	9	13	0.036446	2.85E-12	6	19	19	0.065562	1.82E-11
	15000	5	11	16	0.052373	7.94E-10	6	19	19	0.052307	2.27E-11

which has the following structure

$$f(\theta) = \begin{bmatrix} l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) \\ \\ l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2) \end{bmatrix}, \tag{4.2}$$

where l_1 and l_2 denote the length of the first and second rod, respectively. The vector $\psi_k \in \mathbb{R}^2$ denotes the end-effector position vector. In the context of motion control, the following least squares problem

$$\min \frac{1}{2}\|\psi_{dk} - \psi_k\|^2, \tag{4.3}$$

Table 3. Numerical results of our NASDH and SDHAM methods for problems 10–17

		NASDH					SDHAM				
Problems	**DIM**	**NITER**	**NFEVAL**	**NMVP**	**TIME**	**FVALUE**	**NITER**	**NFEVAL**	**NMVP**	**TIME**	**FVALUE**
	3000	14	29	43	0.23809	5.02E-11	-	-	-	-	-
	6000	13	27	40	1.4334	2.43E-11	-	-	-	-	-
10	**9000**	17	35	52	0.63981	4.99E-11	-	-	-	-	-
	12000	13	27	40	2.1256	1.1E-11	121	477	364	2.9071	7.2135
	15000	13	27	40	0.38888	4.13E-12	115	455	346	2.9483	7.2135
	3000	34	69	103	0.18326	3.08E-11	21	72	64	0.098252	1.87E-13
	6000	34	69	103	1.0705	6.16E-11	21	72	64	0.2035	3.75E-13
11	**9000**	31	63	94	1.0035	5.47E-11	21	72	64	0.16249	5.62E-13
	12000	36	73	109	1.2625	1.38E-11	21	72	64	0.18368	7.49E-13
	15000	31	63	94	1.746	8.77E-11	21	72	64	0.19406	9.37E-13
	3000	4	9	13	0.066853	8.19E-14	1	9	4	0.01224	3.62E-12
	6000	4	9	13	0.04383	1.64E-13	1	9	4	0.006845	7.23E-12
12	**9000**	4	9	13	0.10517	4.6E-17	1	9	4	0.017783	1.08E-11
	12000	4	9	13	0.19762	3.27E-13	1	9	4	0.02624	1.45E-11
	15000	4	9	13	0.069406	5.33E-12	1	9	4	0.055609	1.81E-11
	3000	41	83	124	0.26592	8.24E-12	22	81	67	0.19368	1.88E-11
	6000	27	55	82	0.43101	1.14E-11	22	81	67	0.22723	1.88E-11
13	**9000**	33	67	100	0.94937	1.04E-11	22	81	67	0.34454	1.88E-11
	12000	23	47	70	1.0441	1.64E-11	22	81	67	0.19879	1.88E-11
	15000	21	43	64	0.55033	2.79E-12	22	81	67	0.27599	1.88E-11
	3000	16	33	49	0.04293	1.73E-10	31	101	94	0.1778	1.35E-10
	6000	21	43	64	0.054546	1.47E-11	28	89	85	0.19598	8E-11
14	**9000**	23	47	70	0.099591	4.69E-11	30	97	91	0.17613	7.61E-11
	12000	20	41	61	0.11591	8.17E-12	25	80	76	0.19452	3.55E-10
	15000	22	45	67	0.14188	1.09E-11	26	82	79	0.26135	1.12E-10
	3000	25	51	76	0.064434	2.17E-07	18	69	55	0.065117	2.16E-07
	6000	25	51	76	0.1294	2.17E-07	18	69	55	0.17474	2.16E-07
15	**9000**	25	51	76	0.1785	2.17E-07	18	69	55	0.41657	2.16E-07
	12000	25	51	76	0.43541	2.17E-07	18	69	55	0.17807	2.16E-07
	15000	25	51	76	0.81997	2.17E-07	18	69	55	0.22047	2.16E-07
	3000	4	9	13	0.062876	5.81E-13	6	19	19	0.022326	4.58E-12
	6000	4	9	13	0.036574	1.37E-12	6	19	19	0.027378	9.1E-12
16	**9000**	6	13	19	0.056545	6.88E-16	6	19	19	0.03946	1.36E-11
	12000	4	9	13	0.069589	2.97E-12	6	19	19	0.13291	1.81E-11
	15000	5	11	16	0.15831	7.97E-10	6	19	19	0.16687	2.27E-11
	3000	3	7	10	0.056737	374.8125	-	-	-	-	-
	6000	3	7	10	0.056855	749.8125	-	-	-	-	-
17	**9000**	3	7	10	0.12004	1124.813	-	-	-	-	-
	12000	3	7	10	0.065116	1499.813	-	-	-	-	-
	15000	3	7	10	0.075262	1874.813	-	-	-	-	-

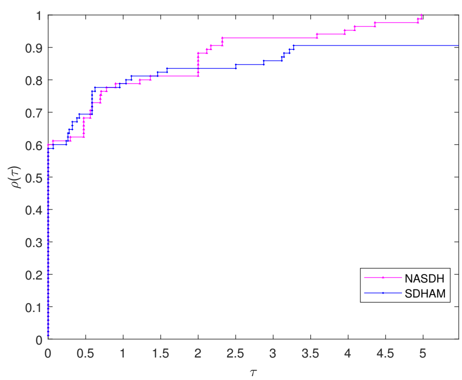

Figure 1. Performance profile with respect to NITER

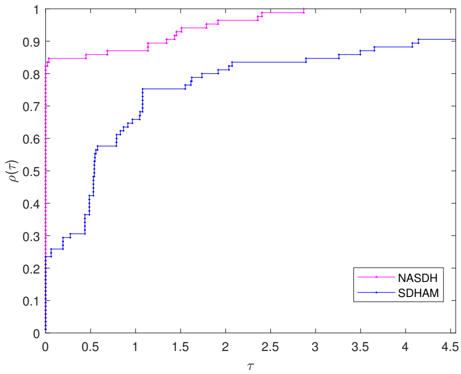

FIGURE 2. Performance profile with respect to NFEVAL

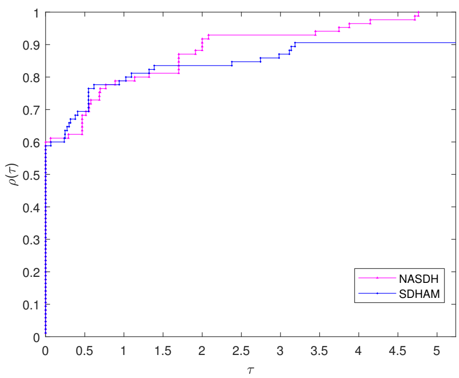

FIGURE 3. Performance profile with respect to NMVP

is solved at each computational time interval $[t_k, t_{k+1})$, where, the end-effector, ψ_k, is controlled to track the following Lissajous curve

$$\psi_{dk} = \begin{bmatrix} 1.5 + 0.2\sin(t_k) \\ \\ \frac{\sqrt{3}}{2} + 0.2\sin(2t_k) \end{bmatrix}. \tag{4.4}$$

To experimentally perform the simulation, we initialize the joint angle vector, $\theta_0 = [0, \frac{\pi}{3}]^T$ and set the rod's length as $l_i = 1$ for $i = 1, 2$ where the task duration, $t_f = 10s$ is partitioned into 200 equal parts.

It can be observed from Figure 4(a) that Algorithm NASDH effectively synthesized the robot trajectories and Figure 4(b) shows that the end effector model is fitted with the desired path that describes the trajectories of the end effector. Finally, we plot the residual errors obtained from the synthesized robot trajectories on both $x - axis$ and $y - axis$. These are reported in Figure 4(c)–4(d). It can be seen that the errors on both axes are as low as 10^{-5}. This shows the applicability of the proposed Algorithm NASDH.

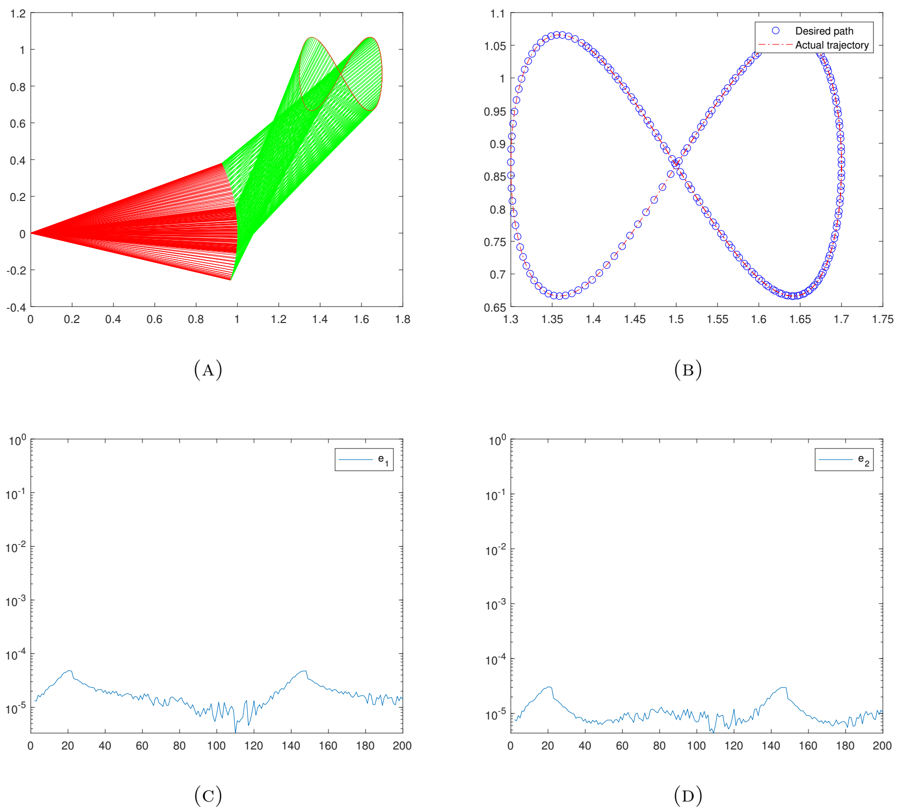

Figure 4. Numerical results generated in the course of robotic motion control experiment: (a) Robot trajectories synthesized by Algorithm NASDH. (b) End effector trajectory and desired path by Algorithm NASDH. (c) Residual error by Algorithm NASDH on x-axis. (d) Residual error by Algorithm NASDH on y-axis.

5. Conclusions

We have proposed an iterative method with an improved structured diagonal Hessian approximation for solving nonlinear least-squares problems. By incorporating some correction matrices into the sequence of our structured diagonal matrix approximation, it contains more information of the objective function than the one offered by Mohammad and Santos [28]. We coded the algorithm of the proposed method such that it neither form nor store matrices throughout the iteration process. This makes it suitable for large-scale problems. We have devised appropriate safeguards to ensure the diagonal entries are positive definite, which subsequently guaranteed that the search directions generated by the new method are sufficiently descending. We have presented numerical experiments on a collection of some benchmark test problems which showed the efficiency of the new method. Finally, we have implemented the Algorithm NASDH on problems arising from robotic motion control. Future work includes applying the new method to solve problems arising from portfolio selection and data fitting. Furthermore, it will be interesting to use the idea in this paper to solve system of nonlinear equations and some applications [41–46].

Acknowledgements

The authors thank the anonymous reviewers of their valuable comments and suggestions.

References

[1] C.X. Xu, Hybrid method for nonlinear least-square problems without calculating derivatives, Journal of optimization Theory and Applications 65 (1990) 555–574.

[2] H. Zhang, A.R. Conn, On the local convergence of a derivative-free algorithm for least-squares minimization, Computational Optimization and Applications 51 (2012) 481–507.

[3] H. Zhang, A.R. Conn, K. Scheinberg, A derivative-free algorithm for least-squares minimization, SIAM Journal on Optimization 20 (2010) 3555–3576.

[4] Y.X. Yuan, Subspace methods for large scale nonlinear equations and nonlinear least squares, Optimization and Engineering 10 (2009) 207–218.

[5] R. Fletcher, C. Xu, Hybrid methods for nonlinear least squares, IMA Journal of Numerical Analysis 7 (1987) 371–389.

[6] H. Mohammad, M.Y. Waziri, S.A. Santos, A brief survey of methods for solving nonlinear least-squares problems, Numerical Algebra, Control and Optimization 9 (2019) 1–13.

[7] Y.X. Yuan, Recent advances in numerical methods for nonlinear equations and nonlinear least squares, Numerical Algebra, Control and Optimization 1 (2011) 15–34.

[8] G. Golub and V. Pereyra, Separable nonlinear least squares: the variable projection method and its applications, Inverse Problems 19 (2003) R1.

[9] A. Padcharoen, P. Kumam, J. Martínez-Moreno, Augmented Lagrangian method for TV-l_1-l_2 based colour image restoration, Journal of Computational and Applied Mathematics 354 (2019) 507–519.

[10] A. Padcharoen, D. Kitkuan, P. Kumam, J. Rilwan, W. Kumam, Accelerated alternating minimization algorithm for Poisson noisy image recovery, Inverse Problems in Science and Engineering 28 (7) (2020) 1031–1056.

[11] D. Kitkuan, K. Muangchoo, A. Padcharoen, N. Pakkaranang, P. Kumam, A viscosity forward-backward splitting approximation method in Banach spaces and its application to convex optimization and image restoration problems, Computational and Mathematical Methods 2(4) (2020) e1098.

[12] A. Padcharoen, D. Kitkuan, Iterative methods for optimization problems and image restoration, Carpathian Journal of Mathematics 37 (3) (2021) 497–512.

[13] J. Abubakar, P. Kumam, A.H. Ibrahim, A. Padcharoen, Relaxed Inertial Tsengs Type Method for Solving the Inclusion Problem with Application to Image Restoration, Mathematics 8 (5) (2020) 818.

[14] A. Padcharoen, D. Kitkuan, W. Kumam, P. Kumam, Tseng methods with inertial for solving inclusion problems and application to image deblurring and image recovery problems, Computational and Mathematical Methods 3 (3) (2021) e1088.

[15] S.J. Kim, K. Koh, M. Lustig, S. Boyd, D. Gorinevsky, An interior-point method for large-scale ℓ_1-regularized least squares, IEEE Journal of Selected Topics in Signal Processing 1 (2007) 606–617.

[16] J. Li, F. Ding, G. Yang, Maximum likelihood least squares identification method for input nonlinear finite impulse response moving average systems, Mathematical and Computer Modelling 55 (2012) 442–450.

[17] A. Cornelio, Regularized nonlinear least squares methods for hit position reconstruction in small gamma cameras, Applied Mathematics and Computation 217 (2011) 5589–5595.

[18] T. Barz, S. Körkel, G. Wozny, et al, Nonlinear ill-posed problem analysis in model-based parameter estimation and experimental design, Computers and Chemical Engineering 77 (2015) 24–42.

[19] L.M. Tang, A regularization homotopy iterative method for il–posed nonlinear least squares problem and its application, In Applied Mechanics and Materials, Trans Tech Publ. (2011) 3268–3273.

[20] A.M. Awwal, P. Kumam, L. Wang, S. Huang, W. Kumam, Inertial-Based Derivative-Free Method for System of Monotone Nonlinear Equations and Application, IEEE Access 8 (2020) 226921–226930.

[21] A.M. Awwal, I.M. Sulaiman, M. Malik, M. Mamat, P. Kumam, K. Sitthithakerngkiet, A Spectral RMIL+ Conjugate Gradient Method for Unconstrained Optimization With Applications in Portfolio Selection and Motion Control, IEEE Access 9 (2021) 75398–75414.

[22] S. Aji, P. Kumam, A.M. Awwal, M.M. Yahaya, W. Kumam, Two hybrid spectral methods with inertial effect for solving system of nonlinear monotone equations with application in robotics, IEEE Access 9 (2021) 30918–30928.

[23] M. Kobayashi, Y. Narushima, H. Yabe, Nonlinear conjugate gradient methods with structured secant condition for nonlinear least squares problems, Journal of Computational and Applied Mathematics 234 (2021) 375–397.

[24] R. Dehghani, N. Mahdavi-Amiri, Scaled nonlinear conjugate gradient methods for nonlinear least squares problems, Numerical Algorithms 82 (2018) 1–20.

[25] H. Mohammad, M.Y. Waziri, Structured two-point stepsize gradient methods for nonlinear least squares, Journal of Optimization Theory and Applications 181 (2019) 298–317.

[26] A.M. Awwal, P. Kumam, L. Wang, M.M. Yahaya, H. Mohammad, On the Barzilai–Borwein gradient methods with structured secant equation for nonlinear least squares problems, Optimization Methods and Software (2020) 1–20.

[27] A.M. Awwal, P. Kumam, H. Mohammad, Iterative algorithm with structured diagonal Hessian approximation for solving nonlinear least squares problems, Journal of Nonlinear and Convex Analysis 22 (2021) 1173–1188.

[28] H. Mohammad, S.A. Santos, A structured diagonal hessian approximation method with evaluation complexity analysis for nonlinear least squares, Computational and

Applied Mathematics 37 (2018) 6619–6653.

[29] J.E. Dennis, H.J. Martinez, R.A. Tapia, Convergence theory for the structured BFGS secant method with an application to nonlinear least squares, Journal of Optimization Theory and Applications 61 (1989) 161–178.

[30] N. Andrei, A diagonal quasi-Newton updating method for unconstrained optimization, Numerical Algorithms 81 (2019) 575-590.

[31] 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, Journal of Computational and Applied Mathematics 395 (2021) 113582.

[32] H. Zhang, W.W. Hager, A nonmonotone line search technique and its application to unconstrained optimization, SIAM journal on Optimization 14 (2004) 1043–1056.

[33] E.D. Dolan, J.J. Moré, Benchmarking optimization software with performance profiles, Mathematical Programming 91 (2002) 201–213.

[34] W. La Cruz, J.M. Martínez, M. Raydan, Spectral residual method without gradient information for solving large-scale nonlinear systems: theory and experiments (2004) http://kuainasi.ciens.ucv.ve/mraydan/download_papers/TechRep.pdf.

[35] J.J. Moré, B.S. Garbow, K.E. Hillstrom, Testing unconstrained optimization software, Technical report, Argonne National Lab., IL (USA), 1978.

[36] L. Lukšan, J. Vlcek, Test problems for unconstrained optimization, Academy of Sciences of the Czech Republic, Institute of Computer Science, Technical Report 897 (2003).

[37] M. Raydan, The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem, SIAM Journal on Optimization 7 (1997) 26–33.

[38] M.M. Yahaya, P. Kumam, A.M. Awwal, S. Aji, Alternative structured spectral gradient method for solving nonlinear least square problems, Heliyon 7 (2021) e07499.

[39] J. Momin, Y. Xin-She, A literature survey of benchmark functions for global optimization problems, Journal of Mathematical Modelling and Numerical Optimization 4 (2013) 150–194.

[40] Y. Zhang, L. He, C. Hu, J. Guo, J. Li, Y. Shi, General four-step discrete-time zeroing and derivative dynamics applied to time-varying nonlinear optimization, Journal of Computational and Applied Mathematics 347 (2019) 314–329.

[41] A.M. Awwal, P. Kumam, H. Mohammad, W. Watthayu, A.B. Abubakar, A Perry-type derivative-free algorithm for solving nonlinear system of equations and minimizing ℓ_1 regularized problem, Optimization 70 (2020) 1231–1259.

[42] A.M. Awwal, L. Wang, P. Kumam, H. Mohammad, W. Watthayu, A projection Hestenes–Stiefel method with spectral parameter for nonlinear monotone equations and signal processing, Mathematical and Computational Applications 25 (2020) 27.

[43] A.M. Awwal, L. Wang, P. Kumam, H. Mohammad, A two-step spectral gradient projection method for system of nonlinear monotone equations and image deblurring problems, Symmetry 12 (2020) 874.

[44] S. Aji, P. Kumam, A.M. Awwal, M.M. Yahaya, K. Sitthithakerngkiet, An efficient DY-type spectral conjugate gradient method for system of nonlinear monotone equations with application in signal recovery, AIMS Mathematics 6 (2021) 8078–8106.

[45] A.M. Awwal, P. Kumam, K. Sitthithakerngkiet, A.M. Bakoji, A.S. Halilu, I.M. Sulaiman, Derivative-free method based on DFP updating formula for solving convex constrained nonlinear monotone equations and application, AIMS Mathematics 6 (2021) 8792–8814.

[46] K.U. Danmalam, H. Mohammad, A.B. Abubakar, A.M. Awwal, Hybrid algorithm for system of nonlinear monotone equations based on the convex combination of Fletcher-Reeves and a new conjugate residual parameters, Thai Journal of Mathematics 18 (2021) 2093–2106.