Research article

Derivative-free HS-DY-type method for solving nonlinear equations and image restoration

Auwal Bala Abubakar\textsuperscript{a,b,c}, Poom Kumam\textsuperscript{a,d,r}, Abdulkarim Hassan Ibrahim\textsuperscript{a}, Jewaidu Rilwan\textsuperscript{a,b}

\textsuperscript{a} Center of Excellence in Theoretical and Computational Science (TaCS-CoE), Science Laboratory Building, Department of Mathematics, Faculty of Science, King Mongkut’s University of Technology Thonburi (KMUTT), 126 Pracha-Uthit Road, Bang Mod, Thung Khrong, Bangkok 10140, Thailand
\textsuperscript{b} Department of Mathematical Sciences, Faculty of Physical Sciences, Bayero University, Kano, Kano, Nigeria
\textsuperscript{c} Numerical Optimization Research Group, Bayero University, Kano, Nigeria
\textsuperscript{d} KMUTT Fixed Point Theory and Applications Research Group, Theoretical and Computational Science Center (TaCS), Science Laboratory Building, Faculty of Science, King Mongkut’s University of Technology Thonburi (KMUTT), 126 Pracha-Uthit Road, Bang Mod, Thung Khrong, Bangkok 10140, Thailand

A R T I C L E   I N F O

Keywords:
Mathematics
Nonlinear equations
Conjugate gradient
Projection method
Image restoration

A B S T R A C T

A derivative-free conjugate gradient algorithm for solving nonlinear equations and image restoration is proposed. The conjugate gradient (CG) parameter of the proposed algorithm is a convex combination of Hestenes-Stiefel (HS) and Dai-Yuan (DY) type CG parameters. The search direction is descent and bounded. Under suitable assumptions, the convergence of the proposed hybrid algorithm is obtained. Using some benchmark test problems, the proposed algorithm is shown to be efficient compared with existing algorithms. In addition, the proposed algorithm is effectively applied to solve image restoration problems.

1. Introduction

Conjugate gradient (CG) method is a well known method that efficiently solves nonlinear equations of the form

\[
H(x) = 0, \quad x \in A,
\]

where \( H : \mathbb{R}^n \rightarrow \mathbb{R}^n \) and \( A \) is a closed and convex subset of the Euclidean space \( \mathbb{R}^n \). The CG algorithm generates an iterative sequence \( \{x_k\} \) via the following formula:

\[
x_{k+1} = x_k + a_k d_k,
\]

where \( a_k \) is the step size obtained from a suitable line search process and \( d_k \) is the search direction defined as

\[
d_k = -Hx_k, \quad k = 0,
\]

\[
d_k = -Hx_k + \beta_k d_{k-1}, \quad k > 0.
\]

The parameter \( \beta_k \) is called the CG parameter. Throughout this paper, \( H_k \) denotes the function evaluation of \( H \) at \( x_k \), and \( \langle \cdot , \cdot \rangle \) denotes the inner product.

Several CG algorithms for solving (1) have been proposed in literature. For example, Feng et al. [16] proposed a CG based algorithm where the search direction is defined as

\[
d_k := \begin{cases} -H_k, & \text{if } k = 0, \\ \left(1 + \beta_k \frac{\langle H_k, d_{k-1} \rangle}{\|H_k\|^2}\right) H_k + \beta_k d_{k-1}, & \text{if } k \geq 1, \end{cases}
\]

where \( \beta_k := \frac{\|H_k\|^2}{\|d_{k-1}\|^2} \).

In [27], Liu and Feng proposed a search direction defined as

\[
d_k := \begin{cases} -H_k, & \text{if } k = 0, \\ -\beta_k H_k + \beta_k d_{k-1}, & \text{if } k \geq 1, \end{cases}
\]

where

\[
\beta_k := \frac{\|H_k\|^2}{\langle d_k, d_{k-1} \rangle}, \quad \beta_k := c - \frac{\|H_k, d_{k-1}\|^2}{\langle d_k, d_{k-1} \rangle},
\]

\[
y_{k-1} := H_k - H_{k-1} + r_{k-1}, \quad x_{k-1} := x_k - x_{k-1},
\]

\[
u_{k-1} := y_{k-1} + t_{k-1} d_{k-1}, \quad t_{k-1} := 1 + \max \left\{0, \frac{-\langle d_{k-1}, y_{k-1} \rangle}{\|d_{k-1}\|^2} \right\}.
\]

The algorithm in [27] was shown to be efficient for solving convex constrained monotone equations. Awawal et al. [10] also proposed a...
modified HS conjugate gradient algorithm for solving problem (1) as well as signal recovery problem. The search direction is defined as

\[
d_k := \begin{cases} -H_k, & \text{if } k = 0, \\ -\lambda_k H_k + \beta_k^{\text{MHS}} s_{k-1}, & \text{if } k \geq 1, \end{cases}
\]

where

\[
\beta_k^{\text{MHS}} := \max\{\beta_k^{\text{MHS}}_+, 0\},
\]

\[
\beta_k^{\text{MHS}} := \frac{\|s_{k-1}\|^2 (H_k, y_{k-1})}{(H_k, y_{k-1}) (d_{k-1}, y_{k-1})} \theta_k - \gamma 
\times \frac{\|s_{k-1}\| \|s_{k-1}\| \|H_k\|^2}{(d_{k-1}, y_{k-1})} \left( H_k, d_{k-1} \right), \quad \gamma > \frac{1}{4},
\]

\[
\lambda_k := \frac{\|s_{k-1}\|^2}{(d_{k-1}, y_{k-1})}, \quad \theta_k := 1 - \frac{(H_k^T d_{k-1})^2}{\|H_k\|^2 \|d_{k-1}\|^2},
\]

\[
y_{k-1} := H_k - H_k + \eta s_{k-1}, \quad s_{k-1} := x_k - x_{k-1}, \quad \eta > 0.
\]

In the same line of research, Awuwa et al. [11] further proposed a modified Polak-Ribière-Polyak (PRP) conjugate gradient algorithm with search direction given by

\[
d_k := \begin{cases} -H_k, & \text{if } k = 0, \\ -\theta_k H_k + \beta_k^{\text{PRP}} s_{k-1}, & \text{if } k \geq 1, \end{cases}
\]

where

\[
\theta_k := \lambda_k + \frac{\beta_k^{\text{PRP}} (H_k, s_{k-1})}{\|H_k\|^2},
\]

\[
\lambda_k := \frac{\|s_{k-1}\|^2}{(s_{k-1}, \psi_{k-1})},
\]

\[
\psi_{k-1} := H_k - H_k + \eta s_{k-1}, \quad s_{k-1} := x_k - x_{k-1}, \quad \eta > 0.
\]

Recently, Yuan et al. [34] proposed a conjugate gradient algorithm which is a convex combination of the steepest descent algorithm and a modified Liu-Storey (LS) conjugate gradient algorithm. The search direction defined by Yuan et al. is

\[
d_k := \begin{cases} -H_k, & \text{if } k = 0, \\ -N_k H_k + (1 - N_k) \frac{(H_k, y_{k-1}) d_{k-1} - (H_k, d_{k-1}) y_{k-1}}{\max\{2\|d_{k-1}\| \|y_{k-1}\| - (H_k, d_{k-1})\}}, & \text{if } k \geq 1, \end{cases}
\]

where

\[
N_k := \frac{\|y_{k-1}\|^2}{(s_{k-1}, y_{k-1})},
\]

\[
u_{k-1} := s_{k-1} + \max\left\{0, -\frac{(s_{k-1}, y_{k-1})}{\|y_{k-1}\|^2} \right\} y_{k-1}, \quad x \in (0, 1),
\]

\[
s_{k-1} := x_k - x_{k-1}, \quad u_{k-1} := H_k - H_k.
\]

For more on the conjugate gradient algorithms, the interested reader is referred to [1, 2, 3, 4, 5, 6, 7, 8, 9, 18, 19, 20, 21, 22, 26, 29, 35].

To the best of our knowledge, very few hybrid algorithms for solving (1) are available in the literature. To this end, we explore the strong convergence property of the DY algorithm together with the good practical behavior of the HS algorithm by proposing a conjugate gradient method with a conjugate gradient parameter computed as a convex combination of a modified HS and DY parameters. Furthermore, the hybrid method is applied to solve image restoration problem arising in compressive sensing.

Section 2 highlights the reason behind modifying the HS and DY parameters, suggesting the modification and its advantages, and then describing the proposed algorithm. Section 3 gives some nice properties of the search direction and the convergence analysis of the proposed algorithm. Numerical experiments on some benchmark test problems for solving (1) and image restoration problem are given in Section 4. Finally, Section 5 concludes the paper.
Therefore,
\[
0 < \frac{\|x_{k-1}\|^2}{\langle x_{k-1}, x_{k-1} \rangle} \leq 1.
\]

**Remark 2.2.** From the definition of \(\rho^k_{HSDY}\) and \(\theta_k\), we have
\[
|\rho^k_{HSDY}| \leq |\rho^k_{MHS}| + |\rho^k_{ADY}|. \tag{16}
\]

To describe the hybrid conjugate gradient algorithm, we first recall the projection map.

**Definition 2.3.** Let \(A \subset \mathbb{R}^n\) be a nonempty, closed and convex set. Then for any \(x \in \mathbb{R}^n\), its projection onto \(A\), denoted by \(P_A(x)\), is defined by
\[
P_A(x) := \text{arg min}\{\|x - y\| : y \in A\}.
\]

A known property of \(P_A\) is that it is non-expansive, that is,
\[
\|P_A(x) - P_A(y)\| \leq \|x - y\|, \quad \forall x, y \in \mathbb{R}^n.
\tag{17}
\]

In what follows, we present the steps of the derivative-free algorithm. Throughout, we refer to the proposed algorithm as Algorithm 1.

**Algorithm 1:**

Step 0. Given an arbitrary initial point \(x_0 \in A\), parameters \(\sigma > 0, \rho \in (0, 1)\), \(\gamma \in (0, 2), \tau_0 > 0\) and set \(k := 0\).

Step 1. If \(\|H_k\| \leq \tau_0\), stop, otherwise go to Step 2.

Step 2. Compute \(d_k\) by (1)–(13).

Step 3. Compute the step size \(\alpha_k = \rho^i\) where \(i\) is the smallest non-negative integer such that
\[
\langle H(x_k + \alpha_k d_k), d_k \rangle \geq \sigma \alpha_k \|d_k\|^2. \tag{18}
\]

Step 4. Compute
\[
z_k = x_k + \alpha_k d_k.
\]

If \(z_k \in A\) and \(\|H(z_k)\| \leq \tau_0\), stop. Else compute
\[
x_{k+1} = P_A[x_k - \gamma \nabla H(x_k)]
\tag{20}
\]

where
\[
\gamma = \frac{\langle H(z_k), x_k - z_k \rangle}{\|H(z_k)\|^2}.
\]

Step 5. Let \(k := k + 1\) and go to Step 1.

**Remark 2.1** implies that (14) is a convex combination of (12) and (13).

**Remark 2.4.** The parameter \(\gamma\) in equation (20) is chosen from the interval \((0, 2)\) so as to have the sequence \(\{\|x_k - \tilde{x}_k\|\}\) non-increasing (see Lemma 3.5). In addition, the parameter \(\gamma\) has a significant impact on the numerical performance of Algorithm 1.

### 3. Convergence analysis

To establish the convergence of Algorithm 1, we begin with the following assumptions:

**A1.** The function \(H\) is monotone, that is,
\[
\langle H(x) - H(y), (x - y) \rangle \geq 0, \quad \forall x, y \in \mathbb{R}^n.
\]

**A2.** The function \(H\) is Lipschitz continuous, that is there exists a positive constant \(L\) such that
\[
\|H(x) - H(y)\| \leq L\|x - y\|, \quad \forall x, y \in \mathbb{R}^n.
\]

**A3.** The solution set of problem (1) denoted by \(A'\) is nonempty. \(A_k H_k \neq 0\) unless the solution of (1) is obtained.

**Lemma 3.1.** Let \(d_k\) be defined by (14)–(15), then \(d_k\) satisfies the sufficient descent condition. That is
\[
\langle H_k, d_k \rangle = -\|H_k\|^2. \tag{21}
\]

**Proof.** For \(k = 0\), we have \(\langle H_0, d_0 \rangle = -\|H_0\|^2\). For \(k \geq 1\), by (14)–(15), we get
\[
\langle H_k, d_k \rangle = -\left(1 + \rho^k_{HSDY}(H_k, d_{k-1})\right)\langle H_k, H_k \rangle + \rho^k_{HSDY}(H_k, d_{k-1})
\]
\[
= -\|H_k\|^2 - \|H_k\|^2 \|H_k\|^2 \|H_k\|^2 \|H_k\|^2
\]
\[
= -\|H_k\|^2 - \|H_k\|^2 \|H_k\|^2 \|H_k\|^2
\]
\[
= -\|H_k\|^2. \tag{22}
\]

**Remark 3.2.** From (21), applying Cauchy-Schwartz inequality we have
\[
\|d_k\| \geq \|H_k\|. \tag{23}
\]

The following Lemma shows that Algorithm 1 is well-defined.

**Lemma 3.3.** If assumption \(A_3\) holds, then there exists a step size \(\alpha_k = \rho^i\) satisfying the line search (18) for some \(i \in \mathbb{N} \cup \{0\}\) and \(\forall k \geq 0\).

**Proof.** Suppose there exists \(k_0 \geq 0\) such that (18) does not hold for any non-negative integer \(i\), that is,
\[
-\langle H(x_{k_0} + \rho d_{k_0}), d_{k_0} \rangle < \sigma \rho \|d_{k_0}\|^2.
\]

By assumption \(A_3\) and allowing \(i \to \infty\), we get
\[
-\langle H(x_{k_0}), d_{k_0} \rangle \leq 0. \tag{24}
\]

Also from (22), we have
\[
-\langle H(x_{k_0}), d_{k_0} \rangle \geq \|H(x_{k_0})\|^2 > 0,
\]
which contradicts (24). The proof is complete. \(\square\)

**Lemma 3.4.** Suppose assumption \(A_3\) holds. If \(z_k\) and \(x_k\) are defined by (19) and (20) in Algorithm 1, then
\[
\alpha_k \geq \max \left\{1, \frac{\rho \|H_k\|^2}{(L + \sigma)\|d_k\|^2} \right\}. \tag{25}
\]

**Proof.** From the line search (18), if \(a_k \neq 1\), then \(a'_k = a_k \rho^{-1}\) does not satisfy (18), that is,
\[
-\langle H(x_k + a'_k d_k), d_k \rangle < \sigma a'_k \|d_k\|^2.
\]

Using (21) and assumption \(A_3\), we have
\[
\|H_k\|^2 \leq -\langle H_k, d_k \rangle
\]
\[
= \langle H(x_k + a'_k d_k) - H_k, d_k \rangle - \langle H(x_k + a'_k d_k), d_k \rangle
\]
\[
\leq a'_k (L + \sigma)\|d_k\|^2.
\]

Solving the above inequality for \(a'_k\), the desired result is obtained. \(\square\)

**Lemma 3.5.** Let assumptions \(A_1\)–\(A_3\) be fulfilled. If \(z_k\) and \(x_k\) are sequences defined by (19) and (20) in Algorithm 1, then \(z_k\) and \(x_k\) are bounded. Furthermore,
\[\lim_{k \to \infty} \|x_k - z_k\| = 0,\]
\[\text{and}\]
\[\lim_{k \to \infty} \|x_{k+1} - x_k\| = 0.\]

**Proof.** We begin by showing that the sequence \(\{x_k\}\) and \(\{z_k\}\) are bounded. Suppose \(\bar{x} \in A'\), then by monotonicity of \(H\), we get
\[
\langle H(z_k), x_k - \bar{x} \rangle \geq \langle H(z_k), x_k - z_k \rangle.
\]
From the definition of \(z_k\) and (18), we have
\[
\langle H(z_k), x_k - z_k \rangle \geq \sigma \gamma_k^2 \|d_k\|^2 \geq 0.
\]
Consequently, by (17), (28), (29), the definition of \(\zeta_k\) and \(\gamma \in (0, 2)\), we have
\[
\|x_{k+1} - \bar{x}\|^2
= \|P_A(x_k - \gamma \zeta_k H(z_k) - P_A(\tilde{x}))\|^2
\leq \|x_k - \gamma \zeta_k H(z_k) - \bar{x}\|^2
= \|x_k - \bar{x}\|^2 - 2\gamma \zeta_k \langle H(z_k), x_k - \bar{x} \rangle + \|\gamma \zeta_k H(z_k)\|^2
= \|x_k - \bar{x}\|^2 - 2\gamma \zeta_k \langle H(z_k), x_k - z_k \rangle \frac{\|H(z_k)\|^2}{\|H(z_k)\|^2}
+ \gamma^2 \left(\frac{\|H(z_k)\|^2}{\|H(z_k)\|^2}\right)^2
\leq \|x_k - \bar{x}\|^2 - 2\gamma \zeta_k \frac{\|H(z_k)\|^2}{\|H(z_k)\|^2} \langle H(z_k), x_k - z_k \rangle
+ \gamma^2 \left(\frac{\|H(z_k)\|^2}{\|H(z_k)\|^2}\right)^2
\leq \|x_k - \bar{x}\|^2 - \gamma(2 - \gamma) \frac{\|H(z_k)\|^2}{\|H(z_k)\|^2}
\leq \|x_k - \bar{x}\|^2 - \gamma(2 - \gamma) \frac{\|x_k - z_k\|^4}{\|H(z_k)\|^2}.
\]
Thus, the sequence \(\{\|x_k - \bar{x}\|\}\) is non-increasing and convergent, and hence \(\{x_k\}\) is bounded. That is,
\[
\|x_k\| \leq b, \quad b > 0.
\]
Moreover, from relation (30), we have
\[
\|x_{k+1} - \bar{x}\|^2 \leq \|x_k - \bar{x}\|^2,
\]
and we can deduce recursively that
\[
\|x_k - \bar{x}\|^2 \leq \|x_0 - \bar{x}\|^2, \quad \forall k \geq 0.
\]
Therefore from assumption \(A_2\), we have that
\[
\|H_k\| = \|H_k - H(\bar{x})\| \leq L_1 \|x_k - \bar{x}\| \leq L_1 \|x_0 - \bar{x}\|.
\]
Letting \(L_1 \|x_0 - \bar{x}\| = B\), then the sequence \(\{H_k\}\) is bounded. That is,
\[
\|H_k\| \leq B, \quad \forall k \geq 0.
\]
Now by monotonicity of \(H\),
\[
\langle H_k - H(z_k), x_k - z_k \rangle \geq 0,
\]
which implies that
\[
\langle H_k, x_k - z_k \rangle - \langle H(z_k), x_k - z_k \rangle \geq 0.
\]
Hence
\[
\langle H(z_k), x_k - z_k \rangle \leq \langle H_k, x_k - z_k \rangle.
\]
By the definition of \(z_k\), (29), (34) and the Cauchy-Schwarz inequality,
\[
s\|x_k - z_k\| = \|x_k - z_k\|^2 = \|x_k - d_k\|^2
\leq \frac{\langle H(z_k), x_k - z_k \rangle}{\|x_k - z_k\|^2} \leq \frac{\langle H(z_k), x_k - z_k \rangle}{\|x_k - z_k\|} \leq \|H_k\|.
\]
By (35) and the reverse triangle inequality,
\[
s\|z_k\| - \|x_k\| \leq s\|z_k - x_k\| \leq \|H_k\|.
\]
The above relation together with (31) and (33) yield
\[
\|z_k\| \leq \frac{1}{\sigma} \|H_k\| + \|x_k\|
\leq \frac{1}{\sigma} B + b.
\]
Therefore the sequence \(\{z_k\}\) is bounded.

Now, for any \(\tilde{x} \in A'\), the sequence \(\{z_k - \tilde{x}\}\) is also bounded, that is, there exists a positive constant \(\nu > 0\) such that
\[
\|z_k - \tilde{x}\| \leq \nu, \quad \forall k \geq 0.
\]
The above inequality together with assumption \(A_2\) yield
\[
\|H(z_k)\| = \|H(z_k) - H(\tilde{x})\| \leq L_1 \|z_k - \tilde{x}\| \leq L_1 \nu.
\]
Therefore, using relation (30), we have
\[
\gamma(2 - \gamma) \frac{\|H(z_k)\|^2}{(L_1 \nu)^2} \leq \|x_k - \bar{x}\|^2 - \|x_{k+1} - \bar{x}\|^2,
\]
which implies
\[
\gamma(2 - \gamma) \frac{\|H(z_k)\|^2}{(L_1 \nu)^2} \sum_{k=0}^{\infty} \|x_k - z_k\|^4
\leq \sum_{k=0}^{\infty} \|x_k - \bar{x}\|^2 - \|x_{k+1} - \bar{x}\|^2 = 0.
\]
Relation (36) implies that
\[
\lim_{k \to \infty} \|x_k - z_k\| = 0.
\]
In addition, using (17), the definition of \(\zeta_k\) and the Cauchy-Schwarz inequality,
\[
\|x_{k+1} - x_k\| = \|P_A(x_k - \gamma \zeta_k H(z_k) - P_A(\tilde{x}))\|
\leq \|x_k - \gamma \zeta_k H(z_k) - x_k\|
= \|\gamma \zeta_k H(z_k)\|
\leq \gamma \|x_k - z_k\|, \quad \forall k \geq 0.
\]
It follows that
\[
\lim_{k \to \infty} \|x_{k+1} - x_k\| = 0. \square
\]

**Remark 3.6.** From (26) and definition of \(z_k\),
\[
\lim_{k \to \infty} \|d_k\| = 0.
\]

**Theorem 3.7.** Let the sequence \(\{x_k\}\) be generated by (20) in Algorithm 1, then
\[
\lim_{k \to \infty} \|H_k\| = 0.
\]
The algorithms are terminated by reaching a maximum of 1000 iterations or achieving a solution with 
\[ \| H_k \| \leq 10^{-6}. \]

Note that the parameters for the algorithms used for comparison are set as reported in the numerical section of their respective papers. We give a list of the benchmark test problems used in our experiment below where the function \( H \) is taken as \( H(x) = (h_1(x), h_2(x), \ldots, h_n(x))^T \) and \( x = (x_1, x_2, \ldots, x_n)^T \).

**Problem 1** [23] Exponential Function.
\[ h_i(x) = e^{x_i} - 1, \quad i = 1, 2, \ldots, n, \]
and \( A = R^n_+ \).

**Problem 2** [23] Modified Logarithmic Function.
\[ h_i(x) = \ln(x_i + 1) - \frac{x_i}{n}, \quad i = 1, 2, \ldots, n, \]
and \( A = \{ x \in R^n : \sum_{i=1}^n x_i \leq n, x_i > -1, \ i = 1, 2, \ldots, n \} \).

**Problem 3** [36] Nonsmooth Function.
\[ h_i(x) = 2 \sin(x_i) - \sin(x_i^2), \quad i = 1, 2, \ldots, n, \]
and \( A = R^n_+ \).

**Problem 4** [24]
\[ h_i(x) = \max \{ \min \{ |x_i|, x_i^2 \} \}, \quad i = 1, 2, \ldots, n, \]
and \( A = R^n_+ \).

**Problem 5** [23] Strictly Convex Function I.
\[ h_i(x) = e^{x_i} - 1, \quad i = 1, 2, \ldots, n, \]
and \( A = R^n_+ \).

**Problem 6** [30] Strictly convex function II.
\[ h_i(x) = \frac{i}{n} e^{x_i} - 1, \quad i = 1, 2, \ldots, n, \]
and \( A = R^n_+ \).

**Problem 7** [12] Tridiagonal Exponential Function.
\[ h_i(x) = e^{\cos(h(x_i + x_{i+1}))}, \quad i = 2, \ldots, n-1, \]
\[ h_n(x) = e^{\cos(h(x_n + x_{n+1}))}, \quad h = \frac{1}{n+1} \]
and \( A = R^n_+ \).

**Problem 8** [33] Nonsmooth Function.
\[ h_i(x) = x_i - \sin(x_i - 1), \quad i = 1, 2, \ldots, n, \]
and \( A = \{ x \in R^n : \sum_{i=1}^n x_i \leq n, x_i \geq -1, \ i = 1, 2, \ldots, n \} \).

**Problem 9** [36]
\[ h_i(x) = 2x_i + \sin(x_i - 1), \quad i = 2, \ldots, n-1, \]
\[ h_n(x) = 2x_n + \sin(x_n - 1), \]
and \( A = \{ x \in R^n : \sum_{i=1}^n x_i \leq n, x_i \geq 0, \ i = 1, 2, \ldots, n \} \).
Problem 10 Pursuit-Evasion problem.

\[ h_i(x) = \sqrt{x_i} - 1, \text{ for } i = 1, 2, \ldots, n, \]
and \( A = R^+_n \).

The algorithms’ numerical results are reported in Table 2-11 of the Appendix section, where “ITER” denotes the number of iterations, “FVAL” denotes the number of function evaluations and “TIME” is the CPU running time in seconds. In order to visualize the behavior of HSDY, we employ the Dolan and Moré performance profile tool [15] for efficiency comparison. The performance profile tool seeks to find how well the solvers perform relative to the other solvers on a set of problems based on the total number of iterations, total number of function evaluations, and the CPU running time. We quickly recall this process.

Denote \( M \) as the set of the methods, and \( E \) as the set of the experiments (the four methods test one problem with the same number of variables and initial point as one experiment). The parameter \( i_{m,e} \) means NITER, NF, or TIME of the method \( m \in M \) in the \( e \)-th experiment. The performance ratio is computed as \( r_{m,e} = i_{m,e} / \min_{m \in M} i_{m,e} \). Then the performance profile is determined by

\[ \chi_m(r) := \frac{1}{n_e} \sum_{i \in E} \frac{\log_2(r_{m,e})}{r \leq m \leq n_e}, \forall r \in R^+ \]

where \( n_e \) denotes the number of methods in the set \( M \). Obviously, the function \( \chi_m : R \rightarrow [0, 1] \) is a distribution function for the performance ratio. And for any \( m \in M \), \( \chi_m \) is a non-decreasing, piecewise constant, continuous function from the right at each breakpoint. Moreover, \( \chi_m(r) \) is the probability for the method \( m \in M \) that \( \log_2(r_{m,e}) \) is within a factor \( r \in R^+ \) of the best possible ratio. Thus, when \( r \) takes certain value, for any \( m \in M \), the method with high value of \( \chi_m(r) \) is preferable or represents the best method. By this technique, we obtain the Figs. 1, 2 and 3. Based on the performance profile obtained, we can observe that with respect to number of iterations and function evaluations HSDY algorithm solves and win in over 50 percent of the problems as against CGD, PDY and ACGD with 18, 10 and 33 percent success, respectively. However, with respect to CPU time HSDY algorithm solves and win in over 32 percent of the problems as against CGD, PDY and ACGD with 11, 10 and 48 percent success, respectively. Therefore, we conclude that the HSDY method is more efficient than CGD, PDY and ACGD.

4.1. Image restoration problem

Image restoration problem is usually aimed at recovering sparse original image \( \hat{s} \) from a degraded observation \( b \) using the equation

\[ b = Ax, \quad (44) \]

where \( A \in R^{m \times n} (m < n) \) is a linear map. However, since (44) is ill-conditioned, then the basic pursuit denoising framework (\( \ell_1 \)-norm problem) is appropriate

\[ \min f(x) \equiv \frac{1}{2} \|y - Ax\|_2^2 + \tau \|x\|_1, \quad \tau > 0, \quad (45) \]

where \( x \in R^n, \ y \in R^m, \ A \in R^{m \times n} \). Throughout this section, we use \( \|x\|_1 = \sum_{i=1}^{m} |x_i| \) and \( \|x\|_2 \) to denote the \( \ell_1 \)-norm of vector \( x \in R^n \) and the Euclidean norm, respectively.

In order to solve (45), we quickly give an overview of its reformulation into a convex quadratic problem by Figueiredo [17]. Any vector \( x \in R^n \) can be written as

\[ x = u - v, \quad u \geq 0, v \geq 0. \]
where $u \in \mathbb{R}^n$, $v \in \mathbb{R}^n$ and $u_i = (x_i, y_i), v_i = (-x_i, y_i)$ for all $i = 1, 2, \ldots, n$ with $\langle \cdot, \cdot \rangle = \max(0, \cdot)$. Subsequently, the $\ell_1$-norm of a vector can be represented as $\|x\|_1 = e_n^T u + e_n^T v$, where $e_n$ is an $n$-dimensional vector with all elements one. Hence, the $\ell_1$-norm problem (45) was transformed into

$$\min_{u,v} \frac{1}{2} \|b - A(u - v)\|^2 + re_n^T u + re_n^T v, \quad u \geq 0, \quad v \geq 0. \quad (46)$$

From [17], the above equation can be easily rewritten as the quadratic program problem with box constraints

$$\min_{z} \frac{1}{2} z^T D z + c^T z, \quad \text{s.t.} \quad z \geq 0, \quad (47)$$

where

$$z = \begin{bmatrix} u \\ v \end{bmatrix}, \quad y = A^T b,$$

$$c = re_n + \begin{bmatrix} -y \\ y \end{bmatrix}, \quad D = \begin{bmatrix} A^T A & -A^T A \\ -A^T A & A^T A \end{bmatrix}.$$ 

Simple calculation shows that $D$ is a semi-definite positive matrix. Hence (47) is a convex quadratic program problem, and it is equivalent to

$$H(z) = \min \{ z, Dz + c \} = 0. \quad (48)$$

The function $D$ is vector-valued and the min interpreted as component-wise minimum. With the reformulation, from [28, Lemma 3] and [31, Lemma 2.2], since $D$ is Lipschitz continuous and monotone, then the HSDY algorithm can be effectively used to solve (48).

Next, we apply the proposed hybrid conjugate gradient algorithm in image restoration. In order to evaluate the efficiency of the proposed algorithm in image restoration, we compare its numerical performance with the CGD algorithm [32] designed for solving monotone equations and image restoration. We consider the following classical test images with color to illustrate the efficiency of the proposed algorithm (Figs. 4 and 5).

The above test images in Fig. 4 are obtained from http://hlevkin.com/06testimages.htm. All simulations are performed in Matlab (R2019b) on a HP with 2.4GHz processor and 8GB RAM. The parameters for the proposed algorithm are set as $\rho = 0.4$, $\sigma = 10^{-4}$. The quality of restoration by the algorithms are determined using Signal-to-ratio (SNR), Peak signal to noise ratio (PSNR) and Structural similarity index (SSIM). For fairness in comparing the algorithms, iteration process of all algorithms begin from $x_0 = A^T b$ and terminates when

$$\frac{|f_k - f_{k-1}|}{|f_{k-1}|} < 10^{-6},$$

where $f(x) = \frac{1}{2} \|Ax - b\|^2 + \tau \|x\|_1$ is the objective function and $f_k$ denotes the function value at $x_k$. The original, blurred and restored images by each of the algorithms are given in Fig. 5.

In the following table, we report the numerical result for the test images used in this experiment.

From the Table 1, it can be observed that both algorithms were able to restore the blurred images. However, HSDY algorithm restored the images with better performance than that of CGD algorithm. This can be seen from the SNR, PSNR and SSIM values. It can be noticed that the

| Image       | CGD            | HSDY           |
|-------------|----------------|----------------|
| SNR         | PSNR           | SSIM           |
| SNR         | PSNR           | SSIM           |
| SNR         | PSNR           | SSIM           |
| SNR         | PSNR           | SSIM           |
| SNR         | PSNR           | SSIM           |
| SNR         | PSNR           | SSIM           |
| SNR         | PSNR           | SSIM           |
| SNR         | PSNR           | SSIM           |
| SNR         | PSNR           | SSIM           |
| SNR         | PSNR           | SSIM           |

SNR, PSNR and SSIM values of the images restored by our algorithm are about 0.01 to 0.05 larger than those restored by CGD. The MATLAB implementation of the SSIM index can be obtained at http://www.cns.nyu.edu/~lcv/ssim/.

5. Conclusions

In this article, we proposed a conjugate gradient algorithm where the direction is a convex combination of two well known CG parameters, HS and DY. Independent of any line search, the proposed direction is sufficiently descent and bounded. Global convergence of the proposed algorithm was established under appropriate assumptions. Compared
with CGD, PDY and ACGD algorithms, the HSDY algorithm performs better in terms of number of iteration and number of function evaluations. However, in terms of CPU time, ACGD algorithm performs better than HSDY, CGD and PDY. This may be as a result of the less computational cost associated with the ACGD algorithm. Finally, after reformulation, the HSDY algorithm was applied to restore blurred image.

**Author contribution statement**

A. B. Abubakar: Conceived and designed the experiments; Wrote the paper.

P. Kumam: Contributed reagents, materials, analysis tools or data.

A. H. Ibrahim: Performed the experiments; Wrote the paper.

J. Rilwan: Analyzed and interpreted the data; Wrote the paper.

**Funding statement**

The authors acknowledge the financial support provided by King Mongkut’s University of Technology Thonburi through the “KMUTT 55th Anniversary Commemorative Fund”. The first author was supported by the Petchra Pr Jom Klao Doctoral Scholarship Academic for Ph.D. Program at KMUTT. Moreover, this project was partially supported by the Thailand Research Fund (TRF) and the King Mongkut’s University of Technology Thonburi (KMUTT) under the TRF Research Scholar Award (Grant No. RSA6080047).

**Declaration of interests statement**

The authors declare no conflict of interest.

**Table 2. Computational results for Problem 1.**

| DM | 209 | HSDY | ITER | FVAL | TIME | NORM |
|----|-----|------|------|------|------|------|
| 1000 | 2 | 7 | 0.005294 | 0 | 42 | 125 | 0.035717 | 9.979E-07 | 16 | 64 | 0.033816 | 3.45E-07 |
| 2 | 7 | 0.06493 | 0 | 45 | 134 | 0.342415 | 4.53E-07 | 16 | 64 | 0.02728 | 7.018E-07 |
| 4 | 7 | 0.068178 | 0 | 48 | 143 | 0.198984 | 9.828E-07 | 17 | 68 | 0.015225 | 6.22E-07 |
| 2 | 7 | 0.004067 | 0 | 50 | 149 | 0.019721 | 7.918E-07 | 18 | 72 | 0.017206 | 4.54E-07 |
| 2 | 7 | 0.003323 | 0 | 51 | 152 | 0.002101 | 9.178E-07 | 18 | 72 | 0.043715 | 1.65E-06 |
| 2 | 7 | 0.003087 | 0 | 51 | 152 | 0.013188 | 5.68E-07 | 18 | 72 | 0.017788 | 3.60E-07 |
| 5 | 10 | 0.013236 | 1.35E-07 | 45 | 125 | 0.026073 | 9.99E-07 | 17 | 68 | 0.01432 | 7.46E-07 |
| 5000 | 2 | 7 | 0.012354 | 0 | 41 | 122 | 0.17967 | 5.34E-07 | 18 | 72 | 0.043568 | 7.61E-07 |
| 2 | 7 | 0.043668 | 0 | 43 | 128 | 0.06739 | 9.81E-07 | 17 | 68 | 0.055717 | 5.15E-07 |
| 2 | 7 | 0.002624 | 0 | 47 | 140 | 0.059299 | 8.05E-07 | 18 | 72 | 0.066327 | 4.63E-07 |
| 2 | 7 | 0.003675 | 0 | 48 | 143 | 0.071964 | 9.93E-07 | 19 | 76 | 0.05885 | 3.36E-07 |
| 2 | 7 | 0.013753 | 0 | 49 | 146 | 0.29583 | 8.36E-07 | 18 | 72 | 0.089462 | 8.12E-07 |
| 2 | 7 | 0.015648 | 0 | 49 | 146 | 0.0323 | 7.68E-07 | 18 | 72 | 0.048756 | 1.01E-06 |
| 2 | 7 | 0.017914 | 0 | 50 | 147 | 0.29583 | 8.36E-07 | 18 | 72 | 0.089462 | 8.12E-07 |
| 2 | 7 | 0.008178 | 0 | 48 | 143 | 0.018984 | 8.38E-07 | 19 | 76 | 0.011015 | 4.77E-07 |
| 3 | 2 | 0.020057 | 0 | 48 | 143 | 0.086187 | 8.73E-07 | 20 | 80 | 0.10581 | 4.52E-07 |
| 3 | 2 | 0.019788 | 0 | 48 | 143 | 0.13679 | 1.97E-07 | 19 | 76 | 0.11446 | 5.15E-07 |
| 3 | 2 | 0.011513 | 0 | 44 | 131 | 0.57169 | 9.104E-07 | 19 | 76 | 0.09621 | 4.86E-07 |
| 3 | 2 | 0.017373 | 0 | 46 | 137 | 0.05817 | 8.46E-07 | 20 | 80 | 0.074508 | 9.70E-07 |
| 3 | 2 | 0.090097 | 0 | 46 | 137 | 0.38662 | 9.34E-07 | 22 | 88 | 0.69514 | 8.63E-07 |
| 4 | 2 | 0.077672 | 0 | 46 | 137 | 0.68552 | 9.78E-07 | 23 | 92 | 0.65302 | 8.62E-07 |
| 4 | 2 | 0.03462 | 0 | 45 | 134 | 0.42841 | 8.71E-07 | 19 | 76 | 0.41133 | 5.63E-07 |
| 4 | 2 | 0.031916 | 0 | 39 | 116 | 0.35628 | 7.72E-07 | 18 | 72 | 0.07035 | 3.76E-07 |
| 4 | 2 | 0.13912 | 0 | 41 | 122 | 1.8842 | 8.38E-07 | 18 | 72 | 0.74436 | 7.69E-07 |
| 4 | 2 | 0.2973 | 0 | 41 | 121 | 1.0826 | 7.92E-07 | 19 | 76 | 0.79387 | 6.68E-07 |
| 4 | 2 | 0.13099 | 0 | 45 | 134 | 0.70861 | 9.66E-07 | 23 | 92 | 1.2396 | 3.63E-06 |
| 4 | 2 | 0.17758 | 0 | 46 | 137 | 0.90572 | 7.96E-07 | 23 | 92 | 1.6058 | 9.61E-07 |
| 5 | 2 | 0.17743 | 0 | 46 | 137 | 0.71118 | 3.38E-07 | 26 | 104 | 1.586 | 3.93E-07 |
| 5 | 2 | 0.86693 | 1.05E-07 | 42 | 125 | 0.75264 | 8.08E-07 | 20 | 80 | 1.0036 | 7.60E-07 |
| **Additional information**

No additional information is available for this paper.

**Acknowledgements**

The authors acknowledge the financial support provided by the Center of Excellence in Theoretical and Computational Science (TaCS-CoE), KMUTT.

**Appendix**

See Tables 2-11.
Table 3. Computational results for Problem 2.

| Dim | Inv | Iter | Fval | TIME | NORM | Inv2 | Iter2 | Fval2 | TIME2 | NORM2 | Inv3 | Iter3 | Fval3 | TIME3 | NORM3 | Inv4 | Iter4 | Fval4 | TIME4 | NORM4 | Inv5 | Iter5 | Fval5 | TIME5 | NORM5 | Inv6 | Iter6 | Fval6 | TIME6 | NORM6 |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 1000 | 5 | 6 | 0.074i19 | 3.1E+09 | 68 | 253 | 0.026761 | 9.14E-07 | 15 | 60 | 0.023736 | 4.9E-07 | 7 | 22 | 0.02528 | 5.63E-07 | 12 | 47 | 0.02684 | 9.63E-07 | 14 | 55 | 0.03041 | 1.26E-06 |
| 5000 | 5 | 6 | 0.068i45 | 5.4E-07 | 78 | 232 | 0.016532 | 6.3E-08 | 7 | 18 | 0.019547 | 5.4E-07 | 5 | 12 | 0.02218 | 2.0E-07 | 4 | 11 | 0.02236 | 4.6E-07 | 14 | 55 | 0.02684 | 9.63E-07 | 14 | 55 | 0.03041 | 1.26E-06 |
| 10000 | 5 | 6 | 0.068i45 | 5.4E-07 | 78 | 232 | 0.016532 | 6.3E-08 | 7 | 18 | 0.019547 | 5.4E-07 | 5 | 12 | 0.02218 | 2.0E-07 | 4 | 11 | 0.02236 | 4.6E-07 | 14 | 55 | 0.02684 | 9.63E-07 | 14 | 55 | 0.03041 | 1.26E-06 |

Table 4. Computational results for Problem 3.

| Dim | Inv | Iter | Fval | TIME | NORM | Inv2 | Iter2 | Fval2 | TIME2 | NORM2 | Inv3 | Iter3 | Fval3 | TIME3 | NORM3 | Inv4 | Iter4 | Fval4 | TIME4 | NORM4 | Inv5 | Iter5 | Fval5 | TIME5 | NORM5 | Inv6 | Iter6 | Fval6 | TIME6 | NORM6 |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 1000 | 5 | 6 | 0.074i19 | 3.1E+09 | 68 | 253 | 0.026761 | 9.14E-07 | 15 | 60 | 0.023736 | 4.9E-07 | 7 | 22 | 0.02528 | 5.63E-07 | 12 | 47 | 0.02684 | 9.63E-07 | 14 | 55 | 0.03041 | 1.26E-06 |
| 5000 | 5 | 6 | 0.068i45 | 5.4E-07 | 78 | 232 | 0.016532 | 6.3E-08 | 7 | 18 | 0.019547 | 5.4E-07 | 5 | 12 | 0.02218 | 2.0E-07 | 4 | 11 | 0.02236 | 4.6E-07 | 14 | 55 | 0.02684 | 9.63E-07 | 14 | 55 | 0.03041 | 1.26E-06 |
| 10000 | 5 | 6 | 0.068i45 | 5.4E-07 | 78 | 232 | 0.016532 | 6.3E-08 | 7 | 18 | 0.019547 | 5.4E-07 | 5 | 12 | 0.02218 | 2.0E-07 | 4 | 11 | 0.02236 | 4.6E-07 | 14 | 55 | 0.02684 | 9.63E-07 | 14 | 55 | 0.03041 | 1.26E-06 |

A.B. Abubakar, P. Kumam, A.H. Ibrahim. Heliyon 6(2020)e05400

HSDY CGD PDY ACGD for Problem 2.
Table 5. Computational results for Problem 4.

| DIM | INP | ITER | FVAL | TIME | NORM | INP | ITER | FVAL | TIME | NORM | INP | ITER | FVAL | TIME | NORM |
|-----|-----|------|------|------|------|-----|------|------|------|------|-----|------|------|------|------|
| 1 000 | 1 | 2 | 0.02449 | 6 | 280 | 0.01932 | 15 | 20 | 0.00469 | 0 | 2 0.01280 | 0 | 2 0.01280 | 0 | 2 0.01280 | 0 |
| 2 0.160 | 0.01245 | 0 | 2 0.01280 | 0 | 2 0.01280 | 0 | 2 0.01280 | 0 |
| 3 0.101 | 0.01199 | 0 | 2 0.01280 | 0 | 2 0.01280 | 0 | 2 0.01280 | 0 |
| 4 0.07904 | 0 | 2 0.01280 | 0 | 2 0.01280 | 0 | 2 0.01280 | 0 |
| 5 0.00708 | 0 | 2 0.01280 | 0 | 2 0.01280 | 0 | 2 0.01280 | 0 |
| 7 0.00140 | 0 | 2 0.01280 | 0 | 2 0.01280 | 0 | 2 0.01280 | 0 |
| 6 0.03120 | 9.92E-07 | 2 0.01280 | 0 | 2 0.01280 | 0 | 2 0.01280 | 0 |
| 7 0.12750 | 0 | 2 0.01280 | 0 | 2 0.01280 | 0 | 2 0.01280 | 0 |
| 8 0.04625 | 3.8E-07 | 2 0.01280 | 0 | 2 0.01280 | 0 | 2 0.01280 | 0 |
| 9 0.17250 | 0 | 2 0.01280 | 0 | 2 0.01280 | 0 | 2 0.01280 | 0 |
| 10 0.02122 | 0 | 2 0.01280 | 0 | 2 0.01280 | 0 | 2 0.01280 | 0 |
| 2 0.160 | 0.03310 | 0 | 2 0.01280 | 0 | 2 0.01280 | 0 | 2 0.01280 | 0 |
| 3 0.08362 | 0 | 2 0.01280 | 0 | 2 0.01280 | 0 | 2 0.01280 | 0 |
| 4 0.05099 | 0 | 2 0.01280 | 0 | 2 0.01280 | 0 | 2 0.01280 | 0 |
| 5 0.03120 | 9.92E-07 | 2 0.01280 | 0 | 2 0.01280 | 0 | 2 0.01280 | 0 |
| 6 0.12750 | 0 | 2 0.01280 | 0 | 2 0.01280 | 0 | 2 0.01280 | 0 |
| 7 0.04625 | 3.8E-07 | 2 0.01280 | 0 | 2 0.01280 | 0 | 2 0.01280 | 0 |
| 8 0.17250 | 0 | 2 0.01280 | 0 | 2 0.01280 | 0 | 2 0.01280 | 0 |
| 9 0.02122 | 0 | 2 0.01280 | 0 | 2 0.01280 | 0 | 2 0.01280 | 0 |
| 10 0.13012 | 9.92E-07 | 2 0.01280 | 0 | 2 0.01280 | 0 | 2 0.01280 | 0 |

Table 6. Computational results for Problem 5.

| DIM | INP | ITER | FVAL | TIME | NORM | INP | ITER | FVAL | TIME | NORM | INP | ITER | FVAL | TIME | NORM |
|-----|-----|------|------|------|------|-----|------|------|------|------|-----|------|------|------|------|
| 1 000 | 1 | 2 | 0.25200 | 5 | 280 | 0.25200 | 28 | 270 | 0.25200 | 28 | 270 | 0.25200 | 28 | 270 | 0.25200 | 28 |
| 2 0.06067 | 0 | 2 0.25200 | 28 | 270 | 0.25200 | 28 | 270 | 0.25200 | 28 | 270 | 0.25200 | 28 | 270 | 0.25200 | 28 |
| 3 0.10047 | 0 | 2 0.25200 | 28 | 270 | 0.25200 | 28 | 270 | 0.25200 | 28 | 270 | 0.25200 | 28 | 270 | 0.25200 | 28 |
| 4 0.08382 | 0 | 2 0.25200 | 28 | 270 | 0.25200 | 28 | 270 | 0.25200 | 28 | 270 | 0.25200 | 28 | 270 | 0.25200 | 28 |
| 5 0.09309 | 0 | 2 0.25200 | 28 | 270 | 0.25200 | 28 | 270 | 0.25200 | 28 | 270 | 0.25200 | 28 | 270 | 0.25200 | 28 |
| 6 0.0776 | 2.96E-07 | 2 0.25200 | 28 | 270 | 0.25200 | 28 | 270 | 0.25200 | 28 | 270 | 0.25200 | 28 | 270 | 0.25200 | 28 |
| 7 0.13242 | 0 | 2 0.25200 | 28 | 270 | 0.25200 | 28 | 270 | 0.25200 | 28 | 270 | 0.25200 | 28 | 270 | 0.25200 | 28 |
| 8 0.32 | 3.15E-07 | 2 0.25200 | 28 | 270 | 0.25200 | 28 | 270 | 0.25200 | 28 | 270 | 0.25200 | 28 | 270 | 0.25200 | 28 |
| 9 0.19472 | 0 | 2 0.25200 | 28 | 270 | 0.25200 | 28 | 270 | 0.25200 | 28 | 270 | 0.25200 | 28 | 270 | 0.25200 | 28 |
| 10 0.29634 | 0 | 2 0.25200 | 28 | 270 | 0.25200 | 28 | 270 | 0.25200 | 28 | 270 | 0.25200 | 28 | 270 | 0.25200 | 28 |
| 11 0.22122 | 0 | 2 0.25200 | 28 | 270 | 0.25200 | 28 | 270 | 0.25200 | 28 | 270 | 0.25200 | 28 | 270 | 0.25200 | 28 |
| 12 0.13933 | 0 | 2 0.25200 | 28 | 270 | 0.25200 | 28 | 270 | 0.25200 | 28 | 270 | 0.25200 | 28 | 270 | 0.25200 | 28 |
| 13 0.16287 | 4.96E-07 | 2 0.25200 | 28 | 270 | 0.25200 | 28 | 270 | 0.25200 | 28 | 270 | 0.25200 | 28 | 270 | 0.25200 | 28 |

10
### Table 7. Computational results for Problem 6.

|          | HSDY   | CGD    | PDET   | ACED   |
|----------|--------|--------|--------|--------|
| DIM INP | ITER   | FVAL   | TIME   | NORM   |
| 1000    |        |        |        |        |
| x1      | 30     | 0.037772 | 2.69E-05 |        |
| x2      | 60     | 0.008853 | 7.06E-06 |        |
| x3      | 90     | 0.010448 | 2.48E-05 |        |
| x4      | 90     | 0.020807 | 4.35E-05 |        |
| x5      | 90     | 0.004935 | 8.22E-05 |        |
| x6      | 90     | 0.014772 | 9.05E-06 |        |
| x7      | 250    | 0.016414 | 9.25E-05 |        |
| x8      | 90     | 0.025601 | 8.77E-05 |        |
| 5000    |        |        |        |        |
| x1      | 20     | 0.105141 | 7.26E-06 |        |
| x2      | 90     | 0.022236 | 2.48E-06 |        |
| x3      | 90     | 0.010156 | 1.43E-06 |        |
| x4      | 90     | 0.068638 | 9.23E-06 |        |
| x5      | 80     | 0.044959 | 8.34E-06 |        |
| x6      | 92     | 0.006759 | 1.19E-07 |        |
| x7      | 288    | 0.071276 | 4.35E-05 |        |
| x8      | 104    | 0.078859 | 2.00E-05 |        |
| 10000   |        |        |        |        |
| x1      | 101    | 0.41584 | 3.06E-06 |        |
| x2      | 106    | 0.41431 | 2.06E-06 |        |
| x3      | 84     | 0.37436 | 5.26E-06 |        |
| x4      | 102    | 0.25206 | 8.10E-05 |        |
| x5      | 112    | 0.43632 | 7.57E-05 |        |
| x6      | 99     | 0.63375 | 5.27E-05 |        |
| x7      | 50     | 0.54864 | 2.57E-05 |        |
| x8      | 70     | 0.6123 | 1.15E-05 |        |
| x9      | 101    | 1.0136 | 9.05E-05 |        |
| x10     | 50     | 0.96564 | 3.49E-05 |        |
| x11     | 38     | 0.017721 | 6.45E-05 |        |
| x12     | 48     | 0.010764 | 6.46E-05 |        |
| 5000    |        |        |        |        |
| x1      | 32     | 0.97143 | 4.62E-05 |        |
| x2      | 30     | 0.41584 | 3.06E-06 |        |
| x3      | 70     | 0.41431 | 2.06E-06 |        |
| x4      | 84     | 0.37436 | 5.26E-06 |        |
| x5      | 102    | 0.25206 | 8.10E-05 |        |
| x6      | 112    | 0.43632 | 7.57E-05 |        |
| x7      | 99     | 0.63375 | 5.27E-05 |        |
| x8      | 50     | 0.54864 | 2.57E-05 |        |
| x9      | 70     | 0.6123 | 1.15E-05 |        |
| x10     | 101    | 1.0136 | 9.05E-05 |        |
| x11     | 38     | 0.017721 | 6.45E-05 |        |
| x12     | 48     | 0.010764 | 6.46E-05 |        |

### Table 8. Computational results for Problem 7.

|          | HSDY   | CGD    | PDET   | ACED   |
|----------|--------|--------|--------|--------|
| DIM INP | ITER   | FVAL   | TIME   | NORM   |
| 1000    |        |        |        |        |
| x1      | 30     | 0.049014 | 3.39E-05 |        |
| x2      | 60     | 0.007063 | 2.54E-05 |        |
| x3      | 90     | 0.016414 | 9.25E-05 |        |
| x4      | 90     | 0.025601 | 8.77E-05 |        |
| 5000    |        |        |        |        |
| x1      | 20     | 0.105141 | 7.26E-06 |        |
| x2      | 90     | 0.022236 | 2.48E-06 |        |
| x3      | 90     | 0.010156 | 1.43E-06 |        |
| x4      | 90     | 0.068638 | 9.23E-06 |        |
| x5      | 80     | 0.044959 | 8.34E-06 |        |
| x6      | 92     | 0.006759 | 1.19E-07 |        |
| x7      | 288    | 0.071276 | 4.35E-05 |        |
| x8      | 104    | 0.078859 | 2.00E-05 |        |
| 10000   |        |        |        |        |
| x1      | 101    | 0.41584 | 3.06E-06 |        |
| x2      | 106    | 0.41431 | 2.06E-06 |        |
| x3      | 84     | 0.37436 | 5.26E-06 |        |
| x4      | 102    | 0.25206 | 8.10E-05 |        |
| x5      | 112    | 0.43632 | 7.57E-05 |        |
| x6      | 99     | 0.63375 | 5.27E-05 |        |
| x7      | 50     | 0.54864 | 2.57E-05 |        |
| x8      | 70     | 0.6123 | 1.15E-05 |        |
| x9      | 101    | 1.0136 | 9.05E-05 |        |
| x10     | 50     | 0.96564 | 3.49E-05 |        |
| x11     | 38     | 0.017721 | 6.45E-05 |        |
| x12     | 48     | 0.010764 | 6.46E-05 |        |

11
### Table 9. Computational results for Problem 8.

| DIM | INITIAL | ITER | VVAL | TIME | NORM | PVAL | TIME | NORM | QVAL | TIME | NORM | RVAL | TIME | NORM |
|-----|---------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| 1000 | 3 | 28 | 0.046001 | 3.186E-07 | 37 | 110 | 0.031526 | 6.785E-07 | 17 | 68 | 0.031775 | 4.34E-07 | 9 | 35 | 0.069894 | 2.46E-06 |
| 5000 | 3 | 28 | 0.007813 | 1.875E-07 | 37 | 110 | 0.016632 | 6.875E-07 | 17 | 68 | 0.017775 | 4.34E-07 | 9 | 35 | 0.096689 | 3.91E-06 |
| 10000 | 3 | 24 | 0.001969 | 1.295E-07 | 30 | 89 | 0.009618 | 6.515E-07 | 5 | 20 | 0.003465 | 6.50E-08 | 8 | 31 | 0.000874 | 7.43E-06 |
| 50000 | 3 | 28 | 0.006787 | 4.985E-08 | 38 | 113 | 0.013524 | 8.05E-08 | 17 | 68 | 0.014285 | 8.62E-07 | 11 | 43 | 0.009853 | 5.94E-06 |
| 100000 | 3 | 28 | 0.005169 | 5.465E-08 | 38 | 113 | 0.018328 | 8.05E-08 | 17 | 68 | 0.017281 | 8.06E-07 | 11 | 43 | 0.006471 | 9.87E-06 |

### Table 10. Computational results for Problem 9.

| DIM | INITIAL | ITER | VVAL | TIME | NORM | PVAL | TIME | NORM | QVAL | TIME | NORM | RVAL | TIME | NORM |
|-----|---------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| 1000 | 3 | 28 | 0.024313 | 7.115E-07 | 39 | 116 | 0.050791 | 3.81E-08 | 18 | 72 | 0.046513 | 5.59E-07 | 9 | 39 | 0.038315 | 5.94E-06 |
| 5000 | 3 | 28 | 0.024353 | 4.195E-07 | 38 | 113 | 0.039456 | 9.66E-08 | 17 | 68 | 0.035141 | 9.76E-07 | 9 | 35 | 0.022551 | 8.74E-06 |
| 10000 | 3 | 28 | 0.003845 | 6.525E-08 | 31 | 92 | 0.051271 | 9.18E-08 | 5 | 20 | 0.020467 | 1.01E-07 | 9 | 35 | 0.020604 | 4.01E-05 |
| 50000 | 4 | 32 | 0.002307 | 4.615E-08 | 40 | 76 | 0.039804 | 7.05E-08 | 17 | 69 | 0.067282 | 7.14E-07 | 12 | 47 | 0.031536 | 3.21E-05 |
| 100000 | 3 | 31 | 0.001155 | 7.913E-08 | 40 | 76 | 0.036192 | 7.05E-08 | 20 | 80 | 0.123868 | 6.82E-07 | 12 | 47 | 0.036847 | 8.46E-06 |
| 500000 | 5 | 31 | 0.001427 | 3.466E-08 | 39 | 115 | 0.050336 | 7.94E-08 | 17 | 75 | 0.052438 | 4.22E-07 | 12 | 46 | 0.032452 | 4.63E-06 |
| 1000000 | 5 | 32 | 0.040993 | 4.186E-08 | 40 | 119 | 0.095685 | 7.39E-08 | 17 | 72 | 0.001337 | 7.90E-07 | 10 | 39 | 0.058469 | 7.77E-05 |

---

12
References

[1] A.B. Abubakar, A.H. Ibrahim, A.B. Muhammad, C. Tammer, A modified descent d-yaun conjugate gradient method for constraint nonlinear monotone operator equations, Appl. Anal. Optim. 4 (1) (2020) 1–24.

[2] A.B. Abubakar, P. Kumam, H. Mammad, A.M. Awalii, A Barzilai-Borwein gradient projection method for sparse signal and blurred image restoration, J. Franklin Inst. 357 (11) (2020) 7266–7295.

[3] Awula Bala Abo, Poom Kumam, An improved three-term derivative-free method for solving nonlinear equations, Comput. Appl. Math. 37 (5) (2018) 6760–6773.

[4] Awula Bala Abo, Poom Kumam, A descent d-yaun conjugate gradient method for nonlinear equations, Numer. Algorithms 81 (1) (2019) 197–210.

[5] Awula Bala Abo, Poom Kumam, Hassan Mohammad, Aliyu Muhammad Awalii, An efficient conjugate gradient method for convex constrained monotone nonlinear equations with applications, Mathematics 7 (9) (2019) 767.

[6] Awula Bala Abo, Poom Kumam, Hassan Mohammad, Aliyu Muhammad Awalii, Kanokwan Sibbitthakherkhet, A modified Fletcher-Reeves conjugate gradient method for monotonic nonlinear equations with some applications, Mathematics 7 (8) (2019) 745.

[7] Awula Bala Abo, Jeawaidi Bilwan, Sefis Endris Yimer, Abdulkarim Hassan Ibrahim, Idris Ahmed, Spectral three-term conjugate descent method for solving monotone nonlinear equations with convex constraints, Thai J. Math. 18 (1) (2020) 501–517.

[8] S. Ajì, P. Kumam, P. Siricharoen, A.B. Abubakar, M.M. Yahiya, A modified conjugate gradient descent projection method for monotone nonlinear equations and image restoration, IEEE Access 8 (2020) 158656–158662.

[9] A.M. Awalii, P. Kumam, H. Mammad, W. Wathayu, A.B. Abubakar, A Perry-type derivative-free algorithm for solving nonlinear system of equations and minimizing $f$, regularized problem, Optimization (2020) 1–29.

[10] Aliyu Muhammad Awali, Poom Kumam, Awula Bala Abo, A modified conjugate gradient method for monotone nonlinear equations with convex constraints, Appl. Numer. Math. 145 (2019) 507–520.

[11] Aliyu Muhammad Awali, Poom Kumam, Awula Bala Abo, Spectral modified polak-ribiere-polyak projection conjugate gradient method for solving monotone systems of nonlinear equations, Appl. Math. Comput. 361 (2019) 124514.

[12] Y. Bing, G. Lin, An efficient implementation of merril's method for sparse or partially separable systems of nonlinear equations, SIAM J. Optim. 1 (2) (1991) 206–221.

[13] E.G. Birgin, J.M. Martinez, A spectral conjugate gradient method for unconstrained optimization, Appl. Math. Optim. 43 (2) (2001) 117–128.

[14] Yanyun Ding, Yunhai Xiao, Jianwei Li, A class of conjugate gradient methods for convex constrained monotone equations, Optimization 66 (12) (2017) 2309–2328.

[15] Elizabeth D. Dolan, Jorge J. Moré, Benchmarking optimization software with performance profiles, Math. Program. 91 (2) (2002) 201–213.

[16] Dingxiang Feng, Min Sheng, Xiaoyong Wang, A generalized conjugate gradient method for large-scale nonlinear equations, J. Inequil. Appl. 17 (1) (2017) Sep 236.

[17] Mário AT Figueredo, Robert D. Nowak, Stephen J. Wright, Gradient projection for sparse reconstruction: convex versus non-convex approaches and other inverse problems, IEEE J. Sel. Top. Signal Process. 1 (4) (2007) 586–597.

[18] Abdulkarim Hassan Ibrahim, Abir Isah Garba, Halima Usman, Jamilu Abubakar, Awula Bala Abubakar, Derivative-free projection algorithm for nonlinear equations with convex constraints, Thai J. Math. 18 (1) (2019).

[19] Abdulkarim Hassan Ibrahim, Poom Kumam, Awula Bala Abubakar, Jamilu Abubakar Bakoji Muhammad, Least-square-based three-term conjugate gradient projection method for $f$, norm problems with application to compressed sensing, Mathematics 8 (4) (2020).

[20] Abdulkarim Hassan Ibrahim, Poom Kumam, Awula Bala Abubakar, Wachirapong Jirakittipanwatt, Jamila Abubakar, A hybrid conjugate gradient algorithm for constrained monotone equations with application in compressive sensing, Heliyon 6 (3) (2020) e03466.

[21] Abdulkarim Hassan Ibrahim, Poom Kumam, Awula Bala Abubakar, Umar Batsari Yusuf, Jeawaidi Bilwan, Derivative-free conjugate residual algorithms for convex constrained nonlinear equations system and signal recovery, J. Nonlinear Convex Anal. 21 (9) (2020) 1959–1972.

[22] Abdulkarim Hassan Ibrahim, Poom Kumam, Wiyada Kumana, A family of derivative-free conjugate gradient methods for constrained nonlinear equations and image restoration, IEEE Access 8 (2020) 1094–1099.

[23] W. La Cruz, J. Martínez, M. Raydan, Spectral residual method without gradient information for solving large-scale nonlinear systems of equations, Math. Comput. 75 (255) (2006) 1429–1448.

[24] William La Cruz, A spectral algorithm for large-scale systems of nonlinear monotone equations, Numer. Algorithms 76 (4) (2017) 1109–1130.

[25] Dong-Hui Li, Masuo Fukushima, A modified biggs method and its global convergence in nonconvex minimization, in: Nonlinear Programming and Variational Inequalities, J. Comput. Appl. Math. 129 (1) (2001) 15–55.

[26] J.K. Liu, S.J. Li, A three-term derivative-free projection method for nonlinear monotone system of equations, Calcolo 53 (3) (2016) 427–450.

[27] Jinkui Liu, Yuming Feng, A derivative-free iterative method for nonlinear monotone equations with convex constraints, Numer. Algorithms (2018) 1–18.

[28] Jing-Shang Pang, Inexact Newton methods for the nonlinear complementarity problem, Math. Program. 58 (1–3) (1993) 149–174.

[29] Zoltan Papp, Sanja Rajapak, Fr type methods for systems of large-scale nonlinear equations, Appl. Math. Comput. 269 (2015) 816–823.

[30] Chuanchai Wang, Vijay Wang, Chuanliang Xu, A projection method for a system of nonlinear monotone equations with convex constraints, Math. Methods Oper. Res. 66 (1) (2007) 33–46.

[31] Yuhai Xiao, Qiuyi Wang, Qingjie Hu, Non-smooth equations based method for $f$, norm problems with applications to compressed sensing, Nonlinear Anal., Theory Methods Appl. 74 (11) (2011) 3570–3577.

[32] Yuhai Xiao, Hong Zhu, A conjugate gradient method to solve convex constrained monotone equations with applications in compressive sensing, J. Math. Anal. Appl. 405 (1) (2013) 310–319.
[33] Z. Yu, J. Lin, J. Sun, Y.H. Xiao, L.Y. Liu, Z.H. Li, Spectral gradient projection method for monotone nonlinear equations with convex constraints, Appl. Numer. Math. 59 (10) (2009) 2416–2423.

[34] Gonglin Yuan, Tingting Li, Wujie Hu, A conjugate gradient algorithm for large-scale nonlinear equations and image restoration problems, Appl. Numer. Math. 147 (2020) 129–141.

[35] Gonglin Yuan, Junyu Lu, Zhan Wang, The prp conjugate gradient algorithm with a modified wwp line search and its application in the image restoration problems, Appl. Numer. Math. 152 (2020) 1–11.

[36] Weijun Zhou, Donghui Li, Limited memory bfgs method for nonlinear monotone equations, J. Comput. Math. (2007) 89–96.