Mathematics

http://www.aimspress.com/journal/Math

AIMS Mathematics, 11(4): 10908–10935.
DOI: 10.3934/math.2026448
Received: 14 December 2025
Revised: 16 March 2026
Accepted: 31 March 2026
Published: 20 April 2026

Research article

Accelerated double step-length method for solving monotone nonlinear equations with convex-constraint and application

Muhammad Abdullahi[1,2], Abubakar Sani Halilu[2,3], Mohammed A. Saleh[4], Abdulgader Z. Almaymuni[4,*], Seyed Yaser Mousavi Siamakani[5,*], Tiamiyu Abd'gafar Tunde[6] and Sulaiman Mohammed Ibrahim[7]

[1] School of Mathematics and Statistics, HNP-LAMA, Central South University, Changsha, Hunan 410083, China
[2] Department of Mathematics, Sule Lamido University Kafin Hausa, Nigeria
[3] Faculty of Informatics and Computing, Universiti Sultan Zainal Abidin, Kuala Terengganu 21300, Malaysia
[4] Department of Cybersecurity, College of Computer, Qassim University, Saudi Arabia
[5] Department of Civil Engineering, College of Engineering, Rangsit University, Mueang, Pathum Thani 12000, Thailand
[6] Department of Mathematics, The Chinese University of Hong Kong, Hong Kong, China
[7] College of Applied and Health Sciences, A'Sharqiyah University, Ibra 400, Sultanate of Oman

* **Correspondence:** Email: almaymuni@qu.edu.sa; yaser.m@rsu.ac.th.

Abstract: The hybrid procedure is an efficient technique for improving the global and numerical performance of iterative algorithms for large-scale monotone nonlinear problems. This is achieved by integrating two or more methods into a unified framework. In this study, we present a hybrid of the double step-length method and the Picard-Mann iterative technique for solving monotone nonlinear equations with convex constraints. By combining the Picard-Mann approach with a newly proposed scheme, we obtain an iterative method that reduces computational cost and achieves faster convergence. The acceleration parameter is determined by evaluating the difference between the Broyden update and its approximation using the Frobenius norm. The global convergence of the proposed method is proved, and a Q-linear convergence rate is also established. Numerical experiments demonstrate that the proposed approach is computationally efficient for solving large-scale nonlinear equations compared with existing methods. Finally, the method is applied to signal processing and image restoration problems, highlighting its practical relevance.

Keywords: Frobenius norm; Picard-Mann; optimization method; signal reconstruction; double step length
Mathematics Subject Classification: 65K05, 90C52, 90C53, 90C56

1. Introduction

Many scientific and engineering fields rely on nonlinear equations; for further background, see [1–3]. Such equations take the general form

$$E(x) = 0, \quad x \in P, \tag{1.1}$$

where the set $P \subseteq \mathbb{R}^n$ is nonempty, closed, and convex and $E : P \to \mathbb{R}^n$ is a nonlinear mapping. Here, $\mathbb{R}^n$ denotes the n-dimensional real space equipped with the Euclidean norm $\|\cdot\|$. To solve (1.1), several researchers have developed iterative methods; see [4–7] for broad discussions of available techniques. Traditional approaches include Newton, quasi-Newton, Gauss-Newton, Levenberg-Marquardt, and trust-region methods. Among these, the Newton method and its variants are widely appreciated for their fast convergence and ease of implementation.

However, these methods are not suitable for large-scale nonlinear problems because computing the search direction requires the Jacobian matrix, that is, $d_k = -(J_E(x_k))^{-1}E(x_k)$, where $J_E(x_k)$ denotes the Jacobian of $E(x_k)$ at x_k; see [8, 9]. A major weakness of these methods is the need to compute or invert the Jacobian matrix at each iteration, which can be computationally expensive or infeasible in large-scale settings.

These limitations have motivated many researchers to address the weaknesses of the Newton method. For instance, Stanimirović et al. [10] proposed an accelerated gradient-descent method that simplifies Hessian computations by approximating the Hessian with a diagonal matrix. This approach transforms the classical Newton framework into a gradient-based scheme for large-scale unconstrained optimization. In addition, Petrović et al. [11], building on [12], proposed a double step-length method for large-scale unconstrained optimization. Their method employs an inexact line-search strategy and a suitably designed diagonal approximation of the Hessian matrix. Inspired by [11], Halilu et al. [13] introduced a double step-length method for systems of nonlinear equations. Their method uses an acceleration parameter to approximate the Jacobian matrix and applies an inexact line search to compute the step length. Global convergence of the proposed method was established under certain assumptions.

Moreover, improving numerical performance and global convergence in iterative approaches often requires hybridizing two or more techniques to obtain a more robust method. Such hybridization is used to enhance both convergence behavior and overall computational efficiency. For example, inspired by [11], Petrović [14] proposed a hybrid acceleration technique with two step-length parameters for large-scale unconstrained optimization. In that work, the double step-length strategy guides a Picard-Mann-type hybrid iterative scheme, and convergence was demonstrated for both strictly convex and uniformly convex cases. Similarly, Petrović et al. [15] proposed a transformed double step-length method by hybridizing it with a three-term technique for unconstrained optimization. Their process incorporates an adaptive backtracking line search with an optimally computed initial value to improve efficiency, and the method was shown to perform well for both strictly convex and uniformly convex quadratic functions. In addition, Abbass et al. [16] proposed a double step-length method for convex-constrained nonlinear equations. Their approach combines the Picard-Krasnoselskii iterative process with double step-length techniques, and the resulting algorithm satisfies global convergence under mild assumptions; it was later applied to signal-processing problems. More recently, Halilu et al. [17] proposed a hybrid two-step method for systems of nonlinear equations with applications in motion control. Their method incorporates the Picard-Mann hybrid procedure in [18] with two correction terms during iteration. The motivation is to stabilize the process when one correction parameter fails, thereby preventing divergence of the search direction.

To the best of our knowledge, the only existing double step-length method for solving (1.1) is the one presented in [16]. Although pioneering, that approach relies on a traditional Taylor-series expansion to derive the acceleration parameter for the second step. For additional related works, see [16, 17, 19–21]. While effective, this strategy inherits local-convergence assumptions and sensitivity to Jacobian approximations that are typical of series-expansion techniques. Moreover, most other double step-length methods in the literature are designed either for unconstrained optimization problems or for general systems of nonlinear equations. Motivated by the favorable properties of Jacobian approximation via the Frobenius norm, the hybrid iterative framework in [18], and the projection technique in [22], we propose a hybrid of the double step-length method and the Picard-Mann iterative technique for solving large-scale convex-constrained monotone nonlinear equations, with applications in signal processing. These results highlight the ability of the hybrid approach to improve both global and numerical convergence of the iterative scheme and also provide a basis for further developments in this area. The primary contributions of this study are summarized as follows:

- This study introduces a new hybrid iterative procedure for solving large-scale convex-constrained monotone nonlinear equations using the Picard-Mann iterative technique.
- The acceleration parameter in the proposed method is derived using the Frobenius norm to minimize the difference between the Broyden update and its diagonal approximation.
- The proposed search direction satisfies sufficient descent without additional conditions and guarantees global convergence with a Q-linear convergence rate.
- Numerical simulations confirm that the correction parameter derived from the Picard-Mann iterative scheme is effective, thereby improving both numerical performance and overall convergence of the proposed method.
- The proposed method is further applied to signal-processing problems, where it demonstrates accurate recovery of degraded signals and practical relevance in scientific applications.

The paper is structured as follows: Section 2 provides an overview of the proposed algorithm, its motivation, and preliminaries. Section 3 presents the convergence analysis of the proposed algorithm. Section 4 reports numerical simulations. Section 5 presents signal-processing applications. Section 6 concludes the paper.

2. Motivation and algorithm description

The hybrid framework, viewed as a relational iterative procedure, was first introduced by Picard [23] and later developed by Mann [24]. This approach generalizes the classical Picard iteration by introducing a sequence of weighting parameters that form convex combinations of successive iterates. In addition, Ishikawa [25] proposed another iterative scheme by extending the method of Mann [24]. Subsequently, Khan [18] proposed a hybrid of the methods in [23] and [24]; see also [18] for further details. Motivated by the iterative procedures in [18, 23–25], we propose another double step-length iterative technique for solving large-scale convex-constrained nonlinear equations.

Next, we define another hybrid version of the accelerated double step-size as follows:

$$T(x_k) = x_k - \overline{\beta}_{(k)}\phi_k^{-1}E(x_k) - \overline{\eta}_{(k)}E(x_k), \tag{2.1}$$

where x_k is the current iterate and $\overline{\beta}_{(k)}$ and $\overline{\eta}_{(k)}$ are the two respective step sizes computed using a suitable line-search procedure, while ϕ_k is the acceleration parameter to be obtained later. By combining the Picard-Mann hybridization process proposed by Khan [18] with the expression in (2.1), we obtain the following relations:

$$\begin{cases} x_0 = x \in P, \\ x_{k+1} = T(y_k) = y_k - \overline{\beta}_{(k)}\phi_k^{-1}E(x_k) - \overline{\eta}_{(k)}E(x_k), \\ y_k = (1-\overline{\gamma}_{(k)})x_k + \overline{\gamma}_{(k)}(x_k - \overline{\beta}_{(k)}\phi_k^{-1}E(x_k) - \overline{\eta}_{(k)}E(x_k)), \quad k \geq 0, \\ \quad = x_k - \overline{\gamma}_{(k)}(\overline{\beta}_{(k)}\phi_k^{-1} + \overline{\eta}_{(k)})E(x_k), \end{cases} \tag{2.2}$$

where the parameter $\overline{\gamma}_{(k)} \in (0, 1)$ is the correction parameter serving as the means to modify the iterative procedure.

Lemma 2.1. *Suppose the hybrid iterative process can be generated by the iterative technique* (2.2)*, then, the iterative scheme is defined as follows:*

$$x_{k+1} = x_k - (\overline{\gamma}_{(k)} + 1)(\overline{\beta}_{(k)}\phi_k^{-1} + \overline{\eta}_{(k)})E(x_k). \tag{2.3}$$

Note that the iterative scheme in (2.3) is derived through Picard-Mann hybridization together with a newly defined function in (2.1). The parameters $\beta_{(k)}$ and $\eta_{(k)}$ adapt the step length by combining curvature information with stabilization. In addition, the acceleration parameter $\gamma_{(k)}$ controls the extent of hybridization to enforce sufficient descent, providing a structural advantage over existing double step-length methods. This decomposition of steps is designed to improve both convergence and stability.

2.1. Derivation of acceleration parameter via Frobenius norm

In this subsection, we focus on deriving the acceleration parameter using the Frobenius norm. The acceleration parameter ϕ_k is motivated by a scalar quasi-Newton philosophy. We approximate the Broyden matrix B_k by a diagonal matrix $\phi_k I$ through Frobenius-norm minimization, where I is the identity matrix. This is equivalent to asking: Which scaling factor ϕ_k best captures the action of the Jacobian-like matrix B_k across all directions? For this purpose, we consider the following Broyden update.

Let $s_{k-1} = x_k - x_{k-1}$ and $y_{k-1} = E(x_k) - E(x_{k-1}) + \gamma s_{k-1}$, with $\gamma > 0$, and assume $B_{k-1} \approx \phi_{k-1}I$. Then, the Broyden update B_k is given by

$$B_k = B_{k-1} + \frac{(y_{k-1} - B_{k-1}s_{k-1})s_{k-1}^T}{s_{k-1}^T s_{k-1}}, \tag{2.4}$$

where B_k is an $n \times n$ matrix.

This scalar approximation is motivated by computational efficiency. To obtain an efficient acceleration parameter ϕ_k, we formulate the following Frobenius-norm minimization problem:

$$\min_{\phi_k \in \mathbb{R}} \|B_k - \phi_k I\|_F^2 = \min_{\phi_k} \operatorname{trace}\left((B_k - \phi_k I)^\top (B_k - \phi_k I)\right). \tag{2.5}$$

Substituting the approximation into the Broyden update in (2.4), we obtain $B_k = \phi_{k-1}I + \frac{(y_{k-1}-\phi_{k-1}s_{k-1})s_{k-1}^\top}{s_{k-1}^\top s_{k-1}}$. Further simplification of (2.5) implies

$$\|B_k - \phi_k I\|_F^2 = \left\|(\phi_{k-1}-\phi_k)I + \frac{(y_{k-1}-\phi_{k-1}s_{k-1})s_{k-1}^\top}{s_{k-1}^\top s_{k-1}}\right\|_F^2. \tag{2.6}$$

From (2.6), we define

$$\vartheta(\phi_k) = \operatorname{trace}\left(\left((\phi_{k-1}-\phi_k)I + \frac{(y_{k-1}-\phi_{k-1}s_{k-1})s_{k-1}^\top}{s_{k-1}^\top s_{k-1}}\right)^\top \left((\phi_{k-1}-\phi_k)I + \frac{(y_{k-1}-\phi_{k-1}s_{k-1})s_{k-1}^\top}{s_{k-1}^\top s_{k-1}}\right)\right). \tag{2.7}$$

Let $\kappa_1 = \phi_{k-1} - \phi_k$ and $\kappa_2 = \frac{y_{k-1}-\phi_{k-1}s_{k-1}}{s_{k-1}^\top s_{k-1}}$. Then, (2.7) transforms to

$$\vartheta(\phi_k) = \operatorname{trace}\left((\kappa_1 I + \kappa_2 s_{k-1}^\top)^\top(\kappa_1 I + \kappa_2 s_{k-1}^\top)\right). \tag{2.8}$$

Expanding (2.8) further, we obtain

$$\vartheta(\phi_k) = \operatorname{trace}\left(\kappa_1^2 I + \kappa_1\kappa_2 s_{k-1}^\top + \kappa_1 s_{k-1}\kappa_2^\top + \kappa_2 s_{k-1}^\top s_{k-1}\kappa_2^\top\right). \tag{2.9}$$

Using linearity and standard properties of the trace operator, we simplify each term as follows:

$$\operatorname{trace}(\kappa_1^2 I) = n\kappa_1^2,\quad \operatorname{trace}(\kappa_1\kappa_2 s_{k-1}^\top) = \kappa_1 s_{k-1}^\top\kappa_2,\quad \operatorname{trace}(\kappa_1 s_{k-1}\kappa_2^\top) = \kappa_1\kappa_2^\top s_{k-1} = \kappa_1 s_{k-1}^\top\kappa_2,$$
$$\operatorname{trace}(\kappa_2 s_{k-1}^\top s_{k-1}\kappa_2^\top) = (s_{k-1}^\top s_{k-1})(\kappa_2^\top\kappa_2).$$

Substituting these terms into (2.9), we obtain

$$\vartheta(\phi_k) = \kappa_1^2 n + 2\kappa_1(s_{k-1}^\top\kappa_2) + (s_{k-1}^\top s_{k-1})(\kappa_2^\top\kappa_2). \tag{2.10}$$

Substituting $\kappa_1 = \phi_{k-1} - \phi_k$ and $\kappa_2 = \frac{y_{k-1}-\phi_{k-1}s_{k-1}}{s_{k-1}^\top s_{k-1}}$ into (2.10) yields

$$\vartheta(\phi_k) = \left(n(\phi_{k-1}-\phi_k)^2 + 2(\phi_{k-1}-\phi_k)\frac{s_{k-1}^\top(y_{k-1}-\phi_{k-1}s_{k-1})}{s_{k-1}^\top s_{k-1}} + \frac{\|y_{k-1}-\phi_{k-1}s_{k-1}\|^2}{s_{k-1}^\top s_{k-1}}\right). \tag{2.11}$$

Finding the minimum of (2.5) is equivalent to differentiating (2.11) with respect to ϕ_k and setting the derivative to zero, which gives

$$\vartheta'(\phi_k) = \left(2n(\phi_{k-1}-\phi_k)(-1) + 2(-1)\frac{s_{k-1}^\top(y_{k-1}-\phi_{k-1}s_{k-1})}{s_{k-1}^\top s_{k-1}}\right) = 0. \tag{2.12}$$

Further simplification of (2.12) gives

$$-2n(\phi_{k-1}-\phi_k) - 2\frac{s_{k-1}^\top(y_{k-1}-\phi_{k-1}s_{k-1})}{s_{k-1}^\top s_{k-1}} = 0. \tag{2.13}$$

Remark 2.2. *To ensure $\phi_k > 0$ for all k when initialized with $\phi_0 = 1$, we examine the update as follows.*

Let $\rho_k = \dfrac{s_{k-1}^\top y_{k-1}}{s_{k-1}^\top s_{k-1}}$. Then, (2.13) *can be rewritten as*

$$\phi_k = \phi_{k-1} + \frac{1}{n}(\rho_k - \phi_{k-1}) = \frac{n-1}{n}\phi_{k-1} + \frac{1}{n}\rho_k.$$

This is a convex combination of ϕ_{k-1} and ρ_k, provided $n > 1$.

If $\rho_k > 0$ for all k, then ϕ_k remains positive because it is a weighted average of positive quantities. The condition $\rho_k > 0$ is equivalent to $s_{k-1}^\top y_{k-1} > 0$, which is the curvature condition *commonly satisfied in quasi-Newton methods for strictly convex functions. Thus, under this mild and standard condition, $\phi_k > 0$ for all k when $\phi_0 = 1$.*

Therefore, the derived acceleration parameter is given by

$$\phi_k = \phi_{k-1} + \frac{s_{k-1}^\top (y_{k-1} - \phi_{k-1} s_{k-1})}{n s_{k-1}^\top s_{k-1}}. \tag{2.14}$$

We further simplify the inverse of the acceleration parameter defined in (2.14) as follows:

$$\phi_k^{-1} = \frac{n s_{k-1}^\top s_{k-1}}{\phi_{k-1}(n-1) s_{k-1}^\top s_{k-1} + s_{k-1}^\top y_{k-1}}. \tag{2.15}$$

Remark 2.3. *We consider the following remark:*

- *(i) From* (2.3)*, for $k \geq 1$, the proposed search direction is given by*

  $$d_k = -(\overline{\gamma}_{(k)} + 1)\phi_k^{-1} E(x_k),$$

  and the general iterative scheme is

  $$x_{k+1} = x_k + \overline{\alpha}_{(k)} d_k,$$

  where $\overline{\alpha}_{(k)} = (\overline{\beta}_{(k)} + \overline{\eta}_{(k)}\phi_k)$. For $k = 0$, we initialize with $d_0 = -E(x_0)$ as stated in Algorithm 2 below.
- *(ii) The correction parameters satisfy $(\overline{\gamma}_{(k)} + 1) \in (1, 2)$, where $\overline{\gamma}_{(k)} \in (0, 1)$ for all $k > 0$, ensuring the efficiency and stability of the proposed iterative procedure.*

Before presenting the steps of the proposed method, we first define the line-search procedure used to compute the step-length in the algorithm. The computation of the step-lengths $\overline{\beta}_{(k)}$ and $\overline{\eta}_{(k)}$ is summarized in Algorithm 1 as follows.

Algorithm 1: Inexact line search for obtaining $\overline{\beta}_{(k)}$ and $\overline{\eta}_{(k)}$.

Input: Given the function $E(x_k)$, the search direction d_k, $\rho \in (0, 1)$, $\varpi \in (0, 1)$, $\xi > 0$, $\zeta > 0$ and $\sigma > 0$.

1. For $\overline{\eta} = \overline{\beta} = 1$.

2. While

$$-E(x_k + \overline{\alpha}_{(k)} d_k)^\top d_k \geq \sigma \overline{\alpha}_{(k)} \|d_k\|^2. \tag{2.16}$$

Let $\overline{\alpha}_{(k)} = (\overline{\beta}_{(k)} + \overline{\eta}_{(k)}\phi_k)$, where $\overline{\beta}_{(k)} := \zeta\varpi^{j_k}$ and $\overline{\eta}_{(k)} := \xi\rho^{j_k}$, with j_k being the smallest non-negative integer j such that (2.16) holds.

3. Return $\overline{\beta}_{(k)} = \zeta\varpi^{j_k}$ and $\overline{\eta}_{(k)} = \xi\rho^{j_k}$.

Remark 2.4. *Note that the line search defined in* (2.16) *differs from the classical Armijo line search in two main respects, despite its restrictive nature: (i) It simultaneously backtracks the two parameters $\beta_{(k)}$ and $\eta_{(k)}$ using independent factors ω and ρ, which reflect their respective roles in curvature scaling; and (ii) it directly enforces contraction of the residual norm $\|E(x)\|$, which is the natural merit function for nonlinear problems.*

The proposed method is summarized in Algorithm 2. The algorithm steps are used to conduct the numerical simulation.

Algorithm 2: Accelerated method for solving monotone nonlinear equations with application.

Input. Given $x_0 \in P$, $\psi \in (0, 2)$, $c > 0$, $tol > 0$, $\phi_0 = 1$, and set $k := 0$.

Step A: Evaluate $E(x_k)$. If $\|E(x_k)\| \leq tol$, then stop; else, go to **Step B**.

Step B. Compute the search direction as follows:

$$d_k = \begin{cases} -E(x_0), & \text{if} \quad k = 0, \\ -(\overline{\gamma}_{(k)} + 1)\phi_k^{-1}E(x_k), & \text{if} \quad k \geq 1. \end{cases} \tag{2.17}$$

Step C. Compute $\overline{\beta}_{(k)}$ and $\overline{\eta}_{(k)}$ using Algorithm 1 above.

Step D. Set $z_k = x_k + \overline{\alpha}_{(k)}d_k$, where $\overline{\alpha}_{(k)} = (\overline{\beta}_{(k)} + \overline{\eta}_{(k)}\phi_k)$. If $\|E(z_k)\| = 0$ or $\|E(z_k)\| \leq tol$, stop; else, go to **Step E**.

Step E. Evaluate next iteration by

$$x_{k+1} = P_P[x_k - \psi\delta_k E(z_k)], \tag{2.18}$$

where

$$\delta_k = \frac{E(z_k)^\top(x_k - z_k)}{\|E(z_k)\|^2}, \quad \|E(z_k)\| \neq 0. \tag{2.19}$$

Step F. Compute the acceleration parameter ϕ_{k+1} using (2.14).

Step G. Let $k = k + 1$ and restart from **Step B**.

Remark 2.5. *(Algorithm interpretation) The proposed Algorithm 2 can be seen in complementary theoretical lenses:*

- ***Connection to projected gradient methods:*** *The projected gradient method [22] solves constrained optimization problems via updates of the form $x_{k+1} = P_P[x_k - \alpha_k\nabla f(x_k)]$.* ***Step E*** *of the proposed Algorithm 2 adopts this same fundamental structure through $x_{k+1} = P_P[x_k - \psi\delta_k E(z_k)]$. The key distinction is that the operator E is evaluated at the auxiliary point z_k rather than at x_k.*
- ***Adaptive projected gradient interpretation:*** *The update $x_{k+1} = P_P[x_k - \psi\delta_k E(z_k)]$ follows the projected gradient paradigm, with the additional feature that the operator is evaluated at an extrapolated point z_k rather than at the current iterate.*

This viewpoint shows that the proposed Algorithm 2 combines the feasibility guarantees of projected gradient schemes while incorporating adaptive mechanisms through $\overline{\gamma}_{(k)}$, $\overline{\eta}_{(k)}$, and ϕ_k.

3. Analysis of convergence

The global convergence of the proposed Algorithm 2 is established in this section. We begin by stating some important assumptions and definitions.

Assumption 3.1. *The following assumptions are sufficient to guarantee that the sequence* $\{x_k\}$ *generated by Algorithm 2 converges to a solution* $\overline{x}$ *of problem* (1.1)*:*

(R1) The solution set of (1.1) *is nonempty and is denoted by* $P^{\times}$.
(R2) The function E is Lipschitz continuous; that is, there exists a positive constant $\theta > 0$ *such that*

$$\|E(x) - E(y)\| \leq \theta\|x - y\|, \quad \forall x, y \in \mathbb{R}^n. \tag{3.1}$$

(R3) The function E is monotone; that is,

$$(E(x) - E(y))^{\top}(x - y) \geq 0, \quad \forall x, y \in \mathbb{R}^n. \tag{3.2}$$

Suppose $P \subseteq \mathbb{R}^n$ is a nonempty set that is both closed and convex. For any point $x \in \mathbb{R}^n$, the orthogonal projection of x onto P, which is denoted as $P_P[x]$, is defined in the following manner:

$$P_P[x] := arg\min\{\|x - y\| : y \in P\}. \tag{3.3}$$

Let $x, y \in \mathbb{R}^n$. The mapping $P_P : \mathbb{R}^n \to P$ is called the projection operator and satisfies the non-expansive property:

$$\|P_P[x] - P_P[y]\| \leq \|x - y\|, \ \forall x, y \in \mathbb{R}^n.$$

Consequently, we have

$$\|P_P[x] - y\| \leq \|x - y\|, \quad \forall y \in P. \tag{3.4}$$

Remark 3.2. *Consider the following remark:*

(a) Using assumption (R3) and the definition of s_{k-1}, y_{k-1}, $\gamma > 0$, *we have*

$$s_{k-1}^{\top}y_{k-1} = s_{k-1}^{\top}(E(x) - E(y)) + \gamma s_{k-1}^{\top}s_{k-1} \geq \gamma\|s_{k-1}\|^2 > 0.$$

(b) Using assumption (R2) and the Cauchy-Schwarz inequality, we have

$$s_{k-1}^{\top}y_{k-1} \leq (\theta + \gamma)\|s_{k-1}\|^2.$$

Lemma 3.3. *Let* $\phi_0 > 0$. *Then, for* $n \geq 1$, $\phi_k > 0$ *and* $\phi_k^{-1} > 0$ *for all* $k \geq 1$.

Proof. (a) From the acceleration parameter defined in (2.14) and part (a) of Remark 3.2, we have

$$\begin{aligned}
\phi_k &= \phi_{k-1} + \frac{s_{k-1}^{\top}(y_{k-1} - \phi_{k-1}s_{k-1})}{ns_{k-1}^{\top}s_{k-1}} \\
&= \phi_{k-1} + \frac{s_{k-1}^{\top}y_{k-1}}{ns_{k-1}^{\top}s_{k-1}} - \frac{\phi_{k-1}s_{k-1}^{\top}s_{k-1}}{ns_{k-1}^{\top}s_{k-1}} \\
&= \phi_{k-1} - \frac{\phi_{k-1}}{n} + \frac{s_{k-1}^{\top}y_{k-1}}{ns_{k-1}^{\top}s_{k-1}}
\end{aligned}$$

$$
\begin{aligned}
&> \phi_{k-1} - \frac{\phi_{k-1}}{n} + \frac{\gamma}{n} \\
&= \phi_{k-1}\left(1 - \frac{1}{n}\right) + \frac{\gamma}{n} > 0.
\end{aligned}
$$

(b) From (2.15) and part (b) of Remark 3.2, we have

$$
\begin{aligned}
\phi_k^{-1} &= \frac{n\|s_{k-1}\|^2}{\phi_{k-1}(n-1)\|s_{k-1}\|^2 + s_{k-1}^\top y_{k-1}} \\
&\geq \frac{n\|s_{k-1}\|^2}{\phi_{k-1}(n-1)\|s_{k-1}\|^2 + (\theta+\gamma)\|s_{k-1}\|^2} \\
&= \frac{n}{\phi_{k-1}(n-1) + (\theta+\gamma)} > 0.
\end{aligned}
$$

Remark 3.4. *Under the conditions of Lemma 3.3, we have the uniform bound*

$$
\phi_k^{-1} \leq \frac{n}{\gamma} \quad \textit{for all } k \geq 1.
$$

From Lemma 3.3, we know $\phi_k > 0$. From Remark 3.2 and Eq (2.14)*,*

$$
\begin{aligned}
\phi_k &= \phi_{k-1}\left(1 - \frac{1}{n}\right) + \frac{s_{k-1}^\top y_{k-1}}{n s_{k-1}^\top s_{k-1}} \geq \frac{s_{k-1}^\top y_{k-1}}{n s_{k-1}^\top s_{k-1}} \quad \textit{(since } \phi_{k-1} > 0\textit{)} \\
&\geq \frac{\gamma}{n} \quad \textit{(from Remark 3.2).}
\end{aligned}
$$

Therefore, $\phi_k \geq \frac{\gamma}{n} > 0$, which implies $\phi_k^{-1} \leq \frac{n}{\gamma}$.

Lemma 3.5. *Assume that assumptions (R1)–(R3) hold and that the sequence $\{x_k\}$ is generated by Algorithm 2. Then, the search direction $\{d_k\}$ satisfies the following descent property:*

$$
E(x_k)^\top d_k \leq -N\|E(x_k)\|^2, \tag{3.5}
$$

where $N = \min\left\{1, \frac{(\overline{\gamma}_{(k)}+1)n}{\gamma}\right\}$.

Proof. For $k = 0$, from (2.17), we have $E(x_0)^\top d_0 = -\|E(x_0)\|^2$.

For $k \geq 1$, from (2.17) and the boundedness of ϕ_k in Lemma 3.3, we have the following:

$$
E(x_k)^\top d_k = -(\overline{\gamma}_{(k)} + 1)\phi_k^{-1}\|E(x_k)\|^2 \leq -\frac{(\overline{\gamma}_{(k)} + 1)n}{\gamma}\|E(x_k)\|^2. \tag{3.6}
$$

Therefore, (3.5) holds.

Lemma 3.6. *The search direction* (2.17) *generated by Algorithm 2 is bounded. That is,*

$$
\|d_k\| \leq M_\phi \|E(x_k)\|, \tag{3.7}
$$

where $M_\phi = \max\left\{1, \frac{(\overline{\gamma}_{(k)}+1)n}{\gamma}\right\}$.

Proof. For $k = 0$, using Lemma 3.3 and the definition of the search direction in (2.17), we have

$$\|d_0\| = \|E(x_0)\|.$$

For $k \geq 1$, observe from (2.17), and we have

$$\|d_k\| = \| - (\overline{\gamma}_{(k)} + 1)\phi_k^{-1} E(x_k)\| \leq (\overline{\gamma}_{(k)} + 1) \left|\phi_k^{-1}\right| \|E(x_k)\| \leq (\overline{\gamma}_{(k)} + 1)\frac{n}{\gamma}\|E(x_k)\|. \tag{3.8}$$

Hence, (3.7) holds.

Lemma 3.7. *Assume that the sequences* $\{x_k\}$ *and* $\{z_k\}$ *are generated by Algorithm 2 and that assumptions (R1)–(R3) hold. Then, there exists* $\overline{\alpha}_{(k)}$ *such that the following statements are true:*
(i) *For any* $k \geq 0$, *there exists* $\overline{\beta}_{(k)} = \zeta\varpi^{j_k}$ *and* $\overline{\eta}_{(k)} = \xi\rho^{j_k}$ *satisfying* (2.16) *for some* $j \in \mathbb{N} \cup \{0\}$.
(ii) *The step-length* $\overline{\alpha}_{(k)}$ *satisfies*

$$\overline{\alpha}_{(k)} > \overline{\alpha} := \min\left\{\xi, \frac{\varrho N}{(\theta + \sigma)((\overline{\gamma}_{(k)} + 1)\frac{n}{\gamma})^2}\right\}. \tag{3.9}$$

Proof. This is proved in two cases as follows:
Case (i). Suppose there exists $k_0 \geq 0$ such that (2.16) does not hold. Then, for every nonnegative integer j, we have

$$-E(x_{k_0} + (\zeta\varpi^{j_{k_0}} + \xi\rho^{j_{k_0}}\phi_k)d_{k_0})^\top d_{k_0} < \sigma(\zeta\varpi^{j_{k_0}} + \xi\rho^{j_{k_0}}\phi_k)\|d_{k_0}\|^2.$$

Using assumption (R3) and setting $j \to \infty$, we have

$$-E(x_{k_0})^\top d_{k_0} \leq 0. \tag{3.10}$$

Conversely, from (3.5), we have

$$-E(x_{k_0})^\top d_{k_0} \geq N\|E(x_{k_0})\|^2 > 0,$$

which contradicts (3.10). Therefore, the step-length is well defined.
Case (ii). Suppose $\overline{\alpha}_{(k)} \neq \xi$, from $\overline{\alpha}_{(k)} = (\overline{\beta}_{(k)} + \overline{\eta}_{(k)}\phi_k)$, then, $\overline{\alpha}_{(k)}\varrho^{-1}$ does not satisfy (2.16); thus,

$$-E(x_k + \overline{\alpha}_{(k)}d_k)^\top d_k < \sigma\overline{\alpha}_{(k)}\|d_k\|^2.$$

Using (3.5) and assumption (R2), we have

$$\begin{aligned}
N\|E(x_k)\|^2 &\leq -E(x_k)^\top d_k \\
&= (E(x_k + \overline{\alpha}_{(k)}d_k) - E(x_k)^\top d_k - E(x_k + \overline{\alpha}_{(k)}d_k)^\top d_k \\
&< \theta\overline{\alpha}_{(k)}\|d_k\|^2 + \sigma\overline{\alpha}_{(k)}\|d_k\|^2 \\
&= (\theta + \sigma)\overline{\alpha}_{(k)}\varrho^{-1}\|d_k\|^2.
\end{aligned}$$

The above inequality can be rewritten as follows:

$$\overline{\alpha}_{(k)} > \frac{\varrho N\|E(x_k)\|^2}{(\theta + \sigma)\|d_k\|^2} \geq \frac{\varrho N}{(\theta + \sigma)((\overline{\gamma}_{(k)} + 1)\frac{n}{\gamma})^2}, \tag{3.11}$$

where the last inequality follows from (3.7). Therefore, (3.9) holds, and the proof is complete.

Lemma 3.8. *Assume that assumptions (R1)–(R3) hold, and let the sequences $\{x_k\}$ and $\{z_k\}$ be generated by Algorithm 2. Then, both $\{x_k\}$ and $\{z_k\}$ are bounded. Additionally,*

$$\lim_{k\to\infty} \|x_k - z_k\| = 0, \tag{3.12}$$

and

$$\lim_{k\to\infty} \|x_{k+1} - x_k\| = 0. \tag{3.13}$$

Proof. Let $\overline{x} \in P^{\times}$ be an arbitrary solution to (1.1). Since E is monotone, it follows that

$$E(z_k)^\top (x_k - \overline{x}) \geq E(z_k)^\top (x_k - z_k). \tag{3.14}$$

Using the line-search condition (2.16) and the definition of z_k, we obtain

$$E(z_k)^\top (x_k - \overline{x}) \geq E(z_k)^\top (x_k - z_k) \geq \sigma \overline{\alpha}_{(k)}{}^2 \|d_k\|^2 > 0. \tag{3.15}$$

Using the inexpensive projection formulas (3.4) and (2.18), together with inequality (3.15), we have

$$\begin{aligned}
\|x_{k+1} - \overline{x}\|^2 &= \|P_P[x_k - \psi\delta_k E(z_k)] - \overline{x}\|^2 \\
&\leq \|x_k - \psi\delta_k E(z_k) - \overline{x}\|^2 \\
&= \|x_k - \overline{x}\|^2 - 2\psi\delta_k E(z_k)^\top (x_k - \overline{x}) + \psi^2\delta_k^2 \|E(z_k)\|^2 \\
&= \|x_k - \overline{x}\|^2 - 2\psi \frac{E(z_k)^\top (x_k - z_k)}{\|E(z_k)\|^2} E(z_k)^\top (x_k - \overline{x}) + \psi^2 \left(\frac{E(z_k)^\top (x_k - z_k)}{\|E(z_k)\|} \right)^2 \\
&\leq \|x_k - \overline{x}\|^2 - 2\psi \frac{E(z_k)^\top (x_k - z_k)}{\|E(z_k)\|^2} E(z_k)^\top (x_k - z_k) + \psi^2 \left(\frac{E(z_k)^\top (x_k - z_k)}{\|E(z_k)\|} \right)^2 \\
&= \|x_k - \overline{x}\|^2 - \widehat{\sigma} \left(\frac{(E(z_k)^\top (x_k - z_k)}{\|E(z_k)\|} \right)^2 &(3.16) \\
&\leq \|x_k - \overline{x}\|^2 - \widehat{\sigma} \|x_k - z_k\|^4, &(3.17)
\end{aligned}$$

where $\widehat{\sigma} = \psi(2-\psi)\sigma^2$, and $0 < \psi < 2$. Thus, from inequality (3.16), we have

$$\|x_{k+1} - \overline{x}\|^2 \leq \|x_k - \overline{x}\|^2. \tag{3.18}$$

Thus, $\|x_k - \overline{x}\|$ is decreasing for all k. Therefore, the sequence $\{x_k\}$ is bounded; that is,

$$\|x_k\| \leq \widehat{\eta}, \quad \forall k \geq 0. \tag{3.19}$$

Additionally, using (3.18) and assumption (R1), we have

$$\|x_{k+1} - \overline{x}\|^2 \leq \|x_k - \overline{x}\|^2 \leq \|x_{k-1} - \overline{x}\|^2 \leq \ldots \|x_0 - \overline{x}\|^2.$$

Again, using assumptions (R1) and (R2), we obtain

$$\|E(x_k)\| = \|E(x_k) - E(\overline{x})\| \leq \theta \|x_k - \overline{x}\| \leq \theta \|x_0 - \overline{x}\|. \tag{3.20}$$

Set $\vartheta = \theta\|x_0 - \overline{x}\|$. Then, from (3.20), it is easy to see that the sequence $\{E(x_k)\}$ is bounded; that is,

$$\|E(x_k)\| \leq \vartheta, \quad \forall k \geq 0. \tag{3.21}$$

Moreover, using assumption (R1), we have the following:

$$E(x_k)^\top (x_k - z_k) - E(z_k)^\top (x_k - z_k) \geq 0.$$

Therefore, this implies that

$$E(z_k)^\top (x_k - z_k) \leq E(x_k)^\top (x_k - z_k). \tag{3.22}$$

From the definition of $\{z_k\}$, (3.15), (3.22), and the Cauchy-Schwarz inequality, it follows that

$$\begin{aligned} \sigma\|x_k - z_k\| =& \frac{\sigma\|x_k - z_k\|^2}{\|x_k - z_k\|} = \frac{\sigma\|\overline{\alpha}_{(k)} d_k\|^2}{\|x_k - z_k\|} \\ \leq& \frac{E(z_k)^\top (x_k - z_k)}{\|x_k - z_k\|} \leq \frac{E(x_k)^\top (x_k - z_k)}{\|x_k - z_k\|} \leq \|E(x_k)\|. \end{aligned} \tag{3.23}$$

Also, from (3.23) and the reverse of the inequality above, we obtain

$$\sigma(\|z_k\| - \|x_k\|) \leq \sigma\|z_k - x_k\| \leq \|E(x_k)\|.$$

Thus, from the above inequality together with (3.19) and (3.21), we have

$$\|z_k\| \leq \frac{1}{\sigma}\|E(x_k)\| + \|x_k\| \leq \frac{1}{\sigma}\vartheta + \widehat{\eta}.$$

Therefore, the sequence $\{z_k\}$ is bounded. Moreover, for any $\overline{x} \in P^\times$, the sequence $\{z_k - \overline{x}\}$ is bounded; that is, there exists $\widehat{L} > 0$ such that

$$\|z_k - \overline{x}\| \leq \widehat{L}, \quad \forall k \geq 0.$$

Using the above inequality together with assumption (R2), we obtain

$$\|E(z_k)\| = \|E(z_k) - E(\overline{x})\| \leq \theta\|z_k - \overline{x}\| \leq \theta\widehat{L}. \tag{3.24}$$

Therefore, using the inequality (3.17), we have

$$\psi(2 - \psi)\frac{\sigma^2}{(\theta\widehat{L})^2}\|x_k - z_k\|^4 \leq \|x_k - \overline{x}\|^2 - \|x_{k+1} - \overline{x}\|^2.$$

This implies

$$\psi(2 - \psi)\frac{\sigma^2}{(\theta\widehat{L})^2}\sum_{k=0}^{\infty}\|x_k - z_k\|^4 \leq \sum_{k=0}^{\infty}(\|x_k - \overline{x}\|^2 - \|x_{k+1} - \overline{x}\|^2) = \|x_0 - \overline{x}\|^2 < \infty. \tag{3.25}$$

Hence, inequality (3.25) implies that

$$\lim_{k\to\infty}\|x_k - z_k\| = 0. \tag{3.26}$$

Thus, (3.12) holds. Additionally, using $P_P[\cdot]$ in (3.4), we obtain

$$\begin{aligned} \|x_{k+1} - x_k\| &= \|P_P[x_k - \psi\delta_k E(z_k)] - x_k\| \\ &= \|x_k - \psi\delta_k E(z_k) - x_k\| \\ &= \|\psi\delta_k E(z_k)\| \le \psi\|x_k - z_k\|, \quad \forall k \ge 0. \end{aligned}$$

Hence, the above inequality implies that (3.13) holds. Therefore, the proof is complete.

Theorem 3.9. *Assume that assumptions (R1)–(R3) hold, and let the sequence* $\{x_k\}$ *be generated by Algorithm 2. Then,* $\{x_k\}$ *converges to a solution of* (1.1).

Proof. First, from (3.9) and (3.26), we have $0 \le \overline{\alpha}\|d_k\| \le \overline{\alpha}_{(k)}\|d_k\| \to 0$. Hence, $\lim_{k\to\infty} \|d_k\| = 0$. Together with (3.5), this yields

$$0 \le N\|E(x_k)\| \le \|d_k\| \to 0,$$

which implies $\lim_{k\to\infty} \|E(x_k)\| = 0$. Moreover, from (3.26) and the boundedness of $\{x_k\}$, the sequence $\{x_k\}$ has at least one accumulation point. Let $\overline{x}$ be an accumulation point of $\{x_k\}$, and let $\mathcal{S} \subset \{0, 1, 2, \ldots\}$ be an infinite index set such that

$$\lim_{k\to\infty,\, k\in\mathcal{S}} x_k = \overline{x} \in P.$$

Since P is closed using assumptions (R1)–(R3), it implies

$$0 = \lim_{k\to\infty} \|E(x_k)\| = \lim_{k\to\infty,\ k\in\mathcal{S}} \|E(x_k)\| = \|E(\overline{x})\|.$$

This shows $\overline{x}$ is a solution of (1.1). Also, from Lemma 3.8, we know the sequence $\{\|x_k - \overline{x}\|\}$ is convergent. Therefore, by setting $\overline{x} := \widehat{x}$, we have

$$\lim_{k\to\infty} \|x_k - \widehat{x}\| = \lim_{k\to\infty,\ k\in\mathcal{S}} \|x_k - \widehat{x}\| = 0.$$

Hence, the sequence $\{x_k\}$ converges to $\overline{x} \in P^{\times}$.

3.1. Convergence rate

To establish the Q-linear convergence rate of the sequence, we require the following additional assumption on the operator E over the solution set.

Assumption 3.10. *For any* $\bar{x} \in P^{\times}$*, there exist positive constants* p *and* Ω *such that the following holds:*

$$p.\,\text{dist}(x, P^{\times}) \le \|E(x)\|, \quad \forall x \in N(\overline{x}, \Omega), \tag{3.27}$$

where $\text{dist}(x, P^{\times})$ *is the distance from* x *to the solution set* $P^{\times}$*, and* $N(\overline{x}, \Omega) = \{x \in \mathbb{R}^n : \|x - \overline{x}\| \le \Omega\}$.

Theorem 3.11. *Assume that Assumptions 3.10 and (R3) hold. Let the sequences* $\{x_k\}$ *and* $\{z_k\}$ *be generated by Algorithm 2. Then, the sequence* $\{\text{dist}(x_k, P^{\times})\}$ *converges Q-linearly to 0.*

Proof. Let $y_k := \text{argmin}\{\|x_k - \overline{y}\| : \overline{y} \in P^{\times}\}$. This means that y_k is the nearest solution to x_k,

$$\|x_k - y_k\| = \text{dist}(x_k, P^{\times}), \quad \forall k \ge 0. \tag{3.28}$$

Using (3.15) and substituting $\overline{y}$ for $\overline{x}$ in (3.16), we have

$$\|x_{k+1} - \overline{y}\|^2 = \|x_k - \overline{y}\|^2 - \frac{\widehat{\sigma}\sigma^2\overline{\alpha}^4_{(k)}\|d_k\|^4}{\|E(z_k)\|^2}. \tag{3.29}$$

By definition of y_k, and using assumption (R2) and Eq (3.7), we have

$$\begin{aligned}\|E(z_k)\| &= \|E(z_k - E(y_k)\| \leq \theta\|z_k - y_k\| = \theta(\|z_k - x_k\| + \|x_k - y_k\|)\\ &= \theta(\overline{\alpha}_{(k)}\|d_k\| + \|x_k - y_k\|) \leq \theta(\|d_k\| + \|x_k - y_k\|)\\ &\leq \theta(M_\phi\|E(x_k)\| - \|E(y_k)\| + \|x_k - y_k\|) \leq \theta(\theta M_\phi + 1)\|x_k - y_k\|\\ &= \theta(\theta M_\phi + 1)dist(x_k, P^\times).\end{aligned}$$

Applying the Cauchy-Schwarz in (3.5), we obtain

$$\|d_k\| \geq N\|E(x_k)\|. \tag{3.30}$$

From (3.28) and the above inequalities, we have

$$\begin{aligned}\text{dist}(h_{k+1}, P^\times) \leq \|x_{k+1} - y_k\|^2 &\leq \text{dist}(x_k, P^\times)^2 - \frac{\widehat{\sigma}\sigma^2\overline{\alpha}^4_k\|d_k\|^4}{\|E(z_k)\|^2}\\ &\leq \text{dist}(x_k, P^\times)^2 - \frac{\widehat{\sigma}\sigma^2\overline{\alpha}^4 N^4\|E(x_k)\|^4}{\|E(z_k)\|^2}\\ &\leq \text{dist}(x_k, P^\times)^2 - \frac{\widehat{\sigma}\sigma^2\overline{\alpha}^4 N^4 p^2\,\text{dist}(x_k, P^\times)^4}{\theta^2(\theta M_\phi + 1)^2\,\text{dist}(x_k, P^\times)^2}\\ &= \left(1 - \frac{\widehat{\sigma}\sigma^2\overline{\alpha}^4 N^4 p^2}{\theta^2(\theta M_\phi + 1)^2}\right)\text{dist}(x_k, P^\times)^2.\end{aligned}$$

Therefore, since the positive scalars $p, N, \sigma, \widehat{\sigma}$, and $\overline{\alpha}$ are all in $(0, 1)$ with $\theta > 1$, it shows that $\{\text{dist}(x_k, P^\times)\}$ converges Q-linearly to 0.

4. Numerical simulation

This section presents numerical results to assess the efficiency of the proposed Algorithm 2. We conduct numerical experiments on large-scale nonlinear problems. Throughout the simulations, we compare Algorithm 2 with recent methods. The MATLAB code was developed in version 8.3.0 (R2023b) and executed on a system equipped with an Intel(R) Core(TM) i7-10750H CPU, 16GB of RAM, and 1TB of storage, running Windows 10.

Parameter selection for the proposed algorithms: The proposed Algorithm 2 involves several parameters, selected according to the following rationale:

- **Relaxation parameter**: The value was set to $\lambda = 0.4$, which gave the best performance for the proposed Algorithm 2.
- **Line-search parameters**: These values were chosen to maximize the satisfaction rate of the adapted line-search conditions across the test set. The line-search parameters for the proposed algorithm were set as $\xi = 1$, $\zeta = 1$, $\rho = 0.5$, $\omega = 0.39$, and $\sigma = 0.0001$.

- **Tolerance**: A common tolerance, `tol` $= 10^{-5}$, was used for all solvers (Algorithm 2, Hybrid double step length method (HADS), modification of a conjugate gradient approach for convex constraints with applications (MCHCG), and Algorithm 2.1) to ensure practical convergence while maintaining reasonable computation time.
- **Acceleration parameter**: The acceleration parameter in Algorithm 2, $(\gamma_{(k)}+1) = 1.2$, was derived from the theoretical framework to maintain uniform convergence throughout the numerical experiments.

Parameter selection for comparison algorithms: For a fair comparison, the parameters of each algorithm were selected individually using a validation set of problems from the original references. This is standard practice to avoid bias. The parameter settings below are those used in the experiments:

- HADS method [16]: The parameters were set as $\sigma = 10^{-4}$, $\rho = 0.90$, $\xi = 1$, and $t = 1.2$, consistent with the recommendations in the original paper.
- MCHCG method [26]: The parameters were set as $\tau = 10$, $\rho = 0.90$, $\xi = 1$, $\sigma = 10^{-4}$, and $t = 1.2$, which provided a good balance between convergence rate and robustness across the test problems.
- Algorithm 2.1 method [27]: The parameters were set as $v = 1.2$, $\xi = 1$, $\rho = 0.55$, and $\sigma = 10^{-6}$ after extensive testing.

The numerical simulations were conducted on five benchmark problems, presented below. Each problem was tested in one dimension, with $n = 100{,}000$ (denoted by #Dim) and with five different initial points (denoted by #Int), listed below. The iteration was terminated when any of the following conditions was satisfied: (i) $\|E(x_k)\| \leq 10^{-6}$, (ii) the iteration count exceeded 1000, or (iii) the symbol $*$ indicated that a method failed to reach a solution.

Throughout the experiment, #Itr denotes the number of iterations, #Fev denotes the number of function evaluations, #Tm denotes CPU time in seconds (s), and #Nrm denotes the residual norm at termination. The initial points used in the numerical experiments are: $x_1 = (10, 10, \cdots, 10)^\top$, $x_2 = (0.1, 0.1, \cdots, 0.1)^\top$, $x_3 = \left(0, \frac{1}{2}, \frac{2}{3}, \ldots, 1 - \frac{1}{n}\right)^\top$, $x_4 = \left(1 - \frac{1}{n},\ 2 - \frac{2}{n},\ 3 - \frac{3}{n},\ \ldots,\ n - 1\right)^\top$, and $x_5 = (0.5, 0.5, \cdots, 0.5)^\top$.

The following is a list of the five benchmark problems used in the numerical experiments:

Problem 1: [28]

$$E_i(x) = 2x_i - \sin(x_i), \quad i = 1, 2, 3, \ldots, n, \tag{4.1}$$

where $P = \mathbb{R}^n_+$.

Problem 2: [29]

$$\begin{aligned} E_1(x) &= x_1 - e^{\frac{\cos(x_1+x_2)}{n+1}}, \\ E_i(x) &= x_i - e^{\frac{\cos(x_{i-1}+x_i+x_{i+1})}{n+1}}, \quad i = 2, 3, \ldots, n-1, \\ E_n(x) &= x_n - e^{\frac{\cos(x_{n-1}+x_n)}{n+1}}, \end{aligned} \tag{4.2}$$

where $P = \mathbb{R}^n_+$.

Problem 3: [28]

$$E_1(x) = 2x_1 + \sin(x_1) - 1, \tag{4.3}$$

$$E_i(x) = 2x_{i-1} + 2x_i + 2\sin(x_i) - 1, \quad i = 2, 3, \ldots, n-1,$$
$$E_n(x) = 2x_n + \sin(x_n) - 1,$$

where $P = \mathbb{R}^n_+$.
Problem 4 : [28]

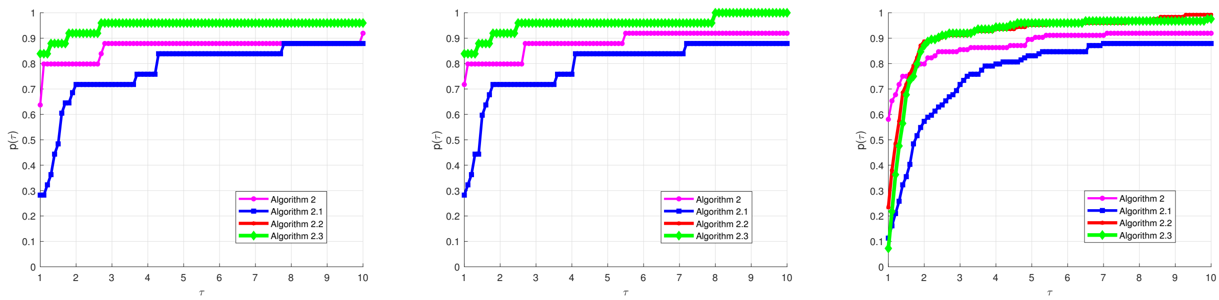
$$E_1(x) = 2x_1 + e^{\sin(x_1)} - 1, \tag{4.4}$$
$$E_i(x) = 2x_{i-1} + e^{\sin(x_i)} + 2x_i - 1, \quad i = 2, 3, \ldots, n-1,$$
$$E_n(x) = 2x_n + e^{\sin(x_n)} - 1,$$

where $P = \mathbb{R}^n_+$.
Problem 5: [28]

$$E_1(x) = 2.5x_1 - e^{\frac{\cos(x_1+x_2)}{n+1}}, \tag{4.5}$$
$$E_i(x) = 2.5x_i - e^{\frac{\cos(x_{i-1}+x_i+x_{i+1})}{n+1}}, \quad i = 2, 3, \ldots, n-1,$$
$$E_n(x) = 2.5x_n - e^{\frac{\cos(x_{n-1}+x_n)}{n+1}},$$

where $P = \mathbb{R}^n_+$.

Figure 1 presents a new set of numerical experiments designed to evaluate the performance and robustness of the proposed Algorithm 2 under different choices of the correction parameter. In particular, we vary $(\gamma_{(k)}+1)$ over the values 1, 1.5, 2.0, and 2.5. For convenience, these four variants are denoted by Algorithm 2, Algorithm 2.1, Algorithm 2.2, and Algorithm 2.3, respectively; this naming convention is used consistently throughout the experimental section.



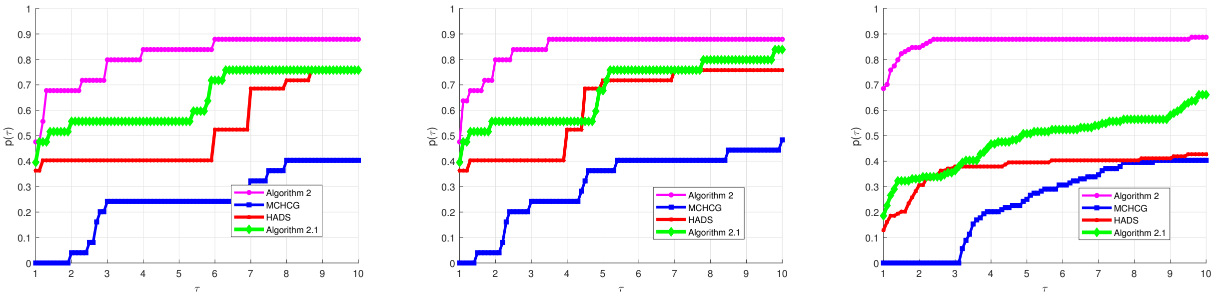
(a) Performance profile of #Itr (b) Performance profile of #Fev (c) Performance profile of #Tm

Figure 1. Performance profiles comparing Algorithm 2, Algorithm 2.1, Algorithm 2.2, and Algorithm 2.3 methods with respect to #Itr, #Fev, and #Tm.

From Figure 1, we observe a clear performance hierarchy among the four algorithms. The curve corresponding to Algorithm 2.3 (i.e., $(\gamma_{(k)}+1) = 2.5$) dominates the others, lying above the curves of Algorithm 2, Algorithm 2.1, and Algorithm 2.2 over the performance-profile range. In this context, a higher curve typically indicates that the method is more frequently among the best (or near-best) solvers across the entire test set. Therefore, this visual evidence strongly suggests that Algorithm 2.3 is the most effective configuration of Algorithm 2 within the tested parameter range.

This monotonic improvement in performance as $(\gamma_{(k)} + 1)$ increases from 1 to 2.5 indicates that the proposed algorithm benefits from larger values of the correction parameter. In other words, increasing $(\gamma_{(k)} + 1)$ enhances the algorithm's ability to adjust its iterates, which in turn leads to better numerical behavior (e.g., faster convergence or higher robustness) across the benchmark problems. Full numerical evidence, including detailed tables of iteration counts, error metrics, and other performance indicators for each test problem and algorithmic variant, is available at `https://github.com/Mash4u/Algorithm/blob/main/Template_New.pdf`. On that basis, we conducted an additional numerical experiment using other existing methods to further evaluate the efficiency of the proposed approach. We set the acceleration parameter of the proposed Algorithm 2 to $(\gamma_{(k)} + 1) = 2.5$, while the parameters for the comparison algorithms (MCHCG, HADS, and Algorithm 2.1) were adopted as reported in their original references.

Figure 2 presents the Dolan-Moré performance profiles [30] for Algorithm 2, HADS, MCHCG, and Algorithm 2.1 with respect to #Itr, #Tm, and #Fev. The horizontal axis represents the performance ratio, while the vertical axis gives the proportion of problems solved within a given factor of the best solver. In Figure 2(a), the curve of Algorithm 2 lies above those of the competing methods, indicating better performance in terms of iterations. In Figure 2(b), Algorithm 2 also performs best in #Fev. In Figure 2(c), the #Tm profile supports the same conclusion, while the competing methods require longer solution times.



(a) Performance profile of #Itr (b) Performance profile of #Fev (c) Performance profile of #Tm

Figure 2. Performance profiles comparing Algorithm 2, MCHCG, HADS, and Algorithm 2.1 methods with respect to #Itr, #Fev, and #Tm.

Tables 1–5 report the performance of the proposed Algorithm 2 against HADS, MCHCG, and Algorithm 2.1 on Problems 1–5. The methods are evaluated using the number of iterations (#Itr), CPU time in seconds (#Tm), number of function evaluations (#Fev), and residual norm (#Nrm). Compared with HADS, MCHCG, and Algorithm 2.1, the proposed Algorithm 2 consistently converges with fewer #Itr, lower #Tm, fewer #Fev, and competitive #Nrm values across the tested problems, as can be observed in Table 6. In addition, Algorithm 2 solves the problems with the fewest iterations in most cases. In Tables 3 and 5, the MCHCG method fails to reach a solution for all initial points and for the initial point x_2, respectively. These results indicate that the acceleration-parameter strategy reduces computational burden. Overall, for large-scale nonlinear equations with convex constraints, Tables 1–5 show that the hybrid accelerated double step-length method is faster and more reliable. Its superior performance is mainly reflected in #Itr, #Tm, and #Fev.

Table 1. Numerical results for Algorithm 2, MCHCG, HADS, and Algorithm 2.1 on Problem 1.

		Algorithm 2				MCHCG				HADS				Algorithm 2.1			
#Dim	Ip	#Itr	#Fev	#Tm	#Nrm	#Itr	#Fev	#Tm	#Nrm	#Itr	#Fev	#Tm	#Nrm	#Itr	#Fev	#Tm	#Nrm
	x1	1	2	0.026059	0	20	22	0.593444	6.75E-08	8	10	0.443685	1.08E-07	1	2	0.037583	0
	x2	1	2	0.022531	0	18	20	0.479453	4.22E-08	6	8	0.212611	7.62E-07	1	2	0.022651	0
100,000	x3	1	2	0.024475	0	19	21	0.520565	7.47E-08	7	9	0.239397	6.19E-07	1	2	0.020062	0
	x4	3	4	0.056823	0	32	34	0.9085	8.6E-08	26	28	1.186826	1.36E-07	38	39	3.146224	0
	x5	1	2	0.019535	0	20	22	0.569993	6.41E-08	6	8	0.244628	2.04E-07	1	2	0.039149	0
	x1	1	2	0.031501	0	20	22	0.490712	6.75E-08	8	10	0.309904	1.08E-07	1	2	0.021103	0
	x2	1	2	0.017395	0	18	20	0.447016	4.22E-08	6	8	0.215026	7.62E-07	1	2	0.01502	0
100,000	x3	1	2	0.016224	0	19	21	0.50165	7.47E-08	7	9	0.2635	6.19E-07	1	2	0.021401	0
	x4	3	4	0.074759	0	32	34	0.928331	8.6E-08	26	28	1.29115	1.36E-07	38	39	3.208644	0
	x5	1	2	0.03826	0	20	22	0.556252	6.41E-08	6	8	0.246522	2.04E-07	1	2	0.027099	0
	x1	1	2	0.02089	0	20	22	0.523223	6.75E-08	8	10	0.319815	1.08E-07	1	2	0.022375	0
	x2	1	2	0.017633	0	18	20	0.504305	4.22E-08	6	8	0.242592	7.62E-07	1	2	0.019659	0
100,000	x3	1	2	0.02394	0	19	21	0.488021	7.47E-08	7	9	0.268033	6.19E-07	1	2	0.017501	0
	x4	3	4	0.055517	0	32	34	0.878871	8.6E-08	26	28	1.219049	1.36E-07	38	39	3.260046	0
	x5	1	2	0.024338	0	20	22	0.560211	6.41E-08	6	8	0.256787	2.04E-07	1	2	0.025783	0
	x1	1	2	0.023248	0	20	22	0.528454	6.75E-08	8	10	0.257758	1.08E-07	1	2	0.027571	0
	x2	1	2	0.019258	0	18	20	0.458894	4.22E-08	6	8	0.227821	7.62E-07	1	2	0.017343	0
100,000	x3	1	2	0.020194	0	19	21	0.48971	7.47E-08	7	9	0.254669	6.19E-07	1	2	0.018478	0
	x4	3	4	0.062997	0	32	34	0.956109	8.6E-08	26	28	1.275297	1.36E-07	38	39	3.331666	0
	x5	1	2	0.020572	0	20	22	0.505226	6.41E-08	6	8	0.25827	2.04E-07	1	2	0.025276	0
	x1	1	2	0.023098	0	20	22	0.526251	6.75E-08	8	10	0.294523	1.08E-07	1	2	0.030912	0
	x2	1	2	0.019119	0	18	20	0.425053	4.22E-08	6	8	0.224483	7.62E-07	1	2	0.021497	0
100,000	x3	1	2	0.020399	0	19	21	0.498818	7.47E-08	7	9	0.250328	6.19E-07	1	2	0.019659	0
	x4	3	4	0.060716	0	32	34	0.878864	8.6E-08	26	28	1.214125	1.36E-07	38	39	3.202562	0
	x5	1	2	0.019428	0	20	22	0.509236	6.41E-08	6	8	0.225396	2.04E-07	1	2	0.033072	0

Table 2. Numerical results for Algorithm 2, MCHCG, HADS, and Algorithm 2.1 on Problem 2.

		Algorithm 2				MCHCG				HADS				Algorithm 2.1			
#Dim	Ip	#Itr	#Fev	#Tm	#Nrm	#Itr	#Fev	#Tm	#Nrm	#Itr	#Fev	#Tm	#Nrm	#Itr	#Fev	#Tm	#Nrm
	x1	10	11	0.32392	4.61E-07	22	24	1.118458	4.04E-08	8	10	0.535756	3.79E-07	50	52	3.143951	8.17E-08
	x2	10	11	0.325032	2.63E-07	21	23	1.069738	4.3E-08	8	10	0.566182	1.36E-07	47	49	3.042961	9.35E-08
100,000	x3	9	11	0.297769	6.61E-07	20	22	1.027744	8.34E-08	8	10	0.556296	8.94E-08	46	48	2.902696	9.02E-08
	x4	10	11	0.340852	7.33E-07	30	32	1.546336	5.49E-08	12	14	0.86137	1.35E-07	54	56	3.491832	8.15E-08
	x5	9	11	0.331905	8.78E-07	21	23	1.044077	3.75E-08	8	10	0.705785	1.19E-07	47	49	3.006308	8.15E-08
	x1	10	11	0.332895	4.61E-07	22	24	1.110226	4.04E-08	8	10	0.710421	3.79E-07	50	52	3.251118	8.17E-08
	x2	10	11	0.340621	2.63E-07	21	23	1.114849	4.3E-08	8	10	0.807567	1.36E-07	47	49	3.062909	9.35E-08
100,000	x3	9	11	0.330852	6.61E-07	20	22	1.04469	8.34E-08	8	10	0.703861	8.94E-08	46	48	2.952873	9.02E-08
	x4	10	11	0.323222	7.33E-07	30	32	1.560635	5.49E-08	12	14	0.966109	1.35E-07	54	56	3.422761	8.15E-08
	x5	9	11	0.289489	8.78E-07	21	23	1.09277	3.75E-08	8	10	1.301972	1.19E-07	47	49	2.992416	8.15E-08
	x1	10	11	0.307966	4.61E-07	22	24	1.121234	4.04E-08	8	10	1.340921	3.79E-07	50	52	3.240607	8.17E-08
	x2	10	11	0.316494	2.63E-07	21	23	1.12681	4.3E-08	8	10	1.796872	1.36E-07	47	49	3.459892	9.35E-08
100,000	x3	9	11	0.320428	6.61E-07	20	22	1.008039	8.34E-08	8	10	0.626108	8.94E-08	46	48	3.625547	9.02E-08
	x4	10	11	0.340417	7.33E-07	30	32	1.458114	5.49E-08	12	14	0.929377	1.35E-07	54	56	4.064186	8.15E-08
	x5	9	11	0.326843	8.78E-07	21	23	1.05954	3.75E-08	8	10	0.618513	1.19E-07	47	49	3.47057	8.15E-08
	x1	10	11	0.359652	4.61E-07	22	24	1.115213	4.04E-08	8	10	0.643201	3.79E-07	50	52	3.601564	8.17E-08
	x2	10	11	0.353203	2.63E-07	21	23	1.128678	4.3E-08	8	10	0.614643	1.36E-07	47	49	3.284611	9.35E-08
100,000	x3	9	11	0.311371	6.61E-07	20	22	1.067161	8.34E-08	8	10	0.577035	8.94E-08	46	48	3.564534	9.02E-08
	x4	10	11	0.34334	7.33E-07	30	32	1.504973	5.49E-08	12	14	0.85638	1.35E-07	54	56	5.154073	8.15E-08
	x5	9	11	0.31033	8.78E-07	21	23	1.168047	3.75E-08	8	10	0.608193	1.19E-07	47	49	4.585584	8.15E-08
	x1	10	11	0.349469	4.61E-07	22	24	1.342728	4.04E-08	8	10	0.592294	3.79E-07	50	52	3.293801	8.17E-08
	x2	10	11	0.354295	2.63E-07	21	23	1.215008	4.3E-08	8	10	0.581178	1.36E-07	47	49	3.624695	9.35E-08
100,000	x3	9	11	0.338326	6.61E-07	20	22	1.097725	8.34E-08	8	10	0.55927	8.94E-08	46	48	3.082743	9.02E-08
	x4	10	11	0.331291	7.33E-07	30	32	1.671045	5.49E-08	12	14	0.839577	1.35E-07	54	56	3.664448	8.15E-08
	x5	9	11	0.331977	8.78E-07	21	23	1.104	3.75E-08	8	10	0.633201	1.19E-07	47	49	3.851736	8.15E-08

Table 3. Numerical results for Algorithm 2, MCHCG, HADS, and Algorithm 2.1 on Problem 3.

		Algorithm 2				MCHCG				HADS				Algorithm 2.1			
#Dim	Ip	#Itr	#Fev	#Tm	#Nrm	#Itr	#Fev	#Tm	#Nrm	#Itr	#Fev	#Tm	#Nrm	#Itr	#Fev	#Tm	#Nrm
	x1	34	36	1.036189	5.7E-07	*	*	*	*	419	421	47.11037	9.98E-07	36	37	4.262354	0
	x2	19	20	0.614843	9.57E-07	*	*	*	*	336	338	37.4555	6.42E-07	37	38	4.190549	0
100,000	x3	27	29	0.833469	5.23E-07	*	*	*	*	412	414	45.77256	7.41E-07	36	37	4.067183	0
	x4	44	46	1.370637	6.09E-07	*	*	*	*	1001	1002	111.0306	0.000458	36	37	4.076761	0
	x5	34	35	1.115224	9.03E-07	*	*	*	*	517	519	57.56489	9.89E-07	36	37	4.08598	0
	x1	34	36	1.283811	5.7E-07	*	*	*	*	419	421	43.0888	9.98E-07	36	37	4.045079	0
	x2	19	20	0.751136	9.57E-07	*	*	*	*	336	338	34.53872	6.42E-07	37	38	4.253006	0
100,000	x3	27	29	0.827756	5.23E-07	*	*	*	*	412	414	42.17481	7.41E-07	36	37	4.233091	0
	x4	44	46	1.361429	6.09E-07	*	*	*	*	1001	1002	101.9039	0.000458	36	37	4.466225	0
	x5	34	35	1.041675	9.03E-07	*	*	*	*	517	519	52.95476	9.89E-07	36	37	4.117398	0
	x1	34	36	1.072608	5.7E-07	*	*	*	*	419	421	43.42637	9.98E-07	36	37	4.018931	0
	x2	19	20	0.584521	9.57E-07	*	*	*	*	336	338	34.555	6.42E-07	37	38	4.262676	0
100,000	x3	27	29	0.891984	5.23E-07	*	*	*	*	412	414	45.7106	7.41E-07	36	37	4.193461	0
	x4	44	46	1.379018	6.09E-07	*	*	*	*	1001	1002	81.84482	0.000458	36	37	4.175533	0
	x5	34	35	1.077897	9.03E-07	*	*	*	*	517	519	40.00884	9.89E-07	36	37	4.2588	0
	x1	34	36	1.050888	5.7E-07	*	*	*	*	419	421	32.55163	9.98E-07	36	37	4.040515	0
	x2	19	20	0.583534	9.57E-07	*	*	*	*	336	338	25.78158	6.42E-07	37	38	4.091604	0
100,000	x3	27	29	0.842917	5.23E-07	*	*	*	*	412	414	31.92259	7.41E-07	36	37	4.121649	0
	x4	44	46	1.360681	6.09E-07	*	*	*	*	1001	1002	78.36572	0.000458	36	37	4.179227	0
	x5	34	35	1.046282	9.03E-07	*	*	*	*	517	519	40.58637	9.89E-07	36	37	4.075484	0
	x1	34	36	1.125758	5.7E-07	*	*	*	*	419	421	32.94272	9.98E-07	36	37	4.118306	0
	x2	19	20	0.649241	9.57E-07	*	*	*	*	336	338	25.97922	6.42E-07	37	38	4.171178	0
100,000	x3	27	29	0.921228	5.23E-07	*	*	*	*	412	414	32.3447	7.41E-07	36	37	4.225511	0
	x4	44	46	1.47532	6.09E-07	*	*	*	*	1001	1002	78.1714	0.000458	36	37	4.114382	0
	x5	34	35	1.170896	9.03E-07	*	*	*	*	517	519	40.06687	9.89E-07	36	37	4.229835	0

Table 4. Numerical results for Algorithm 2, MCHCG, HADS, and Algorithm 2.1 on Problem 4.

		Algorithm 2				MCHCG				HADS				Algorithm 2.1			
#Dim	Ip	#Itr	#Fev	#Tm	#Nrm	#Itr	#Fev	#Tm	#Nrm	#Itr	#Fev	#Tm	#Nrm	#Itr	#Fev	#Tm	#Nrm
	x1	22	23	0.813282	6.54E-07	36	37	2.589909	7.83E-08	7	9	0.676359	3.08E-07	1	2	0.046442	0
	x2	1	2	0.035698	0	26	27	1.960226	4.55E-08	6	8	0.585377	1.7E-07	1	2	0.049196	0
100,000	x3	1	2	0.045269	0	2	3	0.141566	0	7	9	0.769638	1.72E-07	1	2	0.050782	0
	x4	38	39	1.20702	7.24E-07	46	47	3.334226	4.07E-08	133	134	11.17749	9.97E-07	1	2	0.048635	0
	x5	21	22	0.675671	7.68E-07	38	39	2.864539	2.85E-08	7	9	0.663764	3.09E-07	1	2	0.047405	0
	x1	22	23	0.804405	6.54E-07	36	37	2.586159	7.83E-08	7	9	0.647693	3.08E-07	1	2	0.045179	0
	x2	1	2	0.038183	0	26	27	1.928487	4.55E-08	6	8	0.551222	1.7E-07	1	2	0.121036	0
100,000	x3	1	2	0.041493	0	2	3	0.146236	0	7	9	0.641447	1.72E-07	1	2	0.108714	0
	x4	38	39	1.219435	7.24E-07	46	47	3.255958	4.07E-08	133	134	11.1045	9.97E-07	1	2	0.086699	0
	x5	21	22	0.750191	7.68E-07	38	39	2.734396	2.85E-08	7	9	0.642364	3.09E-07	1	2	0.078779	0
	x1	22	23	0.871927	6.54E-07	36	37	2.68629	7.83E-08	7	9	0.596102	3.08E-07	1	2	0.05703	0
	x2	1	2	0.043772	0	26	27	1.933983	4.55E-08	6	8	0.605753	1.7E-07	1	2	0.04789	0
100,000	x3	1	2	0.049794	0	2	3	0.167648	0	7	9	0.697415	1.72E-07	1	2	0.05904	0
	x4	38	39	1.231902	7.24E-07	46	47	3.282265	4.07E-08	133	134	10.99954	9.97E-07	1	2	0.045573	0
	x5	21	22	0.730241	7.68E-07	38	39	2.662892	2.85E-08	7	9	0.633918	3.09E-07	1	2	0.048171	0
	x1	22	23	0.785924	6.54E-07	36	37	2.734258	7.83E-08	7	9	0.695586	3.08E-07	1	2	0.051019	0
	x2	1	2	0.032695	0	26	27	1.893487	4.55E-08	6	8	0.566871	1.7E-07	1	2	0.045434	0
100,000	x3	1	2	0.041651	0	2	3	0.143815	0	7	9	0.66912	1.72E-07	1	2	0.051532	0
	x4	38	39	1.220087	7.24E-07	46	47	3.367007	4.07E-08	133	134	11.05593	9.97E-07	1	2	0.044124	0
	x5	21	22	0.704789	7.68E-07	38	39	2.693593	2.85E-08	7	9	0.66729	3.09E-07	1	2	0.039765	0
	x1	22	23	0.764946	6.54E-07	36	37	2.613091	7.83E-08	7	9	0.661077	3.08E-07	1	2	0.042246	0
	x2	1	2	0.039241	0	26	27	1.900018	4.55E-08	6	8	0.57261	1.7E-07	1	2	0.049199	0
100,000	x3	1	2	0.043035	0	2	3	0.136973	0	7	9	0.637069	1.72E-07	1	2	0.045578	0
	x4	38	39	1.139853	7.24E-07	46	47	3.363421	4.07E-08	133	134	11.97765	9.97E-07	1	2	0.041133	0
	x5	21	22	0.73901	7.68E-07	38	39	2.739253	2.85E-08	7	9	0.707573	3.09E-07	1	2	0.044485	0

Table 5. Numerical results for Algorithm 2, MCHCG, HADS, and Algorithm 2.1 on Problem 5.

		Algorithm 2				MCHCG				HADS				Algorithm 2.1			
#Dim	Ip	#Itr	#Fev	#Tm	#Nrm	#Itr	#Fev	#Tm	#Nrm	#Itr	#Fev	#Tm	#Nrm	#Itr	#Fev	#Tm	#Nrm
	x1	9	10	0.379312	5.42E-07	21	23	2.068601	4.52E-08	3	5	0.321346	1.95E-09	71	73	5.660237	6.54E-08
	x2	8	10	0.357172	9.22E-07	*	*	*	*	2	4	0.258081	2.98E-07	78	80	6.218231	7.77E-08
100,000	x3	12	14	0.5358	8.4E-07	16	18	1.584574	5.33E-08	2	4	0.25511	2.64E-08	29	31	2.373699	6.07E-08
	x4	9	10	0.360082	5.42E-07	30	32	2.739896	5.32E-08	4	6	0.393793	2.81E-08	71	73	5.878822	6.51E-08
	x5	9	10	0.405238	2.8E-07	20	22	1.762054	5.17E-08	3	5	0.346232	8.55E-10	71	73	5.572851	6.54E-08
	x1	9	10	0.343833	5.42E-07	21	23	1.859457	4.52E-08	3	5	0.369905	1.95E-09	71	73	5.57052	6.54E-08
	x2	8	10	0.383186	9.22E-07	*	*	*	*	2	4	0.334458	2.98E-07	78	80	6.127143	7.77E-08
100,000	x3	12	14	0.525343	8.4E-07	16	18	1.466211	5.33E-08	2	4	0.304432	2.64E-08	29	31	2.335001	6.07E-08
	x4	9	10	0.346769	5.42E-07	30	32	2.606673	5.32E-08	4	6	0.511877	2.81E-08	71	73	5.581751	6.51E-08
	x5	9	10	0.358322	2.8E-07	20	22	1.766103	5.17E-08	3	5	0.396942	8.55E-10	71	73	5.534518	6.54E-08
	x1	9	10	0.322944	5.42E-07	21	23	1.88567	4.52E-08	3	5	0.423447	1.95E-09	71	73	5.69181	6.54E-08
	x2	8	10	0.346729	9.22E-07	*	*	*	*	2	4	0.261846	2.98E-07	78	80	6.062298	7.77E-08
100,000	x3	12	14	0.519557	8.4E-07	16	18	1.679169	5.33E-08	2	4	0.251704	2.64E-08	29	31	2.407673	6.07E-08
	x4	9	10	0.380265	5.42E-07	30	32	2.671953	5.32E-08	4	6	0.448461	2.81E-08	71	73	5.728548	6.51E-08
	x5	9	10	0.373541	2.8E-07	20	22	1.799711	5.17E-08	3	5	0.30798	8.55E-10	71	73	5.54644	6.54E-08
	x1	9	10	0.353525	5.42E-07	21	23	1.879992	4.52E-08	3	5	0.415017	1.95E-09	71	73	5.507722	6.54E-08
	x2	8	10	0.348463	9.22E-07	*	*	*	*	2	4	0.233203	2.98E-07	78	80	6.290828	7.77E-08
100,000	x3	12	14	0.520063	8.4E-07	16	18	1.389905	5.33E-08	2	4	0.22689	2.64E-08	29	31	2.352558	6.07E-08
	x4	9	10	0.37768	5.42E-07	30	32	2.66176	5.32E-08	4	6	0.393696	2.81E-08	71	73	5.695666	6.51E-08
	x5	9	10	0.366326	2.8E-07	20	22	1.860621	5.17E-08	3	5	0.243026	8.55E-10	71	73	5.635476	6.54E-08
	x1	9	10	0.410104	5.42E-07	21	23	1.886573	4.52E-08	3	5	0.367092	1.95E-09	71	73	5.704925	6.54E-08
	x2	8	10	0.350388	9.22E-07	*	*	*	*	2	4	0.218369	2.98E-07	78	80	6.549499	7.77E-08
100,000	x3	12	14	0.497245	8.4E-07	16	18	1.505276	5.33E-08	2	4	0.210094	2.64E-08	29	31	2.378063	6.07E-08
	x4	9	10	0.359478	5.42E-07	30	32	3.061022	5.32E-08	4	6	0.382815	2.81E-08	71	73	5.719536	6.51E-08
	x5	9	10	0.364039	2.8E-07	20	22	2.260423	5.17E-08	3	5	0.323277	8.55E-10	71	73	5.732669	6.54E-08

Table 6. Summary of the test results reported in Tables 1–5.

Methods	#Itr	#Fev	#Tm	#Nrm
Algorithm 2	13.72	15	0.46731775	3.94311E-07
HADS	258.56	260.12	18.1890909	400
MCHCG	118.24	120.16	10.76984002	1.56203E-05
Algorithm 2.1	31.68	33.08	2.694376234	9.06482E-08

5. Application description

One of the main challenges in statistical inference and signal processing is identifying sparse solutions of ill-conditioned linear systems. In this setting, underdetermined linear systems of the form $Dx = b$ are considered, where a signal is represented in an n-dimensional space and recovered from k observations with $k \ll n$. This leads to minimizing an objective that combines a quadratic ℓ_2 data-fidelity term with a sparse ℓ_1-regularization term:

$$\min_{x\in\mathbb{R}^n} f(x) := \frac{1}{2}\|Dx - b\|_2^2 + \ddot{\theta}\|x\|_1, \tag{5.1}$$

where $x \in \mathbb{R}^n$, $b \in \mathbb{R}^k$, and $D \in \mathbb{R}^{k\times n}$ are linear operators, and $\ddot{\theta} > 0$ is a regularization parameter. Owing to the sparsity of the original signal, compressive sensing leads to the unconstrained convex optimization problem (5.1), whose solution enables accurate recovery. For additional details, see [31–34]. Several researchers have proposed effective approaches for solving (5.1). For instance, the procedure suggested by Figueiredo and Nowak [31] is as follows. If $x \in \mathbb{R}^n$, then x can be expressed as

$$x = x_+ - x_-, \; x_+ \geq 0, \; x_- \geq 0,$$

where $x_+ = \max\{x_i, 0\}$, $x_- = \max\{-x_i, 0\}$ for all $i = 1, 2, ..., n$. Using $\|x\|_1 = e_n^\top x_+ + e_n^\top x_-$, where $e_n = (1, 1, ..., 1)^\top \in \mathbb{R}^n$. Thus, using this relation, problem (5.1) can be expressed as

$$\min_{x_+,x_-} \ddot{\theta}e_n^\top x_+ + \ddot{\theta}e_n^\top x_- + \frac{1}{2}\|D(x_+ - x_-) - b\|_2^2, \qquad x_+, x_- \geq 0. \tag{5.2}$$

This is a constrained quadratic program. Nonetheless, problem (5.2) can be further simplified as follows:

$$\min_u \frac{1}{2}u^\top Bu + c^\top u, \qquad \text{such that} \quad u \geq 0, \tag{5.3}$$

where

$$u = \begin{bmatrix} x_+ \\ x_- \end{bmatrix}, \quad c = \begin{bmatrix} \ddot{\theta}e_n - D^\top b \\ \ddot{\theta}e_n + D^\top b \end{bmatrix}, \quad B = \begin{bmatrix} D^\top D & -D^\top D \\ -D^\top D & D^\top D \end{bmatrix}.$$

It is evident that B is positive semidefinite. Consequently, (5.3) is a convex quadratic program [31]. By the optimality conditions, u is a minimizer of (5.3). Therefore, u satisfies

$$E(u) = \min\{u_i, (Bu + c)_i\} = 0, \tag{5.4}$$

where E is a vector-valued function, and **min** is interpreted componentwise. According to Lemma 3 in [35] and Lemma 2.2 in [36], the mapping $E(x)$ in (5.4) is Lipschitz continuous and monotone. Therefore, solving (5.1) is equivalent to solving (5.4), which has the same structure as (1.1).

5.1. Signal restoration problem

In the first experiment, recovery of a length-n sparse signal from k observations is evaluated using the mean squared error (MSE). The MSE is defined as follows:

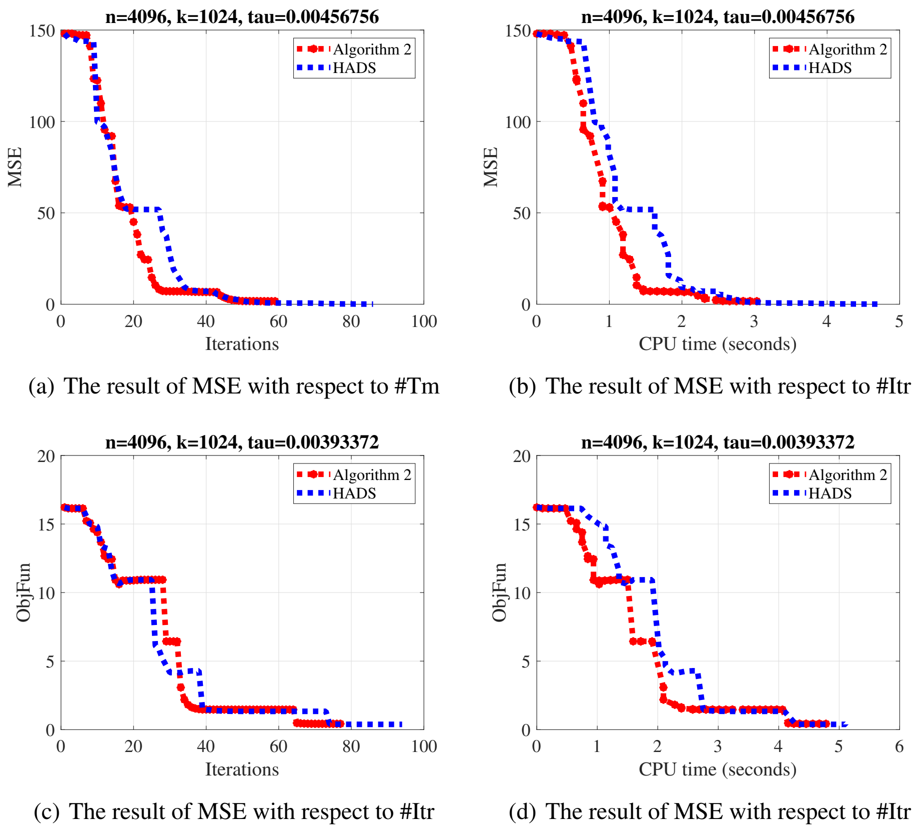
$$MSE := \frac{1}{n}\|\overline{x} - \underline{x}\|^2. \tag{5.5}$$

The original signal is denoted by $\overline{x}$, while the recovered signal is denoted by $\underline{x}$. The matrix D in (5.1) is generated using the MATLAB function $\mathtt{randn}(k, n)$, and the observed data are defined as $b = Dx + \varepsilon$, where $\varepsilon \sim N(0, \overline{\sigma}^2)$ with $\overline{\sigma}^2 = 0.0001$. The estimate $\ddot{\theta}$ is obtained from [37] for all comparison methods, and the initial point is set to $x_0 = D^\top$. The proposed Algorithm 2 is compared with the HADS method [16], using the parameter settings reported in the original article.

Additionally, the following parameters are used in the implementation of the proposed Algorithm 2: $n = 100000$, $\overline{\alpha} = 0.8$, $\overline{\beta} = 0.05$, $(\overline{\gamma}_{(k)} + 1) = 1.76$, $\sigma = 10^{-4}$, $\rho = 0.7$, and $\gamma = 0.01$. The stopping criterion is set as

$$\frac{|f(x_k) - f(x_{k-1})|}{|f(x_{k-1})|} < 10^{-5}.$$

The sparse signal recovery results obtained by Algorithm 2 and the HADS method are shown in Figures 3 and 4. The figures clearly indicate that Algorithm 2 requires fewer #Itr and less #Tm than the HADS method. With fewer repetitions and lower computational time, Algorithm 2 recovers the sparse signal with a lower MSE. Consequently, in these respects, Algorithm 2 is superior to the HADS method.



(a) The result of MSE with respect to #Tm (b) The result of MSE with respect to #Itr

(c) The result of MSE with respect to #Itr (d) The result of MSE with respect to #Itr

Figure 3. Signal processing results using Algorithm 2 and HADS methods.

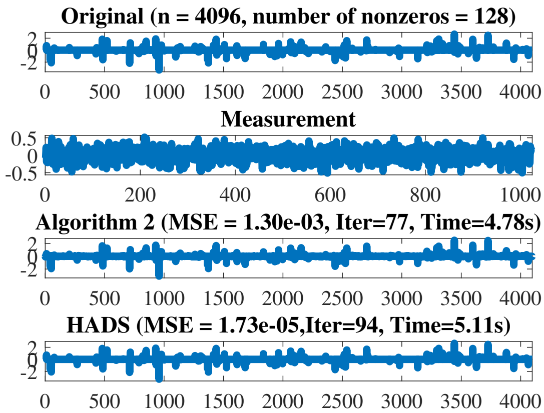


Figure 4. From highest to lowest: the initial signal, the measured signal, and the reconstructed signals by Algorithm 2 and the HADS method.

6. Conclusions, limitation, and future work

This work presents a double step-length approach for solving large-scale nonlinear equations with convex constraints. By combining the Picard-Mann iterative technique with a newly defined function, we developed a hybrid iterative procedure that introduces a correction parameter in the search direction and significantly improves the convergence rate. The acceleration parameter was derived using the Frobenius norm based on the Broyden update and its diagonal approximation. Under appropriate assumptions, global convergence and a Q-linear convergence rate of the proposed Algorithm 2 were established. To assess the effectiveness of Algorithm 2, numerical simulations were conducted against existing methods. As shown in Figure 2, the proposed algorithm demonstrates high efficiency across the experiments, achieving low #Itr, #Tm, and #Fev. In addition, signal processing problems were solved using Algorithm 2. Performance was evaluated using #MSE, #Tm, and #Itr, and the results in Figures 3 and 4 show that the proposed method solves the problems with low #Itr and #Tm.

For future work, promising directions include extending the proposed algorithms to nonconvex settings, developing adaptive or self-tuning parameter-selection strategies, and exploring applications in machine learning and engineering optimization to further improve robustness and practical impact.

Limitations and outlook: We acknowledge that the proposed method's applicability is bounded by its theoretical foundations of the monotonicity operator with convex constraints. It is not designed for general non-monotone or non-convex problems in its current form. Future research will focus on overcoming these limitations by investigating relaxations of the monotonicity condition (e.g., to pseudo-monotonicity or co-coercivity) while retaining the double step-length structure. Additionally, future research will explore operator-splitting or relaxation techniques to handle specific types of non-convex constraints.

Author contributions

Muhammad Abdullahi: Conceptualization, Methodology, Investigation, Data curation, Writing–original draft preparation; Abubakar Sani Halilu: Validation, Methodology, Investigation, Supervision;

Mohammed A. Saleh: Software, Resources, Writing–review and editing, Supervision, Funding acquisition; Abdulgader Z. Almaymuni: Software, Validation, Resources, Writing–review and editing, Funding acquisition; Seyed Yaser Mousavi Siamakani: Conceptualization, Methodology, Investigation, Data curation, Supervision; Tiamiyu Abd'gafar Tunde: Conceptualization, Validation, Investigation, Writing–review and editing; Sulaiman M. Ibrahim: Software, Formal analysis, Data curation, Writing–review and editing, Visualization. All authors have read and agreed to the published version of the manuscript.

Use of Generative-AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this paper.

Acknowledgments

The Researchers would like to thank the Deanship of Graduate Studies and Scientific Research at Qassim University for financial support (QU-APC-2026).

Conflict of interest

The authors declare no conflicts of interest.

References

1. Z. F. Dai, X. D. Dong, J. Kang, L. Y. Hong, Forecasting stock market returns: new technical indicators and two-step economic constraint method, *N. Am. J. Econ. Finance*, **53** (2020), 101216. https://doi.org/10.1016/j.najef.2020.101216
2. K. Meintjes, A. P. Morgan, A methodology for solving chemical equilibrium systems, *Appl. Math. Comput.*, **22** (1987), 333–361. https://doi.org/10.1016/0096-3003(87)90076-2
3. M. Abdullahi, A. S. Halilu, K. J. Pan, A. A. Bala, Two double direction methods for convex-constrained nonlinear monotone equations with image recovery of spherical parts, *Commun. Nonlinear Sci. Numer. Simul.*, **152** (2026), 109315. https://doi.org/10.1016/j.cnsns.2025.109315
4. J. M. Martínez, Practical quasi-Newton methods for solving nonlinear systems, *J. Comput. Appl. Math.*, **124** (2000), 97–121. https://doi.org/10.1016/S0377-0427(00)00434-9
5. D. H. Li, M. Fukushima, A globally and superlinearly convergent Gauss-Newton-based BFGS method for symmetric nonlinear equations, *SIAM J. Numer. Anal.*, **37** (1999), 152–172.
6. S. Bellavia, M. Macconi, B. Morini, STRSCNE: a scaled trust-region solver for constrained nonlinear equations, *Comput. Optim. Appl.*, **28** (2004), 31–50. https://doi.org/10.1023/B:COAP.0000018878.95983.4e
7. N. Yamashita, M. Fukushima, On the rate of convergence of the Levenberg-Marquardt method, In: *Topics in numerical analysis*, Vienna: Springer, 2001, 239–249. https://doi.org/10.1007/978-3-7091-6217-0_18
8. J. Nocedal, S. J. Wright, *Numerical optimization*, New York: Springer, 2006. https://doi.org/10.1007/978-0-387-40065-5

9. M. Raydan, The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem, *SIAM J. Optim.*, **7** (1997), 26–33. https://doi.org/10.1137/S1052623494266365

10. P. S. Stanimirović, M. B. Miladinović, Accelerated gradient descent methods with line search, *Numer. Algorithms*, **54** (2010), 503–520. https://doi.org/10.1007/s11075-009-9350-8

11. M. J. Petrović, An accelerated double step size model in unconstrained optimization, *Appl. Math. Comput.*, **250** (2015), 309–319. https://doi.org/10.1016/j.amc.2014.10.104

12. M. J. Petrović, P. S. Stanimirović, Accelerated double direction method for solving unconstrained optimization problems, *Math. Probl. Eng.*, **2014** (2014), 965104. https://doi.org/10.1155/2014/965104

13. A. S. Halilu, M. Y. Waziri, Y. B. Musa, Inexact double step length method for solving systems of nonlinear equations, *Stat. Optim. Inf. Comput.*, **8** (2020), 165–174. https://doi.org/10.19139/soic-2310-5070-532

14. M. J. Petrović, Hybridization rule applied on accelerated double step size optimization scheme, *Filomat*, **33** (2019), 655–665.

15. M. J. Petrović, V. Rakočević, D. Valjarević, D. Ilić, A note on hybridization process applied on transformed double step size model, *Numer. Algorithms*, **85** (2020), 449–465. https://doi.org/10.1007/s11075-019-00821-8

16. G. Abbass, H. B. Chen, M. Abdullahi, A. S. Halilu, I. A. Memon, Hybrid accelerated double step length approach for convex constrained nonlinear monotone equations with applications, *IMA J. Appl. Math.*, **90** (2025), 75–98. https://doi.org/10.1093/imamat/hxaf009

17. A. S. Halilu, M. A. Mohamed, M. A. Saleh, K. Ahmed, A. Z. Almaymuni, A. Majumder, et al., Modified iterative double step length model for solving nonlinear systems with application to motion control, *Sci. Rep.*, **15** (2025), 40764. https://doi.org/s41598-025-24537-z

18. S. H. Khan, A Picard-Mann hybrid iterative process, *Fixed Point Theory Appl.*, **2013** (2013), 69. https://doi.org/10.1186/1687-1812-2013-69

19. M. J. Petrović, P. S. Stanimirović, N. Kontrec, J. Mladenović, Hybrid modification of accelerated double direction method, *Math. Probl. Eng.*, **2018** (2018), 1523267. https://doi.org/10.1155/2018/1523267

20. A. S. Halilu, A. Majumder, M. Y. Waziri, A. M. Awwal, K. Ahmed, On solving double direction methods for convex constrained monotone nonlinear equations with image restoration, *Comput. Appl. Math.*, **40** (2021), 239. https://doi.org/10.1007/s40314-021-01624-1

21. J. K. Liu, B. Tang, T. Liu, Z. T. Yang, S. Liang, An accelerated double-step derivative-free projection method based algorithm using Picard-Mann iterative process for solving convex constrained nonlinear equations, *J. Comput. Appl. Math.*, **464** (2025), 116541. https://doi.org/10.1016/j.cam.2025.116541

22. M. V. Solodov, B. F. Svaiter, A globally convergent inexact Newton method for systems of monotone equations, In: *Reformulation: nonsmooth, piecewise smooth, semismooth and smoothing methods*, New York: Springer, 1999, 355–369. https://doi.org/10.1007/978-1-4757-6388-1_18

23. É. Picard, Mémoire sur la théorie des équations aux dérivées partielles et la méthode des approximations successives, *J. Math. Pures Appl.*, **6** (1890), 145–210.
24. W. R. Mann, Mean value methods in iteration, *Proc. Amer. Math. Soc.*, **4** (1953), 506–510. https://doi.org/10.2307/2032162
25. S. Ishikawa, Fixed points by a new iteration method, *Proc. Amer. Math. Soc.*, **44** (1974), 147–150. https://doi.org/10.1090/S0002-9939-1974-0336469-5
26. E. Nermeh, M. Abdullahi, A. S. Halilu, H. Abdullahi, Modification of a conjugate gradient approach for convex constrained nonlinear monotone equations with applications in signal recovery and image restoration, *Commun. Nonlinear Sci. Numer. Simul.*, **136** (2024), 108079. https://doi.org/10.1016/j.cnsns.2024.108079
27. N. Zhang, J. K. Liu, B. Tang, A three-term projection method based on spectral secant equation for nonlinear monotone equations, *Japan J. Indust. Appl. Math.*, **41** (2024), 617–635. https://doi.org/10.1007/s13160-023-00624-4
28. W. La Cruz, J. M. Martínez, M. Raydan, Spectral residual method without gradient information for solving large-scale nonlinear systems of equations, *Math. Comput.*, **75** (2006), 1429–1448.
29. A. M. Awwal, P. Kumam, A. B. Abubakar, A modified conjugate gradient method for monotone nonlinear equations with convex constraints, *Appl. Numer. Math.*, **145** (2019), 507–520. https://doi.org/10.1016/j.apnum.2019.05.012
30. E. D. Dolan, J. J. Moré, Benchmarking optimization software with performance profiles, *Math. Program.*, **91** (2002), 201–213. https://doi.org/10.1007/s101070100263
31. M. A. T. Figueiredo, R. D. Nowak, An EM algorithm for wavelet-based image restoration, *IEEE Trans. Image Process.*, **12** (2003), 906–916. https://doi.org/10.1109/TIP.2003.814255
32. A. Beck, M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems, *SIAM J. Imaging Sci.*, **2** (2009), 183–202. https://doi.org/10.1137/080716542
33. E. Van Den Berg, M. P. Friedlander, Probing the Pareto frontier for basis pursuit solutions, *SIAM J. Sci. Comput.*, **31** (2008), 890–912. https://doi.org/10.1137/080714488
34. E. G. Birgin, J. M. Martínez, M. Raydan, Nonmonotone spectral projected gradient methods on convex sets, *SIAM J. Optim.*, **10** (2000), 1196–1211. https://doi.org/10.1137/S1052623497330963
35. W. Sánchez, C. A. Arias, R. Perez, A Jacobian smoothing inexact Newton method for solving the nonlinear complementary problem, *Comput. Appl. Math.*, **43** (2024), 279. https://doi.org/10.1007/s40314-024-02775-7
36. Y. H. Xiao, Q. Y. Wang, Q. J. Hu, Non-smooth equations based method for ℓ_1-norm problems with applications to compressed sensing, *Nonlinear Anal.*, **74** (2011), 3570–3577. https://doi.org/10.1016/j.na.2011.02.040
37. M. A. T. Figueiredo, R. D. Nowak, S. J. Wright, Gradient projection for sparse reconstruction: application to compressed sensing and other inverse problems, *IEEE J. Sel. Top. Signal Process.*, **1** (2007), 586–597. https://doi.org/10.1109/JSTSP.2007.910281

AIMS Press