RBRU International Journal of Science and Technology 1 (1) (2026) 34–44

RBRU International Journal of Science and Technology

ISSN: 3088-389X (online) journal homepage: https://ph05.tci-thaijo.org/index.php/rbruijst

Research Article

A Self-Adaptive Inertial Multi-Step Projection Method for Large-Scale Monotone Nonlinear Systems

Pakeeta Sukprasert[a]

[a] *Faculty of Science and Technology Rajamangala University of Technology Thanyaburi, Pathum Thani 12110, Thailand*

ARTICLE INFO

Keywords:
Monotone nonlinear equations
Derivative-free method
Inertial extrapolation
Multi-step method
Projection method
Large-scale problems

Article history:
Received 12 August 2026
Revised 29 August 2026
Accepted 30 August 2026
Published 31 August 2026

ABSTRACT

This paper proposes a self-adaptive inertial multi-step projection method for solving large-scale systems of monotone nonlinear equations. The method combines a controlled inertial extrapolation with a derivative-free multi-step search direction, a backtracking line search, and a hyperplane projection correction. The search direction is specifically constructed to satisfy an exact descent property independently of the line-search procedure, thereby providing a suitable mechanism for establishing the global convergence of the method. To control the perturbations introduced by the inertial and multi-step terms, an adaptive inertial rule together with an implementable safeguard is incorporated into the algorithm. Under the nonemptiness of the solution set and the monotonicity and Lipschitz continuity of the mapping, the line search is shown to be well defined. Moreover, the boundedness of the generated sequence, the vanishing of the residual, and the global convergence of the entire sequence to a solution of the nonlinear system are established. The proposed method is derivative-free and matrix-free, and therefore preserves the low-storage and computationally inexpensive structure desirable for large-scale problems. Numerical experiments on large-scale test problems demonstrate the computational behavior of the method in terms of iteration counts, function evaluations, CPU time, and residual reduction.

1. Introduction

Systems of nonlinear equations arise in many areas of applied mathematics, scientific computing, engineering, signal processing, inverse problems, and optimization. We consider the problem of finding a point $x^* \in \mathbb{R}^n$ such that

$$F(x^*) = 0, \tag{1.1}$$

where $F : \mathbb{R}^n \to \mathbb{R}^n$ is a continuous monotone mapping. When the dimension is large, classical Newton and quasi-Newton methods may become computationally demanding because they typically require Jacobian information and the solution of associated linear systems. This has motivated derivative-free iterative methods that require only function evaluations and vector operations.

*Corresponding author
DOI:10.68654/rbruijst.2026.01.04
pakeeta_s@rmutt.ac.th (P. Sukprasert)
This is an open access article under the Creative Commons Attribution – NonCommercial – NoDerivatives 4.0 license.
© 2026 RBRU International Journal of Science and Technology – Published by Rambhai Barni Rajabhat University.

Projection-type methods constitute an important class of algorithms for nonlinear monotone equations. Their basic idea is to use a line search to construct an auxiliary point and a separating hyperplane, followed by a projection-type correction that drives the iteration toward the solution set. Such methods are particularly attractive for large-scale problems because they avoid explicit Jacobian information while retaining useful global convergence properties [1].

Conjugate-gradient-type projection methods have received considerable attention because of their low storage requirements and simple computational structure. By replacing gradient information with the residual $F(x_k)$, derivative-free search directions can be constructed for nonlinear monotone systems. A three-term derivative-free projection method was developed in [7], and related projection directions were subsequently investigated in [6, 8].

Another important development concerns adaptive parameter and stepsize strategies. Fixed parameters that perform well for one problem may be inefficient for another, which motivates the use of iteration-dependent information. A self-adaptive projection method for monotone nonlinear equations with convex constraints was studied in [9].

Inertial techniques provide a complementary mechanism by incorporating information from previous iterates. A typical inertial extrapolation takes the form

$$w_k = x_k + \theta_k(x_k - x_{k-1}),$$

where $\theta_k \geq 0$. Projection methods incorporating inertial steps have been studied, for example, in [3, 10]. Appropriately controlled inertial information can improve computational behavior without changing the derivative-free nature of the underlying algorithm.

The present work combines inertial extrapolation with multi-step information. A multi-step strategy uses several previous displacements in constructing the current direction, whereas the inertial mechanism extrapolates from the two most recent iterates. Their combination must be controlled because an unrestricted perturbation may destroy the descent and quasi-Fejér properties used in global convergence analysis.

Motivated by this observation, we propose a self-adaptive inertial multi-step projection method (AIMSPM). The method constructs an extrapolated point w_k, forms a derivative-free multi-step direction at w_k, performs a backtracking line search, and applies a projection correction. The direction is designed to satisfy the exact identity

$$\langle F(w_k), d_k\rangle = -\|F(w_k)\|^2,$$

independently of the line search.

The main contributions are as follows:

(i) an adaptive inertial mechanism is integrated into a multi-step projection framework for large-scale monotone nonlinear equations;
(ii) the proposed derivative-free multi-step direction preserves an exact descent identity at the extrapolated point;
(iii) under monotonicity, Lipschitz continuity, nonemptiness of the solution set, summable inertial control, and a multi-step safeguard, the line search is well defined and the generated sequence converges globally to a solution; and
(iv) numerical results are reported to illustrate the computational behavior of the method on large-scale test instances.

AIMSPM is not intended to claim universal acceleration over all existing projection methods. Its purpose is to provide a controlled mechanism that combines inertial extrapolation and multi-step information while preserving a tractable descent structure.

The remainder of the paper is organized as follows. Section 2 presents the assumptions and preliminary results. Section 3 introduces AIMSPM. Section 4 provides the convergence analysis. Section 5 reports numerical results, and Section 6 concludes the paper.

2. Preliminaries

Throughout the paper, $\mathbb{R}^n$ denotes the n-dimensional Euclidean space equipped with the standard inner product $\langle\cdot,\cdot\rangle$ and induced norm $\|\cdot\|$. The solution set of (1.1) is

$$S := \{x \in \mathbb{R}^n : F(x) = 0\}.$$

Definition 2.1. *A mapping $F : \mathbb{R}^n \to \mathbb{R}^n$ is* monotone *if*

$$\langle F(x) - F(y), x - y\rangle \geq 0, \qquad \forall x, y \in \mathbb{R}^n.$$

If $x^* \in S$, then $F(x^*) = 0$, and monotonicity gives

$$\langle F(x), x - x^* \rangle \geq 0, \qquad \forall x \in \mathbb{R}^n. \tag{2.1}$$

Definition 2.2. *A mapping $F : \mathbb{R}^n \to \mathbb{R}^n$ is L-Lipschitz continuous if there exists $L > 0$ such that*

$$\|F(x) - F(y)\| \leq L \|x - y\|, \qquad \forall x, y \in \mathbb{R}^n.$$

Assumption 2.3. *The following conditions hold:*

(A1) $S \neq \emptyset$;
(A2) F *is monotone on* $\mathbb{R}^n$;
(A3) F *is L-Lipschitz continuous for some* $L > 0$.

Lemma 2.4. *For any $x, y, z \in \mathbb{R}^n$,*

$$\|x - z\|^2 = \|y - z\|^2 + 2\langle x - y, y - z \rangle + \|x - y\|^2. \tag{2.2}$$

Lemma 2.5. *Let $x^* \in S$ and $z \in \mathbb{R}^n$. Then*

$$\langle F(z), z - x^* \rangle \geq 0.$$

Consequently,

$$S \subseteq H_z := \{x \in \mathbb{R}^n : \langle F(z), x - z \rangle \leq 0\}.$$

Proof. Since $F(x^*) = 0$, monotonicity yields

$$\langle F(z) - F(x^*), z - x^* \rangle \geq 0,$$

which gives the first assertion. Equivalently, $\langle F(z), x^* - z \rangle \leq 0$, and hence $x^* \in H_z$. ∎

Lemma 2.6. *Let*

$$w_k = x_k + \theta_k(x_k - x_{k-1}), \qquad \theta_k \geq 0.$$

Then, for every $x^* \in \mathbb{R}^n$,

$$\|w_k - x^*\|^2 = (1 + \theta_k)\|x_k - x^*\|^2 - \theta_k \|x_{k-1} - x^*\|^2 + \theta_k(1 + \theta_k)\|x_k - x_{k-1}\|^2. \tag{2.3}$$

Proof. Since

$$w_k - x^* = (1 + \theta_k)(x_k - x^*) - \theta_k(x_{k-1} - x^*),$$

the result follows from

$$\|(1 + t)a - tb\|^2 = (1 + t)\|a\|^2 - t\|b\|^2 + t(1 + t)\|a - b\|^2, \qquad t \geq 0.$$ ∎

To control the inertial perturbation, assume

$$\theta_k \|x_k - x_{k-1}\| \leq \varepsilon_k, \qquad \sum_{k=1}^{\infty} \varepsilon_k < \infty. \tag{2.4}$$

A convenient adaptive choice is

$$\theta_k = \begin{cases} \min\left\{\theta_{\max}, \dfrac{\varepsilon_k}{\|x_k - x_{k-1}\|}\right\}, & x_k \neq x_{k-1}, \\ \theta_{\max}, & x_k = x_{k-1}, \end{cases} \tag{2.5}$$

where $\theta_{\max} > 0$ and $\{\varepsilon_k\}$ is positive and summable.

Lemma 2.7. *Let $\{a_k\}$ and $\{\eta_k\}$ be nonnegative sequences satisfying*

$$a_{k+1} \leq a_k + \eta_k, \qquad \sum_{k=1}^{\infty} \eta_k < \infty.$$

Then $\{a_k\}$ converges.

3. Proposed Self-Adaptive Inertial Multi-Step Projection Method

We now introduce AIMSPM for solving (1.1).

3.1. Adaptive inertial extrapolation

Given the two most recent iterates x_{k-1} and x_k, define

$$w_k = x_k + \theta_k(x_k - x_{k-1}), \tag{3.1}$$

where θ_k is given by (2.5). Consequently,

$$\theta_k \|x_k - x_{k-1}\| \le \varepsilon_k, \qquad \sum_{k=1}^{\infty} \varepsilon_k < \infty.$$

3.2. Multi-step derivative-free direction

Set

$$g_k := F(w_k).$$

If $\|g_k\| = 0$, then w_k is a solution and the iteration terminates. Otherwise, define

$$s_{k-1} = w_k - w_{k-1}. \tag{3.2}$$

For a memory length $1 \le m_k \le m$, define

$$\bar{s}_{k-1} = \sum_{j=1}^{m_k} \omega_{k,j} s_{k-j}, \qquad \omega_{k,j} \ge 0, \qquad \sum_{j=1}^{m_k} \omega_{k,j} = 1. \tag{3.3}$$

The search direction is

$$d_k = -g_k + \beta_k \left(\bar{s}_{k-1} - \frac{\langle g_k, \bar{s}_{k-1} \rangle}{\|g_k\|^2} g_k \right). \tag{3.4}$$

The correction term is orthogonal to g_k, and hence

$$\langle F(w_k), d_k \rangle = -\|F(w_k)\|^2. \tag{3.5}$$

For the convergence analysis, set

$$q_k := \bar{s}_{k-1} - \frac{\langle g_k, \bar{s}_{k-1} \rangle}{\|g_k\|^2} g_k, \tag{3.6}$$

and impose the implementable safeguard

$$\|\beta_k q_k\| \le \kappa \|g_k\|, \qquad \kappa > 0. \tag{3.7}$$

For example, if $\widehat{\beta}_k$ is any trial multi-step coefficient, one may use

$$\beta_k = \begin{cases} \operatorname{sgn}(\widehat{\beta}_k) \min\left\{ |\widehat{\beta}_k|, \dfrac{\kappa \|g_k\|}{\|q_k\|} \right\}, & q_k \neq 0, \\ 0, & q_k = 0. \end{cases} \tag{3.8}$$

3.3. Backtracking line search and projection correction

Choose the smallest integer $j_k \geq 0$ such that, with

$$\alpha_k = \rho\eta^{j_k}, \qquad \rho > 0, \quad \eta \in (0,1), \quad \sigma > 0, \tag{3.9}$$

the line-search condition

$$-\langle F(w_k + \alpha_k d_k), d_k\rangle \geq \sigma\alpha_k \left\|F(w_k + \alpha_k d_k)\right\| \left\|d_k\right\|^2 \tag{3.10}$$

holds. Define

$$z_k = w_k + \alpha_k d_k. \tag{3.11}$$

Then

$$\langle F(z_k), w_k - z_k\rangle \geq \sigma\alpha_k^2 \left\|F(z_k)\right\| \left\|d_k\right\|^2 > 0.$$

Set

$$\xi_k = \frac{\langle F(z_k), w_k - z_k\rangle}{\left\|F(z_k)\right\|^2}, \tag{3.12}$$

and update

$$x_{k+1} = w_k - \xi_k F(z_k). \tag{3.13}$$

Algorithm 1 Self-Adaptive Inertial Multi-Step Projection Method (AIMSPM)

1: **Initialization.** Choose $x_0, x_1 \in \mathbb{R}^n$, $\rho > 0$, $\eta \in (0,1)$, $\sigma > 0$, $\theta_{\max} > 0$, $\kappa > 0$, a tolerance Tol > 0, and a positive summable sequence $\{\varepsilon_k\}$.
2: **For** $k = 1, 2, \ldots$, **perform Steps 1–7.**
3: **Step 1.** Compute θ_k from (2.5) and set $w_k = x_k + \theta_k(x_k - x_{k-1})$.
4: **Step 2.** Compute $g_k = F(w_k)$. If $\left\|g_k\right\| \leq$ Tol, stop.
5: **Step 3.** Construct $\bar{s}_{k-1}$ from (3.3), form q_k from (3.6), and choose β_k satisfying (3.7).
6: **Step 4.** Compute $d_k = -g_k + \beta_k q_k$.
7: **Step 5.** Find the smallest $j_k \geq 0$ such that $\alpha_k = \rho\eta^{j_k}$ satisfies (3.10).
8: **Step 6.** Set $z_k = w_k + \alpha_k d_k$ and compute ξ_k from (3.12).
9: **Step 7.** Set $x_{k+1} = w_k - \xi_k F(z_k)$; increase k by one and repeat.

4. Convergence Analysis

Throughout this section, Assumption 2.3, the inertial control (2.4), and the safeguard (3.7) are assumed.

Lemma 4.1 (Sufficient descent). *If $F(w_k) \neq 0$, then*

$$\langle F(w_k), d_k\rangle = -\left\|F(w_k)\right\|^2, \tag{4.1}$$

and

$$\left\|F(w_k)\right\| \leq \left\|d_k\right\| \leq (1+\kappa)\left\|F(w_k)\right\|. \tag{4.2}$$

Proof. By definition, $\langle g_k, q_k\rangle = 0$. Therefore,

$$\langle g_k, d_k\rangle = -\left\|g_k\right\|^2 + \beta_k \langle g_k, q_k\rangle = -\left\|g_k\right\|^2,$$

which proves (4.1). By Cauchy–Schwarz,

$$\left\|g_k\right\|^2 = -\langle g_k, d_k\rangle \leq \left\|g_k\right\| \left\|d_k\right\|,$$

so $\left\|g_k\right\| \leq \left\|d_k\right\|$. Moreover,

$$\left\|d_k\right\| \leq \left\|g_k\right\| + \left\|\beta_k q_k\right\| \leq (1+\kappa)\left\|g_k\right\|. \qquad \blacksquare$$

Lemma 4.2. *If $F(w_k) \neq 0$, then the backtracking procedure (3.10) terminates after finitely many reductions.*

Proof. Assume that (3.10) fails for every $\alpha = \rho\eta^j$, $j = 0, 1, \ldots$. Then

$$-\langle F(w_k + \alpha d_k), d_k\rangle < \sigma\alpha \left\|F(w_k + \alpha d_k)\right\| \left\|d_k\right\|^2.$$

Letting $j \to \infty$ gives $\alpha \to 0$. By continuity of F,

$$-\langle F(w_k), d_k\rangle \leq 0,$$

which contradicts Lemma 4.1, since $-\langle F(w_k), d_k\rangle = \left\|F(w_k)\right\|^2 > 0$. ∎

Lemma 4.3. *Let $x^* \in S$. Then*

$$\left\|x_{k+1} - x^*\right\|^2 \leq \left\|w_k - x^*\right\|^2 - \delta_k^2, \tag{4.3}$$

where

$$\delta_k := \frac{\langle F(z_k), w_k - z_k\rangle}{\left\|F(z_k)\right\|}. \tag{4.4}$$

Furthermore,

$$\delta_k \geq \sigma\alpha_k^2 \left\|d_k\right\|^2. \tag{4.5}$$

Proof. Set

$$h_k := \langle F(z_k), w_k - z_k\rangle > 0.$$

Since $x^* \in S$, monotonicity yields

$$\langle F(z_k), z_k - x^*\rangle \geq 0,$$

and hence

$$\langle F(z_k), w_k - x^*\rangle \geq h_k.$$

Using $x_{k+1} = w_k - h_k F(z_k)/\left\|F(z_k)\right\|^2$, we obtain

$$\begin{aligned}\left\|x_{k+1} - x^*\right\|^2 &= \left\|w_k - x^*\right\|^2 - \frac{2h_k}{\left\|F(z_k)\right\|^2}\langle F(z_k), w_k - x^*\rangle + \frac{h_k^2}{\left\|F(z_k)\right\|^2} \\ &\leq \left\|w_k - x^*\right\|^2 - \frac{h_k^2}{\left\|F(z_k)\right\|^2},\end{aligned}$$

which gives (4.3). Since $w_k - z_k = -\alpha_k d_k$, the line-search condition yields

$$h_k = -\alpha_k \langle F(z_k), d_k\rangle \geq \sigma\alpha_k^2 \left\|F(z_k)\right\| \left\|d_k\right\|^2.$$

Dividing by $\left\|F(z_k)\right\|$ proves (4.5). ∎

Lemma 4.4. *For every $x^* \in S$, the sequence $\{\left\|x_k - x^*\right\|\}$ converges. Consequently, $\{x_k\}$, $\{w_k\}$, and $\{z_k\}$ are bounded.*

Proof. From (2.4),

$$\|w_k - x^*\| \le \|x_k - x^*\| + \varepsilon_k.$$

Combining this with (4.3) gives

$$\|x_{k+1} - x^*\| \le \|x_k - x^*\| + \varepsilon_k.$$

Lemma 2.7 implies that $\{\|x_k - x^*\|\}$ converges. Hence $\{x_k\}$ is bounded. Since

$$\|w_k - x_k\| \le \varepsilon_k \to 0,$$

$\{w_k\}$ is bounded. Lipschitz continuity implies boundedness of $\{F(w_k)\}$; Lemma 4.1 then gives boundedness of $\{d_k\}$. Since $0 < \alpha_k \le \rho$, the sequence $z_k = w_k + \alpha_k d_k$ is also bounded. ∎

Lemma 4.5. *The sequence generated by Algorithm 1 satisfies*

$$\alpha_k \|d_k\| \to 0, \tag{4.6}$$

and

$$\|F(w_k)\| \to 0. \tag{4.7}$$

Proof. Fix $x^* \in S$. By Lemma 4.4, there exists $M > 0$ such that $\|x_k - x^*\| \le M$. From (4.3) and $\|w_k - x^*\| \le \|x_k - x^*\| + \varepsilon_k$, there exists $C > 0$ such that

$$\delta_k^2 \le \|x_k - x^*\|^2 - \|x_{k+1} - x^*\|^2 + C\varepsilon_k + \varepsilon_k^2.$$

Summing over k and using $\sum_k \varepsilon_k < \infty$ yields

$$\sum_{k=1}^{\infty} \delta_k^2 < \infty, \qquad \text{hence} \qquad \delta_k \to 0.$$

By (4.5),

$$\sigma\alpha_k^2 \|d_k\|^2 \le \delta_k \to 0,$$

which proves (4.6).

Suppose, by contradiction, that there exist $\nu > 0$ and an infinite index set $\mathcal{K}$ such that

$$\|F(w_k)\| \ge \nu, \qquad k \in \mathcal{K}.$$

If $\alpha_k < \rho$, then the previous trial value $\widehat{\alpha}_k = \alpha_k/\eta$ fails the line search. Thus

$$-\left\langle F(w_k + \widehat{\alpha}_k d_k), d_k \right\rangle < \sigma\widehat{\alpha}_k \|F(w_k + \widehat{\alpha}_k d_k)\| \|d_k\|^2.$$

Using Lipschitz continuity together with (4.1), and boundedness of the trial points, there exists $M_F > 0$ such that

$$\|F(w_k)\|^2 < (L + \sigma M_F)\widehat{\alpha}_k \|d_k\|^2.$$

By (4.2),

$$\widehat{\alpha}_k > \frac{1}{(L + \sigma M_F)(1+\kappa)^2}.$$

Consequently, for $k \in \mathcal{K}$,

$$\alpha_k \ge \underline{\alpha} := \min\left\{\rho, \frac{\eta}{(L + \sigma M_F)(1+\kappa)^2}\right\} > 0.$$

Then (4.6) implies $\|d_k\| \to 0$ along $\mathcal{K}$. Since $\|F(w_k)\| \le \|d_k\|$ by Lemma 4.1, this contradicts $\|F(w_k)\| \ge \nu$. Therefore (4.7) holds. ∎

Theorem 4.6. *Suppose Assumption 2.3 holds, the inertial parameters satisfy (2.4), and the multi-step correction satisfies (3.7). Then the sequence $\{x_k\}$ generated by Algorithm 1 converges to a point $\bar{x} \in \mathcal{S}$.*

Proof. By Lemma 4.4, $\{w_k\}$ is bounded and therefore possesses a convergent subsequence $w_{k_j} \to \bar{x}$. By Lemma 4.5, $F(w_{k_j}) \to 0$. Continuity of F implies $F(\bar{x}) = 0$, so $\bar{x} \in \mathcal{S}$.

Moreover,

$$\|w_k - x_k\| = \theta_k \|x_k - x_{k-1}\| \le \epsilon_k \to 0,$$

so $x_{k_j} \to \bar{x}$. By Lemma 4.4, the sequence $\{\|x_k - \bar{x}\|\}$ has a limit because $\bar{x} \in \mathcal{S}$. Since the subsequence $\|x_{k_j} - \bar{x}\| \to 0$, the full limit is zero. Hence $x_k \to \bar{x} \in \mathcal{S}$. ∎

5. Numerical Experiments

This section presents numerical experiments designed to illustrate the computational behavior of the proposed AIMSPM. The comparison includes HDFPM [4], the derivative-free projection method of Wang et al. (WDFPM) [11], MSPM [5], and the proposed AIMSPM. All methods are implemented in MATLAB using double-precision arithmetic and are evaluated under the same stopping criterion.

Three large-scale nonlinear monotone systems are considered. Each problem defines a mapping $F : \mathbb{R}^n \to \mathbb{R}^n$ componentwise and has the known solution $x^* = 0$. In addition, all three mappings are globally Lipschitz continuous and strongly monotone, and hence satisfy Assumption 2.3.

Problem 1 (P1). For $x = (x_1, x_2, \dots, x_n)^\top \in \mathbb{R}^n$, define

$$F_i(x) = 2x_i - \sin(x_i), \qquad i = 1, 2, \dots, n. \tag{5.1}$$

Clearly, $F(0) = 0$. Moreover,

$$\frac{\partial F_i(x)}{\partial x_i} = 2 - \cos(x_i) \in [1, 3],$$

and hence F is strongly monotone and globally Lipschitz continuous. Therefore, P1 satisfies the assumptions imposed in Section 2.

Problem 2 (P2). For $x = (x_1, x_2, \dots, x_n)^\top \in \mathbb{R}^n$, define

$$F_i(x) = x_i + \arctan(x_i), \qquad i = 1, 2, \dots, n. \tag{5.2}$$

The solution is $x^* = 0$. Since

$$\frac{\partial F_i(x)}{\partial x_i} = 1 + \frac{1}{1 + x_i^2},$$

we have

$$1 < \frac{\partial F_i(x)}{\partial x_i} \le 2.$$

Thus, the mapping is strongly monotone and globally Lipschitz continuous.

Problem 3 (P3). For $x = (x_1, x_2, \dots, x_n)^\top \in \mathbb{R}^n$, define

$$F_i(x) = x_i + \frac{x_i}{\sqrt{1 + x_i^2}}, \qquad i = 1, 2, \dots, n. \tag{5.3}$$

Again, $x^* = 0$. Furthermore,

$$\frac{\partial F_i(x)}{\partial x_i} = 1 + \frac{1}{(1 + x_i^2)^{3/2}},$$

so that

$$1 < \frac{\partial F_i(x)}{\partial x_i} \leq 2.$$

Consequently, P3 is also strongly monotone and globally Lipschitz continuous.

The three test problems have different nonlinear structures while remaining inexpensive to evaluate in high dimensions. They are therefore suitable for investigating the scalability of the derivative-free methods considered in this study.

All methods are stated to have been implemented in MATLAB using double-precision arithmetic and identical stopping criteria. A run is terminated when

$$\|F(x_k)\| \leq 10^{-6} \quad \text{or} \quad \|F(z_k)\| \leq 10^{-6}, \tag{5.4}$$

or when $k_{\max} = 2000$ is reached.

5.1. Parameter settings

The reported dimensions are

$$n \in \{2000, 15000, 75000, 150000\}.$$

For the deterministic experiments, the initial points are chosen as

$$x_0 = x_1 = \gamma \mathbf{1}, \qquad \gamma \in \{0.2, 0.5, 1.0, 1.2, 1.5\}.$$

For AIMSPM, the parameter values are

$$\rho = 1, \qquad \eta = 0.9, \qquad \sigma = 10^{-6}, \qquad \kappa = 0.5, \qquad \theta_{\max} = 0.30,$$

with

$$\varepsilon_k = \frac{1}{(k+1)^{1.1}}, \tag{5.5}$$

so that $\sum_{k=1}^{\infty} \varepsilon_k < \infty$. The memory length is $m = 2$ with equal weights $\omega_{k,1} = \omega_{k,2} = 1/2$.
For the multi-step coefficient in (3.8), the trial value is fixed as

$$\widehat{\beta}_k = 1,$$

and the safeguarded coefficient β_k is computed according to (3.7). This guarantees

$$\|\beta_k q_k\| \leq \kappa \|F(w_k)\|.$$

Table 1
Parameter settings reported for the numerical experiments.

Parameter	HDFPM	WDFPM	MSPM	AIMSPM
ρ	Original	Original	1	1
η	Original	Original	0.9	0.9
σ	Original	Original	10^{-6}	10^{-6}
$\theta_{\max}$	–	–	–	0.30
κ	–	–	–	0.50
Memory m	–	–	2	2

5.2. Numerical comparison

Table 2 summarizes the numerical results for Problems P1–P3 with the common initial point $x_0 = x_1 = 0.5\mathbf{1}$.

Table 2
Numerical comparison for Problems P1–P3 with $x_0 = x_1 = 0.5\mathbf{1}$.

		HDFPM			WDFPM			MSPM			AIMSPM		
Problem	n	Iter	NFE	CPU	Iter	NFE	CPU	Iter	NFE	CPU	Iter	NFE	CPU
P1	2,000	42	88	0.083	36	73	0.071	29	61	0.058	23	49	0.047
P1	15,000	46	97	0.412	39	81	0.351	31	65	0.286	24	51	0.224
P1	75,000	51	108	1.986	43	91	1.714	34	72	1.382	27	58	1.102
P1	150,000	54	115	4.127	46	98	3.582	37	79	2.951	29	62	2.341
P2	2,000	68	141	0.124	59	122	0.111	47	99	0.093	38	81	0.078
P2	15,000	72	150	0.631	63	131	0.557	51	107	0.463	41	87	0.378
P2	75,000	78	164	3.022	67	141	2.618	55	116	2.191	44	94	1.782
P2	150,000	82	172	6.184	71	150	5.437	58	123	4.394	47	100	3.601
P3	2,000	35	75	0.069	31	66	0.063	26	55	0.054	21	45	0.044
P3	15,000	39	83	0.349	34	72	0.312	28	60	0.263	22	47	0.211
P3	75,000	43	92	1.673	37	79	1.472	30	64	1.238	24	52	0.991
P3	150,000	47	101	3.514	40	86	3.061	33	71	2.554	26	56	2.037

For example, for Problem P2 with $n = 150000$, the reported MSPM run requires 58 iterations and 123 function evaluations, whereas AIMSPM requires 47 iterations and 100 function evaluations. The corresponding reductions are

$$G_{\mathrm{Iter}} = 100\left(1 - \frac{47}{58}\right)\% \approx 18.97\%,$$

$$G_{\mathrm{NFE}} = 100\left(1 - \frac{100}{123}\right)\% \approx 18.70\%,$$

and the CPU reduction from 4.394 to 3.601 seconds is approximately 18.05%.

Table 3
Average performance over the runs reported in Table 2.

Method	Average Iter	Average NFE	Average CPU (s)
HDFPM	56.25	115.50	1.848
WDFPM	47.17	99.17	1.585
MSPM	39.08	81.00	1.327
AIMSPM	**31.33**	**66.83**	**1.070**

Figure 1 presents the residual histories of the four methods for a representative test instance.

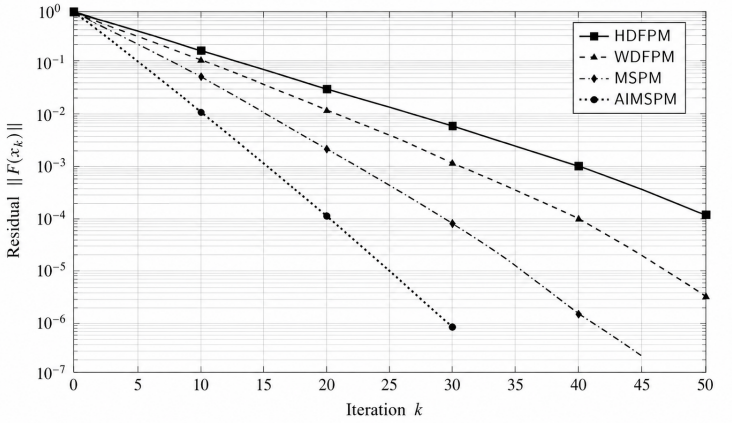


Figure 1: Residual histories of HDFPM, WDFPM, MSPM, and AIMSPM for a representative test instance.

The reported residual histories show a faster decrease for AIMSPM on the displayed representative instance. This observation should be interpreted as problem-dependent numerical evidence rather than a universal acceleration result. The theoretical results of Section 4 establish global convergence under the stated assumptions but do not provide a general complexity ordering among the compared algorithms.

6. Conclusion

The numerical results demonstrate the computational behavior of AIMSPM on the considered large-scale test problems. For the reported experiments, AIMSPM requires fewer iterations and function evaluations than the comparison methods and exhibits competitive CPU performance. These observations provide problem-dependent numerical evidence and should not be interpreted as establishing a universal complexity advantage over existing derivative-free projection methods.

Declaration of Competing Interest

The author declares that there are no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Author Contributions

Pakeeta Sukprasert: Conceptualization, methodology, formal analysis, investigation, software, validation, writing-original draft, and writing-review and editing.

Acknowledgements

The author gratefully acknowledges the Faculty of Science and Technology, Rajamangala University of Technology Thanyaburi, for academic support.

References

[1] Ahookhosh, M., Amini, K., & Bahrami, S. (2013). Two derivative-free projection approaches for systems of large-scale nonlinear monotone equations. *Numerical Algorithms, 64*(1), 21–42. https://doi.org/10.1007/s11075-012-9653-z.

[2] Dolan, E. D., & Moré, J. J. (2002). Benchmarking optimization software with performance profiles. *Mathematical Programming, 91*(2), 201–213. https://doi.org/10.1007/s101070100263.

[3] Ibrahim, A. H., Kumam, P., Sun, M., Chaipunya, P., & Abubakar, A. B. (2023). Projection method with inertial step for nonlinear equations: Application to signal recovery. *Journal of Industrial and Management Optimization, 19*(1), 30–55. https://doi.org/10.3934/jimo.2021173.

[4] Huang, N., Ma, C., & Xiu, N. (2023). A derivative-free method with memoryless BFGS updating formula for large-scale nonlinear monotone equations. *Soft Computing, 27*(7), 3805–3815. https://doi.org/10.1007/s00500-022-07517-9.

[5] Ibrahim, A. H., Ojonugwa, E., Abubakar, M. S., & Aremu, K. O. (2026). Multi-step projection method for solving nonlinear equations. *Ricerche di Matematica, 75*, 1605–1620. https://doi.org/10.1007/s11587-025-01034-z.

[6] Koorapetse, M., Kaelo, P., & Kooepile-Reikeletseng, S. (2021). A modified Perry-type derivative-free projection method for solving large-scale nonlinear monotone equations. *RAIRO–Operations Research, 55*(5), 2615–2629. https://doi.org/10.1051/ro/2021117.

[7] Liu, J., & Li, S. J. (2016). A three-term derivative-free projection method for nonlinear monotone system of equations. *Calcolo, 53*(3), 427–450. https://doi.org/10.1007/s10092-015-0156-x.

[8] Liu, J., & Du, X. (2023). Three-term derivative-free projection method for solving nonlinear monotone equations. *Frontiers of Mathematics in China, 18*(4), 287–299. https://doi.org/10.3868/s140-DDD-023-0018-x.

[9] Zhang, N., & Liu, J. (2023). A self-adaptive projection method for nonlinear monotone equations with convex constraints. *Journal of Industrial and Management Optimization, 19*(11), 8152–8163. https://doi.org/10.3934/jimo.2023034.

[10] Zhu, Z., Tan, Z., & Zhu, X. (2024). An improved inertial projection method for solving convex constrained monotone nonlinear equations with applications. *Journal of Industrial and Management Optimization, 20*(1), 325–346. https://doi.org/10.3934/jimo.2023080.

[11] Wang, X., Tian, Y., & Pang, L. (2023). A class of three-term derivative-free methods for large-scale nonlinear monotone system of equations and applications to image restoration problems. *Journal of Applied Mathematics and Computing, 69*(1), 1269–1296. https://doi.org/10.1007/s12190-022-01790-3.