AIMS Mathematics

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

AIMS Mathematics, 10(2): 3251–3268.
DOI: 10.3934/math.2025151
Received: 30 September 2024
Revised: 14 December 2024
Accepted: 02 January 2025
Published: 20 February 2025

Research article

A modulus-based modified multivariate spectral gradient projection method for solving the horizontal linear complementarity problem

Ting Lin[1], **Hong Zhang**[2] **and Chaofan Xie**[1,*]

[1] School of Big Data and Artificial Intelligence, Fujian Polytechnic Normal University, Fuqing 350300, China

[2] Key Laboratory of Nondestructive Testing, Fujian Province University, Fuqing 350300, China

* **Correspondence:** Email: chaofanxie@hotmail.com; Tel: +15959165013.

Abstract: In this research, we investigated the nonnegative cone horizontal linear complementarity problem (HLCP). Initially, we transformed the HLCP into a fixed-point equation using the modulus defined within the nonnegative cone. By redefining this fixed-point equation as a monotone system, we introduced an improved multivariate spectral gradient projection method for solving it. This study shows that the proposed iterative method converges to the solution of the HLCP, assuming the specified conditions are met. Additionally, numerical examples were included to illustrate the practicality and effectiveness of the modulus-based enhanced multivariate spectral gradient projection method in computational settings.

Keywords: horizontal linear complementarity problem; modulus-based algorithm; multivariate spectral gradient projection; nonnegative cone; monotone system
Mathematics Subject Classification: 65F45, 65G99, 65H20, 90C30

1. Introduction

In this study, we focus on the nonnegative cone horizontal linear complementarity problem (HLCP): Given the matrices $A, B \in \mathbb{R}^{n\times n}$ and a source term $q \in \mathbb{R}^n$, our goal is to find a pair of vectors, denoted as x, y, that meet the following criteria:

$$Ax - By = q, \quad x \in \mathbb{R}^n_+, \quad y \in \mathbb{R}^n_+, \quad \langle x, y\rangle = 0, \tag{1.1}$$

where $\mathbb{R}^n_+ = \{x \in \mathbb{R}^n \mid x \geqslant 0\}$ is the so-called nonnegative cone. Here, $\langle \cdot, \cdot\rangle$ is the Euclidean inner product and $\|\cdot\|$ denotes the Euclidean norm. Obviously, if either $A = I_n$ or $B = I_n$, the HLCP simplifies to the linear complementarity problem (LCP).

The HLCP has many applications in noncooperative games, optimization problems, economic equilibrium problems, and traffic assignment problems [1]. The finite-difference discretization of hydrodynamic lubrication also gives rise to the HLCP [2]. Many scholars have studied this problem. Various methods can be employed to address this issue, including interior point approaches [3], neural network techniques [4], and verification techniques [5]. In [6], Mezzadri and Galligani introduced a class of projected splitting methods.

Recently, the modulus-based matrix-splitting approach, a widely used technique, has been applied to many kinds of linear/nonlinear complementarity problems. For example, Mezzadri and Galligani in 2019 presented the modulus-based matrix-splitting method to solve the HLCP [7]. Furthermore, the convergence of the methods proposed in [7] has been enlarged by Zheng and Vong [8]. Zhang et al. [9] have introduced a two-step parallel iterative approach for solving large sparse HLCP problems.

The first-order methods have been the subject of considerable attention in recent years, largely due to their minimal storage requirements. One of the most widely known first-order methods is the spectral gradient method, which was first introduced by Barzilai and Borwein in reference [10] and subsequently extended by Raydan and La Cruz in references [11–13]. In their study, Zhang and Zhou [14] employed the spectral gradient projection method for unconstrained monotone equations, as proposed by Solodov and Svaiter [15]. This approach is based on the inexact Newton method. Furthermore, Yu et al. demonstrated the efficacy of this approach in solving constrained monotone equations, as evidenced by the favorable outcomes reported in reference [16]. In a recent study, Ibrahim et al. proposed the projection method with inertial step [17] for nonlinear equations, Li et al. studied the modified spectral gradient projection-based algorithm [18] for large-scale constrained nonlinear equations, and Ibrahim et al. proposed the two-step inertial derivative-free projection method [19] for solving nonlinear equations.

Jiang [20] developed a derivative-free descent technique for the nonlinear complementarity problem (NCP) when the nonlinear mapping is directionally differentiable and strongly monotone, utilizing an equivalent system of nonlinear equations derived from the squared Fischer-Burmeister (FB) function. Mangasarian et al. [21] introduced another derivative-free descent method for the strongly monotone NCP by minimizing the implicit Lagrangian function, establishing its global convergence. Ma et al. [22] proposed a smooth Broyden-like method for the NCP, which uses a smooth approximation of the FB function and a derivative-free line search rule, demonstrating global convergence under suitable conditions. This smooth Broyden-like method has also been shown to achieve global and superlinear convergence under appropriate conditions.

Yu et al. [23] converted the absolute value equation into a monotone system and resolved it using a multivariate spectral gradient method. Drawing inspiration from this work, we first reframe the HLCP (1.1) into a fixed-point equation based on the modulus defined within the nonnegative cone. By transforming the implicit fixed-point equation into a monotone system, we introduce a new class of modified multivariate spectral gradient projection methods for its solution. We outline the conditions under which the new equation remains monotone and continuous. It is shown that the proposed iterative method converges to the solution of the HLCP in equation (1.1) under the specified assumptions. Additionally, numerical examples illustrate that the modulus-based modified multivariate spectral gradient projection method is both feasible and efficient.

The proposed algorithm is contrasted with the modulus-based matrix-splitting iterative method, which generally requires matrix inversion, a step not needed here. Since matrix inversion is computationally intensive, the algorithm described in this paper holds a distinct advantage. Compared to conventional gradient descent methods, there is potential for improving the line search technique and the distribution strategy of the spectral gradient in existing algorithms. This paper's algorithm has been refined by incorporating a new line search technique [24] and a more efficient allocation of the spectral gradient, leading to better performance in reducing the number of iterations and CPU time.

The paper is structured as follows: In Section 2, we develop a monotone system of equations that is equivalent to the HLCP. In Section 3, we introduce a modified multivariate spectral gradient projection method grounded in the modulus approach. Section 4 focuses on the convergence analysis of the proposed algorithm. Numerical experiments and their outcomes are detailed and analyzed in Section 5. Finally, concluding observations are provided in Section 6.

2. The establishment of equivalent equations

This section introduces some notations and auxiliary results.

Theorem 2.1. *Let matrices $A, B \in \mathbb{R}^{n\times n}$ and vector $q \in \mathbb{R}^n$. Then,*

- *if (x, y) is a solution to the* HLCP *in* (1.1)*, then $z = \frac{1}{2}(x - y)$ fulfills*

$$(A + B)z + (A - B)|z| = q; \tag{2.1}$$

- *if z satisfies* (2.1)*, then*

$$x = |z| + z, \quad y = |z| - z \tag{2.2}$$

is a solution of the HLCP*, where $z \in \mathbb{R}^n$. $|z|$ denotes the absolute value of z.*

Proof. With regard to the first statement, since (x, y) is a solution of the HLCP in (1.1), it follows that both variables are nonnegative and can be expressed as Eq (2.2) with $z \in \mathbb{R}^n$. If we put (2.2) into (1.1), we obtain (x, y) as a solution to the complementarity problem if and only if

$$A(z + |z|) + B(z - |z|) = q. \tag{2.3}$$

Rearranging the above equation can easily be written as (2.1).

About the second statement, we commence with (2.1), which can be rewritten as Eq (2.3) through the simple rearrangement of terms. By defining x and y as specified in Eq (2.2), we derive

$$Ax - By = q,$$

where x, y is nonnegative. It can be readily confirmed that $x_i > 0$ and $y_i = 0$ when $z_i > 0$, whereas $x_i = 0$ and $y_i > 0$ when $z_i < 0$. If $z_i = 0$, then trivially $x_i = y_i = 0$. Here, z_i denotes the i-th component of z, with a similar notation used for x_i and y_i. Consequently, the complementarity condition is met, and (x, y) constitutes a solution to the HLCP as stated in Eq (1.1). This completes the proof. □

Let $x = |z| + z$, $y = |z| - z$, where $z \in \mathbb{R}^n$. According to Theorem 2.1, the HLCP in (1.1) is equivalent to the fixed-point equation

$$(B + A)z - (B - A)|z| = q.$$

If $B - A$ is invertible, we have

$$(B - A)^{-1}(B + A)z - |z| = (B - A)^{-1}q.$$

We set

$$F(z) := (B-A)^{-1}(B+A)z - |z| - (B-A)^{-1}q, \tag{2.4}$$

where $z \in \mathbb{R}^n$.

Definition 2.1. *A mapping $F : \mathbb{R}^n \to \mathbb{R}^n$ is defined as monotone if, for $\forall \alpha, \beta \in \mathbb{R}^n$,*

$$(\alpha - \beta)^T (F(\alpha) - F(\beta)) \geqslant 0$$

is satisfied.

Definition 2.2. *A mapping $F : \mathbb{R}^n \to \mathbb{R}^n$ is considered Lipschitz continuous if, for $\forall \alpha, \beta \in \mathbb{R}^n$, there exists a constant $L > 0$ such that*

$$\|F(\alpha) - F(\beta)\| \leqslant L\|\alpha - \beta\|.$$

Assumption 2.1. *In Eq (2.4), $(B-A)$ is invertible and $(B-A)^{-1}A$ is semi-positive definite.*

Theorem 2.2. *$F(z)$ is monotone as long as $(B-A)^{-1}A$ is semi-positive definite.*

Proof. For any $\alpha, \beta \in \mathbb{R}^n$, the dot product $\alpha^T\beta = \sum_{i=1}^{n} \alpha_i\beta_i$. Specifically, we have defined the index set

$$I_1 = \{i \in R \mid \alpha_i \geqslant 0, \beta_i \geqslant 0\}, \quad I_2 = \{i \in R \mid \alpha_i \geqslant 0, \beta_i \leqslant 0\},$$

and

$$I_3 = \{i \in R \mid \alpha_i \leqslant 0, \beta_i \geqslant 0\}, \quad I_4 = \{i \in R \mid \alpha_i \leqslant 0, \beta_i \leqslant 0\}.$$

On one side,

$$\begin{aligned}
\langle |\alpha| - |\beta|, \alpha - \beta \rangle &= \sum_{i \in I_1} \langle |\alpha_i| - |\beta_i|, \alpha_i - \beta_i \rangle + \sum_{i \in I_2} \langle |\alpha_i| - |\beta_i|, \alpha_i - \beta_i \rangle \\
&\quad + \sum_{i \in I_3} \langle |\alpha_i| - |\beta_i|, \alpha_i - \beta_i \rangle + \sum_{i \in I_4} \langle |\alpha_i| - |\beta_i|, \alpha_i - \beta_i \rangle \\
&= \sum_{i \in I_1} \langle \alpha_i - \beta_i, \alpha_i - \beta_i \rangle + \sum_{i \in I_2} \langle \alpha_i - \beta_i + 2\beta_i, \alpha_i - \beta_i \rangle \\
&\quad + \sum_{i \in I_3} \langle \beta_i - \alpha_i + 2\alpha_i, \beta_i - \alpha_i \rangle + \sum_{i \in I_4} \langle -\alpha_i + \beta_i, \alpha_i - \beta_i \rangle,
\end{aligned}$$

and expanding it out, we notice that $\sum_{i \in I_2} \langle 2\beta_i, \alpha_i - \beta_i \rangle \leqslant 0$, and $\sum_{i \in I_3} \langle 2\alpha_i, \beta_i - \alpha_i \rangle \leqslant 0$. Hence,

$$\begin{aligned}
\langle |\alpha| - |\beta|, \alpha - \beta \rangle &\leqslant \sum_{i \in I_1} \langle \alpha_i - \beta_i, \alpha_i - \beta_i \rangle + \sum_{i \in I_2} \langle \alpha_i - \beta_i, \alpha_i - \beta_i \rangle \\
&\quad + \sum_{i \in I_3} \langle \alpha_i - \beta_i, \alpha_i - \beta_i \rangle + \sum_{i \in I_4} \langle -\alpha_i + \beta_i, \alpha_i - \beta_i \rangle \\
&= \sum_{i \in I_1} \langle \alpha_i - \beta_i, \alpha_i - \beta_i \rangle + \sum_{i \in I_2} \langle \alpha_i - \beta_i, \alpha_i - \beta_i \rangle \\
&\quad + \sum_{i \in I_3} \langle \alpha_i - \beta_i, \alpha_i - \beta_i \rangle - \sum_{i \in I_4} \langle \alpha_i - \beta_i, \alpha_i - \beta_i \rangle
\end{aligned}$$

$$
\begin{aligned}
&\leqslant \sum_{i\in I_1}\langle \alpha_i-\beta_i,\alpha_i-\beta_i\rangle+\sum_{i\in I_2}\langle \alpha_i-\beta_i,\alpha_i-\beta_i\rangle\\
&\quad+\sum_{i\in I_3}\langle \alpha_i-\beta_i,\alpha_i-\beta_i\rangle+\sum_{i\in I_4}\langle \alpha_i-\beta_i,\alpha_i-\beta_i\rangle\\
&=\sum_{i\in I_1\cup I_2\cup I_3\cup I_4}\langle \alpha_i-\beta_i,\alpha_i-\beta_i\rangle\\
&=(\alpha-\beta)^T(\alpha-\beta).
\end{aligned}
$$

On the flip side,

$$
\begin{aligned}
\langle F(\alpha)-F(\beta),\alpha-\beta\rangle &= \langle (B-A)^{-1}(B+A)\alpha-|\alpha|-(B-A)^{-1}q-(B-A)^{-1}(B+A)\beta\\
&\quad+|\beta|+(B-A)^{-1}q,\alpha-\beta\rangle\\
&=\langle (B-A)^{-1}(B+A)\alpha-(B-A)^{-1}(B+A)\beta-|\alpha|+|\beta|,\alpha-\beta\rangle\\
&=\langle (B-A)^{-1}(B+A)(\alpha-\beta)-(|\alpha|-|\beta|),\alpha-\beta\rangle\\
&=(\alpha-\beta)^T(B-A)^{-1}(B+A)(\alpha-\beta)-\langle(|\alpha|-|\beta|),\alpha-\beta\rangle\\
&\geqslant(\alpha-\beta)^T(B-A)^{-1}(B+A)(\alpha-\beta)-(\alpha-\beta)^T(\alpha-\beta)\\
&\geqslant(\alpha-\beta)^T((B-A)^{-1}(B+A)-I)(\alpha-\beta)\\
&=(\alpha-\beta)^T((B-A)^{-1}(B+A)-(B-A)^{-1}(B-A))(\alpha-\beta)\\
&=2(\alpha-\beta)^T((B-A)^{-1}A)(\alpha-\beta).
\end{aligned}
$$

Obviously, $F(z)$ will be monotone, as long as $(B-A)^{-1}A$ is semi-positive definite. This completes the proof. □

Theorem 2.3. *The function $F(z)$ is Lipschitz continuous, meaning there exists a nonnegative constant L such that the following inequality is satisfied:*

$$
\|F(\alpha)-F(\beta)\|\leqslant L\|\alpha-\beta\|.
$$

Proof. For $\forall\alpha,\beta\in\mathbb{R}^n$, by the Lipschitz condition and (2.4), we have

$$
\begin{aligned}
\|F(\alpha)-F(\beta)\|&=\|(B-A)^{-1}(B+A)\alpha-|\alpha|-(B-A)^{-1}(B+A)\beta+|\beta|\|\\
&=\|(B-A)^{-1}(B+A)(\alpha-\beta)-(|\alpha|-|\beta|)\|\\
&\leqslant\|(B-A)^{-1}(B+A)(\alpha-\beta)\|+\||\alpha|-|\beta|\|\\
&\leqslant\|(B-A)^{-1}(B+A)\|\|\alpha-\beta\|+\|\alpha-\beta\|\\
&=(\|(B-A)^{-1}(B+A)\|+1)\|\alpha-\beta\|.
\end{aligned}
$$

Since $\|(B-A)^{-1}(B+A)\|\geqslant 0$, thus, set $L=\|(B-A)^{-1}(B+A)\|+1$, and it can be demonstrated that the Lipschitz condition is satisfied. This completes the proof. □

According to Theorems 2.2 and 2.3, under the condition of Assumption 2.1, HLCP (1.1) can be equivalently reformulated to a monotone system:

$$
F(z)=0. \tag{2.5}
$$

3. Algorithm

For monotone system (2.5), the matrix-splitting method was commonly used in the past. In each iteration, the fixed-point equation needs to be inverted to obtain a new iteration point. As we know, it takes a certain amount of time to find the inverse. In order to simplify the time cost, we introduce a new multivariate spectral gradient method to solve (2.5), called the modulus-based modified multivariate spectral gradient projection method (M-MMSGP). The M-MMSGP algorithm employs a novel line search methodology [24] that differs from existing gradient algorithms. It utilizes the multivariate spectral gradient vector as the descent direction and assigns the multivariate spectral gradient as a whole, which significantly reduces the number of iterations of the algorithm and CPU time.

Algorithm 3.1. M-MMSGP algorithm for (2.5)

Step 0: We are given $\varepsilon > 0$, $z \in \mathbb{R}^n$. Let $\sigma, \beta \in (0,1)$, $\alpha_0 = 1$, $r, \alpha_{min} \in [0,1]$. Set $k := 0$.

Step 1: If $\left\|F(z^k)\right\| \leqslant \varepsilon$, then halt and consider z^k as an approximate solution.

Step 2: Calculate search direction

$$d^k = -\alpha_k. * F(z^k). \tag{3.1}$$

Step 3: Find $v^k = z^k + \theta_k d^k$, where $\theta_k £ = \beta^{m_k}$, and m_k is the smallest nonnegative integer such that

$$-F(z^k + \theta_k d^k)^T d^k \geqslant \sigma\gamma_k\theta_k \|d^k\|^2, \tag{3.2}$$

where $\gamma_k £ = \dfrac{\|F(z^k)\|}{1 + \|F(z^k)\|}$.

Step 4: Compute the new iterate by

$$z^{k+1} = z^k - \frac{\left\langle F(v^k), z^k - v^k \right\rangle}{\left\|F(v^k)\right\|^2} F(v^k). \tag{3.3}$$

Step 5: Update the spectral vector by

$$\alpha_{k+1} = \max\{\alpha_{\min} ones(n,1), \frac{(s^k)^T y^k}{(s^k)^T s^k} ones(n,1)\}, \tag{3.4}$$

where $s^k = z^{k+1} - z^k$ and $y^k = F(z^{k+1}) - F(z^k) + rs^k$.

Step 6: Increase k by 1 and return to Step 1.

Remark 3.1. *If Algorithm 3.1 terminates at finite iteration k and $F(z^k) = 0$, then z^k is the solution to $F(z) = 0$. If k approaches infinity and $F(z^k) \neq 0$, Algorithm 3.1 produces an infinite sequence $\{z^k\}$. For simplicity, we can assume that $\{z^k\}$ is an infinite sequence.*

Remark 3.2. (1) *Based on the monotonic behavior of $F(z)$, we can derive*

$$(s^k)^T y^k = (z^{k+1} - z^k)^T (F(z^{k+1}) - F(z^k) + rs^k) \geqslant r((s^k)^T s^k). \tag{3.5}$$

(2) *From the Lipschitz continuity of $F(z)$, we can deduce that*

$$(s^k)^T y^k \leqslant (L + r)((s^k)^T s^k). \tag{3.6}$$

(3) *From* (3.4)–(3.6), *we can deduce that*

$$min\{\alpha_{min}, r\}\|F(z^k)\|^2 \leqslant \|d^k\|^2 \leqslant max\{\alpha_{min}, L + r\}\|F(z^k)\|^2. \tag{3.7}$$

Here, $L = \|(B - A)^{-1}(B + A)\| + 1$ is given by Theorem 2.3.

The subsequent lemma illustrates that Algorithm 3.1 is well-defined.

Lemma 3.1. *According to the propositions outlined above, there exists a nonnegative integer m_k ensuring that* (3.2) *is satisfied for all $k \geqslant 0$.*

Proof. Assume there is a $k_0 \geqslant 0$ for which (3.2) fails to hold for any nonnegative integer m, i.e.,

$$\langle F(z^{k_0} + \beta^m d^{k_0}), d^{k_0} \rangle < \sigma \frac{\|F(z^{k_0})\|}{1 + \|F(z^{k_0})\|} \beta^m \|d^{k_0}\|^2.$$

By letting $m \to \infty$ and utilizing the continuity of $F(z)$, we obtain

$$-\langle F(z^{k_0}), d^{k_0} \rangle \leqslant 0. \tag{3.8}$$

Based on Step 1 of Algorithm 3.1 and (3.1), we find that

$$F(z^k) \neq 0, \ d^k \neq 0,$$

for any $k \geqslant 0$. Consequently, it follows that

$$-\langle F(z^0), d^0 \rangle = \langle F(z^0), F(z^0) \rangle > 0,$$

and

$$-\langle F(z^k), d^k \rangle = \langle F(z^k), \alpha_k F(z^k) \rangle \geqslant min\{\alpha_{min}, r\} \|F(z^k)\|^2 > 0, \tag{3.9}$$

where $k \geqslant 0$. This result conflicts with (3.8), thus completing the proof. □

4. Convergence analysis

In this part, we demonstrate that every convergent subsequence $\{z^{k_j}\}$ of the sequence $\{z^k\}$ produced by Algorithm 3.1 approaches a solution of system (2.5). Due to the equivalence, it also converges to a solution of the HLCP (1.1).

Lemma 4.1. *[15] Let $F(z)$ be monotone, and $z, v \in \mathbb{R}^n$ such that $F(v)^T(z - v) > 0$. Let z^* be a solution of $F(z) = 0$ and*

$$z^+ = z - \frac{F(v)^T(z - v)}{\|F(v)\|^2} F(v).$$

Then we have

$$\|z^+ - z^*\|^2 \leqslant \|z - z^*\|^2 - \|z^+ - z\|^2.$$

Theorem 4.1. *Given Assumption 2.1, assume the sequence $\{z^k\}$ is generated by the M-MMSGP method with $\varepsilon = 0$. In this case, either the sequence $\{z^k\}$ is finite, and the final iterate $\{z^k\}$ is a solution to $F(z) = 0$, or the sequence $\{z^k\}$ is infinite, and*

$$\lim_{k \to \infty} \theta_k \|d^k\| = 0.$$

Proof. By Assumption 2.1 and Theorem 2.2, $F(z)$ in (2.4) is monotone. By (3.2) of Algorithm 3.1, we have

$$F(v^k)^T(z^k - v^k) = -\beta^{m_k}F(v^k)^T d^k \geqslant \sigma\gamma_k\beta^{m_k}\|d^k\|^2 > 0.$$

By Lemma 4.1, one has

$$\|z^{k+1} - z^*\|^2 \leqslant \|z^k - z^*\|^2 - \|z^{k+1} - z^k\|^2, \tag{4.1}$$

where z^* solves $F(z) = 0$. Hence, $\{\|z^k - z^*\|\}$ is a decreasing sequence and $\|z^k - z^*\| \leqslant \|z^0 - z^*\|$ holds for all $k \geqslant 1$ which implies that $\{z^k\} \subset \{z \in \mathbb{R}^n : \|z - z^*\| \leqslant \|z^0 - z^*\|\}$. Hence, $\{z^k\}$ is a bounded sequence.

Given the continuity of $F(z)$, the sequence $\{\|F(z^k)\|\}$ is bounded.

$$\|d^k\|^2 = \|\alpha_k\|^2\|F(z^k)\|^2 \leqslant max\{\alpha_{min}, L + r\}\|F(z^k)\|^2. \tag{4.2}$$

By (4.2), $\{d^k\}$ is also bounded. From Step 3 of Algorithm 3.1, we have $v^k = z^k + \theta_k d^k$. Since $\{z^k\}$ and $\{d^k\}$ are bounded, $\theta_k \in (0, 1)$, so $\{v^k\}$ is bounded. Given the continuity of $F(v)$, the sequence $\{\|F(v^k)\|\}$ is bounded. There exists a positive constant M such that $\|F(v^k)\| \leqslant M$ for all $k \geqslant 1$.

Assuming without loss of generality that the sequence $\{z^k\}$ is infinite, from (4.1) we obtain

$$\lim_{k\to\infty} \|z^{k+1} - z^k\| = 0. \tag{4.3}$$

By (3.3) we have

$$\begin{aligned}\|z^{k+1} - z^k\| &= \frac{|F(v^k)^T(z^k - v^k)|}{\|F(v^k)\|^2}\|F(v^k)\| = \frac{|F(v^k)^T\theta_k d^k|}{\|F(v^k)\|^2}\|F(v^k)\| \\ &\geqslant \frac{\sigma(\theta_k)^2\gamma_k\|d^k\|^2}{\|F(v^k)\|^2}\|F(v^k)\| = \frac{\sigma}{1 + \|F(v^k)\|^2}(\theta_k)^2\|d^k\|^2 \\ &\geqslant \frac{\sigma}{1 + M}(\theta_k)^2\|d^k\|^2.\end{aligned}$$

Combining the above inequality and (4.3) yields $\lim_{k\to\infty} \theta_k\|d^k\| = 0$. This concludes the proof. □

The subsequent theorem illustrates the global convergence of the proposed M-MMSGP method.

Theorem 4.2. *Under the condition of Assumption 2.1, suppose that sequence $\{z^k\}$ is generated by the proposed M-MMSGP method with $\varepsilon = 0$. Then, $\{z^k\}$ converges to some point z^* satisfying $F(z^*) = 0$.*

Proof. By Theorem 4.1, we get

$$\lim_{k\to\infty} \theta_k\|d^k\| = 0. \tag{4.4}$$

(1) If $\liminf_{k\to\infty} \|d^k\| = 0$, then from (3.7), it follows that $\liminf_{k\to\infty} \|F(z^k)\| = 0$. Consequently, there exists a subsequence $\{z^{k_j}\} \subset \{z^k\}$ such that $\lim_{k_j\to\infty} \|F(z^{k_j})\| = 0$. Given the continuity of $F(z)$, this implies $\lim_{k_j\to\infty} \|z^{k_j} - z^*\| = 0$, where z^* is a point satisfying $F(z^*) = 0$.

(2) If $\liminf_{k\to\infty} \|d^k\| > 0$, then according to Theorem 4.1, $\lim_{k\to\infty} \theta_k = 0$. From Eq (4.4), it follows that $\lim_{k\to\infty} \|F(z^k)\| > 0$.

Based on Algorithm 3.1, we have

$$-F(z^k + \beta^{m_k-1}d^k)^T d^k < \sigma\gamma_k\beta^{m_k-1}\|d^k\|^2.$$

Given that the sequences $\{z^k\}$ and $\{d^k\}$ are bounded, they must have at least one cluster point. Therefore, there exist subsequences $\{z^{k_j}\} \subset \{z^k\}$ and $\{d^{k_j}\} \subset \{d^k\}$ that converge to this cluster point.

In the above equation, let $k \to \infty$, and we can get

$$-F(\hat{z})^T \hat{d} \leqslant 0, \tag{4.5}$$

where $\hat{z}$ and $\hat{d}$ are limit points of subsequences $\{z^{k_j}\} \subset \{z^k\}$ and $\{d^{k_j}\} \subset \{d^k\}$, respectively. On the other hand, let $k \to \infty$ in (3.9), and we have

$$-F(\hat{z})^T \hat{d} \geqslant 0. \tag{4.6}$$

This contradicts (4.5). Therefore, inequality $\liminf_{k\to\infty} \|d^k\| > 0$ does not hold. This completes the proof. □

5. Numerical results

In this section, we present some numerical results to demonstrate the efficiency of Algorithm 3.1. We choose a modified gradient projection algorithm (MGP) [25] used to solve (2.5), a modified multivariate spectral gradient algorithm (MMSGP) [23] used to solve (2.5), a modulus-based Jacobi algorithm (MJ) [6] used to solve (2.1), and a modified spectral gradient projection-based algorithm (PGP) [18] used to solve (2.5) as the comparison. All methods were implemented in MATLAB R2018a and executed on a personal computer with an Intel Core i7 processor operating at 1.80 GHz and 8GB of RAM.

All experiment results include three aspects: the elapsed CPU time in seconds (CPU), the norm of absolute residual vectors (RES), and the number of iteration steps (IT), respectively. RES is defined as RES := $\|Ax^{(k)} - By^{(k)} - q\|$.

In the following experiments, when the prescribed iteration number $\text{IT}_{\max} = 600$ is exceeded or the residual vector satisfies RES $\leqslant 10^{-6}$, all runs are terminated. We will consider the problems with six dimensions, i.e., $n = 100, 400, 900, 1600, 2500$, and 3600.

The primary numerical outcomes are presented in Tables 1–5, as well as Figures 1–4, to facilitate easy comparison. In these tables and figures, the algorithm parameters are set as follows:
(1) For the MGP algorithm, set $\lambda_0 = 0.4, \delta = 1.01, \alpha = \frac{0.4}{\delta}$;
(2) For the MMSGP algorithm, set $\alpha_0 = 1, \beta = 0.2, \tau = 0.001, \sigma = 0.01, r = 0.001$;
(3) For the MJ algorithm, set $\Omega = 0.5I_n$;
(4) For the PGP algorithm, set $\beta = 1, \rho = 0.8, \sigma = 10^{-5}, l = 10^{-4}, u = 10^{30}$;
(5) For the M-MMSGP algorithm, set $\lambda_{min} = 0.1ones(n, 1), \beta = 0.618, \sigma = 0.01, r = 0.001$.

Example 5.1. [7] To solve the HLCP, consider the tridiagonal matrix

$$S = \text{tridiag}(-1, 4, -1) \in \mathbb{R}^{m\times m},$$

where m is a given positive integer. We define the matrices $A, B \in \mathbb{R}^{n\times n}$, with $n = m^2$, as $A = \hat{A} + \mu I$ and $B = \hat{B} + \nu I$ with μ, ν as the real parameters and

$$\hat{A} = \text{tridiag}\,(-I_m, S, -I_m) = \begin{pmatrix} S & -I_m & & & \\ -I_m & S & -I_m & & \\ & \ddots & \ddots & \ddots & \\ & & -I_m & S & -I_m \\ & & & -I_m & S \end{pmatrix} \in \mathbb{R}^{n\times n},$$

$$\hat{B} = \text{tridiag}(0, S, 0) \in \mathbb{R}^{n\times n}.$$

Let $z^* = (1, -1, 1, -1, \ldots)^T$, $x^* = |z^*| + z^*$, and $y^* = |z^*| - z^*$, and then $q = Ax^* - By^*$.

Example 5.1 can be derived from the discretization form of a two-dimensional boundary problem, which is specifically described as

$$\Delta z + \frac{\partial^2 w}{\partial^2 u} + \mu z + \nu w - q = 0, z \geqslant 0, w \geqslant 0, z^T w = 0,$$

where $z(u, v), w(u, v)$, and $q(u, v)$ are three two-dimensional maps. Therefore, under appropriate boundary conditions, the boundary problem is discretized by the five-point difference discretization method, and its discretization scheme is in the form of an HLCP.

Example 5.2. [10] To solve the HLCP, consider the tridiagonal matrix

$$S = \begin{pmatrix} 4 & -0.5 & & & \\ -1.5 & 4 & -0.5 & & \\ & \ddots & \ddots & \ddots & \\ & & -1.5 & 4 & -0.5 \\ & & & -1.5 & 4 \end{pmatrix} \in \mathbb{R}^{m\times m},$$

where m is a given positive integer. We define the matrices $A, B \in \mathbb{R}^{n\times n}$, with $n = m^2$, as $A = \hat{A} + \mu I$ and $B = \hat{B} + \nu I$ with μ, ν as the real parameters and

$$\hat{A} = \text{tridiag}\,(-1.5I_m, S, -0.5I_m) = \begin{pmatrix} S & -0.5I_m & & & \\ -1.5I_m & S & -0.5I_m & & \\ & \ddots & \ddots & \ddots & \\ & & -1.5I_m & S & -0.5I_m \\ & & & -1.5I_m & S \end{pmatrix} \in \mathbb{R}^{n\times n},$$

$$\hat{B} = \text{tridiag}(0, S, 0) \in \mathbb{R}^{n\times n}.$$

Let $z^* = (1, -1, 1, -1, \ldots)^T$, $x^* = |z^*| + z^*$, and $y^* = |z^*| - z^*$, and then $q = Ax^* - By^*$.

Example 5.3. To solve the HLCP, where $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times n}$, consider the block-tridiagonal matrices

$$A = \text{tridiag}(-1, 7, -1) \in \mathbb{R}^{n\times n}$$

and

$$\begin{aligned} B &= \text{tridiag}(-2I_m, C, -2I_m) \\ &= \begin{pmatrix} C & -2I_m & & & \\ -2I_m & C & -2I_m & & \\ & \ddots & \ddots & \ddots & \\ & & -2I_m & C & -2I_m \\ & & & -2I_m & C \end{pmatrix} \in \mathbb{R}^{n\times n} \end{aligned}$$

with the tridiagonal matrix

$$C = \text{tridiag}(-1, 7, -1) \in \mathbb{R}^{m\times m}.$$

Here, $n = m^2$. Let $z^* = (1, -1, 1, -1, \ldots)^T$, $x^* = |z^*| + z^*$, and $y^* = |z^*| - z^*$, and then $q = Ax^* - By^*$. This example is adapted from the literature [10].

Example 5.4. To solve the HLCP, where $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times n}$, consider the block-tridiagonal matrices

$$A = \text{tridiag}\,(-0.5I_m, C, -0.5I_m) = \begin{pmatrix} C & -0.5I_m & & & \\ -0.5I_m & C & -0.5I_m & & \\ & \ddots & \ddots & \ddots & \\ & & -0.5I_m & C & -0.5I_m \\ & & & -0.5I_m & C \end{pmatrix} \in \mathbb{R}^{n\times n}$$

and

$$B = \text{tridiag}(-1, 7, -1) \in \mathbb{R}^{n\times n}$$

with the tridiagonal matrix

$$C = \text{tridiag}(-1, 4, -1) \in \mathbb{R}^{m\times m}.$$

Here, $n = m^2$. Let $z^* = (1, -1, 1, -1, \ldots)^T$, $x^* = |z^*| + z^*$, and $y^* = |z^*| - z^*$, and then $q = Ax^* - By^*$. This example is adapted from the literature [10].

Example 5.5. To solve the HLCP, where $A \in \mathbb{R}^{n\times n}$ and A is a random matrix with eigenvalues $\{1, 1, 3, 3, 3, ...\}$, consider

$$B = 1.8I_n \in \mathbb{R}^{n\times n}.$$

Here, $n = m^2$. Let $z^* = (1, -1, 1, -1, \ldots)^T$, $x^* = |z^*| + z^*$, and $y^* = |z^*| - z^*$, and then $q = Ax^* - By^*$.

For Examples 5.1–5.5, the numeric results are list in Tables 1–5, respectively.

Table 1 contains the results of the M-MMSGP method, MJ method, MGP method, and PGP method for Example 5.1 under n = 100, 400, 900, 1600, 2500, and 3600, respectively. Here, we take $\mu = 2$, and $\nu = 3$. Among the four algorithms, it can be seen that the M-MMSGP algorithm has obvious advantages in both the number of iterations and the CPU time. The M-MMSGP algorithm has good numerical performance.

Table 1. Numerical results of Example 5.1.

Algorithms	M-MMSGP	MJ	MGP	PGP
n	IT/CPU/RES	IT/CPU/RES	IT/CPU/RES	IT/CPU/RES
100	20/0.0018/9.05e-7	52/0.0460/8.05e-7	98/0.0029/8.81e-7	29/0.0064/4.08e-7
400	28/0.0040/5.99e-7	61/0.4781/9.73e-7	104/0.0062/8.71e-7	30/0.0127/7.07e-7
900	33/0.0077/8.06e-7	63/3.0354/8.55e-7	107/0.0114/8.69e-7	26/0.0177/2.51e-7
1600	38/0.0158/7.26e-7	64/14.5580/7.66e-7	109/0.0188/8.71e-7	32/0.0236/4.30e-7
2500	24/0.0101/7.31e-7	64/40.8561/8.64e-7	110/0.0285/9.48e-7	28/0.0365/6.55e-7
3600	31/0.0218/7.76e-7	64/113.5324/9.52e-7	111/0.0459/9.83e-7	34/0.0526/4.68e-7

Tables 2 and 3 contain the results of the M-MMSGP method, MJ method, MGP method, and PGP method for Examples 5.2 and 5.3 under n = 100, 400, 900, 1600, 2500, and 3600, respectively. The M-MMSGP algorithm has excellent numerical performance. The numerical results show that the M-MMSGP method excels in both CPU time and iteration count. Among the four algorithms, the MGP algorithm has the most iteration steps but consumes less CPU time. The MJ algorithm has the highest

CPU time but fewer iteration steps. The MJ algorithm requires inversion in each iteration, while the other three algorithms do not require inversion, which requires more CPU time. Therefore, the MJ algorithm has the highest CPU time. The selection of gradient descent direction in the MGP algorithm is not effective, while the other three algorithms have better descent directions, so the MGP algorithm has more iteration steps. The PGP algorithm always spends more CPU time than the M-MMSGP algorithm although the number of iteration steps is less than the M-MMSGP algorithm in certain dimensions. The M-MMSGP algorithm only needs to compare and allocate spectral gradients, which provides better CPU time. Especially for the M-MMSGP algorithm, the larger the matrix dimension, the faster the convergence speed.

Table 2. Numerical results of Example 5.2.

Algorithms	M-MMSGP	MJ	MGP	PGP
n	IT/CPU/RES	IT/CPU/RES	IT/CPU/RES	IT/CPU/RES
100	19/0.0009/8.52e-7	57/0.0116/7.91e-7	98/0.0027/8.45e-7	52/0.0151/6.75e-7
400	21/0.0020/3.94e-7	64/0.4632/8.08e-7	104/0.0060/9.15e-7	33/0.0129/4.19e-7
900	31/0.0066/8.66e-7	65/2.9363/9.31e-7	107/0.0114/9.05e-7	33/0.0222/3.78e-7
1600	38/0.0146/7.56e-7	66/12.7681/8.34e-7	109/0.0190/9.00e-7	25/0.0242/2.62e-7
2500	39/0.0225/8.56e-7	66/43.2360/9.43e-7	110/0.0290/9.74e-7	28/0.0364/6.69e-7
3600	34/0.0251/8.57e-7	67/116.0291/7.95e-7	112/0.0416/8.55e-7	29/0.0459/8.34e-7

Table 3. Numerical results of Example 5.3.

Algorithms	M-MMSGP	MJ	MGP	PGP
n	IT/CPU/RES	IT/CPU/RES	IT/CPU/RES	IT/CPU/RES
100	23/0.0041/2.36e-7	65/0.0160/9.41e-7	127/0.0070/9.23e-7	44/0.0129/8.89e-7
400	25/0.0116/8.48e-7	72/0.5340/8.98e-7	138/0.0072/9.48e-7	30/0.0212/7.73e-7
900	28/0.0207/7.26e-7	75/3.8034/8.08e-7	142/0.0159/9.65e-7	41/0.0306/3.92e-7
1600	30/0.0085/2.98e-7	76/15.1479/9.30e-7	145/0.0240/9.00e-7	46/0.0297/8.93e-7
2500	27/0.0116/9.77e-7	77/52.1398/9.58e-7	147/0.0368/8.77e-7	43/0.0403/7.97e-7
3600	25/0.0154/7.99e-7	78/139.2645/9.28e-7	148/0.0527/9.31e-7	33/0.0525/9.29e-7

Table 4 contains the results of the M-MMSGP method, MJ method, MGP method, and MMSGP method for Example 5.4 under $n = 400$, 900, 1600, 2500, and 3600, respectively. The M-MMSGP algorithm has excellent numerical performance. The numerical results indicate that the M-MMSGP method has the optimal CPU time. The iteration times of the M-MMMGP algorithm are only inferior to the MJ algorithm in certain dimensions. Overall, the M-MMSGP algorithm also has better numerical performance in terms of iteration times. The reason why the MJ algorithm takes more time than the M-MMSGP algorithm is that the MJ algorithm requires inversion in each iteration, which requires a certain amount of CPU time. The M-MMSGP algorithm only needs to compare and allocate spectral gradients, which provides better CPU time.

The experimental results demonstrate that despite having a lower number of iterations, the MMSGP algorithm consumes the most CPU time. Specifically, when $n = 900$, the CPU time surpasses 100 seconds. The reason why MMSGP has fewer iterations is that the algorithm needs to assign values to each element in the gradient direction in order to find the optimal gradient descent direction. However, due to the allocation of each element, more CPU time is required when the dimensionality is high. The M-MMSGP algorithm only requires an overall comparison of the distribution spectral gradient, which occupies less CPU time. Especially for the M-MMSGP algorithm, the larger the matrix dimension, the faster the convergence speed.

Table 4. Numerical results of Example 5.4.

Algorithms	M-MMSGP	MJ	MGP	MMSGP
n	IT/CPU/RES	IT/CPU/RES	IT/CPU/RES	IT/CPU/RES
400	21/0.0047/3.25e-7	21/0.1738/4.40e-7	95/0.0052/9.51e-7	16/9.1435/2.50e-7
900	21/0.0044/2.52e-7	21/0.9589/7.55e-7	98/0.0105/8.55e-7	16/104.1921/3.70e-7
1600	23/0.0065/7.48e-7	22/4.3433/3.31e-7	99/0.0169/9.74e-7	*/*/*
2500	21/0.0086/6.62e-7	22/14.7866/4.28e-7	101/0.0260/8.39e-7	*/*/*
3600	23/0.0129/5.05e-7	22/38.1063/5.25e-7	102/0.0371/8.40e-7	*/*/*

Remark: */*/*: CPU > 100.

From Table 5, it can be seen that the M-MMSGP method is sensitive to solving the HLCP. The numerical calculation results include the M-MMSGP algorithm, MJ algorithm, and MGP algorithm, which have good performance. The M-MMSGP method has the optimal CPU time and iteration times. The experimental data indicates that for the M-MMSGP algorithm, an increase in dimensionality does not significantly affect the number of iterations, suggesting robust numerical performance. Compared to the other two algorithms, the MJ algorithm outperforms the MGP algorithm, while the M-MMSGP algorithm surpasses both.

Table 5. Numerical results of Example 5.5.

Algorithms	M-MMSGP	MJ	MGP
n	IT/CPU/RES	IT/CPU/RES	IT/CPU/RES
100	31/0.0157/7.11e-7	34/0.0196/9.86e-7	55/0.0191/7.77e-7
400	32/0.2418/9.36e-7	36/0.2665/5.69e-7	57/0.5623/9.03e-7
900	30/1.1550/4.80e-7	36/1.6824/8.86e-7	58/3.1461/9.65e-7
1600	29/3.8054/4.62e-7	37/7.1720/6.44e-7	59/10.3077/9.08e-7
2500	32/9.9062/6.22e-7	37/23.6482/8.12e-7	60/25.8178/8.10e-7

Figure 1 shows the relationship between the number of iterations and RES for Example 5.1. It can be seen that the RES descent speed of the M-MMSGP algorithm is faster than that of the MJ algorithm and MGP algorithm. This also verifies the effectiveness of the M-MMSGP algorithm. Figure 2 shows the relationship between the number of iterations and RES for Example 5.3. It can be seen that the RES drop speed of the M-MMSGP algorithm is the fastest among the three algorithms. Figure 3 shows the relationship between the number of iterations and RES for Example 5.4. It can be seen that the RES drop speed of the M-MMSGP algorithm is the fastest among the three algorithms. The relationship between the number of iterations and RES in Example 5.5 is shown in Figure 4. The above proves that the M-MMSGP algorithm has good numerical performance.

The aforementioned numerical results highlight that the M-MMSGP algorithm demonstrates greater efficiency in terms of CPU time and iteration count under specific conditions. Therefore, our proposed method could be well-suited for solving the HLCP.

The M-MMSGP algorithm employs a novel linear search method, which is distinct from existing gradient algorithms. It utilizes multivariate spectral gradient vectors as the descent direction and allocates the multivariate spectral gradients as a whole, resulting in a significant increase in the number of iterations and CPU time required by the algorithm.

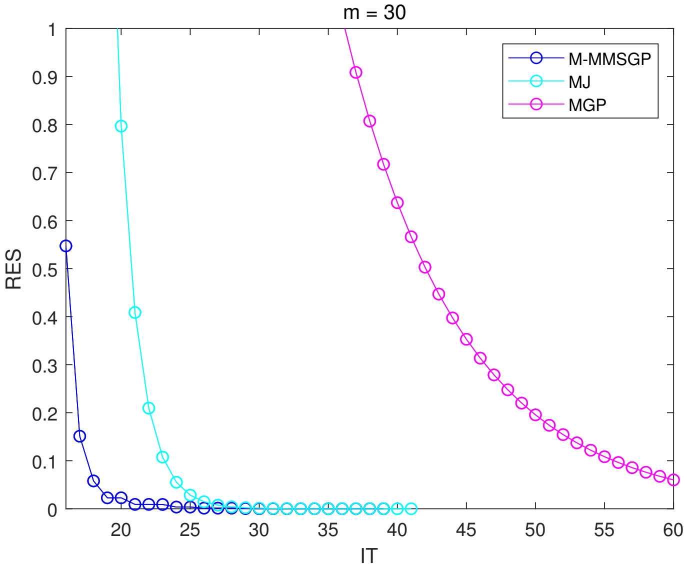

Figure 1. Relationship between IT and RES for Example 5.1.

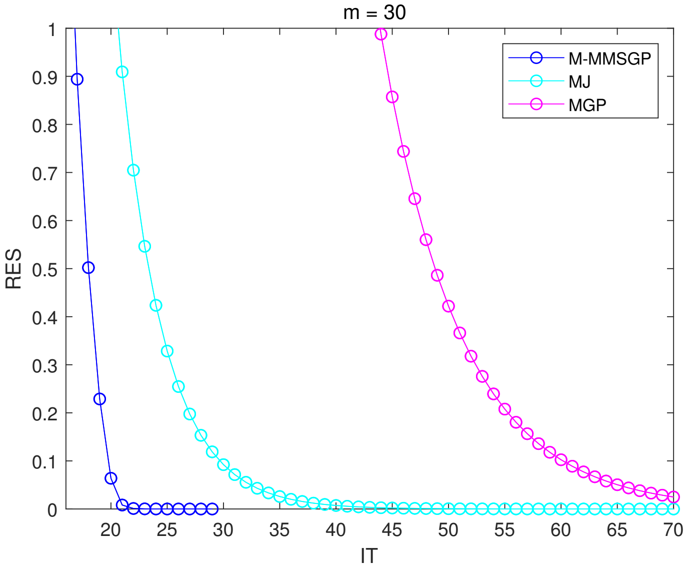

Figure 2. Relationship between IT and RES for Example 5.3.

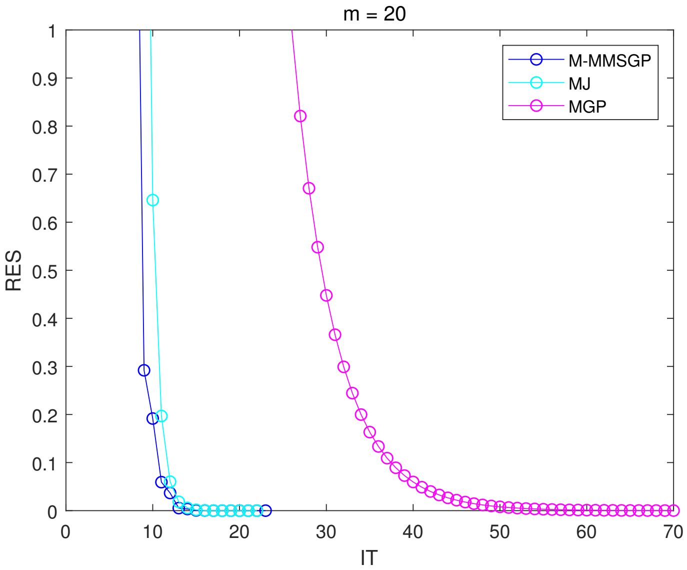

Figure 3. Relationship between IT and RES for Example 5.4.

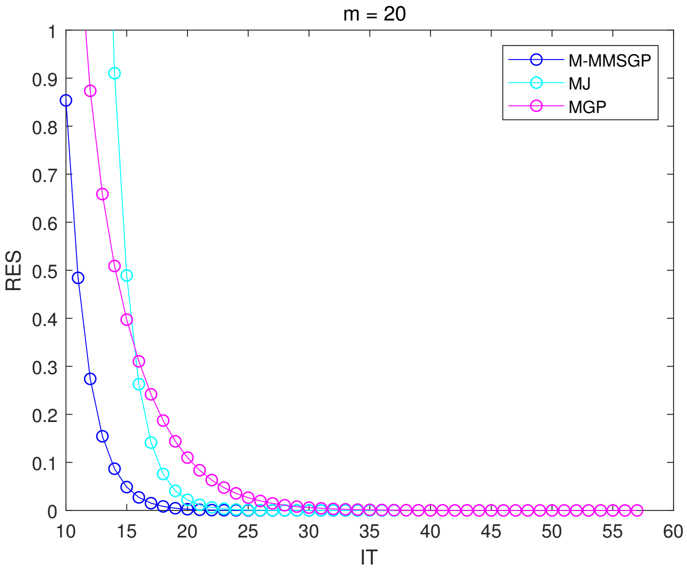

Figure 4. Relationship between IT and RES for Example 5.5.

6. Conclusions

The focus of this study is the HLCP. First, the HLCP is reconstructed into a fixed-point equation based on the modulus defined in the nonnegative cone. By rewriting the fixed-point equation as a monotone system, we put forward a new class of modified multivariate spectral gradient projection method for solving it. It is shown that the proposed iterative method converges to the HLCP solution, assuming the specified conditions are met. Furthermore, the viability and efficacy of the modulus-based modified multivariate spectral gradient projection method are illustrated through numerical examples. The algorithm presented in this paper has been developed to address the horizontal complementarity problem within the theoretical framework of this study. Further efforts could be directed toward solving this problem under more simplified assumptions.

Author contributions

Ting Lin: Conceptualization, Methodology, Validation, Formal analysis, Resources, Writing–review and editing, Supervision, Project administration; Hong Zhang: Data curation, Visualization; Chaofan Xie: Conceptualization, Software, Validation, Data curation, Writing–original draft preparation, 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 article.

Acknowledgments

This research was funded by the National Natural Science Foundation of China under Grant 62071123 and 61601125, Natural Science Foundation of Fujian Province of China (number: 2023J011117), the Fujian Province Education Hall Youth Project (number: JAT220258), and the Fujian Natural Science Foundation Project (number: 2019J01887).

Conflict of interest

The authors declare no conflicts of interest.

References

1. M. C. Ferris, J. S. Pang, Engineering and economic applications of complementarity problems, *SIAM Rev.*, **39** (1997), 669–713. http://doi.org/10.1137/S0036144595285963
2. H. Zheng, S. Vong, On the modulus-based successive overrelaxation iteration method for horizontal linear complementarity problems arising from hydrodynamic lubrication, *Appl. Math. Comput.*, **402** (2021), 126165. https://doi.org/10.1016/j.amc.2021.126165
3. S. Asadi, H. Mansouri, Z. Darvay, M. Zangiabadi, N. Mahdavi-Amiri, Large-neighborhood infeasible predictor-corrector algorithm for horizontal linear complementarity problems over Cartesian product of symmetric cones, *J. Optim. Theory Appl.*, **180** (2019), 811–829. https://doi.org/10.1007/s10957-018-1402-6
4. X. B. Gao, J. Wang, Analysis and application of a one-layer neural network for solving horizontal linear complementarity problems, *Int. J. Comput. Intell. Syst.*, **7** (2014), 724–732. https://doi.org/10.1080/18756891.2013.858903
5. U. Schäfer, Verification methods for the horizontal linear complementarity problem, *Proc. Appl. Math. Mech.*, **8** (2008), 10799–10800. https://doi.org/10.1002/pamm.200810799

6. F. Mezzadri, E. Galligani, Splitting methods for a class of horizontal linear complementarity problems, *J. Optim. Theory Appl.*, **180** (2019), 500–517. https://doi.org/10.1007/s10957-018-1395-1

7. F. Mezzadri, E. Galligani, Modulus-based matrix splitting methods for horizontal linear complementarity problems, *Numer. Algor.*, **83** (2020), 201–219. https://doi.org/10.1007/s11075-019-00677-y

8. H. Zheng, S. Vong, On convergence of the modulus-based matrix splitting iteration method for horizontal linear complementarity problems of H_+-matrices, *Appl. Math. Comput.*, **369** (2020), 124890. https://doi.org/10.1016/j.amc.2019.124890

9. Y. X. Zhang, H. Zheng, S. Vong, X. P. Lu, A two-step parallel iteration method for large sparse horizontal linear complementarity problems, *Appl. Math. Comput.*, **438** (2023), 127609. https://doi.org/10.1016/j.amc.2022.127609

10. J. Barzilai, J. M. Borwein, Two-point step size gradient methods, *IMA J. Numer. Anal.*, **8** (1988), 141–148. https://doi.org/10.1093/imanum/8.1.141

11. W. La Cruz, M. Raydan, Nonmonotone spectral methods for large-scale nonlinear systems, *Optim. Methods Softw.*, **18** (2003), 583–599. https://doi.org/10.1080/10556780310001610493

12. W. La Cruz, J. M. Martínez, M. Raydan, Spectral residual method without gradient information for solving large-scale nonlinear systems of equations, *Math. Comp.*, **75** (2006), 1429–1448. https://doi.org/10.1090/S0025-5718-06-01840-0

13. 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

14. L. Zhang, W. J. Zhou, Spectral gradient projection method for solving nonlinear monotone equations, *J. Comput. Appl. Math.*, **196** (2006), 478–484. https://doi.org/10.1016/j.cam.2005.10.002

15. 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, 1998, 355–369. https://doi.org/10.1007/978-1-4757-6388-1_18

16. Z. S. Yu, J. Lin, J. Sun, Y. H. Xiao, L. Y. Liu, Z. H. Li, Spectral gradient projection method for monotone nonlinear equations with convex constraints, *Appl. Numer. Math.*, **59** (2009), 2416–2423. https://doi.org/10.1016/j.apnum.2009.04.004

17. A. H. Ibrahim, P. Kumam, M. Sun, P. Chaipunya, A. B. Abubakar, Projection method with inertial step for nonlinear equations: application to signal recovery, *J. Ind. Manag. Optim.*, **19** (2022), 30–55. https://doi.org/10.1016/j.apnum.2009.04.004

18. D. D. Li, J. Q. Wu, Y. Li, S. H. Wang, A modified spectral gradient projection-based algorithm for large-scale constrained nonlinear equations with applications in compressive sensing, *J. Comput. Appl. Math.*, **424** (2023), 115006. https://doi.org/10.1016/j.cam.2022.115006

19. A. H. Ibrahim, S. Al-Homidan, Two-step inertial derivative-free projection method for solving nonlinear equations with application, *J. Comput. Appl. Math.*, **451** (2024), 116071. https://doi.org/10.1016/j.cam.2024.116071

20. H. Y. Jiang, Unconstrained minimization approaches to nonlinear complementarity problems, *J. Global Optim.*, **9** (1996), 169–181. https://doi.org/10.1007/BF00121662

21. O. L. Mangasarian, M. V. Solodov, A linearly convergent derivative-free descent method for strongly monotone complementarity problems, *Comput. Optim. Appl.*, **14** (1999), 5–16. https://doi.org/10.1023/A:1008752626695

22. C. F. Ma, L. J. Chen, D. S. Wang, A globally and superlinearly convergent smoothing Broyden-like method for solving nonlinear complementarity problem, *Appl. Math. Comput.*, **198** (2008), 592–604. https://doi.org/10.1016/j.amc.2007.08.057

23. Z. S. Yu, L. Li, Y. Yuan, A modified multivariate spectral gradient algorithm for solving absolute value equations, *Appl. Math. Lett.*, **121** (2021), 107461. https://doi.org/10.1016/j.aml.2021.107461

24. K. Amini, A. Kamandi, A new line search strategy for finding separating hyperplane in projection-based methods, *Numer. Algor.*, **70** (2015), 559–570. https://doi.org/10.1007/s11075-015-9961-1

25. J. Yang, H. W. Liu, A modified projected gradient method for monotone variational inequalities, *J. Optim. Theory Appl.*, **179** (2018), 197–211. https://doi.org/10.1007/s10957-018-1351-0