Journal of Industrial and Management Optimization

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

JIMO, 22(7): 3175–3187.
DOI:10.3934/jimo.2026116
Received: 31 October 2025
Revised: 09 March 2026
Accepted: 21 May 2026
Published: 11 June 2026

Theory article

A computationally improved Levenberg−Marquardt method for systems of inequalities

Kaile Ren and Jingyong Tang*

School of Mathematics and Statistics, Henan Provincial Center for Applied Mathematics, Xinyang Normal University, Xinyang 464000, China

* **Correspondence:** Email: tangjy@xynu.edu.cn.

Abstract: This paper addresses a system of inequalities by first transforming it into a system of nonlinear equations. We then develop a computationally enhanced Levenberg−Marquardt (LM) method to solve this equivalent system. In contrast to the standard LM approach, which solves a large linear system, the proposed method requires solving a linear system of significantly reduced dimensions at each iteration, leading to considerable computational savings. Global convergence is ensured through the incorporation of a derivative-free line search. We demonstrate that any accumulation point of the generated sequence is a solution to the original inequality system, and that a superlinear convergence rate is achievable under a local error bound condition. A key advantage of this method is its consistent ability to compute exact solutions to the inequality system. Numerical experiments validate the effectiveness of the proposed LM method and highlight its superior performance over the standard LM method, particularly for large-scale problems.

Keywords: system of inequalities; Levenberg−Marquardt method; superlinear convergence
Mathematics Subject Classification: 90C33, 65K05

1. Introduction

We consider the system of inequalities

$$\mathrm{F}(x) \leq 0, \tag{1.1}$$

where $\mathrm{F}(x) := (\mathrm{F}_1(x), ..., \mathrm{F}_m(x))^T$ with $\mathrm{F}_i : \Re^n \to \Re$ for any $i \in \{1, ..., m\}$ and F is continuously differentiable.

The system of inequalities (1.1) has garnered significant attention due to its applicability in diverse areas such as data analysis, set separation problems, computer-aided design, and image reconstruction. Numerous numerical algorithms have been developed for solving (1.1), including the predictor−corrector method [1], the Broyden-like method [2], the smoothing homotopy method [3], the smoothing-type algorithm [4], the nonmonotone smoothing Newton method [5], and the regularized smoothing-type algorithm [6]. In addition, some algorithms have been studied for solving general nonlinear systems of equalities and inequalities, such as the smooth Newton method [7], the filter algorithm [8], and the nonmonotone smoothing-type algorithm [9]. Recently, Huang et al. proposed a nonmonotone smoothing-type algorithm for a system of inequalities associated with circular cones [10]. Among these, smoothing Newton-type methods are particularly prominent. These methods typically use a smoothing function to reformulate (1.1) as a smooth system of nonlinear equations, which is then tackled with Newton's method. It is worth noting, however, that in these smoothing Newton-type methods, to ensure the feasibility of the Newton step, it is necessary that $n = m$ and that the function $\mathrm{F}(x)$ has the Cartesian P_0 property. Consequently, smoothing Newton-type methods may not be applicable to general inequality systems.

In this work, we introduce the following system of nonlinear equations:

$$\Phi(x, t) := \mathrm{F}(x) + e^t = 0, \tag{1.2}$$

where $\mathrm{F}(x)$ is given in (1.1), and $e^t := (e^{t_1}, ..., e^{t_m})^T$ for any $t_i \in \Re$ with $i \in \{1, ..., m\}$. Clearly, the function $\Phi(x, t)$ is continuously differentiable, and its Jacobian matrix is given by

$$\Phi'(x, t) = [\mathrm{F}'(x), \mathrm{diag}(e^t)] \in \Re^{m\times(n+m)}, \tag{1.3}$$

where $\mathrm{diag}(e^t)$ denotes a diagonal matrix whose diagonal elements are e^{t_i} $(i = 1, ..., m)$. It is evident that if (x^*, t^*) is a solution to the Eq (1.2), then x^* solves the inequalities in (1.1). Therefore, solving (1.2) provides a pathway to finding solutions for (1.1).

As is well established, the Levenberg–Marquardt (LM) method is highly effective for solving nonlinear equations; see the survey paper [11] for more information. Recently, Aravkin et al. [12] extended the LM method to nonsmooth regularized least squares. Tang and Zhou [13] developed an LM method with a Broyden-like update technique. At each iteration of the standard LM method applied to (1.2), the trial step d_k^{LM} is obtained by solving

$$[\Phi'(x^k, t^k)^T\Phi'(x^k, t^k) + \mu_k \mathrm{I}_{n+m}]d = -\Phi'(x^k, t^k)^T\Phi(x^k, t^k), \tag{1.4}$$

where I_{n+m} is the identity matrix in $\Re^{(n+m)\times(n+m)}$, and $\mu_k > 0$ is an adaptively updated LM parameter. Fan and Yuan [14] proved that by setting $\mu_k = \|\Phi(x^k, t^k)\|^\delta$ with $\delta \in [1, 2]$, the LM method achieves local quadratic convergence under a local error bound condition.

In this paper, we aim to present a computationally improved LM method for solving the nonlinear system (1.2). In each iteration, the proposed method first computes a vector $h_k \in \Re^m$ by solving

$$[\Phi'(x^k, t^k)\Phi'(x^k, t^k)^T + \mu_k \mathrm{I}_m]h = -\Phi(x^k, t^k) \tag{1.5}$$

and subsequently determines the step

$$d_k := \Phi'(x^k, t^k)^T h_k. \tag{1.6}$$

We will show that the direction d_k generated by (1.5) and (1.6) is identical to the standard LM step d_k^{LM} obtained from (1.4). However, the coefficient matrices of the linear systems in (1.4) and (1.5) are

$$[\Phi'(x^k, t^k)^T\Phi'(x^k, t^k) + \mu_k \mathrm{I}_{n+m}] \in \Re^{(n+m)\times(n+m)}$$

and

$$[\Phi'(x^k, t^k)\Phi'(x^k, t^k)^T + \mu_k \mathrm{I}_m] \in \mathfrak{R}^{m\times m},$$

respectively. Since the linear system in (1.5) has a much smaller dimension than (1.4), the computational cost can be significantly reduced, especially for large-scale inequalities. Furthermore, the proposed LM method integrates a derivative-free line search to ensure global convergence. We establish that every accumulation point of the sequence produced by our LM method is a solution to the original inequality system, and that superlinear convergence is attained under a local error bound condition. Finally, numerical results are presented to confirm the efficacy of our approach. These results demonstrate that the proposed LM method not only consistently yields exact solutions to the inequalities but also offers significant computational advantages over the standard LM method, especially when $m \ll n$.

The remainder of this paper is structured as follows. Section 2 details the computationally improved LM algorithm. The analysis of global convergence and the local convergence rate is presented in Sections 3 and 4, respectively. Section 5 reports the numerical findings, and Section 6 provides some concluding remarks and discussions.

2. A computationally improved LM method

For notational convenience, we henceforth denote $z := (x, t)$ and $z^k := (x^k, t^k)$. Let $\Phi(z)$ be defined by (1.2). We define the merit function $\Psi : \mathfrak{R}^{n+m} \to \mathfrak{R}$ as

$$\Psi(z) = \frac{1}{2}\|\Phi(z)\|^2.$$

The function $\Psi(z)$ is continuously differentiable on $\mathfrak{R}^{n+m}$ with the gradient $\nabla\Psi(z) = \Phi'(z)^T\Phi(z)$, where $\Phi'(z)$ is given in (1.3). We now provide a detailed description of our improved LM method, referred to as the I-LMM.

Algorithm I-LMM: Choose the parameters $\rho, \gamma \in (0, 1)$, $\theta > 0$ and $\delta \in [1, 2]$. Choose a positive sequence $\{\zeta_k\}$ satisfying $\sum_{k=0}^{\infty} \zeta_k \le \zeta < \infty$, where $\zeta > 0$ is a constant. Choose $x^0 \in \mathfrak{R}^n$ and $t^0 \in \mathfrak{R}^m$, and set $z^0 := (x^0, t^0)$. Set $k := 0$.

Step 1: If $\|\Phi(z^k)\| = 0$, then stop.

Step 2: Set $\mu_k := \theta\|\Phi(z^k)\|^\delta$. Compute $h_k \in \mathfrak{R}^m$ by solving the linear system

$$[\Phi'(z^k)\Phi'(z^k)^T + \mu_k \mathrm{I}_m]h = -\Phi(z^k), \tag{2.1}$$

and set

$$d_k := \Phi'(z^k)^T h_k. \tag{2.2}$$

Step 3: Find a step size $\lambda_k := \rho^{l_k}$, where l_k is the smallest non-negative integer l satisfying

$$\|\Phi(z^k + \rho^l d_k)\| \le (1 + \zeta_k)\|\Phi(z^k)\| - \gamma\|\rho^l\Phi(z^k)\|^2. \tag{2.3}$$

Step 4: Set $z^{k+1} := z^k + \lambda_k d_k$. Replace k by $k + 1$ and go to Step 1.

Remark 2.1. (i) Algorithm I-LMM is well-defined. In fact, when $\|\Phi(z^k)\| \neq 0$, we have $\mu_k > 0$, ensuring that the matrix $\Phi'(z^k)\Phi'(z^k)^T + \mu_k \mathrm{I}_m$ is positive definite. Consequently, the search direction d_k in Step 2 is uniquely determined. Moreover, because

$$\begin{aligned}\lim_{l\to\infty} \|\Phi(z^k + \rho^l d_k)\| &= \|\Phi(z^k)\| < (1 + \zeta_k)\|\Phi(z^k)\| \\ &= \lim_{l\to\infty}[(1 + \zeta_k)\|\Phi(z^k)\| - \gamma\|\rho^l \Phi(z^k)\|^2],\end{aligned}$$

the inequality (2.3) holds for all sufficiently large values of $l > 0$. Hence, z^{k+1} is well-defined in Step 4.
(ii) It is straightforward to verify that the standard LM step d_k^{LM} given by (1.4) is

$$d_k^{\mathrm{LM}} = -[\Phi'(z^k)^T \Phi'(z^k) + \mu_k \mathrm{I}_{n+m}]^{-1} \Phi'(z^k)^T \Phi(z^k). \tag{2.4}$$

Using the identity

$$[\Phi'(z^k)^T \Phi'(z^k) + \mu_k \mathrm{I}_{n+m}]\Phi'(z^k)^T = \Phi'(z^k)^T [\Phi'(z^k)\Phi'(z^k)^T + \mu_k \mathrm{I}_m],$$

we have

$$\Phi'(z^k)^T [\Phi'(z^k)\Phi'(z^k)^T + \mu_k \mathrm{I}_m]^{-1} = [\Phi'(z^k)^T \Phi'(z^k) + \mu_k \mathrm{I}_{n+m}]^{-1} \Phi'(z^k)^T. \tag{2.5}$$

According to (2.4) and (2.5), we obtain the following from Step 2:

$$\begin{aligned} d_k &= \Phi'(z^k)^T h_k \\ &= -\Phi'(z^k)^T [\Phi'(z^k)\Phi'(z^k)^T + \mu_k \mathrm{I}_m]^{-1} \Phi(z^k) \\ &= -[\Phi'(z^k)^T \Phi'(z^k) + \mu_k \mathrm{I}_{n+m}]^{-1} \Phi'(z^k)^T \Phi(z^k) \\ &= d_k^{\mathrm{LM}}. \end{aligned} \tag{2.6}$$

This indicates that the direction d_k generated by (2.1) and (2.2) is identical to the standard LM step d_k^{LM} obtained by (1.4). However, as noted in the introduction, the problem size of (2.1) is much smaller than that of (1.4). Consequently, solving (2.1) can save significant computational cost compared with solving (1.4), especially for large-scale inequalities.

3. Global convergence

Lemma 3.1. *Let* $\{z^k\}$ *be the iteration sequence generated by Algorithm* I-LMM. *Then there exists a constant* $\Phi^* \geq 0$ *such that* $\lim_{k\to\infty} \|\Phi(z^k)\| = \Phi^*$. *Moreover,* $\lim_{k\to\infty} \lambda_k \|\Phi(z^k)\| = 0$.

Proof From Step 3, we have $\|\Phi(z^{k+1})\| \leq (1+\zeta_k)\|\Phi(z^k)\|$. Since $\sum_{k=0}^{\infty} \zeta_k \leq \zeta < \infty$, according to [15, Lemma 2.2], the sequence $\{\|\Phi(z^k)\|\}$ is convergent. Thus, a constant $\Phi^* \geq 0$ exists such that $\lim_{k\to\infty} \|\Phi(z^k)\| = \Phi^*$. Moreover, by Step 3, for all $k \geq 0$, we have

$$\gamma(\lambda_k \|\Phi(z^k)\|)^2 \leq (1 + \zeta_k)\|\Phi(z^k)\| - \|\Phi(z^{k+1})\|.$$

This, together with $\lim_{k\to\infty} \|\Phi(z^k)\| = \Phi^*$ and $\lim_{k\to\infty} \zeta_k = 0$, proves the second result. □

Theorem 3.1. *Let* $z^* := (x^*, t^*)$ *be an accumulation point of the iteration sequence* $\{z^k = (x^k, t^k)\}$ *generated by Algorithm* I-LMM. *Then* $\Phi(z^*) = 0$ *and hence* $\mathrm{F}(x^*) < 0$.

Proof By Lemma 3.1, a constant $\Phi^* \geq 0$ exists such that $\lim_{k\to\infty} \|\Phi(z^k)\| = \Phi^*$. Consequently,

$$\lim_{k\to\infty} \Psi(z^k) = \frac{1}{2}(\Phi^*)^2 \quad \text{and} \quad \lim_{k\to\infty} \mu_k = \theta(\Phi^*)^{\delta}.$$

Without loss of generality, we may assume that $\lim_{(K\ni)k\to\infty} z^k = z^*$ where $K \subset \{0, 1, ...\}$. Then, by the continuity, we have

$$\lim_{(K\ni)k\to\infty} \Phi(z^k) = \Phi(z^*), \quad \lim_{(K\ni)k\to\infty} \Phi'(z^k) = \Phi'(z^*),$$

and

$$\lim_{(K\ni)k\to\infty} \nabla\Psi(z^k) = \nabla\Psi(z^*) = \Phi'(z^*)^T\Phi(z^*).$$

Clearly, $\Phi^* = \|\Phi(z^*)\|$. Note that if $\Phi^* = 0$, then $\Phi(z^*) = \mathrm{F}(x^*) + e^{t^*} = 0$, which proves the theorem. Now we assume that $\Phi^* > 0$. Since

$$\lim_{(K\ni)k\to\infty} [\Phi'(z^k)\Phi'(z^k)^T + \mu_k \mathrm{I}_m] = \Phi'(z^*)\Phi'(z^*)^T + \theta(\Phi^*)^{\delta}\mathrm{I}_m, \tag{3.1}$$

and the matrix $\Phi'(z^*)\Phi'(z^*)^T + \theta(\Phi^*)^{\delta}\mathrm{I}_m$ is positive definite, from (2.1) and (3.1), it follows that

$$\lim_{(K\ni)k\to\infty} h_k = -[\Phi'(z^*)\Phi'(z^*)^T + \theta(\Phi^*)^{\delta}\mathrm{I}_m]^{-1}\Phi(z^*).$$

This and (2.2) yield

$$\lim_{(K\ni)k\to\infty} d_k = -\Phi'(z^*)^T[\Phi'(z^*)\Phi'(z^*)^T + \theta(\Phi^*)^{\delta}\mathrm{I}_m]^{-1}\Phi(z^*) := d^*. \tag{3.2}$$

By Lemma 3.1, $\lim_{(K\ni)k\to\infty} \lambda_k\|\Phi(z^k)\| = 0$. Since $\lim_{(K\ni)k\to\infty} \|\Phi(z^k)\| = \Phi^* > 0$, it follows that $\lim_{(K\ni)k\to\infty} \lambda_k = 0$. From (2.3), for all $k \in K$, we have

$$\|\Phi(z^k + \rho^{-1}\lambda_k d_k)\| \quad > \quad (1 + \zeta_k)\|\Phi(z^k)\| - \gamma(\rho^{-1}\lambda_k)^2\|\Phi(z^k)\|^2,$$

which implies

$$\frac{\|\Phi(z^k + \rho^{-1}\lambda_k d_k)\| - \|\Phi(z^k)\|}{\rho^{-1}\lambda_k} \geq -\gamma\rho^{-1}\lambda_k\|\Phi(z^k)\|^2.$$

Multiplying both sides of the inequality above by $\frac{\|\Phi(z^k+\rho^{-1}\lambda_k d_k)\|+\|\Phi(z^k)\|}{2}$ gives

$$\frac{\Psi(z^k + \rho^{-1}\lambda_k d_k) - \Psi(z^k)}{\rho^{-1}\lambda_k} \geq -\frac{1}{2}\gamma\rho^{-1}\lambda_k\|\Phi(z^k)\|^2[\|\Phi(z^k + \rho^{-1}\lambda_k d_k)\| + \|\Phi(z^k)\|]. \tag{3.3}$$

Since Ψ is continuously differentiable, letting $k \to \infty$ with $k \in K$ in (3.3) yields $\nabla\Psi(z^*)^T d^* \geq 0$. On the other hand, by the third equation in (2.6), we get

$$\nabla\Psi(z^k)^T d_k \quad = \quad -\nabla\Psi(z^k)^T[\Phi'(z^k)^T\Phi'(z^k) + \mu_k \mathrm{I}_{n+m}]^{-1}\nabla\Psi(z^k). \tag{3.4}$$

Since the matrix $[\Phi'(z^k)^T\Phi'(z^k) + \mu_k \mathrm{I}_{n+m}]^{-1}$ is positive definite, it follows from (3.4) that $\nabla\Psi(z^k)^T d_k \leq 0$, and hence $\nabla\Psi(z^*)^T d^* \leq 0$. Therefore, we conclude that $\nabla\Psi(z^*)^T d^* = 0$. Furthermore, Eq (3.4) implies that

$$\nabla\Psi(z^*)^T d^* = -\nabla\Psi(z^*)^T[\Phi'(z^*)^T\Phi'(z^*) + \theta(\Phi^*)^{\delta}\mathrm{I}_{n+m}]^{-1}\nabla\Psi(z^*) = 0.$$

Because the matrix $[\Phi'(z^*)^T\Phi'(z^*) + \theta(\Phi^*)^\delta \mathrm{I}_{n+m}]^{-1}$ is positive definite, it follows that $\nabla\Psi(z^*) = 0$, i.e.,

$$[\mathrm{F}'(x^*), \mathrm{diag}(e^{t^*})]^T \Phi(z^*) = 0.$$

This gives $\mathrm{diag}(e^{t^*})\Phi(z^*) = 0$, and hence $\Phi(z^*) = \mathrm{F}(x^*) + e^{t^*} = 0$, from which we conclude that $\mathrm{F}(x^*) < 0$. The proof is complete. □

4. Local convergence rate

In this section, we assume that $z^* := (x^*, t^*)$ is an accumulation point of the iteration sequence $\{z^k = (x^k, t^k)\}$ generated by Algorithm I-LMM. We will demonstrate that the entire sequence $\{z^k\}$ converges to z^* superlinearly under the following assumption.

Assumption 4.1. *The positive constants $\varrho > 0$ and $0 < r < 1$ exist such that*

$$\|\Phi(z)\| \geq \varrho\|z - z^*\|, \quad \forall\, z \in N(z^*, r) := \{z \in \Re^{n+m} |\, \|z - z^*\| \leq r\}. \tag{4.1}$$

Assumption 4.1 is a local error bound condition, commonly used in analyzing the local convergence properties of smoothing LM methods [16, 17] and the smoothing Newton method [18].

Lemma 4.1. *Under Assumption* 4.1*, for all z^k sufficiently close to z^*, we have*

$$\|d_k\| = o(\|z^k - z^*\|^{1-\frac{\delta}{2}}).$$

Proof By Theorem 3.1, we have $\Phi(z^*) = 0$. Since Φ is continuously differentiable, it is locally Lipschitz continuous at z^*. Thus, for all z^k sufficiently close to z^*, we have

$$\mu_k = \theta\|\Phi(z^k) - \Phi(z^*)\|^\delta = O(\|z^k - z^*\|^\delta). \tag{4.2}$$

Since Φ is continuously differentiable, it is semismooth. Hence, for all z^k sufficiently close to z^*, we have

$$\|\Phi(z^k) - \Phi(z^*) - \Phi'(z^k)(z^k - z^*)\| = o(\|z^k - z^*\|). \tag{4.3}$$

For any $k \geq 0$, consider the following minimization problem:

$$\min_{d\in\Re^{n+m}} f_k(d) := \|\Phi(z^k) + \Phi'(z^k)d\|^2 + \mu_k\|d\|^2. \tag{4.4}$$

Then the trial step d_k^{LM} given in (2.4) is a solution of (4.4), because $f_k(d)$ is a strictly convex quadratic function and d_k^{LM} is a stationary point of $f_k(d)$. By (2.6), (4.2), and (4.3), for all z^k sufficiently close to z^*, we have

$$\begin{aligned} f_k(d_k) &= f_k(d_k^{\mathrm{LM}}) \leq f_k(z^* - z^k) \\ &= \|\Phi(z^k) - \Phi(z^*) - \Phi'(z^k)(z^k - z^*)\|^2 + \mu_k\|z^k - z^*\|^2 \\ &= o(\|z^k - z^*\|^2). \end{aligned} \tag{4.5}$$

Moreover, by (4.1), for all z^k sufficiently close to z^*, we have

$$\mu_k = \theta\|\Phi(z^k)\|^\delta \geq \theta\varrho^\delta\|z^k - z^*\|^\delta.$$

Combining this with (4.4) and (4.5), for all z^k sufficiently close to z^*, we have

$$\|d_k\| \leq \sqrt{\frac{f_k(d_k)}{\mu_k}} = \sqrt{\frac{o(\|z^k - z^*\|^2)}{\theta\varrho^\delta\|z^k - z^*\|^\delta}} = o(\|z^k - z^*\|^{1-\frac{\delta}{2}}),$$

which proves the lemma. □

Theorem 4.1. *Under Assumption* 4.1*, the whole sequence* $\{z^k\}$ *converges to* z^* *and*

$$\|z^{k+1} - z^*\| = o(\|z^k - z^*\|).$$

Proof Without loss of generality, we assume that $\lim_{(K\ni)k\to\infty} z^k = z^*$ where $K \subset \{0, 1, ...\}$. Then,

$$\lim_{(K\ni)k\to\infty} \Phi(z^k) = \Phi(z^*) = 0, \text{ and } \lim_{(K\ni)k\to\infty} \mu_k = 0.$$

By Lemma 4.1, we have $\lim_{(K\ni)k\to\infty} d_k = 0$. Consequently, $\lim_{(K\ni)k\to\infty} (z^k + d_k) = z^*$. Using the semismoothness of Φ again, for all sufficiently large $k \in K$, we have

$$\|\Phi(z^k + d_k) - \Phi(z^*) - \Phi'(z^k + d_k)(z^k + d_k - z^*)\| = o(\|z^k + d_k - z^*\|). \tag{4.6}$$

Additionally, by the continuity of $\Phi'(x)$,

$$\lim_{(K\ni)k\to\infty} \|\Phi'(z^k + d_k) - \Phi'(z^k)\| = \|\Phi'(z^*) - \Phi'(z^*)\| = 0. \tag{4.7}$$

Therefore, from (4.3), (4.6), and (4.7), for all sufficiently large $k \in K$, we have

$$\begin{aligned} & \|\Phi(z^k + d_k) - \Phi(z^k) - \Phi'(z^k)d_k\| \\ \leq\ & \|\Phi(z^k + d_k) - \Phi(z^*) - \Phi'(z^k + d_k)(z^k + d_k - z^*)\| \\ & +\|\Phi(z^k) - \Phi(z^*) - \Phi'(z^k)(z^k - z^*)\| \\ & +\|(\Phi'(z^k + d_k) - \Phi'(z^k))(z^k + d_k - z^*)\| \\ =\ & o(\|z^k + d_k - z^*\|) + o(\|z^k - z^*\|). \end{aligned} \tag{4.8}$$

Additionally, from (4.4) and (4.5), for all values of z^k sufficiently close to z^*, we have

$$\|\Phi(z^k) + \Phi'(z^k)d_k\| \leq \sqrt{f_k(d_k)} = o(\|z^k - z^*\|). \tag{4.9}$$

It follows from (4.1), (4.8), and (4.9) that for all sufficiently large $k \in K$, we have

$$\begin{aligned} \varrho\|z^k + d_k - z^*\| &\leq \|\Phi(z^k + d_k)\| \\ &\leq \|\Phi(z^k + d_k) - \Phi(z^k) - \Phi'(z^k)d_k\| \\ &\quad +\|\Phi(z^k) + \Phi'(z^k)d_k\| \\ &= o(\|z^k + d_k - z^*\|) + o(\|z^k - z^*\|), \end{aligned}$$

which implies

$$\|z^k + d_k - z^*\| = o(\|z^k - z^*\|). \tag{4.10}$$

Since Φ is locally Lipschitz continuous at z^* and $\Phi(z^*) = 0$, it follows from (4.10) and (4.1) that for all sufficiently large $k \in K$, we have

$$\|\Phi(z^k + d_k)\| = O(\|z^k + d_k - z^*\|) = o(\|z^k - z^*\|) = o(\|\Phi(z^k)\|).$$

Thus, for all sufficiently large $k \in K$, we have

$$\|\Phi(z^k + d_k)\| + \gamma\|\Phi(z^k)\|^2 \leq \|\Phi(z^k)\| \leq (1 + \zeta_k)\|\Phi(z^k)\|.$$

This implies that, for all sufficiently large $k \in K$, $\lambda_k = 1$ and hence $z^{k+1} = z^k + d_k$. Furthermore, by (4.10), for all sufficiently large $k \in K$, we have

$$\|z^{k+1} - z^*\| = o(\|z^k - z^*\|). \tag{4.11}$$

Hence, for any $\varepsilon > 0$, if $z^k \in N(z^*, \varepsilon)$, then $z^{k+1} \in N(z^*, \varepsilon)$. This, together with $\lim\limits_{(K\ni)k\to\infty} z^k = z^*$, implies that the whole sequence $\{z^k\}$ converges to z^*. The superlinear convergence follows from (4.11). We complete the proof. □

Note that if F′ is locally Lipschitz continuous, then Φ is strongly semismooth on $\Re^{n+m}$, which gives

$$\|\Phi(z^k) - \Phi(z^*) - \Phi'(z^k)(z^k - z^*)\| = O(\|z^k - z^*\|^2).$$

For the definition of (strong) semismoothness, one may refer to [19]. In the identical way to the proof of Theorem 4.1, we can obtain the following result.

Theorem 4.2. *Under Assumption* 4.1*, if* F′ *is locally Lipschitz continuous, then*

$$\|z^{k+1} - z^*\| = O(\|z^k - z^*\|^{1+\frac{\delta}{2}}),$$

and the convergence rate of $\{z^k\}$ *is quadratic when* $\delta = 2$.

5. Numerical results

This section presents some numerical results obtained with Algorithm I-LMM. The parameters for Algorithm I-LMM are set as follows: $\theta = 10^{-3}$, $\delta = 1$, $\rho = 0.8$, $\gamma = 10^{-5}$, and $\zeta_k = 0.8^k$. The initial vector t^0 is chosen as $t^0 := (1, ..., 1)^T$, while the initial x^0 is specified for each example. The stopping criterion is $\|\Phi(z^k)\| \leq 10^{-3}$. We first apply Algorithm I-LMM to solve the following six examples coming from [1, 5, 6].

Example 5.1. Consider (1.1), where $F(x) = (F_1(x), ..., F_6(x))^T$ with $x \in \Re^2$, and

$$F_1(x) = \sin x_1, \;\; F_2(x) = -\cos x_2, \;\; F_3(x) = x_1 - 3\pi,$$

$$F_4(x) = x_2 - \frac{\pi}{2} - 2, \;\; F_5(x) = -x_1 - \pi, \;\; F_6(x) = -x_2 - \frac{\pi}{2}.$$

Example 5.2. Consider (1.1), where $F(x) = (F_1(x), ..., F_5(x))^T$ with $x \in \Re^5$, and

$$F_1(x) = x_1 + x_3 - 2, \;\; F_2(x) = 2x_2 + x_4 - 3,$$

$$F_3(x) = -x_3 - x_4 + x_5, \;\; F_4(x) = x_1^2 + x_3^2 - 2, \;\; F_5(x) = x_2^2 + x_4^2 - 4.$$

Example 5.3. Consider (1.1), where $F(x) = (F_1(x), F_2(x), F_3(x))^T$ with $x \in \Re^2$, and

$$F_1(x) = 1 - e^{x_1+x_2}, \;\; F_2(x) = 2e^{x_1} + e^{x_2} - 3, \;\; F_3(x) = x_1 + e^{x_2} - 2.$$

Example 5.4. Consider (1.1), where $F(x) = (F_1(x), F_2(x))^T$ with $x \in \Re^2$, and

$$F_1(x) = x_1 - 0.7\sin x_1 - 0.2\cos x_2, \;\; F_2(x) = x_2 - 0.7\cos x_1 - 0.2\sin x_2.$$

Example 5.5. Consider (1.1), where $F(x) = (F_1(x), F_2(x))^T$ with $x \in \Re^3$, and

$$F_1(x) = x_1 + x_2 e^{0.8x_3} + e^{1.6}, \;\; F_2(x) = x_1^2 + x_2^2 + x_3^2 - 5.2675.$$

Example 5.6. Consider (1.1), where $F(x) = (F_1(x), F_2(x), F_3(x))^T$ with $x \in \Re^2$, and

$$F_1(x) = x_1(x_1 - 2)(x_1 - 3), \;\; F_2(x) = x_2^2 - 3x_2 + 2, \;\; F_3(x) = x_1^2 - x_2.$$

Table 1 summarizes the numerical results for Examples 5.1–5.6. In the table, **IT** denotes the number of iterations, x^* denotes the solution to the inequalities obtained by Algorithm I-LMM, and $\max_F := \max\{F_1(x^*), ..., F_m(x^*)\}$. The results demonstrate that Algorithm I-LMM performs effectively and consistently yields an exact solution to the inequality systems.

Table 1. Numerical results of Algorithm I-LMM for Examples 5.1–5.6.

Example	x^0	IT	x^*	$\max_F$
5.1	(1, 1)	5	(−0.0973, 0.6156)	−0.0971
	(2, 2)	5	(3.4197, 1.0045)	−0.2745
5.2	(2, 1, 2, 1, 2)	4	(0.3827, 0.2784 ,0.8492, 1.2363, 1.3078)	−0.7681
	(−2, −1, −2, −1, −2)	4	(−0.7289, −0.6533, −0.5333, −0.5839, −2.3262)	−1.1843
5.3	(1, 1)	7	(−0.3128, 0.3788)	−0.0683
	(2, 2)	11	(−0.6919, 0.6922)	−0.0003
5.4	(1, 1)	5	(−0.2577, 0.0548)	−0.2790
	(2, 2)	5	(−0.2259, 0.3295)	−0.2584
5.5	(0, 0, 0)	7	(−0.6976, −1.3871, 1.4482)	−0.1628
	(1, 1, 1)	7	(−0.5799, −1.4133, 1.4455)	−0.1189
5.6	(0, 0)	5	(−0.3051, 1.4993)	−0.2500
	(−1, −1)	6	(−0.3585, 1.5064)	−0.2500

Next, we consider the following system of inequalities:

$$F(x) := \begin{pmatrix} b - Ax \\ -x \end{pmatrix} \le 0, \tag{5.1}$$

where $A \in \Re^{m\times n}$ and $b \in \Re^n$. The system in (5.1) usually appears in the linear programming model as constraint conditions. In the experiments, we solve the following equations:

$$\Phi(z) := \Phi(x, u, v) = \begin{pmatrix} b - Ax + e^u \\ -x + e^v \end{pmatrix} = 0. \tag{5.2}$$

It is easy to see that

$$\Phi'(z) := \begin{bmatrix} -A & \mathrm{diag}(e^u) & \mathrm{zeros}(m,n) \\ -I_n & \mathrm{zeros}(n,m) & \mathrm{diag}(e^v) \end{bmatrix} \in \Re^{(n+m)\times(2n+m)}.$$

In our experiments, we generate a random matrix $A \in \Re^{m\times n}$ with a full row rank and set $b = (5, ..., 5)^T$. We compare the following three algorithms.

• The proposed LM method, i.e., Algorithm I-LMM;

• The standard LM method, which computes the direction d_k^{LM} by solving (1.4) and generates the step size λ_k by (2.3), denoted by LMM.

• The parameter-self-adjusting LM method [20], denoted by PSA-LMM. This method uses the trust region technique to ensure the global convergence.

We set $u_0 = (1, \ldots, 1)^T$, $v_0 = (1, \ldots, 1)^T$, and $x^0 = (1, \ldots, 1)^T$ as the starting points, and we use $|\Phi(z^k)| \leq 10^{-3}$ as the stopping criterion.

Numerical results are presented in Table 2, where **CPU** denotes the central processing unit (CPU) time in seconds. From Table 2, we observe that I-LMM and LMM require the same number of iterations and converge to the same solution. This is expected because the search direction generated by I-LMM is theoretically identical to that of LMM. However, I-LMM significantly reduces the CPU time compared with LMM, especially for large-scale inequalities. This finding verifies the theoretical advantage of I-LMM, which solves a linear system of much smaller dimensions than LMM at each iteration. Moreover, I-LMM consistently outperforms PSA-LMM, which requires more iterations and a longer CPU time. These results confirm the computational efficiency and theoretical advantages of the proposed method.

Table 2. Results of I-LMM, LMM, and PSA-LMM for solving (5.1).

		I-LMM			LMM			PSA-LMM		
	n	IT	CPU	$\max_F$	IT	CPU	$\max_F$	IT	CPU	$\max_F$
$m = \frac{n}{4}$	6000	6	33.79	−0.0868	6	70.37	−0.0868	11	134.88	−0.0789
	8000	6	86.43	−0.1145	6	139.84	−0.1145	11	245.31	−0.0922
	10000	6	118.97	−0.0879	6	279.54	−0.0879	12	527.01	−0.0765
	15000	6	330.91	−0.1118	6	857.76	−0.1118	12	2219.58	−0.0942
	20000	6	787.53	−0.1051	6	2051.68	−0.1051	12	3580.97	−0.0907
	25000	6	1439.07	−0.1088	6	3909.00	−0.1088	12	5983.98	−0.0823
$m = \frac{n}{8}$	6000	5	14.68	−0.2076	5	44.49	−0.2076	10	66.47	−0.1581
	8000	5	35.45	−0.2196	5	98.75	−0.2196	11	190.45	−0.1611
	10000	5	72.96	−0.2508	5	187.67	−0.2508	11	303.32	−0.1716
	15000	5	237.73	−0.2387	5	572.53	−0.2387	11	1819.91	−0.1703
	20000	5	450.02	−0.2221	5	1396.41	−0.2221	11	2440.21	−0.1632
	25000	5	955.29	−0.2091	5	2599.42	−0.2091	11	4763.02	−0.1658

It is well known that the choice of parameters plays a crucial role in the performance of algorithms. To assess the parameters' sensitivity, we tested a range of values for I-LMM, LMM, and PSA-LMM when solving (5.1) with $m = n/4$. The results show that the parameters ρ, γ, and θ have a minimal impact on the algorithm's performance, demonstrating the robustness of these LMM methods. In contrast, the parameter δ, which controls the update of the LM parameter $\mu_k = \theta\|\Phi(z^k)\|^\delta$, has a notable impact on computational efficiency. As shown in Table 3, for $\delta = 1$, all three LMM algorithms achieve the lowest CPU time and the fewest iterations, while increasing δ to 1.5 or 2.0 leads to longer computation times and more iterations. It seems that $\delta = 1$ yields the best computational efficiency.

Table 3. Results of I-LMM, LMM, and PSA-LMM with different values of δ.

		I-LMM		LMM		PSA-LMM	
n	δ	IT	CPU	IT	CPU	IT	CPU
2000	1.0	5	1.083	5	2.79	10	4.55
	1.5	6	1.26	6	3.30	12	5.49
	2.0	9	1.90	9	4.96	15	11.92
4000	1.0	5	6.40	5	16.32	11	27.40
	1.5	7	8.98	7	23.20	13	65.66
	2.0	11	23.55	11	48.78	17	60.79

6. Conclusions and discussions

By introducing the nonlinear system in (1.2), we have developed a computationally enhanced Levenberg−Marquardt (LM) method for solving the system of inequalities in (1.1). At each step, the proposed method solves a linear system whose dimension is considerably smaller than that required by the standard LM approach. We have established that the proposed LM method possesses global convergence properties and achieves a superlinear convergence rate under a local error bound condition. Numerical experiments confirm its effectiveness and demonstrate significant computational savings compared with the standard LM method, making it particularly attractive for large-scale inequality systems where $m \ll n$.

Finally, we conclude the paper by comparing the practical computation of our LM method with that of the smoothing Newton-type methods studied in [1, 5, 6]. First, when applying smoothing Newton-type methods to solve the inequalities (1.1) of size $m < n$, one needs to supplement $n - m$ additional inequalities to ensure that the Jacobian of the reformulated function is nonsingular, as seen in the numerical experiments in [1, 5, 6]. In contrast, the proposed LM method can directly solve the inequalities of (1.1) with $m < n$, and our numerical experiments have confirmed this advantage. Second, in order to obtain an exact solution to the inequalities of (1.1), smoothing Newton-type methods usually solve the following modified inequality:

$$\mathrm{F}(x) + \varepsilon \mathrm{e} \leq 0,$$

where $\varepsilon > 0$ is a sufficiently small number and $\mathrm{e} := (1, ..., 1)^T$, as seen again in the numerical experiments in [1, 5, 6]. However, for the proposed LM method, the solution (x^*, t^*) to the Eq (1.2) satisfies $\mathrm{F}(x^*) = -e^{t^*} < 0$. Hence, our LM method consistently generates an exact solution to the original inequality system in (1.1). As a future research direction, we intend to extend the framework of the proposed LM method to handle nonsmooth inequality systems by incorporating smoothing techniques or Clarke's generalized Jacobian.

Author contributions

Kaile Ren proposed the method, conducted the formal analysis and numerical experiments, and wrote the original draft. Jingyong Tang supervised the research, and reviewed and edited the manuscript.

Acknowledgments

This work was supported by the Henan Provincial Natural Science Foundation (252300423005) and the Key Scientific Research Projects of Higher Education of Henan Province (26A110018). We are very grateful to the referees for their valuable comments on the paper, which have considerably improved the paper.

Conflict of interest

The authors declare that they have no conflicts of interest.

Data Availability

The data sets generated during the current study are available from the author on reasonable request.

Use of Generative AI tools declaration

The authors declare that they did not utilize any artificial intelligence (AI) tools in the creation of this article.

References

1. H. Che, A smoothing and regularization predictor-corrector method for nonlinear inequalities, *J. Inequal. Appl.*, **2012** (2012), 214. https://doi.org/10.1186/1029-242X-2012-214
2. H. Che, M. Li, A smoothing and regularization Broyden-like method for nonlinear inequalities, *J. Appl. Math. Comput.*, **41** (2013), 209–227. https://doi.org/10.1007/s12190-012-0588-2
3. X. Fan, Q. Yan, Solving system of inequalities via a smoothing homotopy method, *Numer. Algor.*, **82** (2019), 719–728. https://doi.org/10.1007/s11075-018-0622-z
4. N. Yilmaz, A. Kayacan, A new smoothing algorithm to solve a system of nonlinear inequalities, *Fundam. J. Math. Appl.*, **6** (2023), 137–146. https://doi.org/10.33401/fujma.1261409
5. L. Y. Yuan, F. Cui, Z. P. Wan, W. Li, W. Wang, A nonmonotone smoothing Newton method for system of nonlinear inequalities based on a new smoothing function, *Comput. Appl. Math.*, **38** (2019), 91. https://doi.org/10.1007/s40314-019-0856-y
6. J. Zhu, H. Liu, X. Li, A regularized smoothing-type algorithm for solving a system of inequalities with a P_0-function, *J. Comput. Appl. Math.*, **233** (2010), 2611–2619. https://doi.org/10.1016/j.cam.2009.11.007

7. J. S. Chen, C. H. Ko, Y. D. Liu, S. P. Wang, New smoothing functions for solving a system of equalities and inequalities, *Pac. J. Optim.*, **12** (2016), 185–206.

8. C. Gu, D. T. Zhu, A filter algorithm for nonlinear systems of equalities and inequalities, *Appl. Math. Comput.*, **218** (2012), 10289–10298. https://doi.org/10.1016/j.amc.2012.04.007

9. Y. Zhang, Z. H. Huang, A nonmonotone smoothing-type algorithm for solving a system of equalities and inequalities, *J. Comput. Appl. Math.*, **233** (2010), 2312–2321. https://doi.org/10.1016/j.cam.2009.10.016

10. H. Huang, N. Qi, X. H. Miao, A nonmonotone smoothing-type algorithm for a system of inequalities associated with circular cones, *Asia-Pac. J. Oper. Res.*, **40** (2023), 2250017. https://doi.org/10.1142/S0217595922500178

11. A. Fischer, A. F. Izmailov, M. V. Solodov, The Levenberg−Marquardt method: an overview of modern convergence theories and more. *Comput. Optim. Appl.* **89** (2024), 33–67. https://doi.org/10.1007/s10589-024-00589-1

12. A. Y. Aravkin, R. Baraldi, D. Orban, A Levenberg−Marquardt method for nonsmooth regularized least squares, *SIAM J. Sci. Comput.*, **46** (2024), A2557–A2581. https://doi.org/10.1137/22M1538971

13. J. Y. Tang, J. C. Zhou, A Levenberg−Marquardt type algorithm with a Broyden-like update technique for solving nonlinear equations, *J. Comput. Appl. Math.* **460** (2025), 116401. https://doi.org/10.1016/j.cam.2024.116401

14. J. Y. Fan, Y. X. Yuan, On the quadratic convergence of the Levenberg−Marquardt method without nonsingularity assumption, *Computing*, **74** (2005), 23–39. https://doi.org/10.1007/s00607-004-0083-1

15. D. H. Li, M. Fukushima, A derivative-free line search and global convergence of Broyden-like method for nonlinear equations, *Optim. Methods Softw.*, **13** (2000), 181–201. https://doi.org/10.1080/10556780008805782

16. P. Jin, C. Ling, H. F. Shen, A smoothing Levenberg−Marquardt algorithm for semi-infinite programming, *Comput. Optim. Appl.*, **60** (2015), 675–695. https://doi.org/10.1007/s10589-014-9698-0

17. W. A. Liu, C. Y. Wang, A smoothing Levenberg−Marquardt method for generalized semi-infinite programming, *Comput. Appl. Math.*, **32** (2013), 89–105. https://doi.org/10.1007/s40314-013-0013-y

18. J. Y. Tang, Y. Chen, Smoothing functions and algorithm for nonsymmetric circular cone complementarity problems, *Appl. Math.*, **67** (2022), 209–231. https://doi.org/10.21136/AM.2021.0129-20

19. L. Qi, J. Sun, A nonsmooth version of Newton's method, *Math. Program.*, **58** (1993), 353–367. https://doi.org/10.1007/BF01581275

20. L. Y. Qi, X. T. Xiao, L. W. Zhang, A parameter-self-adjusting Levenberg−Marquardt method for solving nonsmooth equations, *J. Comput. Math.*, **34** (2016), 317–338. https://doi.org/10.4208/jcm.1512-m2015-0333