INTERNATIONAL JOURNAL OF
NUMERICAL ANALYSIS AND MODELING
Volume 21, Number 2, Pages 295–314

doi: 10.4208/ijnam2024-1012

MODIFIED NEWTON-NDSS METHOD FOR SOLVING NONLINEAR SYSTEM WITH COMPLEX SYMMETRIC JACOBIAN MATRICES

XIAOHUI YU AND QINGBIAO WU*

Abstract. An efficient iteration method is provided in this paper for solving a class of nonlinear systems whose Jacobian matrices are complex and symmetric. The modified Newton-NDSS method is developed and applied to the class of nonlinear systems by adopting the modified Newton method as the outer solver and a new double-step splitting (NDSS) iteration scheme as the inner solver. Additionally, we theoretically analyze the local convergent properties of the new method under the weaker Hölder conditions. Lastly, the new method is compared numerically with some existing ones and the numerical experiments solving the nonlinear equations demonstrate the superiority of the Newton-NDSS method.

Key words. Modified Newton-NDSS method, complex nonlinear systems, Hölder continuous condition, symmetric Jacobian matrix, convergence analysis.

1. Introduction

Consider the complex nonlinear systems with the following form

$$F(x) = 0, \tag{1}$$

with $F : \mathbb{D} \subset \mathbb{C}^n \to \mathbb{C}^n$ representing a nonlinear function. Further, the function F is defined on an open convex subset $\mathbb{D}$ of the n- dimensional complex linear space $\mathbb{C}^n$ and continuously differentiable. For the sake of solving the nonlinear systems with effectiveness, we first review the study of the solution technique to linear systems with complex matrices

$$Az = b, \quad A = W + iT \in \mathbb{C}^{n\times n}, \quad z, b \in \mathbb{C}^n. \tag{2}$$

Here the matrices $W, T \in \mathbb{R}^{n\times n}$ are real symmetric and positive semidefinite with at least one of them being positive definite. Here the matrices $W, T \in \mathbb{R}^{n\times n}$ are real symmetric and positive semidefinite with at least one being positive definite. Throughout this paper, $i = \sqrt{-1}$ defines the imaginary unit. Systems (2) appear in a variety of engineering applications and scientific computing, such as diffuse optical tomography, structural dynamics, and quantum mechanics. Readers can refer to the literature [1-3]. Up to now, researchers have made great efforts to seek rapid solution techniques for the above complex linear systems (2). From the very beginning, Bai et al. originated the classical Hermitian and skew-Hermitian splitting (HSS) [4] iteration method and its preconditioned form PHSS [5] method for semidefinite linear or positive definite systems. Afterward, the modified HSS (MHSS) method [6] iteration scheme were constructed by Bai et al. which greatly enhanced the computational efficiency of HSS iteration scheme. Whereafter, variations and improvements of these methods proliferated [7-11]. Especially, various numerical methods have been produced through internal and external iterative techniques. The solution of the large sparse positive definite system of nonlinear equations has

---

Received by the editors on April 19, 2023 and, accepted on January 31, 2024.
2000 *Mathematics Subject Classification.* 65B99, 65N12, 65N15.
*Corresponding author.

been studied by Bai et al. [12], who developed the Newton-HSS methods. For the large sparse systems with complex symmetric Jacobian matrices, Yang and Wu [13] applied the inexact Newton-MHSS method to them. Therewith, the research on large sparse nonlinear systems attracted substantial attention [14-18]. Easily stated, the following block system is equivalent to (2) and can avoid the operations on complex matrices, so that attracted a vast scale of interest.

$$\mathcal{A}x = \begin{bmatrix} W & -T \\ T & W \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} p \\ q \end{bmatrix}, \tag{3}$$

where $z = u + iv$ and $b = p + iq$. Bai et al. [19] introduced a block preconditioned MHSS (PMHSS) iteration method [19] and its alternating-directional versions [20] for the above linear system (3). Taking into account the excellent properties and efficient performance of SOR-like methods, over the recent years, the generalized SOR (GSOR) method [21], accelerated GSOR (AGSOR) method [22], and preconditioned GSOR (PGSOR) method [23] were applied to the two-by-two block linear system (3). A series of iteration schemes based on GSOR-like methods which can converge to the exact solution to complex nonlinear equations rapidly were proposed named modified Newton-GSOR method [24], modified Newton-AGSOR method [25], and modified Newton-PGSOR method [26]. A while back, a fixed-point iteration adding the asymptotical error (FPAE) scheme and its parameterized variant were structured by Xiao and Wang [27]. Then, a class of complex nonlinear systems has been solved by Zhang and Wu [28] using the Newton-FPAE method and the modified Newton-FPAE method. Recently, Huang [29] developed a new double-step splitting (NDSS) iteration method by taking advantage of two-step, parameter accelerating and preconditioning techniques. It has been proved that the NDSS iteration method demands mild convergence conditions and owns a faster convergence speed compared to some known iteration methods. These indicated the effectiveness and practicability of the NDSS method.

The aim of the present work is to formulate a fast and effective iterative method for solving complex nonlinear systems. Inspired by the excellent computing ability of the NDSS method compared with other algorithms for complex linear systems, we elaborate the modified Newton-NDSS (MN-NDSS) iteration method for the complex nonlinear systems by applying the modified Newton method as the outer solver and the NDSS method as the inner solver.

This paper is organized as follows. In the next section, we outline the modified Newton-NDSS iteration scheme for solving the complex nonlinear systems, including its algorithm and iterative formula. Section 3 is devoted to analyse the local convergent properties under the Hölder hypothesis for the new iteration methods. The results of numerical experiments presented in Section 4 support the theoretical findings and explain the superiority of the modified Newton-NDSS method. Finally, some conclusions are given in Section 5.

2. The modified Newton-NDSS method

First, throughout the paper, denote

$$\mathcal{A} = \begin{bmatrix} W & -T \\ T & W \end{bmatrix}, \quad x = \begin{bmatrix} u \\ v \end{bmatrix} \quad \text{and} \quad \tilde{b} = \begin{bmatrix} p \\ q \end{bmatrix},$$

with $W, T \in \mathbb{R}^{n\times n}$ being symmetric positive semidefinite and at least one of them being positive definite. Obviously, system (3) can be reformulated as

$$\tilde{\mathcal{A}}x = \begin{bmatrix} T & W \\ -W & T \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} q \\ -p \end{bmatrix}. \tag{4}$$

Adopt two block matrices P_α and P_β as the following form

$$P_\alpha = \begin{bmatrix} I & \alpha I \\ 0 & I \end{bmatrix}, \quad P_\beta = \begin{bmatrix} I & -\beta I \\ 0 & I \end{bmatrix}.$$

By multiplying P_α on both sides of (3) simultaneously, we get

(5)
$$P_\alpha \mathcal{A} \begin{bmatrix} u \\ v \end{bmatrix} := P_\alpha \mathcal{A} P_\alpha \begin{bmatrix} d_1 \\ e_1 \end{bmatrix} = \begin{bmatrix} W+\alpha T & 2\alpha W + \left(\alpha^2 - 1\right) T \\ T & W + \alpha T \end{bmatrix} \begin{bmatrix} d_1 \\ e_1 \end{bmatrix} = \begin{bmatrix} p + \alpha q \\ q \end{bmatrix}.$$

Similarly,

(6) $$P_\beta \tilde{\mathcal{A}} \begin{bmatrix} u \\ v \end{bmatrix} := P_\beta \tilde{\mathcal{A}} P_\beta \begin{bmatrix} d_2 \\ e_2 \end{bmatrix} = \begin{bmatrix} \beta W + T & (1-\beta^2)W - 2\beta T \\ -W & \beta W + T \end{bmatrix} \begin{bmatrix} d_2 \\ e_2 \end{bmatrix} = \begin{bmatrix} q + \beta p \\ -p \end{bmatrix}.$$

For the above matrices, we have the decomposition

$$\begin{bmatrix} W+\alpha T & 2\alpha W + \left(\alpha^2 - 1\right) T \\ T & W + \alpha T \end{bmatrix} = \begin{bmatrix} W+\alpha T & 0 \\ T & W + \alpha T \end{bmatrix} - \begin{bmatrix} 0 & (1-\alpha^2)T - 2\alpha W \\ 0 & 0 \end{bmatrix},$$

and

$$\begin{bmatrix} \beta W + T & (1-\beta^2)W - 2\beta T \\ -W & \beta W + T \end{bmatrix} = \begin{bmatrix} \beta W + T & 0 \\ -W & \beta W + T \end{bmatrix} - \begin{bmatrix} 0 & -(1-\beta^2)W + 2\beta T \\ 0 & 0 \end{bmatrix}.$$

In accordance with the matrix splitting, the following fixed-point equations are yielded

$$\begin{cases} \begin{bmatrix} W+\alpha T & 0 \\ T & W + \alpha T \end{bmatrix} \begin{bmatrix} d_1 \\ e_1 \end{bmatrix} = \begin{bmatrix} 0 & (1-\alpha^2)T - 2\alpha W \\ 0 & 0 \end{bmatrix} \begin{bmatrix} d_1 \\ e_1 \end{bmatrix} + \begin{bmatrix} p + \alpha q \\ q \end{bmatrix}, \\ \begin{bmatrix} \beta W + T & 0 \\ -W & \beta W + T \end{bmatrix} \begin{bmatrix} d_2 \\ e_2 \end{bmatrix} = \begin{bmatrix} 0 & -(1-\beta^2)W + 2\beta T \\ 0 & 0 \end{bmatrix} \begin{bmatrix} d_2 \\ e_2 \end{bmatrix} + \begin{bmatrix} q + \beta p \\ -p \end{bmatrix}. \end{cases}$$

For the convenience of expression, we let $\tilde{W} = 2\alpha W - (1-\alpha^2)T$ and $\tilde{T} = 2\beta T - (1-\beta^2)W$. According to the above procedure, the NDSS iteration method can be derived.

The NDSS iteration method: Let real parameters α, β are positive and give two arbitrary original guess $u_0, v_0 \in \mathbb{R}^n$. Determine $u_{k+1}, v_{k+1},\ k = 0, 1, 2, \ldots$ according to the algorithm below until iterative sequences satisfy the stopping criteria:

(7)
$$\begin{cases} y_k = u_k - \alpha v_k, \quad w_k = v_k, \\ (W+\alpha T) y_{k+\frac{1}{2}} = \left[\left(1-\alpha^2\right) T - 2\alpha W\right] w_k + p + \alpha q, \\ (W+\alpha T) w_{k+\frac{1}{2}} = -T y_{k+\frac{1}{2}} + q, \\ (\beta W + T) y_{k+1} = \left[2\beta T - \left(1-\beta^2\right) W\right] w_{k+\frac{1}{2}} + q + \beta p, \\ (\beta W + T) w_{k+1} = W y_{k+1} - p, \\ u_{k+1} = y_{k+1} - \beta w_{k+1}, v_{k+1} = w_{k+1}, \end{cases}$$

In other words, the NDSS method can also be summarized as

$$x_{k+1} = H_{\alpha,\beta} x_k + G_{\alpha,\beta} \widetilde{b}, \quad k = 0, 1, 2, ...,$$

equivalently, or

(8)
$$x_{k+1} = H_{\alpha,\beta}^{k+1} x_0 + \sum_{j=0}^{k} H_{\alpha,\omega}^{j} G_{\alpha,\beta} \widetilde{b}, \quad k = 0, 1, 2, ...,$$

where

$$H_{\alpha,\beta} = \begin{bmatrix} 0 & (\beta W + T)^{-1} \tilde{T} (W+\alpha T)^{-1} T (W+\alpha T)^{-1} \tilde{W} \\ 0 & (\beta W + T)^{-1} W (\beta W + T)^{-1} \tilde{T} (W+\alpha T)^{-1} T (W + \alpha T)^{-1} \tilde{W} \end{bmatrix}.$$

Naturally, $H_{\alpha,\beta}$ gives the iteration matrix of the NDSS method. We will use the notations $Re(z)$ and $Im(z)$ to denote the real and imaginary parts of a complex matrix or vector z respectively for the remaining pages. In accordance with the symbol in Section 1, the following relationship is proper.

$$x = \begin{bmatrix} Re(z) \\ Im(z) \end{bmatrix}.$$

Let $p(x) = Re(F(x))$ and $q(x) = Im(F(x))$, so that $F(x) = p(x) + iq(x)$ automatically. Suppose the following decomposition form about the Jacobian matrix $F'(x)$ holds

$$F'(z) = W(x) + iT(x),$$

where $W(x) = Re(F'(x)) \in \mathbb{R}^{n\times n}$, $T(x) = Im(F'(x)) \in \mathbb{R}^{n\times n}$ are real symmetric and positive semidefinite and at least one of them is positive definite. Same as the previous iteration scheme, give a definition that

$$\mathcal{A}(x) = \begin{bmatrix} W(x) & -T(x) \\ T(x) & W(x) \end{bmatrix} \quad \text{and} \quad \mathcal{P}(x) = \begin{bmatrix} p(x) \\ q(x) \end{bmatrix}.$$

For ease of expression, we give the following notations

$$\begin{cases} P_1(\beta, x) = (\beta W(x) + T(x))^{-1}, \\ P_2(\beta, x) = 2\beta T(x) - (1-\beta^2)W(x), \\ P_3(\alpha, x) = (W(x) + \alpha T(x))^{-1}T(x)(W(x) + \alpha T(x))^{-1}, \\ P_4(\alpha, x) = 2\alpha W(x) - (1-\alpha^2)T(x). \end{cases}$$

The NDSS method is combined with the modified Newton method to construct a new iteration scheme for solving the complex nonlinear systems. We name the new iteration scheme the modified Newton-NDSS method in which the two linear systems below are solved by the NDSS iteration method

$$\begin{aligned} &\mathcal{A}(x_k)d_k = -\mathcal{P}(x_k), \quad x_{k+\frac{1}{2}} = x_k + d_k, \\ &\mathcal{A}(x_k)h_k = -\mathcal{P}(x_{k+\frac{1}{2}}), \quad x_{k+1} = x_{k+\frac{1}{2}} + h_k. \end{aligned} \tag{9}$$

Through some substitution calculation to Eq.(8), d_{k,l_k} and h_{k,m_k} have the following expressions

$$d_{k,l_k} = -\sum_{j=0}^{l_k-1} H_{\alpha,\beta}(x_k)^j G_{\alpha,\beta}(x_k)\mathcal{P}(x_k),$$

$$h_{k,m_k} = -\sum_{j=0}^{m_k-1} H_{\alpha,\omega}(x_k)^j G_{\alpha,\beta}(x_k)\mathcal{P}(x_{k+\frac{1}{2}}),$$

where

$$H_{\alpha,\beta}(x) = \begin{bmatrix} 0 & P_1(\beta,x)P_2(\beta,x)P_3(\alpha,x)P_4(\alpha,x) \\ 0 & P_1(\beta,x)W(x)P_1(\beta,x)P_2(\beta,x)P_3(\alpha,x)P_4(\alpha,x) \end{bmatrix}.$$

It's easy to rewrite the MN-NDSS method as its equivalent form

$$\begin{cases} x_{k+\frac{1}{2}} = x_k - \sum_{j=0}^{l_k-1} H_{\alpha,\beta}(x_k)^j G_{\alpha,\beta}(z_k)\mathcal{P}(x_k), \\ x_{k+1} = x_{k+\frac{1}{2}} - \sum_{j=0}^{m_k-1} H_{\alpha,\beta}(x_k)^j G_{\alpha,\beta}(x_k)\mathcal{P}(x_{k+\frac{1}{2}}), \end{cases} \quad k = 0, 1, 2, ... \tag{10}$$

The modified Newton-NDSS iteration method

1. Set $x_0 = [u_0^T, v_0^T]^T$ where $u_0, v_0 \in \mathbb{D}$ are two given original vectors.
2. For k=0,1,2,...until $\|\mathcal{P}(x_k)\| \le tol\|\mathcal{P}(x_0)\|$ execute:

2.1 Let $d_{k,0} = h_{k,0} = 0$, $v_{k,0} = w_{k,0} = t_{k,0} = s_{k,0} = 0$.
2.2 By the NDSS method, solve the first equation in (9) for $l = 0, 1, ...l_k - 1$:

$$\begin{cases} (W(x_k) + \alpha T(x_k))y_{k,l+\frac{1}{2}} = \left[\left(1-\alpha^2\right)T(x_k) - 2\alpha W(x_k)\right] w_{k,l} \\ \qquad\qquad +p(x_k) + \alpha q(x_k), \\ (W(x_k) + \alpha T(x_k))w_{k,l+\frac{1}{2}} = -T(x_k)y_{k,l+\frac{1}{2}} + q(x_k), \\ (\beta W(x_k) + T(x_k))y_{k,l+1} = \left[2\beta T(x_k) - \left(1-\beta^2\right)W(x_k)\right] w_{k,l+\frac{1}{2}} \\ \qquad\qquad +q(x_k) + \beta p(x_k), \\ (\beta W(x_k) + T(x_k))w_{k,l+1} = W(x_k)y_{k,l+1} - p(x_k), \end{cases}$$

and let $d_{k,l+1} = [(y_{k,l+1} - \beta w_{k,l+1})^T, w_{k,l+1}^T]^T$ to get d_{k,l_k} such that

$$\|\mathcal{P}(x_k) + \mathcal{A}(x_k)d_{k,l_k}\| \leq \eta_k\|\mathcal{P}(x_k)\|, \quad \text{for} \quad \eta_k \in [0, 1).$$

2.3 Let $x_{k+\frac{1}{2}} = x_k + d_{k,l_k}$.
2.4 Obtain $\mathcal{P}(x_{k+\frac{1}{2}})$.
2.5 By the NDSS method, solve the second equation in (9) for $m = 0, 1, ..., m_k - 1$:

$$\begin{cases} (W(x_k) + \alpha T(x_k))t_{k,m+\frac{1}{2}} = \left[\left(1-\alpha^2\right)T(x_k) - 2\alpha W(x_k)\right] s_{k,m} \\ \qquad\qquad +p(x_{k+\frac{1}{2}}) + \alpha q(x_{k+\frac{1}{2}}), \\ (W(x_k) + \alpha T(x_k))s_{k,m+\frac{1}{2}} = -T(x_k)t_{k,m+\frac{1}{2}} + q(x_{k+\frac{1}{2}}), \\ (\beta W(x_k) + T(x_k))t_{k,m+1} = \left[2\beta T(x_k) - \left(1-\beta^2\right)W(x_k)\right] s_{k,m+\frac{1}{2}} \\ \qquad\qquad +q(x_{k+\frac{1}{2}}) + \beta p(x_{k+\frac{1}{2}}), \\ (\beta W(x_k) + T(x_k))s_{k,m+1} = W(x_k)t_{k,m+1} - p(x_{k+\frac{1}{2}}), \end{cases}$$

and let $h_{k,m+1} = [(t_{k,m+1} - \beta s_{k,m+1})^T, s_{k,m+1}^T]^T$ to get h_{k,m_k} which enables

$$\|\mathcal{P}(x_{k+\frac{1}{2}}) + \mathcal{A}(x_k)h_{k,m_k}\| \leq \widetilde{\eta_k}\|\mathcal{P}(x_{k+\frac{1}{2}})\|, \quad \text{for} \quad \widetilde{\eta_k} \in [0, 1).$$

2.6 Let $x_{k+1} = x_{k+\frac{1}{2}} + h_{k,m_k}$.
Also, the MN-NDSS method can be reformulated as

$$\begin{cases} x_{k+\frac{1}{2}} = x_k - (I - H_{\alpha,\omega}(x_k)^{l_k})\mathcal{A}(x_k)^{-1}\mathcal{P}(x_k), \\ x_{k+1} = x_{k+\frac{1}{2}} - (I - H_{\alpha,\omega}(x_k)^{m_k})\mathcal{A}(x_k)^{-1}\mathcal{P}(x_{k+\frac{1}{2}}). \end{cases} \tag{11}$$

3. Local convergence property of the MN-NDSS method

Lemma 3.1. *(Banach Lemma) Let the matrices $M, N \in \mathbb{C}^{n\times n}$ satisfy $\|I - MN\| \leq 1$. Then M and N are non-singular. Especially,*

$$\|M^{-1}\| \leq \frac{\|N\|}{1 - \|I - NM\|}, \quad \|N^{-1}\| \leq \frac{\|M\|}{1 - \|I - NM\|},$$

and

$$\|M^{-1} - N\| \leq \frac{\|N\|\|I - NM\|}{1 - \|I - NM\|}, \quad \|M - N^{-1}\| \leq \frac{\|M\|\|I - NM\|}{1 - \|I - NM\|}.$$

Suppose the nonlinear function $F : \mathbb{D} \subset \mathbb{C}^n \to \mathbb{C}^n$ has the G-differentiability in the open field $\mathbb{N}_0 \in \mathbb{D}$ and $F'(z)$ is continuous. The nonlinear system (1) has an exact solution x_*. Next, assume the following conditions are met to research the local convergent properties of the MN-NDSS iteration method.

Assumption 3.1 Suppose $\mathbb{N}(x_*, r)$ is an open sphere with the center at x_* and radius r. And assume that these following aspects are satisfied for each $x \in \mathbb{N}(x_*, r) \subset$

$\mathbb{N}_0$.
(A1) There exist positive constants ξ, δ and γ satisfing

$$\max\{\|W(x_*)\|, \|T(x_*)\|\} \le \xi, \quad \max\{\|W(x_*)^{-1}\|, \|T(x_*)^{-1}\|\} \le \delta$$
$$\text{and} \quad \|\mathcal{A}(x_*)^{-1}\| \le \gamma.$$

(A2) For some $p \in (0, 1]$, there are nonnegative constants H_w and H_t satisfying the Hölder conditions

$$\|W(x_*) - W(x)\| \le H_w \|x_* - x\|^p,$$

$$\|T(x_*) - T(x)\| \le H_t \|x_* - x\|^p.$$

Lemma 3.2. *Let* $r_0 = \min\left\{\left(\frac{\alpha+1}{\delta(H_w+\alpha H_t)}\right)^{\frac{1}{p}}, \left(\frac{\beta+1}{\delta(\beta H_w+H_t)}\right)^{\frac{1}{p}}, \frac{1}{(\gamma H)^{\frac{1}{p}}}\right\}$, *if* $r \in (0, r_0)$, $p \in (0, 1]$, *for all* $x, y \in \mathbb{N}(x_*, r)$, *the following inverse matrices exist and the inequalities hold:*
(1): $\|(\beta W(x) + T(x))^{-1}\| \le \dfrac{\delta}{\beta + 1 - \delta(\beta H_w + H_t)\|x - x_*\|^p}$,
(2): $\|(W(x) + \alpha T(x))^{-1}\| \le \dfrac{\delta}{\alpha + 1 - \delta(H_w + \alpha H_t)\|x - x_*\|^p}$,
(3): $\|\mathcal{P}(x)\| \le \dfrac{H}{p+1}\|x - x_*\|^{p+1} + 2\beta\|x - x_*\|$,
(4): $\|x - x_* - \mathcal{A}(y)^{-1}\mathcal{P}(x)\| \le \dfrac{\gamma}{1 - \gamma H\|y - x_*\|^p}\left(\dfrac{H}{p+1}\|x - x_*\|^p + H\|y - x_*\|^p\right)$
$\times\|x - x_*\|$.

Proof. First, we know

$$\begin{aligned}\|(\beta W(x_*) + T(x_*))^{-1}\| =& \frac{1}{\lambda_{min}(\beta W(x_*) + T(x_*))}\\ \le& \frac{1}{\beta\frac{1}{\|W(x_*)^{-1}\|} + \frac{1}{\|T(x_*)^{-1}\|}} \le \frac{\delta}{\beta+1}.\end{aligned}$$

From a similar discussion,

$$\|(W(x_*) + \alpha T(x_*))^{-1}\| \le \frac{\delta}{\alpha+1}.$$

Therefore, according to Lemma 3.1 and the condition $r \in (0, r_0)$, we have

$$\begin{aligned}&\|(\beta W(x) + T(x))^{-1}\|\\ &\le \frac{\|(\beta W(x_*) + T(x_*))^{-1}\|}{1 - \|(\beta W(x_*) + T(x_*))^{-1}[(\beta W(x) + T(x)) - (\beta W(x_*) + T(x_*))]\|}\\ &\le \frac{\delta}{\beta + 1 - \delta(\beta H_w + H_t)\|x - x_*\|^p},\end{aligned}$$

$$\begin{aligned}&\|(W(x) + \alpha T(x))^{-1}\|\\ &\le \frac{\|(W(x_*) + \alpha T(x_*))^{-1}\|}{1 - \|(W(x_*) + \alpha T(x_*))^{-1}[(W(x) + \alpha T(x)) - (W(x_*) + \alpha T(x_*))]\|}\\ &\le \frac{\delta}{\alpha + 1 - \delta(H_w + \alpha H_t)\|x - x_*\|^p}.\end{aligned}$$

Consequently, the first and second inequalities are correct. By performing some calculations

$$\begin{aligned}\mathcal{P}(x) &= \mathcal{P}(x) - \mathcal{P}(x_*) - \mathcal{A}(x_*)(x - x_*) + \mathcal{A}(x_*)(x - x_*)\\ &= \int_0^1 [\mathcal{A}(x_* + t(x - x_*)) - \mathcal{A}(x_*)]dt(x - x_*) + \mathcal{A}(x_*)(x - x_*).\end{aligned}$$

Here, the third inequality can be derived as follows.

$$\begin{aligned}\|\mathcal{P}(x)\| &\le \int_0^1 \|\mathcal{A}(x_* + t(x - x_*)) - \mathcal{A}(x_*)\|dt\|(x - x_*)\| + \|\mathcal{A}(x_*)\|\|(x - x_*)\|\\ &\le \frac{H}{p+1}\|x - x_*\|^{p+1} + 2\beta\|x - x_*\|.\end{aligned}$$

Due to assumption (A2)

$$\begin{aligned}\|\mathcal{A}(x_*) - \mathcal{A}(x)\| =&\left\| \begin{pmatrix} W(x_*) & -T(x_*) \\ T(x_*) & W(z_*) \end{pmatrix} - \begin{pmatrix} W(x) & -T(z) \\ T(z) & W(z) \end{pmatrix} \right\|\\ \le&\left\| \begin{pmatrix} W(x_*) - W(x) & 0 \\ 0 & W(x_*) - W(x) \end{pmatrix} \right\|\\ &+ \left\| \begin{pmatrix} T(x_*) - T(x) & 0 \\ 0 & T(x_*) - T(x) \end{pmatrix} \right\|\\ \le&(H_w + H_t)\|x_* - x\|^p\\ =&H\|x_* - x\|^p.\end{aligned}$$

Since $r < 1/(\gamma H)^{\frac{1}{p}}$, it holds that

$$\|\mathcal{A}(x)^{-1}\| \le \frac{\|\mathcal{A}(x_*)^{-1}\|}{1 - \|\mathcal{A}(x_*)^{-1}(\mathcal{A}(x_*) - \mathcal{A}(x))\|} \le \frac{\gamma}{1 - \gamma H\|x_* - x\|^p}.$$

Obviously,

$$\begin{aligned}&x - x_* - \mathcal{A}(y)^{-1}\mathcal{P}(x)\\ &\quad= - \mathcal{A}(y)^{-1}(\mathcal{P}(x) - \mathcal{P}(x_*) - \mathcal{A}(x_*)(x - x_*) + \mathcal{A}(x_*)(x - x_*) - \mathcal{A}(y)(x - x_*))\\ &\quad= - \mathcal{A}(y)^{-1}[\int_0^1 \mathcal{A}(x_* + t(x - x_*)) - \mathcal{A}(x_*)dt + \mathcal{A}(x_*) - \mathcal{A}(y)](x - x_*).\end{aligned}$$

Hence,

$$\begin{aligned}&\|x - x_* - \mathcal{A}(y)^{-1}\mathcal{P}(y)\|\\ &\quad\le\|\mathcal{A}(y)^{-1}\|[\int_0^1 \|\mathcal{A}(x_* + t(x - x_*)) - \mathcal{A}(x_*)\|dt + \|\mathcal{A}(x_*) - \mathcal{A}(y)\|]\|(x - x_*)\|\\ &\quad\le\frac{\gamma}{1 - \gamma H\|y - x_*\|^p}\left(\frac{H}{p+1}\|x - x_*\|^p + H\|y - x_*\|^p\right)\|x - x_*\|,\end{aligned}$$

which completes the proof. □

In the theorem that follows, define

$$R(r^p) = \beta + 1 - \delta(\beta H_w + H_t)r^p, \quad Q(r^p) = \alpha + 1 - \delta(H_w + \alpha H_t)r^p, \quad d = H_w + \alpha H_t,$$

$$S = 2\alpha H_w + |1 - \alpha^2|H_t,\ L = |1 - \beta^2|H_w + 2\beta H_t,\ f(t) = (|1 - t^2| + 2t)\xi,\ e = \beta H_w + H_t,$$

and

$$\begin{aligned}V(r^p) =&\frac{\delta^3\left[\xi f(\alpha)Q^2(r^p)(\delta e f(\beta)+(\beta+1)L)+(\beta+1)(\alpha+1)^2S(Lr^p+f(\beta))\right]r^p}{(\beta+1)(\alpha+1)^2R(r^p)Q^2(r^p)}\\&+\frac{\delta^3\left[Lr^p+f(\beta)\right]\left[\delta\xi e f(\alpha)(\alpha+1+Q(r^p))+(\alpha+1)^2H_t(Sr^p+f(\alpha))\right]r^p}{(\alpha+1)^2R(r^p)Q^2(r^p)}.\end{aligned}$$

Theorem 3.1. *Denote $r_p=\min\{r_1,r_2\}$, where r_1 equals the minimum positive solution of the following equation*

$$\begin{aligned}&\delta^4\xi f(\alpha)f(\beta)\left[\delta\xi e+(\beta+1)H_w\right]r^p\\=&(\alpha+1)^2(\beta+1)^2\left[\tau\theta R(r^p)-V(r^p)R(r^p)-\delta(H_w r^p+\xi)V(r^p)\right],\end{aligned}$$

and

$$r_2=\left(\frac{1-2\gamma\beta((1+\tau)\theta)^{\mu_*}}{4\gamma H}\right)^{\frac{1}{p}},$$

with $\mu_*=\min\{m_*,l_*\}, m_*=\liminf_{k\to\infty}m_k, l_*=\liminf_{k\to+\infty}l_k$, and the constant μ_* satisfies

$$\mu_*>\left\lfloor-\frac{\ln(2\xi\gamma)}{\ln((1+\tau)\theta)}\right\rfloor.$$

In the latest-written formula, the symbol $\lfloor x\rfloor$ represents the smallest integer no less than the corresponding real number x, and $\tau\in(0,\frac{1-\theta}{\theta})$ is a prescribed positive constant, besides

$$\theta\equiv\theta(\alpha;x_*)=\|H_{\alpha,\beta}(x_*)\|<1.$$

Under the conditions of Lemma 3.2, when $r\in(0,r_p)$, for any $x_0\in\mathbb{N}(x_*,r)$, $s\in(0,r)$, it can be derived that

$$\begin{aligned}\|H_{\alpha,\beta}(x)\|&\le(\tau+1)\theta,\\g(s,\lambda)&\le g(r_p,\mu_*)<1,\end{aligned}$$

where

$$g(s,\lambda)=\frac{\gamma}{1-\gamma Hs^p}\left[3Hs^p+2\beta((1+\tau)\theta)^\lambda\right]\quad\text{with}\quad\lambda>\mu_*.$$

Proof. Let

$$\begin{aligned}P_{\alpha,\beta}(x)=&(\beta W(x)+T(x))^{-1}\left[2\beta T(x)-\left(1-\beta^2\right)W(x)\right]\\&\times(W(x)+\alpha T(x))^{-1}T(x)(W(x)+\alpha T(x))^{-1}\left[2\alpha W(x)-\left(1-\alpha^2\right)T(x)\right],\end{aligned}$$

and

$$\begin{aligned}L_{\alpha,\beta}(x)=&(\beta W(x)+T(x))^{-1}W(x)(\beta W(x)+T(x))^{-1}\left[2\beta T(x)-\left(1-\beta^2\right)W(x)\right]\\&\times(W(x)+\alpha T(x))^{-1}T(x)(W(x)+\alpha T(x))^{-1}\left[2\alpha W(x)-\left(1-\alpha^2\right)T(x)\right].\end{aligned}$$

By the Hölder condition and Lemma 3.2, we have

$$\begin{aligned}&\|P_1(\beta,x)-P_1(\beta,x_*)\|\\&=\|(\beta W(x)+T(x))^{-1}-(\beta W(x_*)+T(x_*))^{-1}\|\\&\le\|(\beta W(x)+T(x))^{-1}\|\|(\beta W(x)+T(x))-(\beta W(x_*)+T(x_*))\|\\&\times\|(\beta W(x_*)+T(x_*))^{-1}\|\\&\le\frac{\delta^2(\beta H_w+H_t)\|x-x_*\|^p}{(\beta+1)^2-\delta(\beta+1)(\beta H_w+H_t)\|x-x_*\|^p}.\end{aligned}$$

Also, in accordance with Lemma 3.2, we can easily get

$$
\begin{aligned}
&\|P_3(\alpha,x)-P_3(\alpha,x_*)\|\\
\leq&\|(W(x)+\alpha T(x))^{-1}-(W(x_*)+\alpha T(x_*))^{-1}\|\|T(x_*)\|\|(W(x_*)+\alpha T(x_*))^{-1}\|\\
&+\|(W(x)+\alpha T(x))^{-1}\|\|T(x)-T(x_*)\|\|(W(x_*)+\alpha T(x_*))^{-1}\|\\
&+\|(W(x)+\alpha T(x))^{-1}\|\|T(x)\|\|(W(x)+\alpha T(x))^{-1}-(W(x_*)+\alpha T(x_*))^{-1}\|\\
\leq&\frac{\delta^3\xi(H_w+\alpha H_t)\|x-x_*\|^p}{(\alpha+1)^3-\delta(\alpha+1)^2(H_w+\alpha H_t)\|x-x_*\|^p}\\
&+\frac{\delta^2 H_t\|x-x_*\|^p}{(\alpha+1)^2-\delta(\alpha+1)(H_w+\alpha H_t)\|x-x_*\|^p}\\
&+\frac{\delta^3(H_t\|x-x_*\|^p+\xi)(H_w+\alpha H_t)\|x-x_*\|^p}{(\alpha+1)[\alpha+1-\delta(H_w+\alpha H_t)\|x-x_*\|^p]^2}\\
=&\Big\{\delta^2\left[2\delta(\alpha+1)\xi(H_w+\alpha H_t)+(\alpha+1)^2H_t-\delta^2\xi(H_w+\alpha H_t)^2\|x-x_*\|^p\right]\\
&\times\|x-x_*\|^p\Big\}\Big/\Big\{(\alpha+1)^2[\alpha+1-\delta(H_w+\alpha H_t)\|x-x_*\|^p]^2\Big\}.
\end{aligned}
$$

Based on the previous step and Assumption 3.1, it can be derived that

$$
\begin{aligned}
&\|P_{\alpha,\beta}(x)-P_{\alpha,\beta}(x_*)\|\\
\leq&\|P_1(\beta,x)-P_1(\beta,x_*)\|\|P_2(\beta,x_*)\|\|P_3(\alpha,x_*)\|\|P_4(\alpha,x_*)\|\\
&+\|P_1(\beta,x)\|\|P_2(\beta,x)-P_2(\beta,x_*)\|\|P_3(\alpha,x_*)\|\|P_4(\alpha,x_*)\|\\
&+\|P_1(\beta,x)\|\|P_2(\beta,x)\|\|P_3(\alpha,x)-P_3(\alpha,x_*)\|\|P_4(\alpha,x_*)\|\\
&+\|P_1(\beta,x)\|\|P_2(\beta,x)\|\|P_3(\alpha,x)\|\|P_4(\alpha,x)-P_4(\alpha,x_*)\|\\
\leq&\frac{\delta^4\xi ef(\beta)f(\alpha)\|x-x_*\|^p}{(\beta+1)(\alpha+1)^2R(\|x-x_*\|^p)}+\frac{\delta^3\xi f(\alpha)L\|x-x_*\|^p}{(\alpha+1)^2R(\|x-x_*\|^p)}\\
&+\frac{\delta^3(H_t\|x-x_*\|^p+\xi)\left[L\|x-x_*\|^p+f(\beta)\right]S\|x-x_*\|^p}{Q^2(\|x-x_*\|^p)R(\|x-x_*\|^p)}\\
&+\Big\{\delta f(\alpha)\left[L\|x-x_*\|^p+f(\beta)\right]\big[\delta^3\xi d\|x-x_*\|^p(\alpha+1+Q(\|x-x_*\|^p))\\
&+\delta^2(\alpha+1)^2H_t\|x-x_*\|^p\big]\Big\}\Big/\Big\{(\alpha+1)^2R(\|x-x_*\|^p)Q^2(\|x-x_*\|^p)\Big\}\\
=&\Big\{\delta^3\left[\xi f(\alpha)Q^2(\|x-x_*\|^p)(\delta ef(\beta)+(\beta+1)L)+(\beta+1)(\alpha+1)^2S(L\|x-x_*\|^p\right.\\
&\left.+f(\beta))\right]\|x-x_*\|^p\Big\}\Big/\Big\{(\beta+1)(\alpha+1)^2R(\|x-x_*\|^p)Q^2(\|x-x_*\|^p)\Big\}\\
&+\Big\{\delta^3\left[L\|x-x_*\|^p+f(\beta)\right]\big[\delta\xi df(\alpha)(\alpha+1+Q(\|x-x_*\|^p))+(\alpha+1)^2H_t\\
&\times(S\ |x-x_*\|^p+f(\alpha))\big]\|x-x_*\|^p\Big\}\Big/\Big\{(\alpha+1)^2R(\|x-x_*\|^p)Q^2(\|x-x_*\|^p)\Big\}\\
=&V(\|x-x_*\|^p).
\end{aligned}
$$

Likewise, we can get

$$\begin{aligned}
&\|L_{\alpha,\beta}(x) - L_{\alpha,\beta}(x_*)\| \\
\leq&\|P_1(\beta, x) - P_1(\beta, x_*)\|W(x_*)\|\|P_{\alpha,\beta}(x_*)\| \\
&+ \|P_1(\beta, x)\|\|W(x) - W(x_*)\|\|P_{\alpha,\beta}(x_*)\|\| \\
&+ \|P_1(\beta, x)\|\|W(x)\|\|P_{\alpha,\beta}(x) - P_{\alpha,\beta}(x_*)\| \\
\leq&\frac{\delta^5\xi^2 f(\alpha)f(\beta)e\|x - x_*\|^p}{(\beta+1)^2(\alpha+1)^2 R(\|x - x_*\|^p)} + \frac{\delta^4\xi f(\alpha)f(\beta)H_w\|x - x_*\|^p}{(\beta+1)(\alpha+1)^2 R(\|x - x_*\|^p)} \\
&+ \frac{\delta(H_w\|x - x_*\|^p + \xi)}{R(\|x - x_*\|^p))} V(\|x - x_*\|^p). \\
=&\frac{\delta^4\xi f(\alpha)f(\beta)\left[\delta\xi e + (\beta+1)H_w\right]\|x - x_*\|^p}{(\beta+1)^2(\alpha+1)^2 R(\|x - x_*\|^p)} \\
&+ \frac{\delta(H_w\|x - x_*\|^p + \xi)}{R(\|x - x_*\|^p))} \times V(\|x - x_*\|^p).
\end{aligned}$$

According to the nature of the block matrix, the following inequality can be deduced.

$$\begin{aligned}
&\|H_{\alpha,\beta}(x) - H_{\alpha,\beta}(x_*)\| \\
\leq&\|P_{\alpha,\beta}(x) - P_{\alpha,\beta}(x_*)\| + \|L_{\alpha,\beta}(x) - L_{\alpha,\beta}(x_*)\| \\
\leq&V(\|x - x_*\|^p) + \frac{\delta^4\xi f(\alpha)f(\beta)\left[\delta\xi e + (\beta+1)H_w\right]\|x - x_*\|^p}{(\beta+1)^2(\alpha+1)^2 R(\|x - x_*\|^p)} \\
&+ \frac{\delta V(\|x - x_*\|^p)(H_w\|x - x_*\|^p + \xi)}{R(\|x - x_*\|^p))}.
\end{aligned}$$

On account of $r < r_0$, we have

$$R(\|x - x_*\|^p) > 0.$$

Besides, $r < r_1$ hints that

$$\begin{aligned}
V(\|x - x_*\|^p) &+ \frac{\delta^4\xi f(\alpha)f(\beta)\left[\delta\xi e + (\beta+1)H_w\right]\|x - x_*\|^p}{(\beta+1)^2(\alpha+1)^2 R(\|x - x_*\|^p)} \\
&+ \frac{\delta V(\|x - x_*\|^p)(H_w\|x - x_*\|^p + \xi)}{R(\|x - x_*\|^p))} < \tau\theta.
\end{aligned}$$

Hence,

$$\begin{aligned}
&\|H_{\alpha,\omega}(x)\| \\
\leq&\|H_{\alpha,\omega}(x) - H_{\alpha,\omega}(x_*)\| + \|H_{\alpha,\omega}(x_*)\| \\
\leq&V(\|x - x_*\|^p) + \frac{\delta^4\xi f(\alpha)f(\beta)\left[\delta\xi e + (\beta+1)H_w\right]\|x - x_*\|^p}{(\beta+1)^2(\alpha+1)^2 R(\|x - x_*\|^p)} \\
&+ \frac{\delta V(\|z - z_*\|^p)(H_w\|x - x_*\|^p + \xi)}{R(\|x - x_*\|^p))} + \theta \\
\leq&(\tau+1)\theta.
\end{aligned}$$

Since $\tau < \frac{1-\theta}{\theta}$ and $r < \min\{r_1, r_2\}$ then for any $x \in \mathbb{N}(x_*, r)$, we find

$$\rho(H_{\alpha,\beta}(x)) \leq \|H_{\alpha,\beta}(x)\| \leq (\tau+1)\theta < 1.$$

Due to $(\tau+1)\theta < 1$, we know that the function $g(s,\lambda)$ is strictly monotone decreasing about λ. By taking the derivative with respect to s, it's not hard to get the following relationship

$$\frac{\partial g(s,\lambda)}{\partial s} = \frac{\gamma H p s^{p-1}[3+2\gamma\beta((\tau+1)\theta)^{\lambda}]}{[1-\gamma H s^p]^2} > 0.$$

From the above formula, the function $g(s,\lambda)$ has monotonically increasing properties concerning s. Hence, the following equation holds when $x_k \in \mathbb{N}(x_*, r)$

$$g(\|x_k - x_*\|, l_k) < \frac{\gamma}{1-\gamma H r^p}\left[3Hr^p + 2\beta((1+\tau)\theta)^{\mu_*}\right] = g(r, \mu_*).$$

On account of $r < r_2$, it is clear that $g(\|x_k - x_*\|, l_k) < g(r, \mu_*) < 1$. □

Theorem 3.2. *Allowing the conditions of Lemma 3.2 and Theorem 3.1, for any $x_0 \in N(x_*, r)$ and any positive integer sequences $\{l_k\}_{k=0}^{\infty}, \{m_k\}_{k=0}^{\infty}$, the iteration solution sequence $\{x_k\}_{k=0}^{\infty}$ produced by the MN-NDSS method is well-defined and convergent to the exact solution x_*. Furthermore, the following inequality holds*

$$\lim_{k\to\infty} \sup \|x_k - x_*\|^{\frac{1}{k}} \leq g\left(r_p, \mu_*\right)^2. \tag{12}$$

Proof. In view of Lemma 3.2 and Eq.(11), we can illustrate that the residual error $\|x_{k+\frac{1}{2}} - x_*\|$ satisfy the following inequality.

$$\begin{aligned}
\|x_{k+\frac{1}{2}} - x_*\| =& \|x_k - x_* - (I - H_{\alpha,\omega}(x_k)^{l_k})\mathcal{A}(x_k)^{-1}\mathcal{P}(x_k)\| \\
\leq& \|x_k - x_* - \mathcal{A}(z_k)^{-1}\mathcal{P}(z_k)\| + \|H_{\alpha,\omega}(x_k)\|^{l_k}\|\|\mathcal{A}(x_k)^{-1}\mathcal{P}(x_k)\| \\
\leq& \frac{\gamma}{1-\gamma H\|x_k - x_*\|^p}\left(\frac{H}{p+1}\|x_k - x_*\|^p + H\|x_k - x_*\|^p\right)\|x_k - x_*\| \\
&+ \frac{\gamma[(\tau+1)\theta]^{l_k}}{1-\gamma H\|x_k - x_*\|^p}\left(\frac{H}{p+1}\|x_k - x_*\|^{p+1} + 2\beta\|x_k - x_*\|\right) \\
\leq& \frac{\gamma}{1-\gamma H\|x_k - x_*\|^p}\left(\frac{p+2+[(\tau+1)\theta]^{l_k}}{p+1}H\|x_k - x_*\|^p\right. \\
&\left. + 2\beta[(\tau+1)\theta]^{l_k}\right)\|x_k - x_*\| \\
\leq& \frac{\gamma}{1-\gamma H\|x_k - x_*\|^p}[3H\|x_k - x_*\|^p + 2\beta[(\tau+1)\theta]^{l_k}]\|x_k - x_*\| \\
=& g(\|x_k - x_*\|, l_k)\|x_k - x_*\|.
\end{aligned}$$

For the consequent $g(\|x_k - x_*\|, l_k) < g(r, \mu_*) < 1$ in Theorem 3.1, apparently we have

$$\|x_{k+\frac{1}{2}} - x_*\| < \|x_k - x_*\|.$$

Similarly we get the estimate $\|x_{k+1}-x_*\|$,

$$\begin{aligned}\|x_{k+1}-x_*\| =&\|x_{k+\frac{1}{2}}-x_*-(I-H_{\alpha,\omega}(x_k)^{m_k})\mathcal{A}(x_k)^{-1}\mathcal{P}(x_{k+\frac{1}{2}})\|\\ \leq&\|x_{k+\frac{1}{2}}-x_*-\mathcal{A}(x_k)^{-1}\mathcal{P}(x_{k+\frac{1}{2}})\|\\ &+\|H_{\alpha,\omega}(x_k)\|^{m_k}\|\|\mathcal{A}(x_k)^{-1}\mathcal{P}(x_{k+\frac{1}{2}})\|\\ \leq&\frac{\gamma}{1-\gamma H\|x_k-x_*\|^p}\Bigg(\frac{H}{p+1}\|x_{k+\frac{1}{2}}-x_*\|^p\\ &+H\|x_k-x_*\|^p\Bigg)\|x_{k+\frac{1}{2}}-x_*\|\\ &+\frac{\gamma[(\tau+1)\theta]^{m_k}}{1-\gamma H\|x_k-x_*\|^p}\left(\frac{H}{p+1}\|x_{k+\frac{1}{2}}-x_*\|^{p+1}+2\beta\|x_{k+\frac{1}{2}}-x_*\|\right)\\ \leq&\frac{\gamma g(\|x_k-x_*\|,l_k)}{1-\gamma H\|x_k-x_*\|^p}\times\|x_k-x_*\|\\ &\times\Bigg(\frac{g(\|x_k-x_*\|,l_k)^p[1+((\tau+1)\theta)^{m_k}]+p+1}{p+1}H\|x_k-x_*\|^p\\ &+2\beta[(\tau+1)\theta]^{m_k}\Bigg)\\ \leq&\frac{\gamma g(\|x_k-x_*\|,l_k)}{1-\gamma H\|x_k-x_*\|^p}\left(3H\|x_k-x_*\|^p+2\beta[(\tau+1)\theta]^{m_k}\right)\|x_k-x_*\|\\ =&g(\|x_k-x_*\|,l_k)g(\|x_k-x_*\|,m_k)\|x_k-x_*\|\\ <&g(r,\mu_*)^2\|x_k-x_*\|\\ <&\|x_k-x_*\|.\end{aligned}$$

Therefore, for any $x_0\in\mathbb{N}(x_*,r)\subset\mathbb{N}_0$,

$$0\leq ...<\|x_{k+1}-x_*\|<\|x_k-x_*\|<...<\|x_0-x_*\|<r,$$

which shows $x_{k+1}\in\mathbb{N}(x_*,r)$. Additionally, the conclusion that $x_{k+1}\to x_*$ for $k\to\infty$ can be obtained. With the truth that $\|x_{k+1}-x_*\|<g(r,\mu_*)^2\|x_k-x_*\|$, go a step further, we can get

$$\|x_k-x_*\|<g(r_p,\mu_*)^{2k}\|x_0-x_*\|,$$

or equivalently,

$$\|x_k-x_*\|^{\frac{1}{k}}<g(r_p,\mu_*)^2\|x_0-x_*\|^{\frac{1}{k}},$$

when $k\to\infty$, Eq.(12) holds.

□

4. Numerical examples

In this section, we demonstrate the effectiveness and superiority of the MN-NDSS method by several nonlinear equations in [14,18]. These nonlinear equations are computed numerically by MATLAB Version R2019b with 3.40 GHz Intel Core i7 CPU and 64.00 GB RAM. In the section that follows, the MN-NDSS method is compared with a couple of iterative methods proposed in recent years which are the modified Newton-DGPMHSS (MN-DGPMHSS) method [16], the modified Newton-AGSOR (MN-AGSOR) method [22], the modified Newton-DSS (MN-DSS) method [18], and the modified Newton-FPAE (MN-FPAE) method [28]. We compare the computational efficiency of these methods from the perspective of the computational time and iteration steps. We evaluate the computational efficiency of these iterative methods from four criteria which are the internal iteration steps, the external iteration steps, the calculation time and the residual error estimate. In the experimental results, we denote these aspects by In Step, Out Step, CPU time, and RES respectively. Let the product of the maximum allowable number of inner and outer iterations be IT_{max}=InStep×OutStep=1000. we will record the the experimental results as - for the methods that fail to reach the stopping criteria in the maximum allowable number IT_{max}.

Example 4.1. Let us discuss the next nonlinear systems, see [14]:

$$\begin{cases} u_t - (\alpha_1 + i\beta_1)(u_{xx} + u_{yy}) + qu = -(\alpha_2 + i\beta_2)\, u^{\frac{4}{3}}, & \text{in } (0,1] \times \Omega, \\ u(0,x,y) = u_0(x,y), & \text{in } \Omega, \\ u(t,x,y) = 0, & \text{on } (0,1] \times \partial\Omega, \end{cases}$$

with Ω being a square $(0,1)\times(0,1)$ and $\partial\Omega$ denoting its boundary. In the meantime, q is a positive constant that controls the magnitude of the reaction term and the coefficients $\alpha_1 = \beta_1 = 1, \alpha_2 = \beta_2 = 1$. Using the central finite difference method to discretize this nonlinear system on the equidistant discretization grid with mesh size $\Delta t = h = \frac{1}{N+1}$, the following nonlinear equations need to be solved at each temporal step.

$$F(x) = Mx + (\alpha_2 + i\beta_2)h\Delta t\Psi(x) = 0, \tag{13}$$

where

$$\Psi(x) = (x_1^{\frac{4}{3}}, x_2^{\frac{4}{3}}, ..., x_n^{\frac{4}{3}})^T,$$

$$M = h(1 + q\Delta t)I_n + (\alpha_1 + i\beta_1)\frac{\Delta t}{h}(A_N \otimes I_N + I_N \otimes A_N),$$

with $n = N \times N$ and the tridiagonal matrix $A_N = tridiag(-1, 2, -1)$. In addition, $\otimes$ is the kronecker product symbol. For this numerical example, we take the original vector $x_0 = 1$ and the termination criteria of all the outer iteration as

$$\frac{\|F(x_k)\|}{\|F(x_0)\|} \leq 10^{-10}.$$

Next, we take the tolerance of inner iteration $\eta_k = \tilde{\eta_k} = \eta$ for all of these iteration methods. The Jacobian matrix of (13) has the following form:

$$F^{'}(x) = M + \frac{4}{3}h\Delta t(\alpha_2 + i\beta_2) \times diag(x_1^{\frac{1}{3}}, x_2^{\frac{1}{3}}, ..., x_n^{\frac{1}{3}}).$$

The experimentally optimal parameters that refelect the least iterative steps and CPU time are employed to these methods in this numerical experiment, see Table 1.

To ensure the reliability of the numerical experiment, we utilize the same problem parameters for these methods and study the calculation results in various cases. Set the magnitude of reaction term $q = 1, 10, 200$, the scale of problem $N = 30, 40, 50, 100, 150$ and the tolerance of inner iteration $\eta = 0.1, 0.2, 0.4$. For the sake of brevity, we present only some representative experimental results.

TABLE 1. The optimal parameters for Example 4.1.

	N	$q=1$			$q=10$			$q=200$		
		$\eta=0.1$	$\eta=0.2$	$\eta=0.4$	$\eta=0.1$	$\eta=0.2$	$\eta=0.4$	$\eta=0.1$	$\eta=0.2$	$\eta=0.4$
MN-DSS	30	1.93	0.54	0.54	0.48	0.67	0.48	3.78	4.54	4.67
	40	0.48	0.69	0.46	0.53	0.35	0.45	0.18	0.19	0.21
	50	0.53	0.35	0.45	0.45	0.35	0.41	0.18	0.19	4.78
	100	0.36	0.29	0.45	0.35	0.29	0.44	0.18	0.20	0.20
	150	0.24	0.35	0.33	0.25	0.32	0.21	0.19	0.16	0.16
MN-DGPMHSS	30	(1.79,0.21)	(1.90,0.23)	(1.92,0.04)	(2.90,0.26)	(0.99,0.20)	(1.63,0.20)	(1.21,0.71)	(1.40,0.70)	(1.40,0.70)
	40	(3.79,0.20)	(1.02,0.24)	(1.55,0.18)	(8.61,0.26)	(1.24,0.26)	(1.91,0.78)	(1.60,0.90)	(1.40,0.65)	(1.41,0.76)
	50	(3.82,0.12)	(1.02,0.24)	(1.92,0.18)	(1.18,0.26)	(1.27,0.26)	(1.90,0.77)	(1.56,0.83)	(1.60,0.75)	(0.94,0.54)
	100	(1.79,0.58)	(1.80,0.86)	(1.60,0.67)	(1.76,0.62)	(1.83,0.87)	(1.57,0.66)	(1.59,0.82)	(2.09,0.82)	(2.06,0.81)
	150	(1.79,0.55)	(1.80,0.86)	(1.55,0.67)	(1.86,0.62)	(1.83,0.87)	(1.91,0.78)	(1.60,1.83)	(1.40,0.70)	(1.43,0.85)
MN-AGSOR	30	(0.71,0.92)	(1.00,0.68)	(1.00,0.68)	(1.00,0.61)	(1.00,0.68)	(1.00,0.69)	(1.00,0.67)	(1.00,0.71)	(1.00,0.77)
	40	(0.86,0.74)	(1.00,0.65)	(1.00,0.69)	(1.00,0.63)	(1.00,0.68)	(0.92,0.71)	(1.00,0.67)	(1.00,0.71)	(1.00,0.76)
	50	(0.85,0.75)	(1.00,0.71)	(1.00,0.69)	(1.00,0.64)	(1.00,0.66)	(0.92,0.71)	(1.00,0.67)	(1.00,0.72)	(1.00,0.75)
	100	(1.00,0.74)	(1.00,0.70)	(0.91,0.70)	(1.00,0.74)	(1.00,0.70)	(0.88,0.71)	(1.00,0.73)	(1.00,0.71)	(1.00,0.72)
	150	(1.00,0.73)	(1.00,0.70)	(0.90,0.71)	(1.00,0.72)	(1.00,0.70)	(0.91,0.71)	(1.00,0.73)	(1.00,0.73)	(1.00,0.73)
MN-NDSS	30	(0.36,0.68)	(0.36,0.68)	(0.36,0.68)	(0.29,0.50)	(0.29,0.50)	(0.29,0.50)	(0.23,0.62)	(0.23,0.62)	(0.23,0.62)
	40	(0.36,0.68)	(0.16,0.43)	(0.16,0.43)	(0.24,0.50)	(0.24,0.50)	(0.24,0.50)	(0.23,0.62)	(0.23,0.62)	(0.23,0.62)
	50	(0.23,0.49)	(0.23,0.49)	(0.23,0.49)	(0.33,0.63)	(0.35,0.64)	(0.35,0.64)	(0.23,0.62)	(0.23,0.62)	(0.23,0.62)
	100	(0.32,0.72)	(0.32,0.72)	(0.32,0.72)	(0.23,0.68)	(0.24,0.50)	(0.24,0.76)	(0.23,0.62)	(0.23,0.62)	(0.23,0.62)
	150	(0.31,0.76)	(0.32,0.72)	(0.32,0.72)	(0.31,0.76)	(0.31,0.76)	(0.24,0.50)	(0.23,0.62)	(0.23,0.62)	(0.23,0.62)

TABLE 2. Experimental results for $\eta = 0.1, q = 1$ of Example 4.1.

N	Method	RES	Out step	In Step	CPU time(s)
30	MN-FPAE	-	-	-	-
	MN-DGPMHSS	7.7821×10^{-11}	4	22	0.0835
	MN-DSS	1.3072×10^{-11}	3	11	0.0318
	MN-AGSOR	8.8349×10^{-11}	3	13	0.0634
	MN-NDSS	9.3007×10^{-11}	2	4	0.0447
40	MN-FPAE	-	-	-	-
	MN-DGPMHSS	8.1136×10^{-11}	4	23	0.1647
	MN-DSS	1.9010×10^{-11}	3	12	0.0677
	MN-AGSOR	7.4807×10^{-11}	3	14	0.1198
	MN-NDSS	1.1943×10^{-13}	3	6	0.0971
50	MN-FPAE	-	-	-	-
	MN-DGPMHSS	4.4032×10^{-11}	4	27	0.3191
	MN-DSS	6.9896×10^{-11}	4	14	0.1756
	MN-AGSOR	9.3457×10^{-11}	3	14	0.2108
	MN-NDSS	7.9469×10^{-11}	2	4	0.1321
100	MN-FPAE	-	-	-	-
	MN-DGPMHSS	6.9413×10^{-11}	4	27	2.8856
	MN-DSS	2.8360×10^{-11}	4	19	2.2984
	MN-AGSOR	7.6891×10^{-11}	3	13	2.0070
	MN-NDSS	9.7470×10^{-11}	2	4	1.3043
150	MN-FPAE	-	-	-	-
	MN-DGPMHSS	6.9989×10^{-11}	4	28	11.8253
	MN-DSS	6.2227×10^{-11}	4	24	10.7744
	MN-AGSOR	5.7757×10^{-11}	3	12	8.7479
	MN-NDSS	8.8138×10^{-11}	2	4	6.2487

It is dramatically evident that the modified Newton-NDSS method has superior performance no matter the iteration steps or CPU time compared with the MN-DGPMHSS method, the MN-AGSOR method, the MN-DSS method, and the MN-FPAE method. In this experiment, under the given step limit IT_{max}, the MN-FPAE method is unable to obtain a numerical solution. From $q = 1$ to $q = 10$, the number of iteration steps of the other methods increases significantly, while the MN-NDSS method remains the same. This suggests that our approach is more applicable. From the perspective of the total iteration steps, the modified Newton-AGSOR method, and the modified Newton DSS method are at least three times more expensive than the modified Newton-NDSS method. Our method also saves more time than the four previously mentioned methods. So the MN-NDSS method has more outstanding performance compared with the three other existing methods.

Example 4.2. Consider the nonlinear Helmholtz equation, see [18]:

$$-\Delta u + \sigma_1 u + i\sigma_2 u = -e^u,$$

where u meets the Dirichlet boundary condition in the region $\Omega = [0,1] \times [0,1]$ and σ_1, σ_2 are two real numbers. Discretizing this differential equation on an $N \times N$ grid by the finite difference with step width $h = 1/(N+1)$, the following complex nonlinear system is reached.

$$F(u) = Mx + \Psi(x) = 0, \tag{14}$$

where

$$\Psi(x) = (e^{x_1}, e^{x_2}, ..., e^{x_n}),$$

$$M = (B_N \otimes I_N + I_N \otimes B_N + \sigma_1 I_n) + i\sigma_2 I_n,$$

TABLE 3. Experimental results for $\eta = 0.2, q = 10$ of Example 4.1.

N	Method	RES	Out step	In Step	CPU time(s)
30	MN-FPAE	-	-	-	-
	MN-DGPMHSS	7.9209×10^{-11}	5	27	0.0914
	MN-DSS	5.7751×10^{-11}	4	14	0.0356
	MN-AGSOR	7.7139×10^{-12}	4	14	0.0679
	MN-NDSS	4.9665×10^{-11}	2	4	0.0353
40	MN-FPAE	-	-	-	-
	MN-DGPMHSS	8.7803×10^{-11}	5	26	0.1965
	MN-DSS	6.8593×10^{-11}	4	16	0.0892
	MN-AGSOR	1.0412×10^{-11}	4	14	0.1447
	MN-NDSS	6.7612×10^{-11}	2	4	0.0722
50	MN-FPAE	-	-	-	-
	MN-DGPMHSS	7.3313×10^{-11}	5	27	0.3485
	MN-DSS	6.8853×10^{-11}	4	16	0.1810
	MN-AGSOR	2.9389×10^{-11}	4	14	0.2644
	MN-NDSS	9.0436×10^{-11}	2	4	0.1048
100	MN-FPAE	-	-	-	-
	MN-DGPMHSS	8.0502×10^{-11}	5	29	3.3844
	MN-DSS	4.2315×10^{-11}	5	20	2.8649
	MN-AGSOR	5.8755×10^{-11}	4	14	2.6585
	MN-NDSS	6.9412×10^{-13}	3	6	2.0121
150	MN-FPAE	-	-	-	-
	MN-DGPMHSS	3.9834×10^{-11}	5	30	14.9538
	MN-DSS	7.0468×10^{-11}	5	25	13.2545
	MN-AGSOR	4.1385×10^{-11}	4	14	11.3124
	MN-NDSS	9.2248×10^{-13}	2	4	5.9850

TABLE 4. Experimental results for $\eta = 0.4, q = 200$ of Example 4.1.

N	Method	RES	Out step	In Step	CPU time(s)
30	MN-FPAE	-	-	-	-
	MN-DGPMHSS	5.9813×10^{-11}	8	32	0.1507
	MN-DSS	4.5653×10^{-11}	7	28	0.0595
	MN-AGSOR	6.3732×10^{-11}	6	15	0.0948
	MN-NDSS	8.8774×10^{-12}	3	6	0.0550
40	MN-FPAE	-	-	-	-
	MN-DGPMHSS	4.6711×10^{-11}	8	32	0.2974
	MN-DSS	6.2285×10^{-11}	7	28	0.1431
	MN-AGSOR	5.3391×10^{-11}	6	15	0.2032
	MN-NDSS	6.3307×10^{-12}	3	6	0.1073
50	MN-FPAE	-	-	-	-
	MN-DGPMHSS	9.8044×10^{-11}	8	32	0.5519
	MN-DSS	9.8701×10^{-11}	7	28	0.3101
	MN-AGSOR	7.2279×10^{-11}	6	14	0.3762
	MN-NDSS	4.6803×10^{-12}	3	6	0.2028
100	MN-FPAE	-	-	-	-
	MN-DGPMHSS	8.5971×10^{-11}	8	31	5.2984
	MN-DSS	4.2220×10^{-11}	8	32	4.5296
	MN-AGSOR	6.6597×10^{-11}	6	12	3.8316
	MN-NDSS	1.5172×10^{-12}	3	6	2.0121
150	MN-FPAE	-	-	-	-
	MN-DGPMHSS	4.1993×10^{-11}	8	32	22.3303
	MN-DSS	7.9965×10^{-11}	9	36	31.0352
	MN-AGSOR	4.6841×10^{-11}	6	12	16.4720
	MN-NDSS	7.2664×10^{-13}	3	6	8.6545

TABLE 5. The optimal parameters for Example 4.2.

N	η	MN-DGPMHSS	MN-AGSOR	MN-FPAE	MN-NDSS
	0.1	(1.03,0.34)	(0.98,0.89)	1.01	(0.22,0.86)
30	0.2	(0.83,0.53)	(0.95,0.94)	0.97	(0.22,0.86)
	0.4	(1.52,1.54)	(0.96,0.92)	0.85	(0.22,0.86)
	0.1	(1.03,0.34)	(1.02,0.92)	0.88	(0.22,0.86)
60	0.2	(0.83,0.53)	(0.97,0.90)	0.97	(0.22,0.86)
	0.4	(0.71,0.56)	(0.96,0.91)	0.86	(0.22,0.86)
	0.1	(1.03,0.34)	(0.96,0.90)	0.86	(0.22,0.86)
90	0.2	(0.83,0.53)	(0.97,0.91)	0.97	(0.22,0.86)
	0.4	(0.71,0.56)	(0.96,0.90)	0.84	(0.22,0.86)
	0.1	(1.03,0.34)	(0.96,0.90)	0.82	(0.22,0.86)
120	0.2	(0.70,0.60)	(0.97,0.90)	0.97	(0.22,0.86)
	0.4	(0.71.0.60)	(0.96,0.90)	0.83	(0.22,0.86)

with $n = N \times N$ and $B_N = tridiag(-1, 2, -1)/h^2$ representing a real tridiagonal matrix. In this actual experiment, we take the initial guess $x_0 = 1$ and the problem parameters $\sigma_1 = 1, \sigma_2 = 10$. What's more, the stopping criteria for the outer iteration are taken as

$$\frac{\|F(x_k)\|_2}{\|F(x_0)\|_2} \leq 10^{-10}.$$

We adopt the prescribed tolerance $\eta_k = \tilde{\eta}_k = \eta = 0.1, 0.2, 0.4$. Besides, the dimension of problem $N = 30, 60, 90, 120$ are considered in the practical implements. The optimal parameters in experiments that reflect the least iterative steps and calculation time are employed in these methods, see Table 5. From Table 5, we can see that the optimal parameters of the MN-NDSS method always keep steady with the change of the problem parameters, which shows that our new method is feasible in practice.

The inner and outer iteration steps and CPU time of the four methods are shown in Table 6 under those problem parameter choices mentioned above. From Table 6, it is evident that the modified Newton-NDSS method performs excellently whether in terms of computation time or iteration numbers. For the equations in Example 4.2, the MN-DSS method has been unable to obtain a numerical solution that meets the accuracy under the given step limit IT_{max}. In addition, the latest MN-FPAE method also performed poorly. Even the MN-AGSOR method with suboptimal performance still has several times as many iteration steps as the MN-NDSS method. Notably, the iteration steps of the MN-NDSS method keep steady with different choices of problem parameters which suggests the constancy of our method. Moreover, as the scale of the problem increases, our method is still able to give the solution of the equation after a few iteration steps. This shows that the MN-NDSS method is suitable for solving large systems.

5. Conclusions

In this work, we proposed the modified Newton-NDSS iteration method to find the solution to complex nonlinear systems. Besides, under the H?lder assumption rather than the stronger Lipschitz hypothesis, the local convergence analysis and proof of the MN-NDSS method are given in Section 3. Several nonlinear partial differential systems illustrate the implementability and efficiency of the new method.

TABLE 6. Experimental results of Example 4.2.

η	N	Method	RES	Out step	In step	CPU time(s)
0.1	$N = 30$	MN-DSS	-	-	-	-
		MN-DGPMHSS	5.8090×10^{-11}	5	40	0.1147
		MN-FPAE	5.3119×10^{-11}	4	27	0.0327
		MN-AGSOR	9.2702×10^{-11}	3	8	0.0547
		MN-NDSS	3.8327×10^{-11}	2	4	0.0374
	$N = 60$	MN-DSS	-	-	-	-
		MN-DGPMHSS	6.6345×10^{-11}	5	40	0.6971
		MN-FPAE	6.9461×10^{-11}	4	22	0.3286
		MN-AGSOR	7.9857×10^{-11}	3	7	0.3452
		MN-NDSS	1.4177×10^{-11}	2	4	0.2454
	$N = 90$	MN-DSS	-	-	-	-
		MN-DGPMHSS	6.8390×10^{-11}	5	40	2.5387
		MN-FPAE	8.7160×10^{-11}	4	21	1.4915
		MN-AGSOR	3.1546×10^{-11}	4	8	1.7458
		MN-NDSS	7.8216×10^{-12}	2	4	0.8993
	$N = 120$	MN-DSS	-	-	-	-
		MN-DGPMHSS	6.9192×10^{-11}	5	40	6.7848
		MN-FPAE	6.3238×10^{-11}	4	21	4.5667
		MN-AGSOR	2.1265×10^{-11}	4	8	4.9754
		MN-NDSS	5.1128×10^{-12}	2	4	2.5367
0.2	$N = 30$	MN-DSS	-	-	-	-
		MN-DGPMHSS	7.9903×10^{-11}	6	36	0.1129
		MN-FPAE	1.1985×10^{-11}	5	27	0.0359
		MN-AGSOR	7.8209×10^{-11}	4	8	0.0660
		MN-NDSS	3.8327×10^{-11}	2	4	0.0332
	$N = 60$	MN-DSS	-	-	-	-
		MN-DGPMHSS	8.4244×10^{-11}	6	36	0.8053
		MN-FPAE	3.9369×10^{-11}	5	25	0.4018
		MN-AGSOR	2.9781×10^{-11}	4	8	0.4694
		MN-NDSS	1.4177×10^{-11}	2	4	0.2412
	$N = 90$	MN-DSS	-	-	-	-
		MN-DGPMHSS	8.5221×10^{-11}	6	36	2.8719
		MN-FPAE	2.1624×10^{-11}	5	25	1.8331
		MN-AGSOR	1.1794×10^{-11}	4	8	1.7669
		MN-NDSS	7.8216×10^{-12}	2	4	0.9305
	$N = 120$	MN-DSS	-	-	-	-
		MN-DGPMHSS	7.5157×10^{-11}	6	36	7.9001
		MN-FPAE	1.4113×10^{-11}	5	25	5.5826
		MN-AGSOR	1.0887×10^{-11}	4	8	4.9288
		MN-NDSS	5.1128×10^{-12}	2	4	2.5227
0.4	$N = 30$	MN-DSS	-	-	-	-
		MN-DGPMHSS	8.3980×10^{-11}	9	36	0.1540
		MN-FPAE	5.0867×10^{-11}	7	24	0.0518
		MN-AGSOR	6.3433×10^{-11}	4	8	0.0586
		MN-NDSS	3.8327×10^{-11}	2	4	0.0350
	$N = 60$	MN-DSS	-	-	-	-
		MN-DGPMHSS	9.5034×10^{-11}	9	36	1.1372
		MN-FPAE	4.5756×10^{-11}	7	23	0.5422
		MN-AGSOR	2.5748×10^{-11}	4	8	0.4567
		MN-NDSS	1.4177×10^{-11}	2	4	0.2367
	$N = 90$	MN-DSS	-	-	-	-
		MN-DGPMHSS	9.5631×10^{-11}	9	36	4.1318
		MN-FPAE	5.5829×10^{-11}	7	22	2.4685
		MN-AGSOR	3.1546×10^{-11}	4	8	1.7432
		MN-NDSS	7.8216×10^{-12}	2	4	0.9432
	$N = 120$	MN-DSS	-	-	-	-
		MN-DGPMHSS	7.2038×10^{-11}	9	36	11.2966
		MN-FPAE	8.6120×10^{-11}	7	21	7.6355
		MN-AGSOR	2.1265×10^{-11}	4	8	4.9856
		MN-NDSS	5.1128×10^{-12}	2	4	2.5806

Meanwhile, numerical experiments explain that the MN-NDSS method outperforms the MN-DGPMHSS method, the MN-DSS method, the MN-FPAE method, and the MN-AGSOR method in terms of computation time and the number of iteration steps. Actually, it is indeed a puzzle to choose the experimentally optimal parameters for these iteration methods which takes a lot of time in Section 4. In future work, we plan to face the challenge of discussing the optimal parameters.

Acknowledgments

This research was supported by National Natural Science Foundation of China (Grant No. 12271479).

References

[1] H.O. Karlsson. The quasi-minimal residual algorithm applied to complex symmetric linear systems in quantum reactive scattering. The Journal of Chemical Physics, 103(12):4914–4919, 1995.

[2] D.V. Papp and B.I. Vizvri. Effective solution of linear Diophantine equation systems with an application in chemistry. Journal of Mathematical Chemistry, 39(1):15–31, 2006.

[3] Y. Zhang and Q. Sun. Preconditioned bi-conjugate gradient method of large-scale sparse complex linear equation group. Chinese Journal Electronics, 20(1):192–194, 2011.

[4] Z.Z Bai, G.H. Gloub and M.K. Ng. Hermitian and skew-Hermitian splitting methods for non-Hermitian positive definite linear systems. SIAM Journal on Matrix Analysis and Applications, 24(3):603–626, 2003.

[5] Z.Z. Bai, G.H. Gloub and J.Y. Pan. Preconditioned Hermitian and skew-Hermitian splitting methods for non-Hermitian positive semidefinite linear systems. Numerische Mathematik, 98(1):1–32, 2004.

[6] Z.Z. Bai, M. Benzi and F. Chen. Modified HSS iteration methods for a class of complex symmetric linear systems. Computing, 87:93–111, 2010.

[7] Z.Z. Bai, M. Benzi and F. Chen. On preconditioned MHSS iteration methods for complex symmetric linear systems. Numercial Algorithms, 56(2):297–317, 2011.

[8] M. Dehghan, M. Dehghani-Madiseh and M. Hajarian. A generalized preconditioned MHSS method for a class of complex symmetric linear systems. Mathematical Modelling and Analysis, 18(4):561–576, 2013.

[9] C.X. Li and S.L. Wu. A single-step HSS method for non-Hermitian positive definite linear systems. Applied Mathematics Letters, 44:26–29, 2015.

[10] A. Shirilord and M. Dehghan. Double parameter splitting (DPS) iteration method for solving complex symmetric linear systems. Applied Numerical Mathematics, 171:176–192, 2022.

[11] Z.G. Huang. Modified two-step scale-splitting iteration method for solving complex symmetric linear systems. Computational and applied mathematics, 40(1):122–156, 2022.

[12] Z.Z. Bai and X.P. Guo. On Newton-HSS methods for systems of nonlinear equations with positive-definite Jacobian matrices. Journal of Computational Mathematics, 2:235–260, 2010.

[13] A.L. Yang and Y.J. Wu. Newton-MHSS methods for solving systems of nonlinear equations with complex symmetric Jacobian matrices. Numerical Algebra, Control and Optimization, 2(4):839–853, 2012.

[14] H.X. Zhong, G.L. Chen and X.P. Guo. On preconditioned modified Newton-MHSS method for systems of nonlinear equations with complex symmetric Jacobian matrices. Numerical Algorithms, 69:553–567, 2015.

[15] F. Xie, Q.B. Wu and P.F. Dai. Modified NewtonCSHSS method for a class of systems of nonlinear equations. Computational and Applied Mathematics, 38(1):1–25, 2019.

[16] M.H. Chen and Q.B. Wu. On modified NewtonCDGPMHSS method for solving nonlinear systems with complex symmetric Jacobian matrices. Computers and Mathematics with Applications, 76(1):45–57, 2018.

[17] P.F. Dai, Q.B. Wu and M.H. Chen. Modified Newton-NSS method for solving systems of nonlinear equations. Numerical Algorithms, 77(1):1–21, 2017.

[18] F. Xie, R.F. Lin and Q.B. Wu. Modified Newton-DSS method for solving a class of systems of nonlinear equations with complex symmetric Jacobian matrices. Numerical Algorithms, 85: 951–975, 2019.

[19] Z.Z. Bai, M. Benzi, F. Chen and Z.Q. Wang. Preconditioned MHSS iteration methods for a class of block two-by-two linear systems with applications to distributed control problems. IMA Journal of Numerical Analysis, 33:343–369, 2013.

[20] T. Wang and L.Z. Lu, Alternating-directional PMHSS iteration method for a class of two-by-two block linear systems. Applied Mathematics Letters, 58:159–164, 2016.

[21] D.K. Salkuyeh, D. Hezari and V. Edalatpour. Generalized SOR iterative method for a class of complex symmetric linear system of equations. International Journal of Computer Mathematics, 92(4):802–815, 2015.

[22] V. Edalatpour, D. Hezari and D.K. Salkuyeh. Accelerated generalized SOR method for a class of complex systems of linear equations. Mathematical Communications, 20:37–52, 2015.

[23] D. Hezari, V. Edalatpour and D.K. Salkuyeh. Preconditioned GSOR iterative method for a class of complex symmetric system of linear equations. Numerical Linear Algebra with Applications, 22:761–776, 2015.

[24] X. Qi, H.T. Wu and X.Y. Xiao. Modified Newton-GSOR method for solving complex nonlinear systems with symmetric Jacobianmatrices. Computational and Applied Mathematics, 39(3):165–182, 2020.

[25] X. Qi, H.T. Wu and X.Y. Xiao. Modified Newton-AGSOR method for solving nonlinear systems with block two-by-two complex symmetric Jacobian matrices. Calcolo, 57(2):14, 2020.

[26] Y.Y. Feng and Q.B. Wu. MN-PGSOR method for solving nonlinear systems with block two-by-two complex symmetric Jacobian matrices. Journal of Mathematics, 2021:1–18, 2021.

[27] X.Y. Xiao and X. Wang. A new single-step iteration method for solving complex symmetric linear systems. Numerical Algorithms, 78:643–660,2018.

[28] L. Zhang, Q.B. Wu, M.H. Chen and R.F. Lin. Two new effective iteration methods for nonlinear systems with complex symmetric Jacobianmatrices. Computational and Applied Mathematics, 40(3): 97–125, 2021.

[29] Z.G. Huang. A new double-step splitting iteration method for certain block two-by-two linear systems. Computational and Applied Mathematics, 39:193–234, 2020.

School of Mathematical Sciences, Zhejiang University, HangZhou, 310027, Zhejiang, P.R.China
E-mail: `xiaohyu@zju.edu.cn and qbwu@zju.edu.cn`