Numerical Algorithms (2024) 96:333–368
https://doi.org/10.1007/s11075-023-01649-z

ORIGINAL PAPER

Modified Newton-PBS method for solving a class of complex symmetric nonlinear systems

Yuanyuan Zhang[1,2] · **Qingbiao Wu**[2] · **Yao Xiao**[2] · **Zhewei Xie**[2]

Received: 29 June 2022 / Accepted: 11 August 2023 / Published online: 5 October 2023

Abstract

The parameterized block splitting (PBS) is a convergent and efficient iterative method to solve the large complex symmetric linear systems. In this paper, by using PBS iterative technique, the Newton equation is approximately solved, then we establish the modified Newton-PBS iterative method to solve the complex nonlinear systems whose Jacobian matrices are large, sparse, and complex symmetric. Subsequently, the local convergence analysis are explored under appropriate conditions. Ultimately, we apply the new method and several known methods to experimental numerical examples, and experimental results verify the superiority and efficiency of our new method. Especially, in terms of CPU time and iteration steps, our method is obviously better.

Keywords Parameterized block splitting · Modified Newton-PBS method · Complex nonlinear systems · Symmetric Jacobian matrix · Convergence analysis

Mathematics Subject Classification 65F10 · 65F50 · 65H10

✉ Qingbiao Wu
qbwu@zju.edu.cn

Yuanyuan Zhang
yuanyzhang@zju.edu.cn

Yao Xiao
mail_yaox@zju.edu.cn

Zhewei Xie
zhewxie@163.com

1 School of Mathematics, China University of Mining and Technology, 221116 Jiangsu, Xuzhou, People's Republic of China

2 School of Mathematical Science, Zhejiang University, 310027 Zhejiang, Hangzhou, People's Republic of China

1 Introduction

Supposing that function $F : \mathbb{D} \subset \mathbb{C}^n \to \mathbb{C}^n$ is nonlinear and continuously differentiable, we consider solving the following nonlinear equations of large sparse complex systems:

$$F(v) = 0, \tag{1.1}$$

where $F = (F_1, \cdots, F_n)^T$ with $F_i = F_i(v)$, $i = 1, 2, \cdots, n$, $v = (v_1, \cdots, v_n)^T$. The Jacobian matrix $F'(v)$ of function $F(v)$ can be expressed in the following form:

$$F'(v) = W(v) + iT(v), \tag{1.2}$$

which is large, sparse and complex symmetric. $W(v), T(v) \in \mathbb{R}^{n\times n}$ are real, symmetric, and positive semi-definite matrices, but at least one of them is positive definite. Moreover, $i = \sqrt{-1}$ stands for imaginary unit. Theoretically, nonlinear systems (1.1) can exist in many engineering applications such as quantum mechanics, nonlinear waves, chemical reactions, mechanical engineering, and turbulence; we can refer to the literature [1–5].

From the previous literature [6, 7], the most commonly used method for solving nonlinear systems is the Newton method. We can get the solution of Eq. 1.1 by solving the following Newton equation:

$$F'(v_k)d_k = -F(v_k), \quad k \geq 0, \quad d_k = v_{k+1} - v_k, \tag{1.3}$$

$F'(v_k)$ is a Jacobian matrix in the formula. When the scale of problem (1.1) gradually increases, due to the fact that linear equations (1.3) must be exactly solved at each iterative step, the Newton method is regarded as expensive and difficult. In order to overcome the difficulty and accelerate the speed, an inexact Newton method [8,9] is developed, because it is unnecessary to seek the inverse of the Jacobian matrix. Then, we can get the following inexact calculation formula:

$$\|F'(v_k)d_k + F(v_k)\| \leq \eta_k \|F(v_k)\|, \quad k \geq 0, \tag{1.4}$$

here, $\eta_k \in [0, 1)$ is commonly called *forcing term* which is used to control the level of accuracy. Subsequently, some internal and external iterative methods produced by the combination of inexact Newton method and linear iterative methods are proposed [10–13].

In order to solve nonlinear systems (1.1) more efficiently, Darvish and Barati proposed the following modified Newton method [14]:

$$\begin{cases} u_k = v_k - F'(v_k)^{-1} F(v_k), \\ v_{k+1} = u_k - F'(v_k)^{-1} F(u_k), \quad k = 0, 1, 2, \cdots. \end{cases} \tag{1.5}$$

Compared with the inexact Newton method, it only requires one more step of calculation, which can further improve convergence speed and convergence order. For this reason, the modified Newton method has been the mainstream to solve the nonlinear systems so far [15–17]. Especially, for solving the generalized absolute value equations, Newton-based matrix splitting method has been gradually studied [18].

For different types of nonlinear systems, in order to improve the efficiency and speed of solving them, we take the modified Newton method as the outer iteration and find the appropriate linear method as the inner iteration. In fact, if we take the modified Newton method as an external iteration for nonlinear systems (1.1), it is equivalent to our focus on finding an efficient method to solve the following linear equation:

$$Av = b, \quad A = W + iT, \tag{1.6}$$

where $A \in \mathbb{C}^{n\times n}$ is a complex matrix, $W, T \in \mathbb{R}^{n\times n}$ are real, symmetric, and positive semi-definite matrices, but at least one of them is positive definite, and $v, b \in \mathbb{C}^n$ are complex vectors.

As early as 2003, the Hermitian/skew-Hermitian splitting (HSS) method [19] was proposed by Bai et al. for solving the complex symmetric non-Hermitian positive-definite Jacobian linear systems. Subsequently, HSS-type method was deeply studied and developed by some scholars [20–27]. Massive efficient iteration methods such as the modified HSS (MHSS) [28], the preconditioned modified HSS (PMHSS) [29, 30], the generalization of preconditioned MHSS (GPMHSS) [31], the double-parameter GPMHSS (DGPMHSS) (A32), the lopsided PMHSS (LPMHSS) [33], new HSS (NHSS) [34], a parameterized variant of the NHSS [35], and the parameterized and preconditioned variant of NHSS (PPNHSS) [36] were gradually obtained. Besides, Hezari et al. cleverly created a scale-splitting (SCSP) iteration method [37]. Soon after, Zheng et al. introduced a double-step scale splitting (DSS) [38] method. On the previous basis, the two-parameter two-step SCSP (TTSCSP) [39] method was established by Salkuyeh and Siahkolaei. Moreover, Wang et al. proposed the combination method of real part and imaginary part (CRI) [40], which is similar to DSS.

To solve the large sparse complex nonlinear systems (1.1) with symmetric Jacobian matrices more quickly and efficiently, some internal and external iterative methods based on the modified Newton method have been developed. In 2013, Wu et al. constructed the modified Newton-HSS method [41]. Afterwards, the modified Newton-DGPMHSS (MN-DGPMHSS) [42], the modified Newton-SHSS [43], and the modified Newton double-parameter modified generalized HSS (MN-DMGHSS) [44] methods were gradually discussed. Subsequently, Wu et al. developed the modified Newton double-step scale splitting (MN-DSS) [45] and the modified Newton fixed-point iteration adding the asymptotical error (MN-FPAE) [46] again. Nearly 2 years, the successive overrelaxation (SOR) method has been favored by scholars, then the generalized SOR (GSOR), the accelerated GSOR (AGSOR), the preconditioned GSOR (PGSOR), and the modified SOR-like [47] methods were further produced. Under the circumstance, Xiao et al. developed the modified Newton-GSOR (MN-GSOR) method and modified Newton-AGSOR (MN-AGSOR) method [48, 49], and the modified Newton-PGSOR method was put forward by Wu et al. [50].

The complex linear system (1.6) is essentially a special case of generalized saddle point problem [51]. For saddle point problems with complex symmetric systems, a shift-splitting (SS) iterative method was presented by Zheng and Ma [52]. In [53], the parameterized rotated SS (PRSS) iteration method was built. Soon after, a symmetric block triangular splitting (SBTS) iteration method was introduced, then its preconditioned version were devised by Zhang et al. [54]. According to the existing literature, it is obvious that we have to handle complex arithmetic in the CRI iteration method, which may reduce the efficiency of the algorithm and increase the time cost. Motivated by the disadvantage, a new block splitting (NBS) method [55] was established by Huang. It has the same upper bound of the spectral radius of iteration matrices of CRI; nevertheless, the new method avoids complex arithmetic. Meanwhile, Huang cleverly put forward the parameterized BS (PBS) [55] by utilizing the parameter accelerating technique to NBS.

Based on the motivation of improving computational efficiency and broadening the scope of the problems which can be solved, we consider whether we can use the modified Newton method to solve the large sparse complex nonlinear systems (1.1) to obtain modified Newton equations (1.5), then we use the efficient PBS method to further solve equations (1.5), so as to reduce the number of iterative steps of the method, improve the efficiency of the algorithm, and shorten the algorithm time. So we construct the modified Newton-PBS algorithm and directly use the numerical examples in practice to verify the performance of the algorithm. The numerical experimental results show that the algorithm is efficient and feasible, and we have made significant progress and improvement in both CPU time and iteration steps, which is great practical significance for solving engineering problems. Moreover, we find that the parameters of the algorithm are relatively stable. More specifically, even when the problem scale is large in practice, we can still quickly find the optimal parameters and obtain the numerical solution of the problem.

Throughout the full paper, the symbol $\|\Delta\|$ represents the 2-norm of vectors or matrices and spectral radius of matrix is expressed by $\rho(\Delta)$.

The framework of the rest of our paper is organized as follows. In Sect. 2, we mainly declare the essence of PBS method, when it is used to solve complex linear symmetric systems. In Sect. 3, we describe the modified Newton-PBS method for solving complex nonlinear symmetric systems. In Sect. 4, under appropriate conditions, the local convergence of the modified Newton-PBS method is verified. In Sect. 5, several concise numerical examples demonstrate the feasibility and efficiency of our method. In Sect. 6, some brief conclusions and prospects are presented.

2 The parameterized block splitting (PBS) method

Firstly, set $A = W + iT$, $v = y + iz$ and $b = m + in$, then the complex symmetric linear system (1.6) can be expanded into the following form:

$$Av = (W + iT)(y + iz) = (m + in), \tag{2.1}$$

where $W, T \in \mathbb{R}^{n\times n}$ are real, symmetric, and positive semi-definite matrices, but at least one of them is positive definite, and $y, z, m, n \in \mathbb{R}^n$. Therefore, it can be expressed in the equivalent real two-by-two form:

$$\mathcal{A}v = \begin{pmatrix} W & -T \\ T & W \end{pmatrix} \begin{pmatrix} y \\ z \end{pmatrix} = \begin{pmatrix} m \\ n \end{pmatrix}. \tag{2.2}$$

Next, we focus on solving linear equation (2.2) with PBS method.

2.1 The PBS method for solving linear systems

Inspired by the ideological essentials of transformed matrix iteration (TMIT) method [56], Huang considers multiplying the following matrix block left to

$$\mathcal{Z} = \begin{pmatrix} I & \zeta I \\ 0 & I \end{pmatrix}$$

to linear systems (2.2), ζ is a positive constant. We obtain

$$\mathcal{Z}\mathcal{A}v = \begin{pmatrix} \zeta T + W & \zeta W - T \\ T & W \end{pmatrix} \begin{pmatrix} y \\ z \end{pmatrix} = \begin{pmatrix} m + \zeta n \\ n \end{pmatrix} = \mathcal{Z}b := \widetilde{b}. \tag{2.3}$$

Let $\mathcal{B} = \begin{pmatrix} I & I \\ 0 & \alpha I \end{pmatrix}$, we rewrite the vector v as follows:

$$\begin{pmatrix} y \\ z \end{pmatrix} = \begin{pmatrix} I & I \\ 0 & \alpha I \end{pmatrix} \begin{pmatrix} \widetilde{y} \\ \widetilde{z} \end{pmatrix} = \begin{pmatrix} I & I \\ 0 & \alpha I \end{pmatrix} \widetilde{v} = \mathcal{B}\widetilde{v}, \tag{2.4}$$

where α is a positive constant. By substituting Eq. (2.4) into (2.3), we can get

$$\begin{aligned} \begin{pmatrix} \zeta T + W & \zeta W - T \\ T & W \end{pmatrix} \begin{pmatrix} I & I \\ 0 & \alpha I \end{pmatrix} \begin{pmatrix} \widetilde{y} \\ \widetilde{z} \end{pmatrix} &= \begin{pmatrix} \zeta T + W & (\zeta\alpha + 1)W + (\zeta - \alpha)T \\ T & \alpha W + T \end{pmatrix} \begin{pmatrix} \widetilde{y} \\ \widetilde{z} \end{pmatrix} \\ &= \begin{pmatrix} m + \zeta n \\ n \end{pmatrix}. \end{aligned} \tag{2.5}$$

Denote

$$\widetilde{\mathcal{A}} = \begin{pmatrix} \zeta T + W & (\zeta\alpha + 1)W + (\zeta - \alpha)T \\ T & \alpha W + T \end{pmatrix}, \tag{2.6}$$

then (2.2) can be expressed as

$$\widetilde{\mathcal{A}}\widetilde{v} := \begin{pmatrix} \zeta T + W & (\zeta\alpha + 1)W + (\zeta - \alpha)T \\ T & \alpha W + T \end{pmatrix} \begin{pmatrix} \widetilde{y} \\ \widetilde{z} \end{pmatrix} = \widetilde{b}. \tag{2.7}$$

Define the following splitting form of the coefficient matrix $\widetilde{\mathcal{A}}$ that

$$\widetilde{\mathcal{A}} = \begin{pmatrix} \zeta T + W & 0 \\ T & \alpha W + T \end{pmatrix} - \begin{pmatrix} 0 & -[(\zeta\alpha + 1)W + (\zeta - \alpha)T] \\ 0 & 0 \end{pmatrix} = \mathcal{M}_{\zeta,\alpha} - \mathcal{N}_{\zeta,\alpha},$$

the expression of matrices $\mathcal{M}_{\zeta,\alpha}$, $\mathcal{N}_{\zeta,\alpha}$ are defined by the above equation. Based on the theoretical analysis, the parameterized block splitting (PBS) iteration method for solving (2.2) has been obtained:

$$\begin{pmatrix} \zeta T + W & 0 \\ T & \alpha W + T \end{pmatrix} \begin{pmatrix} \widetilde{y}_{k+1} \\ \widetilde{z}_{k+1} \end{pmatrix} = \begin{pmatrix} 0 & -[(\zeta\alpha+1)W + (\zeta-\alpha)T] \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \widetilde{y}_k \\ \widetilde{z}_k \end{pmatrix} + \begin{pmatrix} m+\zeta n \\ n \end{pmatrix}. \tag{2.8}$$

Thereupon, the PBS iteration method is described as follows.

Algorithm 1 The PBS method for solving linear systems (2.2).

1. Given an initialized $\widetilde{v}_0 = (\widetilde{y}_0^T, \widetilde{z}_0^T)^T \in \mathbb{C}^{2n}$, where $\widetilde{y}_0, \widetilde{z}_0 \in \mathbb{R}^n$.
2. Compute $\widetilde{v}_{k+1} = (\widetilde{y}_{k+1}^T, \widetilde{z}_{k+1}^T)^T$ for $k = 0, 1, 2, \cdots$ via the following procedure until $v_k = ({y_k}^T, {z_k}^T)^T$ meets the stopping criterion:

$$\begin{cases} (\zeta T + W)\widetilde{y}_{k+1} = -[(\zeta\alpha+1)W + (\zeta-\alpha)T]\widetilde{z}_k + m + \zeta n, \\ (\alpha W + T)\widetilde{z}_{k+1} = -T\widetilde{y}_{k+1} + n, \\ y_{k+1} = \widetilde{y}_{k+1} + \widetilde{z}_{k+1}, z_{k+1} = \alpha\widetilde{z}_{k+1}. \end{cases} \tag{2.9}$$

Wherein α and ζ are positive constants.

Apparently, in the first two sub-systems of equations (2.9), the coefficient matrices are positive definite, so we can use Cholesky factorization and GMRES methods to solve these sub-systems directly. In a simplified equation (2.8), the PBS method can be rewritten as the following expression:

$$\begin{aligned} \begin{pmatrix} \widetilde{y}_{k+1} \\ \widetilde{z}_{k+1} \end{pmatrix} &= Q_{\zeta,\alpha} \begin{pmatrix} \widetilde{y}_k \\ \widetilde{z}_k \end{pmatrix} + G_{\zeta,\alpha} \begin{pmatrix} m+\zeta n \\ n \end{pmatrix} = Q_{\zeta,\alpha}^{k+1} \begin{pmatrix} \widetilde{y}_0 \\ \widetilde{z}_0 \end{pmatrix} \\ &\quad + \sum_{i=0}^{k} Q_{\zeta,\alpha}^i G_{\zeta,\alpha} \begin{pmatrix} m+\zeta n \\ n \end{pmatrix}, \ k = 0, 1, 2, \cdots, \end{aligned} \tag{2.10}$$

where

$$Q_{\zeta,\alpha} = \begin{pmatrix} \zeta T + W & 0 \\ T & \alpha W + T \end{pmatrix}^{-1} \begin{pmatrix} 0 & -[(\zeta\alpha+1)W + (\zeta-\alpha)T] \\ 0 & 0 \end{pmatrix}, \tag{2.11}$$

and

$$G_{\zeta,\alpha} = \begin{pmatrix} \zeta T + W & 0 \\ T & \alpha W + T \end{pmatrix}^{-1}. \tag{2.12}$$

The matrix $Q_{\zeta,\alpha}$ is called the iteration matrix of PBS method. Obviously, $Q_{\zeta,\alpha}$ and $G_{\zeta,\alpha}$ satisfy

$$Q_{\zeta,\alpha} = \mathcal{M}_{\zeta,\alpha}^{-1}\mathcal{N}_{\zeta,\alpha}, \quad G_{\zeta,\alpha} = \mathcal{M}_{\zeta,\alpha}^{-1}. \tag{2.13}$$

Remark 2.1 It is known that the spectral radius determines the convergence speed of the corresponding iteration method. Refer to the [55, Theorem 3.1]; the spectral radius of the iterative matrix $Q_{\zeta,\alpha}$ satisfies

$$\rho(Q_{\zeta,\alpha}) = \frac{\mu_i[(\alpha\zeta + 1) + (\zeta - \alpha)\mu_i]}{(\alpha + \mu_i)(\zeta\mu_i + 1)} \leq \frac{\mu_{\max}[(\alpha\zeta + 1) + (\zeta - \alpha)\mu_{\max}]}{(\alpha + \mu_{\min})(\zeta\mu_{\min} + 1)}, \quad (2.14)$$

with μ_i is the eigenvalue of the matrix $W^{-1}T$, $\mu_{\max}$ and $\mu_{\min}$ are the maximum and minimum eigenvalues, respectively. If the PBS iteration method is convergent, then parameters ζ and α satisfy

$$2\zeta \geq \alpha > 0, or\ 0 < 2\zeta < \alpha \leq 2\zeta + \frac{2(2\zeta^2 + 1)}{\mu_{\max}}.$$

3 The modified Newton-PBS method

In the section, for solving the complex nonlinear systems (1.1), we establish the modified Newton-PBS method. Firstly, let us define some symbols for convenience.

For any vector or matrix v, denote the symbol:

$$\hat{v} = \begin{pmatrix} Re(v) \\ Im(v) \end{pmatrix},$$

where $Re(v)$ and $Im(v)$ represent its real and imaginary parts, respectively.

Function $F(v)$ is expressed as follows:

$$F(v) = M(v) + iN(v).$$

Similarly, $M(v) = Re(F(v))$ and $N(v) = Im(F(v))$. Then, the Jacobian matrix of $F(v)$ can be written as follows:

$$F'(v) = W(v) + iT(v);$$

here, $W(v) = Re(F'(v))$, $T(v) = Im(F'(v))$, and $W(v), T(v) \in \mathbb{R}^{n\times n}$ are real symmetric and positive semi-definite matrices, but at least one of them is positive definite. Without losing generality, we often assume that $W(v)$ is positive definite.

We establish the modified Newton-PBS method by using PBS method to solve the following modified Newton equations:

$$\begin{cases} \mathcal{A}(v_k)d_k = -\mathcal{H}(v_k), & v_{k+\frac{1}{2}} = d_k + v_k, \\ \mathcal{A}(v_k)h_k = -\mathcal{H}(v_{k+\frac{1}{2}}), & v_{k+1} = h_k + v_{k+\frac{1}{2}}, \quad k = 0, 1, 2, \cdots, \end{cases} \quad (3.1)$$

where

$$\mathcal{A}(v) := \begin{pmatrix} W(v) & -T(v) \\ T(v) & W(v) \end{pmatrix}, \quad (3.2)$$

and

$$\mathcal{H}(v) := \begin{pmatrix} M(v) \\ N(v) \end{pmatrix}.$$

We rewrite d_k and h_k as follows:

$$d_k = \begin{pmatrix} y_k \\ z_k \end{pmatrix} = \begin{pmatrix} I & I \\ 0 & \alpha I \end{pmatrix} \begin{pmatrix} \widetilde{y}_k \\ \widetilde{z}_k \end{pmatrix} = \begin{pmatrix} I & I \\ 0 & \alpha I \end{pmatrix} \widetilde{d}_k = \mathcal{B}\widetilde{d}_k, \tag{3.3}$$

$$h_k = \begin{pmatrix} p_k \\ q_k \end{pmatrix} = \begin{pmatrix} I & I \\ 0 & \alpha I \end{pmatrix} \begin{pmatrix} \widetilde{p}_k \\ \widetilde{q}_k \end{pmatrix} = \begin{pmatrix} I & I \\ 0 & \alpha I \end{pmatrix} \widetilde{h}_k = \mathcal{B}\widetilde{h}_k, \tag{3.4}$$

Multiply the matrix $\mathcal{Z}$ left on the two equations of (3.1), respectively. We can get

$$\begin{cases} \mathcal{Z}\mathcal{A}(v_k)d_k = \widetilde{\mathcal{A}}(v_k)\widetilde{d}_k = -\mathcal{Z}\mathcal{H}(v_k) = -\widetilde{\mathcal{H}}(v_k), \\ \mathcal{Z}\mathcal{A}(v_k)h_k = \widetilde{\mathcal{A}}(v_k)\widetilde{h}_k = -\mathcal{Z}\mathcal{H}(v_{k+\frac{1}{2}}) = -\widetilde{\mathcal{H}}(v_{k+\frac{1}{2}}), \end{cases} \tag{3.5}$$

with $\widetilde{\mathcal{A}} = \begin{pmatrix} \zeta T(v_k) + W(v_k) & 0 \\ T(v_k) & \alpha W(v_k) + T(v_k) \end{pmatrix} - \begin{pmatrix} 0 & -[(\zeta\alpha + 1)W(v_k) + (\zeta - \alpha)T(v_k)] \\ 0 & 0 \end{pmatrix}$ and $\widetilde{\mathcal{H}}(v_k) = \begin{pmatrix} M(v_k) + \zeta N(v_k) \\ N(v_k) \end{pmatrix}$, $k = 0, 1, 2, \cdots$.

From equations (2.8) and (3.5), the next iterate v_{k+1} can be approximately generated as follows:

$$\mathcal{M}_{\zeta,\alpha} \begin{pmatrix} \widetilde{y}_{k+1} \\ \widetilde{z}_{k+1} \end{pmatrix} = \mathcal{N}_{\zeta,\alpha} \begin{pmatrix} \widetilde{y}_k \\ \widetilde{z}_k \end{pmatrix} - \begin{pmatrix} M(v_k) + \zeta N(v_k) \\ N(v_k) \end{pmatrix}. \tag{3.6}$$

For convenience, we uniformly refer to the modified Netown-PBS method as MN-PBS method in the rest of the text. Then, we can directly get the MN-PBS method. The algorithm process is as follows.

According to equations (2.10) and (3.5), $\widetilde{d}_{k,l_k}$ and $\widetilde{h}_{k,m_k}$ have the following uniform expressions:

$$\widetilde{d}_{k,l_k} = -\sum_{j=0}^{l_k - 1} Q^j_{\zeta,\alpha}(v_k) G_{\zeta,\alpha}(v_k) \widetilde{\mathcal{H}}(v_k),$$

$$\widetilde{h}_{k,m_k} = -\sum_{j=0}^{m_k - 1} Q^j_{\zeta,\alpha}(v_k) G_{\zeta,\alpha}(v_k) \widetilde{\mathcal{H}}(v_{k+\frac{1}{2}}),$$

where

$$Q_{\zeta,\alpha}(v) = \begin{pmatrix} \zeta T(v) + W(v) & 0 \\ T(v) & \alpha W(v) + T(v) \end{pmatrix}^{-1} \begin{pmatrix} 0 & -[(\zeta\alpha + 1)W(v) + (\zeta - \alpha)T(v)] \\ 0 & 0 \end{pmatrix},$$

Algorithm 2 The MN-PBS method for solving nonlinear systems (1.1)

1. Given an initialized $v_0 \in \mathbb{D}$. Set two positive constant sequences $\{l_k\}_{k=0}^{\infty}$, $\{m_k\}_{k=0}^{\infty}$ and real positive constants ζ, α, tol.
2. While $\|\mathcal{H}(v_k)\| \geq tol\|\mathcal{H}(v_0)\|$, for $k = 0, 1, \cdots$ create:

2.1 Set $\widetilde{d}_{k,0} = (\widetilde{y}_{k,0}^T, \widetilde{z}_{k,0}^T)^T$, where $\widetilde{y}_{k,0} = 0, \widetilde{z}_{k,0} = 0$.

2.2 For $l = 0, 1, \cdots, l_k - 1$, the PBS method is used to solve the first equation of (3.5).

$$\begin{cases} (\zeta T(v_k) + W(v_k))\widetilde{y}_{k,l+1} = -[(\zeta\alpha + 1)W(v_k) + (\zeta - \alpha)T(v_k)]\widetilde{z}_{k,l} - M(v_k) - \zeta N(v_k), \\ (\alpha W(v_k) + T(v_k))\widetilde{z}_{k,l+1} = -T(v_k)\widetilde{y}_{k,l+1} - N(v_k), \\ y_{k,l+1} = \widetilde{y}_{k,l+1} + \widetilde{z}_{k,l+1}, \qquad z_{k,l+1} = \alpha\widetilde{z}_{k,l+1}, \end{cases}$$

then order the $d_{k,l_k} = ({y_{k,l_k}}^T, {z_{k,l_k}}^T)^T$ to satisfy

$$\|\mathcal{H}(v_k) + \mathcal{A}(v_k)d_{k,l_k}\| \leq \sigma_k\|\mathcal{H}(v_k)\| \text{ for some } \sigma_k \in [0, 1).$$

2.3 Set $v_{k+\frac{1}{2}} = v_k + d_{k,l_k}$.

2.4 Implement $\mathcal{H}(v_{k+\frac{1}{2}})$.

2.5 Set $\widetilde{h}_{k,0} = (\widetilde{p}_{k,0}^T, \widetilde{q}_{k,0}^T)^T$, where $\widetilde{p}_{k,0} = 0, \widetilde{q}_{k,0} = 0$.

2.6 For $m = 0, 1, \cdots, m_k - 1$, the PBS method is used to solve the second equation of (3.5).

$$\begin{cases} (\zeta T(v_k) + W(v_k))\widetilde{p}_{k,m+1} = -[(\zeta\alpha + 1)W(v_k) + (\zeta - \alpha)T(v_k)]\widetilde{q}_{k,m} - M(v_{k+\frac{1}{2}}) - \zeta N(v_{k+\frac{1}{2}}), \\ (\alpha W(v_k) + T(v_k))\widetilde{q}_{k,m+1} = -T(v_k)\widetilde{p}_{k,m+1} - N(v_{k+\frac{1}{2}}), \\ p_{k,m+1} = \widetilde{p}_{k,m+1} + \widetilde{q}_{k,m+1}, \qquad q_{k,m+1} = \alpha\widetilde{q}_{k,m+1}, \end{cases}$$

then obtain the $h_{k,m_k} = ({p_{k,m_k}}^T, {q_{k,m_k}}^T)^T$ such that

$$\|\mathcal{H}(v_{k+\frac{1}{2}}) + \mathcal{A}(v_k)h_{k,m_k}\| \leq \bar{\sigma}_k\|\mathcal{H}(v_{k+\frac{1}{2}})\| \text{ for some } \bar{\sigma_k} \in [0, 1).$$

2.7 Set $v_{k+1} = v_{k+\frac{1}{2}} + h_{k,m_k}$.

3. End.

and

$$G_{\zeta,\alpha} = \begin{pmatrix} \zeta T(v) + W(v) & 0 \\ T(v) & \alpha W(v) + T(v) \end{pmatrix}^{-1},$$

then $d_{k,l_k} = \mathcal{B}\widetilde{d}_{k,l_k}$, $h_{k,m_k} = \mathcal{B}\widetilde{h}_{k,m_k}$ can be gotten.

Consequently, the MN-PBS method is restructured into the following form:

$$\begin{cases} v_{k+\frac{1}{2}} = v_k - \sum_{j=0}^{l_k-1} \mathcal{B}Q_{\zeta,\alpha}^j(v_k)G_{\zeta,\alpha}(v_k)\widetilde{\mathcal{H}}(v_k), \\ v_{k+1} = v_{k+\frac{1}{2}} - \sum_{j=0}^{m_k-1} \mathcal{B}Q_{\zeta,\alpha}^j(v_k)G_{\zeta,\alpha}(v_k)\widetilde{\mathcal{H}}(v_{k+\frac{1}{2}}), , \quad k = 0, 1, 2, \cdots. \end{cases} \tag{3.7}$$

Define the following matrix splitting:

$$\widetilde{\mathcal{A}}(v) = \mathcal{M}_{\zeta,\alpha}(v) - \mathcal{N}_{\zeta,\alpha}(v),$$

with

$$\mathcal{M}_{\zeta,\alpha}(v) = \begin{pmatrix} \zeta T(v) + W(v) & 0 \\ T(v) & \alpha W(v) + T(v) \end{pmatrix},$$
$$\mathcal{N}_{\zeta,\alpha}(v) = \begin{pmatrix} 0 & -[(\zeta\alpha + 1)W(v) + (\zeta - \alpha)T(v)] \\ 0 & 0 \end{pmatrix}.$$

Obviously, the following relationships hold

$$Q_{\zeta,\alpha}(v) = \mathcal{M}_{\zeta,\alpha}(v)^{-1}\mathcal{N}_{\zeta,\alpha}(v), \quad G_{\zeta,\alpha}(v) = \mathcal{M}_{\zeta,\alpha}(v)^{-1}, \tag{3.8}$$

and

$$\widetilde{\mathcal{A}}(v)^{-1} = (\mathcal{M}_{\zeta,\alpha}(v) - \mathcal{N}_{\zeta,\alpha}(v))^{-1} = (G_{\zeta,\alpha}(v)^{-1}(I - Q_{\zeta,\alpha}(v)))^{-1} = (I - Q_{\zeta,\alpha}(v))^{-1}G_{\zeta,\alpha}(v). \tag{3.9}$$

Then, the equivalent form of Eq. (3.7) can be obtained:

$$\begin{cases} v_{k+\frac{1}{2}} = v_k - \mathcal{B}(I - Q_{\zeta,\alpha}(v_k)^{l_k})\widetilde{\mathcal{A}}(v_k)^{-1}\widetilde{\mathcal{H}}(v_k), \\ v_{k+1} = v_{k+\frac{1}{2}} - \mathcal{B}(I - Q_{\zeta,\alpha}(v_k)^{m_k})\widetilde{\mathcal{A}}(v_k)^{-1}\widetilde{\mathcal{H}}(v_{k+\frac{1}{2}}), \quad k = 0, 1, 2, \cdots. \end{cases} \tag{3.10}$$

4 Local convergence of the modified Newton-PBS method

In this section, we mainly derive the local convergence of the modified Newton-PBS method. All the derivation processes and results are carried out under the *Hölder* continuous condition, which is weaker than *Lipschitz* continuity hypothesis. Before theoretical derivation, several necessary definitions and lemmas should be introduced.

Definition 4.1 A mapping $F : \mathbb{D} \subset \mathbb{C}^n \to \mathbb{C}^n$ is nonlinear, if existing a linear operator $B \in \mathbb{L}(\mathbb{R}^n, \mathbb{R}^n)$ satisfies

$$\lim_{s \to 0} \frac{1}{s}\|F(v + sh) - F(v) - sBh\| = 0.$$

for any $h \in \mathbb{C}^n$, then F is Gateaux differentiable (or G-differentiable) at an interior point v of $\mathbb{D}$. Simultaneously, $F : \mathbb{D} \subset \mathbb{C}^n \to \mathbb{C}^n$ is said to be G-differentiable on an open set $D_0 \subset \mathbb{D}$, if it is G-differentiable at any point in D_0.

Lemma 4.1 *For matrices $P, Q \in \mathbb{C}^{n\times n}$ and suppose P is nonsingular satisfying $\|P^{-1}\| \le \epsilon$. If $\|P - Q\| \le \tau$ and $\epsilon\tau \le 1$, then Q is also nonsingular. And*

$$\|Q^{-1}\| \le \frac{\epsilon}{1 - \epsilon\tau}.$$

Before we start the proof, some basic conditions are established. Presume that function $F : \mathbb{D} \subset \mathbb{C}^n \to \mathbb{C}^n$ is continuous and G-differentiable in the neighborhood $\mathbb{D}_0 \subset \mathbb{D}$, which is centered on $v_* \in \mathbb{D}$. Its Jacobian matrix $F'(v)$ is continuous and symmetric, and it can be expressed in the form of the block two-by-two like formula (3.2). In addition, $\mathbb{N}(v_*, r)$ stands for an open ball centered at v_* with radius of $r > 0$, and $F(v_*) = 0$.

Assumption 4.1 Presume that $v_* \in \mathbb{D}_0$ and for arbitrary $v \in \mathbb{N}(v_*, r) \subset \mathbb{D}_0$, all the following conditions are established.

(A1) (The Bounded Condition) There are positive constants δ and γ, satisfying

$$\max \left\{ \|W(v_*)\|,\ \|T(v_*)\| \right\} \leq \delta \ \text{ and } \ \|\mathcal{A}(v_*)^{-1}\| \leq \gamma.$$

(A2) (The *Hölder* Condition) There are nonnegative constants R_w and R_t, for the index $p \in (0, 1]$, satisfying

$$\begin{aligned} \|W(v) - W(v_*)\| &\leq R_w \|v - v_*\|^p, \\ \|T(v) - T(v_*)\| &\leq R_t \|v - v_*\|^p. \end{aligned}$$

Based on *Assumptions* 4.1, we establish the local convergence theorem of the MN-PBS method. To ensure the convergence of the MN-PBS method, we set the appropriate radius r of the neighborhood $\mathbb{N}(v_*, r)$ and give the restrictions of behavior of function F at solution point v_*.

Lemma 4.2 *Assumption 4.1 holds, set* $R := R_w + R_t$*, if* $r \in \left(0, 1/(\gamma R)^{\frac{1}{p}}\right)$*, and* $p \in (0, 1]$*,* $\mathcal{A}(v)$ *is nonsingular. For any* $v, u \in \mathbb{N}(v_*, r) \in \mathbb{D}_0$*, the following inequalities hold:*

$$\begin{aligned} &\|\mathcal{A}(v) - \mathcal{A}(v_*)\| \leq R\|v - v_*\|^p, \\ &\|\mathcal{A}(v)^{-1}\| \leq \frac{\gamma}{1 - \gamma R\|v - v_*\|^p}, \\ &\|\mathcal{H}(u)\| \leq \frac{R}{p+1}\|u - v_*\|^{p+1} + 2\delta\|u - v_*\|, \\ &\|u - v_* - \mathcal{A}(v)^{-1}\mathcal{H}(u)\| \leq \frac{\gamma}{1 - \gamma R\|v - v_*\|^p}\left(\frac{R}{p+1}\|u - v_*\|^p + R\|v - v_*\|^p\right) \\ &\|u - v_*\|. \end{aligned}$$

Proof We can obtain directly from the *Hölder* condition

$$\begin{aligned} \left\|\mathcal{A}(v) - \mathcal{A}(v_*)\right\| &= \left\|\begin{pmatrix} W(v) - W(v_*) & T(v_*) - T(v) \\ T(v) - T(v_*) & W(v) - W(v_*) \end{pmatrix}\right\| \\ &\leq \left\|\begin{pmatrix} W(v) - W(v_*) & 0 \\ 0 & W(v) - W(v_*) \end{pmatrix}\right\| \\ &\quad + \left\|\begin{pmatrix} 0 & T(v_*) - T(v) \\ T(v) - T(v_*) & 0 \end{pmatrix}\right\| \\ &= \|W(v) - W(v_*)\| + \|T(v) - T(v_*)\| \end{aligned}$$

$$\begin{aligned}&\leq R_w\|v-v_*\|^p+R_t\|v-v_*\|^p\\&=R\|v-v_*\|^p.\end{aligned}$$

Furthermore, we can be obtained

$$\begin{aligned}\|\mathcal{A}(v_*)^{-1}(\mathcal{A}(v_*)-\mathcal{A}(v))\|&\leq\|\mathcal{A}(v_*)^{-1}\|\|\mathcal{A}(v_*)-\mathcal{A}(v)\|\\&\leq\gamma R\|v-v_*\|^p<1,\end{aligned}$$

due to $\mathcal{A}(v_*)^{-1}$ is nonsingular, and by making use of Lemma 4.1, we can get

$$\begin{aligned}\|\mathcal{A}(v)^{-1}\|&\leq\frac{\|\mathcal{A}(v_*)^{-1}\|}{1-\|\mathcal{A}(v_*)^{-1}(\mathcal{A}(v_*)-\mathcal{A}(v))\|}\\&\leq\frac{\gamma}{1-\gamma R\|v-v_*\|^p}.\end{aligned}$$

Since

$$\begin{aligned}\mathcal{H}(u)&=\mathcal{H}(u)-\mathcal{H}(v_*)-\mathcal{A}(v_*)(u-v_*)+\mathcal{A}(v_*)(u-v_*)\\&=\int_0^1(\mathcal{A}(v_*+t(u-v_*))-\mathcal{A}(v_*))dt(u-v_*)+\mathcal{A}(v_*)(u-v_*),\end{aligned}$$

from the bounded condition, we have

$$\begin{aligned}\left\|\mathcal{A}(v_*)\right\|&\leq\left\|\begin{pmatrix}W(v_*)&0\\0&W(v_*)\end{pmatrix}\right\|+\left\|\begin{pmatrix}0&-T(v_*)\\T(v_*)&0\end{pmatrix}\right\|\\&=\|W(v_*)\|+\|T(v_*)\|\\&\leq 2\delta,\end{aligned}$$

consequently

$$\begin{aligned}\left\|\mathcal{H}(u)\right\|&\leq\|\int_0^1(\mathcal{A}(v_*+t(u-v_*)))-\mathcal{A}(v_*))dt((u-v_*)\right\|+\left\|\mathcal{A}(v_*)(u-v_*)\right\|\\&\leq\int_0^1\left\|\mathcal{A}(v_*+t(u-v_*)))-\mathcal{A}(v_*)\right\|dt((u-v_*)+\left\|\mathcal{A}(v_*)\right\|\left\|(u-v_*)\right\|\\&\leq\int_0^1R\|t(u-v_*)\|^pdt((u-v_*)+2\delta\|u-v_*\|\leq\frac{R}{p+1}\|u-v_*\|^{p+1}+2\delta\|u-v_*\|.\end{aligned}$$

For the last equation, let's make a transformation

$$\begin{aligned}&u-v_*-\mathcal{A}(v)^{-1}\mathcal{H}(u)\\&=-\mathcal{A}(v)^{-1}(\mathcal{H}(u)-\mathcal{H}(v_*)-\mathcal{A}(v_*)(u-v_*)+\mathcal{A}(v_*)(u-v_*)-\mathcal{A}(v)(u-v_*))\\&=-\mathcal{A}(v)^{-1}\Big(\int_0^1(\mathcal{A}(v_*+t(u-v_*))-\mathcal{A}(v_*))dt(u-v_*)+(\mathcal{A}(v)-\mathcal{A}(v_*))(u-v_*)\Big),\end{aligned}$$

then

$$
\begin{aligned}
&\|u - v_* - \mathcal{A}(v)^{-1}\mathcal{H}(u)\| \\
&\leq \|\mathcal{A}(v)^{-1}\| \left(\int_0^1 \|(\mathcal{A}(v_* + t(u - v_*)) - \mathcal{A}(v_*))\| dt (u - v_*) + \|\mathcal{A}(v) - \mathcal{A}(v_*)\| \|u - v_*\| \right) \\
&\leq \frac{\gamma}{1 - \gamma R \|v - v_*\|^p} \left(\frac{R}{p+1} \|u - v_*\|^p + R\|v - v_*\|^p \right) \|u - v_*\|.
\end{aligned}
$$

The proof of *Lemma* 3.2 is completed. □

Theorem 4.1 *Suppose $r \in (0, r_0)$, ensure that the assumptions are consistent with Lemma 4.2, set $r_0 := \min_{1 \leq j \leq 3}\{r_j\}$, meanwhile*

$$
\begin{aligned}
r_1 &= \left(\frac{1}{[2(1 + \frac{1}{\alpha})(1 + \frac{1}{\zeta})\gamma][(1 + \alpha)R_w + (2 + \zeta)R_t]} \right)^{\frac{1}{p}}, \\
r_2 &= \left(\frac{\tau\theta}{\left(2(1 + \frac{1}{\alpha})(1 + \frac{1}{\zeta})\gamma\right)\left([(\zeta\alpha + 1) + (1 + \tau\theta)(1 + \alpha)]R_w + [(\zeta - \alpha) + (1 + \tau\theta)(2 + \zeta)]R_t\right)} \right)^{\frac{1}{p}}, \\
r_3 &= \left(\frac{1 - 2\delta\gamma[(\tau + 1)\theta]^{\nu_*}}{4\gamma R} \right)^{\frac{1}{p}},
\end{aligned}
$$

in the expression $\nu_* = \min\{l^*, m^*\}$, $l^* = \liminf_{k\to\infty} l_k$, $m^* = \liminf_{k\to\infty} m_k$, and the constant ν_* such that

$$
\nu_* > \lfloor -\frac{\ln(2\delta\gamma)}{\ln((\tau + 1)\theta)} \rfloor,
$$

where the smallest integer not less than the corresponding real number is represented by the symbol $\lfloor\cdot\rfloor$, and $\tau \in \left(0, \frac{1-\theta}{\theta}\right)$ is a predetermined constant, besides

$$
\theta \equiv \theta_{\zeta,\alpha}(v_*) = \|Q_{\zeta,\alpha}(v_*)\| \leq \frac{\mu_{\max}[(\alpha\zeta + 1) + (\zeta - \alpha)\mu_{\max}]}{(\alpha + \mu_{\min})(\zeta\mu_{\min} + 1)} \equiv \vartheta_{\zeta,\alpha}(v_*) < 1,
$$

with $\mu_{\max}$ and $\mu_{\min}$ are the maximum and minimum eigenvalues of $W(v_*)^{-1}T(v_*)$, and $2\zeta \geq \alpha > 0$, or $0 < 2\zeta < \alpha \leq 2\zeta + \frac{2(2\zeta^2+1)}{\mu_{\max}}$. Then, for any $v \in \mathbb{N}(v_*, r) \subset \mathbb{N}_0$, and sequences $\{l_k\}_{k=0}^{\infty}$, $\{m_k\}_{k=0}^{\infty}$ of positive integers, the iteration sequence $\{v_k\}_{k=0}^{\infty}$ generated by the MN-PBS method is well-defined and converges to v_*. In addition, the following inequalities can be proved theoretically.

$$
\limsup_{k\to\infty} \|v_k - v_*\|^{\frac{1}{k}} \leq g(r_0^p; \nu_*)^2,
$$

here

$$
g(t^p; \varrho) = \frac{\gamma}{1 - \gamma R t^p}\left(3Rt^p + 2\delta[(\tau + 1)\theta]^{\varrho}\right) \leq g(r_0^p; \nu_*) < 1.
$$

in the formula $t \in (0, r)$ and $\varrho > \nu_*$. Under such conditions, it holds that

$$\|Q_{\alpha,\zeta}(v)\| \leq (\tau+1)\theta < 1.$$

Proof According to the previous literature, the convergence of the algorithm mainly depends on the spectral radius of the iterative matrix $Q_{\zeta,\alpha}(v)$. Moreover, the spectral radius of $Q_{\zeta,\alpha}(v)$ needs to be less than 1 to make the algorithm converge.

According to the previous derivation, we know

$$\widetilde{\mathcal{A}}(v_*) = \begin{pmatrix} I & \zeta I \\ 0 & I \end{pmatrix} \mathcal{A}(v_*) \begin{pmatrix} I & I \\ 0 & \alpha I \end{pmatrix},$$

then

$$\begin{aligned}
\|\widetilde{\mathcal{A}}(v_*)^{-1}\| &= \left\| \begin{pmatrix} I & I \\ 0 & \alpha I \end{pmatrix}^{-1} \right\| \|\mathcal{A}(v_*)^{-1}\| \left\| \begin{pmatrix} I & \zeta I \\ 0 & I \end{pmatrix}^{-1} \right\| \\
&= \left\| \begin{pmatrix} I & -\frac{1}{\alpha} I \\ 0 & \frac{1}{\alpha} I \end{pmatrix} \right\| \|\mathcal{A}(v_*)^{-1}\| \left\| \begin{pmatrix} I & -\frac{1}{\zeta} I \\ 0 & I \end{pmatrix} \right\| \\
&\leq (1+\frac{1}{\alpha})(1+\frac{1}{\zeta})\gamma.
\end{aligned}$$

From the bounded conditions and identities (3.8), (3.9), the result can be get

$$\begin{aligned}
\|\mathcal{M}_{\zeta,\alpha}(v_*)^{-1}\| &= \|(I - Q_{\zeta,\alpha}(v_*))\widetilde{\mathcal{A}}(v_*)^{-1}\| \\
&\leq \big(1 + \|Q_{\zeta,\alpha}(v_*)\|\big)\|\widetilde{\mathcal{A}}(v_*)^{-1}\| \\
&\leq 2(1+\frac{1}{\alpha})(1+\frac{1}{\zeta})\gamma, \qquad (4.1)
\end{aligned}$$

under the $H\ddot{o}lder$ condition, we can get

$$\begin{aligned}
&\|\mathcal{M}_{\zeta,\alpha}(v_*) - \mathcal{M}_{\zeta,\alpha}(v)\| \\
&\leq \left\| \begin{pmatrix} \zeta T(v_*) + W(v_*) & 0 \\ T(v_*) & \alpha W(v_*) + T(v_*) \end{pmatrix} - \begin{pmatrix} \zeta T(v) + W(v) & 0 \\ T(v) & \alpha W(v) + T(v) \end{pmatrix} \right\| \\
&= \left\| \begin{pmatrix} \zeta(T(v_*) - T(v)) + W(v_*) - W(v) & 0 \\ T(v_*) - T(v) & \alpha(W(v_*) - W(v)) + T(v_*) - T(v) \end{pmatrix} \right\| \\
&\leq \|\zeta(T(v_*) - T(v)) + W(v_*) - W(v)\| + \|T(v_*) - T(v)\| + \|\alpha(W(v_*) - W(v)) + T(v_*) - T(v)\| \\
&\leq (2+\zeta)\|T(v_*) - T(v)\| + (1+\alpha)\|W(v_*) - W(v)\| \\
&\leq (1+\alpha)R_w\|v - v_*\|^p + (2+\zeta)R_t\|v - v_*\|^p. \qquad (4.2)
\end{aligned}$$

Consequently, according to $Lemma$ 1, the following conclusion holds:

$$\|\mathcal{M}_{\zeta,\alpha}(v)^{-1}\| \leq \frac{\|\mathcal{M}_{\zeta,\alpha}(v_*)^{-1}\|}{1 - \|\mathcal{M}_{\zeta,\alpha}(v_*)^{-1}\|\|\mathcal{M}_{\zeta,\alpha}(v_*) - \mathcal{M}_{\zeta,\alpha}(v)\|}$$

$$\leq \frac{2(1+\frac{1}{\alpha})(1+\frac{1}{\zeta})\gamma}{1-[2(1+\frac{1}{\alpha})(1+\frac{1}{\zeta})\gamma][(1+\alpha)R_w\|v-v_*\|^p+(2+\zeta)R_t\|v-v_*\|^p]};$$

hence, we need inequality hold:

$$[2(1+\frac{1}{\alpha})(1+\frac{1}{\zeta})\gamma][(1+\alpha)R_w\|v-v_*\|^p+(2+\zeta)R_t\|v-v_*\|^p]\leq 1,$$

then the following inequality is correct:

$$\|v-v_*\|^p\leq\frac{1}{[2(1+\frac{1}{\alpha})(1+\frac{1}{\zeta})\gamma][(1+\alpha)R_w+(2+\zeta)R_t]},$$

since $r < r_1$, so the above results hold. The same can be obtained

$$\begin{aligned}
&\|\mathcal{N}_{\zeta,\alpha}(v_*)-\mathcal{N}_{\zeta,\alpha}(v)\|\\
\leq&\left\|\begin{pmatrix}0 & -[(\zeta\alpha+1)W(v_*)+(\zeta-\alpha)T(v_*)]\\ 0 & 0\end{pmatrix}-\begin{pmatrix}0 & -[(\zeta\alpha+1)W(v)+(\zeta-\alpha)T(v)]\\ 0 & 0\end{pmatrix}\right\|\\
\leq&\|(\zeta\alpha+1)(W(v)-W(v_*))\|+\|(\zeta-\alpha)(T(v_*)-T(v))\|\\
\leq&(\zeta\alpha+1)R_w\|v-v_*\|^p+(\zeta-\alpha)R_t\|v-v_*\|^p.
\end{aligned}$$

For arbitrary $v\in\mathbb{N}(v_*,r)\in\mathbb{D}_0$, if

$$\|Q_{\zeta,\alpha}(v)\|\leq\|Q_{\zeta,\alpha}(v)-Q_{\zeta,\alpha}(v_*)\|+\|Q_{\zeta,\alpha}(v_*)\|\leq\tau\theta+\theta<1,$$

holds, then we need to prove

$$\|Q_{\zeta,\alpha}(v)-Q_{\zeta,\alpha}(v_*)\|\leq\tau\theta.$$

Then,

$$\begin{aligned}
&\|Q_{\zeta,\alpha}(v)-Q_{\zeta,\alpha}(v_*)\|\\
=&\|\mathcal{M}_{\zeta,\alpha}(v)^{-1}\mathcal{N}_{\zeta,\alpha}(v)-\mathcal{M}_{\zeta,\alpha}(v_*)^{-1}\mathcal{N}_{\zeta,\alpha}(v_*)\|\\
=&\|\mathcal{M}_{\zeta,\alpha}(v)^{-1}\mathcal{N}_{\zeta,\alpha}(v)-\mathcal{M}_{\zeta,\alpha}(v)^{-1}\mathcal{N}_{\zeta,\alpha}(v_*)+\mathcal{M}_{\zeta,\alpha}(v)^{-1}\mathcal{N}_{\zeta,\alpha}(v_*)-\mathcal{M}_{\zeta,\alpha}(v_*)^{-1}\mathcal{N}_{\zeta,\alpha}(v_*)\|\\
=&\|\mathcal{M}_{\zeta,\alpha}(v)^{-1}(\mathcal{N}_{\zeta,\alpha}(v)-\mathcal{N}_{\zeta,\alpha}(v_*))+(\mathcal{M}_{\zeta,\alpha}(v)^{-1}-\mathcal{M}_{\zeta,\alpha}(v_*)^{-1})\mathcal{N}_{\zeta,\alpha}(v_*)\|\\
=&\|\mathcal{M}_{\zeta,\alpha}(v)^{-1}(\mathcal{N}_{\zeta,\alpha}-\mathcal{N}_{\zeta,\alpha}(v_*))+(\mathcal{M}_{\zeta,\alpha}(v)^{-1}\mathcal{M}_{\zeta,\alpha}(v_*)-I)Q_{\zeta,\alpha}(v_*)\|\\
\leq&\|\mathcal{M}_{\zeta,\alpha}(v)^{-1}\|\Big(\|(\mathcal{N}_{\zeta,\alpha}(v)-\mathcal{N}_{\zeta,\alpha}(v_*))+(\mathcal{M}_{\zeta,\alpha}(v_*)-\mathcal{M}_{\zeta,\alpha}(v))Q_{\zeta,\alpha}(v_*)\|\Big)\\
\leq&\|\mathcal{M}_{\zeta,\alpha}(v)^{-1}\|\Big(\|\mathcal{N}_{\zeta,\alpha}(v)-\mathcal{N}_{\zeta,\alpha}(v_*)\|+\|\mathcal{M}_{\zeta,\alpha}(v_*)-\mathcal{M}_{\zeta,\alpha}(v)\|\|Q_{\zeta,\alpha}(v_*)\|\Big)\\
\leq&\frac{\Big((1+\frac{1}{\alpha})(1+\frac{1}{\zeta})\gamma\Big)\Big((\zeta\alpha+\alpha+2)R_w\|v-v_*\|^p+(2+2\zeta-\alpha)R_t\|v-v_*\|^p\Big)}{1-[(1+\frac{1}{\alpha})(1+\frac{1}{\zeta})\gamma][(1+\alpha)R_w\|v-v_*\|^p+(2+\zeta)R_t\|v-v_*\|^p]}.
\end{aligned}\tag{4.3}$$

Hence,

$$\frac{\left(2(1+\frac{1}{\alpha})(1+\frac{1}{\zeta})\gamma\right)\left((\zeta\alpha+\alpha+2)R_w\|v-v_*\|^p+(2+2\zeta-\alpha)R_t\|v-v_*\|^p\right)}{1-[2(1+\frac{1}{\alpha})(1+\frac{1}{\zeta})\gamma][(1+\alpha)R_w\|v-v_*\|^p+(2+\zeta)R_t\|v-v_*\|^p]}\leq\tau\theta, \quad (4.4)$$

then we can get

$$\|v-v_*\|^p\leq\frac{\tau\theta}{\left(2(1+\frac{1}{\alpha})(1+\frac{1}{\zeta})\gamma\right)\left([(\zeta\alpha+1)+(1+\tau\theta)(1+\alpha)]R_w+[(\zeta-\alpha)+(1+\tau\theta)(2+\zeta)]R_t\right)}. \quad (4.5)$$

So when $r < \min\{r_1, r_2\}$, for arbitrary $v \in \mathbb{N}(v_*, r)$, the inequality $\|Q_{\alpha,\zeta}(v)\| \leq (\tau+1)\theta < 1$ holds.

Next, we mainly analyze the error of the obtained iteration sequence $\{v\}_{k}{}_{k=0}^{\infty}$. Firstly, we know

$$\widetilde{\mathcal{A}}(v_k)^{-1}=\mathcal{B}^{-1}\mathcal{A}(v_k)^{-1}\mathcal{Z}^{-1}=\begin{pmatrix} I & I\\ 0 & \alpha I\end{pmatrix}^{-1}\mathcal{A}(v_k)^{-1}\begin{pmatrix} I & \zeta I\\ 0 & I\end{pmatrix}^{-1},$$

$$\widetilde{\mathcal{H}}(v_k)=\mathcal{Z}\mathcal{H}(v_k)=\begin{pmatrix} I & \zeta I\\ 0 & I\end{pmatrix}\mathcal{H}(v_k),$$

then

$$\widetilde{\mathcal{A}}(v_k)^{-1}\widetilde{\mathcal{H}}(v_k)=\mathcal{B}^{-1}\mathcal{A}(v_k)^{-1}\mathcal{Z}^{-1}\mathcal{Z}\mathcal{H}(v_k)=\mathcal{B}^{-1}\mathcal{A}(v_k)^{-1}\mathcal{H}(v_k). \quad (4.6)$$

From Eqs. (3.10) and (4.7) and Lemma 4.2, the following equation can be obtained:

$$\begin{aligned}
\|v_{k+\frac{1}{2}}-v_*\| &= \|v_k-v_*-\mathcal{B}(I-Q_{\zeta,\alpha}(v_k)^{l_k})\widetilde{\mathcal{A}}(v_k)^{-1}\widetilde{\mathcal{H}}(v_k)\|\\
&= \|v_k-v_*-\mathcal{B}(I-Q_{\zeta,\alpha}(v_k)^{l_k})\mathcal{B}^{-1}\mathcal{A}(v_k)^{-1}\mathcal{H}(v_k)\|\\
&\leq \|v_k-v_*-\mathcal{A}(v_k)^{-1}\mathcal{H}(v_k)\|+\|\mathcal{B}\|.\|Q_{\zeta,\alpha}(v_k)\|^{l_k}.\|\mathcal{B}^{-1}\|.\ \|\mathcal{A}(v_k)^{-1}\mathcal{H}(v_k)\|\\
&\leq \|v_k-v_*-\mathcal{A}(v_k)^{-1}\mathcal{H}(v_k)\|+\|Q_{\zeta,\alpha}(v_k)\|^{l_k}.\|\mathcal{A}(v_k)^{-1}\mathcal{H}(v_k)\|\\
&\leq \frac{\gamma R(p+2)}{(p+1)(1-\gamma R\|v_k-v_*\|^p)}\|v_k-v_*\|^{p+1}\\
&\quad+\frac{\gamma[(\tau+1)\theta]^{l_k}}{1-\gamma R\|v_k-v_*\|^p}\left(\frac{R}{p+1}\|v_k-v_*\|^{p+1}+2\delta\|v_k-v_*\|\right)\\
&\leq \frac{\gamma R(p+2)+\gamma R[(\tau+1)\theta]^{l_k}}{(p+1)(1-\gamma R\|v_k-v_*\|^p)}\|v_k-v_*\|^{p+1}+\frac{2\delta\gamma[(\tau+1)\theta]^{l_k}\|v_k-v_*\|}{1-\gamma R\|v_k-v_*\|^p}\\
&\leq \frac{\gamma}{1-\gamma R\|v_k-v_*\|^p}\left(\frac{R(p+2+[(\tau+1)\theta]^{l_k})}{p+1}\|v_k-v_*\|^p+2\delta[(\tau+1)\theta]^{l_k}\right)\|v_k-v_*\|\\
&\leq \frac{\gamma}{1-\gamma R\|v_k-v_*\|^p}\left(3R\|v_k-v_*\|^p+2\delta[(\tau+1)\theta]^{l_k}\right)\|v_k-v_*\|\\
&= g(\|v_k-v_*\|^p;l_k)\|v_k-v_*\|.
\end{aligned}$$

Obviously, when $0 < p < 1$, $t \in (0, r_0)$ and $\varrho > \nu_*$, the function $g(t^p; \varrho) = \frac{\gamma}{1-\gamma R t^p}\left(3Rt^p + 2\delta[(\tau+1)\theta]^{\varrho}\right)$ is strictly monotonically decreasing with the increase of independent variable ϱ. In addition, the function $g(t^p; \varrho)$ takes partial derivative of t:

$$\frac{\partial g(t^p; \varrho)}{\partial t} = \frac{\gamma p R t^{p-1}[3 + 2\gamma\delta[(\tau+1)\theta]^{\varrho}]}{(1-\gamma R t^p)^2} > 0.$$

Easy to get that the function $g(t^p; \varrho)$ is strictly monotonically increasing with respect to the independent variable t.

Set variables $\nu_* = \min\{l^*, m^*\}$, $l^* = \liminf_{k\to\infty} l_k$, $m^* = \liminf_{k\to\infty} m_k$, if the inequality

$$g(\|v_k - v_*\|^p; l_k) \leq \frac{\gamma}{1-\gamma R r^p}\left(3Rr^p + 2\delta[(\tau+1)\theta]^{\nu_*}\right) = g(r^p; \nu_*) < 1$$

holds for any $v_k \in \mathbb{N}(v_*, r)$. That

$$r^p \leq \frac{1 - 2\gamma\delta[(\tau+1)\theta]^{\nu_*}}{4\gamma R},$$

then

$$2\gamma\delta[(\tau+1)\theta]^{\nu_*} \leq 1, \qquad r \leq \left(\frac{1 - 2\gamma\delta[(\tau+1)\theta]^{\nu_*}}{4\gamma R}\right)^{\frac{1}{p}}.$$

Consequently, when $r < r_3$,

$$\|v_{k+\frac{1}{2}} - v_*\| < \|v_k - v_*\|.$$

The same inequality can be obtained

$$\begin{aligned}
\|v_{k+1} - v_*\| &= \|v_{k+\frac{1}{2}} - v_* - \mathcal{B}(I - Q_{\zeta,\alpha}(v_k)^{m_k})\widetilde{\mathcal{A}}(v_k)^{-1}\widetilde{\mathcal{H}}(v_{k+\frac{1}{2}})\| \\
&= \|v_{k+\frac{1}{2}} - v_* - \mathcal{B}(I - Q_{\zeta,\alpha}(v_k)^{m_k})\mathcal{B}^{-1}\mathcal{A}(v_k)^{-1}\mathcal{H}(v_{k+\frac{1}{2}})\| \\
&\leq \|v_{k+\frac{1}{2}} - v_* - \mathcal{A}(v_k)^{-1}\mathcal{H}(v_{k+\frac{1}{2}})\| + \|Q_{\zeta,\alpha}(v_k)\|^{m_k} \cdot \|\mathcal{A}(v_k)^{-1}\mathcal{H}(v_{k+\frac{1}{2}})\| \\
&\leq \frac{\gamma}{1-\gamma R\|v_k - v_*\|^p}\left(\frac{R}{p+1}\|v_{k+\frac{1}{2}} - v_*\|^p + R\|v_k - v_*\|^p\right)\|v_{k+\frac{1}{2}} - v_*\| \\
&\quad + \frac{\gamma[(\tau+1)\theta]^{m_k}}{1-\gamma R\|v_k - v_*\|^p}\left(\frac{R}{p+1}\|v_{k+\frac{1}{2}} - v_*\|^{p+1} + 2\delta\|v_{k+\frac{1}{2}} - v_*\|\right) \\
&\leq \frac{\gamma R}{1-\gamma R\|v_k - v_*\|^p}\left(\frac{1 + [(\tau+1)\theta]^{m_k}}{p+1}\|v_{k+\frac{1}{2}} - v_*\|^{p+1}\right) \\
&\quad + \frac{\gamma}{1-\gamma R\|v_k - v_*\|^p}\left(R\|v_k - v_*\|^p + 2\delta[(\tau+1)\theta]^{m_k}\right)\|v_{k+\frac{1}{2}} - v_*\| \\
&\leq \frac{\gamma g(\|v_k - v_*\|^p; l_k)}{1-\gamma R\|v_k - v_*\|^p}\|v_k - v_*\| \\
&\quad \times\left(\frac{g(\|v_k - v_*\|^p; l_k)^p\left(1 + [(\tau+1)\theta]^{m_k}\right) + 1 + p}{p+1}R\|v_k - v_*\|^p + 2\delta[(\tau+1)\theta]^{m_k}\right) \\
&\leq \frac{\gamma g(\|v_k - v_*\|^p; l_k)}{1-\gamma R\|v_k - v_*\|^p} \times \left(\left(2g(\|v_k - v_*\|^p; l_k)^p + 1\right)R\|v_k - v_*\|^p + 2\delta[(\tau+1)\theta]^{m_k}\right)\|v_k - v_*\|
\end{aligned}$$

$$\leq \frac{\gamma g(\|v_k - v_*\|^p; l_k)}{1 - \gamma R\|v_k - v_*\|^p} \times \left(3R\|v_k - v_*\|^p + 2\delta[(\tau+1)\theta]^{m_k}\right)\|v_k - v_*\|$$
$$\leq g(\|v_k - v_*\|^p; l_k) g(\|v_k - v_*\|^p; m_k)\|v_k - v_*\|$$
$$< g(r; \nu_*)^2\|v_k - v_*\| < \|v_k - v_*\|.$$

For arbitrary $v_0 \in \mathbb{N}(v_*, r)$, $g(r; \nu_*) < 1$ holds, so the inequality is correct

$$0 \leq \|v_{k+1} - v_*\| < \|v_k - v_*\| < \cdots < \|v_0 - v_*\| < r.$$

Therefore, the iteration sequence $\{v_k\}_{k=0}^{\infty}$ is well-defined, and it converges to v_*. Due to $\|v_{k+1} - v_*\| < g(r_0; \nu_*)^2\|v_k - v_*\|$, then the inequality $\|v_k - v_*\| < g(r_0; \nu_*)^{2k}\|v_0 - v_*\|$ is true, it is equivalent to

$$\|v_k - v_*\|^{\frac{1}{k}} < g(r_0; \nu_*)^2\|v_0 - v_*\|^{\frac{1}{k}},$$

so $\lim_{k\to\infty} \sup \|v_k - v_*\|^{\frac{1}{k}} \leq g(r_0^p; v_*)^2$ holds, when $k \to \infty$.

The proof of $Theorem$ 4.1 is completed. □

Corollary 4.1 Under the condition of Theorem 4.1, the iteration sequence $\{v_k\}_{k=0}^{\infty}$ generated by the MN-PBS method converges to v_*. For any $v \in \mathbb{N}(v_*, r) \subset \mathbb{D}_0$ and sequences $\{l_k\}_{k=0}^{\infty}$, $\{m_k\}_{k=0}^{\infty}$ of positive integers, the convergence rate of MN-PBS is super-linear if

$$\lim_{k\to\infty} \sup \|Q_{\zeta,\alpha}(v_k)^{\nu_*}\| = 0,$$

where $\nu_* = \min\{l^*, m^*\}$, $l^* = \liminf_{k\to\infty} l_k$, $m^* = \liminf_{k\to\infty} m_k$.

Proof Similar to the proof of $Theorem$ 5.1 in reference [57], we omit the proof. □

Corollary 4.2 Under the condition of Theorem 4.1, the iteration sequence $\{v_k\}_{k=0}^{\infty}$ generated by the MN-PBS method converges to v_*. For any $v \in \mathbb{N}(v_*, r) \subset \mathbb{D}_0$ and sequences $\{l_k\}_{k=0}^{\infty}$, $\{m_k\}_{k=0}^{\infty}$ of positive integers, the convergence order is of MN-PBS at least of $1 + 2p$ if

$$\|Q_{\zeta,\alpha}(v_k)^{\nu_*}\| = O(F(v_k)^p),$$

where $\nu_* = \min\{l^*, m^*\}$, $l^* = \liminf_{k\to\infty} l_k$, $m^* = \liminf_{k\to\infty} m_k$.

Proof Similar to the proof of $Theorem$ 5.2 in reference [57], we omit the proof. □

5 Numerical examples

In the section, we enumerate two complex nonlinear equations that often appear in practice, and we use MN-PBS and some previously known methods to obtain their numerical solutions, then we compare the experimental results to verify the efficiency and feasibility of our method. The previously known methods we have implemented include MN-DGPMHSS, MN-GSOR, MN-AGSOR, MN-DSS, and MN-FPAE, which use the modified Newton method as external iteration and DGPMHSS, GSOR, AGSOR, DSS, and FPAE as internal iteration, respectively. In order to ensure the reliability of the results, we not only compared their internal and external iteration steps, but also compared the CPU time consumption of the algorithm. The optimal parameters of each method and the details of algorithm results are listed in the following tables for comparison and reference. It is worth mentioning that the numerical experimental results of all methods of each example are carried out with MATLAB Version R2021a on a computer, and the hardware conditions of the computer include a windows had 1.30 GHz Intel Core i7-1065G7 CPU and 16.00 GB RAM, and the accuracy of the machine is $eps = 2.22 \times 10^{-16}$.

We have listed the detailed experimental results of numerical examples in the following table. From the table, we can clearly get error estimates, outer and internal iteration steps, and CPU time in seconds of each method. For convenience, we abbreviate error estimates, outer and internal iteration steps, and CPU time as REST, OT Step and IT Step, and CPUT(s), respectively. For Examples 5.1 and 5.2, the external iteration stopping criterion of all methods needs to be satisfied:

$$\frac{||\mathcal{H}(v_k)||_2}{||\mathcal{H}(v_0)||_2} \leq 10^{-10}.$$

The stop tolerances of inner iteration σ_k and $\bar{\sigma}_k$ in two steps of all methods are set to σ. Then, the following inequalities hold:

$$\frac{\|\mathcal{H}(v_k) + \mathcal{A}(v_k)d_{k,l_k}\|}{\|\mathcal{H}(v_k)\|} \leq \sigma,$$

$$\frac{\|\mathcal{H}(v_{k+\frac{1}{2}}) + \mathcal{A}(v_k)h_{k,m_k}\|}{\|\mathcal{H}(v_{k+\frac{1}{2}})\|} \leq \sigma.$$

Example 5.1 Firstly, consider the following nonlinear systems:

$$\begin{cases} v_t - (\alpha_1 + i\beta_1)(v_{xx} + v_{yy}) + \kappa v = -(\alpha_2 + i\beta_2)v^{\frac{4}{3}}, & in \ \ (0, 1] \times \Omega, \\ v(0, x, y) = v_0(x, y), & in \ \ \Omega, \\ v(t, x, y) = 0, & on \ \ (0, 1] \times \partial\Omega, \end{cases}$$

where $(x, y) \in \Omega = (0, 1) \times (0, 1)$, and its boundary is represented by $\partial\Omega$. For Example 5.1, we set the original conjecture point $v_0 = 1$; we select values of parameters $\alpha_1 = 1, \alpha_2 = -1, \beta_1 = \beta_2 = 0.5$, and κ is a positive constant, which controls the magnitude of the reaction term. In order to reflect the numerical experimental results better, we choose several different values for κ. We wield the central finite difference scheme to discretize the grid of the partial differential equations, in which the equidistant grid is selected $\Delta t = h = 1/(N+1)$, then we can acquire the following nonlinear equation as form (1.6)

$$F(v) = Dv + (\alpha_2 + i\beta_2)h\Delta t\psi(v) = 0, \tag{5.1}$$

with

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

$$\psi(v) = (v_1^{\frac{4}{3}}, v_2^{\frac{4}{3}}, ..., v_n^{\frac{4}{3}})^T,$$

where the vector $v = (v_1, v_2, ..., v_n)^T$, and A_N is a tridiagonal matrix satisfying $A_N = tridiag(-1, 2, -1) \in R^{N\times N}, n = N \times N, \otimes$ is used to represent the Kronecker product, and N is a positive integer representing the scale of the problem.

Firstly, before discussing the comparison of numerical results, we need to explore the process of selecting the optimal parameters ζ, α. As for the selection of the optimal parameters of the method with only one parameter, Bai et al. have carried out relevant theoretical research in the previous documents. However, there is no complete theoretical derivation for the optimal selection of two parameters. Therefore, we can only choose an optimal parameter by numerical means.

Secondly, we can acquire the optimal parameters of each method in detail for Example 5.1 from Table 1. In order to ensure the credibility of the numerical experimental comparison results, in the experimental example, we choose the unified parameters for all methods. Three values are selected for parameter N, namely $2^5,\ 2^6,\ 2^7$, and $\kappa = 1,\ 10,\ 200$. The tolerance of inner iteration σ is specified as $0.1,\ 0.2,\ 0.4$. In order to show the optimal performance of each method, we select the parameters in Table 1 to obtain error estimates, outer and internal iteration steps, and CPU time. The specific experimental information of all methods for Example 5.1 are listed in Tables 2, 3, 4, 5, 6, 7, 8, 9, and 10, then we have made a detailed comparison.

Thirdly, we can obviously find that MN-PBS method outperforms all other known modified Newton in terms of the CPU time and the number of internal and external iteration steps from Tables 2, 3, 4, 5, 6, 7, 8, 9, and 10. In addition, it is obvious from Tables 2, 3, 4, 5, 6, 7, 8, 9, and 10 that when the problem scale N changes, whether it is the number of internal and external iteration steps or CPU time, our method has greater advantages.

Furthermore, we can see from Table 1 that with the change of the problem scale N in the systems, the optimal parameters of MN-PBS are very stable, which will determine the optimal parameters faster for the application of practical examples.

Example 5.2 Consider the complex nonlinear Helmholtz equation:

$$-\Delta v + \kappa_1 v + i\kappa_2 v = -e^v,$$

where κ_1, κ_2 are real coefficient functions. The independent variable v satisfies homogeneous Dirichlet boundary conditions in the square $\Omega = [0, 1] \times [0, 1]$. Similar to Example 5.1, we use the equidistant grid with step $\Delta t = h = 1/(N + 1)$ at each temporal step of the implicit scheme to discretize the nonlinear systems. Then, the following nonlinear equation format will be obtained:

$$F(v) = Dv + \phi(v) = 0, \tag{5.2}$$

Table 1 The optimal parameters of the modified Newton methods for Example 5.1

	N	$\kappa = 1$			$\kappa = 10$			$\kappa = 200$		
		$\sigma = 0.1$	$\sigma = 0.2$	$\sigma = 0.4$	$\sigma = 0.1$	$\sigma = 0.2$	$\sigma = 0.4$	$\sigma = 0.1$	$\sigma = 0.2$	$\sigma = 0.4$
MN-GSOR	2^5	0.92	0.91	0.93	0.93	0.89	0.94	0.91	0.94	0.94
	2^6	0.93	0.92	0.94	0.93	0.92	0.94	0.93	0.88	0.94
	2^7	0.93	0.92	0.94	0.93	0.92	0.94	0.93	0.92	0.94
MN-DSS	2^5	3.29	3.52	3.31	3.58	3.54	3.63	7.10	7.88	7.93
	2^6	3.93	4.20	3.46	4.63	4.41	3.83	8.11	7.40	8.13
	2^7	5.24	6.42	5.71	6.72	6.42	5.76	8.75	9.52	7.82
MN-FPAE	2^5	0.67	0.73	0.80	0.95	0.72	0.81	0.62	0.96	0.75
	2^6	0.93	0.86	0.80	0.93	0.74	0.80	0.96	0.70	0.81
	2^7	0.68	0.76	0.80	0.68	0.85	0.81	0.96	0.74	0.80
MN-AGSOR	2^5	(1.00, 0.92)	(0.86, 0.93)	(1.00, 0.88)	(1.00, 0.90)	(0.95, 0.87)	(0.98, 0.88)	(1.00, 0.89)	(1.00, 0.90)	(1.00, 0.90)
	2^6	(1.00, 0.81)	(0.97, 0.84)	(0.98, 0.89)	(1.00, 0.82)	(0.97, 0.84)	(1.00, 0.87)	(1.00, 0.88)	(1.00, 0.89)	(1.00, 0.89)
	2^7	(1.00, 0.86)	(1.00, 0.88)	(1.00, 0.90)	(1.00, 0.86)	(0.97, 0.84)	(0.98, 0.87)	(0.98, 0.88)	(1.00, 0.89)	(1.00, 0.89)
MN-DGPMHSS	2^5	(1.73, 0.42)	(1.05, 0.63)	(1.48, 0.61)	(1.55, 0.41)	(1.44, 0.83)	(2.87, 0.80)	(1.12, 0.78)	(1.29, 0.65)	(1.15, 0.78)
	2^6	(1.25, 0.83)	(1.06, 0.64)	(1.86, 0.71)	(1.14, 1.77)	(1.43, 0.83)	(1.52, 0.62)	(1.26, 0.82)	(1.48, 0.75)	(1.03, 0.70)
	2^7	(1.02, 0.69)	(1.06, 0.64)	(1.32, 0.71)	(1.11, 0.75)	(1.12, 0.68)	(1.34, 0.56)	(1.10, 0.74)	(1.25, 0.72)	(1.10, 0.69)
MN-PBS	2^5	(0.10, 3.85)	(0.10, 3.85)	(0.10, 3.85)	(0.10, 3.85)	(0.10, 3.85)	(0.10, 3.85)	(0.10, 3.85)	(0.10, 3.85)	(0.10, 3.85)
	2^6	(0.10, 3.85)	(0.10, 3.85)	(0.10, 3.85)	(0.10, 3.85)	(0.10, 3.85)	(0.10, 3.85)	(0.10, 3.85)	(0.10, 3.85)	(0.10, 3.85)
	2^7	(0.10, 3.85)	(0.10, 3.85)	(0.10, 3.85)	(0.10, 3.85)	(0.10, 3.85)	(0.10, 3.85)	(0.10, 3.85)	(0.10, 2.85)	(0.10, 3.85)

Table 2 Numerical results of the modified Newton methods for $\sigma = 0.1 and \kappa = 1$

N	Method	REST	CPUT (s)	OT Step	IT Step
$N = 2^5$	MN-DGPMHSS	7.8802×10^{-11}	0.144311	4	28
	MN-GSOR	1.5406×10^{-11}	0.062711	3	10
	MN-AGSOR	1.9644×10^{-11}	0.067146	3	9
	MN-DSS	1.0118×10^{-12}	0.134404	4	16
	MN-FPAE	5.2089×10^{-11}	0.167721	4	31
	MN-PBS	9.3189×10^{-11}	0.043350	3	6
$N = 2^6$	MN-DGPMHSS	4.7213×10^{-11}	0.959257	4	31
	MN-GSOR	5.6142×10^{-11}	0.591532	3	9
	MN-AGSOR	2.3619×10^{-12}	0.600614	3	9
	MN-DSS	8.9131×10^{-11}	0.856905	4	19
	MN-FPAE	5.1761×10^{-11}	0.788058	4	31
	MN-PBS	3.4152×10^{-13}	0.517466	3	8
$N = 2^7$	MN-DGPMHSS	2.3172×10^{-11}	17.638865	4	32
	MN-GSOR	6.0596×10^{-11}	13.830030	3	9
	MN-AGSOR	1.4726×10^{-11}	14.752944	3	7
	MN-DSS	9.6873×10^{-11}	19.130416	4	26
	MN-FPAE	2.3577×10^{-11}	17.951598	4	32
	MN-PBS	7.2341×10^{-11}	6.773656	2	6

Table 3 Numerical results of the modified Newton methods for $\sigma = 0.1$ *and* $\kappa = 10$

N	Method	REST	CPUT (s)	OT Step	IT Step
$N = 2^5$	MN-DGPMHSS	8.5773×10^{-11}	0.105498	4	29
	MN-GSOR	6.6005×10^{-11}	0.062289	3	9
	MN-AGSOR	1.7309×10^{-11}	0.064995	3	9
	MN-DSS	1.7000×10^{-11}	0.085320	4	16
	MN-FPAE	5.9690×10^{-11}	0.082215	4	31
	MN-PBS	7.7975×10^{-11}	0.044994	3	6
$N = 2^6$	MN-DGPMHSS	3.4095×10^{-12}	1.335736	5	60
	MN-GSOR	5.4899×10^{-11}	0.594764	3	9
	MN-AGSOR	7.6879×10^{-12}	0.587939	3	8
	MN-DSS	9.9539×10^{-11}	0.909900	4	20
	MN-FPAE	5.0021×10^{-11}	0.802096	4	31
	MN-PBS	2.6375×10^{-13}	0.523284	3	8
$N = 2^7$	MN-DGPMHSS	2.2921×10^{-11}	21.642412	4	32
	MN-GSOR	5.9964×10^{-11}	13.906540	3	9
	MN-AGSOR	2.1036×10^{-11}	13.617142	3	7
	MN-DSS	2.2546×10^{-11}	15.893385	4	32
	MN-FPAE	2.3432×10^{-11}	17.624799	4	32
	MN-PBS	6.5088×10^{-11}	6.841126	2	6

Table 4 Numerical results of the modified Newton methods for $\sigma = 0.1 and \kappa = 200$

N	Method	REST	CPUT (s)	OT Step	IT Step
$N = 2^5$	MN-DGPMHSS	8.8973×10^{-11}	0.112221	4	32
	MN-GSOR	2.7979×10^{-11}	0.062446	3	10
	MN-AGSOR	9.2899×10^{-11}	0.055711	3	7
	MN-DSS	3.2225×10^{-11}	0.138523	5	39
	MN-FPAE	7.2088×10^{-11}	0.081159	4	31
	MN-PBS	1.4729×10^{-13}	0.045833	3	8
$N = 2^6$	MN-DGPMHSS	9.4755×10^{-11}	1.017040	4	31
	MN-GSOR	3.9669×10^{-11}	0.577671	3	9
	MN-AGSOR	1.2859×10^{-11}	0.571199	3	8
	MN-DSS	1.0085×10^{-11}	1.216121	5	40
	MN-FPAE	5.5641×10^{-11}	0.782422	4	31
	MN-PBS	7.3279×10^{-11}	0.350362	2	6
$N = 2^7$	MN-DGPMHSS	6.3573×10^{-11}	22.386015	4	31
	MN-GSOR	4.9103×10^{-11}	15.226103	3	9
	MN-AGSOR	7.4412×10^{-11}	13.904789	3	8
	MN-DSS	1.9021×10^{-11}	19.346818	5	43
	MN-FPAE	4.8090×10^{-11}	18.421113	4	32
	MN-PBS	2.2187×10^{-11}	6.628693	2	6

Table 5 Numerical results of the modified Newton methods for $\sigma = 0.2 and \kappa = 1$

N	Method	REST	CPUT (s)	OT Step	IT Step
$N = 2^5$	MN-DGPMHSS	4.0794×10^{-11}	0.131778	5	30
	MN-GSOR	2.2060×10^{-12}	0.079172	4	11
	MN-AGSOR	6.4897×10^{-11}	0.063156	3	10
	MN-DSS	3.6615×10^{-11}	0.098494	4	15
	MN-FPAE	8.0488×10^{-11}	0.097139	5	29
	MN-PBS	9.3189×10^{-11}	0.043569	3	6
$N = 2^6$	MN-DGPMHSS	4.5258×10^{-11}	1.161346	5	30
	MN-GSOR	9.3864×10^{-12}	0.767480	4	10
	MN-AGSOR	1.5505×10^{-11}	0.583685	3	9
	MN-DSS	9.3721×10^{-12}	1.116826	5	20
	MN-FPAE	7.8075×10^{-11}	0.935007	5	29
	MN-PBS	3.4658×10^{-11}	0.532702	3	6
$N = 2^7$	MN-DGPMHSS	9.9614×10^{-11}	41.567987	5	29
	MN-GSOR	1.0731×10^{-11}	16.655726	4	10
	MN-AGSOR	1.3044×10^{-11}	19.593636	4	8
	MN-DSS	7.5336×10^{-11}	17.883278	5	29
	MN-FPAE	7.5228×10^{-11}	20.091986	5	29
	MN-PBS	1.2191×10^{-11}	9.509694	3	6

Table 6 Numerical results of the modified Newton methods for $\sigma = 0.2 and \kappa = 10$

N	Method	REST	CPUT (s)	OT Step	IT Step
$N = 2^5$	MN-DGPMHSS	9.4559×10^{-11}	0.122092	5	29
	MN-GSOR	2.8621×10^{-11}	0.080296	4	11
	MN-AGSOR	1.2679×10^{-12}	0.068469	3	11
	MN-DSS	9.6514×10^{-11}	0.086448	4	15
	MN-FPAE	8.5943×10^{-11}	0.097068	5	29
	MN-PBS	7.7975×10^{-11}	0.043121	3	6
$N = 2^6$	MN-DGPMHSS	9.6374×10^{-11}	1.135188	5	29
	MN-GSOR	9.0884×10^{-12}	0.788057	4	10
	MN-AGSOR	1.6675×10^{-11}	0.608389	3	9
	MN-DSS	4.0988×10^{-11}	1.146924	5	20
	MN-FPAE	3.6302×10^{-11}	0.999195	5	30
	MN-PBS	3.2373×10^{-11}	0.515128	3	6
$N = 2^7$	MN-DGPMHSS	4.6967×10^{-11}	28.516426	5	30
	MN-GSOR	1.0619×10^{-11}	18.373564	4	10
	MN-AGSOR	1.9277×10^{-11}	15.028722	3	9
	MN-DSS	7.4899×10^{-11}	20.337011	5	29
	MN-FPAE	3.5388×10^{-11}	21.748752	5	30
	MN-PBS	1.1915×10^{-11}	9.925258	3	6

Table 7 Numerical results of the modified Newton methods for $\sigma = 0.2 and \kappa = 200$

N	Method	REST	CPUT (s)	OT Step	IT Step
$N = 2^5$	MN-DGPMHSS	1.5948×10^{-11}	0.151061	6	35
	MN-GSOR	7.3662×10^{-13}	0.103026	5	10
	MN-AGSOR	2.8882×10^{-12}	0.072338	4	8
	MN-DSS	2.6674×10^{-11}	0.146275	6	36
	MN-FPAE	5.3981×10^{-11}	0.105886	6	29
	MN-PBS	2.3108×10^{-11}	0.043192	3	6
$N = 2^6$	MN-DGPMHSS	3.0958×10^{-11}	1.354175	6	33
	MN-GSOR	4.8395×10^{-12}	0.815533	4	11
	MN-AGSOR	2.3078×10^{-11}	0.773675	4	8
	MN-DSS	6.9309×10^{-11}	1.429176	6	40
	MN-FPAE	9.7378×10^{-11}	1.009643	5	29
	MN-PBS	1.5769×10^{-11}	0.554260	3	6
$N = 2^7$	MN-DGPMHSS	5.1662×10^{-11}	29.112601	6	31
	MN-GSOR	8.7771×10^{-12}	18.728520	4	10
	MN-AGSOR	1.0865×10^{-11}	19.713357	4	8
	MN-DSS	5.6400×10^{-11}	24.150648	6	44
	MN-FPAE	7.8501×10^{-11}	22.474799	5	29
	MN-PBS	4.3426×10^{-11}	6.970027	2	6

Table 8 Numerical results of the modified Newton methods for $\sigma = 0.4 and \kappa = 1$

N	Method	REST	CPUT (s)	OT Step	IT Step
$N = 2^5$	MN-DGPMHSS	5.6979×10^{-11}	0.151924	7	28
	MN-GSOR	3.3238×10^{-12}	0.108211	5	10
	MN-AGSOR	4.7213×10^{-11}	0.072808	4	8
	MN-DSS	4.5418×10^{-11}	0.123774	7	14
	MN-FPAE	4.1760×10^{-12}	0.134586	8	32
	MN-PBS	9.3189×10^{-11}	0.044243	3	6
$N = 2^6$	MN-DGPMHSS	6.2467×10^{-11}	1.636712	8	29
	MN-GSOR	8.6233×10^{-13}	0.923287	5	10
	MN-AGSOR	2.5821×10^{-11}	0.704503	4	8
	MN-DSS	8.6539×10^{-11}	1.795502	8	22
	MN-FPAE	5.0722×10^{-12}	1.536892	8	32
	MN-PBS	3.4658×10^{-11}	0.509995	3	6
$N = 2^7$	MN-DGPMHSS	7.3334×10^{-12}	29.987529	8	32
	MN-GSOR	1.1228×10^{-12}	21.538351	5	10
	MN-AGSOR	2.5437×10^{-11}	15.747004	4	8
	MN-DSS	8.8939×10^{-12}	23.271966	7	28
	MN-FPAE	5.6848×10^{-12}	32.896993	8	32
	MN-PBS	1.2191×10^{-11}	10.136770	3	6

Table 9 Numerical results of the modified Newton methods for $\sigma = 0.4 and \kappa = 10$

N	Method	REST	CPUT (s)	OT Step	IT Step
$N = 2^5$	MN-DGPMHSS	7.4195×10^{-11}	0.158589	8	30
	MN-GSOR	8.2029×10^{-13}	0.105132	5	10
	MN-AGSOR	3.8408×10^{-11}	0.072292	4	8
	MN-DSS	2.5516×10^{-11}	0.142697	8	16
	MN-FPAE	9.9974×10^{-11}	0.117341	7	28
	MN-PBS	7.7975×10^{-11}	0.044298	3	6
$N = 2^6$	MN-DGPMHSS	6.9926×10^{-11}	1.600569	8	29
	MN-GSOR	8.1753×10^{-13}	0.950699	5	10
	MN-AGSOR	4.6645×10^{-11}	0.755063	4	8
	MN-DSS	5.0193×10^{-11}	1.792095	8	23
	MN-FPAE	4.9230×10^{-12}	1.598624	8	32
	MN-PBS	3.2373×10^{-11}	0.519117	3	6
$N = 2^7$	MN-DGPMHSS	9.7811×10^{-12}	33.037516	8	29
	MN-GSOR	1.0979×10^{-12}	22.307950	5	10
	MN-AGSOR	3.7478×10^{-11}	16.182322	4	8
	MN-DSS	1.4996×10^{-11}	24.780850	7	28
	MN-FPAE	5.6745×10^{-12}	31.433724	8	32
	MN-PBS	1.1915×10^{-11}	9.836060	3	6

Table 10 Numerical results of the modified Newton methods for $\sigma = 0.4 and \kappa = 200$

N	Method	REST	CPUT (s)	OT Step	IT Step
$N = 2^5$	MN-DGPMHSS	8.9873×10^{-11}	0.157556	8	32
	MN-GSOR	7.3662×10^{-11}	0.096019	5	10
	MN-AGSOR	2.8882×10^{-11}	0.074849	4	8
	MN-DSS	2.6285×10^{-11}	0.179908	9	36
	MN-FPAE	7.8463×10^{-11}	0.116932	7	27
	MN-PBS	2.3108×10^{-11}	0.043869	3	6
$N = 2^6$	MN-DGPMHSS	5.7600×10^{-11}	1.762524	8	32
	MN-GSOR	1.0177×10^{-12}	1.005718	5	10
	MN-AGSOR	2.3078×10^{-11}	0.785183	4	8
	MN-DSS	1.0988×10^{-11}	2.317879	10	40
	MN-FPAE	7.6846×10^{-11}	1.343046	7	28
	MN-PBS	1.5769×10^{-11}	0.523234	3	6
$N = 2^7$	MN-DGPMHSS	2.7780×10^{-11}	37.109959	8	32
	MN-GSOR	7.7360×10^{-13}	19.822454	5	10
	MN-AGSOR	1.0865×10^{-11}	21.968407	4	8
	MN-DSS	7.1942×10^{-11}	41.739029	10	45
	MN-FPAE	4.7624×10^{-12}	29.418761	8	32
	MN-PBS	8.2316×10^{-12}	9.508568	3	6

with

$$D = (B_N \otimes I_N + I_N \otimes B_N + \kappa_1 I_n) + i\kappa_2 I_n,$$

$$\phi(v) = (e^{v_1}, e^{v_2}, ..., e^{v_n})^T \text{ for the vector } v = (v_1, v_2, ..., v_n)^T,$$

where $B_N = \frac{1}{h^2} tridiag(-1, 2, -1) \in R^{N \times N}$. The symbol $\otimes$ denotes Kronecker product and $n = N \times N$.

Firstly, the original conjecture point is selected as $v_0 = 0$, and we set the parameters $\kappa_1 = 100$, $\kappa_2 = -1$. The stop tolerances of inner iteration σ is equal to 0.1, 0.2, and 0.4.

In Table 11, we put the optimal parameters of the experimental numerical results of all methods for Example 5.2. In order to show the optimal performance of all methods, we select the optimal parameters in Table 11 to compare the experimental numerical results. For $\sigma = 0.1$ of Example 5.2, the specific performances are listed in Table 12. For $\sigma = 0.2$, all experimental data are placed in Table 13. Similarly, in Table 14, we list the numerical results for $\sigma = 0.4$.

From Tables 12, 13, and 14, we can find that our method still has certain advantages as Example 5.1 through comparison. It is easy for us to find out that the optimal parameters ζ, α of MN-PBS for Example 5.2 are as stable as Example 5.1, when the problem size N varies, which shows that it is convenient and easy to select the optimal

Table 11 The optimal parameters of the modified Newton methods for Example 5.2

Method	$N = 100$			$N = 150$		
	$\sigma = 0.1$	$\sigma = 0.2$	$\sigma = 0.4$	$\sigma = 0.1$	$\sigma = 0.2$	$\sigma = 0.4$
MN-GSOR	0.99	0.99	0.99	0.99	0.99	0.99
MN-DSS	-	-	-	-	-	-
MN-FPAE	0.99	1.01	0.98	0.98	0.99	0.98
MN-DGPMHSS	(0.93, 1.02)	(0.96, 1.04)	(0.96, 1.02)	(1.01, 1.08)	(0.91, 0.96)	(0.91, 0.94)
MN-AGSOR	(0.98, 0.97)	(1.01, 0.98)	(0.98, 1.04)	(0.99, 0.98)	(1.01, 0.98)	(0.99, 1.01)
MN-PBS	(0.08, 3.24)	(0.08, 3.24)	(0.08, 3.24)	(0.08, 3.24)	(0.08, 3.24)	(0.08, 3.24)

parameters in the experiment. Moreover, with the change of problem scale N, the numbers of internal and external iteration steps of MN-PBS maintain 2 and 4, which shows that our method is extraordinarily stable.

Example 5.3 Solve the complex two-dimensional nonlinear convection-diffusion equation as follows.

$$\begin{cases} -(\alpha_1 + i\beta_1)(v_{xx} + v_{yy}) + \kappa v = -(\alpha_2 + i\beta_2)ve^v, & for\ (x, y) \text{ in } \Omega, \\ v(x, y) = 0, & for\ (x, y) \text{ on } \partial\Omega. \end{cases}$$

The symbol definition here is $\Omega = (0, 1) \times (0, 1)$, and its boundary is represented by $\partial\Omega$, and κ is a positive constant which controls the size of the reaction term. Here, we take $\alpha_1 = 1$, $\beta_1 = 1$, $\alpha_2 = \beta_2 = 0.5$. Similarly, we use the equidistant grid with step $\Delta t = h = 1/(N + 1)$ at each temporal step of the implicit scheme to discretize the

Table 12 Numerical results of the modified Newton methods for $\sigma = 0.1$

N	Method	REST	CPUT (s)	OT Step	IT Step
$N = 100$	MN-DSS	-	-	-	-
	MN-DGPMHSS	9.2059×10^{-13}	5.784262	5	40
	MN-GSOR	9.0744×10^{-13}	2.969484	3	6
	MN-AGSOR	5.8335×10^{-11}	2.976541	3	6
	MN-FPAE	3.9544×10^{-12}	2.547493	3	6
	MN-PBS	8.7414×10^{-11}	1.763517	2	4
$N = 150$	MN-DSS	-	-	-	-
	MN-DGPMHSS	9.2864×10^{-13}	99.631230	5	40
	MN-GSOR	9.1015×10^{-13}	51.032516	3	6
	MN-AGSOR	1.0210×10^{-12}	58.106648	3	6
	MN-FPAE	9.2391×10^{-11}	63.409822	3	6
	MN-PBS	8.7131×10^{-11}	35.553395	2	4

Table 13 Numerical results of the modified Newton methods for $\sigma = 0.2$

N	Method	REST	CPUT (s)	OT Step	IT Step
$N = 100$	MN-DSS	-	-	-	-
	MN-DGPMHSS	1.4569×10^{-11}	6.899301	6	36
	MN-GSOR	9.0744×10^{-13}	2.993067	3	6
	MN-AGSOR	9.8120×10^{-13}	2.879692	3	6
	MN-FPAE	4.0962×10^{-12}	2.743238	3	6
	MN-PBS	8.7414×10^{-11}	1.746299	2	4
$N = 150$	MN-DSS	-	-	-	-
	MN-DGPMHSS	1.5130×10^{-11}	141.613043	6	36
	MN-GSOR	9.1015×10^{-13}	50.139118	3	6
	MN-AGSOR	9.8202×10^{-13}	56.093875	3	6
	MN-FPAE	3.9425×10^{-12}	60.746859	3	6
	MN-PBS	8.7131×10^{-11}	35.145183	2	4

nonlinear systems. Then, the following nonlinear equation format will be obtained:

$$F(v) = Dv + (\alpha_2 + i\beta_2)h\Delta t\phi(v) = 0, \tag{5.3}$$

with

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

$$\phi(v) = (v_1 e^{v_1}, v_2 e^{v_2}, ..., v_n e^{v_n})^T \text{ for the vector } v = (v_1, v_2, ..., v_n)^T,$$

Table 14 Numerical results of the modified Newton methods for $\sigma = 0.4$

N	Method	REST	CPUT (s)	OT Step	IT Step
$N = 100$	MN-DSS	-	-	-	-
	MN-DGPMHSS	1.4539×10^{-11}	9.770582	9	36
	MN-GSOR	9.0744×10^{-13}	2.749309	3	6
	MN-AGSOR	6.8832×10^{-11}	2.925869	3	6
	MN-FPAE	9.2546×10^{-11}	2.844930	3	6
	MN-PBS	8.7414×10^{-11}	1.761293	2	4
$N = 150$	MN-DSS	-	-	-	-
	MN-DGPMHSS	1.5330×10^{-11}	146.402580	9	36
	MN-GSOR	9.1015×10^{-13}	57.893345	3	6
	MN-AGSOR	9.9434×10^{-13}	56.217798	3	6
	MN-FPAE	9.2391×10^{-11}	52.416926	3	6
	MN-PBS	8.7131×10^{-11}	35.298532	2	4

the definitions of symbols A_N, N and $\otimes$ are consistent with Example 5.1. In addition, $n = N \times N$.

In Table 15, we put the optimal parameters of the experimental numerical results of all methods for Example 5.3. In this example, we choose the value of N as 80 and 120. The parameter κ controlling the size of the reaction item is selected as 10 and 200. The stop tolerances of inner iteration σ are equal to 0.1, 0.2, and 0.4. When α and ζ are taken as the optimal parameters, the specific performances of all methods for Example 5.3 are listed in Tables 16, 17, 18, 19, 20, and 21.

From Tables 16, 17, 18, 19, 20, and 21, the comparison results of all experiments show that MN-PBS has advantages over other methods. The conclusion is consistent with the previous two examples. And we can find that the optimal parameters α, ζ of MN-PBS for Example 5.3 are as stable as Example 5.1 and Example 5.2. Moreover, as the matrix scale becomes larger, our advantages become more obvious. In addition, with the change of problem scale N, both internal and external iteration steps and CPU time are always stable.

From the numerical experimental results, we can confirm that when the parameters ζ, α are numerically optimal, even when the problem scale N increases to 150, the numbers of internal and external iteration steps of MN-PBS method still retain a small value. However, our method still has some shortcomings, which will be the direction we need to explore in the future.

Future conjectures are as follows:

- Firstly, we only derive the numerical optimal parameters of MN-PBS method through numerical experiments, but we have not found the best theoretical parameters. Actually, some known literatures have presented some detailed and reasonable discussions about practical formulas for containing an optimal parameter. However, it is not easy to discuss their optimal values for two parameters, which will be one of the directions that can be studied in the future.
- Secondly, the selection of parameter ζ in the preconditioning matrix $\mathcal{Z}$ has been inspired by some known literature. How to select the optimal parameter ζ that can help to improve the efficiency of the method is still the direction we need to study.
- Thirdly, we only give the local convergence analysis of MN-PBS, and its semi-local convergence is also worth considering and studying, which will enable us to better obtain the numerical solution through the selection of original points.
- Fourthly, we have consulted a large number of relevant literature on how to select the tolerance of internal iteration, and there is no theoretical proof or analysis. Therefore, we still use the numerical selection in the existing literature. How to select the inner iterative tolerance theoretically is worthy of our in-depth study.
- Finally, deep learning has aroused great interest among scholars, the numerical methods of partial differential equations are gradually combined with deep learning recently, which provides ideas for the future direction of integrating MN-PBS method with the current popular direction.

Table 15 The optimal parameters of the modified Newton methods for Example 5.3

	N	$\kappa = 10$			$\kappa = 200$		
		$\sigma = 0.1$	$\sigma = 0.2$	$\sigma = 0.4$	$\sigma = 0.1$	$\sigma = 0.2$	$\sigma = 0.4$
MN-GSOR	80	0.93	0.94	0.92	0.93	0.89	0.94
	120	0.93	0.94	0.92	0.93	0.92	0.94
MN-DSS	80	2.12	2.24	2.23	6.80	6.66	6.80
	120	2.10	2.24	2.23	6.73	6.56	6.70
MN-FPAE	80	0.68	0.87	0.81	0.94	0.72	0.80
	120	0.68	0.82	0.81	0.93	0.86	0.80
MN-AGSOR	80	(0.92, 0.93)	(0.97, 0.84)	(0.98, 0.87)	(0.94, 0.92)	(0.97, 0.91)	(0.97, 0.91)
	120	(0.97, 0.89)	(0.86, 0.99)	(0.98, 0.87)	(0.92, 0.94)	(0.98, 0.89)	(0.98, 0.89)
MN-DGPMHSS	80	(3.11, 0.35)	(3.06, 1.15)	(2.48, 0.81)	(1.41, 0.82)	(1.67, 0.77)	(1.34, 0.64)
	120	(2.13,0.24)	(1.18, 0.33)	(1.27, 0.54)	(1.27, 0.84)	(1.14, 0.72)	(1.11, 0.67)
MN-PBS	80	(0.15, 4.70)	(0.15, 4.70)	(0.15, 4.70)	(0.15, 4.70)	(0.15, 4.70)	(0.15, 4.70)
	120	(0.15, 4.70)	(0.15, 4.70)	(0.15, 4.70)	(0.15, 4.70)	(0.15, 4.70)	(0.15, 4.70)

Table 16 Numerical results of the modified Newton methods for $\sigma = 0.1 and \kappa = 10$

N	Method	REST	CPUT (s)	OT Step	IT Step
$N = 80$	MN-DGPMHSS	4.3189×10^{-11}	2.060768	4	22
	MN-GSOR	7.6023×10^{-11}	1.568963	3	9
	MN-AGSOR	6.9903×10^{-12}	1.515759	3	9
	MN-DSS	3.4597×10^{-11}	1.503651	3	8
	MN-FPAE	2.4967×10^{-11}	1.995117	4	32
	MN-PBS	3.8317×10^{-12}	1.322312	3	6
$N = 120$	MN-DGPMHSS	5.6750×10^{-11}	20.251476	6	24
	MN-GSOR	7.1992×10^{-11}	12.102357	3	9
	MN-AGSOR	6.2218×10^{-11}	11.805891	3	8
	MN-DSS	2.3828×10^{-11}	10.832867	3	8
	MN-FPAE	2.5515×10^{-11}	12.284388	4	32
	MN-PBS	2.1059×10^{-12}	7.794302	3	6

6 Conclusions

In the paper, by using PBS as internal iteration to solve Newton equations, we obtain the solution of complex nonlinear system. Consequently, we proposed the modified Newton parameterized block splitting (MN-PBS) iteration method, and we confirmed it is an efficient iterative method for solving complex nonlinear systems with symmetric Jacobian matrices. For the new presented method, we give the local convergence analysis and proof under appropriate conditions. In Sect. 5, we applied our new method to practical numerical examples. Compared with MN-DGPMHSS, MN-

Table 17 Numerical results of the modified Newton methods for $\sigma = 0.1 and \kappa = 200$

N	Method	REST	CPUT (s)	OT Step	IT Step
$N = 80$	MN-DGPMHSS	4.0166×10^{-12}	3.147384	5	35
	MN-GSOR	6.0383×10^{-11}	1.510917	3	9
	MN-AGSOR	5.0248×10^{-11}	1.412265	3	9
	MN-DSS	6.7107×10^{-11}	2.253726	4	31
	MN-FPAE	5.2018×10^{-11}	1.946262	4	31
	MN-PBS	7.9777×10^{-11}	1.351768	3	6
$N = 120$	MN-DGPMHSS	5.8761×10^{-11}	14.790215	4	31
	MN-GSOR	5.9465×10^{-11}	18.430753	3	9
	MN-AGSOR	7.1302×10^{-11}	16.300547	3	9
	MN-DSS	5.5783×10^{-11}	16.998169	4	31
	MN-FPAE	2.4925×10^{-11}	38.375541	4	32
	MN-PBS	4.4575×10^{-11}	7.968622	3	6

Table 18 Numerical results of the modified Newton methods for $\sigma = 0.2 and \kappa = 10$

N	Method	REST	CPUT (s)	OT Step	IT Step
$N = 80$	MN-DGPMHSS	5.0853×10^{-11}	2.733555	5	30
	MN-GSOR	3.4588×10^{-11}	1.999012	4	9
	MN-AGSOR	1.5444×10^{-12}	1.400165	3	9
	MN-DSS	1.5874×10^{-11}	1.907091	4	8
	MN-FPAE	4.9360×10^{-11}	2.363457	5	30
	MN-PBS	3.8317×10^{-12}	1.329859	3	6
$N = 120$	MN-DGPMHSS	7.6754×10^{-12}	20.939155	7	28
	MN-GSOR	3.5452×10^{-11}	13.149958	4	9
	MN-AGSOR	3.4025×10^{-11}	11.220533	4	8
	MN-DSS	9.0137×10^{-12}	13.386575	4	8
	MN-FPAE	3.3345×10^{-11}	16.308287	5	30
	MN-PBS	2.1059×10^{-12}	7.936163	3	6

GSOR, MN-AGSOR, MN-DSS, and MN-FPAE methods, the experimental results show that MN-PBS method is not only efficient but also performs well in both the iteration steps and CPU time. Moreover, when the problem size becomes larger, MN-PBS method can still maintain a small number of internal and external iteration steps and a small amount of CPU time, and the optimal parameters are almost constant from the results.

In addition, there are still some problems worth considering. For example, although the corresponding spectral radius of the iteration matrix $Q_{\alpha,\zeta}(v)$ has been determined, but we still have not determined the theoretical optimal parameters ζ and α, only the

Table 19 Numerical results of the modified Newton methods for $\sigma = 0.2 and \kappa = 200$

N	Method	REST	CPUT (s)	OT Step	IT Step
$N = 80$	MN-DGPMHSS	8.1037×10^{-11}	3.042515	6	31
	MN-GSOR	3.0277×10^{-11}	1.865526	4	11
	MN-AGSOR	8.4382×10^{-11}	1.899857	4	8
	MN-DSS	9.9578×10^{-11}	2.832149	5	30
	MN-FPAE	9.1227×10^{-11}	2.339996	5	29
	MN-PBS	7.9777×10^{-11}	1.302343	3	6
$N = 120$	MN-DGPMHSS	v8.6514×10^{-11}	17.461697	5	30
	MN-GSOR	9.5830×10^{-12}	13.985222	4	10
	MN-AGSOR	7.4335×10^{-11}	12.585885	4	8
	MN-DSS	7.8763×10^{-11}	16.238791	5	30
	MN-FPAE	7.9620×10^{-11}	26.955257	5	29
	MN-PBS	4.4575×10^{-11}	7.694848	3	6

Table 20 Numerical results of the modified Newton methods for $\sigma = 0.4 and \kappa = 10$

N	Method	REST	CPUT (s)	OT Step	IT Step
$N = 80$	MN-DGPMHSS	8.2172×10^{-12}	3.450756	7	28
	MN-GSOR	1.2105×10^{-11}	1.897579	4	10
	MN-AGSOR	2.6410×10^{-12}	1.847811	4	8
	MN-DSS	1.5921×10^{-11}	1.888319	4	8
	MN-FPAE	6.3784×10^{-12}	3.855849	8	32
	MN-PBS	3.8317×10^{-12}	1.319788	3	6
$N = 120$	MN-DGPMHSS	7.6754×10^{-12}	20.939155	7	28
	MN-GSOR	1.2453×10^{-11}	13.203421	4	10
	MN-AGSOR	2.4181×10^{-11}	13.505979	4	8
	MN-DSS	8.8945×10^{-12}	11.565212	4	8
	MN-FPAE	6.5040×10^{-12}	24.395661	8	32
	MN-PBS	2.1059×10^{-12}	7.902073	3	6

optimal parameter values of each example are determined by numerical techniques. Finding the optimal theoretical parameters can be the direction of our future research. Besides, the choice of preconditioner is uncertain. Even if we choose matrix $\mathcal{Z}$, according to the current research literature, how to select the appropriate $\mathcal{Z}$ is still worthy of our in-depth thinking and exploration. Furthermore, our new method may consider combining it with the current mainstream direction, so as to expand the applicable scope of our method.

Table 21 Numerical results of the modified Newton methods for $\sigma = 0.4 and \kappa = 200$

N	Method	REST	CPUT (s)	OT Step	IT Step
$N = 80$	MN-DGPMHSS	3.6138×10^{-11}	3.844816	8	32
	MN-GSOR	7.3407×10^{-13}	2.442852	5	10
	MN-AGSOR	8.4382×10^{-11}	2.052114	4	8
	MN-DSS	3.6797×10^{-11}	4.311352	8	32
	MN-FPAE	4.1355×10^{-12}	3.504335	8	32
	MN-PBS	7.9777×10^{-11}	1.328748	3	6
$N = 120$	MN-DGPMHSS	1.8434×10^{-11}	25.780741	8	32
	MN-GSOR	8.9294×10^{-13}	18.215608	5	10
	MN-AGSOR	7.4335×10^{-11}	13.650219	4	8
	MN-DSS	2.9030×10^{-11}	22.396249	8	32
	MN-FPAE	5.1889×10^{-12}	24.448094	8	32
	MN-PBS	4.4575×10^{-11}	7.620980	3	6

Funding This work is supported by the National Natural Science Foundation of China (Grant no. 12271479).

Data availability The data used to support the findings of this study are available from the corresponding author upon request.

Declarations

Conflict of interest The authors declare no competing interests.

References

1. Karlsson, H.-O.: The quasi-minimal residual algorithm applied to complex symmetric linear systems in quantum reactive scattering. J. Chem. Phys. **103**(12), 4914–4919 (1995)
2. Sulem, C., Sulem, P.-L.: The nonlinear Schrödinger equation self-focusing and wave collapse. Springer, New York (1999)
3. Kuramoto, Y.: Oscillations chemical waves and turbulence. Dover, Mineola (2003)
4. Papp, D., Vizvari, B.: Effective solution of linear Diophantine equation systems with an application in chemistry. J. Math. Chem. **39**(1), 15–31 (2006)
5. Zhang, Y., Sun, Q.: Preconditioned bi-conjugate gradient method of large-scale sparse complex linear equation group. Chin. J. Electron. **20**(1), 192–194 (2011)
6. Ortega, J.-M., Rheinboldt, W.-C.: Iterative solution of nonlinear equations in several variables. Academic Press, New York, NY, USA (1970)
7. Lam, B.: On the convergence of a quasi-Newton method for sparse nonlinear systems. Math. Comput. **32**(142), 447 (1978)
8. Dembo, R., Eisenstat, S., Steihaug, T.: Inexact Newton method. SIAM J. Numer. Anal. **19**(2), 400–408 (1982)
9. Peter, D.: Global inexact Newton methods for very large scale nonlinear problems. IMPACT Comput. Sci. Eng. **3**(4), 366–393 (1991)
10. Saad, Y.: Iterative methods for sparse linear systems, 2nd edn. SIAM, Philadelphia (2003)
11. An, H.-B., Bai, Z.-Z.: A globally convergent Newton-GMRES method for large sparse systems of nonlinear equations. Appl. Numer. Math. **57**(3), 235–252 (2007)
12. An, H.-B., Mo, Z.-Y., Liu, X.-P.: A choice of forcing terms in inexact Newton method. J. Comput. Appl. Math. **200**(1), 47–60 (2007)
13. Knoll, D.A., Keyes, D.E.: Jacobian-free Newton-Krylov methods: a survey of approaches and applications. J. Comput. Phys. **193**(2), 357–397 (2004)
14. Darvishi, M.T., Barati, A.: A third-order Newton-type method to solve systems of nonlinear equations. Appl. Math. Comput. **187**(2), 630–635 (2007)
15. Chen, M.-H., Wu, Q.-B.: Convergence analysis of modified Newton-HSS method for solving systems of nonlinear equations. Numer. Algor. **64**(4), 659–685 (2013)
16. Zhong, H.-X., Chen, G.-L., Guo, X.-P.: On preconditioned modified Newton-MHSS method for systems of nonlinear equations with complex symmetric Jacobian matrices. Numer. Algor. **69**(3), 553–567 (2015)
17. Wang, J., Guo, X.-P., Zhong, H.-X.: MN-DPMHSS iteration method for systems of nonlinear equations with block two-by-two complex Jacobian matrices. Numer. Algor. **77**(1), 167–184 (2018)
18. Zhou, H.-Y., Wu, S.-L., Li, C.-X.: Newton-based matrix splitting method for generalized absolute value equation. J. Comput. Appl. Math. **394**, 1–15 (2021)
19. Bai, Z.-Z., Golub, G.H., Ng, M.K.: Hermitian and skew-Hermitian splitting methods for non-Hermitian positive definite linear systems. SIAM J. Matrix Anal. Appl. **24**(3), 603–626 (2003)
20. Bai, Z.-Z., Golub, G.H., Pan, J.-Y.: Preconditioned Hermitian and skew-Hermitian splitting methods for non-Hermitian positive semidefinite linear systems. Numer. Math. **98**(1), 1–32 (2004)
21. Bai, Z.-Z., Golub, G.H., Lu, L.-Z., Yin, J.-F.: Block triangular and skew-Hermitian splitting methods for positive-definite linear systems. SIAM J. Sci. Comput. **26**(3), 844–863 (2005)
22. Bai, Z.-Z., Golub, G.H., Ng, M.K.: On successive-overrelaxation acceleration of the Hermitian and skew-Hermitian splitting iterations. Numer. Linear Algebra Appl. **14**(4), 319–335 (2007)

23. Bai, Z.-Z., Golub, G.H.: Accelerated Hermitian and skew-Hermitian splitting iteration methods for saddle-point problems. IMA J. Numer. Anal. **27**(1), 1–23 (2007)
24. Bai, Z.-Z., Benzi, M., Chen, F., Wang, Z.-Q.: Preconditioned MHSS iteration methods for a class of block two-by-two linear systems with applications to distributed control problems. IMA J. Numer. Anal. **33**(1), 343–369 (2013)
25. Zhu, M.-Z., Zhang, G.-F.: A class of iteration methods based on the HSS for Toeplitz systems of weakly nonlinear equations. J. Comput. Appl. Math. **290**, 433–444 (2015)
26. Bai, Z.-Z., Benzi, M.: Regularized HSS iteration methods for saddle-point linear systems. BIT Numer. Math. **57**(2), 287–311 (2017)
27. Xiao, X.-Y., Wang, X., Yin, H.-W.: Efficient single-step preconditioned HSS iteration methods for complex symmetric linear systems. Comput. Math. Appl. **74**(10), 2269–2280 (2017)
28. Bai, Z.-Z., Benzi, M., Chen, F.: Modified HSS iteration methods for a class of complex symmetric linear systems. Computing **87**(3–4), 93–111 (2010)
29. Bai, Z.-Z., Benzi, M., Chen, F.: On preconditioned MHSS iteration methods for complex symmetric linear systems. Numer. Algor. **56**(6), 297–317 (2011)
30. Bai, Z.-Z.: On preconditioned iteration methods for complex linear systems. J. Eng. Math. **93**(1), 41–60 (2015)
31. Xu, W.-W.: A generalization of preconditioned MHSS iteration method for complex symmetric indefinite linear system. Appl. Math. Comput. **219**(21), 10510–10517 (2013)
32. Li, C.-X., Wu, S.-L.: A double-parameter GPMHSS method for a class of complex symmetric linear systems from Helmholtz equation. Math. Probl. Eng. **2014** (2014)
33. Li, X., Yang, A.-L., Wu, Y.-J.: Lopsided PMHSS iteration method for a class of complex symmetric linear systems. Numer. Algor. **66**, 555–568 (2014)
34. Pour, H.-N., Goughery, H.-S.: New Hermitian and Skew-Hermitian splitting methods for non-Hermitian positive-definite linear systems. Numer. Algor. **69**, 207–225 (2015)
35. Xiao, X.-Y., Yin, H.-W.: Efficient parameterized HSS iteration methods for complex symmetric linear systems. Comput. Math. Appl. **73**, 87–95 (2017)
36. Xiao, X.-Y., Wang, X., Yin, H.-W.: Efficient preconditioned NHSS iteration methods for solving complex symmetric linear systems. Comput. Math. Appl. **75**, 235–247 (2018)
37. Hezari, D., Salkuyeh, D.-K., Edalatpour, V.: A new iterative method for solving a class of complex symmetric system of linear equations. Numer. Algor. **73**, 927–955 (2016)
38. Zheng, Z., Huang, F.-L., Peng, Y.-C.: Double-step scale splitting iteration method for a class of complex symmetric linear systems. Appl. Math. Lett. **73**, 91–97 (2017)
39. Salkuyeh, D.-K., Siahkolaei, T.-S.: Two-parameter TSCSP method for solving complex symmetric system of linear equations. Calcolo, **55**(8) (2018)
40. Wang, T., Zheng, Q.-Q., Lu, L.-Z.: A new iteration method for a class of complex symmetric linear systems. J. Comput. Appl. Math. **325**, 188–197 (2017)
41. Wu, Q.-B., Chen, M.-H.: Convergence analysis of modified Newton-HSS method for solving systems of nonlinear equations. Numer. Algor. **64**(4), 659–683 (2013)
42. Chen, M.-H., Wu, Q.-B.: On modified Newton-DGPMHSS method for solving nonlinear systems with complex symmetric Jacobian matrices. Comput. Math. Appl. **76**(1), 45–57 (2018)
43. Xie, F., Wu, Q.-B., Dai, P.-F.: Modified Newton-SHSS method for a class systems of nonlinear equations. Comput. Appl. Math. **38**(1), 19–43 (2019)
44. Chen, M.-H., Wu, Q.-B., Qin, G., Lin, R.-F.: An Efficient iterative approach to large sparse nonlinear systems with non-Hermitian Jacobian matrices. East Asian J. Appl. Math. **11**(2), 349–368 (2021)
45. Xie, F., Lin, R.-F., Wu, Q.-B.: Modified Newton-DSS method for solving a class of systems of nonlinear equations with complex symmetric Jacobian matrices. Numer. Algor. **85**(3), 951–975 (2020)
46. Zhang, L., Wu, Q.-B., Chen, M.-H., Lin, R.-F.: Two new effective iteration methods for nonlinear systems with complex symmetric Jacobian matrices. Comput. Appl. Math. **40**(3), 1–27 (2021)
47. Huang, B.-H., Li, W.: A modified SOR-like method for absolute value equations associated with second order cones. J. Comput. Appl. Math. **400**, 1–20 (2022)
48. Qi, X., Wu, H.-T., Xiao, X.-Y.: Modified Newton-GSOR method for solving complex nonlinear systems with symmetric Jacobian matrices. Comp. Appl. Math. **39**(3), 165 (2020)
49. Qi, X., Wu, H.-T., Xiao, X.-Y.: Modified Newton-AGSOR method for solving nonlinear systems with block two-by-two complex symmetric Jacobian matrices. Calcolo **57**(2), 14 (2020)
50. Feng, Y.-Y., Wu, Q.-B.: MN-PGSOR method for solving nonlinear systems with block two-by-two complex symmetric Jacobian matrices. J. Math. **2021** (2021)

51. Benzi, M., Golub, G.-H., Liesen, J.: Numerical solution of saddle point problems. Acta Numer. **14**, 1–137 (2005)
52. Zheng, Q.-Q., Lu, L.-Z.: A shift-splitting preconditioner for a class of block two-by-two linear systems. Appl. Math. Lett. **66**, 54–60 (2017)
53. Li, C.-L., Ma, C.-F.: Efficient parameterized rotated shift-splitting preconditioner for a class of complex symmetric linear systems. Numer. Algor. **80**, 337–354 (2019)
54. Zhang, J.-H., Wang, Z.-W., Zhao, J.: Preconditioned symmetric block triangular splitting iteration method for a class of complex symmetric linear systems. Appl. Math. Lett. **86**, 95–102 (2018)
55. Huang, Z.-G.: Efficient block splitting iteration methods for solving a class of complex symmetric linear systems. J. Comput. Appl. Math. **395** (2021)
56. Axelsson, O., Salkuyeh, D.-K.: A new version of a preconditioning method for certain two-by-two block matrices with square blocks. BIT **59**(2), 321–342 (2019)
57. Chen, M.-H., Lin, R.-F., Wu, Q.-B.: Convergence analysis of the modified Newton-HSS method under the Hölder continuous condition. J. Comput. Appl. Math. **264**, 115–130 (2014)