Communications on Applied Mathematics and Computation
https://doi.org/10.1007/s42967-024-00410-0

ORIGINAL PAPER

Modified Newton-PAGSOR Method for Solving Nonlinear Systems with Complex Symmetric Jacobian Matrices

Rong Ma[1] · **Yu-Jiang Wu**[1,2] · **Lun-Ji Song**[1,2]

Received: 27 October 2023 / Revised: 27 February 2024 / Accepted: 5 April 2024

Abstract

We propose, in this paper, the preconditioned accelerated generalized successive overrelaxation (PAGSOR) iteration method for efficiently solving the large complex symmetric linear systems. To solve the nonlinear systems whose Jacobian matrices are complex and symmetric, treating the PAGSOR method as internal iteration, we construct a modified Newton-PAGSOR (MN-PAGSOR) method to provide an effective approach for solving a wide range of problems in various scientific and engineering fields. Based on the Hölder continuous condition we present the theoretical framework of the modified method, demonstrate its local convergence properties, and provide numerical experiments to validate its effectiveness in solving a class of nonlinear systems.

Keywords Preconditioned accelerated generalized successive overrelaxation (PAGSOR) · Complex symmetric Jacobian matrix · Large sparse nonlinear systems · Modified Newton-PAGSOR (MN-PAGSOR) method · Local convergence

Mathematics Subject Classification 65H10 · 65F10 · 65F50

1 Introduction

Let us consider the solution of a complex nonlinear systems of equations,

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

where $F: \mathbb{D} \subset \mathbb{C}^n \rightarrow \mathbb{C}^n$ is a continuously differentiable nonlinear mapping defined on an open convex subset of the complex linear space $\mathbb{C}^n$, and $F = (F_1, \cdots, F_n)^{\mathrm{T}}$, $F_j = F_j(u)$, $j = 1, \cdots, n$, $u = (u_1, \cdots, u_n)^{\mathrm{T}}$. And the Jacobian matrix $F'(x)$ is large sized, sparse, and complex symmetric. This class of nonlinear systems of equations has widespread applications in various fields, such as quantum mechanics [31], fluid mechanics [24, 25, 41], structural mechanics [1, 27], chemical reaction [32, 38], nonlinear wave [47], mechanical engineering [27], and so on. Assuming that the complex symmetric Jacobian matrix $F'(u)$

✉ Yu-Jiang Wu
myjaw@lzu.edu.cn

[1] School of Mathematics and Statistics, Lanzhou University, Lanzhou 730000, Gansu, China

[2] Key Laboratory of Applied Mathematics and Complex Systems, Lanzhou University, Lanzhou 730000, Gansu, China

Published online: 07 August 2024

has the following form:

$$F'(u) = W(u) + \mathrm{i}T(u), \tag{2}$$

where $W(u)$ and $T(u)$ are real symmetric and positive semi-definite matrices, and $\mathrm{i} = \sqrt{-1}$ is the imaginary unit. The effective classical method to solve the nonlinear system of equations (1) is the Newton method [22, 36, 44] with the specific format:

$$u_{k+1} = u_k + s_k, \quad F'(u_k)s_k = -F(u_k), \quad k = 0, 1, \cdots. \tag{3}$$

With a good choice of the initial guess u_0 for the exact solution u_* of the equation (1), the Newton method possesses quadratic convergence speed. However, it becomes challenging to solve exactly the Newton equation $F'(u)s = -F(u)$, especially when the dimension n is large.

To address this issue and raise the efficiency, on one hand, the inexact Newton method [21] was proposed and widely studied. The specific format of the inexact Newton method is as follows:

$$\|F(u_k) + F'(u_k)s_k\| \leqslant \eta_k \|F(u_k)\|, \quad k = 0, 1, \cdots, \tag{4}$$

note that the parameter $\eta_k \in [0, 1)$ here is the forcing term which is used to control the level of accuracy and $F'(u_k)$ represents the Jacobian matrix of the operator $F(u)$ evaluated at the iterate u_k. On the other hand, to expedite the solving process of the Newton method, the modified Newton method was presented and studied in [20], which is formulated as follows:

$$\begin{cases} v_k = u_k - F'(u_k)^{-1}F(u_k), \\ u_{k+1} = v_k - F'(u_k)^{-1}F(v_k), \quad k = 0, 1, 2, \cdots. \end{cases} \tag{5}$$

The theoretical and numerical analysis shows evidently that the modified Newton method (5) has achieved significant improvements. Compared with the Newton method, it requires only one additional evaluation F per step. However, the modified Newton method has at least R-order three of convergence. Over the years, inner-outer methods have attracted lots of attention for nonlinear systems due to their superior numerical performance, see [12, 18, 28, 34, 50, 53] and references therein. Therefore, for the nonlinear system (1), building on the efficient performance of the modified Newton method as an outer iteration, we need to search for more suitable inner methods for different forms of nonlinear problems to further improve effectiveness. To make the solver faster, we focus on finding efficient inner iterative methods to solve the following linear system.

Actually, we consider the linear system of equations

$$Au = b, \quad A \in \mathbb{C}^{n\times n}, \ u, \ b \in \mathbb{C}^n, \tag{6}$$

where A is a complex matrix with $A = W + \mathrm{i}T$, and $W, T \in \mathbb{R}^{n\times n}$ are real symmetric matrices, W being positive definite, and T being positive semi-definite. It is known that (6) is a special case of the generalized saddle-point problem [16, 17] which has broad applications, for instance, in the finite element discretization of constrained optimization problems for elliptic partial differential equations, or distributed control problems [2, 42, 43].

The methods for solving block two-by-two structured real linear equations have garnered significant attention from scholars. Among them, several kinds of preconditioned Krylov subspace methods [19, 45] have shown remarkable performance in handling large sparse matrices, effectively utilizing their structural characteristics to accelerate the solution process. Since 2003, the Hermitian and skew-Hermitian splitting (HSS) methods introduced by Bai et al. [9] have gained considerable attention for their effectiveness in solving large linear systems with non-Hermitian positive definite matrices. Researchers have devoted efforts to studying and refining the HSS-based methods [6, 8, 11, 55], drawn by their elegant formulation and reliable convergence properties. To circumvent the need for complex linear system computations, Bai et al. designed the modified HSS (MHSS) iteration method [3] to tackle the solution of equation (6) without resorting to complex linear system computations. Subsequently, massive improved methods have been proposed both in terms of HSS-version, such as the preconditioned MHSS (PMHSS) method [4] and the double-parameter generalized PMHSS (DGPMHSS) [33], etc., and in terms of SOR-version. However, as for the improved methods in terms of SOR-version, the original idea, and analysis of the generalized successive overrelaxation (GSOR) method were presented early in, 2005 [13] for solving augmented linear systems. Further generalizations and developments were extended by applying the GSOR method to address large sparse generalized saddle-point problems, as exemplified by the Uzawa method [14], and the GSOR-type (SOR-acceleration for HSS) methods were also established in [10]. Nearly 10 years later, certain straightforward applications of the GSOR methods and their accelerations as well as preconditioning were also researched in [23, 29, 46]. Especially inspired also by the preconditioning technique, Huang et al. introduced the preconditioned AGSOR (PAGSOR) method [30] whose aim is to decrease the optimal convergence factor of the AGSOR method and to enhance its convergence efficiency. All the above studies lead us to provide excellent inner iteration methods by developing two variants of the GSOR method, namely the accelerated GSOR (AGSOR) and the preconditioned GSOR (PGSOR) for linear systems.

As for nonlinear systems (1), by making use of the HSS-version iteration as the inner solver for the inexact Newton or modified Newton method, researchers have contributed a lot of excellent algorithms. See, for example, papers [12, 15, 28, 35, 48, 49, 53, 54] and references therein. In recent years the modified Newton-DGPMHSS (MN-DGPMHSS) method [18] and the modified Newton-DSS (MN-DSS) method [52] were also proposed to improve the algorithms. (Note that "D" for the two methods here represents "double" or "double-step". Do not confuse it with the "D" standing for "deteriorated" used in the initial study related to the HSS-type iteration methods [37]. Abbreviations always have limitations!) Due to the advantages of easy programming, efficient computing, and saving storage for SOR-version iteration and its variants, the modified Newton-GSOR (MN-GSOR) method and the modified Newton-AGSOR (MN-AGSOR) method were introduced [39, 40]. The local convergence properties of these methods were performed under a Hölder condition. The efficiency of their numerical results can be compared with those of the MN-DPMHSS and MN-MPPMHSS methods. In 2021, the modified Newton-PGSOR (MN-PGSOR) method was proposed to solve (1) with block two-by-two complex symmetric Jacobian matrices. It can get a similar conclusion while applying on a 2-torsion free block upper triangular matrix algebra [26]. All these methods have made progress in the study of solving a nonlinear system of equations.

To improve the computational efficiency and expand the range of problems when the Jacobian matrix is symmetric indefinite, we consider the possibility of applying the modified Newton method to solve a nonlinear system of equations while utilizing the preconditioned accelerated successive overrelaxation (PAGSOR) method to solve the Newton equations. Thus, we construct the modified Newton-PAGSOR (MN-PAGSOR) method to solve equations (1). Under proper conditions, the local convergence theorem is provided. Then, we use numerical results to show the feasibility and effectiveness of the method compared with several existing algorithms.

The article is organized as follows. In Sect. 2, we introduce the MN-PAGSOR method for handling (1) while elaborating on the properties of local convergence in Sect. 3. To substantiate its practicality and efficiency, Sect. 4 presents compelling numerical results that validate the effectiveness of the MN-PAGSOR method. Finally, in Sect. 5, a succinct conclusion is provided, summarizing the key findings and contributions of this article.

2 The Modified Newton-PAGSOR (MN-PAGSOR) Method

Consider the following complex linear system:

$$Au = b, \quad A \in \mathbb{C}^{n\times n}, \quad u,\ b \in \mathbb{C}^n. \tag{7}$$

The complex symmetric linear system is structured in the following form:

$$Au = (W + \mathrm{i}T)(x + \mathrm{i}y) = (p + \mathrm{i}q) = b, \tag{8}$$

where $\mathrm{i} = \sqrt{-1}$ is the imaginary unit, $W, T \in \mathbb{R}^{n\times n}$ are real, symmetric, and positive semi-definite matrices, and $x, y, p, q \in \mathbb{R}^n$ are real vectors. To avoid complex number operations, we can derive the equivalent real equations in the two-by-two form:

$$\begin{pmatrix} W & -T \\ T & W \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} p \\ q \end{pmatrix}. \tag{9}$$

2.1 The PAGSOR Iteration Method

Multiplying (9) to its left by the matrix

$$\mathcal{Z}_\omega = \begin{pmatrix} \omega I & I \\ -I & \omega I \end{pmatrix}, \quad \omega > 0, \tag{10}$$

we obtain

$$\begin{pmatrix} \omega W + T & W - \omega T \\ \omega T - W & \omega W + T \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \omega p + q \\ \omega q - p \end{pmatrix} := \widetilde{b}. \tag{11}$$

Denote $\widetilde{W} = \omega W + T$, $\widetilde{T} = \omega T - W$, $\widetilde{p} = \omega p + q$, $\widetilde{q} = \omega q - p$, and

$$\widetilde{A} = \begin{pmatrix} \widetilde{W} & -\widetilde{T} \\ \widetilde{T} & \widetilde{W} \end{pmatrix}.$$

Then, (9) can be rewritten as

$$\widetilde{A}u := \begin{pmatrix} \widetilde{W} & -\widetilde{T} \\ \widetilde{T} & \widetilde{W} \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \widetilde{p} \\ \widetilde{q} \end{pmatrix}.$$

Note that the coefficient matrix $\tilde{A}$ can be naturally split into

$$\widetilde{A} = \begin{pmatrix} \widetilde{W} & \boldsymbol{O} \\ \boldsymbol{O} & \widetilde{W} \end{pmatrix} - \begin{pmatrix} \boldsymbol{O} & \boldsymbol{O} \\ -\widetilde{T} & \boldsymbol{O} \end{pmatrix} - \begin{pmatrix} \boldsymbol{O} & \widetilde{T} \\ \boldsymbol{O} & \boldsymbol{O} \end{pmatrix} := \widetilde{D} - \widetilde{L} - \widetilde{U},$$

where the notation $\boldsymbol{O}$ is used to denote the zero matrix.

By [30], the PAGSOR method for $\tilde{A}u = \tilde{b}$ is established as follows.

The definition of $\widetilde{W}$, $\widetilde{T}$, $\widetilde{p}$, $\widetilde{q}$ are the same as those in (11), where α, β are two real parameters and $\omega > 0$.

Furthermore, the iterative method (12) can be equivalently rewritten as follows:

$$\begin{pmatrix} x_{k+1} \\ y_{k+1} \end{pmatrix} = Q_\omega(\alpha, \beta)\begin{pmatrix} x_k \\ y_k \end{pmatrix} + G_\omega(\alpha, \beta)\begin{pmatrix} \widetilde{p} \\ \widetilde{q} \end{pmatrix},$$

Algorithm 1 The PAGSOR method

1: Given an initial guess $u_0 \in (x_0^{\mathrm{T}}, y_0^{\mathrm{T}})^{\mathrm{T}} \in \mathbb{C}^{2n}$.
2: For $k = 0, 1, \cdots$, until $u_k = (x_k^{\mathrm{T}}, y_k^{\mathrm{T}})^{\mathrm{T}}$ converges, compute

$$\begin{cases} \widetilde{W} x_{k+1} = (1-\alpha)\widetilde{W} x_k + \alpha \widetilde{T} y_k + \alpha \widetilde{p}, \\ \widetilde{W} y_{k+1} = -\beta \widetilde{T} x_{k+1} + (1-\beta)\widetilde{W} y_k + \beta \widetilde{q}. \end{cases} \tag{12}$$

or

$$\begin{pmatrix} x_{k+1} \\ y_{k+1} \end{pmatrix} = Q_\omega(\alpha, \beta)^{k+1} \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} + \sum_{j=0}^{k} Q_\omega(\alpha, \beta)^j G_\omega(\alpha, \beta) \begin{pmatrix} \widetilde{p} \\ \widetilde{q} \end{pmatrix}, \ k = 0, 1, \cdots, \tag{13}$$

where

$$Q_\omega(\alpha, \beta) = \begin{pmatrix} \widetilde{W} & \boldsymbol{O} \\ \beta \widetilde{T} & \widetilde{W} \end{pmatrix}^{-1} \begin{pmatrix} (1-\alpha)\widetilde{W} & \alpha \widetilde{T} \\ \boldsymbol{O} & (1-\beta)\widetilde{W} \end{pmatrix}, \tag{14}$$

$$G_\omega(\alpha, \beta) = \begin{pmatrix} \widetilde{W} & \boldsymbol{O} \\ \beta \widetilde{T} & \widetilde{W} \end{pmatrix}^{-1} \begin{pmatrix} \alpha I & \boldsymbol{O} \\ \boldsymbol{O} & \beta I \end{pmatrix}. \tag{15}$$

Obviously, the matrix $Q_\omega(\alpha, \beta)$ is referred to as the iteration matrix of the PAGSOR method. Now, let us consider another decomposition of matrix $\widetilde{A}$,

$$\widetilde{A} = B_\omega(\alpha, \beta) - C_\omega(\alpha, \beta) \tag{16}$$

with

$$B_\omega(\alpha, \beta) = \begin{pmatrix} \alpha I & \boldsymbol{O} \\ \boldsymbol{O} & \beta I \end{pmatrix}^{-1} \begin{pmatrix} \widetilde{W} & \boldsymbol{O} \\ \beta \widetilde{T} & \widetilde{W} \end{pmatrix} = \begin{pmatrix} \frac{1}{\alpha} I & \boldsymbol{O} \\ \boldsymbol{O} & \frac{1}{\beta} I \end{pmatrix} \begin{pmatrix} \widetilde{W} & \boldsymbol{O} \\ \beta \widetilde{T} & \widetilde{W} \end{pmatrix}, \tag{17}$$

$$C_\omega(\alpha, \beta) = \begin{pmatrix} \frac{1}{\alpha} I & \boldsymbol{O} \\ \boldsymbol{O} & \frac{1}{\beta} I \end{pmatrix} \begin{pmatrix} (1-\alpha)\widetilde{W} & \alpha \widetilde{T} \\ \boldsymbol{O} & (1-\beta)\widetilde{W} \end{pmatrix}. \tag{18}$$

Evidently, we get

$$Q_\omega(\alpha, \beta) = B_\omega(\alpha, \beta)^{-1} C_\omega(\alpha, \beta), \quad G_\omega(\alpha, \beta) = B_\omega(\alpha, \beta)^{-1}. \tag{19}$$

Specifically, the AGSOR method, as outlined in [23], represents a particular form of the PAGSOR iterative method when $\omega = 1$.

2.2 The Modified Newton-PAGSOR Method

For convenience, we introduce the following notation:

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

where $\mathrm{Re}(u)$ and $\mathrm{Im}(u)$ represent the real and imaginary parts of any complex vector or matrix u, respectively. Naturally, $F(u)$ has the form $F(u) = P(u) + \mathrm{i}Q(u)$, where $P(u) = \mathrm{Re}(F(u))$, $Q(u) = \mathrm{Im}(F(u))$. We assume that the Jacobian matrix $F'(u)$ can be expressed in the form

$$F'(u) = W(u) + \mathrm{i}T(u),$$

where $W(u) = \mathrm{Re}(F'(u)) \in \mathbb{R}^{n\times n}$, $T(u) = \mathrm{Im}(F'(u)) \in \mathbb{R}^{n\times n}$ are real symmetric matrices, with $W(u)$ being positive definite and $T(u)$ positive semi-definite.

Our MN-PAGSOR iteration method is developed to solve the complex nonlinear system, with which the PAGSOR method applied to solve the following Newton equations:

$$\begin{cases} A(u_k)d_k = -H(u_k), \quad u_{k+\frac{1}{2}} = u_k + d_k, \\ A(u_k)h_k = -H(u_{k+\frac{1}{2}}), \quad u_{k+1} = u_{k+\frac{1}{2}} + h_k, \quad k = 0, 1, 2, \cdots, \end{cases} \tag{20}$$

where

$$A(u) = \begin{pmatrix} W(u) & -T(u) \\ T(u) & W(u) \end{pmatrix},$$

and

$$H(u) = \begin{pmatrix} P(u) \\ Q(u) \end{pmatrix}. \tag{21}$$

Hence, we can use the following expression for the MN-PAGSOR method:

$$\widetilde{A}u_k = \begin{pmatrix} \widetilde{W}(u_k) & -\widetilde{T}(u_k) \\ \widetilde{T}(u_k) & \widetilde{W}(u_k) \end{pmatrix} \begin{pmatrix} x_k \\ y_k \end{pmatrix} = \begin{pmatrix} \widetilde{P}(u_k) \\ \widetilde{Q}(u_k) \end{pmatrix} = \widetilde{H}(u_k) \tag{22}$$

with $\widetilde{W}(u_k) = \omega W(u_k) + T(u_k)$, $\widetilde{T}(u_k) = \omega T(u_k) - W(u_k)$, $u_k = x_k + \mathrm{i}y_k$, $\widetilde{P}(u_k) = \omega P(u_k) + Q(u_k)$, $\widetilde{Q}(u_k) = \omega Q(u_k) - P(u_k)$.

Same as (16), (17), and (18), we need to simplify (22) and provide an approximate method to generate the next iterate, say, namely u_{k+1},

$$B_\omega(\alpha, \beta) \begin{pmatrix} x_{k+1} \\ y_{k+1} \end{pmatrix} = C_\omega(\alpha, \beta) \begin{pmatrix} x_k \\ y_k \end{pmatrix} + \begin{pmatrix} \widetilde{P}(u_k) \\ \widetilde{Q}(u_k) \end{pmatrix}. \tag{23}$$

Thus, we can construct the MN-PAGSOR method. The algorithm process is as follows.

According to (13) and (20), the calculating leads the expressions for d_{k,l_k}, h_{k,m_k},

$$d_{k,l_k} = -\sum_{j=0}^{l_k-1} Q_\omega(\alpha, \beta, u_k)^j G_\omega(\alpha, \beta, u_k)\widetilde{H}(u_k),$$

$$h_{k,m_k} = -\sum_{j=0}^{m_k-1} Q_\omega(\alpha, \beta, u_k)^j G_\omega(\alpha, \beta, u_k)\widetilde{H}(u_{k+\frac{1}{2}}),$$

which are in the formula mentioned above and

$$Q_\omega(\alpha, \beta, u) = \begin{pmatrix} \widetilde{W}(u) & \boldsymbol{O} \\ \beta\widetilde{T}(u) & \widetilde{W}(u) \end{pmatrix}^{-1} \begin{pmatrix} (1-\alpha)\widetilde{W}(u) & \alpha\widetilde{T}(u) \\ \boldsymbol{O} & (1-\beta)\widetilde{W}(u) \end{pmatrix},$$

$$G_\omega(\alpha, \beta, u) = \begin{pmatrix} \widetilde{W}(u) & \boldsymbol{O} \\ \beta\widetilde{T}(u) & \widetilde{W}(u) \end{pmatrix}^{-1} \begin{pmatrix} \alpha I & \boldsymbol{O} \\ \boldsymbol{O} & \beta I \end{pmatrix}.$$

Therefore, the MN-PAGSOR method can be expressed as

$$\begin{cases} u_{k+\frac{1}{2}} = u_k - \sum\limits_{j=0}^{l_k-1} Q_\omega(\alpha, \beta, u_k)^j G_\omega(\alpha, \beta, u_k)\widetilde{H}(u_k), \\ u_{k+1} = u_{k+\frac{1}{2}} - \sum\limits_{j=0}^{m_k-1} Q_\omega(\alpha, \beta, u_k)^j G_\omega(\alpha, \beta, u_k)\widetilde{H}(u_{k+\frac{1}{2}}), \quad k = 0, 1, \cdots. \end{cases} \tag{24}$$

Algorithm 2 The modified Newton-PAGSOR (MN-PAGSOR) method

1: Given an initial guess $u_0 \in \mathbb{D}$, positive constants α, β, tol with $\omega > 0$, and two positive constant sequences $\{l_k\}_{k=0}^{\infty}, \{m_k\}_{k=0}^{\infty}$.

2: For $k = 0, 1, \cdots$, until $\|H(u_k)\| \leqslant \text{tol}\|H(u_0)\|$ do

2.1. Set $d_{k,0} := (x_{k,0}^{\mathrm{T}}, y_{k,0}^{\mathrm{T}})^{\mathrm{T}}$, where $x_{k,0}^{\mathrm{T}} = y_{k,0}^{\mathrm{T}} = 0$.

2.2. For $l = 0, 1, \cdots, l_k - 1$, apply PAGSOR to the first equation of (20),

$$\begin{cases} \widetilde{W}(u_k)x_{k,l+1} = (1-\alpha)\widetilde{W}(u_k)x_{k,l} + \alpha\widetilde{T}(u_k)y_{k,l} - \alpha\widetilde{P}(u_k), \\ \widetilde{W}(u_k)y_{k,l+1} = -\beta\widetilde{T}(u_k)x_{k,l+1} + (1-\beta)\widetilde{W}(u_k)y_{k,l} - \beta\widetilde{Q}(u_k), \end{cases}$$

and obtain $d_{k,l_k} = (x_{k,l_k}^{\mathrm{T}}, y_{k,l_k}^{\mathrm{T}})^{\mathrm{T}}$ such that

$$\|\widetilde{H}(u_k) + \widetilde{A}(u_k)d_{k,l_k}\| \leqslant \sigma_k\|\widetilde{H}(u_k)\|, \ \sigma_k \in [0, 1).$$

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

2.4. Compute $\widetilde{H}(u_{k+\frac{1}{2}})$.

2.5. Set $h_{k,0} := (\overline{x}_{k,0}^{\mathrm{T}}, \overline{y}_{k,0}^{\mathrm{T}})^{\mathrm{T}}$, where $\overline{x}_{k,0}^{\mathrm{T}} = \overline{y}_{k,0}^{\mathrm{T}} = 0$.

2.6. For $m = 0, 1, \cdots, m_k - 1$, apply PAGSOR to the second equation of (20),

$$\begin{cases} \widetilde{W}(u_k)\overline{x}_{k,m+1} = (1-\alpha)\widetilde{W}(u_k)\overline{x}_{k,m} + \alpha\widetilde{T}(u_k)\overline{y}_{k,m} - \alpha\widetilde{P}(u_k), \\ \widetilde{W}(u_k)\overline{y}_{k,m+1} = -\beta\widetilde{T}(u_k)\overline{x}_{k,m+1} + (1-\beta)\widetilde{W}(u_k)\overline{y}_{k,m} - \beta\widetilde{Q}(u_k), \end{cases}$$

and obtain $h_{k,m_k} = (\overline{x}_{k,m_k}^{\mathrm{T}}, \overline{y}_{k,m_k}^{\mathrm{T}})^{\mathrm{T}}$ such that

$$\left\|\widetilde{H}(u_{k+\frac{1}{2}}) + \widetilde{A}(u_k)h_{k,m_k}\right\| \leqslant \widetilde{\sigma_k}\|\widetilde{H}(u_{k+\frac{1}{2}})\|, \ \widetilde{\sigma_k} \in [0, 1).$$

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

3: End for.

Here, the matrix is split into

$$\widetilde{A}(u) = B_\omega(\alpha, \beta, u) - C_\omega(\alpha, \beta, u) \tag{25}$$

with

$$B_\omega(\alpha, \beta, u) = \begin{pmatrix} \frac{1}{\alpha}I & \boldsymbol{O} \\ \boldsymbol{O} & \frac{1}{\beta}I \end{pmatrix} \begin{pmatrix} \widetilde{W}(u) & \boldsymbol{O} \\ \beta\widetilde{T}(u) & \widetilde{W}(u) \end{pmatrix},$$

$$C_\omega(\alpha, \beta) = \begin{pmatrix} \frac{1}{\alpha}I & \boldsymbol{O} \\ \boldsymbol{O} & \frac{1}{\beta}I \end{pmatrix} \begin{pmatrix} (1-\alpha)\widetilde{W}(u) & \alpha\widetilde{T}(u) \\ \boldsymbol{O} & (1-\beta)\widetilde{W}(u) \end{pmatrix}.$$

Since these matrices have the following relationship with the iterative matrix $Q_\omega(\alpha, \beta, u)$:

$$\begin{aligned} &Q_\omega(\alpha, \beta, u) = B_\omega(\alpha, \beta, u)^{-1}\, C_\omega(\alpha, \beta, u), \quad G_\omega(\alpha, \beta, u) = B_\omega(\alpha, \beta, u)^{-1}, \\ &\widetilde{A}(u)^{-1} = (I - Q_\omega(\alpha, \beta, u))^{-1}G_\omega(\alpha, \beta, u), \end{aligned} \tag{26}$$

we obtain finally an equivalent form of the MN-PAGSOR method,

$$\begin{cases} u_{k+\frac{1}{2}} = u_k - (I - Q_\omega(\alpha, \beta, u_k)^{l_k})\ \widetilde{A}(u_k)^{-1}\ \widetilde{H}(u_k), \\ u_{k+1} = u_{k+\frac{1}{2}} - (I - Q_\omega(\alpha, \beta, u_k)^{m_k})\ \widetilde{A}(u_k)^{-1}\ \widetilde{H}(u_{k+\frac{1}{2}}), \quad k = 0, 1, \cdots. \end{cases} \tag{27}$$

3 Local Convergence of MN-PAGSOR Method

In this section, we derive the local convergence of the MN-PAGSOR method, i.e., the iterates u_k generated by the MN-PAGSOR method converge to u_* for any sufficiently good initial guess u_0. Our analysis is based on the Hölder condition, which is weaker than the Lipschitz one.

Definition 1 A mapping $F\colon \mathbb{D} \subseteq \mathbb{C}^n \to \mathbb{C}^n$ is nonlinear, if there exists a linear operator $A \in \mathbb{L}(\mathbb{R}^n, \mathbb{R}^n)$ that satisfies

$$\lim_{t\to 0} \frac{1}{t}\|F(u+th) - F(u) - tAh\| = 0$$

for any $h \in \mathbb{C}^n$, then F is Gateaux differentiable (or G-differentiable) at an interior point $u \in \mathbb{D}$. Additionally, for an open set $\mathbb{D}_0 \subset \mathbb{D}$, if the mapping $F\colon \mathbb{D} \subseteq \mathbb{C}^n \to \mathbb{C}^n$ is G-differentiable at every point in $\mathbb{D}_0$, it is said to be G-differentiable on the open set $\mathbb{D}_0$.

Lemma 1 (Perturbation Lemma [36]) *Let $A, C \in \mathbb{C}^{n\times n}$, and assume that A is nonsingular satisfying $\|A^{-1}\| \leqslant p$. If $\|A - C\| \leqslant q$ and $pq < 1$, then B is also nonsingular and*

$$\|B^{-1}\| \leqslant \frac{p}{1-pq}.$$

We further require that $F\colon \mathbb{D} \subset \mathbb{C}^n \to \mathbb{C}^n$ is G-differentiable on an open neighborhood $\mathbb{D}_0 \subset \mathbb{D}$ centered on $u_* \in \mathbb{D}$ with $F(u_*) = 0$, and that the Jacobian matrix $F'(u)$ is continuous and symmetric. Let $\mathbb{N}(u_*, r)$ denote an open ball centered at u_* with radius r. We find that the perturbation lemma plays a crucial role in the proofs of Lemma 2 and Theorem 1.

Assumption 1 For all $u \in \mathbb{N}(u_*, r) \subset \mathbb{D}_0$, the following conditions are established.

(A1) (The bounded condition) There exist positive constants δ and γ such that

$$\max\{\|W(u_*)\|, \|T(u_*)\|\} \leqslant \delta, \quad \|A(u_*)^{-1}\| \leqslant \gamma.$$

(A2) (The Hölder condition) There exist nonnegative constants K_w and K_t such that

$$\|W(u) - W(u_*)\| \leqslant K_w\|u - u_*\|^p,$$
$$\|T(u) - T(u_*)\| \leqslant K_t\|u - u_*\|^p$$

with the exponential $p \in (0, 1]$.

Lemma 2 *Under Assumption 1, if $r \in (0, (\frac{1}{\gamma K})^{\frac{1}{p}})$, then $A(u)^{-1}$ exists for $u \in \mathbb{N}(u_*, r)$. Besides, the following inequalities hold with $K := K_t + K_w$ for all $u, y \in \mathbb{N}(u_*, r) \subset \mathbb{D}_0$:*

(i) $\|A(u) - A(u_*)\| \leqslant K\|u - u_*\|^p$,
(ii) $\|A(u)^{-1}\| \leqslant \frac{\gamma}{1-\gamma K\|u-u_*\|^p}$,
(iii) $\|H(y)\| \leqslant \frac{K}{p+1}\|y - u_*\|^{p+1} + 2\delta\|y - u_*\|$,
(iv) $\|y - u_* - A(u)^{-1}H(y)\| \leqslant \frac{\gamma}{1-\gamma K\|u-u_*\|^p}\left(\frac{K}{p+1}\|y - u_*\|^p + K\|u - u_*\|^p\right)\|y - u_*\|$.

Proof (i) The Hölder condition directly implies

$$\begin{aligned}&\|A(u) - A(u_*)\|\\ =&\left\|\begin{pmatrix} W(u) - W(u_*) & T(u_*) - T(u) \\ T(u) - T(u_*) & W(u) - W(u_*)\end{pmatrix}\right\|\end{aligned}$$

$$\leqslant \left\| \begin{pmatrix} W(u)-W(u_*) & \boldsymbol{O} \\ \boldsymbol{O} & W(u)-W(u_*) \end{pmatrix} \right\| + \left\| \begin{pmatrix} \boldsymbol{O} & T(u_*)-T(u) \\ T(u)-T(u_*) & \boldsymbol{O} \end{pmatrix} \right\|$$
$$= \|W(u)-W(u_*)\| + \|T(u)-T(u_*)\|$$
$$\leqslant K_w\|u-u_*\|^p + K_t\|u-u_*\|^p = K\|u-u_*\|^p.$$

(ii) Making use of this inequality

$$\|A(u_*)^{-1}(A(u_*)-A(u))\| \leqslant \|A(u_*)^{-1}\|\|A(u_*)-A(u)\| \leqslant \gamma K\|u-u_*\|^p < 1,$$

and the perturbation lemma, we know that $A(u)^{-1}$ exists and satisfies

$$\|A(u)^{-1}\| \leqslant \frac{\|A(u_*)^{-1}\|}{1-\|A(u_*)^{-1}(A(u_*)-A(u))\|} \leqslant \frac{\gamma}{1-\gamma K\|u-u_*\|^p}.$$

(iii) Furthermore, the known bounded condition leads to

$$\|A(u_*)\| \leqslant \left\| \begin{pmatrix} W(u_*) & \boldsymbol{O} \\ \boldsymbol{O} & W(u_*) \end{pmatrix} \right\| + \left\| \begin{pmatrix} \boldsymbol{O} & -T(u_*) \\ T(u_*) & \boldsymbol{O} \end{pmatrix} \right\|$$
$$= \|W(u_*)\| + \|T(u_*)\| \leqslant 2\delta,$$

and

$$H(y) = H(y) - H(u_*) - A(u_*)(y-u_*) + A(u_*)(y-u_*)$$
$$= \int_0^1 (A(u_* + t(y-u_*)) - A(u_*))\mathrm{d}t(y-u_*) + A(u_*)(y-u_*).$$

Consequently, we find

$$\|H(y)\| \leqslant \left\| \int_0^1 (A(u_* + t(y-u_*)) - A(u_*))\mathrm{d}t(y-u_*) \right\| + \|A(u_*)(y-u_*)\|$$
$$\leqslant \frac{K}{p+1}\|y-u_*\|^{p+1} + 2\delta\|y-u_*\|.$$

(iv) Due to the following facts:

$$y - u_* - A(u)^{-1}H(y)$$
$$= -A(u)^{-1}\Big(H(y) - H(u_*) - A(u_*)(y-u_*)\Big) + A(u)^{-1}\Big(A(u) - A(u_*)\Big)(y-u_*)$$
$$= -A(u)^{-1}\int_0^1 \Big(A(u_* + t(y-u_*)) - A(u_*)\Big)\mathrm{d}t(y-u_*)$$
$$+ A(u)^{-1}\Big(A(u) - A(u_*)\Big)(y-u_*),$$

we have finally,

$$\|y - u_* - A_\lambda(u)^{-1}H(y)\|$$
$$\leqslant \| - A(u)^{-1}\int_0^1 \Big(A(u_* + t(y-u_*)) - A(u_*)\Big)\mathrm{d}t(y-u_*)\|$$
$$+ \|A(u)^{-1}\Big(A(u) - A(u_*)\Big)(y-u_*)\|$$
$$\leqslant \|A(u)^{-1}\|\Big(\int_0^1 \|(A(u_* + t(y-u_*)) - A(u_*)\|\mathrm{d}t + \|(A(u) - A(u_*)\|\Big)\|y-u_*\|$$

$$\leqslant \frac{\gamma}{1-\gamma K\|u-u_*\|^p}\left(\frac{K}{p+1}\|y-u_*\|^p + K\|u-u_*\|^p\right)\|y-u_*\|.$$

The proof of Lemma 2 is completed.

Theorem 1 *Under the assumptions of Lemma 2, denote* $\Delta = \min\{\alpha, \beta\}$ *and* $\Lambda = \max\{|1-\alpha|, |1-\beta|\}$, *suppose* $r \in (0, r_0)$, *define* $r_0 := \min\{r_1, r_2, r_3\}$, *where*

$$r_1 = \left(\frac{(1+\omega^2)\Delta}{2\gamma(1+\omega)[(\omega+\beta)K_w + (\beta\omega+1)K_t]}\right)^{\frac{1}{p}},$$
$$r_2 =$$
$$\left(\frac{\tau\theta(1+\omega^2)\Delta}{2\gamma(1+\omega)\Big([(\Lambda\omega+1)+(1+\tau\theta)(\omega+\beta)]K_w + [(\alpha\omega+\Lambda)+(1+\tau\theta)(\beta\omega+1)]K_t\Big)}\right)^{\frac{1}{p}},$$
$$r_3 = \left(\frac{1-2\delta\gamma\,[(\tau+1)\theta]^{\nu_*}}{4K\gamma}\right)^{\frac{1}{p}}$$

with $\nu_* = \min\{l_*, m_*\}$, $l_* = \liminf_{k\to\infty} l_k$, $m_* = \liminf_{k\to\infty} m_k$, *and the quantity* ν_* *satisfies*

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

where the symbol $\lfloor\cdot\rfloor$ *is used to denote the smallest integer no less than the corresponding real number* $\tau \in (0, \frac{1-\theta}{\theta})$ *a predetermined positive constant and*

$$\theta \equiv \theta(\alpha, \beta, u_*) = \|Q(\alpha, \beta, u_*)\| < 1$$

with α, β *satisfying* $0 < \alpha\beta < \alpha+1 < \alpha\beta\frac{1-\rho(\widetilde{W}(u_*)^{-1}\widetilde{T}(u_*))}{2} + 2$. *Then, for any* $u \in \mathbb{N}(u_*, r)$ *and any sequences* $\{l_k\}_{k=0}^{\infty}$, $\{m_k\}_{k=0}^{\infty}$ *of positive integers, the iteration sequence* $\{u_k\}_{k=0}^{\infty}$ *generated by the MN-PAGSOR method is well-defined, and hence converges to* u_*. *In addition, the following inequality holds*:

$$\limsup_{k\to\infty} \|u_k - u_*\|^{\frac{1}{k}} \leqslant g(r_0^p; \nu_*)^2,$$

where, when $z \in (0, r)$ *and* $\iota > \nu_*$, *then*

$$g(z^p, \iota) = \frac{\gamma}{1-\gamma K z^p}\left(3Kz^p + 2\delta[(\tau+1)\theta]^{\iota}\right),$$
$$\|Q_\omega(\alpha, \beta, u)\| \leqslant (\tau+1)\theta < 1.$$

Proof First of all, by making use of the inverse matrix

$$\mathcal{Z}_\omega^{-1} = \begin{pmatrix} \omega I & I \\ -I & \omega I \end{pmatrix}^{-1} = \begin{pmatrix} \frac{\omega}{1+\omega^2} I & -\frac{1}{1+\omega^2} I \\ \frac{1}{1+\omega^2} I & \frac{\omega}{1+\omega^2} I \end{pmatrix},$$

we have

$$\|\widetilde{A}(u_*)^{-1}\| = \|(\mathcal{Z}A(u_*))^{-1}\| \leqslant \|A(u_*)\|^{-1}\|\mathcal{Z}\|^{-1}$$
$$\leqslant \gamma \left\| \begin{pmatrix} \frac{\omega}{1+\omega^2} I & -\frac{1}{1+\omega^2} I \\ \frac{1}{1+\omega^2} I & \frac{\omega}{1+\omega^2} I \end{pmatrix} \right\|$$

$$\leqslant \gamma\left(\left\|\begin{pmatrix} \frac{\omega}{1+\omega^2}I & \boldsymbol{O} \\ \boldsymbol{O} & \frac{\omega}{1+\omega^2}I \end{pmatrix}\right\| + \left\|\begin{pmatrix} \boldsymbol{O} & -\frac{1}{1+\omega^2}I \\ \frac{1}{1+\omega^2}I & \boldsymbol{O} \end{pmatrix}\right\|\right)$$
$$\leqslant \gamma\left(\frac{\omega}{1+\omega^2} + \frac{1}{1+\omega^2}\right) = \gamma\,\frac{1+\omega}{1+\omega^2}.$$

Based on (26) and the inequality $\|Q_\omega(\alpha,\beta,u_*)\| \leqslant \vartheta(\alpha,\beta,u_*) < 1$ [30], the boundedness condition implies

$$\begin{aligned} \|B_\omega(\alpha,\beta,u_*)^{-1}\| &= \|(I - Q_\omega(\alpha,\beta,u_*))\widetilde{A}(u_*)^{-1}\| \\ &\leqslant \|I - Q_\omega(\alpha,\beta,u_*)\|\|\widetilde{A}(u_*)^{-1}\| \\ &\leqslant (1 + \|Q_\omega(\alpha,\beta,u_*)\|)\|\widetilde{A}(u_*)^{-1}\| \leqslant 2\gamma\,\frac{1+\omega}{1+\omega^2}. \end{aligned}$$

From the Hölder condition, we obtain

$$\begin{aligned} \|\widetilde{W}(u) - \widetilde{W}(u_*)\| &= \|\omega W(u) + T(u) - \omega W(u_*) - T(u_*)\| \\ &\leqslant \omega\|W(u) - W(u_*)\| + \|T(u) - T(u_*)\| \\ &\leqslant \omega K_w\|u - u_*\|^p + K_t\|u - u_*\|^p, \\ \|\widetilde{T}(u) - \widetilde{T}(u_*)\| &= \|\omega T(u) - W(u) - \omega T(u_*) + W(u_*)\| \\ &\leqslant \omega\|T(u) - T(u_*)\| + \|W(u) - W(u_*)\| \\ &\leqslant \omega K_t\|u - u_*\|^p + K_w\|u - u_*\|^p. \end{aligned}$$

Furthermore, according to Assumptions (A2), two estimates are produced as follows:

$$\begin{aligned} &\|B_\omega(\alpha,\beta,u) - B_\omega(\alpha,\beta,u_*)\| \\ &= \left\|\begin{pmatrix} \frac{1}{\alpha}I & \boldsymbol{O} \\ \boldsymbol{O} & \frac{1}{\beta}I \end{pmatrix}\begin{pmatrix} \widetilde{W}(u) & \boldsymbol{O} \\ \beta\widetilde{T}(u) & \widetilde{W}(u) \end{pmatrix} - \begin{pmatrix} \frac{1}{\alpha}I & \boldsymbol{O} \\ \boldsymbol{O} & \frac{1}{\beta}I \end{pmatrix}\begin{pmatrix} \widetilde{W}(u_*) & \boldsymbol{O} \\ \beta\widetilde{T}(u_*) & \widetilde{W}(u_*) \end{pmatrix}\right\| \\ &\leqslant \left\|\begin{pmatrix} \frac{1}{\alpha}I & \boldsymbol{O} \\ \boldsymbol{O} & \frac{1}{\beta}I \end{pmatrix}\right\|\left\|\begin{pmatrix} \widetilde{W}(u) - \widetilde{W}(u_*) & \boldsymbol{O} \\ \beta(\widetilde{T}(u) - \widetilde{T}(u_*)) & \widetilde{W}(u) - \widetilde{W}(u_*) \end{pmatrix}\right\| \\ &\leqslant \left\|\begin{pmatrix} \frac{1}{\alpha}I & \boldsymbol{O} \\ \boldsymbol{O} & \frac{1}{\beta}I \end{pmatrix}\right\|\left[\left\|\begin{pmatrix} \widetilde{W}(u) - \widetilde{W}(u_*) & \boldsymbol{O} \\ \boldsymbol{O} & \widetilde{W}(u) - \widetilde{W}(u_*) \end{pmatrix}\right\| + \left\|\begin{pmatrix} \boldsymbol{O} & \boldsymbol{O} \\ \beta(\widetilde{T}(u) - \widetilde{T}(u_*)) & \boldsymbol{O} \end{pmatrix}\right\|\right] \\ &\leqslant \max\left\{\frac{1}{\alpha}, \frac{1}{\beta}\right\}\left[\|\widetilde{W}(u) - \widetilde{W}(u_*)\| + \beta\|\widetilde{T}(u) - \widetilde{T}(u_*)\|\right] \\ &\leqslant \frac{1}{\Delta}\left(\omega K_w\|u - u_*\|^p + K_t\|u - u_*\|^p + \beta\omega K_t\|u - u_*\|^p + \beta K_w\|u - u_*\|^p\right) \\ &= \left(\frac{\omega+\beta}{\Delta}K_w + \frac{\beta\omega+1}{\Delta}K_t\right)\|u - u_*\|^p, \\ &\|C_\omega(\alpha,\beta,u) - C_\omega(\alpha,\beta,u_*)\| \\ &= \left\|\begin{pmatrix} \frac{1}{\alpha}I & \boldsymbol{O} \\ \boldsymbol{O} & \frac{1}{\beta}I \end{pmatrix}\begin{pmatrix} (1-\alpha)\widetilde{W}(u) & \alpha\widetilde{T}(u) \\ \boldsymbol{O} & (1-\beta)\widetilde{W}(u) \end{pmatrix} - \begin{pmatrix} \frac{1}{\alpha}I & \boldsymbol{O} \\ \boldsymbol{O} & \frac{1}{\beta}I \end{pmatrix}\begin{pmatrix} (1-\alpha)\widetilde{W}(u_*) & \alpha\widetilde{T}(u_*) \\ \boldsymbol{O} & (1-\beta)\widetilde{W}(u_*) \end{pmatrix}\right\| \\ &\leqslant \left\|\begin{pmatrix} \frac{1}{\alpha}I & \boldsymbol{O} \\ \boldsymbol{O} & \frac{1}{\beta}I \end{pmatrix}\right\|\left[\left\|\begin{pmatrix} (1-\alpha)(\widetilde{W}(u) - \widetilde{W}(u_*)) & \boldsymbol{O} \\ \boldsymbol{O} & (1-\beta)(\widetilde{W}(u) - \widetilde{W}(u_*)) \end{pmatrix}\right\|\right. \end{aligned}$$

$$
\begin{aligned}
&\quad + \left\| \begin{pmatrix} \boldsymbol{O} & \alpha(\widetilde{T}(u) - \widetilde{T}(u_*)) \\ \boldsymbol{O} & \boldsymbol{O} \end{pmatrix} \right\| \Bigg] \\
&\leqslant \max\left\{\frac{1}{\alpha}, \frac{1}{\beta}\right\} \left[\max\{|1-\alpha|, |1-\beta|\} \|\widetilde{W}(u) - \widetilde{W}(u_*)\| + \alpha \|\widetilde{T}(u) - \widetilde{T}(u_*)\|\right] \\
&\leqslant \frac{1}{\Delta}\left(\Lambda\omega K_w \|u-u_*\|^p + \Lambda K_t \|u-u_*\|^p + \alpha\omega K_t \|u-u_*\|^p + K_w \|u-u_*\|^p\right) \\
&= \left(\frac{\Lambda\omega+1}{\Delta} K_w + \frac{\alpha\omega+\Lambda}{\Delta} K_t\right) \|u-u_*\|^p.
\end{aligned}
$$

If r satisfies

$$
2\gamma \frac{1+\omega}{1+\omega^2}\left[\left(\frac{\omega+\beta}{\Delta} K_w + \frac{\beta\omega+1}{\Delta} K_t\right)\|u-u_*\|^p\right] < 1,
$$

then, by Lemma 1, we can further obtain

$$
\begin{aligned}
\|B_\omega(\alpha,\beta,u)^{-1}\| &= \frac{\|B_\omega(\alpha,\beta,u_*)^{-1}\|}{1 - \|B_\omega(\alpha,\beta,u_*)^{-1}\| \|B_\omega(\alpha,\beta,u_*) - B_\omega(\alpha,\beta,u)\|} \\
&\leqslant \frac{2\gamma \frac{1+\omega}{1+\omega^2}}{1 - 2\gamma \frac{1+\omega}{1+\omega^2}\left[\left(\frac{\omega+\beta}{\Delta} K_w + \frac{\beta\omega+1}{\Delta} K_t\right)\|u-u_*\|^p\right]} \\
&= \frac{2\gamma(1+\omega)}{1+\omega^2 - 2\gamma(1+\omega)\left[\left(\frac{\omega+\beta}{\Delta} K_w + \frac{\beta\omega+1}{\Delta} K_t\right)\|u-u_*\|^p\right]}.
\end{aligned}
$$

Since $r < r_1$, the following inequality holds:

$$
\|u-u_*\|^p < \frac{(1+\omega^2)\Delta}{2\gamma(1+\omega)\left[(\omega+\beta)K_w + (\beta\omega+1)K_t\right]}.
$$

Therefore,

$$
\begin{aligned}
&\quad \|Q_\omega(\alpha,\beta,u) - Q_\omega(\alpha,\beta,u_*)\| \\
&= \|B_\omega(\alpha,\beta,u)^{-1}(C_\omega(\alpha,\beta,u) - C_\omega(\alpha,\beta,u_*)) \\
&\quad + (B_\omega(\alpha,\beta,u)^{-1} - B_\omega(\alpha,\beta,u_*)^{-1})C_\omega(\alpha,\beta,u_*)\| \\
&\leqslant \|B_\omega(\alpha,\beta,u)^{-1}\| \\
&\quad \cdot \Big(\|C_\omega(\alpha,\beta,u) - C_\omega(\alpha,\beta,u_*)\| + \|B_\omega(\alpha,\beta,u) - B_\omega(\alpha,\beta,u_*)\|\|Q_\omega(\alpha,\beta,u_*)\|\Big) \\
&\leqslant \frac{2\gamma\frac{1+\omega}{1+\omega^2}\left[\left(\frac{\Lambda\omega+1}{\Delta}K_w + \frac{\Lambda+\alpha\omega}{\Delta}K_t\right)\|u-u_*\|^p + \left(\frac{\omega+\beta}{\Delta}K_w + \frac{\beta\omega+1}{\Delta}K_t\right)\|u-u_*\|^p\right]}{1 - 2\gamma\frac{1+\omega}{1+\omega^2}\left[\left(\frac{\omega+\beta}{\Delta}K_w + \frac{\beta\omega+1}{\Delta}K_t\right)\|u-u_*\|^p\right]} \\
&= \frac{2\gamma\frac{1+\omega}{1+\omega^2}\left(\frac{(\Lambda+1)\omega+\beta+1}{\Delta}K_w + \frac{(\alpha+\beta)\omega+\Lambda+1}{\Delta}K_t\right)\|u-u_*\|^p}{1 - 2\gamma\frac{1+\omega}{1+\omega^2}\left[\left(\frac{\omega+\beta}{\Delta}K_w + \frac{\beta\omega+1}{\Delta}K_t\right)\|u-u_*\|^p\right]}.
\end{aligned}
$$

Moreover, to establish the inequality $\|Q_\omega(\alpha,\beta,u) - Q_\omega(\alpha,\beta,u_*)\| \leqslant \tau\theta$, the following inequality holds:

$$
\frac{2\gamma\frac{1+\omega}{1+\omega^2}\left(\frac{(\Lambda+1)\omega+\beta+1}{\Delta}K_w + \frac{(\alpha+\beta)\omega+\Lambda+1}{\Delta}K_t\right)\|u-u_*\|^p}{1 - 2\gamma\frac{1+\omega}{1+\omega^2}\left[\left(\frac{\omega+\beta}{\Delta}K_w + \frac{\beta\omega+1}{\Delta}K_t\right)\|u-u_*\|^p\right]} \leqslant \tau\theta,
$$

so it follows that

$$\|u-u_*\|^p \leqslant \frac{\tau\theta(1+\omega^2)(1+\omega)^{-1}\Delta}{2\gamma\left[(\Lambda\omega+1)+(1+\tau\theta)(\omega+\beta)\right]K_w+\left[(\alpha\omega+\Lambda)+(1+\tau\theta)(\beta\omega+1)\right]K_t}.$$

Thus, when $r<\min\{r_1,r_2\}$, for any $u\in\mathbb{N}(u_*,r)$, we have

$$\|Q_\omega(\alpha,\beta,u)\|\leqslant\|Q_\omega(\alpha,\beta,u)-Q_\omega(\alpha,\beta,u_*)\|+\|Q_\omega(\alpha,\beta,u_*)\|\leqslant\tau\theta+\theta=(\tau+1)\theta<1.$$

Next, the error of the iterative sequence$\{u_k\}_{k=0}^{\infty}$ generated by the MN-PAGSOR method can be estimated using (27) and Lemma 2, yielding

$$\begin{aligned}
\|u_{k+\frac{1}{2}}-u_*\| &= \|u_k-u_*-(I-Q_\omega(\alpha,\beta,u_k)^{l_k})A(u_k)^{-1}H(u_k)\|\\
&\leqslant \|u_k-u_*-A(u_k)^{-1}H(u_k)\|+\|Q_\omega(\alpha,\beta,u_k)^{l_k}\|\|A(u_k)^{-1}H(u_k)\|\\
&\leqslant \frac{\gamma}{1-\gamma K\|u_k-u_*\|^p}\left(\frac{K}{p+1}\|u_k-u_*\|^{p+1}+K\|u_k-u_*\|^{p+1}\right)\\
&\quad+\frac{\gamma[(\tau+1)\theta]^{l_k}}{1-\gamma K\|u_k-u_*\|^p}\left(\frac{K}{p+1}\|u_k-u_*\|^{p+1}+2\delta\|u_k-u_*\|\right)\\
&\leqslant \frac{\gamma K(p+2)}{(p+1)(1-\gamma K\|u_k-u_*\|^p)}\|u_k-u_*\|^{p+1}\\
&\quad+\frac{\gamma[(\tau+1)\theta]^{l_k}}{1-\gamma K\|u_k-u_*\|^p}\left(\frac{K}{p+1}\|u_k-u_*\|^{p+1}+2\delta\|u_k-u_*\|\right)\\
&\leqslant \frac{\gamma K(p+2)+\gamma K[(\tau+1)\theta]^{l_k}}{(p+1)(1-\gamma K\|u_k-u_*\|^p)}\|u_k-u_*\|^{p+1}\\
&\quad+\frac{2\delta\gamma[(\tau+1)\theta]^{l_k}}{1-\gamma K\|u_k-u_*\|^p}\|u_k-u_*\|\\
&= \frac{\gamma}{1-\gamma K\|u_k-u_*\|^p}\\
&\quad\cdot\left\{\frac{K(p+2+[(\tau+1)\theta]^{l_k})}{p+1}\|u_k-u_*\|^p+2\delta[(\tau+1)\theta]^{l_k}\right\}\|u_k-u_*\|\\
&:= g(\|u_k-u_*\|^p,l_k)\|u_k-u_*\|<\|u_k-u_*\|.
\end{aligned}$$

Here, we use the notation

$$g(z^p,\iota)=\frac{\gamma}{1-\gamma Kz^p}\{3Kz^p+2\delta[(\tau+1)\theta]^{\iota}\}.$$

Considering the assumption that

$$\gamma\{3Kr_0^p+2\delta[(\tau+1)\theta]^{\nu_*}\}<1-\gamma Kr_0^p,$$

which is equivalent to

$$r_0<\left\{\frac{1-2\delta\gamma[(\tau+1)\theta]^{\nu_*}}{4K\gamma}\right\}^{\frac{1}{p}}:=r_3.$$

From the fact that $u_k\in\mathbb{N}(u_*,r)$, we can conclude that the following inequality holds:

$$g(\|u_k-u_*\|^p,l_k)\leqslant\frac{\gamma}{1-\gamma Kr_0^p}\left\{3Kr_0^p+2\delta[(\tau+1)\theta]^{\nu_*}\right\}=g(r_0^p,\nu_*)<1.$$

Therefore, when $r < r_3$

$$\|u_{k+\frac{1}{2}} - u_*\| < \|u_k - u_*\|.$$

Similarly,

$$\begin{aligned}
&\|u_{k+1} - u_*\| \\
&= \left\|u_{k+\frac{1}{2}} - u_* - (I - Q_\omega(\alpha, \beta, u_k)^{m_k})A(u_k)^{-1}H(u_{k+\frac{1}{2}})\right\| \\
&\leqslant \left\|u_{k+\frac{1}{2}} - u_* - A(u_k)^{-1}H\left(u_{k+\frac{1}{2}}\right)\| + \|A(u_k)^{-1}H(u_{k+\frac{1}{2}})\right\| \\
&\leqslant \frac{\gamma}{1 - \gamma K\|u_k - u_*\|^p}\Big(\frac{K}{p+1}\|u_{k+\frac{1}{2}} - u_*\|^{p+1} + K\|u_k - u_*\|^p\Big)\|u_{k+\frac{1}{2}} - u_*\| \\
&\quad + \frac{[(\tau+1)\theta]^{m_k}\gamma}{1 - \gamma K\|u_k - u_*\|^p}\left(\frac{K}{p+1}\|u_{k+\frac{1}{2}} - u_*\|^{p+1} + 2\delta\|u_{k+\frac{1}{2}} - u_*\|\right) \\
&\leqslant \frac{\gamma K}{1 - \gamma K\|u_k - u_*\|^P}\left\{\frac{1 + [(\tau+1)\theta]^{m_k}}{p+1}\|u_{k+\frac{1}{2}} - u_*\|^{p+1}\right\} \\
&\quad + \frac{\gamma}{1 - \gamma K\|u_k - u_*\|^P}\left\{K\|u_k - u_*\|^p + 2\delta[(\tau+1)\theta]^{m_k}\right\}\|u_{k+\frac{1}{2}} - u_*\| \\
&\leqslant \frac{\gamma g(\|u_k - u_*\|^p, l_k)}{1 - \gamma K\|u_k - u_*\|^p}\|u_k - u_*\|\left\{\frac{g(\|u_k - u_*\|^p, l_k)^p(1 + [(\tau+1)\theta]^{m_k}) + p + 1}{p+1}\right. \\
&\quad \left.\cdot K\|u_k - u_*\|^p + 2\delta[(\tau+1)\theta]^{m_k}\right\} \\
&\leqslant \frac{\gamma g(\|u_k - u_*\|^p, l_k)}{1 - \gamma K\|u_k - u_*\|^p}\|u_k - u_*\|\left\{[(2g(\|u_k - u_*\|^p, l_k)^p + 1)]K\|u_k - u_*\|^p\right. \\
&\quad \left. + 2\delta[(\tau+1)\theta]^{m_k}\right\} \\
&\leqslant \frac{\gamma g(\|u_k - u_*\|^p, l_k)}{1 - \gamma K\|u_k - u_*\|^p}\left\{3K\|u_k - u_*\|^p + 2\delta[(\tau+1)\theta]^{m_k}\right\}\|u_k - u_*\| \\
&\leqslant g(\|u_k - u_*\|^p, l_k)g(\|u_k - u_*\|^p, m_k)\|u_k - u_*\| \\
&< g(r_0^p, \nu_*)^2\|u_k - u_*\| < \|u_k - u_*\|.
\end{aligned}$$

That is

$$\|u_{k+1} - u_*\| < \|u_k - u_*\|.$$

For arbitrary $u_0 \in \mathbb{N}(u_*, r) \subset \mathbb{D}_0$, the following inequality holds:

$$0 \leqslant \cdots < \|u_{k+1} - u_*\| < \|u_k - u_*\| < \cdots < \|u_0 - u_*\| < r.$$

Consequently, the iterative sequence $\{u_k\}_{k=0}^{\infty}$ is well-posed and converges to u_*. Additionally, from $\|u_{k+1} - u_*\| < g(r_0^p, \nu_*)^2\|u_k - u_*\|$, it follows that $\|u_k - u_*\| < g(r_0^p, \nu_*)^{2k}\|u_0 - u_*\|$, the following inequality is correct:

$$\|u_k - u_*\|^{\frac{1}{k}} < g(r_0^p, \nu_*)^2\|u_0 - u_*\|^{\frac{1}{k}},$$

as $k \to \infty$, $\limsup_{k\to\infty}\|u_k - u_*\|^{\frac{1}{k}} \leqslant g(r_0^p, \nu_*)^2$.

Table 1 The optimal parameters of Newton methods for Example 1

Method	N	$\rho = 1$			$\rho = 10$			$\rho = 200$		
		0.1	0.2	0.4	0.1	0.2	0.4	0.1	0.2	0.4
MN-GSOR	30	1.12	1.14	1.12	1.13	1.14	1.13	1.15	1.13	1.12
	40	1.13	1.15	1.14	1.13	1.14	1.14	1.14	1.17	1.15
	50	1.13	1.14	1.14	1.16	1.15	1.14	1.15	1.17	1.15
	100	1.13	1.14	1.12	1.15	1.13	1.13	1.13	1.15	1.14
MN-DGPMHSS	30	(3.51, 2.78)	(1.50, 2.20)	(0.78, 0.05)	(3.89, 2.56)	(1.70, 2.50)	(1.63, 0.19)	(3.60, 2.50)	(1.53, 2.57)	(1.60, 0.20)
	40	(4.98, 2.78)	(1.50, 2.30)	(1.55, 0.04)	(7.10, 2.60)	(1.30, 2.10)	(1.80, 0.07)	(8.30, 2.10)	(1.50, 2.10)	(1.80, 0.20)
	50	(3.75, 2.78)	(1.50, 2.30)	(0.93, 0.03)	(3.55, 2.46)	(1.23, 1.89)	(0.93, 0.03)	(3.11, 2.74)	(1.20, 2.10)	(1.08, 0.09)
	100	(5.70, 2.50)	(4.20, 3.10)	(1.60, 0.18)	(6.00, 2.50)	(4.20, 3.10)	(1.60, 0.04)	(6.00, 2.50)	(0.70, 1.50)	(1.60, 0.05)
MN-AGSOR	30	(1.25, 0.85)	(1.35, 0.77)	(0.91, 1.25)	(1.33, 0.68)	(1.32, 0.71)	(1.32, 0.75)	(1.33, 0.72)	(1.33, 0.77)	(1.32, 0.65)
	40	(1.23, 0.89)	(0.65, 1.43)	(0.90, 1.25)	(1.32, 0.70)	(1.38, 0.65)	(1.30, 0.77)	(1.33, 0.70)	(1.48, 0.50)	(1.30, 0.67)
	50	(1.23, 0.91)	(1.33, 0.81)	(0.90, 1.25)	(1.32, 0.71)	(1.38, 0.65)	(1.30, 0.77)	(1.31, 0.69)	(1.38, 0.67)	(1.30, 0.67)
	100	(1.23, 0.90)	(1.33, 0.80)	(0.91, 1.23)	(1.32, 0.68)	(1.38, 0.63)	(1.26, 0.77)	(1.31, 0.69)	(1.38, 0.65)	(1.30, 0.67)
MN-PAGSOR	30	(1.09, 0.85, 5)	(1.18, 0.77, 5)	(1.05, 0.92, 5)	(1.09, 0.85, 5)	(1.18, 0.75, 5)	(1.05, 0.92, 5)	(1.09, 0.83, 5)	(1.19, 0.75, 5)	(1.05, 0.89, 5)
	40	(1.06, 0.89, 4.8)	(1.18, 0.75, 5.2)	(1.05, 0.97, 4.6)	(1.09, 0.87, 4.8)	(1.18, 0.73, 4.8)	(1.05, 0.95, 5)	(1.09, 0.85, 5)	(1.17, 0.73, 4.8)	(1.05, 0.92, 5)
	50	(1.07, 0.89, 5)	(1.17, 0.74, 4.8)	(1.05, 0.95, 4.8)	(1.09, 0.85, 4.8)	(1.18, 0.73, 5)	(1.05, 0.95, 4.8)	(1.09, 0.85, 5)	(1.17, 0.73, 5)	(1.05, 0.93, 4.8)
	100	(1.07, 0.89, 5.2)	(1.18, 0.75, 5)	(1.05, 0.92, 5)	(1.09, 0.85, 5)	(1.18, 0.75, 4.8)	(1.05, 0.92, 4.8)	(1.09, 0.83, 4.8)	(1.20, 0.77, 4.8)	(1.03, 0.93, 4.8)

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

N	Method	Error	CPU time /s	Outer IT	Inner IT
30	MN-DGPMHSS	7.800 9E−11	0.120 584	5	50
	MN-GSOR	2.675 8E−12	0.098 382	5	38
	MN-AGSOR	1.420 6E−11	0.078 997	5	30
	MN-PAGSOR	5.562 1E−11	0.046 000	3	11
40	MN-DGPMHSS	8.147 5E−11	0.195 103	5	50
	MN-GSOR	1.577 0E−11	0.164 509	5	38
	MN-AGSOR	1.491 3E−11	0.155 867	5	30
	MN-PAGSOR	2.580 8E−11	0.077 629	3	10
50	MN-DGPMHSS	7.362 4E−11	0.335 193	5	50
	MN-GSOR	1.743 3E−11	0.289 515	5	38
	MN-AGSOR	3.989 1E−11	0.284 656	5	30
	MN-PAGSOR	1.131 8E−11	0.154 249	3	11
100	MN-DGPMHSS	1.775 6E−11	4.202 227	5	51
	MN-GSOR	2.176 7E−11	3.628 505	5	38
	MN-AGSOR	2.998 9E−11	3.430 989	5	30
	MN-PAGSOR	1.528 8E−11	1.945 217	3	11

Table 3 Numerical results of the modified Newton methods for $\sigma = 0.2$, $\rho = 1$

N	Method	Error	CPU time /s	Outer IT	Inner IT
30	MN-DGPMHSS	1.902 6E−11	0.121 887	7	56
	MN-GSOR	1.977 0E−11	0.108 956	7	40
	MN-AGSOR	5.970 4E−11	0.106 965	6	36
	MN-PAGSOR	6.249 6E−11	0.055 428	4	15
40	MN-DGPMHSS	4.838 2E−11	0.227 057	7	56
	MN-GSOR	2.332 6E−11	0.195 556	7	43
	MN-AGSOR	9.831 7E−12	0.183 247	6	35
	MN-PAGSOR	5.995 7E−11	0.076 687	4	15
50	MN-DGPMHSS	5.031 5E−11	0.412 128	7	56
	MN-GSOR	2.592 1E−11	0.351 044	7	40
	MN-AGSOR	5.265 4E−11	0.289 406	6	36
	MN-PAGSOR	2.385 1E−11	0.166 705	4	15
100	MN-DGPMHSS	2.741 2E−11	5.373 081	7	56
	MN-GSOR	3.268 3E−11	4.735 138	7	40
	MN-AGSOR	4.228 7E−11	4.263 990	6	36
	MN-PAGSOR	5.530 6E−11	2.826 609	4	15

Table 4 Numerical results of the modified Newton methods for $\sigma = 0.4$, $\rho = 1$

N	Method	Error	CPU time /s	Outer IT	Inner IT
30	MN-DGPMHSS	5.114 1E−11	0.120 345	10	40
	MN-GSOR	1.928 1E−11	0.118 700	9	35
	MN-AGSOR	3.879 7E−11	0.095 492	8	30
	MN-PAGSOR	1.369 9E−11	0.065 757	5	10
40	MN-DGPMHSS	1.155 7E−11	0.303 169	11	40
	MN-GSOR	4.171 2E−11	0.241 678	10	39
	MN-AGSOR	6.204 0E−12	0.195 496	8	30
	MN-PAGSOR	4.746 6E−11	0.081 093	5	10
50	MN-DGPMHSS	1.802 1E−11	0.500 210	11	44
	MN-GSOR	4.615 2E−11	0.474 547	10	39
	MN-AGSOR	3.342 0E−11	0.353 337	8	30
	MN-PAGSOR	3.249 2E−11	0.208 084	5	10
100	MN-DGPMHSS	1.204 8E−11	7.079 849	11	39
	MN-GSOR	3.040 6E−11	6.072 706	9	35
	MN-AGSOR	1.147 6E−11	5.293 261	8	30
	MN-PAGSOR	1.704 0E−11	3.073 592	5	10

Table 5 Numerical results of the modified Newton methods for $\sigma = 0.1$, $\rho = 10$

N	Method	Error	CPU time /s	Outer IT	Inner IT
30	MN-DGPMHSS	2.031 4E−11	0.100 946	5	50
	MN-GSOR	1.134 9E−11	0.085 217	5	38
	MN-AGSOR	4.993 6E−11	0.077 970	5	30
	MN-PAGSOR	5.510 9E−11	0.044 893	3	11
40	MN-DGPMHSS	1.489 7E−11	0.203 865	5	55
	MN-GSOR	1.418 1E−11	0.176 482	5	38
	MN-AGSOR	4.185 1E−11	0.156 505	5	30
	MN-PAGSOR	6.346 7E−11	0.084 406	3	11
50	MN-DGPMHSS	1.075 7E−11	0.340 376	5	50
	MN-GSOR	1.574 0E−11	0.304 275	5	47
	MN-AGSOR	5.913 6E−11	0.262 748	5	30
	MN-PAGSOR	4.696 9E−11	0.148 914	3	11
100	MN-DGPMHSS	1.614 2E−11	4.259 832	5	52
	MN-GSOR	3.724 0E−12	3.705 591	5	47
	MN-AGSOR	3.043 5E−11	3.692 982	5	30
	MN-PAGSOR	6.016 5E−11	2.057 005	3	11

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

N	Method	Error	CPU time /s	Outer IT	Inner IT
30	MN-DGPMHSS	4.365 4E−11	0.118 764	7	56
	MN-GSOR	1.656 4E−11	0.109 521	7	40
	MN-AGSOR	2.157 2E−12	0.090 571	6	34
	MN-PAGSOR	4.917 8E−11	0.052 512	4	15
40	MN-DGPMHSS	6.474 6E−11	0.227 317	7	56
	MN-GSOR	2.092 0E−11	0.200 483	7	40
	MN-AGSOR	2.170 2E−11	0.177 334	6	36
	MN-PAGSOR	5.065 4E−11	0.098 589	4	15
50	MN-DGPMHSS	1.964 4E−11	0.422 523	7	56
	MN-GSOR	7.892 2E−12	0.370 530	7	45
	MN-AGSOR	2.334 2E−11	0.306 032	6	36
	MN-PAGSOR	4.896 8E−11	0.188 928	4	15
100	MN-DGPMHSS	2.742 1E−11	5.427 040	7	56
	MN-GSOR	6.309 9E−12	5.280 223	7	40
	MN-AGSOR	1.581 4E−11	4.196 347	6	36
	MN-PAGSOR	4.722 5E−11	2.579 745	4	15

Table 7 Numerical results of the modified Newton methods for $\sigma = 0.4$, $\rho = 10$

N	Method	Error	CPU time /s	Outer IT	Inner IT
30	MN-DGPMHSS	1.002 0E−11	0.113 105	10	35
	MN-GSOR	7.306 0E−11	0.105 737	9	35
	MN-AGSOR	7.239 4E−11	0.096 697	8	31
	MN-PAGSOR	1.273 4E−11	0.062 436	5	10
40	MN-DGPMHSS	1.700 8E−11	0.262 789	11	40
	MN-GSOR	3.748 0E−11	0.251 012	10	39
	MN-AGSOR	2.886 1E−11	0.222 843	8	31
	MN-PAGSOR	4.128 5E−11	0.121 071	5	10
50	MN-DGPMHSS	3.770 9E−11	0.474 582	11	44
	MN-GSOR	4.290 6E−11	0.429 339	10	39
	MN-AGSOR	3.130 3E−11	0.386 290	8	31
	MN-PAGSOR	3.144 8E−11	0.192 462	5	10
100	MN-DGPMHSS	5.766 1E−11	7.940 930	12	44
	MN-GSOR	1.160 2E−11	6.687 529	10	39
	MN-AGSOR	3.300 4E−11	4.554 565	7	27
	MN-PAGSOR	1.242 3E−11	3.077 437	5	10

Table 8 Numerical results of the modified Newton methods for $\sigma = 0.1$, $\rho = 200$

N	Method	Error	CPU time /s	Outer IT	Inner IT
30	MN-DGPMHSS	2.470 6E−11	0.102 227	5	50
	MN-GSOR	1.469 4E−11	0.085 781	5	38
	MN-AGSOR	3.134 9E−11	0.076 307	5	30
	MN-PAGSOR	9.764 0E−11	0.044 492	3	11
40	MN-DGPMHSS	1.594 9E−11	0.210 889	5	58
	MN-GSOR	1.097 6E−11	0.166 603	5	38
	MN-AGSOR	3.655 6E−11	0.129 767	5	30
	MN-PAGSOR	3.484 3E−11	0.082 394	3	11
50	MN-DGPMHSS	6.714 1E−11	0.333 962	5	50
	MN-GSOR	1.912 2E−12	0.305 178	5	45
	MN-AGSOR	7.607 4E−12	0.279 037	5	30
	MN-PAGSOR	3.905 6E−11	0.142 852	3	11
100	MN-DGPMHSS	1.813 2E−11	4.234 100	5	55
	MN-GSOR	1.420 8E−11	3.841 254	5	38
	MN-AGSOR	1.387 9E−11	3.597 929	5	30
	MN-PAGSOR	3.542 4E−11	2.204 937	3	11

Table 9 Numerical results of the modified Newton methods for $\sigma = 0.2$, $\rho = 200$

N	Method	Error	CPU time /s	Outer IT	Inner IT
30	MN-DGPMHSS	8.423 9E−11	0.124 350	7	56
	MN-GSOR	1.952 8E−11	0.100 784	6	33
	MN-AGSOR	6.076 2E−12	0.088 052	6	34
	MN-PAGSOR	5.773 4E−11	0.062 059	4	15
40	MN-DGPMHSS	3.888 6E−12	0.233 740	7	56
	MN-GSOR	4.214 4E−11	0.188 127	6	44
	MN-AGSOR	1.382 3E−11	0.182 406	6	36
	MN-PAGSOR	5.943 8E−11	0.082 677	4	15
50	MN-DGPMHSS	1.309 7E−11	0.433 615	7	62
	MN-GSOR	8.763 1E−11	0.384 466	6	44
	MN-AGSOR	2.005 5E−11	0.309 260	6	36
	MN-PAGSOR	4.974 1E−11	0.193 905	4	15
100	MN-DGPMHSS	4.802 2E−12	6.409 312	8	80
	MN-GSOR	3.996 6E−12	5.785 244	8	46
	MN-AGSOR	3.181 2E−11	4.295 026	6	36
	MN-PAGSOR	5.774 0E−11	2.592 212	4	16

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

N	Method	Error	CPU time /s	Outer IT	Inner IT
30	MN-DGPMHSS	3.214 9E−11	0.137 166	11	44
	MN-GSOR	4.001 6E−11	0.117 106	8	30
	MN-AGSOR	3.619 3E−11	0.095 203	7	27
	MN-PAGSOR	4.195 4E−12	0.065 514	5	10
40	MN-DGPMHSS	7.011 2E−11	0.275 775	11	43
	MN-GSOR	2.792 3E−11	0.251 744	10	39
	MN-AGSOR	1.969 7E−11	0.190 349	7	27
	MN-PAGSOR	1.474 6E−11	0.098 116	5	10
50	MN-DGPMHSS	5.224 3E−11	0.537 783	12	48
	MN-GSOR	5.518 3E−11	0.443 670	10	39
	MN-AGSOR	2.708 3E−11	0.319 982	7	27
	MN-PAGSOR	8.732 5E−12	0.204 236	5	10
100	MN-DGPMHSS	2.957 0E−11	6.796 726	10	54
	MN-GSOR	3.757 2E−11	6.374 288	10	39
	MN-AGSOR	4.639 7E−11	4.707 271	7	27
	MN-PAGSOR	9.818 5E−11	2.469 681	4	8

4 Numerical Results

In this section, we list two frequently encountered complex nonlinear systems and compute their numerical solutions by using the respective methods. The experimental results are then compared to verify the effectiveness and feasibility of our approach. The known methods we have experimented with are MN-GSOR, MN-DGPMHSS, and MN-AGSOR methods which utilize the modified Newton method as the external iteration and GSOR, DGPMHSS, and AGSOR as the internal iterations. To ensure the reliability of the results, we have compared not only the CPU time consumption of the algorithms, but also their inner and outer iteration steps. The optimal parameters of each method and detailed information on computational results are listed below for comparison. In the tables below, the error estimation is denoted as Error, CPU time in seconds is referred to as CPU time (s), outer iteration steps are indicated as Outer IT, and inner iteration steps as Inner IT. The numerical experiments demonstrate that our method outperforms the MN-GSOR, MN-DGPMHSS, and MN-AGSOR methods. All experiments are implemented on MATLAB [version 9.12.0.2039608 (R2022a)] and run on a personal computer with a configuration of 8-core Central Processing Unit [Apple M2] and 16.00 GB memory.

Example 1 Consider the following nonlinear equations:

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

where $\Omega = (0, 1) \times (0, 1)$, and $\partial\Omega$ is its boundary. The coefficients $\alpha_1 = \alpha_2 = 1,\ \beta_1 = \beta_2 = 0.5$, and ρ is a positive constant used to control the reaction term. By discretizing the above problem on equidistant grids $\Delta t = h = \frac{1}{N+1}$ (N is a given positive constant), at each temporal step of the implicit scheme, we should consider a system of nonlinear equations (1)

Table 11 The optimal parameters of Newton methods for Example 2

Method	N	$\rho = 10$			$\rho = 200$		
		0.1	0.2	0.4	0.1	0.2	0.4
MN-GSOR	120	0.53	0.57	0.55	0.54	0.61	0.64
	130	0.46	0.55	0.65	0.45	0.57	0.60
MN-DGPMHSS	120	(0.90, 2.10)	(1.20, 3.00)	(1.30, 2.70)	(1.10, 2.50)	(1.10, 3.00)	(1.20, 2.60)
	130	(1.10, 2.70)	(1.20, 3.30)	(1.30, 2.50)	(0.90, 2.30)	(1.40, 3.50)	(1.10, 2.80)
MN-AGSOR	120	(1.38, 0.41)	(1.55, 0.53)	(1.44, 0.46)	(1.42, 0.47)	(1.53, 0.31)	(1.55, 0.25)
	130	(1.36, 0.41)	(1.35, 0.71)	(1.44, 0.56)	(1.35, 0.41)	(1.42, 0.73)	(1.51, 0.63)
MN-PAGSOR	120	(1.03, 0.89,5)	(1.03, 0.96, 5)	(1.03, 0.89, 5)	(1.05, 0.98, 5)	(1.03, 0.87, 5)	(1.03, 0.98, 5)
	130	(1.03, 0.87, 5)	(1.03, 0.98, 5)	(1.03, 0.98, 5)	(1.05, 0.96, 5)	(1.03, 0.87, 5)	(1.03, 0.98, 5)

Table 12 Numerical results of the modified Newton methods for $\sigma = 0.1$, $\rho = 10$

N	Method	Error	CPU time /s	Outer IT	Inner IT
120	MN-DGPMHSS	9.192 2E−11	13.917 641	5	80
	MN-GSOR	7.118 3E−12	11.710 870	5	34
	MN-AGSOR	1.081 8E−12	11.689 832	5	39
	MN-PAGSOR	3.843 5E−11	7.320 436	3	9
130	MN-DGPMHSS	1.864 6E−11	22.443 281	5	90
	MN-GSOR	1.409 9E−12	20.923 777	5	44
	MN-AGSOR	1.751 4E−12	20.444 629	5	39
	MN-PAGSOR	8.551 4E−11	11.899 742	3	9

Table 13 Numerical results of the modified Newton methods for $\sigma = 0.2$, $\rho = 10$

N	Method	Error	CPU time /s	Outer IT	Inner IT
120	MN-DGPMHSS	5.692 9E−12	18.841 101	8	96
	MN-GSOR	2.398 2E−11	15.144 429	6	29
	MN-AGSOR	5.205 9E−11	14.257 997	6	59
	MN-PAGSOR	4.554 3E−12	9.007 631	5	10
130	MN-DGPMHSS	5.382 2E−11	30.562 848	7	98
	MN-GSOR	3.753 5E−11	25.115 404	6	30
	MN-AGSOR	1.332 9E−11	24.472 399	6	36
	MN-PAGSOR	1.607 7E−11	19.219 866	5	10

Table 14 Numerical results of the modified Newton methods for $\sigma = 0.4$, $\rho = 10$

N	Method	Error	CPU time /s	Outer IT	Inner IT
120	MN-DGPMHSS	5.823 5E−11	25.716 952	12	72
	MN-GSOR	7.066 5E−11	22.386 256	9	29
	MN-AGSOR	6.010 1E−11	22.235 619	9	36
	MN-PAGSOR	2.038 1E−12	10.040 248	5	10
130	MN-DGPMHSS	7.765 0E−11	43.451 126	11	66
	MN-GSOR	1.150 0E−11	39.995 351	10	24
	MN-AGSOR	4.246 7E−11	39.378 616	10	40
	MN-PAGSOR	1.607 7E−11	19.532 667	5	10

of the form

$$F(u) = Mu + (\alpha_2 + \mathrm{i}\beta_2)h\Delta t\Psi(u) = 0,$$

where

$$M = h(1 + \rho\Delta t)I_n + (\alpha_1 + \mathrm{i}\beta_1)\frac{\Delta t}{h}(A_N \otimes I_N + I_N \otimes A_N),$$
$$\Psi(u) = \left(u_1^{\frac{4}{3}}, u_2^{\frac{4}{3}}, \cdots, u_n^{\frac{4}{3}}\right)^{\mathrm{T}},$$

Table 15 Numerical results of the modified Newton methods for $\sigma = 0.1$, $\rho = 200$

N	Method	Error	CPU time /s	Outer IT	Inner IT
120	MN-DGPMHSS	2.981 8E−11	13.163 033	5	80
	MN-GSOR	3.182 7E−12	12.657 697	5	34
	MN-AGSOR	7.954 6E−13	11.231 970	5	40
	MN-PAGSOR	1.874 4E−11	7.775 676	3	11
130	MN-DGPMHSS	7.271 6E−11	22.075 086	5	90
	MN-GSOR	3.675 4E−12	20.795 815	5	44
	MN-AGSOR	2.621 3E−12	20.284 470	5	38
	MN-PAGSOR	7.244 0E−11	12.453 950	3	10

Table 16 Numerical results of the modified Newton methods for $\sigma = 0.2$, $\rho = 200$

N	Method	Error	CPU time /s	Outer IT	Inner IT
120	MN-DGPMHSS	3.969 7E−11	17.731 372	7	98
	MN-GSOR	3.106 8E−12	14.671 533	6	28
	MN-AGSOR	2.558 6E−11	13.716 425	6	47
	MN-PAGSOR	2.968 1E−11	9.845 576	4	9
130	MN-DGPMHSS	4.502 1E−12	33.451 363	8	96
	MN-GSOR	9.060 3E−11	25.122 212	6	27
	MN-AGSOR	2.123 1E−11	24.255 072	6	47
	MN-PAGSOR	2.846 9E−11	16.227 290	4	9

Table 17 Numerical results of the modified Newton methods for $\sigma = 0.4$, $\rho = 200$

N	Method	Error	CPU time /s	Outer IT	Inner IT
120	MN-DGPMHSS	1.670 0E−11	28.345 480	13	78
	MN-GSOR	4.142 6E−11	24.110 517	10	24
	MN-AGSOR	6.177 0E−11	24.006 785	11	59
	MN-PAGSOR	1.321 2E−11	12.279 764	5	10
130	MN-DGPMHSS	7.124 3E−11	45.747 911	11	88
	MN-GSOR	4.175 5E−11	36.676 261	9	26
	MN-AGSOR	1.949 8E−11	39.796 555	10	59
	MN-PAGSOR	1.353 2E−11	19.451 493	5	10

and A_N is a tridiagonal matrix of the following form:

$$A_N = \text{tridiag}(-1, 2, -1) \in \mathbb{R}^{N \times N}.$$

Here, the vector $u = (u_1, u_2, \cdots, u_n)^{\mathrm{T}}$, $n = N^2$, and $\otimes$ represents the Kronecker product.

We choose $u_0 = 1$ to start the numerical computation and use the following stopping criterion for the outer iteration:

$$\frac{\|H(u_k)\|_2}{\|H(u_0)\|_2} \leqslant 10^{-10}.$$

To control the accuracy of the inner iteration, both σ_k and $\widetilde{\sigma_k}$ are set to be σ. As a result, we need the following inequalities:

$$\frac{\|\widetilde{H}(u_k)+\widetilde{A}(u_k)d_{k,l_k}\|}{\|\widetilde{H}(u_k)\|}\leqslant\sigma,$$

$$\frac{\left\|\widetilde{H}\left(u_{k+\frac{1}{2}}\right)+\widetilde{A}(u_k)h_{k,m_k}\right\|}{\left\|\widetilde{H}\left(u_{k+\frac{1}{2}}\right)\right\|}\leqslant\sigma.$$

In Table 1, we have listed the optimal parameter α for the MN-GSOR method, (α,β) for MN-DGPMHSS and MN-AGSOR methods, and (α,β,ω) for the MN-PAGSOR method. These optimal parameters obtained from each of the methods in the numerical experiments listed here are used to compare their performance. As for a one-parameter method, there is conducted theoretical analysis for the optimal parameter; see, for example, [6, 7, 9] and references therein. But for any kind of multi-parameter method, there is either theoretical analysis for optimal parameters which is too complicated to be available, or actually no such theoretical analysis. So, we directly use numerical experimentation to determine the choice of optimal parameters. Furthermore, the numerical results show us that, as the size of the nonlinear system problem increases, the optimal parameters of the MN-PAGSOR method remain relatively stable.

In our numerical experiments, the size of the discretization N, the control parameter ρ, and the inner tolerance σ are chosen unified as: $N=30,40,50,100$, $\rho=1,10,200$, and $\sigma=0.1,0.2,0.4$.

In Tables 2–10, the four methods MN-GSOR, MN-DGPMHSS, MN-AGSOR, and MN-PAGSOR are compared for efficiency. Our numerical results in these tables demonstrate that the proposed MN-PAGSOR method has a smaller computation time than that of the other three methods. Additionally, the MN-PAGSOR method requires fewer inner and outer iteration steps. It can also be observed that the iteration steps of the MN-PAGSOR method remain relatively stable as the problem size increases. In other words, the MN-PAGSOR method converges more quickly compared to other methods. It is an effective improvement and one gets a promising approach.

Example 2 Consider the following two-dimensional complex nonlinear convection-diffusion equation:

$$\begin{cases}-(\alpha_1+\mathrm{i}\beta_1)(u_{xx}+u_{yy})+\rho u=-(\alpha_2+\mathrm{i}\beta_2)u\mathrm{e}^u, & (x,y)\text{ in }\Omega,\\ u(x,y)=0, & (x,y)\text{ on }\partial\Omega,\end{cases}$$

where $\Omega=(0,1)\times(0,1)$, and $\partial\Omega$ is its boundary. The coefficients $\alpha_1=1,\alpha_2=-1,\ \beta_1=\beta_2=0.5$, and ρ is a positive constant used to control the reaction term. By discretizing above problem on equidistant grids $h=\frac{1}{N+1}$ (N is a given positive constant), we should consider a system of nonlinear equations (1) of the form

$$F(u)=Mu+(\alpha_2+\mathrm{i}\beta_2)h^2\phi(u)=0,$$

where

$$M=\rho h^2 I_n+(\alpha_1+\mathrm{i}\beta_1)(A_N\otimes I_N+I_N\otimes A_N),$$
$$\phi(u)=(u_1\mathrm{e}^{u_1},u_2\mathrm{e}^{u_2},\dots,u_n\mathrm{e}^{u_n})^{\mathrm{T}}$$

with the vector $u = (u_1, u_2, \cdots, u_n)^{\mathrm{T}}$, $n = N^2$. The symbols A_N, N, and $\otimes$ are consistent with Example 1.

Table 11 lists the optimal parameters for the four methods we mentioned before.

In our numerical experiments, the size of the discretization N, the control parameter ρ, and the inner tolerance σ are chosen unified as: $N = 120, 130$, $\rho = 10, 200$, and $\sigma = 0.1, 0.2, 0.4$.

In Tables 12–17, the four methods MN-GSOR, MN-DGPMHSS, MN-AGSOR, and MN-PAGSOR are compared for efficiency. We observe that our MN-PAGSOR method consistently outperforms the MN-GSOR, MN-DGPMHSS, and MN-AGSOR methods in terms of inner iteration steps, outer iteration steps, and CPU time under the same problem size. This highlights the superior performance of our proposed MN-PAGSOR algorithm in solving a class of complex nonlinear systems with complex symmetric Jacobians. Furthermore, the numerical results indicate that the iteration steps of the MN-PAGSOR method remain relatively stable as the problem size varies, demonstrating its robust stability. These findings reinforce the effectiveness and promising potential of the MN-PAGSOR method.

5 Conclusion

In this study, we propose the modified Newton-PAGSOR method for a class of large sparse systems of nonlinear equations with complex symmetric Jacobian matrices. Additionally, we investigate the local convergence of the MN-PAGSOR method under the Hölder continuity conditions. Numerical results demonstrate the feasibility and effectiveness of the new method, showing that MN-PAGSOR outperforms other methods in terms of computational time and number of iterations. Moreover, our numerical experiments reveal that the optimal parameters selected for MN-PAGSOR are more stable compared to other methods. However, further research is still needed to investigate the semi-local convergence analysis and optimal selection of multiple parameters, which remains an interesting problem for future study.

Acknowledgements The authors would like to thank two anonymous referees for their responsible and valuable comments which eventually lead to an improved presentation of the article.

Funding This work is supported by the National Natural Science Foundation of China with Grant Nos. 12161030 and 12171216.

Data Availability The data are available. Our data are mainly expressed by the numerical results.

Declarations

Conflict of Interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

1. Axelsson, O., Kucherov, A.: Real valued iterative methods for solving complex symmetric linear systems. Numer. Linear Algebra Appl. **7**, 197–218 (2000)
2. Bai, Z.-Z.: Block preconditioners for elliptic PDE-constrained optimization problems. Computing **91**, 379–395 (2011)
3. Bai, Z.-Z., Benzi, M., Chen, F.: Modified HSS iteration methods for a class of complex symmetric linear systems. Computing **87**, 93–111 (2010)
4. Bai, Z.-Z., Benzi, M., Chen, F.: On preconditioned MHSS iteration methods for complex symmetric linear systems. Numer. Algorithms **56**, 297–317 (2011)

5. 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**, 343–369 (2013)
6. Bai, Z.-Z., Golub, G.H.: Accelerated Hermitian and skew-Hermitian splitting iteration methods for saddle-point problems. IMA J. Numer. Anal. **27**, 1–23 (2007)
7. Bai, Z.-Z., Golub, G.H., Li, C.-K.: Optimal parameter in Hermitian and skew-Hermitian splitting method for certain two-by-two block matrices. SIAM J. Sci. Comput. **28**, 584–603 (2006)
8. 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**, 844–863 (2005)
9. 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**, 603–626 (2003)
10. 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**, 319–335 (2007)
11. Bai, Z.-Z., Golub, G.H., Pan, J.-Y.: Preconditioned Hermitian and skew-Hermitian splitting methods for non-Hermitian positive semidefinite linear systems. Numerische Mathematik **98**, 1–32 (2004)
12. Bai, Z.-Z., Guo, X.-P.: On Newton-HSS methods for systems of nonlinear equations with positive-definite Jacobian matrices. J. Comput. Math. **28**, 235–260 (2010)
13. Bai, Z.-Z., Parlett, B.N., Wang, Z.-Q.: On generalized successive overrelaxation methods for augmented linear systems. Numerische Mathematik **102**, 1–38 (2005)
14. Bai, Z.-Z., Wang, Z.-Q.: On parameterized inexact Uzawa methods for generalized saddle point problems. Linear Algebra Appl. **428**, 2900–2932 (2008)
15. Bai, Z.-Z., Yang, X.: On HSS-based iteration methods for weakly nonlinear systems. Appl. Numer. Math. **59**, 2923–2936 (2009)
16. Benzi, M., Golub, G.H.: A preconditioner for generalized saddle point problems. SIAM J. Matrix Anal. Appl. **26**, 20–41 (2004)
17. Benzi, M., Golub, G.H., Liesen, J.: Numerical solution of saddle point problems. Acta Numerica **14**, 1–137 (2005)
18. Chen, M.-H., Wu, Q.-B.: On modified Newton-DGPMHSS method for solving nonlinear systems with complex symmetric Jacobian matrices. Comput. Math. Appl. **76**, 45–57 (2018)
19. da Cunha, R.D., Becker, D.: Dynamic block GMRES: an iterative method for block linear systems. Adv. Comput. Math. **27**, 423–448 (2007)
20. Darvishi, M.T., Barati, A.: A third-order Newton-type method to solve systems of nonlinear equations. Appl. Math. Comput. **187**, 630–635 (2007)
21. Dembo, R., Eisenstat, S., Steihaug, T.: Inexact Newton method. SIAM J. Numer. Anal. **19**, 400–408 (1982)
22. Deuflhard, P.: Newton Methods for Nonlinear Problems. Springer-Verlag, Berlin, Heidelberg (2004)
23. Edalatpour, V., Hezari, D., Salkuyeh, D.K.: Accelerated generalized SOR method for a class of complex systems of linear equations. Math. Commun. **20**, 37–52 (2015)
24. Elman, H.C.: Preconditioners for saddle point problems arising in computational fluid dynamics. Appl. Numer. Math. **43**, 75–89 (2002)
25. Elman, H.C., Silvester, D.J., Wathen, A.: Finite Elements and Fast Iterative Solvers: with Applications in Incompressible Fluid Dynamics. Oxford University Press, Oxford (2014)
26. 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**, 1–18 (2021)
27. Feriani, A., Perotti, F., Simoncini, V.: Iterative system solvers for the frequency analysis of linear mechanical systems. Comp. Methods Appl. Mech. Eng. **190**, 1719–1739 (2000)
28. Guo, X.-P., Duff, I.S.: Semilocal and global convergence of the Newton-HSS method for systems of nonlinear equations. Numer. Linear Algebra Appl. **18**, 299–315 (2011)
29. Hezari, D., Edalatpour, V., Salkuyeh, D.K.: Preconditioned GSOR iterative method for a class of complex symmetric system of linear equations. Numer. Linear Algebra Appl. **22**, 761–776 (2015)
30. Huang, Z.-G., Wang, L.-G., Xu, Z., Cui, J.-J.: Preconditioned accelerated generalized successive overrelaxation method for solving complex symmetric linear systems. Comput. Math. Appl. **77**, 1902–1916 (2019)
31. Karlsson, H.O.: The quasi-minimal residual algorithm applied to complex symmetric linear systems in quantum reactive scattering. J. Chem. Phys. **103**, 4914–4919 (1995)
32. Kuramoto, Y.: Oscillations Chemical Waves and Turbulence. Dover, Mineola (2003)
33. Li, C.-X., Wu, S.-L.: A double-parameter GPMHSS method for a class of complex symmetric linear systems from Helmholtz equation. Math. Prob. Eng. **2014**, 1–7 (2014)
34. Li, Y., Guo, X.-P.: Semilocal convergence analysis for MMN-HSS methods under the Hölder conditions. East Asia J. Appl. Math. **7**, 396–416 (2017)

35. Li, Y., Guo, X.-P.: Multi-step modified Newton-HSS methods for systems of nonlinear equations with positive definite Jacobian matrices. Numer. Algorithms **75**, 55–80 (2017)
36. Ortega, J.M., Rheinboldt, W.C.: Iterative Solution of Nonlinear Equations in Several Variables. Society for Industrial and Applied Mathematics, Philadelphia, PA (2000)
37. Pan, J.-Y., Ng, M.K., Bai, Z.-Z.: New preconditioners for saddle point problems. Appl. Math. Comput. **172**, 762–771 (2006)
38. Papp, D., Vizvari, B.: Effective solution of linear Diophantine equation systems with an application in chemistry. J. Math. Chem. **39**, 15–31 (2006)
39. Qi, X., Wu, H.-T., Xiao, X.-Y.: Modified Newton-GSOR method for solving complex nonlinear systems with symmetric Jacobian matrices. Comput. Appl. Math. **39**, 1–18 (2020)
40. 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**, 1–23 (2020)
41. Raviart, P.A., Girault, V.: Finite Element Approximation of the Navier-Stokes Equations. Springer Verlag, Berlin, New York (1979)
42. Rees, T., Dollar, H.S., Wathen, A.J.: Optimal solvers for PDE-constrained optimization. SIAM Journal on Scientific Computing **32**, 271–298 (2010)
43. Rees, T., Stoll, M.: Block-triangular preconditioners for PDE-constrained optimization. Numer. Linear Algebra Appl. **17**, 977–996 (2010)
44. Rheinboldt, W.C.: Methods for Solving Systems of Nonlinear Equations. Society for Industrial and Applied Mathmatics, Philadelphia, PA (1998)
45. Saad, Y.: Iterative Methods for Sparse Linear Systems, 2nd edn. Society for Industrial and Applied Mathematics, Philadelphia, PA (2003)
46. Salkuyeh, D.K., Hezari, D., Edalatpour, V.: Generalized successive overrelaxation iterative method for a class of complex symmetric linear system of equations. Int. J. Comp. Math. **92**, 802–815 (2015)
47. Sulem, C., Sulem, P.L.: The Nonlinear Schrödinger Equation Self-focusing and Wave Collapse. Springer, New York (2007)
48. Wang, J., Guo, X.-P., Zhong, H.-X.: Accelerated GPMHSS method for solving complex systems of linear equations. East Asia J. Appl. Math. **7**, 143–155 (2017)
49. Wang, J., Guo, X.-P., Zhong, H.-X.: DPMHSS iterative method for systems of nonlinear equations with block two-by-two complex Jacobian matrices. Numer. Algorithms **77**, 167–184 (2018)
50. Wu, Q.-B., Chen, M.-H.: Convergence analysis of modified Newton-HSS method for solving systems of nonlinear equations. Numer. Algorithms **64**, 659–683 (2013)
51. Xiao, X.-Y., Wang, X., Yin, H.-W.: Efficient single-step preconditioned HSS iteration methods for complex symmetric linear systems. Comput. Math. Appl. **74**, 2269–2280 (2017)
52. 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. Algorithms **85**, 951–975 (2020)
53. Yang, A.-L., Wu, Y.-J.: Newton-MHSS methods for solving systems of nonlinear equations with complex symmetric Jacobian matrices. Numer. Algebra Control Optim. **2**, 839–853 (2012)
54. 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. Algorithms **69**, 553–567 (2015)
55. Zhu, M.-Z., Zhang, G.-F.: A class of iteration methods based on HSS for Topelitz systems of weakly nonlinear equations. Journal of Computational and Applied Mathematics **290**, 433–444 (2015)