Computational and Applied Mathematics (2024) 43:219
https://doi.org/10.1007/s40314-024-02691-w

Modified Newton–CAPRESB method for solving a class of systems of nonlinear equations with complex symmetric Jacobian matrices

Jialong Chen[1] · Xiaohui Yu[1] · Qingbiao Wu[1]

Received: 15 November 2023 / Revised: 5 March 2024 / Accepted: 6 March 2024

Abstract
In this paper, we contemplate addressing nonlinear problems involving complex symmetric Jacobian matrices. Firstly, we establish a parameter-free method called modified Newton–CAPRESB (MN–CAPRESB) method by harnessing the modified Newton method as the outer iteration and the CAPRESB (Chebyshev accelerated preconditioned square block) method as the inner iteration. Secondly, the local and semilocal convergence theorems of MN–CAPRESB method are proved under some conditions. Eventually, the numerical experiments of two kinds of complex nonlinear equations are presented to validate the feasibility of MN–CAPRESB method compared to other existing iteration methods.

Keywords Complex nonlinear system · Preconditioned square block (PRESB) method · Chebyshev acceleration · Modified Newton method · Convergence analysis

Mathematics Subject Classification 65H10

1 Introduction

In many fields of physics and engineering, we often encounter nonlinear problems. Examples of such systems can be seen in Ortega and Rheinboldt (2000), Rheinboldt (1998) and references therein.

We contemplate addressing the large and sparse nonlinear system:

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

✉ Qingbiao Wu
qbwu@zju.edu.cn

Jialong Chen
jlcsy@zju.edu.cn

Xiaohui Yu
xiaohyu@zju.edu.cn

1 School of Mathematical Sciences, Zhejiang University, Hangzhou 310027, People's Republic of China

Published online: 03 May 2024

Here,

$$F : \mathbb{D} \subset \mathbb{C}^n \to \mathbb{C}^n$$

represents a continuously differentiable function.

We assume that the Jacobian matrix $F'(x)$ has the following form:

$$F'(x) = W(x) + iT(x), \tag{2}$$

where $i = \sqrt{-1}$ and matrices $W(x), T(x) \in \mathbb{R}^{n\times n}$ are symmetric positive definite (SPD) and symmetric positive semidefinite (SPSD), respectively.

For solving Eq. (1), many researchers have devoted themselves to proposing numerous methods (Ortega and Rheinboldt 2000; Rheinboldt 1998; King 1973). The Newton method is considered as a efficient and widely used iterative method for addressing nonlinear problems. The inexact Newton method (Dembo et al. 1982) outperforms the Newton method. It could be formulated as follows:

$$F'(x_k)m_k = -F(x_k) + r_k, \text{ with } x_{k+1} := x_k + m_k.$$

Here, $x_0 \in \mathbb{D}$ stands for the provided initial vector, while r_k denotes a residual acquired during the inner iteration process. The modified Newton method (Darvishi and Barati 2007) was established to reduce the computational load in Newton method, which has the following form:

$$\begin{cases} y_k = x_k - F'(x_k)^{-1}F(x_k), \\ x_{k+1} = y_k - F'(x_k)^{-1}F(y_k). \end{cases}$$

Among the above methods, we can observe that tackling linear equations is an integral part in these methods. So we review the methods for addressing complex linear equation

$$A\tilde{f} = b, \quad A \in \mathbb{C}^{n\times n}, \quad \tilde{f}, b \in \mathbb{C}^n, \tag{3}$$

where $A = W + iT$ and the matrices $W, T \in \mathbb{R}^{n\times n}$ are SPD and SPSD, respectively. In 2003, the Hermitian and skew-Hermitian splitting (HSS) method was constructed by Bai et al., which was based on the specific structure of A (Bai et al. 2003). The modified HSS (MHSS) (Bai et al. 2010) and preconditioned MHSS (PMHSS) (Bai et al. 2011) methods have also been established for enhancing the HSS method. Moreover, Hezari et al. established a single-step iterative method known as the SCSP method in Hezari et al. (2016). Subsequently, other researchers have extended and improved upon this method, leading to the development of more effective iteration methods (Zheng et al. 2017; Salkuyeh and Siahkolaei 2018). Furthermore, Wang et al. presented a novel iteration method combining real and imaginary parts (CRI) (Wang et al. 2017). Since then, a number of researchers have devoted their efforts to developing efficient iteration methods (Xiao and Wang 2018; Huang 2021; Shirilord and Dehghan 2022) to solve Eq. (3).

Additionally, it can be confirmed that Eq. (3) is equivalent to the following equation:

$$\mathcal{A}f \equiv \begin{bmatrix} W & -T \\ T & W \end{bmatrix} \begin{bmatrix} z^{(1)} \\ z^{(2)} \end{bmatrix} = \begin{bmatrix} p \\ q \end{bmatrix} \equiv c. \tag{4}$$

Here, $\tilde{f} = z^{(1)} + iz^{(2)}$ and $b = p + iq$. To address Eq. (4), a block PMHSS iteration method (Bai et al. 2013) and the DSS real-valued iteration method (Zhang et al. 2019) were proposed, which can be seen as the variants of PMHSS method (Bai et al. 2011) and DSS method (Zheng et al. 2017). In recent years, several iteration methods akin to the generalized successive overrelaxation (GSOR) approach have also been developed (Salkuyeh et al. 2015;

Hezari et al. 2015; Edalatpour et al. 2015; Liang and Zhang 2016). Moreover, Axelsson et al. presented the preconditioned square block (PRESB) preconditioner in Axelsson et al. (2014):

$$\mathcal{P}_{\text{PRESB}} = \begin{bmatrix} W & -T \\ T & \alpha^2 W + 2\alpha T \end{bmatrix},$$

where α is a real positive parameter.

The previously mentioned methods are effective in solving linear systems, but require the selection of a positive parameter or even two parameters, which can be time-consuming during practical implementation. For addressing this issue, a number of researchers provided theoretical optimal parameters in their methods (Bai et al. 2003, 2010, 2011; Hezari et al. 2016; Zheng et al. 2017) while the process of computing optimal parameters often involves the calculation of the eigenvalues of matrices, costing relatively much time. To avoid selecting parameters, Liang and Zhang (2021) proposed a parameter-free method called the Chebyshev accelerated PRESB (CAPRESB) method. They took advantage of the fact that the eigenvalues of $\mathcal{P}_{\text{PRESB}}^{-1}\mathcal{A}$ are located in the interval [0.5, 1] when $\alpha = 1$, making the PRESB method comparably efficient without the parameter selection.

By incorporating the Newton method with the HSS method, Bai et al. established the Newton–HSS method to address complex nonlinear systems in Bai and Guo (2010) and it was improved by Wu et al. who established the modified Newton–HSS method (Wu and Chen 2013). Examples of such methods can be seen in Zhang et al. (2021, 2022), Yu and Wu (2022), Xiao et al. (2021), Dai et al. (2018), Feng and Wu (2021). In this paper, drawing inspiration from these ideas, we establish a parameter-free method called MN–CAPRESB method by incorporating the modified Newton method with CAPRESB method.

Throughout the full paper, we utilize the notation $\|\cdot\|$ to represent the Euclidean norm of a vector or a matrix. If there is no specific explanation, the matrices $W, T \in \mathbb{R}^{n\times n}$ are SPD and SPSD, respectively. Moreover, $\Re(\cdot)$ and $\Im(\cdot)$ are used to represent the real and complex components of the corresponding number, respectively.

The structure of the paper is outlined as follows. In Sect. 2, we present the MN–CAPRESB method. In Sects. 3 and 4, the local and semilocal convergence theorems of our approach are demonstrated under proper conditions, respectively. In Sect. 5, we provide numerical results to illustrate the benefits of our approach compared to the MN–MHSS (Yang and Wu 2012), MN–PMHSS (Zhong et al. 2015), MN–SSTS (Yu and Wu 2022) and MN–PSBTS (Zhang et al. 2022) methods. Eventually, we provide a concise conclusion in Sect. 6.

2 MN–CAPRESB method

First of all, we review the CAPRESB method (Liang and Zhang 2021). For simplicity of notation, we denote

$$\mathcal{P} = \begin{bmatrix} W & -T \\ T & W + 2T \end{bmatrix}.$$

$\mathcal{P}$ is the special case of $\mathcal{P}_{\text{PRESB}}$ when $\alpha = 1$. The details of CAPRESB method can be described as follows:

$$\mathcal{P}u_0 = r_0, \quad f_1 = f_0 + \frac{\tau_0}{2}u_0 \tag{5}$$

and

$$\mathcal{P}u_k = r_k, \quad f_{k+1} = \zeta_k f_k + (1-\zeta_k)f_{k-1} + \tau_k u_k, \quad k = 1, 2, \ldots, \tag{6}$$

where $r_k = c - \mathcal{A} f_k$, τ_k and ζ_k are positive acceleration parameters with

$$\tau_0 = \frac{4}{\tilde{\lambda}_{\max} + \tilde{\lambda}_{\min}}$$

and

$$\tau_k = \left(\frac{\tilde{\lambda}_{\max} + \tilde{\lambda}_{\min}}{2} - \left(\frac{\tilde{\lambda}_{\max} - \tilde{\lambda}_{\min}}{4} \right)^2 \tau_{k-1} \right)^{-1}, \quad \zeta_k = \frac{\tilde{\lambda}_{\max} + \tilde{\lambda}_{\min}}{2} \tau_k, \quad k = 1, 2, \ldots$$

with $\tilde{\lambda}_{\max} = 1$ and $\tilde{\lambda}_{\min} = 0.5$. The iteration formulas (5)-(6) are categorized to the Chebyshev acceleration method (Golub and Varga 1961). Moreover, for the CAPRESB method, we can obtain the Algorithms 1 and 2 in Liang and Zhang (2021).

Algorithm 1 Liang and Zhang (2021) Computation of u from $\mathcal{P}u = r$ with $u = [u^{(1)^T}, u^{(2)^T}]^T$ and $r = [r^{(1)^T}, r^{(2)^T}]^T$

1: Address $(W + T)h = r^{(1)} + r^{(2)}$.
2: Set $g = r^{(2)} - Th$.
3: Address $(W + T)u^{(2)} = g$.
4: Set $u^{(1)} = h - u^{(2)}$.

Algorithm 2 Liang and Zhang (2021) CAPRESB iteration method for (4)

1: Choose f_0, take $\tilde{\lambda}_{\min} = 0.5$ and $\tilde{\lambda}_{\max} = 1$, compute $r_0 = c - \mathcal{A} f_0$.
2: Solve $\mathcal{P}u_0 = r_0$ by Algorithm 1.
3: Set $\tau_0 = \frac{4}{\tilde{\lambda}_{\max} + \tilde{\lambda}_{\min}}$.
4: Compute $f_1 = f_0 + \frac{\tau_0}{2} u_0$.
5: **for** $k = 1, 2, \ldots$, until convergence **do**
6: Compute $r_k = c - \mathcal{A} f_k$.
7: Solve $\mathcal{P}u_k = r_k$ by Algorithm 1.
8: Compute $\tau_k = \left(\frac{\tilde{\lambda}_{\max} + \tilde{\lambda}_{\min}}{2} - \left(\frac{\tilde{\lambda}_{\max} - \tilde{\lambda}_{\min}}{4} \right)^2 \tau_{k-1} \right)^{-1}$, $\zeta_k = \frac{\tilde{\lambda}_{\max} + \tilde{\lambda}_{\min}}{2} \tau_k$.
9: Compute $f_{k+1} = \zeta_k f_k + (1 - \zeta_k) f_{k-1} + \tau_k u_k$.
10: **end for**

From Algorithms 1 and 2, we know that the CAPRESB method involves two linear systems about $W + T$ at each iteration step, which can be accurately solved through sparse Cholesky factorization or approximately through a preconditioned conjugate gradient (PCG) approach.

Actually, another parameter-free method called CADSS method was also proposed in Liang and Zhang (2021), which accelerates the double-step scale splitting real-valued method (Zhang et al. 2019) with Chebyshev accelerated iteration method. Similarly, we can also establish MN–CADSS method. However, at each step of CADSS method it involves four linear systems to be solved. Based on it, we think that the CAPRESB method outperforms the CADSS method. Hence, we only establish the modified Newton–CAPRESB method.

Now, we summarize the convergence characteristics of the CAPRESB method in Liang and Zhang (2021). From Axelsson (1996), the iterative error $e_k = f_k - f_*$ satisfies

$$e_k = Q_k(\mathcal{P}^{-1}\mathcal{A})e_0 \quad \text{with} \quad Q_k(z) = \frac{T_k\left(\frac{\tilde{\lambda}_{\max}+\tilde{\lambda}_{\min}-2z}{\tilde{\lambda}_{\max}-\tilde{\lambda}_{\min}}\right)}{T_k\left(\frac{\tilde{\lambda}_{\max}+\tilde{\lambda}_{\min}}{\tilde{\lambda}_{\max}-\tilde{\lambda}_{\min}}\right)}, \quad k = 0, 1, 2, \ldots, \tag{7}$$

where f_* represents the exact solution of (4) and

$$T_k(z) = \frac{1}{2}\left[\left(z + \sqrt{z^2-1}\right)^k + \left(z - \sqrt{z^2-1}\right)^k\right], \quad -\infty < z < +\infty,$$

which conforms to the subsequent recurrence equations:

$$T_0(z) = 1, \quad T_1(z) = z \quad \text{and} \quad T_{k+1}(z) = 2zT_k(z) - T_{k-1}(z), \quad k = 1, 2, \ldots.$$

Consequently, we have

$$\max_{\lambda_{\min}\le\lambda\le\lambda_{\max}} |Q_k(\lambda)| \le \max_{\tilde{\lambda}_{\min}\le\lambda\le\tilde{\lambda}_{\max}} |Q_k(\lambda)| = \frac{1}{|T_k\left(\frac{\tilde{\lambda}_{\max}+\tilde{\lambda}_{\min}}{\tilde{\lambda}_{\max}-\tilde{\lambda}_{\min}}\right)|} \le 2\left(\frac{\sqrt{2}-1}{\sqrt{2}+1}\right)^k, \tag{8}$$

where $\lambda_{\max}$ and $\lambda_{\min}$ represent the largest and smallest eigenvalues of the matrix $\mathcal{P}^{-1}\mathcal{A}$, respectively, satisfying

$$\tilde{\lambda}_{\min} \le \lambda_{\min} \le \lambda_{\max} \le \tilde{\lambda}_{\max}.$$

If $\mathcal{P}^{-1}\mathcal{A}$ is symmetric, it is straightforward to derive the convergence characteristics according to Eqs. (7) and (8). Moreover, the authors in Liang and Zhang (2021) have proved that $\mathcal{P}^{-1}\mathcal{A}$ can be diagonalized, which contributes to the convergence characteristics of the CAPRESB method. Now, we summarize the convergence characteristics of the CAPRESB method in Liang and Zhang (2021).

Theorem 2.1 *We denote* $e_k = f_k - f_*$ *and* $\tilde{e}_k = \tilde{f}_k - \tilde{f}_*$, $k = 0, 1, \ldots$. *Here,* $f_k = \begin{bmatrix} z_k^{(1)} \\ z_k^{(2)} \end{bmatrix}$, $f_* = \begin{bmatrix} z_*^{(1)} \\ z_*^{(2)} \end{bmatrix}$, $\tilde{f}_k = z_k^{(1)} + iz_k^{(2)}$, $\tilde{f}_* = z_*^{(1)} + iz_*^{(2)}$, *and* f_* *and* $\tilde{f}_*$ *are the exact solutions of* (4) *and* (3), *respectively. Then we have*

$$\|e_k\| \le \sqrt{\kappa_2(H)}\kappa_\mu \theta^k \|e_0\|$$

and

$$\|\tilde{e}_k\| \le \sqrt{\kappa_2(H)}\kappa_\mu \theta^k \|\tilde{e}_0\|,$$

$k = 0, 1, \ldots$, *where* $H = W + T$, $\kappa_2(H) = \|H\|\|H^{-1}\|$, $\kappa_\mu = 2 + \mu_{\max}^2 + \mu_{\max}\sqrt{4+\mu_{\max}^2}$, $\theta = \frac{\sqrt{2}-1}{\sqrt{2}+1}$, *and* $\mu_{\max}$ *represents the largest eigenvalue of the matrix* $W^{-1}T$.

Proof From Theorem 3.1 in Liang and Zhang (2021), we know that there exist inverse matrix $\tilde{X}$ and diagonal matrix D such that

$$\mathcal{P}^{-1}\mathcal{A} = \tilde{X}D\tilde{X}^{-1},$$

where $\kappa_2(\tilde{X}) \leq \frac{\sqrt{\kappa_2(H)\kappa_\mu}}{2}$. According to the Eqs. (7) and (8), we can obtain

$$\begin{aligned}\|e_k\| &= \|Q_k(\mathcal{P}^{-1}\mathcal{A})e_0\| = \|Q_k(\tilde{X}D\tilde{X}^{-1})e_0\| \\ &= \|\tilde{X}Q_k(D)\tilde{X}^{-1}e_0\| \leq \|\tilde{X}\|\|Q_k(D)\|\|\tilde{X}^{-1}\|\|e_0\| \\ &= \kappa_2(\tilde{X})\|Q_k(D)\|\|e_0\| \leq \sqrt{\kappa_2(H)\kappa_\mu}\theta^k\|e_0\|,\end{aligned}$$

$k = 0, 1, \ldots$. It is trivial to see $\|\tilde{e}_k\| = \|e_k\|$ and $\|\tilde{e}_*\| = \|e_*\|$. Hence, the other formula holds. □

Moreover, we propose the following theorem to further describe the convergence characteristics.

Theorem 2.2 *In accordance with the conditions stated in Theorem 2.1, we can hold*

$$\|e_k\| \leq \sqrt{M}\left(2 + M^2 + M\sqrt{4 + M^2}\right)\theta^k\|e_0\| \tag{9}$$

and

$$\|\tilde{e}_k\| \leq \sqrt{M}\left(2 + M^2 + M\sqrt{4 + M^2}\right)\theta^k\|\tilde{e}_0\|, \tag{10}$$

$k = 0, 1, \ldots$, *where* $M = \|W^{-1}\|(\|W\| + \|T\|)$ *and* $\theta = \frac{\sqrt{2}-1}{\sqrt{2}+1}$.

Proof Actually, we can obtain

$$\kappa_2(H) \leq \frac{\lambda_{\max}(W) + \lambda_{\max}(T)}{\lambda_{\min}(W)} = M$$

and

$$\mu_{\max} \leq \|W^{-1}\|\|T\| \leq M.$$

Hence, from Theorem 2.1, it holds that

$$\|e_k\| \leq \sqrt{\kappa_2(H)\kappa_\mu}\theta^k\|e_0\| \leq \sqrt{M}\left(2 + M^2 + M\sqrt{4 + M^2}\right)\theta^k\|e_0\|,$$

$k = 0, 1, \ldots$. Moreover, it is straightforward to know the other formula holds. □

The CAPRESB method is converged when $W, T \in \mathbb{R}^{n\times n}$ are both SPSD and $\text{null}(W) \cap \text{null}(T) = 0$, where $\text{null}(A)$ denotes the null space of A (Liang and Zhang 2021). We restrict W to be SPD to better estimate $\kappa_2(H)$ and $\mu_{\max}$ as Theorem 2.2 shows.

Then, we propose the MN–CAPRESB method, which implies that we utilize CAPRESB method for the following equations:

$$\begin{cases} F'(x_k)d_k = -F(x_k), & y_k = x_k + d_k, \\ F'(x_k)h_k = -F(y_k), & x_{k+1} = y_k + h_k. \end{cases} \tag{11}$$

Actually, the MN–CAPRESB method for addressing Eq. (1) can be derived as demonstrated in Algorithm 3. We make the following markings to better illustrate the algorithm. $\mathcal{P}(x)$ and $\mathcal{A}(x)$ denote the special case of $\mathcal{P}$ and $\mathcal{A}$ when $W = W(x)$ and $T = T(x)$, respectively. Moreover, $c(x) \equiv \begin{bmatrix} \Re(-F(x)) \\ \Im(-F(x)) \end{bmatrix}$.

Algorithm 3 MN–CAPRESB method

Given an initial vector x_0, a positive number tol as well as two sequences of positive integers $\{l_k\}_{k=0}^{\infty}$ and $\{m_k\}_{k=0}^{\infty}$.
for $k = 0, 1, \ldots,$ until $\|F(x_k)\| \leq tol\|F(x_0)\|$ **do**
  Assign $d_{k,0} = h_{k,0} = 0$.
  For $l = 0, 1, \ldots, l_k - 1$, implement Algorithm 2 for the first formula in (11), where $f_0 = 0, \mathcal{P} = \mathcal{P}(x_k)$, $\mathcal{A} = \mathcal{A}(x_k)$ and $c = c(x_k)$. Obtain $\begin{bmatrix} z_{k,l_k}^{(1)} \\ z_{k,l_k}^{(2)} \end{bmatrix}$ such that

$$\|F(x_k) + F'(x_k)d_{k,l_k}\| \leq \eta_k \|F(x_k)\| \text{ with } 0 \leq \eta_k < 1, \tag{12}$$

  where $d_{k,l_k} = z_{k,l_k}^{(1)} + iz_{k,l_k}^{(2)}$.
  Assign $y_k := x_k + d_{k,l_k}$.
  Calculate $F(y_k)$.
  For $m = 0, 1, \ldots, m_k - 1$, implement Algorithm 2 to the second formula in (11), where $f_0 = 0$, $\mathcal{P} = \mathcal{P}(x_k)$, $\mathcal{A} = \mathcal{A}(x_k)$ and $c = c(y_k)$. Obtain $\begin{bmatrix} \tilde{z}_{k,m_k}^{(1)} \\ \tilde{z}_{k,m_k}^{(2)} \end{bmatrix}$ such that

$$\|F(y_k) + F'(x_k)h_{k,m_k}\| \leq \tilde{\eta}_k \|F(y_k)\| \text{ with } 0 \leq \tilde{\eta}_k < 1, \tag{13}$$

  where $h_{k,m_k} = \tilde{z}_{k,m_k}^{(1)} + i\tilde{z}_{k,m_k}^{(2)}$.
  Assign $x_{k+1} := y_k + h_{k,m_k}$.
end for

3 Local convergence theorem of the MN–CAPRESB method

Suppose that $x_* \in \mathbb{N}_0 \subseteq \mathbb{D}$ with $F(x_*) = 0$, and $\mathbb{N}(x_*, r)$ represents an open ball centered at x_* with radius r.

Assumption 3.1 For any $x \in \mathbb{N}(x_*, r) \subset \mathbb{N}_0$, we assume that the formulas below are satisfied.
(THE BOUNDED CONDITION) There exist numbers $\beta, \gamma > 0$ such that

$$\max\{\|W(x_*)\|, \|T(x_*)\|\} \leq \beta, \quad \max\{\|W(x_*)^{-1}\|, \|F'(x_*)^{-1}\|\} \leq \gamma.$$

(THE LIPSCHITZ CONDITION) There exist numbers $L_w, L_t \geq 0$ such that

$$\|W(x) - W(x_*)\| \leq L_w\|x - x_*\|, \quad \|T(x) - T(x_*)\| \leq L_t\|x - x_*\|.$$

Lemma 3.1 *Under the condition that* $r \in (0, \frac{1}{\gamma L})$ *and subject to Assumption 3.1,* $W(x)^{-1}$ *and* $F'(x)^{-1}$ *exist for all* $x \in \mathbb{N}(x_*, r) \subset \mathbb{N}_0$. *Additionally, the formulas below are satisfied for any* $x, z \in \mathbb{N}(x_*, r)$ *with* $L := L_w + L_t$*:*

$$\|F'(x) - F'(x_*)\| \leq L\|x - x_*\|,$$

$$\max\{\|W(x)^{-1}\|, \|F'(x)^{-1}\|\} \leq \frac{\gamma}{1 - \gamma L\|x - x_*\|},$$

$$\|F(z)\| \leq \frac{L}{2}\|z - x_*\|^2 + 2\beta\|z - x_*\|,$$

$$\|z - x_* - F'(x)^{-1}F(z)\| \leq \frac{\gamma}{1 - \gamma L\|x - x_*\|}\left(\frac{L}{2}\|z - x_*\| + L\|x - x_*\|\right)\|z - x_*\|.$$

Proof The proof closely resembles that of Lemma 3.3 in Xie et al. (2020), so we omit it here. □

Lemma 3.2 *In accordance with the conditions stated in Lemma 3.1, let* $r \in (0, r_0)$, $r_0 := \min\{r_1, r_2\}$ *and* $u := \min\{l_*, m_*\}$, *where*

$$r_1 = \frac{\beta}{L(1+3\gamma\beta)},$$
$$r_2 = \frac{2(1-2\tau\beta\gamma\theta^u)}{(5+\tau)\gamma L},$$

$l_* = \liminf_{k\to\infty} l_k$ *and* $m_* = \liminf_{k\to\infty} m_k$. *Moreover, the number u is subjected to*

$$u > \left\lfloor -\frac{\ln(2\tau\beta\gamma)}{\ln\theta} \right\rfloor,$$

where the notation $\lfloor\cdot\rfloor$ *represents the smallest integer at least as much as the corresponding real number. Additionally, we set*

$$g(t; v) = \frac{\gamma}{1-\gamma L t}\left[\left(\frac{3+\tau}{2}\right) L t + 2\tau\beta\theta^v\right].$$

Subsequently, for all $x_k \in \mathbb{N}(x_*, r) \subset \mathbb{N}_0$, *we have*

$$\|d_{k,l_k} - d_k\| \le \tau\theta^u \|d_k\|$$

and

$$\|h_{k,m_k} - h_k\| \le \tau\theta^u \|h_k\|.$$

Here, d_{k,l_k} *and* h_{k,m_k} *are obtained by Algorithm 3,* $d_k = -F'(x_k)^{-1}F(x_k)$ *and* $h_k = -F'(x_k)^{-1}F(y_k)$. *Moreover,* $\theta = \frac{\sqrt{2}-1}{\sqrt{2}+1}$ *and* $\tau = \sqrt{\tilde{M}}\left(2 + \tilde{M}^2 + \tilde{M}\sqrt{4+\tilde{M}^2}\right)$ *with* $\tilde{M} = 3\gamma\beta$. *When* $t \in (0, r)$ *and* $v > u$, *we can obtain that*

$$g(t; v) < g(r_0; u) < 1.$$

Proof Firstly, since $r < \frac{1}{\gamma L}$, for all $x_k \in \mathbb{N}(x_*, r)$, it holds that

$$\|W(x_k)\| \le \|W(x_k) - W(x_*)\| + \|W(x_*)\| \le L_w\|x_k - x_*\| + \beta$$

and

$$\|T(x_k)\| \le \|T(x_k) - T(x_*)\| + \|T(x_*)\| \le L_t\|x_k - x_*\| + \beta.$$

Moreover, in accordance with Lemma 3.1, since $r < r_1$,

$$\|W(x_k)^{-1}\|(\|W(x_k)\| + \|T(x_k)\|) \le \frac{\gamma(L\|x_k - x_*\| + 2\beta)}{1-\gamma L\|x_k - x_*\|} \le 3\gamma\beta.$$

According to the formula (10) in Theorem 2.2, we have

$$\begin{aligned}\|d_{k,l_k} - d_k\| &\le \sqrt{M_k}\left(2 + M_k^2 + M_k\sqrt{4+M_k^2}\right)\theta^{l_k}\|d_{k,0} - d_k\| \\ &\le \tau\theta^{l_k}\|d_k\| \le \tau\theta^u\|d_k\|\end{aligned}$$

and

$$\begin{aligned}\|h_{k,m_k} - h_k\| &\le \sqrt{M_k}\left(2 + M_k^2 + M_k\sqrt{4+M_k^2}\right)\theta^{m_k}\|h_{k,0} - h_k\| \\ &\le \tau\theta^{m_k}\|h_k\| \le \tau\theta^u\|h_k\|,\end{aligned}$$

where $M_k = \|W(x_k)^{-1}\|(\|W(x_k)\| + \|T(x_k)\|)$. Furthermore, since $0 < t < r$, $r < r_2$ and $v > u$, we have

$$g(t; v) = \frac{\gamma}{1-\gamma L t}\left[\left(\frac{3+\tau}{2}\right)Lt + 2\tau\beta\theta^{v}\right] < g(r_0; u) < 1.$$

□

Theorem 3.1 *In accordance with the conditions stated in Lemmas 3.1 and 3.2, for all $x_0 \in \mathbb{N}(x_*, r)$ and any positive integers sequences $\{l_k\}_{k=0}^{\infty}$, $\{m_k\}_{k=0}^{\infty}$, the sequence $\{x_k\}_{k=0}^{\infty}$ generated by the Algorithm 3 converges to x_*. Furthermore, it holds that*

$$\limsup_{k\to\infty} \|x_k - x_*\|^{\frac{1}{k}} \le g(r_0; u)^2.$$

Proof Firstly, according to Lemmas 3.1, 3.2 and formula (11), if $x_k \in \mathbb{N}(x_*, r) \subset \mathbb{N}_0$, we can obviously obtain

$$\begin{aligned}
\|y_k - x_*\| &= \|x_k - x_* + d_{k,l_k}\| \\
&\le \|x_k - x_* - F'(x_k)^{-1}F(x_k)\| + \|d_{k,l_k} - d_k\| \\
&\le \frac{\gamma}{1-\gamma L\|x_k - x_*\|}\frac{3L}{2}\|x_k - x_*\|^2 \\
&\quad + \frac{\gamma\tau\theta^{l_k}}{1-\gamma L\|x_k - x_*\|}\left(\frac{L}{2}\|x_k - x_*\|^2 + 2\beta\|x_k - x_*\|\right) \\
&= \frac{\gamma}{1-\gamma L\|x_k - x_*\|}\left(\frac{3+\tau\theta^{l_k}}{2}L\|x_k - x_*\| + 2\tau\beta\theta^{l_k}\right)\|x_k - x_*\| \\
&\le \frac{\gamma}{1-\gamma L\|x_k - x_*\|}\left(\frac{3+\tau}{2}L\|x_k - x_*\| + 2\tau\beta\theta^{l_k}\right)\|x_k - x_*\| \\
&= g(\|x_k - x_*\|; l_k)\|x_k - x_*\| \\
&\le g(r_0; u)\|x_k - x_*\| \\
&\le \|x_k - x_*\|
\end{aligned}$$

and

$$\begin{aligned}
\|x_{k+1} - x_*\| &= \|y_k - x_* + h_{k,m_k}\| \\
&\le \|y_k - x_* - F'(x_k)^{-1}F(y_k)\| + \|h_{k,m_k} - h_k\| \\
&\le \frac{\gamma}{1-\gamma L\|x_k - x_*\|}\left(\frac{L}{2}\|y_k - x_*\| + L\|x_k - x_*\|\right)\|y_k - x_*\| \\
&\quad + \frac{\gamma\tau\theta^{m_k}}{1-\gamma L\|x_k - x_*\|}\left(\frac{L}{2}\|y_k - x_*\|^2 + 2\beta\|y_k - x_*\|\right) \\
&= \frac{\gamma}{1-\gamma L\|x_k - x_*\|}\left(\frac{1+\tau\theta^{m_k}}{2}L\|y_k - x_*\|\right. \\
&\quad \left. + L\|x_k - x_*\| + 2\tau\beta\theta^{m_k}\right)\|y_k - x_*\| \\
&\le \frac{\gamma g(\|x_k - x_*\|; l_k)}{1-\gamma L\|x_k - x_*\|}\left(\frac{1+\tau\theta^{m_k}}{2}Lg(\|x_k - x_*\|; l_k)\|x_k - x_*\|\right. \\
&\quad \left. + L\|x_k - x_*\| + 2\tau\beta\theta^{m_k}\right)\|x_k - x_*\|
\end{aligned}$$

$$
\begin{aligned}
&\leq \frac{\gamma g(\|x_k - x_*\|; l_k)}{1-\gamma L\|x_k - x_*\|}\left(\frac{3+\tau\theta^{m_k}}{2}L\|x_k - x_*\| + 2\tau\beta\theta^{m_k}\right)\|x_k - x_*\| \\
&\leq g(\|x_k - x_*\|; l_k) g(\|x_k - x_*\|; m_k)\|x_k - x_*\| \\
&\leq g(\|x_k - x_*\|; u)^2\|x_k - x_*\| \\
&\leq g(r_0; u)^2\|x_k - x_*\| \\
&\leq \|x_k - x_*\|.
\end{aligned}
$$

Based on the above formulas, we can use mathematical induction to prove that $\{x_k\}_{k=0}^{\infty} \subset \mathbb{N}(x_*, r)$ converges to x_*. Actually, we have $\|x_0 - x_*\| < r$. Moreover, it holds

$$
\|x_1 - x_*\| \leq g(\|x_0 - x_*\|; u)^2\|x_0 - x_*\| \leq \|x_0 - x_*\| < r.
$$

Therefore, we have $\|x_1 - x_*\| < r$. Furthermore, assume that $x_n \in \mathbb{N}(x_*, r)$, then it holds that

$$
\begin{aligned}
\|x_{n+1} - x_*\| &\leq g(\|x_n - x_*\|; u)^2\|x_n - x_*\| \\
&\leq g(r_0; u)^{(2n+2)}\|x_0 - x_*\| < r,
\end{aligned}
$$

which implies that $x_{n+1} \in \mathbb{N}(x_*, r)$ and

$$
\limsup_{k\to\infty} \|x_k - x_*\|^{\frac{1}{k}} \leq g(r_0; u)^2.
$$

Additionally, when $n \to \infty$, we get $x_{n+1} \to x_*$. □

4 Semilocal convergence theorem of MN–CAPRESB method

Assumption 4.1 Let $x_0 \in \mathbb{C}^n$ and assume that the formulas below are satisfied.
(THE BOUNDED CONDITION) There exist positive numbers β, γ and δ such that

$$
\max\{\|W(x_0)\|, \|T(x_0)\|\} \leq \beta, \quad \max\{\|W(x_0)^{-1}\|, \|F'(x_0)^{-1}\|\} \leq \gamma, \quad \|F(x_0)\| \leq \delta.
$$

(THE LIPSCHITZ CONDITION) There exist numbers $L_w, L_t \geq 0$ such that for any $x, z \in \mathbb{N}(x_0, r) \subset \mathbb{N}_0$,

$$
\|W(x) - W(z)\| \leq L_w\|x - z\|, \quad \|T(x) - T(z)\| \leq L_t\|x - z\|.
$$

Lemma 4.1 *Subject to Assumption 4.1, for all $x, z \in \mathbb{N}(x_0, r)$, if $r \in (0, \frac{1}{\gamma L})$, we have $W(x)^{-1}$ and $F'(x)^{-1}$ exist. Additionally, we can obtain the following inequalities with $L := L_w + L_t$:*

$$
\begin{aligned}
&\|F'(x) - F'(z)\| \leq L\|x - z\|, \\
&\|F'(x)\| \leq L\|x - x_0\| + 2\beta, \\
&\|F(x) - F(z) - F'(z)(x - z)\| \leq \frac{L}{2}\|x - z\|^2, \\
&\max\{\|W(x)^{-1}\|, \|F'(x)^{-1}\|\} \leq \frac{\gamma}{1 - \gamma L\|x - x_0\|}.
\end{aligned}
$$

Proof The proof resembles the Lemma 3 of Zhang et al. (2021); therefore, we omit it here. □

Let we denote

$$a := L\gamma(1+\eta), \quad b := 1-\eta, \quad c := 2\gamma\delta, \quad \text{where} \quad \eta := \max_k\{\max\{\eta_k, \tilde{\eta}_k\}\} < 1.$$

The sequences of iteration $\{t_k\}$ and $\{s_k\}$ are subject to the formulas below:

$$\begin{cases} t_0 = 0, \\ s_k = t_k - \dfrac{g(t_k)}{h(t_k)}, \\ t_{k+1} = s_k - \dfrac{g(s_k)}{h(t_k)}. \end{cases} \tag{14}$$

Here,

$$\begin{cases} g(x) = \frac{1}{2}ax^2 - bx + c, \\ h(x) = ax - 1. \end{cases}$$

We have the following property about the two sequences.

Lemma 4.2 *Suppose that the numbers are subject to*

$$\gamma^2\delta L \leq \frac{(1-\eta)^2}{4(1+\eta)}.$$

Then the sequences $\{t_k\}$ and $\{s_k\}$ of (14) converge to t_* with $t_* = \frac{b-\sqrt{b^2-2ac}}{a}$. Furthermore,

$$0 \leq t_k < s_k < t_{k+1} < t_*,$$
$$t_{k+1} - s_k < s_k - t_k.$$

Proof The proof can be seen in Lemma 4.2 and Lemma 4.3 of Wu and Chen (2013). □

Theorem 4.1 *In accordance with the conditions stated in Lemmas 4.1 and 4.2, define $r := \min\{r_1, r_2\}$ and $u := \min\{l_*, m_*\}$, where*

$$r_1 = \frac{\beta}{L(1+3\gamma\beta)},$$
$$r_2 = \frac{b - \sqrt{b^2 - 2ac}}{a},$$

$l_* = \liminf_{k\to\infty} l_k$ and $m_* = \liminf_{k\to\infty} m_k$. Moreover, the number u is subject to

$$u > \left\lfloor \frac{\ln\frac{\eta}{\tau}}{\ln\theta} \right\rfloor,$$

where τ and θ are same in Lemma 3.2. Then the sequence $\{x_k\}_{k=0}^{\infty}$ generated by the Algorithm 3 converges to x_*.

Proof In fact, for any $x_k \in \mathbb{N}(x_*, r)$, analysis resembling that of Lemma 3.2 indicates

$$\|d_{k,l_k} - d_k\| \leq \tau\theta^{l_k}\|d_k\|$$

and

$$\|h_{k,m_k} - h_k\| \leq \tau\theta^{m_k}\|h_k\|.$$

We use mathematical induction to prove the following formulas:

$$\begin{cases} \|x_k - x_0\| \le t_k - t_0, \\ \|F(x_k)\| \le \frac{1-\gamma L t_k}{(1+\eta)\gamma}(s_k - t_k), \\ \|y_k - x_k\| \le s_k - t_k, \\ \|F(y_k)\| \le \frac{1-\gamma L t_k}{(1+\eta)\gamma}(t_{k+1} - s_k), \\ \|x_{k+1} - y_k\| \le t_{k+1} - s_k. \end{cases} \tag{15}$$

According to Lemmas 4.1, 4.2 and formula (12), we can obtain

$$\begin{aligned} &\|x_0 - x_0\| = 0 \le t_0 - t_0, \\ &\|F(x_0)\| \le \delta \le \frac{2\gamma\delta}{\gamma(1+\eta)} = \frac{1-\gamma L t_0}{(1+\eta)\gamma}(s_0 - t_0), \\ &\|y_0 - x_0\| = \|d_{0,l_0}\| \le \|d_{0,l_0} - d_0\| + \|d_0\| \\ &\quad \le (1+\tau\theta^{l_0})\|d_0\| \le (1+\tau\theta^{l_0})\gamma\delta \le 2\gamma\delta = s_0 - t_0, \\ &\|F(y_0)\| \le \|F(y_0) - F(x_0) - F'(x_0)(y_0 - x_0)\| + \|F(x_0) + F'(x_0)(y_0 - x_0)\| \\ &\quad \le \frac{L}{2}\|y_0 - x_0\|^2 + \eta\|F(x_0)\| \le \frac{L}{2}s_0^2 + \eta\delta \le \frac{1-\gamma L t_0}{(1+\eta)\gamma}(t_1 - s_0), \\ &\|x_1 - y_0\| \le \|h_{0,m_0}\| \le \|h_{0,m_0} - h_0\| + \|h_0\| \\ &\quad \le (1+\tau\theta^{m_0})\|h_0\| = (1+\tau\theta^{m_0})\|F'(x_0)^{-1}F(y_0)\| < (1+\eta)\gamma\|F(y_0)\| \le t_1 - s_0. \end{aligned}$$

Hence, when $k = 0$, the inequalities (15) are true. Now, assume that when $n \le k - 1$, the inequalities (15) hold. We consider the case $n = k$. For the first inequality in (15), we can obtain

$$\|x_k - x_0\| \le \|x_k - y_{k-1}\| + \|y_{k-1} - x_{k-1}\| + \|x_{k-1} - x_0\| \le t_k - t_0.$$

Since $x_{k-1}, y_{k-1} \in \mathbb{N}(x_0, r)$ and due to the inequalities(13) as well as formulas in Lemma 4.1, it holds that

$$\begin{aligned} (1+\eta)\gamma\|F(x_k)\| &\le (1+\eta)\gamma\|F(x_k) - F(y_{k-1}) - F'(y_{k-1})(x_k - y_{k-1})\| \\ &\quad + (1+\eta)\gamma\|(F'(y_{k-1}) - F'(x_{k-1}))(x_k - y_{k-1})\| \\ &\quad + (1+\eta)\gamma\|F(y_{k-1}) + F'(x_{k-1})(x_k - y_{k-1})\| \\ &\le \frac{(1+\eta)\gamma L}{2}\|x_k - y_{k-1}\|^2 + (1+\eta)\gamma L\|y_{k-1} - x_{k-1}\|\|x_k - y_{k-1}\| \\ &\quad + \eta(1+\eta)\gamma\|F(y_{k-1})\| \\ &\le \frac{a}{2}(t_k^2 - s_{k-1}^2) - a t_{k-1}(t_k - s_{k-1}) + \eta(1-\gamma L t_{k-1})(t_k - s_{k-1}) \\ &= g(t_k) - g(s_{k-1}) + b(t_k - s_{k-1}) - a t_{k-1}(t_k - s_{k-1}) \\ &\quad + \eta(1-\gamma L t_{k-1})(t_k - s_{k-1}) \\ &= g(t_k) - \frac{h(t_{k-1}) + 1 - a t_{k-1} - \eta\gamma L t_{k-1}}{h(t_{k-1})} g(s_{k-1}) \\ &= g(t_k) + \frac{\eta\gamma L t_{k-1}}{h(t_{k-1})} g(s_{k-1}) \\ &< g(t_k) = -h(t_k)(s_k - t_k) < (1-\gamma L t_k)(s_k - t_k). \end{aligned}$$

Therefore, it holds that

$$\|F(x_k)\| \le \frac{(1-\gamma L t_k)}{(1+\eta)\gamma}(s_k - t_k)$$

and then

$$\begin{aligned}\|y_k - x_k\| &\le (1+\tau\theta^{l_k})\|F'(x_k)^{-1}\|\|F(x_k)\| \\ &\le (1+\eta)\frac{\gamma}{1-\gamma L t_k}\|F(x_k)\| \\ &\le s_k - t_k.\end{aligned}$$

Also, by the inequality (12), we can obtain

$$\begin{aligned}(1+\eta)\gamma\|F(y_k)\| &\le (1+\eta)\gamma\|F(y_k) - F(x_k) - F'(x_k)(y_k - x_k)\| \\ &\quad + (1+\eta)\gamma\|F(x_k) + F'(x_k)(y_k - x_k)\| \\ &\le \frac{(1+\eta)\gamma L}{2}\|y_k - x_k\|^2 + \eta(1+\eta)\gamma\|F(x_k)\| \\ &\le \frac{(1+\eta)\gamma L}{2}(s_k - t_k)^2 + \eta(1-\gamma L t_k)(s_k - t_k) \\ &= g(s_k) - g(t_k) + b(s_k - t_k) - a t_k(s_k - t_k) \\ &\quad + \eta(1-\gamma L t_k)(s_k - t_k) \\ &= g(s_k) - g(t_k) - (1 - a t_k - \eta\gamma L t_k)\frac{g(t_k)}{h(t_k)} \\ &= g(s_k) - \frac{h(t_k) + 1 - a t_k - \eta\gamma L t_k}{h(t_k)} g(t_k) \\ &= g(s_k) + \frac{\eta\gamma L t_k}{h(t_k)} g(t_k) \\ &< g(s_k) = -h(t_k)(t_{k+1} - s_k) < (1-\gamma L t_k)(t_{k+1} - s_k).\end{aligned}$$

It follows that

$$\|F(y_k)\| \le \frac{(1-\gamma L t_k)}{(1+\eta)\gamma}(t_{k+1} - s_k)$$

and then

$$\begin{aligned}\|x_{k+1} - y_k\| &\le (1+\tau\theta^{m_k})\|F'(x_k)^{-1}\|\|F(y_k)\| \\ &\le (1+\eta)\frac{\gamma}{1-\gamma L t_k}\|F(y_k)\| \\ &\le t_{k+1} - s_k.\end{aligned}$$

Consequently, for any k, the formulas (15) are true. Since $\{t_k\}$ and $\{s_k\}$ converge to t_*, $\{x_k\}$ and $\{y_k\}$ also converge, to say x_*. Additionally, according to Eq. (7) and

$$y_k = x_k + d_{k,l_k}, \quad k = 0, 1, \ldots,$$

we have

$$\begin{aligned}\liminf_{k\to\infty}\begin{bmatrix}\Re(d_{k,l_k})\\ \Im(d_{k,l_k})\end{bmatrix} &= \liminf_{k\to\infty}(I - Q_{l_k}(\mathcal{P}(x_k)^{-1}\mathcal{A}(x_k)))\begin{bmatrix}\Re(d_k)\\ \Im(d_k)\end{bmatrix} \\ &= (I - Q_{l_*}(\mathcal{P}(x_*)^{-1}\mathcal{A}(x_*)))\begin{bmatrix}\Re(F'(x_*)^{-1}F(x_*))\\ \Im(F'(x_*)^{-1}F(x_*))\end{bmatrix} = 0.\end{aligned}$$

In addition, since the eigenvalues of $Q_{l_*}(\mathcal{P}(x_*)^{-1}\mathcal{A}(x_*))$ are located in $(-1, 1)$ according to Eq. (8), we have $F(x_*) = 0$. □

5 Numerical examples

In this part, we display two nonlinear problems to show the robustness of the MN–CAPRESB method. We compare our method to the MN–MHSS (Yang and Wu 2012), MN–PMHSS (Zhong et al. 2015), MN–SSTS (Yu and Wu 2022) and MN–PSBTS (Zhang et al. 2022) methods. In practical implementation, selecting parameters can be a complex headache. When conducting the experiments, for MN–MHSS and MN–PMHSS methods, to avoid this repetitive and tedious process, we adopt the same experimental ideas as Xie et al. (2020) to directly use the selected parameters, which are determined experimentally by minimizing the respective iteration steps and errors in relation to the precise solution. We use the following termination condition in the outer iteration:

$$\frac{\|F(x_k)\|}{\|F(x_0)\|} \leq 10^{-6}.$$

Moreover, we use Outer IT and Inner IT to represent the number of outer and inner iterations, respectively, and we denote the elapsed CPU time in seconds as "CPU time". The experimental results draw a conclusion that MN–CAPRESB method achieves better results than both MN–MHSS and MN–PMHSS methods without the process of selecting a parameter and exhibits strong competitiveness by virtue of its inherent lack of dependence on parameter selection compared to the MN–SSTS and MN–PSBTS methods.

Example 1 (Yang and Wu 2012) We examine the subsequent nonlinear equations:

$$\begin{cases} v_t - (\alpha_1 + i\beta_1)(v_{x_1x_1} + v_{x_2x_2}) + \rho v = -(\alpha_2 + i\beta_2)v^{\frac{4}{3}}, & \text{in } (0, 1] \times \Omega, \\ v(0, x_1, x_2) = v_0(x_1, x_2), & \text{in } \Omega, \\ v(t, x_1, x_2) = 0, & \text{in } (0, 1] \times \partial\Omega, \end{cases}$$

where $\Omega = [0, 1] \times [0, 1]$, the coefficients $\alpha_1 = \beta_1 = 1, \alpha_2 = \beta_2 = 1$ and $\rho > 0$.

For discretizing this system, we can utilize a similar methodology in Yang and Wu (2012) and get the following equation:

$$F(x) = Mx + (\alpha_2 + i\beta_2)h\Delta t\psi(x) = 0$$

with

$$M = h(1 + \rho\Delta t)I_n + (\alpha_1 + i\beta_1)\frac{\Delta t}{h}(I_N \otimes B_N + B_N \otimes I_N),$$
$$\psi(x) = \left(x_1^{\frac{4}{3}}, x_2^{\frac{4}{3}}, \ldots, x_n^{\frac{4}{3}}\right)^T,$$

where $B_N = \text{tridiag}(-1, 2, -1)$ and $n = N^2$.

During the experiments, the initial vector $x_0 = [1, 1, \ldots, 1]^T$ and the inner tolerance η_k and $\tilde{\eta}_k$ of linear systems have the same value η.

Tables 1 and 2 present the parameters α used in experiments for the MN–MHSS and MN–PMHSS methods, respectively. These optimal parameters are determined based on extensive experiments in Xie et al. (2020). Moreover, Tables 3 and 4 show the two parameters used in

Table 1 The optimal values α in the MN–MHSS method in Example 1 (Xie et al. 2020)

N	$\rho = 1$			$\rho = 10$			$\rho = 200$		
	$\eta = 0.1$	$\eta = 0.2$	$\eta = 0.4$	$\eta = 0.1$	$\eta = 0.2$	$\eta = 0.4$	$\eta = 0.1$	$\eta = 0.2$	$\eta = 0.4$
2^5	0.45	0.46	0.44	0.47	0.48	0.47	0.79	0.74	0.78
2^6	0.28	0.29	0.27	0.28	0.29	0.29	0.45	0.44	0.43
2^7	0.18	0.18	0.18	0.18	0.18	0.18	0.25	0.25	0.25

Table 2 The optimal values α in the MN–PMHSS method in Example 1 (Xie et al. 2020)

N	$\rho = 1$			$\rho = 10$			$\rho = 200$		
	$\eta = 0.1$	$\eta = 0.2$	$\eta = 0.4$	$\eta = 0.1$	$\eta = 0.2$	$\eta = 0.4$	$\eta = 0.1$	$\eta = 0.2$	$\eta = 0.4$
2^5	0.90	0.90	0.90	0.85	0.84	0.84	0.66	0.65	0.50
2^6	0.82	0.81	0.80	0.78	0.77	0.76	0.76	0.70	0.55
2^7	0.66	0.70	0.70	0.61	0.67	0.68	0.80	0.72	0.55

the experiments for the MN–SSTS and MN–PSBTS methods. It is worth emphasizing that the main advantage of the MN–CAPRESB method is its parameter-free nature, contributing to the effectiveness and convenience of the MN–CAPRESB method in practical implementation.

In Tables 5, 6, 7, 8 and 9, we display the experimental results regarding these five approaches associated with the problem size $N = 2^5$ with inner tolerance $\eta = 0.1, 0.2, 0.4$, as well as for $N = 2^6$ and $N = 2^7$ with the same inner tolerance value $\eta = 0.4$, respectively. Additionally, the parameter $\rho = 1, 10, 200$, respectively. From these tables, we know that the MN–CAPRESB method almost obtains extraordinary results regarding errors, CPU time, and inner and outer iterations compared to the MN–MHSS and MN–PMHSS methods. By comparing the results of different parameter values, we can observe that the MN–CAPRESB method consistently performs well across different values of ρ, η and N, demonstrating lower errors and fewer iterations compared to the MN–MHSS and MN–PMHSS methods. Additionally, MN–CAPRESB demonstrates heightened competitiveness due to its inherent parameter-free nature in comparison with the MN–SSTS and MN–PSBTS methods. Therefore, the MN–CAPRESB method can be considered as an effective method.

Indeed, from Tables 5, 6 and 7, it is evident that the residuals of the MN–CAPRESB method exhibit identical values despite having the same scales N and parameters ρ, but varying inner tolerances η. As the inner tolerance increases, the numbers of outer iterations steps for MN–MHSS and MN–PMHSS methods significantly increase, which leads to longer runtime, while the MN–CAPRESB method remains constant. Thus, the MN–CAPRESB method is robust and efficient. On the other hand, as the scale of the problem increases, the MN–CAPRESB method has become increasingly advantageous in terms of time compared to the MN–MHSS and MN–PMHSS methods, which has more obvious advantages when dealing with large-scale problems. From Table 9, when $N = 2^7$, the MN–CAPRESB method almost takes half and one-third of the time compared to the MN–PMHSS and MN–MHSS methods, respectively. While the MN–CAPRESB method may not yield as strong results as the MN–SSTS and MN–PSBTS methods, its inherent lack of parameter selection confers a competitive edge.

Table 3 The optimal values in the MN–SSTS and MN–PSBTS methods with $N = 2^5$ in Example 1

Method	$\rho = 1$			$\rho = 10$			$\rho = 200$		
	$\eta = 0.1$	$\eta = 0.2$	$\eta = 0.4$	$\eta = 0.1$	$\eta = 0.2$	$\eta = 0.4$	$\eta = 0.1$	$\eta = 0.2$	$\eta = 0.4$
MN–SSTS	(1,2)	(1,2)	(1,2)	(1,2)	(1,2)	(1,2)	(1.1,4)	(1.1,3.5)	(1,3.5)
MN–PSBTS	(0.9,2)	(0.9,2)	(0.9,2)	(0.9,2)	(0.9,2)	(0.9,2)	(1,3)	(1,3.1)	(1,3.1)

Table 4 The optimal values in the MN–SSTS and MN–PSBTS methods with $\eta = 0.4$ in Example 1

N	MN–SSTS			MN–PSBTS		
	$\rho = 1$	$\rho = 10$	$\rho = 200$	$\rho = 1$	$\rho = 10$	$\rho = 200$
2^6	(1,2.1)	(1,2.1)	(1,2.9)	(1,2)	(1,2.1)	(1,2)
2^7	(1.1,2)	(1.1,2)	(1,3.4)	(0.9,2)	(0.9,2)	(0.7,2)

Table 5 Comparison results for $\eta = 0.1$ and $N = 2^5$ in Example 1

ρ	Method	Error	CPU time (s)	Outer IT	Inner IT
1	MN–MHSS	7.2790E−07	0.0857	3	88
	MN–PMHSS	7.2882E−08	0.0623	3	24
	MN–SSTS	2.6191E−08	0.0466	2	4
	MN–PSBTS	3.3308E−09	0.0403	2	4
	MN–CAPRESB	4.2361E−08	0.0408	3	12
10	MN–MHSS	7.4263E−07	0.0668	3	84
	MN–PMHSS	8.7206E−08	0.0544	3	24
	MN–SSTS	3.6688E−08	0.0232	2	4
	MN–PSBTS	3.5208E−09	0.0263	2	4
	MN–CAPRESB	4.2372E−08	0.0354	3	12
200	MN-MHSS	5.3905E−07	0.0601	3	56
	MN–PMHSS	1.8192E−07	0.0571	3	33
	MN–SSTS	3.5999E−07	0.0255	2	4
	MN–PSBTS	9.9839E−08	0.0252	2	4
	MN–CAPRESB	4.2490E−08	0.0385	3	12

Table 6 Comparison results for $\eta = 0.2$ and $N = 2^5$ in Example 1

ρ	Method	Error	CPU time (s)	Outer IT	Inner IT
1	MN–MHSS	4.9280E−08	0.1125	5	109
	MN–PMHSS	7.4718E−08	0.0629	4	24
	MN–SSTS	2.6191E−08	0.0244	2	4
	MN–PSBTS	3.3308E−09	0.0252	2	4
	MN–CAPRESB	4.2361E−08	0.0373	3	12
10	MN–MHSS	5.0143E−08	0.1094	5	104
	MN–PMHSS	9.3472E−08	0.0618	4	24
	MN–SSTS	3.6688E−08	0.0255	2	4
	MN–PSBTS	3.5208E−09	0.0258	2	4
	MN–CAPRESB	4.2372E−08	0.0350	3	12
200	MN–MHSS	7.1307E−07	0.0726	4	55
	MN–PMHSS	7.0172E−07	0.0653	4	30
	MN–SSTS	2.9263E−07	0.0244	2	4
	MN–PSBTS	9.5113E−08	0.0251	2	4
	MN–CAPRESB	4.2490E−08	0.0360	3	12

Table 7 Comparison results for $\eta = 0.4$ and $N = 2^5$ in Example 1

ρ	Method	Error	CPU time (s)	Outer IT	Inner IT
1	MN–MHSS	9.7372E−07	0.1196	7	86
	MN–PMHSS	7.3461E−08	0.0920	6	24
	MN–SSTS	2.6191E−08	0.0288	2	4
	MN–PSBTS	3.3308E−09	0.0276	2	4
	MN–CAPRESB	4.2361E−08	0.0363	3	12
10	MN-MHSS	9.5369E−07	0.1351	7	82
	MN–PMHSS	9.2401E−08	0.0921	6	24
	MN–SSTS	3.6688E−08	0.0253	2	4
	MN–PSBTS	3.5208E−09	0.0278	2	4
	MN-CAPRESB	4.2372E−08	0.0366	3	12
200	MN–MHSS	8.4941E−07	0.1249	7	54
	MN–PMHSS	8.2641E−07	0.0907	6	30
	MN–SSTS	3.2668E−07	0.0237	2	4
	MN–PSBTS	9.5113E−08	0.0249	2	4
	MN–CAPRESB	4.2490E−08	0.0362	3	12

Table 8 Comparison results for $\eta = 0.4$ and $N = 2^6$ in Example 1

ρ	Method	Error	CPU time (s)	Outer IT	Inner IT
1	MN–MHSS	9.1057E−07	2.8687	7	133
	MN–PMHSS	1.1226E−07	2.3434	6	24
	MN–SSTS	2.0666E−08	0.7293	2	4
	MN–PSBTS	1.0308E−08	0.6748	2	4
	MN–CAPRESB	4.1676E−08	1.0550	3	12
10	MN-MHSS	2.3714E−07	3.3109	8	144
	MN–PMHSS	1.4335E−07	2.3003	6	24
	MN–SSTS	5.1667E−08	0.7877	2	4
	MN–PSBTS	1.3274E−08	0.6696	2	4
	MN–CAPRESB	4.1679E−08	1.0910	3	12
200	MN–MHSS	9.6074E−07	2.8979	7	86
	MN–PMHSS	7.8701E−07	2.3283	6	29
	MN–SSTS	2.1291E−07	0.6811	2	4
	MN–PSBTS	9.6046E−07	0.6722	2	4
	MN–CAPRESB	4.1722E−08	1.0617	3	12

Example 2 (Xie et al. 2020) Considering the complex nonlinear Helmholtz system:

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

with σ_1 and σ_2 being both real coefficient functions. We can wield finite differences to discretize the problem, which contributes to the following nonlinear equation:

$$F(x) = Kx + \phi(x) = 0,$$

Table 9 Comparison results for $\eta = 0.4$ and $N = 2^7$ in Example 1

ρ	Method	Error	CPU time (s)	Outer IT	Inner IT
1	MN–MHSS	2.3578E−07	45.6399	8	222
	MN–PMHSS	2.6894E−07	32.7058	6	24
	MN–SSTS	1.4336E−08	10.0528	2	4
	MN–PSBTS	5.7080E−08	9.7595	2	4
	MN–CAPRESB	4.1492E−08	15.0786	3	12
10	MN–MHSS	2.3230E−07	45.1905	8	218
	MN–PMHSS	3.2313E−07	33.5342	6	24
	MN–SSTS	1.4909E−08	9.6249	2	4
	MN–PSBTS	7.0031E−08	9.6085	2	4
	MN–CAPRESB	4.1493E−08	15.5962	3	12
200	MN–MHSS	1.9379E−07	46.1425	8	162
	MN–PMHSS	9.0820E−07	33.5194	6	28
	MN–SSTS	5.8149E−08	9.5555	2	4
	MN–PSBTS	7.8489E−08	9.4226	2	4
	MN–CAPRESB	4.1500E−08	15.5269	3	12

where

$$K = (M + \sigma_1 I) + i\sigma_2 I$$

and

$$\phi(x) = (e^{x_1}, e^{x_2}, \dots, e^{x_n})^T$$

with

$$M = I \otimes C_N + C_N \otimes I \quad \text{and} \quad C_N = \frac{1}{h^2}\text{tridiag}(-1, 2, -1).$$

During our experiments, we take values of parameters $\sigma_1 = 100$ and $\sigma_2 = 1000$. Tables 10 and 11 show the parameters used in experiments. In addition, the initial guess $x_0 = [0, 0, \dots, 0]^T$, the inner tolerance $\eta = 0.1, 0.2, 0.4$ as well as the problem size $N = 30, 60, 90$, respectively. Also, we use above five iteration methods to solve this nonlinear problem. We can know that MN–PMHSS method is almost as efficient as MN–MHSS method while MN–CAPRESB method outperforms them upon Tables 12, 13 and 14. Actually, MN–CAPRESB method almost takes half of the time and one-third of inner iteration steps compared to MN–MHSS and MN–PMHSS methods. Moreover, MN–CAPRESB method is competitive with MN–SSTS and MN–PSBTS methods from cpu, outer and inner iterations. Consequently, we yield the conclusion that MN–CAPRESB method outperforms MN–MHSS and MN–PMHSS methods and stands as a viable competitor to MN–SSTS and MN–PSBTS methods.

6 Conclusion

In this paper, we establish a parameter-free method called the MN–CAPRESB method for addressing nonlinear complex equations by harnessing the modified Newton method as the

Table 10 The optimal values α in the MN-MHSS and MN-PMHSS methods in Example 2 (Xie et al. 2020)

N	MN–MHSS			MN–PMHSS		
	$\eta = 0.1$	$\eta = 0.2$	$\eta = 0.4$	$\eta = 0.1$	$\eta = 0.2$	$\eta = 0.4$
30	553	557	557	1.81	1.79	1.79
60	775	781	788	1.26	1.28	1.37
90	899	890	907	1.11	1.12	1.15

Table 11 The optimal values in the MN–SSTS and MN–PSBTS methods in Example 2

N	MN-SSTS			MN–PSBTS		
	$\eta = 0.1$	$\eta = 0.2$	$\eta = 0.4$	$\eta = 0.1$	$\eta = 0.2$	$\eta = 0.4$
30	(1.39,0.8)	(1.5,1)	(1.44,1.1)	(2,0.8)	(2,0.85)	(2,0.89)
60	(1.4,1.05)	(1.5,1)	(1.5,1)	(1.8,1)	(1.9,1)	(2,1)
90	(1.45,1)	(1.5,1)	(1.4,1)	(2,1)	(2,1)	(2.1,1)

Table 12 Comparison results for $N = 30$ in Example 2

η	Method	Error	CPU time (s)	Outer IT	Inner IT
0.1	MN-MHSS	4.6502E−07	0.0509	3	30
	MN–PMHSS	4.6503E−07	0.0485	3	30
	MN–SSTS	3.1523E−08	0.0334	3	12
	MN–PSBTS	3.0870E−08	0.0359	3	11
	MN–CAPRESB	9.1572E−08	0.0283	2	12
0.2	MN–MHSS	1.7833E−07	0.0416	4	32
	MN–PMHSS	1.7834E−07	0.0474	4	32
	MN–SSTS	6.6470E−07	0.0261	3	11
	MN–PSBTS	4.6608E−07	0.0268	3	9
	MN–CAPRESB	3.7283E−08	0.0257	3	13
0.4	MN–MHSS	1.7833E−07	0.0666	8	32
	MN–PMHSS	1.7834E−07	0.0844	8	32
	MN–SSTS	2.6853E−07	0.0383	5	11
	MN–PSBTS	1.0885E−07	0.0399	5	10
	MN–CAPRESB	3.7283E−08	0.0238	3	13

outer iteration and the CAPRESB (Chebyshev accelerated preconditioned square block) method as the inner iteration. The local and semilocal convergence theorems of our approach are proved. Moreover, to validate the superiority and robustness of the MN–CAPRESB method, we conduct some experiments on two kinds of nonlinear examples compared to other existing iteration methods. Unlike some existing methods, the MN–CAPRESB method can avoid selecting one parameter or even two parameters. However, our research is not exhaustive. We may harness other efficient outer iteration method to accelerate our method, which may be a interesting topic to further explore.

Table 13 Comparison results for $N = 60$ in Example 2

η	Method	Error	CPU time (s)	Outer IT	Inner IT
0.1	MN–MHSS	4.4020E−07	0.5670	3	31
	MN–PMHSS	6.9827E−07	0.5808	3	30
	MN–SSTS	9.3212E−09	0.5227	3	13
	MN–PSBTS	4.8060E−07	0.5384	3	12
	MN–CAPRESB	9.0249E−08	0.3436	2	12
0.2	MN–MHSS	2.7986E−07	0.7160	4	32
	MN–PMHSS	2.7986E−07	0.7616	4	32
	MN–SSTS	6.8504E−07	0.5178	3	11
	MN–PSBTS	7.2537E−08	0.5316	3	12
	MN–CAPRESB	3.6800E−08	0.5253	3	13
0.4	MN–MHSS	1.7806E−07	1.4331	8	33
	MN–PMHSS	7.0880E−07	1.3319	7	30
	MN–SSTS	6.8504E−07	0.8723	5	11
	MN–PSBTS	2.1456E−07	0.8788	5	10
	MN–CAPRESB	3.6800E−08	0.5275	3	13

Table 14 Comparison results for $N = 90$ in Example 2

η	Method	Error	CPU time (s)	Outer IT	Inner IT
0.1	MN–MHSS	2.8124E−07	2.5655	3	33
	MN–PMHSS	2.8126E−07	2.5802	3	33
	MN–SSTS	9.2014E−08	2.4487	3	12
	MN–PSBTS	1.2224E−08	2.4905	3	13
	MN–CAPRESB	8.9789E−08	1.6216	2	12
0.2	MN–MHSS	4.3020E−07	3.3408	4	32
	MN–PMHSS	4.3023E−07	3.4350	4	32
	MN–SSTS	6.8674E−07	2.4570	3	11
	MN–PSBTS	3.2544E−08	2.4794	3	12
	MN–CAPRESB	3.6624E−08	2.4557	3	13
0.4	MN–MHSS	1.8412E−07	5.9144	7	34
	MN–PMHSS	2.8390E−07	6.1905	7	33
	MN–SSTS	1.2136E−07	4.0904	5	11
	MN–PSBTS	2.7452E−07	4.0454	5	10
	MN–CAPRESB	3.6624E−08	2.4000	3	13

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

Data Availability Data available on request from the authors

Declarations

Conflict of interest The authors declare that they have no conflicts of interest to this work

References

Axelsson O (1996) Iterative solution methods. Cambridge University Press, Cambridge

Axelsson O, Neytcheva M, Ahmad B (2014) A comparison of iterative methods to solve complex valued linear algebraic systems. Numer Algorithms 66:811–841

Bai ZZ, Guo XP (2010) On Newton-HSS methods for systems of nonlinear equations with positive-definite Jacobian matrices. J Comput Math 28:235–260

Bai ZZ, Golub GH, Ng MK (2003) Hermitian and skew-Hermitian splitting methods for non-Hermitian positive definite linear systems. SIAM J Matrix Anal Appl 24(3):603–626

Bai ZZ, Benzi M, Chen F (2010) Modified HSS iteration methods for a class of complex symmetric linear systems. Computing 87(3–4):93–111

Bai ZZ, Benzi M, Chen F (2011) On preconditioned MHSS iteration methods for complex symmetric linear systems. Numer Algorithms 56(2):297–317

Bai ZZ, Benzi M, Chen F et al (2013) 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

Dai PF, Wu QB, Chen MH (2018) Modified Newton-NSS method for solving systems of nonlinear equations. Numer Algorithms 77:1–21

Darvishi MT, Barati A (2007) A third-order Newton-type method to solve systems of nonlinear equations. Appl Math Comput 187(2):630–635

Dembo RS, Eisenstat SC, Steihaug T (1982) Inexact newton methods. SIAM J Numer Anal 19(2):400–408

Edalatpour V, Hezari D, Khojasteh Salkuyeh D (2015) Accelerated generalized SOR method for a class of complex systems of linear equations. Math Commun 20(1):37–52

Feng YY, Wu QB (2021) MN-PGSOR method for solving nonlinear systems with block two-by-two complex symmetric Jacobian matrices. J Math 2021:1–18

Golub GH, Varga RS (1961) Chebyshev semi-iterative methods, successive overrelaxation iterative methods, and second order Richardson iterative methods. Numer Math 3(1):157–168

Hezari D, Edalatpour V, Salkuyeh DK (2015) Preconditioned GSOR iterative method for a class of complex symmetric system of linear equations. Numer Linear Algebra Appl 22(4):761–776

Hezari D, Salkuyeh DK, Edalatpour V (2016) A new iterative method for solving a class of complex symmetric system of linear equations. Numer Algorithms 73:927–955

Huang ZG (2021) Modified two-step scale-splitting iteration method for solving complex symmetric linear systems. Comput Appl Math 40(4):122

King RF (1973) A family of fourth order methods for nonlinear equations. SIAM J Numer Anal 10(5):876–879

Liang ZZ, Zhang GF (2016) On SSOR iteration method for a class of block two-by-two linear systems. Numer Algorithms 71:655–671

Liang ZZ, Zhang GF (2021) On Chebyshev accelerated iteration methods for two-by-two block linear systems. J Comput Appl Math 391:113449

Ortega JM, Rheinboldt WC (2000) Iterative solution of nonlinear equations in several variables. SIAM, Philadelphia

Rheinboldt WC (1998) Methods for solving systems of nonlinear equations. SIAM, Philadelphia

Salkuyeh DK, Siahkolaei TS (2018) Two-parameter TSCSP method for solving complex symmetric system of linear equations. Calcolo 55:1–22

Salkuyeh DK, Hezari D, Edalatpour V (2015) Generalized successive overrelaxation iterative method for a class of complex symmetric linear system of equations. Int J Comput Math 92(4):802–815

Shirilord A, Dehghan M (2022) Single step iterative method for linear system of equations with complex symmetric positive semi-definite coefficient matrices. Appl Math Comput 426:127111

Wang T, Zheng Q, Lu L (2017) A new iteration method for a class of complex symmetric linear systems. J Comput Appl Math 325:188–197

Wu Q, Chen M (2013) Convergence analysis of modified Newton-HSS method for solving systems of nonlinear equations. Numer Algorithms 64:659–683
Xiao XY, Wang X (2018) A new single-step iteration method for solving complex symmetric linear systems. Numer Algorithms 78:643–660
Xiao Y, Wu Q, Zhang Y (2021) Newton-PGSS and its improvement method for solving nonlinear systems with saddle point Jacobian matrices. J Math 2021:1–18
Xie F, Lin RF, Wu QB (2020) Modified Newton-DSS method for solving a class of systems of nonlinear equations with complex symmetric Jacobian matrices. Numer Algorithms 85:951–975
Yang AL, Wu YJ (2012) Newton-MHSS methods for solving systems of nonlinear equations with complex symmetric Jacobian matrices. Numer Algebra Control Optim 2(4):839–853
Yu X, Wu Q (2022) Modified Newton-SSTS method for solving a class of nonlinear systems with complex symmetric Jacobian matrices. Comput Appl Math 41(6):258
Zhang J, Wang Z, Zhao J (2019) Double-step scale splitting real-valued iteration method for a class of complex symmetric linear systems. Appl Math Comput 353:338–346
Zhang L, Wu QB, Chen MH et al (2021) Two new effective iteration methods for nonlinear systems with complex symmetric Jacobian matrices. Comput Appl Math 40:1–27
Zhang Y, Wu Q, Feng Y et al (2022) Modified Newton-PSBTS method for solving complex nonlinear systems with symmetric Jacobian matrices. Appl Numer Math 182:308–329
Zheng Z, Huang FL, Peng YC (2017) Double-step scale splitting iteration method for a class of complex symmetric linear systems. Appl Math Lett 73:91–97
Zhong HX, Chen GL, Guo XP (2015) On preconditioned modified Newton-MHSS method for systems of nonlinear equations with complex symmetric Jacobian matrices. Numer Algorithms 69(3):553–567