East Asian Journal on Applied Mathematics Vol. **13**, No. 1, pp. 76-94
doi: 10.4208/eajam.291121.090722 February 2023

A Class of Relaxed TTSCSP Iteration Methods for Weakly Nonlinear Systems

Xiaohui Yu and Qingbiao Wu*

School of Mathematical Sciences, Zhejiang University, HangZhou, 310027, Zhejiang, P.R.China.

Received 29 November 2021; Accepted (in revised version) 9 July 2022.

Abstract. A relaxed TTSCSP (RTTSCSP) iteration method for complex linear systems is constructed. Based on the strong dominance and separability of linear and nonlinear terms, Picard-RTTSCSP and nonlinear RTTSCSP-like iterative methods are developed and applied to complex systems of weakly nonlinear equations. The convergence of the method is investigated. Besides, optimal iterative parameters minimizing the upper bound of the spectral radius are derived. Numerical examples show the effectiveness and applicability of the methods to complex systems of weakly nonlinear equations.

AMS subject classifications: 65M10, 78A48

Key words: Picard-RTTSCSP method, nonlinear RTTSCSP-like method, weakly nonlinear system, convergence.

1. Introduction

Let $\mathbb{D}$ be an open convex subset of the n-dimensional complex linear space $\mathbb{C}^n$ and $\phi : \mathbb{D} \to \mathbb{C}^n$ a continuously differentiable nonlinear function. We consider the complex weakly nonlinear system

$$Ax := (W + iT)x = \phi(x), \tag{1.1}$$

where $W, T \in \mathbb{R}^{n\times n}$ are real symmetric positive definite matrices and $i = \sqrt{-1}$ the imaginary unit. This system can be written as

$$F(x) := Ax - \phi(x) = 0,$$

and in what follows, we mainly use this form of the Eq. (1.1).

If the linear part A is strongly dominant over in a specific norm, then the system (1.1) is called weakly nonlinear — cf. [6, 10, 33]. Finding the solutions of systems (1.1) has numerous applications in engineering, nonlinear partial differential equations, saddle point problems in image processing and nonlinear optimization problems [5–7, 9, 16, 21]. As

*Corresponding author. *Email address:* qbwu@zju.edu.cn (Q. Wu)

far as the general nonlinear equation $F(x) = 0$, the most popular solution method is the second-order classical iteration Newton scheme — i.e.

$$x_{n+1} = x_n - F^{'}(x_n)^{-1} F(x_n), \quad n = 0, 1, 2, \ldots,$$

where $F : \mathbb{D} \to \mathbb{C}^n$ is a continuous differentiable function. Nevertheless, in practical computations it is difficult to construct the Jacobian matrix and derive an exact solution of the above equations. Therefore, the inexact Newton method

$$\begin{aligned} &x_{n+1} = x_n + s_n, \\ &F'(x_n)s_n = -F(x_n) + r_n, \\ &\|r_n\|/\|F(x_n)\| \le \eta_n, \end{aligned}$$

where $\eta_n \in [0, 1)$, attracted a substantial attention — cf. [1, 3, 15, 20]. In particular, combining the Newton method with inner solvers allowed to adjust various numerical methods for linear systems to non-linear equations [1, 4, 8, 12–14, 23, 26, 28, 29, 31].

By adopting the HSS scheme as the internal solver, Bai [8] established a Newton-HSS methods for solving nonlinear systems with large sparse positive definite Jacobian matrices. Yang and Bai [10] proposed nonlinear HSS-like and the Picard-HSS iteration methods for weakly nonlinear systems with specific properties. Considering weakly nonlinear equations with large sparse matrices, Pu and Zhu [22] improved algorithms for linear equations and used them to develop generalized nonlinear compound splitting iterative methods, named nonlinear GPHSS-like and Picard-GPHSS. The solution of Toeplitz systems of weakly nonlinear equations have been studied by Zhu and Zhang [34], who developed nonlinear CSCS-like and Picard-CSCS iterative schemes, which are nonlinear composite iteration algorithms. Adopting the AIPCG iteration technique, Jiang and Guo [18] established Picard-AIPCG algorithms for solving the equations of this type. A class of lopsided PMHSS iteration methods and a nonlinear LHSS-like method converging to the unique solution, have been proposed by Li and Wu [19]. Zeng and Zhang [30] constructed a PTGHSS iteration scheme and two PTGHSS-based iteration methods for weakly nonlinear systems. Chen *et al.* [11] applied nonlinear DPMHSS-like and Picard-DPMHSS methods, based on double-parameter PMHSS iterative technique, to weakly nonlinear systems. Using an HSS scheme, Amiri [2] established a Jacobi-free HSS algorithm for weakly nonlinear systems. Combining accelerated GSOR and preconditioned GSOR method, Wu and Qi [27] introduced Picard-preconditioned GSOR and Picard-accelerated GSOR methods for weakly nonlinear equations with complex matrices.

Taking into account excellent properties and efficient performance of HSS-like iterative methods, Zheng *et al.* [32] applied a DSS iteration scheme to complex symmetric linear equations. Xie and Wu [28] adjusted the Newton-DSS method to nonlinear systems with large sparse complex symmetric Jacobian matrices. Besides, a complex Sylvester matrix equation has been solved by Feng and Wu [17] by the Lopsided DSS iteration method. In order to accelerate the DSS scheme, Siahkolaei *et al.* [24] established a two-parameter two-step scale-splitting (TTSCSP) method, which was then used in the solution of systems of weakly nonlinear equations by Siahkolaei *et al.* [25].

The aim of the present work is to improve the efficiency of TTSCSP iteration method and apply it to weakly nonlinear systems. Introducing a relaxation parameter ω, we can speed up slowly convergent iteration methods and even make divergent iteration methods to converge. Moreover, relaxation parameters can make the iterative methods more flexible. We start with a relaxed TTSCSP (RTTSCSP) iteration method for complex linear systems and introduce a relaxation parameter to the TTSCSP iteration method. Besides, we consider a class of RTTSCSP-based iteration methods, called Picard-RTTSCSP and nonlinear RTTSCSP-like iteration methods, which are aimed to improve the efficiency of numerical methods for complex weakly nonlinear systems.

This paper is organized as follows. In Section 2, we construct an RTTSCSP iteration scheme for complex linear systems and discuss its convergence. In Section 3, we introduce Picard-RTTSCSP and nonlinear RTTSCSP-like iteration methods for complex weakly nonlinear systems and establish their convergence. Section 4 is devoted to optimal parameters for these new iteration methods. The results of numerical experiments presented in Section 5 support the theoretical findings. Finally, our conclusions are given in Section 6.

2. RTTSCSP Iteration Method

If $\phi(x) = b$, $b \in \mathbb{C}^n$, then (1.1) is the linear system

$$(W + iT)x = b$$

with symmetric positive definite matrices $W, T \in \mathbb{R}^{n\times n}$. For such systems, Salkuyeh and Siahkolaei [24] introduced the so-called TTSCSP iteration method, which can be described as follows. Let $x_0 \in \mathbb{C}^n$ be an initial guess. For $k = 0, 1, 2, \ldots$, determine the next iteration value x_{k+1} from the equations

$$\begin{aligned}(\alpha W + T)x_{k+1/2} &= i(W - \alpha T)x_k + (\alpha - i)b,\\ (\beta T + W)x_{k+1} &= i(\beta W - T)x_{k+1/2} + (1 - i\beta)b,\end{aligned}$$

until the iterative sequence $\{x_k\}_{k=0}^{\infty} \subset \mathbb{C}^n$ converges.

To improve the efficiency of this approach, we introduce a relaxation parameter and call the corresponding scheme the RTTSCSP iteration method. More exactly, let $x_0 \in \mathbb{C}^n$ be an initial guess. For $k = 0, 1, 2, \ldots$, find the next iteration value x_{k+1} from the equations

$$\begin{aligned}x_{k+1/2} &= (1-\omega)x_k + \omega(\alpha W + T)^{-1}\big[i(W - \alpha T)x_k + (\alpha - i)b\big],\\ x_{k+1} &= (1-\omega)x_{k+1/2} + \omega(\beta T + W)^{-1}\big[i(\beta W - T)x_{k+1/2} + (1 - i\beta)b\big],\end{aligned}$$

until the iterative sequence $\{x_k\}_{k=0}^{\infty} \subset \mathbb{C}^n$ converges.

Note that α, β are positive constants and ω is a relaxation parameter. It is worth noting that if $\omega = 1$, the RTTSCSP method becomes an TTSCSP method. Simple manipulations show that RTTSCSP iteration scheme can be represented in the form

$$\begin{aligned}x_{k+1/2} &= \big[I + \omega(i-\alpha)(\alpha W + T)^{-1}A\big]x_k + \omega(\alpha - i)(\alpha W + T)^{-1}b,\\ x_{k+1} &= \big[I + \omega(i\beta - 1)(\beta T + W)^{-1}A\big]x_{k+1/2} + \omega(1 - i\beta)(\beta T + W)^{-1}b,\end{aligned}$$

and then written as

$$x_{k+1} = T_{\alpha,\beta}(\omega)x_k + G_{\alpha,\beta}(\omega),$$

where

$$\begin{aligned}
T_{\alpha,\beta}(\omega) &= \big[I + \omega(\beta T + W)^{-1}(i\beta - 1)A\big]\big[I + \omega(\alpha W + T)^{-1}(i - \alpha)A\big] \\
&= (\beta T + W)^{-1}\big[(1 - \omega + i\beta\omega)W + (\beta - \beta\omega - i\omega)T\big] \\
&\quad \times (\alpha W + T)^{-1}\big[(\alpha - \omega\alpha + i\omega)W + (1 - \omega - i\omega\alpha)T\big], \\
G_{\alpha,\beta}(\omega) &= \omega(\beta T + W)^{-1}\big[(2\alpha - \omega\alpha - i + i\omega - i\alpha\beta)W \\
&\qquad + (1 + \alpha\beta - \omega\alpha\beta + i\omega\beta - 2i\beta)T + \omega\alpha - i\omega\big](\alpha W + T)^{-1}.
\end{aligned}$$

The following theorem describe the convergence of the RTTSCSP method.

Theorem 2.1. *Let $W, T \in \mathbb{R}^{n\times n}$ be symmetric positive definite matrices and μ_j, $j = 1, 2, \ldots, n$ the eigenvalues of $S = W^{-1/2}TW^{-1/2}$. If the inequalities*

$$\begin{aligned}
&(4\omega^2 - \omega^3 - 6\omega + 4)(\alpha + \mu_j)^2(\beta\mu_j + 1)^2 - \omega(1-\omega)^2(\beta - \mu_j)^2(\alpha + \mu_j)^2 \\
&> \omega^3(\beta - \mu_j)^2(1 - \alpha\mu_j)^2 + \omega(1-\omega)^2(1 - \alpha\mu_j)^2(\beta\mu_j + 1)^2
\end{aligned} \tag{2.1}$$

hold for all $j = 1, 2, \ldots, n$, then the RTTSCSP iteration method converges.

Proof. Direct calculation show that

$$\begin{aligned}
T_{\alpha,\beta}(\omega) = W^{-1/2}(\beta S + I)^{-1}&\big[(1 - \omega + i\beta\omega)I + (\beta - \beta\omega - i\omega)S\big] \\
&\times (\alpha I + S)^{-1}\big[(\alpha - \omega\alpha + i\omega)I + (1 - \omega - i\omega\alpha)S\big]W^{1/2}.
\end{aligned}$$

Considering

$$\begin{aligned}
\tilde{T}_{\alpha,\beta}(\omega) = (\beta S + I)^{-1}&\big[(1 - \omega + i\beta\omega)I + (\beta - \beta\omega - i\omega)S\big] \\
&\times (\alpha I + S)^{-1}\big[(\alpha - \omega\alpha + i\omega)I + (1 - \omega - i\omega\alpha)S\big],
\end{aligned}$$

we note that the matrices $T_{\alpha,\beta}(\omega)$ and $\tilde{T}_{\alpha,\beta}(\omega)$ are similar, so that they have the same eigenvalues. Since W and T are real symmetric positive definite, the all eigenvalues of the matrix S are positive. Therefore,

$$\rho\big(T_{\alpha,\beta}(\omega)\big) = \rho\big(\tilde{T}_{\alpha,\beta}(\omega)\big) = \max_{\mu_j\in\sigma(S)} |\lambda(\alpha, \beta, \omega, \mu_j)|,$$

where

$$\lambda(\alpha, \beta, \omega, \mu_j) = \frac{[1 - \omega + i\beta\omega + (\beta - \beta\omega - i\omega)\mu_j][\alpha - \omega\alpha + i\omega + (1 - \omega - i\omega\alpha)\mu_j]}{(\beta\mu_j + 1)(\alpha + \mu_j)}.$$

Writing the eigenvalues $\lambda(\alpha, \beta, \omega, \mu_j)$

$$\lambda(\alpha, \beta, \omega, \mu_j) = \left[1 - \omega + \frac{i\omega(\beta - \mu_j)}{\beta\mu_j + 1}\right]\left[1 - \omega + \frac{i\omega(1 - \alpha\mu_j)}{\alpha + \mu_j}\right],$$

we note that the real and imaginary parts of $\lambda(\alpha,\omega,\mu_j)$ have the form

$$Re\big(\lambda(\alpha,\omega,\mu_j)\big) = (1-\omega)^2 - \omega^2\frac{(\beta-\mu_j)(1-\alpha\mu_j)}{(\alpha+\mu_j)(\beta\mu_j+1)},$$
$$Im\big(\lambda(\alpha,\omega,\mu_j)\big) = \omega(1-\omega)\left(\frac{1-\alpha\mu_j}{\alpha+\mu_j}+\frac{\beta-\mu_j}{\beta\mu_j+1}\right).$$

Therefore,

$$\begin{aligned}&Re^2\big(\lambda(\alpha,\omega,\mu_j)\big)+Im^2\big(\lambda(\alpha,\omega,\mu_j)\big)\\ &=(1-\omega)^4+\omega^4\frac{(\beta-\mu_j)^2(1-\alpha\mu_j)^2}{(\alpha+\mu_j)^2(\beta\mu_j+1)^2}+\omega^2(1-\omega)^2\left[\frac{(1-\alpha\mu_j)^2}{(\alpha+\mu_j)^2}+\frac{(\beta-\mu_j)^2}{(\beta\mu_j+1)^2}\right].\end{aligned}$$

Hence, if the inequality (2.1) holds, the spectral radius of iteration matrix $\rho(T_{\alpha,\beta}(\omega)) < 1$. This yields the convergence of the sequence of interest. □

3. Iteration Methods for Weakly Nonlinear Systems

The linearization of nonlinear equations is one of the most attractive strategies in solving nonlinear systems. Since the linear term Ax has strong dominance over the nonlinear term $\phi(x)$, we can use the Picard iterative method to obtain solutions of weakly nonlinear systems (1.1). A linear system has to be solved in each Picard iteration step. Applying the RTTSCSP method as an internal iteration for solving these linear equations, one can then find approximations for the iteration x_k.

3.1. The Picard-RTTSCSP iteration method

Assume that $\phi : \mathbb{D} \to \mathbb{C}^n$ is a continuous function, $A \in C^{n\times n}$ a positive definite matrix, and W and T respectively denote the Hermitian and skew-Hermitian parts os A, i.e. $W = (A+A^*)/2$ and $iT = (A-A^*)/2$. Choose an initial guess $x_0 \in D$ and determine x_{k+1}, $k = 0,1,2,\ldots$ by the method below until $\{x_k\}$ satisfies the stopping criteria.

(a) Let $x_{k,0} := x_k$.

(b) If $l = 0,1,\ldots,l_k-1$, determine $x_{k,l+1}$ from the linear equations

$$x_{k,l+1/2} = \big[I+\omega(i-\alpha)(\alpha W+T)^{-1}A\big]x_{k,l} + \omega(\alpha-i)(\alpha W+T)^{-1}\phi(x_k),$$
$$x_{k,l+1} = \big[I+\omega(i\beta-1)(\beta T+W)^{-1}A\big]x_{k,l+1/2} + \omega(1-i\beta)(\beta T+W)^{-1}\phi(x_k),$$

where $\alpha,\beta > 0$ are given constants and ω is a relaxation parameter.

(c) Set $x_{k+1} := x_{k,l_k}$.

Applying the Picard-RTTSCSP iteration schemes yields

$$x_{k+1} = T_{\alpha,\beta}(\omega)^{l_k} x_k + \sum_{j=0}^{l_k-1} T_{\alpha,\beta}(\omega)^j G_{\alpha,\beta}(\omega)\phi(x_k), \quad k = 0, 1, 2, \ldots,$$

where $T_{\alpha,\beta}(\omega)$ and $G_{\alpha,\beta}(\omega)$ are above defined matrices.

If $x_* \in \mathbb{D}$ is the exact solution of the system (1.1), then

$$x_* = T_{\alpha,\beta}(\omega)^{l_k} x_* + \sum_{j=0}^{l_k-1} T_{\alpha,\beta}(\omega)^j G_{\alpha,\beta}(\omega)\phi(x_*),$$

so that

$$x_{k+1} - x_* = T_{\alpha,\beta}(\omega)^{l_k} (x_k - x_*) + \sum_{j=0}^{l_k-1} T_{\alpha,\beta}(\omega)^j G_{\alpha,\beta}(\omega)\big[\phi(x_k) - \phi(x_*)\big]. \tag{3.1}$$

Let $a \in \mathbb{R}$. We denote by $\lfloor a \rfloor$ the smallest integer greater than or equal to a.

Theorem 3.1. *Assume that $\phi : \mathbb{D} \to \mathbb{C}^n$ is a G-differentiable function in a neighborhood $\mathbb{D}_0$ of an exact solution $x^* \in \mathbb{D}$ of (1.1), $\phi'(x)$ is continuous, and set*

$$\mu(\alpha, \beta, \omega) = \|T_{\alpha,\beta}(\omega)\|, \quad \omega = \|A^{-1}\phi'(x^*)\|, \quad \theta = \|A^{-1}\|.$$

If

$$l_0 \geq \left\lfloor \ln\left(\frac{1-\omega}{1+\omega}\right) / \ln\big(\mu(\alpha, \beta, \omega)\big) \right\rfloor,$$

then for any initial guess $x_0 \in \mathbb{N}$ and any positive integer sequence l_k, $k = 0, 1, \ldots$, there exists an open neighborhood $\mathbb{N}$ of x^* in $\mathbb{D}_0$ such that the iteration sequence $\{x_k\}_{k=0}^{\infty}$ generated by the Picard-RTTSCSP iteration method is well-defined and converges to x_*. More exactly,

$$\limsup_{k\to\infty} \|x_k - x_*\|^{1/k} \leq \omega + (1+\omega)\mu(\alpha, \beta, \omega)^{l_0}, \quad l_0 = \liminf_{k\to\infty} l_k.$$

If $\lim_{k\to\infty} l_k = \infty$, the convergence speed is R-linear with the R-factor being at most ω, i.e.

$$\limsup_{k\to\infty} \|x_k - x_*\|^{1/k} \leq \omega.$$

Proof. Set

$$E(x, x_*) = \phi(x) - \phi(x_*) - \phi'(x_*)(x - x_*),$$

and write the Eq. (3.1) as

$$\begin{aligned} x_{k+1} - x_* = &\left[T_{\alpha,\beta}(\omega)^{l_k} (x_k - x_*) + \sum_{j=0}^{l_k-1} T_{\alpha,\beta}(\omega)^j G_{\alpha,\beta}(\omega)\phi'(x_*)\right](x_k - x_*) \\ &+ \sum_{j=0}^{l_k-1} T_{\alpha,\beta}(\omega)^j G_{\alpha,\beta}(\omega) E(x_k, x_*) \end{aligned}$$

$$
\begin{aligned}
&= \big[T_{\alpha,\beta}(\omega)^{l_k} + \big(I - T_{\alpha,\beta}(\omega)^{l_k}\big)A^{-1}\phi'(x_*)\big](x_k - x_*) \\
&\quad + \big(I - T_{\alpha,\beta}(\omega)^{l_k}\big)A^{-1}E(x_k, x_*) \\
&= \big[A^{-1}\phi'(x_*) + T_{\alpha,\beta}(\omega)^{l_k}(I - A^{-1}\phi'(x_*))\big](x_k - x_*) \\
&\quad + \big(I - T_{\alpha,\beta}(\omega)^{l_k}\big)A^{-1}E(x_k, x_*).
\end{aligned}
$$

It follows that

$$
\begin{aligned}
\|x_{k+1} - x_*\| &\le \Big[\, \big\|A^{-1}\phi'(x_*)\big\| + \big\|T_{\alpha,\beta}(\omega)^{l_k}\big(I - A^{-1}\phi'(x_*)\big)\big\|\,\Big]\|x_k - x_*\| \\
&\quad + \|I - T_{\alpha,\beta}(\omega)^{l_k}\|\|A^{-1}\|\|E(x_k, x_*)\| \\
&\le \big[\omega + \mu(\alpha,\beta,\omega)^{l_k}(1+\omega)\big]\|x_k - x_*\| + \theta\varepsilon\big(1 + \mu(\alpha,\beta,\omega)^{l_k}\big)\|x_k - x_*\|,
\end{aligned}
$$

where the last inequality is implied by the G-differentiability of $\phi(x)$, and ε is a sufficiently small constant. Using induction gives

$$
\|x_{k+1} - x_*\| \le \big[\omega + (1+\omega)\mu(\alpha,\beta,\omega)^{l_k} + \theta\varepsilon\big(1 + \mu(\alpha,\beta,\omega)^{l_k}\big)\big]^{k+1}\|x_0 - x_*\|.
$$

Since ε can be chosen sufficiently small and the inequality (3.1) holds, we obtain

$$
\omega + (1+\omega)\mu(\alpha,\beta,\omega)^{l_k} + \theta\varepsilon\big(1 + \mu(\alpha,\beta,\omega)^{l_k}\big) < 1.
$$

Therefore, $\lim_{k\to\infty} x_k = x_*$, and

$$
\limsup_{k\to\infty} \|x_k - x_*\|^{1/k} \le \omega + (1+\omega)\mu(\alpha,\beta,\omega)^{l_0}, \quad l_0 = \liminf_{k\to\infty} l_k.
$$

In particular, if $\lim_{k\to\infty} l_k = \infty$, then it is easily seen that $\limsup_{k\to\infty} \|x_k - x_*\|^{1/k} \le \omega$. □

3.2. A nonlinear RTTSCSP-like method

Let A and ϕ be as in Subsection 3.1. Given an initial guess $x_0 \in D$, determine x_{k+1} for $k = 0, 1, 2, \ldots$ from the equations

$$
\begin{aligned}
x_{k+1/2} &= \big[I + \omega(i - \alpha)(\alpha W + T)^{-1}A\big]x_k + \omega(\alpha - i)(\alpha W + T)^{-1}\phi(x_k), \\
x_{k+1} &= \big[I + \omega(i\beta - 1)(\beta T + W)^{-1}A\big]x_{k+1/2} + \omega(1 - i\beta)(\beta T + W)^{-1}\phi(x_{k+1/2})
\end{aligned}
$$

until $\{x_k\}$ satisfies a stopping criteria. Note that α, β, ω are given constants and if the exact solution $x_* \in \mathbb{D}$ is the exact solution of (1.1), we suppose that $\phi'(x)$ is continuous at this point. Using the notations

$$
\begin{aligned}
G(x) &= \big[I + \omega(i - \alpha)(\alpha W + T)^{-1}A\big]x + \omega(\alpha - i)(\alpha W + T)^{-1}\phi(x), \\
H(x) &= \big[I + \omega(i\beta - 1)(\beta T + W)^{-1}A\big]x + \omega(1 - i\beta)(\beta T + W)^{-1}\phi(x),
\end{aligned}
$$

and

$$
\chi(x) = H\big(G(x)\big),
$$

we can write the nonlinear RTTSCSP-like iteration method can as

$$x_{k+1} = \chi(x_k), \quad k = 0,1,2,\ldots.$$

Assuming that $x_* \in \mathbb{D}$ is the exact solution of the system (1.1) and using the chain rule gives

$$\begin{aligned}\chi'(x_*) &= H'(x_*)G'(x_*)\\ &= \left[I + \omega(i\beta - 1)(\beta T + W)^{-1}A + \omega(1 - i\beta)(\beta T + W)^{-1}\phi'(x_*)\right]\\ &\quad \times \left[I + \omega(i - \alpha)(\alpha W + T)^{-1}A + \omega(\alpha - i)(\alpha W + T)^{-1}\phi'(x_*)\right].\end{aligned}$$

Theorem 3.2 (cf. Bai & Yang [10]). *Assume that $x_* \in \mathbb{D}$ is the exact solution of* (1.1), *$\phi : \mathbb{D} \to \mathbb{C}^n$ is G-differentiable at x^* and set*

$$\begin{aligned}T_{\alpha,\beta}(\omega, x_*) = &\left[I + \omega(i\beta - 1)(\beta T + W)^{-1}A + \omega(1 - i\beta)(\beta T + W)^{-1}\phi'(x_*)\right]\\ &\times \left[I + \omega(i - \alpha)(\alpha W + T)^{-1}A + \omega(\alpha - i)(\alpha W + T)^{-1}\phi'(x_*)\right].\end{aligned}$$

If the spectral radius $\rho(T_{\alpha,\beta}(\omega, x_*)) < 1$, then $x_* \in \mathbb{D}$ is a point of attraction of the nonlinear RTTSCSP-like iteration.

Theorem 3.3. *Let $S = W^{-1/2}TW^{-1/2}$. Assume that $x_* \in \mathbb{D}$ is the exact solution of* (1.1), *$\phi : \mathbb{D} \to \mathbb{C}^n$ is G-differentiable at x^*, and set*

$$\begin{aligned}&\delta = \max\left\{\|(\alpha W + T)^{-1}\phi'(x_*)\|, \|(\beta T + W)^{-1}\phi'(x_*)\|\right\},\\ &a = \|I + \omega(i - \alpha)(\alpha W + T)^{-1}A\| = \max_{\mu_j \in \lambda(S)}\left|\frac{\alpha - \omega\alpha + i\omega + (1 - \omega - i\omega\alpha)\mu_j}{\alpha + \mu_j}\right|,\\ &b = \|I + \omega(i\beta - 1)(\beta T + W)^{-1}A\| = \max_{\mu_j \in \lambda(S)}\left|\frac{1 - \omega + i\omega\beta + (\beta - \omega\beta - i\omega)\mu_j}{\beta\mu_j + 1}\right|,\\ &R = \omega\sqrt{\left(\sqrt{1+\beta^2}a - \sqrt{1+\alpha^2}b\right)^2 + 4\sqrt{(1+\alpha^2)(1+\beta^2)}} + \omega\left(\sqrt{1+\beta^2}a + \sqrt{1+\alpha^2}b\right).\end{aligned}$$

If

$$\delta < \frac{2(1 - ab)}{R}, \tag{3.2}$$

then

$$\rho\left(T_{\alpha,\beta}(\omega, x_*)\right) < 1.$$

Proof. Direct calculations give

$$\begin{aligned}T_{\alpha,\beta}(\omega, x_*) = T_{\alpha,\beta}(\omega) &+ \omega(\alpha - i)\left[I + \omega(i\beta - 1)(\beta T + W)^{-1}A\right](\alpha W + T)^{-1}\phi'(x_*)\\ &+ \omega(1 - i\beta)(\beta T + W)^{-1}\phi'(x_*)\left[I + \omega(i - \alpha)(\alpha W + T)^{-1}A\right]\\ &+ \omega^2(1 - i\beta)(\alpha - i)(\beta T + W)^{-1}\phi'(x_*)(\alpha W + T)^{-1}\phi'(x_*)\end{aligned}$$

with $T_{\alpha,\beta}(\omega)$ defined similar to the previous section

$$\|T_{\alpha,\beta}(\omega)\| \le \|I+\omega(i\beta-1)(\beta T+W)^{-1}A\| \times \|I+\omega(i-\alpha)(\alpha W+T)^{-1}A\| = ab.$$

Consequently,

$$\begin{aligned}\|T_{\alpha,\beta}(\omega,x_*)\| \le\, & \|T_{\alpha,\beta}(\omega)\| + \left\|\omega(\alpha-i)\left[I+\omega(i\beta-1)(\beta T+W)^{-1}A\right](\alpha W+T)^{-1}\phi'(x_*)\right\| \\ & + \left\|\omega(1-i\beta)(\beta T+W)^{-1}\phi'(x_*)\left[I+\omega(i-\alpha)(\alpha W+T)^{-1}A\right]\right\| \\ & + \|\omega^2(1-i\beta)(\alpha-i)(\beta T+W)^{-1}\phi'(x_*)(\alpha W+T)^{-1}\phi'(x_*)\| \\ \le\, & \|T_{\alpha,\beta}(\omega)\| + \omega\sqrt{1+\alpha^2}\|I+\omega(i\beta-1)(\beta T+W)^{-1}A\|\|(\alpha W+T)^{-1}\phi'(x_*)\| \\ & + \omega\sqrt{1+\beta^2}\|(\beta T+W)^{-1}\phi'(x_*)\|\|I+\omega(i-\alpha)(\alpha W+T)^{-1}A\| \\ & + \omega^2\sqrt{(1+\alpha^2)(1+\beta^2)}\|(\beta W+T)^{-1}\phi'(x_*)\|\|(\alpha W+T)^{-1}\phi'(x_*)\| \\ \le\, & ab + \omega\left(\sqrt{1+\beta^2}a + \sqrt{1+\alpha^2}b\right)\delta + \omega^2\sqrt{(1+\alpha^2)(1+\beta^2)}\delta^2.\end{aligned}$$

Under the condition (3.2), we have

$$ab + \omega\left(\sqrt{1+\beta^2}a + \sqrt{1+\alpha^2}b\right)\delta + \omega^2\sqrt{(1+\alpha^2)(1+\beta^2)}\delta^2 < 1.$$

Therefore,

$$\rho\big(T(\alpha,\beta,x_*)\big) \le \|T(\alpha,\beta;x^*)\| < 1,$$

and the proof is complete. □

4. Parameters Optimization

It is known that the convergence of the iterative method depends on two main factors viz. the weak nonlinearity of the systems and the choice of an optimal parameters minimizing the spectral radius of the iterative matrix. The former is determined by the problem, and the latter is discussed below. Calculations show that the spectral radius of the iteration matrix $T_{\alpha,\beta}(\omega)$ can be estimated as follows:

$$\begin{aligned}\rho\big(T_{\alpha,\beta}(\omega)\big) =\, & \max_{\mu_j\in\sigma(S)} |\lambda(\alpha,\beta,\omega,\mu_j)| \\ \le\, & \max_{\mu_j\in\sigma(S)} \left|1-\omega+\frac{i\omega(\beta-\mu_j)}{\beta\mu_j+1}\right|\left|1-\omega+\frac{i\omega(1-\alpha\mu_j)}{\alpha+\mu_j}\right| \\ \le\, & \sqrt{(1-\omega)^2+\omega^2\max_{\mu_j\in\sigma(S)}\frac{(\beta-\mu_j)^2}{(\beta\mu_j+1)^2}}\sqrt{(1-\omega)^2+\omega^2\max_{\mu_j\in\sigma(S)}\frac{(1-\alpha\mu_j)^2}{(\alpha+\mu_j)^2}} \\ :=\, & \varrho(\alpha,\beta,\omega).\end{aligned}$$

Optimizing the upper bound $\varrho(\alpha,\beta,\omega)$, we express optimal parameters as

$$(\alpha^*,\beta^*,\omega^*) = \arg\min_{\alpha,\beta,\omega} \varrho(\alpha,\beta,\omega).$$

Setting

$$f_\mu(\alpha)=\frac{1-\alpha\mu}{\alpha+\mu},\quad g_\mu(\beta)=\frac{\beta-\mu}{1+\beta\mu},$$

we have

$$\alpha^*=\arg\min_{\alpha>0}\max_{\mu_j\in\sigma(S)}|f_{\mu_j}(\alpha)|,\quad \beta^*=\arg\min_{\beta>0}\max_{\mu_j\in\sigma(S)}|g_{\mu_j}(\beta)|.$$

Following the discussion in [24] gives

$$\alpha^*=\frac{1-\mu_{\min}\mu_{\max}+\sqrt{(1-\mu_{\min}\mu_{\max})^2+(\mu_{\min}+\mu_{\max})^2}}{\mu_{\min}+\mu_{\max}},\quad \beta^*=\frac{1}{\alpha^*},$$

so that

$$\min_{\alpha>0}\max_{\mu_j\in\sigma(S)}|f_{\mu_j}(\alpha)|=\min_{\beta>0}\max_{\mu_j\in\sigma(S)}|g_{\mu_j}(\beta)|=\left|\frac{1-\alpha^*\mu_{\min}}{\alpha^*+\mu_{\min}}\right|=:A.$$

It follows that the optimal relaxation parameter is

$$\omega^*=\arg\min_{\omega}\left\{(1-\omega)^2+A^2\omega^2\right\}=\frac{1}{A^2+1}.$$

5. Numerical Results

In order to verify the theoretical finding for the nonlinear RTTSCSP-like and Picard-RTTSCSP methods, we use examples from [11,19]. All computations are made using MATLAB Version R2019b with 16.00 GB RAM and 1.60 GHz Intel Core i5 CPU. In particular, we compare Picard-RTTSCSP and Picard-LPMHSS methods, Picard-DPMHSS and Picard-TTSCSP methods in terms of CPU and iteration time. In addition, a nonlinear LPMHSS-like method, nonlinear DPMHSS-like method, and nonlinear TTSCSP-like method are compared with the nonlinear RTTSCSP-like method. We chose the initial guess $x_0=0$, and the stopping criteria for external iterations as

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

In all the internal iteration, the stopping criterion is

$$\frac{\|F'(x_k)s_{(k,l_k)}+F(x_k)\|}{\|F(x_k)\|}\le\eta,$$

where a given internal tolerance η is adopted to control the accuracy of all inner iterations.

Note that in what follows, Picard-RTTSCSP, Picard-LPMHSS, Picard-DPMHSS, Picard-TTSCSP and nonlinear RTTSCSP-like, nonlinear LPMHSS-like, nonlinear DPMHSS-like, nonlinear TTSCSP-like methods are respectively abbreviated as P-RTTSCSP, P-LPMHSS, P-DPMHSS, P-TTSCSP and NL-RTTSCSP, NL-LPMHSS, NL-DPMHSS, and NL-TTSCSP.

Example 5.1. Let $\Omega = (0,1) \times (0,1)$ and $\partial\Omega$ refer to the boundary of Ω. Consider the weakly nonlinear equation

$$\begin{aligned} &-(\alpha_1 + i\beta_1)(u_{xx} + u_{yy}) + qu = (\alpha_2 + \beta_2 u)e^u && \text{for} \quad (x,y) \in \Omega, \\ &u(x,y) = 0 && \text{for} \quad (x,y) \in \partial\Omega, \end{aligned} \tag{5.1}$$

where $\alpha_1 = \beta_1 = 1$, $\alpha_2 = \beta_2 = 1$, and q is a positive constant. Discretizing the problem (5.1) on an equidistant grid and using central finite difference method with the step width $h = 1/(N+1)$ leads to the weakly nonlinear system

$$F(x) = Ax - \phi(x) = Ax - h^2(\alpha_2 + \beta_2 x)e^x = 0,$$

where

$$A = qh^2 I_n + (\alpha_1 + i\beta_1)(A_N \otimes I_N + I_N \otimes A_N),$$

and $A_N = \text{tridiag}(-1,2,-1)$ is tridiagonal matrix, N the matrix dimension, $\otimes$ the Kronecker product, and $n = N \times N$. The Jacobian matrix has the following form:

$$F^{'}(x) = A - h^2(\alpha_2 + \beta_2 + \beta_2 x)e^x.$$

Note that in actual computations, we employ the experimental optimal values α, β, and ω minimizing the CPU time of the methods. These parameters are shown in Tables 1-3.

Tables 4-6 show the iteration and CPU times for various N, q, and internal tolerance η. In addition, they also present the numbers of inner iterations, outer iterations, and the sum of iterations, which are respectively denoted by ITint, ITout, and IT. In the sense of satisfying precision, it can be ascertained from the experimental results in Tables 4-6 that all of these iterative methods can give an approximate solution to the weakly nonlinear systems in Example 5.1. In particular, the Picard-RTTSCSP is significantly superior to Picard-DPMHSS and Picard-LPMHSS in respect of computing time and iteration step. We note that the sum of iterations in Picard-DPMHSS method is 4 times of that in Picard-RTTSCSP method, and that on average Picard-DPMHSS method requires approximately 3.8 times more computing time Picard-RTTSCSP method. For the same set of problem parameters our method requires least iterations and computing time than the Picard-TTSCSP method. If the matrix dimension N increases, the required CPU time in Picard-TTSCSP method increases faster than in our method. Especially, when $\eta = 0.1$, $q = 1$, from $N = 80$ to $N = 100$, the Picard-TTSCSP method multiplies calculation time with 4.7 times, while the Picard-RTTSCSP method increases CPU time with only 2.6 times. Thus the Picard-RTTSCSP method is more suitable for solving high matrix dimensional problems than other methods. We also observed that both outer and inner iteration steps in the Picard-RTTSCSP approach remain stable when the size of the problems grows, suggesting the extendibility of the method.

From experimental results shown in Table 7, the nonlinear RTTSCSP-like method is significantly predominant to the nonlinear LPMHSS-like and the nonlinear DPMHSS-like methods. The iteration times of nonlinear DPMHSS-like method are 4 to 7 times of nonlinear RTTSCSP-like method, and the average computing time of nonlinear DPMHSS-like method roughly equal 5 times that of nonlinear RTTSCSP-like method. Likewise, the calculation speed of the nonlinear RTTSCSP-like method is faster than the nonlinear TTSCSP-like

Table 1: Experimental optimal values α, β, ω for $\eta = 0.1$.

N	Method	Parameter	$q = 1$	$q = 10$	$q = 100$
50	P-LPMHSS	α	1.06	1.12	1.12
	P-DPMHSS	(α, β)	$(1.9, 0.7)$	$(1.6, 0.8)$	$(0.5, 1.2)$
	P-TTSCSP	(α, β)	$(1.6, 0.6)$	$(0.7, 0.7)$	$(3.1, 0.7)$
	P-RTTSCSP	(α, β, ω)	$(1.5, 0.6, 0.91)$	$(1.2, 1.3, 1.07)$	$(3.1, 0.7, 1.03)$
80	P-LPMHSS	α	1.16	1.00	1.00
	P-DPMHSS	(α, β)	$(1.4, 0.7)$	$(1.3, 0.7)$	$(0.8, 1.4)$
	P-TTSCSP	(α, β)	$(1.4, 0.7)$	$(1.0, 0.6)$	$(3.0, 0.7)$
	P-RTTSCSP	(α, β, ω)	$(1.4, 0.6, 0.92)$	$(1.3, 1.4, 1.06)$	$(3.0, 0.7, 1.04)$
100	P-LPMHSS	α	1.10	1.04	1.24
	P-DPMHSS	(α, β)	$(1.5, 0.9)$	$(1.3, 0.8)$	$(0.7, 1.8)$
	P-TTSCSP	(α, β)	$(1.5, 0.7)$	$(1.5, 0.9)$	$(3.0, 0.7)$
	P-RTTSCSP	(α, β, ω)	$(1.4, 0.7, 0.89)$	$(1.2, 1.3, 1.04)$	$(3.0, 0.7, 1.03)$

Table 2: Experimental optimal values α, β, ω for $\eta = 0.2$.

N	Method	Parameter	$q = 1$	$q = 10$	$q = 100$
50	P-LPMHSS	α	1.25	1.12	1.14
	P-DPMHSS	(α, β)	$(1.8, 0.7)$	$(1.3, 1.2)$	$(0.7, 1.3)$
	P-TTSCSP	(α, β)	$(1.7, 0.8)$	$(2.1, 0.7)$	$(3.1, 0.7)$
	P-RTTSCSP	(α, β, ω)	$(1.2, 0.6, 0.95)$	$(1.2, 1.2, 1.05)$	$(3.0, 0.7, 1.08)$
80	P-LPMHSS	α	1.00	1.00	1.00
	P-DPMHSS	(α, β)	$(1.5, 0.8)$	$(0.8, 1.3)$	$(0.8, 1.2)$
	P-TTSCSP	(α, β)	$(1.6, 0.8)$	$(1.2, 0.6)$	$(3.5, 0.9)$
	P-RTTSCSP	(α, β, ω)	$(1.2, 0.6, 0.87)$	$(1.1, 1.3, 1.05)$	$(3.0, 0.7, 1.03)$
100	P-LPMHSS	α	1.11	1.10	1.08
	P-DPMHSS	(α, β)	$(1.1, 0.8)$	$(1.5, 0.8)$	$(0.9, 1.6)$
	P-TTSCSP	(α, β)	$(1.7, 0.8)$	$(1.5, 0.7)$	$(3.0, 0.7)$
	P-RTTSCSP	(α, β, ω)	$(1.2, 0.7, 0.92)$	$(1.3, 1.3, 1.06)$	$(3.0, 0.7, 1.06)$

Table 3: Experimental optimal values α, β, ω of nonlinear-like methods.

N	Method	Parameter	$q = 1$	$q = 10$	$q = 100$
N = 50	NL-LPMHSS	α	1.0	1.0	1.07
	NL-DPMHSS	(α, β)	$(1.0, 1.0)$	$(0.7, 1.1)$	$(0.5, 1.1)$
	NL-TTSCSP	(α, β)	$(1.0, 1.0)$	$(1.1, 0.7)$	$(1.2, 0.2)$
	NL-RTTSCSP	(α, β, ω)	$(0.9, 1.1, 0.92)$	$(1.1, 0.8, 0.94)$	$(1.2, 0.2, 1.07)$
N = 80	NL-LPMHSS	α	1.02	1.15	1.01
	NL-DPMHSS	(α, β)	$(1.0, 1.0)$	$(0.8, 1.1)$	$(0.7, 1.4)$
	NL-TTSCSP	(α, β)	$(1.0, 1.0)$	$(1.1, 0.7)$	$(1.3, 0.2)$
	NL-RTTSCSP	(α, β, ω)	$(1.0, 1.0, 0.95)$	$(1.0, 0.7, 1.01)$	$(1.3, 0.2, 1.07)$
N = 100	NL-LPMHSS	α	1.05	1.24	0.98
	NL-DPMHSS	(α, β)	$(1.0, 1.0)$	$(0.9, 0.8)$	$(0.8, 1.2)$
	NL-TTSCSP	(α, β)	$(1.0, 1.0)$	$(1.0, 0.7)$	$(1.1, 0.2)$
	NL-RTTSCSP	(α, β, ω)	$(1.0, 1.0, 0.95)$	$(1.1, 0.8, 0.97)$	$(1.2, 0.2, 0.95)$

Table 4: Example 5.1. Experimental results, $N = 50$.

Method	Results	$\eta = 0.1$			$\eta = 0.2$		
		$q = 1$	$q = 10$	$q = 100$	$q = 1$	$q = 10$	$q = 100$
P-LPMHSS	IT_{int}	6.3333	6.3333	6	4.6250	4.6250	4.3750
	IT_{out}	6	6	6	8	8	8
	IT	38	38	36	37	37	35
	$CPU(s)$	3.8367	3.7518	3.5300	3.6179	3.7333	3.5973
P-DPMHSS	IT_{int}	4	4	3.6667	3	3	3
	IT_{out}	6	6	6	7	8	7
	IT	24	24	22	21	24	21
	$CPU(s)$	3.1086	3.1397	2.8732	3.0925	3.3940	2.9684
P-TTSCSP	IT_{int}	1	1	1	1	1	1
	IT_{out}	5	5	5	5	5	5
	IT	5	5	5	5	5	5
	$CPU(s)$	1.5388	1.0454	1.0038	0.9458	0.9706	0.9590
P-RTTSCSP	IT_{int}	1	1	1	1	1	1
	IT_{out}	5	4	5	5	4	5
	IT	5	4	5	5	4	5
	$CPU(s)$	0.8366	0.6879	0.8399	0.8817	0.6785	0.8612

Table 5: Example 5.1. Experimental results, $N = 80$.

Method	Results	$\eta = 0.1$			$\eta = 0.2$		
		$q = 1$	$q = 10$	$q = 100$	$q = 1$	$q = 10$	$q = 100$
P-LPMHSS	IT_{int}	6.3333	6.3333	6.3333	4.6250	4.6250	4.5000
	IT_{out}	6	6	6	8	8	8
	IT	38	38	38	37	37	36
	$CPU(s)$	26.0478	24.4820	23.5740	24.0914	24.0309	22.8621
P-DPMHSS	IT_{int}	4	4	4	3	3	3
	IT_{out}	5	6	5	7	7	7
	IT	20	24	20	21	21	21
	$CPU(s)$	15.5405	18.6687	15.7676	17.0327	16.6299	16.7125
P-TTSCSP	IT_{int}	1	1	1	1	1	1
	IT_{out}	5	5	5	5	5	5
	IT	5	5	5	5	5	5
	$CPU(s)$	7.2819	7.1629	7.0755	7.8725	6.9885	7.2249
P-RTTSCSP	IT_{int}	1	1	1	1	1	1
	IT_{out}	5	4	5	5	4	5
	IT	5	4	5	5	4	5
	$CPU(s)$	6.0021	4.5820	5.7505	5.8577	4.6065	5.8432

Table 6: Example 5.1. Experimental results, $N = 100$.

Method	Results	$\eta = 0.1$			$\eta = 0.2$		
		$q = 1$	$q = 10$	$q = 100$	$q = 1$	$q = 10$	$q = 100$
P-LPMHSS	IT_{int}	6.3333	6.3333	6.1667	4.5000	4.5000	4.5000
	IT_{out}	6	6	6	8	8	8
	IT	38	38	37	36	36	36
	$CPU(s)$	53.0981	60.7824	64.9580	58.4438	67.8453	65.6291
P-DPMHSS	IT_{int}	4	4	3.8333	3	3	3
	IT_{out}	6	6	6	7	8	7
	IT	24	24	23	21	24	21
	$CPU(s)$	56.4398	53.0125	48.9155	47.3042	54.6676	47.7330
P-TTSCSP	IT_{int}	1	1	1	1	1	1
	IT_{out}	5	5	5	5	5	5
	IT	5	5	5	5	5	5
	$CPU(s)$	34.4898	23.3807	25.4346	28.7855	21.5386	24.5502
P-RTTSCSP	IT_{int}	1	1	1	1	1	1
	IT_{out}	5	4	5	5	4	5
	IT	5	4	5	5	4	5
	$CPU(s)$	15.5631	12.7213	16.7814	18.1893	12.9676	18.7534

Table 7: Example 5.1. Numerical results of the nonlinear-like methods.

	Method	Results	$q = 1$	$q = 10$	$q = 100$
$N = 50$	NL-LPMHSS	IT	38	37	34
		CPU	3.9312	3.7292	3.3958
	NL-DPMHSS	IT	20	21	20
		CPU	1.5535	1.7274	1.6513
	NL-TTSCSP	IT	3	3	5
		CPU	0.3371	0.3842	0.6599
	NL-RTTSCSP	IT	3	3	5
		CPU	0.2808	0.2803	0.5796
$N = 80$	NL-LPMHSS	IT	38	37	35
		CPU	22.4534	23.9697	22.1354
	NL-DPMHSS	IT	20	20	19
		CPU	11.5838	11.9269	11.7739
	NL-TTSCSP	IT	3	3	5
		CPU	2.8161	2.7186	4.4193
	NL-RTTSCSP	IT	3	3	5
		CPU	2.2851	2.3699	3.7452
$N = 100$	NL-LPMHSS	IT	38	37	35
		CPU	62.3784	61.2079	56.7236
	NL-DPMHSS	IT	20	21	22
		CPU	34.3859	35.8757	38.4331
	NL-TTSCSP	IT	3	3	5
		CPU	10.3123	7.3587	15.0742
	NL-RTTSCSP	IT	3	3	5
		CPU	6.7434	5.7396	10.0318

method. Also, it is a certitude that the performance of nonlinear RTTSCSP-like method is preferable than Picard-RTTSCSP method.

Example 5.2. Let Ω be the same domain as in Example 5.1. Consider the two-dimensional nonlinear convection-diffusion equation

$$\begin{aligned} & u_t-(\alpha_1+i\beta_1)(u_{xx}+u_{yy})+qu \\ & =(\alpha_2+i\beta_2)ue^u+\sin\sqrt{1+u_x^2+u_y^2} && \text{in } (0,1]\times\Omega, \\ & u(0,x,y)=u_0(x,y) && \text{in } \Omega, \\ & u(t,x,y)=0 && \text{on } (0,1]\times\partial\Omega, \end{aligned} \tag{5.2}$$

where $\alpha_1=\beta_1=1, \alpha_2=\beta_2=0.5$, and q is a normal constant that controls the amplitude of the reaction term. The corresponding weakly nonlinear system can be obtained by discretizing the Eq. (5.2) at each time step by an implicit scheme on the equidistant grid $\Delta t=h=1/(N+1)$,

$$F(x)=Ax-\phi(x)=0,$$

where

$$A=h(1+q\Delta t)I_n+(\alpha_1+i\beta_1)(A_N\otimes I_N+I_N\otimes A_N).$$

A_N is the same tridiagonal matrix as in Example 5.1 and $n=N\times N$.

In actual computations, we adopt the same method as in Example 5.1 to select parameters α,β,ω for the iteration methods — cf. Tables 8-9. The corresponding numerical

Table 8: Experimental optimal values α,β,ω for each Picard-based methods.

	Method	Parameter	$\eta=0.1$		$\eta=0.2$	
			$q=1$	$q=10$	$q=1$	$q=10$
$N=60$	P-LPMHSS	α	1.3	1.2	1.1	1.2
	P-DPMHSS	(α,β)	$(0.7,1.7)$	$(0.8,1.9)$	$(4.3,0.9)$	$(0.8,1.1)$
	P-TTSCSP	(α,β)	$(1.4,0.4)$	$(1.4,0.4)$	$(4.3,0.9)$	$(2.9,0.8)$
	P-RTTSCSP	(α,β,ω)	$(1.4,0.4,1.09)$	$(1.3,0.3,0.93)$	$(4.3,0.9,0.96)$	$(4.6,0.8,1.05)$
$N=80$	P-LPMHSS	α	1.1	1.2	1.3	1.2
	P-DPMHSS	(α,β)	$(0.8,1.7)$	$(0.8,1.8)$	$(0.6,1.5)$	$(0.6,1.6)$
	P-TTSCSP	(α,β)	$(4.0,0.9)$	$(4.0,0.8)$	$(4.5,0.8)$	$(4.6,0.8)$
	P-RTTSCSP	(α,β,ω)	$(3.9,0.9,1.04)$	$(4.0,0.8,0.95)$	$(4.5,0.8,0.96)$	$(4.5,0.8,1.09)$

Table 9: Experimental optimal values α,β,ω for nonlinear like methods.

Method	Parameter	$N=60$		$N=80$	
		$q=1$	$q=10$	$q=1$	$q=10$
NL-LPMHSS	α	1.1	0.8	1.0	1.2
NL-DPMHSS	(α,β)	$(0.8,1.5)$	$(0.6,1.2)$	$(0.7,1.5)$	$(0.6,1.3)$
NL-TTSCSP	(α,β)	$(0.7,0.9)$	$(1.0,0.6)$	$(1.0,1.0)$	$(1.6,0.9)$
NL-RTTSCSP	(α,β,ω)	$(1.0,1.0,0.97)$	$(1.1,0.7,0.97)$	$(1.0,1.0,0.99)$	$(1.5,1.0,1.03)$

Table 10: Example 5.2. Experimental results, $N = 60$.

Method	Results	$\eta = 0.1$		$\eta = 0.2$	
		$q = 1$	$q = 10$	$q = 1$	$q = 10$
P-LPMHSS	IT_{int}	6.3333	6.3333	4.6250	4.6250
	IT_{out}	6	6	8	8
	IT	38	48	37	37
	$CPU(s)$	12.7489	8.7509	8.9466	8.3174
P-DPMHSS	IT_{int}	4	4	2.8571	3
	IT_{out}	5	6	7	8
	IT	20	24	20	24
	$CPU(s)$	4.7830	4.9871	5.3801	6.4159
P-TTSCSP	IT_{int}	1	1	1	1
	IT_{out}	5	5	5	5
	IT	5	5	5	5
	$CPU(s)$	1.9607	1.9866	1.8496	1.8771
P-RTTSCSP	IT_{int}	1	1	1	1
	IT_{out}	5	5	5	5
	IT	5	5	5	5
	$CPU(s)$	1.7814	1.8690	1.8171	1.7913

Table 11: Example 5.2 Experimental results, $N = 80$.

Method	Results	$\eta = 0.1$		$\eta = 0.2$	
		$q = 1$	$q = 10$	$q = 1$	$q = 10$
P-LPMHSS	IT_{int}	6.3333	6.3333	4.6250	4.5000
	IT_{out}	6	6	8	8
	IT	38	38	37	36
	$CPU(s)$	25.1320	24.9167	25.7081	24.8792
P-DPMHSS	IT_{int}	4	4	2.8571	2.8571
	IT_{out}	5	5	7	7
	IT	20	20	20	20
	$CPU(s)$	13.8127	14.1710	14.8526	15.5286
P-TTSCSP	IT_{int}	1	1	1	1
	IT_{out}	5	5	5	5
	IT	5	5	5	5
	$CPU(s)$	5.9176	11.4024	12.5763	12.0625
P-RTTSCSP	IT_{int}	1	1	1	1
	IT_{out}	5	5	5	5
	IT	5	5	5	5
	$CPU(s)$	5.6902	8.0098	6.1024	6.0622

results are shown in Tables 10-12 which inform iteration times and CPU time with the inner tolerance $\eta = 0.1, 0.2$. Picard-RTTSCSP method is obviously superior to the Picard-DPMHSS and Picard-LPMHSS methods in respect to computing time and iteration times — cf. Tables 10 and 11. Note that in terms of iteration and calculation time, our method is

Table 12: Example 5.2. Numerical results of the nonlinear-like methods.

Method	Results	$N=60$		$N=80$	
		$q=1$	$q=10$	$q=1$	$q=10$
NL-LPMHSS	IT	38	37	38	37
	$CPU(s)$	6.6737	7.5065	24.2613	26.0405
NL-DPMHSS	IT	21	20	21	21
	$CPU(s)$	3.1742	2.9395	10.9286	13.1256
NL-TTSCSP	IT	3	3	2	3
	$CPU(s)$	0.7233	0.7589	1.6200	2.6363
NL-RTTSCSP	IT	2	3	2	3
	$CPU(s)$	0.4961	0.7567	1.5267	2.3454

a dozen times faster than the two methods mentioned. Moreover, CPU time in the Picard-TTSCSP is bigger than in our relaxed iteration method, which illustrates the efficiency of the Picard-RTTSCSP method.

Remarkably, the nonlinear RTTSCSP-like method is superior to the other methods — cf. Table 12. In this two-dimensional nonlinear problem, the nonlinear RTTSCSP-like method saves computation time and iteration steps compared with the other existing methods.

6. Conclusions

A relaxed TTSCSP iteration method for complex linear system is constructed by introducing a relaxation parameter to the TTSCSP iteration method. For the complex weakly nonlinear equation, the Picard-RTTSCSP and nonlinear RTTSCSP-like iteration schemes are proposed and their convergence is studied. Optimal iterative parameters to minimize the upper bound of spectral radius are derived. Examples show that the proposed iteration methods are achievable and effective. They perform better than the Picard-DPMHSS, Picard-LPMHSS, Picard-TTSCSP, nonlinear DPMHSS-like, nonlinear LPMHSS-like and nonlinear TTSCSP-like methods in terms of calculation time and iteration numbers. Besides, the nonlinear RTTSCSP-like method outperforms the Picard-RTTSCSP method.

Acknowledgments

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

References

[1] H.-B. An and Z.-Z. Bai, *A globally convergent Newton-GMRES method for large sparse systems of nonlinear equations*, Appl. Numer. Math. **57**, 235–252 (2007).

[2] A. Amiri, M.T. Darvishi, A. Cordero and J.R. Torregrosa, *An efficient iterative method based on two-stage splitting methods to solve weakly nonlinear systems*, Mathematics-Basel **7**, 815 (2019).

[3] I.K. Argyros, *Affine invariant local convergence theorems for inexact Newton-like methods*, J. Appl. Math. Comput. **6**, 291–304 (1999).

[4] I.K. Argyros, S. Georg and A. Magrenan, *Improved semi-local convergence of the Newton-HSS method for solving large systems of equations*, Appl. Math. Lett. **98**, 29–35 (2019).

[5] Z.-Z. Bai, *The monotone convergence rate of the parallel nonlinear AOR method*, Comput. Math. Appl. **31**(7), 1–8 (1996).

[6] Z.-Z. Bai, *A class of two-stage iterative methods for systems of weakly nonlinear equations*, Numer. Algorithms **14**, 295–319 (1997).

[7] Z.-Z. Bai, *On the convergence of parallel chaotic nonlinear multisplitting Newton-type methods*, J. Comput. Appl. Math. **80**, 317–334 (1997).

[8] Z.-Z. Bai and X.-P. Guo, *On Newton-HSS methods for systems of nonlinear equations with positive-definite Jacobian matrices*, J. Comput. Math. **28**, 235–260 (2010).

[9] Z.-Z. Bai, H.-M. Huang and M.K. Ng, *On preconditioned iterative methods for Burgers equations*, Form. Method. Syst. Des. **42**, 1–2 (2012).

[10] Z.-Z. Bai and X. Yang, *On HSS-based iteration methods for weakly nonlinear systems*, Appl. Numer. Math. **59**, 2923–2936 (2009).

[11] M.-H. Chen, W. Dou and Q.-B. Wu, *DPMHSS-based iteration methods for solving weakly nonlinear systems with complex coefficient matrices*, Appl. Numer. Math. **146**, 328–341 (2019).

[12] M.-H. Chen and Q.-B. Wu, *Modified Newton-MDPMHSS method for solving nonlinear systems with block two-by-two complex symmetric Jacobian matrices*, Numer. Algorithms **80**, 355–375 (2019).

[13] M.-H. Chen, Q.-B. Wu, Q. Cao and R.-F. Lin, *An efficient iterative approach to large sparse nonlinear systems with non-Hermitian Jacobian matrices*, East Asian J. Appl. Math. **11**, 349-368 (2021).

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

[15] P. Deuflhard and G. Heindl, *Affine invariant convergence theorems for Newton's method and extensions to related methods*, SIAM. J. Numer. Anal. **16**, 1–10 (1979).

[16] H.-T. Fan, X.-Y. Zhu and B. Zheng, *A generalized modified SOR-like method for the singular saddle point problems*, J. Appl. Math. Comput. **54**, 199–212 (2017).

[17] Y.-Y. Feng, Q.-B. Wu and Z.-W. Xie, *Lopsided DSS iteration method for solving complex Sylvester matrix equation*, Comput. Appl. Math. **40**(7), 1–19 (2021).

[18] M.-J. Jiang and X.-P. Guo, *A new parameter-free method for Toeplitz systems of weakly nonlinear equations*, J. Comput. Appl. Math. **393**, 113505 (2021).

[19] C.-X. Li and S.-L. Wu, *On LPMHSS-based iteration methods for a class of weakly nonlinear systems*, Comput. Appl. Math. **37**(2), 1232–1249 (2016).

[20] R.-J. Liu and L. Dong, *Nonmonotone smoothing inexact Newton method for the nonlinear complementarity problem*, J. Appl. Math. Comput. **51**, 659–674 (2016).

[21] J.M. Ortega and W.C. Rheinbold, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press (1970).

[22] Z.-N. Pu and M.-Z. Zhu, *A class of iteration methods based on the generalized preconditioned Hermitian and skew-Hermitian splitting for weakly nonlinear systems*, J. Comput. Appl. Math. **250**, 16–27 (2013).

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

[24] T.S. Siahkolaei and D.K. Salkuyeh, *Two-parameter TSCSP method for solving complex symmetric system of linear equations*, Calcolo **55**, 8 (2018).

[25] T.S. Siahkolaei and D.K. Salkuyeh, *TTSCSP-based iteration methods for complex weakly nonlinear systems*, East Asian J. Appl. Math. **10**, 1–21 (2020).

[26] J.-Y. Wang and D.-T. Zhu, *The inexact-Newton via GMRES subspace method without line search technique for solving symmetric nonlinear equations*, Appl. Numer. Math. **110**, 174–189 (2016).

[27] H.-T. Wu, X. Qi and X.-Y. Xiao, *On GSOR-based iteration methods for solving weakly nonlinear systems with complex symmetric coefficient matrices*, J. Appl. Math. Comput. **68**, 601–621 (2022).

[28] F. Xie, R.-F. Lin and Q.-B. Wu, *Modified Newton-DSS method for solving a class of systems of nonlinear equations with complex symmetric jacobian matrices*, Numer. Algorithms **85**, 951–975 (2019).

[29] F. Xie, Q.-B. Wu and D.-F. Dai, *Modified Newton–SHSS method for a class of systems of nonlinear equations*, Comput. Appl. Math. **38**(1), 19–43 2019.

[30] M.-L. Zeng and G.-F. Zhang, *A class of preconditioned TGHSS-based iteration methods for weakly nonlinear systems*, East Asian J. Appl. Math. **6**, 367–383 (2016).

[31] L. Zhang, Q.-B. Wu, M.-H. Chen and R.-F. Lin, *Two new effective iteration methods for nonlinear systems with complex symmetric Jacobian matrices*, Comput. Appl. Math. **40**(3), 1–27 (2021).

[32] Z. Zheng, F.-L. Huang and Y.-C. Peng, *Double-step scale splitting iteration method for a class of complex symmetric linear systems*, Appl. Math. Lett. **73**, 91–97 (2017).

[33] M.-Z. Zhu, *Modified iteration methods based on the asymmetric HSS for weakly nonlinear systems*, J. Comput. Anal. Appl. **15**, 188–195 (2013).

[34] M.-Z. Zhu and G.-F. Zhang, *On CSCS-based iteration methods for Toeplitz system of weakly nonlinear equations*, J. Comput. Appl. Math. **235**, 5095–5104 (2011).