On TTSCSP-based iteration methods for complex weakly nonlinear systems

Tahereh Salimi Siahkoalaei and Davod Khojasteh Salkuyeh

Faculty of Mathematical Sciences, University of Guilan, Rasht, Iran
E-mails: salimi_tahereh@phd.guilan.ac.ir, khojasteh@guilan.ac.ir

Abstract. For a class of weakly system of nonlinear equations, we present the Picard-TTSCSP and the nonlinear TTSCSP-like iteration methods. In these methods we need to solve two subsystems of constant positive definite coefficient matrices. Therefore, we solve these two subsystems by the conjugate gradient method and call the resulting methods inexact Picard-TTSCSP and nonlinear inexact TTSCSP-like. We establish local convergence theorem under suitable conditions. Numerical experiments demonstrate the effectiveness and feasibility of the new methods.

AMS subject classifications: 65F10, 65F50, 65W05.
Keywords: weakly nonlinear systems, TTSCSP iteration method, inner/outer iteration scheme.

1 Introduction

Let $A \in \mathbb{C}^{n\times n}$ be a large, sparse, complex symmetric matrix and $\phi : \mathbb{D} \subset \mathbb{C}^n \longrightarrow \mathbb{C}^n$ be a continuously differentiable function. Consider iterative solutions of systems of weakly nonlinear equations of the form

$$Au = \phi(u), \text{ or equivalently } F(u) = Au - \phi(u) = 0, \tag{1}$$

where $A = W + iT$, $W \in \mathbb{R}^{n\times n}$ and $T \in \mathbb{R}^{n\times n}$ are symmetric positive definite and symmetric positive semidefinite, respectively. The system of nonlinear equations (1) is said to be weakly nonlinear if the linear term Au is strongly dominant over the nonlinear term $\phi(u)$ in certain norm [14, 31]. The system of weakly nonlinear equations (1) may arise in many areas of scientific computing and engineering applications and in particular in discretization of certain nonlinear partial differential equations [4, 8, 9, 11, 24], in collocation approximations of nonlinear integral equation [28], in saddle point problems from image processing [12, 15] and more applications [20].

In the past, many authors studied a large variety of efficient methods for solving the system of nonlinear equations (1), such as the Newton iteration method [1, 17]. At each iteration step, the Newton method requires Jacobian matrix, which are very costly and complicated in actual application. In order to overcome this disadvantage and improve the efficiently of the Newton iteration method, many variants in terms of approximate, quasi-update, inner/outer or inexact Newton methods have been established and analyzed [1, 3, 4, 6, 11, 13, 14, 16, 18, 27].

Recently, Bai et al. in [10], by making use of the Hermitian and skew-Hermitian splitting (HSS) iteration method [7] as the inner solver for the Newton method, established a class of Newton-HSS methods for large sparse system of nonlinear equations. Bai in [4], presented a class of sequential two-stage iteration methods for solving the nonlinear equations (1). See more [6, 13]. The Picard-HSS and the nonlinear HSS-like iteration methods are established in [14]. Zhu et al. in [32] presented the Picard-CSCS and the nonlinear CSCS-like iteration methods for weakly nonlinear systems. By making use of the MHSS iteration in [5] as the inner solver for the Picard method, Yang et al. in [29] established the Picard-MHSS and the nonlinear MHSS-like iteration methods. This method can be summarized as follows.

The Picard-MHSS iteration method: `Given an initial guess` $u^{(0)} \in \mathbb{D}$ `and sequence` $\{l_k\}_{k=0}^{\infty}$ `of positive integers, compute` $u^{(k+1)}$ `for` $k = 0, 1, 2, \ldots$ `using the following iteration scheme until` $\{u^{(k)}\}$ `satisfies the stopping criterion:`

(a) `Set` $u^{(k,0)} := u^{(k)}$.

(b) `For` $l = 0, 1, 2, \ldots, l_k-1$, `solve the following linear systems to obtain` $u^{(k,l+1)}$:
$$\begin{cases}(\alpha I + W)\ u^{(k,l+\frac{1}{2})} = (\alpha I - iT)u^{(k,l)} +\ \phi(u^{(k)}),\\ (\alpha I + T)\ u^{(k,l+1)} = (\alpha I + iW)u^{(k,l+\frac{1}{2})} - i\ \phi(u^{(k)}),\end{cases}$$
`where` α `and` β `are given positive constants.`

(c) `Set` $u^{(k+1)} := u^{(k,l_k)}$.

Similar to the work of [14, 29], Li et al. in [23] applied the lopsided PMHSS (LPMHSS) iteration [22] as the inner solver for the Picard method and presented the Picard-LPMHSS and the nonlinear LPMHSS-like iteration methods. The Picard-LPMHSS can be algorithmically described as follows.

The Picard-LPMHSS iteration method: `Given an initial guess` $u^{(0)} \in \mathbb{D}$ `and sequence` $\{l_k\}_{k=0}^{\infty}$ `of positive integers, compute` $u^{(k+1)}$ `for` $k = 0, 1, 2, \ldots$ `using the following iteration scheme until` $\{u^{(k)}\}$ `satisfies the stopping criterion:`

(a) `Set` $u^{(k,0)} := u^{(k)}$.

(b) `For` $l = 0, 1, 2, \ldots, l_k-1$, `solve the following linear systems to obtain` $u^{(k,l+1)}$:
$$\begin{cases}W\ u^{(k,l+\frac{1}{2})} = -iT\ u^{(k,l)} + \phi(u^{(k)}),\\ (\alpha P + T)\ u^{(k,l+1)} = (\alpha P + iW)u^{(k,l+\frac{1}{2})} - i\phi(u^{(k)}),\end{cases}$$
`where` α `is a given positive constants and` P `is a given symmetric positive definite matrix.`

(c) `Set` $u^{(k+1)} := u^{(k,l_k)}$.

More recently, Salkuyeh and Siahkolaei in [25] proposed the two-parameter two-step scale-splitting (TTSCSP) method for solving the system of linear equations $(W+iT)x = b$

where $W \in \mathbb{R}^{n\times n}$ is symmetric positive definite, $T \in \mathbb{R}^{n\times n}$ is symmetric positive semidefinite and $b \in \mathbb{C}^n$. The TTSCSP method reduces to two-step scale-splitting (TSCSP) method presented by Salkuyeh in [26]. In this paper, using the TTSCSP method we establish the Picard-TTSCSP and the nonlinear TTSCSP-like iteration methods as inner solver in the Picard iteration for solving the large scale system of weakly nonlinear equations (1).

Throughout this paper, we use $\|.\|$ for 2-norm of a vector or a matrix. The spectrum and spectral radius of a matrix are denoted by $\sigma(.)$ and $\rho(.)$, respectively. We also use the notation $\otimes$ for the Kronecker product.

The rest of this paper is organized as follows. In Section 2 we give a brief description of the TTSCSP method for symmetric linear equations. The Picard-TTSCSP method is established in Section 3 and the nonlinear TTSCSP-like iteration method is presented in Section 4. We introduce the inexact version of the Picard-TTSCSP and the nonlinear TTSCSP-like iteration method in Section 5. Numerical experiments are given to illustrate the effectiveness of the new iteration methods in Section 6. Finally, in Section 7, we draw a brief conclusions.

2 The TTSCSP iteration method

When $\phi : \mathbb{D} \subset \mathbb{C}^n \longrightarrow \mathbb{C}^n$ is a constant vector, i.e., $\phi(u) = b$, the system of weakly nonlinear equations (1) changes to the system of linear equations

$$Au = b, \tag{2}$$

where $A \in \mathbb{C}^{n\times n}$ and $u, b \in \mathbb{C}^n$. For solving the linear system (2), many methods presented. see [5, 25, 26, 19, 21, 22, 31]. Salkuyeh and Siahkolaei in [25] efficiently split the coefficient matrix A and presented the TTSCSP method and investigated its convergence properties under some conditions. This method can be written as follows.

The TTSCSP iteration method: `Let` $u^{(0)} \in \mathbb{C}^n$ `be an initial guess. For` $k = 0, 1, 2, \ldots,$ `until` $\{u^{(k)}\}$ `converges, compute` $u^{(k+1)}$ `according to the following sequence`

$$\begin{cases} (\alpha W + T)u^{(k+\frac{1}{2})} = i(W - \alpha T)u^{(k)} + (\alpha - i)b, \\ (W + \beta T)u^{(k+1)} = i(\beta W - T)u^{(k+\frac{1}{2})} + (1 - \beta i)b, \end{cases} \tag{3}$$

`where` α `and` β `are positive numbers.`

The TTSCSP method becomes TSCSP method, when $\alpha = \beta$. Salkuyeh in [26] showed that if both of the matrices W and T are symmetric positive definite, then for any positive α the TSCSP method is convergent. The TTSCSP iteration method can be reformulated as the matrix-vector form

$$\begin{aligned} u^{(k+1)} &= \mathcal{G}(\alpha, \beta)\ u^{(k)} \ + \ \mathcal{C}(\alpha, \beta) \\ &= \ \mathcal{G}(\alpha, \beta)^{k+1}\ u^{(0)} \ + \ \sum_{j=0}^{k} \mathcal{G}(\alpha, \beta)^{j}\ \mathcal{C}(\alpha, \beta)\ b, \end{aligned}$$

where

$$\mathcal{G}(\alpha,\beta) = (W+\beta T)^{-1}(T-\beta W)(\alpha W+T)^{-1}(W-\alpha T),$$

and

$$\mathcal{C}(\alpha,\beta) = (\alpha+\beta)(W+\beta T)^{-1}(W-iT)(\alpha W+T)^{-1}.$$

Setting

$$\begin{aligned} M &= \frac{1}{\alpha+\beta}(\alpha W+T)(W-iT)^{-1}(W+\beta T), \\ N &= \frac{1}{\alpha+\beta}(T-\beta W)(W-iT)^{-1}(W-\alpha T), \end{aligned}$$

we have $A = M - N$ and $\mathcal{G}_{\alpha,\beta} = M^{-1}N$, which shows that $\mathcal{G}_{\alpha,\beta}$ is the iteration matrix of the TTSCSP method.

3 The Picard-TTSCSP method

When the linear term Au and the nonlinear term $\phi(u)$ are well-separated and the former is strongly dominant over the latter, we can use the Picard iteration method,

$$Au^{(k+1)} = \phi(u^{(k)}), \quad k = 0, 1, 2, \ldots, \tag{4}$$

for solving the system (1) (see [1, 2, 4, 24]). At each step of the Picard iteration, we should solve a linear system of equations with the coefficient matrix A. Based on the TTSCSP iteration method in [25], we find $u^{(k+1)}$ in each step of Picard and called Picard-TTSCSP method. Therefore, the Picard-TTSCSP iteration method is described as follows.

The Picard-TTSCSP iteration method: Let $\phi : \mathbb{D} \subset \mathbb{C}^n \longrightarrow \mathbb{C}^n$ be a continuously differentiable function and $A = W + iT$ with W being symmetric positive definite and T be symmetric positive semidefinite. Given an initial guess $u^{(0)} \in \mathbb{D}$ and sequence $\{l_k\}_{k=0}^{\infty}$ of positive integers, compute $u^{(k+1)}$ for $k = 0, 1, 2, \ldots$ using the following iteration scheme until $\{u^{(k)}\}$ satisfies the stopping criterion:

(a) Set $u^{(k,0)} := u^{(k)}$.

(b) For $l = 0, 1, 2, \ldots, l_k - 1$, solve the following linear systems to obtain $u^{(k,l+1)}$:

$$\begin{cases} (\alpha W + T)\ u^{(k,l+\frac{1}{2})} = i(W-\alpha T)u^{(k,l)} + (\alpha - i)\ \phi(u^{(k)}), \\ (W+\beta T)\ u^{(k,l+1)} = i(\beta W - T)u^{(k,l+\frac{1}{2})} + (1-\beta i)\ \phi(u^{(k)}), \end{cases}$$

where α and β are given positive constants.

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

Furthermore, we can also use the TSCSP iteration method as the inner solver for finding the $u^{(k+1)}$ of the Picard iteration (4). This results in the Picard-TSCSP iteration method. The Picard-TTSCSP iteration method becomes the Picard-TSCSP iteration method, when $\alpha = \beta$. In the Picard-TTSCP and the nonlinear TTSCSP-like, the coefficient matrices of the two linear subsystems are symmetric positive definite. Therefore, the two subsystems can be solved exactly using the Cholesky factorization of the coefficient matrices or inexactly by the conjugate gradient (CG) method.

Based on the Picard-TTSCSP iteration, $u^{(k+1)}$ can be derived:

$$u^{(k+1)} = \mathcal{G}(\alpha,\beta)^{l_k}\, u^{(k)} + \sum_{j=0}^{l_k-1} \mathcal{G}(\alpha,\beta)^j\, \mathcal{C}(\alpha,\beta)\, \phi(u^{(k)}), \qquad k = 0,1,\ldots. \tag{5}$$

Suppose that the vector $u^* \in \mathbb{D}$ is the solution of the system (1). It is easy to see that

$$u^* = \mathcal{G}(\alpha,\beta)^{l_k}\, u^* + \sum_{j=0}^{l_k-1} \mathcal{G}(\alpha,\beta)^j\, \mathcal{C}(\alpha,\beta)\, \phi(u^*), \qquad k = 0,1,\ldots. \tag{6}$$

Subtracting both sides of (6) from those of Eq. (5) gives

$$u^{(k+1)} - u^* = \mathcal{G}(\alpha,\beta)^{l_k}\, (u^{(k)} - u^*) + \sum_{j=0}^{l_k-1} \mathcal{G}(\alpha,\beta)^j\, \mathcal{C}(\alpha,\beta)\, [\phi(u^k) - \phi(u^*)].$$

In continuation, using Theorem 3.1 in [14], we study the local convergence theory for the Picard-TTSCSP iteration method.

Theorem 1. *Let $\phi : \mathbb{D} \subset \mathbb{C}^n \longrightarrow \mathbb{C}^n$ be G-differentiable on an open neighborhood $\mathbb{N}_0 \subset \mathbb{D}$ of a point $u^* \in \mathbb{D}$ at which $\phi'(u^*)$ is continuous and $F(u^*) := Au^* - \phi(u^*) = 0$. Suppose $A = W + iT$ with W and T are symmetric positive definite and symmetric positive semidefinite, respectively. Denote by*

$$\theta(\alpha,\beta) = \parallel \mathcal{G}(\alpha,\beta) \parallel, \quad æ = \parallel A^{-1} \parallel, \quad \eta = \parallel A^{-1}\phi'(u^*) \parallel . \tag{7}$$

Then there exists an open neighborhood $\mathbb{N} \subset \mathbb{N}_0$ of u^* such that for any $u^{(0)} \in \mathbb{N}$ and any sequence of positive integers l_k, $k = 0,1,2,\ldots$, the iteration sequence $\{u^{(k)}\}_{k=0}^{\infty}$ generated by the Picard-TTSCSP iteration method is well-defined and converges to u^*, provided $\eta < 1$ and $l_0 \geqslant \lfloor \ln\big((\frac{1-\eta}{1+\eta})/\ln(\theta(\alpha,\beta))\big)\rfloor$, where $\lfloor . \rfloor$ denotes the smallest integer no less than the corresponding real number. Moreover, it holds that

$$\limsup_{k\to\infty} \parallel u^{(k)} - u^* \parallel^{\frac{1}{k}} \leqslant \eta + (1+\eta)\theta(\alpha,\beta)^{l_0} \quad \text{with} \quad l_0 = \liminf_{k\to\infty} l_k; \tag{8}$$

in particular, if $\lim_{k\to\infty} l_k = \infty$, then the convergence rate is R-linear, with the R-factor being at most η, i.e.,

$$\limsup_{k\to\infty} \parallel u^k - u^* \parallel^{\frac{1}{k}} \leqslant \eta. \tag{9}$$

Proof. The proof uses arguments similar to those in the proof of the convergence theorem of the Picard-HSS iteration method in [14]. □

Theorem 1 shows that the convergence rate of the Picard-TTSCSP method depends on the $\theta(\alpha,\beta)$ and η. Therefore, small $\theta(\alpha,\beta)$ and η will lead to fast convergence of the Picard-TTSCSP iteration.

4 The nonlinear TTSCSP-like method

The main drawback of the Picard-TTSCSP iteration method is that the numbers of the inner iteration steps $l_k, k = 0, 1, 2, \ldots,$ are often problem-dependent and difficult to be determined in actual computations. To overcome this disadvantage, based on the nonlinear fixed-point equations

$$(\alpha W + T)u = i(W - \alpha T)u + (\alpha - i)\ \phi(u),$$

and

$$(W + \beta T)u = i(\beta W - T)u + (1 - \beta i)\ \phi(u),$$

we propose the following nonlinear TTSCSP-like iteration method.

The nonlinear TTSCSP-like method: Let $\phi : \mathbb{D} \subset \mathbb{C}^n \longrightarrow \mathbb{C}^n$ be a continuously differentiable function and $A = W + iT$ with W being symmetric positive definite and T being symmetric positive semidefinite. Given an initial guess $u^{(0)} \in \mathbb{D}$, compute $u^{(k+1)}$ for $k = 0, 1, 2, \ldots$ using the following iteration scheme until $\{u^{(k)}\}$ satisfies the stopping criterion:

$$\begin{cases} (\alpha W + T)\ u^{(k+\frac{1}{2})} = i(W - \alpha T)u^{(k)} + (\alpha - i)\ \phi(u^{(k)}), \\ (W + \beta T)\ u^{(k+1)} = i(\beta W - T)u^{(k+\frac{1}{2})} + (1 - \beta i)\ \phi(u^{(k+\frac{1}{2})}), \end{cases} \tag{10}$$

where α and β are positive constants.

In the following we study the convergence of the TTSCSP-like iteration method. We define

$$\begin{cases} F(u) = (\alpha W + T)^{-1}[i(W - \alpha T)u + (\alpha - i)\ \phi(u)], \\ V(u) = (W + \beta T)^{-1}[i(\beta W - T)u + (1 - \beta i)\ \phi(u)], \end{cases} \tag{11}$$

and $\psi(u) = VoF(u) = V(F(u))$. Then, we can rewrite the nonlinear TTSCSP-like iteration into the

$$u^{(k+1)} = \psi(u^{(k)}), \quad k = 0, 1, 2, \ldots.$$

By making use of the Ostrowski theorem (Theorem 10.1.3 in [14]), we know that if $\rho(\psi'(u^*)) < 1$, then u^* is a point of attraction of the nonlinear TTSCSP-like iteration.

Suppose that $u^* \in \mathbb{D}$ is a solution of the system of weakly nonlinear equations (1), we can easily verify the identities

$$F(u^*) = u^*, \quad V(u^*) = u^*,$$

and

$$\begin{cases} F'(u^*) = (\alpha W + T)^{-1}[i(W - \alpha T) + (\alpha - i)\ \phi'(u^*)], \\ V'(u^*) = (W + \beta T)^{-1}[i(\beta W - T) + (1 - \beta i)\ \phi'(u^*)]. \end{cases}$$

Now, by making use of the chain rule, e.g., Theorem 3.1.7 in [24], we have

$$\begin{aligned} \psi'(u^*) &= V'(u^*)F'(u^*) \\ &= (W + \beta T)^{-1}[i(\beta W - T) + (1 - \beta i)\ \phi'(u)]\ (\alpha W + T)^{-1}[i(W - \alpha T) + (\alpha - i)\ \phi'(u)]. \end{aligned}$$

By summarizing the above results we can state the following theorem.

Theorem 2. *Assume that $\phi : \mathbb{D} \subset \mathbb{C}^n \longrightarrow \mathbb{C}^n$ is F-differentiable at a point $u^* \in \mathbb{D}$ such that $Au^* = \phi(u^*)$. Suppose that $A = W + iT$ with W and T are symmetric positive definite and symmetric positive semidefinite, respectively. Let*

$$\mathcal{G}(\alpha, \beta; u^*) = (W+\beta T)^{-1}[i(\beta W - T) + (1-\beta i)\phi'(u^*)]\ (\alpha W + T)^{-1}[i(W-\alpha T) + (\alpha - i)\ \phi'(u^*)].$$

If $\mathcal{G}(\alpha, \beta; u^*) < 1$, then $u^* \in \mathbb{D}$ is a point of attraction of the nonlinear TTSCSP-like iteration.

Theorem 3. *Assume that the conditions of Theorem 2 are satisfied. Let*

$$\delta = \max\{\| \phi'(u^*)(W+\beta T)^{-1} \|, \| \phi'(u^*)(\alpha W + T)^{-1} \|\},$$
$$a = \max\left\{\left|\frac{1-\alpha\mu_1}{\alpha+\mu_1}\right|, \left|\frac{1-\alpha\mu_n}{\alpha+\mu_n}\right|\right\},$$
$$b = \max\left\{\left|\frac{\beta-\mu_1}{1+\beta\mu_1}\right|, \left|\frac{\beta-\mu_n}{1+\beta\mu_n}\right|\right\},$$

where μ_1 and μ_n are smallest and largest eigenvalues of $W^{-1}T$. Then $\rho(\mathcal{G}(\alpha, \beta, u^*)) < 1$ holds, provided that

$$ab + (rb + pa)\delta + rp\delta^2 < 1, \tag{12}$$

with $r = \sqrt{1+\alpha^2}$ and $p = \sqrt{1+\beta^2}$.

Proof. By simple computations, we have

$$\begin{aligned}(W+\beta T)\mathcal{G}(\alpha, \beta; u^*)(W+\beta T)^{-1} = {} & (W+\beta T)\mathcal{G}(\alpha, \beta)(W+\beta T)^{-1} \\ & + (1+\alpha i)(\beta W - T)(\alpha W + T)^{-1}\phi'(u^*)(W+\beta T)^{-1} \\ & + (\beta + i)\phi'(u^*)(\alpha W + T)^{-1}(W - \alpha T)(W+\beta T)^{-1} \\ & + (1-\beta i)(\alpha - i)\phi'(u^*)(\alpha W + T)^{-1}\phi'(u^*)(W+\beta T)^{-1},\end{aligned}$$

and

$$\begin{aligned}\|\mathcal{G}(\alpha, \beta)\| &= \|(\alpha W + T)^{-1}(W - \alpha T)(W+\beta T)^{-1}(T - \beta W)\| \\ &= \|(\alpha I + S)^{-1}(I - \alpha S)(I + \beta S)^{-1}(\beta I - S)\| \\ &\leqslant \|(\alpha I + S)^{-1}(I - \alpha S)\|\ \|(I+\beta S)^{-1}(\beta I - S)\| \\ &= \max_{\mu\in\sigma(S)}\left\{ \left|\frac{1-\alpha\mu}{\alpha+\mu}\right| \right\}. \max_{\mu\in\sigma(S)}\left\{ \left|\frac{\beta-\mu}{1+\beta\mu}\right| \right\} \\ &= \max\left\{ \left|\frac{1-\alpha\mu_1}{\alpha+\mu_1}\right|, \left|\frac{1-\alpha\mu_n}{\alpha+\mu_n}\right| \right\}. \max\left\{ \left|\frac{\beta-\mu_1}{1+\beta\mu_1}\right|, \left|\frac{\beta-\mu_n}{1+\beta\mu_n}\right|\right\} \\ &= ab.\end{aligned}$$

Hence

$$\begin{aligned}\|\mathcal{G}(\alpha,\beta;u^*)\| &= \|(W+\beta T)\mathcal{G}(\alpha,\beta;u^*)(W+\beta T)^{-1}\| \\ &\leqslant \|(W+\beta T)\mathcal{G}(\alpha,\beta)(W+\beta T)^{-1}\| \\ &\quad + \|(1+\alpha i)(\beta W-T)(\alpha W+T)^{-1}\phi'(u^*)(W+\beta T)^{-1}\| \\ &\quad + \|(\beta+i)\phi'(u^*)(\alpha W+T)^{-1}(W-\alpha T)(W+\beta T)^{-1}\| \\ &\quad + \|(1-\beta i)(\alpha-i)\phi'(u^*)(\alpha W+T)^{-1}\phi'(u^*)(W+\beta T)^{-1}\| \\ &\leqslant \|\mathcal{G}(\alpha,\beta)\| + \|(1+\alpha i)(\beta W-T)(W+\beta T)^{-1}\|\ \|\phi'(u^*)(\alpha W+T)^{-1}\| \\ &\quad + \|(\beta+i)(\alpha W+T)^{-1}(W-\alpha T)\|\ \|\phi'(u^*)(W+\beta T)^{-1}\| \\ &\quad + |(1-\beta i)(\alpha-i)|\ \|\phi'(u^*)(\alpha W+T)^{-1}\|\ \|\phi'(u^*)(W+\beta T)^{-1}\| \\ &\leqslant ab+(rb+pa)\delta+rp\delta^2.\end{aligned}$$

Now, if

$$ab+(rb+pa)\delta+rp\delta^2<1,$$

then $\rho(\mathcal{G}(\alpha,\beta;u^*)) \le \|\mathcal{G}(\alpha,\beta;u^*)\| < 1$ □

In [25], some conditions under which $ab<1$, have been presented. Therefore, for sufficiently small value of δ the condition (12) holds true and the convergence of the nonlinear TSCSP-like iteration method is achieved. Note that convergence theorem of the nonlinear TSCSP-like iteration method is similar to the latter theorem when $\alpha=\beta$.

The convergence speed of iteration methods for solving the system (1) depends on two factors: weakly nonlinearity of the weakly nonlinear systems and the optimal parameters. Both of these factors are problem-based and difficult to handle them. However, we consider the latter in the following. As the authors of [25] proved the spectral radius of the iteration matrix $\mathcal{G}(\alpha,\beta)$ satisfies

$$\begin{aligned}\rho(\mathcal{G}(\alpha,\beta)) &\le \|\mathcal{G}(\alpha,\beta)\| \\ &\le \max\left\{ \left|\frac{1-\alpha\mu_1}{\alpha+\mu_1}\right|, \left|\frac{1-\alpha\mu_n}{\alpha+\mu_n}\right| \right\} \max\left\{ \left|\frac{\beta-\mu_1}{1+\beta\mu_1}\right|, \left|\frac{\beta-\mu_n}{1+\beta\mu_n}\right| \right\} := \sigma(\alpha,\beta),\end{aligned}$$

and

$$(\alpha^*,\beta^*) = \underset{\alpha,\beta>0}{\operatorname{argmin}}\, \sigma(\alpha,\beta).$$

where

$$\alpha^* = \frac{1-\mu_1\mu_n+\sqrt{(1-\mu_1\mu_n)^2+(\mu_1+\mu_n)^2}}{\mu_1+\mu_n}, \qquad \text{and} \quad \beta^* = \frac{1}{\alpha^*}. \tag{13}$$

It is note worth to mention that the values α^* and β^* minimize only the upper bound $\sigma(\alpha,\beta)$ of the spectral radius $\mathcal{G}(\alpha,\beta)$, not $\mathcal{G}(\alpha,\beta;u^*)$. However, when the linear term Au is strongly dominant over the nonlinear term $\phi(u)$, one may apply the parameters α^* and β^* in the numerical implementation of the method. As we will see in section of the numerical experiments these values often give suitable results.

5 The inexact Picard-TTSCSP and the nonlinear inexact TTSCSP-like

In the Picard-TTSCSP and the nonlinear TTSCSP-like iteration methods, for obtaining $u^{(k+1)}$ we should solve the two subsystems with the coefficient matrices $\alpha W + T$ and $W + \beta T$. To improve the implementation of these methods, we can use iteration methods for solving the two subproblems. Since $\alpha W + T$ and $W + \beta T$ are symmetric positive definite, we can solve the two subsystems by CG and we establish inexact version of the Picard-TTSCSP and the nonlinear TTSCSP-like iteration methods. The subsystems are solved inexactly by the CG method such that the relative residual norms are less than $\epsilon_{1k} > 0$ and $\epsilon_{2k} > 0$, respectively.

In the Picard-TTSCSP iteration method, suppose $u^{(k,l+\frac{1}{2})} = u^{(k,l)} + z^{(k,l)}$ and substituting it in the first subsystem yields

$$(\alpha W + T)z^{(k,l)} = (\alpha - i)r^{(k,l)}, \tag{14}$$

where $r^{(k,l)} = \phi(u^{(k)}) - Au^{(k,l)}$. In the same way, letting

$$u^{(k,l+1)} = u^{(k,l+\frac{1}{2})} + z^{(k,l+\frac{1}{2})},$$

the second subsystem can be written as

$$(W + \beta T)z^{(k,l+\frac{1}{2})} = (1 - \beta i)r^{(k,l+\frac{1}{2})}, \tag{15}$$

where $r^{(k,l+\frac{1}{2})} = \phi(u^k) - Au^{(k,l+\frac{1}{2})}$. We inexactly solve the systems (14) and (15) by the CG method. In the similar way, the nonlinear inexact TTSCSP-like iteration method can be derived. Therefore, the obtained algorithms are summarized as follows.

The inexact Picard-TTSCSP (Picard-ITTSCSP) iteration method

1. Choose an initial guess $u^{(0)}$
2. For $k = 0, 1, 2, \ldots$ until convergence, Do
3. $\quad b := \phi(u^k)$
4. $\quad$ Set $u^{(k,0)} := u^{(k)}$
5. $\quad$ For $l = 0, 1, 2, \ldots, l_k - 1$ Do
6. $\quad\quad$ Compute $r^{(k,l)} = b - Au^{(k,l)}$ and set $\bar{r}^{(k,l)} = (\alpha - i)r^{(k,l)}$
7. $\quad\quad$ Solve $(\alpha W + T)z^{(k,l)} = \bar{r}^{(k,l)}$ by CG to compute the approximate solution $z^{(k,l)}$ satisfying $\|\bar{r}^{(k,l)} - (\alpha W + T)z^{(k,l)}\|_2 \leqslant \epsilon_{1k}\|\bar{r}^{(k,l)}\|_2$
8. $\quad\quad u^{(k,l+\frac{1}{2})} := u^{(k,l)} + z^{(k,l)}$
9. $\quad\quad$ Compute $r^{(k,l+\frac{1}{2})} = b - Au^{(k,l+\frac{1}{2})}$ and set $\bar{r}^{(k,l+\frac{1}{2})} = (1 - \beta i)r^{(k,l+\frac{1}{2})}$
10. $\quad\quad$ Solve $(W + \beta T)z^{(k,l+\frac{1}{2})} = \bar{r}^{(k,l+\frac{1}{2})}$ by CG to compute the approximate solution $z^{(k,l+\frac{1}{2})}$ satisfying $\|\bar{r}^{(k,l+\frac{1}{2})} - (W+\beta T)z^{(k,l+\frac{1}{2})}\|_2 \leqslant \epsilon_{2k}\|\bar{r}^{(k,l+\frac{1}{2})}\|_2$
11. $\quad\quad u^{(k,l+1)} := u^{(k,l+\frac{1}{2})} + z^{(k,l+\frac{1}{2})}$
12. $\quad$ EndDo
13. $\quad u^{(k+1)} := u^{(k,l_k)}$
14. EndDo

The inexact TTSCSP-like (ITTSCSP-like) iteration method

1. Choose an initial guess $u^{(0)}$
2. For $k = 0, 1, 2, \dots$ until convergence, Do
3. Compute $r^{(k)} = \phi(u^{(k)}) - Au^{(k)}$ and set $\bar{r}^{(k)} = (\alpha - i)r^{(k)}$
4. Solve $(\alpha W + T)z^{(k)} = \bar{r}^{(k)}$ by CG to compute the approximate solution $z^{(k)}$ satisfying $\|\bar{r}^{(k)} - (\alpha W + T)z^{(k)}\|_2 \leqslant \epsilon_{1k}\|\bar{r}^{(k)}\|_2$
5. $u^{(k+\frac{1}{2})} := u^{(k)} + z^{(k)}$
6. Compute $r^{(k+\frac{1}{2})} = \phi(u^{(k+\frac{1}{2})}) - Au^{(k+\frac{1}{2})}$ and set $\bar{r}^{(k+\frac{1}{2})} = (1 - \beta i)r^{(k+\frac{1}{2})}$
7. Solve $(W + \beta T)z^{(k+\frac{1}{2})} = \bar{r}^{(k+\frac{1}{2})}$ by CG to compute the approximate solution $z^{(k+\frac{1}{2})}$ satisfying $\|\bar{r}^{(k+\frac{1}{2})} - (W + \beta T)z^{(k+\frac{1}{2})}\|_2 \leqslant \epsilon_{2k}\|\bar{r}^{(k+\frac{1}{2})}\|_2$
8. $u^{(k+1)} := u^{(k+\frac{1}{2})} + z^{(k+\frac{1}{2})}$
9. EndDo

6 Numerical experiments

In this section, we give some numerical experiments to illustrate the effectiveness of the proposed methods. To do this, we compare the numerical results of the Picard-TTSCSP with those of the Picard-MHSS, the Picard-LPMHSS and the Picard-TSCSP iteration methods and the numerical results of the nonlinear TTSCSP-like iteration method with those of the nonlinear MHSS-like, LPMHSS-like and TSCSP-like methods.

Consider the two-dimensional nonlinear convection-diffusion equation (see [23])

$$\begin{cases} u_t - (\alpha_1 + i\beta_1)(u_{xx} + u_{yy}) + \varrho u = (\alpha_2 + i\beta_2)\ ue^u + \sin\sqrt{1 + u_x^2 + u_y^2}, \text{ in } (0,1] \times \Omega, \\ u(0, x, y) = u_0(x, y), \text{ in } \Omega, \\ u(t, x, y) = 0, \text{ on } (0,1] \times \partial\Omega, \end{cases} \tag{16}$$

where $\Omega = (0,1) \times (0,1)$, $\partial\Omega$ is the boundary of Ω, $\alpha_1 = \beta_1 = 1$, $\alpha_2 = \beta_2 = 0.5$ and ϱ is a positive constant used to control the magnitude of the reaction term. Discretizing (16) on equidistant grids $\Delta t = h = 1/(N+1)$, at each temporal step of the implicit scheme, we get a system of weakly nonlinear equations (1) of the form

$$F(u) = Mu - h^2\phi(u) = 0, \tag{17}$$

where

$$M = h(1 + \varrho\Delta t)I_n + (\alpha_1 + i\beta_1)(A_n \otimes I + I \otimes A_n),$$
$$\phi(u) = (\alpha_2 + i\beta_2)\ \psi(u) + \sin(1 + B(u)).$$

Here, $A_N = \text{tridiag}(-1, 2, -1)$, $B = C_N \otimes C_N$, $C_N = \text{tridiag}(-1/h, 0, 1/h)$, $\psi(u) = (u_1.e^{u_1}, u_2.e^{u_2}, \dots, u_n.e^{u_n},)^T$ with $n = N \times N$ and $\sin(u) = (\sin(u_1), \sin(u_2), \dots, \sin(u_n))^T$.

Table 1: The experimentally optimal parametes α and β for the Picard-MHSS, Picard-LPMHSS, Picard-TSCSP and Picard-TTSCSP iteration methods for $N = 32$.

	Iteration		$\varrho = 0.1$	$\varrho = 1$	$\varrho = 10$
$\eta = 0.1$	Picard-MHSS	α_{opt}	0.34	0.34	0.35
	Picard-LPMHSS	α_{opt}	1.3	1.3	1.5
	Picard-TSCSP	α_{opt}	0.5	0.5	0.5
	Picard-TTSCSP	α_{opt}	1.17	1.17	1.17
		β_{opt}	0.31	0.30	0.30
$\eta = 0.01$	Picard-MHSS	α_{opt}	0.33	0.34	0.36
	Picard-LPMHSS	α_{opt}	1.0	1.0	1.0
	Picard-TSCSP	α_{opt}	0.45	0.45	0.45
	Picard-TTSCSP	α_{opt}	1.17	1.17	1.17
		β_{opt}	0.30	0.30	0.30
$\eta = 0.001$	Picard-MHSS	α_{opt}	0.34	0.34	0.34
	Picard-LPMHSS	α_{opt}	1.5	1.5	1.6
	Picard-TSCSP	α_{opt}	0.43	0.43	0.42
	Picard-TTSCSP	α_{opt}	1.17	1.17	1.17
		β_{opt}	0.30	0.30	0.30

In our numerical experimens, we use a zeros vector as the initial guess. The stopping criterion is set to be

$$\frac{\|F(u^{(k)})\|_2}{\|F(u^{(0)})\|_2} < 10^{-6},$$

and the stopping criterion for the inner iteration of the Picard-MHSS, Picard-LPMHSS, Picard-TSCSP and Picard-TTSCSP iteration method are

$$\frac{\|F(u^{(k,l_k)})\|_2}{\|F(u^{(k,0)})\|_2} < \eta_k,$$

where l_k is the number of the inner iteration steps. If η_k is fixed for all k, then it is simply denoted by η. The coefficient matrices of the subsystems in the Picard-TTSCSP and

Table 2: The experimentally optimal parametes α and β for the Picard-MHSS, Picard-LPMHSS, Picard-TSCSP and Picard-TTSCSP iteration methods for $N = 64$.

	Iteration		$\varrho = 0.1$	$\varrho = 1$	$\varrho = 10$
$\eta = 0.1$	Picard-MHSS	α_{opt}	0.20	0.20	0.20
	Picard-LPMHSS	α_{opt}	1.1	1.1	1.1
	Picard-TSCSP	α_{opt}	0.36	0.36	0.36
	Picard-TTSCSP	α_{opt}	1.17	1.17	1.17
		β_{opt}	0.30	0.30	0.30
$\eta = 0.01$	Picard-MHSS	α_{opt}	0.20	0.20	0.21
	Picard-LPMHSS	α_{opt}	1.0	1.0	1.0
	Picard-TSCSP	α_{opt}	0.36	0.36	0.35
	Picard-TTSCSP	α_{opt}	1.17	1.17	1.17
		β_{opt}	0.30	0.30	0.30
$\eta = 0.001$	Picard-MHSS	α_{opt}	0.19	0.19	0.20
	Picard-LPMHSS	α_{opt}	0.9	0.9	0.9
	Picard-TSCSP	α_{opt}	0.35	0.35	0.35
	Picard-TTSCSP	α_{opt}	1.17	1.17	1.17
		β_{opt}	0.30	0.30	0.30

the nonlinear TTSCSP-like iteration methods are symmetric positive definite. Hence in the exact version of these algorithms these systems are solved exactly using the Cholesky factorization of the coefficient matrices. In the inexact implementation of these algorithms these systems are solved inexactly using the CG method. The CG iteration is stopped as soon as the residual norm is reduced by a factor of 10^2 and the maximum number of iterations is set to be 1000.

All runs are implemented in Matlab R2014b with a laptop with 2.40 GHz central processing unit (Intel(R) Core(TM) i7-5500), 8 GB memory and Windows 10 operating system.

Numerical experiments are presented for problem sizes $N = 32, 64, 128$ (i.e., $n =$

Table 3: The experimentally optimal parametes α and β for the Picard-MHSS, Picard-LPMHSS, Picard-TSCSP and Picard-TTSCSP iteration methods for $N = 128$.

	Iteration		$\varrho = 0.1$	$\varrho = 1$	$\varrho = 10$
$\eta = 0.1$	Picard-MHSS	α_{opt}	0.12	0.12	0.12
	Picard-LPMHSS	α_{opt}	0.80	0.80	0.80
	Picard-TSCSP	α_{opt}	0.26	0.24	0.26
	Picard-TTSCSP	α_{opt}	1.17	1.17	1.17
		β_{opt}	0.20	0.20	0.20
$\eta = 0.01$	Picard-MHSS	α_{opt}	0.13	0.13	0.13
	Picard-LPMHSS	α_{opt}	0.80	0.80	0.80
	Picard-TSCSP	α_{opt}	0.24	0.24	0.25
	Picard-TTSCSP	α_{opt}	1.17	1.17	1.17
		β_{opt}	0.30	0.30	0.30
$\eta = 0.001$	Picard-MHSS	α_{opt}	0.12	0.12	0.12
	Picard-LPMHSS	α_{opt}	0.80	0.80	0.80
	Picard-TSCSP	α_{opt}	0.23	0.23	0.23
	Picard-TTSCSP	α_{opt}	1.17	1.17	1.17
		β_{opt}	0.30	0.30	0.30

$32^2, 64^2, 128^2$) and tolerances $\eta = 0.1, 0.01, 0.001$. In our implementation IT_{int}, IT_{out}, IT, CPU show the average of the inner iterations, the outer iteration, the total iteration and the total CPU-time, respectively. As we mentioned the optimal values of the parameters in the tested methods are problem-based, even if the the nonlinear term $\phi(u)$ is neglected. However, we will see shortly that, in the TTSCSP-based methods, the parameters α^* and β^* (see Eq. (13)) often give suitable results. The optimal parameters α_{opt} and β_{opt} used in all the experiments were found experimentally and are the ones resulting in the least numbers of iterations. Parameters α_{opt} and β_{opt} of Picard-MHSS, Picard-LPMHSS, Picard-TSCSP and Picard-TTSCSP are presented in Tables 1-3.

In Table 7, we compare the Picard-TTSCSP iteration method with the Picard-MHSS,

Table 4: The experimentally optimal parametes α and β for the inexact Picard-MHSS, the inexact Picard-LPMHSS, the inexact Picard-TSCSP and inexact Picard-TTSCSP iteration methods for $N = 32$.

	Iteration		$\varrho = 0.1$	$\varrho = 1$	$\varrho = 10$
$\eta = 0.1$	Picard-IMHSS	α_{opt}	0.43	0.43	0.45
	Picard-ILPMHSS	α_{opt}	0.76	0.76	0.74
	Picard-ITSCSP	α_{opt}	0.50	0.49	0.44
	Picard-ITTSCSP	α_{opt}	1.17	1.17	1.17
		β_{opt}	0.30	0.30	0.30
$\eta = 0.01$	Picard-IMHSS	α_{opt}	0.43	0.42	0.45
	Picard-ILPMHSS	α_{opt}	0 .83	0.83	0.83
	Picard-ITSCSP	α_{opt}	0.45	0.45	0.45
	Picard-ITTSCSP	α_{opt}	1.17	1.17	1.17
		β_{opt}	0.30	0.30	0.30
$\eta = 0.001$	Picard-MHSS	α_{opt}	0.40	0.41	0.42
	Picard-ILPMHSS	α_{opt}	0.83	0.83	0.81
	Picard-ITSCSP	α_{opt}	0.43	0.43	0.43
	Picard-ITTSCSP	α_{opt}	1.17	1.17	1.17
		β_{opt}	0.30	0.30	0.30

Picard-LPMHSS and Picard-TSCSP iteration methods. The Picard-HSS, Picrd-LPMHHS, Picard-TSCSP and Picard-TTSCSP are respectively denoted by "P-MHSS", "P-LPMHHS", "P-TSCSP" and "P-TTSCSP". From these tables, we see that the Picard-TSCSP and Picard-TTSCSP outperforms the the other iteration methods, in terms of the number of iterations and the CPU time.

The numerical results of the nonlinear TTSCSP-like iteration method along with those of the nonlinear MHSS-like, the LPMHSS-like and the TSCSP-like iteration methods are listed in Table 10. As we see from the total number of iterations and the CPU time point of view, the TTSCSP-like method performs better.

Table 9 presents the numerical results of the Picard-TTSCSP method with the optimal parameters α_{opt} and β_{opt} with those of the same method with the parameters α^* and β^*

Table 5: The experimentally optimal parametes α and β for the inexact Picard-MHSS, the inexact Picard-LPMHSS, the inexact Picard-TSCSP and the inexact Picard-TTSCSP iteration methods for $N = 64$.

	Iteration		$\varrho = 0.1$	$\varrho = 1$	$\varrho = 10$
$\eta = 0.1$	Picard-IMHSS	α_{opt}	0.26	0.26	0.27
	Picard-ILPMHSS	α_{opt}	0.78	0.78	0.78
	Picard-ITSCSP	α_{opt}	0.35	0.35	0.31
	Picard-ITTSCSP	α_{opt}	1.17	1.17	1.17
		β_{opt}	0.30	0.30	0.30
$\eta = 0.01$	Picard-IMHSS	α_{opt}	0.27	0.26	0.27
	Picard-ILPMHSS	α_{opt}	0.87	0.87	087
	Picard-ITSCSP	α_{opt}	0.35	0.35	0.35
	Picard-ITTSCSP	α_{opt}	1.17	1.17	1.17
		β_{opt}	0.30	0.30	0.30
$\eta = 0.001$	Picard-IMHSS	α_{opt}	0.25	0.25	0.26
	Picard-ILPMHSS	α_{opt}	0.87	0.879	0.86
	Picard-ITSCSP	α_{opt}	0.32	0.32	0.32
	Picard-ITTSCSP	α_{opt}	1.17	1.17	1.17
		β_{opt}	0.30	0.30	0.30

provided by Eq. (13). As we observe there is no significant difference between the number of iterations, as well as the CPU time, for two different pairs of the parameters.

Also we report the numerical results of the inexact version of the Picard-TTSCSP and the nonlinear TTSCSP-like methods for three values of η in Tables 8, 11. In Table 8, the inexact version of Picard-MHSS, Picard-LPMHSS, Picard-TSCSP and Picard-TTSCSP are denoted by "P-IMHSS", "P-ILPMHSS", "P-ITSCSP" and "P-ITTSCSP". The optimal values of the parameters α and β are given in Tables 4-6. Likewise, the inexact version of the nonlinear MHSS-like, LPMHSS-like, TSCSP-like and TTSCSP-like are denoted by "IMHSS-like", "ILPMHSS-like", "ITSCSP-like" and "ITTSCSP-like", respectively. Also, as we see, the optimal values of the parameters α and β remain constant with the problem size for both of the Picard-TTSCSP and the nonlinear TTSCSP-like iteration methods, in

Table 6: The experimentally optimal parametes α and β for the inexact Picard-MHSS, the inexact Picard-LPMHSS, the inexact Picard-TSCSP and the inexact Picard-TTSCSP iteration methods for $N = 128$.

	Iteration		$\varrho = 0.1$	$\varrho = 1$	$\varrho = 10$
$\eta = 0.1$	Picard-IMHSS	α_{opt}	0.17	0.17	0.18
	Picard-ILPMHSS	α_{opt}	0.80	0.80	0.80
	Picard-ITSCSP	α_{opt}	0.27	0.28	0.27
	Picard-ITTSCSP	α_{opt}	1.17	1.17	1.17
		β_{opt}	0.30	0.30	0.30
$\eta = 0.01$	Picard-IMHSS	α_{opt}	0.18	0.18	0.18
	Picard-ILPMHSS	α_{opt}	0.80	0.80	0.80
	Picard-ITSCSP	α_{opt}	0.28	0.28	0.28
	Picard-ITTSCSP	α_{opt}	1.16	1.17	1.17
		β_{opt}	0.30	0.30	0.30
$\eta = 0.001$	Picard-IMHSS	α_{opt}	0.17	0.17	0.17
	Picard-ILPMHSS	α_{opt}	0.74	0.74	0.74
	Picard-ITSCSP	α_{opt}	0.24	0.24	0.23
	Picard-ITTSCSP	α_{opt}	1.17	1.17	1.17
		β_{opt}	0.30	0.30	0.30

both of the exact and inexact versions. We see that both of the inexact Picard-TTSCSP and TTSCSP-like methods decrease the iteration numbers and the CPU time for different values of η and N.

7 Conclusion

In this work, we have established the Picard-TSCSP, the Picard-TTSCSP, the nonlinear TSCSP-like and the TTSCSP-like iteration methods and their inexact versions for a class of the large sparse system of weakly nonlinear equations. Numerical experiments show that the new iteration methods (exact or inexact versions) are efficient and feasible. Numerical results show that the new methods outperforms the Picard-MHSS, the Picard-LPMHSS, the Picard-TSCSP and the Picard-TTSCSP iteration methods.

Acknowledgements

The work of Davod Khojasteh Salkuyeh is partially supported by University of Guilan. The authors would like to thank the anonymous referees for their comments and suggestions.

References

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

[2] H.-B. An, Z.-Z. Bai, NGLM: A globally convergent Newton-GMRES method, Mathematica Numerica Sinica, 27 (2005) 151-174 (in Chinese).

[3] Z.-Z. Bai, H.-B. An, On efficient variants and global convergence of the Newton-GMRES method, Journal on Numerical Methods and Computer Applications, 26 (2005) 291-300 (in Chinese).

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

[5] Z.-Z. Bai, M. Benzi, F. Chen, Modified HSS iteration methods for a class of complex symmetric linear systems, Computing, 87 (2010) 93-111.

[6] Z-.Z. Bai, Y.-G. Huang, Asynchronous multisplitting two-stage iterations for systems of weakly nonlinear equations, J. Comput. Appl. Math., 93 (1998) 13-33.

[7] Z.-Z. Bai, G.H. Golub, M.K. Ng, Hermitian and skew-Hermitian splitting methods for non-Hermitian positive definite linear systems, SIAM J. Matrix Anal. Appl., 24 (2003) 603-626.

[8] Z.-Z. Bai, Y.-M. Huang, M.k. Ng, On preconditioned iterative methods for Burgers equations, SIAM J. Sci. Comput., 29 (2007) 415-439.

[9] Z.-Z. Bai, Y.-M. Huang, M.k. Ng, On preconditioned iterative methods for certain time-dependent partial differential equations, SIAM J. Numer. Anal., 47 (2009) 1019-1037.

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

[11] Z.-Z. Bai, On the convergence of parallel chaotic nonlinear multisplitting Newton-type methods, Journal of Computational and Applied Mathematics, 80 (1997) 317-334.

[12] Z.-Z. Bai, Optimal parameters in the HSS-like methods for saddle-point problems, Numer. Linear Algebra Appl., 16 (2009) 447-479.

[13] Z.-Z. Bai, Parallel multisplitting two-stage iterative methods for large sparse systems of weakly nonlinear equations, Numer. Algor., 15 (1997) 347-372.

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

[15] M. Benzi, M.K. Ng, Preconditioned iterative methods for weighted toeplitz least squares problems, SIAM J. Matrix Anal. Appl., 27 (2006) 1106-1124.

[16] R.S. Dembo, S.C. Eisenstat, T. Steihaug, Inexact Newton methods, SIAM Journal on Numerical Analysis, 19 (1982) 400-408.

[17] E. Galligani, The newton-arithmetic mean method for the solution of systems of nonlinear equations, Appl. Math. Comput., 134 (2003) 9-34.

[18] X.-P Guo, On semilocal convergence of inexact Newton methods, J. Comput. Math., 25 (2007) 231-242.

[19] D. Hezari, D.K. Salkuyeh, V. Edalatpour, A new iterative method for solving a class of complex symmetric system of linear equathions, Numer. Algor. 73 (2016) 927-955.

[20] [20] C.T. Kelley, Iterative Methods for Linear and Nonlinear Equations, SIAM, Philadelphia, 1995.

[21] L.D. Liao and G.F. Zhang, Efficient Preconditioner and Iterative Method for Large Complex Symmetric Linear Algebraic Systems, East. Asia. J. Appl. Math., 7 (2017) 530-547.

[22] X. Li, A.-L. Yang, Y.-J. Wu, Lopsided PMHSS iteration method for a class of complex symmetric linear systems, Numer. Algor., 66 (2014) 555-568.

[23] C.-X. Li, S.-L. Wu, On LPMHSS-based iteration methods for a class of weakly nonlinear systems, Comp. Appl. Math., 37 (2018) 1232-1249.

[24] J.M. Ortega, W.C. Rheinboldt, Iterative Solution of Nonlinear Equations in Several Variables, Academic Press, New York, 1970.

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

[26] D.K. Salkuyeh, Two-step scale-splitting method for solving complex symmetric system of linear equations, arXiv:1705.02468.

[27] A.H. Sherman, On Newton-iterative methods for the solution of systems of nonlinear equations, SIAM Journal on Numerical Analysis, 15 (1978) 755-771.

[28] T. Tang, Superconvergence of numerical solutions to weakly singular volterra integro-differential equations, Numer. Math., 61 (1992) 373-382.

[29] W. Yang, Y.-J. Wu, J. Fu, On Picard-MHSS methods for weakly nonlinear systems, Math. Numer. Sinca., 36 (2014) 291-302 (in chinese).

[30] Y.-M. Wang, Parallel multisplitting methods for a class of systems of weakly nonlinear equations without isotone mapping, Appl. Math. Comput., 109 (2000) 135-150.

[31] J. Wang , X.-P. Guo and H.-X. Zhong, Accelerated GPMHSS Method for Solving Complex Systems of Linear Equations, East. Asia. J. Appl. Math.,7 (2017) 143-155.

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

Table 7: Numerical results for the Picard-MHSS, Picard-LPMHSS, Picard TSCSP, Picard-TTSCSP iteration methods.

			$N=32$			$N=64$			$N=128$		
	Iteration		$\varrho=0.1$	$\varrho=1$	$\varrho=10$	$\varrho=0.1$	$\varrho=1$	$\varrho=10$	$\varrho=0.1$	$\varrho=1$	$\varrho=10$
$\eta=0.1$	P-MHSS	IT_{int}	1.01	1.01	1.02	1.01	1.01	1.01	1.01	1.01	1.01
		IT_{out}	70	70	67	107	106	103	155	155	153
		IT	71.01	71.01	68.02	108.01	107.01	104.01	156.01	156.01	156.01
		CPU	0.082	0.084	0.079	0.358	0.359	0.348	2.166	2.142	2.116
	P-LPMHSS	IT_{int}	1.03	1.03	1.03	1.03	1.03	1.03	1.03	1.03	1.03
		IT_{out}	35	35	35	34	34	34	34	34	34
		IT	36.03	36.03	36.03	35.03	35.03	35.03	35.03	35.03	35.03
		CPU	0.048	0.047	0.048	0.169	0.165	0.170	1.171	1.169	1.172
	P-TSCSP	IT_{int}	1.2	1.2	1.2	1.14	1.14	1.25	1.11	1.11	1.11
		IT_{out}	6	6	6	8	8	9	10	10	9
		IT	7.2	7.2	7.2	9.14	9.14	10.25	11.11	11.11	10.11
		CPU	0.025	0.025	0.025	0.065	0.065	0.067	0.398	0.397	0.393
	P-TTSCSP	IT_{int}	1.33	1.33	1.33	1.25	1.25	1.25	1.25	1.25	1.25
		IT_{out}	4	4	4	5	5	5	5	5	5
		IT	5.33	5.33	5.33	6.25	6.25	6.25	6.25	6.25	6.25
		CPU	0.024	0.024	0.024	0.057	0.057	0.056	0.318	0.310	0.323
$\eta=0.01$	P-MHSS	IT_{int}	1.02	1.02	1.02	1.02	1.02	1.02	1.01	1.01	1.01
		IT_{out}	57	57	53	84	83	79	122	122	117
		IT	58.02	58.02	54.02	85.02	84.02	80.02	123.01	123.01	118.01
		CPU	0.079	0.078	0.072	0.340	0.331	0.315	2.057	2.023	1.960
	P-LPMHSS	IT_{int}	1.04	1.04	1.04	1.04	1.04	1.04	1.04	1.04	1.04
		IT_{out}	27	27	27	27	27	27	27	27	28
		IT	28.04	28.04	28.04	28.04	28.04	28.04	28.04	28.04	29.04
		CPU	0.046	0.048	0.048	0.165	0.165	0.165	1.162	1.159	1.157
	P-TSCSP	IT_{int}	1.25	1.25	1.25	1.17	1.17	1.17	1.11	1.11	1.11
		IT_{out}	5	5	5	7	7	7	10	10	9
		IT	6.25	6.25	6.25	8.17	8.17	8.17	11.11	11.11	10.11
		CPU	0.025	0.025	0.025	0.064	0.065	0.067	0.402	0.397	0.393
	P-TTSCSP	IT_{int}	1.50	1.50	1.50	1.33	1.33	1.33	1.20	1.20	1.25
		IT_{out}	3	3	3	4	4	4	6	6	5
		IT	4.50	4.50	4.50	5.33	5.33	5.33	7.20	7.20	6.25
		CPU	0.024	0.023	0.024	0.056	0.055	0.056	0.330	0.330	0.324
$\eta=0.001$	P-MHSS	IT_{int}	1.02	1.02	1.03	1.02	1.02	1.02	1.02	1.02	1.02
		IT_{out}	43	43	39	63	62	60	92	92	90
		IT	44.02	44.02	40.03	64.02	63.02	61.02	93.02	93.02	91.02
		CPU	0.074	0.074	0.069	0.335	0.326	0.318	2.014	1.996	1.942
	P-LPMHSS	IT_{int}	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05
		IT_{out}	21	21	22	21	21	21	21	21	21
		IT	22.05	22.05	23.05	22.05	22.05	22.05	22.05	22.05	22.05
		CPU	0.046	0.047	0.047	0.168	0.161	0.158	1.145	1.152	1.169
	P-TSCSP	IT_{int}	1.25	1.25	1.25	1.20	1.20	1.20	1.17	1.17	1.17
		IT_{out}	5	5	5	6	6	6	7	7	7
		IT	6.25	6.25	6.25	7.20	7.20	7.20	8.17	8.17	8.17
		CPU	0.025	0.025	0.026	0.072	0.070	0.072	0.397	0.395	0.396
	P-TTSCSP	IT_{int}	1.50	1.50	1.50	1.33	1.33	1.33	1.25	1.25	1.25
		IT_{out}	3	3	3	4	4	4	5	5	5
		IT	4.50	4.50	4.50	5.33	5.33	5.33	6.25	6.25	6.25
		CPU	0.023	0.023	0.024	0.055	0.055	0.056	0.327	0.325	0.336

Table 8: Numerical results for the inexact Picard-MHSS, inexact Picard-LPMHSS, inexact Picard TSCSP, inexact Picard-TTSCSP iteration methods.

			$N = 32$			$N = 64$			$N = 128$		
	Iteration		$\varrho = 0.1$	$\varrho = 1$	$\varrho = 10$	$\varrho = 0.1$	$\varrho = 1$	$\varrho = 10$	$\varrho = 0.1$	$\varrho = 1$	$\varrho = 10$
$\eta = 0.1$	P-IMHSS	IT_{int}	1.01	1.01	1.01	1.01	1.01	1.01	1.00	1.00	1.00
		IT_{out}	86	86	81	138	137	133	214	214	208
		IT	87.01	87.01	82.01	139.01	138.01	134.01	215.00	215.00	209.00
		CPU	0.218	0.219	0.205	1.242	1.226	1.160	5.663	5.611	5.337
	P-ILPMHSS	IT_{int}	1.03	1.03	1.03	1.03	1.03	1.03	1.03	1.03	1.03
		IT_{out}	34	34	34	34	34	34	34	34	34
		IT	35.03	35.03	35.03	35.03	35.03	35.03	35.03	35.03	35.03
		CPU	0.274	0.281	0.263	1.192	1.200	1.350	3.976	4.261	4.219
	P-ITSCSP	IT_{int}	1.20	1.20	1.20	1.14	1.14	1.13	1.10	1.10	1.10
		IT_{out}	6	6	6	8	8	9	11	11	11
		IT	7.20	7.20	7.20	9.14	9.14	10.13	12.10	12.10	12.10
		CPU	0.064	0.067	0.070	0.430	0.421	0.472	2.089	2.161	2.115
	P-ITTSCSP	IT_{int}	1.33	1.33	1.33	1.25	1.25	1.25	1.20	1.20	1.17
		IT_{out}	4	4	4	5	5	5	6	6	8
		IT	5.33	5.33	5.33	6.25	6.25	6.25	7.20	7.20	8.17
		CPU	0.055	0.055	0.055	0.213	0.209	0.201	0.781	0.771	0.824
$\eta = 0.01$	P-IMHSS	IT_{int}	1.01	1.01	1.02	1.01	1.01	1.01	1.01	1.01	1.01
		IT_{out}	70	70	65	109	109	107	168	167	163
		IT	71.14	71.01	66.02	110.01	110.01	108.01	169.01	168.01	164.01
		CPU	0.213	0.214	0.200	1.203	1.179	1.134	5.224	5.189	5.113
	P-ILPMHSS	IT_{int}	1.04	1.04	1.04	1.04	1.04	1.04	1.04	1.04	1.04
		IT_{out}	27	27	27	27	27	27	27	27	27
		IT	28.04	28.04	28.04	28.04	28.04	28.04	28.04	28.04	28.04
		CPU	0.248	0.263	0.241	1.081	1.231	1.078	4.066	4.275	4.235
	P-ITSCSP	IT_{int}	1.25	1.25	1.20	1.17	1.17	1.17	1.13	1.13	1.13
		IT_{out}	5	5	6	7	7	7	9	9	9
		IT	6.25	6.25	7.20	8.17	8.17	8.17	10.13	10.13	10.13
		CPU	0.067	0.069	0.070	0.430	0.424	0.414	2.315	2.308	2.161
	P-ITTSCSP	IT_{int}	1.33	1.33	1.33	1.33	1.33	1.33	1.20	1.20	1.25
		IT_{out}	4	4	4	4	4	4	6	6	5
		IT	5.33	5.33	5.33	5.33	5.33	5.33	7.20	7.20	6.25
		CPU	0.059	0.062	0.059	0.207	0.201	0.196	0.935	0.905	0.956
$\eta = 0.001$	P-IMHSS	IT_{int}	1.02	1.02	1.02	1.01	1.01	1.01	1.01	1.01	1.01
		IT_{out}	51	50	48	81	81	77	127	127	127
		IT	52.02	51.02	49.02	82.01	82.01	78.01	128.01	128.01	128.01
		CPU	0.230	0.217	0.219	1.238	1.223	1.157	5.428	5.437	5.192
	P-ILPMHSS	IT_{int}	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05
		IT_{out}	20	20	20	20	20	20	21	21	21
		IT	21.05	21.05	21.05	21.05	21.05	21.05	21.05	21.05	21.05
		CPU	0.287	0.248	0.247	1.326	1.404	1.147	4.162	3.922	4.179
	P-ITSCSP	IT_{int}	1.25	1.25	1.25	1.20	1.20	1.20	1.17	1.17	1.14
		IT_{out}	5	5	5	6	6	6	7	7	8
		IT	6.25	6.25	6.25	7.20	7.20	7.20	8.17	8.17	8.14
		CPU	0.075	0.078	0.077	0.491	0.496	0.477	2.718	2.696	2.967
	P-ITTSCSP	IT_{int}	1.33	1.33	1.33	1.33	1.33	1.33	1.25	1.25	1.25
		IT_{out}	4	4	4	4	4	4	5	5	5
		IT	5.33	5.33	5.33	5.33	5.33	5.33	6.25	6.25	6.25
		CPU	0.059	0.059	0.058	0.233	0.223	0.226	0.995	0.970	0.999

Table 9: Numerical results for the Picard-TTSCSP iteration methods with α_{opt}, β_{opt} and α^*, β^*.

	$N=32$			$N=64$			$N=128$		
	$\varrho=0.1$	$\varrho=1$	$\varrho=10$	$\varrho=0.1$	$\varrho=1$	$\varrho=10$	$\varrho=0.1$	$\varrho=1$	$\varrho=10$
α_{opt}	1.17	1.17	1.17	1.17	1.17	1.17	1.17	1.17	1.17
β_{opt}	0.31	0.31	0.30	0.30	0.30	0.30	0.20	0.20	0.20
IT_{int}	1.33	1.33	1.33	1.25	1.25	1.25	1.25	1.25	1.25
IT_{out}	4	4	4	5	5	5	5	5	5
IT	5.33	5.33	5.33	6.25	6.25	6.25	6.25	6.25	6.25
CPU	0.024	0.024	0.024	0.057	0.057	0.056	0.327	0.331	0.329
α^*	1.56	1.57	1.65	1.80	1.81	1.85	2.03	2.03	2.05
β^*	0.61	0.64	0.60	0.56	0.56	0.54	0.49	0.49	0.49
IT_{int}	1.25	1.25	1.25	1.20	1.20	1.20	1.17	1.17	1.20
IT_{out}	5	5	5	6	6	6	7	7	6
IT	6.25	6.25	6.25	7.20	7.20	7.20	8.17	8.17	7.20
CPU	0.027	0.027	0.027	0.071	0.071	0.073	0.338	0.338	0.331

Table 10: Numerical results for the MHSS-like, LPMHSS-like, TSCSP-like and TTSCSP-like iteartin methods.

	Iteration		$\varrho = 0.1$	$\varrho = 1$	$\varrho = 10$
$N = 32$	MHSS-like	α_{opt}	0.34	0.34	0.35
		IT	82	81	78
		CPU	0.0854	0.0863	0.0859
	LPMHSS-like	α_{opt}	3.0	3.0	3.0
		IT	35	35	35
		CPU	0.0589	0.0543	0.0530
	TSCSP-like	α_{opt}	0.43	0.43	0.43
		IT	7	7	7
		CPU	0.0274	0.0282	0.0286
	TTSCSP-like	α_{opt}	1.06	1.06	1.06
		β_{opt}	0.36	0.36	0.36
		IT	4	4	4
		CPU	0.0241	0.0255	0.0236
$N = 64$	MHSS-like	α_{opt}	0.2	0.2	0.2
		IT	125	124	120
		CPU	0.3611	0.3574	0.3498
	LPMHSS-like	α_{opt}	2.7	2.7	2.7
		IT	35	35	35
		CPU	0.1477	0.1500	0.1480
	TSCSP-like	α_{opt}	0.35	0.35	0.35
		IT	9	9	10
		CPU	0.0726	0.0743	0.0734
	TTSCSP-like	α_{opt}	0.94	0.94	0.94
		β_{opt}	0.29	0.29	0.29
		IT	5	5	5
		CPU	0.0596	0.0594	0.0614
$N = 128$	MHSS-like	α_{opt}	0.12	0.12	0.12
		IT	181	181	178
		CPU	2.1688	2.1335	2.1049
	LPMHSS-like	α_{opt}	2.35	2.35	2.35
		IT	35	35	35
		CPU	0.7675	0.7805	0.7681
	TSCSP-like	α_{opt}	0.24	0.24	0.24
		IT	13	13	13
		CPU	0.3731	0.3748	0.3796
	TTSCSP-like	α_{opt}	0.82	0.82	0.82
		β_{opt}	0.22	0.22	0.22
		IT	6	6	6
		CPU	0.3138	0.3094	0.3136

Table 11: Numerical results for the inexact MHSS-like, inexact LPMHSS-like, inexact TSCSP-like and inexact TTSCSP-like iteration methods.

	Iteration		$\varrho = 0.1$	$\varrho = 1$	$\varrho = 10$
$N = 32$	IMHSS-like	α_{opt}	0.34	0.34	0.35
		IT	82	81	78
		CPU	0.4466	0.4339	0.4094
	ILPMHSS-like	α_{opt}	3.0	3.0	3.0
		IT	35	35	35
		CPU	0.2342	0.2326	0.2252
	ITSCSP-like	α_{opt}	0.43	0.43	0.43
		IT	7	7	7
		CPU	0.0702	0.0704	0.0702
	ITTSCSP-like	α_{opt}	1.06	1.06	1.06
		β_{opt}	0.36	0.36	0.36
		IT	4	4	4
		CPU	0.0520	0.0505	0.0492
$N = 64$	IMHSS-like	α_{opt}	0.20	0.20	0.20
		IT	125	124	120
		CPU	2.8561	2.8323	2.7401
	ILPMHSS-like	α_{opt}	2.7	2.7	2.7
		IT	35	35	35
		CPU	1.0628	1.0487	1.0167
	ITSCSP-like	α_{opt}	0.35	0.35	0.35
		IT	9	9	10
		CPU	0.3495	0.3405	0.3473
	ITTSCSP-like	α_{opt}	0.94	0.94	0.94
		β_{opt}	0.29	0.29	0.29
		IT	5	5	5
		CPU	0.1953	0.1930	0.1845
$N = 128$	IMHSS-like	α_{opt}	0.12	0.12	0.12
		IT	181	181	178
		CPU	11.7168	11.4980	11.5725
	ILPMHSS-like	α_{opt}	2.37	2.37	2.37
		IT	35	35	35
		CPU	4.7721	4.7076	4.5706
	ITSCSP-like	α_{opt}	0.24	0.24	0.24
		IT	13	13	13
		CPU	2.2445	2.2148	2.1373
	ITTSCSP-like	α_{opt}	0.82	0.82	0.82
		β_{opt}	0.22	0.22	0.22
		IT	6	6	6
		CPU	0.9811	0.9842	0.9589