



On CSCS-based iteration methods for Toeplitz system of weakly nonlinear equations

Mu-Zheng Zhu^{a,b}, Guo-Feng Zhang^{a,*}

^a School of Mathematics and Statistics, Lanzhou University, Lanzhou 730000, PR China

^b Department of Mathematics, Hexi University, Zhangye 734000, PR China

ARTICLE INFO

Article history:

Received 3 February 2010

Received in revised form 31 December 2010

MSC:

65F10

65F50

65T50

65Y05

Keywords:

Weakly nonlinear equations

Nonlinear iteration scheme

CSCS iteration method

Inner/outer iteration scheme

Toeplitz system

ABSTRACT

For Toeplitz system of weakly nonlinear equations, by using the separability and strong dominance between the linear and the nonlinear terms and using the circulant and skew-circulant splitting (CSCS) iteration technique, we establish two nonlinear composite iteration schemes, called Picard-CSCS and nonlinear CSCS-like iteration methods, respectively. The advantage of these methods is that they do not require accurate computation and storage of Jacobian matrix, and only need to solve linear sub-systems of constant coefficient matrices. Therefore, computational workloads and computer storage may be saved in actual implementations. Theoretical analysis shows that these new iteration methods are local convergent under suitable conditions. Numerical results show that both Picard-CSCS and nonlinear CSCS-like iteration methods are feasible and effective for some cases.

© 2011 Elsevier B.V. All rights reserved.

1. Introduction

Consider iterative solution of the following large Toeplitz system of weakly nonlinear equations

$$Ax = \phi(x), \quad \text{or} \quad F(x) := Ax - \phi(x) = 0, \quad (1.1)$$

where $A \in \mathbb{C}^{n \times n}$ is a large, nonsymmetric and positive definite Toeplitz matrix, $\phi : \mathbb{D} \subset \mathbb{C}^n \rightarrow \mathbb{C}^n$ is a continuously differentiable function defined on the open convex domain \mathbb{D} in the n -dimensional linear space \mathbb{C}^n . Here, the system of nonlinear equations (1.1) is said to be Toeplitz weakly nonlinear if the linear term Ax is strongly dominant over the nonlinear term $\phi(x)$ in certain norm and A is a Toeplitz matrix; see [1–3].

The system of weakly nonlinear equations (1.1) may arise in many areas of scientific computing and engineering applications. For example, in finite-difference or sinc discretizations of nonlinear partial differential equations [4–7], in collocation approximations of nonlinear integral equation [8] and in saddle point problems from image processing [9,10].

A matrix A is said to be Toeplitz if

$$A = \begin{bmatrix} a_0 & a_{-1} & \cdots & a_{2-n} & a_{1-n} \\ a_1 & a_0 & a_{-1} & \cdots & a_{2-n} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ a_{n-2} & \cdots & a_1 & a_0 & a_{-1} \\ a_{n-1} & a_{n-2} & \cdots & a_1 & a_0 \end{bmatrix},$$

* Corresponding author.

E-mail addresses: zhumzh07@lzu.edu.cn (M.-Z. Zhu), gf_zhang@lzu.edu.cn (G.-F. Zhang).

i.e., A is constant along its diagonals; see [11]. A Toeplitz matrix A possesses a circulant and skew-circulant splitting $A = C + S$, where

$$C = \frac{1}{2} \begin{bmatrix} a_0 & a_{-1} + a_{n-1} & \cdots & a_{2-n} + a_2 & a_{1-n} + a_1 \\ a_1 + a_{1-n} & a_0 & \cdots & \cdots & a_{2-n} + a_2 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ a_{n-2} + a_{-2} & \cdots & \cdots & a_0 & a_{-1} + a_{n-1} \\ a_{n-1} + a_{-1} & a_{n-2} + a_{-2} & \cdots & a_1 + a_{1-n} & a_0 \end{bmatrix} \tag{1.2}$$

and

$$S = \frac{1}{2} \begin{bmatrix} a_0 & a_{-1} - a_{n-1} & \cdots & a_{2-n} - a_2 & a_{1-n} - a_1 \\ a_1 - a_{1-n} & a_0 & \cdots & \cdots & a_{2-n} - a_2 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ a_{n-2} - a_{-2} & \cdots & \cdots & a_0 & a_{-1} - a_{n-1} \\ a_{n-1} - a_{-1} & a_{n-2} - a_{-2} & \cdots & a_1 - a_{1-n} & a_0 \end{bmatrix}. \tag{1.3}$$

Note that C is a circulant matrix and S is a skew-circulant matrix. A circulant matrix can be diagonalized by the discrete Fourier matrix F and a skew-circulant matrix can be diagonalized by a discrete Fourier matrix with diagonal scaling, i.e., $\hat{F} = F\Omega$. That is to say, it holds that

$$F^*CF = \Lambda_C, \quad \hat{F}^*S\hat{F} = \Lambda_S, \tag{1.4}$$

where

$$F = (F)_{j,k} = \frac{1}{\sqrt{n}} e^{\frac{2\pi i}{n}jk}, \quad 0 \leq j, k \leq s, \quad \Omega = \text{diag} \left(1, e^{-\frac{\pi i}{n}}, \dots, e^{-\frac{(n-1)\pi i}{n}} \right),$$

and i is the imaginary unit [12,13]. Λ_C and Λ_S are diagonal matrices formed by the eigenvalues of C and S , respectively, which can be obtained in $O(n \log n)$ operations by using the fast Fourier transform (FFT).

As is known, the Newton method may be the most popular, classic and important solver for a general system of nonlinear equations $F(x) = 0$, where $F : \mathbb{D} \subset \mathbb{C}^n \rightarrow \mathbb{C}^n$ is a continuously differentiable function. However, at each iteration step, the Newton method requires not only the computation of $F(x^{(k)})$ and $F'(x^{(k)})$, but also the exact solution of the corresponding Newton equation $F'(x^{(k)})\Delta x^{(k)} = -F(x^{(k)})$, which are very costly and complicated in actual applications [14,15]. In order to overcome these disadvantages and improve the efficiency of the Newton iteration method, a large number of modifications have been proposed to simplify or avoid computation of the Jacobian matrix and reduce the cost of the function evaluation; see [3,14,16–19].

For the weakly nonlinear system (1.1), based on the facts that the linear and the nonlinear terms Ax and $\phi(x)$ are well separated and the former is strongly dominant over the latter, Bai and Yang [3] presented the Picard-HSS and the nonlinear HSS-like iteration methods. The advantage of these methods over the Newton iteration method is that they do not require explicit construction and accurate computation of the Jacobian matrix, and only need to solve linear sub-systems of constant coefficient matrices. Hence, computational workloads and computer memory may be saved.

In this paper, based on the circulant and skew-circulant splitting (CSCS) of the Toeplitz matrix, we establish two classes of nonlinear composite splitting iteration schemes, called Picard-CSCS and nonlinear CSCS-like iteration methods, respectively, for solving the large scale Toeplitz system of weakly nonlinear equations (1.1). Compared with the Newton iteration method, both Picard-CSCS and nonlinear CSCS-like iteration methods neither require explicit form and accurate computation of Jacobian matrix, nor require solution of the changeable-coefficient linear sub-systems, which is similar to the Picard-HSS and the nonlinear HSS-like iteration methods initially introduced in [3]. Moreover, as the circulant and skew-circulant matrices can be diagonalized by the discrete Fourier matrix and diagonally scaled discrete Fourier matrix, respectively, the solutions of the two linear sub-systems can be efficiently obtained by using FFT. In addition, FFT is highly parallelizable and has been implemented on multiprocessors efficiently [20]. Hence, computational workloads may be further saved in actual implementations.

The organization of this paper is as follows. In Section 2, we review the CSCS iteration and give the Newton-CSCS iteration method. In Sections 3 and 4, we establish the Picard-CSCS and the nonlinear CSCS-like iteration methods, and discuss their convergence properties. Numerical results are given in Section 5. Finally, in Section 6 we draw a brief conclusion and give some remarks.

2. The CSCS and Newton-CSCS iteration methods

When the nonlinear term $\phi : \mathbb{D} \in \mathbb{C}^n \rightarrow \mathbb{C}^n$ is a constant vector, i.e., $\phi(x) = b$, the Toeplitz system of weakly nonlinear equations (1.1) reduces to the Toeplitz system of linear equations

$$Ax = b, \quad A \in \mathbb{C}^{n \times n} \text{ and } x, b \in \mathbb{C}^n. \tag{2.1}$$

Many direct methods are proposed for solving this linear system in earlier, but now iteration methods have gained much attention; see [11,21–25]. Based on the circulant and skew-circulant splitting (CSCS) of the Toeplitz matrix A , Ng established in [20] the following CSCS iteration method for solving the positive definite system of linear equations (2.1).

The CSCS iteration method: Given an initial guess $x^{(0)} \in \mathbb{C}^n$, compute $x^{(k+1)}$ for $k = 0, 1, 2, \dots$ using the following iteration scheme until $\{x^{(k)}\}$ satisfies the stopping criterion:

$$\begin{cases} (\alpha I + C)x^{(k+\frac{1}{2})} = (\alpha I - S)x^{(k)} + b, \\ (\alpha I + S)x^{(k+1)} = (\alpha I - C)x^{(k+\frac{1}{2})} + b, \end{cases} \tag{2.2}$$

where α is a given positive constant and I denotes the identity matrix.

We remark that the CSCS iteration method is a special case of the NSS iteration method in [26], which generalizes the HSS iteration method [27] to normal and skew-Hermitian splitting (NSS). For the convergence property of the CSCS iteration method, Ng applied a general convergence theory for a two-step splitting iteration in [27] to obtain the following result.

Lemma 2.1 ([20]). *Let C and S be the circulant and the skew-circulant matrices given in (1.2) and (1.3), and α be a positive constant. If C and S are positive definite, then the iteration matrix $T(\alpha)$ of the CSCS iteration is given by*

$$T(\alpha) = (\alpha I + S)^{-1}(\alpha I - C)(\alpha I + C)^{-1}(\alpha I - S), \tag{2.3}$$

and its spectral radius $\rho(T(\alpha))$ is bounded by

$$\sigma(\alpha) \equiv \max_{\lambda_j \in \lambda(C)} \frac{|\alpha - \lambda_j|}{|\alpha + \lambda_j|} \cdot \max_{\mu_j \in \lambda(S)} \frac{|\alpha - \mu_j|}{|\alpha + \mu_j|}.$$

And it holds that

$$\rho(T(\alpha)) \leq \sigma(\alpha) < 1, \quad \forall \alpha > 0,$$

i.e., the CSCS iteration converges to the exact solution $x_* \in \mathbb{C}^n$ of the system of linear equation $Ax = b$.

As the Newton method is the most classic and important iteration method for a general nonlinear system, we consider using it to solve the weakly nonlinear system (1.1). When the Jacobian matrix of the nonlinear function $F(x)$ at the solution point $x_* \in \mathbb{D}$, denoted as $F'(x_*)$, is a Toeplitz matrix, following the construction of the Newton-HSS iteration scheme in [18], we can use the CSCS iteration method to approximate the solution of the Newton equation instead of solving it exactly (i.e., using the Newton iteration as the outer iteration and the CSCS iteration as the inner iteration), which is algorithmically described as follows.

The Newton-CSCS iteration method. Let $F : \mathbb{D} \subset \mathbb{C}^n \rightarrow \mathbb{C}^n$ be a continuously differentiable function with the Toeplitz Jacobian matrix $F'(x)$ at any $x \in \mathbb{D}$, $C(x)$ and $S(x)$ be the circulant and skew-circulant parts of $F'(x) = C(x) + S(x)$, respectively, and $C(x)$ and $S(x)$ be positive definite matrices. Given an initial guess $x^{(0)} \in \mathbb{D}$ and a sequence $\{l_k\}_{k=0}^\infty$ of positive integers, compute $x^{(k+1)}$ for $k = 0, 1, 2, \dots$ using the following iteration scheme until $\{x^{(k)}\}$ satisfies the stopping criterion:

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

$$\begin{cases} (\alpha I + C(x^{(k)}))s^{(k,l+\frac{1}{2})} = (\alpha I - S(x^{(k)}))s^{(k,l)} - F(x^{(k)}), \\ (\alpha I + S(x^{(k)}))s^{(k,l+1)} = (\alpha I - C(x^{(k)}))s^{(k,l+\frac{1}{2})} - F(x^{(k)}), \end{cases}$$

where α is a given positive constant and I denotes the identity matrix;

- (c) $x^{(k+1)} := x^{(k)} + s^{(k,l_k)}$.

At each iteration step, say k , the Newton-CSCS iteration method requires to solve two linear sub-systems with the shifted circulant coefficient matrix $\alpha I + C(x^{(k)})$ and the shifted skew-circulant coefficient matrix $\alpha I + S(x^{(k)})$. The advantage of this method is that the coefficient matrices of the two linear sub-systems can be diagonalized and well adapted for parallel computing. But the disadvantages are that it requires the explicit form of the Jacobian matrix $F'(x^{(k)})$ at the current iterate $x^{(k)}$ and that the coefficient matrices of the two linear sub-systems are varying with respect to the iteration index k , so the computations of the Newton-CSCS iteration could be much more expensive. Moreover, it is not easy to ensure that $F'(x^{(k)})$ is a Toeplitz matrix. Therefore, this method is not always feasible.

3. The Picard-CSCS iteration method

In order to overcome the above-mentioned disadvantages of the Newton-CSCS iteration method, recalling that the linear and the nonlinear terms Ax and $\phi(x)$ are well separated and the former is strongly dominant over the latter, we can use the Picard iteration method

$$Ax^{(k+1)} = \phi(x^{(k)}), \quad k = 0, 1, 2, \dots,$$

to solve the system of Toeplitz weakly nonlinear equations (1.1); see [3,14,16]. When the matrix $A \in \mathbb{C}^{n \times n}$ is Toeplitz and its circulant part C and skew-circulant part S are all positive definite, the next iterate $x^{(k+1)}$ may be approximately computed by the CSCS iteration method. This naturally leads to a class of nonlinear composite iteration scheme, called Picard-CSCS iteration method.

The Picard-CSCS iteration method. Let $\phi : \mathbb{D} \subset \mathbb{C}^n \rightarrow \mathbb{C}^n$ be a continuously differentiable function and $A \in \mathbb{C}^{n \times n}$ be a Toeplitz matrix. Suppose that $A = C + S$, where C and S are the circulant and skew-circulant parts of A given in (1.2) and (1.3) and they are positive definite. Given an initial guess $x^{(0)} \in \mathbb{D}$ and a sequence $\{l_k\}_{k=0}^{\infty}$ of positive integers, compute $x^{(k+1)}$ for $k = 0, 1, 2, \dots$ using the following iteration scheme until $\{x^{(k)}\}$ satisfies the stopping criterion:

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

$$\begin{cases} (\alpha I + C)x^{(k,l+\frac{1}{2})} = (\alpha I - S)x^{(k,l)} + \phi(x^{(k)}), \\ (\alpha I + S)x^{(k,l+1)} = (\alpha I - C)x^{(k,l+\frac{1}{2})} + \phi(x^{(k)}), \end{cases}$$

where α is a given positive constant and I denotes the identity matrix;

- (c) $x^{(k+1)} := x^{(k,l_k)}$.

Of course, the Picard-CSCS iteration method can be easily reformulated into residual-updating forms by replacing the main steps (a)–(c) as follows.

- (a') $s^{(k,0)} := 0, b^{(k)} := \phi(x^{(k)}) - Ax^{(k)}$;
 (b') For $l = 0, 1, 2, \dots, l_k - 1$, solve the following linear systems to obtain $s^{(k,l+1)}$:

$$\begin{cases} (\alpha I + C)s^{(k,l+\frac{1}{2})} = (\alpha I - S)s^{(k,l)} + b^{(k)}, \\ (\alpha I + S)s^{(k,l+1)} = (\alpha I - C)s^{(k,l+\frac{1}{2})} + b^{(k)}, \end{cases}$$

where α is a given positive constant and I denotes the identity matrix;

- (c') $x^{(k+1)} := x^{(k)} + s^{(k,l_k)}$.

Clearly, the Picard-CSCS iteration method uses a CSCS linear iteration method to approximate the solution of the linear sub-system instead of solving it exactly. The advantages of this method are that it often requires only a small amount of storage and that, when $x^{(k)}$ is near to x_* , it can take advantage of the fact that zeros is a good initial approximation to $s^{(k,0)}$. The Picard-CSCS iteration method can be equivalently rewritten as

$$x^{(k+1)} = T(\alpha)^{l_k} x^{(k)} + \sum_{j=0}^{l_k-1} T(\alpha)^j G(\alpha) \phi(x^{(k)}), \quad k = 0, 1, 2, \dots, \quad (3.1)$$

where

$$T(\alpha) = (\alpha I + S)^{-1}(\alpha I - C)(\alpha I + C)^{-1}(\alpha I - S) \quad (3.2)$$

and

$$G(\alpha) = 2\alpha(\alpha I + S)^{-1}(\alpha I + C)^{-1}. \quad (3.3)$$

Here, $T(\alpha)$ is the iteration matrix of the inner CSCS method. The sequence of outer iteration $\{x^{(k)}\}$ depends upon the CSCS inner iteration and the criteria used to stop the inner iteration.

Compared with the Newton-CSCS iteration method, the Picard-CSCS iteration method neither computes and stores the actual Jacobian matrices, nor requires differentiability of the function $\phi(x)$, which are analogous the Picard-HSS iteration method. In addition, the two linear sub-systems in all CSCS inner iterations have the same shifted circulant and shifted skew-circulant coefficient matrices, which are constant with respect to the iteration index k . Moreover, the exact solutions can be efficiently obtained by using fast Fourier transforms (FFTs). Hence, the computations of the Picard-CSCS iteration could be much cheaper.

Now, we consider the local convergence, as well as the convergence rate and the convergence factor of the Picard-CSCS iteration method. By utilizing the R-convergence concept and modifying Theorem 3.1 in [3], we can establish the following local convergence theory for the Picard-CSCS iteration method.

Theorem 3.1. Let $\phi : \mathbb{D} \subset \mathbb{C}^n \rightarrow \mathbb{C}^n$ be G -differentiable on an open neighborhood $\mathbb{N}_0 \subset \mathbb{D}$ of a point $x_* \in \mathbb{D}$ at which $\phi'(x)$ is continuous and $F(x_*) := Ax_* - \phi(x_*) = 0$. Suppose that C and S are the circulant and the skew-circulant parts of the matrix $A = C + S$ given in (1.2) and (1.3), and both C and S are positive definite matrices. Denote by

$$\theta(\alpha) = \|T(\alpha)\|, \quad \alpha = \|A^{-1}\|, \quad \beta = \|A^{-1}\phi'(x_*)\|. \tag{3.4}$$

Then there exists an open neighborhood $\mathbb{N} \subset \mathbb{N}_0$ of x_* such that for any $x^{(0)} \in \mathbb{N}$ and any sequence of positive integers $l_k, k = 0, 1, 2, \dots$, the iteration sequence $\{x^{(k)}\}_{k=0}^\infty$ generated by the Picard-CSCS iteration method is well defined and convergent to x_* , provided $\beta < 1$ and $l_0 = \liminf_{k \rightarrow \infty} l_k \geq \lfloor \ln(\frac{1-\beta}{1+\beta}) / \ln(\theta) \rfloor$ (where $\lfloor \cdot \rfloor$ is used to denote the smallest integer no less than the corresponding real number).

Moreover, it holds that

$$\limsup_{k \rightarrow \infty} \|x^{(k)} - x_*\|^{\frac{1}{k}} \leq \beta + (1 + \beta)\theta(\alpha)^{l_0}.$$

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

$$\limsup_{k \rightarrow \infty} \|x^{(k)} - x_*\|^{\frac{1}{k}} \leq \beta.$$

Proof. The proof uses arguments similar to those in the proof of the convergence theorem of the Picard-HSS iteration method; see [3]. In fact, we only need to replace the Hermitian matrix H and the skew-Hermitian matrix S of the convergence theorem of the Picard-HSS iteration method by the circulant matrix C and the skew-circulant matrix S , and then obtain the convergence theorem of the Picard-CSCS iteration method. \square

Theorem 3.1 shows that the Picard-CSCS iteration has the same convergence property as the Picard-HSS iteration, which is essentially determined by the quantities β and $\theta(\alpha)$. Small β and $\theta(\alpha)$ will lead to fast convergence of the Picard-CSCS iteration.

4. The nonlinear CSCS-like iteration method

Since the Picard-CSCS and the Picard-HSS iteration methods share the same drawback, that is, the numbers of the inner iteration steps $l_k, k = 0, 1, 2, \dots$, are often problem dependent and difficult to be determined in actual computations, we propose the following nonlinear CSCS-like iteration method to overcome this disadvantage.

The nonlinear CSCS-like iteration method. Let $\phi : \mathbb{D} \subset \mathbb{C}^n \rightarrow \mathbb{C}^n$ be a continuously differentiable function and $A \in \mathbb{C}^{n \times n}$ be a positive definite Toeplitz matrix. Suppose that C and S are the circulant and the skew-circulant parts of $A = C + S$ given in (1.2) and (1.3), C and S are positive definite matrices. Given an initial guess $x^{(0)} \in \mathbb{D}$, compute $x^{(k+1)}$ for $k = 0, 1, 2, \dots$ using the following iteration scheme until $\{x^{(k)}\}$ satisfies the stopping criterion:

$$\begin{cases} (\alpha I + C)x^{(k+\frac{1}{2})} = (\alpha I - S)x^{(k)} + \phi(x^{(k)}), \\ (\alpha I + S)x^{(k+1)} = (\alpha I - C)x^{(k+\frac{1}{2})} + \phi(x^{(k+\frac{1}{2})}), \end{cases}$$

where α is a given positive constant and I denotes the identity matrix.

In the following, we deduce the convergence property for the nonlinear CSCS-like iteration method. We define

$$\begin{aligned} U(x) &= (\alpha I + C)^{-1}((\alpha I - S)x + \phi(x)), \\ V(x) &= (\alpha I + S)^{-1}((\alpha I - C)x + \phi(x)) \end{aligned} \tag{4.1}$$

and

$$\psi(x) = V \circ U(x) := V(U(x)),$$

then the nonlinear CSCS-like iteration scheme can be equivalently expressed as

$$x^{(k+1)} = \psi(x^{(k)}), \quad k = 0, 1, 2, \dots \tag{4.2}$$

Suppose that $x_* \in \mathbb{D}$ is a solution of the system of weakly nonlinear equations (1.1). Then we can easily verify the following fact by using the chain rule for derivative:

$$\begin{aligned} \psi'(x_*) &= V'(x_*)U'(x_*) \\ &= (\alpha I + S)^{-1}(\alpha I - C + \phi'(x_*))(\alpha I + C)^{-1}(\alpha I - S + \phi'(x_*)). \end{aligned}$$

By making use of the Ostrowski theorem [6], we know that if $\rho(\psi'(x_*)) < 1$, then x_* is a point of attraction of the nonlinear CSCS-like iteration. Then we can obtain the following local convergence theory for the CSCS-like iteration method.

Theorem 4.1. Assume that $\phi : \mathbb{D} \subset \mathbb{C}^n \rightarrow \mathbb{C}^n$ is F -differentiable at a point $x_* \in \mathbb{D}$ such that $Ax_* = \phi(x_*)$. Suppose that C and S are the circulant and the skew-circulant parts of the matrix $A = C + S$ given in (1.2) and (1.3), and C and S are positive definite matrices. Denote by

$$T(\alpha; x_*) = (\alpha I + S)^{-1}(\alpha I - C + \phi'(x_*))(\alpha I + C)^{-1}(\alpha I - S + \phi'(x_*)).$$

If $\rho(T(\alpha; x_*)) < 1$, then $x_* \in \mathbb{D}$ is a point of attraction of the nonlinear CSCS-like iteration method.

Now, we apply the above result to obtain the following theorem.

Theorem 4.2. Assume the conditions of Theorem 4.1 be satisfied. Denote by

$$\begin{aligned} \delta &= \max\{\|\phi'(x_*)(\alpha I + S)^{-1}\|, \|\phi'(x_*)(\alpha I + C)^{-1}\|\}, \\ \xi &= \max\{\|(\alpha I - S)(\alpha I + S)^{-1}\|, \|(\alpha I - C)(\alpha I + C)^{-1}\|\}, \end{aligned}$$

and the matrix $T(\alpha)$ is defined in (2.3). If $\delta < 1 - \xi$, then $\rho(T(\alpha; x_*)) < 1$.

Proof. Since

$$\begin{aligned} T(\alpha; x_*) &= (\alpha I + S)^{-1}(\alpha I - C)(\alpha I + C)^{-1}(\alpha I - S) + (\alpha I + S)^{-1}(\alpha I - C)(\alpha I + C)^{-1}\phi'(x_*) \\ &\quad + (\alpha I + S)^{-1}\phi'(x_*)(\alpha I + C)^{-1}(\alpha I - S) + (\alpha I + S)^{-1}\phi'(x_*)(\alpha I + C)^{-1}\phi'(x_*) \end{aligned}$$

and

$$\|T(\alpha)\| \leq \|(\alpha I - C)(\alpha I + C)^{-1}\| \|(\alpha I - S)(\alpha I + S)^{-1}\| \leq \xi^2,$$

we have

$$\begin{aligned} \|T(\alpha; x_*)\| &= \|(\alpha I + S)T(\alpha; x_*)(\alpha I + S)^{-1}\| \\ &= \|(\alpha I + S)T(\alpha)(\alpha I + S)^{-1} + (\alpha I - C)(\alpha I + C)^{-1}\phi'(x_*)(\alpha I + S)^{-1} \\ &\quad + \phi'(x_*)(\alpha I + C)^{-1}(\alpha I - S)(\alpha I + S)^{-1} + \phi'(x_*)(\alpha I + C)^{-1}\phi'(x_*)(\alpha I + S)^{-1}\| \\ &\leq \|(\alpha I + S)T(\alpha)(\alpha I + S)^{-1}\| + \|(\alpha I - C)(\alpha I + C)^{-1}\phi'(x_*)(\alpha I + S)^{-1}\| \\ &\quad + \|\phi'(x_*)(\alpha I + C)^{-1}(\alpha I - S)(\alpha I + S)^{-1}\| + \|\phi'(x_*)(\alpha I + C)^{-1}\phi'(x_*)(\alpha I + S)^{-1}\| \\ &\leq \|T(\alpha)\| + \|(\alpha I - C)(\alpha I + C)^{-1}\| \|\phi'(x_*)(\alpha I + S)^{-1}\| \\ &\quad + \|\phi'(x_*)(\alpha I + C)^{-1}\| \|(\alpha I - S)(\alpha I + S)^{-1}\| + \|\phi'(x_*)(\alpha I + C)^{-1}\| \|\phi'(x_*)(\alpha I + S)^{-1}\| \\ &\leq \xi^2 + 2\xi\delta + \delta^2 = (\xi + \delta)^2. \end{aligned}$$

Now, under the condition $\delta < 1 - \xi$, we easily obtain $\rho(T(\alpha; x_*)) \leq \|T(\alpha; x_*)\| < 1$. \square

Theorem 4.2 shows that $\rho(T(\alpha; x_*)) < 1$ is valid if $\phi'(x_*)$ is reasonably small compared with the matrix A . Otherwise, the nonlinear CSCS-like iteration may be convergent slowly or even divergent.

5. Numerical results

In this section, we illustrate the effectiveness of our Picard-CSCS and nonlinear CSCS-like iteration methods by solving the model problem (5.1).

All tests are started from the zero vector, performed in MATLAB R2009a with machine precision 10^{-16} , and terminated when the current iteration satisfies

$$\frac{\|F(x^{(k)})\|_2}{\|F(x^{(0)})\|_2} \leq 10^{-6}.$$

In addition, the stopping criteria for the inner iterations of the Picard-HSS, the Picard-CSCS and the Newton-GMRES (5) methods are set to be

$$\frac{\|F'(x^{(k)})S^{(k, l_k)} + F(x^{(k)})\|_2}{\|F(x^{(k)})\|_2} \leq \eta_k,$$

where l_k is the number of the inner iteration steps and η_k is the prescribed tolerance for controlling the accuracy of the inner iterations at the k -th outer iterate. In our test, η_k is fixed for all k , which is simply denoted as η .

The two sub-systems of linear equations involved are solved in the way if $Ax = b$, then $x = A^{-1}b$. Moreover, if the two sub-systems of linear equations involved in the Picard-CSCS and the nonlinear CSCS-like iteration methods are solved by making use of the method presented in [28] and using parallel computing, the numerical results of the Picard-CSCS and the nonlinear CSCS-like iteration methods must be better.

In actual computations, we adopt the optimal parameters $\alpha_{\text{HSS}} = \sqrt{\lambda_{\min}\lambda_{\max}}$ given in [27] for the Picard-HSS and the nonlinear HSS-like methods, where λ_{\min} and λ_{\max} are the lower and the upper bounds for the eigenvalues of the Hermitian

Table 1
The optimal values α for Example 1.

N	10	20	40	80	160
α_{CSCS}	45.4545	23.8095	12.1951	6.1728	3.1056
α_{HSS}	25.0486	6.7992	1.7140	0.4012	0.0822

part H of the coefficient matrix A , and adopt the optimal parameters α_{CSCS} given in Theorem 2 in [20] for the Picard-CSCS and the nonlinear CSCS-like methods. Note that they only minimize the bound of the convergence factor of the iteration matrix, but not the spectral radius of the iteration matrix. Admittedly, the optimal parameters α^* are crucial for guaranteeing fast convergence speeds of these parameter-dependent iteration methods, but they are generally very difficult to be determined.

Example 1 ([29]). Consider the nonlinear two-point boundary-value problem with a convective dominated term in one-dimensional setting:

$$\begin{cases} -\varepsilon \frac{d^2 u}{dx^2} + b(x) \frac{du}{dx} = f(u, x), & x \in (0, 1), \\ u(0) = u(1) = 0, \end{cases} \tag{5.1}$$

with $\varepsilon = 1$, $b(x) = 1000$ and $f(u, x) = 10 \sin(u + 1)$. The problem (5.1) is a singularly perturbed problem.

By applying the central difference formula to approximate the second order derivation $\frac{d^2 u}{dx^2}$, and the backward difference formula to approximate the first order derivation $\frac{du}{dx}$, with the step size $\Delta x = 1/(N + 1)$, $x_j = j\Delta x$. We obtain a system of weakly nonlinear equations of the form (1.1).

Table 1 lists the optimal parameters α_{CSCS} and α_{HSS} for Example 1. We note that with the increase of the numbers of the mesh point N , the optimal values α_{CSCS} and α_{HSS} are decreasing quickly, and the former is larger than the latter.

Table 2 lists numerical results corresponding to five iteration methods for Example 1. The Picard-CSCS and the nonlinear CSCS-like methods are compared with the Picard-HSS, the nonlinear HSS-like and the Newton-GMRES (5) methods with different N ($N = 10, 20, 40, 80, 160$) and tolerance η ($\eta = 0.1, 0.01, 0.001$) on aspects of numbers of outer, inner and total iteration steps (denoted as IT_{out} , IT_{int} and IT , respectively) and total CPU times (denoted as CPU), where IT_{int} denotes the average number of inner iteration steps at each outer iterate. From this table, we can see that all the experimented methods can successfully produce approximate solutions to the system of weakly nonlinear equations for all of the numbers of mesh point N .

When the tolerance η for controlling the accuracy of the inner iterations becomes small and N is fixed, the number of inner iteration steps, the number of total iteration steps and the amount of CPU times of the Picard-HSS and the Picard-CSCS are increasing, but the number of outer iteration steps are decreasing. Of course, the computing efficiency of the Picard-HSS and the Picard-CSCS methods depends on the tolerances used for controlling the accuracy of the inner iterations. Hence, good choice of this quantity for the Picard-CSCS and the Picard-HSS methods should be an important and interesting topic for future study.

When N is increasing, the number of outer iteration steps are fixed or increasing slightly, but the number of inner iteration steps are increasing quickly. The number of total iteration steps and the total CPU times of all iteration methods are increasing quickly, especially for the Picard-HSS and the nonlinear HSS-like iteration methods.

To show that the proposed iteration methods can also be applied to solve complex system of nonlinear equations (1.1), we construct and test the following example, which is a Toeplitz system of weakly nonlinear equations with complex matrix.

Example 2. $A \in \mathbb{C}^{n \times n}$ is a nonsymmetric, complex, sparse and positive definite Toeplitz matrix with $A(j, j) = 10$, $A(j, j + 1) = -2i$, $A(j + 1, j) = \frac{1}{2} + 2i$, $A(j, j + 2) = -3i$, $A(j + 2, j) = \frac{1}{2} + 3i$, $j = 1, 2, 3, \dots, n$, and $\phi(x) = \sin(x)$.

Table 3 lists the optimal parameters α_{CSCS} and α_{HSS} for Example 2. We note that with the increase of the matrix dimension N , the optimal parameters α_{CSCS} and α_{HSS} are almost fixed or decreasing slightly.

In Table 4, we list numerical results corresponding to four iteration methods for Example 2, i.e., the Picard-HSS, the nonlinear HSS-like, the Picard-CSCS and the nonlinear CSCS-like iteration methods. From this table, we see that all the experimented methods can successfully produce approximate solutions to the system of weakly nonlinear equations for all of the matrix dimensions.

When the matrix dimension N is increasing, the number of outer and inner iteration steps are almost fixed for all iteration methods, and the number of total iteration steps show the similar phenomena. But the total CPU times for all iteration methods are increasing quickly. Moreover, in terms of outer iteration steps, The Picard-HSS and the Picard-CSCS have almost the same results, but in terms of inner iteration steps, the Picard-CSCS iteration method is better than the Picard-HSS iteration method.

When the tolerance η becomes small and N is fixed, both numbers of inner iteration steps and total iteration steps of the Picard-HSS and the Picard-CSCS are increasing, but the amount of CPU times of the Picard-CSCS is less than that of the

Table 2
The numbers of iterations and total CPU times for Example 1.

N			10	20	40	80	160
$\eta = 0.1$	Picard-HSS	Γ_{out}	6	6	6	6	6
		Γ_{inn}	6.6667	13.8333	29.6667	62.6667	135.3333
		IT	40	83	178	376	812
		CPU	0.0012	0.0020	0.0056	0.0286	0.3408
	Picard-CSCS	Γ_{out}	5	5	5	6	6
		Γ_{inn}	2.6	3.4	5.2	8	16.1667
		IT	13	17	26	48	97
		CPU	0.0007	0.0008	0.0015	0.0061	0.0527
	Newton-GMRES	Γ_{out}	4	4	4	6	8
		Γ_{inn}	9	10.75	14.5	20.6667	23.75
		IT	36	43	58	124	190
		CPU	0.0138	0.0128	0.0192	0.0350	0.0804
$\eta = 0.01$	Picard-HSS	Γ_{out}	3	3	3	3	4
		Γ_{inn}	13	29.3333	61	137.3333	280.5
		IT	39	88	183	412	1122
		CPU	0.0010	0.0020	0.0055	0.0287	0.4648
	Picard-CSCS	Γ_{out}	3	3	3	3	3
		Γ_{inn}	4.3333	7	10.6667	19.3333	37
		IT	13	21	32	58	111
		CPU	0.0005	0.0008	0.0016	0.0064	0.0576
	Newton-GMRES	Γ_{out}	3	3	3	4	6
		Γ_{inn}	10	20	20.3333	29	28.3333
		IT	30	60	61	116	170
		CPU	0.0084	0.0148	0.0216	0.0321	0.0762
$\eta = 0.001$	Picard-HSS	Γ_{out}	3	3	3	3	3
		Γ_{inn}	21	44	96.6667	214.6667	463.6667
		IT	63	132	290	644	1391
		CPU	0.0021	0.0027	0.0082	0.0436	0.5733
	Picard-CSCS	Γ_{out}	3	3	3	3	3
		Γ_{inn}	6	9	14.3333	24.6667	44.6667
		IT	18	27	43	74	134
		CPU	0.0014	0.0009	0.0019	0.0075	0.0678
	Newton-GMRES	Γ_{out}	3	3	2	3	5
		Γ_{inn}	10	20	40	33	42
		IT	30	60	80	99	210
		CPU	0.0083	0.0148	0.0191	0.0309	0.0779
HSS-like	IT	37	81	174	371	797	
	CPU	0.0026	0.0058	0.0148	0.0507	0.4330	
CSCS-like	IT	8	9	14	24	43	
	CPU	0.0008	0.0009	0.0018	0.0058	0.0352	

Table 3
The optimal values α for Example 2.

N	40	80	160	320	640
α_{CSCS}	3.2821	3.2771	3.2761	3.2759	3.2758
α_{HSS}	7.9260	7.8866	7.8762	7.8735	7.8728

Picard-HSS, and they all reach the least when $\eta = 0.01$. Hence, good choice of the tolerance η for the Picard-CSCS and the Picard-HSS methods should be also an important problem, which could decide the computing efficiency of the Picard-HSS and the Picard-CSCS methods.

From Tables 2 and 4, we observe that the nonlinear CSCS-like and the Picard-CSCS iteration methods performs better than the Newton-GMRES (5), the nonlinear HSS-like and the Picard-HSS iteration methods in terms of iteration steps and CPU times for solving the Toeplitz system of weakly nonlinear equations. In particular, the nonlinear CSCS-like method often does better than the Picard-CSCS method in our implementations.

6. Conclusion and remarks

For large scale Toeplitz system of weakly nonlinear equations, we have established the Picard-CSCS and the nonlinear CSCS-like iteration methods. They are all based on the separability and strong dominance between the linear and the nonlinear terms and the circulant and skew-circulant splitting (CSCS) iteration technique. Both theoretical analysis and numerical experiments have shown that the Picard-CSCS and the nonlinear CSCS-like iteration methods are feasible and

Table 4
The numbers of iterations and the total CPU times for Example 2.

		N	40	80	160	320	640
$\eta = 0.1$	Picard-HSS	IT_{out}	6	7	7	6	6
		IT_{int}	2.1667	2.1429	2.1429	2	2
		IT	13	15	15	12	12
		CPU	0.0049	0.0102	0.0477	0.2732	1.7712
	Picard-CSCS	IT_{out}	5	5	5	5	5
		IT_{int}	2	2	2	2	2
IT		10	10	10	10	10	
CPU		0.0039	0.0086	0.0380	0.2399	1.6595	
$\eta = 0.01$	Picard-HSS	IT_{out}	5	5	5	5	5
		IT_{int}	4	4	4	4	4
		IT	20	20	20	20	20
		CPU	0.0033	0.0089	0.0497	0.2922	1.9033
	Picard-CSCS	IT_{out}	5	5	5	5	5
		IT_{int}	3	3	3	3	3
IT		15	15	15	15	15	
CPU		0.0021	0.0077	0.0440	0.2467	1.7461	
$\eta = 0.001$	Picard-HSS	IT_{out}	5	5	5	5	5
		IT_{int}	6	6	6	6	6
		IT	30	30	30	30	30
		CPU	0.0037	0.0096	0.0618	0.3426	2.0800
	Picard-CSCS	IT_{out}	5	5	5	5	5
		IT_{int}	4	4	4	4	4
IT		20	20	20	20	20	
CPU		0.0048	0.0080	0.0459	0.2770	1.8639	
HSS-like	IT	12	12	12	12	11	
	CPU	0.0020	0.0036	0.0154	0.0588	0.2362	
CSCS-like	IT	8	8	7	7	7	
	CPU	0.0012	0.0033	0.0078	0.0336	0.1566	

efficient nonlinear solvers. In particular, the nonlinear CSCS-like method often does better than the Picard-CSCS method in our implementations.

Since the matrix splitting among these iteration methods is a Toeplitz splitting, these new iteration methods can be applied only to the Toeplitz system of weakly nonlinear equations whose linear terms Ax are strongly dominant over the nonlinear term $\phi(x)$ in certain norm and the matrix A is Toeplitz.

Acknowledgments

The authors are very grateful to Professor Zhong-Zhi Bai for many valuable suggestions and thorough modifications which greatly improved the original manuscript of the paper. They are very much indebted to the anonymous referees and the editor Michael Ng, for their valuable comments. The work is supported in part by NSF under grant (No. 11026011) and ministry of special funds for scientific research on public causes (No. GYHY201006035).

References

- [1] Z.-Z. Bai, Parallel nonlinear AOR method and its convergence, *Comput. Math. Appl.* 31 (1996) 21–31.
- [2] Z.-Z. Bai, The monotone convergence rate of the parallel nonlinear AOR method, *Comput. Math. Appl.* 31 (1996) 1–8.
- [3] Z.-Z. Bai, X. Yang, On HSS-based iteration methods for weakly nonlinear systems, *Appl. Numer. Math.* 59 (2009) 2923–2936.
- [4] Z.-Z. Bai, Y.-M. Huang, M.K. Ng, On preconditioned iterative methods for Burgers equations, *SIAM J. Sci. Comput.* 29 (2007) 415–439.
- [5] 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.
- [6] J.M. Ortega, W.C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic press, New York, 1970.
- [7] Z. Zhang, Finite element superconvergence approximation for one-dimensional singularly perturbed problems, *Numer. Meth. Part. D. E.* 18 (2002) 374–395.
- [8] T. Tang, Superconvergence of numerical solutions to weakly singular Volterra integro-differential equations, *Numer. Math.* 61 (1992) 373–382.
- [9] Z.-Z. Bai, Optimal parameters in the HSS-like methods for saddle-point problems, *Numer. Linear Algebra Appl.* 16 (2009) 447–479.
- [10] M. Benzi, M.K. Ng, Preconditioned iterative methods for weighted Toeplitz least squares problems, *SIAM J. Matrix Anal. Appl.* 27 (2006) 1106–1124.
- [11] R.H. Chan, M.K. Ng, Conjugate gradient methods for Toeplitz systems, *SIAM Rev.* 38 (1996) 427–482.
- [12] D. Bertaccini, A circulant preconditioner for the systems of LMF-based ODE codes, *SIAM J. Sci. Comput.* 22 (2001) 767–786.
- [13] D. Bertaccini, M.K. Ng, Block $\{\omega\}$ -circulant preconditioners for the systems of differential equations, *Calcolo* 40 (2003) 71–90.
- [14] 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.
- [15] E. Galligani, The Newton-arithmetic mean method for the solution of systems of nonlinear equations, *Appl. Math. Comput.* 134 (2003) 9–34.
- [16] H.-B. An, Z.-Z. Bai, NGLM: a globally convergent Newton-GMRES method, *Math. Numer. Sin.* 27 (2005) 151–174.
- [17] Z.-Z. Bai, A class of two-stage iterative methods for systems of weakly nonlinear equations, *Numer. Algorithms* 14 (1997) 295–319.
- [18] 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.

- [19] A.H. Sherman, On Newton-iterative methods for the solution of systems of nonlinear equations, *SIAM J. Numer. Anal.* 15 (1978) 755–771.
- [20] M.K. Ng, Circulant and skew-circulant splitting methods for Toeplitz systems, *J. Comput. Appl. Math.* 159 (2003) 101–108.
- [21] G.S. Ammar, W.B. Gragg, Superfast solutions of real positive definite Toeplitz systems, *SIAM J. Matrix Anal. Appl.* 9 (1988) 61–76.
- [22] R.P. Brent, F.G. Gustavson, D.Y.Y. Yun, Fast solution of Toeplitz systems of equations and computation of Padé approximants, *J. Algorithm.* 1 (1980) 259–295.
- [23] J.R. Bunch, Stability of methods for solving Toeplitz systems of equations, *SIAM J. Sci. Comput.* 6 (1985) 349–364.
- [24] M.K. Ng, Band preconditioners for block-Toeplitz-Toeplitz-block systems, *Linear Algebra Appl.* 259 (1997) 307–327.
- [25] M.K. Ng, *Iterative Methods for Toeplitz Systems*, Oxford University Press, 2004.
- [26] Z.-Z. Bai, G.H. Golub, M.K. Ng, On successive overrelaxation acceleration of the Hermitian and skew-Hermitian splitting iterations, *Numer. Linear Algebra Appl.* 14 (2007) 319–335.
- [27] 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.
- [28] M. Chen, On the solution of circulant linear systems, *SIAM J. Numer. Anal.* 24 (1987) 668–683.
- [29] I. Boglaev, A block monotone domain decomposition algorithm for a semilinear convection-diffusion problem, *J. Comput. Appl. Math.* 173 (2005) 259–277.