Journal of Computational and Applied Mathematics 250 (2013) 16–27

Contents lists available at SciVerse ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

A class of iteration methods based on the generalized preconditioned Hermitian and skew-Hermitian splitting for weakly nonlinear systems☆

Zhao-Nian Pu, Mu-Zheng Zhu *

School of Mathematics and Statistics, Hexi University, Zhangye 734000, PR China

ARTICLE INFO

Article history:
Received 24 July 2012
Received in revised form 20 January 2013

MSC:
65F10
65F50
65T50
65W05

Keywords:
Weakly nonlinear systems
GPHSS iteration method
HSS iteration method
Inner/outer iteration scheme

ABSTRACT

For large sparse systems of weakly nonlinear equations, based on the separable property and strong dominance between the linear and the nonlinear terms, Bai and Yang studied two nonlinear composite iteration schemes called Picard-HSS and nonlinear HSS-like methods (see [Z.-Z. Bai, X. Yang, On HSS-based iteration methods for weakly nonlinear systems, Appl. Numer. Math. 59 (12) (2009) 2923–2936]). In this paper, we generalize these methods and propose a class of generalized nonlinear composite splitting iteration schemes called Picard-GPHSS and nonlinear GPHSS-like iteration methods, to solve the large sparse systems of weakly nonlinear equations. We derive conditions for guaranteeing the local convergence of these iterative methods and derive some special case of iterative methods by choosing different parameters and preconditioned matrices. Numerical experiments are used to demonstrate the feasibility and effectiveness. And an efficient preconditioner is presented for the new methods in actual implementation. The efficiency is effectively testified by some comparisons with numerical results.

1. Introduction

We consider the solution of the system of weakly nonlinear equations

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

where $A \in \mathbb{C}^{n\times n}$ is a large, sparse, positive real matrix, and $\phi : \mathbb{D} \subset \mathbb{C}^n \to \mathbb{C}^n$ is a continuously differentiable function. The system of nonlinear equations (1.1) is said to be weakly nonlinear if the linear term Ax is strongly dominant over the nonlinear term $\phi(x)$ in certain norm [1,2].

The system of weakly nonlinear equations (1.1) often arise in many areas of scientific computing and engineering applications, and in particular in discretions of certain nonlinear partial differential equations [3–7], in collocation approximations of nonlinear integral equation [8], and in saddle point problems from image processing [9,10].

For a general system of nonlinear equations $F(x) = 0$, the Newton iteration method is the most popular, classic and important one

$$F'(x^{(k)})\Delta x^{(k)} = -F(x^{(k)}), \qquad x^{(k+1)} = x^{(k)} + \Delta x^{(k)},$$

☆ This paper has been supported by Scientific Research Foundation for Advisor Program of Higher Education of Gansu (1009 - 6) and Scientific Research Innovation and Application Foundation of President of Hexi University (xz2012).

* Corresponding author.
E-mail addresses: puzhaonian@163.com (Z.-N. Pu), zhumzh07@yahoo.com, zhumzh07@lzu.edu.cn (M.-Z. Zhu).

0377-0427/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.cam.2013.02.021

where $F : \mathbb{D} \subset \mathbb{C}^n \to \mathbb{C}^n$ is a continuously differentiable function. However, at each iteration step, the Newton method requires not only the explicit form of the Jacobian matrix but also the exact solution of the corresponding Newton equations, which are very costly and complicated in actual application [11,12]. In order to overcome these disadvantages and improve the efficiency of the Newton iteration method, many variants in terms of approximate, quasi-update, inner/outer or inexact Newton methods have been established and analyzed. See [2,4,5,13]. By making use of the HSS iteration as the inner solver for the Newton method, Bai and Guo [14] established a class of Newton-HSS methods for solving large sparse systems of nonlinear equations with positive definite Jacobian matrices at the solution points.

For the system of weakly nonlinear equations (1.1), Bai [4] established a class of sequential two-stage iteration methods by taking into account concrete properties of the involved matrix and mapping. Based on the matrix multi-splitting technique, Bai [15] presented efficient parallel generalizations of the sequential two-stage iteration methods. Furthermore, Bai and Huang [16] proposed asynchronous multi-splitting two-stage iteration methods. These asynchronous methods have the potentials of converging much faster than their synchronous counterparts in [15], especially when there is load imbalance. Moreover, based on the fact 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 [2] presented the Picard-HSS and the nonlinear HSS-like iteration methods, and Zhu and Zhang [17] presented the Picard-CSCS and the nonlinear CSCS-like iteration methods for the special case when A is a Toeplitz matrix, for weakly nonlinear systems whose linear terms are positive definite.

In this paper, to make these methods in [2] more attractive, we establish two new nonlinear composite splitting iteration schemes, namely, Picard-GPHSS and nonlinear GPHSS-like iteration methods respectively, for solving the large scale system of weakly nonlinear equations (1.1).

The remainder of the paper is organized as follows. In Section 2 we review the HSS and GPHSS iteration methods for the non-Hermitian positive definite linear systems. In Section 3, we establish the Picard-GPHSS and nonlinear GPHSS-like iteration methods and discuss their convergence properties. In Section 5, by choosing different preconditioned matrix and parameters, we derive several existing and new iterative methods. Numerical examples are given to illustrate the effectiveness of the new iteration methods in Section 6. Finally, in Section 7 we draw a brief conclusion.

2. The HSS and GPHSS iteration methods

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

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

To solve the linear system (2.1) iteratively, efficient splitting of the coefficient matrix A is required [18–20]. Since a matrix A naturally possesses a Hermitian and skew-Hermitian splitting (HSS) $A = H + S$,

$$H = \frac{1}{2}(A + A^*) \quad \text{and} \quad S = \frac{1}{2}(A - A^*). \tag{2.2}$$

Bai [21] presented the HSS iteration method and proved that for any positive α the HSS method converges unconditionally to the unique solution of the non-Hermitian positive definite system of linear equations (2.1). This method has caught people's great attention in recent years; see [21–24].

The HSS iteration method. ([21]) Given an initial guess $x^{(0)} \in \mathbb{C}^n$, compute $x^{(k+1)}$ for $k = 0, 1, 2, \ldots$ using the following iteration scheme until $\{x^{(k)}\}$ satisfies the stopping criterion:

$$\begin{cases} (\alpha I + H)x^{\left(k+\frac{1}{2}\right)} = (\alpha I - S)x^{(k)} + b, \\ (\alpha I + S)x^{(k+1)} = (\alpha I - H)x^{\left(k+\frac{1}{2}\right)} + b, \end{cases} \tag{2.3}$$

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

Based on the HSS and preconditioned HSS methods, Yang, et al. [25] presented a generalized preconditioned Hermitian and skew-Hermitian splitting (GPHSS) iteration method and studied its convergence property. This method is described as follows.

The GPHSS iteration method. ([25]) Given an initial guess $x^{(0)} \in \mathbb{C}^n$, compute $x^{(k+1)}$ for $k = 0, 1, 2, \ldots$ using the following iteration scheme until $\{x^{(k)}\}$ satisfies the stopping criterion:

$$\begin{cases} (\alpha P + H)x^{\left(k+\frac{1}{2}\right)} = (\alpha P - S)x^{(k)} + b, \\ (\beta P + S)x^{(k+1)} = (\beta P - H)x^{\left(k+\frac{1}{2}\right)} + b, \end{cases} \tag{2.4}$$

where α is a nonnegative constant, β is a positive constant, and P is a Hermitian positive definite matrix.

The GPHSS method is actually a two-parameter two-step iteration method. It becomes the PHSS method when $\alpha = \beta$. And HSS method is obviously a special case when $\alpha = \beta$ and $P = I$ [25].

3. The new iteration methods

3.1. The Picard-GPHSS iteration method

The linearization is a traditional strategy for solving systems of nonlinear equations. Based on the separability and strong dominance between the linear term Ax and the nonlinear term $\phi(x)$ of the weakly nonlinear system (1.1), following [2] we can use the Picard iteration method

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

to solve the system of weakly nonlinear equations (1.1).

At each Picard iteration step, we need to solve a linear system. Applying the HSS or GPHSS iteration method to solve these linear systems, the next iterate $x^{(k+1)}$ may be approximately computed. This naturally leads to the inexact Picard iteration methods called Picard-HSS and Picard-GPHSS iteration methods. The former is established by Bai and Yang in [2]. The Picard-GPHSS iteration method is algorithmically described as follows.

The Picard-GPHSS iteration method. Let $\phi : \mathbb{D} \subset \mathbb{C}^n \to \mathbb{C}^n$ be a continuously differentiable function and $A \in \mathbb{C}^{n\times n}$ be a positive definite matrix, with $H = \frac{1}{2}(A + A^*)$, $S = \frac{1}{2}(A - A^*)$ be the Hermitian and skew-Hermitian parts of A respectively. 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 $k = 0, 1, 2, \ldots$ using the following iteration scheme until $\{x^{(k)}\}$ satisfies the stopping criterion:

(a) $x^{(k,0)} := x^{(k)}$;

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

$$\begin{cases} (\alpha P + H)x^{\left(k,l+\frac{1}{2}\right)} = (\alpha P - S)x^{(k,l)} + \phi(x^{(k)}), \\ (\beta P + S)x^{(k,l+1)} = (\beta P - H)x^{\left(k,l+\frac{1}{2}\right)} + \phi(x^{(k)}), \end{cases} \tag{3.1}$$

where α is a nonnegative constant, β is a positive constant, and P is a Hermitian positive definite matrix.

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

Note that the Picard-GPHSS iteration method becomes the Picard-PHSS method when $\alpha = \beta$. And the Picard-HSS method is obviously a special case when $\alpha = \beta$ and $P = I$.

Now, we consider the local convergence of the Picard-GPHSS iteration method.

Based on the iteration scheme of Picard-GPHSS, the $(k + 1)$-th Picard-GPHSS iterate $x^{(k+1)}$ can be expressed:

$$x^{(k+1)} = T(\alpha, \beta)^{l_k}x^{(k)} + \sum_{j=0}^{l_k-1} T(\alpha, \beta)^j G(\alpha, \beta)\phi(x^{(k)}), \quad k = 0, 1, 2, \ldots, \tag{3.2}$$

where $T(\alpha, \beta) = (\beta P + S)^{-1}(\beta P - H)(\alpha P + H)^{-1}(\alpha P - S)$ and $G(\alpha, \beta) = (\alpha + \beta)(\beta P + S)^{-1}(\alpha P + H)^{-1}$.

Suppose $x^* \in \mathbb{D}$ is a solution of the system of weakly nonlinear equations (1.1), then

$$x^* = T(\alpha, \beta)^{l_k}x^* + \sum_{j=0}^{l_k-1} T(\alpha, \beta)^j G(\alpha, \beta)\phi(x^*). \tag{3.3}$$

Thus

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

By utilizing the R-convergence and similar arguments of Theorem 3.1 in [2], we obtain the following local convergence theory for the Picard-GPHSS iteration method.

Theorem 3.1. *Let $\phi : \mathbb{D} \subset \mathbb{C}^n \to \mathbb{C}^n$ 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 $H = \frac{1}{2}(A + A^*)$, $S = \frac{1}{2}(A - A^*)$ are the Hermitian and the skew-Hermitian parts of the matrix A respectively. Denote by*

$$\mu(\alpha, \beta) = \|T(\alpha, \beta)\| = \|(\beta P + S)^{-1}(\beta P - H)(\alpha P + H)^{-1}(\alpha P - S)\|, \qquad \omega = \|A^{-1}\phi'(x^*)\|.$$

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, $k = 0, 1, 2, \ldots$, the iteration sequence $\{x^{(k)}\}_{k=0}^{\infty}$ generated by the Picard-GPHSS iteration method is well-defined and convergent to x^*, provided that $l_0 \geq \lfloor \ln(\frac{1-\omega}{1+\omega})/\ln(\mu(\alpha, \beta)) \rfloor$ (where $\lfloor\cdot\rfloor$ is used to denote the smallest integer no less than the corresponding real number). Moreover, it holds that

$$\limsup_{k\to\infty} \|x^{(k)} - x^*\|^{\frac{1}{k}} \leq \omega + (1 + \omega)\mu(\alpha, \beta)^{l_0}, \quad l_0 = \liminf_{k\to\infty} l_k;$$

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

$$\limsup_{k\to\infty} \|x^{(k)} - x^*\|^{\frac{1}{k}} \leq \omega.$$

Proof. The proof uses arguments similar to those in proof of the convergence Theorem of the Picard-HSS iteration method in [2]. □

Theorem 3.1 shows that the convergence rate of the Picard-GPHSS iteration is essentially determined by the quantities ω and $\mu(\alpha, \beta)$. Usually, small ω and $\mu(\alpha, \beta)$ will lead to fast convergence of the Picard-GPHSS iteration. For a system of weakly nonlinear equations having an ill-conditioned matrix $A \in \mathbb{C}^{n\times n}$, $\mu(\alpha, \beta)$ is close to 1 and the Picard-GPHSS iteration will converge very slowly even for reasonably large numbers of inner iteration steps, which is the same as the Picard-HSS. But the convergence rate of the Picard-GPHSS iteration could be accelerated through choosing iteration parameters and preconditioned matrix.

3.2. The nonlinear GPHSS-like iteration method

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

The GPHSS-like iteration method. Let $\phi : \mathbb{D} \subset \mathbb{C}^n \to \mathbb{C}^n$ be a continuously differentiable function and $A \in \mathbb{C}^{n\times n}$ be a positive definite matrix, with $A = H + S$, H and S be the Hermitian and skew-Hermitian parts of A, respectively. Given an initial guess $x^{(0)} \in \mathbb{D}$, compute $x^{(k+1)}$ for $k = 0, 1, 2, \ldots$ using the following iteration scheme until $\{x^{(k)}\}$ satisfies the stopping criterion:

$$\begin{cases} (\alpha P + H)x^{\left(k+\frac{1}{2}\right)} = (\alpha P - S)x^{(k)} + \phi(x^{(k)}), \\ (\beta P + S)x^{(k+1)} = (\beta P - H)x^{\left(k+\frac{1}{2}\right)} + \phi\left(x^{\left(k+\frac{1}{2}\right)}\right), \end{cases} \tag{3.4}$$

where α is a nonnegative constant, β is a positive constant and P is a Hermitian positive definite matrix.

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

$$\begin{cases} U(x) = (\alpha P + H)^{-1}((\alpha P - S)x + \phi(x)), \\ V(x) = (\beta P + S)^{-1}((\beta P - H)x + \phi(x)) \end{cases} \tag{3.5}$$

and

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

Then the nonlinear GPHSS-like iteration scheme can be equivalently expressed as

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

Suppose $x^* \in \mathbb{D}$ is a solution of the system of weakly nonlinear equations (1.1), we can easily verify the following fact by using the derivative chain rule

$$\begin{aligned} \psi'(x^*) &= V'(x^*)U'(x^*) \\ &= (\beta P + S)^{-1}(\beta P - H + \phi'(x^*))\,(\alpha P + H)^{-1}(\alpha P - S + \phi'(x^*)). \end{aligned}$$

By making use of the Ostrowski Theorem, i.e., Theorem 10.1.3 in [3], we know that if $\rho(\psi'(x^*)) < 1$, then x^* is a point of attraction of the nonlinear GPHSS-like iteration. Then we can obtain the following local convergence theory for the GPHSS-like iteration method.

Theorem 3.2. *Assume that* $\phi : \mathbb{D} \subset \mathbb{C}^n \to \mathbb{C}^n$ *is F-differentiable at a point* $x^* \in \mathbb{D}$ *such that* $Ax^* = \phi(x^*)$. *Suppose* $A = H + S$, *where H and S are the Hermitian and the skew-Hermitian parts of the matrix A, respectively. Denote by*

$$T(\alpha, \beta; x^*) = (\beta P + S)^{-1}(\beta P - H + \phi'(x^*))\,(\alpha P + H)^{-1}(\alpha P - S + \phi'(x^*)).$$

If spectral radius $\rho(T(\alpha, \beta; x^*)) < 1$, *then* $x^* \in \mathbb{D}$ *is a point of attraction of the nonlinear GPHSS-like iteration.*

Now we give a sufficient condition for guaranteeing the validity of $\rho(T(\alpha, \beta; x^*)) < 1$.

Corollary 3.3. *Assume that the conditions of Theorem 3.2 are satisfied. Denote by*

$$\delta = \max\{\|\phi'(x^*)(\beta P + S)^{-1}\|_2,\ \|\phi'(x^*)(\alpha P + H^{-1})\|_2\},$$

$$a = \max_{\sigma_i \in \sigma(P^{-1}S)} \frac{\sqrt{\alpha^2 + \sigma_i^2}}{\sqrt{\beta^2 + \sigma_i^2}}, \qquad b = \max_{\lambda_i \in \lambda(P^{-1}H)} \left|\frac{\beta - \lambda_i}{\alpha + \lambda_i}\right|,$$

where $\lambda(P^{-1}H)$ is the spectral set of $P^{-1}H$ and $\sigma(P^{-1}S)$ is the singular-value set of $P^{-1}S$. Then $\rho(T(\alpha, \beta; x^*)) < 1$ holds, provided that

$$\delta < \frac{2(1 - ab)}{(a + b) + \sqrt{(a - b)^2 + 4}}. \tag{3.6}$$

Proof. By straightforward computations we have

$$\begin{aligned}(\beta P + S)T(\alpha, \beta; x^*)(\beta P + S)^{-1} = {} & (\beta P + S)T(\alpha, \beta)(\beta P + S)^{-1} + (\beta P - H)(\alpha P + H)^{-1}\phi'(x^*)(\beta P + S)^{-1} \\ & + \phi'(x^*)(\alpha P + H)^{-1}(\alpha P - S)(\beta P + S)^{-1} \\ & + \phi'(x^*)(\alpha P + H^{-1})\phi'(x^*)(\beta P + S)^{-1},\end{aligned}$$

where

$$T(\alpha, \beta) = (\beta P + S)^{-1}(\beta P - H)(\alpha P + H)^{-1}(\alpha P - S),$$

and

$$\begin{aligned}\|T(\alpha, \beta)\| &\le \|(\alpha P - S)(\beta P + S)^{-1}\|\,\|(\beta P - H)(\alpha P + H)^{-1}\| \\ &\le \max_{\sigma_i \in \sigma(P^{-1}S)} \frac{\sqrt{\alpha^2 + \sigma_i^2}}{\sqrt{\beta^2 + \sigma_i^2}} \max_{\lambda_i \in \lambda(P^{-1}H)} \left|\frac{\beta - \lambda_i}{\alpha + \lambda_i}\right| = ab.\end{aligned}$$

Hence,

$$\begin{aligned}\|T(\alpha, \beta; x^*)\| &= \|(\beta P + S)T(\alpha, \beta; x^*)(\beta P + S)^{-1}\| \\ &\le \|(\beta P + S)T(\alpha, \beta)(\beta P + S)^{-1}\| + \|(\beta P - H)(\alpha P + H)^{-1}\phi'(x^*)(\beta P + S)^{-1}\| \\ &\quad + \|\phi'(x^*)(\alpha P + H)^{-1}(\alpha P - S)(\beta P + S)^{-1}\| + \|\phi'(x^*)(\alpha P + H)^{-1}\phi'(x^*)(\beta P + S)^{-1}\| \\ &\le \|T(\alpha, \beta)\| + \|(\beta P - H)(\alpha P + H)^{-1}\|\,\|\phi'(x^*)(\beta P + S)^{-1}\| \\ &\quad + \|\phi'(x^*)(\alpha P + H)^{-1}\|\,\|(\alpha P - S)(\beta P + S)^{-1}\| + \|\phi'(x^*)(\alpha P + H)^{-1}\|\,\|\phi'(x^*)(\beta P + S)^{-1}\| \\ &\le ab + (a + b)\delta + \delta^2.\end{aligned}$$

Now, under the condition (3.6), we have $\rho(T(\alpha, \beta; x^*)) \le \|T(\alpha, \beta; x^*)\| < 1$. □

Corollary 3.3 shows that $\rho(T(\alpha, \beta; x^*)) < 1$ is valid if $\phi'(x^*)$ is reasonably small compared with the matrix A. Otherwise, the nonlinear GPHSS-like iteration may be convergent slowly or even divergent. Thus, the optimal parameters and suitable preconditioner are required.

4. Optimal parameters

Having studied the convergence properties of the Picard-GPHSS and nonlinear GPHSS-like iteration in the previous section, we find that the convergent speed of iteration methods depends essentially on two factors: (1) weakly nonlinearity of the weakly nonlinear systems and (2) finding the optimal parameters to guarantee the spectral radius of the iteration matrix is less than 1. As the former is problem-dependent, we consider the latter in the following part.

We note that matrices $\alpha P + H$ and $\beta P + S$ are nonsingular for any nonnegative constant α and positive constant β, one can obtain the iteration matrix of the GPHSS method as follows:

$$T(\alpha, \beta) = (\beta P + S)^{-1}(\beta P - H)(\alpha P + H)^{-1}(\alpha P - S).$$

The spectral radius of the iteration matrix $T(\alpha, \beta)$ satisfies clearly,

$$\begin{aligned}\rho(T(\alpha, \beta)) &= \rho((\beta P + S)^{-1}(\beta P - H)(\alpha P + H)^{-1}(\alpha P - S)) \\ &\le \|T(\alpha, \beta)\| \le \|(\alpha P - S)(\beta P + S)^{-1}\|\,\|(\beta P - H)(\alpha P + H)^{-1}\| \\ &\le \max_{\lambda_i \in \lambda(P^{-1}H)} \left|\frac{\beta - \lambda_i}{\alpha + \lambda_i}\right| \max_{\sigma_i \in \sigma(P^{-1}S)} \frac{\sqrt{\alpha^2 + \sigma_i^2}}{\sqrt{\beta^2 + \sigma_i^2}} := v(\alpha, \beta),\end{aligned}$$

and

$$(\bar{\alpha}, \bar{\beta}) = \arg\min_{\alpha,\beta}\{v(\alpha, \beta)\}.$$

Apparently, $v(\alpha, \beta)$ defines an upper bound of the contraction factor of the GPHSS iteration whose convergent speed depends essentially on the choice of the two-parameters α and β. Consequently, we need the properties of the function $v(\alpha.\beta)$ with respect to the two-parameters.

Theorem 4.1 ([25]). *Denote*

$$\lambda_{\max} = \max_{\lambda_k \in P^{-1}H}\{\lambda_k\}, \qquad \lambda_{\min} = \min_{\lambda_k \in P^{-1}H}\{\lambda_k\}, \qquad e_{\max} = \max_{ie_j \in P^{-1}S}\{|e_j|\}, \qquad e_{\min} = \min_{ie_j \in P^{-1}S}\{|e_j|\},$$

with $i = \sqrt{-1}$. *The function* $v(\alpha, \beta)$ *has its minimum on a curve*

$$\beta = \beta^*(\alpha) = \frac{\alpha(\lambda_{\max} + \lambda_{\min}) + 2\lambda_{\max}\lambda_{\min}}{2\alpha + \lambda_{\max} + \lambda_{\min}} \in [\lambda_{\min}, \lambda_{\max}].$$

And the optimal parameters are given by

$$(\bar{\alpha}, \bar{\beta}) = \begin{cases} (\alpha_1, \beta^*(\alpha_1)), & \lambda_{\max}\lambda_{\min} \le e_{\min}^2, \\ (\alpha_0, \beta^*(\alpha_0)), & e_{\min}^2 < \lambda_{\max}\lambda_{\min} \le e_{\max}^2, \\ (\alpha_2, \beta^*(\alpha_2)), & \lambda_{\max}\lambda_{\min} \ge e_{\max}^2, \end{cases} \tag{4.1}$$

with

$$\alpha_0 = \sqrt{\lambda_{\max}\lambda_{\min}},$$

$$\alpha_1 = \frac{-(\lambda_{\max}\lambda_{\min} - e_{\min}^2) + \sqrt{(e_{\min}^2 + \lambda_{\max}^2)(e_{\min}^2 + \lambda_{\min}^2)}}{\lambda_{\max} + \lambda_{\min}},$$

$$\alpha_2 = \frac{-(\lambda_{\max}\lambda_{\min} - e_{\max}^2) + \sqrt{(e_{\max}^2 + \lambda_{\max}^2)(e_{\max}^2 + \lambda_{\min}^2)}}{\lambda_{\max} + \lambda_{\min}}.$$

The minimum value at the optimal parameters is

$$\sigma(\bar{\alpha}, \bar{\beta}) = \begin{cases} \sigma(\alpha_1), & \lambda_{\max}\lambda_{\min} \le e_{\min}^2, \\ \sigma(\alpha_0), & e_{\min}^2 < \lambda_{\max}\lambda_{\min} \le e_{\max}^2, \\ \sigma(\alpha_2), & \lambda_{\max}\lambda_{\min} \ge e_{\max}^2, \end{cases}$$

where

$$\sigma(\alpha) := \sigma(\alpha, \beta^*(\alpha)) = \begin{cases} \dfrac{\beta^*(\alpha) - \lambda_{\min}}{\alpha + \lambda_{\min}}\sqrt{\dfrac{\alpha^2 + e_{\min}^2}{\beta^*(\alpha)^2 + e_{\min}^2}}, & \alpha > \alpha_0, \\ \dfrac{\beta^*(\alpha) - \lambda_{\min}}{\alpha + \lambda_{\min}}\sqrt{\dfrac{\alpha^2 + e_{\max}^2}{\beta^*(\alpha)^2 + e_{\max}^2}}, & \alpha \le \alpha_0. \end{cases}$$

In Theorem 4.1, the optimal parameters $\bar{\alpha}$ and $\bar{\beta}$ minimize only the simplified upper bound $v(\alpha, \beta)$ of the spectral radius. However, one usually cannot expect to minimize the spectral radii of iteration matrices with these optimal parameters [25].

5. Several algorithms

By choosing preconditioned matrix P and the iteration parameters α, β, we can easily get the following special cases of the Picard-GPHSS and the nonlinear GPHSS-like methods to solve the system of weakly nonlinear equations (1.1).

Case I: If $P = I$ and $\alpha = \beta$, then the Picard-GPHSS and the nonlinear GPHSS-like methods reduce to the Picard-HSS and the nonlinear HSS-like methods respectively. They are presented by Bai and Yang in [2] and can be algorithmically described as follows.

Algorithm 5.1 ([2]). The Picard-HSS iteration method

$$\begin{cases} (\alpha I + H)x^{\left(k,l+\frac{1}{2}\right)} = (\alpha I - S)x^{(k,l)} + \phi(x^{(k)}), \\ (\alpha I + S)x^{(k,l+1)} = (\alpha I - H)x^{\left(k,l+\frac{1}{2}\right)} + \phi(x^{(k)}). \end{cases} \tag{5.1}$$

Algorithm 5.2 ([2]). The nonlinear HSS-like iteration method

$$\begin{cases} (\alpha I + H)x^{\left(k+\frac{1}{2}\right)} = (\alpha I - S)x^{(k)} + \phi(x^{(k)}), \\ (\alpha I + S)x^{(k+1)} = (\alpha I - H)x^{\left(k+\frac{1}{2}\right)} + \phi\left(x^{\left(k+\frac{1}{2}\right)}\right). \end{cases} \tag{5.2}$$

Case II: If $P = I$ and $\alpha \neq \beta$, then the Picard-GPHSS and the nonlinear GPHSS-like methods reduced to Picard-AHSS and nonlinear AHSS-like methods presented in [26], respectively.

Algorithm 5.3 ([26]). The Picard-AHSS iteration method

$$\begin{cases} (\alpha I + H)x^{\left(k,l+\frac{1}{2}\right)} = (\alpha I - S)x^{(k,l)} + \phi(x^{(k)}), \\ (\beta I + S)x^{(k,l+1)} = (\beta I - H)x^{\left(k,l+\frac{1}{2}\right)} + \phi(x^{(k)}). \end{cases} \tag{5.3}$$

Algorithm 5.4 ([26]). The nonlinear AHSS-like iteration method

$$\begin{cases} (\alpha I + H)x^{\left(k+\frac{1}{2}\right)} = (\alpha I - S)x^{(k)} + \phi(x^{(k)}), \\ (\beta I + S)x^{(k+1)} = (\beta I - H)x^{\left(k+\frac{1}{2}\right)} + \phi\left(x^{\left(k+\frac{1}{2}\right)}\right). \end{cases} \tag{5.4}$$

In this case, if the iteration parameter $\alpha = 0$, then the Picard-AHSS and nonlinear AHSS-like iteration methods become single-parameter iteration methods given in [27], which are called Picard-LHSS and nonlinear LHSS-like methods respectively.

Case III: If $P \neq I$ and $\alpha = \beta$, then the Picard-GPHSS and the nonlinear GPHSS-like methods give the following new iteration methods, which are called Picard-PHSS and nonlinear PHSS-like methods respectively.

Algorithm 5.5. The Picard-PHSS iteration method

$$\begin{cases} (\alpha P + H)x^{\left(k,l+\frac{1}{2}\right)} = (\alpha P - S)x^{(k,l)} + \phi(x^{(k)}), \\ (\alpha P + S)x^{(k,l+1)} = (\alpha P - H)x^{\left(k,l+\frac{1}{2}\right)} + \phi(x^{(k)}). \end{cases} \tag{5.5}$$

Algorithm 5.6. The nonlinear PHSS-like iteration method

$$\begin{cases} (\alpha P + H)x^{\left(k+\frac{1}{2}\right)} = (\alpha P - S)x^{(k)} + \phi(x^{(k)}), \\ (\alpha P + S)x^{(k+1)} = (\alpha P - H)x^{\left(k+\frac{1}{2}\right)} + \phi\left(x^{\left(k+\frac{1}{2}\right)}\right). \end{cases} \tag{5.6}$$

Furthermore, it is noteworthy that through the selection of various preconditioned matrices, not only can different iteration methods be made available, but also the convergence rate of iteration methods can be accelerated. A natural choice of the preconditioned matrix P is $P = H$, which is developed by Bertaccini etc. in [28].

6. Numerical results

In this section, we will give several numerical experiments to illustrate the effectiveness of our methods and to make a further demonstration of the advantage of the Picard-GPHSS and the nonlinear GPHSS-like methods over the Picard-HSS and the nonlinear HSS-like iteration methods.

The problem under consideration is the classical three-dimensional convection–diffusion equation

$$\begin{cases} -(u_{xx} + u_{yy} + u_{zz}) + q(u_x + u_y + u_z) = \sin(u + 1), \quad (x, y, z) \in \Omega, \\ u(x, y, z) = 0, \quad (x, y, z) \in \partial\Omega, \end{cases} \tag{6.1}$$

on the unit cube $\Omega = [0, 1] \times [0, 1] \times [0, 1]$, and q is a positive constant coefficient used to measure the magnitude of the convective terms. Applying the seven-point finite difference and assuming that the numbers (N) of grid points in all three directions are the same, we obtain the system of weakly nonlinear equations of the form (1.1), where $n = N^3$, and for details to [21,29,30]. Different q and N result in different matrix A.

Table 1
The optimal parameters of iteration for the convection–diffusion equation ($N = 4$).

q	HSS	AHSS		GPHSS	
	α	α	β	α	β
0	3.5267	0	9.7526	0	1
1	3.5267	0	9.7526	0	1
10	3.5267	0.0039	9.7527	0.0039	1
50	3.5267	0.0966	9.7536	0.0983	1
100	3.5267	0.3864	9.7565	0.3933	1
500	3.5267	9.6146	9.8027	2.6196	1

Table 2
The optimal parameters of iteration for the convection–diffusion equation ($N = 8$).

q	HSS	AHSS		GPHSS	
	α	α	β	α	β
0	0.7019	0	11.2802	0	1
1	0.7019	0	11.2802	0	1
10	0.7019	0.0001	11.2802	0.0005	1
50	0.7019	0.0033	11.2802	0.012	1
100	0.7019	0.0133	11.2802	0.0478	1
500	0.7019	0.3314	11.2802	1	1

In our numerical experiments, we use the zeros vector as the initial guess and stop the iteration as soon as

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

Meanwhile, the stopping criteria for the inner iterations of the Picard-HSS and the Picard-GPHSS 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. If η_k is fixed for all k, then it is simply denoted by η. The two sub-systems of linear equations are solved in the way if $Ax = b$, then $x = A^{-1}b$.

The Picard-GPHSS and the nonlinear GPHSS-like methods are compared with the Picard-HSS and the nonlinear HSS-like methods for different problem sizes N ($N = 4, 8$) and tolerance $\eta = 0.1, 0.01, 0.001$, from aspects of numbers of outer, inner and total iteration steps (denoted as $\mathrm{IT}_{\mathrm{out}}$, $\mathrm{IT}_{\mathrm{int}}$ and IT, respectively) and total CPU time (denoted as CPU). To this end, we need to choose an suitable preconditioner. The considered preconditioner used in our methods is $P = H$ developed in [28].

In our implementations, we adopt the optimal parameters $\alpha^* = \sqrt{\lambda_{\min}\lambda_{\max}}$ given in [21] for the Picard-HSS and the nonlinear HSS-like methods and adopt the optimal parameters α^*, β^* given in Theorem 4.1 for the Picard-GPHSS and the nonlinear GPHSS-like iteration methods. See Tables 1–2. Admittedly, the optimal parameters α^* and β^* are crucial for guaranteeing fast convergence speeds of these parameter-dependent iteration methods, but they are often problem-dependent and are generally difficult to be determined [2].

In Tables 3–4, we list the numerical results corresponding to six choices of the q, i.e., $q = 0, 1, 10, 50, 100, 500$. From these tables, we see that all experimented methods (i.e., the Picard-HSS, the nonlinear HSS-like, the Picard-AHSS, the nonlinear AHSS-like, the Picard-GPHSS and the nonlinear GPHSS-like methods) can successfully produce approximate solutions to the system of weakly nonlinear equations for all of the matrix dimensions (i.e., $n = 4^3$ and $n = 8^3$).

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 amount iteration steps and the amount of CPU time of all iteration methods increase, but the numbers of outer iteration steps are fixed or slightly increase. In general, the Picard-GPHSS iteration method is doing better than the Picard-HSS iteration method.

With the increase in parameter q, the number of outer iteration steps are fixed and decrease slightly; the number of inner iteration steps and the total iteration steps of the Picard-HSS and the Picard-GPHSS iterations decrease, but that of the Picard-AHSS iterations increase. The total CPU time increases. The Picard-GPHSS and the nonlinear GPHSS-like methods are more robust than the other iteration methods.

When the matrix dimension N increases, the numbers of outer iteration steps are fixed or increase slightly, but the numbers of inner iteration steps increase quickly, so the total iteration steps also increase. And the amount of total CPU time of all iteration methods increases quickly.

Table 3
The numbers of iteration and the total times for the convection–diffusion equation ($N = 4$).

η	Iteration		$q = 0$	$q = 1$	$q = 10$	$q = 50$	$q = 100$	$q = 500$
10^{-3}	Picard-HSS	IT_{out}	4	4	4	4	4	3
		IT_{int}	11	11	11	10	9	8
		IT	44	44	44	40	36	24
		CPU	0.0147	0.0176	0.0057	0.0066	0.0071	0.0147
	Picard-AHSS	IT_{out}	4	4	4	4	4	3
		IT_{int}	1	2	3	5	7	8
		IT	4	8	12	20	28	24
		CPU	0.0013	0.0044	0.0041	0.0047	0.0052	0.0058
	Picard-GPHSS	IT_{out}	4	4	4	4	4	3
		IT_{int}	1	1	1	1	1	1
		IT	4	4	4	4	4	3
		CPU	0.0028	0.0039	0.0036	0.0036	0.0034	0.0045
10^{-2}	Picard-HSS	IT_{out}	4	4	4	4	4	3
		IT_{int}	7	7	7	7	6	6
		IT	28	28	28	28	24	18
		CPU	0.0207	0.0047	0.0055	0.0044	0.0084	0.0212
	Picard-AHSS	IT_{out}	4	4	4	4	4	3
		IT_{int}	1	1	2	3	5	6.3333
		IT	4	4	8	12	20	19
		CPU	0.0017	0.0038	0.0039	0.0042	0.0064	0.0057
	Picard-GPHSS	IT_{out}	4	4	4	4	4	3
		IT_{int}	1	1	1	1	1	1
		IT	4	4	4	4	4	3
		CPU	0.0022	0.0034	0.0036	0.0036	0.0039	0.0044
10^{-1}	Picard-HSS	IT_{out}	6	6	6	6	6	5
		IT_{int}	4	4	4	4	3	3
		IT	24	24	24	24	18	15
		CPU	0.0059	0.0088	0.0051	0.0046	0.0134	0.0168
	Picard-AHSS	IT_{out}	4	4	5	4	4	6
		IT_{int}	1	1	1	2	3	3
		IT	4	4	5	8	12	18
		CPU	0.0019	0.0037	0.0037	0.0042	0.0054	0.0055
	Picard-GPHSS	IT_{out}	4	4	4	4	4	3
		IT_{int}	1	1	1	1	1	1
		IT	4	4	4	4	4	3
		CPU	0.0025	0.0034	0.0036	0.0036	0.0047	0.0052
	Like-HSS	IT	21	21	21	20	17	14
		CPU	0.0029	0.0051	0.0055	0.0055	0.0066	0.0049
	Like-AHSS	IT	4	4	5	9	13	16
		CPU	0.0015	0.0035	0.0036	0.0042	0.0074	0.0051
	Like-GPHSS	IT	2	2	2	3	3	3
		CPU	0.0032	0.0031	0.0033	0.0033	0.0053	0.0036

On the whole, in terms of iteration step, the nonlinear GPHSS-like method and the Picard-GPHSS perform better than the nonlinear HSS-like and the Picard-HSS iteration methods. In terms of CPU time, the situation is almost the same. Therefore, the nonlinear GPHSS-like method and the Picard-GPHSS method are the winners for solving this test problem.

To this end, the nonlinear two-point boundary-value problem with a convective dominated term in one-dimensional setting was to be considered to further prove the outstanding performances of the Picard-GPHSS and the nonlinear GPHSS-like iteration methods:

$$\begin{cases} -\varepsilon \dfrac{d^2u}{dx^2} + b(x)\dfrac{du}{dx} = f(u, x), \quad x \in (0, 1), \\ u(0) = u(1) = 0, \end{cases} \tag{6.2}$$

where $\varepsilon = 1$, $b(x) = 1000$ and $f(u, x) = \sin(u)$. This problem (6.2) is singularly perturbed [31,32]. Here, the large system (1.1) is generated by applying the central difference formula to approximate the second order derivation $\frac{d^2u}{dx^2}$, and the backward difference formula to approximate the first order derivation $\frac{du}{dx}$, with the step size $\Delta x = 1/(m+1)$, $x_j = j\Delta x$ [17].

The Picard-GPHSS and the nonlinear GPHSS-like methods are compared with the following seven methods from aspects of numbers of outer, inner and total iteration steps and total CPU time, namely, the Picard-HSS, Picard-AHSS, Picard-CSCS, nonlinear HSS-like, nonlinear AHSS-like, nonlinear CSCS-like and Newton-GMRES iteration methods. The preconditioner used in our methods is also $P = H$. The optimal parameters of iteration methods were listed in Table 5. And the numerical results corresponding to $\eta = 0.1$ were listed in Table 6. A "1000+" means that convergence was not attained after 1000 iterations.

Table 4
The numbers of iteration and the total times for the convection–diffusion equation ($N = 8$).

η	Iteration		$q = 0$	$q = 1$	$q = 10$	$q = 50$	$q = 100$	$q = 500$
10^{-3}	Picard-HSS	IT_{out}	4	4	4	4	4	4
		IT_{int}	19.75	19.75	19.75	19.75	19	13
		IT	79	79	79	79	76	52
		CPU	0.6761	0.7002	0.7634	0.6221	0.6139	0.5518
	Picard-AHSS	IT_{out}	4	4	4	4	4	4
		IT_{int}	1	2	2	3	4	13.5
		IT	4	8	8	12	16	54
		CPU	0.0926	0.3119	0.3049	0.3067	0.3192	0.4144
	Picard-GPHSS	IT_{out}	4	4	4	4	4	4
		IT_{int}	1	1	1	1	1	1
		IT	4	4	4	4	4	4
		CPU	0.1894	0.2982	0.2891	0.2844	0.2869	0.2858
10^{-2}	Picard-HSS	IT_{out}	4	4	4	4	4	4
		IT_{int}	13	13	13	13	13	8.25
		IT	52	52	52	52	52	33
		CPU	0.5917	0.6844	0.5447	0.5464	0.5467	0.4935
	Picard-AHSS	IT_{out}	4	4	4	4	4	4
		IT_{int}	1	1	2	2	3	8.5
		IT	4	4	8	8	12	34
		CPU	0.0887	0.3757	0.2892	0.2984	0.311	0.3645
	Picard-GPHSS	IT_{out}	4	4	4	4	4	4
		IT_{int}	1	1	1	1	1	1
		IT	4	4	4	4	4	4
		CPU	0.1784	0.4855	0.2859	0.2869	0.2862	0.2856
10^{-1}	Picard-HSS	IT_{out}	7	7	7	6	6	6
		IT_{int}	6.8571	6.8571	6.8571	6.8333	6.8333	4.3333
		IT	48	48	48	41	41	26
		CPU	0.6144	0.561	0.5346	0.5135	0.5148	0.4738
	Picard-AHSS	IT_{out}	4	4	4	6	4	6
		IT_{int}	1	1	1	1	2	4.8333
		IT	4	4	4	6	8	29
		CPU	0.0913	0.2598	0.2807	0.2928	0.2988	0.3572
	Picard-GPHSS	IT_{out}	4	4	4	4	4	4
		IT_{int}	1	1	1	1	1	1
		IT	4	4	4	4	4	4
		CPU	0.2422	0.2885	0.287	0.2841	0.2889	0.3024
	Like-HSS	IT	40	40	40	39	38	24
		CPU	0.1786	0.3712	0.3807	0.3804	0.3811	0.3454
	Like-AHSS	IT	4	4	4	6	8	27
		CPU	0.1598	0.2914	0.2823	0.2919	0.3041	0.3516
	Like-GPHSS	IT	2	2	2	2	3	3
		CPU	0.2167	0.3104	0.282	0.2803	0.2869	0.2838

Table 5
The optimal parameters of iteration for the nonlinear two-point boundary-value problem (6.2).

m	HSS	AHSS		GPHSS($P = H$)		CSCS
	α	α	β	α	β	α
20	0.2981	0.0444	0	0.0032	3.5754	0.0540
40	0.1531	0.0117	0	0.0001	3.8906	0.0073
80	0.0776	0.0030	0	0	3.9721	0.0009
160	0.0390	0.0008	0	0	3.9930	0.0001
320	0.0196	0.0002	0	0	3.9982	0

From Table 6, we see that the Picard-GPHSS and nonlinear GPHSS-like iteration methods do better and are superior to the other methods. With the increase in matrix dimension N, the numbers of outer iteration steps of all iteration methods except for the Newton-GMRES are fixed and increase slightly; the numbers of inner iteration steps and the total iteration steps of the iteration methods such as the Picard-AHSS, Picard-CSCS and Picard-GPHSS decrease, but that of the Picard-HSS and Newton-GMRES methods increase. The total CPU time increases. And the Picard-GPHSS and nonlinear GPHSS-like iteration methods are more robust to other iteration methods.

Since the special Toeplitz structures of the coefficient matrix A is involved, the nonlinear CSCS-like iteration method is more efficient when $m = 160$ and 320. However, the worst shortcoming of nonlinear CSCS-like iteration method is can be applied only to the Toeplitz systems of weakly nonlinear equations [17].

Table 6
The numbers of iteration and the total times for the nonlinear two-point boundary-value problem ($\eta = 0.1$).

Iteration		$m = 20$	$m = 40$	$m = 80$	$m = 160$	$m = 320$
Picard-HSS	IT_{out}	8	9	9	9	9
	IT_{int}	14.125	29.8889	59.6667	117.4	234.8
	IT	113	269	537	1057	2113
	CPU	0.0031	0.0117	0.0347	0.4750	3.9598
Picard-AHSS	IT_{out}	7	7	7	7	7
	IT_{int}	21.7143	10.1429	5.1429	4	3
	IT	152	71	36	28	21
	CPU	0.0028	0.0024	0.0049	0.0247	0.1314
Picard-GPHSS	IT_{out}	6	6	6	6	6
	IT_{int}	3	2	1	1	1
	IT	**18**	**12**	**6**	**6**	**6**
	CPU	**0.0010**	**0.0012**	**0.0033**	**0.0193**	**0.1072**
Picard-CSCS	IT_{out}	1	1	1	1	1
	IT_{int}	484	279	202	162	136
	IT	484	279	202	162	136
	CPU	0.0075	0.0066	0.0145	0.0753	0.3031
Newton-GMRES	IT_{out}	9	25	66	257	1000+
	IT_{int}	4	8	10	10	10
	IT	36	200	660	2570	10 000+
	CPU	0.0419	0.1681	0.6044	3.1128	39.5089
Like-HSS	IT	91	198	394	783	1000+
	CPU	0.0042	0.0141	0.041	0.4143	2.1578
Like-AHSS	IT	131	51	22	12	9
	CPU	0.0057	0.0034	0.0046	0.0188	0.1041
Like-GPHSS	IT	**13**	**7**	**6**	6	6
	CPU	**0.0010**	**0.0023**	**0.0034**	0.0176	0.1228
Like-CSCS	CPU	22	11	8	**6**	**5**
	CPU	0.0029	0.0021	0.0033	**0.0077**	**0.0415**

7. Concluding remarks

For large scale systems of weakly nonlinear equations, we have established the Picard-GPHSS and the nonlinear GPHSS-like iteration methods and reduced some existing and new iteration methods which are special case of our methods. Numerical examples show that the new iteration methods with suitable preconditioner are feasible and efficient. Moreover, these new methods often do better than the Newton-GMRES, Picard-HSS, Picard-CSCS, nonlinear HSS-like and the nonlinear HSS-like methods in our implementations.

Acknowledgments

The authors would like to thank the referees for their useful suggestions which greatly improved the original manuscript of this paper.

References

[1] 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.
[2] Z.-Z. Bai, X. Yang, On HSS-based iteration methods for weakly nonlinear systems, Appl. Numer. Math. 59 (2009) 2923–2936.
[3] J.M. Ortega, W.C. Rheinboldt, Iterative Solution of Nonlinear Equations in Several Variables, Academic Press, New York, 1970.
[4] Z.-Z. Bai, A class of two-stage iterative methods for systems of weakly nonlinear equations, Numer. Algorithms 14 (1997) 295–319.
[5] Z.-Z. Bai, On the convergence of parallel chaotic nonlinear multisplitting Newton-type methods, J. Comput. Appl. Math. 80 (1997) 317–334.
[6] Z.-Z. Bai, Y.-M. Huang, M.K. Ng, On preconditioned iterative methods for Burgers equations, SIAM J. Sci. Comput. 29 (2007) 415–439.
[7] 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.
[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] 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.
[12] E. Galligani, The Newton-arithmetic mean method for the solution of systems of nonlinear equations, Appl. Math. Comput. 134 (2003) 9–34.
[13] H.-B. An, Z.-Z. Bai, NGLM: a globally convergent Newton-GMRES method, Math. Numer. Sin. 27 (2005) 151–174.
[14] 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.
[15] Z.-Z. Bai, Parallel multisplitting two-stage iterative methods for large sparse systems of weakly nonlinear equations, Numer. Algorithms 15 (1997) 347–372.
[16] 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.

[17] M.-Z. Zhu, G.-F. Zhang, On CSCS-based iteration methods for Toeplitz system of weakly nonlinear equations, J. Comput. Appl. Math. 235 (2011) 5095–5104.
[18] M.K. Ng, Circulant and skew-circulant splitting methods for Toeplitz systems, J. Comput. Appl. Math. 159 (2003) 101–108.
[19] M.K. Ng, Iterative Methods for Toeplitz Systems, Oxford University Press, Oxford, 2004.
[20] D.M. Young, Iterative Solution of Large Linear Systems, Dover Publications, New York, 2003.
[21] 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.
[22] Z.-Z. Bai, G.H. Golub, M.K. Ng, On inexact Hermitian and skew-Hermitian splitting methods for non-Hermitian positive definite linear systems, Linear Algebra Appl. 428 (2008) 413–440.
[23] C. Gu, Z. Tian, On the HSS iteration methods for positive definite Toeplitz linear systems, J. Comput. Appl. Math. 224 (2009) 709–718.
[24] M.-Q. Jiang, Y. Cao, On local Hermitian and skew-Hermitian splitting iteration methods for generalized saddle point problems, J. Comput. Appl. Math. 231 (2009) 973–982.
[25] A.-L. Yang, J. An, Y.-J. Wu, A generalized preconditioned HSS method for non-Hermitian positive definite linear systems, Appl. Math. Comput. 26 (2010) 1715–1722.
[26] M.-Z. Zhu, On asymmetric Hermitian and skew-Hermitian splitting iteration methods for weakly nonlinear systems, J. Comput. Anal. Appl. 14 (2012) 1321–1331.
[27] M.-Z. Zhu, Modified iteration methods based on the Asymmetric HSS for weakly nonlinear systems, J. Comput. Anal. Appl. 15 (2013) 188–195.
[28] D. Bertaccini, G.H. Golub, S.S. Capizzano, C.T. Possio, Preconditioned HSS methods for the solution of non-Hermitian positive definite linear systems and applications to the discrete convection–diffusion equation, Numer. Math. 99 (2005) 441–484.
[29] C. Greif, J. Varah, Iterative solution of cyclically reduced systems arising from discretization of the three-dimensional convection–diffusion equation, SIAM J. Sci. Comput. 19 (1998) 1918–1940.
[30] C. Greif, J. Varah, Block stationary methods for nonsymmetric cyclically reduced systems arising from three-dimensional elliptic equations, SIAM J. Matrix Anal. Appl. 20 (1999) 1038–1059.
[31] I. Boglaev, A block monotone domain decomposition algorithm for a semilinear convection–diffusion problem, J. Comput. Appl. Math. 173 (2005) 259–277.
[32] I. Boglaev, S. Pack, A uniformly convergent method on arbitrary meshes for a semilinear convection–diffusion problem with discontinuous data, Int. J. Numer. Anal. Model. 5 (2008) 24–39.