East Asian Journal on Applied Mathematics
doi: 10.4208/eajam.150116.240516a

Vol. **6**, No. 4, pp. 367-383
November 2016

A Class of Preconditioned TGHSS-Based Iteration Methods for Weakly Nonlinear Systems

Min-Li Zeng[1] and Guo-Feng Zhang[2,*]

[1] *School of Mathematics, Putian University, Putian 351100, China.*
[2] *School of Mathematics and Statistics, Lanzhou University, Lanzhou 730000, China.*

Received 15 January 2016; Accepted (in revised version) 24 May 2016.

Abstract. In this paper, we first construct a preconditioned two-parameter generalized Hermitian and skew-Hermitian splitting (PTGHSS) iteration method based on the two-parameter generalized Hermitian and skew-Hermitian splitting (TGHSS) iteration method for non-Hermitian positive definite linear systems. Then a class of PTGHSS-based iteration methods are proposed for solving weakly nonlinear systems based on separable property of the linear and nonlinear terms. The conditions for guaranteeing the local convergence are studied and the quasi-optimal iterative parameters are derived. Numerical experiments are implemented to show that the new methods are feasible and effective for large scale systems of weakly nonlinear systems.

AMS subject classifications: 65F10, 65F50

Key words: System of weakly nonlinear equations, GHSS iteration method, local convergence, inner iteration, outer iteration.

1. Introduction

Consider the solution of large sparse system of weakly nonlinear equations of the form

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

where $A \in \mathbb{C}^{n\times n}$ is a non-Hermitian positive definite matrix, i.e., the Hermitian part $H = (A+A^*)/2$ is positive definite. Here A^* denotes the conjugate transpose of the matrix A. Φ: $\mathbb{D} \subset \mathbb{C}^n \to \mathbb{C}^n$ is a continuously differentiable function defined on the open convex domain $\mathbb{D}$ in the n—dimensional complex linear space $\mathbb{C}^n$. The system of nonlinear equation (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. For more details, see [5, 10, 12, 25].

*Corresponding author. *Email addresses:* zengml12@lzu.edu.cn (M.-L. Zeng), gf_zhang@lzu.edu.cn (G.-F. Zhang)

 ©2016 Global-Science Press

https://doi.org/10.4208/eajam.150116.240516a Published online by Cambridge University Press

Large sparse system of nonlinear equations of the form (1.1) arises in many areas of scientific computing and engineering applications, e.g., in finite-difference or sinc discretizations of nonlinear partial differential equations [4,19], in saddle point problems from image processing [9, 11] and in collocation approximations of nonlinear integral equation [23], and so on. See [10, 18] and referees therein.

For solving weakly nonlinear equations (1.1), the Newton iteration method is an efficient method, which can be described as:

$$x^{(k+1)} = x^{(k)} + s^{(k)}, \quad \text{with} \quad F'(x^{(k)})s^{(k)} = -F(x^{(k)}), \quad k = 0, 1, \cdots,$$

where $F'(x^{(k)})$ is the Jacobian matrix of $F(x^{(k)})$. It's known that the Newton iteration method has quadratic convergence speed if a good initial guess $x^{(0)}$ for the equation $F(x) = 0$ is obtained. To simplify or avoid computation of the Jacobian matrix and reduce the cost of the function evaluation, one can consider many variants in terms of approximate, quasi-update, inner/outer or inexact Newton methods, see [3, 5, 7, 10, 26].

Based on the matrix multi-splitting technique, Bai [5, 6] present a class of sequential two-stage iteration methods and an efficient parallel generalizations of the sequential two-stage iteration methods. Recently, by making use of separability and dominance between the linear term Ax and the nonlinear term $\Phi(x)$, Bai and Yang [10] present the Picard-HSS and the nonlinear HSS-like iteration methods based on the Hermitian and skew-Hermitian splitting (HSS) for the large sparse non-Hermitian positive-definite system. Furthermore, Bai and Guo [12] established a Newton-HSS iteration method. This method requires one to explicit form of the Jacobian matrix $F'(x^{(k)}) = A - \Phi'(x^{(k)})$ at each iteration step for the current iterate $x^{(k)}$ and solve two linear sub-systems with the coefficient matrices $\alpha I + H(x^{(k)})$ and $\alpha I + S(x^{(k)})$, where α is a prescribed positive parameter, I is the identity matrix. $H(x^{(k)})$ and $S(x^{(k)})$ are the Hermitian and the skew-Hermitian parts of the Jacobian matrix $F'(x^{(k)})$, i.e.,

$$H(x) = H(F'(x)) \quad \text{and} \quad S(x) = S(F'(x)), \quad \forall x \in \mathbb{D}.$$

Here $H(.)$ and $S(.)$ are the Hermitian part and the skew-Hermitian part of the corresponding matrix, respectively. Obviously, the disadvantages of the Newton-HSS iteration methods are often practically prohibitive due to limited computer memory and the admissible computing time. Therefore, authors of [1, 17, 21, 24–26] made further generalizations.

In this paper, based on the generalization of the Hermitian and skew-Hermitian splitting (GHSS) iteration scheme proposed by Benzi [13] and the two-parameter GHSS (TGHSS) [2] iteration method for solving non-Hermitian positive definite linear system, we will first establish a preconditioned TGHSS (PTGHSS) iteration method. Then we propose a class of PTGHSS-based iteration methods, named by Picard-PTGHSS and nonlinear PTGHSS-like iteration methods for solving large sparse weakly nonlinear equations (1.1). Unlike the HSS iteration as the inner iteration, the new iteration methods consist in the fact that the solution of systems with coefficient matrix $\alpha P + S + K$ by inner iterations is made easier, since this matrix is more "diagonally dominant" (loosely speaking) and typically better conditioned than $\alpha P + S$ [13]. Following [2], the quasi-optimal iterative parameters are given. Therefore, the new iteration methods are easy to be implemented.

https://doi.org/10.4208/eajam.150116.240516a Published online by Cambridge University Press

The organization of the paper is as follows. In Section 2, we review the TGHSS iteration scheme for the non-Hermitian positive definite linear systems and establish the preconditioned TGHSS iteration scheme. In Section 3, we introduce the Picard-PTGHSS and the nonlinear PTGHSS-like iteration methods for solving weakly nonlinear systems. Then we analysis the local convergence properties for the new iteration methods. In Section 5, numerical experiments are used to test the efficiency of the new iteration methods. Finally, in Section 6 some conclusions and remarks are drawn to end this paper.

Throughout the paper, for any vector $x \in \mathbb{C}^n$, we use $\|x\|$ to denote $\sqrt{(x,x)} = \sqrt{x^*x}$, where x^* denotes the conjugate transpose of x. For any matrix $A \in \mathbb{C}^{n\times n}$, define $\|A\| = \max_{\|x\|=1} \|Ax\|$.

2. The PTGHSS Iteration Method and the PTGHSS-Based Iteration Methods

When the nonlinear term $\Phi(x)$ is a constant vector, i.e., $\Phi(x) \equiv b$, the weakly nonlinear equations (1.1) reduces to the linear equations of the following form:

$$Ax = b,$$

where $A \in \mathbb{C}^{n\times n}$ is non-Hermitian positive definite matrix. It's known that the matrix A can be split as

$$A = H + S,$$

where $H = \frac{1}{2}(A+A^*)$ and $S = \frac{1}{2}(A-A^*)$ are the Hermitian parts and skew-Hermitian parts of the matrix A. Based on the Hermitian and skew-Hermitian splitting of the matrix A, Bai, Golub and Ng designed the HSS iteration method for solving the linear equations $Ax = b$ and proved that the HSS iteration method converges unconditionally. The HSS iteration method has gained lots of attention recently years, see [8, 15, 20, 22] and referees therein. Further, by splitting H into the sum of two Hermitian positive semi-definite matrices:

$$H = G + K,$$

where K is of simple form (e.g., diagonal) so that $A = G + (S + K)$, Benzi [13] constructed a generalization of the Hermitian and skew-Hermitian splitting iteration method and Aghazadeh constructed a two-parameter generalized HSS iteration [2]. The TGHSS iteration method for solving the non-Hermitian positive definite linear system $Ax = b$ can be described as the following algorithm.

Algorithm 2.1. The TGHSS iteration method [2].

Given an initial guess $x^{(0)}$, for $k = 0, 1, 2, \cdots$ until $x^{(k)}$ converges, compute

$$\begin{cases} (\alpha I + G)x^{(k+\frac{1}{2})} = (\alpha I - S - K)x^{(k)} + b, \\ (\beta I + S + K)x^{(k+1)} = (\beta I - G)x^{(k+\frac{1}{2})} + b, \end{cases}$$

where α and β are positive parameters. I is an identity matrix. G and K are both Hermitian positive semi-definite matrices.

https://doi.org/10.4208/eajam.150116.240516a Published online by Cambridge University Press

It has been shown in [2] that the TGHSS iteration method allows one to have more control on each of the half-steps of the TGHSS than that of the GHSS iteration method. Therefore, by choosing suitable iterative parameters, the TGHSS iteration method will be more efficient than the GHSS iteration method. To accelerate the convergence of the TGHSS iteration method, we will construct a preconditioned TGHSS (PTGHSS) iteration method for solving the linear system $Ax = b$ as the following algorithm.

Algorithm 2.2. The PTGHSS iteration method.

Given an initial guess $x^{(0)}$, for $k = 0, 1, 2, \cdots$ until $x^{(k)}$ converges, compute

$$\begin{cases} (\alpha P + G)x^{(k+\frac{1}{2})} = (\alpha P - S - K)x^{(k)} + b, \\ (\beta P + S + K)x^{(k+1)} = (\beta P - G)x^{(k+\frac{1}{2})} + b, \end{cases} \tag{2.1}$$

where α and β are positive parameters and P is a Hermitian positive definite matrix. I is an identity matrix. G and K are both Hermitian positive semi-definite matrices.

Obviously, if $P = I$, then the PTGHSS iteration method reduces to the TGHSS iteration method [2]. If $\alpha = \beta$ and $P = I$, then the PTGHSS iteration method reduces to the GHSS iteration method [13].

However, the PTGHSS iteration method can be equivalently rewritten in matrix-vector form as

$$\begin{aligned} x^{(l+1)} &= T(\alpha, \beta)x^{(l)} + G(\alpha, \beta)b \\ &= T(\alpha, \beta)^{l+1}x^{(0)} + \sum_{j=0}^{l} T(\alpha, \beta)^j G(\alpha, \beta)b, \quad l = 1, 2, \cdots, \end{aligned} \tag{2.2}$$

where

$$\begin{aligned} T(\alpha, \beta) &= (\beta P + S + K)^{-1}(\beta P - G)(\alpha P + G)^{-1}(\alpha P - S - K), \\ G(\alpha, \beta) &= (\alpha + \beta)(\beta P + S + K)^{-1}P(\alpha P + G)^{-1}. \end{aligned} \tag{2.3}$$

Because of the separable property of linear and nonlinear terms Ax and $\Phi(x)$, and the former strongly dominant over the latter, following [10], we construct the Picard-PTGHSS iteration method and the nonlinear PTGHSS-like iteration method for solving the system of weakly nonlinear equations (1.1). In the following of this section, we will propose the Picard-PTGHSS and the nonlinear PTGHSS-like iteration methods for solving the system (1.1).

Because the linear term Ax is strongly dominant over the nonlinear term $\Phi(x)$, then we can consider the Picard iteration method:

$$Ax^{(k+1)} = \Phi(x^{(k)}), \qquad k = 0, 1, 2, \cdots,$$

to solve the nonlinear equations (1.1). This naturally leads to the following Picard-PTGHSS iteration method in residual-updating form.

https://doi.org/10.4208/eajam.150116.240516a Published online by Cambridge University Press

Algorithm 2.3. The Picard-PTGHSS iteration method.

Let Φ: $\mathbb{D} \subset \mathbb{C}^n \to \mathbb{C}^n$ be continuously differentiable function. $A \in \mathbb{C}^{n\times n}$ be a non-Hermitian positive definite matrix. Let H and S be the Hermitian and skew-Hermitian parts of A. 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 $k = 0, 1, 2, \cdots$ until $x^{(k)}$ satisfies the stopping criterion by using the following iteration scheme:

(a) Set $s^{(k,0)} = 0$, compute $b^{(k)} := \Phi(x^{(k)}) - Ax^{(k)}$;

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

$$\begin{cases} (\alpha P + G)s^{(k,l+\frac{1}{2})} = (\alpha P - S - K)s^{(k,l)} + b^{(k)}, \\ (\beta P + S + K)s^{(k,l+1)} = (\beta P - G)s^{(k,l+\frac{1}{2})} + b^{(k)}, \end{cases}$$

where α and β are given positive constants and $P \in \mathbb{C}^{n\times n}$ is a prescribed Hermitian positive definite matrix;

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

As is well known that the Picard-PTGHSS iteration method requires to solve the same linear subsystem in inner PTGHSS iterations. Therefore, it's more practical for the Picard-PTGHSS iteration method to carry out the iterative steps than the Newton iteration method. However, the inner PTGHSS iterative steps are often problem-dependent and difficult to be determined in practical computations [10]. To overcome those difficulties, we propose the nonlinear PTGHSS-like iteration method at the end of this section.

Algorithm 2.4. The nonlinear PTGHSS-like iteration method.

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

$$\begin{cases} (\alpha P + G)x^{(k+\frac{1}{2})} = (\alpha P - S - K)x^{(k)} + \Phi(x^{(k)}), \\ (\beta P + S + K)x^{(k+1)} = (\beta P - G)x^{(k+\frac{1}{2})} + \Phi(x^{(k+\frac{1}{2})}), \end{cases}$$

where α and β are given positive constants and $P \in \mathbb{C}^{n\times n}$ is a prescribed Hermitian positive definite matrix.

3. The Convergence Properties of the New Iteration Methods

In this section, we will study the convergence properties of the Picard-PTGHSS and the nonlinear PTGHSS-like iteration methods. According to the local convergence theory established by Bai and Yang [10], we can immediately obtain the following theorem.

Theorem 3.1 ([10]). *Let Φ: $\mathbb{D} \subset \mathbb{C}^n \to \mathbb{C}^n$ 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 $F(x_*) = Ax_* - \Phi(x_*) = 0$.*

https://doi.org/10.4208/eajam.150116.240516a Published online by Cambridge University Press

Suppose $H=(A+A^*)/2=G+K$ with G and K being both Hermitian semi-definite matrices, $S=(A-A^*)/2$. Denote

$$\theta(\alpha,\beta)=\|T(\alpha,\beta)\| \quad \textit{and} \quad \xi=\|A^{-1}\Phi'(x_*)\|.$$

Then there exists an open neighborhood $\mathbb{N}\in\mathbb{N}_0$ of x_* such that for any $x^{(0)}\in\mathbb{N}$ and any sequence of positive integers l_k, $k=0,1,2,\cdots$, the iteration sequence $\{x^{(k)}\}_{k=0}^{\infty}$ generated by the Picard-PTGHSS iteration method is well defined and convergent to x_*, provided $\xi<1$ and $l_0>\lfloor \ln(\frac{1-\xi}{1+\xi})/\ln(\theta(\alpha,\beta)))\rfloor$. Moreover, it holds that,

$$\lim_{k\to\infty}\sup\|x^{(k)}-x_*\|^{\frac{1}{k}}\le\xi+(1+\xi)\theta(\alpha,\beta)^{l_0},\ \textit{with}\ \ l_0=\lim_{k\to\infty}\inf l_k.$$

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

$$\lim_{k\to\infty}\sup\|x^{(k)}-x_*\|^{\frac{1}{k}}\le\xi.$$

According to [10], we can obtain the following local convergence theory for the nonlinear PTGHSS-like iteration method by making use of the Ostrowski Theorem [19].

Theorem 3.2 ([10]). *Assume that Φ: $\mathbb{D}\subset\mathbb{C}^n\to\mathbb{C}^n$ is F-differentiable at a point $x_*\in\mathbb{D}$ such that $Ax_*=\Phi(x_*)$. Denote*

$$L(\alpha,\beta,x_*)=(\beta P+S+K)^{-1}\big(\beta P-G+\Phi'(x_*)\big)(\alpha P+G)^{-1}\big(\alpha P-S-K+\Phi'(x_*)\big).$$

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

By making use of the above result, we can obtain the following theorem.

Theorem 3.3. *Assume that Φ: $\mathbb{D}\subset\mathbb{C}^n\to\mathbb{C}^n$ is F-differentiable at a point $x_*\in\mathbb{D}$ such that $Ax_*=\Phi(x_*)$. Denote*

$$\delta(\alpha,\beta)=\max\Big\{\big\|\Phi(x_*)(\beta P+S+K)^{-1}\big\|,\big\|\Phi(x_*)(\alpha P+G)^{-1}\big\|\Big\},$$

where $T(\alpha,\beta)$ is defined in (2.3)*. Then x_* is a point of attraction of the nonlinear PTGHSS-like iteration provided it satisfies*

$$\delta(\alpha,\beta)<\frac{2(1-ab)}{(a+b)+\sqrt{(a-b)^2+4}},$$

where

$$a=\max_{\lambda_i\in\lambda(P^{-1}G)}\Big|\frac{\beta-\lambda_i}{\alpha+\lambda_i}\Big| \quad \textit{and} \quad b=\max_{\mathrm{i}\mu_j\in\lambda(P^{-1}S),k_j\in\lambda(P^{-1}K)}\sqrt{\frac{(\beta-k_j)^2+\mu_j^2}{(\beta+k_j)^2+\mu_j^2}}.$$

https://doi.org/10.4208/eajam.150116.240516a Published online by Cambridge University Press

Proof. First, we have

$$\begin{aligned}\|T(\alpha,\beta)\| &= \left\|(\beta P-G)(\alpha P+G)^{-1}(\alpha P-S-K)(\beta P+S+K)^{-1}\right\| \\ &\le \left\|(\beta P-G)(\alpha P+G)^{-1}\right\| \left\|(\alpha P-S-K)(\beta P+S+K)^{-1}\right\| \le ab.\end{aligned}$$

By substituting (2.3) into the above inequality, it follows

$$\begin{aligned}&(\beta P+S+K)T(\alpha,\beta;x_*)(\beta P+S+K)^{-1}\\ =&(\beta P+S+K)T(\alpha,\beta)(\beta P+S+K)^{-1}+(\beta P-G)(\alpha P+G)^{-1}\Phi'(x_*)(\beta P+S+K)^{-1}\\ &+\Phi'(x_*)(\alpha P+G)^{-1}(\alpha P-S-K)(\beta P+S+K)^{-1}+\Phi'(x_*)(\alpha P+G)^{-1}\Phi'(x_*)(\beta P+S+K)^{-1}.\end{aligned}$$

Therefore,

$$\begin{aligned}\rho\big(T(\alpha,\beta;x_*)\big) =&\rho\big((\beta P+S+K)T(\alpha,\beta;x_*)(\beta P+S+K)^{-1}\big)\\ \le&\left\|(\beta P+S+K)T(\alpha,\beta;x_*)(\beta P+S+K)^{-1}\right\|\\ \le&\left\|(\beta P+S+K)T(\alpha,\beta)(\beta P+S+K)^{-1}\right\|\\ &+\left\|(\beta P-G)(\alpha P+G)^{-1}\Phi'(x_*)(\beta P+S+K)^{-1}\right\|\\ &+\left\|\Phi'(x_*)(\alpha P+G)^{-1}(\alpha P-S-K)(\beta P+S+K)^{-1}\right\|\\ &+\left\|\Phi'(x_*)(\alpha P+G)^{-1}\Phi'(x_*)(\beta P+S+K)^{-1}\right\|,\\ \le&\|T(\alpha,\beta)\|+\left\|(\beta P-G)(\alpha P+G)^{-1}\right\|\left\|\Phi'(x_*)(\beta P+S+K)^{-1}\right\|\\ &+\left\|\Phi'(x_*)(\alpha P+G)^{-1}\right\|\left\|(\alpha P-S-K)(\beta P+S+K)^{-1}\right\|\\ &+\left\|\Phi'(x_*)(\alpha P+G)^{-1}\right\|\cdot\left\|\Phi'(x_*)(\beta P+S+K)^{-1}\right\|\\ \le&ab+(a+b)\delta(\alpha,\beta)+\delta^2.\end{aligned}$$

If $ab+(a+b)\delta(\alpha,\beta)+\delta^2<1$, then x_* is a point of attraction of the nonlinear PTGHSS-like iteration by making use of Theorem 3.2. Equivalently, if

$$\delta(\alpha,\beta)<\frac{2(1-ab)}{(a+b)+\sqrt{(a-b)^2+4}},$$

then x_* is a point of attraction of the nonlinear PTGHSS-like iteration.

From the proof of Theorem 3.3, we can see that if $\Phi'(x_*)$ is reasonably small compared with the matrix A, then $\rho(L(\alpha,\beta;x_*))<1$ is valid under the condition of Theorem 3.3.

4. Quasi-Optimal Parameters

In this section, we will study the quasi-optimal parameters. As is shown in [21], the quasi-optimal parameters can be obtained by minimizing the lowest upper bound of the

https://doi.org/10.4208/eajam.150116.240516a Published online by Cambridge University Press

spectral radius of the iterative matrix $T(\alpha,\beta)$. Because

$$\begin{aligned}
&(\beta P+S+K)T(\alpha,\beta)(\beta P+S+K)^{-1}\\
=&(\beta P-G)(\alpha P+G)^{-1}(\alpha P-S-K)(\beta P+S+K)^{-1}\\
\sim&P^{-\frac{1}{2}}(\beta P-G)P^{-\frac{1}{2}}P^{\frac{1}{2}}(\alpha P+G)^{-1}P^{\frac{1}{2}}P^{-\frac{1}{2}}(\alpha P-S-K)P^{-\frac{1}{2}}P^{\frac{1}{2}}(\beta P+S+K)^{-1}P^{\frac{1}{2}}\\
=&(\beta I-\bar{G})(\alpha I+\bar{G})^{-1}(\alpha I-\bar{S}-\bar{K})(\beta I+\bar{S}+\bar{K})^{-1}:=\hat{T}(\alpha,\beta),
\end{aligned}$$

where $\bar{G}=P^{-\frac{1}{2}}GP^{-\frac{1}{2}}$, $\bar{K}=P^{-\frac{1}{2}}KP^{-\frac{1}{2}}$, $\bar{S}=P^{-\frac{1}{2}}SP^{-\frac{1}{2}}$. Obviously, $\bar{G}$ and $\bar{K}$ are both Hermitian semi-definite, $\bar{S}$ is skew-Hermitian. Therefore, by making use of Corollary 2.3 in [2], we can obtain the quasi-optimal value for α and β immediately.

Theorem 4.1. *Let α and β be two nonnegative constants, P be a Hermitian positive matrix. Split $A(=(G+K)+S)$, where G and K are Hermitian positive semi-definite matrices. $H=G+K$ and S are the Hermitian parts and skew-Hermitian parts of the matrix A, respectively. Denote*

$$\begin{aligned}
\lambda_{\max}&=\max_{\lambda_k\in\lambda(P^{-1}G)}\{\lambda_k\},\qquad \lambda_{\min}=\min_{\lambda_k\in\lambda(P^{-1}G)}\{\lambda_k\},\\
k_{\max}&=\max_{k_j\in\lambda(P^{-1}K)}\{k_j\},\qquad k_{\min}=\min_{k_j\in k(P^{-1}K)}\{k_j\},
\end{aligned}$$

and

$$\begin{aligned}
\Delta&=\sqrt{k_{\min}^2+4(\lambda_{\min}+\lambda_{\max})k_{\min}+16\lambda_{\min}\lambda_{\max}}\,,\\
\tilde{\beta}(\alpha)&=\frac{\alpha(\lambda_{\max}+\lambda_{\min})+2\lambda_{\min}\lambda_{\max}}{2\alpha+\lambda_{\min}+\lambda_{\max}},\\
\tilde{\sigma}(\alpha)&=\frac{\lambda_{\max}^2+\alpha(\lambda_{\max}-\lambda_{\min})-\lambda_{\min}\lambda_{\max}}{(2\alpha+\lambda_{\min}+\lambda_{\max})(\alpha+\lambda_{\max})}.
\end{aligned}$$

(1) If G is positive definite, K is positive semi-definite and α is set to be α^*, then the optimal parameter β is

$$\beta_{opt}=\begin{cases}\alpha^*, & \alpha^*>\sqrt{\lambda_{\min}\lambda_{\max}},\\ \tilde{\beta}(\alpha^*), & \text{otherwise}.\end{cases}$$

Moreover, the corresponding upper bound is given by

$$\nu(\alpha^*,\beta_{opt})=\begin{cases}\dfrac{\alpha^*-\lambda_{\min}}{\alpha^*+\lambda_{\min}}, & \alpha^*>\sqrt{\lambda_{\min}\lambda_{\max}},\\ \tilde{\sigma}(\alpha^*), & \text{otherwise}.\end{cases}$$

(2) If G is positive semi-definite, K is positive definite and β is set to be β^*, then the optimal parameter α is

$$\alpha_{opt}=\beta^*+\frac{1}{2}k_{\min}.$$

https://doi.org/10.4208/eajam.150116.240516a Published online by Cambridge University Press

Moreover, the corresponding upper bound is given by

$$\nu(\alpha_{opt},\beta^*)=\begin{cases}\dfrac{\lambda_{\max}-\beta^*}{\lambda_{\max}+\beta^*+\lambda_{\min}/2}, & \beta^*\le\tilde{\beta}(\alpha_{opt}),\\ \dfrac{\beta^*}{\beta^*+\lambda_{\min}/2}, & \beta^*>\tilde{\beta}(\alpha_{opt}).\end{cases}$$

(3) If G and K are both positive definite matrices and α is set to be α^*, then the optimal parameter β is

$$\beta_{opt}=\begin{cases}\alpha^*-\dfrac{1}{2}k_{\min}, & \alpha^*>\dfrac{1}{2}(k_{\min}+\sqrt{\Delta}),\\ \tilde{\beta}(\alpha^*), & otherwise.\end{cases}$$

Moreover, the corresponding upper bound is given by

$$\nu(\alpha^*,\beta_{opt})=\begin{cases}\dfrac{\alpha^*-k_{\min}/2-\lambda_{\min}}{\alpha^*+\lambda_{\min}}, & \alpha^*>\dfrac{1}{4}(k_{\min}+\sqrt{\Delta}),\\ \tilde{\sigma}(\alpha^*), & otherwise.\end{cases}$$

5. Numerical Results

In this section, we will consider the solutions of the three-dimensional nonlinear convection-diffusion equation [2, 8]

$$\begin{cases}-(u_{xx}+u_{yy}+u_{zz})+q(u_x+u_y+u_z)=\sin(u+1), & \text{for } (x,y)\in\Omega,\\ u(x,y,z)=0, & \text{for } (x,y)\in\partial\Omega,\end{cases}\tag{5.1}$$

where q is a positive constant used to measure the magnitudes of the convective terms and $\Omega=(0,1)\times(0,1)\times(0,1)$, with $\partial\Omega$ being its boundary.

It's well known that by applying the centered differences or the first order upwind approximations to the convective terms to the problem (5.1) will lead to the nonlinear equation (1.1) with the coefficient matrix

$$A=T_x\otimes I\otimes I+I\otimes T_y\otimes I+I\otimes I\otimes T_z,\tag{5.2}$$

where $h=1/(n+1)$ is used in all of the three directions, $\otimes$ denotes the Kronecker product. T_x, T_y and T_z are tridiagonal matrices given by

$$T_x=\text{tridiag}(t_2,t_1,t_3),\quad T_y=\text{tridiag}(t_2,0,t_3)\quad\text{and}\quad T_z=\text{tridiag}(t_2,0,t_3),$$

with $t_1=6$, $t_2=-1-r$, $t_3=-1+r$ if the first order derivatives are approximated by the centered difference scheme. If the first order derivatives are approximated by the upwind difference scheme, then it holds

$$t_1=6+6r,\qquad t_2=-1-2r,\qquad t_3=-1.$$

https://doi.org/10.4208/eajam.150116.240516a Published online by Cambridge University Press

Here, $r = qh/2$ is the mesh Reynold number. For more details, we refer to [8, 16].

The tests are performed in Matlab R2013a with machine precision 10^{-16}, RAM 4.00GB, 32bit and terminated when the current residual satisfies $\|r_k\|/\|r_0\| < 10^{-6}$, where r_k is the residual at the k-th iteration. We use the zero vector as the initial guess.

In actual computations, we take the initial guess to be $x^{(0)}$. The stopping criterion for the Newton iteration is set to be

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

The stopping criterion for all the inner iterations is set to be

$$\frac{\|F'(x^{(k)})s^{(k,l_k)} + F(x^{(k)})\|}{\|F(x^{(k)})\|} \leq \eta,$$

where η is a prescribed tolerance for controlling the accuracy of all the inner iterations.

For the coefficient matrix of (5.2), the Hermitian parts H and the skew-Hermitian parts S are

$$H = H_x \otimes I \otimes I + I \otimes H_y \otimes I + I \otimes I \otimes H_z$$

and

$$S = S_x \otimes I \otimes I + I \otimes S_y \otimes I + I \otimes I \otimes S_z,$$

respectively, where

$$H_x = \text{tridiag}\Big(\frac{t_2+t_3}{2}, t_1, \frac{t_2+t_3}{2}\Big), \qquad H_y = H_z = \text{tridiag}\Big(\frac{t_2+t_3}{2}, 0, \frac{t_2+t_3}{2}\Big),$$
$$S_p = \text{tridiag}\Big(\frac{t_2-t_3}{2}, t_1, -\frac{t_2-t_3}{2}\Big), \quad p \in \{x, y, z\}.$$

To make a further demonstration of the advantage of the new methods, we will compare the Picard-PTGHSS and the nonlinear PTGHSS-like iteration methods with the Picard-PGHSS and the nonlinear GPHSS-like iteration methods [21] in terms of iteration counts and computing times. In our implementations, we adopt the preconditioner as diag(A) for the Picard-PTGHSS and the nonlinear PTGHSS-like iteration methods [14].

Furthermore, by splitting $H = G + K$, where

$$G = T \otimes I \otimes I + I \otimes T \otimes I + I \otimes I \otimes T$$

with $T = \text{diag}(-1, 2, -1)$ [2], we can split the coefficient matrix in (5.2) as $A = G + (K + S)$ and obtain the Picard-PTGHSS and the nonlinear PTGHSS-like iteration methods.

The iterative parameters α and β in the Newton-PGHSS, Picard-PGHSS, the nonlinear PGHSS-like iteration method, the Picard-PTGHSS and the nonlinear PTGHSS-like iteration method are simply fixed by $\alpha = 1$ and $\beta = 1$. For the sake of fairness, we also adopt the preconditioner as diag(A) for the Picard-GPHSS and the nonlinear GPHSS-like iteration methods.

https://doi.org/10.4208/eajam.150116.240516a Published online by Cambridge University Press

We also compare the new iteration methods with the Newton-GMRES method and the Newton-GPHSS method. The preconditioner diag(A) is also used in the latter.

Tables 1-4 list the numbers of inner, outer and total iteration steps (denoted as IT_{in} and IT_{out}) for the Picard-GPHSS and the Picard-PTGHSS iteration method, and the total CPU time (denoted as CPU) for the nonlinear GPHSS-like and the nonlinear PTGHSS-like iteration method with respect to different problem sizes (i.e., $N = 64, 512, 1728, 4096, 8000$) and tolerance (i.e., $\eta = 0.1, 0.01, 0.001$). The experimental results for $q = 100$ and $q = 1000$ are listed in Tables 1-2 and Tables 3-4, respectively. The experimental results for the nonlinear equations arises from the approximation of centered difference scheme for the problem (5.1) are listed in Table 1 and from the approximation of upwind difference scheme for the problem (5.1) are listed in Table 3. "-" in all the tables means that the corresponding methods are not convergent after 1000 iterations or the CPU time is exceeding 2000 in second.

From Tables 1–4, we see that the Newton-GMRES method and the Newton-GPHSS method require fewer inner iteration counts to meet the specified accuracy.

From those tables, we can also find when $q = 1000$, the PTGHSS-based iteration methods become more effective both in iteration counts and CPU times. According to [13], when the same iterative parameters are selected, the coefficient matrices of the PTGHSS-based iteration methods are more "diagonally dominant" and better conditioned. Therefore, the PTGHSS-based will be more efficient, especially in the case that the problem size increases. Moreover, it can be seen from Tables 1-4 that, as the problem dimension increases, the nonlinear PTGHSS-like iteration method obviously performs better than the nonlinear GPHSS-like iteration method both in terms of iteration counts and CPU times when $q = 1000$.

On the whole, by simply taking the parameters $\alpha = 1$, $\beta = 1$ and from the experimental results, we find that the PTGHSS-based iteration methods are not worse than the optimal GPHSS-based iteration methods. Particularly, if we can choose the iterative parameters properly, we can make the coefficient matrices of the systems to be solved always "diagonal dominant".

Therefore, the PTGHSS-based iteration methods outperform the GPHSS-based iteration methods and the Newton-based iteration methods for solving the large sparse tested problem in terms of the computing time.

6. Concluding Remarks

In this paper, we establish a class of preconditioned TGHSS-based iteration methods for solving large scale systems of weakly nonlinear equations, i.e., the Picard-PTGHSS and the nonlinear PTGHSS-like iteration methods. The convergence properties for the new methods are studied in detail. The quasi-optimal iterative parameters are given theoretically. From the experimental results, we can find that the PTGHSS-based iteration methods seem to be more effective in CPU times than the GPHSS-based iteration methods and the Newton-based iteration methods [21] as the problem size increases. According to [13], we can know that it dues to the main reason that the generalized scheme consists in the fact that the solution of systems with coefficient matrix $\beta P + S + K$ by inner iterations is made easier, since this

https://doi.org/10.4208/eajam.150116.240516a Published online by Cambridge University Press

Table 1: Numerical results for nonlinear systems from the centered difference scheme ($q = 100$).

	Method	N :	64	512	1728	4096	8000	13824
$\eta = 10^{-1}$	Newton-GMRES	IT_{in}	20	15	12	12	-	-
		IT_{out}	6	7	7	5	-	-
		CPU	0.05	0.39	17.67	205.1	-	-
	Newton-GPHSS	IT_{in}	4	5	5	4	-	-
		IT_{out}	6	6	7	7	-	-
		CPU	0.04	0.58	28.88	214.34	-	-
	Picard-GPHSS	IT_{in}	3	4	4	4	5	-
		IT_{out}	6	7	8	11	14	-
		CPU	0.008	0.16	2.19	64.84	267.01	-
	Picard-PTGHSS	IT_{in}	3	4	4	4	4	5
		IT_{out}	6	7	8	11	14	19
		CPU	0.008	0.15	2.11	64.77	266.91	1421.4
$\eta = 10^{-2}$	Newton-GMRES	IT_{in}	20	24	15	13	-	-
		IT_{out}	4	4	5	5	-	-
		CPU	0.03	0.25	15.23	226.9	-	-
	Newton-GPHSS	IT_{in}	8	10	9	6	-	-
		IT_{out}	4	4	5	5	-	-
		CPU	0.37	0.54	18.28	230.81	-	-
	Picard-GPHSS	IT_{in}	6	6	5	6	7	-
		IT_{out}	5	6	8	10	12	-
		CPU	0.09	0.19	2.19	64.51	194.5	-
	Picard-PTGHSS	IT_{in}	5	5	5	5	5	5
		IT_{out}	5	6	8	10	12	12
		CPU	0.09	0.17	2.15	64.39	184.7	1416.8
$\eta = 10^{-3}$	Newton-GMRES	IT_{in}	20	28	18	15	-	-
		IT_{out}	4	4	4	4	-	-
		CPU	0.04	0.34	12.38	222.7	-	-
	Newton-GPHSS	IT_{in}	11	14	9	8	-	-
		IT_{out}	3	4	4	5	-	-
		CPU	0.26	0.52	14.44	239.62	-	-
	Picard-GPHSS	IT_{in}	8	9	9	9	11	-
		IT_{out}	5	6	8	10	12	-
		CPU	0.15	0.21	2.53	86.03	192.8	-
	Picard-PTGHSS	IT_{in}	8	8	8	8	9	9
		IT_{out}	5	6	8	10	12	14
		CPU	0.15	0.18	2.38	84.71	186.5	1447.7
	GPHSS-like	IT	20	25	25	24	24	25
		CPU	0.01	0.04	0.34	1.11	4.53	15.58
	PTGHSS-like	IT	**20**	**25**	**25**	**24**	**24**	**25**
		CPU	**0.009**	**0.04**	**0.19**	**1.09**	**4.47**	**15.29**

matrix is more "diagonally dominant" (loosely speaking) and typically better conditioned than $\beta P + S$.

If we can find a more convenient way to choose the optimal parameters and precondi-

https://doi.org/10.4208/eajam.150116.240516a Published online by Cambridge University Press

Table 2: Numerical results for nonlinear systems from the centered difference scheme ($q = 1000$).

	Method	N :	64	512	1728	4096	8000	13824
$\eta = 10^{-1}$	Newton-GMRES	IT_{in}	13	14	15	15	-	-
		IT_{out}	6	6	6	7	-	-
		CPU	0.18	1.42	46.52	503.54	-	-
	Newton-GPHSS	IT_{in}	10	52	41	53	-	-
		IT_{out}	6	6	6	10	-	-
		CPU	0.03	0.51	25.64	866.26	-	-
	Picard-GPHSS	IT_{in}	6	10	10	12	12	-
		IT_{out}	6	6	6	6	6	-
		CPU	0.008	0.24	2.79	68.11	274.96	-
	Picard-PTGHSS	IT_{in}	6	8	8	10	11	11
		IT_{out}	6	6	6	6	6	6
		CPU	0.008	0.22	2.62	66.37	272.19	1426.6
$\eta = 10^{-2}$	Newton-GMRES	IT_{in}	17	18	18	19	-	-
		IT_{out}	4	5	5	5	-	-
		CPU	0.18	1.42	46.52	503.54	-	-
	Newton-GPHSS	IT_{in}	19	102	83	262	-	-
		IT_{out}	3	4	4	8	-	-
		CPU	0.03	1.87	27.35	1006.29	-	-
	Picard-GPHSS	IT_{in}	11	19	17	17	18	-
		IT_{out}	4	4	5	5	5	-
		CPU	0.009	0.39	3.08	69.39	190.7	-
	Picard-PTGHSS	IT_{in}	10	15	15	16	17	17
		IT_{out}	4	4	5	5	5	6
		CPU	0.008	0.33	2.92	66.87	188.5	1464.4
$\eta = 10^{-3}$	Newton-GMRES	IT_{in}	18	19	19	20	-	-
		IT_{out}	3	3	4	4	-	-
		CPU	0.008	0.12	9.01	121.98	-	-
	Newton-GPHSS	IT_{in}	29	150	124	391	-	-
		IT_{out}	2	3	3	17	-	-
		CPU	0.007	0.22	7.62	1762.11	-	-
	Picard-GPHSS	IT_{in}	15	28	26	27	27	-
		IT_{out}	3	4	5	5	5	-
		CPU	0.009	0.57	3.91	107.61	196.7	-
	Picard-PTGHSS	IT_{in}	12	15	16	20	22	27
		IT_{out}	3	4	5	5	5	6
		CPU	0.009	0.45	3.22	99.36	183.6	1472.3
	GPHSS-like	IT	45	96	95	118	101	93
		CPU	0.01	0.31	1.55	7.75	23.31	54.93
	PTGHSS-like	IT	**45**	**96**	**95**	**118**	**101**	**93**
		CPU	**0.01**	**0.31**	**1.55**	**7.59**	**23.17**	**54.74**

tioner, we can obtain a class of more practical and effective PTGHSS-based iteration methods, which will be our further consideration in future study.

https://doi.org/10.4208/eajam.150116.240516a Published online by Cambridge University Press

Table 3: Numerical results for nonlinear systems from the upwind difference scheme ($q = 100$).

	Method	N :	64	512	1728	4096	8000	13824
$\eta = 10^{-1}$	Newton-GMRES	IT_{in}	4	2	4	2	-	-
		IT_{out}	6	6	6	7	-	-
		CPU	0.01	0.31	18.24	291.06	-	-
	Newton-GPHSS	IT_{in}	4	3	4	5	-	-
		IT_{out}	5	6	6	7	-	-
		CPU	0.008	0.54	24.18	310.76	-	-
	Picard-GPHSS	IT_{in}	6	6	12	12	13	-
		IT_{out}	3	7	6	6	13	-
		CPU	0.04	0.14	2.28	65.43	270.84	-
	Picard-PTGHSS	IT_{in}	3	6	11	12	12	13
		IT_{out}	6	7	6	6	13	15
		CPU	0.004	0.14	2.19	65.38	268.17	1492.2
$\eta = 10^{-2}$	Newton-GMRES	IT_{in}	10	6	12	15	-	-
		IT_{out}	3	4	4	5	-	-
		CPU	0.09	0.21	12.12	211.25	-	-
	Newton-GPHSS	IT_{in}	6	7	8	9	-	-
		IT_{out}	3	3	4	5	-	-
		CPU	0.005	0.31	15.26	258.98	-	-
	Picard-GPHSS	IT_{in}	6	12	17	19	19	-
		IT_{out}	4	5	5	9	14	-
		CPU	0.005	0.18	2.32	65.73	194.5	-
	Picard-PTGHSS	IT_{in}	6	12	15	17	17	22
		IT_{out}	4	5	5	9	9	11
		CPU	0.005	0.17	2.28	65.61	188.7	1488.4
$\eta = 10^{-3}$	Newton-GMRES	IT_{in}	10	12	18	20	-	-
		IT_{out}	3	3	4	4	-	-
		CPU	0.008	0.19	12.11	161.89	-	-
	Newton-GPHSS	IT_{in}	8	9	10	17	-	-
		IT_{out}	2	3	4	4	-	-
		CPU	0.005	0.35	14.85	187.81	-	-
	Picard-GPHSS	IT_{in}	9	15	25	25	25	-
		IT_{out}	4	5	5	9	14	-
		CPU	0.005	0.21	3.19	100.81	197.9	-
	Picard-PTGHSS	IT_{in}	9	14	23	23	23	25
		IT_{out}	4	5	5	9	14	19
		CPU	0.005	0.19	3.07	99.36	192.1	1493.1
	GPHSS-like	IT	16	25	37	48	57	65
		CPU	0.003	0.03	0.26	1.58	5.91	19.79
	PTGHSS-like	IT	**14**	**22**	**35**	**44**	**53**	**61**
		CPU	**0.002**	**0.03**	**0.13**	**0.52**	**1.66**	**4.65**

Acknowledgments

The author would like to thank the referees for the helpful suggestions. This work was supported by the National Natural Science Foundation of China (11271174, 11471175,

https://doi.org/10.4208/eajam.150116.240516a Published online by Cambridge University Press

Table 4: Numerical results for nonlinear systems from the upwind difference scheme ($q = 1000$).

	Method	N :	64	512	1728	4096	8000	13824
$\eta = 10^{-1}$	Newton-GMRES	IT_{in}	4	4	4	5	-	-
		IT_{out}	5	6	6	6	-	-
		CPU	0.007	0.31	18.36	259.73	-	-
	Newton-GPHSS	IT_{in}	2	3	3	4	-	-
		IT_{out}	6	6	6	6	-	-
		CPU	0.01	0.51	25.64	263.93	-	-
	Picard-GPHSS	IT_{in}	3	5	7	9	11	-
		IT_{out}	6	6	6	6	7	-
		CPU	0.004	0.14	2.14	65.04	267.21	-
	Picard-PTGHSS	IT_{in}	3	4	4	5	7	13
		IT_{out}	6	6	6	6	7	5
		CPU	0.004	0.12	1.93	59.43	203.57	1487.7
$\eta = 10^{-2}$	Newton-GMRES	IT_{in}	4	5	5	5	-	-
		IT_{out}	3	3	3	6	-	-
		CPU	0.009	0.17	9.05	259.73	-	-
	Newton-GPHSS	IT_{in}	5	6	8	9	-	-
		IT_{out}	3	3	3	3	-	-
		CPU	0.007	0.28	11.21	142.87	-	-
	Picard-GPHSS	IT_{in}	5	9	15	19	23	-
		IT_{out}	3	4	4	5	5	-
		CPU	0.003	0.14	2.18	64.98	193.9	-
	Picard-PTGHSS	IT_{in}	4	6	9	12	17	27
		IT_{out}	3	4	4	5	5	5
		CPU	0.003	0.14	2.04	60.17	187.3	1438.8
$\eta = 10^{-3}$	Newton-GMRES	IT_{in}	10	8	8	7		
		IT_{out}	2	2	3	3	-	-
		CPU	0.008	0.12	9.01	121.98	-	-
	Newton-GPHSS	IT_{in}	7	11	18	9	-	-
		IT_{out}	2	2	2	3	-	-
		CPU	0.005	0.22	7.62	145.45	-	-
	Picard-GPHSS	IT_{in}	8	14	19	25	33	-
		IT_{out}	3	3	4	4	5	-
		CPU	0.008	0.18	2.84	100.16	299.3	-
	Picard-PTGHSS	IT_{in}	7	9	13	19	27	31
		IT_{out}	3	3	4	4	5	5
		CPU	0.007	0.16	2.21	62.73	188.4	1440.1
	GPHSS-like	IT	15	23	30	37	44	51
		CPU	0.003	0.03	0.25	1.23	5.21	18.04
	PTGHSS-like	IT	**11**	**20**	**28**	**33**	**39**	**47**
		CPU	**0.002**	**0.02**	**0.09**	**0.38**	**1.31**	**3.74**

11511130015), the Natural Science Foundation of Fujian Province (2016J05016), the Scientific Research Project of Fujuan Educational Bureau (JAT160450) and the Scientific Research Project of Putian University (2015061, 2016021, 2016075).

https://doi.org/10.4208/eajam.150116.240516a Published online by Cambridge University Press

References

[1] N. Aghazadeh, M. Bastani and D.K. Salkuyeh, *Generalized Hermitian and skew-Hermitian splitting iterative method for image restoration*, Appl. Math. Model., 39 (2015), pp. 6126–6138.

[2] N. Aghazadeh, S.D. Khojasteh and M. Bastani, *Two-parameter generalized Hermitian and skew-Hermitian splitting iteration method*, Int. J. Comput. Math., 2015, DOI:10.1080/00207160.2015.1019873.

[3] H.-B. An and Z.-Z. Bai, *NGLM: A globally convergent Newton-GMRES method*, Math. Numer. Sinica, 27 (2005), pp. 151–174.

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

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

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

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

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

[9] Z.-Z. Bai, Y.-M. Huang and M.K. Ng, *On preconditioned iterative methods for Burgers equations*, SIAM J. Sci. Comput., 29 (2007), pp. 415–439.

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

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

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

[13] M. Benzi, *A generalization of the Hermitian and skew-Hermitian splitting iteration*, SIAM J. Matrix Anal. Appl., 31 (2009), pp. 360–374.

[14] D. Bertaccini, G.H. Golub, S.S. Capizzano and 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), pp. 441–484.

[15] Y. Cao and Z.-R. Ren, *Two variants of the PMHSS iteration method for a class of complex symmetric indefinite linear systems*, Appl. Math. Comput., 264 (2015), pp. 61–71.

[16] H.C. Elman and G.H. Golub, *Iterative methods for cyclically reduced nonselfadjoint linear systems*, Math. Comp., 54 (1990), pp. 671–700.

[17] X. Li and Y.-J. Wu, *Accelerated Newton-GPSS methods for systems of nonlinear equations*, J. Comput. Anal. Appl., 17 (2014), pp. 245–254.

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

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

[20] H.N. Pour and H.S. Goughery, *New Hermitian and skew-Hermitian splitting methods for non-Hermitian positive-definite linear systems*, Numer. Algorithms, 69 (2015), pp. 207–225.

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

[22] D.K. Salkuyeh, *The Picard-HSS iteration method for absolute value equations*, Optim. Lett., 8 (2014), pp. 2191–2202.

https://doi.org/10.4208/eajam.150116.240516a Published online by Cambridge University Press

[23] T. Tang, *Superconvergence of numerical solutions to weakly singular Volterra integro-differential equations*, Numer. Math., 61 (1992), pp. 373–382.
[24] J.-J. Zhang, *The relaxed nonlinear PHSS-like iteration method for absolute value equations*, Appl. Math. Comput., 265 (2015), pp. 266–274.
[25] 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), pp. 5095–5104.
[26] M.-Z. Zhu and G.-F. Zhang, *A class of iteration methods based on the HSS for Toeplitz systems of weakly nonlinear equations*, J. Comput. Appl. Math., 290 (2015), pp. 433–444.

https://doi.org/10.4208/eajam.150116.240516a Published online by Cambridge University Press