Computers and Mathematics with Applications 71 (2016) 213–223

Contents lists available at ScienceDirect

Computers and Mathematics with Applications

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

A shifted nested splitting iterative method with applications to ill-posed problems and image restoration

Mohammad Khorsand Zak [a,*], Faezeh Toutounian [b,c]

[a] Department of Applied Mathematics, Aligoudarz Branch, Islamic Azad University, Aligoudarz, Iran
[b] Department of Applied Mathematics, School of Mathematical Sciences, Ferdowsi University of Mashhad, Mashhad, Iran
[c] The Center of Excellence on Modelling and Control Systems, Ferdowsi University of Mashhad, Mashhad, Iran

ARTICLE INFO

Article history:
Received 18 May 2015
Received in revised form 28 September 2015
Accepted 9 November 2015
Available online 30 November 2015

Keywords:
CGNR
Hermitian and skew-Hermitian splitting
Tikhonov regularization
Image restoration

ABSTRACT

We present a shifted nested iterative method for solving systems of linear equations with a coefficient matrix that contains a dominant skew-Hermitian part. This new scheme is practically the inner/outer iterations, which employs the CGNR method as inner iteration to approximate each outer iterate, while each outer iteration is induced by a convergent splitting of the coefficient matrix. Convergence properties of the new scheme are studied in depth and possible choices of the shift parameter are discussed. Moreover, an adapted version of the method is used for ill-posed problems and image restoration. At the last, numerical examples are used to further examine the effectiveness and robustness of the new method.

1. Introduction

Many problems in science and engineering require the solution of a system of linear equations

$$Ax = b, \tag{1}$$

where $A \in \mathbb{C}^{n\times n}$ is a large sparse matrix. The iterative solution of this system requires some forms of splitting. Among these forms of splitting is the Hermitian and skew-Hermitian splitting [1], which has been shown to have advantages over many existing matrix splitting iteration methods for the solution of Hermitian systems or positive definite linear systems [2,1,3,4]. In the Hermitian and skew-Hermitian splitting, the coefficient matrix A is decomposed into its Hermitian and skew-Hermitian components. We consider this splitting as

$$A = H + S, \tag{2}$$

where $H = (A + A^H)/2$ and $S = (A - A^H)/2$. When H dominates S, we have $\|H^{-1}S\|_2 < 1$ and may use nested splitting conjugate gradient (NSCG) method [2]; but if S dominates H, it means that $\|H^{-1}S\|_2$ is large and therefore we cannot guarantee the convergence of NSCG method. When the skew-Hermitian part S is dominant, it is typically harder to solve the linear system (1) and appears challenging from a numerical point of view [5,6]. In this paper, we present an efficient iterative method, named NS-CGNR, for cases in which the skew-Hermitian component S dominates the Hermitian component H. Note that here the condition of dominant skew-Hermitian part is the sufficient condition but not necessary.

* Corresponding author.
E-mail addresses: mo.khorsand@stu-mail.um.ac.ir (M. Khorsand Zak), toutouni@math.um.ac.ir (F. Toutounian).

http://dx.doi.org/10.1016/j.camwa.2015.11.005
0898-1221/

In the remainder of this paper, we use $\lambda(M)$ and $\|M\|_2$ to denote the eigenvalue and the spectral norm of a matrix $M \in \mathbb{C}^{n\times n}$, respectively. Note that $\|.\|_2$ is also used to represent the 2-norm of a vector. Also, it must be reminded that $A = B - C$ is called a splitting of the matrix A if B is nonsingular. This splitting is a convergent splitting if $\rho(B^{-1}C) < 1$; and a contractive splitting if $\|B^{-1}C\| < 1$ for some matrix norm. For matrix $\mathcal{B}$, we denote by $\kappa(\mathcal{B}) = \|\mathcal{B}\|_2\|\mathcal{B}^{-1}\|_2$ its Euclidean condition number, and for a Hermitian positive definite matrix $\mathcal{B}$, we define the $\|\cdot\|_{\mathcal{B}}$ norm of a vector $x \in \mathbb{C}^n$ as $\|x\|_{\mathcal{B}} = \sqrt{x^H\mathcal{B}x}$. Then the induced $\|\cdot\|_{\mathcal{B}}$ norm of a matrix $H \in \mathbb{C}^{n\times n}$ is define as $\|H\|_{\mathcal{B}} = \|\mathcal{B}^{\frac{1}{2}}H\mathcal{B}^{-\frac{1}{2}}\|_2$. In addition, it holds that $\|Hx\|_{\mathcal{B}} \le \|H\|_{\mathcal{B}}\|x\|_{\mathcal{B}}$, $\|H\|_{\mathcal{B}} \le \sqrt{\kappa(\mathcal{B})}\|H\|_2$ and $\|I\|_{\mathcal{B}} = 1$, where I is the identity matrix.

The organization of this paper is as the following. Section 2 represents the establishment and the analysis of the NS-CGNR method for the linear system (1) and its convergence properties. Section 3 presents an adapted version of the NS-CGNR method for ill-posed problems and image restoration. Section 4 is devoted to numerical experiments and included some experimental results and applications in ill-posed problems and image restoration to illustrate the effectiveness of our approach. Finally, we demonstrate our conclusions in Section 5.

2. Establishment and analysis of the NS-CGNR method

Consider the Hermitian and skew-Hermitian splitting (2). Since the matrix S may be singular, we introduce a shift ($\nu > 0$) and define quasi-Hermitian splitting as

$$A = (H - \nu I) + (S + \nu I) = H_\nu + S_\nu. \tag{3}$$

Then the system of linear equations (1) is equivalent to the fixed-point equation

$$S_\nu x = b - H_\nu x.$$

Given an initial guess $x^{(0)} \in \mathbb{C}^n$, assume that we have computed approximations $x^{(1)}, x^{(2)}, \ldots, x^{(l)}$ to the solution $x^* \in \mathbb{C}^n$ of the system (1). Then the next approximation $x^{(l+1)}$ may be defined as either an exact or inexact solution of the system of linear equations

$$S_\nu x = b - H_\nu x^{(l)}. \tag{4}$$

Now we solve the system of linear equations (4) by the CGNR method [7]. An implementation of the NS-CGNR method is given by the following algorithm. In this algorithm, $k_{\max}$ and $j_{\max}$ are the largest admissible number of the outer and inner iteration steps, respectively. $x^{(0)}$ is an initial guess for the solution, and the outer and inner stopping tolerances are denoted by ϵ and η, respectively.

Algorithm 2.1. The NS-CGNR algorithm

1. $x^{(0,0)} = x^{(0)}$
2. $r^{(0)} = b - Ax^{(0)}$
3. For $k = 0, 1, 2, \ldots, k_{\max}$ Do:
4. $\quad \hat{b} = b - H_\nu x^{(k,0)}$
5. $\quad \hat{r}^{(0)} = \hat{b} - S_\nu x^{(k,0)}$
6. $\quad z^{(0)} = S_\nu^T \hat{r}^{(0)}$, and $p^{(0)} = z^{(0)}$
7. $\quad$ For $j = 0, 1, 2, \ldots, j_{\max}$ Do:
8. $\quad\quad w^{(j)} = S_\nu p^{(j)}$
9. $\quad\quad \alpha_j = \frac{\|z^{(j)}\|_2^2}{\|w^{(j)}\|_2^2}$
10. $\quad\quad x^{(k,j+1)} = x^{(k,j)} + \alpha_j p^{(j)}$
11. $\quad\quad \hat{r}^{(j+1)} = \hat{r}^{(j)} - \alpha_j w^{(j)}$
12. $\quad\quad$ If $\|\hat{r}^{(j+1)}\|_2 \le \eta\|\hat{r}^{(0)}\|_2$ GoTo 17
13. $\quad\quad z^{(j+1)} = S_\nu^T \hat{r}^{(j+1)}$
14. $\quad\quad \beta_j = \frac{\|z^{(j+1)}\|_2^2}{\|z^{(j)}\|_2^2}$
15. $\quad\quad p^{(j+1)} = z^{(j+1)} + \beta_j p^{(j)}$
16. $\quad$ End Do
17. $\quad x^{(k+1)} = x^{(k,j)}$
18. $\quad r^{(k+1)} = b - Ax^{(k+1)}$
19. $\quad$ If $\|r^{(k+1)}\|_2 \le \epsilon\|r^{(0)}\|_2$ Stop
20. $\quad x^{(k+1,0)} = x^{(k+1)}$
21. End Do

To be reminded, for the splitting $A = M - N$, convergence of stationary iterative methods is obtained if the spectral radius of $M^{-1}N$ satisfies the condition $\rho(M^{-1}N) < 1$ [7]. Therefore, the parameter ν should be chosen as the splitting satisfies the condition $\rho(S_\nu^{-1}H_\nu) < 1$.

From the work of Golub and Vandrestraeten [8], If

$$\lambda_{\min}(H)\lambda_{\max}(H) > \min_{\lambda \in \Lambda(S)} |\lambda(S)|, \tag{5}$$

then there exists an ν for which $\rho(S_\nu^{-1}H_\nu) < 1$. Moreover, using

$$\nu = \frac{\lambda_{\min}(H) + \lambda_{\max}(H)}{2}, \tag{6}$$

can cause to decrease the upper bound of $\rho(S_\nu^{-1}H_\nu)$, see [8].

Now, we can prove the following theorem for the convergence of NS-CGNR method, which is a direct conclusion of Theorem 3.1 in [2].

Theorem 2.1. *Let $A \in \mathbb{C}^{n\times n}$ be a nonsingular and non-Hermitian matrix, and $A = H_\nu + S_\nu$ be a contractive (with respect to the $\|\cdot\|_{S_\nu^T S_\nu}$-norm). Suppose that the NS-CGNR method begins from an initial guess $x^{(0)} \in \mathbb{C}^n$, and produces an iterative sequence $\{x^{(l)}\}_{l=0}^{\infty}$, where $x^{(l)} \in \mathbb{C}^n$ is the lth approximation to the solution $x^* \in \mathbb{C}^n$ of the system of linear equations (1), obtained by solving the linear system (4) with k_l steps of CGNR iterations. Then*

(a) $\|x^{(l)} - x^*\|_{S_\nu^H S_\nu} \le \gamma^{(l)} \|x^{(l-1)} - x^*\|_{S_\nu^H S_\nu}$, $l = 1, 2, 3, \ldots,$

(b) $\|b - Ax^{(l)}\|_{S_\nu^H S_\nu} \le \widetilde{\gamma}^{(l)} \|b - Ax^{(l-1)}\|_{S_\nu^H S_\nu}$, $l = 1, 2, 3, \ldots,$

where

$$\gamma^{(l)} = 2\left(\frac{\kappa(S_\nu) - 1}{\kappa(S_\nu) + 1}\right)^{k_l}(1 + \varrho) + \varrho, \qquad \widetilde{\gamma}^{(l)} = \gamma^{(l)}\kappa(S_\nu)\frac{1+\varrho}{1-\varrho}, \quad l = 1, 2, 3, \ldots$$

and $\varrho = \|S_\nu^{-1}H_\nu\|_{S_\nu^H S_\nu} = \|H_\nu S_\nu^{-1}\|_2$.

Moreover, for some $\gamma \in (\varrho, \varrho_1)$ with $\varrho_1 = \min\{1, 2 + 3\varrho\}$, and

$$k_l \ge \frac{\ln((\gamma - \varrho)/(2(1+\varrho)))}{\ln((\kappa(S_\nu) - 1)/(\kappa(S_\nu) + 1))}, \quad l = 1, 2, 3, \ldots,$$

we have $\gamma^{(l)} \le \gamma$ $(l = 1, 2, 3, \ldots)$, and the sequence $\{x^{(l)}\}_{l=0}^{\infty}$ converges to the solution x^* of the system of linear equations (1). For $\varrho \in (0, r)$, which r is the positive root of quadratic equation $\kappa(S_\nu)\varrho^2 + (\kappa(S_\nu) + 1)\varrho - 1 = 0$, and some $\widetilde{\gamma} \in ((1+\varrho)\varrho\kappa(S_\nu)/(1-\varrho), 1)$, and

$$k_l \ge \frac{\ln(((1-\varrho)\widetilde{\gamma} - \varrho(1+\varrho)\kappa(S_\nu))/(2(1+\varrho)^2\kappa(S_\nu)))}{\ln((\kappa(S_\nu) - 1)/(\kappa(S_\nu) + 1))}, \quad l = 1, 2, 3, \ldots,$$

we have $\widetilde{\gamma}^{(l)} \le \widetilde{\gamma}$ $(l = 1, 2, 3, \ldots)$, and the residual sequence $\{b - Ax^{(l)}\}_{l=0}^{\infty}$ converges to zero.

Proof. See Appendix.

3. Application to ill-posed problems and image restoration

Image restoration and ill-posed problems assign an application where we may encounter linear system of equations (1). In image restoration, the obstacle would be reconstruction of an original image that has been digitized and degraded by blur and additive noise. Consider the following data production model for image restoration:

$$Bf = g. \tag{7}$$

The right-hand side vector g in (7) represents the available output and is supposed to be imbrued by an error (noise) η, i.e., $g = \widehat{g} + \eta$, where $\widehat{g}$ is the unknown error-free right-hand side. The matrix B represents the blurring matrix, the vector f that should be approximated symbolizes the original image, the vector η is the additive noise and the vector g represents the blurred and noisy (degraded) image. Some treatments and overviews on image restoration can be found in [9,10].

In general, B is a large and ill-conditioned matrix. The Tikhonov regularization method [11] is used to solve the system (7). Thus, we transform (7) into an equivalent system as follows:

$$\min_f \left(\|Bf - g\|_2^2 + \mu^2\|Lf\|_2^2\right), \tag{8}$$

where μ is a penalty positive parameter and L is an auxiliary operator chosen as either the identity or low order differential operator. The parameter μ depends on the level of noise and is small. In this paper, we consider that L is the identity matrix and we obtain the optimal value of the regularization parameter μ by routine gcv from Hansen's regularization tools package [12]. To solve (8) and obtain the minimum, we can transform the problem (8) into an equivalent normal equation

$$(B^TB+\mu^2I)f=B^Tg, \tag{9}$$

where I denotes the identity matrix. Similar to [13], the normal equation (9) can be translated into an equivalent blocked system

$$\underbrace{\begin{pmatrix} I & B \\ -B^T & \mu^2 I\end{pmatrix}}_{A}\underbrace{\begin{pmatrix} t \\ f\end{pmatrix}}_{x}=\underbrace{\begin{pmatrix} g \\ 0\end{pmatrix}}_{b}, \tag{10}$$

where $t=g-Bf$. Now, we have

$$\begin{aligned} A=\begin{pmatrix} I & B \\ -B^T & \mu^2 I\end{pmatrix}&=\begin{pmatrix} I & 0 \\ 0 & \mu^2 I\end{pmatrix}+\begin{pmatrix} 0 & B \\ -B^T & 0\end{pmatrix}\\ &=\begin{pmatrix} I & 0 \\ 0 & \mu^2 I\end{pmatrix}-\begin{pmatrix} \nu I & 0 \\ 0 & \nu I\end{pmatrix}+\begin{pmatrix} \nu I & 0 \\ 0 & \nu I\end{pmatrix}+\begin{pmatrix} 0 & B \\ -B^T & 0\end{pmatrix}\\ &=\begin{pmatrix} (1-\nu)I & 0 \\ 0 & (\mu^2-\nu)I\end{pmatrix}+\begin{pmatrix} \nu I & B \\ -B^T & \nu I\end{pmatrix}=H_\nu+S_\nu. \end{aligned} \tag{11}$$

Therefore, one can use the Algorithm 2.1 for the image restoration problem. To reach this goal, various innovations and substitutions are needed.

Select an initial guess $f^{(0)}$ for system (7) and set $t^{(0)}=g-Bf^{(0)}$. Therefore, the initial approximation solution for system (10) and its corresponding residual are

$$x^{(0)}=\begin{pmatrix} t^{(0)} \\ f^{(0)}\end{pmatrix},$$

and

$$r^{(0)}=b-Ax^{(0)}=\begin{pmatrix} g \\ 0\end{pmatrix}-\begin{pmatrix} I & B \\ -B^T & \mu^2 I\end{pmatrix}\begin{pmatrix} t^{(0)} \\ f^{(0)}\end{pmatrix}=\begin{pmatrix} 0 \\ B^Tt^{(0)}-\mu^2 f^{(0)}\end{pmatrix}=\begin{pmatrix} r_1^{(0)} \\ r_2^{(0)}\end{pmatrix}.$$

Also, we have

$$\widehat{b}=b-H_\nu x^{(k,0)}=\begin{pmatrix} 9 \\ 0\end{pmatrix}-\begin{pmatrix} (1-\nu)I & 0 \\ 0 & (\mu^2-\nu)I\end{pmatrix}\begin{pmatrix} t^{(k,0)} \\ f^{(k,0)}\end{pmatrix}=\begin{pmatrix} g-(1-\nu)t^{(k,0)} \\ -(\mu^2-\nu)f^{(k,0)}\end{pmatrix},$$

and

$$\begin{aligned} \widehat{r}^{(0)}&=\widehat{b}-S_\nu x^{(k,0)}=\begin{pmatrix} g-(1-\nu)t^{(k,0)} \\ -(\mu^2-\nu)f^{(k,0)}\end{pmatrix}-\begin{pmatrix} \nu I & B \\ -B^T & \nu I\end{pmatrix}\begin{pmatrix} t^{(k,0)} \\ f^{(k,0)}\end{pmatrix}\\ &=\begin{pmatrix} 0 \\ B^Tt^{(k,0)}-\mu^2 f^{(k,0)}\end{pmatrix}=\begin{pmatrix} \widehat{r}_1^{(0)} \\ \widehat{r}_2^{(0)}\end{pmatrix},\\ z^{(0)}&=S_\nu^T\widehat{r}^{(0)}=\begin{pmatrix} \nu I & -B \\ B^T & \nu I\end{pmatrix}\begin{pmatrix} 0 \\ \widehat{r}_2^{(0)}\end{pmatrix}=\begin{pmatrix} -B\widehat{r}_2^{(0)} \\ \nu\widehat{r}_2^{(0)}\end{pmatrix}=\begin{pmatrix} z_1^{(0)} \\ z_2^{(0)}\end{pmatrix}\\ p^{(0)}&=z^{(0)}\Rightarrow\begin{pmatrix} p_1^{(0)} \\ p_2^{(0)}\end{pmatrix}=\begin{pmatrix} z_1^{(0)} \\ z_2^{(0)}\end{pmatrix}. \end{aligned}$$

Furthermore, for $j=0,1,2,\ldots$, we have

$$\begin{aligned} w^{(j)}&=S_\nu p^{(j)}=\begin{pmatrix} \nu I & B \\ -B^T & \nu I\end{pmatrix}\begin{pmatrix} p_1^{(j)} \\ p_2^{(j)}\end{pmatrix}=\begin{pmatrix} \nu p_1^{(j)}+Bp_2^{(j)} \\ -B^Tp_1^{(j)}+\nu p_2^{(j)}\end{pmatrix}=\begin{pmatrix} w_1^{(j)} \\ w_2^{(j)}\end{pmatrix},\\ \alpha_j&=\frac{\|z^{(j)}\|_2^2}{\|w^{(j)}\|_2^2}=\frac{\|z_1^{(j)}\|_2^2+\|z_2^{(j)}\|_2^2}{\|w_1^{(j)}\|_2^2+\|w_2^{(j)}\|_2^2},\\ x^{(k,j+1)}&=x^{(k,j)}+\alpha_j p^{(j)}=\begin{pmatrix} t^{(k,j)}+\alpha_j p_1^{(j)} \\ f^{(k,j)}+\alpha_j p_2^{(j)}\end{pmatrix}=\begin{pmatrix} t^{(k,j+1)} \\ f^{(k,j+1)}\end{pmatrix},\\ \widehat{r}^{(j+1)}&=\widehat{r}^{(j)}-\alpha_j w^{(j)}=\begin{pmatrix} \widehat{r}_1^{(j)}-\alpha_j w_1^{(j)} \\ \widehat{r}_2^{(j)}-\alpha_j w_2^{(j)}\end{pmatrix}=\begin{pmatrix} \widehat{r}_1^{(j+1)} \\ \widehat{r}_2^{(j+1)}\end{pmatrix}, \end{aligned}$$

$$z^{(j+1)} = S_\nu^T \widehat{r}^{(j+1)} = \begin{pmatrix} \nu I & -B \\ B^T & \nu I \end{pmatrix} \begin{pmatrix} \widehat{r}_1^{(j+1)} \\ \widehat{r}_2^{(j+1)} \end{pmatrix} = \begin{pmatrix} \nu \widehat{r}_1^{(j+1)} - B\widehat{r}_2^{(j+1)} \\ B^T \widehat{r}_1^{(j+1)} + \nu \widehat{r}_2^{(j+1)} \end{pmatrix} = \begin{pmatrix} z_1^{(j+1)} \\ z_2^{(j+1)} \end{pmatrix},$$

$$\beta_j = \frac{\|z^{(j+1)}\|_2^2}{\|z^{(j)}\|_2^2} = \frac{\|z_1^{(j+1)}\|_2^2 + \|z_2^{(j+1)}\|_2^2}{\|z_1^{(j)}\|_2^2 + \|z_2^{(j)}\|_2^2},$$

$$p^{(j+1)} = z^{(j+1)} + \beta_j p^{(j)} = \begin{pmatrix} z_1^{(j+1)} + \beta_j p_1^{(j)} \\ z_2^{(j+1)} + \beta_j p_2^{(j)} \end{pmatrix} = \begin{pmatrix} p_1^{(j+1)} \\ p_2^{(j+1)} \end{pmatrix}.$$

Therefore, the Algorithm 2.1 can be rewritten as the following algorithm:

Algorithm 3.1. The NS-CGNR algorithm for ill-posed problems and image restoration

---

1. $t^{(0)} = g - Bf^{(0)}$
2. $r_1^{(0)} = 0\,, r_2^{(0)} = B^T t^{(0)} - \mu^2 f^{(0)}$
3. $f^{(0,0)} = f^{(0)}\,, t^{(0,0)} = t^{(0)}$
4. For $k = 0, 1, \cdots, k_{\max}$ Do
5. $\quad \widehat{r}_1^{(0)} = 0\,, \widehat{r}_2^{(0)} = B^T t^{(k,0)} - \mu^2 f^{(k,0)}$
6. $\quad z_1^{(0)} = -B\widehat{r}_2^{(k,0)}\,, z_2^{(0)} = \nu \widehat{r}_2^{(k,0)}$
7. $\quad p_1^{(0)} = z_1^{(0)}\,, p_2^{(0)} = z_2^{(0)}$
8. $\quad$ For $j = 0, 1, \cdots, j_{\max}$ Do
9. $\qquad w_1^{(j)} = \nu p_1^{(j)} + Bp_2^{(j)}\,, w_2^{(j)} = -B^T p_1^{(j)} + \nu p_2^{(j)}$
10. $\qquad \alpha_j = \frac{||z_1^{(j)}||_2^2 + ||z_2^{(j)}||_2^2}{||w_1^{(j)}||_2^2 + ||w_2^{(j)}||_2^2}$
11. $\qquad t^{(k,j+1)} = t^{(k,j)} + \alpha_j P_1^{(j)}\,, f^{(k,j+1)} = f^{(k,j)} + \alpha_j P_2^{(j)}$
12. $\qquad \widehat{r}_1^{(j+1)} = \widehat{r}_1^{(j)} - \alpha_j w_1^{(j)}\,, \widehat{r}_2^{(j+1)} = \widehat{r}_2^{(j)} - \alpha_j w_2^{(j)}$
13. $\qquad$ If $(||\widehat{r}_1^{(j+1)}||_2 + ||\widehat{r}_2^{(j+1)}||_2) \le \eta(||\widehat{r}_1^{(0)}||_2 + ||\widehat{r}_2^{(0)}||_2)$ GoTo 17
14. $\qquad z_1^{(j+1)} = \nu \widehat{r}_1^{(j+1)} - B\widehat{r}_2^{(j+1)}\,, z_2^{(j+1)} = B^T \widehat{r}_1^{(j+1)} + \nu \widehat{r}_2^{(j+1)}$
15. $\qquad \beta_j = \frac{||z_1^{(j+1)}||_2^2 + ||z_2^{(j+1)}||_2^2}{||z_1^{(j)}||_2^2 + ||z_2^{(j)}||_2^2}$
16. $\qquad p_1^{(j+1)} = z_1^{(j+1)} + \beta_j p_1^{(j)}\,, p_2^{(j+1)} = z_2^{(j+1)} + \beta_j p_2^{(j)}$
17. $\quad$ End Do
18. $\quad t^{(k+1)} = t^{(k,j+1)}\,, f^{(k+1)} = f^{(k,j+1)}$
19. $\quad r_2^{(k+1)} = B^T t^{(k+1)} - \mu^2 f^{(k+1)}$
20. $\quad$ If $||r_2^{(k+1)}||_2 \le \varepsilon ||r_2^{(0)}||_2$ Stop
21. $\quad t^{(k+1,0)} = t^{(k+1)}\,, f^{(k+1,0)} = f^{(k+1)}$
22. End Do

---

4. Numerical experiments

All numerical experiments presented in this section were computed in double precision with a number of MATLAB codes. All iterations started from the zero vector for initial $x^{(0)}$ and terminated when the current iterate satisfied $\|r^{(k)}\|_2 \le 10^{-10}\|r^{(0)}\|_2$. Moreover, we use $\eta = 10^{-3}$ for the inner error tolerance in the inner iterations. For each experiment, we report the number of total outer iterations and total CPU time (in parentheses), and compare the NS-CGNR method with the other methods. At first, we use the Algorithm 2.1 to solve an advection–diffusion equation in Example 4.1.

Example 4.1. Consider the constant coefficient advection–diffusion equation

$$-\triangle u + \beta \frac{\partial u}{\partial x} = f, \tag{12}$$

with $\beta \ge 0$ on the unit square $(0, 1) \times (0, 1)$ and Dirichlet conditions prescribed on the boundary [8]. The domain is discredited with mesh size h and solved by the finite difference method. The coefficient matrix A is considered as $A = H + S$. The symmetric part H has 5 nonzero elements per row and the skew-Hermitian part S is a block diagonal matrix where every block is given by

$$S_{ii} = \frac{\beta}{2h}\mathrm{tridiag}(-1, 0, 1), \quad \text{for } i = 1, 2, \ldots, \frac{1}{h}.$$

The results for this example are given in Tables 1 and 2.

Table 1
Results for Example 4.1 with $h = 0.01$.

$\log\beta$	$\|S\|_2/\|H\|_2$	NS-CGNR	GMRES(20)	IHSS
3	1.6001e+0	109(0.053)	24(0.125)	14(0.325)
4	1.6001e+1	32(0.091)	51(0.277)	18(0.511)
5	1.6001e+2	8(0.037)	300(1.372)	23(1.102)
6	1.6001e+3	8(0.028)	346(1.622)	35(1.568)

Table 2
Results for Example 4.1 with $\beta = 10^5$.

h	$\|S\|_2/\|H\|_2$	NS-CGNR	GMRES(20)	IHSS
1/32	5.0018e+2	5(0.018)	21(0.047)	5(0.112)
1/64	2.5003e+2	6(0.034)	92(0.418)	5(0.093)
1/128	1.2500e+2	27(0.109)	391(1.953)	19(1.531)
1/256	6.2500e+1	66(0.624)	227(1.447)	48(1.839)
1/512	3.1250e+1	67(1.652)	140(1.738)	59(1.980)

Table 1 indicates the fact that as β increases, i.e., when the skew-Hermitian component becomes more and more dominant, the NS-CGNR method is more efficient than the GMRES(20) method and the IHSS method (see [1]).

By focusing on the results presented in Tables 1 and 2, one can observe that when $\|S\|_2 > \|H\|_2$, the NS-CGNR method is superior to the GMRES(20) method and the IHSS method in both terms of iteration numbers and CPU time.

Now, we use the Algorithm 3.1 to solve some ill-posed problems with noisy right hand sides. In all examples, to generate a "noisy" right hand side g, we use MATLAB code `g=g+1e-3*rand(size(g))`.

Example 4.2. Consider test problem `shaw` from [12] with $n = 100$. It is discretization of a Fredholm integral equation of the first kind

$$\int_a^b K(s,t)f(t)dt = g(s), \quad c \le s \le d, \tag{13}$$

with $[-\frac{\pi}{2}, \frac{\pi}{2}]$ as both integration intervals. The kernel K and the solution f are given by

$$\begin{aligned} K(s,t) &= (\cos(s)+\cos(t))^2\left(\frac{\sin(u)}{u}\right)^2 \\ u &= \pi(\sin(s)+\sin(t)) \\ f(t) &= a_1\exp(-c_1(t-t_1)^2) + a_2\exp(-c_2(t-t_2)^2). \end{aligned}$$

The parameters a_1, a_2, etc., are constants that determine the shape of the solution f; in this implementation we use $a_1 = 2$, $a_2 = 1$, $c_1 = 6$, $c_2 = 2$, $t_1 = 0.8$, $t_2 = -0.5$ giving an f with two "humps" [12]. The integral equation (13) is a one dimensional model of an image reconstruction problem from [14].

Fig. 1 presents the results of this problem, while comparing the exact solution against the solutions obtained from the Algorithm 3.1 (NS-CGNR), the GMRES(20) method for system (7), the SVD method and the TSVD(k) method with $k = 5$ for system (7) and a CG-like method for regularized system (9), indicates the relative efficiency of NS-CGNR method over other methods except TSVD(5) and the CG-like methods. The optimal value of the Tikhonov regularization parameter for this example is $\mu = 3.5559\text{e}{-03}$.

Example 4.3. Consider test problem `baart` from [12] with $n = 100$. It is discretization of a Fredholm integral equation (13) of the first kind, with kernel K and right hand side g given by

$$K(s,t) = \exp(s\cos t), \qquad g(s) = 2\frac{\sin s}{s},$$

and with integration intervals $s \in [0, \frac{\pi}{2}]$ and $t \in [0, \pi]$. The solution is given by $f(t) = \sin(t)$ [12].

We present the results of this problem in Fig. 2, as comparing the exact solution against the solutions obtained from the Algorithm 3.1 (NS-CGNR), the GMRES(20) method for system (7), the SVD method and the TSVD(k) method with $k = 5$ for system (7) and a CG-like method for regularized system (9), shows the relative efficiency of NS-CGNR method over other methods except the CG-like method. The optimal value of the Tikhonov regularization parameter for this example is $\mu = 8.8831\text{e}{-03}$.

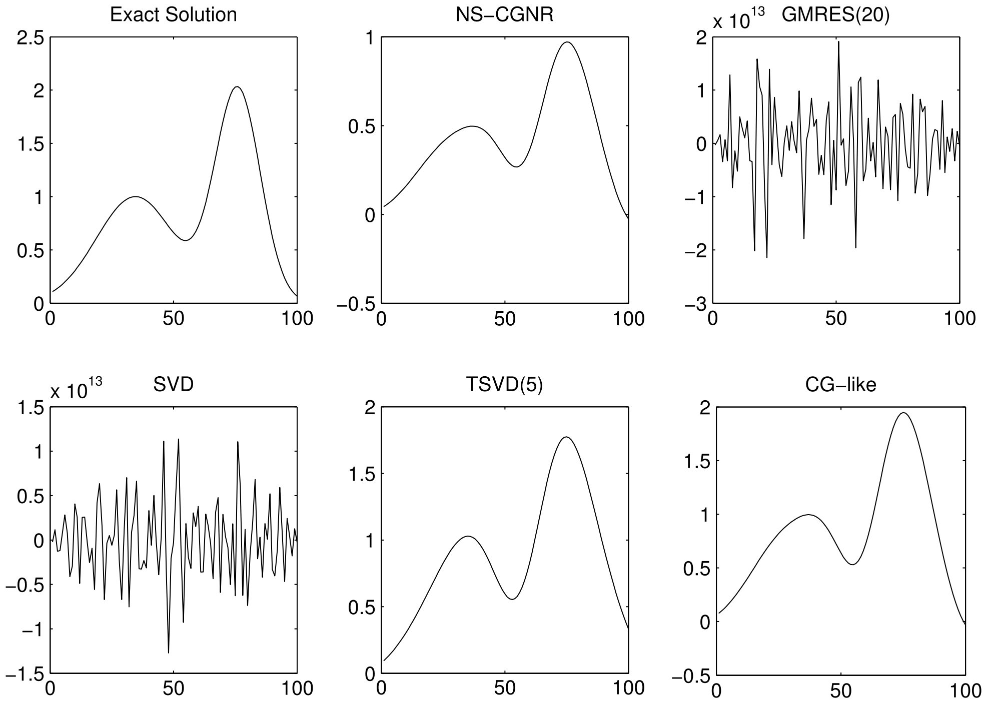

Fig. 1. Results for test problem shaw(100).

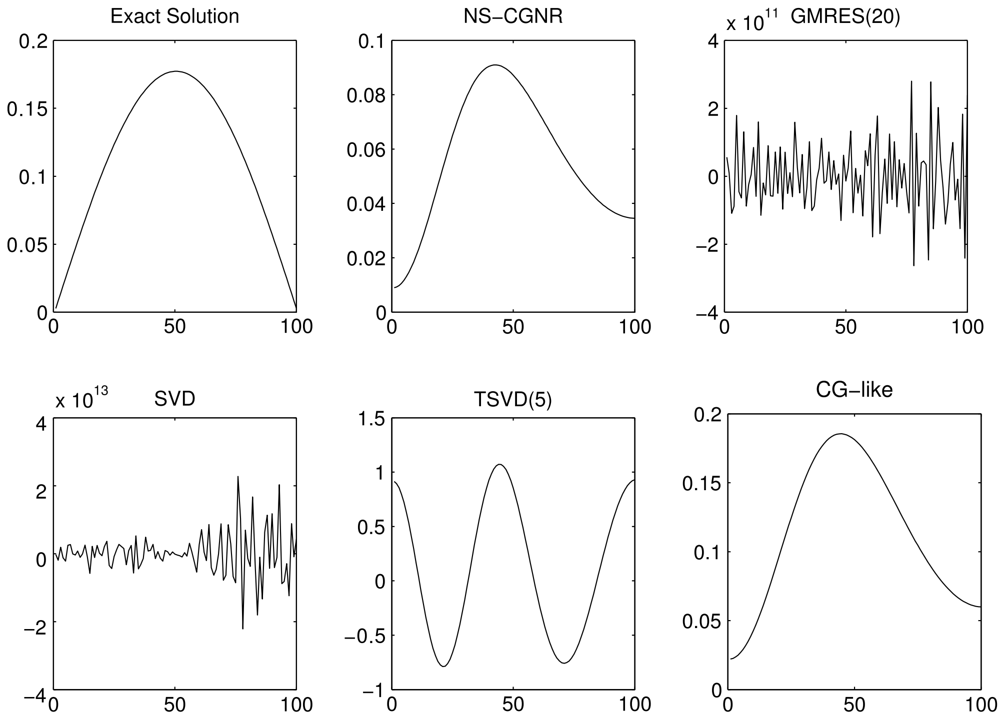

Fig. 2. Results for test problem baart(100).

In the next two examples, we restore blurred and noisy images by some different methods. These methods are the NS-CGNR (Algorithm 3.1), the GMRES(20) method and the special HSS (SpHSS) method [13] which applies on (7), and the CG-like method which applies on (9).

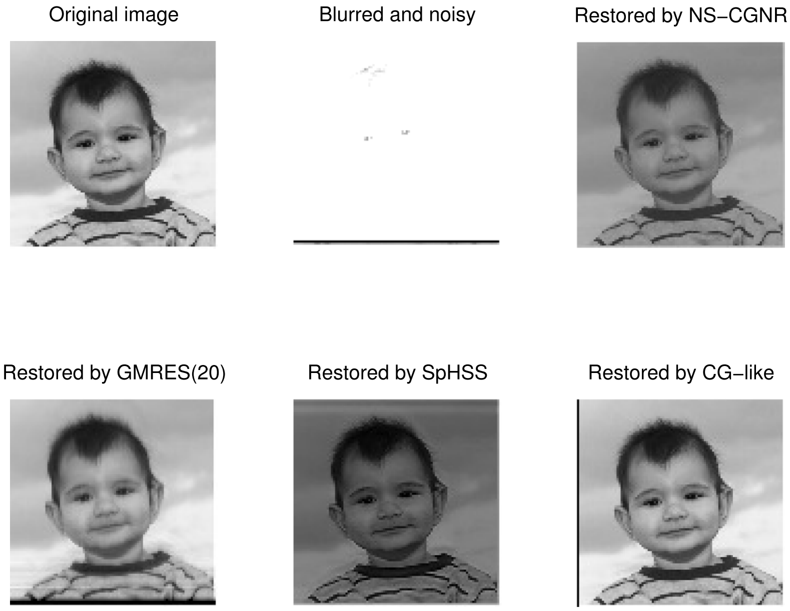

Fig. 3. Results for boy image in the Example 4.4.

Example 4.4. In this example, the original image is a 100×100 boy image. The blurring matrix B is given by $B = I \otimes (H + 10S) \in \mathbb{R}^{100^2 \times 100^2}$, where I is the identity matrix and $H = [h_{ij}]$ is a matrix of dimension 100×100 given by

$$h_{ij} = \begin{cases} \dfrac{1}{\sigma\sqrt{2\pi}} \exp\left(-\dfrac{(i-j)^2}{2\sigma^2}\right), & |i-j| \le r, \\ 0 & \text{otherwise.} \end{cases}$$

while S is a block diagonal matrix as in the Example 4.1 of dimension 100×100. We let $\sigma = 1$ and $r = 5$.

The original image and the blurred and noisy image are shown in the top left and top center of the Fig. 3, respectively. We applied the NS-CGNR method (Algorithm 3.1) and the other methods to the blurred and noisy image and the restored images are shown in the Fig. 3. The restored image by the NS-CGNR method was obtained in 4 outer iterations and the required CPU-time was 0.11 s. The restored image by the GMRES(20) method was obtained in 1000 outer iterations and the required CPU-time was 59.63 s. The restored image by the SpHSS method was shown after 5000 outer iterations and 367.71 s. The restored image by the CG-like method was obtained in 9 outer iterations and the required CPU-time was 0.10 s. The optimal value of the Tikhonov regularization parameter for this example is $\mu = 6.1037\text{e}{-06}$.

Example 4.5. In this example, the original image is a 100×100 peppers image which is shown in the Fig. 4. The blurring matrix B is given by $B = I \otimes K + I \otimes rS \in \mathbb{R}^{100^2 \times 100^2}$, where I is the identity matrix and $K = [k_{ij}]$ is a matrix of dimension 100×100 given by

$$k_{ij} = \begin{cases} \dfrac{1}{2r-1}, & |i-j| \le r, \\ 0 & \text{otherwise.} \end{cases} \tag{14}$$

and S is a block diagonal matrix as in the Example 4.1 of dimension 100×100. We let $r = 3$ in this example.

The original image and the blurred and noisy image are shown in the top left and top center of Fig. 4, respectively. We applied the NS-CGNR method (Algorithm 3.1) and the other methods to the blurred and noisy image and the restored images are shown in the Fig. 4. The restored image by the NS-CGNR method was obtained in 6 outer iterations and the required CPU-time was 0.13 s. The restored image by the GMRES(20) method was obtained in 1000 outer iterations and the required CPU-time was 55.1 3 s. The restored image by the SpHSS method was shown after 5000 outer iterations 314.86 s. The restored image by the CG-like method was obtained in 24 outer iterations and the required CPU-time was 0.15 s. The optimal value of the Tikhonov regularization parameter for this example is $\mu = 1.0987\text{e}{-05}$.

5. Conclusion

A shifted nested iteration method is considered for a class of problems where the skew-Hermitian part of the matrix is dominating. It has been demonstrated that a fixed-point iteration solved by inner iterations to a low inner accuracy

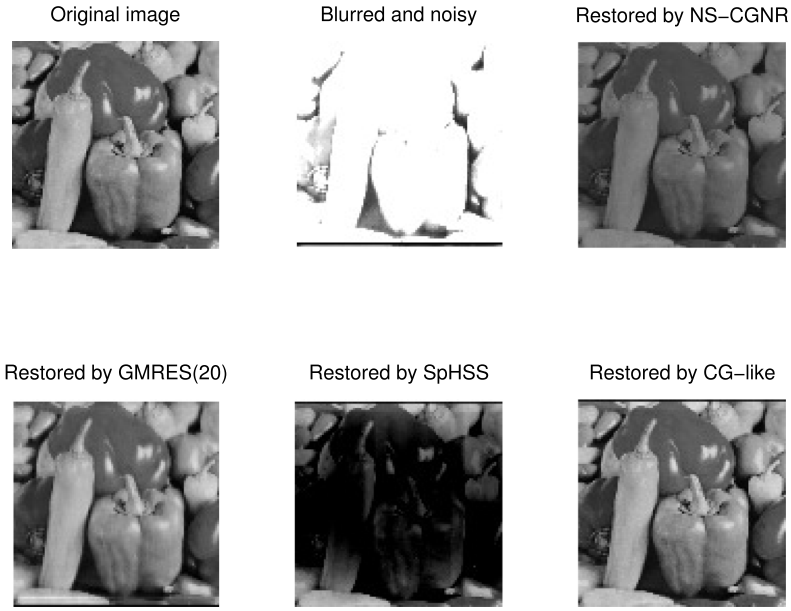

Fig. 4. Results for peppers image in the Example 4.5.

can obtain a method which is more effective. A typical application of this method is for advection–diffusion equations. As expected, the NS-CGNR method gets more accurate as the skew-Hermitian part of the matrix becomes more dominating. Furthermore, the results in the previous section, prove the method to be effective for ill-posed problems and image restoration.

Acknowledgments

The authors are grateful to the anonymous referees for their comments which substantially improved the quality of this paper. The work of the first author was supported in part by the Research Council of I.A.U. Aligoudarz Branch (Grant No. 6643334334).

Appendix

To prove Theorem 2.1, we need the following lemmas.

Lemma A.1 (*[15]*). *If F is an $n \times n$ matrix with $\|F\| < 1$, then $(I+F)^{-1}$ exists and satisfies*

$$\|(I+F)^{-1}\| \leq \frac{1}{1-\|F\|}.$$

Lemma A.2 (*[7]*). *Let $A \in \mathbb{C}^{n\times n}$ be a Hermitian positive definite matrix, and assume that the system of linear equations (1) is solved by the conjugate gradient method. If $x^{(0)} \in \mathbb{C}^n$ is the starting vector, $x^{(k)} \in \mathbb{C}^n$ the kth iterate, and $x^* \in \mathbb{C}^n$ the exact solution of the linear system of Eqs. (1), then*

$$\|x^{(k)} - x^*\|_A \leq 2\left(\frac{\sqrt{\kappa(A)}-1}{\sqrt{\kappa(A)}+1}\right)^k \|x^{(0)} - x^*\|_A.$$

Evidently, by making use of Lemma A.2, we can obtain the following corollary:

Corollary A.3. *Let $A \in \mathbb{C}^{n\times n}$, and assume that the system of linear equations (1) is solved by the CGNR method. If $x^{(0)} \in \mathbb{C}^n$ is the starting vector, $x^{(k)} \in \mathbb{C}^n$ the kth iterate, and $x^* \in \mathbb{C}^n$ the exact solution of the system of linear equations (1), then*

$$\|x^{(k)} - x^*\|_{A^HA} \leq 2\left(\frac{\kappa(A)-1}{\kappa(A)+1}\right)^k \|x^{(0)} - x^*\|_{A^HA}.$$

Now, we can prove Theorem 2.1 as follows:

Proof of Theorem 2.1. Let $x^{(*,l)}$ be the exact solution of the system of linear equations (4). Then it satisfies

$$x^{(*,l)} = S_\nu^{-1} b - S_\nu^{-1} H_\nu x^{(l-1)}.$$

On the other hand, since x^* is the exact solution of the system of linear equations (1), it obeys

$$x^* = S_\nu^{-1} b - S_\nu^{-1} H_\nu x^*.$$

Let $\mu(S_\nu) = (\kappa(S_\nu) - 1)/(\kappa(S_\nu) + 1)$. Then according to Lemma A.2, we have

$$\begin{aligned}
\|x^{(l)} - x^{(*,l)}\|_{S_\nu^H S_\nu} &\le 2\mu(S_\nu)^{k_l} \|x^{(l-1)} - x^{(*,l)}\|_{S_\nu^H S_\nu} \\
&= 2\mu(S_\nu)^{k_l} \|x^{(l-1)} - (S_\nu^{-1} b - S_\nu^{-1} H_\nu x^{(l-1)})\|_{S_\nu^H S_\nu} \\
&= 2\mu(S_\nu)^{k_l} \|(I + S_\nu^{-1} H_\nu) x^{(l-1)} - (I + S_\nu^{-1} H_\nu) x^*\|_{S_\nu^H S_\nu} \\
&= 2\mu(S_\nu)^{k_l} \|(I + S_\nu^{-1} H_\nu)(x^{(l-1)} - x^*)\|_{S_\nu^H S_\nu} \\
&\le 2\mu(S_\nu)^{k_l} (1 + \|S_\nu^{-1} H_\nu\|_{S_\nu^H S_\nu}) \|x^{(l-1)} - x^*\|_{S_\nu^H S_\nu} \\
&= 2\mu(S_\nu)^{k_l} (1 + \varrho) \|x^{(l-1)} - x^*\|_{S_\nu^H S_\nu}.
\end{aligned}$$

Furthermore, we can obtain

$$\begin{aligned}
\|x^{(l)} - x^*\|_{S_\nu^H S_\nu} &= \|(x^{(l)} - x^{(*,l)}) - (x^{(*,l)} - x^*)\|_{S_\nu^H S_\nu} \\
&\le \|x^{(l)} - x^{(*,l)}\|_{S_\nu^H S_\nu} + \|x^{(*,l)} - x^*\|_{S_\nu^H S_\nu} \\
&\le 2\mu(S_\nu)^{k_l} (1 + \varrho) \|x^{(l-1)} - x^*\|_{S_\nu^H S_\nu} + \|S_\nu^{-1} H_\nu (x^{(l-1)} - x^*)\|_{S_\nu^H S_\nu} \\
&\le 2\mu(S_\nu)^{k_l} (1 + \varrho) \|x^{(l-1)} - x^*\|_{S_\nu^H S_\nu} + \varrho \|x^{(l-1)} - x^*\|_{S_\nu^H S_\nu} \\
&= (2\mu(S_\nu)^{k_l} (1 + \varrho) + \varrho) \|x^{(l-1)} - x^*\|_{S_\nu^H S_\nu} \\
&= \gamma^{(l)} \|x^{(l-1)} - x^*\|_{S_\nu^H S_\nu}.
\end{aligned}$$

This proves the validity of (a). We now turn to the proof of (b). Since

$$\|b - Ax^{(l)}\|_{S_\nu^H S_\nu} = \|A(x^{(l)} - x^*)\|_{S_\nu^H S_\nu} \le \|A\|_{S_\nu^H S_\nu} \|x^{(l)} - x^*\|_{S_\nu^H S_\nu},$$

making use of (a), we have

$$\begin{aligned}
\|b - Ax^{(l)}\|_{S_\nu^H S_\nu} &\le \gamma^{(l)} \|A\|_{S_\nu^H S_\nu} \|x^{(l-1)} - x^*\|_{S_\nu^H S_\nu} \\
&\le \gamma^{(l)} \|A\|_{S_\nu^H S_\nu} \|A^{-1}(b - Ax^{(l-1)})\|_{S_\nu^H S_\nu} \\
&\le \gamma^{(l)} \|A\|_{S_\nu^H S_\nu} \|A^{-1}\|_{S_\nu^H S_\nu} \|b - Ax^{(l-1)}\|_{S_\nu^H S_\nu}.
\end{aligned}$$

We easily obtain

$$\begin{aligned}
\|A\|_{S_\nu^H S_\nu} &= \|H_\nu + S_\nu\|_{S_\nu^H S_\nu} \\
&= \|S_\nu (S_\nu^{-1} H_\nu + I)\|_{S_\nu^H S_\nu} \\
&\le \|S_\nu\|_{S_\nu^H S_\nu} (1 + \|S_\nu^{-1} H_\nu\|_{S_\nu^H S_\nu}),
\end{aligned}$$

and since $\|S_\nu^{-1} H_\nu\|_{S_\nu^H S_\nu} < 1$, by Lemma A.1, we can obtain

$$\begin{aligned}
\|A^{-1}\|_{S_\nu^H S_\nu} &= \|(H_\nu + S_\nu)^{-1}\|_{S_\nu^H S_\nu} \\
&= \|(S_\nu^{-1} H_\nu + I)^{-1} S_\nu^{-1}\|_{S_\nu^H S_\nu} \\
&\le \|S_\nu^{-1}\|_{S_\nu^H S_\nu} \|(S_\nu^{-1} H_\nu + I)^{-1}\|_{S_\nu^H S_\nu} \\
&\le \frac{\|S_\nu^{-1}\|_{S_\nu^H S_\nu}}{1 - \|S_\nu^{-1} H_\nu\|_{S_\nu^H S_\nu}}.
\end{aligned}$$

On the other hand, we have $\|S_\nu\|_{S_\nu^H S_\nu} = \|S_\nu\|_2$ and $\|S_\nu^{-1}\|_{S_\nu^H S_\nu} = \|S_\nu^{-1}\|_2$. Therefore, we have:

$$\|A\|_{S_\nu^H S_\nu} \le \|S_\nu\|_2 (1 + \|S_\nu^{-1} H_\nu\|_{S_\nu^H S_\nu}) \quad \text{and} \quad \|A^{-1}\|_{S_\nu^H S_\nu} \le \frac{\|S_\nu^{-1}\|_2}{1 - \|S_\nu^{-1} H_\nu\|_{S_\nu^H S_\nu}}.$$

Therefore, it follows that

$$\begin{aligned}\|b-Ax^{(l)}\|_{S_\nu^H S_\nu} &\le \gamma^{(l)}\frac{\|S_\nu\|_2(1+\|S_\nu^{-1}H_\nu\|_{S_\nu^H S_\nu})\|S_\nu^{-1}\|_2}{1-\|S_\nu^{-1}H_\nu\|_{S_\nu^H S_\nu}}\|b-Ax^{(l-1)}\|_{S_\nu^H S_\nu}\\ &= \gamma^{(l)}\kappa(S_\nu)\frac{1+\|S_\nu^{-1}H_\nu\|_{S_\nu^H S_\nu}}{1-\|S_\nu^{-1}H_\nu\|_{S_\nu^H S_\nu}}\|b-Ax^{(l-1)}\|_{S_\nu^H S_\nu}\\ &= \widetilde{\gamma}^{(l)}\|b-Ax^{(l-1)}\|_{S_\nu^H S_\nu}.\end{aligned}$$

This shows the validity of (b).

It is obvious that, for $\gamma \in (\varrho, \varrho_1)$ with $\varrho_1 = \min\{1, 2+3\varrho\}$, $\gamma^{(l)} \le \gamma$ $(l = 1, 2, \ldots)$ holds under condition

$$k_l \ge \frac{\ln\left(\frac{\gamma-\varrho}{2(1+\varrho)}\right)}{\ln\left(\frac{\kappa(S_\nu)-1}{\kappa(S_\nu)+1}\right)}, \quad l = 1, 2, \ldots,$$

and the estimates

$$\begin{aligned}\|x^{(l)}-x^*\|_{S_\nu^H S_\nu} &\le \gamma^{(l)}\|x^{(l-1)}-x^*\|_{S_\nu^H S_\nu} \le \Pi_{k=0}^{l}\gamma^{(k)}\|x^{(0)}-x^*\|_{S_\nu^H S_\nu}\\ &\le \gamma^{l+1}\|x^{(0)}-x^*\|_{S_\nu^H S_\nu} \to 0, \quad l \to \infty,\end{aligned}$$

hold in accordance with (a). Therefore, the sequence $\{x^{(l)}\}_{l=0}^{\infty}$ converges to the solution x^* of the system of linear equations (1).

In addition, for $\varrho \in (0, r)$, where r is the positive root of quadratic equation $\kappa(S_\nu)\varrho^2 + (\kappa(S_\nu)+1)\varrho - 1 = 0$ and $0 < r < 1$, we have $0 < (1+\varrho)\varrho\kappa(S_\nu)/(1-\varrho) < 1$. So, for

$$\widetilde{\gamma} \in \left(\frac{(1+\varrho)\varrho}{(1-\varrho)}\kappa(S_\nu), 1\right),$$

$\widetilde{\gamma}^{(l)} \le \widetilde{\gamma}$ $(l = 1, 2, 3, \ldots)$ holds under condition

$$k_l \ge \frac{\ln\left(\frac{(1-\varrho)\widetilde{\gamma}-\varrho(1+\varrho)\kappa(S_\nu)}{2(1+\varrho)^2\kappa(S_\nu)}\right)}{\ln\left(\frac{\kappa(S_\nu)-1}{\kappa(S_\nu)+1}\right)}, \quad l = 1, 2, 3, \ldots,$$

and the estimates

$$\begin{aligned}\|b-Ax^{(l)}\|_{S_\nu^H S_\nu} &\le \widetilde{\gamma}^{(l)}\|b-Ax^{(l-1)}\|_{S_\nu^H S_\nu} \le \Pi_{k=0}^{l}\widetilde{\gamma}^{(k)}\|b-Ax^{(0)}\|_{S_\nu^H S_\nu}\\ &\le \widetilde{\gamma}^{l+1}\|b-Ax^{(0)}\|_{S_\nu^H S_\nu} \to 0, \quad l \to \infty,\end{aligned}$$

hold in accordance with (b). □

References

[1] 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.
[2] O. Axelsson, Z.-Z. Bai, S.-X. Qiu, A class of nested iteration schemes for linear systems with a coefficient matrix with a dominant positive definite symmetric part, Numer. Algorithms 35 (2004) 351–372.
[3] Z.-Z. Bai, Splitting iteration methods for non-Hermitian positive definite systems of linear equations, Hokkaido Math. J. 36 (2007) 801–814.
[4] 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.
[5] H.C. Elman, G.H. Golub, Iterative methods for cyclically reduced non-self-adjoint linear systems, Math. Comp. 54 (1990) 671–700.
[6] H.C. Elman, G.H. Golub, Iterative methods for cyclically reduced non-self-adjoint linear systems II, Math. Comp. 56 (1991) 215–242.
[7] Y. Saad, Iterative Methods for Sparse Linear Systems, second ed., SIAM, Philadelphia, 2003.
[8] G.H. Golub, D. Vandrestraeten, On the preconditioning of matrices with skew-symmetric splittings, Numer. Algorithms 25 (2000) 223–239.
[9] P.C. Hansen, J.G. Nagy, D.P. O'Leary, Deblurring Images: Matrices, Spectra, and Filtering, SIAM, Philadelphia, 2006.
[10] A.K. Jain, Fundamentals of Digital Image Processing, Prentice-Hall, Engelwood Cliffs, NJ, 1989.
[11] G.H. Golub, P.C. Hansen, D.P. O'Leary, Tikhonov regularization and total least squares, SIAM J. Matrix Anal. Appl. 21 (1) (1999) 185–194.
[12] P.C. Hansen, Regularization tools version 4.1 for matlab 7.3, Numer. Algorithms 46 (2007) 189–194.
[13] X.-G. Lv, T.-Z. Huang, Z.-B. Xu, X.-L. Zhao, A special Hermitian and skew-Hermitian splitting method for image restoration, Appl. Math. Model. 37 (2013) 1069–1082.
[14] C.B. Show Jr., Improvments of the resolution of an instrument by numerical solution of an integral equation, J. Math. Anal. Appl. 37 (1972) 83–112.
[15] J. Stoer, R. Bulirsch, Introduction to Numerical Analysis, third ed., Springer, 2002.