RESEARCH ARTICLE
doi: 10.2306/scienceasia1513-1874.2024.049

ScienceAsia 50 (5): 2024: ID 2024049: **1–10**

An inexact Krylov subspace method for large generalized Hankel eigenproblems

Zi-Yan Huang, Ting-Ting Feng*

Department of Mathematics, School of Sciences, Hangzhou Dianzi University, Hangzhou 310018 China

*Corresponding author, e-mail: ttfeng@hdu.edu.cn

Received 20 Jun 2023, Accepted 26 Jan 2024
Available online 26 May 2024

ABSTRACT: Krylov subspace method is an effective method for large-scale eigenproblems. The shift-and-invert Arnoldi method is employed to compute a few eigenpairs of a large Hankel matrix pencil. However, a crucial step in the process is computing products between the inversion of a Hankel matrix and vectors. The inversion of the Hankel matrix can be obtained by solving two Hankel systems. By establishing a relationship between the errors of systems and the residuals of the Hankel eigenproblem, we provide a practical stopping criterion for solving Hankel systems and propose an inexact shift-and-invert Arnoldi method for the generalized Hankel eigenproblem. Numerical experiments are presented to demonstrate the efficiency of the new algorithm and our theoretical results.

KEYWORDS: Hankel matrix, generalized eigenproblem, shift-and-invert Arnoldi method, Hankel inverse formula

MSC2020: 65F15 65F10

INTRODUCTION

An $n \times n$ matrix H is referred to as a Hankel matrix if it satisfies

$$H = (h_{i+k})_{i,k=0}^{n-1} = \begin{bmatrix} h_0 & h_1 & \cdots & h_{n-2} & h_{n-1} \\ h_1 & h_2 & \cdots & h_{n-1} & h_n \\ \vdots & \iddots & \iddots & h_n & \vdots \\ h_{n-2} & \iddots & \iddots & \vdots & h_{2n-3} \\ h_{n-1} & h_n & \dots & h_{2n-3} & h_{2n-2} \end{bmatrix},$$

meaning that H is constant along its anti-diagonals. Therefore, it only needs to store the first column and last row elements to represent a Hankel matrix. Hankel matrices and operators occur in a number of applications in mathematics and engineering, including approximation theory, linear system theory, prediction theory, and control theory [1]. Fast algorithms for Hankel matrices have been under in-depth study over the last decades.

The Hankel eigenproblem and generalized Hankel eigenproblem arise in many applications, such as the reconstruction of the shape of a polygon from its moments, the determination of the abscissas of quadrature formulas, and the poles of Padé approximants [2]. However, there have been relatively few works focusing on the Hankel eigenproblem. A fast eigenvalue algorithm for Hankel matrices was proposed [3] based on the Lanczos-type tridiagonalization and QR-type diagonalization methods. Some studies [4–7] have focused specifically on the smallest eigenvalue of large scale Hankel matrices. The sensitivity of the nonlinear application [2] mapping the vector of Hankel entries to its generalized eigenvalues was studied. The parallel algorithm and asymptotic behavior of the smallest eigenvalue of a Hankel matrix were studied [4–6].

The Krylov subspace method is an efficient approach for computing the smallest eigenpair or a few extreme eigenpairs of large-scale matrices [8]. This projection-based method can be achieved using the Lanczos process for symmetric matrices or the Arnoldi process for nonsymmetric matrices, with both procedures requiring matrix-vector multiplications [8]. The shift-and-invert technique, with either the Arnoldi or Lanczos method, has been popularly used for computing a number of eigenvalues close to a given shift and the associated eigenvectors of a large matrix or matrix pair [8]. By using the shift-and-invert technique, the multiplication of a inverse of a matrix and vector is impotant.

Recently, the shift-and-invert Arnoldi or Lanczos method has been used in designing fast algorithms for the generalized Toeplitz eigenproblem [9], and Toeplitz matrix exponential [10, 11]. Toeplitz matrices have various applications [12–15], due to the special structure of Toeplitz matrices, there are many fast algorithms for solving Toeplitz matrix problems [16–19] and various formula for the inversion of Toeplitz matrix [20–22], the products of the inverse of a Toeplitz matrix and a vector can be implemented using several FFTs [10, 11, 21]. For a Hankel matrix, the inverse can be obtained by solving two large Hankel linear systems, and the matrix-vector products in the shift-and-invert Arnoldi method can also be realized efficiently by using FFTs. This motivates us to consider how to solve large Hankel generalized eigenproblems efficiently.

In this paper, we focus on computing a few eigenpairs of the following large Hankel generalized eigenproblem:

$$A\mathbf{x} = \lambda B\mathbf{x}, \tag{1}$$

where A and B are large-scale Hankel matrices and the

matrix pencil (A,B) is regular [8, 23]. The shift-and-invert Arnoldi method is used, in which multiplications of the inverse of a Hankel matrix and vectors are essential. For the inversion of a Hankel matrix, we have to solve two large Hankel systems in advance. However, if the accuracy is too high, the cost of solving such systems becomes prohibitive and wasteful. Thus, it is necessary to explore an "inexact" shift-and-invert Arnoldi method for solving large Hankel generalized eigenproblems.

The remainder of this paper is organized as follows. Firstly, we briefly introduce the shift-and-invert Arnoldi method. Secondly, we analyze the relationship between the errors of solving Hankel systems and the residual of the generalized Hankel eigenproblem, and propose an inexact shift-and-invert Arnoldi method for solving the generalized Hankel eigenproblem. Finally, numerical examples are given to verify the efficiency of our theoretical results.

SHIFT-AND-INVERT ARNOLDI METHOD FOR GENERALIZED HANKEL EIGENPROBLEMS

One of the most effective methods for solving large scale eigenproblems is the shift-and-invert Arnoldi method [8]. Given a shift $\sigma \in \mathbb{C}$, we can derive from (1) that

$$(A-\sigma B)\mathbf{x} = (\lambda-\sigma)B\mathbf{x}.$$

If $A-\sigma B$ is invertible, then the generalized eigenproblem can be reformulated as the following standard eigenproblem:

$$(A-\sigma B)^{-1}B\mathbf{x} = \mu\mathbf{x},$$

where $\mu = 1/(\lambda-\sigma)$. The shift-and-invert technique is to iterate with the matrix $(A-\sigma B)^{-1}B$, and one should only deal with the matrix $A-\sigma B$ once for a given shift, or a few times when σ is changed. The number of iterations required with $(A-\sigma B)^{-1}B$ can be significantly smaller than that needed to solve the original problem (1) directly [8].

The shift-and-invert Arnoldi method is commonly used for computing several eigenvalues closest to a given shift σ and the associated eigenvectors. Given a unit vector $\mathbf{v}_1$, the m-step shift-and-invert Arnoldi process, formulated in exact arithmetic, can be expressed as [8]

$$(A-\sigma B)^{-1}BV_m = V_m H_m + h_{m+1,m}\mathbf{v}_{m+1}\mathbf{e}_m^\top, \qquad (2)$$

where $V_m = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_m]$ is an orthonormal basis for the Krylov subspace $\mathscr{K}_m\big((A-\sigma B)^{-1}B, \mathbf{v}_1\big)$, $\mathbf{e}_m$ is the m-th column of identity matrix of order m, and H_m is an upper Hessenberg matrix of order m. Let $(\widetilde{\mu}, \widetilde{\mathbf{w}})$ with $\|\widetilde{\mathbf{w}}\|_2 = 1$ be an eigenpair of H_m. The shift-and-invert Arnoldi method uses $(\widetilde{\lambda} = 1/\widetilde{\mu}+\sigma,\ \widetilde{\mathbf{x}} = V_m\widetilde{\mathbf{w}})$ as an approximation to $(\lambda, \mathbf{x})$. If we denote the residual corresponding to the Ritz pair $(\widetilde{\lambda}, \widetilde{\mathbf{x}})$ by $\widehat{\mathbf{r}} = A\widetilde{\mathbf{x}} - \widetilde{\lambda}B\widetilde{\mathbf{x}}$, then we have that [24]

$$\frac{\|\widehat{\mathbf{r}}\|_2}{\|A-\sigma B\|_2} \leqslant h_{m+1,m}|\widetilde{\lambda}-\sigma| \cdot |\mathbf{e}_m^\top \widetilde{\mathbf{w}}|, \qquad (3)$$

which can be used as a cheap stopping criterion for the shift-and-invert Arnoldi method.

We notice that, for Hankel matrices A and B, the matrix $A-\sigma B$ is also a Hankel and Hermitian matrix. However, the matrix $(A-\sigma B)^{-1}B$ may be non-Hermitian. Therefore, the shift-and-invert Arnoldi method is utilized in the following sections.

AN INEXACT SHIFT-AND-INVERT ARNOLDI METHOD FOR GENERALIZED HANKEL EIGENPROBLEMS

The computation of m matrix-vector products $(A-\sigma B)^{-1}B\mathbf{v}_j$, where $j = 1, 2, \ldots, m$, are required for the shift-and-invert Arnoldi process. One option is to compute the inverse $(A-\sigma B)^{-1}$ using LU decomposition [23], but this can be costly, especially for large dense matrices. As σ is a given shift, we are interested in computing $(A-\sigma B)^{-1}$ once for all.

Fortunately, for Hankel matrices A and B, $A-\sigma B$ is also a Hankel matrix. Noticing that $H = JT$, where T is a real Toeplitz matrix, H is a real Hankel matrix, and J is a square matrix of order n with ones on the skew diagonal and zeros elsewhere, which is given by

$$J = \begin{bmatrix} 0 & 0 & \ldots & 1 \\ \vdots & \vdots & \iddots & \vdots \\ 0 & 1 & \ldots & 0 \\ 1 & 0 & \ldots & 0 \end{bmatrix}.$$

The inverse of a nonsingular Toeplitz matrix can be represented as the sum of products of circulant and skew-circulant matrices [20]. Therefore, for a nonsingular Hankel matrix $H = (h_{i+k})(i,k = 0, \ldots, n-1)$, there is a formula for computing its inverse. Specifically, the inverse of a Hankel matrix H can be represented by the solutions of two Hankel systems, which allows us to avoid explicitly storing the $(A-\sigma B)^{-1}$. Let $\mathbf{x} = [x_0, x_1, \ldots, x_{n-1}]^\top$ and $\mathbf{y} = [y_0, y_1, \ldots, y_{n-1}]^\top$ be the solutions of the following two Hankel systems:

$$H\mathbf{x} = \mathbf{e}_n \quad \text{and} \quad H\mathbf{y} = \mathbf{e}_1, \qquad (4)$$

where $\mathbf{e}_1$ and $\mathbf{e}_n$ are the first and the last column of the identity matrix of order n, respectively. The inverse of a Hankel matrix can be accordingly represented as follows [20, 21]:

$$H^{-1} = \frac{1}{2x_0}(S_1C_1 - S_2C_2)J, \qquad (5)$$

where

$$S_1 = \begin{bmatrix} x_0 & -x_{n-1} & \cdots & -x_2 & -x_1 \\ x_1 & x_0 & -x_{n-1} & \ddots & -x_2 \\ \vdots & x_1 & \ddots & \ddots & \vdots \\ x_{n-2} & \ddots & \ddots & x_0 & -x_{n-1} \\ x_{n-1} & x_{n-2} & \cdots & x_1 & x_0 \end{bmatrix}$$

$$C_1 = \begin{bmatrix} y_{n-1} & y_{n-2} & \cdots & y_1 & y_0 \\ y_0 & y_{n-1} & y_{n-2} & \ddots & y_1 \\ \vdots & y_0 & \ddots & \ddots & \vdots \\ y_{n-3} & \ddots & \ddots & y_{n-1} & y_{n-2} \\ y_{n-2} & y_{n-3} & \cdots & y_0 & y_{n-1} \end{bmatrix}$$

$$S_2 = \begin{bmatrix} -y_{n-1} & -y_{n-2} & \cdots & -y_1 & -y_0 \\ y_0 & -y_{n-1} & -y_{n-2} & \ddots & -y_1 \\ \vdots & y_0 & \ddots & \ddots & \vdots \\ y_{n-3} & \ddots & \ddots & -y_{n-1} & -y_{n-2} \\ y_{n-2} & y_{n-3} & \cdots & y_0 & -y_{n-1} \end{bmatrix}$$

$$C_2 = \begin{bmatrix} x_0 & x_{n-1} & \cdots & x_2 & x_1 \\ x_1 & x_0 & x_{n-1} & \ddots & x_2 \\ \vdots & x_1 & \ddots & \ddots & \vdots \\ x_{n-2} & \ddots & \ddots & x_0 & x_{n-1} \\ x_{n-1} & x_{n-2} & \cdots & x_1 & x_0 \end{bmatrix}.$$

S_1, S_2 are skew-circulant matrices with $\mathbf{x} = [x_0, x_1, \ldots, x_{n-1}]^\top$ and $\hat{\mathbf{y}} = [-y_{n-1}, y_0, y_1, \ldots, y_{n-2}]^\top$ as their first column, respectively; C_1, C_2 are circulant matrices with $\check{\mathbf{y}} = [y_{n-1}, y_0, y_1, \ldots, y_{n-2}]^\top$ and $\mathbf{x}$ as their first column, respectively.

The computational of circulant matrix-vector products can be alleviated through using Fast Fourier Transforms (FFTs). Circulant matrices can be diagonalized by the Fourier matrix F_n, so the formula (5) can be rewritten as [21]

$$H^{-1} = \frac{1}{2x_0}(S_1 F_n^* \Lambda_{C_1} - S_2 F_n^* \Lambda_{C_2}) F_n J, \qquad (6)$$

where the (j,k)-entry of Fourier matrix F_n is $F_{j,k} = \frac{1}{\sqrt{n}} \mathrm{e}^{\frac{2\pi i (j-1)(k-1)}{n}}$, $1 \leqslant j,k \leqslant n$ and $\Lambda_{C_1}, \Lambda_{C_2}$ are diagonal matries comprised of the eigenvalues of C_1 and C_2, respectively. As for the multiplication of real skew-circulant matrix and vector, there is an order-reduction algorithm. The n-dimensional real skew-circulant matrix is splitted into $n/2$-dimensional submatrices, constructed an imaginary circulant matrix and used the diagonalization scheme to obtain the skew-circulant matrix-vector products. Based on the above, the multiplication of the inverse of a Hankel matrix and a vector requires three FFTs and two IFFTs of length n, as well as four FFTs and two IFFTs of length $n/2$. For further information, see [21, 25] and references therein.

In the shift-and-invert Arnoldi method, one has to compute the products of the inverse of a Hankel matrix and vectors. The inverse of a Hankel matrix can be obtained by solving two large Hankel systems. If the shift-and-invert Arnoldi algorithm requires the exact solution of two large-scale Hankel linear systems in (4), solving the Hankel systems as accurate as possible is preferred. The cost of solving the linear systems, particularly for some ill-conditioned situations, will be prohibitive if the needed precision is too high. Therefore, it is interesting to consider how to solve the Hankel systems in low accuracy to reduce the cost [10, 26]. This approach can be viewed as an "inexact" inverse technique since the Hankel linear systems are solved once and for all.

Let $\widetilde{\mathbf{x}} = [\widetilde{x}_0, \widetilde{x}_1, \ldots, \widetilde{x}_{n-1}]^\top$ and $\widetilde{\mathbf{y}} = [\widetilde{y}_0, \widetilde{y}_1, \ldots, \widetilde{y}_{n-1}]^\top$ be the numerical solutions of the two Hankel systems $H\mathbf{x} = \mathbf{e}_1$ and $H\mathbf{y} = \mathbf{e}_n$, respectively. Correspondingly,

$$\widetilde{H}^{-1} = \frac{1}{2\widetilde{x}_0}(\widetilde{S}_1 \widetilde{C}_1 - \widetilde{S}_2 \widetilde{C}_2) J, \qquad (7)$$

where

$$\widetilde{S}_1 = \begin{bmatrix} \widetilde{x}_0 & -\widetilde{x}_{n-1} & \cdots & -\widetilde{x}_2 & -\widetilde{x}_1 \\ \widetilde{x}_1 & \widetilde{x}_0 & -\widetilde{x}_{n-1} & \ddots & -\widetilde{x}_2 \\ \vdots & \widetilde{x}_1 & \ddots & \ddots & \vdots \\ \widetilde{x}_{n-2} & \ddots & \ddots & \widetilde{x}_0 & -\widetilde{x}_{n-1} \\ \widetilde{x}_{n-1} & \widetilde{x}_{n-2} & \cdots & \widetilde{x}_1 & \widetilde{x}_0 \end{bmatrix}$$

$$\widetilde{C}_1 = \begin{bmatrix} \widetilde{y}_{n-1} & \widetilde{y}_{n-2} & \cdots & \widetilde{y}_1 & \widetilde{y}_0 \\ \widetilde{y}_0 & \widetilde{y}_{n-1} & \widetilde{y}_{n-2} & \ddots & \widetilde{y}_1 \\ \vdots & \widetilde{y}_0 & \ddots & \ddots & \vdots \\ \widetilde{y}_{n-3} & \ddots & \ddots & \widetilde{y}_{n-1} & \widetilde{y}_{n-2} \\ \widetilde{y}_{n-2} & \widetilde{y}_{n-3} & \cdots & \widetilde{y}_0 & \widetilde{y}_{n-1} \end{bmatrix}$$

$$\widetilde{S}_2 = \begin{bmatrix} -\widetilde{y}_{n-1} & -\widetilde{y}_{n-2} & \cdots & -\widetilde{y}_1 & -\widetilde{y}_0 \\ \widetilde{y}_0 & -\widetilde{y}_{n-1} & -\widetilde{y}_{n-2} & \ddots & -\widetilde{y}_1 \\ \vdots & \widetilde{y}_0 & \ddots & \ddots & \vdots \\ \widetilde{y}_{n-3} & \ddots & \ddots & -\widetilde{y}_{n-1} & -\widetilde{y}_{n-2} \\ \widetilde{y}_{n-2} & \widetilde{y}_{n-3} & \cdots & \widetilde{y}_0 & -\widetilde{y}_{n-1} \end{bmatrix}$$

$$\widetilde{C}_2 = \begin{bmatrix} \widetilde{x}_0 & \widetilde{x}_{n-1} & \cdots & \widetilde{x}_2 & \widetilde{x}_1 \\ \widetilde{x}_1 & \widetilde{x}_0 & \widetilde{x}_{n-1} & \ddots & \widetilde{x}_2 \\ \vdots & \widetilde{x}_1 & \ddots & \ddots & \vdots \\ \widetilde{x}_{n-2} & \ddots & \ddots & \widetilde{x}_0 & \widetilde{x}_{n-1} \\ \widetilde{x}_{n-1} & \widetilde{x}_{n-2} & \cdots & \widetilde{x}_1 & \widetilde{x}_0 \end{bmatrix}$$

and $\widetilde{H}^{-1}$ can be considered as a perturbation to the Hankel matrix inverse H^{-1}. Therefore, $\widetilde{H}^{-1}$ can be represented as

$$\widetilde{H}^{-1}=\frac{1}{2\widetilde{x}_0}(\widetilde{S}_1F_n^*\widetilde{\Lambda}_{C_1}-\widetilde{S}_2F_n^*\widetilde{\Lambda}_{C_2})F_nJ.$$

In the practical calculation, we compute $H^{-1}\mathbf{v}$ by using (6) and realised the matrix-vector products through some FFTs. Thus it is more convincible to analyze the stability of formula (6). Inspired by the proof [27], we first give two lemmas for the norm of circulant and skew-circulant matrices.

Lemma 1 *Let C be a circulant matrix of order n. The first column of C is represented as $\mathbf{c}=[c_0,c_1,\ldots,c_{n-1}]^\top$, then*

$$\|C\|_2\leqslant\|\mathbf{c}\|_1 \tag{8}$$

Proof: The eigenvalues of matrix C are as follows. For $j=0,1,\ldots,n-1$, there holds

$$\lambda_j(C)=\sum_{k=0}^{n-1}c_k\left(\omega^j\right)^k,$$

where $\omega=\cos\frac{2\pi}{n}+\mathrm{i}\sin\frac{2\pi}{n}$ ($\mathrm{i}=\sqrt{-1}$). It is evident that the spectral norm of matrix C can be computed by

$$\begin{aligned}\|C\|_2&=\max_{0\leqslant j\leqslant n-1}\left|\lambda_j(C)\right|=\max_{0\leqslant j\leqslant n-1}\left|\sum_{k=0}^{n-1}c_k\left(\omega^j\right)^k\right|\\&\leqslant\max_{0\leqslant j\leqslant n-1}\left\{\sum_{k=0}^{n-1}|c_k|\cdot\left|\left(\omega^j\right)^k\right|\right\}\\&=\sum_{i=0}^{n-1}|c_k|=\|\mathrm{c}\|_1.\end{aligned}$$

□

Similarly, we can obtain the following lemma for skew-circulant matrix.

Lemma 2 *Let S be a skew-circulant matrix of order n. The first column of S is represented as $\mathbf{s}=[s_0,\ s_1,\ \ldots,\ s_{n-1}]^\top$, then*

$$\|S\|_2\leqslant\|\mathbf{s}\|_1 \tag{9}$$

Based on Lemma 1 and Lemma 2, we show the stability analysis for the inverse formula (6) of a Hankel matrix.

Theorem 1 *Let $\varepsilon>0$. If*

$$\frac{\|\widetilde{\mathbf{x}}-\mathbf{x}\|_1}{\|\mathbf{x}\|_1}\leqslant\varepsilon\quad\text{and}\quad\frac{\|\widetilde{\mathbf{y}}-\mathbf{y}\|_1}{\|\mathbf{y}\|_1}\leqslant\varepsilon, \tag{10}$$

then

$$\begin{aligned}&\|H^{-1}-\widetilde{H}^{-1}\|_2\\&\quad\leqslant\left|\frac{1}{x_0}\right|[\varepsilon+(\varepsilon+(1+\varepsilon)\tilde{\varepsilon})(1+\varepsilon)]\|\boldsymbol{x}\|_1\|\boldsymbol{y}\|_1.\end{aligned} \tag{11}$$

Proof:

$$\begin{aligned}\left\|H^{-1}-\widetilde{H}^{-1}\right\|_2&=\left\|\frac{1}{2x_0}(S_1F_n^*\Lambda_{C_1}F_n-S_2F_n^*\Lambda_{C_2}F_n)J\right.\\&\quad\left.-\frac{1}{2\widetilde{x}_0}(\widetilde{S}_1F_n^*\widetilde{\Lambda}_{C_1}F_n-\widetilde{S}_2F_n^*\widetilde{\Lambda}_{C_2}F_n)J\right\|_2\\&\leqslant\left\|\frac{1}{2x_0}(S_1F_n^*\Lambda_{C_1}F_n-S_2F_n^*\Lambda_{C_2}F_n)\right.\\&\quad\left.-\frac{1}{2\widetilde{x}_0}(\widetilde{S}_1F_n^*\widetilde{\Lambda}_{C_1}F_n-\widetilde{S}_2F_n^*\widetilde{\Lambda}_{C_2}F_n)\right\|_2\|J\|_2\\&=\left\|\frac{1}{2x_0}(S_1F_n^*\Lambda_{C_1}F_n-S_2F_n^*\Lambda_{C_2}F_n)\right.\\&\quad\left.-\frac{1}{2\widetilde{x}_0}(\widetilde{S}_1F_n^*\widetilde{\Lambda}_{C_1}F_n-\widetilde{S}_2F_n^*\widetilde{\Lambda}_{C_2}F_n)\right\|_2\\&=\left\|\left(\frac{1}{2x_0}S_1\right)F_n^*\Lambda_{C_1}F_n-S_2\left(\frac{1}{2x_0}F_n^*\Lambda_{C_2}F_n\right)\right.\\&\quad\left.-\left(\frac{1}{2\tilde{x}_0}\tilde{S}_1\right)F_n^*\widetilde{\Lambda}_{C_1}F_n+\tilde{S}_2\left(\frac{1}{2\tilde{x}_0}F_n^*\widetilde{\Lambda}_{C_2}F_n\right)\right\|_2\\&\leqslant\left\|\left(\frac{1}{2x_0}S_1\right)F_n^*\Lambda_{C_1}F_n-\left(\frac{1}{2\tilde{x}_0}\tilde{S}_1\right)F_n^*\widetilde{\Lambda}_{C_1}F_n\right\|_2\\&\quad+\left\|\tilde{S}_2\left(\frac{1}{2\tilde{x}_0}F_n^*\widetilde{\Lambda}_{C_2}F_n\right)-S_2\left(\frac{1}{2x_0}F_n^*\Lambda_{C_2}F_n\right)\right\|_2.\end{aligned} \tag{12}$$

Moreover, we deduce that

$$\begin{aligned}&\left\|\left(\frac{1}{2x_0}S_1\right)F_n^*\Lambda_{C_1}F_n-\left(\frac{1}{2\tilde{x}_0}\tilde{S}_1\right)F_n^*\widetilde{\Lambda}_{C_1}F_n\right\|_2\\&\quad=\left\|\left(\frac{1}{2x_0}S_1\right)F_n^*\Lambda_{C_1}F_n-\left(\frac{1}{2x_0}S_1\right)F_n^*\widetilde{\Lambda}_{C_1}F_n\right.\\&\qquad\left.+\left(\frac{1}{2x_0}S_1\right)F_n^*\widetilde{\Lambda}_{C_1}F_n-\left(\frac{1}{2\tilde{x}_0}\tilde{S}_1\right)F_n^*\widetilde{\Lambda}_{C_1}F_n\right\|_2\\&\quad=\left\|\left(\frac{1}{2x_0}S_1\right)\left(F_n^*\Lambda_{C_1}F_n-F_n^*\widetilde{\Lambda}_{C_1}F_n\right)\right.\\&\qquad\left.+\left(\frac{1}{2x_0}S_1-\frac{1}{2\tilde{x}_0}\tilde{S}_1\right)F_n^*\widetilde{\Lambda}_{C_1}F_n\right\|_2\\&\quad\leqslant\left|\frac{1}{2x_0}\right|\|S_1\|_2\left\|F_n^*\Lambda_{C_1}F_n-F_n^*\widetilde{\Lambda}_{C_1}F_n\right\|_2\\&\qquad+\left\|\frac{1}{2x_0}S_1-\frac{1}{2\tilde{x}_0}\tilde{S}_1\right\|_2\left\|F_n^*\widetilde{\Lambda}_{C_1}F_n\right\|_2\\&\quad\leqslant\left|\frac{1}{2x_0}\right|\|S_1\|_2\left\|C_1-\tilde{C}_1\right\|_2+\left\|\frac{1}{2x_0}S_1-\frac{1}{2\tilde{x}_0}\tilde{S}_1\right\|_2\left\|\tilde{C}_1\right\|_2.\end{aligned} \tag{13}$$

We obtain from (9), (10) that

$$\begin{aligned}\left\|\frac{1}{2x_0}S_1-\frac{1}{2\tilde{x}_0}\tilde{S}_1\right\|_2&\leqslant\left\|\frac{1}{2x_0}\boldsymbol{x}-\frac{1}{2\tilde{x}_0}\tilde{\boldsymbol{x}}\right\|_1\\&=\left|\frac{1}{2x_0}\right|\left\|\boldsymbol{x}-\tilde{\boldsymbol{x}}+\left(1-\frac{x_0}{\tilde{x}_0}\right)\tilde{\boldsymbol{x}}\right\|_1\\&\leqslant\left|\frac{1}{2x_0}\right|(\|\boldsymbol{x}-\tilde{\boldsymbol{x}}\|_1+\tilde{\varepsilon}\|\tilde{\boldsymbol{x}}\|_1)\\&\leqslant\left|\frac{1}{2x_0}\right|[\varepsilon+(1+\varepsilon)\tilde{\varepsilon}]\cdot\|\boldsymbol{x}\|_1,\end{aligned} \tag{14}$$

where $\tilde{\varepsilon} = \frac{|1/x_0 - 1/\tilde{x}_0|}{1/x_0}$ is the relative error of $1/x_0$. Furthermore, we notice from (8), (9) and (10) that

$$\|S_1\|_2 \leqslant \|\mathbf{x}\|_1, \quad \|\widetilde{C}_1\|_2 \leqslant \|\widetilde{\mathbf{y}}\|_1 \leqslant (1+\varepsilon)\|\mathbf{y}\|_1, \quad (15)$$

and

$$\|C_1 - \widetilde{C}_1\|_2 \leqslant \|\mathbf{y} - \widetilde{\mathbf{y}}\|_1 \leqslant \varepsilon\|\mathbf{y}\|_1. \quad (16)$$

From (13)–(16), we can get

$$\begin{aligned}
&\left\| \left(\frac{1}{2x_0} S_1\right) F_n^* \Lambda_{C_1} F_n - \left(\frac{1}{2\tilde{x}_0} \tilde{S}_1\right) F_n^* \widetilde{\Lambda}_{C_1} F_n \right\|_2 \\
&\quad \leqslant \left|\frac{1}{2x_0}\right| \|\boldsymbol{x}\|_1 \|\boldsymbol{y}\|_1 \cdot \varepsilon \\
&\qquad + \left|\frac{1}{2x_0}\right| \cdot [\varepsilon + (1+\varepsilon)\tilde{\varepsilon}] \cdot \|\boldsymbol{x}\|_1 \cdot \|\boldsymbol{y}\|_1 \cdot (1+\varepsilon) \\
&\quad \leqslant \left|\frac{1}{2x_0}\right| [\varepsilon + (\varepsilon + (1+\varepsilon)\tilde{\varepsilon})(1+\varepsilon)] \cdot \|\boldsymbol{x}\|_1 \cdot \|\boldsymbol{y}\|_1. \quad (17)
\end{aligned}$$

Using the same trick, for the second part of the right-hand side of (12), we can prove that

$$\begin{aligned}
&\left\| \tilde{S}_2 \left(\frac{1}{2\tilde{x}_0} F_n^* \widetilde{\Lambda}_{C_2} F_n\right) - S_2 \left(\frac{1}{2x_0} F_n^* \Lambda_{C_2} F_n\right) \right\|_2 \\
&\quad \leqslant \left|\frac{1}{2x_0}\right| [\varepsilon + (\varepsilon + (1+\varepsilon)\tilde{\varepsilon})(1+\varepsilon)] \cdot \|\boldsymbol{x}\|_1 \cdot \|\boldsymbol{y}\|_1, \quad (18)
\end{aligned}$$

and (11) is obtained by combining (12), (17), and (18). □

Denote $H = A - \sigma B$. When the Hankel systems are solved approximately, the errors for the matrix-vector products can be represented as $\mathbf{f}_j = \widetilde{H}^{-1} B\mathbf{v}_j - H^{-1} B\mathbf{v}_j$, $j = 1, 2, \ldots, m$. If we denote $F_m = [\mathbf{f}_1, \mathbf{f}_2, \cdots, \mathbf{f}_m]$, we have the following relation for the m-step "inexact" shift-and-invert Arnoldi procedure:

$$\begin{aligned}
(A-\sigma B)^{-1} B V_m + F_m &= \big((A-\sigma B)^{-1} B + E\big) V_m \\
&= V_m H_m + h_{m+1,m} \mathbf{v}_{m+1} \mathbf{e}_m^\top,
\end{aligned}$$

where $V_m = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_m]$ is an $n \times m$ orthonormal matrix, $E = F_m V_m^\top$, and H_m is an upper Hessenberg matrix of order m. It should be noted that V_m and H_m are different from those in (2).

Let $(\widetilde{\mu}, \widetilde{\mathbf{w}})$ be an eigenpair of H_m, and let $\widetilde{\lambda} = 1/\widetilde{\mu} + \sigma$. Denote by

$$\widehat{\mathbf{r}}^{\text{real}} = A V_m \widetilde{\mathbf{w}} - \widetilde{\lambda} B V_m \widetilde{\mathbf{w}} \quad (19)$$

the "real" residual with respect to the approximate eigenpair $(\widetilde{\lambda}, V_m\widetilde{\mathbf{w}})$ of the matrix pencil (A, B), and by

$$\mathbf{r}^{\text{real}} = (A-\sigma B)^{-1} B V_m \widetilde{\mathbf{w}} - \widetilde{\mu} V_m \widetilde{\mathbf{w}}, \quad (20)$$

and

$$\begin{aligned}
\mathbf{r}^{\text{comp}} &= \big[(A-\sigma B)^{-1} B + E\big] V_m \widetilde{\mathbf{w}} - \widetilde{\mu} V_m \widetilde{\mathbf{w}} \\
&= h_{m+1,m} (\mathbf{e}_m^\top \widetilde{\mathbf{w}}) \cdot \mathbf{v}_{m+1}, \quad (21)
\end{aligned}$$

the "real" and the "computed" residual for the approximate eigenpair $(\widetilde{\mu}, V_m\widetilde{\mathbf{w}})$ of $(A-\sigma B)^{-1}B$, respectively. Multiplying $(\widetilde{\lambda}-\sigma)(A-\sigma B)$ on both sides of (20) yields

$$\begin{aligned}
(\widetilde{\lambda}-\sigma)(A-\sigma B)\mathbf{r}^{\text{real}} &= (\widetilde{\lambda}-\sigma) B V_m \widetilde{\mathbf{w}} - (A-\sigma B) V_m \widetilde{\mathbf{w}} \\
&= \widetilde{\lambda} B V_m \widetilde{\mathbf{w}} - A V_m \widetilde{\mathbf{w}} = -\widehat{\mathbf{r}}^{\text{real}}.
\end{aligned}$$

As a result,

$$\|\widehat{\mathbf{r}}^{\text{real}}\| \leqslant |\widetilde{\lambda} - \sigma| \cdot \|A - \sigma B\| \cdot \|\mathbf{r}^{\text{real}}\|. \quad (22)$$

Thus, it is interesting to investigate the gap between $\mathbf{r}^{\text{real}}$ and $\mathbf{r}^{\text{comp}}$ in the "inexact" Hankel eigensolver.

In the following, we establish a relationship between the error of Hankel systems and the residual of eigenproblem (1) and investigate how to choose the stopping threshold ε for solving the Hankel systems (4). Based on that, we propose an inexact shift-and-invert Arnoldi method for solving the large-scale generalized Hankel eigenproblem.

Theorem 2 *Under the above notations, if $\varepsilon \ll 1$ and*

$$\varepsilon \leqslant \frac{|x_0| \cdot \widetilde{\delta}}{3\sqrt{m} \cdot \|\mathbf{x}\|_1 \|\mathbf{y}\|_1 \cdot \|B\|_2},$$

then

$$\|\mathbf{r}^{\text{real}} - \mathbf{r}^{\text{comp}}\|_2 \lesssim \widetilde{\delta},$$

where m is the step of the shift-and-invert Arnoldi process, δ is a prescribed tolerence.

Proof: By (11), we have

$$\begin{aligned}
\|\mathbf{f}_j\|_2 &= \|\widetilde{H}^{-1} B\mathbf{v}_j - H^{-1} B\mathbf{v}_j\|_2 \leqslant \|H^{-1} - \widetilde{H}^{-1}\|_2 \cdot \|B\mathbf{v}_j\|_2 \\
&\leqslant \left|\frac{1}{x_0}\right| [\varepsilon + (\varepsilon + (1+\varepsilon)\tilde{\varepsilon})(1+\varepsilon)] \|\boldsymbol{x}\|_1 \|\boldsymbol{y}\|_1 \|B\mathbf{v}_j\|_2.
\end{aligned}$$

If $\tilde{\varepsilon} \leqslant \varepsilon \ll 1$, then

$$\begin{aligned}
&\left|\frac{1}{x_0}\right| \cdot [\varepsilon + (\varepsilon + (1+\varepsilon)\widetilde{\varepsilon})(1+\varepsilon)] \cdot \|\mathbf{x}\|_1 \|\mathbf{y}\|_1 \left\|B\mathbf{v}_j\right\|_2 \\
&\qquad \lesssim \left|\frac{1}{x_0}\right| \cdot 3\varepsilon \cdot \|\mathbf{x}\|_1 \|\mathbf{y}\|_1 \|B\|_2 \cdot \left\|\mathbf{v}_j\right\|_2 \\
&\qquad \leqslant \left|\frac{3}{x_0}\right| \cdot \|\mathbf{x}\|_1 \|\mathbf{y}\|_1 \|B\|_2 \cdot \varepsilon
\end{aligned}$$

where we removed the high order term $\mathcal{O}(\varepsilon)$. Consequently, if

$$\varepsilon \leqslant \frac{|x_0| \cdot \widetilde{\delta}}{3\sqrt{m} \cdot \|\mathbf{x}\|_1 \|\mathbf{y}\|_1 \cdot \|B\|_2},$$

i.e.,

$$\|\mathbf{f}_j\|_2 \lesssim \left|\frac{3}{x_0}\right| \cdot \|\mathbf{x}\|_1 \|\mathbf{y}\|_1 \|B\|_2 \cdot \varepsilon \leqslant \frac{\widetilde{\delta}}{\sqrt{m}}, \quad (23)$$

for $j = 1, 2, \ldots, m$, then we have from (20), (21) and (23) that

$$\|\mathbf{r}^{\text{real}} - \mathbf{r}^{\text{comp}}\|_2 = \|EV_m\widetilde{\mathbf{w}}\|_2 = \Big\|\sum_{j=1}^{m}\mathbf{f}_j\widetilde{w}_j\Big\|_2 \leqslant \sum_{j=1}^{m}\|\mathbf{f}_j\|_2\cdot|\widetilde{w}_j| \leqslant \max_{1\leqslant j\leqslant m}\|\mathbf{f}_j\|_2\cdot\sqrt{m}\|\widetilde{\mathbf{w}}\|_2 \lesssim \widetilde{\delta},$$

where $\widetilde{\mathbf{w}} = [\widetilde{w}_1, \ldots, \widetilde{w}_m]^\top$, and we utilized the inequality $\|\widetilde{\mathbf{w}}\|_1 \leqslant \sqrt{m}\|\widetilde{\mathbf{w}}\|_2 = \sqrt{m}$. □

We prefer to use the 2-norm in practical computations since V_m is orthonormal. We want to provide an approximate stopping criterion for solving the Hankel systems (4). We can utilize the following inequality as the stopping criterion for the Hankel eigencomputation by (3):

$$\frac{\|A\widetilde{\mathbf{x}}_i - \widetilde{\lambda}_i B\widetilde{\mathbf{x}}_i\|_2}{\|A-\sigma B\|_2} \leqslant h_{m+1,m}|\widetilde{\lambda}_i - \sigma|\cdot|\mathbf{e}_m^\top\widetilde{\mathbf{w}}_i| \leqslant \delta, \quad (24)$$

for $i = 1, 2, \ldots, \ell$, where δ is the user-prescribed tolerance for the Hankel eigenproblem and ℓ is the required number of eigenpairs. Inspired by (22) and (24), let

$$\widetilde{\delta} = \|A-\sigma B\|_2\cdot\delta,$$

then it follows from Theorem 2 that if

$$\varepsilon \leqslant \frac{|x_0|\cdot\|A-\sigma B\|_2}{3\sqrt{m}\cdot\|\mathbf{x}\|_1\|\mathbf{y}\|_1\cdot\|B\|_2}\cdot\delta, \quad (25)$$

then we obtain that

$$\frac{\|\mathbf{r}^{\text{real}} - \mathbf{r}^{\text{comp}}\|_2}{\|A-\sigma B\|_2} \lesssim \delta.$$

Remark 1 From equation (25), we observe that when

$$\xi \equiv \frac{|x_0|}{\|\mathbf{x}\|_1\|\mathbf{y}\|_1} \quad \text{and} \quad \eta \equiv \frac{\|A-\sigma B\|_2}{3\sqrt{m}\cdot\|B\|_2}$$

are small, it is necessary to solve the Hankel linear systems with a relatively high accuracy, as given in equation (4). Otherwise, we can solve them with a relatively low accuracy. Some parameters that appear in equation (25), such as $\|\mathbf{x}\|_1$ and $\|\mathbf{y}\|_1$, are not always available. If ξ is not too small, we recommend using

$$\big\{\|\mathbf{r}_x\|_2, \|\mathbf{r}_y\|_2\big\} \leqslant \frac{\max(\|\mathbf{fc}_{A-\sigma B}\|_2, \|\mathbf{lr}_{A-\sigma B}\|_2)}{3\sqrt{m}\cdot\max(\|\mathbf{fc}_B\|_2, \|\mathbf{lr}_B\|_2)}\cdot\delta \quad (26)$$

as the stopping criterion for solving the Hankel systems, where $\mathbf{r}_x = \mathbf{e_n} - H\widetilde{\mathbf{x}}$ and $\mathbf{r}_y = \mathbf{e}_1 - H\widetilde{\mathbf{y}}$ are the residuals of the Hankel systems; $\mathbf{fc}_{A-\sigma B}$ and $\mathbf{lr}_{A-\sigma B}$ are the first column and the last row of $A-\sigma B$, respectively; and $\mathbf{fc}_B$ and $\mathbf{lr}_B$ are the first column and the last row of B, respectively. The effectiveness of this scheme is demonstrated in the numerical experiments in the following section. Actually, the solution of $\mathbf{x}$ of Hankel linear system $H\mathbf{x} = \mathbf{b}$ can be obtained by solving $JH\mathbf{x} = J\mathbf{b}$, where JH is a Toeplitz matrix. Thus we can use the GMRES algorithm with Chan's preconditioner [28, 29] for solving (4).

The implementation of the shift-and-invert Arnoldi process in practical computations is limited by the high storage and computational complexity as the Arnoldi step m increases. In our algorithm, we can employ some restarting strategies, such as the implicitly restarted shift-and-invert Arnoldi algorithm [30] or the thick-restarted Arnoldi algorithm [31, 32], to address these difficulties. The algorithm is described as follows:

Algorithm 1 An inexact shift-and-invert Arnoldi algorithm for generalized Hankel eigenproblems

Step 1. Given a shift σ, a convergence threshold δ for the eigenproblem, and four vectors $\mathbf{fc}_A$, $\mathbf{lr}_A$, $\mathbf{fc}_B$, and $\mathbf{lr}_B$, which are the first column and the last row of A and B, respectively. Compute the inverse of $A-\sigma B$. Solve the Hankel linear systems (4) "inexactly" with the stopping criterion given in (26).

Step 2. Compute the desired eigenpairs using a restarted shift-and-invert Krylov subspace algorithm, such as the implicitly restarted shift-and-invert Arnoldi algorithm [30] or the thick-restarted Arnoldi algorithm [31, 32].

NUMERICAL EXPERIMENTS

In this section, we present numerical experiments to demonstrate the efficiency of our new algorithm and validate the theoretical results. All experiments were conducted on a MacOS 13 operating system with 3.20 GHz CPU and 8GB RAM, using a MATLAB 9.11.0 (R2021b) implementation with machine precision of $\epsilon \approx 2.22\times10^{-16}$. To solve the generalized Hankel eigenproblems, we utilize the MATLAB built-in function **eigs.m**, which implements the implicitly restarted shift-and-invert Arnoldi algorithm, and the Hankel matrix-vector products are realised by using the fast implementation [21]. We use the default parameter settings provided by **eigs.m** for the numerical experiments. The algorithms used in this section are described as follows:

• **Inexact-eigs (Algorithm 1)** represents the "inexact" shift-and-invert Arnoldi algorithm that employs the (unrestarted) GMRES algorithm with Chan's preconditioner [28, 29] as the solver for (4). Given a convergence threshold δ, we use (26) as the stopping criterion for the Hankel systems. Specifically, we use

$$\|\mathscr{M}^{-1}\mathbf{b} - \mathscr{M}^{-1}(A-\sigma B)\widetilde{\mathbf{q}}\|_2 \leqslant \frac{\max(\|\mathbf{fc}_{A-\sigma B}\|_2, \|\mathbf{lr}_{A-\sigma B}\|_2)}{3\sqrt{20}\max(\|\mathbf{fc}_B\|_2, \|\mathbf{lr}_B\|_2)}\delta \quad (27)$$

as the stopping criterion for the Hankel systems, where $\mathscr{M}$ stands for Chan's preconditioner, $\widetilde{\mathbf{q}} = \widetilde{\mathbf{x}}$ or $\widetilde{\mathbf{y}}$ is the approximate solution, and $\mathbf{b} = \mathbf{e_n}$ or $\mathbf{e}_1$ represents the right-hand side. This algorithm approximates the solution of the Hankel systems with an iterative method.

- **Exact-eigs** represents the "exact" shift-and-invert Arnoldi algorithm that employs the (unrestarted) GMRES algorithm with Chan's preconditioner for (4). For the Hankel linear systems, the stopping criterion is:

$$\|\mathscr{M}^{-1}\mathbf{b} - \mathscr{M}^{-1}(A-\sigma B)\widetilde{\mathbf{q}}\|_2 \leqslant 10^{-14}.$$

An iterative solver is used to solve the two Hankel systems "exactly" in this algorithm. We use the zero vector as the initial guess for GMRES in both Inexact-eigs and Exact-eigs.

In the following tables, "n" indicates the Hankel matrix size, "(27)" indicates the value of the right-hand side of "(27)", "CPU (s)" signifies the CPU time utilized in seconds, and "–" indicates that the algorithm is "out of memory". To demonstrate the effiency of the inexact approach, we provide the maximum "real" residual norm, cf. (19):

$$\|\widehat{\mathbf{r}}^{\text{real}}\|_2 = \max_{i=1,\ldots,\ell} \|A(V_m\widetilde{\mathbf{w}}_i) - \widetilde{\lambda}_i B(V_m\widetilde{\mathbf{w}}_i)\|_2,$$

where ℓ is the number of required eigenpairs and $(\widetilde{\lambda}_i, V_m\widetilde{\mathbf{w}}_i), i = 1,2,\ldots,\ell$, are Ritz pairs gained from running Inexact-eigs. The CPU time of "Inexact-eigs" and "Exact-eigs" include the time to solve Hankel systems and compute eigenpairs.

We use the built-in MATLAB function `eigs(A,B)` to solve the same problem and calculate the computation time.

Example 1 The purpose of this example is to demonstrate the efficiency of the inexact strategy (27) for large generalized Hankel eigenproblems. We generate the Hankel matrices by

$$H = \begin{bmatrix} h_{-(n-1)} & h_{-(n-2)} & \cdots & h_{-1} & h_0 \\ h_{-(n-2)} & h_{-(n-3)} & \cdots & h_0 & h_1 \\ \vdots & \iddots & \iddots & h_1 & \vdots \\ h_{-1} & \iddots & \iddots & \ldots & h_{n-2} \\ h_0 & h_1 & \ldots & h_{n-2} & h_{n-1} \end{bmatrix},$$

with generating functions f are taken from [26]. We generate matrix A using the function

$$f(\theta) = \theta^2 + t\cdot \mathrm{i}\theta^3, \quad \theta \in [-\pi, \pi],$$

and matrix B using

$$f(\theta) = \theta^2 + t\cdot \mathrm{i}\,\mathrm{sgn}(\theta), \quad \theta \in [-\pi, \pi],$$

where $t > 0$ is a scalar and $\mathrm{sgn}(\theta)$ is the sign function. Our goal is to find the 10 smallest eigenpairs of (A, B) with $t = 1$ and a shift σ of 0.

Table 1 displays the residual, stopping criterion for solving the Hankel systems (4) in Inexact-eigs, and CPU time for solving the generalized Hankel eigensystems using different algorithms. We observe that the stopping criterion for Inexact-eigs in solving the Hankel systems (4) is approximately $\mathscr{O}(10^{-7})$, whereas Exact-eigs uses a tight stopping criterion of 10^{-14}. From Table 1, we can see that the Inexact-eigs strategy significantly reduces the CPU time compared with Exact-eigs and eigs(A, B), especially for larger system sizes. This illustrates the effectiveness of our "inexact" strategy.

Table 2 provides the 10 smallest eigenvalues obtained using the MATLAB built-in function eigs(A, B), as well as the approximations calculated by running Inexact-eigs and Exact-eigs. The results show that the eigenpairs obtained from Inexact-eigs are accurate enough.

Example 2 The test matrices used in this example are derived from the references [33, 34]. We generate the Hankel matrix A using the even function θ^2 defined on $[0, \pi]$, which was introduced in Example 1. The Hankel matrix $B = (b_{ij})$ is given by

$$b_{ij} = \begin{cases} 1 + \frac{\pi^4}{5}, & \text{if } i+j=n+1, \\ (-1)^{|n-i-j-1|}\left(\frac{4\pi^2}{|n+1-i-j|^2} - \frac{24}{|n+1-i-j|^4}\right), & \text{otherwise,} \end{cases}$$

which is derived from the even function $\theta^4 + 1$ limited to $[-\pi, \pi]$. We aim to compute the 8 eigenpairs closet to $\sigma = 0.5$ for this test problem.

We compare three algorithms in this example: the Inexact-eigs scheme, the Exact-eigs scheme, and MATLAB's built-in function eigs(A, B). Table 3 list the CPU times for these algorithms.

We can see from Table 3 that the Inexact-eigs method outperforms both the Exact-eigs method and eigs(A, B) in terms of CPU time. This superiority can be attributed to using the formula (27) for solving Hankel systems, which allows the Inexact-eigs scheme to use a much looser stopping criterion of $\mathscr{O}(10^{-7})$ compared to the Exact-eigs scheme with stopping criterion of 10^{-14}. This improvement is significant especially when n is large.

Example 3 The test practical problem is from the reference [35]. We consider the fractional diffusion equation

$$\begin{cases} \frac{\partial u(x,t)}{\partial t} = d_1 \frac{\partial^\alpha u(x,t)}{\partial_+ x^\alpha} + d_2 \frac{\partial^\alpha u(x,t)}{\partial_- x^\alpha} + f(x,t), \\ \qquad\qquad x \in (x_L, x_R),\ t \in (0, T], \\ u(x_L, t) = u(x_R, t) = 0, \qquad 0 \leqslant t \leqslant T, \\ u(x, 0) = u_0(x), \qquad x \in [x_L, x_R]. \end{cases} \tag{28}$$

By employing the shift Grünwald approximation, the corresponding linear equation of (28) can be written

Table 1 Numerical results of the algorithms for computing the 10 smallest eigenpairs in magnitude with $\sigma = 0$ and the tolerance $\delta = 10^{-6}$, $t = 1$; where "–" represents "out of memory".

n	(27)	$\|\widehat{\mathbf{r}}^{real}\|_2$	CPU (s) Inexact-eigs	CPU (s) Exact-eigs	CPU (s) eigs(A,B)
2^{10}	1.7949×10^{-7}	3.5066×10^{-9}	0.1504	0.2411	0.1242
2^{12}	1.7955×10^{-7}	2.1892×10^{-10}	0.3658	49.3848	1.5663
2^{14}	1.7956×10^{-7}	3.3090×10^{-11}	2.2071	–	88.4073
2^{16}	1.7957×10^{-7}	1.9814×10^{-12}	20.0692	–	–
2^{18}	1.7957×10^{-7}	5.1204×10^{-13}	225.8262	–	–

Table 2 The 10 smallest eigenvalues in magnitude computed by using eigs(A,B) and the approximations obtained from running Inexact-eig and Exact-eig, $n = 3000$, $t = 1$.

eigs: λ_i, $i = 1, 2, \ldots, 10$	Inexact-eigs	Exact-eigs
$5.42135073 \times 10^{-9} \pm 3.00211667 \times 10^{-6}$i	$5.42135053 \times 10^{-9} \pm 3.00211671 \times 10^{-6}$i	$5.42135097 \times 10^{-9} \pm 3.00211667 \times 10^{-6}$i
$5.64830095 \times 10^{-8} \pm 1.45117763 \times 10^{-5}$i	$5.64829970 \times 10^{-8} \pm 1.45117765 \times 10^{-5}$i	$5.64830052 \times 10^{-8} \pm 1.45117763 \times 10^{-5}$i
$2.08752894 \times 10^{-7} \pm 3.47985781 \times 10^{-5}$i	$2.08752852 \times 10^{-7} \pm 3.47985784 \times 10^{-5}$i	$2.08752893 \times 10^{-7} \pm 3.47985781 \times 10^{-5}$i
$5.18404010 \times 10^{-7} \pm 6.38568356 \times 10^{-5}$i	$5.18403893 \times 10^{-7} \pm 6.38568363 \times 10^{-5}$i	$5.18404008 \times 10^{-7} \pm 6.38568356 \times 10^{-5}$i
$1.04205274 \times 10^{-6} \pm 1.01686715 \times 10^{-4}$i	$1.04205245 \times 10^{-6} \pm 1.01686716 \times 10^{-4}$i	$1.04205273 \times 10^{-6} \pm 1.01686715 \times 10^{-4}$i

as

$$\left(\frac{h^\alpha}{\Delta t} I - D^{(m)}\right) u^{(m)} = \frac{h^\alpha}{\Delta t} u^{(m-1)} + h^\alpha f^{(m)},$$

with $h = 1/(n+1)$, $\Delta t = 2h$ are the size of spatial grid and time step respectively, and

$$D^{(m)} = d_1 G_\alpha + d_2 G_\alpha^T,$$

where

$$G_\alpha = \begin{bmatrix} g_1^{(\alpha)} & g_0^{(\alpha)} & 0 & \cdots & 0 & 0 \\ g_2^{(\alpha)} & g_1^{(\alpha)} & g_0^{(\alpha)} & 0 & \cdots & 0 \\ \vdots & g_2^{(\alpha)} & g_1^{(\alpha)} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \\ g_{n-1}^{(\alpha)} & \ddots & \ddots & \ddots & g_1^{(\alpha)} & g_0^{(\alpha)} \\ g_n^{(\alpha)} & g_{n-1}^{(\alpha)} & \cdots & \cdots & g_2^{(\alpha)} & g_1^{(\alpha)} \end{bmatrix}_{n \times n}$$

the entries of G_α are defined as follows:

$$\begin{cases} g_0^{(\alpha)} = 1, \\ g_{k+1}^{(\alpha)} = \left(1 - \frac{\alpha+1}{k+1}\right) g_k^{(\alpha)}, \qquad k = 1, 2, 3, \ldots. \end{cases}$$

We choose the diffusion coefficients $d_1 = 0.8$, $d_2 = 0.2$, $\alpha = 1.4$. Let $A = \frac{h^\alpha}{\Delta t} I - D^{(m)}$, A is a nonsymmetric Toeplitz matrix. The permutation matrix J was introduced [35, 36] to transform the non-symmetric Toeplitz matrix problem into symmetric Hankel matrix problem in order to use the properties of symmetric matrix. To analyze the eigenpairs of A is equal to solving the generalised Hankel eigenproblem $H\mathbf{x} = \lambda J\mathbf{x}$, where $H = JT$. We compute the 10 eigenpairs with $\sigma = 0$ and $\sigma = 0.5$ in Table 4 and Table 5, respectively. We can see from Table 4 and Table 5 that the Inexact-eigs scheme converges faster than Exact-eigs method in most cases. When n is large, both Exact-eigs and eigs(A,B) do not work, due to the heavy computational complexity. As a comparison, our Inexact-eigs algorithm still converges within acceptable CPU time. Thus, the new algorithm is preferable to large Hankel generalized eigenproblems.

CONCLUSION

In this paper, we propose an inexact shift-and-invert Arnoldi algorithm for solving generalized Hankel eigenproblems. Firstly, we need to solve two large Hankel systems, but the cost becomes prohibitive if the desired accuracy is too high. To overcome this

Table 3 Numerical results of the algorithms for computing the 8 eigenpairs closet to $\sigma = 0.5$ and the tolerance $\delta = 10^{-6}$; where "–" represents "out of memory".

n	(27)	$\|\widehat{\mathbf{r}}^{real}\|_2$	CPU (s) Inexact-eigs	CPU (s) Exact-eigs	CPU (s) eigs(A,B)
2^{10}	1.8834×10^{-7}	1.4991×10^{-7}	0.1247	0.1343	0.2231
2^{12}	1.8834×10^{-7}	8.3084×10^{-8}	0.1829	0.2851	3.0463
2^{14}	1.8834×10^{-7}	3.4707×10^{-7}	0.3205	1.1403	219.0806
2^{16}	1.8834×10^{-7}	2.4584×10^{-8}	0.8528	11.9347	–
2^{18}	1.8834×10^{-7}	2.2137×10^{-9}	1.6465	–	–

Table 4 Numerical results of the algorithms for computing the 10 smallest eigenpairs in magnitude with $\sigma = 0$ and the tolerance $\delta = 10^{-6}$; where "–" represents "out of memory".

n	(27)	$\|\widehat{\mathbf{r}}^{real}\|_2$	CPU (s) Inexact-eigs	CPU (s) Exact-eigs	CPU (s) eigs(A,B)
2^{10}	7.2140×10^{-8}	1.7265×10^{-8}	0.1537	0.1700	0.2382
2^{12}	7.2140×10^{-8}	1.7059×10^{-8}	0.2366	1.4420	2.8346
2^{14}	7.2136×10^{-8}	1.8922×10^{-9}	0.4016	4.7940	215.0230
2^{16}	7.2135×10^{-8}	4.3532×10^{-9}	1.4058	–	–
2^{18}	7.2135×10^{-8}	3.8934×10^{-10}	5.4774	–	–

Table 5 Numerical results of the algorithms for computing the 10 eigenpairs closet to $\sigma = 0.5$ and the tolerance $\delta = 10^{-6}$; where "–" represents "out of memory".

n	(27)	$\|\widehat{\mathbf{r}}^{real}\|_2$	CPU (s) Inexact-eigs	CPU (s) Exact-eigs	CPU (s) eigs(A,B)
2^{10}	3.7268×10^{-8}	1.2304×10^{-8}	0.1824	4.9649	0.2969
2^{12}	3.7268×10^{-8}	1.4190×10^{-9}	0.3370	870.5256	3.7431
2^{14}	9.8939×10^{-10}	5.4106×10^{-12}	0.6775	–	181.6075
2^{16}	3.7268×10^{-8}	8.0073×10^{-10}	2.3588	–	–
2^{18}	3.7268×10^{-8}	9.4170×10^{-10}	13.5398	–	–

difficulty, we establish a relationship between the error of the Hankel systems and the residual of the Hankel eigenproblem, and we provide a cheap stopping criterion for solving the Hankel systems inexactly. Numerical results show that our "inexact" strategy outperform solving the Hankel systems "exactly", especially when the Hankel systems are large.

Acknowledgements: We would like to express our sincere thanks to the editor Dr. Xian-Ming Gu and the anonymous referees for their insightful comments and invaluable suggestions that greatly improved the representation of this paper. This work is supported by the National Natural Science Foundation of China (No. 12001146) and the Zhejiang Provincial Natural Science Foundation of China (No. LQ21A010006).

REFERENCES

1. Peller V (2003) *Hankel Operators and Their Applications*, Springer, New York, NY.
2. Beckermann B, Golub GH, Labahn G (2007) On the numerical condition of a generalized Hankel eigenvalue problem. *Numer Math* **106**, 41–68.
3. Luk FT, Qiao S (2000) A fast eigenvalue algorithm for Hankel matrices. *Linear Algebra Appl* **316**, 171–182.
4. Chen Y, Sikorowski J, Zhu M (2019) Smallest eigenvalue of large Hankel matrices at critical point: comparing conjecture with parallelised computation. *Appl Math Comput* **363**, 124628.
5. Emmart N, Chen Y, Weems CC (2015) Computing the smallest eigenvalue of large ill-conditioned Hankel matrices. *Commun Comput Phys* **18**, 104–124.
6. Wang D, Zhu M, Chen Y (2022) The smallest eigenvalue of large Hankel matrices associated with a singularly perturbed Gaussian weight. *Proc Amer Math Soc* **150**, 153–160.
7. Zhu M, Chen Y, Li C (2020) The smallest eigenvalue of large Hankel matrices generated by a singularly perturbed Laguerre weight. *J Math Phys* **61**, 073502.
8. Saad Y (2011) *Numerical Methods for Large Eigenvalue Problems: Revised Edition*, SIAM, Philadelphia, PA.
9. Feng TT, Wu G, Xu TT (2015) An inexact shift-and-invert Arnoldi algorithm for large non-Hermitian generalised Toeplitz eigenproblems. *East Asian J Appl Math* **5**, 160–175.
10. Pang HK, Sun HW (2011) Shift-invert Lanczos method for the symmetric positive semidefinite Toeplitz matrix exponential. *Numer Linear Algebra Appl* **18**, 603–614.
11. Wu G, Feng TT, Wei Y (2015) An inexact shift-and-invert Arnoldi algorithm for Toeplitz matrix exponential. *Numer Linear Algebra Appl* **22**, 777–792.
12. Jiang X, Zhang G, Zheng Y, Jiang Z (2024) Explicit potential function and fast algorithm for computing potentials in $\alpha \times \beta$ conic surface resistor network. *Expert Syst Appl* **238**, 122157.
13. Jiang Z, Zhou Y, Jiang X, Zheng Y (2023) Analytical potential formulae and fast algorithm for a horn torus resistor network. *Phys Rev E* **107**, 044123.
14. Zhang X, Zheng Y, Jiang Z, Byun H (2023) Numerical algorithms for corner-modified symmetric Toeplitz linear system with applications to image encryption and decryption. *J Appl Math Comput* **69**, 1967–1987.
15. Zhou Y, Zheng Y, Jiang X, Jiang Z (2022) Fast algorithm and new potential formula represented by Chebyshev polynomials for an $m \times n$ globe network. *Sci Rep* **12**, 21260.
16. Meng Q, Zheng Y, Jiang Z (2022) Exact determinants and inverses of (2, 3, 3)-Loeplitz and (2, 3, 3)-Foeplitz matrices. *Comput Appl Math* **41**, 35.
17. Meng Q, Zheng Y, Jiang Z (2022) Determinants and inverses of weighted Loeplitz and weighted Foeplitz matrices and their applications in data encryption. *J Appl Math Comput* **68**, 3999–4015.
18. Wang J, Zheng Y, Jiang Z (2023) Norm equalities and

inequalities for tridiagonal perturbed Toeplitz operator matrices. *J Appl Anal Comput* **13**, 671–683.
19. Zhang X, Jiang X, Jiang Z, Byun H (2023) Algorithms for solving a class of real quasi-symmetric Toeplitz linear systems and its applications. *Electron Res Arch* **31**, 1966–1981.
20. Wu J, Gu XM, Zhao YL, Huang YY, Carpentieri B (2023) A note on the structured perturbation analysis for the inversion formula of Toeplitz matrices. *Japan J Indust Appl Math* **40**, 645–663.
21. Zhang X, Jiang X, Jiang Z, Byun H (2022) An improvement of methods for solving the CUPL-Toeplitz linear system. *Appl Math Comput* **421**, 126932.
22. Zheng Y, Fu Z, Shon S (2017) A new Toeplitz inversion formula, stability analysis and the value. *J Nonlinear Sci Appl* **10**, 1089–1097.
23. Golub GH, Van Loan CF (2013) *Matrix Computations*, 4th edn, Johns Hopkins University Press, Baltimore, MA.
24. Jia Z, Zhang Y (2002) A refined shift-and-invert Arnoldi algorithm for large unsymmetric generalized eigenproblems. *Comput Math Appl* **44**, 1117–1127.
25. Narasimha MJ (2007) Linear convolution using skew-cyclic convolutions. *IEEE Signal Process Lett* **14**, 173–176.
26. Lee ST, Pang HK, Sun HW (2010) Shift-invert Arnoldi approximation to the Toeplitz matrix exponential. *SIAM J Sci Comput* **32**, 774–792.
27. Jiang Z, Zhou J (2015) A note on spectral norms of even-order r-circulant matrices. *Appl Math Comput* **250**, 368–371.
28. Chan RHF, Jin XQ (2007) *An Introduction to Iterative Toeplitz Solvers*, SIAM, Philadelphia, PA.
29. Ng MK (2004) *Iterative Methods for Toeplitz Systems*, Numerical Mathematics and Scientific Computation, Oxford University Press, NewYork, NY.
30. Sorensen DC (1992) Implicit application of polynomial filters in a k-step Arnoldi method. *SIAM J Matrix Anal Appl* **13**, 357–385.
31. Morgan RB, Zeng M (2006) A harmonic restarted Arnoldi algorithm for calculating eigenvalues and determining multiplicity. *Linear Algebra Appl* **415**, 96–113.
32. Wu K, Simon H (2000) Thick-restart Lanczos method for large symmetric eigenvalue problems. *SIAM J Matrix Anal Appl* **22**, 602–616.
33. Ng MK (2000) Preconditioned Lanczos methods for the minimum eigenvalue of a symmetric positive definite Toeplitz matrix. *SIAM J Sci Comput* **21**, 1973–1986.
34. Wang YY, Lu LZ (2009) Preconditioned Lanczos method for generalized Toeplitz eigenvalue problems. *J Comput Appl Math* **226**, 66–76.
35. Wang SF, Huang TZ, Gu XM, Luo WH (2016) Fast permutation preconditioning for fractional diffusion equations. *SpringerPlus* **5**, 1109.
36. Pestana J, Wathen AJ (2015) A preconditioned MINRES method for nonsymmetric Toeplitz matrices. *SIAM J Matrix Anal Appl* **36**, 273–288.