An inexact Krylov subspace method for large generalized Hankel eigenproblems

Zi-Yan Huang¹, Ting-Ting Feng²

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.


1 Introduction

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

$$H = [h_{i+k}]_{i,k=0}^{n-1} = \begin{bmatrix}
h_0 & h_1 & \cdots & h_{n-1} \\
h_1 & h_2 & \cdots & h_n \\
\vdots & \vdots & \ddots & \vdots \\
h_{n-1} & h_n & \cdots & h_{2n-2}
\end{bmatrix},$$

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 [12, 17, 20, 21]. 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 [1]. However, there have been relatively few works focusing on the Hankel eigenproblem. In [11], a fast eigenvalue algorithm for Hankel matrices was proposed based on the Lanczos-type tridiagonalization and QR-type diagonalization methods. Some studies [2, 4, 22, 27] have focused specifically on the smallest eigenvalue of large scale Hankel matrices. In [1], the sensitivity of the nonlinear application 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 in [2, 4, 22].

¹School of Science, Hangzhou Dianzi University, Hangzhou, Zhejiang, P.R. China. Email: huangziyan0718@outlook.com.
²Corresponding author. School of Science, Hangzhou Dianzi University, Hangzhou, Zhejiang, P.R. China. Email: ttfeng@hdu.edu.cn. This author is supported by the National Natural Science Foundation of China (No.12001146) and the Zhejiang Provincial Natural Science Foundation of China (No. LQ21A010006).
The Krylov subspace method is an efficient approach for computing the smallest eigenpair or a few extreme eigenpairs of large-scale matrices [18]. 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 only matrix-vector multiplications [18]. 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 [18]. By using the shift-and-invert technique, the multiplication of a inverse of a matrix and vector is important.

Recently, the shift-and-invert Arnoldi or Lanczos method has been used in designing fast algorithms for the generalized Toeplitz eigenproblem [5], and Toeplitz matrix exponential [16, 25]. Due to the special structure of Toeplitz matrices and the famous Gohberg-Semencul formula [7], the products of the inverse of a Toeplitz matrix and a vector can be implemented using several FFTs [16, 25]. 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 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:

\[ Ax = \lambda Bx, \quad (1.1) \]

where \( A \) and \( B \) are large-scale Hankel matrices and the matrix pencil \( (A, B) \) is regular [8, 18]. The generalized eigenproblem reduces to the standard Hankel eigenproblem when \( B \) is the identity matrix. 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 these systems becomes prohibitive. 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. In Section 2, we briefly introduce the shift-and-invert Arnoldi method. In Section 3, 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. Numerical examples are given to verify the efficiency of our theoretical results in Section 4. Finally, in Section 5, we provide concluding remarks.

2 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 [18]. Given a shift \( \sigma \in \mathbb{C} \), we can derive from (1.1) that

\[ (A - \sigma B)x = (\lambda - \sigma)Bx. \]

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

\[ (A - \sigma B)^{-1}Bx = \mu x, \quad (2.1) \]

where \( \mu = \frac{1}{\lambda - \sigma} \). 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 \( \sigma \) 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 (1.1) directly [18].

The shift-and-invert Arnoldi method is commonly used for computing several eigenvalues closest to a given shift \(\sigma\) and/or the associated eigenvectors of a large non-Hermitian matrix. Given a unit vector \(v_1\), the \(m\)-step shift-and-invert Arnoldi process, formulated in exact arithmetic, can be expressed as [18]

\[
(A - \sigma B)^{-1} B V_m = V_m H_m + h_{m+1,m} v_{m+1} e_m^T,
\]

(2.2)

where \(V_m = [v_1, v_2, \ldots, v_m]\) is an orthonormal basis for the Krylov subspace \(K_m((A - \sigma B)^{-1} B, v_1)\), \(e_m\) is the \(m\)-th column of the \(m\)-by-\(m\) identity matrix, and \(H_m\) is an \(m\)-by-\(m\) upper Hessenberg matrix. Let \((\tilde{\lambda}, \tilde{w})\) with \(\|\tilde{w}\|_2 = 1\) be an eigenpair of \(H_m\). The shift-and-invert Arnoldi method uses \((\tilde{\lambda} = 1/\tilde{\mu} + \sigma, \tilde{x} = V_m \tilde{w})\) as an approximation to \((\lambda, x)\). If we denote the residual corresponding to the Ritz pair \((\tilde{\lambda}, \tilde{x})\) by \(\hat{r} = A \tilde{x} - \tilde{\lambda} B \tilde{x}\), then we have that [9]

\[
\frac{||\hat{r}||_2}{||A - \sigma B||_2} \leq h_{m+1,m} |\tilde{\lambda} - \sigma| \cdot |e_m^T \tilde{w}|,
\]

(2.3)

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 not be non-Hermitian. Therefore, the shift-and-invert Arnoldi method is utilized in the following sections.

3 An inexact shift-and-invert Arnoldi method for generalized Hankel eigenproblems

The computation of \(m\) matrix-vector products \((A - \sigma B)^{-1} B v_j\), where \(j = 1, 2, \ldots, m\), is required for the Shift-and-invert Arnoldi process. One option is to compute the inverse \((A - \sigma B)^{-1}\) using LU decomposition [8], but this can be costly, especially for large dense matrices. As \(\sigma\) is a given shift, we are interested in computing \((A - \sigma B)^{-1}\) once for all.

Fortunately, for Hankel matrices \(A\) and \(B\), the matrix \(A - \sigma B\) is also a Hankel matrix. For a nonsingular Hankel matrix \(H = (h_{i+k})(i, k = 0, \ldots, n - 1)\), there is a formula [6] 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 \(x = [x_0, x_1, \ldots, x_{n-1}]^T\) and \(y = [y_0, y_1, \ldots, y_{n-1}]^T\) be the solutions of the following two Hankel systems:

\[
H x = p \quad \text{and} \quad H y = e_n,
\]

(3.4)

where \(p = (h_n, h_{n+1}, \ldots, h_{2n-2}, 0)^T\), and \(e_n\) is the last column of the \(n\)-by-\(n\) identity matrix. We have the following formula [6]:
\[ H^{-1} = \begin{bmatrix} y_1 & y_2 & \cdots & y_{n-1} & 0 \\ y_2 & y_3 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ y_{n-1} & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 \end{bmatrix} - \begin{bmatrix} x_1 & x_2 & \cdots & x_{n-1} & -1 \\ x_2 & x_3 & \cdots & -1 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{n-1} & -1 & \cdots & 0 & 0 \\ -1 & 0 & \cdots & 0 & 0 \end{bmatrix} \equiv Y_0X - X_0Y. \] (3.5)

Then the above formula (3.5) yields an explicit representation of $H^{-1}$ in terms of two anti-triangular Hankel matrices and two triangular Toeplitz matrices. Therefore, the products of the inverse of a Hankel matrix and vectors can be realized by several FFTs.

To relieve the burden of computing the matrix-vector products with respect to $H^{-1}$, we notice that $TH = HJTJ$, where $T$ is an upper (or lower) triangular Toeplitz matrix, $H$ is an upper (or lower) anti-triangular Hankel matrix, and $J$ is the $n \times n$ matrix given by

\[ J = \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & \cdots & 1 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & 0 & \cdots & 0 & 0 \end{bmatrix}. \]

Combining the fact that two upper (or lower) triangular Toeplitz matrices commute, we obtain

\[ H^{-1} = \frac{1}{2} \left[ (Y_0 + JY)(X - JX_0) + (JX + X_0)(JY_0 - Y) \right] \] (3.6)

\[ = \frac{1}{2} J \left[ (JY_0 + Y)(X - JX_0) + (X + JX_0)(JY_0 - Y) \right]. \] (3.7)

As circulant matrices can be diagonalized by the Fourier matrix and skew-circulant matrices have a similar spectral decomposition, the Hankel matrix in (3.5) can be factored into two circulant matrices and two skew-circulant matrices. Six FFTs of length $n$ are needed to obtain the Hankel matrix-vector product. For further information, see \[10, 16, 24\] 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 precise solution of two large-scale Hankel linear systems as in (3.4), solving the Hankel systems with as much precision as possible is preferred. The expense 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 \[10, 16\] to reduce the cost. This approach can be viewed as an “inexact” inverse technology since the Hankel linear systems are solved once and for all.

Let $\tilde{x} = [\tilde{x}_0, \tilde{x}_1, \ldots, \tilde{x}_{n-1}]^T$ and $\tilde{y} = [\tilde{y}_0, \tilde{y}_1, \ldots, \tilde{y}_{n-1}]^T$ be the numerical solutions of the two Hankel systems $H\tilde{x} = p$ and $H\tilde{y} = e_n$, respectively. We define
Proof. It follows from (3.5) and (3.8) that
\[
\begin{aligned}
\tilde{H}^{-1} &= \begin{bmatrix}
\tilde{y}_1 & \tilde{y}_2 & \ldots & \tilde{y}_{n-1} & 0 \\
\tilde{y}_2 & \tilde{y}_3 & \ldots & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
\tilde{y}_{n-1} & 0 & \ldots & 0 & 0 \\
0 & 0 & \ldots & 0 & 0
\end{bmatrix} \\
&= \begin{bmatrix}
\tilde{x}_0 & \tilde{x}_1 & \ldots & \tilde{x}_{n-2} & \tilde{x}_{n-1} \\
0 & \tilde{x}_0 & \ldots & \tilde{x}_{n-3} & \tilde{x}_{n-2} \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \ldots & \tilde{x}_0 & \tilde{x}_1 \\
0 & 0 & \ldots & 0 & \tilde{x}_0
\end{bmatrix}
\end{aligned}
\]

where \(\tilde{H}^{-1}\) can be considered as a perturbation to the Hankel inverse \(H^{-1}\). Inspired by the paper \([5]\), we analyze the stability of formula (3.5) with respect to the 1-norm and 2-norm.

**Theorem 1.** Let \(\varepsilon > 0\). If
\[
\frac{\|X - \tilde{y}_0\|_1}{\|\tilde{y}_0\|_1} \leq \varepsilon \quad \text{and} \quad \frac{\|\tilde{y} - \tilde{y}_0\|_1}{\|\tilde{y}_0\|_1} \leq \varepsilon,
\]
then
\[
\|H^{-1} - \tilde{H}^{-1}\|_\infty \leq (4\varepsilon + \varepsilon^2) \cdot \|X\|_1 \|y\|_1 + \varepsilon \|y\|_1.
\]

**Proof.** It follows from (3.5) and (3.8) that
\[
\|H^{-1} - \tilde{H}^{-1}\|_\infty = \|Y_0X - X_0Y \|_\infty - (\tilde{Y}_0\tilde{X} - \tilde{X}_0\tilde{Y})\|_\infty
\]
\[
= \|Y_0X - \tilde{Y}_0\tilde{X}\|_\infty - (X_0Y - \tilde{X}_0\tilde{Y})\|_\infty
\]
\[
\leq \|Y_0X - \tilde{Y}_0\tilde{X}\|_\infty + \|X_0Y - \tilde{X}_0\tilde{Y}\|_\infty.
\]

Moreover, we deduce that
\[
\|Y_0X - \tilde{Y}_0\tilde{X}\|_\infty = \|Y_0X - Y_0\tilde{X} + Y_0\tilde{X} - \tilde{Y}_0\tilde{X}\|_\infty
\]
\[
= \|Y_0(X - \tilde{X}) + (Y_0 - \tilde{Y}_0)\tilde{X}\|_\infty
\]
\[
\leq \|Y_0\|_\infty \|X - \tilde{X}\|_\infty + \|Y_0 - \tilde{Y}_0\|_\infty \|\tilde{X}\|_\infty.
\]

We obtain from (3.9) that
\[
\|Y_0 - \tilde{Y}_0\|_\infty \leq \|Y - \tilde{Y}\|_\infty = \|y - \tilde{y}\|_1
\]
\[
\leq \varepsilon \|y\|_1.
\]

Furthermore, we notice from (3.5), (3.8), and (3.9) that
\[
\|Y_0\|_\infty \leq \|y\|_1, \quad \|\tilde{X}\|_\infty = \|\tilde{x}\|_1 \leq (1 + \varepsilon)\|x\|_1,
\]
and
\[
\|X - \tilde{X}\|_\infty = \|x - \tilde{x}\|_1 \leq \varepsilon \|x\|_1.
\]
From (3.12)–(3.15), we can get
\[
\|Y_0X - \tilde{Y}_0\tilde{X}\|_\infty \leq \varepsilon \cdot \|x\|_1 \cdot \|y\|_1 + \varepsilon(1+\varepsilon)\|x\|_1 \cdot \|y\|_1 \\
= (\varepsilon + \varepsilon(1+\varepsilon))\|x\|_1 \cdot \|y\|_1.
\] (3.16)

Using the same trick, combining \|X_0\|_1 \leq \|x\|_1 + 1, for the second part of the right-hand side of (3.11),
we can prove that
\[
\|(X_0Y - \tilde{X}_0\tilde{Y})\|_\infty \leq 2\varepsilon \|x\|_1 \cdot \|y\|_1 + \varepsilon \|y\|_1
\] (3.17)
and (3.10) is obtained by combining (3.11), (3.16), and (3.17).

**Corollary 1** Under the assumptions and notations of Theorem 1, we have that
\[
\|H^{-1} - \tilde{H}^{-1}\|_2 \leq (4\varepsilon + \varepsilon^2) \cdot \|x\|_1 \cdot \|y\|_1 + \varepsilon \|y\|_1.
\] (3.18)

**Proof.** Recall that
\[
\|H^{-1} - \tilde{H}^{-1}\|_2 \leq \|H^{-1} - \tilde{H}^{-1}\|_1 \cdot \|\tilde{H}^{-1}\|_\infty.
\] (3.19)
The upper bound on \|H^{-1} - \tilde{H}^{-1}\|_\infty given by (3.10) is also an upper bound on \|H^{-1} - \tilde{H}^{-1}\|_1:
\[
\|H^{-1} - \tilde{H}^{-1}\|_1 \leq (4\varepsilon + \varepsilon^2) \cdot \|x\|_1 \cdot \|y\|_1 + \varepsilon \|y\|_1.
\] (3.20)
The proof is analogous to that of Theorem 1. So, we have from (3.19), (3.10), and (3.20) that
\[
\|H^{-1} - \tilde{H}^{-1}\|_2 \leq (4\varepsilon + \varepsilon^2) \cdot \|x\|_1 \cdot \|y\|_1 + \varepsilon \|y\|_1.
\] (3.21)

Denote \(H = A - \sigma B\). When the Hankel systems are solved approximately, the errors for the matrix-vector products can be represented as \(f_j = H^{-1}Bv_j - H^{-1}Bv_j, j = 1, 2, \ldots, m\). If we denote \(F_m = [f_1, f_2, \ldots, f_m]\),
we have the following relation for the \(m\)-step "inexact" shift-and-invert Arnoldi procedure:
\[
(A - \sigma B)^{-1}BV_m + F_m = ((A - \sigma B)^{-1}B + E)V_m = V_mH_m + h_{m+1,m}v_{m+1}e_m^T,
\] (3.22)
where \(V_m = [v_1, v_2, \ldots, v_m]\) is an \(n \times m\) orthonormal matrix, \(E = F_mV_m^T\), and \(H_m\) is an \(m \times m\) upper Hessenberg matrix. It should be noted that \(V_m\) and \(H_m\) are different from those in (2.2).

Let \((\tilde{\mu}, \tilde{w})\) be an eigenpair of \(G_m\), and let \(\tilde{\lambda} = 1/\tilde{\mu} + \sigma\). Denote by
\[
\hat{r}_{\text{real}} = AV_m\tilde{w} - \tilde{\lambda}BV_m\tilde{w}
\] (3.23)
the “real” residual with respect to the approximate eigenpair \((\tilde{\lambda}, V_m\tilde{w})\) of the matrix pencil \((A, B)\), and by
\[
r_{\text{real}} = (A - \sigma B)^{-1}BV_m\tilde{w} - \tilde{\mu}V_m\tilde{w},
\] (3.24)
and
\[
r_{\text{comp}} = [(A - \sigma B)^{-1}B + E)V_m\tilde{w} - \tilde{\mu}V_m\tilde{w}
\] (3.25)
the “real” and the “computed” residual for the approximate eigenpair \((\tilde{\mu}, V_m\tilde{w})\) of \((A - \sigma B)^{-1}B\), respectively.

Multiplying \((\tilde{\lambda} - \sigma)(A - \sigma B)\) on both sides of (3.24) yields
\[
(\tilde{\lambda} - \sigma)(A - \sigma B)r_{\text{real}} = (\tilde{\lambda} - \sigma)BV_m\tilde{w} - (A - \sigma B)V_m\tilde{w}
\]
As a result,
\[ \| \mathbf{r}_{\text{real}} \| \leq | \lambda - \sigma | \cdot \| A - \sigma B \| \cdot \| \mathbf{r}_{\text{real}} \|. \] (3.26)

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 (1.1) and investigate how to choose the stopping threshold \( \varepsilon \) for solving the Hankel systems (3.4). 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
\[ \varepsilon \leq \frac{\tilde{\text{tol}}}{4\sqrt{m} \cdot (\| \mathbf{x} \|_1 + \frac{1}{4}) \cdot \| \mathbf{y} \|_1 \cdot \| B \|_\infty}, \] (3.27)
then
\[ \| \mathbf{r}_{\text{real}} - \mathbf{r}_{\text{comp}} \|_\infty \lesssim \tilde{\text{tol}}, \] (3.28)
where \( m \) is the step of the shift-and-invert Arnoldi process.

**Proof.** By (3.10), we get
\[ \| \mathbf{f}_j \|_\infty = \| \tilde{H}^{-1} B\mathbf{v}_j - H^{-1} B\mathbf{v}_j \|_\infty \leq \| H^{-1} - \tilde{H}^{-1} \|_\infty \cdot \| B\mathbf{v}_j \|_\infty \]
\[ \leq \varepsilon (4 + \varepsilon) \| \mathbf{x} \|_1 \cdot \| \mathbf{y} \|_1 \cdot \| B\mathbf{v}_j \|_\infty \]
\[ = \varepsilon (4 + \varepsilon) \| \mathbf{x} \|_1 \cdot \| \mathbf{y} \|_1 \cdot \| B\mathbf{v}_j \|_\infty \]

If \( \varepsilon \ll 1 \), then
\[ \varepsilon (4 + \varepsilon) \| \mathbf{x} \|_1 + 1 \cdot \| \mathbf{y} \|_1 \cdot \| B\mathbf{v}_j \|_\infty \lesssim \varepsilon (4 + \varepsilon) \| \mathbf{x} \|_1 + 1 \cdot \| \mathbf{y} \|_1 \cdot \| B\mathbf{v}_j \|_\infty \]
\[ \leq 4 \left( \| \mathbf{x} \|_1 + \frac{1}{4} \right) \| \mathbf{y} \|_1 \cdot \| B \|_\infty \cdot \| \mathbf{v}_j \|_\infty \]
(3.29)

where we removed the high order term \( O(\varepsilon) \) and utilized the inequality \( \| \mathbf{v}_j \|_\infty \leq \| \mathbf{v}_j \|_2 = 1 \). Consequently, if
\[ \varepsilon \leq \frac{\tilde{\text{tol}}}{4\sqrt{m} \cdot (\| \mathbf{x} \|_1 + \frac{1}{4}) \cdot \| \mathbf{y} \|_1 \cdot \| B \|_\infty}, \]
i.e.,
\[ \| \mathbf{f}_j \|_1 \lesssim 4 \left( \| \mathbf{x} \|_1 + \frac{1}{4} \right) \| \mathbf{y} \|_1 \cdot \| B \|_\infty \cdot \varepsilon \leq \frac{\tilde{\text{tol}}}{\sqrt{m}}, \]
(3.30)
then we have from (3.24), (3.25) and (3.30) that
\[ \| \mathbf{r}_{\text{real}} - \mathbf{r}_{\text{comp}} \|_\infty = \| \mathbf{E} \mathbf{V}_m \mathbf{\tilde{w}} \|_\infty \]
\[ \leq \sum_{j=1}^{m} \| \mathbf{f}_j \|_\infty \cdot \| \mathbf{\tilde{w}}_j \|_1 \]
\[ \leq \max_{1 \leq j \leq m} \| \mathbf{f}_j \|_\infty \cdot \sqrt{m} \| \mathbf{\tilde{w}}_j \|_2 \lesssim \tilde{\text{tol}}, \]
where \( \mathbf{\tilde{w}} = [\mathbf{\tilde{w}}_1, \ldots, \mathbf{\tilde{w}}_m]^H \), and we utilized the inequality \( \| \mathbf{\tilde{w}}_1 \|_1 \leq \sqrt{m} \| \mathbf{\tilde{w}}_1 \|_2 = \sqrt{m}. \) □
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 (3.4). We can utilize the following inequality as the stopping criterion for the Hankel eigen-computation by (2.3):

$$\frac{\|A\tilde{x}_i - \tilde{\lambda}_i B\tilde{w}_i\|_2}{\|A - \sigma B\|_2} \leq h_{m+1,m} |\tilde{\lambda}_i - \sigma| \cdot |e_n^T \tilde{w}_i| \leq \text{tol}, \quad i = 1, 2, \ldots, \ell,$$

(3.31)

where $\text{tol}$ is the user-prescribed tolerance for the Hankel eigenproblem and $\ell$ is the required number of eigenpairs. Inspired by (3.26) and (3.31), let

$$\text{tol} = \|A - \sigma B\|_2 \cdot \text{tol},$$

then it follows from Theorem 2 that if

$$\varepsilon \leq \frac{\|A - \sigma B\|_2}{4\sqrt{m} \cdot (\|x\|_1 + \frac{1}{2}) \cdot \|y\|_1 \cdot \|B\|_{\infty} \cdot \text{tol}},$$

(3.32)

then we obtain that

$$\frac{\|r^{\text{real}} - r^{\text{comp}}\|_{\infty}}{\|A - \sigma B\|_2} \lesssim \text{tol}.$$

(3.33)

**Remark 1.** From equation (3.32), we observe that when

$$\xi \equiv \frac{1}{(\|x\|_1 + \frac{1}{2})\|y\|_1} \quad \text{and} \quad \eta \equiv \frac{\|A - \sigma B\|_2}{4\sqrt{m} \cdot \|B\|_{\infty}}$$

are small, it is necessary to solve the Hankel linear systems with a relatively high accuracy, as given in equation (3.4). Otherwise, we can solve them with a relatively low accuracy. Some parameters that appear in equation (3.32), such as $\|x\|_1$ and $\|y\|_1$, are not always available. If $\xi$ is not too small, we recommend using

$$\{\|r_z\|_2, \|r_y\|_2\} \leq \frac{\max(\|f_{\text{col}}\|_2, \|f_{\text{row}}\|_2)}{4\sqrt{m} \cdot \max(\|B_{\text{col}}\|_2, \|B_{\text{row}}\|_2)} \cdot \text{tol}$$

(3.34)

as the stopping criterion for solving the Hankel systems, where $r_z = p - H\tilde{x}$ and $r_y = e_n - H\tilde{y}$ are the residuals of the Hankel systems; $f_{\text{col}}$ and $f_{\text{row}}$ are the first column and the last row of $A - \sigma B$, respectively; and $B_{\text{col}}$ and $B_{\text{row}}$ are the first column and the last row of $B$, respectively. The effectiveness of this scheme is demonstrated in the numerical experiments in Section 4. Actually, the solution of $x$ of Hankel linear system $Hx = b$ can be obtained by solving $JHx = Jb$, where $JH$ is a Toeplitz matrix. Thus we can use the GMRES algorithm with T. Chan’s preconditioner [3, 15] for solving (3.4).

The implementation of the shift-and-invert Arnoldi process in practical computations is hindered 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 [19] or the thick-restarted Arnoldi algorithm [13, 26], 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 $\sigma$, a convergence threshold $\text{tol}$ for the eigenproblem, and four vectors $A_{\text{col}}, A_{\text{row}}, B_{\text{col}},$ and $B_{\text{row}}$, 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 (3.4) “inexactly” with the stopping criterion given in (3.34).

**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 [19] or the thick-restarted Arnoldi algorithm [13, 26].
4 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 an M1 chip that has a 3.20 GHz CPU and 8GB RAM, using a MATLAB 9.11.0 (R2021b) implementation with machine precision of $\epsilon \approx 2.22 \times 10^{-16}$. To solve the generalized Hankel eigensystems, we utilize the MATLAB built-in function `eigs.m`, which implements the implicitly restarted shift-and-invert Arnoldi algorithm, and redefine the matrix-vector product with the solution of the Hankel systems (3.4). 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 T. Chan’s preconditioner [3, 15] as the solver for (3.4). Given a convergence threshold $tol$, we use (3.34) as the stopping criterion for the Hankel systems. Specifically, we use

$$
\|M^{-1}b - M^{-1}(A - \sigma B)\tilde{q}\|_2 \leq \frac{\max(\|fcol\|_2, \|frow\|_2)}{4\sqrt{20} \cdot \max(\|Bcol\|_2,\|Brow\|_2)} \cdot tol, \quad (4.1)
$$

as the stopping criterion for the Hankel systems, where $M$ stands for T. Chan’s preconditioner, $\tilde{q} = \tilde{x}$ or $\tilde{y}$ is the approximate solution, and $b = p$ or $e_n$ 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 T. Chan’s preconditioner for (3.4). For the Hankel linear systems, the stopping criterion is:

$$
\|M^{-1}b - M^{-1}(A - \sigma B)\tilde{q}\|_2 \leq 10^{-14}, \quad (4.2)
$$

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, “(4.1)” indicates the value of the right-hand side of “(4.1)”, “CPU(sec.)” signifies the CPU time utilized in seconds, and “-” indicates that the algorithm is “out of memory.” Remember that our new algorithm does not precisely solve (3.4). To demonstrate the efficiency of the inexact approach, we provide the maximum “real” residual norm, cf. (3.23):

$$
\|\tilde{r}^{real}\|_2 = \max_{i=1,\ldots,\ell} \|A(V_m\tilde{w}_i) - \tilde{\lambda}_i B(V_m\tilde{w}_i)\|_2, \quad (4.3)
$$

where $\ell$ is the number of required eigenpairs and $(\tilde{\lambda}_i, V_m\tilde{w}_i), i = 1, 2, \ldots, \ell$, are Ritz pairs gained from running Inexact-eigs. Both “Inexact-eigs” and “Exact-eigs” are included inCPU time to solve Hankel systems and compute eigenpairs.

**Example 1.** The purpose of this example is to demonstrate the efficiency of the inexact strategy (4.1) for large generalized Hankel eigenproblems. We generate the Hankel matrices $\{h_k\}_{k=-n+1}^{n-1}$

$$
H = \{h_k\}_{k=-n+1}^{n-1} = \begin{bmatrix}
    h_{-(n-1)} & h_{-(n-2)} & \cdots & h_0 \\
    h_{-(n-2)} & h_{-(n-3)} & \cdots & h_1 \\
    \vdots & \vdots & \ddots & \vdots \\
    h_0 & h_1 & \cdots & h_{n-1}
\end{bmatrix},
$$

with generating functions $f$ are taken from [10]. We generate matrix $A$ using the function

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

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

Table 1 displays the residual, stopping criterion for solving the Hankel systems (3.4) 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 (3.4) is approximately \(O(10^{-7})\), whereas Exact-eigs uses a tight stopping criterion of \(10^{-14}\). As a result, the Inexact-eigs strategy significantly reduces the CPU time required compared to Exact-eigs, especially for larger system sizes.

<table>
<thead>
<tr>
<th>(n)</th>
<th>((4.1))</th>
<th>(|\hat{r}_{\text{real}}|_2)</th>
<th>CPU(sec.)</th>
<th>CPU(sec.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inexact-eigs</td>
<td>eigs(A,B): Exact-eigs</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3000</td>
<td>(1.347 \times 10^{-7})</td>
<td>(2.222 \times 10^{-10})</td>
<td>0.091</td>
<td>1.051</td>
</tr>
<tr>
<td>6000</td>
<td>(1.347 \times 10^{-7})</td>
<td>(5.496 \times 10^{-11})</td>
<td>0.386</td>
<td>6.534</td>
</tr>
<tr>
<td>9000</td>
<td>(1.347 \times 10^{-7})</td>
<td>(4.998 \times 10^{-11})</td>
<td>0.774</td>
<td>18.815</td>
</tr>
<tr>
<td>12000</td>
<td>(1.347 \times 10^{-7})</td>
<td>(3.543 \times 10^{-11})</td>
<td>1.289</td>
<td>43.550</td>
</tr>
<tr>
<td>15000</td>
<td>(1.347 \times 10^{-7})</td>
<td>(1.619 \times 10^{-11})</td>
<td>2.007</td>
<td>93.515</td>
</tr>
</tbody>
</table>

**Example 1, Table 1:** Numerical results of the algorithms for computing the 10 smallest eigenpairs in magnitude with \(\sigma = 0\) and \(tol = 10^{-6}, t = 1\).

Table 2 provides the 10 smallest eigenvalues obtained using the MATLAB built-in function \(\text{eig}(A, B)\), 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.

<table>
<thead>
<tr>
<th>(\text{eig}: \lambda_i, i = 1,2, \ldots, 10)</th>
<th>Inexact-eig</th>
<th>Exact-eig</th>
</tr>
</thead>
<tbody>
<tr>
<td>(5.42135073 \times 10^{-9}\pm)</td>
<td>(5.42135013 \times 10^{-9}\pm)</td>
<td>(5.42135166 \times 10^{-9}\pm)</td>
</tr>
<tr>
<td>(3.00211667 \times 10^{-9}i)</td>
<td>(3.00211663 \times 10^{-9}i)</td>
<td>(3.00211667 \times 10^{-9}i)</td>
</tr>
<tr>
<td>(5.64830095 \times 10^{-8}\pm)</td>
<td>(5.64829978 \times 10^{-8}\pm)</td>
<td>(5.64830753 \times 10^{-8}\pm)</td>
</tr>
<tr>
<td>(1.45117763 \times 10^{-5}i)</td>
<td>(1.45117761 \times 10^{-5}i)</td>
<td>(1.45117763 \times 10^{-5}i)</td>
</tr>
<tr>
<td>(2.08752894 \times 10^{-7}\pm)</td>
<td>(2.08752852 \times 10^{-7}\pm)</td>
<td>(2.08752895 \times 10^{-7}\pm)</td>
</tr>
<tr>
<td>(3.47985781 \times 10^{-5}i)</td>
<td>(3.47985777 \times 10^{-5}i)</td>
<td>(3.47985781 \times 10^{-5}i)</td>
</tr>
<tr>
<td>(5.18404010 \times 10^{-7}\pm)</td>
<td>(5.18403892 \times 10^{-7}\pm)</td>
<td>(5.18404009 \times 10^{-7}\pm)</td>
</tr>
<tr>
<td>(6.38568356 \times 10^{-5}i)</td>
<td>(6.38568348 \times 10^{-5}i)</td>
<td>(6.38568356 \times 10^{-5}i)</td>
</tr>
<tr>
<td>(1.04205274 \times 10^{-6}\pm)</td>
<td>(1.04205245 \times 10^{-6}\pm)</td>
<td>(1.04205273 \times 10^{-6}\pm)</td>
</tr>
<tr>
<td>(1.01686715 \times 10^{-4}i)</td>
<td>(1.01686713 \times 10^{-4}i)</td>
<td>(1.01686715 \times 10^{-4}i)</td>
</tr>
</tbody>
</table>

**Example 1, Table 2:** The 10 smallest eigenvalues in magnitude computed by using \(\text{eig}(A, B)\) and the approximations obtained from running Inexact-eig and Exact-eig on Example 1, \(n = 3000, t = 1\).

**Example 2.** The test matrices used in this example are derived from [14, 23]. We generate the Hankel matrix \(A\) using the even function \(\theta^2\) defined on \([0, \pi]\), which was introduced in Example 1. The Hankel
matrix $B = (b_{ij})$ is given by

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

which is derived from the even function $\theta^4 + 1$ limited to $[-\pi, \pi]$. We aim to compute the 10 smallest eigenpairs closest to $\sigma = 0$ for this test problem.

<table>
<thead>
<tr>
<th>$n$</th>
<th>$(4.1)$</th>
<th>$|\hat{r}^{real}|_2$</th>
<th>CPU(sec.)</th>
<th>CPU(sec.)</th>
<th>CPU(sec.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3000</td>
<td>$1.498 \times 10^{-7}$</td>
<td>$2.884 \times 10^{-8}$</td>
<td>0.037</td>
<td>0.148</td>
<td>1.351</td>
</tr>
<tr>
<td>6000</td>
<td>$1.498 \times 10^{-7}$</td>
<td>$8.843 \times 10^{-9}$</td>
<td>0.057</td>
<td>–</td>
<td>6.898</td>
</tr>
<tr>
<td>9000</td>
<td>$1.498 \times 10^{-7}$</td>
<td>$1.042 \times 10^{-7}$</td>
<td>0.087</td>
<td>–</td>
<td>18.892</td>
</tr>
<tr>
<td>12000</td>
<td>$1.498 \times 10^{-7}$</td>
<td>$7.159 \times 10^{-8}$</td>
<td>0.113</td>
<td>–</td>
<td>41.597</td>
</tr>
<tr>
<td>15000</td>
<td>$1.498 \times 10^{-7}$</td>
<td>$7.890 \times 10^{-9}$</td>
<td>0.140</td>
<td>–</td>
<td>189.374</td>
</tr>
</tbody>
</table>

Example 2, Table 3: Numerical results of the algorithms for computing the 10 smallest eigenpairs in magnitude with $\sigma = 0$ and $tol = 10^{-6}$; where “–” represents “out of memory”.

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 lists 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 the use of formula (3.5) for solving Hankel systems, which allows the Inexact-eigs scheme to use a much looser stopping criterion of $O(10^{-7})$ compared to the Exact-eigs scheme’s stopping criterion of $10^{-14}$. This improvement is significant. Furthermore, we observe that the Exact-eigs method fails for $n > 3000$, which may be due to the difficulty of solving the Hankel linear systems using the required accuracy of $O(10^{-14})$ for this example. This further illustrates the effectiveness of our “inexact” strategy.

5 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 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. Our “inexact” strategy is shown to outperform solving the Hankel systems “exactly”, especially when the Hankel systems are challenging to solve.

References


