Krylov Type Subspace Methods for Matrix Polynomials[1]

Leonard Hoffnung[2] Ren-Cang Li[3] Qiang Ye[4]

August 2002

Revised September 2003

ABSTRACT

We consider solving eigenvalue problems or model reduction problems for a quadratic matrix polynomial $I\lambda^2 - A\lambda - B$ with large and sparse A and B. We propose new Arnoldi and Lanczos type processes which operate on the same space as A and B live and construct projections of A and B to produce a quadratic matrix polynomial with the coefficient matrices of much smaller size, which is used to approximate the original problem. We shall apply the new processes to solve eigenvalue problems and model reductions of a second order linear input-output system and discuss convergence properties. Our new processes are also extendable to cover a general matrix polynomial of any degree.

[1]This report is available on the web at `http://www.ms.uky.edu/~rcli/`.

[2]Department of Mathematics, University of Kentucky, Lexington, KY 40506 (`lhoff@ms.uky.edu`.) Supported in part by the National Science Foundation under Grant No. CCR-9875201.

[3]Department of Mathematics, University of Kentucky, Lexington, KY 40506 (`rcli@ms.uky.edu`.) This work was supported in part by the National Science Foundation under Grant No. ACI-9721388 and by the National Science Foundation CAREER award under Grant No. CCR-9875201.

[4]Department of Mathematics, University of Kentucky, Lexington, KY 40506 (`qye@ms.uky.edu`.) Supported in part by the National Science Foundation under Grant No. CCR-0098133.

Krylov Type Subspace Methods for Matrix Polynomials

Leonard Hoffnung * Ren-Cang Li † Qiang Ye ‡

August 2002
Revised September 2003

Abstract

We consider solving eigenvalue problems or model reduction problems for a quadratic matrix polynomial $I\lambda^2 - A\lambda - B$ with large and sparse A and B. We propose new Arnoldi and Lanczos type processes which operate on the same space as A and B live and construct projections of A and B to produce a quadratic matrix polynomial with the coefficient matrices of much smaller size, which is used to approximate the original problem. We shall apply the new processes to solve eigenvalue problems and model reductions of a second order linear input-output system and discuss convergence properties. Our new processes are also extendable to cover a general matrix polynomial of any degree.

1 Introduction

Krylov subspace techniques are widely used for solving linear systems of equations and eigenvalue problems involving large and sparse matrices [8, 16]. It has found applications in many other large scale matrix problems such as model reductions of linear input-output systems [11, 13]. The basic idea of the techniques is to extract information of an $n \times n$ matrix A most relevant to the underlying computational problem through utilizing the so-call *Krylov* subspace

$$\mathcal{K}_k(A, v) = \operatorname{span}\{v, Av, \ldots, A^{k-1}v\},$$

or through utilizing two (row and column) Krylov subspaces

$$\mathcal{K}_k(A, v), \quad \mathcal{K}_k(A^*, w)$$

simultaneously, where v and w are vectors of dimension n, the asterisk denotes the conjugate transpose. This is realized by either the *Arnoldi (or Lanczos) process* when

*Department of Mathematics, University of Kentucky, Lexington, KY 40506 (`lhoff@ms.uky.edu`.) Supported in part by the National Science Foundation under Grant No. CCR-9875201.

†Department of Mathematics, University of Kentucky, Lexington, KY 40506 (`rcli@ms.uky.edu`.) Supported in part by the National Science Foundation under Grant No. ACI-9721388 and by the National Science Foundation CAREER award under Grant No. CCR-9875201.

‡Department of Mathematics, University of Kentucky, Lexington, KY 40506 (`qye@ms.uky.edu`.) Supported in part by the National Science Foundation under Grant No. CCR-0098133.

only $\mathcal{K}_k(A, v)$ is employed or by the *nonsymmetric Lanczos process* if both $\mathcal{K}_k(A, v)$ and $\mathcal{K}_k(A^*, w)$ are used [1, 19]. See also [8, 16, 24, 28, 29].

When $\mathcal{K}_k(A, v)$ has dimension k, the Arnoldi process generates an orthonormal basis $\{q_1, q_2, \ldots, q_k\}$ for $\mathcal{K}_k(A, v)$, and an upper Hessenberg matrix $H_k = Q_k^* A Q_k$, which is the projection of A onto $\mathcal{K}_k(A, v)$, where $Q_k = [q_1, q_2, \ldots, q_k]$. On the other hand, the Lanczos process generates a basis for $\mathcal{K}_k(A, v)$ and a basis for $\mathcal{K}_k(A^*, w)$ such that the two bases are biorthogonal. Simultaneously, a tridiagonal matrix T_k is obtained, which is the projection of A onto $\mathcal{K}_k(A, v)$ along $\mathcal{K}_k(A^*, w)$. For Hermitian A, usually w is taken to be v and the process coincides with the Arnoldi process and is called the (symmetric) Lanczos process.

For modest $k \ll n$, some of the eigenvalues of H_k (and T_k) are good approximations to some eigenvalues (usually extreme part) of A. This approximation of A by H_k (and T_k) can be used in many other applications as well, such as model reductions of linear input-output systems [6, 11, 13]. Over the years, many technical inventions, including a shift-and-invert strategy, look-ahead techniques, rational Krylov, (adaptive) block versions, implicit restart strategies, and so on, have been developed for the Lanczos/Arnoldi algorithm for better numerical efficiency and stability. But we shall not discuss them here, see for example [3, 4, 10, 17, 18, 25, 27, 28, 31, 35] and references therein.

In this paper, we consider related problem for a large and sparse $n \times n$ *monic matrix polynomial*

$$A(\lambda) = I_n \lambda^m - \sum_{i=0}^{m-1} A_i \lambda^i, \tag{1.1}$$

where I_n is the $n \times n$ identity matrix (later we may simply write I if its dimension is clear from the context.). (1.1) is typically associated with an m-th order linear system of ordinary differential equations:

$$x^{(m)}(t) - \sum_{i=0}^{m-1} A_i x^{(i)}(t) = g(t), \tag{1.2}$$

with appropriate initial values for $x(t)$ and its derivatives up to order $m-1$ at $t = 0$, where A_i is an $n \times n$ constant matrix and $x(t)$ is a vector of dimension n, depending on t. A modal analysis of (1.2) is to find those scalars μ and nonzero vectors x and/or y such that

$$A(\mu)x = 0, \quad y^* A(\mu) = 0.$$

μ is called an eigenvalue of (1.1) and x (and y) a right (and left, resp.) eigenvector. On the other hand, in a linear input-output system with the state governed by (1.2), we are interested in approximations and computations of the transfer function

$$f(s) = c^* A(s)^{-1} b. \tag{1.3}$$

Specifically, we would like to find a lower dimensional linear input-output system whose input-output relations (i.e. the transfer function) gives a good approximation to those of the given system. This is often referred to as model reductions.

In most cases, a problem (e.g. eigenvalue problem) concerning the matrix polynomial can be reduced to one for the following $nm \times nm$ matrix [15]

$$A_{\text{LIN}} = \begin{pmatrix} 0 & I & 0 & \cdots & 0 \\ 0 & 0 & I & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & I \\ A_0 & A_1 & A_2 & \cdots & A_{m-1} \end{pmatrix},$$

to which well-established methods can be applied. This is called *linearization*. For the eigenvalue problem or the model reduction problem, one can use the Arnoldi or the Lanczos algorithm on A_{LIN} to produce a Krylov subspace and then a projection, which is used to approximate A_{LIN}. Such a process will have to operate with vectors of dimension mn and therefore may substantially increase the computational cost. Furthermore, the projection of A_{LIN} on a Krylov subspace is usually not a linearization of any matrix polynomial and thus the approximation as obtained loses its intrinsic physical connection to the original problem. In model reductions, for example, a consequence of this is that the reduced model that is obtained by applying the Arnoldi or the Lanczos process to the linearization problem A_{LIN} cannot be synthesized with a physical model of an mth order input-output system [2].

In this paper, we study extensions of the *standard Arnoldi process* and the *standard Lanczos process* for matrix polynomials without going through any linearization. We note that several other methods [22, 30] have been developed that do not rely on the linearization processes (see also [4, 32]). Here, we develop Krylov type projection methods that generate a basis $\{q_1, q_2, \ldots, q_k\}$ for a subspace as defined by A_i and its powers, and then apply projection simultaneously to each matrix A_i of the matrix polynomial to obtain $H_k^{(i)} = Q_k^* A_i Q_k$ which is also in some condensed form, where $Q_k = [q_1, q_2, \ldots, q_k]$. Then we approximate $A(\lambda)$ by the lower dimensional matrix polynomial

$$H_k(\lambda) \equiv I_k \lambda^m - \sum_{i=0}^{m-1} H_k^{(i)} \lambda^i.$$

Compared with the linearization, this approach has advantages of preserving certain properties of the original system $A(\lambda)$ such as

- If the field of values of $A(\lambda)$ (i.e. $\mathcal{F} = \{\lambda : x^* A(\lambda) x = 0 \text{ for some } x \neq 0\}$) is on the left half complex plane, which guarantees stability of the system (1.2), then the field of values of $H_k(\lambda)$ is also on the left half plane, preserving the stability.
- The property that A_i is symmetric, positive definite, etc. is preserved by $H_k^{(i)}$. In particular, if $A(\lambda)$ is an overdamped vibrating system [9, 15], so is $H_k(\lambda)$. So the property that all eigenvalues are real is preserved. The same is true if $A(\lambda)$ comes from a weakly damped system, which has the eigenvalues near the imaginary axis.
- A gyroscopic system about a stable equilibrium [9] has $A_0 > 0$ and A_1 skew-hermitian with $m = 2$. In this case, the eigenvalues are all pure imaginary. Again, the projection problem preserves this property.

An Arnoldi process of this type has already been developed recently for a monic quadratic polynomial [21], where a special case is also considered in which a linear combination of the two matrices is of low rank. This paper, as a continuation of [21], will first investigate the subspaces associated with the Arnoldi type process, especially for the commutable case. This is done in Section 3. In Section 4, we develop a Lanczos type process for monic quadratic polynomials. Briefly analogous extensions to a general monic polynomial of degree m are mentioned. The uses of the Arnoldi/Lanczos type processes for model reductions and eigenvalue computations, and their convergent behaviors are given in Sections 5 and 6. We shall also present a numerical example to illustrate our new algorithms in Section 7. Throughout the paper, however, we will be focusing mostly on explaining the new ideas and leave subtle but important numerical considerations and numerical applications (e.g., more extensive numerical testing) to future studies. We also note that our focus on monic matrix polynomials does not lose any generality because non-monic cases can be handled implicitly as a monic one through some kind of combination of shifting and a factorization of the leading matrix.

Notation. The jth column of an identity matrix (of size that will be clear from the context) is denoted as e_j. $\mathbb{C}^{m\times n}$ is the set of all m-by-n complex matrices, $\mathbb{C}^n = \mathbb{C}^{n\times 1}$ (vectors), and $\mathbb{C} = \mathbb{C}^1$ (scalars). We shall use MATLAB-like notation $X_{(i:j,k:\ell)}$ to denote the submatrix of X, consisting of the intersections of rows i to j and columns k to ℓ, and when $i : j$ is replaced by :, it means all rows, similarly for columns. The generic notation $\mathtt{x}$ is for a possible nonzero scalar, vector, and $\mathtt{X}$ for a possible nonzero matrix. $\|x\|_2$ is the Euclidean norm of a vector x and $\|X\|_2$ is the spectral norm of a matrix X. Given a real number γ, $\lfloor\gamma\rfloor$ is the largest integer that is not greater than γ.

2 Arnoldi/Lanczos Process for Matrices – Review

We first give a brief description of the standard Arnoldi and Lanczos algorithms for a matrix A, and at the same time set our other notation. The Arnoldi and Lanczos Processes yield partial realizations of the following decompositions of a matrix A.

- Given $q_1 \in \mathbb{C}^n$ with $\|q_1\|_2 = 1$, there is a unitary matrix $Q \in \mathbb{C}^{n\times n}$ with $Qe_1 = q_1$ such that $Q^*AQ = H \equiv (h_{ij})$ is upper Hessenberg, i.e., $h_{ij} = 0$ for $i > j + 1$. This can be proved by Hessenberg Reduction, the first step in computing the eigensystem of a (dense) matrix by QR method [8, p.164].
- Given $v_1, w_1 \in \mathbb{C}^n$ such that $w_1^* v_1 = 1$, (unless there is a breakdown[1]) there is a matrix $V \in \mathbb{C}^{n\times n}$ with $Ve_1 = v_1$ and $V^{-*}e_1 = w_1$ such that $V^{-1}AV = T \equiv (t_{ij})$ is tridiagonal, i.e., $t_{ij} = 0$ for $i > j + 1$ or $i + 1 < j$. A similar proof to that of Theorem 4.1 below can be given (see also [14, 23]).

The Arnoldi process starts with q_1 and then looks at $AQ = QH$ one column at a time from the first column onwards to achieve its goal. The following algorithm presents a basic

[1] *Breakdown*, though not defined yet, is intrinsic to the nonsymmetric Lanczos process. A good reference is [25]. See also the paragraph immediately followed Algorithm 2.2.

version of it.

Algorithm 2.1 Standard Arnoldi Process:

> **Given** q_1 **with** $\|q_1\|_2 = 1$**;**
> **for** $j = 1, 2, \ldots, k$ **do**
> $\hat{q} = Aq_j$;
>   **for** $i = 1, 2, \ldots j$ **do**
> $h_{ij} = q_i^* \hat{q}$; $\hat{q} = \hat{q} - q_i h_{ij}$;
>   **end for**
> $h_{j+1,j} = \|\hat{q}\|_2$;
>   **if** $h_{j+1\,j} > 0$ **then**
> $q_{j+1} = \hat{q}/h_{j+1\,j}$;
>   **else**
>   **BREAK**
>   **end if**
> **end for**

If this algorithm concludes after the j-loop finishes at the end for $j = n$, the decomposition described above is realized; if it finishes by **BREAK**ing out, an invariant subspace of A is computed.

Similarly, the Lanczos process is based on the second decomposition and starts with two vectors v_1 and w_1 with $w_1^* v_1 = 1$. The following algorithm presents a basic version of it.

Algorithm 2.2 Standard Non-symmetric Lanczos Process:

> **Given** v_1 **and** w_1 **such that** $w_1^* v_1 = 1$**;**
> **Set** $\beta_0 = \gamma_0 \equiv 0$**;** $v_0 = w_0 \equiv 0$**;**
> **for** $j = 1, 2, \ldots, j$ **do**
> $\alpha_j = w_j^* A v_j$;
> $\hat{v} = Av_j - \alpha_j v_j - \beta_{j-1} v_{j-1}$;
> $\hat{w} = A^* w_j - \bar{\alpha}_j w_j - \bar{\gamma}_{j-1} w_{j-1}$;
> $\gamma_j = \sqrt{|\hat{w}^* \hat{v}|}$;
>   **if** $\gamma_j = 0$ **then BREAK;**
> $\beta_j = \hat{w}^* \hat{v} / \gamma_j$;
> $v_{j+1} = \hat{v}/\gamma_j$, $w_{j+1} = \hat{w}/\bar{\beta}_j$;
> **end for**

A *breakdown* occurs in Algorithm 2.2 whenever a zero γ_j is encountered. There are two kinds of breakdowns: 1) $\gamma_j = 0$ is caused by either $\hat{v} = 0$ or $\hat{w} = 0$; 2) $\gamma_j = 0$ is due to $\hat{v} \perp \hat{w}$ but where neither $\hat{v}$ nor $\hat{w}$ is a zero vector. The first kind is a benign and welcome situation since it signals that an invariant subspace of A has been found; while the second one is more serious, see [12, 25, 35].

3 Arnoldi Type Process for Monic Matrix Polynomials

In this section, we shall first describe the Arnoldi type process for $I\lambda^2 - A\lambda - B$ derived in [21], and then study the Krylov type subspace that it computes. In particular, we discuss the special case when A and B commute. We shall also show how this can be generalized to a general m-th degree monic matrix polynomial.

3.1 Arnoldi type process for $I\lambda^2 - A\lambda - B$

The Arnoldi algorithm for $I\lambda^2 - A\lambda - B$ of [21] is based on the fact that given $q_1 \in \mathbb{C}^n$ with $\|q_1\|_2 = 1$, there is a unitary matrix $Q \in \mathbb{C}^{n\times n}$ with $Qe_1 = q_1$ such that[2]

$$Q^*AQ = H_a \equiv (h_{a;ij}), \quad Q^*BQ = H_b \equiv (h_{b;ij}) \tag{3.1}$$

satisfying

$$h_{a;ij} = 0 \text{ for } i \geq 2j+1, \quad h_{b;ij} = 0 \text{ for } i \geq 2j+2. \tag{3.2}$$

From this, the following algorithm is derived in [21].

Algorithm 3.1 Arnoldi Type Process :

1. **Given** q_1 **with** $\|q_1\|_2 = 1$**;**
2. $N = 1$;
3. **for** $j = 1, 2, \ldots, k$ **do**
4.   **if** $j > N$ **then BREAK;**
5. $\hat{q} = Aq_j$;
6.   **for** $i = 1, 2, \ldots N$ **do**
7. $h_{a;ij} = q_i^*\hat{q}$; $\hat{q} = \hat{q} - q_i h_{a;ij}$;
8.   **end for**
9. $h_{a;N+1,j} = \|\hat{q}\|_2$;
10.   **if** $h_{a;N+1,j} > 0$ **then**
11. $N = N + 1$, $q_N = \hat{q}/h_{a;Nj}$;
12.   **end if**
13. $\hat{q} = Bq_j$;
14.   **for** $i = 1, 2, \ldots N$ **do**
15. $h_{b;ij} = q_i^*\hat{q}$; $\hat{q} = \hat{q} - q_i h_{b;ij}$;
16.   **end for**
17. $h_{b;N+1,j} = \|\hat{q}\|_2$;
18.   **If** $h_{b;N+1,j} > 0$ **then**
19. $N = N + 1$; $q_N = \hat{q}/h_{a;N\,j}$;
20.   **end if**
21. **end for**

In the algorithm, N tracks the number of vectors q_i already generated at any given point. Let α_k and β_k be the values of N at the ends of Lines 12 and 20 respectively for

[2] The entries of H_a and H_b are unfortunately heavily subscripted a little bit – with a lower case letter before the separator ";" indicating the association to A or B, and with two integers i and j as their row and column positions. For better readability, sometimes we insert a comma between the two integers.

the trip $j = k$. It can be seen that $\alpha_k \le \beta_k \le \alpha_k + 1$. Upon completion of the above process, we have (see [21])

$$AQ_{(:,1:k)} = Q_{(:,1:\alpha_k)} H_{a\,(1:\alpha_k,1:k)}, \quad (3.3)$$
$$BQ_{(:,1:k)} = Q_{(:,1:\beta_k)} H_{b\,(1:\beta_k,1:k)}, \quad (3.4)$$

unless the j-loop is forced to **BREAK** out at Line 4, in which case,

$$AQ_{(:,1:N)} = Q_{(:,1:N)} H_{a\,(1:N,1:N)}, \quad (3.5)$$
$$BQ_{(:,1:N)} = Q_{(:,1:N)} H_{b\,(1:N,1:N)}. \quad (3.6)$$

While H_a and H_b are lower banded, their lower bandwidths increase quickly. In [21], the special case that a linear combination of A and B is of low rank is considered. In that case, the lower bandwidth of H_a and H_b is bounded by the rank of the combination plus 1. Later, we shall consider a special case when A and B commute. As we shall see, the lower bandwidth can also be significantly reduced in this case. We first need to describe the subspace $\text{span}\{q_1, q_2, \ldots, q_\ell\}$ in terms of q_1, A, and B.

3.2 Subspace $\text{span}\{q_1, q_2, \ldots, q_\ell\}$

Suppose first that in line 10 and 18 of Algorithm 3.1, all

$$h_{a;N+1,j} > 0, \quad h_{b;N+1,j} > 0.$$

We notice that the process starts with q_1, and at $j = 1$ it generates new directions in vectors Aq_1 first and then Bq_1; at $j = 2$, new directions in vectors A^2q_1 first and then BAq_1; at $j = 3$ new directions in vectors ABq_1 and then B^2q_1; and so on. The following table displays new vectors that expand the same subspace as the vectors generated at step j.

j	1	2	3	4	5	6	7
new vectors	Aq_1 Bq_1	A^2q_1 BAq_1	ABq_1 B^2q_1	A^3q_1 BA^2q_1	$ABAq_1$ B^2Aq_1	A^2Bq_1 $BABq_1$	AB^2q_1 B^3q_1

We list those vectors in the order of their first appearances as

$$\begin{array}{ll} \text{Group 0:} & q_1, \\ \text{Group 1:} & Aq_1, Bq_1, \\ \text{Group 2:} & A^2q_1, BAq_1, ABq_1, B^2q_1, \\ \text{Group 3:} & A^3q_1, BA^2q_1, ABAq_1, B^2Aq_1, A^2Bq_1, BABq_1, AB^2q_1, B^3q_1, \\ \vdots & \vdots \end{array} \quad (3.7)$$

They are produced in the order from top downwards and from left to right. Notice that we divide them naturally into Groups with Group 0 having only q_1, and Group t having 2^t vectors in the forms

$$X_t \cdots X_2 X_1 q_1, \quad X_i \in \{A, B\}.$$

The rule that governs the ordering in (3.7) is

$$X_s \cdots X_2 X_1 q_1 \text{ appears before } Y_t \cdots Y_2 Y_1 q_1 \text{ in (3.7) if } s < t \text{ or if there is an } j\ (1 \le j \le t) \text{ such that } X_i = Y_i \text{ for } 1 \le i \le j-1,\ X_j = A, \text{ and } Y_j = B \text{ when } s = t. \tag{3.8}$$

Define

$$\mathrm{g}\mathcal{K}_\ell(\{A, B\}, q_1) \tag{3.9}$$

to be the subspace spanned by the first ℓ vectors in (3.7) equipped with the ordering just described. We call it a *generalized Krylov subspace*. Then

$$\mathrm{span}\{q_1, q_2, \ldots, q_\ell\} = \mathrm{g}\mathcal{K}_\ell(\{A, B\}, q_1).$$

Now, if $h_{a;N+1,j} = 0$ (or $h_{b;N+1,j} = 0$), the first $N+1$ vectors in (3.7) are linearly dependent. In that case q_{N+1} is constructed by using the next linearly independent vector, say the ℓth vector, in the sequence (3.7). At that point, the number of linearly independent vectors in the first ℓ vectors is exactly $N + 1$. This leads to the following general result.

Theorem 3.1 *1. If* $\dim \mathrm{g}\mathcal{K}_\ell(\{A, B\}, q_1) = N$*, then*

$$\mathrm{span}\{q_1, q_2, \ldots, q_N\} = \mathrm{g}\mathcal{K}_\ell(\{A, B\}, q_1).$$

2. If the j*-loop of* Algorithm 3.1 *runs to its completion, then*

$$\mathrm{span}\{q_1, q_2, \ldots, q_N\} = \mathrm{g}\mathcal{K}_{2k+1}(\{A, B\}, q_1).$$

3. If Algorithm 3.1 *concludes by* **BREAK***ing out at* Line 4*, then*

$$\mathrm{span}\{q_1, q_2, \ldots, q_N\} = \mathrm{g}\mathcal{K}_{2j-1}(\{A, B\}, q_1) = \mathrm{g}\mathcal{K}_{2j}(\{A, B\}, q_1) = \cdots.$$

Proof: The first two claims are rather obvious. We shall now prove the third one. We have

$$Aq_j = \sum_{i=1}^{\alpha_j} q_i h_{a;ij}$$

and

$$Bq_j = \sum_{i=1}^{\beta_j} q_i h_{b;ij}$$

When the **BREAK**ing out occurs, $\beta_j = j$ as β_j is the value of N. Together with $\alpha_j \le \beta_j$, this shows that $\mathrm{span}\{q_1, q_2, \ldots, q_N\}$ is an invariant subspace for both A and B and is the same as $\mathrm{g}\mathcal{K}_{2j-1}(\{A, B\}, q_1)$. The vectors from the $2j$th onwards in (3.7) are linear combinations of vectors from the first to the $(2j-1)$-th multiplied by sequences of A and/or B and thus fall into $\mathrm{g}\mathcal{K}_{2j-1}(\{A, B\}, q_1)$. ∎

3.3 The Case When A and B Commute

When A and B commute, the reduced H_a and H_b will have much fewer nonzero entries below the diagonal than those for the general case in (3.1). We note that, for the eigenvalue problem $I\lambda^2 - A\lambda - B$, A and B share the same invariant subspace and thus we can just run the standard Arnoldi/Lanczos process on either A or B and then solve the quadratic eigenvalue problem. However, for other problems like the reduced order modeling to evaluate $q_1^*(I - As - Bs^2)^{-1}q_1$, the new processes may still be of interest. This subsection also shows that some inherent relations between A and B will have interesting effects on our new Arnoldi type process (and the Lanczos type process later in this paper).

The commutativity between A and B implies that some vectors in (3.7) appears multiple times, i.e., $BAq_1 = ABq_1$. In fact, Group t which has 2^t vectors effectively consists of $t+1$ vectors

$$A^t q_1, BA^{t-1}q_1, \ldots, B^{t-1}Aq_1, B^t q_1$$

in the generic situation. It can be seen that the generated basis vectors

$$q_1, q_2, q_3, \ldots$$

correspond to the sequence

$$\begin{array}{lc} \text{Group 0:} & q_1, \\ \text{Group 1:} & Aq_1, Bq_1, \\ \text{Group 2:} & A^2q_1, BAq_1, B^2q_1, \\ \text{Group 3:} & A^3q_1, BA^2q_1, B^2Aq_1, B^3q_1, \\ \vdots & \vdots \end{array} \tag{3.10}$$

in the sense that the new direction in, e.g., q_5 is from BAq_1. We shall use "$\sim$" to indicate such a correspondence, e.g., $q_5 \sim BAq_1$, $q_8 \sim BA^2q_1$, and so on. We would like to know the nonzero patterns in the generated H_a and H_b. It suffices for us to look at Aq_j and Bq_j and find out the first positions at which the vectors in (3.10) bring out the same new directions as Aq_j and Bq_j do for expanding the generalized Krylov subspace. First we need to know the corresponding position for q_j in (3.10). To this end, let integers s and t be such that

$$j = t(t+1)/2 + s \quad \text{for some } 0 < s \le t+1,$$

i.e., q_j belongs to Group t in (3.10), and $q_j \sim B^{s-1}A^{t-(s-1)}q_1$. Thus

$$Aq_j \sim B^{s-1}A^{t+1-(s-1)}q_1 \sim q_{(t+1)(t+2)/2+s} = q_{j+t+1}, \tag{3.11}$$

$$Bq_j \sim B^s A^{t+1-s}q_1 \sim q_{(t+1)(t+2)/2+s+1} = q_{j+t+2}. \tag{3.12}$$

So the jth column of H_a has nonzero entries from position 1 to $j+t+1$ and the jth column of H_b has nonzero entries from position 1 to $j+t+2$. Figure 1 shows the structures of H_a and H_b for the commutative case.

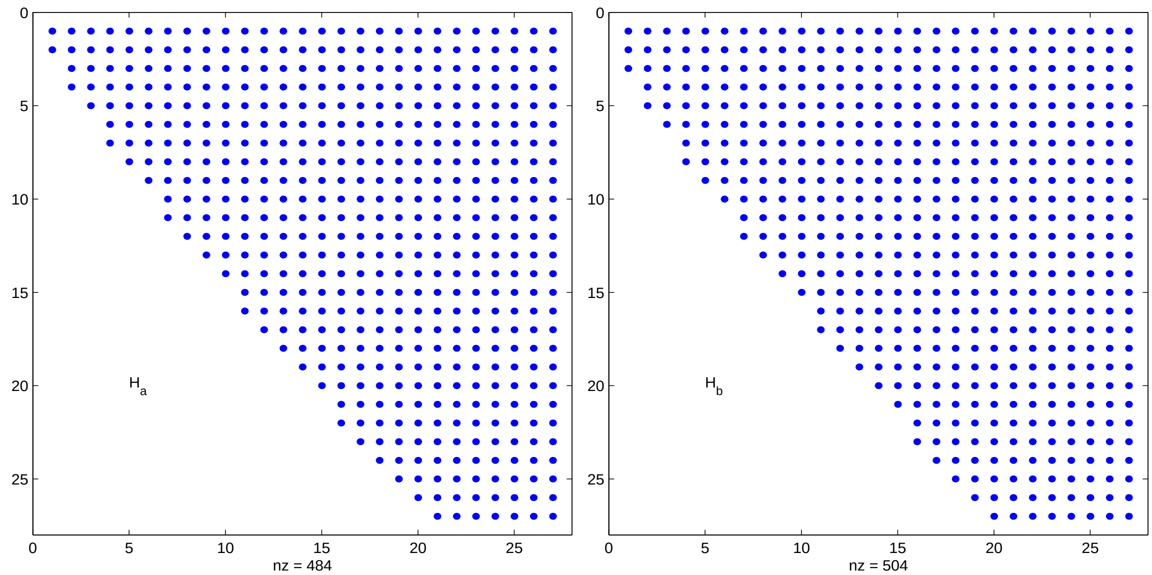

Figure 1: **The sparsity patterns of H_a and H_b: the commutative case**

Theorem 3.2 *Given $q_1 \in \mathbb{C}^n$ with $\|q_1\|_2 = 1$, suppose that A and B commute and that the first n vectors in (3.10) are linearly independent. Then there is a unitary matrix $Q \in \mathbb{C}^{n\times n}$ with $Qe_1 = q_1$ such that*

$$Q^*AQ = H_a \equiv (h_{a;ij}), \quad Q^*BQ = H_b \equiv (h_{b;ij}) \tag{3.13}$$

satisfy

$$h_{a;ij} = 0 \text{ for } i \geq j+t+2, \quad h_{b;ij} = 0 \text{ for } i \geq j+t+3, \tag{3.14}$$

where t is the unique integer such that $t(t+1)/2 + (t+1) \geq j > t(t+1)/2$.

It is of interest to compare nonzero patterns for H_a and H_b here with those for H_a and H_b in (3.1) for general A and B, where there are about j nonzero entries below the diagonal entries in the jth columns. Theorem 3.2, however, says when A and B commute there are about $\sqrt{2j}$ nonzero entries below the diagonal entries since $t \approx \sqrt{2j}$ for large j.

It is of an independent interest to see how many trips to the j-loop in Algorithm 4.2 for commutative A and B must be made in order to produce an orthonormal basis for the entire space $\mathbb{C}^n$, assuming that the first n vectors in (3.10) are linearly independent. (Recall that in the generic case and noncommutative A and B, $(n-1)/2$ trips are enough.) To this end, by (3.11) and (3.12) we need to find the minimal j so that

$$j+t+1 = n \text{ or } j+t+2 = n, \text{ subject to } j = t(t+1)/2 + s \text{ and } 1 \leq s \leq t+1,$$

or equivalently $t(t+1)/2 + 1 \leq j \leq t(t+1)/2 + t + 1$. Write $j + t + 2 = \hat{n}$, where $\hat{n} = n$ or $n+1$ as needed. Then

$$t(t+1)/2 + t + 3 \leq \hat{n} \leq t(t+1)/2 + 2t + 3 \quad \Leftrightarrow \quad t^2 + 3t + 6 - 2\hat{n} \leq 0 \leq t^2 + 5t + 6 - 2\hat{n}$$

which gives

$$\frac{-5+\sqrt{8\hat{n}+1}}{2} \le t \le \frac{-3+\sqrt{8\hat{n}-15}}{2}.$$

Thus

$$\hat{n}-\frac{1}{2}-\frac{\sqrt{8\hat{n}-15}}{2} \le j \le \hat{n}+\frac{1}{2}-\frac{\sqrt{8\hat{n}+1}}{2}.$$

For example when $n = 1000$, this requires $j = 955$. Asymptotically it needs n steps.

3.4 General monic matrix polynomial

The Arnoldi type process developed for A and B can be extended to a general monic matrix polynomial (1.1) of degree m. Recall the theoretical backbone of the algorithm is the decomposition (3.1). So we shall just state a corresponding decomposition result for $A(\lambda)$ but omit a detailed statement of an algorithm. The actual algorithm follows readily from this.

Theorem 3.3 *Let $A_\ell \in \mathbb{C}^{n\times n}$, $0 \le \ell \le m-1$. Given $q_1 \in \mathbb{C}^n$ with $\|q_1\|_2 = 1$, there is a unitary matrix $Q \in \mathbb{C}^{n\times n}$ with $Qe_1 = q_1$ such that*

$$Q^* A_\ell Q = H_\ell \equiv (h_{\ell;ij}), \quad \text{for } 0 \le \ell \le m-1 \tag{3.15}$$

satisfying

$$h_{\ell;ij} = 0 \text{ for } i \ge mj + m - \ell. \tag{3.16}$$

4 Lanczos Type Process for Monic Quadratic Matrix Polynomials

In this section, we develop a Lanczos type processes in parallel to the Arnoldi type process presented in the previous section. We shall present the derivation for monic quadratic matrix polynomial $I\lambda^2 - A\lambda - B$, and indicate a generalization to general m-th degree monic matrix polynomials.

4.1 Decomposition Theorem

The following lemma is essentially in [33] but not explicitly stated. It is in [5] for real vectors x and y but can be easily extended for complex vectors. For a more general version when x and y are replaced by tall (real or complex) matrices, see [20].

Lemma 4.1 *Let $x, y \in \mathbb{C}^n$. If $x^* y \ne 0$, then there exists nonsingular $V \in \mathbb{C}^{n\times n}$ such that*

$$V^{-1} y = \alpha e_1, \quad V^* x = \gamma e_1,$$

and $x^* y = \bar{\gamma}\alpha$.

Lemma 4.2 *Unless there is a breakdown which will be made clear in the proof, there is a nonsingular matrix* $V \in \mathbb{C}^{n\times n}$ *with* $Ve_1 = e_1$ *and* $V^{-*}e_1 = e_1$ *such that*

$$V^{-1}AV = T_a \equiv (t_{a;ij}), \quad V^{-1}BV = T_b \equiv (t_{b;ij})$$

satisfy

$$t_{a;ij} = 0 \text{ for } i \geq 2j+1 \text{ or } j \geq 2i+1, \quad t_{b;ij} = 0 \text{ for } i \geq 2j+2 \text{ or } j \geq 2i+2.$$

Proof: Our proof is constructive. It goes as follows. Partition

$$A = \begin{array}{c} \\ 1 \\ n-1 \end{array} \begin{array}{c} \begin{array}{cc} 1 & n-1 \end{array} \\ \begin{pmatrix} a_{11} & c_1^* \\ a_1 & \mathtt{X} \end{pmatrix} \end{array}.$$

A breakdown occurs if $c_1^* a_1 = 0$; otherwise by Lemma 4.1, we find a $\widehat{V}_{1a} \in \mathbb{C}^{(n-1)\times(n-1)}$ such that $\widehat{V}_{1a}^* c_1 = \gamma_1 e_1$ and $\widehat{V}_{1a}^{-1} a_1 = \alpha_1 e_1$. Let $V_{1a} = \mathrm{diag}(1, \widehat{V}_{1a})$. We have

$$V_{1a}^{-1} A V_{1a} = \left(\begin{array}{c|cc} a_{11} & \bar{\gamma}_1 & 0 \\ \hline \alpha_1 & \multicolumn{2}{c}{\mathtt{X}} \\ 0 & & \end{array}\right), \quad V_{1a}^{-1} B V_{1a} = \begin{array}{c} \\ 1 \\ 1 \\ n-2 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 1 & n-2 \end{array} \\ \begin{pmatrix} b_{11} & b_{12} & f_1^* \\ b_{21} & b_{22} & \mathtt{x} \\ b_1 & \mathtt{x} & \mathtt{X} \end{pmatrix} \end{array}.$$

Another breakdown occurs if $f_1^* b_1 = 0$; otherwise by Lemma 4.1, we find a $\widehat{V}_{1b} \in \mathbb{C}^{(n-2)\times(n-2)}$ such that $\widehat{V}_{1b}^* f_1 = \delta_1 e_1$ and $\widehat{V}_{1b}^{-1} b_1 = \beta_1 e_1$. Let $V_{1b} = \mathrm{diag}(I_2, \widehat{V}_{1b})$ and $V_1 \stackrel{\text{def}}{=} V_{1a}V_{1b}$. We have

$$V_1^{-1} A V_1 = \left(\begin{array}{c|cc} a_{11} & \bar{\gamma}_1 & 0 \\ \hline \alpha_1 & \multicolumn{2}{c}{\mathtt{X}} \\ 0 & & \end{array}\right), \quad V_1^{-1} B V_1 = \left(\begin{array}{c|c|cc} b_{11} & b_{12} & \bar{\delta}_1 & 0 \\ \hline b_{21} & b_{22} & \multicolumn{2}{c}{\mathtt{x}} \\ \hline \beta_1 & \mathtt{x} & \multicolumn{2}{c}{\mathtt{X}} \\ 0 & & & \end{array}\right).$$

This put the first columns and rows of A and B into the desired forms. Next we work on their 2nd columns and rows. Partition

$$V_1^{-1} A V_1 = \begin{array}{c} \\ 1 \\ 1 \\ 1 \\ n-3 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 1 & 1 & n-2 \end{array} \\ \begin{pmatrix} \mathtt{x} & \mathtt{x} & 0 & 0 \\ \mathtt{x} & \mathtt{x} & a_{23} & c_2^* \\ 0 & a_{32} & \mathtt{x} & \mathtt{x} \\ 0 & a_2 & \mathtt{x} & \mathtt{X} \end{pmatrix} \end{array}.$$

A breakdown occurs if $c_2^* a_2 = 0$; otherwise by Lemma 4.1, we find $\widehat{V}_{2a} \in \mathbb{C}^{(n-3)\times(n-3)}$ such that $\widehat{V}_{2a}^* c_2 = \gamma_2 e_1$ and $\widehat{V}_{2a}^{-1} a_2 = \alpha_2 e_1$. Let $V_{2a} = \mathrm{diag}(I_3, \widehat{V}_{2a})$. We have

$$V_{2a}^{-1} V_1^{-1} A V_1 V_{2a} = \left(\begin{array}{c|c|c|cc} \mathtt{x} & \mathtt{x} & 0 & \multicolumn{2}{c}{0} \\ \hline \mathtt{x} & \mathtt{x} & a_{23} & \bar{\gamma}_2 & 0 \\ \hline 0 & a_{32} & \mathtt{x} & \multicolumn{2}{c}{\mathtt{x}} \\ \hline 0 & \begin{array}{c} \alpha_2 \\ 0 \end{array} & \mathtt{x} & \multicolumn{2}{c}{\mathtt{X}} \end{array}\right),$$

$$V_{2a}^{-1}V_1^{-1}BV_1V_{2a} \;=\; \begin{array}{r} \\ 1 \\ 1 \\ 1 \\ 1 \\ n-4 \end{array}\!\!\begin{array}{c} \begin{array}{ccccc} 1 & 1 & 1 & 1 & n-4 \end{array} \\ \begin{pmatrix} \mathtt{x} & \mathtt{x} & \mathtt{x} & 0 & 0 \\ \mathtt{x} & \mathtt{x} & b_{23} & b_{24} & f_2^* \\ \mathtt{x} & b_{32} & \mathtt{x} & \mathtt{x} & \mathtt{x} \\ 0 & b_{42} & \mathtt{x} & \mathtt{x} & \mathtt{x} \\ 0 & b_2 & \mathtt{x} & \mathtt{x} & \mathtt{X} \end{pmatrix} \end{array}.$$

Yet another breakdown occurs if $f_2^*b_2 = 0$; otherwise by Lemma 4.1, we find $\widehat{V}_{2b} \in \mathbb{C}^{(n-4)\times(n-4)}$ such that $\widehat{V}_{2b}^*f_2 = \delta_2e_1$ and $\widehat{V}_{2b}^{-1}b_2 = \beta_2e_1$. Let $V_{2b} = \mathrm{diag}(I_4, \widehat{V}_{2b})$ and $V_2 \stackrel{\text{def}}{=} V_{2a}V_{2b}$. We have

$$V_2^{-1}V_1^{-1}AV_1V_2 \;=\; \left(\begin{array}{c|c|c|c} \mathtt{x} & \mathtt{x} & 0 & 0 \\ \hline \mathtt{x} & \mathtt{x} & a_{23} & \bar{\gamma}_2 \quad 0 \\ \hline 0 & a_{32} & \mathtt{x} & \mathtt{x} \\ \hline 0 & \begin{array}{c}\alpha_2 \\ 0\end{array} & \mathtt{x} & \mathtt{X} \end{array}\right),$$

$$V_2^{-1}V_1^{-1}BV_1V_2 \;=\; \left(\begin{array}{c|c|c|c|c} \mathtt{x} & \mathtt{x} & \mathtt{x} & 0 & 0 \\ \hline \mathtt{x} & \mathtt{x} & b_{23} & b_{24} & \bar{\delta}_2 \quad 0 \\ \hline \mathtt{x} & b_{32} & \mathtt{x} & \mathtt{x} & \mathtt{x} \\ \hline 0 & b_{42} & \mathtt{x} & \mathtt{x} & \mathtt{x} \\ \hline 0 & \begin{array}{c}\beta_2 \\ 0\end{array} & \mathtt{x} & \mathtt{x} & \mathtt{X} \end{array}\right).$$

By now the first two columns and rows of A and B are put into the desired forms. The process continues in a similar fashion from here. At the end, the jth column of transformed A has possible nonzero entries from positions $\lfloor (j+1)/2 \rfloor$ to $2j$, its ith row has possible nonzero entries from positions $\lfloor (i+1)/2 \rfloor$ to $2i$, and the jth column of transformed B has possible nonzero entries from positions $\max\{1, \lfloor j/2 \rfloor\}$ to $2j+1$, its ith row has possible nonzero entries from positions $\max\{1, \lfloor i/2 \rfloor\}$ to $2i+1$. Taking $V = V_1V_2\cdots V_k$ completes the reduction, where $k \le n/2$. It is easy to see $Ve_1 = e_1$ and $V^{-*}e_1 = e_1$. ∎

Theorem 4.1 *Given $v_1, w_1 \in \mathbb{C}^n$ such that $w_1^*v_1 = 1$, (unless there is a breakdown which is explained in the proof of* Lemma 4.2*) there is a matrix $V \in \mathbb{C}^{n\times n}$ with $Ve_1 = v_1$ and $V^{-*}e_1 = w_1$ such that*

$$V^{-1}AV = T_a \equiv (t_{a;ij}), \quad V^{-1}BV = T_b \equiv (t_{b;ij}) \tag{4.1}$$

satisfy

$$t_{a;ij} = 0 \text{ for } i \ge 2j+1 \text{ or } j \ge 2i+1, \quad t_{b;ij} = 0 \text{ for } i \ge 2j+2 \text{ or } j \ge 2i+2. \tag{4.2}$$

Proof: By Lemma 4.1, we can find $V_0 \in \mathbb{C}^{n\times n}$ with $V_0e_1 = v_1$ and $V_0^{-*}e_1 = w_1$. Now apply Lemma 4.2 to $V_0^{-1}AV_0$ and $V_0^{-1}BV_0$ to get $\widehat{V} \in \mathbb{C}^{n\times n}$ with $\widehat{V}e_1 = e_1 = \widehat{V}^{-*}e_1$ such that

$$\widehat{V}^{-1}(V_0^{-1}AV_0)\widehat{V} \equiv T_a, \quad \widehat{V}^{-1}(V_0^{-1}BV_0)\widehat{V} \equiv T_b$$

have the desired forms. Now letting $V = V_0\widehat{V}$ completes the proof. ∎

4.2 Lanczos Type Process

We now present a Lanczos type process to partially realize the decomposition in Theorem 4.1. Let $W = V^{-*}$ and rewrite (4.1) to get

$$\begin{array}{lll} AV = VT_a, & A^*W = WT_a^*, & W^*V = I_n, \\ BV = VT_b, & B^*W = WT_b^*. & \end{array} \tag{4.3}$$

Notice that v_1 and w_1 are given and $w_1^* v_1 = 1$. We shall show how to progressively compute the first few columns of $V = (v_1, v_2, \ldots, v_n)$, $W = (w_1, w_2, \ldots, w_n)$, T_a and T_b from v_1 and w_1 until a breakdown occurs. We have

$$Av_1 = v_1 t_{a;11} + v_2 t_{a;21}, \tag{4.4}$$

$$A^* w_1 = w_1 \bar{t}_{a;11} + w_2 \bar{t}_{a;12}, \tag{4.5}$$

$$Bv_1 = v_1 t_{b;11} + v_2 t_{b;21} + v_3 t_{b;31}, \tag{4.6}$$

$$B^* w_1 = w_1 \bar{t}_{b;11} + w_2 \bar{t}_{b;12} + w_3 \bar{t}_{b;13}. \tag{4.7}$$

Equations (4.4), (4.5), and biorthogonality between V's columns and W's columns yield $t_{a;11} = w_1^* A v_1$. Set

$$\hat{v}_2 = Av_1 - v_1 t_{a;11}, \quad \hat{w}_2 = A^* w_1 - w_1 \bar{t}_{a;11}.$$

We may take $t_{a;21} = \sqrt{|\hat{w}_2^* \hat{v}_2|}$. A breakdown occurs if $t_{a;21} = 0$; otherwise we assign

$$t_{a;12} = \hat{w}_2^* \hat{v}_2 / t_{a;21}, \quad v_2 = \hat{v}_2 / t_{a;21}, \quad w_2 = \hat{w}_2 / \bar{t}_{a;12}.$$

Then, $w_2^* v_2 = 1$. Next we turn to (4.6) and (4.7). Analogously we obtain

$$t_{b;11} = w_1^* B v_1, \quad t_{b;21} = w_2^* B v_1, \quad t_{b;12} = w_1^* B v_2. \tag{4.8}$$

Set

$$\hat{v}_3 = Bv_1 - v_1 t_{b;11} - v_2 t_{b;21}, \quad \hat{w}_3 = B^* w_1 - w_1 \bar{t}_{b;11} - w_2 \bar{t}_{b;12}. \tag{4.9}$$

We may take $t_{b;31} = \sqrt{|\hat{w}_3^* \hat{v}_3|}$. A breakdown occurs if $t_{b;31} = 0$; otherwise we assign

$$t_{b;13} = \hat{w}_3^* \hat{v}_3 / t_{b;31}, \quad v_3 = \hat{v}_3 / t_{b;31}, \quad w_3 = \hat{w}_3 / \bar{t}_{b;13}.$$

Then, $w_3^* v_3 = 1$. In general, we have for $j \geq 2$:

$$Av_j = \sum_{i=\lfloor (j+1)/2 \rfloor}^{2j-1} v_i t_{a;ij} + v_{2j} t_{a;2j\,j}, \tag{4.10}$$

$$A^* w_j = \sum_{i=\lfloor (j+1)/2 \rfloor}^{2j-1} w_i \bar{t}_{a;ji} + w_{2j} \bar{t}_{a;j\,2j}, \tag{4.11}$$

$$Bv_j = \sum_{i=\lfloor j/2 \rfloor}^{2j} v_i t_{b;ij} + v_{2j+1} t_{b;2j+1\,j}, \tag{4.12}$$

$$B^* w_j = \sum_{i=\lfloor j/2 \rfloor}^{2j} w_i \bar{t}_{b;ji} + w_{2j+1} \bar{t}_{b;j\,2j+1}. \tag{4.13}$$

Equations (4.10), (4.11), and biorthogonality between V's columns and W's columns yield

$$t_{a;ij} = w_i^* A v_j, \quad t_{a;ji} = w_j^* A v_i, \quad \text{for } \lfloor (j+1)/2 \rfloor \le i \le 2j-1. \tag{4.14}$$

Set

$$\hat{v}_{2j} = A v_j - \sum_{i=\lfloor (j+1)/2 \rfloor}^{2j-1} v_i t_{a;ij}, \quad \hat{w}_{2j} = A^* w_j - \sum_{i=\lfloor (j+1)/2 \rfloor}^{2j-1} w_i \bar{t}_{a;ji}. \tag{4.15}$$

We may take $t_{a;2j\,j} = \sqrt{|\hat{w}_{2j}^* \hat{v}_{2j}|}$. A breakdown occurs if $t_{a;2j\,j} = 0$; otherwise we assign

$$t_{a;j\,2j} = \hat{w}_{2j}^* \hat{v}_{2j} / t_{a;2j\,j}, \quad v_{2j} = \hat{v}_{2j} / t_{a;2j\,j}, \quad w_{2j} = \hat{w}_{2j} / \bar{t}_{a;j\,2j}.$$

Then, $w_{2j}^* v_{2j} = 1$. Next we turn to (4.12) and (4.13). Analogously we obtain

$$t_{b;ij} = w_i^* B v_j, \quad t_{b;ji} = w_j^* B v_i, \quad \text{for } \lfloor j/2 \rfloor \le i \le 2j. \tag{4.16}$$

Set

$$\hat{v}_{2j+1} = B v_j - \sum_{i=\lfloor j/2 \rfloor}^{2j} v_i t_{b;ij}, \quad \hat{w}_{2j+1} = B^* w_j - \sum_{i=\lfloor j/2 \rfloor}^{2j} w_i \bar{t}_{b;ji}. \tag{4.17}$$

We may take $t_{b;2j+1\,j} = \sqrt{|\hat{w}_{2j+1}^* \hat{v}_{2j+1}|}$. A breakdown occurs if $t_{b;2j+1\,j} = 0$; otherwise we assign

$$t_{b;j\,2j+1} = \hat{w}_{2j+2}^* \hat{v}_{2j+1} / t_{b;2j+1\,j}, \quad v_{2j+1} = \hat{v}_{2j+1} / t_{b;2j+1\,j}, \quad w_{2j+1} = \hat{w}_{2j+1} / \bar{t}_{b;j\,2j+1}.$$

Then, $w_{2j+1}^* v_{2j+1} = 1$. We summarize this process into the following algorithm.

Algorithm 4.1 Lanczos Type Process I:

Given v_1 **and** w_1 **such that** $w_1^* v_1 = 1$**;**
for $j = 1, 2, \dots, k$ **do**
  $\hat{v} = A v_j$; $\hat{w} = A^* w_j$;
  for $i = \lfloor (j+1)/2 \rfloor, \dots, 2j-1$ **do**
    $t_{a;ij} = w_i^* \hat{v}$; $\hat{v} = \hat{v} - v_i t_{a;ij}$;
    $t_{a;ji} = \hat{w}^* v_i$; $\hat{w} = \hat{w} - w_i \bar{t}_{a;ji}$;
  end for
  $t_{a;2j\,j} = \sqrt{|\hat{w}^* \hat{v}|}$;
  if $t_{a;2j\,j} = 0$ **then**
    BREAK;
  else
    $t_{a;j\,2j} = \hat{w}^* \hat{v} / t_{a;2j\,j}$; $v_{2j} = \hat{v} / t_{a;2j\,j}$; $w_{2j} = \hat{w} / \bar{t}_{a;j\,2j}$;
  end if
  $\hat{v} = B v_j$, $\hat{w} = B^* w_j$;
  for $i = \max\{1, \lfloor j/2 \rfloor\}, \dots, 2j$ **do**
    $t_{b;ij} = w_i^* \hat{v}$; $\hat{v} = \hat{v} - v_i t_{b;ij}$;
    $t_{b;ji} = \hat{w}^* v_i$; $\hat{w} = \hat{w} - w_i \bar{t}_{b;ji}$;
  end for
  $t_{b;2j+1\,j} = \sqrt{|\hat{w}^* \hat{v}|}$;

```
    if t_{b;2j+1 j} = 0 then
        BREAK;
    else
    t_{b;j 2j+1} = ŵ*v̂/t_{b;2j+1 j}; v_{2j+1} = v̂/t_{b;2j+1 j}; w_{2j+1} = ŵ/t̄_{b;j 2j+1};
    end if
end for
```

If A and B are Hermitian and $v_1 = w_1$, then T_a and T_b are symmetric and $v_i = w_i$. This results in a symmetric version of the above algorithm, which is also equivalent to the Arnoldi type process.

Upon completion of above process, $V_{(:,1:2k+1)}$, $W_{(:,1:2k+1)}$, and $T_{b\,(1:2k+1,1:2k+1)}$ except its right bottom block $T_{b\,(k+1:2k+1,k+1:2k+1)}$ are computed. Partially to (4.3) we have

$$AV_{(:,1:k)} = V_{(:,1:2k)}T_{a\,(1:2k,1:k)}, \quad A^*W_{(:,1:k)} = W_{(:,1:2k)}[T_{a\,(1:k,1:2k)}]^*, \tag{4.18}$$

$$BV_{(:,1:k)} = V_{(:,1:2k+1)}T_{b\,(1:2k+1,1:k)}, \quad B^*W_{(:,1:k)} = W_{(:,1:2k+1)}[T_{b\,(1:k,1:2k+1)}]^*, \tag{4.19}$$

$$[W_{(:,1:2k+1)}]^*V_{(:,1:2k+1)} = I_{2k+1}. \tag{4.20}$$

The following figures in (4.21) show what the computed part of T_a and T_b look like for $k = 5$, where the entries marked by *unfilled circles* have not been computed by Algorithm 4.1 (Lanczos Type Process I) yet, but they can be readily computed afterwards by

$$t_{a;ij} = w_i^* A v_j, \quad t_{b;ij} = w_i^* B v_j \quad \text{for } k+1 \le i,j \le 2k+1.$$

$$\begin{array}{c}T_{a\,(1:11,1:11)}\\ \begin{array}{ccccccccccc}
\bullet & \bullet & & & & & & & & & \\
\bullet & \bullet & \bullet & \bullet & & & & & & & \\
 & \bullet & \bullet & \bullet & \bullet & \bullet & & & & & \\
 & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & & & \\
 & & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \\
 & & \bullet & \bullet & \bullet & \circ & \circ & \circ & \circ & \circ & \circ \\
 & & & \bullet & \bullet & \circ & \circ & \circ & \circ & \circ & \circ \\
 & & & \bullet & \bullet & \circ & \circ & \circ & \circ & \circ & \circ \\
 & & & & \bullet & \circ & \circ & \circ & \circ & \circ & \circ \\
 & & & & \bullet & \circ & \circ & \circ & \circ & \circ & \circ \\
 & & & & & \circ & \circ & \circ & \circ & \circ & \circ
\end{array}\end{array}
\qquad
\begin{array}{c}T_{b\,(1:11,1:11)}\\ \begin{array}{ccccccccccc}
\bullet & \bullet & \bullet & & & & & & & & \\
\bullet & \bullet & \bullet & \bullet & \bullet & & & & & & \\
\bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & & & & \\
 & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & & \\
 & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
 & & \bullet & \bullet & \bullet & \circ & \circ & \circ & \circ & \circ & \circ \\
 & & \bullet & \bullet & \bullet & \circ & \circ & \circ & \circ & \circ & \circ \\
 & & & \bullet & \bullet & \circ & \circ & \circ & \circ & \circ & \circ \\
 & & & \bullet & \bullet & \circ & \circ & \circ & \circ & \circ & \circ \\
 & & & & \bullet & \circ & \circ & \circ & \circ & \circ & \circ \\
 & & & & \bullet & \circ & \circ & \circ & \circ & \circ & \circ
\end{array}\end{array} \tag{4.21}$$

The following theorem shows that indeed two sets $\{w_i\}_{i=1}^{2k+1}$ and $\{v_i\}_{i=1}^{2k+1}$ enjoy the desired biorthogonality property.

Theorem 4.2 *Suppose* Algorithm 4.1 (Lanczos Type Process I) *runs to its completion without breakdowns to produce* $\{w_i\}_{i=1}^{2k+1}$ *and* $\{v_i\}_{i=1}^{2k+1}$*. Then*

$$w_i^* v_j = 0 \text{ if } i \ne j \text{ and } w_i^* v_i = 1.$$

Proof: It is a special cases of our later proof for Theorem 4.4. ∎

Let $g\mathcal{K}_\ell(\cdot,\cdot,\cdot)$ be defined as in Subsection 3.2. Following the same argument there, we will arrive at the following theorem.

Theorem 4.3 *Suppose there is no breakdown in* Algorithm 4.1 (Lanczos Type Process I)*. Then*

$$\text{span}\{v_1, v_2, \ldots, v_\ell\} = \text{g}\mathcal{K}_\ell(\{A, B\}, v_1), \quad \text{span}\{w_1, w_2, \ldots, w_\ell\} = \text{g}\mathcal{K}_\ell(\{A^*, B^*\}, w_1).$$

We now consider a benign case of breakdown $t_{a;2j\,j} = 0$; that is when $\hat{v}_{2j} = \hat{w}_{2j} = 0$ (see (4.15)). Then, Av_j is linearly dependent on v_i that is already generated and A^*w_j is also linearly dependent on w_i's already generated. In this case, the algorithm can be continued by using Av_{j+1} and A^*w_{j+1}, or the later vectors in the sequence, to construct v_{2j+1} and w_{2j+1}. The case that $t_{b;2j+1\,j} = 0$ with $\hat{v}_{2j+1} = \hat{w}_{2j+1} = 0$ in (4.17) is treated similarly. This leads to Algorithm 4.2 below. We note that, when $\hat{v}_{2j} = 0$ but $\hat{w}_{2j} \neq 0$ (another case of breakdown), the process can also be continued by assigning to $\hat{v}_{2j}$ any vector that is orthogonal to all w_i generated but not to $\hat{w}_{2j}$.

In Algorithm 4.2, N tracks the number of vectors v_i already constructed at any given point, which is also the number of vectors w_i already constructed at that point; α_j is the value of N at the end of Line 15, i.e. the row number of the last nonzero entry in the jth column of T_a and β_j is the value of N at the end of Line 27, i.e. the row number of the last nonzero entry in the jth column of T_b; ℓ_a (and ℓ_b) tracks the row number of the first nonzero entry in the jth column of T_a (and T_b resp.). Then ℓ_a is the smallest integer such that $\alpha_{\ell_a} \geq j$.

Algorithm 4.2 Lanczos Type Process II – An Improved Version:

1. **Given** v_1 **and** w_1 **such that** $w_1^* v_1 = 1$**;**
2. $N = 1$; $\alpha_1 = 1$; $\beta_1 = 1$; $\ell_a = 1$; $\ell_b = 1$;
3. **for** $j = 1, 2, \ldots, k$ **do**
4.   **if** $j > N$ **then BREAK;**
5. $\hat{v} = Av_j$; $\hat{w} = A^* w_j$;
6.   **if** $j > \alpha_{\ell_a}$ **then** $\ell_a = \ell_a + 1$**;**
7.   **for** $i = \ell_a, \ldots, N$ **do**
8. $t_{a;ij} = w_i^* \hat{v}$; $\hat{v} = \hat{v} - v_i t_{a;ij}$;
9. $t_{a;ji} = \hat{w}^* v_i$; $\hat{w} = \hat{w} - w_i \bar{t}_{a;ji}$;
10.   **end for**
11. $t_{a;N+1,j} = \sqrt{|\hat{w}^* \hat{v}|}$;
12.   **if** $t_{a;N+1,j} = 0$ **then**
13.   **if** $\hat{v} \neq 0$ **or** $\hat{w} \neq 0$ **then BREAK;**
14.   **else**
15. $N = N + 1$; $t_{a;j\,N} = \hat{w}^* \hat{v} / t_{a;N\,j}$; $v_N = \hat{v} / t_{a;N\,j}$; $w_N = \hat{w} / \bar{t}_{a;j\,N}$, $\alpha_j = N$;
16.   **end if**
17. $\hat{v} = Bv_j$; $\hat{w} = B^* w_j$;
18.   **if** $j > \beta_{\ell_b}$ **then** $\ell_b = \ell_b + 1$**;**
19.   **for** $i = \ell_b, \ldots, N$ **do**
20. $t_{b;ij} = w_i^* \hat{v}$; $\hat{v} = \hat{v} - v_i t_{b;ij}$;
21. $t_{b;ji} = \hat{w}^* v_i$; $\hat{w} = \hat{w} - w_i \bar{t}_{b;ji}$;

```
22.     end for
23.     t_{b;N+1,j} = sqrt(|ŵ* v̂|);
24.     if t_{b;N+1,j} = 0 then
25.         If v̂ ≠ 0 or ŵ ≠ 0 then BREAK;
26.     else
27.         N = N + 1, t_{b;j N} = ŵ* v̂ / t_{b;N j}; v_N = v̂ / t_{b;N j}; w_N = ŵ / t̄_{b;j N}, β_j = N;
28.     end if
29. end for
```

If the execution of Algorithm 4.2 is completed by the **BREAK** statement at Line 4, a common (right) invariant subspace and a common left invariant subspace for A and B are has been computed, and

$$AV_{(:,1:N)} = V_{(:,1:N)}T_{a\,(1:N,1:N)}, \quad A^*W_{(:,1:N)} = W_{(:,1:N)}[T_{a\,(1:N,1:N)}]^*, \tag{4.22}$$

$$BV_{(:,1:N)} = V_{(:,1:N)}T_{b\,(1:N,1:N)}, \quad B^*W_{(:,1:N)} = W_{(:,1:N)}[T_{b\,(1:N,1:N)}]^*, \tag{4.23}$$

$$[W_{(:,1:N)}]^*V_{(:,1:N)} = I_N. \tag{4.24}$$

If none of the **BREAK** statements is executed, we have

$$AV_{(:,1:k)} = V_{(:,1:\alpha_k)}T_{a\,(1:\alpha_k,1:k)}, \quad A^*W_{(:,1:k)} = W_{(:,1:\alpha_k)}[T_{a\,(1:k,1:\alpha_k)}]^*, \tag{4.25}$$

$$BV_{(:,1:k)} = V_{(:,1:\beta_k)}T_{b\,(1:\beta_k,1:k)}, \quad B^*W_{(:,1:k)} = W_{(:,1:\beta_k)}[T_{b\,(1:k,1:\beta_k)}]^*, \tag{4.26}$$

$$[W_{(:,1:\beta_k)}]^*V_{(:,1:\beta_k)} = I_{\beta_k}, \tag{4.27}$$

In this case, parts of projections T_a and T_b are not computed, but they can be readily obtained afterwards by

$$t_{a;ij} = q_i^*Aq_j, \quad t_{b;ij} = q_i^*Bq_j \quad \text{for } k+1 \le i \le N \text{ and } k+1 \le j \le N.$$

It can be seen that the nonzero patterns, i.e., the positions of nonzero entries, in T_a and T_b here, are contained in those as described in (4.21).

The following theorem shows that indeed two sets $\{w_i\}$ and $\{v_i\}$ by Algorithm 4.2 enjoy the desired biorthogonality property.

Theorem 4.4 *Suppose* Algorithm 4.2 (Lanczos Type Process II) *runs to its completion without breakdowns to produce* $\{w_i\}_{i=1}^N$ *and* $\{v_i\}_{i=1}^N$*, where* $N = \beta_k$*. Then*

$$w_i^*v_j = 0 \text{ if } i \ne j \text{ and } w_i^*v_i = 1.$$

Proof: $w_i^*v_i = 1$ is clear from their definition. We shall only need to show $w_i^*v_j = 0$ if $i \ne j$. Let ℓ_{aj} and ℓ_{bj} be the ℓ_a and ℓ_b used at Lines 7 and 19 respectively at the jth trip of the j-loop, $\hat{v}_i$ and $\hat{w}_i$ be the ones finally used to give v_i and w_i by scalar scaling in Algorithm 4.2.

We prove the claim by induction on k. The case for $k = 1$ is easy to establish. Suppose the claim in the theorem holds for $k = j-1$. We now show it also holds for $k = j$, i.e., $w_s^*\hat{v}_{\alpha_j} = 0 = \hat{w}_{\alpha_j}^*v_s$ for $s < \alpha_j$ and $w_s^*\hat{v}_{\beta_j} = 0 = \hat{w}_{\beta_j}^*v_s$ for $s < \beta_j$.

$w_s^*\hat{v}_{\alpha_j} = 0 = \hat{w}_{\alpha_j}^* v_s$ for $\ell_{aj} \le s \le \alpha_j - 1$ follows from definition. For the case of $s < \ell_{aj}$, we have

$$\begin{aligned} w_s^*\hat{v}_{\alpha_j} &= w_s^*\left(Av_j - \sum_{i=\ell_{aj}}^{\alpha_j-1} v_i t_{a;ij}\right) \\ &= w_s^* A v_j = \left[\hat{w}_{\alpha_s} + \sum_{i=\ell_{as}}^{\alpha_s-1} w_i \bar{t}_{a;si}\right]^* v_j \\ &= 0, \\ \hat{w}_{\alpha_j}^* v_s &= \left[A^* w_j - \sum_{i=\ell_{aj}}^{\alpha_j-1} w_i \bar{t}_{a;ji}\right]^* v_s \\ &= w_j^* A v_s = w_j^*\left(\hat{v}_{\alpha_s} + \sum_{i=\ell_{as}}^{\alpha_s-1} v_i t_{a;is}\right) \\ &= 0, \end{aligned}$$

by induction hypothesis, since $s < \ell_{aj}$ implies $\alpha_s < j$ by Line 6.

Analogously we can show $w_s^*\hat{v}_{\beta_j} = 0 = \hat{w}_{\beta_j}^* v_s$ for $s < \beta_j$. ∎

4.3 General monic matrix polynomial

As in 3.4, the Lanczos type process developed can also be extended to a general monic matrix polynomial(1.1) of degree m. Again, we just state the corresponding decomposition results and omit a detailed statement of an algorithm.

Theorem 4.5 *Let $A_\ell \in \mathbb{C}^{n\times n}$, $0 \le \ell \le m-1$. Given $v_1, w_1 \in \mathbb{C}^n$ such that $w_1^* v_1 = 1$, (unless there is a breakdown) there is a matrix $V \in \mathbb{C}^{n\times n}$ with $Ve_1 = v_1$ and $V^{-*}e_1 = w_1$ such that*

$$V^{-1} A_\ell V = T_a \equiv (t_{\ell;ij}), \quad \text{for } 0 \le \ell \le m-1 \tag{4.28}$$

satisfying

$$t_{\ell;ij} = 0 \text{ for } i \ge mj + m - \ell \text{ or } j \ge mi + m - \ell. \tag{4.29}$$

5 Application to Model Reduction

In a second order single-input and single-output linear system, a quadratic matrix polynomial is involved in its transfer function

$$f(s) = c^*(I - As - Bs^2)^{-1} b$$

where b and c are n-dimensional vectors, A and B are $n \times n$, either sparse or in some kinds of factored forms. In model reductions, it is desirable to approximate the given system by another second order system of lower dimension that is called a reduced system. The approximation is usually in terms of the transfer functions and is often done by requiring that the transfer function of the reduced system $g(s)$ and the original transfer function $f(s)$ to have the same moments up to certain degree (i.e., terms associated with s^0, s^1, s^2, ... of their Taylor expansions at $s = 0$). In the case of the first order systems, Feldman and Freund [11] show that the Lanczos algorithm is a powerful method that can be used to achieve this. Here we shall show that the Arnoldi/Lanczos type algorithm derived in the previous sections can be used in the same way for second order systems.

- For the Arnoldi type process, let Algorithm 3.1 produce $Q_{(:,1:N)}$, $H_{a\,(1:N,1:N)}$, and $H_{b\,(1:N,1:N)}$. Define

$$g_{\rm arnd}(s) = \widetilde{c}_N^* \left(I - H_{a\,(1:N,1:N)}\, s - H_{b\,(1:N,1:N)}\, s^2\right)^{-1} \|b\|_2 e_1, \tag{5.1}$$

  where $q_1 = b/\|b\|_2$ and $\widetilde{c}_N = Q^*_{(:,1:N)} c$.

- For the Lanczos type process, suppose that Algorithm 4.2 runs to its completion without breakdowns and produces $V_{(:,1:N)}$, $W_{(:,1:N)}$, $T_{a\,(1:N,1:N)}$, and $T_{b\,(1:N,1:N)}$. Define

$$g_{\rm lanz}(s) = (c^* b) e_1^* \left(I - T_{a\,(1:N,1:N)}\, s - T_{b\,(1:N,1:N)}\, s^2\right)^{-1} e_1. \tag{5.2}$$

We shall next find out how accurate $g_{\rm arnd}(s)$ and $g_{\rm lanz}(s)$ are as approximations to $f(s)$ by determining the numbers of matching leading terms in their Taylor expansions at $s = 0$. We start by presenting two lemmas.

Lemma 5.1 *Let $X = (x_{ij})$ and $Y = (y_{ij})$ be two $n \times n$ matrices satisfying*

$$x_{ij} = 0 \text{ for } i > k_1 j + \ell_1, \text{ and } y_{ij} = 0 \text{ for } i > k_2 j + \ell_2,$$

and let $Z = XY = (z_{ij})$. Assume that no zero entries in Z are caused by exact arithmetic cancellations. Then $z_{ij} = 0$ if and only if $i > k_1 k_2 j + (k_1 \ell_2 + \ell_1)$.

Proof: We have

$$\begin{aligned} z_{ij} = \sum_{m=1}^{n} x_{im} y_{mj} &= \sum_{\{m\,:\,i \le k_1 m + \ell_1\}} x_{im} y_{mj} + \sum_{\{m\,:\,i > k_1 m + \ell_1\}} x_{im} y_{mj} \\ &= \sum_{\{m\,:\,i \le k_1 m + \ell_1\}} x_{im} y_{mj} + 0 \end{aligned} \tag{5.3}$$

since $x_{im} = 0$ for $i > k_1 m + \ell_1$. Now for $i > (k_1 k_2) j + (k_1 \ell_2 + \ell_1)$ and $i \le k_1 m + \ell_1$, we have

$$k_1 m + \ell_1 > (k_1 k_2) j + (k_1 \ell_2 + \ell_1) \quad \Rightarrow \quad m > k_2 j + \ell_2;$$

so $y_{mj} = 0$. Therefore $z_{ij} = 0$ for $i > (k_1 k_2) j + (k_1 \ell_2 + \ell_1)$ by (5.3). It can be seen that in the generic case if $i \le (k_1 k_2) j + (k_1 \ell_2 + \ell_1)$, then at least one summand in $\sum_{\{m\,:\,i \le k_1 m + \ell_1\}} x_{im} y_{mj}$ is not zero. ∎

The following lemma can be verified in a straightforward way.

Lemma 5.2 *Let X and Y be $n \times n$, partitioned as*

$$X = \begin{array}{c} \\ k \\ n-k \end{array}\!\!\begin{array}{c} \begin{array}{cc} k & n-k \end{array} \\ \begin{pmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{pmatrix} \end{array}, \quad Y = \begin{array}{c} \\ k \\ n-k \end{array}\!\!\begin{array}{c} \begin{array}{cc} k & n-k \end{array} \\ \begin{pmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{pmatrix} \end{array},$$

and let $1 \le j \le k$. If $Y_{21}e_j = 0$, then $XYe_j = \begin{pmatrix} X_{11}Y_{11}e_j \\ \mathtt{x} \end{pmatrix}$.

5.1 Arnoldi Type Process

Theorem 5.1 *Let $g_{\mathrm{arnd}}(s)$ be as defined in* (5.1) *as a result of Algorithm 3.1. Then*

$$f(s) = g_{\mathrm{arnd}}(s) + \mathcal{O}(s^{\lfloor \log_2 N \rfloor + 1}).$$

If, in addition, $c = b$, then $f(s) = g_{\mathrm{arnd}}(s) + \mathcal{O}(s^{\lfloor \log_2 N \rfloor + 2})$.

We shall work with the case $q_1 = b/\|b\|_2$ only; the other case can be treated similarly. If the Arnoldi Type Process is performed to its completion, we will have $n \times n$ matrices Q, $H_a \equiv (h_{a;ij})$, and $H_b \equiv (h_{b;ij})$, satisfying

$$h_{a;ij} = 0 \text{ for } i \ge 2j+1 \text{ and } h_{b;ij} = 0 \text{ for } i \ge 2j+2, \tag{5.4}$$

such that

$$AQ = QH_a, \quad BQ = QH_b.$$

We note that the sparsity pattern of (5.4) defines an envelope enclosing nonzeros of H_a and H_b, while the matrices H_a and H_b as generated by the algorithm may be more condensed. From the reduction, we have

$$f(s)/\|b\|_2 = \widetilde{c}^* \left(I - H_a s - H_b s^2\right)^{-1} e_1,$$

where $\widetilde{c} = Q^* c$. The Taylor expansion of $f(s)/\|b\|_2$ at $s = 0$ is

$$f(s)/\|b\|_2 = \widetilde{c}^* \left(\sum_{\ell=0}^{\infty} H_\ell s^\ell \right) e_1 = \sum_{\ell=0}^{\infty} (\widetilde{c}^* H_\ell e_1) s^\ell,$$

where, as a result of

$$\left(I - H_a s - H_b s^2\right) \left(\sum_{\ell=0}^{\infty} H_\ell s^\ell \right) = I_n = \left(\sum_{\ell=0}^{\infty} H_\ell s^\ell \right) \left(I - H_a s - H_b s^2\right),$$

H_ℓ can be recursively defined as

$$\begin{aligned} H_0 &= I_n, && (5.5) \\ H_1 &= H_a, && (5.6) \\ H_\ell &= H_a H_{\ell-1} + H_b H_{\ell-2} \quad \text{for } \ell \ge 2, && (5.7) \\ &= H_{\ell-1} H_a + H_{\ell-2} H_b \quad \text{for } \ell \ge 2. && (5.8) \end{aligned}$$

On the other hand, for $g_{\rm arnd}(s)$ we have

$$\begin{aligned} g_{\rm arnd}(s)/\|b\|_2 &= \widetilde{c}_N^* \left(I - H_{a\,(1:N,1:N)}\, s - H_{b\,(1:N,1:N)}\, s^2\right)^{-1} e_1 \\ &= \sum_{\ell=0}^{\infty} (\widetilde{c}_N^* \widetilde{H}_\ell e_1) s^\ell, \end{aligned}$$

where, as a result of

$$\begin{aligned} &\left(I - H_{a\,(1:N,1:N)}\, s - H_{b\,(1:N,1:N)}\, s^2\right) \left(\sum_{\ell=0}^{\infty} \widetilde{H}_\ell s^\ell\right) = I_N \\ &= \left(\sum_{\ell=0}^{\infty} \widetilde{H}_\ell s^\ell\right) \left(I - H_{a\,(1:N,1:N)}\, s - H_{b\,(1:N,1:N)}\, s^2\right), \end{aligned}$$

$\widetilde{H}_\ell$ can be recursively defined as

$$\widetilde{H}_0 = I_N, \tag{5.9}$$

$$\widetilde{H}_1 = H_{a\,(1:N,1:N)}, \tag{5.10}$$

$$\widetilde{H}_\ell = H_{a\,(1:N,1:N)} \widetilde{H}_{\ell-1} + H_{b\,(1:N,1:N)} \widetilde{H}_{\ell-2} \quad \text{for } \ell \geq 2, \tag{5.11}$$

$$\phantom{\widetilde{H}_\ell} = \widetilde{H}_{\ell-1} H_{a\,(1:N,1:N)} + \widetilde{H}_{\ell-2} H_{b\,(1:N,1:N)} \quad \text{for } \ell \geq 2. \tag{5.12}$$

Lemma 5.3 *H_ℓ and $\widetilde{H}_\ell$ have the property that their (i,j)th entries are zero if $i > 2^\ell j$.*

Proof: We prove this by induction for H_ℓ; the proof is similar for $\widetilde{H}_\ell$. Clearly the result holds for $H_0 = I_n$ and $H_1 = H_a$. Suppose the result holds for all matrices $H_0, H_1, \ldots, H_\ell$, then the $(i,j)th$ entry of $H_a H_\ell$ is zero for $i > 2^{\ell+1} j$ by Lemma 5.1 and by (3.2). Also by Lemma 5.1, the $(i,j)th$ entry of $H_b H_{\ell-1}$ is zero for $i > 2^\ell j + 1$. But for $\ell \geq 1$ and $j \geq 1$, $2^{\ell+1} j \geq 2^\ell j + 2$. Therefore, the result now follows from the recurrence relation. ∎

Lemma 5.4 *Suppose $N \geq 3$, and let $m = \lfloor \log_2 N \rfloor$, the largest possible integer such that $2^m \leq N$. Then*

1. $H_\ell e_j = \begin{matrix} {\scriptstyle N} \\ {\scriptstyle n-N} \end{matrix} \begin{pmatrix} \widetilde{H}_\ell e_j \\ 0 \end{pmatrix}$ *for $\ell = 0, 1, \ldots, m$ and $j = 1, 2, \ldots, 2^{m-\ell}$.*

2. $H_{m+1} e_1 = \begin{matrix} {\scriptstyle N} \\ {\scriptstyle n-N} \end{matrix} \begin{pmatrix} \widetilde{H}_{m+1} e_1 \\ \mathtt{x} \end{pmatrix}.$

Proof: That the last $n - N$ entries of $H_\ell e_j$ for $0 \leq \ell \leq m$ and $1 \leq j \leq 2^{m-\ell}$ are all zeros is due to Lemma 5.3 since $N + 1 > 2^m = 2^\ell 2^{m-\ell} \geq 2^\ell j$. We shall prove Claim 1 by induction on ℓ. It holds true for $\ell = 0, 1$. Suppose $m \geq \ell \geq 2$ and that the claim holds for $0, 1, \ldots, \ell - 1$. Then

$$\begin{aligned}
H_\ell e_j &= H_a H_{\ell-1} e_j + H_b H_{\ell-2} e_j \\
&= H_a \begin{pmatrix} \widetilde{H}_{\ell-1} e_j \\ 0 \end{pmatrix} + H_b \begin{pmatrix} \widetilde{H}_{\ell-2} e_j \\ 0 \end{pmatrix} \\
&= H_a \begin{pmatrix} \widetilde{H}_{\ell-1} & 0 \\ 0 & 0 \end{pmatrix} e_j + H_b \begin{pmatrix} \widetilde{H}_{\ell-2} & 0 \\ 0 & 0 \end{pmatrix} e_j \\
&= \begin{pmatrix} H_{a\,(1:N,1:N)} \widetilde{H}_{\ell-1} e_j \\ \mathtt{x} \end{pmatrix} + \begin{pmatrix} H_{b\,(1:N,1:N)} \widetilde{H}_{\ell-2} e_j \\ \mathtt{x} \end{pmatrix} \\
&= \begin{pmatrix} \widetilde{H}_\ell e_j \\ \mathtt{x} \end{pmatrix}
\end{aligned}$$

on using Lemma 5.2. The lower block marked by $\mathtt{x}$ is 0 for $\ell \leq m$ and $j \leq 2^{m-\ell}$ as we commented; Claim 1 is proved. With Claim 1 proved, setting $\ell = m + 1$ and $j = 1$ in the above equations leads to Claim 2. ∎

Proof of Theorem 5.1: $\ell \leq m = \lfloor \log_2 N \rfloor$ implies $2^\ell \leq N$; and thus by Lemma 5.4

$$\widetilde{c}^* H_\ell e_1 = \begin{pmatrix} \widetilde{c}_N \\ \mathtt{x} \end{pmatrix}^* \begin{pmatrix} \widetilde{H}_\ell e_1 \\ 0 \end{pmatrix} = \widetilde{c}_N^* \widetilde{H}_\ell e_1,$$

as expected. Now if $c = b$, $\widetilde{c} = \widetilde{c}_N = \|b\|_2 e_1$, then also by Lemma 5.4

$$e_1^* H_{m+1} e_1 = \begin{pmatrix} e_1 \\ 0 \end{pmatrix}^* \begin{pmatrix} \widetilde{H}_{m+1} e_1 \\ \mathtt{x} \end{pmatrix} = e_1^* \widetilde{H}_{m+1} e_1,$$

as was to be shown. ∎

5.2 Lanczos Type Process

Theorem 5.2 *Suppose Algorithm 4.2 runs to its completion without breakdowns with $Ve_1 = \alpha b$ and $We_1 = \beta c$, and as a result let $g_{\rm lanz}(s)$ be as defined in* (5.2)*. Assume also the process can be continued until all n columns of V (and W) are obtained. Then*

$$f(s) = g_{\rm lanz}(s) + \mathcal{O}(s^{2\lfloor \log_2 N \rfloor + 1}).$$

The assumption that the Lanczos type process can be continued until all n columns of V (and W) are obtained is for our theoretical analysis only. In practice the process will stop much earlier than that for large sparse matrices. With the assumption, we have V, W, $T_a \equiv (t_{a;ij})$, and $T_b \equiv (t_{b;ij})$, satisfying

$$t_{a;ij} = 0 \text{ for } i \geq 2j+1 \text{ or } 2i+1 \leq j \text{ and } t_{b;ij} = 0 \text{ for } i \geq 2j+2 \text{ or } 2i+2 \leq j,$$

such that

$$AV = VT_a, \quad W^*A = T_aW^*, \quad BV = VT_b, \quad W^*B = T_bW^*.$$

We note that the sparse pattern of (5.4) defines an envelope enclosing nonzeros of T_a and T_b, while the matrices T_a and T_b as generated may be more condensed. From the reduction, we have

$$f(s)/(c^*b) = e_1^* \left(I - T_a s - T_b s^2\right)^{-1} e_1$$

whose Taylor expansion at $s = 0$ is

$$f(s)/(c^*b) = e_1^* \left(\sum_{\ell=0}^{\infty} T_\ell s^\ell\right) e_1 = \sum_{\ell=0}^{\infty} (e_1^* T_\ell e_1) s^\ell,$$

where T_ℓ can be recursively defined the same as (5.5) – (5.8) with H replaced by T. On the other hand, for $g_{\text{lanz}}(s)$ we have

$$g_{\text{lanz}}(s)/(c^*b) = \sum_{\ell=0}^{\infty} (e_1^* \widetilde{T}_\ell e_1) s^\ell,$$

where $\widetilde{T}_\ell$ can also be recursively defined the same as (5.9) – (5.12) with H replaced by T.

Lemma 5.5 *T_ℓ and $\widetilde{T}_\ell$ have the property that their (i, j)th entries are zero if $i > 2^\ell j$ or $2^\ell i < j$.*

Proof: Analogous to the proof of Lemma 5.3. ∎

Lemma 5.6 *Suppose $N \geq 3$, and let $m = \lfloor \log_2 N \rfloor$, the largest possible integer such that $2^m \leq N$. Then*

1. $T_\ell e_j = \begin{matrix} {\scriptstyle N} \\ {\scriptstyle n-N} \end{matrix} \begin{pmatrix} \widetilde{T}_\ell e_j \\ 0 \end{pmatrix}$ *for $\ell = 0, 1, \ldots, m$ and $j = 1, 2, \ldots, 2^{m-\ell}$.*

2. $T_{m+1} e_1 = \begin{matrix} {\scriptstyle N} \\ {\scriptstyle n-N} \end{matrix} \begin{pmatrix} \widetilde{T}_{m+1} e_1 \\ \mathtt{x} \end{pmatrix}.$

3. $e_i^* T_\ell = \begin{pmatrix} \overset{N}{e_i^* \widetilde{T}_\ell} & \overset{n-N}{0} \end{pmatrix}$ *for $\ell = 0, 1, \ldots, m$ and $i = 1, 2, \ldots, 2^{m-\ell}$.*

4. $e_1^* T_{m+1} = \begin{pmatrix} \overset{N}{e_1^* \widetilde{T}_{m+1}} & \overset{n-N}{\mathtt{x}} \end{pmatrix}.$

5. $e_i^* T_\ell e_1 = e_i^* \widetilde{T}_\ell e_1$ *for $\ell = 0, 1, \ldots, 2m$ and $i = 1, 2, \ldots, \min\{2^m, 2^{2m-\ell}\}$.*

Proof: Claims 1 and 2 follow from Lemma 5.4. We prove Claim 3 by induction on ℓ. One can check that the claim holds for T_0 and T_1. Suppose $i \leq 2^{m-\ell}$, and the claim holds for $0, 1, \ldots, \ell - 1$. Then

$$\begin{aligned} e_i^* T_\ell &= e_i^* T_a T_{\ell-1} + e_i^* T_b T_{\ell-2} \\ &= \left(\sum_{j=\lfloor (i+1)/2 \rfloor}^{2i} t_{a;ij} e_j^*\right) T_{\ell-1} + \left(\sum_{j=\max\{1, \lfloor i/2 \rfloor\}}^{2i+1} t_{b;ij} e_j^*\right) T_{\ell-2}. \end{aligned}$$

Note that $2i \le 2^{m-(\ell-1)}$ and $2i+1 \le 2^{m-(\ell-2)}$, so we can apply the induction hypothesis to get

$$\begin{aligned} e_i^* T_\ell &= \sum_{j=\lfloor (i+1)/2 \rfloor}^{2i} t_{a;ij} (e_j^* \widetilde{T}_{\ell-1} \quad 0) + \sum_{j=\max\{1,\lfloor i/2 \rfloor\}}^{2i+1} t_{b;ij} (e_j^* \widetilde{T}_{\ell-2} \quad 0) \\ &= (e_i^* T_{a\,(1:N,1:N)} \widetilde{T}_{\ell-1} + e_i^* T_{b\,(1:N,1:N)} \widetilde{T}_{\ell-2} \quad 0) \\ &= (e_i^* \widetilde{T}_\ell \quad 0). \end{aligned}$$

With Claim 3 proved, setting $\ell = m+1$ and $i = 1$ in the above equations leads to Claim 4.

We now prove Claim 5 by induction on ℓ. First for $0 \le \ell \le m$, the claim holds due to Claim 1; the claim also holds for $\ell = m+1$ because of Claim 2. In fact, we can say more

$$e_i^* T_\ell e_1 = e_i^* \widetilde{T}_\ell e_1 \text{ for } 0 \le \ell \le m+1 \text{ and } 1 \le i \le 2^m.$$

We now prove Claim 5 for the rest of ℓ:

$$e_i^* T_\ell e_1 = e_i^* \widetilde{T}_\ell e_1 \text{ for } m \le \ell \le m+1 \text{ and } 1 \le i \le 2^{2m-\ell}. \tag{5.13}$$

by induction on ℓ. This claim holds for $\ell = m, m+1$. Suppose that $m+2 \le \ell \le 2m$, $i \le 2^{2m-\ell}$, and that (5.13) holds for $m, m+1, \ldots, \ell-1$. Notice that $2i \le 2^{2m-(\ell-1)}$ and $2i+1 < 2^{2m-(\ell-2)}$. Then

$$\begin{aligned} e_i^* T_\ell e_1 &= e_i^* (T_a T_{\ell-1} + T_b T_{\ell-2}) e_1 \\ &= \left(\sum_{j=\lfloor (i+1)/2 \rfloor}^{2i} t_{a;ij} e_j^* \right) T_{\ell-1} e_1 + \left(\sum_{j=\max\{1,\lfloor i/2 \rfloor\}}^{2i+1} t_{b;ij} e_j^* \right) T_{\ell-2} e_1 \\ &= \sum_{j=\lfloor (i+1)/2 \rfloor}^{2i} t_{a;ij} (e_j^* \widetilde{T}_{\ell-1} e_1) + \sum_{j=\max\{1,\lfloor i/2 \rfloor\}}^{2i+1} t_{b;ij} (e_j^* \widetilde{T}_{\ell-2} e_1), \end{aligned}$$

on using the induction hypothesis. This can be simplified to

$$\begin{aligned} e_i^* T_\ell e_1 &= e_i^* T_{a\,(1:N,1:N)} \widetilde{T}_{\ell-1} e_1 + e_i^* T_{b\,(1:N,1:N)} \widetilde{T}_{\ell-2} e_1 \\ &= e_i^* \widetilde{T}_\ell e_1, \end{aligned}$$

as expected. ∎

Proof of Theorem 5.2: It is a consequence of Claim 5 of Lemma 5.6. ∎

6 Quadratic Eigenvalue Problems

The Krylov type methods that we have derived can also be used to compute eigenvalues and eigenvectors of $I\lambda^2 - A\lambda - B$ as follows.

- For the Arnoldi type process, if Algorithm 3.1 produces $Q_{(:,1:N)}$, $H_{a\,(1:N,1:N)}$, and $H_{b\,(1:N,1:N)}$ and θ_i is an eigenvalue and u_i is a right eigenvector of

$$I\lambda^2 - H_{a\,(1:N,1:N)}\lambda - H_{b\,(1:N,1:N)}, \tag{6.1}$$

then we use (θ_i, x_i) as an approximate eigenpair for the original problem, where

$$x_i = Q_{(:,1:N)}u_i. \tag{6.2}$$

- For the Lanczos type process, if Algorithm 4.2 runs to its completion without breakdowns and produces $V_{(:,1:N)}$, $W_{(:,1:N)}$, $T_{a\,(1:N,1:N)}$, and $T_{b\,(1:N,1:N)}$, and if θ_i is an eigenvalue and u_i (and v_i) is a right (left, resp.) eigenvector of

$$I\lambda^2 - T_{a\,(1:N,1:N)}\lambda - T_{b\,(1:N,1:N)}, \tag{6.3}$$

then we use θ_i as an approximate eigenvalue with x_i (and y_i) as an approximate right (left, resp.) eigenvector for the original problem, where

$$x_i = V_{(:,1:N)}u_i, \quad y_i = W_{(:,1:N)}v_i. \tag{6.4}$$

The above processes approximate a quadratic eigenvalue problem with a projected quadratic eigenvalue problem.

We now discuss convergence properties for the Lanczos type process only. Corresponding results for the Arnoldi type process can be obtained similarly. We first present a residual bound for the Ritz values and vectors.

Theorem 6.1 *For the Ritz values and Ritz vector obtained by the Lanczos type process II (Algorithm 4.2), we have*

$$\|(I\theta_i^2 - A\theta_i - B)x_i\| \le \|V\|(\,\|\theta_i T_{a\,(N+1:\alpha_N,p:N)}\| + \|T_{b\,(N+1:\beta_N,p:N)}\|)\|u_{i,(p:N)}\|$$

and

$$\|y_i^*(I\theta_i^2 - A\theta_i - B)\| \le \|W\|(\,\|\theta_i T_{a\,(p:N,N+1:\alpha_N)}\| + \|T_{b\,(p:N,N+1:\beta_N)}\|)\|v_{i,(p:N)}\|$$

where p is the smallest integer such that $\beta_p > N$ and is equal to the value of ℓ_b at step $N+1$.

Proof: First, from (4.25) and (4.26), we have

$$\begin{aligned} AV_{(:,1:N)} &= V_{(:,1:N)}T_{a\,(1:N,1:N)} + V_{(:,N+1:\alpha_N)}T_{a\,(N+1:\alpha_N,1:N)} \\ BV_{(:,1:N)} &= V_{(:,1:N)}T_{b\,(1:N,1:N)} + V_{(:,N+1:\beta_N)}T_{b\,(N+1:\beta_N,1:N)} \end{aligned}$$

Then

$$\begin{aligned} (I\theta_i^2 - A\theta_i - B)x_i &= (\theta_i^2 V_{(:,1:N)} - \theta_i AV_{(:,1:N)} - BV_{(:,1:N)})u_i \\ &= V_{(:,1:N)}(I\theta_i^2 - T_{a\,(1:N,1:N)}\theta_i - T_{b\,(1:N,1:N)})u_i \\ &\quad -\theta_i V_{(:,N+1:\alpha_N)}T_{a\,(N+1:\alpha_N,1:N)}u_i - V_{(:,N+1:\beta_N)}T_{b\,(N+1:\beta_N,1:N)}u_i \\ &= -\theta_i V_{(:,N+1:\alpha_N)}T_{a\,(N+1:\alpha_N,p:N)}u_{i,(p:N)} \\ &\quad -V_{(:,N+1:\beta_N)}T_{b\,(N+1:\beta_N,p:N)}u_{i,(p:N)}, \end{aligned}$$

where we note that the first $p-1$ columns of $T_{b\,(N+1:\beta_N,1:N)}$ and $T_{a\,(N+1:\alpha_N,1:N)}$ are zeros. Taking the norm of above, we obtain the bound. ∎

We next derive an error bound similar to that of [34]. Let

$$L = \begin{pmatrix} 0 & I \\ T_b & T_a \end{pmatrix}, \quad \text{and} \quad L_N = \begin{pmatrix} 0 & I \\ T_{b\,(1:N,1:N)} & T_{a\,(1:N,1:N)} \end{pmatrix}.$$

Lemma 6.1 *Let S_ℓ and T_ℓ be recursively defined by*

$$\begin{aligned} S_0 &= 0, & T_0 &= I_n, \\ S_1 &= T_b, & T_1 &= T_a, \\ S_\ell &= T_a S_{\ell-1} + T_b S_{\ell-2} & T_\ell &= T_a T_{\ell-1} + T_b T_{\ell-2} \quad \textit{for } \ell \geq 2. \end{aligned}$$

Then

$$L^\ell = \begin{pmatrix} S_{\ell-1} & T_{\ell-1} \\ S_\ell & T_\ell \end{pmatrix}.$$

Similarly, let $\widetilde{S}_\ell$ and $\widetilde{T}_\ell$ be recursively defined from L_N as in S_ℓ and T_ℓ above. Then,

$$L_N^\ell = \begin{pmatrix} \widetilde{S}_{\ell-1} & \widetilde{T}_{\ell-1} \\ \widetilde{S}_\ell & \widetilde{T}_\ell \end{pmatrix}.$$

Proof: It can be verified by induction. ∎

It can be seen that the conclusion of Lemma 5.6 holds for $\widetilde{T}_\ell$ and T_ℓ as well as for $\widetilde{S}_\ell$ and S_ℓ. Thus we have

$$e_1^* S_\ell e_1 = e_1^* \widetilde{S}_\ell e_1,$$

for $\ell = 0, 1, \ldots, 2m$ ($m = \lfloor \log_2 N \rfloor$), which implies

$$e_1^* L^{\ell+1} e_1 = e_1^* L_N^{\ell+1} e_1.$$

Therefore, for any polynomial f of degree $2m+1$,

$$e_1^* f(L) e_1 = e_1^* f(L_N) e_1. \tag{6.5}$$

There are other similar results. They include, for example,

$$e_{n+1}^* L^\ell e_1 = e_{k+1}^* L_N^\ell e_1 \tag{6.6}$$

For the sake of simplicity, we assume now that L and L_N are diagonalizable, and write

$$L_N = U^* \Theta V \quad \text{and} \quad L = X^* \Lambda Y, \tag{6.7}$$

where Θ =diag$(\theta_1, ..., \theta_m)$, $U^*V = I$ and Λ =diag$(\lambda_1, ..., \lambda_n)$, $X^*Y = I$. Write $U = (u_{ij}), V = (v_{ij})$, $X = (x_{ij})$ and $Y = (y_{ij})$. Now substituting (6.7) into (6.5), we obtain

$$e_1^* X^* f(\Lambda) Y e_1 = e_1^* U^* f(\Theta) V e_1.$$

Thus

$$\sum_{i=1}^{n} f(\lambda_i)\bar{x}_{i1}y_{i1} = \sum_{i=1}^{m} f(\theta_i)\bar{u}_{i1}v_{i1}.$$

In particular, using $f(x) = (x - \theta_1)p(x)$, we have

$$\lambda_1 - \theta_1 = \frac{1}{p(\lambda_1)\bar{x}_{11}y_{11}} \left[-\sum_{i=2}^{n} (\lambda_i - \theta_1)p(\lambda_i)\bar{x}_{i1}y_{i1} + \sum_{i=2}^{m} (\theta_i - \theta_1)p(\theta_i)\bar{u}_{i1}v_{i1} \right].$$

Theorem 6.2 *Let* $|\lambda_1 - \theta_1| = \min_j |\lambda_1 - \theta_j|$. *Then we have*

$$|\lambda_1 - \theta_1| \le K\epsilon_{2m} \frac{\left(\sum_{i\neq 1} |x_{i1}|^2 + \sum_{i\neq 1} |u_{i1}|^2\right)^{1/2}}{|x_{11}|} \frac{\left(\sum_{i\neq 1} |y_{i1}|^2 + \sum_{i\neq 1} |v_{i1}|^2\right)^{1/2}}{|y_{11}|}$$

where

$$\epsilon_\ell = \min_{\deg(p)=\ell, p(\lambda_1)=1} \max_{i\neq 1} \{|p(\lambda_i)|, |p(\theta_i)|\}$$

and $K = \max_{i\neq 1}\{|\lambda_i - \theta_1|; |\theta_i - \theta_1|\}$.

7 Numerical Example

We present a numerical example in this section. Our example comes from a finite element discretization of the following problem from dissipative acoustics [7, 32].

Let $\Omega \subset \mathbb{R}^2$ be a rectangular cavity filled with an acoustic fluid (such as air), with one absorbing wall Γ_A and three reflecting walls Γ_R. Let $P(x,t)$ and $U(x,t)$ be the acoustic pressure and the fluid displacement, respectively. Also let ρ be the density of the fluid, and c the speed at which the fluid conducts sound. Then the behavior of the fluid satisfies the equations

$$\rho \frac{\partial^2 U}{\partial t^2} + \nabla P = 0 \qquad (7.1)$$

$$-\rho c^2 \text{ div } U = P \qquad (7.2)$$

with boundary conditions

$$U \cdot \nu = 0 \qquad \text{on } \Gamma_R \qquad (7.3)$$

$$\alpha U \cdot \nu + \beta \frac{\partial U}{\partial t} \cdot \nu = P \qquad \text{on } \Gamma_A \qquad (7.4)$$

where scalar constants α, β are related to the impedance of the absorbing material. As in [7], we choose $\rho = 1$ kg/m^3, $c = 340$ m/s, $\alpha = 5 \times 10^4$ N/m^3, $\beta = 200$ N s/m^3 for our test; this choice of α and β models a very viscous absorbing material.

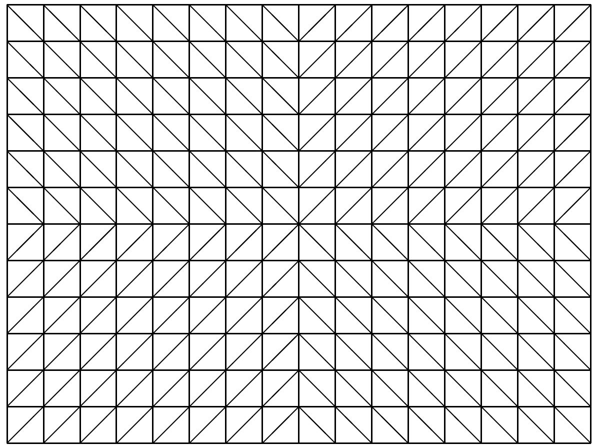

Figure 2: Triangulation of Ω with $N = 2$

We are interested in finding the damped vibration modes of the fluid, which are solutions of the form $U(x,t) = e^{\lambda t}u(x)$, $P(x,t) = e^{\lambda t}p(x)$. Then, equations (7.1) – (7.4) reduce to finding λ, p, u satisfying

$$\begin{aligned} \rho\lambda^2 u + \nabla p &= 0 && \text{in } \Omega \\ p &= -\rho c^2 \text{ div } u && \text{in } \Omega \\ p &= (\alpha + \lambda\beta) u \cdot \nu && \text{on } \Gamma_A \\ u \cdot \nu &= 0 && \text{on } \Gamma_R. \end{aligned}$$

This system can be converted to a variational formulation. Let $\mathcal{V} = \{v \in H(\text{div}, \Omega) : v \cdot \nu \in L^2(\partial\Omega) \text{ and } v \cdot \nu = 0 \text{ on } \Gamma_R\}$. The problem is equivalent to finding $\lambda \in \mathbb{C}$, nonzero $u \in \mathcal{V}$ so that

$$\lambda^2 \int_\Omega \rho u \cdot v + \lambda \int_{\Gamma_A} \beta u \cdot \nu \, v \cdot \nu + \int_{\Gamma_A} \alpha u \cdot \nu \, v \cdot \nu + \int_\Omega \rho c^2 \text{ div } u \text{ div } v = 0 \tag{7.5}$$

for all $v \in \mathcal{V}$. Using finite elements to approximate $\mathcal{V}$ by $\mathcal{V}_h = \text{span}\{\phi_1, \ldots, \phi_n\}$ yields the $n \times n$ quadratic matrix eigenvalue problem

$$\lambda^2 Mx + \lambda\beta Fx + (\alpha F + K)x = 0, \tag{7.6}$$

where

$$M_{ij} = \int_\Omega \rho\phi_i \cdot \phi_j, \; K_{ij} = \int_\Omega \rho c^2 \text{ div } \phi_i \text{ div } \phi_j, \; F_{ij} = \int_{\Gamma_A} \phi_i \cdot \nu \, \phi_j \cdot \nu.$$

To avoid spurious eigenvalues caused by discretization, it is suggested in [7] to use lowest order Raviart-Thomas finite elements [26]. Each basis element ϕ_i is a vector-valued function with piecewise constant divergence on each triangle of the mesh and $\phi_i \cdot \nu$ constant along each edge. With a natural choice of the basis, each finite element corresponds to an edge in the interior or on the absorbing boundary Γ_A. We use a triangulation of Ω with

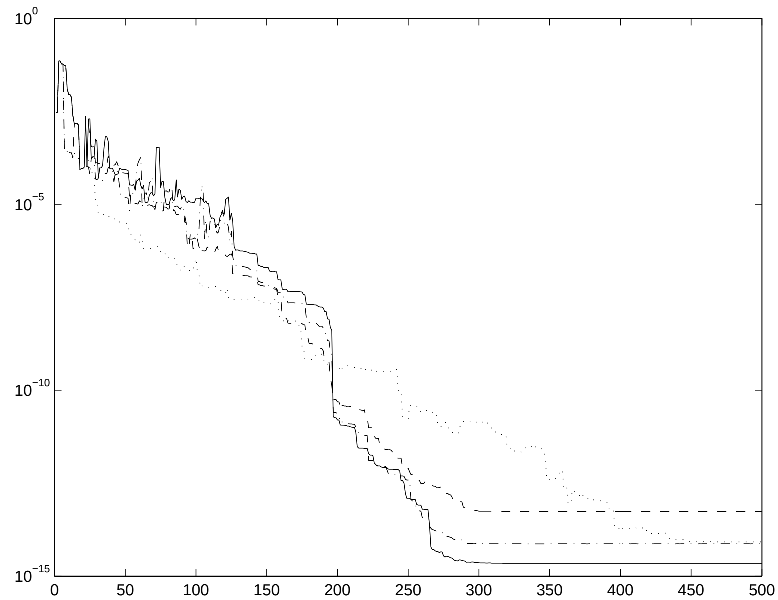

Figure 3: Relative residual norms for selected eigenvalues.

$6N$ edges along the vertical sides and $8N$ edges along the horizontal sides; a model with 9168 degrees of freedom is obtained with the choice of the parameter $N = 8$ (Figure 2).

Let $A = M$, $B = \beta F$, $C = \alpha F + K$ and write (7.6) as the symmetric quadratic eigenvalue problem

$$(\lambda^2 A + \lambda B + C)x = 0. \tag{7.7}$$

Note that A is symmetric positive definite, and B, C are positive semidefinite matrices. The eigenvalues of interest are those with smaller imaginary parts. They have real parts between -250 and -320. We use a shift-and-invert transformation with the shift $\sigma = -253$ to get

$$(\mu^2 \hat{A} + \mu \hat{B} + \hat{C})x = 0;$$

for this choice of σ, $\hat{A}$ remains positive definite. Therefore, we can take the Cholesky decomposition $\hat{A} = LL^T$ and construct the equivalent monic problem

$$(\mu^2 I + \mu(L^{-1}\hat{B}L^{-T}) + (L^{-1}\hat{C}L^{-T}))u = 0 \tag{7.8}$$

where $\hat{A} = LL^T$ is the Cholesky decomposition. Algorithm 4.2 is now applied to (7.8) in a symmetric Lanczos type process to get a basis V_k and banded $k \times k$ matrices T_a, T_b. It follows that if (θ_i, u_i) is an eigenpair to the projected problem

$$(\mu^2 I_k + \mu T_{a(1:k,1:k)} + T_{b(1:k,1:k)})u = 0, \tag{7.9}$$

then $(\lambda_i, x_i) = (\sigma + 1/\mu_i, z_i/\|z_i\|)$ is an approximate eigenpair to the original problem (7.7), where $z_i = L^{-T}V_{k(:,1:i)}u_i$.

Figure 3 shows the results of computing four eigenvalues of (7.7) using Algorithm 4.2. For each eigenvalue and each $i = 1, \ldots, 500$, we compute the corresponding approximate eigenpair (λ_i, x_i) by linearizing and solving (7.9) with the MATLAB `eigs` function. The resulting relative residual norms $r_i = \frac{\|(\lambda_i^2 A + \lambda_i B + C)x_i\|}{|\lambda_i|^2\|A\| + |\lambda_i|\|B\| + \|C\|}$ are plotted in Figure 3.

The following table lists the values of the four selected eigenvalues. For each eigenvalue λ_i (as computed by MATLAB), the table gives the type of line used in Figure 3 to plot the corresponding residual norms, the number of matrix-vector products required to obtain a relative residual norm $r_i < 10^{-8}$, and the error of the corresponding eigenvalue $\hat{\lambda}_i$ at that point.

Eigenvalue λ_i	Plot line	Matrix-vector products	$\|\lambda_i - \hat{\lambda}_i\|$
$-259.23 + 813.27i$	dotted	318	1.7×10^{-8}
$-320.54 + 267.66i$	dashed	322	1.0×10^{-8}
-342.15	dash-dot	356	8.8×10^{-9}
-296.66	solid	386	3.8×10^{-9}

8 Conclusions

We have presented basic Arnoldi and Lanczos type processes for a monic matrix polynomial with large and sparse coefficient matrices. These processes operate on the same space as the matrix polynomial live and the reduced problem are a matrix polynomial itself. This has the advantage of preserving certain properties of the original system in the reduced systems and the process could hold the key on practical applications where such a feature is a necessity.

What we have presented here are basic ideas. Robust implementations of these new algorithms will require carefully dealing with many (subtle) technical details. Bearing similarity in nature to the standard Arnoldi and Lanczos processes, these new Arnoldi and Lanczos type processes could incorporate many proven techniques developed over the years for the former. We shall leave these matters to future investigations.

Acknowledgment

This work started as search for an answer to a question posed to the author by Professor Zhaojun Bai of the University of California at Davis. The question was to produce a physically meaningful reduced order model when the original model is defined by a quadratic matrix polynomial. The authors are indebted to Prof. Bai for the question.

References

[1] W. E. Arnoldi, *The principle of minimized iterations in the solution of the matrix eigenvalue problem*, Quarterly of Applied Mathematics, 9 (1951), pp. 17–29.

[2] Z. Bai, *Personal communication.* 2000.

[3] Z. Bai, D. Day, and Q. Ye, *ABLE: an adaptive block Lanczos method for non-Hermitian eigenvalue problems*, SIAM Journal on Matrix Analysis and Applications, 20 (1999), pp. 1060–1082.

[4] Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. van der Vorst (editors), *Templates for the solution of Algebraic Eigenvalue Problems: A Practical Guide*, SIAM, Philadelphia, 2000.

[5] Z. Bai and R. Freund, *Eigenvalue-based characterization and test for positive realness of scalar transfer functions*, IEEE on Automat. Control, 45 (2000), pp. 2396–2402.

[6] Z. Bai and Q. Ye, *Error estimation of the padé approximation of transfer functions via the Lanczos process*, Electronic Transactions on Numerical Analysis, 7 (1998), pp. 1–17.

[7] A. Bermúdez, R. G. Durán, R. Rodríguez, and J. Solomin, *Finite element analysis of a quadratic eigenvalue problem arising in dissipative acoustics*, SIAM Journal on Numerical Analysis, 38 (2000), pp. 267–291.

[8] J. Demmel, *Applied Numerical Linear Algebra*, SIAM, Philadelphia, 1997.

[9] R. J. Duffin, *The rayleigh-ritz method for dissipative or gyroscopic systems*, Quart. Appl. Math., 18 (1960), pp. 215–221.

[10] T. Ericsson and A. Ruhe, *The spectral transformation Lanczos method for the numerical solution of large sparse generalized symmetric eigenvalue problems*, Mathematics of Computation, 35 (1980), pp. 1251–1268.

[11] P. Feldman and R. W. Freund, *Efficient linear circuit analysis by Padé approximation via the Lanczos process*, IEEE Trans. Computer-Aided Design, 14 (1995), pp. 639–649.

[12] R. W. Freund, M. H. Gutknecht, and N. M. Nachtigal, *An implementation of the look-ahead Lanczos algorithm for non-Hermitian matrices*, SIAM J. Sci. Comput., 14 (1993), pp. 137–158.

[13] K. Gallivan, E. Grimme, and P. van Dooren, *Asymptotic waveform evaluation via a Lanczos method*, Appl. Math. Lett., 7 (1994), pp. 75–80.

[14] A. Geist, *Reduction of a general matrix to tridiagonal form*, SIAM J. Matrix Anal. Appl., 12 (1991), pp. 362–373.

[15] I. Gohberg, P. Lancaster, and L. Rodman, *Matrix Polynomials*, Academic Press, New York, 1982.

[16] G. H. Golub and C. F. van Loan, *Matrix Computations*, Johns Hopkins University Press, Baltimore, Maryland, 3rd ed., 1996.

[17] M. H. Gutknecht, *A complete theory of the unsymmetric Lanczos process and related algorithms: Part I*, SIAM Journal on Matrix Analysis and Applications, 13 (1992), pp. 594–639.

[18] ——, *A complete theory of the unsymmetric Lanczos process and related algorithms: Part II*, SIAM Journal on Matrix Analysis and Applications, 15 (1994), pp. 15–58.

[19] C. Lanczos, *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators*, Journal of Research of the National Bureau of Standards, 45 (1950), pp. 255–282.

[20] R.-C. Li, *Test positive realness of a general transfer function matrix*, Technical Report 2000-20, Department of Mathematics, University of Kentucky, 2000.

[21] R.-C. Li and Q. Ye, *A Krylov subspace method for quadratic matrix polynomials with application to constrained least squares problems*, SIAM J. Matrix Anal. Appl., (2003). to appear.

[22] K. Meerbergen, *Locking and restarting quadratic eigenvalue solvers*, SIAM J. Sci. Comput., 22 (2001), pp. 1814–1839.

[23] B. N. Parlett, *Reduction to tridiagonal form and minimal realization*, SIAM Journal on Matrix Analysis and Applications, 13 (1992), pp. 567–593.

[24] ——, *The Symmetric Eigenvalue Problem*, SIAM, Philadelphia, 1998. This SIAM edition is an unabridged, corrected reproduction of the work first published by Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1980.

[25] B. N. Parlett, D. R. Taylor, and Z. A. Liu, *A look-ahead Lanczos algorithm for unsymmetric matrices*, Mathematics of Computation, 44 (1985), pp. 105–124.

[26] P. A. Raviart and J. M. Thomas, *A mixed finite element method for second order elliptic problems*, in Mathematical Aspects of Finite Element Methods, vol. 606 of Lecture Notes in Mathematics, Springer-Verlag, Berlin, 1977, pp. 292–315.

[27] A. Ruhe, *Rational Krylov sequence methods for eigenvalue computation*, Linear Algebra and Its Applications, 58 (1984), pp. 391–405.

[28] Y. Saad, *Numerical Methods for Large Eigenvalue Problems*, Manchester University Press, Manchester, UK, 1992.

[29] ——, *Iterative Methods for Sparse Linear Systems*, PWS Publishing Company, Bpston, MA, 1996.

[30] G. Sleijpen, J. Booten, D. Fokkema, and H. van der Vorst, *Jacobi-Davidson type methods for generalized eigenproblems and polynomial eigenproblems*, BIT, 36 (1996), pp. 593–633.

[31] D. C. Sorensen, *Implicit application of polynomial filters in a k-step Arnoldi method*, SIAM Journal on Matrix Analysis and Applications, 13 (1992), pp. 357–385.

[32] F. Tisseur and K. Meerbergen, *The quadratic eigenvalue problem*, SIAM Review, 43 (2001), pp. 235–386.

[33] H. Weiss, Q. Wang, and J. L. Speyer, *System characterization of positive real conditions*, IEEE Transactions on Automatic Control, 39 (1994), pp. 541–544.

[34] Q. Ye, *A convergence analysis for nonsymmetric Lanczos algorithms*, Mathematics of Computation, 56 (1991), pp. 677–691.

[35] ——, *A breakdown-free variation of the nonsymmetric Lanczos algorithms*, Mathematics of Computation, 62 (1994), pp. 179–207.