The Structure Exploiting Arnoldi Algorithm for Model Order Reduction of General Higher-Order Linear Dynamical Systems

by

Jeffrey A. Anderson

B.S. Math (University of California, Santa Barbara) 2007

M.A. Math (University of California, Davis) 2010

DISSERTATION

Submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

MATHEMATICS

in the

OFFICE OF GRADUATE STUDIES

of the

UNIVERSITY OF CALIFORNIA

DAVIS

Approved:

Professor Roland Freund, Chair

Professor Thomas Strohmer

Professor Raissa D'Souza

Committee in Charge

2013

To my loving family: past, present and future

Contents

Jeffrey A. Andersson
September 2013
Mathematics

The Structure Exploiting Arnoldi Algorithm for Model Order Reduction of General Higher-Order Linear Dynamical Systems

Abstract

One possible model reduction technique for large-scale higher-order linear dynamical systems is to transform a given higher-order system into an equivalent first-order formulation. The desired reduced-order models are then constructed by employing Krylov subspace-based reduction methods to the resulting first-order system. Since the Krylov subspaces associated with these first-order systems can be viewed as multiple copies of the same underlying space, this technique can be improved. This research focuses on creating a Structure Exploiting Arnoldi (SEA) algorithm that generates an orthonormal basis for the aforementioned multiple-copied subspace. The SEA algorithm is a modification of the Arnoldi algorithm designed to perform updates using key properties of Krylov subspaces associated with general higher-order linear dynamical systems. Applications of this research include reduced-order modeling of both ℓth-order linear dynamical systems as well as systems of first-order integro-differential-algebraic equations.

Acknowledgments

I thank Roland Freund. I am grateful for his patience throughout my education at UC Davis. He leads by example using curiosity, intuition and persistence to solve mathematical problems. Professor Freund continually demonstrates a dedication to high-quality applied mathematics and encourages his students to develop as independent researchers. He also gave a most precious gift to me as I entered this field: a fertile problem. I appreciate his research methods and respect the contributions he has made to the field of Numerical Linear Algebra. Most of all, I have enjoyed the process of developing research under his guidance.

There are many other mentors I wish to thank. I am grateful to Raissa D'Souza for introducing me to the research process. Her support and gentle mentorship during my second year at Davis encouraged me to reach for the best within myself. I thank Thomas Strohmer for being my steward on multiple occasions and for guiding me in the right directions. I thank Jesus DeLoera for being a champion for graduate students, for being a staunch supporter of mathematics education and for being the Chair of my PhD Qualifying Examination. I am grateful to Becca Thomases for her support and kindness throughout my PhD career. She has an amazing ability to inspire confidence in those with whom she works. Finally, I thank the many Professors who have encouraged my progress along the way. These include and are not limited to Naoki Saito, Albert Fannjiang, Matthias Koeppe, Jon McCammond, Jeffrey Stopple, and Claude Bonnet. Thanks also to the National Science Foundation for the financial support (NSF Grant No. DMS-0636297) and to the UC Davis VIGRE committee.

Great thanks are due to the staff members of the UC Davis Mathematics Department including Tina Denena, Perry Gee, Celia Davis, Alla Savrasova, Zachariah Johnson, Leng Lai, Marianne Waage and Gladis Lopez. This group of people work tirelessly to support and guide graduate students like myself through the UC Davis PhD Program in Mathematics and Applied Mathematics. They have directly contributed to many of my successes at UC Davis. They are the strength behind the scenes of the UC Davis Mathematics Department.

On a personal note, I thank Lupita Aguilar for all her support. Lupita helps strengthen my commitment to excellence and my purpose to thrive. She reminds me to slow down and enjoy the pleasures of life. She is my closest friend and one of my deep motivations to be the best I can be.

Thanks to my family for setting the foundation of dedication, persistence, self-confidence and love. Richard, Darcy, Kevin and Sarah Anderson each helped me summon courage in completing my PhD journey. These wonderful people remind me of who I am, where I come from and of what I am capable. Thanks also to Caliope Calomiris Hooper, Benjamin Muniz, Donald Hooper Sr., and Donald Hooper Jr. Each of these people remind me how special it is to be part of a loving family. I also thank the entire Aguilar family for accepting me into the group. I feel very lucky to be a member of the Aguilar tribe.

Last, I thank my friends and colleagues in this program. With their help, I have been able to stay healthy, happy, and progress far beyond what I could have done alone. In particular, thanks to James Dowell, Eileen and Josh Harwood, William E. Tavernetti, Adam Sorkin, and Yoshihiro and Emi Nakahara. I thank my colleagues in the UC Davis Mathematics Department for the countless lessons taught. Included in this group are Jiawei Zhang, Yuji Nakatsukasa, Joohee Hong, Efrem Rensi, Owen Lewis, Robert Hildebrand, Jeff Irion, Charles Brummitt, David Renfrew, Mohamed Omar, Mihaela Ifrim, Rohit Thomas, and Patrick Dragon. I am grateful for everything these individuals have taught me.

CHAPTER 1

Introduction

1.1 The Basic Idea of Reduced-Order Modeling

Complex physical systems are usually dynamic, or changing in time. Powerful mathematical tools exist to accurately model such real-world phenomena. Linear dynamical systems are a popular type of model used in a wide variety of science and engineering problems. These are a set of linear differential-algebraic equations that describe the evolution of a physical or engineered system over a period of time.

The state-space description of a first-order time-invariant linear dynamical system is given by

$$\begin{aligned} \mathcal{E}\frac{d}{dt}\mathbf{x}(t) &= \mathcal{A}\mathbf{x}(t) + \mathcal{B}\mathbf{u}(t), \\ \mathbf{y}(t) &= \mathcal{L}\mathbf{x}(t) + \mathcal{D}\mathbf{u}(t), \end{aligned} \tag{1.1}$$

where $\mathcal{A}, \mathcal{E} \in \mathbb{R}^{N\times N}$, $\mathcal{B} \in \mathbb{R}^{N\times m}$, $\mathcal{L} \in \mathbb{R}^{p\times N}$ and $\mathcal{D} \in \mathbb{R}^{p\times m}$. The positive integer N is said to be the state-space dimension of the linear dynamical system (1.1) and matrices $\mathcal{E}, \mathcal{A}, \mathcal{B}, \mathcal{L}$, and $\mathcal{D}$ are invariant with respect to time. The time-domain, vector-valued functions $\mathbf{x}(t), \mathbf{u}(t)$, and $\mathbf{y}(t)$ are of appropriate sizes with $t \in [t_0, \infty)$. This set of equations (1.1) is known as an m-input p-output time-invariant first-order linear dynamical system.

Numerical simulations of linear dynamical systems lead scientists to understand and even predict the behavior of the underlying physical process [**Dat04**]. Response characteristics of the linear dynamical system under various input conditions are computed using algorithms from numerical linear algebra [**GL96**], [**ABB$^+$99**]. This computational approach to solving scientific problems has been effective in a wide variety of applications.

However, collections of practical problems result in large-scale linear dynamical systems. Direct numerical simulation of models with large state-space dimension is expensive in computational costs and storage requirements. Such systems have so many variables that accurate computations using

the original formulation (1.1) are prohibitive, even on very powerful computers. One popular remedy to overcome this difficulty is model order reduction [**BMS05**], [**SVDVR08**].

The central focus of reduced-order modeling is to create a low-dimensional approximation to the original system (1.1) in the form

$$\begin{aligned} \mathcal{E}_n \frac{d}{dt}\mathbf{x}_n(t) &= \mathcal{A}_n \mathbf{x}_n(t) + \mathcal{B}_n \mathbf{u}_n(t), \\ \mathbf{y}_n(t) &= \mathcal{L}_n \mathbf{x}_n(t) + \mathcal{D}_n \mathbf{u}_n(t), \end{aligned} \tag{1.2}$$

where all dimensions N are replaced with $n \ll N$. Ideally, the model produced by a particular reduction technique (1.2) should exhibit some combination of the following three desirable qualities:

(1) Accuracy: Accurate reduced-order models very closely approximate the original system. In particular, valuable reduced-order modeling techniques ensure that the response characteristics $\mathbf{y}_n(t)$ of the reduced system (1.2) accurately describe the output $\mathbf{y}(t)$ of the large system (1.1). In other words, the relative error

$$\frac{\|\mathbf{y}(t) - \mathbf{y}_n(t)\|_2}{\|\mathbf{y}(t)\|_2}$$

is guaranteed to be "small."

(2) Preservation of important system properties: Properties of linear dynamical systems include stability and passivity. Stable linear systems exhibit similar output behavior under a collection of perturbed input data $\hat{\mathbf{u}}(t) = \mathbf{u}(t) + \delta\mathbf{z}$ for vectors $\mathbf{z} \in \mathbb{C}^m$. This property ensures that engineered systems will behave in a controllable manner even when the external environment perturbs input variables in unpredictable but measurable ways. Passive systems do not generate energy. For physical systems whose corresponding models are passive or stable, techniques that maintain these properties during dimension reduction are desirable.

(3) Computational efficiency and stability: Any reduction technique that expedites the numerical simulations of large-scale systems should be computationally efficient and numerically stable.

Real value in model reduction techniques comes from computationally sound methods that run quickly, produce accurate results and preserve important properties of the underlying system. It is in this spirit that the results presented in this thesis are developed.

1.2 Moment-Matching Model Order Reduction

As discussed above, the main idea of reduced-order modeling is to replace a given set of linear differential-algebraic equations (1.1) by a system of the same form with much smaller state-space dimension (1.2). The challenge is to choose an approximation technique that produces matrices $\mathcal{A}_n, \mathcal{E}_n \in \mathbb{R}^{n\times n}$, $\mathcal{B}_n \in \mathbb{R}^{n\times m}$, $\mathcal{L}_n \in \mathbb{R}^{p\times n}$ and $\mathcal{D}_n \in \mathbb{R}^{p\times m}$ for the reduced system (1.2) that are "good" enough to effectively simulate the output characteristics of the original equations (1.1).

One powerful strategy to measure the quality of a reduced-order model relies on the notion of a transfer function. To form the transfer function corresponding to the state-space description of the linear dynamical system (1.1), apply the Laplace transform to each time-dependent, vector-valued function. The Laplace transform of a time-domain, vector-valued function $\mathbf{f} : [t_0, \infty) \to \mathbb{R}^N$ is the function $F(s)$ defined by

$$F(s) = \int_{t_0}^{\infty} \mathbf{f}(t) e^{-st} dt,$$

for some $s \in \mathbb{C}$. Assuming zero initial conditions, this translates the collection of differential-algebraic equations (1.1) into a system of purely algebraic equations given by

$$\begin{aligned} s\mathcal{E}X(s) &= \mathcal{A}X(s) + \mathcal{B}U(s), \\ Y(s) &= \mathcal{L}X(s) + \mathcal{D}U(s), \end{aligned} \tag{1.3}$$

with $X(s), U(s)$ and $Y(s)$ defined by taking the Laplace transform of the functions $\mathbf{x}(t), \mathbf{u}(t)$ and $\mathbf{y}(t)$ respectively.

Assume that the matrix pencil $s\mathcal{E} - \mathcal{A}$ is singular for finitely many $s \in \mathbb{C}$ and solve for $X(s)$ in terms of $U(s)$. Making the appropriate substitutions, the transfer function $H(s)$ can be stated as

$$Y(s) = H(s)U(s) \quad \text{where} \quad H(s) = \mathcal{D} + \mathcal{L}(s\mathcal{E} - \mathcal{A})^{-1}\mathcal{B}.$$

The function $H(s)$ has poles wherever $s\mathcal{E} - \mathcal{A}$ is singular. Choose a scalar $s_0 \in \mathbb{C}$ such that $(s_0\mathcal{E} - \mathcal{A})^{-1}$ exists and rewrite the transfer function of the original system (1.1) expanded about the point $s_0 \in \mathbb{C}$ as

$$H(s) = \mathcal{D} + \mathcal{L}(I + (s - s_0)\mathcal{M})^{-1}\mathcal{R}, \tag{1.4}$$

where

$$\mathcal{M} = (s_0\mathcal{E} - \mathcal{A})^{-1}\mathcal{E} \qquad \text{and} \qquad \mathcal{R} = (s_0\mathcal{E} - \mathcal{A})^{-1}\mathcal{B}. \tag{1.5}$$

A formal Neumann series expansion of $(I + (s - s_0)\mathcal{M})^{-1}$ yields

$$H(s) = \mathcal{D} + \mathcal{L}\left(\sum_{k=0}^{\infty} (-1)^k \mathcal{M}^k (s - s_0)^k\right)\mathcal{R}. \tag{1.6}$$

Similarly, the Taylor series expansion of $H(s)$ about the point s_0 is defined as

$$H(s) = \mu_0 + \mu_1(s - s_0) + \mu_2(s - s_0)^2 + \cdots + \mu_j(s - s_0)^j + \cdots . \tag{1.7}$$

The coefficients μ_i are known as the moments of H about the point s_0 for every nonnegative integer i. In the case of m-input, p-output systems, each μ_i is a $p \times m$ matrix. Explicit formulas for these moments are given by

$$\mu_0 = \mathcal{D} + \mathcal{L}\mathcal{R}, \qquad \text{and} \qquad \mu_k = (-1)^k \mathcal{L}\mathcal{M}^k\mathcal{R}, \qquad \text{for } k \in \mathbb{N}, \tag{1.8}$$

and arise by comparing the Neumann series expansion (1.6) with the Taylor series expansion (1.7) of $H(s)$ about the point s_0.

A similar argument yields the transfer function $H_n(s)$ for the reduced-order model (1.2)

$$H_n(s) = \mathcal{D}_n + \mathcal{L}_n(s\mathcal{E}_n - \mathcal{A}_n)^{-1}\mathcal{B}_n, \tag{1.9}$$

$$= \mathcal{D}_n + \mathcal{L}_n(I + (s - s_0)\mathcal{M}_n)^{-1}\mathcal{R}_n, \tag{1.10}$$

where

$$\mathcal{M}_n = (s_0\mathcal{E}_n - \mathcal{A}_n)^{-1}\mathcal{E}_n \qquad \text{and} \qquad \mathcal{R}_n = (s_0\mathcal{E}_n - \mathcal{A}_n)^{-1}\mathcal{B}_n. \tag{1.11}$$

Both $H(s)$ and $H_n(s)$ are matrix-valued, rational functions of dimension $p \times m$.

The matrix-valued rational function $H : \mathbb{C} \to (\mathbb{C} \cup \infty)^{p \times m}$ gives an input-output relation of the original system (1.1) in frequency domain without explicit reference to the state-space variables. Moment-matching reduced-order modeling techniques aim to approximate this input-output relation by creating a reduced system (1.2) with transfer function $H_n(s)$ (1.9) matching a significant number of leading moments of $H(s)$. One popular moment-matching scheme comes by Padé approximation theory [**BGM10**]. A Padé model $H_n(s)$ at the expansion point s_0 matches a maximum number of leading moments of the Taylor series expansion of $H(s)$ (1.7). In other words, a Padé model has the property that

$$H(s) = H_n(s) + \mathcal{O}\big((s - s_0)^{q(n)}\big)$$

with $q(n)$ maximal.

In the early 1990s, the Asymptotic Waveform Evaluation (AWE) algorithm emerged as a popular reduced-order modeling technique for applications in VLSI interconnect analysis [**PR90**]. This algorithm produced Padé models for transfer functions arising from large-scale time-invariant first-order linear dynamical systems corresponding to linear RCL circuits with a single input ($m = 1$) and a single output ($p = 1$). By using explicit moment matching, the AWE algorithm and its variants [**RBR92**], [**RP94**], [**CN94**] enabled engineers designing VLSI circuits to accurately simulate the IC interconnect network for important problems including clock delay calculations and circuit noise analysis [**HRBR92**], [**KGP94**], [**Fri01**].

As AWE became a popular tool for IC interconnect analysis it was used to model larger linear circuits; however increased demands on AWE demonstrated a major draw back. The AWE algorithm and its variants matched moments by explicitly calculating the matrix powers for each μ_k (1.8) for $0 \leq k \leq 2n - 1$, with n being the order of the reduced-model. The particular computational method of calculating Padé models via explicit moment matching resulted in extremely ill-conditioned numerical computations. Even though Padé approximants guaranteed strong results in theory, reduced-order models generated by AWE-inspired algorithms calculated in finite-precision arithmetic did not match more than a limited number of leading moments.

The proliferation of this method and corresponding numerical instability encouraged the next major wave of innovation in moment-matching reduced-order modeling. In 1995, Feldman and Freund introduced the Padé Via Lanczos (PVL) algorithm [**FF95b**] to produce Padé models using projection onto Krylov subspaces rather than explicit moment matching. By recognizing the connection between Padé approximations [**BGM10**] and the Lanczos algorithm [**Lan50**], PVL enjoyed all the benefits of AWE while addressing some of the major drawbacks. Because PVL-based model order reduction used Krylov subspace methods, the numerical implementations were both efficient and numerically stable.

From PVL came a number of variants directly relating to Padé models. For example, the MPVL algorithm generalized PVL by allowing multi-input, multi-output systems to be analyzed with a single implementation of this band reduced-order modeling technique [**FF95a**]. Later, MPVL was adapted into a band Lanczos algorithm that included look-ahead Lanczos technology [**ABFH00**]. Similarly, the SyPVL [**FF96**] and SyMPVL [**FF97**] algorithms were specialized versions of the PVL and MPVL algorithms, respectively, for symmetric input data.

While the PVL algorithm and its variants were accurate and numerically stable reduction techniques, in general they did not preserve important system properties like stability and passivity of the original dynamical systems. The challenge of creating reduced-order modeling techniques that reliably preserved system properties spurred a third wave of innovation in moment-matching reduced-order modeling during the late 1990s and early 2000s. Krylov subspace-based methods like PRIMA [**OCP97**] and ENOR [**She99**] arose to produce Padé-type models that were provably passive.

Padé-type models of state-space dimension n are similar to Padé models except they match a fewer number of of leading moments of the Taylor series expansion (1.7). A Padé-type model at the expansion point s_0 of the original system (1.1) is a reduced-order model (1.2) of state-space dimension n, with transfer function $H_n(s)$ (1.9), that matches a number of leading moments of $H(s)$. Specifically, a Padé-type model satisfies

$$H(s) = H_n(s) + \mathcal{O}\big((s - s_0)^{q'}\big) \tag{1.12}$$

with $1 \leq q' \leq q(n)$, where $q(n)$ is the best possible approximation available by Padé models.

Although the PRIMA algorithm was the first popular method for creating provably passive reduced-order models, it ignored inherent structure in the matrices defining the state-space equations (1.1). Recognizing this oversight, Freund introduced the SPRIM algorithms [**Fre04**], [**Fre11**] as reduction techniques designed to handle the special structure of system matrices. By incorporating a mechanism to exploit the structure of J-Hermitian systems [**Fre08**], SPRIM boasted twice the accuracy of PRIMA for the same computational cost. Moreover, SPRIM maintained the underlying system properties in ways that PRIMA and its variants did not.

This thesis carries on where SPRIM left off. The goal of this research is to produce a theoretically more general and computationally more efficient version of the SPRIM algorithm for model order reduction applied to general higher-order linear dynamical systems. The improvements suggested in this thesis rely on special structure of Krylov subspaces associated with linearized higher-order linear dynamical systems. By exploiting such structure, this thesis presents computationally efficient model order reduction algorithms to generate Padé-type models that preserve the structure of the original system equations.

1.3 Assumptions on Notation

The mathematical results presented in this thesis follow the standard notation for numerical linear algebra and matrix computations [**GL96**]. The set of complex numbers is denoted by $\mathbb{C}$ while the set of real numbers is given by $\mathbb{R}$. Similarly, the collection of all $n \times m$ matrices with complex entries is denoted as $\mathbb{C}^{n\times m}$ while the set of all real $n \times m$ matrices is given by $\mathbb{R}^{n\times m}$. For $\mathcal{A}, \mathcal{E} \in \mathbb{C}^{n\times n}$ and $s \in \mathbb{C}$, the matrix pencil $s\mathcal{E} - \mathcal{A}$ is said to be regular if the matrix $s\mathcal{E} - \mathcal{A}$ is singular only for finitely many values $s \in \mathbb{C}$ [**GLR09**, Part II].

The transpose of an $n \times m$ matrix $A = [a_{jk}]$ is denoted as $A^T = [a_{kj}] \in \mathbb{C}^{m\times n}$. Similarly, the conjugate transpose is given by $A^H = [\overline{a_{kj}}]$. The individual entries of an $n \times m$ matrix A are complex numbers

$$A(i, j) = a_{ij}, \qquad \text{for } i = 1, 2, ..., n \text{ and } j = 1, ..., m.$$

The ith row of the matrix A is a $1 \times m$ vector denoted using the colon notation

$$A(i,:) = \begin{bmatrix} a_{i1} & a_{i2} & \cdots & a_{im} \end{bmatrix}, \qquad \text{for } i = 1, 2, ..., n.$$

The jth column of A is an $n \times 1$ vector given by

$$A(:,j) = \begin{bmatrix} a_{1j} \\ \vdots \\ a_{nj} \end{bmatrix}, \qquad \text{for } j = 1, 2, ..., m.$$

Boldface variables, such as $\mathbf{x}$, denote vectors whose size will be explicitly stated if it is not immediately clear from context. When referring to a sequence of vectors, subscripts denote individual sequence elements $\{\mathbf{q}_i\}_{i=1}^{n}$ where $\mathbf{q}_i$ is the ith element in this sequence for $i = 1, 2, ..., n$. The span of a set of k vectors $\mathbf{x}_1, ..., \mathbf{x}_k$ is given by

$$\text{span } \{\mathbf{x}_1, ..., \mathbf{x}_k\} = \langle \mathbf{x}_1, ..., \mathbf{x}_k \rangle .$$

Matrices in math calligraphy such as $\mathcal{M}$ and $\mathcal{R}$ will be used to describe data arising in applications related to reduced-order modeling. Capital romanized letters such as F, S and X will be used to denote matrices whose sizes will be stated if not clear from the context. Given a matrix $A \in \mathbb{C}^{n \times m}$ and a vector $\mathbf{v} \in \mathbb{C}^k$ whose $k \leq m$ entries are strictly increasing integers with $1 \leq v_1 < \cdots < v_k \leq m$, $A(:, \mathbf{v})$ is the matrix of columns of A whose indices are pointed to by the entires of $\mathbf{v}$, i.e.

$$A(:, \mathbf{v}) = \begin{bmatrix} A(:, v_1) & A(:, v_2) & \cdots & A(:, v_k) \end{bmatrix}.$$

The class of matrices discussed in Chapter 3 will be defined using the Kronecker product [**Gra81**]. Given $A \in \mathbb{C}^{n \times m}$ and $B \in \mathbb{C}^{p \times q}$, the Kronecker product of A and B is the matrix

$$A \otimes B = [a_{jk} B] \ \in \mathbb{C}^{(n \cdot p) \times (m \cdot q)}.$$

Chapter 4 includes an important reference to symmetric positive semi-definite matrices. The notation $\mathcal{E} \succeq 0$ means that square matrix $\mathcal{E}$ is symmetric positive semi-definite. The matrix I_n will

denote the $n \times n$ identity matrix and the zero matrix will be denoted by 0 with dimensions apparent from the context.

1.4 Thesis Outline

The first chapter of this thesis introduces moment-matching reduced-order modeling and gives an overview of the state-of-the-art. Chapter 2 provides the background material and appropriate references for this research including a brief introduction to block Krylov subspaces, the linearization of higher-order linear dynamical systems into first-order form, and pertinent Krylov subspace methods for model order reduction. The main contributions of this thesis are found in Chapter 3 and Chapter 4. Chapter 3 presents the Structure Exploiting Arnoldi (SEA) algorithm for input matrices arising from the linearization of ℓth-order linear dynamical systems for $\ell \geq 2$. Both single-input and band versions of the SEA algorithm are included as well as a discussion of computational aspects of the implementations of these algorithms. Chapter 4 develops an analogous Structure Exploiting Arnoldi algorithm applied to matrices arising from the linearization of special second-order systems known as first-order integro-differential-algebraic equations. Chapter 5 concludes with a summary of the work presented and a discussion of further research problems related to this project.

CHAPTER 2

Krylov Subspace Methods for Reduced-Order Modeling

Many popular methods for executing moment-matching reduced-order modeling rely on Krylov subspace-based projections. Important background material on block Krylov subspaces, first-order and higher-order linear dynamical systems, and moment-matching dimension reduction are included in this chapter. Each of these concepts is directly related to the development of the Structure Exploiting Arnoldi algorithm.

2.1 Block Krylov Subspaces

Krylov subspace-based reduction techniques are based on Krylov subspaces [**Saa03**, Chapter 6], [**Fre05**], [**Hog06**, Chapter 49]. This section presents a brief overview of block Krylov subspaces induced by a matrix $\mathcal{M} \in \mathbb{C}^{N\times N}$ and starting vectors $\mathcal{R} \in \mathbb{C}^{N\times m}$. All relevant definitions and results for Krylov subspaces follow from the more general theory presented below by setting $m = 1$.

Definition 2.1.1. Let $\mathcal{M} \in \mathbb{C}^{N\times N}$ and let $\mathcal{R} \in \mathbb{C}^{N\times m}$. The $N \times mN$ **block Krylov matrix** induced by $\mathcal{M}$ and $\mathcal{R}$ is defined by

$$K_N(\mathcal{M}, \mathcal{R}) = \left[\begin{array}{cccc} \mathcal{R} & \mathcal{M}\mathcal{R} & \cdots & \mathcal{M}^{N-1}\mathcal{R} \end{array} \right]. \tag{2.1}$$

The $N \times mN$ block Krylov matrix $K_N(\mathcal{M}, \mathcal{R})$ will not have linearly independent columns in general.

Proposition 2.1.2. Let $\mathcal{M} \in \mathbb{C}^{N\times N}$ and let $\mathbf{r}_i$ denote the ith column of $\mathcal{R} \in \mathbb{C}^{N\times m}$ for each $i \in \{1, 2, ..., m\}$. Starting at the leftmost column vector of $K_N(\mathcal{M}, \mathcal{R})$ and moving to the right, index each column vector using integers $1, 2, ..., Nm$. If column $(n-1)m + i$, given by $\mathcal{M}^{n-1}\mathbf{r}_i$, is in the span of the first $(n-1)m + i - 1$ columns, then all columns $\mathcal{M}^k\mathbf{r}_i$ are also in the same span for $n \leq k \leq N - 1$,.

Since the columns of the block Krylov matrix (2.1) will not be linearly independent in general, there is a smallest k_0 such that the block $\mathcal{M}^{k_0-1}\mathcal{R}$ contains a new basis vector for the column span of $K_N(\mathcal{M},\mathcal{R})$ and all columns of $\mathcal{M}^k\mathcal{R}$ are linearly dependent on columns of $\left[\begin{array}{cccc} \mathcal{R} & \mathcal{M}\mathcal{R} & \cdots & \mathcal{M}^{k_0-1}\mathcal{R} \end{array}\right]$. for $k_0 \leq k \leq N-1$.

Definition 2.1.3. Let $\mathcal{M} \in \mathbb{C}^{N\times N}$ and $\mathcal{R} \in \mathbb{C}^{N\times m}$. The **block grade of the block Krylov subspace induced by $\mathcal{M}$ and $\mathcal{R}$** is the smallest natural number k_0 such that

$$\text{range}\left(\left[\begin{array}{cccc} \mathcal{R} & \mathcal{M}\mathcal{R} & \cdots & \mathcal{M}^{k_0-1}\mathcal{R} \end{array}\right]\right) = \text{range}\left(K_N(\mathcal{M},\mathcal{R})\right). \tag{2.2}$$

Block Krylov subspaces induced by a matrix $\mathcal{M}$ and a starting block $\mathcal{R}$ are defined using linearly independent columns of $K_N(\mathcal{M},\mathcal{R})$. To this end, scan the columns of the block Krylov matrix starting from the leftmost column and moving to the right, deleting each column that is linearly dependent on previous columns. The process of eliminating linearly dependent columns is known as *exact deflation* and provides a concise framework to describe the essential spanning information for block Krylov matrices.

Definition 2.1.4. The **deflated block Krylov matrix induced by $\mathcal{M}$ and $\mathcal{R}$** is

$$\left[\begin{array}{cccc} \mathcal{R}_1 & \mathcal{M}\mathcal{R}_2 & \cdots & \mathcal{M}^{k_0-1}\mathcal{R}_{k_0} \end{array}\right] \tag{2.3}$$

where k_0 is the block grade of the block Krylov subspace induced by $\mathcal{M}$ and $\mathcal{R}$. Set $\mathcal{R}_0 = \mathcal{R}$. For each $j = 1, ..., k_0$, $\mathcal{R}_j \in \mathbb{C}^{N\times m_j}$ is a submatrix of $\mathcal{R}_{j-1} \in \mathbb{C}^{N\times m_{j-1}}$. Moreover, $\mathcal{R}_j \neq \mathcal{R}_{j-1}$ if, and only if, exact deflation is necessary in the j^{th} Krylov block $\mathcal{M}^{j-1}\mathcal{R}_{j-1}$. In particular $\mathcal{R}_j = \mathcal{R}_{j-1}E_j$ with $E_j \in \mathbb{C}^{m_{j-1}\times m_j}$ and $m_j \leq m_{j-1}$ for $j = 1, 2, ..., k_0$. E_j is the deflated identity matrix obtained by deleting the $m_{j-1} - m_j$ columns of $I_{m_{j-1}}$ corresponding to columns of $\mathcal{M}^{j-1}\mathcal{R}_{j-1}$ that can be written as linear combinations of basis vectors in previous blocks of the deflated block Krylov matrix.

The sequence of dimensions $\{m_j\}_{j=1}^{k_0}$ encodes the number of linearly independent columns of $K_N(\mathcal{M},\mathcal{R})$ provided by each block $\mathcal{M}^{j-1}\mathcal{R}$ for $j = 1, 2, ..., k_0$. Notice that

$$m \geq m_1 \geq m_2 \geq \cdots \geq m_{k_0} \geq 1.$$

Let N_0 be the number of columns of the deflated block Krylov matrix (2.3) and define the function $n : \{1, 2, ..., k_0\} \to \mathbb{N}$ as

$$n = n(k) = n_k = m_1 + m_2 + \cdots + m_k \tag{2.4}$$

for $k = 1, 2, ..., k_0$. From above, $n(k_0) = N_0 = m_1 + m_2 + \cdots + m_{k_0} \leq N$. The difference between the block Krylov matrix and the deflated block Krylov matrix is best illustrated with an example including nontrivial exact deflation patterns.

Example 2.1.5. Let $\mathcal{M} \in \mathbb{C}^{15\times 15}$ be a random matrix and let $m = 5$. Suppose $\mathbf{r}_1, \mathbf{r}_3 \in \mathbb{C}^{15}$ are random vectors. Initialize the three other starting vectors as follows

$$\mathbf{r}_2 = \mathbf{r}_1, \qquad \mathbf{r}_4 = \mathcal{M}^2\mathbf{r}_1, \qquad \mathbf{r}_5 = \mathcal{M}^3\mathbf{r}_1.$$

Because there are five starting vectors in the first block, subsequent blocks of $K_N(\mathcal{M}, \mathcal{R})$ also have five columns. The initial assumptions on the starting block $\mathcal{R}$ guarantee that exact deflation occurs in each of the first three blocks of the block Krylov matrix (2.1). A visual representation of the first four blocks of the block Krylov matrix (2.1) is helpful in identifying exact deflation:

$$\begin{pmatrix} \mathbf{R}_1 & \mathbf{R}_2 & \mathbf{R}_3 & \mathbf{R}_4 & \mathbf{R}_5 \end{pmatrix}$$
$$\begin{pmatrix} \mathcal{M}\mathbf{R}_1 & \mathcal{M}\mathbf{R}_2 & \mathcal{M}\mathbf{R}_3 & \mathcal{M}\mathbf{R}_4 & \mathcal{M}\mathbf{R}_5 \end{pmatrix}$$
$$\begin{pmatrix} \mathcal{M}^2\mathbf{R}_1 & \mathcal{M}^2\mathbf{R}_2 & \mathcal{M}^2\mathbf{R}_3 & \mathcal{M}^2\mathbf{R}_4 & \mathcal{M}^2\mathbf{R}_5 \end{pmatrix}$$
$$\begin{pmatrix} \mathcal{M}^3\mathbf{R}_1 & \mathcal{M}^3\mathbf{R}_2 & \mathcal{M}^3\mathbf{R}_3 & \mathcal{M}^3\mathbf{R}_4 & \mathcal{M}^3\mathbf{R}_5 \end{pmatrix}.$$

The color coding indicates the exact deflation pattern necessary to produce the deflated block Krylov matrix (2.3) associated with this starting data. Red highlighting identifies the first instance for which the columns of the block Krylov matrix $K_N(\mathcal{M}, \mathcal{R})$ associated with each $\mathbf{r}_i$ require exact

deflation. The blue highlighting marks vectors that are linearly dependent on previous columns based on Proposition 2.1.2.

Using this visual, it is possible to pick out the linearly independent vectors in the first four blocks and generate the corresponding columns of the deflated block Krylov matrix:

$$\left[\begin{array}{cccc|cccc|cc|cc} \mathbf{r}_1 & \mathbf{r}_3 & \mathbf{r}_4 & \mathbf{r}_5 & \mathcal{M}\mathbf{r}_1 & \mathcal{M}\mathbf{r}_3 & \mathcal{M}\mathbf{r}_5 & & \mathcal{M}^2\mathbf{r}_3 & \mathcal{M}^2\mathbf{r}_5 & \mathcal{M}^3\mathbf{r}_3 & \mathcal{M}^3\mathbf{r}_5 \end{array}\right]$$

The elimination matrices used to create the first four blocks of the deflated block Krylov matrix are

$$E_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad E_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \qquad E_3 = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad E_4 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

In this case, $m = 5, m_1 = 4, m_2 = 3, m_3 = 2 = m_4 = \cdots = m_{k_0}$ and $k_0 = 6$. For a general random matrix, $N_0 = 15$.

Definition 2.1.6. Let $\mathcal{M} \in \mathbb{C}^{N\times N}$ and $\mathcal{R} \in \mathbb{C}^{N\times m}$. For $1 \leq n \leq N_0$, the n**th block Krylov subspace** induced by $\mathcal{M}$ and $\mathcal{R}$, denoted as $K(\mathcal{M}, \mathcal{R}, n)$, is the $n-$dimensional subspace of $\mathbb{C}^N$ spanned by the first n columns of the deflated block Krylov matrix (2.3).

Practical algorithms designed to construct basis matrices for $K(\mathcal{M}, \mathcal{R}, N_0)$ execute in finite-precision arithmetic and will not, in general, be able to determine exact deflation. Instead, these numerical methods proceed by deleting vectors that are "almost" linearly dependent on previous vectors, a process known as *inexact deflation*. Theoretic results, in contrast, are most easily stated assuming exact deflation. Unless explicitly stated otherwise, all results involving block Krylov subspaces presented in this thesis are formulated assuming exact deflation only.

Block Krylov subspaces provide the theoretic foundations for multiple-input, multiple-output, moment-matching reduced-order modeling techniques. Many of these projection-based methods generate basis vectors for the block Krylov subspace induced by $\mathcal{M}$ and $\mathcal{R}$ onto which key system

matrices are projected. Bases for $K(\mathcal{M}, \mathcal{R}, N_0)$ are useful for a number of reasons and deserve special attention.

Definition 2.1.7. Let $\mathcal{M} \in \mathbb{C}^{N\times N}$ and $\mathcal{R} \in \mathbb{C}^{N\times m}$. A matrix

$$\mathcal{V} = \left[\begin{array}{cccc} \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_{N_0} \end{array} \right] \in \mathbb{C}^{N\times N_0}$$

is a **basis matrix** of the block Krylov subspaces induced by $\mathcal{M}$ and $\mathcal{R}$ if

$$\mathcal{K}(\mathcal{M}, \mathcal{R}, n) = \text{ range } \left(\left[\begin{array}{cccc} \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_n \end{array} \right] \right)$$

for all $n = 1, 2, ..., N_0$.

Basis matrices are powerful because they relate the theory of block Krylov subspaces to the computational aspects of basis generation.

Proposition 2.1.8. Let $\mathcal{V}$ be any basis matrix of the block Krylov subspaces induced by $\mathcal{M}$ and $\mathcal{R}$. Then, there exists a nonsingular, upper-triangular matrix $\mathcal{U}$ such that

$$\mathcal{V} = \left[\begin{array}{cccc} \mathcal{R}_1 & \mathcal{M}\mathcal{R}_2 & \cdots & \mathcal{M}^{k_0-1}\mathcal{R}_{k_0} \end{array} \right] \mathcal{U}.$$

Another nice property of basis matrices is that any two basis matrices for the same block Krylov subspaces are related by a nonsingular, upper-triangular matrix.

Proposition 2.1.9. For any two basis matrices $\mathcal{V}_1$ and $\mathcal{V}_2$ of $K(\mathcal{M}, \mathcal{R}, N_0)$, there exists a nonsingular, upper-triangular matrix $\mathcal{U}$ such that $\mathcal{V}_1 = \mathcal{V}_2\ \mathcal{U}$.

In the case that matrix $\mathcal{R}$ contains a single starting vector ($m = 1$), Hessenberg matrices can be used to identify basis matrices for Krylov subspaces induced by $\mathcal{M}$ and $\mathcal{R}$.

Proposition 2.1.10. Let $\mathcal{M}$ be an arbitrary $N \times N$ matrix. Let $\widehat{A}_{n+1} = \left[\widehat{A}_n \quad \hat{\mathbf{a}}_{n+1} \right]$ be an $N \times (n+1)$ rectangular matrix satisfying $\mathcal{M}\widehat{A}_n = \widehat{A}_{n+1}\widehat{H}_n$ for an $(n+1) \times n$ Hessenberg matrix $\widehat{H}_n$. Then there is an upper-triangular matrix $\mathcal{U}_n$ such that

$$\widehat{A}_n\, \mathcal{U}_n = \left[\hat{\mathbf{a}}_1 \quad \mathcal{M}\hat{\mathbf{a}}_1 \quad \cdots \quad \mathcal{M}^{n-1}\hat{\mathbf{a}}_1 \right].$$

Furthermore, if the first $n-1$ subdiagonal elements of $\widehat{H}_n$ are nonzero, the upper-triangular matrix $\mathcal{U}_n$ is nonsingular and the range $(\widehat{A}_n) = K(\mathcal{M}, \hat{\mathbf{a}}_1, n)$.

A nice proof for this proposition can be found in Bai's SOAR paper [**BS05b**].

For general $\mathcal{R} \in \mathbb{C}^{N\times m}$, a band Hessenberg matrix $\widetilde{H_{n_k}}$ can be used to identify basis matrices for $K(\mathcal{M}, \mathcal{R}, N_0)$. The subdiagonal bandwidth structure of $\widetilde{H_{n_k}}$ relates to the exact deflation pattern for $K_N(\mathcal{M}, \mathcal{R})$. Brute-force exact deflation requires a scan of each and every column of the $N \times mN$ block Krylov matrix induced by $\mathcal{M}$ and $\mathcal{R}$. However, an improved method for discovering the exact deflation pattern in the $(k+1)$st block $\mathcal{M}^k\mathcal{R}$ relies on the elimination matrices $\{E_j\}_{j=1}^{k}$ for $k = 1, 2, ..., k_0 - 1$. Begin this method by constructing the elimination matrix E_1 guaranteeing the proper exact deflation pattern for starting block $\mathcal{R}$. Assuming the elimination matrices $E_1, ..., E_k$ are known for some $k = 1, 2, ..., k_0 - 1$, the $(k + 1)$st elimination matrix can be determined by scanning the last m_k columns of

$$\begin{bmatrix} \mathcal{R}_1 & \mathcal{M}\mathcal{R}_2 & \cdots & \mathcal{M}^{k-1}\mathcal{R}_k & \mathcal{M}^k\mathcal{R}_k \end{bmatrix}$$

and identifying the columns of $\mathcal{M}^k\mathcal{R}_k$ that are linearly dependent on previous columns.

Let $n(k) = n_k = m_1 + m_2 + \cdots m_k$ for $k = 1, 2, ..., k_0$. For $k = 1, 2, ..., k_0 - 1$, the subdiagonal bandwidth structure of band Hessenberg matrix $\widetilde{H_{n_k}} \in \mathbb{C}^{n_{k+1}\times n_k}$ relates to this alternative method for detecting the exact deflation pattern of the matrix $K_N(\mathcal{M}, \mathcal{R})$. In particular, partition the band Hessenberg matrix $\widetilde{H_{n_k}}$ into subblocks

$$\widetilde{H_{n_k}} = \begin{bmatrix} \widetilde{H}_{11} & \widetilde{H}_{12} & \cdots & \widetilde{H}_{1k} \\ \widetilde{H}_{21} & \widetilde{H}_{22} & \cdots & \widetilde{H}_{2k} \\ 0 & \widetilde{H}_{32} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \widetilde{H}_{kk} \\ 0 & \cdots & 0 & \widetilde{H}_{k+1,k} \end{bmatrix},$$

where $\widetilde{H}_{ij} \in \mathbb{C}^{m_i\times m_j}$ for $i = 1, 2, ..., k + 1$ and $j = 1, 2, ..., k$. If $m_{j+1} = m_j$, the subblock $\widetilde{H}_{j+1,j}$ is an upper-triangular, nonsingular matrix. On the other hand, if exact deflation occurs in the $(j + 1)$st block, then $m_{j+1} < m_j$ and the matrix $\widetilde{H}_{j+1,j} \in \mathbb{C}^{m_{j+1}\times m_j}$ is a rectangular matrix in row echelon form with exactly m_{j+1} nonzero pivots. In this case, the ith pivot of $\widetilde{H}_{j+1,j}$ has the same column index as the ith nonzero column of E_{j+1}^T for $i = 1, ..., m_{j+1}$. The matrix $\widetilde{H_{n_k}}$ is said to be

a deflation-revealing Hessenberg matrix corresponding to the block Krylov subspaces induced by $\mathcal{M}$ and starting block $\mathcal{R}$.

Proposition 2.1.11. Let $\mathcal{M} \in \mathbb{C}^{N\times N}$ and $\mathcal{R} \in \mathbb{C}^{N\times m}$. Let k_0 be the block grade of the block Krylov subspace induced by $\mathcal{M}$ and $\mathcal{R}$ and let $\{E_j\}_{j=1}^{k_0}$ be the sequence of elimination matrices that indicate the proper exact deflation pattern for $K_N(\mathcal{M}, \mathcal{R})$. Let

$$n_k = m_1 + m_2 + \cdots + m_k$$

for $k = 1, 2, ..., k_0$. Suppose that $\mathcal{R}E_1 = \widehat{A}_1\mathcal{U}_{11}$ for an $m_1 \times m_1$ nonsingular, upper-triangular $\mathcal{U}_{11}$ and let $V_{n_k} \in \mathbb{C}^{N\times n_k}$ be a rectangular matrix such that

$$V_{n_k} = \begin{bmatrix} \widehat{A}_1 & \widehat{A}_2 & \cdots & \widehat{A}_k \end{bmatrix}$$

where $k = 1, 2, ..., k_0$ and $\widehat{A}_j \in \mathbb{C}^{N\times m_j}$ for $j = 1, 2, ..., k_0$. If

$$\mathcal{M}V_{n_k} = V_{n_{k+1}}\widetilde{H_{n_k}}$$

for a $(n_{k+1} \times n_k)$ deflation-revealing Hessenberg matrix $\widetilde{H_{n_k}}$ corresponding to the block Krylov subspaces induced by $\mathcal{M}$ and starting block $\mathcal{R}$, then there is an $n_k \times n_k$ upper-triangular matrix $\mathcal{U}_{n_k}$ such that

$$V_{n_k} = \begin{bmatrix} \mathcal{R}_1 & \mathcal{M}\mathcal{R}_2 & \cdots & \mathcal{M}^{k-1}\mathcal{R}_k \end{bmatrix} \mathcal{U}_{n_k}.$$

for $k = 1, 2, ..., k_0 - 1$. Furthermore, because the $m_2 + m_3 + \cdots + m_{k+1}$ subdiagonal pivots of $\widetilde{H_{n_k}}$ are nonzero, the matrix $\mathcal{U}_{n_k}$ is nonsingular and

$$\text{range } (V_{n_k}) = K(\mathcal{M}, \mathcal{R}, n_k).$$

Proof: This result is established by induction. The base case is $k = 2$ since $V_{n_1}\mathcal{U}_{11} = \mathcal{R}_1$ by assumption. Notice that

$$\begin{bmatrix} \mathcal{R}_1 & \mathcal{M}\mathcal{R}_2 \end{bmatrix} = \begin{bmatrix} \widehat{A}_1\mathcal{U}_{11} & \mathcal{M}\mathcal{R}_1E_2 \end{bmatrix} = \begin{bmatrix} \widehat{A}_1\mathcal{U}_{11} & \begin{bmatrix} \widehat{A}_1 & \widehat{A}_2 \end{bmatrix} \begin{bmatrix} \widetilde{H}_{11} \\ \widetilde{H}_{21} \end{bmatrix} \mathcal{U}_{11}E_2 \end{bmatrix}$$

Set $\mathcal{U}_{12} = \widetilde{H}_{11}\mathcal{U}_{11}E_2 \in \mathbb{C}^{m_1 \times m_2}$ and $\mathcal{U}_{22} = \widetilde{H}_{21}\mathcal{U}_{11}E_2 \in \mathbb{C}^{m_2 \times m_2}$. Matrix $\mathcal{U}_{22}$ is upper-triangular and nonsingular because $\widetilde{H}_{21}\mathcal{U}_{11}$ is in row echelon form and the linearly dependent columns of this matrix-matrix product are eliminated by multiplication with E_2. Thus

$$\begin{bmatrix} \mathcal{R}_1 & \mathcal{M}\mathcal{R}_2 \end{bmatrix} = \begin{bmatrix} \widehat{A}_1 & \widehat{A}_2 \end{bmatrix} \begin{bmatrix} \mathcal{U}_{11} & \mathcal{U}_{12} \\ 0 & \mathcal{U}_{22} \end{bmatrix}$$

and the base case is established. Assume that the relationship hold for $k-1$. If

$$E_{B_{p_k}} = \begin{bmatrix} E_2 & 0 & \cdots & 0 \\ 0 & E_3 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & E_k \end{bmatrix},$$

then

$$\begin{aligned} \begin{bmatrix} \mathcal{R}_1 & \mathcal{M}\mathcal{R}_2 & \cdots & \mathcal{M}^{k-1}\mathcal{R}_k \end{bmatrix} &= \left[\begin{array}{c|c} \widehat{A}_1\mathcal{U}_{11} & \mathcal{M}\begin{bmatrix} \mathcal{R}_1 & \mathcal{M}\mathcal{R}_2 & \cdots & \mathcal{M}^{k-2}\mathcal{R}_{k-1} \end{bmatrix} E_{B_{p_k}} \end{array} \right], \\ &= \left[\begin{array}{c|c} \widehat{A}_1\mathcal{U}_{11} & \mathcal{M}V_{n_{k-1}}\mathcal{U}_{n_{k-1}}E_{B_{p_k}} \end{array} \right], \\ &= V_{n_k} \begin{bmatrix} \mathcal{U}_{11} & \widetilde{H_{n_{k-1}}}\mathcal{U}_{n_{k-1}}E_{B_{p_k}} \\ 0 & \end{bmatrix} = V_{n_k}\mathcal{U}_{n_k}. \end{aligned}$$

The matrix $\mathcal{U}_{n_k} \in \mathbb{C}^{n_k \times n_k}$ is upper-triangular by the definition of the sparsity pattern of $\widetilde{H_{n_{k-1}}}$ combined with the structure of the matrix $E_{B_{p_k}}$. □

2.2 Krylov Subspace Methods

The typical basis used to establish the theory of block Krylov subspaces is defined using block matrices $\mathcal{M}^{i-1}\mathcal{R}$ for $i = 1, 2, ..., k_0$ and is numerically ill-conditioned. In contrast, a number of numerically stable techniques exist to create basis matrices for the nth block Krylov subspace induced by a matrix $\mathcal{M} \in \mathbb{C}^{N \times N}$ and a starting block $\mathcal{R} \in \mathbb{C}^{N \times m}$. Detailed reviews of Krylov subspace methods can be found in the literature [**Saa92**, Chapter 6], [**Wat07**, Chapter 9], [**Hog13**, Chapter 49].

The focus of this thesis will be on model reduction techniques based on the Arnoldi algorithm [**Arn51**] and the band Arnoldi algorithm [**Fre03a**]. The algorithms developed and presented in Chapter 3 and Chapter 4 of this thesis will also adapt some of the characteristics the SOAR algorithm [**BS05b**]. This section presents each of these algorithms and basic results are stated without proof.

2.2.1 The Arnoldi Algorithm

The Arnoldi algorithm builds an orthonormal basis for the Krylov subspace induced by a matrix $\mathcal{M} \in \mathbb{C}^{N\times N}$ and a single starting vector $\mathcal{R} \in \mathbb{C}^N$. In the nth iteration of the Arnoldi algorithm, a new candidate basis vector for the orthonormal basis of $K(\mathcal{M}, \mathcal{R}, n+1)$ is constructed by projecting an $\mathcal{M}$-multiple of the last basis vector against the orthogonal complement of the span of all previously constructed basis vectors. This projection is accomplished via a modified Gram-Schmidt orthogonalization.

Algorithm 1 Arnoldi

Input: Matrix $\mathcal{M} \in \mathbb{R}^{N\times N}$ and vector $\mathcal{R} \in \mathbb{R}^N$
Output: Basis V_{n+1} for $K(\mathcal{M}, \mathcal{R}, n+1)$ and Hessenberg $\widehat{H}_n \in \mathbb{C}^{(n+1)\times n}$.

1. Calculate $\mathbf{v}_1 \coloneqq \mathcal{R}/\|\mathcal{R}\|_2$
2. **for** $n = 1, 2, ...$ **do**
3. $\quad \hat{\mathbf{v}}_{n+1} \coloneqq \mathcal{M}\mathbf{v}_n$
4. $\quad$ **for** $j = 1, 2, ..., n$ **do**
5. $\quad\quad h_{jn} \coloneqq \hat{\mathbf{v}}_{n+1}^T \mathbf{v}_j$
6. $\quad\quad \hat{\mathbf{v}}_{n+1} \coloneqq \hat{\mathbf{v}}_{n+1} - h_{jn}\mathbf{v}_j$
7. $\quad$ **end for**
8. $\quad h_{n+1,n} \coloneqq \|\hat{\mathbf{v}}_{n+1}\|_2$
9. $\quad$ **if** $h_{n+1,n} = 0$ **then**
10. $\quad\quad$ STOP
11. $\quad$ **end if**
12. $\quad$ Set $\mathbf{v}_{n+1} \coloneqq \hat{\mathbf{v}}_{n+1}/h_{n+1,n}$
13. **end for**

At the end of n steps of the Arnoldi algorithm, the following relations hold

$$\mathcal{M}V_n = V_{n+1}\widehat{H}_n = V_n H_n + \mathbf{v}_{n+1}\mathbf{e}_n^T h_{n+1,n}, \qquad V_n^H \mathcal{M} V_n = H_n,$$

where H_n is the square matrix formed by deleting the last row of $\widehat{H}_n$ and $\mathbf{e}_n$ is the last column of the $n \times n$ identity matrix. Further, the Arnoldi algorithm stops at some step n if, and only if, $K(\mathcal{M}, \mathcal{R}, n)$ has dimension n and $K(\mathcal{M}, \mathcal{R}, n) = K(\mathcal{M}, \mathcal{R}, n+1)$. In this case, n is the grade of

the Krylov subspace induced by $\mathcal{M}$ and $\mathcal{R}$. If the Arnoldi algorithm stops during the nth iteration, the space $K(\mathcal{M}, \mathcal{R}, n)$ is invariant under multiplication by $\mathcal{M}$. By construction, the column space of V_n is the space $K(\mathcal{M}, \mathcal{R}, n)$ and the columns of V_n form an orthonormal set. In other words, $V_n^H V_n = I_n$.

2.2.2 The Band Arnoldi Algorithm

The band Arnoldi algorithm is a generalization of the Arnoldi algorithm for multiple starting vectors $\mathcal{R} \in \mathbb{C}^{N\times m}$. This technique enjoys improved theoretic properties over its block counterparts due to an accurate deflation technique and the maintenance of exact projection relations.

Algorithm 2 Band Arnoldi

Input: Matrix $\mathcal{M} \in \mathbb{R}^{N\times N}$ and block of starting vector $\mathcal{R} \in \mathbb{R}^{N\times m}$
Output: Orthonormal basis V_n for the block Krylov subspace $\mathcal{K}(\mathcal{M}, \mathcal{R}, n)$.

1. Set $\hat{\mathbf{v}}_i \coloneqq \mathbf{r}_i$ for $i = 1, 2, ..., m$
2. Set $m_c \coloneqq m$ and $\mathcal{I} \coloneqq \emptyset$
3. **for** $n = 1, 2, ...,$ until convergence or $m_c = 0$ **do**
4. Compute $\|\hat{\mathbf{v}}_n\|_2$ and decide if deflation is necessary
5. **if** Deflation is necessary **then**
6. Set and store $\hat{\mathbf{v}}_{n-m_c}^{\text{defl}} \coloneqq \hat{\mathbf{v}}_n$. Set $\mathcal{I} \coloneqq \mathcal{I} \cup \{n - m_c\}$ and $m_c \coloneqq m_c - 1$.
7. **if** $m_c = 0$ **then**
8. $n \coloneqq n - 1$ and STOP
9. **end if**
10. Set $\hat{\mathbf{v}}_i \coloneqq \hat{\mathbf{v}}_{i+1}$ for $i = n, n + 1, ..., n + m_c - 1$
11. Repeat line 4 - line 10
12. **end if**
13. Set $h_{n,n-m_c} \coloneqq \|\hat{\mathbf{v}}_n\|_2$ and $\mathbf{v}_n \coloneqq \hat{\mathbf{v}}_n / h_{n,n-m_c}$
14. **for** $j = n + 1, n + 2, ..., n + m_c - 1$ **do**
15. $h_{n,j-m_c} \coloneqq \mathbf{v}_n^T \hat{\mathbf{v}}_j$ and $\hat{\mathbf{v}}_j \coloneqq \hat{\mathbf{v}}_j - h_{n,j-m_c}\mathbf{v}_n$
16. **end for**
17. Compute $\hat{\mathbf{v}}_{n+m_c} \coloneqq \mathcal{M}\mathbf{v}_n$
18. **for** $j = 1, 2, ..., n$ **do**
19. $h_{j,n} \coloneqq \mathbf{v}_j^T \hat{\mathbf{v}}_{n+m_c}$ and $\hat{\mathbf{v}}_{n+m_c} \coloneqq \hat{\mathbf{v}}_{n+m_c} - h_{j,n}\mathbf{v}_j$
20. **end for**
21. Set $h_{ni} \coloneqq \mathbf{v}_n \mathbf{v}_i^{\text{defl}}$ for all $i \in \mathcal{I}$
22. Set $H_n \coloneqq \left[h_{jk}\right]_{j,k=1,2,...,n}$
23. Set $k_\rho \coloneqq m + \min\{0, n - m_c\}$ and $\rho_j \coloneqq \left[h_{j,k-m_c}\right]_{j,k=1,2,...,k_\rho}$
24. **if** n is large enough for desired application **then**
25. STOP
26. **end if**
27. **end for**

Assume that the band Arnoldi algorithm runs in exact arithmetic for n iterations. Let $V_n = \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_n \end{bmatrix}$ be the basis vectors stored in line 13 and denote the candidate vectors stored in the algorithm as $\hat{\mathbf{v}}_{n+1}, \hat{\mathbf{v}}_{n+2}, ..., \hat{\mathbf{v}}_{n+m_c}$. The integer m_c is the number of new candidate basis vectors remaining after the nth iteration. At the start of this algorithm, $m_c = m$ and this counter is decremented by one each time a deflation is encountered. The matrix V_n forms an orthonormal basis matrix for the block Krylov subspace $K(\mathcal{M}, \mathcal{R}, n)$. By construction $V_n^H V_n = I_n$ and $V_n^H \hat{\mathbf{v}}_{n+k} = \mathbf{0}$ for $k = 1, 2, ..., m_c$.

The deflation decision for the band Arnoldi algorithm consists of calculating the norm of $\hat{\mathbf{v}}_n$. Exact deflation occurs at step n of the band Arnoldi algorithm if, and only if, $\hat{\mathbf{v}}_n = \mathbf{0}$. In contrast, inexact deflation occurs in step n if, and only if, $\hat{\mathbf{v}}_n \neq \mathbf{0}$ but $\|\hat{\mathbf{v}}_n\|_2 \approx 0$. Thus, the decision whether or not to execute inexact deflation of the candidate vector $\hat{\mathbf{v}}_n$ is made by checking if

$$\|\hat{\mathbf{v}}_n\|_2 \leq \texttt{defl\_tol}, \tag{2.5}$$

where the deflation tolerance `defl_tol` is chosen appropriately.

The positive integer $m_1 \leq m$ is the number of linearly independent columns of the starting block $\mathcal{R}$. If $n > m_1$, then at the end of n steps of the band Arnoldi algorithm,

$$\begin{aligned} \mathcal{R} &= V_n \rho_n + \widehat{R}_m^{\text{defl}}, \\ \mathcal{M} V_n &= V_n H_n + \begin{bmatrix} \mathbf{0} & \cdots & \mathbf{0} & \hat{\mathbf{v}}_{n+1} & \hat{\mathbf{v}}_{n+2} & \cdots & \hat{\mathbf{v}}_{n+m_c} \end{bmatrix} + \widehat{V}_n^{\text{defl}}, \end{aligned}$$

where $\rho_n \in \mathbb{C}^{n \times m}$ and $H_n \in \mathbb{C}^{n \times n}$ is a band Hessenberg matrix.

The matrices $\widehat{R}_m^{\text{defl}}$ and $\widehat{V}_n^{\text{defl}}$ are zero only if exact deflation is executed. In contrast, if both exact and inexact deflation are used, the columns of $\widehat{V}_n^{\text{defl}}$ may be nonzero vectors that satisfy the condition of the inexact deflation check (2.5). Even using inexact deflation,

$$\rho_n = V_n^H \mathcal{R} \qquad \text{and} \qquad H_n = V_n^H \mathcal{M} V_n.$$

This is a feature of the projection executed in line 21 of the band Arnoldi algorithm.

The band of outermost subdiagonal entries of H_n ends each time a deflation occurs at which point the subdiagonal bandwidth of the matrix H_n decreases by one. If inexact deflation is used,

nonzero entries may fill the lower portion of the column of H_n marking the end of the outermost band of nonzero subdiagonal entries. These potentially nonzero entries are designed to track the projection coefficients corresponding to $V_n^H \widehat{V}_n^{\text{defl}}$ regardless of the deflation tolerance used to execute inexact deflation.

Analogously, matrix ρ_n is upper-triangular with the outermost band of diagonal entries ending each time deflation is encountered in the m columns of $\mathcal{R}$. When using inexact deflation, nonzero entries of ρ_n may fill the lower part of the columns of ρ_n whose indeces mark the columns of $\mathcal{R}$ on which inexact deflation was executed. These nonzero entries track projection coefficients corresponding to $V_n^H \widehat{R}_m^{\text{defl}}$.

Example 2.2.1. Suppose that the band Arnoldi algorithm is run for $n = 11$ iterations on the matrices related to Example 2.1.5. Specifically, let the entries of the random matrix $\mathcal{M} \in \mathbb{C}^{15\times 15}$ come from the standard normal distribution. Suppose $\mathcal{R} \in \mathbb{C}^{15\times 5}$ and assume that $\mathbf{r}_1$ and $\mathbf{r}_3$ are random vectors with entries from the standard normal distribution. Define

$$\mathbf{r}_2 = \mathbf{r}_1 + \delta_1 \mathbf{z}_1, \qquad \mathbf{r}_4 = \mathcal{M}^2 \mathbf{r}_1 + \delta_2 \mathbf{z}_2, \qquad \mathbf{r}_5 = \mathcal{M}^3 \mathbf{r}_1 + \delta_3 \mathbf{z}_3,$$

where $\delta_i = \mathcal{O}(\epsilon)$, ϵ is the value of machine epsilon and the entries of $\mathbf{z}_i$ come from the standard normal distribution. Choose a deflation tolerance `defl_tol` $= \mathcal{O}(\sqrt{\epsilon})$. After 11 iterations of the band Arnoldi algorithm, almost certainly there will be 11 basis vectors and 3 inexact deflations will have occurred. The sparsity structure of the resulting projection matrices $\rho_{11} \in \mathbb{C}^{11\times 5}$ and

$H_{11} \in \mathbb{C}^{11\times 11}$ generated by the band Arnoldi process is given as follows:

$$\begin{bmatrix}\rho_{11} & H_{11}\end{bmatrix} = \left[\begin{array}{ccccc|ccccc|ccc|cc|cc} + & * & * & * & * & * & * & * & * & * & * & * & * & * & * & * & * \\ 0 & \times & + & * & * & * & * & * & * & * & * & * & * & * & * & * & * \\ 0 & \times & 0 & + & * & * & * & * & * & * & * & * & * & * & * & * & * \\ 0 & \times & 0 & 0 & + & * & * & * & * & * & * & * & * & * & * & * & * \\ 0 & \times & 0 & 0 & 0 & + & * & * & * & * & * & * & * & * & * & * & * \\ 0 & \times & 0 & 0 & 0 & 0 & + & * & * & * & * & * & * & * & * & * & * \\ 0 & \times & 0 & 0 & 0 & 0 & 0 & \times & + & * & * & * & * & * & * & * & * \\ 0 & \times & 0 & 0 & 0 & 0 & 0 & \times & 0 & \times & + & * & * & * & * & * & * \\ 0 & \times & 0 & 0 & 0 & 0 & 0 & \times & 0 & \times & 0 & + & * & * & * & * & * \\ 0 & \times & 0 & 0 & 0 & 0 & 0 & \times & 0 & \times & 0 & 0 & + & * & * & * & * \\ 0 & \times & 0 & 0 & 0 & 0 & 0 & \times & 0 & \times & 0 & 0 & 0 & + & * & * & * \end{array}\right].$$

Entries marked by $*$ indicate potentially nonzero coefficients while $+$ indicates entries that are provably positive. The entries marked by $\times$ indicate coefficients that may be nonzero only for inexact deflation and are exactly zero for exact deflation. After 11 iterations of the band Arnoldi algorithm executed on this example, $m_c = m_c(11) = 2 = m - 3$.

2.2.3 The SOAR Algorithm

The SOAR algorithm is a customized version of the Arnoldi algorithm tailored to matrix $\mathcal{M}$ and single starting vector $\mathcal{R}$ with special structure. Assume that

$$\mathcal{M} = \begin{bmatrix}1\\0\end{bmatrix} \otimes \begin{bmatrix}M^{(1)} & M^{(2)}\end{bmatrix} + \begin{bmatrix}0 & 0\\ I_{n_0} & 0\end{bmatrix} \qquad \text{and} \qquad \mathcal{R} = \begin{bmatrix}1\\0\end{bmatrix} \otimes \mathbf{r}, \tag{2.6}$$

where $M^{(1)}, M^{(2)} \in \mathbb{C}^{n_0\times n_0}$ and $\mathbf{r} \in \mathbb{C}^{n_0}$. For all positive integers $k \geq 2$, define the sequence of vectors

$$\mathbf{w}_1 = \mathbf{r}, \qquad \mathbf{w}_2 = M^{(1)}\mathbf{w}_1, \qquad \mathbf{w}_k = M^{(1)}\mathbf{w}_{k-1} + M^{(2)}\mathbf{w}_{k-2}.$$

Definition 2.2.2. Let $M^{(1)}, M^{(2)} \in \mathbb{C}^{n_0 \times n_0}$ and suppose that $\mathbf{r} \in \mathbb{C}^{n_0}$ is a nonzero vector. Let S_n be the space

$$S_n = \langle \mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_n \rangle \tag{2.7}$$

and call this space the nth second-order Krylov subspace.

The SOAR algorithm creates an orthonormal basis for the space S_n.

Algorithm 3 Second-Order Arnoldi (SOAR)

Input: Matrices $M^{(1)}, M^{(2)} \in \mathbb{R}^{n_0 \times n_0}$ and single starting vector $\mathbf{r} \in \mathbb{R}^{n_0}$
Output: Matrix $\widehat{Q}_{n+1}$ whose columns span S_{n+1} (2.7).

1. Calculate $\hat{\mathbf{q}}_1 \coloneqq \mathbf{r}/\|\mathbf{r}\|_2$ and set $\mathbf{p}_1 \coloneqq \mathbf{0} \in \mathbb{R}^{n_0}$
2. **for** $n = 1, 2, ...$ **do**
3. $\quad \hat{\mathbf{q}}_{n+1} \coloneqq M^{(1)}\hat{\mathbf{q}}_j + M^{(2)}\mathbf{p}_j$
4. $\quad$ **for** $j = 1, 2, ..., n$ **do**
5. $\quad\quad h_{jn} \coloneqq \hat{\mathbf{q}}_{n+1}^T \hat{\mathbf{q}}_j$
6. $\quad\quad \hat{\mathbf{q}}_{n+1} \coloneqq \hat{\mathbf{q}}_{n+1} - h_{jn}\hat{\mathbf{q}}_j$
7. $\quad$ **end for**
8. $\quad h_{n+1,n} \coloneqq \|\hat{\mathbf{q}}_{n+1}\|_2$
9. $\quad$ **if** $h_{n+1,n} \neq 0$ **then**
10. $\quad\quad \hat{\mathbf{q}}_{n+1} \coloneqq \hat{\mathbf{q}}_{n+1}/h_{n+1,n}$
11. $\quad\quad \mathbf{p}_{n+1} \coloneqq \widehat{Q}_n \widehat{H}(2 : n+1, 1 : n)^{-1}\mathbf{e}_n$
12. $\quad$ **else**
13. $\quad\quad$ Reset $h_{n+1,n} = 1$
14. $\quad\quad \hat{\mathbf{q}}_{n+1} \coloneqq \mathbf{0}$
15. $\quad\quad \mathbf{p}_{n+1} \coloneqq \widehat{Q}_n \widehat{H}(2 : n+1, 1 : n)^{-1}\mathbf{e}_n$
16. $\quad\quad$ Check if this procedure should be stopped
17. $\quad$ **end if**
18. **end for**

If the SOAR procedure completes n iterations of the outer for-loop, then

$$\mathcal{M} \begin{bmatrix} \widehat{Q}_n \\ P_n \end{bmatrix} = \begin{bmatrix} \widehat{Q}_{n+1} \\ P_{n+1} \end{bmatrix} \widehat{H}_n, \tag{2.8}$$

where $\widehat{H}_n$ is the Hessenberg matrix with coefficients h_{jk} formed in each iteration for $j = 1, 2, ..., n+1$ and $k = 1, 2, ..., n$. Using Proposition 2.1.10, it can be shown that the SOAR algorithm stops at step n if, and only if, the Arnoldi algorithm stops at the same step n. The nonzero columns of the matrix $\widehat{Q}_{n+1}$ form an orthonormal basis for the space S_{n+1}. The paper introducing the SOAR algorithm [**BS05b**] has details about checking the stop condition and discusses an implementation of this

algorithm that avoid explicit storage of all auxiliary vectors $\mathbf{p}_{n+1} = \widehat{Q}_n \widehat{H}(2:n+1, 1:n)^{-1}\mathbf{e}_n$. These auxiliary vectors guarantee the SOAR matrix relation (2.8) which in turn assures that the essential information from the Krylov subspaces induced by $\mathcal{M}$ and $\mathcal{R}$ is available for the construction of $\widehat{Q}_{n+1}$.

The three Krylov subspace methods presented above lay the foundations for moment-matching model order reduction techniques developed in this thesis. The next section develops important concepts in linear dynamical systems.

2.3 Time-Invariant Linear Dynamical Systems

One class of mathematical equations amenable to the Structure Exploiting Arnoldi algorithm developed in this thesis are multi-input, multi-output time-invariant linear dynamical systems of order ℓ. These systems of differential-algebraic equations can be formulated using matrices, vectors and differential operators. Let m, p, and ℓ be integers greater than zero, where m denotes number of inputs, p represents the number of outputs and ℓ is the order of such systems.

Definition 2.3.1. An $m-$**input** $p-$**output time-invariant linear dynamical system of order** ℓ is a system of differential-algebraic equations (DAEs) in the form

$$\begin{aligned} P_\ell \frac{d^\ell}{dt^\ell}\mathbf{z}(t) + P_{\ell-1}\frac{d^{\ell-1}}{dt^{\ell-1}}\mathbf{z}(t) + \cdots + P_1\frac{d}{dt}\mathbf{z}(t) + P_0\mathbf{z}(t) &= B\mathbf{u}(t), \\ y(t) = D\mathbf{u}(t) + L_{\ell-1}\frac{d^{\ell-1}}{dt^{\ell-1}}\mathbf{z}(t) + \cdots + L_1\frac{d}{dt}\mathbf{z}(t) + L_0\mathbf{z}(t), \end{aligned} \tag{2.9}$$

with $P_i \in \mathbb{C}^{n_0 \times n_0}$ for $i = 0, 1, ..., \ell$, $B \in \mathbb{C}^{n_0 \times m}$, $D \in \mathbb{C}^{p \times m}$ and $L_j \in \mathbb{C}^{p \times n_0}$ for $j = 0, ..., \ell - 1$. The state-space dimension of this ℓth-order system is n_0. This system is fully determined by the specification of initial conditions given by

$$\left.\frac{d^j}{dt^j}\mathbf{z}(t)\right|_{t=t_0} = \mathbf{z}_0^{(j)}, \tag{2.10}$$

with $\mathbf{z}_0^{(j)} \in \mathbb{C}^{n_0}$ for $0, j = 1, 2, ..., \ell$.

The vector-valued input function $\mathbf{u}(t)$ maps the domain $[t_0, \infty)$ into $\mathbb{C}^m$ and represents the system input controlled by the design engineers. The function $\mathbf{z}(t)$ of state-space variables describes the internal state of the system at any time $t \in [t_0, \infty)$ via its component parts and maps into $\mathbb{C}^{n_0}$.

The output function $\mathbf{y}(t)$ from $[t_0, \infty)$ to $\mathbb{C}^p$ describes the systems interactions with the external environment at time t. Examples of higher-order systems can be found in the literature [**SC91**], [**BS05a**].

In this thesis, the matrix P_ℓ may be singular. The only assumption for matrices P_i with $i \in \{0, 1, ..., \ell\}$ is that the $n_0 \times n_0$ complex, matrix-valued polynomial

$$P(s) = s^\ell P_\ell + s^{\ell-1} P_{\ell-1} + \cdots + sP_1 + P_0,\ s \in \mathbb{C} \tag{2.11}$$

is regular.

An important special case of the general higher-order system (2.9) is a special second-order time-invariant linear dynamical system known as a system of first-order integro-differential-algebraic equations (integro-DAEs).

Definition 2.3.2. An $m-$**input** $p-$**output time-invariant system of first-order integro-DAEs** takes the form

$$\begin{aligned} P_1 \frac{d}{dt}\mathbf{z}(t) + P_0\mathbf{z}(t) + P_{-1}\int_{t_0}^{t} \mathbf{z}(\tau)d\tau &= B\mathbf{u}(t), \\ y(t) &= D\mathbf{u}(t) + L_0\mathbf{z}(t), \end{aligned} \tag{2.12}$$

with $P_i \in \mathbb{C}^{n_0 \times n_0}$ for $i \in \{-1, 0, 1\}$, $B \in \mathbb{C}^{n_0 \times m}$, $D \in \mathbb{C}^{p \times m}$ and $L_0 \in \mathbb{C}^{p \times n_0}$. The state-space dimension of this system is n_0. This system is fully determined by the specification of initial conditions given by

$$\mathbf{z}(t_0) = \mathbf{z}_0, \tag{2.13}$$

with $\mathbf{z}_0 \in \mathbb{C}^{n_0}$.

In the definition of a first-order integro-DAE, the matrix P_1 may be singular. The only assumption on the matrices $P_i \in \mathbb{C}^{n_0 \times n_0}$ for $i \in \{-1, 0, 1\}$ is that the matrix-valued function

$$Q(s) = sP_1 + P_0 + \frac{1}{s}P_1,\ s \in \mathbb{C} \tag{2.14}$$

is regular.

In the applications considered in this thesis, the matrices P_1 and P_0 are sparse. While the matrix P_{-1} may not be sparse, it can generally be written as a product of sparse matrices given by

$$P_{-1} = F_1 G F_2^H \tag{2.15}$$

or

$$P_{-1} = F_1 G^{-1} F_2^H, \text{ with } G \text{ nonsingular,} \tag{2.16}$$

where $F_1, F_2 \in \mathbb{C}^{n_0 \times \hat{n}}$ and $G \in \mathbb{C}^{\hat{n} \times \hat{n}}$ are sparse for some positive integer $\hat{n}$. In the case that $P_{-1} = F_1 G F_2^H$, the matrix G need not be invertible. One of the two factorizations (2.15) or (2.16) will always exist as indicated by the trivial factorization $F_1 = I_{n_0} = F_2$ and $G = P_{-1}$. For this reason, this thesis assumes for any first-order integro-DAEs, P_{-1} is given by one of the two products (2.15) or (2.16). Examples of these systems arise in real-world applications like VLSI interconnect analysis [**Fre00**], [**Fre03b**], [**Fre04**].

With the extra assumptions that $D = 0$ and that the input function $\mathbf{u}(t)$ is differentiable, the special second-order system (2.12) can be transformed into a "true" second-order system by differentiating both side of the first equation to yield

$$P_1 \frac{d^2}{dt^2}\mathbf{z}(t) + P_0 \frac{d}{dt}\mathbf{z}(t) + P_{-1}\mathbf{z}(t) = B\hat{\mathbf{u}}(t), \tag{2.17}$$

$$y(t) = L_0 \mathbf{z}(t), \tag{2.18}$$

with

$$\hat{\mathbf{u}}(t) = \frac{d}{dt}\mathbf{u}(t).$$

Although this transformation is convenient for relating special second-order systems (2.12) to general ℓth-order systems (2.9) with $\ell = 2$, it requires additional assumptions on the input function $\mathbf{u}(t)$. Moreover, if the original system of first-order integro-DAEs is passive, the transfer function $H(s)$ associated with the system of integro-DAEs is positive real while the transfer function of the transformed second-order system (2.17) will not be positive real in general. Because important system properties relating to passivity and the energy consumed by the system are encoded

in the integro-DAE formulation and not in the "equivalent" second-order formulation, first-order integro-DAEs receive special attention in Chapter 4.

Linearization of Higher-Order Systems

It is well known that any general higher-order system can be linearized to take the form of a first-order linear dynamical system.

Definition 2.3.3. The **state-space description of an m-input p-output time-invariant first-order linear dynamical system with N state variables** is given by the set of equations

$$\begin{aligned} \mathcal{E}\frac{d}{dt}\mathbf{x}(t) &= \mathcal{A}\mathbf{x}(t) + \mathcal{B}\mathbf{u}(t), \\ \mathbf{y}(t) &= \mathcal{L}\mathbf{x}(t) + \mathcal{D}\mathbf{u}(t), \end{aligned} \tag{2.19}$$

The system is completed with the initial conditions given by

$$\mathbf{x}(t)|_{t=t_0} = \mathbf{x}_0 \tag{2.20}$$

The matrix $\mathcal{A} \in \mathbb{C}^{N\times N}$ in the first equation of the state-space description (2.19) is known as the system matrix while $\mathcal{E} \in \mathbb{C}^{N\times N}$ is called the the descriptor matrix. The input matrix for the state-space description is $\mathcal{B} \in \mathbb{C}^{N\times m}$. These linear dynamical systems are classified as time-invariant because the matrices $\mathcal{A}$, $\mathcal{E}$, $\mathcal{B}$, $\mathcal{L}$, and $\mathcal{D}$ are invariant with respect to the variable t.

The input (control) vector is described by a time-dependent, vector-valued function $\mathbf{u} : [t_0, \infty) \to \mathbb{R}^m$ as is the state-space vector $\mathbf{x} : [t_0, \infty) \to \mathbb{R}^N$. Intuitively, the differential equation from the state-space description (2.19) indicates that changes in the internal state of the system are linearly related to the present state and the behavior of the input.

The matrix $\mathcal{L} \in \mathbb{C}^{p\times N}$ is called the output matrix and $\mathcal{D} \in \mathbb{C}^{p\times m}$ is known as the feed-through matrix. The output vector $\mathbf{y} : [t_0, \infty) \to \mathbb{R}^p$ represents the output of the system within its external environment. The second algebraic equation in the state-space description (2.19) reveals that the output of the system is a linear function of the internal state of the system and the behavior of the input.

If the matrix $\mathcal{E}$ is nonsingular, the linear system (2.19) is called regular. Regular systems can always be transformed into a system of ODEs by multiplying the first equation in the system (2.19)

by $\mathcal{E}^{-1}$ to generate

$$\frac{d}{dt}\mathbf{x}(t) = (\mathcal{E}^{-1}\mathcal{A})\mathbf{x}(t) + (\mathcal{E}^{-1}\mathcal{B})\mathbf{u}(t),$$
$$\mathbf{y}(t) = \mathcal{L}\mathbf{x}(t) + \mathcal{D}\mathbf{u}(t).$$

In contrast, if $\mathcal{E}$ is singular, the system is known as a descriptor system. Singular $\mathcal{E}$ matrices represent a system of differential-algebraic equations (DAEs). Accurately solving DAEs is much more difficult than solving ODEs. The algorithms developed in Chapters 3 and 4 of this work assume that the input data come from a descriptor system and require only that the matrix pencil $s\mathcal{E} - \mathcal{A}$ is regular.

Any ℓth-order linear dynamical system (2.9) can be transformed into an equivalent first-order system (2.19). Given a higher-order linear dynamical system in general form (2.9), (2.10), one possible set of linearization matrices is defined by

$$\mathcal{E} = \begin{bmatrix} \mathrm{I} & 0 & 0 & \cdots & 0 \\ 0 & \mathrm{I} & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & \mathrm{I} & 0 \\ 0 & \cdots & 0 & 0 & P_\ell \end{bmatrix}, \qquad \mathcal{A} = - \begin{bmatrix} 0 & -\mathrm{I} & 0 & \cdots & 0 \\ 0 & 0 & -\mathrm{I} & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & -\mathrm{I} \\ P_0 & P_1 & P_2 & \cdots & P_{\ell-1} \end{bmatrix}, \tag{2.21}$$

where the state-space variables are given by the equations

$$\mathbf{x}(t) = \begin{bmatrix} \mathbf{z}(t) \\ \frac{d}{dt}\mathbf{z}(t) \\ \frac{d^2}{dt^2}\mathbf{z}(t) \\ \vdots \\ \frac{d^{\ell-1}}{dt^{\ell-1}}\mathbf{z}(t) \end{bmatrix}, \qquad \mathbf{x}_0 = \begin{bmatrix} \mathbf{z}_0^{(0)} \\ \mathbf{z}_0^{(1)} \\ \mathbf{z}_0^{(2)} \\ \vdots \\ \mathbf{z}_0^{(\ell-1)} \end{bmatrix}, \tag{2.22}$$

and the matrices $\mathcal{B}, \mathcal{L}$ and $\mathcal{D}$ are defined

$$\mathcal{B} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ B \end{bmatrix}, \qquad \mathcal{L} = \begin{bmatrix} L_0 & L_1 & \cdots & L_{\ell-2} & L_{\ell-1} \end{bmatrix}, \qquad \mathcal{D} = D. \tag{2.23}$$

Verification that this linearization produces an equivalent first-order state-space description of a time-invariant linear dynamical system (2.19) follows using block matrix-matrix and block matrix-vector multiplication [**Fre05**]. This choice of linearization is not unique. More about linearization matrices can be found in the literature on matrix polynomials [**GLR09**].

Just as general ℓ-th order systems can be linearized, so too can any system of first-order integro-DAEs be transformed into an equivalent first-order system (2.19) [**Fre05**]. For example, consider a system of first-order integro-DAEs (2.12) with $P_{-1} = F_1 G F_2^H$. Set

$$\mathbf{x}_1(t) = \mathbf{z}(t) \qquad \text{and} \qquad \mathbf{x}_2(t) = F_2^H \int_{t_0}^{t} \mathbf{z}(\tau) d\tau$$

Then $\dfrac{d}{dt}\mathbf{x}_2(t) = F_2 \mathbf{x}_1(t)$. The linearization matrices for the equivalent first-order system are given by

$$\mathcal{A} = -\begin{bmatrix} P_0 & F_1 G \\ -F_2^H & 0 \end{bmatrix}, \qquad \mathcal{E} = \begin{bmatrix} P_1 & 0 \\ 0 & I_{n_0} \end{bmatrix}, \tag{2.24}$$

with state-space variables

$$\mathbf{x}(t) = \begin{bmatrix} \mathbf{x}_1(t) \\ \mathbf{x}_2(t) \end{bmatrix}, \qquad \mathbf{x}_0 = \begin{bmatrix} \mathbf{z}_0 \\ 0 \end{bmatrix}. \tag{2.25}$$

Matrices $\mathcal{B}, \mathcal{L}$, and $\mathcal{D}$ are given by

$$\mathcal{B} = \begin{bmatrix} B \\ 0 \end{bmatrix}, \qquad \mathcal{L} = \begin{bmatrix} L_0 & 0 \end{bmatrix}, \qquad \mathcal{D} = D. \tag{2.26}$$

On the other hand, suppose that the system of the first-order integro-DAE (2.12) is such that $P_{-1} = F_1 G^{-1} F_2^H$. To linearize this system, define the vector valued functions

$$\mathbf{x}_1(t) = \mathbf{z}(t) \qquad \text{and} \qquad \mathbf{x}_2(t) = G^{-1} F_2^H \int_{t_0}^{t} \mathbf{z}(\tau) d\tau,$$

and note that $G\dfrac{d}{dt}\mathbf{x}_2(t) = F_2 \mathbf{x}_1(t)$. Then, setting

$$\mathcal{A} = -\begin{bmatrix} P_0 & F_1 \\ -F_2^H & 0 \end{bmatrix}, \qquad \mathcal{E} = \begin{bmatrix} P_1 & 0 \\ 0 & G \end{bmatrix}, \tag{2.27}$$

with state-space variables

$$\mathbf{x}(t) = \begin{bmatrix} \mathbf{x}_1(t) \\ \mathbf{x}_2(t) \end{bmatrix}, \qquad \mathbf{x}_0 = \begin{bmatrix} \mathbf{z}_0 \\ 0 \end{bmatrix}, \tag{2.28}$$

and matrices $\mathcal{B}, \mathcal{L}$, and $\mathcal{D}$ given by

$$\mathcal{B} = \begin{bmatrix} B \\ 0 \end{bmatrix}, \qquad \mathcal{L} = \begin{bmatrix} L_0 & 0 \end{bmatrix}, \qquad \mathcal{D} = D.$$

produces an equivalent first order system (2.19).

Once a general higher-order linear dynamical systems has been transformed into its equivalent first-order form, model reduction techniques applicable to first-order systems can be used to process high-order data. This is a convenient way to create reduced order models for large-scale higher-order systems.

2.4 Krylov Subspace-Based Model Reduction

Padé-type models of first-order time-invariant linear dynamical system (2.19) are generated using basis vectors related to the Krylov-subspaces induced by matrix $\mathcal{M}$ and starting vectors $\mathcal{R}$ (1.5).

Theorem 2.4.1. Suppose $\mathcal{A}, \mathcal{E}, \mathcal{B}, \mathcal{L}$ and $\mathcal{D}$ form the state-space description of an m-input p-output time invariant first-order linear dynamical system with state-space dimension N (2.19). Choose $s_0 \in \mathbb{C}$ and $V_n \in \mathbb{C}^{N \times n_1}$ such that the matrices $s_0\mathcal{E} - \mathcal{A}$ and $s_0\mathcal{E}_n - \mathcal{A}_n$ are nonsingular. Let

$n = n(k) = m_1 + m_2 + \cdots + m_k$ for some $1 \leq k \leq k_0$ (2.4) and suppose

$$K(\mathcal{M}, \mathcal{R}, n) \subseteq \text{ range } \big(V_n\big). \tag{2.29}$$

Then the reduced-order model defined by the projections

$$\mathcal{E}_n = V_n^H \mathcal{E} V_n, \qquad \mathcal{A}_n = V_n^H \mathcal{A} V_n, \qquad \mathcal{B}_n = V_n^H \mathcal{B}, \qquad \mathcal{L}_n = \mathcal{L} V_n. \tag{2.30}$$

is a Padé-type model with

$$H_n(s) = H(s) + \mathcal{O}\big((s - s_0)^k\big).$$

A concise proof of this result can be found in the literature [**Fre08**]. Reduced-order models that satisfy the Padé-type moment-matching criteria (1.12) come from projecting the original system onto a basis V_n whose span contains $K(\mathcal{M}, \mathcal{R}, n)$. In effect, this theoretic result translates the challenge of creating accurate reduced-order models into a basis generation problem involving block Krylov subspaces. The popular (band) Arnoldi algorithm constructs an orthonormal basis for the proper Krylov subspaces (2.29).

Algorithm 4 (Band) Arnoldi-Based Dimension Reduction of First-Order Systems

Input: Matrices $\mathcal{A}, \mathcal{E}, \mathcal{B}, \mathcal{L}$, and $\mathcal{D}$ from first-order system (2.19)
Output: Reduced-order system (1.2)

1. Select an expansion point s_0 and choose reduction dimension n.
2. Run n steps of the (band) Arnoldi algorithm to produce orthonormal basis for the nth-order Krylov subspace induced by $\mathcal{M}$ and $\mathcal{R}$.
3. Obtain the reduced-order model for the original system by setting

$$\mathcal{A}_n = V_n^T \mathcal{A} V_n, \qquad \mathcal{E}_n = V_n^T \mathcal{E} V_n, \qquad \mathcal{B}_n = V_n^T \mathcal{B}, \qquad \mathcal{L}_n = \mathcal{L} V_n, \qquad \mathcal{D}_n = \mathcal{D}. \tag{2.31}$$

The matrix $\mathcal{M}$ is never formed explicitly in the implementation of (band) Arnoldi-based reduction. Instead, a sparse LU-factorization

$$s_0 \mathcal{E} - \mathcal{A} = P^T L U Q^T$$

is calculated using sparse matrix technology [**Dav06**]. Here, P and Q are permutation matrices, L is lower-triangle with ones on the main diagonal and U is upper-triangular with nonzero diagonal

entries. Matrix-vector multiplication of $\mathcal{M}$ with a single vector $\hat{\mathbf{v}}$ proceeds using the equation

$$\mathcal{M}\hat{\mathbf{v}} = Q\left(U^{-1}\left(L^{-1}\big(P\left(\mathcal{E}\,\hat{\mathbf{v}}\right)\big)\right)\right).$$

For most applications of interest, this equivalent method of multiplying by $\mathcal{M}$ is quite fast due to sparse matrix-vector multiplication and sparse forward and backward substitution. Finding the sparse LU factorization for $\mathcal{M}$ dominates the cost of computing (band) Arnoldi-based reduced order models.

Another feature of (band) Arnoldi-based reduction is that the reduced-order matrices (2.31) are not calculated using explicit projection onto the span of V_n. Instead, the desired reduced-order model comes from the (band) Hessenberg matrix $H_n \in \mathbb{C}^{n\times n}$ generated by the (band) Arnoldi algorithm. Recall that the transfer function $H(s)$ is created by applying the Laplace transform to the state-space description of the first-order linear dynamical system (2.19) with zero initial condition. Then, choosing $s_0 \in \mathbb{C}$ such that $s_0\mathcal{E} - \mathcal{A}$ is nonsingular, the Taylor series expansion of $H(s)$ about the point s_0 is stated using matrix $\mathcal{M}$ and vector(s) $\mathcal{R}$ (1.5) synthesized by recognizing the following algebraic trick

$$s\mathcal{E} - \mathcal{A} = \Big((s_0\mathcal{E} - \mathcal{A}) + ((s - s_0)\mathcal{E})\Big) = (s_0\mathcal{E} - \mathcal{A})\Big(I + (s - s_0)(s_0\mathcal{E} - \mathcal{A})^{-1}\mathcal{E}\Big).$$

Algebraic manipulation of the matrix pencil $s\mathcal{E} - \mathcal{A}$ translates into an equivalent first-order linear dynamical system as follows:

$$\mathcal{M}\frac{d}{dt}\mathbf{x}(t) = (s_0\mathcal{M} - I)\mathbf{x}(t) + \mathcal{R}\mathbf{u}(t),$$
$$\mathbf{y}(t) = \mathcal{L}\mathbf{x}(t) + \mathcal{D}\mathbf{u}(t).$$

By comparing coefficient matrices, the projections for the reduced-order system (2.31) are calculated efficiently using data available from the (band) Arnoldi algorithm, where

$$\mathcal{A}_n = V_n^T\mathcal{A}V_n = s_0H_n - I_n, \qquad \mathcal{E}_n = V_n^T\mathcal{E}V_n = H_n, \qquad \mathcal{B}_n = \rho_n,$$

and $\rho_n = V_n^T\mathcal{R}$. If $\mathcal{L} \neq \mathcal{B}^H$, then $\mathcal{L}V_n$ must be calculated explicitly. This (band) Arnoldi-based reduction technique produces Padé-type models for any higher-order linear dynamical system in state-space form.

Algorithm 5 (Band) Arnoldi-Based Dimension Reduction for Higher-Order Systems

Input: An ℓ-th order system (2.9) or special second-order system (2.12).
Output: A reduced first-order system (1.2)

1. Transform general higher-order system into equivalent first-order system (2.19) by defining proper linearization matrices $\mathcal{A}, \mathcal{E}, \mathcal{B}$, and $\mathcal{L}$ with state-space dimension N.
2. Execute (band) Arnoldi-based dimension reduction for the equivalent first-order system using Algorithm 4

Transforming a higher-order linear dynamical system into an equivalent first-order formulation and then applying reduction techniques designed to process first-order data is a popular strategy to approach large-scale higher-order problems [**OCP97**]. However, a legitimate criticism of this procedure is that the data matrices associated with the transformed system have state-space dimension N, which is significantly larger than the matrices determining the original higher-order equations. In addition, brute force projection using V_n ignores inherent structure of linearization matrices. As will be shown in this thesis, customizing Krylov subspace-based projection mechanisms for higher-order linear dynamical systems yields reduced-order models that preserve higher-order structure.

Initial attempts to create Krylov subspace-based reduction techniques that exploit structure inherent in the linearization matrices $\mathcal{A}, \mathcal{E}, \mathcal{B}$ and $\mathcal{L}$ exist in the literature [**SC91**], [**Fre05**], [**BS05a**], [**Mee08**], [**Fre11**]. The SOAR-based dimension reduction [**BS05a**] is designed for applications to second-order systems (2.9).

The resulting reduced-order system (2.33) is a Padé-type model of the original system (2.32). However, the SOAR algorithm does not provide a mechanism to deal with multiple-input, multiple-output higher-order systems. While SOAR suggests a viable reduction technique for single-input second-order systems, it does not provide a mechanism to deal with multiple-input system efficiently nor does it enable a more general reduction technique for any ℓth-order system (2.9) with $\ell \geq 2$.

Algorithm 6 SOAR-Based Dimension Reduction for Second-Order Systems

Input: A single-input, single-output second-order system

$$P_2\frac{d^2}{dt^2}\mathbf{z}(t) + P_1\frac{d}{dt}\mathbf{z}(t) + P_0\mathbf{z}(t) = B\mathbf{u}(t), \qquad y(t) = L_0\mathbf{z}(t). \tag{2.32}$$

Output: An equivalent reduced second-order system

$$\widetilde{P}_2\frac{d^2}{dt^2}\mathbf{z}(t) + \widetilde{P}_1\frac{d}{dt}\mathbf{z}(t) + \widetilde{P}_0\mathbf{z}(t) = \widetilde{B}\mathbf{u}(t), \qquad y(t) = \widetilde{L}_0\mathbf{z}(t). \tag{2.33}$$

1. Select expansion point s_0 such that $\widehat{P} = P(s_0) = s_0^2P_2 + s_0P_1 + P_0$ is nonsingular and choose reduction dimension n.
2. Run $n-1$ steps of the SOAR algorithm with
$$M^{(1)} = -\widehat{P}^{-1}(2s_0P_2 + P_1), \qquad M^{(2)} = -\widehat{P}^{-1}(P_2), \qquad \mathbf{r} = \widehat{P}^{-1}B,$$
to produce matrix $\widehat{Q}_n$ whose column spans S_n.
3. Obtain the reduced order model for the original system by projection:
$$\widetilde{P}_i = Q_n^TP_iQ_n, \qquad \widetilde{B} = Q_n^TBQ_n, \qquad \widetilde{L}_0 = Q_n^TL_0Q_n,$$
for $i \in \{0, 1, 2\}$.

For special first-order integro-DAEs (2.12), the SPRIM algorithm [**Fre04**] is an inventive reduced-order modeling technique designed to preserve structure of the original state-space description. As mentioned above, any system of integro-DAEs can be transformed into an equivalent first-order time-invariant linear dynamical system. The transformed system can be reduced using the (band) Arnoldi algorithm as indicated in Algorithm 5. SPRIM takes this reduction strategy one step further. By initially executing a classic Krylov subspace-based moment-matching reduction technique to get a basis matrix for $K(\mathcal{M}, \mathcal{R}, n)$, the SPRIM algorithm post-processes the resulting basis matrix V_n and executes the necessary projections in a way that maintains the structure of the linearization matrices.

The extra processing in the SPRIM algorithm guarantees that the resulting reduced system is a Padé-type model and ensures that system properties are maintained by the reduced-order model. If the equivalent first-order system exhibits J-Hermetian structure [**Fre08**], the SPRIM-inspired reduced-order models match twice as many moments as the corresponding brute-force Arnoldi-based reduction.

Algorithm 7 SPRIM-Based Dimension Reduction for First-Order Integro-DAEs

Input: Linearization matrices arising from special first-order integro-DAEs in the form

$$\mathcal{E} = \begin{bmatrix} E_{11} & 0 \\ 0 & E_{22} \end{bmatrix}, \qquad \mathcal{A} = -\begin{bmatrix} A_{11} & A_{12} \\ -A_{12}^T & 0 \end{bmatrix}, \qquad \mathcal{B} = \begin{bmatrix} B \\ 0 \end{bmatrix}, \qquad \mathcal{L} = \begin{bmatrix} L_0 & 0 \end{bmatrix}, \tag{2.34}$$

with $\mathcal{E}_{11} \succeq 0$, $\mathcal{E}_{22} \succeq 0$ and $\mathcal{A}_{11} \succeq 0$.

Output: An equivalent reduced-order system

$$\widetilde{\mathcal{E}}_n = \begin{bmatrix} \widetilde{E}_{11} & 0 \\ 0 & \widetilde{E}_{22} \end{bmatrix}, \qquad \widetilde{\mathcal{A}}_n = -\begin{bmatrix} \widetilde{A}_{11} & \widetilde{A}_{12} \\ -\widetilde{A}_{12}^T & 0 \end{bmatrix}, \qquad \widetilde{\mathcal{B}}_n = \begin{bmatrix} \widetilde{B} \\ 0 \end{bmatrix}, \qquad \widetilde{\mathcal{L}}_n = \begin{bmatrix} \widetilde{L}_0 & 0 \end{bmatrix}, \tag{2.35}$$

with $\widetilde{\mathcal{E}}_{11} \succeq 0$, $\widetilde{\mathcal{E}}_{22} \succeq 0$ and $\widetilde{\mathcal{A}}_{11} \succeq 0$.

1. Select an expansion point s_0 such that $\hat{Q} = Q(s_0) = s_0 P_1 + P_0 + \frac{1}{s_0} P_{-1}$ is nonsingular and choose reduction dimension n.
2. Run n steps of any block Krylov subspace method applied to $\mathcal{M}$ and $\mathcal{R}$ to generate matrix V_n such that the column span $(V_n) = K(\mathcal{M}, \mathcal{R}, n)$
3. Partition the columns of the matrix $V_n = \begin{bmatrix} \widetilde{V}_1 \\ \widetilde{V}_2 \end{bmatrix}$ with block sizes corresponding to those of $\mathcal{A}$ and $\mathcal{E}$.
4. Post-process the matrix $\widetilde{V}_i$ to produce a matrix $\widehat{V}_i$ whose columns are orthonormal and span the range of $\widetilde{V}_i$ for $i = 1, 2$.
5. Set

$$\begin{aligned} \widetilde{E}_{11} &= \widehat{V}_1^T E_{11} \widehat{V}_1, & \widetilde{E}_{22} &= \widehat{V}_2^T E_{22} \widehat{V}_2, \\ \widetilde{A}_{11} &= \widehat{V}_1^T P_1 \widehat{V}_1, & \widetilde{A}_{12} &= \widehat{V}_1^T A_{12} \widehat{V}_2, \\ \widetilde{B} &= \widehat{V}_1^T B, & \widetilde{L}_0 &= L_0 \widehat{V}_1 \end{aligned}$$

using matrix-matrix multiplication to compute each projection.

Opportunities exist to improve Krylov-subspace based model order reduction techniques for both general ℓth-order systems and special second-order systems. Chapter 3 generalizes the SOAR algorithm for any ℓth-order linear dynamical systems (2.9) and includes a mechanism to reduce multiple-input, multiple-output systems. Chapter 4 develops an algorithm that reduces special second-order systems by producing the matrices $\widehat{V}_1$ and $\widehat{V}_2$ used in the SPRIM algorithm without the having to post-process any data. The results presented in Chapter 4 represent marked improvements in computational efficiency for model order reduction of first-order integro-DAEs.

CHAPTER 3

The SEA Algorithm for Matrices in Case One Form

The Structure Exploiting Arnoldi (SEA) algorithm presented in this chapter is a novel reduced-order modeling technique for ℓth-order time-invariant linear dynamical systems for any $\ell \geq 2$. As discussed in Section 2.4, a popular approach to model reduction of a given higher-order system is to linearize the system and apply classic Krylov subspace-based moment-matching reduction techniques to the equivalent first-order formulation. Not only does this strategy result in an ℓ-fold increase in the size of the data being processed, it also ignores structure inherent in the linearization matrices. Moreover, the resulting reduced-order model can only be stated as a first-order system, a distinct disadvantage if key system properties are encoded in the higher-order system equations.

Initial attempts to address the weaknesses of the linearization approach to dimension reduction of higher-order systems exploit the structure of linearization matrices for second-order systems ($\ell = 2$) [**SC91**], [**BS05a**]. These techniques do not address the general ℓth-order case ($\ell \geq 2$) nor are these procedures based on the structure of the underlying Krylov subspaces [**Fre05**]. Finally, none of these routines include accurate deflation mechanisms for block Krylov subspaces resulting from a block of starting vectors.

Theoretic and numeric results included in this chapter demonstrate an improved model order reduction technique for ℓth-order systems with multiple input vectors. Section 3.1 extends prior structure results for block Krylov subspaces associated with matrices in case one form, a special class of matrices related to linearized ℓth-order linear dynamical systems [**Fre05**]. Unique to this work is an improved method of executing exact deflation without explicitly manipulating columns of the block Krylov matrix. Section 3.2 describes a generic single-input SEA algorithm for matrices in case one form and establishes the key theoretic properties that any structure exploiting algorithm of this flavor should exhibit. Section 3.3 adapts the generic SEA algorithm for a special subclass of matrices in case one form arising from linearized ℓth-order linear dynamical systems including a computationally efficient stop condition. Section 3.4 presents band versions of the SEA algorithms

for multiple input vectors. Section 3.5 gives the SEA-based model reduction technique for multiple-input, ℓth-order systems with $\ell \geq 2$ and presents numerical results demonstrating the effectiveness of this method. The last section of this chapter concludes with a summary of the advantages of SEA-based model order reduction for ℓth-order linear dynamical systems.

3.1 Exploring Matrices in Case One Form

The first completely general analysis of the structure of block Krylov subspaces associated with linearized higher-order systems defines a generic class of structured matrices that can be used to describe matrices $\mathcal{M}$ and $\mathcal{R}$ corresponding to transfer function $H(s)$ (1.4) arising from the linearization of ℓth-order systems [**Fre05**]. This initial study provides a factorization of any basis matrix for the block Krylov subspaces induced by these types of matrices, so called matrices in case one form. These factorization result can be refined from the stand point of generating an orthonormal basis for the column space of the multiple copied matrix factor. Included in this section are a recurrence relation providing insight into the basis generation problem for the multiple copied subspace, an alternative proof to the original structure result and a full-rank factorization result suggesting an improved deflation mechanism for basis matrices of associated block Krylov subspaces.

Definition 3.1.1 (Freund [**Fre05**])**.** Let n_0 and ℓ be natural numbers. Let $N = n_0 \cdot \ell$. Let $\mathbf{c} = \begin{bmatrix} c_1 & c_2 & \cdots & c_\ell \end{bmatrix}^T \in \mathbb{C}^\ell$ be a vector such that $c_i \neq 0$ for all $i = 1, 2, ..., \ell$. Suppose that

$$F = \begin{bmatrix} M^{(1)} & M^{(2)} & \cdots & M^{(\ell)} \end{bmatrix} \in \mathbb{C}^{n_0 \times N} \qquad \text{and} \qquad M^{(i)} \in \mathbb{C}^{n_0 \times n_0} \text{ for } i = 1, ..., \ell. \tag{3.1}$$

Let $S \in \mathbb{C}^{\ell \times \ell}$ and assume $R \in \mathbb{C}^{n_0 \times m}$. Matrices $\mathcal{M}$ and $\mathcal{R}$ are said to be in **case one form** if, and only if,

$$\mathcal{M} = \left(\mathbf{c} \otimes F + S \otimes I_{n_0}\right) \in \mathbb{C}^{N \times N} \qquad \text{and} \qquad \mathcal{R} = \mathbf{c} \otimes R \in \mathbb{C}^{N \times m}. \tag{3.2}$$

One class of matrices in case one form arise from the Taylor series expansion of the transfer function $H(s)$ (1.4) associated with linearized ℓth-order linear dynamical systems.

Proposition 3.1.2 (Freund [**Fre05**])**.** Transform a given ℓth-order linear dynamical system (2.9) into a first-order equivalent system using the appropriate linearization matrices (2.21), (2.23).

Suppose that the matrix pencil $P(s)$ (2.11) is regular and choose nonzero $s_0 \in \mathbb{C}$ such that $\big(P(s_0)\big)^{-1}$ exists. For $i = 1, 2, ..., \ell$, let

$$M^{(i)} = (P(s_0))^{-1} \left(\sum_{j=0}^{\ell-i} s_0^j P_{i+j} \right), \qquad \text{and} \qquad R = (P(s_0))^{-1} B. \tag{3.3}$$

Then, matrices

$$\begin{aligned} \mathcal{M} &= (s_0\mathcal{E} - \mathcal{A})^{-1}\mathcal{E} = \mathbf{c} \otimes F + S \otimes I_{n_0}, \\ \mathcal{R} &= (s_0\mathcal{E} - \mathcal{A})^{-1}\beta = \mathbf{c} \otimes R \end{aligned} \tag{3.4}$$

for special choices

$$\mathbf{c} = \begin{bmatrix} 1 \\ s_0 \\ s_0^2 \\ \vdots \\ s_0^{\ell-1} \end{bmatrix} \qquad \text{and} \qquad S = -\begin{bmatrix} 0 & 0 & \cdots & \cdots & 0 \\ 1 & 0 & \ddots & & \vdots \\ s_0 & 1 & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ s_0^{\ell-2} & \cdots & s_0 & 1 & 0 \end{bmatrix},$$

Block Krylov subspaces induced by matrices in case one form have inherent structure. Assume matrices $\mathcal{M} \in \mathbb{C}^{N\times N}$ and $\mathcal{R} \in \mathbb{C}^{N\times m}$ are in case one form for the rest of Chapter 3.

Proposition 3.1.3 (Freund [**Fre05**])**.** Let $\mathcal{V}$ be any basis matrix of the block-Krylov subspaces induced by $\mathcal{M}$ and $\mathcal{R}$. Then,

$$\mathcal{V} = \begin{bmatrix} WU^{(1)} \\ WU^{(2)} \\ \vdots \\ WU^{(\ell)} \end{bmatrix} \tag{3.5}$$

where $W \in \mathbb{C}^{n_0\times N_0}$ and $U^{(i)} \in \mathbb{C}^{N_0\times N_0}$ is nonsingular and upper-triangular for $i = 1, 2, ...\ell$.

Let $\mathbf{w}_i \in \mathbb{C}^{n_0}$ be the ith column of the matrix $W \in \mathbb{C}^{n_0\times N_0}$, for $i = 1, 2, ..., N_0$. Let

$$S_n = \langle \mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_n \rangle \subseteq \mathbb{C}^{n_0}, \tag{3.6}$$

for $n = 1, 2, ..., N_0$. This space will of central importance throughout the rest of Chapter 3.

The original proof of the case one structure result (3.5) factorizes the deflated block Krylov matrix induced by $\mathcal{M}$ and $\mathcal{R}$. Using a recurrence relation for the block columns of W coupled with block-columns of $U^{(i)}$, the proof illustrates

$$\begin{bmatrix} \mathcal{R}_1 & \mathcal{M}\mathcal{R}_2 & \cdots & \mathcal{M}^{k_0-1}\mathcal{R}_{k_0} \end{bmatrix} = \begin{bmatrix} WU^{(1)} \\ WU^{(2)} \\ \vdots \\ WU^{(\ell)} \end{bmatrix}.$$

However, block partitions of W can be constructed without specific reference to matrices $U^{(i)}$, for $i = 1, 2, ..., \ell$.

Proposition 3.1.4. Let $m_0 = m$ and suppose that $E_j \in \mathbb{C}^{m_{j-1} \times m_j}$ is the appropriate elimination matrix from the exact deflation process for $j = 1, 2, ..., k_0$. Initialize the coupled recursion

$$W_1 = RE_1, \qquad A_1 = \mathbf{c} \otimes W_1. \tag{3.7}$$

Define the $(k+1)$st coupled recursion pair as

$$W_{k+1} = FA_kE_{k+1}, \qquad A_{k+1} = \mathbf{c} \otimes W_{k+1} + (S \otimes I_{n_0})A_kE_{k+1}, \tag{3.8}$$

for $k = 1, 2, ..., k_0 - 1$. Then, the kth block of the deflated block Krylov matrix induced by $\mathcal{M}$ and $\mathcal{R}$ 2.1.4 can be represented as

$$\mathcal{M}^{k-1}\mathcal{R}_k = A_k, \tag{3.9}$$

for $k = 1, 2, ..., k_0$

Proof: This claim is established by induction. Notice $A_1 = \mathbf{c} \otimes RE_1 = \mathcal{R}_1$. Assume $A_k = \mathcal{M}^{k-1}\mathcal{R}_k$. Now consider

$$A_{k+1} = \mathbf{c} \otimes W_{k+1} + (S \otimes I_{n_0})A_kE_{k+1} = \mathcal{M}^k\mathcal{R}_{k+1},$$

which is the desired result. □

Remark 3.1.5. This result demonstrates that any algorithm design to exploit the structure of block Krylov subspaces induced by matrices $\mathcal{M}$ and $\mathcal{R}$ in case one form (3.5) will need to produce an auxiliary basis matrix $\widehat{A}_n$ for $K(\mathcal{M}, \mathcal{R}, n)$. Updates for candidate basis vectors of S_n will also need to be generated by multiplying columns of this $\widehat{A}_n$ with the matrix F. This is a key theoretical insight of this chapter and most work that follows builds on this realization. Decoupling the columns of W and the columns of $U^{(i)}$ enables basis vectors of S_n to be studied without suggesting a one-to-one correspondence with columns of the corresponding deflated block Krylov matrix.

3.1.1 Alternative Proof of the Factorization Result

The recursive formulation of the deflated block Krylov matrix induced by $\mathcal{M}$ and $\mathcal{R}$ provides insight into an alternative proof for Proposition 3.1.3.

Proof: Partition possible candidate factors W and $U^{(i)}$ using the block sizes of the deflated block Krylov matrix induced by $\mathcal{M}$ and $\mathcal{R}$ where

$$W = \begin{bmatrix} W_1 & W_2 & \cdots & W_{k_0} \end{bmatrix}, \qquad U^{(i)} = \begin{bmatrix} U_{11}^{(i)} & U_{12}^{(i)} & \cdots & U_{1,k_0}^{(i)} \\ 0 & U_{22}^{(i)} & \ddots & U_{2,k_0}^{(i)} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & U_{k_0,k_0}^{(i)} \end{bmatrix},$$

$W_k \in \mathbb{C}^{n_0 \times m_k}$ and $U_{kk}^{(i)} \in \mathbb{C}^{m_k \times m_k}$ for all $i = 1, 2, ..., \ell$ and $k = 1, 2, ..., k_0$. Let W_k is defined as in Proposition 3.1.4 for $k = 1, ..., k_0$ and define the subblocks of $U^{(i)}$ as

$$\begin{bmatrix} U_{kk}^{(1)} \\ \vdots \\ U_{kk}^{(\ell)} \end{bmatrix} = \mathbf{c} \otimes I_{m_k}, \qquad \begin{bmatrix} U_{1k}^{(i)} \\ \vdots \\ U_{k-1,k}^{(i)} \end{bmatrix} = \sum_{t=1}^{\ell} s_{i,t} \begin{bmatrix} U_{1,k-1}^{(t)} \\ \vdots \\ U_{k-1,k-1}^{(t)} \end{bmatrix} E_k, \tag{3.10}$$

for $k = 1, 2, ..., k_0$, $j = 1, 2, ..., k-1$ and $i = 1, 2, ..., \ell$. The desired relation (3.5) holds if, and only if,

$$\mathcal{M}^{k-1}\mathcal{R}_k = \left(I_\ell \otimes \begin{bmatrix} W_1 & W_2 & \cdots & W_k \end{bmatrix}\right) \begin{bmatrix} \begin{bmatrix} U_{1k}^{(1)} \\ \vdots \\ U_{kk}^{(1)} \end{bmatrix} \\ \vdots \\ \begin{bmatrix} U_{1k}^{(\ell)} \\ \vdots \\ U_{kk}^{(\ell)} \end{bmatrix} \end{bmatrix} \tag{3.11}$$

Induction on k illustrates that W and $U^{(i)}$ factorize the deflated block Krylov matrix induced by matrices $\mathcal{M}$ and $\mathcal{R}$ in case one form (3.5). For $k = 1$,

$$\mathcal{R}_1 = \mathbf{c} \otimes RE_1 = \begin{bmatrix} c_1 R \\ \vdots \\ c_\ell R \end{bmatrix} E_1 = (I_\ell \otimes W_1) \begin{bmatrix} U_{11}^{(1)} \\ \vdots \\ U_{11}^{(\ell)} \end{bmatrix}.$$

Assume the equivalent structure relation (3.11) holds for some $k \in \{1, 2, ..., k_0 - 1\}$. Then $\mathcal{M}^k\mathcal{R}_{k+1}$ can be written as

$$\big(\mathbf{c} \otimes F\big)A_k E_{k+1} + \big(S \otimes I_{n_0}\big)\left(I_\ell \otimes \begin{bmatrix} W_1 & \cdots & W_k \end{bmatrix}\right) \begin{bmatrix} \begin{bmatrix} U_{1k}^{(1)} \\ \vdots \\ U_{kk}^{(1)} \end{bmatrix} \\ \vdots \\ \begin{bmatrix} U_{1k}^{(\ell)} \\ \vdots \\ U_{kk}^{(\ell)} \end{bmatrix} \end{bmatrix} E_{k+1} \tag{3.12}$$

Let $n = n(k) = m_1 + \cdots + m_k$ (2.4). By properties of the Kronecker product,

$$\big(S \otimes I_{n_0}\big)\left(I_\ell \otimes \begin{bmatrix} W_1 & \cdots & W_k \end{bmatrix}\right) = \left(I_\ell \otimes \begin{bmatrix} W_1 & \cdots & W_k \end{bmatrix}\right)\big(S \otimes I_n\big).$$

The matrix $\mathcal{M}^k\mathcal{R}_{k+1}$ can be rewritten

$$\mathcal{M}^k\mathcal{R}_{k+1} = \left(\mathbf{c} \otimes W_{k+1}\right) + I_\ell \otimes \begin{bmatrix} W_1 & \cdots & W_k \end{bmatrix} \begin{bmatrix} \sum_{t=1}^{\ell} s_{1,t} \begin{bmatrix} U_{1,k}^{(t)} \\ \vdots \\ U_{k,k}^{(t)} \end{bmatrix} E_{k+1} \\ \vdots \\ \sum_{t=1}^{\ell} s_{\ell,t} \begin{bmatrix} U_{1,k}^{(t)} \\ \vdots \\ U_{k,k}^{(t)} \end{bmatrix} E_{k+1} \end{bmatrix}.$$

By combining appropriate rules of matrix arithmetic with the definitions for $U_{j,k+1}^{(i)}$ for $j = 1, 2, ..., k+1$ and $i = 1, 2, ..., \ell$, the desired equation (3.11) is satisfied. □

3.1.2 Full-Rank Factorization

An immediate corollary to Proposition 3.1.3 replaces the column rank deficient matrix W with a factor Q having full column rank.

Corollary 3.1.6. Let $\mathcal{V}$ be any basis matrix of the block-Krylov subspaces induced by $\mathcal{M}$ and $\mathcal{R}$. Then, $\mathcal{V}$ can be represented in the form

$$\mathcal{V} = (I_\ell \otimes Q) \begin{bmatrix} X^{(1)} \\ X^{(2)} \\ \vdots \\ X^{(\ell)} \end{bmatrix} \tag{3.13}$$

where $Q \in \mathbb{C}^{n_0 \times w_0}$ has full column rank. Each $X^{(i)} \in \mathbb{C}^{w_0 \times N_0}$ is in row echelon form and the sparsity structure of $X^{(i)}$ is identical to that of $X^{(j)}$ for $i, j = 1, 2, ..., \ell$.

Proof: By Proposition 3.1.3, the basis matrix $\mathcal{V}$ can be factored using matrices W and $U^{(i)}$ for $i = 1, 2, ..., \ell$. If W has column rank $w_0 \leq n_0$, then $W = Q\widehat{R}$ where $Q \in \mathbb{C}^{n_0 \times w_0}$ has orthonormal columns and $\widehat{R} \in \mathbb{C}^{w_0 \times N_0}$ stores the relationships between the column vectors of Q and all columns of W. Setting $X^{(i)} = \widehat{R}U^{(i)}$ for $i = 1, 2, ..., \ell$ completes this proof. □

Remark 3.1.7. The focus of this chapter is to accurately generate the columns of the matrix Q one at a time. This matrix will ultimately be used to produce projection-based Padé-type reduced-order models that can be stated as higher-order systems. A modified Gram-Schmidt style algorithm will be adapted for this purpose.

A more detailed discussion of the linearly independent columns of W is in order. Denote the column rank of W as $w_0 \leq n_0$. Partition the set of column indices $\{1, 2, ..., N_0\}$ of the matrix W into two disjoint sets $\beta = \{b_1, ..., b_{w_0}\}$ and $\delta = \{d_1, ..., d_{N_0 - w_0}\}$. Starting from the leftmost column of W and moving to the right, choose b_j to be the column index of the jth linearly independent column of the matrix W, for $j = 1, ..., w_0$. Proceeding in the same order, choose d_j to be the column index of the jth linearly dependent column of W, for $j = 1, ..., N_0 - w_0$. Here, $b_1 < b_2 < \cdots < b_{w_0}$ while $d_1 < d_2 < \cdots < d_{N_0 - w_0}$.

Let the sequence of nonnegative integers $w_1, w_2, ..., w_{k_0}$ indicate the number of new indices contributed to the set β by the subblocks $W_1, W_2, ..., W_{k_0}$, respectively. In other words, w_j is the number of new basis vectors for the columns space of W contributed by W_j. As a consequence, there are $m_j - w_j$ new indices added to the set δ for each block W_j. Note that $0 \leq w_j \leq m_j$ and $w_0 = w_1 + w_2 + \cdots + w_{k_0}$.

These dimensions permit a block column partition of $Q \in \mathbb{C}^{n_0 \times w_0}$ and $X^{(i)} \in \mathbb{C}^{w_0 \times N_0}$ from Corollary 3.1.6 as

$$Q = \begin{bmatrix} Q_1 & Q_2 & \cdots & Q_{k_0} \end{bmatrix}, \qquad X^{(i)} = \begin{bmatrix} X_{11}^{(i)} & X_{12}^{(i)} & \cdots & X_{1k_0}^{(i)} \\ 0 & X_{22}^{(i)} & & X_{2k_0}^{(i)} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & X_{k_0k_0}^{(i)} \end{bmatrix} \tag{3.14}$$

with $Q_j \in \mathbb{C}^{n_0 \times w_j}$, for $j = 1, 2, ..., k_0$ and $X_{jk}^{(i)} \in \mathbb{C}^{w_j \times m_k}$ for $i = 1, 2, ..., \ell$ and $j, k = 1, 2, ..., k_0$. This full-rank factorization provides a powerful method to execute exact deflation on the columns of $K_N(\mathcal{M}, \mathcal{R})$ without explicitly manipulating vectors of size N.

Example 3.1.8. This example that illustrates the difference between the original factorization offered in Proposition 3.1.3 and the streamlined factorization from Corollary 3.1.6. Let $\mathcal{M}$ and $\mathcal{R}$

be defined by

$$M^{(1)} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \qquad M^{(2)} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \qquad R = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix},$$

and set

$$\mathbf{c} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \qquad S = \begin{bmatrix} 0 & 0 \\ 2 & 0 \end{bmatrix},$$

with $n_0 = 5$, $m = 1$, and $\ell = 2$. The first five columns of the factorization suggested by Proposition 3.1.3 are given by

$$\begin{bmatrix}\mathcal{R} & \mathcal{M}\mathcal{R} & \mathcal{M}^2\mathcal{R} & \mathcal{M}^3\mathcal{R} & \mathcal{M}^4\mathcal{R}\end{bmatrix} = \left(I_2 \times \begin{bmatrix} 1 & 1 & 2 & 8 & 24 \\ 0 & 1 & 4 & 8 & 20 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \right) \left[\begin{array}{ccccc} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ \hline 1 & 2 & 0 & 0 & 0 \\ 0 & 1 & 2 & 0 & 0 \\ 0 & 0 & 1 & 2 & 0 \\ 0 & 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 0 & 1 \end{array}\right].$$

The corresponding factorization suggested in Corollary 3.1.6 is

$$\begin{bmatrix}\mathcal{R} & \mathcal{M}\mathcal{R} & \mathcal{M}^2\mathcal{R} & \mathcal{M}^3\mathcal{R} & \mathcal{M}^4\mathcal{R}\end{bmatrix} = \left(I_2 \otimes \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \right) \left[\begin{array}{ccccc} 1 & 1 & 2 & 8 & 24 \\ 0 & 1 & 4 & 8 & 20 \\ \hline 1 & 3 & 4 & 12 & 40 \\ 0 & 1 & 6 & 16 & 36 \end{array}\right].$$

The Krylov matrix $K_{10}(\mathcal{M}, \mathcal{R})$ has rank four while the matrix factor W has rank two. As will be demonstrated below, the rank of $K_{10}(\mathcal{M}, \mathcal{R})$ can be determined by working with the matrix

$$\left[\begin{array}{ccccc} 1 & 1 & 2 & 8 & 24 \\ 0 & 1 & 4 & 8 & 20 \\ \hline 1 & 3 & 4 & 12 & 40 \\ 0 & 1 & 6 & 16 & 36 \end{array}\right].$$

Remark 3.1.9. Corollary 3.1.6 suggests a method by which the exact deflation of candidate basis vectors for the block Krylov matrix induced by $\mathcal{M}$ and $\mathcal{R}$ can be detected using only information encoded in matrices $X^{(i)}$, for $i = 1, 2, ..., \ell$. Because the basis vectors for the block Krylov subspace induced by $\mathcal{M}$ and $\mathcal{R}$ are useful only as auxiliary vectors for producing columns of Q, it is desirable

to avoid manipulating vectors of size N while simultaneously ensuring proper deflation decisions for the basis matrix of $K(\mathcal{M}, \mathcal{R}, N_0)$. Generalizing Proposition 3.1.2 and Corollary 3.1.6 enables just this type of deflation decisions to be made using matrix Q.

3.1.3 Exact Deflation and the Column Space of W

All of the factorization results above assume prior knowledge of both the block grade k_0 of the block Krylov subspace induced by $\mathcal{M}$ and $\mathcal{R}$ as well as the elimination matrices $\{E_j\}_{j=1}^{k_0}$ arising from exact deflation on $K_N(\mathcal{M}, \mathcal{R})$. These theoretic assumptions are not realistic when the strategy is to avoid manipulating columns of $K_N(\mathcal{M}, \mathcal{R})$ directly. An analogous factorization to Proposition 3.1.3 that requires no knowledge of the exact deflation patterns for $K_N(\mathcal{M}, \mathcal{R})$ is given by

$$K_N(\mathcal{M}, \mathcal{R}) = \begin{bmatrix} \widehat{W}\widehat{U}^{(1)} \\ \widehat{W}\widehat{U}^{(2)} \\ \vdots \\ \widehat{W}\widehat{U}^{(\ell)} \end{bmatrix}, \tag{3.15}$$

where $\widehat{W} \in \mathbb{C}^{n_0 \times m \cdot N}$ and $\widehat{U}^{(i)} \in \mathbb{C}^{m \cdot N \times m \cdot N}$ is nonsingular, upper-triangular, and Toeplitz, for $i = 1, 2, ..., \ell$. In this case, the subblock partitions of $\widehat{W}$ are given by

$$\widehat{W}_1 = R, \qquad \widehat{A}_1 = \mathbf{c} \otimes \widehat{W}_1,$$

and

$$\widehat{W}_{k+1} = F\widehat{A}_k, \qquad \widehat{A}_{k+1} = \mathbf{c} \otimes \widehat{W}_{k+1} + (S \otimes I_{n_0})\widehat{A}_k,$$

for $k = 1, 2, ..., N-1$. Unlike the coupled recursion offered in Proposition 3.1.4, this set of recursions does not reference the elimination matrices $\{E_j\}_{j=1}^{k_0}$. Similarly, matrices $\widehat{U}^{(i)}$ can be defined using the same recursive definition in the alternate proof to Proposition 3.1.3 sans reference to the elimination matrices. This factorization suggests a connection between exact deflation of columns of $K_N(\mathcal{M}, \mathcal{R})$ and linear dependence of columns of $\widehat{W}$.

Proposition 3.1.10. Denote the nth column of $K_N(\mathcal{M},\mathcal{R})$ as $\hat{\mathbf{a}}_n \in \mathbb{C}^N$ and the corresponding column of $\widehat{W}$ as $\hat{\mathbf{w}}_n$ for $n = 1, 2, ..., mN$. If $\hat{\mathbf{a}}_{n+1} \in \langle \hat{\mathbf{a}}_1, \hat{\mathbf{a}}_2, ..., \hat{\mathbf{a}}_n \rangle$ then $\hat{\mathbf{w}}_{n+1} \in \langle \hat{\mathbf{w}}_1, \hat{\mathbf{w}}_2, ..., \hat{\mathbf{w}}_n \rangle$ for $n = 1, 2, ..., mN - 1$.

Proof: If $\hat{\mathbf{a}}_{n+1} \in \langle \hat{\mathbf{a}}_1, \hat{\mathbf{a}}_2, ..., \hat{\mathbf{a}}_n \rangle$ for some $n \in \{1, 2, ..., mN-1\}$, there is a nonzero vector $\mathbf{x} \in \mathbb{C}^{n+1}$ such that

$$\left[\begin{array}{cccc} \hat{\mathbf{a}}_1 & \hat{\mathbf{a}}_2 & \cdots & \hat{\mathbf{a}}_n & \hat{\mathbf{a}}_{n+1} \end{array} \right] \mathbf{x} = \mathbf{0} \in \mathbb{C}^N \tag{3.16}$$

and the last coefficient of $\mathbf{x}$ is nonzero. Let $\mathbf{e}_1$ be the first row of I_ℓ. Multiply both sides of this equation (3.16) by $\mathbf{e}_1 \otimes I_{n_0}$ yielding $\mathbf{0} = \left[\begin{array}{cccc} \hat{\mathbf{w}}_1 & \hat{\mathbf{w}}_2 & \cdots & \hat{\mathbf{w}}_n & \hat{\mathbf{w}}_{n+1} \end{array} \right] \mathbf{y} \in \mathbb{C}^{n_0}$. In this case, $\mathbf{y} = \mathcal{U}\,\mathbf{x}$ with $\mathcal{U} = \widehat{U}^{(1)}(1 : n+1, 1 : n+1) \in \mathbb{C}^{(n+1)\times(n+1)}$. Because $\mathcal{U}$ is upper-triangular with nonzero diagonal coefficients and $\mathbf{x} \neq \mathbf{0}$, the entries of $\mathbf{y}$ must be nonzero. Specifically, the last coefficient of $\mathbf{y}$ is nonzero. Thus, $\hat{\mathbf{w}}_{n+1}$ can be written as a linear combination of vectors $\{\hat{\mathbf{w}}_j\}_{j=1}^n$. □

Exact deflation of columns in $K_N(\mathcal{M},\mathcal{R})$ implies linear dependence of the corresponding columns of the candidates of $\widehat{W}$. By contraposition, if a column of $\widehat{W}$ is linearly independent from previous columns, then the corresponding column of $K_N(\mathcal{M},\mathcal{R})$ will necessarily be linearly independent from previous basis vectors for the deflated block Krylov matrix. The converse of Proposition 3.1.10 is not true. Even if $\hat{\mathbf{w}}_{n+1}$ is in the column span of

$$\left[\begin{array}{cccc} \hat{\mathbf{w}}_1 & \hat{\mathbf{w}}_2 & \cdots & \hat{\mathbf{w}}_n \end{array} \right],$$

the corresponding vector $\hat{\mathbf{a}}_{n+1}$ may not require exact deflation, as Example 3.1.8 illustrates. However, necessary and sufficient conditions for exact deflation of columns of the block Krylov matrix $K_N(\mathcal{M},\mathcal{R})$ based on the linear dependence classification of $\hat{\mathbf{w}}_n$ are available by further exploring Corollary 3.1.6.

To begin, consider the first block $\mathcal{R} = \mathbf{c} \otimes R$ of $K_N(\mathcal{M},\mathcal{R})$. The number of linearly dependent columns of $\mathcal{R}$ is equal to the number of linearly independent columns of R since $\mathbf{r}_{n+1} \in \langle \mathbf{r}_1, ..., \mathbf{r}_n \rangle$ if, and only if, $\mathbf{c} \otimes \mathbf{r}_{n+1} \in \langle \mathbf{c} \otimes \mathbf{r}_1, ..., \mathbf{c} \otimes \mathbf{r}_n \rangle$ for $n = 1, 2, ..., m-1$. The full-rank factorization of $\mathcal{R}_1$ from the deflated Krylov matrix given in Corollary 3.1.6 is readily available. Assume $Q_1 \in \mathbb{C}^{n_0 \times m_1}$ is an orthonormal basis for the column space of R and let $RE_1 = Q_1 \widehat{R}_1$ with $\widehat{R}_1$ upper-triangular

and nonsingular. Define $X_{11}^{(i)} \in \mathbb{C}^{m_1 \times m_1} = c_i \widehat{R}_1$ for $i = 1, 2, ..., \ell$. Then

$$\mathcal{R}_1 = \mathcal{R}E_1 = \begin{bmatrix} Q_1 X_{11}^{(1)} \\ \vdots \\ Q_1 X_{11}^{(\ell)} \end{bmatrix}$$

The elimination matrix $E_1 \in \mathbb{C}^{m \times m_1}$ for $\mathcal{R}$ can be determined by working with $R \in \mathbb{C}^{n_0 \times m}$. This is only true for the first block of $K_N(\mathcal{M}, \mathcal{R})$.

Assume that the elimination matrices $\{E_j\}_{j=1}^{k}$ are known for some $k = 1, 2, ..., k_0 - 1$ and suppose the first k blocks of the deflated block Krylov matrix are given by

$$\left[\begin{array}{c|c|c|c} \mathcal{R}_1 & \mathcal{M}\mathcal{R}_2 & \cdots & \mathcal{M}^{k-1}\mathcal{R}_k \end{array} \right] = \left(I_\ell \otimes \begin{bmatrix} Q_1 & Q_2 & \cdots & Q_k \end{bmatrix} \right) \begin{bmatrix} \begin{bmatrix} X_{11}^{(1)} & \cdots & X_{1k}^{(1)} \\ & \ddots & \vdots \\ & & X_{kk}^{(1)} \end{bmatrix} \\ \vdots \\ \begin{bmatrix} X_{11}^{(\ell)} & \cdots & X_{1k}^{(\ell)} \\ & \ddots & \vdots \\ & & X_{kk}^{(\ell)} \end{bmatrix} \end{bmatrix}$$

where $\mathcal{R}_j = \mathcal{R}_{j-1}E_j$, $Q_j \in \mathbb{C}^{n_0 \times w_j}$ and $X_{jt}^{(i)} \in \mathbb{C}^{w_j \times m_t}$ for $j, t = 1, 2, ..., k$ (3.14). The next block $\mathcal{M}^k\mathcal{R}_{k+1}$ is created first by forming the product $\mathcal{M}(\mathcal{M}^{k-1}\mathcal{R}_k)$ and then identifying the elimination matrix E_{k+1}. Suppose that $Q_{k+1} \in \mathbb{C}^{n_0 \times w_{k+1}}$ is next block of orthonormal columns for the column space of W corresponding to $\widehat{W}_{k+1}$. Define a set of candidate coefficients

$$\begin{bmatrix} \begin{bmatrix} \widehat{X}^{(1)}_{1,k+1} \\ \vdots \\ \widehat{X}^{(1)}_{k,k+1} \\ \widehat{X}^{(1)}_{k+1,k+1} \end{bmatrix} \\ \vdots \\ \begin{bmatrix} \widehat{X}^{(\ell)}_{1,k+1} \\ \vdots \\ \widehat{X}^{(\ell)}_{k,k+1} \\ \widehat{X}^{(\ell)}_{k+1,k+1} \end{bmatrix} \end{bmatrix} = \left(I_\ell \otimes \begin{bmatrix} Q_1^H \\ \vdots \\ Q_k^H \\ Q_{k+1}^H \end{bmatrix} \right) \mathcal{M}^k \mathcal{R}_k$$

A deflation condition in later blocks can now be given succinctly in terms of the matrix

$$X = \begin{bmatrix} \begin{bmatrix} X^{(1)}_{11} & \cdots & X^{(1)}_{1k} & \widehat{X}^{(1)}_{1,k+1} \\ & \ddots & \vdots & \vdots \\ & & X^{(1)}_{kk} & \widehat{X}^{(1)}_{k,k+1} \\ & & & \widehat{X}^{(1)}_{k+1,k+1} \end{bmatrix} \\ \vdots \\ \begin{bmatrix} X^{(\ell)}_{11} & \cdots & X^{(\ell)}_{1k} & \widehat{X}^{(\ell)}_{1,k+1} \\ & \ddots & \vdots & \vdots \\ & & X^{(\ell)}_{kk} & \widehat{X}^{(\ell)}_{k,k+1} \\ & & & \widehat{X}^{(\ell)}_{k+1,k+1} \end{bmatrix} \end{bmatrix}$$

Since $\begin{bmatrix} Q_1 & \cdots & Q_k & Q_{k+1} \end{bmatrix}$ has full column rank and

$$\begin{bmatrix} \mathcal{R}_1 & \mathcal{M}\mathcal{R}_2 & \cdots & \mathcal{M}^{k-1}\mathcal{R}_k & \mathcal{M}^k\mathcal{R}_k \end{bmatrix} = I_\ell \otimes \begin{bmatrix} Q_1 & \cdots & Q_k & Q_{k+1} \end{bmatrix} X,$$

exact deflation occurs on the columns of $\mathcal{M}^k\mathcal{R}_k$ if an only if the matrix X is column rank deficient. As discussed in Section 3.3, the columns of X can be formed without use of an explicit inner product and are much smaller than the columns of the block Krylov matrix $K_N(\mathcal{M}, \mathcal{R})$ in applications related to reduced-order modeling.

3.2 The Generic SEA Algorithm for Matrices in Case One Form

The multiple-input version of the Structure Exploiting Arnoldi (SEA) algorithm for

$$\mathcal{M} = \mathbf{c} \otimes F + S \otimes I_{n_0} \in \mathbb{C}^{N \times N} \qquad \text{and} \qquad \mathcal{R} = \mathbf{c} \otimes R \in \mathbb{C}^{N \times m}$$

will be presented in the Section 3.4 of this thesis. This section develops a generic single-input Structure Exploiting Arnoldi algorithm for matrix $\mathcal{M}$ and single input vector $\mathcal{R} \in \mathbb{C}^N$. This algorithm will be adapted to include memory saving devices and a special subroutine to execute the stop condition as discussed in Section 3.3.

Algorithm 8 Generic Single-Input Structure Exploiting Arnoldi (SEA)

Input: $\mathbf{c}, F, S, \mathbf{r}$ in case one form 3.1.1.
Output: Orthonormal basis Q_k for the space S_{n+1} (3.6)

1. Compute $\mathbf{q}_1 \coloneqq \mathbf{r}/\|\mathbf{r}\|_2$, $b_1 \coloneqq 1$ and $k = 1$
2. Set $\hat{\mathbf{a}}_1 \coloneqq \mathbf{c} \otimes \mathbf{q}_1$
3. **for** $n = 1, 2, ...,$ until convergence **do**
4. Set $\hat{\mathbf{q}} \coloneqq F\hat{\mathbf{a}}_n$ and $\hat{\mathbf{q}}_{n+1} \coloneqq \hat{\mathbf{q}}$
5. **for** $i = 1, 2, ..., k$ **do**
6. $h_{b_i,n} \coloneqq \hat{\mathbf{q}}_{n+1}^T \mathbf{q}_i$
7. $\hat{\mathbf{q}}_{n+1} \coloneqq \hat{\mathbf{q}}_{n+1} - h_{b_i,n}\mathbf{q}_i$
8. **end for**
9. Check if $\hat{\mathbf{q}}_{n+1}$ is a basis vector for S_{n+1} by evaluating $\|\hat{\mathbf{q}}_{n+1}\|_2$
10. **if** $\hat{\mathbf{q}}_{n+1}$ is a basis vector **then**
11. Set $h_{n+1,n} \coloneqq \|\hat{\mathbf{q}}_{n+1}\|_2$, $\hat{\mathbf{q}}_{n+1} \coloneqq \hat{\mathbf{q}}_{n+1}/h_{n+1,n}$ and $\mathbf{q}_{k+1} \coloneqq \hat{\mathbf{q}}_{n+1}$
12. Set $b_{k+1} \coloneqq n + 1$ and $k \coloneqq k + 1$
13. **else**
14. Set $h_{n+1,n} \coloneqq 1$
15. **end if**
16. Set $\hat{\mathbf{a}}_{n+1} \coloneqq \frac{1}{h_{n+1,n}} \left(\mathbf{c} \otimes \hat{\mathbf{q}} + (S \otimes I_{n_0})\hat{\mathbf{a}}_n - \widehat{A}_n H(1:n, n) \right)$
17. **if** $\hat{\mathbf{a}}_{n+1} \in$ col span $\left(\widehat{A}_n\right)$ **then**
18. STOP
19. **end if**
20. **end for**

Remark 3.2.1. This generic version of the SEA algorithm produces an orthonormal basis of S_n for a more general class of matrices than those that arise from the linearization of ℓth-order systems. The specific algorithm used for model order reduction of higher-order systems will be presented in Section 3.3.

Basic relations between quantities generated by the generic single-input Structure Exploiting Arnoldi algorithm are readily available. Suppose the SEA algorithm 8 for matrices in case one form runs in exact arithmetic through the end of n iterations of the outer for-loop. Let $\{k_j\}_{j=1}^{n+1}$ be a finite monotonically increasing sequence with element k_j defined to be the number of basis vectors for S_j for $j = 1, .., n+1$. By design, k_1 is stored in the initialization of the generic single-input SEA algorithm and sequence element k_{j+1} is updated during the jth iteration of the outer for-loop, for $j = 1, 2, ..., n$.

The projection matrix, $\widehat{H}_n$, is the $(n+1)\times n$ Hessenberg matrix whose coefficients are calculated and stored in line 6, line 11, and line 14 of the SEA algorithm. Let

$$\widehat{A}_{n+1} = \begin{bmatrix} \hat{\mathbf{a}}_1 & \hat{\mathbf{a}}_2 & \cdots & \hat{\mathbf{a}}_n & \hat{\mathbf{a}}_{n+1} \end{bmatrix},$$

where $\hat{\mathbf{a}}_1 \in \mathbb{C}^N$ is calculated in the initialization and $\hat{\mathbf{a}}_{j+1} \in \mathbb{C}^N$ is the auxiliary vector calculated in the jth iteration of the SEA algorithm, for $j = 1, 2, ..., n$. Let $k = k_{n+1}$ and suppose $Q_k \in \mathbb{C}^{n_0 \times k}$ is the output of the SEA algorithm. By construction, the column vectors of Q_k form an orthonormal set. Define E_b as the $(n+1) \times k$ matrix whose jth column is the b_jth column of I_{n+1}, where b_j is the pointer stored in line 12 of the SEA algorithm, for $j = 1, 2, ..., k$. Let $\widehat{Q}_{n+1} = Q_k E_b^T$. The matrix relations generated by the generic single-input SEA algorithm are given by

$$F\widehat{A}_n = \widehat{Q}_{n+1}\widehat{H}_n, \tag{3.17}$$

$$\mathcal{M}\widehat{A}_n = \widehat{A}_{n+1}\widehat{H}_n. \tag{3.18}$$

Matrix $\widehat{H}_n$ is designed to be an $(n+1) \times n$ Hessenberg matrix with nonzero subdiagonal elements.

By Proposition 2.1.10, $\widehat{A}_n$ forms a basis matrix for $K(\mathcal{M}, \mathcal{R}, n)$ and there exists a nonsingular, upper-triangular matrix $\mathcal{U}_n$ such that

$$\widehat{A}_n = \begin{bmatrix} \mathcal{R} & \mathcal{M}\mathcal{R} & \cdots & \mathcal{M}^{n-1}\mathcal{R} \end{bmatrix} \mathcal{U}_n = A_n \, \mathcal{U}_n. \tag{3.19}$$

where A_n is defined as in Proposition 3.1.4. The matrix Q_k created and stored in the SEA algorithm forms a basis for the space S_{n+1} (3.6).

Theorem 3.2.2. Let $\mathcal{M}$ and $\mathcal{R}$ be in case one form. Suppose the generic single-input SEA algorithm 8 runs in exact arithmetic with input data $\mathbf{c}, F, S, \mathbf{r}$ through the end of n iterations of the outer for-loop and produces exactly $k = k_{n+1}$ basis vectors. Then the sequence of vectors $\mathbf{q}_1, \mathbf{q}_2, ..., \mathbf{q}_k$ forms an orthonormal basis of the space S_{n+1} (3.6).

Proof: By construction, the columns of Q_k are orthonormal. Upon initialization of the SEA algorithm, $\mathbf{q}_1 = \mathbf{w}_1/\|\mathbf{w}_1\|_2$. Also,

$$\widehat{Q}_{n+1}\widehat{H}_n = F\widehat{A}_n = FA_n\mathcal{U}_n = W_{n+1}E_n\mathcal{U}_n,$$

where $\mathcal{U}_n$ is a nonsingular upper-triangular matrix and $E_n \in \mathbb{C}^{(n+1)\times n}$ is formed by deleting the first column of I_{n+1}. The last equality is a consequence of the definition of W_{n+1} by Proposition 3.1.4. Since $W_{n+1}E_n = Q_k E_b^T \widehat{H}_n \mathcal{U}_n^{-1}$, any column of W_{n+1} can be written as a linear combination of the columns of Q_k. □

Finally, the last result of this section connects the Structure Exploiting Arnoldi algorithm with the Arnoldi algorithm explicitly.

Proposition 3.2.3. The generic single-input Structure Exploiting Arnoldi algorithm for matrices in case one form with input data $\mathbf{c}, F, S, \mathbf{r}$ stops at step n if, and only if, the Arnoldi algorithm with matrix $\mathcal{M} = \mathbf{c} \otimes F + S \otimes I_{n_0}$ and starting vector $\mathcal{R} = \mathbf{c} \otimes \mathbf{r}$ stops at step n.

Proof: Suppose that the Arnoldi algorithm stops at step n. Then, $K(\mathcal{M}, \mathcal{R}, n+1)$ is an n-dimensional space and if $\mathbf{a}_k = \mathcal{M}^{k-1}\mathcal{R}$, for $k = 1, 2, ..., n+1$ as in Proposition 3.1.4, the vector $\mathbf{a}_{n+1}$ is contained in the column span of A_n. To show that the SEA algorithm stops, the first goal is to establish that $F\hat{\mathbf{a}}_n \in$ span $\{\mathbf{q}_1, \mathbf{q}_2, ..., \mathbf{q}_{k_n}\}$ so that $\|\hat{\mathbf{q}}\|_2$ calculated in line 9 of the nth iteration of SEA will be zero. Since there exists a nonsingular $\mathcal{U}_n$ such that $\widehat{A}_n = A_n\mathcal{U}_n$,

$$F\hat{\mathbf{a}}_n = FA_n\mathcal{U}_n(1:n, n) = W_{n+1}E_n\mathcal{U}_n(1:n, n),$$

by the definition for the matrix W_{n+1} (3.8).

The vector $\mathbf{w}_{n+1}$ is the small space factor of the vector $\mathbf{a}_{n+1}$. As is demonstrated in Proposition 3.1.10, since $\mathbf{a}_{n+1}$ is in the range of A_n, $\mathbf{w}_{n+1}$ can be written as a linear combination of the

vectors $\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_n$. However, by Theorem 3.2.2,

$$\mathbf{w}_i \in \text{ span } \{\mathbf{q}_1, \mathbf{q}_2, ..., \mathbf{q}_k\}$$

for each $i = 1, 2, ..., n$. Thus, $F\hat{\mathbf{a}}_n$ must also be in the column span of Q_k. In this case, $h_{n+1,n} = 1$ by line 14.

The final trick for this direction of the proof is to recognize that $\hat{\mathbf{a}}_{n+1}$ is in the column span of $\widehat{A}_n$. By the SEA matrix equation (3.17), $\mathcal{M}\hat{\mathbf{a}}_n = \widehat{A}_{n+1}\widehat{H}_n(1 : n+1, n)$. Combining the matrix relation (3.19) with the fact that $h_{n+1,n} = 1$ and a little manipulation, notice that $\hat{\mathbf{a}}_{n+1} = \mathcal{M}A_n\mathbf{x}$ with $\mathbf{x} = \mathcal{U}_n(1 : n, n) - \mathcal{U}_n\widehat{H}_n(1 : n, n)$. Each column of $\mathcal{M}A_n$ is in the column span of A_n since Arnoldi stops at step n and there exists $\mathbf{y} \in \mathbb{C}^n$ such that $\mathcal{M}A_n\mathbf{x} = A_n\mathbf{y} = \widehat{A}_n\mathcal{U}_n^{-1}\mathbf{y} = \hat{\mathbf{a}}_{n+1}$. This is exactly the stop condition for the SEA algorithm given in line 17.

Conversely, if the SEA algorithm stops during the nth iteration of the outer for-loop, then $\hat{\mathbf{a}}_{n+1}$ is in the range of $\widehat{A}_n$. Moreover, the SEA matrix equation (3.18) holds for both the nth and $(n+1)$st iteration. By Proposition 3.1.4,

$$\text{col span } \left(\widehat{A}_n\right) = K(\mathcal{M}, \mathcal{R}, n) \qquad \text{and} \qquad \text{col span } \left(\widehat{A}_{n+1}\right) = K(\mathcal{M}, \mathcal{R}, n+1).$$

After n iterations of the SEA algorithm, the matrix $\widehat{A}_n$ has rank n. Thus, $K(\mathcal{M}, \mathcal{R}, n)$ has dimension n and $K(\mathcal{M}, \mathcal{R}, n) = K(\mathcal{M}, \mathcal{R}, n+1)$ guaranteeing that the Arnoldi algorithm stops after n iterations. □

Remark 3.2.4. In the nth iteration of the SEA algorithm, the vector

$$\hat{\mathbf{a}}_{n+1} = \frac{1}{\alpha}\left(\mathcal{M}\hat{\mathbf{a}}_n - \widehat{A}_n\mathbf{h}\right)$$

is updated in line 16 with $\alpha = h_{n+1,n}$ and $\mathbf{h} = \widehat{H}_n(1 : n, n) \in \mathbb{C}^n$. Any scalar $\alpha \neq 0$ and any $\mathbf{h} \in \mathbb{C}^n$ will produce the correct basis Q_k. The only requirement is that the Hessenberg relation (3.18) need be maintained. For example, the Arnoldi scaling coefficient could be used for the Hessenberg matrix associated with $\widehat{A}_{n+1}$ independent of the the orthogonalization coefficients stored in $\widehat{H}_n$ used to orthogonalize candidate basis vectors for S_{n+1}. Algorithm 8 attempts to maintain a connection between the matrix $\widehat{A}_n$ and the matrix $\widehat{Q}_{n+1}$. However, the realization that

$\widehat{A}_{n+1}$ can be updated independent of the orthogonalization coefficients for $\widehat{Q}_{n+1}$ suggests that the single-input SEA algorithm 8 is really a template for a whole class of algorithms based on the case one structure result (3.5).

3.3 Optimizing SEA for Higher-Order Linear Dynamical Systems

The generic single-input Structure Exploiting Arnoldi algorithm can be used to produce Padé-type reduced-order models of large-scale ℓth-order linear dynamical systems at a fraction of the cost as the Arnoldi-based reduction algorithm 4. Every aspect of this algorithm should be designed to minimize computational and storage requirements with this application in mind. Comparing the proposed SEA algorithm to the classic Arnoldi algorithm, the three main computational costs for producing the data necessary to construct sufficiently accurate reduced-order models in both cases include:

(1) The matrix-vector multiplication of the matrix $\mathcal{M}$ with the most recent basis vector for $K(\mathcal{M}, \mathcal{R}, n)$ to generate a candidate basis vector for $K(\mathcal{M}, \mathcal{R}, n+1)$.

(2) The inner products used to project the new candidate basis vector onto the span of previous basis vectors.

(3) The saxpy operations that update the new basis vector for $K(\mathcal{M}, \mathcal{R}, n+1)$ by subtracting scalar multiples of previous basis vectors for $K(\mathcal{M}, \mathcal{R}, n)$.

Both the Arnoldi algorithm and the SEA algorithm produce new basis vectors for $K(\mathcal{M}, \mathcal{R}, n+1)$ via multiplication with $\mathcal{M}$ during each iteration since both algorithms rely on the deflated block Krylov matrix induced by $\mathcal{M}$ and $\mathcal{R}$ in some form. The equivalent factorization of $\mathcal{M}$ in case one form gives an improved method for executing this multiplication, but this improvement can be realized in both the Arnoldi algorithm and the SEA algorithm. Similarly, the saxpy operations that update the new candidate vectors of $K(\mathcal{M}, \mathcal{R}, n+1)$ must also occur in some fashion in the SEA algorithm since multiplication by $\mathcal{M}$ without some "corrective" saxpy operations results in extremely ill-conditioned numerical computations.

The major computational advantage offered by the SEA algorithm is with respect to the inner products used to project new candidates onto the span of previous basis vectors. In the Arnoldi algorithm, these inner products are executed on vectors of size $N = n_0 \cdot \ell$. In contrast, the analogous

inner products in the SEA algorithm manipulate vectors of size n_0. This is ℓ times cheaper than the orthogonalization process in the Arnoldi algorithm. There is a slight increase in storage requirements for the SEA algorithm because two different set of basis vectors must be stored: the basis vectors for $K(\mathcal{M}, \mathcal{R}, n+1)$ and the basis for S_{n+1} (3.6). However, this trade-off yields more efficient numerical computations and improved theoretic properties. Further, there are specific adaptations that can be made to the SEA algorithm to produce a more efficient and faster moment-matching model order reduction technique for ℓth-order linear dynamical systems. These adaptations are designed to exploit the special structure of the matrices described in Proposition 3.1.2 with strictly lower-triangular

$$S = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ s_{21} & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ s_{\ell,1} & \cdots & s_{\ell,\ell-1} & 0 \end{bmatrix}. \tag{3.20}$$

3.3.1 The Stop Condition via the Full-Rank Factorization

One opportunity to speed up the generic single-input SEA algorithm 8 is to develop a fast and accurate method to check the stop condition in line 17 without manipulating vectors of size N. Assume the SEA algorithm runs in exact arithmetic for n iterations on matrices in case one form and that S is strictly lower-triangular (3.20). Let $\widehat{A}_n$ have full column rank, let k_j denote the number of basis vectors for S_j for $j = 1, 2, ..., n+1$ and let $k = k_{n+1}$. Suppose $\beta = \{b_1, b_2, ..., b_k\}$ is the set of pointers stored in line 12 and let $\delta = \{1, ..., n, n+1\} - \beta$. Denote the jth smallest element of δ by d_j for $j = 1, ..., n-k+1$ where $d_i < d_j$ if $i < j$. Recall that $F\widehat{A}_n = \widehat{Q}_{n+1}\widehat{H}_n$ (3.17) and $\widehat{Q}_{n+1} = Q_k E_b^T$ where the jth column of $E_b \in \mathbb{C}^{(n+1)\times k}$ is the b_jth column of I_{n+1} .

The contrapositive of Proposition 3.1.10 indicates that if

$$\hat{\mathbf{q}}_{n+1} = \frac{1}{h_{n+1,n}} \left(F\hat{\mathbf{a}}_n - \widehat{Q}_n \widehat{H}_n(1:n, n) \right)$$

is nonzero, the corresponding $\hat{\mathbf{a}}_{n+1}$ is not in the range of $\widehat{A}_n$ and no stop condition needs to be checked. Only when $\hat{\mathbf{q}}_{n+1} = \mathbf{0}$ is it necessary to check the column rank of $\widehat{A}_{n+1}$. If $\hat{\mathbf{q}}_{n+1} = \mathbf{0}$ and

$\hat{\mathbf{a}}_{n+1}$ is calculated in line 16, then $\widehat{A}_{n+1}$ can be written as

$$\widehat{A}_{n+1} = (I_\ell \otimes Q_k)X, \qquad \text{with} \qquad X = \begin{bmatrix} X_{n+1}^{(1)} \\ \vdots \\ X_{n+1}^{(\ell)} \end{bmatrix},$$

$Q_k \in \mathbb{C}^{n_0 \times k}$ and $X_{n+1}^{(i)} \in \mathbb{C}^{k \times (n+1)}$ for $i = 1, 2, ..., \ell$, as is demonstrated in Section 3.1. $\widehat{A}_{n+1}$ is column rank deficient if, and only if, the matrix X has linearly dependent columns. With a slight modification of the update for $\hat{\mathbf{a}}_{n+1}$, matrices $X_{n+1}^{(i)}$ are easily calculated such that the rank of X may be quickly ascertained.

3.3.2 Optimizing Updates for the Columns of $\widehat{A}_{n+1}$

The update formula for $\hat{\mathbf{a}}_{n+1}$ in line 16 is suboptimal in the sense that checking the rank of the resulting X requires a workload comparable to the work saved by using inner products of size n_0. Instead, at every iteration, replace line 16 with the update

$$\hat{\mathbf{a}}_{n+1} = \mathbf{c} \otimes \hat{\mathbf{q}}_{n+1} + (S \otimes I_{n_0})\widehat{A}_n \big(\widehat{H}_n(2:n, 1:n)\big)^{-1} \mathbf{e}_n, \tag{3.21}$$

where $\mathbf{e}_n$ is the last column of I_n and $\hat{\mathbf{q}}_{n+1}$ is the $(n+1)$st column vector of $\widehat{Q}_{n+1} = Q_k E_b^T$. The matrix version of this update is

$$\widehat{A}_{n+1} = \mathbf{c} \otimes \widehat{Q}_{n+1} + (S \otimes I_{n_0})\widehat{A}_n \begin{bmatrix} \mathbf{0} & \big(\widehat{H}_n(2:n, 1:n)\big)^{-1} \end{bmatrix}.$$

Multiplying both sides of this equation by the matrix $\widehat{H}_n$ verifies that this modified update still maintains the necessary Hessenberg relationship (3.18). As is demonstrated below, this update formula guarantees

$$\widehat{A}_{n+1} = (I_\ell \otimes Q_k) \begin{bmatrix} c_1 E_b^T \\ X_{n+1}^{(2)} \\ \vdots \\ X_{n+1}^{(\ell)} \end{bmatrix}.$$

Only when $\hat{\mathbf{q}}_{n+1} = \mathbf{0}$ will the stop condition need to be evaluated in which case the last column of E_b^T is $\mathbf{0} \in \mathbb{C}^k$. An equivalent method to check the stop condition in line 17 is readily available.

Lemma 3.3.1 ([**BS05b**])**.** Suppose that $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n \in \mathbb{C}^{(\ell \cdot k)}$ is a sequence of linearly independent vectors with partition $\mathbf{x}_i = \begin{bmatrix} \mathbf{z}_i^T & \mathbf{p}_i^T \end{bmatrix}^T$ where $\mathbf{z}_i \in \mathbb{C}^k$ and $\mathbf{p}_i \in \mathbb{C}^{(\ell-1)\cdot k}$. Assume that the sets $\beta = \{b_j\}_{j=1}^k$ and $\delta = \{d_j\}_{j=1}^{n-k}$ form a partition of $\{1, 2, ..., n\}$. Suppose that the subsequence $\{\mathbf{z}_{b_j}\}_{j=1}^k$ is linearly independent and $\mathbf{z}_{d_j} = \mathbf{0}$ for $j = 1, 2, ..., n-k$. For any vector $\mathbf{x} = \begin{bmatrix} \mathbf{0}^T & \mathbf{p}^T \end{bmatrix}^T$ with $\mathbf{0} \in \mathbb{C}^k$ and $\mathbf{p} \in \mathbb{C}^{(\ell-1)\cdot k}$, $\mathbf{x}$ is in the span of $\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$ if, and only if, $\mathbf{p}$ is in the span of $\{\mathbf{p}_{d_1}, \mathbf{p}_{d_2}, ..., \mathbf{p}_{d_{n-k}}\}$.

Proof: If $\mathbf{x} \in$ span $\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$, there exists scalars $\{\alpha_i\}_{i=1}^n$ such that $\mathbf{x} = \sum_{i=1}^{n} \alpha_i \mathbf{x}_i$ and $\mathbf{0} = \sum_{i=1}^{n} \alpha_i \mathbf{z}_i = \sum_{i=1}^{k} \alpha_{b_i} \mathbf{z}_{b_i}$. Because $\mathbf{z}_{b_1}, ..., \mathbf{z}_{b_k}$ are linearly independent, $\alpha_{b_i} = 0$ for $i = 1, ..., k$. In other words, $\mathbf{p} = \sum_{i=1}^{n-k} \alpha_{d_i} \mathbf{p}_{d_i}$ and the proof is complete. □

The above result indicates that evaluating the stop condition during the nth iteration of the SEA algorithm can be done by working with a sequence of $n - k + 1$ vectors with each of size $(\ell - 1)k$ and $k \leq n + 1 \ll n_0 < N$ for applications in reduced-order modeling. This is a significant savings compared to manipulating vectors of size $N = \ell \cdot n_0$.

3.3.3 Generating the $X^{(i)}$ Matrices

It is relatively inexpensive to generate matrices $X_{n+1}^{(2)}$,...,$X_{n+1}^{(\ell)}$ corresponding to the modified update (3.21). Since $\hat{\mathbf{a}}_1 = \mathbf{c} \otimes \mathbf{q}_1$, $x_{11}^{(i)} = c_i$ for $i = 1, 2, ..., \ell$. By definition of the sequence $\{k_j\}_{j=1}^{n+1}$, after $n - 1$ iterations, there are k_n basis vectors for S_n. Assume

$$\widehat{A}_n = I_\ell \otimes Q_{k_n} \begin{bmatrix} c_1 E_b^T \\ X_n^{(2)} \\ \vdots \\ X_n^{(\ell)} \end{bmatrix}$$

where the b_jth column of $E_b^T \in \mathbb{C}^{k_n \times n}$ is the jth column of I_{k_n}. There are two possibilities to consider for the nth iteration of the SEA algorithm. If $\hat{\mathbf{q}}_{n+1} \neq \mathbf{0}$, then $k_{n+1} = k_n + 1$ and $\hat{\mathbf{a}}_{n+1} = \mathbf{c} \otimes \hat{\mathbf{q}}_{n+1} + (S \otimes I_{n_0}) \widehat{A}_n \mathbf{g}_n$. Here $\mathbf{g}_n$ is the last column of $\left(\hat{H}_n(2 : n+1, 1 : n)\right)^{-1}$. If $\mathbf{e}_{k_{n+1}}$

is the k_{n+1}st column of $I_{k_{n+1}}$, then $\hat{\mathbf{q}}_{n+1} = Q_{k_{n+1}}\mathbf{e}_{k_{n+1}}$. By properties of the Kronecker product

$$\hat{\mathbf{a}}_{n+1} = \mathbf{c} \otimes Q_{k_{n+1}}\mathbf{e}_{k_{n+1}} + (I_\ell \otimes Q_{k_n}) \begin{bmatrix} \sum_{t=1}^{\ell} s_{1,t}\mathbf{y}^{(t)} \\ \vdots \\ \sum_{t=1}^{\ell} s_{\ell,t}\mathbf{y}^{(t)} \end{bmatrix},$$

with $\mathbf{y}^{(1)} = c_1 E_b^T \mathbf{g}_n \in \mathbb{C}^{k_n}$ and $\mathbf{y}^{(i)} = X_n^{(i)}\mathbf{g}_n \in \mathbb{C}^{k_n}$ for $i = 2, ..., \ell$. Because $s_{1j} = 0$ for all $j = 1, ..., \ell$ by the assumption that S is strictly lower-triangular (3.20), $\sum_{t=1}^{\ell} s_{1t}\mathbf{y}^{(t)} = 0$. In the case where $k_{n+1} \neq k_n$, the proper update for the last column of $X_{n+1}^{(i)}$ is

$$X^{(i)}(1 : k_n + 1, n + 1) = \begin{bmatrix} \mathbf{0} \\ c_i \end{bmatrix} + \begin{bmatrix} \sum_{j=1}^{\ell} s_{i,j}\mathbf{y}^{(j)} \\ 0 \end{bmatrix} \in \mathbb{C}^{k_n+1}$$

for $i = 2, ..., \ell$ while $X^{(1)}(1 : k_n + 1, n + 1) = c_1\mathbf{e}_{k_n+1}$.

On the other hand, if $\hat{\mathbf{q}}_{n+1} = \mathbf{0}$, then $k_{n+1} = k_n$. A similar argument to the one above indicates that $X^{(i)}(1 : k_n, n+1) = \sum_{j=1}^{\ell} s_{i,j}\mathbf{y}^{(j)} \in \mathbb{C}^{k_n}$ while $X^{(1)}(1 : k_n, n+1) = \mathbf{0} \in \mathbb{C}^{k_n}$. In either case, these formulas provide an accurate and relatively inexpensive method of producing the matrix factors $X_{n+1}^{(i)}$. According to Lemma 3.3.1, the SEA algorithm stops if, and only if, $\hat{\mathbf{q}}_{n+1} = \mathbf{0}$ and

$$\mathbf{p}_{n-k_n+1} \in \text{ span } \{\mathbf{p}_1, ..., \mathbf{p}_{n-k_n}\} \qquad \text{with} \qquad \mathbf{p}_j = \begin{bmatrix} X_{n+1}^{(2)}(:, d_j) \\ \vdots \\ X_{n+1}^{(\ell)}(:, d_j) \end{bmatrix} \in \mathbb{C}^{(\ell-1)\cdot k_n}, \tag{3.22}$$

for $j = 1, 2, ..., n - k_n + 1$. This check can be executed using the modified Gram-Schmidt algorithm, the SVD algorithm or the rank-revealing QR factorization. The following subroutine generates the coefficients for $X^{(i)}(1 : k_n : n + 1)$ and facilitates the evaluation of the equivalent stop condition.

Algorithm 9 Forming Candidate Coefficients for $X^{(i)}(1 : k_n, n + 1)$

Input: Coefficients $\mathbf{g}_n$ and matrices S, $X_n^{(i)}$ for $i = 1, ..., \ell$.
Output: Coefficients $\mathbf{x}^{(i)} = X^{(i)}(1 : k_n, n + 1)$ for $i = 1, 2, ..., \ell$

1. Set $\mathbf{y}_i \coloneqq X_n^{(i)}\mathbf{g}_n$ for $i = 1, 2, ..., \ell$
2. Set $\begin{bmatrix} \mathbf{x}^{(1)} & \mathbf{x}^{(2)} & \cdots & \mathbf{x}^{(\ell)} \end{bmatrix} \coloneqq \begin{bmatrix} \mathbf{y}_1 & \mathbf{y}_2 & \cdots & \mathbf{y}_\ell \end{bmatrix} S^T$

3.3.4 Computationally Efficient Single-Input SEA Algorithm

The generic single-input SEA algorithm for matrices in case one form with strictly lower-triangular S (3.20) can be restated with the alternative update for $\hat{\mathbf{a}}_{n+1}$ (3.21). This enables a faster algorithm of reduced-order modeling of higher-order linear dynamical systems with an accelerated stop condition (3.22).

Algorithm 10 Specialized Single-Input SEA for Strictly Lower-Triangular S

Input: $F, \mathbf{c}, \mathbf{r}$ and strictly lower triangular S in case one form 3.1.1.
Output: Orthonormal basis Q_k for the space S_{n+1} (3.6)

1. Set $\mathbf{q}_1 \coloneqq \mathbf{r}/\|\mathbf{r}\|_2$, $b_1 \coloneqq 1$, and $k \coloneqq 1$
2. Set $\hat{\mathbf{a}}_1 \coloneqq \mathbf{c} \otimes \mathbf{q}_1$ and $x_{11}^{(i)} \coloneqq c_i$ for $i = 1, 2, ..., \ell$
3. **for** $n = 1, 2, ...$ **do**
4. $\quad \hat{\mathbf{q}}_{n+1} \coloneqq F\hat{\mathbf{a}}_n$
5. $\quad$ **for** $i = 1, 2, ..., k$ **do**
6. $\quad\quad h_{b_i,n} \coloneqq \hat{\mathbf{q}}_{n+1}^T \mathbf{q}_i$
7. $\quad\quad \hat{\mathbf{q}}_{n+1} \coloneqq \hat{\mathbf{q}}_{n+1} - h_{b_i,n}\mathbf{q}_i$
8. $\quad$ **end for**
9. $\quad$ Check if $\hat{\mathbf{q}}_{n+1}$ is a basis vector for S_{n+1}
10. $\quad$ **if** $\hat{\mathbf{q}}_{n+1}$ is a basis vector **then**
11. $\quad\quad$ Set $h_{n+1,n} \coloneqq \|\hat{\mathbf{q}}_{n+1}\|_2$, $\hat{\mathbf{q}}_{n+1} \coloneqq \hat{\mathbf{q}}_{n+1}/h_{n+1,n}$ and $\mathbf{q}_{k+1} \coloneqq \hat{\mathbf{q}}_{n+1}$
12. $\quad\quad$ Set $X^{(i)}(1:k+1, n+1) \coloneqq \begin{bmatrix} \left(\mathbf{x}^{(i)}\right)^T & c_i \end{bmatrix}^T$ with $\mathbf{x}^{(i)}$ from algorithm 9
13. $\quad\quad$ Set $b_{k+1} \coloneqq n+1$ and $k \coloneqq k+1$
14. $\quad$ **else**
15. $\quad\quad$ Set $X^{(i)}(1:k, n+1) = \mathbf{x}^{(i)}$ from algorithm 9
16. $\quad\quad$ Evaluate equivalent stop check using $\{X_{n+1}^{(i)}\}_{i=2}^{\ell}$ and $\{d_j\}_{j=1}^{n-k+1}$ (3.22).
17. $\quad\quad$ **if** Large space is exhausted **then**
18. $\quad\quad\quad$ STOP
19. $\quad\quad$ **else**
20. $\quad\quad\quad$ Set $h_{n+1,n} \coloneqq 1$ and $d_{n-k+1} = n+1$
21. $\quad\quad$ **end if**
22. $\quad\quad$ Set $\hat{\mathbf{a}}_{n+1} \coloneqq \mathbf{c} \otimes \hat{\mathbf{q}}_{n+1} + (S \otimes I_{n_0})\widehat{A}_n \, \widehat{H}_n(2:n+1, 1:n)^{-1} \, \mathbf{e}_n$
23. $\quad$ **end if**
24. **end for**

Remark 3.3.2. The alternative update for the auxiliary vector $\hat{\mathbf{a}}_{n+1}$ (3.21) can be calculated without forming the matrix $(S \otimes I_{n_0})$. Let $\mathbf{g}_n$ be the last column of $\left(\widehat{H}_n(2:n, 1:n)\right)^{-1}$ and let $\mathbf{a}^{(i)} = \widehat{A}_n\big((i-1)n_0 + 1 : i \cdot n_0, 1:n\big)\mathbf{g}_n$ for $i = 1, 2, ..., \ell$. Then set

$$\begin{bmatrix} \mathbf{y}_1 & \mathbf{y}_2 & \cdots & \mathbf{y}_\ell \end{bmatrix} = \hat{\mathbf{q}}_{n+1}\mathbf{c}^T + \begin{bmatrix} \mathbf{a}^{(1)} & \mathbf{a}^{(2)} & \cdots & \mathbf{a}^{(\ell)} \end{bmatrix} S^T.$$

Finally, the alternative update is given by $\hat{\mathbf{a}}_{n+1} = \begin{bmatrix} \mathbf{y}_1^T & \cdots & \mathbf{y}_\ell^T \end{bmatrix}^T$. This takes advantage of fast matrix-matrix multiplication routines that can be executed quickly if parallelized BLAS subroutines are available [**Don02**].

Remark 3.3.3. This computationally efficient version of the SEA algorithm uses the same Hessenberg matrix inversion trick introduced in the SOAR algorithm [**BS05b**]for any ℓth-order system with $\ell \geq 2$. The improved SEA checks the stop condition by working with vectors of size $(\ell - 1)k$ where $k \leq n + 1 \ll n_0$ while the SOAR algorithm evaluates the stop condition by working with vectors of size $(\ell - 1)n_0$ in the special case of $\ell = 2$.

Remark 3.3.4. This improved version of the SEA algorithm designed for dimension reduction of ℓth-order systems is similar to output of the SOAR algorithm [**BS05b**] which solves the special $\ell = 2$ case. However, the terminology used to describe the origins of each algorithm differs. The SOAR algorithm relies on second-order Krylov subspaces, suggesting that suitable extensions of Krylov subspaces form the theoretical underpinnings of structure exploiting algorithms. In contrast, the SEA algorithm is based on the case one structure Proposition 3.1.3 illustrating that it is not necessary to introduce terminology to describe extended Krylov subspaces. Instead, structure exploiting algorithms should generate an orthonormal basis for the multiple copied subspace S_n that arises in the factorization of any basis matrix for the nth-order Krylov subspace induced $\mathcal{M}$ and $\mathcal{R}$ in case one form.

3.4 The Band SEA Algorithm for Matrices in Case One Form

The generic single-input version of the SEA algorithm is designed to process systems with a single input vector. An analogous generic band version can be adapted for systems with multiple inputs. The band SEA algorithm presented below yields an orthornormal basis for S_n (3.6) in the case of a block of starting vectors. The subroutine `band_SEA_start` initializes the band SEA algorithm by preprocessing $R \in \mathbb{C}^{n_0 \times m}$ with a modified Gram-Schmidt orthogonalization to produce an orthonormal basis for S_{m_1} given by $Q_1 = \begin{bmatrix} \mathbf{q}_1 & \cdots & \mathbf{q}_{m_1} \end{bmatrix}$. Part of this preprocessing includes the determination of the proper elimination matrix $E_1 \in \mathbb{C}^{m \times m_1}$ and may also include the calculation and storage of $\rho_m \in \mathbb{C}^{m_1 \times m}$ such that $R = Q_1 \rho_m$. A basis for the column span of $\mathcal{R} = \mathbf{c} \otimes R$ is given by $\begin{bmatrix} \hat{\mathbf{a}}_1 & \cdots & \hat{\mathbf{a}}_{m_1} \end{bmatrix} = \mathbf{c} \otimes \begin{bmatrix} \mathbf{q}_1 & \cdots & \mathbf{q}_{m_1} \end{bmatrix}$.

Algorithm 11 Generic Band Structure Exploiting Arnoldi

Input: $\mathbf{c}, F, S$ and $R \in \mathbb{C}^{n_0 \times m}$ in case one form 3.1.1 and integer nmax $> m$.
Output: ONB Q_k for S_{nmax} (3.6)

1. Initialize Band SEA Algorithm: $[Q_{m_1}, m_1, \widehat{A}_{m_1}] \coloneqq \texttt{band\_SEA\_start}(\mathbf{c}, R)$
2. Set $m_c \coloneqq m_1$, $\{b_i \coloneqq i\}_{i=1}^{m_1}$ and $n \coloneqq m_1$
3. **while** $n <$ nmax **do**
4. Set $\hat{\mathbf{q}} \coloneqq F\hat{\mathbf{a}}_{n-m_c+1}$ and $\hat{\mathbf{q}}_{n+1} \coloneqq \hat{\mathbf{q}}$
5. Set $\mathbf{h} \coloneqq \mathbf{0} \in \mathbb{C}^n$
6. **for** $i = 1, 2, ..., k$ **do**
7. $h(b_i, 1) \coloneqq \hat{\mathbf{q}}_{n+1}^T \mathbf{q}_i$
8. $\hat{\mathbf{q}}_{n+1} \coloneqq \hat{\mathbf{q}}_{n+1} - h(b_i, 1)\mathbf{q}_i$
9. **end for**
10. Set $\widetilde{H}(1 : n, n - m_c + 1) \coloneqq \mathbf{h}$
11. Decide if $\hat{\mathbf{q}}_{n+1}$ is a basis vector.
12. **if** $\hat{\mathbf{q}}_{n+1}$ is a basis vector **then**
13. Set $\widetilde{H}(n + 1, n - m_c + 1) \coloneqq \|\hat{\mathbf{q}}_{n+1}\|_2$
14. $\hat{\mathbf{q}}_{n+1} \coloneqq \hat{\mathbf{q}}_{n+1}/\|\hat{\mathbf{q}}_{n+1}\|_2$ and $\mathbf{q}_{k+1} \coloneqq \hat{\mathbf{q}}_{n+1}$
15. Set $\hat{\mathbf{a}}_{n+1} \coloneqq \frac{1}{\|\hat{\mathbf{q}}_{n+1}\|_2}\left(\mathbf{c} \otimes \hat{\mathbf{q}} + (S \otimes I_{n_0})\hat{\mathbf{a}}_{n-m_c+1} - \widehat{A}_n\mathbf{h}\right)$
16. Set $b_{k+1} \coloneqq n + 1$, $k \coloneqq k + 1$ and $n \coloneqq n + 1$
17. **else**
18. Set $\hat{\mathbf{a}} \coloneqq \left(\mathbf{c} \otimes \hat{\mathbf{q}} + (S \otimes I_{n_0})\hat{\mathbf{a}}_{n-m_c+1} - \widehat{A}_n\mathbf{h}\right)$
19. Decide if $\hat{\mathbf{a}} \in$ range $\widehat{A}_n$
20. **if** Deflation should occur **then**
21. Set $m_c = m_c - 1$.
22. **if** $m_c = 0$ **then**
23. STOP
24. **end if**
25. **else**
26. Set $\widetilde{H}(n + 1, n - m_c + 1) \coloneqq 1$, $\mathbf{a}_{n+1} \coloneqq \hat{\mathbf{a}}$, and $n \coloneqq n + 1$
27. **end if**
28. **end if**
29. **end while**

Suppose that the band SEA algorithm 11 runs until $m_c = 0$. Then $n = N_0$ is the column size of the deflated block Krylov matrix induced by $\mathcal{M}$ and $\mathcal{R}$. Let

$$n_j = m_1 + m_2 + \cdots m_j \tag{3.23}$$

for some $j = 1, 2, ..., k_0$, where $\{m_j\}_{j=1}^{k_0}$ encodes the exact deflation pattern of the deflated block Krylov matrix induced by $\mathcal{M}$ and $\mathcal{R}$ in case one form. Let $\widehat{Q}_1 = Q_1$ and $\widehat{Q}_{i+1} = \begin{bmatrix} \hat{\mathbf{q}}_{m_i+1} & \cdots & \hat{\mathbf{q}}_{m_{i+1}} \end{bmatrix}$ for $i = 1, ..., j$ where $\hat{\mathbf{q}}_{n+1}$ is the vector calculated in the while-loop of the band SEA algorithm for $n = m_1, ..., n_{j+1} - 1$. Suppose that $Q_i \in \mathbb{C}^{n_0 \times w_i}$ stores the w_i linearly independent columns of the

block $\widehat{Q}_i$ for $i = 1, 2, ..., j+1$. In this case,

$$\widehat{Q}_i = Q_i E_{b_i}^T$$

where $E_{b_i} \in \mathbb{C}^{m_i \times w_i}$ is the deflated identity matrix I_{m_i} whose missing columns correspond to the zero columns of $\widehat{Q}_i$, for $i = 1, 2, ..., j+1$.

Let $\widehat{A}_i = \begin{bmatrix} \hat{\mathbf{a}}_{m_{i-1}+1} & \cdots & \hat{\mathbf{a}}_{m_i} \end{bmatrix} \in \mathbb{C}^{N \times m_i}$ for $i = 2, ..., j+1$ with $\hat{\mathbf{a}}_{n+1}$ stored in the while-loop of the band SEA algorithm for $n = m_1, ..., n_{j+1} - 1$. Set $V_{n_{j+1}} = \begin{bmatrix} \widehat{A}_1 & \cdots & \widehat{A}_j & \widehat{A}_{j+1} \end{bmatrix}$. The resulting matrix equations for the band SEA algorithm are given by

$$F\, V_{n_j} = \begin{bmatrix} \widehat{Q}_1 & \widehat{Q}_2 & \cdots & \widehat{Q}_{j+1} \end{bmatrix} \widetilde{H_{n_j}} \qquad \text{and} \qquad \mathcal{M} V_{n_j} = V_{n_{j+1}} \widetilde{H_{n_j}}, \tag{3.24}$$

where $\widetilde{H_{n_j}} \in \mathbb{C}^{n_{j+1} \times n_j}$ is a deflation-revealing Hessenberg matrix corresponding to the Krylov subspace $K(\mathcal{M}, \mathcal{R}, n_{j+1})$. By Proposition 2.1.11, V_{n_j} forms a basis for $K(\mathcal{M}, \mathcal{R}, n_j)$. Moreover, the output vectors $\{\mathbf{q}_i\}_{i=1}^k$ forms an orthonormal basis for S_{n+1}. The proof of this result follows from a straight forward generalization of the analogous proof in Section 3.2.

Many of the computational improvements suggested to improve the single-input SEA for applications in model order reduction of higher-order linear dynamical systems can be adapted to the band version. Assuming strictly lower triangular $S \in \mathbb{C}^{\ell \times \ell}$ (3.20), the band analog to the alternative update (3.21) follows from a modification of $\widetilde{H_{n_j}}$. Multiply both sides of the Band SEA matrix equation (3.24) by

$$E_{B_{p_j}} = \begin{bmatrix} E_2 & 0 & \cdots & 0 \\ 0 & E_3 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & E_{j+1} \end{bmatrix},$$

and set $\widehat{H_{\hat{n}_j}} = \left(\widetilde{H_{n_j}} E_{B_{p_j}} \right) \in \mathbb{C}^{n_{j+1} \times \hat{n}_j}$ with $\hat{n}_j = n_{j+1} - m_1$. Then,

$$F\, V_{n_j}\, E_{B_{p_j}} = \begin{bmatrix} \widehat{Q}_1 & \widehat{Q}_2 & \cdots & \widehat{Q}_{j+1} \end{bmatrix} \widehat{H_{\hat{n}_j}}$$

and the matrix $\widehat{H_{\hat{n}_j}}$ has subdiagonal bandwidth equal to m_1 with $\widehat{H_{\hat{n}_j}}(m_1 + i, i) > 0$ for $i = 1, 2, ..., \hat{n}_j$. Moreover, the matrix $\widehat{H_{\hat{n}_j}}(m_1 + 1 : m_1 + i, 1 : i)$ is nonsingular for $i = 1, 2, ..., \hat{n}_j$. Given

$0 \in \mathbb{C}^{\hat{n}_j \times m_1}$ and $G_{\hat{n}_j} = \left(\widehat{H_{\hat{n}_j}}(m_1+1 : m_1+\hat{n}_j, 1:\hat{n}_j)\right)^{-1}$, the matrix equation for an alternative update is

$$V_{n_{j+1}} = \mathbf{c} \otimes \begin{bmatrix} \widehat{Q}_1 & \widehat{Q}_2 & \cdots & \widehat{Q}_{j+1} \end{bmatrix} + (S \otimes I_{n_0}) V_{n_j} E_{B_{p_j}} \begin{bmatrix} 0 & G_{\hat{n}_j} \end{bmatrix}. \tag{3.25}$$

Multiplying both sides of this equation by $\widehat{H_{\hat{n}_j}}$ yields $\mathcal{M} V_{n_j} E_{B_{p_j}} = V_{n_{j+1}} \widehat{H_{\hat{n}_j}}$.

The equivalent stop condition for this alternative update requires the construction of matrices $X^{(i)}$ such that

$$\begin{bmatrix} \widehat{A}_1 & \widehat{A}_2 & \cdots & \widehat{A}_j \end{bmatrix} = \left(I_\ell \otimes \begin{bmatrix} Q_1 & Q_2 & \cdots & Q_j \end{bmatrix} \right) \begin{bmatrix} c_1 E_b^T \\ X_{n_j}^{(2)} \\ \vdots \\ X_{n_j}^{(\ell)} \end{bmatrix}$$

Subblock partitions for matrices $X_{n_j}^{(i)}$ are given in Section 3.1. The first $m_1 \times m_1$ diagonal blocks for each $X^{(i)}$ is equal to $c_i I_{m_1}$ for $i = 1, ..., \ell$. Subsequent columns of $X^{(i)}$ can be calculated using the column relationships illustrated in the alternative update formula (3.25). If S is strictly lower triangular, then $s_{j,t} = 0$ for $j = 1, ..., \ell$ and $t = j, j+1, ..., \ell$ ensuring that the results of Lemma 3.3.1 apply. While a block form of this update is convenient for stating the results, the band SEA algorithm using alternate update equation (3.25) must generate matrix Q column by column. The algorithm below gives the optimized band SEA method including proper treatment of the equivalent stop condition using matrices $X^{(i)}$.

Remark 3.4.1. The band SEA algorithm 12 for reduced-order modeling of multiple-input higher-order systems can be further optimized to decrease storage requirements and improve the numerical behavior of this algorithm. Refinements of this algorithm come from exploring the structure of the Hessenberg matrix $\widehat{H}_{\hat{n}_j}$. Rows of $\widehat{H}_{\hat{n}_j}$ whose indices are stored by pointers $\{d_j\}_{j=1}^{n_{j+1}}$ will have a single nonzero entry whose value will be one. Using a trick similar to the step-by-step solutions obtained in backward substitution for solving upper-triangular linear systems, the unknown variables corresponding to these "identity" rows may be eliminated when working with the inverse $\widehat{H}_{\hat{n}_j}^{-1}$. The pseudo code below only hints at this trick. In the numerical results presented in Section 3.5, such a modification improved performance for reduced-order modeling purposes.

Algorithm 12 Specialized Band SEA algorithm for Strictly Lower-Triangular S

Input: $F, \mathbf{c}$, $R \in \mathbb{C}^{n_0 \times m}$ 3.1.1, S strictly lower triangular, integer nmax $> m$.
Output: ONB Q_k for S_{nmax} (3.6)

1. Initialize Band SEA Algorithm: $[Q_1, m_1, \widehat{A}_1] \coloneqq \texttt{band\_SEA\_start}(\mathbf{c}, R)$
2. Set $m_c = m_1$, $\{b_i = i\}_{i=1}^{m_1}$, $n \coloneqq m_1$, $X_{m_1}^{(i)} = c_i I_{m_1}$,
3. **while** $n <$ nmax **do**
4. Set $\hat{\mathbf{q}}_{n+1} \coloneqq F\hat{\mathbf{a}}_{n-m_c+1}$
5. Set $\texttt{tmp} \coloneqq \mathbf{0} \in \mathbb{C}^n$
6. **for** $i = 1, 2, ..., k$ **do**
7. $\texttt{tmp}(b_i, 1) \coloneqq \hat{\mathbf{q}}_{n+1}^T \mathbf{q}_i$
8. $\hat{\mathbf{q}}_{n+1} \coloneqq \hat{\mathbf{q}}_{n+1} - \texttt{tmp}(b_i, 1)\mathbf{q}_i$
9. **end for**
10. Decide if $\hat{\mathbf{q}}_{n+1}$ is a basis vector.
11. **if** $\hat{\mathbf{q}}_{n+1}$ is a basis vector **then**
12. Set $\widehat{H}(1 : n, n - m_1 + 1) \coloneqq \texttt{tmp}$ and $\widehat{H}(n + 1, n - m_1 + 1) \coloneqq \|\hat{\mathbf{q}}_{n+1}\|_2$
13. Set $\hat{\mathbf{q}}_{n+1} \coloneqq \frac{1}{\|\hat{\mathbf{q}}_{n+1}\|_2}\hat{\mathbf{q}}_{n+1}$, $\mathbf{q}_{k+1} \coloneqq \hat{\mathbf{q}}_{n+1}$, $b_{k+1} \coloneqq n + 1$ and $B_{p_{n-m_1+1}} \coloneqq n - m_c + 1$
14. Set $\mathbf{g}_{n-m_1+1} = \widehat{H}_n(m_1 + 1 : n + 1, 1 : n - m_1 + 1)^{-1}\, \mathbf{e}_{n-m_1+1}$
15. Set $\hat{\mathbf{a}}_{n+1} \coloneqq \mathbf{c} \otimes \hat{\mathbf{q}}_{n+1} + (S \otimes I_{n_0})\widehat{A}_n(:, B_p)\, \mathbf{g}_{n-m_1+1}$
16. Set $\mathbf{y}_i \coloneqq X^{(i)}(1 : k, B_p)\mathbf{g}_{n-m_1+1}$ and $X^{(i)}(1 : k + 1, n + 1) \coloneqq \begin{bmatrix} \sum_{t=1}^{\ell} s_{i,t}\mathbf{y}_i^T & c_i \end{bmatrix}^T$
17. Set $k \coloneqq k + 1$ and $n \coloneqq n + 1$
18. **else**
19. Set $\mathbf{g}_{n-m_1+1} = -\left(\widehat{H}(m_1 + 1 : n, 1 : n - m_1)\right)^{-1} \texttt{tmp}(m_1 + 1 : n, 1)$
20. Set $\mathbf{x}_i = \sum_{t=1}^{\ell} s_{i,t} \left(X^{(t)}(1 : k, B_p)\mathbf{g}_{n-m_1+1} + X^{(t)}(1 : k, n - m_c + 1)\right)$
21. Decide deflation using matrices $\{X^{(i)}\}_{i=1}^{\ell}$, vectors $\{\mathbf{x}_i\}_{i=1}^{\ell}$ and pointers $\{d_j\}_{j=1}^{n-k+1}$
22. **if** Deflation should occur **then**
23. Set $m_c = m_c - 1$.
24. **if** $m_c = 0$ **then**
25. STOP
26. **end if**
27. **else**
28. Set $X^{(i)}(1 : k, n + 1) \coloneqq \mathbf{x}_i$
29. Set $\mathbf{a}_{n+1} \coloneqq \mathbf{c} \otimes \hat{\mathbf{q}}_{n+1} + (S \otimes I_{n_0}) \left(\widehat{A}_n(:, B_p)\, \mathbf{g}_{n-m_1+1} + \mathbf{a}_{n-m_c+1}\right)$
30. Set $\widehat{H}(1 : n, n - m_1 + 1) \coloneqq \texttt{tmp}$ and $\widehat{H}(n + 1, n - m_1 + 1) \coloneqq 1$
31. Set $d_{n-k+1} = n + 1$, $B_{p_{n-m_1+1}} \coloneqq n - m_c + 1$ and $n \coloneqq n + 1$
32. **end if**
33. **end if**
34. **end while**

3.5 Applications to Model Order Reduction

The Structure Exploiting Arnoldi algorithm for matrices in case one form enables moment-matching reduced-order modeling that preserves the structure of higher-order linear dynamical

systems. The following is a model reduction technique relying on the specialized (band) SEA algorithms developed Sections 3.3 and 3.4:

Algorithm 13 (Band) SEA-Based Dimension Reduction for Higher-Order Systems

Input: A general higher-order system (2.9) given by

$$P_\ell \frac{d^\ell}{dt^\ell}\mathbf{z}(t) + P_{\ell-1}\frac{d^{\ell-1}}{dt^{\ell-1}}\mathbf{z}(t) + \cdots + P_1\frac{d}{dt}\mathbf{z}(t) + P_0\mathbf{z}(t) = B\mathbf{u}(t),$$
$$y(t) = D\mathbf{u}(t) + L_{\ell-1}\frac{d^{\ell-1}}{dt^{\ell-1}}\mathbf{z}(t) + \cdots + L_1\frac{d}{dt}\mathbf{z}(t) + L_0\mathbf{z}(t),$$

Output: An equivalent reduced higher-order system

$$\widetilde{P}_\ell \frac{d^\ell}{dt^\ell}\mathbf{z}(t) + \widetilde{P}_{\ell-1}\frac{d^{\ell-1}}{dt^{\ell-1}}\mathbf{z}(t) + \cdots + \widetilde{P}_1\frac{d}{dt}\mathbf{z}(t) + \widetilde{P}_0\mathbf{z}(t) = \widetilde{B}\mathbf{u}(t),$$
$$y(t) = D\mathbf{u}(t) + \widetilde{L}_{\ell-1}\frac{d^{\ell-1}}{dt^{\ell-1}}\mathbf{z}(t) + \cdots + \widetilde{L}_1\frac{d}{dt}\mathbf{z}(t) + \widetilde{L}_0\mathbf{z}(t),$$

1. Select $s_0 \in \mathbb{C}$ s.t. $\widehat{P} = P(s_0) = \sum_{i=0}^{\ell} s_0^i P_i$ is invertible and choose reduction dimension n.
2. Run $n-1$ steps of the (Band) SEA algorithm with

$$M^{(i)} = \widehat{P}^{-1}\left(\sum_{j=0}^{\ell-i} P_{i+j}\right), \quad R = \hat{P}^{-1}B, \quad \mathbf{c} = \begin{bmatrix} 1 \\ s_0 \\ s_0^2 \\ \vdots \\ s_0^{\ell-1} \end{bmatrix}, \quad S = -\begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ 1 & 0 & 0 & \ddots & \vdots \\ s_0 & 1 & 0 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & 0 \\ s_0^{\ell-2} & \cdots & s_0 & 1 & 0 \end{bmatrix}$$

   for $i = 1, 2, ..., \ell$ and produce and orthonormal basis Q_k for the space S_n.
3. Obtain the reduced order model for the original system by projection:

$$\widetilde{P}_i = Q_k^T P_i Q_k, \qquad \widetilde{B} = Q_k^T B, \qquad \widetilde{L}_j = L_j Q_k, \tag{3.26}$$

   for $i = 1, 2, ..., \ell$ and $j = 0, 1, ..., \ell - 1$.

The matrices $M^{(i)} \in \mathbb{C}^{n_0 \times n_0}$ are never formed explicitly in the implementation of (band) SEA-based reduction. Instead, a sparse LU-factorization

$$\widehat{P} = P(s_0) = P^T L\, U\, Q^T$$

is calculated where P and Q are permutation matrices, L is lower-triangle with ones on the main diagonal and U is upper-triangular with nonzero diagonal entries [**Dav06**]. Then, matrix-vector multiplication of

$$F = \begin{bmatrix} M^{(1)} & M^{(2)} & \cdots & M^{\ell} \end{bmatrix} = \widehat{P}^{-1}\begin{bmatrix} \sum_{j=0}^{\ell-1} P_{1+j} & \sum_{j=0}^{\ell-2} P_{2+j} & \cdots & P_\ell \end{bmatrix}$$

with a single vector $\hat{\mathbf{v}}$ proceeds using the equation

$$F\,\hat{\mathbf{v}} = Q\left(U^{-1}\left(L^{-1}\left(P\left(\begin{bmatrix}\sum_{j=0}^{\ell-1} P_{1+j} & \sum_{j=0}^{\ell-2} P_{2+j} & \cdots & P_\ell\end{bmatrix}\hat{\mathbf{v}}\right)\right)\right)\right).$$

For many applications with sparse data matrices $\{P_i\}_{i=1}^{\ell}$, this equivalent method of multiplying by F is quite fast due to sparse matrix-vector multiplication and sparse forward and backward substitution.

Suppose that specialized (band) SEA algorithm runs for $n-1$ iterations and the output matrix Q_k is an orthonormal basis for S_n. If

$$\mathcal{V}_n = \begin{bmatrix} Q_k & 0 & \cdots & 0 \\ 0 & Q_k & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & Q_k \end{bmatrix} \in \mathbb{C}^{N\times n_1}$$

with $n_1 = k\cdot\ell$, then $K(\mathcal{M},\mathcal{R},n) \subseteq$ range $(\mathcal{V}_n)$, where matrices $\mathcal{M}$ and $\mathcal{R}$ (1.5) arise from the Taylor series expansion of the transfer function coming from the linearization of the original higher-order system. Theorem 2.4.1 guarantees that reduced-order models formed via projection onto $\mathcal{V}_n$ (2.30) results in a Padé-type model of the original system. Because the columns of Q_k are orthonormal, the structure of the linearization matrices is preserved in the sense that

$$\mathcal{V}_n^H\mathcal{E}\mathcal{V}_n = \begin{bmatrix} \mathrm{I}_k & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \mathrm{I}_k & 0 \\ 0 & \cdots & 0 & \widetilde{P}_\ell \end{bmatrix}, \qquad \mathcal{V}_n^H\mathcal{A}\mathcal{V}_n = -\begin{bmatrix} 0 & -\mathrm{I}_k & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & -\mathrm{I}_k \\ \widetilde{P}_0 & \widetilde{P}_1 & \cdots & \widetilde{P}_{\ell-1} \end{bmatrix},$$

while

$$\mathcal{V}_n^H\mathcal{B} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \widetilde{B} \end{bmatrix}, \qquad \mathcal{L}\mathcal{V}_n = \begin{bmatrix} \widetilde{L}_0 & \cdots & \widetilde{L}_{\ell-1} \end{bmatrix},$$

where $\widetilde{B}, \widetilde{P}_i, \widetilde{L}_j$ are defined above (3.26). Some tedious algebra demonstrates

$$\begin{aligned} H_n(s) &= \mathcal{L}_n(s_0\mathcal{E}_n - \mathcal{A}_n)^{-1}\mathcal{B}_n \\ &= \left(\widetilde{L}_0 + s\widetilde{L}_1 + \cdots + s^{\ell-1}\widetilde{L}_{\ell-1}\right)\left(\widetilde{P}_0 + s\widetilde{P}_1 + \cdots + s^{\ell}\widetilde{P}_{\ell}\right)\widetilde{B}. \end{aligned}$$

where $\mathcal{E}_n, \mathcal{A}_n, \mathcal{B}_n, \mathcal{L}_n$ are given by projections onto $\mathcal{V}_n$ (2.30). The cost of forming the projections $\widetilde{B} = Q_k^H B$, $\widetilde{P}_i = Q_k^H P_i Q_k$ and $\widetilde{L}_j = L_j Q_k$ is overhead compared with executing model reduction using the (band) Arnoldi algorithm. However, the (band) Arnoldi algorithm does not produce a reduced-order model that can be stated as a higher-order system. Further, the (band) Arnoldi algorithm manipulates vectors of size $N = n_0 \cdot \ell$ while the SEA algorithm works with matrices of size n_0.

Numerical Examples

This subsection presents numerical results comparing the accuracy of three different model order reduction techniques including SEA-based, SOAR-based, and Arnoldi-based model order reduction. This work establishes the basic properties of the new, more general SEA-based technique. Numerical implementations of the algorithms used for each experiment are based on the pseudo code presented in this thesis. A total of four examples are presented including three experiments on single-input, single-output higher-order systems and one experiment on a multiple-input, multiple-output higher-order system.

Example 3.5.1. The first example illustrates the numerical results for the three reduced-order modeling methods applied to a linear-drive multimode resonator structure [**JCP98**]. The original system is a nonsymmetric second-order system ($\ell = 2$) with state-space dimension $n_0 = 63$ and a single input vector ($m = 1$). The mass matrix P_2 and the damping matrix P_1 are both singular indicating that the original higher-order system is a descriptor system. The stiffness matrix P_0 has condition number at $\mathcal{O}(10^{15})$ as measured in the 1-norm. The left column of Table 3.5.1 presents Bode plots of the frequency response for the original system and the corresponding Bode plots for two different reduced-order models with the reduction dimension $n = 13$. The reduction techniques

use the expansion point $s_0 = 2\pi \times 10^5$. The corresponding relative errors

$$\frac{|H(j\omega) - H_n(j\omega)|}{|H(j\omega)|}$$

with $j = \sqrt{-1}$ are shown in the right column of Table 3.5.1. The results indicate that SEA-based model reduction is an alternative, more general method than SOAR-based reduced-order modeling, both of which are superior to the Arnoldi-based method.

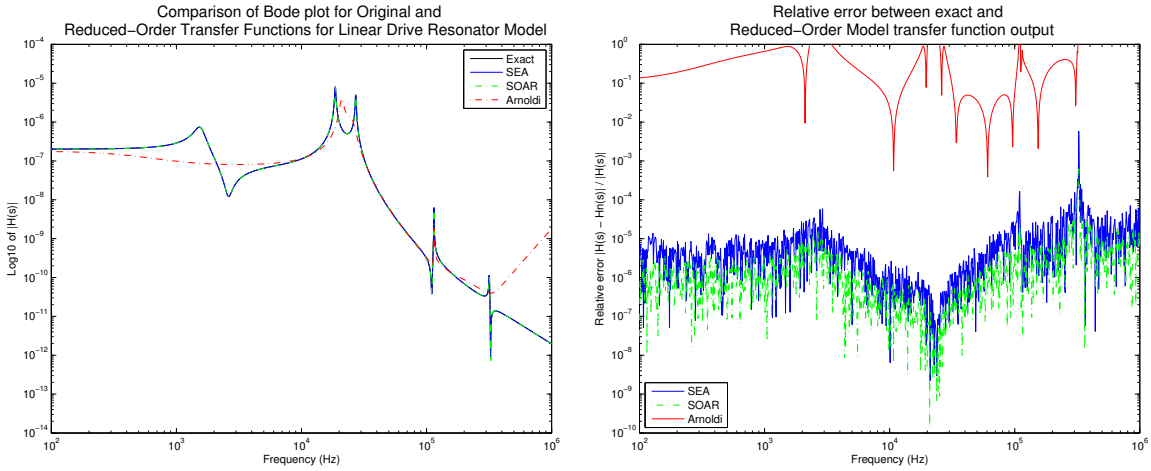

TABLE 3.5.1. Linear-drive multimode resonator structure: The left entry of this table presents Bode plots of Linear-drive resonator for original system (black), SEA-based reduced-order model (blue), SOAR-based reduced-order model (green) and Arnoldi-based reduced-order model (red) for reduced dimension of $n = 13$. The right entry of this table includes the relative error for these reduced models.

Example 3.5.2. This example illustrates numerical results for reduced-order modeling of a torsional micromirror [**CP02**]. The original single-input ($m = 1$), second-order system ($\ell = 2$) is produced using a lumped finite-element analysis with state-space dimension $n_0 = 846$. Symmetric mass and damping matrices P_2 and P_1 have small elements with 1-norms $\|P_2\|_1 = \mathcal{O}(10^{-8})$ and $\|P_1\|_1 = \mathcal{O}(10^{-6})$. The nonsymmetrix stiffness matrix P_0, on the other hand, has relatively large elements with $\|P_0\|_1 = \mathcal{O}(10^9)$. All data matrices in this example are ill-conditioned with 1-norm condition number on the order of 10^{18}. Bode plots of the frequency response for the original system and corresponding reduced-order models are given in the left column of Table 3.5.2. The reduction dimension used for these models is $n = 28$ and the selected expansion point is $s_0 = 2\pi \times 10^4$. The corresponding relative errors between are shown in the right column of Table 3.5.2.

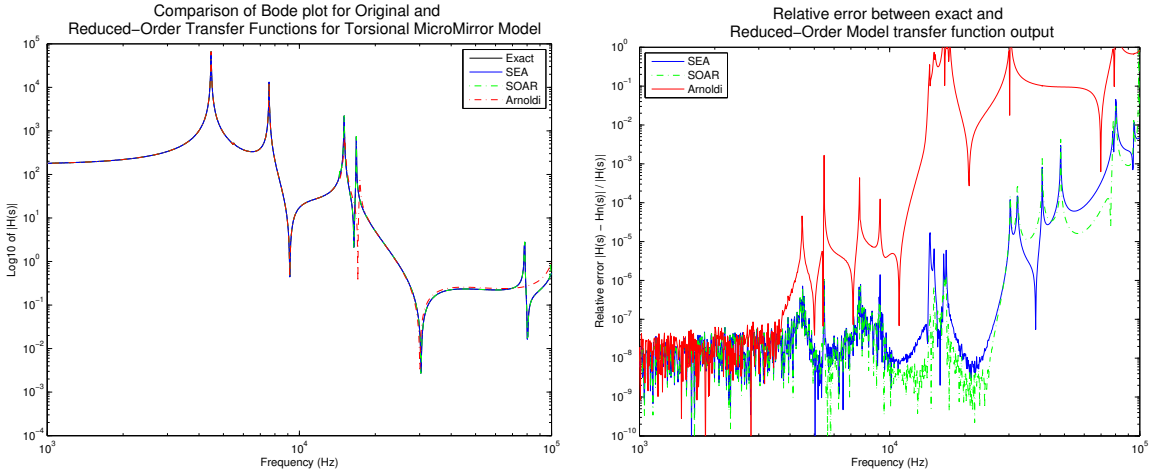

TABLE 3.5.2. Torsional micromirror: The left entry of this table presents Bode plots of torsional micromirror for original system (black), SEA-based reduced-order model (blue), SOAR-based reduced-order model (green), and Arnoldi-based reduced-order model(red) for reduced dimension $n = 28$. The right entry of this table includes the relative error for these reduced models.

Example 3.5.3. In the third example, model order reduction techniques are applied to a single-input ($m = 1$), second-order system ($\ell = 2$) modeling the Los Angeles University Hospital [**ASG01**].

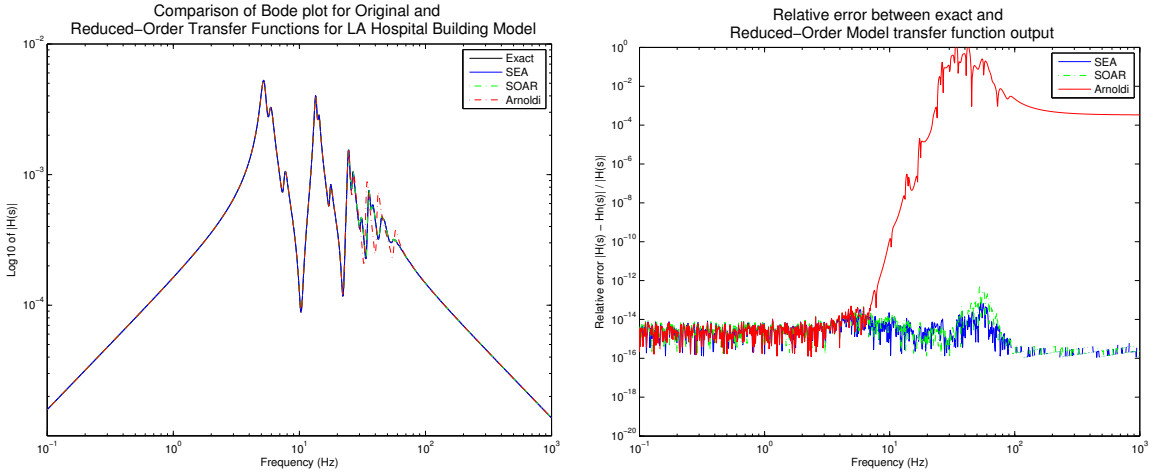

TABLE 3.5.3. Los Angeles (LA) University hospital building model: The left entry of this table presents Bode plots of the LA Hospital Building model for original system (black), SEA-based reduced-order model (blue), SOAR-based reduced-order model (green), and Arnoldi-based reduced-order model(red) for reduced dimension $n = 24$. The right entry of this table includes the relative error for these reduced models.

The original system has state-space dimension $n_0 = 24$, representing 8 floors, with three degrees of freedom on each floor including displacements in both the x and y directions as well as rotation. Matrix P_2 is invertible in this case, leading to a system of second-order differential equations. By multiplying the entire system by P_2^{-1}, an equivalent system arises with I_{n_0} being the scaling matrix for second-order data. Motion in the first coordinate is measured by setting $L_0 = 0$ and $L_1 \neq 0$. The reduction dimension used for these models is $n = 24$ and the selected expansion point is $s_0 = 1$. The left column of Table 3.5.3 illustrates Bode plots of the frequency response for the original system and corresponding reduced-order models while the relative errors are shown in the right column of the same table.

Example 3.5.4. Numerical results for model order reduction applied to the multiple-input ($m = 3$), multiple-output ($p = 3$) second-order ($\ell = 2$) structural model of the Russian service module of the International Space Station are illustrated in the final example of this section [**CV02**]. The original system has state-space dimension $n_0 = 135$ and comes from a system of second-order differential equations, with P_2 being nonsingular. These numerical simulations rely on expansion point $s_0 = 3500$ and reduction dimension $n = 135$. Table 3.5.4 gives the Bode plots corresponding to each component of the 3×3 transfer function for the original system (black), the band SEA-based reduced model (blue) and the band Arnoldi-based reduced model (red). All data shown in the table below result from a single call to the specialized band SEA algorithm and a single call to band Arnoldi algorithm. The relative errors for each component of the transfer functions for both the band SEA-based reduced-order model and the band Arnoldi-based reduced-order model are included below.

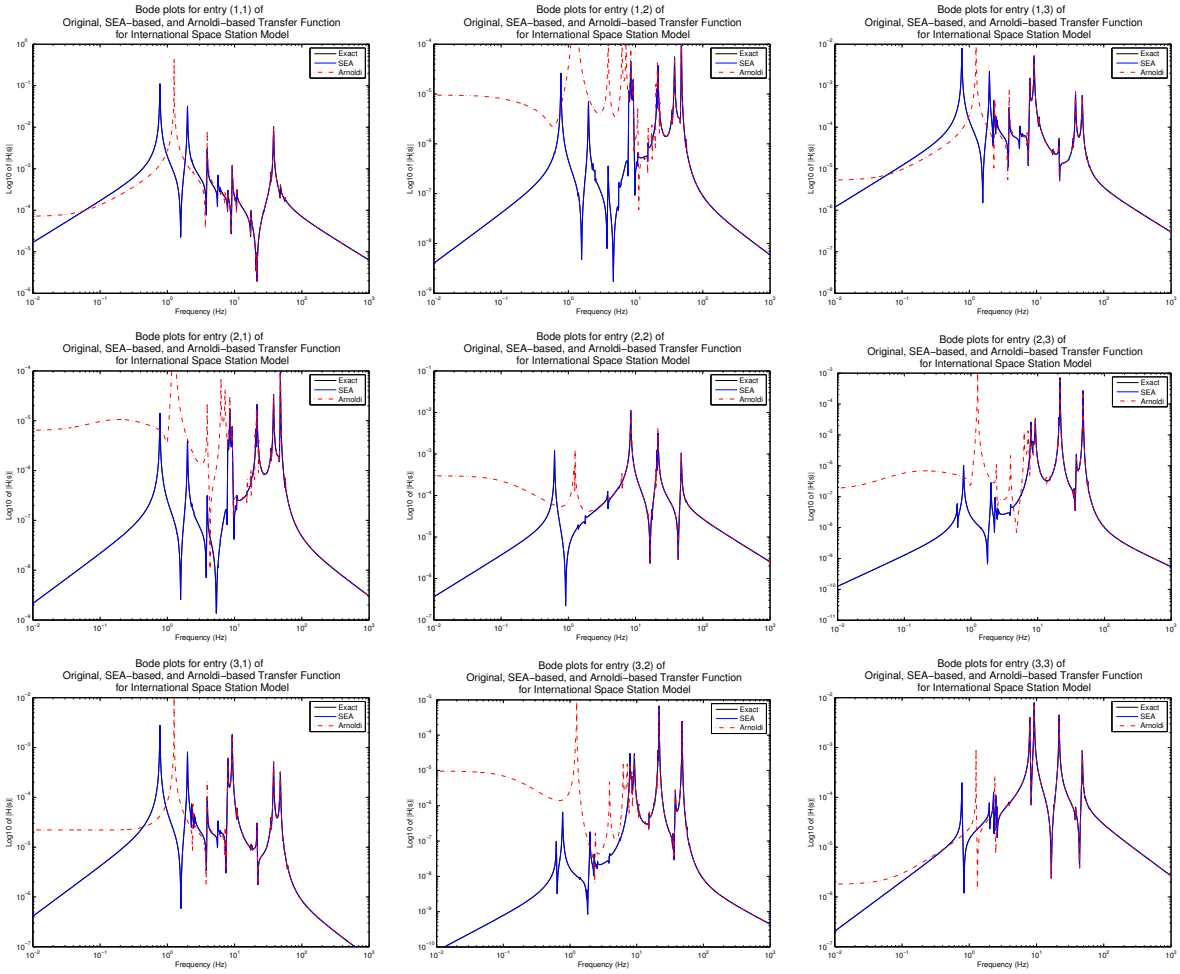

TABLE 3.5.4. International Space Station (ISS) model: The (i, j)th entry of this table presents Bode plots of the (i, j)th entry of the transfer function generated by the original system (black), band SEA-based reduced-order model (blue), and band Arnoldi-based reduced-order model (red) for reduction dimension $n = 135$.

There is a substantial improvement offered by the specialized band SEA algorithm because $n = 135$ is chosen to be equal to the state-space dimension of the higher-order system. For the numerical tests presented here, the expansion point $s_0 = 3500$ produced $k = 135$ basis vectors for S_n, increasing the numerical accuracy of the reduced model. The corresponding relative errors are shown in Table 3.5.4. The relative error between entry (i, j) of the original and reduced order transfer functions is given in entry (i, j) below.

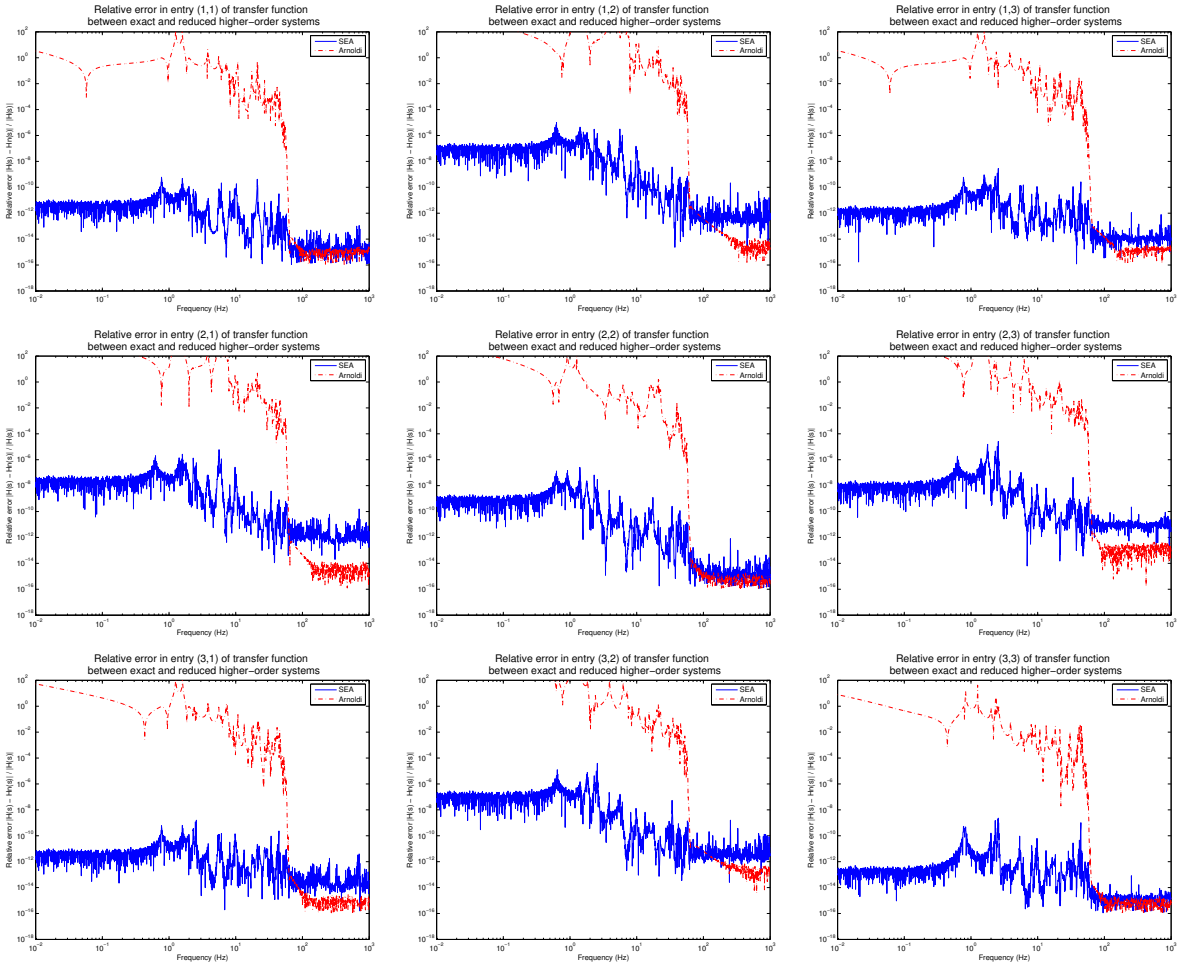

TABLE 3.5.5. International Space Station model: The (i,j)th entry of this table presents the relative error between entry (i,j) of transfer functions from the original system, band SEA-based MOR (blue) and band Arnoldi-based MOR (red) corresponding to each entry in Table 3.4 above.

3.6 Conclusions

This chapter introduces the SEA algorithm for efficient model order reduction of multiple-input higher-order systems. To create this reduction technique, Chapter 3 begins with an analysis of the structure of block Krylov subspaces induced by matrices in case one form from the perspective of generating orthonormal basis of S_n (3.6). This study included an equivalent factorization of the block Krylov matrix induced by $\mathcal{M}$ and $\mathcal{R}$ assuming no specific knowledge of the elimination matrices $\{E_i\}_{i=1}^{k_0}$ permitting the exact deflation patterns for $K_N(\mathcal{M},\mathcal{R})$ to be ascertained by working with this equivalent full-rank factorization.

The corresponding generic single-input and generic band versions of the SEA algorithm suggest a template for structure exploiting algorithms for matrices in case one form. As is shown in Theorem 3.2.2 and Proposition 3.2.3, the most important aspects of these algorithms is the maintenance of the (band) Hessenberg matrix relations. However, as the general template evinces, the (band) Hessenberg matrix relating the columns of $\widehat{Q}_n$ to basis vectors $\widehat{A}_n$ of $K(\mathcal{M}, \mathcal{R}, n)$ (3.17) need not be the same as the (band) Hessenberg update that relates the basis matrix $\widehat{A}_n$ to matrix $\mathcal{M}$ (3.18).

One possible alteration to the generic Structure Exploiting Arnoldi algorithm is to rely on the SOAR-type Hessenberg inversion trick to speed up the check for deflations. These adaptations come in both single-input and band versions and improve the performance of the SEA algorithms applied to model order reduction of large-scale higher-order linear dynamical systems. This permits a more efficient treatment of Structure Exploiting Arnoldi-based model order reduction. Table 3.6 gives a synopsis of the contributions made by this work.

TABLE 3.6.1. Model Reduction Algorithm Comparison for Higher-Order Systems

	SEA	**SOAR**	**Arnoldi**
Order of input system	$\ell \geq 2$	2	1
Band version available	✓	×	✓
Higher-order realization of reduced system	✓	✓	×
Padé-type moment-matching property	✓	✓	✓
Orthogonalization using vectors of size	n_0	n_0	$N = n_0 \cdot \ell$
Exact deflation executed using vectors of size	$\ll n_0$	n_0	N

CHAPTER 4

The SEA Algorithm for Matrices in Case Two Form

The Structure Exploiting Arnoldi algorithm developed in this chapter provides an improved dimension reduction technique for linearized systems of first-order integro-DAEs. Such systems arise in VLSI interconnect analysis where the need for efficient dimension reduction methods is omnipresent, at least as long as Moore's law continues to hold. Most Krylov subspace-based moment-matching reduction techniques used in this popular application area make trade-offs between accuracy, the preservation of important system properties, and efficiency in numerical computations. As discussed in the introduction, a recent trend for dimension reduction of RCL circuits has been to develop reduced models that maintain passivity as a key system property. The current state-of-the-art technique for this purpose is the SPRIM algorithm [**Fre04**], [**Fre11**] with which accurate, provably passive reduced-order models can be generated. However, this algorithm uses a series of separate mechanisms to produce reduced-order models, sacrificing some efficiency in numerical computations.

The Structure Exploiting Arnoldi algorithm presented in this chapter generates provably passive and reciprocal macromodels of multiport RCL circuits matching the same number of leading moments as the SPRIM algorithm at a fraction of the computational cost. By relying on structure results for the factorization of block Krylov subspaces associated with linearized systems of first-order integro-DAEs, SEA-based reduced-order models enjoy the same theoretic properties as SPRIM-based models but can be generated using fewer computational steps.

The organization of this chapter is similar to that of Chapter 3. Section 4.1 develops general structure results for block Krylov subspaces associated with linearized systems of first-order integro-DAEs including an improved method of executing exact deflation without explicitly manipulating columns of the block Krylov matrix. Section 4.2 describes a corresponding generic single-input SEA algorithm for general matrices in case two form. Section 4.3 improves the computational performance of this single-input SEA algorithm for applications in VLSI interconnect analysis and

Section 4.4 presents band versions of the SEA algorithms for multiple input vectors. The fifth section presents the model order reduction technique based on these algorithms including a review of RCL circuit equations, a discussion of the properties of SEA-based dimension reduction and numerical results demonstrating the effectiveness of this method. The final section of this chapter summarizes the advantages of SEA-based model order reduction for systems of first-order integro-DAEs.

4.1 Exploring Matrices in Case Two Form

Matrices in case two form are a generic class of structured matrices used to describe matrices $\mathcal{M}$ and $\mathcal{R}$ associated with the transfer function $H(s)$ (1.4) arising from linearized first-order integro-DAEs [**Fre05**]. Block Krylov subspaces induced by matrices $\mathcal{M}$ and $\mathcal{R}$ have structure that can be used to improve existing dimension reduction techniques.

Definition 4.1.1 (Freund [**Fre05**])**.** Let n_0 and ℓ be natural numbers. Let $\{n_i\}_{i=1}^{\ell}$ be a sequence of positive integers and assume $C^{(i)} \in \mathbb{C}^{n_i \times n_0}$ for all $i = 1, 2, ..., \ell$. Given $N = n_1 + n_2 + \cdots + n_\ell$, let $F = \begin{bmatrix} M^{(1)} & M^{(2)} & \cdots & M^{(\ell)} \end{bmatrix} \in \mathbb{C}^{n_0 \times N}$ where $M^{(i)} \in \mathbb{C}^{n_0 \times n_i}$ for $i = 1, ..., \ell$. Assume $s_i \in \mathbb{C}$ for all $i = 1, 2, ..., \ell$ and let $R \in \mathbb{C}^{n_0 \times m}$. Matrices $\mathcal{M}$ and $\mathcal{R}$ are said to be in **case two form** if, and only if,

$$\begin{aligned} \mathcal{M} &= \begin{bmatrix} C^{(1)} \\ C^{(2)} \\ \vdots \\ C^{(\ell)} \end{bmatrix} \begin{bmatrix} M^{(1)} & \cdots & M^{(\ell)} \end{bmatrix} + \begin{bmatrix} s_1 I_{n_1} & 0 & \cdots & 0 \\ 0 & s_2 I_{n_2} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & s_\ell I_{n_\ell} \end{bmatrix} \in \mathbb{C}^{N \times N}, \\ \mathcal{R} &= \begin{bmatrix} C^{(1)} \\ C^{(2)} \\ \vdots \\ C^{(\ell)} \end{bmatrix} \; R \in \mathbb{C}^{N \times m}. \end{aligned} \tag{4.1}$$

Moments of the Taylor series expansion of transfer function $H(s)$ about $s_0 \in \mathbb{C}$ (1.7) associated with the linearization of a system of first-order integro-DAEs (2.12) can be expressed in terms of matrices in case two form.

Proposition 4.1.2 (Freund [**Fre05**])**.** Transform any system of first-order integro-DAEs (2.12), with $P_{-1} = F_1 G F_2^H$ (2.15), into its equivalent first-order system (2.19) using the appropriate linearization matrices (2.24), (2.26). Suppose the matrix pencil $Q(s)$ (2.14) is regular and choose nonzero $s_0 \in \mathbb{C}$ such that $Q(s_0) = Q_0$ is nonsingular. The matrices used for moment-matching block Krylov subspace-based dimension reduction techniques are given by

$$\begin{aligned} \mathcal{M} &= (s_0\mathcal{E} - \mathcal{A})^{-1}\mathcal{E} = \begin{bmatrix} I_{n_0} \\ \frac{1}{s_0} F_2^H \end{bmatrix} \begin{bmatrix} Q_0^{-1}P_1 & -\frac{1}{s_0} Q_0^{-1} F_1 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & \frac{1}{s_0} I_{n_2} \end{bmatrix}, \\ \mathcal{R} &= (s_0\mathcal{E} - \mathcal{A})^{-1}\mathcal{B} = \begin{bmatrix} I_{n_0} \\ \frac{1}{s_0} F_2^H \end{bmatrix} Q_0^{-1} B. \end{aligned} \tag{4.2}$$

Proposition 4.1.3 (Freund [**Fre05**])**.** Suppose a system of first-order integro-DAEs (2.12), with $P_{-1} = F_1 G^{-1} F_2^H$ (2.16), is transformed into its equivalent first-order equivalent (2.19) via the appropriate linearization matrices (2.27), (2.28). If $s_0 \in \mathbb{C}$ is chosen such that $Q(s_0) = Q_0$ is invertible, then the matrices used for moment-matching block Krylov subspace-based dimension reduction techniques are given by

$$\begin{aligned} \mathcal{M} &= \begin{bmatrix} I_{n_0} \\ \frac{1}{s_0} G^{-1} F_2^H \end{bmatrix} \begin{bmatrix} Q_0^{-1}P_1 & -\frac{1}{s_0} Q_0^{-1} F_1 G \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & \frac{1}{s_0} I_{n_2} \end{bmatrix}, \\ \mathcal{R} &= \begin{bmatrix} I_{n_0} \\ \frac{1}{s_0} G^{-1} F_2^H \end{bmatrix} Q_0^{-1} B. \end{aligned} \tag{4.3}$$

Block Krylov subspaces induced by matrices $\mathcal{M}$ and $\mathcal{R}$ in case two form contain multiple copies of the same underlying subspace. Assume matrices $\mathcal{M} \in \mathbb{C}^{N\times N}$ and $\mathcal{R} \in \mathbb{C}^{N\times m}$ are in case two form for the remainder of this chapter.

Proposition 4.1.4 (Freund [**Fre05**])**.** Let $\mathcal{V}$ be any basis of the block Krylov subspaces induced by $\mathcal{M}$ and $\mathcal{R}$. Then, $\mathcal{V}$ can be factored as

$$\mathcal{V} = \begin{bmatrix} C^{(1)}WU^{(1)} \\ C^{(2)}WU^{(2)} \\ \vdots \\ C^{(\ell)}WU^{(\ell)} \end{bmatrix} \tag{4.4}$$

where $W \in \mathbb{C}^{n_0 \times N_0}$ and $U^{(i)} \in \mathbb{C}^{N_0 \times N_0}$ is upper-triangular with nonzero diagonal elements for each $i = 1, 2, ...\ell$.

A compact formulation of the matrix factor W in the case two factorization result (4.4) arises by relating the individual blocks of the deflated block Krylov matrix induced by $\mathcal{M}$ and $\mathcal{R}$ (2.3) to the corresponding blocks of W.

Proposition 4.1.5. Let $m_0 = m$ and suppose that $E_j \in \mathbb{R}^{m_{j-1} \times m_j}$ is the appropriate elimination matrix from the exact deflation process (2.1.4) for $j = 1, 2, ..., k_0$. Initialize the coupled recursion

$$W_1 = RE_1 \in \mathbb{C}^{n_0 \times m_1}, \quad A_1 = \begin{bmatrix} C^{(1)} \\ \vdots \\ C^{(\ell)} \end{bmatrix} W_k. \tag{4.5}$$

Define the $(k+1)$st coupled recursion pair as

$$\begin{aligned} W_{k+1} &= FA_kE_{k+1} \in \mathbb{C}^{n_0 \times m_{k+1}}, \\ A_{k+1} &= \begin{bmatrix} C^{(1)} \\ \vdots \\ C^{(\ell)} \end{bmatrix} W_{k+1} + \begin{bmatrix} s_1 I_{n_1} & & \\ & \ddots & \\ & & s_\ell I_{n_\ell} \end{bmatrix} A_kE_{k+1}, \end{aligned} \tag{4.6}$$

for $k = 1, 2, ..., k_0 - 1$. The kth block of the deflated block Krylov matrix induced by $\mathcal{M}$ and $\mathcal{R}$ (2.1.4) can be represented in the form

$$\mathcal{M}^{k-1}\mathcal{R}_k = A_k, \tag{4.7}$$

for $k = 1, 2, ..., k_0$

Proof: This result follows by induction. For the initial case, $A_1 = \mathcal{R}_1$ by definition. Assuming $A_k = \mathcal{M}^{k-1}\mathcal{R}_k$, consider

$$A_{k+1} = \begin{bmatrix} C^{(1)} \\ \vdots \\ C^{(\ell)} \end{bmatrix} W_{k+1} + \begin{bmatrix} s_1 I_{n_1} & & \\ & \ddots & \\ & & s_\ell I_{n_\ell} \end{bmatrix} A_k E_{k+1} = \mathcal{M}^k \mathcal{R}_{k+1}.$$

This is exactly what was to be shown. □

4.1.1 Alternative Proof of the Factorization Result

Proposition 4.1.4 can be proved using the recursive formulation of W_k (4.6).

Proof: Partition of the matrices W and $U^{(i)}$ just as in Subsection 3.1.1, for $i = 1, 2, ..., \ell$. The desired relation (4.4) holds true if, and only if,

$$\mathcal{M}^{k-1}\mathcal{R}_k = \begin{bmatrix} C^{(1)} & 0 & \cdots & 0 \\ 0 & C^{(2)} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & C^{(\ell)} \end{bmatrix} \left(I_\ell \otimes \begin{bmatrix} W_1 & W_2 & \cdots & W_k \end{bmatrix} \right) \begin{bmatrix} \begin{bmatrix} U_{1k}^{(1)} \\ \vdots \\ U_{kk}^{(1)} \end{bmatrix} \\ \vdots \\ \begin{bmatrix} U_{1k}^{(\ell)} \\ \vdots \\ U_{kk}^{(\ell)} \end{bmatrix} \end{bmatrix}, \tag{4.8}$$

for $k = 1, 2, ..., k_0$. Assuming W_k is defined as in Proposition 4.1.5 for $k = 1, ..., k_0$, the kth block column of the deflated block Krylov matrix induced by $\mathcal{M}$ and $\mathcal{R}$ is given by $\mathcal{M}^{k-1}\mathcal{R}_k = A_k$. Each subblock of $U^{(i)}$ can be given recursively as

$$\begin{bmatrix} U_{kk}^{(1)} \\ \vdots \\ U_{kk}^{(\ell)} \end{bmatrix} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \otimes I_{m_k}, \qquad \begin{bmatrix} U_{1k}^{(i)} \\ \vdots \\ U_{k-1,k}^{(i)} \end{bmatrix} = \begin{bmatrix} s_1 U_{1,k-1}^{(1)} \\ \vdots \\ s_\ell U_{k-1,k-1}^{(\ell)} \end{bmatrix} E_k, \tag{4.9}$$

for $k = 1, 2, ..., k_0$, $j = 1, 2, ..., k-1$ and $i = 1, 2, ..., \ell$. Induction on k demonstrates that the block partitions of W and $U^{(i)}$ satisfy the structure relation for the basis matrix of the block Krylov subspaces (4.4).

For the base case of this inductive argument ($k = 1$),

$$\mathcal{R}_1 = \begin{bmatrix} C^{(1)} \\ \vdots \\ C^{(\ell)} \end{bmatrix} RE_1 = \begin{bmatrix} C^{(1)} \\ \vdots \\ C^{(\ell)} \end{bmatrix} (I_\ell \otimes W_1) \begin{bmatrix} U_{11}^{(1)} \\ \vdots \\ U_{11}^{(\ell)} \end{bmatrix}.$$

Suppose that the structure relation for the basis matrix of the block Krylov subspaces induced by matrices in case two form (4.4) holds for $1 < k < k_0$. Multiply this structure relation (4.4) from the left by the matrix $\mathcal{M}$ and from the right by E_{k+1} to produce

$$\mathcal{M}^k \mathcal{R}_{k+1} = \begin{bmatrix} C^{(1)} \\ \vdots \\ C^{(\ell)} \end{bmatrix} FA_k E_{k+1} + \begin{bmatrix} s_1 I_{n_1} & & \\ & \ddots & \\ & & s_\ell I_{n_\ell} \end{bmatrix} A_k E_{k+1}. \tag{4.10}$$

By the equivalent form of the the induction hypothesis (4.8) and matrix algebra, the matrix $\mathcal{M}^k \mathcal{R}_{k+1}$ is given by

$$\begin{bmatrix} C^{(1)} \\ \vdots \\ C^{(\ell)} \end{bmatrix} W_{k+1} + \begin{bmatrix} C^{(1)} & & \\ & \ddots & \\ & & C^{(\ell)} \end{bmatrix} \left(I_\ell \otimes \begin{bmatrix} W_1 & \cdots & W_k \end{bmatrix} \right) \begin{bmatrix} s_1 \begin{bmatrix} U_{1,k+1}^{(1)} \\ \vdots \\ U_{k,k+1}^{(1)} \end{bmatrix} \\ \vdots \\ s_\ell \begin{bmatrix} U_{1,k+1}^{(\ell)} \\ \vdots \\ U_{k,k+1}^{(\ell)} \end{bmatrix} \end{bmatrix}. \tag{4.11}$$

The equivalent relation (4.8) holds for $k + 1$ by the definitions of $U_{k+1,k+1}^{(i)}$ for $i = 1, 2, ..., \ell$, completing this proof. □

The column space of W will play an important role throughout the rest of Chapter 4 and it will be convenient to refer to this space succinctly. To this end, let $\mathbf{w}_i \in \mathbb{C}^{n_0}$ be the ith column of the matrix $W \in \mathbb{C}^{n_0 \times N_0}$ for $i = 1, 2, ..., N_0$. Let

$$S_n = \langle \mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_n \rangle \subseteq \mathbb{C}^{n_0}, \tag{4.12}$$

for $n = 1, 2, ..., N_0$.

4.1.2 Full-Rank Factorization

Any basis matrix $\mathcal{V}$ of the block Krylov subspace induced by $\mathcal{M}$ and $\mathcal{R}$ can be written using smaller factors $C^{(i)}$, W and $U^{(i)}$ for $i = 1, 2, ..., \ell$ by Proposition 4.1.4. Suppose that the column rank of the matrix product $C^{(i)}W$ is $w_0^{(i)} \leq \min\{n_i, n_0\}$, for $i = 1, 2, ..., \ell$. A full-rank structure factorization of $\mathcal{V}$ can be written in terms of orthonormal basis for the column space of each $C^{(i)}W$, for $i = 1, 2, ..., \ell$.

Corollary 4.1.6. Let $\mathcal{V}$ be any basis of the block Krylov subspaces induced by $\mathcal{M}$ and $\mathcal{R}$. Then,

$$\mathcal{V} = \begin{bmatrix} V^{(1)} & 0 & \cdots & 0 \\ 0 & V^{(2)} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & V^{(\ell)} \end{bmatrix} \begin{bmatrix} X^{(1)} \\ X^{(2)} \\ \vdots \\ X^{(\ell)} \end{bmatrix}, \tag{4.13}$$

where the matrix $V^{(i)} \in \mathbb{C}^{n_i \times w_0^{(i)}}$ has orthonormal columns and $X^{(i)} \in \mathbb{C}^{w_0^{(i)} \times N_0}$ is in row echelon form for each $i = 1, 2, ..., \ell$.

Proof: Let the basis matrix $\mathcal{V}$ be factored using matrices $C^{(i)} \in \mathbb{C}^{n_i \times n_0}$, $W \in \mathbb{C}^{n_0 \times N_0}$ and $U^{(i)} \in \mathbb{C}^{N_0 \times N_0}$ for $i = 1, 2, ..., \ell$ (4.4). Suppose that the column rank of each matrix product $C^{(i)}W$ is given by $w_0^{(i)}$ for $i = 1, 2, ..., \ell$. Then, $C^{(i)}W = V^{(i)}\widehat{R}^{(i)}$, where the columns of $V^{(i)} \in \mathbb{C}^{n_i \times w_0^{(i)}}$ form an orthonormal basis of the column space of $C^{(i)}W$ and $\widehat{R}^{(i)} \in \mathbb{C}^{w_0^{(i)} \times N_0}$ in row echelon form encodes the linear dependence relationships between $V^{(i)}$ and $C^{(i)}W$ for $i = 1, 2, ..., \ell$. This proof is completed by setting $X^{(i)} = \widehat{R}^{(i)}U^{(i)}$ for $i = 1, 2, ..., \ell$. □

A major difference between the case one factorization (3.5) and the case two factorization (4.4) relates to the matrices $C^{(i)}$. For matrices in case one form, each entry of vector $\mathbf{c} \in \mathbb{C}^{\ell}$ is nonzero. In contrast, the matrix

$$\begin{bmatrix} C^{(1)} & 0 & \cdots & 0 \\ 0 & C^{(2)} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & C^{(\ell)} \end{bmatrix}$$

will have a nontrivial kernel in general. The following example illustrates that the kernels of matrices $C^{(i)}$ affect the linear dependence relationship between the basis vectors of the block Krylov subspaces induced by $\mathcal{M}$ and $\mathcal{R}$ and the columns of the matrix W.

Example 4.1.7. Let $\ell = 2$, $n_0 = n_1 = 3$, $n_2 = 2$ and $N = n_1 + n_2 = 5$. Suppose

$$C^{(1)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \qquad C^{(2)} = \begin{bmatrix} 1 & 0 & -.5 \\ 0 & 1 & 0 \end{bmatrix}.$$

Let the matrices $M^{(i)}$ and starting vector $\mathbf{r}$ be given by

$$M^{(1)} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix}, \qquad M^{(2)} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 1 & 0 \end{bmatrix}, \qquad \mathbf{r} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}.$$

Let $s_1 = 2$ and $s_2 = 3$. The first three columns of the Krylov matrix induced by $\mathcal{M}$ and $\mathcal{R}$ can be factorized, as suggested by Proposition 4.1.4, as

$$\begin{bmatrix} 1 & 2 & 4 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 2 & 4 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & & & \\ 0 & 1 & 0 & & & \\ 0 & 0 & 0 & & & \\ & & & 1 & 0 & -.5 \\ & & & 0 & 1 & 0 \end{bmatrix} \left(I_2 \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 4 \end{bmatrix} \right) \left[\begin{array}{ccc} 1 & 2 & 4 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \\ \hline 1 & 3 & 9 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{array}\right].$$

The analogous full rank factorization is given by

$$\begin{bmatrix} \mathcal{R} & \mathcal{M}\mathcal{R} & \mathcal{M}^2\mathcal{R} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \left[\begin{array}{ccc} 1 & 2 & 4 \\ \hline 1 & 2 & 4 \end{array}\right].$$

In this example, the grade of the Krylov subspace induced by matrix $\mathcal{M}$ and $\mathcal{R}$ is one while the column rank of matrix factor W is two. An analogous example can never be constructed for matrices in case one form, as Proposition 3.1.10 demonstrates.

4.1.3 Exact Deflation and the Column Space of W

Just as in Chapter 3, these initial factorization results require prior knowledge of both the block grade k_0 of the block Krylov subspace induced by $\mathcal{M}$ and $\mathcal{R}$ as well as the elimination matrices $\{E_j\}_{j=1}^{k_0}$ arising from exact deflation on $K_N(\mathcal{M},\mathcal{R})$. A structure exploiting algorithm that works with matrices in case two form will need to be able to construct these elimination matrices without manipulating columns of $K_N(\mathcal{M},\mathcal{R})$ directly. A factorization analogous to the result presented in Proposition 4.1.4 that requires no knowledge of elimination matrices can be easily formed. Let $\widehat{W} = \begin{bmatrix}\widehat{W}_1 & \widehat{W}_2 & \cdots & \widehat{W}_N\end{bmatrix}$ where

$$\widehat{W}_1 = R, \qquad\qquad \widehat{A}_1 = \begin{bmatrix} C^{(1)} \\ \vdots \\ C^{(\ell)} \end{bmatrix} \widehat{W}_1,$$

and

$$\widehat{W}_{k+1} = F\widehat{A}_k, \qquad\qquad \widehat{A}_{k+1} = \begin{bmatrix} C^{(1)} \\ \vdots \\ C^{(\ell)} \end{bmatrix} \widehat{W}_{k+1} + \begin{bmatrix} s_1 I_{n_1} & & \\ & \ddots & \\ & & s_\ell I_{n_\ell} \end{bmatrix} \widehat{A}_k,$$

for $k = 1, 2, ..., N-1$. Then

$$K_N(\mathcal{M},\mathcal{R}) = \begin{bmatrix} C^{(1)}\widehat{W}\widehat{U}^{(1)} \\ C^{(2)}\widehat{W}\widehat{U}^{(2)} \\ \vdots \\ C^{(\ell)}\widehat{W}\widehat{U}^{(\ell)} \end{bmatrix}, \tag{4.14}$$

where $\widehat{U}^{(i)} \in \mathbb{C}^{m\cdot N \times m\cdot N}$ is nonsingular, upper-triangular, and Toeplitz for $i = 1, 2, ..., \ell$ and $\widehat{W} \in \mathbb{C}^{n_0 \times mN}$. The matrices $\widehat{U}^{(i)}$ are defined as in alternative proof to Proposition 4.1.4, leaving out any reference to the elimination matrices $\{E_j\}_{j=1}^{k_0}$.

Without further assumptions on matrices $\{C^{(i)}\}_{i=1}^{\ell}$, linear dependence relations of the columns of $\widehat{W}$ have no connection to the linear dependence relations of columns of the matrix $K_N(\mathcal{M},\mathcal{R})$ (2.1), as demonstrated in Example 4.1.7. However, a modest assumption on matrix $C^{(1)}$ gives a result analogous to Proposition 3.1.10 for matrices in case two form.

Proposition 4.1.8. Suppose that $C^{(1)}$ has full column rank. Then exact deflation on a column of $K_N(\mathcal{M},\mathcal{R})$ implies linear dependence of the corresponding column of $\widehat{W}$.

Proof: Let $\hat{\mathbf{a}}_j$ be the jth column of $K_N(\mathcal{M},\mathcal{R})$ for $j = 1, 2, ..., mN$. If $\hat{\mathbf{a}}_{n+1} \in \langle \hat{\mathbf{a}}_1, ..., \hat{\mathbf{a}}_n \rangle$ for some $n \in \{1, 2, ..., mN-1\}$, then there exists of a nonzero vector $\mathbf{x} \in \mathbb{C}^{(n+1)}$ such that

$$\begin{bmatrix} \hat{\mathbf{a}}_1 & \cdots & \hat{\mathbf{a}}_n & \hat{\mathbf{a}}_{n+1} \end{bmatrix} \mathbf{x} = \mathbf{0} \in \mathbb{C}^N, \tag{4.15}$$

with $x_{n+1} \neq 0$ and at least one nonzero coefficient $x_j \neq 0$ for $j \in \{1, 2, ..., n\}$. The columns of $\begin{bmatrix} \hat{\mathbf{a}}_1 & \cdots & \hat{\mathbf{a}}_n & \hat{\mathbf{a}}_{n+1} \end{bmatrix}$ can be factored as

$$\begin{bmatrix} C^{(1)} & 0 & \cdots & 0 \\ 0 & C^{(2)} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & C^{(\ell)} \end{bmatrix} \left(I_\ell \otimes \begin{bmatrix} \hat{\mathbf{w}}_1 & \cdots & \hat{\mathbf{w}}_n & \hat{\mathbf{w}}_{n+1} \end{bmatrix} \right) \begin{bmatrix} \widehat{U}_{n+1}^{(1)} \\ \widehat{U}_{n+1}^{(2)} \\ \vdots \\ \widehat{U}_{n+1}^{(\ell)} \end{bmatrix}, \tag{4.16}$$

where $\hat{\mathbf{w}}_j \in \mathbb{C}^{n_0}$ is the jth column of $\widehat{W}$ and $\widehat{U}_{n+1}^{(i)} = U^{(i)}(1 : n+1, 1 : n+1)$ for indices $j = 1, 2, ..., n+1$ and $i = 1, 2, ..., \ell$. Multiplying this factorization (4.16) on the left by the matrix $\begin{bmatrix} I_{n_1} & 0 & \cdots & 0 \end{bmatrix}$ and on the right by nonzero $\mathbf{x}$ (4.15), notice that $\begin{bmatrix} \hat{\mathbf{w}}_1 & \cdots & \hat{\mathbf{w}}_n & \hat{\mathbf{w}}_{n+1} \end{bmatrix} \mathbf{y}$ must be in the kernel of $C^{(1)}$, where the vector

$$\mathbf{y} = \widehat{U}_{n+1}^{(1)} \mathbf{x}.$$

Since $\widehat{U}^{(1)}$ has a nonzero diagonal entries and $\mathbf{x} \neq 0$, $\mathbf{y} \in \mathbb{C}^{n+1}$ must also be nonzero with final entry $y_{n+1} \neq 0$. Because $C^{(1)}$ has full column rank, its null space contains only the zero vector $\mathbf{0} \in \mathbb{C}^{n_0}$, meaning $\hat{\mathbf{w}}_{n+1} \in \langle \hat{\mathbf{w}}_1, \hat{\mathbf{w}}_2, ..., \hat{\mathbf{w}}_n \rangle$. □

The observations above indicate that as long as $C^{(1)}$ has full column rank, any new basis vector for the space S_{n+1} (4.12) corresponds to a new basis vector for the block Krylov subspaces

induced by $\mathcal{M}$ and $\mathcal{R}$. In applications of the Structure Exploiting Arnoldi algorithm to model order reduction of first-order integro-DAEs, the assumption that $C^{(1)}$ has full column rank is satisfied.

Assuming $C^{(1)}$ has full column rank, necessary and sufficient conditions for exact deflation of columns of the block Krylov matrix $K_N(\mathcal{M}, \mathcal{R})$ based on the linear dependence classification of the columns of $\widehat{W}$ can be constructed by extending Corollary 4.1.6. The elimination matrix $E_1 \in \mathbb{C}^{m \times m_1}$ for the first block can be determined by working with $R \in \mathbb{C}^{n_0 \times m}$ since $\mathbf{r}_{n+1} \in \langle \mathbf{r}_1, ..., \mathbf{r}_n \rangle$ if, and only if, $C\mathbf{r}_{n+1} \in \langle C\mathbf{r}_1, ..., C\mathbf{r}_n \rangle$ for $n = 1, 2, ..., m-1$, where $C^T = \begin{bmatrix} \left(C^{(1)}\right)^T & \left(C^{(2)}\right)^T & \cdots & \left(C^{(\ell)}\right)^T \end{bmatrix}^T$. The full-rank factorization of $\mathcal{R}_1$ from the deflated Krylov matrix given in Corollary 4.1.6 is readily available. Assume $V_1^{(i)} \in \mathbb{C}^{n_i \times x_1^{(i)}}$ has orthonormal columns and $C^{(i)}RE_1 = V_1^{(i)}X_{11}^{(i)}$ where the $x_1^{(i)} \times m_1$ matrix $X_{11}^{(i)}$ is in row echelon form for $i = 1, 2, ..., \ell$ and $x_1^{(i)}$ is the number of linearly independent columns of $C^{(i)}RE_1$. Then

$$\mathcal{R}_1 = \mathcal{R}E_1 = \begin{bmatrix} V_1^{(1)}X_{11}^{(1)} \\ V_1^{(2)}X_{11}^{(2)} \\ \vdots \\ V_1^{(\ell)}X_{11}^{(\ell)} \end{bmatrix}.$$

For some $k = 1, 2, ..., k_0 - 1$, assume the elimination matrices $\{E_j\}_{j=1}^{k}$ are known and that the first k blocks of the deflated block Krylov matrix are given by

$$\left[\begin{array}{c|c|c|c} \mathcal{R}_1 & \mathcal{M}\mathcal{R}_2 & \cdots & \mathcal{M}^{k-1}\mathcal{R}_k \end{array}\right] = \begin{bmatrix} \begin{bmatrix} V_1^{(1)} & V_2^{(1)} & \cdots & V_k^{(1)} \end{bmatrix} \begin{bmatrix} X_{11}^{(1)} & \cdots & X_{1k}^{(1)} \\ & \ddots & \vdots \\ & & X_{kk}^{(1)} \end{bmatrix} \\ \vdots \\ \begin{bmatrix} V_1^{(\ell)} & V_2^{(\ell)} & \cdots & V_k^{(\ell)} \end{bmatrix} \begin{bmatrix} X_{11}^{(\ell)} & \cdots & X_{1k}^{(\ell)} \\ & \ddots & \vdots \\ & & X_{kk}^{(\ell)} \end{bmatrix} \end{bmatrix}.$$

In this equation, $V_j^{(i)} \in \mathbb{C}^{n_i \times x_j^{(i)}}$ (3.14), $X_{jt}^{(i)} \in \mathbb{C}^{x_j^{(i)} \times m_t}$ for $j, t = 1, 2, ..., k$, and $x_j^{(i)}$ is the number of new basis vectors for $C^{(i)}$ W added by the jth block $C^{(i)}\widehat{W}_j$. The next block $\mathcal{M}^k\mathcal{R}_{k+1}$ is created

first by forming the product

$$\mathcal{M}(\mathcal{M}^{k-1}\mathcal{R}_k)$$

and then identifying the matrix E_{k+1} needed to form $\mathcal{R}_{k+1} = \mathcal{R}_k E_{k+1}$.

Suppose that $V_{k+1}^{(i)} \in \mathbb{C}^{n_0 \times x_{k+1}^{(i)}}$ is the next block of orthonormal columns for the column space of $C^{(i)}W$ added by matrix the $\mathcal{M}^k\mathcal{R}_k$. Define a set of candidate projection coefficients

$$\begin{bmatrix} \begin{bmatrix} \widehat{X}_{1,k+1}^{(1)} \\ \vdots \\ \widehat{X}_{k,k+1}^{(1)} \\ \widehat{X}_{k+1,k+1}^{(1)} \end{bmatrix} \\ \vdots \\ \begin{bmatrix} \widehat{X}_{1,k+1}^{(\ell)} \\ \vdots \\ \widehat{X}_{k,k+1}^{(\ell)} \\ \widehat{X}_{k+1,k+1}^{(\ell)} \end{bmatrix} \end{bmatrix} = \begin{bmatrix} \left(\tilde{V}_{k+1}^{(1)}\right)^H & 0 & \cdots & 0 \\ 0 & \left(\tilde{V}_{k+1}^{(2)}\right)^H & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \left(\tilde{V}_{k+1}^{(\ell)}\right)^H \end{bmatrix} \mathcal{M}^k\mathcal{R}_k$$

where $\widetilde{V}_{k+1}^{(i)} = \begin{bmatrix} V_1^{(i)} & \cdots & V_k^{(i)} & V_{k+1}^{(i)} \end{bmatrix}$. Then, the matrix

$$\begin{bmatrix} \mathcal{R}_1 & \mathcal{M}\mathcal{R}_2 & \cdots & \mathcal{M}^{k-1}\mathcal{R}_k & \mathcal{M}^k\mathcal{R}_k \end{bmatrix} = \begin{bmatrix} \tilde{V}_{k+1}^{(1)} & 0 & \cdots & 0 \\ 0 & \tilde{V}_{k+1}^{(2)} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \tilde{V}_{k+1}^{(\ell)} \end{bmatrix} X,$$

where

$$X = \begin{bmatrix} \begin{bmatrix} X_{11}^{(1)} & \cdots & X_{1k}^{(1)} & \widehat{X}_{1,k+1}^{(1)} \\ & \ddots & \vdots & \vdots \\ & & X_{kk}^{(1)} & \widehat{X}_{k,k+1}^{(1)} \\ & & & \widehat{X}_{k+1,k+1}^{(1)} \end{bmatrix} \\ \vdots \\ \begin{bmatrix} X_{11}^{(\ell)} & \cdots & X_{1k}^{(\ell)} & \widehat{X}_{1,k+1}^{(\ell)} \\ & \ddots & \vdots & \vdots \\ & & X_{kk}^{(\ell)} & \widehat{X}_{k,k+1}^{(\ell)} \\ & & & \widehat{X}_{k+1,k+1}^{(\ell)} \end{bmatrix} \end{bmatrix}$$

The matrix X is column rank deficient if, and only if, exact deflation occurs on the columns of $\mathcal{M}^k\mathcal{R}_k$. The columns of X can be formed in each iteration of a structure exploiting algorithm without use of an explicit inner product and these columns are much smaller than the columns of the block Krylov matrix $K_N(\mathcal{M}, \mathcal{R})$ in practice.

4.2 The Generic SEA Algorithm for Matrices in Case Two Form

The single-input SEA algorithm for matrices in case two form generates an orthonormal basis for the space S_{n+1} (4.12). Computational improvements for applications in reduced-order modeling of systems of first-order integro-DAEs are discussed in Section 4.3. In the following statement of the SEA algorithm, there are no explicit assumptions on the matrix $C^{(1)}$. For ease of reference, let $C = \begin{bmatrix} \left(C^{(1)}\right)^T & \left(C^{(2)}\right)^T & \cdots & \left(C^{(\ell)}\right)^T \end{bmatrix}^T$ and define

$$S = \begin{bmatrix} s_1 I_{n_1} & 0 & \cdots & 0 \\ 0 & s_2 I_{n_2} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & s_\ell I_{n_\ell} \end{bmatrix}.$$

Algorithm 14 Generic Single-Input Structure Exploiting Arnoldi

Input: $F, C, \{s_i\}_{i=1}^{\ell}, \mathbf{r}$ in case two form 4.1.1.
Output: Orthonormal basis Q_k for the space S_{n+1} (4.12)

1. Compute $\mathbf{q}_1 \coloneqq \mathbf{r}/\|\mathbf{r}\|_2$ and $b_1 \coloneqq 1$
2. Set $\hat{\mathbf{a}}_1 \coloneqq C\mathbf{q}_1$
3. **for** $n = 1, 2, ...,$ until convergence **do**
4. Set $\hat{\mathbf{q}} \coloneqq F\hat{\mathbf{a}}_n$ and $\hat{\mathbf{q}}_{n+1} \coloneqq \hat{\mathbf{q}}$
5. **for** $i = 1, 2, ..., k$ **do**
6. $h_{b_i,n} \coloneqq \hat{\mathbf{q}}_{n+1}^T \mathbf{q}_i$
7. $\hat{\mathbf{q}}_{n+1} \coloneqq \hat{\mathbf{q}}_{n+1} - h_{b_i,n}\mathbf{q}_i$
8. **end for**
9. Check if $\hat{\mathbf{q}}_{n+1}$ is a basis vector for S_{n+1} by checking $\|\hat{\mathbf{q}}_{n+1}\|_2$
10. **if** $\|\hat{\mathbf{q}}_{n+1}\|_2 \neq 0$ **then**
11. Set $h_{n+1,n} \coloneqq \|\hat{\mathbf{q}}_{n+1}\|_2$, $\hat{\mathbf{q}}_{n+1} \coloneqq \hat{\mathbf{q}}_{n+1}/h_{n+1,n}$ and $\mathbf{q}_{k+1} \coloneqq \hat{\mathbf{q}}_{n+1}$
12. Set $b_{k+1} \coloneqq n+1$ and $k \coloneqq k+1$
13. **else**
14. Set $h_{n+1,n} \coloneqq 1$
15. **end if**
16. Set $\hat{\mathbf{a}}_{n+1} \coloneqq \frac{1}{h_{n+1,n}} \left(C\hat{\mathbf{q}} + S\hat{\mathbf{a}}_n - \widehat{A}_n H(1:n, n) \right)$
17. **if** $\hat{\mathbf{a}}_{n+1} \in$ range $\widehat{A}_n$ **then**
18. STOP
19. **end if**
20. **end for**

The theoretic properties of the generic single-input Structure Exploiting Arnoldi algorithm for case two matrices are similar to those of the analogous algorithm for case one matrices. Assuming the single-input SEA algorithm for matrices in case two form runs in exact arithmetic through the end of n iterations of the outer for-loop, the matrices $Q_k, \widehat{A}_{n+1}, \widehat{H}_n$ and pointers $\{b_j\}_{j=1}^{k}$ play similar roles as described in Section 3.2.

Specifically, let the jth column of the matrix $E_b \in \mathbb{C}^{(n+1)\times k}$ be the b_jth column of the $(n+1) \times (n+1)$ identity matrix, where b_j is the pointer stored in line 12, for $j = 1, 2, ..., k$. If $\widehat{Q}_{n+1} = Q_k E_b^T$, then,

$$F\widehat{A}_n = \widehat{Q}_{n+1}\widehat{H}_n, \tag{4.17}$$

$$\mathcal{M}\widehat{A}_n = \widehat{A}_{n+1}\widehat{H}_n. \tag{4.18}$$

Again, $\widehat{H}_n \in \mathbb{C}^{(n+1)\times n}$ is a Hessenberg matrix with nonzero subdiagonal elements.

Because $\widehat{A}_n$ forms a basis matrix for $K(\mathcal{M}, \mathcal{R}, n)$ by Proposition 2.1.10, there exists an upper-triangular, nonsingular $\mathcal{U}_n$ such that $\widehat{A}_n = A_n \, \mathcal{U}_n$, where A_n contains the first n columns of the deflated block Krylov matrix. The columns of the matrix Q_k form a basis for the multiple copied subspace S_{n+1}.

Theorem 4.2.1. Suppose the single-input SEA algorithm for case two matrices runs in exact arithmetic with input data $C, F, \{s_i\}_{i=1}^{\ell}, \mathbf{r}$ through the end of n iterations of the outer for-loop. Assume that the matrix Q produced has exactly $k = k_{n+1}$ columns. Then the sequence of vectors $\mathbf{q}_1, \mathbf{q}_2, ..., \mathbf{q}_k$ forms an orthonormal basis of the space S_{n+1} (4.12).

By construction, the matrix $\widehat{A}_{n+1}$ forms a basis for the Krylov Subspace $K(\mathcal{M}, \mathcal{R}, n+1)$. Hence, given the stop condition of the SEA algorithm for matrices in case two form is equivalent to the Arnoldi stop condition.

Proposition 4.2.2. The generic single-input Structure Exploiting Arnoldi algorithm for matrices in case two form with data $C, F, S, \mathbf{r}$ stops at step n if, and only if, the Arnoldi algorithm with matrix $\mathcal{M}$ and starting vector $\mathcal{R}$ stops at step n.

Remark 4.2.3. The proofs of Theorem 4.2.1 and Proposition 4.2.2 follow from similar arguments as the proofs given in Section 3.2. The key relationships necessary to establish these results are the maintenance of the Hessenberg matrix relationships (4.17), (4.18). Many of the same observations as made in Section 2.2 apply to the generic single-input SEA algorithm presented in this section.

Remark 4.2.4. This generic version of the single-input SEA algorithm for matrices in case two form is a reformulation of the generic single-input SEA algorithm for matrices in case one form. In the case two situation, matrix multiplication with $C^{(i)}$ is more expensive than scalar multiplication with c_i. On the other hand, the scalars s_i for matrices in case two form permit a much simpler update formula than the matrix S in the case one situation.

4.3 Optimizing SEA for First-Order Intego-DAEs

The Structure Exploiting Arnoldi algorithm and the corresponding band SEA algorithm developed in Section 4.4 are intended to function as tools for model order reduction of large-scale systems of first-order integro-DAEs. These algorithms are viable alternatives to existing techniques only if they are designed to minimize computational requirements. The discussion offered at the start of

Section 3.3 applies here. The computational savings offered by the SEA algorithm contrasted with the Arnoldi algorithm come from the inner product calculations while the costs of matrix multiplication with $\mathcal{M}$ and saxpy updates remain constant. When dealing with linearization matrices for systems of first-order integro-DAEs, adaptations to the generic single-input SEA algorithm lead to a more efficient and faster algorithm. For the rest of this chapter, assume that $\ell = 2$, $n_1 = n_0$, $s_1 = 0$, $C^{(1)} = I_{n_0}$ and that matrices $\mathcal{M}$ and $\mathcal{R}$ are given by

$$\mathcal{M} = \begin{bmatrix} I_{n_0} \\ C^{(2)} \end{bmatrix} \begin{bmatrix} M^{(1)} & M^{(2)} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & s_2 I_{n_2} \end{bmatrix}, \qquad \mathcal{R} = \begin{bmatrix} I_{n_0} \\ C^{(2)} \end{bmatrix} R. \tag{4.19}$$

The matrices described in Propositions 4.1.2 and 4.1.3 exhibit such structure.

4.3.1 The Stop Condition via the Full-Rank Factorization

The stop condition in line 17 of the single-input SEA algorithm 14 does not lend itself to fast and efficient computations, compared with existing techniques. However, the full-rank factorization suggested in Proposition 4.1.6 provides an equivalent method for checking the rank of $\widehat{A}_{n+1}$. Assume the SEA algorithm for special matrices in case two form (4.19) runs in exact arithmetic for n iterations. Given matrices $\widehat{Q}_{n+1}$ and $C^{(2)}$, define the functions

$$k : \{1, 2, ..., n+1\} \to \{1, 2, ..., n+1\},$$
$$\sigma : \{1, 2, ..., n+1\} \to \{0, 1, 2, ..., n+1\}$$

such that $k(j) = k_j$ and $\sigma(j) = \sigma_j$ represent the number of linearly independent columns of $\widehat{Q}_j$ and $C^{(2)}\widehat{Q}_j$ respectively for $j = 1, 2, ..., n+1$. Suppose $k = k_{n+1}$ and $\beta = \{b_1, b_2, ..., b_k\}$ is the set of pointers stored in line 12. Let $\delta = \{1, ..., n, n+1\} - \beta$, where the jth smallest element of δ is d_j for $j = 1, ..., n-k+1$ and $d_i < d_j$ if $i < j$. Recall that $\widehat{Q}_{n+1} = Q_k E_b^T$ and the jth column of $E_b \in \mathbb{C}^{(n+1)\times k}$ is the b_jth column of I_{n+1}. Just as in Section 3.3, the k nonzero columns of E_b^T are the column vectors of I_k and $F\widehat{A}_n = \widehat{Q}_{n+1}\widehat{H}_n$ (4.17).

As demonstrated below, as long as $\widehat{A}_{n+1}$ forms a basis matrix for $K(\mathcal{M}, \mathcal{R}, n)$, then

$$\widehat{A}_{n+1} = \begin{bmatrix} Q_{k_{n+1}} & 0 \\ 0 & V_{\sigma_{n+1}} \end{bmatrix} \begin{bmatrix} X_{n+1}^{(1)} \\ X_{n+1}^{(2)} \end{bmatrix},$$

with $V_{\sigma_{n+1}}$ and $X_{n+1}^{(i)}$ of appropriate sizes for $i = 1, 2, ..., \ell$. The matrices $Q_{k_{n+1}}$ and $V_{\sigma_{n+1}}$ have orthonormal columns forming a basis for the column spaces $\widehat{Q}_{n+1}$ and $C^{(2)}\widehat{Q}_{n+1}$ respectively. Assuming the matrices $X_{n+1}^{(i)}$ are available within the algorithm, the stop condition in line 17 can be checked by working with the matrix

$$X = \begin{bmatrix} X_{n+1}^{(1)} \\ X_{n+1}^{(2)} \end{bmatrix}.$$

The last column of $\widehat{A}_{n+1}$ is linearly dependent on early columns if, and only if, the matrix X is column rank deficient. A slight restructuring of the SEA algorithm ensures that the rank of matrix X is easily ascertained.

4.3.2 Optimizing Updates for the Columns of $\widehat{A}_{n+1}$

Given the extra assumptions on the input matrices $\mathcal{M}$ and $\mathcal{R}$ (4.19), an efficient way to check the stop condition in line 17 of the single-input SEA algorithm 14 is to redesign the procedure to take advantage of the full-rank factorization suggested in Proposition 4.1.6. The generic update for columns of the matrix $\widehat{A}_{n+1}$ stated in line 16 guarantees the resulting basis matrix Q_k spans the proper space S_{n+1}. However, this update is not a prudent choice from the perspective of optimizing the stop condition check in line 17. A better update for this purpose is given by

$$\hat{\mathbf{a}}_{n+1} = \begin{bmatrix} I_{n_0} \\ C^{(2)} \end{bmatrix} \hat{\mathbf{q}}_{n+1} + \begin{bmatrix} 0 & 0 \\ 0 & s_2 I_{n_2} \end{bmatrix} \widehat{A}_n \; \widehat{H}_n(2:n, 1:n)^{-1} \mathbf{e}_n \tag{4.20}$$

where $\mathbf{e}_n$ is the nth column of I_n. The resulting matrix equation is

$$\widehat{A}_{n+1} = \begin{bmatrix} I_{n_0} \\ C^{(2)} \end{bmatrix} \widehat{Q}_{n+1} + \begin{bmatrix} 0 & 0 \\ 0 & s_2 I_{n_2} \end{bmatrix} \widehat{A}_n \begin{bmatrix} \mathbf{0} & \widehat{H}_n(2:n, 1:n)^{-1} \end{bmatrix}. \tag{4.21}$$

Multiplication by $\widehat{H}_n$ verifies that $\mathcal{M}\widehat{A}_n = \widehat{A}_{n+1}\widehat{H}_n$ and the columns of $\widehat{A}_{n+1}$ span the Krylov subspace $K(\mathcal{M}, \mathcal{R}, n+1)$ according to Proposition 2.1.10. This alternative update for vectors $\hat{\mathbf{a}}_{n+1}$ affords a relatively inexpensive method to check the stop condition during the nth iteration. As is clear from the matrix equation for the alternative update (4.21), $X_{n+1}^{(1)} = E_b^T$, greatly simplifying

the procedure to check the rank of X. All that is needed is an efficient technique to generate the matrix $X^{(2)}$.

4.3.3 Generating the Matrix $X^{(2)}$

Matrix $X^{(2)}$ can be generated as a byproduct of the forming an orthonormal basis for $C^{(2)}\widehat{Q}_{n+1}$. Since the null space of $C^{(2)}$ will be nontrivial in general, it may be the case that $C^{(2)}\mathbf{q}_1 = \mathbf{0}$. Suppose that the first nonzero column of the $n_2 \times (n+1)$ matrix $C^{(2)}\widehat{Q}_{n+1}$ is the jth column $C^{(2)}\hat{\mathbf{q}}_j$ for some $j = 1, 2, ..., n$. If $j > 1$, the first $j-1$ columns of $\widehat{A}_{n+1}$ can be factorized using only the matrix $\widehat{Q}_{j-1}$ whose columns must be nonzero if $\widehat{A}_{j-1}$ has full column rank. To factorize $\widehat{A}_j$, set $x_{11}^{(2)} = \cdots = x_{1,j-1}^{(2)} = 0$ and $x_{1j}^{(2)} = \|C^2\hat{\mathbf{q}}_j\|_2$. Prior to the nth iteration of the SEA algorithm, assume all data from the equivalent factorization

$$\widehat{A}_n = \begin{bmatrix} Q_{k_n} & 0 \\ 0 & V_{\sigma_n} \end{bmatrix} \begin{bmatrix} E_b^T \\ X_n^{(2)} \end{bmatrix}$$

is available within the algorithm with $X_n^{(2)} \in \mathbb{C}^{\sigma_n \times n}$. Then, the alternative update of $\hat{\mathbf{a}}_{n+1}$ (4.20) during the nth iteration of the SEA algorithm is given by

$$\hat{\mathbf{a}}_{n+1} = \begin{bmatrix} I_{n_0} \\ C^{(2)} \end{bmatrix} \hat{\mathbf{q}}_{n+1} + \begin{bmatrix} Q_{k_n} & 0 \\ 0 & V_{\sigma_n} \end{bmatrix} \begin{bmatrix} \mathbf{0} \\ \mathbf{y}_2 \end{bmatrix}$$

with $\mathbf{y}_2 = s_2 X_n^{(2)} \hat{H}_n(2 : n+1, 1 : n)^{-1} \mathbf{e}_n$. There are two possibilities to consider.

If $\hat{\mathbf{q}}_{n+1} = \mathbf{0}$, then $\sigma_{n+1} = \sigma_n$ and $X^{(2)}(:, 1 : n+1) = \mathbf{y}_2 \in \mathbb{C}^{\sigma_n}$. On the other hand, if $\hat{\mathbf{q}}_{n+1} \neq 0$, no stop condition needs to be checked. Given $C^{(2)}\hat{\mathbf{q}}_{n+1}$ is in the column space of V_{σ_n}, then $\sigma_{n+1} = \sigma_n$ and there is a vector $\mathbf{x}$ such that

$$C^{(2)}\hat{\mathbf{q}}_{n+1} = V_{\sigma_n}\mathbf{x} \tag{4.22}$$

Setting $X_{n+1}^{(2)} = \mathbf{x} + \mathbf{y}_2 \in \mathbb{C}^{\sigma_n}$ ensures a proper update of $X^{(2)}$. If $C^{(2)}\mathbf{q}_{n+1}$ is not in the column span of V_{σ_n}, then $\sigma_{n+1} = \sigma_n + 1$. Using a modified Gram-Schmidt orthogonalization, there is a

vector $\mathbf{x} \in \mathbb{C}^{\sigma_n}$ and scalar α such that

$$C^{(2)}\hat{\mathbf{q}}_{n+1} = \begin{bmatrix} V_{\sigma_n} & \mathbf{v}_{\sigma_{n+1}} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \alpha \end{bmatrix} \tag{4.23}$$

The final column $X^{(2)}$ in this case is updated as

$$\begin{bmatrix} \mathbf{x} \\ \alpha \end{bmatrix} + \begin{bmatrix} \mathbf{y}_2 \\ 0 \end{bmatrix}.$$

Since $X^{(1)}_{n+1} = E_b^T \in \mathbb{C}^{k_n \times (n+1)}$, according to Lemma 3.3.1, the matrix $\widehat{A}_{n+1}$ is column rank deficient if, and only if,

$$\mathbf{p}_{n-k_n+1} \in \text{ span } \{\mathbf{p}_1, ..., \mathbf{p}_{n-k_n}\}, \tag{4.24}$$

with $\mathbf{p}_j = X^{(2)}_{n+1}(:, d_j) \in \mathbb{C}^{\sigma_n}$ for $j = 1, 2, ..., n - k_{n+1} + 1$. This check can be executed using the modified Gram-Schmidt algorithm, the SVD algorithm or the rank-revealing QR factorization. The following subroutine generates the coefficients for $X^{(2)}(1 : \sigma_n, n + 1)$ and facilitates the evaluation of the equivalent stop condition.

Algorithm 15 Form Candidate Coefficients for $X^{(2)}(1 : \sigma_n, n + 1)$

Input: Vectors $\mathbf{g}_n$, scalar s_2, matrix $X^{(2)}_n$, and vector $\mathbf{x}$ from either (4.22) or (4.23).
Output: Coefficients $X^{(2)}(1 : \sigma_n, n + 1) \coloneqq \mathbf{x}^{(2)}$.

1. Set $\mathbf{y}_2 \coloneqq X^{(i)}_n \mathbf{g}_n$ for $i = 1, 2, ..., \ell$
2. Set $\mathbf{x}^{(2)} \coloneqq \mathbf{x} + s_2\mathbf{y}_2$

4.3.4 Computationally Efficient Single-Input SEA Algorithm

The single-input SEA algorithm for matrices in case two form can be restated as indicated below. This efficient version of the SEA algorithm uses the alternative update for $\hat{\mathbf{a}}_{n+1}$ (4.20) and the accelerated stop condition (4.24). New columns of matrices $X^{(2)}_{n+1}$ are formed during the nth iteration based on the subroutine presented above. For simplicity, it is assumed that $C^{(2)}\mathbf{q}_1$ is nonzero in this algorithm. The more general case in which the kernel of $C^{(2)}$ affects the initialization can be coded using an if-statement to properly initialize the first column of V_σ and the corresponding first row of $X^{(2)}$. A modified Gram-Schmidt style subroutine is used in this algorithm and will be referred to as MGS.

Algorithm 16 Specialized Single-Input Structure Exploiting Arnoldi

Input: $F, C^{(2)}s_2, \mathbf{r}$ in specialized case two form 4.19.
Output: Orthonormal basis Q_k, V_σ

1. Set $\mathbf{q}_1 \coloneqq \mathbf{r}/\|\mathbf{r}\|_2$, $b_1 \coloneqq 1$, and $k \coloneqq 1$
2. Set $\alpha \coloneqq \|C^{(2)}\mathbf{q}_1\|_2$ and $\mathbf{v}_1 \coloneqq C^{(2)}\mathbf{q}_1/\alpha$
3. Set $\hat{\mathbf{a}}_1 \coloneqq \mathbf{c} \otimes \mathbf{q}_1$ and $x_{11}^{(2)} \coloneqq \alpha$ for $i = 1, 2, ..., \ell$
4. **for** $n = 1, 2, ...$ **do**
5. $\quad \hat{\mathbf{q}}_{n+1} \coloneqq F\hat{\mathbf{a}}_n$
6. $\quad$ **for** $i = 1, 2, ..., k$ **do**
7. $\quad\quad h_{b_i,n} \coloneqq \hat{\mathbf{q}}_{n+1}^T \mathbf{q}_i$
8. $\quad\quad \hat{\mathbf{q}}_{n+1} \coloneqq \hat{\mathbf{q}}_{n+1} - h_{b_i,n}\mathbf{q}_i$
9. $\quad$ **end for**
10. $\quad$ Check if $\hat{\mathbf{q}}_{n+1}$ is a basis vector for S_{n+1}
11. $\quad$ **if** $\hat{\mathbf{q}}_{n+1}$ is a basis vector **then**
12. $\quad\quad$ Set $h_{n+1,n} \coloneqq \|\hat{\mathbf{q}}_{n+1}\|_2$, $\hat{\mathbf{q}}_{n+1} \coloneqq \hat{\mathbf{q}}_{n+1}/h_{n+1,n}$ and $\mathbf{q}_{k+1} \coloneqq \hat{\mathbf{q}}_{n+1}$
13. $\quad\quad$ Run MGS on $(V_{\sigma_n}, C^{(2)}\mathbf{q}_{k+1})$ to produce $\mathbf{v}_{\sigma_{n+1}}$ and $X^{(2)}(1 : \sigma_{n+1}, n+1)$
14. $\quad\quad$ Set $b_{k+1} \coloneqq n+1$, $k \coloneqq k+1$
15. $\quad$ **else**
16. $\quad\quad$ Generate proper $X^{(2)}(1 : k, n+1)$
17. $\quad\quad$ Evaluate equivalent stop check using $X_{n+1}^{(2)}$ and $\{d_j\}_{j=1}^{n-k+1}$ (4.24).
18. $\quad\quad$ **if** Large space is exhausted **then**
19. $\quad\quad\quad$ STOP
20. $\quad\quad$ **else**
21. $\quad\quad\quad$ Set $h_{n+1,n} \coloneqq 1$ and $d_{n-k+1} = n+1$
22. $\quad\quad$ **end if**
23. $\quad\quad$ Set

$$\hat{\mathbf{a}}_{n+1} \coloneqq \begin{bmatrix} I_{n_0} \\ C^{(2)} \end{bmatrix} \hat{\mathbf{q}}_{n+1} + \begin{bmatrix} 0 & 0 \\ 0 & s_2 I_{n_2} \end{bmatrix} \widehat{A}_n \, \widehat{H}_n(2 : n+1, 1 : n)^{-1} \, \mathbf{e}_n$$

24. $\quad$ **end if**
25. **end for**

4.4 The Band SEA Algorithm for Matrices in Case Two Form

The implementation of the generic band version of the SEA algorithm for matrices in case two form is almost identical to algorithm 11 from Section 3.4. The only difference is that matrix-matrix multiplication with $C = \begin{bmatrix} \left(C^{(1)}\right)^T & \left(C^{(2)}\right)^T & \cdots & \left(C^{(\ell)}\right)^T \end{bmatrix}^T$ replaces the Kronecker product with $\mathbf{c}$ and the scalars $\{s_i\}_{i=1}^{\ell}$ must be accounted for properly. In this form, the band SEA algorithm produces an orthonormal basis for the multiple copied subspace S_n (4.12). The assumption that $C^{(1)}$ has full column rank must be enforced to ensure the results of Proposition 4.1.8 still apply. The subroutine `band_SEA_start` initializes the band SEA algorithm by preprocessing $R \in \mathbb{C}^{n_0 \times m}$ with a modified Gram-Schmidt orthogonalization to produce $Q_1 \in \mathbb{C}^{n_0 \times m_1}$. The corresponding

first block of basis vectors for the column span of $\mathcal{R}$ is generated using the update $\widehat{A}_1 = CQ_1$. All other details of this generic band SEA algorithm for matrices in case two form follow from an identical analysis as that provided in Section 3.4.

Given the additional structure assumptions (4.19) arising from applications to reduced-order modeling of first-order integro-DAEs, the generic band SEA algorithm can be modified to improve efficiency. The alternative update yielding a faster deflation detection mechanism for this subclass of matrices is given by

$$V_{n_{j+1}} = \begin{bmatrix} I_{n_0} \\ C^{(2)} \end{bmatrix} \begin{bmatrix} \widehat{Q}_1 & \widehat{Q}_2 & \cdots & \widehat{Q}_{j+1} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & s_2 I_{n_2} \end{bmatrix} V_{n_j} E_{B_{p_j}} \begin{bmatrix} 0 & G_{\tilde{n}_j} \end{bmatrix} \tag{4.25}$$

where $\{n_i\}_{i=1}^{k_0}, \{\widehat{Q}_i\}_{i=1}^{j+1}, V_{n_j}, E_{B_{p_j}}, \widehat{H_{n_j}}$ and $G_{\tilde{n}_j}$ are defined for special case two structure (4.19) analogously to the definitions given in in Section 3.4. Multiplying both sides of this equation by $\widehat{H_{n_j}}$ yields $\mathcal{M} V_{n_j} E_{B_{p_j}} = V_{n_{j+1}} \widehat{H_{n_j}}$. The equivalent stop condition for this alternative update requires the construction of matrices $X^{(2)}$ such that

$$\begin{bmatrix} \hat{\mathbf{a}}_1 & \hat{\mathbf{a}}_2 & \cdots & \hat{\mathbf{a}}_n & \hat{\mathbf{a}}_{n+1} \end{bmatrix} = \left(I_\ell \otimes \begin{bmatrix} \mathbf{q}_1 & \mathbf{q}_2 & \cdots & \mathbf{q}_k \end{bmatrix} \right) \begin{bmatrix} c_1 E_b^T \\ X_{n+1}^{(2)} \end{bmatrix}$$

Block entry $(1,1)$ of $X^{(2)}$ is a $\sigma_{m_1} \times m_1$ upper-triangular matrix in row echelon form and comes from a factorization of $C^{(2)} Q_1 = V_{\sigma_{m_1}} X_{11}^{(2)}$, where the columns of $V_{\sigma_{m_1}}$ are orthonormal and span the range of $C^2 Q_1$. Subsequent columns of $X^{(2)}$ can be calculated using the column relationships illustrated in the alternative update formula (4.25). Because $s_1 = 0$, Lemma 3.3.1 applies. The band SEA algorithm using alternate update equation (3.25) processes special case two data in a column-by-column manner ensuring accurate exact deflation and improved efficiency. This algorithm, given below, is one possible optimization of the generic band SEA process and relies on the equivalent stop condition using matrix $X^{(2)}$. Numerical results for applications of this specialized band SEA algorithm in reduced-order modeling of first-order integro-DAEs are presented in the next section.

Algorithm 17 Specialized Band Structure Exploiting Arnoldi

Input: $C^{(2)}, F, s_2 \in \mathbb{C}, R \in \mathbb{C}^{n_0 \times m}$ in specialized case two form 4.19and integer nmax $> m$.
Output: ONB Q_k for S_{nmax} (4.12)

1. Initialize Band SEA Algorithm: $[Q_1, V_{\sigma_{m_1}}, X_2, m_1, \widehat{A}_1] \coloneqq \texttt{band\_SEA\_start}(C^{(2)}, R)$
2. Set $m_c = m_1$, $\{b_i = i\}_{i=1}^{m_1}$, $n \coloneqq m_1$
3. **while** $n <$ nmax **do**
4. Set $\hat{\mathbf{q}}_{n+1} \coloneqq F\hat{\mathbf{a}}_{n-m_c+1}$
5. Set $\texttt{tmp} \coloneqq \mathbf{0} \in \mathbb{C}^n$
6. **for** $i = 1, 2, ..., k$ **do**
7. $\texttt{tmp}(b_i, 1) \coloneqq \hat{\mathbf{q}}_{n+1}^T \mathbf{q}_i$
8. $\hat{\mathbf{q}}_{n+1} \coloneqq \hat{\mathbf{q}}_{n+1} - \texttt{tmp}(b_i, 1)\mathbf{q}_i$
9. **end for**
10. Decide if $\hat{\mathbf{q}}_{n+1}$ is a basis vector.
11. **if** $\hat{\mathbf{q}}_{n+1}$ is a basis vector **then**
12. Set $\widehat{H}(1 : n, n - m_1 + 1) \coloneqq \texttt{tmp}$ and $\widehat{H}(n + 1, n - m_1 + 1) \coloneqq \|\hat{\mathbf{q}}_{n+1}\|_2$
13. Set $\hat{\mathbf{q}}_{n+1} \coloneqq \frac{1}{\|\hat{\mathbf{q}}_{n+1}\|_2}\hat{\mathbf{q}}_{n+1}$, $\mathbf{q}_{k+1} \coloneqq \hat{\mathbf{q}}_{n+1}$, $b_{k+1} \coloneqq n + 1$ and $B_{p_{n-m_1+1}} \coloneqq n - m_c + 1$
14. Set $\mathbf{g}_{n-m_1+1} = \widehat{H}_n(m_1 + 1 : n + 1, 1 : n - m_1 + 1)^{-1}\, \mathbf{e}_{n-m_1+1}$
15. Set $\hat{\mathbf{a}}_{n+1} \coloneqq \begin{bmatrix} I_{n_0} \\ C^{(2)} \end{bmatrix} \hat{\mathbf{q}}_{n+1} + \begin{bmatrix} 0 & 0 \\ 0 & s_2 \end{bmatrix} \widehat{A}_n(:, B_p)\, \mathbf{g}_{n-m_1+1}$
16. Run MGS on $(V_{\sigma_n}, C^{(2)}\mathbf{q}_{k+1})$ to produce $\mathbf{v}_{\sigma_{n+1}}$ and $X^{(2)}(1 : \sigma_{n+1}, n + 1)$
17. Set $k \coloneqq k + 1$ and $n \coloneqq n + 1$
18. **else**
19. Set $\mathbf{g}_{n-m_1+1} = -\big(\widehat{H}(m_1 + 1 : n, 1 : n - m_1)\big)^{-1}\, \texttt{tmp}(m_1 + 1 : n, 1)$
20. Set $\mathbf{x}_2 = s_2 \left(X^{(2)}(1 : k, B_p)\mathbf{g}_{n-m_1+1} + X^{(2)}(1 : k, n - m_c + 1)\right)$
21. Decide deflation using matrix $X_n^{(2)}$, vectors $\mathbf{x}_2$ and pointers $\{d_j\}_{j=1}^{n-k+1}$
22. **if** Deflation should occur **then**
23. Set $m_c = m_c - 1$.
24. **if** $m_c = 0$ **then**
25. STOP
26. **end if**
27. **else**
28. Set $X^{(2)}(1 : k, n + 1) \coloneqq \mathbf{x}_2$
29. Set $\hat{\mathbf{a}}_{n+1} \coloneqq \begin{bmatrix} I_{n_0} \\ C^{(2)} \end{bmatrix} \hat{\mathbf{q}}_{n+1} + \begin{bmatrix} 0 & 0 \\ 0 & s_2 \end{bmatrix} \left(\widehat{A}_n(:, B_p)\, \mathbf{g}_{n-m_1+1} + \mathbf{a}_{n-m_c+1}\right)$
30. Set $\widehat{H}(1 : n, n - m_1 + 1) \coloneqq \texttt{tmp}$ and $\widehat{H}(n + 1, n - m_1 + 1) \coloneqq 1$
31. Set $d_{n-k+1} = n + 1$, $B_{p_{n-m_1+1}} \coloneqq n - m_c + 1$ and $n \coloneqq n + 1$
32. **end if**
33. **end if**
34. **end while**

4.5 Applications to Model Order Reduction

The SEA algorithm for matrices in case two form is devised as a tool for dimension reduction of RCL circuit models arising in VLSI interconnect analysis [**Fre00**], [**CPO02**], [**BHtM11**]. The VLSI interconnect network is a system of tiny wires that transmit electronic signals between tens of millions of devices embedded in today's integrated circuits. During the design phase for these circuits, numerical simulations are used to analyze circuit behavior and correct possible architecture flaws before the circuit is fabricated in silicon [**She96**]. The entire manufacturing process, from the determination of design specifications to the printing of the completed electronic circuit is implemented with circuit simulation in mind. The history and development of computer aided design tools for VLSI synthesis and analysis is connected to the development of mathematical algorithms for accurate circuit modeling [**Rue86**].

Electronic circuit simulation is the art of modeling the behavior of electronic devices using mathematical equations. In general, models used to simulate integrated circuits are large-scale, sparse systems of nonlinear time-dependent differential-algebraic equations (DAEs) [**VS83**], [**FF95b**]. Numerical computations of time-integrals for these systems are often prohibitive due to the enormous dimensionality of the complete system of equations. However, integrated circuits contain large subcircuits, such as the VLSI interconnect wires, that contribute only linear equations to the entire system of DAEs. By replacing the linear subsystems of equations with accurate reduced-order models having much smaller state-space dimension, time integration of the reduced, complete system via numerical methods becomes feasible. It is with this strategy in mind that reduced-order modeling of linear DAEs, such as those arising as RCL models of the VLSI interconnect, becomes important.

The topology of linear circuits is often modeled as a graph with the circuit elements represented as edges and the connections between these elements represented as nodes [**Deo74**], [**VS83**], [**Ogr94**], [**FF95b**]. Physical laws that govern electrical systems are introduced as mathematical equations to describe the behavior of entire linear circuits. The three types of physical laws used to form the system of equations for linear circuits include Kirchoff's current law (KCL), Kirchoff's voltage law (KVL), and branch constituent relations (BCRs). While KCLs and KVLs depend only

on the connectivity of the circuit, the BCRs describe the functioning of each physical circuit element. Linear RCL circuits are modeled using only ideal resistors, capacitors, inductors, current sources and voltage sources with corresponding BCRs given as linear equations.

Combining a description of every circuit element with the topology of the entire linear circuit, the associated system of differential-algebraic equations are stated directly in terms of the three types of physical laws mentioned above. For linear circuits with thousands of ideal elements, the large-scale, sparse linear systems of time-invariant differential-algebraic equations that result are formulated by most circuit simulation software packages. Design engineers need only to specify a circuit's netlist, a typeset list encoding all relevant information about the electronic circuit. Given this netlist, the electronic design automation software of choice executes the rest of the analysis.

4.5.1 Creating Matrices in Case Two Form from RCL Circuits

Techniques to generate descriptor systems from an RCL circuit can be found in the literature on this subject [**Fre03b**], [**Fre04**], [**BMS05**], [**Fre08**], [**Fre11**], [**LSF11**], [**BHtM11**]. For the sake of completeness, these methods are reviewed here. In this exposition, it is assumed that only ideal current sources are connected to the ports of a given RCL circuit. Starting with a circuit's netlist, there are a series of four steps to create the mathematical model to which the SEA algorithm can be applied:

(1) Formulate the circuit equations using the lumped element approach

(2) Express the entire circuit using the sparse tableu formulation

(3) Eliminate as many circuit variables as possible using modified nodal analysis

(4) Form the input matrices to the SEA algorithm

Begin by transforming the netlist into a directed graph (digraph) model of the original RCL circuit. Let $\mathcal{N}$ denote the set of all nodes and $\mathcal{E}$ as the set of all edges. Each $n \in \mathcal{N}$ represents the wire connection between different circuit elements while each $e \in \mathcal{E}$ represents a physical circuit element such as capacitors, resistors, inductors or current sources. The direction of each edge corresponds to the reference directions chosen for each circuit element, with the reference direction for each current source being the direction of current flow and all other reference directions being chosen to satisfy the passive sign convention. Using this encoding, let $\mathcal{G}(\mathcal{N}, \mathcal{E})$ be the digraph representation of the RCL circuit corresponding to the specified netlist. This directed graph has an

associated incidence matrix $\widetilde{\mathcal{A}} \in \mathbb{C}^{|\mathcal{N}|\times|\mathcal{E}|}$ whose row and column indices correspond to the nodes and edges of $\mathcal{G}$, respectively. Individual coefficients of this incidence matrix are given by

$$a_{ik} = \begin{cases} 1 & \text{if reference direction of element matching edge } e_k \text{ leaves node } n_i, \\ -1 & \text{if reference direction of element matching edge } e_k \text{ enters node } n_i, \\ 0 & \text{otherwise.} \end{cases} \tag{4.26}$$

While the matrix $\widetilde{\mathcal{A}}$ encapsulates the topological information of the physical circuit, it is rank deficient since the rows sum to zero by the definition of the coefficients of this matrix. In order to transform the incidence matrix into a full-rank version, choose one of the nodes $n_g \in \mathcal{N}$ as the datum (or ground) node and assume the voltage at that node is zero. Form the reduced incidence matrix $\mathcal{A}$ by deleting the row corresponding to this ground node from the original incidence matrix $\widetilde{\mathcal{A}}$. This gives rise to a new index set $\mathcal{N}_0 = \mathcal{N} - \{n_g\}$. As discussed in the literature on graph theory (theorem 9-6, [**Deo74**]), the reduced incidence matrix has full row rank, with rank $(\mathcal{A}) = |\mathcal{N}_0|$.

Let $\mathbf{v} : [t_0, \infty) \to \mathbb{R}^{|\mathcal{N}_0|}$ be the vector of node voltages at all non-datum (non-ground) nodes, where the jth entry $v_j(t)$ represents the voltage at node j for $j = 1, 2, ..., |\mathcal{N}_0|$. The voltage of the ground node is assumed to be zero and is used as a reference to measure all other node voltages. Let $\mathbf{i}_{\mathcal{E}}(t)$ from $[t_0, \infty)$ to $\mathbb{R}^{|\mathcal{E}|}$ be the vector whose kth element is the current running through the circuit element corresponding to edge e_k. Finally, let $\mathbf{v}_{\mathcal{E}}(t)$ from $[t_0, \infty)$ into $\mathbb{R}^{|\mathcal{E}|}$ denote the vector whose kth entry is the voltage across the circuit element corresponding to edge e_k.

Choose indices for the columns of $\mathcal{A}$ such that edges corresponding to circuit elements of the same type are grouped together, inspiring the column partition

$$\mathcal{A} = \begin{bmatrix} \mathcal{A}_c & \mathcal{A}_r & \mathcal{A}_l & \mathcal{A}_i \end{bmatrix}.$$

Matrix $\mathcal{A}_c \in \mathbb{R}^{|\mathcal{N}_0|\times n_c}$ represent all edges corresponding to capacitors, where n_c is the number of capacitors specified in the circuits netlist, while $\mathcal{A}_r, \mathcal{A}_l$, and $\mathcal{A}_i$ are defined analogously for the edges corresponding to resistors (r), inductors (l) and current sources (i). The row indices of

vector-valued functions $\mathbf{i}_{\mathcal{E}}$ and $\mathbf{v}_{\mathcal{E}}$ should be chosen in same ordering as the columns of $\mathcal{A}$ so that

$$\mathbf{i}_{\mathcal{E}}(t) = \begin{bmatrix} \mathbf{i}_c(t) \\ \mathbf{i}_r(t) \\ \mathbf{i}_l(t) \\ \mathbf{i}_i(t) \end{bmatrix}, \qquad \mathbf{v}_{\mathcal{E}}(t) = \begin{bmatrix} \mathbf{v}_c(t) \\ \mathbf{v}_r(t) \\ \mathbf{v}_l(t) \\ \mathbf{v}_i(t) \end{bmatrix}.$$

In this case $\mathbf{v}_c : [t_0, \infty) \to \mathbb{R}^{n_c}$ is the vector-valued function describing the voltages across each capacitor while $\mathbf{i}_c : [t_0, \infty) \to \mathbb{R}^{n_c}$ describes the current passing through each capacitor. Similar definitions hold for the other partitions corresponding to resistors, inductors and current sources. In this index mapping, similar elements are lumped together, giving rise to the name of the lumped element approach.

The next stage of the model generation process is the sparse tableau formulation method to represent the modeled RCL circuit completely by a set of linear and differential equations. Both Kirchhoff's current law and Kirchhoff's voltage law can be formulated using the matrix $\mathcal{A}$ with the complete list of KCLs given by

$$\mathcal{A}\ \mathbf{i}_{\mathcal{E}} = \mathcal{A}_c\mathbf{i}_c + \mathcal{A}_r\mathbf{i}_r + \mathcal{A}_l\mathbf{i}_l + \mathcal{A}_i\mathbf{i}_i = \mathbf{0} \tag{4.27}$$

and the complete list of KVLs given as

$$\mathcal{A}^T\mathbf{v} = \mathbf{v}_{\mathcal{E}} \qquad \Longleftrightarrow \qquad \mathcal{A}_c^T\mathbf{v} = \mathbf{v}_c, \qquad \mathcal{A}_r^T\mathbf{v} = \mathbf{v}_r, \qquad \mathcal{A}_l^T\mathbf{v} = \mathbf{v}_l, \qquad \mathcal{A}_i^T\mathbf{v} = \mathbf{v}_i. \tag{4.28}$$

The final system of equations that completely determining the given linear circuit are formed via the BCRs governing the behavior of each physical circuit element and are state succinctly using the matrix equations

$$\mathbf{v}_r(t) = R \cdot \mathbf{i}_r(t), \qquad \mathbf{i}_c(t) = C \cdot \frac{d}{dt}\mathbf{v}_c(t), \qquad \mathbf{v}_l(t) = L \cdot \frac{d}{dt}\mathbf{i}_l(t). \tag{4.29}$$

In these relations, R and C are diagonal matrices with positive entries. Diagonal entries of R represent the resistances of corresponding ideal resistors while the diagonal entries of C encode capacitances of the corresponding ideal capacitors in the circuit schematic. The entries of matrix

L are inductances and, if necessary, mutual inductances. In general, L is a symmetric semidefinite matrix. If there are no mutual inductances, L is diagonal with positive entries.

The long list of linear differential equations resulting from the sparse tableau formulation method are as sparse as possible and no elimination of variables has occurred. The input vector $\mathbf{i}_i(t)$ gives the waveforms provided by all current sources for $t \geq t_0$ and is controlled by the design engineer.

The third step in the process of creating matrices in case two from from a given RCL circuit is known as the modified nodal analysis formulation method and condenses the long list of sparse tableau equations by eliminating as many variables as possible. By combining the KVL equations (4.28) and the BCRs,

$$\mathbf{i}_c(t) = C\frac{d}{dt}\mathcal{A}_c^T\mathbf{v}(t), \qquad \mathbf{i}_r(t) = R^{-1}\mathcal{A}_r^T\mathbf{v}, \qquad \mathbf{i}_l(t) = L^{-1}A_l^T\int_{t_0}^{t}\mathbf{v}(\tau)d\tau.$$

Introducing notation $E_{11} = \mathcal{A}_c C\mathcal{A}_c^T,\ \mathcal{A}_{11} = -\mathcal{A}_r R^{-1}\mathcal{A}_r^T$, and making the appropriate substitutions into the matrix KCL equations (4.27) yields

$$E_{11}\frac{d}{dt}\mathbf{v}(t) + \mathcal{A}_{11}\mathbf{v}(t) + \mathcal{A}_l L^{-1}\mathcal{A}_l^T\int_{0}^{t}\mathbf{v}(\tau)d\tau = -\mathcal{A}_i\mathbf{i}_i, \tag{4.30}$$

$$L\frac{d}{dt}\mathbf{i}_l(t) = A_l^T\mathbf{v}(t), \tag{4.31}$$

where the inductor equation has been explicitly reintroduced to the system without an integral. The output of this system is the vector whose entries are the voltage gains across each current sources given by

$$A_i^T\mathbf{v}(t) = \mathbf{v}_i(t).$$

This system of integro-differential-algebraic equations comprises a concise description of the RCL circuit equations.

The final step is to translate these circuit equations into a form directly amenable to the SEA algorithm for matrices in case two form. Define the state-space vector, system input and system

output as

$$\mathbf{x}(t) = \begin{bmatrix} \mathbf{v}(t) \\ \mathbf{i}_l(t) \end{bmatrix}, \qquad \mathbf{u}(t) = -\mathbf{i}_i(t) \qquad \mathbf{y}(t) = \mathbf{v}_i(t),$$

respectively. The corresponding linearization matrices for this system are given by

$$\mathcal{E} = \begin{bmatrix} E_{11} & 0 \\ 0 & L \end{bmatrix}, \qquad \mathcal{A} = -\begin{bmatrix} A_{11} & \mathcal{A}_l \\ -\mathcal{A}_l^T & 0 \end{bmatrix}, \qquad \mathcal{B} = \begin{bmatrix} A_i \\ 0 \end{bmatrix}, \qquad \mathcal{L} = \mathcal{B}^T. \tag{4.32}$$

Notice that since C, R, and L are symmetric positive semi-definite, with

$$\mathcal{E} \succeq 0 \qquad \text{and} \qquad \mathcal{A} + \mathcal{A}^T \succeq 0. \tag{4.33}$$

The RCL circuit equations can then be written as a descriptor system of first-order linear time-invariant DAEs in state-space form (2.19). Applying the Laplace transform with the assumption of zero initial condition to this system, the associated transfer function (1.4) is given by

$$H(s) = \mathcal{B}^T (s\mathcal{E} - \mathcal{A})^{-1} \mathcal{B}. \tag{4.34}$$

The factorized matrix pencil

$$s\mathcal{E} - \mathcal{A} = \begin{bmatrix} Q(s) & A_l \\ 0 & sL \end{bmatrix} \begin{bmatrix} I_{n_0} & 0 \\ -\frac{1}{s} L^{-1} A_l^T & I_{n_l} \end{bmatrix} \tag{4.35}$$

is regular if, and only if, the matrix pencil$Q(s) = sE_{11} + A_{11} + \frac{1}{s} A_l L^{-1} A_l^T$ is regular. Also, the transfer function $H(s)$ can be written using one of two equivalent forms

$$\mathcal{B}^T (s\mathcal{E} - \mathcal{A})^{-1} \mathcal{B} = A_i^T \big(sE_{11} + A_{11} + \frac{1}{s} A_l L^{-1} A_l^T \big)^{-1} A_i.$$

Assuming $s_0 \in \mathbb{C}$ is chosen such that the matrix pencil $Q(s_0) = Q_0$ is invertible, notice

$$\mathcal{M} = (s_0\mathcal{E} - \mathcal{A})^{-1}\mathcal{E} = \begin{bmatrix} I_{n_0} & 0 \\ \frac{1}{s_0}L^{-1}A_l^T & I_{n_l} \end{bmatrix} \begin{bmatrix} Q_0^{-1} & -\frac{1}{s_0}Q_0^{-1}A_lL^{-1} \\ 0 & \frac{1}{s_0}L^{-1} \end{bmatrix} \begin{bmatrix} E_{11} & 0 \\ 0 & L \end{bmatrix}, \tag{4.36}$$

$$\mathcal{R} = (s_0\mathcal{E} - \mathcal{A})^{-1}\mathcal{B} = \begin{bmatrix} I_{n_0} & 0 \\ \frac{1}{s_0}L^{-1}A_l^T & I_{n_l} \end{bmatrix} \begin{bmatrix} Q_0^{-1} & -\frac{1}{s_0}Q_0^{-1}A_lL^{-1} \\ 0 & \frac{1}{s_0}L^{-1} \end{bmatrix} \begin{bmatrix} A_i \\ 0 \end{bmatrix}. \tag{4.37}$$

The appropriate data matrices in case two form can be read from this factorization for input into the SEA algorithm.

4.5.2 Properties of SEA-Based Dimension Reduction of RCL Circuits

The Structure Exploiting Arnoldi algorithm for matrices in case two form can be used to produce reduced-order models of systems of first-order integro DAEs arising from RCL circuit equations. By translating a given RCL circuit into the matrices in case two form as suggested above, the corresponding model order reduction technique relies on the (band) SEA algorithm for matrices in case two form to generate an associated reduced-order.

Algorithm 18 (Band) SEA-Based Dimension Reduction for RCL Circuit Equations

Input: An RCL circuit with only current sources given in linearized state-space form
Output: An equivalent reduced-order system

1. Select $s_0 \in \mathbb{C}$ s.t. $Q_0 = Q(s_0)$ nonsingular and choose reduction dimension n.
2. Run $n-1$ steps of the SEA algorithm with
$$\begin{bmatrix} C^{(1)} \\ C^{(2)} \end{bmatrix} = \begin{bmatrix} I_{n_0} \\ -\frac{1}{s_0}L^{-1}A_l^T \end{bmatrix}, \qquad \begin{bmatrix} M^{(1)} & M^{(2)} \end{bmatrix} = \begin{bmatrix} Q_0^{-1}E_{11} & Q_0^{-1}A_l \end{bmatrix} \qquad s_1 = 0, \qquad s_2 = \frac{1}{s_0}.$$
and produce orthonormal basis Q_{k_n} and V_{σ_n}.
3. Obtain the reduced order model for the original system by projection:
$$\widetilde{E}_{11} = Q_{k_n}^T E_{11} Q_{k_n}, \quad \widetilde{L} = V_{\sigma_n}^T L V_{\sigma_n}, \quad \widetilde{A}_{11} = Q_{k_n}^T A_{11} Q_{k_n}, \quad \widetilde{A}_l = Q_{k_n}^T A_l V_{\sigma_n}, \quad \widetilde{A}_i = Q_{k_n}^T A_i.$$

As before, the matrices $M^{(1)}$ and $M^{(2)}$ are never formed explicitly in the implementation of (band) SEA-based reduction. Instead, by calculating a sparse LU factorization

$$Q_0 = Q(s_0) = P^T L\, U\, Q^T,$$

the matrix-vector multiplication with $F = \begin{bmatrix} M^{(1)} & M^{(2)} \end{bmatrix}$ proceeds as expected.

Assume that the (band) SEA algorithm runs for $n-1$ iterations and the output matrix Q_k and V_σ are orthonormal basis for the appropriate spaces. Let

$$\mathcal{V}_n = \begin{bmatrix} Q_k & 0 \\ 0 & V_\sigma \end{bmatrix} \in \mathbb{C}^{N \times n_1}. \tag{4.38}$$

Then $K(\mathcal{M}, \mathcal{R}, n) \subseteq \text{range}\,(\mathcal{V}_n)$ and the reduced-order models formed via projection onto $\mathcal{V}_n$ (2.30) results in a Padé-type model of the original system by Theorem 2.4.1. If the expansion point s_0 is chosen such that $s_0 \in \mathbb{R}$ (a real number), then SEA models match twice as many moments as is guaranteed by Theorem 2.4.1. In most applications, s_0 is chosen as a real number to avoid complex arithmetic.

Theorem 4.5.1. Let $s_0 \in \mathbb{C}$ and let $\mathcal{A}, \mathcal{E}, \mathcal{B}, \mathcal{L}$ and $\mathcal{D}$ be the linearization matrices for the system of first-order integro-DAEs associated with an RCL circuit (4.32). Let $\mathcal{M}$ and $\mathcal{R}$ be matrices in case two form (4.36) and suppose $\mathcal{V}_n$ (4.38) is generated using the specialized band SEA algorithm 17. Let $n = n(j) = m_1 + m_2 + \cdots + m_j$ for some $1 \leq j \leq k_0$ (2.4). Then, the first $2j$ moments in the expansions of $H(s)$ (4.34) and $H_n(s)$ (1.9) are identical with

$$H_n(s) = H(s) + \mathcal{O}\big((s - s_0)^{2j}\big).$$

The proof of this result follows using the same logical arguments as the proof of Theorem 3 in the SPRIM paper [**Fre04**] which essentially demonstrates that the linearization matrices associated with RCL circuits exhibit J-Hermitian structure [**Fre08**]. The fact that the basis $\mathcal{V}_n$ produced by the SEA algorithm maintains the block structure of $\mathcal{E}$ and $\mathcal{A}$ through reduction follows because

$$\mathcal{V}_n^T \mathcal{E} \mathcal{V}_n = \begin{bmatrix} \widetilde{E}_{11} & 0 \\ 0 & \widetilde{L} \end{bmatrix}, \qquad \mathcal{V}_n^T \mathcal{A} \mathcal{V}_n = - \begin{bmatrix} \widetilde{A}_{11} & \widetilde{A}_l \\ -\widetilde{A}_l^T & 0 \end{bmatrix}, \qquad \mathcal{V}_n^T \mathcal{B} = \begin{bmatrix} \widetilde{A}_i \\ 0 \end{bmatrix}.$$

This structure preservation is a crucial property of SEA-based reduced-order modeling that guarantees a higher moment-matching property than the analogous (band) Arnoldi reduction algorithm 5.

In addition to higher accuracy, these projection-based reduced-order models maintain important system characteristics of RCL circuits such as passivity and reciprocity. It is well known that a

linear dynamical system is passive if the transfer function $H(s)$ associated with the system is *positive real* [**AV06**]. Positive real transfer functions satisfy the following three properties

i. $H(s)$ has no poles in $\mathbb{C}_+ = \{s \in \mathbb{C} |\ \text{Re}\ (s) > 0\}$

ii. For all $s \in \mathbb{C}$, $H(\overline{s}) = \overline{H(s)}$

iii. For all $s \in \mathbb{C}_+$ and $\mathbf{x} \in \mathbb{C}^m$, $\text{Re}\ (\mathbf{x}^H H(s)\mathbf{x}) \geq 0$

Transfer functions of SEA-inspired reduced-order model are positive real and the corresponding systems are passive.

Theorem 4.5.2. Suppose the assumptions of Theorem 4.5.1 hold. Then, the SEA-based reduced-order models is passive.

The proof to this theorem can be found in the literature on passive reduced-order modeling [**Fre99**]. Due to the structure of the linearization matrices, the transfer function associated with this system can be stated in the form

$$\mathcal{B}_n^T(s_0\mathcal{E}_n - \mathcal{A}_n)^{-1}\mathcal{B}_n = \widetilde{A}_i^T\big(s\widetilde{E}_{11} + \widetilde{A}_{11} + \frac{1}{s}\widetilde{A}_l\widetilde{L}^{-1}\widetilde{A}_l^T\big)^{-1}\widetilde{A}_i,$$

as can be easily observed from a factorization of matrix pencil $s\mathcal{E}_n - \mathcal{A}_n$ analogous to the factorization of $s\mathcal{E} - \mathcal{A}$ (4.35).

4.5.3 Numerical Experiments

The final subsection of this chapter presents numerical results comparing the accuracy of SEA-based reduced-order modeling with SPRIM-based reduced-order modeling. These simulations are meant to demonstrate the basic properties of the proposed (band) SEA-based reduction technique. Numerical implementations of the algorithms used for each experiment are based on the pseudocode presented in this thesis.

Example 4.5.3. This example illustrates numerical results for SEA and SPRIM applied to a finite-element model of a shaft. This is one of the many systems that can be reduced by recognizing the equivalence between RCL circuits and mechanical systems [**Bro07**] such as in the modeling and simulation of MEMS devices. The original single-input ($m = 1$), single-output ($p = 1$) system of first-order integro-DAEs is such that $n_1 = 400 = n_2$ and $N = 800$. Bode plots of the frequency

response for the original system (black), the SEA-based reduced model (blue) and the SPRIM-based reduced model (green) are given in the left column of Table 4.5.3.

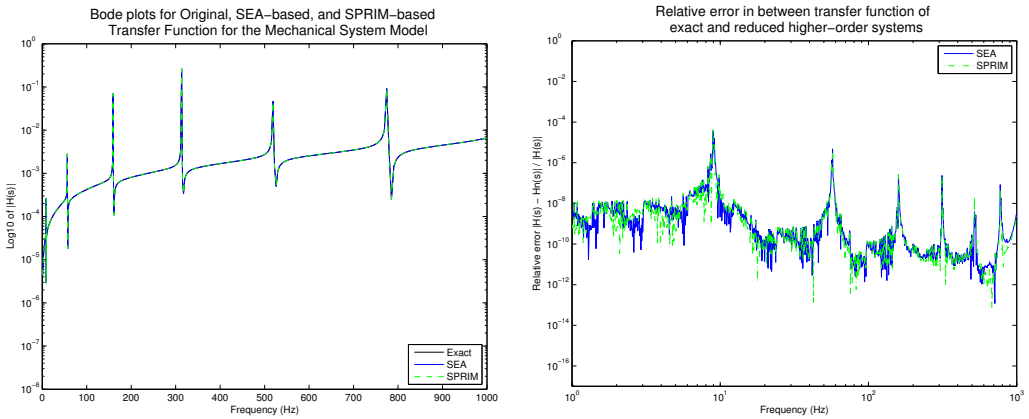

TABLE 4.5.1. Mechanical system model: The left entry of this table presents Bode plots of the mechanical system model for the original system (black), SEA-based reduced-order model (blue), and SPRIM-based reduced-order model (green) for reduced dimension $n = 20$. The right entry of this table includes the relative error for the SEA-based model (blue) and the SPRIM-based model (green).

The reduction dimension used for these models is $n = 15$ and the selected expansion point is $s_0 = \pi \times 10^3$. The relative errors are shown in the right column of Table 4.5.3.

Example 4.5.4. The numerical results presented in this second example are generated applying both SEA- and SPRIM-based reduced-order modeling to a PEEC discretization [**Rue74**] of an electromagnetic problem. This RCL circuit consists of a single resistive source that drives the circuit, 2100 capacitors and 6990 inductive couplings. The circuit is formulated as a multiple-input ($m = 2$), multiple-output ($p = 2$) system of first-order integro-DAEs is such that $n_1 = 136$, $n_2 = 172$ and $N = 308$. Bode plots of the frequency response for the original system (black), the SEA-based reduced model (blue) and the SPRIM-based reduced model (green) are given in Table 4.5.4.

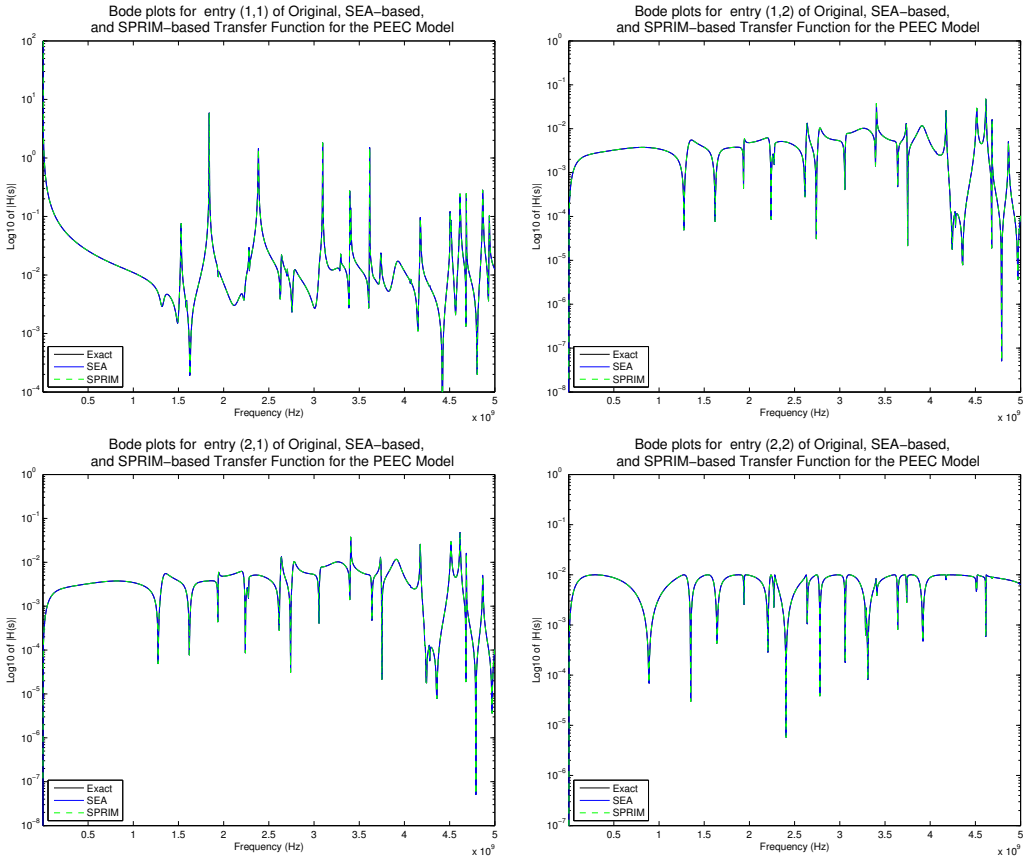

TABLE 4.5.2. PEEC circuit model: Entry (i, j) of this table presents Bode plots of the mechanical system model for entry (i, j) of the transfer function associated with the original system (black), the SEA-based reduced-order model (blue), and the SPRIM-based reduced-order model (green) for reduction dimension $n = 136$.

As is evident, a reduction dimension of $n = 136$ is sufficient to match each entry of the 2×2 transfer function over the frequency range of interest. The selected expansion point is $s_0 = 2\pi \times 10^9$ in this case. The corresponding relative errors for each entry of the transfer function are shown in the right column of Table 4.5.4.

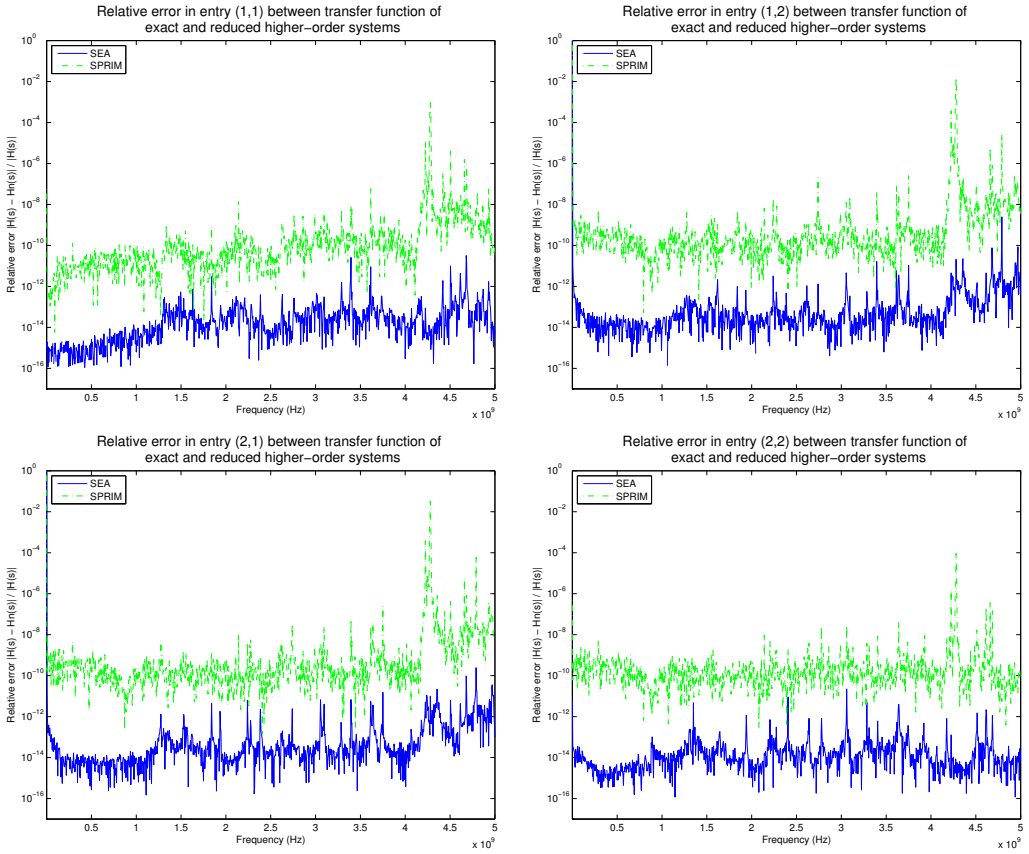

TABLE 4.5.3. PEEC Circuit model: Entry (i, j) of this table presents the relative error between entry (i, j) of the exact transfer function and entry (i, j) of the transfer function arising from the SEA-based reduced-order model (blue) as well as the relative error between entry (i, j) of the exact transfer function and entry (i, j) of the transfer function arising from the SPRIM-based reduced-order model (green) for reduction dimension $n = 136$.

4.6 Conclusions

Chapter 4 develops the Structure Exploiting Arnoldi algorithm for matrices in case two form as an alternative to the SPRIM algorithm for reduced-order modeling of RCL circuits with only current sources. Table 4.6.1 demonstrates that SEA-based models can be generated at a fraction of the cost of the corresponding SPRIM-based models. Each data point in the table below represents the single execution of the corresponding block Krylov subspace technique used to generate the graphs in Section 4.5. These run times, measured in seconds, focus on the block Krylov subspace technique and do not consider data loading, pre-processing, post-processing, nor are any necessary

projections used to form the reduced-order models considered. It is important to note that the post-processing required by the SPRIM algorithm necessary to generate the proper orthonormal bases is not included in this comparison. No such post-processing is necessary for SEA-based dimension reduction, offering additional savings than those suggested below. A machine with an Intel Core 2 Duo processor and 1GB RAM was used to generate these results.

TABLE 4.6.1. Run Time Comparison (in Seconds) between SEA and SPRIM

	SEA	**SPRIM**
Mechanical System Model	0.05	0.13
PEEC Model	0.40	0.47

The generic single-input and band SEA algorithms for matrices in case two form give templates for creating basis for the multiple copied subspace associated with block Krylov subspaces induced by matrices in case two form. In order to utilize these techniques for reduced-order modeling of large-scale RCL circuits, specific adaptations to the general algorithms are made to reflect the more specialized structure of linearized first-order integro-DAEs. These adaptations are available in both single-input and band versions and improve computational efficiency resulting in the speed up offered by SEA-based reduced-order modeling compared with the SPRIM based models. Table 4.6.2 illustrates that the properties of SEA-based reduced-order models match SPRIM-based models.

TABLE 4.6.2. Model Reduction Algorithm Comparison

	SEA	**SPRIM**	**Arnoldi**
Provably Passive	✓	✓	✓
# Moments Match	2j	2j	j
Higher-Order Realization	✓	✓	×

CHAPTER 5

Summary and Future Work

The main focus of this thesis is to establish a precedent for executing model order reduction by exploiting the structure of the block Krylov subspaces associated with linearization matrices of general higher-order linear dynamical systems. The popular method to reduce such systems is to linearize the general higher-order system and use well-known Krylov subspace-based moment-matching dimension reduction techniques. This reduction strategy results in increased computational complexity from dimension augmentation and loss of system properties. However, the SEA algorithms presented in this thesis address these weaknesses by exploring the special structure of these block Krylov subspaces.

Chapter 3 presented the Structure Exploiting Arnoldi algorithm as a reduction technique for ℓth-order linear dynamical systems. This is the first completely general study of algorithms to exploit the structure of such systems. Chapter 4 introduced an analogous Structure Exploiting Arnoldi algorithm for special second-order systems of first-order integro-DAEs. These algorithms process data of the same size as the original general higher-order system, addressing the challenge of the dimension augmentation when processing higher-order systems. Another nice feature of SEA-based model order reduction is the preservation of system properties. The results demonstrated in this thesis indicate the viability of this type of reduction strategy.

One natural question related to this research is to identify the the spectral properties of matrices in case one form and matrices in case two form. A study of the spectral properties of matrices in case one form may shed light on applications of the SEA algorithm to polynomial eigenvalue problems. It seems very likely that the eigenvalues of matrices in case one form can be related to polynomial eigenvalues arising from a given matrix polynomial using the same linearization matrices discussed in Chapter 2.

Other avenues of research should focus on adapting the generic band SEA algorithm to improve the numerical properties of SEA-based reduced-order modeling. In the numerical simulations presented in this thesis, the basis matrix $\widehat{A}_n$ for $K(\mathcal{M}, \mathcal{R}, n)$ generated in the band SEA algorithm was relatively ill-conditioned compared with the basis generated by the band Arnoldi algorithm. Because columns of $\widehat{A}_n$ are not explicitly used to generate the SEA-based reduced-order models, some flexibility in the updates for these column vectors may be permissible. For example, partial orthogonalization of updates for $\widehat{A}_n$ or perhaps a projection based on three-term recurrences used in Chebyshev polynomial methods may be viable alternatives. These would decrease the memory costs for the SEA algorithm and may maintain desirable accuracy results.

Bibliography

[ABB+99] E. Anderson, Z. Bai, C. Bischof, L. S. Blackford, J. Demmel, J. J. Dongarra, J. Du Croz, S. Hammarling, A. Greenbaum, A. McKenney, and D. Sorensen, *LAPACK Users' Guide, Third Edition*, Society for Industrial and Applied Mathematics, 1999.

[ABFH00] J. I. Aliaga, D. L. Boley, R. W. Freund, and V. Hernandez, *A Lanczos-type method for multiple starting vectors*, Mathematics of Computation (2000), 1577–1601.

[Arn51] W. Arnoldi, *The principle of minimized iterations in the solutions of the matrix eigenvalue problem*, Quarterly of Applied Mathematics **19** (1951), 17–29.

[ASG01] A. C. Antoulas, D. C. Sorensen, and S. Gugercin, *A survey of model reduction methods for large-scale systems*, Contemporary Mathematics **280** (2001), 193–219.

[AV06] B. Anderson and S. Vongpanitlerd, *Network Analysis and Synthesis: A Modern Systems Theory Approach*, Dover Publications, 2006.

[BGM10] G. Baker and P. Graves-Morris, *Padé Approximants*, Encyclopedia of Mathematics and its Applications, Cambridge University Press, 2010.

[BHtM11] P. Benner, M. Hinze, and E. J. W. ter Maten., *Model Reduction for Circuit Simulation*, Lecture Notes in Electrical Engineering, Springer, 2011.

[BMS05] P. Benner, V. Mehrmann, and D. Sorensen, *Dimension Reduction of Large-Scale Systems: Proceedings of a Workshop Held in Oberwolfach, Germany, October 19-25, 2003*, Lecture Notes in Computational Science and Engineering, Springer, 2005.

[Bro07] B. Brogliato, *Dissipative Systems Analysis and Control: Theory and Applications*, Communications and Control Engineering, Springer, 2007.

[BS05a] Z. Bai and Y. Su, *Dimension reduction of large-scale second-order dynamical systems via a second-order Arnoldi method*, SIAM Journal on Scientific Computing **26** (2005), 1692–1709.

[BS05b] Z. Bai and Y. Su, *Soar: A second-order Arnoldi method for the solution of the quadratic eigenvalue problem*, SIAM Journal on Matrix Analysis and Applications. **26** (2005), 640–659.

[CN94] E. Chiprout and M. Nakhla, *Asymptotic Waveform Evaluation: And Moment Matching for Interconnect Analysis*, Kluwer Academic Publishers, 1994.

[CP02] J. Clark and K. Pister, *Addressing the needs of complex MEMS design*, Fifteenth IEEE International Conference on Micro Electro Mechanical Systems, Las Vegas, NV, 2002, pp. 204–209.

[CPO02] M. Celik, L. Pileggi, and A. Odabasioglu, *IC Interconnect Analysis*, Kluwer Academic Publishers, 2002.

[CV02] Y. Chahlaoui and P. Vandooren, *A collection of benchmark examples for model reduction of time-invariant dynamical systems*, SLICOT Working Note, 2002-2, 2002.

[Dat04] B. Datta, *Numerical Methods for Linear Control Systems*, Elsevier Academic Press, 2004.

[Dav06] T. Davis, *Direct Methods for Sparse Linear Systems*, Fundamentals of Algorithms, Society for Industrial and Applied Mathematics, 2006.

[Deo74] N. Deo, *Graph Theory with Applications to Engineering and Computer Science*, Prentice Hall, 1974.

[Don02] J. Dongarra, *Basic linear algebra subprograms technical forum standard*, International Journal of High Performance Applications and Super Computing **16** (2002), no. 1, 1–111.

[FF95a] P. Feldmann and R. W. Freund, *Reduced-order modeling of large linear subcircuits via a block Lanczos algorithm*, In Proceedings of the 32nd ACM/IEEE Design Automation Conference, 1995, pp. 474–479.

[FF95b] P. Feldmann and R. Freund, *Efficient linear circuit analysis by Padé approximation via the Lanczos process*, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems **14** (1995), no. 5, 639–649.

[FF96] R. Freund and P. Feldmann, *Reduced-order modeling of large passive linear circuits by means of the SyPVL algorithm*, In Technical Digest 1996 IEEE/ACM International Conference on Computer-Aided Design, IEEE Computer Society Press, 1996, pp. 280–287.

[FF97] ———, *The SyMPVL algorithm and its applications to interconnect simulation*, In Proceedings of 1997 International Conference on Simulation of Semiconductor Processes and Devices, IEEE, 1997, pp. 113–116.

[Fre99] R. W. Freund, *Passive reduced-order models for interconnect simulation and their computation via Krylov-subspace algorithms*, Design Automation Conference (1999), 195–200.

[Fre00] R. Freund, *Krylov-subspace modeling in circuit simulation*, Journal of Computational Applied Mathematics **123** (2000), 395–421.

[Fre03a] R. W. Freund, *Model reduction methods based on Krylov subspaces*, Acta Numerica **12** (2003), 267–319.

[Fre03b] R. W. Freund, *Padé-type model reduction of second-order and higher-order linear dynamical systems*, In Dimension Reduction of Large-Scale Systems, Benner P, Mehrmann V, Sorensen DC (eds), Lecture Notes in Computational Science and Engineering, Springer, 2003, pp. 191–223.

[Fre04] ———, *SPRIM: structure-preserving reduced-order interconnect macromodeling*, ICCAD, 2004, pp. 80–87.

[Fre05] R. W. Freund, *Krylov subspaces associated with higher-order linear dynamical systems*, BIT Numerical Mathematics **45** (2005), 495–516.

[Fre08] R. Freund, *On Padé-type model order reduction of j-hermitian linear dynamical systems*, Linear Algebra and Its Applications **429** (2008), 2451–2464.

[Fre11] R. W. Freund, *The SPRIM algorithm for structure-preserving order reduction of general RCL circuits*, Model Reduction for Circuit Simulation (P. Benner, M. Hinze, and E. J. W. ter Maten, eds.), Lecture Notes in Electrical Engineering, vol. 74, Springer Netherlands, 2011, pp. 25–52.

[Fri01] E. Friedman, *Clock distribution networks in synchronous digital integrated circuits*, Proceedings of the IEEE **89** (2001), no. 5, 665–692.

[GL96] G. Golub and C. V. Loan, *Matrix Computations, Third Edition*, John Hopkins University Press, 1996.

[GLR09] I. Gohberg, P. Lancaster, and L. Rodman, *Matrix Polynomials*, Classics in Applied Mathematics, Society for Industrial and Applied Mathematics, 2009.

[Gra81] A. Graham, *Kronecker Products and Matrix Calculus with Applications*, John Wiley and Sons, 1981.

[Hog06] L. Hogben, *Handbook of Linear Algebra*, Discrete Mathematics and Its Applications, Taylor & Francis, 2006.

[Hog13] ———, *Handbook of Linear Algebra, Second Edition*, Discrete Mathematics and Its Applications, Chapman and Hall/CRC, 2013.

[HRBR92] H. Heeb, A. Ruehli, J. Bracken, and R. Rohrer, *Three dimensional circuit oriented electromagnetic modeling for VLSI interconnects*, In Proceedings of IEEE 1992 International Conference on Computer Design: VLSI in Computers and Processors, 1992, pp. 218–221.

[JCP98] N. Z. J.V. Clark and K. Pister, *MEMS simulation using SUGAR v0.5*, Solid-State Sensors and Actuators Workshop, Hilton Head Island, SC, 1998, pp. 191–196.

[KGP94] S.-Y. Kim, N. Gopal, and L. Pillage, *Time-domain macromodels for VLSI interconnect analysis*, Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on **13** (1994), no. 10, 1257–1270.

[Lan50] C. Lanczos, *An iteration method for the solution of the eigenvalue problems of linear differential and integral operators*, Journal of Research of the National Bureau of Standards **45** (1950), 255–282.

[LSF11] P. Li, L. Silveira, and P. Feldmann, *Simulation and Verification of Electronic and Biological Systems*, Springer Netherlands, 2011.

[Mee08] K. Meerbergen, *The quadratic Arnoldi method for the solution of the quadratic eigenvalue problem*, SIAM Journal Matrix Analysis and Applications **30** (2008), no. 4, 1463–1482.

[OCP97] A. Odabasioglu, M. Celik, and L. T. Pileggi, *PRIMA: passive reduced-order interconnect macromodeling algorithm*, International Conference on Computer Aided Design, 1997, pp. 58–65.

[Ogr94] J. Ogrodzki, *Circuit Simulation Methods and Algorithms*, Electronic Engineering Systems Series, CRC Press, 1994.

[PR90] L. Pillage and R. Rohrer, *Asymptotic waveform evaluation for timing analysis*, IEEE Trans. Computer-Aided Design **CAD-9** (1990), 352–366.

[RBR92] V. Raghavan, J. Bracken, and R. Rohrer, *AWESpice: a general tool for the accurate and efficient simulation of interconnect problems*, In Proceedings of the 29th ACM/IEEE Conference on Design Automation, 1992, pp. 87–92.

[RP94] C. Ratzlaff and L. Pillage, *RICE: rapid interconnect circuit evaluation using AWE*, Computer-Aided Design of Integrated Circuits and Systems, IEEE **13** (1994), no. 6, 763–776.

[Rue74] A. E. Ruehli, *Equivalent circuit models for three-dimensional multiconductor systems*, Microwave Theory and Techniques, IEEE Transactions **22** (1974), no. 3, 216–221.

[Rue86] A. Ruehli, *Circuit Analysis, Simulation and Design, Part 1: General Aspects of Circuit Analysis and Design*, North-Holland, 1986.

[Saa92] Y. Saad, *Numerical Methods for Large Eigenvalue Problems*, Algorithms and Architectures for Advanced Scientific Computing Series, Manchester University Press, 1992.

[Saa03] ———, *Iterative Methods for Sparse Linear Systems*, Society for Industrial and Applied Mathematics, 2003.

[SC91] T. Su and R. Craig, *Model reduction and control of flexible structures using Krylov vectors*, J. Guid. Control Dyn. **14** (1991), 260–267.

[She96] P. Shephard, *Integrated Circuit Design, Fabrication and Test*, MacMillan Press LTD, 1996.

[She99] B. Sheehan, *ENOR: Model order reduction of RLC circuits using nodal equations for efficient factorization*, In Proceedings of the 36th Design Automation Conference, 1999, pp. 17–21.

[SVDVR08] W. Schilders, H. Van Der Vorst, and J. Rommes, *Model Order Reduction: Theory, Research Aspects and Applications*, European Consortium for Mathematics in Industry, Springer, 2008.

[VS83] J. Vlach and K. Singhal, *Computer Methods for Circuit Analysis and Design*, Van Nostrand Reinhold, 1983.

[Wat07] D. Watkins, *The Matrix Eigenvalue Problem: GR and Krylov Subspace Methods*, Society for Industrial and Applied Mathematics, 2007.