

Structural Preserving Model Reductions¹Ren-Cang Li²

January 2004

ABSTRACT

A general framework for structural preserving model reductions by Krylov subspace projection methods is developed. The goal is to preserve any substructures of importance in the matrices L, G, C, B that define the model prescribed by transfer function $H(s) = L^*(G + sC)^{-1}B$. As an application, quadratic transfer functions targeted by Su and Craig (*J. Guidance, Control, and Dynamics*, 14 (1991), pp. 260–267.) is revisited, which leads to an improved algorithm than Su’s and Craig’s original in terms of achieving the same approximation accuracy with smaller reduced systems. Other contributions include new Gram-Schmidt orthogonalization process and new Arnoldi process that only orthogonalize the prescribed portion of all basis vectors as opposing to whole vectors by existing counterparts. These new processes are designed as one way to numerically realize the idea in the general framework.

¹This report is available on the web at <http://www.ms.uky.edu/~rcli/>.

²Department of Mathematics, University of Kentucky, Lexington, KY 40506 (rcli@ms.uky.edu.) This work was supported in part by the National Science Foundation CAREER award under Grant No. CCR-9875201.

Structural Preserving Model Reductions

Ren-Cang Li *

January 2004

Abstract

A general framework for structural preserving model reductions by Krylov subspace projection methods is developed. The goal is to preserve any substructures of importance in the matrices L, G, C, B that define the model prescribed by transfer function $H(s) = L^*(G + sC)^{-1}B$. As an application, quadratic transfer functions targeted by Su and Craig (*J. Guidance, Control, and Dynamics*, 14 (1991), pp. 260–267.) is revisited, which leads to an improved algorithm than Su’s and Craig’s original in terms of achieving the same approximation accuracy with smaller reduced systems. Other contributions include new Gram-Schmidt orthogonalization process and new Arnoldi process that only orthogonalize the prescribed portion of all basis vectors as opposing to whole vectors by existing counterparts. These new processes are designed as one way to numerically realize the idea in the general framework.

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

Contents

1	Introduction	2
2	Projectors	5
3	Grimme's Moment Matching Theorem	6
4	Short Recurrences to Build X and Y Simultaneously	9
5	Moment Matching that Preserves Substructures	11
6	Structures of Krylov Subspaces of Block Matrices	14
7	Quadratic Transfer Functions	15
8	Moment Matching at $s_0 \neq 0$	19
9	Modified Gram-Schmidt Sub-orthogonalization Process	20
10	A Sub-orthogonalized Arnoldi Process	23
11	Issues with Sub-orthogonalization Process	26

1 Introduction

Krylov subspace projection methods are increasingly popular in model reductions owing to their numerical efficiency for very large systems from, e.g., dynamical systems and circuit simulations. Recent survey articles Antoulas and Sorensen [3], Bai [6], and Freund [17] provided in depth review of the subject and comprehensive references. Roughly speaking, these methods project the original system onto a smaller subspace to arrive at a (much) smaller system having properties, among others, that many leading terms (called *moments*) of the associated (matrix-valued) transfer function expanded at given points for the original and reduced systems match.

Consider time-invariant *multi-input multi-output* (MIMO) linear dynamical system

$$\begin{cases} C\dot{x}(t) + Gx(t) &= Bu(t), \\ y(t) &= L^*x(t), \end{cases} \quad (1.1)$$

which gives rise the following transfer function

$$H(s) = L^*(G + sC)^{-1}B, \quad (1.2)$$

upon applying Laplace transformation with zero initial conditions, where $G, C \in \mathbb{C}^{n \times n}$, $B \in \mathbb{C}^{n \times m}$, $L \in \mathbb{C}^{n \times p}$, $D \in \mathbb{C}^{p \times m}$, $x(t)$, $u(t)$, and $y(t)$ are vector-valued functions of apt dimensions. When $m = p = 1$, it is called a *single-input single-output* (SISO) system. The

reader is referred to, e.g., [2] and the survey papers mentioned above, for more background information. Assuming G is nonsingular, we define associated moments as

$$M_i \stackrel{\text{def}}{=} L^*(G^{-1}C)^i G^{-1}B \equiv L^*G^{-1}(CG^{-1})^i B, \quad i = 0, 1, 2, \dots, \quad (1.3)$$

which provides approximations to $H(s)$ upon expanding it at $s = 0$

$$H(s) = \sum_{i=0}^{\infty} (-1)^i s^i M_i.$$

In today's applications of interests, e.g., VLSI designs and structural dynamics [3, 6, 17], n can be quite big – up to hundreds of millions [6, 17] while p and m are quite modest. Approximations to $H(s)$ usually have to be done through some kind of reduction on L, G, C, B [6, 17, 40]. Let $X, Y \in \mathbb{C}^{n \times k}$ such that Y^*GX is nonsingular (and thus $\text{rank}(X) = \text{rank}(Y) = k$). We may reduce the transfer function to

$$H_R(s) = L_R^*(G_R + sC_R)^{-1}B_R, \quad (1.4)$$

where

$$L_R = X^*L, \quad G_R = Y^*GX, \quad C_R = Y^*CX, \quad B_R = Y^*B, \quad (1.5)$$

for which associated moments are

$$M_{Ri} \stackrel{\text{def}}{=} L_R^*(G_R^{-1}C_R)^i G_R^{-1}B_R \equiv L_R^*G_R^{-1}(C_R G_R^{-1})^i B_R, \quad i = 0, 1, 2, \dots \quad (1.6)$$

There are various techniques to pick X and Y to perform reductions. Among them Krylov subspace-based model-reductions are getting much of the attention. The idea is to match as many M_i to M_{Ri} as possible from $i = 0$ and forward, and if $M_i = M_{Ri}$ for $0 \leq i \leq p$, then $H(s) = H_R(s) + \mathcal{O}(s^{p+1})$.

Up until Grimme¹ [25, 1997], and Odabasioglu, Celik, and Pileggi [31, 1998], Krylov subspace based model reductions focused mostly on a transfer function with $G = I_n$ (the $n \times n$ identity matrix) for which $Y^*X = I_k$, with one exception Su and Craig [38, 1991] which treated quadratic transfer function (see (7.1)) but Su's and Craig's paper was somehow overlooked until recently [6, 17]. The more general form (1.2) is treated by rewriting it as, e.g.,

$$H(s) = L^*(I + sG^{-1}C)^{-1}(G^{-1}B). \quad (1.7)$$

By doing so $G^{-1}C$ is projected as one whole matrix, unlike above where G and C are projected separately. See [14, 20, 34]. The potential of (1.5) as to structural preserving model reductions was not exploited as we shall do in this paper. There are advantages and disadvantages associated with the two different ways to do projections. The advantage would be some computational saving because the projection of $G^{-1}C, Y^*G^{-1}CX$, is often computed while bases of Krylov subspaces are being built, regardless of how G and C are

¹Even though Grimme formulated his reduction framework as in (1.5), he did it more for the convenience than anything else and his discussion [25, Table 2.2 on p.32] still centered around the transformed $H(s)$ such as in (1.2).

projected: separately or as a whole in terms of $G^{-1}C$, and so extra work is needed if G and C are projected separately. The disadvantages include loss of structures in G and C , e.g., symmetry, and possibly stabilities inherent in the original systems. One prime example would be PRIMA [31] which preserves an important stability property – *passitivity* of the original system from LRC circuits. Recent trend inclines more to project G and C separately [9, 16, 31, 38] because of its capability of structural preservations.

As for the case $G = I$ (and thus $Y^*X = I_k$), Villemagne and Skeltona [40, 1987] gave a thorough study on the conditions of X and Y under which the number of moments match, and arrived at many results some new and some old dated back as early as [24, 1974]. These results consist of the foundation of so-called PvL of Feldman and Freund [14] and a similar algorithm of Gallivan, Grimme, and van Dooren [20] to provide a more stable implementation of Asymptotic Waveform Evaluation of Pillage and Rohrer [34]. For general nonsingular $G \neq I$, a general theory was developed by Grimme [25]. This paper will exploit Grimme’s theorem to design algorithms that preserve substructures in the matrices L, G, C, B for the first time. As an important application this paper also considers the substructures inherited from linearizing a quadratic transfer function and the application leads to an improved algorithm of Su and Craig [38] in terms of X ($Y = X$) with fewer columns for the same approximate accuracy and thus the same approximation quality with smaller reduced systems. Along the way we offer implementation guidelines to efficiently compute X and Y under various circumstances, including a new Gram-Schmidt orthogonalization process that only orthogonalize some prescribed portions of all vectors and an Arnoldi process that has only some prescribed portions of all vectors orthogonalized. The new Arnoldi process is a direct generalization of Su-Craig algorithm [38] but with deflations; see also Bai and Su [9].

The idea of this paper is extensible to cases when $H(s)$ has to be expanded at some $s_0 \neq 0$, or more generally at a few distinct points simultaneously. See Section 8 for more detail.

The rest of this paper is organized as follows. Section 2 defines two projectors that will be used to prove Grimme’s theorem in Section 3 in a more elegant way. Grimme’s theorem is the theoretical foundation for us later to derive structure preserving model reductions. Section 4 discusses cases when and how X and Y can be computed simultaneously by some short-term recurrences which were not addressed by Grimme [25]. A general framework to design structure preserving model reductions is given in Section 5, and its application to quadratic transfer functions is in Section 7 with the help of the inherent structural properties of Krylov subspaces for certain block matrices, first exploited by Su and Craig [38]. Sections 9 – 11 presents our new sub-orthogonalization processes that can be more efficient than the corresponding Gram-Schmidt and Arnoldi counterparts, at a price of possibly numerical instability at times.

Notation. \mathbb{C} , \mathbb{C}^n , and $\mathbb{C}^{m \times n}$ are the sets of complex numbers, column vectors of dimension n , and $m \times n$ complex matrices, respectively. $I_n \in \mathbb{C}^{n \times n}$ is the identity matrix, and sometimes simply I when its dimension can be judged from the context. Unless otherwise explicitly stated, capital letters are matrices, while lower case letters are vectors or scalars. X^* is the complex conjugate transpose of matrix X , $\text{span}(X)$ is the subspace spanned by the columns of X with dimension $\dim(\text{span}(X)) = \text{rank}(X)$, the rank of X . If

span takes more than one argument matrices, it is the subspace spanned by all the columns of those matrices. For scalar α , $\bar{\alpha} \stackrel{\text{def}}{=} \alpha^*$. Throughout the rest, notation introduced from (1.1) to (1.6) will have the assignments, except in Section 2 where we reuse some symbols above to explain facts of a projector that are true beyond the context in this section, but the reuse fits nicely in the later sections when those symbols do assume their assignments here.

Let A be $n \times n$, and let Z be $n \times \ell$. The k th Krylov subspace generated by A on Z is defined to be

$$\mathcal{K}_k(A, Z) = \text{span}(Z, AZ, \dots, A^{k-1}Z). \quad (1.8)$$

For convenience, when $k = 0$, define $\mathcal{K}_0(A, Z) = \emptyset$, an empty-subspace.

2 Projectors

The use of projection technique is an ancient and powerful idea. It is an indispensable tool in studying infinite dimensional operator theory, but one may be able to get around of it in finite dimensional cases through matrix manipulations, e.g., the moment matching proofs in [7, 10, 16, 31, 38, 41]. However, using projection language can turn those proofs into much more elegant mathematical arguments as already clear in [40].

Any matrix (operator) P that satisfies $P^2 = P$ is a projector onto $\text{span}(P)$. Let $k = \text{rank}(P)$. We have (e.g., through its singular value decomposition (SVD) [13, 23])

$$P = X\Sigma Y^*,$$

where $X, Y \in \mathbb{C}^{n \times k}$ with $\text{rank}(X) = \text{rank}(Y) = k$, and $\Sigma \in \mathbb{C}^{k \times k}$ nonsingular. $P^2 = P$ implies Y^*X is nonsingular and $\Sigma = (Y^*X)^{-1}$. Then

$$P = X(Y^*X)^{-1}Y^*. \quad (2.1)$$

It can be verified that any matrix of this form satisfies $P^2 = P$. Thus (2.1) gives all projectors of rank k . It can also be seen that P projects onto $\text{span}(X) = \text{span}(P)$ which means $Px = x$ for any $x \in \text{span}(X)$ and especially $PX = X$. On the other hand, P^* projects onto $\text{span}(Y)$ and thus $y^*P = y^*$ for any $y \in \text{span}(Y)$ and especially $Y^*P = P$.

The following two seemingly more general but equivalent forms for projectors of rank k :

$$P = X(Y^*GX)^{-1}Y^*G, \quad (P \text{ projects onto } \text{span}(X)) \quad (2.2)$$

$$Q = GX(Y^*GX)^{-1}Y^*, \quad (Q^* \text{ projects onto } \text{span}(Y)) \quad (2.3)$$

will be useful in the sequel. Here G is an $n \times n$ matrix such that Y^*GX is nonsingular. Notice that both P in (2.2) and Q in (2.3) become the P in (2.1) when $G = I$. The introduction of G will become handy for us later. The following lemma is well-known.

Lemma 2.1 *Let projectors P and Q be defined as in (2.2) and (2.3). Then*

$$Px = x, \quad y^*Q^* = y^*, \quad \text{for any } x \in \text{span}(X) \text{ and } y \in \text{span}(Y).$$

3 Grimme's Moment Matching Theorem

Let notation from (1.1) to (1.6) have their assignments there. Grimme [25] developed a general theorem, Theorem 3.1, governing the number of moments matched. We shall give a different proof using the projectors P and Q in Section 2. The Keys are Lemmas 3.1 and Lemma 3.2 which seem to be new. The proof seems a bit more concise. Grimme's proof was not explicitly formulated in terms of P and Q , but rather implicitly.

Lemma 3.1 *Let projector P be defined as in (2.2). Then*

$$XG_{\mathbb{R}}^{-1}B_{\mathbb{R}} = PG^{-1}B, \quad XG_{\mathbb{R}}^{-1}C_{\mathbb{R}} = PG^{-1}C \cdot X. \quad (3.1)$$

Together they imply

$$X(G_{\mathbb{R}}^{-1}C_{\mathbb{R}})^i G_{\mathbb{R}}^{-1}B_{\mathbb{R}} = (PG^{-1}C)^i \cdot PG^{-1}B \quad \text{for } i \geq 0.$$

Proof: Notice (1.5), we have

$$\begin{aligned} XG_{\mathbb{R}}^{-1}B_{\mathbb{R}} &= \underbrace{X(Y^*GX)^{-1} \cdot Y^*G}_{PG^{-1}} G^{-1}B \\ &= PG^{-1}B, \\ XG_{\mathbb{R}}^{-1}C_{\mathbb{R}} &= \underbrace{X(Y^*GX)^{-1} \cdot Y^*G}_{PG^{-1}C} G^{-1}CX \\ &= PG^{-1}C \cdot X, \end{aligned}$$

as expected. ■

Lemma 3.2 *Let projector Q be defined as in (2.3). Then*

$$L_{\mathbb{R}}^* G_{\mathbb{R}}^{-1} Y^* = L^* G^{-1} Q, \quad C_{\mathbb{R}} G_{\mathbb{R}}^{-1} Y^* = Y^* \cdot C G^{-1} Q. \quad (3.2)$$

Together they imply

$$L_{\mathbb{R}}^* G_{\mathbb{R}}^{-1} (C_{\mathbb{R}} G_{\mathbb{R}}^{-1})^i Y^* = L^* G^{-1} Q \cdot (C G^{-1} Q)^i \quad \text{for } i \geq 0.$$

Proof: Notice that projector P in (2.2) gives rise to projector Q in (2.3) under substitutions

$$X \rightarrow Y, \quad Y \rightarrow X, \quad G \rightarrow G^*, \quad (3.3)$$

followed by action $(\cdot)^*$ (taking conjugate transpose). In view of this, with two more substitutions

$$B \rightarrow L, \quad C \rightarrow C^*, \quad (3.4)$$

in addition to those in (3.3), we obtain (3.2) by Lemma 3.1. ■

REMARK 3.1 It is worth noting that Lemmas 3.1 and 3.2 assume only the invertibility of Y^*GX , and they hold regardless of whatever $\text{span}(X)$ and $\text{span}(Y)$ are.

Theorem 3.1 (Grimme) *Let integers $q, r \geq 0$. If*

$$\mathcal{K}_q(G^{-1}C, G^{-1}B) \subseteq \text{span}(X), \quad \mathcal{K}_r(G^{-*}C^*, G^{-*}L) \subseteq \text{span}(Y), \quad (3.5)$$

then

$$X(G_{\text{R}}^{-1}C_{\text{R}})^i G_{\text{R}}^{-1}B_{\text{R}} = (G^{-1}C)^i G^{-1}B \quad \text{for } 0 \leq i \leq q-1, \quad (3.6)$$

$$L_{\text{R}}^* G_{\text{R}}^{-1}(C_{\text{R}} G_{\text{R}}^{-1})^j Y^* = L^* G^{-1}(CG^{-1})^j \quad \text{for } 0 \leq j \leq r-1, \quad (3.7)$$

$$M_i = M_{\text{R}i} \quad \text{for } 0 \leq i \leq q+r-1. \quad (3.8)$$

Proof: Let projectors P and Q be defined as in (2.2) and (2.3). Then (3.5) and Lemma 2.1 yield

$$P(G^{-1}C)^i G^{-1}B = (G^{-1}C)^i G^{-1}B \quad \text{for } 0 \leq i \leq q-1, \quad (3.9)$$

$$L^* G^{-1}(CG^{-1})^j Q^* = L^* G^{-1}(CG^{-1})^j \quad \text{for } 0 \leq j \leq r-1. \quad (3.10)$$

Equality (3.6) is now a consequence of (3.9) and Lemma 3.1, and Equality (3.7) a consequence of (3.10) and Lemma 3.2. Now for $0 \leq i \leq q-1$ and $0 \leq j \leq r-1$ we have

$$\begin{aligned} M_{\text{R}i+j+1} &= L_{\text{R}}^* G_{\text{R}}^{-1}(C_{\text{R}} G_{\text{R}}^{-1})^j C_{\text{R}}(G_{\text{R}}^{-1}C_{\text{R}})^i G_{\text{R}}^{-1}B_{\text{R}} \\ &= L_{\text{R}}^* G_{\text{R}}^{-1}(C_{\text{R}} G_{\text{R}}^{-1})^j Y^* \cdot C \cdot X(G_{\text{R}}^{-1}C_{\text{R}})^i G_{\text{R}}^{-1}B_{\text{R}} \\ &= L^* G^{-1}(CG^{-1})^j \cdot C \cdot (G^{-1}C)^i G^{-1}B \\ &= M_{i+j+1}, \end{aligned}$$

as expected. ■

Grimme's original theorem was stated in a seemingly more general but mathematically equivalent form; See Theorem 8.1. Grimme's Theorem is very general. It immediately gives the widely studied moment matching result for $G = I$ and $m = p = 1$, a well-known fact dated back as early as [24, 1974] (see also [40]) and exploited by, e.g., [8, 14, 20] for better implementations of Asymptotic Waveform Evaluation [34] and possibly automatic error estimations. It applies to the moment matching by PRIMA [31], too. In the PRIMA case, $L = B$, $Y = X$, $G = G^*$, and

$$\text{span}(X) = \mathcal{K}_k(G^{-1}C, G^{-1}B).$$

Theorem 3.1 implies $M_i = M_{\text{R}i}$ for $0 \leq i \leq k-1$. If, addition, C is also Hermitian, then $M_i = M_{\text{R}i}$ for $0 \leq i \leq 2k-1$; See Corollary 3.1 below.

Corollary 3.1 *Assume that G and C are Hermitian. If*

$$\mathcal{K}_q(G^{-1}C, G^{-1}(B \ L)) \subseteq \text{span}(X) \quad (3.11)$$

and set $Y = X$, then $M_i = M_{\text{R}i}$ for $0 \leq i \leq 2q-1$.

Proof: $Y = X$ and (3.11) imply (3.5) with $r = q$. ■

Not obviously, Grimme's Theorem also applies to the moment matching of a partial Padé-via-Lanczos method of Bai and Freund [7]; See Example 3.1. In the sequel, we shall exploit the flexibility built into Grimme's Theorem for the possibility of preserving substructures in the model matrices L, G, C, B and improving the algorithm of Su and Craig Jr. [38] on quadratic transfer functions.

Example 3.1 This is for the moment matching of a partial Padé-via-Lanczos method of Bai and Freund [7]. Here $G = I$ and $p = m = 1$ for which usually $H(s)$ and $H_R(s)$ are written as, assuming $Y^*X = I_k$,

$$H(s) = l^*(I + sA)^{-1}b, \quad H_R(s) = l_R^*(I + sA_R)^{-1}b_R$$

and thus the moments are $M_i = l^*A^i b$ and $M_{Ri} = l_R^*A_R^i b_R$. After k -steps, assuming no breakdowns, the non-symmetric Lanczos generates right and left Lanczos vectors

$$v_1, v_2, \dots, v_{k+1}, \quad \text{and} \quad w_1, w_2, \dots, w_{k+1},$$

where $v_1 = \alpha b$ and $w_1 = \beta l$ for some $\alpha \neq 0 \neq \beta$. Let $V_j = (v_1 \ v_2 \ \dots \ v_j)$ and $W_j = (w_1 \ w_2 \ \dots \ w_j)$. Then² $W_j^*V_j = I_j$ for $1 \leq j \leq k$, $W_k^*v_{k+1} = 0 = V_k^*w_{k+1}$, and

$$\text{span}(V_j) = \mathcal{K}_j(A, b), \quad \text{span}(W_j) = \mathcal{K}_j(A^*, l). \quad (3.12)$$

Thus

$$\text{span}(X) = \mathcal{K}_k(A, b), \quad \text{span}(Y) = \mathcal{K}_k(A^*, l), \quad (3.13)$$

with $X = V_k$ and $Y = W_k$ for which

$$H_R(s) = (l^*b)e_1^*(I + sT_k)^{-1}e_1 = H(s) + \mathcal{O}(s^{2k}),$$

by Theorem 3.1, where $T_k = W_k^*AV_k$. But this H_R as is may be unstable in the sense that T_k has eigenvalues in the right half plan of \mathbb{C} while original A does not. Bai and Freund [7] proposed to update T_k by a rank-one matrix ze_k^* to $T_k + ze_k^*$ and hopefully with judicious pick of vector $z \in \mathbb{C}^k$, $T_k + ze_k^*$ has no eigenvalues in the right half plan, and accordingly used $\tilde{H}_R(s) = (l^*b)e_1^*[I + s(T_k + ze_k^*)]^{-1}e_1$ as a reduced model. Cleverly exploiting the zero structures in $T_k + ze_k^*$, they proved *if z 's entries, except its last ℓ ones, are zeros, then*

$$\tilde{H}_R(s) = H(s) + \mathcal{O}(s^{2k-\ell}). \quad (3.14)$$

We now apply Theorem 3.1 to reach the same conclusion. Notice that Bai's and Freund's motivation of employing the rank-1 update makes the situations $v_{k+1} = 0$ or $w_{k+1} = 0$ uninteresting because any of the two implies all eigenvalues of T_k are A 's. Nevertheless the proof below requires that only one of the two not be zero. Assume $v_{k+1} \neq 0$. Take

²Bi-orthogonalization between v_{k+1} and w_{k+1} has not been done yet, i.e., $w_{k+1}^*v_{k+1}$ can be anything, including zero. Thus possible breakdown may occur at the very next Lanczos step.

$Y = W_k + wz^*$ and $X = V_k$, where w is any vector such that $V_{k+1}^*w = e_{k+1}$. Such w exists because $\text{rank}(V_{k+1}) = k + 1$. It is known that $AV_k = V_kT_k + v_{k+1}e_k^*$. Then

$$Y^*X = W_k^*V_k + zw^*V_k = I, \quad Y^*AX = T_k + (W_k^*v_{k+1} + zw^*v_{k+1})e_k^* = T_k + ze_k^*.$$

Since the first $k - \ell$ columns of Y are the same as those of W_k ,

$$\text{span}(Y) \supseteq \text{span}(W_{k-\ell}) = \mathcal{K}_{k-\ell}(A^*, l).$$

Still $\text{span}(X) = \mathcal{K}_k(A, b)$. By Theorem 3.1, we have (3.14). The proof for the case $w_{k+1} \neq 0$ is similar. \blacksquare

4 Short Recurrences to Build X and Y Simultaneously

What are the efficient ways to compute X and Y as (partially) constrained by Theorem 3.1? We shall now address this question.

If $r = 0$, then essentially Y is free, except of course Y^*GX being nonsingular. X can be built by Arnoldi process [4] if B is a vector or its natural block version extensions [36, p.197] or Ruhe's variant [35]. Similar statement goes to the case $q = 0$ as well. Arnoldi process usually has to run *full-length recurrences*.

If $G = I$, including transforming $H(s)$ to $H(s) = L^*(I + sG^{-1}C)^{-1}(G^{-1}B)$, then non-symmetric Lanczos [28, 29] for $p = m = 1$ or its block extension [1] to generate bi-orthogonal bases for $\mathcal{K}_k(A, B)$ and $\mathcal{K}_k(A^*, L)$ in general. Of course non-symmetric Lanczos can run into stability problems and breakdowns which can be cured with some look-ahead strategies [5, 19, 32, 33, 42]. Lanczos process relies on *short-term recurrences* to build bases of associated Krylov subspaces.

It is desirable to search for ways to build X and Y simultaneously by some kind of short-term recurrences because of its computational efficiency. Recall the short-term recurrences in Lanczos Process of A with single [28, 29] or multiple starting vectors [1] are made possible by the duality of A and A^* with respect to the standard inner product

$$\langle x, y \rangle \stackrel{\text{def}}{=} y^*x. \tag{4.1}$$

But in general $G^{-*}C^* \neq (G^{-1}C)^*$ unless $G = I$ or more generally

$$G \text{ and } C \text{ are commutative.} \tag{4.2}$$

So possibly there is no short-recurrences to build X and Y simultaneously in general (with respect to $\langle \cdot, \cdot \rangle$). But there are cases other than (4.2), which allow short-term recurrences. In the sequel, we shall study one, namely,

$$G \text{ is Hermitian.} \tag{4.3}$$

Define G -inner product [22] which is indefinite if G is:

$$\langle x, y \rangle_G \stackrel{\text{def}}{=} y^*Gx \equiv \langle Gx, y \rangle.$$

It becomes the standard inner product if $G = I$. Given $n \times n$ A , let A^\odot denote its dual with respect to G -inner product, i.e.,

$$\langle Ax, y \rangle_G \equiv \langle x, A^\odot y \rangle_G$$

for all vectors x and y of dimension n , which is equivalent to $\langle GAx, y \rangle \equiv \langle Gx, A^\odot y \rangle$ and to $\langle GAx, y \rangle \equiv \langle (A^\odot)^* Gx, y \rangle$ for all vectors x and y of dimension n . Therefore

$$GA = (A^\odot)^* G \quad \Rightarrow \quad A^\odot = G^{-1} A^* G.$$

Consider for the rest of this section $A = G^{-1}C$. Then $A^\odot = G^{-1}(C^*G^{-1})G = G^{-1}C^*$, i.e.,

$$G^{-1}C^* = (G^{-1}C)^\odot \text{ with respect to } \langle \cdot, \cdot \rangle_G. \quad (4.4)$$

With this observation, all existing Lanczos processes designed under the standard inner product are essentially extensible by substituting all

$$\begin{array}{ll} \langle \cdot, \cdot \rangle_G & \text{for } \langle \cdot, \cdot \rangle, \\ G^{-1}Cv & \text{for } Av, \quad (\text{matrix product}) \\ G^{-1}C^*w & \text{for } A^*w. \quad (\text{matrix product}) \end{array} \quad (4.5)$$

The following is a sample ideal implementation without look-ahead [19, 33] for $p = m = 1$. We call it ideal because comparing γ_j against 0 is impractical in a floating point environment. Nevertheless this ideal version suffices to convey the idea of extending existing Lanczos processes.

Algorithm 4.1 G -LANCZOS PROCESS for $A = G^{-1}C$ with single starting vectors:

```

1  Given  $v_1$  and  $w_1$  such that  $\langle v_1, w_1 \rangle_G \equiv w_1^* G v_1 = 1$ ;
2  Set  $\beta_0 = \gamma_0 \equiv 0$ ;  $v_0 = w_0 \equiv 0$ ;
3  for  $j = 1, 2, \dots, j$  do
4       $\alpha_j = \langle Av_j, w_j \rangle_G \equiv w_j^* C v_j$ ;
5       $\hat{v} = G^{-1}C v_j - \alpha_j v_j - \beta_{j-1} v_{j-1}$ ;
6       $\hat{w} = G^{-1}C^* w_j - \bar{\alpha}_j w_j - \bar{\gamma}_{j-1} w_{j-1}$ ;
7       $\gamma_j = \sqrt{|\langle \hat{v}, \hat{w} \rangle_G|} \equiv \sqrt{|\hat{w}^* G \hat{v}|}$ ;
8      if  $\gamma_j = 0$  then BREAK;
9       $\beta_j = \langle \hat{v}, \hat{w} \rangle_G / \gamma_j \equiv \hat{w}^* G \hat{v} / \gamma_j$ ;
10      $v_{j+1} = \hat{v} / \gamma_j$ ,  $w_{j+1} = \hat{w} / \beta_j$ ;
11 end for
```

Let $V_j = (v_1 \ v_2 \ \dots \ v_j)$ and $W_j = (w_1 \ w_2 \ \dots \ w_j)$. Without **BREAKing**, the recurrences in Algorithm 4.1 can be stated compactly as follows.

$$G^{-1}C V_k = V_k T_k + \gamma_k v_{k+1} e_k^*, \quad (4.6)$$

$$G^{-1}C^* W_k = W_k T_k^* + \beta_k^* w_{k+1} e_k^*, \quad (4.7)$$

where T_k is $k \times k$ and tridiagonal

$$T_k = \begin{pmatrix} \alpha_1 & \beta_1 & & & \\ \gamma_1 & \alpha_2 & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & & \gamma_{k-1} & \alpha_k \end{pmatrix}.$$

REMARK 4.1 If C is also Hermitian, then $A = G^{-1}C$ is self-adjoint with respect to G -inner product. If, in addition, G is positive definite, Algorithm 4.1 will work with initially $w_1 = v_1$, and it can be proved that subsequently all $w_j = v_j$ for all j and T_k is real symmetric. But if G is not definite, Algorithm 4.1 still can start with $w_1 = v_1$, and then the first v_j and w_j (and from there forward) differ when $\langle v_{j-1}, v_{j-1} \rangle_G < 0$. It can be shown $\text{span}(W_k) = \text{span}(V_k) = \mathcal{K}_k(A, v_1)$ even though W_k and V_k are different.

In general $p > 1$ and/or $m > 1$. Similar modifications to the main algorithm of [1] can be done.

The next question is

Any other cases that allow X and Y in Theorem 3.1 to be computed simultaneously via some short-recurrences?

(4.8)

5 Moment Matching that Preserves Substructures

Suppose the matrices L, G, C, B in the transfer function (1.2) enjoy some natural partitioning that is derived from, e.g., the physical layout of a VLSI circuit or a structural dynamical system:

$$L = \begin{matrix} n_1 \\ n_2 \end{matrix} \begin{pmatrix} L_1 \\ L_2 \end{pmatrix}, \quad G = \begin{matrix} n_1 & n_2 \\ n_2 & n_2 \end{matrix} \begin{pmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{pmatrix}, \quad C = \begin{matrix} n_1 & n_2 \\ n_2 & n_2 \end{matrix} \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix}, \quad B = \begin{matrix} n_1 \\ n_2 \end{matrix} \begin{pmatrix} B_1 \\ B_2 \end{pmatrix},$$

where $n_1 + n_2 = n$. It'd be advantageous that the reduced system inherits the same structure, i.e., L_R, G_R, C_R, B_R can be partitioned in such a way

$$L_R = \begin{matrix} k_1 \\ k_2 \end{matrix} \begin{pmatrix} L_{R1} \\ L_{R2} \end{pmatrix}, \quad G_R = \begin{matrix} k_1 & k_2 \\ k_2 & k_2 \end{matrix} \begin{pmatrix} G_{R11} & G_{R12} \\ G_{R21} & G_{R22} \end{pmatrix}, \quad C_R = \begin{matrix} k_1 & k_2 \\ k_2 & k_2 \end{matrix} \begin{pmatrix} C_{R11} & C_{R12} \\ C_{R21} & C_{R22} \end{pmatrix}, \quad B_R = \begin{matrix} k_1 \\ k_2 \end{matrix} \begin{pmatrix} B_{R1} \\ B_{R2} \end{pmatrix},$$

that each sub-block is a direct reduction from the corresponding sub-block in the original system, e.g., G_{R11} from G_{11} , where $k_1 + k_2 = k$. This can be accomplished by picking

$$X = \begin{matrix} n_1 \\ n_2 \end{matrix} \begin{pmatrix} X_1 & \\ & X_2 \end{pmatrix}, \quad Y = \begin{matrix} n_1 \\ n_2 \end{matrix} \begin{pmatrix} Y_1 & \\ & Y_2 \end{pmatrix}, \quad \text{such that } \text{rank}(X_i) = \text{rank}(Y_i) = k_i. \quad (5.1)$$

Then (1.5) becomes

$$L_{Ri} = X_i^* L_i, G_{Rij} = Y_i^* G_{ij} X_i, C_{Rij} = Y_i^* C_{ij} X_j, B_{Ri} = Y_i^* B_i. \quad (5.2)$$

A reduction as in (5.2) is conceivably useful for the system matrices with meaningful substructures. For example, for the time-domain modified nodal analysis (MNA) circuit equations targeted by PRIMA [31] and SyMPVL [18], system matrices have the following natural partitioning (adopting the formulation in [18])

$$G = \begin{pmatrix} G_{11} & G_{12} \\ G_{12}^* & \end{pmatrix}, \quad C = \begin{pmatrix} C_{11} & \\ & -C_{22} \end{pmatrix}, \quad G_{11}^* = G_{11}, \quad C_{ii}^* = C_{ii}, \quad L = B, \quad (5.3)$$

where G and C are the conductance and susceptance matrices; G_{11} , C_{11} , and C_{22} are the matrices that contain the stamps for resistors, capacitors, and inductors, respectively; G_{12} 's entries are either 1 or -1 or 0, representing the current variables in Kirchhoff's Current Law (KCL) equations. Accordingly $B = L$ have a natural partitioning, too. As stated in [31], if the original circuit is composed of passive linear elements only, C_{ii} and G_{11} are all (real) symmetric nonnegative definite. Using reduction (5.2) with $Y = X$, all these substructures will be preserved, except that the entries of G_{R12} could be numbers other than 1 or -1 or 0. Passivity of the system is preserved for the same reason as PRIMA [31].

For the sake of moment matching, Theorem 3.1 remains true here. But how do we find such X and Y ? In what follows, we shall present a general idea through an algorithm of

$$\text{producing } Z = \begin{matrix} & k_1 & k_2 \\ n_1 & \begin{pmatrix} Z_1 \\ Z_2 \end{pmatrix} \\ n_2 & \end{matrix} \quad \text{from any given } \tilde{Z} = \begin{matrix} n_1 \\ n_2 \end{matrix} \begin{pmatrix} \tilde{Z}_1 \\ \tilde{Z}_2 \end{pmatrix} \quad (5.4)$$

such that $\text{span}(Z) = \text{span}(\tilde{Z})$.

Algorithm 5.1 FROM GIVEN \tilde{Z} TO Z :

1. **Compute** $Z_i \in \mathbb{C}^{n_i \times k_i}$ **having full column rank such that** $\text{span}(Z_i) \supseteq \text{span}(\tilde{Z}_i)$;
2. **Output** $Z = \begin{pmatrix} Z_1 \\ Z_2 \end{pmatrix}$.

There are a variety of ways to realize Step 1: Rank revealing QR decompositions and its variations [12, 15, 27, 37], (Modified) Gram-Schmit process, or Singular Value Decompositions [11, 13, 23]. For maximum efficiency, one should make Z_i has as fewer columns as one can. Notice the smallest possible number is $\text{rank}(\tilde{Z}_i)$, but one may have to add a few more columns to make sure X_i and Y_i have the same number of columns, as required by Item 3 of Theorem 5.1, below.

Theorem 5.1 *Let X be the output of Algorithm 5.1 with input \tilde{X} , and likewise let Y be the output with input \tilde{Y} . Assume that X_i and Y_i have the same number of columns and that G_R is nonsingular.*

1. *If $\mathcal{K}_q(G^{-1}C, G^{-1}B) \subseteq \text{span}(\tilde{X})$ and $Y = X$, then $M_i = M_{Ri}$ for $0 \leq i \leq q - 1$.*

6 Structures of Krylov Subspaces of Block Matrices

The results of this section are of general interest, and they are influenced by Su and Criag [38] who appeared to the first in exploiting structures of Krylov subspaces for better numerical behavior. The matrices here do not necessarily have anything to do with the transfer function mentioned in Section 1. Consider

$$A = \begin{matrix} & \begin{matrix} n_1 & n_2 \end{matrix} \\ \begin{matrix} n_1 \\ n_2 \end{matrix} & \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \end{matrix}, \quad B = \begin{matrix} n_1 \\ n_2 \end{matrix} \begin{pmatrix} B_1 \\ B_2 \end{pmatrix}, \quad (6.1)$$

where $n_1 + n_2 = n$. The following theorem describes the structures in a basis matrix of $\mathcal{K}_k(A, B)$ when one of A_{ij} 's is zero.

Theorem 6.1 *Let A and B be partitioned as in (6.1), and let $\text{span}(\tilde{X}) = \mathcal{K}_k(A, B)$ be partitioned as*

$$\tilde{X} = \begin{matrix} & \begin{matrix} \ell_1 & \ell_2 \end{matrix} \\ \begin{matrix} n_1 \\ n_2 \end{matrix} & \begin{pmatrix} \tilde{X}_{11} & \tilde{X}_{12} \\ \tilde{X}_{21} & \tilde{X}_{22} \end{pmatrix} \end{matrix} \equiv \begin{matrix} n_1 \\ n_2 \end{matrix} \begin{pmatrix} \tilde{X}_1 \\ \tilde{X}_2 \end{pmatrix}$$

such that $\text{span} \begin{pmatrix} \tilde{X}_{11} \\ \tilde{X}_{21} \end{pmatrix} = \mathcal{K}_{k-1}(A, B)$, and let $\alpha \neq 0$ be a scalar which may be different at different occurrences. Then

1. If $A_{11} = 0$, then $\text{span}(\tilde{X}_1) = \text{span}(B_1 \ A_{12}\tilde{X}_{21}) \subseteq \text{span}(B_1 \ A_{12}\tilde{X}_2)$. If in addition $A_{12} = \alpha I$ (and thus $n_1 = n_2$), $\text{span}(\tilde{X}_1) = \text{span}(B_1 \ \tilde{X}_{21}) \subseteq \text{span}(B_1 \ \tilde{X}_2)$.
2. If $A_{12} = 0$, then $\text{span}(\tilde{X}_1) = \mathcal{K}_k(A_{11}, B_1)$.
3. If $A_{21} = 0$, then $\text{span}(\tilde{X}_2) = \mathcal{K}_k(A_{22}, B_2)$.
4. If $A_{22} = 0$, then $\text{span}(\tilde{X}_2) = \text{span}(B_2 \ A_{21}\tilde{X}_{11}) \subseteq \text{span}(B_2 \ A_{21}\tilde{X}_1)$. If in addition $A_{21} = \alpha I$ (and thus $n_1 = n_2$), $\text{span}(\tilde{X}_2) = \text{span}(B_2 \ \tilde{X}_{11}) \subseteq \text{span}(B_2 \ \tilde{X}_1)$.

Proof: All claims are consequences of the following observation:

$$\text{if } A^i B = \begin{matrix} n_1 \\ n_2 \end{matrix} \begin{pmatrix} Z_1 \\ Z_2 \end{pmatrix}, \quad \text{then } A^{i+1} B = \begin{pmatrix} A_{11}Z_1 + A_{12}Z_2 \\ A_{21}Z_1 + A_{22}Z_2 \end{pmatrix}.$$

Then combining the assumption that one of $A_{ij} = 0$ will complete the proof. ■

Item 4 of Theorem 6.1 was implicitly stated in [38, 9]. It gives a relation between $\text{span}(\tilde{X}_1)$ and $\text{span}(\tilde{X}_2)$; so does Item 1. It was Item 4 that led Su's and Craig's discovery of quadratic preserving model reduction. See next Section 7.

Theorem 6.1 can be extended to block matrices A other than just 2×2 . A conceivable case that allows us to derive simple relations among block of a basis matrix conformally

partitioned of Krylov subspaces are all block rows, except one, have only one nonzero blocks, i.e., if

$$A = \begin{matrix} & \begin{matrix} n_1 & n_2 & & n_d \end{matrix} \\ \begin{matrix} n_1 \\ n_2 \\ \vdots \\ n_d \end{matrix} & \begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1d} \\ A_{21} & A_{22} & \cdots & A_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ A_{d1} & A_{d2} & \cdots & A_{dd} \end{pmatrix} \end{matrix},$$

then except one block row of A , each of the rest block rows has at most one nonzero block. Even so there are still many subcases, one of which is

$$A = \begin{matrix} & \begin{matrix} n_1 & n_2 & & n_d \end{matrix} \\ \begin{matrix} n_1 \\ n_2 \\ \vdots \\ n_d \end{matrix} & \begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1d} \\ A_{21} & & & \\ & \ddots & & \\ & & A_{dd-1} & \end{pmatrix} \end{matrix}. \quad (6.2)$$

Matrices like this come from linearization of a polynomial eigenvalue problem [21] in which $A_{21} = A_{32} = \cdots = A_{dd-1} = I$ and $n_1 = n_2 = \cdots = n_d$. As an example, we shall state a theorem for A having form (6.2). Partition B conformally as

Theorem 6.2 *Let A be partitioned as in (6.2), and let $\text{span}(\tilde{X}) = \mathcal{K}_k(A, B)$. Partition B and \tilde{X} accordingly as*

$$B = \begin{matrix} n_1 \\ n_2 \\ \vdots \\ n_d \end{matrix} \begin{pmatrix} B_1 \\ B_2 \\ \vdots \\ B_d \end{pmatrix}, \quad \tilde{X} = \begin{matrix} n_1 \\ n_2 \\ \vdots \\ n_d \end{matrix} \begin{pmatrix} \tilde{X}_1 \\ \tilde{X}_2 \\ \vdots \\ \tilde{X}_d \end{pmatrix}.$$

Then

$$\text{span}(\tilde{X}_i) \subseteq \text{span}(B_i \ A_{ii-1} \tilde{X}_{i-1})$$

for $2 \leq i \leq d$. If in addition $A_{ii-1} = \alpha_i I$ for $2 \leq i \leq d$ (and thus $n_1 = n_2 = \cdots = n_d$), then

$$\text{span}(\tilde{X}_i) \subseteq \text{span}(B_i \ \tilde{X}_{i-1}) \subseteq \text{span}(B_i \ B_{i-1} \ \cdots \ B_2 \ \tilde{X}_1).$$

for $2 \leq i \leq d$.

7 Quadratic Transfer Functions

By a quadratic transfer function we mean $H(s)$ taking this form

$$H(s) = (V^* + sT^*)(s^2M + sD + K)^{-1}R \quad (7.1)$$

arising from applying Laplace transformation to the time-invariant multi-input multi-output second order dynamical system

$$\begin{cases} M\ddot{q}(t) + D\dot{q}(t) + Kq(t) &= Ru(t), \\ y(t) &= T^*\dot{q}(t) + V^*q(t), \end{cases} \quad (7.2)$$

where $M, D, K \in \mathbb{C}^{n \times n}$, $R \in \mathbb{C}^{n \times m}$, $T, V \in \mathbb{C}^{n \times p}$, and $q(t)$ and $y(t)$ are vector-valued functions of apt dimensions. The reader is referred to, e.g., [38], for more background information. Notation here is adopted from structural dynamics, where M, D, K are mass, damping, and stiffness matrices and are usually Hermitian, but can be non-Hermitian at times [39, p.244].

It is quite common to deal with (7.1) and (7.2) by a linearization technique to turn them into the forms of (1.1) and (1.2). This is done by setting

$$C = \begin{pmatrix} D & M \\ W & 0 \end{pmatrix}, G = \begin{pmatrix} K & \\ & -W \end{pmatrix}, L = \begin{pmatrix} V \\ T \end{pmatrix}, B = \begin{pmatrix} R \\ 0 \end{pmatrix}, x(t) = \begin{pmatrix} q(t) \\ \dot{q}(t) \end{pmatrix}, \quad (7.3)$$

where W is any $n \times n$ nonsingular matrix, usually taken to be M when M, D, K are Hermitian or simply I otherwise. By now, all existing developments for linear transfer functions (1.2) can be applied in a straightforward way, but then reduced models likely lose the quadratic characteristics, i.e., they may not be turned into quadratic transfer functions with (much) smaller sizes, and consequently the reduced models have little physical significance. To overcome this, Su and Craig [38] made an important observation which is equivalent to Item 4 of Theorem 6.1 and to which we'll return later. Similar efforts were made by [26, 30] where algorithms, provably efficient only for so-called low-rank cases, were proposed.

Su and Craig [38] were interested in reduced models that were quadratic preserving:

$$H_R(s) = (V_R^* + sT_R^*)(s^2M_R + sD_R + K_R)^{-1}R_R \quad (7.4)$$

for $H(s)$ in (7.1), where $X_1, Y_1 \in \mathbb{C}^{n \times k}$ having full column rank, and

$$M_R = Y_1^* M X_1, D_R = Y_1^* D X_1, K_R = Y_1^* K X_1, \quad (7.5)$$

$$V_R = X_1^* V, T_R = X_1^* T, R_R = Y_1^* R. \quad (7.6)$$

In Su's and Craig's case, $Y_1 = X_1$. Together with L, G, C, B as defined by (7.3), we have

$$H(s) = L^*(G + sC)^{-1}B, \quad H_R(s) = L_R^*(G_R + sC_R)^{-1}B_R,$$

where $L_R = X^*L$, $G_R = Y^*GX$, $C_R = Y^*CX$, and $B_R = Y^*B$ as in (1.5), and

$$X = \begin{pmatrix} X_1 & \\ & X_1 \end{pmatrix}, Y = \begin{pmatrix} Y_1 & \\ & Y_1 \end{pmatrix}. \quad (7.7)$$

Assume K and W nonsingular. We have

$$G^{-1}C = \begin{pmatrix} K^{-1}D & K^{-1}M \\ -I & 0 \end{pmatrix}, \quad G^{-*}C^* = \begin{pmatrix} K^{-*}D^* & K^{-*}W^* \\ -W^{-*}M^* & 0 \end{pmatrix}. \quad (7.8)$$

Both have a zero block, a condition of Theorem 6.1. Moreover the (2,1)-block of $G^{-1}C$ is $-I$ always, and that of $G^{-*}C^*$ can be made $-I$, too, if $W = M$. In what follows, \tilde{X} and \tilde{Y} are always partitioned as

$$\tilde{X} = \begin{matrix} n \\ n \end{matrix} \begin{pmatrix} \tilde{X}_1 \\ \tilde{X}_2 \end{pmatrix}, \quad \tilde{Y} = \begin{matrix} n \\ n \end{matrix} \begin{pmatrix} \tilde{Y}_1 \\ \tilde{Y}_2 \end{pmatrix}.$$

Theorem 7.1 *Suppose that M , D , and K are Hermitian, and $W = M$, and that M and K are nonsingular. With L, G, C, B as defined by (7.3), if*

$$\mathcal{K}_q(G^{-1}C, G^{-1}(B \ L)) \subseteq \text{span}(\tilde{X}), \quad \text{span}(T) \subseteq \text{span}(\tilde{X}_1), \quad (7.9)$$

$$\text{span}(\tilde{X}_1) \subseteq \text{span}(X_1), \quad (7.10)$$

and $Y = X$ as in (7.7), then $H_R = H(s) + \mathcal{O}(s^{2q})$.

Proof: That $W = M$ implies that both (2,1)-blocks in (7.8) are $-I$. By Item 4 of Theorem 6.1 and B and L in (7.3),

$$\text{span}(\tilde{X}_1) = \text{span}(T \ \tilde{X}_1) \supseteq \text{span}(\tilde{X}_2).$$

Therefore $\text{span}(X_1) \supseteq \text{span}(\tilde{X}_1)$ implies

$$\begin{aligned} \text{span}(X) &\supseteq \text{span}(\tilde{X}) \supseteq \mathcal{K}_q(G^{-1}C, G^{-1}(B \ L)) \\ &\supseteq \begin{cases} \mathcal{K}_q(G^{-1}C, G^{-1}B), \\ \mathcal{K}_q(G^{-1}C, G^{-1}L) = \mathcal{K}_r(G^{-*}C^*, G^{-*}L), \end{cases} \end{aligned}$$

because G and C are Hermitian. The conclusion that $H_R = H(s) + \mathcal{O}(s^{2q})$ is now a consequence of Theorem 3.1. \blacksquare

The main theorem of Su and Craig [38, p.62], in our notation, essentially is as follows. For Hermitian M, D, K , if

$$\mathcal{K}_q(G^{-1}C, G^{-1}B) \subseteq \text{span}(\tilde{X}), \quad \mathcal{K}_r(G^{-1}C, G^{-1}L) \subseteq \text{span}(\tilde{X}), \quad (7.11)$$

then $H_R = H(s) + \mathcal{O}(s^{q+r})$ with $X_1 = \tilde{X}_1$. Which as we see now is a corollary of Grimme's theorem. In using this result of theirs, Su and Craig proposed an algorithm which essentially computed \tilde{X} such that

$$\mathcal{K}_q(G^{-1}C, G^{-1} \begin{pmatrix} R & V & T \\ 0 & 0 & 0 \end{pmatrix}) \subseteq \text{span}(\tilde{X}), \quad (7.12)$$

by a variant of Algorithm 10.2 with respect to K -inner product $\langle u, v \rangle_K = v^* K u$ on the first n entries of each column of \tilde{X} , and then take $X_1 = \tilde{X}_1$. Notice that Su and Craig's version of the process assumes no rank degeneration during the entire process, i.e., projecting new block of vectors onto \tilde{X}_1 's columns already computed based on subvectors will never result in a rank deficient block, an assumption that can be too strong sometimes. Based upon Theorem 7.1, Algorithm 7.1 below improves this algorithm with less strong assumption (7.9) than (7.12). The benefit for this is \tilde{X} (and thus X_1) having fewer columns, and thus a smaller reduced model.

Algorithm 7.1 Compute X_1 for Theorem 7.1.

1. Compute \widehat{X} such that $\mathcal{K}_q(G^{-1}C, G^{-1}(B \ L)) \subseteq \text{span}(\widehat{X})$, by, e.g., Algorithm 10.2 or any block version Arnoldi process.
2. Partition $\widehat{X} = \begin{matrix} n \\ n \end{matrix} \begin{pmatrix} \widehat{X}_1 \\ \widehat{X}_2 \end{pmatrix}$.
3. Compute $\widetilde{X}_1 \in \mathbb{C}^{n \times k}$ having full column rank such that $\text{span}(\widehat{X}_1 \ T) \subseteq \text{span}(\widetilde{X}_1)$ by, e.g., SVD, QR, or any rank revealing decompositions.
4. Take $X_1 = \widetilde{X}_1$.

The theorem below does not assume that M , D , and K are Hermitian.

Theorem 7.2 *Let integers $q, r \geq 0$, and let $W = M$. With L, G, C, B as defined by (7.3), if*

$$\mathcal{K}_q(G^{-1}C, G^{-1}B) \subseteq \text{span}(\widetilde{X}), \quad (7.13)$$

$$\mathcal{K}_r(G^{-*}C^*, G^{-*}L) \subseteq \text{span}(\widetilde{Y}), \quad \text{span}(T) \subseteq \text{span}(\widetilde{Y}_1), \quad (7.14)$$

$$\text{span}(\widetilde{X}_1) \subseteq \text{span}(X_1), \quad \text{span}(\widetilde{Y}_1) \subseteq \text{span}(Y_1), \quad (7.15)$$

then $H_R = H(s) + \mathcal{O}(s^{q+r})$.

Proof: That $W = M$ implies that both (2,1)-blocks in (7.8) are $-I$. By Item 4 of Theorem 6.1 and the B and L in (7.3),

$$\text{span}(\widetilde{X}_1) \supseteq \text{span}(\widetilde{X}_2), \quad \text{span}(\widetilde{Y}_1) = \text{span}(T \ \widetilde{Y}_1) \supseteq \text{span}(\widetilde{Y}_2).$$

Therefore

$$\begin{aligned} \text{span}(X) &\supseteq \text{span}(\widetilde{X}) \supseteq \mathcal{K}_q(G^{-1}C, G^{-1}B) \\ \text{span}(Y) &\supseteq \mathcal{K}_r(G^{-*}C^*, G^{-*}L). \end{aligned}$$

The conclusion that $H_R = H(s) + \mathcal{O}(s^{q+r})$ is now a consequence of Theorem 3.1. ■

Algorithm 7.2 Compute X_1 and Y_1 for Theorem 7.2.

1. Compute \widetilde{X} and \widehat{Y} such that $\mathcal{K}_q(G^{-1}C, G^{-1}B) \subseteq \text{span}(\widetilde{X})$, and $\mathcal{K}_r(G^{-*}C^*, G^{-*}L) \subseteq \text{span}(\widehat{Y})$ by, e.g., Algorithm 10.2 or any block version Arnoldi process.
2. Partition $\widetilde{X} = \begin{matrix} n \\ n \end{matrix} \begin{pmatrix} \widetilde{X}_1 \\ \widetilde{X}_2 \end{pmatrix}$ and $\widehat{Y} = \begin{matrix} n \\ n \end{matrix} \begin{pmatrix} \widehat{Y}_1 \\ \widehat{Y}_2 \end{pmatrix}$.
3. Compute $\widetilde{Y}_1 \in \mathbb{C}^{n \times k}$ having full column rank such that $\text{span}(\widehat{Y}_1 \ T) \subseteq \text{span}(\widetilde{Y}_1)$ by, e.g., SVD, QR, or any rank revealing decompositions.

4. Expand the columns of \tilde{X}_1 or \tilde{Y}_1 so that both have the same number of columns and full column rank, and call them X_1 and Y_1 .

REMARK 7.1 It is tempting to think the idea we discussed so far could be extended for quadratic transfer function

$$H(s) = (V^* + sT^*)(s^2M + sD + K)^{-1}(R + sU) \quad (7.16)$$

which degenerate to (7.1) when $U = 0$. It turns out this $H(s)$ cannot be linearized as $L^*(G + sC)^{-1}B$ with L, G, C, B as in (7.3), except $B = \begin{pmatrix} R \\ U \end{pmatrix}$. That is the common linearization technique works only for $U = 0$ (or $T = 0$). Of course based upon the application background (7.2), $U = 0$ always!

REMARK 7.2 The idea in this section is naturally extensible to transfer functions of higher degree

$$H(s) = \left(\sum_{i=0}^{d-1} s^i V_i \right) \left(\sum_{i=0}^d s^i A_i \right)^{-1} R,$$

thanks to Theorem 6.2. Detail is omitted.

8 Moment Matching at $s_0 \neq 0$

In case when approximations to $H(s)$ around a given point $s_0 \neq 0$ are sought, a shift

$$s = s_0 + (s - s_0) \quad (8.1)$$

can be performed and then

$$G + sC = G + s_0C + (s - s_0)C \stackrel{\text{def}}{=} G(s_0) + \tilde{s}C. \quad (8.2)$$

Upon substitutions (i.e., renaming)

$$G(s_0) \rightarrow G, \quad \tilde{s} \rightarrow s,$$

the problem of approximating $H(s)$ around $s = s_0$ becomes equivalently approximating the substituted $H(s)$ around $s = 0$. Observe that any reduction on $G(s_0)$ and C by $Y^*G(s_0)X$ and Y^*CX can be done through reducing G and C directly as in (1.5) because

$$G_R(s_0) \stackrel{\text{def}}{=} Y^*G(s_0)X = Y^*GX + s_0Y^*CX = G_R + s_0C_R. \quad (8.3)$$

This is a significant observation because it says that even for approximating $H(s)$ near a different point $s_0 \neq 0$, reduction can still be done directly to the original matrices L, G, C , and B , regardless of the shift (8.1). This will become handy when substructures in the original system are worth preserving.

As a straightforward application of Theorem 3.1, we have

Theorem 8.1 (Grimme) *Let integers $q, r \geq 0$, and let $G(s_0)$ be defined as in (8.2). If*

$$\mathcal{K}_q(G(s_0)^{-1}C, G(s_0)^{-1}B) \subseteq \text{span}(X), \quad \mathcal{K}_r(G(s_0)^{-*}C^*, G(s_0)^{-*}L) \subseteq \text{span}(Y), \quad (8.4)$$

then $M_i(s_0) = M_{Ri}(s_0)$ for $0 \leq i \leq q + r - 1$, where

$$M_i(s_0) \stackrel{\text{def}}{=} L^*(G(s_0)^{-1}C)^i G(s_0)^{-1}B, \quad M_{Ri}(s_0) \stackrel{\text{def}}{=} L_R^*(G_R(s_0)^{-1}C_R)^i G_R(s_0)^{-1}B_R,$$

i.e., $H(s) = H_R(s) + \mathcal{O}((s - s_0)^{q+r})$.

The invariance property (8.3) of the reductions on L , G , C , and B regardless of the shift (8.1) makes it possible to match moments at multiple points by one reduction. This is done by enforcing $\text{span}(X)$ and/or $\text{span}(Y)$ containing more appropriate Krylov subspaces associated at multiple points. To avoid repetition, we shall omit explicitly stating it. See Grimme [25] for more detail.

REMARK 8.1 Other theorems for at $s_0 = 0$ in the later sections have analogous counterparts for at $s_0 \neq 0$, too. But we shall avoid explicitly stating them, too.

9 Modified Gram-Schmidt Sub-orthogonalization Process

Gram-Schmidt Orthogonalization Process generates an orthonormal basis from any given m vectors of dimension n . But usually the so-called Modified Gram-Schmidt Process is used for better orthogonalized basis vectors. But still from time to time, the computed basis vectors are not fully orthogonal. One way to cure this is to perform some kind of (selective) re-orthogonalization. Algorithm 9.1 below is a possible implementation of Modified Gram-Schmidt Process with selective re-orthogonalization, where N records the number of basis vectors already computed, τ_1 ($= \epsilon \|w\|_2$ here) is the tolerance for deciding whether w is linearly dependent upon basis vectors q_i already computed, τ_2 ($= \eta \|w\|_2$ here) is the tolerance for deciding whether to perform a full reorthogonalization. Roughly speaking there are about $(-\log_{10} \eta)$ many decimal digits cancelled during orthogonalizing w against already computed q_i 's, which again roughly speaking is translated into the incorrect decimal digits in the computed q_i from w without preforming reorthogonalization. Possible ϵ and η are $n\epsilon_m$ and 10^{-2} , for example, where ϵ_m is unit machine roundoff.

Algorithm 9.1 Modified Gram-Schmidt:

Input: $B = (b_1, b_2, \dots, b_m) \in \mathbb{C}^{n \times m}$, ϵ , and η .

1. $N = 0$;
2. **for** $j = 1, 2, \dots, m$ **do**
3. $w = b_j$; $\tau_1 = \epsilon \|w\|_2$; $\tau_2 = \eta \|w\|_2$;
4. **for** $i = 1, 2, \dots, N$ **do**
5. $r_{ij} = q_i^* w$; $w = w - r_{ij} q_i$;
6. **enddo**
7. $r_{N+1j} = \|w\|_2$;
8. **if** $r_{N+1j} \leq \tau_2$;
9. /* re-orthogonalize w against q_i 's */

```

10.         for  $i = 1, 2, \dots, N$  do
11.              $\xi = q_i^* w; w = w - \xi q_i; r_{ij} = r_{ij} + \xi;$ 
12.         enddo
13.          $r_{N+1j} = \|w\|_2;$ 
14.     endif
15.     if  $r_{N+1j} \leq \tau_1;$ 
16.          $r_{N+1j} = 0;$ 
17.     else
18.          $N = N + 1; q_N = w/r_{Nj};$ 
19.     endif
20. enddo

```

A version of Modified Gram-Schmidt Process in *exact* arithmetic can be gotten from Algorithm 9.1 by setting $\tau_1 = \tau_2 = 0$ always in Line 3 and deleting Lines 8 – 14 which are for the selective re-orthogonalization. It can be seen this exact arithmetic version will always compute an orthonormal basis for $\text{span}(B)$, and $B = Q_N R$, where $Q_N = (q_1 \ q_2 \ \dots \ q_N)$, $R = (r_{ij}) \in \mathbb{C}^{N \times m}$ with entries not explicitly computed by Algorithm 9.1 being zeros.

Instead of enforcing the orthogonality among whole basis vectors q_i , the process can be modified to, roughly speaking, enforce the same portions of all q_i to be orthogonal against each other. This is done in Algorithm 9.2 below.

Let $\mathcal{I} \subseteq \{1, 2, \dots, n\}$, and $q \in \mathbb{C}^n$. We shall use MATLAB-like notation $q(\mathcal{I})$ for the subvector of q with entries whose indices are in \mathcal{I} . $q(\mathcal{I})$ is called the \mathcal{I} -subvector of q , or just a subvector of q if \mathcal{I} can read off from the context. Set $\mathcal{I}^c = \{\ell : 1 \leq \ell \leq n, \ell \notin \mathcal{I}\}$. Let n_1 be the number of entries in \mathcal{I} , and $n_2 = n - n_1$ which is the number of entries in \mathcal{I}^c .

Definition 9.1 Given $\mathcal{I} \in \{1, 2, \dots, n\}$, two vectors $x, y \in \mathbb{C}^n$ are said to be \mathcal{I} -sub-orthogonal if $y(\mathcal{I})^* x(\mathcal{I}) = 0$. A set of vectors $\{q_1, q_2, \dots, q_N\}$ of dimension n is said to be \mathcal{I} -sub-orthonormal if they satisfy the following three properties:

1. $q_\ell(\mathcal{I}) = 0$ for $\ell \in \mathcal{J} \subseteq \{1, 2, \dots, N\}$, and $q_i(\mathcal{I}) \neq 0$ for $i \notin \mathcal{J}$,
2. $\{q_i(\mathcal{I}) : i \notin \mathcal{J}\}$ is orthonormal, and
3. $\{q_i(\mathcal{I}^c) : i \in \mathcal{J}\}$ is orthonormal.

Notice that the index set \mathcal{J} is uniquely determined by Item 1. The following lemma is closely related to [9, Lemma 2].

Lemma 9.1 \mathcal{I} -sub-orthonormal set $\{q_1, q_2, \dots, q_N\}$ is linearly independent.

Proof: Consider this linear combination $\sum_{i=1}^N \alpha_i q_i = 0$ which is equivalent to

$$\sum_{i=1}^N \alpha_i q_i(\mathcal{I}) = 0, \quad \sum_{i=1}^N \alpha_i q_i(\mathcal{I}^c) = 0.$$

The first equation gives $\sum_{i \notin \mathcal{J}} \alpha_i q_i(\mathcal{I}) = 0$ by the first condition in the lemma. Now using the second condition of the lemma, we deduce that $\alpha_i = 0$ for $i \notin \mathcal{J}$, substituting which

into the second equation to get $\sum_{i \in \mathcal{J}} \alpha_i q_i(\mathcal{I}^c) = 0$. Now use the third condition of the lemma to arrive at $\alpha_i = 0$ for $i \in \mathcal{J}$. \blacksquare

Algorithm 9.2 below attempts to compute an \mathcal{I} -sub-orthonormal basis of $\text{span}(B)$, where N , τ_1 , and τ_2 play the same roles as those of Algorithm 9.1, \mathcal{J} stores the indices ℓ of those q_ℓ satisfying $q_\ell(\mathcal{I}) = 0$.

Algorithm 9.2 Modified Gram-Schmidt Sub-orthogonalization:

Given $B = (b_1, b_2, \dots, b_m)$. Initially $N = 0$, $\mathcal{J} = \emptyset$, and $j = 1$.

1. Set $w = b_j$.
2. Orthogonalize $w(\mathcal{I})$ against $\{q_i(\mathcal{I})\}_{i=1}^N$:
$$r_{ij} = q_i(\mathcal{I})^* w(\mathcal{I}), \quad w = w - r_{ij} q_i, \quad \text{for } i = 1, 2, \dots, N.$$
3. If necessary, re-orthogonalize $w(\mathcal{I})$ against $\{q_i(\mathcal{I})\}_{i=1}^N$.
4. If $w(\mathcal{I})$ cannot be considered a zero vector, then set $N = N + 1$, $r_{Nj} = \|w(\mathcal{I})\|_2$, $q_N = w/r_{Nj}$, and $j = j + 1$. Go to Step 1.
5. Now $w(\mathcal{I}) = 0$. If $w(\mathcal{I}^c)$ can be considered a zero vector also, then set $j = j + 1$. Go to Step 1.
6. Orthogonalize $w(\mathcal{I}^c)$ against all $\{q_\ell(\mathcal{I}^c), \ell \in \mathcal{J}\}$:
$$r_{\ell j} = q_\ell(\mathcal{I}^c)^* w(\mathcal{I}^c), \quad w(\mathcal{I}^c) = w(\mathcal{I}^c) - r_{\ell j} q_\ell(\mathcal{I}^c), \quad \text{for each } \ell \in \mathcal{J}.$$
7. If necessary, re-orthogonalize $w(\mathcal{I}^c)$ against all $\{q_\ell(\mathcal{I}^c), \ell \in \mathcal{J}\}$.
8. If $w(\mathcal{I}^c)$ cannot be considered a zero vector, then set $N = N + 1$, $q_N(\mathcal{I}) = 0$, $q_N(\mathcal{I}^c) = w(\mathcal{I}^c)/r_{Nj}$, $\mathcal{J} = \mathcal{J} \cup \{N\}$. Go to Step 1.
9. Now $w(\mathcal{I}^c) = 0$. Set $j = j + 1$, and go to Step 1.

What properties can we say about the vectors q_i 's by Algorithm 9.2. To this end we shall prove a theorem for its exact arithmetic version by which we mean

<p>The <i>exact arithmetic version</i> of Algorithm 9.2: All computations and comparisons are done in exact arithmetic (and thus re-orthogonalization Steps 3 and 7 can be safely deleted).</p>	(9.1)
---	-------

Of course this is too ideal to be useful. Any actual implementation will have to set up tolerances to determine whether the residual of a vector after projecting onto q_i 's already computed should be set to zero and consequently the linear dependency or independency relationships among them. See Remark 9.1.

Theorem 9.1 *At the conclusion of Algorithm 9.2 (exact arithmetic version), q_1, q_2, \dots, q_N are \mathcal{I} -sub-orthonormal, and form a basis of $\text{span}(B)$, and thus $\dim \text{span}(B) = N$.*

Proof: It follows from the design of the algorithm and Lemma 9.1. ■

REMARK 9.1 Algorithm 9.1 was stated with its numerical implementation in mind. Criteria for whether to re-orthogonalize $w(\mathcal{I})$ at Step 3 or $w(\mathcal{I}^c)$ at Step 7 can be borrowed from the similar idea presented in Algorithm 9.1. To be specific, one picks a relative tolerance η , say 10^{-2} . Then the re-orthogonalization at Step 3 is performed if

$$(\|w(\mathcal{I})\| \text{ before Step 2}) \leq \eta(\|w(\mathcal{I})\| \text{ after Step 2}),$$

and similarly the re-orthogonalization at Step 7 is performed if

$$(\|w(\mathcal{I}^c)\| \text{ before Step 6}) \leq \eta(\|w(\mathcal{I}^c)\| \text{ after Step 6}).$$

What about “ $w(\mathcal{I})$ cannot be considered a zero vector” at Step 4? A reasonable way is to check if

$$(\|w(\mathcal{I})\| \text{ before Step 2}) \leq \epsilon(\|w(\mathcal{I})\| \text{ after Step 3}), \quad \text{or} \quad (9.2)$$

$$\|w(\mathcal{I})\| \leq \epsilon\|w(\mathcal{I}^c)\| \quad \text{at Step 4}, \quad (9.3)$$

where ϵ is a pre-chosen relative tolerance, say $n\epsilon_m$. $w(\mathcal{I})$ is considered zero if one of the two holds, and not zero otherwise. Here ϵ_m is unit machine roundoff. At Step 5, “ $w(\mathcal{I}^c)$ can be considered a zero vector also” if

$$(\|w(\mathcal{I}^c)\| \text{ before Step 2}) \leq \epsilon(\|w(\mathcal{I}^c)\| \text{ after Step 3}).$$

As to whether to consider $w(\mathcal{I}^c)$ zero or not at Step 8, one may check if

$$(\|w(\mathcal{I}^c)\| \text{ before Step 6}) \leq \epsilon(\|w(\mathcal{I}^c)\| \text{ after Step 7}).$$

REMARK 9.2 The concepts of newly introduced \mathcal{I} -sub-orthogonal and \mathcal{I} -sub-orthonormal coincide with the ordinary concepts of *orthogonal* and *orthonormal* when $\mathcal{I} = \{1, 2, \dots, n\}$ and $\mathcal{J} = \emptyset$.

10 A Sub-orthogonalized Arnoldi Process

The standard vector-version Arnoldi process generates an orthonormal basis $\{q_1, q_2, \dots, q_N\}$ for the Krylov subspace $\mathcal{K}_k(A, b)$, where $q_1 = b/\|b\|_2$ and $N = \dim \mathcal{K}_k(A, b)$. This section extends it to the so-called *Sub-orthogonalized Arnoldi Process* which computes an \mathcal{I} -sub-orthonormal basis of $\mathcal{K}_k(A, b)$.

Let $A \in \mathbb{C}^{n \times n}$, and $b \in \mathbb{C}^n$. In the algorithm below, \mathcal{J} records the indices of those q_ℓ whose \mathcal{I} -subvectors are zeros. The sub-orthogonalization process is in the same spirit of Algorithm 9.2 for the modified Gram-Schmidt sub-orthogonalization process.

Algorithm 10.1 Sub-orthogonalized Arnoldi Process:

Given $A \in \mathbb{C}^{n \times n}$, $0 \neq b \in \mathbb{C}^n$, the number k of Arnoldi steps. Initially $\mathcal{J} = \emptyset$.

1. If $b(\mathcal{I})$ cannot be considered zero, then set $q_1 = b/\|b(\mathcal{I})\|_2$; Otherwise set $\mathcal{J} = \{1\}$ and $q_1 = b/\|b(\mathcal{I}^c)\|_2$. Set $j = 1$.
2. If $j \geq k$, then **STOP**; Otherwise $w = Aq_j$.
3. Orthogonalize $w(\mathcal{I})$ against $\{q_i(\mathcal{I})\}_{i=1}^j$:

$$h_{ij} = q_i(\mathcal{I})^* w(\mathcal{I}), \quad w = w - h_{ij}q_i, \quad \text{for } i = 1, 2, \dots, j.$$

4. If necessary, re-orthogonalize $w(\mathcal{I})$ against $\{q_i(\mathcal{I})\}_{i=1}^j$.
5. If $w(\mathcal{I})$ cannot be considered zero vector, then set $h_{j+1j} = \|w(\mathcal{I})\|_2$, $q_{j+1} = w/h_{j+1j}$. Set $j = j + 1$, and go to Step 2.
6. Now $w(\mathcal{I}) = 0$. If $w(\mathcal{I}^c)$ can be considered a zero vector also, then **STOP**. (An invariant subspace of A has been found!)
7. Orthogonalize $w(\mathcal{I}^c)$ against all $\{q_\ell(\mathcal{I}^c), \ell \in \mathcal{J}\}$:

$$h_{\ell j} = q_\ell(\mathcal{I}^c)^* w(\mathcal{I}^c), \quad w(\mathcal{I}^c) = w(\mathcal{I}^c) - h_{\ell j}q_\ell(\mathcal{I}^c), \quad \text{for each } \ell \in \mathcal{J}.$$

8. If necessary, re-orthogonalize $w(\mathcal{I}^c)$ against all $\{q_\ell(\mathcal{I}^c), \ell \in \mathcal{J}\}$.
9. If $w(\mathcal{I}^c)$ cannot be considered a zero vector, then set $q_{j+1}(\mathcal{I}) = 0$, $q_{j+1}(\mathcal{I}^c) = w(\mathcal{I}^c)/h_{j+1j}$, $\mathcal{J} = \mathcal{J} \cup \{j + 1\}$. Set $j = j + 1$, and go to Step 2.
10. Now $w(\mathcal{I}^c) = 0$. **STOP**. (An invariant subspace of A has been found!)

Denote $Q_k = (q_1, q_2, \dots, q_k) \in \mathbb{C}^{n \times j}$, and H_k is the $k \times k$ upper Hessenberg matrix whose (i, j) th entry is defined by Algorithm 10.1 and those that have not been explicitly defined are set to zeros.

The *exact arithmetic version* of Algorithm 10.1: All computations and comparisons are done in exact arithmetic (and thus re-orthogonalization Steps 4 and 8 can be safely deleted).

(10.1)

It can be seen that

Theorem 10.1 *At the conclusion of Algorithm 10.1 (exact arithmetic version), if the process ends by the **STOP** at Step 2, then*

1. $\dim \mathcal{K}_k(A, b) = k$, and $\{q_1, q_2, \dots, q_k\}$ is \mathcal{I} -sub-orthonormal and a basis of $\mathcal{K}_k(A, b)$;
2. $AQ_k = Q_k H_k + h_{k+1k} q_{k+1} e_k^*$.

*If, however, the algorithm concludes by the **STOP** at Step 6 or 10, then*

1. $\dim \mathcal{K}_j(A, b) = \dim \mathcal{K}_{j+1}(A, b) = \dots = j$, and $\{q_1, q_2, \dots, q_j\}$ is \mathcal{I} -sub-orthonormal and a basis of $\mathcal{K}_j(A, b)$;

2. $AQ_j = Q_jH_j$.

Next we consider a sub-orthogonalized block Arnoldi process. Let $B \in \mathbb{C}^{n \times m}$ (whose column may be linearly dependent). Algorithm 10.2 computes an \mathcal{I} -sub-orthonormal basis of $\mathcal{K}_k(A, B)$. It combines the feature of Algorithm 10.1 and Ruhe's variant of the block Arnoldi method [35].

In Algorithm 10.2, N_j records the number of basis vectors q_i generated essentially from A^jB (those columns of A^jB that are in $\mathcal{K}_j(A, B)$ will be deflated during the sub-orthogonalization process of course), N is the number of basis vectors q_i at any given point, $\{q_i\}_{i=J_1}^{J_2}$ are basis vectors computed from $A^{j-1}B$ and will be applied by A in sequence to compute those basis vectors from A^jB .

Algorithm 10.2 Sub-orthogonalized Block Arnoldi Process:

Given $A \in \mathbb{C}^{n \times n}$, $B \in \mathbb{C}^{n \times m}$, the number k of Arnoldi steps.

1. Run Algorithm 9.2 on B , and record its outputs N , \mathcal{J} , and q_1, q_2, \dots, q_N . Assume $N \geq 1$; Otherwise $B = 0$ and no need to go any further. Set $j = 1$, $J_2 = 0$.
2. If $j \geq k$, then **STOP**; Otherwise $J_1 = J_2 + 1$, $J_2 = N$, $J = J_1 + 1$, and $N_j = 0$.
3. If $J > N$ and $N_j = 0$, then **STOP**. (An invariant subspace of A has been found!)
4. If $J > N$ (and now must $N_j \geq 1$), then $j = j + 1$, and go to Step 2; Otherwise $w = Aq_J$.
5. Orthogonalize $w(\mathcal{I})$ against $\{q_i(\mathcal{I})\}_{i=1}^N$:

$$h_{iJ} = q_i(\mathcal{I})^* w(\mathcal{I}), \quad w = w - h_{iJ} q_i, \quad \text{for } i = 1, 2, \dots, N.$$

6. If necessary, re-orthogonalize $w(\mathcal{I})$ against $\{q_i(\mathcal{I})\}_{i=1}^N$.
7. If $w(\mathcal{I})$ cannot be considered zero, then set $N = N + 1$, $N_j = N_j + 1$, $h_{NJ} = \|w(\mathcal{I})\|_2$, $q_N = w/h_{NJ}$. Set $J = J + 1$, and go to Step 3.
8. Now $w(\mathcal{I}) = 0$. If $w(\mathcal{I}^c)$ can be considered zero also, then set $J = J + 1$ and go to Step 3.
9. Orthogonalize $w(\mathcal{I}^c)$ against all $\{q_\ell(\mathcal{I}^c), \ell \in \mathcal{J}\}$:

$$h_{\ell J} = q_\ell(\mathcal{I}^c)^* w(\mathcal{I}^c), \quad w(\mathcal{I}^c) = w(\mathcal{I}^c) - h_{\ell J} q_\ell(\mathcal{I}^c), \quad \text{for each } \ell \in \mathcal{J}.$$

10. If necessary, re-orthogonalize $w(\mathcal{I}^c)$ against all $\{q_\ell(\mathcal{I}^c), \ell \in \mathcal{J}\}$.
11. If $w(\mathcal{I}^c)$ cannot be considered a zero vector, then set $N = N + 1$, $N_j = N_j + 1$, $q_N(\mathcal{I}^c) = 0$, $h_{NJ} = \|w(\mathcal{I}^c)\|_2$, $q_N(\mathcal{I}^c) = w(\mathcal{I}^c)/h_{NJ}$, $\mathcal{J} = \mathcal{J} \cup \{N\}$.
12. Set $J = J + 1$, and go to Step 3.

We define an exact arithmetic version of Algorithm 10.2, similar to (10.1).

Theorem 10.2 *At the conclusion of Algorithm 10.2 (exact arithmetic version), if the process ends by the **STOP** at Step 2, then*

1. $\dim \mathcal{K}_k(A, B) = N$, and $\{q_1, q_2, \dots, q_N\}$ is \mathcal{I} -sub-orthonormal and a basis of $\mathcal{K}_k(A, B)$;
2. $AQ_{N-N_{k-1}} = Q_N H_N$, where H_N is $N \times (N - N_{k-1})$.

*If, however, the algorithm concludes by the **STOP** at Step 3, then*

1. $\dim \mathcal{K}_j(A, B) = \dim \mathcal{K}_{j+1}(A, B) = \dots = N$, and $\{q_1, q_2, \dots, q_N\}$ is \mathcal{I} -sub-orthonormal and a basis of $\mathcal{K}_j(A, B)$;
2. $AQ_N = Q_N H_N$, where H_N is $N \times N$.

REMARK 10.1 Both Algorithms 10.1 and 10.2 were stated with their numerical implementations in mind. At a few places, one has to decide whether to do a re-orthogonalization or consider a vector zero. Criteria for those decision makings can very much be adopted from what we mentioned in Remark 9.1.

11 Issues with Sub-orthogonalition Process

A major issue with the sub-orthogonalization processes described in Sections 9 and 10 is their potential element growth in a floating point environment and thus possibly unstable. Let us explain why. Recall this common feature: the next basis vector q is from a vector w after being sub-orthogonalized and sometimes re-sub-orthogonalized against the basis vectors already computed, and

$$q = \begin{cases} w/\|w(\mathcal{I})\|_2, & \text{if } w(\mathcal{I}) \text{ is not considered zero,} \\ \text{more work to do,} & \text{otherwise.} \end{cases}$$

Possible element growth comes from the situation when $\|w(\mathcal{I})\|_2$ is so tiny, but not tiny enough to be safely considered zero, because then $q(\mathcal{I}^c)$ could be huge. Specifically let us look at (9.2) and (9.3) in Remark 9.1. If one of the two inequality is satisfied, we regard $w(\mathcal{I}) = 0$. Suppose that both are violated with (9.3) barely missed in the sense that $\|w(\mathcal{I})\|_2 = \epsilon\|w(\mathcal{I}^c)\|_2(1 + \delta)$, where $0 < \delta \ll 1$. Then $\|q\|_2 \geq \|q(\mathcal{I}^c)\|_2 \approx 1/\epsilon$, a huge number. In summary instability occurs if the case

$$\boxed{w(\mathcal{I}) \text{ is not considered zero, and } \|w(\mathcal{I})\|_2 \ll \|w(\mathcal{I}^c)\|_2.} \quad (11.1)$$

happens during the process. If, however, (11.1) does not happen during a particular run, then that particular run is stable.

All (sub-)orthogonalization was stated with respect to the standard inner-product so far. Obviously this standard inner-product can be replaced by any (definite) inner product. It's tempting to replace it by an indefinite inner product, too, but an additional complication may occur, i.e., the inner product of a non-zero vector during the process may be zero – a breakdown situation that may not be curable.

This sub-orthogonalization, though not called by this name, was implicitly described by Su and Craig [38] with K -inner product, where K is positive definite. Su's and Craig's algorithm is generic, i.e., assuming no deflation ever occurs. Recently Bai and Su [9] uses a similar sub-orthogonalized Arnoldi process to solve quadratic eigenvalue problems, though again not called by this name.

Could there be a sub-orthogonalized Lanczos process? It appears unlikely that there is any with short-recurrences. We shall explain. Recall that a short-recurrence is a result of the duality between A and A^* :

$$\langle w_i, Av_j \rangle = \langle A^* w_i, v_j \rangle$$

which is zero if i less than j is sufficiently apart from j . Here matrix A is treated as a whole. With MATLAB notation, for \mathcal{I} -subvector orthogonalization,

$$\langle w_i(\mathcal{I}), A(\mathcal{I}, :)v_j \rangle = \langle A(\mathcal{I}, :)^* w_i, v_j \rangle$$

which may not be zero, no matter how far apart i and j are. Even so, one can always have a sub-orthogonalized Lanczos process but with *full-length recurrences*, and then again it won't not be very attractive computationally *vs.* sub-orthogonalized Arnoldi process.

References

- [1] J. I. ALIAGA, D. L. BOLEY, R. W. FREUND, AND V. HERNÁNDEZ, *A Lanczos-type method for multiple starting vectors*, *Math. Comp.*, 69 (2000), pp. 1577–1601.
- [2] B. D. O. ANDERSON AND S. VONGPANITLERD, *Network Analysis and Synthesis*, Prentice-Hall, Englewood Cliffs, NJ, 1973.
- [3] A. C. ANTOULAS AND D. C. SORENSEN, *Approximation of large-scale dynamical system: an overview*, technical report, Department of Electrical and Computer Engineering, Rice University, Houston, Texas, August 2001. Presented at the MTNS, June 19-23, 2000.
- [4] W. E. ARNOLDI, *The principle of minimized iterations in the solution of the matrix eigenvalue problem*, *Quarterly of Applied Mathematics*, 9 (1951), pp. 17–29.
- [5] Z. BAI, *Error analysis of the lanczos algorithm for the nonsymmetric eigenvalue problem*, *Mathematics of Computation*, 62 (1994), pp. 209–226.
- [6] Z. BAI, *Krylov subspace techniques for reduced-order modeling of large-scale dynamical systems*, *Applied Numerical Mathematics*, 43 (2002), pp. 9–44.
- [7] Z. BAI AND R. FREUND, *A partial padé-via-lanczos method for reduced-order modeling*, *Linear Algebra and Its Applications*, 332–334 (2001), pp. 139–164.
- [8] Z. BAI, R. SLONE, W. SMITH, AND Q. YE, *Error bound for reduced system model by padé approximation via the lanczos process*, *IEEE Transactions on Computer-Aided Design*, 18 (1999), pp. 133–141.
- [9] Z. BAI AND Y. SU, *A second-order Krylov subspace and its application to the quadratic eigenvalue problem*. Department of Computer Science, University of California at Davis, September 2003.
- [10] Z. BAI AND Q. YE, *Error estimation of the padé approximation of transfer functions via the Lanczos process*, *Electronic Transactions on Numerical Analysis*, 7 (1998), pp. 1–17.
- [11] Å. BJÖRCK, *Numerical Methods for Least Squares Problems*, SIAM, Philadelphia, 1996.
- [12] T. CHAN, *Rank-revealing QR factorizations*, *Linear Algebra and its Application*, 88/89 (1987), pp. 67–82.
- [13] J. DEMMEL, *Applied Numerical Linear Algebra*, SIAM, Philadelphia, 1997.
- [14] P. FELDMAN AND R. W. FREUND, *Efficient linear circuit analysis by Padé approximation via the Lanczos process*, *IEEE Trans. Computer-Aided Design*, 14 (1995), pp. 639–649.
- [15] L. FOSTER, *Rank and null space calculations using matrix decomposition without column interchanges*, *Linear Algebra and its Applications*, 74 (1986), pp. 47–71.
- [16] R. W. FREUND, *Krylov-subspace methods for reduced-order modeling in circuit simulation*, *Journal of Computational and Applied Mathematics*, 123 (2000), pp. 395–421.
- [17] ———, *Model reduction methods based on krylov subspaces*, Numerical Analysis Manuscript 03-4-01, Bell Laboratories, Murray Hill, New Jersey, January 2003.
- [18] R. W. FREUND AND P. FELDMANN, *The SyMPVL algorithm and its applications to interconnect simulation*, in Proc. 1997 International Conference on Simulation of Semiconductor Processes and Devices, Piscataway, New Jersey, 1997, IEEE, pp. 113–116.

- [19] R. W. FREUND, M. H. GUTKNECHT, AND N. M. NACHTIGAL, *An implementation of the look-ahead Lanczos algorithm for non-Hermitian matrices*, SIAM J. Sci. Comput., 14 (1993), pp. 137–158.
- [20] K. GALLIVAN, E. GRIMME, AND P. VAN DOOREN, *Asymptotic waveform evaluation via a Lanczos method*, Appl. Math. Lett., 7 (1994), pp. 75–80.
- [21] I. GOHBERG, P. LANCASTER, AND L. RODMAN, *Matrix Polynomials*, Academic Press, New York, 1982.
- [22] ———, *Matrices and Indefinite Scalar Products*, Birkhäuser, Boston, 1983.
- [23] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, Johns Hopkins University Press, Baltimore, Maryland, 3rd ed., 1996.
- [24] W. B. GRAGG, *Matrix interpretations and applications of the continued fraction algorithm*, Rocky Mountain J. Math., 4 (1974), pp. 213–225.
- [25] E. J. GRIMME, *Krylov Projection Methods For Model Reduction*, PhD thesis, University of Illinois at Urbana-Champaign, Urbana, Illinois, 1997.
- [26] L. HOFFNUNG, R.-C. LI, AND Q. YE, *Krylov type subspace methods for matrix polynomials*, Linear Algebra and its Applications, (2003). to appear.
- [27] T.-M. HWANG, W.-W. LIN, AND E. K. YANG, *Rank revealing LU factorization*, Linear Algebra and Its Applications, 175 (1992), pp. 115–141.
- [28] C. LANCZOS, *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators*, Journal of Research of the National Bureau of Standards, 45 (1950), pp. 255–282.
- [29] ———, *Solution of systems of linear equations by minimized iterations*, Journal of Research of the National Bureau of Standards, 49 (1952), pp. 33–53.
- [30] R.-C. LI AND Q. YE, *A Krylov subspace method for quadratic matrix polynomials with application to constrained least squares problems*, SIAM J. Matrix Anal. Appl., (2003). to appear.
- [31] A. ODABASIOGLU, M. CELIK, AND L. T. PILEGGI, *PRIMA: passive reduced-order interconnect macromodeling algorithm*, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 17 (1998), pp. 645–654.
- [32] B. N. PARLETT, *Reduction to tridiagonal form and minimal realization*, SIAM Journal on Matrix Analysis and Applications, 13 (1992), pp. 567–593.
- [33] B. N. PARLETT, D. R. TAYLOR, AND Z. A. LIU, *A look-ahead Lanczos algorithm for unsymmetric matrices*, Mathematics of Computation, 44 (1985), pp. 105–124.
- [34] L. T. PILLAGE AND R. A. ROHRER, *Asymptotic waveform evaluation for timing analysis*, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 9 (1990), pp. 352–366.
- [35] A. RUHE, *Implementation aspects of band Lanczos algorithms for computation of eigenvalues of large sparse symmetric matrices*, Mathematics of Computation, 33 (1979), pp. 680–687.
- [36] Y. SAAD, *Iterative Methods for Sparse Linear Systems*, PWS Publishing Company, Bpston, MA, 1996.
- [37] G. W. STEWART, *Rank degeneracy*, SIAM Journal on Scientific and Statistical Computing, 5 (1984), pp. 403–413.

- [38] T.-J. SU AND R. R. CRAIG, *Model reduction and control of flexible structures using Krylov vectors*, J. Guidance, Control, and Dynamics, 14 (1991), pp. 260–267.
- [39] F. TISSEUR AND K. MEERBERGEN, *The quadratic eigenvalue problem*, SIAM Review, 43 (2001), pp. 235–386.
- [40] C. E. VILLEMAGNE AND R. E. SKELTON, *Model reduction using a projection formulation*, Int. J. Control, 46 (1987), pp. 2141–2169.
- [41] Q. YE, *A convergence analysis for nonsymmetric Lanczos algorithms*, Mathematics of Computation, 56 (1991), pp. 677–691.
- [42] ———, *A breakdown-free variation of the nonsymmetric Lanczos algorithms*, Mathematics of Computation, 62 (1994), pp. 179–207.