Model reduction via an LFT-based explicitly restarted nonsymmetric Lanczos algorithm

Vasilios Papakos* and Imad M. Jaimoukha†

Abstract

The nonsymmetric Lanczos algorithm, which belongs to the class of Krylov subspace methods, is increasingly being used for model reduction of large scale systems of the form $f(s) = c^T(sI - A)^{-1}b$, to exploit the sparse structure and reduce the computational burden. However, a good approximation is, usually, achieved only with relatively high order reduced models. Moreover, the computational cost of the Lanczos algorithm is dominated by the full rebiorthogonalization procedure, which is necessary because the Lanczos vectors tend to lose their biorthogonality. A method based on linear fractional transformations (LFTs) is proposed to compute a reduced mth order model by applying k "small" Lanczos algorithms with m/k steps each; thus reducing the computational cost and storage requirements. Applying this method, one can compute a tridiagonal similar realization of $f(s)$ and when combined with conventional model reduction techniques, a minimal or reduced realization.

1 Introduction

Model reduction is widely employed to deal with large scale linear systems, which are usually sparse and large dimensional. In this paper, stable, linear single-input single-output (SISO) systems are considered, described by the equations

$$\dot{x}_1(t) = A_1x_1(t) + b_1u(t), \quad y_1(t) = c_1^Tx_1(t), \tag{1.1}$$

where $x_1(t) \in \mathbb{R}^n$ denotes the state vector, $u(t)$ and $y_1(t)$ the scalar input and output, respectively. The matrix $A_1 \in \mathbb{R}^{n\times n}$ is assumed to be of large dimension and sparse and $b_1, c_1 \in \mathbb{R}^n$. The subscript 1 is used for consistency of notation with subsequent sections. The transfer function of (1.1) is denoted as $f_1(s) = c_1^T(sI_n - A_1)^{-1}b_1 \stackrel{s}{=} (A_1, b_1, c_1^T, 0)$ where I_n denotes the $n \times n$ identity matrix. The objective of model reduction is to produce a stable approximate rth order model,

$$\dot{x}_{1,r}(t) = A_{1,r}x_{1,r}(t) + b_{1,r}u(t), \quad y_{1,r}(t) = c_{1,r}^Tx_{1,r}(t),$$

in which $x_{1,r}(t) \in \mathbb{R}^r$ and $r \ll n$. Conventional model reduction techniques, such as balanced truncation [4], are not suitable for large scale systems because they require $O(n^3)$ operations, and hence, for large n, they are not practical. The Lanczos algorithm, which belongs to the class of Krylov subspace projection methods, is suitable for large sparse matrix computations, since only matrix-vector multiplications and inner products in the large dimension are involved. A Krylov subspace $\mathcal{K}_m(A, v)$, given a vector v and a matrix A, is defined as $\mathcal{K}_m(A, v) = \text{span}\{v, Av, \ldots, A^{m-1}v\}$. The nonsymmetric Lanczos algorithm is an oblique projection method and simultaneously constructs a pair of biorthogonal bases V_m and W_m for two Krylov subspaces such that

*Center for Process Systems Engineering, Imperial College, London SW7 2BY, UK. (v.papakos@ic.ac.uk).

†Department of Electrical and Electronic Engineering, Imperial College, London SW7 2BT, UK. (i.jaimouka@ic.ac.uk).

$W_m^T V_m = I_m$. However, in finite precision arithmetic, the bases V_m and W_m tend to lose their biorthogonality. To remedy this, either full rebiorthogonalization [6] against the previous vectors is employed at each step, or a compromise known as maintaining semiorthogonality [3] by monitoring the loss of biorthogonality. Alternatively, to avoid rebiorthogonalization, more steps may be taken in order to deal with the presence of spurious eigenvalues [2].

In this paper, a method employing linear fractional transformations (LFTs) is described, to obtain a reduced mth order model using k "small" Lanczos algorithm of size m_j each, where $\sum_{j=1}^{k} m_j = m$. A modified matrix of the form $A_1 - F_j G_j^T$ is used at each iteration, but sparsity is preserved since it is not explicitly formed. The matrices F_j and G_j contain $2j$ vectors each, where j is the iteration number. Applying this method, an exact decomposition of $f_1(s)$ in terms of smaller order models can be computed, if $\sum_{j=1}^{k} m_j = n$. Approximating these reduced order realizations using balanced truncation, one can compute an approximation of $f_1(s)$. This procedure, therefore, avoids some of the drawbacks of the nonsymmetric Lanczos algorithms. Model reduction using the Lanczos algorithm is outlined in Section 2 and the theoretical foundation for the proposed method is provided in Section 3. Description of the algorithms to compute Lanczos-derived realizations and minimal or reduced realizations are given in Sections 4 and 5, respectively, along with numerical examples. Finally, conclusions are drawn in Section 6.

2 The nonsymmetric Lanczos algorithm for model reduction

The nonsymmetric Lanczos algorithm with full reorthogonalization [6] is employed to construct a pair of biorthogonal bases, $V_{1,m_1}, W_{1,m_1} \in \mathbb{R}^{n \times m_1}$, for the Krylov subspaces, $\mathcal{K}_{m_1}(A_1, b_1)$ and $\mathcal{L}_{m_1}(A_1^T, c_1^T)$, which span the first m_1 columns of the controllability and observability subspaces, respectively. The algorithm may break down[5]; however, in this paper, it will be assumed that no breakdowns occur. The following equations, referred to as the Lanczos equations [5], hold,

$$A_1 V_{1,m_1} = V_{1,m_1} A_{1,m_1} + \tilde{v}_{1,m_1} \tilde{a}_{1,m_{1_V}}^T, \qquad b_1 = V_{1,m_1} b_{1,m_1}, \tag{2.2}$$

$$A_1^T W_{1,m_1} = W_{1,m_1} A_{1,m_1}^T + \tilde{w}_{1,m_1} \tilde{a}_{1,m_{1_W}}^T, \qquad c_1 = W_{1,m_1} c_{1,m_1}, \tag{2.3}$$

where $A_{1,m_1} = W_{1,m_1}^T A_1 V_{1,m_1}$ is a tridiagonal matrix. Then, the m_1th order model is given by $f_{1,m_1}(s) \stackrel{s}{=} (W_{1,m_1}^T A_1 V_{1,m_1}, W_{1,m_1}^T b_1, c_1^T V_{1,m_1}, 0) = (A_{1,m_1}, b_{1,m_1}, c_{1,m_1}^T, 0)$.

3 An exact decomposition of state space systems

The following lemma shows that $f(s)$ can be decomposed using and LFT-based formulation. This results is the starting point for deriving an exact decomposition of $f(s)$.

Lemma 3.1. *Let $f_{1,m_1}(s) \stackrel{s}{=} (A_{1,m_1}, b_{1,m_1}, c_{1,m_1}^T, 0)$ denote the reduced order approximation of $f_1(s)$ obtained after m_1 steps of the nonsymmetric Lanczos algorithm have been taken, so that equations (2.2)–(2.3) hold. Then,*

$$f_1(s) = \mathcal{F}_l(F_{1,m_1}(s), f_2(s)) \tag{3.4}$$

where $\mathcal{F}_l(\cdot,\cdot)$ denotes the lower linear fractional transformation (LFT) and

$$F_{1,m_1}(s) = \begin{bmatrix} F_{1,m_1}^{(11)}(s) & F_{1,m_1}^{(12)}(s) \\ F_{1,m_1}^{(21)}(s) & F_{1,m_1}^{(22)}(s) \end{bmatrix} \stackrel{s}{=} \left[\begin{array}{c|cc} A_{1,m_1} & b_{1,m_1} & \tilde{a}_{1,m_{1_W}} \\ \hline c_{1,m_1}^T & 0 & 0 \\ \tilde{a}_{1,m_{1_V}}^T & 0 & 0 \end{array}\right], \tag{3.5}$$

$$f_2(s) \stackrel{s}{=} \left[\begin{array}{c|c} A_1 - \tilde{v}_{1,m_1} \tilde{a}_{1,m_{1_V}}^T W_{1,m_1}^T - V_{1,m_1} \tilde{a}_{1,m_{1_W}} \tilde{w}_{1,m_1}^T & \tilde{v}_{1,m_1} \\ \hline \tilde{w}_{1,m_1}^T & 0 \end{array}\right] = \left[\begin{array}{c|c} A_2 & b_2 \\ \hline c_2^T & 0 \end{array}\right]. \tag{3.6}$$

Proof. Expanding (3.4) (see [8]) gives

$$\mathcal{F}_l(F_{1,m_1}(s), f_2(s)) \stackrel{s}{=} \left[\begin{array}{cc|c} A_{1,m_1} & \tilde{a}_{1,m_{1_W}}\tilde{w}_{1,m_1}^T & b_{1,m_1} \\ \tilde{v}_{1,m_1}\tilde{a}_{1,m_{1_V}}^T & A_2 & 0 \\ \hline c_{1,m_1}^T & 0 & 0 \end{array}\right] \tag{3.7}$$

and applying a similarity transformation with

$$T = \left[\begin{array}{cc} I_m & -W_{1,m_1}^T \\ V_{1,m_1} & I_n - V_{1,m_1}W_{1,m_1}^T \end{array}\right], \quad T^{-1} = \left[\begin{array}{cc} 0 & W_{1,m_1}^T \\ -V_{1,m_1} & I_n \end{array}\right]$$

to (3.7) gives

$$\mathcal{F}_l(F_{1,m_1}(s), f_2(s)) \stackrel{s}{=} \left[\begin{array}{cc|c} A_{1,m_1} & \tilde{a}_{1,m_{1_W}}\tilde{w}_{1,m_1}^T & b_{1,m_1} \\ 0 & A_1 & b_1 \\ \hline 0 & c_1^T & 0 \end{array}\right] = \left[\begin{array}{c|c} A_1 & b_1 \\ \hline c_1^T & 0 \end{array}\right] \stackrel{s}{=} f_1(s),$$

where the last equation follows by removing the unobservable part. □

Corollary 3.1. *Let $f_{2,m_2}(s) \stackrel{s}{=} (A_{2,m_2}, b_{2,m_2}, c_{2,m_2}^T, 0)$ denote the reduced order approximation of $f_2(s)$ obtained after m_2 steps of the nonsymmetric Lanczos algorithm have been taken, and the corresponding Lanczos equation, similar to (2.2)–(2.3), hold. Then,*

$$f_2(s) = \mathcal{F}_l(F_{2,m_2}(s), f_3(s)), \tag{3.8}$$

$$f_1(s) = \mathcal{F}_l(F_{1,m_1}(s) \star F_{2,m_2}(s), f_3(s)) = \mathcal{F}_l(F_{12,m_{12}}(s), f_3(s)) \tag{3.9}$$

where $F_{2,m_2}(s)$, $f_3(s)$ are defined similarly to (3.5), (3.6), respectively, and

$$F_{12,m_{12}}(s) = F_{1,m_1}(s) \star F_{2,m_2}(s) \stackrel{s}{=} \left[\begin{array}{cc|cc} A_{1,m_1} & \tilde{a}_{1,m_{1_W}}c_{2,m_2}^T & b_{1,m_1} & 0 \\ b_{2,m_2}\tilde{a}_{1,m_{1_V}}^T & A_{2,m_2} & 0 & \tilde{a}_{2,m_{2_W}} \\ \hline c_{1,m_1}^T & 0 & 0 & 0 \\ 0 & \tilde{a}_{2,m_{2_V}}^T & 0 & 0 \end{array}\right] \tag{3.10}$$

and $(\cdot \star \cdot)$ denotes the Redheffer star product [8].

Proof. Equation (3.8) follow directly from Lemma 3.1. Substituting (3.8) into (3.4) gives

$$f_1(s) = \mathcal{F}_l(F_{1,m_1}(s), \mathcal{F}_l(F_{2,m_2}(s), f_3(s))) = \mathcal{F}_l(F_{1,m_1}(s) \star F_{2,m_2}(s), f_3(s)) = \mathcal{F}_l(F_{12,m_{12}}(s), f_3(s))$$

as described in [8]. □

The following theorem shows that any mth order Lanczos approximation of $f_1(s)$ can be constructed by applying k times the Lanczos algorithm with m_j steps each, such that $\sum_{j=1}^k m_j = m$. Furthermore, exploiting this property, one can also compute an exact decomposition of $f_1(s)$ in terms of reduced order models using the Lanczos algorithm.

Theorem 3.1. *Let* $f_{1,m}(s) \stackrel{s}{=} (A_{1,m}, b_{1,m}, c_{1,m}^T, 0)$ *denote the reduced order approximation of* $f_1(s)$ *obtained after* $m = m_1 + m_2$ *steps of the nonsymmetric Lanczos algorithm have been taken, and the corresponding Lanczos equations hold. Then,*

1. *the realizations* $F_{12,m_{12}}(s)$ *in (3.10) and* $F_{1,m}(s) \stackrel{s}{=} \left[\begin{array}{c|cc} A_{1,m} & b_{1,m} & \tilde{a}_{1,m_W} \\ \hline c_{1,m}^T & 0 & 0 \\ \tilde{a}_{1,m_V}^T & 0 & 0 \end{array}\right]$ *are identical.*

2. *The realizations of* $f_{12,m_{12}}(s)$ *and* $f_{1,m}(s)$ *are also identical, where* $f_{12,m_{12}}(s) = F_{12,m_{12}}^{(11)}(s)$.

3. *Assuming that* m *steps of the nonsymmetric Lanczos algorithm can be taken without breakdown and* $m_1+m_2+\cdots+m_k = m$, *then* $f_{1\cdots k,m_{1\cdots k}}(s) = f_{1,m}(s)$ *where* $f_{1\cdots k,m_{1\cdots k}}(s) = F_{1\cdots k,m_{1\cdots k}}^{(11)}(s)$. *Furthermore, if* $m = n$, $f_1(s) = f_{1\cdots k,m_{1\cdots k}}(s) = f_{1,m}(s)$.

The proof can be derived from Corollary 3.1 and the Lanczos equations. An interpretation of part 3 of Theorem 3.1 is that a similar tridiagonal realization of $f(s)$ can be constructed, provided that no breakdowns occur. Consequently, one can draw conclusions about its minimality, as shown in the following corollary.

Corollary 3.2. *A realization of* $f_1(s)$ *is minimal if and only if the realization of* $f_{1\cdots k,m_{1\cdots k}}(s)$ *as derived in Theorem 3.1 is minimal.*

Proof. In the third part of Theorem 3.1, the realizations of $f_{1,m}(s)$ and $f_1(s)$ are similar, which implies that if $f_1(s)$ is minimal, $f_{1,m}(s)$ is also minimal. Therefore $f_{1\cdots k,m_{1\cdots k}}(s)$ is minimal since $f_{1\cdots k,m_{1\cdots k}}(s) = f_{1,m}(s)$. Similarly, if $f_{1\cdots k,m_{1\cdots k}}(s)$ is minimal, $f_{1,m}(s)$ is also minimal and, hence, $f_1(s)$ is minimal. Note that the dimensions of the state space realizations are the same. □

The aforementioned theoretical results can be applied to derive the solution for two problems; firstly, to compute a Lanczos reduced order model in terms of a number of low order models instead of directly reducing the original one, and secondly, to reduce $f_1(s)$ via its decomposition in a number of low order models. In exact arithmetic, the first problem gives an exact solution, but for the second an approximation is obtained. In the following section, these two problems are described along with numerical examples.

4 Computing Lanczos approximations

The third part of Theorem 3.1 can be used for computing an mth order Lanczos approximation, where $\sum_{j=1}^k m_j = m$. The matrix A_{j+1} is not formed explicitly but only the necessary vectors are stored; namely, A_{j+1} can be expressed in an outer product form as $A_{j+1} = A_1 - F_j G_j^T$, where F_j and G_j contain $2j$ vectors each; a property following from the structure of the Lanczos quantities. To avoid explicitly forming A_{j+1}, an outer product version of the standard Lanczos algorithm is employed. A significant saving in computational cost and storage results from rebiorthogonalization of the bases. Specifically, when constructing $f_{1,m}(s)$, each new vector is rebiorthogonalized against all previous ones up to m, whereas for $f_{1\cdots k,m_{1\cdots k}}(s)$ only up to m_j vectors are used for each step. This saving comes at the expense of performing the outer product Lanczos algorithm; however, the latter is much cheaper.

Another important issue is that although local biorthogonality holds, i.e., $W_{j,m_j}^T V_{j,m_j} = I_{m_j}$, global biorthogonality may be lost, i.e., $W_{j_1,m_{j_1}}^T V_{j_2,m_{j_1}} \neq I_{m_j}$ for $j_1 \neq j_2$ and $m_{j_1} = m_{j_2} = m_j$. As an example, assume that $\sum_{j=1}^k m_j = m_L$; also, for simplicity, let $m_j = m$, hence $m_L = km$. Denote the dominant cost of applying m_L

Table 1: Computational costs and CPU times for an example with 1000 states.

Algorithm	Flops	CPU time
Standard Lanczos (L)	6.544e+8	1.916e+2
LFT-based Lanczos (LFT)	1.894e+8	5.110e+1
Ratio (L)/(LFT)	3.46	3.75

steps of the standard Lanczos algorithm by K_L, and the dominant cost of the proposed method by K_{LFT}; then, $K_L = kK_{\text{LFT}}$. Therefore, the cost of the LFT-based approach is k times smaller than the standard approach. This result is verified with a state space model with $n = 1000$ states, for which the state matrix A_1 arises from a three dimensional simulation model of an oil reservoir [1] and the vectors b_1, c_2 contain normally distributed random numbers. The values of the parameters used are $m = 100$, $k = 4$ and $m_L = mk = 400$. The computational cost and CPU times are shown in Table 1, where the ratio of flops and CPU time is close to k. The error norms between the resulting realizations are

$$\|f_{1\cdots k,m_{1\cdots k}}(s) - f_{1,m_L}(s)\|_\infty = 4.200 \cdot 10^{-5}, \qquad \|f_{1\cdots k,m_{1\cdots k}}(s) - f_{1,m_L}(s)\|_2 = 9.522 \cdot 10^{-5}$$

where $f_{1\cdots k,m_{1\cdots k}}(s)$ denotes the model obtained from the LFT-based approach, and $f_{1,m_L}(s)$ the model from the standard Lanczos algorithm. The error norms indicate that there is a small difference between the two realization; the main reason being the absence of global biorthogonality. Although in theory the two realizations should be identical, in floating point arithmetic, an approximation is computed. It should be noted that this an illustrative example; further investigation may shed light to the problem of choosing suitable values for m and k in order to minimize the error norms, and, possibly, employing a different rebiorthogonalization scheme.

5 Approximate model reduction of state space models

The second problem is to compute approximate reduced order models in terms of a number of low order models instead of directly reducing the original one. Following a similar notation for the reduced models, the idea is to combine $F_{1\cdots j-1,r_{1\cdots j-1}}(s)$ with $F_{j,m_j}(s)$ and, then, reduce the resulting realization, where $\sum_{j=1}^{k} m_j = n$. As an example, assume that $m_j = m$, hence $n = km$. Denote the dominant cost of directly reducing $f_1(s)$ with K_{MR}, and the dominant cost of the proposed method by $K_{\text{LFT-MR}}$. For the numerical example used in Section 4 with $n = 1000$, $m = 100$ and $k = 10$, the average value of r_j was $r_j = m/2$; then,

$$K_{\text{MR}} = \frac{mk^2}{2m + 3k^2 + k} K_{\text{LFT-MR}} = 19.61 K_{\text{LFT-MR}}$$

where r_j denotes the order of $F_{1\cdots j-1,r_{1\cdots j-1}}(s)$. The computational cost and CPU times are shown in Table 2. The tolerance for model reduction was set equal to the machine precision, $\epsilon = 2.22 \cdot 10^{-16}$, hence a minimal realization was computed. This problem is closely related to LFT model reduction and controller reduction and may be carried out more efficiently than presented here; however, to illustrate the idea balanced truncation was used. The SLICOT library [7] was used for model reduction; this explains why the ratio of the CPU times is much smaller than expected. The error norms between the resulting realizations are

$$\|f_{1\cdots k,r_{1\cdots k}}(s) - f(s)\|_\infty = 7.560 \cdot 10^{-5} \qquad \|f_{1,r}(s) - f(s)\|_\infty = 3.498 \cdot 10^{-9}$$
$$\|f_{1\cdots k,r_{1\cdots k}}(s) - f(s)\|_2 = 1.535 \cdot 10^{-4} \qquad \|f_{1,r}(s) - f(s)\|_2 = 9.057 \cdot 10^{-9}$$

where $f_{1\cdots k,r_{1\cdots k}}(s)$ denotes the model obtained from the LFT-based approach, and $f_{1,r}(s)$ the model obtained from balanced truncation on $f(s)$. The orders of $f_{1\cdots k,r_{1\cdots k}}(s)$ and $f_{1,r}(s)$ are 56 and 30, respectively. The main reason for the difference in the error norms is, as before, the absence of global biorthogonality. Also, $f_{1\cdots k,r_{1\cdots k}}(s)$ is not necessarily stable, since the Lanczos algorithm does not guarantee stability. Note that $f_{1,r}(s)$ was computed for comparison purposes; for a system of 1000 states, balanced truncation is not practical.

Table 2: Computational costs and CPU times for an example with 1000 states.

Algorithm	Flops	CPU time
Standard model reduction (MR)	3.000e+10	4.738e+2
LFT-based Lanczos model reduction (LFT-MR)	1.744e+09	1.890e+2
Ratio (MR)/(LFT-MR)	17.20	2.51

6 Conclusions

A method combining the Lanczos algorithm and LFTs was developed to obtain an exact approximation of a state space model. The idea was to compute a number of "small" Lanczos algorithms instead of a "big" one. Avoiding global rebiorthogonalization that dominates the computational cost and storage requirements of the Lanczos algorithm, is the main advantage. Computation of Lanczos-derived reduced order realizations and approximate model reduction are two potential application of the proposed method. Algorithms and preliminary numerical results were presented. Further investigation regarding the values of the two parameters, m and k, and their effect on the loss of global biorthogonality in floating point arithmetic is required.

References

[1] Nolan A. Sherman and Associates. Oil reservoir modeling. Matrix Market (http://math.nist.gov/MatrixMarket/index.html), Autumn 1984.

[2] J. Cullum and R. Willoughby. A generalized nonsymmetric Lanczos procedure. *Comp. Phys. Comm.*, 53:19–48, 1989.

[3] D. Day. An efficient implementation of the nonsymmetric Lanczos algorithm. *SIAM J. Matrix Anal. Appl.*, 18(3):566–589, 1997.

[4] B. C. Moore. Principal component analysis in linear systems: controllability, observability and model reduction. *IEEE TAC*, AC-26:17–32, 1981.

[5] V. Papakos and I. M. Jaimoukha. Implicitly restarted lanczos algorithm for model reduction. *Proc. 40th IEEE Conf. Decision Control, Orlando, FL*, pages 3671–3672, December 2001.

[6] Y. Saad. *Iterative Methods for Sparse Linear Systems*. PWS Publishing Company, Boston, MA, 1996.

[7] A. Varga. Model reduction software in the SLICOT library. *Proc. of IEEE Int. Symp. Computer Aided Control System Design, CACSD 2000, Anchorage, Alaska*, 2000.

[8] K. Zhou and J. C. Doyle. *Essentials of robust control*. Prentice-Hall, Upper Saddle River, NJ, 1998.