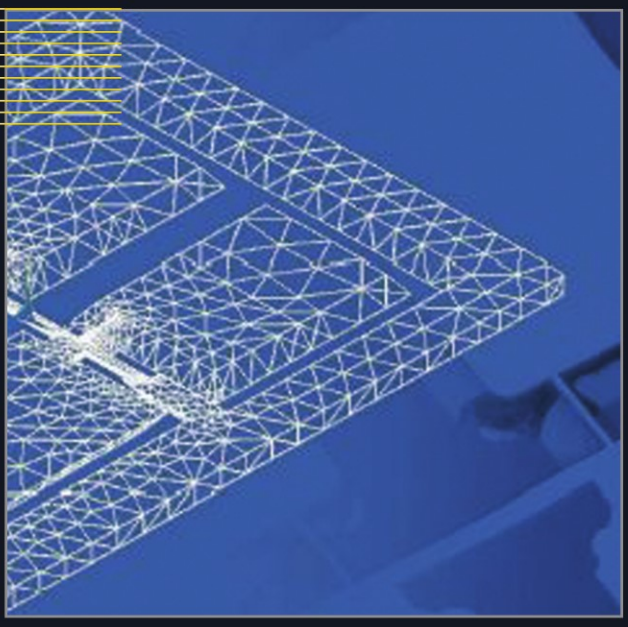
Lecture Notes in Computational Science and Engineering

45



Peter Benner
Volker Mehrmann
Danny C. Sorensen
Editors

Dimension Reduction of Large-Scale Systems

Springer

Lecture Notes
in Computational Science
and Engineering

45

Peter Benner
Volker Mehrmann
Danny C. Sorensen
Editors

Dimension Reduction of Large-Scale Systems

Proceedings of a Workshop held in Oberwolfach, Germany, October 19–25, 2003

With 95 Figures and 29 Tables

Springer

Editors

Peter Benner

Fakultät für Mathematik
Technische Universität Chemnitz
09107 Chemnitz, Germany
email: benner@mathematik.tu-chemnitz.de

Volker Mehrmann

Institut für Mathematik
Technische Universität Berlin
Straße des 17. Juni 136
10623 Berlin, Germany
email: mehrmann@math.tu-berlin.de

Danny C. Sorensen

Department of Computational
and Applied Mathematics
Rice University
Main Street 6100
77005-1892 Houston, TX, USA
email: sorensen@rice.edu

Library of Congress Control Number: 2005926253

Mathematics Subject Classification (2000): 93B11, 93B40, 34-02, 37M05, 65F30, 93C15, 93C20, 76M25

ISSN 1439-7358
ISBN-10 3-540-24545-6 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-24545-2 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media
springeronline.com

Printed in The Netherlands

Cover production: *design & production*, Heidelberg
Typeset by the authors using a Springer TEX macro package
Production: LE-TEX Jelonek, Schmidt & Vöckler GbR, Leipzig

Printed on acid-free paper 46/3142/YL - 5 4 3 2 1 0

Preface

This volume is a result of the mini workshop *Dimension Reduction of Large-Scale Systems* which took place at the Mathematisches Forschungsinstitut Oberwolfach, Germany, October 19–25, 2003. The purpose was to bring together experts from different communities and application areas in an attempt to synthesize major ideas in dimension reduction that have evolved simultaneously but separately in several areas involving simulation and control of complex physical processes. The systems that inevitably arise in such simulations are often too complex to meet the expediency requirements of interactive design, optimization, or real time control. Model order reduction has been devised as a means to reduce the dimensionality of these complex systems to a level that is amenable to such requirements.

Model order reduction seeks to replace a large-scale system of differential or difference equations by a system of substantially lower dimension that has nearly the same response characteristics. Dimension reduction is a common theme within the simulation and control of complex physical processes. Generally, large systems arise due to accuracy requirements on the spatial discretization of control problems for fluids or structures, in the context of lumped-circuit approximations of distributed circuit elements, such as the interconnect or package of VLSI chips. Dimension reduction is generally required for purposes of expediency and/or storage reduction. Applications can be found in

- Simulation of conservative systems, e.g., in Molecular Dynamics,
- Control and regulation of fluid flow (CFD),
- Simulation and stabilization of large structures,
- Control design for (land, air, sea) vehicles,
- VLSI chip design,
- Simulation of micro-electro-mechanical systems (MEMS),
- Semiconductor simulations,
- Image processing,

and many other areas.

Various reduction techniques have been devised, but many of these are described in terms that are discipline-oriented or even application-specific even though they share many common features and origins. This workshop was aimed at bringing together specialists from several fields and application areas in order to expose the similarities of these approaches, to identify common features, to address application-specific challenges, and to investigate how successful reduction methods for linear systems might be applied to nonlinear dynamic systems and very large scale problems with state-space dimensions of order in the millions.

The problems in dimension reduction are challenging from the mathematical and algorithmic points of view. For example, the selection of appropriate basis functions in reduced-order basis approaches like proper orthogonal decomposition (POD) is highly problem-specific and requires a deeper mathematical understanding. On the algorithmic side there is a clear need for additional work in the area of large scale numerical linear algebra. Moreover, it is of considerable interest to introduce some non-traditional techniques such as wavelet bases.

Methods with global computable error bounds are missing in almost all application areas except for medium-size control problems. Here, Gramian-based methods (e.g., balanced truncation) have been successfully applied to approximating the input-output behavior of linear systems and *a posteriori* error bounds can be easily computed. For very large-scale problems or systems based on differential-algebraic equations (DAEs), it is not yet clear how to apply these techniques. For very large scale problems, advanced numerical linear algebra techniques are needed to address the huge matrix dimensions and difficulties resulting, e.g., from irregular sparsity patterns as in circuit simulation. For the special DAE systems arising, e.g., in circuit simulation, methods based on partial realization (moment matching or Padé approximation) have been developed. Though they are successful in some areas, they still lack global error bounds and have difficulties when special system properties such as stability or passivity are to be preserved by the reduced-order model.

During the workshop there were presentations on a variety of theories and methods associated with the above mentioned applications. With this book, we wish to give an overview of the range of topics and to generate interest in

- analyzing the available methods and mathematical theory,
- extracting the best features from different methods,
- developing a deeper mathematical understanding of the methods and application-specific challenges,
- combining good features and new mathematical ideas with the goal of designing superior methods.

A goal of the workshop and this book is to describe some of the most prominent approaches, to discuss common features and point out issues in need of further investigation. We hope to stimulate a broader effort in the area of order reduction for large-scale systems that will lead to new mathematical

and algorithmic tools with the ability to tackle challenging problems in scientific computing ranging from control of nonlinear PDEs to the DC analysis of future generation VLSI chips.

An equally important aspect to this workshop is the collection and distribution of an extensive set of test problems and application specific benchmarks. This should make it much easier to develop relevant methods and to systematically test them.

The participants (in alphabetical order) were Athanasios C. Antoulas (Rice University, Houston, USA), Zhaojun Bai (University of California at Davis, USA), Peter Benner (TU Chemnitz, Germany), Roland W. Freund (Bell Laboratories, Murray Hill, USA), Serkan Gugercin (Virginia Tech, Blacksburg, USA), Michael Hinze (TU Dresden, Germany), Jing-Rebecca Li (INRIA, Rocquencourt, France), Karl Meerbergen (FFT, Leuven, Belgium), Volker Mehrmann (TU Berlin, Germany), Danny C. Sorensen (Rice University, Houston, USA), Tatjana Stykel (TU Berlin, Germany), Paul Van Dooren (Université Catholique de Louvain, Belgium), Andras Varga (DLR Oberpfaffenhofen, Germany), Stefan Volkwein (Universität Graz, Austria), and as a visitor for one day, Jan Korvink (IMETK, University of Freiburg, Germany).

The lively discussions inside this group really inspired this effort to write a collection of articles serving as tutorials to a general audience in the same spirit of the talks as they were presented during the workshop. The decision to provide a set of benchmark examples that should serve as test cases in the development and evaluation of new algorithms for model and dimension reduction was also a product of these discussions. We, the organizers, wish to thank the participants and we hope that the wider research community will find this effort useful.

We would like to thank the Mathematisches Forschungsinstitut Oberwolfach for providing the possibility to organize this Mini-workshop on Dimension Reduction. This opportunity and the fantastic research environment has made this initiative possible.

Chemnitz, Berlin, Houston
February 2005

Peter Benner
Volker L. Mehrmann
Danny C. Sorensen

Contents

Part I

Papers

The first and main part of this book contains ten papers that are written by the participants of the Oberwolfach mini-workshop *Dimension Reduction of Large-Scale Systems*. In most parts, they are kept in a tutorial style in order to allow non-experts to get an overview over some major ideas in current dimension reduction methods. The first 4 papers (Chapters 1–4) discuss various aspects of balancing-related techniques for large-scale systems, structured systems, and descriptor systems. Model reduction techniques for time-varying systems are presented in Chapter 5. The next three papers (Chapters 6–8) treat model reduction for second- and higher-order systems, which can be considered as one of the major research directions in dimension reduction for linear systems. Chapter 9 discusses controller reduction techniques—here, large-scale has a somewhat different meaning than in classical model reduction as controllers are considered as "large" already when the number of states describing the controller's dynamics exceeds 10. The last paper in this part (Chapter 10) concentrates on proper orthogonal decomposition—currently probably the mostly used and most successful model reduction technique for nonlinear systems.

We hope that the surveys on current trends presented here can be used as a starting point for research in dimension reduction methods and stimulates discussions on improving and extending the currently available approaches.

1

Model Reduction Based on Spectral Projection Methods

Peter Benner[1] and Enrique S. Quintana-Ortí[2]

[1] Fakultät für Mathematik, TU Chemnitz, 09107 Chemnitz, Germany; benner@mathematik.tu-chemnitz.de.
[2] Departamento de Ingeniería y Ciencia de Computadores, Universidad Jaume I, 12.071-Castellón, Spain; quintana@icc.uji.es.

Summary. We discuss the efficient implementation of model reduction methods such as modal truncation, balanced truncation, and other balancing-related truncation techniques, employing the idea of spectral projection. Mostly, we will be concerned with the sign function method which serves as the major computational tool of most of the discussed algorithms for computing reduced-order models. Implementations for large-scale problems based on parallelization or formatted arithmetic will also be discussed. This chapter can also serve as a tutorial on Gramian-based model reduction using spectral projection methods.

1.1 Introduction

Consider the linear, time-invariant (LTI) system

$$\begin{array}{ll} \dot{x}(t) = Ax(t) + Bu(t), & t > 0, \quad x(0) = x^0, \\ y(t) = Cx(t) + Du(t), & t \geq 0, \end{array} \tag{1.1}$$

where $A \in \mathbb{R}^{n\times n}$ is the state matrix, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{p\times n}$, $D \in \mathbb{R}^{p\times m}$, and $x^0 \in \mathbb{R}^n$ is the initial state of the system. Here, n is the order (or state-space dimension) of the system. The associated transfer function matrix (TFM) obtained from taking Laplace transforms in (1.1) and assuming $x_0 = 0$ is

$$G(s) = C(sI - A)^{-1}B + D. \tag{1.2}$$

In model reduction we are faced with the problem of finding a reduced-order LTI system,

$$\begin{array}{ll} \dot{\hat{x}}(t) = \hat{A}\hat{x}(t) + \hat{B}\hat{u}(t), & t > 0 \quad \hat{x}(0) = \hat{x}^0, \\ \hat{y}(t) = \hat{C}\hat{x}(t) + \hat{D}\hat{u}(t), & t \geq 0, \end{array} \tag{1.3}$$

of order r, $r \ll n$, and associated TFM $\hat{G}(s) = \hat{C}(sI - \hat{A})^{-1}\hat{B} + \hat{D}$ which approximates $G(s)$. Model reduction of discrete-time LTI systems can be formulated in an analogous manner; see, e.g., [OA01]. Most of the methods and approaches discussed here carry over to the discrete-time setting as well. Here, we will focus our attention on the continuous-time setting, the discrete-time case being discussed in detail in [BQQ03a].

Balancing-related model reduction methods are based on finding an appropriate coordinate system for the state-space in which the chosen Gramian matrices of the system are diagonal and equal. In the simplest case of balanced truncation, the controllability Gramian W_c and the observability Gramian W_o are used. These Gramians are given by the solutions of the two dual *Lyapunov equations*

$$AW_c + W_cA^T + BB^T = 0, \quad A^TW_o + W_oA + C^TC = 0. \tag{1.4}$$

After changing to the coordinate system giving rise to diagonal Gramians with positive decreasing diagonal entries, which are called the Hankel singular values (HSVs) of the system, the reduced-order model is obtained by truncating the states corresponding to the $n - r$ smallest HSVs.

Balanced truncation and its relatives such as singular perturbation approximation, stochastic truncation, etc., are the most popular model reduction techniques used in control theory. The advantages of these methods, guaranteed preservation of several system properties like stability and passivity, as well as the existence of computable error bounds that permit an adaptive selection of the order of the reduced-order model, are unmatched by any other approach. However, thus far, in many other engineering disciplines the use of balanced truncation and other related methods has not been considered feasible due to its computational complexity. Quite often, these disciplines have a preferred model reduction technique as modal analysis and Guyan reduction in structural dynamics, proper orthogonal decomposition (POD) in computational fluid dynamics, Padé and Padé-like approximation techniques based on Krylov subspace methods in circuit simulation and microsystem technology, etc. *A goal of this tutorial is to convince the reader that balanced truncation and its relatives are viable alternatives in many of these areas if efficient algorithms from numerical linear algebra are employed and/or basic level parallel computing facilities are available.*

The ideas presented in this paper are part of an ongoing effort to facilitate the use of balancing-related model reduction methods in large-scale problems arising in the control of partial differential equations, the simulation of VLSI and ULSI circuits, the generation of compact models in microsystems, and other engineering disciplines. This effort mainly involves breaking the $\mathcal{O}(n^2)$ memory and $\mathcal{O}(n^3)$ flops (floating-point arithmetic operations) barriers. Several issues related to this challenge are addressed in this paper. By working with (approximations of) the full-rank factors of the system Gramians rather than using Cholesky factors as in previous balanced truncation algorithms, the complexity of all remaining calculations following the computation of the

factors of the Gramians usually only grows linearly with the dimension of the state-space. This idea is pursued in several approaches that essentially only differ in the way the factors of the Gramians are computed. Approximation methods suitable for sparse systems based mainly on Smith- and ADI-type methods are discussed in Chapters 2 and 3. These allow the computation of the factors at a computational cost and a memory requirement proportional to the number of nonzeros in A. Thus, implementations of balanced truncation based on these ideas are in the same complexity class as Padé-approximation and POD. In this chapter, we focus on the computation of full-rank factors of the Gramians by the sign function method which is based on spectral projection techniques. This does not lead immediately to a reduced overall complexity of the induced balanced truncation algorithm as we deal with general dense systems. However, for special classes of dense problems, a linear-polylogarithmic complexity can be achieved by employing hierarchical matrix structures and the related formatted arithmetic. For the general case, the $\mathcal{O}(n^2)$ memory and $\mathcal{O}(n^3)$ flops complexity remains, but the resulting algorithms are perfectly suited for parallel computations and are highly efficient on current desktops or clusters of workstations. Provided efficient parallel computational kernels for the necessary linear algebra operations are available, balanced truncation can be applied to systems with state-space dimension $n = \mathcal{O}(10^4)$ and dense A-matrix on commodity clusters. By re-using these efficient parallel kernels for computing reduced-order models with a sign function-based implementation of balanced truncation, the application of many other related model reduction methods to large-scale, dense systems becomes feasible. We briefly describe some of the related techniques in this chapter, particularly we discuss sign function-based implementations of the following methods:

- balanced truncation,
- singular perturbation approximation,
- optimal Hankel norm approximation,
- balanced stochastic truncation, and
- truncation methods based on positive real, bounded real, and LQG balancing,

for stable systems. Using a specialized algorithm for the additive decomposition of transfer functions, again based on spectral projection techniques, all the above balancing-related model reduction techniques can also be applied to unstable systems. At this point, we would also like to mention that the same ideas can be applied to balanced truncation for descriptor systems, as described in Chapter 3—for preliminary results see [BQQ04c]—but we will not elaborate on this as this is mostly work in progress.

This paper is organized as follows. In Section 1.2 we provide the necessary background from system and realization theory. Spectral projection, which is the basis for many of the methods described in this chapter, is presented in Section 1.3. Model reduction methods for stable systems of the form (1.1) based on these ideas are described in Section 1.4, where we also in-

clude modal truncation for historical reasons. The basic ideas needed to apply balanced truncation and its relatives to large-scale systems are summarized in Section 1.5. Conclusions and open problems are given in Section 1.6.

Throughout this paper, we will use I_n for the identity matrix in $\mathbb{R}^{n\times n}$ and I for the identity when the order is obvious from the context, $\Lambda(A)$ will denote the spectrum of the matrix A. Usually, capital letters will be used for matrices; lower case letters will stand for vectors with the exception of t denoting time, and i, j, k, m, n, p, r, s employed for integers such as indices and dimensions; Greek letters will be used for other scalars; and calligraphic letters will indicate vector and function spaces. Without further explanation, Π will always denote a permutation matrix of a suitable dimension, usually resulting from row or column pivoting in factorization algorithms. The left and right (open) complex half planes will be denoted by $\mathbb{C}^-$ and $\mathbb{C}^+$, respectively, and we will write $\jmath$ for $\sqrt{-1}$.

1.2 System-Theoretic Background

In this section, we introduce some basic notation and properties of LTI systems used throughout this paper. More detailed introductions to LTI systems can be found in many textbooks [GL95, Son98, ZDG96] or handbooks [Lev96, Mut99]. We essentially follow these references here without further citations, but many other sources can be used for a good overview on the subjects covered in this section.

1.2.1 Linear Systems, Frequency Domain, and Norms

An LTI system is (Lyapunov or exponentially) stable if all its poles are in the left half plane. Sufficient for this is that A is stable (or *Hurwitz*), i.e., the spectrum of A, denoted by $\Lambda(A)$, satisfies $\Lambda(A) \subset \mathbb{C}^-$. It should be noted that the relation between the controllability and observability Gramians of an LTI system and the solutions of the Lyapunov equations in (1.4) only holds if A is stable.

The particular model imposed by (1.1), given by a differential equation describing the behavior of the states x and an algebraic equation describing the outputs y is called a *state-space representation*. Alternatively, the relation between inputs and outputs can also be described in the *frequency domain* by an algebraic expression. Applying the *Laplace transform* to the two equations in (1.1), and denoting the transformed arguments as $x(s)$, $y(s)$, $u(s)$ where s is the Laplace variable, we obtain

$$\begin{aligned} sx(s) - x(0) &= Ax(s) + Bu(s), \\ y(s) &= Cx(s) + Du(s). \end{aligned}$$

By solving for $x(s)$ in the first equation and inserting this into the second equation, we obtain

$$y(s) = \left(C(sI_n - A)^{-1}B + D\right) u(s) + C(sI_n - A)^{-1}x^0.$$

For a zero initial state, the relation between inputs and outputs is therefore completely described by the *transfer function*

$$G(s) := C(sI_n - A)^{-1}B + D. \tag{1.5}$$

Many interesting characteristics of an LTI system are obtained by evaluating $G(s)$ on the positive imaginary axis, that is, setting $s = \jmath\omega$. In this context, ω can be interpreted as the operating frequency of the LTI system.

A stable transfer function defines a mapping

$$G : \mathcal{L}_2 \to \mathcal{L}_2 : u \to y = Gu \tag{1.6}$$

where the two function spaces denoted by $\mathcal{L}_2$ are actually different spaces and should more appropriately be denoted by $\mathcal{L}_2(\mathbb{C}^m)$ and $\mathcal{L}_2(\mathbb{C}^p)$, respectively. As the dimension of the underlying spaces will always be clear from the context, i.e., the dimension of the transfer function matrix $G(s)$ or the dimension of input and output spaces, we allow ourselves the more sloppy notation used in (1.6). The function space $\mathcal{L}_2$ contains the square integrable functions in the frequency domain, obtained via the Laplace transform of the square integrable functions in the time domain, usually denoted as $\mathcal{L}_2(-\infty, \infty)$. The $\mathcal{L}_2$-functions that are analytic in the open right half plane $\mathbb{C}^+$ form the *Hardy space* $\mathcal{H}_2$. Note that $\mathcal{H}_2$ is a closed subspace of $\mathcal{L}_2$. Under the Laplace transform $\mathcal{L}_2$ and $\mathcal{H}_2$ are isometric isomorphic to $\mathcal{L}_2(-\infty, \infty)$ and $\mathcal{L}_2[0, \infty)$, respectively. (This is essentially the *Paley-Wiener Theorem* which is the Laplace transform analog of Parseval's identity for the Fourier transform.) Therefore it is clear that the frequency domain spaces $\mathcal{H}_2$ and $\mathcal{L}_2$ can be endowed with the corresponding norms from their time domain counterparts. Due to this isometry, our notation will not distinguish between norms for the different spaces so that we will denote by $\|f\|_2$ the induced 2-norm on any of the spaces $\mathcal{L}_2(-\infty, \infty)$, $\mathcal{L}_2$, $\mathcal{L}_2[0, \infty)$, and $\mathcal{H}_2$. Using the definition (1.6), it is therefore possible to define an operator norm for G by

$$\|G\| := \sup_{\|u\|_2 \le 1} \|Gu\|_2.$$

It turns out that this operator norm equals the $\mathcal{L}_\infty$-norm of the transfer function G, which for rational transfer functions can be defined as

$$\|G\|_\infty := \sup_{\omega \in \mathbb{R}} \sigma_{\max}(G(\jmath\omega)). \tag{1.7}$$

The $p \times m$-matrix-valued functions G for which $\|G\|_\infty$ is bounded, i.e., those essentially bounded on the imaginary axis, form the function space $\mathcal{L}_\infty$. The subset of $\mathcal{L}_\infty$ containing all $p \times m$-matrix-valued functions that are analytical and bounded in $\mathbb{C}^+$ form the Hardy space $\mathcal{H}_\infty$. As a consequence of the maximum modulus theorem, $\mathcal{H}_\infty$ functions must be bounded on the imaginary

axis so that the essential supremum in (1.7) simplifies to a supremum for rational functions G. Thus, the $\mathcal{H}_\infty$-norm of the rational transfer function $G \in \mathcal{H}_\infty$ can be defined as

$$\|G\|_\infty := \sup_{\omega \in \mathbb{R}} \sigma_{\max}(G(\jmath\omega)). \tag{1.8}$$

A fact that will be of major importance throughout this paper is that the transfer function of a stable LTI system is rational with no poles in the closed right-half plane. Thus, $G \in \mathcal{H}_\infty$ for all stable LTI systems.

Although the notation is somewhat misleading, the $\mathcal{H}_\infty$-norm is the 2-induced operator norm. Hence the sub-multiplicativity condition

$$\|y\|_2 \leq \|G\|_\infty \|u\|_2 \tag{1.9}$$

holds. This inequality implies an important way to tackle the model reduction problem: suppose the original system and the reduced-order model (1.3) are driven by the same input function $u \in \mathcal{H}_2$, so that

$$y(s) = G(s)u(s), \qquad \hat{y}(s) = \hat{G}(s)u(s),$$

where $\hat{G}$ is the transfer function corresponding to (1.3); then we obtain the error bound

$$\|y - \hat{y}\|_2 \leq \|G - \hat{G}\|_\infty \|u\|_2. \tag{1.10}$$

Due to the aforementioned Paley-Wiener theorem, this bound holds in the frequency domain and the time domain. Therefore a goal of model reduction is to compute the reduced-order model so that $\|G - \hat{G}\|_\infty$ is smaller than a given tolerance threshold.

1.2.2 Balanced Realizations

A realization of an LTI system is the set of the four matrices

$$(A, B, C, D) \in \mathbb{R}^{n\times n} \times \mathbb{R}^{n\times m} \times \mathbb{R}^{p\times n} \times \mathbb{R}^{p\times m}$$

corresponding to (1.1). In general, an LTI system has infinitely many realizations as its transfer function is invariant under state-space transformations,

$$\mathcal{T} : \begin{cases} x & \to & Tx, \\ (A, B, C, D) & \to & (TAT^{-1}, TB, CT^{-1}, D), \end{cases} \tag{1.11}$$

as the simple calculation

$$D + (CT^{-1})(sI - TAT^{-1})^{-1}(TB) = C(sI_n - A)^{-1}B + D = G(s)$$

demonstrates. But this is not the only non-uniqueness associated to LTI system representations. Any addition of states that does not influence the input-output relation, meaning that for the same input u the same output y is

achieved, leads to a realization of the same LTI system. Two simple examples are

$$\frac{d}{dt}\begin{bmatrix} x \\ x_1 \end{bmatrix} = \begin{bmatrix} A & 0 \\ 0 & A_1 \end{bmatrix}\begin{bmatrix} x \\ x_1 \end{bmatrix} + \begin{bmatrix} B \\ B_1 \end{bmatrix} u(t), \quad y(t) = \begin{bmatrix} C & 0 \end{bmatrix}\begin{bmatrix} x \\ x_1 \end{bmatrix} + Du(t),$$

$$\frac{d}{dt}\begin{bmatrix} x \\ x_2 \end{bmatrix} = \begin{bmatrix} A & 0 \\ 0 & A_2 \end{bmatrix}\begin{bmatrix} x \\ x_2 \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u(t), \quad y(t) = \begin{bmatrix} C & C_2 \end{bmatrix}\begin{bmatrix} x \\ x_2 \end{bmatrix} + Du(t),$$

for arbitrary matrices $A_j \in \mathbb{R}^{n_j \times n_j}$, $j = 1, 2$, $B_1 \in \mathbb{R}^{n_1 \times m}$, $C_2 \in \mathbb{R}^{p \times n_2}$ and any $n_1, n_2 \in \mathbb{N}$. An easy calculation shows that both of these systems have the same transfer function $G(s)$ as (1.1) so that

$$(A, B, C, D),\ \left(\begin{bmatrix} A & 0 \\ 0 & A_1 \end{bmatrix}, \begin{bmatrix} B \\ B_1 \end{bmatrix}, \begin{bmatrix} C & 0 \end{bmatrix}, D\right),\ \left(\begin{bmatrix} A & 0 \\ 0 & A_2 \end{bmatrix}, \begin{bmatrix} B \\ 0 \end{bmatrix}, \begin{bmatrix} C & C_2 \end{bmatrix}, D\right)$$

are both realizations of the same LTI system described by the transfer function $G(s)$ in (1.5). Therefore, the order n of a system can be arbitrarily enlarged without changing the input-output mapping. On the other hand, for each system there exists a unique minimal number of states which is necessary to describe the input-output behavior completely. This number $\hat{n}$ is called the *McMillan degree* of the system. A *minimal realization* is a realization $(\hat{A}, \hat{B}, \hat{C}, \hat{D})$ of the system with order $\hat{n}$. Note that only the McMillan degree is unique; any state-space transformation (1.11) leads to another minimal realization of the same system. Finding a minimal realization for a given system can be considered as a first step of model reduction as redundant (non-minimal) states are removed from the system. Sometimes this is part of a model reduction procedure, e.g. optimal Hankel norm approximation, and can be achieved via balanced truncation.

Although realizations are highly non-unique, stable LTI systems have a set of invariants with respect to state-space transformations that provide a good motivation for finding reduced-order models. From Lyapunov stability theory (see, e.g., [LT85, Chapter 13]) it is clear that for stable A, the Lyapunov equations in (1.4) have unique positive semidefinite solutions W_c and W_o. These solutions define the *controllability Gramian* (W_c) and *observability Gramian* (W_o) of the system. If W_c is positive definite, then the system is controllable and if W_o is positive definite, the system is observable. Controllability plus observability is equivalent to minimality of the system so that for minimal systems, all eigenvalues of the product $W_c W_o$ are strictly positive real numbers. The square roots of these eigenvalues, denoted in decreasing order by

$$\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n > 0,$$

are known as the *Hankel singular values (HSVs)* of the LTI system and are invariants of the system: let

$$(\hat{A}, \hat{B}, \hat{C}, D) = (TAT^{-1}, TB, CT^{-1}, D)$$

be the transformed realization with associated controllability Lyapunov equation

$$0 = \hat{A}\hat{W}_c + \hat{W}_c\hat{A}^T + \hat{B}\hat{B}^T = TAT^{-1}\hat{W}_c + \hat{W}_cT^{-T}A^TT^T + TBB^TT^T.$$

This is equivalent to

$$0 = A(T^{-1}\hat{W}_cT^{-T}) + (T^{-1}\hat{W}_cT^{-T})A^T + BB^T.$$

The uniqueness of the solution of the Lyapunov equation (see, e.g., [LT85]) implies that $\hat{W}_c = TW_cT^T$ and, analogously, $\hat{W}_o = T^{-T}W_oT^{-1}$. Therefore,

$$\hat{W}_c\hat{W}_o = TW_cW_oT^{-1},$$

showing that $\Lambda(\hat{W}_c\hat{W}_o) = \Lambda(W_cW_o) = \{\sigma_1^2, \ldots, \sigma_n^2\}$. Note that extending the state-space by non-minimal states only adds HSVs of magnitude equal to zero, while the non-zero HSVs remain unchanged.

An important (and name-inducing) type of realizations are *balanced realizations*. A realization (A, B, C, D) is called *balanced* iff

$$W_c = W_o = \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_n \end{bmatrix};$$

that is, the controllability and observability Gramians are diagonal and equal with the decreasing HSVs on their respective diagonal entries. For a minimal realization there always exists a balancing state-space transformation of the form (1.11) with nonsingular matrix $T_b \in \mathbb{R}^{n\times n}$; for non-minimal systems the Gramians can also be transformed into diagonal matrices with the leading $\hat{n} \times \hat{n}$ submatrices equal to $\mathrm{diag}(\sigma_1, \ldots, \sigma_{\hat{n}})$, and

$$\hat{W}_c\hat{W}_o = \mathrm{diag}(\sigma_1^2, \ldots, \sigma_{\hat{n}}^2, 0, \ldots, 0);$$

see, e.g., [TP87]. Using a balanced realization obtained via the transformation matrix T_b, the HSVs allow an energy interpretation of the states; see also [Van00] for a nice treatment of this subject. Specifically, the minimal energy needed to reach x^0 is

$$\inf_{\substack{u\in\mathcal{L}_2(-\infty,0] \\ x(0)=x^0}} \int_{-\infty}^{0} u(t)^Tu(t)\,dt = (x^0)^TW_c^{-1}x^0 = (\hat{x}^0)^T\hat{W}_c^{-1}\hat{x}^0 = \sum_{k=1}^{n}\frac{1}{\sigma_k}\hat{x}_k^2,$$

where $\hat{x}^0 := \begin{bmatrix} \hat{x}_1 \\ \vdots \\ \hat{x}_n \end{bmatrix} = T_bx^0$; hence small HSVs correspond to states that are difficult to reach. The output energy resulting from an initial state x^0 and $u(t) \equiv 0$ for $t > 0$ is given by

$$\|y\|_2^2 = \int_0^\infty y(t)^T y(t)\,dt = x_0^T W_o x_0 = (\hat{x}^0)^T \hat{W}_o \hat{x}^0. = \sum_{k=1}^{n} \sigma_k \hat{x}_j^2;$$

hence large HSVs correspond to the states containing most of the energy in the system. The energy transfer from past inputs to future outputs can be computed via

$$E := \sup_{\substack{u\in\mathcal{L}_2(-\infty,0]\\ x(0)=x^0}} \frac{\|y\|_2^2}{\int_{-\infty}^{0} u(t)^T u(t)\,dt} = \frac{(x^0)^T W_o x^0}{(x^0)^T W_c^{-1} x^0} = \frac{(\bar{x}^0)^T W_c^{\frac{1}{2}} W_o W_c^{\frac{1}{2}} \bar{x}^0}{(\bar{x}^0)^T \bar{x}^0},$$

where $\bar{x}^0 := W_c^{-\frac{1}{2}} x^0$. Thus, the HSVs $(\Lambda(W_c W_o))^{\frac{1}{2}} = \left(\Lambda(W_c^{\frac{1}{2}} W_o W_c^{\frac{1}{2}})\right)^{\frac{1}{2}}$ measure how much the states are involved in the energy transfer from inputs to outputs.

In summary, it seems reasonable to obtain a reduced-order model by removing the least controllable states, keeping the states containing the major part of the system energy as these are the ones which are most involved in the energy transfer from inputs to outputs—that is, keeping the states corresponding to the largest HSVs. This is exactly the idea of balanced truncation, to be outlined in Section 1.4.2.

1.3 Spectral Projection Methods

In this section we will give the necessary background on spectral projection methods and the related computational tools leading to easy-to-implement and easy-to-parallelize iterative methods. These iterative methods will form the backbone of all the model reduction methods discussed in the next section.

1.3.1 Spectral Projectors

First, we give some fundamental definitions and properties of projection matrices.

Definition 1.3.1. *A matrix $P \in \mathbb{R}^{n\times n}$ is a* projector (onto a subspace $\mathcal{S} \subset \mathbb{R}^n$) *if* range $(P) = \mathcal{S}$ *and* $P^2 = P$.

Definition 1.3.2. *Let $Z \in \mathbb{R}^{n\times n}$ with $\Lambda(Z) = \Lambda_1 \cup \Lambda_2$, $\Lambda_1 \cap \Lambda_2 = \emptyset$, and let $\mathcal{S}_1$ be the (right) Z-invariant subspace corresponding to Λ_1. Then a projector onto $\mathcal{S}_1$ is called a* spectral projector.

From this definition we obtain the following properties of spectral projectors.

Lemma 1.3.3. *Let $Z \in \mathbb{R}^{n\times n}$ be as in Definition 1.3.2, and let $P \in \mathbb{R}^{n\times n}$ be a spectral projector onto the right Z-invariant subspace corresponding to Λ_1. Then*

a) $\operatorname{rank}(P) = |\Lambda_1| =: k$,
b) $\operatorname{range}(P) = \operatorname{range}(ZP)$,
c) $\ker(P) = \operatorname{range}(I - P)$, $\operatorname{range}(P) = \ker(I - P)$,
d) $I - P$ *is a spectral projector onto the right* Z*-invariant subspace corresponding to* Λ_2.

Given a spectral projector P we can compute an orthogonal basis for the corresponding Z-invariant subspace $\mathcal{S}_1$ and a *spectral* or block decomposition of Z in the following way: let

$$P = QR\Pi, \quad R = \begin{bmatrix} R_{11} & R_{12} \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} \searrow & \square \\ & \end{bmatrix}, \qquad R_{11} \in \mathbb{R}^{k \times k},$$

be a QR decomposition with column pivoting (or a rank-revealing QR decomposition (RRQR)) [GV96] where Π is a permutation matrix. Then the first k columns of Q form an orthonormal basis for $\mathcal{S}_1$ and we can transform Z to *block-triangular form*

$$\tilde{Z} := Q^T Z Q = \begin{bmatrix} Z_{11} & Z_{12} \\ 0 & Z_{22} \end{bmatrix}, \tag{1.12}$$

where $\Lambda(Z_{11}) = \Lambda_1$, $\Lambda(Z_{22}) = \Lambda_2$.

The block decomposition given in (1.12) will prove very useful in what follows.

1.3.2 The Sign Function Method

Consider a matrix $Z \in \mathbb{R}^{n \times n}$ with no eigenvalues on the imaginary axis, that is, $\Lambda(Z) \cap \jmath\mathbb{R} = \emptyset$, and let $Z = S \begin{bmatrix} J^- & 0 \\ 0 & J^+ \end{bmatrix} S^{-1}$ be its Jordan decomposition. Here, the Jordan blocks in $J^- \in \mathbb{R}^{k \times k}$ and $J^+ \in \mathbb{R}^{(n-k)\times(n-k)}$ contain, respectively, the stable and unstable parts of $\Lambda(Z)$. The *matrix sign function* of Z is defined as $\operatorname{sign}(Z) := S \begin{bmatrix} -I_k & 0 \\ 0 & I_{n-k} \end{bmatrix} S^{-1}$. Note that $\operatorname{sign}(Z)$ is unique and independent of the order of the eigenvalues in the Jordan decomposition of Z, see, e.g., [LR95]. Many other definitions of the sign function can be given; see [KL95] for an overview. Some important properties of the matrix sign function are summarized in the following lemma.

Lemma 1.3.4. *Let* $Z \in \mathbb{R}^{n \times n}$ *with* $\Lambda(Z) \cap \jmath\mathbb{R} = \emptyset$. *Then:*

a) $(\operatorname{sign}(Z))^2 = I_n$, *i.e.,* $\operatorname{sign}(Z)$ *is a square root of the identity matrix;*
b) $\operatorname{sign}\left(T^{-1}ZT\right) = T^{-1}\operatorname{sign}(Z)\,T$ *for all nonsingular* $T \in \mathbb{R}^{n \times n}$;
c) $\operatorname{sign}\left(Z^T\right) = \operatorname{sign}(Z)^T$.

d) Let p_+ and p_- be the numbers of eigenvalues of Z with positive and negative real part, respectively. Then

$$p_+ = \frac{1}{2}(n + \operatorname{tr}(\operatorname{sign}(Z))), \quad p_- = \frac{1}{2}(n - \operatorname{tr}(\operatorname{sign}(Z))).$$

(Here, $\operatorname{tr}(M)$ denotes the trace of the matrix M.)

e) Let Z be stable, then

$$\operatorname{sign}(Z) = -I_n, \quad \operatorname{sign}(-Z) = I_n.$$

Applying Newton's root-finding iteration to $Z^2 = I_n$, where the starting point is chosen as Z, we obtain the Newton iteration for the matrix sign function:

$$Z_0 \leftarrow Z, \quad Z_{j+1} \leftarrow \frac{1}{2}(Z_j + Z_j^{-1}), \quad j = 0, 1, 2, \ldots. \tag{1.13}$$

Under the given assumptions, the sequence $\{Z_j\}_{j=0}^{\infty}$ converges with an ultimately quadratic convergence rate and

$$\operatorname{sign}(Z) = \lim_{j\to\infty} Z_j;$$

see [Rob80]. As the initial convergence may be slow, the use of acceleration techniques is recommended. There are several acceleration schemes proposed in the literature, a thorough discussion can be found in [KL92], and a survey and comparison of different schemes is given in [BD93]. For accelerating (1.13), in each step Z_j is replaced by $\frac{1}{\gamma_j} Z_j$, where the most prominent choices for γ_j are briefly discussed in the sequel.

Determinantal scaling [Bye87]: here,

$$\gamma_j = |\det(Z_j)|^{\frac{1}{n}}.$$

This choice minimizes the distance of the geometric mean of the eigenvalues of Z_j from 1. Note that the determinant $\det(Z_j)$ is a by-product of the computations required to implement (1.13).

Norm scaling [Hig86]: here

$$c_j = \sqrt{\frac{\|Z_j\|_2}{\|Z_j^{-1}\|_2}},$$

which has certain minimization properties in the context of computing polar decompositions. It is also beneficial regarding rounding errors as it equalizes the norms of the two addends in the finite-norm calculation $(\frac{1}{\gamma_j} Z_j) + (\frac{1}{\gamma_j} Z_j)^{-1}$.

Approximate norm scaling: as the spectral norm is expensive to calculate, it is suggested in [Hig86, KL92] to approximate this norm by the Frobenius norm or to use the bound (see, e.g., [GV96])

$$\|Z_j\|_2 \le \sqrt{\|Z_j\|_1 \|Z_j\|_\infty}. \tag{1.14}$$

Numerical experiments and partial analytic considerations [BQQ04d] suggest that norm scaling is to be preferred in the situations most frequently encountered in the sign function-based calculations discussed in the following; see also Example 1.3.6 below. Moreover, the Frobenius norm approximation usually yields a better approximation than the one given by (1.14). As the computation of the Frobenius norm parallelizes very well, we will mostly use the Frobenius norm scaling in the algorithms based on (1.13).

There are also plenty of other iterative schemes for computing the sign function; many of those have good properties regarding convergence and parallelization (see [KL95] for an overview). Nevertheless, the basic Newton iteration (1.13) appears to yield the most robust implementation and the fastest execution times, both in serial and parallel implementations. Implementing (1.13) only requires computing matrix sums and inverses using LU factorization or Gauß-Jordan elimination. These operations are efficiently implemented in many software packages for serial and parallel computations; efficient parallelization of the matrix sign function has been reported, e.g., in [BDD+97, HQOSW00].

Computations based on the matrix sign function can be considered as *spectral projection methods* as they usually involve

$$P_- := \frac{1}{2}(I_n - \operatorname{sign}(Z)), \tag{1.15}$$

which is a spectral projector onto the stable Z-invariant subspace. Also, $P_+ := (I_n + \operatorname{sign}(Z))/2$ is a spectral projector onto the Z-invariant subspace corresponding to the eigenvalues in the open right half plane. But note that P_- and P_+ are not orthogonal projectors, but skew projectors along the complementary Z-invariant subspace.

Remark 1.3.5. The matrix sign function is criticized for several reasons, the most prominent one being the need to compute an explicit inverse in each step. Of course, it is undefined for matrices with purely imaginary eigenvalues and hence suffers from numerical problems in the presence of eigenvalues close to the imaginary axis. But numerical instabilities basically only show up if there exist eigenvalues with imaginary parts of magnitude less than the square root of the machine precision. Hence, significant problems can be expected in double precision arithmetic (as used in MATLAB) for imaginary parts of magnitude less than 10^{-8}. (A thorough numerical analysis requires the condition of the stable subspace which is given by the reciprocal of the separation of stable and anti-stable invariant subspaces, though—the distance of eigenvalues to the imaginary axis is only an upper bound for the separation!) Fortunately, in the control applications considered here, poles are usually further apart from the imaginary axis. On the other hand, if we have no problems with the spectral dichotomy, then the sign function method solves a problem that is usually better conditioned than the Schur vector approach as it only separates the stable from the anti-stable subspace while the Schur vector method essentially

requires to separate n subspaces from each other. For a thorough analysis of sign function-based computation of invariant subspaces, see [BD98, BHM97]. The difference in the conditioning of the Schur form and a block triangular form (as computed by the sign function) is discussed in [KMP01]. Moreover, in the applications considered here, mostly $\mathrm{cond}\,(\mathrm{sign}\,(Z)) = 1$ as Z is stable or anti-stable, hence the computation of $\mathrm{sign}\,(Z)$ itself is a well-conditioned problem!

Therefore, counter to intuition, it should not be surprising that often, results computed by the sign function method are more accurate than those obtained by using Schur-type decompositions; see, e.g., [BQO99].

Example 1.3.6. A typical convergence history (based on $\|Z_j - \mathrm{sign}\,(Z)\,\|_F$) is displayed in Figure 1.1, showing the fast quadratic convergence rate. Here, we computed the sign function of a dense matrix A coming from transforming a generalized state-space system (the $n = 1357$ case of the steel cooling problem described in Chapter 19 of this book) to standard state-space form. We compare the determinantal scaling and the Frobenius norm scaling. Here, the eigenvalue of A closest to $j\mathbb{R}$ is $\approx 6.7 \cdot 10^{-6}$ and the eigenvalue of largest magnitude is ≈ -5.8. Therefore the condition of A is about 10^6. Obviously, norm scaling performs much better for this example. This is a typical behavior for problems with real spectrum. The computations were done using MATLAB 7.0.1 on a Intel Pentium M processor at 1.4 GHz with 512 MBytes of RAM.

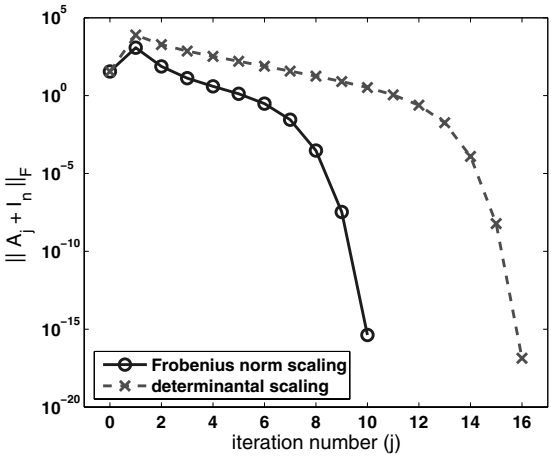


Fig. 1.1. Example 1.3.6, convergence history for $\mathrm{sign}\,(Z)$ using (1.13).

1.3.3 Solving Linear Matrix Equations with the Sign Function Method

In 1971, Roberts [Rob80] introduced the matrix sign function and showed how to solve Sylvester and Lyapunov equations. This was re-discovered several times; see [BD75, DB76, HMW77]. We will briefly review the method for Sylvester equations and will then discuss some improvements useful for model reduction applications.

Consider the Sylvester equation

$$AX + XB + W = 0, \quad A \in \mathbb{R}^{n\times n}, B \in \mathbb{R}^{m\times m}, W \in \mathbb{R}^{n\times m}, \tag{1.16}$$

with $\Lambda(A) \cap \Lambda(-B) = \emptyset$. The latter assumption is equivalent to (1.16) having a unique solution [LT85]. Let $X \in \mathbb{R}^{n\times m}$ be this unique solution. Then the straightforward calculation

$$\begin{bmatrix} I_n & 0 \\ X & I_m \end{bmatrix} \begin{bmatrix} A & 0 \\ W & -B \end{bmatrix} \begin{bmatrix} I_n & 0 \\ -X & I_m \end{bmatrix} = \begin{bmatrix} A & 0 \\ 0 & -B \end{bmatrix} \tag{1.17}$$

reveals that the columns of $\begin{bmatrix} I_n \\ -X_* \end{bmatrix}$ span the invariant subspace of $Z := \begin{bmatrix} A & 0 \\ W & -B \end{bmatrix}$ corresponding to $\Lambda(A)$. In principle, this subspace, and after an appropriate change of basis, also the solution matrix X, can be computed from a spectral projector onto this Z-invariant subspace. The sign function is an appropriate tool for this whenever A, B are stable as in this case, P_- from (1.15) is the required spectral projector. A closer inspection of (1.13) applied to Z shows that we do not even have to form P_- in this case, as the solution can be directly read off the matrix $\operatorname{sign}(Z)$: using (1.17) and Lemma 1.3.4 reveals that

$$\operatorname{sign}(Z) = \operatorname{sign}\left(\begin{bmatrix} A & 0 \\ W & -B \end{bmatrix}\right) = \begin{bmatrix} -I_n & 0 \\ 2X & I_m \end{bmatrix}$$

so that the solution of (1.16) is given as the lower left block of the limit of (1.13), divided by 2. Moreover, the block-triangular structure of Z allows to decouple (1.13) as

$$\begin{aligned} &A_0 \leftarrow A, \quad B_0 \leftarrow B, \quad W_0 \leftarrow W, \\ &\texttt{for } j = 0, 1, 2, \ldots \\ &\quad A_{j+1} \leftarrow \frac{1}{2\gamma_j}\left(A_j + \gamma_j^2 A_j^{-1}\right), \\ &\quad B_{j+1} \leftarrow \frac{1}{2\gamma_j}\left(B_j + \gamma_j^2 B_j^{-1}\right), \\ &\quad W_{j+1} \leftarrow \frac{1}{2\gamma_j}\left(W_j + \gamma_j^2 A_j^{-1} W_j B_j^{-1}\right). \end{aligned} \tag{1.18}$$

so that $X_* = \frac{1}{2}\lim_{j\to\infty} W_j$. As A, B are assumed to be stable, A_j tends to $-I_n$ and B_j tends to $-I_m$ so that we can base a stopping criterion on

$$\max\{\|A_j + I_n\|, \|B_j + I_m\|\} < \tau, \tag{1.19}$$

where τ is an error tolerance and $\|\,.\,\|$ is an appropriate matrix norm.

For Lyapunov equations

$$AX + XA^T + W = 0, \quad A \in \mathbb{R}^{n\times n}, W = W^T \in \mathbb{R}^{n\times n}, \tag{1.20}$$

we simply replace B by A^T in defining Z. Assuming again stability of A, and observing that the iteration for B_j in (1.18) is redundant (see also Lemma 1.3.4 c)), the sign function method for Lyapunov equation becomes

$$\begin{aligned} &A_0 \leftarrow A, \quad W_0 \leftarrow W, \\ &\texttt{for } j = 0, 1, 2, \ldots \\ &\quad A_{j+1} \;\leftarrow\; \frac{1}{2\gamma_j}\left(A_j + \gamma_j^2 A_j^{-1}\right), \\ &\quad W_{j+1} \;\leftarrow\; \frac{1}{2\gamma_j}\left(W_j + \gamma_j^2 A_j^{-1} W_j A_j^{-T}\right). \end{aligned} \tag{1.21}$$

with $X_* = \frac{1}{2}\lim_{j\to\infty} W_j$. Here, a reasonable stopping criterion is given by $\|A_j + I_n\| < \tau$, see (1.19).

If we consider the Lyapunov equations (1.4) defining the controllability and observability Gramians of stable LTI systems, we observe the following facts which will be of importance for an efficient implementation of (1.21) in the context of model reduction:

1. The right-hand side is given in factored form, that is, $W = BB^T$ or $W = C^TC$, and hence semidefinite. Thus, X is positive semidefinite [LT85], and can therefore also be factored as $X = SS^T$. A possibility here is a Cholesky factorization.
2. Usually, the number of states in (1.1) is much larger than the number of inputs and outputs, that is, $n \gg m, p$. In many cases, this yields a solution matrix with rapidly decaying eigenvalues so that its numerical rank is small; see [ASZ02, Gra04, Pen00] for partial explanations of this fact. Figure 1.2 demonstrates this behavior for the controllability Gramian of a random stable LTI system with $n = 500$, $m = 10$, and *stability margin* (minimum distance of $\Lambda(A)$ to $\jmath\mathbb{R}$) ≈ 0.055. Hence, if n_ε is the numerical rank of X, then there is a matrix $S_\varepsilon \in \mathbb{R}^{n\times n_\varepsilon}$ so that $X \approx S_\varepsilon S_\varepsilon^T$ at the level of machine accuracy.

The second observation also serves as the basic idea of most algorithms for large-scale Lyapunov equations; see [Pen00, AS01] as well as Chapters 2 and 3. Storing S_ε is much cheaper than storing X or S as instead of n^2 only $n \cdot n_\varepsilon$ real numbers need to be stored. In the example used above to illustrate the eigenvalue decay, this leads already to a reduction factor of about 10 for storing the solution of the controllability Gramian; in Example 1.3.6 this factor is close to 100 so that 99% of the storage is saved. We will make use of this fact in the method proposed for solving (1.4).

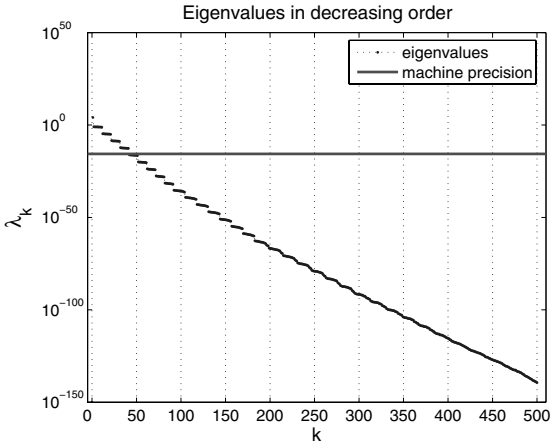


Fig. 1.2. Eigenvalue decay rate for the controllability Gramian of a random LTI system with $n = 500$, $m = 10$, and stability margin ≈ 0.055.

For the derivation of the proposed implementation of the sign function method for computing system Gramians, we will use the Lyapunov equation defining the observability Gramian,

$$A^T Y + Y A + C^T C = 0.$$

Re-writing the iteration for $W_j$in (1.21), we obtain with $W_0 = C_0^T C_0 := C^T C$:

$$W_{j+1} = \frac{1}{2\gamma_j}\left(W_j + \gamma_j^2 A_j^{-T} W_j A_j^{-1}\right) = \frac{1}{2\gamma_j}\begin{bmatrix} C_j \\ \gamma_j C_j A_j^{-1} \end{bmatrix}^T \begin{bmatrix} C_j \\ \gamma_j C_j A_j^{-1} \end{bmatrix}.$$

Thus, in order to compute a factor R of $Y = R^T R$ we can instead directly iterate on the factors:

$$C_0 \leftarrow C, \qquad C_{j+1} \leftarrow \frac{1}{\sqrt{2\gamma_j}}\begin{bmatrix} C_j \\ \gamma_j C_j A_j^{-1} \end{bmatrix}. \tag{1.22}$$

A problem with this iteration is that the number of columns in C_j doubles in each iteration step so that after $j \geq \log_2 \frac{n}{p}$ steps, the required workspace for C_j becomes even larger than n^2. There are several ways to limit this workspace. The first one, initially suggested in [LA93], works with an $n \times n$-matrix, sets C_0 to the Cholesky factor of $C^T C$, computes a QR factorization of $\begin{bmatrix} C_j \\ \gamma_j C_j A_j^{-1} \end{bmatrix}$ in each iteration, and uses its R-factor as next C_j-iterate. A slightly cheaper version of this is given in [BQO99], where (1.22) is used as long as $j \leq \log_2 \frac{n}{p}$ and only then starts computing QR factorizations in each step. In both cases, it can be shown that $\lim_{j\to\infty} C_j$ is a Cholesky factor of the solution Y of (1.20).

In order to exploit the second observation from above, in [BQO99] it is suggested to keep the number of rows in C_j less than or equal to the (numerical) rank of Y by computing in each iteration step a rank-revealing QR factorization

$$\frac{1}{\sqrt{2\gamma_j}} \begin{bmatrix} C_j \\ \gamma_j C_j A_j^{-1} \end{bmatrix} = U_{j+1} \begin{bmatrix} R_{j+1} & T_{j+1} \\ 0 & S_{j+1} \end{bmatrix} \Pi_{j+1}, \tag{1.23}$$

where $R_{j+1} \in \mathbb{R}^{p_{j+1} \times p_{j+1}}$ is nonsingular, $p_{j+1} = \operatorname{rank}\left(\begin{bmatrix} C_j \\ \gamma_j C_j A_j^{-1} \end{bmatrix} \right)$, and $\|S_{j+1}\|_2$ is "small enough" (with respect to a given tolerance threshold for determining the numerical rank) to safely set $S_{j+1} = 0$. Then, the next iterate becomes

$$C_{j+1} \leftarrow [\, R_{j+1} \; T_{j+1} \,] \Pi_{j+1}, \tag{1.24}$$

and $\frac{1}{\sqrt{2}} \lim_{j\to\infty} C_j$ is a (numerical) full-rank factor of the solution Y of (1.20).

The criterion that will be used to select the tolerance threshold for $\|S_{j+1}\|_2$ is based on the following considerations. Let

$$M = \begin{bmatrix} M_1 & M_2 \\ E_1 & E_2 \end{bmatrix}, \quad \tilde{M} = \begin{bmatrix} M_1 & M_2 \end{bmatrix}$$

so that $M^T M$ and $\tilde{M}^T \tilde{M}$ are approximations to a positive semidefinite matrix $K \in \mathbb{R}^{n\times n}$. Assume

$$\|E_j\|_2 \le \sqrt{\varepsilon} \|M\|_2, \quad j = 1, 2,$$

for some $0 < \varepsilon < 1$. Then

$$\begin{aligned} K - M^T M &= K - \begin{bmatrix} M_1^T & E_1^T \\ M_2^T & E_2^T \end{bmatrix} \begin{bmatrix} M_1 & M_2 \\ E_1 & E_2 \end{bmatrix} \\ &= K - \tilde{M}^T \tilde{M} - \begin{bmatrix} E_1^T E_1 & E_1^T E_2 \\ E_2^T E_1 & E_2^T E_2 \end{bmatrix} \end{aligned}$$

If M is a reasonable approximation with $\|M\|_2^2 \approx \|K\|_2$, then the relative error of the two approximations satisfies

$$\frac{\|K - \tilde{M}^T \tilde{M}\|_2}{\|K\|_2} \lessapprox \frac{\|K - M^T M\|_2}{\|K\|_2} + \mathcal{O}(\varepsilon). \tag{1.25}$$

If $\varepsilon \sim \mathbf{u}$, where $\mathbf{u}$ is the machine precision, this shows that neglecting the blocks E_1, E_2 in the factor of the approximation to K yields a relative error of size $\mathcal{O}(\mathbf{u})$ which is negligible in the presence of roundoff errors. Therefore, in our calculations we choose the numerical rank with respect to $\varepsilon = \sqrt{\mathbf{u}}$.

Example 1.3.7. For the same random LTI system as used in the illustration of the eigenvalue decay in Figure 1.2, we computed a numerical full-rank factor of the controllability Gramian. The computed rank is 31, and 10 iterations are needed to achieve convergence in the sign function based iteration. Figure 1.3 shows the development of $p_j = \operatorname{rank}(C_j)$ during the iteration. Comparing (1.24) with the currently best available implementation of Ham-

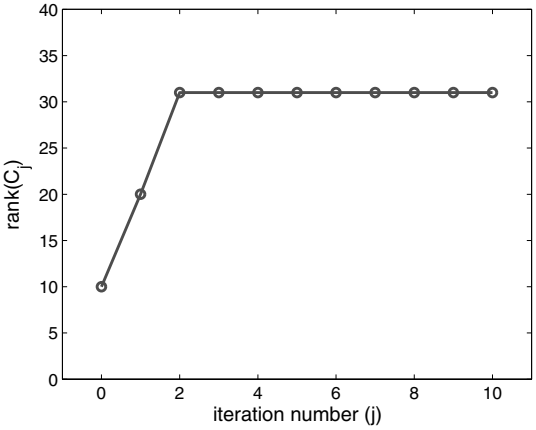


Fig. 1.3. Example 1.3.7, number of columns in C_j in the full-rank iteration composed of (1.22), (1.23), and (1.24).

marling's method [Ham82] for computing the Cholesky factor of the solution of a Lyapunov equation, contained in the SLICOT library [BMS+99], we note that the sign function-based method (pure MATLAB code) required 4.69 sec. while the SLICOT function (compiled and optimized Fortran 77 code, called via a mex file from MATLAB) needed 7.75 sec., both computed using MATLAB 7.0.1 on a Intel Pentium M processor at 1.4 GHz with 512 MBytes of RAM. The computed relative residuals

$$\frac{\|AX + XA^T + BB^T\|_F}{2\|A\|_F\|X\|_F + \|BB^T\|_F}$$

are comparable, $4.6 \cdot 10^{-17}$ for the sign function method and $3.1 \cdot 10^{-17}$ for Hammarling's method.

It is already observed in [LL96] that the two sign function iterations needed to solve both equations in (1.4) can be coupled as they contain essentially the same iteration for the A_j-matrices (the iterates are transposes of each other), hence only one of them is needed. This was generalized and combined with the full-rank iteration (1.24) in [BCQO98, BQQ00a]. The resulting sign function-based "spectral projection method" for computing (numerical) full-rank fac-

Algorithm 1 Coupled Newton Iteration for Dual Lyapunov Equations.

INPUT: Realization $(A, B, C) \in \mathbb{R}^{n\times n} \times \mathbb{R}^{n\times m} \times \mathbb{R}^{p\times n}$ of an LTI system, tolerances τ_1 for convergence of (1.21) and τ_2 for rank detection.

OUTPUT: Numerical full-rank factors of the controllability and observability Gramians of the LTI system such that $W_c = S^T S$, $W_o = R^T R$.

1: **while** $\|A + I_n\|_1 > \tau_1$ **do**

2: Use the LU decomposition or the Gauß-Jordan elimination to compute A^{-1}.

3: Set $\gamma := \sqrt{\frac{\|A\|_F}{\|A^{-1}\|_F}}$ and $Z := \gamma A^{-1}$.

4: Compute a rank-revealing LQ factorization

$$\frac{1}{\sqrt{2\gamma}} \begin{bmatrix} B & ZB \end{bmatrix} =: \Pi \begin{bmatrix} L & 0 \\ T & S \end{bmatrix} Q$$

with $\|S\|_2 \leq \tau_2 \|\frac{1}{\sqrt{2\gamma}} \begin{bmatrix} B & ZB \end{bmatrix} \|_2$.

5: Set $B := \Pi \begin{bmatrix} R \\ T \end{bmatrix}$.

6: Compute a rank-revealing QR factorization

$$\frac{1}{\sqrt{2\gamma}} \begin{bmatrix} C \\ CZ \end{bmatrix} =: Q \begin{bmatrix} R & T \\ 0 & S \end{bmatrix} \Pi$$

with $\|S\|_2 \leq \tau_2 \|\frac{1}{\sqrt{2\gamma}} \begin{bmatrix} C \\ CZ \end{bmatrix} \|_2$.

7: Set $C := \Pi \begin{bmatrix} R & T \end{bmatrix}$.

8: Set $A := \frac{1}{2}(\frac{1}{\gamma} A + Z)$.

9: **end while**

10: Set $S := B^T$, $R := C$.

tors of the controllability and observability Gramians of the LTI system (1.1) is summarized in Algorithm 1.

1.3.4 Block-Diagonalization

In the last section we used the block-diagonalization properties of the sign function method to derive an algorithm for solving linear matrix equations. This feature will also turn out to be useful for other problems such as modal truncation and model reduction of unstable systems. The important equation in this context is (1.17), which allows us to eliminate the off-diagonal block of a block-triangular matrix by solving a Sylvester equation.

A spectral projection method for the block-diagonalization of a matrix Z having no eigenvalues on the imaginary axis is summarized in Algorithm 2. In case of purely imaginary eigenvalues, it can still be used if applied to $Z + \alpha I_n$, where $\alpha \in \mathbb{R}$ is an appropriate spectral shift which is not the real part of an eigenvalue of Z. Note that the computed transformation matrix is not orthogonal, but its first n columns are orthonormal.

Algorithm 2 Sign Function-based Spectral Projection Method for Block-Diagonalization.

INPUT: $Z \in \mathbb{R}^{n\times n}$ with $\Lambda(Z) \cap \jmath\mathbb{R} = \emptyset$.
OUTPUT: $U \in \mathbb{R}^{n\times n}$ nonsingular such that

$$U^{-1}ZU = \begin{bmatrix} Z_{11} & \\ & Z_{22} \end{bmatrix}, \quad \Lambda(Z_{11}) = \Lambda(Z) \cap \mathbb{C}^-, \ \Lambda(Z_{22}) = \Lambda(Z) \cap \mathbb{C}^+.$$

1: Compute $\operatorname{sign}(Z)$ using (1.13).
2: Compute a rank-revealing QR factorization

$$I_n - \operatorname{sign}(Z) =: URH.$$

3: Block-triangularize A as in (1.12); that is, set

$$Z := U^T Z U =: \begin{bmatrix} Z_{11} & Z_{12} \\ 0 & Z_{22} \end{bmatrix}.$$

4: Solve the Sylvester equation $Z_{11}Y - YZ_{22} + Z_{12} = 0$ using (1.18).
{Note: $Z_{11}, -Z_{22}$ are stable!}
5: Set

$$Z := \begin{bmatrix} I_k & -Y \\ 0 & I_{n-k} \end{bmatrix} \begin{bmatrix} Z_{11} & Z_{12} \\ 0 & Z_{22} \end{bmatrix} \begin{bmatrix} I_k & Y \\ 0 & I_{n-k} \end{bmatrix} = \begin{bmatrix} Z_{11} & \\ & Z_{22} \end{bmatrix}, \quad U := U \begin{bmatrix} I_k & Y \\ 0 & I_{n-k} \end{bmatrix}.$$

1.4 Model Reduction Using Spectral Projection Methods

1.4.1 Modal Truncation

Modal truncation is probably one of the oldest model reduction techniques [Dav66, Mar66]. In some engineering disciplines, modified versions are still in use, mainly in structural dynamics. In particular, the model reduction method in [CB68] and its relatives, called nowadays *substructuring methods*, which combine the modal analysis with a static compensation following Guyan [Guy68], are frequently used. We will not elaborate on these type of methods, but will only focus on the basic principles of modal truncation and how it can be implemented using spectral projection ideas.

The basic idea of modal truncation is to project the dynamics of the LTI system (1.1) onto an A-invariant subspace corresponding to the dominant modes of the system (poles of $G(s)$, eigenvalues of A that are not canceled by zeros). In structural dynamics software as ANSYS [ANS] or Nastran [MSC], usually an eigenvector basis of the chosen modal subspace is used. Employing the block-diagonalization abilities of the sign function method described in Subsection 1.3.4, it is easy to derive a spectral projection method for modal truncation. This was first observed by Roberts in his original paper on the

matrix sign function [Rob80]. It has the advantage that we avoid a possible ill-conditioning in the eigenvector basis.

An obvious, though certainly not always optimal, choice of dominant modes is to select those eigenvalues of A having nonnegative or small negative real parts. Basically, these eigenvalues dominate the long-term dynamics of the solution of the linear ordinary differential equation describing the dynamics of (1.1)—solution components corresponding to large negative real parts decay rapidly and mostly play a less important (negligible) role in vibration analysis or control design. This viewpoint is rather naive as it neither takes into account the transient behavior of the dynamical system nor the oscillations caused by large imaginary parts or the sensitivity of the eigenvalues with respect to small perturbations. Nevertheless, this approach is often successful when A comes from an FEM analysis of an elliptic operator such as those arising in linear elasticity or heat transfer processes.

An advantage of modal truncation is that the poles of the reduced-order system are also poles of the original system. This is important in applications such as vibration analysis since the modes correspond to the resonance frequencies of the original system; the most important resonances are thus retained in the reduced-order model.

In the sequel we will use the naive mode selection criterion described above in order to derive a simple implementation of modal truncation employing a spectral projector. The approach, essentially already contained in the original work by Roberts [Rob80], is based on selecting a *stability margin* $\alpha > 0$, which determines the maximum modulus of the real parts of the modes to be preserved in the reduced-order model. Now, the eigenvalues of $A + \alpha I_n$ are the eigenvalues of A, shifted by α to the right. That is, all eigenvalues with stability margin less than α become unstable eigenvalues of $A + \alpha I_n$. Then, applying the sign function to $A + \alpha I_n$ yields the spectral projector $\frac{1}{2}(I_n + \operatorname{sign}(A + \alpha I_n))$ onto the unstable invariant subspace of $A + \alpha I_n$ which equals the A-invariant subspace corresponding to the modes that are dominant with respect to the given stability margin. Block-triangularization of A using (1.12), followed by block-diagonalization based on (1.17) give rise to the modal truncation implementation outlined in Algorithm 3. In principle, Algorithm 2 could also be used here, but the variant in Algorithm 3 is adapted to the needs of modal truncation and slightly cheaper.

The error of modal truncation can easily be quantified. It follows immediately that

$$G(s) - \hat{G}(s) = C_2(sI - A_{22})^{-1}B_2;$$

see also (1.42) below or [GL95, Lemma 9.2.1]. As A_{22}, B_2, C_2 are readily available, the $\mathcal{L}_2$-error for the outputs or $\mathcal{H}_\infty$-error for the transfer function (see (1.10)) is computable. For diagonalizable A_{22}, we obtain the upper bound

$$\|G - \hat{G}\|_\infty \leq \operatorname{cond}_2(T)\, \|C_2\|_2 \|B_2\|_2 \frac{1}{\min_{\lambda \in \Lambda(A_{22})} |\operatorname{Re}(\lambda)|}, \tag{1.26}$$

Algorithm 3 Spectral Projection Method for Modal Truncation.

INPUT: Realization $(A, B, C, D) \in \mathbb{R}^{n\times n} \times \mathbb{R}^{n\times m} \times \mathbb{R}^{p\times n} \times \mathbb{R}^{p\times m}$ of an LTI system (1.1); a stability margin $\alpha > 0$, $\alpha \neq \mathrm{Re}(\lambda)$ for all $\lambda \in \Lambda(A)$.

OUTPUT: Realization $(\hat{A}, \hat{B}, \hat{C}, \hat{D})$ of a reduced-order model.

1: Compute $S := \mathrm{sign}(A + \alpha I_n)$.

2: Compute a rank-revealing QR factorization $S =: QR\Pi$.

3: Compute (see (1.12))

$$Q^T A Q =: \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix}, \quad Q^T B =: \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}, \quad CQ =: \begin{bmatrix} C_1 \\ C_2 \end{bmatrix}.$$

4: Solve the Sylvester equation $(A_{11} - \beta I_k)Y - Y(A_{22} - \beta I_k) + A_{12} = 0$ using (1.18). {Note: If A is stable, $\beta = 0$ can be chosen; otherwise set

$$\beta \geq \max_{\lambda \in \Lambda(A_{11}) \cap \mathbb{C}^+} (\mathrm{Re}(\lambda)),$$

e.g., $\beta = 2\|A_{11}\|_F$.}

5: The reduced-order model is then

$$\hat{A} := A_{11}, \quad \hat{B} := B_1 - Y B_2, \quad \hat{C} := C_1, \quad \hat{D} := D.$$

where $T^{-1} A_{22} T = D$ is the spectral decomposition of A_{22} and $\mathrm{cond}_2(T)$ is the spectral norm condition number of its eigenvector matrix T.

As mentioned at the beginning of this section, several extensions and modifications of modal truncation are possible. In particular, static compensation can account for the steady-state error inherent in the reduced-order model; see, e.g., [Föl94] for an elaborate variant. This is related to singular perturbation approximation; see also subsection 1.4.3 below.

1.4.2 Balanced Truncation

The basic idea of balanced truncation is to compute a balanced realization

$$(TAT^{-1}, TB, CT^{-1}, D) = \left(\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}, \begin{bmatrix} C_1 & C_2 \end{bmatrix}, D \right), \tag{1.27}$$

where $A_{11} \in \mathbb{R}^{r\times r}$, $B_1 \in \mathbb{R}^{r\times m}$, $C_1 \in \mathbb{R}^{p\times r}$, with r less than the McMillan degree $\hat{n}$ of the system, and then to use as the reduced-order model the truncated realization

$$(\hat{A}, \hat{B}, \hat{C}, \hat{D}) = (A_{11}, B_1, C_1, D). \tag{1.28}$$

This idea dates essentially back to [Moo81, MR76]. Collecting results from [Moo81, Glo84, TP87], the following result summarizes the properties of balanced truncation.

Proposition 1.4.1. *Let (A, B, C, D) be a realization of a stable LTI system with McMillan degree $\hat{n}$ and transfer function $G(s)$ and let $(\hat{A}, \hat{B}, \hat{C}, \hat{D})$ with associated transfer function $\hat{G}$ be computed as in (1.27)–(1.28). Then the following holds:*

a) The reduced-order system $\hat{G}$ is balanced, minimal, and stable. Its Gramians are

$$\hat{P} = \hat{Q} = \hat{\Sigma} = \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_r \end{bmatrix}.$$

b) The absolute error bound

$$\|G - \hat{G}\|_\infty \leq 2 \sum_{k=r+1}^{\hat{n}} \sigma_k. \tag{1.29}$$

holds.

c) If $r = \hat{n}$, then (1.28) is a minimal realization of G and $G = \hat{G}$.

Of particular importance is the error bound (1.29) as it allows an adaptive choice of the order of the reduced-order model based on a prescribed tolerance threshold for the approximation quality. (The error bound (1.29) can be improved in the presence of Hankel singular values with multiplicity greater than one—they need to appear only once in the sum on the right-hand side.)

It is easy to check that for a controllable and observable (minimal) system, i.e., a system with nonsingular Gramians, the matrix

$$T = \Sigma^{\frac{1}{2}} U^T R^{-T} \tag{1.30}$$

provides a balancing state-space transformation. Here $W_c = R^T R$ and $R W_o R^T = U \Sigma^2 U^T$ is a singular value decomposition. A nice observation in [LHPW87, TP87] allows us to compute (1.28) also for non-minimal systems without the need to compute the full matrix T. The first part of this observation is that for $W_o = S^T S$,

$$S^{-T}(W_c W_o) S^T = (SR^T)(SR^T)^T = (U \Sigma V^T)(V \Sigma U^T) = U \Sigma^2 U^T$$

so that U, Σ can be computed from an SVD of SR^T,

$$SR^T = \begin{bmatrix} U_1 \; U_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix}, \qquad \Sigma_1 = \operatorname{diag}(\sigma_1, \ldots, \sigma_r). \tag{1.31}$$

The second part needed is the fact that computing

$$T_l = \Sigma_1^{-1/2} V_1^T R, \quad T_r = S^T U_1 \Sigma_1^{-1/2}, \tag{1.32}$$

and

$$\hat{A} := T_l A T_r, \quad \hat{B} := T_l B, \quad \hat{C} := C T_r \tag{1.33}$$

Algorithm 4 Spectral Projection Method for Balanced Truncation.

INPUT: Realization $(A, B, C, D) \in \mathbb{R}^{n\times n} \times \mathbb{R}^{n\times m} \times \mathbb{R}^{p\times n} \times \mathbb{R}^{p\times m}$ of an LTI system (1.1); a tolerance τ for the absolute approximation error or the order r of the reduced-order model.

OUTPUT: Stable reduced-order model, error bound δ.

1: Compute full-rank factors S,R of the system Gramians using Algorithm 1.

2: Compute the SVD

$$SR^T =: \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & \\ & \Sigma_2 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix},$$

such that $\Sigma_1 \in \mathbb{R}^{r\times r}$ is diagonal with the r largest Hankel singular values in decreasing order on its diagonal. Here r is either the fixed order provided on input or chosen as minimal integer such that $2\sum_{j=r+1}^{\hat{n}} \sigma_j \leq \tau$.

3: Set $T_l := \Sigma_1^{-1/2} V_1^T R$, $T_r = S^T U_1 \Sigma_1^{-1/2}$.

4: Compute the reduced-order model,

$$\hat{A} := T_l A T_r, \quad \hat{B} := T_l B, \quad \hat{C} := C T_r, \quad \hat{D} := D,$$

and the error bound $\delta := 2\sum_{j=r+1}^{\hat{n}} \sigma_j$.

is equivalent to first computing a minimal realization of (1.1), then balancing the system as in (1.27) with T as in (1.30), and finally truncating the balanced realization as in (1.28). In particular, the realizations obtained in (1.28) and (1.33) are the same, T_l contains the first r rows of T and T_r the first r columns of T^{-1}—those parts of T needed to compute A_{11}, B_1, C_1 in (1.27). Also note that the product $T_r T_l$ is a projector onto an r-dimensional subspace of the state-space and model reduction via (1.33) can therefore be seen as projecting the dynamics of the system onto this subspace.

The algorithm resulting from (1.33) is often referred to as the *SR method* for balanced truncation. In [LHPW87, TP87] and all textbooks treating balanced truncation, S and R are assumed to be the (square, triangular) Cholesky factors of the system Gramians. In [BQQ00a] it is shown that everything derived so far remains true if full-rank factors of the system Gramians are used instead of Cholesky factors. This yields a much more efficient implementation of balanced truncation whenever $\hat{n} \ll n$ (numerically). Low numerical rank of the Gramians usually signifies a rapid decay of their eigenvalues, as shown in Figure 1.3, and implies a rapid decay of the Hankel singular values. The resulting algorithm, derived in [BQQ00a], is summarized in Algorithm 4.

It is often stated that balanced truncation is not suitable for large-scale problems as it requires the solution of two Lyapunov equations, followed by an SVD, and that both steps require $\mathcal{O}(n^2)$ storage and $\mathcal{O}(n^3)$ flops. This is not true for Algorithm 4 although it does not completely break the $\mathcal{O}(n^2)$ storage and $\mathcal{O}(n^3)$ flops barriers. In Subsection 1.5.2 it will be shown that by reducing the complexity of the first stage of Algorithm 4 down to $\mathcal{O}(n \cdot q(\log n))$, where

q is a quadratic or cubic polynomial, it is possible to break this curse of dimensionality for certain problem classes.

An analysis of Algorithm 4 reveals the following: assume that A is a full matrix with no further structure to be exploited, and define

$$n_{co} := \max\{\operatorname{rank}(S), \operatorname{rank}(R)\} \ll n,$$

where by abuse of notation "rank" denotes the numerical rank of the factors of the Gramians. Then the storage requirements and computational cost are as follows:

1. The solution of the dual Lyapunov equations splits into three separate iterations:
   a) The iteration for A_j requires the inversion of a full matrix and thus needs $\mathcal{O}(n^2)$ storage and $\mathcal{O}(n^3)$ flops.
   b) The iterations for B_j and C_j need an additional $\mathcal{O}(n \cdot n_{co})$ storage, all computations can be performed in $\mathcal{O}(n^2 n_{co})$ flops. The n^2 part in the complexity comes from applying A_j^{-1} using either forward and backward substitution or matrix multiplication—if this can be achieved in a cheaper way as in Subsection 1.5.2, the complexity reduces to $\mathcal{O}(n \cdot n_{co}^2)$.
2. Computing the SVD of SR^T only needs $\mathcal{O}(n_{co}^2)$ workspace and $\mathcal{O}(n \cdot n_{co})$ flops and therefore does not contribute significantly to the cost of the algorithm.
3. The computation of the ROM via (1.32) and (1.33) requires $\mathcal{O}(r^2)$ additional workspace and $\mathcal{O}(nn_{co}r + n^2 r)$ flops where the n^2 part corresponds to the cost of matrix-vector multiplication with A and is not present if this is cheaper than the usual $2n^2$ flops.

An even more detailed analysis shows that the implementation of the SR method of balanced truncation outlined in Algorithm 4 can be significantly faster than the one using Hammarling's method for computing Cholesky factors of the Gramians as used in SLICOT [BMS+99, Var01] and MATLAB; see [BQQ00a]. It is important to remember that if A has a structure that allows to store A in less than $\mathcal{O}(n^2)$, to solve linear systems in less than $\mathcal{O}(n^3)$ and to do matrix-vector multiplication in less than $\mathcal{O}(n^2)$, the complexity of Algorithm 4 is *less than* $\mathcal{O}(n^2)$ *in storage and* $\mathcal{O}(n^3)$ *in computing time!*

If the original system is highly unbalanced (and hence, the state-space transformation matrix T in (1.27) is ill-conditioned), the *balancing-free square-root* (BFSR) balanced truncation algorithm suggested in [Var91] may provide a more accurate reduced-order model in the presence of rounding errors. It combines the SR implementation from [LHPW87, TP87] with the balancing-free model reduction approach in [SC89]. The BFSR algorithm only differs from the SR implementation in the procedure to obtain T_l and T_r from the SVD (1.31) of SR^T, and in that the reduced-order model is not balanced. The main idea is that in order to compute the reduced-order model it is sufficient

to use orthogonal bases for range (T_l) and range (T_r). These can be obtained from the following two QR factorizations:

$$S^T U_1 = [P_1\ P_2] \begin{bmatrix} \hat{R} \\ 0 \end{bmatrix}, \qquad R^T V_1 = [Q_1\ Q_2] \begin{bmatrix} \bar{R} \\ 0 \end{bmatrix}, \tag{1.34}$$

where $P_1, Q_1 \in \mathbb{R}^{n\times r}$ have orthonormal columns, and $\hat{R}, \bar{R} \in \mathbb{R}^{r\times r}$ are upper triangular. The reduced-order system is then given by (1.33) with

$$T_l = (Q_1^T P_1)^{-1} Q_1^T, \qquad T_r = P_1, \tag{1.35}$$

where the $(Q_1^T P_1)^{-1}$ factor is needed to preserve the projector property of $T_r T_l$.

The absolute error of a realization of order r computed by the BFSR implementation of balanced truncation satisfies the same upper bound (1.29) as the reduced-order model computed by the SR version.

Numerical Experiments

We compare modal truncation, implemented as MATLAB function `modaltrunc` following Algorithm 3 and balanced truncation, implemented as MATLAB function `btsr` following Algorithm 4 for some of the benchmark examples presented in Part II of this book. The MATLAB codes are available from

`http://www.tu-chemnitz.de/~benner/software.php`

In the comparison we included several MATLAB implementations of balanced truncation based on using the Bartels-Stewart or Hammarling's method for computing the system Gramians:

- the SLICOT [BMS+99] implementation of balanced truncation, called via a mex-function from the MATLAB function `bta` [Var01],
- the MATLAB Control Toolbox (Version 6.1 (R14SP1)) function `balreal` followed by `modred`,
- the MATLAB Robust Control Toolbox (Version 3.0 (R14SP1)) function `balmr`.

The examples that we chose to compare the methods are:

EX-RAND This is Example 1.3.7 from above.

RAIL1357 This is the steel cooling example described in Chapter 19. Here, we chose the smallest of the provided test sets with $n = 1357$.

FILTER2D This is the optical tunable filter example described in Chapter 15. For the comparison, we chose the 2D problem — the 3D problem is well beyond the scope of the discussed implementations of modal or balanced truncation.

ISS-II This is a model of the extended service module of the International Space Station, for details see Chapter 24.

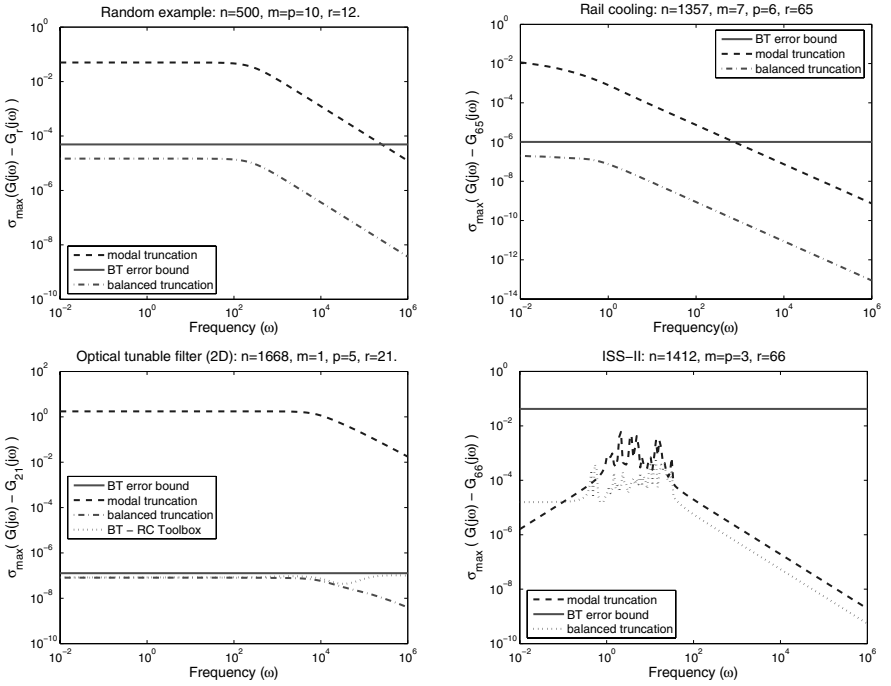


Fig. 1.4. Frequency response (pointwise absolute) error for the Examples EX-RAND, RAIL1357, FILTER2D, ISS-II.

(For a more complete comparison of balanced truncation based on Algorithm 4 and the SLICOT model reduction routines see [BQQ03b].)

The frequency response errors for the chosen examples are shown in Figure 1.4. For the implementations of balanced truncation, we only plotted the error curve for `btsr` as the graphs produced by the other implementations are not distinguishable with the exception of FILTER2D where the Robust Control Toolbox function yields a somewhat bigger error for high frequencies (still satisfying the error bound (1.29)). Note that the frequency response error here is measured as the pointwise absolute error

$$\|G(j\omega) - \hat{G}(j\omega)\|_2 = \sigma_{\max}\left(G(j\omega) - \hat{G}(j\omega)\right),$$

where $\|.\|_2$ is the spectral norm (matrix 2-norm).

From Figure 1.4 it is obvious that for equal order of the reduced-order model, modal truncation usually gives a much worse approximation than balanced truncation. Note that the order r of the reduced-order models was selected based on the reduced-order model computed via Algorithm 3 for a specific, problem-dependent stability margin α. We chose $\alpha = 244$ for EX-RAND, $\alpha = 0.01$ for RAIL1357, $\alpha = 5 \cdot 10^3$ for FILTER2D, and $\alpha = 0.005$ for ISS-II. That is, the reduced-order models computed by balanced truncation

Table 1.1. CPU times needed in the comparison of modal truncation and different balanced truncation implementations for the chosen examples.

Example	Modal Trunc.	Balanced Truncation			
	Alg. 3	Alg. 4	SLICOT	`balreal/modred`	`balmr`
EX-RAND	11.36	5.35	14.24	21.44	34.78
RAIL1357	203.80	101.78	241.44	370.56	633.25
FILTER2D	353.27	152.85	567.46	351.26	953.54
ISS-II	399.65	1402.13	247.21	683.72	421.69

used a fixed order rather than an adaptive selection of the order based on (1.29).

The computation times obtained using MATLAB 7.0.1 on a Intel Pentium M processor at 1.4 GHz with 512 MBytes of RAM are given in Table 1.1.

Some peculiarities we found in the results:

- the error bound (1.29) for EX-RAND as computed by the Robust Control Toolbox function is $2.2 \cdot 10^{-2}$; this compares unfavorably to the correct bound $4.9 \cdot 10^{-5}$, returned correctly by the other implementations of balanced truncation. Similarly, for FILTER2D, the Robust Control Toolbox function computes an error bound $10,000$ times larger than the other routines and the actual error. This suggests that the smaller Hankel singular values computed by `balmr` are very incorrect.
- The behavior for the first 3 examples regarding computing time is very much consistent while the ISS-II example differs significantly. The reason is that the sign function does converge very slowly for this particular example and the full-rank factorization computed reveals a very high numerical rank of the Gramians (roughly $n/2$). This results in fairly expensive QR factorizations at later stages of the iteration in Algorithm 1.

Altogether, spectral projection-based balanced truncation is a viable alternative to other balanced truncation implementations in MATLAB. If the Gramians have low numerical rank, the execution times are generally much smaller than for approaches based on solving the Lyapunov equations (1.4) employing Hammarling's method. On the other hand, Algorithm 4 suffers much from a high numerical rank of the Gramians due to high execution times of Algorithm 1 in that case. The accuracy of all implementations is basically the same for all investigated examples—an observation in accordance to the tests reported in [BQQ00a, BQQ03b]. Moreover, the efficiency of Algorithm 4 allows an easy and highly scalable parallel implementation in contrast to versions based on Hammarling's method, see Subsection 1.5.1. Thus, much larger problems can be tackled using a spectral projection-based approach.

1.4.3 Balancing-Related Methods

Singular Perturbation Approximation

In some situations, a reduced-order model with perfect matching of the transfer function at $s = 0$ is desired. In technical terms, this means that the DC gain is perfectly reproduced. In state-space, this can be interpreted as zero steady-state error. In general this can not be achieved by balanced truncation which performs particularly well at high frequencies ($\omega \to \infty$), with a perfect match at $\omega = \infty$. However, DC gain preservation is achieved by *singular perturbation approximation* (SPA), which proceeds as follows: let $(\tilde{A}, \tilde{B}, \tilde{C}, D)$ denote a minimal realization of the LTI system (1.1), and partition

$$\tilde{A} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \quad \tilde{B} = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}, \quad \tilde{C} = [\, C_1 \;\; C_2],$$

according to the desired size r of the reduced-order model, that is, $A_{11} \in \mathbb{R}^{r \times r}$, $B_1 \in \mathbb{R}^{r \times m}$, and $C_1 \in \mathbb{R}^{p \times r}$. Then the SPA reduced-order model is obtained by the following formulae [LA86]:

$$\begin{array}{ll} \hat{A} := A_{11} + A_{12} A_{22}^{-1} A_{21}, & \hat{B} := B_1 + A_{12} A_{22}^{-1} B_2, \\ \hat{C} := C_1 + C_2 A_{22}^{-1} A_{21}, & \hat{D} := \;\; D + C_2 A_{22}^{-1} B_2. \end{array} \tag{1.36}$$

The resulting reduced-order model satisfies the absolute error bound in (1.29).

When computing the minimal realization with Algorithm 4 or its balancing-free variant, followed by (1.36), we can consider the resulting model reduction algorithm as a spectral projection method for SPA. Further details regarding the parallelization of this implementation of SPA, together with several numerical examples demonstrating its performance, can be found in [BQQ00b].

Cross-Gramian Methods

In some situations, the product $W_c W_o$ of the system Gramians is the square root of the solution of the Sylvester equation

$$AW_{co} + W_{co} A + BC = 0. \tag{1.37}$$

The solution W_{co} of (1.37) is called the *cross-Gramian* of the system (1.1). Of course, for (1.37) to be well-defined, the system must be square, i.e., $p = m$. Then we have $W_{co}^2 = W_c W_o$ if

- the system is symmetric, which is trivially the case if $A = A^T$ and $C = B^T$ (in that case, both equations in (1.4) equal (1.37)) [FN84a];
- the system is a single-input/single-output (SISO) system, i.e., $p = m = 1$ [FN84b].

In both cases, instead of solving (1.4) it is possible to use (1.37). Also note that the cross-Gramian carries information of the LTI system and its internally balanced realization if it is not the product of the controllability and observability Gramian and can still be used for model reduction; see [Ald91, FN84b]. The computation of a reduced-order model from the cross-Gramian is based on computing the dominant W_{co}-invariant subspace which can again be achieved using (1.13) and (1.12) applied to a shifted version of W_{co}.

For $p, m \ll n$, a factorized version of (1.18) can be used to solve (1.37). This again can reduce significantly both the work space needed for saving the cross-Gramian and the computation time in case W_{co} is of low numerical rank; for details see [Ben04]. Also note that the B_j-iterates in (1.18) need not be computed as they equal the A_j's. This further reduces the computational cost of this approach significantly.

Stochastic Truncation

We assume here that $0 < p \leq m$, rank $(D) = p$, which implies that $G(s)$ must not be strictly proper. For strictly proper systems, the method can be applied introducing an ϵ-regularization by adding an artificial matrix $D = [\epsilon I_p \ 0]$ [Glo86].

Balanced stochastic truncation (BST) is a model reduction method based on truncating a balanced stochastic realization. Such a realization is obtained as follows; see [Gre88] for details. Define the *power spectrum* $\Phi(s) = G(s)G^T(-s)$, and let W be a *square minimum phase right spectral factor* of Φ, satisfying $\Phi(s) = W^T(-s)W(s)$. As D has full row rank, $E := DD^T$ is positive definite, and a minimal state-space realization (A_W, B_W, C_W, D_W) of W is given by (see [And67a, And67b])

$$\begin{aligned} A_W &:= A, & B_W &:= BD^T + W_c C^T, \\ C_W &:= E^{-\frac{1}{2}}(C - B_W^T X_W), & D_W &:= E^{\frac{1}{2}}, \end{aligned}$$

where $W_c = S^T S$ is the controllability Gramian defined in (1.4), while X_W is the observability Gramian of $W(s)$ obtained as the stabilizing solution of the *algebraic Riccati equation* (ARE)

$$F^T X + XF + XB_W E^{-1} B_W^T X + C^T E^{-1} C = 0, \tag{1.38}$$

with $F := A - B_W E^{-1} C$. Here, X_W is symmetric positive (semi-)definite and thus admits a decomposition $X_W = R^T R$. If a reduced-order model is computed from an SVD of SR^T as in balanced truncation, then the reduced-order model $(\hat{A}, \hat{B}, \hat{C}, \hat{D})$ is stochastically balanced. That is, the Gramians $\hat{W}_c$, $\hat{X}_W$ of the reduced-order model satisfy

$$\hat{W}_c = \text{diag}\,(\sigma_1, \ldots, \sigma_r) = \hat{X}_W, \tag{1.39}$$

where $1 = \sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > 0$. The BST reduced-order model satisfies the following relative error bound:

$$\sigma_{r+1} \leq \|\Delta_r\|_\infty \leq \prod_{j=r+1}^{n} \frac{1+\sigma_j}{1-\sigma_j} - 1, \tag{1.40}$$

where $G\Delta_r = G - \hat{G}$. From that we obtain

$$\frac{\|G-\hat{G}\|_\infty}{\|G\|_\infty} \leq \prod_{j=r+1}^{n} \frac{1+\sigma_j}{1-\sigma_j} - 1. \tag{1.41}$$

Therefore, BST is also a member of the class of relative error methods which aim at minimizing $\|\Delta_r\|$ for some system norm.

Implementing BST based on spectral projection methods differs in several ways from the versions proposed in [SC88, VF93], though they are mathematically equivalent. Specifically, the Lyapunov equation for W_c is solved using the sign function iteration described in subsection 1.3.3, from which we obtain a full-rank factorization $W_c = S^T S$. The same approach is used to compute a full-rank factor R of X_W from a stabilizing approximation $\tilde{X}_W$ to X_W using the technique described in [Var99]: let $D = \begin{bmatrix} \hat{D}^T & 0 \end{bmatrix} U$ be an LQ decomposition of D. Note that $\hat{D} \in \mathbb{R}^{p\times p}$ is a square, nonsingular matrix as D has full row rank. Now set

$$H_W := \hat{D}^{-T} C, \quad \hat{B}_W := B_W \hat{D}^{-1}, \quad \hat{C} := (H_W - \hat{B}_W^T X).$$

Then the ARE (1.38) is equivalent to $A^T X + XA + \hat{C}^T \hat{C} = 0$. Using a computed approximation $\tilde{X}_W$ of X_W to form $\hat{C}$, the Cholesky or full-rank factor R of X_W can be computed directly from the Lyapunov equation

$$A(R^T R) + (R^T R)A + \hat{C}^T \hat{C} = 0.$$

The approximation $\tilde{X}_W$ is obtained by solving (1.38) using Newton's method with exact line search as described in [Ben97] with the sign function method used for solving the Lyapunov equations in each Newton step; see [BQQ01] for details. The Lyapunov equation for R is solved using the sign function iteration from subsection 1.3.3.

Further Riccati-Based Truncation Methods

There is a variety of other balanced truncation methods for different choices of Gramians to be balanced; see, e.g., [GA03, Obe91]. Important methods are

positive-real balancing: here, passivity is preserved in the reduced-order model which is an important task in circuit simulation;

bounded-real balancing: preserves the $\mathcal{H}_\infty$ gain of the system and is therefore useful for robust control design;

LQG balancing: a closed-loop model reduction technique that preserves closed-loop performance in an LQG design.

In all these methods, the Gramians are solutions of two dual Riccati equations of a similar structure as the stochastic truncation ARE (1.38). The computation of full-rank factors of the system Gramians can proceed in an analogous manner as in BST, and the subsequent computation of the reduced-order system is analogous to the SR or BFSR method for balanced truncation. Therefore, implementations of these model reduction approaches with the computational approaches described so far can also be considered as spectral projection methods. The parallelization of model reduction based on positive-real balancing is described in [BQQ04b]; numerical results demonstrating the accuracy of the reduced-order models and the parallel performance can also be found there.

1.4.4 Unstable Systems

Model reduction for unstable systems can be performed in several ways. One idea is based on the fact that unstable poles are usually important for the dynamics of the system, hence they should be preserved. This can be achieved via an *additive decomposition* of the transfer function as

$$G(s) \;=\; G_-(s) + G_+(s),$$

with $G_-(s)$ stable, $G_+(s)$ unstable, applying balanced truncation to G_- to obtain $\hat{G}_-$, and setting

$$\hat{G}(s) := \hat{G}_-(s) + G_+(s),$$

thereby preserving the unstable part of the system. Such a procedure can be implemented using the spectral projection methods for block-diagonalization and balanced truncation: first, apply Algorithm 2 to A and set

$$\begin{aligned}
\tilde{A} &:= U^{-1}AU = \begin{bmatrix} A_{11} & 0 \\ 0 & A_{22} \end{bmatrix}, \\
\tilde{B} &:= U^{-1}B =: \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}, \quad \tilde{C} := CU =: [\, C_1 \; C_2 \,], \quad \tilde{D} := D.
\end{aligned}$$

This yields the desired additive decomposition as follows:

$$\begin{aligned}
G(s) \;&=\; C(sI - A)^{-1}B + D = \tilde{C}(sI - \tilde{A})^{-1}\tilde{B} + \tilde{D} \\
&=\; \begin{bmatrix} C_1 \; C_2 \end{bmatrix} \begin{bmatrix} (sI_k - A_{11})^{-1} & \\ & (sI_{n-k} - A_{22})^{-1} \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} + D \\
&=\; \big\{C_1(sI_k - A_{11})^{-1}B_1 + D\big\} + \big\{C_2(sI_{n-k} - A_{22})^{-1}B_2\big\} \\
&=: G_-(s) + G_+(s).
\end{aligned} \tag{1.42}$$

Then apply Algorithm 4 to G_- and obtain the reduced order model by adding the transfer functions of the stable reduced and the unstable unreduced parts

as summarized above. This approach is described in more detail in [BCQQ04] where also some numerical examples are given. An extension of this approach using balancing for appropriately defined Gramians of unstable systems is discussed in [ZSW99]. This approach can also be implemented using sign function-based spectral projection techniques similar to the ones used so far.

Alternative model reduction techniques for unstable systems based on coprime factorization of the transfer function and application of balanced truncation to the stable coprime factors are surveyed in [Var01]. Of course, the spectral projection-based balanced truncation algorithm described in Section 1.4.2 could be used for this purpose. The computation of spectral factorizations of transfer functions purely based on spectral projection methods requires further investigation, though.

1.4.5 Optimal Hankel Norm Approximation

BT and SPA model reduction methods aim at minimizing the $\mathcal{H}_\infty$-norm of the error system $G - \hat{G}$. However, they usually do not succeed in finding an optimal approximation; see [AA02]. If a best approximation is desired, a different option is to use the *Hankel norm* of a stable rational transfer function, defined by

$$\|G\|_H := \sigma_1(G), \tag{1.43}$$

where $\sigma_1(G)$ is the largest Hankel singular value of G. Note that $\|G\|_H$ is only a semi-norm on the Hardy space $\mathcal{H}_\infty$ as $\|G\|_H = 0$ does not imply $G \equiv 0$. However, semi-norms are often easier to minimize than norms. In particular, using the Hankel norm it is possible to compute a best order-r approximation to a given transfer function in H_∞. It is shown in [Glo84] that a reduced-order transfer function $\hat{G}$ of order r can be computed that minimizes the Hankel norm of the approximation error in the following sense:

$$\|G - \hat{G}\|_H = \sigma_{r+1} \leq \|G - \tilde{G}\|_H$$

for all stable transfer functions $\tilde{G}$ of McMillan degree less than or equal to r. Moreover, there are explicit formulae to compute such a realization of $\hat{G}$. That is, we can compute a best approximation of the system for a given McMillan degree of the reduced-order model which is usually not possible for other system norms such as the $\mathcal{H}_2$- or $\mathcal{H}_\infty$-norms.

The derivation of a realization of $\hat{G}$ is quite involved, see, e.g., [Glo84, ZDG96]. Here, we only describe the essential computational tools required in an implementation of the HNA method.

The computation of a realization $(\hat{A}, \hat{B}, \hat{C}, \hat{D})$ of the reduced-order model essentially consists of four steps.

In the first step, a balanced minimal realization of G is computed. This can be done using the SR version of the BT method as given in Algorithm 4. Next a transfer function

$$\tilde{G}(s) = \tilde{C}(sI - \tilde{A})^{-1}\tilde{B} + \tilde{D}$$

with the same McMillan degree as the original system (1.1) is computed as follows: first, the order r of the reduced-order model is chosen such that the Hankel singular values of G satisfy

$$\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > \sigma_{r+1} = \ldots = \sigma_{r+k} > \sigma_{r+k+1} \geq \ldots \geq \sigma_{\hat{n}} > 0,\ k \geq 1.$$

Then, by applying appropriate permutations, the minimal balanced realization of G is re-ordered such that the Gramians become

$$\begin{bmatrix} \check{\Sigma} & \\ & \sigma_{r+1} I_k \end{bmatrix}.$$

In a third step, the resulting balanced realization given by $\check{A}, \check{B}, \check{C}, \check{D}$ is partitioned according to the partitioning of the Gramians, that is,

$$\check{A} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \quad \check{B} = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}, \quad \check{C} = [\, C_1 \;\; C_2],$$

where $A_{11} \in \mathbb{R}^{n-k \times n-k}, B_1 \in \mathbb{R}^{n-k \times m}$, $C_1 \in \mathbb{R}^{p \times n-k}$. Then the following formulae define a realization of $\tilde{G}$:

$$\begin{aligned} \tilde{A} &= \Gamma^{-1}(\sigma_{r+1}^2 A_{11}^T + \check{\Sigma} A_{11} \check{\Sigma} + \sigma_{r+1} C_1^T U B_1^T), \\ \tilde{B} &= \Gamma^{-1}(\check{\Sigma} B_1 - \sigma_{r+1} C_1^T U), \\ \tilde{C} &= C_1 \check{\Sigma} - \sigma_{r+1} U B_1^T, \\ \tilde{D} &= D + \sigma_{r+1} U. \end{aligned} \tag{1.44}$$

Here, $U := (C_2^T)^\dagger B_2$, where $M^\dagger$ denotes the pseudoinverse of M, and $\Gamma := \check{\Sigma}^2 - \sigma_{r+1}^2 I_{n-k}$.

Finally, we compute an additive decomposition of $\tilde{G}$ such that $\tilde{G}(s) = \tilde{G}_-(s) + \tilde{G}_+(s)$ where $\tilde{G}_-$ is stable and $\tilde{G}_+$ is anti-stable. For this additive decomposition we use exactly the same algorithm described in the last subsection. Then $\hat{G} := \tilde{G}_-$ is an optimal r-th order Hankel norm approximation of G.

Thus, the main computational tasks of a *spectral projection implementation of optimal Hankel norm approximation* is a combination of Algorithm 4, the formulae (1.44), and Algorithm 2; see [BQQ04a] for further details.

1.5 Application to Large-Scale Systems

1.5.1 Parallelization

Model reduction algorithms based on spectral projection methods are composed of basic matrix computations such as solving linear systems, matrix

products, and QR factorizations. Efficient parallel routines for all these matrix computations are provided in linear algebra libraries for distributed memory computers such as PLAPACK and ScaLAPACK [BCC+97, van97]. The use of these libraries enhances both the reliability and portability of the model reduction routines. The performance will depend on the efficiency of the underlying serial and parallel computational linear algebra libraries and the communication routines.

Here we will employ the ScaLAPACK parallel library [BCC+97]. This is a freely available library that implements parallel versions of many of the kernels in LAPACK [ABB+99], using the message-passing paradigm. ScaLAPACK is based on the PBLAS (a parallel version of the serial BLAS) for computation and BLACS for communication. The BLACS can be ported to any (serial and) parallel architecture with an implementation of the MPI or the PVM libraries [GBD+94, GLS94].

In ScaLAPACK the computations are performed by a logical grid of $n_p = p_r \times p_c$ processes. The processes are mapped onto the physical processors, depending on the available number of these. All data (matrices) have to be distributed among the process grid prior to the invocation of a ScaLAPACK routine. It is the user's responsibility to perform this data distribution. Specifically, in ScaLAPACK the matrices are partitioned into $mb \times nb$ blocks and these blocks are then distributed (and stored) among the processes in column-major order (see [BCC+97] for details).

Using the kernels in ScaLAPACK, we have implemented a library for model reduction of LTI systems, PLiCMR[3], in Fortran 77. The library contains a few driver routines for model reduction and several computational routines for the solution of related equations in control. The functionality and naming convention of the parallel routines closely follow analogous routines from SLICOT. As part of PLiCMR, three parallel driver routines are provided for absolute error model reduction, two parallel driver routines for relative error model reduction, and an expert driver routine capable of performing any of the previous functions on stable and unstable systems. Table 1.2 lists all the driver routines. The driver routines are based on several computational routines included in PLiCMR and listed in Table 1.3. Note that the missing routines in the discrete-time case are available in the Parallel Library in Control (PLiC) [BQQ99], but are not needed in the PLiCMR codes for model reduction of discrete-time systems.

A more detailed introduction to PLiCMR and numerical results showing the model reduction abilities of the implemented methods and their parallel performance can be found in [BQQ03b].

1.5.2 Data-Sparse Implementation of the Sign Function Method

The key to a balanced truncation implementation based on Algorithm 4 with reduced complexity lies in reducing the complexity of storing A and of per-

[3] Available from `http://spine.act.uji.es/~plicmr.html`.

Table 1.2. Driver routines in PLiCMR.

Purpose	Routine	
Expert driver	`pab09mr`	
SR/BFSR BT alg.	`pab09ax`	
SR/BFSR SPA alg.	`pab09bx`	
HNA alg.	`pab09cx`	
SR/BFSR BST alg.	`pab09hx`	
	Continuous-time	Discrete-time
SR/BFSR PRBT alg.	`pab09px`	–

Table 1.3. Computational routines in PLiCMR.

Purpose	Routine	
Solve dual Lyapunov equations and compute HSV	`pab09ah`	
Compute T_l, T_r from SR formulae	`pab09as`	
Compute T_l, T_r from BFSR formulae	`pab09aw`	
Obtain reduced-order model from T_l, T_r	`pab09at`	
Spectral division by sign function	`pmb05rd`	
Factorize TFM into stable/unstable parts	`ptb01kd`	
	Continuous-time	Discrete-time
ARE solver	`pdgecrny`	–
Sylvester solver	`psb04md`	–
Lyapunov solver	`pdgeclnw`	–
Lyapunov solver (for the full-rank factor)	`pdgeclnc`	–
Dual Lyapunov/Stein solver	`psb03odc`	`psb03odd`

forming the required computations with A. Recall that the solution of the Lyapunov equation

$$A^T X + XA + C^T C = 0 \tag{1.45}$$

(or its dual in (1.4)) with the sign function method (1.21) involves the inversion, addition and multiplication of $n \times n$ matrices. Using an approximation of A in $\mathcal{H}$-matrix format [GH03, GHK03] and formatted $\mathcal{H}$-matrix arithmetic, the complexity of storing A and the aforementioned computations reduces to $\mathcal{O}(n \log^2 n)$.

We will briefly describe this approach in the following; for more details and numerical examples see [BB04].

Hierarchical ($\mathcal{H}$-)matrices are a data-sparse approximation of large, dense matrices arising from the discretization of non-local integral operators occurring in the boundary element method or as inverses of FEM discretized elliptic differential operators, but can also be used to represent FEM matrices directly.

Important properties of $\mathcal{H}$-matrices are:

- only few data are needed for the representation of the matrix,
- matrix-vector multiplication can be performed in almost linear complexity ($\mathcal{O}(n \log n)$),

- sums, products, inverses of $\mathcal{H}$-matrices are of "almost" linear complexity.

The basic construction principle of $\mathcal{H}$-matrices can be described as follows: consider matrices over a product index set $\mathcal{I} \times \mathcal{I}$ and partition $\mathcal{I} \times \mathcal{I}$ by an $\mathcal{H}$-*tree* $T_{\mathcal{I}\times\mathcal{I}}$, where a problem dependent *admissibility condition* is used to decide whether a block $t \times s \subset \mathcal{I} \times \mathcal{I}$ allows for a low rank approximation of this block.

Definition 1.5.1. *[GH03] The set of* hierarchical matrices *is defined by*

$$\mathcal{H}(T_{\mathcal{I}\times\mathcal{I}}, k) := \{M \in \mathbb{R}^{\mathcal{I}\times\mathcal{I}} | \operatorname{rank}(M|_{t\times s}) \leq k \text{ for all admissible leaves } t \times s \text{ of } T_{\mathcal{I}\times\mathcal{I}}\}.$$

Submatrices of $M \in \mathcal{H}(T_{\mathcal{I}\times\mathcal{I}}, k)$ corresponding to inadmissible leaves are stored as dense blocks whereas those corresponding to admissible leaves are stored in factorized form as rank-k matrices, called R_k-*format.* Figure 1.5 shows the $\mathcal{H}$-matrix representation with $k = 4$ of the stiffness matrix of the FEM discretization for a 2D heat equation with distributed control and isolation boundary conditions using linear elements on a uniform mesh, resulting in $n = 1024$.

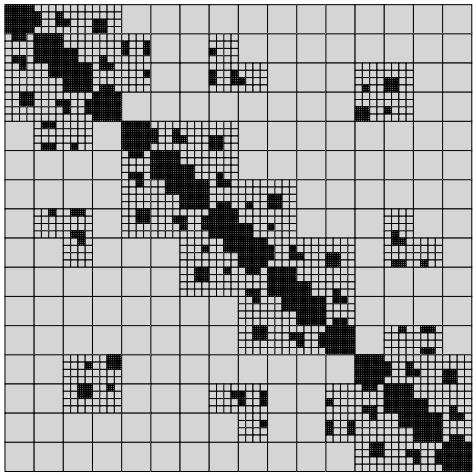


Fig. 1.5. $\mathcal{H}$-matrix representation of stiffness matrix for 2D heat equation with distributed control and isolation boundary conditions. Here $n = 1024$ and $k = 4$.

The formatted arithmetic for $\mathcal{H}$-matrices is not a usual arithmetic as $\mathcal{H}(T_{\mathcal{I}\times\mathcal{I}}, k)$ is not a linear subspace of $\mathbb{R}^{\mathcal{I}\times\mathcal{I}}$, hence sums, products, and inverses of $\mathcal{H}$-matrices need to be projected into $\mathcal{H}(T_{\mathcal{I}\times\mathcal{I}}, k)$. In short, the operations needed here are

Formatted addition ($\oplus$) with complexity $\mathcal{N}_{\mathcal{H}\oplus\mathcal{H}} = \mathcal{O}(nk^2 \log n))$; the computed $\mathcal{H}$-matrix is the best approximation (with respect to the Frobenius-norm) in $\mathcal{H}(T_{\mathcal{I}\times\mathcal{I}}, k)$ of the sum of two $\mathcal{H}$-matrices.
Formatted multiplication ($\odot$) with complexity $\mathcal{N}_{\mathcal{H}\odot\mathcal{H}} = \mathcal{O}(nk^2 \log^2 n)$;
Formatted inversion ($\widetilde{\text{Inv}}$) with complexity $\mathcal{N}_{\mathcal{H},\widetilde{\text{Inv}}} = \mathcal{O}(nk^2 \log^2 n)$.

For the complexity results, some technical assumptions on the $\mathcal{H}$-tree $T_{\mathcal{I}\times\mathcal{I}}$ are needed.

The sign function iteration (1.21) for (1.45) using formatted $\mathcal{H}$-matrix arithmetic with $A_{\mathcal{H}}$ denoting the $\mathcal{H}$-matrix representation in $\mathcal{H}(T_{\mathcal{I}\times\mathcal{I}}, k)$ then becomes

$$\begin{aligned} &A_0 \leftarrow A_{\mathcal{H}}, \quad C_0 \leftarrow C, \\ &\texttt{for } j = 0, 1, 2, \ldots \\ &\quad A_{j+1} \leftarrow \frac{1}{2\gamma_j}\left(A_j \oplus \gamma_j^2 \widetilde{\text{Inv}}(A_j)\right), \\ &\quad \tilde{C}_{j+1} \leftarrow \frac{1}{2\sqrt{\gamma_j}}\left[C_j,\ \gamma_j C_j \odot \widetilde{\text{Inv}}(A_j)\right], \\ &\quad C_{j+1} \leftarrow R\text{-factor of RRQR as in (1.24).} \end{aligned} \tag{1.46}$$

Using this method for solving the Lyapunov equations in the first step of Algorithm 4, we obtain an implementation of balanced truncation requiring only $\mathcal{O}(n_{co}nk \log^2 n)$ storage and $\mathcal{O}(rn_{co}k^2 n \log^2 n)$ flops. Work on this topic is in progress, first numerical results reported in [BB04] are promising that this approach will lend itself to efficient model reduction methods for the control of parabolic partial differential equations.

1.6 Conclusions and Open Problems

Spectral projection methods, in particular those based on the matrix sign function, provide an easy-to-use and easy-to-implement framework for many model reduction techniques. Using the implementations suggested here, balanced truncation and related methods can easily be applied to systems of order $\mathcal{O}(10^3)$ on desktop computers, of order $\mathcal{O}(10^4)$ using parallel programming models, and to more or less unlimited orders if sparse implementations based on matrix compression techniques and formatted arithmetic can be used.

Further investigations could lead to a combination of spectral projection methods based on the sign function with wavelet techniques for the discretization of partial differential equations.

Open problems are the derivation of error bounds for several balancing-related techniques that allow an adaptive choice of the order of the reduced-order model for a given tolerance threshold. This would be particulary important for positive-real balancing as this technique could be very useful in circuit simulation and microsystem technology. The extension of the Riccati-based truncation techniques related to stochastic, positive-real, bounded-real, and

LQG balancing to descriptor systems is another topic for further investigations, both theoretically and computationally.

Acknowledgements

We would like to thank our co-workers Ulrike Baur, Maribel Castillo, José M. Claver, Rafa Mayo, and Gregorio Quintana-Ortí— only through our collaboration all the results discussed in this work could be achieved. We also gratefully acknowledge the helpful remarks and suggestions of an anonymous referee which significantly improved the presentation of this paper.

This work was partially supported by the DFG Sonderforschungsbereich SFB393 "Numerische Simulation auf massiv parallelen Rechnern" at TU Chemnitz, the CICYT project No. TIC2002-004400-C03-01 and project No. P1B-2004-6 of the *Fundación Caixa-Castellón/Bancaixa and UJI.*

References

[AA02] Antoulas, A.C., Astolfi, A.: H_∞-norm approximation. In: Blondel, V.D., Megretski, A. (editors), 2002 MTNS Problem Book, Open Problems on the Mathematical Theory of Networks and Systems, pages 73–76 (2002). Available online from `http://www.nd.edu/~mtns/OPMTNS.pdf`.

[ABB+99] Anderson, E., Bai, Z., Bischof, C., Demmel, J., Dongarra, J., Du Croz, J., Greenbaum, A., Hammarling, S., McKenney, A., Sorensen, D.: LAPACK Users' Guide, SIAM, Philadelphia, PA, third edition (1999).

[Ald91] Aldhaheri, R.W.: Model order reduction via real Schur-form decomposition. Internat. J. Control, **53**:3, 709–716 (1991).

[And67a] Anderson, B.D.O.: An algebraic solution to the spectral factorization problem. IEEE Trans. Automat. Control, **AC-12**, 410–414 (1967).

[And67b] Anderson, B.D.O.: A system theory criterion for positive real matrices. SIAM J. Cont., **5**, 171–182 (1967).

[ANS] ANSYS, Inc., http://www.ansys.com. ANSYS.

[AS01] Antoulas, A.C., Sorensen, D.C.: Approximation of large-scale dynamical systems, An overview. Int. J. Appl. Math. Comp. Sci., **11**:5, 1093–1121 (2001).

[ASZ02] Antoulas, A.C., Sorensen, D.C., Zhou, Y.: On the decay rate of Hankel singular values and related issues. Sys. Control Lett., **46**:5, 323–342 (2002).

[BB04] Baur, U., Benner, P.: Factorized solution of the Lyapunov equation by using the hierarchical matrix arithmetic. Proc. Appl. Math. Mech., **4**:1, 658–659 (2004).

[BCC+97] Blackford, L.S., Choi, J., Cleary, A., D'Azevedo, E., Demmel, J., Dhillon, I., Dongarra, J., Hammarling, S., Henry, G., Petitet, A., Stanley, K., Walker, D., Whaley, R.C.: ScaLAPACK Users' Guide. SIAM, Philadelphia, PA (1997).

[BCQO98] Benner, P., Claver, J.M., Quintana-Ortí, E.S.: Efficient solution of coupled Lyapunov equations via matrix sign function iteration. In: Dourado, A. et al. (editors), Proc. 3rd Portuguese Conf. on Automatic Control CONTROLO'98, Coimbra, pages 205–210 (1998).

[BCQQ04] Benner, P., Castillo, M., Quintana-Ortí, E.S., Quintana-Ortí, G.: Parallel model reduction of large-scale unstable systems. In: Joubert, G.R., Nagel, W.E., Peters, F.J., Walter, W.V. (editors), Parallel Computing: Software Technology, Algorithms, Architectures & Applications. Proc. Intl. Conf. ParCo2003, Dresden, Germany, volume 13 of Advances in Parallel Computing, pages 251–258. Elsevier B.V. (North-Holland) (2004).

[BD75] Beavers, A.N., Denman, E.D.: A new solution method for the Lyapunov matrix equations. SIAM J. Appl. Math., **29**, 416–421 (1975).

[BD93] Bai, Z., Demmel, J.: Design of a parallel nonsymmetric eigenroutine toolbox, Part I. In: R.F. Sincovec et al., editor, Proceedings of the Sixth SIAM Conference on Parallel Processing for Scientific Computing, pages 391–398, SIAM, Philadelphia, PA (1993). *See also:* Tech. Report CSD-92-718, Computer Science Division, University of California, Berkeley, CA 94720.

[BD98] Bai, Z., Demmel, J.: Using the matrix sign function to compute invariant subspaces. SIAM J. Matrix Anal. Appl., **19**:1, 205–225 (1998).

[BDD+97] Bai, Z., Demmel, J., Dongarra, J, Petitet, A., Robinson, H., Stanley, K.: The spectral decomposition of nonsymmetric matrices on distributed memory parallel computers. SIAM J. Sci. Comput., **18**, 1446–1461 (1997).

[Ben97] Benner, P.: Numerical solution of special algebraic Riccati equations via an exact line search method. In: Proc. European Control Conf. ECC 97 (CD-ROM), Paper 786. BELWARE Information Technology, Waterloo, Belgium (1997).

[Ben04] Benner, P.: Factorized solution of Sylvester equations with applications in control. In: Proc. Intl. Symp. Math. Theory Networks and Syst. MTNS 2004, `http://www.mtns2004.be` (2004).

[BHM97] Byers, R., He, C., Mehrmann, V.: The matrix sign function method and the computation of invariant subspaces. SIAM J. Matrix Anal. Appl., **18**:3, 615–632 (1997).

[BMS+99] Benner, P., Mehrmann, V., Sima, V., Van Huffel, S., Varga, A.: SLICOT - a subroutine library in systems and control theory. In: Datta, B.N. (editor), Applied and Computational Control, Signals, and Circuits, volume 1, chapter 10, pages 499–539. Birkhäuser, Boston, MA (1999).

[BQO99] Benner, P., Quintana-Ortí, E.S.: Solving stable generalized Lyapunov equations with the matrix sign function. Numer. Algorithms, **20**:1, 75–100 (1999).

[BQQ99] Benner, P., Quintana-Ortí, E.S., Quintana-Ortí, G.: A portable subroutine library for solving linear control problems on distributed memory computers. In: Cooperman, G., Jessen, E., Michler, G.O. (editors), Workshop on Wide Area Networks and High Performance Computing, Essen (Germany), September 1998, Lecture Notes in Control and Information, pages 61–88. Springer-Verlag, Berlin/Heidelberg, Germany (1999).

[BQQ00a] Benner, P., Quintana-Ortí, E.S., Quintana-Ortí, G.: Balanced truncation model reduction of large-scale dense systems on parallel computers. Math. Comput. Model. Dyn. Syst., **6**:4, 383–405 (2000).

[BQQ00b] Benner, P., Quintana-Ortí, E.S., Quintana-Ortí, G.: Singular perturbation approximation of large, dense linear systems. In: Proc. 2000 IEEE Intl. Symp. CACSD, Anchorage, Alaska, USA, September 25–27, 2000, pages 255–260. IEEE Press, Piscataway, NJ (2000).

[BQQ01] Benner, P., Quintana-Ortí, E.S., Quintana-Ortí, G.: Efficient numerical algorithms for balanced stochastic truncation. Int. J. Appl. Math. Comp. Sci., **11**:5, 1123–1150 (2001).

[BQQ03a] Benner, P., Quintana-Ortí, E.S., Quintana-Ortí, G.: Parallel algorithms for model reduction of discrete-time systems. Int. J. Syst. Sci., **34**:5, 319–333 (2003).

[BQQ03b] Benner, P., Quintana-Ortí, E.S., Quintana-Ortí, G.: State-space truncation methods for parallel model reduction of large-scale systems. Parallel Comput., **29**, 1701–1722 (2003).

[BQQ04a] Benner, P., Quintana-Ortí, E.S., Quintana-Ortí, G.: Computing optimal Hankel norm approximations of large-scale systems. In: Proc. 43rd IEEE Conf. Decision Contr., pages 3078–3083. Omnipress, Madison, WI (2004).

[BQQ04b] Benner, P., Quintana-Ortí, E.S., Quintana-Ortí, G.: Computing passive reduced-order models for circuit simulation. In: Proc. Intl. Conf. Parallel Comp. in Elec. Engrg. PARELEC 2004, pages 146–151. IEEE Computer Society, Los Alamitos, CA (2004).

[BQQ04c] Benner, P., Quintana-Ortí, E.S., Quintana-Ortí, G.: Parallel model reduction of large-scale linear descriptor systems via Balanced Truncation. In: High Performance Computing for Computational Science. Proc. 6th Intl. Meeting VECPAR'04, June 28–30, 2004, Valencia, Spain, pages 65–78 (2004).

[BQQ04d] Benner, P., Quintana-Ortí, E.S., Quintana-Ortí, G.: Solving linear matrix equations via rational iterative schemes. Technical Report SFB393/04-08, Sonderforschungsbereich 393 *Numerische Simulation auf massiv parallelen Rechnern*, TU Chemnitz, 09107 Chemnitz, FRG (2004). Available from `http://www.tu-chemnitz.de/sfb393/preprints.html`.

[Bye87] Byers, R.: Solving the algebraic Riccati equation with the matrix sign function. Linear Algebra Appl., **85**, 267–279 (1987).

[CB68] Craig, R.R., Bampton, M.C.C.: Coupling of substructures for dynamic analysis. AIAA J., **6**, 1313–1319 (1968).

[Dav66] Davison, E.J.: A method for simplifying linear dynamic systems. IEEE Trans. Automat. Control, **AC-11**, 93–101 (1966).

[DB76] Denman, E.D., Beavers, A.N.: The matrix sign function and computations in systems. Appl. Math. Comput., **2**, 63–94 (1976).

[FN84a] Fernando, K.V., Nicholson, H.: On a fundamental property of the cross-Gramian matrix. IEEE Trans. Circuits Syst., **CAS-31**:5, 504–505 (1984).

[FN84b] Fernando, K.V., Nicholson, H.: On the structure of balanced and other principal representations of linear systems. IEEE Trans. Automat. Control, **AC-28**:2, 228–231 (1984).

[Föl94] Föllinger, O.: Regelungstechnik. Hüthig-Verlag, 8. edition (1994).

[GA03] Gugercin, S., Antoulas, A.C.: A survey of balancing methods for model reduction. In: Proc. European Control Conference ECC 2003, Cambridge, UK (2003). CD Rom.

[GBD+94] Geist, A., Beguelin, A., Dongarra, J., Jiang, W., Manchek, B., Sunderam, V.: PVM: Parallel Virtual Machine – A Users Guide and Tutorial for Network Parallel Computing. MIT Press, Cambridge, MA (1994).

[GH03] Grasedyck, L., W. Hackbusch, W.: Construction and arithmetics of $\mathcal{H}$-matrices. Computing, **70**, 295–334 (2003).

[GHK03] Grasedyck, L., Hackbusch, W., Khoromskij, B.N.: Solution of large scale algebraic matrix Riccati equations by use of hierarchical matrices. Computing, **70**, 121–165 (2003).

[GL95] Green, M., Limebeer, D.J.N.: Linear Robust Control. Prentice-Hall, Englewood Cliffs, NJ (1995).

[Glo84] Glover, K.: All optimal Hankel-norm approximations of linear multivariable systems and their L^∞ norms. Internat. J. Control, **39**, 1115–1193 (1984).

[Glo86] Glover, K.: Multiplicative approximation of linear multivariable systems with L_∞ error bounds. In: Proc. American Control Conf., pages 1705–1709 (1986).

[GLS94] Gropp, W., Lusk, E., Skjellum, A.: Using MPI: Portable Parallel Programming with the Message-Passing Interface. MIT Press, Cambridge, MA (1994).

[Gra04] Grasedyck, L.: Existence of a low rank or H-matrix approximant to the solution of a Sylvester equation. Numer. Lin. Alg. Appl., **11**, 371–389 (2004).

[Gre88] Green, M.: Balanced stochastic realization. Linear Algebra Appl., **98**, 211–247 (1988).

[Guy68] Guyan, R.J.: Reduction of stiffness and mass matrices. AIAA J., **3**, 380 (1968).

[GV96] Golub, G.H., Van Loan, C.F.: Matrix Computations. Johns Hopkins University Press, Baltimore, third edition (1996).

[Ham82] Hammarling, S.J.: Numerical solution of the stable, non-negative definite Lyapunov equation. IMA J. Numer. Anal., **2**, 303–323 (1982).

[Hig86] Higham, N.J.: Computing the polar decomposition—with applications. SIAM J. Sci. Statist. Comput., **7**, 1160–1174 (1986).

[HMW77] Hoskins, W.D., Meek, D.S., Walton, D.J.: The numerical solution of $A'Q + QA = -C$. IEEE Trans. Automat. Control, **AC-22**, 882–883 (1977).

[HQOSW00] Huss, S., Quintana-Ortí, E.S., Sun, X., Wu, J.: Parallel spectral division using the matrix sign function for the generalized eigenproblem. Int. J. of High Speed Computing, **11**:1, 1–14 (2000).

[KL92] Kenney, C., Laub, A.J.:. On scaling Newton's method for polar decomposition and the matrix sign function. SIAM J. Matrix Anal. Appl., **13**, 688–706 (1992).

[KL95] Kenney, C., Laub, A.J.:. The matrix sign function. IEEE Trans. Automat. Control, **40**:8, 1330–1348 (1995).

[KMP01] Konstantinov, M.M., Mehrmann, V., Petkov, P.Hr.: Perturbation analysis for the Hamiltonian Schur form. SIAM J. Matrix Anal. Appl., **23**:2, 387–424 (2001).

[LA86] Liu, Y., Anderson, B.D.O. : Controller reduction via stable factorization and balancing. Internat. J. Control, **44**, 507–531 (1986).

[LA93] Larin, V.B., Aliev, F.A.: Construction of square root factor for solution of the Lyapunov matrix equation. Sys. Control Lett., **20**, 109–112 (1993).

[Lev96] Levine, W.S. (editor): The Control Handbook. CRC Press (1996).

[LHPW87] Laub, A.J., Heath, M.T., Paige, C.C., Ward, R.C.: Computation of system balancing transformations and other application of simultaneous diagonalization algorithms. IEEE Trans. Automat. Control, **34**, 115–122 (1987).

[LL96] Lang, W., Lezius, U.: Numerical realization of the balanced reduction of a control problem. In: H. Neunzert, editor, Progress in Industrial Mathematics at ECMI94, pages 504–512, John Wiley & Sons Ltd and B.G. Teubner, New York and Leipzig (1996).

[LR95] Lancaster, P., Rodman, L.: The Algebraic Riccati Equation. Oxford University Press, Oxford (1995).

[LT85] Lancaster, P., Tismenetsky, M.: The Theory of Matrices. Academic Press, Orlando, 2nd edition (1985).

[Mar66] Marschall, S.A.: An approximate method for reducing the order of a linear system. Contr. Eng., **10**, 642–648 (1966).

[Moo81] Moore, B.C.: Principal component analysis in linear systems: Controllability, observability, and model reduction. IEEE Trans. Automat. Control, **AC-26**, 17–32 (1981).

[MR76] Mullis, C., Roberts, R.A.: Synthesis of minimum roundoff noise fixed point digital filters. IEEE Trans. Circuits and Systems, **CAS-23**:9, 551–562 (1976).

[MSC] MSC.Software Corporation, `http://www.mscsoftware.com`. MSC.Nastran.

[Mut99] Mutambara, A.G.O.: Design and Analysis of Control Systems. CRC Press, Boca Raton, FL (1999).

[OA01] Obinata, G., Anderson, B.D.O.: Model Reduction for Control System Design. Communications and Control Engineering Series. Springer-Verlag, London, UK (2001).

[Obe91] Ober, R.: Balanced parametrizations of classes of linear systems. SIAM J. Cont. Optim., **29**, 1251–1287 (1991).

[Pen00] Penzl, T.: A cyclic low rank Smith method for large sparse Lyapunov equations. SIAM J. Sci. Comput., **21**:4, 1401–1418 (2000).

[Rob80] Roberts, J.D.: Linear model reduction and solution of the algebraic Riccati equation by use of the sign function. Internat. J. Control, 32, 677–687 (1980). (Reprint of Technical Report No. TR-13, CUED/B-Control, Cambridge University, Engineering Department (1971).)

[SC88] Safonov, M.G., Chiang, R.Y.: Model reduction for robust control: A Schur relative error method. Int. J. Adapt. Cont. and Sign. Proc., **2**, 259–272 (1988).

[SC89] Safonov, M.G., Chiang, R.Y.: A Schur method for balanced-truncation model reduction. IEEE Trans. Automat. Control, **AC-34**, 729–733 (1989).

[Son98] Sontag, E.D.: Mathematical Control Theory. Springer-Verlag, New York, NY, 2nd edition (1998).

[TP87] Tombs, M.S., I. Postlethwaite, I.: Truncated balanced realization of a stable non-minimal state-space system. Internat. J. Control, **46**:4, 1319–1330 (1987).

[van97] van de Geijn, R.A.: Using PLAPACK: Parallel Linear Algebra Package. MIT Press, Cambridge, MA (1997).

[Van00] Van Dooren,P.: Gramian based model reduction of large-scale dynamical systems. In: D.F. Griffiths G.A. Watson, editors, Numerical Analysis 1999. Proc. 18th Dundee Biennial Conference on Numerical Analysis, pages 231–247, Chapman & Hall/CRC, London, UK (2000).

[Var91] Varga, A.: Efficient minimal realization procedure based on balancing. In: Prepr. of the IMACS Symp. on Modelling and Control of Technological Systems, volume 2, pages 42–47 (1991).

[Var99] Varga, A.: Task II.B.1 – selection of software for controller reduction. SLICOT Working Note 1999–18, The Working Group on Software (WGS) (1999). Available from `http://www.win.tue.nl/niconet/NIC2/reports.html`.

[Var01] Varga, A.: Model reduction software in the SLICOT library. In: B.N. Datta, editor, Applied and Computational Control, Signals, and Circuits, volume 629 of The Kluwer International Series in Engineering and Computer Science, pages 239–282, Kluwer Academic Publishers, Boston, MA (2001).

[VF93] Varga, A., Fasol, K.H.: A new square–root balancing–free stochastic truncation model reduction algorithm. In: Prepr. 12th IFAC World Congress, volume 7, pages 153–156, Sydney, Australia (1993).

[ZDG96] Zhou, K., Doyle, J.C., Glover, K.: Robust and Optimal Control. Prentice-Hall, Upper Saddle River, NJ (1996).

[ZSW99] Zhou, K., Salomon, G., Wu, E.: Balanced realization and model reduction for unstable systems. Int. J. Robust Nonlinear Control, **9**:3, 183–198, (1999).

2

Smith-Type Methods for Balanced Truncation of Large Sparse Systems

Serkan Gugercin[1] and Jing-Rebecca Li[2]

[1] Virginia Tech., Dept. of Mathematics, Blacksburg, VA, 24061-0123, USA
gugercin@math.vt.edu
[2] INRIA-Rocquencourt, Projet Ondes, Domaine de Voluceau - Rocquencourt - B.P. 105, 78153 Le Chesnay Cedex, France jingrebecca.li@inria.fr

2.1 Introduction

Many physical phenomena, such as heat transfer through various media, signal propagation through electric circuits, vibration suppression of bridges, the behavior of Micro-Electro-Mechanical Systems (MEMS), and flexible beams are modelled with linear time invariant (LTI) systems

$$\mathbf{\Sigma}: \begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{cases} \quad \Leftrightarrow \quad \mathbf{\Sigma} := \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$$

where $x(t) \in \mathbb{R}^n$ is the state, $u(t) \in \mathbb{R}^m$ is the input and $y(t) \in \mathbb{R}^p$ is the output; moreover $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{p\times n}$, $D \in \mathbb{R}^{p\times m}$ are constant matrices. The number of the states, n, is called the *dimension* or *order* of the system $\mathbf{\Sigma}$. Closely related to this system are two continuous-time *Lyapunov equations*:

$$A\mathcal{P} + \mathcal{P}A^T + BB^T = 0, \quad A^T\mathcal{Q} + \mathcal{Q}A + C^TC = 0. \tag{2.1}$$

The matrices $\mathcal{P} \in \mathbb{R}^{n\times n}$ and $\mathcal{Q} \in \mathbb{R}^{n\times n}$ are called the *reachability* and *observability Gramians*, respectively. Under the assumptions that A is asymptotically stable, i.e. $\lambda_i(A) \in \mathbb{C}_-$ (the open left half-plane), and that $\mathbf{\Sigma}$ is minimal (that is the pairs (A, B) and (C, A) are, respectively, reachable and observable), the Gramians $\mathcal{P}$, $\mathcal{Q}$ are unique and positive definite. In many applications, such as circuit simulation or time dependent PDE control problems, the dimension, n, of $\mathbf{\Sigma}$ is quite large, in the order of tens of thousands or higher, while the number of inputs m and outputs p usually satisfy $m, p \ll n$. In these large-scale settings, it is often desirable to approximate the given system with a much lower dimensional system

$$\mathbf{\Sigma}_r: \begin{cases} \dot{x}_r(t) = A_r x(t) + B_r u(t) \\ y_r(t) = C_r x(t) + D_r u(t) \end{cases} \quad \Leftrightarrow \quad \mathbf{\Sigma}_r := \left[\begin{array}{c|c} A_r & B_r \\ \hline C_r & D_r \end{array}\right]$$

where $A_r \in \mathbb{R}^{r\times r}$, $B_r \in \mathbb{R}^{r\times m}$, $C_r \in \mathbb{R}^{p\times r}$, $D_r \in \mathbb{R}^{p\times m}$, with $r \ll n$. The problem of model reduction is to produce such a low dimensional system $\mathbf{\Sigma}_r$ that has similar response characteristic as the original system $\mathbf{\Sigma}$ to any given input u.

The Lyapunov matrix equations in (2.1) play an important role in model reduction. One of the most effective model reduction approaches, called *balanced truncation* [MOO81, MR76], requires solving (2.1) to obtain $\mathcal{P}$ and $\mathcal{Q}$. A state space transformation based on $\mathcal{P}$ and $\mathcal{Q}$ is then derived to balance the system in the sense that the two Gramians become diagonal and equal. In this new co-ordinate system, states that are difficult to reach are simultaneously difficult to observe. Then, the reduced model is obtained by truncating the states that are both difficult to reach and difficult to observe. When applied to stable systems, balanced truncation preserves stability and provides an *a priori* bound on the approximation error.

For small-to-medium scale problems, balanced truncation can be implemented efficiently using the Bartels-Stewart [BS72] method, as modified by Hammarling [HAM82], to solve the two Lyapunov equations in (2.1). However, the method requires computing a Schur decomposition and results in $\mathcal{O}(n^3)$ arithmetic operations and $\mathcal{O}(n^2)$ storage; therefore, it is not appropriate for large-scale problems.

For large-scale sparse problems, iterative methods are preferred since they retain the sparsity of the problem and are much more suitable for parallelization. The Smith method [SMI68], the alternating direction implicit (**ADI**) iteration method [WAC88a], and the Smith(l) method [PEN00b] are the most popular iterative schemes developed for large sparse Lyapunov equations. Unfortunately, even though the number of arithmetic operations is reduced, all of these methods compute the solution in dense form and hence require $\mathcal{O}(n^2)$ storage.

It is well known that the Gramians $\mathcal{P}$ and $\mathcal{Q}$ often have low numerical rank (i.e. the eigenvalues of $\mathcal{P}$ and $\mathcal{Q}$ decay rapidly). This phenomenon is explained to a large extent in [ASZ02, PEN00a]. One must take advantage of this low-rank structure to obtain approximate solutions in low-rank factored form. In other words, one should construct a matrix $Z \in \mathbb{R}^{n\times r}$ such that $\mathcal{P} \approx ZZ^T$. The matrix Z is called *the approximate low-rank Cholesky factor* of $\mathcal{P}$. If the effective rank r is much smaller than n, i.e. $r \ll n$, then the storage is reduced from $\mathcal{O}(n^2)$ to $\mathcal{O}(nr)$. We note that such low-rank schemes are the only existing methods that can effectively solve very large sparse Lyapunov equations.

Most low-rank methods, such as [HPT96, HR92, JK94, SAA90], are Krylov subspace methods. As stated in [PEN00b], even though these methods reduce the memory requirement, they usually fail to yield approximate solutions of high accuracy. To reach accurate approximate solutions, one usually needs a large number of iterations, and therefore obtain approximations with relatively high numerical ranks; see [PEN00b]. For large-scale sparse Lyapunov equations, a more efficient low-rank scheme based on the ADI iteration was

introduced, independently, by Penzl [PEN00b], and Li and White [LW02]. The method was called the low-rank ADI iteration (**LR-ADI**) in [PEN00b] and the Cholesky factor ADI iteration (**CF-ADI**) in [LW02]. Even though LR-ADI and CF-ADI are theoretically the same, CF-ADI is less expensive and more efficient to implement. Indeed, LR-ADI can be considered as an intermediate step in deriving the CF-ADI algorithm. Another low-rank scheme based on the ADI iteration was also introduced in [PEN00b]. The method is called the cyclic low-rank Smith method (**LR-Smith(l)**) and is a special case of LR-ADI where l number of shifts are re-used in a cyclic manner.

While solving the Lyapunov equation $A\mathcal{P} + \mathcal{P}A^T + BB^T = 0$ where B has m columns, the LR-ADI and the LR-Smith(l) methods add m and $m \times l$ columns respectively to the current solution at each step, where l is the number of shifts. Therefore, for slowly converging iterations and for the case where m is big, e.g. $m = 10$, the number of columns of the approximate low-rank Cholesky factor can exceed manageable memory capacity. To overcome this, Gugercin *et. al.* [GSA03] introduced a Modified LR-Smith(l) method that prevents the number of columns from increasing arbitrarily at each step. In fact, the method only requires the number of columns r which are needed to meet the pre-specified balanced truncation tolerance. Due to the rapid decay of the Hankel singular values, this r is usually quite small relative to n. Consequently the memory requirements are drastically reduced.

This paper surveys Smith-type methods used for solving large-scale sparse Lyapunov equations and consequently for balanced truncation of the underlying large sparse dynamical system. Connections between different Smith-type methods, convergence results, and upper bounds for the approximation errors are presented. Moreover, numerical examples are given to illustrate the performance of these algorithms.

2.2 Balancing and Balanced Truncation

One model reduction scheme that is well grounded in theory is *Balanced Truncation*, first introduced by Mullis and Roberts [MR76] and later in the systems and control literature by Moore [MOO81]. The approximation theory underlying this approach was developed by Glover [GLO84]. Several researchers have recognized the importance of balanced truncation for model reduction because of its theoretical properties. Computational schemes for small-to-medium scale problems already exist. However, the development of computational methods for large-scale settings is still an active area of research; see [GSA03, PEN99, BQQ01, AS02], and the references therein.

2.2.1 The Concept of Balancing

Let $\mathcal{P}$ and $\mathcal{Q}$ be the unique Hermitian positive definite solutions to equations (2.1). The square roots of the eigenvalues of the product $\mathcal{PQ}$ are the

singular values of the Hankel operator associated with $\mathbf{\Sigma}$ and are called the *Hankel singular values*, $\sigma_i(\mathbf{\Sigma})$, of the system $\mathbf{\Sigma}$:

$$\sigma_i(\mathbf{\Sigma}) = \sqrt{\lambda_i(\mathcal{PQ})}.$$

In most cases, the eigenvalues of $\mathcal{P}, \mathcal{Q}$ as well as the Hankel singular values $\sigma_i(\mathbf{\Sigma})$ decay very rapidly. This phenomena is explained to a large extent in [ASZ02].

Define the two functionals J_r and J_o as follows:

$$J_r = \min_{x(-\infty)=0,\ x(0)=x} \|u(t)\|^2, \quad t \leq 0, \tag{2.2}$$

$$J_o = \|y(t)\|^2, \quad x(0) = x_o,\ u(t) = 0,\ t \geq 0. \tag{2.3}$$

The quantity J_r is the minimal energy required to drive the system from the zero state at $t = -\infty$ to the state x at $t = 0$. On the other hand, J_o is the energy obtained by observing the output with the initial state x_o under no input. The following lemma is crucial to the concept of balancing:

Lemma 2.2.1. *Let $\mathcal{P}$ and $\mathcal{Q}$ be the reachability and observability Gramians of the asymptotically stable and minimal system $\mathbf{\Sigma}$ and J_r and J_o be defined as above. Then*

$$J_r = x^T \mathcal{P}^{-1} x$$

and

$$J_o = x_o^T \mathcal{Q}\, x_o.$$

It follows from the above lemma that the states which are difficult to reach, i.e., require a large energy J_r, are spanned by the eigenvectors of $\mathcal{P}$ corresponding to small eigenvalues. Moreover, the states which are difficult to observe, i.e., yield small observation energy J_o, are spanned by the eigenvectors of $\mathcal{Q}$ corresponding to small eigenvalues. Hence Lemma 2.2.1 yields a way to evaluate the degree of reachability and the degree of observability for the states of the given system. One can obtain a reduced model by eliminating the states which are difficult to reach and observe. However, it is possible that the states which are difficult to reach are not difficult to observe and vice-versa. See [ANT05] for more details and examples. Hence the following question arises: Given $\mathbf{\Sigma}$, does there exist a basis where the states which are difficult to reach are simultaneously difficult to observe? It is easy to see from the Lyapunov equations in (2.1) that under a state transformation by a nonsingular matrix T, the Gramians are transformed as

$$\bar{\mathcal{P}} = T\mathcal{P}T^T, \qquad \bar{\mathcal{Q}} = T^{-T}\mathcal{Q}T^{-1}.$$

Hence, the answer to the above question reduces to finding a nonsingular state transformation T such that, in the transformed basis, the Gramians $\bar{\mathcal{P}}$ and $\bar{\mathcal{Q}}$ are equal.

Definition 2.2.2. *The reachable, observable and stable system* $\mathbf{\Sigma}$ *is called balanced if* $\mathcal{P} = \mathcal{Q}$. $\mathbf{\Sigma}$ *is called principal-axis-balanced if*

$$\mathcal{P} = \mathcal{Q} = \Sigma = \operatorname{diag}(\sigma_1 I_{m_1}, \cdots, \sigma_q I_{m_q}), \tag{2.4}$$

where $\sigma_1 > \sigma_2 > \cdots > \sigma_q > 0$, m_i, $i = 1, \cdots, q$, *are the multiplicities of* σ_i, *and* $m_1 + \cdots + m_q = n$.

In the following, by balancing we mean **principal-axis-balancing** unless otherwise stated. It follows from the above definition that balancing amounts to the simultaneous diagonalization of the two positive definite matrices $\mathcal{P}$ and $\mathcal{Q}$.

Let U denote the Cholesky factor of $\mathcal{P}$, i.e., $\mathcal{P} = UU^T$, and let $U^T \mathcal{Q} U = R\Sigma^2 R^T$ be the eigenvalue decomposition of $U^T \mathcal{Q} U$. The following result explains how to compute the balancing transformation T:

Lemma 2.2.3. Principal-Axis-Balancing Transformation:
Given the minimal and asymptotically stable LTI system $\mathbf{\Sigma}$ *with the corresponding Gramians* $\mathcal{P}$ *and* $\mathcal{Q}$, *a principal-axis-balancing transformation* T *is*

$$T = \Sigma^{1/2} R^T U^{-1}. \tag{2.5}$$

The next result gives a generalization of all possible balancing transformations:

Corollary 2.2.4. *Let there be* q *distinct Hankel singular values* σ_i *with multiplicities* m_i. *Every principal-axis-balancing transformation* $\hat{T}$ *has the form* $\hat{T} = VT$ *where* T *is given by (2.5) and* V *is a block diagonal unitary matrix with an arbitrary* $m_i \times m_i$ *unitary matrix as the* i^{th} *block for* $i = 1, \cdots, q$.

2.2.2 Model Reduction by Balanced Truncation

The balanced basis has the property that the states which are difficult to reach are simultaneously difficult to observe. Hence, a reduced model is obtained by truncating the states which have this property, i.e., those which correspond to small Hankel singular values σ_i.

Theorem 2.2.5. *Let the asymptotically stable and minimal system* $\mathbf{\Sigma}$ *have the following balanced realization:*

$$\mathbf{\Sigma} = \left[\begin{array}{c|c} A_b & B_b \\ \hline C_b & D_b \end{array}\right] = \left[\begin{array}{cc|c} A_{11} & A_{12} & B_1 \\ A_{21} & A_{22} & B_2 \\ \hline C_1 & C_2 & D \end{array}\right],$$

with $\mathcal{P} = \mathcal{Q} = \operatorname{diag}(\Sigma_1, \Sigma_2)$ *where*

$$\Sigma_1 = \operatorname{diag}(\sigma_1 I_{m_1}, \cdots, \sigma_k I_{m_k}) \quad \text{and} \quad \Sigma_2 = \operatorname{diag}(\sigma_{k+1} I_{m_{k+1}}, \cdots, \sigma_q I_{m_q}).$$

Then the reduced order model $\mathbf{\Sigma}_r = \left[\begin{array}{c|c} A_{11} & B_1 \\ \hline C_1 & D \end{array}\right]$ *obtained by balanced truncation is asymptotically stable, minimal and satisfies*

$$\|\mathbf{\Sigma} - \mathbf{\Sigma}_r\|_{\mathcal{H}_\infty} \leq 2\,(\sigma_{k+1} + \cdots + \sigma_q). \tag{2.6}$$

The equality holds if Σ_2 *contains only* σ_q.

The above theorem states that if the neglected Hankel singular values are small, then the systems $\mathbf{\Sigma}$ and $\mathbf{\Sigma}_r$ are guaranteed to be close. Note that (2.6) is an *a priori* error bound. Hence, given an error tolerance, one can decide how many states to truncate without forming the reduced model.

The balancing method explained above is also called Lyapunov balancing since it requires solving two Lyapunov equations. Besides the Lyapunov balancing method, other types of balancing exist such as stochastic balancing [DP84, GRE88a, GRE88b], bounded real balancing , positive real balancing [DP84], LQG balancing [OJ88], and frequency weighted balancing [ENN84, LC92, SAM95, WSL99, ZHO95, VA01, GJ90, GA04]. For a recent survey of balancing related model reduction, see [GA04].

2.2.3 A Numerically Robust Implementation of Balanced Reduction

The above discussion on the balancing transformation and the balanced reduction requires balancing the whole system $\mathbf{\Sigma}$ followed by the truncation. This approach is numerically inefficient and very ill-conditioned to implement. Instead, below we will give another implementation of the balanced reduction which directly obtains a reduced balanced system without balancing the whole system.

Let $\mathcal{P} = UU^T$ and $\mathcal{Q} = LL^T$. This is always possible since both $\mathcal{P}$ and $\mathcal{Q}$ are symmetric positive definite matrices. The matrices U and L are called the *Cholesky factors* of the Gramians $\mathcal{P}$ and $\mathcal{Q}$, respectively. Let $U^T L = ZSY^T$ be a singular value decomposition (SVD). It is easy to show that the singular values of $U^T L$ are indeed the Hankel singular values, hence, we have

$$U^T L = Z\Sigma Y^T$$

where

$$\Sigma = \mathrm{diag}(\sigma_1 I_{m_1}, \sigma_2 I_{m_2}, \ldots, \sigma_q I_{m_q}),$$

q is the number of distinct Hankel singular values, with $\sigma_i > \sigma_{i+1} > 0$, m_i is the multiplicity of σ_i, and $m_1 + m_2 + \cdots + m_q = n$. Let

$$\Sigma_1 = \mathrm{diag}(\sigma_1 I_{m_1}, \sigma_2 I_{m_2}, \ldots, \sigma_k I_{m_k}), \quad k < q, \quad r := m_1 + \cdots + m_k,$$

and define

$$W_1 := LY_1\Sigma_1^{-1/2} \quad \text{and} \quad V_1 := UZ_1\Sigma_1^{-1/2},$$

where Z_1 and Y_1 are composed of the leading r columns of Z and Y, respectively. It is easy to check that $W_1^T V_1 = I_r$ and hence $V_1 W_1^T$ is an oblique projector. We obtain a reduced model of order r by setting

$$A_r = W_1^T A V_1, \quad B_r = W_1^T B, \quad C_r = C V_1.$$

Noting that $\mathcal{P} W_1 = V_1 \Sigma_1$ and $\mathcal{Q} V_1 = W_1 \Sigma_1$ gives

$$\begin{aligned} W_1^T (A\mathcal{P} + \mathcal{P} A^T + B B^T) W_1 &= A_r \Sigma_1 + \Sigma_1 A_r^T + B_r B_r^T \\ V_1^T (A^T \mathcal{Q} + \mathcal{Q} A + C^T C) V_1 &= A_r^T \Sigma_1 + \Sigma_1 A_r + C_r^T C_r. \end{aligned}$$

Thus, the reduced model is balanced and asymptotically stable (due to the Lyapunov inertia theorem) for any $k \leq q$. As mentioned earlier, the formulae above provide a numerically stable scheme for computing the reduced order model based on a numerically stable scheme for computing the Cholesky factors U and L directly in upper triangular and lower triangular form, respectively. It is important to truncate Z, Σ, Y to Z_1, Σ_1, Y_1 prior to forming W_1 or V_1. It is also important to avoid formulae involving inversion of L or U as these matrices are typically ill-conditioned due to the decay of the eigenvalues of the Gramians.

2.3 Iterative ADI Type Methods for Solving Large-Scale Lyapunov Equations

The numerically stable implementation of the balanced truncation method described in Section 2.2.3 requires the solutions to two Lyapunov equations of order n. For small-to-medium scale problems, the solutions can be obtained through the Bartels-Stewart [BS72] method as modified by Hammarling [HAM82]. This method requires the computation of a Schur decomposition, and thus is not appropriate for large-scale problems. The problem of obtaining the full-rank exact solution to a Lyapunov equation is a numerically ill-conditioned problem in the large-scale setting.

As explained previously, $\mathcal{P}$ and $\mathcal{Q}$ often have *numerically* low-rank compared to n. In most cases, the eigenvalues of $\mathcal{P}, \mathcal{Q}$ as well as the Hankel singular values $\sigma_i(\mathbf{\Sigma})$ decay very rapidly, see [ASZ02]. This *low-rank phenomenon* leads to the idea of approximating the Gramians with low-rank approximate Gramians.

In the following, we will focus on the approximate solution of the reachability Lyapunov equation

$$A\mathcal{P} + \mathcal{P} A^T + B B^T = 0, \tag{2.7}$$

where $A \in \mathbb{R}^{n \times n}$ is asymptotically stable and diagonalizable and $B \in \mathbb{R}^{n \times m}$. The discussion applies equally well to the observability Lyapunov equation $A^T \mathcal{Q} + \mathcal{Q} A + C^T C = 0$.

In this section we survey the ADI, Smith, and Smith(l) methods. In these methods the idea is to transform a continuous time Lyapunov equation (2.7) into a discrete time Stein equation using spectral transformations of the type $\omega(\lambda) = \frac{\mu^* - \lambda}{\mu + \lambda}$, where $\mu \in \mathbb{C}_-$ (the open left half-plane). Note that ω is a bilinear transformation mapping the open left half-plane onto the open unit disk with $\omega(\infty) = -1$. The number μ is called the *shift* or the *ADI parameter*.

2.3.1 The ADI Iteration

The alternating direction implicit (ADI) iteration was first introduced by Peaceman and Rachford [PR55] to solve linear systems arising from the discretization of elliptic boundary value problems. In general, the ADI iteration is used to solve linear systems of the form

$$My = b,$$

where M is symmetric positive definite and can be split into the sum of two symmetric positive definite matrices $M = M_1 + M_2$ for which the following iteration is efficient:

$$\begin{aligned} y_0 &= 0, \\ (M_1 + \mu_j I) y_{j-1/2} &= b - (M_2 - \mu_j I) y_{j-1}, \\ (M_2 + \eta_j I) y_j &= b - (M_1 - \eta_j I) y_{j-1/2}, \text{ for } j = 1, 2, \cdots, J. \end{aligned}$$

The ADI shift parameters μ_j and η_j are determined from spectral bounds on M_1 and M_2 to increase the convergence rate. When M_1 and M_2 commute, this is classified as a "model problem".

One should notice that (2.7) is a model ADI problem in which there is a linear system with the sum of two commuting operators acting on the unknown $\mathcal{P}$, which is a matrix in this case. Therefore, the iterates $\mathcal{P}_i^A$ of the ADI iteration are obtained through the iteration steps

$$(A + \mu_i I)\mathcal{P}_{i-1/2}^A = -BB^T - \mathcal{P}_{i-1}^A (A^T - \mu_i I) \tag{2.8}$$

$$(A + \mu_i I)\mathcal{P}_i^A = -BB^T - (\mathcal{P}_{i-1/2}^A)^* (A^T - \mu_i I), \tag{2.9}$$

where $\mathcal{P}_0^A = 0$ and the shift parameters $\{\mu_1, \mu_2, \mu_3, \ldots\}$ are elements of $\mathbb{C}_-$ (here $*$ denotes complex conjugation followed by transposition). These two equations are equivalent to the following single iteration step:

$$\begin{aligned} \mathcal{P}_i^A = (A - \mu_i^* I)(A + \mu_i I)^{-1} \mathcal{P}_{i-1}^A [(A - \mu_i^* I)(A + \mu_i I)^{-1}]^* \\ -2\rho_i (A + \mu_i I)^{-1} BB^T (A + \mu_i I)^{-*}, \end{aligned} \tag{2.10}$$

where $\rho_i = \text{Real}(\mu_i)$. Note that if $\mathcal{P}_{i-1}$ is Hermitian positive semi-definite, then so is $\mathcal{P}_i$.

The spectral radius of the matrix $\left(\prod_{i=1}^{l}(A-\mu_i^* I)(A+\mu_i I)^{-1}\right)$, denoted by ρ_{ADI}, determines the rate of convergence, where l is the number of shifts used. Note that since A is asymptotically stable, $\rho_{ADI} < 1$. Smaller ρ_{ADI} yields faster convergence. The minimization of ρ_{ADI} with respect to shift parameters μ_i is called the *ADI minimax problem*:

$$\{\mu_1, \mu_2, \ldots, \mu_l\} = \arg \min_{\{\mu_1,\ldots,\mu_l\}\in\mathbb{C}_-} \max_{\lambda\in\sigma(A)} \frac{|\,(\lambda-\mu_1^*)\ldots(\lambda-\mu_l^*)\,|}{|\,(\lambda+\mu_1)\ldots(\lambda+\mu_l)\,|}. \quad (2.11)$$

We refer the reader to [EW91, STA91, WAC90, CR96, STA93, WAC88b, PEN00b] for contributions to the solution of the ADI minimax problem. It can be shown that if A is diagonalizable, the l^{th} ADI iterate satisfies the inequality

$$\left\|\mathcal{P}-\mathcal{P}_l^A\right\|_F \le \|W\|_2^2 \left\|W^{-1}\right\|_2^2 \rho_{ADI}^2 \left\|\mathcal{P}\right\|_F, \quad (2.12)$$

where W is the matrix of eigenvectors of A.

The basic computational costs in the ADI iterations are that each individual shift μ_i requires a sparse direct factorization of $(A+\mu_i I)$ and each application of $(A+\mu_i I)^{-1}$ requires triangular solves from that factorization. Moreover, in the case of complex shifts, these operations have to be done in complex arithmetic. To keep the solution $\mathcal{P}$ real, complex conjugate pairs of shifts have to be applied, one followed immediately by the other. However, even with this, one would have to form $(A+\mu_i I)(A+\mu_i^* I) = A^2+2\rho_i A+|\mu_i|^2 I$ in order to keep the factorizations in real arithmetic. This matrix squaring would most likely have an adverse effect on sparsity. In the following, we wish to avoid the additional details required to discuss complex shifts. Therefore, *we will restrict our discussions to real shifts for the remainder of the paper.* If necessary, all of the operations can be made valid for complex shifts.

2.3.2 Smith's Method

For every real scalar $\mu < 0$, the continuous-time Lyapunov equation (2.7) is equivalent to

$$\mathcal{P} = (A-\mu I)(A+\mu I)^{-1}\mathcal{P}(A+\mu I)^{-^T}(A-\mu I)^T - 2\mu(A+\mu I)^{-1}BB^T(A^T+\mu I)^{-1}.$$

Then one obtains the Stein equation

$$\mathcal{P} = A_\mu \mathcal{P} A_\mu^T - 2\mu B_\mu B_\mu^T, \quad (2.13)$$

where

$$A_\mu := (A-\mu I)(A+\mu I)^{-1},\ B_\mu := (A+\mu I)^{-1}B. \quad (2.14)$$

Hence using the bilinear transformation $\omega(\lambda) = \frac{\mu-\lambda}{\mu+\lambda}$, the problem has been transformed into discrete time, where the Stein equation (2.13) has the same

solution as the continuous time Lyapunov equation (2.7). Since A is asymptotically stable, $\rho(A_\mu) < 1$ and the sequence $\{\mathcal{P}_i^S\}_{i=0}^{\infty}$ generated by the iteration

$$\mathcal{P}_1^S = -2\mu B_\mu B_\mu^T \text{ and } \mathcal{P}_{j+1}^S = A_\mu \mathcal{P}_j^S A_\mu^T + \mathcal{P}_1^S$$

converges to the solution $\mathcal{P}$. Thus, the Smith iterates can be written as

$$\mathcal{P}_k^S = -2\mu \sum_{j=0}^{k-1} A_\mu^j B_\mu B_\mu^T (A_\mu^j)^T. \tag{2.15}$$

If one uses the same shift through out the ADI iteration, ($\mu_j = \mu$, $j = 1, 2, \dots$), then the ADI iteration reduces to the Smith method. Generally, the convergence of the Smith method is slower than ADI. An accelerated version, the so called squared Smith method, has been proposed in [PEN00b] to improve convergence. However, despite a better convergence, the squared Smith methods destroys the sparsity of the problem which is not desirable in large-scale settings.

2.3.3 Smith(l) Iteration

Penzl [PEN00b] illustrated that the ADI iteration with a single shift converges very slowly, while a moderate increase in the number of shifts l accelerates the convergence nicely. However, he also observed that the speed of convergence is hardly improved by a further increase of l; see Table 2.1 in [PEN00b]. These observations led to the idea of the cyclic Smith(l) iteration, a special case of ADI where l different shifts are used in a cyclic manner, i.e. $\mu_{i+jl} = \mu_i$ for $j = 1, 2, \cdots$.

The Smith(l) iterates are generated by

$$\mathcal{P}_k^{Sl} = \sum_{j=0}^{k-1} A_d^k T (A_d^k)^T, \tag{2.16}$$

where

$$A_d = \prod_{i=1}^{l} (A - \mu_i I)(A + \mu_i I)^{-1} \qquad \text{and} \qquad T = \mathcal{P}_l^A, \tag{2.17}$$

i.e., T is the l^{th} ADI iterate with the shifts $\{\mu_1, \cdots, \mu_l\}$. As in Smith's methods, $\mathcal{P} - A_d \mathcal{P} A_d^T = T$ is equivalent to (2.7), where A_d and T are defined in (2.17).

2.4 Low-rank Iterative ADI-Type Methods

The original versions of the ADI, Smith, and Smith(l) methods outlined above form and store the entire dense solution $\mathcal{P}$ explicitly, resulting in extensive

storage requirement. In many cases the storage requirement is the limiting factor rather than the amount of computation. The observation that $\mathcal{P}$ is numerically low-rank compared to n leads to the low-rank formulations of the ADI iterations, namely, LR-ADI [PEN00b], CF-ADI [LW02], LR-Smith(l) [PEN00b], and Modified LR-Smith(l) [GSA03] where, instead of explicitly forming the solution $\mathcal{P}$, only the low-rank approximate Cholesky factors are computed and stored, reducing the storage requirement to $\mathcal{O}(nr)$ where r is the numerical rank of $\mathcal{P}$.

2.4.1 LR-ADI and CF-ADI Iterations

Recall that the two steps in (2.8) and (2.9) of the ADI iteration can be combined into the single iteration step in (2.10), as rewritten below:

$$\begin{aligned}\mathcal{P}_i^A = (A-\mu_i I)(A+\mu_i I)^{-1}\mathcal{P}_{i-1}^A[(A-\mu_i I)(A+\mu_i I)^{-1}]^T \\ -2\mu_i(A+\mu_i I)^{-1}BB^T(A+\mu_i I)^{-T}.\end{aligned} \tag{2.18}$$

The key idea in the low-rank versions of the ADI method is to rewrite the iterate $\mathcal{P}_i^A$ in (2.18) as an outer product:

$$\mathcal{P}_i^A = Z_i^A(Z_i^A)^T. \tag{2.19}$$

This is always possible since starting with the initial guess $\mathcal{P}_i^A = 0$, the iterates $\mathcal{P}_i^A$ can be shown recursively to be positive definite and symmetric.

Using (2.19) in (2.18) results in

$$\begin{aligned}Z_i^A(Z_i^A)^T = (A-\mu_i I)(A+\mu_i I)^{-1}Z_{i-1}^A[(A-\mu_i I)(A+\mu_i I)^{-1}Z_{i-1}^A]^T \\ -2\mu_i(A+\mu_i I)^{-1}BB^T(A+\mu_i I)^{-T}.\end{aligned} \tag{2.20}$$

Since the left-hand side of (2.20) is an outer product, and the right hand side is the sum of two outer products, Z_i^A can be rewritten as

$$Z_i^A = [\ (A-\mu_i I)(A+\mu_i I)^{-1}Z_{i-1}^A \qquad \sqrt{-2\mu_i}(A+\mu_i I)^{-1}B\ ]. \tag{2.21}$$

Therefore, the ADI algorithm (2.18) can be reformulated in terms of the Cholesky factor Z_i^A as

$$Z_1^A = \sqrt{-2\mu_1}(A+\mu_1 I)^{-1}B, \tag{2.22}$$

$$Z_i^A = [\ (A-\mu_i I)(A+\mu_i I)^{-1}Z_{i-1}^A \qquad \sqrt{-2\mu_i}(A+\mu_i I)^{-1}B\ ]. \tag{2.23}$$

This low-rank formulation of the ADI iteration was independently developed in [PEN00b] and [LW02]. We will call this the LR-ADI iteration as in [PEN00b] since it is the *preliminary* form of the final CF-ADI iteration [LW02]. In the LR-ADI formulation (2.22) and (2.23), at the i^{th} step, the $(i-1)^{\text{st}}$ Cholesky factor Z_{i-1}^A is multiplied from left by $(A-\mu_i I)(A+\mu_i I)^{-1}$.

Therefore, the number of columns to be modified at each step increases by m, the number of columns in B. In [LW02], the steps (2.22) and (2.23) are reformulated to keep the number of columns modified at each step as constant. The resulting algorithm, outlined below, is called the CF-ADI iteration.

The columns of k^{th} LR-ADI iterate Z_i^A can be written out explicitly as

$$Z_k^A = [S_k\sqrt{-2\mu_k}B,\ S_k(T_kS_{k-1})\sqrt{-2\mu_{k-1}}B,\ \cdots,\ S_kT_k\cdots S_2(T_2S_1)\sqrt{-2\mu_1}B]$$

where

$$S_i := (A+\mu_i I)^{-1}, \quad \text{and} \quad T_i := (A-\mu_i I) \quad \text{for} \quad i=1,\ldots,k.$$

Since S_i and T_j commute, i.e.

$$S_iS_j = S_jS_i, \quad T_iT_j = T_jT_i, \quad S_iT_j = T_jS_i, \quad \forall i,j,$$

Z_k^A can be written as

$$Z_k^A = [\ z_k \quad P_{k-1}(z_k), \quad P_{k-2}(P_{k-1}z_k), \quad \cdots\cdots \quad P_1(P_2\cdots P_{k-1}z_k)], \tag{2.24}$$

where

$$z_k := \sqrt{-2\mu_k}\ (A+\mu_k I)^{-1}B, \tag{2.25}$$

$$P_i := \frac{\sqrt{-2\mu_i}}{\sqrt{-2\mu_{i+1}}}\ [I-(\mu_{i+1}+\mu_i)(A+\mu_i I)^{-1}]. \tag{2.26}$$

Since the order of the ADI parameters μ_i is not important, the ordering of μ_i can be reversed resulting in the CF-ADI iteration:

$$Z_1^{CFA} = z_1 = \sqrt{-2\mu_1}\ (A+\mu_1 I)^{-1}B, \tag{2.27}$$

$$z_i = \left(\frac{\sqrt{-2\mu_i}}{\sqrt{-2\mu_{i-1}}}\right)\ \left(I-(\mu_i+\mu_{i-1})(A+\mu_i I)^{-1}\right) z_{i-1}, \tag{2.28}$$

$$Z_i^{CFA} = [Z_{i-1}^{CFA} \quad z_i], \qquad \text{for } i=2,\cdots,k. \tag{2.29}$$

Unlike the LR-ADI iteration (2.22)-(2.23) where at the i^{th} step $(i-1)m$ number of columns need to be modified, the CF-ADI iteration (2.27)-(2.29) requires only that a *constant* number of columns, namely, m, to be modified at each step. Therefore, the implementation of CF-ADI is numerically more efficient compared to LR-ADI.

Define $\mathcal{P}_j^{CFA} := Z_j^{CFA}(Z_j^{CFA})^T$. Clearly, the stopping criterion $\|\mathcal{P}_j^{CFA} - \mathcal{P}_{j-1}^{CFA}\|_2 \leq tol^2$ can be implemented as $\|z_j\|_2 \leq tol$, since

$$\|Z_j^{CFA}(Z_j^{CFA})^T - Z_{j-1}^{CFA}(Z_{j-1}^{CFA})^T\|_2 = \|z_jz_j^T\|_2 = \|z_j\|_2^2.$$

It is not necessarily true that a small z_j implies that all further z_{j+k} will be small, but this has been observed in practice. Relative error can also be used, in which case the stopping criterion is

$$\frac{\|z_j\|_2}{\|Z_{j-1}^{CFA}\|_2} \leq tol.$$

The 2-norm of Z_{j-1}^{CFA}, which is also its largest singular value, can be estimated by performing power iterations to estimate the largest eigenvalue of $Z_{j-1}^{CFA}(Z_{j-1}^{CFA})^T$, taking advantage of the fact that $j \ll n$. This cost is still high, and this estimate should only be used after each segment of several iterations.

The next result shows the relation between the ADI, LR-ADI and CF-ADI iterations. For a proof, see the original source [LW02].

Theorem 2.4.1. *Let $\mathcal{P}_k^A$ be the approximation obtained by k steps of the ADI iteration with shifts $\{\mu_1, \mu_2, \ldots, \mu_k\}$. Moreover, for the same shift selection, let Z_k^A and Z_k^{CFA} be the approximations obtained by the LR-ADI and the CF-ADI iterations as above, respectively. Then,*

$$\mathcal{P}_k^A = Z_k^A(Z_k^A)^T = Z_k^{CFA}(Z_k^{CFA})^T.$$

2.4.2 LR-Smith(l) Iteration

The ADI, LR-ADI, and CF-ADI iterations are of interest if a sequence $\{\mu_i\}_{i=1}^k$ of different shifts is available. When the number of shift parameters is limited, the cyclic low-rank Smith method (LR-Smith(l)) is a more efficient alternative. As in the LR-ADI formulation of the ADI iteration, the key idea is to write the i^{th} Smith(l) iterate as

$$\mathcal{P}_i^{Sl} = Z_i^{Sl}(Z_i^{Sl})^T. \tag{2.30}$$

Given the l cyclic-shifts $\{\mu_1, \mu_2, \ldots, \mu_l\}$, the LR-Smith($l$) method consists of two steps. First the iterate Z_1^{Sl} is obtained by an l step low-rank ADI iteration; i.e. $P_l^A = Z_l^A(Z_l^A)^T$ is the low-rank l step ADI iterate. Then, the LR-Smith(l) method is initialized by

$$Z_1^{Sl} = B_d = Z_l^A, \tag{2.31}$$

followed by the actual LR-Smith(l) iteration:

$$\begin{aligned} Z^{(i+1)} &= A_d Z^{(i)} \\ Z_{i+1}^{Sl} &= [\ Z_i^{Sl} \quad Z^{(i+1)}\ ], \end{aligned} \tag{2.32}$$

where A_d is defined in (2.17). It then follows that

$$Z_k^{Sl} = [\ B_d \ \ A_d B_d \ \ A_d^2 B_d \ \ \cdots \ \ A_d^{k-1} B_d\ ]. \tag{2.33}$$

One should notice that while k step LR-ADI and CF-ADI iterations require k matrix factorizations, a k step LR-Smith(l) iteration computes only l matrix

factorizations. Moreover, the equality (2.33) reveals that similar to the CF-ADI iteration, the number of columns to be modified at the i^{th} step of the LR-Smith(l) iteration is constant, equal to the number of columns of B_d, namely $l \times m$. If the shifts $\{\mu_1, \cdots, \mu_l\}$ are used in a cyclic manner, the cyclic LR-Smith(l) iteration gives the same approximation as the LR-ADI iteration.

Remark 2.4.2. A system theoretic interpretation of using l cyclic shifts (the Smith(l) iteration) is that the continuous time system

$$\mathbf{\Sigma} = \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$$

which has order n, m inputs, and p outputs is embedded into a discrete time system

$$\mathbf{\Sigma}_d = \left[\begin{array}{c|c} A_d & B_d \\ \hline C_d & D_d \end{array}\right]$$

which has order n, lm inputs, and lp outputs; they have the same reachability and observability Gramians $\mathcal{P}$ and $\mathcal{Q}$. Therefore, at the cost of increasing the number of inputs and outputs, one reduces the spectral radius $\rho(A_d)$ and hence increases the convergence.

Remark 2.4.3. Assume that we know all the eigenvalues of A and the system

$$\mathbf{\Sigma} = \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$$

is single input single output, i.e. $B, C^T \in \mathbb{R}^n$. Then choosing $\mu_i = \lambda_i(A)$ for $i = 1, \cdots, n$ results in

$$A_d = 0 \qquad \text{and} \qquad \mathcal{P} = \mathcal{P}_1^{Sl} = \mathcal{P}_l^A.$$

In other words, the exact solution $\mathcal{P}$ of (2.7) is obtained at the first step. The resulting discrete time system has n inputs, and n outputs.

Convergence Results for the Cyclic LR-Smith(l) Iteration

In this section some convergence results for the Cyclic LR-Smith(l) iteration are presented. For more details, we refer the reader to the original source [GSA03].

Let Z_k^{Sl} be the k^{th} LR-Smith(l) iterate as defined in (2.33) corresponding to the Lyapunov equation

$$A\mathcal{P} + \mathcal{P}A^T + BB^T = 0.$$

Similar to Z_k^{Sl}, let Y_k^{Sl} be the k^{th} LR-Smith(l) iterate corresponding to the observability Lyapunov equation

$$A^T\mathcal{Q} + \mathcal{Q}A + C^TC = 0$$

for the same cyclic shift selection as used in computing Z_k^{Sl}.

Denote by $\mathcal{P}_k^{Sl}$ and $\mathcal{Q}_k^{Sl}$ the k step LR-Smith(l) iterates for $\mathcal{P}$ and $\mathcal{Q}$ respectively, i.e., $\mathcal{P}_k^{Sl} = Z_k^{Sl}(Z_k^{Sl})^T$ and $\mathcal{Q}_k^{Sl} = Y_k^{Sl}(Y_k^{Sl})^T$. Similar to (2.12), the following result holds:

Proposition 2.4.4. *Let $E_{kp} := \mathcal{P} - \mathcal{P}_k^{Sl}$ and $E_{kq} = \mathcal{Q} - \mathcal{Q}_k^{Sl}$ and $A = W(\Lambda)W^{-1}$ be the eigenvalue decomposition of A. The k step LR-Smith(l) iterates satisfy*

$$\begin{aligned} 0 &\leq \operatorname{trace}(E_{kp}) = \operatorname{trace}(\mathcal{P} - \mathcal{P}_k^{Sl}) \leq K\, m\, l\, (\rho(A_d))^{2k} \operatorname{trace}(\mathcal{P}) \quad (2.34)\\ 0 &\leq \operatorname{trace}(E_{kq}) = \operatorname{trace}(\mathcal{Q} - \mathcal{Q}_k^{Sl}) \leq K\, p\, l\, (\rho(A_d))^{2k} \operatorname{trace}(\mathcal{Q}), \quad (2.35)\end{aligned}$$

where

$$K = \kappa(W)^2, \tag{2.36}$$

and $\kappa(W)$ denotes the 2-norm condition number of W.

Since the low-rank Cholesky factors Z_k^{Sl} and Y_k^{Sl} will be used for balanced truncation of the underlying dynamical system, it is important to see how well the exact Hankel singular values are approximated. Let σ_i and $\hat{\sigma}_i$ denote the Hankel singular values resulting from the full-rank exact Gramians and the low-rank approximate Gramians, respectively, i.e.,

$$\sigma_i^2 = \lambda_i(\mathcal{P}\mathcal{Q}) \text{ and } \hat{\sigma}_i^2 = \lambda_i(\mathcal{P}_k^{Sl}\mathcal{Q}_k^{Sl}). \tag{2.37}$$

The following lemma holds:

Lemma 2.4.5. *Let σ_i and $\hat{\sigma}_i$ be given by (2.37). Define $\hat{n} = kl\min(m,p)$. Then,*

$$\begin{aligned} 0 &\leq \sum_{i=1}^{n}\sigma_i^2 - \sum_{i=1}^{\hat{n}}\hat{\sigma}_i^2\\ &\leq K\, l\, (\rho(A_d))^{2k}\Big(K\ \min(m,p)(\rho(A_d))^{2k}\operatorname{trace}(\mathcal{P})\operatorname{trace}(\mathcal{Q})\\ &\quad + m\operatorname{trace}(\mathcal{P})\sum_{i=0}^{k-1}\|C_dA_d^i\|_2^2 + p\operatorname{trace}(\mathcal{Q})\sum_{i=0}^{k-1}\|A_d^iB_d\|_2^2\Big) \end{aligned} \tag{2.38}$$

where K is as defined in (2.36).

As mentioned in [GSA03], these error bounds critically depend on $\rho(A_d)$ and K. Hence when $\rho(A_d)$ is almost 1 and/or A is highly non-normal, the bounds may be pessimistic. On the other hand, when $\rho(A_d)$ is small, for example less than 0.9, the convergence of the iteration is extremely fast and also the error bounds are tight.

2.4.3 The Modified LR-Smith(l) Iteration

It follows from the implementations of the LR-ADI, the CF-ADI, and the LR-Smith(l) iterations that at each step the number of columns of the current iterates is increased by m for the LR-ADI and CD-ADI methods, and by $m \times l$ for the LR-Smith(l) method. Hence, when m is large, i.e. for MIMO systems, or when the convergence is slow, i.e., $\rho(A_d)$ is close to 1, the number of columns of Z_k^A, Z_k^{CFA}, and Z_k^{Sl} might exceed available memory. In light of these observations, Gugercin *et. al.* [GSA03] introduced a modified LR-Smith(l) iteration where the number of columns in the low-rank Cholesky factor does not increase unnecessarily at each step. The idea is to compute the singular value decomposition of the iterate at each step and, given a tolerance τ, to replace the iterate with its best low-rank approximation as outlined below.

Let Z_k^{Sl} be the k^{th} LR-Smith(l) iterate as defined in (2.33) corresponding to the Lyapunov equation $A\mathcal{P}+\mathcal{P}A^T+BB^T=0$. Let the short singular value decomposition (S-SVD) of Z_k^{Sl} be

$$Z_k^{Sl} = V\Phi F^T,$$

where $V \in \mathbb{R}^{n\times(mlk)}$, $\Phi \in \mathbb{R}^{(mlk)\times(mlk)}$, and $F \in \mathbb{R}^{(mlk)\times(mlk)}$. Then the S-SVD of $\mathcal{P}_k^{Sl} = Z_k^{Sl}(Z_k^{Sl})^T$ is given by $\mathcal{P}_k^{Sl} = V\Phi^2V^T$. Therefore, it is enough to store only V and Φ, and

$$\tilde{Z}_k := V\Phi$$

is also a low-rank Cholesky factor for $\mathcal{P}_k^{Sl}$.

For a pre-specified tolerance value $\tau > 0$, assume that until the k^{th} step of the algorithm all the iterates $\mathcal{P}_i^{Sl}$ satisfy

$$\frac{\sigma_{min}(\mathcal{P}_i^{Sl})}{\sigma_{max}(\mathcal{P}_i^{Sl})} > \tau^2 \text{ or equivalently } \frac{\sigma_{min}(Z_i^{Sl})}{\sigma_{max}(Z_i^{Sl})} = \frac{\sigma_{min}(\tilde{Z}_i)}{\sigma_{max}(\tilde{Z}_i)} > \tau$$

for $i = 1, 2, \cdots, k$, where σ_{min} and σ_{max} denote the minimum and maximum singular values, respectively. It readily follows from the implementation of the LR-Smith(l) method that at the $(k+1)^{st}$ step, the approximants Z_{k+1}^{Sl} and $\mathcal{P}_{k+1}^{Sl}$ are given by

$$Z_{k+1}^{Sl} = [Z_k^{Sl} \quad A_d^k B_d] \qquad \text{and} \qquad \mathcal{P}_{k+1}^{Sl} = \mathcal{P}_k^{Sl} + A_d^k B_d B_d^T (A_d^k)^T.$$

Decompose $A_d^k B_d$ into the two spaces $Im(V)$ and $(Im(V))^{\perp}$; i.e., write

$$A_d^k B_d = V\Gamma + \hat{V}\Theta, \tag{2.39}$$

where $\Gamma \in \mathbb{R}^{(mlk)\times(ml)}$, $\Theta \in \mathbb{R}^{(ml)\times(ml)}$, $V^T\hat{V} = 0$ and $\hat{V}^T\hat{V} = I_{ml}$. Define the matrix

$$\hat{Z}_{k+1} = [\ V \quad \hat{V}\ ] \underbrace{\begin{bmatrix} \Phi & \Gamma \\ 0 & \Theta \end{bmatrix}}_{\hat{S}}. \tag{2.40}$$

Let $\hat{S}$ have the following SVD: $\hat{S} = T\hat{\Phi}Y^T$. Then it follows that $\tilde{Z}_{k+1}$ is given by

$$\tilde{Z}_{k+1} = \tilde{V}\hat{\Phi}, \qquad \tilde{V} = [\ V \quad \hat{V}\ ]T, \tag{2.41}$$

where $\tilde{V} \in \mathbb{R}^{n\times((k+1)ml)}$ and $\hat{\Phi} \in \mathbb{R}^{((k+1)ml)\times((k+1)ml)}$. Note that computation of $\tilde{Z}_{k+1}$ requires the knowledge of $\tilde{Z}_k$, which is already available, and the SVD of $\hat{S}$, which is easy to compute. Next, partition $\hat{\Phi}$ and $\tilde{V}$ conformally:

$$\tilde{Z}_{k+1} = [\ \tilde{V}_1 \quad \tilde{V}_2\ ] \begin{bmatrix} \hat{\Phi}_1 & \\ & \hat{\Phi}_2 \end{bmatrix} \text{ so that } \frac{\hat{\Phi}_2(1,1)}{\hat{\Phi}_1(1,1)} < \tau. \tag{2.42}$$

Then, the $(k+1)^{st}$ low-rank Cholesky factor is approximated by

$$\tilde{Z}_{k+1} \approx \tilde{V}_1\hat{\Phi}_1. \tag{2.43}$$

$\tilde{Z}_{k+1}$ in (2.43) is the $(k+1)^{st}$ modified LR-Smith(l) iterate. In computing $\tilde{Z}_{k+1}$, the singular values which are less than the given tolerance τ are truncated. Hence, in going from the k^{th} to the $(k+1)^{st}$ step, the number of columns of $\tilde{Z}_{k+1}$ generally does not increase. An increase will only occur if more than r singular values of $\tilde{Z}_{k+1}$ are above the tolerance $\tau\sigma_1$. In the worst case, at most ml additional columns will be added at any step which is the same as the unmodified LR-Smith(l) iteration discussed in Section 2.4.1.

Using $\tilde{Z}_{k+1}$ in (2.43), the $(k+1)^{st}$ step modified low-rank Smith Gramian is given by

$$\tilde{\mathcal{P}}_{k+1} := \tilde{Z}_{k+1}(\tilde{Z}_{k+1})^T = \tilde{V}_1\hat{\Phi}_1\hat{\Phi}_1^T\tilde{V}_1^T.$$

Convergence Properties of the Modified LR-Smith(l) Iteration

Let $\tilde{\mathcal{P}}_k$ and $\tilde{\mathcal{Q}}_k$ be the k step modified LR-Smith(l) solutions to the two Lyapunov equations $A\mathcal{P} + \mathcal{P}A^T + BB^T = 0$, $\quad A^T\mathcal{Q} + \mathcal{Q}A + C^TC = 0$, respectively, where $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, and $C \in \mathbb{R}^{p\times n}$. Moreover let $\mathcal{I}_\mathcal{P}$ denote the set of indices i for which some columns have been eliminated from the i^{th} step approximant during the modified Smith iteration:

$$\mathcal{I}_\mathcal{P} = \{i : \text{such that in (2.42) } \hat{\Phi}_2 \neq 0 \text{ for } \tilde{Z}_i,\ i = 1, 2, \cdots, k\}.$$

Then for each $i \in \mathcal{I}_\mathcal{P}$, let $n_i^\mathcal{P}$ denote the number of the neglected singular values. Similarly define $\mathcal{I}_\mathcal{Q}$ and $n_i^\mathcal{Q}$. The following convergence result holds [GSA03].

Theorem 2.4.6. *Let $\mathcal{P}_k^{Sl}$ be the k^{th} step LR-Smith(l) iterate. $\Delta_{kp} := \mathcal{P}_k^{Sl} - \tilde{\mathcal{P}}_k$, the error between the LR-Smith(l) and Modified LR-Smith(l) iterates, satisfies*

$$\|\Delta_{kp}\| = \|\mathcal{P}_k^{Sl} - \tilde{\mathcal{P}}_k\| \leq \tau^2 \sum_{i \in \mathcal{I}_{\mathcal{P}}} (\sigma_{max}(\tilde{Z}_i))^2, \tag{2.44}$$

where τ is the tolerance value of the modified LR-Smith(l) algorithm. Moreover, define $\tilde{E}_{kp} = \mathcal{P} - \tilde{\mathcal{P}}_k$, the error between the exact solution and the k^{th} Modified LR-Smith(l) iterates. Then,

$$\begin{aligned} 0 &\leq \operatorname{trace}(\tilde{E}_{kp}) \\ &\leq K\, m\, l\, (\rho(A_d))^{2k} \operatorname{trace}(\mathcal{P}) + \tau^2 \sum_{i \in \mathcal{I}_{\mathcal{P}}} n_i^{\mathcal{P}}\ (\sigma_{max}(\tilde{Z}_i))^2, \end{aligned} \tag{2.45}$$

where K is given by (2.36).

Note that the error $\|\Delta_{kp}\|$ is in the order of $\mathcal{O}(\tau^2)$. This means with a lower number of columns in the approximate Cholesky factor, the Modified Smith method will yield almost the same accuracy as the exact Smith method.

The next result concerns the convergence of the computed Hankel singular values in a way analogous to Lemma 2.4.5.

Lemma 2.4.7. *Let σ_i and $\tilde{\sigma}_i$ denote Hankel singular values resulting from the full-rank exact Gramians $\mathcal{P}$ and $\mathcal{Q}$ and from the modified LR-Smith(l) approximants $\tilde{\mathcal{P}}_k$ and $\tilde{\mathcal{Q}}_k$ respectively: ${\sigma_i}^2 = \lambda_i(\mathcal{P}\mathcal{Q})$ and $\tilde{\sigma}_i^2 = \lambda_i(\tilde{\mathcal{P}}_k\tilde{\mathcal{Q}}_k)$. Define $\hat{n} = kl\min(m,p)$. Then,*

$$\begin{aligned} 0 &\leq \sum_{i=1}^{n} \sigma_i^2 - \sum_{i=1}^{\hat{n}} \tilde{\sigma}_i^2 \\ &\leq K\, l\, (\rho(A_d))^{2k} \Big(K\ \min(m,p)(\rho(A_d))^{2k} \operatorname{trace}(\mathcal{P}) \operatorname{trace}(\mathcal{Q}) \\ &\quad + m \operatorname{trace}(\mathcal{P}) \sum_{i=0}^{k-1} \|C_d A_d^j\|_2^2 + p \operatorname{trace}(\mathcal{Q}) \sum_{i=0}^{k-1} \|A_d^j B_d\|_2^2 \Big) \\ &\quad + \tau_{\mathcal{P}}^2 \|\mathcal{Q}_k^{Sl}\|_2 \sum_{i \in \mathcal{I}_{\mathcal{P}}} n_i^{\mathcal{P}}\ (\sigma_{max}(\tilde{Z}_i))^2 \\ &\quad + \tau_{\mathcal{Q}}^2 \|\mathcal{P}_k^{Sl}\|_2 \sum_{i \in \mathcal{I}_{\mathcal{Q}}} n_i^{\mathcal{Q}}\ (\sigma_{max}(\tilde{Y}_i))^2) \end{aligned} \tag{2.46}$$

where $\tau_{\mathcal{P}}$ and $\tau_{\mathcal{Q}}$ are the given tolerance values; and K is as defined in (2.36).

Once again the bounds in Lemma 2.4.5 and Lemma 2.4.7 differ only by the summation of terms of $\mathcal{O}(\tau_{\mathcal{P}}^2)$ and $\mathcal{O}(\tau_{\mathcal{Q}}^2)$.

2.4.4 ADI Parameter Selection

As the selection of good parameters is vitally important to the successful application of the ADI and derived algorithms, in this section we discuss two possible approaches. Both seek to solve the minimax problem (2.11), in other words, minimizing the right hand side of the error bound in (2.12).

Because it is not practical to assume the knowledge of the complete spectrum of the matrix A, i.e., not practical to solve (2.11) over $\lambda \in \sigma(A)$, the first approach [WAC95] solves a different problem. It begins by bounding the spectrum of A inside a domain $\mathcal{R} \subset \mathbb{C}_-$, in other words,

$$\lambda_1(A), \cdots, \lambda_n(A) \in \mathcal{R} \subset \mathbb{C}_-,$$

and then solves the following rational minimax problem:

$$\min_{\mu_1,\mu_2,\cdots,\mu_l} \max_{x\in\mathcal{R}} \left| \prod_{j=1}^{l} \frac{(\mu_j - x)}{(\mu_j + x)} \right|, \tag{2.47}$$

where the maximization is done over $x \in \mathcal{R}$ (rather than $\lambda \in \sigma(A)$). In this general formulation, $\mathcal{R}$ can be any region in the open left half plane.

If the eigenvalues of A are strictly real, then one takes the domain $\mathcal{R}$ to be a line segment, with the end points being the extremal eigenvalues of A. In this case the solution to (2.47) is known (see [WAC95]). Power and inverse iterations can be used to estimate the extremal eigenvalues of A at a low cost.

If A has complex eigenvalues, finding a good domain $\mathcal{R}$ which provides an efficient covering of the spectrum of A can be involved, since the convex hull of the spectrum of an arbitrary stable matrix can take on widely varying shapes. Typically one estimates extremal values of the spectrum of A, along the real and the imaginary axes, and then assumes that the spectrum is bounded inside some region which can be simply defined by the extremal values one has obtained.

However, even after a good $\mathcal{R}$ has been obtained, there remains the serious difficulty of solving (2.47). The solution to (2.47) is not known when $\mathcal{R}$ is an arbitrary region in the open left half plane. However, the problem of finding optimal and near-optimal parameters for a few given shapes was investigated in several papers [IT95, EW91, STA91, STA93, WAC62, WAC95] and we give some of the useful results below.

In particular, we summarize a parameter selection procedure from [WAC95] which defines the spectral bounds a, b, and α for the matrix A as

$$a = \min_i(Re\{\gamma_i\}), \quad b = \max_i(Re\{\gamma_i\}), \quad \alpha = \tan^{-1} \max_i \left| \frac{Im\{\gamma_i\}}{Re\{\gamma_i\}} \right|, \tag{2.48}$$

where $\gamma_1, \cdots, \gamma_n$ are the eigenvalues of $-A$. It is assumed that the spectrum of $-A$ lies entirely inside a region which was called in that reference the "elliptic function domain" determined by the numbers a, b, α. The specific definition

of the "elliptic function domain" can be found in [WAC95]. If this assumption does not hold, one should try to apply a more general parameter selection algorithm. If it does hold, then let

$$\cos^2\beta = \frac{2}{1+\frac{1}{2}(\frac{a}{b}+\frac{b}{a})},$$
$$m = \frac{2\cos^2\alpha}{cos^2\beta} - 1.$$

If $m < 1$, the parameters are complex, and are given in [EW91, WAC95]. If $m \geq 1$, the parameters are real, and we define

$$k' = \frac{1}{m+\sqrt{m^2-1}}, \quad k = \sqrt{1-k'^2}.$$

Note $k' = \frac{a}{b}$ if all the eigenvalues of A are real. Define the elliptic integrals K and v as,

$$F[\psi,k] = \int_0^{\psi} \frac{dx}{\sqrt{1-k^2\sin^2 x}},$$
$$K = K(k) = F\left[\frac{\pi}{2},k\right], \qquad v = F\left[\sin^{-1}\sqrt{\frac{a}{bk'}},k'\right].$$

The number of ADI iterations required to achieve $\rho^2_{ADI} \leq \epsilon_1$ is given by $l = \left\lceil \frac{K}{2v\pi}\log\frac{4}{\epsilon_1}\right\rceil$, and the ADI parameters are given by

$$\mu_j = -\sqrt{\frac{ab}{k'}}dn\left[\frac{(2j-1)K}{2l},k\right], \quad j = 1,2,\cdots,l, \tag{2.49}$$

where $dn(u,k)$ is the elliptic function. It was noted in [LW91] that for many practical problems ADI converges in a few iterations with these parameters.

A second approach to the problem of determining ADI parameters is a heuristic one and was given in [PEN00b]. It chooses potential parameters from a list $\mathcal{S} = \{\rho_1,\rho_2,\cdots,\rho_k\}$ which is taken to be the union of the Ritz values of A and the reciprocals of the Ritz values of A^{-1}, obtained by two Arnoldi processes, with A and A^{-1}. From this list $\mathcal{S}$, one chooses the list of l ADI parameters, $\mathcal{L}$, in the following way. First, we define the quantity

$$s_{\mathcal{M}}(x) := \frac{|(x-\mu_1)\times\cdots\times(x-\mu_m)|}{|(x+\mu_1)\times\cdots\times(x+\mu_m)|},$$

where $\mathcal{M} = \{\mu_1,\cdots,\mu_m\}$. The algorithm proceeds as follows:

1. Find i such that $\max\limits_{x\in\mathcal{S}} s_{\rho_i}(x) = \min\limits_{\rho_i\in\mathcal{S}}\max\limits_{x\in\mathcal{S}} s_{\rho_i}(x)$ and let

$$\mathcal{L} := \begin{cases} \{\rho_i\} & \text{if } \rho_i \text{ real,} \\ \{\rho_i,\bar{\rho}_i\} & \text{otherwise.} \end{cases}$$

2. While $\text{card}(\mathcal{L}) < l$, find i such that $s_{\mathcal{L}}(\rho_i) = \max_{x \in \mathcal{L}} s_{\mathcal{L}}(x)$ and let

$$\mathcal{L} := \begin{cases} \mathcal{L} \cup \{\rho_i\} & \text{if } \rho_i \text{ real,} \\ \mathcal{L} \cup \{\rho_i, \bar{\rho}_i\} & \text{otherwise.} \end{cases}$$

The procedure is easy to implement and good results have been obtained [PEN00b].

2.5 Smith's Method and Eigenvalue Decay Bounds for Gramians

As discussed earlier, in most cases, the eigenvalues of the reachability and observability Gramians $\mathcal{P}, \mathcal{Q}$, as well as the Hankel singular values, i.e., $\sqrt{\lambda_i(\mathcal{PQ})}$, decay very rapidly. In this section, we briefly review the results of [ASZ02, ZHO02] and reveal the connection to convergence of Smith-type iterations. We will again consider the Lyapunov equation

$$A\mathcal{P} + \mathcal{P}A^T + BB^T = 0, \tag{2.50}$$

where $B \in \mathbb{R}^{n \times m}$ with $m \ll n$, $A \in \mathbb{R}^{n \times n}$ is asymptotically stable, and the pair (A, B) is reachable.

2.5.1 Eigenvalue Decay Bounds for the Solution $\mathcal{P}$

Given the Lyapunov equation (2.50), let an l step ADI iteration be computed using the shifts μ_i, with $\mu_i < 0$ where $i = 1, \ldots, l$ and $lm < n$. Then it simply follows from (2.10) that

$$\text{rank}(\mathcal{P}_{i-1}^A) \leq \text{rank}(\mathcal{P}_i^A) \leq \text{rank}(\mathcal{P}_{i-1}^A) + m.$$

Hence, at the l^{th} step, one has

$$\text{rank}(\mathcal{P}_l^A) \leq lm.$$

Then by Schmidt-Mirsky theorem and considering $\mathcal{P}_l^A$ as a low-rank approximation to $\mathcal{P}$, one simply obtains

$$\frac{\lambda_{lm+1}(\mathcal{P})}{\lambda_1(\mathcal{P})} \leq \|A_d\|_2^2,$$

where A_d is given by (2.17). The following result holds:

Theorem 2.5.1. *Given the above set-up, let A be diagonalizable. Then, eigenvalues of the solution $\mathcal{P}$ to the Lyapunov equation (2.50) satisfy*

$$\frac{\lambda_{lm+1}(\mathcal{P})}{\lambda_1(\mathcal{P})} \leq K(\rho(A_d))^2, \tag{2.51}$$

where $lm < n$, K is given by (2.36), $\rho_{ADI} = \rho(A_d)$ as before and the shifts μ_i are chosen by solving the ADI minimax problem (2.11).

See the original source [ASZ02] and [ZHO02] for details and a proof.

2.5.2 Connection Between Convergence of the Smith Iteration and Theorem 2.5.1

Smith (or ADI) type iterations try to approximate the exact Gramian $\mathcal{P}$ with a low-rank version in which the convergence of the iteration is given by either (2.12) or Proposition (2.4.4). Hence, if $\rho(A_d)$ is close to 1 and/or K is big, we expect slow convergence. The slow convergence leads to more steps in the Smith iteration, and, consequently, the rank of the approximant is higher. Since $\mathcal{P}$ is positive definite, in turn, this means that eigenvalues of $\mathcal{P}$ do not decay rapidly. Therefore $\rho(A_d) \approx 1$ and/or K is big mean that $\lambda_i(\mathcal{P})$ might decay slowly. This final remark is consistent with the above decay bound (2.51). These relations are expected since (2.51) is derived via the ADI iteration.

As stated in [ZHO02] and [ASZ02], (2.51) yields the following remarks:

1. If $\lambda_i(A)$ are clustered in the complex plane, choosing the shifts μ_i as the clustered points yields a small $\rho(A_d)$, and consequently fast decay of $\lambda_i(\mathcal{P})$. Hence, the convergence of an ADI-type iteration is fast.
2. If $\lambda_i(A)$ have mostly dominant real parts, then the decay rate is again fast. Hence, as above, the convergence of an ADI-type iteration is fast.
3. If $\lambda_i(A)$ have mostly dominant imaginary parts, while the real parts are relatively small, the decay rate $\lambda_i(\mathcal{P})$ is slow. Then an ADI iteration converges slowly.

These observations agree with the numerical simulations. In Example 2.7.2, the Smith(l) method is applied to a CD player example, a system of order 120, where the eigenvalues of A are scattered in the complex plane with dominant complex parts. Even with a high number of shifts, $\rho(A_d)$ cannot be reduced less than 0.98, and the Smith methods converge very slowly. Indeed, an exact computation of $\mathcal{P}$ reveals that $\mathcal{P}$ does not have rapidly decaying eigenvalues. Also, it was shown in [ASG01] that the Hankel singular values of this system decay slowly as well, and the CD player was among the hardest models to approximate. These results are consistent with item 3. above.

Item 2. is encountered in Example 2.7.2, where the Smith method is applied to a model of order 1006. 1000 of the eigenvalues are real and only the remaining 6 are complex. By choosing the shifts as the complex eigenvalues, $\rho(A_d)$ is reduced to a small value and convergence is extremely fast. Indeed, using the modified Smith method, the exact Gramians are approximated very well with low-rank Gramians having rank of only 19. We note that the shifts are even not the optimal ones.

2.6 Approximate Balanced Truncation and its Stability

Recall the implementation of balanced truncation presented in Section 2.2.3. An exact balanced truncation requires the knowledge of Cholesky factors U

and L of the Gramians $\mathcal{P}$ and $\mathcal{Q}$, i.e. $\mathcal{P} = UU^T$ and $\mathcal{Q} = LL^T$ where $\mathcal{P}$ and $\mathcal{Q}$ are the solutions to the two Lyapunov equations

$$A\mathcal{P} + \mathcal{P}A^T + BB^T = 0 \quad \text{and} \quad A^T\mathcal{Q} + \mathcal{Q}A + C^TC = 0.$$

As mentioned earlier, in large-scale settings, obtaining U and L is a formidable task. In this section, we will discuss approximate balanced truncation of large-sparse dynamical systems, where the approximate low-rank Cholesky factors are used in place of the exact Gramians in computing the reduced-order model. Hence, we will replace the full-rank Cholesky factors U and L with the low-rank ones, namely $\tilde{U}$ and $\tilde{L}$ which are obtained through a k step Smith-type iteration. For details see [GSA03]. For simplicity, let us assume that the original model is SISO. Proceeding similarly to Section 2.2.3, let $\tilde{U}^T\tilde{L} = \tilde{Z}\tilde{\Sigma}\tilde{Y}^T$ be the singular value decomposition (SVD) with $\tilde{\Sigma} = \mathrm{diag}(\tilde{\sigma}_1, \cdots, \tilde{\sigma}_k)$ where $\tilde{\sigma}_i$ are the approximate Hankel singular values with $\tilde{\sigma}_1 > \tilde{\sigma}_2 > \cdots > \tilde{\sigma}_k$. Here we have assumed, for the brevity of the discussion, that the Hankel singular values are distinct. Now define

$$\tilde{W}_1 := \tilde{L}\tilde{Y}_1\tilde{\Sigma}_1^{-1/2} \quad \text{and} \quad \tilde{V}_1 := \tilde{U}\tilde{Z}_1\tilde{\Sigma}_1^{-1/2},$$

where $\tilde{Z}_1$ and $\tilde{Y}_1$ are composed of the leading r columns of $\tilde{Z}$ and $\tilde{Y}$ respectively, and $\tilde{\Sigma}_1 = \mathrm{diag}(\tilde{\sigma}_1, \cdots, \tilde{\sigma}_r)$. We note that the equality $\tilde{W}_1^T\tilde{V}_1 = I_r$ still holds and hence that $\tilde{V}_1\tilde{W}_1^T$ is an oblique projection. The approximately balanced reduced model $\tilde{\mathbf{\Sigma}}_r$ of order r is obtained as

$$\tilde{A}_r = \tilde{W}_1^T A\tilde{V}_r, \quad \tilde{B}_r = \tilde{W}_1^T B, \quad C_r = C\tilde{V}_1, \quad \text{and} \quad \tilde{D}_r = D.$$

To examine the stability of this reduced model, we first define the error term in $\mathcal{P}$. Define Δ as

$$\Delta := \tilde{U}\tilde{U}^T - UU^T = \tilde{\mathcal{P}} - \mathcal{P}.$$

Then one can show that

$$\tilde{A}_r\tilde{\Sigma}_1 + \tilde{\Sigma}_1\tilde{A}_r^T + \tilde{B}\tilde{B}_r{}^T = \tilde{W}_1^T(A\Delta + \Delta A^T)\tilde{W}_1 \tag{2.52}$$

We know that $\tilde{\Sigma}_1 > 0$. Hence to apply Lyapunov's inertia theorem, we need

$$\tilde{B}\tilde{B}_r{}^T - \tilde{W}_1^T(A\Delta + \Delta A^T)\tilde{W}_1 = \tilde{W}_1^T(BB^T - A\Delta - \Delta A^T)\tilde{W}_1 \geq 0. \tag{2.53}$$

Unfortunately, this is not always satisfied, and therefore *one cannot guarantee the stability of the reduced system.* However, we would like to note many researchers have observed that this does not seem to be a difficulty in practice; in most cases approximate balanced truncation via a Smith-type iteration yields a stable reduced system and instability is not an issue; see, for example, [GSA03], [GA01], [PEN99], [LW01], [LW99] and the references there in.

Let $\mathbf{\Sigma}_r = \left[\begin{array}{c|c} A_r & B_r \\ \hline C_r & D \end{array}\right]$ and $\tilde{\mathbf{\Sigma}}_r = \left[\begin{array}{c|c} \tilde{A}_r & \tilde{B}_r \\ \hline \tilde{C}_r & D \end{array}\right]$ be the r^{th} order reduced systems obtained by exact and approximate balancing, respectively. Now we examine

closeness of $\mathbf{\Sigma}_r$ to $\tilde{\mathbf{\Sigma}}_r$. Define $\Delta_V := V_1 - \tilde{V}_1$ and $\Delta_W := W_1 - \tilde{W}_1$, and let $\|\Delta_V\| \leq \tau$ and $\|\Delta_W\| \leq \tau$ where τ is a small number; in other words, we assume that $\tilde{V}_1$ and $\tilde{W}_1$ are close to V_1 and W_1, respectively. Under certain assumptions (see [GSA03]), one can show that

$$\|\mathbf{\Sigma}_r - \tilde{\mathbf{\Sigma}}_r\|_\infty \leq \tau \left(\|C_r\| \, \|B_r\| \, \|A_r\| \, \left(\|W_1\| + \|V_1\| \right) + \|\mathbf{\Sigma}_1\|_\infty \|B_r\| + \|\mathbf{\Sigma}_2\|_\infty \|C_r\| \right) + \mathcal{O}(\tau^2) \quad (2.54)$$

where $\mathbf{\Sigma}_1 := \left[\begin{array}{c|c} A_r & I \\ \hline C_r & \end{array}\right]$ and $\mathbf{\Sigma}_2 := \left[\begin{array}{c|c} A_r & B_r \\ \hline I & \end{array}\right]$. Hence for small τ, i.e., when $\tilde{V}_1$ and $\tilde{W}_1$ are, respectively, close to V_1 and W_1, we expect $\mathbf{\Sigma}_r$ to be close to $\tilde{\mathbf{\Sigma}}_r$. Indeed as the examples in Section 2.7 show, $\tilde{\mathbf{\Sigma}}_r$ behaves much better than the above upper bound predicts and $\tilde{\mathbf{\Sigma}}_r$, the approximately balanced system using low-rank Gramians, is almost the same as the exactly balanced system. These observations reveal the effectiveness of the Smith-type methods for balanced truncation of large-sparse dynamical systems.

2.7 Numerical Examples

In this section we give numerical results on the CF–ADI method as well as the LR-Smith(l) and Modified LR-Smith(l) methods.

2.7.1 CF–ADI and the Spiral Inductor

We begin with the CF–ADI approximation to the Lyapunov equation

$$A\mathcal{X} + \mathcal{X}A^T + BB^T = 0.$$

The example in Figure 2.1 comes from the inductance extraction of an on-chip planar square spiral inductor suspended over a copper plane [KWW98], shown in Figure 1(a). (See Chapter 23 for a detailed description of the spiral inductor.) The original order 500 system has been symmetrized according to [SKEW96]. The matrix A is a symmetric 500×500 matrix, and the input coefficient matrix $B \in \mathbb{R}^n$ has one column.

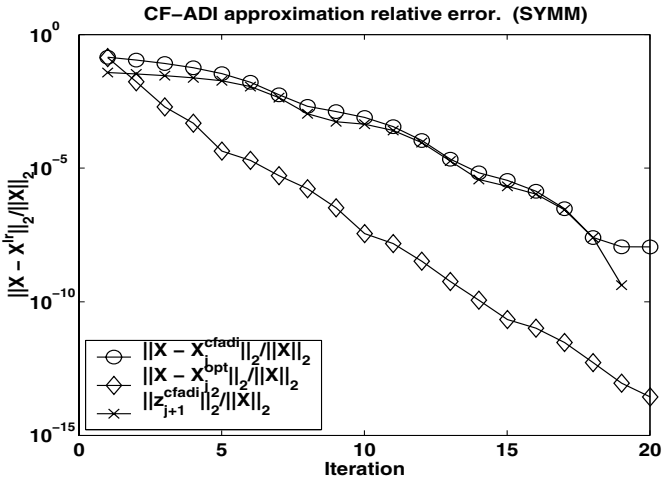
Because A is symmetric, the eigenvalues of A are real and good CF–ADI parameters are easy to find. The procedure given in Section 2.4.4 was followed. CF–ADI was run to convergence in this example, which took 20 iterations.

Figure 1(b) shows the relative 2-norm error of the CF–ADI approximation, i.e.

$$\frac{\|\mathcal{X} - \mathcal{X}_j^{cfadi}\|_2}{\|\mathcal{X}\|_2},$$

where $\mathcal{X}$ is the exact solution to $A\mathcal{X} + \mathcal{X}A^T + BB^T = 0$ and $\mathcal{X}_j^{cfadi}$ is the jth CF–ADI approximation, for $j = 1, \cdots, 20$. To illustrate the quality of

(a) Spiral inductor



(b) CF–ADI approximation

Fig. 2.1. Spiral inductor, a symmetric system.

the low-rank approximation, we compare it with the optimal 2-norm rank-j approximation to X[GVL96], denoted X_j^{opt}, obtained from the singular value decomposition of exact solution X. At $j = 20$, the relative error of the CF–ADI approximation has reached 10^{-8}, which is about the same size as the error of the optimal rank 11 approximation. The error estimate $\|z_{j+1}^{CFA}\|_2^2$ approximates the actual error $\|\mathcal{X} - \mathcal{X}_j^{cfadi}\|$ closely for all j.

2.7.2 LR-Smith(l) and Modified LR-Smith(l) Methods

In this section we apply LR-Smith(l) and Modified LR-Smith(l) methods to two dynamical systems. In each example, both the LR-Smith(l) iterates $\mathcal{P}_k^{Sl}$, and $\mathcal{Q}_k^{Sl}$; and the modified LR-Smith(l) iterates $\tilde{\mathcal{P}}_k$, and $\tilde{\mathcal{Q}}_k$ are computed. Also balanced reduction is applied using the full rank Gramians $\mathcal{P}$, $\mathcal{Q}$ and the approximate Gramians $\mathcal{P}_k^{Sl}$, $\mathcal{Q}_k^{Sl}$; and $\tilde{\mathcal{P}}_k$, $\tilde{\mathcal{Q}}_k$. The resulting reduced order systems are compared.

CD Player Model

This example is described in Chapter 24, Section 4, this volume. The full order model (FOM) describes the dynamics of a portable CD player, is of order 120, and single-input single-output. The eigenvalues of A are scattered in the complex plane with relatively large imaginary parts. This makes it harder to obtain a low $\rho(A_d)$. A single shift results in $\rho(A_d) = 0.99985$. Indeed, even with a high number of multiple shifts, $l = 40$, $\rho(A_d)$ could not be reduced to less than 0.98. Hence only a single shift is considered. This observation agrees with the discussion in Section 2.5 that when the eigenvalues of A are scattered in the complex plane, ADI-type iterations converge slowly. LR-Smith(l) and the modified LR-Smith(l) iterations are run for $k = 70$ iterations. For the Modified Smith(l) iteration, the tolerance values are chosen to be

$$\tau_{\mathcal{P}} = 1 \times 10^{-6} \qquad \text{and} \qquad \tau_{\mathcal{Q}} = 8 \times 10^{-6}.$$

The low-rank LR-Smith(l) yields Cholesky factors Z_k^{Sl} and Y_k^{Sl} with 70 columns. On the other hand, the modified LR-Smith(l) yields low-rank Cholesky factors $\tilde{Z}_k$ and $\tilde{Y}_k$ with only 25 columns. To check the closeness of modified Smith iterates to the exact Smith iterates, we compute the following relative error norms:

$$\frac{\|\mathcal{P}_k^{Sl} - \tilde{\mathcal{P}}_k\|}{\|\mathcal{P}_k^{Sl}\|} = 4.13 \times 10^{-10}, \text{ and} \qquad \frac{\|\mathcal{Q}_k^{Sl} - \tilde{\mathcal{Q}}_k\|}{\|\mathcal{Q}_k^{Sl}\|} = 2.33 \times 10^{-10}.$$

Although the number of columns of the Cholesky factor have been reduced from 70 to 25, the Modified Smith method yields almost the same accuracy. We also look at the error between the exact and approximate Gramians:

$$\frac{\|\mathcal{P} - \mathcal{P}_k^{Sl}\|}{\|\mathcal{P}\|} = \frac{\|\mathcal{P} - \tilde{\mathcal{P}}_k\|}{\|\mathcal{P}\|} = 3.95 \times 10^{-3},$$

$$\frac{\|\mathcal{Q} - \mathcal{Q}_k^{Sl}\|}{\|\mathcal{Q}\|} = \frac{\|\mathcal{Q} - \tilde{\mathcal{Q}}_k\|}{\|\mathcal{Q}\|} = 8.24 \times 10^{-1}.$$

Next, we reduce the order of the FOM to $r = 12$ by balanced truncation using both the approximate and the exact solutions. $\mathbf{\Sigma}_k$, $\mathbf{\Sigma}_k^{Sl}$ and $\tilde{\mathbf{\Sigma}}_k$ denote

the 12^{th} order reduced systems obtained through balanced reduction using the exact Cholesky factors Z and Y; the LR-Smith(l) iterates Z_k^{Sl} and Y_k^{Sl}; and the modified LR-Smith(l) iterates $\tilde{Z}_k$ and $\tilde{Y}_k$ respectively. Also $\mathbf{\Sigma}$ denotes the FOM.

Figure 2.2 depicts the amplitude Bode plots of the FOM $\mathbf{\Sigma}$ and the reduced balanced systems $\mathbf{\Sigma}_k$, $\mathbf{\Sigma}_k^{Sl}$ and $\tilde{\mathbf{\Sigma}}_k$. As can be seen, although relative error between the exact and the approximate Gramians are not very small, $\mathbf{\Sigma}_k^{Sl}$ and $\tilde{\mathbf{\Sigma}}_k$ show a very similar behavior to $\mathbf{\Sigma}_k$. This observation reveals that even if the relative error in the approximate Gramians are big, if the dominant eigenspace of $\mathcal{PQ}$, and hence the largest HSV are matched well, approximate balanced truncation performs very closely to the exact balanced truncation. Similar observations can be found in [GA01, GUG03]. The amplitude Bode plots of the error systems $\mathbf{\Sigma} - \mathbf{\Sigma}_k$, $\mathbf{\Sigma} - \mathbf{\Sigma}_k^{Sl}$ and $\mathbf{\Sigma} - \tilde{\mathbf{\Sigma}}_k$ are illustrated in Figure 2.3. It is also important to note that since the errors between $\tilde{\mathcal{P}}_k$ and $\mathcal{P}_k^{Sl}$, and $\tilde{\mathcal{Q}}_k$ and $\mathcal{Q}_k^{Sl}$ are small, $\mathbf{\Sigma}_k^{Sl}$ and $\tilde{\mathbf{\Sigma}}_k$ are almost equal as expected. The relative $\mathcal{H}_\infty$ norms of the error systems are tabulated in Table 2.1.

Table 2.1. Numerical Results for CD Player Model

$\|\mathbf{\Sigma} - \mathbf{\Sigma}_k\|_{\mathcal{H}_\infty}$	$\|\mathbf{\Sigma} - \mathbf{\Sigma}_k^{Sl}\|_{\mathcal{H}_\infty}$	$\left\|\mathbf{\Sigma} - \tilde{\mathbf{\Sigma}}_k\right\|_{\mathcal{H}_\infty}$	$\left\|\mathbf{\Sigma}_k^{Sl} - \tilde{\mathbf{\Sigma}}_k\right\|_{\mathcal{H}_\infty}$
9.88×10^{-4}	9.71×10^{-4}	9.69×10^{-4}	5.11×10^{-6}

$\|\mathbf{\Sigma}_k - \mathbf{\Sigma}_k^{Sl}\|_{\mathcal{H}_\infty}$	$\left\|\mathbf{\Sigma}_k - \tilde{\mathbf{\Sigma}}_k\right\|_{\mathcal{H}_\infty}$
1.47×10^{-4}	1.47×10^{-4}

A Random System

This model is from [PEN99] and the example from [GSA03, GUG03]. The FOM is a dynamical system of order 1006. The state-space matrices of the full-order model $\mathbf{\Sigma} = \left[\begin{array}{c|c} A & B \\ \hline C & 0 \end{array}\right]$ are given by

$$A = \mathrm{diag}(A_1, A_2, A_3, A_4), \quad B^T = C = [\ \underbrace{10\ \cdots\ 10}_{6}\ \underbrace{1\ \cdots 1}_{1000}]$$

where

$$A_1 = \begin{bmatrix} -1 & 100 \\ -100 & -1 \end{bmatrix}, \quad A_2 = \begin{bmatrix} -1 & 200 \\ -200 & -1 \end{bmatrix}, \quad A_3 = \begin{bmatrix} -1 & 400 \\ -400 & -1 \end{bmatrix},$$
$$\text{and} \quad A_4 = \mathrm{diag}(-1, \cdots, -1000). \qquad .$$

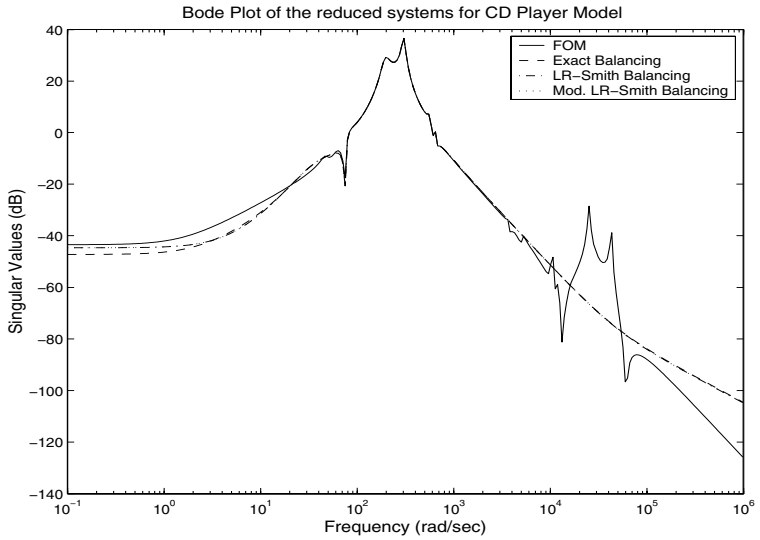


Fig. 2.2. The amplitude Bode plots of the FOM $\mathbf{\Sigma}$ and the reduced systems $\mathbf{\Sigma}_k$ (Exact Balancing), $\mathbf{\Sigma}_k^{Sl}$ (LR-Smith Balancing) and $\tilde{\mathbf{\Sigma}}_k$ (Mod. LR-Smith Balancing) for the CD Player Model

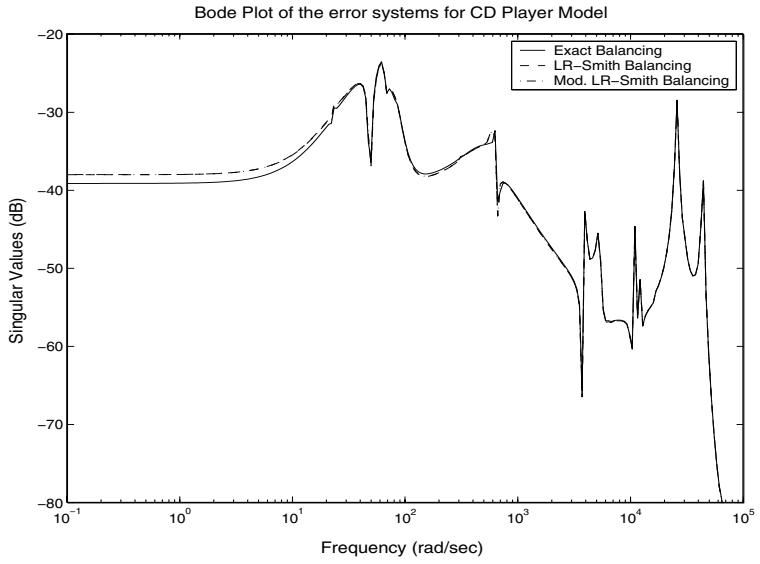


Fig. 2.3. The amplitude Bode plots of error systems $\mathbf{\Sigma} - \mathbf{\Sigma}_k$ (Exact Balancing), $\mathbf{\Sigma} - \mathbf{\Sigma}_k^{Sl}$ (LR-Smith Balancing) and $\mathbf{\Sigma} - \tilde{\mathbf{\Sigma}}_k$ (Mod. LR-Smith Balancing) for the CD Player Model

The spectrum of A is

$$\sigma(A) = \{-1, -2, \cdots, -1000, -1 \pm 100\jmath, -1 \pm 200\jmath, -1 \pm 400\jmath\}.$$

LR-Smith(l) and modified LR-Smith(l) methods are applied using $l = 10$ cyclic shifts. Six of the shifts are chosen so that the 6 complex eigenvalues of A are eliminated. This shift selection reduces the ADI spectral radius $\rho(A_d)$ to 0.7623, and results in a fast convergence. Once more, the numerical results support the discussion in Section 2.5. Since the eigenvalues are mostly real, with an appropriate choice of shifts, the spectral radius can be easily reduced to a small number yielding a fast convergence. Both LR-Smith(l) and the modified LR-Smith(l) iterations are run for $k = 30$ iterations with the tolerance values

$$\tau_{\mathcal{P}} = \tau_{\mathcal{Q}} = 3 \times 10^{-5}$$

for the latter. The resulting LR-Smith(l) and modified LR-Smith(l) Cholesky factors has 300 and 19 columns, respectively. Even though the number of columns in the modified method is much less than the exact LR-Smith(l) method, almost there is no lost of accuracy in the computed Gramian as the following numbers show:

$$\frac{\|\mathcal{P}_k^{Sl} - \tilde{\mathcal{P}}_k\|}{\|\mathcal{P}_k^{Sl}\|} = 1.90 \times 10^{-8}, \text{ and } \qquad \frac{\|\mathcal{Q}_k^{Sl} - \tilde{\mathcal{Q}}_k\|}{\|\mathcal{Q}_k^{Sl}\|} = 3.22 \times 10^{-8}.$$

The errors between the exact and computed Gramians are as follows:

$$\frac{\|\mathcal{P} - \mathcal{P}_k^{Sl}\|}{\|\mathcal{P}\|} = 4.98 \times 10^{-10}, \quad \frac{\|\mathcal{P} - \tilde{\mathcal{P}}_k\|}{\|\mathcal{P}\|} = 1.88 \times 10^{-8}$$

$$\frac{\|\mathcal{Q} - \mathcal{Q}_k^{Sl}\|}{\|\mathcal{Q}\|} = 4.98 \times 10^{-10}, \quad \frac{\|\mathcal{Q} - \tilde{\mathcal{Q}}_k\|}{\|\mathcal{Q}\|} = 3.21 \times 10^{-8}.$$

Unlike the CD Player model, since $\rho(A_d)$ is small, the iterations converge fast, and both $\mathcal{P}_k^{Sl}$ and $\tilde{\mathcal{P}}_k$ ($\mathcal{Q}_k^{Sl}$ and $\tilde{\mathcal{Q}}_k$) are very close to the exact Gramian $\mathcal{P}$ (to $\mathcal{Q}$).

We reduce the order of the FOM to $r = 11$ using both exact and approximate balanced truncation. As in the CD Player example, $\mathbf{\Sigma}_k$, $\mathbf{\Sigma}_k^{Sl}$ and $\tilde{\mathbf{\Sigma}}_k$ denote the reduced systems obtained through balanced reduction using the exact Cholesky factors Z and Y; the LR-Smith(l) iterates Z_k^{Sl} and Y_k^{Sl}; and the modified LR-Smith(l) iterates $\tilde{Z}_k$ and $\tilde{Y}_k$ respectively. Figure 2.4 depicts the amplitude Bode plots of the FOM $\mathbf{\Sigma}$ and the reduced systems $\mathbf{\Sigma}_k$, $\mathbf{\Sigma}_k^{Sl}$ and $\tilde{\mathbf{\Sigma}}_k$. As Figure 2.4 illustrates, all the reduced models match the FOM quite well. More importantly, the approximate balanced truncation using the low-rank Gramians yields almost the same result as the exact balanced truncation. These results once more prove the effectiveness of the Smith-type methods. The amplitude Bode plots of the error systems $\mathbf{\Sigma} - \mathbf{\Sigma}_k$, $\mathbf{\Sigma} - \mathbf{\Sigma}_k^{Sl}$ and $\mathbf{\Sigma} - \tilde{\mathbf{\Sigma}}_k$

are illustrated in Figure 2.5 and all the relative $\mathcal{H}_\infty$ norms of the error systems are tabulated in Table 2.2. As in the previous example, $\mathbf{\Sigma}_k^{Sl}$ and $\tilde{\mathbf{\Sigma}}_k$ are almost identical. The relative $\mathcal{H}_\infty$ norm of the error $\mathbf{\Sigma}_k^{Sl} - \tilde{\mathbf{\Sigma}}_k$ is $\mathcal{O}(10^{-9})$. We note that $\mathbf{\Sigma}_k^{Sl}$ has been obtained using a Cholesky factor with 300 columns; on the other hand $\tilde{\mathbf{\Sigma}}_k$ has been obtained using a Cholesky factor with only 19 columns, which proves the effectiveness of the modified Smith's method.

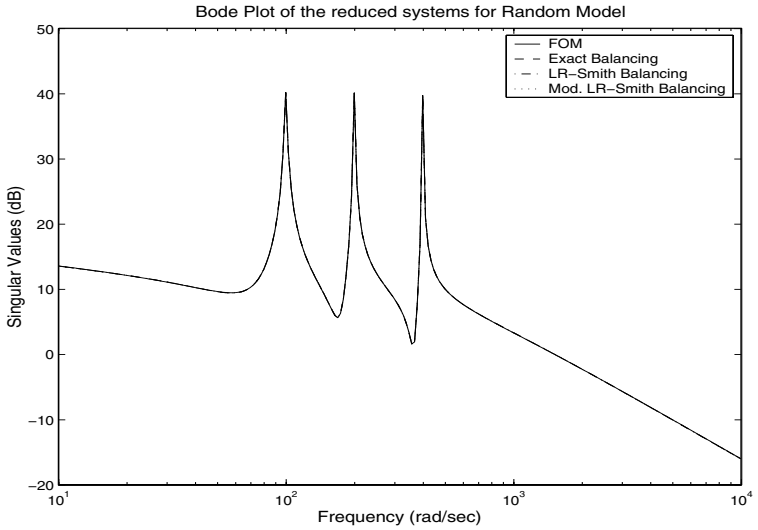


Fig. 2.4. The amplitude Bode plots of the FOM $\mathbf{\Sigma}$ and the reduced systems $\mathbf{\Sigma}_k$ (Exact Balancing), $\mathbf{\Sigma}_k^{Sl}$ (LR-Smith Balancing) and $\tilde{\mathbf{\Sigma}}_k$ (Mod. LR-Smith Balancing) for the Random Model

Table 2.2. Numerical Results for the Random Model

$\|\mathbf{\Sigma} - \mathbf{\Sigma}_k\|_{\mathcal{H}_\infty}$	$\left\|\mathbf{\Sigma} - \mathbf{\Sigma}_k^{Sl}\right\|_{\mathcal{H}_\infty}$	$\left\|\mathbf{\Sigma} - \tilde{\mathbf{\Sigma}}_k\right\|_{\mathcal{H}_\infty}$	$\left\|\mathbf{\Sigma}_k^{Sl} - \tilde{\mathbf{\Sigma}}_k\right\|_{\mathcal{H}_\infty}$
1.47×10^{-4}	1.47×10^{-4}	1.47×10^{-4}	2.40×10^{-9}

$\|\mathbf{\Sigma}_k - \mathbf{\Sigma}_k^{Sl}\|_{\mathcal{H}_\infty}$	$\|\mathbf{\Sigma}_k - \tilde{\mathbf{\Sigma}}_k\|_{\mathcal{H}_\infty}$
7.25×10^{-11}	7.25×10^{-11}

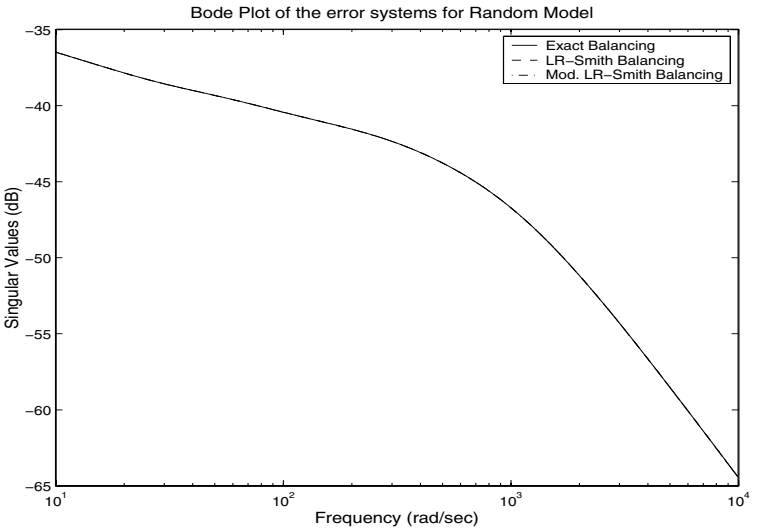


Fig. 2.5. The amplitude Bode plots of error systems $\mathbf{\Sigma} - \mathbf{\Sigma}_k$ (Exact Balancing), $\mathbf{\Sigma} - \mathbf{\Sigma}_k^{Sl}$ (LR-Smith Balancing) and $\mathbf{\Sigma} - \tilde{\mathbf{\Sigma}}_k$ (Mod. LR-Smith Balancing) for the Random Model

2.8 Conclusions

We have reviewed several low-rank methods to solve Lyapunov equations which are based on Smith-type methods, with the goal of facilitating the efficient model reduction of large-scale linear systems. The low-rank methods covered included the Low-Rank ADI method, the Cholesky Factor ADI method, the Low-Rank Smith(l) method, and the modified Low-Rank Smith (l) method. The low-rank factored versions of the ADI method reduced the work required from $\mathcal{O}(n^3)$ to $\mathcal{O}(n)$ for sparse matrices and the required storage from $\mathcal{O}(n^2)$ to $O(nr)$ where r is the numerical rank of the solution. Because these low-rank methods produce the Cholesky factor of the solution to the Lyapunov equation, they are especially well-suited to be used in conjunction with approximate balanced truncation to reduce large-scale linear systems.

References

[ASG01] A. C. Antoulas, D. C. Sorensen, and S. Gugercin. A survey of model reduction methods for large-scale systems. *Contemporary Mathematics, AMS Publications*, **280**, 193–219 (2001).

[ASZ02] A.C. Antoulas, D.C. Sorensen, and Y.K. Zhou. On the decay rate of Hankel singular values and related issues. Systems and Control Letters, **46:5**, 323–342 (2002).

[AS02] A.C. Antoulas and D.C. Sorensen. The Sylvester equation and approximate balanced reduction. Linear Algebra and Its Applications, **351–352**, 671–700 (2002).

[ANT05] A.C. Antoulas. Lectures on the approximation of linear dynamical systems. Advances in Design and Control, SIAM, Philadelphia (2005).

[BS72] R. H. Bartels and G. W. Stewart. Solution of the matrix equation $AX + XA = C$: Algorithm 432. Comm. ACM, **15**, 820–826 (1972).

[BQ99] P. Benner and E. S. Quintana-Ortí. Solving stable generalized Lyapunov equation with the matrix sign function. *Numerical Algorithms*, **20**, 75–100 (1999).

[BQQ01] P. Benner, E. S. Quintana-Ortí, and G. Quintana-Ortí. Efficient Numerical Algorithms for Balanced Stochastic Truncation. International Journal of Applied Mathematics and Computer Science, **11**:5, 1123–1150 (2001).

[CR96] D. Calvetti and L, Reichel. Application of ADI iterative methods to the restoration of noisy images. SIAM J. Matrix Anal. Appl., **17**, 165–186 (1996).

[DP84] U.B. Desai and D. Pal. A transformation approach to stochastic model reduction. IEEE Trans. Automat. Contr., vol **AC-29**, 1097–1100 (1984).

[EW91] N. Ellner and E. Wachspress. Alternating direction implicit iteration for systems with complex spectra. SIAM J. Numer. Anal., **28**, 859–870 (1991).

[ENN84] D. Enns. Model reduction with balanced realizations: An error bound and a frequency weighted generalization. In *Proc.* 23^{rd} *IEEE Conf. Decision and Control* (1984).

[GRE88a] M. Green. A relative error bound for balanced stochastic truncation. IEEE Trans. Automat. Contr., **AC-33**:10, 961–965 (1988).

[GRE88b] M. Green. Balanced stochastic realizations. Journal of Linear Algebra and its Applications, **98**, 211–247 (1988).

[GJ90] W. Gawronski and J.-N. Juang. Model reduction in limited time and frequency intervals. Int. J. Systems Sci., **21**:2, 349–376 (1990).

[GLO84] K. Glover. All Optimal Hankel-norm Approximations of Linear Mutilvariable Systems and their L^{∞}-error Bounds. Int. J. Control, **39**, 1115–1193 (1984).

[GVL96] G. Golub. and C. Van Loan. Matrix computations, 3rd Ed., Johns Hopkins University Press, Baltimore, MD (1996).

[GSA03] S. Gugercin, D.C. Sorensen, and A.C. Antoulas. A modified low-rank Smith method for large-scale Lyapunov equations. Numerical Algorithms, **32**:1, 27–55 (2003).

[GA01] S. Gugercin and A. C. Antoulas. Approximation of the International Space Station 1R and 12A models. In *Proc.* 40^{th} *CDC* (2001).

[GUG03] S. Gugercin. Projection methods for model reduction of large-scale dynamical systems. Ph.D. Dissertation, ECE Dept., Rice University, Houston, TX, USA, May 2003.

[GA04] S. Gugercin and A.C. Antoulas. A survey of model reduction by balaned truncation and some new results. Int. J. Control, **77**:8, 748–766 (2004).

[IT95] M.-P. Istace and J.-P. Thiran. On the third and fourth Zolotarev problems in the complex plane. SIAM J. Numer. Anal., **32**:1, 249–259 (1995).

[HAM82] S. Hammarling. Numerical solution of the stable, non-negative definite Lyapunov equation. IMA J. Numer. Anal., **2**, 303–323 (1982).

[HPT96] A. S. Hodel, K.P. Poola, and B. Tenison. Numerical solution of the Lyapunov equation by approximate power iteration. Linear Algebra Appl., **236**, 205–230 (1996).

[HR92] D. Y. Hu and L. Reichel. Krylov subspace methods for the Sylvester equation. Linear Algebra Appl., **172**, 283–313, (1992).

[JK94] I. M. Jaimoukha and E. M. Kasenally. Krylov subspace methods for solving large Lyapunov equations. SIAM J. Numerical Anal., **31**, 227–251 (1994).

[JK97] I.M. Jaimoukha, E.M. Kasenally. Implicitly restarted Krylov subspace methods for stable partial realizations. SIAM J. Matrix Anal. Appl., **18**, 633–652 (1997).

[KWW98] M. Kamon and F. Wang and J. White. Recent improvements for fast inductance extracton and simulation [packaging]. *Proceedings of the IEEE 7th Topical Meeting on Electrical Performance of Electronic Packaging*, 281–284 (1998).

[LW02] J.-R. Li and J. White. Low rank solution of Lyapunov equations. SIAM J. Matrix Anal. Appl., **24**:1, 260–280 (2002).

[LW99] J.-R. Li and J. White. Efficient model reduction of interconnect via approximate system Gramians. In *Proc. IEEE/ACM Intl. Conf. CAD*, 380–383, San Jose, CA (1999).

[LW01] J.-R. Li and J. White. Reduction of large-circuit models via approximate system Gramians. Int. J. Appl. Math. Comp. Sci., **11**, 1151–1171 (2001).

[LC92] C.-A Lin and T.-Y Chiu. Model reduction via frequency weighted balanced realization. Control Theory and Advanced Technol., **8**, 341–351 (1992).

[LW91] A. Lu and E. Wachspress. Solution of Lyapunov equations by alternating direction implicit iteration. Comput. Math. Appl., **21**:9, 43–58 (1991).

[MOO81] B. C. Moore. Principal Component Analysis in Linear System: Controllability, Observability and Model Reduction. *IEEE Transactions on Automatic Control*, **AC-26**, 17–32 (1981).

[MR76] C. T. Mullis and R. A. Roberts. Synthesis of minimum roundoff noise fixed point digital filters. IEEE Trans. on Circuits and Systems, **CAS-23**, 551–562, (1976).

[OJ88] P.C. Opdenacker and E.A. Jonckheere. A contraction mapping preserving balanced reduction scheme and its infinity norm error bounds. IEEE Trans. Circuits and Systems, (1988).

[PR55] D. W. Peaceman and H. H. Rachford. The numerical solutions of parabolic and elliptic differential equations. J. SIAM, **3**, 28–41 (1955).

[PEN00a] T. Penzl. Eigenvalue Decay Bounds for Solutions of Lyapunov Equations: The Symmetric Case. Systems and Control Letters, **40**: 139–144 (2000).

[PEN00b] T. Penzl. A cyclic low-rank Smith method for large sparse Lyapunov equations. SIAM J. Sci. Comput., **21**;4, 1401–1418 (2000).

[PEN99] T. Penzl. Algorithms for model reduction of large dynamical systems. Technical Report SFB393/99-40, Sonderforschungsbereich 393

Numerische Simulation auf massiv parallelen Rechnern, TU Chemnitz (1999). Avaliable from `http://www.tu-chemnitz.de/sfb393/sfb99pr.html`.

[ROB80] J. D. Roberts. Linear model reduction and solution of the algebraic Riccati equation by use of the sign function. *International Journal of Control,* **32**, 677–687 (1980).

[SAA90] Y. Saad. Numerical solution of large Lyapunov equations. In *Signal Processing, Scattering, Operator Theory and Numerical Methods*, M. Kaashoek, J.V. Schuppen, and A. Ran, eds., Birkhäuser,Boston, MA, 503–511 (1990).

[SKEW96] M. Silveira, M. Kamon, I. Elfadel and J. White. A coordinate-transformed Arnoldi algorithm for generating guaranteed stable reduced-order models of RLC circuits. In *Proc. IEEE/ACM Intl. Conf. CAD, San Jose, CA*, 288–294 (1996).

[SMI68] R. A. Smith. Matrix Equation, $XA + BX = C$. SIAM J. Appl. Math, **16**, 198–201 (1968).

[SAM95] V. Sreeram, B.D.O Anderson and A.G. Madievski. Frequency weighted balanced reduction technique: A generalization and an error bound. In *Proc.* 34^{th} *IEEE Conf. Decision and Control* (1995).

[WSL99] G. Wang, V. Sreeram and W.Q. Liu. A new frequency weighted balanced truncation method and an error bound. IEEE Trans. Automat. Contr., **44**;9, 1734–1737 (1999).

[STA91] G. Starke. Optimal alternating direction implicit parameters for non-symmetric systems of linear equations. SIAM J. Numer. Anal., **28**:5, 1431–1445 (1991).

[STA93] G. Starke. Fejer-Walsh points for rational functions and their use in the ADI iterative method. J. Comput. Appl. Math., **46**, 129–141, (1993).

[VA01] A. Varga and B.D.O Anderson. Accuracy enhancing methods for the frequency-weighted balancing related model reduction. In *Proc.* 40^{th} *IEEE Conf. Decision and Control* (2001).

[WAC62] E. Wachspress. Optimum alternating-direction-implicit iteration parameters for a model problem. J. Soc. Indust. Appl. Math., **10**, 339–350 (1962).

[WAC88a] E. Wachspress. Iterative solution of the Lyapunov matrix equation. Appl. Math. Lett., **1**, 87–90 (1988).

[WAC88b] E. Wachspress. The ADI minimax problem for complete spectra. Appl. Math. Lett., **1**, 311–314 (1988).

[WAC90] E. Wachspress. The ADI minimax problem for complex spectra. In *Iterative Methods for Large Linear Systems,* D. Kincaid and L. Hayes, edss, Academic Press, San Diego, 251–271 (1990).

[WAC95] E. Wachspress. The ADI model problem. *Self published,* Windsor, CA (1995).

[ZHO02] Y, Zhou. Numerical methods for large scale matrix equations with applications in LTI system model reduction. Ph. D. Thesis, CAAM Department, Rice University, Houston, TX, USA, May (2002).

[ZHO95] K. Zhou. Frequency-weighted $\mathcal{L}_\infty$ norm and optimal Hankel norm model reduction. IEEE Trans. Automat. Contr., **40**:10, 1687–1699 (1995).

3

Balanced Truncation Model Reduction for Large-Scale Systems in Descriptor Form

Volker Mehrmann[1] and Tatjana Stykel[2]

[1] Institut für Mathematik, MA 4-5, Technische Universität Berlin, Straße des 17. Juni 136, 10623 Berlin, Germany, mehrmann@math.tu-berlin.de.
[2] Institut für Mathematik, MA 3-3, Technische Universität Berlin, Straße des 17. Juni 136, 10623 Berlin, Germany, stykel@math.tu-berlin.de.

Summary. In this paper we give a survey on balanced truncation model order reduction for linear time-invariant continuous-time systems in descriptor form. We first give a brief overview of the basic concepts from linear system theory and then present balanced truncation model reduction methods for descriptor systems and discuss their algorithmic aspects. The efficiency of these methods is demonstrated by numerical experiments.

3.1 Introduction

We study model order reduction for linear time-invariant continuous-time systems

$$\begin{aligned} E\,\dot{x}(t) &= A\,x(t) + B\,u(t), \quad x(0) = x_0, \\ y(t) &= C\,x(t), \end{aligned} \tag{3.1}$$

where $E, A \in \mathbb{R}^{n,n}$, $B \in \mathbb{R}^{n,m}$, $C \in \mathbb{R}^{p,n}$, $x(t) \in \mathbb{R}^n$ is the state vector, $u(t) \in \mathbb{R}^m$ is the control input, $y(t) \in \mathbb{R}^p$ is the output and $x_0 \in \mathbb{R}^n$ is the initial value. The number of state variables n is called the *order* of system (3.1). If $I = E$, then (3.1) is a *standard state space system*. Otherwise, (3.1) is a *descriptor system* or *generalized state space system*. Such systems arise in a variety of applications including multibody dynamics with constrains, electrical circuit simulation and semidiscretization of partial differential equations, see [Ber90, BCP89, Cam80, Dai89, GF99, Sch95].

Modeling of complex physical and technical processes such as fluid flow, very large system integrated (VLSI) chip design or mechanical systems simulation, leads to descriptor systems of very large order n, while the number m of inputs and the number p of outputs are typically small compared to n. Despite the ever increasing computational speed, simulation, optimization or real time controller design for such large-scale systems is difficult because of storage requirements and expensive computations. In this case *model order*

reduction plays an important role. It consists in approximating the descriptor system (3.1) by a reduced-order system

$$\begin{aligned} \widetilde{E}\,\dot{\widetilde{x}}(t) &= \widetilde{A}\,\widetilde{x}(t) + \widetilde{B}\,u(t), \quad \widetilde{x}(0) = \widetilde{x}_0, \\ \widetilde{y}(t) &= \widetilde{C}\,\widetilde{x}(t), \end{aligned} \tag{3.2}$$

where $\widetilde{E}$, $\widetilde{A} \in \mathbb{R}^{\ell,\ell}$, $\widetilde{B} \in \mathbb{R}^{\ell,m}$, $\widetilde{C} \in \mathbb{R}^{p,\ell}$ and $\ell \ll n$. Note that systems (3.1) and (3.2) have the same input $u(t)$. We require the approximate model (3.2) to preserve properties of the original system (3.1) like regularity, stability and passivity. It is also desirable for the approximation error to be small. Moreover, the computation of the reduced-order system should be numerically reliable and efficient.

There exist various model reduction approaches for standard state space systems such as balanced truncation [LHPW87, Moo81, SC89, TP84, Var87], moment matching approximation [Bai02, FF95, Fre00, GGV94], singular perturbation approximation [LA89] and optimal Hankel norm approximation [Glo84]. Surveys on standard state space system approximation and model reduction can be found in [Ant04, ASG01, FNG92], see also Chapters 1 and 9 in this book.

A popular model reduction technique for large-scale standard state space systems is *moment matching approximation* considered first in [FF95, GGV94]. This approach consists in projecting the dynamical system onto Krylov subspaces computed by an Arnoldi or Lanczos process. Krylov subspace methods are attractive for large-scale sparse systems, since only matrix-vector multiplications are required, and they can easily be generalized for descriptor systems, e.g., [BF01, Fre00, GGV96, Gri97]. Drawbacks of this technique are that stability and passivity are not necessarily preserved in the reduced-order system and that there is no global approximation error bound, see [Bai02, BF01, BSSY99, Bea04, Gug03] for recent contributions on this topic.

Balanced truncation [LHPW87, Moo81, SC89, TP84, Var87] is another well studied model reduction approach for standard state space systems. The method makes use of the two Lyapunov equations

$$A\mathcal{P} + \mathcal{P}A^T = -BB^T, \qquad A^T\mathcal{Q} + \mathcal{Q}A = -C^TC.$$

The solutions $\mathcal{P}$ and $\mathcal{Q}$ of these equations are called the *controllability* and *observability Gramians*, respectively. The balanced truncation method consists in transforming the state space system into a balanced form whose controllability and observability Gramians become diagonal and equal, together with a truncation of those states that are both difficult to reach and to observe [Moo81]. An important property of this method is that the asymptotic stability is preserved in the reduced-order system. Moreover, the existence of a priory error bounds [Enn84, Glo84] allows an adaptive choice of the state space dimension ℓ of the reduced model depending on how accurate the

approximation is needed. A difficulty in balanced truncation model reduction for large-scale problems is that two matrix Lyapunov equations have to be solved. However, recent results on low rank approximations to the solutions of Lyapunov equations [ASG03, Gra04, LW02, Pen99a, Pen00b] make the balanced truncation model reduction approach attractive for large-scale systems, see [Li00, LWW99, Pen99b]. The extension of balanced truncation model reduction to descriptor systems has only recently been considered in [LS00, PS94, Sty04a, Sty04b].

In this paper we briefly review some basic linear system concepts including fundamental solution matrix, transfer function, realizations, controllability and observability Gramians, Hankel operators as well as Hankel singular values that play a key role in balanced truncation. We also present generalizations of balanced truncation model reduction methods for descriptor systems and discuss their numerical aspects.

Throughout the paper we will denote by $\mathbb{R}^{n,m}$ the space of $n \times m$ real matrices. The complex plane is denoted by $\mathbb{C}$, the open left half-plane is denoted by $\mathbb{C}^-$, and $\mathrm{i}\mathbb{R}$ is the imaginary axis. Furthermore, $\mathbb{R}^- = (-\infty, 0)$ and $\mathbb{R}_0^+ = [0, \infty)$. The matrix A^T stands for the transpose of $A \in \mathbb{R}^{n,m}$ and $A^{-T} = (A^{-1})^T$. We will denote by $\mathrm{rank}(A)$ the rank, by $\mathrm{Im}\,(A)$ the image and by $\mathrm{Ker}\,(A)$ the null space of a matrix A. An identity matrix of order n is denoted by I_n. We will use $\mathbb{L}_2^m(\mathbb{I})$ to denote the Hilbert space of vector-valued functions of dimension m whose elements are quadratically integrable on $\mathbb{I}$, where $\mathbb{I} \subseteq \mathbb{R}$ or $\mathbb{I} = \mathrm{i}\mathbb{R}$.

3.2 Descriptor Systems

In this section we give a brief overview of linear system concepts and discuss the main differences between standard state space systems and systems in descriptor form.

Consider the continuous-time descriptor system (3.1). Assume that the pencil $\lambda E - A$ is *regular*, i.e., $\det(\lambda E - A) \neq 0$ for some $\lambda \in \mathbb{C}$. In this case $\lambda E - A$ can be reduced to the *Weierstrass canonical form* [SS90]. There exist nonsingular matrices W and T such that

$$E = W \begin{bmatrix} I_{n_f} & 0 \\ 0 & N \end{bmatrix} T \qquad \text{and} \qquad A = W \begin{bmatrix} J & 0 \\ 0 & I_{n_\infty} \end{bmatrix} T, \tag{3.3}$$

where J and N are matrices in Jordan canonical form and N is nilpotent with index of nilpotency ν. The numbers n_f and n_∞ are the dimensions of the deflating subspaces of $\lambda E - A$ corresponding to the finite and infinite eigenvalues, respectively, and ν is the *index* of the pencil $\lambda E - A$ and also the

index of the descriptor system (3.1). The matrices

$$P_r = T^{-1}\begin{bmatrix} I_{n_f} & 0 \\ 0 & 0 \end{bmatrix} T \qquad \text{and} \qquad P_l = W\begin{bmatrix} I_{n_f} & 0 \\ 0 & 0 \end{bmatrix} W^{-1} \tag{3.4}$$

are the *spectral projections* onto the right and left deflating subspaces of the pencil $\lambda E - A$ corresponding to the finite eigenvalues.

Using the Weierstrass canonical form (3.3), we obtain the following Laurent expansion at infinity for the *generalized resolvent*

$$(\lambda E - A)^{-1} = \sum_{k=-\infty}^{\infty} F_k \lambda^{-k-1}, \tag{3.5}$$

where the coefficients F_k have the form

$$F_k = \begin{cases} T^{-1}\begin{bmatrix} J^k & 0 \\ 0 & 0 \end{bmatrix} W^{-1}, & k = 0, 1, 2 \ldots, \\ T^{-1}\begin{bmatrix} 0 & 0 \\ 0 & -N^{-k-1} \end{bmatrix} W^{-1}, & k = -1, -2, \ldots . \end{cases} \tag{3.6}$$

Let the matrices

$$W^{-1}B = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} \qquad \text{and} \qquad CT^{-1} = [\, C_1, \;\; C_2 \,]$$

be partitioned in blocks conformally to E and A in (3.3). Under the coordinate transformation $Tx(t) = \left[\, z_1^T(t), \;\; z_2^T(t) \,\right]^T$, system (3.1) is decoupled in the *slow* subsystem

$$\dot{z}_1(t) = Jz_1(t) + B_1 u(t), \quad z_1(0) = z_1^0, \tag{3.7}$$

and the *fast* subsystem

$$N\dot{z}_2(t) = z_2(t) + B_2 u(t), \quad z_2(0) = z_2^0, \tag{3.8}$$

with $y(t) = C_1 z_1(t) + C_2 z_2(t)$ and $Tx_0 = \left[\, (z_1^0)^T, \;\; (z_2^0)^T \,\right]^T$.

Equation (3.7) has a unique solution for any integrable input $u(t)$ and any given initial value $z_1^0 \in \mathbb{R}^{n_f}$, see [Kai80]. This solution has the form

$$z_1(t) = e^{tJ} z_1^0 + \int_0^t e^{(t-\tau)J} B_1 u(\tau)\, d\tau.$$

The unique solution of (3.8) is given by

$$z_2(t) = -\sum_{k=0}^{\nu-1} N^k B_2 u^{(k)}(t). \tag{3.9}$$

We see from (3.9) that for the existence of a classical smooth solution $z_2(t)$, it is necessary that the input function $u(t)$ is sufficiently smooth and the initial

value z_2^0 satisfies

$$z_2^0 = -\sum_{k=0}^{\nu-1} N^k B_2 u^{(k)}(0).$$

Therefore, unlike for standard state space systems, the initial value x_0 of the descriptor system (3.1) has to be *consistent*, i.e., it must satisfy the condition

$$(I - P_r)x_0 = \sum_{k=0}^{\nu-1} F_{-k-1} B u^{(k)}(0),$$

where P_r is the spectral projector as in (3.4) and the matrices F_k are given in (3.6).

Thus, if the pencil $\lambda E - A$ is regular, $u(t)$ is ν times continuously differentiable and the initial value x_0 is consistent, then system (3.1) has a unique, continuously differentiable solution $x(t)$ given by

$$x(t) = \mathcal{F}(t)Ex_0 + \int_0^t \mathcal{F}(t-\tau)Bu(\tau)\,d\tau + \sum_{k=0}^{\nu-1} F_{-k-1} B u^{(k)}(t),$$

where

$$\mathcal{F}(t) = T^{-1} \begin{bmatrix} \mathrm{e}^{tJ} & 0 \\ 0 & 0 \end{bmatrix} W^{-1} \tag{3.10}$$

is a *fundamental solution matrix* of system (3.1).

If the initial condition x_0 is inconsistent or the input $u(t)$ is not sufficiently smooth, then the solution of the descriptor system (3.1) may have impulsive modes [Cob84, Dai89].

3.2.1 The Transfer Function

Consider the *Laplace transform* of a function $f(t)$, $t \in \mathbb{R}$, given by

$$\mathbf{f}(s) = \mathfrak{L}[f(t)] = \int_0^\infty e^{-st} f(t)\,dt, \tag{3.11}$$

where s is a complex variable called *frequency*. A discussion of the convergence region of the integral (3.11) in the complex plane and properties of the Laplace transform may be found in [Doe71]. Applying the Laplace transform to (3.1) and taking into account that $\mathfrak{L}[\dot{x}(t)] = s\mathbf{x}(s) - x(0)$, we have

$$\mathbf{y}(s) = C(sE - A)^{-1}B\mathbf{u}(s) + C(sE - A)^{-1}Ex(0), \tag{3.12}$$

where $\mathbf{x}(s)$, $\mathbf{u}(s)$ and $\mathbf{y}(s)$ are the Laplace transforms of $x(t)$, $u(t)$ and $y(t)$, respectively. The rational matrix-valued function $\mathbf{G}(s) = C(sE - A)^{-1}B$ is called the *transfer function* of the continuous-time descriptor system (3.1). Equation (3.12) shows that if $Ex(0) = 0$, then $\mathbf{G}(s)$ gives the relation between

the Laplace transforms of the input $u(t)$ and the output $y(t)$. In other words, $\mathbf{G}(s)$ describes the input-output behavior of (3.1) in the frequency domain.

A *frequency response* of the descriptor system (3.1) is given by $\mathbf{G}(\mathrm{i}\omega)$, i.e., the values of the transfer function on the imaginary axis. For an input function $u(t) = \mathrm{e}^{\mathrm{i}\omega t}u_0$ with $\omega \in \mathbb{R}$ and $u_0 \in \mathbb{R}^m$, we get from (3.1) that

$$y(t) = \mathbf{G}(\mathrm{i}\omega)\mathrm{e}^{\mathrm{i}\omega t}u_0.$$

Thus, the frequency response $\mathbf{G}(\mathrm{i}\omega)$ gives a transfer relation from the periodic input $u(t) = \mathrm{e}^{\mathrm{i}\omega t}u_0$ into the output $y(t)$.

Definition 3.2.1. *The transfer function* $\mathbf{G}(s)$ *is* proper *if* $\lim\limits_{s\to\infty} \mathbf{G}(s) < \infty$, *and* improper *otherwise. If* $\lim\limits_{s\to\infty} \mathbf{G}(s) = 0$, *then* $\mathbf{G}(s)$ *is called* strictly proper.

Using the generalized resolvent equation (3.5), the transfer function $\mathbf{G}(s)$ can be expanded into a Laurent series at $s = \infty$ as

$$\mathbf{G}(s) = \sum_{k=-\infty}^{\infty} CF_{k-1}Bs^{-k},$$

where $CF_{k-1}B$ are the *Markov parameters* of (3.1). Note that $CF_{k-1}B = 0$ for $k \le -\nu$, where ν is the index of the pencil $\lambda E - A$. One can see that the transfer function $\mathbf{G}(s)$ can be additively decomposed as $\mathbf{G}(s) = \mathbf{G}_{sp}(s)+\mathbf{P}(s)$, where

$$\mathbf{G}_{sp}(s) = \sum_{k=1}^{\infty} CF_{k-1}Bs^{-k} \qquad \text{and} \qquad \mathbf{P}(s) = \sum_{k=-\nu+1}^{0} CF_{k-1}Bs^{-k} \tag{3.13}$$

are, respectively, the *strictly proper part* and the *polynomial part* of $\mathbf{G}(s)$. The transfer function $\mathbf{G}(s)$ is strictly proper if and only if $CF_{k-1}B = 0$ for $k \le 0$. Moreover, $\mathbf{G}(s)$ is proper if and only if $CF_{k-1}B = 0$ for $k < 0$. Obviously, if the pencil $\lambda E - A$ is of index at most one, then $\mathbf{G}(s)$ is proper.

Let $\mathbb{H}_\infty$ be a space of all proper rational transfer functions that are analytic and bounded in the closed right half-plane. The $\mathbb{H}_\infty$-*norm* of $\mathbf{G}(s) \in \mathbb{H}_\infty$ is defined via

$$\|\mathbf{G}\|_{\mathbb{H}_\infty} = \sup_{\mathbf{u}\neq 0} \frac{\|\mathbf{G}\mathbf{u}\|_{\mathbb{L}_2^p(\mathrm{i}\mathbb{R})}}{\|\mathbf{u}\|_{\mathbb{L}_2^m(\mathrm{i}\mathbb{R})}} = \sup_{\omega\in\mathbb{R}} \|\mathbf{G}(i\omega)\|_2,$$

where $\|\cdot\|_2$ denotes the spectral matrix norm. By the Parseval identity [Rud87] we have $\|\mathbf{G}\|_{\mathbb{H}_\infty} = \sup\limits_{u\neq 0} \|y\|_{\mathbb{L}_2^p(\mathbb{R})}/\|u\|_{\mathbb{L}_2^m(\mathbb{R})}$, i.e., the $\mathbb{H}_\infty$-norm of $\mathbf{G}(s)$ gives the ratio of the output energy to the input energy of the descriptor system (3.1).

3.2.2 Controllability and Observability

In contrast to standard state space systems, for descriptor systems, there are several different notions of controllability and observability, see [BBMN99, Cob84, Dai89, YS81] and the references therein. We consider only complete controllability and observability here.

Definition 3.2.2. *The descriptor system* (3.1) *is called* completely controllable (C-controllable) *if*

$$\operatorname{rank}[\,\alpha E - \beta A,\ \ B\,] = n \quad \textit{for all} \quad (\alpha, \beta) \in (\mathbb{C} \times \mathbb{C}) \setminus \{(0,0)\}.$$

C-controllability implies that for any given initial state $x_0 \in \mathbb{R}^n$ and final state $x_f \in \mathbb{R}^n$, there exists a control input $u(t)$ that transfers the system from x_0 to x_f in finite time. This notion follows [BBMN99, YS81] and is consistent with the definition of *controllability* given in [Dai89].

Observability is the dual property of controllability.

Definition 3.2.3. *The descriptor system* (3.1) *is called* completely observable (C-observable) *if*

$$\operatorname{rank}[\,\alpha E^T - \beta A^T,\ \ C^T\,] = n \quad \textit{for all} \quad (\alpha, \beta) \in (\mathbb{C} \times \mathbb{C}) \setminus \{(0,0)\}.$$

C-observability implies that if the output is zero for all solutions of the descriptor system (3.1) with a zero input, then this system has only the trivial solution.

The following theorem gives equivalent conditions for system (3.1) to be C-controllable and C-observable.

Theorem 3.2.4. [YS81] *Consider a descriptor system* (3.1)*, where* $\lambda E - A$ *is regular.*

1. *System* (3.1) *is C-controllable if and only if* $\operatorname{rank}[\,\lambda E - A,\ \ B\,] = n$ *for all finite* $\lambda \in \mathbb{C}$ *and* $\operatorname{rank}[\,E,\ \ B\,] = n$.
2. *System* (3.1) *is C-observable if and only if* $\operatorname{rank}[\,\lambda E^T - A^T,\ \ C^T\,] = n$ *for all finite* $\lambda \in \mathbb{C}$ *and* $\operatorname{rank}[\,E^T,\ \ C^T\,] = n$.

Other equivalent algebraic and geometric characterizations of controllability and observability for descriptor systems can be found in [Cob84, Dai89].

3.2.3 Stability

In this subsection we present some results from [Dai89, Sty02a] on stability for the descriptor system (3.1).

Definition 3.2.5. *The descriptor system* (3.1) *is called* asymptotically stable *if* $\lim\limits_{t\to\infty} x(t) = 0$ *for all solutions* $x(t)$ *of* $E\dot{x}(t) = Ax(t)$.

The following theorem collects equivalent conditions for system (3.1) to be asymptotically stable.

Theorem 3.2.6. [Dai89, Sty02a] *Consider a descriptor system* (3.1) *with a regular pencil* $\lambda E - A$*. The following statements are equivalent.*

1. *System* (3.1) *is asymptotically stable.*

2. All finite eigenvalues of the pencil $\lambda E - A$ lie in the open left half-plane.
3. The projected generalized continuous-time Lyapunov equation

$$E^T X A + A^T X E = -P_r^T Q P_r, \qquad X = P_l^T X P_l$$

has a unique Hermitian, positive semidefinite solution X for every Hermitian, positive definite matrix Q.

In the sequel, the pencil $\lambda E - A$ will be called *c-stable* if it is regular and all the finite eigenvalues of $\lambda E - A$ have negative real part. Note that the infinite eigenvalues of $\lambda E - A$ do not affect the behavior of the homogeneous system at infinity.

3.2.4 Gramians and Hankel Singular Values

Assume that the pencil $\lambda E - A$ is c-stable. Then the integrals

$$\mathcal{G}_{pc} = \int_0^\infty \mathcal{F}(t) B B^T \mathcal{F}^T(t)\, dt \quad \text{and} \quad \mathcal{G}_{po} = \int_0^\infty \mathcal{F}^T(t) C^T C \mathcal{F}(t)\, dt$$

exist, where $\mathcal{F}(t)$ is as in (3.10). The matrix $\mathcal{G}_{pc}$ is called the *proper controllability Gramian* and the matrix $\mathcal{G}_{po}$ is called the *proper observability Gramian* of the continuous-time descriptor system (3.1), see [Ben97, Sty02a]. The *improper controllability Gramian* and the *improper observability Gramian* of the system (3.1) are defined by

$$\mathcal{G}_{ic} = \sum_{k=-\nu}^{-1} F_k B B^T F_k^T \quad \text{and} \quad \mathcal{G}_{io} = \sum_{k=-\nu}^{-1} F_k^T C^T C F_k,$$

respectively. Here the matrices F_k are as in (3.6). If $E = I$, then $\mathcal{G}_{pc}$ and $\mathcal{G}_{po}$ are the usual controllability and observability Gramians for standard state space systems [Glo84]. Using the Parseval identity [Rud87], the Gramians can be rewritten in frequency domain as

$$\begin{aligned}
\mathcal{G}_{pc} &= \frac{1}{2\pi} \int_{-\infty}^{\infty} (\mathrm{i}\omega E - A)^{-1} P_l B B^T P_l^T (-\mathrm{i}\omega E - A)^{-T} d\omega, \\
\mathcal{G}_{po} &= \frac{1}{2\pi} \int_{-\infty}^{\infty} (-\mathrm{i}\omega E - A)^{-T} P_r^T C^T C P_r (\mathrm{i}\omega E - A)^{-1} d\omega, \\
\mathcal{G}_{ic} &= \frac{1}{2\pi} \int_0^{2\pi} (e^{\mathrm{i}\omega} E - A)^{-1} (I - P_l) B B^T (I - P_l)^T (e^{-\mathrm{i}\omega} E - A)^{-T} d\omega, \\
\mathcal{G}_{io} &= \frac{1}{2\pi} \int_0^{2\pi} (e^{-\mathrm{i}\omega} E - A)^{-T} (I - P_r)^T C^T C (I - P_r) (e^{\mathrm{i}\omega} E - A)^{-1} d\omega.
\end{aligned}$$

It has been proven in [Sty02a] that the proper controllability and observability Gramians are the unique symmetric, positive semidefinite solutions of the *projected generalized continuous-time algebraic Lyapunov equations (GCALEs)*

$$E\,\mathcal{G}_{pc}A^T + A\,\mathcal{G}_{pc}E^T = -P_l B B^T P_l^T, \qquad \mathcal{G}_{pc} = P_r \mathcal{G}_{pc} P_r^T, \tag{3.14}$$
$$E^T\mathcal{G}_{po}A + A^T\mathcal{G}_{po}E = -P_r^T C^T C P_r, \qquad \mathcal{G}_{po} = P_l^T \mathcal{G}_{po} P_l. \tag{3.15}$$

Furthermore, the improper controllability and observability Gramians are the unique symmetric, positive semidefinite solutions of the *projected generalized discrete-time algebraic Lyapunov equations (GDALEs)*

$$A\,\mathcal{G}_{ic}A^T - E\,\mathcal{G}_{ic}E^T = (I - P_l)BB^T(I - P_l)^T, \quad P_r\mathcal{G}_{ic}P_r^T = 0, \tag{3.16}$$
$$A^T\mathcal{G}_{io}A - E^T\mathcal{G}_{io}E = (I - P_r)^T C^T C (I - P_r), \quad P_l^T\mathcal{G}_{io}P_l = 0. \tag{3.17}$$

Similarly as in standard state space systems [Glo84], the controllability and observability Gramians can be used to define Hankel singular values for the descriptor system (3.1) that are of great importance in model reduction via balanced truncation.

Consider the matrices $\mathcal{G}_{pc}E^T\mathcal{G}_{po}E$ and $\mathcal{G}_{ic}A^T\mathcal{G}_{io}A$. These matrices play the same role for descriptor systems as the product of the controllability and observability Gramians for standard state space systems [Glo84, ZDG96]. It has been shown in [Sty04b] that all the eigenvalues of $\mathcal{G}_{pc}E^T\mathcal{G}_{po}E$ and $\mathcal{G}_{ic}A^T\mathcal{G}_{io}A$ are real and non-negative. The square roots of the largest n_f eigenvalues of the matrix $\mathcal{G}_{pc}E^T\mathcal{G}_{po}E$, denoted by ς_j, are called the *proper Hankel singular values* of the continuous-time descriptor system (3.1). The square roots of the largest n_∞ eigenvalues of the matrix $\mathcal{G}_{ic}A^T\mathcal{G}_{io}A$, denoted by θ_j, are called the *improper Hankel singular values* of system (3.1). Recall that n_f and n_∞ are the dimensions of the deflating subspaces of the pencil $\lambda E - A$ corresponding to the finite and infinite eigenvalues, respectively.

We will assume that the proper and improper Hankel singular values are ordered decreasingly, i.e., $\varsigma_1 \geq \varsigma_2 \geq \ldots \geq \varsigma_{n_f} \geq 0$ and $\theta_1 \geq \theta_2 \geq \ldots \geq \theta_{n_\infty} \geq 0$. For $E = I$, the proper Hankel singular values are the classical Hankel singular values of standard state space systems [Glo84, Moo81].

Since the proper and improper controllability and observability Gramians are symmetric and positive semidefinite, there exist *Cholesky factorizations*

$$\begin{aligned} \mathcal{G}_{pc} &= R_p R_p^T, & \mathcal{G}_{po} &= L_p L_p^T, \\ \mathcal{G}_{ic} &= R_i R_i^T, & \mathcal{G}_{io} &= L_i L_i^T, \end{aligned} \tag{3.18}$$

where the lower triangular matrices R_p, L_p, R_i, $L_i \in \mathbb{R}^{n,n}$ are Cholesky factors [GV96] of the Gramians. In this case the proper Hankel singular values of system (3.1) can be computed as the n_f largest singular values of the matrix $L_p^T E R_p$, and the improper Hankel singular values of (3.1) are the n_∞ largest singular values of the matrix $L_i^T A R_i$, see [Sty04b].

For the descriptor system (3.1), we consider a *proper Hankel operator* $\mathcal{H}_p$ that transforms the past inputs $u_-(t)$ ($u_-(t) = 0$ for $t \geq 0$) into the present and future outputs $y_+(t)$ ($y_+(t) = 0$ for $t < 0$) through the state $x(0) \in \mathrm{Im}\,(P_r)$, see [Sty03]. This operator is defined via

$$y_+(t) = (\mathcal{H}_p u_-)(t) = \int_{-\infty}^{0} G_{sp}(t-\tau)u_-(\tau)\,d\tau, \qquad t \geq 0, \tag{3.19}$$

where $G_{sp}(t) = C\mathcal{F}(t)B$, $t \geq 0$. If the pencil $\lambda E - A$ is c-stable, then $\mathcal{H}_p$ acts from $\mathbb{L}_2^m(\mathbb{R}^-)$ into $\mathbb{L}_2^p(\mathbb{R}_0^+)$. In this case one can show that $\mathcal{H}_p$ is a Hilbert-Schmidt operator and its non-zero singular values coincide with the non-zero proper Hankel singular values of system (3.1).

Unfortunately, we do not know a physically meaningful improper Hankel operator. We can only show that the non-zero improper Hankel singular values of system (3.1) are the non-zero singular values of the improper Hankel matrix

$$\mathcal{H}_i = \begin{bmatrix} CF_{-1}B & CF_{-2}B & \cdots & CF_{-\nu}B \\ CF_{-2}B & & \iddots & 0 \\ \vdots & \iddots & \iddots & \vdots \\ CF_{-\nu}B & 0 & \cdots & 0 \end{bmatrix}$$

with the Markov parameters $CF_{k-1}B$, see [Sty03].

3.2.5 Realizations

For any rational matrix-valued function $\mathbf{G}(s)$, there exist matrices E, A, B and C such that $\mathbf{G}(s) = C(sE - A)^{-1}B$, see [Dai89]. A descriptor system (3.1) with these matrices is called a *realization* of $\mathbf{G}(s)$. We will also denote a realization of $\mathbf{G}(s)$ by $\mathbf{G} = [\, E,\, A,\, B,\, C\,]$ or by

$$\mathbf{G} = \left[\begin{array}{c|c} sE - A & B \\ \hline C & \end{array}\right].$$

Note that the realization of $\mathbf{G}(s)$ is, in general, not unique [Dai89]. Among different realizations of $\mathbf{G}(s)$ we are interested only in particular realizations that are useful for reduced-order modeling.

Definition 3.2.7. *A realization* $[E, A, B, C]$ *of the transfer function* $\mathbf{G}(s)$ *is called* minimal *if the dimension of the matrices* E *and* A *is as small as possible.*

The following theorem gives necessary and sufficient conditions for a realization of $\mathbf{G}(s)$ to be minimal.

Theorem 3.2.8. [Dai89, Sty04b] *Consider a descriptor system* (3.1)*, where the pencil* $\lambda E - A$ *is c-stable. The following statements are equivalent:*

1. *The realization* $[\, E,\, A,\, B,\, C\,]$ *is minimal.*
2. *The descriptor system* (3.1) *is C-controllable and C-observable.*
3. *The rank conditions* $\operatorname{rank}(\mathcal{G}_{pc}) = \operatorname{rank}(\mathcal{G}_{po}) = \operatorname{rank}(\mathcal{G}_{pc}E^T\mathcal{G}_{po}E) = n_f$ *and* $\operatorname{rank}(\mathcal{G}_{ic}) = \operatorname{rank}(\mathcal{G}_{io}) = \operatorname{rank}(\mathcal{G}_{ic}A^T\mathcal{G}_{io}A) = n_\infty$ *hold.*
4. *The proper and improper Hankel singular values of* (3.1) *are positive.*
5. *The rank conditions* $\operatorname{rank}(\mathcal{H}_p) = n_f$ *and* $\operatorname{rank}(\mathcal{H}_i) = n_\infty$ *hold.*

Remark 3.2.9. So far we have considered only descriptor systems without a feed-through term, i.e., $D = 0$ in the output equation $y(t) = Cx(t) + Du(t)$. However, if we allow for the matrix D to be non-zero, then the condition for the realization of the transfer function $\mathbf{G}(s) = C(sE - A)^{-1}B + D$ to be minimal should be reformulated as follows: the realization $[\,E,\, A,\, B,\, C,\, D\,]$ is minimal if and only if the descriptor system is C-controllable and C-observable, and $A\,\mathrm{Ker}\,(E) \subseteq \mathrm{Im}\,(E)$, see [Sok03, VLK81]. The latter condition implies that the nilpotent matrix N in the Weierstrass canonical form (3.3) does not have any 1×1 Jordan blocks.

Definition 3.2.10. *A realization* $[\,E,\, A,\, B,\, C\,]$ *of the transfer function* $\mathbf{G}(s)$ *is called* balanced *if*

$$\mathcal{G}_{pc} = \mathcal{G}_{po} = \begin{bmatrix} \Sigma & 0 \\ 0 & 0 \end{bmatrix} \qquad \textit{and} \qquad \mathcal{G}_{ic} = \mathcal{G}_{io} = \begin{bmatrix} 0 & 0 \\ 0 & \Theta \end{bmatrix},$$

where $\Sigma = \mathrm{diag}(\varsigma_1, \ldots, \varsigma_{n_f})$ *and* $\Theta = \mathrm{diag}(\theta_1, \ldots, \theta_{n_\infty})$.

For a minimal realization $[\,E, A, B, C\,]$ with a c-stable pencil $\lambda E - A$, it is possible to find nonsingular transformation matrices W_b and T_b such that the transformed realization $[\,W_b^T E T_b,\, W_b^T A T_b,\, W_b^T B,\, C T_b\,]$ is balanced, see [Sty04a]. These matrices are given by

$$\begin{aligned} W_b &= \left[\, L_p U_p \Sigma^{-1/2},\;\; L_i U_i \Theta^{-1/2} \,\right], \\ T_b &= \left[\, R_p V_p \Sigma^{-1/2},\;\; R_i V_i \Theta^{-1/2} \,\right]. \end{aligned} \tag{3.20}$$

Observe that, as for standard state space systems [Glo84, Moo81], the balancing transformation for descriptor systems is not unique. It should also be noted that for the matrices W_b and T_b as in (3.20), we have

$$E_b = W_b^T E T_b = \begin{bmatrix} I_{n_f} & 0 \\ 0 & E_2 \end{bmatrix}, \qquad A_b = W_b^T A T_b = \begin{bmatrix} A_1 & 0 \\ 0 & I_{n_\infty} \end{bmatrix}, \tag{3.21}$$

where the matrix $E_2 = \Theta^{-1/2} U_i^T L_i^T E R_i V_i \Theta^{-1/2}$ is nilpotent and the matrix $A_1 = \Sigma^{-1/2} U_p^T L_p^T A R_p V_p \Sigma^{-1/2}$ is nonsingular. Thus, the pencil $\lambda E_b - A_b$ of a balanced descriptor system is in a form that resembles the Weierstrass canonical form.

3.3 Balanced Truncation

In this section we present a generalization of balanced truncation model reduction to descriptor systems.

Note that computing the balanced realization may be an ill-conditioned problem if the descriptor system (3.1) has small proper or improper Hankel singular values. Moreover, if system (3.1) is not minimal, then it has states

that are uncontrollable or/and unobservable. These states correspond to the zero proper and improper Hankel singular values and can be truncated without changing the input-output relation in the system. Note that the number of non-zero improper Hankel singular values of (3.1) is equal to $\operatorname{rank}(\mathcal{G}_{ic}A^T\mathcal{G}_{io}A)$, which can in turn be bounded by

$$\operatorname{rank}(\mathcal{G}_{ic}A^T\mathcal{G}_{io}A) \le \min(\nu m, \nu p, n_\infty),$$

where ν is the index of the pencil $\lambda E - A$, m is the number of inputs, p is the number of outputs and n_∞ is the dimension of the deflating subspace of $\lambda E - A$ corresponding to the infinite eigenvalues. This estimate shows that if the number of inputs or outputs multiplied by the index ν is much smaller than the dimension n_∞, then the order of system (3.1) can be reduced significantly by the truncation of the states corresponding to the zero improper Hankel singular values.

Furthermore, we have the following theorem that gives an energy interpretation of the proper controllability and observability Gramians.

Theorem 3.3.1. [Sty04b] *Consider a descriptor system* (3.1) *that is asymptotically stable and C-controllable. Let $\mathcal{G}_{pc}$ and $\mathcal{G}_{po}$ be the proper controllability and observability Gramians of* (3.1) *and let*

$$\mathbf{E}_y := \|y\|^2_{\mathbb{L}_2^p(\mathbb{R}_0^+)} = \int_0^\infty y^T(t)y(t)\,dt, \qquad \mathbf{E}_u := \|u\|^2_{\mathbb{L}_2^m(\mathbb{R}^-)} = \int_{-\infty}^0 u^T(t)u(t)\,dt$$

be a future output energy and a past input energy, respectively. If $x_0 \in \operatorname{Im}(P_r)$ and $u(t) = 0$ for $t \ge 0$, then $\mathbf{E}_y = x_0^T E^T \mathcal{G}_{po} E\, x_0$. Moreover, for $u_{\min}(t) = B^T\mathcal{F}^T(-t)\mathcal{G}_{pc}^- x_0$, we have

$$\mathbf{E}_{u_{\min}} = \min_{u\in\mathbb{L}_2^m(\mathbb{R}^-)} \mathbf{E}_u = x_0^T\mathcal{G}_{pc}^- x_0,$$

where the matrix $\mathcal{G}_{pc}^-$ is a solution of the three matrix equations

$$\mathcal{G}_{pc}\mathcal{G}_{pc}^-\mathcal{G}_{pc} = \mathcal{G}_{pc}, \qquad \mathcal{G}_{pc}^-\mathcal{G}_{pc}\mathcal{G}_{pc}^- = \mathcal{G}_{pc}^-, \qquad (\mathcal{G}_{pc}^-)^T = \mathcal{G}_{pc}^-.$$

Theorem 3.3.1 implies that a large past input energy $\mathbf{E}_u$ is required to reach the state $x(0) = P_r x_0$ which lies in an invariant subspace of $\mathcal{G}_{pc}$ corresponding to its small non-zero eigenvalues from the state $x(-\infty) = 0$. Moreover, if x_0 is contained in an invariant subspace of the matrix $E^T\mathcal{G}_{po}E$ corresponding to its small non-zero eigenvalues, then the initial state $x(0) = x_0$ has a small effect on the future output energy $\mathbf{E}_y$. For the balanced system, we have

$$\mathcal{G}_{pc} = E^T\mathcal{G}_{po}E = \begin{bmatrix} \Sigma & 0 \\ 0 & 0 \end{bmatrix}.$$

In this case the states related to the small proper Hankel singular values are difficult to reach and to observe at the same time. The truncation of these states essentially does not change the system properties.

Unfortunately, this does not hold for the improper Hankel singular values. If we truncate the states that correspond to the small non-zero improper Hankel singular values, then the pencil of the reduced-order system may get finite eigenvalues in the closed right half-plane, see [LS00]. In this case the approximation may be inaccurate.

Remark 3.3.2. The equations associated with the improper Hankel singular values describe constraints of the system, i.e., they define a manifold in which the solution dynamics takes place. For this reason, a truncation of these equations corresponds to ignoring constraints and, hence, physically meaningless results may be expected.

Note that to perform order reduction we do not need to transform the descriptor system into a balanced form explicitly. It is sufficient to determine the subspaces associated with dominant proper and non-zero improper Hankel singular values and project the descriptor system on these subspaces. To compute a reduced-order system we can use the following algorithm which is a generalization of the *square root balanced truncation method* [LHPW87, TP84] to the descriptor system (3.1).

Algorithm 3.3.1. Generalized Square Root (GSR) method.

INPUT: A realization $\mathbf{G} = [\, E,\, A,\, B,\, C\,]$ such that $\lambda E - A$ is c-stable.
OUTPUT: A reduced-order system $\widetilde{\mathbf{G}} = [\, \widetilde{E},\, \widetilde{A},\, \widetilde{B},\, \widetilde{C}\,]$.

1. Compute the Cholesky factors R_p and L_p of the proper Gramians $\mathcal{G}_{pc} = R_p R_p^T$ and $\mathcal{G}_{po} = L_p L_p^T$ that satisfy (3.14) and (3.15), respectively.
2. Compute the Cholesky factors R_i and L_i of the improper Gramians $\mathcal{G}_{ic} = R_i R_i^T$ and $\mathcal{G}_{io} = L_i L_i^T$ that satisfy (3.16) and (3.17), respectively.
3. Compute the skinny singular value decomposition

$$L_p^T E R_p = [\, U_1,\, U_2\,] \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} [\, V_1,\, V_2\,]^T, \tag{3.22}$$

   where the matrices $[\, U_1,\, U_2\,]$ and $[\, V_1,\, V_2\,]$ have orthonormal columns, $\Sigma_1 = \operatorname{diag}(\varsigma_1, \ldots, \varsigma_{\ell_f})$, $\Sigma_2 = \operatorname{diag}(\varsigma_{\ell_f+1}, \ldots, \varsigma_{r_p})$ with $r_p = \operatorname{rank}(L_p^T E R_p)$.
4. Compute the skinny singular value decomposition

$$L_i^T A R_i = U_3 \Theta_3 V_3^T, \tag{3.23}$$

   where U_3 and V_3 have orthonormal columns, $\Theta_3 = \operatorname{diag}(\theta_1, \ldots, \theta_{\ell_\infty})$ with $\ell_\infty = \operatorname{rank}(L_i^T A R_i)$.
5. Compute the projection matrices

$$W_\ell = [\, L_p U_1 \Sigma_1^{-1/2},\ \ L_i U_3 \Theta_3^{-1/2}\,], \quad T_\ell = [\, R_p V_1 \Sigma_1^{-1/2},\ \ R_i V_3 \Theta_3^{-1/2}\,].$$

6. Compute the reduced-order system

$$[\,\widetilde{E},\ \widetilde{A},\ \widetilde{B},\ \widetilde{C}\,] = [\,W_\ell^T E T_\ell,\ W_\ell^T A T_\ell,\ W_\ell^T B,\ C T_\ell\,].$$

This method has to be used with care, since if the original system (3.1) is highly unbalanced or if the angle between the deflating subspaces of the pencil $\lambda E - A$ corresponding to the finite and infinite eigenvalues is small, then the projection matrices W_ℓ and T_ℓ will be ill-conditioned. To avoid accuracy loss in the reduced-order model, a *square root balancing free method* has been proposed in [Var87] for standard state space systems. This method can be generalized to descriptor systems as follows.

Algorithm 3.3.2. Generalized Square Root Balancing Free (GSRBF) method.

INPUT: A realization $\mathbf{G} = [\,E,\ A,\ B,\ C\,]$ such that $\lambda E - A$ is c-stable.
OUTPUT: A reduced-order system $\widehat{\mathbf{G}} = [\,\widehat{E},\ \widehat{A},\ \widehat{B},\ \widehat{C}\,]$.

1. Compute the Cholesky factors R_p and L_p of the proper Gramians $\mathcal{G}_{pc} = R_p R_p^T$ and $\mathcal{G}_{po} = L_p L_p^T$ that satisfy (3.14) and (3.15), respectively.
2. Compute the Cholesky factors R_i and L_i of the improper Gramians $\mathcal{G}_{ic} = R_i R_i^T$ and $\mathcal{G}_{io} = L_i L_i^T$ that satisfy (3.16) and (3.17), respectively.
3. Compute the skinny singular value decomposition (3.22).
4. Compute the skinny singular value decomposition (3.23).
5. Compute the skinny QR decompositions

$$[\,R_p V_1,\ R_i V_3\,] = Q_R R_0, \qquad [\,L_p U_1,\ L_i U_3\,] = Q_L L_0,$$

   where $Q_R,\ Q_L \in \mathbb{R}^{n,\ell}$ have orthonormal columns and $R_0,\ L_0 \in \mathbb{R}^{\ell,\ell}$ are nonsingular.
6. Compute the reduced-order system

$$[\,\widehat{E},\ \widehat{A},\ \widehat{B},\ \widehat{C}\,] = [\,Q_L^T E Q_R,\ Q_L^T A Q_R,\ Q_L^T B,\ C Q_R\,].$$

The GSR and GSRBF methods are formally equivalent in the sense that in exact arithmetic they return reduced systems with the same transfer function. However, since the projection matrices Q_L and Q_R computed by the GSRBF method have orthonormal columns, they may be significantly less sensitive to perturbations than the projection matrices W_ℓ and T_ℓ computed by the GSR method. Observe that the realization $[\,\widehat{E},\ \widehat{A},\ \widehat{B},\ \widehat{C}\,]$ is, in general, not balanced and the pencil $\lambda\widehat{E} - \widehat{A}$ is not in the block diagonal form (3.21).

3.3.1 Stability and Approximation Error

Computing the reduced-order descriptor system via balanced truncation can be interpreted as follows. At first we transform the asymptotically stable descriptor system (3.1) to the block diagonal form

$$\left[\begin{array}{c|c} \check{W}(sE-A)\check{T} & \check{W}B \\ \hline C\check{T} & \end{array}\right] = \left[\begin{array}{cc|c} sE_f - A_f & 0 & B_f \\ 0 & sE_\infty - A_\infty & B_\infty \\ \hline C_f & C_\infty & \end{array}\right],$$

where $\check{W}$ and $\check{T}$ are nonsingular, the pencil $\lambda E_f - A_f$ has only those finite eigenvalues that are the finite eigenvalues of $\lambda E - A$, and all the eigenvalues of $\lambda E_\infty - A_\infty$ are infinite. Then we reduce the order of the subsystems $[\,E_f,\,A_f,\,B_f,\,C_f\,]$ and $[\,E_\infty,\,A_\infty,\,B_\infty,\,C_\infty\,]$ separately. Clearly, the reduced-order system (3.2) is asymptotically stable and minimal.

The described decoupling of system matrices is equivalent to the additive decomposition of the transfer function as $\mathbf{G}(s) = \mathbf{G}_{sp}(s) + \mathbf{P}(s)$, where

$$\mathbf{G}_{sp}(s) = C_f(sE_f - A_f)^{-1}B_f \quad \text{and} \quad \mathbf{P}(s) = C_\infty(sE_\infty - A_\infty)^{-1}B_\infty$$

are the strictly proper part and the polynomial part of $\mathbf{G}(s)$. The reduced-order system (3.2) has the transfer function $\widetilde{\mathbf{G}}(s) = \widetilde{\mathbf{G}}_{sp}(s) + \widetilde{\mathbf{P}}(s)$, where

$$\widetilde{\mathbf{G}}_{sp}(s) = \widetilde{C}_f(s\widetilde{E}_f - \widetilde{A}_f)^{-1}\widetilde{B}_f \quad \text{and} \quad \widetilde{\mathbf{P}}(s) = \widetilde{C}_\infty(s\widetilde{E}_\infty - \widetilde{A}_\infty)^{-1}\widetilde{B}_\infty$$

are the transfer functions of the reduced-order subsystems. For the subsystem $\mathbf{G}_{sp} = [\,E_f,\,A_f,\,B_f,\,C_f\,]$ with nonsingular E_f, we have the following upper bound on the $\mathbb{H}_\infty$-norm of the absolute error

$$\|\mathbf{G}_{sp} - \widetilde{\mathbf{G}}_{sp}\|_{\mathbb{H}_\infty} = \sup_{\omega\in\mathbb{R}} \|\mathbf{G}_{sp}(i\omega) - \widetilde{\mathbf{G}}_{sp}(\mathrm{i}\omega)\|_2 \le 2(\varsigma_{\ell_f+1} + \ldots + \varsigma_{n_f})$$

that can be derived similarly as in [Enn84, Glo84] for the standard state space case.

Reducing the order of the subsystem $\mathbf{P} = [\,E_\infty,\,A_\infty,\,B_\infty,\,C_\infty\,]$ is equivalent to the balanced truncation model reduction of the discrete-time system

$$\begin{aligned} A_\infty\xi_{k+1} &= E_\infty\xi_k + B_\infty\eta_k, \\ w_k &= C_\infty\xi_k, \end{aligned}$$

with a nonsingular matrix A_∞. The Hankel singular values of this system are just the improper Hankel singular values of (3.1). Since we truncate only the states corresponding to the zero improper Hankel singular values, the equality $\mathbf{P}(s) = \widetilde{\mathbf{P}}(s)$ holds and the index of the reduced-order system is equal to $\deg(\mathbf{P})+1$, where $\deg(\mathbf{P})$ denotes the degree of the polynomial $\mathbf{P}(s)$, or, equivalently, the multiplicity of the pole at infinity of the transfer function $\mathbf{G}(s)$. In this case the error system $\mathbf{G}(s) - \widetilde{\mathbf{G}}(s) = \mathbf{G}_{sp}(s) - \widetilde{\mathbf{G}}_{sp}(s)$ is strictly proper, and we have the following $\mathbb{H}_\infty$-norm error bound

$$\|\mathbf{G} - \widetilde{\mathbf{G}}\|_{\mathbb{H}_\infty} \leq 2(\varsigma_{\ell_f+1} + \ldots + \varsigma_{n_f}).$$

Existence of this error bound is an important property of the balanced truncation model reduction approach for descriptor systems. It makes this approach preferable compared, for instance, to moment matching techniques as in [FF95, Fre00, GGV96, Gri97].

3.3.2 Numerical Aspects

To reduce the order of the descriptor system (3.1) we have to compute the Cholesky factors of the proper and improper controllability and observability Gramians that satisfy the projected generalized Lyapunov equations (3.14), (3.15), (3.16) and (3.17). These factors can be determined using the *generalized Schur-Hammarling method* [Sty02a, Sty02b] without computing the solutions of Lyapunov equations explicitly. Combining this method with the GSR method, we obtain the following algorithm for computing the reduced-order descriptor system (3.2).

Algorithm 3.3.3. Generalized Schur-Hammarling Square Root method.

INPUT: A realization $\mathbf{G} = [\,E,\, A,\, B,\, C\,]$ such that $\lambda E - A$ is c-stable.
OUTPUT: A reduced-order realization $\widetilde{\mathbf{G}} = [\,\widetilde{E},\, \widetilde{A},\, \widetilde{B},\, \widetilde{C}\,]$.

1. Compute the generalized Schur form

$$E = V \begin{bmatrix} E_f & E_u \\ 0 & E_\infty \end{bmatrix} U^T \qquad \text{and} \qquad A = V \begin{bmatrix} A_f & A_u \\ 0 & A_\infty \end{bmatrix} U^T, \tag{3.24}$$

   where U and V are orthogonal, E_f is upper triangular nonsingular, E_∞ is upper triangular nilpotent, A_f is upper quasi-triangular and A_∞ is upper triangular nonsingular.

2. Compute the matrices $V^T B = \begin{bmatrix} B_u \\ B_\infty \end{bmatrix}$ and $CU = [\,C_f, \quad C_u\,]$.

3. Solve the system of generalized Sylvester equations

$$\begin{aligned} E_f Y - Z E_\infty &= -E_u, \\ A_f Y - Z A_\infty &= -A_u. \end{aligned} \tag{3.25}$$

4. Compute the Cholesky factors R_f, L_f, R_∞ and L_∞ of the solutions $X_{pc} = R_f R_f^T$, $X_{po} = L_f L_f^T$, $X_{ic} = R_\infty R_\infty^T$ and $X_{io} = L_\infty L_\infty^T$ of the generalized Lyapunov equations

$$E_f X_{pc} A_f^T + A_f X_{pc} E_f^T = -(B_u - Z B_\infty)(B_u - Z B_\infty)^T, \tag{3.26}$$

$$E_f^T X_{po} A_f + A_f^T X_{po} E_f = -C_f^T C_f, \tag{3.27}$$

$$A_\infty X_{ic} A_\infty^T - E_\infty X_{ic} E_\infty^T = B_\infty B_\infty^T, \tag{3.28}$$

$$A_\infty^T X_{io} A_\infty - E_\infty^T X_{io} E_\infty = (C_f Y + C_u)^T (C_f Y + C_u). \tag{3.29}$$

5. Compute the skinny singular value decompositions

$$L_f^T E_f R_f = [\,U_1,\ U_2\,] \begin{bmatrix} \Sigma_1 & \\ & \Sigma_2 \end{bmatrix} [\,V_1,\ V_2\,]^T, \qquad L_\infty^T A_\infty R_\infty = U_3 \Theta_3 V_3^T,$$

where $[\,U_1,\ U_2\,]$, $[\,V_1,\ V_2\,]$, U_3 and V_3 have orthonormal columns, $\Sigma_1 = \mathrm{diag}(\varsigma_1, \ldots, \varsigma_{\ell_f})$, $\Sigma_2 = \mathrm{diag}(\varsigma_{\ell_f+1}, \ldots, \varsigma_r)$, $\Theta_3 = \mathrm{diag}(\theta_1, \ldots, \theta_{\ell_\infty})$ with $r = \mathrm{rank}(L_f^T E_f R_f)$ and $\ell_\infty = \mathrm{rank}(L_\infty^T A_\infty R_\infty)$.

6. Compute $W_f = L_f U_1 \Sigma_1^{-1/2}$, $W_\infty = L_\infty U_3 \Theta_3^{-1/2}$, $T_f = R_f V_1 \Sigma_1^{-1/2}$ and $T_\infty = R_\infty V_3 \Theta_3^{-1/2}$.

7. Compute the reduced-order system $[\,\widetilde{E},\ \widetilde{A},\ \widetilde{B},\ \widetilde{C}\,]$ with

$$\widetilde{E} = \begin{bmatrix} I_{\ell_f} & 0 \\ 0 & W_\infty^T E_\infty T_\infty \end{bmatrix}, \qquad \widetilde{A} = \begin{bmatrix} W_f^T A_f T_f & 0 \\ 0 & I_{\ell_\infty} \end{bmatrix},$$

$$\widetilde{B} = \begin{bmatrix} W_f^T (B_u - Z B_\infty) \\ W_\infty^T B_\infty \end{bmatrix}, \qquad \widetilde{C} = [\,C_f T_f, \quad (C_f Y + C_u) T_\infty\,].$$

To compute the generalized Schur form (3.24) we can use the QZ algorithm [GV96, Wat00], the GUPTRI algorithm [DK93a, DK93b], or algorithms proposed in [BV88, Var98]. To solve the generalized Sylvester equation (3.25) one can use the generalized Schur method [KW89] or its recursive blocked modification [JK02] that is more suitable for large problems. The upper triangular Cholesky factors R_f, L_f^T, R_∞ and L_∞^T of the solutions of the generalized Lyapunov equations (3.26)-(3.29) can be determined without computing the solutions themselves using the generalized Hammarling method [Ham82, Pen98]. Furthermore, the singular value decomposition of $L_f^T E_f R_f$ and $L_\infty^T A_\infty R_\infty$, where all three factors are upper triangular, can be computed without forming these products explicitly, see [BELV91, Drm00, GSV00] and references therein.

Algorithm 3.3.3 and its balancing free version have been implemented as a MATLAB-based function `gbta` in the Descriptor Systems Toolbox[1] [Var00].

Since the generalized Schur-Hammarling method is based on computing the generalized Schur form (3.24), it costs $O(n^3)$ flops and has the memory complexity $O(n^2)$. Thus, this method can be used for problems of small and medium size. Unfortunately, it does not take into account the sparsity or any structure of the system and is not attractive for parallelization. Recently, iterative methods related to the alternating direction implicit (ADI) method and the Smith method have been proposed to compute low rank approximations of the solutions of standard large-scale sparse Lyapunov equations [Li00, LW02, Pen99a]. It was observed that the eigenvalues of the symmetric solutions of Lyapunov equations with low rank right-hand side generally decay very rapidly, and such solutions may be well approximated by low rank matrices, see [ASZ02, Pen00a, SZ02]. A similar result holds for projected generalized Lyapunov equations. Consider, for example, the projected GCALE

[1] `http://www.robotic.dlr.de/control/num/desctool.html`

(3.14). If it is possible to find a matrix X with a small number of columns such that XX^T is an approximate solution of (3.14), then X is referred to as the *low rank Cholesky factor* of the solution $\mathcal{G}_{pc}$ of the projected GCALE (3.14). It can be computed by the following algorithm that is a generalization of the *low rank alternating direction implicit* (LR-ADI) *method* for standard Lyapunov equation as suggested in [Li00, LW02, Pen99a].

Algorithm 3.3.4. Generalized LR-ADI method.

INPUT: Matrices $E, A \in \mathbb{R}^{n,n}$, $Q = P_l B \in \mathbb{R}^{n,m}$, shift parameters $\tau_1, \ldots, \tau_q \in \mathbb{C}^-$.
OUTPUT: A low rank Cholesky factor X_k of the Gramian $\mathcal{G}_{pc} \approx X_k X_k^T$.

1. $X^{(1)} = \sqrt{-2\mathrm{Re}(\tau_1)}\,(E + \tau_1 A)^{-1} Q, \quad X_1 = X^{(1)}$,
2. FOR $k = 2, 3, \ldots$
   a. $X^{(k)} = \sqrt{\dfrac{\mathrm{Re}(\tau_k)}{\mathrm{Re}(\tau_{k-1})}}\,\big(I - (\overline{\tau}_{k-1} + \tau_k)(E + \tau_k A)^{-1} A\big)\, X^{(k-1)}$,
   b. $X_k = [\, X_{k-1}, \;\; X^{(k)}\,]$.

   END FOR

If all the finite eigenvalues of the pencil $\lambda E - A$ lie in the open left half-plane, then X_k converges to the solution of the projected GCALE (3.14). The rate of convergence depends strongly on the choice of the shift parameters $\tau_1, \ldots, \tau_q$. The optimal shift parameters satisfy the generalized ADI minimax problem

$$\{\tau_1, \ldots, \tau_q\} = \underset{\{\tau_1,\ldots,\tau_q\} \in \mathbb{C}^-}{\arg\min} \; \max_{t \in \mathrm{Sp}_f(E,A)} \frac{|(1 - \overline{\tau}_1 t) \cdot \ldots \cdot (1 - \overline{\tau}_q t)|}{|(1 + \tau_1 t) \cdot \ldots \cdot (1 + \tau_q t)|},$$

where $\mathrm{Sp}_f(E, A)$ denotes the finite spectrum of the pencil $\lambda E - A$, see [Sty05]. The computation of the optimal shift parameters is a difficult problem, since the finite eigenvalues of the pencil $\lambda E - A$ (in particular, if it is large and sparse) are in general unknown and expensive to compute. Instead, sub-optimal ADI shift parameters $\tau_1, \ldots, \tau_q$ can be determined by a heuristic procedure as in [Pen99a, Algorithm 5.1] from a set of largest and smallest (in modulus) approximate finite eigenvalues of $\lambda E - A$ that may be computed by an Arnoldi process.

As a stopping criterion one can use the condition $\|X^{(k)}\| / \|X_k\| \leq tol$ with some matrix norm $\|\cdot\|$ and a user-defined tolerance tol. The iteration can also be stopped as soon as a *normalized residual norm*

$$\eta(E, A, P_l B; X_k) = \frac{\|E X_k X_k^T A^T + A X_k X_k^T E^T + P_l B B^T P_l^T\|}{\|P_l B B^T P_l^T\|}$$

satisfies $\eta(E, A, P_l B; X_k) \leq tol$ or a stagnation of $\eta(E, A, P_l B; X_k)$ is observed, see [Pen99a] for an efficient computation of the Frobenius norm based

normalized residuals. Note that if the low rank ADI method needs more iterations than the number of available ADI shift parameters, then we reuse these parameters in a cyclic manner.

It should also be noted that the matrices $(E + \tau_k A)^{-1}$ in Algorithm 3.3.4 do not have to be computed explicitly. Instead, we solve linear systems $(E + \tau_k A)x = P_l b$ either by computing (sparse) LU factorizations and forward/backward substitutions or by using iterative Krylov subspace methods [Saa96]. In the latter case the generalized low rank ADI method has the memory complexity $O(k_{ADI}mn)$ and costs $O(k_{\text{ls}}k_{ADI}mn)$ flops, where k_{ls} is the number of linear solver iterations and k_{ADI} is the number of ADI iterations. This method becomes efficient for large-scale sparse Lyapunov equations only if $k_{\text{ls}}k_{ADI}m$ is much smaller than n. Note that if the matrices E and A have a particular structure for which the hierarchical matrix arithmetic can be used, then also the methods proposed in [Hac00, HGB02] can be applied to compute the inverse of $E + \tau_k A$.

A major difficulty in the numerical solution of the projected Lyapunov equations by the low rank ADI method is that we need to compute the spectral projections P_l and P_r onto the left and right deflating subspaces of the pencil $\lambda E - A$ corresponding to the finite eigenvalues. This is in general very difficult, but in many applications, such as control of fluid flow, electrical circuit simulation and constrained multibody systems, the matrices E and A have some special block structure. This structure can be used to construct the projections P_l and P_r explicitly and cheaply, see [ET00, Mar96, Sch95, Sty04a].

3.3.3 Remarks

We close this section with some concluding remarks.

Remark 3.3.3. The GSR and the GSRBF methods can also be used to reduce the order of unstable descriptor systems. To do this we first compute the additive decomposition [KV92] of the transfer function $\mathbf{G}(s) = \mathbf{G}_-(s) + \mathbf{G}_+(s)$, where $\mathbf{G}_-(s) = C_-(sE_- - A_-)^{-1}B_-$ and $\mathbf{G}_+(s) = C_+(sE_+ - A_+)^{-1}B_+$. Here the matrix pencil $\lambda E_- - A_-$ is c-stable and all the eigenvalues of the pencil $\lambda E_+ - A_+$ are finite and have non-negative real part. Then we determine the reduced-order system $\widetilde{\mathbf{G}}_-(s) = \widetilde{C}_-(s\widetilde{E}_- - \widetilde{A}_-)^{-1}\widetilde{B}_-$ by applying the balanced truncation model reduction method to the subsystem $\mathbf{G}_- = [\, E_-,\, A_-,\, B_-,\, C_- \,]$. Finally, the reduced-order approximation of $\mathbf{G}(s)$ is given by $\widetilde{\mathbf{G}}(s) = \widetilde{\mathbf{G}}_-(s) + \mathbf{G}_+(s)$, where $\mathbf{G}_+(s)$ is included unmodified.

Remark 3.3.4. To compute a low order approximation to a large-scale descriptor system of index one with dense matrix coefficients E and A we can apply the spectral projection method [BQQ04]. This method is based on the disc and sign function iterative procedures and can be efficiently implemented on parallel computers.

Remark 3.3.5. An alternative model reduction approach for descriptor systems is the moment matching approximation which can be formulated as follows. Suppose that $s_0 \in \mathbb{C}$ is not an eigenvalue of the pencil $\lambda E - A$. Then the transfer function $\mathbf{G}(s) = C(sE - A)^{-1}B$ can be expanded into a Laurent series at s_0 as

$$\begin{aligned}\mathbf{G}(s) &= C\left(I - (s - s_0)(s_0E - A)^{-1}E\right)^{-1}(s_0E - A)^{-1}B \\ &= M_0 + M_1(s - s_0) + M_2(s - s_0)^2 + \dots,\end{aligned}$$

where the matrices $M_j = -C\left((s_0E - A)^{-1}E\right)^j(s_0E - A)^{-1}B$ are called the *moments* of the descriptor system (3.1) at s_0. The moment matching approximation problem for the descriptor system (3.1) consists in determining a rational matrix-valued function $\widetilde{\mathbf{G}}(s)$ such that the Laurent series expansion of $\widetilde{\mathbf{G}}(s)$ at s_0 has the form

$$\widetilde{\mathbf{G}}(s) = \widetilde{M}_0 + \widetilde{M}_1(s - s_0) + \widetilde{M}_2(s - s_0)^2 + \dots, \tag{3.30}$$

where the moments $\widetilde{M}_j$ satisfy the moment matching conditions

$$M_j = \widetilde{M}_j, \qquad j = 0, 1, \dots, k. \tag{3.31}$$

If $s_0 = \infty$, then $M_j = CF_{j-1}B$ are the Markov parameters of (3.1) and the corresponding approximation problem is known as *partial realization* [GL83]. Computation of the partial realization for descriptor systems is an open problem. For $s_0 = 0$, the approximation problem (3.30), (3.31) reduces to the *Padé approximation* problem [BG96]. Efficient algorithms based on Arnoldi and Lanzcos procedures for solving this problem have been presented in [FF95, GGV94]. For an arbitrary complex number $s_0 \neq 0$, the moment matching approximation is the problem of *rational interpolation* or *shifted Padé approximation* that has been considered in [Bai02, BF01, FF95, Fre00, GGV96]. Apart from a single interpolation point one can construct a reduced-order system with the transfer function $\widetilde{\mathbf{G}}(s)$ that matches $\mathbf{G}(s)$ at multiple points $\{s_0, s_1, \dots, s_k\}$. Such an approximation is called a *multi-point Padé approximation* or a *rational interpolant* [AA00, BG96]. It can be computed efficiently for descriptor systems by the rational Krylov subspace method [GGV96, Gri97, Ruh84].

3.4 Numerical Examples

In this section we present some numerical examples to illustrate the effectiveness of the described model reduction methods for descriptor systems. The computations were done on IBM RS 6000 44P Modell 270 with machine precision $\varepsilon = 2.22 \times 10^{-16}$ using MATLAB 6.5. We apply these methods to two different models: a semidiscretized Stokes equation and a constrained damped mass-spring system.

Semidiscretized Stokes Equation

Consider the instationary Stokes equation describing the flow of an incompressible fluid

$$\begin{aligned} \frac{\partial v}{\partial t} &= \Delta v - \nabla \rho + f, && (\xi, t) \in \Omega \times (0, t_e), \\ 0 &= \operatorname{div} v, && (\xi, t) \in \Omega \times (0, t_e) \end{aligned} \tag{3.32}$$

with appropriate initial and boundary conditions. Here $v(\xi, t) \in \mathbb{R}^d$ is the velocity vector ($d = 2$ or 3 is the dimension of the spatial domain), $\rho(\xi, t) \in \mathbb{R}$ is the pressure, $f(\xi, t) \in \mathbb{R}^d$ is the vector of external forces, $\Omega \subset \mathbb{R}^d$ is a bounded open domain and $t_e > 0$ is the endpoint of the time interval. The spatial discretization of the Stokes equation (3.32) by the finite difference method on a uniform staggered grid leads to a descriptor system

$$\begin{aligned} \dot{\mathbf{v}}_h(t) &= A_{11}\mathbf{v}_h(t) + A_{12}\boldsymbol{\rho}_h(t) + B_1 u(t), \\ 0 &= A_{12}^T \mathbf{v}_h(t) \qquad\qquad + B_2 u(t), \\ y(t) &= C_1 \mathbf{v}_h(t) + C_2 \boldsymbol{\rho}_h(t), \end{aligned} \tag{3.33}$$

where $\mathbf{v}_h(t) \in \mathbb{R}^{n_\mathbf{v}}$ and $\boldsymbol{\rho}_h(t) \in \mathbb{R}^{n_\rho}$ are the semidiscretized vectors of velocities and pressures, respectively, see [Ber90]. The matrix $A_{11} \in \mathbb{R}^{n_\mathbf{v}, n_\mathbf{v}}$ is the discrete Laplace operator, $-A_{12} \in \mathbb{R}^{n_\mathbf{v}, n_\rho}$ and $-A_{12}^T \in \mathbb{R}^{n_\rho, n_\mathbf{v}}$ are, respectively, the discrete gradient and divergence operators. Due to the non-uniqueness of the pressure, the matrix A_{12} has a rank defect one. In this case, instead of A_{12} we can take a full column rank matrix obtained from A_{12} by discarding the last column. Therefore, in the following we will assume without loss of generality that A_{12} has full column rank. In this case system (3.33) is of index 2. The matrices $B_1 \in \mathbb{R}^{n_\mathbf{v}, m}$, $B_2 \in \mathbb{R}^{n_\rho, m}$ and the control input $u(t) \in \mathbb{R}^m$ are resulting from the boundary conditions and external forces, the output $y(t)$ is the vector of interest. The order $n = n_\mathbf{v} + n_{\boldsymbol{\rho}}$ of system (3.33) depends on the level of refinement of the discretization and is usually very large, whereas the number m of inputs and the number p of outputs are typically small. Note that the matrix coefficients in (3.33) given by

$$E = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \qquad \text{and} \qquad A = \begin{bmatrix} A_{11} & A_{12} \\ A_{12}^T & 0 \end{bmatrix}$$

are sparse and have a special block structure. Using this structure, the projections P_l and P_r onto the left and right deflating subspaces of the pencil $\lambda E - A$ can be computed as

$$P_l = \begin{bmatrix} \Pi & -\Pi A_{11} A_{12} (A_{12}^T A_{12})^{-1} \\ 0 & 0 \end{bmatrix}, \quad P_r = \begin{bmatrix} \Pi & 0 \\ -(A_{12}^T A_{12})^{-1} A_{12}^T A_{11} \Pi & 0 \end{bmatrix},$$

where $\Pi = I - A_{12}(A_{12}^T A_{12})^{-1} A_{12}^T$ is the orthogonal projection onto $\operatorname{Ker}(A_{12}^T)$ along $\operatorname{Im}(A_{12})$, see [Sty04a].

The spatial discretization of the Stokes equation (3.32) on a square domain $\Omega = [0,1] \times [0,1]$ by the finite difference method on a uniform staggered 80×80 grid leads to a problem of order $n = 19520$. The dimensions of the deflating subspaces of the pencil $\lambda E - A$ corresponding to the finite and infinite eigenvalues are $n_f = 6400$ and $n_\infty = 13120$, respectively. In our experiments $B = [\, B_1^T,\ B_2^T\,]^T \in \mathbb{R}^{n,1}$ is chosen at random and we are interested in the first velocity component, i.e., $C = [\, 1,\ 0, \ldots, 0\,] \in \mathbb{R}^{1,n}$.

To reduce the order of the semidiscretized Stokes equation (3.33), we use the GSR and the GSRBF methods, where the exact Cholesky factors R_p and L_p of the proper Gramians are replaced by low rank Cholesky factors R_k and L_k, respectively, such that $\mathcal{G}_{pc} \approx R_k R_k^T$ and $\mathcal{G}_{po} \approx L_k L_k^T$. The matrices R_k and L_k have been computed by the generalized low rank ADI method with 20 shift parameters applied to $(E, A, P_l B)$ and $(E^T, A^T, P_r^T C^T)$, respectively.

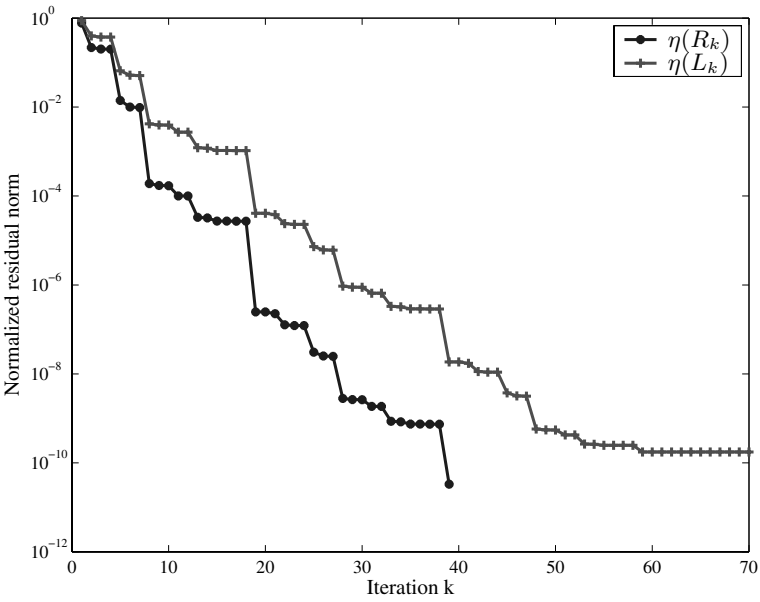


Fig. 3.1. Convergence history for the normalized residuals $\eta(R_k) = \eta(E, A, P_l B; R_k)$ and $\eta(L_k) = \eta(E^T, A^T, P_r^T C^T; L_k)$ for the semidiscretized Stokes equation.

In Figure 3.1 we present the convergence history for the normalized residuals $\eta(E, A, P_l B; R_k)$ and $\eta(E^T,\ A^T,\ P_r^T C^T; L_k)$ versus the iteration step k. Figure 3.2 shows the approximate dominant proper Hankel singular values $\widetilde{\varsigma}_j$ computed from the singular value decomposition of the matrix $L_{70}^T E R_{39}$ with $R_{39} \in \mathbb{R}^{n,39}$ and $L_{70} \in \mathbb{R}^{n,70}$. Note the Cholesky factors R_i and L_i of the improper Gramians of (3.33) can be computed in explicit form without solving the generalized Lyapunov equations (3.16) and (3.17) numerically,

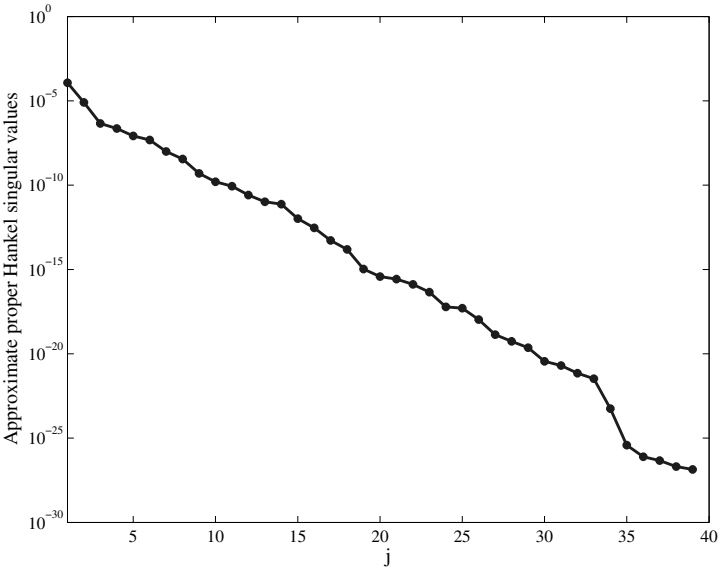


Fig. 3.2. Approximate proper Hankel singular values for the semidiscretized Stokes equation.

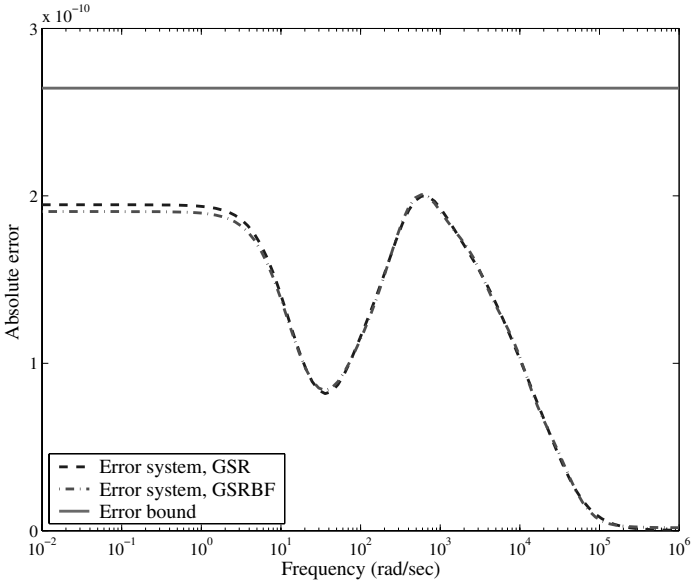


Fig. 3.3. Absolute error plots and error bound for the semidiscretized Stokes equation.

see [Sty04a]. System (3.33) has only one non-zero improper Hankel singular value $\theta_1 = 0.0049743$.

We approximate the semidiscretized Stokes equation (3.33) by two models of order $\ell = 11$ ($\ell_f = 10$, $\ell_\infty = 1$) computed by the approximate GSR and GSRBF methods. The absolute values of the frequency responses of the full order and the reduced-order systems are not presented, since they were impossible to distinguish. In Figure 3.3 we display the absolute errors $\|\mathbf{G}(\mathrm{i}\omega)-\widetilde{\mathbf{G}}(\mathrm{i}\omega)\|_2$ and $\|\mathbf{G}(\mathrm{i}\omega)-\widehat{\mathbf{G}}(\mathrm{i}\omega)\|_2$ for a frequency range $\omega \in [\,10^{-2},\ 10^{6}\,]$ as well as the approximate error bound computed as twice the sum of the truncated approximate Hankel singular values $\widetilde{\varsigma}_{11}, \ldots \widetilde{\varsigma}_{39}$. One can see that over the displayed frequency range the absolute errors are smaller than 2×10^{-10} which is much smaller than the discretization error which is of order 10^{-4}.

Constrained Damped Mass-Spring System

Consider the holonomically constrained damped mass-spring system illustrated in Figure 3.4.

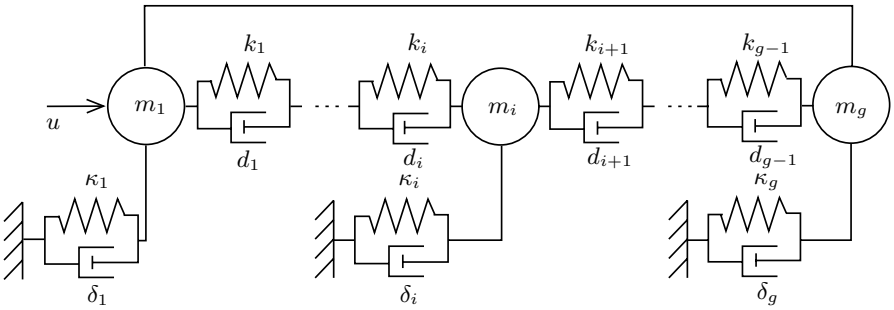


Fig. 3.4. A damped mass-spring system with a holonomic constraint.

The ith mass of weight m_i is connected to the $(i+1)$st mass by a spring and a damper with constants k_i and d_i, respectively, and also to the ground by a spring and a damper with constants κ_i and δ_i, respectively. Additionally, the first mass is connected to the last one by a rigid bar and it is influenced by the control $u(t)$. The vibration of this system is described by a descriptor system

$$\begin{aligned} \dot{\mathbf{p}}(t) &= \mathbf{v}(t), \\ M\dot{\mathbf{v}}(t) &= K\,\mathbf{p}(t) + D\mathbf{v}(t) - G^T\boldsymbol{\lambda}(t) + B_2 u(t), \\ 0 &= G\,\mathbf{p}(t), \\ y(t) &= C_1\mathbf{p}(t), \end{aligned} \tag{3.34}$$

where $\mathbf{p}(t) \in \mathbb{R}^g$ is the position vector, $\mathbf{v}(t) \in \mathbb{R}^g$ is the velocity vector, $\boldsymbol{\lambda}(t) \in \mathbb{R}$ is the Lagrange multiplier, $M = \mathrm{diag}(m_1, \ldots, m_g)$ is the

mass matrix, D and K are the tridiagonal damping and stiffness matrices, $G = [\,1, 0, \ldots, 0, -1\,] \in \mathbb{R}^{1,g}$ is the constraint matrix, $B_2 = e_1$ and $C_1 = [\,e_1,\ e_2,\ e_{g-1}\,]^T$. Here e_i denotes the ith column of the identity matrix I_g. The descriptor system (3.34) is of index 3 and the projections P_l and P_r can be computed as

$$P_l = \begin{bmatrix} \Pi_1 & 0 & -\Pi_1 M^{-1} D G_1 \\ -\Pi_1^T D (I - \Pi_1) & \Pi_1^T & -\Pi_1^T (K + D\Pi_1 M^{-1} D) G_1 \\ 0 & 0 & 0 \end{bmatrix},$$

$$P_r = \begin{bmatrix} \Pi_1 & 0 & 0 \\ -\Pi_1 M^{-1} D (I - \Pi_1) & \Pi_1 & 0 \\ G_1^T (K\Pi_1 - D\Pi_1 M^{-1} D (I - \Pi_1)) & G_1^T D \Pi_1 & 0 \end{bmatrix},$$

where $G_1 = M^{-1} G^T (G M^{-1} G^T)^{-1}$ and $\Pi_1 = I - G_1 G$ is a projection onto $\mathrm{Ker}\,(G)$ along $\mathrm{Im}\,(M^{-1} G^T)$, see [Sch95].

In our experiments we take $m_1 = \ldots = m_g = 100$ and

$$\begin{aligned} k_1 = \ldots = k_{g-1} = \kappa_2 = \ldots = \kappa_{g-1} = 2, \quad \kappa_1 = \kappa_g = 4, \\ d_1 = \ldots = d_{g-1} = \delta_2 = \ldots = \delta_{g-1} = 5, \quad \delta_1 = \delta_g = 10. \end{aligned}$$

For $g = 6000$, we obtain a descriptor system of order $n = 12001$ with $m = 1$ input and $p = 3$ outputs. The dimensions of the deflating subspaces of the pencil corresponding to the finite and infinite eigenvalues are $n_f = 11998$ and $n_\infty = 3$, respectively.

Figure 3.5 shows the normalized residual norms for the low rank Cholesky factors R_k and L_k of the proper Gramians computed by the generalized ADI method with 20 shift parameters. The approximate dominant proper Hankel singular values presented in Figure 3.6 have been determined from the singular value decomposition of the matrix $L_{33}^T E R_{31}$ with $L_{33} \in \mathbb{R}^{n,99}$ and $R_{31} \in \mathbb{R}^{n,31}$. All improper Hankel singular values are zero. This implies that the transfer function $\mathbf{G}(s)$ of (3.34) is proper. We approximate the descriptor system (3.34) by a standard state space system of order $\ell = \ell_f = 10$ computed by the approximate GSR method. In Figure 3.7 we display the magnitude and phase plots of the $(3,1)$ components of the frequency responses $\mathbf{G}(\mathrm{i}\omega)$ and $\widetilde{\mathbf{G}}(\mathrm{i}\omega)$. Note that there is no visible difference between the magnitude plots for the full order and reduced-order systems. Similar results have been observed for other components of the frequency response. Figure 3.8 shows the absolute error $\|\mathbf{G}(\mathrm{i}\omega) - \widetilde{\mathbf{G}}(\mathrm{i}\omega)\|_2$ for a frequency range $\omega \in [\,10^{-4},\ 10^4\,]$ and the approximate error bound computed as twice the sum of the truncated approximate proper Hankel singular values. We see that the reduced-order system approximates the original system satisfactorily.

3.5 Conclusions and Open Problems

In this paper we have presented a survey on balanced truncation model order reduction for linear time-invariant continuous-time descriptor systems.

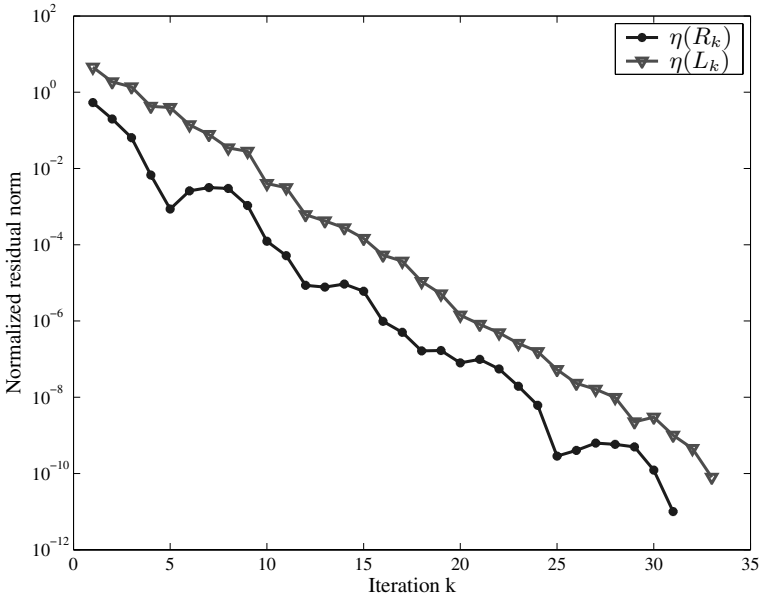


Fig. 3.5. Convergence history for the normalized residuals $\eta(R_k) = \eta(E, A, P_l B; R_k)$ and $\eta(L_k) = \eta(E^T, A^T, P_r^T C^T; L_k)$ for the damped mass-spring system.

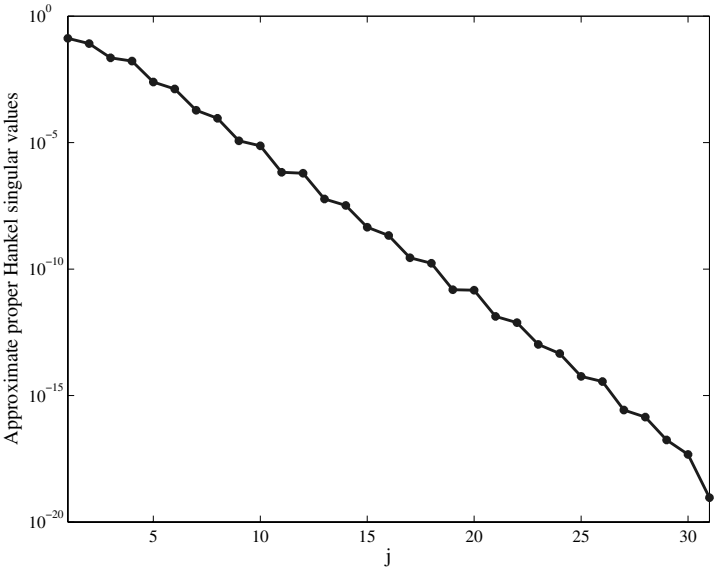


Fig. 3.6. Approximate proper Hankel singular values for the damped mass-spring system.

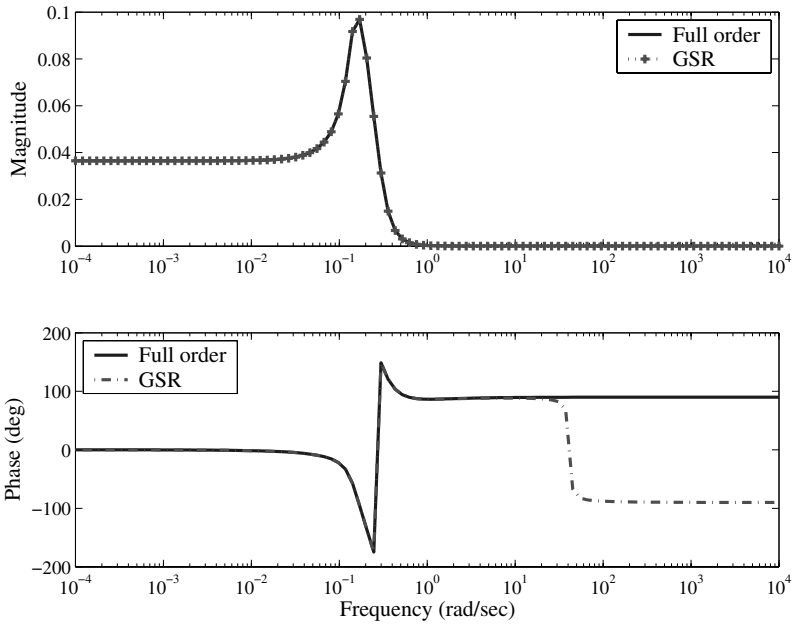


Fig. 3.7. Magnitude and phase plots of $G_{31}(\mathrm{i}\omega)$ for the damped mass-spring system.

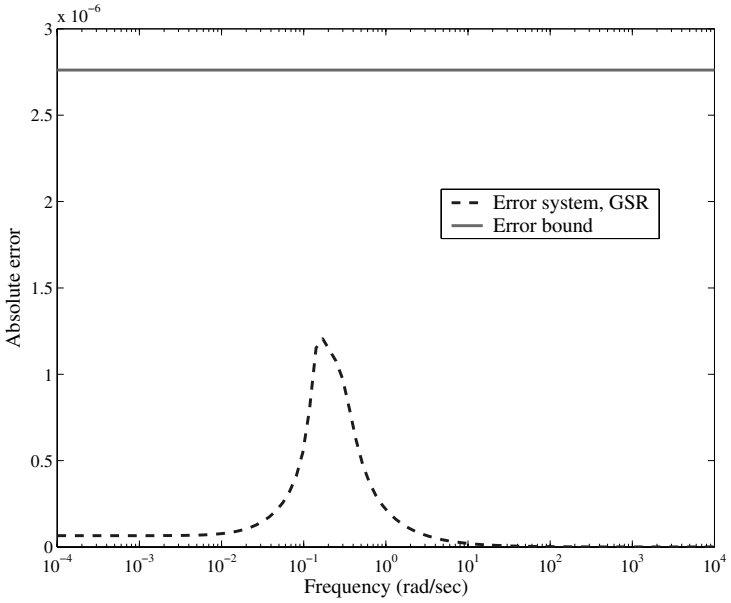


Fig. 3.8. Absolute error plot and error bound for the damped mass-spring system.

This approach is related to the proper and improper controllability and observability Gramians that can be computed by solving projected generalized Lyapunov equations.

The Gramians and Hankel singular values can also be generalized for discrete-time descriptor systems, see [Sty03] for details. In this case an extension of balanced truncation model reduction methods to such systems is straightforward.

More research in model reduction is needed. Here we collect some open problems:

- extension of decay rate bounds on the eigenvalues of the solutions of standard Lyapunov equations [ASZ02, Pen00a, SZ02] to generalized Lyapunov equations;
- development of more efficient algorithms for large-scale generalized Lyapunov equations;
- development of efficient algorithms for computing the optimal ADI shift parameters;
- extension of passivity preserving model reduction methods to descriptor systems that arise in electrical circuit simulation;
- development of structure preserving model reduction methods for systems of second order, for some work in this direction see Chapters 6, 7, and 8 in this book;
- development of model reduction methods for linear time-varying, nonlinear and coupled systems.

3.6 Acknowledgments

Supported by the DFG Research Center MATHEON "Mathematics for Key Technologies" in Berlin.

References

[AA00] Anderson, B.D.O., Antoulas, A.C.: Rational interpolation and state-variable realizations. Linear Algebra Appl., **137-138**, 479–509 (1990)

[Ant04] Antoulas, A.C.: Lectures on the Approximation of Large-Scale Dynamical Systems. SIAM, Philadelphia (2004)

[ASG01] Antoulas, A.C., Sorensen, D.C., Gugercin, S.: A survey of model reduction methods for large-scale systems. In: Olshevsky, V. (ed) Structured Matrices in Mathematics, Computer Science and Engineering, Vol. I. Contemporary Mathematics Series, 280, pages 193–219. American Mathematical Society (2001)

[ASG03] Antoulas, A.C., Sorensen, D.C., Gugercin, S.: A modified low-rank Smith method for large-scale Lyapunov equations. Numerical Algorithms, **32**, 27–55 (2003)

[ASZ02] Antoulas, A.C., Sorensen, D.C., Zhou, Y.: On the decay rate of the Hankel singular values and related issues. Systems Control Lett., **46**, 323–342 (2002)

[Bai02] Bai, Z.: Krylov subspace techniques for reduced-order modeling of large-scale dynamical systems. Appl. Numer. Math., **43**, 9–44 (2002)

[BBMN99] Bunse-Gerstner, A., Byers, R., Mehrmann, V., Nichols, N.K.: Feedback design for regularizing descriptor systems. Linear Algebra Appl., **299**, 119–151 (1999)

[Bea04] Beattie, Ch.: Potential theory in the analysis of iterative methods. Lecture Notes, Technische Universität Berlin, Berlin (2004)

[BCP89] Brenan, K.E., Campbell, S.L., Petzold, L.R.: The Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations. Elsevier, North-Holland, New York (1989)

[BELV91] Bojanczyk, A.W., Ewerbring, L.M., Luk, F.T., Van Dooren, P.: An accurate product SVD algorithm. Signal Process., **25**, 189–201 (1991)

[Ben97] Bender, D.J.: Lyapunov-like equations and reachability/observability Gramians for descriptor systems. IEEE Trans. Automat. Control, **32**, 343–348 (1987)

[Ber90] Bernert, K.: Differenzenverfahren zur Lösung der Navier-Stokes-Gleichungen über orthogonalen Netzen. Wissenschaftliche Schriftenreihe 10/1990, Technische Universität Chemnitz (1990) [German]

[BF01] Bai, Z., Freund, R.W.: A partial Padé-via-Lanczos method for reduced-order modeling. Linear Algebra Appl., **332–334**, 141–166 (2001)

[BG96] Baker Jr., G.A., Graves-Morris P.R.: Padé Approximants. Second edition. Encyclopedia of Mathematics and its Applications, 59. Cambridge University Press, Cambridge (1996)

[BQQ04] Benner, P., Quintana-Ortí, E.S., Quintana-Ortí, G.: Parallel model reduction of large-scale linear descriptor systems via balanced truncation. In: High Performance Computing for Computational Science. Proceedings of the 6th International Meeting VECPAR'04 (Valencia, Spain, June 28-30, 2004), pages 65–78 (2004)

[BSSY99] Bai, Z., Slone, R.D., Smith, W.T., Ye, Q.: Error bound for reduced system model by Padé approximation via the Lanczos process. IEEE Trans. Comput. Aided Design, **18**, 133–141 (1999)

[BV88] Beelen, T., Van Dooren, P.: An improved algorithm for the computation of Kronecker's canonical form of a singular pencil. Linear Algebra Appl., **105**, 9–65 (1988)

[Cam80] Campbell, S.L.: Singular Systems of Differential Equation, I. Pitman, San Francisco (1980)

[Cob84] Cobb, D.: Controllability, observability, and duality in singular systems. IEEE Trans. Automat. Control, **29**, 1076–1082 (1984)

[Dai89] Dai, L.: Singular Control Systems. Lecture Notes in Control and Information Sciences, 118. Springer, Berlin Heidelberg New York (1989)

[DK93a] Demmel, J.W., Kågström, B.: The generalized Schur decomposition of an arbitrary pencil $A - \lambda B$: robust software with error bounds and applications. Part I: Theory and algorithms. ACM Trans. Math. Software, **19**, 160–174 (1993)

[DK93b] Demmel, J.W., Kågström, B.: The generalized Schur decomposition of an arbitrary pencil $A - \lambda B$: robust software with error bounds and

applications. Part II: Software and applications. ACM Trans. Math. Software, **19**, 175–201 (1993)

[Doe71] Doetsch, G.: Guide to the Applications of the Laplace and Z-Transforms. Van Nostrand Reinhold Company, London (1971)

[Drm00] Drmač, Z.: New accurate algorithms for singular value decomposition of matrix triplets. SIAM J. Matrix Anal. Appl., **21**, 1026–1050 (2000)

[Enn84] Enns, D.: Model reduction with balanced realization: an error bound and a frequency weighted generalization. In: Proceedings of the 23rd IEEE Conference on Decision and Control (Las Vegas, 1984), pages 127–132. IEEE, New York (1984)

[ET00] Estévez Schwarz, D., Tischendorf, C.: Structural analysis for electric circuits and consequences for MNA. Int. J. Circ. Theor. Appl., **28**, 131–162 (2000)

[FF95] Feldmann, P., Freund, R.W.: Efficient linear circuit analysis by Padé approximation via the Lanczos process. IEEE Trans. Computer-Aided Design, **14**, 639–649 (1995)

[Fre00] Freund, R.W.: Krylov-subspace methods for reduced-order modeling in circuit simulation. J. Comput. Appl. Math., **123**, 395–421 (2000)

[FNG92] Fortuna, L., Nunnari, G., Gallo, A.: Model Order Reduction Techniques with Applications in Electrical Engineering. Springer, London (1992)

[GF99] Günther, M., Feldmann, U.: CAD-based electric-circuit modeling in industry. I. Mathematical structure and index of network equations. Surveys Math. Indust., **8**, 97–129 (1999)

[GGV94] Gallivan, K., Grimme, E., Van Dooren, P.: Asymptotic waveform evaluation via a Lanczos method. Appl. Math. Lett., **7**, 75–80 (1994)

[GGV96] Gallivan, K., Grimme, E., Van Dooren, P.: A rational Lanczos algorithm for model reduction. Numerical Algorithms, **12**, 33–63 (1996)

[GL83] Gragg, W.B., Lindquist, A.: On the partial realization problem. Linear Algebra Appl., **50**, 277–319 (1983)

[Glo84] Glover, K.: All optimal Hankel-norm approximations of linear multivariable systems and their L^∞-errors bounds. Internat. J. Control, **39**, 1115–1193 (1984)

[Gra04] Grasedyck, L.: Existence of a low rank of $\mathcal{H}$-matrix approximation to the solution of the Sylvester equation. Numer. Linear Algebra Appl., **11**, 371–389 (2004)

[Gri97] Grimme, E.: Krylov projection methods for model reduction. Ph.D. Thesis, Iniversity of Illinois, Urbana-Champaign (1997)

[GSV00] Golub, G.H., Sølna, K., Van Dooren, P.: Computing the SVD of a general matrix product/quotient. SIAM J. Matrix Anal. Appl., **22**, 1–19 (2000)

[Gug03] Gugercin, S.: Projection methods for model reduction of large-scale dynamical systems. Ph.D. Thesis, Rice University, Houston (2003)

[GV96] Golub, G.H., Van Loan, C.F.: Matrix Computations. 3rd ed. The Johns Hopkins University Press, Baltimore, London (1996)

[Hac00] Hackbusch, W.: A sparse matrix arithmetic based on $\mathcal{H}$-matrices. Part I: Introduction to $\mathcal{H}$-matrices. Computing, **62**, 89–108 (2000)

[Ham82] Hammarling, S.J.: Numerical solution of the stable non-negative definite Lyapunov equation. IMA J. Numer. Anal., **2**, 303–323 (1982)

[HGB02] Hackbusch, W., Grasedyck, L., Börm, S.: An introduction to hierarchical matrices. Math. Bohem., **127**, 229–241 (2002)

[JK02] Jonsson, I., Kågström, B.: Recursive blocked algorithms for solving triangular systems – Part I: One-sided and coupled Sylvester-type matrix equations. ACM Trans. Math. Software, **28**, 392–415 (2002)

[Kai80] Kailath, T.: Linear Systems. Prentice-Hall Information and System Sciences Series. Prentice Hall, Englewood Cliffs (1980)

[KV92] Kågström, B., Van Dooren, P.: A generalized state-space approach for the additive decomposition of a transfer function. J. Numer. Linear Algebra Appl., **1**, 165–181 (1992)

[KW89] Kågström, B., Westin, L.: Generalized Schur methods with condition estimators for solving the generalized Sylvester equation. IEEE Trans. Automat. Control, **34**, 745–751 (1989)

[LA89] Liu, Y., Anderson, B.D.O.: Singular perturbation approximation of balanced systems. Internat. J. Control, **50**, 1379–1405 (1989)

[LHPW87] Laub, A.J., Heath, M.T., Paige, C.C., Ward, R.C.: Computation of system balancing transformations and other applications of simultaneous diagonalization algorithms. IEEE Trans. Automat. Control, **32**, 115–122 (1987)

[Li00] Li, J.-R.: Model reduction of large linear systems via low rank system Gramians. Ph.D. Thesis, Department of Mathematics, Massachusetts Institute of Technology, Cambridge (2000)

[LS00] Liu, W.Q., Sreeram, V.: Model reduction of singular systems. In: Proceedings of the 39th IEEE Conference on Decision and Control (Sydney, Australia, 2000), pages 2373–2378. IEEE (2000)

[LW02] Li, J.-R., White, J.: Low rank solution of Lyapunov equations. SIAM J. Matrix Anal. Appl., **24**, 260–280 (2002)

[LWW99] Li, J.-R., Wang, F., White, J.: An efficient Lyapunov equation-based approach for generating reduced-order models of interconnect. In: Proceedings of the 36th Design Automation Conference (New Orleans, USA, 1999), pages 1–6. IEEE (1999)

[Mar96] März, R.: Canonical projectors for linear differential algebraic equations. Comput. Math. Appl., **31**, 121–135 (1996)

[Moo81] Moore, B.C.: Principal component analysis in linear systems: controllability, observability, and model reduction. IEEE Trans. Automat. Control, **26**, 17–32 (1981)

[Pen98] Penzl, T.: Numerical solution of generalized Lyapunov equations. Adv. Comput. Math., **8**, 33–48 (1998)

[Pen99a] Penzl, T.: A cyclic low-rank Smith method for large sparse Lyapunov equations. SIAM J. Sci. Comput., **21**, 1401–1418 (1999/2000)

[Pen99b] Penzl, T.: Algorithms for model reduction of large dynamical systems. Preprint SFB393/99-40, Fakultät für Mathematik, Technische Universität Chemnitz, D-09107 Chemnitz, Germany, (1999). Available from `http://www.tu-chemnitz.de/sfb393/sfb99pr.html`

[Pen00a] Penzl, T.: Eigenvalue decay bounds for solutions of Lyapunov equations: the symmetric case. Systems Control Lett., **40**, 139-144 (2000)

[Pen00b] Penzl, T.: LYAPACK Users Guide. Preprint SFB393/00-33, Fakultät für Mathematik, Technische Universität Chemnitz, D-09107 Chemnitz, Germany (2000). Available from `http://www.tu-chemnitz.de/sfb393/sfb00pr.html`

[PS94] Perev, K., Shafai, B.: Balanced realization and model reduction of singular systems. Internat. J. Systems Sci., **25**, 1039–1052 (1994)

[Rud87] Rudin, W.: Real and Complex Analysis. McGraw-Hill, New York (1987)

[Ruh84] Ruhe, A.: Rational Krylov sequence methods for eigenvalue computation. Linear Algebra Appl., **58**, 391–405 (1984)

[Saa96] Saad, Y.: Iterative Methods for Sparse Linear Systems. PWS Publishing Company, Boston (1996)

[SC89] Safonov, M.G., Chiang R.Y.: A Schur method for balanced-truncation model reduction. IEEE Trans. Automat. Control, **34**, 729–733 (1989)

[Sch95] Schüpphaus, R.: Regelungstechnische Analyse und Synthese von Mehrkörpersystemen in Deskriptorform. Ph.D. Thesis, Fachbereich Sicherheitstechnik, Bergische Universität-Gesamthochschule Wuppertal. Fortschritt-Berichte VDI, Reihe 8, Nr. 478. VDI Verlag, Düsseldorf (1995) [German]

[Sok03] Sokolov, V.I.: On realization of rational matrices. Technical Report 31-2003, Institut für Mathematik, Technische Universität Berlin, D-10263 Berlin, Germany (2003)

[SS90] Stewart, G.W., Sun, J.-G.: Matrix Perturbation Theory. Academic Press, New York (1990)

[Sty02a] Stykel, T.: Analysis and numerical solution of generalized Lyapunov Equations. Ph.D. Thesis, Institut für Mathematik, Technische Universität Berlin, Berlin (2002)

[Sty02b] Stykel, T.: Numerical solution and perturbation theory for generalized Lyapunov equations. Linear Algebra Appl., **349**, 155–185 (2002)

[Sty03] Stykel, T.: Input-output invariants for descriptor systems. Preprint PIMS-03-1, The Pacific Institute for the Mathematical Sciences, Canada (2003)

[Sty04a] Stykel, T.: Balanced truncation model reduction for semidiscretized Stokes equation. To appear in Linear Algebra Appl. (2004)

[Sty04b] Stykel, T.: Gramian-based model reduction for descriptor systems. Math. Control Signals Systems, **16**, 297–319 (2004)

[Sty05] Stykel, T.: Low rank iterative methods for projected generalized Lyapunov equations. Preprint 198, DFG Research Center Matheon, Technische Universität Berlin (2005).

[SZ02] Sorensen, D.C., Zhou, Y.: Bounds on eigenvalue decay rates and sensitivity of solutions of Lyapunov equations. Technical Report TR02-07, Department of Computational and Applied Mathematics, Rice University, Houston (2002). Available from `http://www.caam.rice.edu/caam/trs/2002/TR02-07.pdf`

[TP84] Tombs, M.S., Postlethweite, I.: Truncated balanced realization of a stable non-minimal state-space system. Internat. J. Control, **46**, 1319–1330 (1987)

[Var87] Varga, A.: Efficient minimal realization procedure based on balancing. In: EL Moudni, A., Borne, P., Tzafestas, S.G. (eds) Proceedings of the IMACS/IFAC Symposium on Modelling and Control of Technological Systems (Lille, France, May 7-10, 1991), volume 2, pages 42–47 (1991)

[Var98] Varga, A.: Computation of coprime factorizations of rational matrices. Linear Algebra Appl., **271**, 83–115 (1998)

[Var00] Varga, A.: A Descriptor Systems Toolbox for MATLAB. In: Proceedings of the 2000 IEEE International Symposium on Computer Aided Control System Design (Anchorage, Alaska, September 25-27, 2000),

pages 150–155 (2000). Available from `http://www.robotic.dlr.de/control/publications/2000/varga_cacsd2000p2.pdf`

[VLK81] Verghese, G.C., Lévy, B.C., Kailath, T.: A generalized state-space for singular systems. IEEE Trans. Automat. Control, **26**, 811–831 (1981)

[Wat00] Watkins, D.S.: Performance of the QZ algorithm in the presence of infinite eigenvalues. SIAM J. Matrix Anal. Appl., **22**, 364–375 (2000)

[YS81] Yip, E.L., Sincovec, R.F.: Solvability, controllability and observability of continuous descriptor systems. IEEE Trans. Automat. Control, **26**, 702–707 (1981)

[ZDG96] Zhou, K., Doyle, J.C., Glover, K.: Robust and Optimal Control. Prentice Hall, Upper Saddle River (1996)

4

On Model Reduction of Structured Systems

Danny C. Sorensen[1] and Athanasios C. Antoulas[2]

[1] Department of Computational and Applied Mathematics
Rice University
Houston, Texas 77251-1892, USA.

e-mail: sorensen@rice.edu

[2] Department of Electrical and Computer Engineering Rice University
Houston, Texas 77251-1892, USA e-mail: sorensen@rice.edu, aca@rice.edu

Summary. A general framework for defining the reachability and controllability Gramians of structured linear dynamical systems is proposed. The novelty is that a formula for the Gramian is given in the frequency domain. This formulation is surprisingly versatile and may be applied in a variety of structured problems. Moreover, this formulation enables a rather straightforward development of error bounds for model reduction in the $\mathcal{H}_2$ norm. The bound applies to a reduced model derived from projection onto the dominant eigenspace of the appropriate Gramian. The reduced models are structure preserving because they arise as direct reduction of the original system in the reduced basis. A derivation of the bound is presented and verified computationally on a second order system arising from structural analysis.

4.1 Introduction

The notion of reachability and observability Gramians is well established in the theory of linear time invariant first order systems. However, there are several competing definitions of these quantities for higher order or structured systems. In particular, for second order systems, at least two different concepts have been proposed (see [7, 8]).

One of the main interests in defining these Gramians is to develop a notion that will be suitable for model reduction via projection onto dominant invariant subspaces of the Gramians. The goal is to provide model reductions that posses error bounds analogous to those for balanced truncation of first order systems. The Gramian definitions proposed in [7] for second order systems attempt to achieve a balanced reduction that preserves the second order structure of the system. The work reported in [8] and [9] is also concerned with preservation of second order structure. While the definitions in these investigations are reasonable and reduction schemes based upon the proposed Gramians have been implemented, none of them have provided the desired error bounds.

In this paper, a fairly standard notion of Gramian is proposed. The novelty is that a formula for the Gramian is posed in the frequency domain. This formulation is surprisingly versatile and may be applied in a variety of structured problems. Moreover, this formulation in the frequency domain leads to error bounds in the $\mathcal{H}_2$ norm in a rather straightforward way.

The themes discussed here are the subject of a number of other contributions in this volume; we refer in particular to Chapters 5, 7 and 8.

In the remainder of this paper, we shall lay out the general framework and show how the formulation leads to natural Gramian definitions for a variety of structured problems. We then give a general derivation of an $\mathcal{H}_2$ norm error bound for model reduction based upon projection onto the dominant invariant subspace of the appropriate Gramian. An example of a structure preserving reduction of a second order system is provided to experimentally verify the validity of the bound. The numerical results indicate that the new bound is rather tight for this example.

4.2 A Framework for Formulating Structured System Gramians

Given is a system $\mathbf{\Sigma}$ described by the usual equations $\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t)$, $\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t)$, where $\mathbf{u}$, $\mathbf{x}$, $\mathbf{y}$ are the input, state, output and

$$\mathbf{\Sigma} = \left(\begin{array}{c|c} \mathbf{A} & \mathbf{B} \\ \hline \mathbf{C} & \mathbf{D} \end{array}\right) \in \mathbb{R}^{(n+p)\times(n+m)}. \tag{4.1}$$

We will assume that the system is stable, that is, $\mathbf{A}$ has eigenvalues in the left-half of the complex plane. The *reachability Gramian* of $\mathbf{\Sigma}$ is defined as

$$\mathcal{P} = \int_0^\infty \mathbf{x}(t)\mathbf{x}(t)^*\, dt, \tag{4.2}$$

where $\mathbf{x}$ is the solution of the state equation for $\mathbf{u}(t) = \delta(t)$, the unit impulse. Using Parseval's theorem, the Gramian can also be expressed in the frequency domain as

$$\mathcal{P} = \frac{1}{2\pi}\int_{-\infty}^\infty \mathbf{x}(i\omega)\mathbf{x}^*(-i\omega)\, d\omega, \tag{4.3}$$

where $\mathbf{x}$ denotes the Laplace transform of the time signal $\mathbf{x}$[3] Since the state due to an impulse is $\mathbf{x}(t) = e^{\mathbf{A}t}\mathbf{B}$ and equivalently $\mathbf{x}(i\omega) = (i\omega\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}$, the Gramian of $\mathbf{\Sigma}$ in time and in frequency is:

$$\mathcal{P} = \int_0^\infty e^{\mathbf{A}t}\mathbf{B}\mathbf{B}^* e^{\mathbf{A}^* t}\, dt = \frac{1}{2\pi}\int_{-\infty}^\infty (i\omega\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}\,\mathbf{B}^*(-i\omega\mathbf{I} - \mathbf{A}^*)^{-1}\, d\omega. \tag{4.4}$$

[3] For simplicity of notation, quantities in the time and frequency domains will be denoted by the same symbol.

This Gramian has the following variational interpretation. Let $\mathbf{J}(\mathbf{u}, t_1, t_2) = \int_{t_1}^{t_2} \mathbf{u}^*(t)\mathbf{u}(t)\,dt$, i.e. $\mathbf{J}$ is the norm of the input function $\mathbf{u}$ in the time interval $[t_1,\ t_2]$. The following statement holds

$$\min_{\mathbf{u}} \mathbf{J}(\mathbf{u}, -\infty, 0) = \mathbf{x}_0^* \mathcal{P}^{-1} \mathbf{x}_0 \ \text{ subject to } \ \dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u},\ \mathbf{x}(0) = \mathbf{x}_0;$$

That is, the minimal energy required to steer the system from rest at $t = -\infty$, to $\mathbf{x}_0$ at time $t = 0$ is given by $\mathbf{x}_0^* \mathcal{P}^{-1} \mathbf{x}_0$.

By duality, we also define the *observability Gramian* as follows:

$$\mathcal{Q} = \int_0^\infty e^{\mathbf{A}^* t} \mathbf{C}^* \mathbf{C} e^{\mathbf{A}t}\,dt = \frac{1}{2\pi} \int_{-\infty}^{\infty} (-i\omega \mathbf{I} - \mathbf{A}^*)^{-1} \mathbf{C}^*\, \mathbf{C} (i\omega \mathbf{I} - \mathbf{A})^{-1}\,d\omega.$$

A similar discussion of this observability Gramian will yield that the energy released by observing an uncontrolled state evolving from an initial position $\mathbf{x}_0$ at $t = 0$ and decaying to 0 at $t = \infty$ is given by $\mathbf{x}_0^* \mathcal{Q} \mathbf{x}_0$.

4.2.1 Gramians for Structured Systems

We will now turn our attention to the following types of structured systems, namely: weighted, second-order, closed-loop and unstable systems. In terms of their transfer functions, these systems are as follows.

Weighted systems:	$\mathbf{G_W}(s) = \mathbf{W}_o(s)\mathbf{G}(s)\mathbf{W}_i(s)$
Second order systems:	$\mathbf{G_2}(s) = (s\mathbf{C}_1 + \mathbf{C}_0)(s^2\mathbf{M} + s\mathbf{D} + \mathbf{K})^{-1}\mathbf{B}$
Systems in closed loop:	$\mathbf{G}_{\rm cl}(s) = \mathbf{G}(s)(\mathbf{I} + \mathbf{K}(s)\mathbf{G}(s))^{-1}$
Unstable systems:	$\mathbf{G}(s)$ with poles in $\mathbb{C}_+$.

4.2.2 Gramians for Structured Systems in Frequency Domain

In analogy with the case above, the reachability Gramian of these systems will be defined as $\int \mathbf{x}\mathbf{x}^*$. In the case of input weighted systems with weight $\mathbf{W}$, the state of the system is $t\mathbf{x}_\mathbf{W}(i\omega) = (i\omega\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}\mathbf{W}(i\omega)$. Similarly for systems in a closed loop the system state is $\mathbf{x}_{\rm cl}(i\omega) = (i\omega\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}(\mathbf{I} + \mathbf{K}(i\omega)\mathbf{G}(i\omega))^{-1}$. In the case of second-order systems where $\mathbf{x}$ is position and $\dot{\mathbf{x}}$ the velocity, we can define two Gramians, namely the position and velocity reachability Gramians. Let the system in this case be described as follows:

$$\mathbf{M}\ddot{\mathbf{x}}(t) + \mathbf{D}\dot{\mathbf{x}}(t) + \mathbf{K}\mathbf{x}(t) = \mathbf{B}\mathbf{u}(t), \quad \mathbf{y}(t) = \mathbf{C}_0\mathbf{x}(t) + \mathbf{C}_1\dot{\mathbf{x}}(t),$$

where $\det(\mathbf{M}) \neq 0$. In this case we can define the position Gramian

$$\mathcal{P}_0 = \int_0^\infty \mathbf{x}(t)\mathbf{x}^*(t)\,dt = \frac{1}{2\pi} \int_{-\infty}^{\infty} \mathbf{x}(i\omega)\mathbf{x}^*(-i\omega)\,d\omega,$$

and the velocity Gramian $\mathcal{P}_1 = \int \dot{\mathbf{x}}\dot{\mathbf{x}}^*$

$$\begin{aligned}\mathcal{P}_1 &= \int_0^\infty \dot{\mathbf{x}}(t)\dot{\mathbf{x}}^*(t)\,dt = \\ &= \frac{1}{2\pi}\int_{-\infty}^{\infty} (i\omega)\mathbf{x}(i\omega)\mathbf{x}^*(-i\omega)(-i\omega)\,d\omega = \\ &= \frac{1}{2\pi}\int_{-\infty}^{\infty} \omega^2\mathbf{x}(i\omega)\mathbf{x}^*(-i\omega)\,d\omega.\end{aligned}$$

$$\min_{\dot{\mathbf{x}}_0}\min_{\mathbf{u}} \mathbf{J}(\mathbf{u}, -\infty, 0)$$

subject to

$$\mathbf{M}\ddot{\mathbf{x}}(t) + \mathbf{D}\dot{\mathbf{x}}(t) + \mathbf{K}\mathbf{x}(t) = \mathbf{B}\mathbf{u}(t), \quad \mathbf{x}(0) = \mathbf{x}_0,$$

implies

$$\mathbf{J}_{\min} = \mathbf{x}_0^*\mathcal{P}_0^{-1}\mathbf{x}_0,$$

and

$$\min_{\mathbf{x}_0}\min_{\mathbf{u}} \mathbf{J}(\mathbf{u}, -\infty, 0)$$

subject to

$$\mathbf{M}\ddot{\mathbf{x}}(t) + \mathbf{D}\dot{\mathbf{x}}(t) + \mathbf{K}\mathbf{x}(t) = \mathbf{B}\mathbf{u}(t), \quad \dot{\mathbf{x}}(0) = \dot{\mathbf{x}}_0$$

implies

$$\mathbf{J}_{\min} = \dot{\mathbf{x}}_0^*\mathcal{P}_1^{-1}\dot{\mathbf{x}}_0.$$

Finally, for systems which are unstable (i.e. their poles are both in the right- and the left-half of the complex plane), the Gramian is the following expression in the frequency domain

$$\begin{aligned}\mathcal{P}_{\text{unst}} &= \frac{1}{2\pi}\int_{-\infty}^{\infty} \mathbf{x}(i\omega)\mathbf{x}^*(-i\omega)\,d\omega = \\ &= \frac{1}{2\pi}\int_{-\infty}^{\infty} (i\omega\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}\,\mathbf{B}^*(-i\omega\mathbf{I} - \mathbf{A}^*)^{-1}\,d\omega.\end{aligned}$$

These Gramians are summarized in table 4.1.

4.2.3 Gramians in the Time Domain

Our next goal is to express these Gramians in the time domain as (part of the) solutions of appropriately defined Lyapunov equations. Recall that if $\mathbf{A}$ has eigenvalues in $\mathbb{C}_-$, the reachability Gramian defined by (4.4) satisfies the following *Lyapunov equation*

$$\mathbf{A}\mathcal{P}(\mathbf{A}, \mathbf{B}) + \mathcal{P}(\mathbf{A}, \mathbf{B})\mathbf{A}^* + \mathbf{B}\mathbf{B}^* = \mathbf{0} \tag{4.5}$$

where for clarity the dependence of the Gramian on $\mathbf{A}$ and $\mathbf{B}$ is shown explicitly. With this notation, given that the transfer function of the original system

Table 4.1. Gramians of structured systems

$\mathcal{P}_{\mathbf{W}} = \frac{1}{2\pi}\int_{-\infty}^{\infty}(i\omega\mathbf{I}-\mathbf{A})^{-1}\mathbf{B}\,\mathbf{W}(i\omega)\mathbf{W}^*(-i\omega)\,\mathbf{B}^*(-i\omega\mathbf{I}-\mathbf{A}^*)^{-1}\,d\omega$
$\mathcal{P}_{\mathbf{2}} = \frac{1}{2\pi}\int_{-\infty}^{\infty}(-\omega^2\mathbf{M}+i\omega\mathbf{D}+\mathbf{K})^{-1}\mathbf{B}\,\mathbf{B}^*(-\omega^2\mathbf{M}^*-i\omega\mathbf{D}^*+\mathbf{K}^*)^{-1}\,d\omega$
$\mathcal{P}_{\mathbf{cl}} = \frac{1}{2\pi}\int_{-\infty}^{\infty}(i\omega\mathbf{I}-\mathbf{A})^{-1}\mathbf{B}\,(\mathbf{I}+\mathbf{K}(i\omega)\mathbf{G}(i\omega))^{-1}\cdot$ $\cdot\,(\mathbf{I}+\mathbf{K}^*(-i\omega)\mathbf{G}^*(-i\omega))^{-1}\,\mathbf{B}^*(-i\omega\mathbf{I}-\mathbf{A}^*)^{-1}\,d\omega$
$\mathcal{P}_{\mathbf{unst}} = \frac{1}{2\pi}\int_{-\infty}^{\infty}(i\omega\mathbf{I}-\mathbf{A})^{-1}\mathbf{B}\,\mathbf{B}^*(-i\omega\mathbf{I}-\mathbf{A}^*)^{-1}\,d\omega$

is $\mathbf{G}$, let the transfer function the weighted system be $\mathbf{W}_o\mathbf{G}\mathbf{W}_i$, where $\mathbf{W}_o$, $\mathbf{W}_i$ are the input and output weights respectively. The transfer function of the second-order system is $\mathbf{G_2}(s) = (\mathbf{C}_0 + \mathbf{C}_1 s)(\mathbf{M}s^2 + \mathbf{G}s + \mathbf{K})^{-1}\mathbf{B}$, while that of the closed loop system $\mathbf{G}_{\rm cl} = \mathbf{G}(\mathbf{I}+\mathbf{K}\mathbf{G})^{-1}$. Given the state space realizations for the three systems $\mathbf{\Sigma_W}$, $\mathbf{\Sigma_2}$, $\mathbf{\Sigma}_{\rm cl}$, collectively denoted as $\left(\begin{array}{c|c}\mathbf{A}_t & \mathbf{B}_t\\ \hline \mathbf{C}_t & \mathbf{D}_t\end{array}\right)$, the Gramians are as shown in Table 4.2.

Table 4.2. Gramians of structured systems in frequency domain

$\mathbf{\Sigma_W} = \left[\begin{array}{ccc\|c} \mathbf{A}_o & \mathbf{B}_o\mathbf{C} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A} & \mathbf{B}\mathbf{C}_i & \mathbf{B}\mathbf{D}_i \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_i & \mathbf{B}_i \\ \hline \mathbf{C}_o & \mathbf{D}_o\mathbf{C} & \mathbf{0} & \mathbf{0} \end{array}\right]$	$\mathcal{P}_{\mathbf{W}} = \begin{bmatrix}\mathbf{0} & \mathbf{I} & \mathbf{0}\end{bmatrix}\mathcal{P}(\mathbf{A}_t,\mathbf{B}_t)\begin{bmatrix}\mathbf{0}\\ \mathbf{I}\\ \mathbf{0}\end{bmatrix},$ $\mathcal{Q}_{\mathbf{W}} = [\mathcal{Q}(\mathbf{C}_t,\mathbf{A}_t)]_{11}$
$\mathbf{\Sigma_2} = \left[\begin{array}{cc\|c} \mathbf{0} & \mathbf{I} & \mathbf{0} \\ -\mathbf{M}^{-1}\mathbf{K} & -\mathbf{M}^{-1}\mathbf{D} & \mathbf{B} \\ \hline \mathbf{C}_1 & \mathbf{C}_0 & \mathbf{0} \end{array}\right]$	$\mathcal{P}_0 = \begin{bmatrix}\mathbf{I} & \mathbf{0}\end{bmatrix}\mathcal{P}(\mathbf{A}_t,\mathbf{B}_t)\begin{bmatrix}\mathbf{I}\\ \mathbf{0}\end{bmatrix}, \mathcal{Q}_0 = [\mathcal{Q}(\mathbf{C}_t,\mathbf{A}_t)]_{11}$ $\mathcal{P}_1 = \begin{bmatrix}\mathbf{0} & \mathbf{I}\end{bmatrix}\mathcal{P}(\mathbf{A}_t,\mathbf{B}_t)\begin{bmatrix}\mathbf{0}\\ \mathbf{I}\end{bmatrix}, \mathcal{Q}_1 = [\mathcal{Q}(\mathbf{C}_t,\mathbf{A}_t)]_{22}$
$\mathbf{\Sigma}_{\rm cl} = \left[\begin{array}{cc\|c} \mathbf{A} & -\mathbf{B}\mathbf{C}_c & \mathbf{B} \\ \mathbf{B}_c\mathbf{C} & \mathbf{A}_c & \mathbf{0} \\ \hline \mathbf{C} & \mathbf{0} & \mathbf{0} \end{array}\right]$	$\mathcal{P}_{\rm cl} = \begin{bmatrix}\mathbf{I} & \mathbf{0}\end{bmatrix}\mathcal{P}(\mathbf{A}_t,\mathbf{B}_t)\begin{bmatrix}\mathbf{I}\\ \mathbf{0}\end{bmatrix}, \mathcal{Q}_{\rm cl} = [\mathcal{Q}(\mathbf{C}_t,\mathbf{A}_t)]_{11}$

Lyapunov equations for unstable systems. The Gramian defined above for unstable systems satisfies a Lyapunov equation as well; for details see [4]:

$$\mathbf{A}\mathcal{P} + \mathcal{P}\mathbf{A}^* = \mathit{\Pi}\mathbf{B}\mathbf{B}^*\mathit{\Pi} - (\mathbf{I}-\mathit{\Pi})\mathbf{B}\mathbf{B}^*(\mathbf{I}-\mathit{\Pi})$$

where Π is the projection onto the **stable** eigenspace of $\mathbf{A}$. It turns out that $\Pi = \frac{1}{2}\mathbf{I} + \mathbf{S}$, where

$$\mathbf{S} = \frac{1}{2\pi}\int_{-\infty}^{\infty}(i\omega\mathbf{I}-\mathbf{A})^{-1}\,d\omega = \frac{i}{2\pi}\,\ln\left[(i\omega\mathbf{I}-\mathbf{A})^{-1}(-i\omega\mathbf{I}-\mathbf{A})\right]\Big|_{\omega=\infty}.$$

4.3 A Bound for the Approximation Error of Structured Systems

In order to introduce the class of systems under consideration we need the following notation. Let $\mathbf{Q}(s)$, $\mathbf{P}(s)$ be a polynomial matrices in s:

$$\mathbf{Q}(s) = \sum_{j=1}^{r}\mathbf{Q}_j s^j,\ \mathbf{Q}_j \in \mathbb{R}^{n\times n},\quad \mathbf{P}(s) = \sum_{j=1}^{r-1}\mathbf{P}_j s^j,\ \mathbf{P}_j \in \mathbb{R}^{n\times m},$$

where $\mathbf{Q}$ is invertible and $\mathbf{Q}^{-1}\mathbf{P}$ is a strictly proper rational matrix. We will denote by $\mathbf{Q}(\frac{d}{dt})$, $\mathbf{P}(\frac{d}{dt})$ the differential operators

$$\mathbf{Q}(\frac{d}{dt}) = \sum_{j=1}^{r}\mathbf{Q}_j\frac{d^j}{dt^j},\quad \mathbf{P}(\frac{d}{dt}) = \sum_{j=1}^{r-1}\mathbf{P}_j\frac{d^j}{dt^j}.$$

The systems are now defined by the following equations:

$$\mathbf{\Sigma}:\ \begin{cases}\mathbf{Q}(\frac{d}{dt})\mathbf{x} = \mathbf{P}(\frac{d}{dt})\mathbf{u}\\ \mathbf{y}(t) = \mathbf{C}\mathbf{x}(t)\end{cases} \tag{4.6}$$

where $\mathbf{C} \in \mathbb{R}^{p\times n}$.

Here, we give a direct reduction of the above system based upon the dominant eigenspace of a Gramian $\mathcal{P}$ that leads to an error bound in the $\mathcal{H}_2$ norm. An orthogonal basis for the dominant eigenspace of dimension k is used to construct a reduced model:

$$\hat{\mathbf{\Sigma}}:\ \begin{cases}\hat{\mathbf{Q}}(\frac{d}{dt})\hat{\mathbf{x}}(t) = \hat{\mathbf{P}}(\frac{d}{dt})\mathbf{u}(t),\\ \hat{\mathbf{y}}(t) = \hat{\mathbf{C}}\hat{\mathbf{x}}(t),\end{cases} \tag{4.7}$$

The *Gramian* is defined as the Gramian of $\mathbf{x}(t)$ when the input is an impulse:

$$\mathcal{P} := \int_0^{\infty}\mathbf{x}(t)\mathbf{x}(t)^*\mathrm{d}t.$$

Let

$$\mathcal{P} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^*\ \text{ with }\ \mathbf{V} = [\mathbf{V}_1, \mathbf{V}_2]\ \text{ and }\ \mathbf{\Lambda} = diag(\mathbf{\Lambda}_1, \mathbf{\Lambda}_2),$$

be the eigensystem of $\mathcal{P}$, where the diagonal elements of $\mathbf{\Lambda}$ are in decreasing order, and $\mathbf{V}$ is orthogonal. The reduced model is derived from

$$\hat{\mathbf{Q}}_j = \mathbf{V}_1^*\mathbf{Q}_j\mathbf{V}_1,\quad \hat{\mathbf{P}}_j = \mathbf{V}_1^*\mathbf{P}_j,\quad \hat{\mathbf{C}} = \mathbf{C}\mathbf{V}_1. \tag{4.8}$$

Our main result is the following:

Theorem 4.3.1. *The reduced model $\hat{\mathbf{\Sigma}}$ derived from the dominant eigenspace of the Gramian $\mathcal{P}$ for $\mathbf{\Sigma}$ as described above satisfies*

$$\boxed{\|\mathbf{\Sigma} - \hat{\mathbf{\Sigma}}\|_{\mathcal{H}_2}^2 \leq \operatorname{trace}\{\mathbf{C}_2\mathbf{\Lambda}_2\mathbf{C}_2^*\} + \kappa \operatorname{trace}\{\mathbf{\Lambda}_2\}}$$

where κ is a modest constant depending on $\mathbf{\Sigma}$, $\hat{\mathbf{\Sigma}}$, and the diagonal elements of $\mathbf{\Lambda}_2$ are the neglected smallest eigenvalues of $\mathcal{P}$.

The following discussion will establish this result.

4.3.1 Details

It is readily verified that the *transfer function* for (4.6) in the frequency domain is

$$\mathbf{H}(s) = \mathbf{C}\mathbf{Q}^{-1}(s)\mathbf{P}(s)$$

Moreover, in the frequency domain, the input-to-$\mathbf{x}$ and the input-to-output maps are

$$\hat{\mathbf{x}}(s) = \mathbf{Q}(s)^{-1}\mathbf{P}(s)\hat{\mathbf{u}}(s), \quad \hat{\mathbf{y}}(s) = \mathbf{H}(s)\hat{\mathbf{u}}(s)$$

If the input is an impulse: $\mathbf{u}(t) = \delta(t)\mathbf{I}$ and $\mathbf{u}(s) = \mathbf{I}$,

$$\hat{\mathbf{x}}(s) = \mathbf{Q}^{-1}(s)\mathbf{P}(s) \quad \text{and} \quad \hat{\mathbf{y}}(s) = \mathbf{H}(s).$$

In the time domain

$$\begin{aligned}\int_0^\infty \mathbf{y}^*\mathbf{y}\mathrm{d}t &= \operatorname{trace}\left\{\int_0^\infty \mathbf{y}\mathbf{y}^*\mathrm{d}t\right\} \\ &= \operatorname{trace}\left\{\int_0^\infty \mathbf{C}\mathbf{x}\mathbf{x}^*\mathbf{C}^*\mathrm{d}t\right\} = \operatorname{trace}(\mathbf{C}\mathcal{P}\mathbf{C}^*).\end{aligned}$$

Define $\mathbf{F}(s) := \mathbf{Q}^{-1}(s)\mathbf{P}(s)$. From the Parseval theorem, the above expression is equal to

$$\operatorname{trace}\left\{\int_0^\infty \mathbf{y}\mathbf{y}^*\mathrm{d}t\right\} = \operatorname{trace}\{\mathbf{C}\underbrace{\left(\frac{1}{2\pi}\int_{-\infty}^{\infty}\mathbf{F}(i\omega)\mathbf{F}(i\omega)^*\mathrm{d}\omega\right)}_{\mathcal{P}}\mathbf{C}^*\}.$$

Thus the Gramian in the frequency domain is

$$\mathcal{P} = \frac{1}{2\pi}\int_{-\infty}^{\infty}\mathbf{F}(i\omega)\mathbf{F}(i\omega)^*\mathrm{d}\omega.$$

Remark 4.3.2. Representation (4.6) is general as every system with a strictly proper rational transfer function can be represented this way. In particular the usual form of second-order systems introduced earlier falls into this category. Key for our considerations is the fact that the (square of the) $\mathcal{H}_2$ norm is given by $\operatorname{trace}(\mathbf{C}\mathcal{P}\mathbf{C}^*)$. If instead of the output map, the input map is constant, the same framework can be applied by considering the transpose (dual) of the original system. □

4.3.2 Reduction via the Gramian

For model reduction, we consider again the eigen-decomposition of the symmetric positive definite matrix $\mathcal{P}$. Let

$$\mathcal{P} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^* \text{ with } \mathbf{V} = [\mathbf{V}_1, \ \mathbf{V}_2] \text{ and } \mathbf{\Lambda} = diag(\mathbf{\Lambda}_1, \ \mathbf{\Lambda}_2),$$

where the diagonal elements of $\mathbf{\Lambda}$ are in decreasing order, and $\mathbf{V}$ is orthogonal. The system is now transformed using $\mathbf{V}$ as in (4.8), to wit, $\mathbf{Q}_j \leftarrow \mathbf{V}^*\mathbf{Q}_j\mathbf{V}$, $\mathbf{P}_j \leftarrow \mathbf{V}^*\mathbf{P}_j$, $\mathbf{C} \leftarrow \mathbf{C}\mathbf{V}$ which implies $\mathbf{F}(s) \leftarrow \mathbf{V}^*\mathbf{F}(s)$. In this new coordinate system the resulting Gramian is diagonal. We now partition

$$\mathbf{Q}(s) = \begin{bmatrix} \mathbf{Q}_{11}(s) \ \mathbf{Q}_{12}(s) \\ \mathbf{Q}_{21}(s) \ \mathbf{Q}_{22}(s) \end{bmatrix}, \quad [\mathbf{C}_1, \ \mathbf{C}_2] = \mathbf{C}\mathbf{V},$$

$$\mathbf{P}(s) = \begin{bmatrix} \mathbf{P}_1(s) \\ \mathbf{P}_2(s) \end{bmatrix} \text{ and } \mathbf{F}(s) = \begin{bmatrix} \mathbf{F}_1(s) \\ \mathbf{F}_2(s) \end{bmatrix}.$$

Note the relationship $\mathbf{Q}(s)\mathbf{F}(s) = \mathbf{P}(s)$. Let $\hat{\mathbf{Q}}(s) := \mathbf{Q}_{11}(s)$; since $\mathbf{\Lambda} = \frac{1}{2\pi}\int_{-\infty}^{\infty} \mathbf{F}(i\omega)\mathbf{F}(i\omega)^* \mathrm{d}\omega$, the following relationships hold

$$\begin{aligned} \mathbf{\Lambda}_1 &= \frac{1}{2\pi}\int_{-\infty}^{\infty} \mathbf{F}_1(i\omega)\mathbf{F}_1(i\omega)^* \mathrm{d}\omega, \\ \mathbf{\Lambda}_2 &= \frac{1}{2\pi}\int_{-\infty}^{\infty} \mathbf{F}_2(i\omega)\mathbf{F}_2(i\omega)^* \mathrm{d}\omega, \\ \mathbf{0} &= \frac{1}{2\pi}\int_{-\infty}^{\infty} \mathbf{F}_2(i\omega)\mathbf{F}_1(i\omega)^* \mathrm{d}\omega, \end{aligned}$$

while

$$\mathrm{trace}\,\{\mathbf{\Lambda}_1\} = \frac{1}{2\pi}\int_{-\infty}^{\infty} \|\mathbf{F}_1(i\omega)\|_F^2 \mathrm{d}\omega, \quad \mathrm{trace}\,\{\mathbf{\Lambda}_2\} = \frac{1}{2\pi}\int_{-\infty}^{\infty} \|\mathbf{F}_2(i\omega)\|_F^2 \mathrm{d}\omega.$$

The reduced system is now constructed as follows:

$$\hat{\mathbf{Q}}_j = [\mathbf{Q}_j]_{11}, \ \hat{\mathbf{P}}_j = [\mathbf{P}_j]_{11}, \ \hat{\mathbf{C}} = \mathbf{C}_1.$$

Given $\hat{\mathbf{Q}}(s)$ as above we define $\hat{\mathbf{F}}$ by means of the equation $\hat{\mathbf{Q}}(s)\hat{\mathbf{F}}(s) = \mathbf{P}_1$. As a consequence of these definitions the Gramian corresponding to the reduced system is

$$\hat{\mathcal{P}} = \frac{1}{2\pi}\int_{-\infty}^{\infty} \hat{\mathbf{F}}(i\omega)\hat{\mathbf{F}}(i\omega)^* \mathrm{d}\omega,$$

and from the defining equation for $\mathbf{F}(s)$ we have

$$\mathbf{F}_1(s) = \mathbf{Q}_{11}(s)^{-1}[\mathbf{P}_1(s) - \mathbf{Q}_{12}(s)\mathbf{F}_2(s)] = \hat{\mathbf{F}}(s) - \mathbf{Q}_{11}(s)^{-1}\mathbf{Q}_{12}(s)\mathbf{F}_2(s).$$

Let $\mathbf{L}(s) := \mathbf{Q}_{11}(s)^{-1}\mathbf{Q}_{12}(s)$; if the reduced system has no poles on the imaginary axis, $\sup_\omega \|\mathbf{L}(i\omega)\|_2$ is finite. Thus,

$$\hat{\mathbf{F}}(s) = \mathbf{F}_1(s) + \mathbf{L}(s)\mathbf{F}_2(s).$$

4.3.3 Bounding the $\mathcal{H}_2$ Norm of the Error System

Applying the same input $\mathbf{u}$ to both the original and the reduced systems, let $\mathbf{y} = \mathbf{C}\mathbf{x}$, $\hat{\mathbf{y}} = \hat{\mathbf{C}}\hat{\mathbf{x}}$, be the resulting outputs. If we denote by $\mathbf{H}_e(s)$ the transfer function of the error system $\mathcal{E} = \boldsymbol{\Sigma} - \hat{\boldsymbol{\Sigma}}$, we have

$$\mathbf{y}(s) - \hat{\mathbf{y}}(s) = \mathbf{H}_e(s)\mathbf{u}(s) = \left[\mathbf{C}\mathbf{Q}(s)^{-1}\mathbf{P}(s) - \hat{\mathbf{C}}\hat{\mathbf{Q}}(s)^{-1}\hat{\mathbf{P}}(s)\right]\mathbf{u}(s).$$

The $\mathcal{H}_2$-norm in the of the error system is therefore

$$\begin{aligned}
\|\mathcal{E}\|_{\mathcal{H}_2}^2 &= \operatorname{trace}\{\frac{1}{2\pi}\int_{-\infty}^{\infty}\mathbf{H}_e(i\omega)\mathbf{H}_e(i\omega)^*\mathrm{d}t\} \\
&= \underbrace{\frac{1}{2\pi}\int_{-\infty}^{\infty}\operatorname{trace}\{\mathbf{C}\mathbf{F}(i\omega)(\mathbf{C}\mathbf{F}(i\omega))^*\}\mathrm{d}t}_{\eta_1} \quad - \\
&\quad - 2\underbrace{\frac{1}{2\pi}\int_{-\infty}^{\infty}\operatorname{trace}\{\mathbf{C}\mathbf{F}(i\omega)(\hat{\mathbf{C}}\hat{\mathbf{F}}(i\omega))^*\}\mathrm{d}t}_{\eta_2} \quad + \\
&\quad + \underbrace{\frac{1}{2\pi}\int_{-\infty}^{\infty}\operatorname{trace}\{\hat{\mathbf{C}}\hat{\mathbf{F}}(i\omega)(\hat{\mathbf{C}}\hat{\mathbf{F}}(i\omega))^*\}\mathrm{d}t}_{\eta_3}.
\end{aligned}$$

Each of the three terms in this expression can be simplified as follows:

$$\eta_1 = \operatorname{trace}\{\mathbf{C}_1 S_1 \mathbf{C}_1^*\} + \operatorname{trace}\{\mathbf{C}_2 S_2 \mathbf{C}_2^*\},$$

$$\eta_2 = \operatorname{trace}\{\mathbf{C}_1 S_1 \mathbf{C}_1^*\} + \frac{1}{2\pi}\int_{-\infty}^{\infty}\operatorname{trace}\{\mathbf{C}\mathbf{F}(i\omega)\mathbf{F}_2(i\omega)^*\mathbf{L}(i\omega)^*\mathbf{C}_1^*\}\mathrm{d}t,$$

$$\begin{aligned}
\eta_3 &= \operatorname{trace}\{\mathbf{C}_1 S_1 \mathbf{C}_1^*\} + \frac{1}{2\pi}\int_{-\infty}^{\infty} 2\operatorname{trace}\{\mathbf{C}_1\mathbf{F}_1(i\omega)\mathbf{F}_2(i\omega)^*\mathbf{L}(i\omega)^*\mathbf{C}_1^*\}\mathrm{d}t + \\
&\quad + \frac{1}{2\pi}\int_{-\infty}^{\infty}\operatorname{trace}\{(\mathbf{C}_1\mathbf{L}(i\omega)\mathbf{F}_2(i\omega))(\mathbf{C}_1\mathbf{L}(i\omega)\mathbf{F}_2(i\omega))^*\}\mathrm{d}t.
\end{aligned}$$

Combining the above expressions we obtain

$$\begin{aligned}
\|\mathcal{E}\|_{\mathcal{H}_2}^2 &= \operatorname{trace}\{\mathbf{C}_2 S_2 \mathbf{C}_2^*\} + \\
&\quad + \frac{1}{2\pi}\int_{-\infty}^{\infty}\operatorname{trace}\{(\mathbf{C}_1\mathbf{L}(i\omega) - 2\mathbf{C}_2)\mathbf{F}_2(i\omega)(\mathbf{C}_1\mathbf{L}(i\omega)\mathbf{F}_2(i\omega))^*\}\mathrm{d}t.
\end{aligned}$$

The first term in the above expression is the $\mathcal{H}_2$ norm of the neglected term. The second term has the following upper bound

$$\sup_{\omega}\|(\mathbf{C}_1\mathbf{L}(i\omega))^*(\mathbf{C}_1\mathbf{L}(i\omega) - 2\mathbf{C}_2)\|_2 \operatorname{trace}\{\boldsymbol{\Lambda}_2\}.$$

This leads to the main result

$$\|\mathcal{E}\|_{\mathcal{H}_2}^2 \leq \operatorname{trace}\{\mathbf{C}_2\mathbf{\Lambda}_2\mathbf{C}_2^*\} + \kappa \operatorname{trace}\{\mathbf{\Lambda}_2\} \tag{4.9}$$

$$\text{where} \quad \kappa = \sup_{\omega} \|(\mathbf{C}_1\mathbf{L}(i\omega))^*(\mathbf{C}_1\mathbf{L}(i\omega) - 2\mathbf{C}_2)\|_2 \tag{4.10}$$

4.3.4 Special Case: Second-Order Systems

We shall now consider second-order systems. These are described by equations (4.7) where $\mathbf{Q}(s) = \mathbf{M}s^2 + \mathbf{D}s + \mathbf{K}$ and $\mathbf{P}(s) = \mathbf{B}$:

$$\mathbf{\Sigma}: \ \mathbf{M}\ddot{\mathbf{x}} + \mathbf{D}\dot{\mathbf{x}} + \mathbf{K}\mathbf{x} = \mathbf{B}\mathbf{u}, \quad \mathbf{y}(t) = \mathbf{C}\mathbf{x}(t), \tag{4.11}$$

with $\mathbf{M}$, $\mathbf{D}$, $\mathbf{K} \in \mathbb{R}^{n\times n}$, $\mathbf{B} \in \mathbb{R}^{n\times m}$, $\mathbf{C} \in \mathbb{R}^{p\times n}$.

It is standard to convert this system to an equivalent first order linear time invariant (LTI) system and then to apply existing reduction techniques to reduce the first order system. A difficulty with this approach is that the second-order form is lost in the reduction process and there is a mixing of the state variables and their first derivatives. Several researchers (see e.g. [8], [9], [7]) have noted undesirable consequences and have endeavored to provide either direct reductions of the second-order form or structure preserving reductions of the equivalent first order system. This has required several alternative definitions of a Gramian. However, while successful structure preserving reductions have been obtained, none of these possess error bounds.

Here, we give a direct reduction of the second-order system based upon the dominant eigenspace of a Gramian $\mathcal{P}$ that does lead to an error bound in the $\mathcal{H}_2$ norm. An orthogonal basis for the dominant eigenspace of dimension k is used to construct a reduced model in second-order form:

$$\hat{\mathbf{\Sigma}}: \ \hat{\mathbf{M}}\ddot{\hat{\mathbf{x}}}(t) + \hat{\mathbf{D}}\dot{\hat{\mathbf{x}}}(t) + \hat{\mathbf{K}}\hat{\mathbf{x}}(t) = \hat{\mathbf{B}}\mathbf{u}(t), \quad \hat{\mathbf{y}}(t) = \hat{\mathbf{C}}\hat{\mathbf{x}}(t).$$

The *Gramian* is defined as before, i.e. $\mathcal{P} = \int_0^\infty \mathbf{x}\mathbf{x}^* dt$. Let $\mathcal{P} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^*$, with $\mathbf{V} = [\mathbf{V}_1, \mathbf{V}_2]$ and $\mathbf{\Lambda} = diag(\mathbf{\Lambda}_1, \mathbf{\Lambda}_2)$. The reduced model is derived by letting $\hat{\mathbf{M}} = \mathbf{V}_1^*\mathbf{M}\mathbf{V}_1$, $\hat{\mathbf{D}} = \mathbf{V}_1^*\mathbf{D}\mathbf{V}_1$, $\hat{\mathbf{K}} = \mathbf{V}_1^*\mathbf{K}\mathbf{V}_1$, $\hat{\mathbf{B}} = \mathbf{V}_1^*\mathbf{B}$, $\hat{\mathbf{C}} = \mathbf{C}\mathbf{V}_1$.

Remark 4.3.3. The above method applies equally to first-order systems, that is systems described by the equations $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$, $\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u}$. We will not pursue the details in this case here. □

An Illustrative Example

The bound derived in the previous section involves the computation of the constant κ. The purpose of this section is to provide an example that will demonstrate that this constant is likely to be of reasonable magnitude. Our example is constructed to be representative of the structural analysis of a

building under the assumption of proportional damping ($\mathbf{D} = \alpha\mathbf{M} + \beta\mathbf{K}$, for specified positive scalars α and β). In this case the matrices $\mathbf{M}, \mathbf{D}, \mathbf{K}$ may be simultaneously diagonalized. Moreover, since both $\mathbf{M}$ and $\mathbf{K}$ are positive definite, the system can be transformed to an equivalent one where $\mathbf{M} = \mathbf{I}$ and $\mathbf{K}$ is a diagonal matrix with positive diagonal entries.

The example may then be constructed by specification of the diagonal matrix $\mathbf{K}$, the proportionality constants α and β, and the vectors $\mathbf{B}$ and $\mathbf{C}$. We constructed $\mathbf{K}$ to have its smallest 200 eigenvalues specified as the smallest 200 eigenvalues of an actual building model of dimension $26,000$. (For a description of the model, see Chapter 24, Section 6, this volume.) These eigenvalues are in the range $[7.7, 5300]$. We augmented these with equally spaced eigenvalues $[5400 : 2000 : 400000]$ to obtain a diagonal matrix $\mathbf{K}$ of order $n = 398$. We chose the proportionality constants $\alpha = .67$, $\beta = .0033$, to be consistent with the original building model. We specified $\mathbf{B} = \mathbf{C}^*$ to be vectors with all ones as entries. This is slightly inconsistent with the original building model but still representative. The eigenvalues of the second-order system resulting from this specification are shown in Figure 4.1

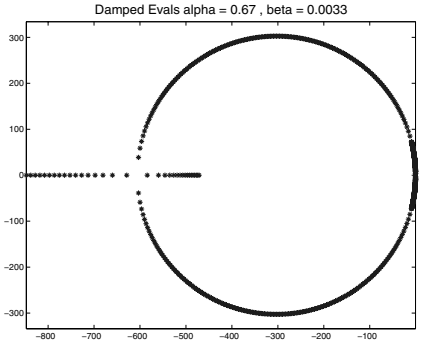


Fig. 4.1. Eigenvalues of a proportionally damped structure

The Gramian for Proportional Damping

To proceed we need to compute the Gramian of this system. Recall from table 4.1 that

$$\mathcal{P} = \frac{1}{2\pi}\int_{-\infty}^{\infty} (-\omega^2\mathbf{M} + i\omega\mathbf{D} + \mathbf{K})^{-1}\mathbf{B}\mathbf{B}^*(-\omega^2\mathbf{M}^* - i\omega\mathbf{D}^* + \mathbf{K}^*)^{-1}\, d\omega.$$

Since $\mathbf{M} = \mathbf{I}$, $\mathbf{D} = \mathrm{diag}\,(d_1, \ \cdots\ , d_n)$ and $\mathbf{K} = \mathrm{diag}\,(k_1, \ \cdots\ , k_n)$, the $(p,q)^{\text{th}}$ entry of the Gramian is

$$\mathcal{P}_{pq} = \frac{1}{2\pi}\int_{-\infty}^{\infty} (-\omega^2 + i\omega d_p + k_p)^{-1} b_p b_q^* (-\omega^2 - i\omega d_q^* + k_q^*)^{-1}\, d\omega.$$

In order to compute this integral, we make use of the following partial fraction expansion:

$$\frac{b_p}{s^2 + d_p s + k_p} = \frac{\alpha_p}{s + \gamma_p} + \frac{\beta_p}{s + \delta_p}.$$

Then

$$\begin{aligned}\mathcal{P}_{pq} &= \int_0^\infty \left[\alpha_p e^{-\gamma_p t} + \beta_p e^{-\delta_p t}\right] \left[\alpha_q^* e^{-\gamma_q^* t} + \beta_q e^{-\delta_q^* t}\right] dt \\ &= \frac{\alpha_p \alpha_q^*}{\gamma_p + \gamma_q^*} + \frac{\alpha_p \beta_q^*}{\gamma_p + \delta_q^*} + \frac{\beta_p \alpha_q^*}{\delta_p + \gamma_q^*} + \frac{\beta_p \beta_q^*}{\delta_p + \delta_q^*}.\end{aligned}$$

With this formula, it is possible to explicitly construct the required Gramian and diagonalize it. We set a tolerance of $\tau = 10^{-5}$, and truncated the second-order system to (a second-order system of) order k, such that $\sigma_{k+1}(\mathcal{P}) < \tau \cdot \sigma_1(\mathcal{P})$; the resulting reduced system has order $k = 51$.

$\|\mathbf{\Sigma}\|_{\mathcal{H}_2}^2$	`3.9303e+000`
$\|\hat{\mathbf{\Sigma}}\|_{\mathcal{H}_2}^2$	`3.9302e+000`
$\mathcal{H}_2$ norm of neglected system $\mathbf{C}_2\mathbf{\Lambda}_2\mathbf{C}_2^*$	`4.3501e-005`
κ	`2.8725e+002`
$\kappa \,\mathrm{trace}\,(\mathbf{\Lambda}_2)$	`1.7936e-003`
Relative error bound	`4.6743e-004`
Computed relative error $\frac{\|\mathcal{E}\|_{\mathcal{H}_2}^2}{\|\mathbf{\Sigma}\|_{\mathcal{H}_2}^2}$	`1.2196e-005`

These results indicate that the constant κ in (4.9) is of moderate size and that the bound gives a reasonable error prediction.

A graphical illustration of the frequency response of the reduced model (order 51) compared to full system (order 398) is shown in Figure 4.2.

4.4 Summary

We have presented a unified way of defining Gramians for structured systems, in particular, weighted, second-order, closed loop and unstable systems. The key is to start with the frequency domain. Consequently we examined the reduction of a high-order (structured) system based upon the dominant eigenspace of an appropriately defined Gramian, that preserves the high-order form. An error bound in the $\mathcal{H}_2$ norm for this reduction was derived. An equivalent definition of the Gramian was obtained through a Parseval relationship

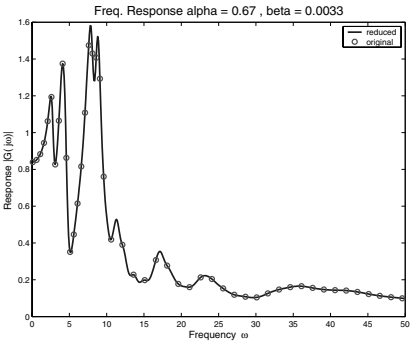


Fig. 4.2. Frequency response of reduced model (order 51) compared to full system (order 398).

and this was key to the derivation of the bound. Here, we just sketched the derivations. Full details and computational issues will be reported in the future.

Acknowledgements

This work was supported in part by the NSF through Grants DMS-9972591, CCR-9988393 and ACI-0082645.

References

1. A.C. Antoulas, *Approximation of large-scale dynamical systems*, SIAM Book series "Advances in Design and Control", Philadelphia (2004) (in press).
2. A.C. Antoulas and D.C. Sorensen, *Lanczos, Lyapunov and Inertia*, Linear Algebra and Its Applications, **326**: 137-150 (2001).
3. A.C. Antoulas, D.C. Sorensen, and S. Gugercin, *A survey of model reduction methods for large-scale systems*, Contemporary Mathematics, vol. 280, (2001), p. 193-219.
4. S.K. Godunov, *Modern aspects of linear algebra*, Translations of Mathematical Monographs, volume **175**, American Math. Society, Providence (1998).
5. S. Gugercin and A.C. Antoulas, *On balancing related model reduction methods and the corresponding error*, Int. Journal of Control, accepted for publication (2003).
6. D.C. Sorensen and A.C. Antoulas, *The Sylvester equation and approximate balanced reduction*, Linear Algebra and Its Applications. Fourth Special Issue on Linear Systems and Control, Edited by V. Blondel, D. Hinrichsen, J. Rosenthal, and P.M. van Dooren., **351-352**: 671-700 (2002).
7. D.G. Meyer and S. Srinivasan, *Balancing and model reduction for second-order form linear systems*, IEEE Trans. Automatic Control, **AC-41**: 1632-1644 (1996).

8. Y. Chahlaoui, D. Lemonnier, A. Vandendorpe, and P. Van Dooren, *Second-order structure preserving model reduction*, Proc. MTNS, Leuven (2004).
9. K. Chahlaoui, D. Lemonnier, K. Meerbergen, A. Vandendorpe, and P. Van Dooren, *Model reduction of second-order systems*, Proc. International Symposium Math. Theory. Netw. Syst., Paper 26984–4 (2002)

5

Model Reduction of Time-Varying Systems

Younes Chahlaoui[1] and Paul Van Dooren[2]

[1] School of Computational Science, Florida State University, Tallahassee, U.S.A.
younes.chahlaou@laposte.net
[2] CESAME, Université catholique de Louvain, Louvain-la-Neuve, Belgium
vdooren@csam.ucl.ac.be

Summary. This paper presents new recursive projection techniques to compute reduced order models of time-varying linear systems. The methods produce a low-rank approximation of the Gramians or of the Hankel map of the system and are mainly based on matrix operations that can exploit sparsity of the model. We show the practical relevance of our results with a few benchmark examples.

5.1 Introduction

The basic idea of model reduction is to represent a complex linear dynamical system by a much simpler one. This may refer to many different techniques, but in this paper we focus on projection-based model reduction of linear systems. It can be shown in the time-invariant case [GVV03] that projection methods allow to generate almost all reduced order models and that they are in that sense quite general. Here we construct the projection based on the dominant invariant subspaces of products of the Gramians, which are energy functions for ingoing and outgoing signals of the system. When the system matrices are large and sparse, the Gramians are nevertheless dense and efficient methods will therefore have to approximate these dominant spaces without explicitly forming the Gramians themselves.

Balanced Truncation [Moo81] is probably the most popular projection-based method. This is mainly due to its simplicity: the construction is based on simple linear algebra decompositions and there is no need to first choose a set of essential parameters. Moreover an a priori upper bound is given for the $\mathcal{H}_\infty$-norm of the error between the original plant and the reduced-order model [Enn81].

An important issue in model reduction is the choice of the order of the approximation, since it affects the quality of the approximation. One would like to be able to choose this during the construction of the reduced order model, i.e. without having to evaluate in advance quality measures like the Hankel singular values (computing them all would become prohibitive for large-scale

systems). The use of iterative methods seem appealing in this context since they may offer the possibility to perform order selection during the computation of the projection spaces and not in advance.

The approach that we propose in this paper is iterative and applies as well to time-varying systems. Earlier work on model reduction of time-varying systems was typically based on the explicit computation of the time-varying solution of a matrix difference (or differential) equation [SSV83, IPM92, SR02] and such results were mainly used to prove certain properties or bounds of the reduced order model. They were in other words not presented as an efficient computational tool. We propose to update at each step two sets of basis vectors that allow to identify the dominant states. The updating equations are cheap since they only require sparse matrix vector multiplications. The ideas are explained in Chapter 24 and [CV03a, CV03b, Cha03], to which we refer for proofs and additional details. Another recent approach is to use fast matrix decomposition methods on matrices with particular structure such as a Hankel structure. Such an approach is presented in [DV98] and could be competitive with the methods presented here.

5.2 Linear Time-Varying Systems

Linear discrete *time-varying systems* are described by systems of difference equations:

$$\mathcal{S}: \quad \begin{cases} x_{k+1} = A_k x_k + B_k u_k \\ y_k = C_k x_k + D_k u_k \end{cases} \tag{5.1}$$

with input $u_k \in \mathbb{R}^m$, state $x_k \in \mathbb{R}^N$ and output $y_k \in \mathbb{R}^p$. In this paper we will assume $m, p \ll N$, the input sequence to be square-summable (i.e. $\sum_{-\infty}^{\infty} u_k^T u_k \leq \infty$), $D_k = 0$, and the matrices $\{A_k\}_{-\infty}^{\infty}$, $\{B_k\}_{-\infty}^{\infty}$, and $\{C_k\}_{-\infty}^{\infty}$ to be bounded for all k. Using the recurrence (5.1) over several time steps, one obtains the state at step k in function of past inputs over the interval $[k_i, k-1]$:

$$x_k = \Phi(k, k_i) x_{k_i} + \sum_{i=k_i}^{k-1} \Phi(k, i+1) B_i u_i$$

where $\Phi(k, k_i) := A_{k-1} \dots A_{k_i}$ is the discrete *transition matrix* over time period $[k_i, k-1]$. The transition matrix has the following properties:

$$\begin{cases} \Phi(k_2, k_0) = \Phi(k_2, k_1)\Phi(k_1, k_0), \ k_0 \leq k_1 \leq k_2 \\ \Phi(k, k) \ = I_N \qquad \forall k. \end{cases}$$

We will assume the time-varying system $\mathcal{S}$ to be *asymptotically stable*, meaning

$$\forall k \geq k_i \quad \|\Phi(k, k_i)\| \leq c \cdot a^{(k-k_i)}, \text{ with } c > 0, \ 0 < a < 1.$$

The *Gramians* over intervals $[k_i, k-1]$ and $[k, k_f]$ are then defined as follows:

$$\mathcal{G}_c(k) = \sum_{i=k_i}^{k-1} \Phi(k, i+1) B_i B_i^T \Phi^T(k, i+1),$$

$$\mathcal{G}_o(k) = \sum_{i=k}^{k_f} \Phi^T(i, k) C_i^T C_i \Phi(i, k),$$

where k_i may be $-\infty$ and k_f may be $+\infty$. It follows from the identities

$$\Phi(k_1, k_2) = \Phi(k_1, k_2+1) A_{k_2} \quad \text{and} \quad \Phi(k_1+1, k_2) = A_{k_1} \Phi(k_1, k_2),$$

where $k_1 \geq k_2$, that these Gramians can also be obtained from the Stein recurrence formulas:

$$\mathcal{G}_c(k+1) = A_k \mathcal{G}_c(k) A_k^T + B_k B_k^T \quad \text{and} \quad \mathcal{G}_o(k) = A_k^T \mathcal{G}_o(k+1) A_k + C_k^T C_k, \tag{5.2}$$

with respective initial conditions

$$\mathcal{G}_c(k_i) = 0, \quad \mathcal{G}_o(k_f + 1) = 0.$$

Notice that the first equation evolves "forward" in time, while the second one evolves "backward" in time.

These Gramians can also be related to the input/output map in a particular window $[k_i, k_f]$. Let us at each instant k ($k_i < k < k_f$) restrict inputs to be nonzero in the interval $[k_i, k)$ (i.e. "the past") and let us consider the outputs in the interval $[k, k_f]$ (i.e. the "future"). The state-to-outputs and inputs-to-state maps on this window are then given by :

$$\underbrace{\begin{bmatrix} y_k \\ y_{k+1} \\ \vdots \\ y_{k_f} \end{bmatrix}}_{Y} = \begin{bmatrix} C_k \\ C_{k+1} A_k \\ \vdots \\ C_{k_f} \Phi(k_f, k) \end{bmatrix} \underbrace{\begin{bmatrix} B_{k-1} & A_{k-1} B_{k-2} & \dots & \Phi(k, k_i+1) B_{k_i} \end{bmatrix}}_{x(k)} \underbrace{\begin{bmatrix} u_{k-1} \\ u_{k-2} \\ \vdots \\ u_{k_i} \end{bmatrix}}_{U}.$$

The finite dimensional *Hankel matrix* $\mathcal{H}(k_f, k, k_i)$ mapping U to Y is defined as

$$\mathcal{H}(k_f, k, k_i) = \begin{bmatrix} C_k B_{k-1} & C_k A_{k-1} B_{k-2} & \dots & C_k \Phi(k, k_i+1) B_{k_i} \\ C_{k+1} A_k B_{k-1} & C_{k+1} A_k A_{k-1} B_{k-2} & & C_{k+1} \Phi(k+1, k_i+1) B_{k_i} \\ \vdots & & \ddots & \vdots \\ C_{k_f} \Phi(k_f, k) B_{k-1} & C_{k_f} \Phi(k_f, k-1) B_{k-2} & \dots & C_{k_f} \Phi(k_f, k_i+1) B_{k_i} \end{bmatrix}.$$

Notice that this matrix has at most rank N since $x(k) \in \mathbb{R}^N$ and that it factorizes as

$$\mathcal{H}(k_f,k,k_i)=\underbrace{\begin{bmatrix} C_k \\ C_{k+1}A_k \\ \vdots \\ C_{k_f}\Phi(k_f,k)\end{bmatrix}}_{\mathcal{O}(k_f,k)}\underbrace{\begin{bmatrix} B_{k-1} & A_{k-1}B_{k-2} & \dots & \Phi(k,k_i+1)B_{k_i}\end{bmatrix}}_{\mathcal{C}(k,k_i)} \quad (5.3)$$

where $\mathcal{O}(k_f,k)$ and $\mathcal{C}(k,k_i)$ are respectively the *observability* and the *controllability* matrices at instant k over the finite window $[k_i,k_f]$. They satisfy the recurrences

$$\mathcal{O}(k_f,k)=\begin{bmatrix} C_k \\ \mathcal{O}(k_f,k+1)A_k\end{bmatrix}, \quad \mathcal{C}(k+1,k_i)=\begin{bmatrix} B_k & A_k\mathcal{C}(k,k_i)\end{bmatrix} \quad (5.4)$$

evolving forward and backward in time, respectively. From these matrices one then constructs the Gramians and Hankel map via the identities

$$\begin{aligned}\mathcal{H}(k_f,k,k_i)&=\mathcal{O}(k_f,k)\mathcal{C}(k,k_i),\\ \mathcal{G}_c(k)&=\mathcal{C}(k,k_i)\mathcal{C}(k,k_i)^T,\\ \mathcal{G}_o(k)&=\mathcal{O}(k_f,k)^T\mathcal{O}(k_f,k).\end{aligned}$$

Notice that in the time-invariant case the above matrices become function only of the differences $k-k_i$ and k_f-k. In this case one typically chooses both quantities equal to $\tau:=(k_f-k_i)/2$, i.e. half the considered window length. In the time-invariant case it is also typical to consider the infinite window case, i.e. where $k_f=-k_i=\infty$.

5.3 Balanced Truncation

The method of *Balanced Truncation* is a very popular technique of model reduction for stable linear time-invariant systems because it has several appealing properties related to sensitivity, stability and approximation error [Moo81, ZDG95]. The extension to time-varying systems is again based on the construction of a new state-space coordinate system in which both Gramians are diagonal and equal [SSV83, VK83, SR02]. This is always possible when the system is uniformly controllable and observable over the considered interval [SSV83, VK83], meaning that the Gramians are uniformly bounded and have uniformly bounded inverses. It is then known that there exists a time-varying state space transformation T_k such that the Gramians $\hat{\mathcal{G}}_c(k):=T_k^{-1}\mathcal{G}_c(k)T_k^{-T}$ and $\hat{\mathcal{G}}_o(k):=T_k^T\mathcal{G}_o(k)T_k$ of the transformed system $\{T_{k+1}^{-1}A_kT_k, T_{k+1}^{-1}B_k, C_kT_k\}$, satisfy

$$T_k^{-1}\mathcal{G}_c(k)\mathcal{G}_o(k)T_k=\hat{\mathcal{G}}_c(k)\hat{\mathcal{G}}_o(k)=\Sigma^2(k), \quad 0<\Sigma(k)<\infty I.$$

One then partitions the matrix $\Sigma(k)$ into $\mathrm{diag}\{\Sigma_+(k),\Sigma_-(k)\}$ where $\Sigma_+(k)$ contains the n largest singular values of $\Sigma(k)$ and $\Sigma_-(k)$ the smallest ones. In

that coordinate system the truncated system $\{\hat{A}_k, \hat{B}_k, \hat{C}_k\}$ is just the system corresponding to the leading n columns and rows of the transformed system $\{T_{k+1}^{-1} A_k T_k, T_{k+1}^{-1} B_k, C_k T_k\}$. If we denote the first n columns of T_k by X_k and the first n rows of T_k^{-1} by Y_k^T then $Y_k^T X_k = I_n$ and

$$\{\hat{A}_k, \hat{B}_k, \hat{C}_k\} := \{Y_{k+1}^T A_k X_k, Y_{k+1}^T B_k, C_k X_k\}. \tag{5.5}$$

If for all k there is also a gap between the singular values of $\Sigma_+(k)$ and those of $\Sigma_-(k)$, then similar properties to the time-invariant case can be obtained, namely asymptotic stability and uniform controllability and observability of the truncated model [SSV83] and an error bound for the truncation error between both input/output maps in terms of the neglected singular values $\Sigma_-(k)$ or of related matrix inequalities (see [LB03, SR02] for a more detailed formulation).

Rather than computing the complete transformations T_k, one only needs to compute the matrices $X_k, Y_k \in \mathbb{R}^{N \times n}$ whose columns span the "dominant" left and right eigenvector spaces of the product $\mathcal{G}_c(k)\mathcal{G}_o(k)$ and normalize them such that $Y_k^T X_k = I_n$ to obtain the reduced model as given above. One can show that both Gramians are no longer required to be non-singular, and this can therefore be applied as well to the finite window case. In general, one can not even guarantee the gap property of the eigenvalues of the product of the Gramians.

In order to reduce the complexity of the model reduction procedure one can try to approximate the dominant left invariant subspaces X_k and Y_k by an iterative procedure which possibly exploits the sparsity of the original model $\{A_k, B_k, C_k\}$. The projection matrices will hopefully be close to invariant subspaces and one can hope to derive bounds for the approximation error between both systems. Such a procedure is explained in the next two sections and is inspired by efficient approximation techniques found in the time-invariant case [GSA03]. Bounds will be derived for the time-invariant version of this algorithm.

5.4 Recursive Low-Rank Gramian Algorithm (RLRG)

Large scale system models $\{A_k, B_k, C_k\}$ are often sparse and since the construction of a good approximate time-varying system model $\{\hat{A}_k, \hat{B}_k, \hat{C}_k\}$ requires an approximation at every time step k it seems crucial to find a method that is of low complexity at every time step and therefore exploits the sparsity of the original model.

If the Gramians $\mathcal{G}_c(k)$ and $\mathcal{G}_o(k)$ of the system $\{A_k, B_k, C_k\}$ were of rank $n << N$, for all $k \in [k_i, k_f]$ then the system would be actually of degree n. The idea is thus to replace

$$\mathcal{G}_c(k) = \mathcal{C}(k, k_i)\mathcal{C}(k, k_i)^T \quad \text{and} \quad \mathcal{G}_o(k) = \mathcal{O}(k_f, k)^T \mathcal{O}(k_f, k)$$

by semi-definite rank n_k approximations

$$\mathcal{P}_k := S_k S_k^T \quad \text{and} \quad \mathcal{Q}_k := R_k R_k^T,$$

respectively (for simplicity, we will assume n_k constant and equal to n). If such a *factorized approximation* is available, then

$$\mathcal{G}_c(k)\mathcal{G}_o(k) \approx S_k S_k^T R_k R_k^T$$

and the right hand side has clearly $X_k := S_k$ as right invariant subspace, and $Y_k := R_k$ as left invariant subspace. Normalizing X_k and Y_k such that $Y_k^T X_k = I_n$ will then yield an appropriate projected system (5.5) at each step k.

Note that the Gramian recurrences (5.2) evolve forward and backward in time and so will the recurrences for the approximations. We introduce the indices

$$l := k_i + i, \quad r := k_f + 1 - i$$

to simplify the indexing of the low-rank updating equations. At step i we compute the singular value decompositions of the matrices

$$\left[\, B_{l-1} \middle| A_{l-1} S_{l-1} \,\right] \quad \text{and} \quad \left[\frac{C_r}{R_{r+1}^T A_r}\right],$$

which yield transformation matrices $U := \left[\, U_+ \middle| U_- \,\right]$ and $V := \left[\, V_+ \middle| V_- \,\right]$ defining

$$\left[\, S_l \middle| E_c(l) \,\right] := \left[\, B_{l-1} \middle| A_{l-1} S_{l-1} \,\right] \left[\, V_+ \middle| V_- \,\right], \tag{5.6}$$

$$\left[\, R_r \middle| E_o(r) \,\right] := \left[\, C_r^T \middle| A_r^T R_{r+1} \,\right] \left[\, U_+ \middle| U_- \,\right], \tag{5.7}$$

where $V_+ \in \mathbb{R}^{(m+n)\times n}$ and $U_+ \in \mathbb{R}^{(p+n)\times n}$. These iterations are initialized at step $i = 0$ with

$$S_{k_i} = 0 \quad \text{and} \quad R_{k_f+1} = 0.$$

A MATLAB-like procedure corresponding to these recurrences would be as follows.

Algorithm RLRG
$l = k_i; r = k_f + 1; \tau = r - l - 1; S_l = 0; R_r = 0;$
for $i = 1 : \tau;$
  $l = l + 1; M = \left[\, B_{l-1} \middle| A_{l-1} S_{l-1} \,\right];$
  $[U, \Sigma, V] = svd(M, 0); S_l = M * V(:, 1 : n);$
  $r = r - 1; M = \left[\, C_r^T \middle| A_r^T R_{r+1} \,\right];$
  $[V, \Sigma, U] = svd(M, 0); R_r = M * U(:, 1 : n);$
end

At each iteration, we need to multiply $A_{l-1}S_{l-1}$ and $R_{r+1}^T A_r$ (which requires $4Nn\alpha$ flops, where α is the average number of nonzero elements in

each row or column of the sparse matrices A_i) and perform the transformations U and V (which require $O(N(n+m)^2)$ flops and $O(N(n+p)^2)$ flops, respectively [GV96]). When $N \gg n > m, p, \alpha$ this is altogether linear in the largest dimension N. Notice that the matrices S_{l-1} and R_{r+1} are multiplied at each step by time-varying matrices, which seems to preclude adaptive SVD updating techniques such as those used in [GSA03].

At each iteration step, $E_c(l)$ and $E_o(r)$ are neglected, which corresponds to the best rank n approximations at that step. But we would like to bound the global errors

$$\mathcal{E}_c(l) := \mathcal{G}_c(l) - \mathcal{P}_l = \mathcal{G}_c(l) - S_l S_l^T, \quad \text{and} \quad \mathcal{E}_o(r) := \mathcal{G}_o(r) - \mathcal{Q}_r = \mathcal{G}_o(r) - R_r R_r^T.$$

The following lemma [CV02] is proven in [Cha03] and leads to such bounds.

Lemma 5.4.1. *At each iteration, there exists orthogonal matrices*

$$V^{(i)} \in \mathbb{R}^{(n+im)\times(n+im)} \quad \textit{and} \quad U^{(i)} \in \mathbb{R}^{(n+ip)\times(n+ip)},$$

satisfying:

$$\mathcal{C}(l, k_i) V^{(i)} = \left[\, S_l \middle| E_c(l) \middle| A_{l-1} E_c(l-1) \middle| \ldots \middle| \Phi(l, k_i+1) E_c(k_i+1) \,\right],$$

and

$$\mathcal{O}(k_f, r)^T U^{(i)} = \left[\, R_r \middle| E_o(r) \middle| A_r^T E_o(r+1) \middle| \ldots \middle| \Phi(k_f, r)^T E_o(k_f) \,\right],$$

where $E_c(i)$ and $E_o(i)$ are the neglected parts at each iteration.

The above identities then lead to expressions for the errors:

$$\mathcal{E}_c(l) = \sum_{j=1}^{i} \Phi(l, k_i+j) E_c(k_i+j) E_c(k_i+j)^T \Phi(l, k_i+j)^T, \tag{5.8}$$

$$\mathcal{E}_o(r) = \sum_{j=0}^{i-1} \Phi(k_f - j, r)^T E_o(k_f - j) E_o(k_f - j)^T \Phi(k_f - j, r). \tag{5.9}$$

It is shown in [CV02, Cha03] that the norms of $\mathcal{E}_c(l)$ and $\mathcal{E}_o(r)$ can then be bounded in terms of

$$\eta_c(l) = \max_{k_i+1 \le j \le l} \|E_c(j)\|_2, \quad \text{and} \quad \eta_o(r) = \max_{r \le j \le k_f} \|E_o(j)\|_2,$$

which we refer to as the "noise" levels η_c and η_o of the recursive singular value decompositions (5.6,5.7).

Theorem 5.4.2. *If the system (5.1) is stable, i.e.,*

$$\|\Phi(k, k_0)\| \le c \cdot a^{(k-k_0)}, \text{ with } c > 0,\ 0 < a < 1,$$

then

$$\|\mathcal{E}_c(l)\|_2 \le \frac{\eta_c^2(l) c^2}{1-a^2}, \quad \textit{and} \quad \|\mathcal{E}_o(r)\|_2 \le \frac{\eta_o^2(r) c^2}{1-a^2}.$$

5.4.1 Time-Invariant Case

It is interesting to note that for linear time-invariant systems $\{A, B, C\}$, the differences $\mathcal{E}_c(l)$ and $\mathcal{E}_o(r)$ remain bounded for large i, and this shows the strength of Theorem 5.4.2. We then have the following result, shown in [CV02, Cha03].

Theorem 5.4.3. *Let P and Q be the solutions of*

$$P = APA^T + I, \quad \textit{and} \quad Q = A^T QA + I,$$

then

$$\begin{aligned} \|\mathcal{E}_c(l)\|_2 &\leq \eta_c^2(l)\|P\|_2 \leq \eta_c^2(l)\tfrac{\kappa(A)^2}{1-\rho(A)^2}, \\ \|\mathcal{E}_o(r)\|_2 &\leq \eta_o^2(r)\|Q\|_2 \leq \eta_o^2(r)\tfrac{\kappa(A)^2}{1-\rho(A)^2}, \end{aligned} \tag{5.10}$$

$$\|\mathcal{G}_c(l)\mathcal{G}_o(r) - \mathcal{P}_l\mathcal{Q}_r\|_2 \leq \frac{\kappa(A)^2}{1-\rho(A)^2}\left(\eta_c^2(l)\|\mathcal{G}_o(r)\|_2 + \eta_o^2(r)\|\mathcal{G}_c(l)\|_2\right), \tag{5.11}$$

where $\kappa(A)$ is the condition number and $\rho(A)$ is the spectral radius of A.

In [GSA03], bounds very similar to (5.10) were obtained but the results in that paper only apply to the time-invariant case. The bound (5.11) says that if one Gramian is not well approximated, the product of the Gramians, which is related to the Hankel singular values, will not be well approximated. Notice that this only makes sense when $l = r$. In the time-invariant case one can also estimate the convergence to the infinite horizon Gramians, which we denote by $\mathcal{G}_c$ and $\mathcal{G}_o$ and are defined by he identities

$$\mathcal{G}_c = A\mathcal{G}_c A^T + BB^T, \quad \text{and} \quad \mathcal{G}_o = A^T\mathcal{G}_o A + C^T C.$$

Theorem 5.4.4. *At each step i of (5.6,5.7) we have the following error bounds*

$$\begin{aligned} \|\mathcal{P}_{i-1} - \mathcal{G}_c\|_2 &\leq \|\mathcal{P}_i - \mathcal{P}_{i-1} + E_c(i)E_c^T(i)\|_2\|P\|_2 \\ &\leq \|\mathcal{P}_i - \mathcal{P}_{i-1} + E_c(i)E_c^T(i)\|_2\frac{\kappa(A)^2}{1-\rho(A)^2}, \\ \|\mathcal{Q}_{i+1} - \mathcal{G}_o\|_2 &\leq \|\mathcal{Q}_i - \mathcal{Q}_{i+1} + E_o(i)E_o^T(i)\|_2\|Q\|_2 \\ &\leq \|\mathcal{Q}_i - \mathcal{Q}_{i+1} + E_o(i)E_o^T(i)\|_2\frac{\kappa(A)^2}{1-\rho(A)^2}, \end{aligned}$$

where $\kappa(A)$ is the condition number and $\rho(A)$ is the spectral radius of A.

Proof. We prove the result only for $\mathcal{P}_{i-1}$ since both results are dual. Start from

$$\mathcal{P}_i + E_c(i)E_c^T(i) = A\mathcal{P}_{i-1}A^T + BB^T,$$

to obtain

$$(\mathcal{G}_c - \mathcal{P}_{i-1}) = A(\mathcal{G}_c - \mathcal{P}_{i-1})A^T + (\mathcal{P}_i - \mathcal{P}_{i-1} + E_c(i)E_c(i)^T).$$

Use the solution P of the linear system $P = APA^T + I$ and its growth factor $\frac{\kappa(A)^2}{1-\rho(A)^2}$ to obtain from there the desired bound. □

This theorem says that when convergence is observed, we can bound the accuracy of the current estimates of the Gramians in terms of quantities computed in the last step only. Using very different arguments, is was mentioned in [Cha03] that this in fact holds approximately for the time-varying case as well.

5.4.2 Periodic Case

The simplest class of time-varying models is the class of periodic systems. This is because every K-periodic system,

$$\{A_{K+k}, B_{K+k}, C_{K+k}\} = \{A_k, B_k, C_k\}$$

is in fact equivalent [MB75] to K *lifted* time-invariant systems:

$$\begin{cases} \hat{x}_{k+1}^{(h)} = \hat{A}^{(h)}\hat{x}_k^{(h)} + \hat{B}^{(h)}\hat{u}_k^{(h)} \\ \hat{y}_k^{(h)} = \hat{C}^{(h)}\hat{x}_k^{(h)} + \hat{D}^{(h)}\hat{u}_k^{(h)} \end{cases} \tag{5.12}$$

where the state $\hat{x}_k^{(h)} := x_{h+kK}$ evolves over K time steps with state transition matrix $\hat{A}^{(h)} := \Phi(h+K, h)$, where $\hat{u}_k^{(h)}$ and $\hat{y}_k^{(h)}$ are the stacked input and output vectors:

$$\hat{u}_k^{(h)} := [u_{h+kK}^T,\ u_{h+kK+1}^T,\ \ldots,\ u_{h+kK+K-1}^T]^T$$

$$\hat{y}_k^{(h)} := [y_{h+kK}^T,\ y_{h+kK+1}^T,\ \ldots,\ y_{h+kK+K-1}^T]^T$$

and where $\hat{B}^{(h)}$, $\hat{C}^{(h)}$ and $\hat{D}^{(h)}$ are defined in terms of the matrices $\{A_k, B_k, C_k\}$ (see [MB75]). Obviously, there are K such time invariant liftings for $h = 1, \ldots, K$, and each one has a transfer function. For such systems a theorem similar to Theorem 5.4.3 was obtained in [CV02, Cha03].

Theorem 5.4.5. *Let P and Q be the solutions of, respectively, $P = \tilde{A}P\tilde{A}^T + I_{KN}$ and $Q = \tilde{A}^T Q\tilde{A} + I_{KN}$, where*

$$\tilde{A} := \begin{pmatrix} 0 & \ldots & 0 & A_K \\ A_1 & 0 & \ldots & 0 \\ 0 & \ddots & \ddots & \vdots \\ 0 & \ldots & A_{K-1} & 0 \end{pmatrix} \quad \textit{and} \quad \begin{array}{c} P := \operatorname{diag}(P_1, \ldots, P_{K-1}, P_K) \\ Q := \operatorname{diag}(Q_1, \ldots, Q_{K-1}, Q_K) \end{array}$$

then

$$\|\mathcal{E}_c(l)\|_2 \leq \eta_c^2(l)\|P\|_2 \leq \eta_c^2(l)\frac{\kappa(\tilde{A})^2}{1-\rho(\tilde{A})^2},$$

$$\|\mathcal{E}_o(r)\|_2 \leq \eta_c^2(r)\|Q\|_2 \leq \eta_c^2(r)\frac{\kappa(\tilde{A})^2}{1-\rho(\tilde{A})^2}.$$

Using multirate sampling [TAS01], we constructed in [CV02] a time-varying system model of period $K = 2$ and dimension $N = 122$ of the arm of the CD player described in Chapter 24, Section 4 of this volume. We refer to [CV02] for more details but we recall here some results illustrating the convergence of the Gramian estimates $\mathcal{P}_k = S_k S_k^T$, which were chosen of rank 20. Every two steps these should converge to the steady state solutions corresponding to the even and odd infinite horizon controllability Gramians.

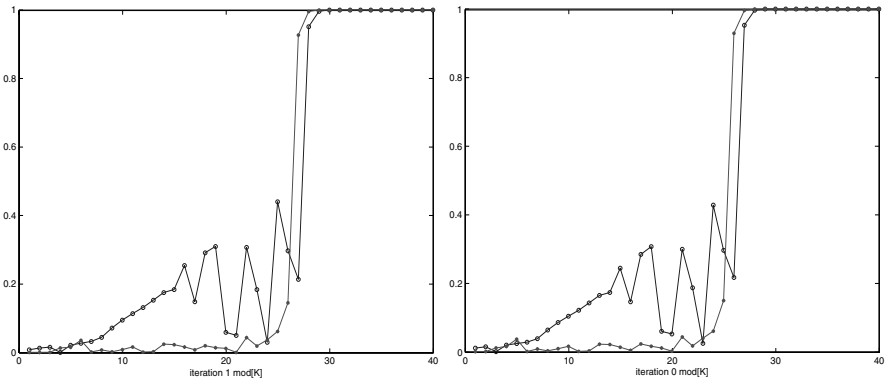


Fig. 5.1. $\circ$: $\cos(\measuredangle(S_k, S_{k-2}))$, $*$: $\cos(\measuredangle(S_k, S_\infty))$ for odd and even k

Since only the spaces matter and not the actual matrices, we show in Figure 5.1 (left) the cosine of the canonical angle between the dominant subspace of odd iterations $(k-2)$ and k, i.e. $\cos(\measuredangle(S_{k-2}, S_k))$, and the canonical angle with the exact dominant subspace, denoted as S_∞, of the controllability Gramian of the lifted LTI system (5.12), i.e. $(\cos(\measuredangle(S_k, S_\infty))$. This is repeated in Figure 5.1 (right) for the even iterates. The results for the observability Gramians are similar and are not shown here. Figure 5.1 shows the convergence and the accuracy of our algorithm. It can be seen that convergence is quick and is well predicted by the errors performed in the last updating steps.

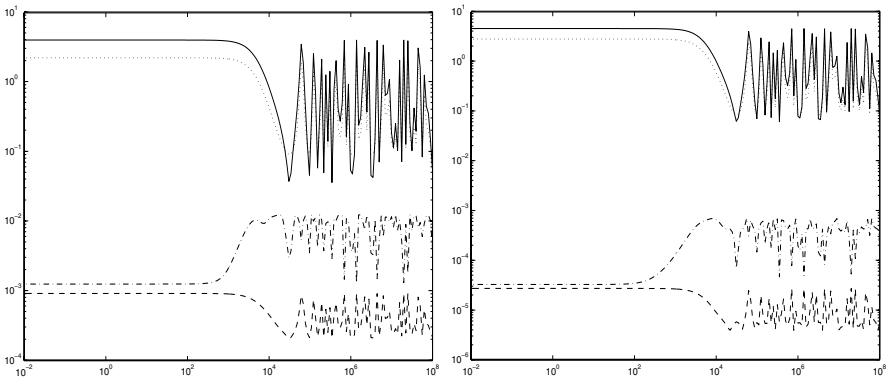


Fig. 5.2. –: full model, $\cdots$: approx. errors (20 steps), -·- approx. errors (60 steps), - - approx.errors (exact Gramian)

In Figure 5.2 we compare frequency responses of the time-invariant lifted systems (5.12) for odd and even iterates. In each figure we give the amplitude of the frequency response of the original model, the absolute errors in the frequency response of the projected systems using projectors obtained after 20 steps and 60 steps, and the absolute errors in the frequency response of the projected systems using the exact dominant subspace of the Gramians of the lifted system. The graphs show that after 60 steps an approximation comparable to Balanced Truncation is obtained.

5.5 Recursive Low-Rank Hankel Algorithm (RLRH)

The algorithm of the previous section yields an independent approximation of the two Gramians. If the original system was poorly balanced, it often happens that the approximation of the product of the two Gramians is far less accurate than that of the individual Gramians. This will affect the quality of the approximation of the reduced model since the product of the Gramians plays an important role in the frequency domain error.

In [CV03a, CV03b] an algorithm is presented which avoids this problem. The key idea is to use the underlying recurrences defining the time-varying Hankel map $\mathcal{H}(k_f, k, k_i) = \mathcal{O}(k_f, k)\mathcal{C}(k, k_i)$. Because the system order at each instant is given by the rank of the Hankel matrix at that instant, it is a good idea to approximate the system by approximating the Hankel matrix via a recursive SVD performed at each step. The technique is very similar to that of the previous section but now we perform at each step the singular value decomposition of a product similar to the products $\mathcal{O}(k_f, k)\mathcal{C}(k, k_i)$. Consider indeed the singular value decomposition of the matrix

$$\left[\frac{C_r}{R_{r+1}^T A_r}\right] \cdot \left[\, B_{l-1} \middle| A_{l-1} S_{l-1} \,\right] = U \Sigma V^T \tag{5.13}$$

and partition $U := \left[\, U_+ \middle| U_- \,\right]$, $V := \left[\, V_+ \middle| V_- \,\right]$ where $U_+ \in \mathbb{R}^{(p+n)\times n}$ and $V_+ \in \mathbb{R}^{(m+n)\times n}$. Define then

$$\left[\, S_l \middle| E_c(l) \,\right] := \left[\, B_{l-1} \middle| A_{l-1} S_{l-1} \,\right] \left[\, V_+ \middle| V_- \,\right], \tag{5.14}$$

$$\left[\, R_r \middle| E_o(r) \,\right] := \left[\, C_r^T \middle| A_r^T R_{r+1} \,\right] \left[\, U_+ \middle| U_- \,\right]. \tag{5.15}$$

It then follows that

$$\left[\frac{R_r^T}{E_o^T(r)}\right] \left[\, S_l \middle| E_c(l) \,\right] = \left[\begin{array}{c|c} \Sigma_+ & 0 \\ \hline 0 & \Sigma_- \end{array}\right], \tag{5.16}$$

where Σ_- contains the neglected singular values at this step. For the initialization at step $i = 0$ we use again

$$S_{k_i} = 0 \quad \text{and} \quad R_{k_f+1} = 0$$

and iterate for $i = 1, \ldots, \tau$ where $\tau := (k_f - k_i)/2$ is the half interval length. The approximate factorizations that one obtains are those indicated in Figure 5.3 and the corresponding MATLAB-like algorithm is now as follows.

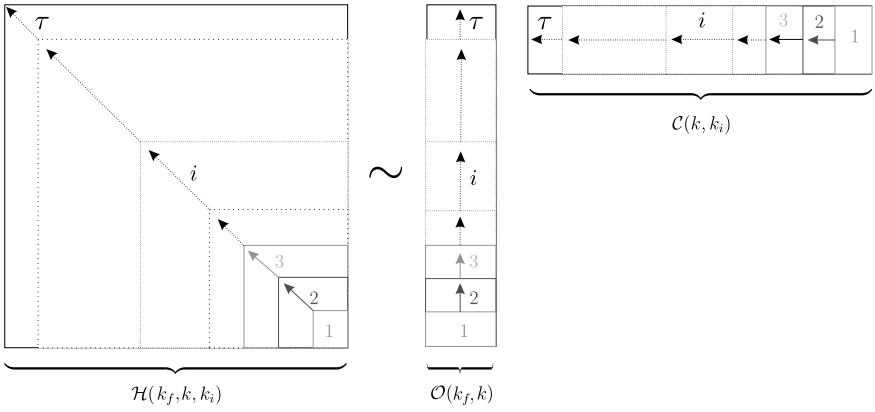


Fig. 5.3. Submatrix sequence approximated by low rank approximations

Algorithm RLRH
$l = k_i; r = k_f + 1; \tau = (r - l - 1)/2; S_l = 0; R_r = 0;$
for $i = 1 : \tau;$
$\quad l = l + 1; M = \left[\, B_{l-1} \middle| A_{l-1} S_{l-1} \,\right]; r = r - 1; N = \left[\, C_r^T \middle| A_r^T R_{r+1} \,\right];$
$\quad [U, \Sigma, V] = svd(N^T M); S_l = M * V(:, 1 : n); R_r = N * U(:, 1 : n);$
end

The amount of work involved in this algorithm is comparable to the earlier algorithm. We need to form the products $A_{l-1}S_{l-1}$ and $R_{r+1}^T A_r$, which requires $4Nn\alpha$ flops. The construction of the left hand side of (5.13) requires an additional $2N(n+m)(n+p)$ flops and the application of the transformations U and V requires $O((p+n)(m+n)(2n+p+m))$ flops, and so the complexity of this algorithm is $O(N(p+n)(m+n))$ for each iteration if $N \gg n > m, p, \alpha$.

As before we have a lemma, shown in [CV03a, CV03b, Cha03], linking the intermediate error matrices and the matrices $\mathcal{O}(k_f, r)$ and $\mathcal{C}(l, k_i)$.

Theorem 5.5.1. *At each iteration, there exist orthogonal matrices* $V^{(i)} \in \mathbb{R}^{(n+im)\times(n+im)}$ *and* $U^{(i)} \in \mathbb{R}^{(n+ip)\times(n+ip)}$ *satisfying:*

$$\mathcal{C}(l, k_i)V^{(i)} = \left[\, S_l \middle| E_c(l) \middle| A_{l-1}\mathcal{C}_e(l, k_i + 1) \,\right]$$

$$\mathcal{O}(k_f, r)^T U^{(i)} = \left[\, R_r \middle| E_o(r) \middle| A_r^T \mathcal{O}_e(k_f, r + 1) \,\right]$$

where $E_c(l)$ *and* $E_o(r)$ *are the neglected parts at each iteration, and the matrices* $\mathcal{C}_e(j, k_i)$ *and* $\mathcal{O}_e(k_f, j)$ *are defined as follows:*

$$\mathcal{C}_e(j, k_i) := \left[\, E_c(j-1) \middle| \ldots \middle| \Phi(j-1, k_i)E_c(k_i) \,\right],$$

$$\mathcal{O}_e(k_f, j)^T := \left[\, E_o(j) \middle| \ldots \middle| \Phi(k_f, j)^T E_o(k_f) \,\right].$$

As a consequence of this theorem we show in [CV03a, CV03b, Cha03] the following result which yields an approximation of the original Hankel map $\mathcal{H}(k_f, k, k_i)$.

Theorem 5.5.2. *There exist orthogonal matrices* $V^{(\tau)} \in \mathbb{R}^{(n+\tau m)\times(n+\tau m)}$ *and* $U^{(\tau)} \in \mathbb{R}^{(n+\tau p)\times(n+\tau p)}$ *such that* $U^{(\tau)T}\mathcal{H}(k_f, k, k_i)V^{(\tau)}$ *is equal to*

$$\left[\begin{array}{c|c|c} R_\tau^T S_\tau & 0 & R_\tau^T A_{\tau-1}\mathcal{C}_e(\tau, k_i) \\ \hline 0 & E_o^T(\tau)E_c(\tau) & E_o^T(\tau)A_{\tau-1}\mathcal{C}_e(\tau, k_i) \\ \hline \mathcal{O}_e(k_f, \tau+1)A_\tau S_\tau & \mathcal{O}_e(k_f, \tau+1)A_\tau E_c(\tau) & \mathcal{O}_e(k_f, \tau+1)A_\tau A_{\tau-1}\mathcal{C}_e(\tau, k_i) \end{array}\right].$$

This result enables us to evaluate the quality of our approximations by using the Hankel map without passing via the Gramians, which is exploited in [CV03a, CV03b, Cha03] to obtain bounds for the error. Notice also that since we are defining projectors for finite time windows, these algorithms could be applied to linear time-invariant systems that are unstable. One can then not show any property of stability for the reduced order model, but the finite horizon Hankel map will at least be well approximated.

5.5.1 Time-Invariant Case

As for the Gramian based approximation, we can analyze the quality of this approach in the time-invariant case. Since all matrices A, B and C are then

constant, all Hankel maps are time-invariant as well and only the interval width plays a role in the obtained decomposition. We can e.g. run the RLRH algorithm on an interval $[k_i, k_f] = [-\tau, \tau]$ for $\tau \in \mathbb{N}$ and approximate the Gramians $\mathcal{G}_c(0)$ and $\mathcal{G}_o(0)$ of the original model by $S_0 S_0^T$ and $R_0 R_0^T$, respectively, at the origin of the symmetric interval $[-\tau, \tau]$. The differences between the approximate low-rank Gramians and the exact Gramians

$$\mathcal{E}_c(0) := \mathcal{G}_c(0) - \mathcal{P}_0, \quad \mathcal{E}_o(0) := \mathcal{G}_o(0) - \mathcal{Q}_0$$

then remain bounded for intervals of growing length 2τ, as indicated in the following theorem ([CV03a, CV03b, Cha03]).

Theorem 5.5.3. *Let P and Q be respectively the solutions of $P = APA^T + I$, and $Q = A^T QA + I$, then*

$$\|\mathcal{E}_c(0)\|_2 \leq \eta_c^2 \|P\|_2 \leq \eta_c^2 \frac{\kappa(A)^2}{1-\rho(A)^2}, \quad \|\mathcal{E}_o(0)\|_2 \leq \eta_o^2 \|Q\|_2 \leq \eta_o^2 \frac{\kappa(A)^2}{1-\rho(A)^2}$$

where $\eta_c := \max\limits_{-\tau \leq k \leq 0} \|E_c(k)\|_2$ *and* $\eta_o := \max\limits_{0 \leq k \leq \tau} \|E_o(k)\|_2$.

Similarly, we obtain an approximation of the Hankel map as follows (see [CV03a, CV03b, Cha03]).

Theorem 5.5.4. *Using the first n columns $U_+^{(0)}$ of $U^{(0)}$ and $V_+^{(0)}$ of $V^{(0)}$, we obtain a rank n approximation of the Hankel map:*

$$\mathcal{H}(\tau, 0, -\tau) - U_+^{(0)} R_0^T \cdot S_0 V_+^{(0)T} = \mathcal{E}_h(0),$$

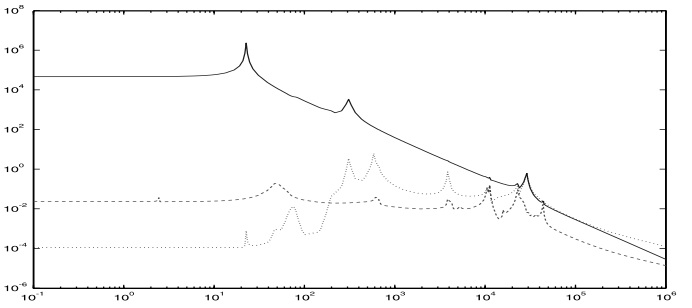
for which we have the error bound:

$$\|\mathcal{E}_h(0)\|_2 \leq \frac{\kappa(A)}{\sqrt{1-\rho(A)^2}} \max\{\eta_c \|R_0^T A\|_2, \eta_o \|A S_0\|_2\} + \frac{\kappa(A)^2}{1-\rho(A)^2} \eta_o \eta_c.$$

An important advantage of the RLRH method is that the computed projectors are independent of the coordinate system used to describe the original system $\{A, B, C\}$. This can be seen as follows. When performing a state-space transformation T we obtain a new system $\{\hat{A}, \hat{B}, \hat{C}\} := \{T^{-1}AT, T^{-1}B, CT\}$. It is easy to see that under such transformations the updating equations of R_r and S_l transform to $\hat{R}_k = T^T R_k$ and $\hat{S}_l = T^{-1} S_l$, and this is preserved by the iteration. One shows that the constructed projector therefore follows the same state-space transformation as the system model. Therefore, the constructed reduced order model does not depend on whether or not one starts with a balanced realization for the original system. For the RLRG method, on the other hand, one can lose a lot of accuracy when using a poorly balanced realization to construct a reduced order model.

5.6 Numerical Examples

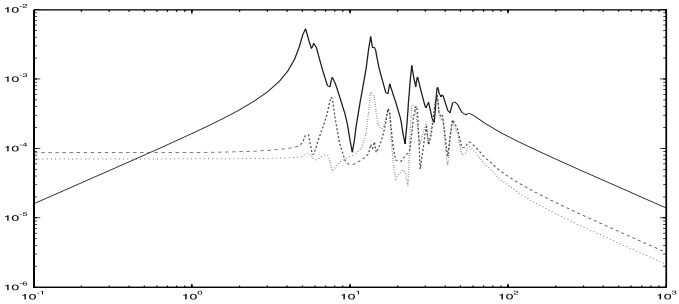
In this section we apply our algorithm to discretizations of three different dynamical systems: a Building model, a CD Player model, and the International Space Station model. These benchmarks are described in more details in Chapter 24, Sections 4, 6, 7. It was shown in [CV03a, Cha03], that for the same problem, the RLRG method gives less accurate results: as predicted by the discussion of the previous section, the RLRG method deteriorates especially when the original system is poorly balanced. Since the RLRH method is to be preferred over the RLRG method, we only compare here the RLRH method with Balanced Truncation. The approximate system $\mathcal{S}_{BT}$ for balanced truncation and $\mathcal{S}_{RLRH}$ for the recursive low rank Hankel method, are both calculated for a same degree. We show the maximal singular value of the frequency responses of the system and the maximal singular value of the two error functions.



σ_{max}-plot of the frequency responses.
— full model, - - - BT error system, ··· RLRH error system.

$cond(T)$	$\rho(A)$	$cond(A)$	$\|\mathcal{S}\|_{\mathcal{H}_\infty}$	$\|\mathcal{S}-\mathcal{S}_{BT}\|_{\mathcal{H}_\infty}$	$\|\mathcal{S}-\mathcal{S}_{RLRH}\|_{\mathcal{H}_\infty}$
40.7341	1	1.00705	$2.3198.10^6$	0.2040	6.1890

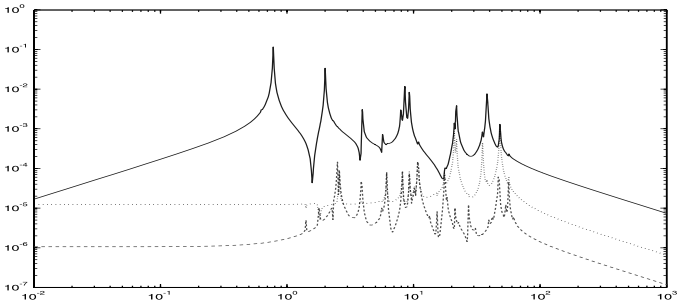
Fig. 5.4. CD-player model $N = 120, \quad m = p = 2, \quad n = 24$



σ_{max}-plot of the frequency responses.
— full model, - - - BT error system, ⋯ RLRH error system.

$cond(T)$	$\rho(A)$	$cond(A)$	$\|\mathcal{S}\|_{\mathcal{H}_\infty}$	$\|\mathcal{S}-\mathcal{S}_{BT}\|_{\mathcal{H}_\infty}$	$\|\mathcal{S}-\mathcal{S}_{RLRH}\|_{\mathcal{H}_\infty}$
347.078	0.9988	5.8264	0.0053	$6.0251.10^{-4}$	$6.7317.10^{-4}$

Fig. 5.5. Building model $N = 48, \quad m = p = 1, \quad n = 10$



σ_{max}-plot of the frequency responses.
— full model, - - - BT error system, ⋯ RLRH error system.

$cond(T)$	$\rho(A)$	$cond(A)$	$\|\mathcal{S}\|_{\mathcal{H}_\infty}$	$\|\mathcal{S}-\mathcal{S}_{BT}\|_{\mathcal{H}_\infty}$	$\|\mathcal{S}-\mathcal{S}_{RLRH}\|_{\mathcal{H}_\infty}$
740178	0.9998	5.82405	0.1159	$2.3630.10^{-4}$	0.0011

Fig. 5.6. ISS model $N = 270, \quad m = p = 3, \quad n = 32$

The corresponding H_∞ norms are also given in the table following each example. Each table also contains the condition number $cond(T)$ of the balancing state-space transformation T, the spectral radius $\rho(A)$ and the condition number $cond(A)$ since they play a role in the error bounds obtained in this paper. It can be seen from these examples that the RLRH method performs reasonably well in comparison to the balanced truncation method, and this independently from whether or not the original system was poorly balanced. Even though these models are not large they are good benchmarks in the sense

that their transfer functions are not easy to approximate. Larger experiments are reported in [Cha03].

5.7 Conclusion

In this paper we show how to construct low-dimensional projected systems of time-varying systems. The algorithms proposed are based on low-rank approximations of the Gramians and of the Hankel map which defines the input-output mapping. Both methods have the advantage of exploiting sparsity in the data to yield a complexity that is linear in the state dimension of the original model.

The key idea is to compute only a finite window of the Gramians or Hankel map of the time-varying system and to compute recursively projection matrices that capture the dominant behavior of the Gramians or Hankel map. The *Recursive Low-Rank Hankel* approximation method is to be preferred over the *Recursive Low-Rank Gramian* approximation method because it is not sensitive to the coordinate system in which the original system is described.

The two algorithms are mainly meant for time-varying systems but their performance is illustrated using time-invariant and periodic systems because the quality of the methods can then be assessed by the frequency responses of the error functions.

Acknowledgments

This paper presents research supported by NSF contracts CCR-99-12415 and ITR ACI-03-24944 and by the Belgian Programme on Inter-university Poles of Attraction, initiated by the Belgian State, Prime Minister's Office for Science, Technology and Culture. The scientific responsibility rests with its authors. The work of the first author has been partially carried out within the framework of a collaboration agreement between CESAME (Université Catholique de Louvain, Belgium) and LINMA of the Faculty of Sciences (Université Chouaib Doukkali, Morocco), funded by the Secretary of the State for Development Cooperation and by the CIUF (Conseil Interuniversitaire de la Communauté Française, Belgium).

References

[Cha03] Chahlaoui, Y.: *Recursive low rank Hankel approximation and model reduction.* Doctoral Thesis, Université catholique de Louvain, Louvain-la-Neuve (2003)

[CV02] Chahlaoui, Y. and Van Dooren, P.: Estimating Gramians of large-scale time-varying systems. In: *Proc. IFAC World Congress*, Barcelona, Paper 2440 (2002)

[CV03a] Chahlaoui, Y. and Van Dooren, P.: Recursive Gramian and Hankel map approximation of large dynamical systems. In: *CD-Rom Proceedings SIAM Applied Linear Algebra Conference*, Williamsburg, Paper MS14-1 (2003)

[CV03b] Chahlaoui, Y. and Van Dooren, P.: Recursive low rank Hankel approximation and model reduction. In: *CD-Rom Proceedings ECC 2003*, Cambridge, Paper 553 (2003)

[DV98] Dewilde, P. and van der Veen, A.-J.: *Time-varying systems and computations*. Kluwer Academic Publishers, Boston (1998)

[Enn81] Enns, D.: Model reduction with balanced realizations: An error bound and frequency weighted generalization. In: *Proc. of the IEEE Conference on Decision and Control*, San Diego, 127–132 (1981)

[GVV03] Gallivan, K., Vandendorpe, A. and Van Dooren, P.: Sylvester equations and projection-based model reduction. *J. Comp. Appl. Math.*, **162**, 213–229 (2003)

[GV96] Golub, G. and Van Loan, C.: *Matrix Computations*. Johns Hopkins University Press, Baltimore (1996)

[GSA03] Gugercin, S., Sorenson, D. and Antoulas, A.: A modified low-rank Smith method for large-scale Lyapunov equations. *Numerical Algorithms*, **32(1)**, 27–55 (2003)

[IPM92] Imae, J., Perkins, J.E. and Moore, J.B.: Toward time-varying balanced realization via Riccati equations. *Math. Control Signals Systems*, **5**, 313–326 (1992)

[LB03] Lall, S., and Beck, C.: Error-bounds for balanced model-reduction of linear time-varying systems. *IEEE Trans. Automat. Control*, **48(6)**, 946–956 (2003)

[MB75] Meyer, R. and Burrus, C.: A unified analysis of multirate and periodically time-varying digital filters. *IEEE Trans. Circ. Systems*, **22**, 162–168 (1975)

[Moo81] Moore, B.: Principal component analysis in linear systems: controllability, observability, and model reduction. *IEEE Trans. Automat. Control*, **26**, 17–31 (1981)

[SR02] Sandberg, H., and Rantzer, H.: Balanced model reduction of linear time-varying systems. In: *Proc. IFAC02, 15th Triennial World Congress*, Barcelona (2002)

[SSV83] Shokoohi, S., Silverman, L., and Van Dooren, P.: Linear time-variable systems: Balancing and model reduction. *IEEE Trans. Automat. Control*, **28**, 810–822 (1983)

[TAS01] Tornero, J., Albertos, P., and Salt, J.: Periodic optimal control of multirate sampled data systems. In: *Proc. PSYCO2001, IFAC Conf. Periodic Control Systems*, Como, 199–204 (2001)

[VK83] Verriest, E., and Kailath, T.: On generalized balanced realizations. *IEEE Trans. Automat. Control*, **28(8)**, 833–844 (1983)

[ZDG95] Zhou, K., Doyle, J., and Glover, K.: *Robust and optimal control*. Prentice Hall, Upper Saddle River (1995)

6

Model Reduction of Second-Order Systems

Younes Chahlaoui[1], Kyle A. Gallivan[1], Antoine Vandendorpe[2], and Paul Van Dooren[2]

[1] School of Computational Science, Florida State University, Tallahassee, U.S.A.
younes.chahlaoui@laposte.net, gallivan@csit.fsu.edu
[2] CESAME, Université catholique de Louvain, Louvain-la-Neuve, Belgium,
vandendorpe@csam.ucl.ac.be, vdooren@csam.ucl.ac.be

6.1 Introduction

In this chapter, the problem of constructing a reduced order system while preserving the second-order structure of the original system is discussed. After a brief introduction on second-order systems and a review of first order model reduction techniques, two classes of second-order structure preserving model reduction techniques – Krylov subspace-based and SVD-based – are presented. For the Krylov techniques, conditions on the projectors that guarantee the reduced second-order system tangentially interpolates the original system at given frequencies are derived and an algorithm is described. For SVD-based techniques, a Second-Order Balanced Truncation method is derived from *second-order Gramians*.

Second-order systems arise naturally in many areas of engineering (see, for example, [Pre97, WJJ87]) and have the following form:

$$\begin{cases} M\ddot{q}(t) + D\dot{q}(t) + Sq(t) = F^{in}\ u(t), \\ \qquad\qquad\qquad\quad y(t) = F^{out}\ q(t). \end{cases} \tag{6.1}$$

We assume that $u(t) \in \mathbb{R}^m$, $y(t) \in \mathbb{R}^p$, $q(t) \in \mathbb{R}^N$, $F^{in} \in \mathbb{R}^{N\times m}$, $F^{out} \in \mathbb{R}^{p\times N}$, and $M, D, S \in \mathbb{R}^{N\times N}$ with M invertible. For *mechanical systems* the matrices M, D and S represent, respectively, the *mass* (or *inertia*), *damping* and *stiffness* matrices, $u(t)$ corresponds to the vector of *external* forces, F^{in} to the input distribution matrix, $y(t)$ to the output measurement vector, F^{out} to the output measurement matrix, and $q(t)$ to the vector of *internal generalized coordinates*.

The transfer function associated with the system (6.1) links the outputs to the inputs in the Laplace domain and is given by

$$R(s) := F^{out}P(s)^{-1}F^{in}, \tag{6.2}$$

where

$$P(s) := Ms^2 + Ds + S \tag{6.3}$$

is the *characteristic polynomial matrix*. The zeros of $\det(P(s))$ are also known as the *characteristic frequencies* of the system and play an important role in model reduction, e.g., the system is stable if these zeros lie in the open left half plane.

Often, the original system is too large to allow the efficient solution of various control or simulation tasks. In order to address this problem, techniques that produce a reduced system of size $n \ll N$ that possesses the essential properties of the full order model have been developed. Such a reduced model can then be used effectively, e.g., in real-time, for controlling or simulating the phenomena described by the original system under various types of external forces $u(t)$. We therefore need to build a reduced model,

$$\begin{cases} \hat{M}\ddot{\hat{q}}(t) + \hat{D}\dot{\hat{q}}(t) + \hat{S}\hat{q}(t) = \hat{F}^{in}u(t) \\ \hat{y}(t) = \hat{F}^{out}\hat{q}(t) \end{cases} \tag{6.4}$$

where $\hat{q}(t) \in \mathbb{R}^n$, $\hat{M}$, $\hat{D}$, $\hat{S} \in \mathbb{R}^{n\times n}$, $\hat{F}^{in} \in \mathbb{R}^{n\times m}$, $\hat{F}^{out} \in \mathbb{R}^{p\times n}$, such that its transfer function is "close" to the original transfer function.

In contrast with second-order systems, first order systems can be represented as follows:

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) \end{cases} \tag{6.5}$$

where again $u(t) \in \mathbb{R}^m$, $y(t) \in \mathbb{R}^p$, $x(t) \in \mathbb{R}^N$, $C \in \mathbb{R}^{p\times N}$, $A \in \mathbb{R}^{N\times N}$ and $B \in \mathbb{R}^{N\times m}$.

The transfer function associated with the system (6.5) is given by

$$R(s) := C(sI_N - A)^{-1}B. \tag{6.6}$$

Second-order systems can be seen as a particular class of linear systems. Indeed, since the mass matrix M is assumed to be invertible, we can rewrite the system (6.1) as follows

$$\begin{cases} \dot{x}(t) = \begin{bmatrix} 0 & I_N \\ -S_M & -D_M \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ F_M^{in} \end{bmatrix} u(t) \\ \\ y(t) = \begin{bmatrix} F_M^{out} & 0 \end{bmatrix} x(t) \end{cases} \tag{6.7}$$

where the state $x(t)$ is $\begin{bmatrix} q(t)^T & \dot{q}(t)^T \end{bmatrix}^T$, and where $S_M = M^{-1}S$, $D_M = M^{-1}D$, $F_M^{in} = M^{-1}F^{in}$, $F_M^{out} = F^{out}$, which is of the form (6.5). We can thus rewrite the transfer function defined in (6.2) as

$$R(s) = C(sI_{2N} - A)^{-1}B \tag{6.8}$$

with

$$A := \begin{bmatrix} 0 & I_N \\ -S_M & -D_M \end{bmatrix} \quad , \quad B := \begin{bmatrix} 0 \\ F_M^{in} \end{bmatrix} \quad , \quad C := \begin{bmatrix} F_M^{out} & 0 \end{bmatrix}. \tag{6.9}$$

Note that if the dimension of the state $q(t)$ of the original second-order system (6.1) is equal to N, the order of its corresponding linearized state space realization (6.9) (also called the McMillan degree of $R(s)$ if the state space realization (C, A, B) is minimal) is equal to $2N$.

A reduced model for the second-order system (6.1) could be produced by applying standard linear model reduction techniques to (C, A, B) in (6.9) to yield a small linear system $(\hat{C}, \hat{A}, \hat{B})$. Unfortunately, there is no guarantee that the matrices defining the reduced system $(\hat{C}, \hat{A}, \hat{B})$ have the block structure necessary to preserve the second-order form of the original system. Such a guarantee requires the development of second-order structure-preserving model reduction techniques.

This chapter is organized as follows. In Section 6.2, general results concerning model reduction of first order systems are summarized. In Section 6.3, a simple sufficient condition for constructing reduced order systems that preserve the second-order structure is developed. Generalizations of Balanced Truncation and Krylov subspace-based methods that enforce this sufficient condition for second-order systems are presented in Sections 6.4 and 6.5, respectively. After some numerical experiments in Section 6.6, concluding remarks are given in Section 6.7.

6.2 Model Reduction of Linear Systems

Most popular model reduction techniques for linear systems can be put in one of two categories [Ant05]: SVD-based and Krylov subspace-based techniques. Perhaps the most popular model reduction technique for linear systems is the Balanced Truncation method. This SVD-based technique has many advantages: the stability of the original system is preserved and there exists an a priori global bound on the error between the original and the reduced system. The main drawback is that this technique cannot be applied to large-scale systems of order N, i.e., those systems where $O(N^3)$ computations is an unacceptably large cost. On the other hand, Krylov subspace-based techniques that are based on imposing moment matching conditions between the original and the reduced transfer function, such as rational/tangential interpolation methods, can be applied to large-scale systems but do not provide global error bounds and depend significantly on the choice of certain parameters.

In this section, we present an overview of examples of each category applied to a linear system described by (6.5). The corresponding transfer functions is then *strictly proper*, i.e. $\lim_{s\to\infty} R(s) = 0$. Since M is invertible, the transfer function considered in (6.2) is also strictly proper.

6.2.1 Balanced Truncation

If A is stable, then the system (6.5) is also a linear (convolution) operator mapping square integrable inputs $u(t) \in \mathcal{L}_2[-\infty, +\infty]$ to square integrable

outputs $y(t) \in \mathcal{L}_2[-\infty, +\infty]$. Following the development in [CLVV05], we recall the concept of a dual operator to discuss the *Balanced Truncation* method.

Definition 6.2.1. *Let L be a linear operator acting from a Hilbert space U to a Hilbert space Y equipped respectively with the inner products $< , >_U$ and $< , >_Y$. The dual of L, denoted by L^*, is defined as the linear operator acting from Y to U such that $< Lu, y >_Y = < u, L^*y >_U$ for all $y \in Y$ and all $u \in U$.* □

It is easily verified that the transfer function associated with the dual operator of (6.6) is $B^T(sI_N - A^T)^{-1}C^T$, (see [ZDG95]).

Now consider the input/output behavior of the system (6.5). If we apply an input $u(t) \in \mathcal{L}_2[-\infty, 0]$ to the system for $t < 0$, the position of the state at time $t = 0$, assuming the zero initial condition $x(-\infty) = 0$, is equal to

$$x(0) = \int_{-\infty}^{0} e^{-At} B u(t) dt := \mathcal{C}_o u(t).$$

If a zero input is applied to the system for $t > 0$, then for all $t \geq 0$, the output $y(t) \in \mathcal{L}_2[0, +\infty]$ of the system (6.5) is equal to

$$y(t) = Ce^{At}x(0) := \mathcal{O}_b x(0).$$

So the mapping of past inputs to future outputs is characterized by two operators – the so-called controllability operator $\mathcal{C}_o : \mathcal{L}_2[-\infty, 0] \mapsto \mathbb{R}^n$ (mapping past inputs $u(t)$ to the present state) and observability operator $\mathcal{O}_b : \mathbb{R}^n \mapsto \mathcal{L}_2[0, +\infty]$ (mapping the present state to future outputs $y(t)$).

Both $\mathcal{C}_o$ and $\mathcal{O}_b$ have dual operators, $\mathcal{C}_o^*$ and $\mathcal{O}_b^*$, respectively. The operators and their duals are related by two fundamental matrices associated with the linear system (6.5). These are the "controllability Gramian" $\mathcal{P}$ and the "observability Gramian" $\mathcal{Q}$. If A is stable, they are the unique solutions of the Lyapunov equations:

$$A\mathcal{P} + \mathcal{P}A^T + BB^T = 0 \quad , \quad A^T\mathcal{Q} + \mathcal{Q}A + C^TC = 0. \tag{6.10}$$

It follows that $\mathcal{C}_o$ and $\mathcal{O}_b$ are related to their dual operators by the identities $\mathcal{P} = \mathcal{C}_o^*\mathcal{C}_o$ and $\mathcal{Q} = \mathcal{O}_b\mathcal{O}_b^*$ [ZDG95].

Another physical interpretation of the Gramians results from two optimization problems. Let

$$J(v(t), a, b) := \int_a^b v(t)^T v(t) dt$$

be the *energy* of the vector function $v(t)$ in the interval $[a, b]$. It can be shown that (see [ZDG95])

$$\min_{\mathcal{C}_o u(t) = x_0} J(u(t), -\infty, 0) = x_0^T \mathcal{P}^{-1} x_0, \tag{6.11}$$

and, symmetrically, we have the dual property

$$\min_{\mathcal{O}_b^* y(t)=x_0} J(y(t), -\infty, 0) = x_0^T \mathcal{Q}^{-1} x_0. \tag{6.12}$$

Two algebraic properties of Gramians $\mathcal{P}$ and $\mathcal{Q}$ are essential to the development of Balanced Truncation. First, under a coordinate transformation $x(t) = T\bar{x}(t)$, the new Gramians $\bar{\mathcal{P}}$ and $\bar{\mathcal{Q}}$ corresponding to the state-space realization $(\bar{C}, \bar{A}, \bar{B}) := (CT, T^{-1}AT, T^{-1}B)$ undergo the following (so-called *contragradient*) transformation:

$$\bar{\mathcal{P}} = T^{-1}\mathcal{P}T^{-T}, \quad \bar{\mathcal{Q}} = T^T \mathcal{Q} T. \tag{6.13}$$

This implies that the eigenvalues of the product $\bar{\mathcal{P}}\bar{\mathcal{Q}} = T^{-1}\mathcal{P}\mathcal{Q}T$ depend only on the transfer function $R(s)$ and not on a particular choice of state-space realization. It implies also that there exists a state-space realization $(C_{bal}, A_{bal}, B_{bal})$ of $R(s)$ such that the corresponding Gramians are equal and diagonal $\bar{\mathcal{P}} = \bar{\mathcal{Q}} = \Sigma$ [ZDG95].

Second, because the Gramians appear in the solutions of the optimization problems (6.11) and (6.12), they give information about the energy that goes through the system, more specifically, about the distribution of this energy among the state variables. The smaller $x_0^T \mathcal{P}^{-1} x_0$ is, the more "controllable" the state x_0 is, since it can be reached with a input of small energy. By duality, the smaller $x_0^T \mathcal{Q}^{-1} x_0$ is, the more "observable" the state x_0 is. Thus when both Gramians are equal and diagonal, the order of magnitude of a diagonal value of the product $\mathcal{P}\mathcal{Q}$ is a good measure for the influence of the corresponding state variable on the mapping $y(t) = \mathcal{O}_b \mathcal{C}_o u(t)$ that maps past inputs $u(t) \in \mathcal{L}_2[-\infty, 0]$ to future outputs $y(t) \in \mathcal{L}_2[0, +\infty]$ passing via that particular state at time $t = 0$.

Given a transfer function $R(s)$, the Balanced Truncation model reduction method consists of finding a state-space realization $(C_{bal}, A_{bal}, B_{bal})$ of $R(s)$ such that the Gramians are equal and diagonal (this is the *balanced realization* of the system) and then constructing the reduced model by keeping the states corresponding to the largest eigenvalues of the product $\mathcal{P}\mathcal{Q}$ and discarding the others. In other words, the balanced truncation technique chooses Z and V such that $Z^T V = I$, and

$$\begin{cases} \mathcal{P}\mathcal{Q}V = V\Lambda_+ \\ \mathcal{Q}\mathcal{P}Z = Z\Lambda_+ \end{cases} \tag{6.14}$$

where Λ_+ is a square diagonal matrix containing the largest eigenvalues of $\mathcal{P}\mathcal{Q}$. A state-space realization of the reduced transfer function is given by $(CV, Z^T AV, Z^T B)$. The idea of the balanced truncation technique thus consists in keeping those states that are most controllable and observable according to the Gramians defined in (6.10).

Finally, we note that Balanced Truncation can be related to the Hankel operator that maps the past inputs to the future outputs and is defined as

$\mathcal{H} := \mathcal{O}_b\mathcal{C}_o$. Since $\mathcal{PQ} = \mathcal{C}_o\mathcal{C}_o^*\mathcal{O}_b^*\mathcal{O}_b$ and $\mathcal{QP} = \mathcal{O}_b^*\mathcal{O}_b\mathcal{C}_o\mathcal{C}_o^*$, the dominant eigenspaces $\mathcal{V}$ of $\mathcal{PQ}$ and $\mathcal{Z}$ of $\mathcal{QP}$ are linked with the dominant eigenspaces $\mathcal{X}$ of $\mathcal{HH}^*$ and $\mathcal{Y}$ of $\mathcal{H}^*\mathcal{H}$ via the equalities $\mathcal{X} = \mathcal{O}_b\mathcal{V}$ and $\mathcal{Y} = \mathcal{C}_o^*\mathcal{Z}$. Therefore, projecting onto the spaces $\mathcal{V}$ and $\mathcal{Z}$ also approximates the Hankel map $\mathcal{H}$ well. We refer the reader to [ZDG95], for a more detailed study and discussion of the Balanced Truncation method.

Unfortunately, the Balanced Truncation method cannot be applied directly to the state-space realization (C, A, B) (6.7) of the second-order system without destroying its second-order structure in the reduced realization. An approach that solves this problem is discussed in Section 6.4. Also note that, due to its dependence on transformations with $O(N^3)$ complexity, the Balanced Truncation method cannot be applied, as described, to large-scale systems. Recent work by Antoulas and Sorensen considers this problem and describes an Approximate Balanced Truncation approach for large-scale linear systems [SA02].

6.2.2 Krylov Subspace-Based Model Reduction

The Krylov subspace-based model reduction methods have been developed in order to produce reduced order models of large-scale linear systems efficiently and stably via projection onto subspaces that satisfy specific conditions. These conditions are based on requiring the reduced order transfer function to match selected moments of the transfer function $R(s)$ of the original system.

A rational matrix function $R(s)$ is said to be $O(\lambda - s)^k$ in s with $k \in \mathbb{Z}$ if its Taylor expansion about the point λ can be written as

$$R(s) = O(\lambda - s)^k \Longleftrightarrow R(s) = \sum_{i=k}^{+\infty} R_i(\lambda - s)^i, \tag{6.15}$$

where the coefficients R_i are constant matrices. If $R_k \neq 0$, then we say that $R(s) = \Theta(\lambda - s)^k$. As a consequence, if $R(s) = \Theta(\lambda - s)^k$ and k is strictly negative, then λ is a pole of $R(s)$ and if k is strictly positive, then λ is a zero of $R(s)$. Analogously, we say that $R(s)$ is $O(s^{-1})^k$ if the following condition is satisfied:

$$R(s) = O(s^{-1})^k \Longleftrightarrow R(s) = \sum_{i=k}^{+\infty} R_i s^{-i}, \tag{6.16}$$

where the coefficients R_i are constant matrices. It should be stressed that, in general, $R(s)$ being $O(s)^{-k}$ is not equivalent to $R(s)$ being $O(s^{-1})^k$.

Rational Interpolation

Krylov subspaces play an important role in the development of these methods and are defined as follows:

Definition 6.2.2. *Let $M \in \mathbb{C}^{n\times n}$ and $X \in \mathbb{C}^{n\times m}$. A Krylov subspace $\mathcal{K}_k(M, X)$ of order k of the pair (M, X) is the image of the matrix $\begin{bmatrix} M & MX & \dots & M^{k-1}X \end{bmatrix}$.*

If A is stable, $R(s)$ expanded about infinity gives

$$R(s) = C(sI_N - A)^{-1}B = \sum_{i=0}^{\infty} CA^iBs^{-i-1} := \sum_{i=0}^{\infty} R_i^{(\infty)} s^{-i-1},$$

where the coefficients R_i^{∞} are called the Markov parameters of the system. One intuitive way to approximate $R(s)$ is to construct a transfer function $\hat{R}(s)$ of McMillan degree $n \ll N$,

$$\hat{R}(s) := \hat{C}(sI_n - \hat{A})^{-1}\hat{B} := \sum_{i=1}^{\infty} \hat{R}_i^{(\infty)} s^{-i} \tag{6.17}$$

such that $\hat{R}_i^{(\infty)} = R_i^{(\infty)}$ for $1 \le i \le r$, where r is as large as possible and is generically equal to $2n$. The resulting reduced transfer function $\hat{R}(s)$ generally approximates quite well the original transfer function for large values of s.

If a good approximation for low frequencies is desired, one can construct a transfer function

$$\hat{R}(s) = \hat{C}(sI_n - \hat{A})^{-1}\hat{B} = \sum_{k=0}^{\infty} \hat{R}_k^{(\lambda)} (\lambda - s)^k,$$

such that

$$\hat{R}_k^{(\lambda)} = R_k^{(\lambda)} \qquad for\ 1 \le k \le K, \tag{6.18}$$

with

$$R_k^{(\lambda)} := C(\lambda I_N - A)^{-k}B, \quad \hat{R}_k^{(\lambda)} := \hat{C}(\lambda I_n - \hat{A})^{-k}\hat{B}.$$

In short, (6.18) can be rewritten as follows:

$$R(s) - \hat{R}(s) = O(\lambda - s)^K.$$

More generally, one can choose a transfer function $\hat{R}(s)$ that interpolates $R(s)$ at several points in the complex plane, up to several orders. The main results concerning this problem for MIMO standard state space systems are summarized in the following theorem.

Theorem 6.2.3. *Let the original system be*

$$R(s) := C(sI_N - A)^{-1}B, \tag{6.19}$$

and the reduced system be

$$\hat{R}(s) := CV\left(Z^T(sI_N - A)V\right)^{-1} Z^T B, \tag{6.20}$$

with $Z^T V = I_n$. *If*

$$\bigcup_{k=1}^{K} \mathcal{K}_{b_k}((\lambda_k I - A)^{-1}, (\lambda_k I - A)^{-1} B) \subseteq Im(V) \tag{6.21}$$

and

$$\bigcup_{k=1}^{K} \mathcal{K}_{c_k}((\lambda_k I - A)^{-T}, (\lambda_k I - A)^{-T} C^T) \subseteq Im(Z) \tag{6.22}$$

where the interpolation points λ_k *are chosen such that the matrices* $\lambda_k I_N - A$ *are invertible* $\forall k \in \{1, \ldots, K\}$ *then the moments of the systems (6.19) and (6.20) at the points* λ_k *satisfy*

$$R(s) - \hat{R}(s) = O(s - \lambda_k)^{b_k + c_k}, \tag{6.23}$$

provided these moments exist, i.e. provided the matrices $\lambda_k I_n - \hat{A}$ *are invertible.*

For a proof, see [dVS87] and [Gri97]. A proof for MIMO generalized state space systems is given in [GVV04b]. □

Matching Markov parameters, i.e., $\lambda = \infty$, is known as *partial realization*. When $\lambda = 0$, the corresponding problem is known as *Padé approximation*. If λ takes a finite number of points λ_i, it is called a *multi-point Padé approximation*. In the general case, the problem is known as *rational interpolation*. Rational interpolation generally results in a good approximation of the original transfer function in a region near the expansion points (and increasing the order at a point tends to expand the region), but may not be accurate at other frequencies (see for instance [Ant05]).

The advantage of these moment matching methods is that they can be implemented in a numerically stable and efficient way for large-scale systems with sparse coefficient matrices (see for example [GVV04b] and [Gri97]). Also, the local approximation property means that good approximations can be achieved in specific regions over a wide dynamic range, typically at the cost of a larger global error. This requires however, that the interpolation points and their corresponding order of approximation must be specified. For some applications, the user may have such information but for blackbox library software a heuristic automatic selection strategy is needed (see [Gri97]) and the design of such a strategy is still an open question. The other main drawback is the lack of an error bound on the global quality of the approximation, e.g., the H_∞-norm of the difference between original and reduced transfer functions. Recent research has begun to address the evaluation of the H_∞-norm given a reduced order model that may help in selecting points [CGV04].

One could apply these methods to the state space realization (6.9) of a second-order transfer function. Unfortunately, if the methods are used in the forms described, the resulting reduced order transfer function will generically not be in second-order form. An approach to maintain second-order form is discussed in Section 6.5.

Tangential Interpolation

The Krylov subspace-based methods that produce reduced order models based on rational interpolation can be applied to MIMO systems efficiently as long as the number of inputs and outputs, m and p, stay suitably moderate in size. For MIMO systems where m and p are too large, a more general *tangential interpolation* problem has recently been considered (see [GVV04a]). Instead of imposing interpolation condition of the form $R(\lambda_i) = \hat{R}(\lambda_i)$, one could be interested, for example, in only imposing interpolation conditions of the following form:

$$\hat{R}(\lambda_i)x_i = R(\lambda_i)x_i \quad , \quad y_i\hat{R}(\lambda_{i+n}) = y_iR(\lambda_{i+n}), \quad 1 \leq i \leq n, \tag{6.24}$$

where the n column vectors x_i are called the right interpolation directions and the n row vectors y_i are called the left interpolation directions. As with rational interpolation, higher order tangential interpolation conditions can be imposed at each point to improve the approximation.

Stable and efficient methods for tangential interpolation of MIMO systems can be developed using theorems and techniques similar to those used for Krylov subspace-based rational interpolation. However, the problem of constructing a reduced transfer function that satisfies a set of tangential interpolation conditions and that preserves the second-order structure of the original transfer function requires additional consideration as discussed in Section 6.5.

6.3 Second-Order Structure Preserving Model Reduction

In this section, a simple sufficient condition for obtaining a second-order reduced system from a second-order system is presented. The following result can be found in a slightly different form in [CLVV05].

Lemma 6.3.1. *Let (C, A, B) be the state space realization defined in* (6.9)*. If one projects such a state space realization with $2N \times 2n$ bloc diagonal matrices*

$$\bar{Z} := \begin{bmatrix} Z_1 & 0 \\ 0 & Z_2 \end{bmatrix}, \quad \bar{V} := \begin{bmatrix} V_1 & 0 \\ 0 & V_2 \end{bmatrix}, \quad \bar{Z}^T\bar{V} = I_{2n},$$

where $Z_1, V_1, Z_2, V_2 \in \mathbb{C}^{N\times n}$, then the reduced transfer function

$$\hat{R}(s) := C\bar{V}\left(\bar{Z}^T(sI_{2N} - A)\bar{V}\right)^{-1}\bar{Z}^TB$$

is a second-order transfer function, provided the matrix $Z_1^TV_2$ is invertible.

Proof. First, notice that the transfer function does not change under any similarity transformation of the system matrices. Let us consider the similarity transformation $M \in \mathbb{C}^{2n\times 2n}$ such that

$$M := \begin{bmatrix} X & \\ & Y \end{bmatrix},$$

with $X, Y \in \mathbb{C}^{n\times n}$ verifying

$$X^{-1}(Z_1^T V_2)Y = I_n.$$

From the preceding results,

$$\begin{aligned}\hat{R}(s) &:= C\bar{V}M\left(M^{-1}\bar{Z}^T(sI_{2N}-A)\bar{V}M\right)^{-1}M^{-1}\bar{Z}^TB\\ &= F_M^{out}V_1X\left(s^2I_n + sY^{-1}Z_2^TD_MV_2Y + Y^{-1}Z_2^TS_MV_1X\right)^{-1}Y^{-1}Z_2^TF_M^{in}.\end{aligned}$$

This is clearly a second-order transfer function. □

6.4 Second-Order Balanced Truncation

The earliest balanced truncation technique for second-order systems known to the authors is described in [MS96]. Based on this work, an alternative technique was developed in [CLVV05]. In this section an overview of the latter method, called SOBT (Second-Order Balanced Truncation), is given.

The first step in the development of SOBT, based on a balance and truncate process similar to that discussed in Section 6.2.1, involves the definition of *two* pairs of $N \times N$ Gramians ("second-order Gramians") that change according to contragradient transformations, and that have some energetic interpretation. The first pair $(\mathcal{P}_{pos}, \mathcal{Q}_{pos})$ corresponds to an energy optimization problem depending only on the *positions* $q(t)$ and not on the *velocities* $\dot{q}(t)$. Reciprocally, the second pair $(\mathcal{P}_{vel}, \mathcal{Q}_{vel})$ correspond to an optimization problem depending only on the velocities $\dot{q}(t)$ and not the on the positions $q(t)$. By analogy to the first order case, the Gramians $\mathcal{Q}_{pos}$ and $\mathcal{Q}_{vel}$ are defined from the dual systems. Given the Gramians, a balancing step in the method is defined by transforming to a coordinate system in which the second-order Gramians are equal and diagonal: $\bar{\mathcal{P}}_{pos} = \bar{\mathcal{Q}}_{pos} = \Sigma_{pos}, \quad \bar{\mathcal{P}}_{vel} = \bar{\mathcal{Q}}_{vel} = \Sigma_{vel}$. Their diagonal values enable us to identify the *important* positions and the *important* velocities, i.e. those with (hopefully) large effect on the I/O map. Once identified, the reduced second-order model follows by truncation of all variables not identified as important.

In order to define a pair of second-order Gramians measuring the contribution of the position coordinates (independently of the velocities) with respect to the I/O map, consider an optimization problem naturally associated with the second-order system (see [MS96]) of the form

$$\min_{q_0\in\mathbb{R}^n}\ \min_{u(t)}\ J(u(t), -\infty, 0), \tag{6.25}$$

subject to

$$\ddot{q}(t) + D_M \dot{q}(t) + S_M q(t) = F_M^{in} u(t),\ q(0) = q_0.$$

One easily sees that the optimum is $q_0^T {\mathcal{P}_{11}}^{-1} q_0$, where $\mathcal{P}_{11}$ is the $N \times N$ left upper block of the controllability Gramian $\mathcal{P}$ satisfying equation (6.10) with (C, A, B) given in (6.9). Starting with (6.11) we must solve

$$\min_{\dot{q}_0 \in \mathbb{R}^n} J_{q_0}(\dot{q}_0) \ = \ \left[\, q_0^T \ \dot{q}_0^T \, \right] \mathcal{P}^{-1} \begin{bmatrix} q_0 \\ \dot{q}_0 \end{bmatrix}.$$

Partitioning $\mathcal{P}^{-1}$ as follows

$$\mathcal{P}^{-1} \ = \ \begin{bmatrix} R_1 & R_2 \\ R_2^T & R_3 \end{bmatrix}$$

and annihilating the gradient of $J_{q_0}(\dot{q}_0)$ gives the relation $\dot{q}_0 = -R_3^{-1} R_2^T q_0$. The value of J_{q_0} at this point is $q_0^T (R_1 - R_2 R_3^{-1} R_2^T) q_0$. This is simply the Schur complement of R_3 which is ${\mathcal{P}_{11}}^{-1}$. Similarly, the solution of the dual problem corresponds to $q_0^T {\mathcal{Q}_{11}}^{-1} q_0$, where $\mathcal{Q}_{11}$ is the $N \times N$ left upper block of the observability Gramian $\mathcal{Q}$ (6.10).

Note that the transfer function is seen as a linear operator acting between two Hilbert spaces. The dual of such an operator is defined in Definition 6.2.1. It follows that the dual of a second-order transfer function might not be a second-order transfer function. This has no consequences here since only the energy transfer interpretation between the inputs, the outputs, the initial positions and velocities is important. Under the change of coordinates $q(t) = T\bar{q}(t)$, it is easy to verify that this pair of Gramians undergoes a contragradient transformation:

$$(\bar{\mathcal{P}}_{11}, \bar{\mathcal{Q}}_{11}) = (T^{-1} \mathcal{P}_{11} T^{-T}, T^T \mathcal{Q}_{11} T).$$

This implies that there exists a new coordinate system such that both $\mathcal{P}_{11}$ and $\mathcal{Q}_{11}$ are equal and diagonal. Their energetic interpretation is seen by considering the underlying optimization problem. In (6.25), the energy necessary to reach the given position q_0 over all past inputs and initial velocities is minimized. Hence, these Gramians describe the distribution of the I/O energy among the positions.

A pair of second-order Gramians that gives the contribution of the velocities with respect to the I/O map can be defined analogously. The associated optimization problem is

$$\min_{q_0 \in \mathbb{R}^n} \min_{u(t)} \ J(u(t), -\infty, 0) \tag{6.26}$$

subject to

$$\ddot{q}(t) + D_M \dot{q}(t) + S_M q(t) \ = \ F_M^{in} u(t), \quad \dot{q}(0) \ = \ \dot{q}_0.$$

Following the same reasoning as before for the optimization problem (6.25), one can show that the solution of (6.26) is $\dot{q}_0^T {\mathcal{P}_{22}}^{-1} \dot{q}_0$, where $\mathcal{P}_{22}$ is the $N \times N$

right lower block of $\mathcal{P}$. The solution of the dual problem is $\dot{q}_0^T \mathcal{Q}_{22}{}^{-1} \dot{q}_0$, where $\mathcal{Q}_{22}$ is the $N \times N$ right lower block of $\mathcal{Q}$. As before, under the change of coordinates $q(t) = T\bar{q}(t)$ one can check that this pair of Gramians undergoes a contragradient transformation and that the energetic interpretation is given by considering the underlying optimization problem. In (6.26), the energy necessary to reach the given velocity $\dot{q}_0$ over all past inputs and initial positions is minimized. Hence, these Gramians describe the distribution of the I/O energy among the velocities.

Given the interpretation above these second-order Gramians are good candidates for balancing and truncating. Therefore, we choose:

$$(\mathcal{P}_{pos},\ \mathcal{Q}_{pos}) = (\mathcal{P}_{11},\ \mathcal{Q}_{11}) \quad \text{and} \quad (\mathcal{P}_{vel},\ \mathcal{Q}_{vel}) = (\mathcal{P}_{22},\ \mathcal{Q}_{22})\ . \tag{6.27}$$

It is not possible to balance both pairs of second-order Gramians at the same time with a single change of coordinates of the type $q(t) = T\bar{q}(t)$. A change of coordinates is required for both positions and velocities (unlike the approach in [MS96]). Therefore, we work in a state-space context, starting with the system (6.9). The SOBT method, therefore, first computes both pairs of second-order Gramians, $(\mathcal{P}_{pos},\ \mathcal{Q}_{pos})$ and $(\mathcal{P}_{vel},\ \mathcal{Q}_{vel})$. Given the Gramians, the contragradient transformations that make $\mathcal{P}_{pos} = \mathcal{Q}_{pos} = \Lambda_{pos}$ and $\mathcal{P}_{vel} = \mathcal{Q}_{vel} = \Lambda_{vel}$, where Λ_{pos} and Λ_{vel} are positive definite diagonal matrices, are computed. Finally, truncate the positions corresponding to the smallest eigenvalues of Λ_{pos} and the velocities corresponding to the smallest eigenvalues of Λ_{vel}.

At present, there exists no a priori global error bound for SOBT and the stability of the reduced system is not guaranteed. Nevertheless, SOBT yields good numerical results, providing reduced transfer functions with approximation error comparable with the traditional Balanced Truncation technique.

6.5 Second-Order Structure Preserving Krylov Techniques

The Krylov subspace-based methods discussed in Section 6.2.2 do not preserve second-order structure when applied to the linear system (6.9). It is possible to modify them to satisfy the constraint presented in Section 6.3 and thereby produce a second-order reduced system. Section 6.5.1 summarizes the earliest Krylov subspace-based method for second-order systems [SC91]. The simple technique constructs, via projection, a second-order reduced transfer function that matches the Markov parameters ($\lambda = \infty$) of the original transfer function. The limitation of the technique when applied to a complex interpolation point is also discussed. Section 6.5.2, addresses this limitation using a generalization that allows multipoint rational interpolation. Finally, the problem of second-order structure preserving tangential interpolation is solved in 6.5.3.

6.5.1 A Particular Case: Matching the Markov Parameters

Su and Craig proposed a Krylov subspace-based projection method that preserves second-order structure while matching the Markov parameters of the original transfer function [SC91]. The method is based on the observation that the right Krylov subspace corresponding to interpolation at $\lambda = \infty$ for the system (6.9) has the form

$$\begin{aligned}\begin{bmatrix} B & AB & A^2B & \dots \end{bmatrix} &= \begin{bmatrix} 0 & F_M^{in} & -D_M F_M^{in} & \dots \\ F_M^{in} & -D_M F_M^{in} & -S_M F_M^{in} + D_M^2 F_M^{in} & \dots \end{bmatrix} \quad (6.28)\\ &= \begin{bmatrix} 0 & Q_{v,0} & Q_{v,1} & \dots \\ Q_{v,0} & Q_{v,1} & Q_{v,2} & \dots \end{bmatrix}. \quad (6.29)\end{aligned}$$

and that if we write

$$\mathcal{K}_k(A, B) = \begin{bmatrix} V_1 \\ V_2 \end{bmatrix},$$

it follows that

$$Im(V_1) \subseteq Im(V_2).$$

So by projecting the state space realization (6.9) with

$$\bar{V} := \begin{bmatrix} V_2 & 0 \\ 0 & V_2 \end{bmatrix}, \quad \bar{Z} := \begin{bmatrix} Z & 0 \\ 0 & Z \end{bmatrix}$$

such that $Z^T V_2 = I_n$, we obtain an interpolating second-order transfer function of the form

$$\hat{R}(s) = F_M^{out} V_2 \left(Z^T (s^2 I_N + s D_M + S_M)^{-1} V_2 \right) Z^T F_M^{in}. \quad (6.30)$$

Hence, a second-order system with the same n first Markov parameters as the original second-order system can be constructed by projecting with $Z, V \in \mathbb{C}^{N \times n}$ such that $Z^T V = I_n$ and the image of V contains the image of $Q_{v,0}, \dots, Q_{v,n-1}$. Since $\mathcal{K}_n(A, B) \subseteq \bar{V}$, it follows from Theorem 6.2.3 that the first n Markov parameters of $R(s)$ and $\hat{R}(s)$ are equal.

If we apply the construction for any interpolation point $\lambda \in \mathbb{C}$, the corresponding right Krylov space is such that

$$\mathcal{K}_k((\lambda I - A)^{-1}, (\lambda I - A)^{-1} B) = Im \begin{bmatrix} V_1 \\ V_2 \end{bmatrix},$$

with A and B defined in (6.9) and

$$Im(V_1) \subseteq Im(V_2).$$

Unfortunately, a similar statement can not be made for the *left* Krylov subspaces $\mathcal{K}_k((\lambda I - A)^{-T}, (\lambda I - A)^{-T} C^T)$. This implies that when the second-order Krylov technique is extended to interpolation at arbitrary points in the complex plane by projecting as in (6.30), only n interpolation conditions can be imposed for a reduced second-order system of McMillan degree $2n$.

6.5.2 Second-Order Rational Interpolation

The projection technique of Su and Craig has been generalized independently by several authors (see [VV04, BS04] and also Chapter 7 and Chapter 8) to solve the rational interpolation problem that produces a *second-order* transfer function of order n, i.e., of McMillan degree $2n$, $\hat{R}(s)$, that interpolates $R(s)$ at $2n$ points in the complex plane. After some preliminary discussion of notation, the conditions that determine the projections are given in Theorem 6.5.1 and the associated algorithm is presented.

By combining the results of Sections 6.2 and 6.3, the following theorem can be proven.

Theorem 6.5.1. *Let* $R(s) := F_M^{out}(s^2 I_N + D_M s + S_M)^{-1} F_M^{in} = C(sI_{2N} - A)^{-1}B$, *with*

$$A := \begin{bmatrix} 0 & I_N \\ -S_M & -D_M \end{bmatrix}, \quad B := \begin{bmatrix} 0 \\ F_M^{in} \end{bmatrix}, \quad C := \begin{bmatrix} F_M^{out} & 0 \end{bmatrix},$$

be a second-order transfer function of McMillan degree $2N$, *i.e.* $S_M, D_M \in \mathbb{C}^{N\times N}$). *Let* $Z, V \in \mathbb{C}^{2N\times n}$ *be defined as*

$$V := \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} \quad , \quad Z := \begin{bmatrix} Z_1 \\ Z_2 \end{bmatrix},$$

with V_1, V_2, Z_1 *and* $Z_2 \in \mathbb{C}^{N\times n}$ *such that*

$$Z_1^T V_1 = Z_2^T V_2 = I_n.$$

Let us define the $2N \times 2n$ *projecting matrices*

$$\bar{V} := \begin{bmatrix} V_1 & 0 \\ 0 & V_2 \end{bmatrix} \quad , \quad \bar{Z} := \begin{bmatrix} Z_1 & 0 \\ 0 & Z_2 \end{bmatrix}.$$

Define the second-order transfer function $\hat{R}(s)$ *of order* n *(and of McMillan degree* $2n$*) by*

$$\begin{aligned} \hat{R}(s) &:= C\bar{V}\left(\bar{Z}^T(sI_{2N} - A)\bar{V}\right)^{-1}\bar{Z}^T B \\ &:= \hat{C}(sI_{2n} - \hat{A})^{-1}\hat{B}. \end{aligned} \tag{6.31}$$

If

$$\bigcup_{k=1}^{K} \mathcal{K}_{b_k}((\lambda_k I_{2N} - A)^{-1}, (\lambda_k I_{2N} - A)^{-1}B) \subseteq Im(V) \tag{6.32}$$

and

$$\bigcup_{k=1}^{K} \mathcal{K}_{c_k}((\lambda_k I_{2N} - A)^{-T}, (\lambda_k I_{2N} - A)^{-T}C^T) \subseteq Im(Z) \tag{6.33}$$

where the interpolation points λ_k are chosen such that the matrices $\lambda_k I_{2N} - A$ are invertible $\forall k \in \{1, \ldots, K\}$ then, if the matrix $Z_1^T V_2$ is invertible,

$$R(s) - \hat{R}(s) = O(s - \lambda_k)^{b_k + c_k} \tag{6.34}$$

for the finite points λ_k, provided these moments exist, i.e. provided the matrices $\lambda_k I_{2n} - \hat{A}$ are invertible and

$$R(s) - \hat{R}(s) = O(s^{-1})^{b_k + c_k} \tag{6.35}$$

if $\lambda_k = \infty$.

Proof. Clearly, $\bar{Z}^T \bar{V} = I_{2n}$. The second-order structure of $\hat{R}(s)$ follows from Lemma 6.3.1. It is clear that

$$Im(V) \subset Im(\bar{V}) \quad , \quad Im(Z) \subset Im(\bar{Z}).$$

The interpolation conditions are then satisfied because of Theorem 6.2.3. □

The form of the projectors allows the development of an algorithm similar to the Rational Krylov family of algorithms for first order systems [Gri97]. The algorithm, shown below, finds a *second-order* transfer function of order n, i.e. of McMillan degree $2n$, $\hat{R}(s)$, that interpolates $R(s)$ at $2n$ interpolation points λ_1 up to λ_{2n}, i.e.,

$$R(s) - \hat{R}(s) = O(\lambda_i - s) \qquad for\ 1 \leq i \leq 2n. \tag{6.36}$$

We assume for simplicity that the interpolation points are finite, distinct and not poles of $R(s)$. The algorithm is easily modified to impose higher order conditions at the interpolation points.

Algorithm 1 *1. Construct Z and V such that*

$$V = \left[(\lambda_1 I_{2N} - A)^{-1} B \ \ldots \ (\lambda_n I_{2N} - A)^{-1} B \right] = \begin{bmatrix} V_1 \\ V_2 \end{bmatrix}$$

$$Z^T = \begin{bmatrix} C(\lambda_{n+1} I_{2N} - A)^{-1} \\ \vdots \\ C(\lambda_{2n} I_{2N} - A)^{-1} \end{bmatrix} = \left[Z_1^T \ Z_2^T \right],$$

where $V_1, V_2 \in \mathbb{C}^{N \times n}$ are the first N rows and the last N rows of V respectively and $Z_1, Z_2 \in \mathbb{C}^{N \times n}$ are the first N rows and the last N rows of Z respectively. Choose the matrices $M_1, M_2, N_1, N_2 \in \mathbb{C}^{n \times n}$ such that $N_1^T Z_1^T V_1 M_1 = N_2^T Z_2^T V_2 M_2 = I_n$.

2. Construct

$$\bar{V} := \begin{bmatrix} V_1 M_1 & \\ & V_2 M_2 \end{bmatrix} \quad , \quad \bar{Z} := \begin{bmatrix} Z_1 N_1 & \\ & Z_2 N_2 \end{bmatrix}.$$

3. Construct the matrices

$$\hat{C} := C\bar{V} \quad , \quad \hat{A} := \bar{Z}^T A \bar{V} \quad , \quad \hat{B} := \bar{Z}^T B.$$

and define the reduced transfer function

$$\hat{R}(s) := \hat{C}(sI_{2n} - \hat{A})^{-1}\hat{B}.$$

From Theorem 6.5.1, $\hat{R}(s)$ is a second-order transfer function of order n that satisfies the interpolation conditions (6.36). The algorithm above has all of the freedom in the method of forming the bases and selecting interpolation points and their associated orders found in the Rational Krylov family of algorithms [Gri97]. As a result, the second-order rational interpolation problem can be solved while exploiting the sparsity of the matrices and parallelism of the computing platform in a similar fashion.

6.5.3 Second-order Structure Preserving Tangential Interpolation

It is possible to generalize the earlier results for MIMO systems to perform tangential interpolation and preserve second-order structure. This is accomplished by replacing Krylov subspaces at each interpolation point, λ_i, with *generalized* Krylov subspaces as done in [GVV04a]. The spaces are defined as follows:

Definition 6.5.2. *Let $M \in \mathbb{C}^{n\times n}$, $X \in \mathbb{C}^{n\times m}$, $y^{[i]} \in \mathbb{C}^m$, $i = 0, \ldots, k-1$ and define $Y \in \mathbb{C}^{km\times k}$ as*

$$Y = \begin{bmatrix} y^{[0]} & \ldots & y^{[k-1]} \\ & \ddots & \vdots \\ & & y^{[0]} \end{bmatrix}.$$

A generalized Krylov subspace of order k, denoted $\mathcal{K}_k(M, X, Y)$, is the image of the matrix $\begin{bmatrix} X & MX & \ldots & M^{k-1}X \end{bmatrix} Y$.

For example, by using Algorithm 2 below to compute bases for *generalized* Krylov subspaces and forming the appropriate projections, one can construct a second-order transfer function $\hat{R}(s)$ of order n that satisfies the following interpolation conditions with respect to the second-order transfer function $R(s)$ of order N:

$$x_i \left(R(s) - \hat{R}(s)\right) = O(\lambda_i - s) \quad , \quad \left(R(s) - \hat{R}(s)\right) x_{i+n} = O(\lambda_{i+n} - s), \tag{6.37}$$

where $x_1, \ldots, x_n \in \mathbb{C}^{1\times p}$ and $x_{n+1}, \ldots, x_{2n} \in \mathbb{C}^{m\times 1}$.

Algorithm 2 *1. Construct Z and V such that*

$$V = \left[(\lambda_{n+1} I_{2N} - A)^{-1} B x_{n+1} \ \ldots \ (\lambda_{2n} I_{2n} - A)^{-1} B x_{2n} \right] = \begin{bmatrix} V_1 \\ V_2 \end{bmatrix}$$

$$Z^T = \begin{bmatrix} x_1 C(\lambda_1 I_{2N} - A)^{-1} \\ \vdots \\ x_n C(\lambda_n I_{2N} - A)^{-1} \end{bmatrix} = \left[Z_1^T \ Z_2^T \right],$$

where $Z_1, Z_2, V_1, V_2 \in \mathbb{C}^{N \times n}$. *Choose the matrices* $M_1, M_2, N_1, N_2 \in \mathbb{C}^{n \times n}$ *such that* $N_1^T Z_1^T V_1 M_1 = N_2^T Z_2^T V_2 M_2 = I_n$.

2. *Construct*

$$\bar{V} := \begin{bmatrix} V_1 M_1 & \\ & V_2 M_2 \end{bmatrix} \quad , \quad \bar{Z} := \begin{bmatrix} Z_1 N_1 & \\ & Z_2 N_2 \end{bmatrix}.$$

3. *Construct the matrices*

$$\hat{C} := C\bar{V} \quad , \quad \hat{A} := \bar{Z}^T A \bar{V} \quad , \quad \hat{B} := \bar{Z}^T B.$$

and define the reduced transfer function

$$\hat{R}(s) := \hat{C}(sI_{2n} - \hat{A})^{-1} \hat{B}.$$

It can be shown that $\hat{R}(s)$ is a second-order transfer function of order n that satisfies the interpolation conditions (6.37) (see [GVV04a]).

It is also possible to impose higher order conditions while preserving the structure of the algorithm and the reduced order system. Consider, for instance, right tangential interpolation conditions of higher order (similar results hold for left tangential interpolation). Let the polynomial vector $x(s) := \sum_{i=0}^{k-1} x^{[i]}(s-\lambda)^i$. To impose the tangential interpolation condition

$$\left(R(s) - \hat{R}(s) \right) x(s) = O(s-\lambda)^k,$$

we construct $\hat{R}(s)$ as in Algorithm 2 using the generalized Krylov subspace $\mathcal{K}((\lambda I - A)^{-1}, (\lambda I - A)^{-1} B, X)$ where X is formed from the $x^{[i]}$, $i = 0, \ldots, k-1$, i.e.,

$$Im \left\{ \left[(\lambda I - A)^{-1} B \ \ldots \ (\lambda I - A)^{-k} B \right] \begin{bmatrix} x^{[0]} & \ldots & x^{[k-1]} \\ & \ddots & \vdots \\ & & x^{[0]} \end{bmatrix} \right\} \subseteq Im \left\{ \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} \right\}.$$

We refer to [GVV04a] for more details on this topic.

6.6 Numerical Experiments

In this section, model reduction techniques are applied to a large scale second-order system representing the vibrating structure of a building. The objective

is to compare the performance of second-order structure preserving model reduction techniques, namely the SOBT technique introduced in Section 6.4 and the Second-Order Krylov technique introduced in Section 6.5, with respect to the standard first order techniques, namely the Balanced Truncation and the Multipoint Padé techniques.

The characteristics of the second-order system to be reduced are the following. The stiffness and mass matrix S and M are of dimension $N = 26,394$. (See Chapter 24, Section 4, this volume, for a description of the example.) The mass matrix M is diagonal and the stiffness matrix S is symmetric and sparse (S contains approximately 2×10^5 non zero elements). The input vector is the transpose of the output vector:

$$C = B^T = \left[1 \ldots 1\right].$$

The damping matrix is *proportional*, meaning it is a linear combination of the mass matrix M and the stiffness matrix S:

$$D := \alpha M + \beta S,$$

with $\alpha = 0.675$ and $\beta = 0.00315$. The second-order transfer function of McMillan degree $2N = 52788$ to be reduced is

$$R(s) := B^T(s^2M + sD + S)^{-1}B = B^T(s^2M + s(\alpha M + \beta S) + S)^{-1}B.$$

Given the structure of M we normalize the equation so that the mass matrix is the identity as follows:

$$\begin{aligned} R(s) \ &= \ B^T M^{-1/2} \Big(s^2 I + s(\alpha I + \beta M^{-1/2} S M^{-1/2}) + \\ &\qquad\qquad M^{-1/2} S M^{-1/2}\Big)^{-1} M^{-1/2} B \\ &:= \bar{C}\left(s^2 I + s(\alpha I + \beta \bar{S}) + \bar{S}\right)^{-1} \bar{B}, \end{aligned}$$

where $\bar{S} := M^{-1/2}SM^{-1/2}$ and $\bar{B} := M^{-1/2}B = \bar{C}^T$.

One intermediate system and five reduced order systems will be constructed from $R(s)$. Three reasons led us to construct an intermediate transfer function. First, concerning the SVD techniques, it is not possible to apply the Balanced Truncation or the Second-Order Balanced Truncation methods directly to the transfer function $R(s)$ because its McMillan degree $2N$ is too large for applying $O(N^3)$ algorithms. Second, the intermediate transfer function, assumed very close to $R(s)$, will also be used to approximate of the error bound between the different reduced transfer functions and the original transfer function $R(s)$. Finally, the intermediate transfer function will also be used in order to choose interpolation points for the Krylov techniques.

For these reasons, an intermediate second-order transfer function of order 200 (i.e. of McMillan degree 400), called $R_{200}(s)$, is first constructed from $R(s)$ using Modal Approximation by projecting $\bar{S}$ onto its eigenspace corresponding

to its 200 eigenvalues of smallest magnitude. This corresponds to keeping the 400 eigenvalues of $s^2 I + s(\alpha I + \beta \bar{S}) + \bar{S}$ that are closest to the imaginary axis. Let $V_{f200} \in \mathbb{R}^{26364 \times 200}$ be the projection matrix corresponding to the 200 eigenvalues of $\bar{S}$ the closest to the imaginary axis (with $V_{f200}^T V_{f200} = I_{200}$) ($V_{f200}$ is computed with the Matlab function *eigs*). The intermediate transfer function is

$$R_{200}(s) := \bar{C} V_{f200} \left(s^2 I + s(\alpha I + \beta V_{f200}^T \bar{S} V_{f200}) + V_{f200}^T \bar{S} V_{f200} \right)^{-1} V_{f200}^T \bar{B}.$$

By checking the difference between $R(s)$ and $R_{200}(s)$ at different points in the complex plane, it has been verified that the transfer functions are very close to each other. The Hankel singular values of $R_{200}(s)$ are shown in Figure 6.1.

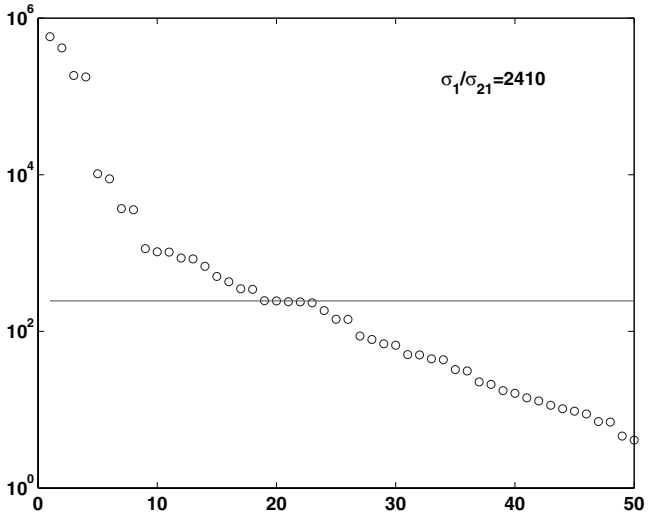


Fig. 6.1. Hankel singular values of $R_{200}(s)$

From $R_{200}(s)$, we compute the first reduced transfer function of McMillan degree 20 obtained by using balanced truncation (with the *sysred* Matlab function of the Niconet library), called $R_{bt}(s)$. Note that $R_{bt}(s)$ is no longer in second-order form. Another second order transfer function of order 20 (and McMillan degree 40), called $R_{sobt}(s)$, is constructed from $R_{200}(s)$ using the SOBT algorithm [CLVV05].

For the Krylov techniques, the reduced order transfer functions are computed directly from the original transfer function $R(s)$. Three reduced order systems are compared. The first one is constructed using the standard first order Krylov procedure. The two other reduced systems (corresponding to different choices of interpolation points) are constructed using a second-order Krylov technique.

In order to apply Krylov techniques, a first important step consists in choosing the interpolation points. Indeed, the quality of the reduced order system is very sensitive to the choice of interpolation points.

An interesting fact is that there are 42 interpolation points between $R_{200}(s)$ and $R_{bt}(s)$ that have a positive real part (among the 420 zeros of $R_{200}(s) - R_{bt}(s)$). From several experiments, it has been observed that when using the standard Balanced Truncation technique, the number of interpolation points in the right-half plane between the original and the reduced transfer function is roughly equal to twice the McMillan degree of the reduced transfer function. The interpolation points in the right-half plane have the advantage that they are neither close to the poles of the system to be reduced nor to the poles of the Balanced Truncation reduced system because both transfer functions are stable. This implies that both transfer functions do not vary too much there and this is preferable in order to avoid numerical difficulties.

Because the McMillan degree of $R_{bt}(s)$ is equal to 20, it is well known that 40 points are sufficient in order to describe $R_{bt}(s)$. In other words, the only transfer function of Mc Millan degree smaller than 20 that interpolates $R_{bt}(s)$ at 40 points in the complex plane is $R_{bt}(s)$ itself [GVV03]. So, we take the 40 interpolation points (these are 20 complex conjugate pairs of points) between $R_{200}(s)$ and $R_{bt}(s)$ with largest real part as our choice for computing the transfer function of McMillan degree 20, denoted $R_{Kryl}(s)$, that interpolates the original transfer function $R(s)$ at these points. The poles and interpolation points are shown in Figure 6.2. Because $R_{200}(s)$ is very close to $R(s)$, $R_{Kryl}(s)$ should be close to $R_{bt}(s)$.

Using the second-order Krylov technique, a reduced second-order transfer function $R_{sokryl}(s)$ of McMillan degree 28 is also constructed. Its McMillan degree was first chosen to be 20 but the resulting reduced transfer function was not stable. For this reason, additional interpolation conditions were added until the reduced transfer function was stable, resulting in a McMillan degree equal to 28. The transfer function $R_{sokryl}(s)$ interpolates $R(s)$ at the 28 rightmost interpolation points between $R_{200}(s)$ and $R_{bt}(s)$.

For comparison purposes a set of interpolation points randomly generated (with symmetry with respect to the real axis in order to obtain a real interpolating transfer function) in a rectangle delimited by the extreme zeros in the left half plane of $R_{200}(s) - R_{bt}(s)$ is also used in the second-order Krylov method to generate $R_{randsokryl}(s)$. These two sets of interpolation points are shown in Figure 6.3.

The Bode magnitude diagrams $R_{200}(s)$, $R_{bt}(s)$, $R_{sobt}(s)$, $R_{randsokryl}(s)$, $R_{kryl}(s)$ and $R_{sokryl}(s)$ are plotted in Figure 6.4. Recall, that $R_{200}(s)$ is used here as computationally tractable approximation of $R(s)$. More can be learned by considering the the H_∞-norm errors relative to $\|R_{200}(s)\|_\infty$ shown in Table 6.1.

As a first observation, it looks as if the six transfer functions are close to each other, especially for frequencies smaller than 10 rad/sec (where the bode

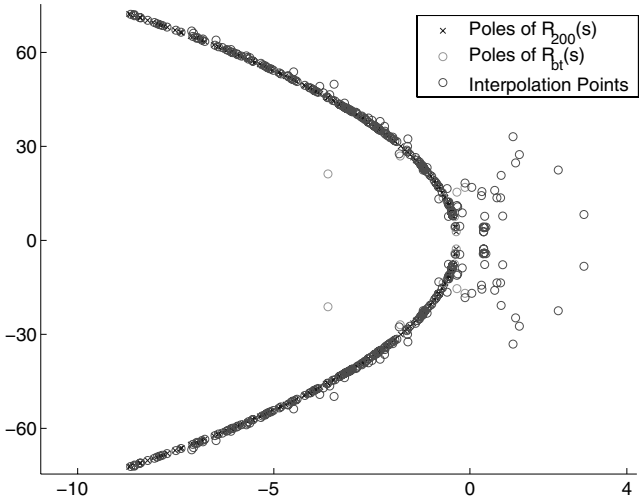


Fig. 6.2. Poles and interpolation points for $R_{200}(s)$ and $R_{bt}(s)$

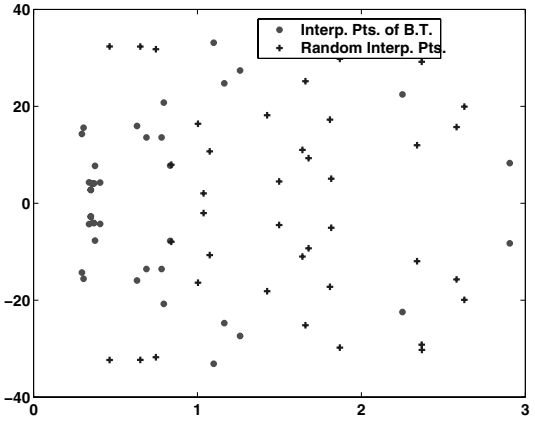


Fig. 6.3. Interpolation points for $R_{bt}(s)$, $R_{sokryl}(s)$ and $R_{randsokryl}(s)$

magnitude diagrams are undistinguishable, see Figure 6.4). This is a good news because they should all approximate well the same transfer function $R(s)$.

One observes from Table 6.1 that the SVD techniques perform better than the Krylov techniques. Two remarks are in order. First, it should be kept in mind that only the Krylov reduced transfer functions are directly computed from the original data of $R(s)$. Second, concerning the Krylov techniques, the quality of the approximation depends strongly on the choice of the inter-

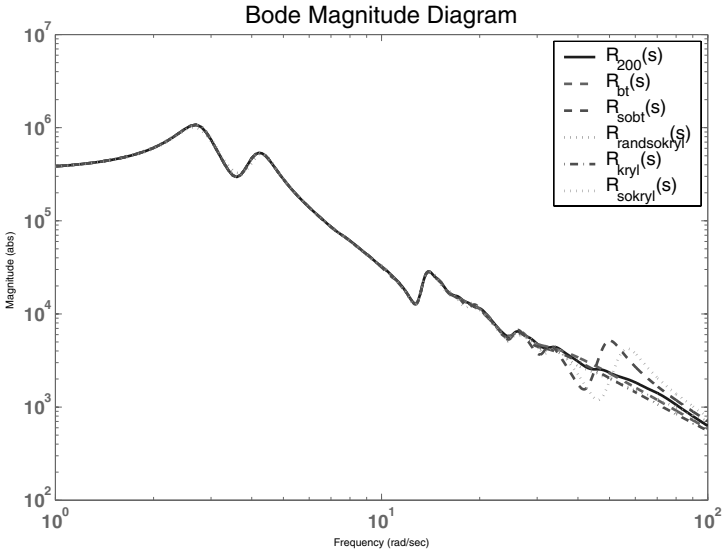


Fig. 6.4. The six transfer functions

Table 6.1. Relative errors for reduced order models

Reduced Transfer function	Model Reduction technique	McMillan degree	$\frac{\|R_{200}(s)-R_{reduced}(s)\|_\infty}{\|R_{200}(s)\|_\infty}$
$R_{bt}(s)$	Balanced Truncation	20	$4.3\ 10^{-4}$
$R_{sobt}(s)$	Second-Order Bal. Trunc.	40	$2.6\ 10^{-4}$
$R_{kryl}(s)$	Krylov	20	$8.3\ 10^{-4}$
$R_{sokryl}(s)$	Second-Order Krylov	28	$5.8\ 10^{-2}$
$R_{randsokryl}(s)$	Random Second-Order Krylov	20	$7\ 10^{-2}$

polation points. Because for SISO systems, any transfer function can be constructed from Krylov subspaces from any transfer function of larger McMillan degree, there should exist interpolation conditions that produce reduced order transfer functions with smaller error bound than what can be obtained with balanced techniques, but of course, it is not easy to find such interpolation conditions.

A surprising fact concerning SVD techniques is that the best approximation is obtained with $R_{sobt}(s)$ and not $R_{bt}(s)$. Nevertheless, one should not forget that the McMillan degree of $R_{sobt}(s)$ is twice as large as the McMillan degree of $R_{bt}(s)$.

In contrast with SVD techniques, the error obtained with the first order transfer function $R_{kryl}(s)$ is 100 times smaller than for the second-order transfer functions $R_{sokryl}(s)$ and $R_{randsokryl}(s)$. This tends to indicate that Second-Order Krylov techniques perform quite poorly compared to the first

order techniques, perhaps indicating that a more sophisticated algorithm for choosing the interpolation points for these methods is needed.

Finally, by choosing random interpolation points, the error remains roughly the same than by taking the *balanced truncation* interpolation points: 0.058 for $R_{sokryl}(s)$ and 0.07 for $R_{randsokryl}(s)$. This is probably due to the fact that the area chosen to generate the interpolation points for $R_{randsokryl}(s)$ contains good information about the original transfer function.

6.7 Concluding Remarks

Concerning the second-order Krylov technique, the following observation is worth mentioning. For SISO systems of pair Mc Millan degree, it has been shown in [BSGL04] and [MS96] that for every first order system (c, A, b) such that $cb = 0$, there exists a state space transformation that puts it into a second-order form. In other words, every SISO system (with first Markov parameter equal to zero) can be rewritten as a second-order system. This implies that in the SISO case, it is possible to impose $4n - 1$ interpolation conditions for a reduced second-order system of McMillan degree $2n$ by first using the standard Multipoint Padé technique of Theorem 6.2.3 and then reconstructing a second-order form with an appropriate state space coordinate transformation. Currently, no proof is available for the MIMO case.

As for generalized state space realizations of first order systems, it is also possible to apply Krylov techniques to second-order systems without requiring the mass matrix M to be equal to the identity. Concerning the SOBT technique, special care must taken in deriving the second-order Gramians.

For second-order balanced truncation, numerical results are very encouraging, but many important questions remain open. For instance, does there exist an a priori global error bound with SOBT, as for Balanced Truncation? Even simpler, is stability of the reduced system always guaranteed? If the answer to the preceding questions is negative, does there exist a better choice of second-order Gramians? Also, the development of an approximate version applicable to large scale systems is needed.

Acknowledgement

This paper presents research supported by NSF grants CCR-99-12415 and ACI-03-24944, and by the Belgian Programme on Inter-university Poles of Attraction, initiated by the Belgian State, Prime Minister's Office for Science, Technology and Culture. The scientific responsibility rests with its authors.

References

[Ant05] Antoulas, A.: *Lectures on the Approximation of Large-scale Dynamical Systems*. SIAM, Philadelphia, to appear (2005)

[BS04] Z. Bai and Y. Su. Dimension reduction of second order dynamical systems via a second-order arnoldi method. Technical Report CSE-2004-1, University of California, Davis, 2004.

[BSGL04] Bunse-Gerstner, A., Salimbahrami, B., Grotmaack, R. and Lohmann, B.: Existence and Computation of Second Order Reduced Systems using Krylov Subspace Methods. In: *Proceedings of 16th Symp. on the Mathematical Theory of Networks and Systems*, Leuven (2004)

[CGV04] Chahlaoui, Y., Gallivan, K. and Van Dooren, P.: The H_∞ norm calculation for large sparse systems. In: *Proceedings of 16th Symp. on the Mathematical Theory of Networks and Systems*, Leuven (2004)

[CLVV05] Chahlaoui, Y., Lemonnier, D., Vandendorpe, A. and Van Dooren, P.: Second-order balanced truncation. *Linear Algebra and its Applications*, to appear (2005)

[dVS87] de Villemagne, C. and Skelton, R.: Model reductions using a projection formulation. *Int. J. Control*, **46**, 2141–2169 (1987)

[Gri97] Grimme, E.: *Krylov Projection Methods for Model Reduction*. PhD thesis, University of Illinois, Urbana-Champaign, (1997)

[GVV03] Gallivan, K., Vandendorpe, A. and Van Dooren, P.: Model Reduction via truncation : an interpolation point of view. *Linear Algebra and its Applications*, **375**, 115–134 (2003)

[GVV04a] Gallivan, K., Vandendorpe, A. and Van Dooren, P.: Model reduction of MIMO systems via tangential interpolation. *SIAM Journal on Matrix Analysis and Applications*, **26(2)**, 328–349 (2004)

[GVV04b] Gallivan, K., Vandendorpe, A. and Van Dooren, P.: Sylvester equations and projection-based model reduction. *Journal of Computational and Applied Mathematics*, **162**, 213–229 (2004)

[MS96] Meyer, D. and Srinivasan, S.: Balancing and model reduction for second-order form linar systems. *IEEE Trans. Automat. Control*, **41(11)**, 1632–1644 (1996)

[Pre97] Preumont, A.: *Vibration Control of Active Structures*. Kluwer Academic Publishers, Dordrecht (1997)

[SA02] Sorensen, D. and Antoulas, A.: The Sylvester equation and approximate balanced reduction. *Linear Algebra and its Applications*, **351-352**, 671–700 (2002)

[SC91] Su, T. and Craig, J.: Model reduction and control of flexible structures using krylov vectors. *J. Guidance Control Dynamics*, **14(2)**, 1311–1313 (1991)

[VV04] Vandendorpe, A. and Van Dooren, P.: *Krylov techniques for model reduction of second-order systems*. Int. Rept. CESAME TR07-2004, Université catholique de Louvain, Louvain-la-Neuve (2004)

[WJJ87] Weaver, W. and Johnston, P.: *Structural Dynamics by Finite Elements*. Prentice Hall, Upper Saddle River (1987)

[ZDG95] Zhou, K., Doyle, J. and Glover, K.: *Robust and Optimal Control*. Prentice Hall, Upper Saddle River (1995)

7

Arnoldi Methods for Structure-Preserving Dimension Reduction of Second-Order Dynamical Systems

Zhaojun Bai[1], Karl Meerbergen[2], and Yangfeng Su[3]

[1] Department of Computer Science and Department of Mathematics, University of California, Davis, CA 95616, USA, bai@cs.ucdavis.edu
[2] Free Field Technologies, place de l'Université 16, 1348 Louvain-la-Neuve, Belgium, Karl.Meerbergen@fft.be
[3] Department of Mathematics, Fudan University, Shanghai 2200433, P. R. China, yfsu@fudan.edu.cn

7.1 Introduction

Consider the multi-input multi-output (MIMO) time-invariant second-order problem

$$\Sigma_N : \begin{cases} M\ddot{q}(t) + D\dot{q}(t) + Kq(t) = F\,u(t) \\ y(t) = L^T q(t) \end{cases} \tag{7.1}$$

with initial conditions $q(0) = q_0$ and $\dot{q}(0) = \dot{q}_0$. Here t is the time variable. $q(t) \in \mathbb{R}^N$ is a vector of state variables. N is the state-space dimension. $u(t)$ and $y(t)$ are the input force and output measurement functions, respectively. M, D, $K \in \mathbb{R}^{N\times N}$ are system matrices, such as mass, damping and stiffness as known in structural dynamics, and acoustics. We have $F \in \mathbb{R}^{N\times p}$ and $L \in \mathbb{R}^{N\times m}$ are input distribution and output measurement matrices, respectively.

Second-order systems Σ_N of the form (7.1) arise in the study of many types of physical systems, with common examples being electrical, mechanical and structural systems, electromagnetics and microelectromechanical systems (MEMS) [Cra81, Bal82, CZB+00, BBC+00, RW00, Slo02, WMSW02].

We are concerned with the system Σ_N of very large state-space dimension N. The analysis and design of large models becomes unfeasible with reasonable computing resources and computation time. It is necessary to obtain a reduced-order model which retains important properties of the original system, and yet is efficient for practical use. A common approach for reduced-order modeling is to first rewrite Σ_N as a mathematically equivalent linear system and then apply linear system dimension reduction techniques, such as explicit and implicit moment-matching and balanced truncation. The reader can find surveys of these methods, for example, in [Fre00, ASG01, Bai02].

There are two major drawbacks with such a linearization approach: the corresponding linear system has a state space of double dimension which increases memory requirements, and the reduced system is typically linear and the second-order structure of the original system is not preserved.

The preservation of the second-order structure is important for physical interpretation of the reduced system in applications. In addition, respecting the second-order structure also leads more stable, accurate and efficient reduced systems. This book contains three chapters on second (or higher) order systems. Chapter 6 discusses Krylov-subspace based and SVD-based methods for second-order structure preserving model reduction. For the Krylov-subspace based techniques, conditions on the projectors that guarantee the reduced second-order system tangentially interpolates the original system at given frequencies are derived. For SVD-based techniques, a second-order balanced truncation method is derived from second order gramians. Chapter 8 presents Krylov methods based on projections onto a subspace which is spanned by a properly partitioned Krylov basis matrices obtained by applying standard Krylov-subspace techniques to an equivalent linearized system. In this chapter, we present modified Arnoldi methods which are specifically designed for the second-order system, without via linearization. We call them as second-order Krylov subspace techniques. In a unified style, we will review recently developed Arnoldi-like dimension reduction methods that preserve the second-order structure. We will focus on the presentation of essential ideas behind these methods, without going into details on elaborate issues on robustness and stability of implementations and others.

For simplicity, we only consider the single-input single output (SISO) system in this paper. Denote $F = f$ and $L = l$, where f and l are column vectors of dimension N. The extension to the MIMO case requires block Arnoldi-like methods, which is beyond the scope this paper. The matrices M, D, and K often have particular properties such as symmetry, skew-symmetry, and positive (semi-)definiteness. We do not exploit nor assume any of such properties. We only assume that K is invertible. If this would not be the case, we assume there is an $s_0 \in \mathbb{R}$ so that $s_0^2 M + s_0 D + K$ is nonsingular.

7.2 Second-Order System and Dimension Reduction

The second-order system Σ_N of the form (7.1) is the representation of Σ_N in the time domain, or the state space. Equivalently, one can also represent the system in the frequency domain via the Laplace transform. Under the assumption of the initial conditions $q(0) = q_0 = 0$ and $\dot{q}(0) = \dot{q}_0 = 0$ and $u(0) = 0$. Then the input $U(s)$ and output $Y(s)$ in the frequency domain are related by the *transfer function*

$$H(s) = l^T (s^2 M + sD + K)^{-1} f, \tag{7.2}$$

where physically meaningful values of the complex variable s are $s = \mathtt{i}\omega$, $\omega \geq 0$ is referred to as the *frequency*. The power series expansion of $H(s)$ is formally given by

$$H(s) = m_0 + m_1 s + m_2 s^2 + \cdots = \sum_{\ell=0}^{\infty} m_\ell s^\ell,$$

where m_ℓ for $\ell \geq 0$ are called *moments*. The moment m_ℓ can be expressed as the inner product of the vectors l and r_ℓ:

$$m_\ell = l^T r_\ell \quad \text{for} \quad \ell \geq 0, \tag{7.3}$$

where the vector sequence $\{r_\ell\}$ is defined by the following linear homogeneous second-order recurrence relation

$$\begin{aligned} r_0 &= K^{-1} b \\ r_1 &= -K^{-1} D r_0 \\ r_\ell &= -K^{-1}(D r_{\ell-1} + M r_{\ell-2}) \quad \text{for} \quad \ell = 2, 3, \ldots \end{aligned} \tag{7.4}$$

As mentioned above, we assume that K is nonsingular, otherwise, see the discussion in section 5. The vector sequence $\{r_\ell\}$ is called a *second-order Krylov vector sequence*. Correspondingly, the subspace spanned by the vector sequence $\{r_\ell\}$ is called a *second-order Krylov subspace*:

$$\mathcal{G}_n(A, B; r_0) = \text{span}\{r_0, r_1, r_2, \ldots r_{n-1}\}, \tag{7.5}$$

where $A = -K^{-1}D$ and $B = -K^{-1}M$. When the matrices A and B, i.e., the matrices M, D and K, and r_0 are known from the context, we will drop them in our notation, and simply write $\mathcal{G}_n$.

Let Q_n be an orthonormal basis of $\mathcal{G}_n$, i.e.,

$$\mathcal{G}_n = \text{span}\{Q_n\} \text{ and } Q_n^T Q_n = I.$$

An orthogonal projection technique of dimension reduction onto the subspace $\mathcal{G}_n$ seeks an approximation of $q(t)$, constrained to stay in the subspace spanned by the columns of Q_n, namely

$$q(t) \approx Q_n z(t) \ .$$

This is often referred to as the *change-of-state coordinates*. Then by imposing the so-called Galerkin condition:

$$M Q_n \ddot{z}(t) + D Q_n \dot{z}(t) + K Q_n z(t) - f\, u(t) \ \perp \ \mathcal{G}_n,$$

we obtain the following reduced-order system:

$$\Sigma_n : \begin{cases} M_n \ddot{z}_n(t) + D_n \dot{z}_n(t) + K_n z(t) = f_n\, u(t) \\ \tilde{y}(t) = l_n^T z(t) \end{cases}, \tag{7.6}$$

where $M_n = Q_n^T M Q_n$, $D_n = Q_n^T D Q_n$, $K_n = Q_n^T K Q_n$, $f_n = Q_n^T f$ and $l_n = Q_n^T l$. We note that by explicitly formulating the matrices M_n, D_n and K_n in Σ_n, essential structures of M, D and K are preserved. For example, if M is symmetric positive definite, so is M_n. As a result, we can preserve the stability, symmetry and physical meaning of the original second-order system Σ_N. This is in the same spirit of the widely used PRIMA algorithm for passive reduced-order modeling of linear dynamical systems arising from interconnect analysis in circuit simulations [OCP98].

The use of the second-order Krylov subspace $\mathcal{G}_n$ for structure-preserving dimension reduction of the second-order system Σ_N has been studied by Su and Craig back to 1991 [SCJ91], although the subspace $\mathcal{G}_n$ is not explicitly defined and exploited as presented here. It has been revisited in recent years [RW00, Bai02, Slo02, BS04a, SL04, MR03]. It has been applied to very large second-order systems from structural analysis and MEMS simulations. The work of Meyer and Srinivasan [MS96] is an extension of balancing truncation methods for the second-order system. Recent such effort includes [CLM$^+$02]. Another structure-preserving model reduction technique is recently presented in [GCFP03]. Those two approaches focus on the application of moderate size second-order systems.

The transfer function $h_n(s)$ and moments $m_\ell^{(n)}$ of the reduced second-order system Σ_n in (7.6) are defined similar to the ones of the original system Σ_N, namely,

$$h_n(s) = l_n^T (s^2 M_n + s D_n + K_n)^{-1} f_n$$

and

$$m_\ell^{(n)} = l_n^T r_\ell^{(n)} \quad \text{for} \quad \ell \geq 0 \ ,$$

where $r_\ell^{(n)}$ are the second-order Krylov vectors as defined in (7.4) associated with the matrices M_n, D_n and K_n.

One way to assess the quality of the approximation is by comparing the number of moments matched between the original system Σ_N and the reduced-order system Σ_n. The following theorem shows that the structure-preserving reduced system Σ_n matches as many moments as the linearization approach (see section 3). A rigorous proof of the theorem can be found in [BS04a].

Moment-matching Theorem. *The first n moments of the original system Σ_N in (7.1) and the reduced system Σ_n in (7.6) are matched, i.e., $m_\ell = m_\ell^{(n)}$ for $\ell = 0, 1, 2, \ldots, n-1$. Hence $h_n(s)$ is an n-th Padé-type approximant of the transfer function $h(s)$:*

$$h(s) = h_n(s) + \mathcal{O}(s^n).$$

Furthermore, if the original system Σ_N is symmetric, i.e., M, D and K are symmetric and $f = l$, then the first $2n$ moments of $h(s)$ and $h_n(s)$ are equal and $h_n(s)$ is an n-th Padé approximant of $h(s)$:

$$h(s) = h_n(s) + \mathcal{O}(s^{2n}).$$

The gist of structure-preserving dimension reduction of the second-order system Σ_N is now on how to efficiently compute an orthonormal basis Q_n of the second-order Krylov subspace $\mathcal{G}_n$. In section 4, we will discuss recently developed Arnoldi-like procedures for computing such an orthonormal basis.

7.3 Linearization Method

In this section, we review the Arnoldi-based linearization approach for the dimension reduction of Σ_N. By exploiting the underlying second-order structure of this approach, it leads to the recently proposed structure-preserving methods to be discussed in the following sections.

It is easy to see that the original second-order system Σ_N is mathematically equivalent to the following linear system:

$$\Sigma_N^L : \begin{cases} C\dot{x}(t) + Gx(t) = \widehat{f}\, u(t) \\ y(t) = \widehat{l}^T x(t) \end{cases}, \tag{7.7}$$

where

$$x(t) = \begin{bmatrix} q(t) \\ \dot{q}(t) \end{bmatrix},\ C = \begin{bmatrix} D & M \\ -Z & 0 \end{bmatrix},\ G = \begin{bmatrix} K & 0 \\ 0 & Z \end{bmatrix},\ \widehat{f} = \begin{bmatrix} f \\ 0 \end{bmatrix},\ \widehat{l} = \begin{bmatrix} l \\ 0 \end{bmatrix}. \tag{7.8}$$

Z is an arbitrary $N \times N$ nonsingular matrix.

An alternative linear system can be defined by the following system matrices

$$x(t) = \begin{bmatrix} q(t) \\ \dot{q}(t) \end{bmatrix},\ C = \begin{bmatrix} 0 & M \\ -Z & 0 \end{bmatrix},\ G = \begin{bmatrix} K & D \\ 0 & Z \end{bmatrix},\ \widehat{f} = \begin{bmatrix} f \\ 0 \end{bmatrix},\ \widehat{l} = \begin{bmatrix} l \\ 0 \end{bmatrix}. \tag{7.9}$$

Various linearizations have been proposed in the literature, see [TM01] for a survey. We consider the above two, since they can be used in the methods we discuss. The linearization discussed by [MW01] does not fit in this framework.

Note that both linearizations produce

$$-G^{-1}C = \begin{bmatrix} -K^{-1}D & K^{-1}M \\ I & 0 \end{bmatrix}. \tag{7.10}$$

The zero block in (7.10) is very important for Arnoldi-like methods discussed in this paper.

Let $\mathcal{K}_n(-G^{-1}C; \widehat{r}_0)$ denote the Krylov subspace based on the matrix $-G^{-1}C$ and the starting vector $\widehat{r}_0 = G^{-1}\widehat{f}$:

$$\mathcal{K}_n(-G^{-1}C; \widehat{r}_0) = \text{span}\{\widehat{r}_0, (-G^{-1}C)\widehat{r}_0, \ldots, (-G^{-1}C)^{n-1}\widehat{r}_0\}.$$

The following Arnoldi procedure is a popular numerically stable procedure to generate an orthonormal basis V_n of the Krylov subspace $\mathcal{K}_n(-G^{-1}C; \widehat{r}_0) \subseteq \mathbb{R}^{2N}$, namely,

$$\text{span}\{V_n\} = \mathcal{K}_n(-G^{-1}C; \widehat{r}_0)$$

and $V_n^T V_n = I$.

Algorithm 1 *Arnoldi procedure*

Input: $C, G, \widehat{f}, n$
Output: V_n
1. $v_1 = G^{-1}\widehat{f}/\|G^{-1}\widehat{f}\|_2$
2. *for* $j = 1, 2, \ldots, n$ *do*
3. $\quad r = -G^{-1}Cv_j$
4. $\quad h_j = V_j^T r$
5. $\quad r = r - V_j h_j$
6. $\quad h_{j+1,j} = \|r\|_2$
7. $\quad$ *stop if* $h_{j+1,j} = 0$
8. $\quad v_{j+1} = r/h_{j+1,j}$
9. *end for*

The governing equation of the Arnoldi procedure is

$$(-G^{-1}C)\, V_n = V_{n+1}\widehat{H}_n, \tag{7.11}$$

where $\widehat{H}_n = (h_{ij})$ is an $(n+1) \times n$ upper Hessenberg matrix and $V_{n+1} = [V_n\ v_{n+1}]$ is a $2N \times (n+1)$ matrix with orthonormal columns. By making the use of the orthonormality of the columns of V_{n+1}, it follows that

$$V_n^T(-G^{-1}C)\, V_n = H_n$$

where H_n is the $n \times n$ leading principal submatrix of $\widehat{H}_n$.

By the framework of an orthogonal projection dimension reduction technique, one seeks an approximation of $x(t)$, constrained to the subspace spanned by the columns of V_n, namely

$$x(t) \approx V_n z(t).$$

Then by imposing the so-called Galerkin condition:

$$G^{-1}CV_n\dot{z}(t) + V_n z(t) - G^{-1}\widehat{f}\, u(t) \ \perp\ \text{span}\{V_n\},$$

we obtain the following reduced-order system in linear form:

$$\Sigma_n^L:\ \begin{cases} C_n\dot{z}(t) + G_n z(t) = \widehat{f}_n\, u(t) \\ \tilde{y}(t) = \widehat{l}_n^T z(t) \end{cases} \tag{7.12}$$

where $C_n = -H_n$, $G_n = I_n$, $\widehat{f}_n = e_1\|G^{-1}\widehat{f}\|_2$, and $\widehat{l}_n = V_n^T\widehat{l}$.

It can be shown that the reduced linear system Σ_n^L matches the first n moments of the original linear system Σ_N^L, which are equal to the first n moments of the original second-order system Σ_N. In finite precision arithmetic, reorthogonalization may lead to a smaller order for the same precision, see [Mee03]. The major disadvantages of this method include doubling the storage requirement, and the loss of the second-order structure for the reduced-order model.

We note that the Arnoldi procedure breaks down when $h_{j+1,j} = 0$ at iteration j. This happens if and only if the starting vector $\widehat{r}_0$ is a linear combination of j eigenvectors of $-G^{-1}C$. In addition, $\mathcal{K}_j(-G^{-1}C; \widehat{r}_0)$ is an invariant subspace and $\mathcal{K}_k(-G^{-1}C; \widehat{r}_0) = \mathcal{K}_j(-G^{-1}C; \widehat{r}_0)$ for all $k \geq j$. It can be shown that at the breakdown, the moments of the reduced-order system are identical to those of the original system, i.e., $h(s) \equiv h_j(s)$. Therefore, the breakdown is considered as a rare but lucky situation.

7.4 Modified Arnoldi Procedures

Define the Krylov matrix K_n by

$$K_n = [\, \widehat{r}_0,\ (-G^{-1}C)\widehat{r}_0,\ (-G^{-1}C)^2\widehat{r}_0,\ \ldots,\ (-G^{-1}C)^{n-1}\widehat{r}_0\,].$$

It is easy to see that the Krylov matrix K_n can be rewritten in the following form:

$$K_n = \begin{bmatrix} r_0 & r_1 & r_2 & \cdots & r_{n-1} \\ 0 & r_0 & r_1 & \cdots & r_{n-2} \end{bmatrix}, \tag{7.13}$$

where the vectors $\{r_0, r_1, r_2, \ldots, r_{n-1}\}$ are defined by the second-order recurrences (7.4). It is well-known, for example see [Ste01, section 5.1], that the orthonormal basis V_n, generated by the Arnoldi procedure (Algorithm 1), is the orthogonal Q-factor of the QR factorization of the Krylov matrix K_n:

$$K_n = V_n R_n, \tag{7.14}$$

where R_n is some $n \times n$ upper triangular matrix. Partition V_n into the 2×1 block matrix

$$V_n = \begin{bmatrix} U_n \\ W_n \end{bmatrix},$$

then equation (7.14) can be written in the form

$$\begin{bmatrix} r_0 & r_1 & r_2 & \cdots & r_{n-1} \\ 0 & r_0 & r_1 & \cdots & r_{n-2} \end{bmatrix} = \begin{bmatrix} U_n \\ W_n \end{bmatrix} R_n.$$

It shows that we can generate an orthonormal basis Q_n of $\mathcal{G}_n$ by orthonormalizing the U-block vectors or the W-block vectors. The leads to the Q-Arnoldi method to be described in §7.4.1. The SOAR in §7.4.2 is a procedure to compute the orthonormal basis Q_n directly, without computing the U- or W-block first.

Before we present these procedures, we note that one can show that the Krylov subspace $\mathcal{K}_n(-G^{-1}C; \widehat{r}_0)$ can be *embedded* in the second-order Krylov subspace $\mathcal{G}_n(A, B; r_0)$, namely

$$\text{span}\{V_n\} \subseteq \text{span}\left\{ \begin{bmatrix} Q_n & 0 \\ 0 & Q_n \end{bmatrix} \right\}.$$

This is a very useful observation and can be applied to a number of cases, for example, to prove the moment-matching theorem. See [BS04a] for details.

7.4.1 Q-Arnoldi Procedure

Recall from (7.10) that

$$-G^{-1}C = \begin{bmatrix} -K^{-1}D & -K^{-1}M \\ I & 0 \end{bmatrix}.$$

From the second block row of the governing equation (7.11) of the Arnoldi procedure, we have

$$U_n = W_{n+1}\widehat{H}_n. \tag{7.15}$$

We can exploit this relation to avoid the storage of the U-vectors with a slight increase of the computational cost. All products with U_n are to be replaced by the products of W_{n+1} and $\widehat{H}_n$. This observation has been made in [MR03] for the solution of the quadratic eigenvalue problem and parametrized equations. With the motivation of constructing an orthonormal basis of the second-order Krylov subspace $\mathcal{G}_n$, we derive the following algorithm.

Algorithm 2 *Q-Arnoldi procedure (W-version)*

Input: M, D, K, r_0, n
Output: Q_n
1. $u = r_0/\|r_0\|_2$ *and* $w_1 = 0$
2. *for* $j = 1, 2, \ldots, n$ *do*
3. $\quad r = -K^{-1}(Du + Mw_j)$
4. $\quad t = u$
5. $\quad h_j = \begin{bmatrix} \widehat{H}_{j-1}^T(W_j^T r) + W_{j-1}^T t \\ u^T r + w_j^T t \end{bmatrix}$
6. $\quad r = r - \begin{bmatrix} W_j & u \end{bmatrix} \left(\begin{bmatrix} \widehat{H}_{j-1} & 0 \\ 0 & 1 \end{bmatrix} h_j \right)$
7. $\quad t = t - W_j h_j$
8. $\quad h_{j+1,j} = (\|r\|_2^2 + \|t\|_2^2)^{1/2}$
9. $\quad$ *stop if* $h_{j+1,j} = 0$
10. $\quad u = r/h_{j+1,j}$
11. $\quad w_{j+1} = t/h_{j+1,j}$
12. *end for*
13. $Q_{n+1} = \texttt{orth}([W_{n+1} \quad u])$ $\quad$ *% orthogonalization*

We note that the function $\texttt{orth}(X)$ in step 13 stands for the modified Gram-Schmidt process or QR decomposition for generating an orthonormal basis for the range of X.

An alternative approach of the Q-Arnoldi method is to avoid the storage of the W-vectors. By equation (7.15) and noting that $w_1 = 0$, we have

$$W_{n+1}(:, 2:n+1) = U_n \widehat{H}(2:n+1, 1:n)^{-1}\ . \tag{7.16}$$

Operations with W_n can then use the expression (7.16). We obtain another modified Arnoldi procedure.

Algorithm 3 *Q-Arnoldi procedure (U-version)*

Input: M, D, K, r_0, n
Output: Q_n
1. $u_1 = r_0/\|r_0\|_2$ *and* $w = 0$
2. *for* $j = 1, 2, \ldots, n$ *do*
3. $r = -K^{-1}(Du_j + Mw)$
4. $t = u_j$
5. $h_j = U_j^T r + \begin{bmatrix} 0 \\ \widehat{H}(2:j, 1:j-1)^{-T} U_{j-1}^T \end{bmatrix} t$
6. $r = r - U_j h_j$
7. $t = t - \begin{bmatrix} 0 \; U_{j-1} \widehat{H}(2:j, 1:j-1)^{-1} \end{bmatrix} h_j$
8. $h_{j+1,j} = (\|r\|_2^2 + \|t\|_2^2)^{1/2}$
9. *stop if* $h_{j+1,j} = 0$
10. $u_{j+1} = r/h_{j+1,j}$
11. $w = t/h_{j+1,j}$
12. *end for*
13. $Q_{n+1} = \texttt{orth}(U_{n+1})$ *% orthogonalization*

Note that both modified Arnoldi procedures 2 and 3 produce the same $\widehat{H}_n$ as the Arnoldi procedure in exact arithmetic. If we would compute the U block using (7.15) after the execution of Algorithm 2, we would obtain exactly the same U block as the one produced by Algorithm 3. The breakdown of both Q-Arnoldi procedures happens in the same situation as the standard Arnoldi procedure.

7.4.2 Second-Order Arnoldi Procedure

The Second-Order ARnoldi (SOAR) procedure computes an orthonormal basis of the second-order Krylov subspace $\mathcal{G}_n$ directly, without first computing the U- or W-block. It is based on the observation that the elements of the upper Hessenberg matrix $\widehat{H}_n$ in the governing equation (7.11) of the Arnoldi procedure can be chosen to enforce the orthonormality of the U-vectors directly. The procedure is first proposed by Su and Craig [SCJ91], and further improved in the recent work of Bai and Su [BS04b]. The simplest version of the procedure is as follows.

Algorithm 4 *SOAR procedure*

Inputs: M, D, K, r_0, n
Output: Q_n
1. $q_1 = r_0/\|r_0\|$
2. $w = 0$
3. *for* $j = 1, 2, \ldots, n$ *do*
4. $r = -K^{-1}(Dq_j + Mw)$
5. $h_j = Q_j^T r$
6. $r := r - Q_j h_j$
7. $h_{j+1\,j} = \|r\|_2$
8. *stop if* $h_{j+1\,j} = 0$,
9. $q_{j+1} = r/h_{j+1\,j}$
10. *solve* $\widehat{H}_j(2 : j+1, 1 : j)g = e_j$ *for* g
11. $w = Q_j g$
12. *end for*

Special attention needs to be paid to the case of breakdown for the SOAR procedure. This occurs when $h_{j+1\,j} = 0$ at iteration j. There are two possible cases. One is that the vector sequence $\{r_i\}_{i=0}^{j-1}$ is linearly dependent, but the double length vector sequence $\{[\,r_i^T\ r_{i-1}^T\,]^T\}_{i=0}^{j-1}$ is linearly independent. We call this situation *deflation*. With a proper treatment, the SOAR procedure can continue. Deflation is regarded as an advantage of the SOAR procedure. A modified SOAR procedure with the treatment of deflation is presented in [BS04b]. Another possible case is that both vector sequences $\{r_i\}_{i=0}^{j-1}$ and $\{[\,r_i^T\ r_{i-1}^T\,]^T\}_{i=0}^{j-1}$ are linearly dependent, respectively. In this case, the SOAR procedure terminates. We call this *breakdown*. At the breakdown of the SOAR, one can prove that the transfer functions $h(s)$ and $h_j(s)$ of the original system Σ_N and the reduced system Σ_j are identical, the same as in the linearization method [BS04a].

7.4.3 Complexity

Table 7.1 summarizes the memory requirements and computational costs of the Arnoldi and modified procedures discussed in this section.

Table 7.1. Complexity of Arnoldi procedure and modifications

Procedure	memory	flops
Arnoldi	$2(n+1)N$	$2Nn(n+3)$
Q-Arnoldi (W-version)	$(n+1)N$	$2Nn(n+1)$
Q-Arnoldi (U-version)	$(n+2)N$	$2Nn(n+3)$
SOAR	$(n+2)N$	$(3/2)Nn(n+4/3)$

We only consider the storage of the Arnoldi vectors, since this is the dominant factor. The storage of Q_{n+1} in W-version of the Q-Arnoldi procedure (Algorithm 2) uses the same locations as W_{n+1} and in U-version procedure (Algorithm 3) the same locations as U_{n+1}. The storage of w_1 is not required since it is zero. This explains the slightly lower cost for the W-version of Q-Arnoldi procedure.

For the computational costs, first note that the matrix-vector products involving matrices M, D and K are typically far more expensive than the other operations. All three procedure use the same number of matrix-vector products. The remaining cost is dominated by the orthogonalization procedures. For the Q-Arnoldi procedures, the cost is dominated by the inner products with W_j and U_j respectively. The cost of the U-version is slightly higher, because w_1 is zero. For SOAR, we assume that there are no zero columns in Q_{n+1}. These costs do not include the computation of Q_{n+1} in Step 13 of the Q-Arnoldi procedures 2 and 3. This cost is of the order of Nn^2.

7.5 Structure-Preserving Dimension Reduction Algorithm

We now present the Q-Arnoldi or SOAR-based method for structure-preserving dimension reduction of the second-order system Σ_N.

In practice, we are often interested in the approximation of the original system Σ_N around a prescribed expansion point $s_0 \neq 0$. In this case, the transfer function $h(s)$ of Σ_N can be written in the form:

$$\begin{aligned} h(s) &= l^T(s^2M + sD + K)^{-1}f \\ &= l^T((s-s_0)^2M + (s-s_0)\widetilde{D} + \widetilde{K})^{-1}f, \end{aligned}$$

where

$$\widetilde{D} = 2s_0M + D \quad \text{and} \quad \widetilde{K} = s_0^2M + s_0D + K.$$

Note that s_0 can be an arbitrary, but fixed value such that the matrix $\widetilde{K}$ is nonsingular. The moments of $h(s)$ about s_0 can be defined in a similar way as in (7.3).

By applying the Q-Arnoldi or SOAR procedure, we can generate an orthonormal basis Q_n of the second-order Krylov subspace $\mathcal{G}_n(A, B; r_0)$:

$$\text{span}\{Q_n\} = \mathcal{G}_n(A, B; r_0)$$

with

$$A = -\widetilde{K}^{-1}\widetilde{D},\ B = -\widetilde{K}^{-1}M \text{ and } r_0 = \widetilde{K}^{-1}f.$$

Following the orthogonal projection technique as discussed in section 2, the subspace spanned by the columns of Q_n can be used as the projection subspace, and subsequently, to define a reduced system Σ_n as in (7.6). The transfer function $h_n(s)$ of Σ_n about the expansion point s_0 is given by

$$h_n(s) = l_n^T((s-s_0)^2 M_n + (s-s_0)\widetilde{D}_n + \widetilde{K}_n)^{-1} f_n,$$

where $M_n = Q_n^T M Q_n$, $\widetilde{D}_n = Q_n^T \widetilde{D} Q_n$, $\widetilde{K}_n = Q_n^T \widetilde{K} Q_n$ and $l_n^T = Q_n^T l$ and $f_n^T = Q_n^T f$. By a straightforward algebraic manipulation, $h_n(s)$ can be simply expressed as

$$h_n(s) = l_n^T(s^2 M_n + sD_n + K_n)^{-1} f_n, \tag{7.17}$$

where

$$M_n = Q_n^T M Q_n,\ D_n = Q_n^T D Q_n,\ K_n = Q_n^T K Q_n,\ l_n = Q_n^T l,\ f_n = Q_n^T f.$$

In other words, the transformed matrix triplet $(M, \widetilde{D}, \widetilde{K})$ is used to generate an orthonormal basis Q_n of the projection subspace $\mathcal{G}_n$, but the original matrix triplet (M, D, K) is directly projected onto the subspace $\mathcal{G}_n$ to define a reduced system Σ_n about the selected expansion point s_0.

The moment-matching theorem in section 2 is still applied here. We can show that the first n moments about the expansion point s_0 of $h(s)$ and $h_n(s)$ are the same. Therefore, $h_n(s)$ is an n-th Padé-type approximant of $h(s)$ about s_0. Furthermore, if Σ_N is a symmetric second-order system, then the first $2n$ moments about s_0 of $h(s)$ and $h_n(s)$ are the same, which implies that $h_n(s)$ is an n-th Padé approximant of $h(s)$ about s_0.

The following algorithm is a high-level description of the second-order structure-preserving dimension reduction algorithm based on Q-Arnoldi or SOAR procedure.

Algorithm 5 *Structure-preserving dimension reduction algorithm*

1. *Select an order n for the reduced system, and an expansion point s_0.*
2. *Run n steps of Q-Arnoldi or SOAR procedure to generate an orthonormal basis Q_n of $\mathcal{G}_n(A, B; r_0)$, where $A = -\widetilde{K}^{-1}\widetilde{D}$, $B = -\widetilde{K}^{-1}M$ and $r_0 = \widetilde{K}^{-1} f$.*
3. *Compute $M_n = Q_n^T M Q_n$, $D_n = Q_n^T D Q_n$, $K_n = Q_n^T K Q_n$, $l_n = Q_n^T l$, and $f_n = Q_n^T f$. This defines a reduced system Σ_n as in (7.6) about the selected expansion point s_0.*

As we have noticed, by the definitions of the matrices M_n, D_n and K_n in the reduced system Σ_n, essential properties of the matrices M, D and K of the original system Σ_N are preserved. For example, if M is symmetric positive definite, so is M_n. Consequently, we can preserve stability, possible symmetry and the physical meaning of the original second-order system Σ_N.

The explicit formulation of the matrices M_n, D_n and K_n is done by using first matrix-vector product operations Mq, Dq and Kq for an arbitrary vector q and vector inner products. This is an overhead compared to the linearization method discussed in section 7.3. In the linearization method as described in section 3, the matrix $C_n = -H_n$ and $G_n = I$ in the reduced system Σ_n^L is obtained as a by-product of the Arnoldi procedure without additional cost. However, we believe that the preservation of the structure of the underlying

problem outweights the extra cost of floating point operations in a modern computing environment. In fact, we observed that this step takes only a small fraction of the total work, due to extreme sparsity of the matrices M and D and K in practical problems we encountered. The bottleneck of the computational costs is often associated with the matrix-vector product operations involving $\widetilde{K}^{-1}$.

7.6 Numerical Examples

In this section, we report our numerical experiments on the performance of the structure-preserving dimension reduction algorithm based on Q-Arnoldi and SOAR procedures. The superior numerical properties of the SOAR-based method over the linearization approach as described in section 3 have been reported in [BS04a] for examples from structural dynamics and MEMS systems. In this section, we focus on the performance of the Q-Arnoldi-based and SOAR-based structure-preserving dimension reduction methods. All numerical examples do not use reorthogonalization.

Example 1. This example is from the simulation of a linear-drive multi-mode resonator structure [CZP98]. This is a nonsymmetric second-order system. The mass and damping matrices M and D are singular. The stiffness matrix K is ill-conditioned due to the multi-scale of the physical units used to define the elements of K, such as the beam's length and cross sectional area, and its moment of inertia and modulus of elasticity. (See Chapter 21 for more details on this example.)

For this numerical experiment, the order of 1-norm condition number of K is at $\mathcal{O}(10^{15})$. We use the expansion point s_0 to approximate the bode plot of interest, the same as in [CZP98]. The condition number of the transformed stiffness matrix $\widetilde{K} = s_0^2 M + s_0 D + K$ is is slightly improved to $\mathcal{O}(10^{13})$. In Figure 7.1, the Bode plots of frequency responses of the original second-order system Σ_N of order $N = 63$, and the reduced-order systems of orders $n = 10$ via the Q-Arnoldi (W-version) and SOAR methods are reported. The corresponding relative errors are also shown over the frequency range of interest. From the relative errors, we see that the SOAR-based method is slightly more accurate than the Q-Arnoldi-based method.

Example 2. This is an example from an acoustic radiation problem discussed in [PA91]. Consider a circular piston subtending a polar angle $0 < \theta < \theta_p$ on a submerged massless and rigid sphere of radius δ. The piston vibrates harmonically with a uniform radial acceleration. The surrounding acoustic domain is unbounded and is characterized by its density ρ and sound speed c. (See Chapter 21 for more details on this example.)

We denote by p and a_r the prescribed pressure and normal acceleration respectively. In order to have a steady state solution $\tilde{p}(r, \theta, t)$ verifying

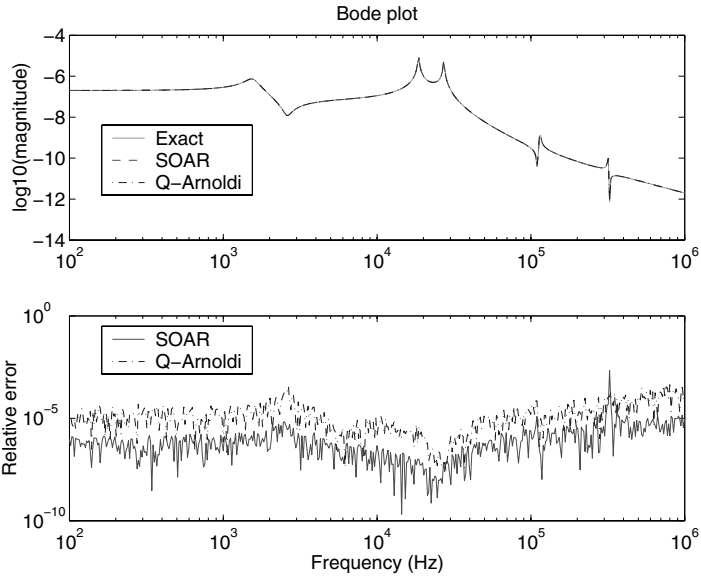


Fig. 7.1. Bode plots of $h(\mathtt{j}\omega)$ of the resonator, approximations by Q-Arnoldi and SOAR, and relative errors.

$$\tilde{p}(r,\theta,t) = \mathcal{R}\mathrm{e}\left(p(r,\theta)e^{i\omega t}\right),$$

the transient boundary condition is chosen as:

$$a_r = \frac{-1}{\rho}\frac{\partial p(r,\theta)}{\partial r}\bigg|_{r=a} = \begin{cases} a_0 \sin(\omega t), & 0 \le \theta \le \theta_p, \\ 0, & \theta > \theta_p. \end{cases}$$

The axisymmetric discrete finite-infinite element model relies on a mesh of linear quadrangle finite elements for the inner domain (region between spherical surfaces $r = \delta$ and $r = 1.5\delta$). The numbers of divisions along radial and circumferential directions are 5 and 80, respectively. The outer domain relies on conjugated infinite elements of order 5. For this example we used $\delta = 1(\mathrm{m})$, $\rho = 1.225(\mathrm{kg/m^3})$, $c = 340(\mathrm{m/s})$, $a_0 = 0.001(\mathrm{m/s^2})$ and $\omega = 1000(\mathrm{rad/s})$.

The matrices K, D, M and the right-hand side f are computed by ACTRAN [Fre03]. The dimension of the second-order system is $N = 2025$. For numerical tests, an expansion point $s_0 = 2 \times 10^2\pi$ is used. Figure 7.2 shows the magnitudes (in log of base 10) of the exact transfer function $h(s)$ and approximate ones computed by the Q-Arnoldi (W-version) and SOAR-based methods with the reduced dimension $n = 100$. For this example, the accuracy of two methods are essentially the same.

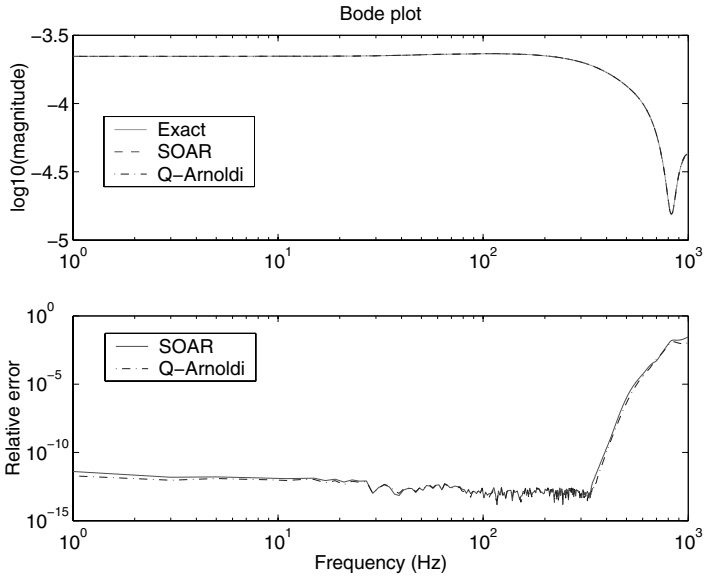


Fig. 7.2. Bode plot of $h(\mathrm{j}\omega)$ of ACTRAN2025, approximations by Q-Arnoldi and SOAR, and relative errors.

7.7 Conclusions

In this paper, using a unified style, we discussed the recent progress in the development of Arnoldi-like methods for structure-preserving dimension reduction of a second-order dynamical system Σ_N. The reduced second-order system Σ_n enjoys the same moment-matching properties as the Arnoldi-based algorithm via linearization. The major difference between the Q-Arnoldi and SOAR procedures lies in the orthogonalization.

We only focused on the basic schemes and the associated properties of structure-preserving algorithms. There are a number of interesting research issues for further study, such as numerical stability and the effect of reorthogonalization.

Acknowledgments ZB is supported in part by the National Science Foundation under Grant No. 0220104. YS is supported in part by NSFC research project No. 10001009 and NSFC research key project No. 90307017.

References

[ASG01] A. C. Antoulas, D. C. Sorensen, and S. Gugercin. *A survey of model reduction methods for large-scale systems.* Structured Matrices in Operator Theory, Numerical Analysis, Control, Signal and Image Processing. Contemporary Mathematics. AMS publications, 2001.

[Bai02] Z. Bai. Krylov subspace techniques for reduced-order modeling of large-scale dynamical systems. *Applied Numerical Mathematics*, 43:9–44, 2002.

[Bal82] M. J. Balas. Trends in large space structure control theory: fondest theory, wildest dreams. *IEEE Trans. Automat. Control*, AC-27:522–535, 1982.

[BBC+00] Z. Bai, D. Bindel, J. Clark, J. Demmel, K. S. J. Pister, and N. Zhou. New numerical techniques and tools in SUGAR for 3D MEMS simulation. In *Technical Proceedings of the Fourth International Conference on Modeling and Simulation of Microsystems*, pages 31–34, 2000.

[BS04a] Z. Bai and Y. Su. Dimension reduction of second-order dynamical systems via a second-order Arnoldi method. *SIAM J. Sci. Comp.*, 2004. to appear.

[BS04b] Z. Bai and Y. Su. SOAR: A second-order arnoldi method for the solution of the quadratic eigenvalue problem. *SIAM J. Matrix Anal. Appl.*, 2004. to appear.

[CLM+02] Y. Chahlaoui, D. Lemonnier, K. Meerbergen, A. Vandendorpe, and P. Van Dooren. Model reduction of second order systems. In *Proceedings of 15th International Symposium on Mathematical Theory of Networks and Systems*, University of Notre Dame, 2002.

[Cra81] R. R. Craig, Jr. *Structural Dynamics: An Introduction to Computer Methods*. John Wiley & Sons, 1981.

[CZB+00] J. V. Clark, N. Zhou, D. Bindel, L. Schenato, W. Wu, J. Demmel, and K. S. J. Pister. 3D MEMS simulation using modified nodal analysis. In *Proceedings of Microscale Systems: Mechanics and Measurements Symposium*, pages 68–75, 2000.

[CZP98] J. V. Clark, N. Zhou, and K. S. J. Pister. MEMS simulation using SUGAR v0.5. In *Proc. Solid-State Sensors and Actuators Workshop, Hilton Head Island, SC*, pages 191–196, 1998.

[Fre00] R. W. Freund. Krylov-subspace methods for reduced-order modeling in circuit simulation. *J. Comput. Appl. Math.*, 123:395–421, 2000.

[Fre03] Free Field Technologies. MSC.Actran 2003, User's Manual, 2003.

[GCFP03] S. D. Garvey, Z. Chen, M. I. Friswell, and U. Prells. Model reduction using structure-preserving transformations. In *Proceedings of the International Modal Analysis Conference IMAC XXI*, pages 361–377. Kissimmee, Florida, Feb., 2003.

[Mee03] K. Meerbergen. The solution of parametrized symmetric linear systems. *SIAM J. Matrix Anal. Appl.*, 24(4):1038–1059, 2003.

[MR03] K. Meerbergen and M. Robbé. The Arnoldi method for the solution of the quadratic eigenvalue problem and parametrized equations, 2003. Submitted for publication.

[MS96] D. G. Meyer and S. Srinivasan. Balancing and model reduction for second-order form linear systems. *IEEE Trans. Automatic Control*, 41:1632–1644, 1996.

[MW01] V. Mehrmann and D. Watkins. Structure-preserving methods for computing eigenpairs of large sparse skew-hamiltonian/hamiltonian pencils. *SIAM J. Matrix Anal. Applic.*, 22(6):1905–1925, 2001.

[OCP98] A. Odabasioglu, M. Celik, and L.T. Pileggi. PRIMA: passive reduced-order interconnect macromodeling algorithm. *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, 17:645–654, 1998.

[PA91] P. M. Pinsky and N. N. Abboud. Finite element solution of the transient exterior structural acoustics problem based on the use of radially asymptotic boundary conditions. *Computer Methods in Applied Mechanics and Engineering*, 85:311–348, 1991.

[RW00] D. Ramaswamy and J. White. Automatic generation of small-signal dynamic macromodels from 3-D simulation. In *Technical Proceedings of the Fourth International Conference on Modeling and Simulation of Microsystems*, pages 27–30, 2000.

[SCJ91] T.-J. Su and R. R. Craig Jr. Model reduction and control of flexible structures using Krylov vectors. *J. of Guidance, Control and Dynamics*, 14:260–267, 1991.

[SL04] B. Salimbahrami and B. Lohmann. Order reduction of large scale second order systems using Krylov subspace methods. *Lin. Alg. Appl.*, 2004. to appear.

[Slo02] R. D. Slone. *Fast frequency sweep model order reduction of polynomial matrix equations resulting from finite element discretization*. PhD thesis, Ohio State University, Columbus, OH, 2002.

[Ste01] G. W. Stewart. *Matrix Algorithms, Vol II: Eigensystems*. SIAM, Philadelphia, 2001.

[TM01] F. Tisseur and K. Meerbergen. The quadratic eigenvalue problem. *SIAM Rev.*, 43(2):235–286, 2001.

[WMSW02] T. Wittig, I. Munteanu, R. Schuhmann, and T. Weiland. Two-step Lanczos algorithm for model order reduction. *IEEE Trans. Magn.*, 38:673–676, 2002.

8

Padé-Type Model Reduction of Second-Order and Higher-Order Linear Dynamical Systems

Roland W. Freund

Department of Mathematics, University of California at Davis, One Shields Avenue, Davis, CA 95616, U.S.A.
freund@math.ucdavis.edu

Summary. A standard approach to reduced-order modeling of higher-order linear dynamical systems is to rewrite the system as an equivalent first-order system and then employ Krylov-subspace techniques for reduced-order modeling of first-order systems. While this approach results in reduced-order models that are characterized as Padé-type or even true Padé approximants of the system's transfer function, in general, these models do not preserve the form of the original higher-order system. In this paper, we present a new approach to reduced-order modeling of higher-order systems based on projections onto suitably partitioned Krylov basis matrices that are obtained by applying Krylov-subspace techniques to an equivalent first-order system. We show that the resulting reduced-order models preserve the form of the original higher-order system. While the resulting reduced-order models are no longer optimal in the Padé sense, we show that they still satisfy a Padé-type approximation property. We also introduce the notion of Hermitian higher-order linear dynamical systems, and we establish an enhanced Padé-type approximation property in the Hermitian case.

8.1 Introduction

The problem of model reduction is to replace a given mathematical model of a system or process by a model that is much smaller than the original model, yet still describes—at least approximately—certain aspects of the system or process. Model reduction involves a number of interesting issues. First and foremost is the issue of selecting appropriate approximation schemes that allow the definition of suitable reduced-order models. In addition, it is often important that the reduced-order model preserves certain crucial properties of the original system, such as stability or passivity. Other issues include the characterization of the quality of the models, the extraction of the data from the original model that is needed to actually generate the reduced-order models, and the efficient and numerically stable computation of the models.

In recent years, there has been a lot of interest in model-reduction techniques based on Krylov subspaces; see, for example, the survey pa-

pers [Fre97, Fre00, Bai02, Fre03]. The development of these methods was motivated mainly by the need for efficient reduction techniques in VLSI circuit simulation. An important problem in that application area is the reduction of very large-scale RCL subcircuits that arise in the modeling of the chip's wiring, the so-called *interconnect*. In fact, many of the Krylov-subspace reduction techniques that have been proposed in recent years are tailored to RCL subcircuits.

Krylov-subspace techniques can be applied directly only to first-order linear dynamical systems. However, there are important applications that lead to second-order, or even general higher-order, linear dynamical systems. For example, RCL subcircuits are actually second-order linear dynamical systems. The standard approach to employing Krylov-subspace techniques to the dimension reduction of a second-order or higher-order system is to first rewrite the system as an equivalent first-order system and then apply Krylov-subspace techniques for reduced-order modeling of first-order systems. While this approach results in reduced-order models that are characterized as Padé-type or even true Padé approximants of the system's transfer function, in general, these models do not preserve the form of the original higher-order system.

In this paper, we describe an approach to reduced-order modeling of higher-order systems based on projections onto suitably partitioned Krylov basis matrices that are obtained by applying Krylov-subspace techniques to an equivalent first-order system. We show that the resulting reduced-order models preserve the form of the original higher-order system. While the resulting reduced-order models are no longer optimal in the Padé sense, we show that they still satisfy a Padé-type approximation property. We further establish an enhanced Padé-type approximation property in the special case of Hermitian higher-order linear dynamical systems.

The remainder of the paper is organized as follows. In Section 8.2, we review the formulations of general RCL circuits as as first-order and second-order linear dynamical systems. In Section 8.3, we present our general framework for special second-order and higher-oder linear dynamical systems. In Section 8.4, we consider the standard reformulation of higher-order systems as equivalent first-order systems. In Section 8.5, we discuss some general concepts of dimension reduction of special second-order and general higher-order systems via dimension reduction of corresponding first-order systems. In Section 8.6, we review the concepts of block-Krylov subspaces and Padé-type reduced-order models. In Section 8.7, we introduce the notion of Hermitian higher-order linear dynamical systems, and we establish an enhanced Padé-type approximation property in the Hermitian case. In Section 8.8, we present the SPRIM algorithm for special second-order systems. In Section 8.9, we report the results of some numerical experiments with the SPRIM algorithm. Finally, in Section 8.10, we mention some open problems and make some concluding remarks.

Throughout this paper the following notation is used. Unless stated otherwise, all vectors and matrices are allowed to have real or complex entries.

For a complex number α or a complex matrix M, we denote its complex conjugate by $\overline{\alpha}$ or $\overline{M}$, respectively. For a matrix $M = \left[m_{jk}\right]$, $M^T := \left[m_{kj}\right]$ is the transpose of M, and $M^H := \overline{M}^T = \left[\overline{m_{kj}}\right]$ is the conjugate transpose of M. For a square matrix P, we write $P \succeq 0$ if $P = P^H$ is Hermitian and if P is positive semidefinite, i.e., $x^H P x \geq 0$ for all vectors x of suitable dimension. We write $P \succ 0$ if $P = P^H$ is positive definite, i.e., $x^H P x > 0$ for all vectors x, except $x = 0$. The $n \times n$ identity matrix is denoted by I_n and the zero matrix by 0. If the dimension of I_n is apparent from the context, we drop the index and simply use I. The actual dimension of 0 will always be apparent from the context. The sets of real and complex numbers are denoted by $\mathbb{R}$ and $\mathbb{C}$, respectively.

8.2 RCL Circuits as First-Order and Second-Order Systems

An important class of electronic circuits is linear RCL circuits that contain only resistors, capacitors, and inductors. For example, such RCL circuits are used to model the interconnect of VLSI circuits; see, e.g., [CLLC00, KGP94, OCP98]. In this section, we briefly review the RCL circuit equations and their formulations as first-order and second-order linear dynamical systems.

8.2.1 RCL Circuit Equations

General electronic circuits are usually modeled as networks whose branches correspond to the circuit elements and whose nodes correspond to the interconnections of the circuit elements; see, e.g., [VS94]. Such networks are characterized by *Kirchhoff's current law* (KCL), *Kirchhoff's voltage law* (KVL), and the *branch constitutive relations* (BCRs). The Kirchhoff laws depend only on the interconnections of the circuit elements, while the BCRs characterize the actual elements. For example, the BCR of a linear resistor is Ohm's law. The BCRs are linear equations for simple devices, such as linear resistors, capacitors, and inductors, and they are nonlinear equations for more complex devices, such as diodes and transistors.

The connectivity of such a network can be captured using a directional graph. More precisely, the nodes of the graph correspond to the nodes of the circuit, and the edges of the graph correspond to each of the circuit elements. An arbitrary direction is assigned to graph edges, so one can distinguish between the source and destination nodes. The adjacency matrix, A, of the directional graph describes the connectivity of a circuit. Each row of A corresponds to a graph edge and, therefore, to a circuit element. Each column of A corresponds to a graph or circuit node. The column corresponding to the datum (ground) node of the circuit is omitted in order to remove redundancy. By convention, a row of A contains $+1$ in the column corresponding to the

source node, -1 in the column corresponding to the destination node, and 0 everywhere else. Kirchhoff's laws can be expressed in terms of A as follows:

$$\begin{aligned} \text{KCL:} \quad & A^T i_b = 0, \\ \text{KVL:} \quad & A v_n = v_b. \end{aligned} \tag{8.1}$$

Here, the vectors i_b and v_b contain the branch currents and voltages, respectively, and v_n the non-datum node voltages.

We now restrict ourselves to linear RCL circuits, and for simplicity, we assume that the circuit is excited only by current sources. In this case, A, v_b, and i_b can be partitioned according to circuit-element types as follows:

$$A = \begin{bmatrix} A_i \\ A_g \\ A_c \\ A_l \end{bmatrix}, \quad v_b = v_b(t) = \begin{bmatrix} v_i \\ v_g \\ v_c \\ v_l \end{bmatrix}, \quad i_b = i_b(t) = \begin{bmatrix} i_i \\ i_g \\ i_c \\ i_l \end{bmatrix}. \tag{8.2}$$

Here, the subscripts i, g, c, and l stand for branches containing current sources, resistors, capacitors, and inductors, respectively. Using (8.2), the KCL and KVL equations (8.1) take on the following form:

$$\begin{aligned} A_i^T i_i + A_g^T i_g + A_c^T i_c + A_l^T i_l = 0, \\ A_i v_n = v_i, \quad A_g v_n = v_g, \quad A_c v_n = v_c, \quad A_l v_n = v_l. \end{aligned} \tag{8.3}$$

Furthermore, the BCRs can be stated as follows:

$$i_i = -I(t), \quad i_g = G v_g, \quad i_c = C \frac{d}{dt} v_c, \quad v_l = L \frac{d}{dt} i_l. \tag{8.4}$$

Here, $I(t)$ is the vector of current-source values, $G \succ 0$ and $C \succ 0$ are diagonal matrices whose diagonal entries are the conductance and capacitance values of the resistors and capacitors, respectively, and $L \succeq 0$ is the inductance matrix. In the absence of inductive coupling, L is also a diagonal matrix, but in general, L is a full matrix. However, an important special case is inductance matrices L whose inverse, the so-called susceptance matrix, $S = L^{-1}$ is sparse; see [ZKBP02, ZP02].

Equations (8.3) and (8.4), together with initial conditions for $v_n(t_0)$ and $i_l(t_0)$ at some initial time t_0, provide a complete description of a given RCL circuit. For simplicity, in the following we assume $t_0 = 0$ with zero initial conditions:

$$v_n(0) = 0 \quad \text{and} \quad i_l(0) = 0. \tag{8.5}$$

Instead of solving (8.3) and (8.4) directly, one usually first eliminates as many variables as possible; this procedure is called modified nodal analysis [HRB75, VS94]. More precisely, using the last three equations in (8.3) and the first three equations in (8.4), one can eliminate v_g, v_c, v_l, i_i, i_g, i_c, and is left with the coupled equations

$$\begin{aligned} A_i^T I(t) &= A_g^T G A_g v_n + A_c^T C A_c \frac{d}{dt} v_n + A_l^T i_l, \\ A_l v_n &= L \frac{d}{dt} i_l \end{aligned} \tag{8.6}$$

for v_n and i_l. Note that the equations (8.6) are completed by the initial conditions (8.5).

For later use, we remark that the energy supplied to the RCL circuit by the current sources is given by

$$E(t) = \int_0^t \big(v_i(\tau)\big)^T I(\tau)\, d\tau. \tag{8.7}$$

Recall that the entries of the vector v_i are the voltages at the current sources. In view of the second equation in (8.3), v_i is connected to v_n by the output relation

$$v_i = A_i v_n. \tag{8.8}$$

8.2.2 RCL Circuits as First-Order Systems

The RCL circuit equations (8.6) and (8.8) can be viewed as a first-order time-invariant linear dynamical system with state vector

$$z(t) := \begin{bmatrix} v_n(t) \\ i_l(t) \end{bmatrix},$$

and input and output vectors

$$u(t) := I(t) \quad \text{and} \quad y(t) := v_i(t), \tag{8.9}$$

respectively. Indeed, the equations (8.6) and (8.8) are equivalent to

$$\begin{aligned} \mathcal{E}\, \frac{d}{dt} z(t) - \mathcal{A}\, z(t) &= \mathcal{B}\, u(t), \\ y(t) &= \mathcal{B}^T z(t), \end{aligned} \tag{8.10}$$

where

$$\mathcal{E} := \begin{bmatrix} A_c^T C A_c & 0 \\ 0 & L \end{bmatrix}, \quad \mathcal{A} := \begin{bmatrix} -A_g^T G A_g & -A_l^T \\ A_l & 0 \end{bmatrix}, \quad \mathcal{B} := \begin{bmatrix} A_i^T \\ 0 \end{bmatrix}. \tag{8.11}$$

Note that (8.10) is a system of *differential-algebraic equations* (DAEs) of first order. Furthermore, in view of (8.9), the energy (8.7), which is supplied to the RCL circuit by the current sources, is just the integral

$$E(t) = \int_0^t \big(y(\tau)\big)^T u(\tau)\, d\tau \tag{8.12}$$

of the inner product of the input and output vectors of (8.10). RCL circuits are passive systems, i.e., they do not generate energy, and (8.12) is an important formula for the proper treatment of passivity; see, e.g., [AV73, LBEM00].

8.2.3 RCL Circuits as Second-Order Systems

In this subsection, we assume that the inductance matrix L of the RCL circuit is nonsingular. In this case, the RCL circuit equations (8.6) and (8.8) can also be viewed as a second-order time-invariant linear dynamical system with state vector

$$x(t) := v_n(t),$$

and the same input and output vectors (8.9) as before. Indeed, by integrating the second equation of (8.6) and using (8.5), we obtain

$$L\, i_l(t) = A_l \int_0^t v_n(\tau)\, d\tau. \tag{8.13}$$

Since L is assumed to be nonsingular, we can employ the relation (8.13) to eliminate i_l in (8.6). The resulting equation, combined with (8.8), can be written as follows:

$$\begin{aligned} P_1 \frac{d}{dt} x(t) + P_0\, x(t) + P_{-1} \int_0^t x(\tau)\, d\tau &= B\, u(t), \\ y(t) &= B^T x(t). \end{aligned} \tag{8.14}$$

Here, we have set

$$P_1 := A_c^T C A_c, \quad P_0 := A_g^T G A_g, \quad P_{-1} := A_l^T L^{-1} A_l, \quad B := A_i^T. \tag{8.15}$$

Note that the first equation in (8.14) is a system of integro-DAEs. We will refer to (8.14) as a *special* second-order time-invariant linear dynamical system. We remark that the input and output vectors of (8.14) are the same as in the first-order formulation (8.10). In particular, the important formula (8.12) for the energy supplied to the system remains valid for the special second-order formulation (8.10).

If the input vector $u(t)$ is differentiable, then by differentiating the first equation of (8.14) we obtain the "true" second-order formulation

$$\begin{aligned} P_1 \frac{d^2}{dt^2} x(t) + P_0 \frac{d}{dt} x(t) + P_{-1}\, x(t) &= B \frac{d}{dt} u(t), \\ y(t) &= B^T x(t). \end{aligned} \tag{8.16}$$

However, besides the additional assumption on the differentiability of $u(t)$, the formulation (8.16) also has the disadvantage that the energy supplied to the system is no longer given by the integral of the inner product of the input and output vectors

$$\hat{u}(t) := \frac{d}{dt} u(t) \quad \text{and} \quad \hat{y}(t) := y(t)$$

of (8.16). Related to this lack of a formula of type (8.12) is the fact that the transfer function of (8.16) is no longer positive real. For these reasons, we prefer to use the special second-order formulation (8.14), rather than the more common formulation (8.16).

8.3 Higher-Order Linear Dynamical Systems

In this section, we discuss our general framework for special second-order and higher-oder linear dynamical systems. We denote by m and p the number of inputs and outputs, respectively, and by l the order of such systems. In the following, the only assumption on m, p, and l is that $m, p, l \geq 1$.

8.3.1 Special Second-Order Systems

A *special second-order m-input p-output time-invariant linear dynamical system of order l* is a system of integro-DAEs of the following form:

$$\begin{aligned} P_1 \frac{d}{dt} x(t) + P_0\, x(t) + P_{-1} \int_{t_0}^{t} x(\tau)\, d\tau &= B\, u(t), \\ y(t) &= D\, u(t) + L\, x(t), \\ x(t_0) &= x_0. \end{aligned} \tag{8.17}$$

Here, P_{-1}, P_0, $P_1 \in \mathbb{C}^{N\times N}$, $B \in \mathbb{C}^{N\times m}$, $D \in \mathbb{C}^{p\times m}$, and $L \in \mathbb{C}^{p\times N}$ are given matrices, $t_0 \in \mathbb{R}$ is a given initial time, and $x_0 \in \mathbb{C}^N$ is a given vector of initial values. We assume that the $N \times N$ matrix

$$sP_1 + P_0 + \frac{1}{s} P_{-1}$$

is singular only for finitely many values of $s \in \mathbb{C}$.

The frequency-domain transfer function of (8.17) is given by

$$H(s) = D + L\Big(sP_1 + P_0 + \frac{1}{s} P_{-1}\Big)^{-1} B. \tag{8.18}$$

Note that

$$H : \mathbb{C} \mapsto (\mathbb{C} \cup \infty)^{p\times m}$$

is a matrix-valued rational function.

In practical applications, such as the case of RCL circuits described in Section 8.2, the matrices P_0 and P_1 are usually sparse. The matrix P_{-1}, however, may be dense, but has a sparse representation of the form

$$P_{-1} = F_1 G F_2^H \tag{8.19}$$

or

$$P_{-1} = F_1 G^{-1} F_2^H, \quad \text{with nonsingular } G, \tag{8.20}$$

where F_1, $F_2 \in \mathbb{C}^{N\times N_0}$ and $G \in \mathbb{C}^{N_0\times N_0}$ are sparse matrices. We stress that in the case (8.19), the matrix G is not required to be nonsingular. In particular, for any matrix $P_{-1} \in \mathbb{C}^{N\times N}$, there is always the trivial factorization (8.19) with $F_1 = F_2 = I$ and $G = P_{-1}$. Therefore, without loss of generality, in the following, we assume that the matrix P_{-1} in (8.17) is given by a product of the form (8.19) or (8.20).

8.3.2 General Higher-Order Systems

An *m-input p-output time-invariant linear dynamical system of order l* is a system of DAEs of the following form:

$$\begin{aligned} P_l \frac{d^l}{dt^l}x(t) + P_{l-1}\frac{d^{l-1}}{dt^{l-1}}x(t) + \cdots + P_1\frac{d}{dt}x(t) + P_0\,x(t) &= B\,u(t), \\ y(t) = D\,u(t) + L_{l-1}\frac{d^{l-1}}{dt^{l-1}}x(t) + \cdots + L_1\frac{d}{dt}x(t) + L_0\,x(t). & \end{aligned} \tag{8.21}$$

Here, $P_i \in \mathbb{C}^{N\times N}$, $0 \le i \le l$, $B \in \mathbb{C}^{N\times m}$, $D \in \mathbb{C}^{p\times m}$, and $L_j \in \mathbb{C}^{p\times N}$, $0 \le j < l$, are given matrices, and N is called the state-space dimension of (8.21). Moreover, in (8.21), $u : [t_0, \infty) \mapsto \mathbb{C}^m$ is a given input function, $t_0 \in \mathbb{R}$ is a given initial time, the components of the vector-valued function $x : [t_0, \infty) \mapsto \mathbb{C}^N$ are the so-called state variables, and $y : [t_0, \infty) \mapsto \mathbb{C}^p$ is the output function. The system is completed by initial conditions of the form

$$\frac{d^i}{dt^i}x(t)\,\Big|_{t=t_0} = x_0^{(i)}, \quad 0 \le i < l, \tag{8.22}$$

where $x_0^{(i)} \in \mathbb{C}^n$, $0 \le i < l$, are given vectors.

The frequency-domain transfer function of (8.21) is given by

$$H(s) := D + L(s)\big(P(s)\big)^{-1}B, \quad s \in \mathbb{C}, \tag{8.23}$$

where

$$P(s) := s^l P_l + s^{l-1}P_{l-1} + \cdots + sP_1 + P_0 \tag{8.24}$$

and

$$L(s) := s^{l-1}L_{l-1} + s^{l-2}L_{l-2} + \cdots + sL_1 + L_0.$$

Note that

$$P : \mathbb{C} \mapsto \mathbb{C}^{N\times N} \quad \text{and} \quad L : \mathbb{C} \mapsto \mathbb{C}^{p\times N}$$

are matrix-valued polynomials, and that

$$H : \mathbb{C} \mapsto (\mathbb{C} \cup \infty)^{p\times m}$$

again is a matrix-valued rational function. We assume that the polynomial (8.24), P, is *regular*, that is, the matrix $P(s)$ is singular only for finitely many values of $s \in \mathbb{C}$; see, e.g., [GLR82, Part II]. This guarantees that the transfer function (8.23) has only finitely many poles.

8.3.3 First-Order Systems

For the special case $l = 1$, systems of the form (8.21) are called first-order systems. In the following, we use calligraphic letters for the data matrices and

z for the vector of state-space variables of first-order systems. More precisely, we always write first-order systems in the form

$$\begin{aligned} \mathcal{E}\,\frac{d}{dt}z(t) - \mathcal{A}\,z(t) &= \mathcal{B}\,u(t), \\ y(t) &= \mathcal{D}\,u(t) + \mathcal{L}\,z(t), \\ z(t_0) &= z_0. \end{aligned} \tag{8.25}$$

Note that the transfer function of (8.25) is given by

$$H(s) = \mathcal{D} + \mathcal{L}\left(s\,\mathcal{E} - \mathcal{A}\right)^{-1}\mathcal{B}. \tag{8.26}$$

8.4 Equivalent First-Order Systems

A standard approach to treat higher-order systems is to rewrite them as equivalent first-order systems. In this section, we present such equivalent first-order formulations of special second-order and general higher-order systems.

8.4.1 The Case of Special Second-Order Systems

We start with special second-order systems (8.17), and we distinguish the two cases (8.19) and (8.20).

First assume that P_{-1} is given by (8.19). In this case, we set

$$z_1(t) := x(t) \quad \text{and} \quad z_2(t) := F_2^H \int_{t_0}^{t} x(\tau)\,d\tau. \tag{8.27}$$

By (8.19) and (8.27), the first relation in (8.17) can be rewritten as follows:

$$P_1\,\frac{d}{dt}z_1(t) + P_0\,z_1(t) + F_1 G\,z_2(t) = B\,u(t). \tag{8.28}$$

Moreover, (8.27) implies that

$$\frac{d}{dt}z_2(t) = F_2^H z_1(t). \tag{8.29}$$

It follows from (8.27)–(8.29) that the special second-order system (8.17) (with P_{-1} given by (8.19)) is equivalent to a first-order system (8.25) where

$$\begin{gathered} z(t) := \begin{bmatrix} z_1(t) \\ z_2(t) \end{bmatrix}, \quad z_0 := \begin{bmatrix} x_0 \\ 0 \end{bmatrix}, \quad \mathcal{L} := \begin{bmatrix} L & 0 \end{bmatrix}, \quad \mathcal{B} := \begin{bmatrix} B \\ 0 \end{bmatrix}, \\ \mathcal{D} := D, \quad \mathcal{A} := \begin{bmatrix} -P_0 & -F_1 G \\ F_2^H & 0 \end{bmatrix}, \quad \mathcal{E} := \begin{bmatrix} P_1 & 0 \\ 0 & I_{N_0} \end{bmatrix}. \end{gathered} \tag{8.30}$$

The state-space dimension of this first-order system is $N_1 := N + N_0$, where N and N_0 denote the dimensions of $P_1 \in \mathbb{C}^{N \times N}$ and $G \in \mathbb{C}^{N_0 \times N_0}$. Note that (8.26) is the corresponding representation of the transfer function (8.18), H, in terms of the data matrices defined in (8.30).

Next, we assume that P_{-1} is given by (8.20). We set

$$z_1(t) := x(t) \quad \text{and} \quad z_2(t) := G^{-1} F_2^H \int_{t_0}^{t} x(\tau)\, d\tau.$$

The first relation in (8.17) can then be rewritten as

$$P_1 \frac{d}{dt} z_1(t) + P_0\, z_1(t) + F_1\, z_2(t) = B\, u(t).$$

Moreover, we have

$$G \frac{d}{dt} z_2(t) = F_2^H z_1(t).$$

It follows that the special second-order system (8.17) (with P_{-1} given by (8.20)) is equivalent to a first-order system (8.25) where

$$\begin{aligned} z(t) &:= \begin{bmatrix} z_1(t) \\ z_2(t) \end{bmatrix}, \quad z_0 := \begin{bmatrix} x_0 \\ 0 \end{bmatrix}, \quad \mathcal{L} := \begin{bmatrix} L & 0 \end{bmatrix}, \quad \mathcal{B} := \begin{bmatrix} B \\ 0 \end{bmatrix}, \\ \mathcal{D} &:= D, \quad \mathcal{A} := \begin{bmatrix} -P_0 & -F_1 \\ F_2^H & 0 \end{bmatrix}, \quad \mathcal{E} := \begin{bmatrix} P_1 & 0 \\ 0 & G \end{bmatrix}. \end{aligned} \tag{8.31}$$

The state-space dimension of this first-order system is again $N_1 := N + N_0$. Note that (8.26) is the corresponding representation of the transfer function (8.18), H, in terms of the data matrices defined in (8.31).

8.4.2 The Case of General Higher-Order Systems

It is well known (see, e.g., [GLR82, Chapter 7]) that any l-th order system with state-space dimension N is equivalent to a first-order system with state-space dimension $N_1 := lN$. Indeed, it is easy to verify that the l-th order system (8.21) with initial conditions (8.22) is equivalent to the first-order system (8.25) with

$$
\begin{aligned}
z(t) &:= \begin{bmatrix} x(t) \\ \frac{d}{dt}x(t) \\ \vdots \\ \frac{d^{l-1}}{dt^{l-1}}x(t) \end{bmatrix}, \quad z_0 := \begin{bmatrix} x_0^{(0)} \\ x_0^{(1)} \\ \vdots \\ x_0^{(l-1)} \end{bmatrix}, \quad \mathcal{B} := \begin{bmatrix} 0 \\ \vdots \\ 0 \\ B \end{bmatrix}, \\
\mathcal{L} &:= \begin{bmatrix} L_0 & L_1 & \cdots & L_{l-1} \end{bmatrix}, \quad \mathcal{D} := D, \\
\mathcal{E} &:= \begin{bmatrix} I & 0 & 0 & \cdots & 0 \\ 0 & I & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & I & 0 \\ 0 & \cdots & 0 & 0 & P_l \end{bmatrix}, \quad \mathcal{A} := - \begin{bmatrix} 0 & -I & 0 & \cdots & 0 \\ 0 & 0 & -I & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 0 & -I \\ P_0 & P_1 & P_2 & \cdots & P_{l-1} \end{bmatrix}.
\end{aligned}
\tag{8.32}
$$

We remark that (8.26) is the corresponding representation of the l-order transfer function (8.23), H, in terms of the data matrices defined in (8.32).

8.5 Dimension Reduction of Equivalent First-Order Systems

In this section, we discuss some general concepts of dimension reduction of special second-order and general higher-order systems via dimension reduction of equivalent first-order systems.

8.5.1 General Reduced-Order Models

We start with general first-order systems (8.25). For simplicity, from now on we always assume zero initial conditions, i.e., $z_0 = 0$ in (8.25). We can then drop the initial conditions in (8.25), and consider first-order systems (8.25) of the following form:

$$
\begin{aligned}
\mathcal{E}\,\frac{d}{dt}z(t) - \mathcal{A}\,z(t) &= \mathcal{B}\,u(t), \\
y(t) &= \mathcal{D}\,u(t) + \mathcal{L}\,z(t).
\end{aligned}
\tag{8.33}
$$

Here, $\mathcal{A}$, $\mathcal{E} \in \mathbb{C}^{N_1 \times N_1}$, $\mathcal{B}_1 \in \mathbb{C}^{N_1 \times m}$, $\mathcal{D} \in \mathbb{C}^{p \times m}$, and $\mathcal{L} \in \mathbb{C}^{p \times N_1}$ are given matrices. Recall that N_1 is the state-space dimension of (8.33). We assume that the matrix pencil $s\,\mathcal{E} - \mathcal{A}$ is *regular*, i.e., the matrix $s\,\mathcal{E} - \mathcal{A}$ is singular only for finitely many values of $s \in \mathbb{C}$. This guarantees that the transfer function of (8.33),

$$
H(s) := \mathcal{D} + \mathcal{L}\left(s\,\mathcal{E} - \mathcal{A}\right)^{-1}\mathcal{B}, \tag{8.34}
$$

exists.

A *reduced-order model* of (8.33) is a system of the same form as (8.33), but with smaller state-space dimension. More precisely, a reduced-order model of (8.33) with state-space dimension n_1 ($< N_1$) is a system of the form

$$\begin{aligned} \tilde{\mathcal{E}}\,\frac{d}{dt}\tilde{z}(t) - \tilde{\mathcal{A}}\,z(t) &= \tilde{\mathcal{B}}\,u(t), \\ \tilde{y}(t) &= \tilde{\mathcal{D}}\,u(t) + \tilde{\mathcal{L}}\,\tilde{z}(t), \end{aligned} \tag{8.35}$$

where $\tilde{\mathcal{A}}$, $\tilde{\mathcal{E}} \in \mathbb{C}^{n_1 \times n_1}$, $\tilde{\mathcal{B}} \in \mathbb{C}^{n_1 \times m}$, $\tilde{\mathcal{D}} \in \mathbb{C}^{p \times m}$, and $\tilde{\mathcal{L}} \in \mathbb{C}^{p \times n_1}$. Again, we assume that the matrix pencil $s\,\tilde{\mathcal{E}} - \tilde{\mathcal{A}}$ is regular. The transfer function of (8.35) is then given by

$$\tilde{H}(s) := \tilde{\mathcal{D}} + \tilde{\mathcal{L}}\,\bigl(s\,\tilde{\mathcal{E}} - \tilde{\mathcal{A}}\bigr)^{-1}\tilde{\mathcal{B}}. \tag{8.36}$$

Of course, (8.35) only provides a framework for model reduction. The real problem, namely the choice of suitable matrices $\tilde{\mathcal{A}}$, $\tilde{\mathcal{E}}$, $\tilde{\mathcal{B}}$, $\tilde{\mathcal{L}}$, $\tilde{\mathcal{D}}$, and sufficiently large reduced state-space dimension n_1 still remains to be addressed.

8.5.2 Reduction via Projection

A simple, yet very powerful (when combined with Krylov-subspace machinery) approach for constructing reduced-order models is projection. Let

$$\mathcal{V} \in \mathbb{C}^{N_1 \times n_1} \tag{8.37}$$

be a given matrix, and set

$$\tilde{\mathcal{A}} := \mathcal{V}^H \mathcal{A}\,\mathcal{V}, \quad \tilde{\mathcal{E}} := \mathcal{V}^H \mathcal{E}\,\mathcal{V}, \quad \tilde{\mathcal{B}} := \mathcal{V}^H \mathcal{B} \quad \tilde{\mathcal{L}} := \mathcal{L}\,\mathcal{V}, \quad \tilde{\mathcal{D}} := \mathcal{D}. \tag{8.38}$$

Then, provided that the matrix pencil $s\,\tilde{\mathcal{E}} - \tilde{\mathcal{A}}$ is regular, the system (8.35) with matrices given by (8.38) is a reduced-order model of (8.33) with state-space dimension n_1.

A more general approach employs two matrices,

$$\mathcal{V},\, \mathcal{W} \in \mathbb{C}^{N_1 \times n_1},$$

and two-sided projections of the form

$$\tilde{\mathcal{A}} := \mathcal{W}^H \mathcal{A}\,\mathcal{V}, \quad \tilde{\mathcal{E}} := \mathcal{W}^H \mathcal{E}\,\mathcal{V}, \quad \tilde{\mathcal{B}} := \mathcal{V}^H \mathcal{B} \quad \tilde{\mathcal{L}} := \mathcal{L}\,\mathcal{W}, \quad \tilde{\mathcal{D}} := \mathcal{D}.$$

For example, the PVL algorithm [FF94, FF95] can be viewed as a two-sided projection method, where the columns of the matrices $\mathcal{V}$ and $\mathcal{W}$ are the first n_1 right and left Lanczos vectors generated by the nonsymmetric Lanczos process [Lan50].

All model-reduction techniques discussed in the remainder of this paper are based on projections of the type (8.38).

Next, we discuss the application of projections (8.38) to first-order systems (8.33) that arise as equivalent formulations of special second-order and

higher-oder linear dynamical systems. Recall from Section 8.4 that such equivalent first-order systems exhibit certain structures. For general matrices (8.37), $\mathcal{V}$, the projected matrices (8.38) do not preserve these structures. However, as we will show now, these structures are preserved for certain types of matrices $\mathcal{V}$.

8.5.3 Preserving Special Second-Order Structure

In this subsection, we consider special second-order systems (8.17), where P_{-1} is either of the form (8.19) or (8.20). Recall that the data matrices of the equivalent first-order formulations of (8.17) are defined in (8.30), respectively (8.31).

Let $\mathcal{V}$ be any matrix of the block form

$$\mathcal{V} = \begin{bmatrix} V_1 & 0 \\ 0 & V_2 \end{bmatrix}, \quad V_1 \in \mathbb{C}^{N \times n}, \quad V_2 \in \mathbb{C}^{N_0 \times n_0}, \tag{8.39}$$

such that the matrix

$$\tilde{G} := V_2^H G V_2 \quad \text{is nonsingular.}$$

First, consider the case of matrices P_{-1} of the form (8.19). Using (8.30) and (8.39), one readily verifies that in this case, the projected matrices (8.38) are as follows:

$$\begin{aligned} &\tilde{\mathcal{A}} = \begin{bmatrix} -\tilde{P}_0 & -\tilde{F}_1 \tilde{G} \\ \tilde{F}_2^H & 0 \end{bmatrix}, \quad \tilde{\mathcal{E}} = \begin{bmatrix} \tilde{P}_1 & 0 \\ 0 & I_{n_0} \end{bmatrix}, \quad \tilde{\mathcal{B}} = \begin{bmatrix} \tilde{B} \\ 0 \end{bmatrix}, \\ &\tilde{\mathcal{L}} = \begin{bmatrix} \tilde{L} \; 0 \end{bmatrix}, \quad \tilde{\mathcal{D}} = D. \end{aligned} \tag{8.40}$$

Here, we have set

$$\tilde{P}_0 := V_1^H P_0 V_1, \quad \tilde{P}_1 := V_1^H P_1 V_1, \quad \tilde{B} := V_1^H B, \quad \tilde{L} := L V_1, \tag{8.41}$$

and

$$\tilde{F}_1 := \left(V_1^H F_1 G V_2\right) \tilde{G}^{-1}, \quad \tilde{F}_2 := V_1^H F_2 V_2.$$

Note that the matrices (8.40) are of the same form as the matrices (8.30) of the first-order formulation (8.33) of the original special second-order system (8.17) (with P_{-1} of the form (8.19). It follows that the matrices (8.40) define a reduced-order model in special second-order form,

$$\begin{aligned} \tilde{P}_1 \frac{d}{dt} \tilde{x}(t) + \tilde{P}_0 \, \tilde{x}(t) + \tilde{P}_{-1} \int_{t_0}^{t} \tilde{x}(\tau) \, d\tau &= \tilde{B} \, u(t), \\ \tilde{y}(t) &= \tilde{D} \, u(t) + \tilde{L} \, \tilde{x}(t), \end{aligned} \tag{8.42}$$

where

$$\tilde{P}_{-1} := \tilde{F}_1 \tilde{G} \tilde{F}_2^H.$$

We remark that the state-space dimension of (8.42) is n, where n denotes the number of columns of the submatrix V_1 in (8.39).

Next, consider the case of matrices P_{-1} of the form (8.20). Using (8.31) and (8.39), one readily verifies that in this case, the projected matrices (8.38) are as follows:

$$\begin{aligned} \tilde{\mathcal{A}} &= \begin{bmatrix} -\tilde{P}_0 & -\tilde{F}_1 \\ \tilde{F}_2^H & 0 \end{bmatrix}, \quad \tilde{\mathcal{E}} = \begin{bmatrix} \tilde{P}_1 & 0 \\ 0 & \tilde{G} \end{bmatrix}, \quad \tilde{\mathcal{B}} = \begin{bmatrix} \tilde{B} \\ 0 \end{bmatrix}, \\ \tilde{\mathcal{L}} &= \begin{bmatrix} \tilde{L} & 0 \end{bmatrix}, \quad \tilde{\mathcal{D}} = D. \end{aligned} \tag{8.43}$$

Here, $\tilde{P}_0$, $\tilde{P}_1$, $\tilde{B}$, $\tilde{L}$ are the matrices defined in (8.41), and

$$\tilde{F}_1 := V_1^H F_1 V_2, \quad \tilde{F}_2 := V_1^H F_2 V_2.$$

Again, the matrices (8.43) are of the same form as the matrices (8.31) of the first-order formulation (8.33) of the original special second-order system (8.17) (with P_{-1} of the form (8.20). It follows that the matrices (8.43) define a reduced-order model in special second-order form (8.42), where

$$\tilde{P}_{-1} = \tilde{F}_1 \tilde{G}^{-1} \tilde{F}_2^H.$$

8.5.4 Preserving General Higher-Order Structure

We now turn to systems (8.33) that are equivalent first-order formulations of general l-th order linear dynamical systems (8.21). More precisely, we assume that the matrices in (8.33) are the ones defined in (8.32).

Let $\mathcal{V}$ be any $lN \times ln$ matrix of the block form

$$\mathcal{V}_n = \begin{bmatrix} S_n & 0 & 0 & \cdots & 0 \\ 0 & S_n & 0 & \cdots & 0 \\ 0 & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & 0 & S_n \end{bmatrix}, \quad S_n \in \mathbb{C}^{N \times n}, \quad S_n^H S_n = I_n. \tag{8.44}$$

Although such matrices appear to be very special, they do arise in connection with block-Krylov subspaces and lead to Padé-type reduced-order models; see Subsection 8.6.4 below. The block structure (8.44) implies that the projected matrices (8.38) are given by

$$\tilde{\mathcal{A}} = -\begin{bmatrix} 0 & -I & 0 & \cdots & 0 \\ 0 & 0 & -I & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 0 & -I \\ \tilde{P}_0 & \tilde{P}_1 & \tilde{P}_2 & \cdots & \tilde{P}_{l-1} \end{bmatrix}, \quad \tilde{\mathcal{E}} := \begin{bmatrix} I & 0 & 0 & \cdots & 0 \\ 0 & I & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & I & 0 \\ 0 & \cdots & 0 & 0 & \tilde{P}_l \end{bmatrix},$$
$$\tilde{\mathcal{B}} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \tilde{B} \end{bmatrix}, \quad \tilde{\mathcal{L}} = \begin{bmatrix} \tilde{L}_0 & \tilde{L}_1 & \cdots & \tilde{L}_{l-1} \end{bmatrix}, \quad \tilde{\mathcal{D}} = \mathcal{D}, \tag{8.45}$$

where

$$\tilde{P}_i := S_n^H P_i S_n, \quad 0 \le i \le l, \quad \tilde{B} := S_n^H B, \quad \tilde{L}_j := L_j S_n, \quad 0 \le j < l.$$

It follows that the matrices (8.45) define a reduced-order model in l-th order form,

$$\begin{aligned} &\tilde{P}_l \frac{d^l}{dt^l}\tilde{x}(t) + \tilde{P}_{l-1} \frac{d^{l-1}}{dt^{l-1}}\tilde{x}(t) + \cdots + \tilde{P}_1 \frac{d}{dt}\tilde{x}(t) + \tilde{P}_0\, \tilde{x}(t) = \tilde{B}\, u(t), \\ &\tilde{y}(t) = \tilde{D}\, u(t) + \tilde{L}_{l-1} \frac{d^{l-1}}{dt^{l-1}}\tilde{x}(t) + \cdots + \tilde{L}_1 \frac{d}{dt}\tilde{x}(t) + \tilde{L}_0\, \tilde{x}(t), \end{aligned} \tag{8.46}$$

of the original l-th order system (8.21). We remark that the state-space dimension of (8.46) is n, where n denotes the number of columns of the matrix S_n in (8.44).

8.6 Block-Krylov Subspaces and Padé-type Models

In this section, we review the concepts of block-Krylov subspaces and Padé-type reduced-order models.

8.6.1 Padé-Type Models

Let $s_0 \in \mathbb{C}$ be any point such that the matrix $s_0\,\mathcal{E} - \mathcal{A}$ is nonsingular. Recall that the matrix pencil $s\,\mathcal{E} - \mathcal{A}$ is assumed to be regular, and so the matrix $s_0\,\mathcal{E} - \mathcal{A}$ is nonsingular except for finitely many values of $s_0 \in \mathbb{C}$. In practice, $s_0 \in \mathbb{C}$ is chosen such that $s_0\,\mathcal{E} - \mathcal{A}$ is nonsingular and at the same time, s_0 is in some sense "close" to a problem-specific relevant frequency range in the complex Laplace domain. Furthermore, for systems with real matrices $\mathcal{A}$ and $\mathcal{E}$ one usually selects $s_0 \in \mathbb{R}$ in order to avoid complex arithmetic.

We consider first-order systems of the form (8.33) and their reduced-order models of the form (8.35). By expanding the transfer function (8.34), H, of the original system (8.33) about s_0, we obtain

$$H(s) = \mathcal{L}\left(s\,\mathcal{E} - \mathcal{A}\right)^{-1}\mathcal{B} = \mathcal{L}\left(I + (s-s_0)\mathcal{M}\right)^{-1}\mathcal{R} = \sum_{i=0}^{\infty}(-1)^i\,\mathcal{L}\,\mathcal{M}^i\,\mathcal{R}\,(s-s_0)^i, \tag{8.47}$$

where

$$\mathcal{M} := \left(s_0\,\mathcal{E} - \mathcal{A}\right)^{-1}\mathcal{E} \quad \text{and} \quad \mathcal{R} := \left(s_0\,\mathcal{E} - \mathcal{A}\right)^{-1}\mathcal{B}. \tag{8.48}$$

Provided that the matrix $s_0\,\tilde{\mathcal{E}} - \tilde{\mathcal{A}}$ is nonsingular, we can also expand the transfer function (8.36), $\tilde{H}$, of the reduced-order model (8.35) about s_0. This gives

$$\tilde{H}(s) = \tilde{\mathcal{L}}\left(s\,\tilde{\mathcal{E}} - \tilde{\mathcal{A}}\right)^{-1}\mathcal{B} = \sum_{i=0}^{\infty}(-1)^i\,\tilde{\mathcal{L}}\,\tilde{\mathcal{M}}^i\,\tilde{\mathcal{R}}\,(s-s_0)^i, \tag{8.49}$$

where

$$\tilde{\mathcal{M}} := \left(s_0\,\tilde{\mathcal{E}} - \tilde{\mathcal{A}}\right)^{-1}\tilde{\mathcal{E}} \quad \text{and} \quad \mathcal{R} := \left(s_0\,\tilde{\mathcal{E}} - \tilde{\mathcal{A}}\right)^{-1}\tilde{\mathcal{B}}.$$

We call the reduced-order model (8.35) a *Padé-type model* (with expansion point s_0) of the original system (8.33) if the Taylor expansions (8.47) and (8.49) agree in a number of leading terms, i.e.,

$$\tilde{H}(s) = H(s) + \mathcal{O}\big((s-s_0)^q\big) \tag{8.50}$$

for some $q = q(\tilde{\mathcal{A}}, \tilde{\mathcal{E}}, \tilde{\mathcal{B}}, \tilde{\mathcal{L}}, \tilde{\mathcal{D}}) > 0$.

Recall that the state-space dimension of the reduced-order model (8.35) is n_1. If for a given n_1, the matrices $\tilde{\mathcal{A}}$, $\tilde{\mathcal{E}}$, $\tilde{\mathcal{B}}$, $\tilde{\mathcal{L}}$, $\tilde{\mathcal{D}}$ in (8.35) are chosen such that $q = q(n_1)$ in (8.50) is optimal, i.e., as large as possible, then the reduced-order model (8.35) is called a *Padé model*. All the reduced-order models discussed in the remainder of this paper are Padé-type models, but they are not optimal in the Padé sense.

The (matrix-valued) coefficients in the expansions (8.47) and (8.49) are often referred to as *moments*. Strictly speaking, the term "moments" should only be used in the case $s_0 = 0$; in this case, the Taylor coefficients of Laplace-domain transfer functions directly correspond to the moments in time domain. However, the use of the term "moments" has become common even in the case of general s_0. Correspondingly, the property (8.50) is now generally referred to as "moment matching".

We remark that the moment-matching property (8.50) is important for the following two reasons. First, for large-scale systems, the matrices $\mathcal{A}$ and $\mathcal{E}$ are usually sparse, and the dominant computational work for moment-matching reduction techniques is the computation of a sparse LU factorization of the matrix $s_0\,\mathcal{E} - \mathcal{A}$. Note that such a factorization is required already even if one only wants to evaluate the transfer function H at the point s_0. Once a sparse LU factorization of $s_0\,\mathcal{E} - \mathcal{A}$ has been generated, moments can be computed cheaply. Indeed, in view of (8.47) and (8.48), only sparse back solves, sparse

matrix products (with $\mathcal{E}$), and vector operations are required. Second, the moment-matching property (8.50) is inherently connected to block-Krylov subspaces. In particular, Padé-type reduced-order models can be computed easily be combining Krylov-subspace machinery and projection techniques. In the remainder of the section, we describe this connection with block-Krylov subspaces.

8.6.2 Block-Krylov Subspaces

In this subsection, we review the concept of block-Krylov subspaces induced by the matrices $\mathcal{M}$ and $\mathcal{R}$ defined in (8.48). Recall that $\mathcal{A}$, $\mathcal{E} \in \mathbb{C}^{N_1 \times N_1}$ and $\mathcal{B} \in \mathbb{C}^{N_1 \times m}$. Thus we have

$$\mathcal{M} \in \mathbb{C}^{N_1 \times N_1} \quad \text{and} \quad \mathcal{R} \in \mathbb{C}^{N_1 \times m}. \tag{8.51}$$

Next, consider the infinite *block-Krylov matrix*

$$\left[\mathcal{R}\ \mathcal{M}\mathcal{R}\ \mathcal{M}^2\mathcal{R} \cdots\ \mathcal{M}^j\mathcal{R} \ldots\right]. \tag{8.52}$$

In view of (8.51), the columns of the matrix (8.52) are vectors in $\mathbb{C}^{N_1}$, and so only at most N_1 of these vectors are linearly independent. By scanning the columns of the matrix (8.52) from left to right and deleting each column that is linearly dependent on columns to its left, one obtains the so-called *deflated* finite block-Krylov matrix

$$\left[\mathcal{R}^{(1)}\ \mathcal{M}\mathcal{R}^{(2)}\ \mathcal{M}^2\mathcal{R}^{(3)} \cdots\ \mathcal{M}^{j_{\max}-1}\mathcal{R}^{j_{\max}}\right], \tag{8.53}$$

where each block $\mathcal{R}^{(j)}$ is a subblock of $\mathcal{R}^{(j-1)}$, $j = 1, 2, \ldots, j_{\max}$, and $\mathcal{R}^{(0)} := \mathcal{R}$. Let m_j denote the number of columns of the j-th block $\mathcal{R}^{(j)}$. Note that by construction, the matrix (8.53) has full column rank. The *n-th block-Krylov subspace* (induced by $\mathcal{M}$ and $\mathcal{R}$) $\mathcal{K}_n\big(\mathcal{M}, \mathcal{R}\big)$ is defined as the subspace of $\mathbb{C}^{N_1}$ spanned by the first n columns of the matrix (8.53); see, [ABFH00] for more details of this construction. We stress that our notion of block-Krylov subspaces is more general than the standard definition, which ignores the need for deflation; again, we refer the reader to [ABFH00] and the references given there.

Here, we will only use those block-Krylov subspaces that correspond to the end of the blocks in (8.53). More precisely, let n be of the form

$$n = n(j) := m_1 + m_2 + \cdots + m_j, \quad \text{where} \quad 1 \le j \le j_{\max}. \tag{8.54}$$

In view of the above construction, the n-th block-Krylov subspace is given by

$$\mathcal{K}_n\big(\mathcal{M}, \mathcal{R}\big) = \operatorname{range}\left[\mathcal{R}^{(1)}\ \mathcal{M}\mathcal{R}^{(2)}\ \mathcal{M}^2\mathcal{R}^{(3)} \cdots\ \mathcal{M}^{j-1}\mathcal{R}^{(j)}\right]. \tag{8.55}$$

8.6.3 The Projection Theorem Revisited

It is well known that the projection approach described in Subsection 8.5.2 generates Padé-type reduced-order models, provided that the matrix $\mathcal{V}$ in (8.37) is chosen as a basis matrix for the block-Krylov subspaces induced by the matrices $\mathcal{M}$ and $\mathcal{R}$ defined in (8.48). This result is called the projection theorem, and it goes back to at least [dVS87]. It was also established in [Oda96, OCP97, OCP98] in connection with the PRIMA reduction approach; see [Fre00] for more details. A more general result, which includes the case of multi-point Padé-type approximations, can be found in [Gri97].

One key insight to obtain structure-preserving Padé-type reduced-order models via block-Krylov subspaces and projection is the fact that the projection theorem remains valid when the above assumption on $\mathcal{V}$ is replaced by the weaker condition

$$\mathcal{K}_n\big(\mathcal{M},\mathcal{R}\big) \subseteq \operatorname{range}\mathcal{V}_n. \tag{8.56}$$

In this subsection, we present an extension of the projection theorem (as stated in [Fre00]) to the case (8.56).

From now on, we always assume that n is an integer of the form (8.54) and that

$$\mathcal{V}_n \in \mathbb{C}^{N_1\times n_1} \tag{8.57}$$

is a matrix satisfying (8.56). Note that (8.56) implies $n_1 \geq n$. We stress that we make no further assumptions about n_1. We consider projected models given by (8.38) with $\mathcal{V} = \mathcal{V}_n$. In order to indicate the dependence on the dimension n of the block-Krylov subspace in (8.56), we use the notation

$$\begin{aligned} &\mathcal{A}_n := \mathcal{V}_n^H \mathcal{A}\,\mathcal{V}_n, \quad \mathcal{E}_n := \mathcal{V}_n^H \mathcal{E}\,\mathcal{V}_n, \quad \mathcal{B}_n := \mathcal{V}_n^H \mathcal{B}, \\ &\mathcal{L}_n := \mathcal{L}\,\mathcal{V}_n, \quad \mathcal{D}_n := \mathcal{D} \end{aligned} \tag{8.58}$$

for the matrices defining the projected reduced-order model. In addition to (8.56), we also assume that the matrix pencil $s\,\mathcal{E}_n - \mathcal{A}_n$ is regular, and that at the expansion point s_0, the matrix $s_0\,\mathcal{E}_n - \mathcal{A}_n$ is nonsingular. Then the reduced-order transfer function

$$\begin{aligned} H_n(s) &:= \mathcal{L}_n\left(s\,\mathcal{E}_n - \mathcal{A}_n\right)^{-1}\mathcal{B}_n \\ &= \mathcal{L}_n\left(I + (s-s_0)\mathcal{M}_n\right)^{-1}\mathcal{R}_n \\ &= \sum_{i=0}^{\infty}(-1)^i \mathcal{L}_n\,\mathcal{M}_n^i\,\mathcal{R}_n\,(s-s_0)^i \end{aligned} \tag{8.59}$$

is a well-defined rational function. Here, we have set

$$\mathcal{M}_n := \big(s_0\,\mathcal{E}_n - \mathcal{A}_n\big)^{-1}\mathcal{E}_n \quad \text{and} \quad \mathcal{R}_n := \big(s_0\,\mathcal{E}_n - \mathcal{A}_n\big)^{-1}\mathcal{B}_n. \tag{8.60}$$

We remark that the regularity of the matrix pencil $s\,\mathcal{E}_n - \mathcal{A}_n$ implies that the matrix $\mathcal{V}_n$ must have full column rank.

After these preliminaries, the extension of the projection theorem can be stated as follows.

Theorem 8.6.1. *Let $n = n(j)$ be of the form* (8.54)*, and let $\mathcal{V}_n$ be a matrix satisfying* (8.56)*. Then the reduced-order model defined by* (8.58) *is a Padé-type model with*

$$H_n(s) = H(s) + \mathcal{O}\big((s - s_0)^j\big). \tag{8.61}$$

Proof. In view of (8.47) and (8.59), the claim (8.61) is equivalent to

$$\mathcal{M}^i\,\mathcal{R} = \mathcal{V}_n\,\mathcal{M}_n^i\,\mathcal{R}_n \quad \text{for all} \quad i = 0, 1, \ldots, j-1, \tag{8.62}$$

and thus we need to show (8.62).

By (8.55) and (8.56), for each $i = 0, 1, \ldots, j-1$, there exists a matrix ρ_i such that

$$\mathcal{M}^i\,\mathcal{R} = \mathcal{V}_n\,\rho_i. \tag{8.63}$$

Moreover, since $\mathcal{V}_n$ has full column rank, each matrix ρ_i is unique. In fact, we will show that the matrices ρ_i in (8.63) are given by

$$\rho_i = \mathcal{M}_n^i\,\mathcal{R}_n, \quad i = 0, 1, \ldots, j-1. \tag{8.64}$$

The claim (8.62) then follows by inserting (8.64) into (8.63).

We prove (8.64) by induction on i. Let $i = 0$. In view of (8.48) and (8.63), we have

$$\mathcal{V}_n\,\rho_0 = \mathcal{R} = \big(s_0\,\mathcal{E} - \mathcal{A}\big)^{-1}\mathcal{B}. \tag{8.65}$$

Multiplying (8.65) from the left by

$$\big(s_0\,\mathcal{E}_n - \mathcal{A}_n\big)^{-1}\mathcal{V}_n^H\big(s_0\,\mathcal{E} - \mathcal{A}\big) \tag{8.66}$$

and using the definition of $\mathcal{R}_n$ in (8.60), it follows that $\rho_0 = \mathcal{R}_n$. This is just the relation (8.64) for $i = 0$.

Now let $1 \le i \le j-1$, and assume that

$$\rho_{i-1} = \mathcal{M}_n^{i-1}\,\mathcal{R}_n. \tag{8.67}$$

Recall that ρ_{i-1} satisfies the equation (8.63) (with i replaced by $i-1$), and thus we have $\mathcal{M}^{i-1}\,\mathcal{R} = \mathcal{V}_n\,\rho_{i-1}$. Together with (8.63) and (8.67), it follows that

$$\mathcal{V}_n\,\rho_i = \mathcal{M}^i\,\mathcal{R} = \mathcal{M}\,\big(\mathcal{M}^{i-1}\,\mathcal{R}\big) = \mathcal{M}\,\big(\mathcal{V}_n\,\rho_{i-1}\big) = \mathcal{M}\,\mathcal{V}_n\big(\mathcal{M}_n^{i-1}\,\mathcal{R}_n\big). \tag{8.68}$$

Note that, in view of the definition of $\mathcal{M}$ in (8.48), we have

$$\mathcal{V}_n^H\big(s_0\,\mathcal{E} - \mathcal{A}\big)\mathcal{M}\,\mathcal{V}_n = \mathcal{V}_n^H\,\mathcal{E}\,\mathcal{V}_n = \mathcal{E}_n. \tag{8.69}$$

Multiplying (8.68) from the left by the matrix (8.66) and using (8.69) as well as the definition of $\mathcal{M}_n$ in (8.60), we obtain

$$\rho_i = \big(s_0\,\mathcal{E}_n - \mathcal{A}_n\big)^{-1}\mathcal{E}_n\,\big(\mathcal{M}_n^{i-1}\,\mathcal{R}_n\big) = \mathcal{M}_n^i\,\mathcal{R}_n.$$

Thus the proof is complete.

We remark that, for the single-input case $m = 1$, the result of Theorem 8.6.1 is a special case of [Gri97, Lemma 3.2]. However, in [Gri97], the extension ([Gri97, Corollary 3.1]) to the case $m \geq 1$, is stated only for the standard notion of block-Krylov subspaces without deflation, and not for our more general definition described in [ABFH00] and sketched in Subsection 8.6.2. Therefore, for the sake of completeness, the short proof of Theorem 8.6.1 was included in this paper.

8.6.4 Structure-Preserving Padé-Type Models

We now turn to structure-preserving Padé-type models. Recall that, in Subsections 8.5.3 and 8.5.4, we have shown how special second-order and general higher-order structure is preserved by choosing projection matrices of the form (8.39) and (8.44), respectively. Moreover, in Subsection 8.6.3 we pointed out that projected models are Padé-type models if (8.56) is satisfied. It follows that the reduced-order models given by the projected data matrices (8.58) are structure-preserving Padé-type models, provided that the matrix $\mathcal{V}_n$ in (8.57) is of the form (8.39), respectively (8.44), and at the same time fulfills the condition (8.56). Next we show how to construct such matrices $\mathcal{V}_n$.

Let

$$\hat{\mathcal{V}}_n \in \mathbb{C}^{N_1 \times n} \tag{8.70}$$

be any matrix whose columns span the n-th block-Krylov subspace $\mathcal{K}_n\big(\mathcal{M}, \mathcal{R}\big)$, i.e.,

$$\mathcal{K}_n\big(\mathcal{M}, \mathcal{R}\big) = \operatorname{range} \hat{\mathcal{V}}_n. \tag{8.71}$$

First, consider the case of special second-order systems (8.17), where P_{-1} is either of the form (8.19) or (8.20). In this case, we partition $\hat{\mathcal{V}}_n$ as follows:

$$\hat{\mathcal{V}}_n = \begin{bmatrix} V_1 \\ V_2 \end{bmatrix}, \quad V_1 \in \mathbb{C}^{N \times n}, \quad V_2 \in \mathbb{C}^{N_0 \times n}. \tag{8.72}$$

Using the blocks in (8.72), we set

$$\mathcal{V}_n := \begin{bmatrix} V_1 & 0 \\ 0 & V_2 \end{bmatrix}. \tag{8.73}$$

Clearly, the matrix (8.73) is of the form (8.39), and thus the projected models generated with $\mathcal{V}_n$ preserve the special second-order structure. Moreover, from (8.71)–(8.73), it follows that

$$\mathcal{K}_n\big(\mathcal{M}, \mathcal{R}\big) = \operatorname{range} \hat{\mathcal{V}}_n \subseteq \operatorname{range} \mathcal{V}_n,$$

and so condition (8.56) is satisfied. Thus, projected models are Padé-type models and preserve second-order structure.

Next, we turn to the case of general higher-order systems (8.21). In [Fre04b], we have shown that in this case, the block-Krylov subspaces induced

by the matrices $\mathcal{M}$ and $\mathcal{R}$, which are given by (8.32) and (8.48), exhibit a very special structure. More precisely, the n-dimensional subspace $\mathcal{K}_n\big(\mathcal{M},\mathcal{R}\big)$ of $\mathbb{C}^{lN}$ can be viewed as l copies of an n-dimensional subspace of $\mathbb{C}^N$. Let $S_n \in \mathbb{C}^{N\times n}$ be a matrix whose columns form an orthonormal basis of this n-dimensional subspace of $\mathbb{C}^N$, and set

$$\mathcal{V}_n := \begin{bmatrix} S_n & 0 & 0 & \cdots & 0 \\ 0 & S_n & 0 & \cdots & 0 \\ 0 & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & 0 & S_n \end{bmatrix}. \tag{8.74}$$

From the above structure of the n-dimensional subspace $\mathcal{K}_n\big(\mathcal{M},\mathcal{R}\big)$, it follows that $\mathcal{V}_n$ satisfies the condition (8.56). Furthermore, the matrix $\mathcal{V}_n$ is of the form (8.44). Thus, projected models generated with $\mathcal{V}_n$ are Padé-type models and preserve higher-order structure.

In the remainder of this paper, we assume that $\mathcal{V}_n$ are matrices given by (8.73) in the case of special second-order systems, respectively (8.74) in the case of higher-order systems, and we consider the corresponding structure-preserving reduced-order models with data matrices given by (8.58).

8.7 Higher Accuracy in the Hermitian Case

For the structure-preserving Padé-type models introduced in Subsection 8.6.4, the result of Theorem 8.6.1 can be improved further, provided the underlying special second-order or higher-order linear dynamical system is Hermitian, and the expansion point s_0 is real, i.e.,

$$s_0 \in \mathbb{R}. \tag{8.75}$$

More precisely, in the Hermitian case, the Padé-type models obtained via $\mathcal{V}_n$ match $2j(n)$ moments, instead of just $j(n)$ in the general case; see Theorem 8.7.2 below. We remark that for the special case of real symmetric second-order systems and expansion point $s_0 = 0$, this result can be traced back to [SC91].

In this section, we first give an exact definition of Hermitian special second-order systems and higher-order systems, and then we prove the stronger moment-matching property stated in Theorem 8.7.2.

8.7.1 Hermitian Special Second-Order Systems

We say that a special second-order system (8.17) is *Hermitian* if the matrices in (8.17) and (8.19), respectively (8.20), satisfy the following properties:

$$L = B^H, \quad P_0 = P_0^H, \quad P_1 = P_1^H, \quad F_1 = F_2, \quad G = G^H. \tag{8.76}$$

Recall that RCL circuits are described by special second-order systems of the form (8.14) with real matrices defined in (8.15). Clearly, these systems are Hermitian.

We distinguish the two cases (8.19) and (8.20). First assume that P_{-1} is of the form (8.19). Recall that the data matrices of the equivalent first-order formulation (8.33) are defined in (8.30) in this case. Using (8.75), (8.76), and (8.19), one readily verifies that the data matrices (8.30) satisfy the relations

$$\begin{aligned} \mathcal{J}\,(s_0\,\mathcal{E} - \mathcal{A}) = (s_0\,\mathcal{E} - \mathcal{A})^H\,\mathcal{J}, \quad \mathcal{J}\,\mathcal{E} = \mathcal{E}\,\mathcal{J}, \quad \mathcal{J} = \mathcal{J}^H, \\ \mathcal{L}^H = \mathcal{J}\,\mathcal{B}, \end{aligned} \tag{8.77}$$

where

$$\mathcal{J} := \begin{bmatrix} I_N & 0 \\ 0 & -G \end{bmatrix}.$$

Since the reduced-order model is structure-preserving, the data matrices (8.58) satisfy analogous relations. More precisely, we have

$$\begin{aligned} \mathcal{J}_n\,(s_0\,\mathcal{E}_n - \mathcal{A}_n) = (s_0\,\mathcal{E}_n - \mathcal{A}_n)^H\,\mathcal{J}_n, \quad \mathcal{J}_n\,\mathcal{E}_n = \mathcal{E}_n\,\mathcal{J}_n, \quad \mathcal{J}_n = \mathcal{J}_n^H, \\ \mathcal{L}_n^H = \mathcal{J}_n\,\mathcal{B}_n, \end{aligned} \tag{8.78}$$

where

$$\mathcal{J}_n := \begin{bmatrix} I_n & 0 \\ 0 & -G_n \end{bmatrix}.$$

Now assume that P_{-1} is of the form (8.20). Recall that the data matrices of the equivalent first-order formulation (8.33) are defined in (8.31) in this case. Using (8.75), (8.76), and (8.20), one readily verifies that the data matrices (8.31) again satisfy the relations (8.77), where now

$$\mathcal{J} := \begin{bmatrix} I_N & 0 \\ 0 & -I_{N_0} \end{bmatrix}.$$

Furthermore, since the reduced-order model is structure-preserving, the data matrices (8.58) satisfy the relations (8.78), where

$$\mathcal{J}_n := \begin{bmatrix} I_n & 0 \\ 0 & -I_n \end{bmatrix}.$$

8.7.2 Hermitian Higher-Order Systems

We say that a higher-order system (8.21) is *Hermitian* if the matrices in (8.21) satisfy the following properties:

$$P_i = P_i^H, \quad 0 \le i \le l, \quad L_0 = B^H, \quad L_j = 0, \quad 1 \le j \le l-1. \tag{8.79}$$

In this case, we define matrices

$$\hat{P}_j := \sum_{i=0}^{l-j} s_0^i P_{j+i}, \quad j - 0, 1, \ldots, l,$$

and set

$$\mathcal{J} := \begin{bmatrix} I & -s_0 I & 0 & \cdots & 0 \\ 0 & I & -s_0 I & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & I & -s_0 I \\ 0 & 0 & \cdots & 0 & I \end{bmatrix} \begin{bmatrix} \hat{P}_1 & \hat{P}_2 & \cdots & \hat{P}_{l-1} & I \\ \hat{P}_2 & \iddots & \iddots & \hat{P}_l & 0 \\ \vdots & \iddots & \iddots & 0 & \vdots \\ \hat{P}_{l-1} & \iddots & \iddots & \vdots & \vdots \\ \hat{P}_l & 0 & \cdots & 0 & 0 \end{bmatrix}. \tag{8.80}$$

Note that, in view of (8.79), we have

$$\hat{P}_j = \hat{P}_j^H, \quad j = 0, 1, \ldots, l. \tag{8.81}$$

Using (8.79)–(8.81), one can verify that the data matrices $\mathcal{A}$, $\mathcal{E}$, $\mathcal{B}$, $\mathcal{L}$ given in (8.32) satisfy the following relations:

$$\mathcal{J}\left(s_0\,\mathcal{E} - \mathcal{A}\right) = \left(s_0\,\mathcal{E} - \mathcal{A}\right)^H \mathcal{J}, \quad \mathcal{J}\,\mathcal{E} = \mathcal{E}^H \mathcal{J}, \quad \mathcal{L}^H = \mathcal{J}\,\mathcal{B}. \tag{8.82}$$

Since the reduced-order model is structure-preserving, the data matrices (8.58) satisfy the same relations. More precisely, we have

$$\begin{aligned} \mathcal{J}_n\left(s_0\,\mathcal{E}_n - \mathcal{A}_n\right) = \left(s_0\,\mathcal{E}_n - \mathcal{A}_n\right)^H \mathcal{J}_n, \quad \mathcal{J}_n\,\mathcal{E}_n = \mathcal{E}_n^H \mathcal{J}_n, \\ \mathcal{L}_n^H = \mathcal{J}_n\,\mathcal{B}_n, \end{aligned} \tag{8.83}$$

where $\mathcal{J}_n$ is defined in analogy to $\mathcal{J}$.

8.7.3 Key Relations

Our proof of the enhanced moment-matching property in the Hermitian case is based on some key relations that hold true for both special second-order and higher-order systems. In this subsection, we state these key relations.

Recall the definition of the matrix $\mathcal{M}$ in (8.48). The relations (8.77), respectively (8.82), readily imply the following identity:

$$\mathcal{M}^H \mathcal{J} = \mathcal{J}\,\mathcal{E}\left(s_0\,\mathcal{E} - \mathcal{A}\right)^{-1}. \tag{8.84}$$

It follows from (8.84) that

$$\left(\mathcal{M}^H\right)^i \mathcal{J} = \mathcal{J}\left(\mathcal{E}\left(s_0\,\mathcal{E} - \mathcal{A}\right)^{-1}\right)^i, \quad i = 0, 1, \ldots . \tag{8.85}$$

Similarly, the relations (8.78), respectively (8.83), imply

$$\mathcal{M}_n^H \mathcal{J}_n = \mathcal{J}_n \mathcal{E}_n \left(s_0 \mathcal{E}_n - \mathcal{A}_n\right)^{-1}.$$

It follows that

$$\left(\mathcal{M}_n^H\right)^i \mathcal{J} = \mathcal{J}_n \left(\mathcal{E}_n \left(s_0 \mathcal{E}_n - \mathcal{A}_n\right)^{-1}\right)^i, \quad i = 0, 1, \ldots . \tag{8.86}$$

Also recall from (8.77), respectively (8.82), that

$$\mathcal{L}^H = \mathcal{J} \mathcal{B}, \tag{8.87}$$

and from (8.78), respectively (8.83), that

$$\mathcal{L}_n^H = \mathcal{J}_n \mathcal{B}_n. \tag{8.88}$$

Finally, one readily verifies the following relation:

$$\mathcal{V}_n^H \mathcal{J} \mathcal{E} \mathcal{V}_n = \mathcal{J}_n \mathcal{E}_n. \tag{8.89}$$

8.7.4 Matching Twice as Many Moments

In this subsection, we present our enhanced version of Theorem 8.6.1 for the case of Hermitian special second-order or higher-order systems.

First, we establish the following proposition.

Proposition 8.7.1. *Let $n = n(j)$ be of the form* (8.54)*. Then, the data matrices* (8.58) *of the structure-preserving Padé-type model satisfy*

$$\mathcal{L} \mathcal{M}^i \mathcal{V}_n = \mathcal{L}_n M_n^i \quad \textit{for all} \quad i = 0, 1, \ldots, j. \tag{8.90}$$

Proof. Recall that $\mathcal{L}_n = \mathcal{L} \mathcal{V}_n$. Thus (8.90) holds true for $i = 0$.

Let $1 \leq i \leq j$. In view of (8.85), we have

$$\left(\mathcal{M}^H\right)^i \mathcal{J} = \mathcal{J} \left(\mathcal{E} \left(s_0 \mathcal{E} - \mathcal{A}\right)^{-1}\right)^i.$$

Together with (8.87), it follows that

$$\left(\mathcal{M}^H\right)^i \mathcal{L}^H = \left(\mathcal{M}^H\right)^i \mathcal{J} \mathcal{B} = \mathcal{J} \left(\mathcal{E} \left(s_0 \mathcal{E} - \mathcal{A}\right)^{-1}\right)^i \mathcal{B}.$$

Since $\left(s_0 \mathcal{E} - \mathcal{A}\right)^{-1} \mathcal{B} = \mathcal{R}$, it follows that

$$\left(\mathcal{M}^H\right)^i \mathcal{L}^H = \mathcal{J} \mathcal{E} \left(\left(s_0 \mathcal{E} - \mathcal{A}\right)^{-1} \mathcal{E}\right)^{i-1} \mathcal{R} = \mathcal{J} \mathcal{E} \mathcal{M}^{i-1} \mathcal{R}.$$

Using (8.62) (with i replaced by $i - 1$), (8.89), (8.86), and (8.88), we obtain

$$\begin{aligned}
\mathcal{V}_n^H \left(\mathcal{M}^H\right)^i \mathcal{L}^H &= \mathcal{V}_n^H \, \mathcal{J} \, \mathcal{E} \left(\mathcal{M}^{i-1} \mathcal{R}\right) \\
&= \mathcal{V}_n^H \, \mathcal{J} \, \mathcal{E} \, \mathcal{V}_n \, \mathcal{M}_n^{i-1} \mathcal{R}_n \\
&= \left(\mathcal{V}_n^H \, \mathcal{J} \, \mathcal{E} \, \mathcal{V}_n\right) \left(\mathcal{M}_n^{i-1} \mathcal{R}_n\right) \\
&= \mathcal{J}_n \, \mathcal{E}_n \, \mathcal{M}_n^{i-1} \mathcal{R}_n \\
&= \mathcal{J}_n \, \mathcal{E}_n \, \mathcal{M}_n^{i-1} \left(s_0 \, \mathcal{E} - \mathcal{A}\right)^{-1} \mathcal{B}_n \\
&= \mathcal{J}_n \left(\mathcal{E}_n \left(s_0 \, \mathcal{E} - \mathcal{A}\right)^{-1}\right)^i \mathcal{B}_n \\
&= \left(\mathcal{M}_n^H\right)^i \mathcal{J}_n \, \mathcal{B}_n = \left(\mathcal{M}_n^H\right)^i \mathcal{L}_n^H.
\end{aligned}$$

Thus the proof is complete.

The following theorem contains the main result of this section.

Theorem 8.7.2. *Let $n = n(j)$ be of the form* (8.54)*. In the Hermitian case, the structure-preserving Padé-type model defined by the data matrices* (8.58) *satisfies:*

$$H_n(s) = H(s) + \mathcal{O}\big((s - s_0)^{2j(n)}\big). \tag{8.91}$$

Proof. Let $j = j(n)$. We need to show that

$$\mathcal{L} \, \mathcal{M}^i \, \mathcal{R} = \; c_n \, \mathcal{M}_n^i \, \mathcal{R}_n \quad \text{for all} \quad i = 0, 1, \ldots, 2j - 1. \tag{8.92}$$

By (8.62) and (8.90), we have

$$\begin{aligned}
\mathcal{L} \, \mathcal{M}^{i_1 + i_2} \, \mathcal{R} &= \left(\mathcal{L} \, \mathcal{M}^{i_1}\right) \left(\mathcal{M}^{i_2} \, \mathcal{R}\right) \\
&= \left(\mathcal{L} \, \mathcal{M}^{i_1}\right) \left(\mathcal{V}_n \, \mathcal{M}_n^{i_2} \, \mathcal{R}_n\right) \\
&= \left(\mathcal{L} \, \mathcal{M}^{i_1} \, \mathcal{V}_n\right) \left(\mathcal{M}_n^{i_2} \, \mathcal{R}_n\right) \\
&= \left(\mathcal{L}_n \, \mathcal{M}_n^{i_1}\right) \left(\mathcal{M}_n^{i_2} \, \mathcal{R}_n\right) = \mathcal{L}_n \, \mathcal{M}_n^{i_1 + i_2} \, \mathcal{R}_n
\end{aligned}$$

for all $i_1 = 0, 1, \ldots, j - 1$ and $i_2 = 0, 1, \ldots, j$. This is just the desired relation (8.92), and thus the proof is complete.

8.8 The SPRIM Algorithm

In this section, we apply the machinery of structure-preserving Padé-type reduced-order modeling to the class of Hermitian special second-order systems that describe RCL circuits.

Recall from Section 8.2 that a first-order formulation of RCL circuit equations is given by (8.10) with data matrices defined in (8.11). Here, we consider first-order systems (8.10) with data matrices of the slightly more general form

$$\mathcal{A} = \begin{bmatrix} -P_0 & -F \\ F^H & 0 \end{bmatrix}, \quad \mathcal{E} = \begin{bmatrix} P_1 & 0 \\ 0 & G \end{bmatrix}, \quad \mathcal{B} = \begin{bmatrix} B \\ 0 \end{bmatrix}. \tag{8.93}$$

Here, it is assumed that the subblocks P_0, P_1, and B have the same number of rows, and that the subblocks of $\mathcal{A}$ and $\mathcal{E}$ satisfy $P_0 \succeq 0$, $P_1 \succeq 0$, and $G \succ 0$. Note that systems (8.10) with matrices (8.93) are in particular Hermitian. Furthermore, the transfer function of such systems is given by

$$H(s) = \mathcal{B}^H \big(s\,\mathcal{E} - \mathcal{A}\big)^{-1} \mathcal{B}.$$

The PRIMA algorithm [OCP97, OCP98] is a reduction technique for first-order systems (8.10) with matrices of the form (8.93). PRIMA is a projection method that uses suitable basis matrices for the block-Krylov subspaces $\mathcal{K}_n\big(\mathcal{M}, \mathcal{R}\big)$; see [Fre99]. More precisely, let $\hat{\mathcal{V}}_n$ be any matrix satisfying (8.70) and (8.71). The corresponding n-th PRIMA model is then given by the projected data matrices

$$\hat{\mathcal{A}}_n := \hat{\mathcal{V}}_n^H \hat{\mathcal{A}}\, \hat{\mathcal{V}}_n, \quad \hat{\mathcal{E}}_n := \hat{\mathcal{V}}_n^H \hat{\mathcal{E}}\, \hat{\mathcal{V}}_n, \quad \hat{\mathcal{B}}_n := \hat{\mathcal{V}}_n^H \hat{\mathcal{B}}.$$

The associated transfer function is

$$\hat{H}_n(s) = \hat{\mathcal{B}}_n^H \big(s\,\hat{\mathcal{E}}_n - \hat{\mathcal{A}}_n\big)^{-1} \hat{\mathcal{B}}_n.$$

For n of the form (8.54), the PRIMA transfer function satisfies

$$\hat{H}(s) = H(s) + \mathcal{O}\left((s - s_0)^{j(n)}\right). \tag{8.94}$$

Recently, we introduced the SPRIM algorithm [Fre04a] as a structure-preserving and more accurate version of PRIMA. SPRIM employs the matrix $\mathcal{V}_n$ obtained from $\hat{\mathcal{V}}_n$ via the construction (8.72) and (8.73). The corresponding n-th SPRIM model is then given by the projected data matrices

$$\mathcal{A}_n := \mathcal{V}_n^H \mathcal{A}\, \mathcal{V}_n, \quad \mathcal{E}_n := \mathcal{V}_n^H \mathcal{E}\, \mathcal{V}_n, \quad \mathcal{B}_n := \mathcal{V}_n^H \mathcal{B}.$$

The associated transfer function is

$$H_n(s) = \mathcal{B}_n^H \big(s\,\mathcal{E}_n - \mathcal{A}_n\big)^{-1} \mathcal{B}_n.$$

In view of Theorem 8.7.2, we have

$$H(s) = H(s) + \mathcal{O}\left((s - s_0)^{2j(n)}\right),$$

which suggests that SPRIM is "twice" as accurate as PRIMA.

An outline of the SPRIM algorithm is as follows.

Algorithm 1 (SPRIM algorithm for special second-order systems)

- *Input: matrices*

$$\mathcal{A} = \begin{bmatrix} -P_0 & -F \\ F^H & 0 \end{bmatrix}, \quad \mathcal{E} = \begin{bmatrix} P_1 & 0 \\ 0 & G \end{bmatrix}, \quad \mathcal{B} = \begin{bmatrix} B \\ 0 \end{bmatrix},$$

  where the subblocks P_0, P_1, and B have the same number of rows, and the subblocks of $\mathcal{A}$ and $\mathcal{E}$ satisfy $P_0 \succeq 0$, $P_1 \succeq 0$, and $G \succ 0$;
  an expansion point $s_0 \in \mathbb{R}$.

- *Formally set*

$$\mathcal{M} = (s_0\,\mathcal{E} - \mathcal{A})^{-1}\,\mathcal{C}, \quad \mathcal{R} = (s_0\,\mathcal{E} - \mathcal{A})^{-1}\,\mathcal{B}.$$

- *Until n is large enough, run your favorite block Krylov subspace method (applied to $\mathcal{M}$ and $\mathcal{R}$) to construct the columns of the basis matrix*

$$\hat{\mathcal{V}}_n = \begin{bmatrix} v_1 & v_2 & \cdots & v_n \end{bmatrix}$$

of the n-th block Krylov subspace $\mathcal{K}_n\big(\mathcal{M}, \mathcal{R}\big)$, i.e.,

$$\operatorname{span} \hat{\mathcal{V}}_n = \mathcal{K}_n\big(\mathcal{M}, \mathcal{R}\big).$$

- *Let*

$$\hat{\mathcal{V}}_n = \begin{bmatrix} V_1 \\ V_2 \end{bmatrix}$$

be the partitioning of $\hat{\mathcal{V}}_n$ corresponding to the block sizes of $\mathcal{A}$ and $\mathcal{E}$.

- *Set*

$$\tilde{P}_0 = V_1^H P_1 V_1, \quad \tilde{F} = V_1^H F V_2, \quad \tilde{P}_1 = V_1^H P_1 V_1, \quad \tilde{G} = V_2^H G V_2,$$
$$\tilde{B} = V_1^H B,$$

and

$$\mathcal{A}_n = \begin{bmatrix} -\tilde{P}_0 & -\tilde{F} \\ \tilde{F}^H & 0 \end{bmatrix}, \quad \mathcal{E}_n = \begin{bmatrix} \tilde{P}_1 & 0 \\ 0 & \tilde{G} \end{bmatrix}, \quad \mathcal{B}_n = \begin{bmatrix} \tilde{B} \\ 0 \end{bmatrix}, \tag{8.95}$$

- *Output: the reduced-order model $\widetilde{H}_n$ in first-order form*

$$H_n(s) = \mathcal{B}_n^H \big(s\,\mathcal{E}_n - \mathcal{A}_n\big)^{-1} \mathcal{B}_n \tag{8.96}$$

and in second-order form

$$H_n(s) = \tilde{B}^H \left(s\,\tilde{P}_1 + \tilde{P}_0 + \frac{1}{s} \tilde{F} \tilde{G}^{-1} \tilde{F}^H \right)^{-1} \tilde{B}. \tag{8.97}$$

We remark that the main computational cost of the SPRIM algorithm is running the block Krylov subspace method to obtain $\hat{\mathcal{V}}_n$. This is the same as for PRIMA. Thus generating the PRIMA reduced-order model $\hat{H}_n$ and the SPRIM reduced-order model H_n involves the same computational costs.

On the other hand, when written in first-order form (8.96), it would appear that the SPRIM model has state-space dimension $2n$, and thus it would be twice as large as the corresponding PRIMA model. However, unlike the PRIMA model, the SPRIM model can always be represented in special second-order form (8.97); see Subsection 8.5.3. In (8.97), the matrices $\tilde{P}_1$, $\tilde{P}_0$, and $\tilde{P}_{-1} := \tilde{F}\tilde{G}^{-1}\tilde{F}^H$ are all of size $n \times n$, and the matrix $\tilde{B}$ is of size $n \times m$. These are the same dimensions as in the PRIMA model (8.94). Therefore, the SPRIM model H_n (written in second-order form (8.97)) and of the corresponding PRIMA model $\hat{H}_n$ indeed have the same state-space dimension n.

8.9 Numerical Examples

In this section, we present results of some numerical experiments with the SPRIM algorithm for special second-order systems. These results illustrate the higher accuracy of the SPRIM reduced-order models vs. the PRIMA reduced-order models.

8.9.1 A PEEC Circuit

The first example is a circuit resulting from the so-called PEEC discretization [Rue74] of an electromagnetic problem. The circuit is an RCL network consisting of 2100 capacitors, 172 inductors, 6990 inductive couplings, and a single resistive source that drives the circuit. See Chapter 22 for a more detailed description of this example. The circuit is formulated as a 2-port. We compare the PRIMA and SPRIM models corresponding to the same dimension n of the underlying block Krylov subspace. The expansion point $s_0 = 2\pi \times 10^9$ was used. In Figure 8.1, we plot the absolute value of the $(2,1)$ component of the 2×2-matrix-valued transfer function over the frequency range of interest. The dimension $n = 120$ was sufficient for SPRIM to match the exact transfer function. The corresponding PRIMA model of the same dimension, however, has not yet converged to the exact transfer function in large parts of the frequency range of interest. Figure 8.1 clearly illustrates the better approximation properties of SPRIM due to matching of twice as many moments as PRIMA.

8.9.2 A Package Model

The second example is a 64-pin package model used for an RF integrated circuit. Only eight of the package pins carry signals, the rest being either unused or carrying supply voltages. The package is characterized as a 16-port component (8 exterior and 8 interior terminals). The package model is described by approximately 4000 circuit elements, resistors, capacitors, inductors, and inductive couplings. See Chapter 22 for a more detailed description of this example and its mathematical model.

We again compare the PRIMA and SPRIM models corresponding to the same dimension n of the underlying block Krylov subspace. The expansion point $s_0 = 5\pi \times 10^9$ was used. In Figure 8.2, we plot the absolute value of one of the components of the 16×16-matrix-valued transfer function over the frequency range of interest. The state-space dimension $n = 80$ was sufficient for SPRIM to match the exact transfer function. The corresponding PRIMA model of the same dimension, however, does not match the exact transfer function very well near the high frequencies; see Figure 8.3.

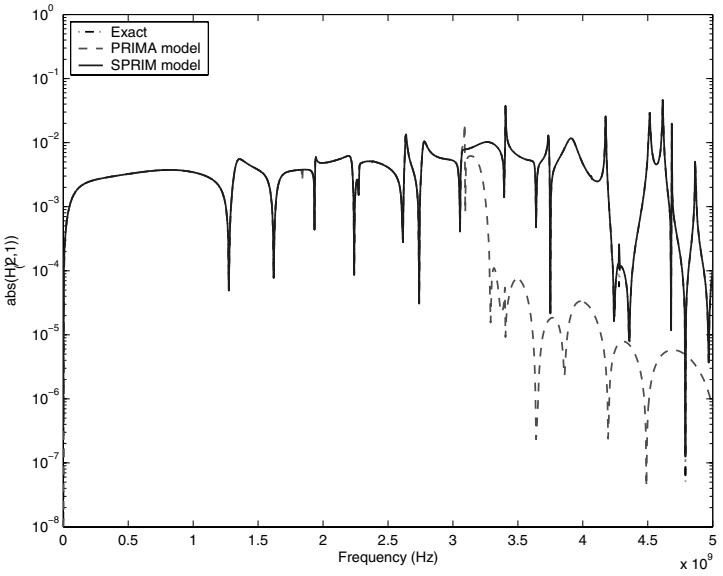


Fig. 8.1. $|H_{2,1}|$ for PEEC circuit

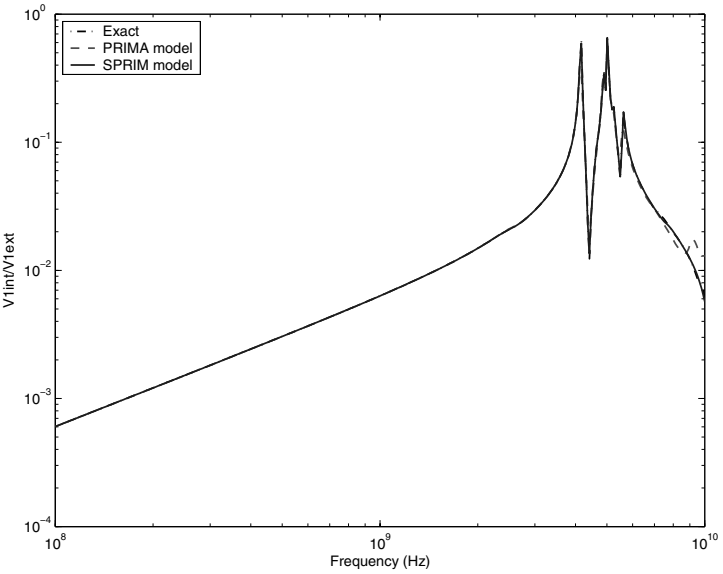


Fig. 8.2. The package model

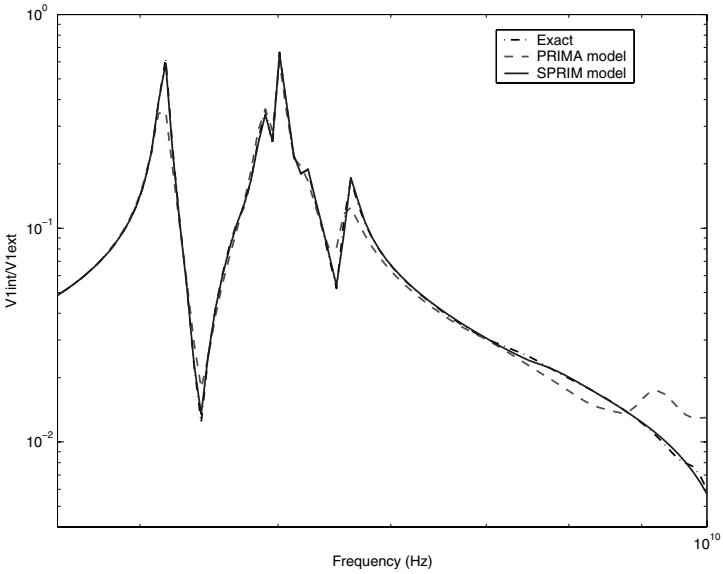


Fig. 8.3. The package model, high frequencies

8.9.3 A Mechanical System

Exploiting the equivalence (see, e.g., [LBEM00]) between RCL circuits and mechanical systems, both PRIMA and SPRIM can also be applied to reduced-order modeling of mechanical systems. Such systems arise for example in the modeling and simulation of MEMS devices. In Figure 8.4, we show a comparison of PRIMA and SPRIM for a finite-element model of a shaft. The expansion point $s_0 = \pi \times 10^3$ was used. The dimension $n = 15$ was sufficient for SPRIM to match the exact transfer function in the frequency range of interest. The corresponding PRIMA model of the same dimension, however, has not converged to the exact transfer function in large parts of the frequency range of interest. Figure 8.4 again illustrates the better approximation properties of SPRIM due to the matching of twice as many moments as PRIMA.

8.10 Concluding Remarks

We have presented a framework for constructing structure-preserving Padé-type reduced-order models of higher-order linear dynamical systems. The approach employs projection techniques and Krylov-subspace machinery for equivalent first-order formulations of the higher-order systems. We have shown that in the important case of Hermitian higher-order systems, our structure-preserving Padé-type model reduction is twice as accurate as in the general case. Despite this higher accuracy, the models produced by our approach are

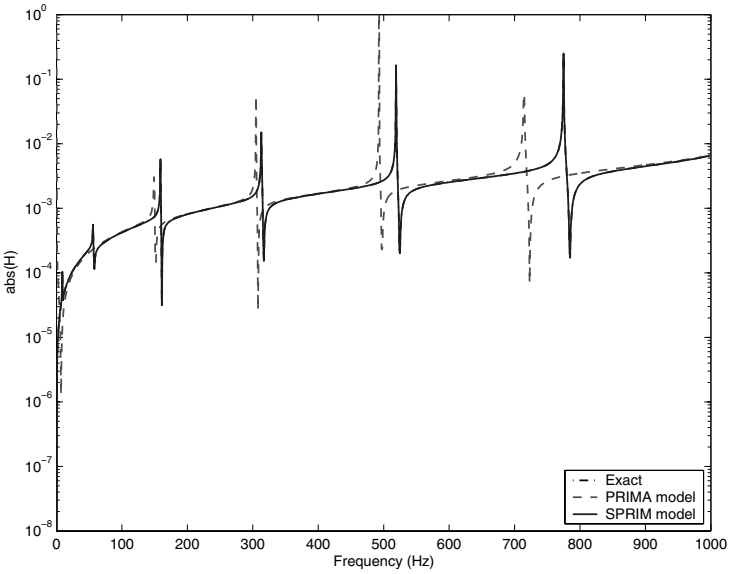


Fig. 8.4. A mechanical system

still not optimal in the Padé sense. This can be seen easily by comparing the degrees of freedom of general higher-order reduced models of prescribed state-space dimension, with the number of moments matched by the Padé-type models generated by our approach. Therefore, structure-preserving true Padé model reduction remains an open problem.

Our approach generates reduced models in higher-order form via equivalent first-order formulations. It would be desirable to have algorithms that construct the same reduced-order models in a more direct fashion, without the detour via first-order formulations. Another open problem is the most efficient and numerically stable algorithm to construct basis vectors of the structured Krylov subspaces that arise for the equivalent first-order formulations. Some related work on this problem is described in the recent report [Li04], but many questions remain open. Finally, the proposed approach is a projection technique, and as such, it requires the storage of all the vectors used in the projection. This clearly becomes an issue for systems with very large state-space dimension.

References

[ABFH00] J. I. Aliaga, D. L. Boley, R. W. Freund, and V. Hernández. A Lanczos-type method for multiple starting vectors. *Math. Comp.*, 69:1577–1601, 2000.

[AV73] B. D. O. Anderson and S. Vongpanitlerd. *Network Analysis and Synthesis*. Prentice-Hall, Englewood Cliffs, New Jersey, 1973.

[Bai02] Z. Bai. Krylov subspace techniques for reduced-order modeling of large-scale dynamical systems. *Appl. Numer. Math.*, 43(1–2):9–44, 2002.

[CLLC00] C.-K. Cheng, J. Lillis, S. Lin, and N. H. Chang. *Interconnect analysis and synthesis*. John Wiley & Sons, Inc., New York, New York, 2000.

[dVS87] C. de Villemagne and R. E. Skelton. Model reductions using a projection formulation. *Internat. J. Control*, 46(6):2141–2169, 1987.

[FF94] P. Feldmann and R. W. Freund. Efficient linear circuit analysis by Padé approximation via the Lanczos process. In *Proceedings of EURO-DAC '94 with EURO-VHDL '94*, pages 170–175, Los Alamitos, California, 1994. IEEE Computer Society Press.

[FF95] P. Feldmann and R. W. Freund. Efficient linear circuit analysis by Padé approximation via the Lanczos process. *IEEE Trans. Computer-Aided Design*, 14:639–649, 1995.

[Fre97] R. W. Freund. Circuit simulation techniques based on Lanczos-type algorithms. In C. I. Byrnes, B. N. Datta, D. S. Gilliam, and C. F. Martin, editors, *Systems and Control in the Twenty-First Century*, pages 171–184. Birkhäuser, Boston, 1997.

[Fre99] R. W. Freund. Passive reduced-order models for interconnect simulation and their computation via Krylov-subspace algorithms. In *Proc. 36th ACM/IEEE Design Automation Conference*, pages 195–200, New York, New York, 1999. ACM.

[Fre00] R. W. Freund. Krylov-subspace methods for reduced-order modeling in circuit simulation. *J. Comput. Appl. Math.*, 123(1–2):395–421, 2000.

[Fre03] R. W. Freund. Model reduction methods based on Krylov subspaces. *Acta Numerica*, 12:267–319, 2003.

[Fre04a] R. W. Freund. SPRIM: structure-preserving reduced-order interconnect macromodeling. In *Technical Digest of the 2004 IEEE/ACM International Conference on Computer-Aided Design*, pages 80–87, Los Alamitos, California, 2004. IEEE Computer Society Press.

[Fre04b] R. W. Freund. Krylov subspaces associated with higher-order linear dynamical systems. Technical report, December 2004. Submitted for publication. Available online from `http://www.math.ucdavis.edu/~freund/`.

[GLR82] I. Gohberg, P. Lancaster, and L. Rodman. *Matrix Polynomials*. Academic Press, New York, New York, 1982.

[Gri97] E. J. Grimme. Krylov projection methods for model reduction. PhD thesis, Department of Electrical Engineering, University of Illinois at Urbana-Champaign, Urbana-Champaign, Illinois, 1997.

[HRB75] C.-W. Ho, A. E. Ruehli, and P. A. Brennan. The modified nodal approach to network analysis. *IEEE Trans. Circuits and Systems*, CAS-22:504–509, June 1975.

[KGP94] S.-Y. Kim, N. Gopal, and L. T. Pillage. Time-domain macromodels for VLSI interconnect analysis. *IEEE Trans. Computer-Aided Design*, 13:1257–1270, 1994.

[Lan50] C. Lanczos. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *J. Res. Nat. Bur. Standards*, 45:255–282, 1950.

[LBEM00] R. Lozano, B. Brogliato, O. Egeland, and B. Maschke. *Dissipative Systems Analysis and Control*. Springer-Verlag, London, 2000.

[Li04] R.-C. Li. Structural preserving model reductions. Technical Report 04–02, Department of Mathematics, University of Kentucky, Lexington, Kentucky, 2004.

[OCP97] A. Odabasioglu, M. Celik, and L. T. Pileggi. PRIMA: passive reduced-order interconnect macromodeling algorithm. In *Technical Digest of the 1997 IEEE/ACM International Conference on Computer-Aided Design*, pages 58–65, Los Alamitos, California, 1997. IEEE Computer Society Press.

[OCP98] A. Odabasioglu, M. Celik, and L. T. Pileggi. PRIMA: passive reduced-order interconnect macromodeling algorithm. *IEEE Trans. Computer-Aided Design*, 17(8):645–654, 1998.

[Oda96] A. Odabasioglu. Provably passive RLC circuit reduction. M.S. thesis, Department of Electrical and Computer Engineering, Carnegie Mellon University, 1996.

[Rue74] A. E. Ruehli. Equivalent circuit models for three-dimensional multiconductor systems. *IEEE Trans. Microwave Theory Tech.*, 22:216–221, 1974.

[SC91] T.-J. Su and R. R. Craig, Jr. Model reduction and control of flexible structures using Krylov vectors. *J. Guidance Control Dynamics*, 14:260–267, 1991.

[VS94] J. Vlach and K. Singhal. *Computer Methods for Circuit Analysis and Design*. Van Nostrand Reinhold, New York, New York, second edition, 1994.

[ZKBP02] H. Zheng, B. Krauter, M. Beattie, and L. T. Pileggi. Window-based susceptance models for large-scale RLC circuit analyses. In *Proc. 2002 Design, Automation and Test in Europe Conference*, Los Alamitos, California, 2002. IEEE Computer Society Press.

[ZP02] H. Zheng and L. T. Pileggi. Robust and passive model order reduction for circuits containing susceptance elements. In *Technical Digest of the 2002 IEEE/ACM Int. Conf. on Computer-Aided Design*, pages 761–766, Los Alamitos, California, 2002. IEEE Computer Society Press.

9

Controller Reduction Using Accuracy-Enhancing Methods

Andras Varga

German Aerospace Center, DLR - Oberpfaffenhofen
Institute of Robotics and Mechatronics, D-82234 Wessling, Germany.
Andras.Varga@dlr.de

Summary. The efficient solution of several classes of controller approximation problems by using frequency-weighted balancing related model reduction approaches is considered. For certain categories of performance and stability enforcing frequency-weights, the computation of the frequency-weighted controllability and observability Gramians can be achieved by solving reduced order Lyapunov equations. All discussed approaches can be used in conjunction with square-root and balancing-free accuracy enhancing techniques. For a selected class of methods robust numerical software is available.

9.1 Introduction

The design of low order controllers for high order plants is a challenging problem both theoretically as well as from a computational point of view. The advanced controller design methods like the LQG/LTR loop-shaping, H_∞-synthesis, μ and linear matrix inequalities based synthesis methods produce typically controllers with orders comparable with the order of the plant. Therefore, the orders of these controllers tend often to be too high for practical use, where simple controllers are preferred over complex ones. To allow the practical applicability of advanced controller design methods for high order systems, the model reduction methods capable to address controller reduction problems are of primary importance. Comprehensive presentations of controller reduction methods and the reasons behind different approaches can be found in the textbook [ZDG96] and in the monograph [OA00].

The goal of controller reduction is to determine a low order controller starting from a high order one to ensure that the closed loop system formed from the original (high order) plant and low order controller behaves like the original plant with the original high order controller. Thus a basic requirement for controller reduction is preserving the closed-loop stability and many controller

reduction approaches have been derived to fulfil just this goal [AL89, LAL90]. However, to be useful, the low order controller resulting in this way must provide an acceptable performance degradation of the closed loop behavior. This led to methods which try to enforce also the preservation of closed-loop performance [AL89, GG98, Gu95, WSL01, EJL01].

In our presentation we focus on controller reduction methods related to balancing techniques. The *balanced truncation* (BT) based approach proposed in [Moo81] is a general method to reduce the order of stable systems. Bounds on the additive approximation errors have been derived in [Enn84, Glo84] and they theoretically establish the remarkable approximation properties of this approach. In a series of papers [LHPW87, TP87, SC89, Var91b] the underlying numerical algorithms for this method have been progressively improved and accompanying robust numerical software is freely available [Var01a]. The main computations in the so-called *square-root* and *balancing-free* accuracy enhancing method of [Var91b] is the high-accuracy computation of the controllability/observability Gramians (using square-root techniques) and employing well-conditioned truncation matrices (via a balancing-free approach). Note that the BT method is able to handle the reduction of unstable systems either via modal decomposition or coprime factorization techniques [Wal90, Var93]. A closely related approach is the *singular perturbation approximation* (SPA) [LA89] which later has been turned into a reliable computational technique in [Var91a].

Controller reduction problems are often formulated as frequency-weighted model reduction problems [AL89]. An extension of balancing techniques to address *frequency-weighted model reduction* (FWMR) problems has been proposed in [Enn84] by defining so-called frequency-weighted controllability and observability Gramians. The main difficulty with this method, is the lack of guarantee of stability of the reduced models in the case of two-sided weighting. To overcome this weakness, several improvements of the basic method of [Enn84] have been suggested in [LC92, WSL99, VA03], by proposing alternative choices of the frequency-weighted controllability and observability Gramians and/or employing the SPA approach instead of BT. Although still no *a priory* approximation error bounds for this method exist, the *frequency-weighted balanced truncation* (FWBT) or *frequency-weighted singular perturbation approximation* (FWSPA) approaches with the proposed enhancements are well-suited to solve many controller reduction problems. In contrast, *Hankel-norm approximation* (HNA) related approaches [Glo84, LA85] appear to be less suited for this class of problems due to special requirements to be fulfilled by the weights (e.g., anti-stable and anti-minimum-phase).

The recent developments in computational algorithms for controller reduction focus on fully exploiting the structural features of the *frequency-weighted controller reduction* (FWCR) problems [VA03, Var03b, Var03a]. In these papers it is shown that for several categories of performance and stability enforcing frequency-weights, the computation of the frequency-weighted controllability and observability Gramians can be done by solving reduced order

Lyapunov equations. Moreover, all discussed approaches can be used in conjunction with *square-root* and *balancing-free* accuracy enhancing techniques. For a selected class of methods robust numerical software is available.

The paper is organized as follows. In Section 9.2 we describe shortly the basic approaches to controller reduction. A general computational framework using balancing-related frequency-weighted methods is introduced in Section 9.3 and the related main aspects are addressed like the definition of frequency-weighted Gramians, using accuracy enhancing techniques, and algorithmic performance issues. The general framework is specialized to several controller reduction problems in Section 9.4, by addressing the reduction of both general as well as state feedback and observer-based controllers, in conjunction with various stability and performance preserving problem formulations. In each case, we discuss the applicability of square-root techniques and show the achievable computational effort saving by exploiting the problem structure. In Section 9.5 we present an overview of existing software. In Section 9.6 we present an example illustrating the typical controller reduction problematic.

Notation. Throughout the paper, the following notational convention is used. The bold letter notation $\mathbf{G}$ is used to denote a state-space system $\mathbf{G} := (A, B, C, D)$ with the *transfer-function matrix* (TFM)

$$G(\lambda) = C(\lambda I - A)^{-1}B + D := \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right].$$

Depending on the system type, λ is either the complex variable s appearing in the Laplace transform in the case of a continuous-time system or the variable z appearing in the Z-transform in the case of a discrete-time system. Throughout the paper we denote $G(\lambda)$ simply as G, when the system type is not relevant. The bold-notation is used consistently to denote system realizations corresponding to particular TFMs: $\mathbf{G}_1\mathbf{G}_2$ denotes the series coupling of two systems having the TFM $G_1(\lambda)G_2(\lambda)$, $\mathbf{G}_1 + \mathbf{G}_2$ represents the (additive) parallel coupling of two systems with TFM $G_1(\lambda) + G_2(\lambda)$, $\mathbf{G^{-1}}$ represents the inverse systems with TFM G^{-1}, $[\,\mathbf{G}_1\ \mathbf{G}_2\,]$ represents the realization of the compound TFM $[\,G_1\ G_2\,]$, etc.

9.2 Controller Reduction Approaches

Let $\mathbf{K} = (A_c, B_c, C_c, D_c)$ be a stabilizing controller of order n_c for an n-th order plant $\mathbf{G} = (A, B, C, D)$. We want to find $\mathbf{K}_r$, an r_c-th order approximation of $\mathbf{K}$ such that the reduced controller $\mathbf{K}_r$ is stabilizing and essentially preserves the closed-loop system performances of the original controller. To guarantee closed-loop stability, sometimes we would like to additionally preserve the same number of unstable poles in $\mathbf{K}_r$ as in $\mathbf{K}$.

To solve controller reduction problems, virtually any model reduction method in conjunction with the modal separation approach (to preserve the

unstable poles) can be employed. However, when employing general purpose model reduction methods to perform controller order reduction, the closed-loop stability and performance aspects are completely ignored and the resulting controllers are usually unsatisfactory.

To address stability and performance preserving issues, controller reduction problems are frequently formulated as FWMR problems with special weights [AL89]. This amounts to find $\mathbf{K}_r$, the r_c-th order approximation of $\mathbf{K}$ (having possibly the same number of unstable poles as $\mathbf{K}$), such that a weighted error of the form

$$\|W_o(K - K_r)W_i\|_\infty, \tag{9.1}$$

is minimized, where W_o and W_i are suitably chosen weighting TFMs.

Commonly used frequency-weights (see Section 9.3 and [AL89]) have minimal state-space realizations of orders as large as $n + n_c$ and thus employing general FWMR techniques could be expensive for high order plants/controllers, because they involve the computation of Gramians for systems of order $n + 2n_c$. A possible approach to alleviate the situation is to reduce first the weights using any of the standard methods (e.g., BT, SPA or HNA) and then apply the general FWBT or FWSPA approach with the enhancements proposed in [VA03]. Although apparently never discussed in the literature, this approach could be effective in some cases.

The idea to apply frequency-weighted balancing techniques to reduce the stable coprime factors of the controller has been discussed in several papers [AL89, LAL90, ZC95]. For example, given a *right coprime factorization* (RCF) $K = UV^{-1}$ of the controller, we would like to find a reduced controller in the RCF form $K_r = U_r V_r^{-1}$ such that

$$\left\| W_o \begin{bmatrix} U - U_r \\ V - V_r \end{bmatrix} W_i \right\|_\infty = \min . \tag{9.2}$$

Similarly, given a *left coprime factorization* (LCF) $K = V^{-1}U$ of the controller, we would like to find a reduced controller in the LCF form $K_r = V_r^{-1}U_r$ such that

$$\left\| \widetilde{W}_o [\, U - U_r \;\; V - V_r \,] \widetilde{W}_i \right\|_\infty = \min . \tag{9.3}$$

In (9.2) and (9.3) the weights have usually special forms to enforce either closed-loop stability [AL89, LAL90] or to preserve the closed-loop performance bounds for $\mathcal{H}_\infty$ controllers [GG98, Gu95, WSL01, EJL01]. The main appeal of coprime factorization based techniques is that in many cases (e.g., feedback controllers resulting from LQG, $\mathcal{H}_2$ or $\mathcal{H}_\infty$ designs) fractional representations of the controller can be obtained practically without any computation from the underlying synthesis approach. For example, this is the case for state feedback and observer-based controllers as well as for $\mathcal{H}_\infty$ controllers.

Interestingly, many stability/performance preserving controller reduction problems have very special structure which can be exploited when developing efficient numerical algorithms for controller reduction. For example, it

has been shown in [VA02] that for the frequency-weighted balancing related approaches applied to several controller reduction problems with the special stability/performance enforcing weights proposed in [AL89], the computation of Gramians can be done by solving reduced order Lyapunov equations. Similarly, it was recently shown in [Var03b] that this is also true for a class of frequency-weighted coprime factor controller reduction methods.

The approach which we pursue in this paper is the specialization of the FWMR methods to derive FWCR approaches which exploit all particular features of the underlying frequency-weighted problem. The main benefit of such a specialization in the case of arbitrary controllers is the cheaper computation of frequency-weighted Gramians by solving reduced order Lyapunov equations (typically of order $n+n_c$ instead the expected order $n+2n_c$). A further simplification arises when considering reduction of controllers resulting from LQG, $\mathcal{H}_2$ or $\mathcal{H}_\infty$ designs. For such controllers, the Gramians can be computed by solving Lyapunov equations only of order n_c. In what follows, we present an overview of recent enhancements obtained for different categories of problems. More details on each problem can be found in several recent works of the author [VA02, VA03, Var03b, Var03a].

9.3 Frequency-Weighted Balancing Framework

In this section we describe the general computational framework to perform FWCR using balancing-related approaches. The following procedure to solve the frequency-weighted approximation problem (9.1), with a possible unstable controller $\mathbf{K}$, is applicable (with obvious replacements) to solve the coprime factor approximation problems (9.2) and (9.3) as well, where obvious simplifications arise because the factors are stable systems.

FWCR Procedure.

1. Compute the additive stable-unstable spectral decomposition
$$\mathbf{K} = \mathbf{K}_s + \mathbf{K}_u,$$
where $\mathbf{K}_s$, of order n_{cs}, contains the stable poles of $\mathbf{K}$ and $\mathbf{K}_u$, of order $n_c - n_{cs}$, contains the unstable poles of $\mathbf{K}$.
2. Compute the controllability Gramian of $\mathbf{K}_s\mathbf{W}_i$ and the observability Gramian of $\mathbf{W}_o\mathbf{K}_s$ and define, according to [Enn84], [WSL99] or [VA03], appropriate n_{cs} order frequency-weighted controllability and observability Gramians P_w and Q_w, respectively.
3. Using P_w and Q_w in place of standard Gramians of $\mathbf{K}_s$, determine a reduced order approximation $\mathbf{K}_{sr}$ by applying the BT or SPA methods.
4. Form $\mathbf{K}_r = \mathbf{K}_{sr} + \mathbf{K}_u$.

This procedure originates from the works of Enns [Enn84] and automatically ensures that the resulting reduced order controller $\mathbf{K}_r$ has exactly the same

unstable poles as the original one, provided the approximation $\mathbf{K}_{sr}$ of the stable part $\mathbf{K}_s$ is stable. To guarantee the stability of $\mathbf{K}_{sr}$, specific choices of frequency-weighted Gramians have been proposed in [VA03] to enhance the original method proposed by Enns. In the following subsection, we present shortly the possible choices of the frequency-weighted controllability and observability Gramians to be employed in the **FWCR Procedure** and indicate the related computational aspects when employed in conjunction with square-root techniques.

9.3.1 Frequency-Weighted Gramians

To simplify the discussions we temporarily assume that the controller $\mathbf{K} = (A_c, B_c, C_c, D_c)$ is stable and the two weights W_o and W_i are also stable TFMs having minimal realizations of orders n_o and n_i, respectively. In the case of an unstable controller, the discussion applies to the stable part $\mathbf{K}_s$ of the controller.

Consider the minimal realizations of the frequency weights

$$\mathbf{W_o} = (A_o, B_o, C_o, D_o), \quad \mathbf{W_i} = (A_i, B_i, C_i, D_i)$$

and construct the realizations of $\mathbf{K}\mathbf{W}_i$ and $\mathbf{W_o}\mathbf{K}$ as

$$\mathbf{KW_i} = \left[\begin{array}{c|c} \overline{A}_i & \overline{B}_i \\ \hline \overline{C}_i & \overline{D}_i \end{array}\right] =: \left[\begin{array}{cc|c} A_c & B_cC_i & B_cD_i \\ 0 & A_i & B_i \\ \hline C_c & D_cC_i & D_cD_i \end{array}\right], \tag{9.4}$$

$$\mathbf{W_oK} = \left[\begin{array}{c|c} \overline{A}_o & \overline{B}_o \\ \hline \overline{C}_o & \overline{D}_o \end{array}\right] =: \left[\begin{array}{cc|c} A_o & B_oC_c & B_oD_c \\ 0 & A_c & B_c \\ \hline C_o & D_oC_c & D_oD_c \end{array}\right]. \tag{9.5}$$

Let $\overline{P}_i$ and $\overline{Q}_o$ be the controllability Gramian of $\mathbf{KW_i}$ and the observability Gramian of $\mathbf{W_oK}$, respectively. Depending on the system type, continuous-time (c) or discrete-time (d), $\overline{P}_i$ and $\overline{Q}_o$ satisfy the corresponding Lyapunov equations

$$(c)\ \left\{\begin{array}{l} \overline{A}_i\overline{P}_i + \overline{P}_i\overline{A}_i^T + \overline{B}_i\overline{B}_i^T = 0 \\ \overline{A}_o^T\overline{Q}_o + \overline{Q}_o\overline{A}_o + \overline{C}_o^T\overline{C}_o = 0 \end{array}\right., \quad (d)\ \left\{\begin{array}{l} \overline{A}_i\overline{P}_i\overline{A}_i^T + \overline{B}_i\overline{B}_i^T = \overline{P}_i \\ \overline{A}_o^T\overline{Q}_o\overline{A}_o + \overline{C}_o^T\overline{C}_o = \overline{Q}_o \end{array}\right. . \tag{9.6}$$

Partition $\overline{P}_i$ and $\overline{Q}_o$ in accordance with the structure of the matrices $\overline{A}_i$ and $\overline{A}_o$, respectively, i.e.

$$\overline{P}_i = \begin{bmatrix} P_{11} & P_{12} \\ P_{12}^T & P_{22} \end{bmatrix}, \quad \overline{Q}_o = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{12}^T & Q_{22} \end{bmatrix}, \tag{9.7}$$

where $P_E := P_{11}$ and $Q_E := Q_{22}$ are $n_c \times n_c$ matrices. The approach proposed by Enns [Enn84] defines

$$P_w = P_E, \quad Q_w = Q_E \tag{9.8}$$

as the frequency-weighted controllability and observability Gramians, respectively. Although successfully employed in many applications, the stability of the reduced controller is not guaranteed in the case of two-sided weighting, unless either $W_o = I$ or $W_i = I$. Occasionally, quite poor approximations result even for one-sided weighting.

In the context of FWMR, alternative choices of frequency-weighted Gramians guaranteeing stability have been proposed in [LC92] and [WSL99] (only for continuous-time systems). The choice proposed in [LC92] assumes that no pole-zero cancellations occur when forming $\mathbf{KW_i}$ and $\mathbf{W_oK}$, a condition which generally is not fulfilled by the special weights used in controller reduction problems. The alternative choice of [WSL99] has been improved in [VA03] by reducing the gap to Enns' choice and also extended to discrete-time systems.

The Gramians P_w and Q_w in the modified method of Enns proposed in [VA03] are determined as

$$P_w = P_V, \quad Q_w = Q_V, \tag{9.9}$$

where P_V and Q_V are the solutions of the appropriate pair of Lyapunov equations

$$(c)\ \begin{cases} A_c P_V + P_V A_c^T + \widetilde{B}_c \widetilde{B}_c^T = 0 \\ Q_V A_c + A_c^T Q_V + \widetilde{C}_c^T \widetilde{C}_c = 0 \end{cases}, \quad (d)\ \begin{cases} A_c P_V A_c^T + \widetilde{B}_c \widetilde{B}_c^T = P_V \\ A_c^T Q_V A_c + \widetilde{C}_c^T \widetilde{C}_c = Q_V \end{cases}. \tag{9.10}$$

Here, $\widetilde{B}_c$ and $\widetilde{C}_c$ are fictitious input and output matrices determined from the orthogonal eigendecompositions of the symmetric matrices X and Y defined as

$$(c)\ \begin{cases} X = -A_c P_E - P_E A_c^T \\ Y = -A_c^T Q_E - Q_E A_c \end{cases}, \quad (d)\ \begin{cases} X = -A_c P_E A_c^T + P_E \\ Y = -A_c^T Q_E A_c + Q_E \end{cases}. \tag{9.11}$$

The eigendecompositions of X and Y are given by

$$X = U \Theta U^T, \quad Y = V \Gamma V^T, \tag{9.12}$$

where Θ and Γ are real diagonal matrices. Assume that $\Theta = \operatorname{diag}(\Theta_1, \Theta_2)$ and $\Gamma = \operatorname{diag}(\Gamma_1, \Gamma_2)$ are determined such that $\Theta_1 > 0$ and $\Theta_2 \leq 0$, $\Gamma_1 > 0$ and $\Gamma_2 \leq 0$. Partition $U = [\, U_1 \; U_2 \,]$ and $V = [\, V_1 \; V_2 \,]$ in accordance with the partitioning of Θ and Γ, respectively. Then $\widetilde{B}$ and $\widetilde{C}$ are defined in [VA03] as

$$\widetilde{B}_c = U_1 \Theta_1^{\frac{1}{2}}, \quad \widetilde{C}_c = \Gamma_1^{\frac{1}{2}} V_1^T. \tag{9.13}$$

It is easy to see that with this choice of Gramians we have $P_V - P_E \geq 0$ and $Q_V - Q_E \geq 0$, thus, the triple $(A_c, \widetilde{B}_c, \widetilde{C}_c)$ is minimal provided the original triple (A_c, B_c, C_c) is minimal. Note that any combination of Gramians (P_E, Q_V), (P_V, Q_E), or (P_V, Q_V) guarantees the stability of approximations for two-sided weighting.

9.3.2 Accuracy Enhancing Techniques

There are two main techniques to enhance the accuracy of computations in model and controller reduction. One of them is the *square-root* technique introduced in [TP87] and relies on computing exclusively with better conditioned "square-root" quantities, namely, with the Cholesky factors of Gramians, instead of the Gramians themselves. In the context of unweighted additive error model reduction (e.g., employing BT, SPA or HNA methods), this involves to solve the Lyapunov equations satisfied by the Gramians directly for their Cholesky factors by using the well-know algorithms proposed by Hammarling [Ham82]. This is not generally possible in the case of FWMR/FWCR since the frequency-weighted Gramians P_w and Q_w are "derived" quantities defined, for example, via (9.8) or (9.9). In this subsection we show how square-root formulas can be employed to compute the frequency-weighted Gramians for the specific choices described in the previous subsection.

Assume $\overline{S}_i$ and $\overline{R}_o$ are the Cholesky factors of $\overline{P}_i$ and $\overline{Q}_o$ in (9.7), respectively, satisfying $\overline{P}_i = \overline{S}_i \overline{S}_i^T$ and $\overline{Q}_o = \overline{R}_o^T \overline{R}_o$. These factors are upper triangular and can be computed using the method of Hammarling [Ham82] to solve the Lyapunov equations (9.6) directly for the Cholesky factors. The solution of these Lyapunov equations involves the reduction of each of the matrices $\overline{A}_i$ and $\overline{A}_o$ to a *real Schur form* (RSF). For efficiency reasons the reduction of A, A_i and A_o to RSF is preferably done independently and only once. This ensures that $\overline{A}_i$ and $\overline{A}_o$ in the realizations (9.4) of $\mathbf{KW_i}$ and (9.5) of $\mathbf{W_oK}$ are automatically in RSF.

If we partition $\overline{S}_i$ and $\overline{R}_o$ in accordance with the partitioning of $\overline{P}_i$ and $\overline{Q}_o$ in (9.7) as

$$\overline{S}_i = \begin{bmatrix} S_{11} & S_{12} \\ 0 & S_{22} \end{bmatrix}, \quad \overline{R}_o = \begin{bmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{bmatrix}$$

we have immediately that the Cholesky factors of $P_E = S_E S_E^T$ and $Q_E = R_E^T R_E$ corresponding to Enns' choice satisfy

$$S_E S_E^T = S_{11} S_{11}^T + S_{12} S_{12}^T = [\, S_{11} \; S_{12} \,][\, S_{11} \; S_{12} \,]^T, \tag{9.14}$$

$$R_E^T R_E = R_{12}^T R_{12} + R_{22}^T R_{22} = \begin{bmatrix} R_{12} \\ R_{22} \end{bmatrix}^T \begin{bmatrix} R_{12} \\ R_{22} \end{bmatrix}. \tag{9.15}$$

Thus, to obtain S_E the RQ-factorization of the matrix $[\, S_{11} \; S_{12} \,]$ must be additionally performed, while for obtaining R_E the QR-factorization of $[\, R_{12}^T \; R_{22}^T \,]^T$ must be performed. Both these factorizations can be computed using well established factorization updating techniques [GGMS74] which fully exploit the upper triangular shapes of S_{11} and R_{22}.

For the choice (9.9) of Gramians, the Cholesky factors of $P_V = S_V S_V^T$ and $Q_V = R_V^T R_V$ result by solving (9.10) directly for these factors using the algorithm of Hammarling [Ham82]. Note that for computing X and Y, we can use the Cholesky factors S_E and R_E determined above for Enns' choice.

Assume that $P_w = S_w S_w^T$ and $Q_w = R_w^T R_w$ are the Cholesky factorizations of the frequency weighted Gramians corresponding to one of the above choices of the Gramians (9.8) or (9.9). To determine the reduced order controller we determine two *truncation matrices* L and T such that the reduced controller is given by

$$(A_{cr}, B_{cr}, C_{cr}, D_{cr}) = (LA_cT,\ LB_c,\ C_cT,\ D_c).$$

The computation of L and T can be done from the singular value decomposition (SVD)

$$R_w S_w = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \mathrm{diag}(\Sigma_1, \Sigma_2) \begin{bmatrix} V_1 & V_2 \end{bmatrix}^T, \tag{9.16}$$

where

$$\Sigma_1 = \mathrm{diag}(\sigma_1, \ldots, \sigma_{r_c}), \quad \Sigma_2 = \mathrm{diag}(\sigma_{r_c+1}, \ldots, \sigma_{n_c}),$$

and $\sigma_1 \geq \ldots \geq \sigma_{r_c} > \sigma_{r_c+1} \geq \ldots \geq \sigma_{n_c} \geq 0$. To compute the SVD in (9.16), instead of using standard algorithms as those described in [GV89], special numerically stable algorithms for matrix products can be employed to avoid the forming of the product $R_w S_w$ [GSV00].

The so-called *square-root* (**SR**) methods determine L and T as [TP87]

$$L = \Sigma_1^{-1/2} U_1^T R_w, \qquad T = S_w V_1 \Sigma_1^{-1/2}. \tag{9.17}$$

A potential disadvantage of this choice is that accuracy losses can be induced in the reduced controller if either of the truncation matrices L or T is ill-conditioned (i.e., nearly rank deficient). Note that in the case of BT based model reduction, the above choice leads, in the continuous-time, to *balanced* reduced models (i.e., the corresponding Gramians are equal and diagonal).

The second technique to enhance accuracy is the computation of well-conditioned truncation matrices L and T, by avoiding completely any kind of balancing implied by using the (**SR**) formulas (9.17). This leads to a *balancing-free* (**BF**) approach (originally proposed in [SC89]) in which L and T are always well-conditioned. A *balancing-free square-root* (**BFSR**) algorithm which combines the advantages of the **BF** and **SR** approaches has been introduced in [Var91b]. L and T are determined as

$$L = (Y^T X)^{-1} Y^T, \qquad T = X,$$

where X and Y are $n_c \times r_c$ matrices with orthogonal columns computed from two QR decompositions

$$S_w V_1 = XW, \quad R_w^T U_1 = YZ$$

with W and Z non-singular and upper-triangular. The reduced controller obtained in this way is related to that one obtained by the **SR** approach by a non-orthogonal state coordinate transformation. Since the accuracy of the **BFSR** algorithm is usually better than either of **SR** or **BF** techniques, this approach is the default option in high performance controller reduction software (see Section 9.5).

Assume now that the singular value decomposition of $R_w S_w$ is

$$R_w S_w = \left[\, U_1 \; U_2 \; U_3 \,\right] \mathrm{diag}(\Sigma_1, \Sigma_2, 0) \left[\, V_1 \; V_2 \; V_3 \,\right]^T,$$

where

$$\Sigma_1 = \mathrm{diag}(\sigma_1, \ldots, \sigma_{r_c}), \quad \Sigma_2 = \mathrm{diag}(\sigma_{r_c+1}, \ldots, \sigma_{\overline{n}_c}),$$

and $\sigma_1 \geq \ldots \geq \sigma_{r_c} > \sigma_{r_c+1} \geq \ldots \geq \sigma_{\overline{n}_c} > 0$. Assume we employ the **SR** formulas to compute a minimal realization of the controller of order $\overline{n}_c$ as

$$\left[\begin{array}{c|c} LA_cT & LB_c \\ \hline C_cT & D_c \end{array}\right] = \left[\begin{array}{cc|c} A_{c,11} & A_{c,12} & B_{c,1} \\ A_{c,21} & A_{c,22} & B_{c,2} \\ \hline C_{c,1} & C_{c,2} & D_c \end{array}\right],$$

where the system matrices are compatibly partitioned with $A_{c,11} \in R^{r_c \times r_c}$. The SPA method (see [LA89]) determines the reduced controller matrices as

$$\left[\begin{array}{c|c} A_{cr} & B_{cr} \\ \hline C_{cr} & D_{cr} \end{array}\right] = \left[\begin{array}{c|c} A_{c,11} - A_{c,12}A_{c,22}^{-1}A_{c,21} & B_{c,1} - A_{c,12}A_{c,22}^{-1}B_{c,2} \\ \hline C_{c,1} - C_{c,2}A_{c,22}^{-1}A_{c,21} & D_c - C_{c,2}A_{c,22}^{-1}B_{c,2} \end{array}\right].$$

This approach has been termed the **SR** SPA method. Note that the resulting reduced controller is in a balanced state-space coordinate form both in continuous- as well as in discrete-time cases.

A **SRBF** version of the SPA method has been proposed in [Var91a] to combine the advantages of the **BF** and **SR** approaches. The truncation matrices L and T are determined as

$$L = \left[\begin{array}{c} (Y_1^T X_1)^{-1} Y_1^T \\ (Y_2^T X_2)^{-1} Y_2^T \end{array}\right], \qquad T = \left[\, X_1 \; X_2 \,\right],$$

where X_1 and Y_1 are $\overline{n}_c \times r_c$ matrices, and X_2 and Y_2 are $\overline{n}_c \times (\overline{n}_c - r_c)$ matrices. All these matrices with orthogonal columns are computed from the QR decompositions

$$S_w V_i = X_i W_i, \quad R_w^T U_i = Y_i Z_i, \quad i = 1, 2$$

with W_i and Z_i non-singular and upper-triangular.

9.3.3 Algorithmic Efficiency Issues

The two main computational problems of controller reduction by using the frequency weighted BT or SPA approaches are the determination of frequency-weighted Gramians and the computation of the corresponding truncation matrices. All computation ingredients for these computations are available as robust numerical implementations either in the LAPACK [ABB99] or SLICOT [BMSV99] libraries. To compare the effectiveness of different methods, we roughly evaluate in what follows the required computational effort for

the main computations in terms of required *floating-point operations* (*flops*). Note that 1 *flop* corresponds to 1 addition/subtraction or 1 multiplication/division performed on the floating point processor. In our evaluations we tacitly assume that the number of system inputs m and system outputs p satisfy $m, p \ll n_c$, thus many computations involving the input and output matrices (e.g., products) are negligible.

The main computational ingredient for computing Gramians is the solution of Lyapunov equations as those in (9.6). This involves the reduction of the matrices $\overline{A}_i$ and $\overline{A}_o$ to the *real Schur form* (RSF) using the Francis' QR-algorithm [GV89]. By exploiting the block upper triangular structure of these matrices, this reduction can be performed by reducing independently A_i, A_c and A_o, which amounts to about $25n_i^3$, $25n_c^3$ and $25n_o^3$ *flops*, respectively. The Cholesky factors $\overline{S}_i$ and $\overline{R}_o$ of Gramians $\overline{P}_i$ and $\overline{Q}_o$ in (9.6) can be computed using the method of Hammarling [Ham82] and this requires about $8(n_i+n_c)^3$ and $8(n_o+n_c)^3$ *flops*, respectively. The computation of the Cholesky factors S_E and R_E using the algorithm of [GGMS74] for the updating formulas (9.14) and (9.15) requires additionally about $2n_in_c^2$ and $2n_on_c^2$ *flops*, respectively. Thus, the computation of the pair (S_E, R_E) requires

$$N_E = 25(n_i^3 + n_c^3 + n_o^3) + 8(n_i + n_c)^3 + 8(n_o + n_c)^3 + 2(n_i + n_o)n_c^2 \quad (9.18)$$

flops. Note that N_E represents the cost of evaluating Gramians when applying the FWBT or FWSPA approaches to solve the controller reduction problem as a general FWMR problem, without any structure exploitation. In certain problems with two-sided weights, the input and output weights share the same state matrix. In this case $n_i = n_o$ and N_E reduces with $25n_i^3$ *flops*.

The computation of one of the factors S_V (or R_V) corresponding to the modified Lyapunov equations (9.10) requires up to $19.5n_c^3$ *flops*, of which about $9n_c^3$ *flops* account for the eigendecomposition of X in (9.12) to form the constant term of the Lyapunov equation satisfied by P_V and $8n_c^3$ *flops* account to solve the Lyapunov equation (9.10) for the factor S_V. Note that the reduction of A_c to a RSF is performed only once, when computing the factors S_E and R_E. The additional number of operations required by different choices of the frequency-weighted Gramians is

$$N_V = \begin{cases} 0, & (S_w, R_w) = (S_E, R_E) \\ 19.5n_c^3, & (S_w, R_w) = (S_V, R_E) \text{ or } (S_w, R_w) = (S_E, R_V) \\ 39n_c^3, & (S_w, R_w) = (S_V, R_V) \end{cases} .$$

The determination of the truncation matrices L and T involves the computation of the singular value decomposition of the $n_c \times n_c$ matrix R_wS_w, which requires at least $N_T = 22n_c^3$ *flops*. The rest of computations is negligible if $r_c \ll n_c$.

From the above analysis it follows that for n_i and n_o of comparable sizes with n_c, the term N_E, which accounts for the computations of the Cholesky factors for Enns' choice of the frequency weighted Gramians, has the largest

contribution to $N_{tot} = N_E + N_V + N_T$, the total number of operations. Note that $N_V + N_T$ depends only on the controller order n_c and the choice of Gramian modification scheme, thus this part of N_{tot} appears as "constant" in all evaluations of the computational efforts. It is interesting to see the relative values of N_E and N_{tot} for some typical cases. For an unweighted controller reduction problem $N_E = 41n_c^3$ and $N_{tot} = 63n_c^3$, thus $N_E/N_{tot} = 0.65$. These values of N_E and N_{tot} can be seen as lower limits for all controller reduction problems using balancing related approaches. In the case when $n_i, n_o \ll n_c$, $N_E \approx 41n_c^3$ and $63n_c^3 \leq N_{tot} \leq 102n_c^3$, thus in this case $0.40 \leq N_E/N_{tot} \leq 0.65$. At the other extreme, assuming the typical values of $n_c = n$, $n_i = n_o = 2n$ for a state feedback and observer-based controller, we have $N_E = 865n^3$ and $887n_c^3 \leq N_{tot} \leq 926n_c^3$, and thus the ratio of N_E/N_{tot} satisfies $0.93 \leq N_E/N_{tot} \leq 0.98$. These figures show that solving FWCR problems can be tremendously expensive when employing general purpose model reduction algorithms. In the following sections we show that for several classes of controller reduction problems, structure exploitation can lead to significant computation savings expressed by much smaller values of N_E.

9.4 Efficient Solution of Controller Reduction Problems

To develop efficient numerical methods for controller reduction, the general framework for controller reduction described in the previous section needs to be specialized to particular classes of problems by fully exploiting the underlying problem structures. When deriving efficient specialized versions of the **FWCR Algorithm**, the main computational saving arises in determining the frequency-weighted Gramians for each particular case via the corresponding Cholesky factors. In what follows we consider several controller reduction problems with particular weights and give the main results concerning the computation of Gramians. We focus only on Enns' choice, since it enters also in all other alternative choices discussed in the previous section.

9.4.1 Frequency-Weighted Controller Reduction

We consider the solution of the FWCR problem (9.1) for the specific stability and performance preserving weights discussed in [AL89]. To enforce closed-loop stability, one-sided weights of the form

$$\text{SW1:} \qquad W_o = (I + GK)^{-1}G, \quad W_i = I, \tag{9.19}$$

or

$$\text{SW2:} \qquad W_o = I, \quad W_i = G(I + KG)^{-1}, \tag{9.20}$$

can be used, while performance-preserving considerations lead to two-sided weights

$$\text{PW:} \qquad W_o = (I + GK)^{-1}G, \quad W_i = (I + GK)^{-1}\,, \tag{9.21}$$

The unweighted reduction corresponds to the weights

$$\text{UW:} \qquad W_o = I, \quad W_i = I. \tag{9.22}$$

It can be shown (see [ZDG96]), that for the weights (9.19) and (9.20) the stability of the closed-loop system is guaranteed if $\|W_o(K - K_r)W_i\|_\infty < 1$, provided K and K_r have the same number of unstable poles. Similarly, minimizing $\|W_o(K - K_r)W_i\|_\infty$ for the weights in (9.21) ensures the best matching of the closed-loop TFM for a given order of K_r.

To solve the FWCR problems corresponding to the above weights, we consider both the case of a general stabilizing controller as well as the case of state feedback and observer-based controllers. In each case we show how to compute efficiently the Cholesky factors of frequency-weighted Gramians in order to apply the **SR** and **SRBF** accuracy enhancing techniques. Finally, we give estimates of the necessary computational efforts and discuss the achieved saving by using structure exploitation.

General Controller

Since the controller can be generally unstable, only the stable part of the controller is reduced and a copy of the unstable part is kept in the reduced controller. Therefore, we assume a state-space representation of the controller with A_c already reduced to a block-diagonal form

$$\mathbf{K} = \left[\begin{array}{c|c} A_c & B_c \\ \hline C_c & D_c \end{array}\right] = \left[\begin{array}{cc|c} A_{c1} & 0 & B_{c1} \\ 0 & A_{c2} & B_{c2} \\ \hline C_{c1} & C_{c2} & D_c \end{array}\right], \tag{9.23}$$

where $\Lambda(A_{c1}) \subset \mathrm{C}^+$ and $\Lambda(A_{c2}) \subset \mathrm{C}^-$. Here C^- denotes the open left half complex plane of C in a continuous-time setting or the interior of the unit circle in a discrete-time setting, while C^+ denotes the complement of C^- in C. The above form corresponds to an additive decomposition of the controller TFM as $K = K_u + K_s$, where $\mathbf{K}_u = (A_{c1}, B_{c1}, C_{c1}, 0)$ contains the unstable poles of $\mathbf{K}$ and $\mathbf{K}_s = (A_{c2}, B_{c2}, C_{c2}, D_c)$, of order n_{cs}, contains the stable poles of $\mathbf{K}$.

For our developments, we build the state matrix of the realizations of the weights in (9.19), (9.20), or (9.21) in the form

$$A_w = \left[\begin{array}{cc} A - BD_cR^{-1}C & B\widetilde{R}^{-1}C_c \\ -B_cR^{-1}C & A_c - B_cR^{-1}DC_c \end{array}\right],$$

where $R = I + DD_c$ and $\widetilde{R} = I + D_cD$. Since the controller is stabilizing, A_w has all its eigenvalues in C^-.

The following theorem, proved in [VA02], extends the results of [LAL90, SM96] to the case of an arbitrary stabilizing controller:

Theorem 9.4.1 *For a given n-th order system* $\mathbf{G} = (A, B, C, D)$ *assume that* $\mathbf{K} = (A_c, B_c, C_c, D_c)$ *is an* n_c*-th order stabilizing controller with* $I + DD_c$ *nonsingular. Then the frequency-weighted Gramians for Enns' method [Enn84] applied to the frequency-weighted controller reduction problems with weights defined in (9.19), (9.20), or (9.21) can be computed by solving the corresponding Lyapunov equations of order at most* $n + n_c$ *as follows:*

1. *For* $W_o = (I + GK)^{-1}G$ *and* $W_i = I$, P_E *satisfies*

$$(c)\ A_{c2}P_E + P_E A_{c2}^T + B_{c2}B_{c2}^T = 0, \quad (d)\ A_{c2}P_E A_{c2}^T + B_{c2}B_{c2}^T = P_E \tag{9.24}$$

and Q_E *is the* $n_{cs} \times n_{cs}$ *trailing block of* Q_o *satisfying*

$$(c)\ A_w^T Q_o + Q_o A_w + C_o^T C_o = 0, \quad (d)\ A_w^T Q_o A_w + C_o^T C_o = Q_o \tag{9.25}$$

with $C_o = \left[-R^{-1}C \ -R^{-1}DC_c \right]$.

2. *For* $W_o = I$ *and* $W_i = G(I + GK)^{-1}$, P_E *is the* $n_{cs} \times n_{cs}$ *trailing block of* P_i *satisfying*

$$(c)\ A_w P_i + P_i A_w^T + B_i B_i^T = 0, \quad (d)\ A_w P_i A_w^T + B_i B_i^T = P_i \tag{9.26}$$

with $B_i = \begin{bmatrix} -B\widetilde{R}^{-1} \\ B_c D\widetilde{R}^{-1} \end{bmatrix}$ *and* Q_E *satisfies*

$$(c)\ A_{c2}^T Q_E + Q_E A_{c2} + C_{c2}^T C_{c2} = 0, \quad (d)\ A_{c2}^T Q_E A_{c2} + C_{c2}^T C_{c2} = Q_E \tag{9.27}$$

3. *For* $W_o = (I+GK)^{-1}G$ *and* $W_i = (I+GK)^{-1}$, P_E *is the* $n_{cs} \times n_{cs}$ *trailing block of* P_i *satisfying (9.26) with* $B_i = \begin{bmatrix} BD_c R^{-1} \\ B_c R^{-1} \end{bmatrix}$ *and* Q_E *is the* $n_{cs} \times n_{cs}$ *trailing block of* Q_o *satisfying (9.25).*

State Feedback and Observer-Based Controller

Simplifications arise also in the case of a state feedback and full order observer-based controller of the form

$$\mathbf{K} = \left[\begin{array}{c|c} A + BF + LC + LDF & -L \\ \hline F & 0 \end{array}\right]. \tag{9.28}$$

The following result extends *Lemma 1* of [LAL90] to the case of possibly unstable controllers.

Corollary 9.4.2 *For a given n-th order system* $\mathbf{G} = (A, B, C, D)$ *suppose that* F *is a state feedback gain and* L *is a state estimator gain, such that* $A + BF$ *and* $A + LC$ *are stable. Then the frequency-weighted Gramians for Enns' method [Enn84] applied to the frequency-weighted controller reduction problems with weights defined in (9.19), (9.20), or (9.21) can be computed by solving Lyapunov equations of order at most* $2n$.

In the case of state feedback and observer-based controllers important computational effort saving results if we further exploit the problem structure. In this case

$$A_w = \begin{bmatrix} A & BF \\ -LC & A+BF+LC \end{bmatrix}$$

and this matrix can be put in an upper block diagonal form using the transformation matrix

$$T = \begin{bmatrix} I & 0 \\ I & I \end{bmatrix}.$$

We obtain the transformed matrices $\widetilde{A}_w := T^{-1}A_wT$, $\widetilde{B}_i := T^{-1}B_i$, and $\widetilde{C}_o := C_oT$, where

$$\widetilde{A}_w = \begin{bmatrix} A+BF & BF \\ 0 & A+LC \end{bmatrix}.$$

If $\widetilde{P}_i$ and $\widetilde{Q}_o$ satisfy

$$(c)\ \begin{cases} \widetilde{A}_w\widetilde{P}_i + \widetilde{P}_i\widetilde{A}_w^T + \widetilde{B}_i\widetilde{B}_i^T = 0 \\ \widetilde{A}_w^T\widetilde{Q}_o + \widetilde{Q}_o\widetilde{A}_w + \widetilde{C}_o^T\widetilde{C}_o = 0 \end{cases},\quad (d)\ \begin{cases} \widetilde{A}_w\widetilde{P}_i\widetilde{A}_w^T + \widetilde{B}_i\widetilde{B}_i^T = \widetilde{P}_i \\ \widetilde{A}_w^T\widetilde{Q}_o\widetilde{A}_w + \widetilde{C}_o^T\widetilde{C}_o = \widetilde{Q}_o \end{cases}, \quad (9.29)$$

then P_i in (9.26) and Q_o in (9.25) are given by $P_i = T\widetilde{P}_iT^T$ and $Q_o = T^{-T}\widetilde{Q}_oT^{-1}$, respectively. The computational saving arises from the need to reduce A_w to a RSF when solving the Lyapunov equations (9.25) and (9.26). Instead of reducing the $2n \times 2n$ matrix A_w, we can reduce two $n \times n$ matrices $A+BF$ and $A+LC$ to obtain $\widetilde{A}_w$ in a RSF. This means a 4 times speedup of computations for this step.

Square-Root Techniques

We can employ the method of [Ham82] to solve (9.26) and (9.25) directly for the Cholesky factors S_i of $P_i = S_iS_i^T$ and R_o of $Q_o = R_o^TR_o$, respectively. In the case of an unstable controller, we assume a state-space realization of $\mathbf{K}$ as in (9.23) with the $n_{cs} \times n_{cs}$ matrix A_{c2} containing the stable eigenvalues of A_c. If we partition S_i and R_o in the form

$$S_i = \begin{bmatrix} S_{11} & S_{12} \\ 0 & S_{22} \end{bmatrix}, \quad R_o = \begin{bmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{bmatrix},$$

where both S_{22} and R_{22} are $n_{cs} \times n_{cs}$, then the Cholesky factor of the trailing block of P_i in (9.26) corresponding to the stable part of $\mathbf{K}$ is simply $S_E = S_{22}$, while the Cholesky factor R_E of the trailing block of Q_o in (9.25) satisfies $R_E^TR_E = R_{22}^TR_{22} + R_{12}^TR_{12}$. Thus the computation of R_E involves an additional QR-decomposition of $[\,R_{22}^T\ R_{12}^T\,]^T$ and can be computed using standard updating techniques [GGMS74]. Updating can be avoided in the case of the one-sided weight $W_o = (I+GK)^{-1}G$, by using alternative state-space

realizations of $\mathbf{W_o}$ and $\mathbf{K}$. For details, see [VA02]. Still in the case of two-sided weighting with $W_o = (I + GK)^{-1}G$ and $W_i = (I + GK)^{-1}$ we prefer the approach of the Theorem 9.4.1 with $\mathbf{W_i}$ and $\mathbf{W_o}$ sharing the same state matrix A_w, because the computation of both Gramians can be done with a single reduction of this $(n + n_c) \times (n + n_c)$ matrix to the RSF. In this case the cost to compute the two Gramians is only slightly larger than for one Gramian.

For a state feedback and full order observer-based controller, let $\widetilde{S}_i$ be the Cholesky factor of $\widetilde{P}_i$ in (9.29) partitioned as

$$\widetilde{S}_i = \begin{bmatrix} \widetilde{S}_{11} & \widetilde{S}_{12} \\ 0 & \widetilde{S}_{22} \end{bmatrix}.$$

The $n_{cs} \times n_{cs}$ Cholesky factor S_E corresponding to the trailing $n_{cs} \times n_{cs}$ part of P_i is the trailing $n_{cs} \times n_{cs}$ block of an upper triangular matrix $\widehat{S}_{22}$ which satisfies

$$\widehat{S}_{22}\widehat{S}_{22}^T = \widetilde{S}_{11}\widetilde{S}_{11}^T + (\widetilde{S}_{12} + \widetilde{S}_{22})(\widetilde{S}_{12} + \widetilde{S}_{22})^T.$$

$\widehat{S}_{22}$ can be computed easily from the RQ-decomposition of $\begin{bmatrix} \widetilde{S}_{11} & \widetilde{S}_{12} + \widetilde{S}_{22} \end{bmatrix}$ using standard factorization updating formulas [GGMS74]. No difference appears in the computation of the Cholesky factor R_E.

Efficiency Issues

In Table 9.1 we give for the different weights (assuming $n_{cs} = n_c$) the number of operations $\widetilde{N}_E$ necessary to determine the Cholesky factors of the frequency-weighted Gramians and the achieved operation savings $\Delta_E = N_E - \widetilde{N}_E$, (see also (9.18) for N_E) with respect to using standard FWMR techniques to reduce a general controller:

Table 9.1. Operation counts: general controller

Weight	$\widetilde{N}_E$	Δ_E
SW1/SW2	$33(n + n_c)^3 + 33n_c^3$	$24n^2n_c + 74nn_c^2 + 58n_c^3$
PW	$41(n + n_c)^3 + 2nn_c^2$	$48n^2n_c + 146nn_c^2 + 141n_c^3$

In the case of a state feedback and observer-based controller ($n_c = n$), the corresponding values are shown in Table 9.2:
Observe the large computational effort savings obtained in all cases through structure exploitation for both general as well as state feedback controllers. For example, for the SW1/SW2 and PW problems with a state feedback controller the effort to compute the Gramians is about 2.7 times less than without structure exploitation.

Table 9.2. Operation counts: observer-based controller

Weight	$\widetilde{N}_E$	Δ_E
SW1/SW2	$122n^3$	$331n^3$
PW	$181n^3$	$484n^3$

9.4.2 Stability Preserving Coprime Factor Reduction

In this subsection, we discuss the efficient solution of frequency-weighted balancing-related coprime factor controller reduction problems for the special stability preserving frequency-weights proposed in [LAL90]. We show that for both general controllers as well as for state feedback and observer-based controllers, the computation of frequency-weighted Gramians for the coprime factor controller reduction can be done efficiently by solving lower order Lyapunov equations. Further, we show that these factors can be directly obtained in Cholesky factored forms allowing the application of the **SRBF** accuracy enhancing technique.

The following stability enforcing one-sided weights are used: for the right coprime factor reduction problem the weights are

$$\text{SRCF:} \qquad W_o = V^{-1}(I + GK)^{-1}[\,G\;I\,], \quad W_i = I, \tag{9.30}$$

while for the left coprime factor reduction the weights are

$$\text{SLCF:} \qquad \widetilde{W}_o = I, \quad \widetilde{W}_i = \begin{bmatrix} G \\ I \end{bmatrix} (I + KG)^{-1}\widetilde{V}^{-1}, \tag{9.31}$$

All above weights are stable TFMs with realizations of order $n + n_c$. It can be shown (see for example [ZDG96]) that with the above weights, the stability of the closed-loop system is guaranteed if $\left\| W_o \begin{bmatrix} U - U_r \\ V - V_r \end{bmatrix} \right\|_\infty < 1$ or $\|[\,\widetilde{U} - \widetilde{U}_r \;\; \widetilde{V} - \widetilde{V}_r\,]\widetilde{W}_i\|_\infty < 1$. These results justify the frequency-weighted coprime factor controller reduction methods introduced in [LAL90] for the reduction of state feedback and observer-based controllers. The case of arbitrary stabilizing controllers has been considered in [ZDG96]. Both cases are addressed in what follows. Note that in contrast to the approach of the previous subsection, the reduction of coprime factors can be performed even for completely unstable controllers.

RCF of a General Controller

We consider the efficient computation of the frequency-weighted controllability Gramian for the weights defined in (9.30). Let F_c be any matrix such that $A_c + B_c F_c$ is stable (i.e., the eigenvalues of $A_c + B_c F_c$ lie in the open left half plane for a continuous-time system or in the interior of the unit circle for a discrete-time system). Then, a RCF of $K = UV^{-1}$ is given by

$$\begin{bmatrix} \mathbf{U} \\ \mathbf{V} \end{bmatrix} = \left[\begin{array}{c|c} A_c + B_c F_c & B_c \\ \hline C_c + D_c F_c & D_c \\ F_c & I \end{array}\right] .$$

The output weighting $\mathbf{W_o}$ is a stable TFM having a state-space realization $\mathbf{W_o} = (A_o, *, C_o, *)$ of order $n + n_c$ [ZDG96, p.503], where

$$A_o = \begin{bmatrix} A_c - B_c R^{-1} D C_c & -B_c R^{-1} C \\ B\widetilde{R}^{-1} C_c & A - B D_c R^{-1} C \end{bmatrix},$$
$$C_o = \begin{bmatrix} R^{-1} D C_c - F_c & -R^{-1} C \end{bmatrix} .$$

The solution of the controller reduction problem for the special weights defined in (9.30) involves the solution of a Lyapunov equation of order n_c to compute the controllability Gramian P_E and the solution of a Lyapunov equation of order $n + 2n_c$ to determine the frequency-weighted observability Gramian Q_E. The following theorem [Var03b] shows that it is always possible to solve a Lyapunov equation of order $n + n_c$ to compute the frequency-weighted observability Gramian for the special weights in (9.30).

Theorem 9.4.3 *For a given n-th order system* $\mathbf{G} = (A, B, C, D)$ *assume that* $\mathbf{K} = (A_c, B_c, C_c, D_c)$ *is an* n_c*-th order stabilizing controller with* $I + DD_c$ *nonsingular. Then the frequency-weighted Gramians for Enns' method [Enn84] applied to the frequency-weighted right coprime factorization based controller reduction problem with weights defined in (9.30) can be computed by solving the corresponding Lyapunov equations of order at most* $n + n_c$ *as follows:* P_E *satisfies*

$$\begin{array}{l} (c)\ (A_c + B_c F_c) P_E + P_E (A_c + B_c F_c)^T + B_c B_c^T = 0 \\ (d)\ (A_c + B_c F_c) P_E (A_c + B_c F_c)^T + B_c B_c^T = P_E \end{array},$$

while Q_E *is the leading* $n_c \times n_c$ *diagonal block of* Q_o *satisfying*

$$(c)\ A_o^T Q_o + Q_o A_o + C_o^T C_o = 0\ , \quad (d)\ A_o^T Q_o A_o + C_o^T C_o = Q_o\ . \tag{9.32}$$

RCF of a State Feedback and Observer-Based Controller

In the case of a state feedback and full order observer-based controller (9.28), we obtain a significant reduction of computational costs. In this case, with $F_c = -(C + DF)$ we get (see [ZDG96])

$$\begin{bmatrix} \mathbf{U} \\ \mathbf{V} \end{bmatrix} = \left[\begin{array}{c|c} A + BF & -L \\ \hline F & 0 \\ C + DF & I \end{array}\right]$$

and the output weighting $\mathbf{W_o}$ has the following state-space realization of order n [ZDG96, p.503]

$$\mathbf{W_o} = \left[\begin{array}{c|cc} A+LC & -B-LD & L \\ \hline C & -D & I \end{array}\right]. \tag{9.33}$$

The following is a dual result to *Lemma 2* of [LAL90] to the case of nonzero feedthrough matrix D and covers also the discrete-time case.

Corollary 9.4.4 *For a given n-th order system* $\mathbf{G} = (A, B, C, D)$ *and the observer-based controller* $\mathbf{K}$ *(9.28), suppose* F *is a state feedback gain and* L *is a state estimator gain, such that* $A + BF$ *and* $A + LC$ *are stable. Then the frequency-weighted Gramians for Enns' method [Enn84] applied to frequency-weighted right coprime factorization based controller reduction problem with weights defined in (9.30) can be computed by solving the corresponding Lyapunov equations of order n, as follows:*

$$(c)\ \begin{array}{l}(A+BF)P_E + P_E(A+BF)^T + LL^T = 0 \\ (A+LC)^T Q_E + Q_E(A+LC) + C^T C = 0\end{array},$$

$$(d)\ \begin{array}{l}(A+BF)P_E(A+BF)^T + LL^T = P_E \\ (A+LC)^T Q_E(A+LC) + C^T C = Q_E\end{array}.$$

LCF of a General Controller

Let L_c be any matrix such that $A_c + L_c C_c$ is stable. Then, a LCF of $K = \widetilde{V}^{-1}\widetilde{U}$ is given by

$$[\,\widetilde{\mathbf{U}}\ \widetilde{\mathbf{V}}\,] = \left[\begin{array}{c|cc} A_c + L_c C_c & B_c + L_c D_c & L_c \\ \hline C_c & D_c & I \end{array}\right].$$

The input weight $\widetilde{W_i}$ is a stable TFM having a state-space realization $\widetilde{\mathbf{W_i}} := (A_i, B_i, *, *)$ of order $n + n_c$ [ZDG96, see p.503], where

$$A_i = \begin{bmatrix} A - B\widetilde{R}^{-1}D_c C & B\widetilde{R}^{-1}C_c \\ -B_c R^{-1} C & A_c - B_c D\widetilde{R}^{-1}C_c \end{bmatrix},\ B_i = \begin{bmatrix} -B\widetilde{R}^{-1} \\ B_c D\widetilde{R}^{-1} - L_c \end{bmatrix}, \tag{9.34}$$

with $R := I + DD_c$ and $\widetilde{R} = I + D_c D$.

We have a result similar to Theorem 9.4.3 showing that P_E can be efficiently determined by solving only a reduced order Lyapunov equation.

Theorem 9.4.5 *For a given n-th order system* $\mathbf{G} = (A, B, C, D)$ *assume that* $\mathbf{K} = (A_c, B_c, C_c, D_c)$ *is an* n_c*-th order stabilizing controller with* $I + DD_c$ *nonsingular. Then the frequency-weighted Gramians for Enns' method [Enn84] applied to the frequency-weighted left coprime factorization based controller reduction problem with weights defined in (9.31) can be computed by solving the corresponding Lyapunov equations of order at most* $n + n_c$ *as follows:* P_E *is the trailing* $n_c \times n_c$ *block of* P_i *satisfying*

$$(c)\ A_i P_i + P_i A_i^T + B_i B_i^T = 0, \quad (d)\ A_i P_i A_i^T + B_i B_i^T = P_i\ , \tag{9.35}$$

while Q_E satisfies

$$(c)\ (A_c + L_cC_c)^TQ_E + Q_E(A_c + L_cC_c) + C_c^TC_c = 0\ ,$$
$$(d)\ (A_c + L_cC_c)^TQ_E(A_c + L_cC_c) + C_c^TC_c \qquad = Q_E\ .$$

LCF of a State Feedback and Observer-Based Controller

Significant simplifications arise in the case of a state feedback and full order observer-based controller (9.28), where it is assumed that $A+BF$ and $A+LC$ are both stable. In this case (see [ZDG96]), with $L_c = -(B + LD)$ we get

$$[\,\widetilde{\mathbf{U}}\ \widetilde{\mathbf{V}}\,] = \left[\begin{array}{c|cc} A+LC & -L & -(B+LD) \\ \hline F & 0 & I \end{array}\right]$$

and the input weighting $\widetilde{\mathbf{W}}_\mathbf{i}$ has the following state-space realization of order n [ZDG96, p.503]

$$\widetilde{\mathbf{W}}_\mathbf{i} = \left[\begin{array}{c|c} A+BF & B \\ \hline C+DF & D \\ F & I \end{array}\right].$$

The following result is an extension of *Lemma 2* of [LAL90] to the case of a nonzero feedthrough matrix D and covers both the continuous- as well as the discrete-time case.

Corollary 9.4.6 *For a given n-th order system $\mathbf{G} = (A, B, C, D)$ and the observer-based controller $\mathbf{K}$ (9.28), suppose F is a state feedback gain and L is a state estimator gain, such that $A + BF$ and $A + LC$ are stable. Then the frequency-weighted Gramians for Enns' method [Enn84] applied to the frequency-weighted left coprime factorization based controller reduction problem with weights defined in (9.31) can be computed by solving the corresponding Lyapunov equations of order n as follows:*

$$(c)\ \begin{array}{l}(A+BF)P_E + P_E(A+BF)^T + BB^T = 0 \\ (A+LC)^TQ_E + Q_E(A+LC) + F^TF = 0\end{array}$$

$$(d)\ \begin{array}{l}(A+BF)P_E(A+BF)^T + BB^T = P_E \\ (A+LC)^TQ_E(A+LC) + F^TF = Q_E\end{array}$$

Square-Root Techniques

In the case of general right coprime factorized controllers, the method of Hammarling [Ham82] can be employed to solve (9.32) directly for the $(n + n_c) \times (n + n_c)$ Cholesky factor R_o of $Q_o = R_o^TR_o$. By partitioning R_o in the form

$$R_o = \begin{bmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{bmatrix},$$

with R_{11} an $n_c \times n_c$ matrix, the Cholesky factor R_E of the leading block of Q_o is $R_E = R_{11}$.

Similarly, in the case of general left coprime factorized controllers, (9.35) can be solved directly for the $(n + n_c) \times (n + n_c)$ Cholesky factor S_i of $P_i = S_i S_i^T$. By partitioning S_i in the form

$$S_i = \begin{bmatrix} S_{11} & S_{12} \\ 0 & S_{22} \end{bmatrix},$$

with S_{22} an $n_c \times n_c$ matrix, the Cholesky factor of the trailing block of P_i is $S_E = S_{22}$.

The Cholesky factors of Gramians for the remaining cases are directly obtained by solving the appropriate Lyapunov equations using Hammarling's algorithm [Ham82].

Efficiency Issues

In Table 9.3 we give for the RCF and LCF based approaches the number of operations $\widetilde{N}_E$ necessary to determine the Cholesky factors of the frequency-weighted Gramians and the achieved operation savings $\Delta_E = N_E - \widetilde{N}_E$, (see (9.18) for N_E) with respect to using standard FWMR techniques to reduce the coprime factors of the controller:

Table 9.3. Operation counts: general coprime factorized controller

Weight	$\widetilde{N}_E$	Δ_E
SRCF/SLCF	$33(n + n_c)^3 + 33n_c^3$	$24n^2 n_c + 74nn_c^2 + 58n_c^3$

To these figures we have to add the computational effort involved to compute a stabilizing state feedback (output injection) gain to determine the RCF (LCF) of the controller. When employing the Schur method of [Var81], it is possible to arrange the computations such that the resulting closed-loop state matrix $A_c + B_c F_c$ ($A_c + L_c C_c$) is in a RSF. In this way it is possible to avoid the reduction of this matrix to determine the unweighted Gramian P_E (Q_E) when solving the corresponding Lyapunov equation.

In the case of a state feedback and observer-based controller ($n_c = n$), the corresponding values are shown in Table 9.4.

Table 9.4. Operation counts: observer-based coprime factorized controller

Weight	$\widetilde{N}_E$	Δ_E
SRCF/SLCF	$66n^3$	$58n^3$

Observe the substantial computational effort savings obtained through structure exploitation for both general as well as state feedback controllers.

9.4.3 Performance Preserving Coprime Factors Reduction

In this subsection we consider the efficient computation of low order controllers by using the coprime factors reduction procedures to solve the frequency-weighted coprime factorization based $\mathcal{H}_\infty$ controller reduction problems formulated in [GG98]. Let

$$M = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \tag{9.36}$$

be the TFM used to parameterize all admissible γ-suboptimal controllers [ZDG96] in the form

$$K = M_{11} + M_{12}Q(I - M_{22}Q)^{-1}M_{21},$$

where Q is a stable and proper rational matrix satisfying $\|Q\|_\infty < \gamma$. Since for standard $\mathcal{H}_\infty$ problems both M_{12} and M_{21} are invertible and minimum-phase [ZDG96], a "natural" RCF of the central controller ($Q = 0$) as $K_0 = UV^{-1}$ can be obtained with

$$U = M_{11}M_{21}^{-1}, \quad V = M_{21}^{-1},$$

while a "natural" LCF of the central controller as $K_0 = \widetilde{V}^{-1}\widetilde{U}$ can be obtained with

$$\widetilde{U} = M_{12}^{-1}M_{11}, \quad \widetilde{V} = M_{12}^{-1}.$$

These factorizations can be used to perform unweighted coprime factor controller reduction using accuracy-enhanced model reduction algorithms [Var92].

A frequency-weighted right coprime factor reduction can be formulated with the one sided weights [ZDG96, GG98]

$$\text{PRCF:} \qquad W_o = \begin{bmatrix} \gamma^{-1}I & 0 \\ 0 & I \end{bmatrix} \Theta^{-1}, \quad W_i = I, \tag{9.37}$$

where

$$\Theta = \begin{bmatrix} \Theta_{11} & \Theta_{12} \\ \Theta_{21} & \Theta_{22} \end{bmatrix} := \begin{bmatrix} M_{12} - M_{11}M_{21}^{-1}M_{22} & M_{11}M_{21}^{-1} \\ -M_{21}^{-1}M_{22} & M_{21}^{-1} \end{bmatrix}.$$

With the help of the submatrices of Θ it is possible to express K also as

$$K = (\Theta_{12} + \Theta_{11}Q)(\Theta_{22} + \Theta_{21}Q)^{-1}$$

and thus the central controller is factorized as $K_0 = \Theta_{12}\Theta_{22}^{-1}$.

Similarly, a frequency-weighted left coprime factor reduction formulated in [GG98] is one sided with

$$\text{PLCF:} \qquad \widetilde{W}_o = I, \quad \widetilde{W}_i = \widetilde{\Theta}^{-1} \begin{bmatrix} \gamma^{-1} I & 0 \\ 0 & I \end{bmatrix}, \tag{9.38}$$

where

$$\widetilde{\Theta} = \begin{bmatrix} \widetilde{\Theta}_{11} & \widetilde{\Theta}_{12} \\ \widetilde{\Theta}_{21} & \widetilde{\Theta}_{22} \end{bmatrix} := \begin{bmatrix} M_{21} - M_{22} M_{12}^{-1} M_{11} & -M_{22} M_{12}^{-1} \\ M_{12}^{-1} M_{11} & M_{12}^{-1} \end{bmatrix}.$$

This time we have the alternative representation of K as

$$K = (\widetilde{\Theta}_{22} + Q\widetilde{\Theta}_{12})^{-1} (\widetilde{\Theta}_{21} + Q\widetilde{\Theta}_{11})$$

and the central controller is factorized as $K_0 = \widetilde{\Theta}_{22}^{-1} \widetilde{\Theta}_{21}$. Note that both Θ and $\widetilde{\Theta}$ are stable, invertible and minimum-phase.

The importance of the above frequency-weighted coprime factor reduction can be seen from the results of [GG98]. If K_0 is a stabilizing continuous-time γ-suboptimal $\mathcal{H}_\infty$ central controller, and K_r is an approximation of K_0 computed by applying the coprime factors reduction approach with the weight defined above, then K_r stabilizes the closed-loop system and preserves the γ-suboptimal performance, provided the weighted approximation error (9.2) or (9.3) is less than $1/\sqrt{2}$. We conjecture that this result holds also in the discrete-time case, and can be proved along the lines of the proof provided in [ZDG96].

RCF Controller Reduction

We consider the efficient computation of the frequency-weighted controllability Gramian for the weights defined in (9.37). Let us consider a realization of the parameterization TFM M (9.36) in the form

$$\mathbf{M} = \left[\begin{array}{c|cc} \widehat{A} & \widehat{B}_1 & \widehat{B}_2 \\ \hline \widehat{C}_1 & \widehat{D}_{11} & \widehat{D}_{12} \\ \widehat{C}_2 & \widehat{D}_{21} & \widehat{D}_{22} \end{array}\right].$$

Note that for the central controller we have $(A_c, B_c, C_c, D_c) = (\widehat{A}, \widehat{B}_1, \widehat{C}_1, \widehat{D}_{11})$. Since M_{12} and M_{21} are stable, minimum-phase and invertible TFMs, it follows that $\widehat{D}_{12}$ and $\widehat{D}_{21}$ are invertible, $\widehat{A}$, $\widehat{A} - \widehat{B}_2 \widehat{D}_{12}^{-1} \widehat{C}_1$ and $\widehat{A} - \widehat{B}_1 \widehat{D}_{21}^{-1} \widehat{C}_2$ are all stable matrices, i.e., have eigenvalues in the open left half plane for a continuous-time controller and in the interior of the unit circle for a discrete-time controller.

The realizations of $\mathbf{\Theta}$ and $\mathbf{\Theta}^{-1}$ can be computed as [ZDG96]

$$\mathbf{\Theta} = \left[\begin{array}{c|c} A_\Theta & B_\Theta \\ \hline C_\Theta & D_\Theta \end{array}\right] = \left[\begin{array}{c|cc} \widehat{A} - \widehat{B}_1 \widehat{D}_{21}^{-1} \widehat{C}_2 & \widehat{B}_2 - \widehat{B}_1 \widehat{D}_{21}^{-1} \widehat{D}_{22} & \widehat{B}_1 \widehat{D}_{21}^{-1} \\ \hline \widehat{C}_1 - \widehat{D}_{11} \widehat{D}_{21}^{-1} \widehat{C}_2 & \widehat{D}_{12} - \widehat{D}_{11} \widehat{D}_{21}^{-1} \widehat{D}_{22} & \widehat{D}_{11} \widehat{D}_{21}^{-1} \\ -\widehat{D}_{21}^{-1} \widehat{C}_2 & -\widehat{D}_{21}^{-1} \widehat{D}_{22} & \widehat{D}_{21}^{-1} \end{array}\right],$$

$$\mathbf{\Theta^{-1}} = \left[\begin{array}{c|c} A_{\Theta^{-1}} & B_{\Theta^{-1}} \\ \hline C_{\Theta^{-1}} & D_{\Theta^{-1}} \end{array}\right] = \left[\begin{array}{c|cc} \widehat{A} - \widehat{B}_2\widehat{D}_{12}^{-1}\widehat{C}_1 & \widehat{B}_2\widehat{D}_{12}^{-1} & \widehat{B}_1 - \widehat{B}_2\widehat{D}_{12}^{-1}\widehat{D}_{11} \\ \hline -\widehat{D}_{12}^{-1}\widehat{C}_1 & \widehat{D}_{12}^{-1} & -\widehat{D}_{12}^{-1}\widehat{D}_{11} \\ \widehat{C}_2 - \widehat{D}_{22}\widehat{D}_{12}^{-1}\widehat{C}_1 & \widehat{D}_{22}\widehat{D}_{12}^{-1} & \widehat{D}_{21} - \widehat{D}_{22}\widehat{D}_{12}^{-1}\widehat{D}_{11} \end{array}\right].$$

Since the realization of $\mathbf{W_o}\begin{bmatrix}\mathbf{U}\\ \mathbf{V}\end{bmatrix}$ has apparently order $2n_c$, it follows that the solution of the controller reduction problem for the special weights defined in (9.37) involves the solution of a Lyapunov equation of order n_c to determine the frequency-weighted controllability Gramian P_E and a Lyapunov equation of order $2n_c$ to compute the observability Gramian Q_E. The following result [Var03a] shows that it is always possible to solve two Lyapunov equations of order n_c to compute the frequency-weighted Gramians for the special weights in (9.37).

Theorem 9.4.7 *The controllability Gramian P_E and the frequency-weighted observability Gramian Q_E according to Enns' choice [Enn84] for the frequency-weighted RCF controller reduction problem with weights (9.37) satisfy, according to the system type, the corresponding Lyapunov equations*

$$(c)\left\{\begin{array}{rcl} A_\Theta P_E + P_E A_\Theta^T + \widetilde{B}_\Theta \widetilde{B}_\Theta^T & = & 0 \\ A_{\Theta^{-1}}^T Q_E + Q_E A_{\Theta^{-1}} + \widetilde{C}_{\Theta^{-1}}^T \widetilde{C}_{\Theta^{-1}} & = & 0 \end{array}\right. ,$$

$$(d)\left\{\begin{array}{rcl} A_\Theta P_E A_\Theta^T + \widetilde{B}_\Theta \widetilde{B}_\Theta^T & = & P_E \\ A_{\Theta^{-1}}^T Q_E A_{\Theta^{-1}} + \widetilde{C}_{\Theta^{-1}}^T \widetilde{C}_{\Theta^{-1}} & = & Q_E \end{array}\right. ,$$

where $\widetilde{B}_\Theta = B_\Theta \begin{bmatrix}0\\ I\end{bmatrix} = \widehat{B}_1\widehat{D}_{21}^{-1}$ *and* $C_{\Theta^{-1}} = \mathrm{diag}\,(\gamma^{-1}I, I)C_{\Theta^{-1}}$.

LCF Controller Reduction

We consider now the efficient computation of the frequency-weighted controllability Gramian for the weights defined in (9.38). The realizations of $\widetilde{\mathbf{\Theta}}$ and $\widetilde{\mathbf{\Theta}}^{-1}$ can be computed as [ZDG96]

$$\widetilde{\mathbf{\Theta}} = \left[\begin{array}{c|c} A_{\widetilde{\Theta}} & B_{\widetilde{\Theta}} \\ \hline C_{\widetilde{\Theta}} & D_{\widetilde{\Theta}} \end{array}\right] = \left[\begin{array}{c|cc} \widehat{A} - \widehat{B}_2\widehat{D}_{12}^{-1}\widehat{C}_1 & \widehat{B}_1 - \widehat{B}_2\widehat{D}_{12}^{-1}\widehat{D}_{11} & -\widehat{B}_2\widehat{D}_{12}^{-1} \\ \widehat{C}_2 - \widehat{D}_{22}\widehat{D}_{12}^{-1}\widehat{C}_1 & \widehat{D}_{21} - \widehat{D}_{22}\widehat{D}_{12}^{-1}\widehat{D}_{11} & -\widehat{D}_{22}\widehat{D}_{12}^{-1} \\ \widehat{D}_{12}^{-1}\widehat{C}_1 & \widehat{D}_{12}^{-1}\widehat{D}_{11} & \widehat{D}_{12}^{-1} \end{array}\right],$$

$$\widetilde{\mathbf{\Theta}}^{-1} = \left[\begin{array}{c|c} A_{\widetilde{\Theta}^{-1}} & B_{\widetilde{\Theta}^{-1}} \\ \hline C_{\widetilde{\Theta}^{-1}} & D_{\widetilde{\Theta}^{-1}} \end{array}\right] = \left[\begin{array}{c|cc} \widehat{A} - \widehat{B}_1\widehat{D}_{21}^{-1}\widehat{C}_2 & -\widehat{B}_1\widehat{D}_{21}^{-1} & \widehat{B}_2 - \widehat{B}_1\widehat{D}_{21}^{-1}\widehat{D}_{22} \\ \hline \widehat{D}_{21}^{-1}\widehat{C}_2 & \widehat{D}_{21}^{-1} & \widehat{D}_{21}^{-1}\widehat{D}_{22} \\ \widehat{C}_1 - \widehat{D}_{11}\widehat{D}_{21}^{-1}\widehat{C}_2 & -\widehat{D}_{11}\widehat{D}_{21}^{-1} & \widehat{D}_{12} - \widehat{D}_{11}\widehat{D}_{21}^{-1}\widehat{D}_{22} \end{array}\right].$$

Since the realization of $[\,\widetilde{\mathbf{U}}\ \ \widetilde{\mathbf{V}}\,]\widetilde{\mathbf{W}}_\mathbf{i}$ has apparently order $2n_c$, it follows that the solution of the controller reduction problem for the special weights defined in (9.38) involves the solution of a Lyapunov equation of order $2n_c$

to determine the frequency-weighted controllability Gramian P_E and a Lyapunov equation of order n_c to compute the observability Gramian Q_E. The following result [Var03a] shows that it is always possible to solve two Lyapunov equations of order n_c to compute the frequency-weighted Gramians for the special weights in (9.38).

Theorem 9.4.8 *The frequency-weighted controllability Gramian P_E and observability Gramian Q_E according to Enns' choice [Enn84] for the frequency-weighted LCF controller reduction problem with weights (9.38) satisfy the corresponding Lyapunov equations*

$$(c)\begin{cases} A_{\widetilde{\Theta}^{-1}}P_E + P_E A_{\widetilde{\Theta}^{-1}}^T + \widetilde{B}_{\widetilde{\Theta}^{-1}}\widetilde{B}_{\widetilde{\Theta}^{-1}}^T &= 0 \\ A_{\widetilde{\Theta}}^T Q_E + Q_E A_{\widetilde{\Theta}} + \widetilde{C}_{\widetilde{\Theta}}^T \widetilde{C}_{\widetilde{\Theta}} &= 0 \end{cases},$$

$$(d)\begin{cases} A_{\widetilde{\Theta}^{-1}}P_E A_{\widetilde{\Theta}^{-1}} + \widetilde{B}_{\widetilde{\Theta}^{-1}}\widetilde{B}_{\widetilde{\Theta}^{-1}}^T &= P_E \\ A_{\widetilde{\Theta}}^T Q_E A_{\widetilde{\Theta}} + \widetilde{C}_{\widetilde{\Theta}}^T \widetilde{C}_{\widetilde{\Theta}} &= Q_E \end{cases},$$

where $\widetilde{B}_{\widetilde{\Theta}^{-1}} = B_{\widetilde{\Theta}^{-1}}\mathrm{diag}\,(\gamma^{-1}I, I)$ *and* $\widetilde{C}_{\widetilde{\Theta}} = \widehat{D}_{12}^{-1}\widehat{C}_1$.

Efficiency Issues

In Table 9.5 we give for the RCF and LCF based approaches the number of operations $\widetilde{N}_E$ necessary to determine the Cholesky factors of the frequency-weighted Gramians and the achieved operation savings $\Delta_E = N_E - \widetilde{N}_E$, (see (9.18) for N_E) with respect to using standard FWMR techniques to reduce the coprime factors of the controller.

Table 9.5. Operation counts: coprime factorized $\mathcal{H}_\infty$-controller

Weight	$\widetilde{N}_E$	Δ_E
PRCF/PLCF	$66n_c^3$	$58n_c^3$

Observe the substantial (47%) computational effort savings obtained through structure exploitation.

9.4.4 Relative Error Coprime Factors Reduction

An alternative approach to $\mathcal{H}_\infty$ controller reduction uses the relative error method as suggested in [Zho95]. Using this approach in conjunction with the RCF reduction we can define the weights as

$$W_o = I, \quad W_i = \begin{bmatrix} U \\ V \end{bmatrix}^+, \tag{9.39}$$

where $\begin{bmatrix} U \\ V \end{bmatrix}^+$ denotes a stable left inverse of $\begin{bmatrix} U \\ V \end{bmatrix}$. A variant of this approach (see [ZDG96]) is to perform a relative error coprime factor reduction on an invertible augmented minimum-phase system $\begin{bmatrix} U_a \\ V_a \end{bmatrix}$ instead of $\begin{bmatrix} U \\ V \end{bmatrix}$. In our case, Θ can be taken as the augmented system. Thus this method essentially consists of determining an approximation Θ_r of Θ by solving the relative error minimization problems

$$\|(\Theta - \Theta_r)\Theta^{-1}\|_\infty = \min \tag{9.40}$$

or

$$\|\Theta^{-1}(\Theta - \Theta_r)\|_\infty = \min . \tag{9.41}$$

These are a frequency-weighted problems with the corresponding weights

$$\text{RCFR1:} \qquad W_o = I, \quad W_i = \Theta^{-1} \tag{9.42}$$

and respectively

$$\text{RCFR2:} \qquad W_o = \Theta^{-1}, \quad W_i = I . \tag{9.43}$$

The reduced controller is recovered from the sub-blocks (1,2) and (2,2) of Θ_r as $K_r = \Theta_{r,12}\Theta_{r,22}^{-1}$. This method has been also considered in [EJL01] for the case of normalized coprime factor $\mathcal{H}_\infty$ controller reduction.

In the same way, a relative error LCF reduction can be formulated with the weights

$$\widetilde{W}_o = [\,\widetilde{U}\ \widetilde{V}\,]^+, \quad \widetilde{W}_i = I \tag{9.44}$$

where $[\,\widetilde{U}\ \widetilde{V}\,]^+$ denotes a stable right inverse of $[\,\widetilde{U}\ \widetilde{V}\,]$. Alternatively, an augmented relative error problem can be solved by approximating $\widetilde{\Theta}$ by a reduced order system $\widetilde{\Theta}_r$ by solving the relative error norm minimization problems

$$\|\widetilde{\Theta}^{-1}(\widetilde{\Theta} - \widetilde{\Theta}_r)\|_\infty \tag{9.45}$$

or

$$\|(\widetilde{\Theta} - \widetilde{\Theta}_r)\widetilde{\Theta}^{-1}\|_\infty . \tag{9.46}$$

These are frequency-weighted problems with weights

$$\text{LCFR1:} \qquad \widetilde{W}_o = \widetilde{\Theta}^{-1}, \quad \widetilde{W}_i = I \tag{9.47}$$

and respectively

$$\text{LCFR2:} \qquad \widetilde{W}_o = I, \quad \widetilde{W}_i = \widetilde{\Theta}^{-1} . \tag{9.48}$$

The reduced controller is recovered from the sub-blocks (2,1) and (2,2) of $\widetilde{\Theta}_r$ as $K_r = \widetilde{\Theta}_{r,22}^{-1}\widetilde{\Theta}_{r,21}$.

Relative Error RCF Reduction

For the solution of the relative error approximation problems (9.40) and (9.41) we have the following straightforward results [ZDG96, Theorem 7.5]:

Theorem 9.4.9 *The frequency-weighted controllability Gramian P_E and observability Gramian Q_E for Enns' method [Enn84] applied to the frequency-weighted approximation problems (9.40) and (9.41) satisfy, depending on the system type, the corresponding Lyapunov equations, as follows:*

1. *For the problem (9.40)*

$$(c)\begin{cases} A_{\Theta^{-1}}P_E + P_E A_{\Theta^{-1}}^T + B_{\Theta^{-1}}B_{\Theta^{-1}}^T &= 0 \\ A_{\Theta}^T Q_E + Q_E A_{\Theta} + C_{\Theta}^T C_{\Theta} &= 0 \end{cases},$$
$$(d)\begin{cases} A_{\Theta^{-1}}P_E A_{\Theta^{-1}} + B_{\Theta^{-1}}B_{\Theta^{-1}}^T &= P_E \\ A_{\Theta}^T Q_E A_{\Theta} + C_{\Theta}^T C_{\Theta} &= Q_E \end{cases}.$$

2. *For the problem (9.41)*

$$(c)\begin{cases} A_{\Theta}P_E + P_E A_{\Theta}^T + B_{\Theta}B_{\Theta}^T &= 0 \\ A_{\Theta^{-1}}^T Q_E + Q_E A_{\Theta^{-1}} + C_{\Theta^{-1}}^T C_{\Theta^{-1}} &= 0 \end{cases},$$
$$(d)\begin{cases} A_{\Theta}P_E A_{\Theta}^T + B_{\Theta}B_{\Theta}^T &= P_E \\ A_{\Theta^{-1}}^T Q_E A_{\Theta^{-1}} + C_{\Theta^{-1}}^T C_{\Theta^{-1}} &= Q_E \end{cases}.$$

Relative Error LCF Reduction

For the solution of the relative error approximation problems (9.45) and (9.46) we have the following straightforward results [ZDG96, Theorem 7.5]:

Theorem 9.4.10 *The frequency-weighted controllability Gramian P_E and observability Gramian Q_E for Enns' method [Enn84] applied to the frequency-weighted approximation problem (9.45) and (9.46) satisfy, according to the system type, the corresponding Lyapunov equations, as follows:*

1. *For the problem (9.45)*

$$(c)\begin{cases} A_{\widetilde{\Theta}}P_E + P_E A_{\widetilde{\Theta}}^T + B_{\widetilde{\Theta}}B_{\widetilde{\Theta}}^T &= 0 \\ A_{\widetilde{\Theta}^{-1}}^T Q_E + Q_E A_{\widetilde{\Theta}^{-1}} + C_{\widetilde{\Theta}^{-1}}^T C_{\widetilde{\Theta}^{-1}} &= 0 \end{cases},$$
$$(d)\begin{cases} A_{\widetilde{\Theta}}P_E A_{\widetilde{\Theta}} + B_{\widetilde{\Theta}}B_{\widetilde{\Theta}}^T &= P_E \\ A_{\widetilde{\Theta}^{-1}}^T Q_E A_{\widetilde{\Theta}^{-1}} + C_{\widetilde{\Theta}^{-1}}^T C_{\widetilde{\Theta}^{-1}} &= Q_E \end{cases}.$$

2. *For the problem (9.46)*

$$(c)\begin{cases} A_{\widetilde{\Theta}^{-1}}P_E + P_E A_{\widetilde{\Theta}^{-1}}^T + B_{\widetilde{\Theta}^{-1}}B_{\widetilde{\Theta}^{-1}}^T &= 0 \\ A_{\widetilde{\Theta}}^T Q_E + Q_E A_{\widetilde{\Theta}} + C_{\widetilde{\Theta}}^T C_{\widetilde{\Theta}} &= 0 \end{cases},$$
$$(d)\begin{cases} A_{\widetilde{\Theta}^{-1}}P_E A_{\widetilde{\Theta}^{-1}}^T + B_{\widetilde{\Theta}^{-1}}B_{\widetilde{\Theta}^{-1}}^T &= P_E \\ A_{\widetilde{\Theta}}^T Q_E A_{\widetilde{\Theta}} + C_{\widetilde{\Theta}}^T C_{\widetilde{\Theta}} &= Q_E \end{cases}.$$

Efficiency Issues

In Table 9.6 we give for the RCF and LCF based approaches the number of operations $\widetilde{N}_E$ necessary to determine the Cholesky factors of the frequency-weighted Gramians and the achieved operation savings $\Delta_E = N_E - \widetilde{N}_E$, with respect to using standard FWMR techniques to reduce the coprime factors of the controller.

Table 9.6. Operation counts: coprime factorized $\mathcal{H}_\infty$-controller parametrizations

Weight	$\widetilde{N}_E$	Δ_E
RCFR1/RCFR2	$66n_c^3$	$58n_c^3$
LCFR1/LCFR2	$66n_c^3$	$58n_c^3$

Observe the substantial (47%) computational effort savings obtained through structure exploitation.

9.5 Software for Controller Reduction

In this section we present an overview of available software tools to support controller reduction. We focus on tools developed within the NICONET project. For details about other tools see Chapter 7 of [Var01a].

9.5.1 Tools for Controller Reduction in SLICOT

A powerful collection of Fortran 77 subroutines for model and controller reduction has been implemented within the NICONET project [Var01a, Var02b] as part of the SLICOT library. The model and controller reduction software in SLICOT implements the latest algorithmic developments for the following approaches:

- absolute error model reduction using the balanced truncation [Moo81], singular perturbation approximation [LA89], and Hankel-norm approximation [Glo84] methods;
- relative error model reduction using the balanced stochastic truncation approach [DP84, SC88, VF93];
- frequency-weighted balancing related model reduction methods [Enn84, LC92, WSL99, VA01, VA03] and frequency-weighted Hankel-norm approximation methods [LA85, HG86, Var01b];
- controller reduction methods using frequency-weighted balancing related methods [LAL90, VA02, VA03] and unweighed and frequency-weighted coprime factorization based techniques [LAL90].

The model and controller reduction routines in SLICOT are among the most powerful and numerically most reliable software tools available for model and controller reduction. All routines can be employed to reduce both stable and unstable, continuous- or discrete-time models or controllers. The underlying numerical algorithms rely on *square-root* (**SR**) [TP87] and *balancing-free square-root* (**BFSR**) [Var91b] accuracy enhancing techniques. The Table 9.7 contains the list of the user callable subroutines available for controller reduction in SLICOT.

Table 9.7. User callable SLICOT controller reduction routines

Name	Function
SB16AD	FWBT/FWSPA-based controller reduction for closed-loop stability and performance preserving weights
SB16BD	state feedback/observer-based controller reduction using coprime factorization in conjunction with FWBT and FWSPA techniques
SB16CD	state feedback/observer-based controller reduction using frequency-weighted coprime factorization in conjunction with FWBT technique

In implementing these routines, a special attention has been paid to ensure their numerical robustness. All implemented routines rely on the **SR** and **BFSR** accuracy enhancing techniques [TP87, Var91b, Var91a]. Both techniques substantially contribute to improve the numerical reliability of computations. Furthermore, all routines optionally perform the scaling of the original system. When calling each routine, the order of the reduced controller can be selected by the user or can be determined automatically on the basis of computed quantities which can be assimilated to the usual Hankel singular values. Each of routines can handle both continuous- and discrete-time controllers. In what follows we shortly discuss some particular functionality provided by these user callable routines.

The FWCR routine SB16AD is a specialization of a general purpose FWMR routine, for the special one-sided weights (9.19) and (9.20) used to enforce closed-loop stability as well as two-sided weights (9.21) for performance preservation. This routine works on a general stabilizing controller. Unstable controllers are handled by separating their stable and unstable parts and applying the controller reduction only to the stable parts. This routine has a large flexibility in combining different choices of the Gramians (see subsection 9.3.1) and can handle the unweighted case as well.

The coprime factorization based controller reduction routines SB16BD and SB16CD are specially adapted to reduce state feedback and observer-based controllers. The routine SB16BD allows arbitrary combinations of BT and SPA methods with "natural" left and right coprime factorizations of the controller. The routine SB16CD, implementing the frequency-weighted coprime factorization based stability preserving approach, can be employed only in

conjunction with the BT technique. This routine allows to work with both left and right coprime factorization based approaches.

In implementing the new controller reduction software, a special emphasis has been put on an appropriate modularization of the routines by isolating some basic computational tasks and implementing them in supporting computational routines. For example, the balancing related approach (implemented in SB16AD) and the frequency-weighted coprime factorization based controller reduction method (implemented in SB16CD), share a common two step computational scheme: (1) compute two non-negative definite matrices called generically "frequency-weighted Gramians"; (2) determine suitable truncation matrices and apply them to obtain the matrices of the reduced model/controller using the BT or SPA methods. For the first step, separate routines have been implemented to compute appropriate Gramians according to the specifics of each method. To employ the accuracy enhancing **SR** or **BFSR** techniques, these routines compute in fact, instead of Gramians, their Cholesky factors. For the second step, a unique routine has been implemented, which is called by both above routines. For a detailed description of the controller reduction related software available in SLICOT see [Var02a].

9.5.2 SLICOT Based User-Friendly Tools

One of the main objectives of the NICONET project was to provide, additionally to standardized Fortran codes, high quality software embedded into user-friendly environments for *computer aided control system design*. The popular computational environment MATLAB[1] allows to easily add external functions implemented in general purpose programming languages like C or Fortran. The external functions are called *mex*-functions and have to be programmed according to precise programming standards. Two *mex*-functions have been implemented as main MATLAB interfaces to the controller reduction routines available in SLICOT. To provide a convenient interface to work with control objects defined in the MATLAB Control Toolbox, easy-to-use higher level controller reduction *m*-functions have been additionally implemented. The list of available *mex*- and *m*-functions is given in Table 9.8.

Table 9.8. *mex*- and *m*-functions for controller reduction

Name	Function
mex: `conred` *m*: `fwbconred`	frequency-weighted balancing related controller reduction (based on SB16AD)
mex: `sfored` *m*: `sfconred`	coprime factorization based reduction of state feedback controllers (based on SB16BD and SB16CD)

[1] MATLAB is a registered trademark of The MathWorks, Inc.

All these functions are able to reduce both continuous- and discrete-time, stable as well as unstable controllers. The functions can be used for unweighted reduction as well, without any significant computational overhead.

In the implementation of the *mex*- and *m*-functions, one main goal was to allow the access to the complete functionality provided by the underlying Fortran routines. To manage the multitude of possible user options, a so-called SYSRED structure has been defined. The controller reduction relevant fields which can be set in the SYSRED structure are shown below:

```
      BalredMethod: [ {bta} | spa ]
AccuracyEnhancing: [ {bfsr} | sr ]
            Tolred: [ positive scalar {0} ]
        TolMinreal: [ positive scalar {0} ]
             Order: [ integer {-1} ]
  FWEContrGramian: [ {standard} | enhanced ]
 FWEObservGramian: [ {standard} | enhanced ]
    CoprimeFactor: [ left | {right} ]
     OutputWeight: [ {stab} | perf | none]
      InputWeight: [ {stab} | none]
   CFConredMethod: [ {fwe}  | nofwe ]
  FWEConredMethod: [ none | outputstab | inputstab | {performance} ]
```

This structure is created and managed via special functions. For more details on this structure see [Var02a].

Functionally equivalent user-friendly tools can be also implemented in the MATLAB-like environment `Scilab` [Gom99]. In `Scilab`, external functions can be similarly implemented as in MATLAB and only several minor modifications are necessary to the MATLAB *mex*-functions to adapt them to `Scilab`.

9.6 Controller Reduction Example

We consider the standard $\mathcal{H}_\infty$ optimization setup for the four-disk control system [ZDG96] described by

$$\begin{aligned} \dot{x} &= Ax + b_1 w + b_2 u \\ z &= \begin{bmatrix} 10^{-3}h \\ 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \\ y &= c_2 x + [\,0\ 1\,] w \end{aligned}$$

where u and w are the control and disturbance inputs, respectively, z and y are the performance and measurement outputs, respectively, and $x \in \mathbb{R}^7$ is the state vector. For completeness, we give the matrices of the model

$$A = \begin{bmatrix} -0.161 & -6.004 & -0.58215 & -9.9835 & -0.40727 & -3.982 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \quad b_2 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$b_1 = \begin{bmatrix} b_2 & 0 \end{bmatrix}, \quad h = \begin{bmatrix} 0 & 0 & 0 & 0 & 0.55 & 11 & 1.32 & 18 \end{bmatrix}$$

$$c_2 = \begin{bmatrix} 0 & 0 & 0.00064432 & 0.0023196 & 0.071252 & 1.0002 & 0.10455 & 0.99551 \end{bmatrix}$$

Using the `hinf` function of the Robust Control Toolbox [CS02], we computed the $\mathcal{H}_\infty$ controller $K(s)$ and the controller parameterization $M(s)$ using the loop-shifting formulae of [SLC89]. The optimal $\mathcal{H}_\infty$-norm of the TFM T_{zw} from the disturbance input w to performance output z is $\gamma_{opt} = 1.1272$. We employed the same value $\gamma = 1.2$ as in [ZDG96] to determine an 8th order γ-suboptimal controller and the corresponding parameterization. The resulting controller is itself stable and has been reduced to orders between 0 and 7 using the methods presented in this paper. Provided the corresponding closed-loop system was stable, we computed for each reduced order controller the value of the $\mathcal{H}_\infty$-norm of the TFM T_{zw}. The results are presented in the Table 9.9, where U signifies that the closed-loop system with the resulting reduced order controller is unstable.

For each controller order, the bolded numbers indicate the best achieved approximation of the closed-loop TFM T_{zw} in terms of the corresponding $\mathcal{H}_\infty$-norms. Observe that the FWSPA approach is occasionally superior for this example to the FWBT method. Several methods were able to obtain very good approximations until orders as low as 4. Even the best second order approximation appears to be still satisfactory. Interestingly, this controller provides a better approximation of the closed-loop TFM than the best third order controller. None of the employed methods was able to produce a stabilizing first order controller, although such a controller apparently exists (see results reported for the frequency-weighted HNA in [ZDG96]). As a curiosity, the standard unweighted SPA provided a stabilizing constant output feedback gain controller albeit this exhibits a very poor closed-loop performance.

9.7 Conclusions

We discussed recent enhancements of several frequency-weighted balancing related controller reduction methods. These enhancements are in three main directions: (1) enhancing the capabilities of underlying approximation methods by employing new choices of Gramians guaranteeing stability for two-sided weights or by employing alternatively the SPA approach instead of traditionally employed BT method; (2) improving the accuracy of computations by

Table 9.9. $\mathcal{H}_\infty$-norm of the closed-loop TFM T_{zw}

Order of K_r	7	6	5	4	3	2	1	0
UW (BT)	U	1.318	U	U	U	U	U	U
UW (SPA)	1.200	1.200	U	U	U	U	U	**6490.9**
RCF (BT)	1.198	**1.196**	1.198	**1.196**	385.99	494.1	U	U
RCF (SPA)	1.196	**1.196**	U	**1.196**	U	34.99	U	**6490.9**
LCF (BT)	2.061	1.260	33.810	5.197	U	U	U	U
LCF (SPA)	1.196	**1.196**	1.588	2.045	U	U	U	**6490.9**
SW1 (BT)	1.321	1.199	2.287	1.591	23.381	U	U	U
SW1 (SPA)	1.196	**1.196**	**1.196**	1.484	3.218	U	U	**6490.9**
SRCF (BT)	1.232	1.197	1.254	1.202	13.514	**1.413**	U	U
SRCF (SPA)	1.196	**1.196**	16.274	**1.196**	U	U	U	**6490.9**
SLCF (BT)	1.418	1.216	37.647	3.062	U	U	U	U
SLCF (SPA)	1.196	**1.196**	1.197	1.799	15.151	U	U	**6490.9**
PRCF (BT)	1.199	**1.196**	1.207	**1.196**	2.760	1.734	U	U
PRCF (SPA)	1.196	**1.196**	1.542	**1.196**	U	U	U	**6490.9**
PLCF (BT)	1.196	**1.196**	U	1.197	U	U	U	U
PLCF (SPA)	1.196	**1.196**	**1.196**	**1.196**	7.609	U	U	**6490.9**
PW (BT)	1.334	1.198	U	1.212	U	U	U	U
PW (SPA)	1.196	**1.196**	**1.196**	**1.196**	3.465	U	U	**6490.9**
RCFR1 (BT)	U	1.197	U	**4.1233**	U	U	U	U
RCFR1 (SPA)	**1.195**	**1.196**	U	U	U	U	U	**6490.9**
LCFR1 (BT)	U	1.197	U	**4.1233**	U	U	U	U
LCFR1 (SPA)	**1.195**	**1.196**	U	U	U	U	U	**6490.9**
RCFR2 (BT)	**1.195**	**1.196**	1.199	**1.196**	**2.758**	1.6811	U	U
RCFR2 (SPA)	1.196	**1.196**	U	**1.196**	U	U	U	**6490.9**
LCFR2 (BT)	U	1.197	U	**4.1233**	U	U	U	U
LCFR2 (SPA)	**1.195**	**1.196**	U	U	U	U	U	**6490.9**

extending the **SR** and **BFSR** accuracy enhancing techniques to frequency-weighted balancing; and (3) improving the computational efficiency of several balancing related controller reduction approaches by fully exploiting the underlying problem structure when computing frequency-weighted Gramians. To ease the implementation of these approaches, we provide complete directly implementable formulas for frequency-weighted Gramian computations.

As can be seen clearly from Table 9.9, none of existing methods seems to be universally applicable and their performances are very hard to predict. However, having several alternative approaches at our disposal certainly increases the chance of obtaining acceptable low order controller approximations. For several approaches, ready to use controller reduction software is freely available in the Fortran 77 library SLICOT, together with user friendly interfaces to the computational environments MATLAB and `Scilab`. For the rest of methods described in this paper, similar software can be easily implemented using standard computational tools provided in SLICOT.

References

[ABB99] E. Anderson, Z. Bai, J. Bishop, J. Demmel, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov, and D. Sorensen. *LAPACK User's Guide, Third Edition*. SIAM, Philadelphia, 1999.

[AL89] B. D. O. Anderson and Y. Liu. Controller reduction: concepts and approaches. *IEEE Trans. Automat. Control*, 34:802–812, 1989.

[BMSV99] P. Benner, V. Mehrmann, V. Sima, S. Van Huffel, and A. Varga. SLICOT – a subroutine library in systems and control theory. In B. N. Datta (Ed.), *Applied and Computational Control, Signals and Circuits*, vol. 1, pp. 499–539, Birkhäuser, 1999.

[CS02] R. Y. Chiang and M. G. Safonov. *Robust Control Toolbox Version 2.0.9 (R13)*. The MathWorks Inc., Natick, MA, 2002.

[DP84] U. B. Desai and D. Pal. A transformation approach to stochastic model reduction. *IEEE Trans. Automat. Control*, 29:1097–1100, 1984.

[EJL01] H. M. H. El-Zobaidi, I. M. Jaimoukha, and D. J. N. Limebeer. Normalized $\mathcal{H}_\infty$ controller reduction with a priori error bounds. *IEEE Trans. Automat. Control*, 46:1477–1483, 2001.

[Enn84] D. Enns. *Model Reduction for Control Systems Design*. PhD thesis, Dept. Aeronaut. Astronaut., Stanford Univ., Stanford, CA, 1984.

[GG98] P. J. Goddard and K. Glover. Controller approximation: approaches for preserving $\mathcal{H}_\infty$ performance. *IEEE Trans. Automat. Control*, 43:858–871, 1998.

[GGMS74] P. E. Gill, G. H. Golub, W. Murray, and M. A. Saunders. Methods for modifying matrix factorizations. *Math. Comput.*, 28:505–535, 1974.

[Glo84] K. Glover. All optimal Hankel-norm approximations of linear multivariable systems and their L^∞-error bounds. *Int. J. Control*, 39:1115–1193, 1984.

[Gom99] C. Gomez (Ed.) *Engineering and Scientific Computing with Scilab*. Birkhauser, Boston, 1999.

[GSV00] G. H. Golub, K. Solna, and P. Van Dooren. Computing the SVD of a general matrix product/quotient. *SIAM J. Matrix Anal. Appl.*, 22:1–19, 2000.

[Gu95] G. Gu. Model reduction with relative/multiplicative error bounds and relations to controller reduction. *IEEE Trans. Automat. Control*, 40:1478–1485, 1995.

[GV89] G. H. Golub and C. F. Van Loan. *Matrix Computations*. John Hopkins University Press, Baltimore, 1989.

[Ham82] S. J. Hammarling. Numerical solution of the stable, non-negative definite Lyapunov equation. *IMA J. Numer. Anal.*, 2:303–323, 1982.

[HG86] Y. S. Hung and K. Glover. Optimal Hankel-norm approximation of stable systems with first-order stable weighting functions. *Systems & Control Lett.*, 7:165–172, 1986.

[LA85] G. A. Latham and B. D. O. Anderson. Frequency-weighted optimal Hankel norm approximation of stable transfer functions. *Systems & Control Lett.*, 5:229–236, 1985.

[LA89] Y. Liu and B. D. O. Anderson. Singular perturbation approximation of balanced systems. *Int. J. Control*, 50:1379–1405, 1989.

[LAL90] Y. Liu, B. D. O. Anderson, and U. L. Ly. Coprime factorization controller reduction with Bezout identity induced frequency weighting. *Automatica*, 26:233–249, 1990.

[LC92] C.-A. Lin and T.-Y. Chiu. Model reduction via frequency weighted balanced realization. *CONTROL - Theory and Advanced Technology*, 8:341–351, 1992.

[LHPW87] A. J. Laub, M.T. Heath, C.C. Paige, and R.C. Ward. Computation of system balancing transformations and other applications of simultaneous diagonalization algorithms. *IEEE Trans. Automat. Control*, 32:115–122, 1987.

[Moo81] B. C. Moore. Principal component analysis in linear system: controllability, observability and model reduction. *IEEE Trans. Automat. Control*, 26:17–32, 1981.

[OA00] G. Obinata and B. D. O. Anderson. *Model Reduction for Control System Design*. Springer Verlag, Berlin, 2000.

[SC88] M. G. Safonov and R. Y. Chiang. Model reduction for robust control: a Schur relative error method. *Int. J. Adapt. Contr.&Sign. Proc.*, 2:259–272, 1988.

[SC89] M. G. Safonov and R. Y. Chiang. A Schur method for balanced-truncation model reduction. *IEEE Trans. Automat. Control*, 34:729–733, 1989.

[SLC89] M. G. Safonov, D. J. N. Limebeer, and R. Y. Chiang. Simplifying the $\mathcal{H}_\infty$ theory via loop shifting, matrix-pencil and descriptor concepts. *Int. J. Control*, 50:24672488, 1989.

[SM96] G. Schelfhout and B. De Moor. A note on closed-loop balanced truncation. *IEEE Trans. Automat. Control*, 41:1498–1500, 1996.

[TP87] M. S. Tombs and I. Postlethwaite. Truncated balanced realization of a stable non-minimal state-space system. *Int. J. Control*, 46:1319–1330, 1987.

[VA01] A. Varga and B. D. O. Anderson. Square-root balancing-free methods for the frequency-weighted balancing related model reduction. *Proc. of CDC'2001, Orlando, FL*, pp. 3659–3664, 2001.

[VA02] A. Varga and B. D. O. Anderson. Frequency-weighted balancing related controller reduction. *Proc. of IFAC'2002 Congress, Barcelona, Spain*, 2002.

[VA03] A. Varga and B. D. O. Anderson. Accuracy-enhancing methods for balancing-related frequency-weighted model and controller reduction. *Automatica*, 39:919–927, 2003.

[Var81] A. Varga. A Schur method for pole assignment. *IEEE Trans. Automat. Control*, 26:517–519, 1981.

[Var91a] A. Varga. Balancing-free square-root algorithm for computing singular perturbation approximations. *Proc. of 30th IEEE CDC, Brighton, UK*, pp. 1062–1065, 1991.

[Var91b] A. Varga. Efficient minimal realization procedure based on balancing. In A. El Moudni, P. Borne, and S. G. Tzafestas (Eds.), *Prepr. of IMACS Symp. on Modelling and Control of Technological Systems*, vol. 2, pp. 42–47, 1991.

[Var92] A. Varga. Coprime factors model reduction based on square-root balancing-free techniques. In A. Sydow (Ed.), *Computational System*

Analysis 1992, Proc. 4-th Int. Symp. Systems Analysis and Simulation, Berlin, Germany, pp. 91–96, Elsevier, Amsterdam, 1992.

[Var93] A. Varga. Coprime factors model reduction based on accuracy enhancing techniques. *Systems Analysis Modelling and Simulation*, 11:303–311, 1993.

[Var01a] A. Varga. Model reduction software in the SLICOT library. In B. N. Datta (Ed.), *Applied and Computational Control, Signals and Circuits*, vol. 629 of *The Kluwer International Series in Engineering and Computer Science*, pp. 239–282, Kluwer Academic Publishers, Boston, 2001.

[Var01b] A. Varga. Numerical approach for the frequency-weighted Hankel-norm approximation. *Proc. of ECC'2001, Porto, Portugal*, pp. 640–645, 2001.

[Var02a] A. Varga. New Numerical Software for Model and Controller Reduction. NICONET Report 2002-5, June 2002.

[Var02b] A. Varga. Numerical software in SLICOT for low order controller design. *Proc. of CACSD'2002, Glasgow, UK*, 2002.

[Var03a] A. Varga. Coprime factor reduction of $\mathcal{H}_\infty$ controllers. *Proc. of ECC'2003, Cambridge, UK*, 2003.

[Var03b] A. Varga. On frequency-weighted coprime factorization based controller reduction. *Proc. of ACC'2003, Denver, CO, USA*, 2003.

[VF93] A. Varga and K. H. Fasol. A new square-root balancing-free stochastic truncation model reduction algorithm. *Prepr. of 12th IFAC World Congress, Sydney, Australia*, vol. 7, pp. 153–156, 1993.

[Wal90] D. J. Walker. Robust stabilizability of discrete-time systems with normalized stable factor perturbation. *Int. J. Control*, 52:441–455, 1990.

[WSL99] G. Wang, V. Sreeram, and W. Q. Liu. A new frequency-weighted balanced truncation method and error bound. *IEEE Trans. Automat. Control*, 44:1734–1737, 1999.

[WSL01] G. Wang, V. Sreeram, and W. Q. Liu. Performance preserving controller reduction via additive perturbation of the closed-loop transfer function. *IEEE Trans. Automat. Control*, 46:771–775, 2001.

[ZC95] K. Zhou and J. Chen. Performance bounds for coprime factor controller reductions. *Systems & Control Lett.*, 26:119–127, 1995.

[ZDG96] K. Zhou, J. C. Doyle, and K. Glover. *Robust and Optimal Control*. Prentice Hall, 1996.

[Zho95] K. Zhou. Frequency-weighted L_∞ norm and optimal Hankel norm model reduction. *IEEE Trans. Automat. Control*, 40:1687–1699, 1995.

10

Proper Orthogonal Decomposition Surrogate Models for Nonlinear Dynamical Systems: Error Estimates and Suboptimal Control

Michael Hinze[1] and Stefan Volkwein[2]

[1] Institut für Numerische Mathematik, TU Dresden, D-01069 Dresden, Germany
hinze@math.tu-dresden.de
[2] Institut für Mathematik und Wissenschaftliches Rechnen, Karl-Franzens Universität Graz, Heinrichstrasse 36, A-8010 Graz, Austria
stefan.volkwein@uni-graz.at

10.1 Motivation

Optimal control problems for nonlinear partial differential equations are often hard to tackle numerically so that the need for developing novel techniques emerges. One such technique is given by reduced order methods. Recently the application of reduced-order models to optimal control problems for partial differential equations has received an increasing amount of attention. The reduced-order approach is based on projecting the dynamical system onto subspaces consisting of basis elements that contain characteristics of the expected solution. This is in contrast to, e.g., finite element techniques, where the elements of the subspaces are uncorrelated to the physical properties of the system that they approximate. The reduced basis method as developed, e.g., in [IR98] is one such reduced-order method with the basis elements corresponding to the dynamics of expected control regimes.

Proper orthogonal decomposition (POD) provides a method for deriving low order models of dynamical systems. It was successfully used in a variety of fields including signal analysis and pattern recognition (see [Fuk90]), fluid dynamics and coherent structures (see [AHLS88, HLB96, NAMTT03, RF94, Sir87]) and more recently in control theory (see [AH01, AFS00, LT01, SK98, TGP99]) and inverse problems (see [BJWW00]). Moreover, in [ABK01] POD was successfully utilized to compute reduced-order controllers. The relationship between POD and balancing was considered in [LMG, Row04, WP01]. Error analysis for nonlinear dynamical systems in finite dimensions were carried out in [RP02].

In our application we apply POD to derive a Galerkin approximation in the spatial variable, with basis functions corresponding to the solution of the physical system at pre-specified time instances. These are called the snap-

shots. Due to possible linear dependence or almost linear dependence, the snapshots themselves are not appropriate as a basis. Rather a singular value decomposition (SVD) is carried out and the leading generalized eigenfunctions are chosen as a basis, referred to as the POD basis.

The paper is organized as follows. In Section 10.2 the POD method and its relation to SVD is described. Furthermore, the snapshot form of POD for abstract parabolic equations is illustrated. Section 10.3 deals with reduced order modeling of nonlinear dynamical systems. Among other things, error estimates for reduced order models of a general equation in fluid mechanics obtained by the snapshot POD method are presented. Section 10.4 deals with suboptimal control strategies based on POD. For optimal open-loop control problems an adaptive optimization algorithm is presented which in every iteration uses a surrogate model obtained by the POD method instead of the full dynamics. In particular, in Section 10.4.2 first steps towards error estimation for optimal control problems are presented whose discretization is based on POD. The practical behavior of the proposed adaptive optimization algorithm is illustrated for two applications involving the time-dependent Navier-Stokes system in Section 10.5. For closed-loop control we refer the reader to [Gom02, KV99, KVX04, LV03], for instance. Finally, we draw some conclusions and discuss future research perspectives in Section 10.6.

10.2 The POD Method

In this section we propose the POD method and its numerical realization. In particular, we consider both POD in $\mathbb{C}^n$ (finite-dimensional case) and POD in Hilbert spaces; see Sections 10.2.1 and 10.2.2, respectively. For more details we refer to, e.g., [HLB96, KV99, Vol01a].

10.2.1 Finite-Dimensional POD

In this subsection we concentrate on POD in the finite dimensional setting and emphasize the close connection between POD and the singular value decomposition (SVD) of rectangular matrices; see [KV99]. Furthermore, the numerical realization of POD is explained.

POD and SVD

Let Y be a possibly complex valued $n \times m$ matrix of rank d. In the context of POD it will be useful to think of the columns $\{Y_{\cdot,j}\}_{j=1}^m$ of Y as the spatial coordinate vector of a dynamical system at time t_j. Similarly we consider the rows $\{Y_{i,\cdot}\}_{i=1}^n$ of Y as the time-trajectories of the dynamical system evaluated at the locations x_i.

From SVD (see, e.g., [Nob69]) the existence of real numbers $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_d > 0$ and unitary matrices $U \in \mathbb{C}^{n\times n}$ with columns $\{u_i\}_{i=1}^n$ and $V \in \mathbb{C}^{m\times m}$ with columns $\{v_i\}_{i=1}^m$ such that

$$U^H Y V = \begin{pmatrix} D & 0 \\ 0 & 0 \end{pmatrix} =: \Sigma \in \mathbb{C}^{n\times m}, \tag{10.1}$$

where $D = \operatorname{diag}(\sigma_1, \ldots, \sigma_d) \in \mathbb{R}^{d\times d}$, the zeros in (10.1) denote matrices of appropriate dimensions, and the superindex H stands for complex conjugation. Moreover, the vectors $\{u_i\}_{i=1}^d$ and $\{v_i\}_{i=1}^d$ satisfy

$$Y v_i = \sigma_i u_i \quad \text{and} \quad Y^H u_i = \sigma_i v_i \quad \text{for } i = 1, \ldots, d. \tag{10.2}$$

They are eigenvectors of YY^H and $Y^H Y$ with eigenvalues σ_i^2, $i = 1, \ldots, d$. The vectors $\{u_i\}_{i=d+1}^m$ and $\{v_i\}_{i=d+1}^m$ (if $d < n$ respectively $d < m$) are eigenvectors of YY^H and $Y^H Y$, respectively, with eigenvalue 0. If $Y \in \mathbb{R}^{n\times m}$ then U and V can be chosen to be real-valued.

From (10.2) we deduce that $Y = U\Sigma V^H$. It follows that Y can also be expressed as

$$Y = U^d D (V^d)^H, \tag{10.3}$$

where $U^d \in \mathbb{C}^{n\times d}$ and $V^d \in \mathbb{C}^{m\times d}$ are given by

$$\begin{aligned} U^d_{i,j} &= U_{i,j} \quad \text{for } 1 \le i \le n, 1 \le j \le d, \\ V^d_{i,j} &= V_{i,j} \quad \text{for } 1 \le i \le m, 1 \le j \le d. \end{aligned}$$

It will be convenient to express (10.3) as

$$Y = U^d B \quad \text{with } B = D(V^d)^H \in \mathbb{C}^{d\times m}.$$

Thus the column space of Y can be represented in terms of the d linearly independent columns of U^d. The coefficients in the expansion for the columns $Y_{\cdot,j}$, $j = 1, \ldots, m$, in the basis $\{U^d_{\cdot,i}\}_{i=1}^d$ are given by the $B_{\cdot,j}$. Since U is Hermitian we easily find that

$$Y_{\cdot,j} = \sum_{i=1}^d B_{i,j} U^d_{\cdot,i} = \sum_{i=1}^d \langle U_{\cdot,i}, Y_{\cdot,j} \rangle_{\mathbb{C}^n} U^d_{\cdot,i},$$

where $\langle \cdot, \cdot \rangle_{\mathbb{C}^n}$ denotes the canonical inner product in $\mathbb{C}^n$. In terms of the columns y_j of Y we express the last equality as

$$y_j = \sum_{i=1}^d B_{i,j} u_i = \sum_{i=1}^d \langle u_i, y_j \rangle_{\mathbb{C}^n} u_i, \quad j = 1, \ldots, m.$$

Let us now interpret singular value decomposition in terms of POD. One of the central issues of POD is the reduction of data expressing their "essential information" by means of a few basis vectors. The problem of approximating all spatial coordinate vectors y_j of Y simultaneously by a single, normalized vector as well as possible can be expressed as

$$\max \sum_{j=1}^{m} \left| \langle y_j, u \rangle_{\mathbb{C}^n} \right|^2 \text{ subject to (s.t.) } |u|_{\mathbb{C}^n} = 1. \tag{P}$$

Here, $|\cdot|_{\mathbb{C}^n}$ denotes the Euclidean norm in $\mathbb{C}^n$. Utilizing a Lagrangian framework a necessary optimality condition for (P) is given by the eigenvalue problem

$$YY^H u = \sigma^2 u. \tag{10.4}$$

Due to singular value analysis u_1 solves (P) and argmax (P) $= \sigma_1^2$. If we were to determine a second vector, orthogonal to u_1 that again describes the data set $\{y_i\}_{i=1}^m$ as well as possible then we need to solve

$$\max \sum_{j=1}^{m} \left| \langle y_j, u \rangle_{\mathbb{C}^n} \right|^2 \quad \text{s.t.} \quad |u|_{\mathbb{C}^n} = 1 \text{ and } \langle u, u_1 \rangle_{\mathbb{C}^n} = 0. \tag{P_2}$$

Rayleigh's principle and singular value decomposition imply that u_2 is a solution to (P_2) and argmax (P_2) $= \sigma_2^2$. Clearly this procedure can be continued by finite induction so that u_k, $1 \le k \le d$, solves

$$\max \sum_{j=1}^{m} \left| \langle y_j, u \rangle_{\mathbb{C}^n} \right|^2 \quad \text{s.t.} \quad |u|_{\mathbb{C}^n} = 1 \text{ and } \langle u, u_i \rangle_{\mathbb{C}^n} = 0,\ 1 \le i \le k-1. \tag{P_k}$$

The following result which states that for every $\ell \le k$ the approximation of the columns of Y by the first ℓ singular vectors $\{u_i\}_{i=1}^{\ell}$ is optimal in the mean among all rank ℓ approximations to the columns of Y is now quite natural. More precisely, let $\hat{U} \in \mathbb{C}^{n \times d}$ denote a matrix with pairwise orthonormal vectors $\hat{u}_i$ and let the expansion of the columns of Y in the basis $\{\hat{u}_i\}_{i=1}^d$ be given by

$$Y = \hat{U}\hat{B}, \text{ where } \hat{B}_{i,j} = \langle \hat{u}_i, y_j \rangle_{\mathbb{C}^n} \quad \text{for } 1 \le i \le d, 1 \le j \le m.$$

Then for every $\ell \le k$ we have

$$\|Y - \hat{U}^\ell \hat{B}^\ell\|_F \ge \|Y - U^\ell B^\ell\|_F. \tag{10.5}$$

Here, $\|\cdot\|_F$ denotes the Frobenius norm, U^ℓ denotes the first ℓ columns of U, B^ℓ the first ℓ rows of B and similarly for $\hat{U}^\ell$ and $\hat{B}^\ell$. Note that the j-th column of $U^\ell B^\ell$ represents the Fourier expansion of order ℓ of the j-th column y_j of Y in the orthonormal basis $\{u_i\}_{i=1}^{\ell}$. Utilizing the fact that $\hat{U}\hat{B}^\ell$ has rank ℓ and recalling that $B^\ell = (D(V^k)^H)^\ell$ estimate (10.5) follows directly from singular value analysis [Nob69]. We refer to U^ℓ as the POD-basis of rank ℓ. Then we have

$$\sum_{i=\ell+1}^{d} \sigma_i^2 = \sum_{i=\ell+1}^{d} \left(\sum_{j=1}^{m} |B_{i,j}|^2 \right) \le \sum_{i=\ell+1}^{d} \left(\sum_{j=1}^{m} |\hat{B}_{i,j}|^2 \right). \tag{10.6}$$

and

$$\sum_{i=1}^{\ell} \sigma_i^2 = \sum_{i=1}^{\ell} \left(\sum_{j=1}^{m} |B_{i,j}|^2 \right) \geq \sum_{i=1}^{\ell} \left(\sum_{j=1}^{m} |\hat{B}_{i,j}|^2 \right). \tag{10.7}$$

Inequalities (10.6) and (10.7) establish that for every $1 \leq \ell \leq d$ the POD-basis of rank ℓ is optimal in the sense of representing in the mean the columns of Y as a linear combination by a basis of rank ℓ. Adopting the interpretation of the $Y_{i,j}$ as the velocity of a fluid at location x_i and at time t_j, inequality (10.7) expresses the fact that the first ℓ POD-basis functions capture more energy on average than the first ℓ functions of any other basis.

The POD-expansion Y^ℓ of rank ℓ is given by

$$Y^\ell = U^\ell B^\ell = U^\ell \left(D(V^d)^H \right)^\ell,$$

and hence the "t-average" of the coefficients satisfies

$$\langle B^\ell_{i,\cdot}, B^\ell_{j,\cdot} \rangle_{\mathbb{C}^m} = \sigma_i^2 \delta_{ij} \quad \text{for } 1 \leq i,j \leq \ell.$$

This property is referred to as the fact that the POD-coefficients are uncorrelated.

Computational Issues

Concerning the practical computation of a POD-basis of rank ℓ let us note that if $m < n$ then one can choose to determine m eigenvectors v_i corresponding to the largest eigenvalues of $Y^H Y \in \mathbb{C}^{m \times m}$ and by (10.2) determine the POD-basis from

$$u_i = \frac{1}{\sigma_i} Y v_i, \quad i = 1, \ldots, \ell. \tag{10.8}$$

Note that the square matrix $Y^H Y$ has the dimension of number of "time-instances" t_j. For historical reasons [Sir87] this method of determine the POD-basis is sometimes called the method of snapshots.

For the application of POD to concrete problems the choice of ℓ is certainly of central importance, as is also the number and location of snapshots. It appears that no general a-priori rules are available. Rather the choice of ℓ is based on heuristic considerations combined with observing the ratio of the modeled to the total information content contained in the system Y, which is expressed by

$$\mathcal{E}(\ell) = \frac{\sum_{i=1}^{\ell} \sigma_i^2}{\sum_{i=1}^{d} \sigma_i^2} \quad \text{for } \ell \in \{1, \ldots, d\}. \tag{10.9}$$

For a further discussion, also of adaptive strategies based e.g. on this term we refer to [MM03] and the literature cited there.

Application to Discrete Solutions to Dynamical Systems

Let us now assume that $Y \in \mathbb{R}^{n\times m}$, $n \geq m$, arises from discretization of a dynamical system, where a finite element approach has been utilized to discretize the state variable $y = y(x,t)$, i.e.,

$$y_h(x,t_j) = \sum_{i=1}^{n} Y_{i,j}\varphi_i(x) \quad \text{for } x \in \Omega,$$

with φ_i, $1 \leq i \leq n$, denoting the finite element functions and Ω being a bounded domain in $\mathbb{R}^2$ or $\mathbb{R}^3$. The goal is to describe the ensemble $\{y_h(\cdot\,,t_j)\}_{j=1}^m$ of L^2-functions simultaneously by a single normalized L^2-function ψ as well as possible:

$$\max \sum_{j=1}^{m} \left|\langle y_h(\cdot,t_j),\psi\rangle_{L^2(\Omega)}\right|^2 \quad \text{s.t.} \quad \|\psi\|_{L^2(\Omega)} = 1, \tag{$\tilde{\mathsf{P}}$}$$

where $\langle\cdot\,,\cdot\rangle_{L^2(\Omega)}$ is the canonical inner product in $L^2(\Omega)$. Since $y_h(\cdot\,,t_j) \in \operatorname{span}\{\varphi_1,\ldots,\varphi_n\}$ holds for $1 \leq j \leq n$, we have $\psi \in \operatorname{span}\{\varphi_1,\ldots,\varphi_n\}$. Let v be the vector containing the components v_i such that

$$\psi(x) = \sum_{i=1}^{n} v_i\varphi_i(x)$$

and let $S \in \mathbb{R}^{n\times n}$ denote the positive definite mass matrix with the elements $\langle\varphi_i,\varphi_j\rangle_{L^2(\Omega)}$. Instead of (10.4) we obtain that

$$YY^TSv = \sigma^2 v. \tag{10.10}$$

The eigenvalue problem (10.10) can be solved by utilizing singular value analysis. Multiplying (10.10) by the positive square root $S^{1/2}$ of S from the left and setting $u = S^{1/2}v$ we obtain the $n \times n$ eigenvalue problem

$$\tilde{Y}\tilde{Y}^Tu = \sigma^2 u, \tag{10.11}$$

where $\tilde{Y} = S^{1/2}Y \in \mathbb{R}^{n\times m}$. We mention that (10.11) coincides with (10.4) when $\{\varphi_i\}_{i=1}^n$ is an orthonormal set in $L^2(\Omega)$. Note that if Y has rank k the matrix $\tilde{Y}$ has also rank d. Applying the singular value decomposition to the rectangular matrix $\tilde{Y}$ we have

$$\tilde{Y} = U\Sigma V^T$$

(see (10.1)). Analogous to (10.3) it follows that

$$\tilde{Y} = U^dD(V^d)^T, \tag{10.12}$$

where again U^d and V^d contain the first k columns of the matrices U and V, respectively. Using (10.12) we determine the coefficient matrix $\Psi = S^{-1/2}U^d \in \mathbb{R}^{n\times d}$, so that the first k POD-basis functions are given by

$$\psi_j(x) = \sum_{i=1}^{n} \Psi_{i,j}\varphi_i(x), \quad j = 1, \dots, d.$$

Due to (10.11) and $\Psi_{\cdot,j} = S^{-1/2}U^d_{\cdot,j}$, $1 \le j \le d$, the vectors $\Psi_{\cdot,j}$ are eigenvectors of problem (10.10) with corresponding eigenvalues σ_j^2:

$$YY^T S\Psi_{\cdot,j} = YY^T SS^{-1/2}U^k_{\cdot,j} = S^{-1/2}\tilde{Y}\tilde{Y}^T U^k_{\cdot,j} = \sigma_j^2 S^{-1/2}U^k_{\cdot,j} = \sigma_j^2\Psi_{\cdot,j}.$$

Therefore, the function ψ_1 solves $(\tilde{\mathsf{P}})$ with argmax $(\tilde{\mathsf{P}}) = \sigma_1^2$ and, by finite induction, the function ψ_k, $k \in \{2, \dots, d\}$, solves

$$\max \sum_{j=1}^{m} \left|\langle y_h(\cdot, t_j), \psi\rangle_{L^2(\Omega)}\right|^2 \text{ s.t. } \|\psi\|_{L^2(\Omega)} = 1,\ \langle\psi, \psi_i\rangle_{L^2(\Omega)} = 0, i < d,\ (\tilde{\mathsf{P}}_k)$$

with argmax $(\tilde{\mathsf{P}}_k) = \sigma_k^2$. Since we have $\Psi_{\cdot,j} = S^{-1/2}U^d_{\cdot,j}$, the functions $\psi_1, \dots, \psi_d$ are orthonormal with respect to the L^2-inner product:

$$\langle\psi_i, \psi_j\rangle_{L^2(\Omega)} = \langle\Psi_{\cdot,i}, S\Psi_{\cdot,j}\rangle_{\mathbb{C}^n} = \langle u_i, u_j\rangle_{\mathbb{C}^n} = \delta_{ij}, \quad 1 \le i, j \le d.$$

Note that the coefficient matrix Ψ can also be computed by using generalized singular value analysis. If we multiply (10.10) with S from the left we obtain the generalized eigenvalue problem

$$SYY^T Su = \sigma^2 Su.$$

From generalized SVD [GL89] there exist orthogonal $V \in \mathbb{R}^{m\times m}$ and $U \in \mathbb{R}^{n\times n}$ and an invertible $R \in \mathbb{R}^{n\times n}$ such that

$$V(Y^T S)R = \begin{pmatrix} E & 0 \\ 0 & 0 \end{pmatrix} =: \Sigma_1 \in \mathbb{R}^{m\times n}, \tag{10.13a}$$

$$US^{1/2}R = \Sigma_2 \in \mathbb{R}^{n\times n}, \tag{10.13b}$$

where $E = \operatorname{diag}(e_1, \dots, e_d)$ with $e_i > 0$ and $\Sigma_2 = \operatorname{diag}(s_1, \dots, s_n)$ with $s_i > 0$. From (10.13b) we infer that

$$R = S^{-1/2}U^T\Sigma_2. \tag{10.14}$$

Inserting (10.14) into (10.13a) we obtain that

$$\Sigma_2^{-1}\Sigma_1^T = \Sigma_2^{-1}R^T SYV^T = US^{1/2}YV^T,$$

which is the singular value decomposition of the matrix $S^{1/2}Y$ with $\sigma_i = e_i/s_i > 0$ for $i = 1, \dots, d$. Hence, Ψ is again equal to the first k columns of $S^{1/2}U$.

If $m \leq n$ we proceed to determine the matrix Ψ as follows. From $u_j = (1/\sigma_j)\, S^{1/2} Y v_j$ for $1 \leq j \leq d$ we infer that

$$\Psi_{\cdot,j} = \frac{1}{\sigma_j}\, Y v_j,$$

where v_j solves the $m \times m$ eigenvalue problem

$$Y^T S Y v_j = \sigma_i^2 v_j, \quad 1 \leq j \leq d.$$

Note that the elements of the matrix $Y^T S Y$ are given by the integrals

$$\langle y(\cdot, t_i), y(\cdot, t_j) \rangle_{L^2(\Omega)}, \quad 1 \leq i, j \leq n, \tag{10.15}$$

so that the matrix $Y^T S Y$ is often called a *correlation matrix*.

10.2.2 POD for Parabolic Systems

Whereas in the last subsection POD has been motivated by rectangular matrices and SVD, we concentrate on POD for dynamical (non-linear) systems in this subsection.

Abstract Nonlinear Dynamical System

Let V and H be real separable Hilbert spaces and suppose that V is dense in H with compact embedding. By $\langle \cdot, \cdot \rangle_H$ we denote the inner product in H. The inner product in V is given by a symmetric bounded, coercive, bilinear form $a : V \times V \to \mathbb{R}$:

$$\langle \varphi, \psi \rangle_V = a(\varphi, \psi) \quad \text{for all } \varphi, \psi \in V \tag{10.16}$$

with associated norm given by $\| \cdot \|_V = \sqrt{a(\cdot, \cdot)}$. Since V is continuously injected into H, there exists a constant $c_V > 0$ such that

$$\|\varphi\|_H \leq c_V \, \|\varphi\|_V \quad \text{for all } \varphi \in V. \tag{10.17}$$

We associate with a the linear operator A:

$$\langle A\varphi, \psi \rangle_{V',V} = a(\varphi, \psi) \quad \text{for all } \varphi, \psi \in V,$$

where $\langle \cdot, \cdot \rangle_{V',V}$ denotes the duality pairing between V and its dual. Then, by the Lax-Milgram lemma, A is an isomorphism from V onto V'. Alternatively, A can be considered as a linear unbounded self-adjoint operator in H with domain

$$D(A) = \{\varphi \in V : A\varphi \in H\}.$$

By identifying H and its dual H' it follows that

$$D(A) \hookrightarrow V \hookrightarrow H = H' \hookrightarrow V',$$

each embedding being continuous and dense, when $D(A)$ is endowed with the graph norm of A.

Moreover, let $F : V \times V \to V'$ be a bilinear continuous operator mapping $D(A) \times D(A)$ into H. To simplify the notation we set $F(\varphi) = F(\varphi, \varphi)$ for $\varphi \in V$. For given $f \in C([0,T]; H)$ and $y_0 \in V$ we consider the nonlinear evolution problem

$$\frac{\mathrm{d}}{\mathrm{d}t} \langle y(t), \varphi \rangle_H + a(y(t), \varphi) + \langle F(y(t)), \varphi \rangle_{V',V} = \langle f(t), \varphi \rangle_H \tag{10.18a}$$

for all $\varphi \in V$ and $t \in (0,T]$ a.e. and

$$y(0) = y_0 \quad \text{in } H. \tag{10.18b}$$

Assumption (A1). *For every $f \in C([0,T]; H)$ and $y_0 \in V$ there exists a unique solution of* (10.18) *satisfying*

$$y \in C([0,T]; V) \cap L^2(0,T; D(A)) \cap H^1(0,T; H). \tag{10.19}$$

Computation of the POD Basis

Throughout we assume that Assumption **(A1)** holds and we denote by y the unique solution to (10.18) satisfying (10.19). For given $n \in \mathbb{N}$ let

$$0 = t_0 < t_1 < \ldots < t_n \le T \tag{10.20}$$

denote a grid in the interval $[0,T]$ and set $\delta t_j = t_j - t_{j-1}$, $j = 1, \ldots, n$. Define

$$\Delta t = \max\ (\delta t_1, \ldots, \delta t_n) \quad \text{and} \quad \delta t = \min\ (\delta t_1, \ldots, \delta t_n). \tag{10.21}$$

Suppose that the *snapshots* $y(t_j)$ of (10.18) at the given time instances t_j, $j = 0, \ldots, n$, are known. We set

$$\mathcal{V} = \operatorname{span}\{y_0, \ldots, y_{2n}\},$$

where $y_j = y(t_j)$ for $j = 0, \ldots, n$, $y_j = \overline{\partial}_t y(t_{j-n})$ for $j = n+1, \ldots, 2n$ with $\overline{\partial}_t y(t_j) = (y(t_j) - y(t_{j-1}))/\delta t_j$, and refer to $\mathcal{V}$ as the ensemble consisting of the snapshots $\{y_j\}_{j=0}^{2n}$, at least one of which is assumed to be nonzero. Furthermore, we call $\{t_j\}_{j=0}^{n}$ the snapshot grid. Notice that $\mathcal{V} \subset V$ by construction. Throughout the remainder of this section we let X denote either the space V or H.

*Remark 10.2.1 (*compare [KV01, Remark 1]). It may come as a surprise at first that the finite difference quotients $\overline{\partial}_t y(t_j)$ are included into the set $\mathcal{V}$ of snapshots. To motivate this choice let us point out that while the finite difference quotients are contained in the span of $\{y_j\}_{j=0}^{2n}$, the POD bases differ depending on whether $\{\overline{\partial}_t y(t_j)\}_{j=1}^{n}$ are included or not. The linear dependence does not constitute a difficulty for the singular value decomposition which is required to compute the POD basis. In fact, the snapshots themselves can be linearly dependent. The resulting POD basis is, in any case, maximally linearly independent in the sense expressed in $(\mathbf{P}_\ell)$ and Proposition 10.2.5. Secondly, in anticipation of the rate of convergence results that will be presented in Section 10.3.3 we note that the time derivative of y in (10.18) must be approximated by the Galerkin POD based scheme. In case the terms $\{\overline{\partial}_t y(t_j)\}_{j=1}^{n}$ are included in the snapshot ensemble, we are able to utilize the estimate

$$\sum_{j=1}^{n} \alpha_j \left\| \overline{\partial}_t y(t_j) - \sum_{i=1}^{\ell} \langle \overline{\partial}_t y(t_j), \psi_i \rangle_X \psi_i \right\|_X^2 \leq \sum_{i=\ell+1}^{d} \lambda_i. \tag{10.22}$$

Otherwise, if only the snapshots $y_j = y(t_j)$ for $j = 0, \ldots, n$, are used, we obtain instead of (10.37) the error formula

$$\sum_{j=0}^{n} \alpha_j \left\| y(t_j) - \sum_{i=1}^{\ell} \langle y(t_j), \psi_i \rangle_X \psi_i \right\|_X^2 = \sum_{i=\ell+1}^{d} \lambda_i,$$

and (10.22) must be replaced by

$$\sum_{j=1}^{n} \alpha_j \left\| \overline{\partial}_t y(t_j) - \sum_{i=1}^{\ell} \langle \overline{\partial}_t y(t_j), \psi_i \rangle_X \psi_i \right\|_X^2 \leq \frac{2}{(\delta t)^2} \sum_{i=\ell+1}^{d} \lambda_i, \tag{10.23}$$

which in contrast to (10.22) contains the factor $(\delta t)^{-2}$ on the right-hand side. In [HV03] this fact was observed numerically. Moreover, in [LV03] it turns out that the inclusion of the difference quotients improves the stability properties of the computed feedback control laws. Let us mention the article [AG03], where the time derivatives were also included in the snapshot ensemble to get a better approximation of the dynamical system. ◊

Let $\{\psi_i\}_{i=1}^{d}$ denote an orthonormal basis for $\mathcal{V}$ with $d = \dim \mathcal{V}$. Then each member of the ensemble can be expressed as

$$y_j = \sum_{i=1}^{d} \langle y_j, \psi_i \rangle_X \psi_i \quad \text{for } j = 0, \ldots, 2n. \tag{10.24}$$

The method of POD consists in choosing an orthonormal basis such that for every $\ell \in \{1, \ldots, d\}$ the mean square error between the elements y_j, $0 \leq j \leq 2n$, and the corresponding ℓ-th partial sum of (10.24) is minimized on average:

$$\begin{aligned} &\min J(\psi_1,\dots,\psi_\ell) = \sum_{j=0}^{2n} \alpha_j \Big\| y_j - \sum_{i=1}^{\ell} \langle y_j, \psi_i\rangle_X \psi_i \Big\|_X^2 \\ &\text{s.t. } \langle \psi_i, \psi_j\rangle_X = \delta_{ij} \quad \text{for } 1 \le i \le \ell, 1 \le j \le i. \end{aligned} \tag{$\mathbf{P}_\ell$}$$

Here $\{\alpha_j\}_{j=0}^{2n}$ are positive weights, which for our purposes are chosen to be

$$\alpha_0 = \frac{\delta t_1}{2}, \quad \alpha_j = \frac{\delta t_j + \delta t_{j+1}}{2} \text{ for } j = 1,\dots,n-1, \quad \alpha_n = \frac{\delta t_n}{2}$$

and $\alpha_j = \alpha_{j-n}$ for $j = n+1,\dots,2n$.

Remark 10.2.2. 1) Note that

$$\mathcal{I}_n(y) = J(\psi_1,\dots,\psi_\ell)$$

can be interpreted as a trapezoidal approximation for the integral

$$\mathcal{I}(y) = \int_0^T \Big\| y(t) - \sum_{i=1}^{\ell} \langle y(t), \psi_i\rangle_X \psi_i \Big\|_X^2 + \Big\| y_t(t) - \sum_{i=1}^{\ell} \langle y_t(t), \psi_i\rangle_X \psi_i \Big\|_X^2 \,\mathrm{d}t.$$

For all $y \in C^1([0,T];X)$ it follows that $\lim_{n\to\infty} \mathcal{I}_n(y) = \mathcal{I}(y)$. In Section 10.4.2 we will address the continuous version of POD (see, in particular, Theorem 10.4.3).

2) Notice that $(\mathbf{P}_\ell)$ is equivalent with

$$\max \sum_{i=1}^{\ell} \sum_{j=0}^{2n} \alpha_j \big| \langle y_j, \psi_i\rangle_X \big|^2 \quad \text{s.t.} \quad \langle \psi_i, \psi_j\rangle_X = \delta_{ij},\ 1 \le j \le i \le \ell. \tag{10.25}$$

For $X = \mathbb{C}^n$, $\ell = 1$ and $\alpha_j = 1$ for $1 \le j \le n$ and $\alpha_j = 0$ otherwise, (10.25) is equivalent with (P). ◊

A solution $\{\psi_i\}_{i=1}^{\ell}$ to $(\mathbf{P}_\ell)$ is called *POD basis of rank* ℓ. The subspace spanned by the first ℓ POD basis functions is denoted by V^ℓ, i.e.,

$$V^\ell = \operatorname{span}\{\psi_1,\dots,\psi_\ell\}. \tag{10.26}$$

The solution of $(\mathbf{P}_\ell)$ is characterized by necessary optimality conditions, which can be written as an eigenvalue problem; compare Section 10.2.1. For that purpose we endow $\mathbb{R}^{2n+1}$ with the weighted inner product

$$\langle v, w\rangle_\alpha = \sum_{j=0}^{2n} \alpha_j v_j w_j \tag{10.27}$$

for $v = (v_0,\dots,v_{2n})^T, w = (w_0,\dots,w_{2n})^T \in \mathbb{R}^{2n+1}$ and the induced norm.

Remark 10.2.3. Due to the choices for the weights α_j's the weighted inner product $\langle \cdot, \cdot \rangle_\alpha$ can be interpreted as the trapezoidal approximation for the H^1-inner product

$$\langle v, w \rangle_{H^1(0,T)} = \int_0^T vw + v_t w_t \, \mathrm{d}t \quad \text{for } v, w \in H^1(0,T)$$

so that (10.27) is a discrete H^1-inner product (compare Section 10.4.2). $\Diamond$

Let us introduce the bounded linear operator $\mathcal{Y}_n : \mathbb{R}^{2n+1} \to X$ by

$$\mathcal{Y}_n v = \sum_{j=0}^{2n} \alpha_j v_j y_j \quad \text{for } v \in \mathbb{R}^{2n+1}. \tag{10.28}$$

Then the adjoint $\mathcal{Y}_n^* : X \to \mathbb{R}^{2n+1}$ is given by

$$\mathcal{Y}_n^* z = \big(\langle z, y_0 \rangle_X, \ldots, \langle z, y_{2n} \rangle_X \big)^T \quad \text{for } z \in X. \tag{10.29}$$

It follows that $\mathcal{R}_n = \mathcal{Y}_n \mathcal{Y}_n^* \in \mathcal{L}(X)$ and $\mathcal{K}_n = \mathcal{Y}_n^* \mathcal{Y}_n \in \mathbb{R}^{(2n+1)\times(2n+1)}$ are given by

$$\mathcal{R}_n z = \sum_{j=0}^{2n} \alpha_j \langle z, y_j \rangle_X y_j \quad \text{for } z \in X \quad \text{and} \quad \big(\mathcal{K}_n\big)_{ij} = \alpha_j \, \langle y_j, y_i \rangle_X \tag{10.30}$$

respectively. By $\mathcal{L}(X)$ we denote the Banach space of all linear and bounded operators from X into itself and the matrix $\mathcal{K}_n$ is again called a *correlation matrix*; compare (10.15).

Using a Lagrangian framework we derive the following optimality conditions for the optimization problem $(\mathbf{P}_\ell)$:

$$\mathcal{R}_n \psi = \lambda \psi, \tag{10.31}$$

compare e.g. [HLB96, pp. 88-91] and [Vol01a, Section 2]. Thus, it turns out that analogous to finite-dimensional POD, we obtain an eigenvalue problem; see (10.4).

Note that $\mathcal{R}_n$ is a bounded, self-adjoint and nonnegative operator. Moreover, since the image of $\mathcal{R}_n$ has finite dimension, $\mathcal{R}_n$ is also compact. By Hilbert-Schmidt theory (see e.g. [RS80, p. 203]) there exist an orthonormal basis $\{\psi_i\}_{i \in \mathbb{N}}$ for X and a sequence $\{\lambda_i\}_{i \in \mathbb{N}}$ of nonnegative real numbers so that

$$\mathcal{R}_n \psi_i = \lambda_i \psi_i, \quad \lambda_1 \geq \ldots \geq \lambda_d > 0 \quad \text{and } \lambda_i = 0 \quad \text{for} \quad i > d. \tag{10.32}$$

Moreover, $\mathcal{V} = \operatorname{span} \{\psi_i\}_{i=1}^d$. Note that $\{\lambda_i\}_{i \in \mathbb{N}}$ as well as $\{\psi_i\}_{i \in \mathbb{N}}$ depend on n.

Remark 10.2.4. a) Setting $\sigma_i = \sqrt{\lambda_i}$, $i = 1, \ldots, d$, and

$$v_i = \frac{1}{\sigma_i}\, \mathcal{Y}_n^* \psi_i \quad \text{for } i = 1, \ldots, d \tag{10.33}$$

we find

$$\mathcal{K}_n v_i = \lambda_i v_i \quad \text{and} \quad \langle v_i, v_j \rangle_\alpha = \delta_{ij},\ 1 \le i, j \le d. \tag{10.34}$$

Thus, $\{v_i\}_{i=1}^d$ is an orthonormal basis of eigenvectors of $\mathcal{K}_n$ for the image of $\mathcal{K}_n$. Conversely, if $\{v_i\}_{i=1}^d$ is a given orthonormal basis for the image of $\mathcal{K}_n$, then it follows that the first d eigenfunctions of $\mathcal{R}_n$ can be determined by

$$\psi_i = \frac{1}{\sigma_i}\, \mathcal{Y}_n v_i \quad \text{for } i = 1, \ldots, d, \tag{10.35}$$

see (10.8). Hence, we can determine the POD basis by solving either the eigenvalue problem for $\mathcal{R}_n$ or the one for $\mathcal{K}_n$. The relationship between the eigenfunctions of $\mathcal{R}_n$ and the eigenvectors for $\mathcal{K}_n$ is given by (10.33) and (10.35), which corresponds to SVD for the finite-dimensional POD.

b) Let us introduce the matrices

$$D = \operatorname{diag}(\alpha_0, \ldots, \alpha_{2n}) \in \mathbb{R}^{(2n+1)\times(2n+1)},$$
$$\widetilde{\mathcal{K}}_n = \big(\big(\langle y_j, y_i \rangle_X\big)\big)_{0 \le i,j \le 2n} \in \mathbb{R}^{(2n+1)\times(2n+1)}.$$

Note that the matrix $\widetilde{\mathcal{K}}_n$ is symmetric and positive semi-definite with $\operatorname{rank} \widetilde{\mathcal{K}}_n = d$. Then the eigenvalue problem (10.34) can be written in matrix-vector-notation as follows:

$$\widetilde{\mathcal{K}}_n D v_i = \lambda_i v_i \quad \text{and} \quad v_i^T D v_j = \delta_{ij},\ 1 \le i, j \le d. \tag{10.36}$$

Multiplying the first equation in (10.36) with $D^{1/2}$ from the left and setting $w_i = D^{1/2} v_i$, $1 \le i \le d$, we derive

$$D^{1/2} \widetilde{\mathcal{K}}_n D^{1/2} w_i = \lambda_i w_i \quad \text{and} \quad w_i^T w_j = \delta_{ij},\ 1 \le i, j \le d.$$

where the matrix $\widehat{\mathcal{K}}_n = D^{1/2} \widetilde{\mathcal{K}}_n D^{1/2}$ is symmetric and positive semi-definite with $\operatorname{rank} \widehat{\mathcal{K}}_n = d$. Therefore, it turns out that (10.34) can be expressed as a symmetric eigenvalue problem. ◊

The sequence $\{\psi_i\}_{i=1}^\ell$ solves the optimization problem $(\mathbf{P}_\ell)$. This fact as well as the error formula below were proved in [HLB96, Section 3], for example.

Proposition 10.2.5. *Let $\lambda_1 \ge \ldots \ge \lambda_d > 0$ denote the positive eigenvalues of $\mathcal{R}_n$ with the associated eigenvectors $\psi_1, \ldots, \psi_d \in X$. Then, $\{\psi_i^n\}_{i=1}^\ell$ is a POD basis of rank $\ell \le d$, and we have the error formula*

$$J(\psi_1, \ldots, \psi_\ell) = \sum_{j=0}^{2n} \alpha_j \Big\| y_j - \sum_{i=1}^{\ell} \langle y_j, \psi_i \rangle_X \psi_i \Big\|_X^2 = \sum_{i=\ell+1}^{d} \lambda_i. \tag{10.37}$$

10.3 Reduced-Order Modeling for Dynamical Systems

In the previous section we have described how to compute a POD basis. In this section we focus on the Galerkin projection of dynamical systems utilizing the POD basis functions. We obtain reduced-order models and present error estimates for the POD solution compared to the solution of the dynamical system.

10.3.1 A General Equation in Fluid Dynamics

In this subsection we specify the abstract nonlinear evolution problem that will be considered in this section and present an existence and uniqueness result, which ensures Assumption **(A1)** introduced in Section 10.2.2.

We introduce the continuous operator $R : V \to V'$, which maps $D(A)$ into H and satisfies

$$\begin{aligned} \|R\varphi\|_H &\le c_R \ \|\varphi\|_V^{1-\delta_1} \|A\varphi\|_H^{\delta_1} && \text{for all } \varphi \in D(A), \\ |\langle R\varphi, \varphi\rangle_{V',V}| &\le c_R \ \|\varphi\|_V^{1+\delta_2} \|\varphi\|_H^{1-\delta_2} && \text{for all } \varphi \in V \end{aligned}$$

for a constant $c_R > 0$ and for $\delta_1, \delta_2 \in [0, 1)$. We also assume that $A + R$ is coercive on V, i.e., there exists a constant $\eta > 0$ such that

$$a(\varphi, \varphi) + \langle R\varphi, \varphi\rangle_{V',V} \ge \eta \ \|\varphi\|_V^2 \quad \text{for all } \varphi \in V. \tag{10.38}$$

Moreover, let $B : V \times V \to V'$ be a bilinear continuous operator mapping $D(A) \times D(A)$ into H such that there exist constants $c_B > 0$ and $\delta_3, \delta_4, \delta_5 \in [0, 1)$ satisfying

$$\begin{aligned} \langle B(\varphi, \psi), \psi\rangle_{V',V} &= 0, \\ \big|\langle B(\varphi, \psi), \phi\rangle_{V',V}\big| &\le c_B \ \|\varphi\|_H^{\delta_3} \|\varphi\|_V^{1-\delta_3} \|\psi\|_V \|\phi\|_V^{\delta_3} \|\phi\|_H^{1-\delta_3}, \\ \|B(\varphi, \chi)\|_H + \|B(\chi, \varphi)\|_H &\le c_B \ \|\varphi\|_V \|\chi\|_V^{1-\delta_4} \|A\chi\|_H^{\delta_4}, \\ \|B(\varphi, \chi)\|_H &\le c_B \ \|\varphi\|_H^{\delta_5} \|\varphi\|_V^{1-\delta_5} \|\chi\|_V^{1-\delta_5} \|A\chi\|_H^{\delta_5}, \end{aligned}$$

for all $\varphi, \psi, \phi \in V$, for all $\chi \in D(A)$.

In the context of Section 10.2.2 we set $F = B + R$. Thus, for given $f \in C(0, T; H)$ and $y_0 \in V$ we consider the nonlinear evolution problem

$$\frac{\mathrm{d}}{\mathrm{d}t} \ \langle y(t), \varphi\rangle_H + a(y(t), \varphi) + \big\langle F(y(t)), \varphi\big\rangle_{V',V} = \langle f(t), \varphi\rangle_H \tag{10.39a}$$

for all $\varphi \in V$ and almost all $t \in (0, T]$ and

$$y(0) = y_0 \quad \text{in } H. \tag{10.39b}$$

The following theorem guarantees **(A1)**.

Theorem 10.3.1. *Suppose that the operators R and B satisfy the assumptions stated above. Then, for every $f \in C(0,T;H)$ and $y_0 \in V$ there exists a unique solution of* (10.39) *satisfying*

$$y \in C([0,T];V) \cap L^2(0,T;D(A)) \cap H^1(0,T;H). \tag{10.40}$$

Proof. The proof is analogous to that of Theorem 2.1 in [Tem88, p. 111], where the case with time-independent f was treated. □

Example 10.3.2. Let Ω denote a bounded domain in $\mathbb{R}^2$ with boundary Γ and let $T > 0$. The *two-dimensional Navier-Stokes equations* are given by

$$\varrho\left(u_t + (u\cdot\nabla)u\right) - \nu\Delta u + \nabla p = f \quad \text{in } Q = (0,T)\times\Omega, \tag{10.41a}$$

$$\operatorname{div} u = 0 \quad \text{in } Q, \tag{10.41b}$$

where $\varrho > 0$ is the density of the fluid, $\nu > 0$ is the kinematic viscosity, f represents volume forces and

$$(u\cdot\nabla)u = \left(u_1\frac{\partial u_1}{\partial x_1} + u_2\frac{\partial u_1}{\partial x_2}, u_1\frac{\partial u_2}{\partial x_1} + u_2\frac{\partial u_2}{\partial x_2}\right)^T.$$

The unknowns are the velocity field $u = (u_1, u_2)$ and the pressure p. Together with (10.41) we consider nonslip boundary conditions

$$u = u_d \quad \text{on } \Sigma = (0,T)\times\Gamma \tag{10.41c}$$

and the initial condition

$$u(0,\cdot) = u_0 \quad \text{in } \Omega. \tag{10.41d}$$

In [Tem88, pp. 104-107, 116-117] it was proved that (10.41) can be written in the form (10.18) and that **(A1)** holds provided the boundary Γ is sufficiently smooth. ◇

10.3.2 POD Galerkin Projection of Dynamical Systems

Given a snapshot grid $\{t_j\}_{j=0}^n$ and associated snapshots $y_0, \ldots, y_n$ the space V^ℓ is constructed as described in Section 10.2.2. We obtain the POD-Galerkin surrogate of (10.39) by replacing the space of test functions V by $V^\ell = \operatorname{span}\{\psi_1, \ldots, \psi_\ell\}$, and by using the ansatz

$$Y(t) = \sum_{i=1}^{\ell} \alpha_i(t)\psi_i \tag{10.42}$$

for its solution. The result is a ℓ-dimensional nonlinear dynamical system of ordinary differential equations for the functions α_i $(i = 1, \ldots, \ell)$ of the form

$$M\dot{\alpha} + A\alpha + n(\alpha) = \mathcal{F}, \quad M\alpha(0) = (\langle y_0, \psi_j\rangle_H)_{j=1}^{\ell}, \tag{10.43}$$

where $M = (\langle \psi_i, \psi_j \rangle_H)_{i,j=1}^{\ell}$ and $A = (a(\psi_i, \psi_j))_{i,j=1}^{\ell}$ denote the POD mass and stiffness matrices, $n(\alpha) = (\langle F(Y), \psi_j \rangle_{V',V})_{j=1}^{\ell}$ the nonlinearity, and $\mathcal{F} = (\langle f, \psi_j \rangle_H)_{j=1}^{\ell}$. We note that M is the identity matrix if in $(\mathbf{P}_\ell)$ $X = H$ is chosen.

For the time discretization we choose $m \in \mathbb{N}$ and introduce the time grid

$$0 = \tau_0 < \tau_1 < \ldots < \tau_m = T, \quad \delta\tau_j = \tau_j - \tau_{j-1} \quad \text{for } j = 1, \ldots, m,$$

and set

$$\delta\tau = \min\{\delta\tau_j : 1 \le j \le m\} \quad \text{and} \quad \Delta\tau = \max\{\delta\tau_j : 1 \le j \le m\}.$$

Notice that the snapshot grid and the time grid usually does not coincide. Throughout we assume that $\Delta\tau/\delta\tau$ is bounded uniformly with respect to m. To relate the snapshot grid $\{t_j\}_{j=0}^{n}$ and the time grid $\{\tau_j\}_{j=0}^{m}$ we set for every τ_k, $0 \le k \le m$, an associated index $\bar{k} = \operatorname{argmin}\{|\tau_k - t_j| : 0 \le j \le n\}$ and define $\sigma_n \in \{1, \ldots, n\}$ as the maximum of the occurrence of the same value $t_{\bar{k}}$ as k ranges over $0 \le k \le m$.

The problem consists in finding a sequence $\{Y_k\}_{k=0}^{m}$ in V^ℓ satisfying

$$\langle Y_0, \psi \rangle_H = \langle y_0, \psi \rangle_H \quad \text{for all } \psi \in V^\ell \tag{10.44a}$$

and

$$\langle \overline{\partial}_\tau Y_k, \psi \rangle_H + a(Y_k, \psi) + \langle F(Y_k), \psi \rangle_{V',V} = \langle f(\tau_k), \psi \rangle_H \tag{10.44b}$$

for all $\psi \in V^\ell$ and $k = 1, \ldots, m$, where we have set $\overline{\partial}_\tau Y_k = (Y_k - Y_{k-1})/\delta\tau_k$. Note that (10.44) is a backward Euler scheme for (10.39).

For every $k = 1, \ldots, m$ there exists at least one solution Y_k of (10.44). If $\Delta\tau$ is sufficiently small, the sequence $\{Y_k\}_{k=1}^{m}$ is uniquely determined. A proof was given in [KV02, Theorem 4.2].

10.3.3 Error Estimates

Our next goal is to present an error estimate for the expression

$$\sum_{k=0}^{m} \beta_k \, \|Y_k - y(\tau_k)\|_H^2,$$

where $y(\tau_k)$ is the solution of (10.39) at the time instances $t = \tau_k$, $k = 1, \ldots, m$, and the positive weights β_j are given by

$$\beta_0 = \frac{\delta\tau_1}{2}, \quad \beta_j = \frac{\delta\tau_j + \delta\tau_{j+1}}{2} \text{ for } j = 1, \ldots, m-1, \quad \text{and} \quad \beta_m = \frac{\delta\tau_m}{2}.$$

Let us introduce the orthogonal projection $\mathcal{P}_n^\ell$ of X onto V^ℓ by

$$\mathcal{P}_n^\ell \varphi = \sum_{i=1}^{\ell} \langle \varphi, \psi_i \rangle_X \psi_i \quad \text{for } \varphi \in X. \tag{10.45}$$

In the context of finite element discretizations, $\mathcal{P}_n^\ell$ is called the *Ritz projection*.

Estimate for the Choice $X = V$

Let us choose $X = V$ in the context of Section 10.2.2. Since the Hilbert space V is endowed with the inner product (10.16), the Ritz-projection $\mathcal{P}_n^\ell$ is the orthogonal projection of V on V^ℓ.

We make use of the following assumptions:

(H1) $y \in W^{2,2}(0,T;V)$, *where* $W^{2,2}(0,T;V) = \{\varphi \in L^2(0,T;V) \ : \ \varphi_t, \varphi_{tt} \in L^2(0,T;V)\}$ *is a Hilbert space endowed with its canonical inner product.*

(H2) *There exists a normed linear space W continuously embedded in V and a constant $c_a > 0$ such that $y \in C([0,T];W)$ and*

$$a(\varphi,\psi) \le c_a \ \|\varphi\|_H \|\psi\|_W \quad \textit{for all } \varphi \in V \textit{ and } \psi \in W. \tag{10.46}$$

Example 10.3.3. For $V = H_0^1(\Omega)$, $H = L^2(\Omega)$, with Ω a bounded domain in $\mathbb{R}^l$ and

$$a(\varphi,\psi) = \int_\Omega \nabla\varphi \cdot \nabla\psi \, \mathrm{d}x \text{ for all } \varphi,\psi \in H_0^1(\Omega),$$

choosing $W = H^2(\Omega) \cap H_0^1(\Omega)$ implies $a(\varphi,\psi) \le \|\varphi\|_W \|\psi\|_H$ for all $\varphi \in W$, $\psi \in V$, and (10.46) holds with $c_a = 1$. ◇

Remark 10.3.4. In the case $X = V$ we infer from (10.16) that

$$a(\mathcal{P}_n^\ell\varphi,\psi) = a(\varphi,\psi) \quad \text{for all } \psi \in V^\ell,$$

where $\varphi \in V$. In particular, we have $\|\mathcal{P}_n^\ell\|_{L(V)} = 1$. Moreover, **(H2)** yields

$$\|\mathcal{P}_n^\ell\|_{\mathcal{L}(H)} \le c_P \quad \text{for all } 1 \le \ell \le d$$

where $c_P = c\ell/\lambda_\ell$ (see [KV02, Remark 4.4]) and $c > 0$ depends on y, c_a, and T, but is independent of ℓ and of the eigenvalues λ_i. ◊

The next theorem was proved in [KV02, Theorem 4.7 and Corollary 4.11].

Theorem 10.3.5. *Assume that* **(H1)**, **(H2)** *hold and that $\Delta\tau$ is sufficiently small. Then there exists a constant C depending on T, but independent of the grids $\{t_j\}_{j=0}^n$ and $\{\tau_j\}_{j=0}^m$, such that*

$$\begin{aligned}\sum_{k=0}^m \beta_k \ \|Y_k - y(\tau_k)\|_H^2 &\le C\sigma_n\Delta\tau(\Delta\tau+\Delta t)\|y_{tt}\|^2_{L^2(0,T;V)} \\ &+ C\Big(\sum_{i=\ell+1}^d \Big(\big|\langle\psi_i, y_0\rangle_V\big|^2 + \frac{\sigma_n\Delta\tau}{\delta t}\lambda_i\Big) + \sigma_n\Delta\tau\Delta t\|y_t\|^2_{L^2(0,T;V)}\Big).\end{aligned} \tag{10.47}$$

Remark 10.3.6. a) If we take the snapshot set

$$\tilde{\mathcal{V}} = \operatorname{span}\{y(t_0),\dots,y(t_n)\}$$

instead of $\mathcal{V}$, we obtain instead of (10.47) the following estimate:

$$\begin{aligned}
&\sum_{k=0}^{m} \beta_k \, \|Y_k - y(\tau_k)\|_H^2 \\
&\leq C \sum_{i=\ell+1}^{d} \left(\left| \langle \psi_i, y_0 \rangle_V \right|^2 + \frac{\sigma_n}{\delta t} \Big(\frac{1}{\delta \tau} + \Delta \tau \Big) \lambda_i \right) + C \sigma_n \Delta \tau \Delta t \, \|y_t\|^2_{L^2(0,T;V)} \\
&\quad + C \sigma_n \Delta \tau (\Delta \tau + \Delta t) \|y_{tt}\|^2_{L^2(0,T;H)}
\end{aligned}$$

(compare [KV02, Theorem 4.7]). As we mentioned in Remark 10.2.1 the factor $(\delta t \; \delta \tau)^{-1}$ arises on the right-hand of the estimate. While computations for many concrete situations show that $\sum_{i=\ell+1}^{d} \lambda_i$ is small compared to $\Delta \tau$, the question nevertheless arises whether the term $1/(\delta \tau \delta t)$ can be avoided in the estimates. However, we refer the reader to [HV03, Section 4], where significantly better numerical results were obtained using the snapshot set $\mathcal{V}$ instead of $\tilde{\mathcal{V}}$. We refer also to [LV04], where the computed feedback gain was more stabilizing providing information about the time derivatives was included.

b) If the number of POD elements for the Galerkin scheme coincides with the dimension of $\mathcal{V}$ then the first additive term on the right-hand side disappears. ◊

Asymptotic Estimate

Note that the terms $\{\lambda_i\}_{i=1}^d$, $\{\psi_i\}_{i=1}^d$ and σ_n depend on the time discretization of $[0, T]$ for the snapshots as well as the numerical integration. We address this dependence next. To obtain an estimate that is independent of the spectral values of a specific snapshot set $\{y(t_j)\}_{j=0}^n$ we assume that $y \in W^{2,2}(0, T; V)$, so that in particular **(H1)** holds, and introduce the operator $\mathcal{R} \in \mathcal{L}(V)$ by

$$\mathcal{R} z = \int_0^T \langle z, y(t) \rangle_V y(t) + \langle z, y_t(t) \rangle_V y_t(t) \, \mathrm{d}t \quad \text{for } z \in V. \tag{10.48}$$

Since $y \in W^{2,2}(0, T; V)$ holds, it follows that $\mathcal{R}$ is compact, see, e.g., [KV02, Section 4]. From the Hilbert-Schmidt theorem it follows that there exists a complete orthonormal basis $\{\psi_i^\infty\}_{i \in \mathbb{N}}$ for X and a sequence $\{\lambda_i^\infty\}_{i \in \mathbb{N}}$ of nonnegative real numbers so that

$$\mathcal{R} \psi_i^\infty = \lambda_i^\infty \psi_i^\infty, \quad \lambda_1^\infty \geq \lambda_2^\infty \geq \ldots, \quad \text{and } \lambda_i^\infty \to 0 \text{ as } i \to \infty.$$

The spectrum of $\mathcal{R}$ is a pure point spectra except for possibly 0. Each non-zero eigenvalue of $\mathcal{R}$ has finite multiplicity and 0 is the only possible accumulation point of the spectrum of $\mathcal{R}$, see [Kat80, p. 185]. Let us note that

$$\int_0^T \|y(t)\|_X^2 \, \mathrm{d}t = \sum_{i=1}^{\infty} \lambda_i \quad \text{and} \quad \|y_\circ\|_X^2 = \sum_{i=1}^{\infty} \left| \langle y_\circ, \psi_i \rangle_X \right|^2.$$

Due to the assumption $y \in W^{2,2}(0,T;V)$ we have

$$\lim_{\Delta t \to 0} \|\mathcal{R}_n - \mathcal{R}\|_{\mathcal{L}(V)} = 0,$$

where the operator $\mathcal{R}_n$ was introduced in (10.30). The following theorem was proved in [KV02, Corollary 4.12].

Theorem 10.3.7. *Let all hypothesis of Theorem* 10.3.5 *be satisfied. Let us choose and fix ℓ such that $\lambda_\ell^\infty \neq \lambda_{\ell+1}^\infty$. If $\Delta t = O(\delta\tau)$ and $\Delta\tau = O(\delta t)$ hold, then there exists a constant $C > 0$, independent of ℓ and the grids $\{t_j\}_{j=0}^n$ and $\{\tau_j\}_{j=0}^m$, and a $\overline{\Delta t} > 0$, depending on ℓ, such that*

$$\begin{aligned}\sum_{k=0}^{m} \beta_k \, \|Y_k - y(\tau_k)\|_H^2 \leq C \sum_{i=\ell+1}^{\infty} \left(\left| \langle y_0, \psi_i^\infty \rangle_V \right|^2 + \lambda_i^\infty \right) \\ + C \left(\Delta\tau \Delta t \, \|y_t\|_{L^2(0,T;V)}^2 + \Delta\tau(\Delta\tau + \Delta t) \, \|y_{tt}\|_{L^2(0,T;V)}^2 \right)\end{aligned} \tag{10.49}$$

for all $\Delta t \leq \overline{\Delta t}$.

Remark 10.3.8. In case of $X = H$ the spectral norm of the POD stiffness matrix with the elements $\langle \psi_j, \psi_i \rangle_V$, $1 \leq i, j, \leq d$, arises on the right-hand side of the estimate (10.47); see [KV02, Theorem 4.16]. For this reason, no asymptotic analysis can be done for $X = H$. ◊

10.4 Suboptimal Control of Evolution Problems

In this section we propose a reduced-order approach based on POD for optimal control problems governed by evolution problems. For linear-quadratic optimal control problems we among other things present error estimates for the suboptimal POD solutions.

10.4.1 The Abstract Optimal Control Problem

For $T > 0$ the space $W(0,T)$ is defined as

$$W(0,T) = \left\{ \varphi \in L^2(0,T;V) : \varphi_t \in L^2(0,T;V') \right\},$$

which is a Hilbert space endowed with the common inner product (see, for example, in [DL92, p. 473]). It is well-known that $W(0,T)$ is continuously embedded into $C([0,T];H)$, the space of continuous functions from $[0,T]$ to H, i.e., there exists an embedding constant $c_e > 0$ such that

$$\|\varphi\|_{C([0;T];H)} \leq c_e \, \|\varphi\|_{W(0,T)} \quad \text{for all } \varphi \in W(0,T). \tag{10.50}$$

We consider the abstract problem introduced in Section 10.2.2. Let $\mathcal{U}$ be a Hilbert space which we identify with its dual $\mathcal{U}'$, and let $\mathcal{U}_{\mathsf{ad}} \subset \mathcal{U}$ a closed and

convex subset. For $y_0 \in H$ and $u \in \mathcal{U}_{\mathsf{ad}}$ we consider the nonlinear evolution problem on $[0,T]$

$$\frac{\mathrm{d}}{\mathrm{d}t}\,\langle y(t),\varphi\rangle_H + a(y(t),\varphi) + \langle F(y(t)),\varphi\rangle_{V',V} = \langle (\mathcal{B}u)(t)),\varphi\rangle_{V',V} \quad (10.51a)$$

for all $\varphi \in V$ and

$$y(0) = y_0 \quad \text{in } H, \quad (10.51b)$$

where $\mathcal{B} : \mathcal{U} \to L^2(0,T;V')$ is a continuous linear operator. We suppose that for every $u \in \mathcal{U}_{\mathsf{ad}}$ and $y_0 \in H$ there exists a unique solution y of (10.51) in $W(0,T)$. This is satisfied for many practical situations, including, e.g., the controlled viscous Burgers and two-dimensional incompressible Navier-Stokes equations, see, e.g., [Tem88, Vol01b].

Next we introduce the cost functional $J : W(0,T) \times \mathcal{U} \to \mathbb{R}$ by

$$J(y,u) = \frac{\alpha_1}{2}\,\|\mathcal{C}y - z_1\|^2_{W_1} + \frac{\alpha_2}{2}\,\|\mathcal{D}y(T) - z_2\|^2_{W_2} + \frac{\sigma}{2}\|u\|^2_{\mathcal{U}}, \quad (10.52)$$

where W_1, W_2 are Hilbert spaces and $\mathcal{C} : L^2(0,T;H) \to W_1$ and $\mathcal{D} : H \to W_2$ are bounded linear operators, $z_1 \in W_1$ and $z_2 \in W_2$ are given desired states and $\alpha_1, \alpha_2, \sigma > 0$.

The optimal control problem is given by

$$\min J(y,u) \quad \text{s.t.} \quad (y,u) \in W(0,T) \times \mathcal{U}_{\mathsf{ad}} \text{ solves (10.51).} \quad (\mathbf{CP})$$

In view of Example 10.3.2 a standard discretization (based on, e.g., finite elements) of (**CP**) may lead to a large-scale optimization problem which can not be solved with the currently available computer power. Here we propose a suboptimal solution approach that utilizes POD. The associated suboptimal control problem is obtained by replacing the dynamical system (10.51) in (**CP**) through the POD surrogate model (10.43), using the Ansatz (10.42) for the state. With $\mathcal{F}$ replaced by $(\langle (\mathcal{B}u)(t), \psi_j\rangle_H)_{j=1}^{l}$ it reads

$$\min J(\alpha,u) \quad \text{s.t.} \quad (\alpha,u) \in H^1(0,T)^\ell \times \mathcal{U}_{\mathsf{ad}} \text{ solves (10.43).} \quad (\mathbf{SCP})$$

At this stage the question arises which snapshots to use for the POD surrogate model, since it is by no means clear that the POD model computed with snapshots related to a control u_1 is also able to resolve the presumably completely different dynamics related to a control $u_2 \neq u_1$. To cope with this difficulty we present the following adaptive pseudo-optimization algorithm which is proposed in [AH00, AH01]. It successively updates the snapshot samples on which the the POD surrogate model is to be based upon. Related ideas are presented in [AFS00, Rav00].

Choose a sequence of increasing numbers N_j.

Algorithm 10.4.1 (POD-based adaptive control)

1. Let a set of snapshots $y_i^0, i = 1, \ldots, N_0$ be given and set j=0.

2. Set (or determine) l, and compute the POD modes and the space V^l.
3. Solve the reduced optimization problem (**SCP**) for u^j.
4. Compute the state y^j corresponding to the current control u^j and add the snapshots $y_i^{j+1}, i = N_j+1, \ldots, N_{j+1}$ to the snapshot set $y_i^j, i = 1, \ldots, N_j$.
5. If $|u^{j+1} - u^j|$ is not sufficiently small, set j = j+1 and goto 2.

We note that the term snapshot here may also refer to difference quotients of snapshots, compare Remark 10.2.1. We note further that it is also possible to replace its step 4. by

4.' Compute the state y^j corresponding to the current control u^j and store the snapshots $y_i^{j+1}, i = N_j + 1, \ldots, N_{j+1}$ while the snapshot set $y_i^j, i = 1, \ldots, N_j$ is neglected.

Many numerical investigations on the basis of Algorithm 10.4.1 with step 4' can be found in [Afa02]. This reference also contains a numerical comparison of POD to other model reduction techniques, including their applications to optimal open-loop control.

To anticipate discussion we note that the number N_j of snapshots to be taken in the j-th iteration ideally should be determined during the adaptive optimization process. We further note that the choice of ℓ in step 2 might be based on the information content $\mathcal{E}$ defined in (10.9), compare Section 10.5.2. We will pick up these items again in Section 10.6.

Remark 10.4.1. It is numerically infeasible to compute an optimal closed-loop feedback control strategy based on a finite element discretization of (10.51), since the resulting nonlinear dynamical system in general has large dimension and numerical solution of the related Hamilton-Jacobi-Bellman (HJB) equation is infeasible. In [KVX04] model reduction techniques involving POD are used to numerically construct suboptimal closed-loop controllers using the HJB equations of the reduced order model, which in this case only is low dimensional. ◊

10.4.2 Error Estimates for Linear-Quadratic Optimal Control Problems

It is still an open problem to estimate the error between solutions of (**CP**) and the related suboptimal control problem (**SCP**), and also to prove convergence of Algorithm 10.4.1. As a first step towards we now present error estimates for discrete solutions of linear-quadratic optimal control problems with a POD model as surrogate. For this purpose we combine techniques of [KV01, KV02] and [DH02, DH04, Hin05].

We consider the abstract control problem (**CP**) with $F \equiv 0$ and $\mathcal{U}_{\mathsf{ad}} \equiv \mathcal{U}$. We note that J from (10.52) is twice continuously Fréchet-differentiable. In particular, the second Fréchet-derivative of J at a given point $x = (y, u) \in W(0,T) \times \mathcal{U}$ in a direction $\delta x = (\delta y, \delta u) \in W(0,T) \times \mathcal{U}$ is given by

$$\nabla^2 J(x)(\delta x, \delta x) = \alpha_1 \|\mathcal{C}\delta y\|^2_{W_1} + \alpha_2 \|\mathcal{D}\delta y(T)\|^2_{W_2} + \sigma\|\delta u\|^2_{\mathcal{U}} \geq 0.$$

Thus, $\nabla^2 J(x)$ is a non-negative operator.

The goal is to minimize the cost J subject to (y, u) solves the linear evolution problem

$$\langle y_t(t), \varphi\rangle_H + a(y(t), \varphi) = \langle (\mathcal{B}u)(t), \varphi\rangle_H \tag{10.53a}$$

for all $\varphi \in V$ and almost all $t \in (0, T)$ and

$$y(0) = y_0 \quad \text{in } H. \tag{10.53b}$$

Here, $y_0 \in H$ is a given initial condition. It is well-known that for every $u \in \mathcal{U}$ problem (10.53) admits a unique solution $y \in W(0, T)$ satisfying

$$\|y\|_{W(0,T)} \leq C \left(\|y_0\|_H + \|u\|_{\mathcal{U}}\right)$$

for a constant $C > 0$; see, e.g., [DL92, pp. 512-520]. If, in addition, $y_0 \in V$ and if there exist two constants $c_1, c_2 > 0$ with

$$\langle A\varphi, -\Delta\varphi\rangle_H \geq c_1 \ \|\varphi\|^2_{D(A)} - c_2 \ \|\varphi\|^2_H \quad \text{for all } \varphi \in D(A) \cap V,$$

then we have

$$y \in L^2(0, T; D(A) \cap V) \cap H^1(0, T; H), \tag{10.54}$$

compare [DL92, p. 532]. From (10.54) we infer that y is almost everywhere equal to an element of $C([0, T]; V)$.

The minimization problem, which is under consideration, can be written as a linear-quadratic optimal control problem

$$\min J(y, u) \quad \text{s.t.} \quad (y, u) \in W(0, T) \times \mathcal{U} \text{ solves (10.53).} \tag{\textbf{LQ}}$$

Applying standard arguments one can prove that there exists a unique optimal solution $\bar{x} = (\bar{y}, \bar{u})$ to (**LQ**).

There exists a unique Lagrange-multiplier $\bar{p} \in W(0, T)$ satisfying together with $\bar{x} = (\bar{y}, \bar{u})$ the first-order necessary optimality conditions, which consist in the *state equations* (10.53), in the *adjoint equations*

$$-\langle \bar{p}_t(t), \varphi\rangle_H + a(\bar{p}(t), \varphi) = -\alpha_1 \langle \mathcal{C}^*(\mathcal{C}\bar{y}(t) - z_1(t)), \varphi\rangle_H \tag{10.55a}$$

for all $\varphi \in V$ and almost all $t \in (0, T)$ and

$$\bar{p}(T) = -\alpha_2 \mathcal{D}^*(\mathcal{D}\bar{y}(T) - z_2) \quad \text{in } H, \tag{10.55b}$$

and in the *optimality condition*

$$\sigma\bar{u} - \mathcal{B}^*\bar{p} = 0 \quad \text{in } \mathcal{U}. \tag{10.56}$$

Here, the linear and bounded operators $\mathcal{C}^* : W_1 \to L^2(0,T;H)$, $\mathcal{D}^* : W_2 \to H$, and $\mathcal{B}^* : L^2(0,T;H) \to \mathcal{U}$ stand for the Hilbert space adjoints of $\mathcal{C}$, $\mathcal{D}$, and $\mathcal{B}$, respectively.

Introducing the reduced cost functional

$$\hat{J}(u) = J(y(u), u),$$

where $y(u)$ solves (10.53) for the control $u \in \mathcal{U}$, we can express (**LQ**) as the reduced problem

$$\min \hat{J}(u) \quad \text{s.t.} \quad u \in \mathcal{U}. \tag{$\hat{\mathbf{P}}$}$$

From (10.56) it follows that the gradient of $\hat{J}$ at $\bar{u}$ is given by

$$\hat{J}'(\bar{u}) = \sigma \bar{u} - \mathcal{B}^* \bar{p}. \tag{10.57}$$

Let us define the operator $G : \mathcal{U} \to \mathcal{U}$ by

$$G(u) = \sigma u - \mathcal{B}^* p, \tag{10.58}$$

where $y = y(u)$ solves the state equations with the control $u \in \mathcal{U}$ and $p = p(y(u))$ satisfies the adjoint equations for the state y. As a consequence of (10.56) it follows that the first-order necessary optimality conditions for ($\hat{\mathbf{P}}$) are

$$G(u) = 0 \quad \text{in } \mathcal{U}. \tag{10.59}$$

In the POD context the operator G will be replaced by an operator $G_\ell : \mathcal{U} \to \mathcal{U}$ which then represents the optimality condition of the optimal control problem (**SCP**). The construction of G_ℓ is described in the following.

Computation of the POD Basis

Let $u \in \mathcal{U}$ be a given control for (**LQ**) and $y = y(u)$ the associated state satisfying $y \in C^1([0,T];V)$. To keep the notation simple we apply only a spatial discretization with POD basis functions, but no time integration by, e.g., an implicit Euler method. Therefore, we apply a continuous POD, where we choose $X = V$ in the context of Section 10.2.2. Let us mention the work [HY02], where estimates for POD Galerkin approximations were derived utilizing also a continuous version of POD.

We define the bounded linear $\mathcal{Y} : H^1(0,T;\mathbb{R}) \to V$ by

$$\mathcal{Y}\varphi = \int_0^T \varphi(t) y(t) + \varphi_t(t) y_t(t) \, \mathrm{d}t \quad \text{for } \varphi \in H^1(0,T;\mathbb{R}).$$

Notice that the operator $\mathcal{Y}$ is the continuous variant of the discrete operator $\mathcal{Y}_n$ introduced in (10.28). The adjoint $\mathcal{Y}^* : V \to H^1(0,T;\mathbb{R})$ is given by

$$\big(\mathcal{Y}^* z\big)(t) = \langle z, y(t) + y_t(t) \rangle_V \quad \text{for } z \in V.$$

(compare (10.29)). The operator $\mathcal{R} = \mathcal{Y}\mathcal{Y}^* \in \mathcal{L}(V)$ is already introduced in (10.48).

Remark 10.4.2. Analogous to the theory of singular value decomposition for matrices, we find that the operator $\mathcal{K} = \mathcal{Y}^*\mathcal{Y} \in \mathcal{L}(H^1(0,T;\mathbb{R}))$ given by

$$\big(\mathcal{K}\varphi\big)(t) = \int_0^T \langle y(s), y(t)\rangle_V \varphi(s) + \langle y_t(s), y_t(t)\rangle_V \varphi_t(s)\,\mathrm{d}s \quad \text{for } \varphi \in H^1(0,T;\mathbb{R})$$

has the eigenvalues $\{\lambda_i^\infty\}_{i=1}^\infty$ and the eigenfunctions

$$v_i^\infty(t) = \frac{1}{\sqrt{\lambda_i^\infty}}\,\big(\mathcal{Y}^*\psi_i^\infty\big)(t) = \frac{1}{\sqrt{\lambda_i^\infty}}\,\langle \psi_i^\infty, y(t) + y_t(t)\rangle_V$$

for $i \in \{j \in \mathbb{N} : \lambda_j^\infty > 0\}$ and almost all $t \in [0,T]$. ◊

In the following theorem we formulate properties of the eigenvalues and eigenfunctions of $\mathcal{R}$. For a proof we refer to [HLB96], for instance.

Theorem 10.4.3. *For every $\ell \in \mathbb{N}$ the eigenfunctions $\psi_1^\infty, \ldots, \psi_\ell^\infty \in V$ solve the minimization problem*

$$\min \mathfrak{J}(\psi_1, \ldots, \psi_\ell) \quad \textit{s.t.} \quad \langle \psi_j, \psi_i\rangle_X = \delta_{ij} \quad \textit{for } 1 \le i, j \le \ell, \tag{10.60}$$

where the cost functional $\mathfrak{J}$ is given by

$$\mathfrak{J}(\psi, \ldots, \psi_\ell) = \int_0^T \Big\| y(t) - \sum_{i=1}^\ell \langle y(t), \psi_i\rangle_V \psi_i \Big\|_X^2 + \Big\| y_t(t) - \sum_{i=1}^\ell \langle y_t(t), \psi_i\rangle_V \psi_i \Big\|_V^2 \,\mathrm{d}t.$$

Moreover, the eigenfunctions $\{\lambda_i^\infty\}_{i\in\mathbb{N}}$ and eigenfunctions $\{\psi_i^\infty\}_{i\in\mathbb{N}}$ of $\mathcal{R}$ satisfy the formula

$$\mathfrak{J}(\psi_1^\infty, \ldots, \psi_\ell^\infty) = \sum_{i=\ell+1}^\infty \lambda_i^\infty. \tag{10.61}$$

Proof. The proof of the theorem relies on the fact that the eigenvalue problem

$$\mathcal{R}\psi_i^\infty = \lambda_i^\infty \psi_i^\infty \quad \text{for } i = 1, \ldots, \ell$$

is the first-order necessary optimality condition for (10.60). For more details we refer the reader to [HLB96].

Galerkin POD Approximation

Let us introduce the set $V^\ell = \text{span}\,\{\psi_1^\infty, \ldots, \psi_\ell^\infty\} \subset V$. To study the POD approximation of the operator G we introduce the orthogonal projection $\mathcal{P}^\ell$ of V onto V^ℓ by

$$\mathcal{P}^\ell\varphi = \sum_{i=1}^\ell \langle \varphi, \psi_i^\infty\rangle_V \psi_i^\infty \quad \text{for } \varphi \in V. \tag{10.62}$$

(compare (10.45)). Note that

$$\begin{aligned}&\mathfrak{J}(\psi,\dots,\psi_\ell)\\&\quad=\int_0^T \left\|y(t)-\mathcal{P}^\ell y(t)\right\|_V^2+\left\|y_t(t)-\mathcal{P}^\ell y_t(t)\right\|_V^2\,\mathrm{d}t=\sum_{i=\ell+1}^{\infty}\lambda_i^\infty.\end{aligned}\tag{10.63}$$

From (10.16) it follows directly that

$$a(\mathcal{P}^\ell\varphi,\psi)=a(\varphi,\psi)\quad\text{for all }\psi\in V^\ell,$$

where $\varphi\in V$. Clearly, we have $\|\mathcal{P}^\ell\|_{\mathcal{L}(V)}=1$.

Next we define the approximation $G_\ell:\mathcal{U}\to\mathcal{U}$ of the operator G by

$$G_\ell(u)=\sigma u-\mathcal{B}^*p^\ell,\tag{10.64}$$

where $p^\ell\in W(0,T)$ is the solution to

$$-\langle p_t^\ell(t),\psi\rangle_H+a(p^\ell(t),\psi)=-\alpha_1\,\langle\mathcal{C}^*(\mathcal{C}y^\ell-z_1),\psi\rangle_H\tag{10.65a}$$

for all $\psi\in V^\ell$ and $t\in(0,T)$ a.e. and

$$p^\ell(T)=-\alpha_2\,\mathcal{P}^\ell\big(\mathcal{D}^*(\mathcal{D}y^\ell(T)-z_2)\big)\tag{10.65b}$$

and $y^\ell\in W(0,T)$, which solves

$$\langle y_t^\ell(t),\psi\rangle_H+a(y^\ell(t),\psi)=\langle(\mathcal{B}u)(t),\psi\rangle_H\tag{10.66a}$$

for all $\psi\in V^\ell$ and almost all $t\in(0,T)$ and

$$y^\ell(0)=\mathcal{P}^\ell y_0\tag{10.66b}$$

Notice that $G_\ell(u)=0$ are the first-order optimality conditions for the optimal control problem

$$\min\hat{J}^\ell(u)\quad\text{s.t.}\quad u\in\mathcal{U},$$

where $\hat{J}^\ell(u)=J(y^\ell(u),u)$ and $y^\ell(u)$ denotes the solution to (10.66).

It follows from standard arguments (Lax-Milgram lemma) that the operator G_ℓ is well-defined. Furthermore we have

Theorem 10.4.4. *The equation*

$$G_\ell(u)=0\quad\textit{in }\mathcal{U}\tag{10.67}$$

admits a unique solution $u^\ell\in\mathcal{U}$ which together with the unique solution u of (10.59) *satisfies the estimate*

$$\|u-u^\ell\|_{\mathcal{U}}\le\frac{1}{\sigma}\Big(\|\mathcal{B}^*(P-P^\ell)\mathcal{B}u\|_{\mathcal{U}}+\|\mathcal{B}^*(S^*-S_\ell^*)\mathcal{C}^*z_1\|_{\mathcal{U}}\Big).\tag{10.68}$$

Here, $P:=S^*\mathcal{C}^*\mathcal{C}S$, $P^\ell:=S_\ell^*\mathcal{C}^*\mathcal{C}S_\ell$, with S,S_l denoting the solution operators in (10.53) *and* (10.66)*, respectively.*

A proof of this theorem immediately follows from the fact, that u^l is a admissible test function in (10.59), and u in (10.67). Details will be given in [HV05].

Remark 10.4.5. We note, that Theorem 10.4.4 remains also valid in the situation where admissible controls are taken from a closed convex subset $\mathcal{U}_{\mathsf{ad}} \subset \mathcal{U}$. The solutions u, u^l in this case satisfy the variational inequalities

$$\langle G(u), v - u\rangle_{\mathcal{U}} \geq 0 \quad \text{for all } v \in \mathcal{U}_{\mathsf{ad}},$$

and

$$\langle G_\ell(u^\ell), v - u^\ell\rangle_{\mathcal{U}} \geq 0 \quad \text{for all } v \in \mathcal{U}_{\mathsf{ad}},$$

so that adding the first inequality with $v = u^\ell$ and the second with $v = u$ and straightforward estimation finally give (10.68) also in the present case. The crucial point here is that the set of admissible controls is not discretized a-priori. The discretization of the optimal control u^ℓ is determined by that of the corresponding Lagrange multiplier p^ℓ. For details of this discrete concept we refer to [Hin05].

It follows from the structure of estimate (10.68), that error estimates for $y - y^\ell$ and $p - p^l$ directly lead to an error estimate for $u - u^\ell$.

Proposition 10.4.6. *Let $\ell \in \mathbb{N}$ with $\lambda_\ell^\infty > 0$ be fixed, $u \in \mathcal{U}$ and $y = y(u)$ and $p = p(y(u))$ the corresponding solutions of the state equations* (10.53) *and adjoint equations* (10.55) *respectively. Suppose that the POD basis of rank ℓ is computed by using the snapshots $\{y(t_j)\}_{j=0}^n$ and its difference quotients. Then there exist constants $c_y, c_p > 0$ such that*

$$\|y^\ell - y\|^2_{L^\infty(0,T;H)} + \|y^\ell - y\|^2_{L^2(0,T;V)} \leq c_y \sum_{i=\ell+1}^{\infty} \lambda_i^\infty \tag{10.69}$$

and

$$\begin{aligned} &\|p^\ell - p\|^2_{L^2(0,T;V)} \\ &\quad \leq c_p \Bigg(\sum_{i=\ell+1}^{\infty} \lambda_i^\infty + \|\mathcal{P}^\ell p - p\|^2_{L^2(0,T;V)} + \|\mathcal{P}^\ell p_t - p_t\|^2_{L^2(0,T;V)} \Bigg), \end{aligned} \tag{10.70}$$

where y^ℓ and p^ℓ solve (10.66) *and* (10.65)*, respectively, for the chosen u inserted in* (10.66a)*.*

Proof. Let

$$y^\ell(t) - y(t) = y^\ell(t) - \mathcal{P}^\ell y(t) + \mathcal{P}^\ell y(t) - y(t) = \vartheta(t) + \varrho(t),$$

where $\vartheta = y^\ell - \mathcal{P}^\ell y$ and $\varrho = \mathcal{P}^\ell y - y$. From (10.16), (10.62), (10.63) and the continuous embedding $H^1(0,T;V) \hookrightarrow L^\infty(0,T;H)$ we find

$$\|\varrho\|^2_{L^\infty(0,T;H)} + \|\varrho\|^2_{L^2(0,T;V)} \le c_E \sum_{i=\ell+1}^{\infty} \lambda_i^\infty \tag{10.71}$$

with an embedding constant $c_E > 0$. Utilizing (10.53) and (10.66) we obtain

$$\langle \vartheta_t(t), \psi \rangle_H + a(\vartheta(t), \psi) = \langle y_t(t) - \mathcal{P}^\ell y_t(t), \psi \rangle_H$$

for all $\psi \in V^\ell$ and almost all $t \in (0,T)$. From (10.16), (10.17) and Young's inequality it follows that

$$\frac{\mathrm{d}}{\mathrm{d}t} \|\vartheta(t)\|^2_H + \|\vartheta(t)\|^2_V \le c_V^2 \, \|y_t(t) - \mathcal{P}^\ell y_t(t)\|^2_V. \tag{10.72}$$

Due to (10.66b) we have $\vartheta(0) = 0$. Integrating (10.72) over the interval $(0,t)$, $t \in (0,T]$, and utilizing (10.37), (10.45) and (10.63) we arrive at

$$\|\vartheta(t)\|^2_H + \int_0^t \|\vartheta(s)\|^2_V \, \mathrm{d}s \le c_V^2 \sum_{i=\ell+1}^{\infty} \lambda_i^\infty$$

for almost all $t \in (0,T)$. Thus,

$$\operatorname*{ess\,sup}_{t\in[0,T]} \|\vartheta(t)\|^2_H + \int_0^T \|\vartheta(s)\|^2_V \, \mathrm{d}s \le c_V^2 \sum_{i=\ell+1}^{\infty} \lambda_i^\infty. \tag{10.73}$$

Estimates (10.71) and (10.73) imply the existence of a constant $c_y > 0$ such that (10.69) holds. We proceed by estimating the error arising from the discretization of the adjoint equations and write

$$p^\ell(t) - p(t) = p^\ell(t) - \mathcal{P}^\ell p(t) + \mathcal{P}^\ell p(t) - p(t) = \theta(t) + \rho(t),$$

where $\theta = p^\ell - \mathcal{P}^\ell p$ and $\rho = \mathcal{P}^\ell p - p$. From (10.16), (10.50), and (10.65b) we get

$$\begin{aligned} \|\theta(T)\|^2_H &\le \alpha_2^2 \, \|\mathcal{D}\|^2_{\mathcal{L}(H,W_1)} \, \|y^\ell(T) - y(T)\|^2_H \\ &\le \alpha_2^2 \, \|\mathcal{D}\|^2_{\mathcal{L}(H,W_1)} \|y^\ell - y\|^2_{C([0,T];H)}. \end{aligned}$$

Thus, applying (10.50), (10.69) and the techniques used above for the state equations, we obtain

$$\begin{aligned} &\operatorname*{ess\,sup}_{t\in[0,T]} \|\theta(t)\|^2_H + \int_0^T \|\theta(s)\|^2_V \, \mathrm{d}s \\ &\quad \le 2c_V^2 \Big(c_V^2 c_e^2 c_y \, \|\mathcal{D}\|^4_{\mathcal{L}(H,W_1)} \sum_{i=\ell+1}^{\infty} \lambda_i^\infty + \|p_t - \mathcal{P}^\ell p_t\|^2_{L^2(0,T;V)} \Big). \end{aligned}$$

Hence, there exists a constant $c_p > 0$ satisfying (10.70).

Remark 10.4.7.

a) The error in the discretization of the state variable is only bounded by the sum over the not modeled eigenvalues λ_i^∞ for $i > \ell$. Since the POD basis is not computed utilizing adjoint information, the term $\mathcal{P}^\ell p - p$ in the $H^1(0,T;V)$-norm arises in the error estimate for the adjoint variables. For POD based approximation of partial differential equations one cannot rely on results clarifying the approximation properties of the POD-subspaces to elements in function spaces as e.g. L^p or C. Such results are an essential building block for e.g. finite element approximations to partial differential equations.

b) If we have already computed a second POD basis of rank $\tilde{\ell} \in \mathbb{N}$ for the adjoint variable, then we can express the term involving the difference $\mathcal{P}^{\tilde{\ell}} p - p$ by the sum over the eigenvalues corresponding to eigenfunctions, which are not used as POD basis functions in the discretization.

c) Recall that $\{\psi_i^\infty\}_{i\in\mathbb{N}}$ is a basis of V. Thus we have

$$\int_0^T \|p(t) - \mathcal{P}^\ell p(t)\|_V^2 \,\mathrm{d}t \le \int_0^T \sum_{i=\ell+1}^{\infty} |a(p(t), \psi_i^\infty)|^2 \,\mathrm{d}t.$$

The sum on the right-hand side converges to zero as ℓ tends to ∞. However, usually we do not have a rate of convergence result available. In numerical applications we can evaluate $\|p - \mathcal{P}^\ell p\|_{L^2(0,T;V)}$. If the term is large then we should increase ℓ and include more eigenfunctions in our POD basis.

d) For the choice $X = H$ we have instead of (10.71) the estimate

$$\|\varrho\|^2_{L^\infty(0,T;H)} + \|\varrho\|^2_{L^2(0,T;V)} \le C\, \|S\|_2 \sum_{i=\ell+1}^{\infty} \lambda_i^\infty,$$

where C is a positive constant, S denotes the stiffness matrix with the elements $S_{ij} = \langle \psi_j^\infty, \psi_i^\infty \rangle_V$, $1 \le i, j \le \ell$, and $\|\cdot\|_2$ stands the spectral norm for symmetric matrices, see [KV02, Lemma 4.15]. ◊

Applying (10.58), (10.64), and Proposition 10.4.6 we obtain for every $u \in \mathcal{U}$

$$\begin{aligned} &\|G_\ell(u) - G(u)\|^2_{\mathcal{U}} \\ &\quad \le c_G \Big(\sum_{i=\ell+1}^{\infty} \lambda_i^\infty + \|\mathcal{P}^\ell p - p\|^2_{L^2(0,T;H)} + \|\mathcal{P}^\ell p_t - p_t\|^2_{L^2(0,T;H)} \Big) \end{aligned} \tag{10.74}$$

for a constant $c_G > 0$ depending on c_λ and $\mathcal{B}$.

Suppose that $u_1, u_2 \in \mathcal{U}$ are given and that $y_1^\ell = y_1^\ell(u_1)$ and $y_2^\ell = y_2^\ell(u_2)$ are the corresponding solutions of (10.66). Utilizing Young's inequality it follows that there exists a constant $c_V > 0$ such that

$$\begin{aligned}\|y_1^\ell - y_2^\ell\|^2_{L^\infty(0,T;H)} + \|y_1^\ell - y_2^\ell\|^2_{L^2(0,T;V)} \\ \le c_V^2 \ \|\mathcal{B}\|^2_{\mathcal{L}(\mathcal{U},L^2(0,T;H))} \ \|u_1 - u_2\|^2_{\mathcal{U}}.\end{aligned} \tag{10.75}$$

Hence, we conclude from (10.65) and (10.75) that

$$\begin{aligned}&\|p_1^\ell - p_2^\ell\|^2_{L^\infty(0,T;H)} + \|p_1^\ell - p_2^\ell\|^2_{L^2(0,T;V)} \\ &\quad \le \max\left(\alpha_1 c_V^2 \ \|\mathcal{C}\|^2_{\mathcal{L}(L^2(0,T;H),W_1)}, \alpha_2 \ \|\mathcal{D}\|^2_{\mathcal{L}(H,W_2)}\right)\cdot \\ &\quad \cdot \left(\|y_1^\ell - y_2^\ell\|^2_{L^\infty(0,T;H)} + \|y_1^\ell - y_2^\ell\|^2_{L^2(0,T;V)}\right) \\ &\quad \le C \ \|u_1 - u_2\|^2_{\mathcal{U}}.\end{aligned} \tag{10.76}$$

where

$$C = \frac{c_V^4}{2} \ \|\mathcal{B}\|^2_{\mathcal{L}(\mathcal{U},L^2(0,T;H))} \max\left(\alpha_1^2 c_V^2 \ \|\mathcal{C}\|^2_{\mathcal{L}(L^2(0,T;H),W_1)}, \alpha_2 \ \|\mathcal{D}\|^2_{\mathcal{L}(H,W_2)}\right).$$

If the POD basis of rank ℓ is computed for the control u_1, then (10.64), (10.74) and (10.76) lead to the existence of a constant $\hat{C} > 0$ satisfying

$$\begin{aligned}&\|G_\ell(u_2) - G(u_1)\|^2_{\mathcal{U}} \le 2 \ \|G_\ell(u_2) - G_\ell(u_1)\|^2_{\mathcal{U}} + 2 \ \|G_\ell(u_1) - G(u_1)\|^2_{\mathcal{U}} \\ &\le \hat{C} \ \|u_2 - u_1\|^2_{\mathcal{U}} \\ &\quad + \hat{C} \left(\sum_{i=\ell+1}^{\infty} \lambda_i^\infty + \|\mathcal{P}^\ell p_1 - p_1\|^2_{L^2(0,T;V)} + \|\mathcal{P}^\ell (p_1)_t - (p_1)_t\|^2_{L^2(0,T;V)}\right).\end{aligned}$$

Hence, $G_\ell(u_2)$ is close to $G(u_1)$ in the $\mathcal{U}$-norm provided the terms $\|u_1 - u_2\|_{\mathcal{U}}$ and $\sum_{i=\ell+1}^{\infty} \lambda_i^\infty$ are small and provided the ℓ POD basis functions $\psi_1^\infty, \ldots, \psi_\ell^\infty$ leads to a good approximation of the adjoint variable p_1 in the $H^1(0,T;V)$-norm. In particular, $G_\ell(u)$ in this case is small, if u denotes the unique optimal control of the continuous control problem, i.e., the solution of $G(u) = 0$.

We further have that both, G and G_ℓ are Fréchet differentiable with constant derivatives $G' \equiv \sigma Id - \mathcal{B}^* p'$ and $G'_\ell \equiv \sigma Id - \mathcal{B}^*(p^\ell)'$. Moreover, since $-\mathcal{B}^* p'$ and $-\mathcal{B}^* p'$ are selfadjoint positive operators, G' and G'_ℓ are invertible, satisfying

$$\|(G')^{-1}\|_{\mathcal{L}(U)}, \|(G'_\ell)^{-1}\|_{\mathcal{L}(\mathcal{U})} \le \frac{1}{\sigma}.$$

Since G_l also is Lipschitz continuous with some positive constant K we now may argue with a Newton-Kantorovich argument [D85, Theorem 15.6] that the equation

$$G_\ell(v) = 0 \quad \text{in } \mathcal{U}$$

admits a unique solution in $u^\ell \in B_{2\epsilon}(u)$, provided

$$\|(G'_\ell)^{-1} G_\ell(u)\|_{\mathcal{U}} \le \epsilon \quad \text{and} \quad \frac{2K\epsilon}{\sigma} < 1.$$

Thus, we in a different fashion again proved existence of a unique solution u^l of (10.67), compare Theorem 10.4.4, and also provided an error estimate for $u-u^l$ in terms of $\sum_{i=\ell+1}^{\infty}\lambda_i^{\infty}+\|\mathcal{P}^\ell p_1-p_1\|^2_{L^2(0,T;V)}+\|\mathcal{P}^\ell(p_1)_t-(p_1)_t\|^2_{L^2(0,T;V)}$.

We close this section with noting that existence and local uniqueness of discrete solutions u^ℓ may be proved following the lines above also in the nonlinear case , i.e., in the case $F \neq 0$ in (10.51).

10.5 Navier-Stokes Control Using POD Surrogate Models

In the present section we demonstrate the potential of the POD method applied as suboptimal open-loop control method for the example of the Navier-Stokes system in (10.41a)-(10.41d) as subsidiary condition in control problem (**CP**).

10.5.1 Setting

We present two numerical examples. The flow configuration is taken as flow around a circular cylinder in 2 spatial dimensions and is depicted in Figure 10.1 for Example 10.5.2, compare the benchmark of Schäfer and Turek in [ST96], and in Figure 10.8 for Example 10.5.3. At the inlet and at the up-

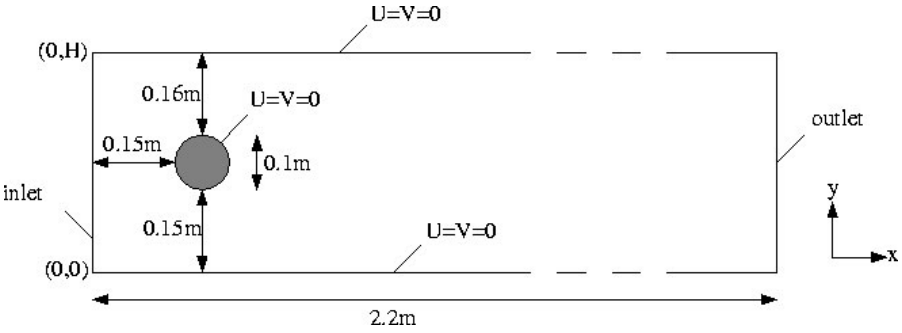


Fig. 10.1. Flow configuration for Example 10.5.2

per and lower boundaries inhomogeneous Dirichlet conditions are prescribed, and at the outlet the so called 'do-nothing' boundary conditions are used [HRT96]. As a consequence the boundary conditions for the Navier-Stokes equations have to be suitably modified. The control objective is to track the Navier Stokes flow to some pre-specified flow field z, which in our numerical experiments is either taken as Stokes flow or mean of snapshots. As control we take distributed forces in the spatial domain. Thus, the optimal control problem in the primitive setting is given by

$$\left.\begin{aligned}
&\min_{(y,u)\in W\times\mathcal{U}} J(y,u) := \frac{1}{2}\int_0^T\int_\Omega |y-z|^2\,\mathrm{d}x\mathrm{d}t + \frac{\alpha}{2}\int_0^T\int_\Omega |u|^2\,\mathrm{d}x\mathrm{d}t\\
&\text{subject to}\\
&y_t + (y\cdot\nabla)y - \nu\Delta y + \nabla p = \mathcal{B}u \quad \text{in } Q = (0,T)\times\Omega,\\
&\operatorname{div} y = 0 \qquad\qquad\qquad\qquad \text{in } Q,\\
&y(t,\cdot) = y_d \qquad\qquad\qquad\qquad \text{on } (0,T)\times\Gamma_d,\\
&\nu\partial_\eta y(t,\cdot) = p\eta \qquad\qquad\qquad\qquad \text{on } (0,T)\times\Gamma_{out},\\
&y(0,\cdot) = y_0 \qquad\qquad\qquad\qquad \text{in } \Omega,
\end{aligned}\right\} \tag{10.77}$$

where $Q := \Omega\times(0,T)$ denotes the time-space cylinder, Γ_d the Dirichlet boundary at the inlet and Γ_{out} the outflow boundary. In this example the volume for the flow measurements and the control volume for the application of the volume forces each cover the whole spatial domain, i.e. $\mathcal{B}$ denotes the injection from $L^2(Q)$ into $L^2(0,T;V')$, $W_1 := L^2(Q)$ and $\mathcal{C} \equiv Id$. Further we have $\mathcal{U}_{ad} = \mathcal{U} = L^2(Q)$, $\alpha_1 = \frac{1}{2}$, $\alpha_2 = 0$, and $\sigma = \alpha$. Since we are interested in open-loop control strategies it is certainly feasible to use the whole of Q as observation domain (use as much information as attainable). Furthermore, from the practical point of view distributed control in the whole domain may be realized by Lorentz forces if the fluid is a electro-magnetically conductive, say [BGGBW97]. From the numerical standpoint this case can present difficulties, since the inhomogeneities in the primal and adjoint equations are large.

We note that it is an open problem to prove existence of global smooth solutions in two space dimensions for the instationary Navier-Stokes equations with do-nothing boundary conditions [Ran00].

The weak formulation of the Navier-Stokes system in (10.77) in primitive variables reads: Given $u \in \mathcal{U}$ and $y_0 \in H$, find $p(t) \in L^2(\Omega)$, $y(t) \in H^1(\Omega)^2$ such that $y(0) = y_0$, and

$$\begin{aligned}
\nu\,\langle\nabla y, \nabla\phi\rangle_H + \langle y_t + y\cdot\nabla y, \phi\rangle_H - \langle p, \operatorname{div}\phi\rangle = \langle\mathcal{B}u,\phi\rangle_H \text{ for all } \phi\in V,\\
\langle\chi, \operatorname{div} y\rangle_H = 0 \text{ for all } \chi\in L^2(\Omega),
\end{aligned} \tag{10.78}$$

holds a.e. in $(0,T)$, where $V := \{\phi\in H^1(\Omega)^2, \phi_{\Gamma_D} = 0\}$, compare [HRT96].

The Reynolds number Re$= 1/\nu$ for the configurations used in our numerical studies is determined by

$$\mathrm{Re} = \frac{\bar{U}d}{\mu},$$

with $\bar{U}$ denoting the bulk velocity at the inlet, d the diameter of the cylinder, μ the molecular viscosity of the fluid and $\rho = 1$.

We now present two numerical examples. The first example presents a detailed description of the POD method as suboptimal control strategy in flow control. In the first step, the POD model for a particular control is validated against the full Navier-Stokes dynamics, and in the second step Algorithm 10.4.1 successfully is applied to compute suboptimal open-loop controls. The

flow configuration is taken from [ST96]. The second example presents optimization results of Algorithm 10.4.1 for an open flow.

10.5.2 Example 1

In the first numerical experiment to be presented we choose a parabolic inflow profile at the inlet, homogeneous Dirichlet boundary conditions at upper and lower boundary, $d = 1$, Re=100 and the channel length is $l = 20d$. For the spatial discretization the Taylor-Hood finite elements on a grid with 7808 triangles, 16000 velocity and 4096 pressure nodes are used. As time interval in (10.77) we use $[0, T]$ with $T = 3.4$ which coincides with the length of one period of the wake flow. The time discretization is carried out by a fractional step Θ-scheme [Bän91] or a semi-implicit Euler-scheme on a grid containing $n = 500$ points. This corresponds to a time step size of $\delta t = 0.0068$. The total number of variables in the optimization problem (10.77) therefore is of order 5.4×10^7 (primal, adjoint and control variables). Subsequently we present a suboptimal approach based on POD in order to obtain suboptimal solutions to (10.77).

Construction and Validation of the POD Model

The reduced-order approach to optimal control problems such as (**CP**) or, in particular, (10.77) is based on approximating the nonlinear dynamics by a Galerkin technique utilizing basis functions that contain characteristics of the controlled dynamics. Since the optimal control is unknown, we apply a heuristic (see [AH01, AFS00]), which is well tested for optimal control problems, in particular for nonlinear boundary control of the heat equation, see [DV01].

Here we use the snapshot variant of POD introduced by Sirovich in [Sir87] to obtain a low-dimensional approximation of the Navier-Stokes equations. To describe the model reduction let $y^1, \ldots, y^m$ denote an ensemble of snapshots of the flow corresponding to different time instances which for simplicity are taken on an equidistant snapshot grid over the time horizon $[0, T]$. For the approximated flow we make the ansatz

$$y = \bar{y} + \sum_{i=1}^{m} \alpha_i \Phi_i \tag{10.79}$$

with modes Φ_i that are obtained as follows (compare Section 10.2.2):

1. Compute the mean $\bar{y} = \frac{1}{m} \sum_{i=1}^{m} y^i$.
2. Build the correlation matrix $K = k_{ij}$, $k_{ij} = \int_\Omega (y^i - \bar{y})(y^j - \bar{y})\, \mathrm{d}x$.
3. Compute the eigenvalues $\lambda_1, \ldots, \lambda_m$ and eigenvectors $v^1, \ldots, v^m$ of K.
4. Set $\Phi_i := \sum_{j=1}^{m} v_j^i (y^j - \bar{y})$, $1 \le i \le d$.

5. Normalize $\Phi_i = \frac{\Phi_i}{\|\Phi_i\|_{L^2(\Omega)}}$, $1 \le i \le d$.

The modes Φ_i are pairwise orthonormal and are optimal with respect to the L^2 inner product in the sense that no other basis of $D := \mathrm{span}\{y_1 - \bar{y}, \ldots, y_m - \bar{y}\}$ can contain more energy in fewer elements, compare Proposition 10.2.5 with $X = H$. We note that the term energy is meaningful in this context, since the vectors y are related to flow velocities. If one would be interested in modes which are optimal w.r.t. enstrophy, say, the H^1-norm should be used instead of the L^2-norm in step 2 above.

The Ansatz (10.79) is commonly used for model reduction in fluid dynamics. The theory of Sections 10.2,10.3 also applies to this situation.

In order to obtain a low-dimensional basis for the Galerkin Ansatz modes corresponding to small eigenvalues are neglected. To make this idea more precise let $D^M := \mathrm{span}\{\Phi_1, \ldots, \Phi_M\}$ $(1 \le M \le N := \dim D)$ and define the relative information content of this basis by

$$\mathrm{I}(M) \;:=\; \sum_{k=1}^{M} \lambda_k \,/\, \sum_{k=1}^{N} \lambda_k,$$

compare (10.9). If the basis is required to describe $\gamma\%$ of the total information contained in the space D, then the dimension M of the subspace D^M is determined by

$$M = \mathrm{argmin}\left\{\mathrm{I}(M) : \mathrm{I}(M) \ge \frac{\gamma}{100}\right\}. \tag{10.80}$$

The reduced dynamical system is obtained by inserting (10.79) into the Navier-Stokes system and using a subspace D^M containing sufficient information as test space. Since all functions Φ_i are solenoidal by construction this results in

$$\langle y_t, \Phi_j\rangle_H + \nu \langle \nabla y, \nabla \Phi_j\rangle_H + \langle (y \cdot \nabla) y, \Phi_j\rangle_H = \langle \mathcal{B}u, \Phi_j\rangle \quad (1 \le j \le M),$$

which may be rewritten as

$$\dot{\alpha} + A\alpha = n(\alpha) + \beta + r, \quad \alpha(0) = a_0, \tag{10.81}$$

compare (10.43). Here, $\langle \cdot, \cdot \rangle$ denotes the $L^2 \times L^2$ inner product. The components of a_0 are computed from $\bar{y} + \sum_{k=1}^{M} (y_0 - \bar{y}, \Phi_k)\Phi_k$. The matrix A is the POD stiffness matrix and the inhomogeneity r results from the contribution of the mean $\bar{y}$ to the ansatz in (10.79). For the entries of β we obtain

$$\beta_j = \langle \mathcal{B}u, \Phi_j\rangle,$$

i.e. the control variable is not discretized. However, we note that it is also feasible to make an Ansatz for the control.

To validate the model in (10.81) we set $u \equiv 0$ and take as initial condition y_0 the uncontrolled wake flow at Re=100. In Figure 10.2 a comparison of the

full Navier-Stokes dynamics and the reduced order model based on 50 (left) as well as on 100 snapshots (right) is presented. As one can see the reduced order model based on 50 snapshots already provides a very good approximation of the full Navier-Stokes dynamics. In Figure 10.3 the long-term behavior of

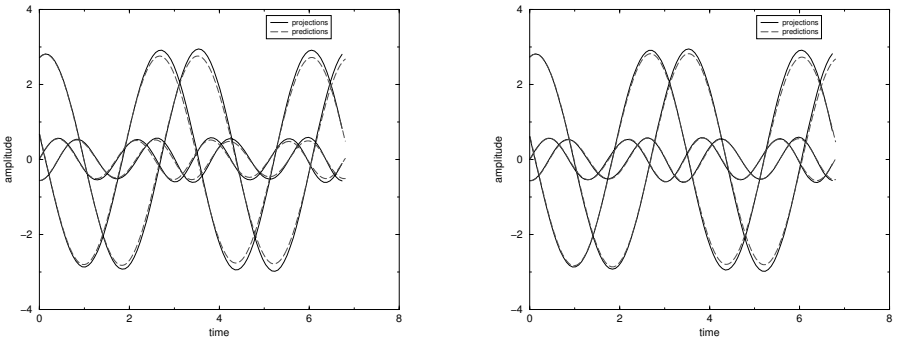


Fig. 10.2. Evolution of $\alpha_i(t)$ compared to that of $(y(t)-\bar{y}, \Phi_i)$ for $i = 1, \ldots, 4$. Left 50 snapshots, right 100 snapshots

the reduced order model based on 100 snapshots for different dimensions of the reduced order model are presented. Graphically the dynamics are already recovered utilizing eight modes. Note, that the time horizon shown in this figure is $[34, 44]$ while the snapshots are taken only in the interval $[0, 3.4]$. Finally, in Figure 10.4 the vorticities of the first ten modes generated from the uncontrolled snapshots are presented. Thus, the reduced order model obtained by snapshot POD captures the essential features of the full Navier-Stokes system, and in a next step may serve as surrogate of the full Navier-Stokes system in the optimization problem (10.77).

Optimization with the POD Model

The reduced optimization problem corresponding to (10.77) is obtained by plugging (10.79) into the cost functional and utilizing the reduced dynamical system (10.81) as constraint in the optimization process. Altogether we obtain

$$(\mathrm{ROM}) \begin{cases} \min \tilde{J}(\alpha, u) = J(y, u) \\ \text{s.t.} \\ \dot{\alpha} + A\alpha = n(\alpha) + \beta + r, \quad \alpha(0) = a_0. \end{cases} \tag{10.82}$$

At this stage we recall that the flow dynamics strongly depends on the control u, and it is not clear at all from which kind of dynamics snapshots should be taken in order to compute an approximation of a solution u^* of (10.77). For

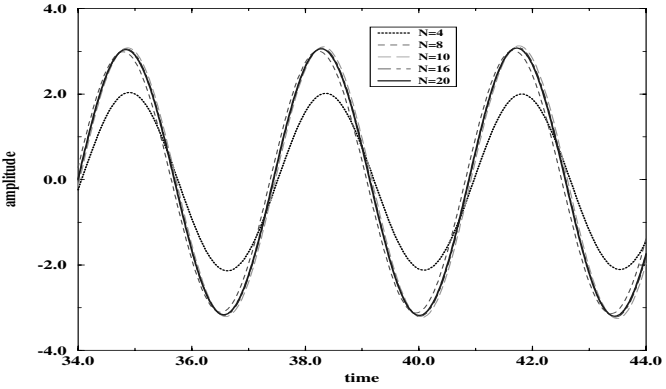


Fig. 10.3. Development of amplitude $\alpha_1(t)$ for varying number N of snapshots

the present examples we apply Algorithms 10.4.1 with a sequence of increasing numbers N_j, where in step 2 the dimension of the space D^M, i.e. the value of M, for a given value $\gamma \in (0, 1]$ is chosen according to (10.80).

In the present application the value for α in the cost functional is chosen to be $\alpha = 2.10^{-2}$. For the POD method we add 100 snapshots to the snapshot set in every iteration of Algorithm 10.4.1. The relative information content of the basis formed by the modes is required to be larger than 99.99%, i.e. $\gamma = 99.99$. We note that within this procedure a storage problem pops up with increasing iteration number of Algorithm 10.4.1. However, in practice it is sufficient to keep only the modes of the previous iteration while adding to this set the snapshots of the current iteration. An application of Algorithm 10.4.1 with step 4' instead of step 4 is presented in Example 10.5.3 below.

The suboptimal control u is sought in the space of deviations from the mean, i.e we make the ansatz

$$u = \sum_{i=1}^{M} \beta_i \Phi_i, \tag{10.83}$$

and the control target is tracking of the Stokes flow whose streamlines are depicted in Figure 10.5 (bottom). The same figure also shows the vorticity and the streamlines of the uncontrolled flow (top). For the numerical solution of the reduced optimization problems the Schur-complement SQP-algorithm is used, in the optimization literature frequently referred to as dual or range-space approach [NW99].

We first present a comparison between the optimal open-loop control strategy computed by Newton's method, and Algorithm 10.4.1. For details of the the implementation of Newton's method and further numerical results we refer

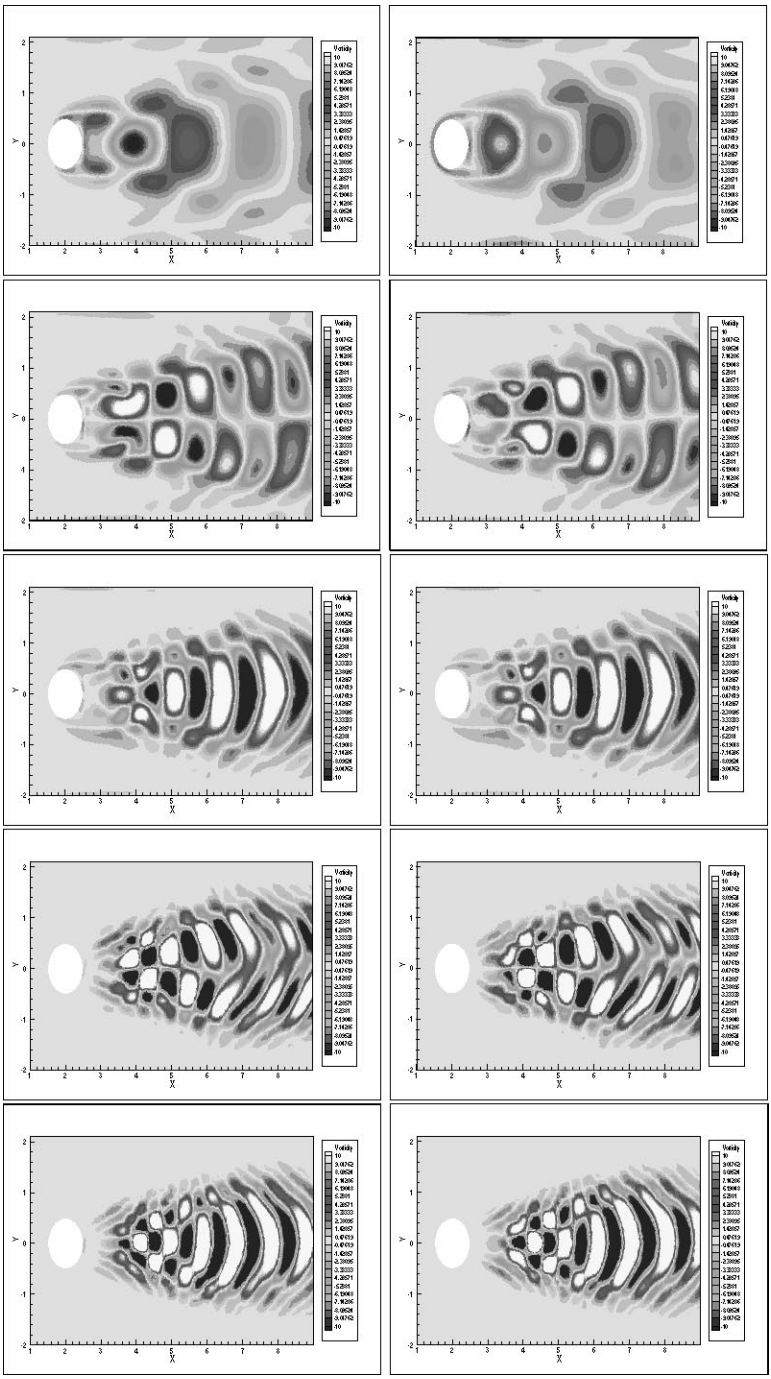


Fig. 10.4. First 10 modes generated from uncontrolled snapshots, vorticity

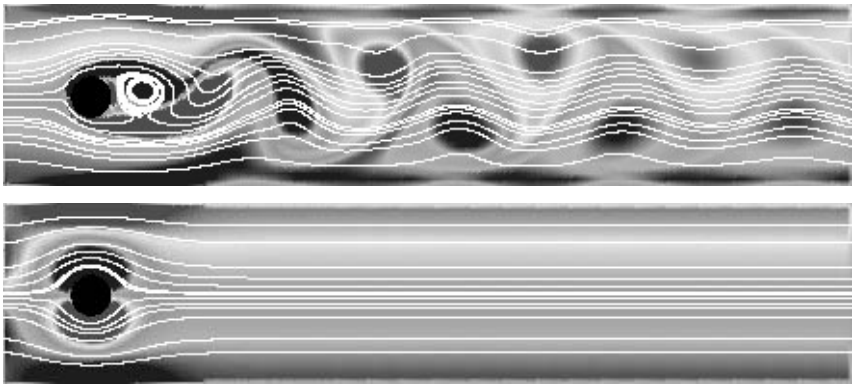


Fig. 10.5. Uncontrolled flow (top) and Stokes flow (bottom)

the reader to [Hin99, HK00, HK01]. In Figure 10.6 selected iterates of the evolution of the cost in $[0, T]$ for both approaches are given. The adaptive algorithm 10.4.1 terminates after 5 iterations to obtain the suboptimal control $\tilde{u}^*$. The termination criterium of step 5 in Algorithm 10.4.1 here is replaced by

$$\frac{|\hat{J}(u^{i+1}) - \hat{J}(u^i)|}{\hat{J}(u^i)} \leq 10^{-2}, \tag{10.84}$$

where

$$\hat{J}(u) = J(y(u), u)$$

denotes the so-called reduced cost functional and $y(u)$ stands for the solution to the Navier-Stokes equations for given control u. The algorithm achieves a remarkable cost reduction decreasing the value of the cost functional for the uncontrolled flow $\hat{J}(u^0) = 22.658437$ to $\hat{J}(\tilde{u}^*) = 6.440180$. It is also worth recording that to recover 99.99% of the energy stored in the snapshots in the first iteration 10 modes have to be taken, 20 in the second iteration, 26 in the third, 30 in the fourth, and 36 in the final iteration.

The computation of the optimal control with the Newton method takes approximately 17 times more cpu than the suboptimal approach. This includes an initialization process with a step-size controlled gradient algorithm. To obtain a relative error $|\nabla \hat{J}(u^n)|/|\nabla \hat{J}(u^0)|$ lower than 10^{-2}, 32 gradient iterations are needed with $\hat{J}(u^{32}) = 1.138325$. As initial control $u^0 = 0$ is taken. Note that every gradient step amounts to solving the non-linear Navier-Stokes equations in (10.77), the the corresponding adjoint equations, and a further Navier-Stokes system for the computation of the step-size in the gradient algorithm, compare [HK01]. Newton's algorithm then is initialized with u^{32} and 3 Newton steps further reduce the value of the cost functional to $\hat{J}(u^*) = 1.090321$. The controlled flow based on the Newton method is graphically almost indistinguishable from the Stokes flow in Figure 10.5. Figure 10.7

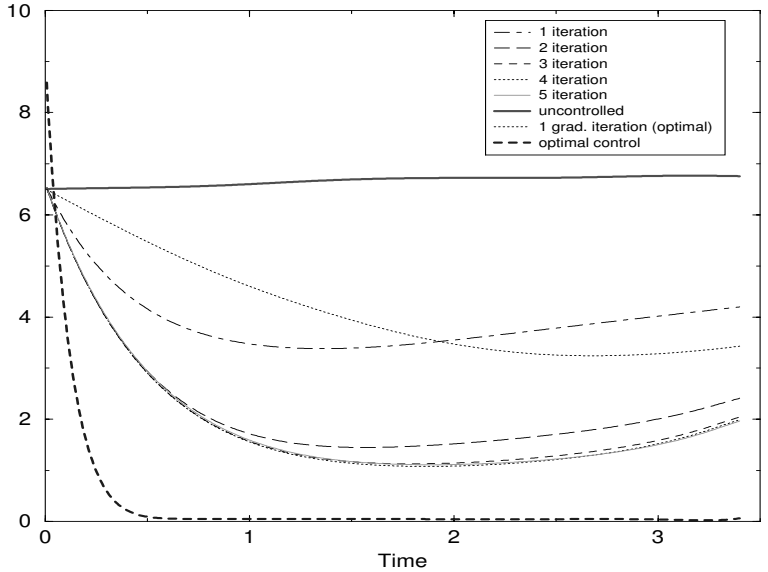


Fig. 10.6. Evolution of cost

shows the streamlines and the vorticity of the flow controlled by the adaptive approach at $t = 3.4$ (top) and the mean flow $\bar{y}$ (bottom), the latter formed with the snapshots of all 5 iterations. The controlled flow no longer contains vortex sheddings and is approximately stationary. Recall that the controls are sought in the space of deviations from the mean flow. This explains the remaining recirculations behind the cylinder. We expect that they can be reduced if the Ansatz for the controls in (10.83) is based on a POD of the snapshots themselves rather than on a POD of the deviation from their mean.

10.5.3 Example 2

The numerical results of the second application are taken from [AH00], compare also [Afa02]. The computational domain is given by $[-5, 15] \times [-5, 5]$ and is depicted in Figure 10.8. At the inflow a block-profile is prescribed, at the outflow do-nothing boundary conditions are used, and at the top and bottom boundary the velocity of the block profile is prescribed, i.e. the flow is open. The Reynolds number is chosen to be Re=100, so that the period of the flow covers the time horizon $[0, T]$ with $T = 5.8$. The numerical simulations are performed on an equidistant grid over this time interval containing 500 gridpoints. The control target z is given by the mean of the uncontrolled flow simulation, the regularization parameter in the cost functional is taken as

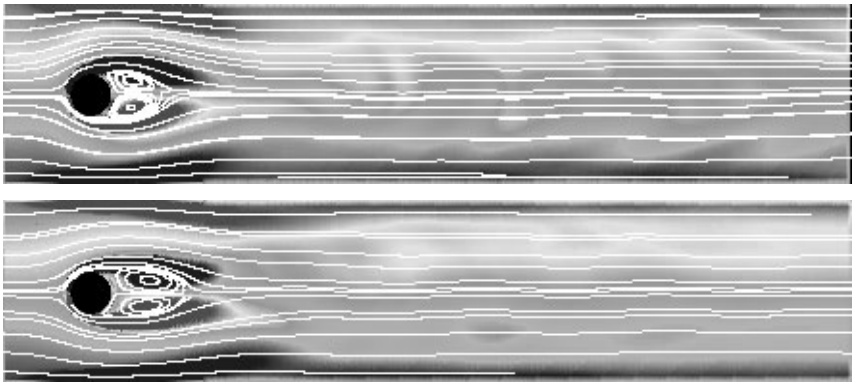


Fig. 10.7. Example 1: POD controlled flow (top) and mean flow $\bar{y}$ (bottom)

$\alpha = \frac{1}{10}$. The termination criterion in Algorithm 10.4.1 is chosen as in (10.84), the initial control is taken as $u^0 \equiv 0$. The iteration history for the value of the cost functional is shown in Figure 10.9, Figure 10.10 contains the iteration history for the control cost.

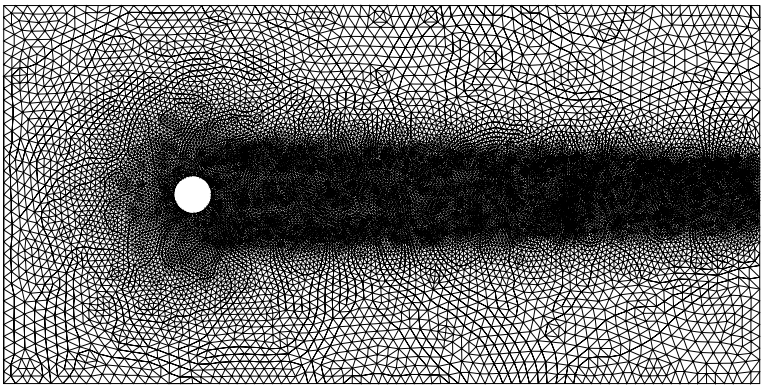


Fig. 10.8. Computational domain for the second application, 15838 velocity nodes.

The convergence criterium in Algorithm 10.4.1 is met after 7 iterations, where step 4 is replaced with step 4'. The value of the cost functional is $\hat{J}(\tilde{u}^*) = 0.941604$. Newton's method (without initialization by a gradient

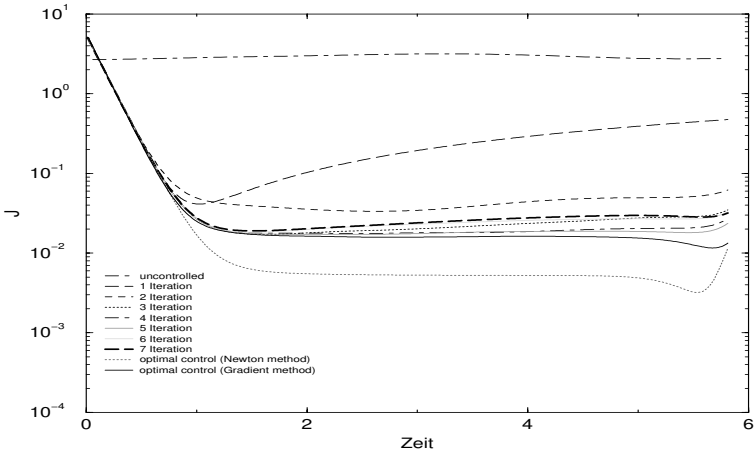


Fig. 10.9. Iteration history of functional values for Algorithm 10.4.1, second application

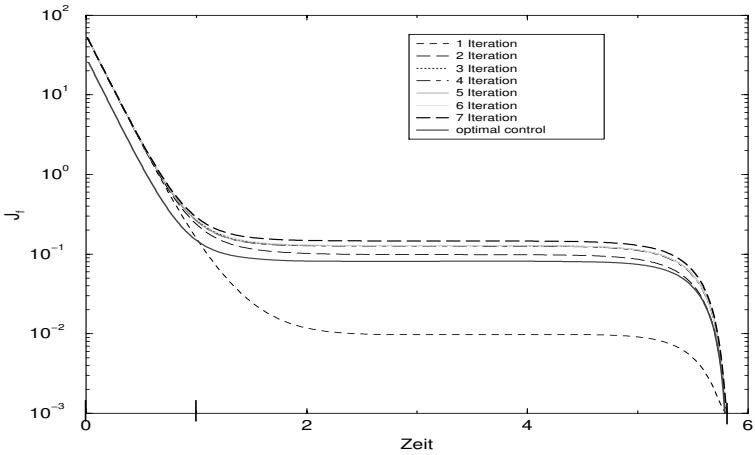


Fig. 10.10. Iteration history of control costs for Algorithm 10.4.1, second application

method) met the convergence criterium after 11 iterations with $\hat{J}(u_N^*) = 0.642832$, the gradient method needs 29 iterations with $\hat{J}(u_G^*) = 0.798193$. The total numerical amount for the computation of the suboptimal control $\tilde{u}^*$ for this numerical example is approximately 25 times smaller than that for the computation of u_N^*. The resulting open-loop control strategies are visually nearly indistinguishable. For a further discussion of the approach presented in this section we refer the reader to [Afa02, AH01].

We close this section with noting that the basic numerical ingredient in Algorithm 10.4.1 is the flow solver. The optimization with the surrogate model can be performed with MATLAB. Therefore, it is not necessary to develop numerical integration techniques for adjoint systems, which are one of the major ingredients of Newton- and gradient-type algorithms when applied to the full optimization problem (10.77).

10.6 Future Work and Conclusions

10.6.1 Future Research Directions

To the authors knowledge it is an open problem in many applications

1) to estimate how many snapshots to take, and
2) where to take them.

In this context goal-oriented concepts should be a future research direction. For an overview of goal oriented concepts in a-posteriori error analysis for finite elements we refer the reader to [BR01].

To report on first attempts for 1) and 2) we now sketch the idea of the goal-oriented concept. Denoting by $J(y)$ the quantity of interest, frequently called the goal (for example the drag or lift of the wake flow) and by $J(y_h)$ the response of the discrete model, the difference $J(y) - J(y_h)$ can be expressed approximately in terms of the residual of the state equation ρ and an appropriate adjoint variable z, i.e.

$$J(y) - J(y_h) = \langle \rho(y), z \rangle, \tag{10.85}$$

where $\langle \cdot, \cdot \rangle$ denotes an appropriate pairing.

With regard to 1) above, it is proposed in [HH04] to substitute y, z in (10.85) by their discrete counterparts y_h, z_h obtained from the POD model, and, starting on a coarse snapshot grid, to refine the snapshot grid and forming new POD models as long as the difference $J(y) - J(y_h)$ is larger than a given tolerance.

With regard to 2) a goal-oriented concept for the choice of modes out of a given set is presented in [MM03]. In [HH05] a goal-oriented adaptive time-stepping method for time-dependent pdes is proposed which uses POD models to compute the adjoint variables. In view of optimization of complex time dependent systems based on POD models adaptive goal oriented time stepping here serves a dual purpose; it provides a time-discrete model of minimum complexity in the full spatial setting w.r.t. the goal, and the time grid suggested by the approach may be considered as ideal snapshot grid upon which the model reduction should be based.

Let us also refer to [AG03], where the authors presented a technique to choose a fixed number of snapshots from a fine snapshot grid.

A further research area is the development of robust and efficient suboptimal feedback strategies for nonlinear partial differential equations. Here, we refer to the [KV99, KVX04, LV03, LV04]. However, the development of feedback laws based on partial measurement information still remains a challenging research area.

10.6.2 Conclusions

In the first part of this paper we present a mathematical introduction to finite- and infinite dimensional POD. It is shown that POD is closely related to the singular value decomposition for rectangular matrices. Of particular interest is the case when the columns of such matrices are snapshots of dynamical systems, such as parabolic equations, or the Navier-Stokes system. In this case POD allows to compute coherent structures, frequently called modes, which cary the relevant information of the underlying dynamical process. It then is a short step to use these modes in a Galerkin method to construct low order surrogate models for the full dynamics. The major contribution in the first part consists in presenting error estimates for solutions of these surrogate models.

In the second part we work out how POD surrogate models might be used to compute suboptimal controls for optimal control problems involving complex, nonlinear dynamics. Since controls change the dynamics, POD surrogate models need to be adaptively modified during the optimization process. With Algorithm 10.4.1 we present a method to cope with this difficulty. This algorithm in combination with the snapshot form of POD then is successfully applied to compute suboptimal controls for the cylinder flow at Reynolds number 100. It is worth noting that the numerical ingredients for this suboptimal control concept are a forward solver for the Navier-Stokes system, and an optimization environment for low-dimensional dynamical systems, such as MATLAB. As a consequence coding of adjoints, say is not necessary. As a further consequence the number of functional evaluations to compute suboptimal controls in essence is given by the number of iterations needed by Algorithm 10.4.1. The suboptimal concept therefore is certainly a candidate to obey the rule

$$\frac{\text{effort of optimization}}{\text{effort of simulation}} \leq \text{constant},$$

with a constant of moderate size. We emphasize that obeying this rule should be regarded as one of the major goals for every algorithm developed for optimal control problems with PDE-constraints.

Finally, we present first steps towards error estimation of suboptimal controls obtained with POD surrogate models. For linear-quadratic control problems the size of the error in the controls can be estimated in terms of the error of the states, and of the adjoint states. We note that for satisfactory estimates also POD for the adjoint system needs to be performed.

Acknowledgments

The authors would like to thank the both anonymous referees for the careful reading and many helpful comments on the paper.

The first author acknowledges support of the Sonderforschungsbereich 609 *Elektromagnetische Strömungskontrolle in Metallurgie, Kristallzüchtung und Elektrochemie*, located at the Technische Universität Dresden and granted by the German Research Foundation.

The second author has been supported in part by *Fonds zur Förderung des wissenschaftlichen Forschung* under Special Research Center *Optimization and Control*, SFB 03.

References

[AG03] Adrover, A., Giona, M.: Modal reduction of PDE models by means of snapshot archetypes. Physica D, **182**, 23–45 (2003).

[Afa02] Afanasiev, K.: Stabilitätsanalyse, niedrigdimensionale Modellierung und optimale Kontrolle der Kreiszylinderumströmung. PhD thesis, Technische Universität Dresden, Fakultät für Maschinenwesen (2002).

[AH00] Afanasiev K., Hinze, M.: Entwicklung von Feedback-Controllern zur Beeinflussung abgelöster Strömungen. Abschlußbericht TP A4, SFB 557, TU Berlin (2000).

[AH01] Afanasiev, K., Hinze, M.: Adaptive control of a wake flow using proper orthogonal decomposition. Lect. Notes Pure Appl. Math., **216**, 317–332 (2001).

[AFS00] Arian, E., Fahl, M., Sachs, E.W.: Trust-region proper orthogonal decomposition for flow control. Technical Report 2000-25, ICASE (2000).

[ABK01] Atwell, J.A., Borggaard, J.T., King, B.B.: Reduced order controllers for Burgers' equation with a nonlinear observer. Int. J. Appl. Math. Comput. Sci., **11**, 1311–1330 (2001).

[AHLS88] Aubry, N., Holmes, P., Lumley, J.L., Stone, E.: The dynamics of coherent structures in the wall region of a turbulent boundary layer. J. Fluid Mech., **192**, 115–173 (1988).

[Bän91] Bänsch, E.: An adaptive finite-element-strategy for the three-dimensional time-dependent Navier-Stokes-Equations. J. Comp. Math., **36**, 3–28 (1991).

[BJWW00] Banks, H.T., Joyner, M.L., Winchesky, B., Winfree, W.P.: Nondestructive evaluation using a reduced-order computational methodology. Inverse Problems, **16**, 1–17 (2000).

[BGGBW97] Barz, R.U., Gerbeth, G., Gelfgat, Y., Buhrig, E., Wunderwald, U.: Modelling of the melt flow due to rotating magnetic fields in crystal growth. Journal of Crystal Growth, **180**, 410–421 (1997).

[BR01] Becker, R., Rannacher, R.: An optimal control approach to a posteriori error estimation in finite elements. Acta Numerica, **10**, 1–102 (2001).

[DL92] Dautray, R., Lions, J.-L.: Mathematical Analysis and Numerical Methods for Science and Technology. Volume 5: Evolution Problems I. Springer-Verlag, Berlin (1992).

[DH02] Deckelnick, K., Hinze, M.: Error estimates in space and time for tracking-type control of the instationary Stokes system. ISNM, **143**, 87–103 (2002).

[DH04] Deckelnick, K., Hinze, M.: Semidiscretization and error estimates for distributed control of the instationary Navier-Stokes equations. Numerische Mathematik, **97**, 297–320 (2004).

[D85] Deimling, K.: Nonlinear Functional Analysis. Berlin, Springer (1985).

[DV01] Diwoky, F., Volkwein, S.: Nonlinear boundary control for the heat equation utilizing proper orthogonal decomposition. In: Hoffmann, K.-H., Hoppe, R.H.W., Schulz, V., editors, *Fast solution of discretized optimization problems*, International Series of Numerical Mathematics 138 (2001), 73–87.

[Fuk90] Fukunaga, K.: Introduction to Statistical Recognition. Academic Press, New York (1990).

[Gom02] Gombao, S.: Approximation of optimal controls for semilinear parabolic PDE by solving Hamilton-Jacobi-Bellman equations. In: Proc. of the 15th International Symposium on the Mathematical Theory of Networks and Systems, University of Notre Dame, South Bend, Indiana, USA, August 12–16 (2002).

[GL89] Golub, G.H., Van Loan, C.F.: Matrix Computations. The Johns Hopkins University Press, Baltimore and London (1989).

[HY02] Henri, T., Yvon, M.: Convergence estimates of POD Galerkin methods for parabolic problems. Technical Report No. 02-48, Institute of Mathematical Research of Rennes (2002).

[HH05] Heuveline, V., Hinze, M.: Adjoint-based adaptive time-stepping for partial differential equations using proper orthogonal decomposition, in preparation.

[HRT96] Heywood, J.G., Rannacher, R., Turek, S.: Artificial Boundaries and Flux and Pressure Conditions for the Incompressible Navier-Stokes Equations. Int. J. Numer. Methods Fluids, **22**, 325–352 (1996).

[Hin99] Hinze, M.: Optimal and Instantaneous control of the instationary Navier-Stokes equations. Habilitationsschrift, Technischen Universität Berlin (1999).

[HH04] Hinze, M.: Model reduction in control of time-dependent pdes. Talk given at the Miniworkshop on Optimal control of nonlinear time dependent problems, January 2004, Organizers K. Kunisch, A. Kunoth, R. Rannacher. Talk based on joint work with V. Heuveline, Karlsruhe.

[Hin05] Hinze, M.: A variational discretization concept in control constrained optimization: the linear-quadratic case. Computational Optimization and Applications **30**, 45–61 (2005).

[HK00] Hinze, M., Kunisch, K.: Three control methods for time - dependent Fluid Flow. Flow, Turbulence and Combustion **65**, 273–298 (2000).

[HK01] Hinze, M., Kunisch, K.: Second order methods for optimal control of time-dependent fluid flow. SIAM J. Control Optim., **40**, 925–946 (2001).

[HV05] Hinze, M., Volkwein, S.: POD Approximations for optimal control problems govered by linear and semi-linear evolution systems, in preparation.

[HV03] Hömberg, D., Volkwein, S.: Control of laser surface hardening by a reduced-order approach utilizing proper orthogonal decomposition. Mathematical and Computer Modelling, **38**, 1003–1028 (2003).

[HLB96] Holmes, P., Lumley, J.L., Berkooz, G.: Turbulence, Coherent Structures, Dynamical Systems and Symmetry. Cambridge Monographs on Mechanics, Cambridge University Press (1996).

[IR98] Ito, K., Ravindran, S.S.: A reduced basis method for control problems governed by PDEs. In: Desch, W., Kappel, F., Kunisch, K. (ed), Control and Estimation of Distributed Parameter Systems. Proceedings of the International Conference in Vorau, 1996, 153–168 (1998).

[Kat80] Kato, T.: Perturbation Theory for Linear Operators. Springer-Verlag, Berlin (1980).

[KV99] Kunisch, K., Volkwein, S.: Control of Burgers' equation by a reduced order approach using proper orthogonal decomposition. J. Optimization Theory and Applications, **102**, 345–371 (1999).

[KV01] Kunisch, K., Volkwein, S.: Galerkin proper orthogonal decomposition methods for parabolic problems. Numerische Mathematik, **90**, 117–148 (2001).

[KV02] Kunisch, K., Volkwein, S.: Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics. SIAM J. Numer. Anal., **40**, 492–515 (2002).

[KVX04] Kunisch, K., Volkwein, S., Lie, X.: HJB-POD based feedback design for the optimal control of evolution problems. To appear in SIAM J. on Applied Dynamical Systems (2004).

[LMG] Lall, S., Marsden, J.E., Glavaski, S.: Empirical model reduction of controlled nonlinear systems. In: Proceedings of the IFAC Congress, vol. F, 473–478 (1999).

[LV03] Leibfritz, F., Volkwein, S.: Reduced order output feedback control design for PDE systems using proper orthogonal decomposition and nonlinear semidefinite programming. Linear Algebra Appl., to appear.

[LV04] Leibfritz, F., Volkwein, S.: Numerical feedback controller design for PDE systems using model reduction: techniques and case studies. Submitted (2004).

[MM03] Meyer, M., Matthies, H.G.: Efficient model reduction in nonlinear dynamics using the Karhunen-Loève expansion and dual weighted residual methods. Comput. Mech., **31**, 179–191 (2003).

[NAMTT03] Noack, B., Afanasiev. K., Morzynski, M., Tadmor, G., Thiele, F.: A hierachy of low-dimensional models for the transient and post-transient cylinder wake. J. Fluid. Mech., **497**, 335-363 (2003).

[Nob69] Noble, B.: Applied Linear Algebra. Englewood Cliffs, NJ : Prentice-Hall (1969).

[NW99] Nocedal, J, Wright, S.J.: Numerical Optimization. Springer, NJ (1999).

[LT01] Ly, H.V., Tran, H.T.: Modelling and control of physical processes using proper orthogonal decomposition. Mathematical and Computer Modeling, **33**, 223–236, (2001).

[Ran00] Rannacher, R.: Finite element methods for the incompressible Navier-Stokes equations. In: Galdi, G.P. (ed) et al., Fundamental directions in mathematical fluid mechanics. Basel: Birkhuser. 191-293 (2000).

[RP02] Rathinam, M., Petzold, L.: Dynamic iteration using reduced order models: a method for simulation of large scale modular systems. SIAM J. Numer. Anal., **40**, 1446–1474 (2002).

[Rav00] Ravindran, S.S.: Reduced-order adaptive controllers for fluid flows using POD. J. Sci. Comput., **15**:457–478 (2000).

[RS80] Reed, M., Simon, B.: Methods of Modern Mathematical Physics I: Functional Analysis. Academic Press, New York (1980).

[RF94] Rempfer, D., Fasel, H.F.: Dynamics of three-dymensional coherent structures in a flat-plate boundary layer. J. Fluid Mech. **275**, 257–283 (1994).

[Row04] Rowley, C.W.: Model reduction for fluids, using balanced proper orthogonal decomposition. To appear in Int. J. on Bifurcation and Chaos (2004).

[ST96] Schäfer, M., Turek, S.: Benchmark computations of laminar flow around a cylinder. In: Hirschel, E.H. (ed), Flow simulation with high-performance computers II. DFG priority research programme results 1993 - 1995. Wiesbaden: Vieweg. Notes Numer. Fluid Mech., **52**, 547–566 (1996).

[SK98] Shvartsman, S.Y., Kevrikidis, Y.: Nonlinear model reduction for control of distributed parameter systems: a computer-assisted study. AIChE Journal, **44**, 1579–1595 (1998).

[Sir87] Sirovich, L.: Turbulence and the dynamics of coherent structures, parts I-III. Quart. Appl. Math., **XLV**, 561–590 (1987).

[TGP99] Tang, K.Y., Graham, W.R., Peraire, J.: Optimal control of vortex shedding using low-order models. I: Open loop model development. II: Model based control. Int. J. Numer. Methods Eng., **44**, 945–990 (1999).

[Tem88] Temam, R.: Infinite-Dimensional Dynamical Systems in Mechanics and Physics, volume 68 of Applied Mathematical Sciences, Springer-Verlag, New York (1988).

[Vol01a] Volkwein, S.: Optimal control of a phase-field model using the proper orthogonal decomposition. Zeitschrift für Angewandte Mathematik und Mechanik, **81**, 83–97 (2001).

[Vol01b] Volkwein, S.: Second-order conditions for boundary control problems of the Burgers equation. Control and Cybernetics, **30**, 249–278 (2001).

[WP01] Willcox, K., Peraire, J.: Balanced model reduction via the proper orthogonal decomposition. AIAA Journal, **40**:11, 2323–2330, (2002).

Part II

Benchmarks

This part contains a collection of models that can be used for evaluating the properties and performance of new model reduction techniques and new implementations of existing techniques. The first paper (Chapter 11) describes the main features of the Oberwolfach Benchmark Collection, which is maintained at

`http://www.imtek.de/simulation/benchmark`.

It should be noted that this is an open project, so new additions are always welcome. The submission procedure is also described in this first paper. The data for linear-time invariant systems in all benchmarks are provided in the common Matrix Market format, see

`http://math.nist.gov/MatrixMarket/`.

In order to have a common format to deal with nonlinear models, in Chapter 12, a data exchange format for nonlinear systems is proposed. Most of the remaining papers describe examples in the Oberwolfach Benchmark Collection, where the first six entries (Chapters 13–18) come from microsystem technology applications, then Chapter 19 presents an optimal control problem for partial differential equations, and an example from computational fluid dynamics is contained in Chapter 20. Chapter 21 describes second-order models in vibration and acoustics while Chapters 22 and 23 present models arising in circuit simulation.

Also included (see Chapter 24) is a revised version of SLICOT's model reduction benchmark collection, see

`http://www.win.tue.nl/niconet/NIC2/benchmodred.html`.

For integration in the Oberwolfach Benchmark Collection only those examples from the SLICOT collection are chosen that exhibit interesting model features and that are not covered otherwise. It should also be noted that the SLICOT benchmark collection merely focuses on control applications and not all examples are large-scale as understood in the context of the Oberwolfach mini-workshop. Therefore, only those examples considered appropriate are included in Chapter 24.

11

Oberwolfach Benchmark Collection

Jan G. Korvink and Evgenii B. Rudnyi

Institute for Microsystem Technology, Albert Ludwig University
Georges Köhler Allee 103, 79110 Freiburg, Germany
{korvink,rudnyi}@imtek.uni-freiburg.de

Summary. A Web-site to store benchmarks for model reduction is described. The site structure, submission rules and the file format are presented.

11.1 Introduction

Model order reduction is a multi-disciplinary area of research. The driving force from industry are engineering design requirements. The development of theory to solve these problems remains clearly in the hands of mathematicians. Numerical analysts and programmers are solving issues of an efficient, reliable and scalable implementation.

A benchmark is a natural way to allow different groups to communicate results with each other. Engineers convert a physical problem into a system of ordinary differential equations (ODEs) and specify requirements. Provided the system is written in a computer-readable format, this supplies an easy-to-use problem in order to try different algorithms for model reduction and compare different software packages.

During the Oberwolfach mini-Workshop on Dimensional Reduction of Large-Scale Systems, IMTEK agreed to host as well as develop rules for a related benchmark Web site. The site is running since spring of 2004 at `http://www.imtek.uni-freiburg.de/simulation/benchmark/` and the rules are described below.

The file format to represent a nonlinear system of ODEs has been developed during the joint DFG project between IMTEK, Freiburg University and Institute of Automation, University of Bremen: The Dynamic System Interchange Format (DSIF, `http://www.imtek.uni-freiburg.de/simulation/mstkmpkt/`). It is presented in Chapter 12 where also the background for model reduction benchmarks is described in more detail.

Unfortunately, there are two problems with the DSIF format. First, it does not scale well to high-dimensional systems. For example, when a benchmark for a system of linear ODEs of dimension of about 70 000 with sparse system

matrices containing about 4 000 000 nonzeros has been written in the DSIF format, Matlab 6 crashed while reading the file. Second, it is not easy to parse it outside of Matlab. As result, we present an alternative format to store linear ODEs based on the Matrix Market format [BPR96]. For nonlinear ODE systems, the DSIF format seems to be the only alternative and we highly recommend its use in this case.

11.2 Documents

The collection consists of documents, benchmarks and reports. A benchmark and a related report may be written by different authors.

Each document is written according to conventional scientific practice, that is, it describes matters in such a way that, at least in principle, anyone could reproduce the results presented. The authors should understand that the document may be read by people from quite different disciplines. Hence, abbreviations should be avoided or at least explained and references to the background ideas should be made.

11.2.1 Benchmark

The goal of a benchmark document is to describe the origin of the dynamic system and its relevance to the application area. It is important to present the mathematical model, the meaning of the inputs and outputs and the desired behavior from the application viewpoint.

A few points to be addressed:

- The purpose of the model should be explained clearly. (For instance, simulation, iterative system design, feedback control design, ...)
- Why should the model be reduced at all? (For instance, reducing simulation time, reducing implementation effort in observers, controllers...)
- What are the QUALITATIVE requirements of the reduced model? What variables are to be approximated well? Is the step response to be approximated or is it the Bode plot? What are typical input signals? (Some systems are driven by a step function and nothing else, others are driven by a wide variety of input signals, others are used in closed loop and can cause instability, although being stable themselves).
- What are the QUANTITATIVE requirements of the reduced model? Best would be if the authors of any individual model can suggest some cost functions (performance indices) to be used for comparison. These can be in the time domain, or in the frequency domain (including special frequency band), or both.
- Are there limits of input and state variables known? (Application related or generally)? What are the physical limits where the model becomes useless/false? If known a-priori: Out of the technically typical input signals, which one will cause "the most nonlinear" behavior?

If the dynamic system is obtained from partial differential equations, then the information about material properties, geometrical data, initial and boundary conditions should be given. The exception to this rule is the case when the original model came from industry. In this case, if trade secrets are tied with the information mentioned, it may be kept hidden.

The authors are encouraged to produce several dynamic models of different dimensions in order to provide an opportunity to apply different software and to research scalability issues. If an author has an interactive page on his/her server to generate benchmarks, a link to this page is welcomed.

The dynamic system may be obtained by means of compound matrices, for example, when the second-order system is converted to first-order. In this case, the document should describe such a transformation but in the datafile the original and not the compound matrices should be given. In this way, this will allow users to research other ways of model reduction of the original system.

11.2.2 Report

A report document may contain:

a) The solution of the original benchmark that contains sample outputs for the usual input signals. Plots and numerical values of time and frequency response. Eigenvalues and eigenvectors, singular values, poles, zeros, etc.

b) Model reduction and its results as compared to the original system.

c) Description of any other related results.

We stress the importance to describe the software employed as well as its related options.

11.2.3 Document Format

Any document is considered as a Web-page. As such it should have a main page in the HTML format and all other objects linked to the main page such as pictures and plots (GIF, JPEG), additional documents (PDF, HTML). In particular, a document can have just a small introductory part written in HTML and the main part as a linked PDF document.

The authors are advised to keep the layout simple.

Scripts included in the Web-page should be avoided, or at least they should not be obligatory to view the page.

Numerical data including the original dynamic system and the simulation results should be given in a special format described below.

11.3 Publishing Method

A document is submitted to IMTEK in the electronic form as an archive of all the appropriate files (tar.gz or zip) at `http://www.imtek.uni-freiburg.de/`

`simulation/benchmark/`. Then it is placed in a special area and enters a reviewing stage. Information about the new document is posted to a benchmark mailing list `benchmark@elmo.imtek.uni-freiburg.de` and send to reviewers chosen by a chief editor. Depending on the comments, the document is published, rejected or sent to authors to make corrections. The decision is taken by an editorial board.

11.3.1 Rules for Online Submission

- Only ZIP or TAR.GZ archives are accepted for the submission.
- The archive should contain at least one HTML file, named `index.html`. This file represents the main document file.
- The archive must only contain files of the following types: *.html, *.htm, *.pdf, *.gif, *.jpg, *.png, *.zip, *.tar.gz
- After the submission, the files are post-processed:
  - File types not specified above are deleted.
  - Only the body part of every HTML file is kept.
  - All the format/style/css information, like `style=..`, `class=..` are removed from the body part.
- If you decide to use PDF documents, use the `index.html` to include links to them.
- There are three states of the submission:
  - Submitted: The author and the chief editor receive a notification mail. The submission is only accessible for the chief editor to accept the submission.
  - Opened for review: The submission is open for users to post their comments and reviews. After that the chief editor can accept the paper.
  - Accepted: The submission is open for everybody.

11.4 Datafiles

Below we suggest a format for linear dynamic systems. The development of a scalable data format for time-dependent and nonlinear dynamic systems is considered to be a challenge to be solved later on. At present, for time-dependent and nonlinear systems, we suggest to use the Dynamic System Interchange Format described in Chapter 12.

All the numerical data for the collection can be considered as a list of matrices, a vector being an $m \times 1$ matrix. As a result, first one should follow a naming convention for the matrices, second one should write each matrix in the format described below.

11.4.1 Naming Convention

For the two cases of a linear dynamic system of the first and second orders, the naming convention is as follows

$$\begin{aligned} E\dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \tag{11.1}$$

$$\begin{aligned} M\ddot{x} + E\dot{x} + Kx &= Bu \\ y &= Cx + Du \end{aligned} \tag{11.2}$$

An author can use another notation in the case when the convention above is not appropriate. This should be clearly specified in the benchmark document.

11.4.2 Matrix Format for Linear Systems

A matrix should be written in the the Matrix Market format [BPR96].

A file with a matrix should be named as *problem_name.matrix_name*.

If there is no file for a matrix, it is assumed to be identity for the M, E, K, A matrices and 0 for the D matrix.

All matrix files for a given problem should be compressed in a single zip or tar.gz archive.

11.5 Acknowledgments

We appreciate many useful comments on the initial draft of the document from Karl Meerbergen, Peter Benner, Volker Mehrmann, Paul Van Dooren and Boris Lohmann.

The interactive Web-site has been developed and maintained by C. Friese.

This work is partially funded by the DFG project MST-Compact (KO-1883/6), the Italian research council CNR together with the Italian province of Trento PAT, by the German Ministry of Research BMBF (SIMOD), and an operating grant of the University of Freiburg.

References

[BPR96] Boisvert, R.F., Pozo, R., Remington, K.A.: The Matrix Market exchange formats: Initial design. National Institute of Standards and Technology, NIST Interim Report 5935 (1996)

12

A File Format for the Exchange of Nonlinear Dynamical ODE Systems

Jan Lienemann[1], Behnam Salimbahrami[2], Boris Lohmann[2], and Jan G. Korvink[1]

[1] Institute for Microsystem Technology, Albert Ludwig University
Georges Köhler Allee 103, 79110 Freiburg, Germany
{lieneman,korvink}@imtek.de
[2] Lehrstuhl für Regelungstechnik, Technische Universität München, Boltzmannstr. 15, D-85748 Garching bei München, Germany
{Salimbahrami, Lohmann}@tum.de

Summary. We propose an ASCII file format for the exchange of large systems of nonlinear ordinary differential matrix equations, e.g., from a finite element discretization. The syntax of the format is similar to a MATLAB [Mat] `.m` file. It supports both dense and sparse matrices as well as certain macros for special matrices like zero and unity matrices. The main feature is that nonlinear functions are allowed, and that nonlinear coupling between the state variables or to an external input can be represented.

12.1 Introduction

In many fields of physics and engineering, computer simulation of complex devices has become an important tool for device designers. When building prototypes is expensive or takes a long time, or when design optimizations require the testing of a device with a large number of small changes, simulation becomes an absolute prerequisite for an efficient design process.

As this process continues, the models get more detailed, and a larger number of coupling effects is included. Unfortunately, this often leads to a large increase in computational effort, in particular if transient behavior is to be optimized.

Another challenge is that a device is usually a part of a larger system, which the designer wishes to simulate as a whole. This requires to couple a considerable number of devices and simulate them simultaneously, leading to an immense growth of computational complexity.

Therefore, there is an urgent need to find methods to reduce the computational effort for transient and harmonic simulation. Fortunately at present, there is large interest of the scientific community in *Model order reduction*

(MOR). The promise of MOR is to replace the large system of equations occurring from detailed models with a much smaller system, whose results are still "good enough" (by a certain measure) to be able to draw conclusions from the simulation results. There is a number of results available; for linear systems, one could even say that the problem is almost solved. However for nonlinear systems, a lot of research still needs to be done.

In order to accelerate and facilitate this research, it is important to have a standard set of benchmarks to be able to compare different algorithms based on real life applications. Such a collection must be in a common format, else the scientist needs to waste too much time on file format conversion issues.

We therefore discuss a file format which we called *Dynamic System Interchange Format* (DSIF), and which we want to encourage others to use for their benchmarks of nonlinear dynamic systems.

12.2 PDEs and their Discretization

We first discuss the physical origins of the ODEs, their linear approximation and where nonlinear systems come from.

12.2.1 Linear PDEs and Discretization

In many fields, the underlying equations are linear, or at least can be linearized within sufficient accuracy. Examples are structures with small displacements, heat transfer for small temperature changes (i.e., the material properties do not change with temperature), and the flow of electric current. They are described by partial differential equations. One example is the heat transfer equation

$$\nabla \cdot (\kappa(\mathbf{r})\nabla T(\mathbf{r},t)) + Q(\mathbf{r},t) - \rho(\mathbf{r})C_p(\mathbf{r})\frac{\partial T(\mathbf{r},t)}{\partial t} = 0 \tag{12.1}$$

with $\mathbf{r}$ the position, t the time, κ the thermal conductivity of the material, C_p the specific heat capacity, ρ the mass density, Q the heat generation rate, and T is the unknown temperature distribution to be determined.

For numerical solution, this equation has to be discretized, e.g., with the finite element method [HU94]. As long as κ, Q, ρ and C_p are constant in time and temperature, the resulting system of equations can be written in matrix-vector notation in the form

$$\mathsf{E}\dot{\mathbf{T}}(t) = \mathsf{A}\mathbf{T}(t) + \mathbf{Q}(t). \tag{12.2}$$

12.2.2 Nonlinear Equations

For many applications, the linear approximation does not hold any more. In the example above the dependence of material properties on temperature may

not be neglected for large temperature changes. In other cases, the equation itself is already nonlinear. One important example is the Navier–Stokes equation for fluid dynamics:

$$\rho\left(\frac{\partial \mathbf{u}(\mathbf{r},t)}{\partial t} + \mathbf{u}(\mathbf{r},t)\cdot\nabla\mathbf{u}(\mathbf{r},t)\right) = -\nabla p(\mathbf{r},t) + \nabla\cdot\tau(\mathbf{r},t) + \rho\mathbf{f}(\mathbf{r},t), \quad (12.3)$$

where $\mathbf{u}$ is the velocity of the fluid, ρ is the density of the fluid, and p is pressure. τ is the viscous stress tensor, which can be calculated from the derivatives of $\mathbf{u}$. $\mathbf{f}$ is an external force like gravity. The term $\mathbf{u}\cdot\nabla\mathbf{u}$ introduces a nonlinearity which is the cause of many surprising effects, but also of difficulties solving the equation.

It is still possible to perform a discretization of this problem. However, the resulting equations contain a nonlinear part. Using $\mathbf{x}$ for the system state, e.g., the searched coefficient values for the FEM basis functions, those systems can be expressed by the following matrix equation:

$$\mathsf{E}\dot{\mathbf{x}}(t) = \mathsf{A}\mathbf{x}(t) + \mathsf{B}\mathbf{u}(t) + \mathbf{b} + \mathsf{F}\mathbf{f}(t,\mathbf{x}(t),\mathbf{u}(t)). \quad (12.4)$$

Here, $\mathbf{u}$ stands for a number of inputs or loads to the system which are distributed by the matrix B, $\mathbf{b}$ provides constant loads (e.g., for Dirichlet boundary conditions), $\mathbf{f}$ is a vector of all nonlinear parts of the equations, and E and A are constant matrices with material and geometry parameters. The matrix F serves only a practical purpose: it allows us to decrease the size of $\mathbf{f}$ and use linear combinations of only a few common nonlinear functions, with the weights given by the entries of F.

The main reason to separate linear and nonlinear parts in the equation is that the linear parts are much easier to handle. It is thus easy to retrieve an linearized version of the system. This means that $\mathbf{f}$ should not include linear parts.

12.2.3 Outputs

In many applications, engineers are not interested in the complete field solution; especially the interior is often of minor interest. MOR algorithms can take this into account and optimize their result to be accurate mostly at certain computational nodes. Hence, it is useful to allow a method to pick only a small number of states. A general description should also include a possibility to apply a nonlinear function to these states. We then end up with the system of equations

$$\begin{aligned}\mathsf{E}\dot{\mathbf{x}}(t) &= \mathsf{A}\mathbf{x}(t) + \mathsf{B}\mathbf{u}(t) + \mathbf{b} + \mathsf{F}\mathbf{f}(t,\mathbf{x}(t),\mathbf{u}(t))\\ \mathbf{y}(t) &= \mathsf{C}\mathbf{x}(t) + \mathsf{D}\mathbf{u}(t) + \mathbf{d} + \mathsf{G}\mathbf{g}(t,\mathbf{x}(t),\mathbf{u}(t)),\end{aligned} \quad (12.5)$$

where $\mathbf{y}$ is called the output of the system.

12.2.4 Higher Order Systems

Systems considered so far feature only first order time derivatives. Higher order time derivatives are in principle not a problem, since methods exist to transfer these systems to the first order by introducing velocity state variables, resulting in double the number of equations.

However, it may be useful to preserve the higher order term explicitly. Therefore, another notation is introduced:

$$\begin{aligned} \mathsf{M}\ddot{\mathbf{x}} + \mathsf{E}\dot{\mathbf{x}}(t) + \mathsf{K}\mathbf{x}(t) = & \quad \mathsf{B}\mathbf{u}(t) + \mathbf{b} + \mathsf{F}\mathbf{f}(t, \mathbf{x}(t), \dot{\mathbf{x}}(t), \mathbf{u}(t)) \\ \mathbf{y}(t) = \mathsf{C}\mathbf{x}(t) + \mathsf{D}\mathbf{u}(t) + \mathbf{d} + \mathsf{G}\mathbf{g}(t, \mathbf{x}(t), \mathbf{u}(t)). & \end{aligned} \tag{12.6}$$

Both forms (12.5) and (12.6) follow the conventional notations in many engineering fields.

12.2.5 Initial Conditions

Time dependant PDEs (i.e., hyperbolic and parabolic PDEs) need the system state at the beginning of the simulation. The simulation is assumed to start at time $t = 0$. For (12.5), giving the value of the current state vector $\mathbf{x}(0)$ is sufficient; we will denote this vector by `x0`. For (12.6), it is necessary also to give the time derivatives (velocities) $\dot{\mathbf{x}}(0)$, which will be denoted by `v0`.

12.3 The Dynamic System Interchange Format

In order to put these equations to a computer readable format, we use the MATLAB format as starting point. MATLAB is a computer algebra system used by many scientists and engineers for numerical computations. One advantage is that a file describing a linear system can be read into MATLAB and is thus ready for immediate processing. Since the file might also be read in by custom parsers, we do not use the full capabilities of MATLAB command files, but limit the acceptable input as follows.

12.3.1 General

The file is a plain ASCII text. All numbers are real numbers in floating point or scientific exponential notation (e.g., `5`, `0.1`, `8.8542e-12`). Comments start with a "`%`" character; they are allowed everywhere in the file, also in the middle of a line. They stop at the next line break:

```
% This is a comment
a = 1 % + 2
% a will be 1
```

The file format is sensitive to line breaks; to continue a line, use "␣..." at its end (note the leading whitespace):

```
a = 1 ...
 + 2
% a will be 3
```

If there is a "%" on the line before the continuation, the latter will be ignored. Statements are ended by linebreaks or by ";".

Matrices are enclosed by "[ ]". Elements in a row are separated by either "," or whitespace (space or tab). Matrix rows are separated by either ";" or line breaks:

```
a = [1, 2 ...
3; 4 5 6
7 8 9]
% a will be
% 1 2 3
% 4 5 6
% 7 8 9
```

Vectors are matrices where one dimension is 1.

Functions are written in lower case letters, with their argument between round parentheses:

```
a = sin(3.14159265)
a = sin(x(3)+u(1))
```

We recommend the use of the functions in Table 12.1; the list is essentially based on the ISO C99 standard [ISO]. If necessary, own functions may be introduced, but their implementation and properties must be documented elsewhere. Only functions from $\mathbb{R}^n \mapsto \mathbb{R}$ or subsets thereof are possible. All identifiers are case sensitive.

The functions may take the time `t`, elements of the state vector `x(`i`)`, the time derivatives (velocities) `v(`i`)` and the input vector `u(`i`)` as argument, with i the index of the element.

Table 12.1. Recommended mathematical functions for the DSIF file format

`a+b`	$a + b$ (addition)
`a-b`	$a - b$ (subtraction; missing a means negation)
`a*b`	$a \times b$ (multiplication)
`a/b`	$a \div b$ (division)
`a^b` or `a**b`	a^b (power)
`(cond)?a:b`	If `cond` is true, return a else b
`abs(a)`	$\lvert a \rvert$ (absolute value)
`acos(a)`	$\cos^{-1} a \in [0, \pi]$ (inverse cosine)
`acosh(a)`	$\cosh^{-1} a \in [0, \infty]$ (inverse hyperbolic cosine)
`asin(a)`	$\sin^{-1} a \in [-\pi/2, \pi/2]$ (inverse sine)
`asinh(a)`	$\sinh^{-1} a$ (inverse hyperbolic cosine)
`atan(a)`	$\tan^{-1} a \in [-\pi/2, \pi/2]$ (inverse tangent)
`atan2(y,x)`	$\tan^{-1}(y/x) \in [-\pi, \pi]$ (inverse tangent: returns the angle whose tangent is y/x. Full angular range)
`cbrt(a)`	$\sqrt[3]{a} \in [-\infty, \infty]$ (real cubic root)
`ceil(a)`	$\lceil a \rceil$ (smallest integer $\geq$ a)
`cos(a)`	$\cos a$ (cosine)
`cosh(a)`	$\cosh a$ (hyperbolic cosine)
`erf(a)`	$\operatorname{erf} a$ (error function)
`erfc(a)`	$\operatorname{erfc} a$ (complementary error function)
`exp(a)`	e^a (exponential)
`floor(a)`	$\lfloor a \rfloor$ (largest integer $\leq$ a)
`lgamma(a)`	$\ln \lvert \Gamma(a) \rvert$ (natural logarithm of the absolute value of the gamma function)
`log(a)`	$\ln a$ (natural logarithm)
`log10(a)`	$\log_{10} a$ (base-10 logarithm)
`log2(a)`	$\log_2 a$ (base-2 logarithm)
`max(a,b,...)`	the largest of a, b, etc.
`min(a,b,...)`	the smallest of a, b, etc.
`mod(a,b)`	$a - \lfloor a/b \rfloor b$ (the remainder of the integer division of a by b)
`pow(a,b)`	a^b (power)
`round(a,b)`	nearest integer, or value with larger magnitude if a is exactly in between two integers, i.e., $n + 0.5,\ n \in \mathbb{N}$
`sign(a)`	sign of a or 0 if $a = 0$
`sin(a)`	$\sin a$ (sine)
`sinh(a)`	$\sinh a$ (hyperbolic sine)
`sqrt(a)`	$\sqrt{a}$ (square root)
`tan(a)`	$\tan a$ (tangent)
`tanh(a)`	$\tanh a$ (hyperbolic tangent)
`tgamma(a)`	$\Gamma(a)$ (gamma function)
`trunc(a)`	nearest integer not larger in magnitude (towards zero)

12.3.2 File Header

The first line of the file is a version string to distinguish between the different versions having occurred during the development:

```
DSIF_version='0.1.0'
```

This is followed by a few lines describing the dimensions of the system:

```
n = 3
m = 2
p = 1
r = 2
s = 1
q = 3
o = 2
```

The parameters have the following meaning:

- `n` State space size
  (number of components of state vector `x`)
- `m` Number of control input signals
  (number of components of input vector `u`)
- `p` Number of output variables
  (number of components of output vector `y`)
- `r` Number of state nonlinearities
  (number of components of vector `f`)
- `s` Number of output nonlinearities
  (number of components of vector `g`)
- `q` Number of equations
  (most times `q=n`)
- `o` Maximum order of time derivatives; 1 for a system of form (12.5), 2 for a system of form (12.6), 0 if no time derivative occurs at all.

12.3.3 System Matrices and Vectors

Following the header, the actual system data is given. Depending on the order of the system the nomenclature and number of matrices to be given changes. Matrices and vectors not given take default values; matrices with zero size in one dimension should also be not specified. The matrices required for (12.5) and (12.6) and the default values are shown in Table 12.2. `E`, `A`, `B`, `b`, `F`, `C`, `D`, `d`, `G`, `M`, `K`, `x0` and `v0` should be constant, i.e. with explicitly given values. `f` and `g` can contain functions of time, states, velocities and input. They should not contain any linear part to simplify linearization.

A number of macros can be used to facilitate entering some special matrices. The macros are described in Table 12.3.

Table 12.2. Matrices required to describe a system of first order in time (*left*) and second order in time (*right*)

Matrix	Dimensions	Default
E	q×n	eye(q,n)
A	q×n	eye(q,n)
B	q×m	eye(q,m)
b	q×1	zeros(q,1)
F	q×r	eye(q,r)
C	p×n	eye(p,n)
D	p×m	zeros(p,m)
d	p×1	zeros(p,1)
G	p×s	eye(p,s)
f	r×1	zeros(r,1)
g	s×1	zeros(s,1)
x0	n×1	zeros(n,1)

Matrix	Dimensions	Default
M	q×n	eye(q,n)
E	q×n	eye(q,n)
K	q×n	eye(q,n)
B	q×m	eye(q,m)
b	q×1	zeros(q,1)
F	q×r	eye(q,r)
C	p×n	eye(p,n)
D	p×m	zeros(p,m)
d	p×1	zeros(p,1)
G	p×s	eye(p,s)
f	r×1	zeros(r,1)
g	s×1	zeros(s,1)
x0	n×1	zeros(n,1)
v0	n×1	zeros(n,1)

12.4 Example

Assume we have the following system of equations:

$$\begin{aligned}\begin{bmatrix} 1 & 0.2 & 0 \\ 0.2 & 1 & 0.2 \\ 0 & 0.2 & 1 \end{bmatrix} \ddot{\mathbf{x}} + \begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 0.1 \end{bmatrix} \dot{\mathbf{x}} + \begin{bmatrix} 1 & 0 & 0 \\ -1 & 2 & 0 \\ 0 & -1 & 2 \end{bmatrix} \mathbf{x} \\ = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \mathbf{u} + \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{pmatrix} \sin(u_1 + x_2) \\ \exp(u_2/x_1) \end{pmatrix}\end{aligned} \tag{12.7}$$

$$\mathbf{y} = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \mathbf{x} + (\lfloor \exp(x_3 t) u_1 \rfloor) \,. \tag{12.8}$$

A possible file describing this system could look like the following:

```
DSIF_version='0.1.0'
n = 3
m = 2
p = 1
r = 2
s = 1
q = 3
o = 2
M = [ 1 0.2 0; 2e-1 1 2E-1; 0 0.2 1 ]
E = veye( 0.1, 3 )
% could also be E = diag( [0.1 0.1 0.1] )
K = ndiag( [-1 -1 0; 1 2 2], [-1 0] )
B = eye( 3, 2 )
```

```
F = sparse( [ 1 2 3 ], [ 1 1 2 ], [ 1 1 1 ] )
C = sparse( [ 1 ], [ 2 ], [ 1 ], 1, 3 )
D = [ 1 ]
f = [ sin( u(1) + x(2) ); exp( u(2) / x(1) ) ]
g = [ floor( exp( x(3) * t ) * u(1) ) ]
x0 = zeros( 3, 1 )
v0 = [ 0 0 0 ]'
```

Table 12.3. Macros for entering matrices in a DSIF file. All forms with both `N` and `M` in their arguments return a possibly rectangular matrix with `N` rows and `M` columns; with `N` only, a square matrix is returned. In the following, `N`, `M` and `D` are scalars, `V`, `R` and `C` are row vectors, and `A` is matrix

`eye(N,M)` `eye(N)`	Returns the identity matrix
`veye(D,N,M)` `veye(D,N)`	Returns a matrix with `D` on the diagonal and 0 elsewhere
`zeros(N,M)` `zeros(N)`	Returns a matrix whose elements are all 0
`ones(N,M)` `ones(N)`	Returns a matrix whose elements are all 1
`rep(D,N,M)` `rep(D,N)`	Returns a matrix whose elements are all `D`
`repmat(A,N,M)` `repmat(A,N)`	Returns a block matrix with a copy of matrix `A` as each element.

`diag(V,D,N,M)`
`diag(V,D,N)`
`diag(V,D)`
`diag(V)`
	Returns a diagonal matrix with vector `V` on diagonal `D` (`D` > 0 is above the main diagonal, `D` < 0 below). If `N` and `M` are omitted, the matrix size is the minimal size to contain `V`. If `D` is omitted, it is assumed to be 0.

`ndiag(A,V,N,M)`
`ndiag(A,V,N)`
`ndiag(A,V)`
	The first argument of this function is a matrix of row vectors to be included as diagonals to the final matrix. Trailing unused places must be filled with zeros. The second argument is a row vector, whose elements specify at which diagonal to include them. Returns a matrix with each of the vectors in matrix `A` at the corresponding diagonal represented by the entry in vector `V`. If the matrix size is omitted, it is the minimal size to contain the diagonals.

`sparse(R,C,V,N,M)`
`sparse(R,C,V,N)`
`sparse(R,C,V)`
	This function allows to specify a sparse matrix. `R`, `C` and `V` list the row and column numbers and the corresponding nonzero value such that the resulting matrix m is $m_{\mathtt{R}(k),\mathtt{C}(k)} = \mathtt{V}_k$.

`A'` `V'`	Transpose of a matrix or vector

12.5 Conclusions

We have specified a file format for the exchange of a nonlinear system of ODEs. The format is similar to the MATLAB file format, allowing to read in the linear parts to MATLAB; it features a number of nonlinear functions and macros for matrix creation. We hope that it will serve the model order reduction community by promoting the creation of a large number of benchmarks to test MOR algorithms for the nonlinear case.

12.6 Acknowledgments

This work is partially funded by the DFG project MST-Compact (KO-1883/6), the Italian research council CNR together with the Italian province of Trento PAT, by the German Ministry of Research BMBF (SIMOD), and an operating grant of the University of Freiburg.

References

[Mat] Matlab 7, http://www.mathworks.com
[HU94] Huang, H.H., Usmani, A.S.: Finite Element Analysis for Heat Transfer. Springer, London (1994)
[ISO] ISO/IEC: ISO/IEC 9899:1999(E) Programming languages – C

13

Nonlinear Heat Transfer Modeling

Jan Lienemann[1], Amirhossein Yousefi[2], and Jan G. Korvink[1]

[1] Institute for Microsystem Technology, Albert Ludwig University
Georges Köhler Allee 103, 79110 Freiburg, Germany
{lieneman,korvink}@imtek.de
[2] Lehrstuhl für Regelungstechnik, Technische Universität München,
Boltzmannstr. 15, D-85748 Garching bei München, Germany
Yousefi@tum.de

Summary. The simulation of heat transport for a single device is easily tackled by current computational resources, even for a complex, finely structured geometry; however, the calculation of a multi-scale system consisting of a large number of those devices, e.g., assembled printed circuit boards, is still a challenge. A further problem is the large change in heat conductivity of many semiconductor materials with temperature. We model the heat transfer along a 1D beam that has a nonlinear heat capacity which is represented by a polynomial of arbitrary degree as a function of the temperature state. For accurate modeling of the temperature distribution, the resulting model requires many state variables to be described adequately. The resulting complexity, i.e., number of first order differential equations and nonlinear parts, is such that a simplification or model reduction is needed in order to perform a simulation in an acceptable amount of time for the applications at hand.

In this paper, we describe the modeling considerations leading to a large nonlinear system of equations. Sample results from this model and examples of successful model order reduction can be found in [YLLK04] and the corresponding benchmark document, available online on the Oberwolfach Model Reduction Benchmark Collection website [OBC] ("Nonlinear heat transfer modeling").

13.1 Modeling

We model the heat transfer along a 1D beam with length L, cross sectional area A and nonlinear heat conductivity κ. The heat conductivity is represented by a polynomial in temperature $T(x,t)$ of arbitrary degree n

$$\kappa(T) = a_0 + a_1 T + \cdots + a_n T^n = \sum_{i=0}^{n} a_i T^i. \tag{13.1}$$

The right end of the beam (at $x = L$) is fixed at ambient temperature. The model features two inputs, a time-dependent uniform heat flux f at the left

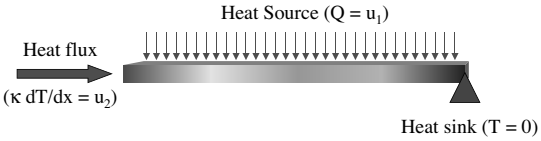


Fig. 13.1. The modeled beam with heat flux inputs and heat sink.

end (at $x = 0$) and a time dependant heat source Q along the beam. We denote the beam volume where we wish to solve the equations by Ω.

By including (13.1) in the differential form of the heat transfer equation,

$$-\nabla \cdot (\kappa(T)\nabla T) + \rho c_p \dot{T} = Q, \tag{13.2}$$

we obtain the following expression,

$$-\sum_{i=0}^{n} a_i \nabla \cdot \left(T^i \nabla T\right) + \rho c_p \dot{T} = Q, \tag{13.3}$$

where ρ is the density and c_p is the heat capacity, which are both assumed to be constant for the considered temperature range. This approximation can be justified from measurements of semiconductors, which show that the temperature dependency of c_p is much smaller than that of κ. This rapid change is a result of the special band structure of the material. It follows an exponential law:

$$\kappa = \kappa_0 \mathrm{e}^{\alpha(T-T_0)}. \tag{13.4}$$

The heat capacity for silicon changes from 1.3 to 2 in the temperature range of 200 to 600 Kelvin, while κ changes from 280 W/m K to 60 W/m K.

13.1.1 Finite Element Discretization

Following the Ritz-Galerkin finite element formulation, we require orthogonality with respect to a set of test functions $N_k(x)$, $k = 1, \ldots, N$:

$$-\sum_{i=0}^{n} a_i \int_\Omega N_k \nabla \cdot \left(T^i \nabla T\right) \mathrm{d}\Omega + \int_\Omega N_k \rho c_p \dot{T} \mathrm{d}\Omega = \int_\Omega N_k Q \mathrm{d}\Omega \quad \forall N. \tag{13.5}$$

By using the Green-Gauß theorem, we get the weak form

$$\begin{aligned} \sum_{i=0}^{n} a_i \int_\Omega \nabla N_k T^i \nabla T \mathrm{d}\Omega - \int_{\partial\Omega} \underbrace{\kappa(T)\nabla T \cdot \mathbf{n}}_{J} N_k \mathrm{d}\partial\Omega + \int_\Omega N_k \rho c_p \dot{T} \mathrm{d}\Omega \\ = \int_\Omega N_k Q \mathrm{d}\Omega, \end{aligned} \tag{13.6}$$

where a positive J denotes a heat flux into one end of the beam. We approximate the temperature profile by shape functions

$$T(x) = \sum_{j=1}^{N} T_j N_j(x), \tag{13.7}$$

which are the same as the test functions N_k and, after moving all inputs to the right side, obtain

$$\begin{aligned} \sum_{i=0}^{n} a_i \sum_{j=1}^{N} T_j \int_\Omega \nabla N_k T^i \nabla N_j \mathrm{d}\Omega + \rho c_p \sum_{j=1}^{N} \dot{T}_j \int_\Omega N_k N_j \mathrm{d}\Omega \\ = Q \int_\Omega N_k \mathrm{d}\Omega + J \int_{\partial\Omega} N_k \mathrm{d}\partial\Omega. \end{aligned} \tag{13.8}$$

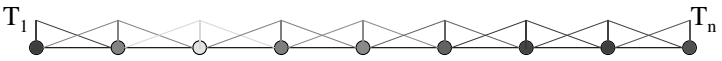


Fig. 13.2. Linear shape functions for FEM discretization

The second, third and fourth term in this equation are linear and yield a constant mass matrix M and a scattering matrix B on the right side to distribute the two inputs J and Q to the load vector. For a linear 1D beam element e of length l with nodes m and $m+1$, we have the element contributions

$$\mathsf{M}_e = \begin{bmatrix} 2/3 & 1/6 \\ 1/6 & 2/3 \end{bmatrix}, \quad \mathsf{B}_e = \begin{bmatrix} 0 & Al/2 \\ 0 & Al/2 \end{bmatrix} \tag{13.9a}$$

except for the leftmost element, where

$$\mathsf{B}_1 = \begin{bmatrix} A & Al/2 \\ 0 & Al/2 \end{bmatrix}. \tag{13.9b}$$

When using linear shape functions, the gradients are constant. The element stiffness matrix then reads

$$\mathsf{A}_e = \sum_{i=0}^{n} a_i \frac{A}{l^2} \int_0^l \left(T_m(1 - x/l) + T_{m+1} x/l\right)^i \mathrm{d}x \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \tag{13.10a}$$

$$= \sum_{i=0}^{n} a_i \frac{A}{l} \frac{T_{m+1}^{i+1} - T_m^{i+1}}{(i+1)(T_{m+1} - T_m)} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}. \tag{13.10b}$$

For $i > 0$, this yields a nonlinear stiffness matrix, while for $i = 0$ after performing the multiplication of the matrix A with $\mathbf{x}$, the denominator is constant. We introduce a vector $\mathbf{f}(T)$ on the right side which collects all nonlinear parts of the discretized equation:

$$\mathsf{A}_{\text{linear}} \mathbf{T} + \rho c_p \mathsf{M} \dot{\mathbf{T}} = \mathsf{B} \begin{pmatrix} J \\ Q \end{pmatrix} + \mathbf{f}(\mathbf{T}). \tag{13.11}$$

To move the nonlinear terms in (13.10b) to the right side, we multiply them with $T_m - T_{m+1}$ and subtract them from both sides of the equation. Every element e contributes two entries to the vector $\mathbf{f}(T)$:

$$\mathbf{f}_e = \sum_{i=1}^{n} a_i \frac{A}{l} \frac{T_{m+1}^{i+1} - T_m^{i+1}}{i+1} \begin{pmatrix} 1 \\ -1 \end{pmatrix}. \tag{13.12}$$

We observe that the nonlinearities are polynomial.

We then denote $\mathsf{E} = \rho c_p \mathsf{M}$ and introduce a gather matrix C which returns some linear combinations of the degrees of freedom (or more often, selects some single DOFs) which are the most interesting for the application. In this particular example, C is a row vector with 1 at the first position, 1 at the entry in the middle ($\lceil n/2 \rceil$) and 0 everywhere else. This returns the temperatures at the leftmost end (where the heat flux is applied) and in the middle of the beam.

After renaming $\mathbf{T}$ to $\mathbf{x}$ to comply with the DSI file format specifications described in Chapter 12, we end up with the following system of equations:

$$\mathsf{E}\dot{\mathbf{x}} + \mathsf{A}\mathbf{x} = \mathsf{B}\mathbf{u} + \mathsf{F}\mathbf{f}(\mathbf{x}, u) \tag{13.13}$$

$$\mathbf{y} = \mathsf{C}\mathbf{x} \tag{13.14}$$

13.1.2 Implementation

The scheme above was implemented in the computer algebra system *Mathematica* [Mat]. *Mathematica*'s symbolic capabilities allow for an easy implementation of vectors of nonlinear functions. The data is then exported to a file in the DSI format; see Chapter 12. We have also created an interactive web application which allows one to specify the parameters of the model for customized matrix generation, available on [Mst].

A number of linear and nonlinear precomputed examples are available from the benchmark.

13.2 Discussion and Conclusion

A general model for the heat conduction with temperature dependent heat conductivity in a 1D beam was developed. It is possible to include polynomial nonlinearities with an arbitrary polynomial degree. The effects of nonlinearities are clearly visible from simulation results.

13.3 Acknowledgments

This work was supported by Deutsche Forschungsgemeinschaft (DFG) project MST-Compact under contract numbers KO-1883/6 and LO 408/3-1 and by an operating grant of the University of Freiburg.

References

[YLLK04] Yousefi, A., Lienemann, J., Lohmann, B., Korvink, J.G.: Nonlinear Heat Transfer Modelling and Reduction. In: Proceedings of the 12th Mediterranean Conference on Control and Automation, Kusadasi, Aydin, Turkey, June 6–9 (2004)

[OBC] http://www.imtek.uni-freiburg.de/simulation/benchmark/

[Mst] http://www.imtek.uni-freiburg.de/simulation/mstkmpkt.

[Mat] http://www.wolfram.com.

14

Microhotplate Gas Sensor

Jürgen Hildenbrand[1], Tamara Bechtold[1], and Jürgen Wöllenstein[2]

[1] Institute for Microsystem Technology, Albert Ludwig University
Georges Köhler Allee 103, 79110 Freiburg, Germany
{hildenbr,bechtold}@imtek.uni-freiburg.de
[2] Institute for Physical Measurements Techniques
Heidenhofstr. 8, 79110, Freiburg
woellen@ipm.fhg.de

Summary. A benchmark for the heat transfer problem, related to modeling of a microhotplate gas sensor, is presented. It can be used to apply model reduction algorithms to a linear first-order problem as well as when an input function is nonlinear.

14.1 Modeling

The goal of European project Glassgas (IST-99-19003) was to develop a novel metal oxide low power microhotplate gas sensor [WBP03]. In order to assure a robust design and good thermal isolation of the membrane from the surrounding wafer, the silicon microhotplate is supported by glass pillars emanating from a glass cap above the silicon wafer, as shown in Figure 14.1. In present design, four different sensitive layers can be deposited on the membrane. The thermal management of a microhotplate gas sensor is of crucial importance.

The benchmark contains a thermal model of a single gas sensor device with three main components: a silicon rim, a silicon hotplate and glass structure [Hil03]. It allows us to simulate important thermal issues, such as the homogeneous temperature distribution over gas sensitive regions or thermal decoupling between the hotplate and the silicon rim. The original model is the heat transfer partial differential equation

$$\nabla \cdot (\kappa(\mathbf{r})\nabla T(\mathbf{r},t)) + Q(\mathbf{r},t) - \rho(\mathbf{r})C_p(\mathbf{r})\frac{\partial T(\mathbf{r},t)}{\partial t} = 0 \qquad (14.1)$$

where $\mathbf{r}$ is the position, t is the time, κ is the thermal conductivity of the material, C_p is the specific heat capacity, ρ is the mass density, Q is the heat generation rate, that is nonzero only within the heater, and T is the unknown temperature distribution to be determined.

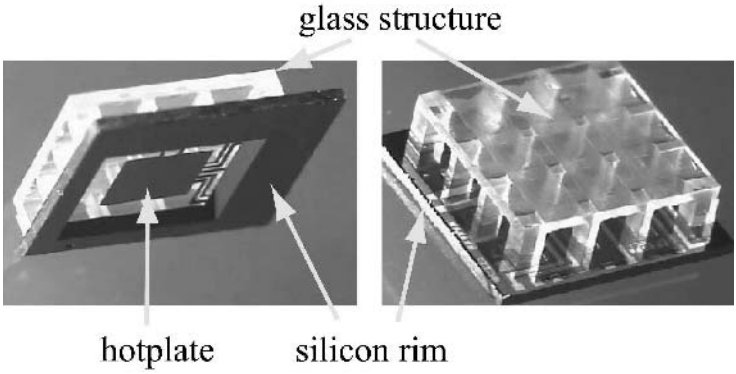


Fig. 14.1. Micromachined metal oxide gas sensor array; Bottom view (left), top view (right).

14.2 Discretization

The device solid model has been made and then meshed and discretized in ANSYS 6.1 by means of the finite element method (SOLID70 elements were used). It contains 68000 elements and 73955 nodes. Material properties were considered as temperature independent. Temperature is assumed to be in degree Celsius with the initial state of 0°C. The Dirichlet boundary conditions of $T = 0$°C is applied at the top and bottom of the chip (at 7038 nodes).

The output nodes are described in Table 14.1. In Figure 14.2 the nodes 2 to 7 are positioned on the silicon rim. Their temperature should be close to the initial temperature in the case of good thermal decoupling between the membrane and the silicon rim. Other nodes are placed on the sensitive layers above the heater and are numbered from left to right row by row, as schematically shown in Fig 14.2. They allow us to prove whether the temperature distribution over the gas sensitive layers is homogeneous (maximum difference of 10°C is allowed by design).

Table 14.1. Outputs for the gas sensor model

Number	Code	Comment
1	aHeater	within a heater, to be used for nonlinear input
2-7	SiRim1 to SiRim7	silicon rim
8-28	Memb1 to Memb21	gas sensitive layer

The benchmark contains a constant load vector. The input function equal to 1 corresponds to the constant input power of 340mW. One can insert a weak input nonlinearity related to the dependence of heater's resistivity on temperature given as:

$$R(T) = R_0(1 + \alpha T) \tag{14.2}$$

where $\alpha = 1.469 \times 10^{-3} K^{-1}$. To this end, one has to multiply the load vector by a function:

$$\frac{U^2 274.94(1 + \alpha T)}{0.34(274.94(1 + \alpha T) + 148.13)^2} \tag{14.3}$$

where U is a desired constant voltage. The temperature in (14.3) should be replaced by the temperature at the output 1.

The linear ordinary differential equations of the first order are written as:

$$\begin{aligned} E\dot{x} &= Ax + Bu \\ y &= Cx \end{aligned} \tag{14.4}$$

where E and A are the symmetric sparse system matrices (heat capacity and heat conductivity matrix), B is the load vector, C is the output matrix, and x is the vector of unknown temperatures. The dimension of the system is 66917, the number of nonzero elements in matrix E is 66917, in matrix A is 885141.

The outputs of the transient simulation at output 18 (Memb11) over the rise time of the device of 5 s for the original linear (with constant input power of 340 mW) and nonlinear (with constant voltage of 14 V) model are placed

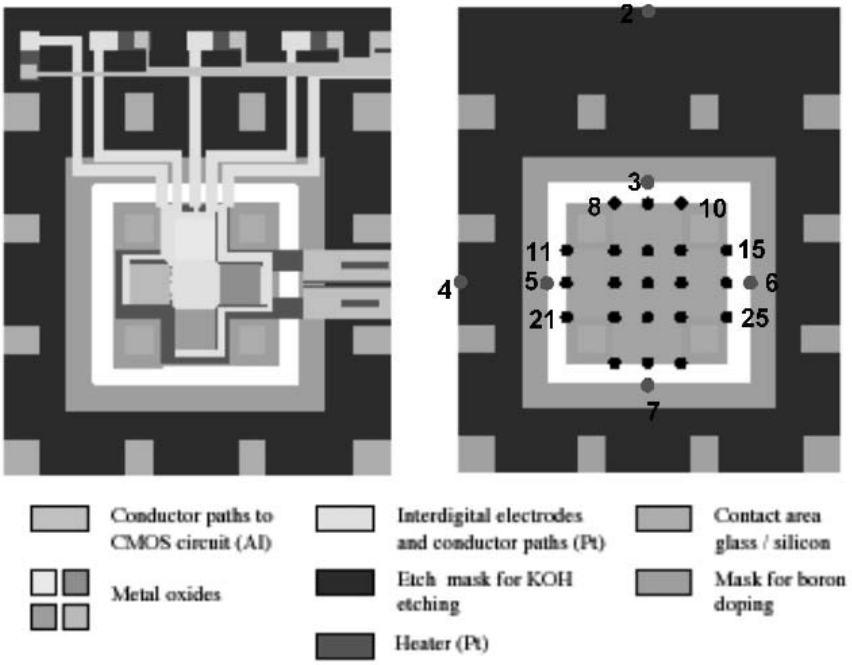


Fig. 14.2. Masks disposition (left) and the schematical position of the chosen output nodes (right).

in files `LinearResults` and `NonlinearResults`, respectively. The results can be used to compare the solution of a reduced model with the original one. The time integration has been performed in ANSYS with accuracy of about 0.001. The results are given as matrices where the first row is made of times, the second of the temperatures.

The discussion of electro-thermal modeling related to the benchmark including the nonlinear input function can be found in [BHWK04].

14.3 Acknowledgments

This work is partially funded by the DFG project MST-Compact (KO-1883/6) and an operating grant of the University of Freiburg.

References

[WBP03] Wöllenstein, J., Böttner, H., Pláza, J.A., Carné, C., Min, Y., Tuller H. L.: A novel single chip thin film metal oxide array. Sensors and Actuators B: Chemical **93**, (1-3) 350–355 (2003)

[Hil03] Hildenbrand, J.: Simulation and Characterisation of a Gas sensor and Preparation for Model Order Reduction. Diploma Thesis, University of Freiburg, Germany (2003)

[BHWK04] Bechtold, T., Hildenbrand, J., Wöllenstein, J., Korvink, J. G.: Model Order Reduction of 3D Electro-Thermal Model for a Novel, Micromachined Hotplate Gas Sensor. In: Proceedings of 5th International conference on thermal and mechanical simulation and experiments in microelectronics and microsystems, EUROSIME2004, May 10-12, Brussels, Belgium, pp. 263–267 (2004)

15

Tunable Optical Filter

Dennis Hohlfeld, Tamara Bechtold, and Hans Zappe

Institute for Microsystem Technology, Albert Ludwig University
Georges Köhler Allee 103, 79110 Freiburg, Germany
{hohlfeld,bechtold,zappe}@imtek.uni-freiburg.de

Summary. A benchmark for the heat transfer problem, related to modeling of a tunable optical filter, is presented. It can be used to apply model reduction algorithms to a linear first-order problem.

15.1 Modeling

The DFG project AFON aimed at the development of an optical filter, which is tunable by thermal means. The thin-film filter is configured as a membrane (see Figure 15.1) in order to improve thermal isolation. Fabrication is based on silicon technology. Wavelength tuning is achieved through thermal modulation of resonator optical thickness, using metal resistor deposited onto the membrane. The devices features low power consumption, high tuning speed and excellent optical performance [HZ03].

The benchmark contains a simplified thermal model of a filter device. It helps designers to consider important thermal issues, such as what electrical power should be applied in order to reach the critical temperature at the membrane or homogeneous temperature distribution over the membrane. The original model is the heat transfer partial differential equation

$$\nabla \cdot (\kappa(\mathbf{r})\nabla T(\mathbf{r},t)) + Q(\mathbf{r},t) - \rho(\mathbf{r})C_p(\mathbf{r})\frac{\partial T(\mathbf{r},t)}{\partial t} = 0 \qquad (15.1)$$

where $\mathbf{r}$ is the position, t is the time, κ is the thermal conductivity of the material, C_p is the specific heat capacity, ρ is the mass density, Q is the heat generation rate that is nonzero only within the heater, and T is the unknown temperature distribution to be determined. There are two different benchmarks, 2D model and 3D model (see Table 15.1). Due to modeling differences, their simulation results cannot be compared with each other directly.

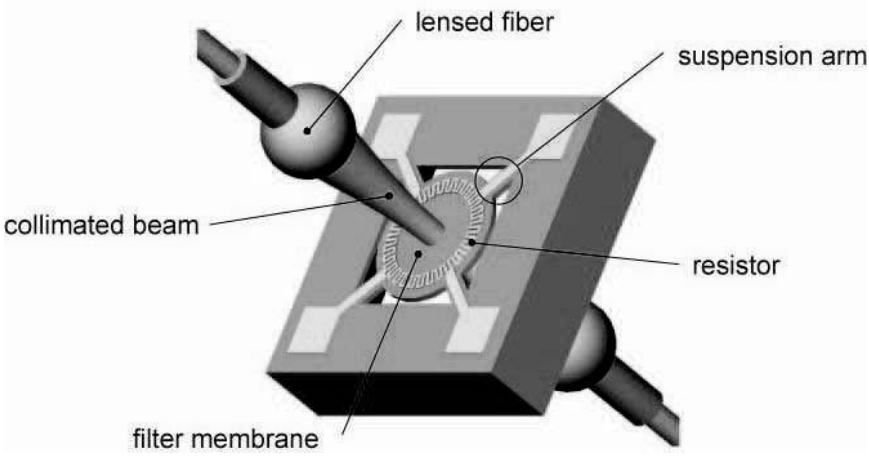


Fig. 15.1. Tunable optical filter.

Table 15.1. Tunable optical filter benchmarks

Code	comment	dimension	nnz(A)	nnz(E)
filter2D	2D, linear elements, PLANE55	1668	6209	1668
filter3D	3D, linear elements, SOLID90	108373	1406808	1406791

15.2 Discretization

The device solid models have been made, meshed and discretized in ANSYS 6.1 by the finite element method. All material properties are considered as temperature independent. Temperature is assumed to be in Celsius with the initial state of 0°C. The Dirichlet boundary conditions of $T = 0$°C have been applied at the bottom of the chip. The output nodes for the models are described in Table 15.2 and schematically displayed in Figure 15.2. Output 1 is located at the very center of the membrane. By simulating its temperature one can prove what input power is needed to reach the critical membrane temperature for each wavelength. Furthermore, the output 2 to 5 must be very close to output 1 (homogenous temperature distribution) in order to provide the same optical properties across the complete diameter of the laser beam.

The benchmark contains a constant load vector. The input function equal to 1 corresponds to the constant input power of of 1 mW for 2D model and 10 mW for 3D model. The linear ordinary differential equations of the first order are written as:

$$\begin{aligned} E\dot{x} &= Ax + Bu \\ y &= Cx \end{aligned} \tag{15.2}$$

Table 15.2. Outputs for the optical filter model

Number	Code	Comment
1	Memb1	Membrane center
2	Memb2	Membrane node with radius 25E-6, theta 90°
3	Memb3	Membrane node with radius 50E-6 theta 90°
4	Memb4	Membrane node with radius 25E-6, theta 135°
5	Memb5	Membrane node with radius 50E-6 theta 135°

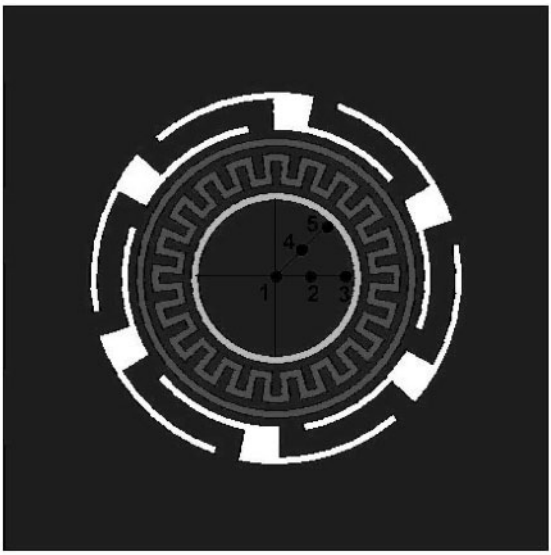


Fig. 15.2. Schematic position of the chosen output nodes.

where E and A are the symmetric sparse system matrices (heat capacity and heat conductivity matrix), B is the load vector, C is the output matrix, and x is the vector of unknown temperatures.

The output of the transient simulation for node 1 over the rise time of the device (0.25 s) for 3D model can be found in `Filter3DTransResults`. The results can be used to compare the solution of a reduced model with the original one. The time integration has been performed in ANSYS with accuracy of about 0.001. The results are given as matrices where the first row is made of times, the second of the temperatures.

The discussion of electro-thermal modeling related to the benchmark can be found in [Bec05].

15.3 Acknowledgments

This work is partially funded by the DFG projects AFON (ZA 276/2-1), MST-Compact (KO-1883/6) and an operating grant of the University of Freiburg.

References

[HZ03] Hohlfeld, D., Zappe, H.: All-dielectric tunable optical filter based on the thermo-optic effect. Journal of Optics A: Pure and Applied Optics, **6**(6), 504–511 (2003)

[Bec05] Bechtold, T.: Model Order Reduction of Electro-Thermal MEMS. PhD thesis, University of Freiburg, Germany (2005)

16

Convective Thermal Flow Problems

Christian Moosmann and Andreas Greiner

Institute for Microsystem Technology, Albert Ludwig University
Georges Köhler Allee 103, 79110 Freiburg, Germany
{moosmann,greiner}@imtek.uni-freiburg.de

Summary. A benchmark for the convective heat transfer problem, related to modeling of a anemometer and a chip cooled by forced convection, is presented. It can be used to apply model reduction algorithms to a linear first-order problem.

16.1 Modeling

Many thermal problems require simulation of heat exchange between a solid body and a fluid flow. The most elaborate approach to this problem is computational fluid dynamics (CFD). However, CFD is computationally expensive. A popular solution is to exclude the flow completely from the computational domain and to use convection boundary conditions for the solid model. However, caution has to be taken to select the film coefficient.

An intermediate level is to include a flow region with a given velocity profile, that adds convective transport to the model. The partial differential equation for the temperature T in this case reads:

$$\rho c \left(\frac{\partial T}{\partial t} + \mathbf{v} \nabla T \right) + \nabla \cdot (-\kappa \nabla T) = \dot{q} \tag{16.1}$$

where ρ is the mass density, c is the specific heat of the fluid, $\mathbf{v}$ is the fluid speed, κ is the thermal conductivity, $\dot{q}$ is the heat generation rate.

Compared to convection boundary conditions this approach has the advantage that the film coefficient does not need to be specified and that information about the heat profile in the flow can be obtained. A drawback of the method is the greatly increased number of elements needed to perform a physically valid simulation, because the solution accuracy when employing upwind finite element schemes depends on the element size. While this problem still is linear, due to the forced convection, the conductivity matrix changes from a symmetric matrix to an un-symmetric one. So this problem type can be used as a benchmark for problems containing un-symmetric matrices.

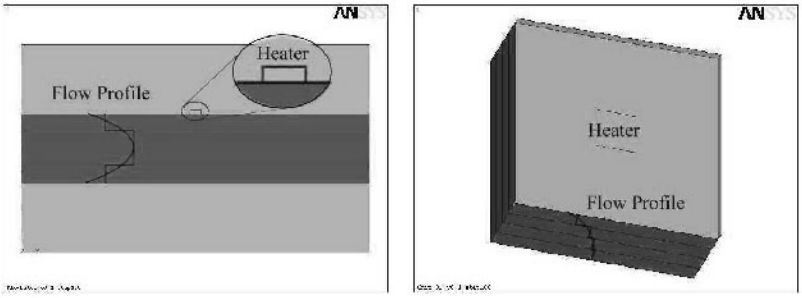


Fig. 16.1. Convective heat flow examples: 2D anemometer model (left), 3D cooling structure (right)

16.2 Discretization

Two different designs are tested: a 2D model of an anemometer-like structure mainly consisting of a tube and a small heat source (Figure 16.1 left) [Ern01]. The solid model has been generated and meshed in ANSYS. Triangular PLANE55 elements have been used for meshing and discretizing by the finite element method, resulting in 19 282 elements and 9710 nodes. The second design is a 3D model of a chip cooled by forced convection (Figure 16.1 right) [Har97]. In this case the tetrahedral element type SOLID70 was used, resulting in 107 989 elements and 20542 nodes. Since the implementation of the convective term in ANSYS does not allow for definition of the fluid speed on a per element, but on a per region basis, the flow profile has to be approximated by piece-wise step functions. The approximation used for this benchmarks is shown in figure 16.1.

The Dirichlet boundary conditions are applied to the original system. In both models the reference temperature is set to 300 K, Dirichlet boundary conditions as well as initial conditions are set to 0 with respect to the reference. The specified Dirichlet boundary conditions are in both cases the inlet of the fluid and the outer faces of the solids. Matrices are supplied for the symmetric case (fluid speed is zero; no convection), and the unsymmetric case (with forced convection). Table 16.1 shows the output nodes specified for the two benchmarks, table 16.2 shows the filenames according to the different cases.

Further information on the models can be found in [MRGK04] where model reduction by means of the Arnoldi algorithm is also presented.

16.3 Acknowledgments

This work is partially funded by the DFG project MST-Compact (KO-1883/6), the Italian research council CNR together with the Italian province of Trento PAT, by the German Ministry of Research BMBF (SIMOD), and an operating grant of the University of Freiburg.

Table 16.1. Output nodes for the two models

Model	Number	Code	Comment
Flow Meter	1	out1	outlet position
	2	out2	outlet position
	3	SenL	left sensor position
	4	Heater	within the heater
	5	SenR	right sensor position
cooling Structure	1	out1	outlet position
	2	out2	outlet position
	3	out3	outlet position
	4	out4	outlet position
	5	Heater	within the heater

Table 16.2. Provided files

Model	fluid speed (m/s)	Filenames
Flow Meter	0	flow_meter_model_v0.*
	0.5	flow_meter_model_v0.5.*
cooling Structure	0	chip_cooling_model_v0.*
	0.1	chip_cooling_model_v0.1.*

References

[Ern01] Ernst, H.: High-Resolution Thermal Measurements in Fluids. PhD thesis, University of Freiburg, Germany (2001)

[Har97] Harper, C. A.: Electronic packaging and interconnection handbook. New York McGraw-Hill, USA (1997)

[MRGK04] Moosmann, C., Rudnyi, E.B., Greiner, A., Korvink, J.G.: Model Order Reduction for Linear Convective Thermal Flow. In: Proceedings of 10th International Workshops on THERMal INvestigations of ICs and Systems, THERMINIC2004, 29 Sept - 1 Oct , Sophia Antipolis, France, p. 317-322 (2004)

17

Boundary Condition Independent Thermal Model

Evgenii B. Rudnyi and Jan G. Korvink

Institute for Microsystem Technology, Albert Ludwig University
Georges Köhler Allee 103, 79110 Freiburg, Germany
{rudnyi,korvink}@imtek.uni-freiburg.de

Summary. A benchmark for the heat transfer problem with variable film coefficients is presented. It can be used to apply parametric model reduction algorithms to a linear first-order problem.

17.1 Modeling

One of important requirements for a compact thermal model is that it should be boundary condition independent. This means that a chip producer does not know conditions under which the chip will be used and hence the chip compact thermal model must allow an engineer to research on how the change in the environment influences the chip temperature. The chip benchmarks representing boundary condition independent requirements are described in [Las01].

Let us briefly describe the problem mathematically. The thermal problem can be modeled by the heat transfer partial differential equation

$$\nabla \cdot (\kappa(\mathbf{r})\nabla T(\mathbf{r},t)) + Q(\mathbf{r},t) - \rho(\mathbf{r})C_p(\mathbf{r})\frac{\partial T(\mathbf{r},t)}{\partial t} = 0 \tag{17.1}$$

with $\mathbf{r}$ is the position, t is the time, κ is the thermal conductivity of the material, C_p is the specific heat capacity, ρ is the mass density, Q is the heat generation rate, and T is the unknown temperature distribution to be determined. The heat exchange through device interfaces is usually modeled by convection boundary conditions

$$q = h_i(T - T_{bulk}) \tag{17.2}$$

where q is the heat flow through a given point, h_i is the film coefficient to describe the heat exchange for the i-th interface, T is the local temperature at this point and T_{bulk} is the bulk temperature in the neighboring phase (in most cases $T_{bulk} = 0$).

After the discretization of Equations (17.1) and (17.2) one obtains a system of ordinary differential equations as follows

$$E\dot{x} = (A - \sum_i h_i A_i)x + Bu \tag{17.3}$$

where E, A are the device system matrices, A_i is the matrix resulting from the discretization of Equation (17.2) for the i-th interface, x is the vector with unknown temperatures.

In terms of Equation (17.3), the engineering requirements specified above read as follows. A chip producer specifies the system matrices but the film coefficient, h_i, is controlled later on by another engineer. As such, any reduced model to be useful should preserve h_i in the symbolic form. This problem can be mathematically expressed as parametric model reduction [WMGG99, GKN03, DSC04].

Unfortunately, the benchmark from [Las01] is not available in the computer readable format. For research purposes, we have modified a microthruster benchmark [LRK04] (see Figure 17.1). In the context of the present work, the model is as a generic example of a device with a single heat source when the generated heat dissipates through the device to the surroundings. The exchange between surrounding and the device is modeled by convection boundary conditions with different film coefficients at the top, h_{top}, bottom, h_{bottom}, and the side, h_{side}. From this viewpoint, it is quite similar to a chip model used as a benchmark in [Las01]. The goal of parametric model reduction in this case is to preserve h_{top}, h_{bottom}, and h_{side} in the reduced model in the symbolic form.

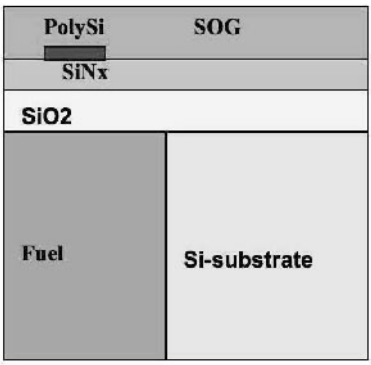


Fig. 17.1. A 2D-axisymmetrical model of the micro-thruster unit (not scaled). The axis of the symmetry on the left side. A heater is shown by a red spot.

17.2 Discretization

We have used a 2D-axisymmetric microthruster model (T2DAL in [LRK04]). The model has been made in ANSYS and system matrices have been extracted by means of mor4ansys [RK04]. The benchmark contains a constant load vector. The input function equal to one corresponds to the constant input power of 15 mW.

The linear ordinary differential equations of the first order are written as:

$$\begin{aligned} E\dot{x} &= (A - h_{top}A_{top} - h_{bottom}A_{bottom} - h_{side}A_{side})x + Bu \\ y &= Cx \end{aligned} \tag{17.4}$$

where E and A are the symmetric sparse system matrices (heat capacity and heat conductivity matrix), B is the load vector, C is the output matrix, A_{top}, A_{bottom}, and A_{side} are the diagonal matrices from the discretization of the convection boundary conditions and x is the vector of unknown temperatures.

The numerical values of film coefficients can be from 1 to 10^9. Typical important sets of film coefficients can be found in [Las01]. The allowable approximation error is 5 % [Las01].

The benchmark has been used in [FRK04a, FRK04b] where the problem is also described in more detail.

17.3 Acknowledgments

This work is partially funded by the DFG project MST-Compact (KO-1883/6), the Italian research council CNR together with the Italian province of Trento PAT, by the German Ministry of Research BMBF (SIMOD), and an operating grant of the University of Freiburg.

References

[Las01] Lasance, C. J. M.: Two benchmarks to facilitate the study of compact thermal modeling phenomena. IEEE Transactions on Components and Packaging Technologies, **24**, 559–565 (2001)

[WMGG99] Weile, D.S., Michielssen, E., Grimme, E., Gallivan, K.: A method for generating rational interpolant reduced order models of two-parameter linear systems. Applied Mathematics Letters, **12**, 93–102 (1999)

[GKN03] Gunupudi, P.K., Khazaka, R., Nakhla, M.S., Smy, T., Celo, D.: Passive parameterized time-domain macromodels for high-speed transmission-line networks. IEEE Transactions on Microwave Theory and Techniques, **51**, 2347–2354 (2003)

[DSC04] Daniel, L., Siong, O.C., Chay, L.S., Lee, K.H., White J.: A Multiparameter Moment-Matching Model-Reduction Approach for Generating Geometrically Parameterized Interconnect Performance Models. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, **23**, 678–693 (2004)

[LRK04] Lienemann, J., Rudnyi, E.B., Korvink, J.G.: MST MEMS model order reduction: Requirements and Benchmarks. Linear Algebra and its Applications, to appear.

[RK04] Rudnyi, E.B., Korvink, J.G.: Model Order Reduction of MEMS for Efficient Computer Aided Design and System Simulation. In: MTNS2004, Sixteenth International Symposium on Mathematical Theory of Networks and Systems, Katholieke Universiteit Leuven, Belgium, July 5-9 (2004)

[FRK04a] Feng, L., Rudnyi, E.B., Korvink, J.G.: Parametric Model Reduction to Generate Boundary Condition Independent Compact Thermal Model. In: Proceedings of 10th International Workshops on THERMal INvestigations of ICs and Systems, THERMINIC2004, 29 Sept - 1 Oct , Sophia Antipolis, France, p. 281-285 (2004)

[FRK04b] Feng, L., Rudnyi, E.B., Korvink, J.G.: Preserving the film coefficient as a parameter in the compact thermal model for fast electro-thermal simulation. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, to appear.

18

The Butterfly Gyro

Dag Billger

The Imego Institute
Arvid Hedvalls Backe 4, SE-411 33
Göteborg, Sweden
dag.billger@imego.com

Summary. A benchmark for structural mechanics, related to modeling of a microgyroscope, is presented. It can be used to apply model reduction algorithms to a linear second-order problem.

18.1 Brief Project Overview

The Butterfly gyro is developed at the Imego Institute in an ongoing project with Saab Bofors Dynamics AB. The Butterfly is a vibrating micro-mechanical gyro that has sufficient theoretical performance characteristics to make it a promising candidate for use in inertial navigation applications. The goal of the current project is to develop a micro unit for inertial navigation that can be commercialized in the high-end segment of the rate sensor market. This project has reached the final stage of a three-year phase where the development and research efforts have ranged from model based signal processing, via electronics packaging to design and prototype manufacturing of the sensor element. The project has also included the manufacturing of an ASIC, named μSIC, that has been especially designed for the sensor (Figure 18.1).

The gyro chip consists of a three-layer silicon wafer stack, in which the middle layer contains the sensor element. The sensor consists of two wing pairs that are connected to a common frame by a set of beam elements (Figure 18.2 and 18.3); this is the reason the gyro is called the Butterfly. Since the structure is manufactured using an anisotropic wet-etch process, the connecting beams are slanted. This makes it possible to keep all electrodes, both for capacitive excitation and detection, confined to one layer beneath the two wing pairs. The excitation electrodes are the smaller dashed areas shown in Figure 18.2. The detection electrodes correspond to the four larger ones.

By applying DC-biased AC-voltages to the four pairs of small electrodes, the wings are forced to vibrate in anti-phase in the wafer plane. This is the excitation mode. As the structure rotates about the axis of sensitivity (Figure 18.2), each of the masses will be affected by a Coriolis acceleration. This

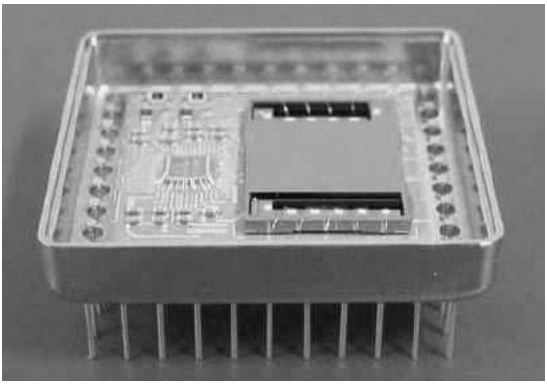


Fig. 18.1. The Butterfly and μSIC mounted together.

acceleration can be represented as an inertial force that is applied at right angles with the external angular velocity and the direction of motion of the mass. The Coriolis force induces an anti-phase motion of the wings out of the wafer plane. This is the detection mode. The external angular velocity can be related to the amplitude of the detection mode, which is measured via the large electrodes.

The partial differential equation for the displacement field of the gyro is governed by the standard linear equations of three-dimensional elastodynamics:

$$\sigma_{ij,j} + f_i = \rho \ddot{u}_i, \tag{18.1}$$

where ρ is the mass density, σ_{ij} is the stress tensor, f_i represents external loads (such as Coulomb forces) and u_i are the components of the displacement field. The constitutive stress-strain relation of a linear, anisotropic solid is given by

$$\sigma_{ij} = \frac{1}{2} C_{ijkl} \left(u_{i,j} + u_{j,i} \right), \tag{18.2}$$

where C_{ijkl} is the elastic moduli tensor.

18.2 The Benefits of Model Order Reduction

When planning for and making decisions on future improvements of the Butterfly, it is of importance to improve the efficiency of the gyro simulations. Repeated analyses of the sensor structure have to be conducted with respect to a number of important issues. Examples of such are sensitivity to shock, linear and angular vibration sensitivity, reaction to large rates and/or acceleration, different types of excitation load cases and the effect of force-feedback.

The use of model order reduction indeed decreases runtimes for repeated simulations. Moreover, the reduction technique enables a transformation of

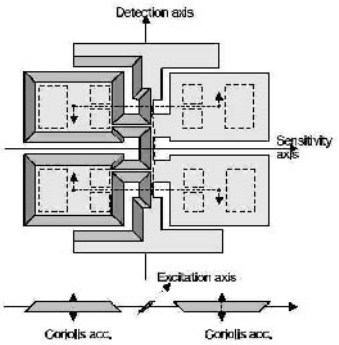


Fig. 18.2. Schematic layout of the Butterfly design.

the FE representation of the gyro into a state space equivalent formulation. This will prove helpful in testing the model based Kalman signal processing algorithms that are being designed for the Butterfly gyro.

The structural model of the gyroscope has been done in ANSYS using quadratic tetrahedral elements (SOLID187, Figure 18.3). The model shown is a simplified one with a coarse mesh as it is designed to test the model reduction approaches. It includes the pure structural mechanics problem only. The load vector is composed from time-varying nodal forces applied at the centers of the excitation electrodes (Figure 18.2). The amplitude and frequency of each force is equal to 0.055 μN and 2384 Hz, respectively. The Dirichlet boundary conditions have been applied to all DOFs of the nodes belonging to the top and bottom surfaces of the frame. The output nodes are listed in Table 18.2 and correspond to the centers of the detection electrodes.

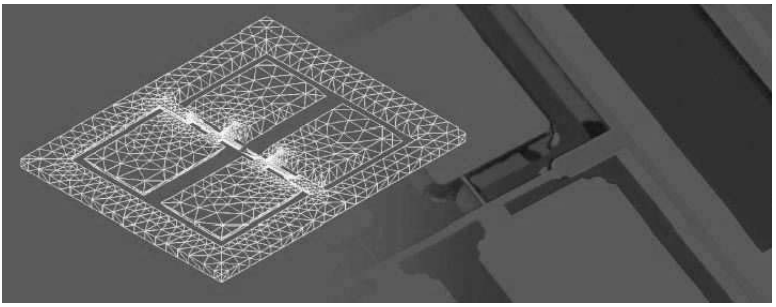


Fig. 18.3. Finite element mesh of the gyro with a background photo of the gyro wafer pre-bonding.

The discretized structural model

$$\begin{aligned} M\ddot{x} + E\dot{x} + Kx &= Bu \\ y &= Cx \end{aligned} \tag{18.3}$$

contains the mass and stiffness matrices. The damping matrix is modeled as $\alpha\mathbb{M} + \beta\mathbb{K}$, where the typical values are $\alpha = 0$ and $\beta = 10^{-6}$, respectively. The nature of the damping matrix is in reality more complex (squeeze film damping, thermo-elastic damping, etc.) but this simple approach has been chosen with respect to the model reduction benchmark.

The dynamic model has been converted to Matrix Market format by means of mor4ansys. The statistics for the matrices is shown in Table 18.1.

Table 18.1. System matrices for the gyroscope.

matrix	m	n	nnz	Is symmetric?
$\mathbb{M}$	17361	17361	178896	yes
$\mathbb{K}$	17361	17361	519260	yes
$\mathbb{B}$	17361	1	8	no
$\mathbb{C}$	12	17361	12	no

Table 18.2. Outputs for the Butterfly Gyro Model.

#	Code	Comment
1-3	det1m_Ux, det1m_Uy, det1m_Uz	Displ. of det. elect. 1, hardpoint #601
4-6	det1p_Ux, det1p_Uy, det1p_Uz	Displ. of det. elect. 2, hardpoint #602
7-9	det2m_Ux, det2m_Uy, det2m_Uz	Displ. of det. elect. 3, hardpoint #603
10-12	det2p_Ux, det2p_Uy, det2p_Uz	Displ. of det. elect. 4, hardpoint #604

The benchmark has been used in [LDR04] where the problem is also described in more detail.

References

[LDR04] Lienemann, J., Billger, D., Rudnyi, E.B., Greiner, A., Korvink, J.G.: MEMS Compact Modeling Meets Model Order Reduction: Examples of the Application of Arnoldi Methods to Microsystem Devices. In: The Technical Proceedings of the 2004 Nanotechnology Conference and Trade Show, Nanotech 2004, March 7-1, Boston, Massachusetts, USA, vol. 2, p. 303-306 (2004)

19

A Semi-Discretized Heat Transfer Model for Optimal Cooling of Steel Profiles

Peter Benner[1] and Jens Saak[1]

Fakultät für Mathematik, TU Chemnitz, 09107 Chemnitz, Germany.
{benner,jens.saak}@mathematik.tu-chemnitz.de

Summary. Several generalized state-space models arising from a semi-discretization of a controlled heat transfer process for optimal cooling of steel profiles are presented. The model orders differ due to different levels of refinement applied to the computational mesh.

19.1 The Model Equations

We consider the problem of optimal cooling of steel profiles. This problem arises in a rolling mill when different steps in the production process require different temperatures of the raw material. To achieve a high production rate, economical interests suggest to reduce the temperature as fast as possible to the required level before entering the next production phase. At the same time, the cooling process, which is realized by spraying cooling fluids on the surface, has to be controlled so that material properties, such as durability or porosity, achieve given quality standards. Large gradients in the temperature distributions of the steel profile may lead to unwanted deformations, brittleness, loss of rigidity, and other undesirable material properties. It is therefore the engineers goal to have a preferably even temperature distribution. For a picture of a such cooling plant see Figure 19.1.

The scientific challenge here is to give the engineers a tool to pre-calculate different control laws yielding different temperature distributions in order to decide which cooling strategy to choose.

We can only briefly introduce the model here; for details we refer to [Saa03] or [BS04]. We assume an infinitely long steel profile so that we may restrict ourselves to a 2D model. Exploiting the symmetry of the workpiece, the computational domain $\Omega \subset \mathbb{R}^2$ is chosen as the half of a cross section of the rail profile. The heat distribution is modeled by the instationary linear heat equation on Ω:

$$\begin{aligned} c\varrho\partial_t x(t,\xi) - \lambda\Delta x(t,\xi) &= 0 && \text{in } \mathbb{R}_{>0}\times\Omega,\\ x(0,\xi) &= x_0(\xi) && \text{in } \Omega,\\ \lambda\partial_\nu x(t,\xi) &= g_i && \text{on } \mathbb{R}_{>0}\times\Gamma_i, \partial\Omega = \bigcup_i \Gamma_i, \end{aligned} \tag{19.1}$$

where x is the temperature distribution ($x \in H^1([0,\infty], X)$ with $X := H^1(\Omega)$ being the state space), c the specific heat capacity, λ the heat conductivity and ϱ the density of the rail profile. We split the boundary into several parts Γ_i on which we have different boundary functions g_i, allowing us to vary the controls on different parts of the surface. By ν we denote the outer normal of the boundary.

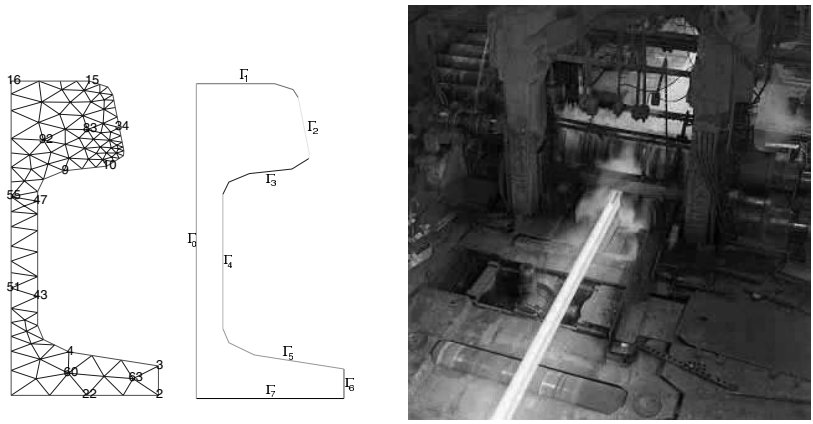


Fig. 19.1. Initial mesh, partitioning of the boundary, and a picture of a cooling plant.

We want to establish the control by a feedback law, i.e., we define the boundary functions g_i to be functions of the state x and the control u_i, where $(u_i)_i =: u = Fy$ for a linear operator F which is chosen such that the cost functional

$$\mathcal{J}(x_0,u) := \int_0^\infty (Qy,y)_Y + (Ru,u)_U dt, \qquad \text{with } y = Cx \tag{19.2}$$

is minimized. Here, Q and R are linear selfadjoint operators on the output space Y and the control space U with $Q \geq 0$, $R > 0$, and $C \in \mathcal{L}(X,Y)$.

The variational formulation of (19.1) with $g_i(t,\xi) = q_i(u_i - x(\xi,t))$ leads to:

$$(\partial_t x, v) = -\int_\Omega \alpha\nabla x\nabla v dx + \sum_k \left(q_k u_k \int_{\Gamma_k} \frac{1}{c\varrho} v \; d\sigma - \int_{\Gamma_k} \frac{q_k}{c\varrho} xv \; d\sigma \right) \tag{19.3}$$

for all $v \in \mathcal{C}_0^\infty(\Omega)$. Here the u_k are the exterior (cooling fluid) temperatures used as the controls, q_k are constant heat transfer coefficients (i.e. parameters

for the spraying intensity of the cooling nozzles) and $\alpha := \frac{\lambda}{c\varrho}$. Note that $q_0 = 0$ yields the Neumann isolation boundary condition on the artificial inner boundary on the symmetry axis.

In view of (19.3), we can now apply a standard Galerkin approach for discretizing the heat transfer model in space, resulting a first-order ordinary differential equation. This is described in the following section.

19.2 The Discretized Mathematical Model

For the discretization we use the ALBERTA-1.2 fem-toolbox (see [SS00] for details). We applied linear Lagrange elements and used a projection method for the curved boundaries. The initial mesh (see Figure 19.1. on the left) was produced by MATLABs `pdetool` which implements a Delaunay triangulation algorithm. The finer discretizations were produce by global mesh refinement using a bisection refinement method.

The discrete LQR problem is then: minimize (19.2) with respect to

$$\begin{aligned} E\dot{x}(t) &= Ax(t) + Bu(t), \text{ with } t > 0, \ x(0) = x_0, \\ y(t) &= Cx(t). \end{aligned} \tag{19.4}$$

This benchmark includes four different mesh resolutions. The best approximation error of the finite element discretization that one can expect (under suitable smoothness assumptions on the solution) is of order $O(h^2)$ where h is the maximum edge size in the corresponding mesh. This order should be matched in a model reduction approach. The following table lists some relevant quantities for the provided models.

matrix dimension	non-zeros in A	non-zeros in E	maximum mesh width (h)
1357	8985	8997	$5.5280\ 10^{-2}$
5177	35185	35241	$2.7640\ 10^{-2}$
20209	139233	139473	$1.3820\ 10^{-2}$
79841	553921	554913	$6.9100\ 10^{-3}$

Note that A is negative definite while E is positive definite, so that the resulting linear time-invariant system is stable.

The data sets are named `rail_`(problem dimension)`_C60.`(matrix name). Here C60 is a specific output matrix which is defined to minimize the temperature in the node numbered 60 (see Figure 19.1) and to keep temperature gradients small. The latter task is taken into account by the inclusion of temperature differences between specific points in the interior and reference points on the boundary, e.g. temperature difference between nodes 83 and 34. Again refer to Figure 19.1. for the nodes used. The definitions of other output matrices that we tested can be found in [Saa03].

The problem resides at temperatures of approximately 1000℃ down to about 500-700℃ depending on calculation time. The state values are scaled to 1000℃ being equivalent to 1.000. This results in a scaling of the time line with factor 100, meaning that calculated times have to be divided by 100 to get the real time in seconds.

Acknowledgments

This benchmark example serves as a model problem for the project A15: *Efficient numerical solution of optimal control problems for instationary convection-diffusion-reaction-equations* of the *Sonderforschungsbereich* SFB393 *Parallel Numerical Simulation for Physics and Continuum Mechanics*, supported by the *Deutsche Forschungsgemeinschaft*. It is motivated by the model described in [TU01] which was used to test several suboptimal control strategies in [ET01b, ET01a]. A very similar problem is used as model problem in the LYAPACK software package [Pen00].

References

[BS04] P. Benner and J. Saak. Efficient numerical solution of the LQR-problem for the heat equation. *Proc. Appl. Math. Mech.*, 4(1):648–649, 2004.

[ET01a] K. Eppler and F. Tröltzsch. Discrete and continuous optimal control strategies in the selective cooling of steel. *Z. Angew. Math. Mech.*, 81(Suppl. 2):247–248, 2001.

[ET01b] K. Eppler and F. Tröltzsch. Fast optimization methods in the selective cooling of steel. In M. Grötschel et al., editor, *Online optimization of large scale systems*, pages 185–204. Springer-Verlag, Berlin/Heidelberg, 2001.

[Pen00] T. Penzl. LYAPACK Users Guide. Technical Report SFB393/00-33, Sonderforschungsbereich 393 *Numerische Simulation auf massiv parallelen Rechnern*, TU Chemnitz, 09107 Chemnitz, FRG, 2000. Available from `http://www.tu-chemnitz.de/sfb393/sfb00pr.html`.

[Saa03] J. Saak. Effiziente numerische Lösung eines Optimalsteuerungsproblems für die Abkühlung von Stahlprofilen. Diplomarbeit, Fachbereich 3/Mathematik und Informatik, Universität Bremen, D-28334 Bremen, September 2003. Available from `http://www-user.tu-chemnitz.de/~saak/Data/index.html`.

[SS00] A. Schmidt and K. Siebert. *ALBERT: An adaptive hierarchical finite element toolbox*. Preprint 06/2000 / Institut für Angewandte Mathematik, Albert-Ludwigs-Universität Freiburg, edition: albert-1.0 edition, 2000. Available from `http://www.mathematik.uni-freiburg.de/IAM/ALBERT/doc.html`.

[TU01] F. Tröltzsch and A. Unger. Fast solution of optimal control problems in the selective cooling of steel. *Z. Angew. Math. Mech.*, 81:447–456, 2001.

20

Model Reduction of an Actively Controlled Supersonic Diffuser

Karen Willcox[1] and Guillaume Lassaux

Massachusetts Institute of Technology, Cambridge, MA, USA kwillcox@mit.edu

Summary. A model reduction test case is presented, which considers flow through an actively controlled supersonic diffuser. The problem setup and computational fluid dynamic (CFD) model are described. Sample model reduction results for two transfer functions of interest are then presented.

20.1 Supersonic Inlet Flow Example

20.1.1 Overview and Motivation

This example considers unsteady flow through a supersonic diffuser as shown in Figure 20.1. The diffuser operates at a nominal Mach number of 2.2, however it is subject to perturbations in the incoming flow, which may be due (for example) to atmospheric variations. In nominal operation, there is a strong shock downstream of the diffuser throat, as can be seen from the Mach contours plotted in Figure 20.1. Incoming disturbances can cause the shock to move forward towards the throat. When the shock sits at the throat, the inlet is unstable, since any disturbance that moves the shock slightly upstream will cause it to move forward rapidly, leading to unstart of the inlet. This is extremely undesirable, since unstart results in a large loss of thrust. In order to prevent unstart from occurring, one option is to actively control the position of the shock. This control may be effected through flow bleeding upstream of the diffuser throat. In order to derive effective active control strategies, it is imperative to have low-order models which accurately capture the relevant dynamics.

20.1.2 Active Flow Control Setup

Figure 20.2 presents the schematic of the actuation mechanism. Incoming flow with possible disturbances enters the inlet and is sensed using pressure sensors. The controller then adjusts the bleed upstream of the throat in order

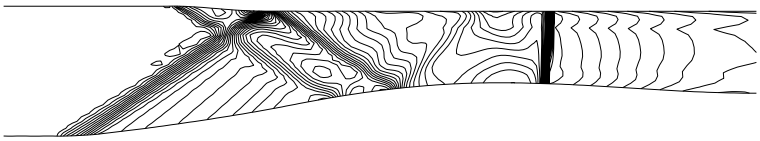


Fig. 20.1. Steady-state Mach contours inside diffuser. Freestream Mach number is 2.2.

to control the position of the shock and to prevent it from moving upstream. In simulations, it is difficult to automatically determine the shock location. The average Mach number at the diffuser throat provides an appropriate surrogate that can be easily computed.

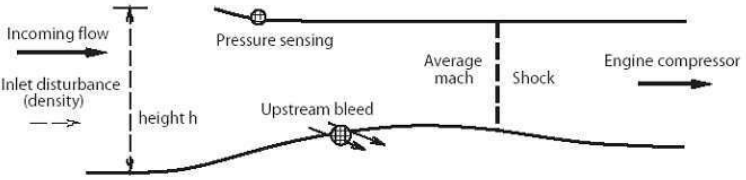


Fig. 20.2. Supersonic diffuser active flow control problem setup.

There are several transfer functions of interest in this problem. The shock position will be controlled by monitoring the average Mach number at the diffuser throat. The reduced-order model must capture the dynamics of this output in response to two inputs: the incoming flow disturbance and the bleed actuation. In addition, total pressure measurements at the diffuser wall are used for sensing. The response of this output to the two inputs must also be captured.

20.1.3 CFD Formulation

The unsteady, two-dimensional flow of an inviscid, compressible fluid is governed by the Euler equations. The usual statements of mass, momentum, and energy can be written in integral form as

$$\frac{\partial}{\partial t}\iint \rho \, dV + \oint \rho \mathbf{Q} \cdot \mathbf{dA} \;=\; 0 \tag{20.1}$$

$$\frac{\partial}{\partial t}\iint \rho \mathbf{Q} \, dV + \oint \rho \mathbf{Q} \, (\mathbf{Q} \cdot \mathbf{dA}) \;+\; \oint p \, \mathbf{dA} \;=\; 0 \tag{20.2}$$

$$\frac{\partial}{\partial t}\iint \rho E \, dV + \oint \rho H \, (\mathbf{Q} \cdot \mathbf{dA}) + \oint p \, \mathbf{Q} \cdot \mathbf{dA} = 0, \tag{20.3}$$

where ρ, $\mathbf{Q}$, H, E, and p denote density, flow velocity, total enthalpy, energy, and pressure, respectively.

The CFD formulation for this problem uses a finite volume method and is described fully in [Las02, LW03]. The unknown flow quantities used are the density, streamwise velocity component, normal velocity component, and enthalpy at each point in the computational grid. Note that the local flow velocity components q and $q^\perp$ are defined using a streamline computational grid that is computed for the steady-state solution. q is the projection of the flow velocity on the meanline direction of the grid cell, and $q^\perp$ is the normal-to-meanline component. To simplify the implementation of the integral energy equation, total enthalpy is also used in place of energy. The vector of unknowns at each node i is therefore

$$\mathbf{x}_i \;=\; \left[\rho_i\,,\, q_i\,,\, q_i^\perp\,,\, H_i\right]^T \tag{20.4}$$

Two physically different kinds of boundary conditions exist: inflow/outflow conditions, and conditions applied at a solid wall. At a solid wall, the usual no-slip condition of zero normal flow velocity is easily applied as $q^\perp = 0$. In addition, we will allow for mass addition or removal (bleed) at various positions along the wall. The bleed condition is also easily specified. We set

$$q^\perp \;=\; \frac{\dot{m}}{\rho}, \tag{20.5}$$

where $\dot{m}$ is the specified mass flux per unit length along the bleed slot. At inflow boundaries, Riemann boundary conditions are used. For the diffuser problem considered here, all inflow boundaries are supersonic, and hence we impose inlet vorticity, entropy and Riemann's invariants. At the exit of the duct, we impose outlet pressure.

20.1.4 Linearized CFD Matrices

The two-dimensional integral Euler equations are linearized about the steady-state solution to obtain an unsteady system of the form

$$E\frac{d\mathbf{x}}{dt} = A\mathbf{x} + B\mathbf{u} \qquad\qquad \mathbf{y} = C\mathbf{x} \tag{20.6}$$

The descriptor matrix E arises from the particular CFD formulation. In addition, the matrix E contains some zero rows that are due to implementation of boundary conditions.

For the results given here, the CFD model has 3078 grid points and 11,730 unknowns.

20.2 Model Reduction Results

Model reduction results are presented using the Fourier model reduction (FMR) method. A description of this method and more detailed discussion of its application to this test case can be found in [WM04].

The first transfer function of interest is that between bleed actuation and average Mach number at the throat. Bleed occurs through small slots located on the lower wall between 46% and 49% of the inlet overall length. Frequencies of practical interest lie in the range $f/f_0 = 0$ to $f/f_0 = 2$, where $f_0 = a_0/h$, a_0 is the freestream speed of sound and h is the height of the diffuser. Figure 20.3 shows the magnitude and phase of this transfer function as calculated by the CFD model and FMR reduced-order models with five and ten states. While the model with five states has some error, with just ten states the results are almost indistinguishable.

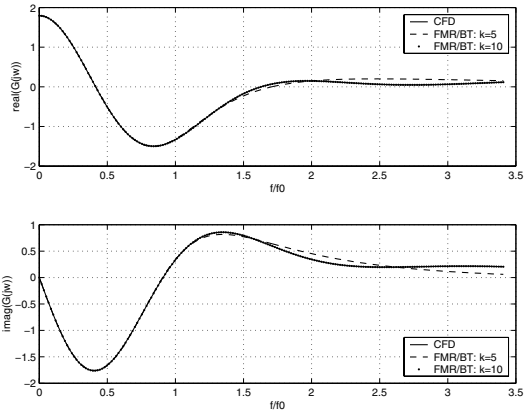


Fig. 20.3. Transfer function from bleed actuation to average throat Mach number for supersonic diffuser. Results from CFD model ($n = 11,730$) are compared to reduced-order models with five and ten states derived from an FMR model using 200 Fourier coefficients to derive a Hankel matrix that is further reduced via balanced truncation.

FMR is also applied to the transfer function between an incoming density perturbation and the average Mach number at the diffuser throat. This transfer function represents the dynamics of the disturbance to be controlled and is shown in Figure 20.4. As the figure shows, the dynamics contain a delay, and are thus more difficult for the reduced-order model to approximate. Results are shown for FMR with using 200 Fourier coefficients. The parameter ω_0 is used to define the bilinear transformation to the discrete frequency domain. Results are shown for two values of $\omega_0 = 5$ and $\omega_0 = 10$. With $\omega_0 = 5$, the model has significant error for frequencies above $f/f_0 = 2$. Choosing a higher value of ω_0 improves the fit, although some discrepancy remains. These higher frequencies are unlikely to occur in typical atmospheric disturbances, however if they are thought to be important, the model could be further improved by either evaluating more Fourier coefficients, or by choosing a higher value of

ω_0. The $\omega_0 = 10$ model is further reduced via balanced truncation to a system with thirty states without a noticeable loss in accuracy.

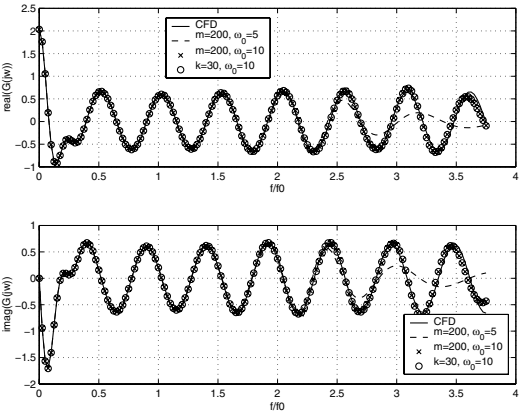


Fig. 20.4. Transfer function from incoming density perturbation to average throat Mach number for supersonic diffuser. Results from CFD model ($n = 11,730$) are compared to 200^{th}-order FMR models with $\omega_0 = 5, 10$. The $\omega_0 = 10$ model is further reduced to $k = 30$ via balanced truncation.

References

[Las02] G. Lassaux. High-Fidelity Reduced-Order Aerodynamic Models: Application to Active Control of Engine Inlets. Master's thesis, Dept. of Aeronautics and Astronautics, MIT, June 2002.

[LW03] G. Lassaux and K Willcox. Model reduction for active control design using multiple-point Arnoldi methods. AIAA Paper 2003-0616, 2003.

[WM04] K. Willcox and A. Megretski. Fourier Series for Accurate, Stable, Reduced-Order Models for Linear CFD Applications. SIAM J. Scientific Computing, **26**:3, 944–962 (2004).

21

Second Order Models: Linear-Drive Multi-Mode Resonator and Axi Symmetric Model of a Circular Piston

Zhaojun Bai[1], Karl Meerbergen[2], and Yangfeng Su[3]

[1] Department of Computer Science and Department of Mathematics, University of California, Davis, CA 95616, USA, bai@cs.ucdavis.edu
[2] Free Field Technologies, place de l'Université 16, 1348 Louvain-la-Neuve, Belgium, Karl.Meerbergen@fft.be
[3] Department of Mathematics, Fudan University, Shanghai 2200433, P. R. China, yfsu@fudan.edu.cn

21.1 Introduction

Second order systems take the form

$$M\ddot{x} + C\dot{x} + Kx = f \ . \tag{21.1}$$

Equations of this form typically arise in vibrating systems in structures and acoustics. The number of equations in (21.1) varies from a few thousands to a few million. In this section, we present two small test cases.

21.2 Linear-Drive Multi-Mode Resonator

This example is from the simulation of a linear-drive multi-mode resonator structure [CZP98]. This is a nonsymmetric second-order system. The mass and damping matrices M and D are singular. The stiffness matrix K is ill-conditioned due to the multi-scale of the physical units used to define the elements of K, such as the beam's length and cross sectional area, and its moment of inertia and modulus of elasticity.

Padé type methods usually require linear solves with K. The 1-norm condition number of K is of the order of $\mathcal{O}(10^{15})$. Therefore, we suggest the use of the expansion point $s_0 = 2 \times 10^5\pi$, which is the same as in [CZP98]. The condition number of the transformed stiffness matrix $\widetilde{K} = s_0^2 M + s_0 D + K$ is slightly improved to $\mathcal{O}(10^{13})$. The unreduced problem has dimension $N = 63$. The frequency range of interest of this problem is $[10^2, 10^6]$Hz.

21.3 Axi Symmetric Model of Circular Piston

The numerical simulation of large-size acoustic problems is a major concern in many industrial sectors. Such simulations can rely on various techniques (boundary elements, finite elements, finite differences). Exterior acoustic problems are characterized by unbounded acoustic domains. In this context, the above numerical techniques have particular features that could affect computational performances. Boundary element methods (BEM) are based on a suitable boundary integral representation and allow for a preliminary reduction of the problem to be solved (use of a surface mesh instead of a volume mesh) and for the automatic handling of the Sommerfeld radiation condition. Related matrices are however dense and non-uniqueness issues require an appropriate treatment (overdetermination procedure, combined integral form). Domain-based methods, on the other hand, do not provide direct capabilities for handling exterior acoustics. This is why finite elements (FEM) should be combined with non-reflecting boundary conditions (as the Dirichlet-to-Neumann technique) or infinite elements in order to address the problem properly. The resulting matrices are generally sparse but involve more unknowns. A more complete description and comparison of numerical techniques for exterior acoustics can be found in [Giv92] [HH92] [SB98].

This is an example from an acoustic radiation problem discussed in [PA91]. Consider a circular piston subtending a polar angle $0 < \theta < \theta_p$ on a submerged massless and rigid sphere of radius δ. The piston vibrates harmonically with a uniform radial acceleration. The surrounding acoustic domain is unbounded and is characterized by its density ρ and sound speed c.

We denote by p and a_r the prescribed pressure and normal acceleration respectively. In order to have a steady state solution $\tilde{p}(r, \theta, t)$ verifying

$$\tilde{p}(r, \theta, t) = \mathcal{R}\mathrm{e}\left(p(r, \theta)e^{i\omega t}\right),$$

the transient boundary condition is chosen as:

$$a_r = \frac{-1}{\rho}\frac{\partial p(r, \theta)}{\partial r}\bigg|_{r=a} = \begin{cases} a_0 \sin(\omega t), & 0 \le \theta \le \theta_p, \\ 0, & \theta > \theta_p. \end{cases}$$

The axisymmetric discrete finite-infinite element model relies on a mesh of linear quadrangle finite elements for the inner domain (region between spherical surfaces $r = \delta$ and $r = 1.5\delta$). The numbers of divisions along radial and circumferential directions are 5 and 80, respectively. The outer domain relies on conjugated infinite elements of order 5. For this example we used $\delta = 1(\mathrm{m})$, $\rho = 1.225(\mathrm{kg/m^3})$, $c = 340(\mathrm{m/s})$, $a_0 = 0.001(\mathrm{m/s^2})$ and $\omega = 1000(\mathrm{rad/s})$.

This example is a model of the form (21.1) with M, C, and K nonsymmetric matrices and M singular. This is thus a differential algebraic equation. It is shown that it has index one [CMR03]. The input of the system is f, the output is the state vector x. The motivation for using model reduction for this type of problems is the reduction of the computation time of a simulation.

The matrices K, C, M and the right-hand side f are computed by MSC.Actran [FFT04]. The dimension of the second-order system is $N = 2025$.

References

[CMR03] Coyette, J.-P., Meerbergen, K., Robbé, M.: Time integration for spherical acoustic finite-infinite element models (2003).

[CZP98] Clark, J. V., Zhou, N., Pister, K.S.J.: MEMS simulation using SUGAR v0.5. In *Proc. Solid-State Sensors and Actuators Workshop, Hilton Head Island, SC*, 191–196 (1998).

[FFT04] Free Field Technologies. MSC.Actran 2004. User's Manual (2004).

[Giv92] Givoli, D.: Numerical methods for problems in infinite domains. Elsevier Science Publishers (1992).

[HH92] Harari, I., Hughes, T.J.R.: A cost comparison of boundary element and finite element methods for problems of time-harmonic acoustics. Computer Methods in Applied Mechanics and Engineering, **97**:1, 103–124 (1992).

[PA91] Pinsky, P.M., Abboud, N.N.: Finite element solution of the transient exterior structural acoustics problem based on the use of radially asymptotic boundary conditions. Computer Methods in Applied Mechanics and Engineering, **85**, 311–348 (1991).

[SB98] Shirron, J.J., Babuska, I.: A comparison of approximate boundary conditions and infinite element methods for exterior Helmholtz problems. Computer Methods in Applied Mechanics and Engineering, **164**, 121–140 (1998).

22

RCL Circuit Equations

Roland W. Freund

Department of Mathematics, University of California at Davis,
One Shields Avenue, Davis, CA 95616, U.S.A.
freund@math.ucdavis.edu

Summary. RCL networks are widely used for the modeling and simulation of the interconnect of today's complex VLSI circuits. In realistic simulations, the number of these RCL networks and the number of circuit elements of each of these networks is so large that model reduction has become indispensable. We describe the general class of descriptor systems that arise in the simulation of RCL networks, and mention two particular benchmark problems.

22.1 Motivation

Today's state-of-the-art VLSI circuits contain hundreds of millions of transistors on a single chip, together with a complex network of "wires", the so-called interconnect. In fact, many aspects of VLSI circuits, such as timing behavior, signal integrity, energy consumption, and power distribution, are increasingly dominated by the chip's interconnect. For simulation of the interconnect's effects, the standard approach is to stay within the well-established lumped-circuit paradigm [VS94] and model the interconnect by simple, but large subcircuits that consist of only resistors, capacitors, and inductors; see, e.g., [CLLC00, KGP94, OCP98]. However, realistic simulations require a very large number of such RCL subcircuits, and each of these subcircuits usually consists of a very large number of circuit elements. In order to handle these large subcircuits, model-order reduction methods have become standard tools in VLSI circuit simulation. In fact, many of the Krylov subspace-based reduction techniques for large-scale linear dynamical systems were developed in the context of VLSI circuit simulation; see, e.g., [FF94, Fre00, Fre03] and the references given there.

In this brief note, we describe the general class of descriptor systems that arise in the simulation of RCL subcircuits, and mention two particular benchmark problems.

22.2 Modeling

We consider general linear RCL circuits that consist of only resistors, capacitors, inductors, voltage sources, and current sources. The voltage and current sources drive the circuit, and the voltages and currents of these sources are viewed as the inputs and outputs of the circuit. Such RCL circuits are modeled as directed graphs whose edges correspond to the circuit elements and whose nodes correspond to the interconnections of the circuit elements; see, e.g., [VS94]. For current sources, the direction of the corresponding edge is chosen as the direction of the current flow, and for voltages sources, the direction of the corresponding edge is chosen from "+" to "-" of the source. For the resistors, capacitors, and inductors, the direction of the currents through these elements is not known beforehand, and so arbitrary directions are assigned to the edges corresponding to these elements. The directed graph is described by its *incidence matrix* $A = \left[a_{jk}\right]$. The rows and columns of A correspond to the nodes and edges of the directed graph, respectively, where $a_{jk} = 1$ if edge k leaves node j, $a_{jk} = -1$ if edge k enters node j, and $a_{jk} = 0$ otherwise.

We denote by v_n the vector of nodal voltages, i.e., the j-th entry of v_n is the voltage at node j. We denote by v_e and i_e the vectors of edge voltages and currents, respectively, i.e., the k-th entry of v_e is the voltage across the circuit element corresponding to edge k, and the k-th entry of i_e is the current through the circuit element corresponding to edge k. Finally, we use subscripts r, c, l, v, and i to denote edge quantities that correspond to resistors, capacitors, inductors, voltage sources, and current sources of the RCL circuit, respectively, and we assume that the edges are ordered such that we have the following partitionings:

$$A = \left[A_r\ A_c\ A_l\ A_v\ A_i\right], \quad v_e = \begin{bmatrix} v_r \\ v_c \\ v_l \\ v_v \\ v_i \end{bmatrix}, \quad i_e = \begin{bmatrix} i_r \\ i_c \\ i_l \\ i_v \\ i_i \end{bmatrix}. \tag{22.1}$$

The RCL circuit is described completely by three types of equations: *Kirchhoff's current laws* (KCLs), *Kirchhoff's voltage laws* (KVLs), and the *branch constitutive relations* (BCRs); see, e.g., [VS94]. Using the partitionings (22.1), these equations can be written compactly as follows. The KCLs state that

$$A_r i_r + A_c i_c + A_l i_l + A_v i_v + A_i i_i = 0, \tag{22.2}$$

the KVLs state that

$$A_r^T v_n = v_r, \quad A_c^T v_n = v_c, \quad A_l^T v_n = v_l, \quad A_v^T v_n = v_v, \quad A_i^T v_n = v_i, \tag{22.3}$$

and the BCRs state that

$$i_r = R^{-1} v_r, \quad i_c = C \frac{d}{dt} v_c, \quad v_l = L \frac{d}{dt} i_l. \tag{22.4}$$

Here, R and C are positive definite diagonal matrices whose diagonal entries are the resistances and capacitances of the resistors and capacitors, respectively. The diagonal entries of the symmetric positive definite matrix L are the inductances of the inductors. Often L is also diagonal, but in general, when mutual inductances are included, L is not diagonal. In (22.2)–(22.4), the known vectors are the time-dependent functions $v_v = v_v(t)$ and $i_i = i_i(t)$ the entries of which are the voltages and currents of the voltage and current sources, respectively. All other vectors are unknown time-dependent functions.

22.3 Formulation as First-Order Descriptor Systems

The circuit equations (22.2)–(22.4) can be rewritten in a number of different ways. For example, for the special case of RCL circuits driven only by voltage sources, a formulation as systems of first-order integro-DAEs is given in Chapter 8.

Here, we present a formulation of (22.2)–(22.4) as a structured descriptor system. Recall that the currents $i_i(t)$ of the current sources, and the voltages $v_v(t)$ of the voltage sources are known functions of time. In the setting of a descriptor system, these quantities are the entries of the system's input vector $u(t)$ as follows:

$$u(t) = \begin{bmatrix} -i_i(t) \\ v_v(t) \end{bmatrix}. \tag{22.5}$$

The voltages $v_i(t)$ across the current sources, and the currents $i_v(t)$ through the voltage sources, are unknown functions of time, and these quantities are the entries of the system's output vector $y(t)$ as follows:

$$y(t) = \begin{bmatrix} v_i(t) \\ -i_v(t) \end{bmatrix}. \tag{22.6}$$

Note that we can use the first three equations in (22.3) and the BCRs (22.4) to readily eliminate the parts v_r, v_c, v_l of the edge voltages and the parts i_r, i_c of the edge currents. Therefore, in addition to the input and output variables (22.5) and (22.6), only the nodal voltages v_n and the inductor currents i_l remain as unknowns, and we define the system's state vector $x(t)$ as follows:

$$x(t) = \begin{bmatrix} v_n(t) \\ i_l(t) \\ i_v(t) \end{bmatrix}. \tag{22.7}$$

Performing the above eliminations of v_r, v_c, v_l, i_r, i_c and using (22.5)–(22.7), one easily verifies that the RCL circuit equations (22.2)–(22.4) are equivalent to the descriptor system,

$$\begin{aligned} \mathcal{E}\,\frac{d}{dt}x(t) &= \mathcal{A}\,x(t) + \mathcal{B}\,u(t), \\ y(t) &= \mathcal{B}^T x(t), \end{aligned} \tag{22.8}$$

where

$$\mathcal{A} := -\begin{bmatrix} A_r R^{-1} A_r^T & A_l & A_v \\ -A_l^T & 0 & 0 \\ -A_v^T & 0 & 0 \end{bmatrix}, \quad \mathcal{E} := \begin{bmatrix} A_c C A_c^T & 0 & 0 \\ 0 & L & 0 \\ 0 & 0 & 0 \end{bmatrix}, \tag{22.9}$$

$$\mathcal{B} := \begin{bmatrix} A_i & 0 \\ 0 & 0 \\ 0 & -I \end{bmatrix},$$

and I denotes the identity matrix. Moreover, the block sizes in (22.9) correspond to the partitionings of the input, output, and state vectors in (22.5)–(22.7).

22.4 Two Particular Benchmark Problems

The first benchmark problem, called the *PEEC problem*, is a circuit resulting from the so-called PEEC discretization [Rue74] of an electromagnetic problem. The circuit is an RCL circuit consisting of 2100 capacitors, 172 inductors, 6990 inductive couplings, and a resistive source that drives the circuit.

Table 22.1. System matrices for the PEEC problem.

matrix	n	m	nnz	Is symmetric?
$\mathcal{A}$	306	306	696	no
$\mathcal{E}$	306	306	18290	yes
$\mathcal{B}$	306	2	2	no

The second example, called the *package problem*, is a 64-pin package model used for an RF integrated circuit. Only eight of the package pins carry signals, the rest being either unused or carrying supply voltages. The package is characterized as a 16-port component (8 exterior and 8 interior terminals). The package model is described by approximately 4000 circuit elements, resistors, capacitors, inductors, and inductive couplings.

Table 22.2. System matrices for the package problem.

matrix	n	m	nnz	Is symmetric?
$\mathcal{A}$	1841	1841	5881	no
$\mathcal{E}$	1841	1841	5196	yes
$\mathcal{B}$	1841	16	24	no

22.5 Acknowledgment

The author is indebted to Peter Feldmann for first introducing him to VLSI circuit simulation, and also for providing the two benchmark problems mentioned in this paper.

References

[CLLC00] C.-K. Cheng, J. Lillis, S. Lin, and N. H. Chang. *Interconnect analysis and synthesis*. John Wiley & Sons, Inc., New York, New York, 2000.

[FF94] P. Feldmann and R. W. Freund. Efficient linear circuit analysis by Padé approximation via the Lanczos process. In *Proceedings of EURO-DAC '94 with EURO-VHDL '94*, pages 170–175, Los Alamitos, California, 1994. IEEE Computer Society Press.

[Fre00] R. W. Freund. Krylov-subspace methods for reduced-order modeling in circuit simulation. *J. Comput. Appl. Math.*, 123(1–2):395–421, 2000.

[Fre03] R. W. Freund. Model reduction methods based on Krylov subspaces. *Acta Numerica*, 12:267–319, 2003.

[KGP94] S.-Y. Kim, N. Gopal, and L. T. Pillage. Time-domain macromodels for VLSI interconnect analysis. *IEEE Trans. Computer-Aided Design*, 13:1257–1270, 1994.

[OCP98] A. Odabasioglu, M. Celik, and L. T. Pileggi. PRIMA: passive reduced-order interconnect macromodeling algorithm. *IEEE Trans. Computer-Aided Design*, 17(8):645–654, 1998.

[Rue74] A. E. Ruehli. Equivalent circuit models for three-dimensional multiconductor systems. *IEEE Trans. Microwave Theory Tech.*, 22:216–221, 1974.

[VS94] J. Vlach and K. Singhal. *Computer Methods for Circuit Analysis and Design*. Van Nostrand Reinhold, New York, New York, second edition, 1994.

23

PEEC Model of a Spiral Inductor Generated by Fasthenry

Jing-Rebecca Li[1] and Mattan Kamon[2]

[1] INRIA-Rocquencourt, Projet Ondes, Domaine de Voluceau - Rocquencourt - B.P. 105, 78153 Le Chesnay Cedex, France
jingrebecca.li@inria.fr
[2] Coventor, Inc. 625 Mt. Auburn St, Cambridge, Ma 02138, USA
matt@coventor.com

Summary. A symmetric generalized state-space model of a spiral inductor is obtained by the inductance extraction software package Fasthenry.

23.1 Fasthenry

Fasthenry[KTW94] is a software program which computes the frequency-dependent resistances and inductances of complicated three-dimensional packages and interconnect, assuming operating frequencies up to the multi-gigahertz range. Specifically, it computes the complex frequency-dependent impedance matrix $Z_p(\omega) \in \mathbb{C}^{p\times p}$ of a p-terminal set of conductors, such as an electrical package or a connector, where $Z_p(\omega)$ satisfies

$$Z_p(\omega)I_p(\omega) = V_p(\omega).$$

The quantities $I_p, V_p \in \mathbb{C}^p$ are the vectors of terminal current and voltage phasors, respectively. The frequency-dependent resistance and inductance matrices $R_p(\omega)$ and $L_p(\omega)$ are related to $Z_p(\omega)$ by:

$$Z_p(\omega) = R_p(\omega) + i\omega L_p(\omega), \tag{23.1}$$

and are important physical quantities to be preserved in reduced models.

To compute $Z_p(\omega)$, Fasthenry generates an equivalent circuit for the structure to be analyzed from the magneto-quasistatic Maxwell equations via the mesh-formulated partial element equivalent circuit (PEEC) approach using multipole acceleration. To model current flow, the interior of the conductors is divided into volume *filaments*, each of which carries a constant current density along its length. In order to capture skin and proximity effects, the cross section of each conductor is divided into bundles of filaments. In fact, many thin filaments are needed near the surface of the conductors to capture the

current crowding near the conductor surfaces at high frequencies (the skin effect). The interconnection of the filaments, plus the sources at the terminal pairs, generates a "circuit" whose solution gives the desired inductance and resistance matrices. For complicated structures, filaments numbering in the tens of thousands are not uncommon.

To derive a system of equations for the circuit of filaments, sinusoidal steady-state is assumed, and following the partial inductance approach in [Rue72], the filament current phasors can be related to the filament voltage phasors by

$$ZI_b = V_b, \tag{23.2}$$

where $V_b,\ I_b \in \mathbb{C}^b$, b is the number of filaments (number of branches in the circuit), and $Z \in \mathbb{C}^{b\times b}$ is the complex impedance matrix given by

$$Z = R + i\omega L, \tag{23.3}$$

where ω is the excitation frequency. The entries of the diagonal matrix $R \in \mathbb{R}^{b\times b}$ represent the DC resistance of each current filament, and $L \in \mathbb{R}^{b\times b}$ is the *dense* matrix of partial inductances. The partial inductance matrix is dense since every filament is magnetically coupled to every other filament in the problem.

To apply the circuit analysis technique known as Mesh Analysis, Kirchhoff's voltage law is explicitly enforced, which implies that the sum of the branch voltages around each mesh in the network is zero (a mesh is any loop of branches in the graph which does not enclose any other branches). This relation is represented by

$$MV_b = V_s \qquad M^T I_m = I_b, \tag{23.4}$$

where $V_s \in \mathbb{C}^m$ is the mostly zero vector of source branch voltages, $I_m \in \mathbb{C}^m$ is the vector of mesh currents, $M \in \mathbb{R}^{m\times b}$ is the mesh matrix. Here, m is the number of meshes, which is typically somewhat less than b, the number of filaments. The terminal source currents and voltages of the p-conductor system I_p and V_p are related to the mesh quantities by $I_p = N^T I_m, V_s = NV_p$, where $N \in \mathbb{R}^{m\times p}$ is a terminal incidence matrix determined by the mesh formulation.

Combining (23.4) and (23.2) yields

$$MZM^T I_m = V_s,$$

from which we obtain

$$I_p = N^T (MZM^T)^{-1} NV_p,$$

which gives the desired complex impedance matrix

$$Z_p(\omega) = (N^T(\tilde{R} + i\omega\tilde{L})^{-1}N)^{-1},$$

where $\tilde{L} = MLM^T \in \mathbb{R}^{m\times m}$ is the *dense* mesh inductance matrix, $\tilde{R} = MRM^T \in \mathbb{R}^{m\times m}$ is the sparse mesh resistance matrix.

Finally, we can write the mesh analysis circuit equations in the generalized state-space form:

$$E\frac{dx}{dt} = Ax + Bu, \tag{23.5}$$

$$y = B^T x, \tag{23.6}$$

where $E := \tilde{L}$ and $A := -\tilde{R}$ are both symmetric matrices, $B = N$, and u and y are the time-domain transforms of V_p and I_p, respectively. The transfer function of (23.5-23.6) evaluated on the imaginary axis gives the inverse of $Z_p(\omega)$:

$$Z_p(\omega) = (G(i\omega))^{-1}.$$

23.2 Spiral Inductor

This inductor which first appeared in [KWW00] is intended as an integrated RF passive inductor. To make it also a proximity sensor, a $0.1\mu m$ plane of copper is added $45\mu m$ above the spiral. The spiral is also copper with turns $40\mu m$ wide, $15\mu m$ thick, with a separation of $40\mu m$. The spiral is suspended $55\mu m$ over the substrate by posts at the corners and centers of the turns in order to reduce the capacitance to the substrate. (Note that neither the substrate nor the capacitance is modeled in this example.) The overall extent of the suspended turns is $1.58mm \times 1.58mm$. The spiral inductor, including part of the overhanging plane, is shown in Figure 23.1. In Figures 2(a) and 2(b), we show the resistance and impedance responses (the $R_p(\omega)$ and $L_p(\omega)$ matrices from (23.1)) of the spiral inductor corresponding to a PEEC model using 2117 filaments (state-space matrices of order 1434, single-input single-output). The frequency dependence of the resistance shows two effects, first a rise due to currents induced in the copper plane and then a much sharper rise due to the skin effect. Capturing the rise due to the skin effect while also maintaining the low frequency response is a challenge for many model reduction algorithms.

23.3 Symmetric Standard State-Space System

For certain applications one may prefer to change the generalized state-space model (23.5-23.6) to the standard state-space form. The following is a way of effecting the transformation while preserving symmetry and follows the approach used in [MSKEW96].

The mesh inductance matrix $\tilde{L}$ is symmetric and positive definite. Hence, it has a unique symmetric positive definite square root, $\tilde{L}^{\frac{1}{2}}$, satisfying $\tilde{L}^{\frac{1}{2}}\tilde{L}^{\frac{1}{2}} = \tilde{L}$. Then we use the coordinate transformation $\tilde{x} = \tilde{L}^{\frac{1}{2}}x$ to obtain the standard state-space system:

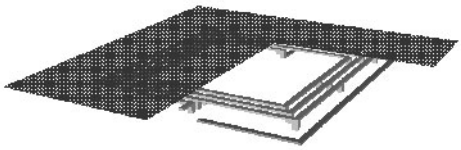
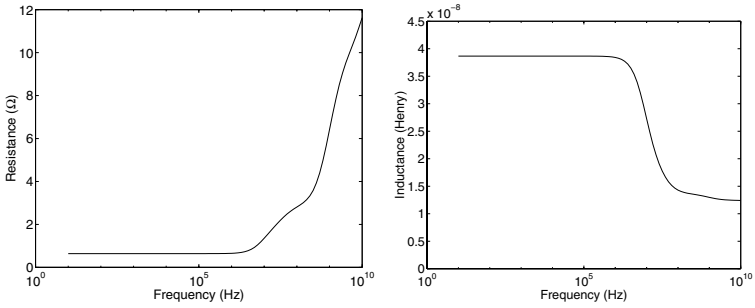


Fig. 23.1. Spiral inductor with part of overhanging copper plane



(a) Resistance (b) Inductance

Fig. 23.2. PEEC model of spiral inductor using 2117 filaments

$$\frac{d\tilde{x}}{dt} = \tilde{A}\tilde{x} + \tilde{B}u, \tag{23.7}$$

$$y = \tilde{B}^T\tilde{x}, \tag{23.8}$$

where $\tilde{A} = -\tilde{L}^{-\frac{1}{2}}\tilde{R}\tilde{L}^{-\frac{1}{2}}$ is symmetric ($\tilde{L}^{-\frac{1}{2}}$ is symmetric) and $\tilde{B} = B\tilde{L}^{-\frac{1}{2}}$.

If the original matrices are too large for testing purposes when comparing with methods requiring $O(n^3)$ work, applying the Prima[OCP98] algorithm to the generalized state-space model in (23.5-23.6) with a reduction order of around 100 will produce a smaller system with virtually the same frequency response. Indeed this is what has been done when this example was used previously in numerous papers, including [LWW99].

Note

We make a note here that the example first used in [LWW99] and subsequently in other papers comes from a finer discretization of the spiral inductor than shown here. That example started with state-spaces matrices of order 1602 (compared to order 1434 here). The order 500 system was obtained by running

Prima with a reduction order of 503. Due to the loss of orthogonality of the Arnoldi vectors the reduced matrix E_r has three zero eigenvalues. The modes corresponding to the zero eigenvalues were simply removed to give a new positive definite E_r matrix and a system of order 500. The frequency response of the resulting system is indistinguishable from the original.

Acknowledgments

Most of the description of the spiral inductor and how the model was generated by Fasthenry is paraphrased from [KTW94] and [KWW00].

References

[KTW94] Kamon, M., Tsuk, M.J., White, J.:. Fasthenry: A multipole-accelerated 3-d inductance extraction program. IEEE Trans. Microwave Theory and Techniques, **42**:9, 1750–1758 (1994).

[KWW00] Kamon, M., Wang, F., White, J.: Generating nearly optimally compact models from Krylov-subspace based reduced order models. IEEE Trans. Circuits and Systems-II:Analog and Digital Signal Processing, **47**:4, 239–248 (2000).

[LWW99] Li, J.-R., Wang, F., White, J.: An efficient Lyapunov equation-based approach for generating reduced-order models of interconnect. In *Proceedings of the 36th Design Automation Conference*, 1–6 (1999).

[MSKEW96] Miguel Silveira, L., Kamon, M., Elfadel, I., White, J.: A coordinate-transformed Arnoldi algorithm for generating guaranteed stable reduced-order models of RLC circuits. In *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, 288–294 (1996).

[OCP98] Odabasioglu, A., Celik, M., Pileggi, L.T.: PRIMA: Passive Reduced-order Interconnect Macromodeling Algorithm. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, **17**:8, 645–654 (1998).

[Rue72] Ruehli, A.E.: Inductance calculations in a complex integrated circuit environment. IBM J. Res. Develop., **16**, 470–481 (1972).

24

Benchmark Examples for Model Reduction of Linear Time-Invariant Dynamical Systems

Younes Chahlaoui[1] and Paul Van Dooren[2]

[1] School of Computational Science, Florida State University, Tallahassee, U.S.A.
younes.chahlaoui@laposte.net
[2] CESAME, Université catholique de Louvain, Louvain-la-Neuve, Belgium
vdooren@csam.ucl.ac.be

Summary. We present a benchmark collection containing some useful real world examples, which can be used to test and compare numerical methods for model reduction. All systems can be downloaded from the web and we describe here the relevant characteristics of the benchmark examples.

24.1 Introduction

In this paper we describe a number of *benchmark examples* for model reduction of linear time-invariant systems of the type

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{cases} \tag{24.1}$$

with an associated transfer function matrix

$$G(s) = C(sI_N - A)^{-1}B + D. \tag{24.2}$$

The matrices of these models are all real and have the following dimensions : $A \in \mathbb{R}^{N\times N}$, $B \in \mathbb{R}^{N\times m}$, $C \in \mathbb{R}^{p\times N}$, and $D \in \mathbb{R}^{p\times m}$. The systems are all stable and minimal and the number of state variables N is thus the order of the system. In model reduction one tries to find a reduced order model,

$$\begin{cases} \dot{\hat{x}}(t) = \hat{A}\hat{x}(t) + \hat{B}\hat{u}(t) \\ \hat{y}(t) = \hat{C}\hat{x}(t) + \hat{D}\hat{u}(t) \end{cases} \tag{24.3}$$

of order $n \ll N$, such that the transfer function matrix $\hat{G}(s) = \hat{C}(sI_n - \hat{A})^{-1}\hat{B} + \hat{D}$ approximates $G(s)$ in a particular sense, and model reduction methods differ typically in the error measure that is being minimized. In assessing the quality of the reduced order model, one often looks at the following characteristics of the system to be approximated

- the *eigenvalues* of A (or at least the closest ones to the $j\omega$ axis), which are also the poles of $G(s)$
- the *controllability Gramian* $\mathcal{G}_c$ and *observability Gramian* $\mathcal{G}_o$ of the system, which are the solutions of the Lyapunov equations

$$A\mathcal{G}_c + \mathcal{G}_c A^T + BB^T = 0, \quad A^T\mathcal{G}_o + \mathcal{G}_o A + C^T C = 0$$

- the singular values of the Hankel map – called the *Hankel singular values* (HSV) – which are also the square-roots of the eigenvalues of $\mathcal{G}_c\mathcal{G}_o$
- the largest singular value of the transfer function as function of frequency – called the *frequency response* –

$$\sigma(\omega) = \|G(j\omega)\|_2.$$

These characteristics can be compared with those of the reduced order model $\hat{G}(s)$. Whenever they are available, we give all of the above properties for the benchmark examples we discuss in this paper. The data files for the examples can be recovered from *http://www.win.tue.nl/niconet/niconet.html*. For each example we provide the matrix model $\{A, B, C, D\}$, and (when available) the poles, the Gramians, the Hankel singular values, a frequency vector and the corresponding frequency response. For more examples and additional details of the examples of this paper, we refer to [CV02]. Some basic parameters of the benchmarks discussed in the paper are given below.

Section	Example (Acronym)	Sparsity	N	m	p
2	Earth Atmosphere (ATMOS)	no	598	1	1
3	Orr-Sommerfeld (ORR-S)	no	100	1	1
4	Compact Disc player (C-DISC)	yes	120	2	2
5	Random (RAND)	yes	200	1	1
6	Building (BUILD-I)	yes	48	1	1
6	Building (BUILD-II)	yes	52788	1	1
6	Clamped Beam (BEAM)	yes	348	1	1
7	Intern. Space Station (ISS-I)	yes	270	3	3
7	Intern. Space Station (ISS-II)	yes	1412	3	3

24.2 Earth Atmospheric Example (ATMOS)

This is a model of an atmospheric storm track [FI95]. In order to simulate the lack of coherence of the cyclone waves around the *Earth's atmosphere*, linear damping at the storm track's entry and exit region is introduced. The perturbation variable is the perturbation geopotential height. The perturbation equations for single harmonic perturbations in the meridional (y) direction of the form $\phi(x, z, t)e^{ily}$ are :

$$\frac{\partial \phi}{\partial t} = \nabla^{-2}\Big[- z\nabla^2 D\phi - r(x)\nabla^2\phi\Big],$$

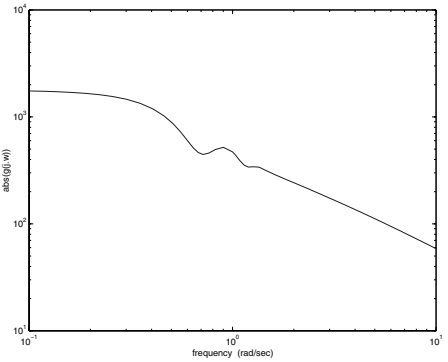


Fig. 24.1. Frequency response (ATMOS)

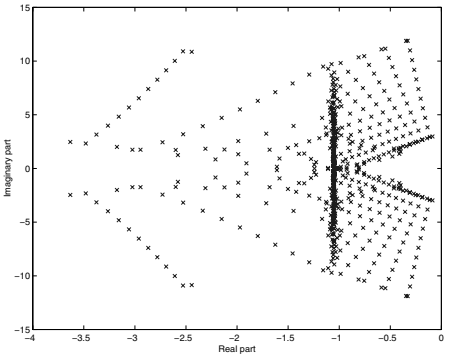


Fig. 24.2. Eigenvalues of A (ATMOS)

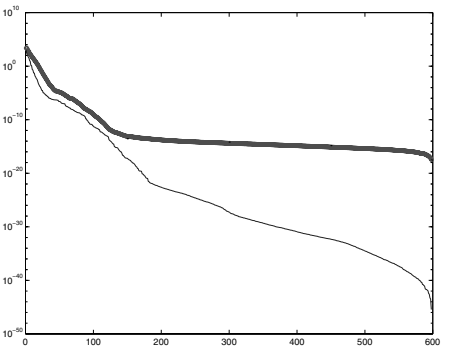


Fig. 24.3. $\cdots$ svd($\mathcal{G}_c$), o svd($\mathcal{G}_o$), $-$ hsv

where ∇^2 is the Laplacian $\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial z^2} - l^2$ and $D = \frac{\partial}{\partial x}$. The linear damping rate $r(x)$ is taken to be $r(x) = h(2 - \tanh[(x - \frac{\pi}{4})/\delta] + \tanh[(x - \frac{7\pi}{2})/\delta])$. The boundary conditions are expressing the conservation of potential temperature (entropy) along the solid surfaces at the ground and tropopause:

$$\frac{\partial^2 \phi}{\partial t \partial z} = -zD\frac{\partial \phi}{\partial z} + D\phi - r(x)\frac{\partial \phi}{\partial z} \quad \text{at} \quad z = 0,$$

$$\frac{\partial^2 \phi}{\partial t \partial z} = -zD\frac{\partial \phi}{\partial z} + D\phi - r(x)\frac{\partial \phi}{\partial z} \quad \text{at} \quad z = 1.$$

The dynamical system is written in generalized velocity variables $\psi = (-\nabla^2)^{\frac{1}{2}}\phi$ so that the dynamical system is governed by the dynamical operator:

$$A = (-\nabla^2)^{\frac{1}{2}}\nabla^{-2}\Big(- zD\nabla^2 + r(x)\nabla^2\Big)(-\nabla^2)^{-\frac{1}{2}}.$$

where the boundary equations have rendered the operators invertible. We refer to [FI95] for more details, including the type of discretization that was used.

24.3 Orr-Sommerfeld Equation (ORR-S)

The *Orr-Sommerfeld* operator for the Couette flow in perturbation velocity variables is given by :

$$A = (-D^2)^{\frac{1}{2}} D^{-2} \Big(- ijkD^2 + \frac{1}{Re} D^4 \Big) (-D^2)^{-\frac{1}{2}}$$

where $D := \frac{d}{dy}$ and appropriate boundary conditions have been introduced so that the inverse operator is defined. Here, Re is the Reynolds number and k is the wave-number of the perturbation. This operator governs the evolution of 2-dimensional perturbations. The considered matrix is a 100×100 discretization for a Reynolds number $Re = 800$ and for $k = 1$. We refer to [FI01] for more details, including the type of discretization that was used.

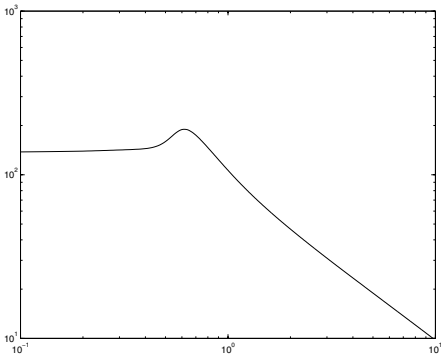


Fig. 24.4. Frequency response (ORR-S)

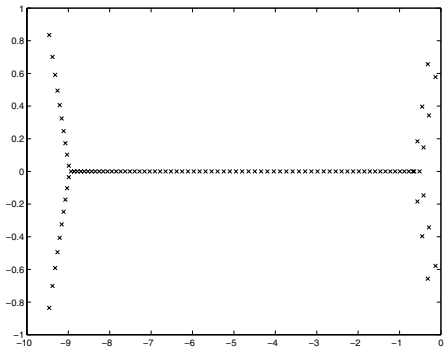


Fig. 24.5. Eigenvalues of A (ORR-S)

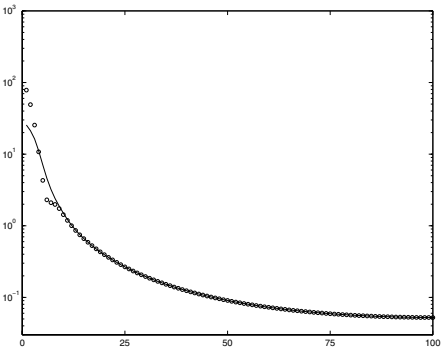


Fig. 24.6. $\cdots$ svd($\mathcal{G}_c$), o svd($\mathcal{G}_o$), $-$ hsv

24.4 Compact Disc Player Example (C-DISC)

The *CD player* control task is to achieve track following, which amounts to pointing the laser spot to the track of pits on a CD that is rotating. The mechanism that is modeled consists of a swing arm on which a lens is mounted by means of two horizontal leaf springs. The rotation of the arm in the horizontal plane enables reading of the spiral-shaped disc-tracks, and the suspended lens is used to focus the spot on the disc. Since the disc is not perfectly flat and since there are irregularities in the spiral of pits on the disc, the challenge is to find a low-cost controller that can make the servo-system faster and less sensitive to external shocks. We refer to [DSB92, WSB96] for more details.

It is worth mentioning here that this system is already a reduced order model obtained via modal approximation from a larger rigid body model (which is a second order model).

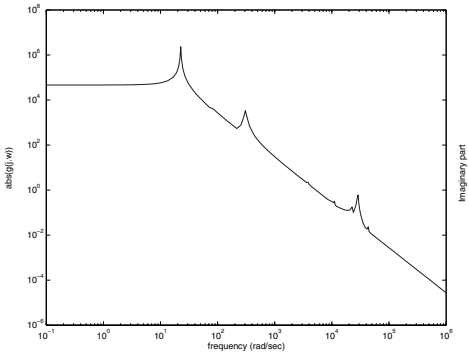


Fig. 24.7. Frequency response (C-DISC)

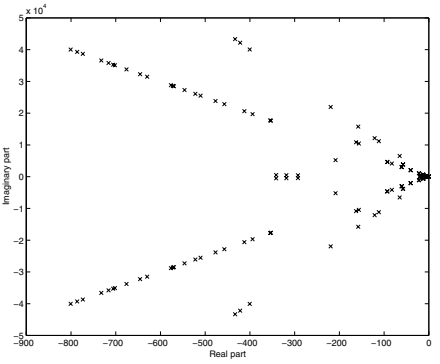


Fig. 24.8. Eigenvalues of A (C-DISC)

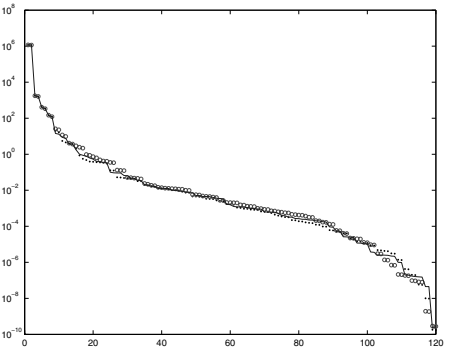


Fig. 24.9. $\cdots$ svd($\mathcal{G}_c$), o svd($\mathcal{G}_o$), $-$ hsv

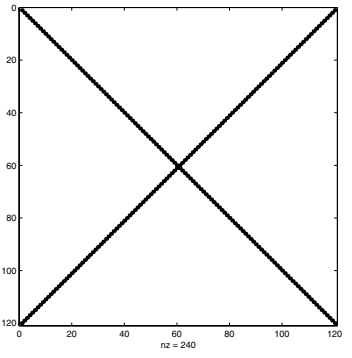


Fig. 24.10. Sparsity of A (C-DISC)

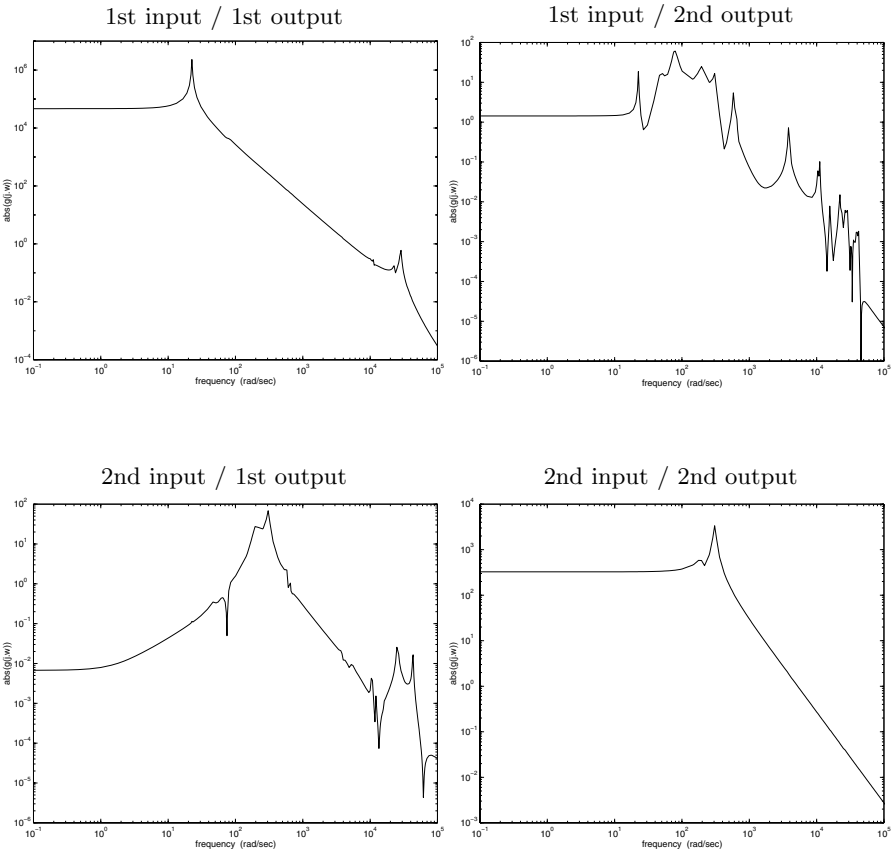


Fig. 24.11. Frequency responses of the 2-input 2-output system (C-DISC)

24.5 Random Example (RAND)

This is a randomly generated example with an A matrix that is sparse and stable, and has a prescribed percentage of nonzero elements. This is a simple example to approximate but it is useful to compare convergence rates of iterative algorithms. It is extracted from the Engineering thesis of V. Declippel [DeC97].

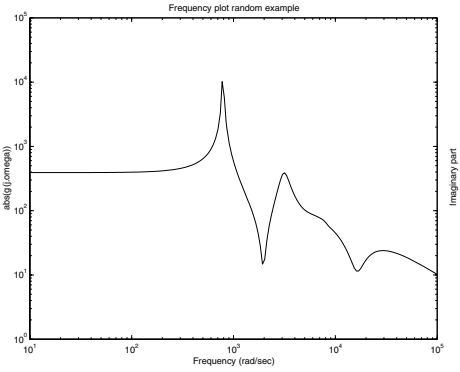


Fig. 24.12. Frequency response (RAND)

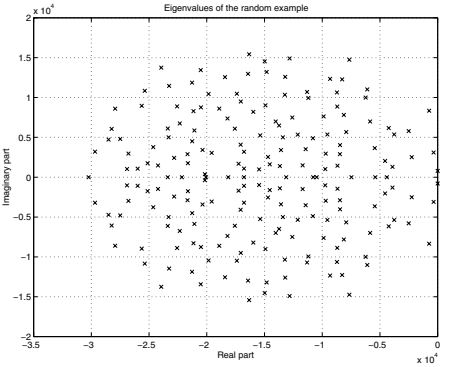


Fig. 24.13. Eigenvalues of A (RAND)

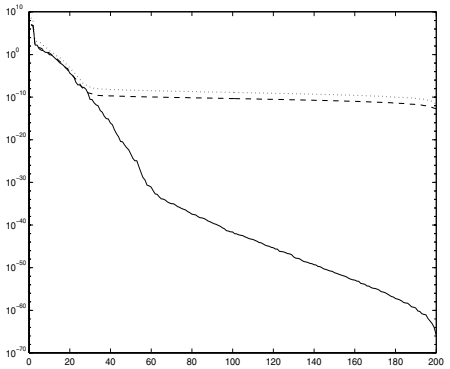


Fig. 24.14. $\cdots$ svd($\mathcal{G}_c$), o svd($\mathcal{G}_o$), $-$ hsv

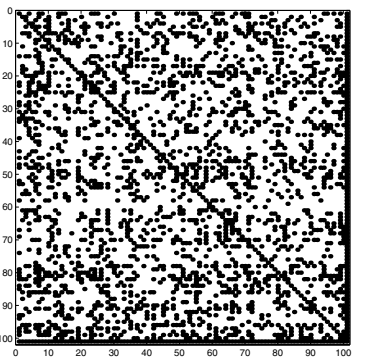


Fig. 24.15. Sparsity of A (RAND)

24.6 Building Model

Mechanical systems are typically modeled as second order differential equations

$$\begin{cases} M\ddot{q}(t) + D\dot{q}(t) + Sq(t) = B_q u(t), \\ y(t) = C_q q(t) \end{cases}$$

where $u(t)$ is the input or forcing function, $q(t)$ is the position vector, and where the output vector $y(t)$ is typically a function of the position vector. Here M is the (positive definite) mass matrix, D is the damping matrix and S is the stiffness matrix of the mechanical system. Since M is invertible, one can use the extended state

$$x(t)^T = \left[q(t)^T \ \dot{q}(t)^T \right]$$

to derive a linearized state space realization

$$A := \begin{bmatrix} 0 & I \\ -M^{-1}S & -M^{-1}D \end{bmatrix}, \quad B := \begin{bmatrix} 0 \\ M^{-1}B_q \end{bmatrix}, \quad C := \begin{bmatrix} C_q & 0 \end{bmatrix}$$

or a weighted extended state

$$x(t)^T = \left[q(t)^T M^{-\frac{1}{2}} \ \dot{q}(t)^T M^{-\frac{1}{2}} \right]$$

yielding a more "symmetric" model

$$A := \begin{bmatrix} 0 & I \\ -\hat{S} & -\hat{D} \end{bmatrix}, \quad B := \begin{bmatrix} 0 \\ \hat{B}_q \end{bmatrix}, \quad C := \begin{bmatrix} \hat{C}_q & 0 \end{bmatrix}$$

and where $\hat{D} = M^{-\frac{1}{2}} D M^{-\frac{1}{2}}$, $\hat{S} = M^{-\frac{1}{2}} S M^{-\frac{1}{2}}$, $\hat{B} = M^{-\frac{1}{2}} B$ and $\hat{C} = C M^{-\frac{1}{2}}$. When M is the identity matrix, one can recover the original matrices from the linearized model. If this is not the case, those matrices are also provided in the benchmark data.

24.6.1 Simple Building Model (BUILD-I)

This is a small model of state dimension $N = 48$. It is borrowed from [ASG01].

24.6.2 Earth Quake Model (BUILD-II)

This is a model of a building for which the effect of earthquakes is to be analyzed (it is provided by Professor Mete Sozen of Purdue University). The mass matrix M is diagonal and of dimension $N = 26394$. The stiffness matrix S is symmetric and has the sparsity pattern given in Figure 24.19.

The damping matrix is chosen to be $D = \alpha M + \beta S$, with $\alpha = 0.675$ and $\beta = 0.00315$. The matrix B_q is a column vector of all ones and $C_q = B_q^T$. No exact information is available on the frequency response and on the Gramians of this large scale system.

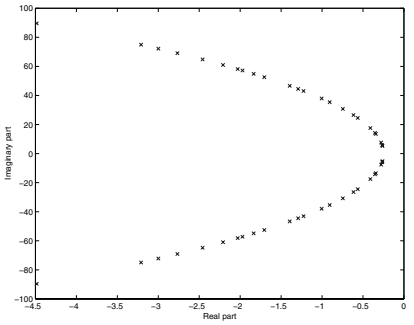


Fig. 24.16. Eigenvalues of A (BUILD-I)

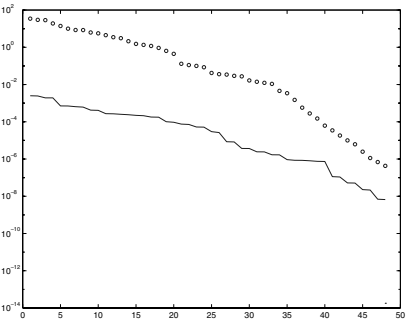


Fig. 24.17. $\cdots$ svd($\mathcal{G}_c$), o svd($\mathcal{G}_o$), $-$ hsv

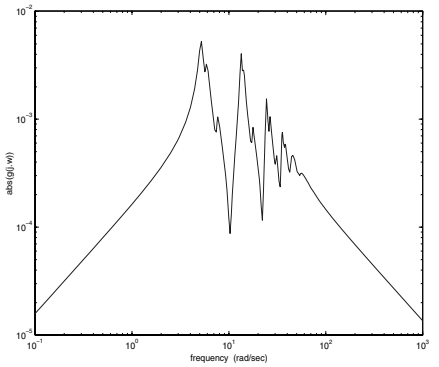


Fig. 24.18. Freq. response (BUILD-I)

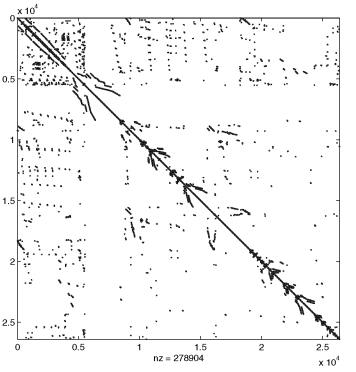


Fig. 24.19. Sparsity of S (BUILD-II)

24.6.3 Clamped Beam Model (BEAM)

The *clamped beam* model has 348 states, it is obtained by spatial discretization of an appropriate partial differential equation. The input represents the force applied to the structure at the free end, and the output is the resulting displacement. The data were obtained from [ASG01].

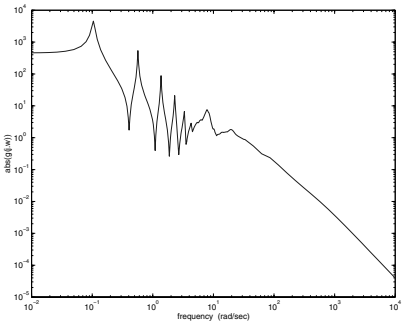


Fig. 24.20. Frequency response (BEAM)

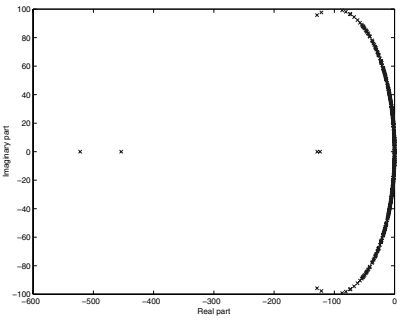


Fig. 24.21. Eigenvalues of A (BEAM)

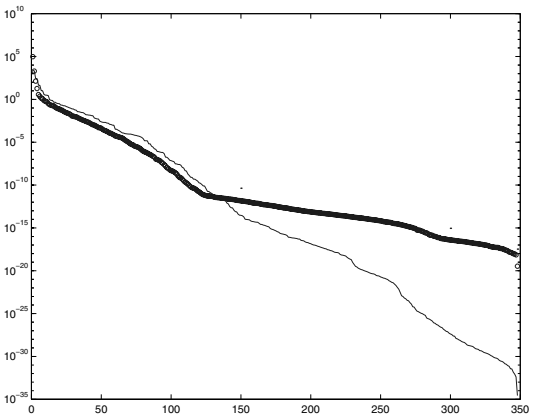


Fig. 24.22. $\cdots$ svd($\mathcal{G}_c$), o svd($\mathcal{G}_o$), $-$ hsv

24.7 International Space Station

This is a structural model of the *International Space Station* being assembled in various stages. The aim is to model vibrations caused by a docking of an incoming spaceship. The required control action is to dampen the effect of these vibrations as much as possible. The system is lightly damped and control actions will be constrained. Two models are given, which relate to different stages of completion of the Space Station [SAB01]. The sparsity pattern of A shows that it is in fact derived from a mechanical system model.

24.7.1 Russian Service Module (ISS-I)

This consists of a first assembly stage (the so-called Russian service module 1R [SAB01]) of the International Space Station. The state dimension is $N = 270$.

24.7.2 Extended Service Module (ISS-II)

This consists of a second assembly stage (the so-called 12A model [SAB01]) of the International Space Station. The state dimension is $N = 1412$.

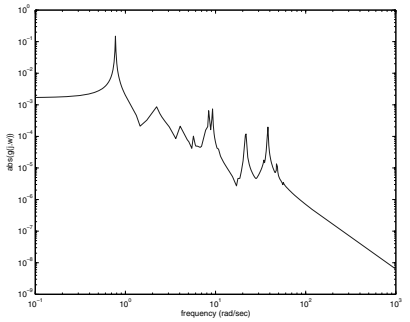


Fig. 24.23. Frequency response (ISS-I)

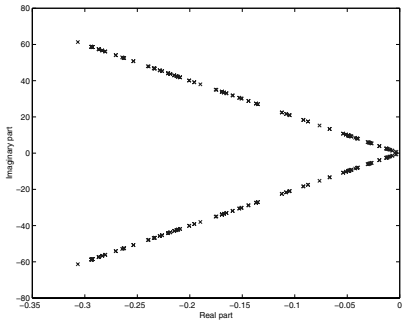


Fig. 24.24. Eigenvalues of A (ISS-I)

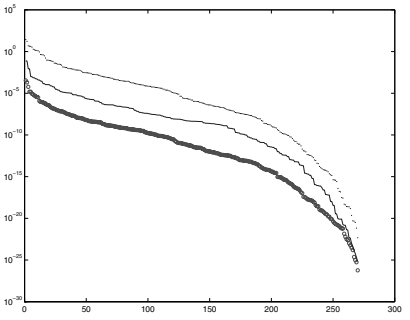


Fig. 24.25. $\cdots$ svd($\mathcal{G}_c$), o svd($\mathcal{G}_o$), $-$ hsv

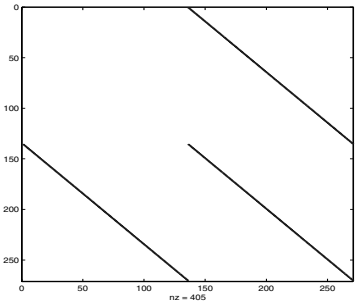


Fig. 24.26. Sparsity of A (ISS-I)

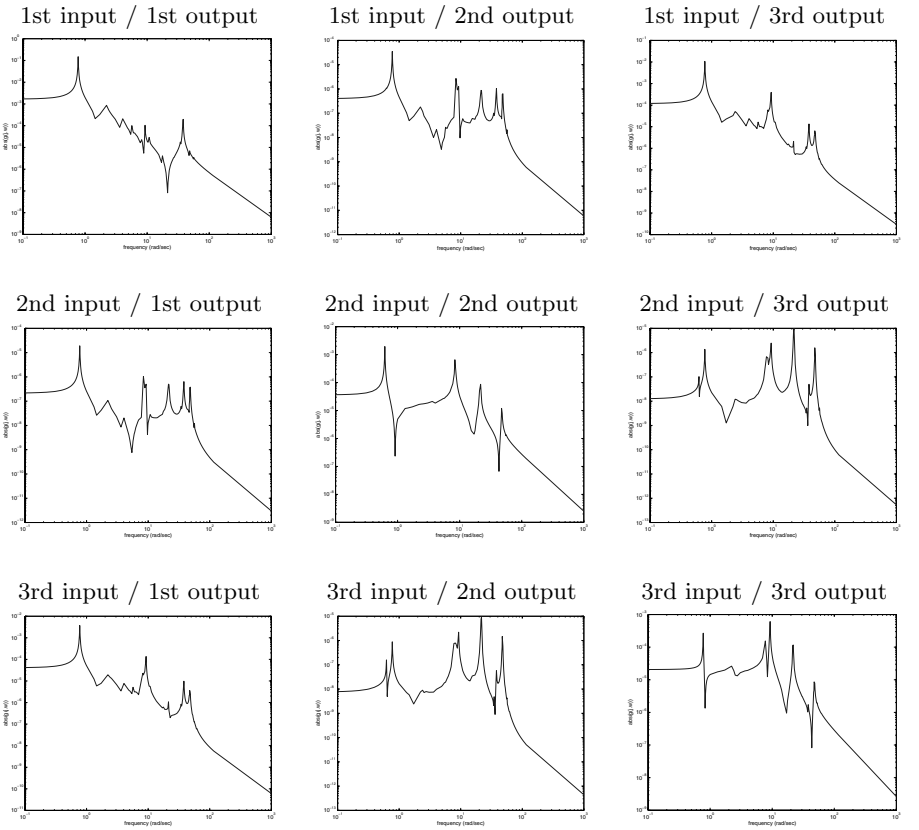


Fig. 24.27. Frequency response of the 3-input 3-output system (ISS-II)

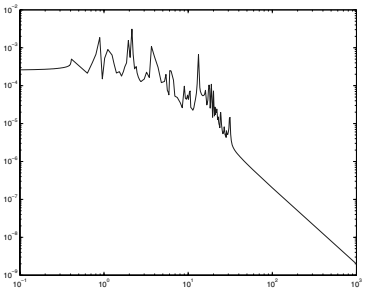


Fig. 24.28. Frequency response (ISS-II)

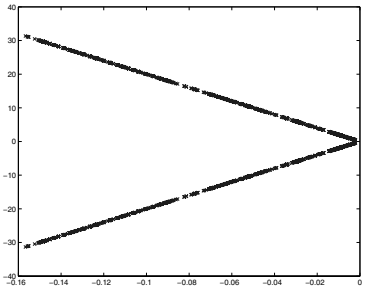


Fig. 24.29. Eigenvalues of A (ISS-II)

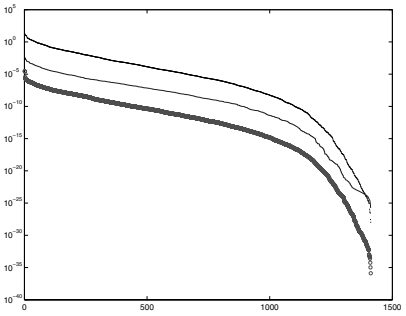


Fig. 24.30. $\cdots$ svd($\mathcal{G}_c$), o svd($\mathcal{G}_o$), $-$ hsv (ISS-II)

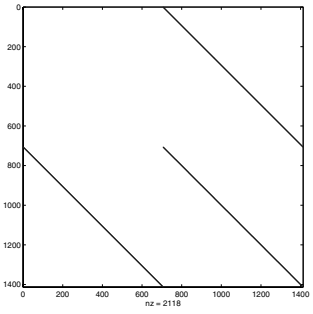


Fig. 24.31. Sparsity of A (ISS-II)

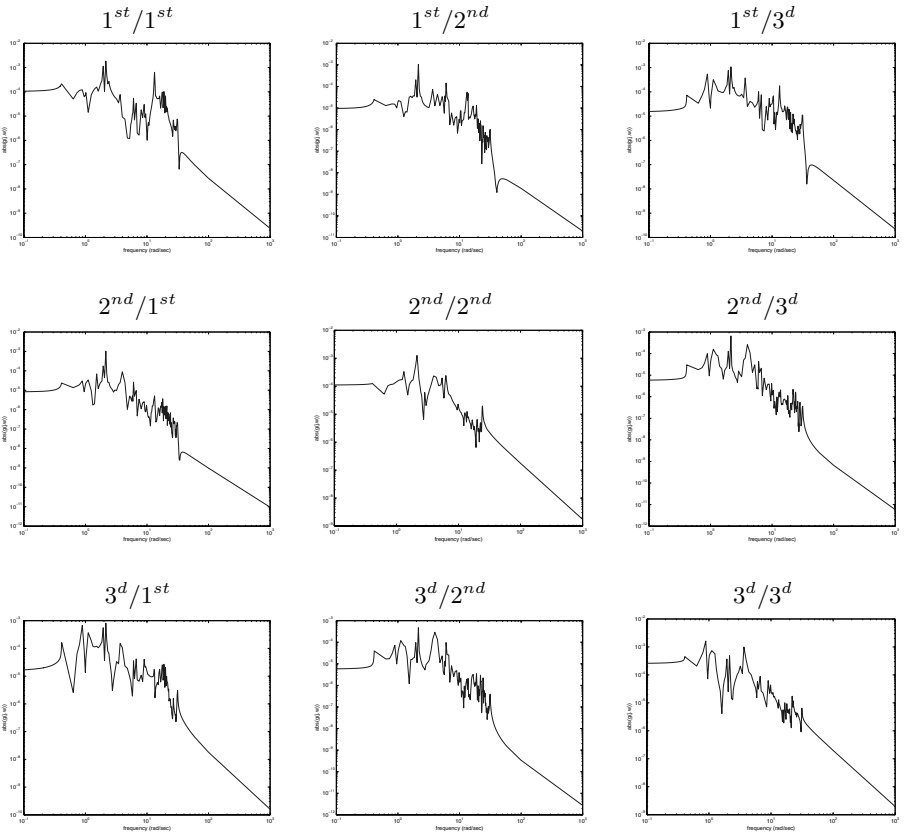


Fig. 24.32. Frequency response of the ISS12A model (i^{th} input/j^{th} output).

Acknowledgment

We would like to thank all contributors who sent us their examples for inclusion in this report : A. Antoulas, V. De Clippel, B. Farrell, P. Ioannou, M. Sozen and P. Wortelboer. This paper presents research supported by NSF contracts CCR-99-12415 and ITR ACI-03-24944 and by the Belgian Programme on Inter-university Poles of Attraction, initiated by the Belgian State, Prime Minister's Office for Science, Technology and Culture. The scientific responsibility rests with its authors.

References

[ASG01] Antoulas, A., Sorenson, D. and Gugercin, S.: A Survey of Model Reduction Methods for Large-Scale Systems. *Contemporary Mathematics*, **280**, 193–219 (2001)

[CV02] Chahlaoui, Y. and Van Dooren, P.: A collection of benchmark examples for model reduction of linear time invariant dynamical systems. *SLICOT Working Note*, `ftp://wgs.esat.kuleuven.ac.be/pub/WGS/REPORTS/SLWN2002-2.ps.Z`.

[DeC97] De Clippel, V.: *Modèles réduits de grands systèmes dynamiques*. Engineering Thesis, Université catholique de Louvain, Louvain-la-Neuve (1997)

[DSB92] Draijer, W., Steinbuch, M. and Bosgra, O.: Adaptive Control of the Radial Servo System of a Compact Disc Player. *Automatica*, **28(3)**, 455–462 (1992)

[FI95] Farrell, B.F. and Ioannou, P.J.: Stochastic dynamics of the mid-latitude atmospheric jet. *Journal of the Atmospheric Sciences*, **52(10)**, 1642–1656 (1995)

[FI01] Farrell, B.F. and Ioannou, P.J.: Accurate Low Dimensional Approximation of the Linear Dynamics of Fluid Flow. *Journal of the Atmospheric Sciences*, **58(18)**, 2771–2789 (2001)

[SAB01] Gugercin, S., Antoulas, A. and Bedrossian, N.: Approximation of the International Space Station 1R and 12A flex models. In: *Proc. of the IEEE Conference on Decision and Control*, Orlando, Paper WeA08 (2001)

[WSB96] Wortelboer, P., Steinbuch, M. and Bosgra, O.: Closed-Loop Balanced Reduction with Application to a Compact Disc Mechanism. *Selected Topics in Identification, Modeling and Control*, **9**, 47–58, (1996)

Index

Editorial Policy

§1. Volumes in the following three categories will be published in LNCSE:

i) Research monographs
ii) Lecture and seminar notes
iii) Conference proceedings

Those considering a book which might be suitable for the series are strongly advised to contact the publisher or the series editors at an early stage.

§2. Categories i) and ii). These categories will be emphasized by Lecture Notes in Computational Science and Engineering. **Submissions by interdisciplinary teams of authors are encouraged.** The goal is to report new developments – quickly, informally, and in a way that will make them accessible to non-specialists. In the evaluation of submissions timeliness of the work is an important criterion. Texts should be well-rounded, well-written and reasonably self-contained. In most cases the work will contain results of others as well as those of the author(s). In each case the author(s) should provide sufficient motivation, examples, and applications. In this respect, Ph.D. theses will usually be deemed unsuitable for the Lecture Notes series. Proposals for volumes in these categories should be submitted either to one of the series editors or to Springer-Verlag, Heidelberg, and will be refereed. A provisional judgment on the acceptability of a project can be based on partial information about the work: a detailed outline describing the contents of each chapter, the estimated length, a bibliography, and one or two sample chapters – or a first draft. A final decision whether to accept will rest on an evaluation of the completed work which should include

– at least 100 pages of text;
– a table of contents;
– an informative introduction perhaps with some historical remarks which should be accessible to readers unfamiliar with the topic treated;
– a subject index.

§3. Category iii). Conference proceedings will be considered for publication provided that they are both of exceptional interest and devoted to a single topic. One (or more) expert participants will act as the scientific editor(s) of the volume. They select the papers which are suitable for inclusion and have them individually refereed as for a journal. Papers not closely related to the central topic are to be excluded. Organizers should contact Lecture Notes in Computational Science and Engineering at the planning stage.

In exceptional cases some other multi-author-volumes may be considered in this category.

§4. Format. Only works in English are considered. They should be submitted in camera-ready form according to Springer-Verlag's specifications.
Electronic material can be included if appropriate. Please contact the publisher.
Technical instructions and/or TEX macros are available via
http://www.springeronline.com/sgw/cda/frontpage/0,10735,5-111-2-71391-0,00.html
The macros can also be sent on request.

General Remarks

Lecture Notes are printed by photo-offset from the master-copy delivered in camera-ready form by the authors. For this purpose Springer-Verlag provides technical instructions for the preparation of manuscripts. See also *Editorial Policy*.

Careful preparation of manuscripts will help keep production time short and ensure a satisfactory appearance of the finished book.

The following terms and conditions hold:

Categories i), ii), and iii):
Authors receive 50 free copies of their book. No royalty is paid. Commitment to publish is made by letter of intent rather than by signing a formal contract. Springer-Verlag secures the copyright for each volume.

For conference proceedings, editors receive a total of 50 free copies of their volume for distribution to the contributing authors.

All categories:
Authors are entitled to purchase further copies of their book and other Springer mathematics books for their personal use, at a discount of 33,3 % directly from Springer-Verlag.

Addresses:

Timothy J. Barth
NASA Ames Research Center
NAS Division
Moffett Field, CA 94035, USA
e-mail: barth@nas.nasa.gov

Michael Griebel
Institut für Angewandte Mathematik
der Universität Bonn
Wegelerstr. 6
53115 Bonn, Germany
e-mail: griebel@ins.uni-bonn.de

David E. Keyes
Department of Applied Physics
and Applied Mathematics
Columbia University
200 S. W. Mudd Building
500 W. 120th Street
New York, NY 10027, USA
e-mail: david.keyes@columbia.edu

Risto M. Nieminen
Laboratory of Physics
Helsinki University of Technology
02150 Espoo, Finland
e-mail: rni@fyslab.hut.fi

Dirk Roose
Department of Computer Science
Katholieke Universiteit Leuven
Celestijnenlaan 200A
3001 Leuven-Heverlee, Belgium
e-mail: dirk.roose@cs.kuleuven.ac.be

Tamar Schlick
Department of Chemistry
Courant Institute of Mathematical
Sciences
New York University
and Howard Hughes Medical Institute
251 Mercer Street
New York, NY 10012, USA
e-mail: schlick@nyu.edu

Springer-Verlag, Mathematics Editorial IV
Tiergartenstrasse 17
Ð 69121 Heidelberg, Germany
Tel.: *49 (6221) 487-8185
Fax: *49 (6221) 487-8355
e-mail: Martin.Peters@springer-sbm.com

Lecture Notes in Computational Science and Engineering

Vol. 1 D. Funaro, *Spectral Elements for Transport-Dominated Equations.* 1997. X, 211 pp. Softcover. ISBN 3-540-62649-2

Vol. 2 H. P. Langtangen, *Computational Partial Differential Equations.* Numerical Methods and Diffpack Programming. 1999. XXIII, 682 pp. Hardcover. ISBN 3-540-65274-4

Vol. 3 W. Hackbusch, G. Wittum (eds.), *Multigrid Methods V.* Proceedings of the Fifth European Multigrid Conference held in Stuttgart, Germany, October 1-4, 1996. 1998. VIII, 334 pp. Softcover. ISBN 3-540-63133-X

Vol. 4 P. Deuflhard, J. Hermans, B. Leimkuhler, A. E. Mark, S. Reich, R. D. Skeel (eds.), *Computational Molecular Dynamics: Challenges, Methods, Ideas.* Proceedings of the 2nd International Symposium on Algorithms for Macromolecular Modelling, Berlin, May 21-24, 1997. 1998. XI, 489 pp. Softcover. ISBN 3-540-63242-5

Vol. 5 D. Kröner, M. Ohlberger, C. Rohde (eds.), *An Introduction to Recent Developments in Theory and Numerics for Conservation Laws.* Proceedings of the International School on Theory and Numerics for Conservation Laws, Freiburg / Littenweiler, October 20-24, 1997. 1998. VII, 285 pp. Softcover. ISBN 3-540-65081-4

Vol. 6 S. Turek, *Efficient Solvers for Incompressible Flow Problems.* An Algorithmic and Computational Approach. 1999. XVII, 352 pp, with CD-ROM. Hardcover. ISBN 3-540-65433-X

Vol. 7 R. von Schwerin, ***M**ulti **B**ody **S**ystem **SIM**ulation.* Numerical Methods, Algorithms, and Software. 1999. XX, 338 pp. Softcover. ISBN 3-540-65662-6

Vol. 8 H.-J. Bungartz, F. Durst, C. Zenger (eds.), *High Performance Scientific and Engineering Computing.* Proceedings of the International FORTWIHR Conference on HPSEC, Munich, March 16-18, 1998. 1999. X, 471 pp. Softcover. 3-540-65730-4

Vol. 9 T. J. Barth, H. Deconinck (eds.), *High-Order Methods for Computational Physics.* 1999. VII, 582 pp. Hardcover. 3-540-65893-9

Vol. 10 H. P. Langtangen, A. M. Bruaset, E. Quak (eds.), *Advances in Software Tools for Scientific Computing.* 2000. X, 357 pp. Softcover. 3-540-66557-9

Vol. 11 B. Cockburn, G. E. Karniadakis, C.-W. Shu (eds.), *Discontinuous Galerkin Methods.* Theory, Computation and Applications. 2000. XI, 470 pp. Hardcover. 3-540-66787-3

Vol. 12 U. van Rienen, *Numerical Methods in Computational Electrodynamics.* Linear Systems in Practical Applications. 2000. XIII, 375 pp. Softcover. 3-540-67629-5

Vol. 13 B. Engquist, L. Johnsson, M. Hammill, F. Short (eds.), *Simulation and Visualization on the Grid.* Parallelldatorcentrum Seventh Annual Conference, Stockholm, December 1999, Proceedings. 2000. XIII, 301 pp. Softcover. 3-540-67264-8

Vol. 14 E. Dick, K. Riemslagh, J. Vierendeels (eds.), *Multigrid Methods VI.* Proceedings of the Sixth European Multigrid Conference Held in Gent, Belgium, September 27-30, 1999. 2000. IX, 293 pp. Softcover. 3-540-67157-9

Vol. 15 A. Frommer, T. Lippert, B. Medeke, K. Schilling (eds.), *Numerical Challenges in Lattice Quantum Chromodynamics.* Joint Interdisciplinary Workshop of John von Neumann Institute for Computing, Jülich and Institute of Applied Computer Science, Wuppertal University, August 1999. 2000. VIII, 184 pp. Softcover. 3-540-67732-1

Vol. 16 J. Lang, *Adaptive Multilevel Solution of Nonlinear Parabolic PDE Systems.* Theory, Algorithm, and Applications. 2001. XII, 157 pp. Softcover. 3-540-67900-6

Vol. 17 B. I. Wohlmuth, *Discretization Methods and Iterative Solvers Based on Domain Decomposition.* 2001. X, 197 pp. Softcover. 3-540-41083-X

Vol. 18 U. van Rienen, M. Günther, D. Hecht (eds.), *Scientific Computing in Electrical Engineering.* Proceedings of the 3rd International Workshop, August 20-23, 2000, Warnemünde, Germany. 2001. XII, 428 pp. Softcover. 3-540-42173-4

Vol. 19 I. Babuška, P. G. Ciarlet, T. Miyoshi (eds.), *Mathematical Modeling and Numerical Simulation in Continuum Mechanics.* Proceedings of the International Symposium on Mathematical Modeling and Numerical Simulation in Continuum Mechanics, September 29 - October 3, 2000, Yamaguchi, Japan. 2002. VIII, 301 pp. Softcover. 3-540-42399-0

Vol. 20 T. J. Barth, T. Chan, R. Haimes (eds.), *Multiscale and Multiresolution Methods.* Theory and Applications. 2002. X, 389 pp. Softcover. 3-540-42420-2

Vol. 21 M. Breuer, F. Durst, C. Zenger (eds.), *High Performance Scientific and Engineering Computing.* Proceedings of the 3rd International FORTWIHR Conference on HPSEC, Erlangen, March 12-14, 2001. 2002. XIII, 408 pp. Softcover. 3-540-42946-8

Vol. 22 K. Urban, *Wavelets in Numerical Simulation.* Problem Adapted Construction and Applications. 2002. XV, 181 pp. Softcover. 3-540-43055-5

Vol. 23 L. F. Pavarino, A. Toselli (eds.), *Recent Developments in Domain Decomposition Methods.* 2002. XII, 243 pp. Softcover. 3-540-43413-5

Vol. 24 T. Schlick, H. H. Gan (eds.), *Computational Methods for Macromolecules: Challenges and Applications.* Proceedings of the 3rd International Workshop on Algorithms for Macromolecular Modeling, New York, October 12-14, 2000. 2002. IX, 504 pp. Softcover. 3-540-43756-8

Vol. 25 T. J. Barth, H. Deconinck (eds.), *Error Estimation and Adaptive Discretization Methods in Computational Fluid Dynamics.* 2003. VII, 344 pp. Hardcover. 3-540-43758-4

Vol. 26 M. Griebel, M. A. Schweitzer (eds.), *Meshfree Methods for Partial Differential Equations*. 2003. IX, 466 pp. Softcover. 3-540-43891-2

Vol. 27 S. Müller, *Adaptive Multiscale Schemes for Conservation Laws*. 2003. XIV, 181 pp. Softcover. 3-540-44325-8

Vol. 28 C. Carstensen, S. Funken, W. Hackbusch, R. H. W. Hoppe, P. Monk (eds.), *Computational Electromagnetics*. Proceedings of the GAMM Workshop on "Computational Electromagnetics", Kiel, Germany, January 26-28, 2001. 2003. X, 209 pp. Softcover. 3-540-44392-4

Vol. 29 M. A. Schweitzer, *A Parallel Multilevel Partition of Unity Method for Elliptic Partial Differential Equations*. 2003. V, 194 pp. Softcover. 3-540-00351-7

Vol. 30 T. Biegler, O. Ghattas, M. Heinkenschloss, B. van Bloemen Waanders (eds.), *Large-Scale PDE-Constrained Optimization*. 2003. VI, 349 pp. Softcover. 3-540-05045-0

Vol. 31 M. Ainsworth, P. Davies, D. Duncan, P. Martin, B. Rynne (eds.), *Topics in Computational Wave Propagation*. Direct and Inverse Problems. 2003. VIII, 399 pp. Softcover. 3-540-00744-X

Vol. 32 H. Emmerich, B. Nestler, M. Schreckenberg (eds.), *Interface and Transport Dynamics*. Computational Modelling. 2003. XV, 432 pp. Hardcover. 3-540-40367-1

Vol. 33 H. P. Langtangen, A. Tveito (eds.), *Advanced Topics in Computational Partial Differential Equations*. Numerical Methods and Diffpack Programming. 2003. XIX, 658 pp. Softcover. 3-540-01438-1

Vol. 34 V. John, *Large Eddy Simulation of Turbulent Incompressible Flows*. Analytical and Numerical Results for a Class of LES Models. 2004. XII, 261 pp. Softcover. 3-540-40643-3

Vol. 35 E. Bänsch (ed.), *Challenges in Scientific Computing - CISC 2002*. Proceedings of the Conference *Challenges in Scientific Computing*, Berlin, October 2-5, 2002. 2003. VIII, 287 pp. Hardcover. 3-540-40887-8

Vol. 36 B. N. Khoromskij, G. Wittum, *Numerical Solution of Elliptic Differential Equations by Reduction to the Interface*. 2004. XI, 293 pp. Softcover. 3-540-20406-7

Vol. 37 A. Iske, *Multiresolution Methods in Scattered Data Modelling*. 2004. XII, 182 pp. Softcover. 3-540-20479-2

Vol. 38 S.-I. Niculescu, K. Gu (eds.), *Advances in Time-Delay Systems*. 2004. XIV, 446 pp. Softcover. 3-540-20890-9

Vol. 39 S. Attinger, P. Koumoutsakos (eds.), *Multiscale Modelling and Simulation*. 2004. VIII, 277 pp. Softcover. 3-540-21180-2

Vol. 40 R. Kornhuber, R. Hoppe, J. Périaux, O. Pironneau, O. Wildlund, J. Xu (eds.), *Domain Decomposition Methods in Science and Engineering*. 2005. XVIII, 690 pp. Softcover. 3-540-22523-4

Vol. 41 T. Plewa, T. Linde, V.G. Weirs (eds.), *Adaptive Mesh Refinement – Theory and Applications.* 2005. XIV, 552 pp. Softcover. 3-540-21147-0

Vol. 42 A. Schmidt, K.G. Siebert, *Design of Adaptive Finite Element Software.* The Finite Element Toolbox ALBERTA. 2005. XII, 322 pp. Hardcover. 3-540-22842-X

Vol. 43 M. Griebel, M.A. Schweitzer (eds.), *Meshfree Methods for Partial Differential Equations II.* 2005. XIII, 303 pp. Softcover. 3-540-23026-2

Vol. 44 B. Engquist, P. Lötstedt, O. Runborg (eds.), *Multiscale Methods in Science and Engineering.* 2005. XII, 291 pp. Softcover. 3-540-25335-1

Vol. 45 P. Benner, V. Mehrmann, D.C. Sorensen (eds.), *Dimension Reduction of Large-Scale Systems.* 2005. XII, 402 pp. Softcover. 3-540-24545-6

For further information on these books please have a look at our mathematics catalogue at the following URL: `www.springeronline.com/series/3527`

Texts in Computational Science and Engineering

Vol. 1 H. P. Langtangen, *Computational Partial Differential Equations.* Numerical Methods and Diffpack Programming. 2nd Edition 2003. XXVI, 855 pp. Hardcover. ISBN 3-540-43416-X

Vol. 2 A. Quarteroni, F. Saleri, *Scientific Computing with MATLAB.* 2003. IX, 257 pp. Hardcover. ISBN 3-540-44363-0

Vol. 3 H. P. Langtangen, *Python Scripting for Computational Science.* 2004. XXII, 724 pp. Hardcover. ISBN 3-540-43508-5

For further information on these books please have a look at our mathematics catalogue at the following URL: `www.springeronline.com/series/5151`