A Thick-Restarted Krylov Subspace Projection Method for Model Order Reduction

By

EFREM B. RENSI

B.S. (San Jose State University, San Jose) 2006

THESIS

Submitted in partial satisfaction of the requirements for the degree of

MASTER OF SCIENCE

in

Mathematics

in the

OFFICE OF GRADUATE STUDIES

of the

UNIVERSITY OF CALIFORNIA

DAVIS

Approved:

Roland Freund

Robert Guy

Zhaojun Bai

Committee in Charge

2014

Efrem Rensi
June 2014
GGAM (Applied Mathematics)

Abstract

The major contribution of this thesis is a new thick-restarted Krylov method that allows for re-starting the process at different interpolation-points while preserving moment-matching and providing a controllable degree of linear-dependence of the resulting basis. It reduces computational-cost by recycling a controllable number of vectors per cycle. The algorithms comprising the method are outlined and their efficiency is demonstrated by applying it to selected test-models.

Contents

Acknowledgments

Thanks to my parents Cornelia and Giuseppe Rensi, and the rest of my family!

Maria “Bem” Cayco!!!!! who has been my friend, confidant, and mentor since my undergraduate years at San Jose State!

My office-mate for 5 years and colleague Yuji Nakatsukasa!

Thanks to the Math departement staff, especially Sylvia Davis and Tina Deneena.

The Davis Bike Collective!

Chapter 1

Introduction

1.0.1 What is model order reduction?

Model order-reduction (MOR), also known as model-reduction, is the process by which a mathematical-model characterized by some number of parameters N, referred to here as the unreduced model (URM), is approximated by a so-called reduced-order model (ROM) of size $n < N$.

Model reduction can be likened to digital-media compression. We rarely work with images, video, or audio files or streams at their original resolution; that usually requires more computing power, memory, or bandwidth than we have available. Instead, we use lower resolution approximations that are "good enough" for our purposes. We compress the data. Significant data compression is generally lossy, meaning the reduced data contains less information than the original in some sense, which is observed as a reduction in quality. Even as available memory and computation power increase, we continue to have problems that require more power or memory than we have available.

In general, model reduction can be applied to a model of any working thing whose behavior over time can be observed and influenced (i.e. has output and input). We seek a simpler description of that thing while preserving its behavior in response to various inputs. Order-reduction of the specific type of model that we will address has its origins in systems and control theory, where it has been applied to automate processes. If the model correctly predicts a system's response to given input, we can use the model to determine how to efficiently control or influence the system. An

early example of modern model reduction is [10] from 1966. Davison introduced the MOR method now known as modal truncation,

> *Often it is possible to represent physical systems by a number of simultaneous linear differential equations with constant coefficients,* $\dot{x} = Ax + r$ *but for many processes (e.g., chemical plants, nuclear reactors), the order of the matrix* A *may be quite large, say* 50×50*,* 100×100*, or even* 500×500*. It is difficult to work with these large matrices and a means of approximating the system matrix by one of lower order is needed. A method is proposed for reducing such matrices by constructing a matrix of lower order which has the same dominant eigenvalues and eigenvectors as the original system.*

The "large" systems we work with today are much larger. At the time of this writing, Krylov-subspace methods are being used for systems on the order of $N = 10^9$. Krylov-subspace methods are particularly suited for working with systems that are too large for other methods, many of which are quite elegant. A more in-depth survey of model order reduction methods can be found in [38]. The reason why Krylov methods are suited for such large systems is that the only large matrix operation required is a matrix multiplication, typically using a sparse matrix (more correctly a sparse solve of an order N system). The major computational expense is simply, repeatedly multiplying a large sparse matrix with a vector or block of vectors.

1.0.2 Some background

Krylov projection methods for working with large matrices have been around for a while, and are even claimed to be in the top 10 most important classes of algorithms of the 20th century [12]. The first Krylov-subspace projection method for model order-reduction was PVL (Padévia Lanczos) [15] developed by Feldman and Freund in 1995. Prior to that, Krylov methods were very successful in

- approximating eigenvalues of large matrices,[28, 4] circa early 1950s,
- approximating the solution $Ax = b$ to large sparse systems ([42] offers a great background),
- approximating matrix function multiplications $f(A)b$ (like $e^A b$) when A is very large [14].

It is not a surprise then that the most successful methods for model reduction (MOR) of a very large, sparse, descriptor system model are iterative methods, and Krylov methods in particular.

Relation to the eigenvalue problem

Krylov-subspace methods for model order-reduction have been developed out of methods for finding eigenvalues of very large, sparse matrices. The problem of finding certain eigenvalues of a large matrix and that of finding an approximate solution of a large dynamical system model are closely related. This is because the system can be represented as a linear operator whose action is expressed as operations (multiplication and back-solve) with large sparse matrices. Constructing a good reduced-order approximation in that respect is equivalent to constructing smaller approximate system matrices that retain "essential" eigen-information of the original system, which is usually not known beforehand. We wish to construct approximations that retain only enough essential information to be "pretty good", where that depends on the particular application of the model. A Krylov method iteratively searches the solution space of the model via the eigen-space of its system matrices.

An issue that is particular to model order-reduction, and not to eigen-analysis in general, is that the start vector for the search is pre-defined. That is because iterations of the Krylov process are identified with terms of a Taylor series approximation to the model's system function about some interpolation-point(s) σ. This is known as moment matching (about σ) and is important for determining exactly where the reduced order approximation is accurate, and to what degree. As such, the moment matching property of a Krylov subspace method only ensures local approximation. MOR methods that guarantee global error-bounds, such as balanced-truncation [30] do exist but are more expensive.

Restarted Krylov-subspace methods

A major issue is the computational cost of current methods. Unmodified Krylov processes become more expensive as they progress incrementally, as each new basis vector must be made orthogonal to every previous vector. A few restart schemes have been tried and implemented successfully for the eigenvalue problem and other applications of Krylov processes, some of which involve re-starting the search with a new start vector that better approximates the desired eigenvector(s).

The thick-restart was introduced by Wu in [44] and [47] as an alternative to Lehoucq and

Sorensen's implicit restart applied to Krylov methods for finding eigenvalues of very large matrices or matrix-pencils. The reason for a restart in an eigenvalue method is to deal with storage limitations on the number of basis vectors we can hold, and/or computation cost of orthogonalizing each iterate against n previous vectors on the n-th step. Suppose our memory limits us to storing n vectors, or $\mathcal{O}(nN)$ is the longest we can wait to orthogonalize a new basis vector. Then we must stop the process and throw out some vectors every time we fill n positions. More precisely, we transform the vector basis V into another basis $\begin{bmatrix} Y & Y^{\perp} \end{bmatrix}$ such that we can throw out $Y^{\perp}$ in the so-called deflation phase, and then continue, (re-starting) with Y, until we fill up n positions again. The subspace $Y^{\perp}$ that we throw out typically corresponds to unconverged eigen-information, or perhaps we keep certain nearly-converged approximate eigen-information in Y that will separate the remaining space and make certain unconverged eigenvalues converge faster. We continue to refine our set of n vectors, cycle after restarted cycle, until we have the set of n vectors we want.

Restart in MOR

Restarted methods for eigenvalue finding refine one limited collection of vectors until they converge into a desired set. The restart for model reduction serves a different purpose, although there are methods [26] and notable [24] that construct a ROM using such a process of refining a projection subspace. The idea has usually been to throw out bad poles of an implicitly projected ROM, refining the projection subspace until there are no poles with positive $\Re$-part or some other undesirable trait. The problem with this is that matching moments of a model transfer-function about an interpolation point $\sigma \in \mathbb{C}$ via projection on to a subspace V requires that the desired Krylov subspace is contained in V. If we throw out basis vectors then we have no guarantee of moments matched about σ.

Our thick-restarted method proposed in §5.2 does not throw out any basis vectors so moment matching about an interpolation point is preserved. The thick-restart in this case allows us to change the interpolation-point after a cycle and avoid re-discovering sufficiently converged invariant subspace on the next cycle.

1.1 Motivation from the literature

In lieu of a full literature review, we quote S. Gugercin from the 2005 paper [25], with references.

> **Krylov-based model reduction in the MIMO case:** Even though [a theorem about multi-point moment matching] explains how to solve the rational interpolation problem theoretically for the MIMO case as well as the SISO case, implementation of the underlying Krylov-based method becomes a much harder task for MIMO systems. The main difficulties are the needs for a block rational Krylov method and an effective deflation strategy. As for the single interpolation point case, a MIMO version of the Arnoldi procedure with deflation has been introduced by Boley in [6]. On the other hand, the problem for the MIMO Lanczos procedure has not been completely solved until very recently. For the non-symmetric case, many authors have provided only block versions of the algorithm without the deflation procedure; see for example [5, 27, 33]. However these block-wise approaches only tackled the problems with the same number of inputs and outputs. This obstacle has been overcome in [39, 31, 9] only for the symmetric case. *For the most general problem, namely the nonsymmetric case with arbitrary number of inputs and outputs, a solution has been recently presented by Freund et al. in [3] via a vector-wise construction of the underlying Krylov subspace with deflation. As for the MIMO systems and multiple interpolation points, the author is not aware of an effective implementation to address the MIMO (block) rational Krylov reduction except the special case of tangential interpolation by Gallivan et al. [19].*

In this thesis we provide the method that the emphasized passage[1] seems to imply. Our method extends the band-Krylov procedure [3] to address rational-interpolation, and introduces thick-restarts as part of the process. As far as we know, there is only one other published implementation of a restarted band-Arnoldi process, the "Restarted Block Arnoldi Method in a Vector-Wise Fashion" [48] by Yin and Lu, and their method is proposed for the eigenvalue problem. Our method is a similar extension of the band-Arnoldi process, but with model order-reduction in mind.

We note that our implementation is of an *explicit*-restart, although an implicit-restart could be implemented as well. The implicit-restart is mathematically equivalent, but more efficient. Since we were primarily interested in exploring the efficacy of a restarted band-Krylov method for model reduction, we decided to leave implementation of an implicit-restart to further research.

[1] our emphasis

1.2 Content overview

We address the question of whether augmenting a Band-Krylov process with Ritz-vectors can be usefully implemented in the development of more efficient Krylov-subspace projection methods for model order-reduction. Chapters 2 and 3 introduce the basic concepts for a theory of model order-reduction via Krylov-subspace projection.

Chapters 4 and 5 use the analytical framework outlined in prior chapters to argue the potential usefulness of thick-restarting the band-Krylov algorithm for more efficient model-reduction methods.

Chapter 2

Chapter 2 introduces the matrix-valued system transfer-function

$$\mathcal{H} : \mathbb{C} \to \mathbb{C}^{m \times p} \tag{1}$$

that characterizes the input-output relationship of a system with m inputs and p outputs. An example of such a model is a signal processor, which takes an input signal and outputs a modified response signal. For a given fundamental-signal $s \in \mathbb{C}$, the i, j-th component of $\mathcal{H}(s)$ gives the system's response from output-terminal j when s is input into input-terminal i. We would like to create a ROM such that its transfer-function $\widetilde{\mathcal{H}}$ approximates the URM transfer-function $\mathcal{H}$, so in some sense the problem of model reduction is that of approximating the function (1) over a given subset of $\mathbb{C}$.

The transfer function can be represented and approximated by a quotient of two polynomials, or as a Taylor-series, and either representation is taken with respect to an expansion-point, also referred to as an interpolation-point. Approximating (1) with some number of Taylor series terms is known as moment-matching.

The transfer function can also be represented as a rational function, which for $m = p = 1$ is the quotient

$$\mathcal{H}(s) = \frac{P(s)}{Q(s)}$$

of two polynomials, where zeros of $Q(s)$ are known as poles of the transfer-function. One way to approximate (1) is via its poles and zeros; poles are of particular interest in model reduction. Some poles have more influence on the quality of approximation than others, and we introduce a measure of this influence called pole-weight.

Chapter 3

In chapter 3 we begin a discussion of Krylov subspace based model-reduction methods. We introduce the Krylov subspace, and the Arnoldi algorithm, which is the most basic Krylov process used for subspace-projection based MOR. The Arnoldi algorithm was originally used to find eigenvalues/vectors of large matrices and we address how that application is related to model-reduction.

We define two different ROM transfer-functions that can be constructed with a given Krylov-subspace basis: the ROM via *implicit-projection* (or implicitly-projected), and that via *explicit-projection*. The former is cheaper to construct but is not desirable for use in applications because it can have "bad" poles. It is useful for analysis of the converging approximate model while we are constructing it. The explicit-projection ROM is the end-product of our method. It is not constructed until the process is done.

Chapter 4

Chapter 4 addresses some of the issues that arise with use of multiple interpolation-points in $\mathbb{C}$ as opposed to one point in $\mathbb{R}$. A single real expansion-point is often chosen to avoid the computational cost of complex arithmetic. We provide a translation of a Ritz-space associated with one interpolation-point to another in §4.1.2, which may be novel.

In §4.2.2 we introduce a possible improvement for the orthogonalization step of a Krylov-process that reduces orthogonalization costs by half, and is fairly simple to implement. As far as we know this is a novel contribution to the field.

Chapter 5

Chapter 5 introduces the Band-Arnoldi algorithm, a generalization of the Arnoldi algorithm that allows for iterating simultaneously with several candidate vectors while remaining essentially a single-vector iterative process. The vector-wise iteration of a band process sacrifices the ability to take advantage of efficient matrix-matrix multiplication algorithms, which is why block-wise methods like [29] currently prevail for MIMO model reduction. These methods iterate an entire block of vectors on each step, but we feel that the band-process with vector-wise iteration allows for a simpler and more intuitive restart-scheme. Restarting a Krylov process is done generally to reduce computational costs of orthogonalizing each iterate vector against every previous one, and we can take advantage of this to restart at a different interpolation-point. The "thick" modifier indicates that on each cycle we re-use some of the information from the previous cycle, but not all of it as with full-orthogonalization.

Exploration of thick-restarting the Band-Arnoldi process was the original purpose of our research that led to this document, and is the thesis of our work. We show how thick-restarting the Band-Arnoldi process can be incorporated into a model-reduction method. It is debatable whether the method presents an improvement over existing methods in general, but we do show via tests with a few publicly available test models that that a comparable or better degree of reduction can be achieved with multiple complex points than with a single real interpolation point; also, with minimal drawbacks, thick-restarting improves efficiency of basis-construction over a multiple-point method using full-orthogonalization.

Chapter 2

The model and its transfer-function

2.1 The LTI-system model

Our basic problem is to approximate the (N-dimensional) linear, time-invariant (LTI) descriptor-system

$$\begin{aligned} \boldsymbol{E}\frac{dx}{dt} &= \boldsymbol{A}x + \boldsymbol{B}u \\ y &= \boldsymbol{C}^T x \end{aligned} \tag{2}$$

with a system that is realized by smaller matrices $\boldsymbol{A}_n$, $\boldsymbol{E}_n$, $\boldsymbol{B}_n$, and $\boldsymbol{C}_n$. The collection of constant matrices $(\boldsymbol{A}, \boldsymbol{E}, \boldsymbol{B}, \boldsymbol{C})$ is called a realization of the model, which we sometimes call the unreduced model (URM).

Matrices $\boldsymbol{E}$ and $\boldsymbol{A}$ are singular in general. We only assume that $\boldsymbol{A} - s\boldsymbol{E}$ is invertible for all $s \in \mathbb{C}$, except for a finite set of so-called eigenvalues.

$\boldsymbol{B} \in \mathbb{R}^{N\times m}$ and $\boldsymbol{C} \in \mathbb{R}^{N\times p}$ are matrices that condition the input signal $u(t) \in \mathbb{R}^m$ and output signal $y(t) \in \mathbb{R}^p$. If $p = m = 1$ then (2) is a single-input, single-output (SISO) system; its response (output) is a scalar-valued function. If the dimension of $\boldsymbol{B}$ and $\boldsymbol{C}$ are both greater than one then (2) is a multi-input, multi-output (MIMO) system.

We assume that the order-N system (2) is too large to work with and and we want a model that behaves like (2), but with significantly reduced state-space dimension n.

Suppose that, for some reason, we believe restricting the state space of the model (2) to an

n-dimensional subspace $\mathcal{K}$, will yield a good reduced-order model. If V is an orthogonal basis of $\mathcal{K}$ then the reduced-order model (ROM) obtained via orthogonal projection onto $\mathcal{K}$ is a new descriptor system

$$\begin{aligned} \boldsymbol{E}_n \frac{d\tilde{x}}{dt} &= \boldsymbol{A}_n \tilde{x} + \boldsymbol{B}_n u \\ \tilde{y} &= \boldsymbol{C}_n^T \tilde{x}, \end{aligned} \tag{3}$$

with orthogonal projections

$$\boldsymbol{A}_n = V^T \boldsymbol{A} V, \quad \boldsymbol{E}_n = V^T \boldsymbol{E} V, \quad \boldsymbol{C}_n = V^T \boldsymbol{C}, \quad \boldsymbol{B}_n = V^T \boldsymbol{B}, \tag{4}$$

where $\tilde{x}(t) \in \mathbb{C}^n$ is the state of the reduced-order system such that $V\tilde{x}(t)$ approximates the state of the unreduced model. The p output(s) $\tilde{y}(t) \in \mathbb{R}^p$ approximate $y(t) \in \mathbb{R}^p$ from (2), given the same m input(s) $u(t) \in \mathbb{R}^m$, and ideally $\|y - \tilde{y}\|$ is small.

The reduced-order model (3), (4) is called the *explicitly-projected* ROM, because in the computational setting we must actually compute (4). Ultimately the desired ROM is in this form.

2.2 System transfer-function

The transfer-function is a direct relationship between input and output of the model in the frequency domain. If we temporarily ignore the state of the model and view it simply as a mapping of an input signal u, to an output signal y, the system (2) acts as a system-function $y = h(u)$. The transfer-function is obtained by applying the Laplace transform (eg. $X(s) = \mathcal{L}\{x(t)\}$) to (2) and assuming a zero initial condition $X(0) = 0$, which yields the algebraic equations

$$\begin{aligned} s\boldsymbol{E}X &= \boldsymbol{A}X + \boldsymbol{B}U, \\ Y &= \boldsymbol{C}^T X. \end{aligned}$$

Then $Y(s) = \mathcal{H}(s)U(s)$, where

$$\mathcal{H}(s) = \boldsymbol{C}^T \left(s\boldsymbol{E} - \boldsymbol{A}\right)^{-1} \boldsymbol{B} \tag{5}$$

is the transfer-function over $\mathbb{C}$. Note that $\mathcal{H}(s)$ is defined only if the matrix pencil $(\boldsymbol{A}, \boldsymbol{E})$ is regular, meaning that the matrix $\boldsymbol{A} - s\boldsymbol{E}$ is invertible for all but a finite set of eigenvalues.

For a general MIMO transfer-function (5) where $\boldsymbol{C} = \begin{bmatrix} \boldsymbol{c}_1 & \boldsymbol{c}_2 & \dots & \boldsymbol{c}_p \end{bmatrix}$ and $\boldsymbol{B} = \begin{bmatrix} \boldsymbol{b}_1 & \boldsymbol{b}_2 & \dots & \boldsymbol{b}_m \end{bmatrix}$, we can consider (5) to be mp scalar-valued SISO (single input single output) transfer-functions

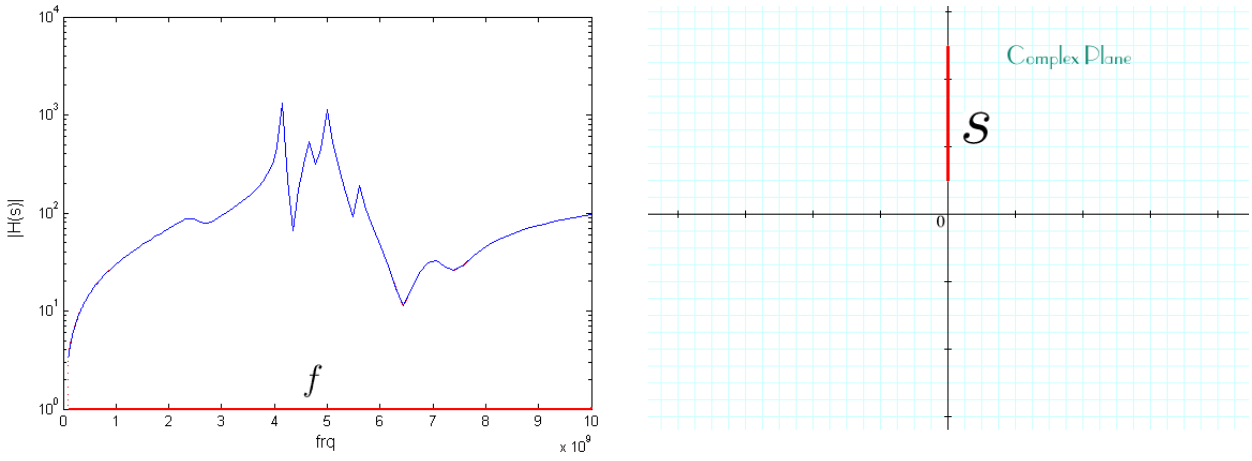
$$\mathcal{H}_{ij}(s) = \boldsymbol{c}_i^T \left(s\boldsymbol{E} - \boldsymbol{A}\right)^{-1} \boldsymbol{b}_j \in \mathbb{C},$$

and for example in the 2×2 case we have

$$\mathcal{H}(s) = \begin{bmatrix} \mathcal{H}_{11}(s) & \mathcal{H}_{12}(s) \\ \mathcal{H}_{21}(s) & \mathcal{H}_{22}(s) \end{bmatrix}.$$

Frequency response

We are primarily interested in producing a reduced-order system that exhibits the same frequency response as the original system, up to some error. Frequency response $\mathcal{H}(i\omega)$ is the transfer function over the positive $\Im$-axis. Figure 2.1 illustrates the relationship between the transfer function domain and its frequency response gain $|\mathcal{H}(2\pi i f)|$ plot of a test-system. 2.1a in particular is the $(1,1)$ component $\mathcal{H}_{11}(2\pi i f)$ of a 16×16 MIMO circuit model `ex1841` over frequencies $f \in \left[10^8, 10^{10}\right]$.



(a) (b)

Figure 2.1: Frequency response gain $|\mathcal{H}(2\pi i f)|$ for a single-input, single-output (SISO) model with $f \in \left[10^8, 10^{10}\right]$. It is plotted on a semi-log scale (logarithmic for gain-axis, linear for f-axis). The plot (a) is what $|\mathcal{H}(s)|$ looks like over the segment S in (b). We are generally interested in approximating $\mathcal{H}(s)$ accurately over an interval $S = i(\omega_0, \omega_1)$ on the $\Im$-axis.

2.2.1 Moments

The transfer-function is a rational function, and thus can be represented by a Taylor series about an expansion-point $\sigma \in \mathbb{C}$, having the general form

$$\mathcal{H}(s) = \sum_{j=0}^{\infty} (s-\sigma)^j \mathcal{H}^{(j)}, \quad \text{or equivalently,} \quad \mathcal{H}(s+\sigma) = \sum_{j=0}^{\infty} s^j \mathcal{H}^{(j)} \tag{6}$$

where the Taylor coefficient

$$\mathcal{H}^{(j)} = \frac{1}{j!} \frac{d^j \mathcal{H}}{ds^j}\bigg|_{s=\sigma} \tag{7}$$

is called the j-th *moment* of the transfer-function about σ.

Moment-matching

Krylov-subspace projection methods boast moment-matching properties, which we prove as theorems 1 and 2 in §3.4. Moment matching means that the reduced-order model transfer-function implied by a Krylov-subspace method is guaranteed to share a number of terms of the Taylor series with that of the full unreduced model, about a given expansion-point σ.

Suppose the URM (unreduced model) transfer-function expressed as a Taylor series about σ is

$$\mathcal{H}(s) = \mathcal{H}^{(0)} + (s-\sigma)\mathcal{H}^{(1)} + (s-\sigma)^2\mathcal{H}^{(2)} + \cdots + (s-\sigma)^{n-1}\mathcal{H}^{(n-1)} + \cdots .$$

A reduced-order model (ROM) whose transfer-function can be written as

$$\widehat{\mathcal{H}}(s) = \widehat{\mathcal{H}}^{(0)} + (s-\sigma)\widehat{\mathcal{H}}^{(1)} + (s-\sigma)^2\widehat{\mathcal{H}}^{(2)} + \cdots + (s-\sigma)^{n-1}\widehat{\mathcal{H}}^{(n-1)} + \cdots$$

where

$$\widehat{\mathcal{H}}^{(j)} = \mathcal{H}^{(j)} \quad \text{for} \quad j = 0, 1, 2, \cdots, n-1$$

is said to match n-moments about σ.

Moments can be matched about any number of expansion-points; also called interpolation-points.

2.2.2 Shift-invert representation of the transfer-function

Moment matching properties of Krylov-subspace methods are accomplished via the the following reformulation of the transfer-function (5).

Let $\sigma \in \mathbb{C}$ be a point for which $\sigma \boldsymbol{E} - \boldsymbol{A}$ is invertible. Then

$$\begin{aligned} \mathcal{H}(s) &= \boldsymbol{C}^T \left(s\boldsymbol{E} - \boldsymbol{A}\right)^{-1} \boldsymbol{B} \\ &= \boldsymbol{C}^T \left(\sigma\boldsymbol{E} - \boldsymbol{A} + (s-\sigma)\boldsymbol{E}\right)^{-1} \boldsymbol{B} \\ &= \boldsymbol{C}^T \left(I - (s-\sigma)\mathbf{H}\right)^{-1} \mathbf{R} \end{aligned} \tag{8}$$

where[1]

$$\mathbf{H} := (\boldsymbol{A} - \sigma\boldsymbol{E})^{-1}\boldsymbol{E} \quad \text{and} \quad \mathbf{R} := (\sigma\boldsymbol{E} - \boldsymbol{A})^{-1}\boldsymbol{B}. \tag{9}$$

(8) is sometimes called the shifted transfer-function formulation, with shift σ, although it does not depend on σ. The shift matters when we consider the ROM transfer-function that approximates (8).

In Krylov subspace methods the generally non-sparse $\mathbf{H} = \mathbf{H}(\sigma) \in \mathbb{C}^{N\times N}$ is a sort of operator or multiplier that acts on $\mathbf{R} = \mathbf{R}(\sigma) \in \mathbb{C}^{N\times p}$. $\mathbf{H}$ is dense in general and is rarely if ever explicitly formed. We only need a way to obtain matrix-vector products $\mathbf{H}v$ for vectors $v \in \mathbb{C}^N$.

The shifted transfer-function representation (8) can alternatively be considered the transfer-function for the shifted descriptor system

$$\begin{aligned} \mathbf{H}\frac{dx}{dt} &= (I - \sigma\mathbf{H})x + \mathbf{R}u \\ y &= \boldsymbol{C}^T x, \end{aligned} \tag{10}$$

which is equivalent to (2) for any σ such that $\boldsymbol{A} - \sigma\boldsymbol{E}$ is invertible. Some order-reduction schemes notably work by replacing $\mathbf{H}$, $\mathbf{R}$, and and $\boldsymbol{C}$ with reduced-order approximations $\widetilde{\mathbf{H}} = V^T\mathbf{H}V$, $\widetilde{\boldsymbol{\rho}}_n = V^T\mathbf{R}$, and $\boldsymbol{C}_n = V^T\boldsymbol{C}$. We call such a ROM *implicitly* projected on to span V, as opposed to the *explicitly* projected model (3). A model obtained via implicit-projection is not equivalent to (3) in general and is undesirable for some applications because the ROM can contain unstable modes, but it is cheaper to construct because $\widetilde{\mathbf{H}}$ and $\widetilde{\boldsymbol{\rho}}$ are produced by the same (Krylov) process that

[1] To verify (8), note that $I = (\sigma\boldsymbol{E} - \boldsymbol{A})^{-1}(\sigma\boldsymbol{E} - \boldsymbol{A})$.

creates the basis V. Then the only explicit computation needed to construct the ROM transfer-function is $\boldsymbol{C}_n = V^T \boldsymbol{C}$.

Moment representation

We now express transfer-function moments about σ in terms of $\mathbf{H}$ and $\mathbf{R}$. Via Neumann series expansion (power series for matrices) re-write (8) as

$$\begin{aligned} \mathcal{H}(s) &= \boldsymbol{C}^T \left(\sum_{j=0}^{\infty} (s-\sigma)^j \mathbf{H}^j \right) \mathbf{R} \\ &= \sum_{j=0}^{\infty} (s-\sigma)^j \boldsymbol{C}^T \mathbf{H}^j \mathbf{R}. \end{aligned} \tag{11}$$

The moments $\mathcal{H}^{(j)}$ from (6) are specified exactly in (11):

$$\mathcal{H}^{(j)} = \boldsymbol{C}^T \mathbf{H}^j \mathbf{R}. \tag{12}$$

2.2.3 Invariant-subspaces

Given a transformation $\mathbf{H} : \mathbb{C}^N \to \mathbb{C}^N$, a subspace $\mathcal{Q}$ of $\mathbb{C}^N$ is called $\mathbf{H}$-invariant if

$$\mathbf{H}\mathcal{Q} \subseteq \mathcal{Q}. \tag{13}$$

Fore example, the span of a set of eigenvectors of $\mathbf{H}$ is an invariant subspace under $\mathbf{H}$.

If Q is a basis for $\mathcal{Q}$ then

$$\mathbf{H}Q = QT \tag{14}$$

for some matrix $T \in \mathbb{C}^{\ell \times \ell}$. If the basis vectors are eigenvectors Z then (14) becomes

$$\mathbf{H}Z = \Lambda Z,$$

where $\Lambda = \text{diag}\{\lambda_1, \lambda_2, ..., \lambda_\ell\}$ is a diagonal matrix of eigenvectors associated with the vectors $Z = \begin{bmatrix} z_1 & z_2 & \cdots & z_\ell \end{bmatrix}$.

If the basis $Q = \begin{bmatrix} u_1 & u_2 & \cdots & u_\ell \end{bmatrix}$ for the $\mathbf{H}$-invariant-subspace

$$\mathcal{Q} = \operatorname{span} \begin{bmatrix} u_1 & u_2 & \cdots & u_\ell \end{bmatrix} = \operatorname{span} \begin{bmatrix} z_1 & z_2 & \cdots & z_\ell \end{bmatrix}$$

is orthonormal then we call vectors u_j Schur-vectors and sometimes call $\mathcal{Q}$ a Schur space. Also, (14) is called a Schur-decomposition, and T is upper triangular with eigenvalues λ_j associated with z_j along its diagonal.

Generalized-eigenvalues and invariance

An eigenvalue of matrix pencil $(\boldsymbol{A}, \boldsymbol{E})$, called a generalized eigenvalue, is a $\mu \in \mathbb{C}$ such that $(\boldsymbol{A} - \mu \boldsymbol{E})z = 0$ has nonzero solutions $z \neq 0 \in \mathbb{C}^N$, which are the right-eigenvectors of $(\boldsymbol{A}, \boldsymbol{E})$.

The notion of invariance under a general operator $\mathbf{H}$ extends to that of a matrix pencil. The subspace $\mathcal{Q} = \operatorname{span} Z$ is called invariant, or deflating, [45] with respect to $(\boldsymbol{A}, \boldsymbol{E})$ if

$$\dim (\boldsymbol{A}\mathcal{Q} + \boldsymbol{E}\mathcal{Q}) \leq \dim \mathcal{Q}. \tag{15}$$

For a regular matrix pencil $(\boldsymbol{A}, \boldsymbol{E})$ and any $\sigma \in \mathbb{C}$ that is not an eigenvalue of $(\boldsymbol{A}, \boldsymbol{E})$ it is shown in [20] that $(\boldsymbol{A}, \boldsymbol{E})$-invariance is equivalent to $\mathbf{H}$-invariance for

$$\mathbf{H} = (\boldsymbol{A} - \sigma \boldsymbol{E})^{-1} \boldsymbol{E}, \tag{16}$$

which happens to be the shift-invert operator defined by (9).

Then $\mathbf{H}(\sigma) = (\boldsymbol{A} - \sigma \boldsymbol{E})^{-1} \boldsymbol{E}$ has the same eigenvectors (regardless of σ) as the pencil $(\boldsymbol{A}, \boldsymbol{E})$. An eigenvalue λ of $\mathbf{H}$ and its corresponding eigenvalue μ of $(\boldsymbol{A}, \boldsymbol{E})$ are related by

$$\lambda = \frac{1}{\mu - \sigma}, \qquad \mu = \sigma + \frac{1}{\lambda} \tag{17}$$

and they share the same eigenvector. That is,

$$\begin{aligned} \mathbf{H}z &= \lambda z \\ &= \left(\frac{1}{\mu - \sigma} \right) z \end{aligned} \quad \Longleftrightarrow \quad \begin{aligned} \boldsymbol{A}z &= \mu \boldsymbol{E} z \\ &= \left(\sigma + \frac{1}{\lambda} \right) \boldsymbol{E} z \end{aligned}$$

To see this, observe that

$$\begin{aligned}(\mu\boldsymbol{E}-\boldsymbol{A}) &= [(\mu-\sigma)\boldsymbol{E}+(\sigma\boldsymbol{E}-\boldsymbol{A})] \\ &= [(\mu-\sigma)\underbrace{(\sigma\boldsymbol{E}-\boldsymbol{A})^{-1}\boldsymbol{E}}_{-\mathbf{H}}+I] \\ &= [(\mu-\sigma)\mathbf{H}-I] \\ &= \left[\mathbf{H}-\left(\frac{1}{\mu-\sigma}\right)I\right] \\ &= (\mathbf{H}-\lambda I).\end{aligned} \tag{18}$$

(18) shows that invariant-subspaces under the shifted operator $\mathbf{H}(\sigma)$ do not depend on σ. This is useful because if we decide to change the shift σ, any previously discovered invariant-subspace will be still be invariant under the new operator.

2.2.4 Pole-Residue representation

Recall the transfer-function

$$\mathcal{H}(s) = \boldsymbol{C}^T(s\boldsymbol{E}-\boldsymbol{A})^{-1}\boldsymbol{B}, \tag{5}$$

which includes the linear matrix pencil $(\boldsymbol{A},\boldsymbol{E})$. Poles of the transfer-function (5) are values $\mu \in \mathbb{C}\cup\infty$ such that $\|\mathcal{H}(\mu)\| = \infty$. Poles of $\mathcal{H}(s)$ are eigenvalues of the matrix pencil $(\boldsymbol{A},\boldsymbol{E})$, but their significance is determined by $\boldsymbol{B}$ and $\boldsymbol{C}$.

Figure 2.2 illustrates influence of poles and zeros on the transfer-function.

Poles and residues of the implicit (shifted) transfer-function

Consider the so-called σ-shifted transfer-function

$$\mathcal{H}(s) = \boldsymbol{C}^T(I-(s-\sigma)\mathbf{H})^{-1}\mathbf{R}, \tag{8}$$

rather than the standard formulation (5).

Assume that $\mathbf{H}$ is diagonalizable with right-eigenbasis Z, so that

$$\mathbf{H}Z = Z\Lambda$$

for a $N \times N$ diagonal matrix $\Lambda = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_N)$ of eigenvalues.

Then

$$\begin{aligned}\mathcal{H}(s) &= \boldsymbol{C}^T(I - (s-\sigma)\mathbf{H})^{-1}\mathbf{R} \\ &= \boldsymbol{C}^T\left(Z(I - (s-\sigma)\Lambda)Z^{-1}\right)^{-1}\mathbf{R} \\ &= (\boldsymbol{C}^T Z)\Delta(s)(Z^{-1}\mathbf{R}) \end{aligned} \tag{19}$$

where $\Delta(s) = (I - (s-\sigma)\Lambda)^{-1}$ is a diagonal matrix with diagonal entries $\delta_j(s) = 1 - (s-\sigma)\lambda_j$, or equivalently

$$\delta_j(s) = \begin{cases} \dfrac{\sigma - \mu_j}{s - \mu_j}, & \mu_j \neq \infty \\ 1, & \mu_j = \infty. \end{cases} \tag{20}$$

Then for

$$\boldsymbol{C}^T Z = \begin{bmatrix} \hat{f}_1 & \hat{f}_2 & \cdots & \hat{f}_N \end{bmatrix} \quad \text{and} \quad (Z^{-1}\mathbf{R})^T = \begin{bmatrix} \hat{g}_1 & \hat{g}_2 & \cdots & \hat{g}_N \end{bmatrix},$$

we have

$$\begin{aligned}\mathcal{H}(s) &= \sum_j \frac{\hat{f}_j \hat{g}_j^T}{1 - (s-\sigma)\lambda_j} \qquad (21) \\ &= \sum_{\lambda_j = 0} \hat{f}_j \hat{g}_j^T + \sum_{\lambda_j \neq 0} \frac{\sigma - \mu_j}{s - \mu_j} \hat{f}_j \hat{g}_j^T \\ &= \sum_j \delta_j(s) \hat{f}_j \hat{g}_j^T, \qquad (22) \end{aligned}$$

where $\delta_j(s)$ is from (20). The transpose $\hat{g}_j^T$ is in fact a standard (not-conjugated) transpose, even if $\hat{g}_j$ is complex-valued. Both $\hat{f}_j$ and $\hat{g}_j^T$ are scalars in the SISO case. Note that a zero eigenvalue $\lambda_j = 0$ of $\mathbf{H}$ corresponds to an infinite $\mu_j = \infty$ pole (eigenvalue of $(\boldsymbol{A}, \boldsymbol{E})$).

2.2.5 Pole-weight

Poles of the transfer-function $\mathcal{H}(s) = \boldsymbol{C}^T (s\boldsymbol{E} - \boldsymbol{A})^{-1} \boldsymbol{B}$ are values $\mu \in \mathbb{C} \cup \infty$ such that $\|\mathcal{H}(\mu)\| = \infty$. Poles of $\mathcal{H}(s)$ are eigenvalues of the matrix pencil $(\boldsymbol{A}, \boldsymbol{E})$, but their significance is determined by $\boldsymbol{B}$ and $\boldsymbol{C}$. Pole dominance is a notion of a pole's influence on the transfer-function frequency

response $\mathcal{H}(i\omega)$ on an interval of the $\Im$-axis (or all of it). Pole-residue formulation (22) suggests a hierarchy of poles' importance for approximation.

We define a measure of pole-dominance, which we call its *weight* with respect to the frequency response domain $i[\omega_1, \omega_2] \subset \mathbb{C}$. It is similar to the modal dominance index (MDI) of [1], but it considers a pole's influence over the frequency response domain rather than all of the positive $\Im$-axis, and it does not blow-up for poles on or near the $\Im$-axis. Basically we add 1 to the denominator.

If we take the norm of (22) over the interval $i[\omega_1, \omega_2]$ of interest on the $\Im$-axis, we have

$$\|\mathcal{H}(i\omega)\| \leq \sum_j \|\delta_j(i\omega)\| \, \|\hat{f}_j\| \, \|\hat{g}_j\| \,, \tag{23}$$

which is a sum of positive numbers, each one associated with a pole μ_j. We call this positive number the weight of the pole. A relatively large pole-weight

$$\gamma_j = \|\delta_j(i\omega)\| \, \|\hat{f}_j\| \, \|\hat{g}_j\| \tag{24}$$

indicates that μ_j is a so-called dominant pole.

The scalar-valued function $\delta_j(i\omega)$ represents the influence of pole μ_j on the system frequency response via its proximity to the segment $i[\omega_1, \omega_2]$ of interest, and we take its maximum value

$$\begin{aligned} \|\delta_j(i\omega)\|_\infty &= \max_{\omega \in [\omega_1, \omega_2]} |\delta_j(i\omega)| \\ &= \begin{cases} \dfrac{|\sigma - \mu_j|}{1 + \min\{|\mu_j - \omega_1|, |\mu_j - \omega_2|, |\Re(\mu_j)|\}}, & \mu_j \neq \infty \\ 1, & \mu_j = \infty \end{cases} \end{aligned} \tag{25}$$

over that interval as a sense of its over-all influence on the transfer-function in that region. The value $\min\{|\mu_j - \omega_1|, |\mu_j - \omega_2|, |\Re(\mu_j)|\}$ is merely the distance of μ_j to the segment $i[\omega_1, \omega_2]$, as illustrated in figure 2.3. Conjugate pairs must be considered together, so when determining weight we actually consider $\Re(\mu_j) + i|\Im(\mu_j)|$, rather than μ_j. That way, each member of the pair gets assigned the same weight.

Total system-weight

The right-hand-side of (23) (i.e. $\sum_j \gamma_j$), is the total weight of the system (2) with respect to $i[\omega_1, \omega_2]$, and we attempted to use it as a measure of ROM convergence. In numerical experiments we found that the the combined weight of a few dominant poles often comprises most of a system's total weight. A plotted example of that is in figure 5.5

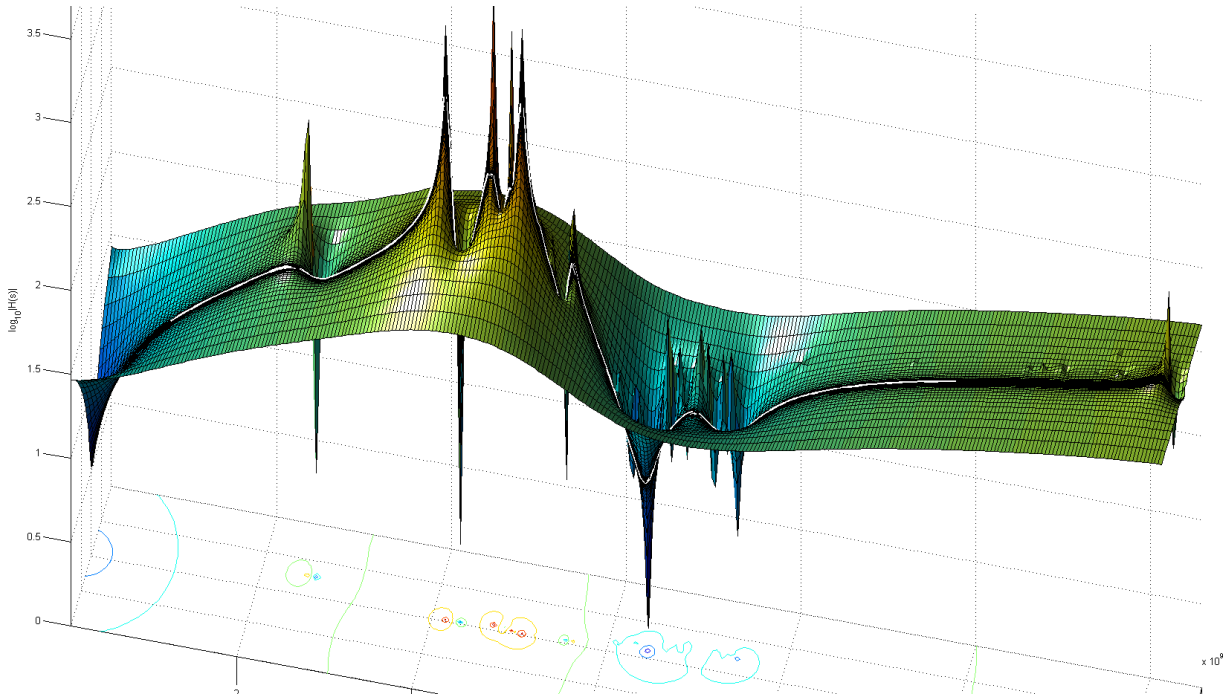


Figure 2.2: Transfer function gain $|\mathcal{H}(s)|$ of example system `ex1481s11`, plotted over a rectangular region $(-10^9, 10^9) \times i(10^8, 10^{10.1})$ of the complex-plane, with linear scaling for the axes. The frequency response (also shown in figure 2.1a) is highlighted over $i(10^9, 10^{10})$ in white, and the contour plot below indicates the dominant poles and zeros near the $\Im$-axis. Observe the influence of poles and zeros on the frequency response (the white curve). Poles with large residues appear more massive, like large splashes on a pond. We also see large poles and zeros that appear to be separating into several smaller ones. Proximity to the $\Im$-axis determines whether a pole or zero causes a sharp peak in gain or has a gentler influence. This transfer function plot is actually an $n = 300$ ROM of a model whose original size is $N = 1841$. For this model, $n = 300$, by virtually any method, is large enough to be a very good approximation. Indeed, we rely on model reduction to make such a plot, as computing it with the original model would take too long to be practical with available computers. It involves solving a 300×300 system $\mathcal{H}(s) = \boldsymbol{C}^T(\boldsymbol{A} - s\boldsymbol{E})^{-1}\boldsymbol{B}$ for every value of s, rather than an 1841×1841 system.

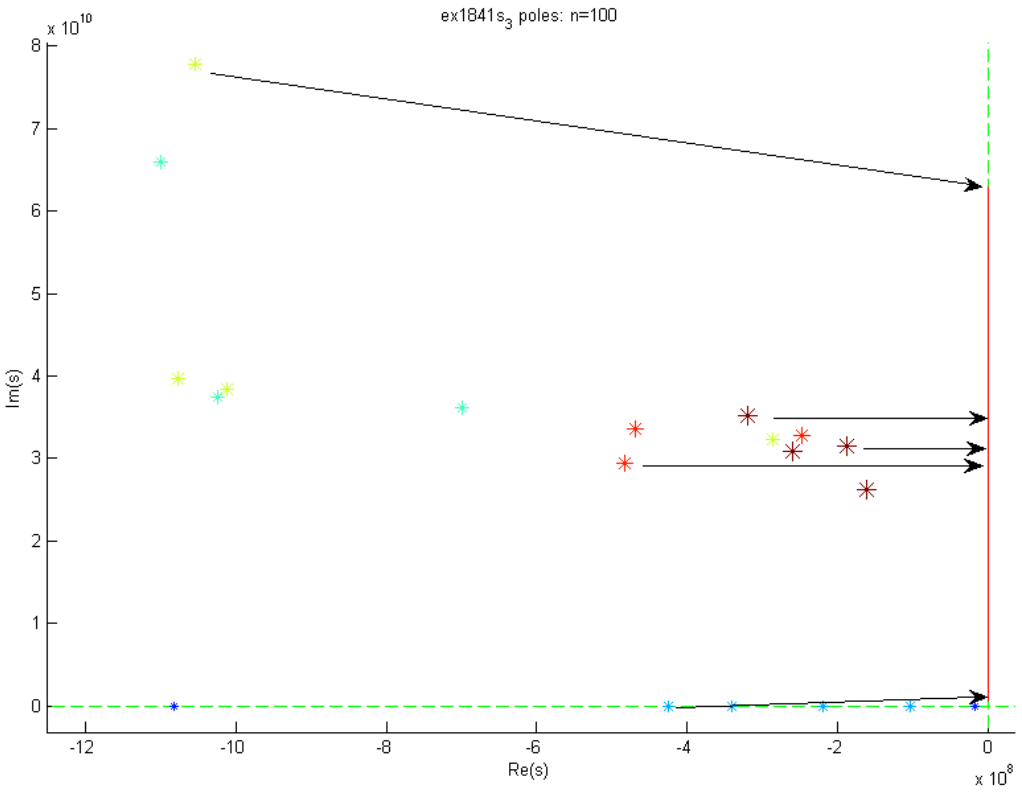


Figure 2.3: This plot illustrates minimum distance $\min\{|\mu_j - \omega_1|, |\mu_j - \omega_2|, |\Re(\mu_j)|\}$ to the "segment of interest" that we use to compute (25). The red (solid) segment on the $\Im$-axis is the segment $i(\omega_0, \omega_1)$ "of interest". The arrows indicate how we define a pole's distance to the segment, which we use to determine the pole's weight. Conjugate pairs must be considered together, so when determining weight we always consider $\Re(\mu) + i|\Im(\mu)|$, rather than μ. That way, each member of the pair gets assigned the same weight.

2.3 Reduced-order transfer-function via projection

The URM (unreduced-model) transfer-function (5) and its shifted variant (8) are mathematically equivalent. For large applications however, we do not expect to use either formulation in its unreduced form, but rather a suitable reduced-order model. Projection onto a subspace with basis V implies two different ROM transfer-functions that are not equal in general. The current literature does not seem to have different names for these two formulations so we will name them here. First there is the ROM transfer function that we define via *implicit* projection onto V. The implicitly-projected ROM formulation was introduced as part of the PVL (Padé via Lanczos) method [15] for model-reduction in 1995. The second projected-ROM transfer-function formulation is called *explicitly* projected onto V, and was introduced as the result of PRIMA (Passive Reduced-order Interconnect Macromodeling Algorithm) [32] introduced a few years later. Both methods use Krylov subspaces for projection.

Let $V \in \mathbb{C}^{N\times n}$ be a matrix with orthogonal columns that form a basis of our projection subspace. If we make the orthogonal projections

$$\boldsymbol{A}_n := V^T \boldsymbol{A} V, \quad \boldsymbol{E}_n := V^T \boldsymbol{E} V, \quad \boldsymbol{C}_n := V^T \boldsymbol{C}, \quad \boldsymbol{B}_n := V^T \boldsymbol{B}, \tag{4}$$

of realization $(\boldsymbol{A}, \boldsymbol{E}, \boldsymbol{B}, \boldsymbol{C})$,[2] the **explicitly-projected** model $(\boldsymbol{A}_n, \boldsymbol{E}_n, \boldsymbol{B}_n, \boldsymbol{C}_n)$ has transfer-function

$$\widehat{\mathcal{H}}_n(s) = \boldsymbol{C}_n^T \left(s\boldsymbol{E}_n - \boldsymbol{A}_n\right)^{-1} \boldsymbol{B}_n \tag{26}$$

The explicitly-projected ROM has the property that it is stable if the projection basis V is real-valued, which is not true in general of the implicit projected transfer-function, described next.

Krylov-subspace methods use the operator $\mathbf{H}$ and operand $\mathbf{R}$ from (9) to construct a basis V and a projected operator matrix $\widetilde{\mathbf{H}} \in \mathbb{C}^{n\times n}$ such that

$$\widetilde{\mathbf{H}} = V^T \mathbf{H} V. \tag{27}$$

This permits what we call the ROM transfer-function via implicit-projection, or implicit transfer-

[2] assuming the matrix pencil $(\boldsymbol{A}_n, \boldsymbol{E}_n)$ is regular

function

$$\widetilde{\mathcal{H}}_n(s) = \boldsymbol{C}_n^H \left(I - (s-\sigma)\widetilde{\mathbf{H}}\right)^{-1} \widetilde{\boldsymbol{\rho}}_n, \tag{28}$$

where

$$\boldsymbol{C}_n := W^H \boldsymbol{C} \quad \text{and} \quad \widetilde{\boldsymbol{\rho}}_n := V^T \mathbf{R}.$$

The ROM transfer-function (28) is analogous to the "shifted" transfer-function (8), only now the shift $\sigma \in \mathbb{C}$ is the interpolation point about which (28) approximates (8).

We say (28) is a transfer-function via implicit-projection because it exists without a projected realization $(\boldsymbol{A}_n, \boldsymbol{E}_n, \boldsymbol{B}_n, \boldsymbol{C}_n)$ from (4). The reduced model implied by (28) is actually

$$\begin{aligned} \widetilde{\mathbf{H}} \frac{d\tilde{x}}{dt} &= (I + \sigma \widetilde{\mathbf{H}})\tilde{x} + \widetilde{\boldsymbol{\rho}}_n u \\ \hat{y} &= \boldsymbol{C}_n^T \tilde{x}, \end{aligned} \tag{29}$$

which is not equivalent to the explicitly-projected system

$$\begin{aligned} \boldsymbol{E}_n \frac{dz}{dt} &= \boldsymbol{A}_n z + \boldsymbol{B}_n u \\ \hat{y} &= \boldsymbol{C}_n^T z, \end{aligned} \tag{3}$$

given the same basis V, unless V spans $\mathbb{C}^N$ (i.e. $n = N$) in which case they are both equivalent to the original system (2).

Even if the original system is passive and/or stable, the implicitly projected ROM is not necessarily passive or stable which makes it less suitable as a ROM. Some efforts [43, 24, 26] have been made to remedy this situation, but these methods tend to sacrifice moment-matching properties in the process.

Chapter 3

Order-reduction via Krylov-subspace projection

3.1 Krylov-subspace projection methods

Recall two ways to express the system transfer-function $\mathcal{H} : \mathbb{C} \to \mathbb{C}^{m \times p}$

$$\mathcal{H}(s) = \boldsymbol{C}^T \left(s\boldsymbol{E} - \boldsymbol{A}\right)^{-1} \boldsymbol{B}, \tag{5}$$

and

$$\mathcal{H}(s) = \boldsymbol{C}^T \left(I - (s - \sigma)\mathbf{H}\right)^{-1} \mathbf{R} \tag{8}$$

with $\mathbf{H} = (\boldsymbol{A} - \sigma\boldsymbol{E})^{-1}\boldsymbol{E}$ and $\mathbf{R} = (\sigma\boldsymbol{E} - \boldsymbol{A})^{-1}\boldsymbol{B}$.

(5) is the standard formulation and (8) is the so-called σ-shifted formulation, and $\mathbf{H}$ is sometimes called a shift-inverse or σ-shifted operator.

3.1.1 The Krylov subspace

Krylov-subspace projection methods are developed out of the power iteration with $\mathbf{H}$ acting on $\mathbf{R} = \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \cdots & \mathbf{r}_m \end{bmatrix}$. The goal of subspace projection-based model order-reduction is to obtain a basis V of a subspace $\mathcal{K}$ of $\mathbb{C}^N$ on which to project our system realization in order to obtain a reduced model. The Krylov-subspace is the ideal subspace to use for projection because reduced-

order models obtained via Krylov-subspaces have moment-matching properties (§2.2.1). Reduced-order models obtained by projection onto non-Krylov-subspaces do not have moment-matching properties in general. The *n-th Krylov-subspace* induced by $\mathbf{H}$ and a vector $\mathbf{r}$ is

$$\mathcal{K}_n(\mathbf{H}, \mathbf{r}) = \text{span}\left\{\mathbf{r}, \mathbf{H}\mathbf{r}, \mathbf{H}^2\mathbf{r}, \ldots, \mathbf{H}^{n-1}\mathbf{r}\right\}. \tag{30}$$

For the n-th *block* Krylov-subspace

$$\mathcal{K}_n(\mathbf{H}, \mathbf{R}) = \text{span}\left\{\mathbf{R}, \mathbf{H}\mathbf{R}, \mathbf{H}^2\mathbf{R}, \ldots \mathbf{H}^{\eta}\mathbf{R}\right\} \tag{31}$$

n is the dimension of the subspace, not the number η of powers of $\mathbf{H}$ that are involved.

Krylov-subspace projection methods for MOR come out of methods for finding eigenvalues, so we will first address general projection methods and Krylov projection methods for eigensolving, then delve into their application to MOR.

3.2 The Arnoldi process

An n-iteration cycle of Arnoldi's algorithm constructs a basis V for the Krylov subspace $\mathcal{K}_n(\mathbf{H}, \mathbf{r})$, and the projected operator $\widetilde{\mathbf{H}} = V^T\mathbf{H}V$, a strictly upper-Hessenberg matrix.

Algorithm 1: ARNOLDI

Input: $\mathbf{r} \in \mathbb{C}^N$, $\mathbf{H} \in \mathbb{C}^{N\times N}$ (or some way to compute $\mathbf{H}v$ for $v \in \mathbb{C}^N$)
Output: orthonormal $V \in \mathbb{C}^{N\times n}$, Upper Hessenberg $\widetilde{\mathbf{H}} \in \mathbb{C}^{n\times n}$ where span $V = \mathcal{K}_n(\mathbf{H}, \mathbf{r})$, and $\widetilde{\mathbf{H}} = V^T\mathbf{H}V$

```
r_0 := r
v_1 := r_0 / ||r_0||_2
for k = 1 to n do
    r_k := H v_k
    for j = 1 to k do      % Make r_k orthogonal to previous {v_1, v_2..., v_k}
        h_jk := v_k^H r_k
        r_k := r_k − h_jk v_j
    if ||r_k||_2 ≠ 0 then
        h_{j+1,j} := ||r_k||_2
        v_{k+1} := r_k / ||r_k||_2
    else exit k-loop
return V = [v_1  v_2  ···  v_n], v̂_n = v_{n+1} h_{n+1,n}, H̃ = [h_ij]
```

The Arnoldi process (Algorithm 1) basically performs a power iteration and orthogonalizes each iterate against previous ones, thus producing an orthonormal basis for (30). The most costly part of the algorithm is the matrix-vector product (line 4), followed by the orthogonalization part (lines 5-7). Algorithm 1 uses Modified Gram-Schmidt for orthogonalization but there are variants of the Arnoldi process that use other orthogonalization methods. A notable alternative uses Householder reflectors making for a more stable and more costly method.

3.2.1 Complexity of Arnoldi (with MGS orthogonalization)

Take an n-iteration Arnoldi cycle with a general matrix $\mathbf{H}$: there are n matrix-vector products $\mathbf{H}v_k$ (line 4), each requiring N^2 scalar multiplications (flops). With Modified Gram-Schmidt (MGS) as the orthogonalization process, we have $1 + 2 + \cdots + n = n(n+1)/2$ inner-products (line 6) and an equal number of AXPYs[1] (line 7), each requiring N flops. Note that the k-th step of Arnoldi requires kN flops for orthogonalization. The process takes longer for each iterate, eventually grinding to a crawl if N is large. The total cost of an n-iteration cycle of Arnoldi method is roughly $nN^2 + n^2N$ flops: nN^2 flops for matrix-vector products (sometimes called matvecs), and n^2N flops for orthogonalization. For large N the computational cost of Arnoldi is dominated by matvecs. It should be noted that the σ-shifted inverse operator $\mathbf{H} = (\boldsymbol{A} - \sigma\boldsymbol{E})^{-1}\boldsymbol{E}$ used for model reduction is not a general, dense matrix. The "matrix-vector product"

$$\mathbf{H}v = Q\left[U^{-1}L^{-1}\boldsymbol{B}(Pv)\right]$$

is actually implemented as a pair of sparse triangular solves requiring at most

$$2\,\mathrm{nnz}(U) + 2\,\mathrm{nnz}(L) \le 2N(N+1)$$

flops, where $\mathrm{nnz}(T) \le N(N+1)/2$ is the number of nonzero entries of an $N \times N$ triangular matrix T. The computation of these "matvecs" is still $\mathcal{O}\left(N^2\right)$ so we may view the operation as a matrix-vector product, as long as we consider the one-time cost of sparse $LU = P\mathbf{H}Q$ factorization.

The Arnoldi algorithm requires $(n+1)N$ units of storage for the basis vectors $v_j \in V$, which is

[1] $\alpha X + Y$ operations where α is a scalar and X and Y are vectors.

also an issue in large applications.

We consider the ROM size n to be negligible in comparison to the order N of the full model. Computation and storage cost are major issues when N is large. For a model of size n we have no choice but to compute n applications of $\mathbf{H}$ to a vector in $\mathbb{C}^N$, each costing $\mathcal{O}\left(N^2\right)$. Restarted Krylov methods attempt to make the process more computationally manageable by reducing the amount of orthogonalization. Since latter iterations require the most computation, the idea is to start over at a certain point.

3.2.2 ROM size vs. construction cost

Ultimately the goal of model order-reduction is to produce a small, accurate model (we would like to minimize n and model approximation error[2]), but the time taken to construct the model needs to be considered as well. In some applications, once a ROM is constructed it gets used repeatedly for several computations. Consider the case where producing the model (or several models) is itself the major expense. We may, for example, only need to solve the system once and the original system of order N is just too large to solve. Maybe a new ROM needs to be generated at every step in some sequence. ROM construction efficiency is where Krylov methods excel. This distinction is important because it sets the context when discussing the best model reduction method for a particular application.

3.2.3 The Arnoldi relation

An n-step cycle of the Arnoldi process with $\mathbf{H}$ and $\mathbf{r}$ yields the so-called Arnoldi relation

$$\begin{aligned}\mathbf{H}V &= \begin{bmatrix} V & v_{n+1} \end{bmatrix} \begin{bmatrix} \longleftarrow & \widetilde{\mathbf{H}} & \longrightarrow \\ 0 & \cdots & h_{n+1,n} \end{bmatrix} \\ &= V\widetilde{\mathbf{H}} + \ h_{n+1,n} v_{n+1} e_n^T \\ &= V\widetilde{\mathbf{H}} + \ \hat{v}_n e_n^T \end{aligned} \tag{32}$$

[2] one measure of this is $\|\mathcal{H} - \mathcal{H}_n\|$ in some norm.

where $V \in \mathbb{C}^{N\times n}$ is the orthogonal basis matrix for $\mathcal{K}_n(\mathbf{H}, \mathbf{r})$, starting with $v_1 = \mathbf{r}/\left\|\mathbf{r}\right\|_2$, and the upper Hessenberg matrix $\widetilde{\mathbf{H}} \in \mathbb{C}^{n\times n}$ is the Petrov-Galerkin projection of $\mathbf{H}$ on to that space (also known as the Arnoldi matrix), and can be considered a reduced-order spectral approximation to $\mathbf{H}$, because eigenvalues of $\widetilde{\mathbf{H}}$ approximate those of $\mathbf{H}$. The largest eigenvalues of $\widetilde{\mathbf{H}}$ are the most accurate, a property inherited from the power iteration.

3.2.4 Approximate eigenvalues from Arnoldi relation

Compared to $\mathbf{H}$, the matrix $\widetilde{\mathbf{H}}$ is small enough for its eigenvalue decomposition

$$\widetilde{\mathbf{H}}W = W\Lambda$$

to be computed cheaply.

Ritz-values (eigenvalues λ of $\widetilde{\mathbf{H}}$) are approximate eigenvalues of the large operator $\mathbf{H}$, and long[3] Ritz-vectors $Vw \in \mathcal{K}_n(\mathbf{H}, \mathbf{r})$ are the associated approximate eigenvectors of $\mathbf{H}$.

Left multiplying (32) with W yields

$$\begin{aligned}\mathbf{H}VW &= V\widetilde{\mathbf{H}}W + \hat{v}_n e_n^T W \\ &= VW\Lambda + \hat{v}_n e_n^T W\end{aligned} \tag{33}$$

$$\mathbf{H}Z = Z\Lambda + \hat{v}_n e_n^T W \tag{34}$$

so for $z_j = Vw_j$,

$$\begin{aligned}\mathbf{H}z_j &= \lambda z_j + \xi_j \hat{v}_n \\ &= \lambda z_j + \xi_j h_{n+1,n} v_{n+1}\end{aligned}$$

where $\xi_j = (e_n^T W)_j = W_{nj} \in \mathbb{C}$ is the j-th entry of the bottom (n-th) row of W. Here we see that every Ritz residual-vector

$$\mathbf{H}z_j - \lambda_j z_j = \xi_j \hat{v}_n$$

given by (34) is a scalar multiple of the residual-vector $\hat{v}_n$. Assuming $\left\|z_j\right\|_2 = 1$, the Arnoldi-relation

[3] vectors $w \in W$ are sometimes called short Ritz-vectors.

(32) thus implies a simple formulation

$$\mathrm{rr}_j = \frac{\|\mathbf{H}z_j - \lambda_j z_j\|_2}{\|\lambda_j z_j\|_2} = \frac{|\xi_j|}{|\lambda_j|}\,\|\hat{v}_n\|_2 = \frac{|\xi_j|}{|\lambda_j|}|h_{n+1,n}| \tag{35}$$

for the relative residual-errors of the Ritz-values/vectors. Ritz-pairs with low associated relative residual norms (35) are good approximations to eigenvalues/vectors of $\mathbf{H}$ if they are well-conditioned. This is because (35) actually indicates that (λ_j, z_j) is an exact eigen-pair of the perturbed matrix $\mathbf{H} + \mathcal{E}$ where the norm of the perturbation $\|\mathcal{E}\| = \|\hat{v}_n\|$. Rearrangement of the Arnoldi relation (32) reveals

$$\mathbf{H}V - \hat{v}_n e_n^T = (\mathbf{H} - \hat{v}_n v_n^T)V = V\widetilde{\mathbf{H}}.$$

If an eigenvalue of $\mathbf{H}$ is highly sensitive to perturbation (is badly conditioned) then a low or zero residual-norm (35) could be misleading. We can avoid this situation by noting that the largest eigenvalues of $\mathbf{H}$ converge first. The relative residual errors are likely to be accurate for Ritz values of largest magnitude, which we expect to converge first. A very small eigenvalue of $\mathbf{H}$ with small relative-residual error may be suspect.

3.3 Implicit vs. explicit Ritz-values and vectors

We motivate this discussion by recalling that poles of the URM transfer-function (5) are eigenvalues of $(\boldsymbol{A}, \boldsymbol{E})$.

It should be noted that although every eigenvalue λ of $\mathbf{H} = (\boldsymbol{A} - \sigma\boldsymbol{E})^{-1}\boldsymbol{E}$ corresponds to a μ of $(\boldsymbol{A}, \boldsymbol{E})$, and are related by $\lambda = 1/(\mu - \sigma)$ and share common eigenvectors, the same cannot be said for approximate eigenvalues $\hat{\lambda}$ of $\widetilde{\mathbf{H}} = V^T\mathbf{H}V$ and $\hat{\mu}$ of $(\boldsymbol{A}_n, \boldsymbol{E}_n) = (V^T\boldsymbol{A}V, V^T\boldsymbol{E}V)$. This means there are two different sets of approximate poles implied by the two ROM transfer functions defined in §2.3.

There are two different sets of approximations to the spectrum of $(\boldsymbol{A}, \boldsymbol{E})$, both implied by projection on to $\mathcal{K}_n(\boldsymbol{A}, \mathbf{R})$ via basis V:

Implicit Ritz-values The set of implicit Ritz-values

$$\left\{ 1/\hat{\lambda} + \sigma \;\middle|\; \hat{\lambda} \in \Lambda(\widetilde{\mathbf{H}}) \right\} \tag{36}$$

of $(\boldsymbol{A}, \boldsymbol{E})$ with respect to $\mathcal{K}_n(\mathbf{H}, \mathbf{R})$ are determined by eigenvalues of the projected operator

$$\widetilde{\mathbf{H}} = V^T \mathbf{H} V = V^T (\boldsymbol{A} - \sigma \boldsymbol{E})^{-1} \boldsymbol{E} V,$$

which is a byproduct of constructing V by n steps of the Arnoldi algorithm. An implicit Ritz-pair $(\hat{\lambda}, z)$ is said to be converged if its residual-norm $\|\mathbf{H}z - \hat{\lambda}z\|$, which is of the same order as the residual $(\boldsymbol{A}z - (1/\hat{\lambda} + \sigma)\boldsymbol{E})$, is smaller than some tolerance.

Explicit Ritz-values The set of of explicit Ritz-values

$$\{\, \hat{\mu} \in \Lambda(\boldsymbol{A}_n, \boldsymbol{E}_n) \,\} \tag{37}$$

of $(\boldsymbol{A}, \boldsymbol{E})$ with respect to $\mathcal{K}_n(\mathbf{H}, \mathbf{R})$ where

$$(\boldsymbol{A}_n, \boldsymbol{E}_n) = (V^T \boldsymbol{A} V, V^T \boldsymbol{E} V),$$

is not implied by the Arnoldi process and must be computed.

(36) and (37) are not equal in general but are related in that they both converge to the same spectrum $\Lambda(\boldsymbol{A}, \boldsymbol{E})$. Note that both sets of approximate eigenvalues are dependent on the shift σ; we expect eigenvalue approximations closer to σ to be more accurate for both (36) and (37), because they both result from projection on to the Krylov-subspace $\mathcal{K}_n(\boldsymbol{A}, \mathbf{R})$, where $\mathbf{H} = \mathbf{H}(\sigma)$ and $\mathbf{R} = \mathbf{R}(\sigma)$.

The associated approximate eigen-*vectors* are not dependent on σ. Only the order in which they converge depends on σ. Eigenvectors associated with (36) and with (37) are not equal in general, but sufficiently converged vectors are nearly equal.

When converged, implicit and explicit eigen-pairs are nearly identical. We consider approximate eigenvalues/vectors coming from explicit and implicit computation to be interchangeable if they are near σ and have low relative residual-error norm (35). Thus, if an eigen-pair $(\hat{\lambda}_j, w_j)$ of $\widetilde{\mathbf{H}}$ is converged, then we can expect that $(\sigma + 1/\hat{\lambda}_j, V w_j)$ is a converged eigen-pair of $(\boldsymbol{A}_n, \boldsymbol{E}_n)$ with

about the same order of error of approximation to an eigen-pair of $(\boldsymbol{A}, \boldsymbol{E})$.

The reason we consider both sets of approximate eigenvalues/vectors is that implicit (36) Ritz-values/vectors are far cheaper to compute than the explicit variety (37), but the explicit formulation (3) is the end goal of explicit-projection-based MOR. Un-converged poles of the implicitly projected model transfer-function can and often do have positive real-part, which is unfavorable for ROM applications. These eigenvalue approximations all move to the left half of the complex plane as they converge to their final resting values, but as long as there are any implicit Ritz-values $1/\hat{\lambda} + \sigma$ with positive real part, the implicitly projected model (28) is possibly unstable and not attractive for model order-reduction.[4] Implicitly obtained eigen-information is useful feedback to gage and possibly direct progress of an adaptive method. Some MOR methods, typically called *restarted* methods including [24, 35, 26, 2], have been developed which attempt to purge subspace components associated with "bad" (destabilizing) or otherwise unwanted eigenvalues from the constructed basis V, but they destroy moment-matching properties and introduce other problems. Explicitly-projected eigenvalues $\hat{\mu}$ of $(\boldsymbol{A}_n, \boldsymbol{E}_n)$ are always (for any n) well-behaved as long as the projection basis V is real, and as long as $\mathcal{K}_n(\boldsymbol{A}(\sigma), \mathbf{R}(\sigma)) \subseteq \operatorname{span} V$, the explicitly-projected ROM on to V is guaranteed to be of matrix-Padé-type (match moments) with respect to σ.

3.4 Moment-matching property of Krylov-subspace projected ROM

In this section we offer proofs that a reduced-order model implied by orthogonal projection (via one orthonormal basis $V = \begin{bmatrix} v_1 & v_2 & \cdots & v_n \end{bmatrix}$) on to a Krylov-subspace matches l moments about σ, where l is the block-degree of the Krylov-subspace

$$\operatorname{span}\begin{bmatrix} v_1 & v_2 & \cdots & v_n \end{bmatrix} = \mathcal{K}_l(\mathbf{H}, \mathbf{R}) = \operatorname{span}\begin{bmatrix} \mathbf{R} & \mathbf{HR} & \mathbf{H}^2\mathbf{R} & \cdots & \mathbf{H}^{l-1}\mathbf{R} \end{bmatrix}.$$

We will show this for implicitly projected ROMs (28) in Theorem 1, and explicitly-projected ROMs (26) in Theorem 2. Significant differences in the two ROM approximations are present away from expansion-point(s) σ, but near σ they are approximations of the same order.

[4] Implicitly projected ROMs, such as those produced by PVL [15] often work fine in many practical applications despite being unstable, but they are currently unpopular.

Recall the URM (unreduced model) transfer-function $\mathcal{H}(s) = \boldsymbol{C}^T(\boldsymbol{A}-s\boldsymbol{E})^{-1}\boldsymbol{B}$ of LTI descriptor system (2), and its equivalent shift-invert formulation $\mathcal{H}(s) = \boldsymbol{C}^T\left(I-(s-\sigma)\mathbf{H}\right)^{-1}\mathbf{R}$ with shift σ, or

$$\mathcal{H}(s+\sigma) = \boldsymbol{C}^T\left(I-s\mathbf{H}\right)^{-1}\mathbf{R} \tag{38}$$

$$= \sum_{j=0}^{\infty} s^j \mathcal{H}^{(j)} \tag{6}$$

where (6) is the Taylor series expansion of $\mathcal{H}(s)$ about $\sigma \in \mathbb{C}$. The j-th moment $\mathcal{H}^{(j)}$ was shown in §2.2.1 to be

$$\mathcal{H}^{(j)} = \boldsymbol{C}^T\mathbf{H}^j\mathbf{R}. \tag{39}$$

3.4.1 Moment matching of the implicitly-projected ROM

The implicitly projected ROM transfer-function

$$\widetilde{\mathcal{H}}(s+\sigma) = \boldsymbol{C}_n^T\left(I-s\widetilde{\mathbf{H}}\right)^{-1}\widetilde{\boldsymbol{\rho}}_n \tag{40}$$

is defined via projection of (38), as

$$\widetilde{\mathbf{H}} = V^T\mathbf{H}V, \quad \boldsymbol{C}_n = V^T\boldsymbol{C}, \quad \widetilde{\boldsymbol{\rho}}_n = V^T\mathbf{R} \tag{41}$$

rather than by projecting the system realization $(\boldsymbol{A},\boldsymbol{E},\boldsymbol{B},\boldsymbol{C})$, hence its specification as the transfer-function for an *implicitly* projected model. Moments of (40) about σ are given as $\widetilde{\mathcal{H}}^{(j)} = \boldsymbol{C}_n^T\widetilde{\mathbf{H}}_n^j\widetilde{\boldsymbol{\rho}}_n$.

Theorem 1. *Suppose the span of an orthonormal basis $V \in \mathbb{R}^{N\times n}$ contains the Krylov-subspace $\mathcal{K}_l(\mathbf{H},\mathbf{R})$ of block-degree l for some $l \leq n$. Then moments of $\widetilde{\mathcal{H}}^{(j)}(\sigma)$ of the implicitly projected ROM transfer-function (28), (40) and moments $\mathcal{H}^{(j)}(\sigma)$ of the URM transfer-function (5) about σ are related by*

$$\widetilde{\mathcal{H}}^{(j)} = \boldsymbol{C}_n^T\widetilde{\mathbf{H}}^j\widetilde{\boldsymbol{\rho}}_n = \boldsymbol{C}^T\mathbf{H}^j\mathbf{R} = \mathcal{H}^{(j)} \tag{42}$$

for $j = 0, 1, \ldots, l-1$.

Proof. The theorem follows from left-applying $\boldsymbol{C}^T$ to

$$\mathbf{H}^j\mathbf{R} = V\widetilde{\mathbf{H}}^j\widetilde{\boldsymbol{\rho}}_n \tag{43}$$

for $j = 0, 1, \ldots, l-1$, which we will show by induction.

For $j = 0$, (43) follows from (41). Now assume (43) holds for some $j \in \{0, 1, \ldots, l-2\}$. Applying $\mathbf{H}$ to (43) yields

$$\begin{aligned}\mathbf{H}(\mathbf{H}^j\mathbf{R}) = \mathbf{H}^{j+1}\mathbf{R} = \mathbf{H}(V\widetilde{\mathbf{H}}^j\widetilde{\boldsymbol{\rho}}_n) \\ &= V\widetilde{\mathbf{H}}\widetilde{\mathbf{H}}^j\widetilde{\boldsymbol{\rho}}_n, \quad \text{since } \mathbf{H}V = V\widetilde{\mathbf{H}} \\ &= V\widetilde{\mathbf{H}}^{j+1}\widetilde{\boldsymbol{\rho}}_n.\end{aligned}$$

□

3.4.2 Moment matching of the explicitly-projected ROM

Proof of moment-matching for the explicitly-projected ROM (3) transfer-function is a little more involved than Theorem 1 for the implicitly projected model. It is included as Theorem 2. The proof is adapted from [16, proposition 6 and theorem 7].

Recall the explicitly-projected ROM (3) with transfer-function

$$\widehat{\mathcal{H}}_n(s) = \boldsymbol{C}_n^T\left(s\boldsymbol{E}_n - \boldsymbol{A}_n\right)^{-1}\boldsymbol{B}_n, \tag{26}$$

where the system realization $(\boldsymbol{A}, \boldsymbol{E}, \boldsymbol{B}, \boldsymbol{C})$ is said to be *explicitly* projected as

$$\boldsymbol{A}_n := V^T\boldsymbol{A}V, \quad \boldsymbol{E}_n := V^T\boldsymbol{E}V, \quad \boldsymbol{C}_n := V^T\boldsymbol{C}, \quad \boldsymbol{B}_n := V^T\boldsymbol{B}.$$

Moments of the ROM transfer-function (26) are

$$\widehat{\mathcal{H}}^{(j)} = \boldsymbol{C}_n^T\widehat{\mathbf{H}}^j\widehat{\boldsymbol{\rho}}_n,$$

where the structures

$$\widehat{\mathbf{H}} := \left(\boldsymbol{A}_n - \sigma\boldsymbol{E}_n\right)^{-1}\boldsymbol{E}_n \quad \text{and} \quad \widehat{\boldsymbol{\rho}}_n := \left(\sigma\boldsymbol{E}_n - \boldsymbol{A}_n\right)^{-1}\boldsymbol{B}_n. \tag{44}$$

are analogous to the shift-invert operator and start-block

$$\mathbf{H} := (\boldsymbol{A} - \sigma \boldsymbol{E})^{-1}\boldsymbol{E}, \quad \mathbf{R} := (\sigma \boldsymbol{E} - \boldsymbol{A})^{-1}\boldsymbol{B} \tag{9}$$

of the unreduced model (2). The proof of Theorem 1 depended on $\widetilde{\mathbf{H}} = V^T\mathbf{H}V$, which we do not have for the explicitly-projected ROM. In general, $\widehat{\mathbf{H}} \neq V^T\mathbf{H}V$. However, for an appropriate choice of F_n,

$$\widehat{\mathbf{H}} = V^T F_n V$$

implies that (26) matches l moments.

Theorem 2. *Suppose the span of an orthonormal basis $V \in \mathbb{R}^{N\times n}$ contains the Krylov-subspace $\mathcal{K}_l(\mathbf{H}, \mathbf{R})$ of block-degree l for some $l \leq n$, and let*

$$F_n := V(\boldsymbol{A}_n - \sigma \boldsymbol{E}_n)^{-1}V^T\boldsymbol{E}. \tag{45}$$

Then for $j \leq l \leq n$, the j-th moment $\widehat{\mathcal{H}}^{(j)}$ of the explicitly-projected ROM transfer-function (26) and moment $\mathcal{H}^{(j)}$ of the unreduced model (5) are related by

$$\widehat{\mathcal{H}}^{(j)} = \boldsymbol{C}^T F_n^i \mathbf{R} \quad \textit{for} \quad i = 0, 1, \ldots \tag{46}$$

$$= \mathcal{H}^{(i)} \qquad \textit{for} \quad i = 0, 1, \ldots, j-1. \tag{47}$$

Proof. First we show (46). Since $\operatorname{span}\mathbf{H}^i\mathbf{R} \subseteq \mathcal{K}_l(\mathbf{H}, \mathbf{R})$ for $i = 0, 1, \ldots, j$ and $\mathcal{K}_l(\mathbf{H}, \mathbf{R}) \subseteq \operatorname{span} V$, for each $i = 1, 2, \ldots, j$ there is a matrix X_i such that

$$\mathbf{H}^{i-1}\mathbf{R} = V X_i. \tag{48}$$

Recall that $\mathbf{R} = (\sigma \boldsymbol{E} - \boldsymbol{A})^{-1}\boldsymbol{B}$. Then for $i = 1$,

$$\boldsymbol{B} = (\sigma \boldsymbol{E} - \boldsymbol{A})\mathbf{R} = (\sigma \boldsymbol{E}V - \boldsymbol{A}V)X_1,$$

which when left-multiplied by V^T results in

$$V^T \boldsymbol{B} = V^T(\sigma \boldsymbol{E} V - \boldsymbol{A} V) X_1$$

$$\boldsymbol{B}_n = (\sigma \boldsymbol{E}_n - \boldsymbol{A}_n) X_1.$$

Then

$$X_1 = (\sigma \boldsymbol{E}_n - \boldsymbol{A}_n)^{-1} \boldsymbol{B}_n = \widehat{\boldsymbol{\rho}}_n. \tag{49}$$

Right-multiplying (45) with V gives $F_n V = V(\boldsymbol{A}_n - \sigma \boldsymbol{E}_n)^{-1} \boldsymbol{E}_n = V \widehat{\mathbf{H}}$, and by induction on i,

$$F_n^i V = V \widehat{\mathbf{H}}^i \quad \text{for} \quad i = 0, 1, \ldots. \tag{50}$$

Then moments of the ROM transfer-function

$$\begin{aligned} \widehat{\mathcal{H}}^{(i)} = \boldsymbol{C}_n^T \widehat{\mathbf{H}}^i \widehat{\boldsymbol{\rho}}_n &= \boldsymbol{C}^T V \widehat{\mathbf{H}}^i \widehat{\boldsymbol{\rho}}_n \\ &= \boldsymbol{C}^T (F_n^i V) X_1 \quad \text{by (49) and (50)} \\ &= \boldsymbol{C}^T F_n^i \mathbf{R} \qquad \text{by (48) with } i = 1, \end{aligned}$$

which is (46).

Proof of (47) is implied by

$$\mathbf{H}^j \mathbf{R} = F_n^j \mathbf{R} \quad \text{for} \quad i = 0, 1, \ldots, j-1 \tag{51}$$

which we show by induction on i. (51) is trivial for $i = 0$. Now assume (51) is satisfied for some $i \in \{0, 1, j-2\}$. We will show that

$$F_n^{i+1} \mathbf{R} = \mathbf{H}^{i+1} \mathbf{R}$$

as follows:

$$\big((\boldsymbol{A} - \sigma \boldsymbol{E})^{-1} \boldsymbol{E}\big)(F_n^i \mathbf{R}) = \mathbf{H}(\mathbf{H}^i \mathbf{R}) = \mathbf{H}^{i+1} \mathbf{R} = V X_{i+2} \tag{52}$$

where the rightmost expression follows from (48). Left-multiplying (52) with $V^T(\boldsymbol{A} - \sigma \boldsymbol{E})$ yields

$$(V^T \boldsymbol{E})(F_n^i \mathbf{R}) = (V^T(\boldsymbol{A} - \sigma \boldsymbol{E}) V) X_{i+2} = (\boldsymbol{A}_n - \sigma \boldsymbol{E}_n) X_{i+2}. \tag{53}$$

Then

$$
\begin{aligned}
F_n^{i+1}\mathbf{R} &= F_n(F_n^i\mathbf{R}) \\
&= \Big(V(\boldsymbol{A}_n - \sigma\boldsymbol{E}_n)^{-1}V^T\boldsymbol{E}\Big)(F_n^i\mathbf{R}) \\
&= V(\boldsymbol{A}_n - \sigma\boldsymbol{E}_n)^{-1}(V^T\boldsymbol{E})(F_n^i\mathbf{R}) \\
&= VX_{i+2}, \qquad \text{by (53)} \\
&= \mathbf{H}^{i+1}\mathbf{R}, \qquad \text{by (48)}
\end{aligned}
$$

which proves (51). Applying $\boldsymbol{C}^T$ yields (47), i.e.

$$
\widehat{\mathcal{H}}^{(j)} = \boldsymbol{C}^T F_n^j\mathbf{R} = \boldsymbol{C}^T\mathbf{H}^j\mathbf{R} = \mathcal{H}^{(j)}
$$

□

Chapter 4

Interpolation-point selection and resulting projection-bases

4.0.3 Eigenvalue convergence sequence

A Krylov-subspace method iteratively constructs a basis but we often think of the progression in terms of a sequence of converging eigenvalues of $\mathbf{H}$, starting with the largest. In our case $\mathbf{H}$ is a shift-and-invert operator, so large eigenvalues λ of $\mathbf{H}$ are eigenvalues

$$\mu = \sigma + 1/\lambda$$

of $(\boldsymbol{A}, \boldsymbol{E})$ which are closest to σ. Since eigenvalues μ of $(\boldsymbol{A}, \boldsymbol{E})$ are poles of the transfer-function $\mathcal{H}(s)$, we speak of "poles converging" as progress towards an accurate ROM. This is consistent with the notion of a Taylor series giving a better approximation near σ with each additional moment.

A fundamental feature (and drawback) of Krylov-subspace methods for model order-reduction is that for a given shift σ there is only one way for the method to progress: it is generally from poles μ closest to σ (i.e. smallest $|\mu - \sigma|$), to those farthest away. This presents a problem because often only a few dominant poles (§2.2.5) influence the transfer-function over the segment of interest $i[\omega_0, \omega_1]$ on the $\Im$-axis. For example, suppose poles $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ are dominant poles but are separated

(in distance from σ) by several insignificant poles, as in

$$|\boldsymbol{\mu}_1 - \sigma| > |\mu_3 - \sigma| > |\mu_4 - \sigma| > \cdots > |\mu_\ell - \sigma| > |\boldsymbol{\mu}_2 - \sigma|.$$

Then, using a straightforward Krylov process, after $\boldsymbol{\mu}_1$ converges, $\mu_3, \ldots, \mu_\ell$ must all converge before $\boldsymbol{\mu}_2$ does, and all of the associated vector information is added to the projection basis V, creating a larger than necessary model. One way around this is to change the shift σ after some number of iterations. It should also be noted that information about significant transfer-function *zeros* may be included with that of insignificant poles, so convergence of dominant poles is a somewhat dubious indicator of approximate model convergence. Ideally, both dominant pole and zero information should be considered and at the time of this writing there are no Krylov methods that do this.

4.1 Multiple point moment-matching

Theorems 1 and 2 can be extended to imply moment-matching about any number of expansion-points if the projection subspace contains the appropriate Krylov-subspaces. Much of the pioneering rational interpolation research, notably the rational-Lanczos method [18] (and [21]) for model order-reduction was done by Grimme in the mid and late 1990s. It is somewhat based on Ruhe's Rational-Krylov [40, 41] eigenvalue method and formalization. Of particular interest are [22] and [13], both of which discuss interpolation-point selection. We refer the reader to those sources for the details of point selection.

Suppose that for each $j = 1, 2, \ldots, \tau$ the Krylov-subspace

$$\mathcal{K}_j = \mathcal{K}_{n_j}(\mathbf{H}_j, \mathbf{R}_j) = \operatorname{span}\left[\mathbf{R}_j \quad \mathbf{H}_j\mathbf{R}_j \quad \mathbf{H}_j^2\mathbf{R}_j \quad \cdots \quad \mathbf{H}_j^{\eta_j-1}\mathbf{R}_j\right]$$

of dimension n_j (and block-degree η_j), induced by

$$\mathbf{H}_j := \mathbf{H}(\sigma_j) = (\boldsymbol{A} - \sigma_j\boldsymbol{E})^{-1}\boldsymbol{E} \quad \text{and} \quad \mathbf{R}_j := \mathbf{R}(\sigma_j) = (\sigma_j\boldsymbol{E} - \boldsymbol{A})^{-1}\boldsymbol{B}$$

is contained in the span of V, so that

$$\mathcal{K}_1 \cup \mathcal{K}_2 \cup \cdots \cup \mathcal{K}_\tau \subseteq \operatorname{span} V \tag{54}$$

Then the ROM implied by orthogonal projection on to span V matches l_j moments about interpolation-point σ_j for each $j = 1, 2, \ldots, \tau$.

4.1.1 Merging bases

There are several ways to produce a basis for the composite space (54). The naive method suggested by our previous discussion of single-point Krylov methods is to use τ consecutive runs of a basic Krylov method like Algorithm 1 (Arnoldi), each producing an orthonormal basis V_j for which

$$\operatorname{span} V_j = \mathcal{K}_j,$$

and then somehow putting the bases together into $V = \begin{bmatrix} v_1 & v_2 & \cdots & v_n \end{bmatrix}$, where

$$\operatorname{span} \begin{bmatrix} v_1 & v_2 & \cdots & v_n \end{bmatrix} = \operatorname{span} V_1 \cup \operatorname{span} V_2 \cup \cdots \cup \operatorname{span} V_\tau. \tag{55}$$

For the general application of rational-interpolation we assume that complex interpolation-points are used. As will be discussed in §4.2 we are typically required to split the basis vectors into $\Re$ and $\Im$ parts. For this reason it is not necessary for the individual bases V_j to be orthogonal to one-another, or even linearly-independent. However, an $\mathbf{H}_j$-invariant-subspace $\mathcal{Y}$ contained in $\mathcal{K}_j$ is also $(\boldsymbol{A}, \boldsymbol{E})$-invariant and thus has global significance.

The naive approach of producing bases for $\mathcal{K}_j$ separately and combining them in a post-processing step is inefficient because there is a significant degree of overlap between spaces. Recall that an invariant-subspace under $\mathbf{H}(\sigma)$ is independent of the expansion-point (shift) σ. Suppose $\mathbf{H}_1 = \mathbf{H}(\sigma_1)$ and $\mathbf{H}_2 = \mathbf{H}(\sigma_2)$. Then an invariant-subspace $\mathcal{Y} \subset \operatorname{span} V_1$ under $\mathbf{H}_1$ is also invariant under $\mathbf{H}_2$.

It would be wasteful to spend computational effort re-discovering $(\boldsymbol{A}, \boldsymbol{E})$-invariant-subspace while computing a basis for $\mathcal{K}_j$, if we already discovered it while constructing the basis V_{j-1} for $\mathcal{K}_{j-1}$. Traditional rational-interpolation methods for MOR such as rational-Lanczos [18] avoid this issue by doing full, Arnoldi-style orthogonalization of an iterate against every previous vector, generating one orthogonal basis V for (55). For complex-valued interpolation-points we will have to split the basis (55) anyway, thus losing any orthogonality.

4.1.2 Interpolation-point translation

Suppose we have an approximate eigen-pair (λ, z) of $\mathbf{H}_1 = \mathbf{H}(\sigma_1)$ so that

$$\mathbf{H}_1 z = \lambda z + r$$

and we would like to know its residual under another member $\mathbf{H}_2 = \mathbf{H}(\sigma_2)$ of the shift-invert operator family

$$\left\{\mathbf{H}(\sigma) = (\boldsymbol{A} - \sigma \boldsymbol{E})^{-1}\boldsymbol{E}\right\}. \tag{56}$$

Then

$$\mathbf{H}_2 z = T\mathbf{H}_1 z = \lambda T z + Tr, \tag{57}$$

where T is the transformation

$$\begin{aligned} T(\sigma_1, \sigma_2) &:= (\boldsymbol{A} - \sigma_2 \boldsymbol{E})^{-1}(\boldsymbol{A} - \sigma_1 \boldsymbol{E}) \\ &= (\sigma_2 - \sigma_1)\mathbf{H}_2 + I. \end{aligned} \tag{58}$$

Note that $T\mathbf{H}_1 = \mathbf{H}_2$ and $T\mathbf{R}_1 = \mathbf{R}_2$.

Proof of (58). The expression (58) comes from observing that

$$\tilde{v} = Tv = (\boldsymbol{A} - \sigma_2 \boldsymbol{E})^{-1}(\boldsymbol{A} - \sigma_1 \boldsymbol{E})v,$$

$$\begin{aligned} (\boldsymbol{A} - \sigma_2 \boldsymbol{E})\widetilde{v} &= (\boldsymbol{A} - \sigma_1 \boldsymbol{E})v \\ \boldsymbol{A}\tilde{v} - \sigma_2 \boldsymbol{E}\tilde{v} &= \boldsymbol{A}v - \sigma \boldsymbol{E}v \\ \boldsymbol{A}(\tilde{v} - v) - \sigma_2 \boldsymbol{E}(\tilde{v} - v) &= (\sigma_2 - \sigma_1)\boldsymbol{E}v \\ \tilde{v} - v &= (\sigma_2 - \sigma_1)(\boldsymbol{A} - \sigma_2 \boldsymbol{E})^{-1}\boldsymbol{E}v \\ &= (\sigma_2 - \sigma_1)\mathbf{H}_2 v. \end{aligned}$$

□

The translation transformation (58) must be invertible, with

$$\big(T(\sigma_1,\sigma_2)\big)^{-1} = T(\sigma_2,\sigma_1) = (\sigma_1-\sigma_2)\mathbf{H}_1 + I. \tag{59}$$

For easier notation let $\Delta = \sigma_2 - \sigma_1$, so that $T = T(\sigma_1,\sigma_2) = \Delta\mathbf{H}_2 + I$ and $T^{-1} = -\Delta\mathbf{H}_1 + I$. Then (58) and (59) imply that

$$(\Delta\mathbf{H}_2 + I)(-\Delta\mathbf{H}_1 + I) = (-\Delta\mathbf{H}_1 + I)(\Delta\mathbf{H}_2 + I) = I,$$

from which it follows that

$$\mathbf{H}_2\mathbf{H}_1 = \frac{\mathbf{H}_2 - \mathbf{H}_1}{\sigma_2 - \sigma_1} \tag{60}$$

and for $\sigma_2 \neq \sigma_1$,

$$\mathbf{H}_1\mathbf{H}_2 = \mathbf{H}_2\mathbf{H}_1. \tag{61}$$

(61) shows that the set (56) of operators, commutes. (60) implies that

$$\frac{d}{d\sigma}\mathbf{H}(\sigma) = (\mathbf{H}(\sigma))^2\,.$$

It might interest the reader to observe that the shift-invert transfer-function representation

$$\mathcal{H}(s) = \boldsymbol{C}^T\big(I - (s-\sigma)\mathbf{H}(\sigma)\big)^{-1}\mathbf{R}(\sigma), \tag{62}$$

defined about the interpolation-point σ but independent of σ, involves a transformation of the form (58), since

$$I - (s-\sigma)\mathbf{H} = (\sigma - s)\mathbf{H} + I = T(s,\sigma).$$

Then we may re-write (62) as

$$\begin{aligned}\mathcal{H}(s) &= \boldsymbol{C}^T T(\sigma,s)\mathbf{R}(\sigma)\\ &= \boldsymbol{C}^T\mathbf{R}(s)\\ &= \boldsymbol{C}^T(\boldsymbol{A} - s\boldsymbol{E})^{-1}\boldsymbol{B}.\end{aligned}$$

Now back to Ritz-residual translation. Recall from (57) that for eigen-pair (λ, z) of $\mathbf{H}_1$ we have

$$\mathbf{H}_2 z = T\mathbf{H}_1 z = \lambda T z + Tr$$

with the translation $T = T(\sigma_1, \sigma_2) = \Delta\mathbf{H}_2 + I$ from (58) and $\Delta = \sigma_2 - \sigma_1$.

Then

$$\begin{aligned}\mathbf{H}_2 z &= \lambda(\Delta\mathbf{H}_2 + I)z + Tr \\ &= \lambda z + \lambda\Delta\mathbf{H}_2 z + Tr,\end{aligned}$$

so

$$(1 - \lambda\Delta)\mathbf{H}_2 z = \lambda z + Tr. \tag{63}$$

Define the scaling factor

$$\zeta := \frac{1}{1 - \lambda\Delta} = \frac{\mu - \sigma_1}{\mu - \sigma_2}$$

where $\mu = 1/\lambda + \sigma_1$ is the approximate eigenvalue of $(\boldsymbol{A}, \boldsymbol{E})$ associated with λ. Then

$$\begin{aligned}\mathbf{H}_2 z - (\zeta\lambda)z &= \zeta Tr \\ &= \zeta(\Delta\mathbf{H}_2 + I)r\end{aligned} \tag{64}$$

4.2 Complex expansion-points

If a shift $\sigma_j \in \mathbb{C}$ is not strictly-real then the basis $V_j \in \mathbb{C}^{N \times n_j}$ of its associated Krylov-subspace is also complex. Although Grimme discusses interpolation point selection in some depth in [22], properties of a ROM obtained via projection with a complex basis have not been fully explored. They are generally avoided in part due to the extra computation and storage required for complex arithmetic.

It should be noted however that using $\sigma \in \mathbb{R}$ is only half as efficient as it appears to be and $\sigma \in \mathbb{C}$ with $\Re(\sigma) \neq 0$ is potentially twice as efficient as it appears. That is because the system pencil $(\boldsymbol{A}, \boldsymbol{E})$ is real and its complex eigenvalues are conjugate pairs. If $\sigma \in \mathbb{R}$, eigenvalues of $\mathbf{H} = (\boldsymbol{A} - \sigma\boldsymbol{E})^{-1}E$ must converge pairwise, so each conjugate-pair of converged vectors provide only one piece of complex spectral information. Eigenvalues of $\mathbf{H}$ for complex σ do not converge in pairs, but each converged eigenvalue λ implies that the pole $\mu = \sigma + 1/\lambda$ *and* its conjugate $\overline{\mu}$ is converged. For reasons discussed next we generally split a complex basis into $\Re$ and $\Im$ parts,

so there is not as much difference between using a real and general complex interpolation-point as there seems.

Real bases are preferred because the system (2) realization $(\boldsymbol{A}, \boldsymbol{E}, \boldsymbol{B}, \boldsymbol{C})$ consists of real matrices; explicit-projection with a real basis yields a ROM characterized by $(\boldsymbol{A}_n, \boldsymbol{E}_n, \boldsymbol{B}_n, \boldsymbol{C}_n)$ which is also real and thus retains desired properties of the original model. One such property is symmetry of the transfer-function about the $\Re$-axis.

The typical procedure to obtain a real basis for a complex Krylov-subspace is to split the n vector basis V into $2n$ real vectors $v_j^{\mathbf{r}} = \Re(v_j)$ and $v_j^{\mathbf{i}} = \Im(v_j)$, forming the so-called split-basis $V^* \in \mathbb{R}^{N\times 2n}$, which spans the so-called *split-subspace*[1]

$$\begin{aligned}
&\operatorname{span}\begin{bmatrix} v_1^{\mathbf{r}} & v_1^{\mathbf{i}} & v_2^{\mathbf{r}} & v_2^{\mathbf{i}} & \cdots & v_n^{\mathbf{r}} & v_n^{\mathbf{i}} \end{bmatrix} \\
&\quad = \operatorname{span} \mathcal{K}_\eta(\mathbf{H}, \mathbf{R}) \cup \mathcal{K}_\eta(\overline{\mathbf{H}}, \overline{\mathbf{R}}) \\
&\quad = \operatorname{span} \mathcal{K}_\eta(\mathbf{H}(\sigma), \mathbf{R}(\sigma)) \cup \mathcal{K}_\eta(\mathbf{H}(\overline{\sigma}), \mathbf{R}(\overline{\sigma})) \\
&\quad = \mathcal{K}_\eta(\mathbf{H}, \mathbf{R})^*
\end{aligned} \tag{65}$$

of dimension $\eta \leq 2n$. The basis for a standard Krylov-subspace may have complex vectors but its span is generally considered over $\mathbb{R}$. The split complex Krylov-subspace admits a real basis but its span is over $\mathbb{C}$, so it should still be considered a complex space that contains $\mathcal{K}_\eta(\mathbf{H}, \mathbf{R})$.

Notice that the split Krylov-subspace (65) is the union of two Krylov-subspaces with complex conjugate shifts σ and $\overline{\sigma}$ so we may consider a complex shift to be two shifts. Saad calls this idea "double-shifting" in [36], where it was first given significant analysis. Matching moments about a conjugate pair of points σ and $\overline{\sigma}$ is not is not advantageous so much as an unavoidable effect of requiring a real basis. Indeed, convergence of an eigenvalue $\mu \in \mathbb{C}$ of $(\boldsymbol{A}, \boldsymbol{E})$ with associated vector z is equivalent to convergence of the eigen-pair $(\overline{\mu}, \overline{z})$ of $(\boldsymbol{A}, \boldsymbol{E})$ as well. It would seem that the basis and the resulting ROM are potentially twice as large. This is true in theory, but in practice a complex quantity $z = \alpha + i\beta \in \mathbb{C}$ is represented by two real quantities $\alpha, \beta \in \mathbb{R}$ anyway. A complex ROM realization of order n is deceptively small because of the complex quantities involved. We avoid this ambiguity by always referring to the model size n as the number of vectors of the real

[1] this procedure is not novel, although only in this text do we call the resulting space a "split" subspace.

projection basis V.

4.2.1 Producing a real basis for a complex Krylov-subspace

If we use a shift σ with nonzero $\Im$ part but use real basis for projection, we have no choice but to project on to a split-Krylov space of the form (65). One way to do that is to perform a run of the Arnoldi process (algorithm 1) in complex arithmetic with matrices $\mathbf{H}(\sigma)$ and $\mathbf{R}(\sigma)$, yielding the complex orthogonal basis $V = \begin{bmatrix} v_1 & v_2 & \cdots & v_n \end{bmatrix}$ and then split V into $\Re$ and $\Im$, such as

$$\begin{bmatrix} v_1^{\mathbf{r}} & v_1^{\mathbf{i}} & v_2^{\mathbf{r}} & v_1^{\mathbf{i}} & \cdots & v_n^{\mathbf{r}} & v_n^{\mathbf{i}} \end{bmatrix}. \tag{66}$$

But the set (66) is no longer orthogonal, and possibly linearly dependent. Orthogonalization of (66) requires up to $(2n)^2 N$ flops.

Ruhe's method

It seems that it would be ideal to create an orthogonal, real basis for (65) directly during the iterative process. Ruhe addresses this in [41], in the context of a single-vector general rational-Krylov method for eigenvalue finding. His method involves considering $\Re$ and $\Im$ parts of the power iterate separately, so that each iteration yields two new real vectors. Implemented as a modification of the Arnoldi algorithm, line 4 of Algorithm 1 becomes

$$a_k + ib_k = \mathbf{H} v_k,$$

and we orthogonalize vectors a_k and b_k separately. However, it is not clear what vector we should iterate with next in order to build a basis for (65). It is not clear whether the real basis produced by Ruhe's method spans a Krylov-subspace, let alone a basis for the split-Krylov-subspace (65), nor whether projection with this basis matches moments. Surprisingly there is neither much literature nor results on this topic with regards to model order-reduction, but further work in this area could be promising, as the typical split and re-orthogonalize procedure seems redundant.

4.2.2 Using a real inner-product

Let us assume that we work strictly in complex arithmetic followed by splitting the complex basis into $\Re$ and $\Im$ parts and re-orthogonalizing as a post-processing step. One way to cut complex-vector orthogonalization costs in half for a Gram-Schmidt based process during the the Krylov process is to use the alternate real-valued inner-product that we denote by $\langle \cdot, \cdot \rangle_{\mathbb{R}}$.

For complex vectors $v = a + ib$ and $w = x + iy$, define

$$\langle v, w \rangle_{\mathbb{R}} = x^T a - y^T b \in \mathbb{R}. \tag{67}$$

Consider "orthogonalization" using (67) instead of the standard complex Euclidean inner-product

$$v^H w = (x^T a - y^T b) + i(x^T b + y^T a). \tag{68}$$

The real inner-product (67) is cheaper to compute than (68). The basis V produced by a cycle of the Arnoldi process (algorithm 1) using this inner-product for orthogonalization is not real-valued, nor is it orthogonal in the Euclidean sense. We cannot use it to make orthogonal projections as in (4), and an n-dimensional basis V orthogonal with respect to (67) does not span $\mathcal{K}_n(\mathbf{H}, \mathbf{R})$, in general. However, it satisfies (65); that is,

$$\operatorname{span} V^* = \mathcal{K}_n(\mathbf{H}, \mathbf{R})^* = \operatorname{span} \mathcal{K}_n(\mathbf{H}(\sigma), \mathbf{R}(\sigma)) \cup \mathcal{K}_n(\mathbf{H}(\overline{\sigma}), \mathbf{R}(\overline{\sigma})),$$

which works out because we must split and re-orthogonalize anyway.

Constructing such an equivalent-real orthogonal basis for $\mathcal{K}_n(\mathbf{H}, \mathbf{R})$ can be accomplished by replacing line 6

$$h_{jk} = v_k^H r_k$$

in algorithm 1 with

$$g_{jk} = \langle v_k, r_k \rangle_{\mathbb{R}}$$

where $\langle \cdot, \cdot \rangle$ is defined by (67).

However, the matrix

$$\boldsymbol{G} = [g_{jk}] \neq V^H \mathbf{H} V \tag{69}$$

of orthogonalization coefficients is no longer an orthogonal-projection in the Euclidean sense which limits this idea's utility for model order-reduction.

4.3 Equivalent-real formulations

A Krylov process that orthogonalizes iterates with respect to (67) is effective for constructing a split-worthy basis because it is an Euclidean inner-product with respect to the "equivalent-real" Krylov-subspace $\mathcal{K}_n(\ddot{\mathbf{H}}, \ddot{\mathbf{R}}) \subset \mathbb{R}^{2N}$ induced by the *equivalent-real*, or *realified* formulations

$$\ddot{\mathbf{H}} = \begin{bmatrix} \mathbf{H^r} & -\mathbf{H^i} \\ \mathbf{H^i} & \mathbf{H^r} \end{bmatrix} \quad \text{and} \quad \ddot{\mathbf{R}} = \begin{bmatrix} \mathbf{R^r} \\ \mathbf{R^i} \end{bmatrix}. \tag{70}$$

of $\mathbf{H} = \mathbf{H^r} + i\mathbf{H^i}$ and $\mathbf{R} = \mathbf{R^r} + i\mathbf{R^i}$ from (9). This idea is from [36, Sec. 5] and [11, '$K1$-formulation']. There is a general definition and discussion of realified spaces as a pure-mathematics topic in [34].

A basis

$$\ddot{V} = \begin{bmatrix} \ddot{v}_1 & \ddot{v}_2 & \cdots & \ddot{v}_n \end{bmatrix} \tag{71}$$

for the Krylov-subspace $\mathcal{K}_n(\ddot{\mathbf{H}}, \ddot{\mathbf{R}})$ induced by the realified matrices (70) consists of vectors

$$\ddot{v} = \begin{bmatrix} \ddot{v}^{\mathbf{t}} \\ \ddot{v}^{\mathbf{b}} \end{bmatrix}$$

where we call $\ddot{v}^{\mathbf{t}}$ and $\ddot{v}^{\mathbf{b}}$ in $\mathbb{R}^N$ the *top* and *bottom* parts of $\ddot{v}$. We define a split of the equivalent-real basis (71) as

$$\ddot{V}_n^* := \begin{bmatrix} \ddot{v}_1^{\mathbf{t}} & \ddot{v}_1^{\mathbf{b}} & \ddot{v}_2^{\mathbf{t}} & \ddot{v}_2^{\mathbf{b}} & \cdots & \ddot{v}_n^{\mathbf{t}} & \ddot{v}_n^{\mathbf{b}} \end{bmatrix}, \tag{72}$$

which is analogous to the split (65) of set of complex vectors into $\Re$ and $\Im$-parts.

The next result establishes that

$$\operatorname{span} \ddot{V}_n^* = \mathcal{K}_n(\mathbf{H}, \mathbf{R})^*.$$

That is, whether we construct a basis for a complex Krylov-subspace $\mathcal{K}_\eta(\mathbf{H}, \mathbf{R})$ using complex arithmetic, or using real arithmetic with equivalent real forms $\ddot{\mathbf{H}}$ and $\ddot{\mathbf{R}}$, splitting the basis yields

the same spanning set.

Note that equivalent-real forms never need to be explicitly formed. They are implied by the use of the real inner-product (67) on complex vectors.

4.3.1 Equivalence of split complex and equivalent-real subspaces

Lemma 1. *Consider the equivalent-real formulations* $\ddot{\mathbf{H}}$ *and* $\ddot{\mathbf{R}}$ *of* $\mathbf{H}$ *and* $\mathbf{R}$ *as defined by* (70). *Then equivalent real formulation of* $\mathbf{H}^j\mathbf{R}$ *is* $\ddot{\mathbf{H}}^j\ddot{\mathbf{R}}$ *for any integer* $j = 0, 1, 2, \ldots$, *i.e.*

$$(\mathbf{H}^j\mathbf{R})^* = \ddot{\mathbf{H}}^j\ddot{\mathbf{R}}. \tag{73}$$

Equivalently,

$$\ddot{\mathbf{H}}^j\ddot{\mathbf{R}} = \begin{bmatrix} \Re(\mathbf{H}^j\mathbf{R}) \\ \Im(\mathbf{H}^j\mathbf{R}) \end{bmatrix}. \tag{73*}$$

Proof. Trivially for $j = 0$ we have $\ddot{\mathbf{R}} := \begin{bmatrix} \mathbf{R^r} & \mathbf{R^i} \end{bmatrix}^T$. For $j \geq 1$, let $K = \mathbf{H}^{j-1}R$. Then $\widehat{K} = \begin{bmatrix} K^{\mathbf{r}} & K^{\mathbf{i}} \end{bmatrix}^T$ is the equivalent-real form of $K = K^{\mathbf{r}} + iK^{\mathbf{i}}$, so

$$\ddot{\mathbf{H}}^j\ddot{\mathbf{R}} = \ddot{\mathbf{H}}\widehat{K} = \begin{bmatrix} \mathbf{H^r} & -\mathbf{H^i} \\ \mathbf{H^i} & \mathbf{H^r} \end{bmatrix} \begin{bmatrix} K^{\mathbf{r}} \\ K^{\mathbf{i}} \end{bmatrix} = \begin{bmatrix} \mathbf{H^r}K^{\mathbf{r}} - \mathbf{H^i}K^{\mathbf{i}} \\ \mathbf{H^r}K^{\mathbf{i}} + \mathbf{H^i}K^{\mathbf{r}} \end{bmatrix}$$

is the equivalent real formulation of

$$\begin{aligned} \mathbf{H}^jR = \mathbf{H}K &= (\mathbf{H^r} + i\mathbf{H^i})(K^{\mathbf{r}} + iK^{\mathbf{i}}) \\ &= \left(\mathbf{H^r}K^{\mathbf{r}} - \mathbf{H^i}K^{\mathbf{i}}\right) + i\left(\mathbf{H^r}K^{\mathbf{i}} + \mathbf{H^i}K^{\mathbf{r}}\right) \end{aligned}$$

□

It follows as a corollary that the split-Krylov-subspaces (65) and (72) induced by each pair, are equal.

$$\mathcal{K}_n(\ddot{\mathbf{H}}, \ddot{\mathbf{R}})^* = \mathcal{K}_n(\mathbf{H}, \mathbf{R})^* \tag{74}$$

The inner-products (68) and (67) yield different notions of orthogonality of a complex basis

and its equivalent-real counterpart, and ultimately incompatible spaces. The inner-product (67) implies a weaker orthogonality than (68): if two complex vectors $v, w \in \mathbb{C}^N$ are orthogonal then it follows that their equivalent real forms $\hat{v}, \hat{w} \in \mathbb{R}^{2N}$ are also orthogonal, but the converse is not true in general. A basis $\ddot{V}$ of the block-Krylov-subspace $\mathcal{K}_n(\ddot{\mathbf{H}}, \ddot{\mathbf{R}})$ cannot be identified with a basis of $\mathcal{K}_n(\mathbf{H}, \mathbf{R})$: if we express each basis vector $\ddot{v}_j$ as a complex vector

$$v_j = \ddot{v}_j^{\mathbf{t}} + i\ddot{v}_j^{\mathbf{b}},$$

the resulting set of complex vectors $\{v_j\}$ will generally neither be orthogonal, nor will it span $\mathcal{K}_n(\mathbf{H}, \mathbf{R})$. However, we are not interested in $\mathcal{K}_n(\mathbf{H}, \mathbf{R})$, but rather its split variation $\mathcal{K}_n(\mathbf{H}, \mathbf{R})^*$, which is why the result (74) of Lemma 73 is promising.

The norms implied by (68) and (67) for a complex vector $v \in \mathbb{C}^N$ and its equivalent real form $\ddot{v} \in \mathbb{R}^{2N}$ are equal:

$$\|\ddot{v}\|_2^2 = \ddot{v}^T\ddot{v} = v^H v = \langle v, v\rangle = \|v\|_2^2 . \tag{75}$$

Lemma 1 establishes that complex and realified forms of $\mathbf{H}$ and $\mathbf{R}$ run for equal numbers of iterations induce the same split Krylov-subspace $\mathcal{K}_n(\mathbf{H}, \mathbf{R})^*$. The next result establishes that the basis vectors produced by an iteration of the Arnoldi process advance the split-Krylov-subspace (65) in the same order, regardless of whether we use complex or equivalent-real formulation (i.e. complex formulation with inner-product (67)).

We show this for the simpler case that $\mathbf{R} = \mathbf{r} \in \mathbb{C}^N$ is a single vector. The result of Theorem 3 can be extended to the more general *Band*-Krylov process which is applicable to MIMO model reduction.

Theorem 3. *Consider* $\mathbf{H}$, $\mathbf{r}$ *from* (9) *and their realified formulations* $\ddot{\mathbf{H}}$ *and* $\ddot{\mathbf{r}}$ *defined by* (70). *Let* $V = \begin{bmatrix} v_1 & v_2 & \cdots & v_n \end{bmatrix}$ *be the orthonormal basis implied by* n *Arnoldi iterations of* $\mathbf{H}$ *with start vector* $\mathbf{r}$, *and let* $\ddot{V} = \begin{bmatrix} \ddot{v}_1 & \ddot{v}_2 & \cdots & \ddot{v}_n \end{bmatrix}$, *with* $\ddot{v}_j = \begin{bmatrix} \ddot{v}_j^{\mathbf{t}} & \ddot{v}_j^{\mathbf{b}} \end{bmatrix}^T$, *be the analogous vectors produced by Arnoldi iterations using* $\ddot{\mathbf{H}}$ *and* $\ddot{\mathbf{r}}$. *Then there exist scalars* $\alpha, \beta \in \mathbb{R}$ *and real vectors* $w, z \in \mathcal{K}_n(\mathbf{H}, \mathbf{r})^*$ *such that*

$$\Re(v_n) = \alpha\ddot{v}_n^{\mathbf{t}} + w \qquad \textit{and} \qquad \Im(v_n) = \beta\ddot{v}_n^{\mathbf{b}} + z. \tag{76}$$

Proof. We will prove (76) by induction. For $n = 1$ we have $v_1 = \mathbf{r}/\|\mathbf{r}\|_2$, $\ddot{v}_1 = \ddot{\mathbf{r}}/\|\ddot{\mathbf{r}}\|_2$, where $\|\ddot{\mathbf{r}}\|_2 = \|\mathbf{r}\|_2$ by (75), so $\Re(v_1) = \ddot{v}_1^{\mathbf{t}}$ and $\Im(v_1) = \ddot{v}_1^{\mathbf{b}}$, trivially satisfying (76).

Now assume we have performed $n \geq 1$ steps of the standard Arnoldi process to obtain an orthonormal basis V for $\mathcal{K}_n(\mathbf{H}, \mathbf{r})$, and assume a complex span for its split space $\mathcal{K}_n(\mathbf{H}, \mathbf{r})^*$ (of dimension n), so that

$$\mathcal{K}_n(\mathbf{H}, \mathbf{r}) \subseteq \mathcal{K}_n(\mathbf{H}, \mathbf{r})^* = \operatorname{span}\begin{bmatrix} \widetilde{v}_1 & \widetilde{v}_2 & \cdots & \widetilde{v}_\eta \end{bmatrix} \tag{77}$$

for real basis vectors $\widetilde{v}_j \in \mathbb{R}^N$. On the $n \geq 1$-th step, the Arnoldi process with $\mathbf{H}$ and $\mathbf{r}$ computes scalar orthogonalization coefficients $\{h_{jn}\}_{j=1}^n \subset \mathbb{C}$ and $h_{n+1,n} \in \mathbb{R}$ such that

$$\begin{aligned} h_{n+1,n} v_{n+1} &= \mathbf{H} v_n - \sum_{j=1}^{n} h_{jn} v_j \\ &= \mathbf{H}^n \mathbf{r} + \sum_{j=1}^{n} c_j v_j, \qquad c_j \in \mathbb{R} \\ &= \mathbf{H}^n \mathbf{r} + \sum_{j=1}^{\eta} d_j \widetilde{v}_j, \qquad d_j \in \mathbb{C}, \quad \text{by (77)} \\ &= \mathbf{H}^n \mathbf{r} + w_1 + i z_1, \qquad w_1, z_1 \in \mathcal{K}_n(\mathbf{H}, \mathbf{r})^* \cap \mathbb{R}^N. \end{aligned} \tag{78}$$

Lemma 1 implies that we can re-write (78) in realified form as

$$h_{n+1,n} \begin{bmatrix} \Re(v_{n+1}) \\ \Im(v_{n+1}) \end{bmatrix} = \ddot{\mathbf{H}}^n \ddot{\mathbf{r}} + \begin{bmatrix} w_1 \\ z_1 \end{bmatrix}. \tag{79}$$

On the other hand, after n iterations Arnoldi iterations with $\ddot{\mathbf{H}}$ and $\ddot{\mathbf{r}}$ we have

$$\hat{h}_{n+1,n} \ddot{v}_{n+1} = \ddot{\mathbf{H}}^n \ddot{\mathbf{r}} - \sum_{j=1}^{n} \hat{h}_{jn} \ddot{v}_j,$$

so

$$\begin{aligned}
\hat{h}_{n+1,n}\begin{bmatrix}\ddot{v}^{\mathbf{t}}_{n+1}\\ \ddot{v}^{\mathbf{b}}_{n+1}\end{bmatrix} &= \ddot{\mathbf{H}}^n\ddot{\mathbf{r}} + \sum_{j=1}^{n}\hat{c}_j\begin{bmatrix}\ddot{v}^{\mathbf{t}}_{j}\\ \ddot{v}^{\mathbf{b}}_{j}\end{bmatrix}, \qquad \hat{c}_j \in \mathbb{R}\\
&= \ddot{\mathbf{H}}^n\ddot{\mathbf{r}} + \sum_{j=1}^{\eta}\begin{bmatrix}\hat{a}_j\widetilde{v}_j\\ \hat{b}_j\widetilde{v}_j\end{bmatrix}, \qquad \hat{a}_j, \hat{b}_j \in \mathbb{R}\\
&= \ddot{\mathbf{H}}^n\ddot{\mathbf{r}} + \begin{bmatrix}w_2\\ z_2\end{bmatrix},
\end{aligned}$$

where $w_2, z_2 \in \mathcal{K}_n(\mathbf{H}, \mathbf{r})^* \cap \mathbb{R}^N$. □

Theorem 3 establishes that Arnoldi vectors generated using $\ddot{\mathbf{H}}$ and $\ddot{\mathbf{r}}$ yield basis vectors for $\mathcal{K}_n(\mathbf{H}, \mathbf{r})^*$ in the same order as those obtained from $\mathbf{H}$ and $\mathbf{r}$; in fact, up to finite precision error they yield exactly the same basis.

4.3.2 Reduced-order models via equivalent-real formulations

via explicit projection

We showed in Theorem 3 that splitting a complex basis that is orthogonal with respect to the real inner-product (67) yields the same spanning-set as if we had used the standard complex Euclidean inner-product (68). Then the explicitly-projected ROM (3) using a basis (65) for the split-Krylov-subspace $\mathcal{K}_\eta(\mathbf{H}, \mathbf{R})^*$ is the same regardless of which of those inner-products we use.

via implicit projection

The matrix $\boldsymbol{G}$ of orthogonalization coefficients from the equivalent-real Arnoldi process, (69), is a Rayleigh-quotient approximation to the equivalent-real operator

$$\ddot{\mathbf{H}} = \begin{bmatrix}\mathbf{H}^{\mathbf{r}} & -\mathbf{H}^{\mathbf{i}}\\ \mathbf{H}^{\mathbf{i}} & \mathbf{H}^{\mathbf{r}}\end{bmatrix}$$

of (70), and not the original complex-shifted operator $\mathbf{H}$. It is the projection

$$\boldsymbol{G} = \ddot{V}^T \ddot{\mathbf{H}} \ddot{V}$$

that would be formed by orthogonal projection using $\ddot{V} \in \mathbb{R}^{2N \times \eta}$, where $\operatorname{span} \ddot{V} = \mathcal{K}_n(\ddot{\mathbf{H}}, \ddot{\mathbf{R}})$ for the implied Krylov-subspace induced by equivalent-real forms (70). Thus, an implicitly projected ROM (28) is not so simple to characterize. For example, it is known that the spectrum

$$\Lambda(\ddot{\mathbf{H}}) = \Lambda(\mathbf{H}) \cup \Lambda(\overline{\mathbf{H}})$$

of $\ddot{\mathbf{H}}$ contains spectral information for the complex-conjugate $\overline{\mathbf{H}}$ which complicates matters because we are interested only in the spectrum of $\mathbf{H}$. This makes analyzing a ROM transfer-function via implicit-projection onto a realified Krylov-subspace non-trivial, but it might be a promising improvement if developed further.

Chapter 5

The band-Arnoldi process and a proposed thick-restarted variant

"Thick"-starting a Krylov-process that iterates with $\mathbf{H}$ starting on $\mathbf{R}$ means that we instead start on $\begin{bmatrix} Z & \mathbf{R} \end{bmatrix}$ where Z is a basis of known Ritz-vectors of $\mathbf{H}$. The notion of a re-start for multi-point MOR is that once we have iterated enough with $\mathbf{H}_1 = \mathbf{H}(\sigma_1)$ on $\mathbf{R}_1 = \mathbf{R}(\sigma_1)$, creating a ROM approximation about σ_1, we can start over, iterating about σ_2 with $\mathbf{H}_2$ and $\mathbf{R}_2$, avoiding wasting computation on rediscovering invariant-subspace. Recall that all $\mathbf{H}(\sigma)$ (over σ at which $\mathbf{H}(\sigma)$ is defined) share invariant-subspaces. If we determine a convergent-enough invariant-subspace Y_1 of $\mathbf{H}_1$, then we can thick-restart the process with $\mathbf{H}_2$ acting on $\begin{bmatrix} Y_1 & \mathbf{R}_2 \end{bmatrix}$.

First we will introduce the Band-Arnoldi algorithm and formalism that facilitate a multiple-vector iteration, since our start block $\mathbf{R} = \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \cdots & \mathbf{r}_m \end{bmatrix}$ contains m vectors for a general (MIMO) model.

The thesis of this document is a thick-restarted Arnoldi-type algorithm for multiple-interpolation-point MOR (rational-interpolation), where the model to be reduced is assumed to be MIMO. An advantage to our method over previous attempts at a restarted MOR method that assume the standard single-vector formalism is that we are not forced to choose between re-starting with a proper "start" vector or block $\mathbf{R}$ (preserving moment matching), a combination of Ritz-vectors, or the left-over candidate vector from the previous cycle: we can restart with any number of vectors,

or all of them. We can start with an arbitrary set of vectors in addition to $\mathbf{R}$. For simplicity we will assume the additional vectors are Ritz-vectors, but that is by no means required.

To the best of our knowledge this is the first restarted Krylov-subspace for MIMO model reduction that uses the band-Arnoldi process, rather than a block process. It appears that the implicitly-restarted method [48] for finding eigenvalues restarts band-Arnoldi process in a similar way, but not for model reduction.

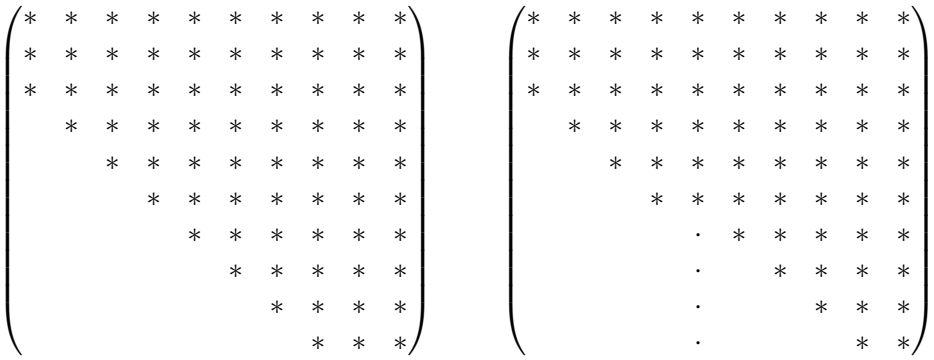
5.1 Band-Arnoldi algorithm

Band-Arnoldi process of [17] is included here as algorithm 2. An older (possibly the original) version of a band-Krylov process was given by [39] in 1979. The band-type iteration is considered less efficient than a block method like the block-Arnoldi of [29], because it is a single-vector iteration that advances the space with one matrix-vector product at a time. A block method can take advantage of very efficient algorithms for matrix-matrix products. However, block methods suffer with the changing block-size resulting from deflation of linear-dependence within the iterating band. Deflation and changing block-dimension are an essential part of our method so the band-iteration is more fitting.

Band-Arnoldi cycles through a "band" of candidate-vectors $\begin{bmatrix}\hat{v}_n & \hat{v}_{n+1} & \dots & \hat{v}_{n+m_c}\end{bmatrix}$, where $m_c(n)$ is the current band size on the n-th iteration of the main loop. The initial band is the start block

$$\begin{bmatrix}\hat{v}_1 & \hat{v}_2 & \cdots & \hat{v}_m\end{bmatrix} = \begin{bmatrix}\mathbf{r}_1 & \mathbf{r}_2 & \cdots & \mathbf{r}_m\end{bmatrix} = \mathbf{R}.$$

On the j-th iteration of the algorithm, the candidate-vector $\hat{v}_j$ either gets deflated or becomes Krylov basis-vector v_j, which is then advanced via Arnoldi iteration to be candidate-vector $\hat{v}_{j+m_c}$. If we deflate $\hat{v}_j$ then the band size m_c is decremented. Since the algorithm proceeds as a continuous cycle rather than a block iteration, at any step n it is simpler to refer to the computed basis $V \in \mathbb{C}^{N\times n}$ for $\mathcal{K}_n(\mathbf{H},\mathbf{R})$, where n is the dimension of the basis, rather than a block-degree.



(a) (b)

Table 5.1: (a) is an example of the nonzero structure of $\widetilde{\mathbf{H}}$ after 10 iterations of a band-Arnoldi process with band-size $m = 2$. (b) is what $\widetilde{\mathbf{H}}$ would look like if an inexact deflation occurred on the 5-th iteration, where '$\cdot$' represents the small contribution of $\widetilde{\mathbf{H}}_{\mathcal{E}} = V^H \dot{V}$, comprised values smaller than dtol. If an exact deflation occurred then those values would be zero.

5.1.1 Band-Arnoldi relation

The band-Arnoldi algorithm run for n-iterations with operator $\mathbf{H}$ and start-block $\mathbf{R}$ returns a basis $V \in \mathbb{C}^{N\times n}$ for $\mathcal{K}_n(\mathbf{H}, \mathbf{R})$, remaining candidate-vectors $\hat{V} = \begin{bmatrix} \hat{v}_{n+1} & \hat{v}_{n+2} & \dots & \hat{v}_{n+m_c} \end{bmatrix}$, deflated vectors $\dot{V} = \begin{bmatrix} \dot{v}_1 & \dot{v}_2 & \dots & \dot{v}_d \end{bmatrix}$ if any deflation occured, and Rayleigh-quotient $\widetilde{\mathbf{H}} = V^H \mathbf{H} V$ that satisfy the band-Arnoldi relation

$$\mathbf{H}V = V\widetilde{\mathbf{H}} + \begin{bmatrix} (I - VV^H)\dot{V} & \hat{V} \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \end{bmatrix} \tag{80}$$

$$= V\widetilde{\mathbf{H}} + \widehat{\dot{V}}F. \tag{80*}$$

$\widetilde{\mathbf{H}}$ is strictly upper-Hessenberg in the single vector setting. For general $\mathbf{R}$ with dimension m, $\widetilde{\mathbf{H}}$ has zeros below the $(m+1)$-th diagonal if no deflation was performed, or possibly non-zero (but smaller than deflation tolerance dtol) columns in the typically zero region, corresponding to the deflated vectors $\dot{V}$. For an example, see table 5.1. F_1 and F_2 are indexing matrices that position vectors $\dot{v}_j$ and $\hat{v}_j$ respectively into the n available positions of the $N \times n$ block (80).

The compact shorthand $\widehat{\dot{V}}$ in (80*) reflects that $\dot{V}$ is often zero or negligible for most computation purposes, but we must include it in order for equality to hold.

In addition to V, $\hat{V}$, $\dot{V}$, and $\widetilde{\mathbf{H}}$, algorithm 2 returns $\widetilde{\boldsymbol{\rho}}$, and $\widetilde{\boldsymbol{\rho}}^{\text{defl}}$ where

$$\begin{aligned}\mathbf{R} &= V\widetilde{\boldsymbol{\rho}} + \widetilde{\boldsymbol{\rho}}^{\text{defl}} \\ &= V\widetilde{\boldsymbol{\rho}} + \dot{V}F_0,\end{aligned} \tag{81}$$

and $V^H\mathbf{R} = \widetilde{\boldsymbol{\rho}}$.

5.1.2 Candidates/residual term

$$\begin{aligned}\hat{V}F_2 &= \begin{bmatrix}0 & 0 & \cdots & 0 & \hat{V}\end{bmatrix} \in \mathbb{C}^{N\times n} \\ &= \begin{bmatrix}\hat{v}_{n+1} & \hat{v}_{n+2} & \ldots & \hat{v}_{n+m_c}\end{bmatrix} \begin{bmatrix}0 & \cdots & 1 & & \\ 0 & \cdots & & 1 & \\ 0 & \cdots & & & \ddots \\ 0 & \cdots & & & & 1\end{bmatrix}\end{aligned}$$

is the residual term involving the band $\hat{V}$ of candidate-vectors after the n-th iteration of the main loop. The matrix $F_2 \in \{0,1\}^{m_c\times n} = \begin{bmatrix}0 & 0 & \cdots & 0 & I\end{bmatrix}$ simplifies to e_n^T for the single vector iteration. F_2 places $\hat{V}$ in the last m_c of n positions. Note that $V^H\hat{V} = 0$.

5.1.3 Deflation term

$\dot{V}F_1 \in \mathbb{C}^{N\times n}$ is the zero or mostly-zero matrix $\hat{V}^{\text{defl}}$ implied by algorithm 2 (band-Arnoldi). If no deflation or only exact deflation occurred then $\dot{V} = 0$ and $\dot{V}F_1$ is an $N \times n$ matrix of zeros. If inexact deflation was performed on the j-th candidate-vector then $j \in \mathcal{I}$ and $\hat{v}_j^{\text{defl}} \neq 0$. Negative or zero indices in $\mathcal{I}$ correspond to deflations that happened within the start block $\mathbf{R}$. For example, if $j - m \leq 0$ then $\widetilde{\boldsymbol{\rho}}_j^{\text{defl}} = \hat{v}_{j-m}^{\text{defl}}$. We may similarly define F_0 so that $\widetilde{\boldsymbol{\rho}}^{\text{defl}} = \dot{V}F_0$.

As an example of a deflation matrix $\hat{V}^{\text{defl}}$, suppose $d = 2$ vectors $\dot{v}_1 = \hat{v}_2^{\text{defl}}$ and $\dot{v}_2 = \hat{v}_5^{\text{defl}}$ were deflated at iterations $m_c + 2$ and $m_c + 5$ of a band-Arnoldi process of $n = m_c + 10$ iterations, with

band-size m_c. Then for standard basis vectors $e_2, e_5 \in \mathbb{R}^{10}$,

$$\begin{aligned}\dot{V}F_1 = \hat{V}^{\text{defl}} &= \begin{bmatrix} 0 & \hat{v}_2^{\text{defl}} & 0 & 0 & \hat{v}_5^{\text{defl}} & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\ &= \begin{bmatrix} 0 & \dot{v}_1 & 0 & 0 & \dot{v}_2 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\ &= \begin{bmatrix} \dot{v}_1 & \dot{v}_2 \end{bmatrix} \begin{bmatrix} e_2^T \\ e_5^T \end{bmatrix}.\end{aligned} \tag{82}$$

The band-Arnoldi algorithm deflates a candidate-vector $\hat{v}_j$ (i.e. removes it from further iterations) if $\|\hat{v}_j\| \leq \text{dtol}$[1] after orthogonalizing $\hat{v}_j$ against $V = \{v_1, v_2, ..., v_j\}$, which means it is almost linearly dependent with previous basis vectors. Algorithm 2 then sets $\hat{v}_j^{\text{defl}} := \hat{v}_n$ and removes it as a candidate, and the current band size m_c is decreased by one. $\hat{v}^{\text{defl}}$ is no longer used for iterations and basis vectors $v_{j+1}, v_{j+2}, \ldots, v_{n+m_c}$ are not made orthogonal to $\hat{v}_j^{\text{defl}}$. Then

$$V^H \hat{v}_j^{\text{defl}} = \begin{bmatrix} 0 & 0 & \cdots & 0 & v_{j+1}^H \hat{v}_j^{\text{defl}} & v_{j+2}^H \hat{v}_j^{\text{defl}} & \cdots & \hat{v}_j^H \hat{v}_j^{\text{defl}} \end{bmatrix}^T. \tag{83}$$

(83) implies that $\|V^H \dot{v}\| \leq \|\dot{v}\| \leq \text{dtol}$.

If no/exact deflation was performed, $\widetilde{\mathbf{H}}$ is of the form in table 5.1a, otherwise $\widetilde{\mathbf{H}}$ may have non-zero entries in the typically-zero region $\widetilde{\mathbf{H}}_{\mathcal{E}} = V^H \dot{V}$. If an inexact deflation occurred on the j-th iteration, (83) is included in the Rayleigh-quotient $\widetilde{\mathbf{H}}$ as the j-th column of $\widetilde{\mathbf{H}}_{\mathcal{E}}$. Then

$$\|\widetilde{\mathbf{H}}_{\mathcal{E}}\| = \|V^H \dot{V}\|_F \leq \|\dot{V}\|_F \leq \text{dtol}\sqrt{d}, \tag{84}$$

and

$$\|(I - VV^H)\dot{V}\|_F \leq \|\dot{V}\|_F. \tag{85}$$

$\widetilde{\boldsymbol{\rho}}^{\text{defl}}$ in (81) is also an all or mostly-zero matrix of very small norm, representing deflations that occurred during the first m-iterations, i.e. linear dependence within the start block $\mathbf{R}$.

[1] [37] suggests $\text{dtol} = \sqrt{\epsilon}$, where machine-$\epsilon = 2^{-52} \approx$ 2.22e-16 in double-precision (64-bit) floating point.

5.1.4 Residual-norms for determining Ritz-convergence

Given basis $W = \begin{bmatrix} w_1 & w_2 & \cdots & w_n \end{bmatrix}$ of the eigenvalue-decomposition $\widetilde{\mathbf{H}}W = W\Lambda$ and basis $Z = VW$ of long Ritz-vectors, the residual for a Ritz-pair (λ_j, z_j), is

$$\mathbf{H}z_j - z_j\lambda_j = \begin{bmatrix} (I - VV^H)\dot{V} & \hat{V} \end{bmatrix} Fw_j$$

The square of the so-called residual-norm of the Ritz-pair is

$$\begin{aligned} \|\mathbf{H}z_j - z_j\lambda_j\|_2^2 &\leq d\,\varepsilon_M + \|\hat{V}F_2w_j\|_2^2 \\ &= d\,\varepsilon_M + \|\hat{V}\tilde{w}_j\|_2^2. \end{aligned} \tag{86}$$

where $\tilde{w}_j \in \mathbb{C}^{m_c}$ is the last m_c elements (rows) of short Ritz-vector w_j.

(86) suggests a few different ways to cheaply estimate *relative* residual-norm

$$\mathtt{rr}_j = \frac{\|\mathbf{H}z_j - z_j\lambda_j\|}{\|\lambda_j z_j\|} \tag{87}$$

for a Ritz-pair (λ_j, z_j). We assume $\|z_j\| = 1$, so that $\|\lambda z_j\| = |\lambda|$.

Some methods estimate the relative residual-norm as $\|\mathbf{H}z_j - z_j\lambda_j\|/\|\mathbf{H}z_j\|$ with an estimate of $\|\mathbf{H}\|$ or with $\|\widetilde{\mathbf{H}}\|$. We use $|\lambda|$ because it is uncertain whether $\|\mathbf{H}\|$ or $\|\widetilde{\mathbf{H}}\|$ are good estimates of $\|\mathbf{H}\|$.

Relative residual-norm bounds

Assuming $d\,\varepsilon_M$ is negligible,

$$\mathtt{rr}_j \leq \left\|\hat{V}\right\| \frac{\|\tilde{w}_j\|}{|\lambda_j|} \tag{88}$$

is an estimate of (87), as is

$$\mathtt{rr}_j \leq \frac{1}{|\lambda|} \begin{bmatrix} \|\hat{v}_1\| & \|\hat{v}_2\| & \cdots & \|\hat{v}_{m_c}\| \end{bmatrix} \begin{bmatrix} |\tilde{w}_j^{(1)}| \\ |\tilde{w}_j^{(2)}| \\ \vdots \\ |\tilde{w}_j^{(m_c)}| \end{bmatrix}. \tag{89}$$

Both (88) and (89) are cheaper to compute than norms $\|\hat{V}\tilde{w}_j\|$ of potentially large matrix-vector products, but (89) might be better if $\hat{V}F_2$ has rank greater than one. Estimates (88) and (89) are equal for rank-1 residuals.

Algorithm 2: BAND-ARNOLDI

Input: $\mathbf{H}$ and start-block $\mathbf{R} = \begin{bmatrix}\mathbf{r}_1 & \mathbf{r}_2 & \cdots & \mathbf{r}_m\end{bmatrix}$, $n_{\texttt{max}}$

Output: basis V for $\mathcal{K}_n(\mathbf{H}, \mathbf{R})$, deflated vectors $\dot{V}$, candidate-vectors $\hat{V}$, $\widetilde{\mathbf{H}}$, and $\widetilde{\boldsymbol{\rho}}$ that satisfy (80)

$\hat{v}_i := \mathbf{r}_i \quad$ for $i = 1, 2, \ldots, m$
$m_c := m$
$\mathcal{I} := \emptyset$
for $n = 1$ **to** n_{max} **do**
  while $\|\hat{v}_n\|_2 < dtol \cdot \|\mathbf{H}\|_{est}$ **do** % remove $\hat{v}_n$ if necessary (deflation)
    $\hat{v}^{\text{defl}}_{n-m_c} := \hat{v}_n$
    $\mathcal{I} = \mathcal{I} \cup \{n - m_c\}$ % locations in $\hat{V}^{\texttt{defl}}$ (or $\widetilde{\boldsymbol{\rho}}^{\texttt{defl}}$) that contain deflated vectors
    $m_c := m_c - 1$ % we assume no early termination
    $\hat{v}_j := \hat{v}_{j+1} \quad$ for $j = n, n+1, \ldots, n + m_c - 1$
  $h_{n,n-m_c} := \|\hat{v}_n\|_2$
  $v_n := \hat{v}_n / \|\hat{v}_n\|_2$
  for $j = n + 1$ **to** $n + m_c - 1$ **do** % Make candidates $\{\hat{v}_1, \hat{v}_2, ..., \hat{v}_n\}$ orthogonal to v_n
    $h_{n,j-m_c} := v_n^H \hat{v}_j$
    $\hat{v}_j := \hat{v}_j - h_{n,j-m_c} v_j$
  $\hat{v}_{m_c+n} := \mathbf{H} v_n$
  for $j = 1$ **to** n **do** % Make $\hat{v}_{m_c+n}$ orthogonal to previous $\{v_1, v_2, \ldots, v_n\}$
    $h_{jn} := v_j^H \hat{v}_{m_c+n}$
    $\hat{v}_{m_c+n} := \hat{v}_{m_c+n} - h_{jn} v_j$
  for $j \in \mathcal{I}$ **do**
    $h_{nj} := v_n^H \hat{v}^{\text{defl}}_j$
return V, $\widetilde{\mathbf{H}}$, $\widetilde{\boldsymbol{\rho}} = \left[h_{ij}\right]_{i=1,2,\ldots,m}^{j=1-m\ldots,1,0}$, $\hat{V}$, $\hat{V}^{defl}$, $\widetilde{\boldsymbol{\rho}}^{defl} = \left[\hat{v}_j^{defl}\right], j = 1-m, ..., 1, 0,$

5.2 Thick-restarting the Band-Arnoldi process

In §5.2.1 we will discuss the theory of thick-starting a band-Arnoldi process explicitly with ℓ Ritz-vectors. In §5.2.4 we will show how we implemented the thick-restart cycle in a multiple-shift model-reduction context where an orthogonal basis Y of "locked" vectors are explicitly passed into the band-Arnoldi algorithm as part of the start-block $\begin{bmatrix} Y & \mathbf{R} \end{bmatrix}$. We will address forcing deflation of the first ℓ residual-candidate vectors and why we do it.

5.2.1 Explicitly thick-starting with Ritz-vectors

Suppose we identified ℓ Ritz-pairs (λ_j, z_j) with residuals γ_j so that

$$\mathbf{H}z_j - \lambda z_j = \gamma_j \tag{90}$$

for $j = 1, 2, ..., \ell$. If we run the band-Arnoldi algorithm with $\mathbf{H}$ and start-block $\begin{bmatrix} Z & \mathbf{R} \end{bmatrix}$ where $Z = \begin{bmatrix} z_1 & z_2 & \cdots & z_\ell \end{bmatrix}$ and $\mathbf{R} = \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \cdots & \mathbf{r}_m \end{bmatrix}$, then assuming no linear-dependence within the start block, ℓ iterations yields an $\ell \times \ell$ upper-triangular coefficient matrix U, and $Y = \begin{bmatrix} y_1 & y_2 & \cdots & y_\ell \end{bmatrix}$ is an orthonormal basis for span Z where

$$u_{jj} y_j = (I - Y_{j-1} Y_{j-1}^H) z_j$$

for $j = 1, 2, \ldots, \ell$, and $u_{jj} = \lambda_j$ is the diagonal element of U. Note that Y and U are just different names for the quantities we defined previously as V and $\widetilde{\mathbf{H}}$, but constructed after only ℓ iterations of the "thick" start-block $\begin{bmatrix} Z & \mathbf{R} \end{bmatrix}$. $Z = YU$ can be regarded as a QR factorization. Assuming unit Ritz-vectors, the next ℓ candidate-vectors are

$$\begin{aligned}
\hat{v}_{\ell+m+j} &= (I - Y_j Y_j^H)\mathbf{H} v_j \\
&= (I - Y_j Y_j^H)\mathbf{H}(I - Y_{j-1} Y_{j-1}^H) z_j \\
&= (I - Y_j Y_j^H)\mathbf{H} z_j, \quad \text{because } \mathbf{H} Y_{j-1} \subset \operatorname{span} Y_j \\
&= (I - Y_j Y_j^H)(\lambda_j z_j + \gamma_j) \\
&= (I - Y_j Y_j^H)\gamma_j \qquad \text{because } z_j \in \operatorname{span} Y_j.
\end{aligned} \tag{91}$$

$$\widetilde{\mathbf{H}} = \begin{bmatrix} h & h & h & h & h & h & h & h & h & h \\ h & h & h & h & h & h & h & h & h & h \\ h & h & h & h & h & h & h & h & h & h \\ & h & h & h & h & h & h & h & h & h \\ & & h & h & h & h & h & h & h & h \\ & & & h & h & h & h & h & h & h \\ & & & & h & h & h & h & h & h \\ & & & & & h & h & h & h & h \\ & & & & & & h & h & h & h \\ & & & & & & & h & h & h \end{bmatrix}$$

Table 5.2: The Rayleigh-quotient produced by 10 iterations of standard band-Arnoldi with band-size $m = 2$. The triangular region below the $(m+1)$-th subdiagonal is zero. For $m = 1$, $\widetilde{\mathbf{H}}$ would be strictly upper-Hessenberg.

for $j = 1, 2, ..., \ell$. Then the $\ell + m$ band of candidate-vectors at that point is

$$\left[(I - Y_\ell Y_\ell^H)\mathbf{R} \quad \hat{v}_{\ell+m+1} \quad \hat{v}_{\ell+m+2} \quad \dots \quad \hat{v}_{2\ell+m} \right]. \tag{92}$$

Note that $(I - Y_j Y_j^H)\gamma_j = \gamma_j$ if the residual γ_j is orthogonal to $\{z_1, z_2, ..., z_j\}$, which is the case when the basis Z of Ritz-vectors came from one cycle of a Krylov process such as algorithm 2. If we consider restarts however, the Ritz-vectors from each restart will have a different orthogonal residual, so the set of Ritz-vectors will not be orthogonal to the set of residuals in general.

If the Ritz-pairs (90) have "deflatable" residuals, i.e. $\|\gamma_j\| < \text{dtol}$, then the associated candidate vectors (91) will be near-zero and will get deflated when we continue iterating on step $\ell + 1$.

5.2.2 A thick-restart example

In table 5.2 we give a structural-example of a coefficient matrix from a $n = 10$ iteration cycle of Band-Arnoldi with band-size $m = 2$, followed by a thick-restarted cycle in tables 5.3 and 5.4. The $\mathbf{H}$ and $\mathbf{R}$ used to produce it were random matrices with $N = 100$, but the reader should note that N is not important.

After after identifying 2 (or 3) Ritz-vectors of $\widetilde{\mathbf{H}}$ we restarted the process, augmented by those Ritz-vectors. We deliberately chose Ritz-vectors that were not converged so that the residuals were too large to be deflated naturally.

$$\begin{bmatrix}
u & u & g & g & g & g & g & g & g & g \\
 & u & g & g & g & g & g & g & g & g \\
 & & h & h & h & h & h & h & h & h \\
 & & h & h & h & h & h & h & h & h \\
y & y & h & h & h & h & h & h & h & h \\
 & y & h & h & h & h & h & h & h & h \\
 & & h & h & h & h & h & h & h & h \\
 & & & h & h & h & h & h & h & h \\
 & & & & h & h & h & h & h & h \\
 & & & & & h & h & h & h & h
\end{bmatrix}$$

(a) cycle 2: natural deflation

$$\begin{bmatrix}
u & u & g & g & g & g & g & g & g & g \\
 & u & g & g & g & g & g & g & g & g \\
 & & h & h & h & h & h & h & h & h \\
 & & h & h & h & h & h & h & h & h \\
 & & h & h & h & h & h & h & h & h \\
 & & & h & h & h & h & h & h & h \\
 & & & & h & h & h & h & h & h \\
 & & & & & h & h & h & h & h \\
 & & & & & & h & h & h & h \\
 & & & & & & & h & h & h
\end{bmatrix}$$

(b) cycle 2: forced deflation

Table 5.3: These are both taken after re-starting the cycle from table 5.2, with 2 Ritz-vectors. No deflation occurred in (a) because the Ritz vectors were not sufficiently converged, so it is similar in structure to a standard cycle with $m = 2 + 2$. In contrast, for (b) we forced deflation of all Ritz-residuals. (a) is a proper Rayleigh-quotient but (b) is not. The advantage to (b) is that is cheaper to produce and permits a Krylov-Schur-type implicit-restart.

$$\begin{bmatrix}
u & u & u & g & g & g & g & g & g & g \\
 & u & u & g & g & g & g & g & g & g \\
 & & u & g & g & g & g & g & g & g \\
 & & & h & h & h & h & h & h & h \\
 & & & h & h & h & h & h & h & h \\
y & y & y & h & h & h & h & h & h & h \\
 & y & y & h & h & h & h & h & h & h \\
 & & \varepsilon & h & h & h & h & h & h & h \\
 & & \varepsilon & 0 & h & h & h & h & h & h \\
 & & \varepsilon & & 0 & h & h & h & h & h
\end{bmatrix}$$

(a) natural deflation

$$\begin{bmatrix}
u & u & u & g & g & g & g & g & g & g \\
 & u & u & g & g & g & g & g & g & g \\
 & & u & g & g & g & g & g & g & g \\
 & & & h & h & h & h & h & h & h \\
 & & & h & h & h & h & h & h & h \\
 & & & h & h & h & h & h & h & h \\
 & & & & h & h & h & h & h & h \\
 & & & & & h & h & h & h & h \\
 & & & & & & h & h & h & h \\
 & & & & & & & h & h & h
\end{bmatrix}$$

(b) forced deflation

Table 5.4: This is a restarted-cycle following 5.2, similar to table 5.3 except that we kept 3 (unconverged) Ritz-vectors instead of 2. Since the band-size of the first cycle was $m = 2$, the Ritz-residual had rank-2 so one residual-vector was deflated on iteration 8, which we indicated by leaving those zero-entries visible. After a candidate vector is removed from iterations via inexact-deflation, band-Arnoldi continues to orthogonalize new iterates against it, indicated by the ε entries which are smaller than the deflation-tolerance dtol and may be considered negligible. There are $\dim \hat{Y} \leq \ell$ rows of 'y' entries in (a) that reflect orthogonalization of V against the residual $\hat{Y}$. In the forced-deflation, all three residuals were manually deflated on step 6.

5.2.3 Thick-restart with natural deflation of iterates

For this discussion there is no loss of generality to assume we augment the start block with an already-orthogonal basis Y for the subspace spanned by the kept Ritz-vectors, so that the Krylov-Schur relation associated with Y under $\mathbf{H}$ is

$$\mathbf{H}Y = YU + \hat{Y}B, \tag{93}$$

Letting the process proceed without intervention (naturally) for $n + \ell$ iterations, band-Arnoldi returns basis $\widetilde{V} = \begin{bmatrix} Y & V \end{bmatrix}$ and Rayleigh-quotient $\widetilde{\mathbf{H}} = \widetilde{V}^H \mathbf{H} \widetilde{V}$, that satisfy

$$\mathbf{H} \begin{bmatrix} Y & V \end{bmatrix} = \begin{bmatrix} Y & V \end{bmatrix} \underbrace{\begin{bmatrix} \tilde{U} & G \\ \mathcal{E} & \hat{H} \end{bmatrix}}_{\widetilde{\mathbf{H}}} + \begin{bmatrix} (I - YY^H)\hat{Y} & \\ (I - VV^H)\hat{Y} & \hat{\dot{V}} \end{bmatrix} \begin{bmatrix} B & \\ & F \end{bmatrix} \tag{94}$$

where $\hat{\dot{V}}$, and F are defined as in (80). The blocks

$$\tilde{U} = U + Y^H \hat{Y} \tag{95}$$

and

$$\mathcal{E} = V^H \hat{Y} B \tag{96}$$

represent operations that must be carried out by band-Arnoldi in order for (94) to be orthogonal, i.e. in order for left-multiplication by $\widetilde{V} = \begin{bmatrix} Y & V \end{bmatrix}$ to yield $\widetilde{\mathbf{H}} = \widetilde{V}^H \mathbf{H} \widetilde{V}$. This is because, unlike a typical Arnoldi process, the residual $\hat{Y}$ is not included in the basis $\begin{bmatrix} Y & V \end{bmatrix}$.

Examples of $\widetilde{\mathbf{H}}$ are given in tables 5.3a and 5.4a, except they might be misleading because in both of those examples, the restart is on the second cycle, so $\tilde{U} = U$ is upper triangular. This will not be true in a general multiple-restart context.

For a basis of Y Ritz-vectors from one cycle of standard band-Arnoldi, the residual $\hat{Y}$ is orthogonal to Y (as in example tables 5.3a and 5.4a), but since every cycle produces a different residual, we can expect rank $\hat{Y} \leq \ell = \dim Y$ and $\hat{Y}$ not orthogonal to Y in general. Then the candidate vectors

corresponding to $\hat{Y}$ are made orthogonal to Y as they are processed, resulting in the translation

$$\mathbf{H}Y = Y\tilde{U} + (I - YY^H)\hat{Y}B \tag{97}$$

of (93).

Similarly, $\mathcal{E}$ (96) has rank $\hat{Y}$ nonzero rows and comes from orthogonalizing iterates v_i against span $\hat{Y}$.

In the thick-restart [44] and Krylov-Schur [46] schemes for single-vector iterations, they assume $\|\mathcal{E}\| \leq \|\hat{Y}\|$ is small enough to be negligible, so that $U' \approx U$ is upper-triangular. Also, they restart with the residual from a previous cycle, which in our case would be $\hat{Y}$. In that case $\hat{Y}$ would be included in V and $\mathcal{E}$ would be zero. Those methods then take advantage of the upper-triangular structure of the leading $\ell \times \ell$ principle sub-matrix U. We can mimic this somewhat by forcing deflation of $\hat{Y}$, which is discussed in the next section.

5.2.4 Forced deflation of iterates

Rather than let band-Arnoldi naturally deflate the residuals, our thick-restart manually deflates them. What that means is, after the first ℓ iterations of band-Arnoldi starting with $\begin{bmatrix} Y & \mathbf{R} \end{bmatrix}$, we set to zero the candidate vectors

$$\hat{v}_{\ell+m+1}, \hat{v}_{\ell+m+2}, \ldots, \hat{v}_{2\ell+m} := 0$$

of (92). We treat Y as an exactly-invariant subspace basis. In that case, a total of $\ell + n$ iterations yields

$$\mathbf{H}\begin{bmatrix} Y & V \end{bmatrix} = \begin{bmatrix} Y & V \end{bmatrix}\begin{bmatrix} U & G \\ & \hat{H} \end{bmatrix} + \begin{bmatrix} \hat{Y}B & \hat{\hat{V}}F \end{bmatrix} \tag{98}$$

5.2.5 Determining poles of a thick-restarted ROM

Taking a full or partial similarity decomposition $\widetilde{\mathbf{H}}W = W\Lambda$, the associated block $Z = \begin{bmatrix} Y & V \end{bmatrix} W$ has residual

$$\mathbf{H}Z - Z\Lambda = \begin{bmatrix} (I - YY^H)\hat{Y} & & \\ (I - VV^H)\hat{Y} & (I - VV^H)\dot{V} & \hat{V} \end{bmatrix} \begin{bmatrix} B & \\ & F_1 \\ & F_2 \end{bmatrix} W.$$

Let $\hat{y}^T$, $\dot{v}^T$, and $\hat{v}^T$ be the row vectors of norms of the columns of $\hat{Y}$, $\dot{V}$, and $\hat{V}$, respectively. For example,

$$\hat{y}^T = \begin{bmatrix} \|\hat{y}_1\| & \|\hat{y}_2\| & \cdots & \|\hat{y}_\nu\| \end{bmatrix}.$$

Let

$$f^T := \begin{bmatrix} \hat{y}^T & \dot{v}^T & \hat{v}^T \end{bmatrix} \begin{bmatrix} B & \\ & F_1 \\ & F_2 \end{bmatrix}.$$

For many applications $\|\dot{V}\|_F \leq \text{dtol}\sqrt{d}$ is negligible, in which case $f^T \approx \begin{bmatrix} \hat{y}^T B & \hat{v}^T F_2 \end{bmatrix}$. The residual-norm $\|\hat{Y}\|$ of Y is small in the sense that it represents a nearly-invariant-subspace to some degree, but it is not negligible in general.

For an individual Ritz-pair $z_j = \begin{bmatrix} Y & V \end{bmatrix} w_j \in Z_2$ and $\lambda_j \in \Lambda$, a bound for residual-norm is

$$\|\mathbf{H}z_j - \lambda_j z_j\| \leq |f^T w_j|. \tag{99}$$

Note that

$$|f^T w_j| \geq \|\|\hat{Y}\|,$$

so our residual-norm estimate can never be better than $\|\hat{Y}\|$. We consider Ritz-vector z_j to be converged if the relative residual-norm

$$\text{rr}_j = \frac{|f^T w_j|}{\|\mathbf{H}\|_{\text{est}}} \leq \text{ctol}. \tag{100}$$

$\|\mathbf{H}\|_{\text{est}} = \max_v \|\mathbf{H}v\|$ is an estimated operator-norm of $\mathbf{H}$ obtained during iterations. A value suggested by [37] is $\text{ctol} = \sqrt{\epsilon}$, which is the same value used for dtol in [17].

The bound used by ARPACK suggests that (λ_j, z_j) is converged if

$$|f^T w_j| \leq \max\{\epsilon_M \|\widetilde{\mathbf{H}}\|, \text{ctol} \cdot |\lambda|\}.$$

5.2.6 Intermediate ROM from a thick-restarted band-Arnoldi process

The following describes how we can form an intermediate ROM via implicit-projection cheaply in order to observe convergence of the model.

Recall that the band-Arnoldi algorithm

$$\begin{bmatrix} V & \widetilde{\mathbf{H}} & \widetilde{\boldsymbol{\rho}} \end{bmatrix} = \mathbf{bArnoldi}(\mathbf{H}, \mathbf{R}) \tag{101}$$

where $\widetilde{\mathbf{H}} = V^H \mathbf{H} V$ and $\widetilde{\boldsymbol{\rho}} = V^H \mathbf{R}$, is the ROM approximation via implicit-projection

$$\widetilde{\mathcal{H}}(s) = (\boldsymbol{C}^T V)\big(I - (s-\sigma)\widetilde{\mathbf{H}}\big)^{-1} \underbrace{(V^H \mathbf{R})}_{\widetilde{\boldsymbol{\rho}}} \tag{102}$$

to the URM transfer-function

$$\mathcal{H}(s) = \boldsymbol{C}^T \big(I - (s-\sigma)\mathbf{H}\big)^{-1} \mathbf{R}.$$

For simplicity let us assume we will augment, or "thicken" the start block of the band-Arnoldi process with a basis Y for an *exactly* $\mathbf{H}$-invariant-subspace, so that $\mathbf{H}Y = YU$.

Then the thick-started process

$$\begin{bmatrix} V & \widetilde{\mathbf{H}} & \widetilde{\boldsymbol{\rho}} \end{bmatrix} = \mathbf{bArnoldi}(\mathbf{H}, \begin{bmatrix} Y & \mathbf{R} \end{bmatrix}) \tag{103}$$

yields

$$V = \begin{bmatrix} Y & V' \end{bmatrix}, \quad \widetilde{\mathbf{H}} = \begin{bmatrix} U & G \\ & \widetilde{\mathbf{H}}' \end{bmatrix}, \quad \text{and} \quad \widetilde{\boldsymbol{\rho}} = \begin{bmatrix} Y & V' \end{bmatrix}^H \begin{bmatrix} Y & \mathbf{R} \end{bmatrix} = \begin{bmatrix} \widetilde{\boldsymbol{\rho}}_1 & \widetilde{\boldsymbol{\rho}}_2 \end{bmatrix}$$

Note that $\widetilde{\boldsymbol{\rho}}_2 = \begin{bmatrix} Y & V' \end{bmatrix}^H \mathbf{R}$, so the implied ROM transfer-function

$$\widetilde{\mathcal{H}}(s) = \left(\boldsymbol{C}^T \begin{bmatrix} Y & V' \end{bmatrix}\right) \left(I - (s-\sigma)\widetilde{\mathbf{H}}\right)^{-1} \underbrace{\left(\begin{bmatrix} Y & V' \end{bmatrix}^H \mathbf{R}\right)}_{\widetilde{\boldsymbol{\rho}}_2} \tag{104}$$

makes use of only $\widetilde{\boldsymbol{\rho}}_2$ rather than all of $\widetilde{\boldsymbol{\rho}}$, and $\widetilde{\boldsymbol{\rho}}_1 = \begin{bmatrix} I \\ 0 \end{bmatrix}$ is left out.

5.3 Proposing a new model-reduction method

Our restarted ROM method can described as a multiple-shift method with invariant-subspace recycling. We construct a basis for a Krylov-subspaces at a (potentially) different shift on every cycle. The thick-restart mechanism allows us to hold on to a growing basis Y of known converged $(\boldsymbol{A},\boldsymbol{E})$-invariant-subspace, and continue orthogonalizing new basis vectors against Y. Assuming $Y_0 = \{\}$ and $\ell_j = \dim Y_{j-1}$, the general process is

1. For $j = 1, 2, \ldots, \tau$, use Krylov-operator $\mathbf{H}_j = (\boldsymbol{A} - \sigma_j \boldsymbol{E})^{-1}\boldsymbol{E}$ and operand $\mathbf{R}_j = (\sigma_j \boldsymbol{E} - \boldsymbol{A})^{-1}\boldsymbol{B}$.

   (a) **Expand** Y_{j-1} into basis $\begin{bmatrix} Y_{j-1} & V_j \end{bmatrix}$ for $\mathcal{Y}_{j-1} \cup \mathcal{K}_{n_j}(\mathbf{H}_j, \mathbf{R}_j)$ via $\ell_j + m$ iterations of thick-start Arnoldi process.

   (b) The computed quantities $\widetilde{\boldsymbol{\rho}}_2 = \widetilde{V}_j^H \mathbf{R}_j$ and $\widetilde{\mathbf{H}} = \widetilde{V}_j^H \mathbf{H}_j \widetilde{V}_j$ can be used to **observe** the ROM transfer-function

   $$\widetilde{\mathcal{H}}_j(s) = \boldsymbol{C}^T \widetilde{V}_j \big(I - (s - \sigma_j)\widetilde{\mathbf{H}}\big)^{-1} \widetilde{\boldsymbol{\rho}}_2$$

   of (104) via implicit-projection on to the basis $\widetilde{V}_j = \begin{bmatrix} Y_{j-1} & V_j \end{bmatrix}$.

   (c) **Deflate** $\begin{bmatrix} Y_{j-1} & V_j \end{bmatrix}$ into $\begin{bmatrix} Y_{j-1} & Y_j' \end{bmatrix}$, where $Y_j' \subset \operatorname{span} V_j$ is newly converged $(\boldsymbol{A},\boldsymbol{E})$-invariant-subspace, and set $Y_j = \begin{bmatrix} Y_{j-1} & Y_j' \end{bmatrix}$.

2. Now we have a set of orthogonal blocks $\{V_1, V_2, ..., V_\tau\}$ (but not orthogonal to each other) such that

   $$\operatorname{span} \begin{bmatrix} V_1 & V_2 & \cdots & V_\tau \end{bmatrix} = \bigcup_{j=1}^{\tau} \mathcal{K}_{n_j}(\mathbf{H}_j, \mathbf{R}_j).$$

   Let $\widehat{V} = \operatorname{span} \begin{bmatrix} V_1 & V_2 & \cdots & V_\tau \end{bmatrix} \in \mathbb{R}^{N \times n}$ (most-likely split into $\Re$ and $\Im$ parts and re-orthogonalized). The explicitly-projected ROM realization is then

   $$\boldsymbol{A}_n = \widehat{V}^T \boldsymbol{A} \widehat{V}, \quad \boldsymbol{E}_n = \widehat{V}^T \boldsymbol{E} \widehat{V}, \quad \boldsymbol{B}_n = \widehat{V}^T \boldsymbol{B}, \quad \boldsymbol{C}_n = \widehat{V}^T \boldsymbol{C},$$

and it has transfer-function

$$\widehat{\mathcal{H}}(s) = \boldsymbol{C}_n^T(\boldsymbol{A}_n - s\boldsymbol{E}_n)^{-1}\boldsymbol{B}_n.$$

Step 1 of the above outline is given as algorithm 3.

Algorithm 3: EXPLCIT THICK-RESTARTED BAND-ARNOLDI CYCLE

Input: System realization $(\boldsymbol{A}, \boldsymbol{E}, \boldsymbol{B}, \boldsymbol{C})$, initial interpolation-point $\sigma_1 \in \mathbb{C}$.

1 Set $Y_0 = \{\}, V_{\text{ROM}} = \{\}$, $m := \dim \boldsymbol{B}$
2 **for** $j = 1, 2, \ldots$ **do**
3 Set $\ell_j := \dim Y_{j-1}$
4 Let $\mathbf{H}_j := (\boldsymbol{A} - \sigma_j \boldsymbol{E})^{-1}\boldsymbol{E}$ and $\mathbf{R}_j := (\sigma_j \boldsymbol{E} - \boldsymbol{A})^{-1}\boldsymbol{B}$
5 Compute $\left(V, \hat{V}, \dot{V}, \widetilde{\mathbf{H}}, \widetilde{\boldsymbol{\rho}}\right) := \textbf{bArnoldi}\left(1 : \ell_j, \mathbf{H}_j, \begin{bmatrix} Y_{j-1} & \mathbf{R}_j \end{bmatrix}\right)$.
`% manually set candidates resulting from processing` Y_{j-1}`, to zero.`
6 Set: $\hat{v}_i := 0$, for $i = \ell_j + m + (1, 2, ..., \ell_j)$
7 Continue $\left(V, \hat{V}, \dot{V}, \widetilde{\mathbf{H}}, \widetilde{\boldsymbol{\rho}}\right) := \textbf{bArnoldi}\left(\ell_j + 1 : n_j, \mathbf{H}_j, \begin{bmatrix} Y_{j-1} & \mathbf{R}_j \end{bmatrix}\right)$.
8 Set $V_{\text{ROM}} := \begin{bmatrix} V_{\text{ROM}} & V_{\mathbf{R}_j} \end{bmatrix}$, where $V = \begin{bmatrix} v_1 & v_2 & \cdots & v_\ell & V_{\mathbf{R}_j} \end{bmatrix}$.
`% The` j`-th implicitly projected ROM transfer-function is given by` (104).
9 Take eigen-decomposition $\widetilde{\mathbf{H}}W = W\Lambda$. The corresponding poles are $\mu_i = \sigma_j + 1/\lambda_i$.
Convergence of a Ritz-pair (λ_i, z_i) where $z_i = \begin{bmatrix} Y & V \end{bmatrix} w_i$ is given by (89).
10 Compute pole-weights $\gamma_1, \gamma_2, \ldots, \gamma_{n_j}$ as (24) and (25).
11 Let Z_j consist of converged Ritz-vectors and those with large relative-weight $|\gamma_i|/\Sigma|\gamma_i|$.
12 Let $(Y_j, T_j) := QR(\begin{bmatrix} Y_{j-1} & Z_j \end{bmatrix})$
13 Select new interpolation-point σ_{j+1}.

Output: Basis V_{ROM} for $\bigcup_j \mathcal{K}_{n_j}(\mathbf{H}_j, \mathbf{R}_j)$.

The explicit thick-restarted Band-Arnoldi algorithm is given as algorithm 3. It consists of restarting the band-Arnoldi algorithm (algorithm 2) with a basis of Ritz-vectors and setting to zero the candidate vectors resulting from processing those Ritz-vectors. We experimented with allowing the Ritz-vectors to be processed normally, but it requires more computation and generally resulted in a less accurate ROM for a given size. In practice (for large N), we would not process Y_{j-1} explicitly (perform steps 5 and 6 of algorithm 2). We would perform an implicit-restart method like Krylov-Schur[46], by pre-loading V with the already orthogonal-basis Y_{j-1} and $\widetilde{\mathbf{H}}$ with $U_{j-1} = T_{j-1}\Lambda_j T_{j-1}^{-1}$.

Selection of a new interpolation-point (line 13) is left up to whatever method the user chooses; given that we have fairly cheap access to pole distribution data for the implicit ROM transfer-function at any iteration, we assume a point-selection method will take advantage of that. An example of a simple adaptive method is to choose σ_{j+1} to be close to the location of the un-converged pole with largest weight. That would be something like

$$\sigma_{j+1} = \Im(\mu_\tau)$$

where $\tau = \text{argmax}_i|\gamma_i|$ the un-converged pole with largest weight.

5.4 Results

Due to time constraints we limit our numerical results to a few examples of the method at work on two test models. `ex308` and `ex1841` approximated at a single interpolation-point. We recorded the number of iterations of bArnoldi required to reach a relative transfer-function error

$$\frac{\|\mathcal{H} - \widetilde{\mathcal{H}}\|}{\|\mathcal{H}\|} \leq \texttt{tf\_tol} = 0.01, \tag{105}$$

at each of three points that are canonical in some sense. Those are a real point $\pi 10^{10}$, the $\Im$-point $\pi i 10^{10}$ located roughly at the midpoint of the segment of interest, and the complex point $(1+i)\pi 10^{10}$, shown in figure 5.1. The resulting ROM size in each case depends on the dimension of the split-Krylov-subspace

$$\mathcal{K}_{n'}(\mathbf{H}, \mathbf{R})^* = \mathcal{K}_n(\mathbf{H}, \mathbf{R}) \cup \mathcal{K}_n(\overline{\mathbf{H}}, \overline{\mathbf{R}}),$$

so the dimension n' of the ROM explicitly-projected onto a real basis is no larger than n. If no deflation occurred during re-orthogonalization of conjugate parts of the projection basis, then $n' = 2n$ and that was typical for our experiments.

We will give a count of floating-point operations (flops) for producing ROMS. We consider flop-counts to be scalar products in $\mathbb{R}^n$, so when complex arithmetic is being used ($\sigma \notin \mathbb{R}$) we must multiply the count by 4. Band-Arnoldi run for n-iterations with a constant band-size of $m_c = m$ requires approximately

$$\texttt{bA\_count}(n) = nN^2 + N(n)(n-1)/2 + Nmn$$

flops. That is nN^2 flops for n-matrix-vector products, $N(n)(n-1)/2$ flops for orthogonalization of $1 + 2 + \cdots + n$ basis vectors, and Nmn flops for orthogonalization of m candidate-vectors at each iteration.

We include the flop count for tests because we wish to reduce this number using restarts, even if the ROM dimension itself is not appreciably smaller. We would like that for l cycles of

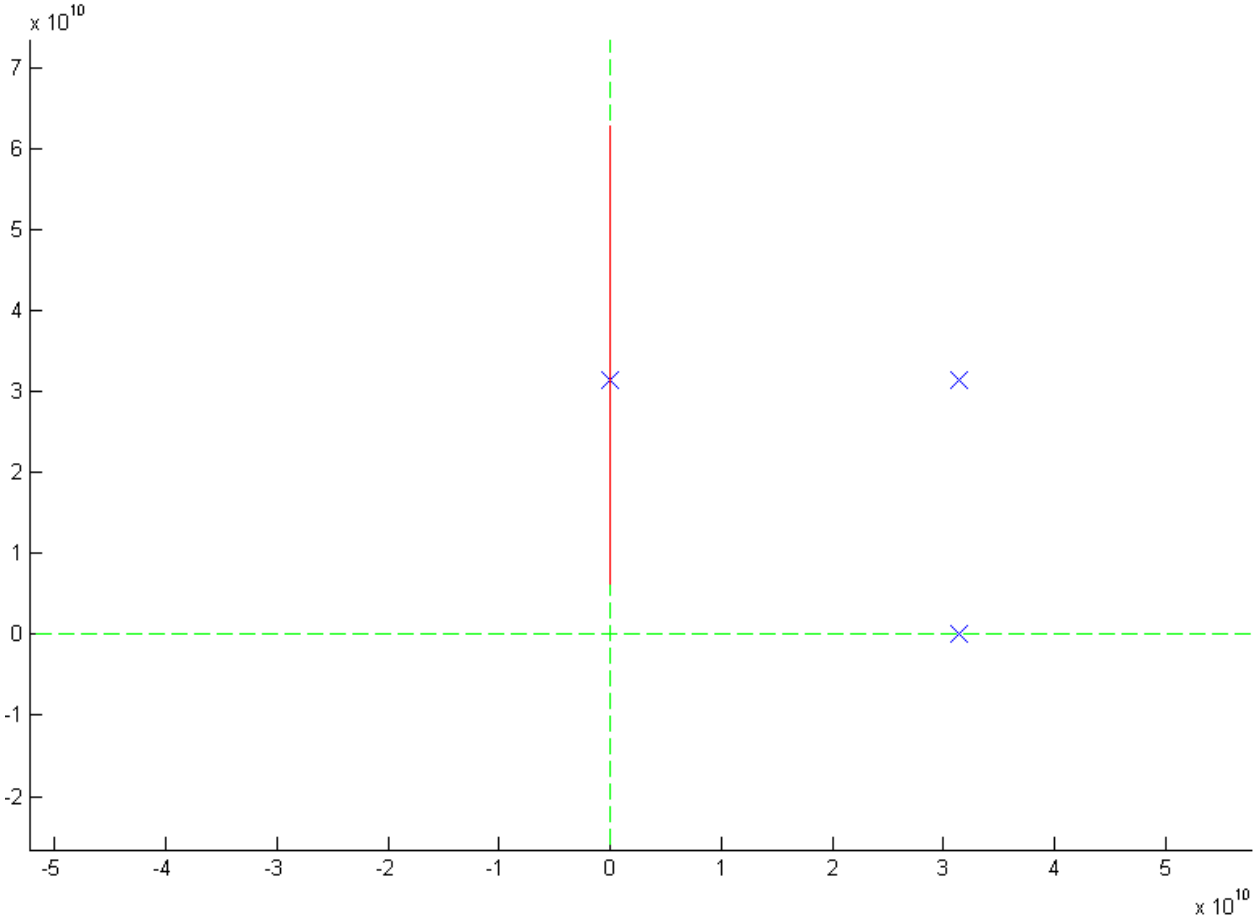


Figure 5.1: The three interpolation-points used for single point benchmarks. The segment of interest $[10^9, 10^{10}]i$ on the $\Im$-axis, is highlighted. Note how small $[0, 10^9]$ is, in comparison.

H

σ	iterations (n)	ROM size (n')	LI	rel-err	flops	figure
$\pi 10^{10}$	144	144	1	7.368e-05	16,920,288 + M	5.4
$i\pi 10^{10}$	71	142	0.992958	5.913e-4	30,177,840 + M	5.5
$(1+i)\pi 10^{10}$	97	194	0.793814	5.1713e-3	42,782,432 + M	5.6

Table 5.5: Benchmark data for `ex308`. flops is a count of real (in $\mathbb{R}$), non-zero scalar products required for matrix-vector multiplication and inner-products.

band-Arnoldi, each run for n_j iterations,

$$lM + \sum_j \texttt{bA\_count}(n_j) < M + \texttt{bA\_count}\left(\sum n_j\right)$$

M represents the cost of factoring or (re)forming $\mathbf{H}_j$ and $\mathbf{R}_j$ which, for a restarted method, must be done l times (for each $j = 1, 2, ..., l$). It only needs to be done once for a single-point method. We do not have a value for M because it varies with the application. It may be negligibly small or prohibitively large, and depends on the size and sparsity of the model realization $(\boldsymbol{A}, \boldsymbol{E}, \boldsymbol{B}, \boldsymbol{C})$.

5.4.1 ex308

`ex308` is a 2×2 MIMO model of a RCL circuit with 2 input and 2 output terminals, that comes from from a test problem for PEEC modelling of interconnect from IBM or Carnegie Mellon University. `ex308` is characterized by many poles very near the $\Im$-axis, giving its transfer-function gain a spiked appearance.

ex308 Benchmarks

Benchmark data for `ex308` is given in table 5.5.

ROM size (projection basis dimension) is given as the dimension n' of the real basis V_{ROM} obtained by splitting and re-orthogonalizing $\widehat{V}$. "LI" is a linear-independence measure defined as

$$\text{LI}\left(V_{\text{ROM}}\right) = \frac{\text{rank}_{\text{eff}}\left(V_{\text{ROM}}\right)}{n}$$

where rank_{eff} is the "effective-rank" of V_{ROM} as determined by Matlab. We expect the restarted method to produce a less "effectively" linearly-independent basis.

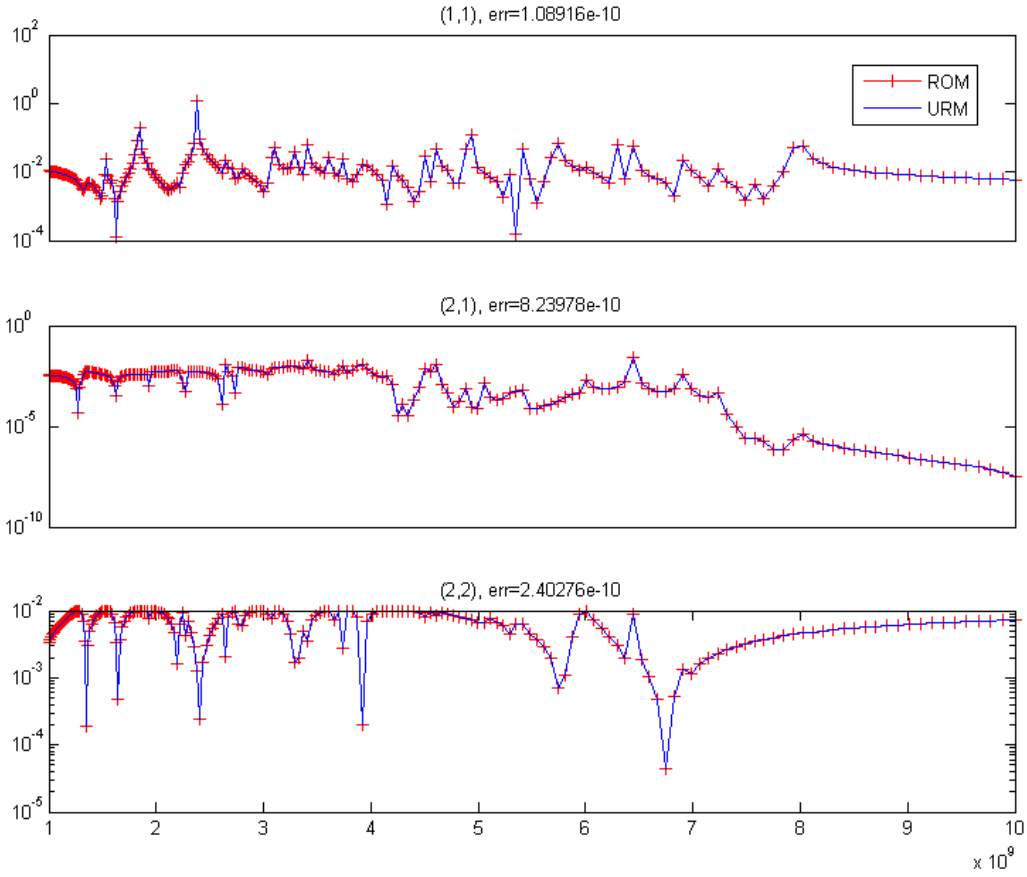


Figure 5.2: These are the three unique gain plots for `ex308`.

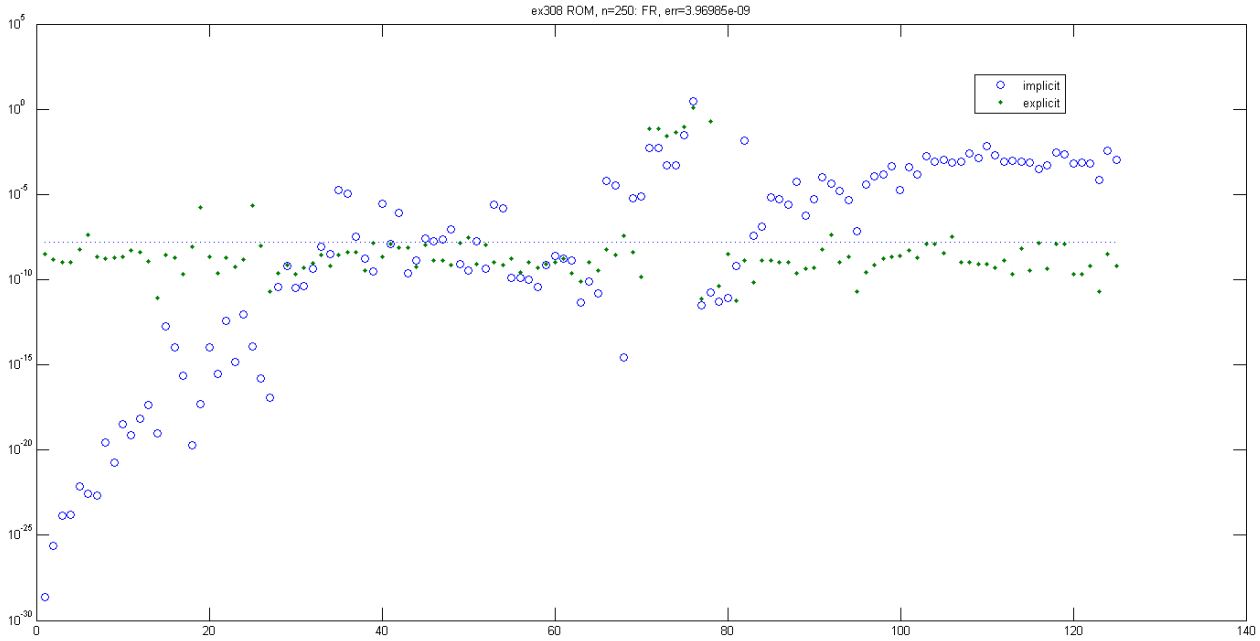


Figure 5.3: This is a plot of relative-residual errors for the 250 poles of an $n = 250$ ROM (about $\sigma = (1+i)\pi 10^{10}$) of `ex308`. The circles are the poles derived from eigenvalues of $\widetilde{\mathbf{H}} = V^H \mathbf{H} V$ (the implicit ROM), and the dots are the eigenvalues of the explicitly-projected matrix pencil $(\boldsymbol{A}_n, \boldsymbol{E}_n) = (V^H \boldsymbol{A} V, V^H \boldsymbol{E} V)$ (the explicit ROM). These are different sets of poles for the most part, except that they converge to the same set of eigenvalues of $(\boldsymbol{A}, \boldsymbol{E})$ as n increases. We can expect the two ROMs to share *converged* poles. In practice, only the implicit ROM poles (the circles) will be available because relative residual norms are cheap to compute for Ritz-values from $\widetilde{\mathbf{H}}$. Computing eigen-pairs (μ, z) of $(\boldsymbol{A}_n, \boldsymbol{E}_n)$ would require an expensive explicit-projection and there is no cheap formula like (99) for the residual norm $\|\boldsymbol{A}z - \mu \boldsymbol{E}z\|$.

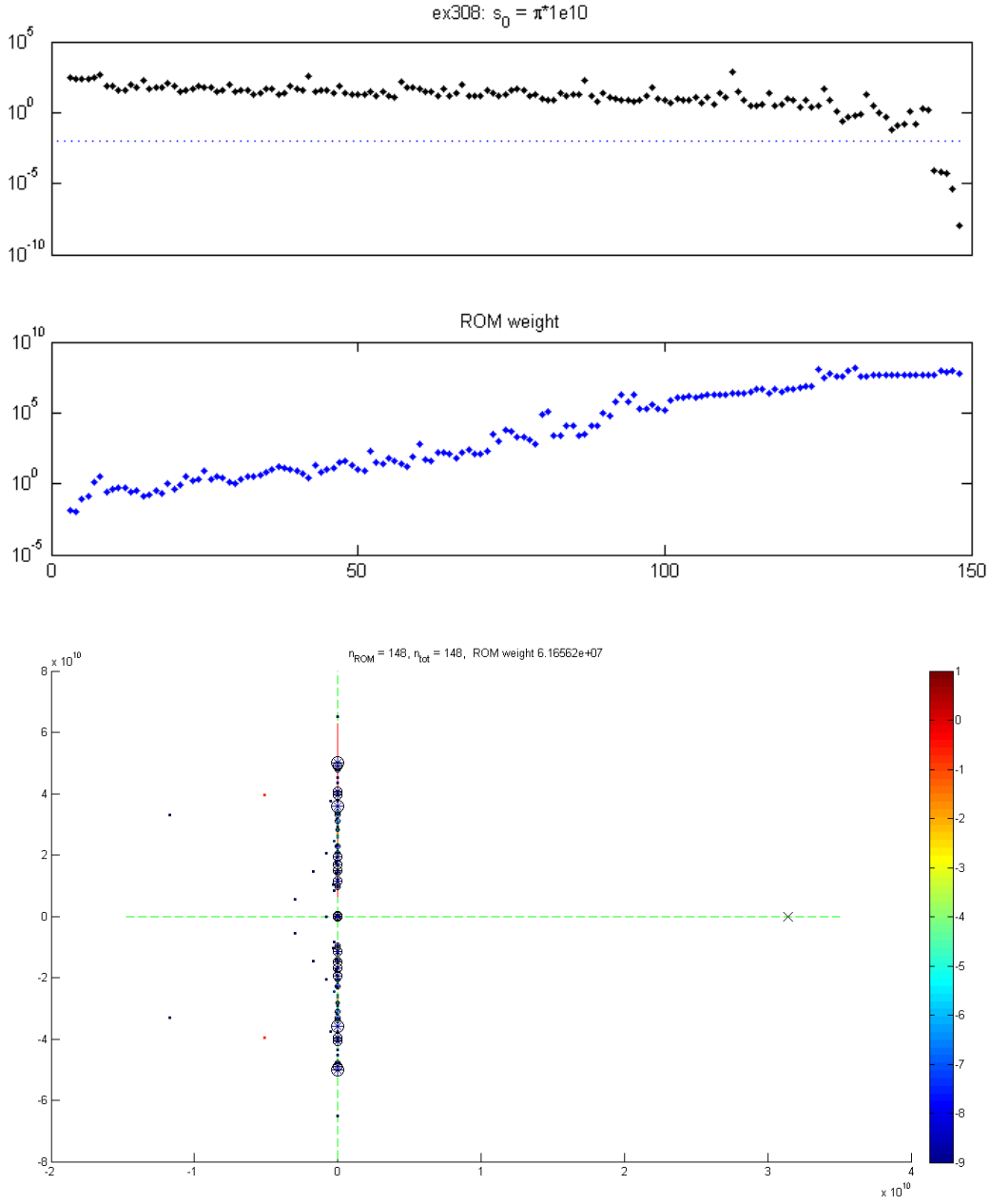


Figure 5.4: Transfer-function relative-error (105) (of explicitly-projected ROM) and ROM weight (of the implicitly projected ROM) vs. n for `ex308`, at real interpolation-point $s_0 = \pi 10^{10} \in \mathbb{R}$, indicated by '$\times$'. The dotted line in the first plot represents a relative-error of 0.01. `ex308` is characterized by a dense distribution of poles on or very near the $\Im$-axis, which is evident from the pole distribution of the implicitly defined ROM transfer-function at 148 iterations. That particular ($n = 148$) ROM implies a ROM via *explicit* projection with relative transfer-function error $\approx 10^{-8}$.

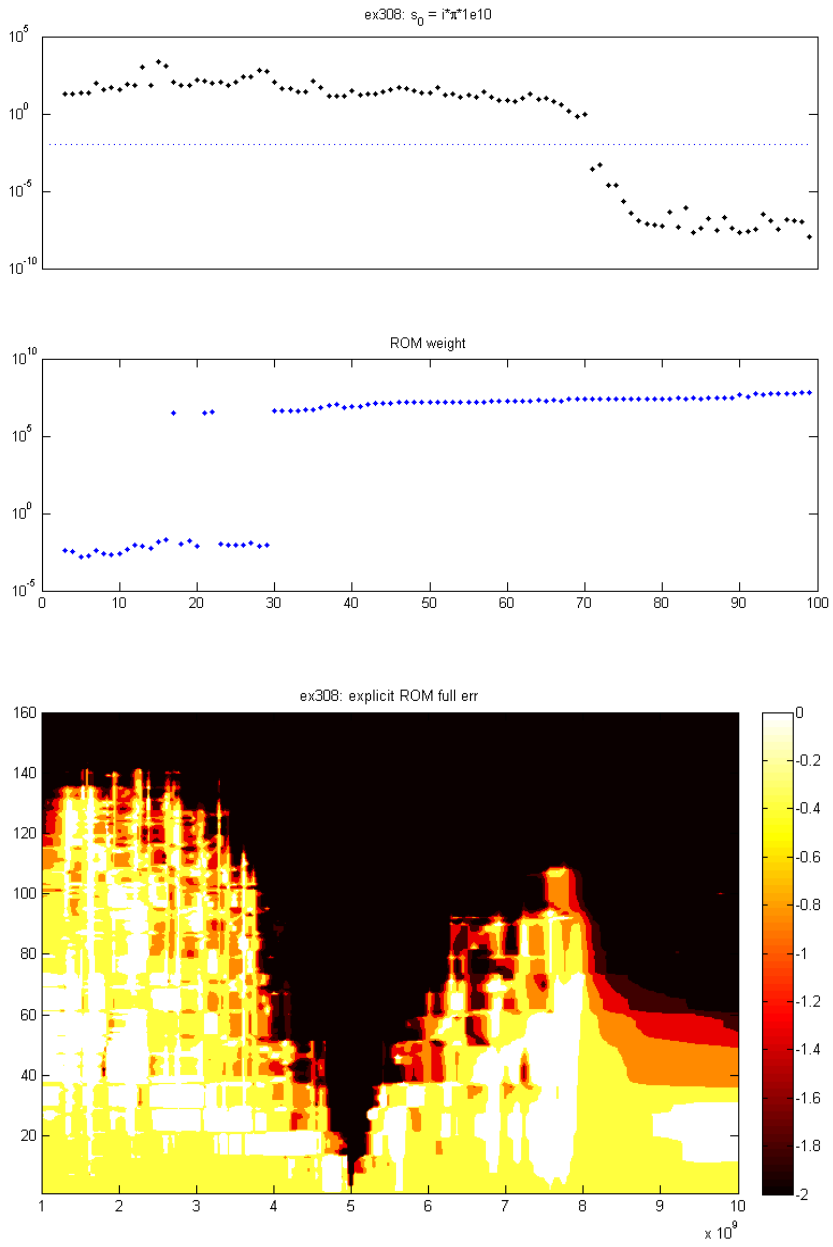


Figure 5.5: Transfer-function relative-error and ROM weight vs. n for `ex308`, at $\Im$ interpolation-point $s_0 = i\pi \cdot 10^{10} \in i\mathbb{R}$. Unfortunately it appears that ROM weight is not a consistently reliable indicator of transfer-function convergence when using a single interpolation-point. In this example it looks like ROM weight converges after about 30 iterations and after that, only its distribution changes. It is also possible that there is one very dominant pole that appears at $n = 30$ and it remains one pole as it converges to its resting position. The second plot is relative (explicit) ROM error over iterations $1, 2, \ldots, 160$. This plot gives a sense of localized convergence of the transfer-function. Since the single interpolation-point is placed near the center of the segment of interest, we see that the transfer-function approximation is most accurate (dark region indicates rel-error is less than 0.01) near the center and convergence works outward from there.

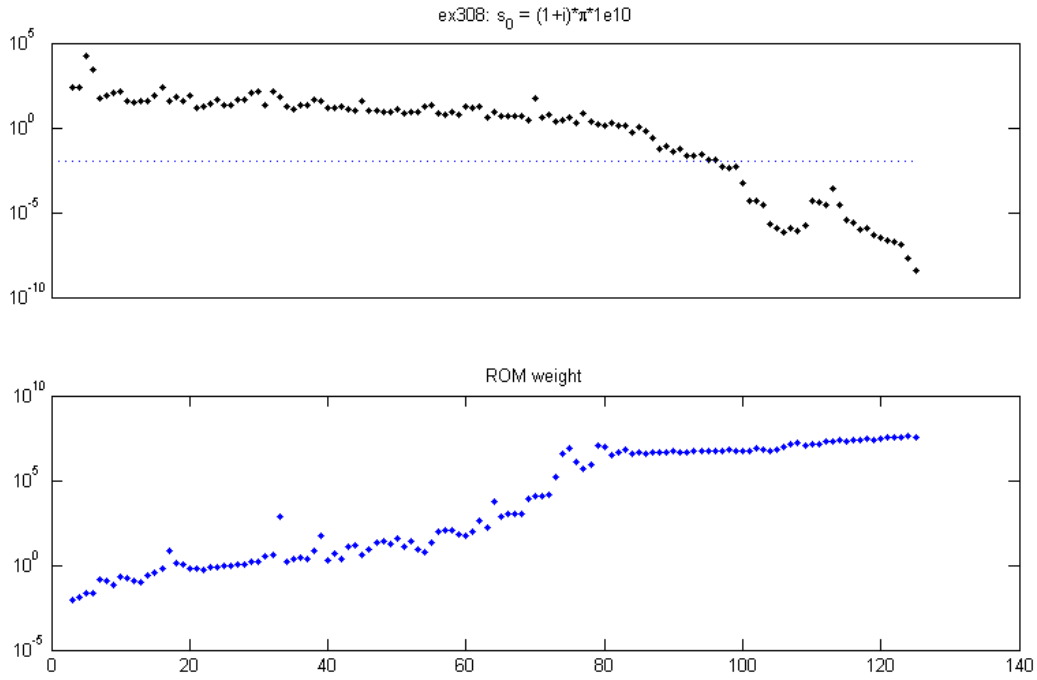


Figure 5.6: Transfer-function relative-error and ROM-weight vs. n for `ex308`, at $\sigma = (1+i)\pi \cdot 10^{10}$. This is much the way we would like the relationship between ROM-weight and transfer-function error to look. ROM-weight leveling-off would indicate transfer-function error convergence.

ex308 thick-restart example 1

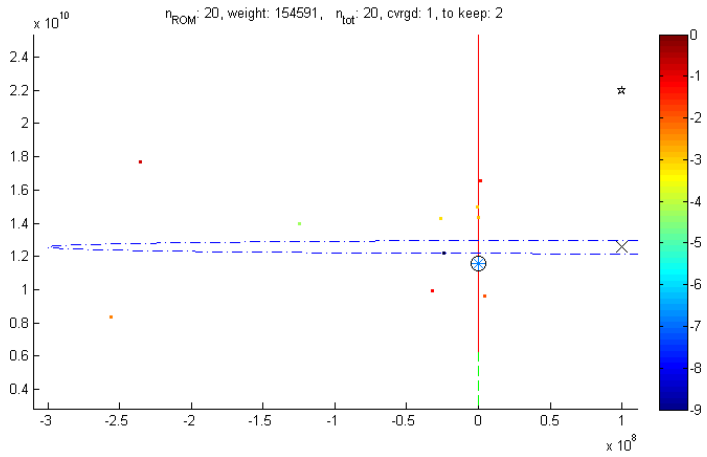
Here we show an example run of the thick-restarted band-Arnoldi process using three pre-set interpolation-points $\sigma_j = 10^8 + 2\pi i \cdot 10^{10} p_j$ for $p_j = 2, 3.5, 6$. When scaled this way, the frequency range of interest $p \in (1, 10)$ corresponds to $s \in i(10^9, 10^{10})$ so our choices of p suggest convergence of the frequency-response at those localities first, and outward from there. We ran the algorithm for for $n_j = 20, 25, 25$ iterations.

Converged Ritz-vectors (those with relative residual less than $\texttt{ctol} = \sqrt{\epsilon} \approx \texttt{1.49e-8}$) and those associated with dominant poles ($\texttt{wt}_i / \sum \texttt{wt}_i \leq 0.05$) were recycled.

The resulting ROM required a total of 70 iterations (not including re-processing thick-restarting Ritz-vectors), was of size $n' = 140$ and had a relative-error of `7.14155e-05`, making it compare favorably with the benchmark examples in table 5.5. It required $30,217,264 + 3M$ flops, where M is the cost of creating $\mathbf{H}_j$ and $\mathbf{R}_j$.

Execution of `test4('ex308',1e8+2i*pi*1e9*[2 3.5 6],[20 25 25])` yields

```
cycle 1 expanding at s_0 = 2\pi10^9(0.0159155 + 2i),  band_size = 2 + 0
... ROM: 20, n_tot: 20,   converged: 1,   keep: 2  weight: 154591
...updating thick-restart basis...dim Y = 2
```

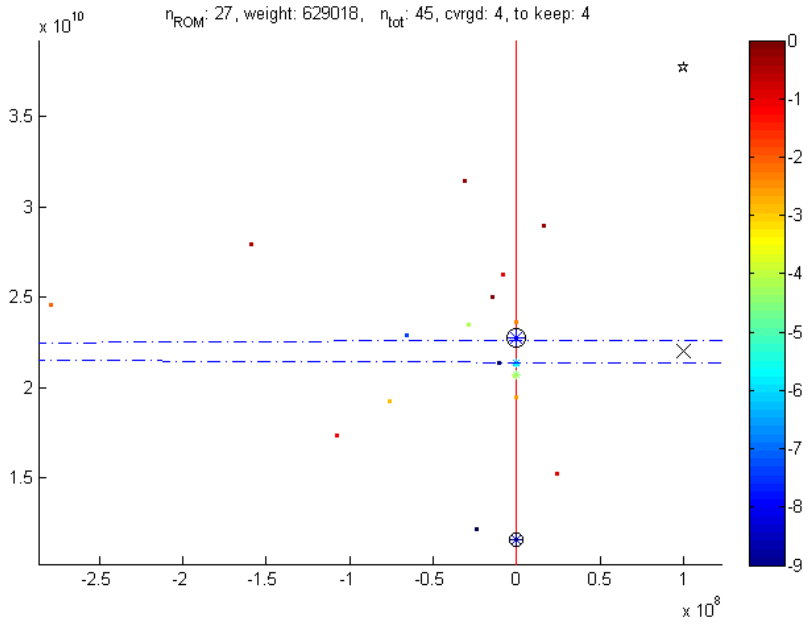


In the above plot, poles of the implicitly projected ROM of the first cycle are indicated by '$*$' symbols. The color of a pole indicates its degree of convergence. The interpolation-point is indicated by '$\times$', and a dashed-circle[2] around the interpolation-point indicates distance to the first

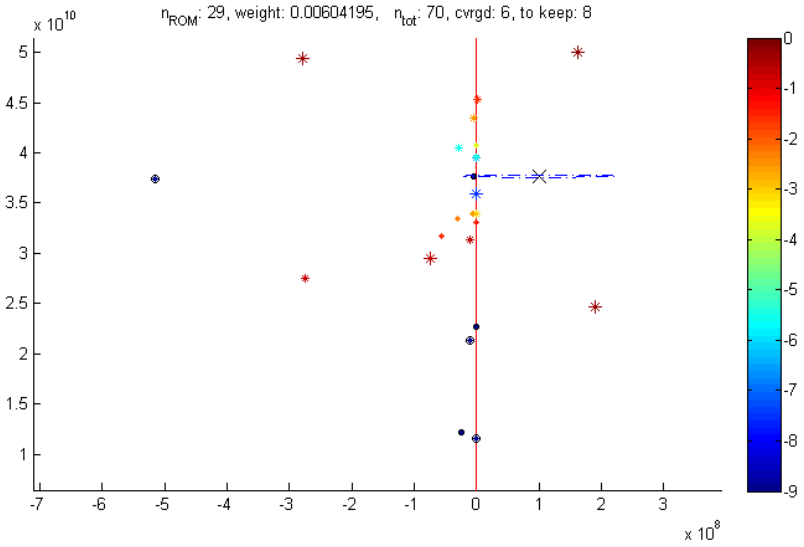
[2] elongated in the plot due to greatly asymmetric scaling. This is one reason why interpolation-points on or near the $\Im$-axis can result in the convergence of unnecessarily large ROMs.

converged pole. The '⋆' symbol indicates placement of the next interpolation-point. Pole-size in the plot corresponds to weight. Some poles have circles around them, indicating that those will be kept for thick-restarting the next cycle. In this example we are lucky to have had a very dominant pole among the two that converged on the first cycle.

```
cycle 2 expanding at s_0 = 2\pi10^9(0.0159155 + 3.5i),  band_size = 2 + 2
v_defl(5)/H_est  = 0 < 1.49012e-08,   mc = 3
v_defl(5)/H_est  = 0 < 1.49012e-08,   mc = 2
... ROM: 27, n_tot: 45,   converged: 4,   keep: 4  weight: 629018
...updating thick-restart basis...dim Y = 4
```




```
cycle 3 expanding at s_0 = 2\pi10^9(0.0159155 + 6i),  band_size = 2 + 4
v_defl(7)/H_est  = 0 < 1.49012e-08,   mc = 5
v_defl(7)/H_est  = 0 < 1.49012e-08,   mc = 4
v_defl(7)/H_est  = 0 < 1.49012e-08,   mc = 3
v_defl(7)/H_est  = 0 < 1.49012e-08,   mc = 2
... ROM: 29, n_tot: 70,   converged: 6,   keep: 8  weight: 0.00604195
...updating thick-restart basis...dim Y = 8
```


n_ROM: 29, weight: 0.00604195, n_tot: 70, cvrgd: 6, to keep: 8
x 10^10
5
4.5
4
3.5
3
2.5
2
1.5
1
-7
-6
-5
-4
-3
-2
-1
0
1
2
3
x 10^8
0
-1
-2
-3
-4
-5
-6
-7
-8
-9

	$\Re(\mu)$	$\Im(\mu)$	rr	wt	keep
1	-2.3746e+07	1.2186e+10i	3.6646e-11	0.0138714	1
2	4.8802e+01	1.1562e+10i	1.75228e-07	154560	1
3	-1.2457e+08	1.3997e+10i	5.15493e-05	0.0166039	0
4	-3.2363e+08	1.0214e+10i	0.000147586	0.0136769	0
5	-1.7281e+09	1.4688e+10i	0.00067386	0.0223279	0
6	-2.6226e+07	1.4295e+10i	0.000702401	0.0122022	0
7	-9.2396e+05	1.4978e+10i	0.00092006	14.3325	0
8	1.2915e+05	1.4334e+10i	0.00126677	1.344	0
9	-2.5567e+08	8.3294e+09i	0.00515879	0.0234752	0
10	4.1068e+06	9.6085e+09i	0.0136536	0.503765	0

(a) Cycle 1

	$\Re(\mu)$	$\Im(\mu)$	rr	wt	keep
1	-2.3746e+07	1.2186e+10i	0	0.0110811	1
2	1.9498e+02	1.1562e+10i	0	44474.2	1
3	-1.0463e+07	2.1391e+10i	3.12586e-09	0.151564	1
4	1.1927e-01	2.2714e+10i	7.41398e-09	567340	1
5	-6.6011e+07	2.2864e+10i	4.3831e-08	0.013322	0
6	-4.9619e+00	2.1332e+10i	2.98909e-07	11524.3	0
7	5.1626e+00	2.1293e+10i	2.11094e-06	834.36	0
8	-8.2289e+08	2.0635e+10i	1.2973e-05	0.0239924	0
9	-4.4901e+01	2.0682e+10i	3.32799e-05	4827.55	0
10	-2.8524e+07	2.3487e+10i	6.76036e-05	0.0287073	0

(b) Cycle 2

	$\Re(\mu)$	$\Im(\mu)$	rr	wt	keep
1	-2.3746e+07	1.2186e+10i	0	3.06212e-08	1
2	4.9552e+00	2.2714e+10i	0	6.69624e-09	1
3	-1.0463e+07	2.1391e+10i	0	5.41799e-08	1
4	-2.4655e+02	1.1562e+10i	0	1.49686e-07	1
5	-4.3476e+06	3.7641e+10i	5.19823e-23	1.98325e-08	1
6	-5.1591e+08	3.7457e+10i	2.15413e-14	2.57324e-07	1
7	-1.2248e+00	3.5908e+10i	3.04755e-08	4.08767e-07	0
8	-3.5711e+00	3.9570e+10i	4.79573e-07	5.11497e-08	0
9	-2.8146e+07	4.0486e+10i	3.09744e-06	8.83886e-08	0
10	-3.6394e+04	3.9670e+10i	3.63502e-06	3.98871e-09	0

(c) Cycle 3

Table 5.6: The 10 Ritz-poles of lowest relative-residual for each cycle. One thing to note is that pole-weight does not stay consistent from cycle to cycle for this test. Two converged poles (indicated by shaded rows in the tables) appear to change dominance quite drastically. This could be due to the way we define pole-weight.

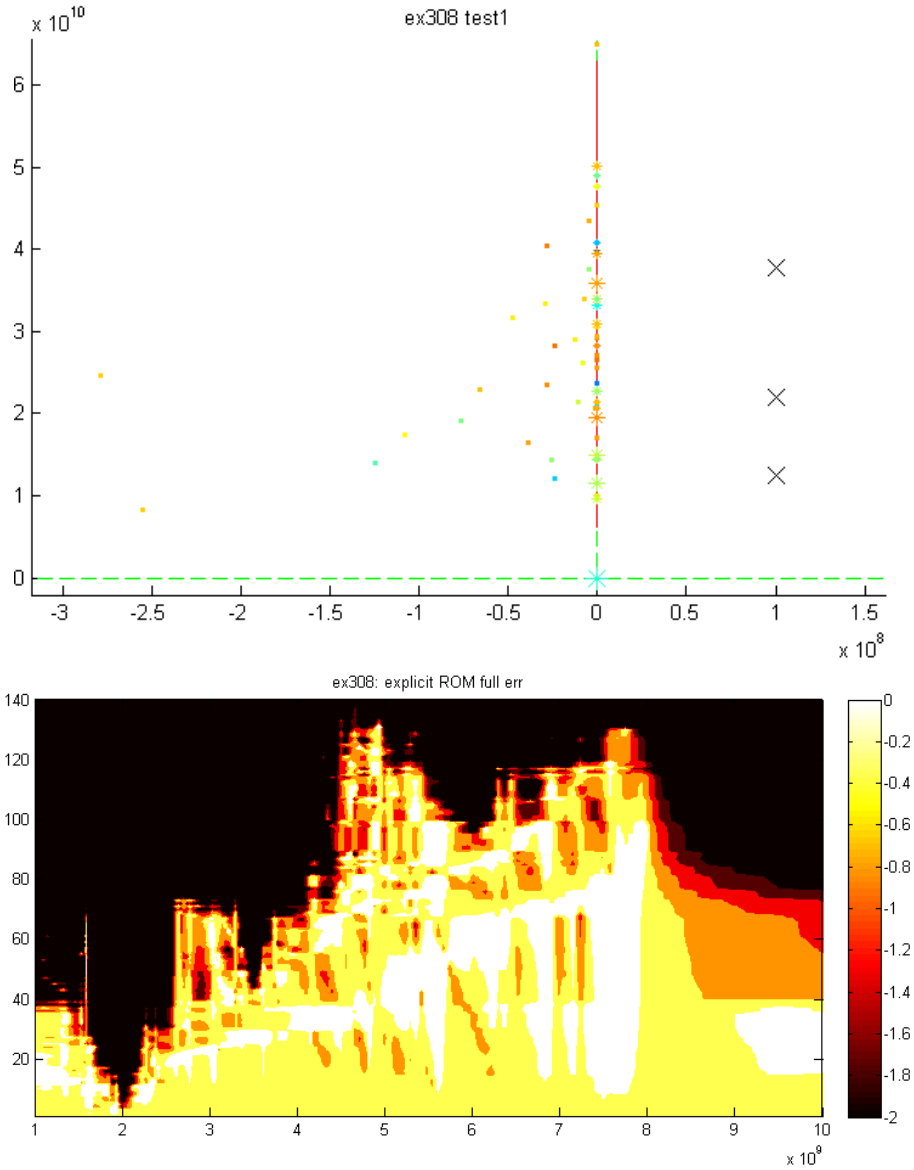


Figure 5.7: The pole distribution for the explicitly-projected transfer-function and interpolation-points are in the first plot. The second plot, of local transfer-function-error over the frequency range of interest evolving with inclusion of basis vectors in V_{ROM} reflects expansion about the points $p = 2$, 3.5, and 6. It appears that a smaller and more accurate ROM could have been constructed with fewer iterations at $p = 2$ and more iterations at $p = 6$, or possibly two more interpolation-points at $p = 5$ and 8. We found the `1e8` offset from the $\Im$-axis to yield good results in numerous test-runs of the process for this example.

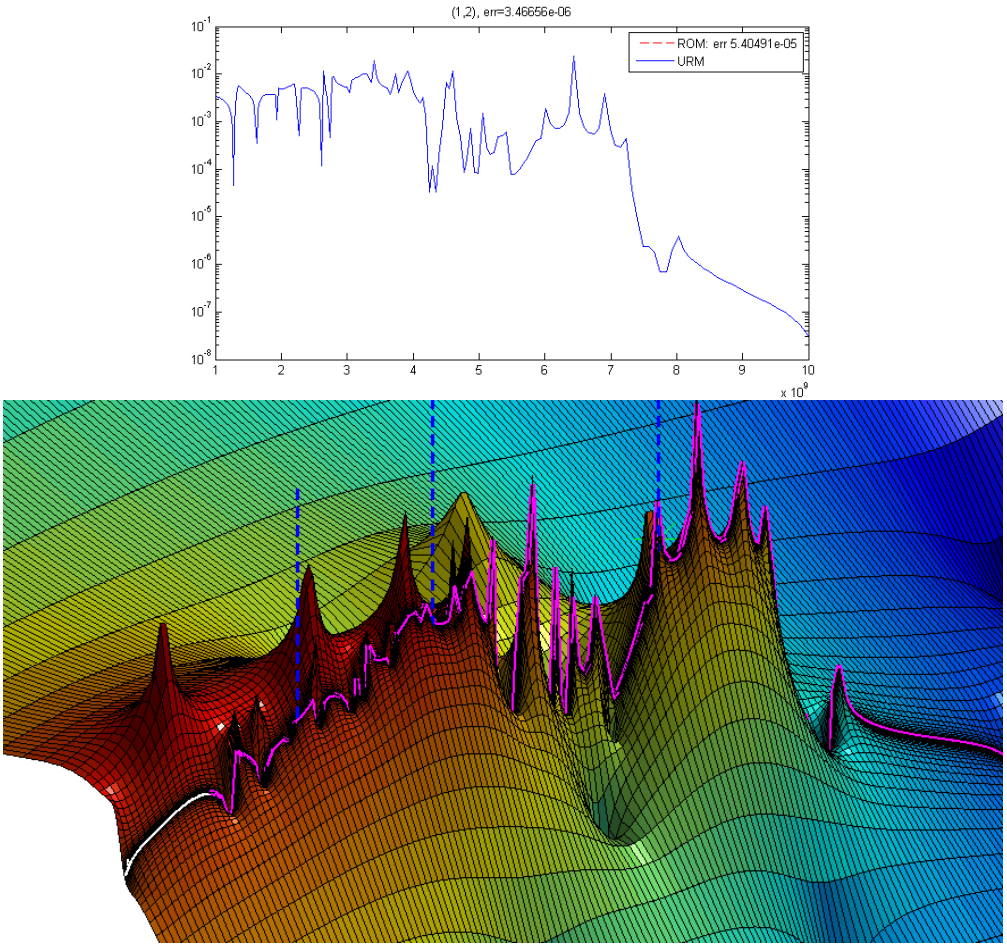


Figure 5.8: Plots of the $(1, 2)$ component of the frequency-response and transfer-function surface for example 1. It looks like the interpolation-point placement was almost ideal for `ex308`, in the sense that the points are near centers of pole-mass. Note that the interpolation-points are actually offset `1e8` into the $\Re$ half-plane, but the scale of the surface plot is such that they appear to be on the segment of interest.

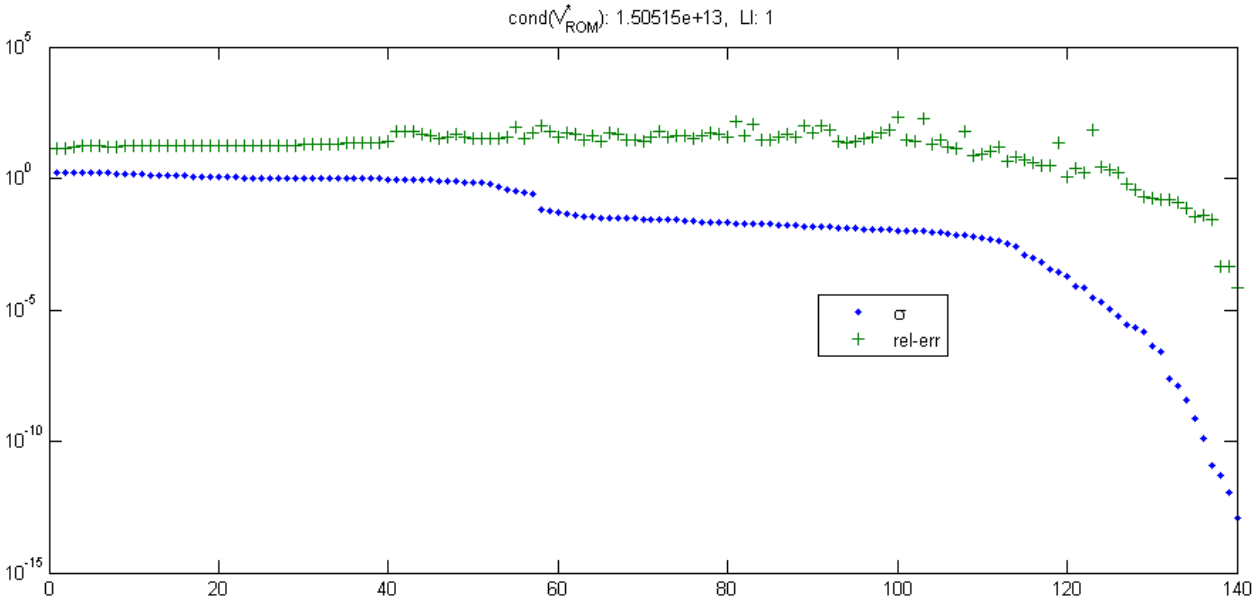


Figure 5.9: Here we looked at relative-error for explicitly-projected ROM transfer-functions for successively larger bases U_n for $n = 1, 2, \ldots, n' = 140$, where U is the basis of left singular-vectors of $\widehat{V}$, and σ is the corresponding singular value.

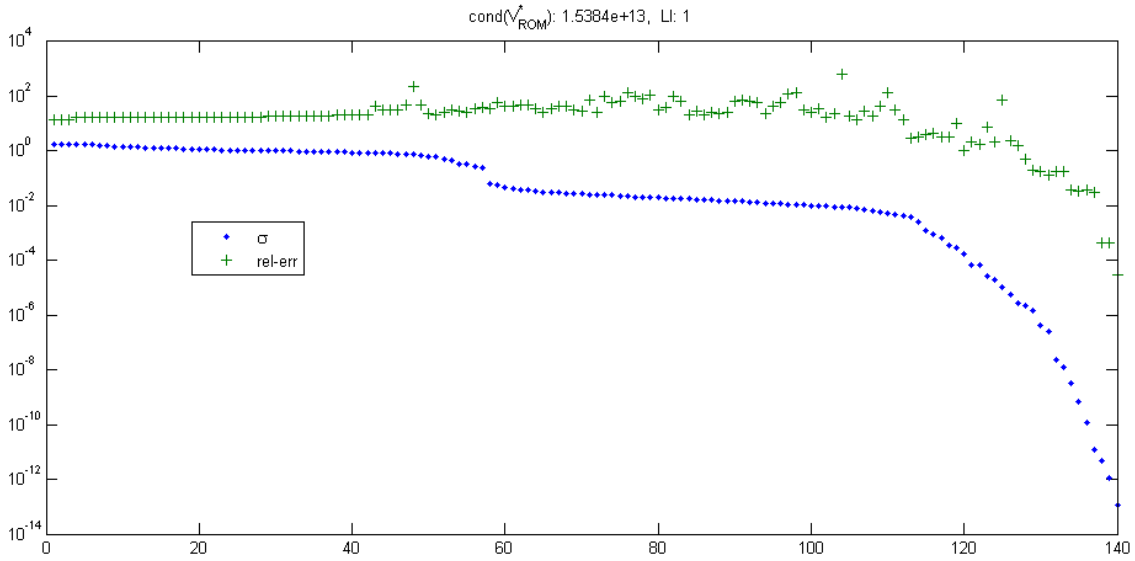


Figure 5.10: Transfer-function relative error for `ex308` test 2, plotted in order of decreasing singular values. The condition number of the split basis in this case is `1.5384e13`, which is not much different from that from test 1, indicating that keeping Ritz-vectors did improve linear independence of the projection basis, but not by much.

ex308 thick-restart example 2

The next test is similar to the previous one except that we did not keep any Ritz-vectors from cycle to cycle. In this case the model of size $n' = 140$ had about the same accuracy at with a relative transfer-function error of `2.84772e-05`. It was negligibly cheaper to construct at $27,707,680 + 3M$ flops. There was very little overlap between Krylov-subspaces in this case, possibly because of the low number of iterations we performed.

	$\Re(\mu)$	$\Im(\mu)$	rr	wt	keep
1	-2.3746e+07	1.2186e+10i	3.6646e-11	0.0138714	0
2	4.8802e+01	1.1562e+10i	1.75228e-07	154560	0
3	-1.2457e+08	1.3997e+10i	5.15493e-05	0.0166039	0
4	-3.2363e+08	1.0214e+10i	0.000147586	0.0136769	0
5	-1.7281e+09	1.4688e+10i	0.00067386	0.0223279	0
6	-2.6226e+07	1.4295e+10i	0.000702401	0.0122022	0
7	-9.2396e+05	1.4978e+10i	0.00092006	14.3325	0
8	1.2915e+05	1.4334e+10i	0.00126677	1.344	0
9	-2.5567e+08	8.3294e+09i	0.00515879	0.0234752	0
10	4.1068e+06	9.6085e+09i	0.0136536	0.503765	0

(a) Cycle 1

	$\Re(\mu)$	$\Im(\mu)$	rr	wt	keep
1	-1.0463e+07	2.1391e+10i	3.15983e-09	0.151564	0
2	1.3560e-01	2.2714e+10i	7.5411e-09	559182	0
3	-6.6011e+07	2.2864e+10i	4.42028e-08	0.013322	0
4	-2.4243e-02	2.1332e+10i	3.18392e-07	67080.5	0
5	4.2240e+00	2.1293e+10i	2.25896e-06	984.276	0
6	-8.2289e+08	2.0635e+10i	1.3741e-05	0.0239923	0
7	-9.0853e+02	2.0682e+10i	3.62199e-05	243.656	0
8	-2.8524e+07	2.3487e+10i	6.645e-05	0.0287091	0
9	-1.3230e+06	2.0649e+10i	7.29665e-05	0.0201709	0
10	-7.5781e+07	1.9234e+10i	0.0017068	0.0163881	0

(b) Cycle 2

	$\Re(\mu)$	$\Im(\mu)$	rr	wt	keep
1	-4.3476e+06	3.7641e+10i	5.23777e-23	2.51929e-08	0
2	-5.1591e+08	3.7457e+10i	2.16358e-14	3.26875e-07	0
3	-2.2323e+00	3.5908e+10i	3.06443e-08	5.1925e-07	0
4	-5.0044e+00	3.9570e+10i	4.83015e-07	6.49747e-08	0
5	-2.8146e+07	4.0486e+10i	3.1165e-06	1.12279e-07	0
6	-3.6431e+04	3.9670e+10i	3.65723e-06	5.06681e-09	0
7	2.2037e+03	4.0796e+10i	0.000200954	5.49435e-09	0
8	4.6830e+04	3.3983e+10i	0.000950016	1.14312e-07	0
9	-3.9908e+06	4.3477e+10i	0.0023368	1.45873e-07	0
10	-7.1975e+06	3.3975e+10i	0.0024024	5.09175e-09	0

(c) Cycle 3

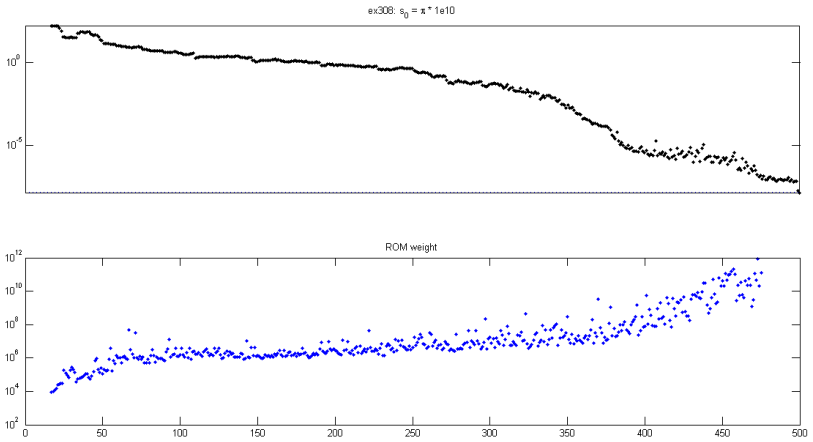
Table 5.7: Here are the most converged eigenvalues of each cycle of the same example, except that we kept no Ritz values from cycle to cycle.

5.4.2 ex1841

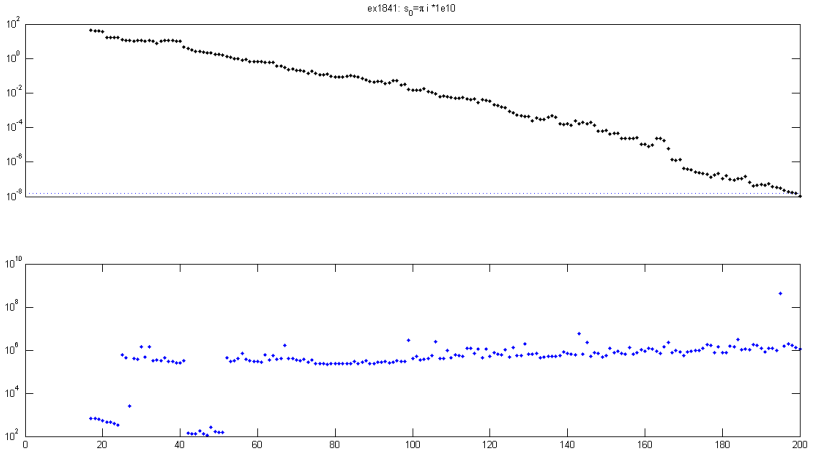
`ex1841` is a 16×16 (inputs $\times$ outputs) MIMO model that comes from a MNA expression of an RCL circuit model of interconnect.

σ	iterations (n)	ROM size (n')	LI	rel-err	flops	figure
$\pi 10^{10}$	310	310	1	9.1289e-3	1,147,983,165 + M	5.11a
$i\pi 10^{10}$	106	212	1	8.6730e-3	1,490,525,148 + M	5.11b
$(1+i)\pi 10^{10}$	142	284	1	8.4797e-3	2,015,563,620 + M	5.11c

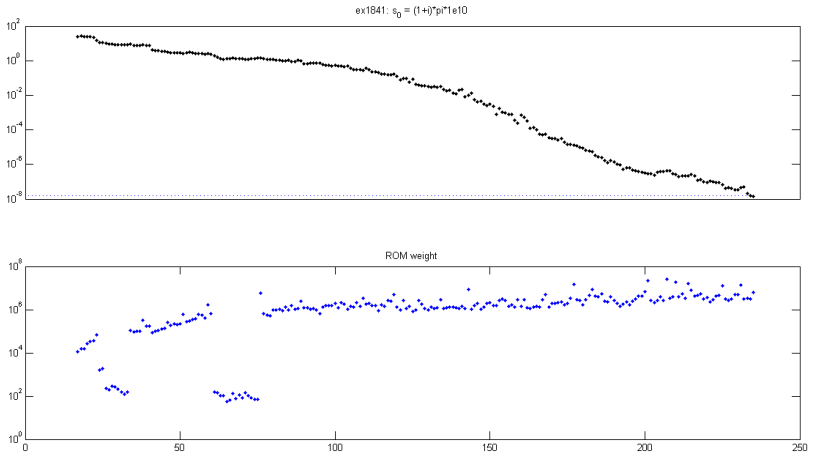
Table 5.8: Benchmark data for `ex1841`. flops is a count of real (in $\mathbb{R}$), non-zero scalar products required for matrix-vector multiplication and inner-products.



(a) $\sigma = \pi 10^{10}$



(b) $\sigma = i\pi 10^{10}$



(c) $\sigma = (1+i)\pi 10^{10}$

Figure 5.11: Transfer-function relative-error (188) and ROM weight vs. n for `ex1841`, at each of the three points shown in figure 5.1.

ex1841 test1

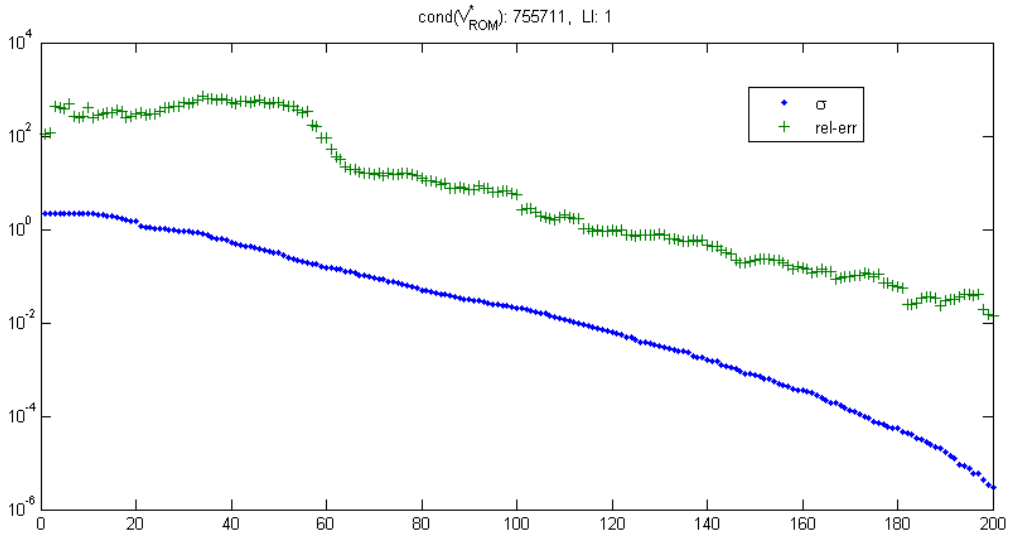
We ran the thick-restarted band-Arnoldi algorithm for 20 iterations at each of 5 interpolation points $10^3 + 2ip_j \cdot \pi \cdot 10^9$, for $p_j = 1, 3, 5, 7, 9$.

```
test4('ex1841',1e3+[1 3 5 7 9]*2i*pi*1e9,[20 20 20 20 20]);
```

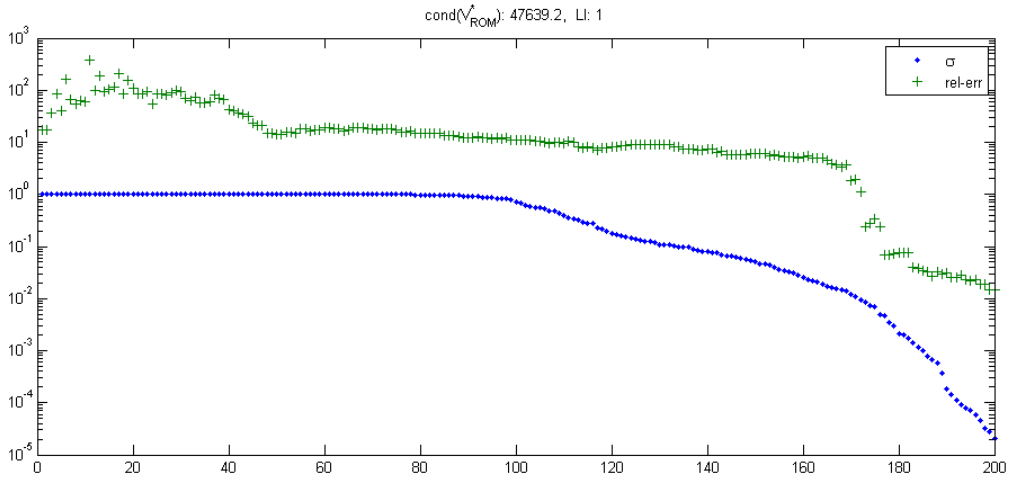
The purpose of this test was to compare ROMs produced by restarted band-Arnoldi with and without "thick-restarting", i.e. keeping Ritz vectors. For low iteration counts, the difference is almost negligible with this model, even when interpolation points are fairly closely located.

`keep_tol`	n	n'	rel-error	flops	cond #
0	100	200	0.0143945	1374490600 + 5M	755711
∞	100	200	0.0143945	4270751800 + 5M	47639

In fact, for this test the only differences in models are the operations required to produce them and the linear-independence of the basis vectors.

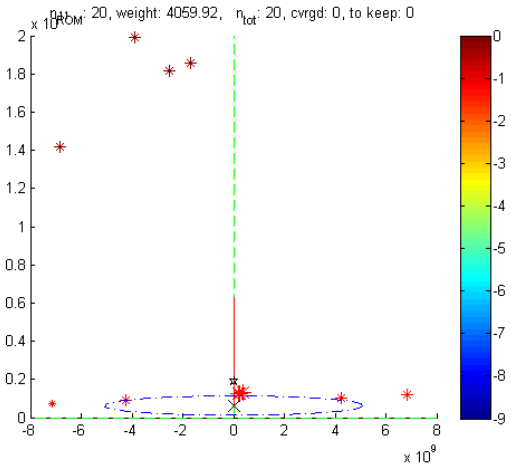
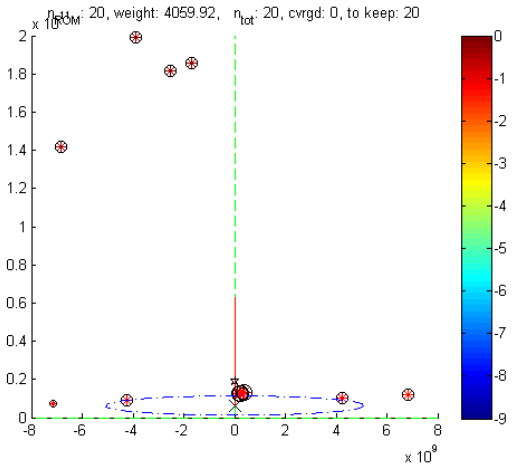
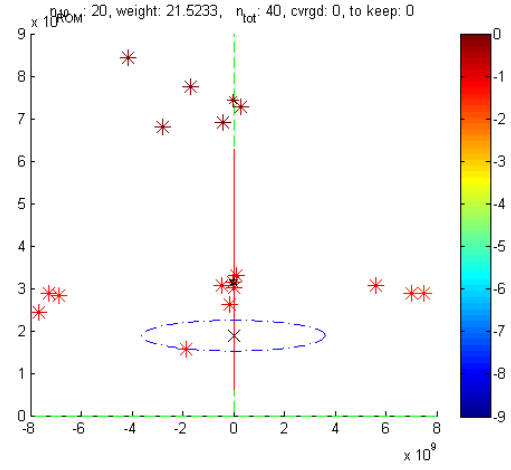
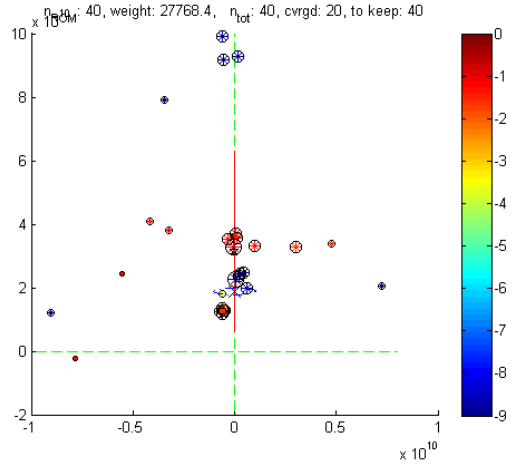
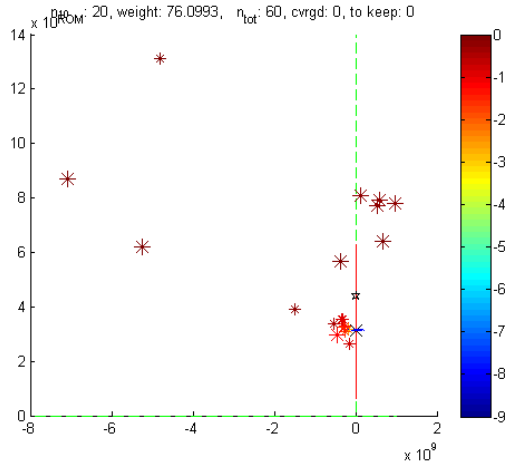
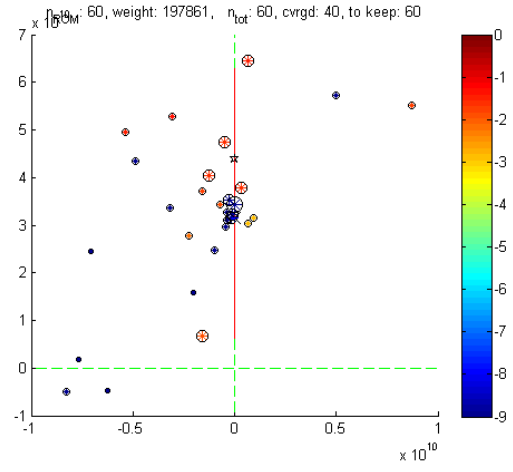


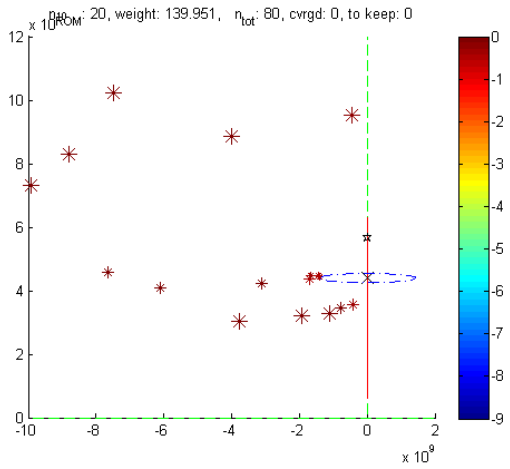
(a) keeping no Ritz-vectors, condition number: 755711

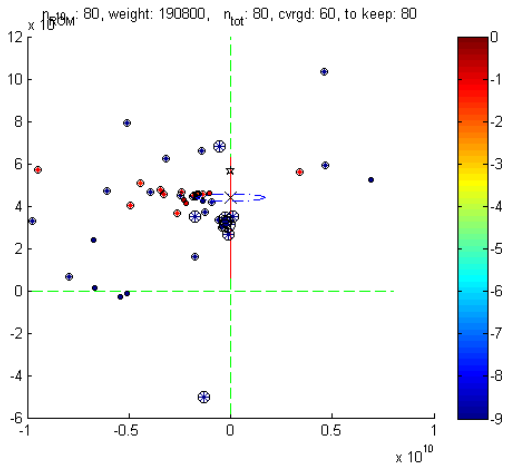


(b) keeping all Ritz-vectors, condition number: 47639

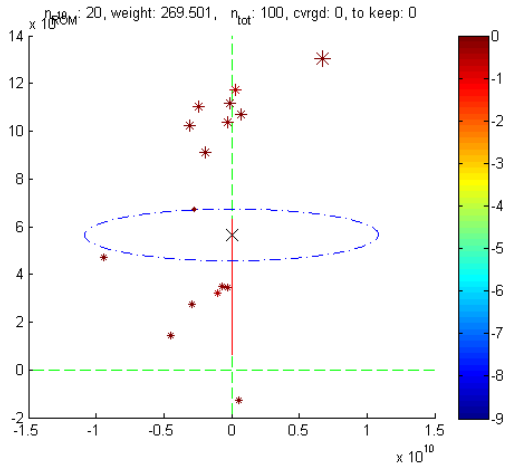
Figure 5.12: singular values and relative transfer-function error for each ROM of test 1.

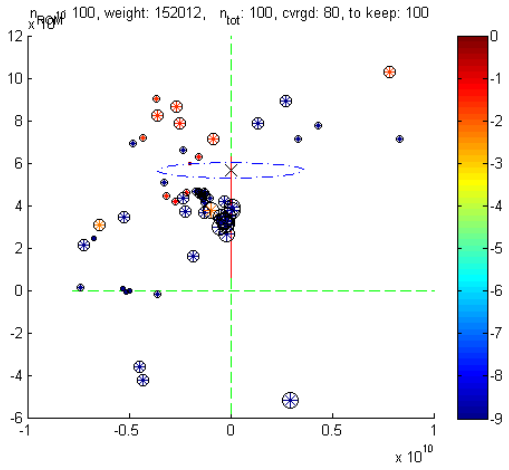
cycle	keep none (`keep_tol` = 0)	keep all (`keep_tol` = ∞)
1		
2		
3		


n_{RCM}: 20, weight: 139.951, n_{tot}: 80, cvrgd: 0, to keep: 0
x 10^9


n_{RCM}: 80, weight: 190800, n_{tot}: 80, cvrgd: 60, to keep: 80
x 10^10

4


n_{RCM}: 20, weight: 269.501, n_{tot}: 100, cvrgd: 0, to keep: 0
x 10^10


n_{RCM}: 100, weight: 152012, n_{tot}: 100, cvrgd: 80, to keep: 100
x 10^10

5

Test-model MNA_2

`MNA_2` is a MIMO (18×18) circuit model of order $N = 9223$ from the benchmark (test-model) collection [7]. Its transfer function has $(18)(17)/2 = 153$ unique components. Here we present surface plots for five of them: $\mathcal{H}_{ij}$ for $(i,j) = (1,1), (1,10), (2,2), (8,10), (5,15)$. These choices have no particular significance, although there may be relationships between components that could be exploited for further MIMO reduction. For this test we considered the frequencies of interest to be $f \in [10^8, 10^{10}]$.

We ran our thick-restarted method for the following set of rational-interpolation parameters:

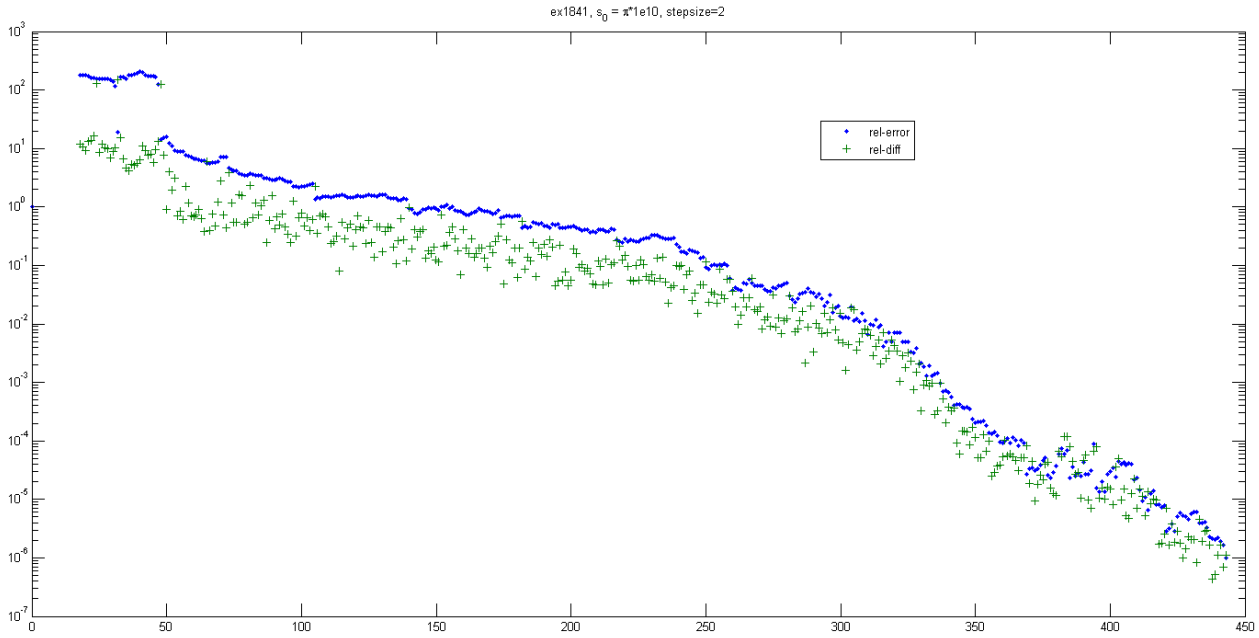
j	p_j	n_j
1	$0.150354 + 1.43386i$	22
2	$0.269667 + 9.03516i$	22
3	$-0.882785 + 5.4727i$	22
4	$0.165058 + 4.87005i$	22
5	$-0.189358 + 8.58293i$	22
6	$-1.49645 + 8.00428i$	46

This means we performed restarts after n_j iterations of band-Arnoldi with $\mathbf{H}(\sigma_j)$, $\mathbf{R}(\sigma_j)$, where $\sigma_j = 2\pi 10^9 p_j$ was the interpolation-point for the j-th cycle, for $j = 1, 2, ..., 6$.

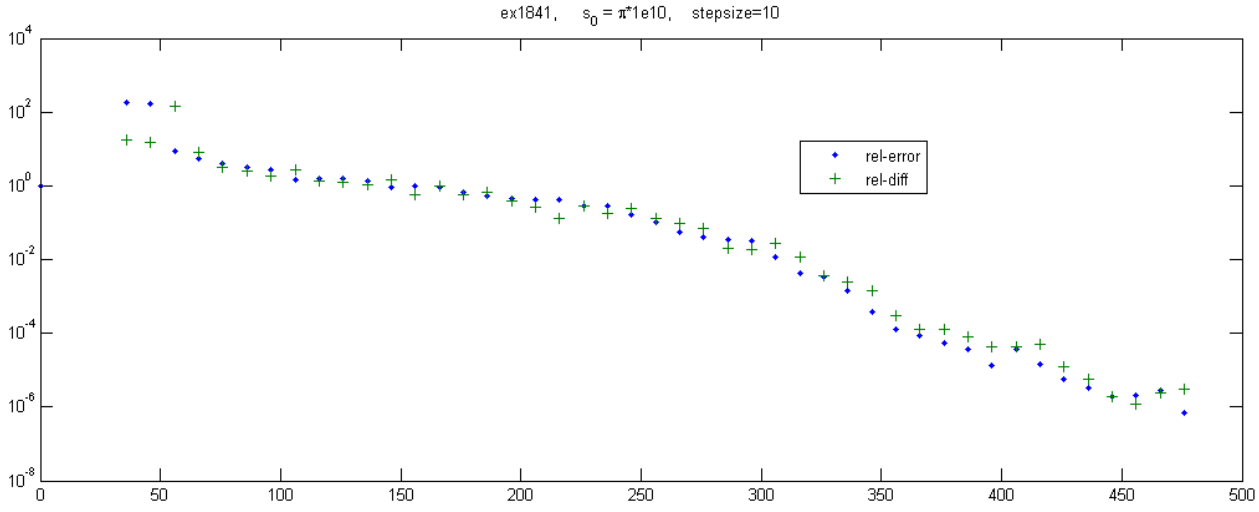
There were a total of 156 iterations performed producing a ROM of order $n = 312$. The estimated relative-error was `2.53015e-05`. We estimated relative-error by relative-difference as explained in figures 5.13 and 5.14. We set the Ritz-convergence tolerance relatively low, at `ctol` $= 10^{-4}$. Still, no Ritz-values converged on any cycle. Basis-construction (before split and re-orthogonalize) took $53264153112 + 6M$ flops, where M is the cost of preparing $\mathbf{H}_j$ and $\mathbf{R}_j$ for computation.

Figure 5.14 illustrates that it requires a ROM of order $n \approx 480$ to achieve an estimated relative-error on the order of `1e-05` using a single, real interpolation point located at $\pi \cdot 10^{10}$,[3] we can improve upon it using rational-interpolation with an almost haphazardly-selected set of 5 interpolation-points. Our ROM is roughly (312/480) two-thirds the size of the original. This is not a large reduction in model size, but some effort to optimize point placement could yield better improvements.

[3] which would be near-optimal placement for a real interpolation point according to [23]



(a) stepsize 2



(b) stepsize 10

Figure 5.13: For very large test-models we did not have URM Frequency response data available so we estimated relative-error with relative iterate-difference $\|\widehat{\mathcal{H}}_n - \widehat{\mathcal{H}}_{n-k}\|/\|\widehat{\mathcal{H}}_n\|$ of consecutive implicit transfer-functions. Here we compare relative-error and relative-difference of explicit projection ROMs of `ex1841`, for step-sizes $k = 2$ and 10. As far as we know, these comparisons are model independent. We decided $k = 10$ is a good-enough measure of relative-error.

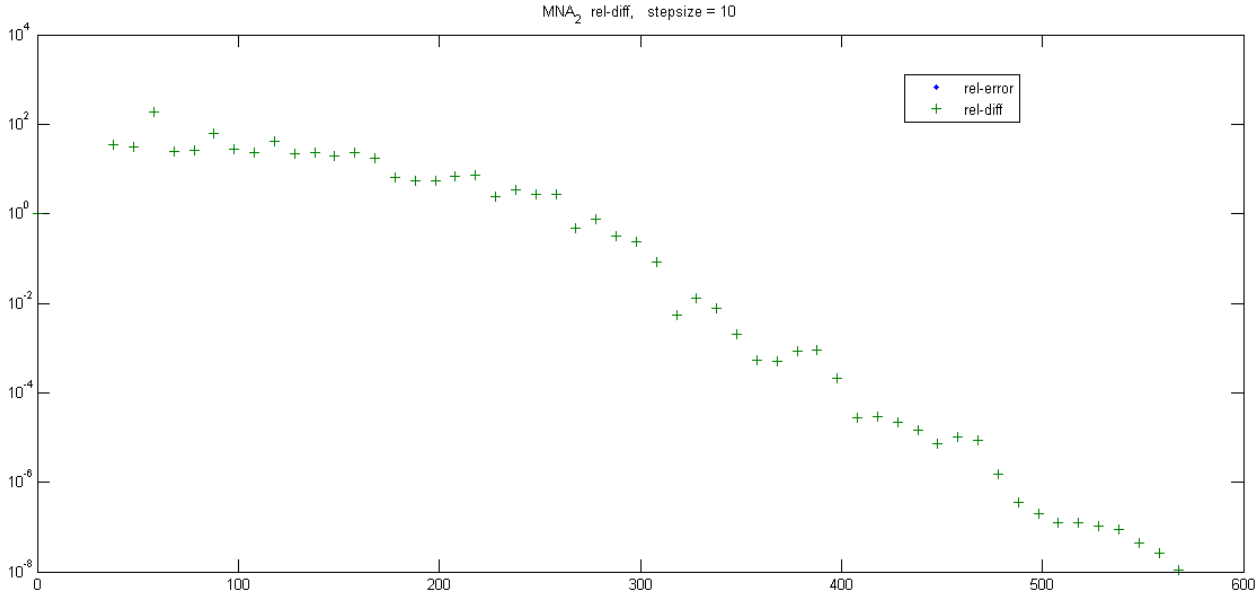


Figure 5.14: This plot of relative iterate-difference (we will always use stepsize $k = 10$) for `MNA_2` about expansion-point $\pi \cdot 10^{10}$ indicates our estimate of its accuracy vs ROM size. It requires a ROM of size about $n \approx 580$ to have an accuracy on the order of $\sqrt{\epsilon} \approx 10^{-8}$.

ROM transfer-function gain plots for MNA_2 example

Here we show gain plots for the five ROM transfer-function components. The locations of the interpolation points are indicated by $*$ symbols.

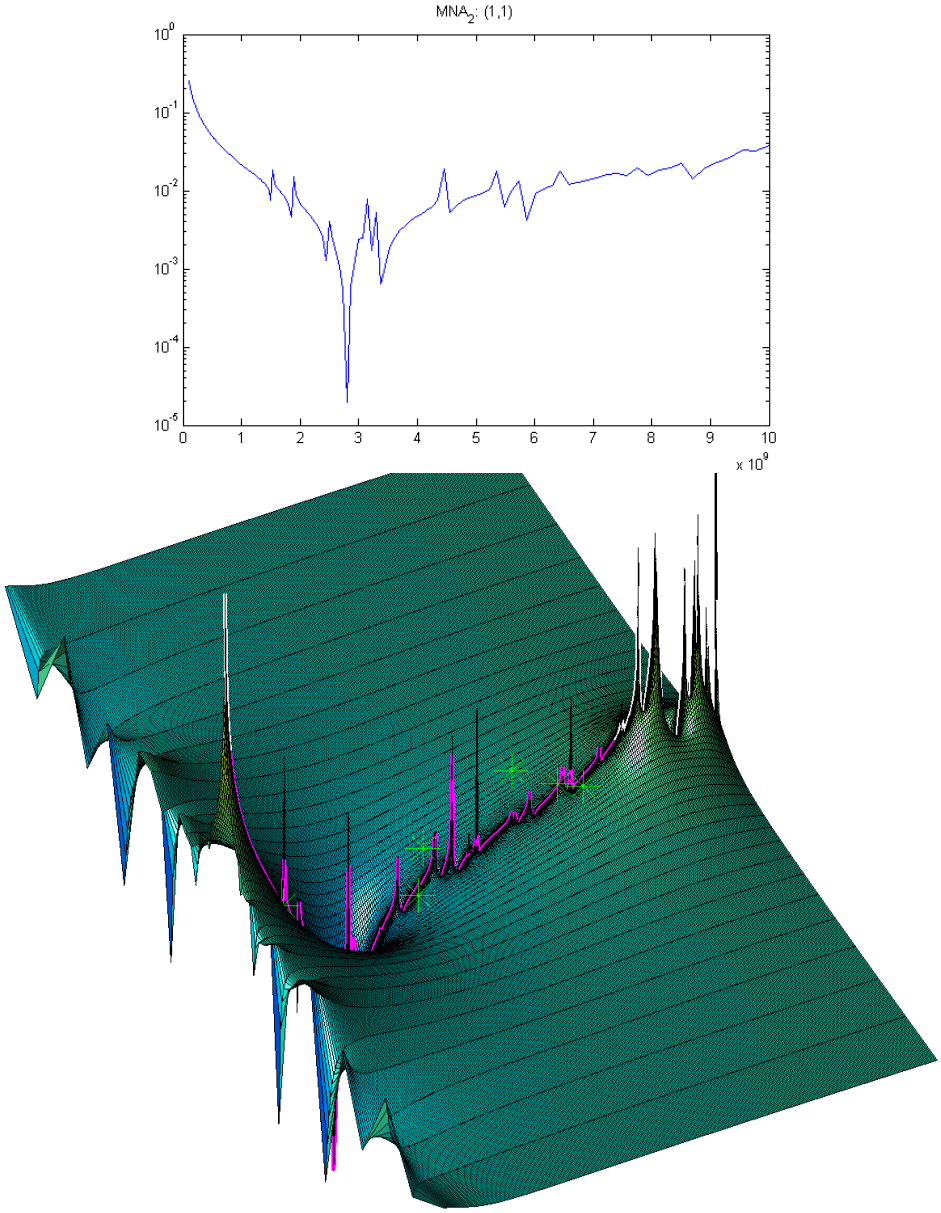


Figure 5.15: $(1,1)$ of $n = 312$ ROM of MNA_2

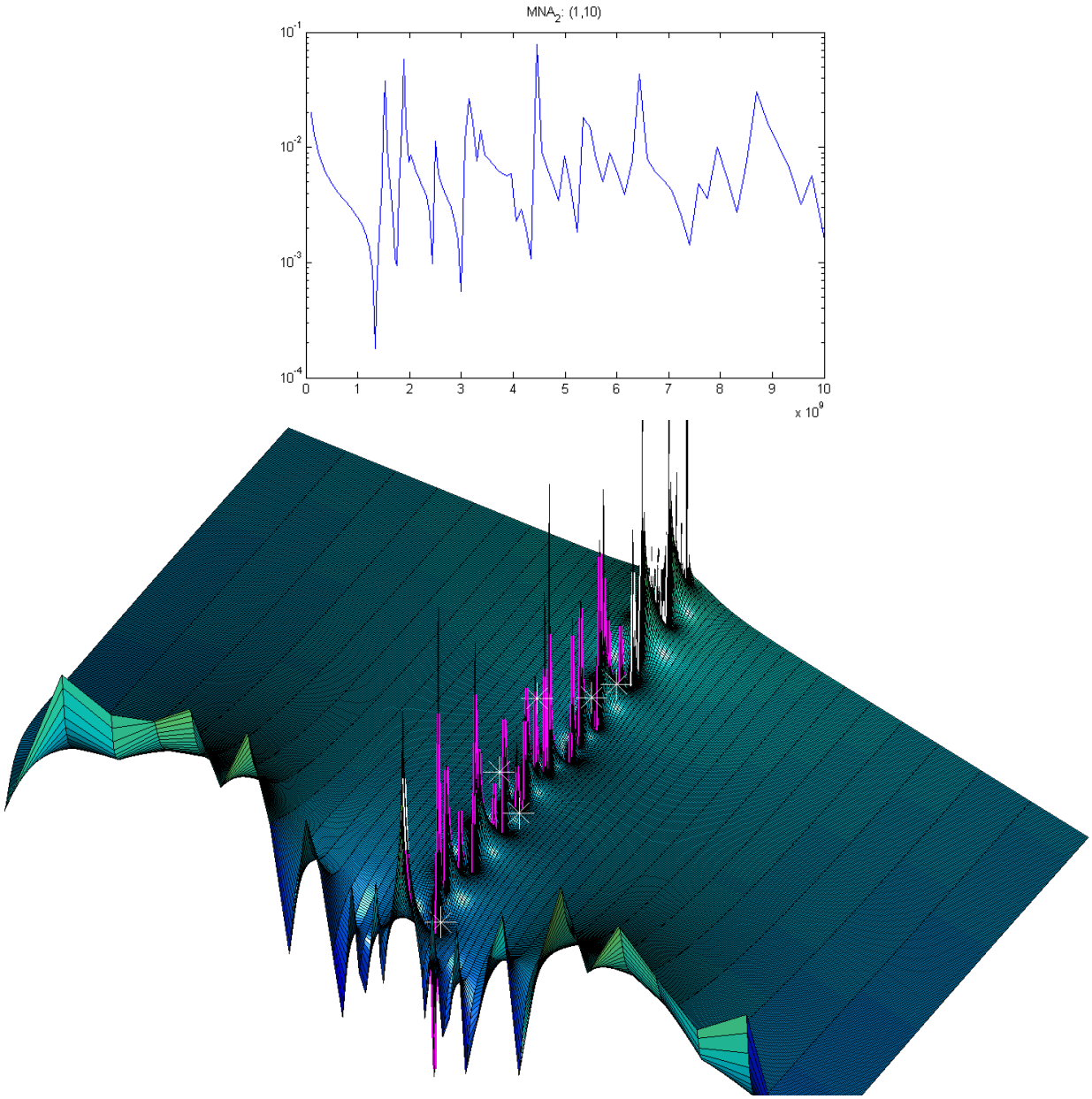


Figure 5.16: $(1, 10)$ of $n = 312$ ROM of `MNA_2`

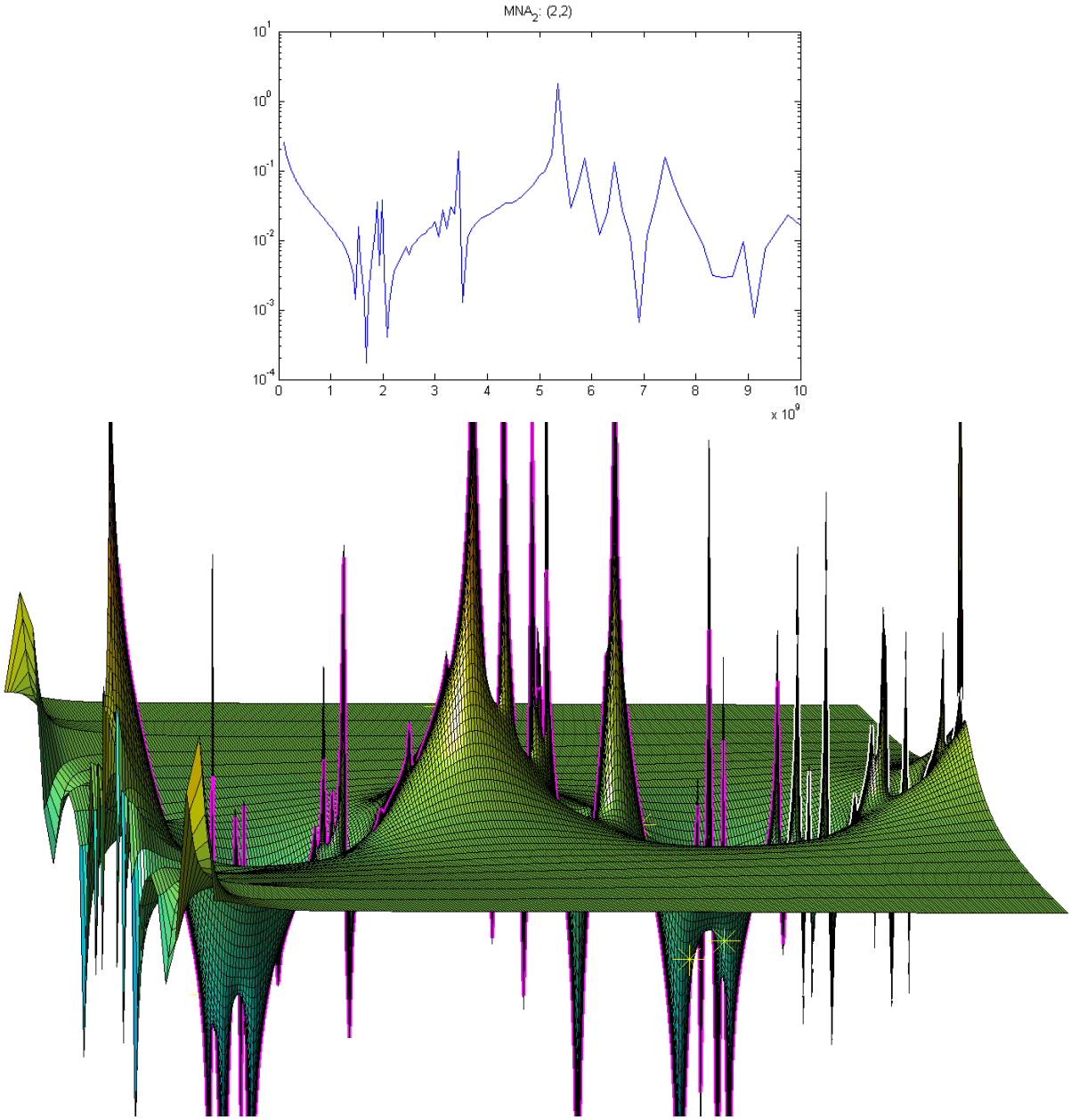


Figure 5.17: $(2,2)$ of $n = 312$ ROM of `MNA_2`

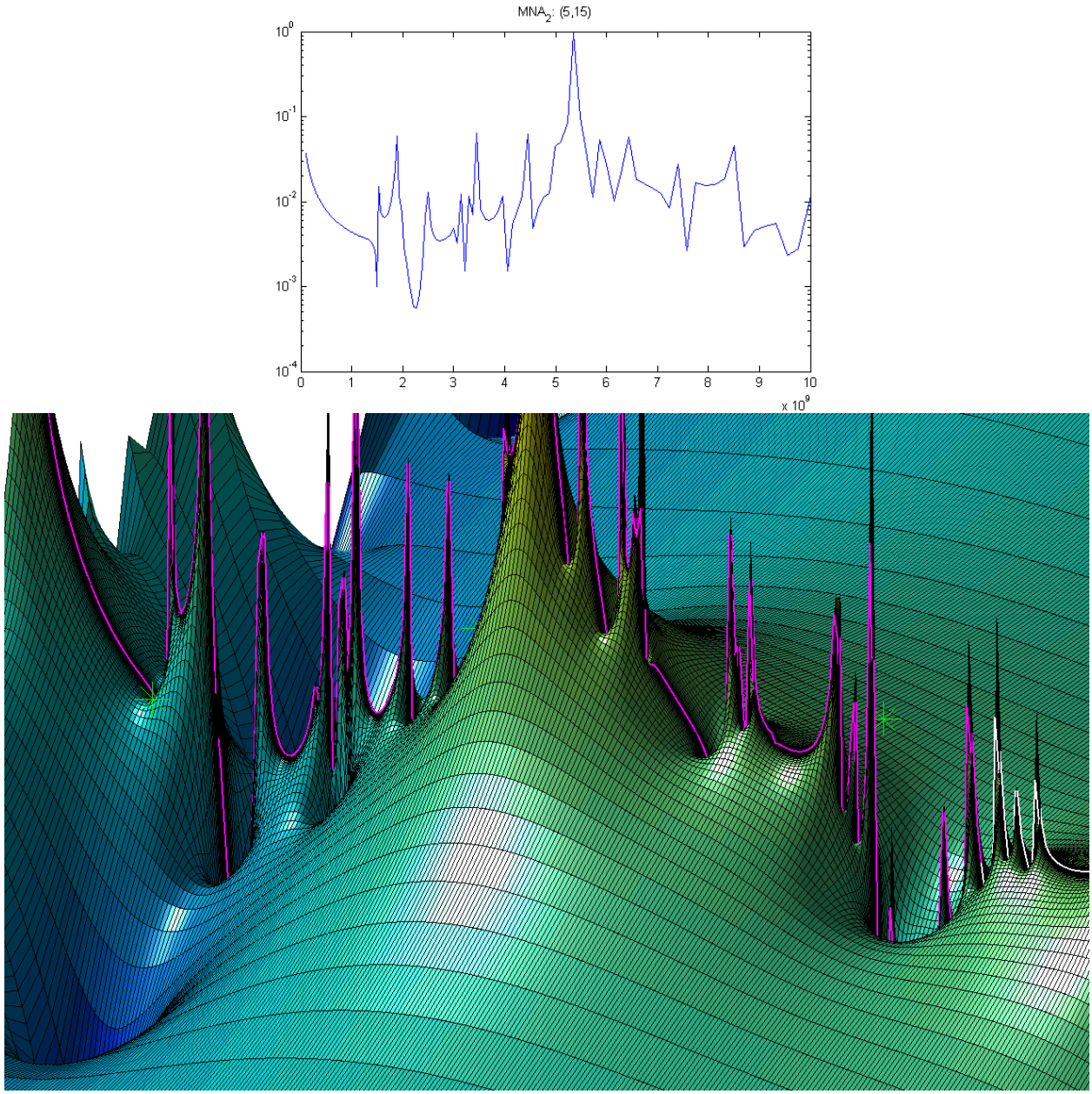


Figure 5.18: $(5, 15)$ of $n = 312$ ROM of `MNA_2`

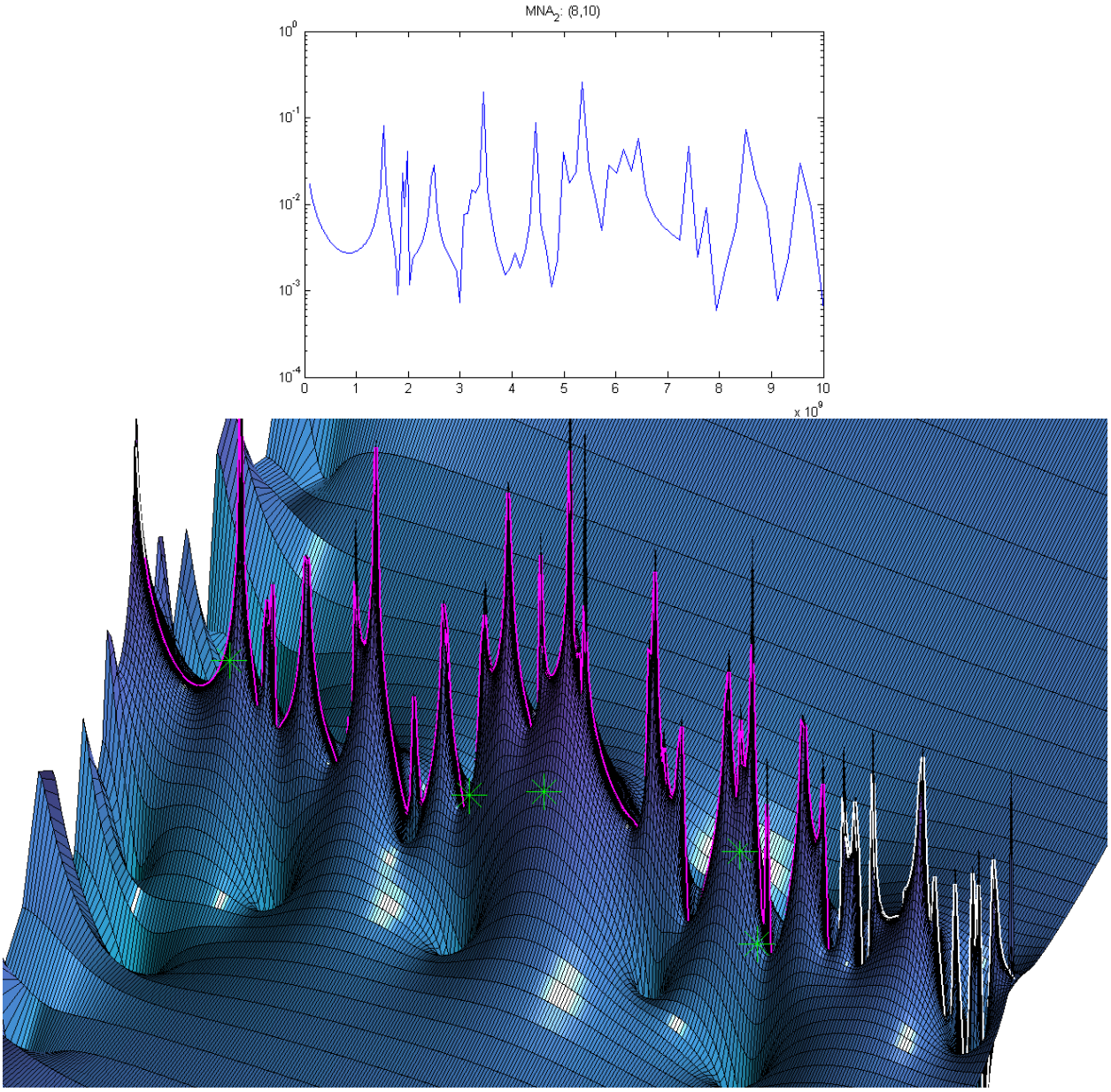


Figure 5.19: $(8, 10)$ of $n = 312$ ROM of `MNA_2`

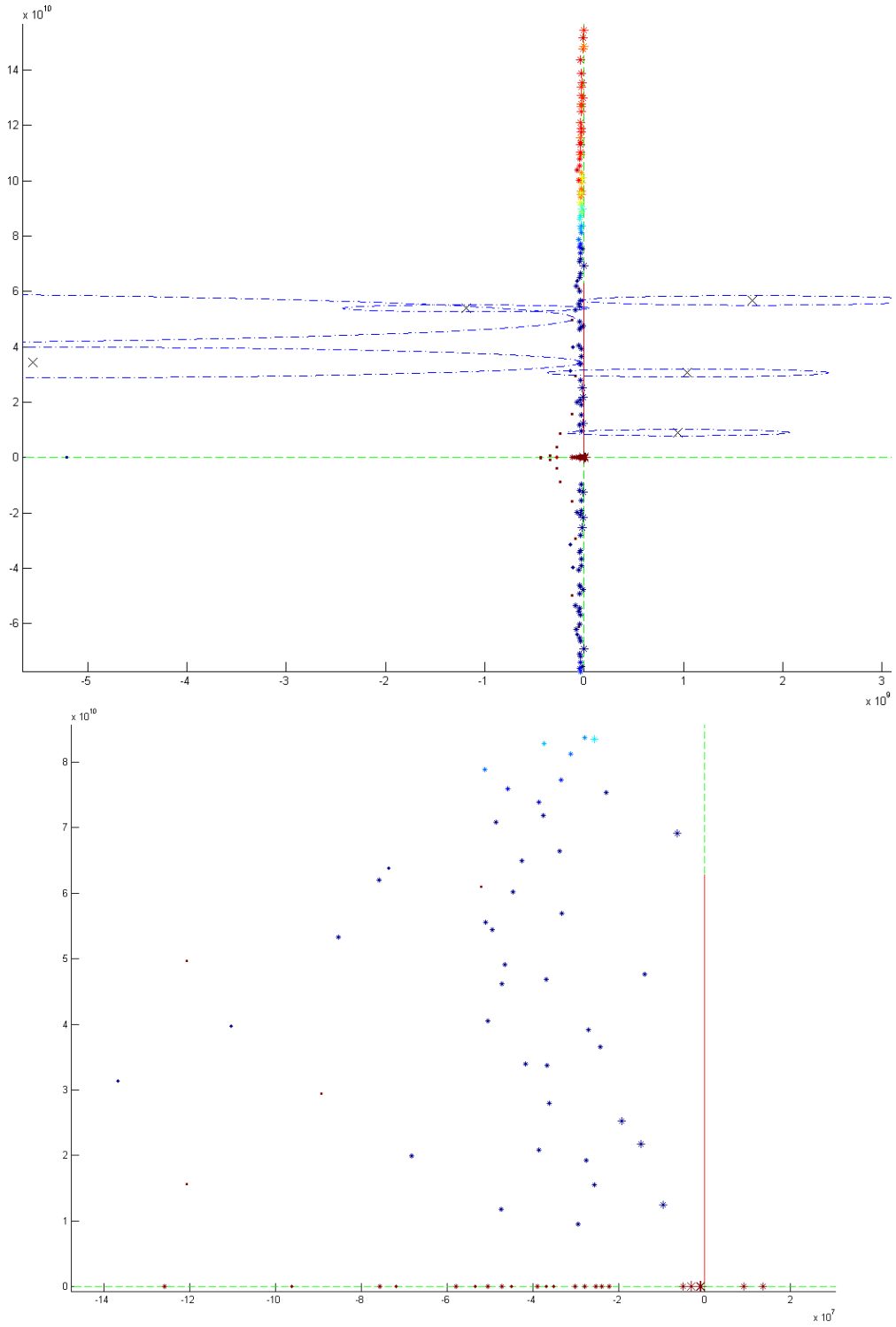


Figure 5.20: Explicit poles of the $n = 312$ ROM of `MNA_2`. Color indicates degree of convergence (relative-residual error). The second plot is a "zoom-in" of the first, where we only show poles near the segment of interest. Note no zeros are indicated. Krylov-subspace methods have no simple way to determine zeros, as far as we know. The first plot includes the interpolation points, and a circle indicating distance to the first pole. The "circles" appear elongated due to scaling.

Chapter 6

Conclusion

In this document we have provided the theoretical background for a thick-restarted band-Arnoldi process for multi-input multi-output model order-reduction. In the results section we showed that such a restarted method has potential to be developed into a viable model-reduction method. Unfortunately we ran out of time before we could do in-depth analysis and extensive testing and comparison with comparable methods, but we feel there is potential for further research and practical application of the method.

We developed a library of Matlab functions that implement the method and produce all of the results included in this document. At the core is an implementation of the band-Arnoldi algorithm of [17], which was provided by Prof. R.W. Freund. All of it is available upon request; in fact we would love to see this work developed further. One natural extension of our method would be to implement an implicit-restart scheme using something akin to the Krylov-Schur method of [46]. In order to do this over changing interpolation-points while preserving natural deflation, one would need to translate Ritz-residual vectors from one Krylov-subspace to another. The translation provided in §4.1.2 was introduced for that purpose, although we think a theory of continuous interpolation-point translation could also be developed out of that discussion.

Another promising avenue of exploration would be the basis "realification" technique via real-inner product discussed in §4.3. We have only touched the surface of it here. Realification is a topic in pure-algebra. T. Palmer discusses properties of realified spaces in [34].

Bibliography

[1] L.A. Aguirre. Quantitative measure of modal dominance for continuous systems. In *Decision and Control, 1993., Proceedings of the 32nd IEEE Conference on*, pages 2405–2410vol.3, 1993. 2.2.5

[2] Nisar Ahmed and MM Awais. Implicit restart scheme for large scale Krylov subspace model reduction method. In *Multi Topic Conference, 2001. IEEE INMIC 2001. Technology for the 21st Century. Proceedings. IEEE International*, pages 131–138. IEEE, 2001. 3.3

[3] J. I. Aliaga, D. L. Boley, R. W. Freund, and V. Hernandez. A Lanczos-type method for multiple starting vectors. *MATH. COMP*, pages 1577–1601, 2000. 1.1

[4] W. E. Arnoldi. The principle of minimized iterations in the solution of the matrix eigenvalue problem. *Q. Appl. Math*, 9(17):17–29, 1951. 1.0.2

[5] Zhaojun Bai, David Day, and Qiang Ye. ABLE: an adaptive block lanczos method for non-hermitian eigenvalue problems. *SIAM Journal on Matrix Analysis and Applications*, 20(4):1060–1082, 1999. 1.1

[6] Daniel L Boley. Krylov space methods on state-space control models. *Circuits, Systems and Signal Processing*, 13(6):733–758, 1994. 1.1

[7] Younes Chahlaoui and Paul Van Dooren. A collection of benchmark examples for model reduction of linear time invariant dynamical systems. 2002. 5.4.2

[8] W.K. Chen. *The circuits and filters handbook*. The electrical engineering handbook series. CRC Press, 2009.

[9] Jane Cullum and WE Donath. A block Lanczos algorithm for computing the q algebraically largest eigenvalues and a corresponding eigenspace of large, sparse, real symmetric matrices. In *Decision and Control including the 13th Symposium on Adaptive Processes, 1974 IEEE Conference on*, volume 13, pages 505–509. IEEE, 1974. 1.1

[10] E.J. Davison. A method for simplifying linear dynamic systems. *Automatic Control, IEEE Transactions on*, 11(1):93–101, 1966. 1.0.1

[11] David Day and Michael A. Heroux. Solving complex-valued linear systems via equivalent real formulations. *SIAM J. Sci. Comput.*, 23(2):480–498, 2001. 4.3

[12] Jack Dongarra and Francis Sullivan. Guest editors' introduction: The top 10 algorithms. *Computing in Science & Engineering*, pages 22–23, 2000. 1.0.2

[13] V. Druskin and V. Simoncini. Adaptive rational Krylov subspaces for large-scale dynamical systems. *Systems & Control Letters*, 60(8):546–560, 2011. 4.1

[14] Vladimir L Druskin and Leonid A Knizhnerman. Two polynomial methods of calculating functions of symmetric matrices. *USSR Computational Mathematics and Mathematical Physics*, 29(6):112–121, 1989. 1.0.2

[15] P. Feldmann and R.W. Freund. Efficient linear circuit analysis by pade approximation via the Lanczos process. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 14(5):639–649, 1995. 1.0.2, 2.3, 4

[16] Roland W. Freund. Krylov-subspace methods for reduced-order modeling in circuit simulation. *J. Comput. Appl. Math.*, 123(1-2):395–421, 2000. 3.4.2

[17] Roland W. Freund. Model reduction methods based on Krylov subspaces. *Acta Numerica*, 12:267–319, 2003. 5.1, 5.2.5, 6

[18] Kyle Gallivan, G Grimme, and Paul Van Dooren. A rational Lanczos algorithm for model reduction. *Numerical Algorithms*, 12(1):33–63, 1996. 4.1, 4.1.1

[19] Kyle Gallivan, A Vandendorpe, and Paul Van Dooren. Model reduction of MIMO systems via tangential interpolation. *SIAM Journal on Matrix Analysis and Applications*, 26(2):328–349, 2004. 1.1

[20] Juan M Gracia and Francisco E Velasco. Stability of invariant subspaces of regular matrix pencils. *Linear algebra and its applications*, 221:219–226, 1995. 2.2.3

[21] E Grimme and K Gallivan. Krylov projection methods for rational interpolation. 1997. 4.1

[22] E Grimme and K Gallivan. A rational Lanczos algorithm for model reduction II: Interpolation point selection. In *Numerical Algorithms*, 1998. 4.1, 4.2

[23] Eric J. Grimme. *Krylov Projection Methods for Model Reduction*. PhD thesis, University of Illinois, Urbana Champaign, Dept. of Electrical Engineering, 1997. 3

[24] Eric James Grimme, Danny C Sorensen, and Paul Van Dooren. Model reduction of state space systems via an implicitly restarted Lanczos method. *Numerical Algorithms*, 12(1):1–31, 1996. 1.0.2, 2.3, 3.3

[25] Serkan Gugercin. An iterative SVD-Krylov based method for model reduction of large-scale dynamical systems. In *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC'05. 44th IEEE Conference on*, pages 5905–5910. IEEE, 2005. 1.1

[26] Imad M Jaimoukha and Ebrahim M Kasenally. Implicitly restarted Krylov subspace methods for stable partial realizations. *SIAM Journal on Matrix Analysis and Applications*, 18(3):633–652, 1997. 1.0.2, 2.3, 3.3

[27] Hyoung M Kim and Roy R Craig. Structural dynamics analysis using an unsymmetric block Lanczos algorithm. *International journal for numerical methods in engineering*, 26(10):2305–2318, 1988. 1.1

[28] Cornelius Lanczos. *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators*. 1950. 1.0.2

[29] RB Lehoucq and KJ Maschhoff. Implementation of an implicitly restarted block Arnoldi method. *Preprint MCS-P649-0297, Argonne National Lab*, 1997. 1.2, 5.1

[30] Bruce Moore. Principal component analysis in linear systems: Controllability, observability, and model reduction. *Automatic Control, IEEE Transactions on*, 26(1):17–32, 1981. 1.0.2

[31] AA Nikishin and A Yu Yeremin. Variable block CG algorithms for solving large sparse symmetric positive definite linear systems on parallel computers, I: general iterative scheme. *SIAM journal on matrix analysis and applications*, 16(4):1135–1153, 1995. 1.1

[32] A. Odabasioglu, M. Celik, and L.T. Pileggi. PRIMA: passive reduced-order interconnect macromodeling algorithm. *IEEE Transactions on computer-aided design of integrated circuits and systems*, 17(8):645–654, 1998. 2.3

[33] Dianne P O'Leary. The block conjugate gradient algorithm and related methods. *Linear algebra and its applications*, 29:293–322, 1980. 1.1

[34] Theodore W Palmer. *Banach Algebras and the General Theory of *-algebras: Volume 1*, volume 2. Cambridge University Press, 2001. 4.3, 6

[35] Vasilios Papakos and IM Jaimoukha. A deflated implicitly restarted Lanczos algorithm for model reduction. In *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*, volume 3, pages 2902–2907. IEEE, 2003. 3.3

[36] Beresford N. Parlett and Youcef Saad. Complex shift and invert strategies for real matrices. *Linear Algebra and its Applications*, 8889(0):575–595, 1987. 4.2, 4.3

[37] Beresford N Parlett and David S Scott. The Lanczos algorithm with selective orthogonalization. *Mathematics of computation*, 33(145):217–238, 1979. 1, 5.2.5

[38] Joost Rommes, Wilhelmus H. A. Schilders, and Henk A. van der Vorst, editors. *Model Order Reduction: Theory, Research Aspects and Applications*. Springer, Berlin, 2008. 1.0.1

[39] Axel Ruhe. Implementation aspects of band Lanczos algorithms for computation of eigenvalues of large sparse symmetric matrices. *Mathematics of Computation*, 33(146):680–687, 1979. 1.1, 5.1

[40] Axel Ruhe. Rational Krylov sequence methods for eigenvalue computation. *Linear Algebra and its Applications*, 58(0):391–405, 1984. 4.1

[41] Axel Ruhe. The rational Krylov algorithm for nonsymmetric eigenvalue problems. II: Complex shifts for real matrices. 1994. 4.1, 4.2.1

[42] Yousef Saad. Krylov subspace methods for solving large unsymmetric linear systems. *Mathematics of computation*, 37(155):105–126, 1981. 1.0.2

[43] L Miguel Silveira, Mattan Kamon, Ibrahim Elfadel, and Jacob White. A coordinate-transformed Arnoldi algorithm for generating guaranteed stable reduced-order models of RLC circuits. *Computer Methods in Applied Mechanics and Engineering*, 169(3):377–389, 1999. 2.3

[44] Andreas Stathopoulos, Yousef Saad, and Kesheng Wu. Dynamic thick restarting of the Davidson, and the implicitly restarted Arnoldi methods. *SIAM J. Sci. Comput*, 19:227–245, 1996. 1.0.2, 5.2.3

[45] Gilbert W Stewart. On the sensitivity of the eigenvalue problem ax=λbx. *SIAM Journal on Numerical Analysis*, 9(4):669–686, 1972. 2.2.3

[46] GW Stewart. A Krylov–Schur algorithm for large eigenproblems. *SIAM Journal on Matrix Analysis and Applications*, 23(3):601–614, 2002. 5.2.3, 13, 6

[47] Kesheng Wu, Andrew Canning, HD Simon, and L-W Wang. Thick-restart Lanczos method for electronic structure calculations. *Journal of Computational Physics*, 154(1):156–173, 1999. 1.0.2

[48] Qian Yin and Linzhang Lu. An implicitly restarted block arnoldi method in a vector-wise fashion. 2006. 1.1, 5