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

By

EFREM B. RENSI

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

DISSERTATION

Submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

Mathematics

in the

OFFICE OF GRADUATE STUDIES

of the

UNIVERSITY OF CALIFORNIA

DAVIS

Approved:

Roland Freund

Robert Guy

Zhaojun Bai

Committee in Charge

2013

Contents

Abstract

Krylov subspace methods for model order reduction have been developed out of Krylov subspace methods for finding eigenvalues of very large, sparse matrices. The problem of finding certain eigenvalues of a large matrix and the problem 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 with large sparse matrices. Constructing a good reduced-order approximation to the model is in many ways equivalent to constructing smaller approximate system matrices that retain certain 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; generally, the smaller the better. A Krylov method iteratively searches the solution space of the model via the eigenspace of its system matrices.

A major issue is the computational expense of finding the essential eigen-information and building the reduced order model. Unmodified Krylov processes tend to become more expensive as they progress incrementally, as each iteration of the process must look at all of the information discovered in previous iterations. 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 from a new start-point that is closer to the desired destination, which is usually a particular set of eigenvalues.

An issue that is particular to model order-reduction, and not to other problems that can be solved with eigen-analysis, is that the start point 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) s_0. This is known as moment matching and is important for determining exactly *where* the reduced order approximation is accurate, and *to what degree*. Otherwise we cannot say for sure. Nevertheless, there are a few variants of restarted-Krylov subspace methods that sacrifice moment matching for restarts.

The major contribution of this dissertation is a new thick-restarted Krylov method that allows for re-starting the process at different interpolation points while preserving moment matching and some degree of linear independence of the resulting basis. It reduces computational cost by recognizing and retaining system *invariant* information and re-using it from cycle to restarted, for each different interpolation-point. The algorithms comprising the method are outlined and their efficiency is demonstrated by applying it to selected test data sets.

We also propose an alternate inner-product for orthogonalizing complex Krylov subspaces which is less computationally expensive and is applicable to any Krylov subspace projection method that requires a real-valued basis to a Krylov subspace with a complex span.

Acknowledgments

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

Maria "Bem" Cayco!!!!! who has been my friend, confidant, and mentor for years, helped me make it through my undergraduate years at San Jose State, helped me not shoot myself in the foot with graduate school applications.

My office-mate and collegue Yuji Nakatsukasa for his intricate knowlegde of numerical linear algebra and snacks from Kim's market, Boba tea, and great company while sharing an office for 5 years!

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

The Davis Bike Collective! Robert McMurry, Jason Moore, Johnathan Woolley, Matt Seitzler, Jan, Darach, Angel York, Darin Wick, Joshua Endow, Elissa, Tim Lane, Fernando Amaral, Sarah McCullough, Johnny Orozco, Jeff Hsieh, Cat Calloway.

Roy Grabow, Tim Willison

Davis community. others.

Chapter 1

1.1 Model order-reduction via Subspace Projection

Our basic problem is to approximate the (N-dimensional) Linear, Time Invariant Descriptor System

$$\begin{aligned} \boldsymbol{E}\frac{dx}{dt} &= \boldsymbol{A}x + \boldsymbol{B}u \\ y &= \boldsymbol{C}^T x \end{aligned} \tag{1.1}$$

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.

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 (1.1) 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 (1.1) is a multi-input, multi-output (MIMO) system.

We assume that the order-N system (1.1) is too large to work with and and we want a model that behaves like (1.1), but with significantly reduced state-space dimension n. For example, it may not be necessary for our approximate model to have unobservable states, or uncontrollable ones.

Suppose that, for some reason, we believe restricting the state space of the model (1.1) 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

$$\begin{aligned} \boldsymbol{E}_n\frac{d\tilde{x}}{dt} &= \boldsymbol{A}_n\tilde{x} + \boldsymbol{B}_n u \\ \tilde{y} &= \boldsymbol{C}_n^T\tilde{x}, \end{aligned} \tag{1.2}$$

with orthogonal projections

$$\boldsymbol{A}_n = V^T\boldsymbol{A}V, \quad \boldsymbol{E}_n = V^T\boldsymbol{E}V, \quad \boldsymbol{C}_n = V^T\boldsymbol{C}, \quad \boldsymbol{B}_n = V^T\boldsymbol{B}, \tag{1.3}$$

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 (1.1), given the same m input(s) $u(t) \in \mathbb{R}^m$, and ideally $\|y - \tilde{y}\|$ is small.

The reduced-order model (1.2), (1.3) is called the *explicitly-projected* ROM, because in the computational setting we must actually compute (1.3). Ultimately the desired ROM is in this form.

1.1.1 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 (1.1) 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 (1.1) 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

$$\mathcal{H}(s) = \boldsymbol{C}^T \left(s\boldsymbol{E} - \boldsymbol{A}\right)^{-1} \boldsymbol{B} \tag{1.4}$$

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

For a general MIMO transfer function (1.4) where $\boldsymbol{C} = \begin{bmatrix} \boldsymbol{c}_1 & \boldsymbol{c}_2 & \cdots & \boldsymbol{c}_p \end{bmatrix}$ and $\boldsymbol{B} = \begin{bmatrix} \boldsymbol{b}_1 & \boldsymbol{b}_2 & \ldots & \boldsymbol{b}_m \end{bmatrix}$, we can consider (1.4) 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

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

A note about MIMO models

A multi-input, multi-output (MIMO) model with m inputs and p outputs can be regarded as mp SISO models, with that many scalar-valued transfer functions $\mathcal{H}_{ij}(s)$. The example SISO transfer function `1841s11` plotted in figure **??** is actually the $(\boldsymbol{b}_1, \boldsymbol{c}_1)$ component of a MIMO model with $\boldsymbol{B} = \boldsymbol{C} \in \mathbb{R}^{1841 \times 16}$. It is a circuit model with $m = 16$ input terminals and $p = 16$ output terminals. Each component SISO model $(\boldsymbol{A}, \boldsymbol{E}, \boldsymbol{b}_i, \boldsymbol{c}_j)$ specifies frequency response of the j-th output terminal to excitation of the i-th input terminal. For a general MIMO transfer function $\mathcal{H}(s) = \boldsymbol{C}^T(s\boldsymbol{E} - \boldsymbol{A})^{-1}\boldsymbol{B}$,

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

which implies that $\mathcal{H}_{ij}(s) = \overline{\mathcal{H}_{ji}(s)}$, or more importantly,

$$|\mathcal{H}_{ij}(s)| = |\mathcal{H}_{ji}(s)|,$$

so it suffices to consider $\mathcal{H}_{ij}(s)$ only for $i < j$.

Transfer function moments

Krylov-subspace projection methods boast moment matching properties. The reduced-order model transfer function implied by a Krylov-subspacemethod is guaranteed to share a number of terms of the Taylor series about one or several points, with that of the full unreduced model.

The transfer function is a rational function, and thus can be represented by a Taylor series about an expansion-point $s_0 \in \mathbb{C}$, having the general form

$$\mathcal{H}(s) = \sum_{j=0}^{\infty} (s - s_0)^j \mathcal{H}^{(j)}, \quad \text{or equivalently,} \quad \mathcal{H}(s + s_0) = \sum_{j=0}^{\infty} s^j \mathcal{H}^{(j)} \tag{1.5}$$

where the Taylor coefficient

$$\mathcal{H}^{(j)} = \frac{1}{j!} \frac{d^j \mathcal{H}}{ds^j}\bigg|_{s=s_0} \tag{1.6}$$

is called the j-th *moment* of the transfer function about s_0.

Moment Matching

Suppose the URM (unreduced model) transfer function expressed a a Taylor series about s_0 is

$$\mathcal{H}(s) = \mathcal{H}^{(0)} + (s - s_0)\mathcal{H}^{(1)} + (s - s_0)^2 \mathcal{H}^{(2)} + \cdots + (s - s_0)^{n-1} \mathcal{H}^{(n-1)} + \cdots .$$

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

$$\widehat{\mathcal{H}}(s) = \widehat{\mathcal{H}}^{(0)} + (s - s_0)\widehat{\mathcal{H}}^{(1)} + (s - s_0)^2 \widehat{\mathcal{H}}^{(2)} + \cdots + (s - s_0)^{n-1} \widehat{\mathcal{H}}^{(n-1)} + \cdots$$

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 s_0.

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

Shift-invert representation of the transfer function

Moment matching properties of Krylov-subspacemethods are accomplished via the the following reformulation of the transfer function (1.4).

Let $s_0 \in \mathbb{C}$ be a point for which $s_0\boldsymbol{E} - \boldsymbol{A}$ is invertible. Then

$$\begin{aligned} \mathcal{H}(s) &= \boldsymbol{C}^T (s\boldsymbol{E} - \boldsymbol{A})^{-1} \boldsymbol{B} \\ &= \boldsymbol{C}^T (s_0\boldsymbol{E} - \boldsymbol{A} + (s - s_0)\boldsymbol{E})^{-1} \boldsymbol{B} \\ &= \boldsymbol{C}^T (I - (s - s_0)\mathbf{H})^{-1} \mathbf{R} \end{aligned} \tag{1.7}$$

where[1]

$$\mathbf{H} := (\boldsymbol{A} - s_0\boldsymbol{E})^{-1}\boldsymbol{E} \quad \text{and} \quad \mathbf{R} := (s_0\boldsymbol{E} - \boldsymbol{A})^{-1}\boldsymbol{B}. \tag{1.8}$$

[1] To verify (1.7), note that $I = (s_0\boldsymbol{E} - \boldsymbol{A})^{-1}(s_0\boldsymbol{E} - \boldsymbol{A})$.

(1.7) is sometimes called the shifted transfer function formulation, with shift s_0, although it does not depend on s_0. The shift matters when we consider the ROM transfer function that approximates (1.7).

The generally non-sparse $\mathbf{H} = \mathbf{H}(s_0) \in \mathbb{C}^{N\times N}$ is a sort of operator or multiplier that acts on $\mathbf{R} = \mathbf{R}(s_0) \in \mathbb{C}^{N\times p}$. $\mathbf{H}$ 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 $v \in \mathbb{C}^N$. $\mathbf{H}$ and $\mathbf{R}$ are the building blocks for the moments of the transfer function about s_0.

The shifted transfer function representation (1.7) can alternatively be considered the transfer function for the shifted descriptor system

$$\begin{aligned} \mathbf{H}\frac{dx}{dt} &= (I - s_0\mathbf{H})x + \mathbf{R}u \\ y &= \boldsymbol{C}^T x, \end{aligned} \tag{1.9}$$

which is equivalent to (1.1) for any s_0 such that $\boldsymbol{A} - s_0\boldsymbol{E}$ is invertible. This is notable because some order-reduction schemes 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 will call such a ROM *implicitly* projected on to span V, as opposed to the *explicitly* projected model (1.2). A model obtained via implicit projection is not equivalent to (1.2) in general and is undesirable for some applications, but is much cheaper to construct.

Moment representation

We now express transfer function moments about s_0 in terms of $\mathbf{H}$ and $\mathbf{R}$. Via Neumann series expansion (power series for matrices) re-write (1.7) as

$$\begin{aligned} \mathcal{H}(s) &= \boldsymbol{C}^T \left(\sum_{j=0}^{\infty} (s - s_0)^j \mathbf{H}^j \right) \mathbf{R} \\ &= \sum_{j=0}^{\infty} (s - s_0)^j \boldsymbol{C}^T \mathbf{H}^j \mathbf{R}. \end{aligned} \tag{1.10}$$

The moments $\mathcal{H}^{(j)}$ from (1.5) are specified exactly in (1.10):

$$\mathcal{H}^{(j)} = \boldsymbol{C}^T \mathbf{H}^j \mathbf{R}. \tag{1.11}$$

Region of convergence for moment matching

The power (Taylor) series representation seems to imply that (1.10) is only valid for s in a disc of radius $1/\left\|\mathbf{H}\right\|_{op}$ around s_0, where the operator norm

$$\left\|\mathbf{H}\right\|_{op} = \sup_{v\neq 0} \left\{ \frac{\|\mathbf{H}v\|}{\|v\|} \right\}.$$

Then certainly $\left\|\mathbf{H}\right\|_{op} \geq |\lambda_1|$, where λ_1 is the largest eigenvalue of $\mathbf{H}$. Equivalently, $\left\|\mathbf{H}\right\|_{op} \geq 1/|(\mu_1 - s_0)|$ where μ_1 is the closest pole to s_0. Thus, the region of convergence for (1.10) is the largest disc centered at s_0 that does not contain a pole. The closer s_0 is to a pole of the transfer

function, the smaller region of convergence we theoretically have for moment matching about s_0. In practice Krylov-subspacemethods are observed to converged well outside of the theoretical region of convergence.

1.1.2 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

$$\mathbf{H}\mathcal{Q} \subseteq \mathcal{Q}. \tag{1.12}$$

The span of a set of eigenvectors of $\mathbf{H}$ is an ($\mathbf{H}$-)invariant subspace, and an invariant subspace always has a basis consisting of eigenvectors of $\mathbf{H}$.

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

$$\mathbf{H}Q = QT \tag{1.13}$$

for some matrix $T \in \mathbb{C}^{\ell\times\ell}$. If the basis vectors are eigenvectors Z then (1.13) becomes

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

where $\Lambda = \operatorname{diag}\{\lambda_1, \lambda_2, ..., \lambda_\ell\}$ 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, (1.13) is called a Schur-decomposition, and T is upper triangular with eigenvalues λ_j associated with z_j along its diagonal. A Schur decomposition is often preferred over an eigen-decomposition because it is easier to compute and Schur vectors are more numerically stable.

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 called right-eigenvectors since they multiply from the right.

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, [30] with respect to $(\boldsymbol{A}, \boldsymbol{E})$ if

$$\dim(\boldsymbol{A}\mathcal{Q} + \boldsymbol{E}\mathcal{Q}) \leq \dim \mathcal{Q}. \tag{1.14}$$

For a regular matrix pencil $(\boldsymbol{A}, \boldsymbol{E})$ and any $s_0 \in \mathbb{C}$ that is not an eigenvalue of $(\boldsymbol{A}, \boldsymbol{E})$ it is shown in [12] that $(\boldsymbol{A}, \boldsymbol{E})$-invariance is equivalent to $\mathbf{H}$-invariance for

$$\mathbf{H} = (\boldsymbol{A} - s_0\boldsymbol{E})^{-1}\boldsymbol{E}, \tag{1.15}$$

which happens to be the shift-invert operator defined by (1.8).

Then $\mathbf{H}(s_0) = (\boldsymbol{A} - s_0\boldsymbol{E})^{-1}\boldsymbol{E}$ has the same eigenvectors (regardless of s_0) 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

$$\lambda = \frac{1}{\mu - s_0}, \qquad \mu = s_0 + \frac{1}{\lambda} \tag{1.16}$$

and they share the same eigenvector. That is,

$$\begin{aligned} \mathbf{H}z &= \lambda z \\ &= \left(\frac{1}{\mu - s_0}\right) z \end{aligned} \quad \Longleftrightarrow \quad \begin{aligned} \boldsymbol{A}z &= \mu \boldsymbol{E} z \\ &= \left(s_0 + \frac{1}{\lambda}\right) \boldsymbol{E} z \end{aligned}$$

To see this, observe that

$$\begin{aligned} (\mu \boldsymbol{E} - \boldsymbol{A}) &= [(\mu - s_0)\boldsymbol{E} + (s_0 \boldsymbol{E} - \boldsymbol{A})] \\ &= [(\mu - s_0) \underbrace{(s_0 \boldsymbol{E} - \boldsymbol{A})^{-1} \boldsymbol{E}}_{-\mathbf{H}} + I] \\ &= [(\mu - s_0)\mathbf{H} - I] \\ &= \left[\mathbf{H} - \left(\frac{1}{\mu - s_0}\right) I\right] \\ &= (\mathbf{H} - \lambda I). \end{aligned} \tag{1.17}$$

In other words, invariant subspaces under the shifted operator $\mathbf{H}(s_0)$ are shift-invariant. This is useful because if we decide to change the shift s_0, any previously discovered invariant subspace will be still be invariant under the new operator.

1.1.3 Pole-Residue representation

Recall the transfer function

$$\mathcal{H}(s) = \boldsymbol{C}^T (s\boldsymbol{E} - \boldsymbol{A})^{-1} \boldsymbol{B}, \tag{1.4}$$

which includes the linear matrix pencil $(\boldsymbol{A}, \boldsymbol{E})$.

The $\boldsymbol{A}$ and $\boldsymbol{E}$ that arise from Modified Nodal Analysis [16] circuit representations are singular in general. In particular, $\boldsymbol{E}$ is singular, which means $(\boldsymbol{A}, \boldsymbol{E})$ has eigenvalues at ∞ at least some of which are associated with the null space of $\boldsymbol{E}$.

Poles and residues of the standard transfer function formulation

Let us assume that $(\boldsymbol{A}, \boldsymbol{E})$ has a full eigen-decomposition

$$\boldsymbol{A}Z = \boldsymbol{E}Z\mathcal{M} \quad \text{and} \quad \boldsymbol{A}^T W = \boldsymbol{E}^T W \mathcal{M}$$

where $Z, W \in \mathbb{C}^{N \times N}$ represent the right and left eigenspaces of $(\boldsymbol{A}, \boldsymbol{E})$, respectively, and $\mathcal{M}$ is the diagonal matrix of eigenvalues $\mu_j \in \mathbb{C} \cup \{\infty\}$. Note that since $\boldsymbol{A}$ and $\boldsymbol{E}$ are real, every eigenvalue is real, infinite, or is one of a complex conjugate pair.

The left and right eigenvectors are orthogonal, and can be scaled[2] so that $W^H \boldsymbol{E} Z = I$. Then

$$\begin{aligned} \mathcal{H}(s) &= \boldsymbol{C}^T (s\boldsymbol{E} - \boldsymbol{A})^{-1} \boldsymbol{B} \\ &= \boldsymbol{C}^T (W^{-H}(sI - \mathcal{M})Z^{-1})^{-1} \boldsymbol{B} \\ &= \boldsymbol{C}^T Z (sI - \mathcal{M})^{-1} W^H \boldsymbol{B} \\ &= \boldsymbol{C}^T Z \left(\sum_{j=1}^{q} \frac{1}{s - \mu_j}\right) W^H \boldsymbol{B}. \end{aligned} \tag{1.18}$$

[2] This scaling is not generally the default for eigensolver algorithms.

The pole (eigen) decomposition (1.18) of the transfer function suggests that it can be approximated using eigen-pairs of $(\boldsymbol{A}, \boldsymbol{E})$.

For the sake of expressing (1.18) without clutter we assumed all eigenvalues are simple and finite. Actually, multiple eigenvalues are possible and eigevalues at infinity are inevitable because $\boldsymbol{E}$ is singular. The eigenspace associated with infinite eigenvalues is the nullspace of $\boldsymbol{E}$.

Recall the left and right hand sides $\boldsymbol{C}^T Z \in \mathbb{C}^{p\times N}$ and $W^H \boldsymbol{B} \in \mathbb{C}^{N\times m}$ from (1.18) and consider the partitions

$$\boldsymbol{C}^T Z = \begin{bmatrix} \hat{c}_1 & \hat{c}_2 & \cdots & \hat{c}_N \end{bmatrix}$$

and

$$\boldsymbol{B}^T W = \begin{bmatrix} \hat{b}_1 & \hat{b}_2 & \cdots & \hat{b}_N \end{bmatrix}$$

into N columns.

Then we can express the the pole-residue form of transfer function (1.4) as

$$\mathcal{H}(s) = \sum_{\mu_j=\infty} \hat{c}_j \hat{b}_j^T + \sum_{\mu_j\neq\infty} \frac{\hat{c}_j \hat{b}_j^T}{s-\mu_j}. \tag{1.19}$$

Note that $\hat{c}_j$ and $\hat{b}_j$ are scalars if this is a SISO model. Generally, $\hat{c}_j \hat{b}_j^T \in \mathbb{C}^{p\times m}$. The transpose $\hat{b}_j^T$ is in fact a transpose and not a conjugate-transpose, even if $\hat{b}_j$ is complex-valued.

Our necessary assumption $W^H \boldsymbol{E} Z = I$ is not the default scaling for eigensolvers in practice. In that case, we must consider the scaling factor

$$\xi_j = 1/w_j^H \boldsymbol{E} z_j$$

for $j = 1, 2, \ldots, q$ and (1.19) generalizes to

$$\mathcal{H}(s) = \sum_{\mu_j=\infty} \xi_j \hat{c}_j \hat{b}_j^T + \sum_{\mu_j\neq\infty} \xi_j \frac{\hat{c}_j \hat{b}_j^T}{s-\mu_j}. \tag{1.20}$$

Poles and residues from shifted transfer-function formulation

For model reduction it is often favorable to work with the so-called s_0-shifted transfer function

$$\mathcal{H}(s) = \boldsymbol{C}^T \left(I - (s-s_0)\mathbf{H}\right)^{-1} \mathbf{R}, \tag{1.7}$$

rather than the standard formulation (1.4).

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

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

for a $N \times N$ diagonal matrix $\Lambda = \begin{bmatrix} \lambda_1, \lambda_2, \ldots, \lambda_N \end{bmatrix}$ of eigenvalues.

Then

$$\begin{aligned} \mathcal{H}(s) &= \boldsymbol{C}^T (I - (s-s_0)\mathbf{H})^{-1} \mathbf{R} \\ &= \boldsymbol{C}^T \left(Z(I - (s-s_0)\Lambda) Z^{-1}\right)^{-1} \mathbf{R} \\ &= (\boldsymbol{C}^T Z) \Delta(s) (Z^{-1}\mathbf{R}) \end{aligned} \tag{1.21}$$

where $\Delta(s) = (I - (s - s_0)\Lambda)^{-1}$ is a diagonal matrix with diagonal entries $\delta_j(s) = 1 - (s - s_0)\lambda_j$, or equivalently

$$\delta_j(s) = \begin{cases} \dfrac{s_0 - \mu_j}{s - \mu_j}, & \mu_j \neq \infty \\ 1, & \mu_j = \infty. \end{cases} \tag{1.22}$$

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

$$\begin{aligned} \mathcal{H}(s) &= \sum_j \frac{\hat{f}_j \hat{g}_j^T}{1 - (s - s_0)\lambda_j} && (1.23) \\ &= \sum_{\lambda_j = 0} \hat{f}_j \hat{g}_j^T + \sum_{\lambda_j \neq 0} \frac{s_0 - \mu_j}{s - \mu_j} \hat{f}_j \hat{g}_j^T \\ &= \sum_j \delta_j(s) \hat{f}_j \hat{g}_j^T, && (1.24) \end{aligned}$$

where $\delta_j(s)$ is from (1.22). 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 eiqenvalue $\lambda_j = 0$ of $\mathbf{H}$ corresponds to an infinite $\mu_j = \infty$ pole (eigenvalue of $(\boldsymbol{A}, \boldsymbol{E})$).

1.1.4 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 formulations (1.20) and (1.24) suggest a hierarchy of poles' importance for approximation.

We will define a measure of pole-dominance, which we will call its *mass* or *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.

If we take the norm of (1.24) over the interval $i[\omega_1, \omega_2]$ of interest on the $\Im$-axis, we have

$$\|\mathcal{H}(i\omega)\|_\infty \leq \sum_j \|\delta_j(i\omega)\|_\infty \|\hat{f}_j\|_1 \|\hat{g}_j\|_1, \tag{1.25}$$

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

$$\gamma_j = \|\delta_j(i\omega)\|_\infty \|\hat{f}_j\|_1 \|\hat{g}_j\|_1 \tag{1.26}$$

indicates that μ_j is a so-called dominant pole.

For a SISO model, $\hat{f}_j$ is a scalar so $\|\hat{f}_j\|_1 = |\hat{f}_j|$ and it represents the weighting of the pole μ_j by the left-hand multiplier $\boldsymbol{C} = \boldsymbol{c}$ of the transfer function (1.4).

A MIMO system has p such left-multipliers in the form of $\boldsymbol{C} = \begin{bmatrix} \boldsymbol{c}_1 & \boldsymbol{c}_2 & \cdots & \boldsymbol{c}_p \end{bmatrix}$ and each one has an associated element in the column vector $\hat{f}_j = (\hat{f}_{1j}, \hat{f}_{2j}, \ldots, \hat{f}_{pj}) \in \mathbb{C}^p$. By summing them we get an overall sense of how much μ_j is favored by $\boldsymbol{C}$. Thus we use the 1-norm (column-sum)

$$\|\hat{f}_j\|_1 = \sum_i |\hat{f}_{ij}|.$$

The reasoning for using $\|\hat{g}_j\|_1$ in (1.25) is similar.

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

$$\begin{aligned} \|\delta_j(i\omega)\|_\infty &= \max_{\omega \in [\omega_1, \omega_2]} |\delta_j(i\omega)| \\ &= \begin{cases} \dfrac{|s_0 - \mu_j|}{\min\{|\mu_j - \omega_1|, |\mu_j - \omega_2|, |\Re(\mu_j)|\}}, & \mu_j \neq \infty \\ 1, & \mu_j = \infty \end{cases} \end{aligned} \tag{1.27}$$

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 1.1. 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-mass

The right-hand-side of (1.25) (i.e. $\sum_j \gamma_j$), is the total mass of the system (1.1) with respect to $i[\omega_1, \omega_2]$, and we attempted to use it as a measure of ROM convergence. The combined weight of a few dominant poles often comprises most of a system's total mass.

1.1.5 ROM transfer function via projection

The URM (unreduced model) transfer function formulations via explicit (1.4) and implicit (1.7) projection exist only in theory for large applications, which can be on the order of 10^9 at the time of this writing. The two transfer functions (1.4) and (1.7) are mathematically equivalent. Subspace projected ROM transfer functions can be obtained via *explicit* projection, or *implicit* projection, yielding two forms similar to (1.4) and (1.7) that are *not* mathematically equivalent but they converge to the URM (unreduced) transfer function (1.4), (1.7) as the projection subspace approaches $\mathbb{C}^N$. We will refer to the two formulations as the implicit, and explicit transfer functions associated with a given subspace for projection.

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

$$\boldsymbol{A}_n := V^T \boldsymbol{A} V, \quad \boldsymbol{E}_n := V^T \boldsymbol{E} V, \quad \boldsymbol{C}_n := V^T \boldsymbol{C}, \quad \boldsymbol{B}_n := V^T \boldsymbol{B}, \tag{1.3}$$

of realization $(\boldsymbol{A}, \boldsymbol{E}, \boldsymbol{B}, \boldsymbol{C})$ on to $\mathcal{K}$,[3] the **explicitly projected** model $(\boldsymbol{A}_n, \boldsymbol{E}_n, \boldsymbol{B}_n, \boldsymbol{C}_n)$ has transfer function

$$\widehat{\mathcal{H}}_n(s) = \boldsymbol{C}_n^T \left(s\boldsymbol{E}_n - \boldsymbol{A}_n\right)^{-1} \boldsymbol{B}_n. \tag{1.28}$$

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

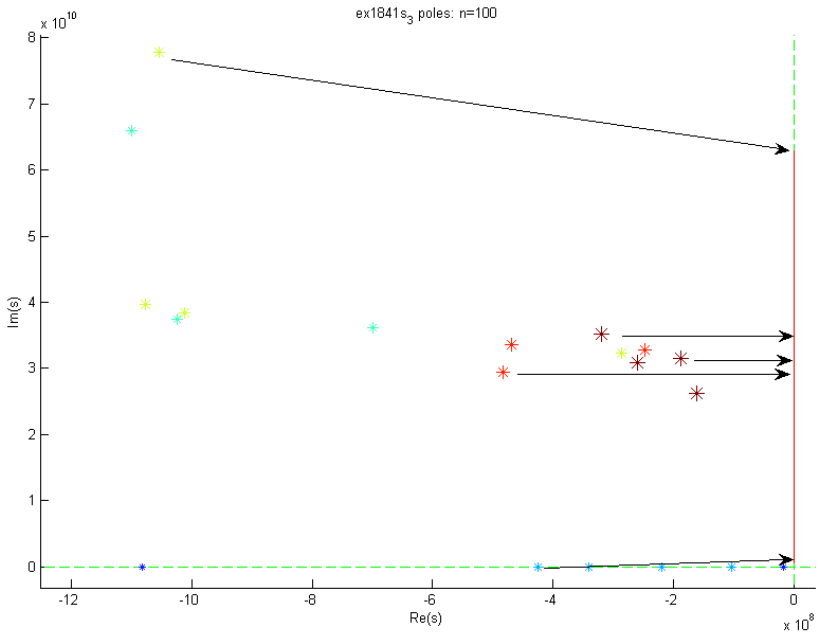


Figure 1.1: The red (solid) segment on the $\Im$-axis is the segment $i(\omega_0, \omega_1)$ of interest. In this case it looks like it extends to the origin but it does not. The arrows indicate how we define a pole's distance to the segment, which we use to determine the pole's weight. Conjugate pairs 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.

The explicitly projected model has the property that the projected ROM (1.28) of a stable system is stable if the projection basis V is real-valued, which is not true in general of the implicit projected transfer function, described next.

Some iterative subspace methods, notably Krylov-subspacemethods, use the operator $\mathbf{H}$ and operand $\mathbf{R}$ to construct a basis V and a projected operator matrix $\widetilde{\mathbf{H}} \in \mathbb{C}^{n\times n}$ such that

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

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

$$\widetilde{\mathcal{H}}_n(s) = \boldsymbol{C}_n^H \left(I - (s - s_0)\widetilde{\mathbf{H}}\right)^{-1} \widetilde{\boldsymbol{\rho}}_n, \tag{1.29}$$

where

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

(1.29) is the projected ROM analog to the "shifted" transfer function (1.7).

We say (1.29) 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 (1.3). The reduced model implied by (1.29) is actually

$$\begin{aligned} \widetilde{\mathbf{H}} \frac{d\tilde{x}}{dt} &= (I + s_0 \widetilde{\mathbf{H}})\tilde{x} + \widetilde{\boldsymbol{\rho}}_n u \\ \hat{y} &= \boldsymbol{C}_n^T \tilde{x}, \end{aligned} \tag{1.30}$$

which is not equivalent to the explicitly projected system

$$\begin{aligned} \boldsymbol{E}_n \frac{dz}{dt} &= \boldsymbol{A}_n z + \boldsymbol{B}_n u \\ \hat{y} &= \boldsymbol{C}_n^T z, \end{aligned} \tag{1.2}$$

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 (1.1).

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 [29, 15, 17] have been made to remedy this situation, but these methods tend to sacrifice moment matching properties in the process. The reduced models produced are still pretty good, but do not have proven error bounds. As a result the explicitly projected model is generally preferred in practice, although it requires quite a bit more computation to carry out the projections (1.3).

Chapter 2

2.1 Krylov-subspace projection methods

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

$$\mathcal{H}(s) = \boldsymbol{C}^T (s\boldsymbol{E} - \boldsymbol{A})^{-1} \boldsymbol{B} \tag{1.4}$$
$$= \boldsymbol{C}^T (I - (s - s_0)\mathbf{H})^{-1} \mathbf{R} \tag{1.7}$$

with

$$\mathbf{H} = (\boldsymbol{A} - s_0\boldsymbol{E})^{-1}\boldsymbol{E} \quad \text{and} \quad \mathbf{R} = (s_0\boldsymbol{E} - \boldsymbol{A})^{-1}\boldsymbol{B}, \tag{1.8}$$

where (1.4) is the standard formulation and (1.7) is the so-called s_0-shifted formulation, and $\mathbf{H}$ sometimes called a shift-inverse operator. We approximate $\mathcal{H}(s)$ by approximating $\mathbf{H}$ and $\mathbf{R}$, since it is known that successive applications of $\mathbf{H}$ to $\mathbf{R}$, as in $\mathbf{HR}, \mathbf{H}^2\mathbf{R}, \ldots$, creates progressively better approximations to the spectrum of $\mathbf{H}$.

A Krylov-subspacemethod 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

$$\mu = s_0 + 1/\lambda$$

of $(\boldsymbol{A}, \boldsymbol{E})$ which are closest to s_0. 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 s_0 with each additional moment.

A fundamental feature (and drawback) of Krylov-subspacemethods for model order-reduction is that for a given shift s_0 there is only one way for the method to progress: from poles μ closest to s_0 (i.e. smallest $|\mu - s_0|$), to those farthest away. This presents a problem because often only a few dominant poles (§1.1.4) 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 s_0) by several insignificant poles, as in

$$|\boldsymbol{\mu}_1 - s_0| > |\mu_3 - s_0| > |\mu_4 - s_0| > \cdots > |\mu_\ell - s_0| > |\boldsymbol{\mu}_2 - s_0|.$$

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 s_0 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.

Krylov subspace

Krylov-subspace projection methods are basically developed out of the power iteration with $\mathbf{H}$ and $\mathbf{R} = \begin{bmatrix}\mathbf{r}_1 & \mathbf{r}_2 & \cdots & \mathbf{r}_m\end{bmatrix}$. 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-subspaceis the ideal subspace to use for projection because reduced-order models obtained via Krylov subspaces have moment matching properties (§1.1.1). reduced-order models obtained by projection on to 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

$$\mathcal{K}_n(\mathbf{H}, \mathbf{r}) = \text{span}\left\{\mathbf{r}, \mathbf{H}\mathbf{r}, \mathbf{H}^2\mathbf{r}, \ldots, \mathbf{H}^{n-1}\mathbf{r}\right\}. \tag{2.1}$$

For the n-th *band* Krylov subspace

$$\mathcal{K}_n(\mathbf{H}, \mathbf{R}) = \text{span}\left\{\mathbf{R}, \mathbf{H}\mathbf{R}, \mathbf{H}^2\mathbf{R}, \ldots \mathbf{H}^{\eta}\mathbf{R}\right\} \tag{2.2}$$

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 MOR.

2.1.1 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

$$\widetilde{\mathbf{H}} = V^T\mathbf{H}V,$$

which is an upper-Hessenberg matrix. An upper-Hessenberg matrix such as

$$\mathbf{H}_4 = \begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{14} \\ h_{21} & h_{22} & h_{23} & h_{24} \\ & h_{32} & h_{33} & h_{34} \\ & & h_{43} & h_{44} \end{bmatrix}$$

is like an upper-triangular matrix, but with nonzero entries on the 1st subdiagonal.

The Arnoldi process (Algorithm 1) basically performs a power iteration and orthogonalizes each iterate against previous ones, thus producing an orthonormal basis for (2.1). 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 orthogonalizing methods. A notable alternative uses Householder reflectors making for a more stable and more costly method.

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 $\widetilde{\mathbf{H}} = V^T\mathbf{H}V$

1 $r_0 := \mathbf{r}$
2 $v_1 := r_0/\left\|r_0\right\|_2$
3 **for** $k = 1$ **to** n **do**
4 $\quad r_k := \mathbf{H}v_k$
5 $\quad$ **for** $j = 1$ **to** k **do** `% Make` r_k `orthogonal to previous` $\{v_1, v_2..., v_k\}$
6 $\quad\quad h_{jk} := v_k^H r_k$
7 $\quad\quad r_k := r_k - h_{jk}v_j$
8 $\quad$ **if** $\left\|r_k\right\|_2 \neq 0$ **then**
9 $\quad\quad h_{j+1,j} := \left\|r_k\right\|_2$
10 $\quad\quad v_{k+1} := r_k/\left\|r_k\right\|_2$
11 $\quad$ **else** exit k-loop
12 **return** $V = \begin{bmatrix} v_1 & v_2 & \cdots & v_n \end{bmatrix}$, $\hat{v}_n = v_{n+1}h_{n+1,n}$, $\widetilde{\mathbf{H}} = \begin{bmatrix} h_{ij} \end{bmatrix}$

2.1.2 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. Clearly the computational cost of Arnoldi is dominated by matvecs. It should be noted that the s_0-shifted inverse operator $\mathbf{H} = (\boldsymbol{A} - s_0\boldsymbol{E})^{-1}\boldsymbol{E}$ 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

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

flops, where $\mathrm{nnz}(T) \leq N(N+1)/2$ 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 don't commit a major crime by viewing the operation as a matrix-vector product, as long as we consider the one-time cost of sparse $LU = P\mathbf{H}Q$ factorization.

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

The Arnoldi algortihm requires $(n+1)N$ units of storage for the basis vectors $v_j \in V$, which is 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 $\mathbf{H}$, each $\mathcal{O}\left(N^2\right)$. 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.

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.

Next, 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.

Arnoldi relation

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

$$\begin{aligned} \mathbf{H}V &= \begin{bmatrix} V & v_{n+1} \end{bmatrix} \begin{bmatrix} \longleftarrow & \widetilde{\mathbf{H}} & \longrightarrow \\ 0 & \cdots & h_{n+1,n} \end{bmatrix} \\ &= V\widetilde{\mathbf{H}} + \ h_{n+1,n} v_{n+1} e_n^T \\ &= V\widetilde{\mathbf{H}} + \ \hat{v}_n e_n^T \end{aligned} \tag{2.3}$$

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.

Residual vector The last (n-th) candidate-vector $\hat{v}_n = h_{n+1,n} v_{n+1}$ of algorithm 1 is a notable quantity because it represents the error of the approximation $\mathbf{H}V \approx V\widetilde{\mathbf{H}}$ (of V to an invariant subspace). One expects the sequence to decrease in general, since $\|\hat{v}_n\|$ is zero for $n \geq d(\mathbf{H}, \mathbf{r}) \leq N$, but it is not monotonically decreasing. From a model reduction standpoint, a satisfactory model can be obtained without having n large enough to make $\|\hat{v}_n\|$ small. This is because $\|\hat{v}_n\|$ represents the amount of new spectral information of $\mathbf{H}$ discovered on the n-th step, after previously discovered directions $v_j, j = 1, 2, ..., n$ have been subtracted off. A rapidly decreasing $\|\hat{v}_n\|$ indicates that further iterations with $\mathbf{H}$ are not producing much new spectral information. Recall that eigenvalues of $\mathbf{H}$ correspond to poles of the transfer function. As long as new poles are being discovered (starting

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

from s_0 and moving outward), $\hat{v}_n$ is rich with information. The poles being discovered may or may not be significant in the sense of frequency response approximation, however.

Rayleigh-Ritz (approximate eigenvalues) from Arnoldi

The Arnoldi relation (2.3) (or Krylov relation in general) is what makes Krylov subspaces ideal for subspace projection eigenvalue methods. Compared to $\mathbf{H}$, the matrix $\widetilde{\mathbf{H}}$ is small enough for its eigenvalue decomposition

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

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 (2.3) with W yields

$$\begin{aligned} \mathbf{H}VW &= V\widetilde{\mathbf{H}}W + \hat{v}_n e_n^T W \\ &= VW\Lambda + \hat{v}_n e_n^T W \end{aligned} \tag{2.4}$$

$$\mathbf{H}Z = Z\Lambda + \hat{v}_n e_n^T W \tag{2.5}$$

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 (2.5) is a scalar multiple of the residual-vector $\hat{v}_n$. Assuming $\|z_j\|_2 = 1$, the Arnoldi-relation (2.3) thus implies a simple formulation

$$\mathrm{rr}_j = \frac{\|\mathbf{H}z_j - \lambda_j z_j\|_2}{\|\lambda_j z_j\|_2} = \frac{|\xi_j|}{|\lambda_j|}\|\hat{v}_n\|_2 = \frac{|\xi_j|}{|\lambda_j|}|h_{n+1,n}| \tag{2.6}$$

for the relative residual-errors of the Ritz-values/vectors. Ritz-pairs with low associated relative residual norms (2.6) are good approximations to eigenvalues/vectors of $\mathbf{H}$ if they are well-conditioned. This is because (2.6) 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 (2.3) 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 (2.6) 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] vectors $w \in W$ are sometimes called short Ritz-vectors.

Implicit vs. Explicit Ritz-values and vectors

It should be noted that although eigenvalues λ of $\mathbf{H} = (\boldsymbol{A} - s_0\boldsymbol{E})^{-1}\boldsymbol{E}$ and μ of $(\boldsymbol{A}, \boldsymbol{E})$ are related by $\lambda = 1/(\mu - s_0)$ 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 $(\boldsymbol{A}_n, \boldsymbol{E}_n) = (V^T\boldsymbol{A}V, V^T\boldsymbol{E}V)$.

The reader should note that 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: The set of **implicit Ritz-values**

$$\left\{ 1/\hat{\lambda} + s_0 \;\middle|\; \hat{\lambda} \in \sigma(\widetilde{\mathbf{H}}) \right\} \tag{2.7}$$

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

$$\widetilde{\mathbf{H}} = V^T\mathbf{H}V = V^T(\boldsymbol{A} - s_0\boldsymbol{E})^{-1}\boldsymbol{E}V$$

is a byproduct of constructing V by n steps of the Arnoldi algorithm, and the set of of **explicit Ritz-values**

$$\{ \hat{\mu} \in \sigma(\boldsymbol{A}_n, \boldsymbol{E}_n) \} \tag{2.8}$$

(of (A, 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. (2.7) and (2.8) are not equal in general but are related in that they both converge to the same spectrum $\sigma(\boldsymbol{A}, \boldsymbol{E})$. Note that both sets of approximate eigenvalues are dependent on s_0; we expect eigenvalue approximations closer to s_0 to be more accurate for both (2.7) and (2.8), because they both result from projection on to the Krylov-subspace$\mathcal{K}_n(\boldsymbol{A}, \mathbf{R})$, where $\mathbf{H} = \mathbf{H}(s_0)$ and $\mathbf{R} = \mathbf{R}(s_0)$.

The values of the associated approximate eigenvectors are not dependent on s_0. Only the order in which they converge depends on s_0. Eigenvectors associated with (2.7) and with (2.8) 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 s_0 and have low relative residual-error norm (2.6). Thus, if an eigen-pair $(\hat{\lambda}_j, w_j)$ of $\widetilde{\mathbf{H}}$ is converged, then we can expect that $(s_0 + 1/\hat{\lambda}_j, Vw_j)$ 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 care about both sets of approximate eigenvalues/vectors is that implicit (2.7) Ritz-values/vectors are far cheaper to compute than the explicit variety (2.8), but the explicit formulation (1.2) 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 $1/\hat{\lambda} + s_0$ with positive real part, the implicitly projected model (1.29) 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 [15, 24, 17, 2], have been developed which attempt to purge subspace

[4]Implicitly projected ROMs, such as those produced by PVL [7] often work fine in many practical applications despite being unstable, but they are currently unpopular.

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 $\mathcal{K}_n(\boldsymbol{A}(s_0), \mathbf{R}(s_0)) \subseteq \operatorname{span} V$, the explicitly projected ROM on to V is guaranteed to be of matrix-Padé-type (match moments) with respect to s_0.

2.1.3 Moment matching property of Krylov-subspaceprojected models

We are going to prove 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-subspacematches l moments about s_0, 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 (1.29) in Theorem 1, and explicitly projected ROMs (1.28) in Theorem 2. It is interesting to note that both implicitly projected and and explicitly projected ROMs match the same moments about an expansion-point. Significant differences in the two ROM approximations are present away from expansion-point(s) s_0, but near s_0 they are approximations of the same order.

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 (1.1), and its equivalent shift-invert formulation $\mathcal{H}(s) = \boldsymbol{C}^T \left(I - (s - s_0)\mathbf{H}\right)^{-1} \mathbf{R}$ with shift s_0, or

$$\begin{aligned} \mathcal{H}(s + s_0) &= \boldsymbol{C}^T \left(I - s\mathbf{H}\right)^{-1} \mathbf{R} && (2.9) \\ &= \sum_{j=0}^{\infty} s^j \mathcal{H}^{(j)} && (1.5) \end{aligned}$$

where (1.5) is the Taylor series expansion of $\mathcal{H}(s)$ about $s_0 \in \mathbb{C}$. The j-th moment $\mathcal{H}^{(j)}$ was shown in §1.1.1 to be

$$\mathcal{H}^{(j)} = \boldsymbol{C}^T \mathbf{H}^j \mathbf{R}. \tag{2.10}$$

Moment matching of the implicitly projected ROM

The implicitly projected ROM transfer function

$$\widetilde{\mathcal{H}}(s + s_0) = \boldsymbol{C}_n^T \left(I - s\widetilde{\mathbf{H}}\right)^{-1} \widetilde{\boldsymbol{\rho}}_n \tag{2.11}$$

is defined via projection of (2.9), as

$$\widetilde{\mathbf{H}} = V^T \mathbf{H} V, \quad \boldsymbol{C}_n = V^T \boldsymbol{C}, \quad \widetilde{\boldsymbol{\rho}}_n = V^T \mathbf{R} \tag{2.12}$$

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 (2.11) about s_0 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 $\widetilde{\mathcal{H}}^{(j)}(s_0)$ of the implicitly projected ROM transfer function (1.29), (2.11) and moments $\mathcal{H}^{(j)}(s_0)$ of the URM transfer function (1.4) about s_0 are related by*

$$\widetilde{\mathcal{H}}^{(j)} = \boldsymbol{C}_n^T \widetilde{\mathbf{H}}^j \widetilde{\boldsymbol{\rho}}_n = \boldsymbol{C}^T \mathbf{H}^j \mathbf{R} = \mathcal{H}^{(j)} \tag{2.13}$$

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

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

$$\mathbf{H}^j \mathbf{R} = V \widetilde{\mathbf{H}}^j \widetilde{\boldsymbol{\rho}}_n \tag{2.14}$$

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

For $j = 0$, (2.14) follows from (2.12). Now assume (2.14) holds for some $j \in \{0, 1, \ldots, l-2\}$. Applying $\mathbf{H}$ to (2.14) yields

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

□

Moment matching of the explicitly projected ROM

Proof of moment matching for the explicitly projected ROM (1.2) 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 [9, proposition 6 and theorem 7].

Recall the explicitly-projected ROM (1.2) with transfer function

$$\widehat{\mathcal{H}}_n(s) = \boldsymbol{C}_n^T \left(s\boldsymbol{E}_n - \boldsymbol{A}_n\right)^{-1} \boldsymbol{B}_n, \tag{1.28}$$

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 (1.28) are

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

where the structures

$$\widehat{\mathbf{H}} := (\boldsymbol{A}_n - s_0 \boldsymbol{E}_n)^{-1} \boldsymbol{E}_n \quad \text{and} \quad \widehat{\boldsymbol{\rho}}_n := (s_0 \boldsymbol{E}_n - \boldsymbol{A}_n)^{-1} \boldsymbol{B}_n. \tag{2.15}$$

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

$$\mathbf{H} := (\boldsymbol{A} - s_0 \boldsymbol{E})^{-1} \boldsymbol{E}, \quad \mathbf{R} := (s_0 \boldsymbol{E} - \boldsymbol{A})^{-1} \boldsymbol{B} \tag{1.8}$$

of the unreduced model (1.1). 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 (1.28) 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 $l \le n$, and let*

$$F_n := V(\boldsymbol{A}_n - s_0\boldsymbol{E}_n)^{-1}V^T\boldsymbol{E}. \tag{2.16}$$

Then for $j \le l \le n$, the j-th moment $\widehat{\mathcal{H}}^{(j)}$ of the explicitly projected ROM transfer function (1.28) and moment $\mathcal{H}^{(j)}$ of the unreduced model (1.4) are related by

$$\widehat{\mathcal{H}}^{(j)} = \boldsymbol{C}^T F_n^i \mathbf{R} \quad \textit{for} \quad i = 0, 1, \ldots \tag{2.17}$$

$$= \mathcal{H}^{(i)} \qquad \textit{for} \quad i = 0, 1, \ldots, j-1. \tag{2.18}$$

Proof. First we show (2.17). Since 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

$$\mathbf{H}^{i-1}\mathbf{R} = VX_i. \tag{2.19}$$

Recall that $\mathbf{R} = (s_0\boldsymbol{E} - \boldsymbol{A})^{-1}\boldsymbol{B}$. Then for $i = 1$,

$$\boldsymbol{B} = (s_0\boldsymbol{E} - \boldsymbol{A})\mathbf{R} = (s_0\boldsymbol{E}V - \boldsymbol{A}V)X_1,$$

which when left-multiplied by V^T results in

$$\begin{aligned} V^T\boldsymbol{B} &= V^T(s_0\boldsymbol{E}V - \boldsymbol{A}V)X_1 \\ \boldsymbol{B}_n &= (s_0\boldsymbol{E}_n - \boldsymbol{A}_n)X_1. \end{aligned}$$

Then

$$X_1 = (s_0\boldsymbol{E}_n - \boldsymbol{A}_n)^{-1}\boldsymbol{B}_n = \widehat{\boldsymbol{\rho}}_n. \tag{2.20}$$

Right-multiplying (2.16) with V gives $F_nV = V(\boldsymbol{A}_n - s_0\boldsymbol{E}_n)^{-1}\boldsymbol{E}_n = V\widehat{\mathbf{H}}$, and by induction on i,

$$F_n^iV = V\widehat{\mathbf{H}}^i \quad \text{for} \quad i = 0, 1, \ldots. \tag{2.21}$$

Then moments of the ROM transfer function

$$\begin{aligned} \widehat{\mathcal{H}}^{(i)} = \boldsymbol{C}_n^T\widehat{\mathbf{H}}^i\widehat{\boldsymbol{\rho}}_n &= \boldsymbol{C}^TV\widehat{\mathbf{H}}^i\widehat{\boldsymbol{\rho}}_n \\ &= \boldsymbol{C}^T(F_n^iV)X_1 \quad \text{by (2.20) and (2.21)} \\ &= \boldsymbol{C}^TF_n^i\mathbf{R} \qquad \text{by (2.19) with } i = 1, \end{aligned}$$

which is (2.17).

Proof of (2.18) is implied by

$$\mathbf{H}^j\mathbf{R} = F_n^j\mathbf{R} \quad \text{for} \quad i = 0, 1, \ldots, j-1 \tag{2.22}$$

which we show by induction on i. (2.22) is trivial for $i = 0$. Now assume (2.22) 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:

$$\big((\boldsymbol{A}-s_0\boldsymbol{E})^{-1}\boldsymbol{E}\big)(F_n^i\mathbf{R}) = \mathbf{H}(\mathbf{H}^i\mathbf{R}) = \mathbf{H}^{i+1}\mathbf{R} = VX_{i+2} \tag{2.23}$$

where the rightmost expression follows from (2.19). Left-multiplying (2.23) with $V^T(\boldsymbol{A}-s_0\boldsymbol{E})$ yields

$$(V^T\boldsymbol{E})(F_n^i\mathbf{R}) = (V^T(\boldsymbol{A}-s_0\boldsymbol{E})V)X_{i+2} = (\boldsymbol{A}_n - s_0\boldsymbol{E}_n)X_{i+2}. \tag{2.24}$$

Then

$$\begin{aligned}
F_n^{i+1}\mathbf{R} &= F_n(F_n^i\mathbf{R}) \\
&= \Big(V(\boldsymbol{A}_n - s_0\boldsymbol{E}_n)^{-1}V^T\boldsymbol{E}\Big)(F_n^i\mathbf{R}) \\
&= V(\boldsymbol{A}_n - s_0\boldsymbol{E}_n)^{-1}(V^T\boldsymbol{E})(F_n^i\mathbf{R}) \\
&= VX_{i+2}, \qquad \text{by (2.24)} \\
&= \mathbf{H}^{i+1}\mathbf{R}, \qquad \text{by (2.19)}
\end{aligned}$$

which proves (2.22). Applying $\boldsymbol{C}^T$ yields (2.18), i.e.

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

□

2.1.4 Complex expansion-points

If a shift $s_0^j \in \mathbb{C}$ is not strictly-real then the basis $V_j \in \mathbb{C}^{N\times n_j}$ of its associated Krylov-subspaceis also complex. Grimme discusses rational-Krylov interpolation-point selection for MOR in depth, in [14].

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. However, it should be noted that using $s_0 \in \mathbb{R}$ is only half as efficient as it appears to be and $s_0 \in \mathbb{C}$ with $\Re(s_0) \neq 0$ 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 $s_0 \in \mathbb{R}$, eigenvalues of $\mathbf{H} = (\boldsymbol{A} - s_0\boldsymbol{E})^{-1}E$ must converge pairwise, so each conjugate-pair of converged vectors provide only one piece of complex spectral information. Eigenvalues of $\mathbf{H}$ for complex s_0 do not converge in pairs, but each converged eigenvalue λ implies that the pole $\mu = s_0 + 1/\lambda$ *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 real and complex interpolation-points s_0 as there seems.

Real bases are preferred because the system (1.1) 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 then also real and thus retains 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-subspaceis 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-space*[5]

$$\begin{aligned}
&\operatorname{span}\begin{bmatrix} v_1^{\mathbf{r}} & v_1^{\mathbf{i}} & v_2^{\mathbf{r}} & v_2^{\mathbf{i}} & \cdots & v_n^{\mathbf{r}} & v_n^{\mathbf{i}} \end{bmatrix} \\
&\quad = \operatorname{span} \mathcal{K}_\eta(\mathbf{H}, \mathbf{R}) \cup \mathcal{K}_\eta(\overline{\mathbf{H}}, \overline{\mathbf{R}}) \\
&\quad = \operatorname{span} \mathcal{K}_\eta(\mathbf{H}(s_0), \mathbf{R}(s_0)) \cup \mathcal{K}_\eta(\mathbf{H}(\overline{s_0}), \mathbf{R}(\overline{s_0})) \\
&\quad = \mathcal{K}_\eta(\mathbf{H}, \mathbf{R})^*
\end{aligned} \tag{2.25}$$

of dimension $\eta \leq 2n$. The basis for a standard Krylov-subspacemay have complex vectors but its span is generally considered over $\mathbb{R}$. The split complex Krylov-subspaceadmits 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(2.25) is the union of two Krylov subspaces with complex conjugate shifts s_0 and $\overline{s_0}$ so we may consider a complex shift to be two shifts. Saad calls this idea "double-shifting" in [25], where it was first given significant analysis. Matching moments about a conjugate pair of points s_0 and $\overline{s_0}$ 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 projection basis V.

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

2.1.5 Producing a real basis for a complex Krylov-subspace

If we use a shift s_0 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 (2.25). One way to do that is to perform a run of the Arnoldi process (algorithm 1) in complex arithmetic with matrices $\mathbf{H}(s_0)$ and $\mathbf{R}(s_0)$, 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

$$\begin{bmatrix} v_1^{\mathbf{r}} & v_1^{\mathbf{i}} & v_2^{\mathbf{r}} & v_1^{\mathbf{i}} & \cdots & v_n^{\mathbf{r}} & v_n^{\mathbf{i}} \end{bmatrix}. \tag{2.26}$$

But the set (2.26) is no longer orthogonal, and possibly linearly dependent. Orthogonalization of (2.26) requires up to $(2n)^2 N$ flops.

Ruhe's method

It seems that it would be ideal to create an orthogonal, real basis for (2.25) directly during the iterative process. Ruhe addresses this in [28], 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. The problem with this method just described is that it is not clear what vector we should iterate with next in order to build a basis for (2.25). 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 (2.25), 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 somewhat redundant.

2.1.6 Orthgognalization of a complex Krylov-subspace basis using a real inner-product

Ruhe's method for generating a real basis yields unpredictable results, compared with working strictly in complex arithmetic followed by splitting the complex basis into $\Re$ and $\Im$ parts and reorthogonalizing as a post processing step. One way to cut complex-vector orthogonalization costs in half for a Gram-Schmidt based based process during the the Krylov process is to use the alternate real-valued inner-product

$$\langle a + ib, x + iy \rangle_{\mathbb{R}} = x^T a + y^T b \in \mathbb{R}, \tag{2.27}$$

instead of the complex Euclidean inner-product

$$(x + iy)^H (a + ib) = x^T a + y^T b + i(x^T b - y^T a) \in \mathbb{C}. \tag{2.28}$$

for two complex vectors $a + ib$ and $x + iy$. Note that $\langle u, v \rangle = \Re(u^H v)$. The inner-product (2.27) is decoupling in the sense that $\Re$ and $\Im$ parts of the involved vectors remain separate during orthogonalization, almost like two separate sets of vectors. (2.27) is cheaper to compute than (2.28), and vector scaling with the real-valued (2.27) is cheaper as well. The basis V produced by

an l-iteration run 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 (1.3), and it may not even span $\mathcal{K}_l(\mathbf{H}, \mathbf{R})$. However, it satisfies (2.25); that is,

$$\operatorname{span} V^* = \mathcal{K}_l(\mathbf{H}, \mathbf{R})^* = \operatorname{span} \mathcal{K}_l(\mathbf{H}(s_0), \mathbf{R}(s_0)) \cup \mathcal{K}_l(\mathbf{H}(\overline{s_0}), \mathbf{R}(\overline{s_0})),$$

which works out because we must split and re-orthogonalize anyway. We call a set *equivalent-real* orthogonal if it is orthogonal with respect to (2.27), where the notion of "equivalent-realness" is explained next.

Constructing such an equivalent-real orthogonal basis for $\mathcal{K}_l(\mathbf{H}, \mathbf{R})$ 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 (2.27).

However, the matrix

$$\boldsymbol{G} = \left[g_{jk} \right] \neq V^H \mathbf{H} V \tag{2.29}$$

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

Equivalent real formulations

A Krylov process that orthogonalizes iterates with respect to (2.27) is effective for constructing a split-worthy basis because it is the Euclidean inner-product on the Krylov-subspace$\mathcal{K}_\eta(\ddot{\mathbf{H}}, \ddot{\mathbf{R}}) \subset \mathbb{R}^{2N}$ induced by the *equivalent-real*, or *realfied* formulations

$$\ddot{\mathbf{H}} = \begin{bmatrix} \mathbf{H^r} & -\mathbf{H^i} \\ \mathbf{H^i} & \mathbf{H^r} \end{bmatrix} \quad \text{and} \quad \ddot{\mathbf{R}} = \begin{bmatrix} \mathbf{R^r} \\ \mathbf{R^i} \end{bmatrix}. \tag{2.30}$$

of $\mathbf{H} = \mathbf{H^r} + i\mathbf{H^i}$ and $\mathbf{R} = \mathbf{R^r} + i\mathbf{R^i}$ from (1.8). This idea is from [25, Sec. 5] and [5, 'K1-formulation']. There is a general definition and discussion of realified spaces as a pure-mathematics topic in [23].

A basis

$$\ddot{V} = \begin{bmatrix} \ddot{v}_1 & \ddot{v}_2 & \cdots & \ddot{v}_n \end{bmatrix} \tag{2.31}$$

for the Krylov-subspace $\mathcal{K}_n(\ddot{\mathbf{H}}, \ddot{\mathbf{R}})$ induced by the realified matrices (2.30) 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 (2.31) as

$$\ddot{V}_n^* := \begin{bmatrix} \ddot{v}_1^{\mathbf{t}} & \ddot{v}_1^{\mathbf{b}} & \ddot{v}_2^{\mathbf{t}} & \ddot{v}_2^{\mathbf{b}} & \cdots & \ddot{v}_n^{\mathbf{t}} & \ddot{v}_n^{\mathbf{b}} \end{bmatrix}, \tag{2.32}$$

which is analogous to the split (2.25) 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.

We remind the reader that equivalent-real forms never need to be explicitly formed. They are only implied by the use of the inner-product (2.27).

Equivalence of split spaces obtained via complex and equivalent-real formulation

Lemma 1. *Consider the equivalent-real formulations* $\ddot{\mathbf{H}}$ *and* $\ddot{\mathbf{R}}$ *of* $\mathbf{H}$ *and* $\mathbf{R}$ *as defined by* (2.30). *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.*

$$(\mathbf{H}^j\mathbf{R})^* = \ddot{\mathbf{H}}^j\ddot{\mathbf{R}}. \tag{2.33}$$

Equivalently,

$$\ddot{\mathbf{H}}^j\ddot{\mathbf{R}} = \begin{bmatrix} \Re(\mathbf{H}^j\mathbf{R}) \\ \Im(\mathbf{H}^j\mathbf{R}) \end{bmatrix}. \tag{2.33*}$$

Proof. Trivially for $j = 0$ we have $\ddot{\mathbf{R}} := \begin{bmatrix} \mathbf{R}^{\mathbf{r}} & \mathbf{R}^{\mathbf{i}} \end{bmatrix}^T$. 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

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

is the equivalent real formulation of

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

□

It follows as a corollary that the split-Krylov subspaces (2.25) and (2.32) induced by each pair, are equal.

$$\mathcal{K}_n(\ddot{\mathbf{H}}, \ddot{\mathbf{R}})^* = \mathcal{K}_n(\mathbf{H}, \mathbf{R})^* \tag{2.34}$$

The inner-products (2.28) and (2.27) yield different notions of orthogonality of a complex basis and its equivalent-real counterpart, and ultimately incompatible spaces. The inner-product (2.27) implies a weaker orthogonality than (2.28): 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 (2.34) of Lemma 2.33 is promising.

The norms implied by (2.28) and (2.27) for a complex vector $v \in \mathbb{C}^N$ and its equivalent real form $\ddot{v} \in \mathbb{R}^{2N}$ are equal:

$$\|\ddot{v}\|_2^2 = \ddot{v}^T \ddot{v} = v^H v = \langle v, v \rangle = \|v\|_2^2 . \tag{2.35}$$

Lemma 2.33 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(2.25) in the same order, regardless of whether we use complex or equivalent-real formulation (i.e. complex formulation with inner-product (2.27)).

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* (1.8) *and their realified formulations* $\ddot{\mathbf{H}}$ *and* $\ddot{\mathbf{r}}$ *defined by* (2.30). *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*

$$\Re(v_n) = \alpha \ddot{v}_n^{\mathbf{t}} + w \qquad \textit{and} \qquad \Im(v_n) = \beta \ddot{v}_n^{\mathbf{b}} + z. \tag{2.36}$$

Proof. We will prove (2.36) 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 (2.35), so $\Re(v_1) = \ddot{v}_1^{\mathbf{t}}$ and $\Im(v_1) = \ddot{v}_1^{\mathbf{b}}$, trivially satisfying (2.36).

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

$$\mathcal{K}_n(\mathbf{H}, \mathbf{r}) \subseteq \mathcal{K}_n(\mathbf{H}, \mathbf{r})^* = \operatorname{span} \begin{bmatrix} \widetilde{v}_1 & \widetilde{v}_2 & \cdots & \widetilde{v}_\eta \end{bmatrix} \tag{2.37}$$

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

$$\begin{aligned} h_{n+1,n} v_{n+1} &= \mathbf{H} v_n - \sum_{j=1}^{n} h_{jn} v_j \\ &= \mathbf{H}^n \mathbf{r} + \sum_{j=1}^{n} c_j v_j, \qquad c_j \in \mathbb{R} \\ &= \mathbf{H}^n \mathbf{r} + \sum_{j=1}^{\eta} d_j \widetilde{v}_j, \qquad d_j \in \mathbb{C}, \quad \text{by (2.37)} \\ &= \mathbf{H}^n \mathbf{r} + w_1 + i z_1, \qquad w_1, z_1 \in \mathcal{K}_n(\mathbf{H}, \mathbf{r})^* \cap \mathbb{R}^N. \end{aligned} \tag{2.38}$$

Lemma 1 implies that we can re-write (2.38) in realified form as

$$h_{n+1,n} \begin{bmatrix} \Re(v_{n+1}) \\ \Im(v_{n+1}) \end{bmatrix} = \ddot{\mathbf{H}}^n \ddot{\mathbf{r}} + \begin{bmatrix} w_1 \\ z_1 \end{bmatrix}. \tag{2.39}$$

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.

Reduced-order models via equivalent-real formulations

The explicitly projected ROM (1.2) using a basis (2.25) for the split-Krylov-subspace $\mathcal{K}_\eta(\mathbf{H}, \mathbf{R})^*$ is equivalent regardless of the orthogonalization method used to construct it. The matrix $\boldsymbol{G}$ of orthogonalization coefficients from the equivalent-real Arnoldi process, (2.29), is a Rayleigh-quotient approximant 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 (2.30), 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 (2.30). Thus, an implicitly projected ROM (1.29) is not so simple to characterize. For example, it is known that the spectrum

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

of $\ddot{\mathbf{H}}$ contains spectral information for $\overline{\mathbf{H}}$ which complicates matters because since 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 3

3.1 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 [11] (and [13]) for model order-reduction was done by Grimme in the mid and late 1990s. It is somewhat based on Ruhe's Rational-Krylov [27, 28] eigenvalue method and formalization, and possibly Olsson's [22]. Of particular interest are [14] and [6], both of which discuss interpolation-point selection. We refer the reader to those sources for the details of point selection.

[19, 20, 18, 8]) are more recent multi-point rational-interpolation methods. Also a Jacobi-Davidson MOR method [3]. Lee, Chu, and Feng's RAMAO/AORA method (Rational Arnoldi Method with Adaptive Order selection/ Adaptive-Order Rational-Arnoldi) [19, 20] breaths new life into an adaptive point-selection method introduced by [11], based on the sequence of ROM *moment-errors* implied by the sequence of candidate-vectors $\hat{v}_k$ of the Arnoldi process §2.1.1.

In rational-Krylov method literature, the shifts/interpolation-points are usually denoted by σ and we will adopt that convention. Suppose that for each $j = 1, 2, \ldots, \tau$ the Krylov-subspace

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

of dimension n_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

$$\mathcal{K}_1 \cup \mathcal{K}_2 \cup \cdots \cup \mathcal{K}_\tau \subseteq \text{span}\, V \tag{3.1}$$

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

Merging bases

There are several ways to produce a basis for the composite space (3.1). 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

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

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

$$\operatorname{span} \begin{bmatrix} v_1 & v_2 & \cdots & v_n \end{bmatrix} = \operatorname{span} V_1 \cup \operatorname{span} V_2 \cup \cdots \cup \operatorname{span} V_\tau. \tag{3.2}$$

For the general application of rational-interpolation we assume that complex interpolation-points are used, so as discussed in §2.1.4 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 $\mathcal{K}_{l_j}(\mathbf{H}_j, \mathbf{R}_j)$ is also $(\boldsymbol{A}, \boldsymbol{E})$-invariant and thus has global significance.

The naive approach of producing bases for $\mathcal{K}_{n_j}(\mathbf{H}(\sigma_j), \mathbf{R}(\sigma_j))$ 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 $\mathcal{K}_{l_j}(\mathbf{H}_j, \mathbf{R}_j)$, if we already discovered it while constructing the basis V_{j-1} for $\mathcal{K}_{l_{j-1}}(\mathbf{H}_{j-1}, \mathbf{R}_{j-1})$. Traditional rational-interpolation methods for MOR such as rational-Lanczos [11] avoid this issue by doing full, Arnoldi-style orthogonalization of an iterate against every previous vector, generating one orthogonal basis V for (3.2) directly. Considering that for complex-valued interpolation-points we will have to split the basis (3.2) anyway, thus losing any orthogonality, doing full orthogonalization of every vector is overkill.

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

$$\{\mathbf{H}(\sigma) = (\boldsymbol{A} - \sigma \boldsymbol{E})^{-1} \boldsymbol{E}\}. \tag{3.3}$$

Then

$$\mathbf{H}_2 z = T \mathbf{H}_1 z = \lambda T z + T r, \tag{3.4}$$

where T is the transformation

$$\begin{aligned} T(\sigma_1, \sigma_2) &:= (\boldsymbol{A} - \sigma_2 \boldsymbol{E})^{-1} (\boldsymbol{A} - \sigma_1 \boldsymbol{E}) \\ &= (\sigma_2 - \sigma_1) \mathbf{H}_2 + I. \end{aligned} \tag{3.5}$$

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

proof of (3.5). The expression (3.5) 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 (3.5) must be invertible, with

$$
\big(T(\sigma_1, \sigma_2)\big)^{-1} = T(\sigma_2, \sigma_1) = (\sigma_1 - \sigma_2)\mathbf{H}_1 + I. \tag{3.6}
$$

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 (3.5) and (3.6) 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

$$
\mathbf{H}_2\mathbf{H}_1 = \frac{\mathbf{H}_2 - \mathbf{H}_1}{\sigma_2 - \sigma_1} \tag{3.7}
$$

and for $\sigma_2 \neq \sigma_1$,

$$
\mathbf{H}_1\mathbf{H}_2 = \mathbf{H}_2\mathbf{H}_1. \tag{3.8}
$$

(3.8) shows that the set (3.3) of operators, commutes. (3.7) implies that

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

for what it's worth.

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

$$
\mathcal{H}(s) = \boldsymbol{C}^T\big(I - (s - \sigma)\mathbf{H}(\sigma)\big)^{-1}\mathbf{R}(\sigma), \tag{3.9}
$$

defined about the interpolation-point σ but independent of σ, involves a transformation of the form (3.5), since

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

Then we may re-write (3.9) 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 (3.4) that for eigen-pair (λ, z) of $\mathbf{H}_1$ we have

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

with the translation $T = T(\sigma_1, \sigma_2) = \Delta\mathbf{H}_2 + I$ from (3.5) 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

$$(1 - \lambda\Delta)\mathbf{H}_2 z = \lambda z + Tr. \tag{3.10}$$

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

$$\begin{aligned}\mathbf{H}_2 z - (\zeta\lambda)z &= \zeta Tr \\ &= \zeta(\Delta\mathbf{H}_2 + I)r\end{aligned} \tag{3.11}$$

3.2 Thick-Restarted band-Arnoldi method for MOR

"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 an 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-subspace. 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 $\begin{bmatrix} Y_1 & \mathbf{R}_2 \end{bmatrix}$.

The primary contribution of this dissertation to the field of model order-reduction is development and testing of a thick-restarted Arnoldi-type algorithm for multi-point MOR, where the model to be reduced is assumed to be MIMO. We show that we can obtain smaller models of the same or better accuracy, at less computational cost, using this process.

3.2.1 Band-Arnoldi algorithm

First we address the band-Arnoldi algorithm and the band-Arnoldi relation, which are the basic engine for the MOR method. The *band*-Arnoldi process of [10] (2003) is included here as algorithm 2.

Band-Arnoldi differs from a block-Krylov process (like block-Arnoldi [21]) in that it 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

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

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 a 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.

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 $\mathcal{K}_n(\mathbf{H},\mathbf{R})$, deflated vectors $\dot{V} = \begin{bmatrix}\dot{v}_1 & \dot{v}_2 & \dots & \dot{v}_d\end{bmatrix}$, remaining candidate-vectors $\hat{V} = \begin{bmatrix}\hat{v}_{n+1} & \hat{v}_{n+2} & \dots & \hat{v}_{n+m_c}\end{bmatrix}$, and Rayleigh-quotient $\widetilde{\mathbf{H}} = V^H\mathbf{H}V$ that satisfy the band-Arnoldi relation

$$\mathbf{H}V = V\widetilde{\mathbf{H}} + \begin{bmatrix}(I - VV^H)\dot{V} & \hat{V}\end{bmatrix} F \tag{3.1}$$

where $F = \begin{bmatrix}F_1\\F_2\end{bmatrix}$. $\widetilde{\mathbf{H}}$ is block-upper-Hessenberg (strictly upper-Hessenberg in the single vector setting $m = 1$) with possibly non-zero columns in the typically zero (lower co-Hessenberg?) region corresponding to the deflated vectors $\dot{V}$. 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 (3.1).

In addition, algorithm 2 returns $\widetilde{\boldsymbol{\rho}}$, and $\widetilde{\boldsymbol{\rho}}^{\text{defl}}$ where

$$\mathbf{R} = V\widetilde{\boldsymbol{\rho}} + \widetilde{\boldsymbol{\rho}}^{\text{defl}}, \tag{3.2}$$

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

Candidates/residual term

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

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 $V^H\hat{V} = 0$.

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 $e_2, e_5 \in \mathbb{R}^{10}$,

$$\begin{aligned}\dot{V}F_1 = \hat{V}^{\text{defl}} &= \begin{bmatrix}0 & \hat{v}_2^{\text{defl}} & 0 & 0 & \hat{v}_5^{\text{defl}} & 0 & 0 & 0 & 0 & 0\end{bmatrix}\\ &= \begin{bmatrix}0 & \dot{v}_1 & 0 & 0 & \dot{v}_2 & 0 & 0 & 0 & 0 & 0\end{bmatrix}\\ &= \begin{bmatrix}\dot{v}_1 & \dot{v}_2\end{bmatrix}\begin{bmatrix}e_2^T\\e_5^T\end{bmatrix}.\end{aligned} \tag{3.3}$$

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 se ts $\hat{v}_j^{\text{defl}} := \hat{v}_n$ 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

$$V^H \hat{v}_j^{\text{defl}} = \begin{bmatrix} 0 & 0 & \cdots & 0 & v_{j+1}^H \hat{v}_j^{\text{defl}} & v_{j+2}^H \hat{v}_j^{\text{defl}} & \cdots & \hat{v}_j^H \hat{v}_j^{\text{defl}} \end{bmatrix}^T. \tag{3.4}$$

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

If no/exact deflation was performed, $\widetilde{\mathbf{H}}$ is strictly block-upper-Hessenberg, otherwise $\widetilde{\mathbf{H}}$ may have non-zero entries in the triangular region, $\widetilde{\mathbf{H}}_{\mathcal{E}} = V^H \dot{V}$, below the 1st subdiagonal of $\widetilde{\mathbf{H}}$. If an inexact deflation occurred on the j-th iteration, (3.4) is included in the Rayleigh-quotient $\widetilde{\mathbf{H}}$ as the j-th column of $\widetilde{\mathbf{H}}_{\mathcal{E}}$. Then

$$\|\widetilde{\mathbf{H}}_{\mathcal{E}}\| = \|V^H \dot{V}\|_F \leq \|\dot{V}\|_F \leq \text{dtol}\sqrt{d}, \tag{3.5}$$

and

$$\|(I - VV^H)\dot{V}\|_F \leq \|\dot{V}\|_F. \tag{3.6}$$

$\widetilde{\boldsymbol{\rho}}^{\text{defl}}$ in (3.2) 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 $\mathbf{R}$.

Residual norms

A similarity decomposition $\widetilde{\mathbf{H}}S = SU$ such as a Schur or eigenvalue-decomposition of the Rayleigh-quotient together with (3.1) and setting $Y = VS$ gives the block residual-norm bound

$$\begin{aligned} \|\mathbf{H}Y - YU\|_F^2 &= \left\| \left((I - VV^H)\dot{V}F_1 + \hat{V}F_2 \right) S \right\|_F^2 \\ &\leq \|\dot{V}\|_F^2 + \|\hat{V}\|_F^2 \\ &\leq (\text{dtol})^2 d + \|\hat{V}\|_F^2 \\ &= d\,\varepsilon_M + \|\hat{V}\|_F^2, \qquad \text{for} \quad \text{dtol} = \sqrt{\varepsilon_M} \end{aligned}$$

in the Frobenius-norm (entry-wise 2-norm).

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

$$\begin{aligned} \mathbf{H}z_j - z_j\lambda_j &= \begin{bmatrix} (I - VV^H)\dot{V} & \hat{V} \end{bmatrix} F w_j \\ &= (I - VV^H)\hat{V}^{\text{defl}} w_j + \hat{V}F_2 w_j \end{aligned}$$

For determining convergence of Ritz-pairs,

$$\begin{aligned} \|\mathbf{H}z_j - z_j\lambda_j\|_2^2 &\leq d\,\varepsilon_M + \|\hat{V}F_2 w_j\|_2^2 \\ &= d\,\varepsilon_M + \|\hat{V}\tilde{w}_j\|_2^2. \end{aligned} \tag{3.7}$$

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

[1] [26] suggests $\text{dtol} = \sqrt{\epsilon}$, where machine-$\epsilon = 2^{-52} \approx 2.22\text{e-}16$ in double-precision (64-bit) floating point.

(3.7) suggests a few different ways to cheaply estimate the relative residual norm

$$\frac{\|\mathbf{H}z_j - z_j\lambda_j\|}{\|\lambda_j z_j\|} \tag{3.8}$$

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 $\|\mathbf{H}\|$.

Assuming $d\,\varepsilon_M$ is negligible,

$$\left\|\hat{V}\right\| \frac{\|\tilde{w}_j\|}{|\lambda_j|} \tag{3.9}$$

is an estimate of relative residual norm, as is

$$\frac{1}{|\lambda|}\begin{bmatrix}\|\hat{v}_1\| & \|\hat{v}_2\| & \cdots & \|\hat{v}_{m_c}\|\end{bmatrix}\begin{bmatrix}|\tilde{w}_j^{(1)}| \\ |\tilde{w}_j^{(2)}| \\ \vdots \\ |\tilde{w}_j^{(m_c)}|\end{bmatrix}. \tag{3.10}$$

Both (3.9) and (3.10) are cheaper to compute than norms $\|\hat{V}\tilde{w}_j\|$ of potentially large matrix-vector products, but (3.10) might be better if $\hat{V}F_2$ has rank greater than one. Estimates (3.9) and (3.10) are equal for rank-1 residuals.

Algorithm 2: BAND-ARNOLDI

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

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 (3.1)

$\hat{v}_i := \mathbf{r}_i \quad$ 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, $\widetilde{\mathbf{H}}$, $\widetilde{\boldsymbol{\rho}} = \left[h_{ij}\right]_{i=1,2,\ldots,m}^{j=1-m\ldots,1,0}$, $\hat{V}$, $\hat{V}^{defl}$, $\widetilde{\boldsymbol{\rho}}^{defl} = \left[\hat{v}_j^{defl}\right], j = 1 - m, ..., 1, 0,$

3.2.2 Implicitly-projected ROM from a thick-restarted band-Arnoldi process

Recall that the band-Arnoldi algorthm

$$\begin{bmatrix} V & \widetilde{\mathbf{H}} & \widetilde{\boldsymbol{\rho}} \end{bmatrix} = \mathbf{bArnoldi}(\mathbf{H}, \mathbf{R}) \tag{3.11}$$

where $\widetilde{\mathbf{H}} = V^H \mathbf{H} V$ and $\widetilde{\boldsymbol{\rho}} = V^H \mathbf{R}$, implies the implicitly projected ROM approximation

$$\widetilde{\mathcal{H}}(s) = (\boldsymbol{C}^T V)\big(I - (s-\sigma)\widetilde{\mathbf{H}}\big)^{-1} \underbrace{(V^H \mathbf{R})}_{\widetilde{\boldsymbol{\rho}}} \tag{3.12}$$

to the 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 bArnoldi process with a basis Y for an *exactly* $\mathbf{H}$-invariant subspace, so that $\mathbf{H}Y = YU$.

Then the thick-started process

$$\begin{bmatrix} V & \widetilde{\mathbf{H}} & \widetilde{\boldsymbol{\rho}} \end{bmatrix} = \mathbf{bArnoldi}(\mathbf{H}, \begin{bmatrix} Y & \mathbf{R} \end{bmatrix}) \tag{3.13}$$

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

$$\widetilde{\mathcal{H}}(s) = \big(\boldsymbol{C}^T \begin{bmatrix} Y & V' \end{bmatrix}\big)\left(I - (s-\sigma)\widetilde{\mathbf{H}}\right)^{-1} \underbrace{\left(\begin{bmatrix} Y & V' \end{bmatrix}^H \mathbf{R}\right)}_{\widetilde{\boldsymbol{\rho}}_2} \tag{3.14}$$

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.

Chapter 4

4.1 Proposing a new model-reduction method

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 $Y_0 = \{\}, V_{\text{ROM}} = \{\}$
2 **for** $j = 1, 2, \ldots$ **do**
3 Set $\ell_j := \dim Y_j, \quad m := \dim \mathbf{R}_j$
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` (3.14).
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 (3.10).
10 Compute pole-weights $\gamma_1, \gamma_2, \ldots, \gamma_{n_j}$ as (1.26) and (1.27).
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 $\widehat{V}$ 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[31], by pre-loading V with 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.

4.2 Results

First we consider our two example 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

$$\frac{\|\mathcal{H} - \widetilde{\mathcal{H}}\|}{\|\mathcal{H}\|} \leq \texttt{tf\_tol} = 0.01, \tag{4.1}$$

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 4.1. 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 ($s_0 \notin \mathbb{R}$) 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 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 $(\boldsymbol{A}, \boldsymbol{E}, \boldsymbol{B}, \boldsymbol{C})$.

4.2.1 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 spikey appearance.

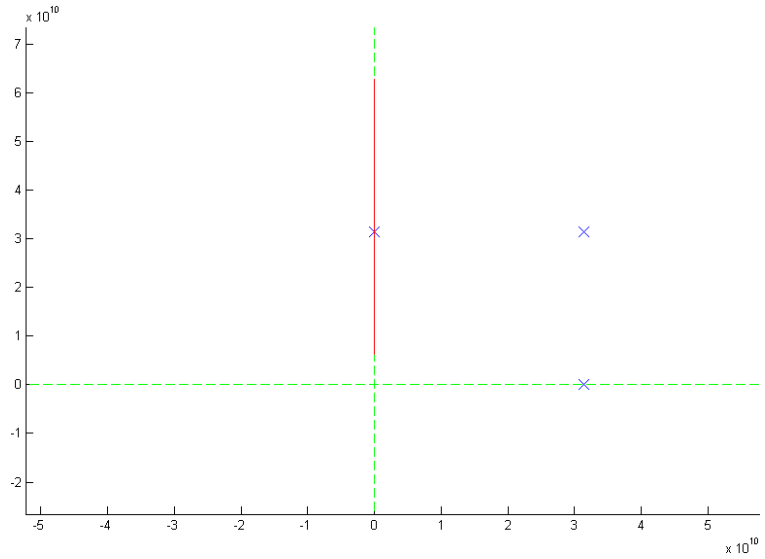


Figure 4.1: The three interpolation-points used for single point benchmarks. The segment of interest $[10^9, 10^{10}]i$ on the $\Im$-axis, is highlighted. Note how small $[0, 10^9]$ is, in comparison.

s_0	iterations (n)	ROM size (n')	LI	rel-err	flops	figure
$\pi 10^{10}$	144	144	1	7.368e-05	16,920,288 + M	4.4
$i\pi 10^{10}$	71	142	0.992958	5.913e-4	30,177,840 + M	4.5
$(1+i)\pi 10^{10}$	97	194	0.793814	5.1713e-3	42,782,432 + M	4.6

Table 4.1: 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.

ex308 Benchmarks

Benchmark data for `ex308` is given in table 4.1.

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.

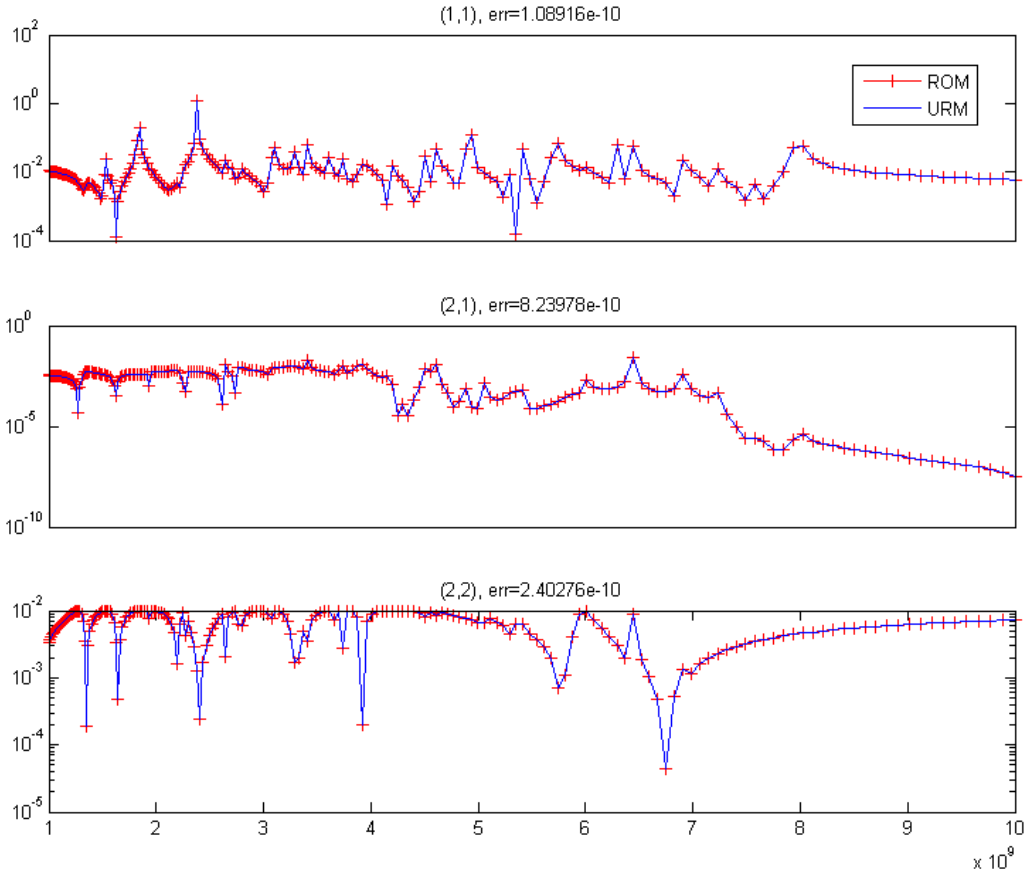


Figure 4.2: These are the three unique gain plots for `ex308`.

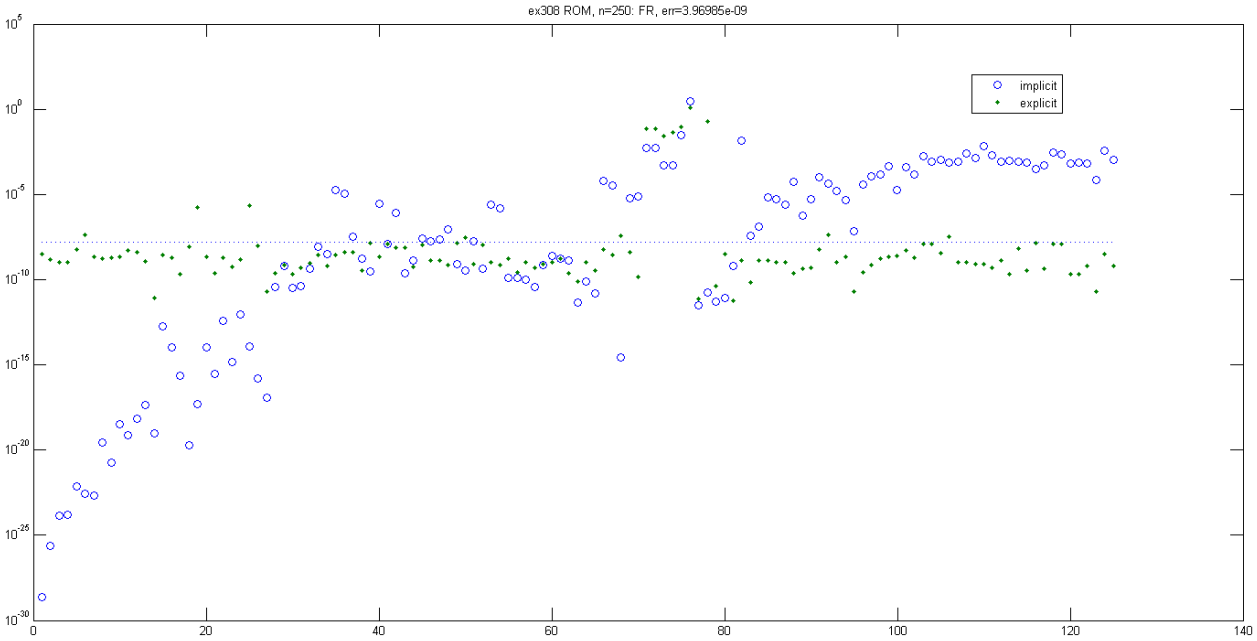


Figure 4.3: This is a plot of relative-residual errors for the 250 poles of an $n = 250$ ROM (about $s_0 = (1+i)\pi 10^{10}$) of `ex308`. The circles are the poles derived from eigenvalues of $\widetilde{\mathbf{H}} = V^H \mathbf{H} V$ (the implicit ROM), and the dots are the eigenvalues of the explicitly 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 (**??**) for the residual norm $\|\boldsymbol{A}z - \mu \boldsymbol{E}z\|$.

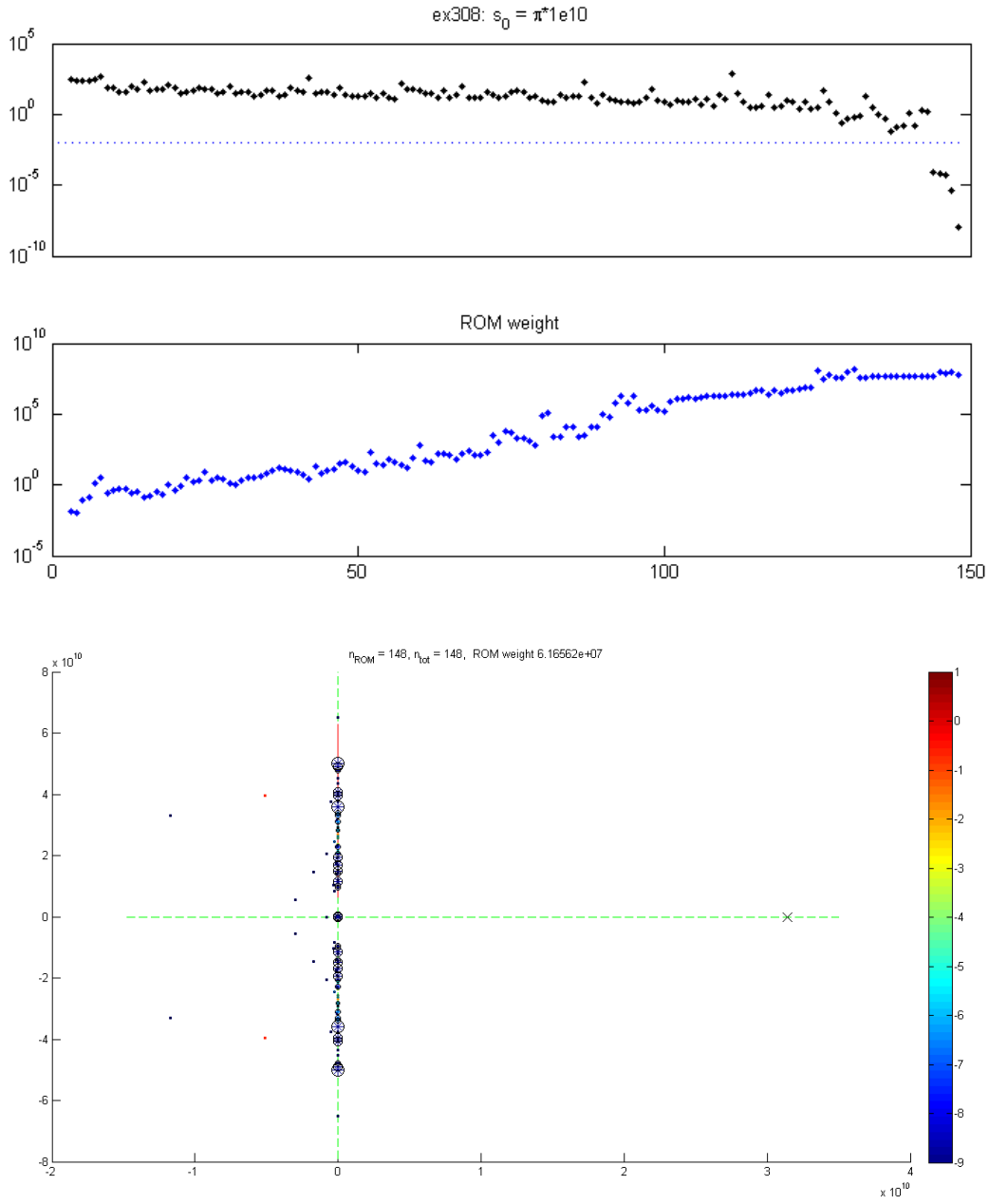


Figure 4.4: Transfer function relative-error (4.1) (of explicitly projected ROM) and ROM weight (of the implicitly projected ROM) vs. n for `ex308`, at real interpolation-point $s_0 = \pi 10^{10} \in \mathbb{R}$, indicated by '$\times$'. The dotted line in the first plot represents a relative-error of 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 $\approx 10^{-8}$.

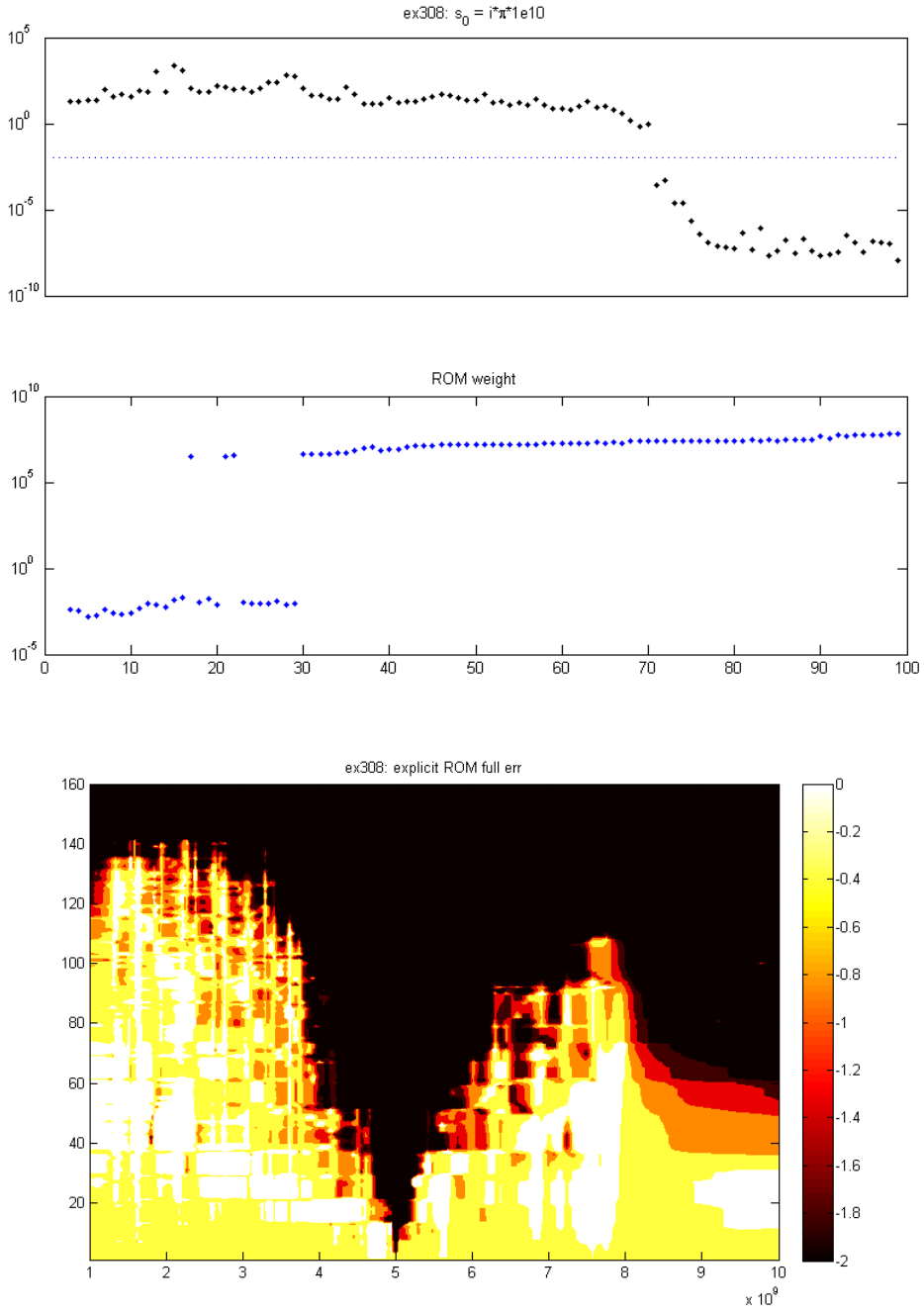


Figure 4.5: Transfer function relative-error and ROM weight vs. n for `ex308`, at $\Im$ interpolation-point $s_0 = i\pi \cdot 10^{10} \in i\mathbb{R}$. Unfortunately it appears that ROM weight is not a consistently reliable indicator of transfer function convergence when using a single interpolation-point. 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.

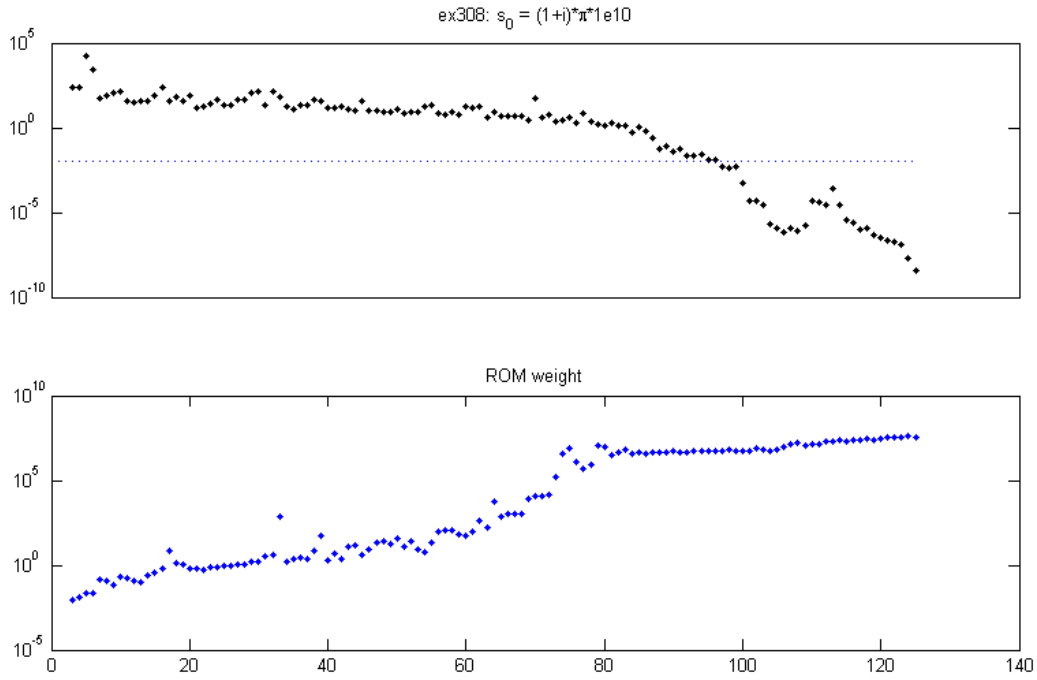


Figure 4.6: Transfer function relative-error and ROM-weight vs. n for `ex308`, at $s_0 = (1+i)\pi \cdot 10^{10}$. This is much the way we would like the relationship between ROM-weight and transfer-function error to look. ROM-weight leveling-off would indicate transfer-function error convergence.

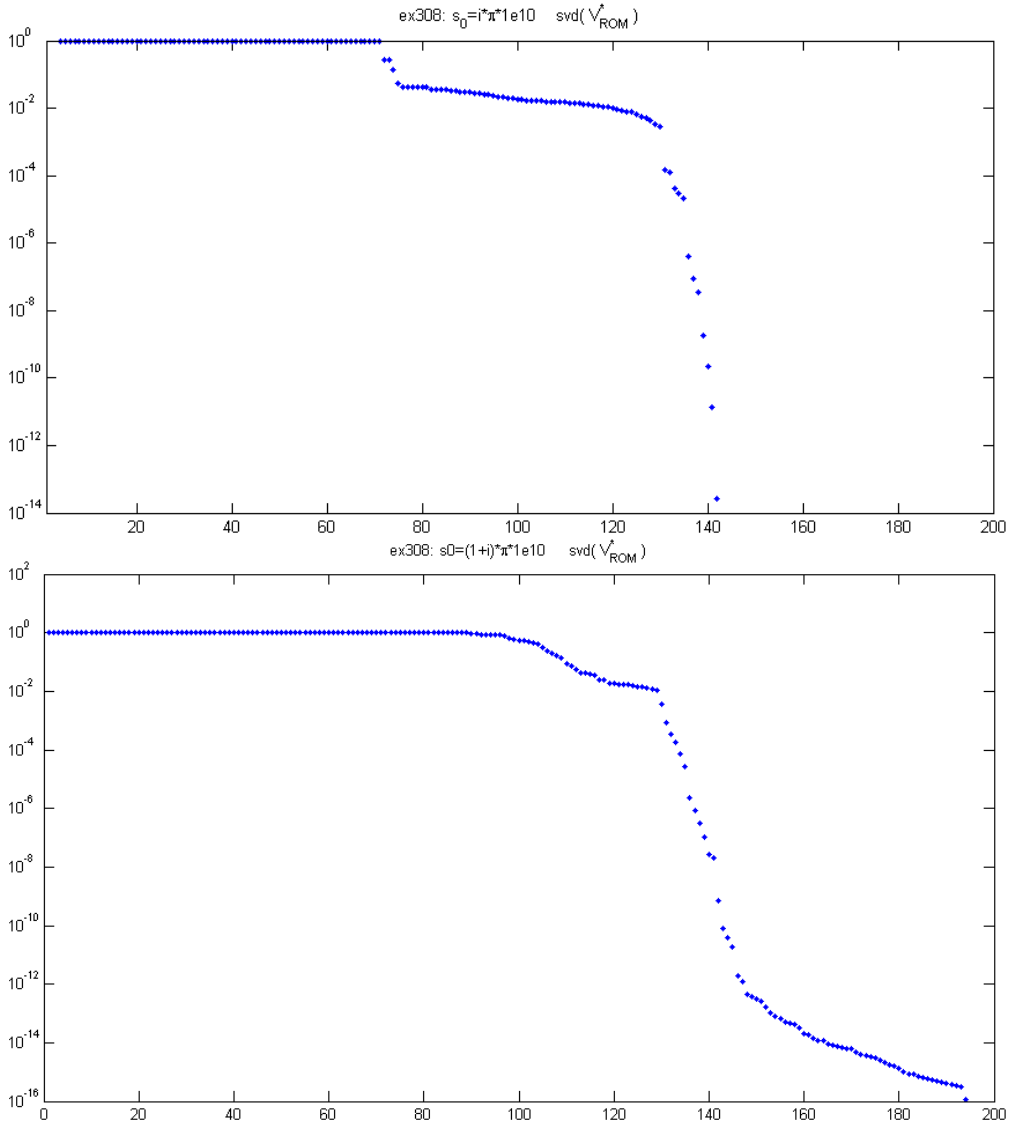


Figure 4.7: These are plots of singular values of the split basis matrix V^* that spans $\Re(V) \cup \Im(V)$ (before being orthogonalized), for bases generated about $s_0 = i\pi \cdot 10^{10}$ and $(1+i)\pi \cdot 10^{10}$. The basis for the ROM expanded about $i\pi \cdot 10^{10}$ has effective rank 141, the same as the model size, and the ROM about $(1+i)\pi \cdot 10^{10}$ has basis with an effective rank of 154. Effective-rank is defined as the number of singular values of the basis that are above the default tolerance used in Matlab. We expect a restarted method to generate a composite basis with lower effective linear-independence in general.

ex308 thick-restart example 1

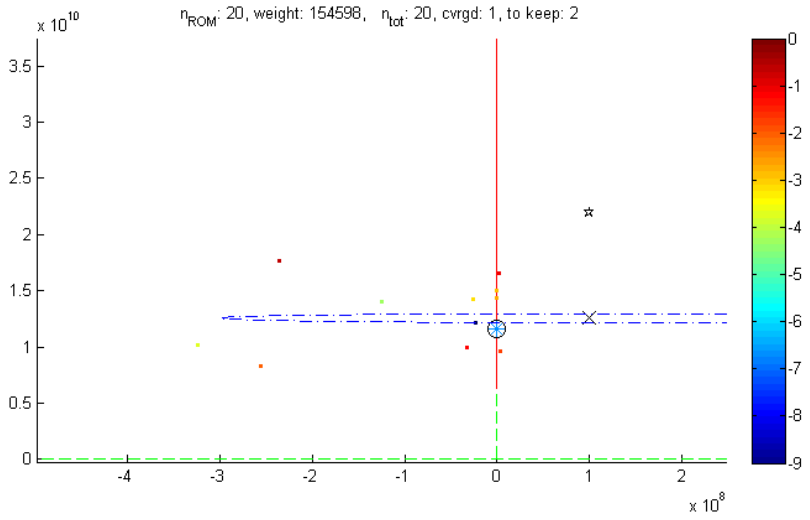
Here we show an example run of the thick-restarted band-Arnoldi process using three pre-set interpolation-points $\sigma_j = 10^8 + 2\pi i \cdot 10^{10} p_j$ for $p_j = 2, 3.5, 6$. When scaled this way, the frequency range of interest $p \in (1, 10)$ corresponds to $s \in i(\omega_1, \omega_2)$ so our 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 `ctol` $= \sqrt{\epsilon} \approx$ `1.49e-8`) and those associated with dominant poles ($\texttt{wt}_i / \sum \texttt{wt}_i \leq 0.05$) were recycled. For this example we just included Ritz-vectors in the start band for the next cycle, and let the non-zero residuals be deflated as normal for the bArnoldi process. A cheaper alternative is to manually set the residuals to zero.

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 `5.40491e-05`, making it compare favorably with the benchmark examples in table 4.1. It required $12,871,320 + 3M$ 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],true)` 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: 154598
...updating thick-restart basis...dim Y = 3
```

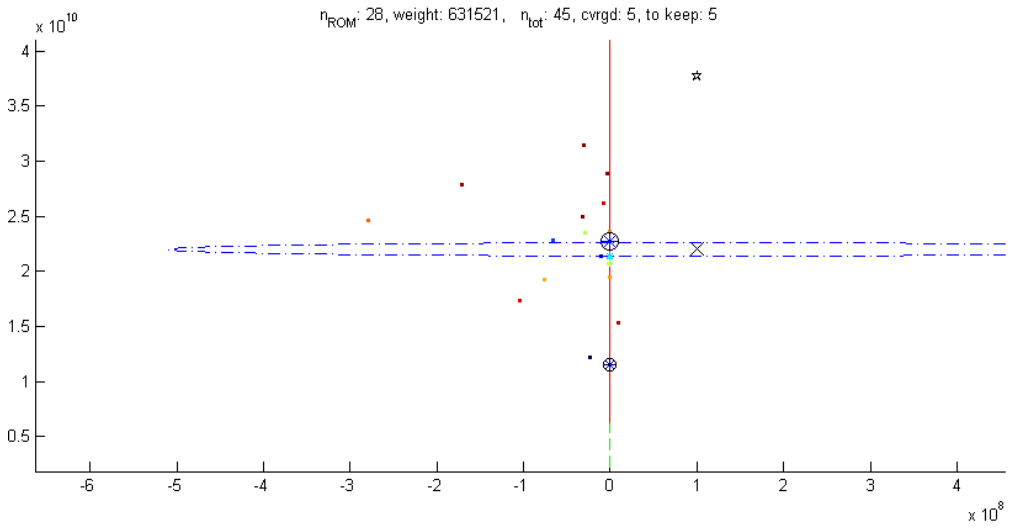


In the above plot, poles of the implicitly projected ROM of the first cycle are indicated by '$*$' symbols. The color of a pole indicates its degree of convergence. The interpolation-point is indicated by '$\times$', and a dashed-circle[1] around the interpolation-point indicates distance to the first converged pole. The '$\star$' 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 + 3
```

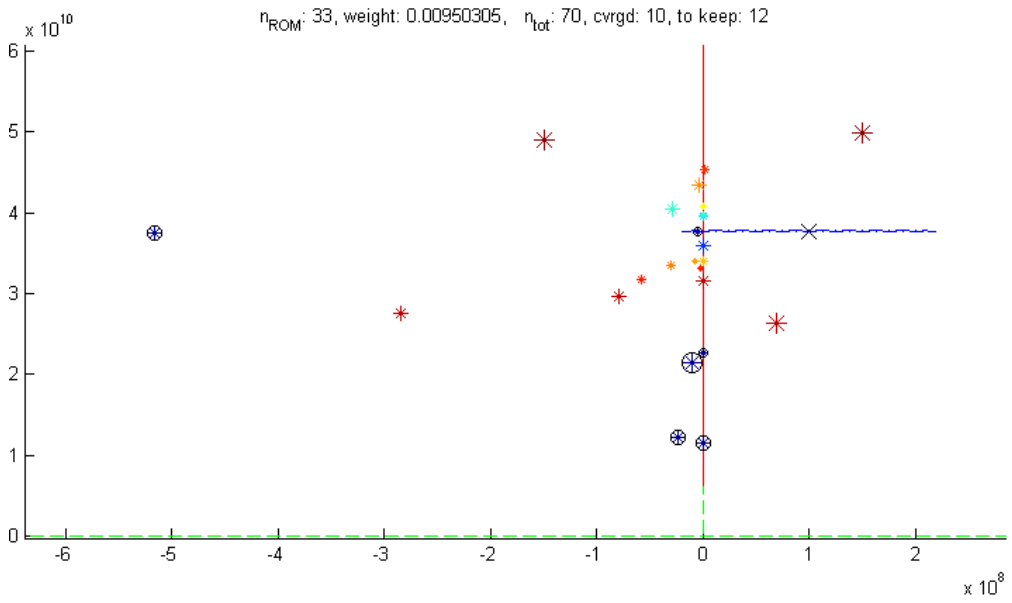
[1] elongated in the plot due to greatly asymmetric scaling. This is one reason why interpolation-points on or near the $\Im$-axis can result in the convergence of un-necessarily large ROMs.

```
v_defl(6)/H_est  = 0 < 1.49012e-08,   mc = 4
v_defl(6)/H_est  = 0 < 1.49012e-08,   mc = 3
v_defl(6)/H_est  = 0 < 1.49012e-08,   mc = 2
... ROM: 28, n_tot: 45,   converged: 5,   keep: 5  weight: 631521
...updating thick-restart basis...dim Y = 8
```




```
cycle 3 expanding at s_0 = 2\pi10^9(0.0159155 + 6i),  band_size = 2 + 8
v_defl(11)/H_est  = 0 < 1.49012e-08,   mc = 9
v_defl(11)/H_est  = 0 < 1.49012e-08,   mc = 8
v_defl(11)/H_est  = 0 < 1.49012e-08,   mc = 7
v_defl(11)/H_est  = 0 < 1.49012e-08,   mc = 6
v_defl(11)/H_est  = 0 < 1.49012e-08,   mc = 5
v_defl(11)/H_est  = 0 < 1.49012e-08,   mc = 4
v_defl(11)/H_est  = 0 < 1.49012e-08,   mc = 3
v_defl(11)/H_est  = 0 < 1.49012e-08,   mc = 2
... ROM: 33, n_tot: 70,   converged: 10,   keep: 12  weight: 0.00950305
...updating thick-restart basis...dim Y = 14

iterations: 70, ROM size: 140, LII: 1, rel-error: 5.40491e-05,  flops: 12871320 + 3M
```


n_ROM: 33, weight: 0.00950305, n_tot: 70, cvrgd: 10, to keep: 12
x 10^10
6
5
4
3
2
1
0
-6
-5
-4
-3
-2
-1
0
1
2
x 10^8

	$\Re(\mu)$	$\Im(\mu)$	rr	wt	keep
1	-2.3746e+07	1.2186e+10i	4.22199e-11	0.0260911	1
2	4.8802e+01	1.1562e+10i	1.9333e-07	154560	1
3	-1.2457e+08	1.3997e+10i	5.79701e-05	0.0308771	0
4	-3.2363e+08	1.0214e+10i	0.000210072	0.0218821	0
5	-1.7281e+09	1.4688e+10i	0.000767965	0.037747	0
6	-2.6226e+07	1.4295e+10i	0.000794019	0.0191108	0
7	-9.2396e+05	1.4978e+10i	0.00144059	14.3762	0
8	1.2915e+05	1.4334e+10i	0.0015452	1.35788	0
9	-2.5567e+08	8.3294e+09i	0.0102311	0.0438477	0
10	4.1068e+06	9.6085e+09i	0.0151834	0.520942	0

(a) Cycle 1

	$\Re(\mu)$	$\Im(\mu)$	rr	wt	keep
1	1.9509e+02	1.1562e+10i	0	52252.2	1
2	-2.3746e+07	1.2186e+10i	0	0.548231	1
3	-2.3746e+07	-1.2186e+10i	0	9.99915	1
4	-1.0463e+07	2.1391e+10i	4.14825e-09	0.227448	1
5	1.1893e-01	2.2714e+10i	8.88829e-09	567512	1
6	-6.6011e+07	2.2864e+10i	4.82273e-08	0.0240876	0
7	-6.2228e+00	2.1332e+10i	4.28048e-07	9512.5	0
8	6.0902e+00	2.1293e+10i	2.89992e-06	725.286	0
9	-8.2289e+08	2.0635e+10i	1.90154e-05	0.0476409	0
10	-1.4824e+02	2.0682e+10i	3.92794e-05	1484.75	0

(b) Cycle 2

	$\Re(\mu)$	$\Im(\mu)$	rr	wt	keep
1	4.5772e+00	2.2714e+10i	0	1.21817e-07	1
2	-1.0463e+07	2.1391e+10i	0	3.2373e-06	1
3	-1.0738e+03	1.1562e+10i	0	1.23019e-06	1
4	-2.3746e+07	1.2186e+10i	0	2.15423e-06	1
5	3.0079e+00	-2.2714e+10i	0	3.03128e-07	1
6	-1.0463e+07	-2.1391e+10i	0	2.26298e-05	1
7	-2.3746e+07	-1.2186e+10i	0	6.07987e-06	1
8	-3.6465e+03	-1.1562e+10i	0	2.2665e-06	1
9	-4.3476e+06	3.7641e+10i	5.41559e-23	8.89941e-08	1
10	-5.1591e+08	3.7457e+10i	3.12184e-14	8.36932e-07	1

(c) Cycle 3

Table 4.2: The 10 Ritz-poles of lowest relative-residual for each cycle.

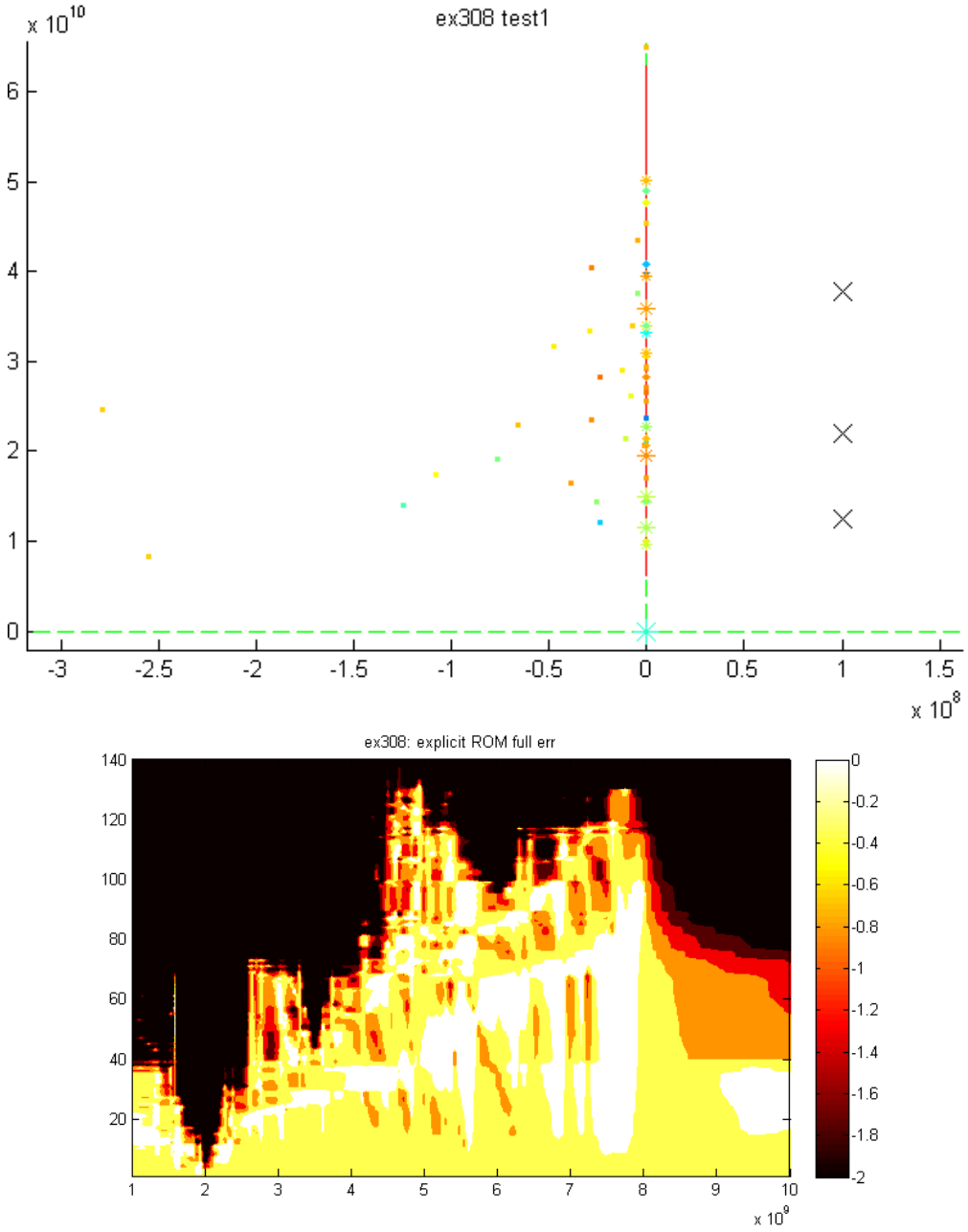


Figure 4.8: The pole distribution for the explicitly-projected transfer function and interpolation-points are in the first plot. The second plot, of local transfer function-error over the frequency range of interest evolving with inclusion of basis vectors in V_{ROM} reflects expansion about the points $p = 2$, 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.

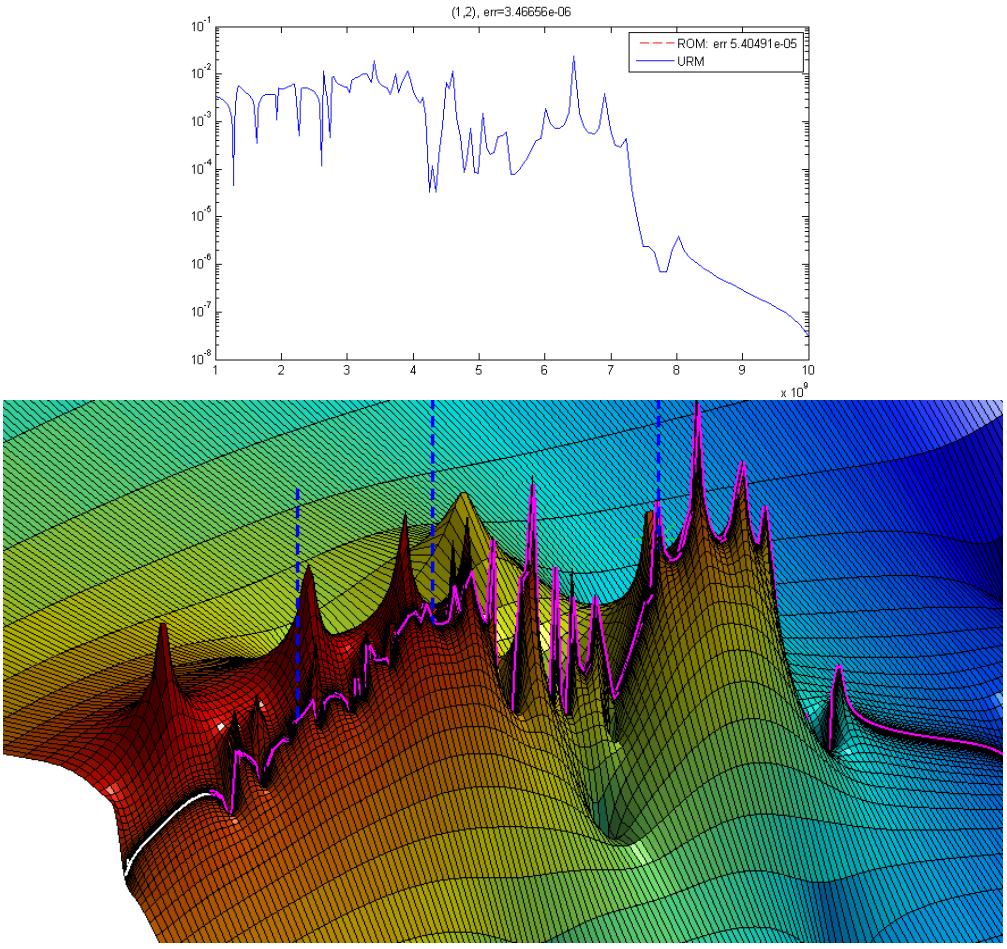


Figure 4.9: Plots of the $(1, 2)$ component of the frequency-response and transfer-function surface for example 1. It is apparent that the interpolation-point placement was almost ideal for `ex308`, in the sense that the points are near centers of pole-mass. Note that the interpolation-points are actually 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.

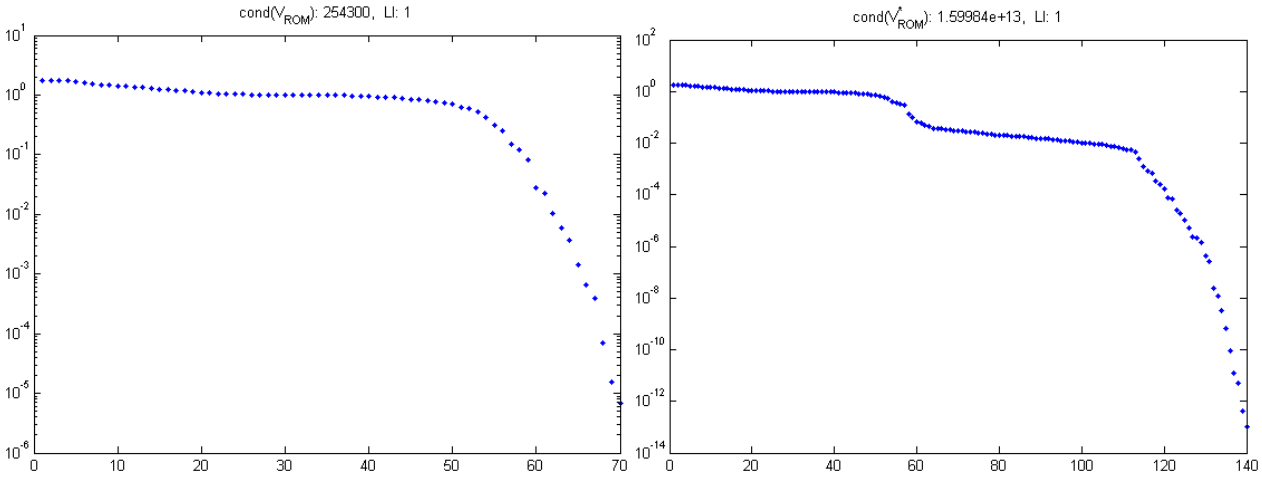


Figure 4.10: Here we provide singular values for the complex basis $\widehat{V}$ produced by thick-restarted band-Arnoldi, and its realification $\widehat{V}^*$.

References

[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. 1.1.4

[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. 2.1.2

[3] P. Benner, M.E. Hochstenbach, and P. Kurschner. Model order reduction of large-scale dynamical systems with jacobi-davidson style eigensolvers. In *Communications, Computing and Control Applications (CCCA), 2011 International Conference on*, pages 1–6, 2011. 3.1

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

[5] David Day and Michael A. Heroux. Solving complex-valued linear systems via equivalent real formulations. *SIAM J. Sci. Comput.*, 23(2):480–498, 2001. 2.1.6

[6] V. Druskin and V. Simoncini. Adaptive rational krylov subspaces for large-scale dynamical systems. *Systems & Control Letters*, 60(8):546 – 560, 2011. 3.1

[7] 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. 4

[8] Michalis Frangos and Imad M. Jaimoukha. Adaptive rational interpolation: Arnoldi and lanczos-like equations. *European Journal of Control*, 14(4):342 – 354, 2008. 3.1

[9] Roland W. Freund. Krylov-subspace methods for reduced-order modeling in circuit simulation. *J. Comput. Appl. Math.*, 123(1-2):395–421, 2000. 2.1.3

[10] Roland W. Freund. Model reduction methods based on Krylov subspaces. *Acta Numerica*, 12:267–319, 2003. 3.2.1

[11] Kyle Gallivan, G Grimme, and Paul Van Dooren. A rational lanczos algorithm for model reduction. *Numerical Algorithms*, 12(1):33–63, 1996. 3.1, 3.1

[12] Juan M Gracia and Francisco E Velasco. Stability of invariant subspaces of regular matrix pencils. *Linear algebra and its applications*, 221:219–226, 1995. 1.1.2

[13] E Grimme and K Gallivan. Krylov projection methods for rational interpolation. 1997. 3.1

[14] E Grimme and K Gallivan. A rational lanczos algorithm for model reduction ii: Interpolation point selection. In *Numerical Algorithms*, 1998. 2.1.4, 3.1

[15] 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. 1.1.5, 2.1.2

[16] Chung-Wen Ho, Albert E. Ruehli, and Pierce A. Brennan. The modified nodal approach to network analysis. *Circuits and Systems, IEEE Transactions on*, 22(6):504–509, 1975. 1.1.3

[17] 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. 1.1.5, 2.1.2

[18] Guillaume Lassaux and K Willcox. Model reduction for active control design using multiple-point arnoldi methods. *AIAA Paper*, 616:2003, 2003. 3.1

[19] Herng-Jer Lee, Chia-Chi Chu, and Wu-Shiung Feng. Multi-point model reductions of vlsi interconnects using the rational arnoldi method with adaptive orders (RAMAO). In *Circuits and Systems, 2004. Proceedings. The 2004 IEEE Asia-Pacific Conference on*, volume 2, pages 1009–1012 vol.2, 2004. 3.1

[20] Herng-Jer Lee, Chia-Chi Chu, and Wu-Shiung Feng. An adaptive-order rational arnoldi method for model-order reductions of linear time-invariant systems. *Linear Algebra and its Applications*, 415:235 – 261, 2006. ¡ce:title¿Special Issue on Order Reduction of Large-Scale Systems¡/ce:title¿. 3.1

[21] RB Lehoucq and KJ Maschhoff. Implementation of an implicitly restarted block arnoldi method. *Preprint MCS-P649-0297, Argonne National Lab*, 1997. 3.2.1

[22] K Henrik A Olsson and Axel Ruhe. Rational krylov for eigenvalue computation and model order reduction. *BIT Numerical Mathematics*, 46(1):99–111, 2006. 3.1

[23] Theodore W Palmer. *Banach Algebras and the General Theory of *-algebras: Volume 1*, volume 2. Cambridge University Press, 2001. 2.1.6

[24] 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. 2.1.2

[25] Beresford N. Parlett and Youcef Saad. Complex shift and invert strategies for real matrices. *Linear Algebra and its Applications*, 8889(0):575–595, 1987. 2.1.4, 2.1.6

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

[27] Axel Ruhe. Rational krylov sequence methods for eigenvalue computation. *Linear Algebra and its Applications*, 58(0):391 – 405, 1984. 3.1

[28] Axel Ruhe. The rational krylov algorithm for nonsymmetric eigenvalue problems. iii: Complex shifts for real matrices. *BIT Numerical Mathematics*, 34(1):165–176, 1994. 2.1.5, 3.1

[29] 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. 1.1.5

[30] Gilbert W Stewart. On the sensitivity of the eigenvalue problem ax=λbx. *SIAM Journal on Numerical Analysis*, 9(4):669–686, 1972. 1.1.2

[31] GW Stewart. A krylov–schur algorithm for large eigenproblems. *SIAM Journal on Matrix Analysis and Applications*, 23(3):601–614, 2002. 13