A Thick-Restarted Krylov Subspace Method for Model Order Reduction

Efrem Rensi

Last update: 6:48pm, April 16, 2014

Contents

Abstract

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

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

Model order reduction (MOR) is a kind of compression. A mathematical model is a description of a working thing, a system. We will address models of a very specific format, namely the system (0.1), but theoretically model reduction can be applied to a model of any working thing whose behavior over time can be observed and influenced (i.e. has output and input). We seek a simpler description of that thing while preserving its behavior in response to various inputs. Order reduction of systems of differential equations like (0.1) has its origins in systems and control theory, where it has been applied to automate processes. If the model correctly predicts a system's response to given input, we can use the model to determine how to efficiently control or influence the system. An early example of modern model reduction is [11] from 1966. Davison introduces the MOR method now known as modal truncation,

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

Of course the large systems we work with today are much larger. At the time of this writing, Krylov subspace methods are being used for systems on the order of $N = 10^9$. Krylov subspace methods are particularly suited for working with systems that are too large for other methods, many of which are quite elegant. The reason why is that the only large matrix operation required is a sparse matrix multiplication (more correctly a sparse solve) of an order N system. If N is large enough, there is not much more we can do.

Krylov projection methods for working with large matrices have been around for a while, and are even claimed to be in the top 10 most important classes of algorithms of the 20th century [13]. The first Krylov subspace projection method for model order reduction was PVL (Pade via Lanczos) [16] developed by Feldman and Freund in 1995. Prior to that, Krylov methods were very successful in

- approximating eigenvalues of large matrices,[29, 4] circa early 1950s,
- approximating the solution $Ax = b$ to large sparse systems ([48] offers a great background), and
- approximating matrix function multiplications $f(A)b$ (for example $e^{A}b$) when A is very large [15].

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

0.1 Model Order Reduction via Subspace Projection

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

$$\begin{aligned} \boldsymbol{E}\frac{dx}{dt} &= \boldsymbol{A}x + \boldsymbol{B}u \\ y &= \boldsymbol{C}^T x \end{aligned} \tag{0.1}$$

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.

A realization is not unique; one model can have multiple realizations. However, two different systems cannot have the same realization. If the model is of a circuit the matrices $\boldsymbol{A}$,$\boldsymbol{E}$,$\boldsymbol{B}$, and $\boldsymbol{C}$ typically come from a circuit representation method called *Modified Nodal Analysis* (MNA) [27] and have a specific structure. A *structure-preserving* model reduction method produces a lower order model that in in fact describes a working circuit. MOR methods are not structure preserving in general – the reduced internal state does not retain any physical meaning.

The state-space realization is perhaps best understood in the context of control theory. $\boldsymbol{A} \in \mathbb{R}^{N\times N}$ is called the *system matrix*, and $E \in \mathbb{R}^{N\times N}$ is the *descriptor matrix*. $\boldsymbol{A}$ and $\boldsymbol{E}$ determine the internal state of the system in the absence of external influence. Note that if there is no input, i.e. $u(t) \equiv 0$, and we don't care about the output y, (0.1) reduces to the unobserved, uninfluenced system

$$\boldsymbol{E}\frac{dx}{dt} = \boldsymbol{A}x. \tag{0.2}$$

The state-space variable $x(t) \in \mathbb{R}^N$ represents the internal state of the system at $t \geq 0$ and we assume a zero initial state $x(0) = 0$.

The possibly singular descriptor matrix $\boldsymbol{E}$ makes (0.2) more general than a set of Ordinary Differential Equations $x' = \boldsymbol{A}x$. Observe that if $\boldsymbol{E}$ is invertible, the first line of (0.1) can be re-written as a set of ODEs

$$\frac{dx}{dt} = \boldsymbol{E}^{-1}\boldsymbol{A}x + \boldsymbol{E}^{-1}\boldsymbol{B}u$$

which is a different and easier to solve problem. System matrix $\boldsymbol{A}$ may be singular as well. 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.

Matrix $\boldsymbol{B} \in \mathbb{R}^{N\times m}$ determines controllability of the system. For example if $\boldsymbol{B} = 0$, the system is (0.2); it effectively has no input. $\boldsymbol{B}$ acts as a filter for input(s) $u(t) \in \mathbb{R}^m$. If $u(t)$ is a scalar function ($m = 1$) then (0.1) is a single-input (SI) system, otherwise it is multi-input (MI).

Matrix $\boldsymbol{C} \in \mathbb{R}^{N\times p}$ determines the observability of the system. $\boldsymbol{C}$ maps the internal state $x(t)$ of the system to p outputs. Observe that $\boldsymbol{C} = 0$ means our system (0.1) has no observable output, i.e. $y(t) \equiv 0$. $y(t) = \boldsymbol{C}^T x \in \mathbb{R}^p$ represents the output(s) of the system – the observable manifestation of its internal state x. If $p = 1$ then (0.1) is a single-output (SO) system; its response (output) is a scalar-valued function, otherwise it is multi-output (MO).

We assume that the order-N system (0.1) is too large to work with and and we want a model that behaves like (0.1), 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 (0.1) 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

$$\begin{aligned} \boldsymbol{E}_n\frac{d\tilde{x}}{dt} &= \boldsymbol{A}_n\tilde{x} + \boldsymbol{B}_n u \\ \tilde{y} &= \boldsymbol{C}_n^T\tilde{x}, \end{aligned} \tag{0.3}$$

with

$$\boldsymbol{A}_n = V^T \boldsymbol{A} V, \quad \boldsymbol{E}_n = V^T \boldsymbol{E} V, \quad \boldsymbol{C}_n = V^T \boldsymbol{C}, \quad \boldsymbol{B}_n = V^T \boldsymbol{B}, \tag{0.4}$$

where $\tilde{x}(t) \in \mathbb{C}^n$ is the state of the reduced order system such that $V\tilde{x}(t)$ approximates the state of the unreduced model. The p output(s) $\tilde{y}(t) \in \mathbb{R}^p$ approximate $y(t) \in \mathbb{R}^p$ from (0.1), given the same m input(s) $u(t) \in \mathbb{R}^m$, and ideally $\|y - \tilde{y}\|$ is small. All model reduction methods that approximate solutions to (0.1) with solutions to a system (0.3) given one orthogonal basis V are known as orthogonal projection methods.

We will call (0.3), (0.4) the *explicitly-projected* ROM. There is another ROM formulation derived from subspace projection and we will call that the implicitly projected model, but the primary objective of this text is an exploration of subspaces and their bases V that yield (0.3).

0.1.1 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 (0.1) 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 (0.1) and assuming a zero initial condition $X(0) = 0$, which yields the algebraic equations

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

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

$$\mathcal{H}(s) = \boldsymbol{C}^T \left(s\boldsymbol{E} - \boldsymbol{A}\right)^{-1} \boldsymbol{B} \quad \in \quad (\mathbb{C} \cup \infty)^{p \times m} \tag{0.5}$$

is the transfer function over $\mathbb{C}$. Note that $\mathcal{H}(s)$ is defined only if the *matrix pencil* $(\boldsymbol{A}, \boldsymbol{E})$ is *regular*, meaning $(\mu\boldsymbol{E} - \boldsymbol{A})$ is singular for only a finite number of eigenvalues $\mu \in \mathbb{C} \cup \infty$. A state-space model can have several different realizations, but its transfer function (0.5) is unique.

For a general MIMO transfer function (0.5) where $\boldsymbol{C} = \begin{bmatrix}\boldsymbol{c}_1 & \boldsymbol{c}_2 & \cdots & \boldsymbol{c}_p\end{bmatrix}$ and $\boldsymbol{B} = \begin{bmatrix}\boldsymbol{b}_1 & \boldsymbol{b}_2 & \dots & \boldsymbol{b}_m\end{bmatrix}$, we can consider (0.5) to be mp scalar-valued SISO (single input single output) transfer functions

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

and for example in the 2×2 case we have

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

For most discussion there is no loss of generality to have our examples be SISO models (scalar-valued transfer functions). It does matter that our basis V for projection is determined considering every input-output (controllability-observability) pair $\boldsymbol{b}_j, \boldsymbol{c}_i$ of a MIMO model.

Typically in model reduction we are interested in approximating what is known as the ***frequency response*** $\mathcal{H}(2\pi if)$ of the system, with $f \in [f_{\min}, f_{\max}]$ (equivalent notation is $\mathcal{H}(i\omega)$ for $\omega \in [\omega_0, \omega_1]$), which is the transfer function over a segment on the the imaginary axis. The frequency response of a system (0.1) indicates its steady-state response to an impulse. If the reader considers that a passive model is one that continues indefinitely after being given a little shove, the frequency response represents the behavior that the model settles into after said shove. To be precise, the "shove" is Dirac's Delta function as input ($u(t) = \delta(t)$), which is covered in §**??** or any linear system theory text. Note that the domain for frequency response $\mathcal{H}(i\omega)$ can be any range of frequencies $\omega > 0$, which corresponds to $\mathcal{H}(s)$ with $s \in \mathbb{C}$ on the positive $\Im$-axis. The application for MOR usually suggests a reason to look at a particular subset of frequencies. For the circuit model examples that we mostly use here, we consider $f \in [0, 10^{10}]$ and more often, higher frequency response $\|\mathcal{H}(2\pi if)\|$ for $f \in [10^8, 10^{10}]$. It may not be obvious so we will point out that $[10^8, 10^{10}]$ is a much larger interval than $[10^0, 10^8]$; 100 times larger, in fact, which is why the interval S in figure 1b corresponding to so-called "high frequencies" appears to be disproportionally large.

Visual representations

The traditional way to visualize the frequency-response of a SISO LTI system is with the gain *Bode plot* figure 1a, which is the transfer function magnitude $\|\mathcal{H}(s)\|$ (known as gain), over the frequency range of interest. Gain is usually plotted on a logarithmic scale (sometimes in decibels). All of the gain plots in this document are plotted with gain on a log-scale and linear scaling for the domain (f-axis or $\Re(s), \Im(s)$-axes). Model order reduction method publications often visually compare methods with superimposed gain plots of an example system, as in figure 4. Sharp peaks and valleys in the plot, sometimes called features, indicate nearby poles and zeros of the function. Poles are points μ such that $\|\mathcal{H}(\mu)\| = \infty$, and zeros are z such that $\mathcal{H}(z) = 0$, or with plotting in mind, $\log \|\mathcal{H}(z)\| = -\infty$; gain is plotted on a logarithmic scale, so a zero of the function corresponds to $-\infty$ on the plot. Poles and zeros are opposites of each other in that sense. Poles are apparently named so because the function magnitude resembles a canvas being held up by poles, as in figure 3. To get an intuitive understanding of the pole/zero decomposition of a transfer function, it may be helpful to consider the gain plot over a larger region of $\mathbb{C}$, as in figure 2, where the frequency response can be seen as a path along a sort of generalized mountain range created by dominant poles and zeros.

Pole-zero or pole-only plot representations, such as the pole plots in figure 6 convey similar information about the transfer function. In this document we will mostly ignore zeros of the function, not because they are not important, but because they are not as readily available with the methods covered here. A Krylov subspace projection method determines poles of the transfer function (0.5) as a byproduct of constructing a basis for subspace projection. Determining zeros is possible but requires extra computation.

We cannot address it here, but finding and employing information about transfer function zeros as part of Krylov method may be a promising area for further research, as well as a better theory of transfer function interpolating points, that includes zeros defined independently of poles. It is clear from figure 2 that transfer function zeros, as well as poles are equally involved in characterizing the function.

Pole-zero plots, or in our case pole-only plots, like those in figure 6, sometimes have a way to indicate the dominance, or "weight" of a pole. The transfer function can be expressed as a sum of terms, each term involving a distinct pole, in the so-called pole-residue representation

$$\mathcal{H}(s) = x_0 + \sum_j \frac{x_j}{s - \mu_j}.$$

For a pole μ_j, the magnitude $|x_j|$ of the associated residue, as well as its proximity to the domain of $\mathcal{H}(s)$, determine its significance in the sum. Figure 5 illustrates how we represent pole weight in pole plots. Notions and measures of pole weight, or dominance, will be discussed in §0.1.1.

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 2 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 $i < j$.

Poles and zeros of a MIMO model transfer function must be defined differently than for SISO models. For general $\mathcal{H}(s) \in \mathbb{C}^{p\times m}$, we have that $\|\mathcal{H}(s)\| = \infty$ for any values s such that $|\mathcal{H}_{ij}(s)| = \infty$ for any of the mp component SISO models. Then we can define a pole of a MIMO model transfer function as $\mu \in \mathbb{C}$ such that

$$\max_{ij} \left[|\mathcal{H}(\mu)|\right]_{ij} = \infty.$$

In other words μ is a pole of $\mathcal{H}$ if *any* entry of $\mathcal{H}(\mu)$ is ∞. Similarly, we may define a MIMO transfer function zero to be a value $\zeta \in \mathbb{C}$ for which *any* $\mathcal{H}_{ij}(\zeta) = 0$, i.e.

$$\min_{i,j}[|\mathcal{H}(\zeta)|]_{ij} = 0.$$

A note about Implementation In order to make one of the included transfer plots we must evaluate the transfer function $\mathcal{H}(s) = \boldsymbol{c}^T(s\boldsymbol{E} - \boldsymbol{A})^{-1}\boldsymbol{b}$ at a number of sample points $s \in \mathbb{C}$, which can be large. In order to make this efficient, we generally must work with a reduced-order model. Assuming that the size of the model (or ROM) is manageable, it helps to make function evaluation as efficient as possible. One way to make evaluation faster is to work with the QZ factorization of $(\boldsymbol{A}, \boldsymbol{E})$, which yields orthonormal basis matrices Q and Z, and upper-triangular $\widehat{\boldsymbol{A}}$ and $\widehat{\boldsymbol{E}}$ such that

$$\widehat{\boldsymbol{A}} = Q\boldsymbol{A}Z \qquad \text{and} \qquad \widehat{\boldsymbol{E}} = Q\boldsymbol{E}Z.$$

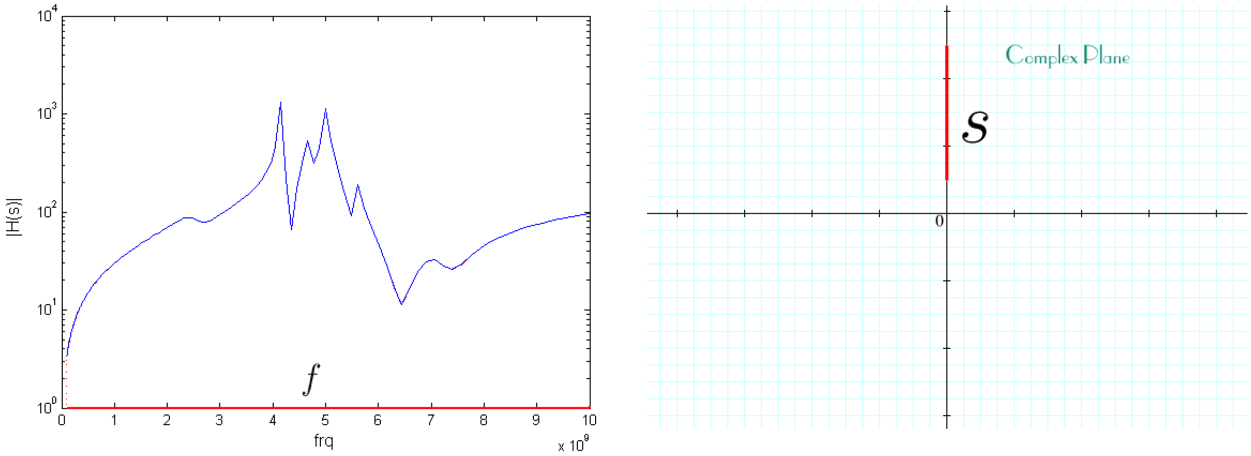
Then

$$s\boldsymbol{E} - \boldsymbol{A} = Q^H(s\widehat{\boldsymbol{E}} - \widehat{\boldsymbol{A}})Z^H,$$

for any $s \in \mathbb{C}$, so

$$\begin{aligned}
\mathcal{H}(s) &= \boldsymbol{c}^T(s\boldsymbol{E} - \boldsymbol{A})^{-1}\boldsymbol{b} \\
&= \boldsymbol{c}^T\left[Q^H(s\widehat{\boldsymbol{E}} - \widehat{\boldsymbol{A}})Z^H\right]^{-1}\boldsymbol{b} \\
&= (\boldsymbol{c}^T Z)(s\widehat{\boldsymbol{E}} - \widehat{\boldsymbol{A}})^{-1}(Q\boldsymbol{b}).
\end{aligned}$$

QZ factorization and computation of $\boldsymbol{c}^T Z$ and $Q\boldsymbol{b}$ are done only once, and each solve $(s\widehat{\boldsymbol{E}} - \widehat{\boldsymbol{A}})^{-1}(Q\boldsymbol{b})$ is a triangular back-solve. This is how most of the transfer function plots in this document were computed.



(a)

(b)

Figure 1: Frequency response gain $|\mathcal{H}(2\pi i f)|$ for a single-input, single-output (SISO) model with $f \in \left[10^8, 10^{10}\right]$. It is plotted on a semilog scale (logarithmic for gain-axis, linear for f-axis). The plot (a) is what $|\mathcal{H}(s)|$ looks like over the segment S in (b). We are generally interested in approximating $\mathcal{H}(s)$ accurately over an interval $S = i(\omega_0, \omega_1)$ on the $\Im$-axis.

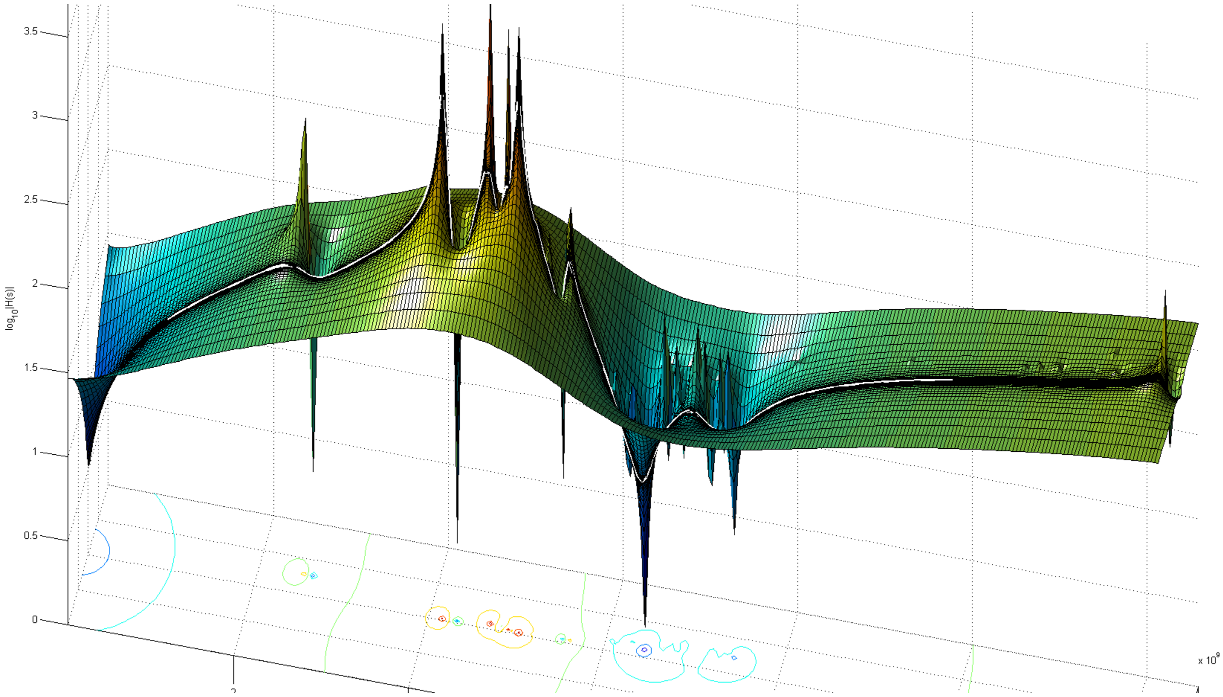


Figure 2: Transfer function gain $|\mathcal{H}(s)|$ of example system `ex1481s11`, plotted over a rectangular region $(-10^9, 10^9) \times i(10^8, 10^{10.1})$ of the complex-plane. Scaling is kept as close to square as possible. The frequency response (also shown in figure 1a) is highlighted over $i(10^9, 10^{10})$ in white, and the contour plot below indicates the dominant poles and zeros near the $\Im$-axis. The influence of poles and zeros on the frequency response is clear in this context. We can assume there is an influential zero somewhere to the left of this plot. It is actually at $s = 0$. Scaling of this plot is linear for the domain (both axes) and logarithmic for the function. This transfer function plot is actually an $n = 300$ ROM of a model whose original size is $N = 1841$. For this model, $n = 300$, by virtually any method, is large enough to be a very good approximation. Indeed, we rely on model reduction to make such a plot, as computing it with the original model would take too long to be practical with available computers. It involves solving a 300×300 system $\mathcal{H}(s) = \boldsymbol{C}^T(\boldsymbol{A} - s\boldsymbol{E})^{-1}\boldsymbol{B}$ for every value of s, rather than an 1841×1841 system.

Figure 3: A tent with poles. This is one way to conceptualize the transfer function magnitude (gain) $\|\mathcal{H}(s)\|$ over $\mathbb{C}$, with poles μ_j where $\|\mathcal{H}(\mu_j)\| = \infty$. Source:[**?**]

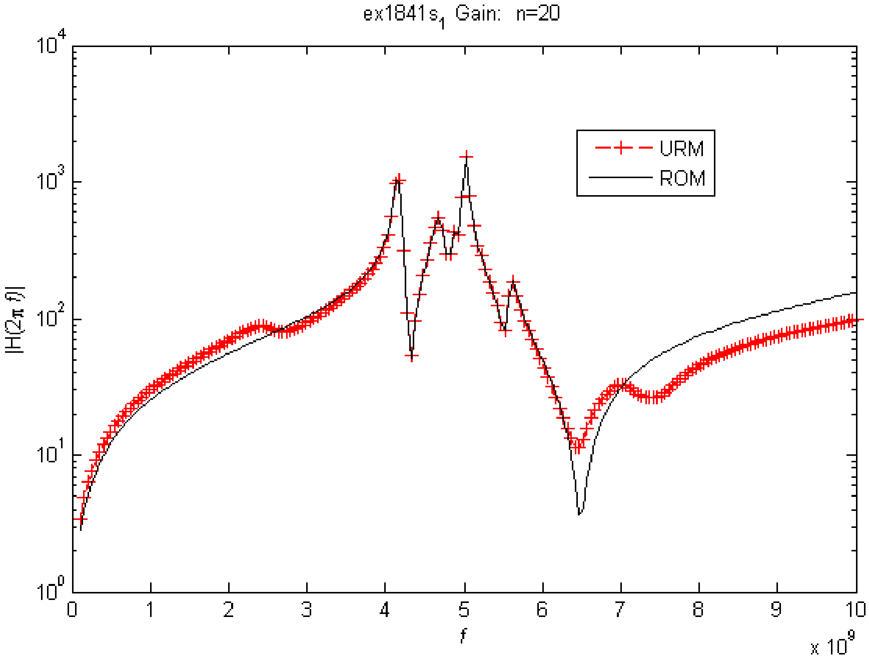


Figure 4: A visual comparison of system frequency response $|\mathcal{H}(2\pi i f)|$ over $f \in [10^8, 10^{10}]$ for two models. In this case we are comparing the original, unreduced LTI sysem model `ex1481s11`, of order $N = 1841$, with a reduced order model of size $n = 20$. It is plotted here with 200 linearly-distributed sample points. Rarely in practice do we know what the URM transfer function looks like in advance, but we often do tests and comparisons using benchmark examples like `ex1841s11`. This example shows fairly good approximation around the center of the frequency range displayed, but not on either side.

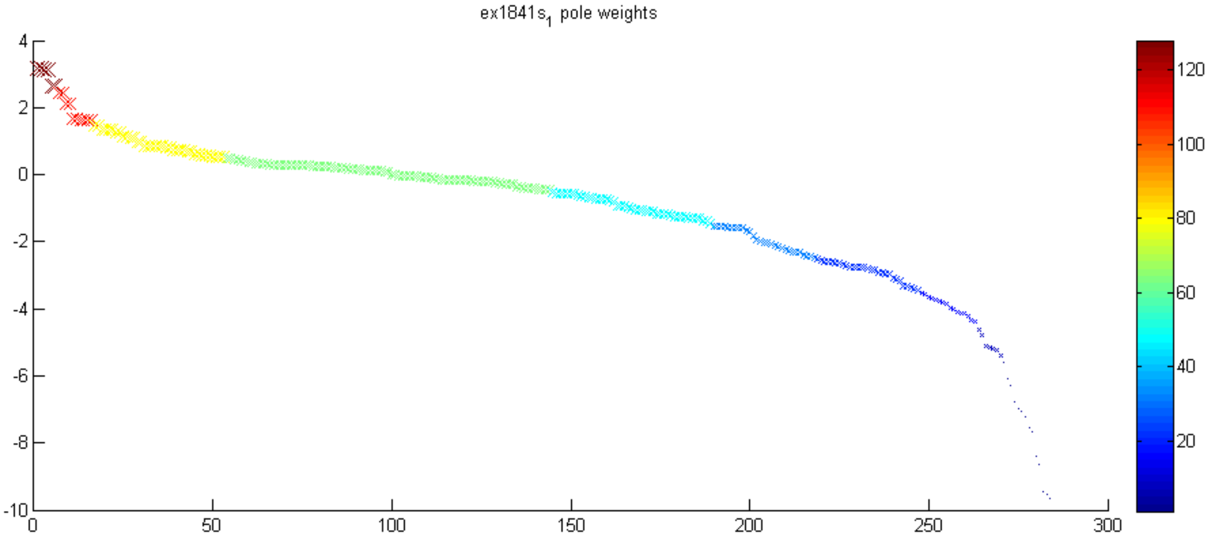


Figure 5: This is the descending sorted *order-of-magnitude* (log) of the pole weights, for the most significant (aka dominant) poles of an order $n = 300$ ROM of our example SISO model `ex1841s11`. Each pole's marker is sized according to $\log_{10}$ of the pole's weight. Each pole corresponds to one term in a sum, and order of pole weight corresponds to the order (roughly what power of 10) of its term in the sum. In this plot we use the same mapping of pole weight to size, to define a color-map, thus dramatizing the variation of pole significance. The range of consideration is set at 12 orders of magnitude, which here is about 10^{-9} to 10^3. Any pole with weight less than 10^{-9} is thus ignored, or given a marker of size zero. Each increase in marker size $(1, 2, ..., 12)$ represents one increase in order-of-magnitude.

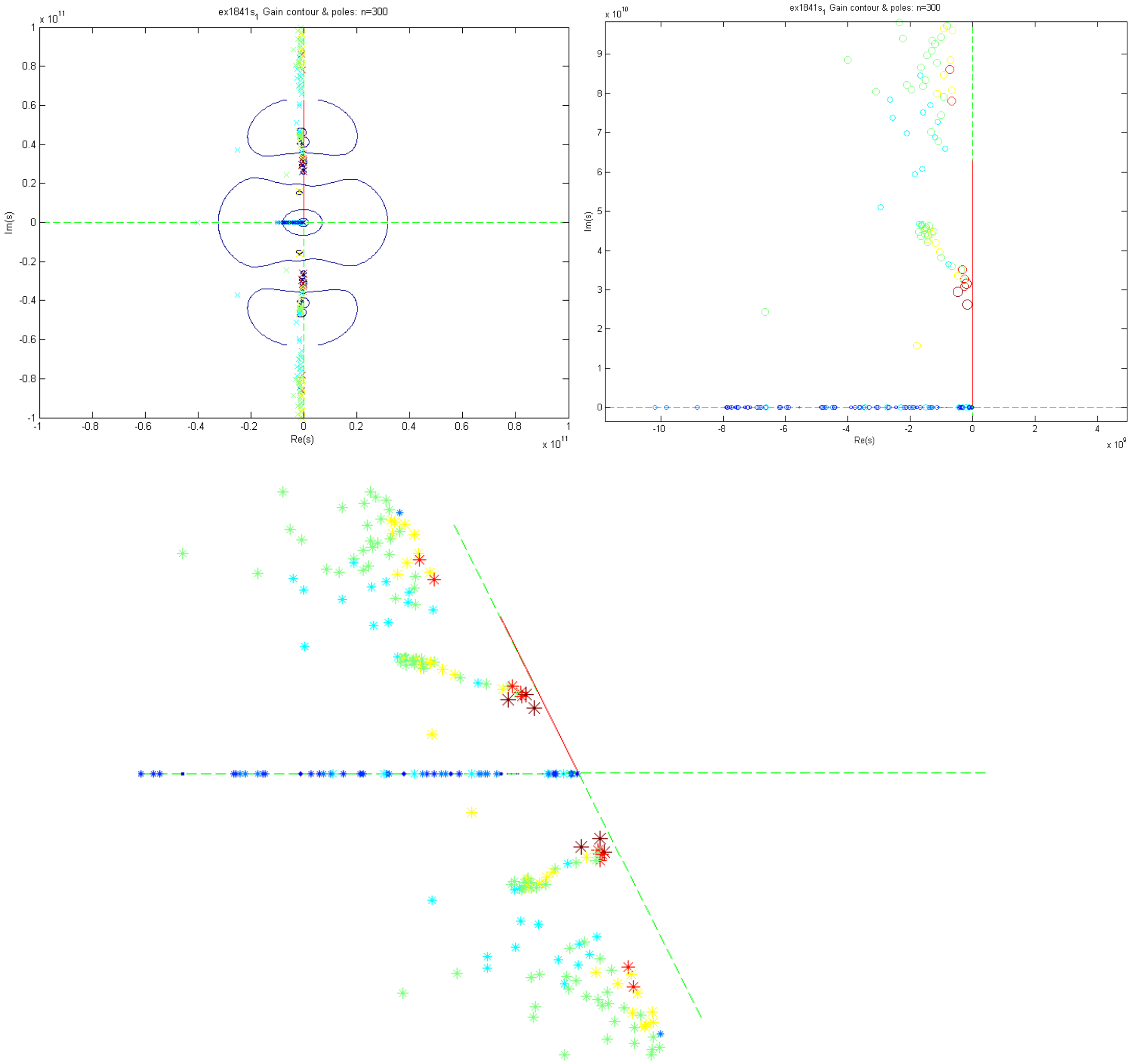


Figure 6: Here are a few plots of poles (and not zeros) of example transfer function `ex1841s11` from figures 4 and 2. Poles are represented with differing markers but sizing and coloring scheme is defined in figure 5. The upper-left pole plot includes a contour plot of the transfer function. Note that the poles are symmetric about the $\Re$-axis. The transfer function is symmetric about the real axis. All poles are either real, or complex-conjugate pairs. Observe that no poles have positive real-part, which is a necessary condition for a stable model. The segment of interest on the positive $\Im$-axis corresponds to $2\pi i f$ for $f \in [10^8, 10^{10}]$. This segment appears to extend to the origin because the scaling is linear; the length of the segment $[10^8, 10^{10}]$ is actually 100 times larger than that of $[10^0, 10^8]$. Dominant poles are defined in this document as poles that are most influential on the transfer function over the segment of interest. Observe that the most dominant of the poles are located near the center of this region, indicating peaks there, which is in agreement with figures 4 and 2. Pole locations, notably, *do not* indicate the locations of valleys in a frequency response plot, which is a significant feature of the frequency response.

The transfer function (0.5) can be represented as a quotient of two polynomials, which is another way to look at poles and zeros. Recall from the derivation of transfer function $\mathcal{H}(s)$, that the input $U(s) \in \mathbb{C}^m$ and output $Y(s) \in \mathbb{C}^p$ of the system (0.1) in the frequency domain are related by

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

where $\mathcal{H}(s) \in \mathbb{C}^{p\times m}$ is the Transfer Function. For a SISO model ($m = p = 1$) of order N, where the transfer function $\mathcal{H}(s) \in \mathbb{C}$ is scalar-valued, it is the quotient

$$\begin{aligned} \mathcal{H}(s) &= \boldsymbol{c}^T \left(s\boldsymbol{E} - \boldsymbol{A}\right)^{-1} \boldsymbol{b} \\ &= \frac{\beta_0 + \beta_1 s + \beta_2 s^2 + \cdots + \beta_\gamma s^\gamma}{\alpha_0 + \alpha_1 s + \alpha_2 s^2 + \cdots + \alpha_q s^q} \\ &= k_o \frac{(s - z_1)(s - z_2)\cdots(s - z_\gamma)}{(s - \mu_1)(s - \mu_2)\cdots(s - \mu_q)} \end{aligned} \tag{0.6}$$

of two polynomials with $\gamma < q \leq N$. The denominator is of degree $q \leq N$ and its roots are the finite poles of $\mathcal{H}(s)$. The number of zeros γ is some $\gamma < q$; for some reason we assume there are fewer zeros than poles.

The simple transfer function

$$\mathcal{H}(s) = \frac{6s^2 + 18s + 12}{2s^3 + 10s^2 + 16s + 12} = 3\frac{(s + 1)(s + 2)}{(s + 1 + i)(s + 1 - i)(s + 3)} \tag{0.7}$$

which is plotted in figure 7, has three poles and two zeros.

Transfer function moments

Krylov subspace projection methods boast *moment matching* properties. The reduced order model transfer function implied by a Krylov subspace method 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

$$\mathcal{H}(s) = \sum_{j=0}^{\infty}(s - s_0)^j \mathcal{H}^{(j)}, \quad \text{or equivalently,} \quad \mathcal{H}(s + s_0) = \sum_{j=0}^{\infty} s^j \mathcal{H}^{(j)} \tag{0.8}$$

where the Taylor coefficient

$$\mathcal{H}^{(j)} = \frac{1}{j!}\frac{d^j\mathcal{H}}{ds^j}\bigg|_{s=s_0} \tag{0.9}$$

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

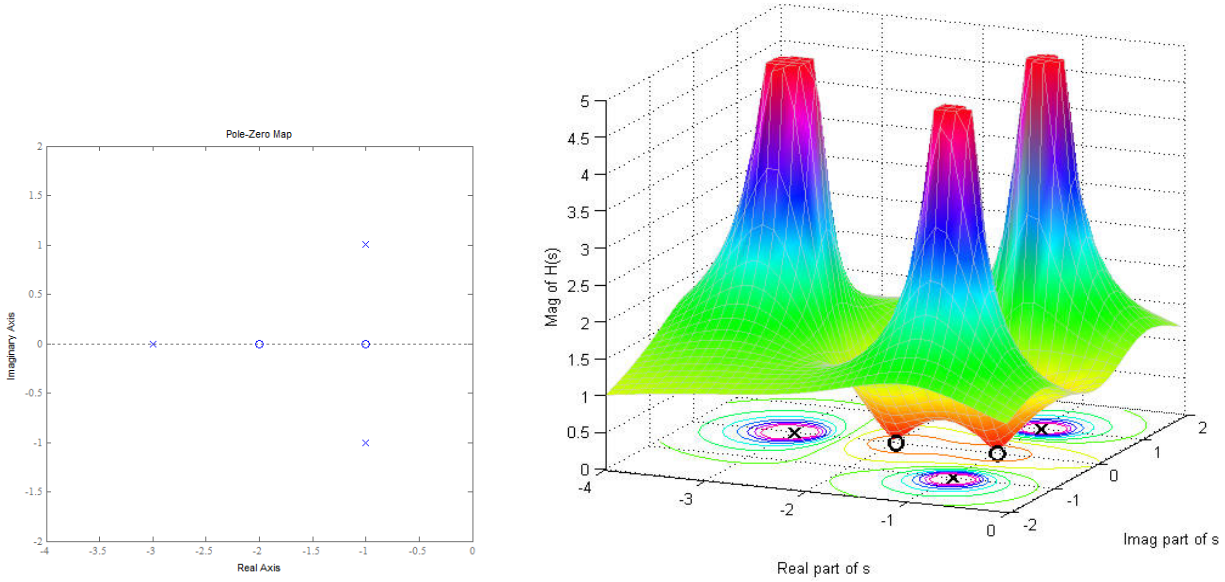
$$\widehat{\mathcal{H}}(s) = \widehat{\mathcal{H}}^{(0)} + (s - s_0)\widehat{\mathcal{H}}^{(1)} + (s - s_0)^2\widehat{\mathcal{H}}^{(2)} + \cdots + (s - s_0)^{n-1}\widehat{\mathcal{H}}^{(n-1)} + \cdots$$

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; indeed, they are often referred to as *interpolation* points and using several such points is called *rational-interpolation.*



(a)

(b)

Figure 7: This is the plot of gain $\|\mathcal{H}(s)\|$ for the transfer function (0.7), which has 3 poles and 2 zeros (source: [9]). Poles are points $\mu \in \mathbb{C}$ where $|\mathcal{H}(\mu)| = \infty$. If our reduced order model has about the same distribution and influence of major (dominant) poles and zeros, then we can say it is a good approximation. Unfortunately there are no proven results for error bounds with pole/zero matching, but it makes intuitive sense.

Shift-invert representation

Moment matching properties of Krylov subspace methods are accomplished via the the following reformulation of the transfer function (0.5).

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

$$\begin{aligned}\mathcal{H}(s) &= \boldsymbol{C}^T \left(s\boldsymbol{E} - \boldsymbol{A}\right)^{-1} \boldsymbol{B} \\ &= \boldsymbol{C}^T \left(s_0\boldsymbol{E} - \boldsymbol{A} + (s - s_0)\boldsymbol{E}\right)^{-1} \boldsymbol{B} \\ &= \boldsymbol{C}^T \left(I - (s - s_0)\mathbf{H}\right)^{-1} \mathbf{R} \end{aligned} \tag{0.10}$$

where

$$\mathbf{H} := (\boldsymbol{A} - s_0\boldsymbol{E})^{-1}\boldsymbol{E} \quad \text{and} \quad \mathbf{R} := (s_0\boldsymbol{E} - \boldsymbol{A})^{-1}\boldsymbol{B}. \tag{0.11}$$

(0.10) is sometimes called the *shifted* transfer function formulation, with *shift* s_0, although it does not depend on s_0. (0.10) is the same function regardless of the value of s_0 as long as $\boldsymbol{A} - s_0\boldsymbol{E}$ is invertible.[1] The generally non-sparse $\mathbf{H} = \mathbf{H}(s_0) \in \mathbb{C}^{N\times N}$ is a sort of operator or multiplier, and $\mathbf{R} = \mathbf{R}(s_0) \in \mathbb{C}^{N\times p}$ is a start-block or start-vector. $\mathbf{H} = \mathbf{H}(s_0)$ is sometimes called a *shifted-inverse operator*, *shift-and-invert operator*, or *rational operator*, with shift s_0. $\mathbf{H}$ and $\mathbf{R}$ are the building blocks for the moments of the transfer function about s_0. It should be noted that $\mathbf{H}$ is dense in general and is rarely if ever explicitly formed. We only need a way to obtain matrix-vector products $\mathbf{H}v$ for vectors $v \in \mathbb{C}^N$.

The shifted transfer function representation (0.10) can alternatively be considered the transfer function for the shifted descriptor system

$$\begin{aligned}\mathbf{H}\frac{dx}{dt} &= (I - s_0\mathbf{H})x + \mathbf{R}u \\ y &= \boldsymbol{C}^T x,\end{aligned} \tag{0.12}$$

which is equivalent to (0.1) 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 $\operatorname{span} V$, as opposed to the *explicitly* projected model (0.3). A model obtained via implicit projection is not equivalent to (0.3) 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 (0.10) as

$$\begin{aligned}\mathcal{H}(s) &= \boldsymbol{C}^T \left(\sum_{j=0}^{\infty}(s - s_0)^j\mathbf{H}^j\right)\mathbf{R} \\ &= \sum_{j=0}^{\infty}(s - s_0)^j\boldsymbol{C}^T\mathbf{H}^j\mathbf{R}.\end{aligned} \tag{0.13}$$

The moments $\mathcal{H}^{(j)}$ from (0.8) are specified exactly in (0.13):

$$\mathcal{H}^{(j)} = \boldsymbol{C}^T\mathbf{H}^j\mathbf{R} \tag{0.14}$$

Moments (0.14) suggest the following moment matching method: Compute n terms of the sequence

$$\mathbf{R}, \quad \mathbf{HR}, \quad \mathbf{H}^2\mathbf{R}, \quad \ldots, \tag{0.15}$$

and then left-multiply the observability constraint $\boldsymbol{C}^T$ by each one, giving us n terms of the Taylor series (0.13). The first Krylov subspace projection methods of the early 90s did just that, although moments were not recognized as such at the time. It was not thought feasible or necessary to match more than two or three.

[1] To verify (0.10), note that $I = (s_0\boldsymbol{E} - \boldsymbol{A})^{-1}(s_0\boldsymbol{E} - \boldsymbol{A})$.

Region of convergence for moment matching The power (Taylor) series representation seems to imply that (0.13) 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 (0.13) 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.

Padé approximation

Moment matching is arguably the only reliable means by which we can ensure the quality of a reduced order model, and moment matching techniques have been used at least since the early 70s. The concept is simple: a reduced model matching more moments about s_0 is a more accurate approximation around s_0. Moment matching is obviously only a local quality measure. We can match moments about several points, thus approximating the transfer function via rational interpolation, but there is currently no consistent (from model to model) and easily computable notion of *global* convergence.

How many moments about s_0 can we hope to match with a ROM of size (order) $q \ll N$?

There is, in fact, an upper limit to how many moments a size n ROM can match. For a SISO model, that is $2q$. Equivalently, a reduced model that matches n moments must be at least of order $n/2$. A reduced model with this optimal moment matching property is called a Padé model. Padé approximation is historically important in model reduction. In the past, reduction of LTI systems via transfer function approximation were done with low-order rational approximants, using coefficients listed in so-called Routh and Padé tables [49, 50].

- For a SISO model the transfer function (0.5) is a scalar-valued proper rational function over $\mathbb{C}$. which suggests the **Padé approximant** *about* s_0, *of order* (p, q),

$$\mathcal{H}_{p,q}(s + s_0) = \frac{b_0 + b_1 s + b_2 s^2 + \cdots + b_p s^p}{1 + a_1 s + a_2 s^2 + \cdots + a_q s^q}, \tag{0.16}$$

  with $p < q$.[2] Note that $a_0 = 1$, so that a Padé representation of order (p, q) has $p + q + 1$ parameters. MIMO model transfer functions have an analogous *Matrix Padé* representation.

For a LTI (linear) SISO system model of order N, the transfer function is in fact a quotient of two polynomials and can be represented *exactly* by (0.16), with appropriate $p < q \leq N$. The convention in model reduction via Padé approximation is to assume that $p = q - 1$ in (0.16), and refer to the Padé approximant $\mathcal{H}_{q-1,q}$ with $2q$ parameters, as $\mathcal{H}_q$. The Padé model of order q is an optimal $2q$-parameter model and we cannot do better with $2q$ coefficients.

Consider the following equivalent representations of the transfer function (0.5): The order-q Padé approximant

$$\mathcal{H}_q(s + s_0) = \frac{b_0 + b_1 s + b_2 s^2 + \cdots + b_p s^{q-1}}{1 + a_1 s + a_2 s^2 + \cdots + a_q s^q} \tag{0.17}$$

$$= x_\infty + \sum_{j=1}^{q'} \frac{x_j}{s - \mu_j} \qquad q' \leq q \tag{0.18}$$

[2] note that $\mathcal{H}(s + s_0) = \frac{P(s)}{Q(s)}$ is equivalent to $\mathcal{H}(s) = \frac{P(s-s_0)}{Q(s-s_0)}$. We use the former for simpler notation.

with $q \leq N$, and the Taylor series expansion

$$\mathcal{H}(s+s_0) = \sum_{j=0}^{\infty} \mathcal{H}^{(j)} s^j, \tag{0.19}$$

both approximations centered at s_0. (0.17) is a restatement of (0.16) with $p = q - 1$. (0.18) is the partial fraction decomposition of (0.17) and is called the *pole-residue* formulation of the transfer function. There are $q' \leq q$ finite poles $\mu_j \in \mathbb{C}$ of the transfer function, and the x_j are the associated residues. The constant term x_∞ is the residue associated with poles at ∞, which we will address later.

Padé approximation and moment matching are related by equating (0.17),(0.18) with (0.19). A Padé model of order q is theoretically constructed by setting

$$\sum_{j=0}^{2q-1} \mathcal{H}^{(j)} s^j = \frac{b_0 + b_1 s + b_2 s^2 + \cdots + b_p s^{q-1}}{1 + a_1 s + a_2 s^2 + \cdots + a_q s^q}$$

and solving a $q \times q$ system for coefficients a_j of the denominator polynomial of (0.18), given that moments $\mathcal{H}^{(j)}$ about s_0 are given by $\mathcal{H}^{(j)} = \boldsymbol{C}^T \mathbf{H}^j \mathbf{r}$.

Constructing an order-q Padé model from $2q$ explicitly computed moments $\mathcal{H}^{(j)} = \boldsymbol{C}^T \mathbf{H}^j \mathbf{r}$ is a viable model-reduction method in exact arithmetic, but it involves a badly-conditioned solve and the iterates $\mathbf{H}^j \mathbf{r}$ converge to an eigenvector of $\mathbf{H}$ as $j > 0$ increases. It is not practical for $j > 10$, however it was at one time a popular model reduction method called AWE, which has since given way to iterative subspace methods, such as those exploiting the Krylov iteration. The underlying moment matching property is still the appeal of these methods, which are considered Padé or *Padé-type* methods and considered to be descendants of AWE. A Padé-type method is one that employs a moment matching strategy, allthough the reduced model of a given size is not optimal in the number of moments matched. Most contemporary moment matching methods produce Padé-type models, mostly because true Padé reduced models are not guaranteed to be stable, even if the original model is stable.

The analog to Padé for for a general MIMO model is called matrix-Padé, and since moments are $p \times m$ matrices we have a different bound for the number of moments matched by a matrix-Padé model.

The approximation $\widehat{\mathcal{H}}_n(s)$ of size n that satisfies

$$\mathcal{H}(s) = \widehat{\mathcal{H}}_n(s) + \mathcal{O}\left((s-s_0)^{q(n)}\right) \tag{0.20}$$

is called a *Padé approximation* if the number of matched-moments $q(n)$ is as large as possible. Otherwise, if any moments are matched we call it a Padé-type approximation. It was shown by [19] that

$$q(n) \leq \left\lfloor \frac{n}{m} \right\rfloor + \left\lfloor \frac{n}{p} \right\rfloor, \tag{0.21}$$

where m is the number of inputs and p is the number of outputs of the system. This is because the moments are $p \times m$ blocks. Since each moment

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

is a $p \times m$ (possibly complex-valued) block, each further moment matched requires a possibly large increase in the size of the reduced model. In the $m = p = 1$ (SISO) case the Padé (optimal) approximation exhibits $q(n) = 2n$, or $2n$ moments matched for an approximate model of dimension n. Although it is possible to produce a Padé approximation of the transfer function, we know from [19] that such a model in general does not preserve stability or passivity. For a more reliable approximation we expect $q(n) = n$; a SISO model of size n (determined by n-iterations about s_0) matches n moments about s_0.

The definition of a Padé model about an expansion point s_0 extends to multiple points and is discussed in [8].

Asymptotic Waveform Evaluation (AWE) This is a brief description of the AWE method of [43], which has historical and computational significance in that it exemplifies *explicit moment matching*. The moments $\mathcal{H}^{(j)}(s_0) = \boldsymbol{C}^T\mathbf{H}^j\mathbf{r}$ about s_0 are explicitly computed. In contrast, modern methods often produce models with moment matching properties, but moments of the model are never actually computed. AWE was one of the early attempts at matching more than one or two moments of a system transfer function. It is also a good example of how mathematically equivalent methods can be very different when implemented on a computer. AWE and the later PVL (Padé via Lanczos), which is a Krylov subspace projection method, in theory both produce a Padé model by matching transfer function moments about s_0, so they are mathematically equivalent.

The SISO transfer function has the Padé form

$$\mathcal{H}_q(s) = \frac{P(s)}{Q(s)} \tag{0.22}$$

where

$$P(s) = \sum_{j=0}^{q-1}(s-s_0)^j b_j \qquad \text{and} \qquad Q(s) = \sum_{j=0}^{q}(s-s_0)^j a_j.$$

Then

$$\underbrace{\left(\sum_{j=0}^{q-1}(s-s_0)^j b_j\right)}_{P(s)} = \underbrace{\left(\sum_{j=0}^{2q-1}(s-s_0)^j \mathcal{H}^{(j)}\right)}_{\widehat{\mathcal{H}}_q(s)} \underbrace{\left(\sum_{j=0}^{q}(s-s_0)^j a_j\right)}_{Q(s)},$$

which suggests we solve the system

$$\begin{bmatrix} \mathcal{H}^{(0)} & \mathcal{H}^{(1)} & \cdots & \mathcal{H}^{(q-1)} \\ \mathcal{H}^{(1)} & \mathcal{H}^{(2)} & \cdots & \mathcal{H}^{(q)} \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{H}^{(q-1)} & \mathcal{H}^{(q)} & \cdots & \mathcal{H}^{(2q-1)} \end{bmatrix} \begin{bmatrix} a_q \\ a_{q-1} \\ \vdots \\ a_1 \end{bmatrix} = \begin{bmatrix} \mathcal{H}^{(q)} \\ \mathcal{H}^{(q+1)} \\ \vdots \\ \mathcal{H}^{(2q-2)} \end{bmatrix} \tag{0.23}$$

to determine the coeffcients a_j of the characteristic polynomial $Q(s)$.

Recall that we can compute transfer function moments

$$\mathcal{H}^{(j)} = \boldsymbol{C}^T\mathbf{H}^j\mathbf{r}. \tag{0.14}$$

AWE method for SISO model reduction:[43]

AWE constructs an order-q Padé approximant about s_0

$$\mathcal{H}_q(s+s_0) = \frac{b_0 + b_1 s + b_2 s^2 + \cdots + b_p s^{q-1}}{1 + a_1 s + a_2 s^2 + \cdots + a_q s^q} \tag{0.17}$$

$$= x_\infty + \sum_{j=1}^{q'} \frac{x_j}{s-\mu_j}, \qquad q' \leq q \tag{0.18}$$

of the transfer function (0.5), from $2q$ moments of the original model.

Given $\mathbf{H}$ and $\mathbf{r}$ as defined in (0.11), we compute $2q-1$ terms of the sequence

$$\mathbf{r}, \mathbf{H}\mathbf{r}, \mathbf{H}^2\mathbf{r}, \ldots, \mathbf{H}^{2q-1}\mathbf{r}, \tag{0.24}$$

and left multiply $\boldsymbol{C}^T$ to obtain $2q$ system moments $\mathcal{H}^{(j)} = \boldsymbol{C}^T\mathbf{H}^j\mathbf{r}$, for $j = 0, 2, ..., 2q-1$. We then solve (0.23) for coefficients a_j of the denominator polynomial

$$1 + a_1 s + a_2 s^2 + \cdots + a_q s^q \quad = \quad (s-\mu_1)(s-\mu_2)\cdots(s-\mu_q),$$

and compute its roots μ_j, which are poles of the transfer function. We then solve another order-q system of equations for residues x_j and x_∞, thus yielding the pole-residue form (0.18).

AWE suffers from numerical trouble resulting from the fact that the power sequence (0.15) converges to an eigenvector of $\mathbf{H}$. The terms of the sequence quickly become linearly dependent so the systems to be solved (e.g. (0.23)) are badly conditioned. Thus, AWE is not practical for $n > 10$ or so. However, moment matching is one of the few ways to gage the quality of a reduced model , and although we will not explicitly compute transfer function moments, we try to develop methods with moment matching properties. The convergence of the sequence (0.24) to an eigenvector is the basis of the Power Iteration, addressed in §0.1.3.

Invariant subspaces

We first introduce the notion of an invariant subspace under an operator and extend it to that of a general matrix pencil.

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

$$\mathbf{H}\mathcal{Q} \subseteq \mathcal{Q}. \tag{0.25}$$

The span of a set of eigenvectors of $\mathbf{H}$ is an (**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

$$\mathbf{H}Q = QT \tag{0.26}$$

for some matrix $T \in \mathbb{C}^{\ell\times\ell}$. If the basis vectors are eigenvectors Z then (0.26) 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 **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, (0.26) 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.

Now we extend the notion of invariance to the linear matrix pencil $\boldsymbol{A} - s\boldsymbol{E}$. A *matrix pencil* is a polynomial

$$\sum s^j M_j$$

over $s \in \mathbb{C}$ with matrix coefficients $M_j \in \mathbb{C}^{N\times N}$. The *linear matrix pencil* $(\boldsymbol{A} - s\boldsymbol{E})$ is generally denoted as $(\boldsymbol{A}, \boldsymbol{E})$.

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 eigenvectors.

An eigenvalue μ of $(\boldsymbol{A}, \boldsymbol{E})$ has two associated eigenvectors: a left vector w and a right vector z, both in $\mathbb{C}^N$, such that

$$\boldsymbol{A}z = \mu\boldsymbol{E}z$$

and

$$w^H\boldsymbol{A} = \mu w^H\boldsymbol{E}, \quad \text{or equivalently } \boldsymbol{A}^H w = \mu\boldsymbol{E}^H w,$$

where $(\cdot)^H$ denotes the hermitian-conjugate transpose. It is common in linear algebra to speak of an eigen-*pair* (μ, z), or an eigen-triplet (μ, w, z).

A collection of ℓ right-eigenvectors as columns of $Z = \begin{bmatrix} z_1 & z_2 & \cdots & z_\ell \end{bmatrix}$ has the property that

$$\boldsymbol{A}Z = \boldsymbol{E}Z\mathcal{M}, \tag{0.27}$$

where $\mathcal{M} = \operatorname{diag}\begin{bmatrix}\mu_1 & \mu_2 & \cdots & \mu_\ell\end{bmatrix}$ is a diagonal matrix of the associated eigenvalues. Z is a basis for a subspace of the *right-eigenspace* of $(\boldsymbol{A}, \boldsymbol{E})$.

The notion of invariance under a general operator $\mathbf{H}$ extends to that of a matrix pencil. The subspace $\mathcal{Q} = \operatorname{span} Z$ is called invariant or *deflating* [53] with respect to $(\boldsymbol{A}, \boldsymbol{E})$ if

$$\dim(\boldsymbol{A}\mathcal{Q} + \boldsymbol{E}\mathcal{Q}) \leq \dim \mathcal{Q}. \tag{0.28}$$

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 [22] that $(\boldsymbol{A}, \boldsymbol{E})$-invariance is equivalent to $\mathbf{H}$-invariance for

$$\mathbf{H} = (\boldsymbol{A} - s_0\boldsymbol{E})^{-1}\boldsymbol{E}, \tag{0.29}$$

which happens to be the shift-invert operator defined by (0.11).

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

$$\lambda = \frac{1}{\mu - s_0}, \qquad \mu = s_0 + \frac{1}{\lambda} \tag{0.30}$$

and they share the same eigenvector. That is,

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

To see this, observe that

$$\begin{aligned} (\mu\boldsymbol{E} - \boldsymbol{A}) &= [(\mu - s_0)\boldsymbol{E} + (s_0\boldsymbol{E} - \boldsymbol{A})] \\ &= [(\mu - s_0)\underbrace{(s_0\boldsymbol{E} - \boldsymbol{A})^{-1}\boldsymbol{E}}_{-\mathbf{H}} + I] \\ &= [(\mu - s_0)\mathbf{H} - I] \\ &= \left[\mathbf{H} - \left(\frac{1}{\mu - s_0}\right) I\right] \\ &= (\mathbf{H} - \lambda I). \end{aligned} \tag{0.31}$$

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.

Pole-Residue representation

Recall the transfer function

$$\mathcal{H}(s) = \boldsymbol{C}^T (s\boldsymbol{E} - \boldsymbol{A})^{-1} \boldsymbol{B}, \tag{0.5}$$

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

The $\boldsymbol{A}$ and $\boldsymbol{E}$ that arise from Modified Nodal Analysis [27] 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/residues from 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[3] so that $W^H\boldsymbol{E}Z = I$. Then

$$\begin{aligned}
\mathcal{H}(s) &= \boldsymbol{C}^T(s\boldsymbol{E} - \boldsymbol{A})^{-1}\boldsymbol{B} \\
&= \boldsymbol{C}^T(W^{-H}(sI - \mathcal{M})Z^{-1})^{-1}\boldsymbol{B} \\
&= \boldsymbol{C}^T Z(sI - \mathcal{M})^{-1}W^H\boldsymbol{B} \\
&= \boldsymbol{C}^T Z\left(\sum_{j=1}^{q} \frac{1}{s - \mu_j}\right) W^H\boldsymbol{B}.
\end{aligned} \tag{0.32}$$

The pole (eigen) decomposition (0.32) of the transfer function suggests that it can be approximated if we have some eigen information about $(\boldsymbol{A}, \boldsymbol{E})$; poles $\mu_j \in \mathbb{C}$ located nearest to the segment $S \in i\mathbb{R}^+$ as illustrated in figure 1, because they cause $1/(s - \mu_j)$ to be large.

For the sake of expressing (0.32) 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 (0.32) 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 (0.5) as

$$\mathcal{H}(s) = \sum_{\mu_j=\infty} \hat{c}_j\hat{b}_j^T + \sum_{\mu_j\neq\infty} \frac{\hat{c}_j\hat{b}_j^T}{s - \mu_j}. \tag{0.33}$$

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 (0.33) generalizes to

$$\mathcal{H}(s) = \sum_{\mu_j=\infty} \xi_j\hat{c}_j\hat{b}_j^T + \sum_{\mu_j\neq\infty} \xi_j\frac{\hat{c}_j\hat{b}_j^T}{s - \mu_j}. \tag{0.34}$$

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

Poles and residues from shifted formulation For model reduction it is often favorable to work with the so-called s_0-shifted transfer function

$$\mathcal{H}(s) = \boldsymbol{C}^T \left(I - (s - s_0)\mathbf{H}\right)^{-1} \mathbf{R}, \tag{0.10}$$

rather than the standard formulation (0.5). For that we derive a sum that is term-by-term equivalent to (0.32) and (0.33) but involves the eigen-decomposition of $\mathbf{H}$, rather than $(\boldsymbol{A}, \boldsymbol{E})$.

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

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

for a $N \times N$ diagonal matrix $\Lambda = \left[\lambda_1, \lambda_2, \ldots, \lambda_N\right]$ of eigenvalues.

Then

$$\begin{aligned}\mathcal{H}(s) &= \boldsymbol{C}^T (I - (s - s_0)\mathbf{H})^{-1} \mathbf{R} \\ &= \boldsymbol{C}^T \left(Z(I - (s - s_0)\Lambda)Z^{-1}\right)^{-1} \mathbf{R} \\ &= (\boldsymbol{C}^T Z)\Delta(s)(Z^{-1}\mathbf{R}) \end{aligned} \tag{0.35}$$

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

$$\delta_j(s) = \begin{cases} \dfrac{s_0 - \mu_j}{s - \mu_j}, & \mu_j \neq \infty \\ 1, & \mu_j = \infty. \end{cases} \tag{0.36}$$

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

$$\mathcal{H}(s) = \sum_j \frac{\hat{f}_j \hat{g}_j^T}{1 - (s - s_0)\lambda_j} \tag{0.37}$$

$$\begin{aligned} &= \sum_{\lambda_j = 0} \hat{f}_j \hat{g}_j^T + \sum_{\lambda_j \neq 0} \frac{s_0 - \mu_j}{s - \mu_j} \hat{f}_j \hat{g}_j^T \\ &= \sum_j \delta_j(s) \hat{f}_j \hat{g}_j^T, \end{aligned} \tag{0.38}$$

where $\delta_j(s)$ is from (0.36). The transpose $\hat{g}_j^T$ is in fact a non-conjugate 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 $(\boldsymbol{A}, \boldsymbol{E})$).

Pole weight

Recall that poles of the transfer function $\mathcal{H}(s) = \boldsymbol{C}^T \left(s\boldsymbol{E} - \boldsymbol{A}\right)^{-1} \boldsymbol{B}$ 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 measure 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 (0.34) and (0.38) suggest a hierarchy of poles' importance for approximation, often called pole-dominance [1].

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.

From (0.38), over the interval $i[\omega_1, \omega_2]$ of interest on the $\Im$-axis we have

$$\|\mathcal{H}(i\omega)\|_\infty \leq \sum_j \|\delta_j(i\omega)\|_\infty \|\hat{f}_j\|_1 \|\hat{g}_j\|_1 , \tag{0.39}$$

which is a sum of positive numbers, each one associated with a pole μ_j. Then a relatively large pole-weight, or *pole-mass*

$$\gamma_j = \|\delta_j(i\omega)\|_\infty \|\hat{f}_j\|_1 \|\hat{g}_j\|_1 \tag{0.40}$$

from (0.39) indicates that μ_j is a 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 (0.5).

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 (0.39) 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

$$\begin{aligned} \|\delta_j(i\omega)\|_\infty &= \max_{\omega \in [\omega_1, \omega_2]} |\delta_j(i\omega)| \\ &= \begin{cases} \dfrac{|s_0 - \mu_j|}{\min\{|\mu_j - \omega_1|, |\mu_j - \omega_2|, |\Re(\mu_j)|\}}, & \mu_j \neq \infty \\ 1, & \mu_j = \infty \end{cases} \end{aligned} \tag{0.41}$$

over that interval. 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 8. 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 (0.39) (i.e. $\sum_j \gamma_j$), is the total mass of the system (0.1) with respect to $i[\omega_1, \omega_2]$, and can be used as a measure of ROM convergence. The combined weight of a few dominant poles often comprises most of a system's total mass.

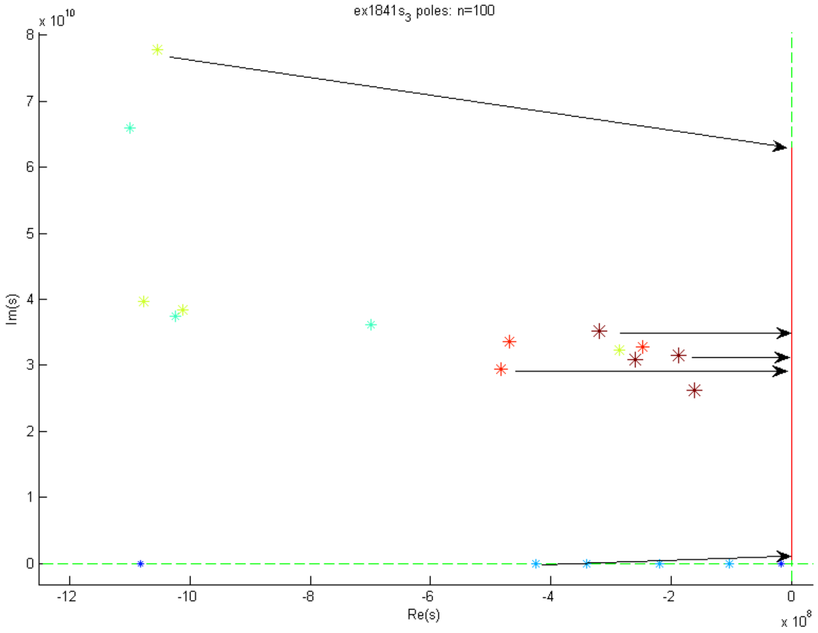


Figure 8: The red (solid) segment on the $\Im$-axis is the segment $i(\omega_0, \omega_1)$ of interest. In this case it looks like it extends to the origin but it does not. The arrows indicate how we define a pole's distance to the segment, which we use to determine the pole's weight. Conjugate pairs 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.

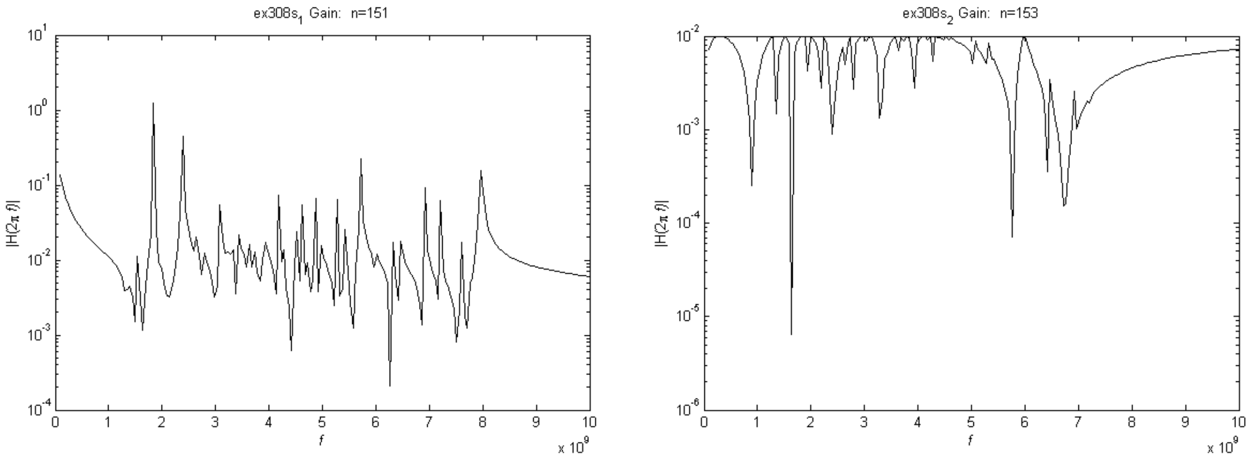
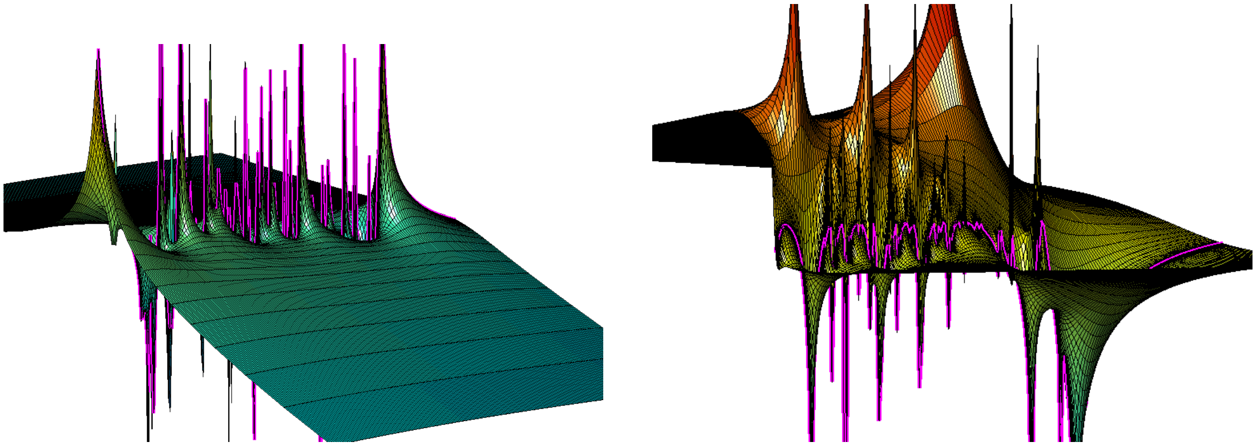


Figure 9: Frequency response gain plots for `ex308s11` and `ex308s22`, two SISO components of a single MIMO system. They share the same system pencil $(\boldsymbol{A}, \boldsymbol{E})$ and thus the same system eigenvalues. They differ in controllability and observability condition vector pairs: $(\boldsymbol{b}_1, \boldsymbol{c}_1)$ and $(\boldsymbol{b}_2, \boldsymbol{c}_2)$, respectively. Both systems have the same poles, but weighted differently by $\boldsymbol{b}_j$ and $\boldsymbol{c}_j$, giving two rather different frequency responses. The respective surface plots of $\mathcal{H}(s)$ below, with frequency response gain (over the $\Im-$axis) highlighted, show that the influence of poles and zeros is fairly uniform in the first example, and in the second we see a clear distinction between heavy and light poles.



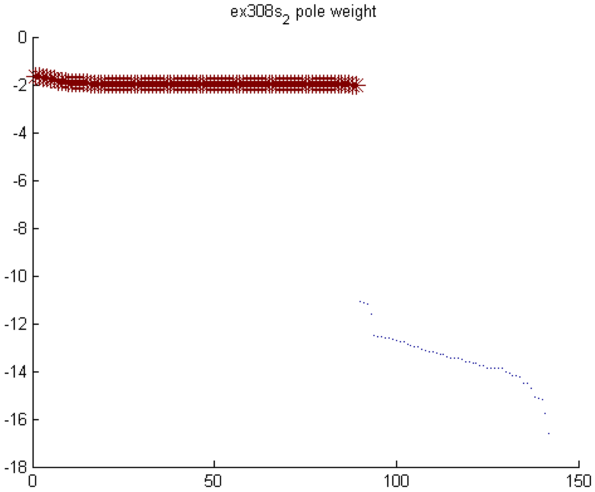
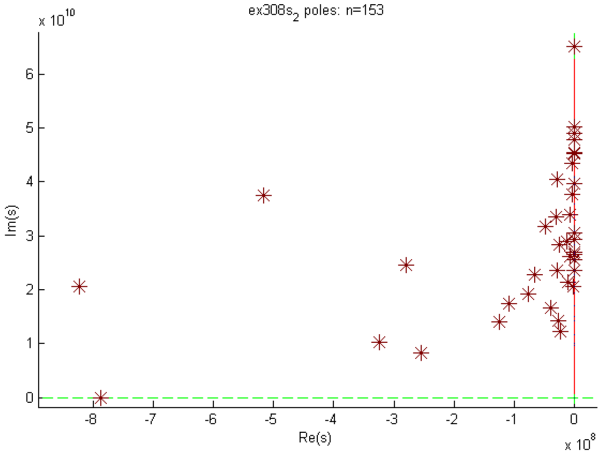


Figure 10: Pole weight order-of-magnitude distribution for `ex308s11` and `ex308s22`. In the second example, there is a clear separation between almost insignificant poles, and poles of nearly uniform influence. In the first example there is one dis-proportionally dominant pole (located at the origin) and the rest vary. Plots of the poles are below. Note that the locations the of poles are all the same, only their weights vary. Several appear to be on the $\Im$-axis, but they are only close, meaning within 10^7 of the axis in this scaling.



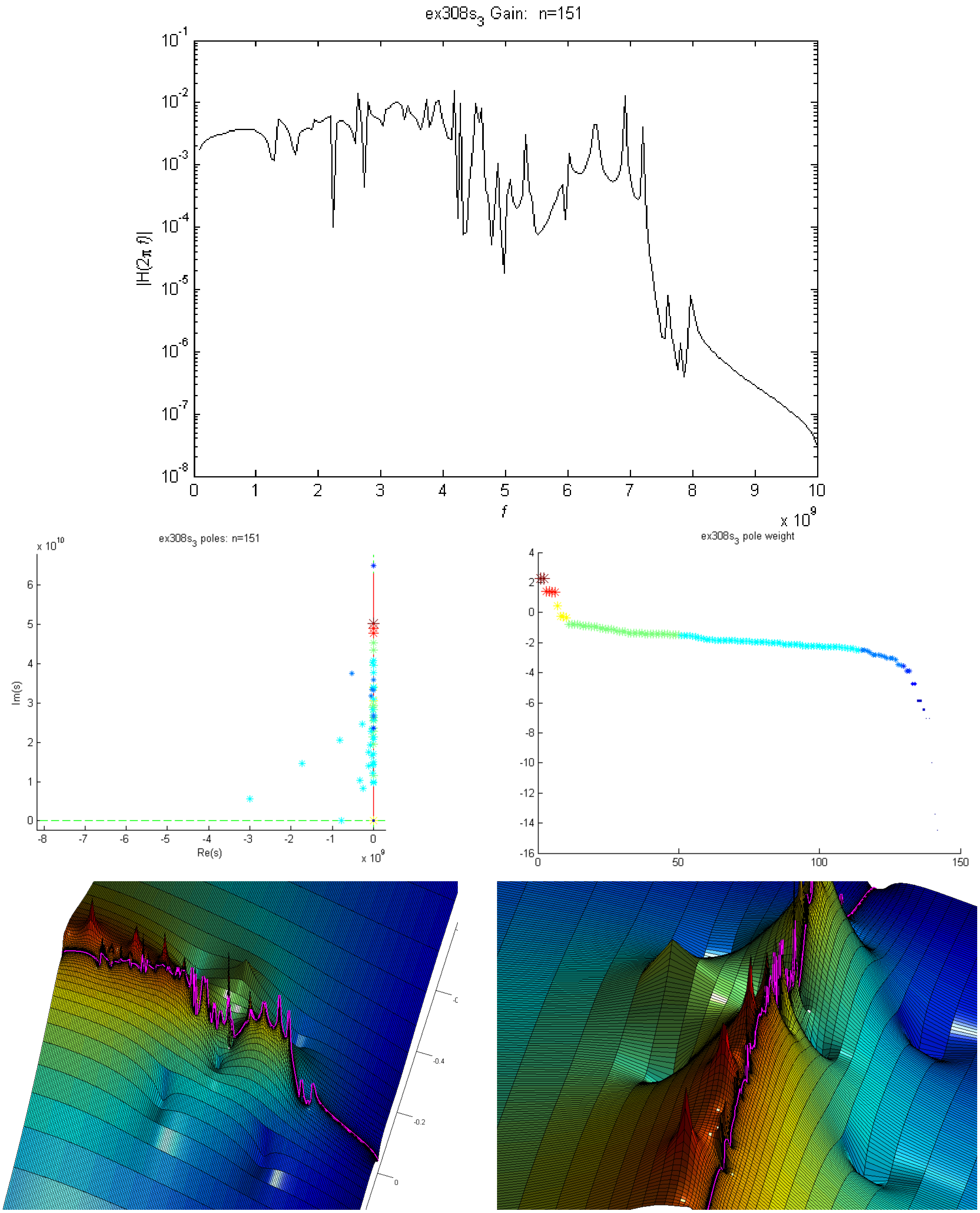


Figure 11: Here are similar plots for a third SISO model `ex308s12` from the same family as those in figure 9. The three `ex308` SISO models `s11`, `s22`, and `s12` are from one MIMO circuit model with $\boldsymbol{B} = \boldsymbol{C} \in \mathbb{R}^{308\times 2}$ ($m = 2$ inputs, $p = 2$ outputs). This example is more like `s11` than `s22` in that the pole weights are relatively uniform. In `s12`, more poles are dominant.

ROM transfer function via projection

The URM (unreduced model) transfer function formulations (0.5) and (0.10) exist only in theory for large applications, which can be on the order of 10^9 at the time of this writing. (0.5) and (0.10) are mathematically equivalent. Subspace projected ROM transfer functions can be obtained via *explicit* projection, or *implicit* projection, yielding two forms similar to (0.5) and (0.10) that are *not* mathematically equivalent but they converge to the URM (unreduced) transfer function (0.5), (0.10) as the projection subspace approaches $\mathbb{C}^N$.

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

$$\boldsymbol{A}_n := V^T\boldsymbol{A}V, \quad \boldsymbol{E}_n := V^T\boldsymbol{E}V, \quad \boldsymbol{C}_n := V^T\boldsymbol{C}, \quad \boldsymbol{B}_n := V^T\boldsymbol{B}, \tag{0.4}$$

of realization $(\boldsymbol{A}, \boldsymbol{E}, \boldsymbol{B}, \boldsymbol{C})$ on to $\mathcal{K}$,[4] the **explicitly projected** model $(\boldsymbol{A}_n, \boldsymbol{E}_n, \boldsymbol{B}_n, \boldsymbol{C}_n)$ has transfer function

$$\widehat{\mathcal{H}}_n(s) = \boldsymbol{C}_n^T\left(s\boldsymbol{E}_n - \boldsymbol{A}_n\right)^{-1}\boldsymbol{B}_n. \tag{0.42}$$

The explicitly projected model has the nice property that the projected ROM (0.42) of a stable system is stable if the projection basis V is real-valued. No poles of the ROM transfer function have positive real part, with the possible exception of remote approximations to poles of (0.5) at ∞, and even this can most-likely be attributed to numerical error. The classic explicit projection ROM method is PRIMA, given by [35]. In that case, the subspace for projection is a Krylov subspace and the ROM is shown to be of Padé-type, matching n moments with a model of dimension n.

More generally we may construct a pair of bi-orthogonal bases V and W (with $V^TW = I$), and explicitly make the oblique projections

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

Most projected methods do have one and two-basis variants; they are respectively called *orthogonal* and *oblique* projection methods.

Some iterative subspace methods, notably Krylov subspace methods, use the operator $\mathbf{H} = \mathbf{H}(s_0)$ simultaneously construct bases $V$and W *and* a projected operator matrix $\widetilde{\mathbf{H}} \in \mathbb{C}^{n\times n}$, where

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

This permits what we will call the **implicitly projected** ROM transfer function

$$\widetilde{\mathcal{H}}_n(s) = \boldsymbol{C}_n^H\left(I - (s - s_0)\widetilde{\mathbf{H}}\right)^{-1}\widetilde{\boldsymbol{\rho}}_n, \tag{0.43}$$

where

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

Clearly this is the projected ROM analog to the “shifted” transfer function (0.10). The implicitly projected type of MOR transfer function was made popular by [16] (and independently by [20]) where it was shown that such a ROM is Padé, hence *Padé Via Lanczos*. The insight of [16, 20] to apply a projected process for moment matching was provided by [23] in 1974. PVL and its predecessor AWE are mathematically equivalent but the Krylov subspace projected variant PVL works much better in practice. Instead of models matching a handful of moments, any number of moments can be matched by PVL, or all of them. A variant of PVL using the one-sided Arnoldi process and using one orthogonal basis $V = W$ to make (0.43) was shown to preserve passivity for RCL circuit models by [51].

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

We call (0.43) the transfer function for the implicitly projected model because it exists without a projected realization $(\boldsymbol{A}_n, \boldsymbol{E}_n, \boldsymbol{B}_n, \boldsymbol{C}_n)$ from (0.4). The reduced model implied by (0.43) is actually

$$\begin{aligned} \widetilde{\mathbf{H}}\frac{d\tilde{x}}{dt} &= (I + s_0\widetilde{\mathbf{H}})\tilde{x} + \widetilde{\boldsymbol{\rho}}_n u \\ \hat{y} &= \boldsymbol{C}_n^T \tilde{x}, \end{aligned} \tag{0.44}$$

which is not equivalent to the explicitly projected system

$$\begin{aligned} \boldsymbol{E}_n\frac{dz}{dt} &= \boldsymbol{A}_n z + \boldsymbol{B}_n u \\ \hat{y} &= \boldsymbol{C}_n^T z, \end{aligned} \tag{0.3}$$

given the same basis V, unless V spans $\mathbb{C}^N$ (i.e. $n = N$) in which case they are both equivalent to the original system (0.1).

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 [51, 26, 28] 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 (0.4).

0.1.2 Stability and Passivity of the model

Stability and passivity are two important properties that if present in the model (0.1), should be preserved. A time domain solution of a SISO model (0.1) takes the form

$$y(t) = \alpha_0 + \sum_{j=1}^{N} \alpha_j e^{\mu_j t},$$

where the modes $e^{\mu_j t}$ are either transient ($\Re(\mu) < 0$), strictly oscillatory ($\Re(\mu) = 0$), or unstable ($\Re(\mu) > 0$). Values $\mu_j \in \mathbb{C}$ are the poles of the system transfer function $\mathcal{H}(s) = \boldsymbol{C}^T \left(s\boldsymbol{E} - \boldsymbol{A}\right)^{-1} \boldsymbol{B}$, which are eigenvalues of $(\boldsymbol{A}, \boldsymbol{E})$. A system is **stable** if solutions $y(t)$ do not blow up for $t \geq 0$, which means for every $\alpha_j e^{\mu_j t}$, either $\Re(\mu_j) \leq 0$ or $\alpha_j = 0$. We can state this in terms of properties of the transfer function: (0.1) is stable if and only if $\Re(\mu_j) \leq 0$, and any pole on the $\Im$-axis ($\Re(\mu_j) = 0$) is simple.

Passivity of the model is not as simple to describe as stability, in general. For circuit models (0.1), a passive circuit is one that has no power sources, i.e. it does not generate energy. In that case (0.1) is passive if it has the same number of inputs as outputs ($\dim \boldsymbol{B} = \dim \boldsymbol{C}$), and its transfer function (0.5) is *positive-real*, which means

- $\mathcal{H}(s)$ is analytic (infinitely differentiable) on $\mathbb{C}^+ = \{\Re(s) > 0\}$.
- $\mathcal{H}(\overline{s}) = \overline{\mathcal{H}(s)}$ for $s \in \mathbb{C}$.
- $\mathcal{H}(s) + [\mathcal{H}(s)]^H$ is Hermitian positive semi-definite. Equivalently, $\Re(x^H \mathcal{H}(s) x) \geq 0$ for all $s \in \mathbb{C}$ and $x \in \mathbb{C}^N$.

If the ROM is already constructed, [5] provides a way to determine its passivity via the transfer function, but it is computationally cumbersome. We would like to guarantee that the ROM (0.3) is passive by construction.

With explicit projection methods, having a real projection basis $V \in \mathbb{R}^{N \times n}$ is possibly the most important thing we can do to preserve stability and/or passivity. A basic assumption about the descriptor system (0.1) is that the realization $(\boldsymbol{A}, \boldsymbol{E}, \boldsymbol{B}, \boldsymbol{C})$ consists of real matrices, which is also true of the projected ROM (0.3) if the projection basis V is real. This is trivial if the expansion point s_0 is real, but using a non-strictly real expansion point $s_0 \in \mathbb{C}$ results in a complex basis $V \in \mathbb{C}^{N \times n}$.

A standard way to skirt the issue is to assume s_0 is real; otherwise, we assume our real basis V is the result of splitting real and imaginary parts of the complex basis $\widetilde{V} \in \mathbb{C}^{N \times n}$ so that

$$\operatorname{span} V = \operatorname{span} \begin{bmatrix} \widetilde{v}_1^{\mathbf{r}} & \widetilde{v}_1^{\mathbf{i}} & \widetilde{v}_2^{\mathbf{r}} & \widetilde{v}_2^{\mathbf{i}} & \cdots & \widetilde{v}_n^{\mathbf{r}} & \widetilde{v}_n^{\mathbf{i}} \end{bmatrix},$$

where $\widetilde{v}_j^{\mathbf{r}} = \Re(\widetilde{v}_j)$ and $\widetilde{v}_j^{\mathbf{i}} = \Im(\widetilde{v}_j)$. Then $V \in \mathbb{R}^{N \times \ell}$ for $n < \ell \leq 2n$, which is (up to) twice as large as the basis produced with a real value for s_0. Furthermore, the process of splitting the basis into $\Re$ and $\Im$ parts and making it orthogonal requires a significant additional amount of computation, resulting a distasteful post-processing step that many authors would rather not talk about because there is currently no elegant and efficient solution. We discuss one very easy to implement improvement that we call "lazy" orthogonalization in §0.1.5.

0.1.3 Iterative methods

In order to address the means by which we construct projection bases, we first introduce the underlying iterative methods that they come from. Recall the matrices

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

from the definition (0.11) of the transfer function, which are responsible for Krylov methods' moment matching properties. It may be helpful to be aware that $\mathbf{H} \in \mathbb{C}^{N\times N}$ is very large (because N is very large!), dense in general, and never explicitly formed. The only significant thing we can hope to do with $\mathbf{H}$ is multiply it by a vector, which is only possible because the large matrices $\boldsymbol{A}$ and $\boldsymbol{E}$ that comprise it are *sparse* (contain mostly zeros). It is important for the reader to be aware of the scale involved. Imagine such an enormous matrix $\mathbf{H}$ that one matrix-vector product takes several hours, or days to compute. The classic example of very large scale matrix multiplication is that of the Google matrix, which represents links between every site on the world wide web. It's largest eigenvalue is used to calculate page-rank, described in [37]. There are several model reduction methods available, some of them quite mathematically elegant. However, many require manipulation of $(\boldsymbol{A}, \boldsymbol{E}, \boldsymbol{B}, \boldsymbol{C})$ that, in our case, would result in very large, dense (non-sparse) matrices that we have neither memory nor computational power to handle. Iterative methods are ideal for this scenario because the only large matrix operation they require is a matrix-vector product, which is assumed to be the computational bottleneck. With that in mind we proceed with a discussion of matrix iterations that motivates Krylov subspace model reduction methods.

The (Block) Krylov sequence

$$\mathbf{R}, \quad \mathbf{HR}, \quad \mathbf{H}^2\mathbf{R}, \quad \ldots \tag{0.45}$$

is the central structure of (block) Krylov subspace methods and it has some interesting features. For one, it generally converges to the invariant space of $(\boldsymbol{A}, \boldsymbol{E})$ associated with eigenvalues closest to s_0. Thus, it encodes poles of the transfer function. It is also what allows us to create a Padé type model (match Taylor series coefficients), since these coefficients (the so-called moments about s_0) can be defined involving powers of $\mathbf{H}$ multiplied with $\mathbf{R}$. The Krylov sequence and the subspaces that its span defines are used in a lot of applications that involve operations with the very large matrix $\mathbf{H}$. Each term in the sequence requires one matrix-vector product to compute, and provides increasing information about $\mathbf{H}$, up to $\mathbf{H}^d\mathbf{R}$, where $d \leq N-1$ is called the grade of $\mathbf{H}$ with respect to $\mathbf{R}$. Any $\mathbf{H}^j\mathbf{R}$ for $j \geq d$ no longer provides additional information about $\mathbf{H}$. Depending on how much computation we are willing to carry out, we can extract only as good an approximation of $\mathbf{H}$ as we need or can afford.

In the previous section we saw that the j-th moment of the Taylor series about s_0 of the transfer function

$$\mathcal{H}(s) = \boldsymbol{C}^T\left(s\boldsymbol{E} - \boldsymbol{A}\right)^{-1}\boldsymbol{B} \tag{0.5}$$

is given by

$$\mathcal{H}^{(j)}(s_0) = \boldsymbol{C}^T\mathbf{H}^j\mathbf{R},$$

where the moment $\mathcal{H}^{(j)}(s_0)$. The equivalent formulation $\boldsymbol{C}^T\mathbf{H}^jR$ from (0.13) is more suited to computation because it involves only successive multiplications of static matrices $\mathbf{H}$ and $\mathbf{R}$, and in fact a widely used MOR method of the early 90s [43] did exactly that. The AWE method for MOR explicitly computes several terms of (0.45), then left-multiplied $\boldsymbol{C}^T$ to obtain the moments.

The methods involving $\mathbf{H}$ that we will explore require only that we have a way to perform the multiplication $\mathbf{H}v$ for several vectors v. This can done efficiently by an initial, sparse LU-factorization

$$LU = P(\boldsymbol{A} - s_0\boldsymbol{E})Q. \tag{0.46}$$

Then the multiplication routine computes

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

as the product $\boldsymbol{B}Pv$ and two triangular back-solves for each matrix-vector product $\mathbf{H}v$. If we choose to use another expansion point s_0, we have to re-factor (0.46) which is generally to be avoided if the system is extremely large. Computation can be reduced by observing that the shift s_0 does not affect sparsity of the matrix pencil $(\boldsymbol{A} - s_0\boldsymbol{E})$. Then permutations P and Q only need to be determined once and can remain the same for further LU factorizations. It should be noted that Matlab does not do this by default.

Note that "multiplication" by $\mathbf{H}$ is actually a solve. The operator $\mathbf{H}(s_0)$ is sometimes called a *shifted-inverse* operator, with *shift* s_0. This may cause confusion since in the literature, *power* iterations with a matrix M are often treated separately from *inverse* or *rational* iterations with $(M - \sigma I)^{-1}$. The reader should be aware that we speak of iterations as multiplication by the matrix $\mathbf{H}$, but the underlying operation is a shifted inverse iteration with $\mathbf{H} = (\boldsymbol{A} - s_0\boldsymbol{E})^{-1}\boldsymbol{E}$. As such, the Krylov methods we are concerned with are called *Rational-Krylov* methods by some authors, due to the shifted "rational" iterations with variable shift s_0.

Power iteration

Let us motivate this discussion by supposing that v is an eigenvector of $\mathbf{H}$, with eigenvalue λ. Then

$$\mathbf{H}v = \lambda v.$$

Now suppose that $\mathbf{r} = v + \mathbf{r}_0$, where $\mathbf{r}_0$ has no component in the direction of v. Then

$$\begin{aligned}\mathbf{H}^j\mathbf{r} &= \mathbf{H}^j v + \mathbf{H}^j\mathbf{r}_0\\ &= \lambda^j v + \mathbf{H}^j\mathbf{r}_0\\ &\approx \lambda^j v\end{aligned}$$

for large j, if $|\lambda|$ is large compared with other eigenvalues of $\mathbf{H}$, because the term invloving λ_j dominates the sum. This is the idea behind the power iteration. If we multiply $\mathbf{H}$ by $\mathbf{r}$ enough times, the result will essentially be a multiple of v.

The power iteration is perhaps the most basic matrix iteration. Given a start vector $\mathbf{r} \in \mathbb{C}^N$, terms v_k in power iteration sequence are given by

$$v_0 = \mathbf{r}, \qquad \text{and} \qquad v_{j+1} = \frac{\mathbf{H}v_j}{\|\mathbf{H}v_j\|}, \tag{0.47}$$

where $\|\cdot\|$ is a scaling factor to prevent the iterate from growing too large or small. Any norm will do. Suppose λ_1 is the eigenvalue of $\mathbf{H}$ with largest magnitude and z_1 is the associated eigenvector. As long as start vector $\mathbf{r}$ has a nonzero component in the direction of z_1, the sequence $\{v_j\}$ will converge to z_1.

This is because if we express $\mathbf{r}$ in terms of the eigenvectors z_j, of $\mathbf{H}$ (for $j = 1, 2, ..., q$),

$$\mathbf{r} = \alpha_1 z_1 + \alpha_2 z_2 + \cdots + \alpha_q z_q, \tag{0.48}$$

$$\tag{0.49}$$

then n applications of $\mathbf{H}$ yield

$$\begin{aligned}\mathbf{H}^n\mathbf{r} &= \alpha_1\lambda_1^n z_1 + \alpha_2\lambda_2^n z_2 + \cdots + \alpha_q\lambda_q^n z_q\\ &= \lambda_1^n\left[\alpha_1 z_1 + \left(\frac{\lambda_2}{\lambda_1}\right)^n \alpha_2 z_2 + \cdots + \left(\frac{\lambda_q}{\lambda_1}\right)^n \alpha_q z_q\right]\end{aligned}$$

where eigenvalues λ_j of $\mathbf{H}$ are smaller in magnitude than λ_1. Each successive multiplication by $\mathbf{H}$ results in a linear combination that has a smaller contribution from eigendirections other than z_1. As long as $\alpha_1 \neq 0$,

$$\frac{\mathbf{H}^n\mathbf{r}}{\|\mathbf{H}^n\mathbf{r}\|_2} \to z_1 \quad \text{as} \quad n \to \infty.$$

- Equation (0.49) is only valid if $\mathbf{H}$ is diagonalizeable. The case for general $\mathbf{H}$ is analogous to that presented above, except we consider the more cumbersome Jordan (or *spectral*) decomposition of $\mathbf{H}$, which always exists. If any square matrix $\mathbf{H}$ has q distinct eigenvalues λ_1 then the spectral decomposition of $\mathbf{H}$ is

  $$\mathbf{H} = \sum_{j=1}^{q} (\lambda_j P_j + D_j) \tag{0.50}$$

  where P_j is the spectral projector associated with λ_j, so that $\mathbf{H}P_j = \lambda_j P_j$. We express r in terms of spectral operators of $\mathbf{H}$

  $$\mathbf{r} = \sum_{j=1}^{q} P_j \mathbf{r},$$

  The operator $D_j = (\mathbf{H} - \lambda_j I)P_j$ is the nilpotent operator associated with λ_j.

  This is covered in depth in any respectable text covering the topic of eigendecomposition, and a very good one is [47], chapter 4. Unless absolutely necessary however, we shall continue to assume that $\mathbf{H}$ is diagonalizable, and due to finite precision arithmetic this will generally be true in practice.

The power method converges linearly with n, meaning an eigenvalue estimate $\lambda^{(n)}$ from the n-th iteration of the power method satisfies

$$|\lambda^{(n)} - \lambda_1| = \mathcal{O}\left(\left| \frac{\lambda_2}{\lambda_1} \right|^n \right)$$

where λ_2 is the second-largest eigenvalue of $\mathbf{H}$.

We have seen that a power iteration with $\mathbf{H}$ generally converges to the eigenvector associated with its largest eigenvalue (in magnitude). In our specific case,

$$\mathbf{H} := (\boldsymbol{A} - s_0\boldsymbol{E})^{-1}\boldsymbol{E}, \qquad \mathbf{r} = (s_0\boldsymbol{E} - \boldsymbol{A})^{-1}\boldsymbol{b} \tag{0.11}$$

from our transfer function definition (0.10), $\mathbf{H}$ involves an inverse. $\mathbf{H}$ is a shifted-inverse operator, with shift s_0. Then it is more fitting to call this a *shifted-inverse iteration*, or more simply an *inverse-iteration*. Inverse iteration [42], developed in 1920, was once the most widely used method of finding eigenvalues of large matrices. It is well-studied and is the basis for the Arnoldi iteration. A modification of the inverse iteration where shift s_0 varies is called the *Rayleigh quotient iteration*. Iterative methods that utilize a shifted-inverse operator and allow for a variable shift are sometimes called *rational iterations*, and the Krylov methods they imply are sometimes called *Rational-Krylov* methods, a name coined by Ruhe in [45]. For now we will focus on iterations with a constant shift.

The idea behind the shifted inverse iteration is that we have some control over which eigenvalues converge first. The basic power iteration with a matrix M converges to the eigenvalue of M with greatest magnitude. Then the inverse iteration (with M^{-1}) converges to the *smallest* eigenvalue of M. The shifted-inverse iteration (with $(M - \sigma I)^{-1}$ by means of a solve) converges to the eigenvalue of M closest to σ. Likewise, iterations with our operator $\mathbf{H} = (\boldsymbol{A} - s_0\boldsymbol{E})^{-1}\boldsymbol{E}$ converge to the eigenvalue of $(\boldsymbol{A}, \boldsymbol{E})$ closest to s_0.

Our iteration $v_{n+1} = k_n \mathbf{H} v_n$ is equivalent to solving

$$(\boldsymbol{A} - s_0\boldsymbol{E})v_n = k_n \boldsymbol{E} v_{n-1}. \tag{0.51}$$

for v_n, where k_n is a scaling factor chosen so that $\|v_n\| = 1$.

Suppose $(\boldsymbol{A}, \boldsymbol{E})$ has q distinct eigenvalues μ_j with associated vectors z_j. Given start vector

$$v_0 = r = \sum_{j=1}^{q} \alpha_j z_j,$$

n-iterations with $\mathbf{H}$ yields

$$v_n = (k_1 k_2 \cdots k_n) \sum_{j=1}^{n} \frac{\alpha_j}{(\mu_j - s_0)^n} z_j. \tag{0.52}$$

The inverse-iteration with $(\boldsymbol{A} - s_0\boldsymbol{E})$ on start vector $\boldsymbol{E}\mathbf{r}$, (aka power iteration with $\mathbf{H}$) generally converges to the eigenvector z_j associated with μ_j closest to s_0.

Recall that an eigenvalue λ of $\mathbf{H}$ is related to an eigenvalue μ of $(\boldsymbol{A}, \boldsymbol{E})$ by

$$\lambda = \frac{1}{\mu - s_0},$$

with identical eigenvectors. Then the eigenvalue λ_1 of largest magnitude of $\mathbf{H}$ corresponds with the μ_1 nearest s_0. Thus, the closer s_0 is to a pole of the transfer function, theoretically the faster it will converge to the vector associated with that pole. This is related to our observation in §0.1.1 that the region of convergence for the Taylor series about s_0 is the largest disk centered at s_0 that does not contain a pole. The power iteration converges to one vector and ceases to provide any information about other eigenvectors. This is good if we seek only one eigenvalue or vector, but problematic if we want more general spectral information about $\mathbf{H}$.

There is one complication that we should address regarding complex eigenvalues. If $\mathbf{H}$ and $\mathbf{r}$ are real, then the power iteration obviously will not converge to a complex eigenvector. It will converge to the largest real eigenvalue of $\mathbf{H}$, or diverge. The iterates are still useful and provide spectral information as we will see in §0.1.4. If the eigenvalue(s) of largest magnitude is a complex conjugate pair, even with a complex start vector the power iteration may or may not converge. It can infinitely cycle between the two conjugate eigenvectors.

What if our shift s_0 is very close to being an exact eigenvalue of $(\boldsymbol{A}, \boldsymbol{E})$? Then our shifted-inverse operator $(\boldsymbol{A} - s_0\boldsymbol{E})^{-1}\boldsymbol{E}$ is badly conditioned and appears to be impossible or unreliable. Apparently this is not a problem according to [42], in part because the "ill-conditioned" solve (0.51) in the inverse iteration involves a right hand side that is nearly in the range of $(\boldsymbol{A} - s_0\boldsymbol{E})$.

If there are several eigenvalues μ_j of $(\boldsymbol{A}, \boldsymbol{E})$ around the same distance from s_0 or the closest μ_j to s_0 is a multiple eigenvalue, then there may be trouble with convergence. This is a drawback of the naive power (and shifted-inverse) iteration.

Deflation of converged eigenvalues

Deflation is an essential feature of any iterative eigenvalue method that finds more than one eigenvalue. Most eigenvalues of $\mathbf{H}$ are poles of the system transfer function and their convergence is closely related to convergence of the ROM. One important difference between MOR and the general eigenvalue problem is that we are not as free to modify $\mathbf{H} = \mathbf{H}(s_0)$ or $\mathbf{R} = \mathbf{R}(s_0)$ from (0.11), since the moment matching property of the method is defined using $\mathbf{H}$ and $\mathbf{R}$. The closest thing we can do in this regard is choose another s_0' and compute another $\mathbf{H}(s_0')$ and $\mathbf{R}(s_0')$, and continue iteration with the new pair. Any eigenvectors that converge are eigenvectors of $\mathbf{H}$ for any shift s_0, as was illustrated by (0.31). Changing the shift s_0 requires the re-factorization $LU = (s_0'\boldsymbol{E} - \boldsymbol{A})$, which is generally avoided. Ultimately we will exploit that possibility, but for this section we assume $\mathbf{H}$ and $\mathbf{R}$ are fixed.

The power method (0.47) or equivalently, shifted-inverse iteration (0.51), finds one eigenvalue of $\mathbf{H}$, namely that with the largest magnitude. When further iterations of (0.47) no longer differ significantly, we say the process has converged. Suppose the power iterates with $\mathbf{H}$ and r converge to z_1, with $\|z_1\| = 1$. The associated eigenvalue λ_1 of $\mathbf{H}$ is given by the Rayleigh-quotient

$$\lambda_1 = z_1^H \mathbf{H} z_1.$$

which we will assume is in fact the largest eigenvalue of $\mathbf{H}$. Then the eigenvalue

$$\mu_1 = z_1^H (s_0\boldsymbol{E} - \boldsymbol{A}) z_1$$

of $(\boldsymbol{A}, \boldsymbol{E})$ associated with z_1 is the closest eigenvalue to s_0. In order to retrieve another eigen-pair (λ_2, z_2) of $\mathbf{H}$, we need to somehow remove λ_1 from the spectrum of $\mathbf{H}$. For this we perform a *deflation* of $\mathbf{H}$, which can be done a few ways. *Wielandt* deflation theoretically replaces $\mathbf{H}$ with another matrix $\mathbf{H}_1$ that has the same spectrum except for λ_1. The rank-1 modification

$$\mathbf{H}_1 = \mathbf{H} - \lambda_1 z_1 z_1^H$$

has the property that $\mathbf{H}z_1 = 0$, but otherwise H_1 has the same eigenvalues as $\mathbf{H}$. For the generally non-Hermitian $\mathbf{H}$ this particular deflation does not preserve right eigenvectors of $\mathbf{H}$ since

$$\begin{aligned}\mathbf{H}_1 z_j &= \mathbf{H}z_j - \lambda_1 z_1 z_1^H z_j \\ &= \lambda_j z_j - \gamma_j z_1 \\ &\neq \lambda_j z_j\end{aligned}$$

for $j \neq 1$ (it does preserve the left ones), but it preserves Schur vectors.

- For an ordered collection
$$(\lambda_1, \lambda_2, \ldots, \lambda_\ell)$$
of eigenvalues of $\mathbf{H}$ with $\ell \leq q$ (meaning not necessarily all of them), the **Schur vectors**
$$(u_1 = z_1, u_2, \ldots, u_\ell)$$
associated with that ordering form the orthonormal basis of the subspace spanned by $(z_1, z_2, \ldots, z_\ell)$ so that
$$\mathbf{H}Q_\ell = Q_\ell R_\ell$$
where $Q_\ell = \begin{bmatrix} u_1 & u_2 & \cdots & u_\ell \end{bmatrix}$ is orthogonal and R_ℓ[5] is upper triangular and has the eigenvalues $(\lambda_1, \lambda_2, \ldots, \lambda_\ell)$, in that order, along its diagonal.

The deflated matrix $\mathbf{H}_1$ has the same Schur vectors as $\mathbf{H}$, but associated with the ordered set of eigenvalues $(0, \lambda_2, \ldots, \lambda_q)$. If the Schur decomposition of $\mathbf{H}$ is $\mathbf{H}Q = QR$, then

$$\mathbf{H}_1 Q \quad = \quad (\mathbf{H} - \lambda_1 u_1 u_1^H)Q \quad = \quad QR - \lambda_1 u_1 e_1^T \quad = \quad U(R - \lambda_1 e_1 e_1^T).$$

Power iterations with $\mathbf{H}_1 = (\mathbf{H} - \lambda_1 u_1 u_1^H)$ and start vector $\mathbf{r}$ *should* converge to the Schur vector u_2 associated with λ_2, the next largest eigenvalue of $\mathbf{H}$.

- Whether or not it is in fact the vector associated with the second largest eigenvalue, the converged eigenvector u_2 of $\mathbf{H}_1$ is the second Schur vector of $\mathbf{H}$, associated with (approximate) eigenvalue
$$\tilde{\lambda_2} = u_2^H \mathbf{H}_1 u_2 \approx v_n^H v_{n-1}.$$

We can continue to "deflate" $\mathbf{H}$ into

$$\mathbf{H}_\ell = \mathbf{H} - \lambda_1 u_1 u_1^H - \lambda_2 u_2 u_2^H - \cdots - \lambda_\ell u_\ell u_\ell^H. \tag{0.53}$$

An iteration with the deflated matrix (0.53) is equivalent to an iteration with $\mathbf{H}$ followed by subtracting off $\mathbf{H}$-invariant components to suppress influence from those already converged directions.

$$\begin{aligned} v'_{n+1} &= \mathbf{H}v_n \\ \alpha_{n+1} v_{n+1} &= v'_{n+1} - \sum_{j=1}^{\ell} (\lambda_j u_j^H v'_{n+1}) u_j \\ &= v'_{n+1} - \left(Q_\ell \Lambda_\ell Q_\ell^H\right) v'_{n+1}. \end{aligned} \tag{0.54}$$

[5] R_ℓ has no relation to our "start block" $\mathbf{R}$ defined in (0.11).

Deflating a general subspace

The Wielandt deflation shown in the previous section is a means of "purging" converged eigenvectors from future iterations of the power method. (0.54) shows that we can do this with any partial eigenspace of $\mathbf{H}$ too, so as eigenvectors converge we can add them to the set of eigenvectors that are "locked" by the process, or "purged" from further iterations.

Suppose instead that we want to deflate an arbitrary subspace $\mathcal{Q}$, with orthonormal basis

$$Q = \begin{bmatrix} u_1 & u_2 & \cdots & u_\ell \end{bmatrix}.$$

Most likely, $\mathcal{Q}$ is an invariant (Schur) space associated with converged eigenvalues, but $\mathcal{Q}$ can be any subspace that we want to remove from the process. An important detail is that in this case we have a basis for $\mathcal{Q}$, but no eigenvalues.

The $\mathcal{Q}$-deflated power iteration is

$$\begin{aligned} \alpha_{n+1} v_{n+1} &= \mathbf{H} v_n - \sum_{j=1}^{\ell} (u_j^H \mathbf{H} v_n) u_j \\ &= \mathbf{H} v_n - \left(QQ^H\right) \mathbf{H} v_n \\ &= \left(I - QQ^H\right) \mathbf{H} v_n. \end{aligned} \tag{0.55}$$

Every new iteration is made orthogonal to $\mathcal{Q}$, thus "deflating" $\mathcal{Q}$ out of further iterates, so that we have an orthogonal set $\{ u_1, u_2, \cdots, u_\ell, v_{n+1} \}$ where

$$\operatorname{span} \begin{bmatrix} u_1 & u_2 & \cdots & u_\ell & v_{n+1} \end{bmatrix} = \operatorname{span} \begin{bmatrix} u_1 & u_2 & \cdots & H v_n \end{bmatrix}.$$

In the special (and typical) case that $\mathcal{Q}$ is an invariant subspace of $\mathbf{H}$, this deflation effectively removes the associated eigenvalues from the spectrum of $\mathbf{H}$. The deflated iteration (0.55) converges to a Schur vector associated with the largest eigenvalue of $\mathbf{H}$ that is not part of the deflated set.

Deflation is a recurring theme in eigenvalue methods. Every Krylov method is a sort of deflated power iteration. Even after producing a basis V on which to project the model, we can choose to deflate eigenvectors associated with insignificant or unwanted poles of the transfer function.

0.1.4 Krylov subspace projection methods

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

$$\begin{aligned}\mathcal{H}(s) &= \boldsymbol{C}^T \left(s\boldsymbol{E} - \boldsymbol{A}\right)^{-1} \boldsymbol{B} && (0.5)\\ &= \boldsymbol{C}^T \left(I - (s - s_0)\mathbf{H}\right)^{-1} \mathbf{R} && (0.10)\end{aligned}$$

with

$$\mathbf{H} = (\boldsymbol{A} - s_0\boldsymbol{E})^{-1}\boldsymbol{E} \quad \text{and} \quad \mathbf{R} = (s_0\boldsymbol{E} - \boldsymbol{A})^{-1}\boldsymbol{B}, \tag{0.11}$$

where (0.5) is the standard formulation and (0.10) 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 subspace method iteratively constructs a basis but we often think of the progression in terms of a sequence of converging eigenvalues of $\mathbf{H}$, starting with the largest. In our case $\mathbf{H}$ is a shift-and-invert operator, so large eigenvalues λ of $\mathbf{H}$ are eigenvalues $\mu = s_0 + 1/\lambda$ 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 subspace methods 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 only a few "dominant" poles (§0.1.1) 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, 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, but determining optimal selection of shifts and number of iterations is *rational-interpolation*, a non-trivial problem. 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 methods that iterate with a shift-and-invert operator are sometimes called *rational-Krylov* methods: Rational Arnoldi, Rational Lanczos, etc. This term was introduced for eigenvalue problems in 1984 by [44] for a general Krylov method that uses multiple shifts, possibly changing at every iteration. Every Krylov subspace method used for model reduction that uses an arbitrary shift $s_0 \in \mathbb{C}$ (as opposed to only zero or ∞) is an instance of a rational Krylov method in the sense that it uses a shift-invert operator. For the current discussion we will limit ourselves to one shift s_0 and maybe cover multiple shifted rational Krylov methods if there is enough time.

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 subspace is the ideal subspace to use for projection because reduced order models obtained via Krylov subspaces have moment matching properties (§0.1.1). 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

$$\mathcal{K}_n(\mathbf{H}, \mathbf{r}) = \operatorname{span}\left\{\mathbf{r}, \mathbf{Hr}, \mathbf{H}^2\mathbf{r}, \ldots, \mathbf{H}^{n-1}\mathbf{r}\right\}, \tag{0.56}$$

and we may define a similar n-th *block* Krylov subspace

$$\mathcal{K}_n(\mathbf{H},\mathbf{R}) = \operatorname{span}\left\{\mathbf{R},\mathbf{H}\mathbf{R},\mathbf{H}^2\mathbf{R},\ldots\mathbf{H}^{n-1}\mathbf{R}\right\} \tag{0.57}$$

$$= \operatorname{span}\mathcal{K}_n(\mathbf{H},\mathbf{r}_1)\cup\mathcal{K}_n(\mathbf{H},\mathbf{r}_2)\cup\cdots\cup\mathcal{K}_n(\mathbf{H},\mathbf{r}_m). \tag{0.58}$$

In the block case, n is the block-degree of the space, indicating the number of powers of $\mathbf{H}$ are involved. The dimension of the space is generally larger, typically $n\dim\mathbf{R} = n\dim\boldsymbol{B}$ where $\dim\boldsymbol{B}$ is the number of input terminals of a circuit model, or the controllability constraint for a control system model. In some publications the actual dimension of the subspace is indicated in its name.

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.

General subspace projection

For now, let us assume only a general subspace $\mathcal{K}\subset\mathbb{C}^N$ on which we will project our model. Projection onto a subspace $\mathcal{K}$ is denoted with the general linear projection operator $\mathcal{P}_{\mathcal{K}}$. It is often sufficient and simpler to express projection that way, but if we want to express $\mathcal{K}$ in terms of a basis or spanning set V, then $\mathcal{P}_{\mathcal{K}}$ is defined in terms of V. $\mathcal{P}_{\mathcal{K}}$ is projection onto the subspace $\mathcal{K}$ spanned by V. If V has orthonormal columns (i.e. $V^TV = I$) then

$$\mathcal{P}_{\mathcal{K}} = VV^T. \tag{0.59}$$

We can similarly define projection for any set V that spans $\mathcal{K}$ but here we will assume that V is orthogonal, with a few exceptions. If V spans $\mathcal{K}$, we may write $\mathcal{P}_V$ rather than $\mathcal{P}_{\mathcal{K}}$ and it means the same thing.

The dimension of $\mathcal{K}$ is n and we assume n is much smaller than N, which is huge. We cannot directly obtain eigenvalues and vectors of $\mathbf{H}\in\mathbb{C}^{N\times N}$, but suppose we have a relatively easy way to work with the projected operator

$$\mathbf{H}_{\mathcal{K}} = \mathcal{P}_{\mathcal{K}}\mathbf{H}\mathcal{P}_{\mathcal{K}},$$

which is $\mathbf{H}$ restricted to $\mathcal{K}$, and acting only on $\mathcal{K}$; a sort of "low resolution" version of $\mathbf{H}$. Subspace projection methods exploit the **Rayleigh-Ritz** procedure, which approximates eigenvalues/vectors of $\mathbf{H}$ by those of the projected lower dimension operator $\mathbf{H}_{\mathcal{K}}$. Rayleigh-Ritz determines the vector y in $\mathcal{K}$ that best approximates an eigenvector of $\mathbf{H}$. The associated approximate eigenvector is given by the Rayleigh-quotient

$$\tilde{\lambda} = \frac{y^H\mathbf{H}y}{\|y\|}. \tag{0.60}$$

Usually we ensure that $\|y\| = 1$. $\tilde{\lambda}$ is called a *Ritz-value* of $\mathbf{H}$, with respect to $\mathcal{K}$. y is the associated Ritz-vector. The Ritz-pair $(\tilde{\lambda}, y)$ of $\mathbf{H}$ implies an approximate eigenpair $(\tilde{\mu}, y)$ of the matrix pencil $(\boldsymbol{A},\boldsymbol{E})$, where $\tilde{\mu} = s_0 + 1/\tilde{\lambda}$.

Alternatively, a direct approximation $\hat{\mu}$ to an eigenvalue of $(\boldsymbol{A},\boldsymbol{E})$ is given by the *generalized Rayleigh-quotient*

$$\hat{\mu} = \frac{y^H\boldsymbol{A}y}{y^H\boldsymbol{E}y}. \tag{0.61}$$

In this case the approximate eigenvector y of $(\boldsymbol{A},\boldsymbol{E})$ is an eigenvector of the projected matrix pencil

$$(\boldsymbol{A}_{\mathcal{K}},\boldsymbol{E}_{\mathcal{K}}) = (\mathcal{P}_{\mathcal{K}}\boldsymbol{A}\mathcal{P}_{\mathcal{K}},\mathcal{P}_{\mathcal{K}}\boldsymbol{E}\mathcal{P}_{\mathcal{K}})$$

and $\hat{\mu}$ is called a generalized Ritz-value of $(\boldsymbol{A},\boldsymbol{E})$, with respect to $\mathcal{K}$.

Given the same approximate eigenvector y, $\tilde{\mu}$ derived from (0.60) and $\hat{\mu}$ from (0.61) are not equal, but they converge to the same value as $\mathcal{K}$ grows in dimension.

- Suppose we find a potential eigen-pair $y \in \mathcal{K} \subset \mathbb{C}^N$ and $\hat{\lambda} \in \mathbb{C}$ that satisfy the *Galerkin* condition

  $$\mathcal{P}_{\mathcal{K}}(\mathbf{H}y - \hat{\lambda}y) = 0, \qquad \text{equivalently} \qquad (\mathbf{H}y - \hat{\lambda}y) \perp \mathcal{K} \tag{0.62}$$

  The Galerkin condition is that the *residual-vector* $(\mathbf{H} - \hat{\lambda}I)y$ associated with the pair $(\hat{\lambda}, y)$ is orthogonal to $\mathcal{K}$. Its projection onto $\mathcal{K}$ is zero. Such a vector y is called a *Ritz-vector* of $\mathbf{H}$ with respect to the subspace $\mathcal{K}$, and $\hat{\lambda}$ is the associated Ritz-value. It can be shown that, of all vectors in $\mathcal{K}$, the best approximates to eigenvectors of $\mathbf{H}$ are Ritz-vectors. Certainly an actual eigen-pair (λ, z) of $\mathbf{H}$ satisfies the Galerkin condition if $z \in \mathcal{K}$. It should also be clear that a Ritz-pair can satisfy (0.62), and not be a very good approximation to an eigen-pair of $\mathbf{H}$, so we still need some other way to verify the quality of the approximation.

  There is also a generalized Galerkin condition, defined similarly.

  $$\mathcal{P}_{\mathcal{K}}(\boldsymbol{A}y - \hat{\mu}\boldsymbol{E}y) = 0, \qquad \text{equivalently} \qquad (\boldsymbol{A}y - \hat{\mu}\boldsymbol{E}y) \perp \mathcal{K} \tag{0.63}$$

  (0.62) and (0.63) are related by $\hat{\mu} = s_0 + 1/\hat{\lambda}$, and are interchangeable, in theory. There may be numerical differences.

For a Ritz-pair $(\hat{\lambda}, y)$, since $y = \mathcal{P}_{\mathcal{K}}y$ we can rewrite (0.62) as

$$\mathbf{H}_{\mathcal{K}}y = \mathcal{P}_{\mathcal{K}}\mathbf{H}\mathcal{P}_{\mathcal{K}}y = \hat{\lambda}y.$$

This shows that Ritz-pairs, the best approximations to eigenpairs of $\mathbf{H}$ that we can get in $\mathcal{K}$, are eigen-pairs of the reduced dimension projected operator $\mathbf{H}_{\mathcal{K}}$. Since the subspace $\mathcal{K}$ is of dimension n, our computation only needs to involve a basis of size n. Suppose V is an othonormal basis for $\mathcal{K}$. Then, expressed in terms of V, the projected operator $\mathbf{H}_{\mathcal{K}}$ is

$$\widetilde{\mathbf{H}} = V^T\mathbf{H}V \in \mathbb{C}^{n \times n}.$$

Its eigenvectors are Ritz vectors of $\mathbf{H}$ with respect to $\mathcal{K}$, expressed in terms of the smaller basis V. In that sense, we say solutions $w \in \mathbb{C}^n$ to the eigenvalue problem

$$\widetilde{\mathbf{H}}w = \hat{\lambda}w$$

are the *short* Ritz-vectors (with respect to V) of $\mathbf{H}$, and the *long* Ritz-vectors $y = Vw \in \mathcal{K} \subset \mathbb{C}^N$ are the $\mathcal{K}$-approximate eigenvectors of $\mathbf{H}$. Typically when we refer to Ritz vectors we mean the long vectors y, expressed in the standard basis for $\mathbb{C}^N$. Being Ritz-vectors, these vectors of course satisfy the Galerkin condition, which with respect to V, is

$$V^T(\mathbf{H}y - \hat{\lambda}y) = 0$$

for $y = Vw$.

To summarize,

- A general projection method for finding eigenvalues/vectors of $\mathbf{H}$ uses the Rayleigh-Ritz procedure to approximate eigenvectors of $\mathbf{H}$ by eigenvectors of $\widetilde{\mathbf{H}}$ in an n-dimensional subspace $\mathcal{K}$, which are called Ritz-vectors. Given a basis V for the subspace $\mathcal{K}$, it computes eigenvectors/values $(\hat{\lambda}, w)$ of $\widetilde{\mathbf{H}} = V^T\mathbf{H}V$. The Ritz-vectors $y = Vw$ are approximate eigenvectors of $\mathbf{H}$ whose quality we can check by computing the relative residual norm

  $$\frac{\left\|\mathbf{H}y - \hat{\lambda}y\right\|}{\|\mathbf{H}\|} \quad \text{or} \quad \frac{\left\|\mathbf{H}y - \hat{\lambda}y\right\|}{|\hat{\lambda}|}. \tag{0.64}$$

  Vectors y with low relative residual norms are considered to be converged.

An essential feature of a *Krylov* subspace $\mathcal{K}$ is that projection of $\mathbf{H}$ onto $\mathcal{K}$ yields a simple expression for Ritz residuals $\mathbf{H}y - \hat{\lambda}y$ for all the Ritz vectors y without resorting to multiplying $\mathbf{H}y$. Otherwise, explicitly computing (0.64) for every Ritz-vector y is somewhat expensive. As a result most Krylov methods attempt to preserve this property, known as the Krylov relation.

Krylov subspace projection

A Krylov subspace projection method for model reduction owes its moment matching (§0.1.1) properties to the subspace

$$\mathcal{K}_n(\mathbf{H}, \mathbf{r}) = \operatorname{span}\left\{\mathbf{r}, \mathbf{Hr}, \mathbf{H}^2\mathbf{r}, \ldots, \mathbf{H}^{n-1}\mathbf{r}\right\}. \tag{0.56}$$

The first Krylov subspace method for finding eigenvalues of a Hermitian matrix $\mathbf{H}$ was developed in 1950 by Lanczos [29], and is perhaps unsurprisingly based on power iterations of $\mathbf{H}$ on $\mathbf{r}$.

For general (non-Hermitian) $\mathbf{H}$ there are two basic Krylov methods: The Arnoldi process and the non-symmetric Lanczos process. Nonsymmetric, or *two-sided* Lanczos also iterates with $\mathbf{H}^H$ ($\mathbf{H}^T$ for $s_0 \in \mathbb{R}$). A two-sided Krylov process for model reduction uses the observation that the j-th transfer function moment $\mathcal{H}^{(j)} = \boldsymbol{C}^T\mathbf{H}^j\mathbf{R}$ can be re-written as

$$\mathcal{H}^{(j)} = ((\mathbf{H}^{j-i})^T\boldsymbol{C})^T(\mathbf{H}^i\mathbf{R}) \quad \text{for any} \quad 0 \leq i \leq j, \tag{0.65}$$

to motivate construction of bases for *right* and *left* Krylov subspaces.

The right-Krylov subspace is already given as (0.56). The left Krylov subspace[6] is then

$$\mathcal{K}_n(\mathbf{H}, \boldsymbol{C}^T) = \operatorname{span}\left\{\boldsymbol{C}^T, \boldsymbol{C}^T\mathbf{H}, \boldsymbol{C}^T\mathbf{H}^2, \ldots, \boldsymbol{C}^T\mathbf{H}^{n-1}\right\}, \tag{0.66}$$

with $\mathbf{H}^H$ and $\boldsymbol{C}$, where $\boldsymbol{C}$ is the observability condition matrix (or vector) from (0.1).

Two bases V and W of the two Krylov subspaces imply the explicitly projected ROM (0.42) with realization

$$\boldsymbol{A}_n := V^T\boldsymbol{A}W, \quad \boldsymbol{E}_n := V^T\boldsymbol{E}W, \quad \boldsymbol{C}_n := W^H\boldsymbol{C}, \quad \boldsymbol{B}_n := V^T\boldsymbol{B}. \tag{0.67}$$

The first Krylov method for model reduction, PVL [16] developed in 1994, uses a two sided Lanczos process to compute the *implicitly* projected ROM transfer function (0.43).

There is also a method known as two-sided Arnoldi that produces the two bases V and W. The typical (one-sided) Arnoldi method produces only the right-Krylov subspace (0.56), and $W = V$ for explicit projections (0.67). It can be shown that a model produced with oblique projections (0.67) onto left and right Krylov subspaces matches twice as many moments as with just one. However, for the sake of simplicity we will limit ourselves to the Arnoldi-based krylov subspace methods and projecting with one orthonormal basis.

The Arnoldi process

Although generally thought of as a method for finding eigenvalues, Arnoldi actually introduced his method [4] in 1951 as a way to construct a similarity transformation of a matrix to upper Hessenberg form, and was not particularly interested in eigenvalues. It was not considered an eigenvalue method not until Saad re-introduced it with [46] in 1980 as a better numerically behaved alternative to the Lanczos method. 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.

[6] Technically the correct notation is $\mathcal{K}_n(\mathbf{H}^H, \boldsymbol{C})$, but terms like $(\mathbf{H}^H)^{n-1}$ look awkward.

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

The Arnoldi process (Algorithm 1) basically performs a power iteration and orthogonalizes each iterate against previous ones, thus producing an orthonormal basis for (0.56). 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. We introduce a cheaper inner product than that of line 5 in §0.1.5 that cuts computation in half for complex vectors and still provides a suitable spanning set V.

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[7] (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

$$nN^2 + n^2 N$$

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"

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

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

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.

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. Restarts present some difficulties for model reduction, and we might suggest some ways to implement them. Other than restarts or other ways to reduce the cost of orthogonalization, we can reduce computational cost by choosing a shift s_0 (or or a combination of shifts) in such a way that a reasonably approximate ROM converges for low n.

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[8]), 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. An example of this is transfer function surface plots like figure 2. For every sample point s, evaluating $\mathcal{H}(s)$ of the full, unreduced model requires solution of a $N \times N$ system. Figure 2 in particular was plotted on a 200×300 grid, so the plot required 60000 solves. For fast computation of the plot, we use a ROM of size n, where n is only as large as we need for a reasonable visible approximation. Construction of the ROM was a one-time cost, but the major computational cost was repeatedly employing the model.

Now consider the case where producing the model 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. For small and moderately sized ROM applications, for example, *balanced truncation* is considered optimal and favored over moment-matching methods.

Orthogonalization is projection Regardless of the numerical method used for orthogonalization, lines 4-7 are mathematically equivalent to

$$\begin{aligned} r_k &= \mathbf{H}v_k - \sum_{j=1}^{k} v_j^H \mathbf{H} v_j \\ &= \mathbf{H}v_k - V_k V_k^H \mathbf{H} v_j \\ &= (I - V_k V_k^H)\mathbf{H} v_k \\ &= (I - \mathcal{P}_{V_k})\mathbf{H} v_k, \end{aligned} \tag{0.68}$$

which is projection of the iterate $\mathbf{H}v_k$ onto the orthogonal complement $\mathcal{K}_k(\mathbf{H}, \mathbf{r})^\perp$ of the Krylov subspace $\mathcal{K}_k(\mathbf{H}, \mathbf{r})$ spanned by $V_k = \begin{bmatrix} v_1 & v_2 & \cdots & v_k \end{bmatrix}$. We call

$$r_k \in \mathcal{K}_k(\mathbf{H}, \mathbf{r})^\perp$$

the k-th *residual* vector. The k-th residual-vector r_k is what remains of the iterate $\mathbf{H}v_k$ after previously visited directions v_j have been subtracted away; it contains only the *new* spectral information introduced by the k-th iteration.

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

Block/band-Arnoldi If instead of a start vector we have a start-block $\mathbf{R} = \begin{bmatrix}\mathbf{r}_1 & \mathbf{r}_2 & \cdots & \mathbf{r}_m\end{bmatrix}$ then we can use a band or block-Krylov process. A block-Krylov process advances the block Krylov-subspace $\mathcal{K}_j(\mathbf{H}, \mathbf{R})$ on the j-th step by iterating blocks AV_j and orthogonalizing the basis block-wise. If we can assume the Krylov-blocks

$$\begin{bmatrix}\mathbf{R} & \mathbf{HR} & \mathbf{H}^2\mathbf{R} & \cdots & \mathbf{H}^j\mathbf{R}\end{bmatrix}$$

are linearly-independent then a block process may suffice, but in general we cannot assume that the constituent vectors will be linearly-independ ent. A so-called *band*-Krylov process handles vectors one at a time, downsizing the block if linear-dependence is encountered. A band-Arnoldi and a band-Lanczos were developed and discussed in detail in [3], and further developed a bit in [19]. When talking about MIMO Krylov-subspace model reduction we will assume we are using the band-Arnoldi process algorithm 3, which is that from [19].

The band-Arnoldi generate a basis $V = \begin{bmatrix}v_1 & v_2 & \cdots & v_n\end{bmatrix}$ for the block-Krylov subspace

$$\operatorname{span}\begin{bmatrix}v_1 & v_2 & \cdots & v_n\end{bmatrix} = \operatorname{span}\{R_0, R_1, \ldots, R_l\} = \mathcal{K}_l(\mathbf{H}, \mathbf{R}) \tag{0.69}$$

for which linear dependence between and within the blocks R_k, $k = 1, 2, \ldots, l$ is deflated. The blocks R_k are analogous to the residual-vectors r_k of algorithm 1. Note that we still denote n as the dimension of the projection basis V, although that is now achieved with $l \leq n$ iterations of band-Arnoldi that correspond to powers of $\mathbf{H}$ applied to $\mathbf{R}$. Here the l indicates the block-dimension of the subspace (0.69). This notation differs from that of some publications (such as [19, 18]) that denote the actual dimension n of the subspace in its name, as in "span $\begin{bmatrix}v_1 & v_2 & \cdots & v_n\end{bmatrix} = \mathcal{K}_n(\mathbf{H}, \mathbf{R})$".

Early termination of Arnoldi If $\|r_k\| = 0$ then that means r_k provides no further information and the process terminates. It means k has reached the *grade* $d(\mathbf{H}, \mathbf{r})$ of $\mathbf{H}$ with respect to $\mathbf{r}$. The grade is defined as the minimum $d \in \mathbb{N}$ such that

$$\mathcal{K}_d(\mathbf{H}, \mathbf{r}) = \mathcal{K}_{d+1}(\mathbf{H}, \mathbf{r}).$$

If we are lucky then $d(\mathbf{H}, \mathbf{r}) \leq N$ is very small. For example, if $\mathbf{r}$ happens to be an eigenvector of $\mathbf{H}$ then we are done after one iteration, and a reduced order model of size $n = 1$ completely characterizes the unreduced model! In practice d tends to be much larger than the number of iterations required to make a suitable ROM. Similarly, if d is greater than 1 but still fairly low, it means V_d spans an exact $\mathbf{H}$-invariant space of dimension d, i.e. the span of d eigenvectors.

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

$$\begin{aligned}\mathbf{H}V &= \begin{bmatrix}V & v_{n+1}\end{bmatrix}\begin{bmatrix}\longleftarrow & \widetilde{\mathbf{H}} & \longrightarrow \\ 0 & \cdots & h_{n+1,n}\end{bmatrix} \\ &= V\widetilde{\mathbf{H}} + \; h_{n+1,n}v_{n+1}e_n^T \\ &= V\widetilde{\mathbf{H}} + \; r_n e_n^T\end{aligned} \tag{0.70}$$

where $V \in \mathbb{C}^{N\times n}$ is the orthogonal basis matrix for $\mathcal{K}_n(\mathbf{H}, \mathbf{r})$, starting with $v_1 = \mathbf{r}/\|\mathbf{r}\|_2$, 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) residual-vector $r_n = h_{n+1,n}v_{n+1}$ of algorithm 1 is a notable quantity because it represents the error of the approximation $\mathbf{H}V \approx V\widetilde{\mathbf{H}}$. It can be shown that

$$\|r_n\|_2 = \frac{|\sigma_{\max}|}{|\sigma_{\min}|}$$

where $\sigma_{\min}$ and $\sigma_{\max}$ are the smallest and largest singular value of the matrix $\begin{bmatrix} V & \mathbf{H}v_n \end{bmatrix}$. Values $\|r_k\|_2$ for $k = 1, 2, \ldots, n$ are the sub-diagonal entries $h_{k+1,k}$ of $\widetilde{\mathbf{H}}$. One expects them to decrease in general, since $\|r_n\|$ is zero for $n \geq d(\mathbf{H}, \mathbf{r}) \leq N$, but in practice tend not to be monotonically decreasing. From a model reduction standpoint, a satisfactory model can be obtained without having n large enough to make $\|r_n\|$ small. This is because $\|r_n\|$ represents the amount of new spectral information of $\mathbf{H}$ discovered on the n-th step. A rapidly decreasing $\|r_n\|$ 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 from s_0 and moving outward), r_k is rich with information. The poles being discovered may or may not be significant in the sense of frequency response approximation, however.

Nevertheless, this value has been suggested as a stopping criterion for Krylov subspace methods in one publication [6]. An interesting multiple-shift method uses a scaling of the value $\boldsymbol{c}^T r_k$ (where $\boldsymbol{c}$ is the from the system realization (0.1)) to determine which shift to use for the next iteration. The reasoning is that $\boldsymbol{c}^T r_k$ (or some scaling thereof) is an indicator of moment error $\sum_{j=k+1}^{\infty} \boldsymbol{c}^T \mathcal{H}^{(j)} \mathbf{r}$. Given several Krylov processes indexed by j, each currently at the k_j-th iteration, they choose the one with greatest moment error, thus choosing the shift $s_0^{(j)}$ that results in the greatest reduction in moment error. Two variations of this idea are given in a rational Lanczos variant [21], and a Rational-Arnoldi variant [32, 31].

general Krylov relation The Arnoldi relation is a specific case of the general *Krylov-relation*

$$\mathbf{H}V = V\widetilde{\mathbf{H}} + r_n f^H, \tag{0.71}$$

in which $f = e_n = \begin{bmatrix} 0 & 0 & \cdots & 1 \end{bmatrix} \in \mathbb{R}^n$ is the n-th standard basis vector. Most Krylov methods produce structures that imply a Krylov relation (0.71) with rank-1 error matrix $r_n f^H$, but the Arnoldi relation's rank-1 error matrix $r_n e_n^T$ is particularly easy to work with.

Rayleigh-Ritz (approximate eigenvalues) from Arnoldi The Arnoldi relation (0.70) (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 relatively small and its eigenvalue decomposition

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

with diagonal matrix $\Lambda = \operatorname{diag}\begin{bmatrix} \lambda_1 & \lambda_2 & \cdots & \lambda_n \end{bmatrix}$ is presumably much easier to compute. Rayleigh-Ritz (§0.1.4) implies that the Ritz-values (eigenvalues λ of $\widetilde{\mathbf{H}}$) are approximate eigenvalues of $\mathbf{H}$, and long Ritz-vectors $Vw \in \mathcal{K}_n(\mathbf{H}, \mathbf{r})$ are the associated approximate eigenvectors of $\mathbf{H}$.

Left multiplying (0.70) with W yields

$$\begin{aligned} \mathbf{H}VW &= V\widetilde{\mathbf{H}}W + r_n e_n^T W \\ &= VW\Lambda + r_n e_n^T W \end{aligned}$$

$$\mathbf{H}Z = Z\Lambda + r_n e_n^T W$$

so for $z_j = Vw_j$,

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

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 the Ritz "error" vectors

$$\mathbf{H}z_j - \lambda_j z_j = \xi_j r_n$$

are all scalar multiples of the residual-vector r_n. Assuming $\|z_j\|_2 = 1$, the Arnoldi relation (0.70) thus implies a simple formulation

$$\mathrm{rr}_j = \frac{\|\mathbf{H}z_j - \lambda_j z_j\|_2}{\|\lambda_j z_j\|_2} = \frac{|\xi_j|}{|\lambda_j|} \|r_n\|_2 = \frac{|\xi_j|}{|\lambda_j|} |h_{n+1,n}| \tag{0.72}$$

for the relative residual-errors of the Rtiz values/vectors. Ritz vectors or values with low associated relative residuals (0.72) are good approximations to eigenvalues/vectors of $\mathbf{H}$ if they are well-conditioned. This is because (0.72) 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 $\|\mathcal{E}\| = \|r_n\|$. Rearrangement of the Arnoldi relation (0.70) reveals

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

If an eigenvalue of $\mathbf{H}$ is highly sensitive to perturbation (is badly conditioned) then a low or zero residual-error (0.72) 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. Very small eigenvalues of $\mathbf{H}$ (orders of magnitude lower than the largest) correspond to poles remote from s_0, which usually means far away from our segment of interest $i[\omega_0, \omega_1] \subset \mathbb{C}$ on the $\Im$-axis, so they can be considered to have constant influence on the system frequency response, i.e. their location is at ∞.

It is not clear how to treat infinite poles, or approximations to them, in a Krylov-subspace based model order reduction scheme that uses a shift-invert operator $\mathbf{H} = \mathbf{H}(s_0)$. The importance (or lack thereof) of matching poles at infinity may be a worthwhile area of research.

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 $(\boldsymbol{A}_n, \boldsymbol{E}_n) = (V^T\boldsymbol{A}V, V^T\boldsymbol{E}V)$. Some of this was covered for a general subspace in §0.1.4, whereas here we address approximate eigenvalues/vectors resulting from orthogonal projection with basis V of the rational Krylov subspace $\mathcal{K}_n(\mathbf{H}, \mathbf{R}) = \mathcal{K}_n((\boldsymbol{A} - s_0\boldsymbol{E})^{-1}\boldsymbol{E}, (s_0\boldsymbol{E} - \boldsymbol{A})^{-1}\boldsymbol{B})$ with a constant shift s_0.

- Eigenvalues $\hat{\lambda}$ of $\widetilde{\mathbf{H}}$ are called Ritz-values/vectors of $\mathbf{H}$ with respect to $\mathcal{K}_n(\mathbf{H}, \mathbf{R})$. Then values $\widetilde{\mu} = 1/\hat{\lambda} + s_0$, are approximations to eigenvalues of $(\boldsymbol{A}, \boldsymbol{E})$. Due to the shift-invert nature of $\mathbf{H}$, these approximate eigenvalues $\widetilde{\mu}$ of $(\boldsymbol{A}, \boldsymbol{E})$ are sometimes called *rational Ritz-values*. In some publications, these values are just called Ritz-values of $(\boldsymbol{A}, \boldsymbol{E})$. We will call them *implicit Ritz-values* of $(\boldsymbol{A}, \boldsymbol{E})$, since they are not eigenvalues of the explicitly-projected matrix pencil $(V^TAV, V^TEV) = (\boldsymbol{A}_n, \boldsymbol{E}_n)$, but they are implied by a Rayleigh-Ritz relation with the projection $\widetilde{\mathbf{H}} = V^T\mathbf{H}V$.
- Eigenvalues $\hat{\mu}$ of $(\boldsymbol{A}_n, \boldsymbol{E}_n)$ are sometimes called *generalized Ritz-values*, or just Ritz-values of $(\boldsymbol{A}, \boldsymbol{E})$ with respect to $\mathcal{K}_n(\boldsymbol{A}, \mathbf{R})$, but we will refer to them them as *explicit* Ritz-values to indicate that they are obtained by explicitly making the projections $\boldsymbol{A}_n = V^TAV$ and $\boldsymbol{E}_n = V^TEV$ from $\boldsymbol{A}$ and $\boldsymbol{E}$ as in (0.42).

So we have two different sets of approximations to $\sigma(\boldsymbol{A}, \boldsymbol{E})$, both implied by projection on to $\mathcal{K}_n(\boldsymbol{A}, \mathbf{R})$ via basis V: The set of **implicit Ritz-values**

$$\left\{ 1/\hat{\lambda} + s_0 \;\middle|\; \hat{\lambda} \in \sigma(\widetilde{\mathbf{H}}) \right\} \tag{0.73}$$

(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**

$$\{\, \hat{\mu} \in \sigma(\boldsymbol{A}_n, \boldsymbol{E}_n) \,\} \tag{0.74}$$

(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. (0.73) and (0.74) 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 (0.73) and (0.74), because they both result from projection on to the Krylov subspace $\mathcal{K}_n(\boldsymbol{A}, \mathbf{R})$, where $\boldsymbol{A} = \boldsymbol{A}(s_0)$ 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 (0.73) and with (0.74) 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 (0.72). Thus, if an eigen-pair $(\hat{\lambda}_j, w_j)$ of $\widetilde{\mathbf{H}}$ is converged, then we can expect that $(s_0 + 1/\hat{\lambda}_j, V w_j)$ is a converged eigen-pair of $(\boldsymbol{A}_n, \boldsymbol{E}_n)$ with about the same order of error of approximation to an eigen-pair of $(\boldsymbol{A}, \boldsymbol{E})$.

The reason we care about both sets of approximate eigenvalues/vectors is that implicit (0.73) Ritz-values/vectors are far cheaper to compute than the explicit variety (0.74), but the explicit formulation (0.3) is the end goal of explicit projection based MOR. Un-converged poles of the implicitly projected model transfer function can and often do have positive real-part, which is unfavorable for ROM applications. These eigenvalue approximations all move to the left half of the complex plane as they converge to their final resting values, but as long as there are any implicit Ritz-values $1/\hat{\lambda} + s_0$ with positive real part, the implicitly projected model (0.43) is possibly unstable and not attractive for model order reduction.[9] 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 [26, 39, 28, 2], have been developed which attempt to purge subspace components associated with "bad" (destabilizing) or otherwise unwanted eigenvalues from the constructed basis V, but they destroy moment matching properties and introduce other problems. Explicitly projected eigenvalues $\hat{\mu}$ of $(\boldsymbol{A}_n, \boldsymbol{E}_n)$ are always (for any n) well-behaved as long as the projection basis V is real, and as long as $\mathcal{K}_n(\boldsymbol{A}(s_0), \mathbf{R}(s_0)) \subseteq \operatorname{span} V$, the explicitly projected ROM on to V is guaranteed to be of matrix-Padé-type (match moments) with respect to s_0.

Moment matching property of Krylov subspace projected 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 subspace matches l moments about s_0, where l is the block-degree of the Krylov subspace

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

We will show this for implicitly projected ROMs (0.43) in Theorem 1, and explicitly projected ROMs (0.42) 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 (0.1), 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

$$\mathcal{H}(s + s_0) = \boldsymbol{C}^T \left(I - s\mathbf{H}\right)^{-1} \mathbf{R} \tag{0.75}$$

$$= \sum_{j=0}^{\infty} s^j \mathcal{H}^{(j)} \tag{0.8}$$

[9] Implicitly projected ROMs, such as those produced by PVL [16] often work fine in many practical applications despite being unstable, but they are currently unpopular.

where (0.8) 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 §0.1.1 to be

$$\mathcal{H}^{(j)} = \boldsymbol{C}^T \mathbf{H}^j \mathbf{R}. \tag{0.76}$$

Moment matching of the implicitly projected ROM The implicitly projected ROM transfer function

$$\widetilde{\mathcal{H}}(s + s_0) = \boldsymbol{C}_n^T \left(I - s\widetilde{\mathbf{H}} \right)^{-1} \widetilde{\boldsymbol{\rho}}_n \tag{0.77}$$

is defined via projection of (0.75), as

$$\widetilde{\mathbf{H}} = V^T \mathbf{H} V, \quad \boldsymbol{C}_n = V^T \boldsymbol{C}, \quad \widetilde{\boldsymbol{\rho}}_n = V^T \mathbf{R} \tag{0.78}$$

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 (0.77) 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 (0.43), (0.77) and moments $\mathcal{H}^{(j)}(s_0)$ of the URM transfer function (0.5) about s_0 are related by*

$$\widetilde{\mathcal{H}}^{(j)} = \boldsymbol{C}_n^T \widetilde{\mathbf{H}}^j \widetilde{\boldsymbol{\rho}}_n = \boldsymbol{C}^T \mathbf{H}^j \mathbf{R} = \mathcal{H}^{(j)} \tag{0.79}$$

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

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

$$\mathbf{H}^j \mathbf{R} = V \widetilde{\mathbf{H}}^j \widetilde{\boldsymbol{\rho}}_n \tag{0.80}$$

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

For $j = 0$, (0.80) follows from (0.78). Now assume (0.80) holds for some $j \in \{0, 1, \ldots, l - 2\}$. Applying $\mathbf{H}$ to (0.80) 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 (0.3) transfer function is a little more involved than Theorem 1 for the implicitly projected model. It is included as Theorem 2. The proof is adapted from [18, proposition 6 and theorem 7].

Recall the explicitly-projected ROM (0.3) with transfer function

$$\widehat{\mathcal{H}}_n(s) = \boldsymbol{C}_n^T (s\boldsymbol{E}_n - \boldsymbol{A}_n)^{-1} \boldsymbol{B}_n, \tag{0.42}$$

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 (0.42) are

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

where the structures

$$\widehat{\mathbf{H}} := (\boldsymbol{A}_n - s_0\boldsymbol{E}_n)^{-1}\boldsymbol{E}_n \quad \text{and} \quad \widehat{\boldsymbol{\rho}}_n := (s_0\boldsymbol{E}_n - \boldsymbol{A}_n)^{-1}\boldsymbol{B}_n. \tag{0.81}$$

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

$$\mathbf{H} := (\boldsymbol{A} - s_0\boldsymbol{E})^{-1}\boldsymbol{E}, \quad \mathbf{R} := (s_0\boldsymbol{E} - \boldsymbol{A})^{-1}\boldsymbol{B} \tag{0.11}$$

of the unreduced model (0.1). 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 (0.42) matches l moments.

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

$$F_n := V(\boldsymbol{A}_n - s_0\boldsymbol{E}_n)^{-1}V^T\boldsymbol{E}. \tag{0.82}$$

Then for $j \leq l \leq n$, the j-th moment $\widehat{\mathcal{H}}^{(j)}$ of the explicitly projected ROM transfer function (0.42) and moment $\mathcal{H}^{(j)}$ of the unreduced model (0.5) are related by

$$\widehat{\mathcal{H}}^{(j)} = \boldsymbol{C}^T F_n^i \mathbf{R} \quad \textit{for} \quad i = 0, 1, \ldots \tag{0.83}$$

$$= \mathcal{H}^{(i)} \qquad \textit{for} \quad i = 0, 1, \ldots, j-1. \tag{0.84}$$

Proof. First we show (0.83). Since $\operatorname{span} \mathbf{H}^i\mathbf{R} \subseteq \mathcal{K}_l(\mathbf{H}, \mathbf{R})$ for $i = 0, 1, \ldots, j$ and $\mathcal{K}_l(\mathbf{H}, \mathbf{R}) \subseteq \operatorname{span} V$, for each $i = 1, 2, \ldots, j$ there is a matrix X_i such that

$$\mathbf{H}^{i-1}\mathbf{R} = VX_i. \tag{0.85}$$

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

$$V^T\boldsymbol{B} = V^T(s_0\boldsymbol{E}V - \boldsymbol{A}V)X_1$$

$$\boldsymbol{B}_n = (s_0\boldsymbol{E}_n - \boldsymbol{A}_n)X_1.$$

Then

$$X_1 = (s_0\boldsymbol{E}_n - \boldsymbol{A}_n)^{-1}\boldsymbol{B}_n = \widehat{\boldsymbol{\rho}}_n. \tag{0.86}$$

Right-multiplying (0.82) 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,

$$F_n^iV = V\widehat{\mathbf{H}}^i \quad \text{for} \quad i = 0, 1, \ldots. \tag{0.87}$$

Then moments of the ROM transfer function

$$\begin{aligned}\widehat{\mathcal{H}}^{(i)} = \boldsymbol{C}_n^T\widehat{\mathbf{H}}^i\widehat{\boldsymbol{\rho}}_n = \boldsymbol{C}^TV\widehat{\mathbf{H}}^i\widehat{\boldsymbol{\rho}}_n \\ &= \boldsymbol{C}^T(F_n^iV)X_1 \quad \text{by (0.86) and (0.87)} \\ &= \boldsymbol{C}^TF_n^i\mathbf{R} \qquad \text{by (0.85) with } i = 1,\end{aligned}$$

which is (0.83).

Proof of (0.84) is implied by

$$\mathbf{H}^j\mathbf{R} = F_n^j\mathbf{R} \quad \text{for} \quad i = 0, 1, \ldots, j-1 \tag{0.88}$$

which we show by induction on i. (0.88) is trivial for $i = 0$. Now assume (0.88) 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:

$$\big((\boldsymbol{A} - s_0\boldsymbol{E})^{-1}\boldsymbol{E}\big)(F_n^i\mathbf{R}) = \mathbf{H}(\mathbf{H}^i\mathbf{R}) = \mathbf{H}^{i+1}\mathbf{R} = VX_{i+2} \tag{0.89}$$

where the rightmost expression follows from (0.85). Left-multiplying (0.89) with $V^T(\boldsymbol{A} - s_0\boldsymbol{E})$ yields

$$(V^T\boldsymbol{E})(F_n^i\mathbf{R}) = (V^T(\boldsymbol{A} - s_0\boldsymbol{E})V)X_{i+2} = (\boldsymbol{A}_n - s_0\boldsymbol{E}_n)X_{i+2}. \tag{0.90}$$

Then

$$\begin{aligned}F_n^{i+1}\mathbf{R} &= F_n(F_n^i\mathbf{R}) \\ &= \Big(V(\boldsymbol{A}_n - s_0\boldsymbol{E}_n)^{-1}V^T\boldsymbol{E}\Big)(F_n^i\mathbf{R}) \\ &= V(\boldsymbol{A}_n - s_0\boldsymbol{E}_n)^{-1}(V^T\boldsymbol{E})(F_n^i\mathbf{R}) \\ &= VX_{i+2}, \qquad \text{by (0.90)} \\ &= \mathbf{H}^{i+1}\mathbf{R}, \qquad \text{by (0.85)}\end{aligned}$$

which proves (0.88). Applying $\boldsymbol{C}^T$ yields (0.84), i.e.

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

□

0.1.5 Complex expansion points

If a shift s_0^j is not real (i.e. $\Im(s_0^j) \neq 0$) then the basis $V_j \in \mathbb{C}^{N\times n_j}$ of its associated Krylov subspace is also complex. Grimme discusses rational-Krylov interpolation point selection for MOR in depth, in [25]. 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 two new vectors provide only one piece of 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 also because the system (0.1) 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 subspace is to split the n vector basis V into $2n$ real vectors $v_j^{\mathbf{r}} = \Re(v_j)$ and $v_j^{\mathbf{i}} = \Im(v_j)$, forming $V^* \in \mathbb{R}^{N\times 2n}$, which spans the so-called *split-space*[10]

$$\begin{aligned}
&\operatorname{span}\begin{bmatrix} v_1^{\mathbf{r}} & v_1^{\mathbf{i}} & v_2^{\mathbf{r}} & v_2^{\mathbf{i}} & \cdots & v_n^{\mathbf{r}} & v_n^{\mathbf{i}} \end{bmatrix} \\
&\quad = \operatorname{span} \mathcal{K}_l(\mathbf{H}, \mathbf{R}) \cup \mathcal{K}_l(\overline{\mathbf{H}}, \overline{\mathbf{R}}) \\
&\quad = \operatorname{span} \mathcal{K}_l(\mathbf{H}(s_0), \mathbf{R}(s_0)) \cup \mathcal{K}_l(\mathbf{H}(\overline{s_0}), \mathbf{R}(\overline{s_0})) \\
&\quad = \mathcal{K}_l(\mathbf{H}, \mathbf{R})^*
\end{aligned} \tag{0.91}$$

of dimension $\eta \leq 2n$. The basis for a standard Krylov subspace may have complex vectors but its span is generally considered over $\mathbb{R}$. A split complex Krylov subspace admits a real basis but its span is over $\mathbb{C}$, so it should still be considered a complex space and

$$\mathcal{K}_l(\mathbf{H}, \mathbf{R}) \subseteq \mathcal{K}_l(\mathbf{H}, \mathbf{R})^*. \tag{0.92}$$

Notice that the split Krylov subspace (0.91) 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 [40], 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.

Producing a real basis

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 (0.91). 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 $\Im$ parts as $\begin{bmatrix} V^{\mathbf{r}} & V^{\mathbf{i}} \end{bmatrix}$, or in an order preserving way, like

$$\begin{bmatrix} v_1^{\mathbf{r}} & v_1^{\mathbf{i}} & v_2^{\mathbf{r}} & v_1^{\mathbf{i}} & \cdots & v_n^{\mathbf{r}} & v_n^{\mathbf{i}} \end{bmatrix}. \tag{0.93}$$

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

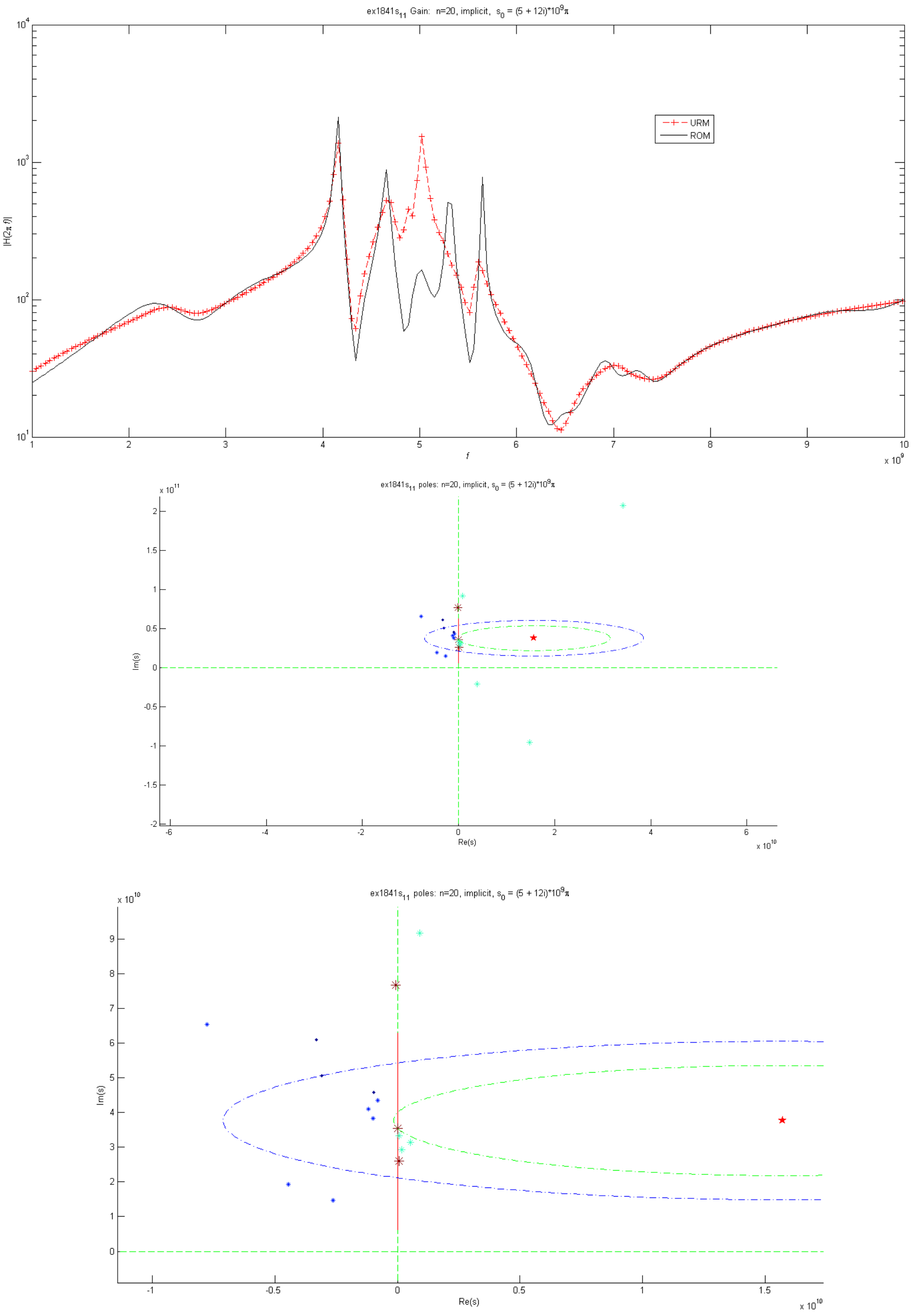


Figure 12: Frequency response and weighted-pole plots for a $n = 20$ implicitly projected ROM of `1841s11` with complex shift $s_0 = (5 + 12i) \cdot 10^9$ indicated by $\star$. The second pole plot is a zoomed-in view of the first one. Note the asymmetry of poles about the $\Re$-axis, and the "bad" poles on the positive side of the $\Im$-axis, which make this ROM undesirable. Circles around s_0 indicating distance from the 1-st and 10-th closest poles appear like flat ellipses due to plot scaling. They give a visual impression of why putting s_0 on the $\Im$-axis can cause more insignificant poles to converge, making the ROM larger than necessary.

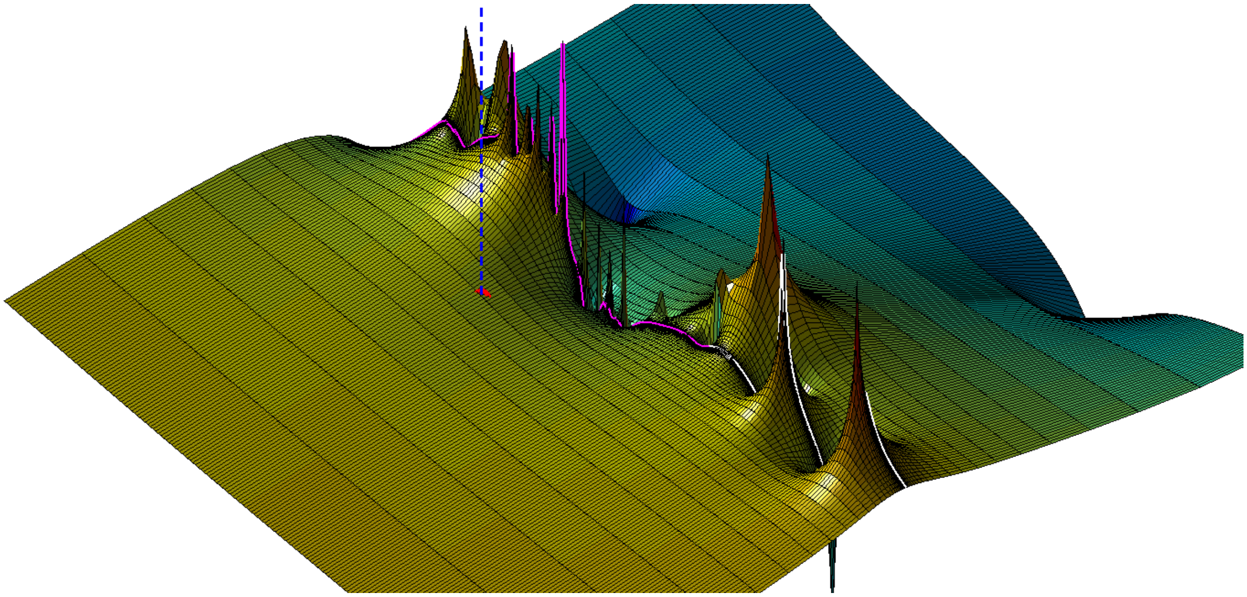


Figure 13: Surface plots of the $n = 20$ (complex) implicitly projected ROM of `1841s11` with shift $s_0 = (5 + 12i) \cdot 10^9$ from figure 12 indicated by a dot and vertical dashed line. It is plotted with 250 points over $\Re(s) \in [10^{-10}, 10^{10}] \times 350$ points $\Im(s) \in [10^8, 10^{10.2}]$, with square scaling. The entire frequency response is highlighted but the range of interest $i[10^9, 10^{10}]$ is darker. The projected operator $\widetilde{\mathbf{H}} \in \mathbb{C}^{20\times 20}$ from which the transfer function (0.43) is computed is complex so it is actually numerically twice as large as $n = 20$ indicates.

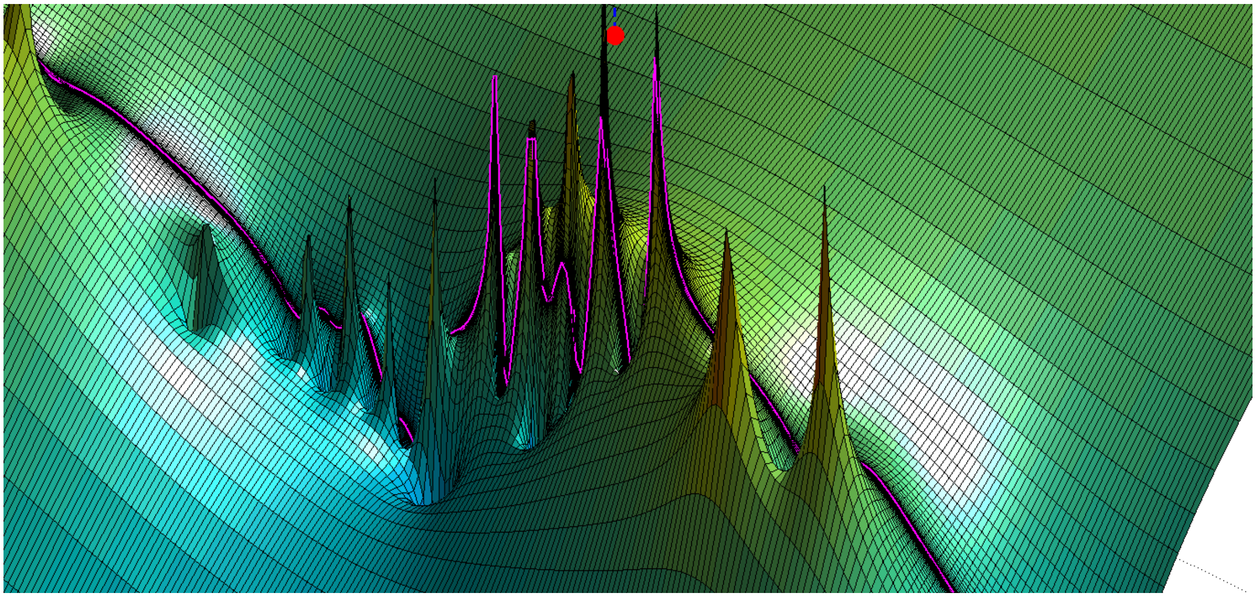


Figure 14: Viewed from the other side, it is apparent that although the peaks approximate peaks of the original model, some poles (and zeros) are on the "wrong" side of the frequency response gain curve. Is it also bad to have *zeros* with positive $\Re$-part? That appears not to be addressed in math model reduction literature. The expansion point s_0 can be seen as a dot near top-center of this plot.

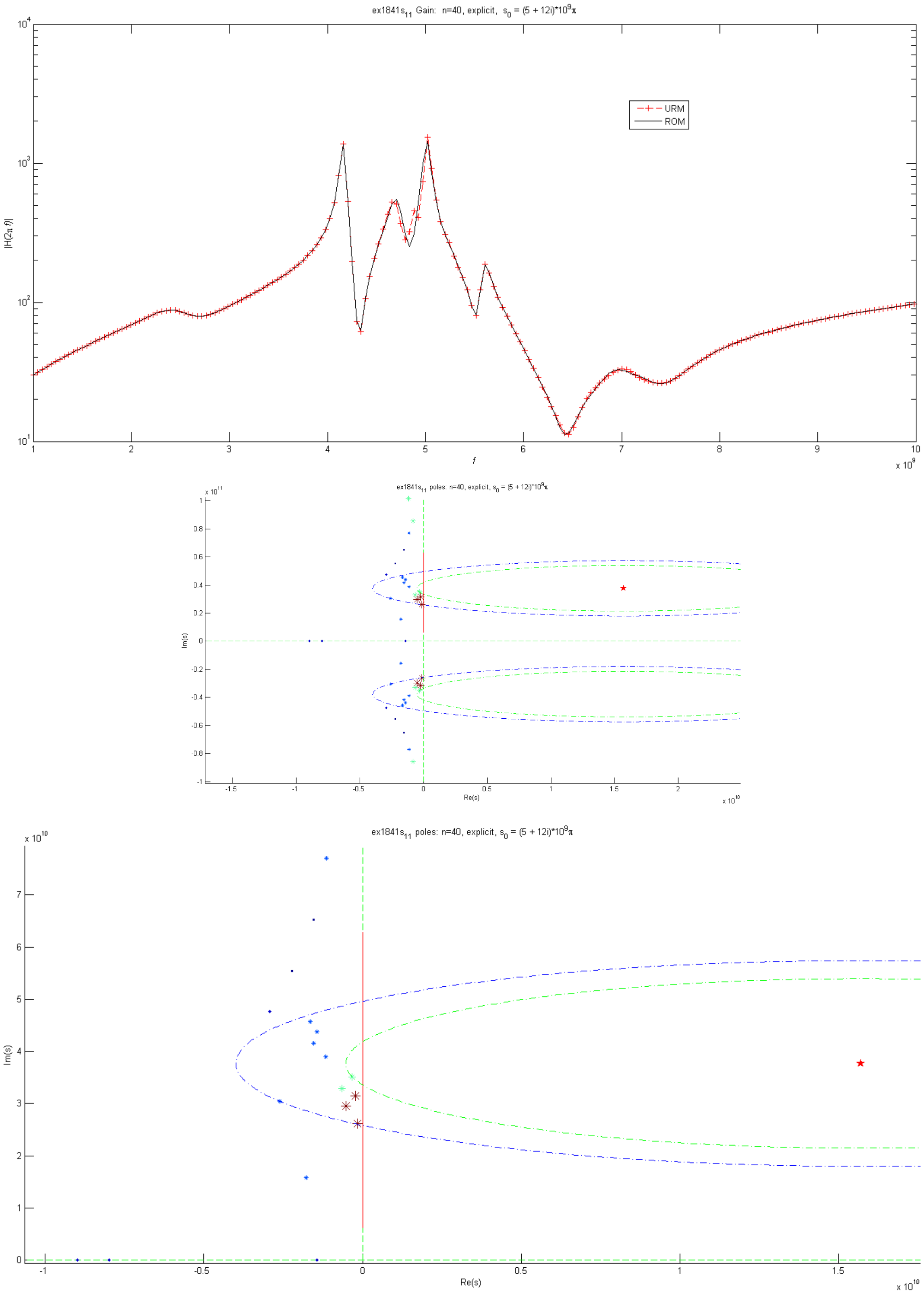


Figure 15: Frequency response and weighted-pole plots for a $n = 40$ explicitly projected ROM of `1841s11` with complex shift $s_0 = (5 + 12i) \cdot 10^9$ indicated by $\star$. The second pole plot is a zoomed-in view of the first one. Since this is an explicit projection of `1841s11` with a real basis, poles are distributed symmetrically about the $\Re$-axis, and there are no "bad" poles on the positive side of the $\Im$-axis. This ROM was produced by a 20 iteration cycle of Arnoldi and thus matches 20 moments about s_0 (and 20 moments about $\overline{s_0}$). Due to the scaling of these plots, circles around s_0 (and $\overline{s_0}$) appear as flat ellipses. They indicate distances of the 1-st and 10-th closest poles of the ROM to s_0, and give a sense the order in which they are expected to converge. This s_0 placement is close to optimal for this model, as dominant poles are some of the first to converge.

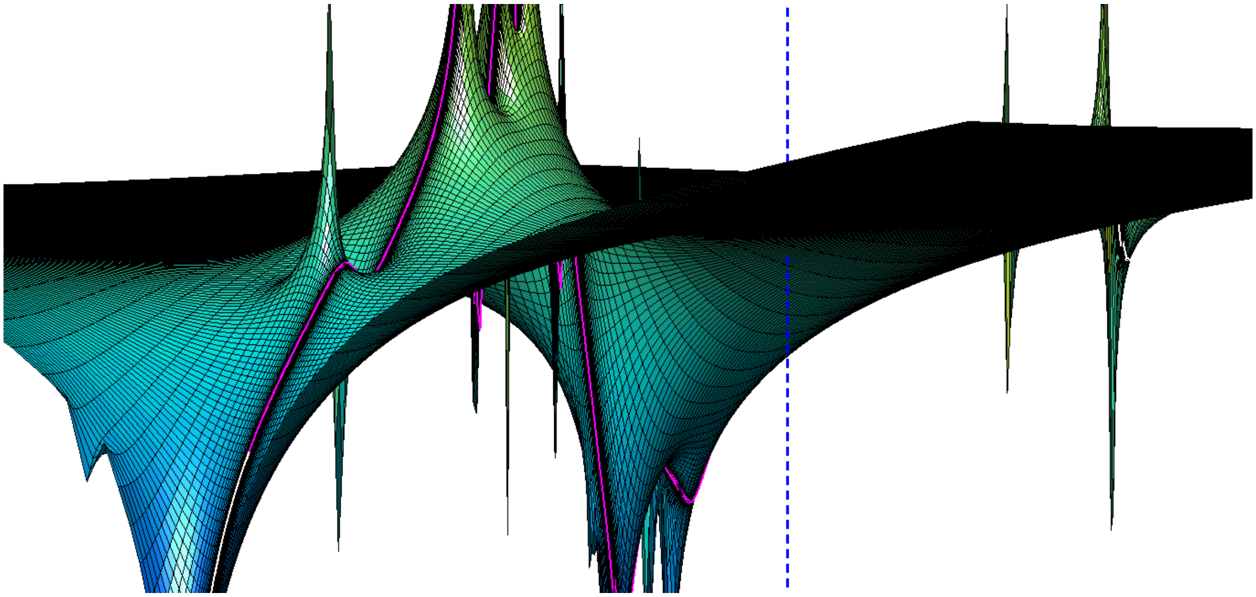


Figure 16: Surface plots of the $n = 40$ (real) explicitly projected ROM of `1841s11` with complex shift $s_0 = (5 + 12i) \cdot 10^9$ from figure 15 indicated by a dot and vertical dashed line. The entire frequency response is highlighted but the range of interest $i[10^9, 10^{10}]$ is darker. Clearly this is a better ROM approximation of `1841s11` than the implicit ROM from figure 15. It is of order 20 but numerical size $n = 40$, created by 20 iterations of the Arnoldi process and explicitly projected with a real basis $V \in \mathbb{R}^{1841\times 40}$.

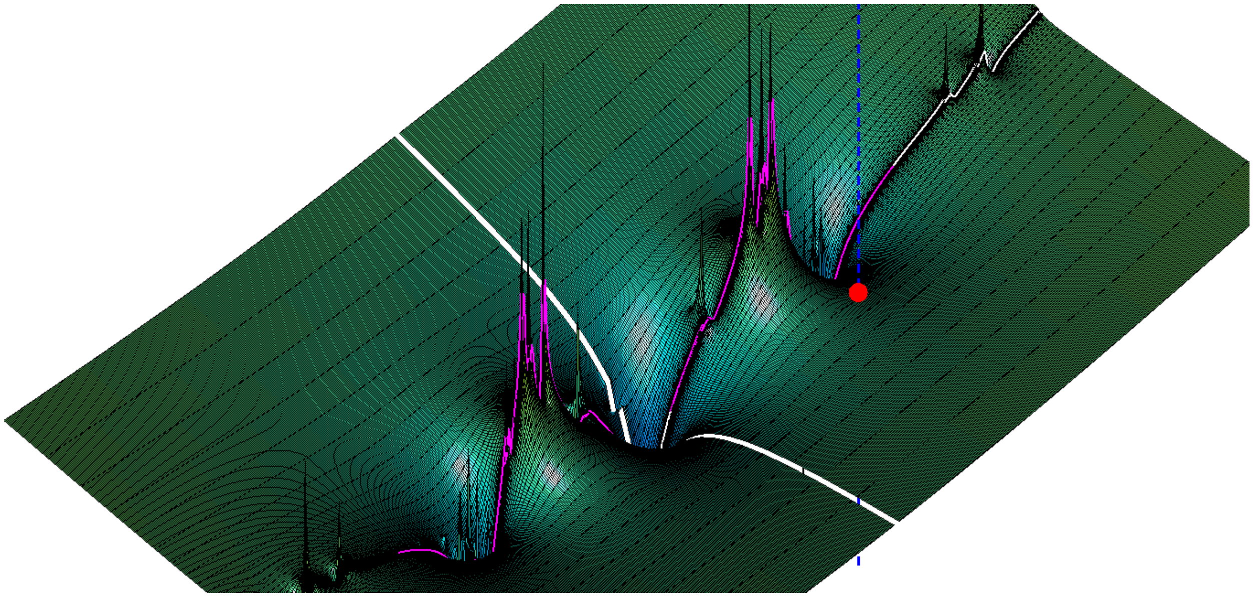


Figure 17: Here is the same transfer function but since it is symmetric about the $\Re$-axis we mirrored the plot. The small gap in the center is $[-10^8, 10^8] \subset [10^{-10.2}, 10^{10.2}]$.

But the set (0.93) is no longer orthogonal, and possibly linearly dependent. Making an orthonormal basis then requires orthogonalization of (0.93) consisting of at least n and most $2n$ vectors in $\mathbb{R}^N$, requiring another $(2n)^2N$ flops.

Ruhe's method For the reason previously described (splitting and costly orthogonalization), it would be ideal to create an orthogonal, real basis for (0.91) directly during the iterative process. Ruhe addresses this in [45], 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

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

We orthogonalize each real vector separately against previous real vectors $V_k = \begin{bmatrix} v_1 & v_2 & \cdots & v_k \end{bmatrix}$ so that

$$r_k = a_k - V_k h_k,$$

where $h_k = V_k^T a_k$ is the k-th column of the real upper-Hessenberg Arnoldi matrix $\widetilde{\mathbf{H}}$ Then $v_{k+1} = r_k / \|r_k\|$ yielding $V_{k+1} = \begin{bmatrix} v_1 & v_2 & \cdots & v_{k+1} \end{bmatrix}$.

We do a similar thing for the $\Im$ component:

$$r_{k+1} = b_k - V_{k+1} h_{k+1},$$

where $h_{k+1} = V_{k+1}^T b_k$ is the $(k+1)$-th column of $\widetilde{\mathbf{H}}$. Then $v_{k+2} = r_{k+1} / \|r_{k+1}\|$. Then we have

$$V_{k+2}(h_k + ih_{k+1}) = \mathbf{H}v_k$$

for $k = 0, 2, 4, \ldots$, with $\mathbf{H}v_0 := \mathbf{r}$.

The problem with the method just described is that it is not clear what vector we should iterate with next in order to build a basis for (0.91). 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 (0.91), 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.

Complex "lazy" orthgognalization with 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

$$\langle a + ib, x + iy \rangle_{\mathbb{R}} = x^T a + y^T b \in \mathbb{R}, \tag{0.94}$$

instead of the complex Euclidean inner product

$$(x + iy)^H (a + ib) = x^T a + y^T b + i(x^T b - y^T a) \in \mathbb{C}. \tag{0.95}$$

for two complex vectors $a+ib$ and $x+iy$. Note that $\langle u, v \rangle = \Re(u^H v)$. The inner-product (0.94) 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. (0.94) is cheaper to compute than (0.95), and vector scaling with the real-valued (0.94) 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 (0.4), and it may not even span $\mathcal{K}_l(\mathbf{H}, \mathbf{R})$. However, it satisfies (0.91); 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. For lack of a better term we will call a set *kinda orthgonal* if it is orthogonal with respect to (0.94).

Constructing such a kinda-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 (0.94). The matrix

$$\boldsymbol{G} = \begin{bmatrix} g_{jk} \end{bmatrix} \neq V^H \mathbf{H} V \tag{0.96}$$

of orthogonalization coefficients is no longer an orthogonal projection in the Euclidean sense.

Equivalent real formulations A Krylov process that orthogonalizes iterates with respect to (0.94) is effective for constructing a split-worthy basis because it is the Euclidean inner product on the Krylov subspace $\mathcal{K}_l(\ddot{\mathbf{H}}, \ddot{\mathbf{R}}) \subset \mathbb{R}^{2N}$ induced by the *equivalent real*, or *realfied* formulations

$$\ddot{\mathbf{H}} = \begin{bmatrix} \mathbf{H}^{\mathbf{r}} & -\mathbf{H}^{\mathbf{i}} \\ \mathbf{H}^{\mathbf{i}} & \mathbf{H}^{\mathbf{r}} \end{bmatrix} \quad \text{and} \quad \ddot{\mathbf{R}} = \begin{bmatrix} \mathbf{R}^{\mathbf{r}} \\ \mathbf{R}^{\mathbf{i}} \end{bmatrix}. \tag{0.97}$$

of $\mathbf{H} = \mathbf{H}^{\mathbf{r}} + i\mathbf{H}^{\mathbf{i}}$ and $\mathbf{R} = \mathbf{R}^{\mathbf{r}} + i\mathbf{R}^{\mathbf{i}}$ from (0.11). This idea is from [40, Sec. 5] and [12, 'K1-formulation']. There is a general definition and discussion of realified spaces as a pure-mathematics topic in [38].

A basis

$$\ddot{V} = \begin{bmatrix} \ddot{v}_1 & \ddot{v}_2 & \cdots & \ddot{v}_n \end{bmatrix} \tag{0.98}$$

for the Krylov subspace $\mathcal{K}_l(\ddot{\mathbf{H}}, \ddot{\mathbf{R}})$ induced by the realified matrices (0.97) 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 (0.98) as

$$\ddot{V}_n^* := \begin{bmatrix} \ddot{v}_1^{\mathbf{t}} & \ddot{v}_1^{\mathbf{b}} & \ddot{v}_2^{\mathbf{t}} & \ddot{v}_2^{\mathbf{b}} & \cdots & \ddot{v}_n^{\mathbf{t}} & \ddot{v}_n^{\mathbf{b}} \end{bmatrix}, \tag{0.99}$$

which is analogous to the split (0.91) of set of complex vectors.

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 $\mathcal{K}_l(\mathbf{H}, \mathbf{R})$ 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 (0.94).

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 (0.97). 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.*

$$(\mathbf{H}^j\mathbf{R})^* = \ddot{\mathbf{H}}^j\ddot{\mathbf{R}}. \tag{0.100}$$

Equivalently,

$$\ddot{\mathbf{H}}^j\ddot{\mathbf{R}} = \begin{bmatrix} \Re(\mathbf{H}^j\mathbf{R}) \\ \Im(\mathbf{H}^j\mathbf{R}) \end{bmatrix}. \tag{0.100}$$

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

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

□

It follows as a corollary that the split-Krylov subspaces (0.91) and (0.99) induced by each pair, are equal.

$$\mathcal{K}_n(\ddot{\mathbf{H}}, \ddot{\mathbf{R}})^* = \mathcal{K}_n(\mathbf{H}, \mathbf{R})^* \tag{0.101}$$

The inner products (0.95) and (0.94) yield different notions of orthogonality of a complex basis and its equivalent-real counterpart, and ultimately incompatible spaces. The inner product (0.94) implies a weaker orthogonality than (0.95): 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 (0.101) of Lemma 0.100 is nice.

The norms implied by (0.95) and (0.94) for a complex vector $v \in \mathbb{C}^N$ and its equivalent real form $\ddot{v} \in \mathbb{R}^{2N}$ are equal:

$$\|\ddot{v}\|_2^2 = \ddot{v}^T\ddot{v} = v^H v = \langle v, v\rangle = \|v\|_2^2\,. \tag{0.102}$$

Lemma 0.100 establishes that complex and realified forms of $\mathbf{H}$ and $\mathbf{R}$ induce the same split Krylov subspace $\mathcal{K}_l(\mathbf{H}, \mathbf{R})^*$ with block-degree l. The next result establishes that the basis vectors produced by an iteration of the Arnoldi process advance the split-Krylov subspace (0.91) in the same order, regardless of whether we use complex or equivalent-real formulation (i.e. complex formulation with inner product (0.94)).

We show this for the simpler case that $\mathbf{R} = \mathbf{r} \in \mathbb{C}^N$ is a single vector so that the block-degree l of the Krylov subspace in question is equal to the number n of basis vectors. We are convinced that the result of Theorem 3 can be extended to the general block-Krylov case, which is applicable to MIMO model reduction. We leave it up to an ambitious researcher to prove it for a general block-Krylov process.

Theorem 3. *Consider* $\mathbf{H}$, $\mathbf{r}$ *from* (0.11) *and their realified formulations* $\ddot{\mathbf{H}}$ *and* $\ddot{\mathbf{r}}$ *defined by* (0.97)*. 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*

$$\Re(v_n) = \alpha\ddot{v}_n^{\mathbf{t}} + w \qquad \textit{and} \qquad \Im(v_n) = \beta\ddot{v}_n^{\mathbf{b}} + z. \tag{0.103}$$

Proof. We will prove (0.103) by induction. For $n = 1$ we have $v_1 = \mathbf{r}/\left\|\mathbf{r}\right\|_2$, $\ddot{v}_1 = \ddot{\mathbf{r}}/\left\|\ddot{\mathbf{r}}\right\|_2$, where $\left\|\ddot{\mathbf{r}}\right\|_2 = \left\|\mathbf{r}\right\|_2$ by (0.102), so $\Re(v_1) = \ddot{v}_1^{\mathbf{t}}$ and $\Im(v_1) = \ddot{v}_1^{\mathbf{b}}$, trivially satisfying (0.103).

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 $\mathcal{K}_n(\mathbf{H}, \mathbf{r})^*$ with dimension η, so that

$$\mathcal{K}_n(\mathbf{H}, \mathbf{r}) \subseteq \mathcal{K}_n(\mathbf{H}, \mathbf{r})^* = \operatorname{span}\begin{bmatrix} \widetilde{v}_1 & \widetilde{v}_2 & \cdots & \widetilde{v}_\eta \end{bmatrix} \tag{0.104}$$

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

$$\begin{aligned} h_{n+1,n} v_{n+1} &= \mathbf{H} v_n - \sum_{j=1}^{n} h_{jn} v_j \\ &= \mathbf{H}^n \mathbf{r} + \sum_{j=1}^{n} c_j v_j, \qquad c_j \in \mathbb{R} \\ &= \mathbf{H}^n \mathbf{r} + \sum_{j=1}^{\eta} d_j \widetilde{v}_j, \qquad d_j \in \mathbb{C}, \quad \text{by (0.104)} \\ &= \mathbf{H}^n \mathbf{r} + w_1 + i z_1, \qquad w_1, z_1 \in \mathcal{K}_n(\mathbf{H}, \mathbf{r})^* \cap \mathbb{R}^N. \end{aligned} \tag{0.105}$$

Lemma 1 implies that we can re-write (0.105) in realified form as

$$h_{n+1,n} \begin{bmatrix} \Re(v_{n+1}) \\ \Im(v_{n+1}) \end{bmatrix} = \ddot{\mathbf{H}}^n \ddot{\mathbf{r}} + \begin{bmatrix} w_1 \\ z_1 \end{bmatrix}. \tag{0.106}$$

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

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

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 (0.3) using a basis (0.91) for the split-Krylov subspace $\mathcal{K}_l(\mathbf{H}, \mathbf{R})^*$ is equivalent regardless of the orthogonalization method used to construct it. The matrix $\boldsymbol{G}$ of orthogonalization coefficients from the equivalent-real Arnoldi process, (0.96), 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 (0.97), 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 $\ddot{V} \in \mathbb{R}^{2N \times n}$, where $\operatorname{span} \ddot{V} = \mathcal{K}_l(\ddot{\mathbf{H}}, \ddot{\mathbf{R}})$ for the implied Krylov subspace of block-dimension l, induced by equivalent-real forms (0.97). Thus, an implicitly projected ROM (0.43) 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 does not seem particularly useful for convergence analysis, since we are interested in the spectrum of $\mathbf{H}$. If there is more time we will address this.

0.2 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 [21] (and [24]) for model order reduction was done by Grimme in the mid and late 1990s. It is somewhat based on Ruhe's Rational-Krylov [44, 45] eigenvalue method and formalization, and possibly Olsson's [36]. Of particular interest are [25] and [14], both of which discuss interpolation-point selection. We refer the reader to those sources for the details of point selection.

[31, 32, 30, 17]) are more recent multi-point rational-interpolation methods. Also a Jacobi-Davidson MOR method [7]. Lee, Chu, and Feng's RAMAO/AORA method (Rational Arnoldi Method with Adaptive Order selection/ Adaptive-Order Rational-Arnoldi) [31, 32] breaths new life into an adaptive point-selection method introduced by [21], based on the sequence of ROM *moment-errors* implied by the sequence of residual vectors of the Arnoldi process (r_k in algorithm 1).

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

$$\mathcal{K}_j = \mathcal{K}_{l_j}(\mathbf{H}_j, \mathbf{R}_j) = \operatorname{span}\begin{bmatrix}\mathbf{R}_j & \mathbf{H}_j\mathbf{R}_j & \mathbf{H}_j^2\mathbf{R}_j & \cdots & \mathbf{H}_j^{l_j-1}\mathbf{R}_j\end{bmatrix}$$

of dimension n_j (block-dimension l_j), induced by

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

is contained in the span of V, so that

$$\mathcal{K}_1 \cup \mathcal{K}_2 \cup \cdots \cup \mathcal{K}_\tau \subseteq \operatorname{span} V \tag{0.1}$$

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 (0.1). 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

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

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

$$\operatorname{span}\begin{bmatrix}v_1 & v_2 & \cdots & v_n\end{bmatrix} = \operatorname{span} V_1 \cup \operatorname{span} V_2 \cup \cdots \cup \operatorname{span} V_\tau. \tag{0.2}$$

For the general application of rational-interpolation we assume that complex interpolation points are used, so as discussed in §0.1.5 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 $\mathcal{K}_{l_j}(\mathbf{H}(\sigma_j), \mathbf{R}(\sigma_j))$ 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 [21] avoid this issue by doing full, Arnoldi-style orthogonalization of an iterate against every previous vector, generating one orthogonal basis V for (0.2) directly. We provide an example of this as algorithm 2. Considering that for complex-valued interpolation points we will have to split the basis (0.2) anyway, thus losing any orthogonality, doing full orthogonalization of every vector is overkill.

Rational-Arnoldi method

```
Algorithm 2: Rational-Arnoldi
Input: State-space realization $(\boldsymbol{A}, \boldsymbol{E}, \boldsymbol{b}, \boldsymbol{c})$, interpolation points $\sigma_1, \sigma_2, \ldots, \sigma_\tau \in \mathbb{C}$, and number
       $l_1, l_2, \ldots, l_\tau$ of moments to match at each one.
Output: Basis $V$, where $\operatorname{span} V = \bigcup_{j=1}^{\tau} \mathcal{K}_{l_j}(\mathbf{H}(\sigma_j), \mathbf{r}(\sigma_j))$
$k := 0$
for $j = 1$ to $\tau$ do
    compute/factor $\mathbf{H}_j := (\boldsymbol{A} - \sigma_j \boldsymbol{E})^{-1}\boldsymbol{E}$ and $\mathbf{r}_j := (\sigma_j \boldsymbol{E} - \boldsymbol{A})^{-1}\boldsymbol{b}$
    $r_k := \mathbf{r}_j$
    for $i = 1$ to $k$ do   % make new start vector $\mathtt{r}_j$ orthogonal to previous $\{v_1, v_2 ..., v_k\}$
        $h_{ik} := v_k^H r_k$
        $r_k := r_k - h_{ik} v_j$      % $h_k$ becomes $k$-th column of $\widetilde{\mathtt{H}}$
    $v_{k+1} := r_k / \|r_k\|_2$
    while $k < l_1 + l_2 + \cdots + l_j$ do
        $k := k + 1$
        $r_k := \mathbf{H} v_k$
        for $i = 1$ to $k$ do      % make $r_k$ orthogonal to previous $\{v_1, v_2 ..., v_k\}$
            $h_{ik} := v_k^H r_k$
            $r_k := r_k - h_{ik} v_j$      % $h_k$ becomes $k$-th column of $\widetilde{\mathtt{H}}$
        if $\|r_k\|_2 \neq 0$ then
            $h_{k+1,k} := \|r_k\|_2$
            $v_{k+1} := r_k / \|r_k\|_2$
        else exit $k$-loop
return $\widetilde{\mathbf{H}} = \begin{bmatrix} h_{ij} \end{bmatrix}$, $V = \begin{bmatrix} v_1 & v_2 & \cdots & v_n \end{bmatrix}$, $r_n = v_{n+1} h_{n+1,n}$ where $n = l_1 + l_2 + \cdots + l_\tau$
```

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

$$\{\mathbf{H}(\sigma) = (\boldsymbol{A} - \sigma \boldsymbol{E})^{-1}\boldsymbol{E}\}. \tag{0.3}$$

Then

$$\mathbf{H}_2 z = T\mathbf{H}_1 z = \lambda T z + T r, \tag{0.4}$$

where T is the transformation

$$\begin{aligned} T(\sigma_1, \sigma_2) &:= (\boldsymbol{A} - \sigma_2 \boldsymbol{E})^{-1}(\boldsymbol{A} - \sigma_1 \boldsymbol{E}) \\ &= (\sigma_2 - \sigma_1)\mathbf{H}_2 + I. \end{aligned} \tag{0.5}$$

Then

$$T\mathbf{H}_1 = \Big((\boldsymbol{A} - \sigma_2 \boldsymbol{E})^{-1}(\boldsymbol{A} - \sigma_1 \boldsymbol{E})\Big)(\boldsymbol{A} - \sigma_1 \boldsymbol{E})^{-1}\boldsymbol{E} = \mathbf{H}_2$$

and

$$T\mathbf{R}_1 = \Big((\boldsymbol{A} - \sigma_2 \boldsymbol{E})^{-1}(\boldsymbol{A} - \sigma_1 \boldsymbol{E})\Big)(\sigma_1 \boldsymbol{E} - \boldsymbol{A})^{-1}\boldsymbol{B} = \mathbf{R}_2.$$

proof of (0.5). The expression (0.5) 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 (0.5) must be invertible, with

$$\big(T(\sigma_1, \sigma_2)\big)^{-1} = T(\sigma_2, \sigma_1) = (\sigma_1 - \sigma_2)\mathbf{H}_1 + I. \tag{0.6}$$

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 (0.5) and (0.6) 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

$$\mathbf{H}_2\mathbf{H}_1 = \frac{\mathbf{H}_2 - \mathbf{H}_1}{\sigma_2 - \sigma_1} \tag{0.7}$$

and for $\sigma_2 \neq \sigma_1$,

$$\mathbf{H}_1\mathbf{H}_2 = \mathbf{H}_2\mathbf{H}_1. \tag{0.8}$$

(0.8) shows that the set (0.3) of operators, commutes. (0.7) implies that

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

for what it's worth.

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

$$\mathcal{H}(s) = \boldsymbol{C}^T\big(I - (s - \sigma)\mathbf{H}(\sigma)\big)^{-1}\mathbf{R}(\sigma), \tag{0.9}$$

defined about the interpolation-point σ but independent of σ, involves a transformation of the form (0.5), since

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

Then we may re-write (0.9) 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 (0.4) that for eigen-pair (λ, z) of $\mathbf{H}_1$ we have

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

with the translation $T = T(\sigma_1, \sigma_2) = \Delta\mathbf{H}_2 + I$ from (0.5) 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

$$(1 - \lambda\Delta)\mathbf{H}_2 z = \lambda z + Tr. \tag{0.10}$$

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

$$\begin{aligned}\mathbf{H}_2 z - (\zeta\lambda)z &= \zeta Tr \\ &= \zeta(\Delta\mathbf{H}_2 + I)r\end{aligned} \tag{0.11}$$

We can also determine the residual vector for the approximate eigen-pair (μ, z) of the matrix pencil $(\boldsymbol{A}, \boldsymbol{E})$, where $\mu = 1/\lambda + \sigma_1$. Since $\mathbf{H}_1 z = (\boldsymbol{A} - \sigma_1\boldsymbol{E})^{-1}\boldsymbol{E}z = \lambda z + r$, note that

$$\boldsymbol{E}z = \lambda(\boldsymbol{A} - \sigma_1\boldsymbol{E})z + (\boldsymbol{A} - \sigma_1\boldsymbol{E})r.$$

Then

$$\begin{aligned}(\boldsymbol{A} - \mu\boldsymbol{E})z &= (\boldsymbol{A} - \sigma_1\boldsymbol{E})z - \frac{1}{\lambda}\boldsymbol{E}z \\ &= (\boldsymbol{A} - \sigma_1\boldsymbol{E})z - \frac{1}{\lambda}\Big(\lambda(\boldsymbol{A} - \sigma_1\boldsymbol{E}z + (\boldsymbol{A} - \sigma_1\boldsymbol{E})r)\Big) \\ &= -\frac{1}{\lambda}(\boldsymbol{A} - \sigma_1\boldsymbol{E})r \\ &= (\sigma_1 - \mu)(\boldsymbol{A} - \sigma_1\boldsymbol{E})r.\end{aligned}$$

Then

$$\|(\boldsymbol{A} - \mu\boldsymbol{E})z\| \leq \frac{\|\boldsymbol{E}\|}{\|\mathbf{H}_1\|}\frac{\|r\|}{|\lambda|}. \tag{0.12}$$

The bound (0.12) comes from observing that since $\mathbf{H}_1 = (\boldsymbol{A} - \sigma_1\boldsymbol{E})^{-1}\boldsymbol{E}$,

$$\|\mathbf{H}_1\| \leq \frac{\|\boldsymbol{E}\|}{\|\boldsymbol{A} - \sigma_1\boldsymbol{E}\|}.$$

Dividing (0.12) by $\|\mu\boldsymbol{E}z\| = |\mu|\|\boldsymbol{E}\|$ suggests the bound

$$\frac{\|r\|}{|\lambda||\mu|\,\|\mathbf{H}_1\|} = \frac{\|r\|}{(1 + |\lambda||\sigma_1|)\,\|\mathbf{H}_1\|} \tag{0.13}$$

for the generalized relative residual norm

$$\frac{\|(\boldsymbol{A} - \mu\boldsymbol{E})z\|}{\|\mu\boldsymbol{E}z\|}.$$

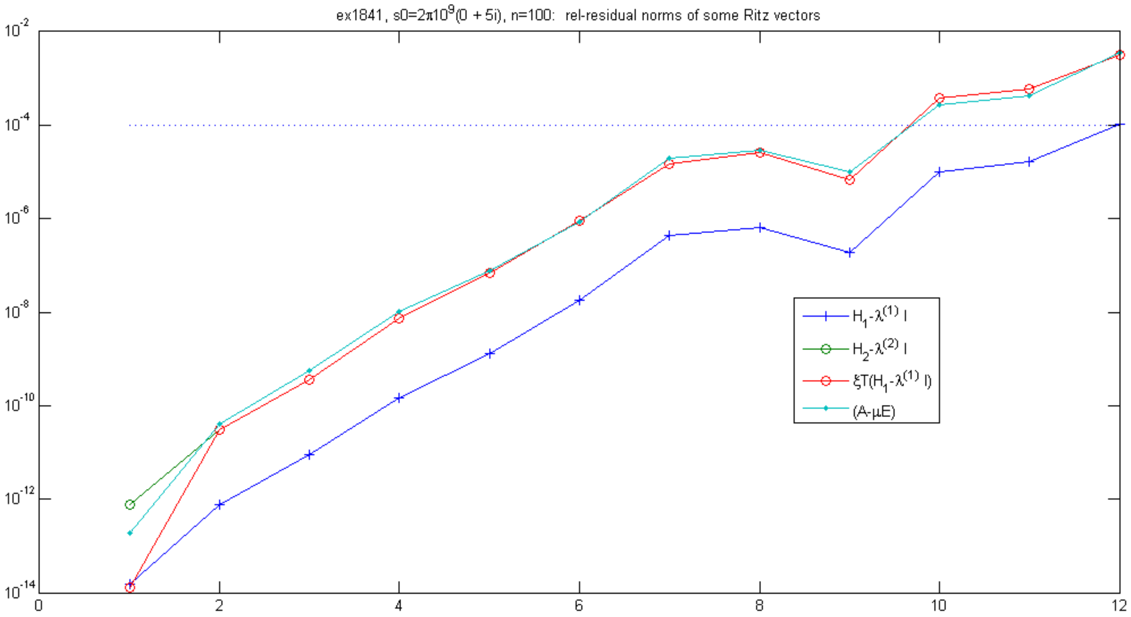


Figure 18: Here is a visual comparison of the norms from explicitly computed relative residuals $\|\mathbf{H}_1 z_j - \lambda_j z_j\|/|\lambda_j|$, compared with residuals $\|\mathbf{H}_2 z_j - \zeta\lambda_j z_j\|/|\zeta\lambda_j|$ (explicitly computed) of Z under $\mathbf{H}_2$, and shifted $\mathbf{H}_1$-residuals $\zeta T_{12}(\mathbf{H}_1 Z - \Lambda_1 Z)$ of (0.11). The example MIMO system in this case is `ex1841` with $m = p = 16$ inputs and outputs. The two shifts considered are $\sigma_1 = 10^{10}\pi i \in \mathbb{C}$ and $\sigma_2 = 10^{10}\pi \in \mathbb{R}$. The curve labeled $(\boldsymbol{A} - \mu\boldsymbol{E})$ is explicitly computed relative-residuals $\|\boldsymbol{A}z_j - \mu_j\boldsymbol{E}z_j\|/\|\boldsymbol{A}z_j\|$ of Z under $(\boldsymbol{A}, \boldsymbol{E})$, where $\mu_j = 1/\lambda_j + \sigma_1$.

0.3 Thick-Restart Method for MOR

In this section we generalize the Arnoldi method for MOR to one that uses multiple start vectors so that we can efficiently handle MIMO model reduction. Then, we will further generalize the band-Arnoldi process to one that takes multiple start vectors, and Ritz-vectors too! We will describe a MOR method that this thick restart makes possible. And we will show results from running this method on test models.

0.3.1 Band-Arnoldi algorithm

The precursor to the band-Krylov process was the "Lanczos-type method for multiple starting vectors" [3] in 2000, and an Arnoldi-specific version of that algorithm was given in [18]. The *band*-Arnoldi process of [19] (2003) is included here as algorithm 3. The band-Krylov algorithms of [19] are modified versions of [3, 18], most notably in the inclusion of the loop at line 19.

Band-Arnoldi differs from a block-Krylov process (like block-Arnoldi [33]) in that it cycles through a "band" of candidate vectors $\begin{bmatrix}\hat{v}_n & \hat{v}_{n+1} & \ldots & \hat{v}_{n+m_c}\end{bmatrix}$, 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 $\dot{V} = \begin{bmatrix}\dot{v}_1 & \dot{v}_2 & \ldots & \dot{v}_d\end{bmatrix}$, remaining candidate vectors $\hat{V} = \begin{bmatrix}\hat{v}_{n+1} & \hat{v}_{n+2} & \ldots & \hat{v}_{n+m_c}\end{bmatrix}$, and Rayleigh-quotient $\widetilde{\mathbf{H}} = V^H\mathbf{H}V$ that satisfy the band-Arnoldi relation

$$\mathbf{H}V = V\widetilde{\mathbf{H}} + \begin{bmatrix}(I - VV^H)\dot{V} & \hat{V}\end{bmatrix} F \tag{0.1}$$

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 (0.1).

In addition, algorithm 3 returns $\widetilde{\boldsymbol{\rho}}$, and $\widetilde{\boldsymbol{\rho}}^{\text{defl}}$ where

$$\mathbf{R} = V\widetilde{\boldsymbol{\rho}} + \widetilde{\boldsymbol{\rho}}^{\text{defl}}, \tag{0.2}$$

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

Candidates/residual term

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

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

$$\begin{aligned} \dot{V}F_1 = \hat{V}^{\text{defl}} &= \begin{bmatrix} 0 & \hat{v}_2^{\text{defl}} & 0 & 0 & \hat{v}_5^{\text{defl}} & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\ &= \begin{bmatrix} 0 & \dot{v}_1 & 0 & 0 & \dot{v}_2 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\ &= \begin{bmatrix} \dot{v}_1 & \dot{v}_2 \end{bmatrix} \begin{bmatrix} e_2^T \\ e_5^T \end{bmatrix}. \end{aligned} \tag{0.3}$$

The band-Arnoldi algorithm deflates a candidate vector $\hat{v}_j$ (i.e. removes it from further iterations) if $\|\hat{v}_j\| \leq \text{dtol}$[11] 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 3 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

$$V^H \hat{v}_j^{\text{defl}} = \begin{bmatrix} 0 & 0 & \cdots & 0 & v_{j+1}^H \hat{v}_j^{\text{defl}} & v_{j+2}^H \hat{v}_j^{\text{defl}} & \cdots & \hat{v}_j^H \hat{v}_j^{\text{defl}} \end{bmatrix}^T. \tag{0.4}$$

(0.4) 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, (0.4) is included in the Rayleigh-quotient $\widetilde{\mathbf{H}}$ as the j-th column of $\widetilde{\mathbf{H}}_{\mathcal{E}}$. Then

$$\|\widetilde{\mathbf{H}}_{\mathcal{E}}\| = \|V^H \dot{V}\|_F \leq \|\dot{V}\|_F \leq \text{dtol}\sqrt{d}, \tag{0.5}$$

and

$$\|(I - VV^H)\dot{V}\|_F \leq \|\dot{V}\|_F. \tag{0.6}$$

$\widetilde{\boldsymbol{\rho}}^{\text{defl}}$ in (0.2) 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 (0.1) and setting $Y = VS$ gives the block residual-norm bound

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

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

[11] [41] suggests dtol $= \sqrt{\epsilon}$, where machine-$\epsilon = 2^{-52} \approx$ 2.22e-16 in double-precision (64-bit) floating point arthmetic in IEEE 754 - 2008 standard.

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,

$$\begin{aligned} \|\mathbf{H}z_j - z_j\lambda_j\|_2^2 &\leq d\,\varepsilon_M + \|\hat{V}F_2 w_j\|_2^2 \\ &= d\,\varepsilon_M + \|\hat{V}\tilde{w}_j\|_2^2. \end{aligned} \tag{0.7}$$

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

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

$$\frac{\|\mathbf{H}z_j - z_j\lambda_j\|}{\|\lambda_j z_j\|} \tag{0.8}$$

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,

$$\left\|\hat{V}\right\| \frac{\|\tilde{w}_j\|}{|\lambda_j|} \tag{0.9}$$

is an estimate of relative residual norm, as is

$$\frac{1}{|\lambda|} \begin{bmatrix} \|\hat{v}_1\| & \|\hat{v}_2\| & \cdots & \|\hat{v}_{m_c}\| \end{bmatrix} \begin{bmatrix} |\tilde{w}_j^{(1)}| \\ |\tilde{w}_j^{(2)}| \\ \vdots \\ |\tilde{w}_j^{(m_c)}| \end{bmatrix}. \tag{0.10}$$

Both (0.9) and (0.10) are cheaper to compute than norms $\|\hat{V}\tilde{w}_j\|$ of potentially large matrix-vector products, but (0.10) might be better if $\hat{V}F_2$ has rank greater than one. Estimates (0.9) and (0.10) are equal for rank-1 residuals.

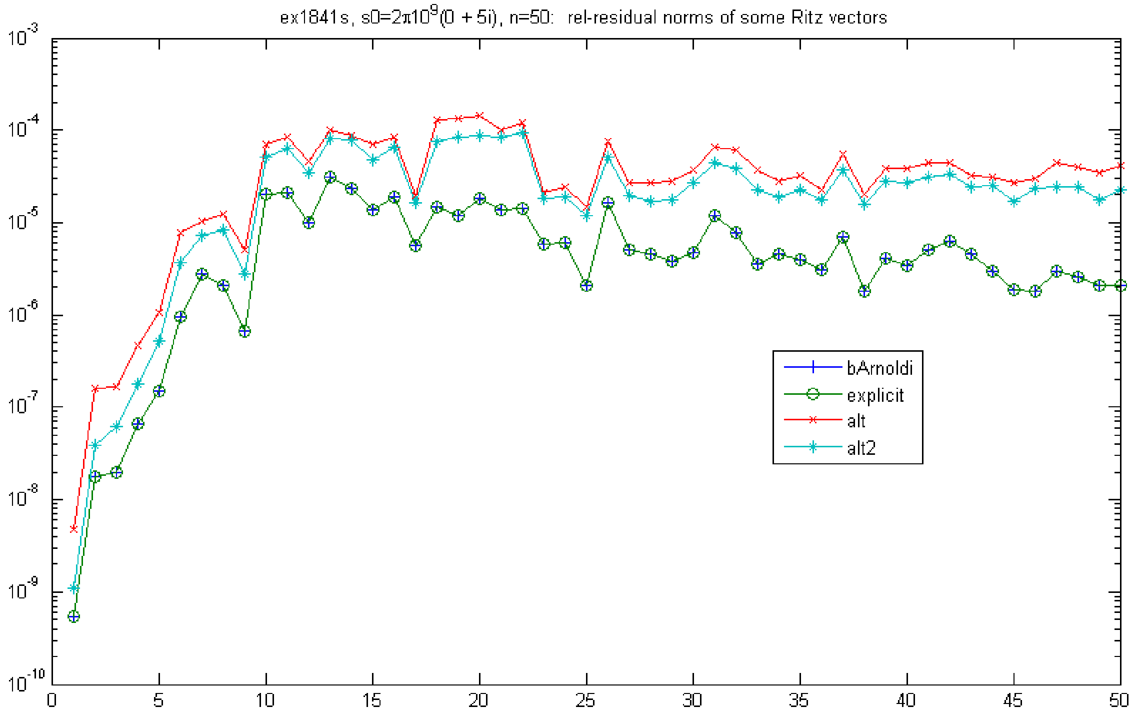
Figure 19 gives a comparison of relative residual-norm estimates for two MIMO models. The curve labeled `bArnoldi` was obtained by computing $\|\hat{V}\tilde{w}_j\| / |\lambda_j|$ for each Ritz-pair, and `explicit` is explicit computation of (0.8). The explicit computation of relative Ritz-residual norms is less accurate for more converged Ritz-values because we are limited to double precision arithmetic. The curve labeled `alt` was obtained via (0.9) and `alt2` is (0.10).

proof of band-Arnoldi relation (0.1) Before going on to the thick-restart, (0.1) may not be obvious so we will show why it is true.

Proof. The precursor to band-Arnoldi well-documented in [18, 3] computes V_n, T_n, and $\hat{V}_n^{\text{defl}}$ so that

$$\mathbf{H}V_n = V_n T_n + \hat{V}_n^{\text{defl}} + \begin{bmatrix} 0 & \hat{V}_n \end{bmatrix} \qquad \in \mathbb{C}^{N \times n}, \tag{0.11}$$

where $\hat{V}$ is the matrix of candidates for the next $n + m_c$ basis vectors. The deflation matrix $\hat{V}_n^{\text{defl}}$ is a zero or mostly-zero matrix with very small norm, consisting of deflated ex-candidate vectors and $\|\hat{V}_n^{\text{defl}}\|_F \leq$ `dtol`$\sqrt{m - m_c}$. The remaining candidate vectors $\hat{V}_n$ are orthogonal to V_n. T_n is nearly a Rayleigh-quotient



(a) $n = 50$ ROM of MIMO example model `ex1841`, with $m = p = 16$ inputs and outputs

Figure 19: These are plots of estimated relative residual norms of Ritz-pairs for two example ROMS. Both ROMs were obtained via algorithm 3 (band-Arnoldi) with one expansion point $\sigma = \pi{\cdot}10^{10}i$. The curve labeled `bArnoldi` was obtained by computing $\|\hat{V}\tilde{w}_j\|/|\lambda_j|$ for each Ritz-pair, and `explicit` is explicit computation of (0.8). The explicit computation of relative Ritz-residual norms tends to be less accurate for more converged Ritz-values because we are limited to double precision arithmetic. The curve labeled `alt` was obtained via (0.9) and `alt2` is (0.10). The accuracy of the estimates is somewhat system-specific. A typical convergence tolerance ctol is $\sqrt{\varepsilon_M} \approx 10^{-8}$.

but not quite, because $\hat{V}_n^{\text{defl}}$ and V_n are not orthogonal. Specifically, basis vectors $v_{j+1}, v_{j+2}, \ldots$ are not constructed to be orthogonal to $\hat{v}_j^{\text{defl}}$.

The Rayleigh-quotient of $\mathbf{H}$ under V_n is the perturbed matrix

$$\widetilde{\mathbf{H}}_n := V_n^H \mathbf{H} V = T_n + V^H \hat{V}_n^{\text{defl}}. \tag{0.12}$$

Inclusion of the loop at line (19) of algorithm 3 (band-Arnoldi) computes and adds $V^H \hat{V}_n^{\text{defl}}$, thus constructing Rayleigh-quotient (0.12).

If we re-write (0.11) with $\widetilde{\mathbf{H}}$ instead of T_n, we get

$$\mathbf{H}V_n = V_n \left(\widetilde{\mathbf{H}}_n - V_n^H \hat{V}_n^{\text{defl}} \right) + \hat{V}_n^{\text{defl}} + \begin{bmatrix} 0 & \hat{V}_n \end{bmatrix} \tag{0.13}$$

$$= V_n \widetilde{\mathbf{H}}_n + (I - V_n V_n^H) \hat{V}_n^{\text{defl}} + \begin{bmatrix} 0 & \hat{V}_n \end{bmatrix} \tag{0.14}$$

□

0.3.2 Implicit thick-restarting the Band-Arnoldi process

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

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

for $j = 1, 2, ..., \ell$. If we run the band-Arnoldi algorithm with $\mathbf{H}$ and start block $\begin{bmatrix} Z & \mathbf{R} \end{bmatrix}$ where $Z = \begin{bmatrix} z_1 & z_2 & \cdots & z_\ell \end{bmatrix}$ and $\mathbf{R} = \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \cdots & \mathbf{r}_m \end{bmatrix}$, then after ℓ iterations we have orthonormal basis $V_\ell = \begin{bmatrix} v_1 & v_2 & \cdots & v_\ell \end{bmatrix}$ for span Z where

$$u_{jj} v_j = (I - V_{j-1} V_{j-1}^H) z_j$$

for $j = 1, 2, \ldots, \ell$. $u_{jj} = \|z_j\|_2 \in \mathbb{R}$ is the diagonal element of the upper-triangular Rayleigh-quotient U, where $Z = VU$ can be regarded as a QR factorization. Assuming unit Ritz-vectors, the next ℓ candidate vectors are

$$\begin{aligned} \hat{v}_{\ell+m+j} &= (I - V_j V_j^H) \mathbf{H} v_j \\ &= (I - V_j V_j^H) \mathbf{H} (I - V_{j-1} V_{j-1}^H) z_j \\ &= (I - V_j V_j^H) \mathbf{H} z_j, \quad \text{because } \mathbf{H} V_{j-1} \subset \operatorname{span} V_j \\ &= (I - V_j V_j^H)(\lambda_j z_j + \gamma_j) \\ &= (I - V_j V_j^H) \gamma_j \quad \text{because } z_j \in \operatorname{span} V_j. \end{aligned} \tag{0.15}$$

for $j = 1, 2, ..., \ell$. Then the candidate vectors at that point are

$$\hat{V}_\ell = \begin{bmatrix} (I - V_\ell V_\ell^H) \mathbf{R} & \hat{v}_{\ell+m+1} & \hat{v}_{\ell+m+2} & \ldots & \hat{v}_{2\ell+m} \end{bmatrix}. \tag{0.16}$$

Note that $(I - V_j V_j^H) \gamma_j = \gamma_j$ if the residual γ_j is orthogonal to $\{z_1, z_2, ..., z_j\}$, which is the case when the basis Z of Ritz-vectors came from one cycle of a Krylov process such as algorithm 3. If we consider restarts however, the Ritz-vectors from each restart will have a different orthogonal residual, so the set of Ritz-vectors will not be orthogonal to the set of residuals in general.

It would be wasteful to actually compute $HV_\ell = V_\ell U + \hat{V}$ via algorithm 3 since we already know (0.16). Instead, let $Y := V_\ell$ be the orthonormal basis for the collection of Ritz-vectors Z, and let $\hat{Y} := \hat{V}$ be the set of candidate vectors defined by (0.15) and (0.16).

Now let's assume we have a general band-Krylov-Schur relation (not orthogonal in general), with basis $Y = \begin{bmatrix} y_1 & y_2 & \cdots & y_\ell \end{bmatrix} \in \mathbb{C}^{N\times\ell}$ for subspace $\mathcal{Y}$, and with basis $\hat{Y} = \begin{bmatrix} \hat{y}_1 & \hat{y}_2 & \cdots & \hat{y}_\nu \end{bmatrix}$ (with $\nu \le \ell$) for the residual subspace so that

$$\begin{aligned} \mathbf{H}Y &= \begin{bmatrix} Y & \hat{Y} \end{bmatrix} \begin{bmatrix} U \\ B \end{bmatrix} \\ &= YU + \hat{Y}B. \end{aligned} \tag{0.17}$$

Then

$$\mathbf{H}Y = YU' + (I - YY^H)\hat{Y}B \tag{0.18}$$

with $U' = U + Y^H\hat{Y}$ is an orthogonal relation. If U is upper triangular then U' is no longer upper triangular, but Y is usually chosen so that the norm $\|Y^H\hat{Y}\|$ of the perturbation is small.

If we run band-Arnoldi with $\mathbf{H}$ on start block $\begin{bmatrix} Y & \mathbf{R} \end{bmatrix}$ instead of just $\mathbf{R}$, after the first ℓ iterations we have Arnoldi basis $V = Y$ and candidate vectors $\hat{V} = \begin{bmatrix} \hat{v}_{\ell+1} & \hat{v}_{\ell+2} & \cdots & \hat{v}_{2\ell} \end{bmatrix}$ such that $\operatorname{span}\{\hat{V}\} = \operatorname{span}\{\hat{Y}B\}$. If $\|\hat{Y}B\| \le \mathrm{dtol}\sqrt{\ell}$ then candidates $\hat{v}_{\ell+1}, \hat{v}_{\ell+2}, \ldots, \hat{v}_{2\ell}$ will get deflated by the standard band-Arnoldi process, i.e. they become $\hat{v}^{\mathrm{defl}}_{\ell+1}, \hat{v}^{\mathrm{defl}}_{\ell+2}, \ldots, \hat{v}^{\mathrm{defl}}_{2\ell}$.

If $\hat{Y}$ is not deflatable ($\|\hat{Y}B\| > \mathrm{dtol}\sqrt{\ell}$) we can manually deflate it as follows:

- Set $V := Y$, $\widetilde{\mathbf{H}} := U'$,
- Set $\begin{bmatrix} \hat{v}^{\mathrm{defl}}_{\ell+1} & \hat{v}^{\mathrm{defl}}_{\ell+2} & \cdots & \hat{v}^{\mathrm{defl}}_{2\ell} \end{bmatrix} := \hat{Y}$
- Set $\hat{V} := \mathbf{R}$, $m_c := m = \dim \mathbf{R}$

and continue band-Arnoldi iterations from step $\ell + 1$. The resulting relation after n iterations for a total of $n + \ell$, is

$$\mathbf{H}\begin{bmatrix} Y & V \end{bmatrix} = \begin{bmatrix} Y & V \end{bmatrix} \begin{bmatrix} U' & G \\ \mathcal{E} & \hat{H} \end{bmatrix} + \begin{bmatrix} (I - YY^H)\hat{Y} & & \\ (I - VV^H)\hat{Y} & (I - VV^H)\dot{V} & \hat{V} \end{bmatrix} \begin{bmatrix} B & & \\ & F_1 & \\ & F_2 & \end{bmatrix} \tag{0.19}$$

where $\dot{V}$, $\hat{V}$, and F are defined as in (0.1). The $(\ell+n)\times(\ell+n)$ Rayleigh-quotient

$$\widetilde{\mathbf{H}} := \begin{bmatrix} U' & G \\ \mathcal{E} & \hat{H} \end{bmatrix} \tag{0.20}$$

includes the block $\mathcal{E} = V^H\hat{Y}B$.

In the thick-restart [52] and Krylov-Schur [55] schemes for single-vector iterations, they assume $\|\mathcal{E}\| \le \|\hat{Y}B\|$ is small enough to be negligible, so that $U' \approx U$ is upper-triangular. Also, they restart with the residual from a previous cycle, which in our case would be $\hat{Y}$. In that case $\hat{Y}$ would be included in V and $\mathcal{E}$ would be zero. Those methods then take advantage of the upper-triangular structure of the leading $\ell\times\ell$ principle submatrix U of (0.20). For this analysis we assume $\hat{Y}$ gets deflated so that V and $\hat{Y}$ are not orthogonal.

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

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

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

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

Let

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

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

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

$$\|\mathbf{H} z_j - \lambda_j z_j\| \leq |f^T w_j|. \tag{0.21}$$

Note that

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

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

$$\mathrm{rr}_j = \frac{|f^T w_j|}{\|\mathbf{H}\|_{\mathrm{est}}} \leq \mathrm{ctol}. \tag{0.22}$$

$\|\mathbf{H}\|_{\mathrm{est}} = \max_v \|\mathbf{H}v\|$ is an estimated operator-norm of $\mathbf{H}$ obtained during iterations. A value suggested by [41] is $\mathrm{ctol} = \sqrt{\epsilon}$, which is the same value used for dtol in [19].

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

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

0.3.3 Implicitly-projected ROM from a thick-restarted bArnoldi process

Recall that the bArnoldi algorthm

$$\begin{bmatrix} V & \widetilde{\mathbf{H}} & \widetilde{\boldsymbol{\rho}} \end{bmatrix} = \mathbf{bArnoldi}(\mathbf{H}, \mathbf{R}) \tag{0.23}$$

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

$$\widetilde{\mathcal{H}}(s) = (\boldsymbol{C}^T V)\big(I - (s-\sigma)\widetilde{\mathbf{H}}\big)^{-1} \underbrace{(V^H \mathbf{R})}_{\widetilde{\boldsymbol{\rho}}} \tag{0.24}$$

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

$$\begin{bmatrix} V & \widetilde{\mathbf{H}} & \widetilde{\boldsymbol{\rho}} \end{bmatrix} = \mathbf{bArnoldi}(\mathbf{H}, \begin{bmatrix} Y & \mathbf{R} \end{bmatrix}) \tag{0.25}$$

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

$$\widetilde{\mathcal{H}}(s) = \big(\boldsymbol{C}^T \begin{bmatrix} Y & V' \end{bmatrix}\big) \left(I - (s-\sigma)\widetilde{\mathbf{H}}\right)^{-1} \underbrace{\left(\begin{bmatrix} Y & V' \end{bmatrix}^H \mathbf{R}\right)}_{\widetilde{\boldsymbol{\rho}}_2} \tag{0.26}$$

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.

0.3.4 Thick-restarted Band-Arnoldi Algorithm for MOR

The explicit thick-restarted Band-Arnoldi algorithm is given as algorithm 4. It consists of restarting the band-Arnoldi algorithm (algorithm 3) with Ritz-vectors. In practice (for large N), an implicit-restart method like Krylov-Schur[55] would be used; also discussed in §0.3.2.

Selection of a new interpolation point (line 9) 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.

Algorithm 3: 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 (0.1)
$\hat{v}_i := \mathbf{r}_i$   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)
        $\hat{v}^{\mathrm{defl}}_{n-m_c} := \hat{v}_n$
        $\mathcal{I} = \mathcal{I} \cup \{n - m_c\}$ % 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
        $\hat{v}_j := \hat{v}_{j+1}$   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
        $h_{nj} := v_n^H \hat{v}_j^{\mathrm{defl}}$
return $V$, $\widetilde{\mathbf{H}}$, $\widetilde{\boldsymbol{\rho}} = \left[h_{ij}\right]_{i=1,2,\ldots,m}^{j=1-m\ldots,1,0}$ , $\hat{V}$, $\hat{V}^{defl}$, $\widetilde{\boldsymbol{\rho}}^{defl} = \left[\hat{v}_j^{defl}\right]$ , $j = 1 - m, ..., 1, 0$,
```

Algorithm 4: Explcit thick-restarted Band-Arnoldi cycle

```
Input: System realization $(\boldsymbol{A}, \boldsymbol{E}, \boldsymbol{B}, \boldsymbol{C})$, initial interpolation point $\sigma_1 \in \mathbb{C}$.
Set $Z_0 = \{\}, \widehat{V} = \{\}$
for $j = 1, 2, \dots$ do
    Let $\mathbf{H}_j := (\boldsymbol{A} - \sigma_j \boldsymbol{E})^{-1}\boldsymbol{E}$ and $\mathbf{R}_j := (\sigma_j \boldsymbol{E} - \boldsymbol{A})^{-1}\boldsymbol{B}$
    Compute $\left(V, \hat{V}, \dot{V}, \widetilde{\mathbf{H}}, \widetilde{\boldsymbol{\rho}}\right) :=$ bArnoldi $\left(n_j, \mathbf{H}_j, \begin{bmatrix} Z_{j-1} & \mathbf{R}_j \end{bmatrix}\right)$ that satisfies (0.1).
    Set $\widehat{V}_j := \begin{bmatrix} \widehat{V}_{j-1} & V_{\mathbf{R}_j} \end{bmatrix}$, where $V = \begin{bmatrix} Y_{j-1} & V_{\mathbf{R}_j} \end{bmatrix}$. % ($Y_{j-1}$ is the orthogonalization of
      $Z_{j-1}$.)
    % The $j$-th implicitly projected ROM transfer function is given by (0.26).
    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 $(\lambda_i, z_i)$ where $z_i = \begin{bmatrix} Y & V \end{bmatrix} w_i$ is given by (0.22).
    Compute pole-weights $\gamma_1, \gamma_2, \dots, \gamma_{n_j}$ as (0.40) and (0.41).
    Let $Z_j$ consist of converged Ritz-vectors and those with large relative-weight $|\gamma_i|/\Sigma|\gamma_i|$.
    Select new interpolation-point $\sigma_{j+1}$.
Output: Basis $\widehat{V}$ for $\bigcup_j \mathcal{K}_{n_j}(\mathbf{H}_j, \mathbf{R}_j)$.
```

0.4 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

$$\frac{\|\mathcal{H}-\widetilde{\mathcal{H}}\|}{\|\mathcal{H}\|} \leq \texttt{tf\_tol} = 0.01, \tag{0.1}$$

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 20. 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 $(\boldsymbol{A}, \boldsymbol{E}, \boldsymbol{B}, \boldsymbol{C})$.

0.4.1 ex308

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

ex308 Benchmarks

Benchmark data for `ex308` is given in table 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

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

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.

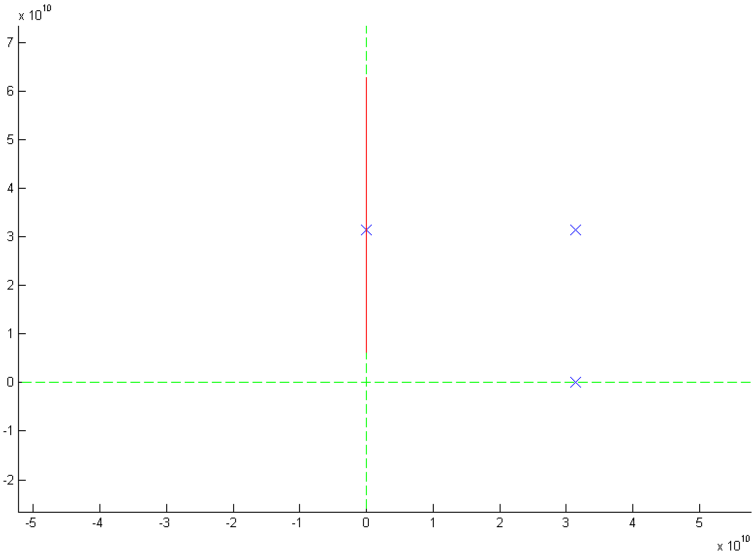


Figure 20: The three interpolation points used for single point benchmarks. The segment of interest $[10^9, 10^{10}]i$ on the $\Im$-axis, is highlighted. Note how small $[0, 10^9]$ is, in comparison.

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	23
$i\pi 10^{10}$	71	142	0.992958	5.913e-4	30,177,840 + M	24
$(1+i)\pi 10^{10}$	97	194	0.793814	5.1713e-3	42,782,432 + M	25

Table 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.

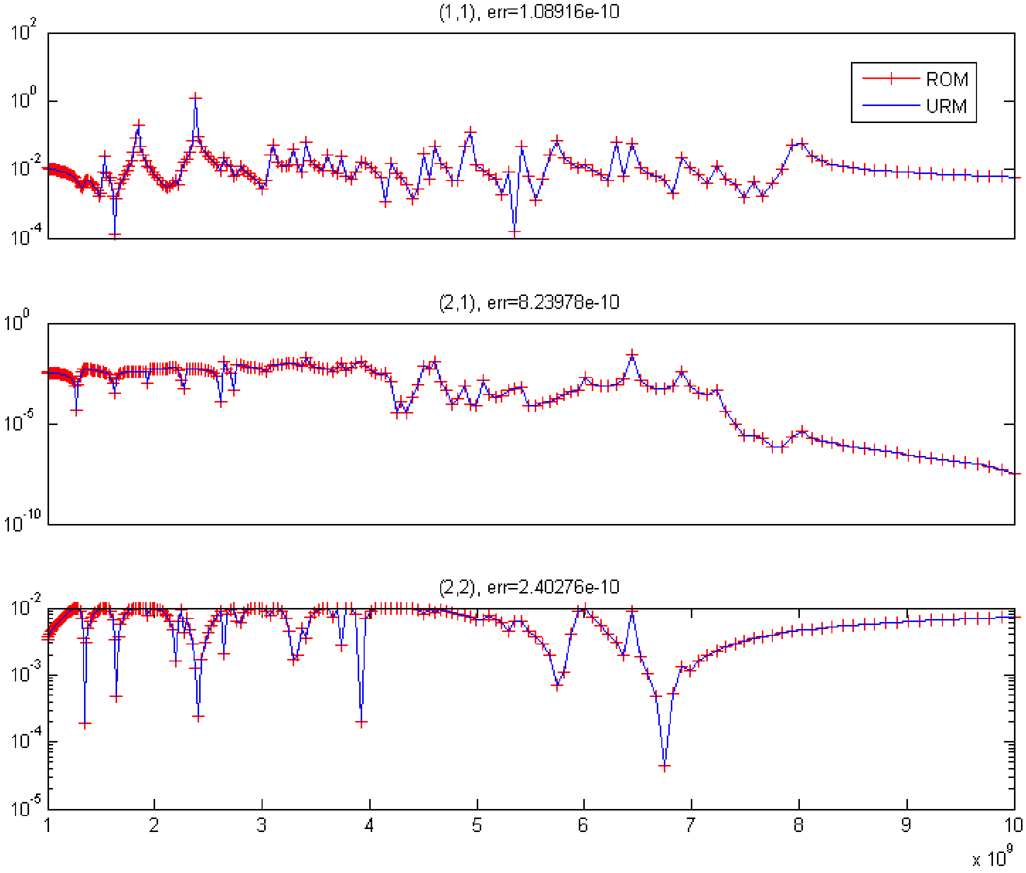


Figure 21: These are the three unique gain plots for `ex308`.

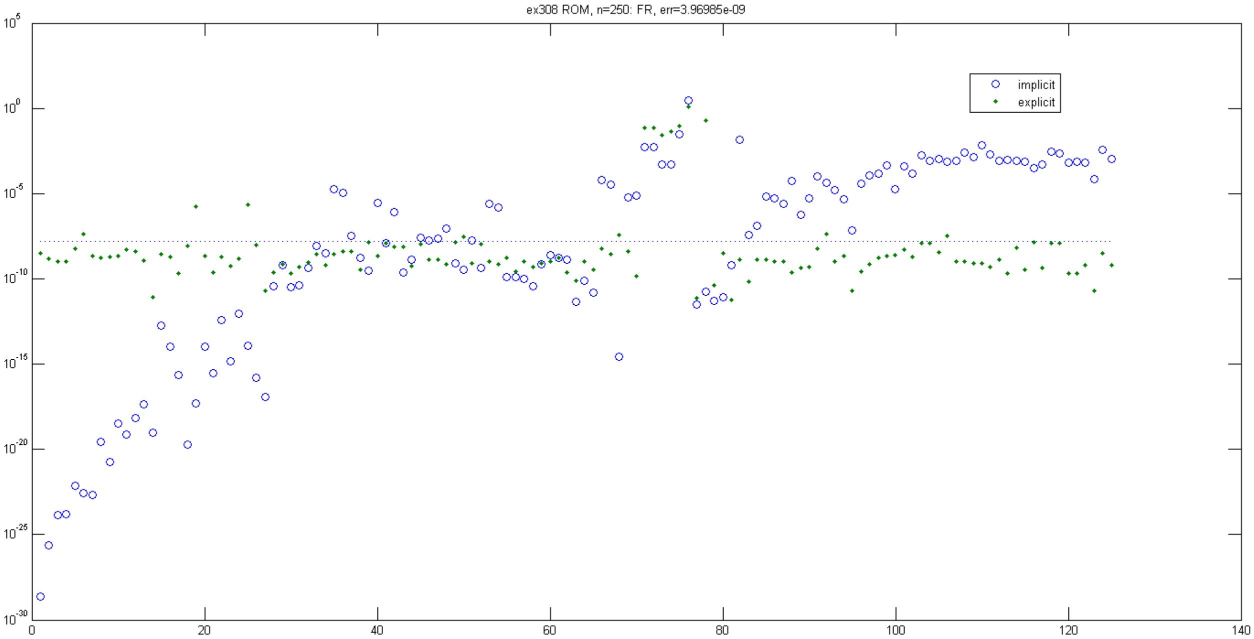


Figure 22: This is a plot of relative-residual errors for the 250 poles of an $n = 250$ ROM (about $s_0 = (1+i)\pi 10^{10}$) of `ex308`. The circles are the poles derived from eigenvalues of $\widetilde{\mathbf{H}} = V^H \mathbf{H} V$ (the implicit ROM), and the dots are the eigenvalues of the explicitly 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 (0.21) for the residual norm $\|\boldsymbol{A}z - \mu \boldsymbol{E}z\|$.

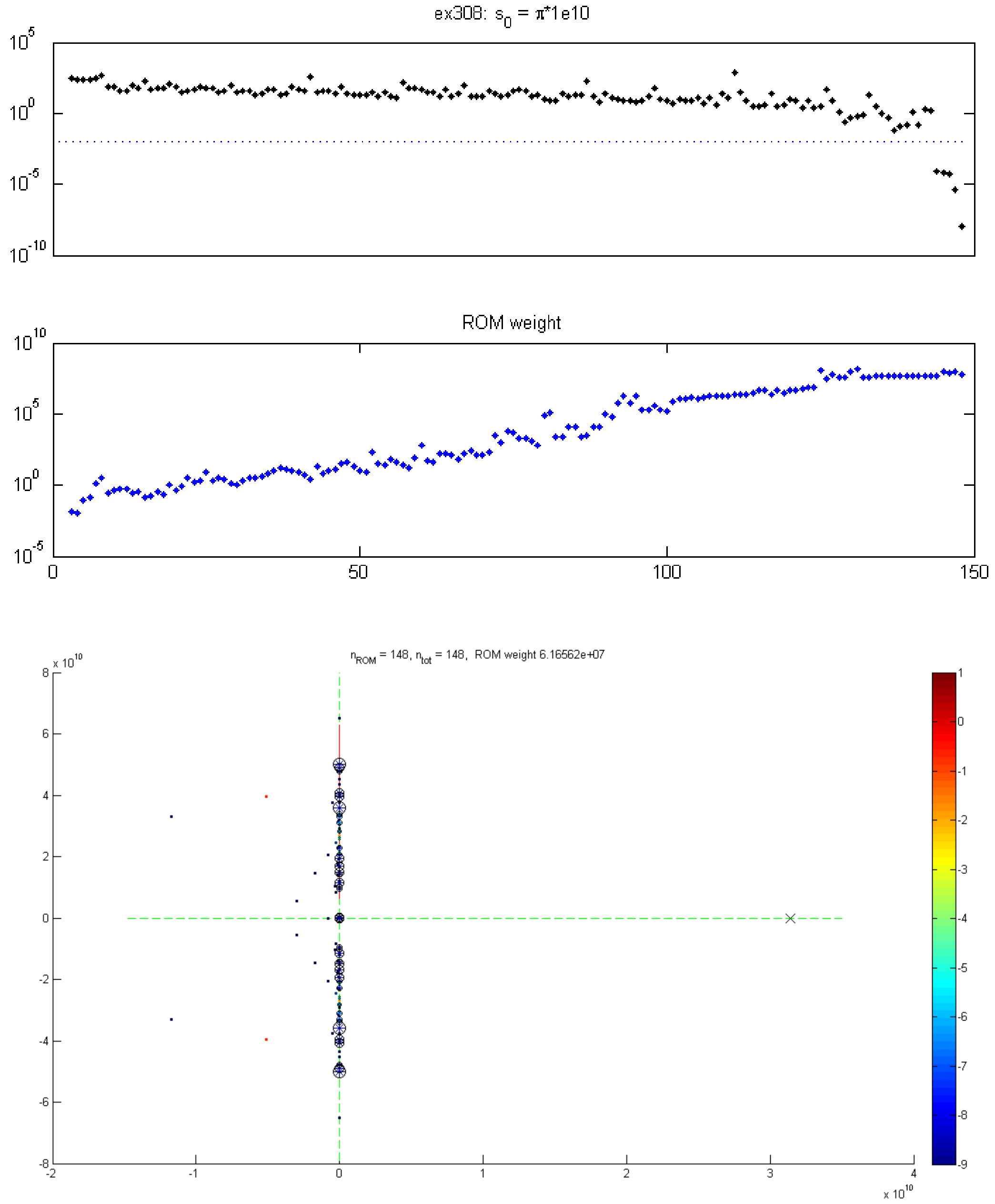


Figure 23: Transfer function relative-error (0.1) (of explicitly projected ROM) and ROM weight (of the implicitly projected ROM) vs. n for `ex308`, at real interpolation-point $s_0 = \pi 10^{10} \in \mathbb{R}$, indicated by '×'. The dotted line in the first plot represents a relative-error of 0.01. `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 $\approx 10^{-8}$.

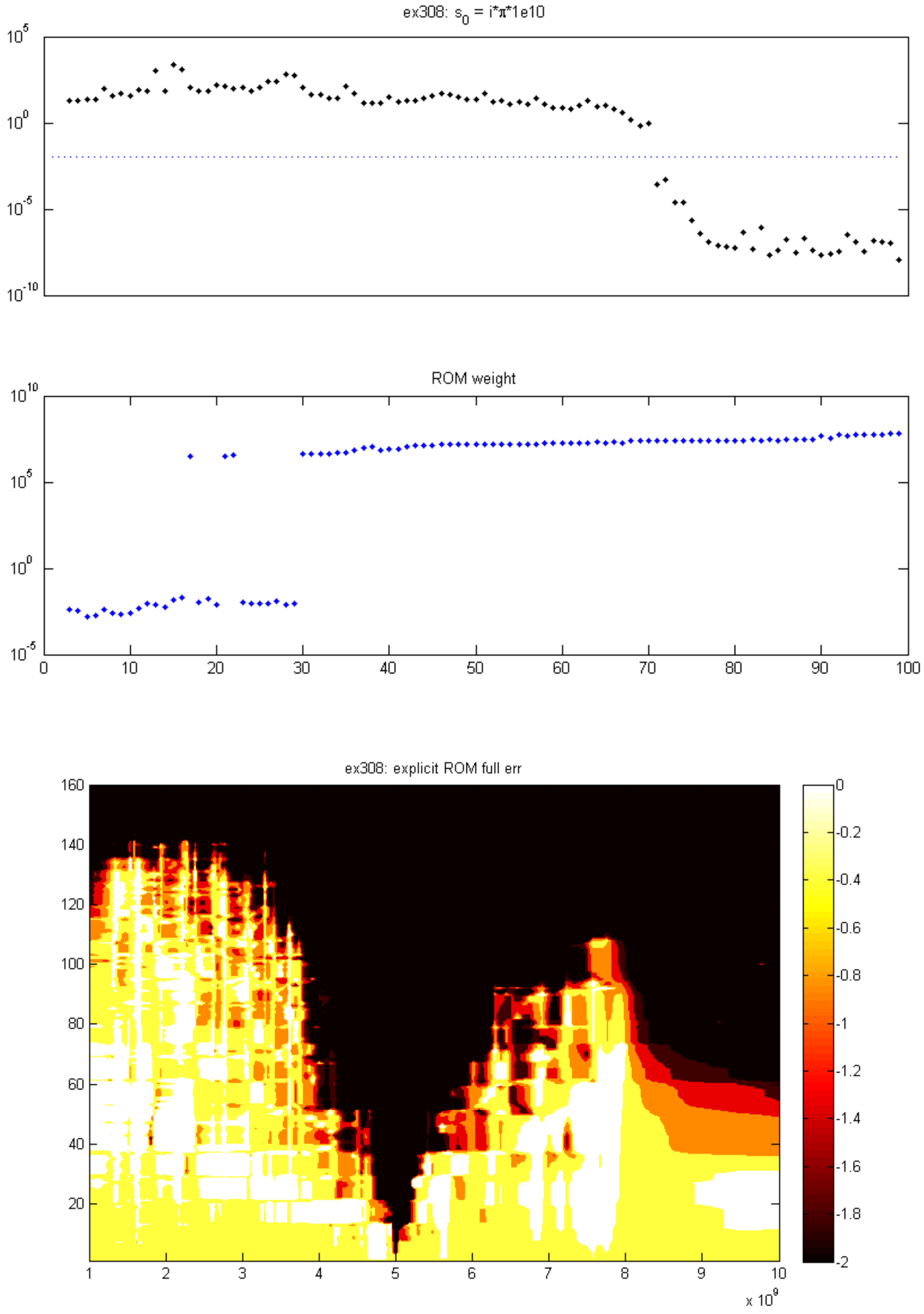


Figure 24: Transfer function relative-error and ROM weight vs. n for `ex308`, at $\Im$ interpolation-point $s_0 = i\pi \cdot 10^{10} \in i\mathbb{R}$. Unfortunately it appears that ROM weight is not a consistently reliable indicator of transfer function convergence when using a single interpolation point. 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.

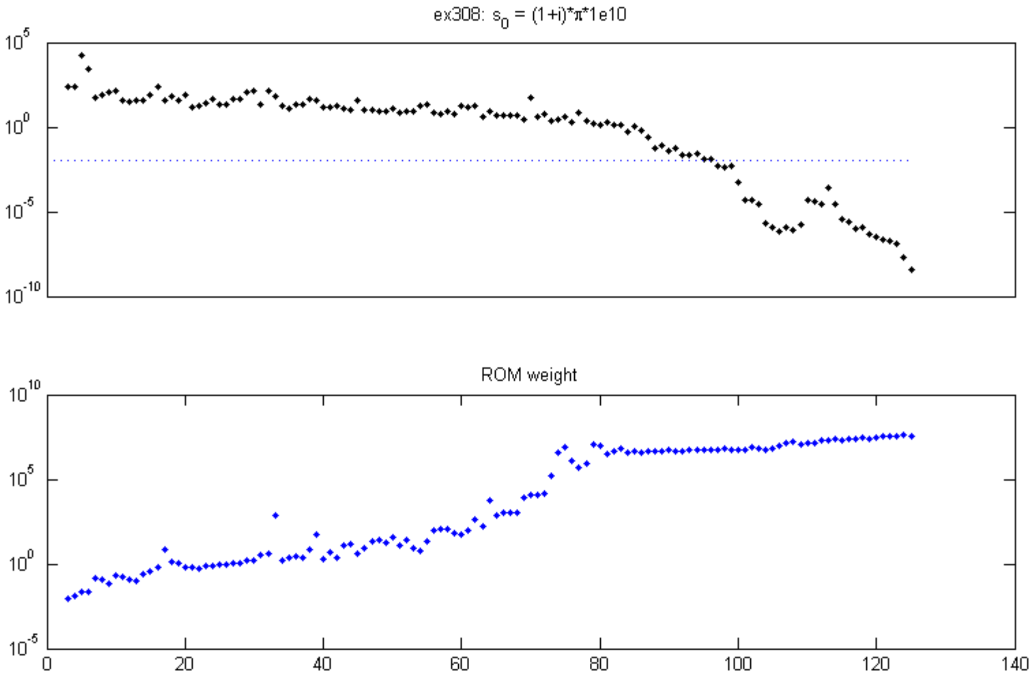


Figure 25: Transfer function relative-error and ROM-weight vs. n for `ex308`, at $s_0 = (1 + i)\pi \cdot 10^{10}$. This is much the way we would like the relationship between ROM-weight and transfer-function error to look. ROM-weight leveling-off would indicate transfer-function error convergence.

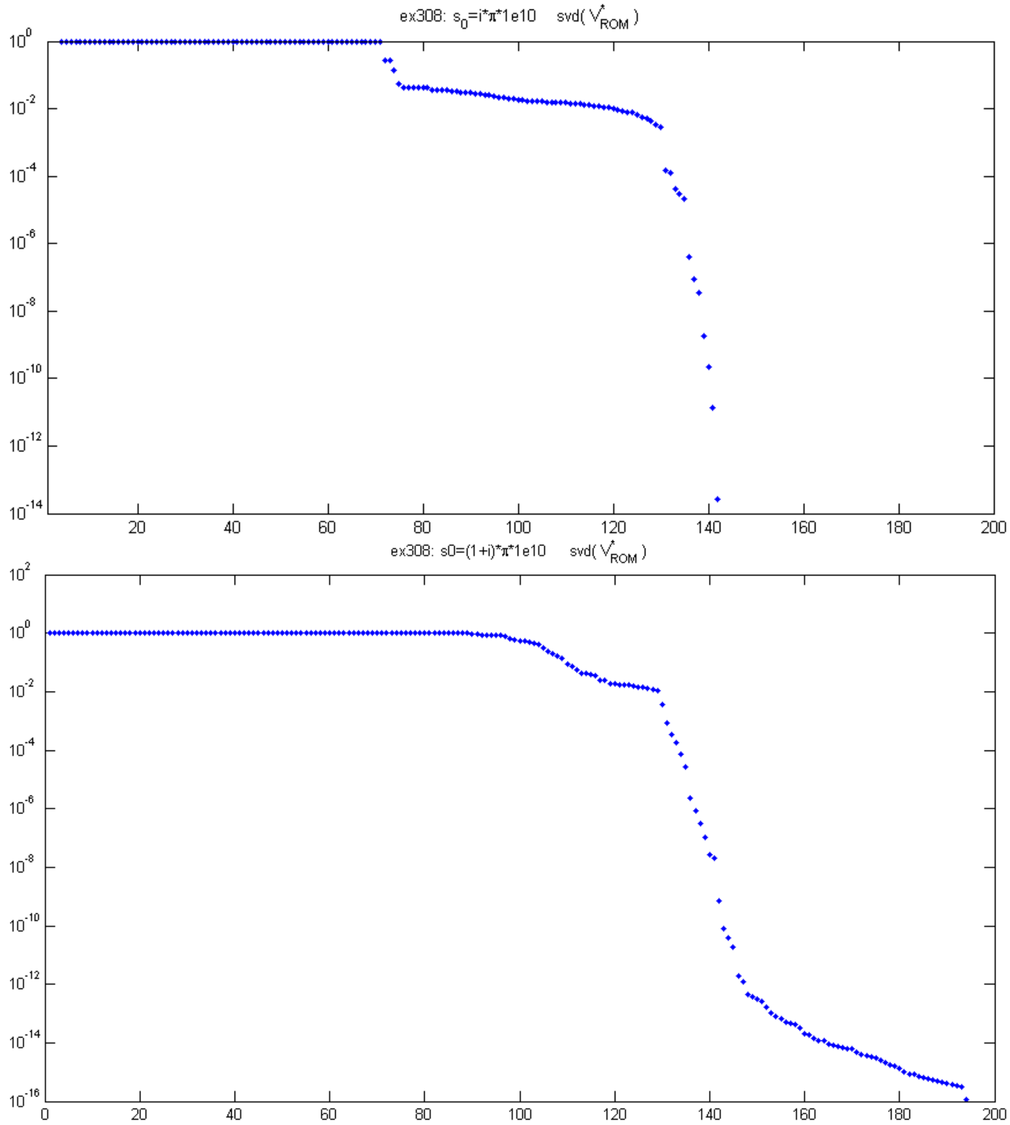


Figure 26: These are plots of singular values of the split basis matrix V^* that spans $\Re(V) \cup \Im(V)$ (before being orthogonalized), for bases generated about $s_0 = i\pi \cdot 10^{10}$ and $(1+i)\pi \cdot 10^{10}$. The basis for the ROM expanded about $i\pi \cdot 10^{10}$ has effective rank 141, 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.

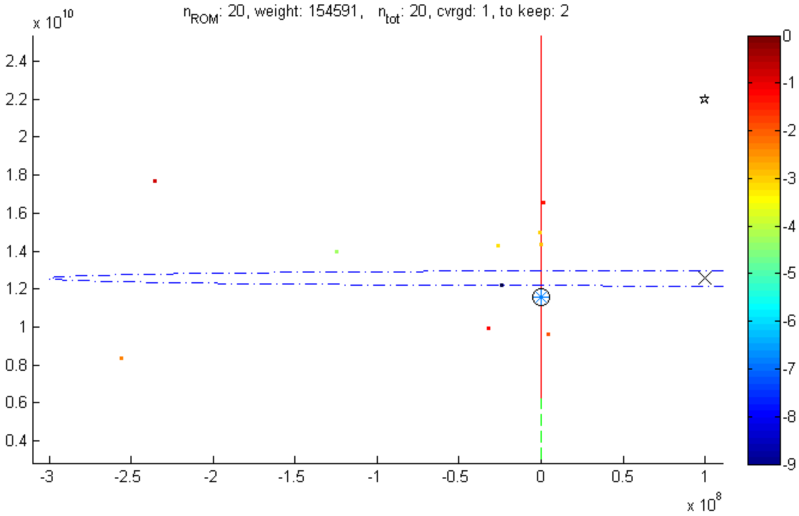
The resulting ROM required a total of 70 iterations (not including re-processing thick-restarting Ritz-vectors), was of size $n' = 140$ and had a relative-error of `5.40491e-05`, making it compare favorably with the benchmark examples in table 1. It required $12,871,320 + 3M$ flops where M is the cost 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[12] 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.

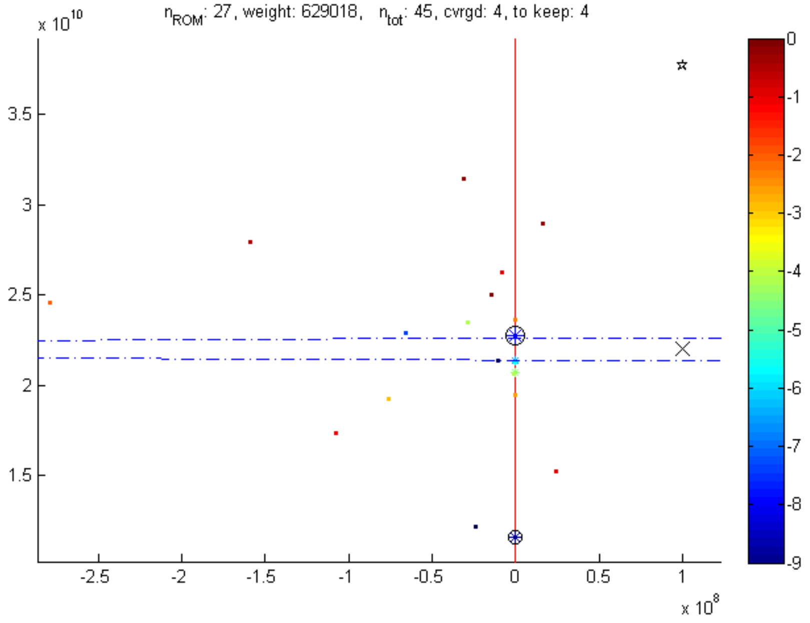
```
cycle 2 expanding at s_0 = 2\pi10^9(0.0159155 + 3.5i),  band_size = 2 + 3

v_defl(6)/H_est  = 0 < 1.49012e-08,   mc = 4

v_defl(6)/H_est  = 0 < 1.49012e-08,   mc = 3
```

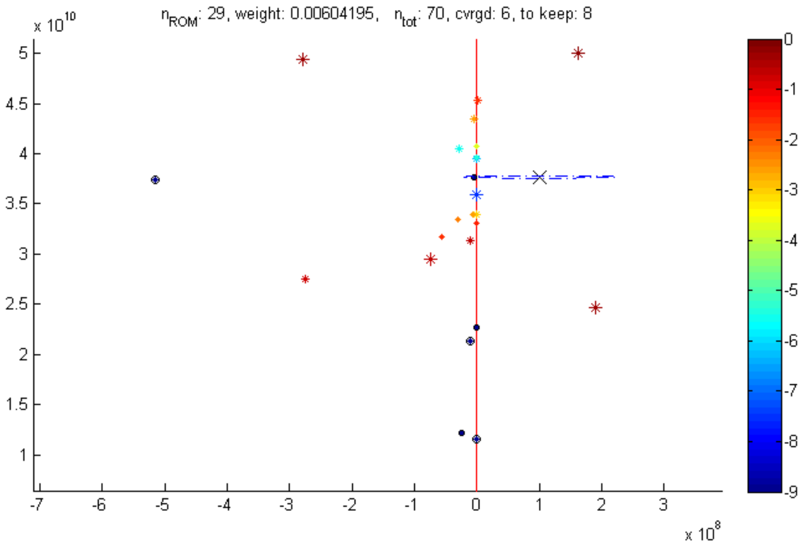
[12] elongated in the plot due to greatly asymmetric scaling. This is one reason why interpolation points on or near the $\Im$-axis can result in the convergence of un-necessarily large ROMs.

```
v_defl(6)/H_est  = 0 < 1.49012e-08,   mc = 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}: 29, weight: 0.00604195, n_{tot}: 70, cvrgd: 6, to keep: 8
x 10^{10}
5
4.5
4
3.5
3
2.5
2
1.5
1
-7
-6
-5
-4
-3
-2
-1
0
1
2
3
x 10^{8}
0
-1
-2
-3
-4
-5
-6
-7
-8
-9

	$\Re(\mu)$	$\Im(\mu)$	rr	wt	keep
1	-2.3746e+07	1.2186e+10i	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 2: The 10 Ritz-poles of lowest relative-residual for each cycle.

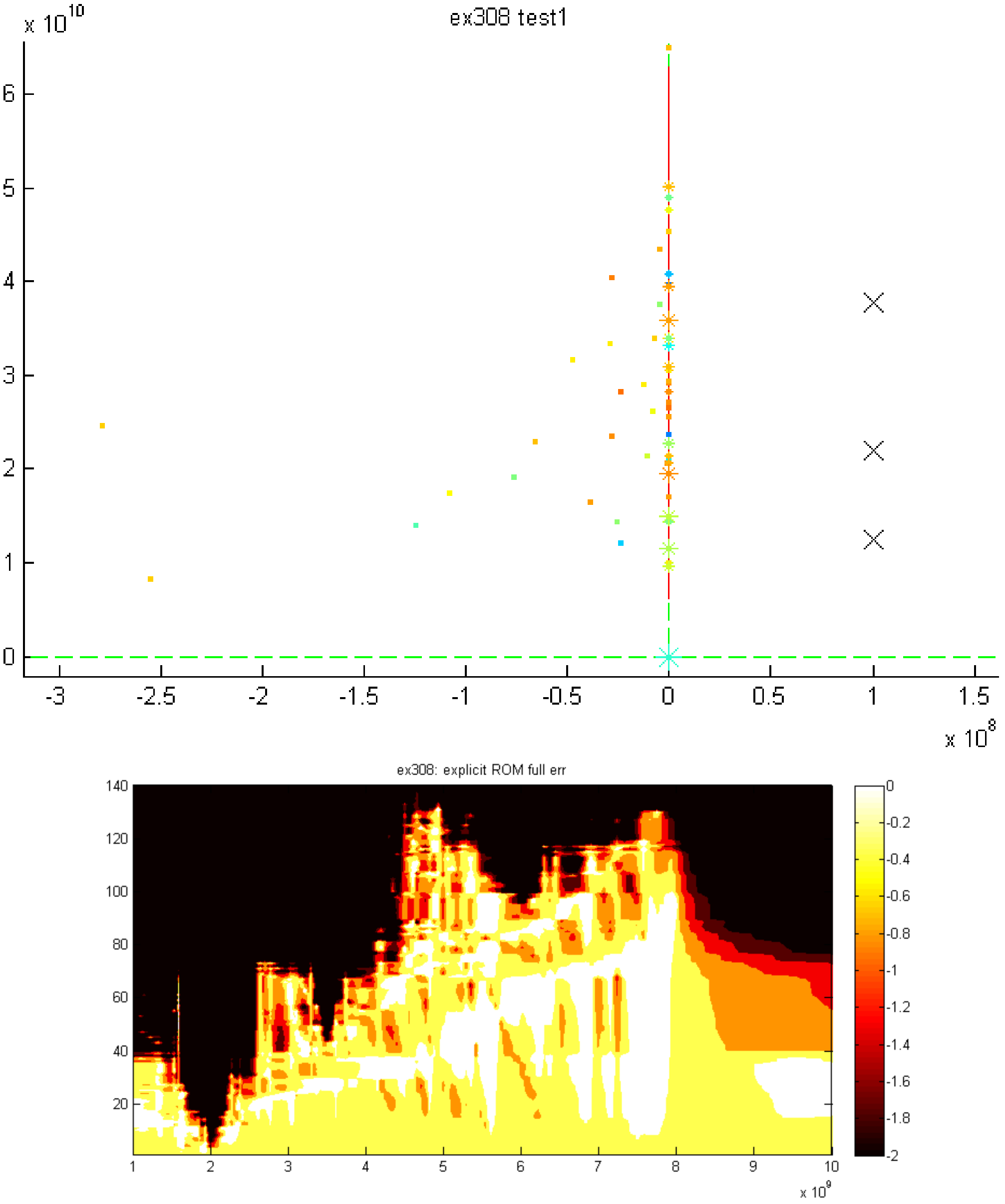


Figure 27: The pole distribution for the explicitly-projected transfer function and interpolation points are in the first plot. The second plot, of local transfer function-error over the frequency range of interest evolving with inclusion of basis vectors in V_{ROM} reflects expansion about the points $p = 2$, 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.

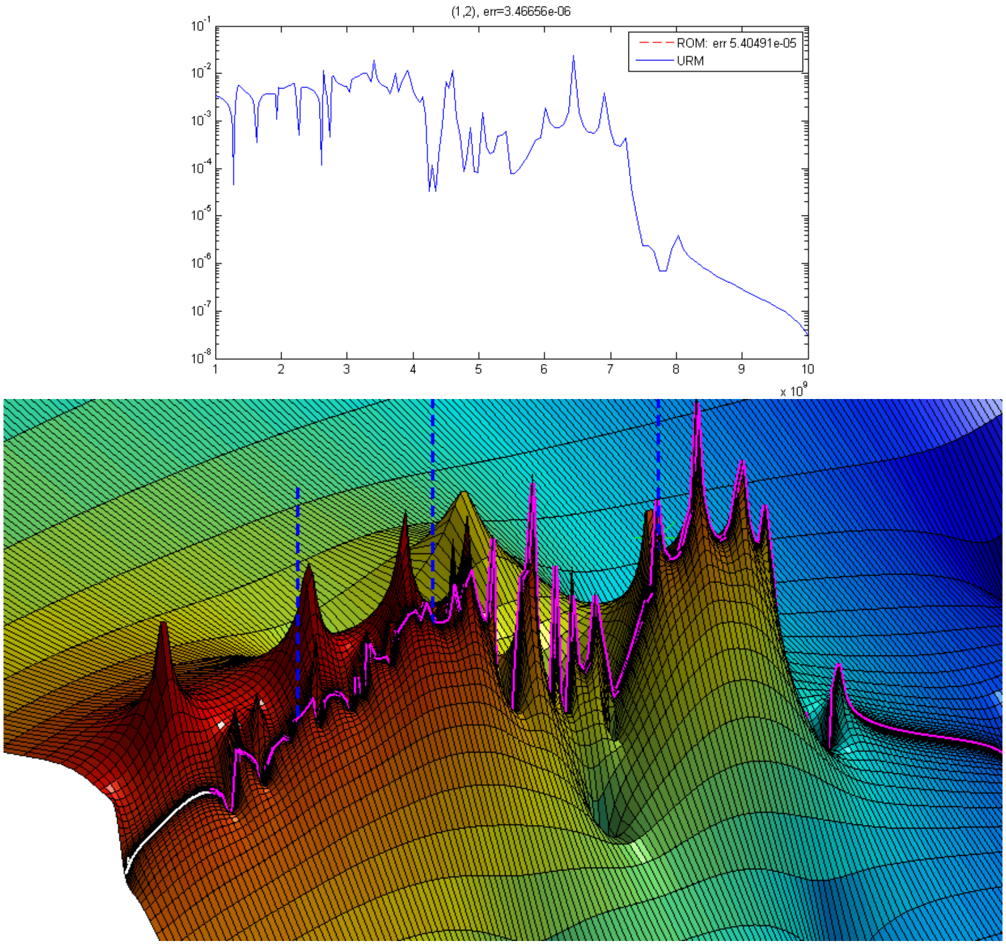


Figure 28: Plots of the $(1, 2)$ component of the frequency-response and transfer-function surface for example 1. It is apparent that the interpolation-point placement was almost ideal for `ex308`, in the sense that the points are near centers of pole-mass. Note that the interpolation points are actually 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.

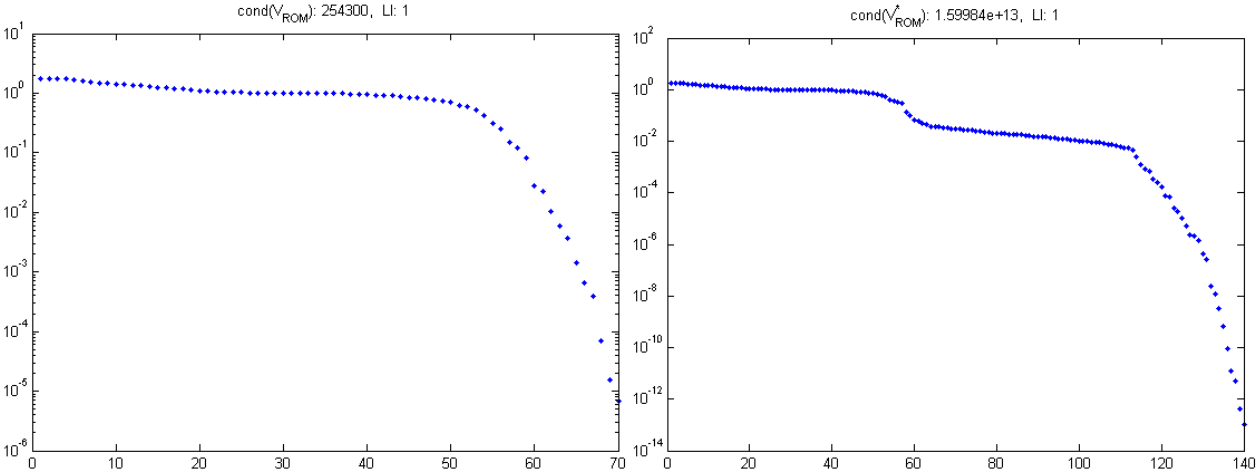


Figure 29: Here we provide singular values for the complex basis $\widehat{V}$ produced by thick-restarted band-Arnoldi, and its realification $\widehat{V}^*$.

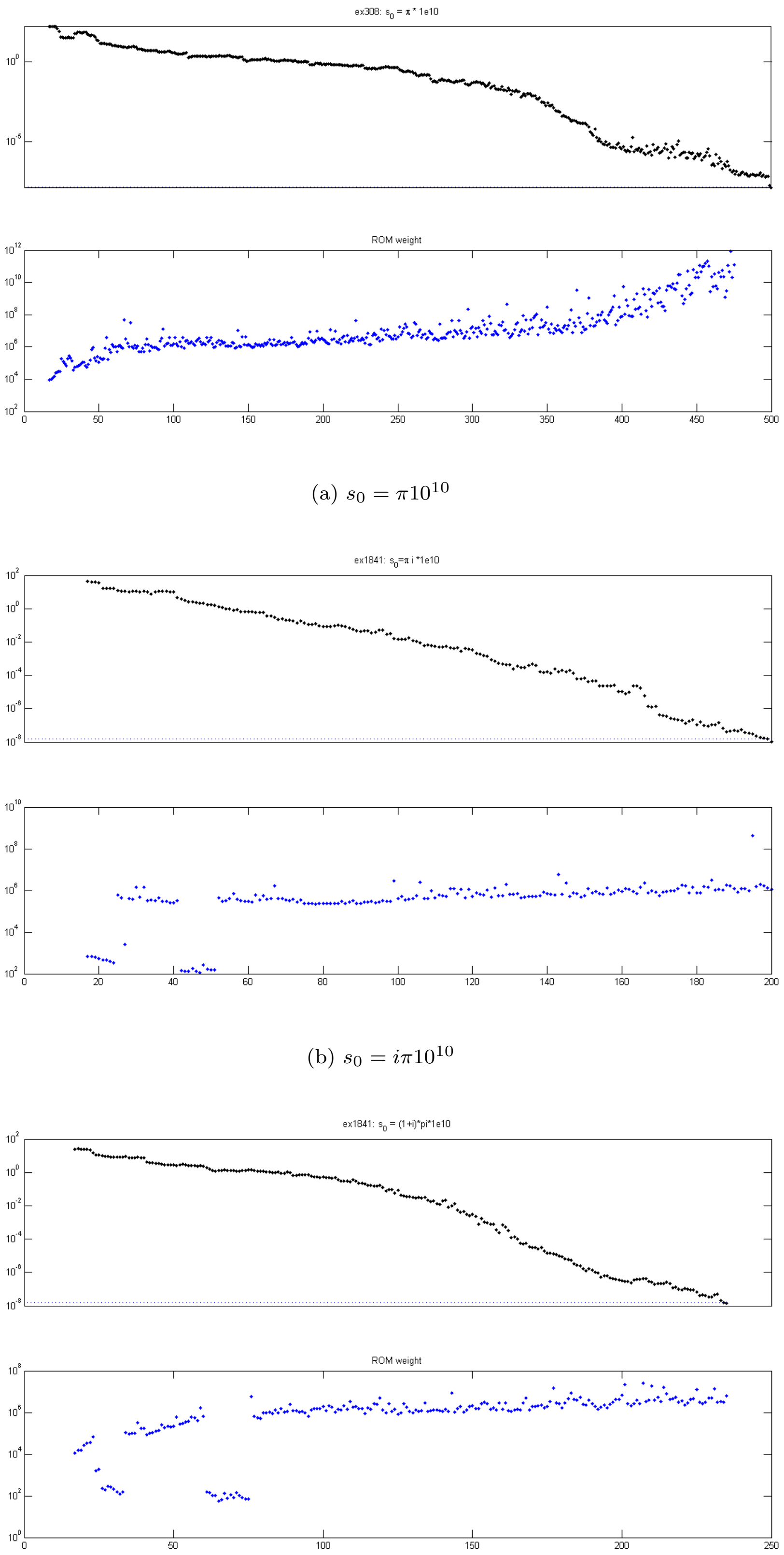
s_0	iterations (n)	ROM size (n')	LI	rel-err	flops	figure
$\pi 10^{10}$	310	310	1	9.1289e-3	1,147,983,165 + M	30a
$i\pi 10^{10}$	106	212	1	8.6730e-3	1,490,525,148 + M	30b
$(1+i)\pi 10^{10}$	142	284	1	8.4797e-3	2,015,563,620 + M	30c

Table 3: Benchmark data for `ex1841`. flops is a count of real (in $\mathbb{R}$), non-zero scalar products required for matrix-vector multiplication and inner-products.

0.4.2 ex1841

`ex1841` is a 16×16 (inputs × outputs) MIMO model. According to R.W. Freund,

> [`ex1841`] came from the design stage of a chip that AT&T produced for a wireless application. (Back then AT&T still had actual manufacturing facilities and still produced its own chips. All of this is gone now.) The RCL circuit in ex1841 was used to model the pin package of that chip. The "pin package" is the wiring used to connect the pins on the chip to the interior of the chip. We got that example directly from the engineers that designed the chip.



(a) $s_0 = \pi 10^{10}$

(b) $s_0 = i\pi 10^{10}$

(c) $s_0 = (1 + i)\pi 10^{10}$

Figure 30: Transfer function relative-error (**??**) and ROM weight vs. n for `ex1841`, at each of the three points shown in figure 20.

Part I

****

0.5 Junk

The thick-restart was introduced by Wu in [52] and [56] as an alternative to Lehoucq and Sorensen's implicit restart applied to Krylov methods for finding eigenvalues of very large matrices or matrix-pencils. The reason for a restart in an eigenvalue method is to deal with storage limitations on the number of basis vectors we can hold, and/or computation cost of orthogonalizing each iterate against n previous vectors on the n-th step. Suppose our memory limits us to storing n vectors, or $\mathcal{O}(nN)$ is the longest we can wait to orthogonalize a new basis vector. Then we must stop the process and throw out some vectors every time we fill n positions. More precisely, we transform the vector basis V into another basis $\begin{bmatrix} Y & Y^{\perp} \end{bmatrix}$ such that we can throw out $Y^{\perp}$ in the so-called deflation phase, and then continue, (re-starting) with Y, until we fill up n positions again. The subspace $Y^{\perp}$ that we throw out typically corresponds to unconverged eigen-information, or perhaps we keep certain nearly-conveged approximate eigen-information in Y that will separate the remaining space and make certain unconverged eigenvalues converge faster. We continue to refine our set of n vectors, cycle after restarted cycle, until we have the set of n vectors we want.

There are several restart methods for eigenvalue finding that achieve the same goal of refining one limited collection of vectors until they converge into a desired set.

The restart for model reduction serves a different purpose, although there are methods [28] and notable [26] that construct a ROM using such a process of refining a projection subspace. The idea has usually been to throw out bad poles of an implicitly projected ROM, refining the projection subspace until there are no poles with positive $\Re$-part or some other undesirable trait. The problem with this is that matching l moments of a model transfer function $\mathcal{H}(s) = \boldsymbol{C}^T(I - (s-\sigma)\mathbf{H})^{-1}\mathbf{R}$ about an interpolation point σ via projection on to a subspace V requires that

$$\mathcal{K}_l(\mathbf{H}, \mathbf{R}) \subseteq \operatorname{span} V.$$

If we throw out basis vectors of $\mathcal{K}_l(\mathbf{H}, \mathbf{R})$ then we lose moment-matching. Suppose $\mathcal{Y} \subset \mathcal{K}_l(\mathbf{H}, \mathbf{R})$ remains after deflation of a restart cycle. The subspace span $\begin{bmatrix} Y & V' \end{bmatrix}$ that we have after the expansion phase of a typical (implicit or explicit) restart with deflated Ritz-vectors is known to be Krylov subspace, but with different starting vectors (not $\mathbf{R}$). It has been shown that ROMs via implicit-projection on to such modified Krylov subspaces can still be good without guaranteed moment matching, but apparently the quality of such ROMs is not reliable.

Our restarted ROM method can described as invariant-subspace recycling. We construct bases for Krylov subspaces at a different shift on every cycle. The thick-restart mechanism allows us to hold on to a growing basis Y of known converged $(\boldsymbol{A}, \boldsymbol{E})$-invariant subspace, and continue orthogonalizing new basis vectors against Y. Assuming $\mathcal{Y}_0 = \emptyset$, the general process is

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

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

    (b) The computed quantities $\widetilde{\boldsymbol{\rho}} = \widetilde{V}_j^H \mathbf{R}_j$ and $\widetilde{\mathbf{H}} = \widetilde{V}_j^H \mathbf{H}_j \widetilde{V}_j$ can be used to **observe** the ROM transfer function
    $$\widetilde{\mathcal{H}}_j(s) = \boldsymbol{C}^T \widetilde{V}_j \big(I - (s-\sigma_j)\widetilde{\mathbf{H}}\big)^{-1} \widetilde{\boldsymbol{\rho}}$$
    via implicit-projection on to the basis $\widetilde{V}_j = \begin{bmatrix} Y_{j-1} & V_j \end{bmatrix}$.

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

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

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

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

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

and it has transfer function

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

Why restart a Krylov-subspace projection ROM method? Our primary reason for thick-restarts is to save cost by not doing full orthogonalization of basis vectors. Rational-interpolation produces the basis for the union of multiple Krylov subspaces, so we can construct them separately. We wish to reduce computational redundancy in constructing their bases, and reduce or eliminate linear dependence.

Since we are likely to be using complex interpolation points (§0.1.5), we will have to split the basis into $\Re$ and $\Im$ parts and re-orthogonalize the resulting real basis anyway, so making all of the constituent bases orthogonal to one-another is unnecessary. Also, for that matter we can further reduce orthogonalization costs by using the alternate inner-product described in §0.1.5.

Why a restarted rational Krylov method might mean smaller ROM size The obvious advantage to a restarted method is lower orthogonalization cost and fewer basis vectors to store. It might imply smaller ROMs as well. Krylov-subspace projection methods for MOR are developed out of eigenvalue methods, the goal of which is to determine some eigenvalues of a matrix or a matrix-pencil as fast as possible, usually ones in a particular region of the complex-plane. To that end, we choose the interpolation-point σ somewhere near the suspected location of the desired eigenvalues, and perform Krylov iterations. Of course it depends on which eigen-directions are strong in the start vector (or general start-space span$\{\mathbf{R}\}$) but in general we will observe that eigenvalues of $(\boldsymbol{A}, \boldsymbol{E})$ near σ converge first, and with further iterations more eigenvalues converge, typically in sequential order of their distance from σ. Well-separated eigenvalues near σ will converge faster than those located in a cluster. For example if σ is located far away from all of the eigenvalues, the entire spectrum is a cluster relative to σ. All of the eigenvalues will converge at more-or-less the same slow rate. As a realistic example, we may observe no convergence for 50 or more iterations; suppose 50 eigenvalues converge on the 51-st iteration of a particular Krylov cycle. If had removed removed a few known eigenvalues from the search by pre-loading their directions in V, the remaining cluster would be more separated. We might see faster and more sequential convergence due to better separation of the eigenvalues. In MOR this gives us a better chance to optimize the ROM. Suppose that of the 50 poles that simultaneously converged on the 51-st iteration, only 2 had significant pole-weight; this would mean that we wasted several iterations obtaining useless eigen-information. With more separated convergence of eigenvalues, by monitoring total converged pole-weight or moment-error we can more easily determine when further iterations are no longer yielding significant information, and it is time to stop or move to a new interpolation-point. This implies a possibly smaller ROM as well, given an adaptive method with shift changes based on estimated regions of high pole-density and using these local-convergence based stopping criteria.

Changing σ changes the order in which the eigenvalues converge because their proximities to σ change. The closer σ_2 is to σ_1, the more similar the sequences of converging eigenvalues. If we run an Arnoldi process starting at σ_2 after having done that with σ_1, and they are near each other, we can expect to see many of the same eigenvalues converge after a number of iterations. If σ_1 and σ_2 are far apart, there may not be a lot of common eigenvalues for the small number of iterations we perform.

Thick-restart cycle

Some of the formalism here is adapted from Stewart's Krylov-Schur method of [55, 54], which is a generalization of Wu's thick-restart.

Suppose we want to construct a basis V for $\mathcal{K}_l(\mathbf{H}, \mathbf{R})$ minus a previously discovered approximately $(\boldsymbol{A}, \boldsymbol{E})$-invariant subspace $\mathcal{Y}$.

Take $Y = \begin{bmatrix} y_1 & y_2 & \cdots & y_\ell \end{bmatrix} \in \mathbb{C}^{N\times\ell}$ as a basis for $\mathcal{Y}$, and $\hat{Y} = \begin{bmatrix} \hat{y}_1 & \hat{y}_2 & \cdots & \hat{y}_\nu \end{bmatrix} \in \mathbb{C}^{N\times\nu}$ the basis for the ν-dimensional residual-space so that we have a block-Krylov relation

$$\begin{aligned} \mathbf{H}Y &= \begin{bmatrix} Y & \hat{Y} \end{bmatrix} \begin{bmatrix} U \\ B \end{bmatrix} \\ &= YU + \hat{Y}B. \end{aligned} \tag{0.1}$$

The coefficient matrix $B = \begin{bmatrix} b_1 & b_2 & \cdots & b_\ell \end{bmatrix} \in \mathbb{C}^{\nu\times\ell}$ is such that the residual

$$\mathbf{H}y_j - Yu_j = \hat{Y}b_j \tag{0.2}$$

of approximate Schur-vector y_j is a linear combination of columns of $\hat{Y}$. We assume $\begin{bmatrix} Y & \hat{Y} \end{bmatrix}$, which is called the basis of the decomposition, has orthogonal columns. Y is orthonormal, but columns of $\hat{Y}$ are not normalized in general. When the residual $\hat{Y}B$ is of rank-1 it is usually written as an outer-product yb^H, but we do away with the $\cdot^H$ notation for the general residual.

The reader should note that, although the $N\times\nu$ residual basis matrix $\hat{Y}$ can be quite large, we only need to know (and store) its representative magnitude in some form. That is because we are primarily interested in $\hat{Y}$ and other residuals for determining residual-norms. For example, one residual norm associated with (0.2) is

$$\|\mathbf{H}y_j - Yu_j\|_2 = \|\hat{Y}b_j\|_2 \leq \|\hat{Y}\|_F \|b_j\|_2. \tag{0.3}$$

Even if we are doing SISO model reduction, we can expect the residual $\hat{Y}$ to be of rank greater than one in general, because every thick-restart cycle produces a space with a different residual vector. That is because we start with a new operator $\mathbf{H}_j$ and operand $\mathbf{r}_j$ on the j-th restart cycle. Then on the j-th restart we potentially have a residual of rank-j. For MIMO models the rank ν of the residual is up to jm on the j-th cycle, where $m = \dim \mathbf{R}_j = \dim \boldsymbol{B}$ (the input dimension).

The subspace $\mathcal{Y}$ is likely a deflated Krylov subspace, or perhaps orthogonalized Ritz-vectors obtained from a dominant-pole finding algorithm such as [34]. The thick-restart method (sometimes called *Krylov-Spectral*) can be regarded as a specific case of Krylov-Schur where Y is a basis of Ritz-vectors with orthogonal residual $\hat{Y}$ that we plan to continue iterating with. The thick-restart proper, as introduced in [52] was intended for a Hermitian operator $\mathbf{H}$ so its Ritz vectors already form an orthogonal set. **Our thick-restart for MOR is different from Wu and Simon's Thick-Restart, and different from Stewart's Krylov-Schur method (for finding eigenvalues) in that we start the expansion phase with arbitrary vectors rather than the residual vectors from another Krylov process.** There may be other applications for using an arbitrary start vector or block, but for model order-reduction it allows us to express the Krylov operand $\mathbf{R}$ in terms of the expanded basis $\widetilde{V} = \begin{bmatrix} Y & V \end{bmatrix}$. Then we can (relatively) cheaply form pole location and weight distributions (§0.1.1) of intermediate ROMs while the method progresses. Typically the residual $\hat{Y}$ is used for continuing iterations but we are going to resume building a basis with $\mathbf{R}$.

We resume basis construction $\begin{bmatrix} Y & v_1 & v_2 & \cdots \end{bmatrix}$ with start block $\mathbf{R} \in \mathbb{C}^{N\times m}$ (made orthogonal to $\mathcal{Y}$). We set the initial "residual"

$$R_0 := (I - YY^H)\mathbf{R}, \tag{0.4}$$

which sets up $\mathbf{R} = \begin{bmatrix} Y & V \end{bmatrix} \widetilde{\boldsymbol{\rho}}$.

Performing k iterations of thick-start Arnoldi with operator $\mathbf{H}$ on start block $\mathbf{R}$ expands (0.5) into

$$\begin{aligned} \mathbf{H}\begin{bmatrix} Y & V \end{bmatrix} &= \begin{bmatrix} Y & \hat{Y} & V & R_k \end{bmatrix} \begin{bmatrix} U & G \\ B & 0 \\ 0 & \widetilde{\mathbf{H}}' \\ 0 & F \end{bmatrix} \\ &= \begin{bmatrix} Y & V \end{bmatrix} \begin{bmatrix} U & G \\ 0 & \widetilde{\mathbf{H}}' \end{bmatrix} + \begin{bmatrix} \hat{Y} & R_k \end{bmatrix} \begin{bmatrix} B & 0 \\ 0 & F \end{bmatrix} \\ &= \begin{bmatrix} Y & V \end{bmatrix} \begin{bmatrix} U & G \\ 0 & \widetilde{\mathbf{H}}' \end{bmatrix} + \begin{bmatrix} \hat{Y}B & R_kF \end{bmatrix}, \end{aligned} \tag{0.5}$$

where $F = \begin{bmatrix} f_1 & f_2 & \cdots & f_k \end{bmatrix} \in \mathbb{C}^{m \times k}$, so that $R_k f_j$ is the residual vector for the j-th column of that block.

Since linear dependence might have been deflated out during iterations, we can expect $\dim R_k = m_c \leq m$ in general, where m_c is the dimension of the current residual, called $\widehat{V}_{defl}$ in [18]. The paper that introduced band-Krylov processes for MOR was [3], and we refer the reader to [3] for a more in-depth discussion. If we use the band-Arnoldi procedure of [18, 19] with exact deflation,

$$F = E_k^{(m_c)} = \begin{bmatrix} 0 & 0 & \cdots & I \end{bmatrix} \in \mathbb{R}^{m_c \times k} \tag{0.6}$$

where the last m_c positions are occupied by the identity matrix $I \in \mathbb{R}^{m_c^2}$. For the single-vector iteration ($m = 1$), we see that (0.6) is the the familiar $F = e_k^T = \begin{bmatrix} 0 & 0 & \cdots & 1 \end{bmatrix}$.

Thick-Arnoldi coefficient-matrix is not the Rayleigh-quotient (but that is ok) Assuming U is upper-triangular, the matrix

$$\begin{bmatrix} U & G \\ 0 & \widetilde{\mathbf{H}}' \end{bmatrix} \tag{0.7}$$

of orthogonalization-coefficients in (0.5) has the form

$$\begin{bmatrix} u & u & u & g & g & g & g \\ & u & u & g & g & g & g \\ & & u & g & g & g & g \\ & & & h & h & h & h \\ & & & h & h & h & h \\ & & & & h & h & h \\ & & & & & h & h \end{bmatrix} \tag{0.7*}$$

for $\ell = \dim \mathcal{Y} = 3$ and $k = \dim V = 4$ (of a SISO model). It (0.7) is used by standard Krylov methods as the Rayleigh-quotient of $\mathbf{H}$ with respect to the constructed basis, which in our case is $\widetilde{V} = \begin{bmatrix} Y & V \end{bmatrix}$ and spans $\mathcal{Y} \cup \mathcal{K}_k(\mathbf{H}, \mathbf{R})$. However (0.7) is not a proper Rayleigh-quotient because $\begin{bmatrix} Y & V \end{bmatrix}$ is not orthogonal to $\hat{Y}B$. Specifically, V is not orthogonal to $\hat{Y}$ because we started construction of V with (0.4) rather than $\hat{Y}$.

We could force (0.7) to be the Rayleigh-quotient by orthogonalizing V against $\begin{bmatrix} Y & \hat{Y} \end{bmatrix}$ during construction rather than just Y, even though $\hat{Y}$ is not included in the resulting basis $\begin{bmatrix} Y & V \end{bmatrix}$. Then we would have span $\begin{bmatrix} Y & V \end{bmatrix} = \mathcal{Y} \cup \mathcal{K}_k(\mathbf{H}, \mathbf{r}) \setminus \operatorname{span}\{\hat{Y}\}$. Unfortunately that means $\mathcal{K}_k(\mathbf{H}, \mathbf{r}) \nsubseteq$ span $\begin{bmatrix} Y & V \end{bmatrix}$, so a ROM constructed via projection on to $\begin{bmatrix} Y & V \end{bmatrix}$ might not have moment matching properties. It is not obvious whether $\operatorname{span}\{\hat{Y}\}$ and $\mathcal{K}_k(\mathbf{H}, \mathbf{R})$ share any common subspace, or whether their intersection contains information that would make the model much different if it were absent. We could overlook this discrepancy for intermediate ROM analysis, and add $\hat{Y}$ back to the basis used for explicit projection. For now we will assume that $V\hat{Y} \neq 0$ (i.e. that we do not orthogonalize V against $\hat{Y}$).

Although we do not necessarily need it, it is helpful to see what the actual Rayleigh-quotient looks like. Assume that $\begin{bmatrix} Y & \hat{Y} \end{bmatrix}$ and $\begin{bmatrix} V & R_k \end{bmatrix}$ are orthogonal bases, and that $\begin{bmatrix} V & R_k \end{bmatrix}$ is orthogonal to Y but not to $\hat{Y}$. We left multiply (0.5) by $\widetilde{V}^H = \begin{bmatrix} Y^H \\ V^H \end{bmatrix}$ to obtain

$$\begin{aligned} \widetilde{\mathbf{H}} = \widetilde{V}^H \mathbf{H} \widetilde{V} &= \begin{bmatrix} U & G \\ 0 & \widetilde{\mathbf{H}}' \end{bmatrix} + \begin{bmatrix} Y^H \\ V^H \end{bmatrix} \begin{bmatrix} \hat{Y}B & R_k F \end{bmatrix} \\ &= \begin{bmatrix} U & G \\ 0 & \widetilde{\mathbf{H}}' \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ V^H \hat{Y} B & 0 \end{bmatrix} \\ &= \begin{bmatrix} U & G \\ \hat{Y}' B & \widetilde{\mathbf{H}}' \end{bmatrix}, \end{aligned} \tag{0.8}$$

where $y' = V^H \hat{Y}$. (0.8) has the form

$$\begin{bmatrix} u & u & u & g & g & g \\ & u & u & g & g & g \\ & & u & g & g & g \\ y & y & y & h & h & h \\ y & y & y & h & h & h \\ y & y & y & & h & h \\ y & y & y & & & h \end{bmatrix} \tag{0.8*}$$

If we want to work with (0.8) we must explicitly compute $\hat{Y}'B = V^H \hat{Y} B$ and add it to (0.7) as part of our process. The extra expense is the same as if we performed the standard Krylov-Schur expansion, but with $\begin{bmatrix} Y & \hat{Y} \end{bmatrix}$ as the thick-restart basis instead of just Y. For that expense it may be preferable to just construct V to be orthogonal to $\hat{Y}$, as mentioned.

Using (0.7) vs (0.8) for eigenvalue decomposition How does the presence of $\hat{Y}'B$ in the Rayleigh-quotient affect Ritz vectors and values? We will look at Arnoldi-Ritz expressions obtained from the eigen-decompositions of (0.7), and of (0.8), and show that (0.7) is better for computing residual errors of Ritz-pairs.

For that implied by the Rayleigh-quotient (0.8), take an eigenvalue decomposition $\widetilde{\mathbf{H}} W = W \Lambda$ of (0.8), expressed in blocks as

$$\begin{bmatrix} U & G \\ \hat{Y}'B & \widetilde{\mathbf{H}}' \end{bmatrix} \begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{bmatrix} = \begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{bmatrix} \begin{bmatrix} \Lambda_1 & \\ & \Lambda_2 \end{bmatrix}. \tag{0.9}$$

Re-write (0.5) as

$$\mathbf{H} \widetilde{V} = \widetilde{V} \left(\widetilde{\mathbf{H}} - \begin{bmatrix} 0 & 0 \\ \hat{Y}'B & 0 \end{bmatrix} \right) + \begin{bmatrix} \hat{Y} & R_k \end{bmatrix} \begin{bmatrix} B & \\ & F \end{bmatrix}$$

and right-multiply it by W, giving us the Krylov-Ritz relation

$$\begin{aligned} \mathbf{H} \widetilde{V} W &= \widetilde{V} W \Lambda - \left(\widetilde{V} \begin{bmatrix} 0 & 0 \\ V^H \hat{Y} B & 0 \end{bmatrix} - \begin{bmatrix} \hat{Y} & R_k \end{bmatrix} \begin{bmatrix} B & \\ & F \end{bmatrix} \right) \begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{bmatrix} \\ &= \widetilde{V} W \Lambda - \left(\begin{bmatrix} Y & V \end{bmatrix} \begin{bmatrix} 0 & 0 \\ V^H \hat{Y} B & 0 \end{bmatrix} \begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{bmatrix} - \begin{bmatrix} \hat{Y} B W_{11} + R_k F W_{21} & \hat{Y} B W_{12} + R_k F W_{22} \end{bmatrix} \right) \\ &= \widetilde{V} W \Lambda - \left(\begin{bmatrix} V V^H \hat{Y} B W_{11} & V V^H \hat{Y} B W_{12} \end{bmatrix} - \begin{bmatrix} \hat{Y} B W_{11} + R_k F W_{21} & \hat{Y} B W_{12} + R_k F W_{22} \end{bmatrix} \right) \\ &= \widetilde{V} W \Lambda + \begin{bmatrix} (I - V V^H) \hat{Y} B W_{11} + R_k F W_{21} & (I - V V^H) \hat{Y} B W_{12} + R_k F W_{22} \end{bmatrix} \\ &= \widetilde{V} W \Lambda + \begin{bmatrix} (I - V V^H) \hat{Y} B & R_k F \end{bmatrix} W \end{aligned}$$

Then for Ritz-vectors $Z = \widetilde{V} W$ where (W, Λ) is the eigen-decomposition of Rayleigh-quotient (0.8), the residual is

$$\mathbf{H} Z - Z \Lambda = \begin{bmatrix} (I - V V^H) \hat{Y} B & R_k F \end{bmatrix} W, \tag{0.10}$$

which may be of use to somebody.

If, instead of taking the eigen-decomposition of the Rayleigh-quotient (0.8), we factor (0.7), as

$$\begin{bmatrix} U & G \\ & \widetilde{\mathbf{H}}' \end{bmatrix} \begin{bmatrix} \tilde{W}_{11} & \tilde{W}_{12} \\ & \tilde{W}_{22} \end{bmatrix} = \begin{bmatrix} \tilde{W}_{11} & \tilde{W}_{12} \\ & \tilde{W}_{22} \end{bmatrix} \begin{bmatrix} U_{\ell\ell} & \\ & \Lambda \end{bmatrix} \tag{0.11}$$

(which is different from (0.9) in general) and right multiply (0.5) by eigenbasis $\tilde{W} = \begin{bmatrix} \tilde{W}_{11} & \tilde{W}_{12} \\ & \tilde{W}_{22} \end{bmatrix}$, we get

$$\mathbf{H} \begin{bmatrix} \tilde{Z}_1 & \tilde{Z}_2 \end{bmatrix} = \begin{bmatrix} \tilde{Z}_1 & \tilde{Z}_2 \end{bmatrix} \begin{bmatrix} U_{\ell\ell} & \\ & \Lambda \end{bmatrix} + \begin{bmatrix} \hat{Y}B & R_k F \end{bmatrix} \begin{bmatrix} \tilde{W}_{11} & \tilde{W}_{12} \\ & \tilde{W}_{22} \end{bmatrix}.$$

In this case we have

$$\mathbf{H}\tilde{Z}_1 = \tilde{Z}_1\Lambda_1 + \hat{Y}B\tilde{W}_{11}$$

and

$$\mathbf{H}\tilde{Z}_2 = \tilde{Z}_2\Lambda_2 + \hat{Y}B\tilde{W}_{12} + R_k F\tilde{W}_{22}$$

Then the block-Ritz-residual

$$\mathbf{H}\tilde{Z}_2 - \tilde{Z}_2\lambda_2 = \begin{bmatrix} \hat{Y} & R_k \end{bmatrix} \begin{bmatrix} B & \\ & F \end{bmatrix} \begin{bmatrix} \tilde{W}_{12} \\ \tilde{W}_{22} \end{bmatrix} \tag{0.12}$$

can be estimated in norm and used to determine convergence of Ritz-vectors.

ROM implied by one thick-restart cycle

The aforementioned eigen-decomposition can be used to look at the order-κ (where $\kappa = \ell + k$) ROM transfer function via implicit-projection on to span $\widetilde{V} = \mathcal{Y} \cup \mathcal{K}_k(\mathbf{H}, \mathbf{R})$, which is defined as

$$\widetilde{\mathcal{H}}(s) = \boldsymbol{C}^T \begin{bmatrix} Y & V \end{bmatrix} \left(I - (s - \sigma)\widetilde{\mathbf{H}} \right)^{-1} \widetilde{\boldsymbol{\rho}}. \tag{0.13}$$

$\widetilde{\mathbf{H}}$ and $\widetilde{\boldsymbol{\rho}} := \begin{bmatrix} Y & V \end{bmatrix}^H \mathbf{R}$ were computed as part of the expansion phase that produced (0.5), but we must explicitly compute (norms of the columns of) $\boldsymbol{C}^T V$, which can cost up to $\mathcal{O}(Nkp)$ where $p = \dim \boldsymbol{C}$.

Eigen-decomposition of $\widetilde{\mathbf{H}}$ yields ROM pole weight distribution Take an eigen-decomposition (0.9) or (0.11) and call it (W, Λ). Let $Z = \begin{bmatrix} Z_1 & Z_2 \end{bmatrix} = \begin{bmatrix} Y & V \end{bmatrix} W$ be the basis of long Ritz-vectors, where Z_1 and Z_2 are the Ritz-vector bases associated with $\mathcal{Y}$ and V respectively. Poles μ_j of (0.13) correspond to eigenvalues $\lambda_j \in \Lambda$ with

$$\mu_j = \sigma + 1/\lambda_j.$$

The weight (sometimes called dominance) of pole μ_j in the pole-residue expansion of (0.13), is

$$|\gamma_j| = \|\delta_j\| \, \|f_j\| \|g_j\|, \tag{0.14}$$

where f_j and g_j are the j-th column of

$$\boldsymbol{C}^T Z = \begin{bmatrix} f_1 & f_2 & \cdots & f_\kappa \end{bmatrix}, \quad \text{and} \quad (W^{-1}\widetilde{\boldsymbol{\rho}})^T = \begin{bmatrix} g_1 & g_2 & \cdots & g_\kappa \end{bmatrix},$$

and

$$\|\delta_j(i\omega)\|_\infty = \begin{cases} \dfrac{|\sigma - \mu_j|}{\min\left\{ |\mu_j - \omega_1|, \ |\mu_j - \omega_2|, \ |\Re(\mu_j)| \right\}}, & \mu_j \neq \infty \\ 1, & \mu_j = \infty \end{cases} \tag{0.15}$$

is the factor representing μ_j's proximity to $i[\omega_1, \omega_2]$ on the $\Im$-axis. The apparent dependence of (0.15) on the shift/interpolation-point σ is misleading. When taken as a whole, a pole's weight (0.14) does not depend on σ. For a more detailed discussion of pole-weight, see §0.1.1.

Note that $\boldsymbol{C}^T Z = \boldsymbol{C}^T \begin{bmatrix} Z_1 & Z_2 \end{bmatrix}$, where $\boldsymbol{C}^T Z_1$ does not need to be re-computed every cycle. We can re-use it from cycle to cycle, appending $\boldsymbol{C}^T z_{\ell+1}$ when a new Ritz-vector $z_{\ell+1}$ is locked. In that way, poles $\mu_1, \mu_2, \ldots, \mu_\ell$, and their weights, are locked in (0.13) and every ROM of future restart cycles.

The factors $\|f_j\|$ and $\|g_j\|$ represent μ_j's weight with respect to filtering by the output and input interfaces of the ROM. The pole μ_j's weight (0.14) (aka dominance) is a converging quantity, like μ_j itself. Analysis of very-low-order models identifies emerging regions of pole-density of the transfer function. It is not clear whether convergence of poles implies convergence of near-by zeros as well, although moment matching about σ certainly implies that near σ.

The total system weight of (0.13) is $\sum |\gamma_j|$, and is itself a converging quantity. It is a sort of transfer function norm $\|\widetilde{\mathcal{H}}(s)\|$.

Deflating a ROM thick-restart cycle

After a cycle of the process, we want to extract a subset (in span) of the basis we constructed, and append that to the thick-start basis for the next cycle. A marked difference between thick-restart/Krylov-Schur for the eigenvalue problem and for our rational-interpolation method for MOR is that we will use a different operator $\mathbf{H}_2$ and operand $\mathbf{R}_2$ on the next cycle, so the eigenvalues of the decomposition will need to be shifted. We will address that.

By performing k iterations of the Arnoldi process with $\mathbf{H}$ on $\mathbf{R}$, and enforcing orthogonality with thick-start basis Y for some ℓ-dimensional subspace $\mathcal{Y}$ such that

$$\mathbf{H}Y = YU + \hat{Y}B, \tag{0.1}$$

we expanded (0.1) into

$$\mathbf{H}\begin{bmatrix} Y & V \end{bmatrix} = \begin{bmatrix} Y & V \end{bmatrix}\begin{bmatrix} U & G \\ & \widetilde{\mathbf{H}}' \end{bmatrix} + \begin{bmatrix} \hat{Y}B & R_k F \end{bmatrix}, \tag{0.5}$$

where the $(\ell+k)\times(\ell+k)$ matrix $\begin{bmatrix} U & G \\ & \widetilde{\mathbf{H}}' \end{bmatrix}$ has the form

$$\begin{bmatrix} u & u & u & g & g & g & g \\ & u & u & g & g & g & g \\ & & u & g & g & g & g \\ & & & h & h & h & h \\ & & & h & h & h & h \\ & & & & h & h & h \\ & & & & & h & h \end{bmatrix} \tag{0.7*}$$

for a SISO ROM with $\ell = 3$ and $k = 4$.

The mechanism of deflation in this scheme is to reduce (0.5) to a new orthonormal Krylov-Schur relation

$$\mathbf{H}_2\widehat{Y} \approx \widehat{Y}\widehat{U} \tag{0.16}$$

where

$$\widehat{Y} = \begin{bmatrix} Y & Y' \end{bmatrix}, \qquad \widehat{U} = \begin{bmatrix} U & * \\ & U' \end{bmatrix} \tag{0.17}$$

represent the enlarged subspace $\widehat{\mathcal{Y}}$ that we will use as the thick-restart basis for the next cycle. The basis for the subspace $\widehat{\mathcal{Y}}$ will contain the same vectors Y, appended with basis Y' for

$$\mathcal{Y}' \subseteq \operatorname{span} V = \mathcal{K}_k(\mathbf{H}, \mathbf{r}) \setminus \mathcal{Y},$$

most likely consisting of newly converged Schur-vectors.

Deflation with eigen-decomposition Recall the thick-restart Krylov decomposition

$$\mathbf{H}\begin{bmatrix} Y & V \end{bmatrix} = \begin{bmatrix} Y & V \end{bmatrix}\begin{bmatrix} U & G \\ & \widetilde{\mathbf{H}}' \end{bmatrix} + \begin{bmatrix} \hat{Y}B & R_kF \end{bmatrix} \tag{0.18}$$

that we want to deflate, and consider the eigen-decomposition

$$\begin{bmatrix} U & G \\ & \widetilde{\mathbf{H}}' \end{bmatrix}\begin{bmatrix} W_{11} & W_{12} \\ & W_{22} \end{bmatrix} = \begin{bmatrix} W_{11} & W_{12} \\ & W_{22} \end{bmatrix}\begin{bmatrix} U_{\ell\ell} & \\ & \Lambda \end{bmatrix} \tag{0.19}$$

of the $(\ell+k)\times(\ell+k)$ (perturbed) Rayleigh-quotient. Note that the $\ell\times\ell$ block W_{11} is upper-triangular and

$$UW_{11} = W_{11}U_{\ell\ell}$$

is an eigenvalue decomposition of U. We assume that the decomposition (0.19) leaves the diagonal entries (eigenvalues) u_{ii} of U in the same order[13], so that

$$\begin{bmatrix} U_{\ell\ell} & \\ & \Lambda \end{bmatrix} = \begin{bmatrix} u_{11} & & & & & \\ & \ddots & & & & \\ & & u_{\ell\ell} & & & \\ & & & \lambda_1 & & \\ & & & & \ddots & \\ & & & & & \lambda_k \end{bmatrix}.$$

Applying the eigen-basis $W = \begin{bmatrix} W_{11} & W_{12} \\ & W_{22} \end{bmatrix}$ to (0.18) from the right yields

$$\mathbf{H}\begin{bmatrix} Z_1 & Z_2 \end{bmatrix} = \begin{bmatrix} Z_1 & Z_2 \end{bmatrix}\begin{bmatrix} U_{\ell\ell} & \\ & \Lambda \end{bmatrix} + \begin{bmatrix} \hat{Y}B & R_kF \end{bmatrix}\begin{bmatrix} W_{11} & W_{12} \\ & W_{22} \end{bmatrix}. \tag{0.20}$$

where the Ritz-vectors

$$\begin{bmatrix} Z_1 & Z_2 \end{bmatrix} = \begin{bmatrix} Y & V \end{bmatrix}\begin{bmatrix} W_{11} & W_{12} \\ & W_{22} \end{bmatrix}.$$

are separated into Z_1, associated with the locked-subspace $\mathcal{Y}$, and Z_2, the new Ritz-vectors generated on this cycle. We want to determine which Ritz-vectors z_j are converged and append them, as a Schur-basis, to Y for the next cycle.

For $Z_1 = YW_{11}$, (0.20) gives us

$$\mathbf{H}Z_1 = Z_1U_{\ell\ell} + \hat{Y}BW_{11} \tag{0.21}$$

which is nothing new $\|\hat{Y}BW_{11}\| = \|\hat{Y}B\|$, except to observe that we will need to translate (0.21) to a relation that involves $\mathbf{H}_2 \neq \mathbf{H}$ and that justifies retaining the Schur-basis Y.

For new Ritz-pairs $\lambda_j \in \Lambda$ and $z_j \in Z_2$, (0.20) implies

$$\begin{aligned} \mathbf{H}Z_2 - Z_2\Lambda &= \hat{Y}BW_{12} + R_kFW_{22} \\ &= \begin{bmatrix} \hat{Y} & R_k \end{bmatrix}\begin{bmatrix} BW_{12} \\ FW_{22} \end{bmatrix}. \end{aligned} \tag{0.22}$$

[13] Matlab's `eig` does this.

Returning to a Schur-decomposition Suppose we are keeping $k' \leq k$ Ritz-values and we rearranged and truncated the eastern part of (0.11) (corresponding to Λ).

Recall that W_{11} is upper-triangular, and W_{22} is not triangular in general. If we take the Schur-decomposition

$$W_{22}S = ST$$

where S is orthonormal and T is upper-triangular. The larger Schur-decomposition

$$\begin{bmatrix} W_{11} & W_{12} \\ & W_{22} \end{bmatrix} \begin{bmatrix} I & \\ & S \end{bmatrix} = \begin{bmatrix} I & \\ & S \end{bmatrix} \begin{bmatrix} W_{11} & W_{12}S \\ & T \end{bmatrix} \tag{0.23}$$

follows. Then applying $\begin{bmatrix} I & \\ & S \end{bmatrix}$ to (0.18) yields

$$\mathbf{H} \begin{bmatrix} Y & VS \end{bmatrix} = \begin{bmatrix} Y & VS \end{bmatrix} \begin{bmatrix} U & GS \\ & T \end{bmatrix} + \begin{bmatrix} \hat{Y}B & R_k FS \end{bmatrix}. \tag{0.24}$$

Returning to a Schur-decomposition via QR decomposition if we take the QR factorization

$$W_{22} = QT \tag{0.25}$$

where Q is orthonormal and T is upper-triangular then the larger QR factorization

$$\begin{bmatrix} W_{11} & W_{12} \\ & W_{22} \end{bmatrix} = \begin{bmatrix} I & \\ & Q \end{bmatrix} \begin{bmatrix} W_{11} & W_{12} \\ & T \end{bmatrix}.$$

follows.

Note that from (0.11) we have $\widetilde{\mathbf{H}}' W_{22} = W_{22}\Lambda$ so (0.25) implies that Q is a Schur basis of $\widetilde{\mathbf{H}}'$, i.e.

$$\widetilde{\mathbf{H}}' Q = Q \left(T\Lambda T^{-1} \right).$$

Then applying $\begin{bmatrix} I & \\ & Q \end{bmatrix}$ to (0.18) from the right yields the Krylov-Schur relation

$$\mathbf{H} \begin{bmatrix} Y & VQ \end{bmatrix} = \begin{bmatrix} Y & VQ \end{bmatrix} \begin{bmatrix} U & GQ \\ & U' \end{bmatrix} + \begin{bmatrix} \hat{Y}B & R_k FQ \end{bmatrix}, \tag{0.26}$$

where $U' = T\Lambda T^{-1}$ is upper-triangular so now we have an updated Schur representation

$$\mathbf{H}Y_2 = Y_2 U_2 + \hat{Y}_2 B_2 \tag{0.27}$$

of the locked subspace $\mathcal{Y}_2 = \operatorname{span} Y_2$ where

$$Y_2 = \begin{bmatrix} Y & VQ \end{bmatrix} \in \mathbb{C}^{N \times (\ell + k')}, \quad \hat{Y}_2 = \begin{bmatrix} \hat{Y} & R_k \end{bmatrix} \in \mathbb{C}^{N \times ?},$$
$$U_2 = \begin{bmatrix} U & GQ \\ & U' \end{bmatrix} \in \mathbb{C}^{(\ell + k')^2}, \quad B_2 = \begin{bmatrix} B & \\ & FQ \end{bmatrix} \in \mathbb{C}^{(m+?)^2}.$$

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. 0.1.1

[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. 12

[3] J. I. Aliaga, D. L. Boley, R. W. Freund, and V. Hernandez. A Lanczos-type method for multiple starting vectors. *MATH. COMP*, pages 1577–1601, 2000. 12, 0.3.1, 21, 0.5

[4] W. E. Arnoldi. The principle of minimized iterations in the solution of the matrix eigenvalue problem. *Q. Appl. Math*, 9(17):17–29, 1951. (document), 0.1.4

[5] Z. Bai and R.W. Freund. Eigenvalue-based characterization and test for positive realness of scalar transfer functions. *Automatic Control, IEEE Transactions on*, 45(12):2396–2402, December 2000. 0.1.2

[6] M.A. Bazaz, Mashuq un Nabi, and S. Janardhanan. A stopping criterion for Krylov-subspace based model order reduction techniques. In *Modelling, Identification Control (ICMIC), 2012 Proceedings of International Conference on*, pages 921–925, 2012. 12

[7] 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. 0.2

[8] Mustafa Celik, Ogan Ocali, Mehmet A Tan, and Abdullah Atalar. Pole-zero computation in microwave circuits using multipoint Padé approximation. *Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on*, 42(1):6–13, 1995. 0.1.1

[9] Erik Cheever. Linear physical systems analysis, pole-zero representations of linear physical systems. This site was designed to accompany a course in linear systems taught at the Department of Engineering at Swarthmore College, but should be useful to anybody interested in these systems. 7

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

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

[12] David Day and Michael A. Heroux. Solving complex-valued linear systems via equivalent real formulations. *SIAM J. Sci. Comput.*, 23(2):480–498, 2001. 0.1.5

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

[14] V. Druskin and V. Simoncini. Adaptive rational Krylov subspaces for large-scale dynamical systems. *Systems & Control Letters*, 60(8):546–560, 2011. 0.2

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

[16] 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. (document), 0.1.1, 0.1.4, 9

[17] Michalis Frangos and Imad M. Jaimoukha. Adaptive rational interpolation: Arnoldi and Lanczos-like equations. *European Journal of Control*, 14(4):342–354, 2008. 0.2

[18] Roland W. Freund. Krylov-subspace methods for reduced-order modeling in circuit simulation. *J. Comput. Appl. Math.*, 123(1-2):395–421, 2000. 12, 12, 0.3.1, 21, 0.5

[19] Roland W. Freund. Model reduction methods based on Krylov subspaces. *Acta Numerica*, 12:267–319, 2003. 0.1.1, 0.1.1, 12, 12, 0.3.1, 0.3.2, 0.5

[20] Kyle Gallivan, Eric Grimme, and Paul Van Dooren. Asymptotic waveform evaluation via a Lanczos method. *Applied Mathematics Letters*, 7(5):75–80, 1994. 0.1.1

[21] Kyle Gallivan, G Grimme, and Paul Van Dooren. A rational Lanczos algorithm for model reduction. *Numerical Algorithms*, 12(1):33–63, 1996. 12, 0.2, 0.2

[22] Juan M Gracia and Francisco E Velasco. Stability of invariant subspaces of regular matrix pencils. *Linear algebra and its applications*, 221:219–226, 1995. 0.1.1

[23] William B Gragg. Matrix interpretations and applications of the continued fraction algorithm. *JOURNAL OF MATHEMATICS*, 4(2), 1974. 0.1.1

[24] E Grimme and K Gallivan. Krylov projection methods for rational interpolation. 1997. 0.2

[25] E Grimme and K Gallivan. A rational Lanczos algorithm for model reduction II: Interpolation point selection. In *Numerical Algorithms*, 1998. 0.1.5, 0.2

[26] 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. 0.1.1, 12, 0.5

[27] 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. 0.1, 0.1.1

[28] 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. 0.1.1, 12, 0.5

[29] Cornelius Lanczos. *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators*. 1950. (document), 0.1.4

[30] Guillaume Lassaux and K Willcox. Model reduction for active control design using multiple-point Arnoldi methods. *AIAA Paper*, 616:2003, 2003. 0.2

[31] 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 1009–1012vol.2, 2004. 12, 0.2

[32] 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. 12, 0.2

[33] RB Lehoucq and KJ Maschhoff. Implementation of an implicitly restarted block Arnoldi method. *Preprint MCS-P649-0297, Argonne National Lab*, 1997. 0.3.1

[34] N. Martins, L. T G Lima, and H. J C P Pinto. Computing dominant poles of power system transfer functions. *Power Systems, IEEE Transactions on*, 11(1):162–170, 1996. 0.5

[35] A. Odabasioglu, M. Celik, and L.T. Pileggi. PRIMA: passive reduced-order interconnect macromodeling algorithm. *IEEE Transactions on computer-aided design of integrated circuits and systems*, 17(8):645–654, 1998. 0.1.1

[36] K Henrik A Olsson and Axel Ruhe. Rational Krylov for eigenvalue computation and model order reduction. *BIT Numerical Mathematics*, 46(1):99–111, 2006. 0.2

[37] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November 1999. Previous number = SIDL-WP-1999-0120. 0.1.3

[38] Theodore W Palmer. *Banach Algebras and the General Theory of *-algebras: Volume 1*, volume 2. Cambridge University Press, 2001. 0.1.5

[39] 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. 12

[40] Beresford N. Parlett and Youcef Saad. Complex shift and invert strategies for real matrices. *Linear Algebra and its Applications*, 8889(0):575–595, 1987. 0.1.5, 0.1.5

[41] Beresford N Parlett and David S Scott. The Lanczos algorithm with selective orthogonalization. *Mathematics of computation*, 33(145):217–238, 1979. 11, 0.3.2

[42] G. Peters and J. H. Wilkinson. Inverse iteration, ill-conditioned equations and Newton's method. *SIAM Review*, 21(3), 1979. 0.1.3, 0.1.3

[43] L.T. Pillage and R.A. Rohrer. Asymptotic waveform evaluation for timing analysis. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 9(4):352–366, 1990. 0.1.1, 0.1.1, 0.1.3

[44] Axel Ruhe. Rational Krylov sequence methods for eigenvalue computation. *Linear Algebra and its Applications*, 58(0):391–405, 1984. 0.1.4, 0.2

[45] Axel Ruhe. The rational Krylov algorithm for nonsymmetric eigenvalue problems. II: Complex shifts for real matrices. 1994. 0.1.3, 0.1.5, 0.2

[46] Y. Saad. Variations on Arnoldi's method for computing eigenelements of large unsymmetric matrices. *Linear Algebra and its Applications*, 34(0):269–295, 1980. 0.1.4

[47] Y. Saad. *Numerical Methods for Large Eigenvalue Problems*. Classics in applied mathematics. Society for Industrial and Applied Mathematics, 2011. 0.1.3

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

[49] Y. SHAMASH. Stable reduced-order models using Padè-type approximations. *Automatic Control, IEEE Transactions on*, 19(5):615–616, 1974. 0.1.1

[50] Y Shamash. Model reduction using the Routh stability criterion and the padé approximation technique. *International Journal of Control*, 21(3):475–484, 1975. 0.1.1

[51] 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. 0.1.1, 0.1.1

[52] Andreas Stathopoulos, Yousef Saad, and Kesheng Wu. Dynamic thick restarting of the Davidson, and the implicitly restarted Arnoldi methods. *SIAM J. Sci. Comput*, 19:227–245, 1996. 0.3.2, 0.5, 0.5

[53] Gilbert W Stewart. On the sensitivity of the eigenvalue problem ax=λbx. *SIAM Journal on Numerical Analysis*, 9(4):669–686, 1972. 0.1.1

[54] Gilbert W Stewart. Addendum to" a Krylov–Schur algorithm for large eigenproblems". *SIAM journal on matrix analysis and applications*, 24(2):599–601, 2002. 0.5

[55] GW Stewart. A Krylov–Schur algorithm for large eigenproblems. *SIAM Journal on Matrix Analysis and Applications*, 23(3):601–614, 2002. 0.3.2, 9, 0.5

[56] Kesheng Wu, Andrew Canning, HD Simon, and L-W Wang. Thick-restart Lanczos method for electronic structure calculations. *Journal of Computational Physics*, 154(1):156–173, 1999. 0.5

Index