Proposal for Qualifying Exam

Efrem Rensi, GGAM

June 3, 2009

Exam Committee

Committee Chairperson:

Prof. Monica Vazirani

Committee Members:

Prof. Zhaojun Bai

Prof. Robert Guy

Prof. Angela Cheer

Prof. Roland Freund

Exam Logistics

Date:

Wednesday, June 3, 2009

Time:

1:10 PM

Location:

MSB 2240

Proposed Research Talk

Restarted Krylov subspace model-reduction methods for RCL circuit simulation

Summary: A general RCL circuit can be described by a time-invariant linear dynamical system, whose state-space dimension N corresponds to the number of circuit components (Resistors, Capacitors, and Inductors), which is often prohibitively large for efficient numerical computation. The goal of model reduction is to find a system of equations of order $n \ll N$, whose solution approximates that of the original system. In addition, the reduced order system should retain the important properties of *stability* and *passivity*. A number of model reduction schemes have been developed in the last 20 years and are used in the electronics industry; however problems with the existing methods keep this a very open and active area of research. I will outline one particular model reduction technique, as well as modifications that I intend to develop and evaluate.

1 Formulation (Generalized State-Space Realization)

For this research we are concerned with the RCL circuit model described by the linear time invariant (LTI) first-order dynamical system of the form

$$\begin{aligned} Ex' &= Ax + Bu \\ y &= B^T x, \end{aligned} \tag{1}$$

where $A, E \in \mathbb{R}^{N\times N}$, E is singular (in general) and $B \in \mathbb{R}^{N\times p}$. This is an *input-output* system with p inputs and p outputs: $u(t) \in \mathbb{R}^p$ represents inputs to the circuit we wish to model, and $y(t) \in \mathbb{R}^p$ is the output vector. The *state-space* vector $x(t) \in \mathbb{R}^N$ represents the internal state of the model at time t. For modeling a VLSI (Very Large Scale Integrated) circuit, the state space dimension N roughly

corresponds to the number of circuit components, which is often large enough to prohibit efficient simulation. For the purposes of simulation we are only concerned with the output of the model in relation to a given input, so as long as the model behaves the way it was designed, its internal state is not important. Viewed this way, the input-output system (1) is a function $y = F(u)$. Ideally we want a model that behaves like that described by (1), but with as small a state-space dimension as possible: For some $n \ll N$ we want matrices E_n, A_n, B_n so that

$$\begin{aligned} E_n z' &= A_n z + B_n u \\ \hat{y} &= B_n^T z, \end{aligned} \tag{2}$$

and the output $\hat{y} \in \mathbb{R}^p$ approximates $y \in \mathbb{R}^p$ from (1), given the same input $u \in \mathbb{R}^p$. It may be possible to create a reduced model that avoids internal states altogether; however, the advantage to having a model of the form (2) is that, as long as E_n, A_n, B_n preserve a certain structure, the reduced model is also physically realizable. In the case of circuits this means we effectively reduce the number of necessary components, yielding a smaller and more efficient circuit.

1.1 Transfer Function

To compare the original and reduced order model, we need a notion of a direct relationship between input and output. This is accomplished in the frequency domain by means of a *transfer function*. Applying the Laplace transform to (1) yields the algebraic equations

$$\begin{aligned} sEX &= AX(s) + BU(s), \\ Y(s) &= B^T X(s). \end{aligned}$$

Then $Y(s) = H(s)U(s)$, where

$$H(s) = B^T \left(sE - A\right)^{-1} B \quad \in \quad (\mathbb{C} \cup \infty)^{p \times p} \tag{3}$$

is the transfer function over a set $S \subset \mathbb{C}$ with which we are concerned. The transfer function is actually defined on a frequency interval:

$$s = 2\pi i f, \qquad f \in [f_{\min}, f_{\max}], \tag{4}$$

so S is an interval on the i-axis of the complex plane.

For SISO models, we typically visualize the transfer function by plotting its magnitude over the frequency range of interest, such as in figure 1. Peaks and valleys in the plot are referred to as its *features* and the order of magnitude of the function in the absence of features is called its *baseline*. A multi-input, multi-output (MIMO) model with p inputs and p outputs can be regarded as p^2 SISO models, with that many scalar-valued transfer functions.

Note that $H(s)$ is defined only if the *matrix pencil* $(sE - A)$ is *regular*, meaning $(\mu E - A)$ is singular for only a finite number of values $\mu \in \mathbb{C} \cup \infty$.

It is not a great loss of generality to assume that the pencil $sE - A$ has an eigenvalue decomposition, which is expressed as

$$AZ = EZ\mathcal{M}, \tag{5}$$

where $\mathcal{M} \in \mathbb{C}^{N \times N}$ is the diagonal matrix consisting of eigenvalues μ_j and $Z \in \mathbb{C}^{N \times N}$ is the invertible matrix whose columns are the eigenvectors z_j.

We introduce other formulations of $H(s)$ that will be useful:

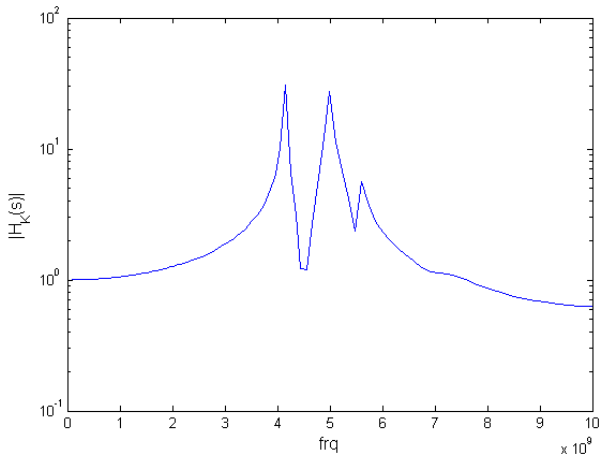


Figure 1: Magnitude $|H(s)|$ of the transfer function for a single-input, single-output (SISO) model. $s = 2\pi i f$ for frequency range $f \in [10^8, 10^{10}]$.

1.1.1 Single matrix formulation

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

$$\begin{aligned} H(s) &= -B^T \left(A - s_0 E - (s - s_0)E\right)^{-1} B \\ &= B^T \left(I - (s - s_0)\widehat{H}\right)^{-1} R \end{aligned} \tag{6}$$

where

$$\widehat{H} := -(s_0 E - A)^{-1} E \quad \text{and} \quad R := (s_0 E - A)^{-1} B. \tag{7}$$

This is known as the *single matrix formulation* of the transfer function, named for the generally non-sparse matrix $\widehat{H} = \widehat{H}(s_0) \in \mathbb{C}^{N \times N}$.

1.1.2 Pole-residue formulation

The generalized eigenvalue decomposition (5) of $(sE - A)$ implies that

$$s_0 EZ - AZ = s_0 EZ - EZ\mathcal{M},$$

so

$$-\widehat{H} = (s_0 E - A)^{-1} E = Z(s_0 I - \mathcal{M})^{-1} Z^{-1}.$$

Substituting this into (6) yields the *pole-residue formulation*

$$\begin{aligned} H(s) &= B^T Z \left(I + (s - s_0)(s_0 I - \mathcal{M})^{-1}\right)^{-1} Z^{-1} R \\ &= X_\infty + \sum_{\substack{j=1 \\ \mu_j \neq \infty}}^{N} \frac{(s_0 - \mu_j)\, X_j}{s - \mu_j}, \end{aligned} \tag{8}$$

where scalars $\mu_j \in \mathbb{C} \cup \infty$ are the *poles* of the transfer function and $(s_0 - \mu_j)X_j(s_0)$ are called *residues*. The leading term $k_\infty = k_\infty(s)$ is a polynomial in s encompassing terms involving infinite poles.

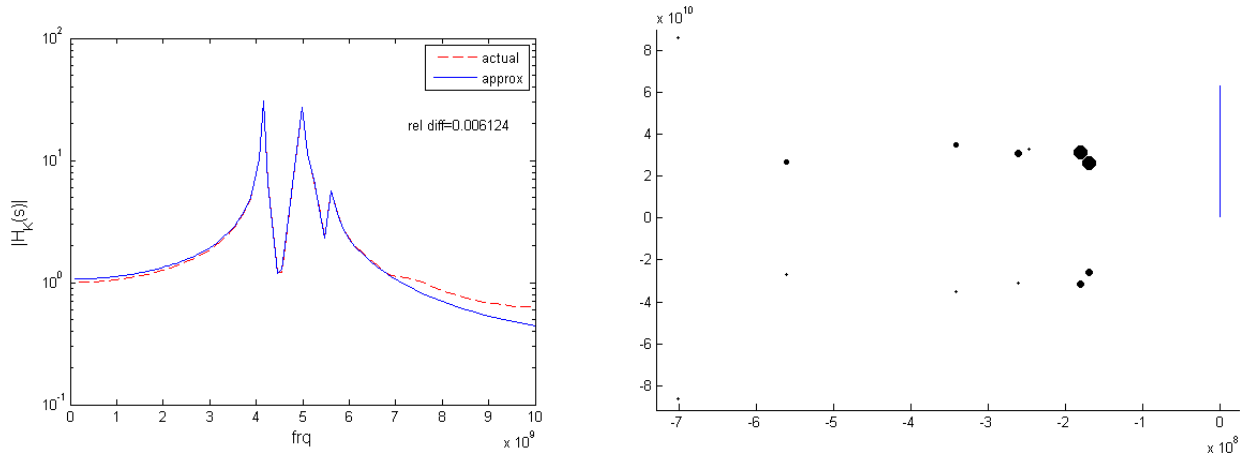
Eigenvalues $\lambda = \lambda(s_0)$ of $\widehat{H}$ and poles μ (eigenvalues of $sE - A$) are related by

$$\lambda_j = \frac{1}{\mu_j - s_0}, \quad j = 1, 2, ..., N, \tag{9}$$

and they share the same eigenvectors z_j.

From (8) it is apparent that the features of $H(s)$ are determined by the poles (and zeros) near the region S with which we are concerned. Dominant poles located 'far' (orders of magnitude) from S, as well as true infinite poles, determine the baseline of $H(s)$ (for $s \approx \infty$). Non dominant poles located far from S are insignificant.

Note. In some cases the transfer function $H(s)$ cannot be expressed in the form (8). (Take for example $H(s) = 1/s^2$.) The formulation (3), however, is always valid and its poles are defined as values μ_j such that $H(\mu_j) = \infty$.



(a) (b)

Figure 2: (a) The pole expansion (8) of the transfer function from figure 1, truncated at 14 most relevant terms out of $N = 1841$. (b) is the (finte) poles themselves plotted in the complex plane, with each pole's size indicating its term's *weight* in (8). The vertical segment $S = 2\pi * [10^8, 10^{10}]$ on the i-axis is the region associated with the frequency range over which we consider $H(s)$. The dominant poles correspond to peaks in the magnitude of the transfer function.

1.1.3 Stability and Passivity

In order for the circuit model to be realistic, the dynamical system (1) describing it must be *stable*, meaning that the state-space variable remains bounded as $t \to \infty$. The model possesses this property only if $\operatorname{Re} \mu_j \leq 0$ for every finite pole μ_j of the transfer function $H(s)$, and poles on the i-axis are simple.

A *passive* circuit is one that does not generate energy. The RCL circuits we model do not contain voltage sources; in fact they only consume energy. They are strictly passive, and any valid model of an RCL circuit must retain this property, which in our formulation is equivalent to *positive realness* of the transfer function (3). Our formulation (1) of the model is such that $H(s)$ is positive real only if

$$-(A + A^T) \succeq 0 \quad \text{and} \quad E \succeq 0 \tag{10}$$

(are positive semi-definite) [2, Theorem 4.5]. We assume that in the unreduced model (1), A, E are formulated correctly and satisfy (10). It is important that the transfer function of any reduced model is positive real, if we are to believe that it represents a physically possible RLC circuit. For a more detailed discussion of stability and passivity of the models, see [2, Sec. 4].

1.2 Padé and Padé-type approximations

Another way to express the transfer function $H(s)$ is as a Taylor series expansion about the point s_0 (introduced in section 1.1.1), which we hereby refer to as the *expansion point*.

$$H(s) = \sum_{j=0}^{\infty} (s - s_0)^j M_j, \tag{11}$$

where $M_j \in \mathbb{C}^{p \times p}$ are the *moments* of H about s_0. The approximation $H_n(s)$ that satisfies

$$H(s) = H_n(s) + \mathcal{O}\left((s - s_0)^{q(n)}\right) \tag{12}$$

is called a Padé approximation if $q(n)$ is as large as possible, i.e. the Taylor expansions of H and H_n agree at as many terms as possible; otherwise we call it a Padé-type approximation. An upper bound for $q(n)$ has been established (see [2]). Although it is possible to produce a Padé approximation of the transfer function, it is well known that such a model in general cannot preserve stability or passivity.

Equation (12) implies that the approximation is only guaranteed in a small neighborhood around s_0, but in practice the reduced model tends to converge in a much larger region.

The Krylov subspace projection method we will describe produces Padé-type models based on the following fact: Via Neumann series expansion we can express (6) as

$$H(s) = B^T \left(\sum_{j=0}^{\infty} (s - s_0)^j \widehat{H}^j \right) R. \tag{13}$$

The moments M_j from (11) are specified exactly: $M_j = B^T \widehat{H}^j R$.

1.3 Krylov subspaces

The *n-th Krylov subspace* induced by $\widehat{H} \in \mathbb{C}^{N \times N}$ and starting vector $r \in \mathbb{C}^N$ is defined as

$$\mathcal{K}_n\left(\widehat{H}, r\right) = \text{span}\left\{ r, \widehat{H}r, \widehat{H}^2 r, \ldots, \widehat{H}^{n-1} r \right\}. \tag{14}$$

The n-th *block* Krylov subspace is the generalization of (14) to multiple starting vectors represented as columns of the matrix $R = \begin{bmatrix} r_1 & r_2 & \ldots & r_p \end{bmatrix}$, resulting in the space $\mathcal{K}_n\left(\widehat{H}, R\right)$.

The matrix $\widehat{H}$ is very large and sparse in general and it is not feasible to explicitly construct it, but we assume we have some relatively efficient way to compute the matrix-vector product $\widehat{H}v$ for a vector $v \in \mathbb{R}^N$. Given (13), one naive approach (*explicit moment matching*) to computing the transfer function $H(s)$ involves generating the block Krylov-sequence

$$R, \widehat{H}R, \widehat{H}^2 R, \ldots \tag{15}$$

to as many terms as we wish via successively left multiplying the previous term by $\widehat{H}$. Unfortunately, in finite-precision arithmetic the sequence (15) converges to the eigenspace associated with the dominant eigenvector(s) of $\widehat{H}$, (which are also eigenvectors of the pencil $sE - A$). In general we will not get a good approximation using explicit moment matching, but this property of selective convergence to dominant eigenvalues can be used to our advantage (see section 1.5). Rather than explicit moment matching, we reduce the order of the model (1) using a projection on to the vector space implied by (15).

1.4 The Arnoldi process

The Arnoldi process (Arnoldi 1951) generates a basis for the Krylov subspace $\mathcal{K}_n\left(\widehat{H}, r\right)$. Given a start vector r and a method to compute the matrix-vector product $\widehat{H}v$ for $v \in \mathbb{R}^N$, the Arnoldi process produces an orthonormal basis $\{v_1, v_2, \ldots, v_n\}$ for $\mathcal{K}_n\left(\widehat{H}, r\right)$ via the following recursion:

1. $v_1 := r/\|r\|$
2. For $n \geq 2$, compute $q = \widehat{H}v_{n-1}$, and orthogonalize q against $\{v_1, \ldots, v_{n-1}\}$ using the Grahm-Schmidt process.
3. set $v_n := q/\|q\|$

The *Band-Arnoldi* algorithm [2] is a generalization of the Arnoldi process to generate a basis for a block-Krylov subspace, for which the possibility of linear dependence between and within the blocks $\widehat{H}^j R$ must be dealt with.

1.4.1 Reduced order models via explicit projection

The *n-th block Krylov subspace* induced by $\widehat{H}$ and R is defined as

$$\mathcal{K}_n\left(\widehat{H}, R\right) := \operatorname{span}\left\{R, \widehat{H}R, \widehat{H}^2R, \ldots, \widehat{H}^{n-1}R\right\}.$$

Given a matrix $V_n \in \mathbb{R}^{N\times n}$ with full column rank and

$$\mathcal{K}_n\left(\widehat{H}, R\right) \subseteq \operatorname{colspan} V_n, \tag{16}$$

set

$$A_n := V_n^T A V_n, \quad E_n := V_n^T E V_n, \quad B_n := V_n^T B. \tag{17}$$

$H_n(s)$ is not guaranteed to be positive real unless V_n is real. We can ensure realness by using

$$\tilde{V}_n = \operatorname{colspan}\left[\operatorname{Re} V_n \ \operatorname{Im} V_n\right]$$

in place of V_n, if necessary.

The reduced order model (2) with transfer function

$$H_n(s) = B_n^T \left(sE_n - A_n\right)^{-1} B_n \tag{18}$$

obtained this way is of Padé-type (for proof, see [3, Theorem 2]), and preserves stability and passivity of the unreduced model, since (10) and (17) imply that

$$-(A_n + A_n^T) \succeq 0 \quad \text{and} \quad E_n \succeq 0$$

The matrix V_n computed by the Arnoldi process satisfies strict equality in (16). The PRIMA algorithm [7] generates a reduced model (18) this way. A more recent variation, SPRIM [4] takes advantage of the more relaxed containment (16) requirement and produces reduced-order projected matrices A_n,E_n and B_n that retain the block structure from the unreduced formulation, resulting in a model that is physically realizable as a circuit. For a more thorough discussion of structure preservation, see [4].

Let $(\tilde{\mu}, w)$ be an eigen-pair of the matrix pencil $sE_n - A_n$. Then $\tilde{\mu}$ is a pole of (18) and we will call it a *Ritz-pole*. Via (17),

$$\left\|V_n^T(\tilde{\mu}E - A)y\right\| = 0 \tag{19}$$

for the Ritz-vector $y := V_n w$, which is a candidate for an eigenvector of $sE - A$. We explicitly compute the relative residual

$$r_{\tilde{\mu}} = \frac{\|\tilde{\mu}Ey - Ay\|}{\|Ay\|} \tag{20}$$

to determine whether $\tilde{\mu}$ is a good approximation to a pole of the unreduced model. Ideally want Ritz-poles to converge to dominant poles of (3), (8), as quickly as possible.

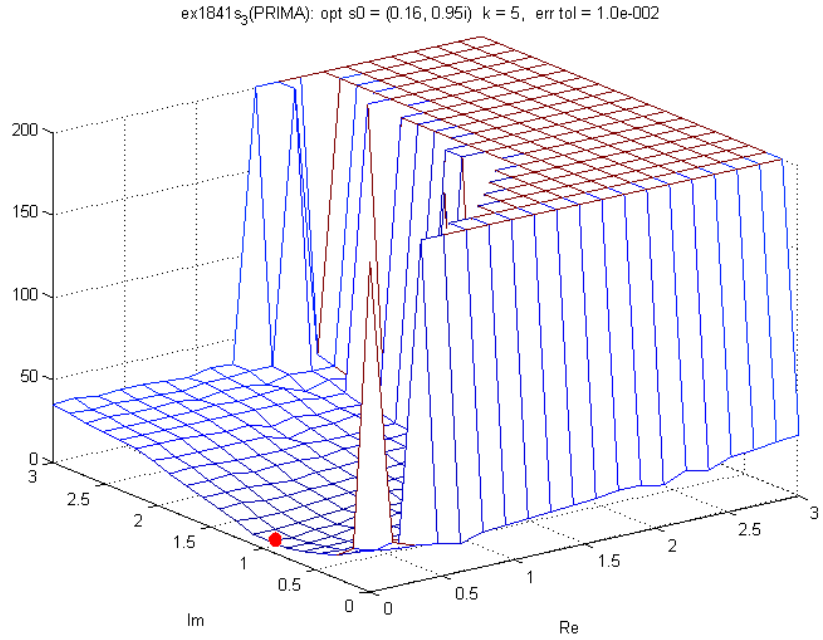


Figure 3: number of iterations to converge (or 200) to approximation with 0.01 relative ∞-norm error Vs. location of s_0 for the model of size $N = 1841$ from figure 4. Units x, y on the axes indicate $s_0 = 2\pi \cdot 10^{10} \cdot (x + iy)$. According to this plot the optimal location is $s_0 = i\pi \cdot 10^{10}$

1.5 Selection of expansion point s_0

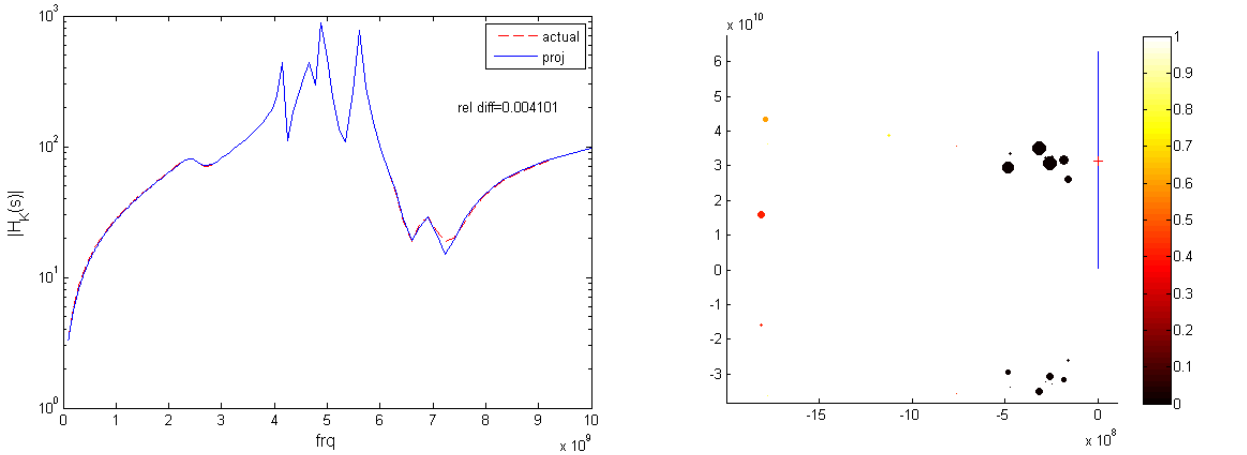
In (6), the point $s_0 \in \mathbb{C}$ was used as a shift to facilitate the single matrix form of the transfer function. It was also used in (11) in the definition of the Padé property of the model. The two uses of s_0 are somewhat related, and it is from (11) that we get the name *expansion point* for s_0. The expansion point always must be chosen so that $s_0E - A$ is invertible. Typically the circuit model is very large and we do not know where its poles are, but since any RCL circuit model is stable, its transfer function $H(s)$ cannot have any poles $\mu_j \in \mathbb{C}^+$ (with positive real part). Then it is always safe to choose $s_0 \in \mathbb{C}^+$.

1.5.1 Expansion point s_0 and local/global convergence

The expansion point influences convergence of the Arnoldi process roughly as follows: Given a vector $v \in \mathbb{C}^N$ expressed in terms of the eigenvalue decomposition of $\widehat{H}$,

$$\widehat{H}v = \sum_{j=1}^{N} \alpha_j \widehat{H} z_j = \sum_{j=1}^{N} \alpha_j \lambda_j z_j = \sum_{j=1}^{N} \frac{\alpha_j z_j}{\mu_j - s_0}. \tag{21}$$

Ignoring for now the *strength* $|\alpha_j|$ of v in the j-th eigen-direction, (21) is dominated by terms involving μ_j close to s_0. In finite arithmetic, these terms can render the lesser ones insignificant, essentially



(a) (b)

Figure 4: Reduced order transfer function (a) and poles (b) for a projected model of $k = 15$ iterations. '+' in (b) indicates location of the optimal expansion point $s_0 = i\pi \cdot 10^{10}$ used to compute this model. Size of pole indicates its dominance in (8), color of the pole indicates its relative residual (degree of convergence). Note that s_0 is located near a cluster of dominant poles, hence the fast convergence of the model there. Also note that the complex conjugate poles converge at the same rate.

confining $\widehat{H}v$ to an $\widehat{H}$-invariant subspace. Thus, the vectors produced by the Arnoldi process converge first to the eigenspace associated with those poles near s_0, then slowly (or never) spread out to more of Krylov subspace as the iterations continue. We speak of the Arnoldi process 'searching' or 'discovering' the Krylov subspace in this way. The process is highly nonlinear and there are few proven statements about the nature of this convergence. Instead, a general heuristic for fast convergence to an accurate reduced model suggests to place s_0 near where we suspect dominant poles to be. Of course the strength α of v in a particular eigen-direction plays an important role. For example, if $v = z_k$ is an eigenvector of $\widehat{H}$, then $\alpha_j = 0$ for $j \neq k$, and $\widehat{H}v$ will never escape that eigenspace.

For a more extensive discussion of the topic of interpolation (expansion) point we refer the reader to [6, chapter 6].

1.5.2 Complex vs strictly real s_0

One reason to use a strictly real expansion point is that it possibly reduces the number of necessary computations. For $s_0 \in \mathbb{R}$, the matrices $\widehat{H}$ and R are also real. Matrix multiplications $\widehat{H}\widehat{H}^j R$ for $j \geq 0$ are four times cheaper than for general complex matrices. For complex $\widehat{H}$, the basis vectors v_j produced by the Arnoldi process are also complex, meaning we must store twice as much data (essentially twice as many vectors). The orthogonalization step in the Arnoldi algorithm thus also potentially involves twice as many inner products per iteration. The projections (17) also require up to 4 times as many matrix-vector products.

It appears that using $s_0 \in \mathbb{R}$ so that all data remains real makes for a much more efficient algorithm. However, it is possible that the faster convergence gained by more aggressive placement of $s_0 \in \mathbb{C}^+$ may make up for the additional computational cost of complex arithmetic. This is an open area of research that we would like to pursue.

1.6 Thick-Restarted Arnoldi with shifting expansion point

1.6.1 Restarted Arnoldi

The Arnoldi process computes a basis matrix V_n for $\mathcal{K}_n\left(\widehat{H}, r\right)$ using Grahm-Schmidt orthogonalization. Each new candidate vector v_n must be orthogonalized against every v_j for $j = 1, 2, ..., n-1$, which is computationally expensive as n gets large. One way to make the process more efficient is to restart the process after m steps. Assuming we have a way to produce strictly real basis vectors (which we do), r restarted runs of m iterations yields the block matrix

$$\widehat{V}_n = \begin{bmatrix} V_1 & V_2 & \cdots & V_r \end{bmatrix},$$

where each $V_j \in \mathbb{R}^{N \times m}$ is orthogonal. We lose orthogonality between the blocks and as a result of restarting, the Arnoldi process will "forget" and then "re-discover" information, which is inefficient and introduces redundancy (linear dependence) in $\widehat{V}_n$, causing computation of the transfer function to be ill-conditioned.

1.6.2 Deflation/Thick Restarts

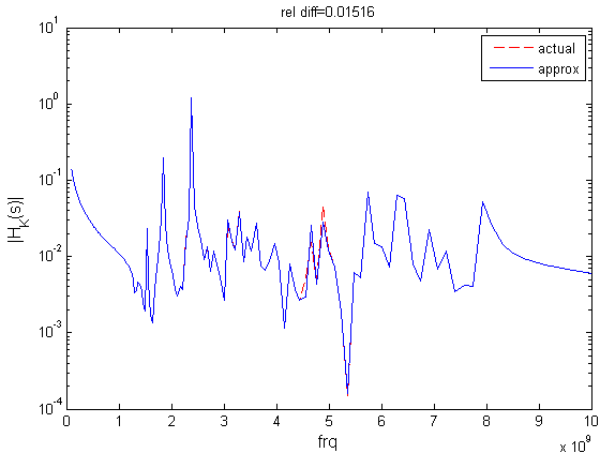
Our solution to this end is to employ *thick* restarts. Given one block Arnoldi matrix V_k of length m, we compute A_m, E_m a la (17) and extract Ritz vectors y_j and their corresponding relative residuals (20), (19). The Ritz vectors with low residuals form a (near) $\widehat{H}$-invariant subspace Y_k of V_k, and correspond to (nearly) converged poles that dominate (21) and prevent further progression of the Arnoldi process. This process of distilling V_k into its converged subspace Y_k is known as *deflation*. Alternatively, we can extract the invariant subspace using the generalized Schur (QZ) decomposition. On the next run of the Arnoldi process, we eliminate this converged eigen-information from the search by orthogonalizing potential Arnoldi vectors against Y_j for $j = 1, 2, ..., k$. Depending on how many poles have converged thus far, thick restarting is more computationally expensive than simple restarting, but still much less costly than orthogonalizing against every previously computed vector.

1.6.3 Experiments with shifting s_0

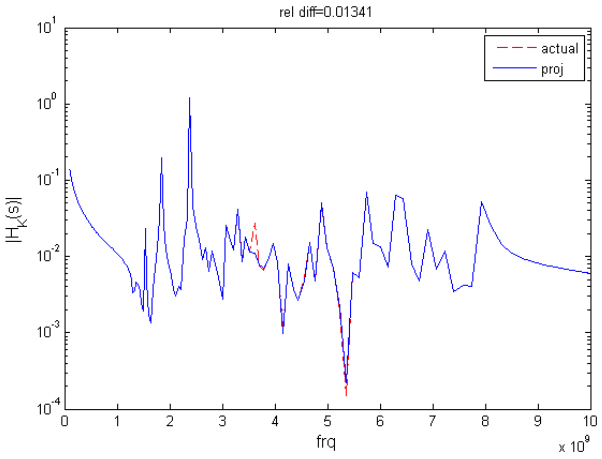
Deflation and assessment of converged poles presents the opportunity to shift the expansion point s_0 to a location where faster convergence is desired. Ideally the algorithm will be totally adaptive based on discovery of new information as the process continues. As of now, we have implemented a simple s_0 shifting scheme: s_0 is purely imaginary and moves up the i-axis along the segment S once per restart. The hope is that such a sweep will bring s_0 near any significant poles, for relatively fast convergence over S.

One preliminary result that looks promising is illustrated in figure 5. The linearly moving s_0 scheme with thick restart ($r = 4$ restarts of $m = 15$) yields a fairly accurate reduced model of size $n = 60$. To achieve the same accuracy with full (no restarts) Arnoldi using a standard fixed $s_0 = \pi \cdot 10^{10}$, we need a reduced model of size $n = 110$. Also included are examples of restarted Arnoldi without s_0 shift, one resulting in a size $n = 60$ model and the other, of size $n = 110$. In both cases the reduced models are not very good.

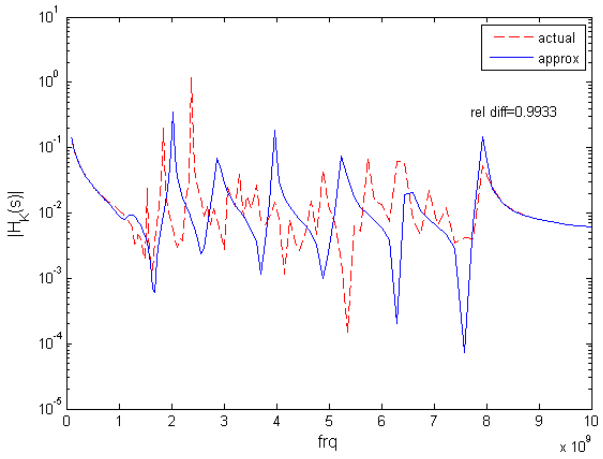
A more in-depth assessment of computational cost for this scheme compared to that of full Arnoldi is in order. A major issue to take into account is that $\widehat{H}$ and R from (7) depend on s_0. We must compute R and possibly an LU decomposition of $\widehat{H}$ every time s_0 changes.



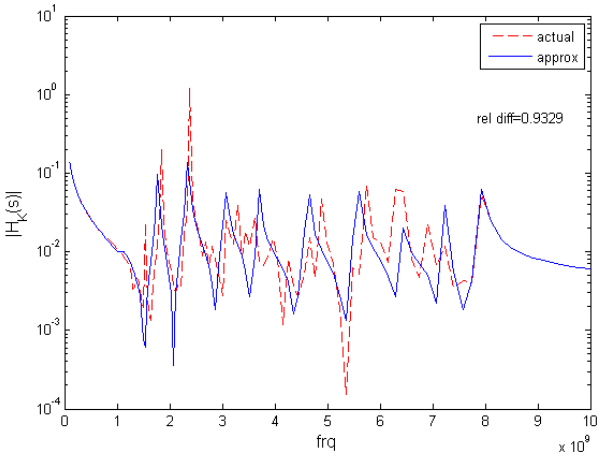
(a) Full Arnoldi,size $n = 110$ model



(b) Thick restarts ($m = 15, r = 4$), dynamic s_0, $n = 60$



(c) Thick ($m = 15, r = 4$), static s_0, $n = 60$



(d) Thick ($m = 22, r = 5$), static s_0, $n = 110$

Figure 5: A comparison of reduced models obtained via of full Arnoldi, and thick restarted Arnoldi using s_0 shifts or not. The unreduced model being approximated,`ex308s1`, is of size $N = 308$.

Proposed Exam Syllabus

Linear Algebra: Numerical Linear Algebra

- Matrix Computations, Orthogonalization, QZ Algorithm, Power Iterations, Iterative methods, Eigenvalue approximation, Krylov Subspaces
  Reference: Golub and Van Loan: *Matrix Computations*[5]
  - MAT 229A-C (now 226) *Numerical Methods: Large Scale Matrix Computations*
- Model Reduction
  Reference: *Model Order Reduction: Theory, Research Aspects and Applications*
  - MAT 229C (now 226C) *Numerical Methods: ODEs*

Analysis:

- Metric, Banach, Hilbert Spaces, Bounded Linear Operators
  Reference: *Applied Analysis*: Hunter, Nachtergaele
  - MAT 201A,B *Analysis*

Ordinary Differential Equations:

- Bifurcation Theory, Nonlinear Dynamics
  Reference: Steven H. Strogatz, *Nonlinear Dynamics and Chaos*
  - MAT 119A *Ordinary Differential Equations*

References

[1] Michael Eiermann, Oliver G. Ernst, and Stefan Güttel. Thick-restart Arnoldi methods for the evaluation of matrix functions. In *MIMS New Directions Workshop, Functions of Matrices*. University of Manchester, May 2008.

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

[3] Roland W. Freund. On pad-type model order reduction of J-hermitian linear dynamical systems. *Linear Algebra and its Applications*, 429(10):2451 – 2464, 2008. Special Issue in honor of Richard S. Varga.

[4] Roland W. Freund. Structure-preserving model order reduction of RCL circuit equations. In W.H.A. Schilders, H.A. van der Vorst, and J. Rommes, editors, *Model Order Reduction: Theory, Research Aspects and Applications*, pages 49–73, Berlin/Heidelberg, 2008. Springer-Verlag. This work was supported in part by the National Science Foundation through Grant DMS-0613032.

[5] G. H. Golub and C. F. van Loan. *Matrix Computations*. Johns Hopkins University Press, 3rd edition, 1996.

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

[7] 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.