Data Driven Model Reduction for Linear Time-Invariant Dynamical Systems

Mark Kessler and Alex Bluestein

Department of Computational and Applied Mathematics
Rice University, Houston, Texas
mak17@rice.edu, arb19@rice.edu

June & July, 2019

REU Summer 2019

Outline

Problem

Given

$\mathbf{A}, \mathbf{E} \in \mathbb{R}^{n\times n}$, $\mathbf{B} \in \mathbb{R}^{n\times m}$, $\mathbf{C} \in \mathbb{R}^{k\times n}$, and $\mathbf{D} \in \mathbb{R}^{k\times m}$

consider a system with m inputs and k outputs of the type:

$$\begin{aligned} \mathbf{E}\frac{d}{dt}\mathbf{y}(t) &= \mathbf{A}\mathbf{y}(t) + \mathbf{B}\mathbf{u}(t) \\ \mathbf{z}(t) &= \mathbf{C}\mathbf{y}(t) + \mathbf{D}\mathbf{u}(t) \end{aligned}$$

with n large (> 1000). !!Expensive to solve!!

Goal - Create a reduced order model

$\widehat{\mathbf{A}}, \widehat{\mathbf{E}} \in \mathbb{R}^{r\times r}$, $\widehat{\mathbf{B}} \in \mathbb{R}^{r\times m}$, $\widehat{\mathbf{C}} \in \mathbb{R}^{k\times r}$, and $\widehat{\mathbf{D}} \in \mathbb{R}^{k\times m}$

with r small (< 20) that approximates the original model

Example Problem: Average Room Temp.

Imagine being in charge of writing code to run a thermostat. You'd like to keep a room at a set average temperature.

Some problems:

- The dynamical system is 3D, making the model size very large
- You only care about the average temperature of the room
- The model is too large to run efficiently on a tiny chip

With model reduction:

- Model size is small
- Can be run quickly on a small chip
- No excess information
- Model still is accurate enough for effective use

Reduced Order Model Techniques

Our research focuses on:

- Interpolatory techniques
  - Projection Based
  - Loewner Method

Each technique will require working in frequency space.

To operate in frequency space, we must apply the Laplace Transform to the time-dependent system.

Outline

1D Time-Dependent Advection Diffusion Equation

Partial differential equation (PDE):

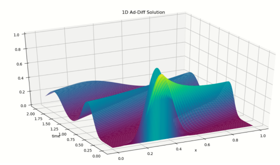
$$\begin{aligned}
\frac{\partial}{\partial t} y(x,t) - \alpha \frac{\partial^2}{\partial x^2} y(x,t) + \beta \frac{\partial}{\partial x} y(x,t) &= 0 && x \in (0,1), t \in (0,T), \\
y(0,t) &= u(t), && t \in (0,T), \\
y_x(1,t) &= 0, && t \in (0,T), \\
y(x,0) &= y_0(x), && x \in (0,1),
\end{aligned}$$

with $\alpha > 0$ and $\beta > 0$. We use

$$\alpha = 0.01, \beta = 1, T = 0.5.$$

Output:

$$z(t) = \int_0^1 y(x,t)dx$$



PDE solution

SISO Notation

Proposed system is single input ($\mathbf{u}(t) \in \mathbb{R}$) and single output ($\mathbf{z}(t) \in \mathbb{R}$)

Change of notation:

$\mathbf{A}, \mathbf{E} \in \mathbb{R}^{n \times n}$, $\mathbf{b}, \mathbf{c} \in \mathbb{R}^{n}$, and $\mathbf{d} \in \mathbb{R}$

$$\mathbf{E}\frac{d}{dt}\mathbf{y}(t) = \mathbf{A}\mathbf{y}(t) + \mathbf{b}\mathbf{u}(t)$$
$$\mathbf{z}(t) = \mathbf{c}^T\mathbf{y}(t) + \mathbf{d}\mathbf{u}(t)$$

1D Advection Diffusion Discretization

Upwind finite difference discretization in space leads to

$$\frac{d}{dt}\mathbf{y}(t) = \mathbf{A}\mathbf{y}(t) + \mathbf{b}\mathbf{u}(t), \quad t \in (0, T), \qquad \mathbf{y}(0) = \mathbf{y}_0,$$

where $\mathbf{A} = \alpha \mathbf{A}^{diff} + \beta \mathbf{A}^{conv}$

$$\mathbf{A}^{diff} = \begin{bmatrix} -2 & 1 & & & \\ 1 & -2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 2 & -2 \end{bmatrix} \mathbf{A}^{conv} = \begin{bmatrix} -1 & & & & \\ 1 & -1 & & & \\ & \ddots & \ddots & & \\ & & 1 & -1 & \\ & & & 1 & -1 \end{bmatrix}$$

$$\mathbf{y}_0 = \Big(y_0(x_0, t), \ldots, y_0(x_{n_x - 1}, t) \Big)^T \in \mathbb{R}^{n_x}$$

$$\mathbf{b} = \Big(\frac{\alpha}{h^2} + \frac{\beta}{h}, 0, \ldots, 0 \Big)^T \in \mathbb{R}^{n_x}$$

1D Advection Diffusion Discretization (cont.)

Composite Trapezoidal Rule:

$$
\begin{aligned}
\int_0^1 y(x,t)dx &\approx \frac{h}{2}y(x_0,t) + hy(x_1,t) + \ldots + hy(x_{n_x-1},t) + \frac{h}{2}y(x_n,t) \\
&= \frac{h}{2}u(t) + hy(x_1,t) + \ldots + hy(x_{n_x-1},t) + \frac{h}{2}y(x_n,t)
\end{aligned}
$$

Leads to:

$$
\begin{aligned}
\int_0^1 y(x,t)dx &\approx \frac{h}{2}u(t) + hy_1(t) + \ldots + hy_{n_x-1}(t) + \frac{h}{2}y_{n_x}(t) \\
&= \mathbf{c}^T\mathbf{y}(t) + \mathbf{d}\mathbf{u}(t)
\end{aligned}
$$

with

$$
\begin{aligned}
\mathbf{c} &= \left(h, h, \ldots, h, \frac{h}{2}\right)^T \in \mathbb{R}^{n_x} \\
\mathbf{d} &= \frac{h}{2}
\end{aligned}
$$

Laplace Transform

Transforms a function of time to function of a frequency

Let $q(t)$ be function of time where $t \in \mathbb{R}$

Denote $\mathcal{L}(q)$ as $q(s)$ where $s \in \mathbb{C}$ is frequency

Laplace Transform

Apply the Laplace Transform to the system:

$$\begin{aligned} \mathbf{E}\frac{d}{dt}\mathbf{y}(t) &= \mathbf{A}\mathbf{y}(t) + \mathbf{b}\mathbf{u}(t) \\ \mathbf{z}(t) &= \mathbf{c}^T\mathbf{y}(t) + \mathbf{d}\mathbf{u}(t) \end{aligned}$$

which turns into:

$$\begin{aligned} s\mathbf{E}\mathbf{y}(s) - \mathbf{y}(0) &= \mathbf{A}\mathbf{y}(s) + \mathbf{b}\mathbf{u}(s) \\ \mathbf{z}(s) &= \mathbf{c}^T\mathbf{y}(s) + \mathbf{d}\mathbf{u}(s) \end{aligned}$$

Laplace Transform

Assume $\mathbf{y}(0) = 0$

Rearrange and solve:

$$\mathbf{y}(s) = \left(\left(s\mathbf{E} - \mathbf{A}\right)^{-1}\mathbf{b}\right)\mathbf{u}(s)$$

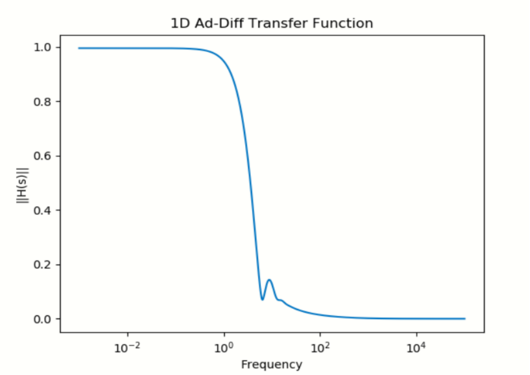
Substitute to get:

$$\mathbf{z}(s) = \left(\mathbf{c}^T\left(s\mathbf{E} - \mathbf{A}\right)^{-1}\mathbf{b} + \mathbf{d}\right)\mathbf{u}(s)$$

Which gives the following transfer function $\mathbf{H}(s)$:

$$\mathbf{H}(s) = \mathbf{c}^T\left(s\mathbf{E} - \mathbf{A}\right)^{-1}\mathbf{b} + \mathbf{d}, \quad s \in \mathbb{C}$$

1D Advection Diffusion Transfer Function



Outline

Projection Based ROM

To compute the reduced order model $\widehat{\mathbf{H}}(s)$, need to compute projection matrices $\mathbf{V}, \mathbf{W} \in \mathbb{R}^{n \times r}$ with $r << n$ such that:

$$\widehat{\mathbf{A}} = \mathbf{W^T A V}$$

$$\widehat{\mathbf{b}} = \mathbf{W^T b}$$

$$\widehat{\mathbf{c}}^T = \mathbf{c}^T \mathbf{V}$$

$$\widehat{\mathbf{E}} = \mathbf{W^T E V}$$

$$\widehat{\mathbf{H}}(s) = \widehat{\mathbf{c}}^T \left(s\widehat{\mathbf{E}} - \widehat{\mathbf{A}} \right)^{-1} \widehat{\mathbf{b}} + \widehat{\mathbf{d}}$$

Given a set of training frequencies $\{\sigma_1, \dots \sigma_r\} \subset \mathbb{I}$, want to find $\mathbf{V}, \mathbf{W}$ such that $\mathbf{H}(\sigma) = \widehat{\mathbf{H}}(\sigma)$ for all the training frequencies

How to compute $\mathbf{V}$ and $\mathbf{W}$?

Projection Based ROM

Theorem

Let $\sigma \in \mathbb{C}$ be such that $\big(\sigma\mathbf{E} - \mathbf{A}\big)$ and $\big(\sigma\widehat{\mathbf{E}} - \widehat{\mathbf{A}}\big)$ are non-singular. Also, let $\mathbf{V}, \mathbf{W} \in \mathbb{C}^{n\times r}$ have full rank. Then,

1. *If $\big(\sigma\mathbf{E} - \mathbf{A}\big)^{-1}\mathbf{b} \in \mathcal{R}\big(\mathbf{V}\big)$, then $\mathbf{H}(\sigma) = \widehat{\mathbf{H}}(\sigma)$*
2. *If $\big(\sigma\mathbf{E}^T - \mathbf{A}^T\big)^{-1}\mathbf{c} \in \mathcal{R}\big(\mathbf{W}\big)$, then $\mathbf{H}(\sigma) = \widehat{\mathbf{H}}(\sigma)$*
3. *If (1) and (2) hold, then $\mathbf{H}'(\sigma) = \widehat{\mathbf{H}}'(\sigma)$*

Proof of Interpolation Property

Assume $\left(\sigma\mathbf{E}-\mathbf{A}\right)^{-1}\mathbf{b} \in \mathcal{R}\left(\mathbf{V}\right)$

Define:

$$\mathbb{P}(s) = \mathbf{V}\Big(s\mathbf{E}-\mathbf{A}\Big)^{-1}\mathbf{W}^{\mathbf{T}}\Big(s\mathbf{E}-\mathbf{A}\Big)$$
$$\mathbb{Q}(s) = \Big(s\mathbf{E}-\mathbf{A}\Big)\mathbf{V}\Big(s\mathbf{E}-\mathbf{A}\Big)^{-1}\mathbf{W}^{\mathbf{T}}$$

$\mathbb{P}(s)^2 = \mathbb{P}(s)$ and $\mathbb{Q}(s)^2 = \mathbb{Q}(s) \implies \mathbb{P}(s)$ and $\mathbb{Q}(s)$ are projections.

Also, make note that $\mathcal{R}\big(\mathbf{V}\big) = \mathcal{R}\big(\mathbb{P}(\sigma)\big)$ and $\mathcal{R}\big(\mathbf{W}\big) = \mathcal{R}\big(\mathbf{I}-\mathbb{Q}(\sigma)\big)$.

Proof of Interpolation Property

Consider the identity:

$$\mathbf{H}(s)-\widehat{\mathbf{H}}(s) = \mathbf{c}^T\left(s\mathbf{E}-\mathbf{A}\right)^{-1}\left(\mathbf{I}-\mathbb{Q}(s)\right)\left(s\mathbf{E}-\mathbf{A}\right)\left(\mathbf{I}-\mathbb{P}(s)\right)\left(s\mathbf{E}-\mathbf{A}\right)^{-1}\mathbf{b}$$

Evaluate at σ, to get:

$$\mathbf{H}(s)-\widehat{\mathbf{H}}(s) = \dots\left(\mathbf{I}-\mathbb{P}(\sigma)\right)\left(\sigma\mathbf{E}-\mathbf{A}\right)^{-1}\mathbf{b}$$

By assumption, $\left(\sigma\mathbf{E}-\mathbf{A}\right)^{-1}\mathbf{b} \in \mathcal{R}\left(\mathbf{V}\right)$ or $\in \mathcal{N}\left(\mathbf{I}-\mathbb{P}(\sigma)\right)$

So:

$$\mathbf{H}(\sigma)-\widehat{\mathbf{H}}(\sigma) = 0 \implies \mathbf{H}(\sigma) = \widehat{\mathbf{H}}(\sigma)$$

Proof of part (2) follows similarly.

Proof of Interpolation Property

Assume parts (1) and (2) of the theorem hold

Consider the identity from before:

$$\mathbf{H}(s) - \widehat{\mathbf{H}}(s) = \mathbf{c}^T \Big(s\mathbf{E} - \mathbf{A}\Big)^{-1} \Big(\mathbf{I} - \mathbb{Q}(s)\Big) \Big(s\mathbf{E} - \mathbf{A}\Big) \Big(\mathbf{I} - \mathbb{P}(s)\Big) \Big(s\mathbf{E} - \mathbf{A}\Big)^{-1} \mathbf{b}$$

Evaluate at $s = \sigma + \epsilon$:

$$\mathbf{H}(\sigma + \epsilon) - \widehat{\mathbf{H}}(\sigma + \epsilon) = \mathcal{O}(\epsilon^2)$$

Since $\mathbf{H}(\sigma) = \widehat{\mathbf{H}}(\sigma)$ by the assumption,

$$\lim_{\epsilon \to 0} \frac{1}{\epsilon}\Big(\mathbf{H}(\sigma + \epsilon) - \mathbf{H}(\sigma)\Big) - \frac{1}{\epsilon}\Big(\widehat{\mathbf{H}}(\sigma + \epsilon) - \widehat{\mathbf{H}}(\sigma)\Big) = 0$$

which proves part (3) of the theorem.

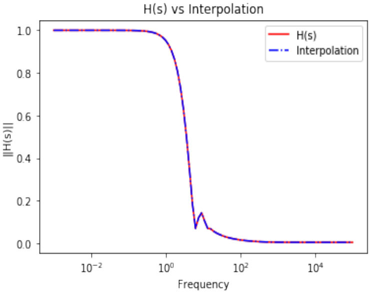
Projection Matrix Generation

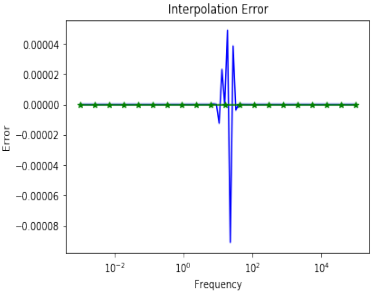
Choose a list of interpolation frequencies, $[\sigma_1, \sigma_2, ..., \sigma_r]$.

Then let us construct the projection matrices and as follows:

$$\mathbf{V} = \left[\left(\sigma_1 \mathbf{E} - \mathbf{A}\right)^{-1} \mathbf{b}, \ldots, \left(\sigma_r \mathbf{I} - \mathbf{A}\right)^{-1} \mathbf{E}\right]$$
$$\mathbf{W} = \left[\left(\sigma_1 \mathbf{E}^T - \mathbf{A}^T\right)^{-1} \mathbf{c}, \ldots, \left(\sigma_r \mathbf{E}^T - \mathbf{A}^T\right)^{-1} \mathbf{c}\right]$$

Projection ROM Performance





Projection ROM: Complex to Real

Since σ_j is complex, $\mathbf{V}$ and $\mathbf{W} \in \mathbb{C}$.
The original problem is in the time domain.

- Need real matrices $\widehat{\mathbf{A}}$, $\widehat{\mathbf{b}}$, $\widehat{\mathbf{c}}$, $\widehat{\mathbf{d}}$, and $\widehat{\mathbf{E}}$
- To get $\mathbf{A} \in \mathbb{R}^{n \times n}, \mathbf{b} \in \mathbb{R}^{n}$, etc., need real valued **V**,**W**

- Problem:
  - How do we retain the info in **V**,**W**?

Split into their real and imaginary parts?

Projection ROM: Complex to Real

- Split into real and imaginary parts:

Let $\mathbf{V} = \mathbf{V_a} + i\mathbf{V_b}$ and $\mathbf{W} = \mathbf{W}_a + i\mathbf{W}_b$.

Choose $\mathbf{V_{real}} = [\mathbf{V_a}\mathbf{V_b}]$ and $\mathbf{W_{real}} = [\mathbf{W_a}\mathbf{W_b}]$.

- Problem:
  - **V**,**W** may be singular

Solution? SVD!

Projection ROM: Complex to Real

SVD purpose =

- remove dependent information
- ensure non-singularity of $\left(\sigma\widehat{\mathbf{E}} - \widehat{\mathbf{A}}\right)$

Let $\mathbf{U_v \Sigma_v V_v^T} = [\mathbf{V_a V_b}]$ and $\mathbf{U_w \Sigma_w V_w^T} = [\mathbf{W_a W_b}]$.

Given a tolerance for the normalized singular values, let $k_v, k_w =$ number of normalized singular values $>$ the tolerance. Set k = max(k_v, k_w).

Then $\mathbf{V_{real}} = \mathbf{U_{v_k}}$ and $\mathbf{W_{real}} = \mathbf{U_{w_k}}$

Loewner Method Motivation

The projection based ROM works very well, as proven by the error plot. What do we need the Loewner Method for?

Reasons:

- To use the projection method, need $\mathbf{A}$,$\mathbf{b}$,$\mathbf{c}$,$\mathbf{d}$, and $\mathbf{E}$
- In some applications, matrices not given
- Can be used with only transfer function data
- Retains interpolation property, mimics projection method

Downsides:

- Loewner matrices might become singular if too many sample frequencies are used
- Which frequencies to sample data from?
- Needs $\mathbf{d}$ in FOM

Loewner Data Setup

First, lets assume that in the FOM $\mathbf{d} = 0$

Left frequencies:

$$\mathbf{\Sigma} = \{\sigma_1, \ldots, \sigma_r\} \subset \mathbb{I}$$

Right frequencies:

$$\mathbf{\Theta} = \{\theta_1, \ldots, \theta_r\} \subset \mathbb{I}$$

with

$$\mathbf{\Sigma} \cap \mathbf{\Theta} = \emptyset$$

Loewner Matrices and ROM

Construct the Loewner Matrix:

$$\mathbb{L}_{ij} = \frac{\mathbf{H}(\sigma_i) - \mathbf{H}(\theta_j)}{\sigma_i - \theta_j}$$

Construct the Shifted Loewner Matrix:

$$\mathbb{M}_{ij} = \frac{\sigma_i \mathbf{H}(\sigma_i) - \theta_j \mathbf{H}(\theta_j)}{\sigma_i - \theta_j}$$

Loewner Matrices and ROM

Construct the Loewner ROM:

$$
\begin{aligned}
\widehat{\mathbf{A}} &= -\mathbb{M} \\
\widehat{\mathbf{b}} &= [\mathbf{H}(\sigma_1), \ldots, \mathbf{H}(\sigma_r)]^T \\
\widehat{\mathbf{c}}^T &= [\mathbf{H}(\theta_1), \ldots, \mathbf{H}(\theta_r)] \\
\widehat{\mathbf{E}} &= -\mathbb{L} \\
\widehat{\mathbf{d}} &= 0
\end{aligned}
$$

$$
\widehat{\mathbf{H}}(s) = \widehat{\mathbf{c}}^T \left(s\widehat{\mathbf{E}} - \widehat{\mathbf{A}}\right)^{-1} \widehat{\mathbf{b}} + \widehat{\mathbf{d}}
$$

Loewner Interpolation Theorem

Theorem

Assume $\mathbb{M} - s\mathbb{L}$ *is invertible for all* $s \in \Sigma \cup \Theta$.

Then, the previously described ROM transfer function interpolates at the left and right training frequencies:

$\mathbf{H}(s) = \widehat{\mathbf{H}}(s)$ *for all* $s \in \Sigma \cup \Theta$

Loewner Interpolation Property Proof

Simplify the constructed Loewner ROM

$$\widehat{\mathbf{H}}(s) = \widehat{\mathbf{c}}^T \left(s\widehat{\mathbf{E}} - \widehat{\mathbf{A}}\right)^{-1} \widehat{\mathbf{b}}$$
$$\widehat{\mathbf{H}}(s) = \left[\mathbf{H}(\theta_1), \ldots, \mathbf{H}(\theta_r)\right] (\mathbb{M} - s\mathbb{L})^{-1} \left[\mathbf{H}(\sigma_1), \ldots, \mathbf{H}(\sigma_r)\right]^T$$

Consider the matrix $\mathbb{M} - s\mathbb{L}$ where $s \in \mathbb{C}$.

$$(\mathbb{M} - s\mathbb{L})_{i,j} = \frac{s - \theta_j}{\sigma_i - \theta_j}\mathbf{H}(\theta_j) + \frac{\sigma_i - s}{\sigma_i - \theta_j}\mathbf{H}(\sigma_i)$$

Two interpolatory cases for s:

1. $s = \theta_k$ for some k
2. $s = \sigma_k$ for some k

Loewner Interpolation Property Proof

Consider case where $s = \theta_k$.

Recall:

$$(\mathbb{M} - s\mathbb{L})_{i,j} = \frac{s - \theta_j}{\sigma_i - \theta_j}\mathbf{H}(\theta_j) + \frac{\sigma_i - s}{\sigma_i - \theta_j}\mathbf{H}(\sigma_i)$$

$$(\mathbb{M} - \theta_k\mathbb{L})e_k = [\mathbf{H}(\sigma_1), \ldots, \mathbf{H}(\sigma_r)]^T \implies$$
$$e_k = (\mathbb{M} - \theta_k\mathbb{L})^{-1}[\mathbf{H}(\sigma_1), \ldots, \mathbf{H}(\sigma_r)]^T$$

$$[\mathbf{H}(\theta_1), \ldots, \mathbf{H}(\theta_r)]\, e_k = \mathbf{H}(\theta_k)$$
$$[\mathbf{H}(\theta_1), \ldots, \mathbf{H}(\theta_r)]\,(\mathbb{M} - \theta_k\mathbb{L})^{-1}[\mathbf{H}(\sigma_1), \ldots, \mathbf{H}(\sigma_r)]^T = \widehat{\mathbf{H}}(\theta_k)$$

Therefore, $\mathbf{H}(\theta_k) = \widehat{\mathbf{H}}(\theta_k)$

Loewner Interpolation Property Proof

Consider case where $s = \sigma_k$.

Recall:

$$(\mathbb{M} - s\mathbb{L})_{i,j} = \frac{s - \theta_j}{\sigma_i - \theta_j}\mathbf{H}(\theta_j) + \frac{\sigma_i - s}{\sigma_i - \theta_j}\mathbf{H}(\sigma_i)$$

$$e_k^T(\mathbb{M} - \sigma_k\mathbb{L}) = [\mathbf{H}(\theta_1), \ldots, \mathbf{H}(\theta_r)] \implies$$
$$e_k^T = [\mathbf{H}(\theta_1), \ldots, \mathbf{H}(\theta_r)]\,(\mathbb{M} - \sigma_k\mathbb{L})^{-1}$$

$$e_k^T\,[\mathbf{H}(\sigma_1), \ldots, \mathbf{H}(\sigma_r)]^T = \mathbf{H}(\sigma_k)$$
$$[\mathbf{H}(\theta_1), \ldots, \mathbf{H}(\theta_r)]\,(\mathbb{M} - \sigma_k\mathbb{L})^{-1}\,[\mathbf{H}(\sigma_1), \ldots, \mathbf{H}(\sigma_r)]^T = \widehat{\mathbf{H}}(\sigma_k)$$

Therefore, $\mathbf{H}(\sigma_k) = \widehat{\mathbf{H}}(\sigma_k)$

Dealing with Singular Loewner Matrices

Necessary that Loewner matrices are nonsingular

- More frequencies leads to singular Loewner matrices
- ROM transfer function will be undefined

Use SVD to extract independent columns of matrices:

$$\mathbf{U_1 \Sigma_1 V_1^*} = [\mathbb{L}\ \mathbb{M}]$$
$$\mathbf{U_2 \Sigma_2 V_2^*} = [\mathbb{L}\ \mathbb{M}]^T$$

Given a tolerance for the normalized singular values, let $k_1, k_2 =$ number of normalized singular values $>$ the tolerance. Set k $=$ max(k_v, k_w).

Let $\mathbf{U} = \mathbf{U_{1k}}$ and $\mathbf{V} = \mathbf{V_{2k}}$.

Dealing with Singular Loewner Matrices

Construct the new Loewner ROM as:

$$
\begin{aligned}
\widehat{\mathbf{A}} &= -\mathbf{U}^* \mathbb{M} \mathbf{V} \\
\widehat{\mathbf{b}} &= \mathbf{U}^* \left[\mathbf{H}(\sigma_1), \ldots, \mathbf{H}(\sigma_r)\right]^T \\
\widehat{\mathbf{c}}^T &= \left[\mathbf{H}(\theta_1), \ldots, \mathbf{H}(\theta_r)\right] \mathbf{V} \\
\widehat{\mathbf{E}} &= -\mathbf{U}^* \mathbb{L} \mathbf{V} \\
\widehat{\mathbf{d}} &= 0
\end{aligned}
$$

$$\widehat{\mathbf{H}}(s) = \widehat{\mathbf{c}}^T \left(s\widehat{\mathbf{E}} - \widehat{\mathbf{A}}\right)^{-1} \widehat{\mathbf{b}} + \widehat{\mathbf{d}}$$

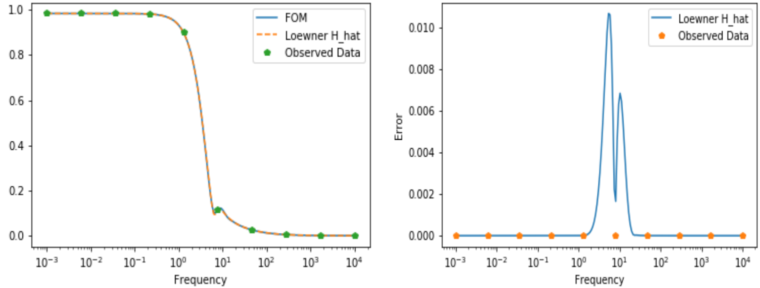
Interpolatory properties of the new Loewner ROM are beyond the scope of this presentation

Lowner Framework Implementation

1. Choose disjoint imaginary left and right frequencies $\{\sigma_1, \ldots, \sigma_r\}$ and $\{\theta_1, \ldots, \theta_r\}$, respectively
2. Compute $\mathbb{L}$ and $\mathbb{M}$
   - $\mathbb{L}_{ij} = \frac{\mathbf{H}(\sigma_i) - \mathbf{H}(\theta_j)}{\sigma_i - \theta_j}$
   - $\mathbb{M}_{ij} = \frac{\sigma_i \mathbf{H}(\sigma_i) - \theta_j \mathbf{H}(\theta_j)}{\sigma_i - \theta_j}$
3. Compute $\mathbf{U}$ and $\mathbf{V}$ from the SVD of the Loewner pencils
4. Use the Loewner matrices, $\mathbf{U}$, $\mathbf{V}$, and the left and right transfer function data to compute the ROM

Actual H(s) vs Loewner H(s)

Using 5 left and 5 right frequencies:



Converting Loewner to Real

Since the right and left frequencies are complex, the ouput of the transfer function at these frequencies will be complex as well. This leads to complex Loewner matrices.

The original problem is in the time domain.

- Need real matrices $\widehat{\mathbf{A}}$, $\widehat{\mathbf{b}}$, $\widehat{\mathbf{c}}$, $\widehat{\mathbf{d}}$, and $\widehat{\mathbf{E}}$

Compute real $\mathbb{L}$ and $\mathbb{M}$

- Problem:
  - How do we retain the info in $\mathbb{L}$, $\mathbb{M}$?

Utilize the fact that $\overline{\mathbf{H}(\mathbf{s})} = \mathbf{H}(\overline{\mathbf{s}})$.

Converting Loewner to Real

First include the complex conjugate data in the setup.

Left frequencies:

$$\mathbf{\Sigma} = \{\sigma_1, \ldots, \sigma_{2r}\} \subset \mathbb{I}$$

where $\sigma_{2i} = \overline{\sigma}_{2i-1}$

Right frequencies:

$$\mathbf{\Theta} = \{\theta_1, \ldots, \theta_{2r}\} \subset \mathbb{I}$$

where $\theta_{2i} = \overline{\theta}_{2i-1}$

Converting Loewner to Real

Let $\mathbf{J} \in \mathbb{C}^{n \times n}$ be a block diagonal matrix with $\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & -i \\ 1 & i \end{bmatrix}$ repeated on the diagonal.

Use the following real Loewner matrices to compute $\mathbf{U}_R$ and $\mathbf{V}_R$:

$$\mathbb{L}_R = \mathbf{J}^* \mathbb{L} \mathbf{J}$$
$$\mathbb{M}_R = \mathbf{J}^* \mathbb{M} \mathbf{J}$$

Converting Loewner to Real

Construct the real Loewner ROM as:

$$
\begin{aligned}
\widehat{\mathbf{A}} &= -\mathbf{U}_R^T \mathbb{M}_R \mathbf{V}_R \\
\widehat{\mathbf{b}} &= \mathbf{U}^T \mathbf{J}^* \left[\mathbf{H}(\sigma_1), \ldots, \mathbf{H}(\sigma_r)\right]^T \\
\widehat{\mathbf{c}}^T &= \left[\mathbf{H}(\theta_1), \ldots, \mathbf{H}(\theta_r)\right] \mathbf{J}\mathbf{V} \\
\widehat{\mathbf{E}} &= -\mathbf{U}_R^T \mathbb{L}_R \mathbf{V}_R \\
\widehat{\mathbf{d}} &= 0
\end{aligned}
$$

$$
\widehat{\mathbf{H}}(s) = \widehat{\mathbf{c}}^T \left(s\widehat{\mathbf{E}} - \widehat{\mathbf{A}}\right)^{-1} \widehat{\mathbf{b}} + \widehat{\mathbf{d}}
$$

How does this work?

Consider computing $\mathbb{L}_R$ in the case where $\Sigma = \{\sigma, \overline{\sigma}\}$ and $\Theta = \{\theta, \overline{\theta}\}$:

$$\mathbb{L}_R = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ i & -i \end{bmatrix} \begin{bmatrix} \frac{\mathbf{H}(\sigma)-\mathbf{H}(\theta)}{\sigma-\theta} & \frac{\overline{\mathbf{H}(\sigma)}-\mathbf{H}(\theta)}{\overline{\sigma}-\theta} \\ \frac{\mathbf{H}(\sigma)-\overline{\mathbf{H}(\theta)}}{\sigma-\overline{\theta}} & \frac{\overline{\mathbf{H}(\sigma)}-\overline{\mathbf{H}(\theta)}}{\overline{\sigma}-\overline{\theta}} \end{bmatrix} \begin{bmatrix} 1 & -i \\ 1 & i \end{bmatrix}$$

$$\mathbb{L}_R = \begin{bmatrix} re(\frac{\mathbf{H}(\sigma)-\mathbf{H}(\theta)}{\sigma-\theta}) & im(\frac{\mathbf{H}(\sigma)-\mathbf{H}(\theta)}{\sigma-\theta}) \\ -im(\frac{\mathbf{H}(\sigma)-\mathbf{H}(\theta)}{\sigma-\theta}) & re(\frac{\mathbf{H}(\sigma)-\mathbf{H}(\theta)}{\sigma-\theta}) \end{bmatrix}$$

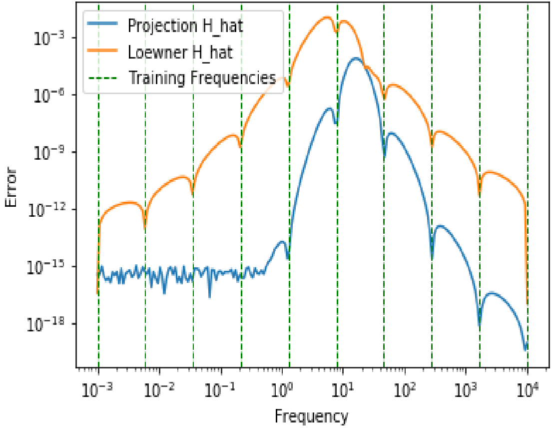
The matrix $\mathbb{L}_R$ is real but encapsulates the same information as the complex number $\frac{\mathbf{H}(\sigma)-\mathbf{H}(\theta)}{\sigma-\theta}$.

Same holds for $\mathbb{M}_R$.

Comparison of Methods

- 10 training frequencies logarithmically spaced between 10^{-3} and 10^{4}
- Tolerance of 10^{-12} for cutting off SVD's in both methods

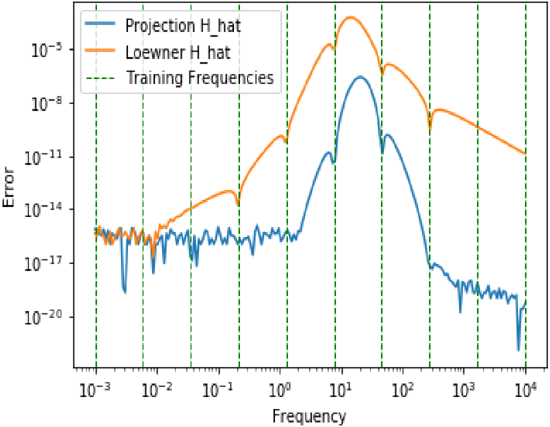
- Size of ROMs:
  - Loewner Framework - 5 by 5
  - Projection - 10 by 10



Comparison of Methods

- Covert both ROMs to real domain

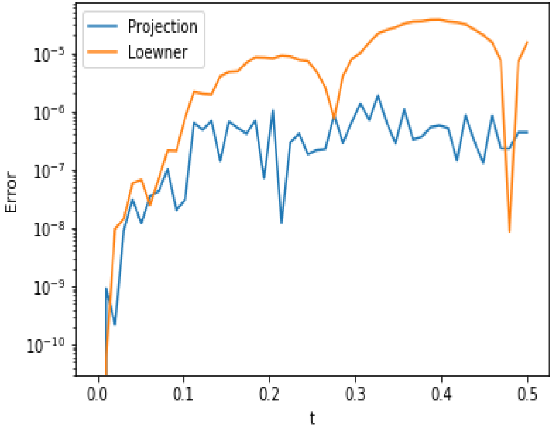
- Size of ROMs:
  - Loewner Framework - 8 by 8
  - Projection - 15 by 15



Comparison of Methods

Compare ROM methods to the FOM in the time domain:

Let $u(t) = sin(2\pi t)$, for $0 \leq t \leq 0.5$



Outline

Further Research

- Loewner with noise
  - In most data collection, there is error. In our experiments, noise $\Longrightarrow$ large error. Error correction techniques may be necessary to develop.
- Generalized eigenvalue problem and error spikes
  - Closely related to the first problem. With noise, the eigenvalues shift closer to the imaginary axis, causing problems. As a result the Loewner matrices become singular.
- Performace optimization (Loewner Method)
  - It would be interesting to explore optimal selection of frequencies and other related factors that could influence results.

Outline

Resources

[1] A.C. Antoulas, C. Beattie, S. Gugercin, *Interpolatory Methods for Model Reduction*, pg. 33-85, June 14, 2019

[2] M. Heinkenschloss, a big thanks for mentoring this project!