MAX PLANCK INSTITUTE
FOR DYNAMICS OF COMPLEX
TECHNICAL SYSTEMS
MAGDEBURG

CSC
COMPUTATIONAL METHODS IN
SYSTEMS AND CONTROL THEORY

Implicit and Explicit Matching of Non-Proper Transfer Functions in the Loewner Framework

Jan Heiland
joint work with Ion Victor Gosea

ECC24 European Control Conference
Stockholm, Sweden

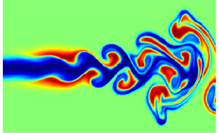
June 28th, 2024

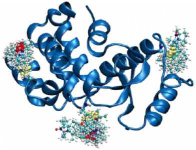
Outline

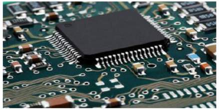
1. Reduced Order Modelling in Frequency Domain
2. Data-driven ROM (Frequency Domain)
3. Rational Interpolation and the Loewner Matrix
4. The AAA Algorithm
5. Loewner and AA With Implicitly Defined Polynomial Parts

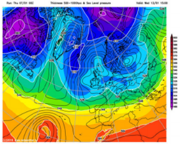
Reduced Order Modelling in Frequency Domain

- **Model Order Reduction** (**MOR**) is used to transform large, complex models of time-dependent processes into smaller, simpler models that are still capable of representing accurately the behavior of the original process.



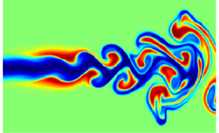


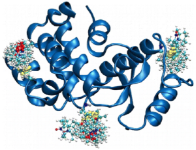


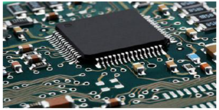


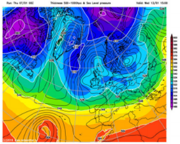
Reduced Order Modelling in Frequency Domain

- **Model Order Reduction** (**MOR**) is used to transform large, complex models of time-dependent processes into smaller, simpler models that are still capable of representing accurately the behavior of the original process.





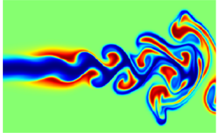
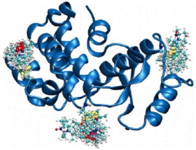
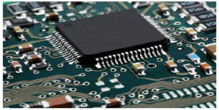
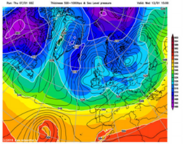




- Data-driven reduced-order modeling (DD-ROM) and learning methods have become more and more appealing and sought after over the years.

Reduced Order Modelling in Frequency Domain

- **Model Order Reduction** (**MOR**) is used to transform large, complex models of time-dependent processes into smaller, simpler models that are still capable of representing accurately the behavior of the original process.

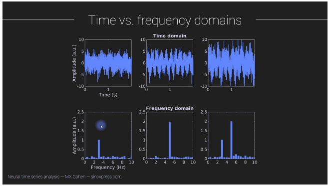





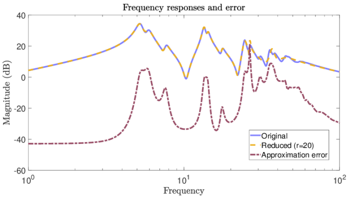
- Data-driven reduced-order modeling (DD-ROM) and learning methods have become more and more appealing and sought after over the years.
- At MPI Magdeburg, we study and develop different DD-ROM + learning methods:

  $\rightarrow$ Operator Inference (OpInf), Loewner Framework, AAA, DMD, or QuadBT

  $\rightarrow$ Typical learning methods (with ANNs: LQResNet, CNNs, RNNs), SINDy, ...

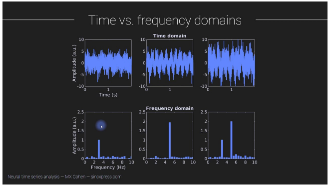
Reduced Order Modelling in Frequency Domain

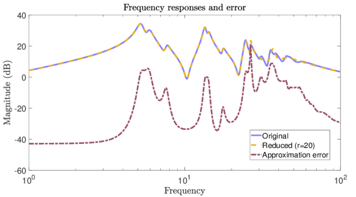




- In electronics or control systems engineering, the frequency domain refers to the analysis of mathematical functions or signals with respect to frequency.
- A frequency response describes the steady-state response of a system to sinusoidal inputs of varying frequencies $\rightsquigarrow$ can be measured in practice (VNAs, EIS, etc.).

Reduced Order Modelling in Frequency Domain

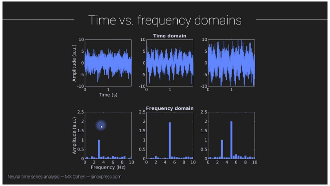


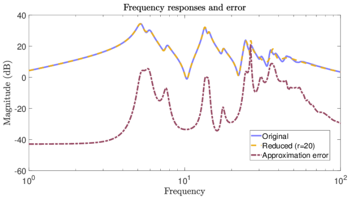


- In electronics or control systems engineering, the frequency domain refers to the analysis of mathematical functions or signals with respect to frequency.
- A frequency response describes the steady-state response of a system to sinusoidal inputs of varying frequencies $\rightsquigarrow$ can be measured in practice (VNAs, EIS, etc.).
- By applying the (unilateral) Laplace transform to the system of ODEs/DAEs:

$$\begin{cases} \mathcal{L}\left[\mathbf{E}\dot{\mathbf{x}}(t)\right] = \mathcal{L}\left[\mathbf{A}\mathbf{x}(t)\right] + \mathcal{L}\left[\mathbf{B}\mathbf{u}(t)\right], \\ \mathcal{L}\left[\mathbf{y}(t)\right] = \mathcal{L}\left[\mathbf{C}\mathbf{x}(t)\right] + \mathcal{L}\left[\mathbf{D}\mathbf{u}(t)\right], \end{cases} \Rightarrow \begin{cases} s\mathbf{E}\hat{\mathbf{x}}(s) - \mathbf{x}_0 = \mathbf{A}\hat{\mathbf{x}}(s) + \mathbf{B}\hat{\mathbf{u}}(s), \\ \hat{\mathbf{y}}(s) = \mathbf{C}\hat{\mathbf{x}}(s) + \mathbf{D}\hat{\mathbf{u}}(s). \end{cases}$$

Reduced Order Modelling in Frequency Domain





- In electronics or control systems engineering, the frequency domain refers to the analysis of mathematical functions or signals with respect to frequency.
- A frequency response describes the steady-state response of a system to sinusoidal inputs of varying frequencies $\rightsquigarrow$ can be measured in practice (VNAs, EIS, etc.).
- By applying the (unilateral) Laplace transform to the system of ODEs/DAEs:

$$\begin{cases} \mathcal{L}\left[\mathbf{E}\dot{\mathbf{x}}(t)\right] = \mathcal{L}\left[\mathbf{A}\mathbf{x}(t)\right] + \mathcal{L}\left[\mathbf{B}\mathbf{u}(t)\right], \\ \mathcal{L}\left[\mathbf{y}(t)\right] = \mathcal{L}\left[\mathbf{C}\mathbf{x}(t)\right] + \mathcal{L}\left[\mathbf{D}\mathbf{u}(t)\right], \end{cases} \Rightarrow \begin{cases} s\mathbf{E}\hat{\mathbf{x}}(s) - \mathbf{x}_0 = \mathbf{A}\hat{\mathbf{x}}(s) + \mathbf{B}\hat{\mathbf{u}}(s), \\ \hat{\mathbf{y}}(s) = \mathbf{C}\hat{\mathbf{x}}(s) + \mathbf{D}\hat{\mathbf{u}}(s). \end{cases}$$

- By solving the algebraic equation above in terms $\hat{\mathbf{x}}(s)$, we get that:

$$\hat{\mathbf{y}}(s) = \underbrace{\left[\mathbf{C}(s\mathbf{E} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D}\right]}_{\text{the transfer function } \mathbf{H}(s)} \hat{\mathbf{u}}(s). \tag{1}$$

Data-driven ROM (Frequency Domain)

- What if we don't have direct access to the model (only frequency response data is provided)?

Data-driven ROM (Frequency Domain)

- What if we don't have direct access to the model (only frequency response data is provided)?
- Use available measurements and apply data-driven approaches:
  1. Vector fitting - [Gustavsen/Semlyen '99]; [Drmac/Gugercin/Beattie '15];
  2. The RKFIT algorithm - [Berljafa/Güttel '17] (RK toolbox);
  3. The AAA algorithm - [Nakatsukasa/Sete/Trefethen '18] (Chebfun toolbox);
  4. The Loewner framework - [Mayo/Antoulas '07]; [Antoulas/Lefteriu/Ionita '17] . . .

Data-driven ROM (Frequency Domain)

- What if we don't have direct access to the model (only frequency response data is provided)?
- Use available measurements and apply data-driven approaches:
  1. Vector fitting - [Gustavsen/Semlyen '99]; [Drmac/Gugercin/Beattie '15];
  2. The RKFIT algorithm - [Berljafa/Güttel '17] (RK toolbox);
  3. The AAA algorithm - [Nakatsukasa/Sete/Trefethen '18] (Chebfun toolbox);
  4. The Loewner framework - [Mayo/Antoulas '07]; [Antoulas/Lefteriu/Ionita '17] . . .
- Use data samples to construct an approximated fitting model:
  1. Given by matrices $\mathbf{E}, \mathbf{A}, \mathbf{B}, \mathbf{C}$ and TF: $\mathbf{H}(s) = \mathbf{C}(s\mathbf{E} - \mathbf{A})^{-1}\mathbf{B}$.
  2. Given in barycentric representation: $\mathbf{H}(s) = \frac{\sum_{k=0}^{d} \frac{w_k f_k}{s-\xi_k}}{\sum_{k=0}^{d} \frac{w_k}{s-\xi_k}}$.
  3. Given in pole-residue representation: $\mathbf{H}(s) = \eta_0 + \sum_{k=1}^{d} \frac{\beta_k}{s-\xi_k}$.

Rational Interpolation and the Loewner Matrix

- **Lagrange basis** for the linear space of polynomials of degree at most n.

  Given $\lambda_i \in \mathbb{C}$, $i = 1, \cdots, n+1$: $\lambda_i \neq \lambda_j$, $i \neq j$,

$$\mathbf{q}_i(s) := \Pi_{i' \neq i} (s - \lambda_{i'}), \; i = 1, \cdots, n+1,$$

Rational Interpolation and the Loewner Matrix

- **Lagrange basis** for the linear space of polynomials of degree at most n.

  Given $\lambda_i \in \mathbb{C}$, $i = 1, \cdots, n+1$: $\lambda_i \neq \lambda_j$, $i \neq j$,

$$\mathbf{q}_i(s) := \Pi_{i' \neq i} (s - \lambda_{i'}), \; i = 1, \cdots, n+1,$$

- The **barycentric Lagrange interpolation** for a rational function $g(s)$ satisfying the (right) interpolation conditions for any $1 \leq i \leq n+1$

$$\mathbf{g}(\lambda_i) = \mathbf{z}_i,$$

  is given by:

$$\boxed{\mathbf{g}(s) = \frac{\sum_{i=1}^{n+1} \frac{w_i \mathbf{z}_i}{s - \lambda_i}}{\sum_{i=1}^{n+1} \frac{w_i}{s - \lambda_i}}} = \frac{\sum_i t_i \, \mathbf{q}_i(s)}{\sum_i w_i \, \mathbf{q}_i(s)}, \quad t_i = w_i \, \mathbf{z}_i.$$

Rational Interpolation and the Loewner Matrix

- **Lagrange basis** for the linear space of polynomials of degree at most n.

  Given $\lambda_i \in \mathbb{C}$, $i = 1, \cdots, n+1$: $\lambda_i \neq \lambda_j$, $i \neq j$,

$$\mathbf{q}_i(s) := \Pi_{i' \neq i}(s - \lambda_{i'}), \; i = 1, \cdots, n+1,$$

- The **barycentric Lagrange interpolation** for a rational function $g(s)$ satisfying the (right) interpolation conditions for any $1 \leq i \leq n+1$

$$\mathbf{g}(\lambda_i) = \mathbf{z}_i,$$

  is given by:

$$\boxed{\mathbf{g}(s) = \frac{\sum_{i=1}^{n+1} \frac{w_i \mathbf{z}_i}{s - \lambda_i}}{\sum_{i=1}^{n+1} \frac{w_i}{s - \lambda_i}}} = \frac{\sum_i t_i \, \mathbf{q}_i(s)}{\sum_i w_i \, \mathbf{q}_i(s)}, \quad t_i = w_i \, \mathbf{z}_i.$$

- The free parameters (weights) w_i are found so that *additional* interpolation conditions hold:

$$\mathbf{g}(\mu_j) = \mathbf{v}_j, \; j = 1, \cdots, r,$$

  where $(\mu_j, \mathbf{v}_j)$, with $\mu_i \neq \mu_j$, are given.

J. P. Berrut and N. Trefethen, *Barycentric Lagrange Interpolation*, SIAM Review, 2004.

Rational Interpolation and the Loewner Matrix

- For these extra conditions to be satisfied, one needs to enforce $\mathbb{L}\,\mathbf{c} = 0$, where

$$\mathbb{L} = \begin{bmatrix} \frac{\mathbf{v}_1 - \mathbf{z}_1}{\mu_1 - \lambda_1} & \cdots & \frac{\mathbf{v}_1 - \mathbf{z}_{n+1}}{\mu_1 - \lambda_{n+1}} \\ \vdots & \ddots & \vdots \\ \frac{\mathbf{v}_r - \mathbf{z}_1}{\mu_r - \lambda_1} & \cdots & \frac{\mathbf{v}_r - \mathbf{z}_{n+1}}{\mu_r - \lambda_{n+1}} \end{bmatrix} \in \mathbb{C}^{r\times(n+1)}, \quad \mathbf{c} = \begin{bmatrix} w_1 \\ \vdots \\ w_{n+1} \end{bmatrix} \in \mathbb{C}^{n+1}.$$

- Here, $\mathbb{L}$ is a **Loewner matrix** (from Charles Loewner) with:

  left (row) array $(\mu_j, \mathbf{v}_j)$, $j = 1, \ldots, r$, and **right (column) array** $(\lambda_i, \mathbf{z}_i)$, $i = 1, \ldots, n+1$.

Rational Interpolation and the Loewner Matrix

- For these extra conditions to be satisfied, one needs to enforce $\mathbb{L}\,\mathbf{c} = 0$, where

$$\mathbb{L} = \begin{bmatrix} \frac{\mathbf{v}_1 - \mathbf{z}_1}{\mu_1 - \lambda_1} & \cdots & \frac{\mathbf{v}_1 - \mathbf{z}_{n+1}}{\mu_1 - \lambda_{n+1}} \\ \vdots & \ddots & \vdots \\ \frac{\mathbf{v}_r - \mathbf{z}_1}{\mu_r - \lambda_1} & \cdots & \frac{\mathbf{v}_r - \mathbf{z}_{n+1}}{\mu_r - \lambda_{n+1}} \end{bmatrix} \in \mathbb{C}^{r \times (n+1)}, \quad \mathbf{c} = \begin{bmatrix} w_1 \\ \vdots \\ w_{n+1} \end{bmatrix} \in \mathbb{C}^{n+1}.$$

- Here, $\mathbb{L}$ is a **Loewner matrix** (from Charles Loewner) with:

  left (row) array $(\mu_j, \mathbf{v}_j)$, $j = 1, \ldots, r$, and **right (column) array** $(\lambda_i, \mathbf{z}_i)$, $i = 1, \ldots, n+1$.

Main property

Let $\mathbb{L}$ be a $p \times k$ Loewner matrix. Then the following holds:

$$\mathbf{p}, \mathbf{k} \geq \deg(\mathbf{g}) \quad \Rightarrow \quad \operatorname{rank} \mathbb{L} = \deg(\mathbf{g}).$$

Consequently, every square Loewner matrix of size $\deg(\mathbf{g})$, is non-singular.

A.C. Antoulas and B.D.O. Anderson, On the scalar rational interpolation problem, IMA Journal of Mathematical Control and Information, 3: 61–88, 1986.

A toy example

- Let $\mathbf{f}(s) = (s^2 + 4)/(s + 1)$ be a rational function of complexity $n := \deg(\mathbf{f}) = 2$.
- By evaluating $(\mathbf{s})$ on $\lambda = [1, 3, 5]$ and $\mu = [2, 4, 6, 8]$, one obtains $\mathbf{z} = [5/2, 13/4, 29/6]$ and $\mathbf{v} = [8/3, 4, 40/7, 68/9]$.

A toy example

- Let $\mathbf{f}(s) = (s^2+4)/(s+1)$ be a rational function of complexity $n := \deg(\mathbf{f}) = 2$.
- By evaluating $(\mathbf{s})$ on $\lambda = [1,3,5]$ and $\mu = [2,4,6,8]$, one obtains $\mathbf{z} = [5/2, 13/4, 29/6]$ and $\mathbf{v} = [8/3, 4, 40/7, 68/9]$.
- Then, we construct the Loewner matrix, its null space (rank$(\mathbb{L}) = 2$), and a rational function interpolating the data as,

$$\mathbb{L} = \begin{bmatrix} \frac{1}{6} & \frac{7}{12} & \frac{13}{18} \\ \frac{1}{2} & \frac{3}{4} & \frac{5}{6} \\ \frac{9}{14} & \frac{23}{28} & \frac{37}{42} \\ \frac{13}{18} & \frac{31}{36} & \frac{49}{54} \end{bmatrix}, \mathbf{c} = \begin{bmatrix} \frac{1}{3} \\ -\frac{4}{3} \\ 1 \end{bmatrix}, \mathbf{g}(s) = \frac{\frac{5}{6\,(s-1)} - \frac{13}{3\,(s-3)} + \frac{29}{6\,(s-5)}}{\frac{1}{3\,(s-1)} - \frac{4}{3\,(s-3)} + \frac{1}{s-5}}.$$

- In this case, $\mathbf{g}(s)$ perfectly recovers the original function $\mathbf{f}(s)$, i.e., $\mathbf{g}(s) = \mathbf{f}(s)$.

A toy example

- Let $\mathbf{f}(s) = (s^2+4)/(s+1)$ be a rational function of complexity $n := \deg(\mathbf{f}) = 2$.
- By evaluating $(\mathbf{s})$ on $\lambda = [1,3,5]$ and $\mu = [2,4,6,8]$, one obtains $\mathbf{z} = [5/2, 13/4, 29/6]$ and $\mathbf{v} = [8/3, 4, 40/7, 68/9]$.
- Then, we construct the Loewner matrix, its null space (rank$(\mathbb{L}) = 2$), and a rational function interpolating the data as,

$$\mathbb{L} = \begin{bmatrix} \frac{1}{6} & \frac{7}{12} & \frac{13}{18} \\ \frac{1}{2} & \frac{3}{4} & \frac{5}{6} \\ \frac{9}{14} & \frac{23}{28} & \frac{37}{42} \\ \frac{13}{18} & \frac{31}{36} & \frac{49}{54} \end{bmatrix}, \mathbf{c} = \begin{bmatrix} \frac{1}{3} \\ -\frac{4}{3} \\ 1 \end{bmatrix}, \mathbf{g}(s) = \frac{\frac{5}{6\,(s-1)} - \frac{13}{3\,(s-3)} + \frac{29}{6\,(s-5)}}{\frac{1}{3\,(s-1)} - \frac{4}{3\,(s-3)} + \frac{1}{s-5}}.$$

- In this case, $\mathbf{g}(s)$ perfectly recovers the original function $\mathbf{f}(s)$, i.e., $\mathbf{g}(s) = \mathbf{f}(s)$.
- A matrix-format realization can be obtained as $\hat{\mathbf{H}}(s) = \mathbf{W}\boldsymbol{\Phi}(s)^{-1}\mathbf{G}$, where

$$\boldsymbol{\Phi}(s) = \left[\begin{array}{ccc} s-1 & 3-s & 0 \\ s-1 & 0 & 5-s \\ \hline -\frac{1}{3} & \frac{4}{3} & -1 \end{array}\right] \text{ and } \begin{cases} \mathbf{W} = \left[\begin{array}{cc|c} 0 & 0 & -1 \end{array}\right], \\ \mathbf{G}^\top = \begin{bmatrix} \frac{5}{6} & -\frac{13}{3} & \frac{29}{6} \end{bmatrix}. \end{cases}$$

The AAA algorithm - a summary

- The **AAA** algorithm was introduced in [Nakatsukasa/Sete/Trefethen '18] .
- It stands for "*Adaptive Antoulas-Anderson*" in honor of the authors who introduced this type of interpolation scheme in the 80s.

• ***A.C. Antoulas*** *and* ***B.D.O. Anderson****,*
On the scalar rational interpolation problem,
IMA Journal of Mathematical Control and Information, 3: 61–88, 1986.

The main steps of the **AAA** algorithm are:

1. Write down rational approximants in a "barycentric" representation.
2. Select the interpolation points ("support points") via a Greedy scheme.
3. Compute the other variables ("weights") to enforce **least squares** approximation.

⇝ The block-**AAA** algorithm was developed in [Gosea/Güttel '21] ;
⇝ The set-valued **AAA** algorithm was proposed in [Lietaert et al. '22] ;
⇝ The **AA** approach was extended to the DAE case (of index 2) [Gosea/H. '24] ;
⇝ Ongoing work for extending **AAA** to generic DAE cases (index-aware approach) [Pradovera/Gosea/H. '24] – upcoming ;

Loewner and AA With Implicitly Defined Polynomial Parts

Why matching polynomial terms?

- We saw that for the toy example:

$$\mathbf{f}(s) = (s^2 + 4)/(s + 1)$$

the standard AA method works well (completely recovers the function).

Loewner and AA With Implicitly Defined Polynomial Parts

Why matching polynomial terms?

- We saw that for the toy example:
$$\mathbf{f}(s) = (s^2 + 4)/(s + 1)$$
the standard AA method works well (completely recovers the function).
- Also, this function has a polynomial part of degree $n = 1$ (meaning that $f(s) = \mathcal{O}_\infty(s)$)

Loewner and AA With Implicitly Defined Polynomial Parts

Why matching polynomial terms?

- We saw that for the toy example:

$$\mathbf{f}(s) = (s^2 + 4)/(s + 1)$$

  the standard AA method works well (completely recovers the function).
- Also, this function has a polynomial part of degree $n = 1$ (meaning that $f(s) = \mathcal{O}_\infty(s)$)
- Nonetheless, in practical (more complex) examples, the classical methods fail to accurately reproduce the behavior at high frequencies:

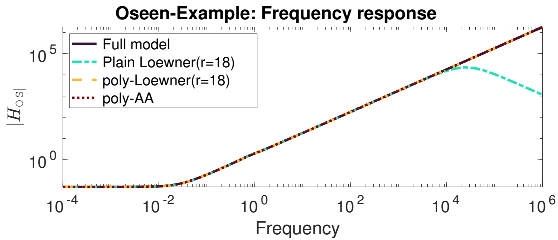


Figure: A typical frequency response plot for systems with polynomial parts...

- In [Gosea/H. '24] the classical barycentric form is modified to account for the case of higher-index DAEs, i.e., with index $\nu = 2$, as simple as:

$$\boxed{\tilde{\mathbf{g}}(s) = \frac{q + \displaystyle\sum_{i=1}^{n+1} \frac{w_i \mathbf{z}_i}{s - \lambda_i}}{\displaystyle\sum_{i=1}^{n+1} \frac{w_i}{s - \lambda_i}}}$$

- The free parameters (weights) w_i + the coefficient q can be also found as before, i.e., so that additional interpolation conditions hold:

$$\tilde{\mathbf{g}}(\mu_j) = \mathbf{v}_j,\ j = 1, \cdots, r.$$

- In [Gosea/H. '24] the classical barycentric form is modified to account for the case of higher-index DAEs, i.e., with index $\nu = 2$, as simple as:

$$\boxed{\tilde{\mathbf{g}}(s) = \frac{q + \sum_{i=1}^{n+1} \frac{w_i \mathbf{z}_i}{s - \lambda_i}}{\sum_{i=1}^{n+1} \frac{w_i}{s - \lambda_i}}}$$

- The free parameters (weights) w_i + the coefficient q can be also found as before, i.e., so that additional interpolation conditions hold:

$$\tilde{\mathbf{g}}(\mu_j) = \mathbf{v}_j,\ j = 1, \cdots, r.$$

- To do so, we need to solve the following equation:

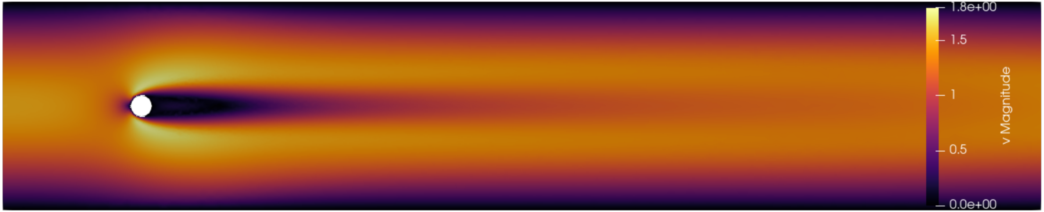
$$\tilde{\mathbb{L}}\tilde{\mathbf{c}} = \mathbf{0},$$

where the augmented Loewner matrix is written as:

$$\tilde{\mathbb{L}} = \begin{bmatrix} \mathbb{L} & -\mathbf{1}_{n+1} \end{bmatrix}, \text{ and } \tilde{\mathbf{c}} = \begin{bmatrix} \mathbf{c} \\ q \end{bmatrix}.$$

Numerical Example

- We consider the flow past a cylinder in 2 dimensions
- at Reynolds number 20 calculated with the averaged inflow velocity and the cylinder diameter as reference quantities;
- see [Gosea/H. ’24] adapted from [Ahmad et al. ’17] .



Snapshot of magnitude of the steady-state NS velocity solution in the considered setup.

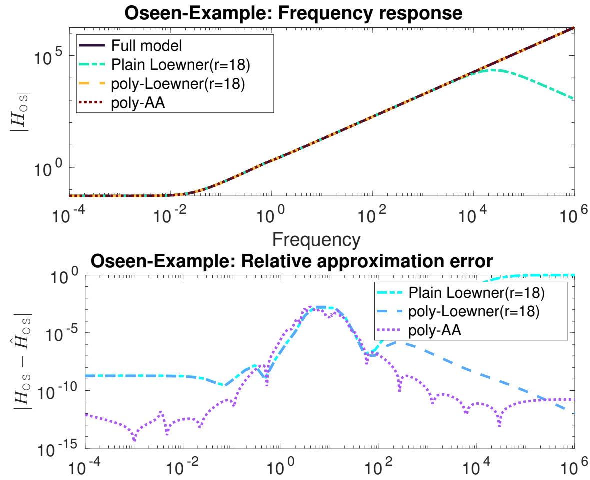
- The considered flow problem with boundary control is modeled by a finite element discretization of the incompressible Oseen equations.
- The Oseen equations are obtained from the Navier-Stokes equations by a Newton linearization about a steady state solution.
- The control $\nu(t,x)$ distributed over the boundary, is modeled as $\nu(t,x) = g(x)u(t)$ through a function $g\colon \Gamma \to \mathbb{R}^2$ that describes the spatial extension.
- Overall, the spatially-discretized model for the velocity v and pressure p reads

$$\begin{aligned}\begin{bmatrix} M & M_\Gamma \end{bmatrix} \begin{bmatrix} \dot v(t) \\ \dot v_\Gamma(t) \end{bmatrix} &= \begin{bmatrix} A & A_\Gamma \end{bmatrix} \begin{bmatrix} v(t) \\ v_\Gamma(t) \end{bmatrix} + J^T p(t), \\ 0 &= \begin{bmatrix} J & J_\Gamma \end{bmatrix} \begin{bmatrix} v(t) \\ v_\Gamma(t) \end{bmatrix}, \quad 0 = v_\Gamma(t) - b_\Gamma u(t), \\ y(t) &= C_v v(t) + C_p p(t). \end{aligned} \tag{2}$$

- The transfer function when considering the y_p output only, with $\mathcal{C}_p = \begin{bmatrix} 0 & C_p \end{bmatrix}$, is:

$$H_{\mathrm{OS}}(s) := \mathcal{C}_p (s\mathcal{E} - \mathcal{A})^{-1} (\mathcal{B}_1 + s\mathcal{B}_2). \tag{3}$$

We compare with the classical (plain) Loewner framework (LF) [Mayo/Antoulas '07] , and with the post processing LF method in [Antoulas/Gosea/Heinkenschloss '20] .



Summary and Conclusion

- Proposed a variant of Loewner-based system identification with free parameters in the Antoulas-Anderson algorithm
- that implicitly covers polynomial parts of the transfer function, avoiding the need for high-frequency data points.
- Drawback: Reduced error control on coefficients, leading to larger approximation errors at high frequencies.
- Future work: adaptive algorithms, like the adaptive Antoulas-Anderson approach.
- Ongoing work: extending the approach to higher polynomial terms and automatic detection of the polynomial degree.

Thank you!

Bibliography I

Mahmad I. Ahmad, Peter Benner, Pawan Goyal, and Jan Heiland.
Moment-matching based model reduction for Navier-Stokes type quadratic-bilinear descriptor systems.
Z. Angew. Math. Mech., 97(10):1252–1267, 2017.
URL: http://www2.mpi-magdeburg.mpg.de/preprints/2015/MPIMD15-18.pdf, doi:10.1002/zamm.201500262.

A. C. Antoulas, S. Lefteriu, and A. C. Ionita.
A tutorial introduction to the Loewner framework for model reduction.
In Model Reduction and Approximation, chapter 8, pages 335–376. SIAM, 2017.
doi:10.1137/1.9781611974829.ch8.

M. Berljafa and S. Güttel.
The rkfit algorithm for nonlinear rational approximation.
SIAM Journal on Scientific Computing, 39(5):A2049–A2071, 2017.
doi:10.1137/15M1025426.

J.-P. Berrut and L. N. Trefethen.
Barycentric Lagrange interpolation.
SIAM Rev., 46(3):501–517, 2004.
doi:10.1137/S0036144502417715.

Bibliography II

I. V. Gosea and S. Güttel.
Algorithms for the rational approximation of matrix-valued functions.
SIAM Journal on Scientific Computing, 43(5):A3033–A3054, 2021.

I. V. Gosea and J. Heiland.
Implicit and explicit matching of non-proper transfer functions in the Loewner framework.
Technical Report 2311.05519, arXiv, 2023.
math.DS.
doi:10.48550/arXiv.2311.05519.

B. Gustavsen and A. Semlyen.
Rational approximation of frequency domain responses by vector fitting.
IEEETransPD, 14(3):1052–1061, 1999.
doi:10.1109/61.772353.

P. Lietaert, K. Meerbergen, J. Pérez, and B. Vandereycken.
Automatic rational approximation and linearization of nonlinear eigenvalue problems.
IMA J. Numer. Anal., 42(2):1087–1115, 2022.
doi:10.1093/imanum/draa098.

Bibliography III

A. J. Mayo and A. C. Antoulas.
A framework for the solution of the generalized realization problem.
Linear Algebra Appl., 425(2-3):634–662, 2007.
doi:10.1016/j.laa.2007.03.008.

Y. Nakatsukasa, O. Sete, and L. N. Trefethen.
The AAA algorithm for rational approximation.
SIAM J. Sci. Comput., 40(3):A1494–A1522, 2018.
doi:10.1137/16M1106122.