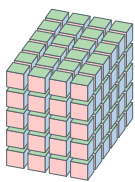
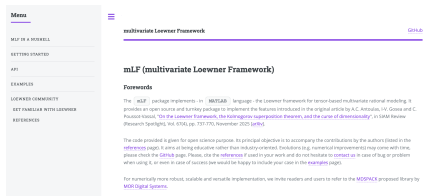
The multivariate Loewner framework

The Kolmogorov superposition theorem, the curse of dimensionality, tensors & more

C. Poussot-Vassal, in coll. with A.C. Antoulas [Rice Univ.], I.V. Gosea [MPI] and P. Vuillemin [ONERA]
November 24, 2026

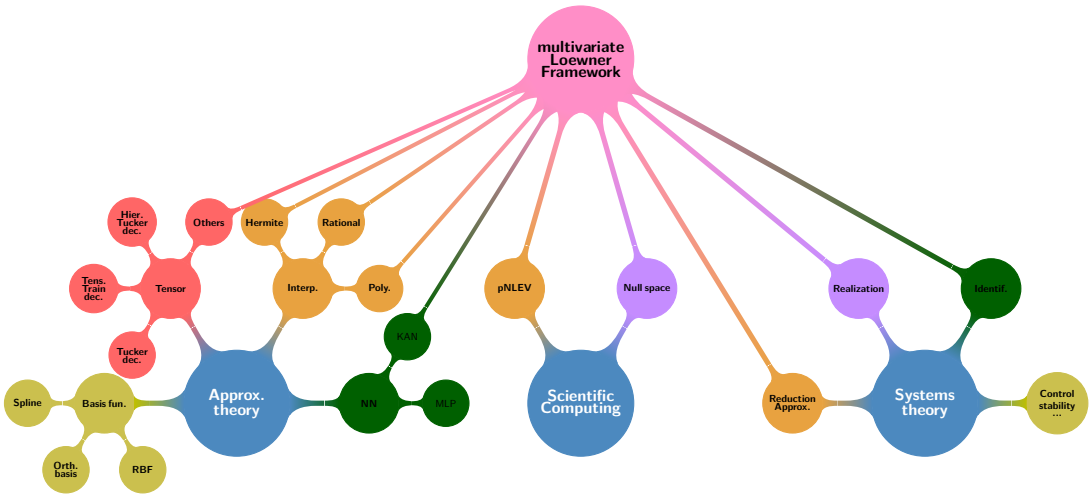
`https://doi.org/10.1137/24M1656657` [in SIAM Review - Research Spotlight]
`https://arxiv.org/abs/2506.04791` [extensive benchmark]
`https://cpoussot.github.io/mlf/` [mLF code & documentation]





Forewords

Approximation, scientific computing & systems theory... where we stand

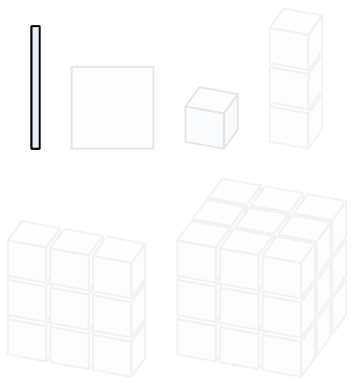


Column / Row data

$$\mathbf{x}_1 = \lambda_1(j_1), \mu_1(i_1) \;\Big\}\xrightarrow{\mathbf{H}(x_1)}\Big\{\; w(j_1), v(i_1)$$

x_1	
$\lambda_1(1, \cdots, k_1)$	$\mathbf{W}^{\otimes}_{k_1}$
$\mu_1(1, \cdots, q_1)$	$\mathbf{V}^{\otimes}_{q_1}$

Tensors (1-D) $\mathcal{T}_1^{\otimes}$

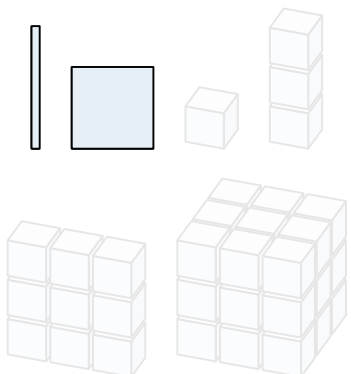


Column / Row data

$$\left.\begin{array}{l}\mathbf{x}_1 = \lambda_1(j_1), \mu_1(i_1) \\ \mathbf{x}_2 = \lambda_2(j_2), \mu_2(i_2)\end{array}\right\} \xrightarrow{\mathbf{H}(x_1,x_2)} \left\{\begin{array}{l} w(j_1, j_2) \\ v(i_1, i_2)\end{array}\right.$$

x_1 \ x_2	$\lambda_2(1, \cdots, k_2)$	$\mu_2(1, \cdots, q_2)$
$\lambda_1(1, \cdots, k_1)$	$\mathbf{W}^{\otimes}_{k_1,k_2}$	ϕ_{cr}
$\mu_1(1, \cdots, q_1)$	ϕ_{rc}	$\mathbf{V}^{\otimes}_{q_1,q_2}$

Tensors (2-D) $\mathcal{T}_2^{\otimes}$



Column / Row data

$$\left.\begin{array}{l}\mathbf{x}_1 = \lambda_1(j_1), \mu_1(i_1)\\ \mathbf{x}_2 = \lambda_2(j_2), \mu_2(i_2)\\ \mathbf{x}_3 = \lambda_3(j_3), \mu_3(i_3)\end{array}\right\} \xrightarrow{\mathbf{H}(x_1,x_2,x_3)} \left\{\begin{array}{l} w(j_1,j_2,j_3)\\ v(i_1,i_2,i_3)\end{array}\right.$$

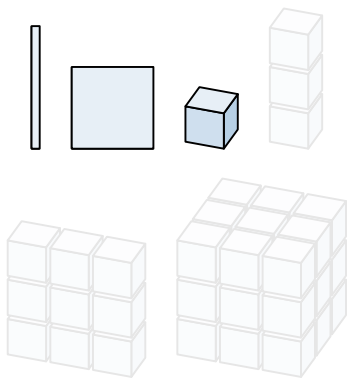
$x_3 = \lambda_3(1, \cdots, k_3)$

x_1 \ x_2	$\lambda_2(1, \cdots, k_2)$	$\mu_2(1, \cdots, q_2)$
$\lambda_1(1, \cdots, k_1)$	$\mathbf{W}^{\otimes}_{k_1,k_2,k_3}$	ϕ_{crc}
$\mu_1(1, \cdots, q_1)$	ϕ_{rcc}	ϕ_{rrc}

$x_3 = \mu_3(1, \cdots, q_3)$

x_1 \ x_2	$\lambda_2(1, \cdots, k_2)$	$\mu_2(1, \cdots, q_2)$
$\lambda_1(1, \cdots, k_1)$	ϕ_{crr}	ϕ_{crr}
$\mu_1(1, \cdots, q_1)$	ϕ_{rcr}	$\mathbf{V}^{\otimes}_{q_1,q_2,q_3}$

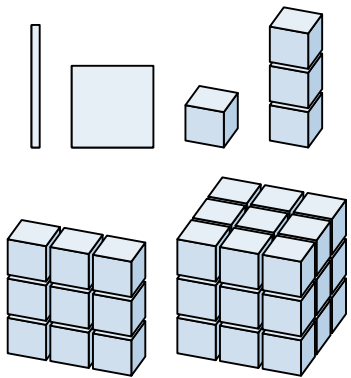
Tensors (3-D) $\mathcal{T}_3^{\otimes}$



Column / Row data

$$\left.\begin{array}{l} \mathbf{x}_1 = \lambda_1(j_1), \mu_1(i_1) \\ \mathbf{x}_2 = \lambda_2(j_2), \mu_2(i_2) \\ \mathbf{x}_3 = \lambda_3(j_3), \mu_3(i_3) \\ \vdots \\ \mathbf{x}_n = \lambda_n(j_n), \mu_n(i_n) \end{array}\right\} \xrightarrow{\mathbf{H}(x_1,\cdots)} \left\{\begin{array}{l} w(j_1,\cdots,j_n) \\ v(i_1,\cdots,i_n) \end{array}\right.$$

Tensors (n-D) $\mathcal{T}_n^{\otimes}$

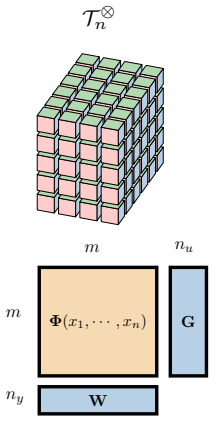


Data-driven model approximation

Being given a n-dimensional tensor $\mathcal{T}_n^{\otimes}$, we seek a multivariate **rational function** $\hat{\mathbf{H}}$ and **realization** $(\mathbf{W}, \mathbf{\Phi}, \mathbf{G})$

$$\hat{\mathbf{H}}(x_1, x_2, \ldots, x_n) = \mathbf{W}\mathbf{\Phi}(x_1, x_2, \ldots, x_n)^{-1}\mathbf{G} \in \mathbb{C}$$

interpolating the data.

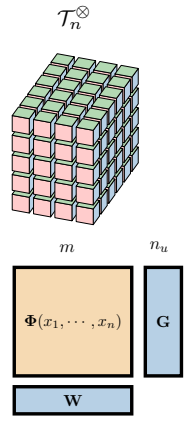


Data-driven model approximation

Being given a n-dimensional tensor $\mathcal{T}_n^{\otimes}$, we seek a multivariate **rational function** $\hat{\mathbf{H}}$ and **realization** $(\mathbf{W}, \mathbf{\Phi}, \mathbf{G})$

$$\hat{\mathbf{H}}(x_1, x_2, \ldots, x_n) = \mathbf{W}\mathbf{\Phi}(x_1, x_2, \ldots, x_n)^{-1}\mathbf{G} \in \mathbb{C}$$

interpolating the data.



Connection to dynamical system realization

A linear-in-state dynamical system $x_1 := s$ parameterized with $\mathcal{S} := [x_2, \ldots, x_n]^\top \subset \mathbb{C}^{n-1}$ leads to

$$\begin{aligned} \mathbf{E}(\mathcal{S})\dot{\mathbf{x}}(t;\mathcal{S}) &= \mathbf{A}(\mathcal{S})\mathbf{x}(t;\mathcal{S}) + \mathbf{B}\mathbf{u}(t) \\ \mathbf{y}(t;\mathcal{S}) &= \mathbf{C}(\mathcal{S})\mathbf{x}(t;\mathcal{S}) \\ &= \\ \hat{\mathbf{H}}(s, x_2, \ldots, x_n) &= \mathbf{C}(\mathcal{S})\left[s\mathbf{E}(\mathcal{S}) - \mathbf{A}(\mathcal{S})\right]^{-1}\mathbf{B} \in \mathbb{C}. \end{aligned}$$

Forewords

Where we stand (some references)

1-D Two-sided Loewner
⇒ (interpolation) barycentric form
⇒ realization minimality
⇒ direct algorithm

1-D One-sided Loewner
⇒ (interpolation) barycentric form
⇒ direct algorithm

1-D AAA (Adaptive Anderson Antoulas - one-sided)
⇒ (mixed interpolation LS) barycentric form
⇒ iterative algorithm

2-D Parametric one-sided Loewner
⇒ (interpolation) barycentric form
⇒ realization (non-minimal)
⇒ direct algorithm

3-D Parametric AAA
⇒ (mixed interpolation LS) barycentric form
⇒ iterative algorithm

>3-D **few results**

J-P. Berrut and N. Trefethen, *"Barycentric Lagrange Interpolation"*, SIAM Review, 46(3), 2004.

A.J. Mayo and A.C. Antoulas, *"A framework for the solution of the generalized realisation problem"*, LAA, 425(2-3), 2007.

A.C. Ionita and A.C. Antoulas, *"Data-Driven Parametrized Model Reduction in the Loewner Framework"*, SIAM Journal on Scientific Computing, 36(3), 2014.

T. Vojkovic, D. Quero, C. P-V and P. Vuillemin, *"Low-Order Parametric State-Space Modeling of MIMO Systems in the Loewner Framework"*, SIAM Journal on Applied Dynamical Systems, 22(4), 2023.

A.C. Rodriguez, L. Balicki and S. Gugercin, *"The p-AAA algorithm for data driven modeling of parametric dynamical systems"*, SIAM Journal on Scientific Computing, 45(3), 2023.

Forewords

Contributions claim & trajectory of the presentation

- n-D tensor data to n-D Loewner matrix $\mathbb{L}_n$
- n-variable transfer functions
- Taming the curse of dimensionality (`flop`/`Bytes`/accuracy)
- n-variable **decoupling**
- **KST** formulation for rational functions
- Comparison with **MLP, KAN, pAAA**
- n-variable generalized realization
- `https://cpoussot.github.io/mlf/`





A.C. Antoulas, I-V. Gosea and C. P-V., *"On the Loewner framework, the Kolmogorov superposition theorem, and the curse of dimensionality"*, SIAM Review, 2025 (`https://arxiv.org/abs/2405.00495`).

C. P-V., I-V. Gosea, P. Vuillemin and A.C. Antoulas, *"Tensor-based multivariate function approximation: methods benchmarking and comparison"*, arXiv:2506.04791, 2025.

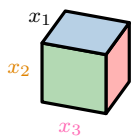
A.C. Antoulas, I-V. Gosea and C. P-V., *"Variable decoupling and the Kolmogorov superposition theorem for rational functions"*, arXiv:2605.07246, 2025.

C. P-V. , I-V. Gosea, P. Vuillemin and A.C. Antoulas, *"mLF package"*, `https://github.com/cpoussot/mLF`, 2026.

Forewords

An overview of the contribution (Ingredient and code)

$$\mathbf{H}(x_1, x_2, x_3) = x_1x_2 + x_1x_3 + x_2x_3$$



- Interpolation points

$$\lambda_1 = \begin{bmatrix} 2 & 4 \end{bmatrix} \in \mathbb{C}^{k_1} \qquad \mu_1 = \begin{bmatrix} 1 & 3 \end{bmatrix} \in \mathbb{C}^{q_1}$$
$$\lambda_2 = \begin{bmatrix} 6 & 8 \end{bmatrix} \in \mathbb{C}^{k_2} \qquad \mu_2 = \begin{bmatrix} 5 & 7 \end{bmatrix} \in \mathbb{C}^{q_2}$$
$$\lambda_3 = \begin{bmatrix} 10 & 12 \end{bmatrix} \in \mathbb{C}^{k_3} \qquad \mu_3 = \begin{bmatrix} 9 & 11 \end{bmatrix} \in \mathbb{C}^{q_3}$$

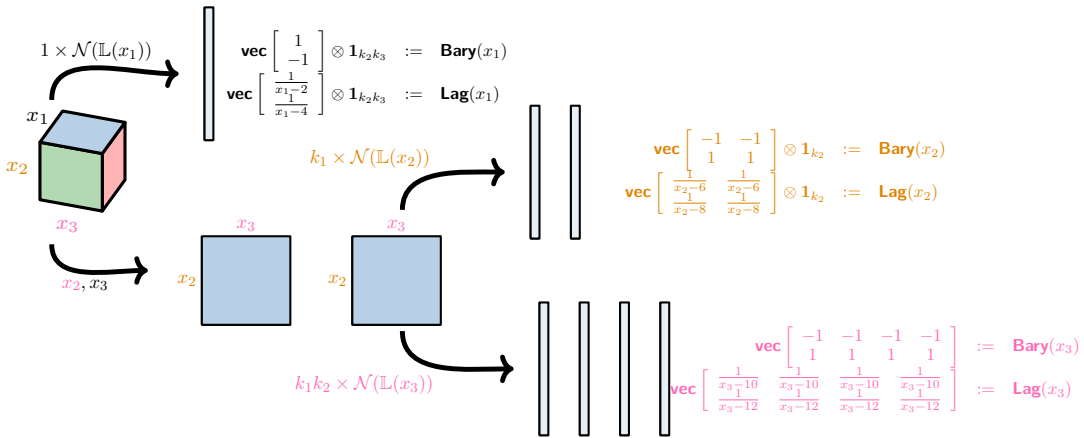
- Function evaluation at $\lambda_{1,2,3}$ leads to $\mathbf{w} =$

$$\begin{bmatrix} 92 & 108 & 116 & 136 & 124 & 144 & 152 & 176 \end{bmatrix}$$

```
% Define model
n = 3;
H = @(x) x(:,1).*x(:,2) + x(:,1).*x(:,3)
+ x(:,2).*x(:,3);
% Define IP
ip1 = [1 2 3 4];
ip2 = [5 6 7 8];
ip3 = [9 10 11 12];
% Split IP and build tensor
for i = 1:n
  pc{i} = ip{i}(2:2:end);
  pr{i} = ip{i}(1:2:end);
end
[y,x,dim] = mlf.make_tab_vec(H,pc,pr);
T = mlf.vec2mat(y,dim);
% mLF
[g,iloe] = mlf.alg1(T,pc,pr);
```

Forewords

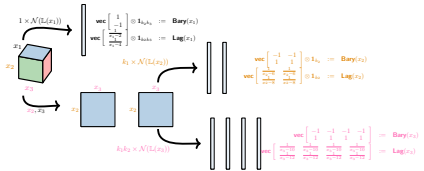
An overview of the contribution (Variables decoupling and Kolmogorov form)



Forewords

An overview of the contribution (Variables decoupling and Kolmogorov form)

$$\mathbf{H}(x_1, x_2, x_3) = x_1x_2 + x_1x_3 + x_2x_3$$

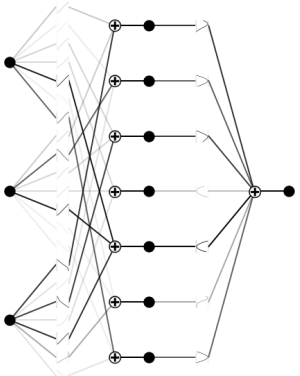


$$\underbrace{\mathbf{\Phi}_1(x_1) := \mathbf{Bary}(x_1)\mathbf{Lag}(x_1)}_{\begin{bmatrix} \frac{1}{x_1-2} \\ \frac{1}{x_1-2} \\ \frac{1}{x_1-2} \\ \frac{1}{x_1-2} \\ -\frac{1}{x_1-4} \\ -\frac{1}{x_1-4} \\ -\frac{1}{x_1-4} \\ -\frac{1}{x_1-4} \end{bmatrix}} \quad \underbrace{\mathbf{\Phi}_2(x_2) := \mathbf{Bary}(x_2)\mathbf{Lag}(x_2)}_{\begin{bmatrix} -\frac{1}{x_2-6} \\ -\frac{1}{x_2-6} \\ \frac{1}{x_2-8} \\ \frac{1}{x_2-8} \\ -\frac{1}{x_2-6} \\ -\frac{1}{x_2-6} \\ \frac{1}{x_2-8} \\ \frac{1}{x_2-8} \end{bmatrix}} \quad \underbrace{\mathbf{\Phi}_3(x_3) := \mathbf{Bary}(x_3)\mathbf{Lag}(x_3)}_{\begin{bmatrix} -\frac{1}{x_3-10} \\ \frac{1}{x_3-12} \\ -\frac{1}{x_3-10} \\ \frac{1}{x_3-12} \\ -\frac{1}{x_3-10} \\ \frac{1}{x_3-12} \\ -\frac{1}{x_3-10} \\ \frac{1}{x_3-12} \end{bmatrix}}$$

$$\mathbf{H}_{\mathrm{kst}}(x_1, x_2, x_3) = \frac{\sum_{\mathsf{rows}} \mathbf{w} \odot \mathbf{\Phi}_1(x_1) \odot \mathbf{\Phi}_2(x_2) \odot \mathbf{\Phi}_3(x_3)}{\sum_{\mathsf{rows}} \mathbf{\Phi}_1(x_1) \odot \mathbf{\Phi}_2(x_2) \odot \mathbf{\Phi}_3(x_3)}$$

$$\mathbf{H}(x_1, x_2, x_3) = \sum_{q=1}^{7} \mathbf{\Phi}_q \left(\sum_{p=1}^{3} \phi_{q,p}(x_p) \right) =$$

$$1.01\, x_1\, x_2 + 0.964\, x_1\, x_3 + 0.964\, x_2\, x_3 +$$
$$0.0149\, {x_1}^2 - 0.0217\, x_1 + 0.016\, {x_2}^2 + 0.0521\, x_2\, {x_3}^2 +$$
$$0.0547\, x_1\, {x_3}^2 - 0.0193\, x_2 - 0.0576\, {x_3}^4 - 0.307\, {x_3}^3 +$$
$$0.0224\, {x_3}^2 + 0.188\, x_3 - 0.00749$$



- $n = 3$, thus $2n + 1 = 7$
- top activations: 7 splines $\mathbf{\Phi}_q$
- bottom activations: 7×3 splines $\phi_{q,p}$

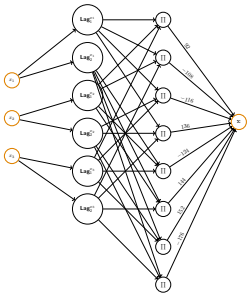
Z. Liu et al., *"KAN: Kolmogorov-Arnold Networks"*, arXiv:2404.19756, 2025.

Forewords

An overview of the contribution (mLF vs. pyKAN)

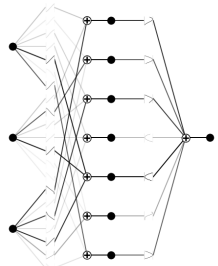
$$\mathbf{H}_{\mathrm{mLF}} = \frac{\sum_{\mathsf{rows}} \mathbf{w} \odot \mathbf{\Phi}_1(x_1) \odot \cdots \odot \mathbf{\Phi}_n(x_n)}{\sum_{\mathsf{rows}} \mathbf{\Phi}_1(x_1) \odot \cdots \odot \mathbf{\Phi}_n(x_n)}$$

$$\mathbf{H}_{\mathrm{pyKAN}} = \sum_{q=1}^{2n+1} \mathbf{\Phi}_q \left(\sum_{p=1}^{n} \phi_{q,p}(x_p) \right)$$



- Activation functions: Lagrangian basis

  $\mathbf{Lag}_j^{x_l} = \dfrac{1}{x_l - x_l(j)}$



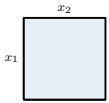
- Activation functions: splines

- **pNLEVP approximation** (Balicki/Gugercin/Embree, 2026)
- **Airbus flutter problem** (dos Reis de Souza et al., 2023)
- **(Irrational) acoustic problem** (Schwoebel et al., 2026)
- **Uncertain identification, realization & LFR** (Vuillemin/P-V., 2026)

Forewords

Some applications (pNEVP approximation)

$$\mathbf{H}(x_1, x_2) = \mathbf{H}(s, p) = \mathbf{1}_n^\top \underbrace{\Big((s - 0.001e^{-ps})I - E\Big)^{-1}}_{\mathbf{\Phi}(s,p)^{-1}} \mathbf{1}_n \quad , n = 10$$

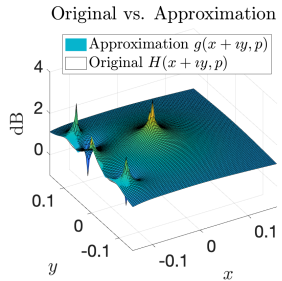


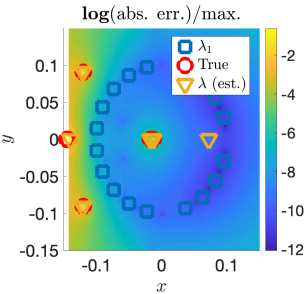
$$\begin{array}{ccc} x_1 & \times & x_2 \\ \frac{1}{10} e^{\imath[0,2\pi]} & \times & [18, 52] \end{array}$$

$$\mathcal{T}_3^{\otimes} \in \mathbb{C}^{40\times 35}$$

Solve $\mathbf{\Phi}(\lambda(p), p)\mathbf{v}(p) = 0$

$p = 18.4336$





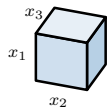
L. Balicki, M. Embree and S. Gugercin, *"A parametric Keldysh decomposition"*, arXiv:2601.01553, 2026.

A.C Antoulas, I-V. Gosea and C. P-V., *"Multivariate descriptor realizations and non linear eigenvalue problems"*, ILAS, 2026.

Forewords

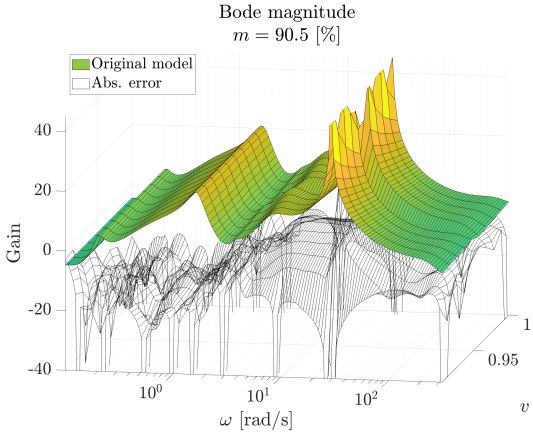
Some applications (Airbus flutter model & data-driven model approximation problem)

$$\mathbf{\Sigma}(x_1, x_2, x_3) = \mathbf{\Sigma}(s, m, v) : s^2 M(m)\mathbf{x}(s) + sB(m)\mathbf{x}(s) + K(m)\mathbf{x}(s) - G(s, v) = \mathbf{u}(s),\ \mathbf{y}(s) = C\mathbf{x}(s)$$



x_1	$\times$	x_2	$\times$	x_3
$\imath[10, 35]$	$\times$	$[\underline{m}, \overline{m}]$	$\times$	$[\underline{v}, \overline{v}]$

$$\mathcal{T}_2^{\otimes} \in \mathbb{C}^{300\times 10\times 10}$$

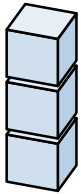


A. dos Reis de Souza et al., *"Aircraft flutter suppression: from a parametric model to robust control"*, ECC, 2023.

Forewords

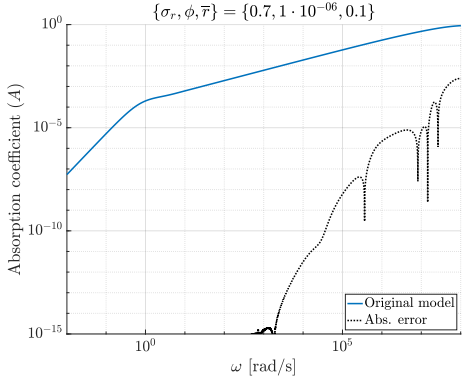
Some applications (Porous material irrational model approximation)

$$Z = -\imath\sqrt{\frac{c_1\alpha(\omega,\sigma_r,\phi,r)}{\beta(\omega,\sigma_r,\phi,r)}}\cot\left(\omega h\sqrt{c_2\alpha(\omega,\sigma_r,\phi,r)\beta(\omega,\sigma_r,\phi,r)}\right) \quad A = 1 - \left|\frac{Z - Z_0}{Z + Z_0}\right|^2$$



$$\begin{array}{ccccccc} x_1 & \times & x_2 & \times & x_3 & \times & x_4 \\ \imath[10,35] & \times & [\underline{\sigma}_r, \overline{\sigma}_r] & \times & [\underline{\phi}, \overline{\phi}] & \times & [\underline{r}, \overline{r}] \end{array}$$

$$\mathcal{T}_4^{\otimes} \in \mathbb{C}^{50\times10\times20\times20}$$



L. Schwoebel et al., *"Application of the multivariate Loewner Framework to the modelling of acoustic propagation in rigid-frame porous materials"*, WAVES, 2026.

Some applications (Uncertain identification, realization & LFR)

$$\mathbf{H}(s,a,b) = \frac{10((2+0.3a+0.4b)s+(1+0.2b))}{s^2+0.5s(a+3)^2+1}$$, modified Zhou/Doyle (1998) example

$$\hat{\mathbf{H}}(s,a,b) = \mathbf{W}(a,b)\mathbf{\Phi}(s,a)^{-1}\mathbf{G}$$



$$\mathbf{\Phi}(s,a,b) = \mathbf{\Phi}(s,a) = \begin{bmatrix} s & -1.0 & 0 \\ 0 & s & -1.0 \\ 0.22 & a\,(0.11\,a+0.67)+1.0 & 0.22 \end{bmatrix}$$

$\mathcal{T}_4^{\otimes} \in \mathbb{C}^{50\times 10\times 20}$

$$\begin{aligned} \mathbf{W}(a,b) &= \begin{bmatrix} 0.44\,b+2.2 & 0.67\,a+0.89\,b+4.4 & 0 \end{bmatrix} \\ \mathbf{G}^{\top} &= \begin{bmatrix} 0 & 0 & 1.0 \end{bmatrix} \end{aligned}$$

K. Zhou and J.C. Doyle, *"Essentials of robust control"*, Prentice Hall, 1998.

P. Vuillemin and C. P-V., *"Uncertain modeling through multivariate Loewner interpolation"*, ECC, 2026.

$$\hat{\mathbf{H}}(s,a,b) = \mathbf{W}\mathbf{\Phi}(s,a,b)^{-1}\mathbf{G}$$

$$\mathbf{\Phi}(s,a,b) = \left[\begin{array}{ccc|ccccc|ccccccc}
s & -1.0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & s & -1.0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.22 & 1.0 & 0.22 & a\,b & a & 0 & 0 & b & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & -1.0\,a & 0 & 0 & 0 & -1.0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0.67 & 0 & -1.0\,b & -1.0 & a\,b & a & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1.0 & 0 & -1.0\,a & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0.11 & 0 & 0 & 0 & -1.0\,b & -1.0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1.0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
2.2 & 4.4 & 0 & 0 & 0 & 0 & 0 & 0 & a\,b & a & 0 & 0 & b & 1.0 \\
0.44 & 0.89 & 0 & 0 & 0 & 0 & 0 & 0 & -1.0\,a & 0 & 0 & 0 & -1.0 & 0 \\
0 & 0.67 & 0 & 0 & 0 & 0 & 0 & 0 & -1.0\,b & -1.0 & a\,b & a & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.0 & 0 & -1.0\,a & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1.0\,b & -1.0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.0 & 0 & 0 & 0
\end{array}\right]$$

$$\mathbf{W} = \left[\begin{array}{cccccccc|cccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1.0 \end{array}\right]$$

$$\mathbf{G}^\top = \left[\begin{array}{ccc|cccccccccccc} 0 & 0 & 1.0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right]$$

$$\hat{\mathbf{H}}(s,a,b) = \mathbf{W}(a,b)\mathbf{\Phi}(s,a)^{-1}\mathbf{G}$$

$$\mathbf{\Phi}(s,a,b) = \mathbf{\Phi}(s,a) = \begin{bmatrix} s & -1.0 & 0 \\ 0 & s & -1.0 \\ 0.22 & a\,(0.11\,a + 0.67) + 1.0 & 0.22 \end{bmatrix} \text{ (independent on } b\text{!)}$$

$$\mathbf{W}(a,b) = \begin{bmatrix} 0.44\,b + 2.2 & 0.67\,a + 0.89\,b + 4.4 & 0 \end{bmatrix}, \mathbf{G}^\top = \begin{bmatrix} 0 & 0 & 1.0 \end{bmatrix}$$

$$\hat{\mathbf{H}}(s,a,b) = \mathbf{W}(s,a)\mathbf{\Phi}(b,s,a)^{-1}\mathbf{G}$$

$$\mathbf{\Phi}(b,s,a) = \begin{bmatrix} b & -1.0 \\ a\,(0.67\,s + 0.11\,s\,a) + s\left(0.11\,a^2 + 0.67\,a + 0.22\,s + 1.0\right) - 1.0\,s\,a\,(0.11\,a + 0.67) + 0.22 & 0 \end{bmatrix}$$

$$\mathbf{W}(s,a) = \begin{bmatrix} 4.4\,s + 0.67\,s\,a + 2.2 & 0.89\,s + 0.44 \end{bmatrix}, \mathbf{G}^\top = \begin{bmatrix} 0 & 1.0 \end{bmatrix}$$

$$\hat{\mathbf{H}}(s,a,b) = \mathbf{W}(a,b)\mathbf{\Phi}(s,a)^{-1}\mathbf{G}$$

$$\mathbf{\Phi}(s,a,b) = \mathbf{\Phi}(s,a) = \begin{bmatrix} s & -1.0 & 0 \\ 0 & s & -1.0 \\ 0.22 & a\,(0.11\,a + 0.67) + 1.0 & 0.22 \end{bmatrix} \text{ (independent on } b\text{!)}$$

$$\mathbf{W}(a,b) = \begin{bmatrix} 0.44\,b + 2.2 & 0.67\,a + 0.89\,b + 4.4 & 0 \end{bmatrix}, \mathbf{G}^\top = \begin{bmatrix} 0 & 0 & 1.0 \end{bmatrix}$$

$$\hat{\mathbf{H}}(s,a,b) = \mathbf{W}(s,a)\mathbf{\Phi}(b,s,a)^{-1}\mathbf{G}$$

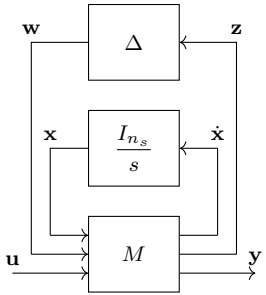
$$\mathbf{\Phi}(b,s,a) = \begin{bmatrix} b & -1.0 \\ a\,(0.67\,s + 0.11\,s\,a) + s\left(0.11\,a^2 + 0.67\,a + 0.22\,s + 1.0\right) - 1.0\,s\,a\,(0.11\,a + 0.67) + 0.22 & 0 \end{bmatrix}$$

$$\mathbf{W}(s,a) = \begin{bmatrix} 4.4\,s + 0.67\,s\,a + 2.2 & 0.89\,s + 0.44 \end{bmatrix}, \mathbf{G}^\top = \begin{bmatrix} 0 & 1.0 \end{bmatrix}$$

$$
\begin{aligned}
E &= \mathbf{diag}([I_{n_1-1}, 0, I_4]) \\
M_{11} &= \left[\begin{array}{ccc|cccc}
0 & 1.0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1.0 & 0 & 0 & 0 & 0 \\
-0.22 & -1.0 & -0.22 & -1.0 & 0 & 0 & 0 \\
\hline
0 & 0.67 & 0 & 0 & 0.11 & 0 & 0 \\
0 & 1.0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1.0 & 0 & 0 & 0 & 0 & 0 \\
-0.44 & -0.89 & 0 & 0 & 0 & 0 & 0
\end{array}\right] \\
M_{12}^\top &= \left[\begin{array}{ccc|cccc} 0 & 0 & 1.0 & 0 & 0 & 0 & 0 \end{array}\right] \\
M_{21} &= \left[\begin{array}{ccc|cccc} 2.2 & 4.4 & 0 & 0 & 0 & 0.67 & -1.0 \end{array}\right] \\
M_{22} &= 0 \\
\mathbf{\Delta} &= \left[\begin{array}{ccc|cccc}
\frac{1}{s} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{1}{s} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \frac{1}{s} & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & a & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & a & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & a & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & b
\end{array}\right]
\end{aligned}
$$

The upper LFR $\mathcal{F}_u(M, \mathbf{\Delta}) :=$

$$M_{21}\mathbf{\Delta}(E - M_{11}\mathbf{\Delta})^{-1}M_{12} + M_{22}$$



Content

Forewords

Multi-variate data, function & Loewner matrix

Taming the curse of dimensionality

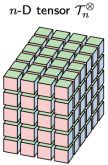
Variables decoupling, KST and KANs

Comparisons

Multi-variate realization

The MATLAB `mLF` package

Conclusion





Multi-variate data, function & Loewner matrix

1-D case

$$\begin{cases} P_c^{(1)} & := & \{(\lambda_1(j_1); w(j_1)),\ j_1 = 1, \ldots, k_1\} \\ P_r^{(1)} & := & \{(\mu_1(i_1); v(i_1)),\ i_1 = 1, \ldots, q_1\} \end{cases}$$

Loewner matrix

$$\mathbb{L}_1 \in \mathbb{C}^{q_1 \times k_1}$$

$$(\mathbb{L}_1)_{i_1, j_1} = \frac{v(i_1) - w(j_1)}{\mu_1(i_1) - \lambda_1(j_1)}$$

$$\mathbf{M}_1 \mathbb{L}_1 - \mathbb{L}_1 \mathbf{\Lambda}_1 = \mathbb{V}_1 \mathbf{R}_1 - \mathbf{L}_1 \mathbb{W}_1$$

Lagrangian form

$$\mathbf{g}(x_1) = \frac{\sum_{j_1=1}^{k_1} \frac{c(j_1) w(j_1)}{x_1 - \lambda_1(j_1)}}{\sum_{j_1=1}^{k_1} \frac{c(j_1)}{x_1 - \lambda_1(j_1)}}$$

Null space

$$\mathbf{span}\ (\mathbf{c}_1) = \mathcal{N}(\mathbb{L}_1)$$

$$\mathbf{c}_1 = \begin{bmatrix} c(1) \\ c(2) \\ \vdots \\ c(k_1) \end{bmatrix} \in \mathbb{C}^{k_1}$$

Multi-variate data, function & Loewner matrix

1-D case (example)

Data generated from $\mathbf{H}(x_1) = \mathbf{H}(s) = (s^2+4)/(s+1)$ of complexity (2)

$$\left.\begin{array}{rcl} \lambda_1(j_1) & = & [1,3,5] \\ \mu_1(i_1) & = & [2,4,6,8] \end{array}\right\} \xrightarrow{\mathbf{H}} \left\{\begin{array}{rcl} \mathbf{w}(j_1) & = & [5/2, 13/4, 29/6] \\ \mathbf{v}(i_1) & = & [8/3, 4, 40/7, 68/9] \end{array}\right.$$

Loewner matrix

$$\mathbb{L}_1 = \begin{bmatrix} \frac{1}{6} & \frac{7}{12} & \frac{13}{18} \\ \frac{1}{2} & \frac{3}{4} & \frac{5}{6} \\ \frac{9}{14} & \frac{23}{28} & \frac{37}{42} \\ \frac{13}{18} & \frac{31}{36} & \frac{49}{54} \end{bmatrix}$$

Null space

$$\mathbf{c}_1 = \begin{bmatrix} \frac{1}{3} \\ -\frac{4}{3} \\ 1 \end{bmatrix}$$

Lagrangian form

$$\mathbf{g}(s) = \frac{\frac{5}{6\,(s-1)} - \frac{13}{3\,(s-3)} + \frac{29}{6\,(s-5)}}{\frac{1}{3\,(s-1)} - \frac{4}{3\,(s-3)} + \frac{1}{s-5}} = \mathbf{H}(s)$$

Multi-variate data, function & Loewner matrix

2-D case

$$\begin{cases} P_c^{(2)} & := & \{(\lambda_1(j_1), \lambda_2(j_2); w(j_1, j_2)),\ j_1 = 1, \ldots, k_1 \quad j_2 = 1, \ldots, k_2\} \\ P_r^{(2)} & := & \{(\mu_1(i_1), \mu_2(i_2); v(i_1, i_2)),\ i_1 = 1, \ldots, q_1 \quad i_2 = 1, \ldots, q_2\} \end{cases}$$

Loewner matrix

$$\mathbb{L}_2 \in \mathbb{C}^{q_1 q_2 \times k_1 k_2}$$

$$\ell_{j_1, j_2}^{i_1, i_2} = \frac{v(i_1, i_2) - w(j_1, j_2)}{(\mu_1(i_1) - \lambda_1(j_1))\,(\mu_2(i_2) - \lambda_2(j_2))}$$

$$\begin{cases} \mathbf{M}_2 \mathbb{X} - \mathbb{X} \mathbf{\Lambda}_2 & = & \mathbb{V}_2 \mathbf{R}_2 - \mathbf{L}_2 \mathbb{W}_2 \\ \mathbf{M}_1 \mathbb{L}_2 - \mathbb{L}_2 \mathbf{\Lambda}_1 & = & \mathbb{X} \end{cases}$$

Lagrangian form

$$\mathbf{g}(x_1, x_2) = \frac{\sum_{j_1=1}^{k_1} \sum_{j_2=1}^{k_2} \frac{c(j_1, j_2) w(j_1, j_2)}{(x_1 - \lambda_1(j_1))(x_2 - \lambda_2(j_2))}}{\sum_{j_1=1}^{k_1} \sum_{j_2=1}^{k_2} \frac{c(j_1, j_2)}{(x_1 - \lambda_1(j_1))(x_2 - \lambda_2(j_2))}}$$

Null space

$$\mathbf{span}\ (\mathbf{c}_2) = \mathcal{N}(\mathbb{L}_2)$$

$$\mathbf{c}_2 = \begin{bmatrix} c(1,1) \\ \vdots \\ c(1,k_2) \\ \hline \vdots \\ \hline c(k_1,1) \\ \vdots \\ c(k_1,k_2) \end{bmatrix} \in \mathbb{C}^{k_1 k_2}$$

Multi-variate data, function & Loewner matrix

2-D case (example)

Data generated from $\mathbf{H}(x_1, x_2) = \mathbf{H}(s,t) = (s^2 t)/(s - t + 1)$ of complexity $(2, 1)$

$$\left.\begin{array}{rcl} \lambda_1(j_1) & = & [1,3,5] \\ \mu_1(i_1) & = & [0,2,4] \\ \lambda_2(j_2) & = & [-1,-3] \\ \mu_2(i_2) & = & [-2,-4] \end{array}\right\} \xrightarrow{\mathbf{H}} \left[\begin{array}{cc|cc} -\frac{1}{3} & -\frac{3}{5} & -\frac{1}{2} & -\frac{2}{3} \\ -\frac{9}{5} & -\frac{27}{7} & -3 & -\frac{9}{2} \\ -\frac{25}{7} & -\frac{25}{3} & -\frac{25}{4} & -10 \\ \hline 0 & 0 & 0 & 0 \\ -1 & -2 & -\frac{8}{5} & -\frac{16}{7} \\ -\frac{8}{3} & -6 & -\frac{32}{7} & -\frac{64}{9} \end{array}\right]$$

Loewner matrix

$$\mathbb{L}_2 = \left[\begin{array}{cc|cc|cc} \frac{1}{3} & -\frac{3}{5} & \frac{3}{5} & -\frac{9}{7} & \frac{5}{7} & -\frac{5}{3} \\ \frac{1}{9} & \frac{3}{5} & \frac{1}{5} & \frac{9}{7} & \frac{5}{21} & \frac{5}{3} \\ \frac{19}{15} & -1 & \frac{1}{5} & -\frac{79}{35} & \frac{23}{35} & -\frac{101}{45} \\ \hline \frac{41}{63} & \frac{59}{35} & -\frac{17}{105} & \frac{11}{7} & \frac{1}{7} & \frac{127}{63} \\ \frac{89}{63} & -\frac{139}{105} & \frac{97}{35} & -\frac{5}{7} & -1 & -\frac{79}{21} \\ \frac{61}{81} & \frac{293}{135} & \frac{239}{135} & \frac{205}{63} & -\frac{223}{189} & \frac{11}{9} \end{array}\right]$$

Null space

$$\mathbf{c}_2 = \left[\begin{array}{c} -\frac{1}{3} \\ \frac{5}{9} \\ \hline \frac{10}{9} \\ -\frac{14}{9} \\ \hline -\frac{7}{9} \\ 1 \end{array}\right]$$

Multi-variate data, function & Loewner matrix

2-D case (example)

Data generated from $\mathbf{H}(x_1, x_2) = \mathbf{H}(s,t) = (s^2 t)/(s - t + 1)$ of complexity $(2,1)$

$$\left.\begin{array}{rcl} \lambda_1(j_1) & = & [1,3,5] \\ \mu_1(i_1) & = & [0,2,4] \\ \lambda_2(j_2) & = & [-1,-3] \\ \mu_2(i_2) & = & [-2,-4] \end{array}\right\} \xrightarrow{\mathbf{H}} \left[\begin{array}{cc|cc} -\frac{1}{3} & -\frac{3}{5} & -\frac{1}{2} & -\frac{2}{3} \\ -\frac{9}{5} & -\frac{27}{7} & -3 & -\frac{9}{2} \\ -\frac{25}{7} & -\frac{25}{3} & -\frac{25}{4} & -10 \\ \hline 0 & 0 & 0 & 0 \\ -1 & -2 & -\frac{8}{5} & -\frac{16}{7} \\ -\frac{8}{3} & -6 & -\frac{32}{7} & -\frac{64}{9} \end{array}\right]$$

Lagrangian form

$$\mathbf{g}(s,t) = -\frac{\frac{1}{9(s-1)(t+1)} - \frac{1}{3(s-1)(t+3)} - \frac{2}{(s-3)(t+1)} + \frac{6}{(s-3)(t+3)} + \frac{25}{9(s-5)(t+1)} - \frac{25}{3(s-5)(t+3)}}{\frac{1}{3(s-1)(t+1)} - \frac{5}{9(s-1)(t+3)} - \frac{10}{9(s-3)(t+1)} + \frac{14}{9(s-3)(t+3)} + \frac{7}{9(s-5)(t+1)} - \frac{1}{(s-5)(t+3)}} = \mathbf{H}(s,t)$$

Multi-variate data, function & Loewner matrix

n-D case

$$\begin{cases} P_c^{(n)} := \{(\lambda_1(j_1), \lambda_2(j_2), \cdots, \lambda_n(j_n); w(j_1, j_2, \cdots, j_n)),\ j_l = 1, \ldots, k_l, \quad l = 1, \cdots, n\} \\ P_r^{(n)} := \{(\mu_1(i_1), \mu_2(i_2), \cdots, \mu_n(i_n); v(i_1, i_2, \cdots, i_n)),\ i_l = 1, \ldots, q_l, \quad l = 1, \cdots, n\} \end{cases}$$

Loewner matrix

$$\mathbb{L}_n \in \mathbb{C}^{q_1 q_2 \cdots q_n \times k_1 k_2 \cdots k_n}$$

$$\ell_{j_1, j_2, \cdots, j_n}^{i_1, i_2, \cdots, i_n} = \frac{v(i_1, i_2, \cdots, i_n) - w(j_1, j_2, \cdots, j_n)}{(\mu_1(i_1) - \lambda_1(j_1)) \cdots (\mu_n(i_n) - \lambda_n(j_n))}$$

$$\begin{cases} \mathbf{M}_n \mathbb{X}_1 - \mathbb{X}_1 \mathbf{\Lambda}_n &= \mathbb{V}_n \mathbf{R}_n - \mathbf{L}_n \mathbb{W}_n, \\ & \vdots \\ \mathbf{M}_1 \mathbb{L}_n - \mathbb{L}_n \mathbf{\Lambda}_1 &= \mathbb{X}_{n-1}. \end{cases}$$

Lagrangian form

$$\mathbf{g}(x_1, \cdots, x_n) = \frac{\sum_{j_1=1}^{k_1} \cdots \sum_{j_n=1}^{k_n} \frac{c(j_1, \cdots, j_n) w(j_1, \cdots, j_n)}{(x_1 - \lambda_1(j_1)) \cdots (x_n - \lambda_n(j_n))}}{\sum_{j_1=1}^{k_1} \cdots \sum_{j_n=1}^{k_n} \frac{c(j_1, \cdots, j_n)}{(x_1 - \lambda_1(j_1)) \cdots (x_n - \lambda_n(j_n))}}$$

Null space

$$\mathbf{span}\ (\mathbf{c}_n) = \mathcal{N}(\mathbb{L}_n)$$

$$\mathbf{c}_n = \begin{bmatrix} c(1, \cdots, 1) \\ \vdots \\ c(1, \cdots, k_n) \\ \hline \vdots \\ \hline c(k_1, \cdots, 1) \\ \vdots \\ c(k_1, \cdots, k_n) \end{bmatrix} \in \mathbb{C}^{k_1 \cdots k_n}$$

Content

Forewords

Multi-variate data, function & Loewner matrix

Taming the curve of dimensionality

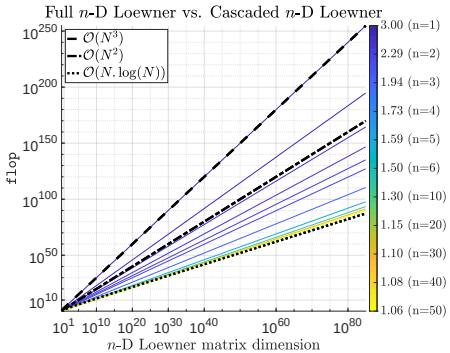
Variables decoupling, KST and KANs

Comparisons

Multi-variate realization

The MATLAB `mLF` package

Conclusion

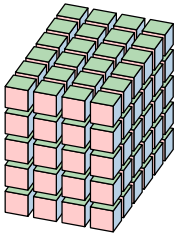


Taming the curse of dimensionality

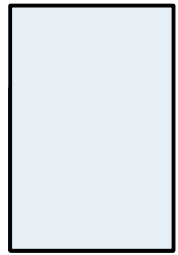
n-variable Loewner matrix operator[1]

$$\begin{array}{ccc} \mathbb{C}^{k_1} \times \mathbb{C}^{q_1} \times \ldots \times \mathbb{C}^{k_n} \times \mathbb{C}^{q_n} \times \mathbb{C}^{(k_1+q_1)\times\cdots\times(k_n+q_n)} & \longrightarrow & \mathbb{C}^{Q\times K} \\ \left(\lambda_1(j_1), \mu_1(i_1), \ldots, \lambda_n(j_n), \mu_n(i_n), \mathcal{T}_n^{\otimes}\right) & \longmapsto & \mathbb{L}_n \end{array}$$

n-D tensor $\mathcal{T}_n^{\otimes}$



matrix $\mathbb{L}_n$

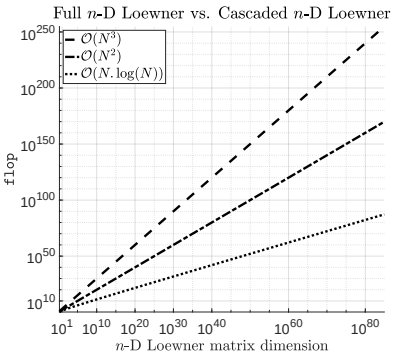


[1] where $Q = q_1 q_2 \cdots q_n$ and $K = k_1 k_2 \cdots k_n$.

Taming the curse of dimensionality

Null space `flop` and memory issues

log-log scale



(rows) $Q = q_1 q_2 \dots q_n$
(columns) $K = k_1 k_2 \dots k_n$

$$\mathbb{L}_n \in \mathbb{C}^{Q \times K}$$

Computational issue

Note that $Q \times K$ matrix SVD `flop` estimation is

- QK^2 (if $Q > K$)
- N^3 (if $Q = K = N$)

Storage issue

Note that $Q \times K$ matrix storage estimation is

- in real double $QK \frac{8}{2^{20}}$ MB
- in complex double $2QK \frac{8}{2^{20}}$ MB

Taming the curse of dimensionality

2-D case (example cont'd, to get the idea)

Data generated from $\mathbf{H}(x_1, x_2) = \mathbf{H}(s, t) = (s^2 t)/(s - t + 1)$ of complexity $(2, 1)$

x_1 \ x_2	$\lambda_2(1) = -1$	$\lambda_2(2) = -3$	$\mu_2(1) = -2$	$\mu_2(2) = -4$
$\lambda_1(1) = 1$	$h_{1,1} = -\frac{1}{3}$	$h_{1,2} = -\frac{3}{5}$	$h_{1,3} = -\frac{1}{2}$	$h_{1,4} = -\frac{2}{3}$
$\lambda_1(2) = 3$	$h_{2,1} = -\frac{9}{5}$	$h_{2,2} = -\frac{27}{7}$	$h_{2,3} = -3$	$h_{2,4} = -\frac{9}{2}$
$\lambda_1(3) = 5$	$h_{3,1} = -\frac{25}{7}$	$h_{3,2} = -\frac{25}{3}$	$h_{3,3} = -\frac{25}{4}$	$h_{3,4} = -10$
$\mu_1(1) = 0$	$h_{4,1} = 0$	$h_{4,2} = 0$	$h_{4,3} = 0$	$h_{4,4} = 0$
$\mu_1(2) = 2$	$h_{5,1} = -1$	$h_{5,2} = -2$	$h_{5,3} = -\frac{8}{5}$	$h_{5,4} = -\frac{16}{7}$
$\mu_1(3) = 4$	$h_{6,1} = -\frac{8}{3}$	$h_{6,2} = -6$	$h_{6,3} = -\frac{32}{7}$	$h_{6,4} = -\frac{64}{9}$

$$\xrightarrow{\mathcal{N}(\mathbb{L}_2)} \mathbf{c}_2 = \begin{bmatrix} -\frac{1}{3} \\ \frac{5}{9} \\ \hline \frac{10}{9} \\ -\frac{14}{9} \\ \hline -\frac{7}{9} \\ 1 \end{bmatrix}$$

- 1 $\mathbb{L}_1$ along x_1, for $x_2 = \lambda_2(2) = -3$
- 3 $\mathbb{L}_1$ along x_2 for $x_1 = \{\lambda_1(1), \lambda_1(2), \lambda_1(3)\}$
- Scaled null space $\mathbf{c}_2^\top =$

$$\left[\begin{array}{cccc} \mathbf{c}_1^{\lambda_1(1)\top} \cdot [\mathbf{c}_1^{\lambda_2(2)}]_1 & \mathbf{c}_1^{\lambda_1(2)\top} \cdot [\mathbf{c}_1^{\lambda_2(2)}]_2 & \mathbf{c}_1^{\lambda_1(3)\top} \cdot [\mathbf{c}_1^{\lambda_2(2)}]_3 & \end{array} \right]^\top$$

Taming the curse of dimensionality

2-D case (example cont'd, to get the idea)

Data generated from $\mathbf{H}(x_1, x_2) = \mathbf{H}(s, t) = (s^2 t)/(s - t + 1)$ of complexity $(2, 1)$

x_1 \ x_2	$\lambda_2(1) = -1$	$\lambda_2(2) = -3$	$\mu_2(1) = -2$	$\mu_2(2) = -4$
$\lambda_1(1) = 1$	$h_{1,1} = -\frac{1}{3}$	$h_{1,2} = -\frac{3}{5}$	$h_{1,3} = -\frac{1}{2}$	$h_{1,4} = -\frac{2}{3}$
$\lambda_1(2) = 3$	$h_{2,1} = -\frac{9}{5}$	$h_{2,2} = -\frac{27}{7}$	$h_{2,3} = -3$	$h_{2,4} = -\frac{9}{2}$
$\lambda_1(3) = 5$	$h_{3,1} = -\frac{25}{7}$	$h_{3,2} = -\frac{25}{3}$	$h_{3,3} = -\frac{25}{4}$	$h_{3,4} = -10$
$\mu_1(1) = 0$	$h_{4,1} = 0$	$h_{4,2} = 0$	$h_{4,3} = 0$	$h_{4,4} = 0$
$\mu_1(2) = 2$	$h_{5,1} = -1$	$h_{5,2} = -2$	$h_{5,3} = -\frac{8}{5}$	$h_{5,4} = -\frac{16}{7}$
$\mu_1(3) = 4$	$h_{6,1} = -\frac{8}{3}$	$h_{6,2} = -6$	$h_{6,3} = -\frac{32}{7}$	$h_{6,4} = -\frac{64}{9}$

$$\xrightarrow{\mathcal{N}(\mathbb{L}_2)} \mathbf{c}_2 = \begin{bmatrix} -\frac{1}{3} \\ \frac{5}{9} \\ \hline \frac{10}{9} \\ -\frac{14}{9} \\ \hline -\frac{7}{9} \\ 1 \end{bmatrix}$$

- 1 $\mathbb{L}_1$ along x_1, for $x_2 = \lambda_2(2) = -3$
- 3 $\mathbb{L}_1$ along x_2 for $x_1 = \{\lambda_1(1), \lambda_1(2), \lambda_1(3)\}$
- Scaled null space $\mathbf{c}_2^\top =$

$$\left[\begin{array}{cc|c} \mathbf{c}_1^{\lambda_1(1)\top} \cdot [\mathbf{c}_1^{\lambda_2(2)}]_1 & \mathbf{c}_1^{\lambda_1(2)\top} \cdot [\mathbf{c}_1^{\lambda_2(2)}]_2 & \mathbf{c}_1^{\lambda_1(3)\top} \cdot [\mathbf{c}_1^{\lambda_2(2)}]_3 \end{array} \right]^\top$$

$$\mathbf{c}_1^{\lambda_2(2)} = \begin{bmatrix} \frac{5}{9} \\ -\frac{14}{9} \\ 1 \end{bmatrix}$$

Taming the curse of dimensionality

2-D case (example cont'd, to get the idea)

Data generated from $\mathbf{H}(x_1, x_2) = \mathbf{H}(s, t) = (s^2 t)/(s - t + 1)$ of complexity $(2, 1)$

x_1 \ x_2	$\lambda_2(1) = -1$	$\lambda_2(2) = -3$	$\mu_2(1) = -2$	$\mu_2(2) = -4$
$\lambda_1(1) = 1$	$h_{1,1} = -\frac{1}{3}$	$h_{1,2} = -\frac{3}{5}$	$h_{1,3} = -\frac{1}{2}$	$h_{1,4} = -\frac{2}{3}$
$\lambda_1(2) = 3$	$h_{2,1} = -\frac{9}{5}$	$h_{2,2} = -\frac{27}{7}$	$h_{2,3} = -3$	$h_{2,4} = -\frac{9}{2}$
$\lambda_1(3) = 5$	$h_{3,1} = -\frac{25}{7}$	$h_{3,2} = -\frac{25}{3}$	$h_{3,3} = -\frac{25}{4}$	$h_{3,4} = -10$
$\mu_1(1) = 0$	$h_{4,1} = 0$	$h_{4,2} = 0$	$h_{4,3} = 0$	$h_{4,4} = 0$
$\mu_1(2) = 2$	$h_{5,1} = -1$	$h_{5,2} = -2$	$h_{5,3} = -\frac{8}{5}$	$h_{5,4} = -\frac{16}{7}$
$\mu_1(3) = 4$	$h_{6,1} = -\frac{8}{3}$	$h_{6,2} = -6$	$h_{6,3} = -\frac{32}{7}$	$h_{6,4} = -\frac{64}{9}$

$$\xrightarrow{\mathcal{N}(\mathbb{L}_2)} \mathbf{c}_2 = \begin{bmatrix} -\frac{1}{3} \\ \frac{5}{9} \\ \hline \frac{10}{9} \\ -\frac{14}{9} \\ \hline -\frac{7}{9} \\ 1 \end{bmatrix}$$

- 1 $\mathbb{L}_1$ along x_1, for $x_2 = \lambda_2(2) = -3$
- 3 $\mathbb{L}_1$ along x_2 for $x_1 = \{\lambda_1(1), \lambda_1(2), \lambda_1(3)\}$
- Scaled null space $\mathbf{c}_2^\top = \begin{bmatrix} \mathbf{c}_1^{\lambda_1(1)\top} \cdot [\mathbf{c}_1^{\lambda_2(2)}]_1 & \mathbf{c}_1^{\lambda_1(2)\top} \cdot [\mathbf{c}_1^{\lambda_2(2)}]_2 & \mathbf{c}_1^{\lambda_1(3)\top} \cdot [\mathbf{c}_1^{\lambda_2(2)}]_3 \end{bmatrix}^\top$

$$\mathbf{c}_1^{\lambda_2(2)} = \begin{bmatrix} \frac{5}{9} \\ -\frac{14}{9} \\ 1 \end{bmatrix}, \mathbf{c}_1^{\lambda_1(1)} = \begin{bmatrix} -\frac{3}{5} \\ 1 \end{bmatrix}, \mathbf{c}_1^{\lambda_1(2)} = \begin{bmatrix} -\frac{5}{7} \\ 1 \end{bmatrix}, \mathbf{c}_1^{\lambda_1(3)} = \begin{bmatrix} -\frac{7}{9} \\ 1 \end{bmatrix}$$

Taming the curse of dimensionality

2-D case (example cont'd, to get the idea)

Data generated from $\mathbf{H}(x_1, x_2) = \mathbf{H}(s,t) = (s^2 t)/(s - t + 1)$ of complexity $(2, 1)$

x_1 \ x_2	$\lambda_2(1) = -1$	$\lambda_2(2) = -3$	$\mu_2(1) = -2$	$\mu_2(2) = -4$
$\lambda_1(1) = 1$	$h_{1,1} = -\frac{1}{3}$	$h_{1,2} = -\frac{3}{5}$	$h_{1,3} = -\frac{1}{2}$	$h_{1,4} = -\frac{2}{3}$
$\lambda_1(2) = 3$	$h_{2,1} = -\frac{9}{5}$	$h_{2,2} = -\frac{27}{7}$	$h_{2,3} = -3$	$h_{2,4} = -\frac{9}{2}$
$\lambda_1(3) = 5$	$h_{3,1} = -\frac{25}{7}$	$h_{3,2} = -\frac{25}{3}$	$h_{3,3} = -\frac{25}{4}$	$h_{3,4} = -10$
$\mu_1(1) = 0$	$h_{4,1} = 0$	$h_{4,2} = 0$	$h_{4,3} = 0$	$h_{4,4} = 0$
$\mu_1(2) = 2$	$h_{5,1} = -1$	$h_{5,2} = -2$	$h_{5,3} = -\frac{8}{5}$	$h_{5,4} = -\frac{16}{7}$
$\mu_1(3) = 4$	$h_{6,1} = -\frac{8}{3}$	$h_{6,2} = -6$	$h_{6,3} = -\frac{32}{7}$	$h_{6,4} = -\frac{64}{9}$

$$\xrightarrow{\mathcal{N}(\mathbb{L}_2)} \mathbf{c}_2 = \begin{bmatrix} -\frac{1}{3} \\ \frac{5}{9} \\ \hline \frac{10}{9} \\ -\frac{14}{9} \\ \hline -\frac{7}{9} \\ 1 \end{bmatrix}$$

- 1 $\mathbb{L}_1$ along x_1, for $x_2 = \lambda_2(2) = -3$
- 3 $\mathbb{L}_1$ along x_2 for $x_1 = \{\lambda_1(1), \lambda_1(2), \lambda_1(3)\}$
- Scaled null space $\mathbf{c}_2^\top =$

$$\left[\begin{array}{ccc} {\mathbf{c}_1^{\lambda_1(1)}}^\top \cdot [\mathbf{c}_1^{\lambda_2(2)}]_1 & {\mathbf{c}_1^{\lambda_1(2)}}^\top \cdot [\mathbf{c}_1^{\lambda_2(2)}]_2 & {\mathbf{c}_1^{\lambda_1(3)}}^\top \cdot [\mathbf{c}_1^{\lambda_2(2)}]_3 \end{array} \right]^\top$$

$$\mathbf{c}_1^{\lambda_2(2)} = \begin{bmatrix} \frac{5}{9} \\ -\frac{14}{9} \\ 1 \end{bmatrix}, \mathbf{c}_1^{\lambda_1(1)} = \begin{bmatrix} -\frac{3}{5} \\ 1 \end{bmatrix}, \mathbf{c}_1^{\lambda_1(2)} = \begin{bmatrix} -\frac{5}{7} \\ 1 \end{bmatrix}, \mathbf{c}_1^{\lambda_1(3)} = \begin{bmatrix} -\frac{7}{9} \\ 1 \end{bmatrix}$$

Theorem: 2-D to 1-D

Being given the tableau $\mathcal{T}_2^{\otimes}$ tensor in response of the 2-variables $\mathbf{H}(x_1, x_2)$ function, the null space of the corresponding 2-D Loewner matrix $\mathbb{L}_2$, is spanned by

$$\mathbf{c}_2 = \mathcal{N}(\mathbb{L}_2) = \mathbf{vec}\left[\mathbf{c}_1^{\lambda_2(1)} \cdot \left[\mathbf{c}_1^{\lambda_1(k_1)}\right]_1, \cdots, \mathbf{c}_1^{\lambda_2(k_2)} \cdot \left[\mathbf{c}_1^{\lambda_1(k_1)}\right]_{k_2}\right],$$

where

- $\mathbf{c}_1^{\lambda_1(k_1)} = \mathcal{N}(\mathbb{L}_1^{\lambda_1(k_1)})$,
  i.e. the null space of the **1-D Loewner matrix** for frozen $x_1 = \lambda_1(k_1)$, and
- $\mathbf{c}_1^{\lambda_2(j_2)} = \mathcal{N}(\mathbb{L}_1^{\lambda_2(j_2)})$,
  i.e. the j_2-th null space of the **1-D Loewner matrices** for frozen $x_2 = \{\lambda_2(1), \cdots, \lambda_2(k_2)\}$.

Theorem: n-D to $(n-1)$-D

Being given the tableau $\mathcal{T}_n^{\otimes}$ tensor in response of the n-variables $\mathbf{H}(x_1, \cdots, x_n)$ function, the null space $\mathcal{N}(\mathbb{L}_n)$ of the corresponding n-D Loewner matrix $\mathbb{L}_n$, is spanned by

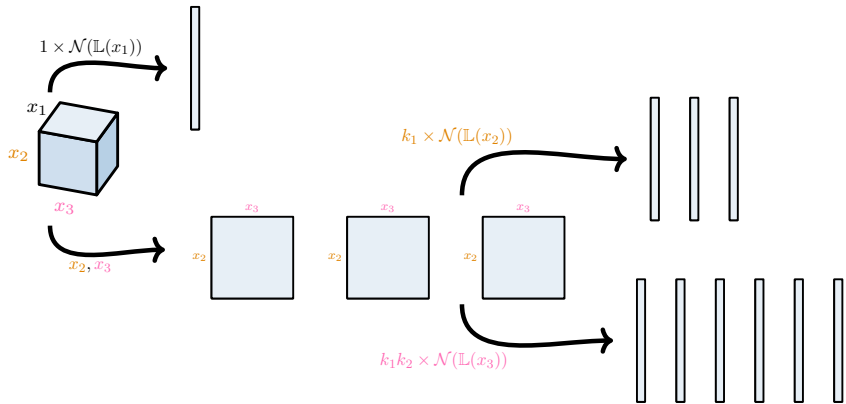
$$\mathbf{c}_n = \mathbf{vec}\left[c_{n-1}^{\lambda_1(1)} \cdot \left[\mathbf{c}_1^{(\lambda_2(k_2), \lambda_3(k_3), \cdots, \lambda_n(k_n))} \right]_1, \cdots, \mathbf{c}_{n-1}^{\lambda_1(k_1)} \cdot \left[\mathbf{c}_1^{(\lambda_2(k_2), \lambda_3(k_3), \cdots, \lambda_n(k_n))} \right]_{k_1} \right],$$

where

- $\mathbf{c}_1^{(\lambda_2(k_2), \lambda_3(k_3), \cdots, \lambda_n(k_n))}$ spans $\mathcal{N}(\mathbb{L}_1^{(\lambda_2(k_2), \lambda_3(k_3), \cdots, \lambda_n(k_n))})$,
  i.e. the null space of the **1-D Loewner matrix** for frozen $\{\lambda_{k_2 2}, \lambda_{k_3 3}, \cdots, \lambda_{k_n n}\}$, and
- $c_{n-1}^{\lambda_1(j_1)}$ spans $\mathcal{N}(\mathbb{L}_{n-1}^{\lambda_1(j_1)})$,
  i.e. the j_1-th null space of the $(n-1)$-**D Loewner matrix** for frozen $x_1 = \{\lambda_1(1), \cdots, \lambda_1(k_1)\}$.

Taming the curse of dimensionality

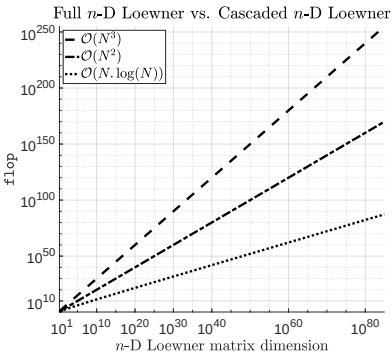
3-D case (illustration with $k_1 = 3, k_2 = 2, k_3$)



Taming the curse of dimensionality

Null space - `flop` complexity

log-log scale



(rows) $Q = q_1 q_2 \dots q_n$
(columns) $K = k_1 k_2 \dots k_n$

$$\mathbb{L}_n \in \mathbb{C}^{Q \times K}$$

Computational issue

Note that $Q \times K$ matrix SVD `flop` estimation is

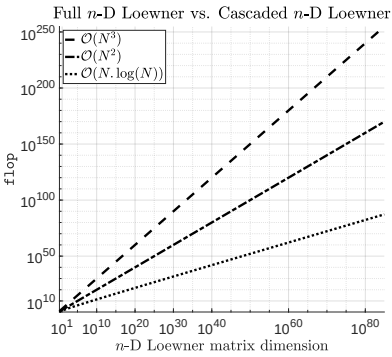
- QK^2 (if $Q > K$)
- N^3 (if $Q = K = N$)

$\Rightarrow$ **The CURSE of dimensionality**

Taming the curse of dimensionality

Null space - `flop` complexity

log-log scale



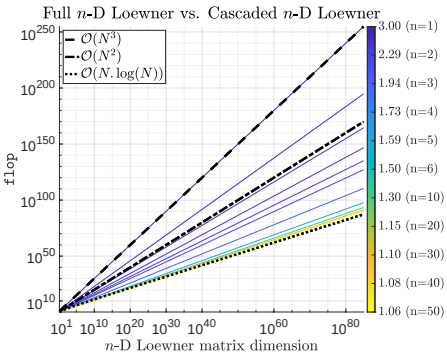
Theorem: complexity

$$\texttt{flop}_1(n) = \sum_{j=1}^{n} \left(k_j^3 \prod_{l=1}^{j} k_{l-1} \right) \text{ where } k_0 = 1.$$

Taming the curse of dimensionality

Null space - `flop` complexity

log-log scale



Corrolary: worst case complexity

k interpolation points per variables.

$$\overline{\texttt{flop}_1(n)} = k^3 \frac{1-k^n}{1-k} = k^3 \frac{1-N}{1-k},$$

which is a (n finite) geometric series of ratio k.

$\Rightarrow$ **The CURSE of dimensionality is TAMED**

$$\begin{aligned} \mathcal{O}(N^3) &\rightarrow \mathcal{O}(N^{2.29}) \quad \text{for } n=2 \\ &\rightarrow \mathcal{O}(N^{1.94}) \quad \text{for } n=3 \\ &\vdots \\ &\rightarrow \mathcal{O}(N^{1.06}) \quad \text{for } n=50 \end{aligned}$$

Taming the curve of dimensionality

Null space - memory and storage

The data (tableau) storage is (in complex and double precision)

$$\frac{8}{2^{20}} \prod_{l}^{n} q_l + k_l \quad \texttt{MB} \text{ (example tableau } 2 \cdot [20, 6, 4, 6, 8, 2] = 2 \cdot [k_1, k_2, k_3, k_4, k_5, k_6] \text{ needs } 45 \texttt{ MB})$$

Full n-D Loewner

Construction of

$$\mathbb{L}_n \in \mathbb{C}^{N \times N}$$

where $N = k_1 k_2 \cdots k_n$, needs

$$\frac{8}{2^{20}} N^2 \quad \texttt{MB}$$

Example: $N = 46,080$
Memory: 31.64 GB
flop: $9.78 \cdot 10^{13}$

Taming the curve of dimensionality

Null space - memory and storage

The data (tableau) storage is (in complex and double precision)

$$\frac{8}{2^{20}} \prod_{l}^{n} q_l + k_l \quad \texttt{MB} \text{ (example tableau } 2 \cdot [20, 6, 4, 6, 8, 2] = 2 \cdot [k_1, k_2, k_3, k_4, k_5, k_6] \text{ needs 45 MB)}$$

Full n-D Loewner

Construction of

$$\mathbb{L}_n \in \mathbb{C}^{N \times N}$$

where $N = k_1 k_2 \cdots k_n$, needs

$$\frac{8}{2^{20}} N^2 \quad \texttt{MB}$$

Example: $N = 46,080$
Memory: 31.64 GB
flop: $9.78 \cdot 10^{13}$

Cascaded n-D Loewner

Construction of

$$\mathbb{L}_1 \in \mathbb{C}^{\overline{k} \times \overline{k}}$$

where $\overline{k} = \max_j k_j$, needs

$$\frac{8}{2^{20}} \overline{k}^2 \quad \texttt{MB}$$

Example: $\overline{k} = 20$
Memory: 6.25 KB
flop: $8.13 \cdot 10^{5}$

Content

Forewords

Multi-variate data, function & Loewner matrix

Taming the curse of dimensionality

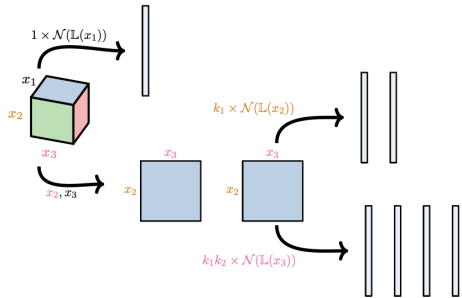
Variables decoupling, KST and KANs

Comparisons

Multi-variate realization

The MATLAB mLF package

Conclusion



Variables decoupling, KST and KANs

Kolmogorov Superposition Theorem and Hilbert's 13th problem

Kolmogorov, Arnold, Kahane, Lorentz and Sprecher

For every (continuous) function $\mathbf{f} : [0,1]^n \mapsto \mathbb{R}$ and any $n \in \mathbb{N}$, $n \geq 2$, there exist

- real numbers $\eta, \lambda_1, \ldots, \lambda_n$;
- (inner) continuous functions $\phi_{q,k} : [0,1]^n \mapsto \mathbb{R}$ (universal)
- (outer) continuous functions $\mathbf{\Phi}_q : \mathbb{R} \mapsto \mathbb{R}$;

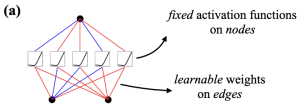
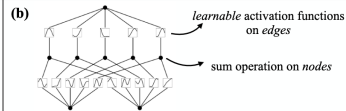
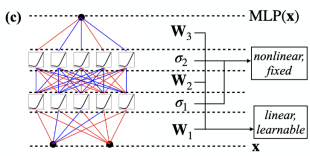
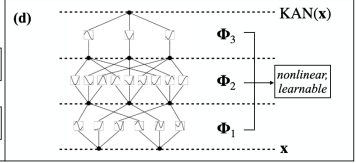
$$\begin{aligned} \mathbf{f}(x_1, \cdots, x_n) &= \sum_{q=1}^{2n+1} \mathbf{\Phi}_q \left(\sum_{p=1}^{n} \phi_{q,p}(x_p) \right) \\ &= \sum_{q=1}^{2n+1} \mathbf{\Phi} \left(\sum_{p=1}^{n} \lambda_p \phi(x_p + \eta q) + q \right) \end{aligned}$$ / Lorentz (1962), Sprecher (1965)

"Kolmogorov proved that every continuous function of several variables can be represented as a superposition of continuous functions of one variable and the operation of addition (1957). Thus, it is as if there are no functions of several variables at all. Seriously speaking, Kolmogorov's theorem is a brilliant example of his mastery. In particular, the theorem shows that Hilbert's conjecture (to its 13th problem) is wrong."

A.G. Vitushkin, *"On Hilbert's thirteenth problem and related questions"*, Russian Math. Surveys 59(1), 2004.

Variables decoupling, KST and KANs

MLP and KAN break

Model	Multi-Layer Perceptron (MLP)	Kolmogorov-Arnold Network (KAN)
Theorem	**Universal Approximation Theorem**	**Kolmogorov-Arnold Representation Theorem**
Formula (Shallow)	$f(\mathbf{x}) \approx \sum_{i=1}^{N(\epsilon)} a_i \sigma(\mathbf{w}_i \cdot \mathbf{x} + b_i)$	$f(\mathbf{x}) = \sum_{q=1}^{2n+1} \Phi_q \left(\sum_{p=1}^{n} \phi_{q,p}(x_p) \right)$
Model (Shallow)		
Formula (Deep)	$\mathrm{MLP}(\mathbf{x}) = (\mathbf{W}_3 \circ \sigma_2 \circ \mathbf{W}_2 \circ \sigma_1 \circ \mathbf{W}_1)(\mathbf{x})$	$\mathrm{KAN}(\mathbf{x}) = (\mathbf{\Phi}_3 \circ \mathbf{\Phi}_2 \circ \mathbf{\Phi}_1)(\mathbf{x})$
Model (Deep)		

Z. Liu et al., *"KAN: Kolmogorov–Arnold Networks"*, arXiv:2404.19756, 2025.

Remember that (in 2-D)

$$\mathbf{c}_2 = \mathcal{N}(\mathbb{L}_2) = \mathbf{vec}\left[\mathbf{c}_1^{\lambda_2(1)} \cdot \left[\mathbf{c}_1^{\lambda_1(k_1)}\right]_1, \cdots, \mathbf{c}_1^{\lambda_2(k_2)} \cdot \left[\mathbf{c}_1^{\lambda_1(k_1)}\right]_{k_2}\right],$$

Variable decoupling

Given data $\mathcal{T}_n^{\otimes}$, the latter achieves variables decoupling, and the null space can be equivalently written as:

$$\mathbf{c}_n = \underbrace{\mathbf{c}^{x_n}}_{\mathbf{Bary}(x_n)} \odot \underbrace{(\mathbf{c}^{x_{n-1}} \otimes \mathbf{1}_{k_n})}_{\mathbf{Bary}(x_{n-1})} \odot \underbrace{(\mathbf{c}^{x_{n-2}} \otimes \mathbf{1}_{k_n k_{n-1}})}_{\mathbf{Bary}(x_{n-2})} \odot \cdots \odot \underbrace{(\mathbf{c}^{x_1} \otimes \mathbf{1}_{k_n \ldots k_2})}_{\mathbf{Bary}(x_1)}.$$

where $\mathbf{c}^{x_l}$ denotes the vectorized barycentric coefficients related to the l-th variable.

This is decoupling !

Remember that (in 2-D)

$$\mathbf{c}_2 = \mathcal{N}(\mathbb{L}_2) = \mathbf{vec}\left[\mathbf{c}_1^{\lambda_2(1)} \cdot \left[\mathbf{c}_1^{\lambda_1(k_1)}\right]_1, \cdots, \mathbf{c}_1^{\lambda_2(k_2)} \cdot \left[\mathbf{c}_1^{\lambda_1(k_1)}\right]_{k_2}\right],$$

Variable decoupling

Given data $\mathcal{T}_n^{\otimes}$, the latter achieves variables decoupling, and the null space can be equivalently written as:

$$\mathbf{c}_n = \underbrace{\mathbf{c}^{x_n}}_{\mathbf{Bary}(x_n)} \odot \underbrace{(\mathbf{c}^{x_{n-1}} \otimes \mathbf{1}_{k_n})}_{\mathbf{Bary}(x_{n-1})} \odot \underbrace{(\mathbf{c}^{x_{n-2}} \otimes \mathbf{1}_{k_n k_{n-1}})}_{\mathbf{Bary}(x_{n-2})} \odot \cdots \odot \underbrace{(\mathbf{c}^{x_1} \otimes \mathbf{1}_{k_n \ldots k_2})}_{\mathbf{Bary}(x_1)}.$$

where $\mathbf{c}^{x_l}$ denotes the vectorized barycentric coefficients related to the l-th variable.

This is decoupling !

Variables decoupling, KST and KANs

Decoupling, KST and KANs via Loewner with rational activation functions ($\mathbf{H} = x_1 \cdot x_2$)

$$
\begin{aligned}
\lambda_1(j_1) &= \begin{pmatrix} -1 & 1 \end{pmatrix} \\
\lambda_2(j_2) &= \begin{pmatrix} -1 & 1 \end{pmatrix} \\
\mu_1(i_1) &= \lambda_1(j_1) + \tfrac{1}{2} \\
\mu_2(i_2) &= \lambda_2(j_2) + \tfrac{1}{2}
\end{aligned}
$$

$$
\mathcal{T}_2^{\otimes} = \left(\begin{array}{cc|cc}
1 & -1 & \frac{1}{2} & -\frac{3}{2} \\
-1 & 1 & -\frac{1}{2} & \frac{3}{2} \\
\hline
\frac{1}{2} & -\frac{1}{2} & \frac{1}{4} & -\frac{3}{4} \\
-\frac{3}{2} & \frac{3}{2} & -\frac{3}{4} & \frac{9}{4}
\end{array}\right)
$$

$$
\left(\begin{array}{ccc|c}
\mathbf{c} & \mathbf{w} & \mathbf{c} \odot \mathbf{w} & \mathbf{Lag}(x_1, x_2) \\
1.0 & 1.0 & 1.0 & \frac{1}{(x_1+1.0)\,(x_2+1.0)} \\
-1.0 & -1.0 & 1.0 & \frac{1}{(x_1+1.0)\,(x_2-1.0)} \\
-1.0 & -1.0 & 1.0 & \frac{1}{(x_1-1.0)\,(x_2+1.0)} \\
1.0 & 1.0 & 1.0 & \frac{1}{(x_1-1.0)\,(x_2-1.0)}
\end{array}\right)
$$

$$
= \frac{\sum_{j_1=1}^{2} \sum_{j_n=1}^{2} \frac{\mathbf{C}_{2,2}^{\otimes}(j_1,j_2)\mathbf{W}_{2,2}^{\otimes}(j_1,j_2)}{(x_1-\lambda_1(j_1))(x_2-\lambda_2(j_2))}}{\sum_{j_1=1}^{2} \sum_{j_2=1}^{2} \frac{\mathbf{C}_{2,2}^{\otimes}(j_1,j_2)}{(x_1-\lambda_1(j_1))(x_2-\lambda_2(j_2))}}
$$

$\mathbf{C}_{2,2}^{\otimes}$ & $\mathbf{W}_{2,2}^{\otimes}$ are the tensor forms of $\mathbf{c}$ & $\mathbf{w}$.

Variables decoupling, KST and KANs

Decoupling, KST and KANs via Loewner with rational activation functions ($\mathbf{H} = x_1 \cdot x_2$)

$$
\begin{aligned}
\lambda_1(j_1) &= \begin{pmatrix} -1 & 1 \end{pmatrix} \\
\lambda_2(j_2) &= \begin{pmatrix} -1 & 1 \end{pmatrix} \\
\mu_1(i_1) &= \lambda_1(j_1) + \tfrac{1}{2} \\
\mu_2(i_2) &= \lambda_2(j_2) + \tfrac{1}{2}
\end{aligned}
$$

$$
\mathcal{T}_2^{\otimes} = \left(\begin{array}{cc|cc}
1 & -1 & \frac{1}{2} & -\frac{3}{2} \\
-1 & 1 & -\frac{1}{2} & \frac{3}{2} \\
\hline
\frac{1}{2} & -\frac{1}{2} & \frac{1}{4} & -\frac{3}{4} \\
-\frac{3}{2} & \frac{3}{2} & -\frac{3}{4} & \frac{9}{4}
\end{array}\right)
$$

$$
\left(\begin{array}{ccc|c}
\mathbf{c} & \mathbf{w} & \mathbf{c} \odot \mathbf{w} & \mathbf{Lag}(x_1, x_2) \\
\hline
1.0 & 1.0 & 1.0 & \frac{1}{(x_1+1.0)\,(x_2+1.0)} \\
-1.0 & -1.0 & 1.0 & \frac{1}{(x_1+1.0)\,(x_2-1.0)} \\
-1.0 & -1.0 & 1.0 & \frac{1}{(x_1-1.0)\,(x_2+1.0)} \\
1.0 & 1.0 & 1.0 & \frac{1}{(x_1-1.0)\,(x_2-1.0)}
\end{array}\right)
$$

Decoupling theorem

$$
\begin{aligned}
\mathbf{c}^{x_2} &= \mathbf{vec}\begin{pmatrix} -1.0 & -1.0 \\ 1.0 & 1.0 \end{pmatrix} \\
\mathbf{c}^{x_1} &= \begin{pmatrix} -1.0 \\ 1.0 \end{pmatrix}
\end{aligned}
$$

$$
\begin{aligned}
\mathbf{c}_2 &= \mathbf{c}^{x_2} \odot (\mathbf{c}^{x_1} \otimes \mathbf{1}_{k_2}) \\
&= \underbrace{\begin{pmatrix} -1 \\ 1 \\ -1 \\ 1 \end{pmatrix}}_{\mathbf{Bary}(x_2)} \odot \underbrace{\begin{pmatrix} -1 \\ -1 \\ 1 \\ 1 \end{pmatrix}}_{\mathbf{Bary}(x_1)}
\end{aligned}
$$

Variables decoupling, KST and KANs

Decoupling, KST and KANs via Loewner with rational activation functions ($\mathbf{H} = x_1 \cdot x_2$)

$$
\begin{aligned}
\lambda_1(j_1) &= \begin{pmatrix} -1 & 1 \end{pmatrix} \\
\lambda_2(j_2) &= \begin{pmatrix} -1 & 1 \end{pmatrix} \\
\mu_1(i_1) &= \lambda_1(j_1) + \tfrac{1}{2} \\
\mu_2(i_2) &= \lambda_2(j_2) + \tfrac{1}{2}
\end{aligned}
$$

$$
\mathcal{T}_2^{\otimes} = \left(\begin{array}{cc|cc}
1 & -1 & \frac{1}{2} & -\frac{3}{2} \\
-1 & 1 & -\frac{1}{2} & \frac{3}{2} \\
\hline
\frac{1}{2} & -\frac{1}{2} & \frac{1}{4} & -\frac{3}{4} \\
-\frac{3}{2} & \frac{3}{2} & -\frac{3}{4} & \frac{9}{4}
\end{array}\right)
$$

$$
\left(\begin{array}{ccc|c}
\mathbf{c} & \mathbf{w} & \mathbf{c} \odot \mathbf{w} & \mathbf{Lag}(x_1, x_2) \\
1.0 & 1.0 & 1.0 & \frac{1}{(x_1+1.0)\,(x_2+1.0)} \\
-1.0 & -1.0 & 1.0 & \frac{1}{(x_1+1.0)\,(x_2-1.0)} \\
-1.0 & -1.0 & 1.0 & \frac{1}{(x_1-1.0)\,(x_2+1.0)} \\
1.0 & 1.0 & 1.0 & \frac{1}{(x_1-1.0)\,(x_2-1.0)}
\end{array}\right)
$$

Denominator / Numerator

$$
\mathbf{D} = \left(\begin{array}{cc}
\overbrace{\mathbf{Bary}(x_1) \cdot \mathbf{Lag}(x_1)}^{\mathbf{\Phi}_1(x_1)} & \overbrace{\mathbf{Bary}(x_2) \cdot \mathbf{Lag}(x_2)}^{\mathbf{\Phi}_2(x_2)} \\
-\frac{1.0}{x_1+1.0} & -\frac{1.0}{x_2+1.0} \\
-\frac{1.0}{x_1+1.0} & \frac{1}{x_2-1.0} \\
\frac{1}{x_1-1.0} & -\frac{1.0}{x_2+1.0} \\
\frac{1}{x_1-1.0} & \frac{1}{x_2-1.0}
\end{array}\right)
$$

Equivalent denominator and numerator read:

$$
\sum_{i\text{-th row}} \prod_{j\text{-th col}} [\mathbf{D}]_{i,j} \quad \text{and} \quad \sum_{i\text{-th row}} \mathbf{w} \cdot \prod_{j\text{-th col}} [\mathbf{D}]_{i,j}
$$

Variables decoupling, KST and KANs

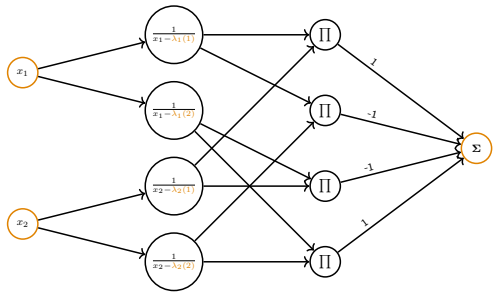
Decoupling, KST and KANs via Loewner with rational activation functions ($\mathbf{H} = x_1 \cdot x_2$)

$$
\begin{aligned}
\lambda_1(j_1) &= \begin{pmatrix} -1 & 1 \end{pmatrix} \\
\lambda_2(j_2) &= \begin{pmatrix} -1 & 1 \end{pmatrix} \\
\mu_1(i_1) &= \lambda_1(j_1) + \tfrac{1}{2} \\
\mu_2(i_2) &= \lambda_2(j_2) + \tfrac{1}{2}
\end{aligned}
$$

$$
\mathcal{T}_2^{\otimes} = \left(\begin{array}{cc|cc}
1 & -1 & \frac{1}{2} & -\frac{3}{2} \\
-1 & 1 & -\frac{1}{2} & \frac{3}{2} \\
\hline
\frac{1}{2} & -\frac{1}{2} & \frac{1}{4} & -\frac{3}{4} \\
-\frac{3}{2} & \frac{3}{2} & -\frac{3}{4} & \frac{9}{4}
\end{array}\right)
$$

$$
\left(\begin{array}{ccc|c}
\mathbf{c} & \mathbf{w} & \mathbf{c} \odot \mathbf{w} & \mathbf{Lag}(x_1, x_2) \\
\hline
1.0 & 1.0 & 1.0 & \frac{1}{(x_1+1.0)\,(x_2+1.0)} \\
-1.0 & -1.0 & 1.0 & \frac{1}{(x_1+1.0)\,(x_2-1.0)} \\
-1.0 & -1.0 & 1.0 & \frac{1}{(x_1-1.0)\,(x_2+1.0)} \\
1.0 & 1.0 & 1.0 & \frac{1}{(x_1-1.0)\,(x_2-1.0)}
\end{array}\right)
$$

Denominator Network view



Variables decoupling, KST and KANs

Decoupling, KST and KANs via Loewner with rational activation functions ($\mathbf{H} = x_1 \cdot x_2$)

KST via Loewner

$$\mathbf{H}_{\text{lag}}(x_1, x_2) = \frac{\sum_{j_1=1}^{2} \sum_{j_2=1}^{2} \dfrac{c(j_1, j_2) w(j_1, j_2)}{(x_1 - \lambda_1(j_1))\,(x_2 - \lambda_2(j_2))}}{\sum_{j_1=1}^{2} \sum_{j_2=1}^{2} \dfrac{c(j_1, j_2)}{(x_1 - \lambda_1(j_1))\,(x_2 - \lambda_2(j_2))}}$$

$$\Updownarrow$$

$$\mathbf{H}_{\text{kst}}(x_1, x_2) = \frac{\sum_{\text{rows}} \mathbf{w} \odot \mathbf{\Phi}_1(x_1) \odot \mathbf{\Phi}_2(x_2)}{\sum_{\text{rows}} \mathbf{\Phi}_1(x_1) \odot \mathbf{\Phi}_2(x_2)} = \frac{\sum_{\text{rows}} \exp\left(\log(\mathbf{w}) + \log\left(\mathbf{\Phi}_1(x_1)\right) + \log\left(\mathbf{\Phi}_2(x_2)\right)\right)}{\sum_{\text{rows}} \exp\left(\log\left(\mathbf{\Phi}_1(x_1)\right) + \log\left(\mathbf{\Phi}_2(x_2)\right)\right)}$$

This is the solution of KST for rational forms !

Content

Forewords

Multi-variate data, function & Loewner matrix

Taming the curse of dimensionality

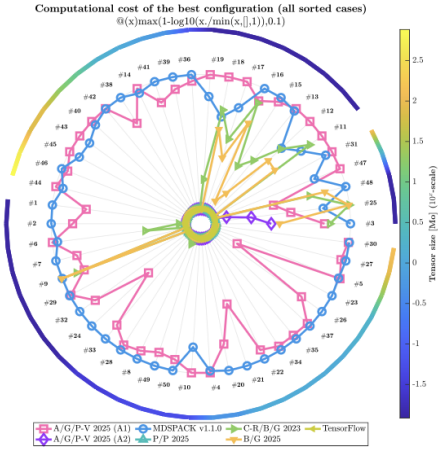
Variables decoupling, KST and KANs

Comparisons

Multi-variate realization

The MATLAB mLF package

Conclusion



Comparisons

Some competitors

Rat. app [B/G 2025]

- Lagrangian interpolation theorem
- p-AAA

KAN [P/P 2025]

- Kolmogorov Arnold theorem
- Kolmogorov Arnold Network

MLP [TensorFlow by Google, via Keras 2025]

- Universal approximation theorem
- Dense connected / ReLU / ADAM / 1000 it. / rand. init.

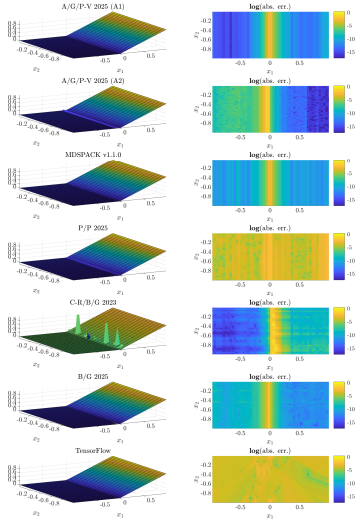
L. Balicki and S. Gugercin, *"Multivariate Rational Approximation via Low-Rank Tensors and the p-AAA Algorithm"*, SIAM Journal on Scientific Computing, 2025.

M. Poluektov and A. Polar, *"Construction of the Kolmogorov-Arnold representation using the Newton-Kaczmarz method"*, arXiv:2305.08194, 2025.

M. Abadi et al., *"TensorFlow: Large-scale machine learning on heterogeneous systems, 2015"*, Software available from `tensorflow.org`.

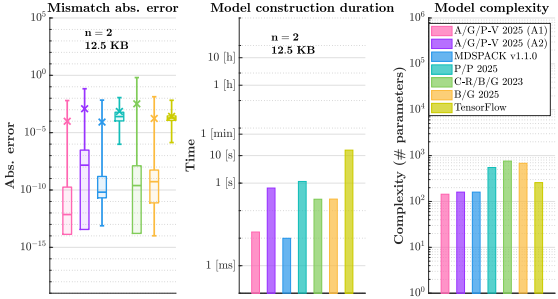
Comparisons

Function #1



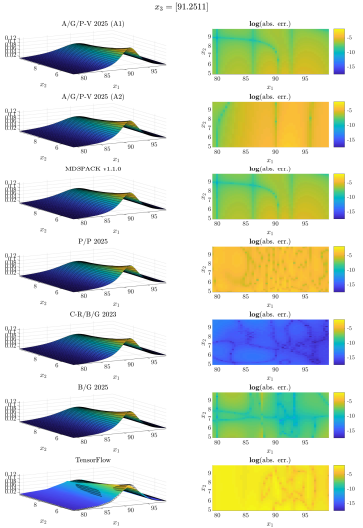
$$\mathrm{ReLU}(x_1) + \tfrac{1}{100} x_2$$

- Reference: Personal communication
- Tensor size: **12.5 KB**



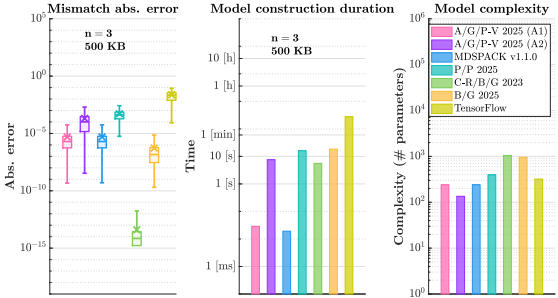
Comparisons

Function #20



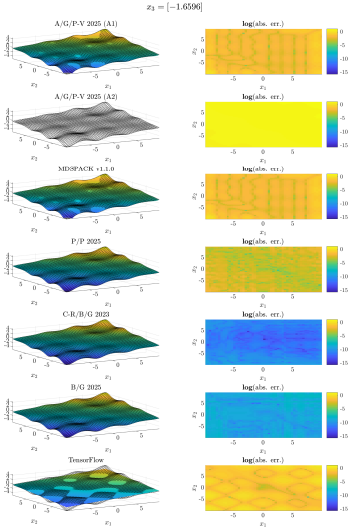
Breit Wigner function

- Reference: A/al. 2021 (A.5.15)
- Tensor size: **500 KB**



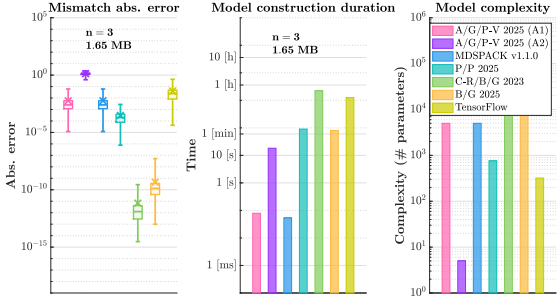
Comparisons

Function #26



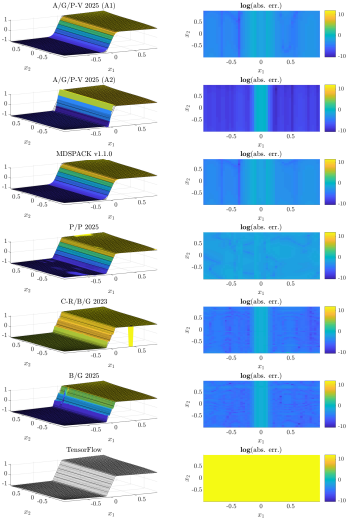
$$\frac{x_1+x_2+x_3}{6+\cos(x_1)+\cos(x_2)+\cos(x_3)}$$

- Reference: B/G 2025
- Tensor size: **1.65 MB**



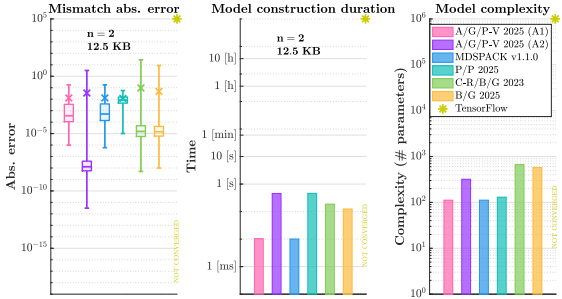
Comparisons

Function #29



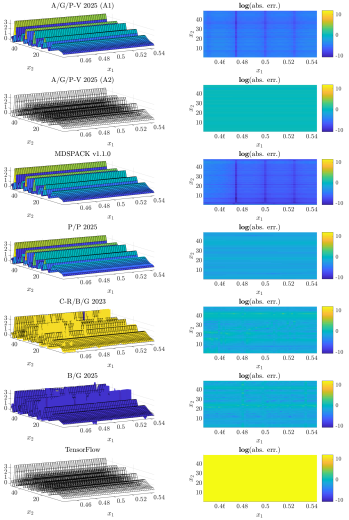
$$\min(10|x_1|, 1)\text{sign}(x_1) + \frac{x_1 x_2^3}{10}$$

- Reference: Personal communication
- Tensor size: **12.5 KB**



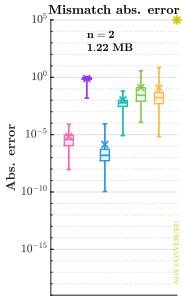
Comparisons

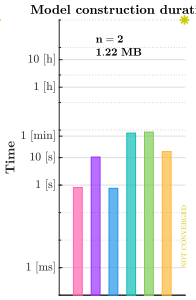
Function #34

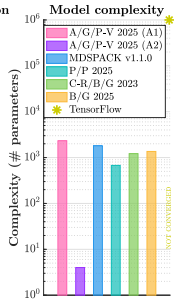


Riemann ζ function (real part)

- Reference: Riemann ζ function (real part)
- Tensor size: **1.22 MB**

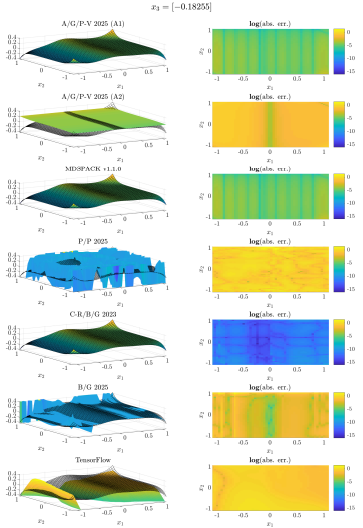






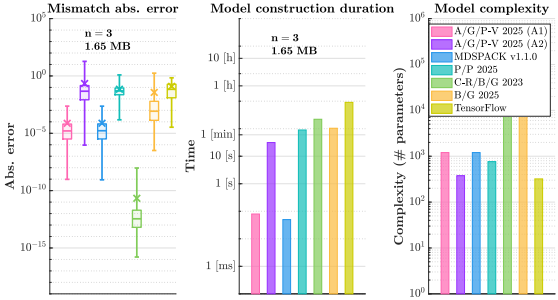
Comparisons

Function #38



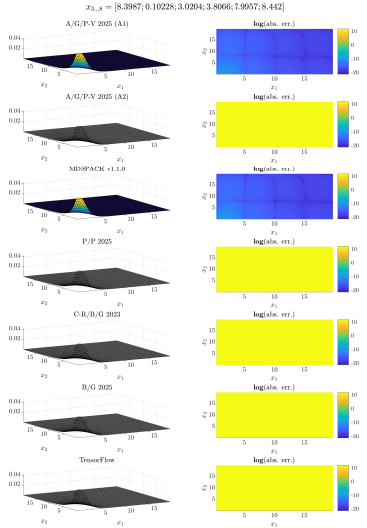
$$\frac{x_1^9 x_2^7 + x_1^3 + 5x_3^2}{5x_1^4 + 4x_1^2 + x_3 x_2^3 + 1}$$

- Reference: A.C. Antoulas presentation
- Tensor size: **1.65 MB**



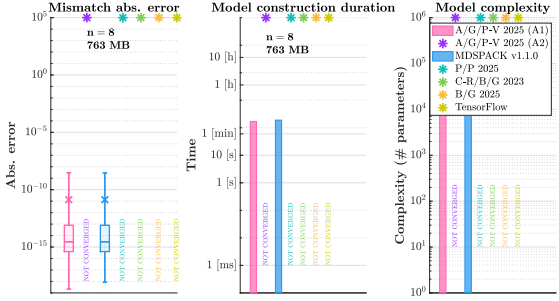
Comparisons

Function #44



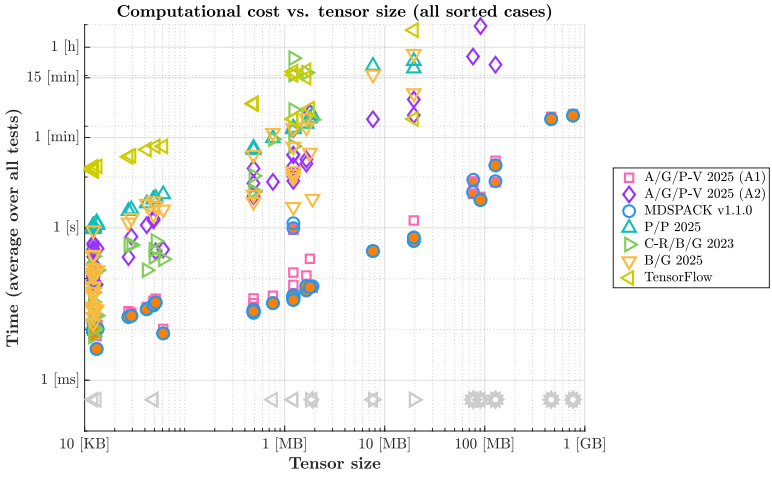
$$\frac{1}{x_1^4+x_2^2x_3+x_4^2+x_5+x_6+x_7+x_8}$$

- Reference: Personal communication
- Tensor size: **763 MB**



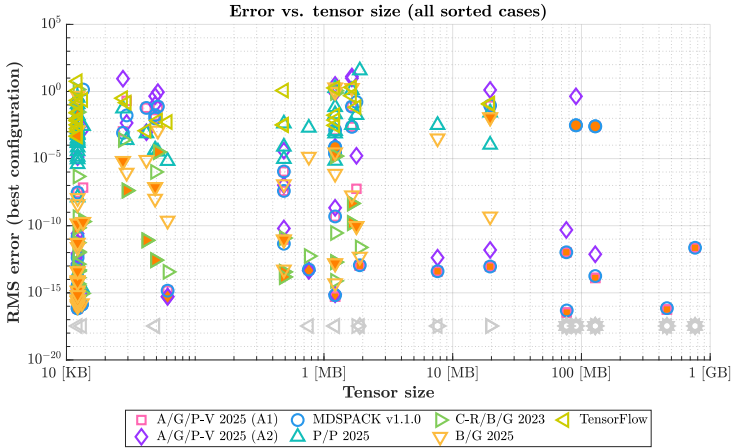
Comparisons

Size vs. computation time



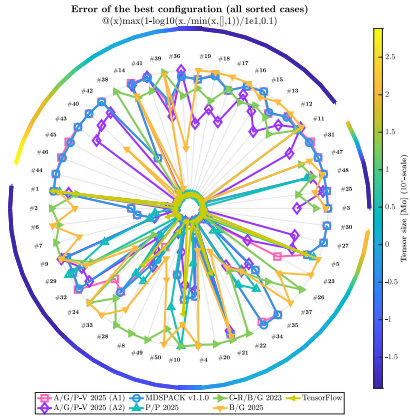
Comparisons

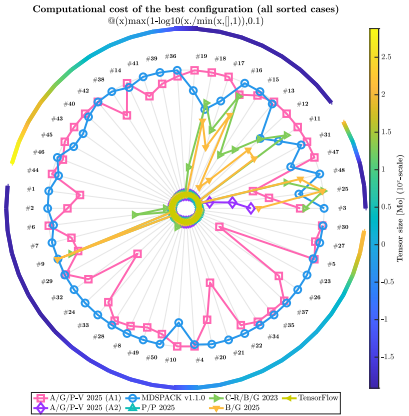
Size vs. error



Comparisons

Radar error & computational time





Content

Forewords

Multi-variate data, function & Loewner matrix

Taming the curse of dimensionality

Variables decoupling, KST and KANs

Comparisons

Multi-variate realization

The MATLAB mLF package

Conclusion

Lagrangian realization $\hat{\mathbf{H}}(s,t) = \mathbf{W}\mathbf{\Phi}(s,t)^{-1}\mathbf{G}$

$$\mathbf{\Phi}(s,t) = \left[\begin{array}{cc:c:cc:cc} s-1 & 3-s & 0 & 0 & 0 & 0 \\ s-1 & 0 & 5-s & 0 & 0 & 0 \\ \hdashline -\frac{1}{3} & \frac{10}{9} & -\frac{7}{9} & t+1 & 0 & 0 \\ \frac{5}{9} & -\frac{14}{9} & 1 & -t-3 & 0 & 0 \\ \hdashline \frac{1}{9} & -2 & \frac{25}{9} & 0 & t+1 & \frac{1}{2} \\ -\frac{1}{3} & 6 & -\frac{25}{3} & 0 & -t-3 & -\frac{1}{2} \end{array}\right]$$

Multi-variate realization

1-D case (example cont'd)

Data generated from $\mathbf{H}(x_1) = \mathbf{H}(s) = (s^2+4)/(s+1)$ of complexity (2)

$$\mathbf{g}(s) = \frac{\frac{5}{6\,(s-1)} - \frac{13}{3\,(s-3)} + \frac{29}{6\,(s-5)}}{\frac{1}{3\,(s-1)} - \frac{4}{3\,(s-3)} + \frac{1}{s-5}}$$

Lagrangian realization $\hat{\mathbf{H}}(s) = \mathbf{W}\mathbf{\Phi}(s)^{-1}\mathbf{G}$

$$\mathbf{\Phi}(s) = \left[\begin{array}{ccc} s-1 & 3-s & 0 \\ s-1 & 0 & 5-s \\ \hline -\frac{1}{3} & \frac{4}{3} & -1 \end{array}\right]$$

$$\mathbf{W} = \left[\begin{array}{ccc} \frac{5}{6} & -\frac{13}{3} & \frac{29}{6} \end{array}\right]$$

$$\mathbf{G}^\top = \left[\begin{array}{cc:c} 0 & 0 & -1 \end{array}\right]$$

Multi-variate realization

2-D case (example cont'd)

Data generated from $\mathbf{H}(x_1, x_2) = \mathbf{H}(s,t) = (s^2 t)/(s-t+1)$ of complexity $(2,1)$

$$\mathbf{g}(s,t) = -\frac{\frac{1}{9(s-1)(t+1)} - \frac{1}{3(s-1)(t+3)} - \frac{2}{(s-3)(t+1)} + \frac{6}{(s-3)(t+3)} + \frac{25}{9(s-5)(t+1)} - \frac{25}{3(s-5)(t+3)}}{\frac{1}{3(s-1)(t+1)} - \frac{5}{9(s-1)(t+3)} - \frac{10}{9(s-3)(t+1)} + \frac{14}{9(s-3)(t+3)} + \frac{7}{9(s-5)(t+1)} - \frac{1}{(s-5)(t+3)}}$$

Lagrangian realization $\hat{\mathbf{H}}(s,t) = \mathbf{W}\mathbf{\Phi}(s,t)^{-1}\mathbf{G}$

$$\mathbf{\Phi}(s,t) = \left[\begin{array}{cc:c:cc:c} s-1 & 3-s & 0 & 0 & 0 & 0 \\ s-1 & 0 & 5-s & 0 & 0 & 0 \\ \hdashline -\frac{1}{3} & \frac{10}{9} & -\frac{7}{9} & t+1 & 0 & 0 \\ \frac{5}{9} & -\frac{14}{9} & 1 & -t-3 & 0 & 0 \\ \hdashline \frac{1}{9} & -2 & \frac{25}{9} & 0 & t+1 & \frac{1}{2} \\ -\frac{1}{3} & 6 & -\frac{25}{3} & 0 & -t-3 & -\frac{1}{2} \end{array}\right]$$

$$\mathbf{W} = \left[\begin{array}{ccccc:c} 0 & 0 & 0 & 0 & 0 & -1 \end{array}\right]$$

$$\mathbf{G}^\top = \left[\begin{array}{cc:cc:cc} 0 & 0 & 1/2 & -1/2 & 0 & 0 \end{array}\right]$$

→ (3,3) block is unimodular !

Multi-variate realization

2-D case (example cont'd)

Data generated from $\mathbf{H}(x_1, x_2) = \mathbf{H}(s,t) = (s^2 t)/(s - t + 1)$ of complexity $(2,1)$

$$\mathbf{g}(s,t) = -\frac{\frac{1}{9(s-1)(t+1)} - \frac{1}{3(s-1)(t+3)} - \frac{2}{(s-3)(t+1)} + \frac{6}{(s-3)(t+3)} + \frac{25}{9(s-5)(t+1)} - \frac{25}{3(s-5)(t+3)}}{\frac{1}{3(s-1)(t+1)} - \frac{5}{9(s-1)(t+3)} - \frac{10}{9(s-3)(t+1)} + \frac{14}{9(s-3)(t+3)} + \frac{7}{9(s-5)(t+1)} - \frac{1}{(s-5)(t+3)}}$$

Lagrangian realization $\hat{\mathbf{H}}_\mathbf{c}(s,t) = \mathbf{W}_\mathbf{c}(t)\mathbf{\Phi}_\mathbf{c}(s,t)^{-1}\mathbf{G}_\mathbf{c}$

$$\mathbf{\Phi}_\mathbf{c}(s,t) = \left[\begin{array}{ccc:c} s-1 & 3-s & 0 & 0 \\ s-1 & 0 & 5-s & 0 \\ -\frac{1}{3} & \frac{10}{9} & -\frac{7}{9} & t+1 \\ \frac{5}{9} & -\frac{14}{9} & 1 & -t-3 \end{array}\right]$$

$$\mathbf{W}_\mathbf{c}(t) = \left[\begin{array}{ccc:c} -\frac{2\,t}{9} & 4\,t & -\frac{50\,t}{9} & 0 \end{array}\right]$$

$$\mathbf{G}_\mathbf{c}^\top = \left[\begin{array}{cc:cc} 0 & 0 & \frac{1}{2} & -\frac{1}{2} \end{array}\right]$$

Multi-variate realization

Generalized n-D Lagrangian realization

$$\mathbf{g}(x_1, x_2, \cdots, x_n) = \frac{\sum_{j_1=1}^{k_1} \sum_{j_2=1}^{k_2} \cdots \sum_{j_n=1}^{k_n} \frac{c(j_1,j_2,\cdots,j_n)\mathbf{w}(j_1,j_2,\cdots,j_n)}{(x_1-\lambda_1(j_1))(x_2-\lambda_2(j_2))\cdots(x_n-\lambda_n(j_n))}}{\sum_{j_1=1}^{k_1} \sum_{j_2=1}^{k_2} \cdots \sum_{j_n=1}^{k_n} \frac{c(j_1,j_2,\cdots,j_n)}{(x_1-\lambda_1(j_1))(x_2-\lambda_2(j_2))\cdots(x_n-\lambda_n(j_n))}},$$

Theorem: n-D Lagrangian realization

A $2\ell + \kappa - 1 = m$-th order realization $(\mathbf{G}, \mathbf{\Phi}, \mathbf{W})$ of the multi-variate function $\hat{\mathbf{H}}$ in barycentric form, satisfying $\hat{\mathbf{H}}(x_1, \cdots, x_n) = \mathbf{W}\mathbf{\Phi}(x_1, x_2, \cdots, x_n)^{-1}\mathbf{G}$, is given by,

$$\mathbf{\Phi}(x_1, \cdots, x_n) = \left[\begin{array}{c|c|c} \mathbf{\Gamma}(1:\kappa-1,:) & \mathbf{0}_{\kappa-1,\ell-1} & \mathbf{0}_{\kappa-1,\ell} \\ \hline \mathbb{A}^{\mathrm{Lag}} & \mathbf{\Delta}(1:\ell-1,:)^\top & \mathbf{0}_{\ell,\ell} \\ \hline \mathbb{B}^{\mathrm{Lag}} & \mathbf{0}_{\ell,\ell-1} & \mathbf{\Delta}^\top \end{array}\right] \in \mathbb{C}^{m\times m}$$

$$\mathbf{G} = \left[\begin{array}{c} \mathbf{0}_{\kappa-1,1} \\ \hline \bar{\mathbf{\Delta}}(\ell,:)^\top \\ \hline \mathbf{0}_{\ell,1} \end{array}\right] \in \mathbb{C}^{m\times 1}$$

$$\mathbf{W} = \left[\begin{array}{c|c|c} \mathbf{0}_{1,\kappa} & \mathbf{0}_{1,\ell-1} & -\mathbf{e}_\ell^\top \end{array}\right] \in \mathbb{C}^{1\times m}$$

where $\mathbb{A}^{\mathrm{Lag}}, \mathbb{B}^{\mathrm{Lag}} \in \mathbb{C}^{\ell\times\kappa}$ are appropriately chosen, according to the chosen pseudo-companion basis.

Multi-variate realization

Generalized n-D Lagrangian realization (focus on left / right variable sets)

$$\mathbf{\Phi}(x_1, \cdots, x_n) = \left[\begin{array}{c|c|c} \mathbf{\Gamma}(1:\kappa-1,:) & \mathbf{0}_{\kappa-1,\ell-1} & \mathbf{0}_{\kappa-1,\ell} \\ \hline \mathbb{A}^{\mathrm{Lag}} & \mathbf{\Delta}(1:\ell-1,:)^{\top} & \mathbf{0}_{\ell,\ell} \\ \hline \mathbb{B}^{\mathrm{Lag}} & \mathbf{0}_{\ell,\ell-1} & \mathbf{\Delta}^{\top} \end{array}\right] \in \mathbb{C}^{m\times m}$$

$$\mathbf{\Gamma} = \mathbb{X}^{\mathrm{Lag}}{}_1 \otimes \mathbb{X}^{\mathrm{Lag}}{}_2 \otimes \cdots \otimes \mathbb{X}^{\mathrm{Lag}}{}_k \in \mathbb{C}^{\kappa\times\kappa}[x_1, \cdots, x_k]$$

$$\mathbf{\Delta} = \mathbb{X}^{\mathrm{Lag}}{}_{k+1} \otimes \mathbb{X}^{\mathrm{Lag}}{}_{k+2} \otimes \cdots \otimes \mathbb{X}^{\mathrm{Lag}}{}_n \in \mathbb{C}^{\ell\times\ell}[x_{k+1}, \cdots, x_n]$$

$$\mathbb{X}^{\mathrm{Lag}}{}_j = \left[\begin{array}{ccccc} \mathbf{x}_{1j} & -\mathbf{x}_{2j} & 0 & \cdots & 0 \\ \mathbf{x}_{1j} & 0 & -\mathbf{x}_{3j} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{x}_{1j} & 0 & \cdots & 0 & -\mathbf{x}_{n_j j} \\ \hline q_{1j} & q_{2j} & \cdots & q_{n_j-1\,j} & q_{n_j j} \end{array}\right] \in \mathbb{C}^{n_j\times n_j}$$

$$\mathbf{x}_{ij} = x_j - \lambda_j(i)$$

Facts

- Left / right variables splitting

  $\mathbf{\Gamma}$ and $\mathbf{\Delta}$

- $\mathbb{X}^{\mathrm{Lag}}{}_j$ is unimodular, i.e.

  $\det(\mathbb{X}^{\mathrm{Lag}}{}_j) = 1$

- ... so are $\mathbf{\Gamma}$ and $\mathbf{\Delta}$

Multi-variate realization

Generalized n-D Lagrangian realization (focus on barycentric weights $\mathbb{A}^{\mathrm{Lag}}$ and $\mathbb{B}^{\mathrm{Lag}}$)

$$\mathbf{\Phi}(x_1, \cdots, x_n) = \left[\begin{array}{c|c|c} \mathbf{\Gamma}(1:\kappa-1,:) & \mathbf{0}_{\kappa-1,\ell-1} & \mathbf{0}_{\kappa-1,\ell} \\ \hline \mathbb{A}^{\mathrm{Lag}} & \mathbf{\Delta}(1:\ell-1,:)^\top & \mathbf{0}_{\ell,\ell} \\ \hline \mathbb{B}^{\mathrm{Lag}} & \mathbf{0}_{\ell,\ell-1} & \mathbf{\Delta}^\top \end{array}\right] \in \mathbb{C}^{m\times m}$$

$$\mathbb{A}^{\mathrm{Lag}} = \begin{bmatrix} \alpha_{1,1} & \alpha_{1,2} & \cdots & \alpha_{1,m+1} \\ \alpha_{2,1} & \alpha_{2,2} & \cdots & \alpha_{2,m+1} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{n+1,1} & \alpha_{n+1,2} & \cdots & \alpha_{n+1,m+1} \end{bmatrix}$$

$$\mathbf{c}_n = \mathsf{vec}\left(\mathbb{A}^{\mathrm{Lag}}\right)$$

Facts

- $\mathbb{A}^{\mathrm{Lag}}$ is simply some rearrangement of $\mathcal{N}(\mathbb{L}_n) = \mathbf{c}_n$
- $\mathbb{B}^{\mathrm{Lag}}$ follows

Multi-variate realization

Generalized n-D Lagrangian realization (control the complexity)

$$\mathbf{\Phi}(x_1, \cdots, x_n) = \left[\begin{array}{c|c|c} \mathbf{\Gamma}(1:\kappa-1,:) & \mathbf{0}_{\kappa-1,\ell-1} & \mathbf{0}_{\kappa-1,\ell} \\ \hline \mathbb{A}^{\mathrm{Lag}} & \mathbf{\Delta}(1:\ell-1,:)^\top & \mathbf{0}_{\ell,\ell} \\ \hline \mathbb{B}^{\mathrm{Lag}} & \mathbf{0}_{\ell,\ell-1} & \mathbf{\Delta}^\top \end{array}\right] \in \mathbb{C}^{m\times m}$$

$$\begin{array}{rcl} m & = & 2\ell + \kappa - 1 \\ \\ \kappa & = & \prod_{j=1}^{k} n_j \\ \ell & = & \prod_{j=k+1}^{n} n_j \end{array}$$

Facts

- Γ gathers the first group of parameters $x_1, \cdots, x_k$
- Δ gathers the second group of parameters $x_{k+1}, \cdots, x_n$

Complexity

Re-ordering allows complexity control, e.g. according to the order of each variable x_j

Multi-variate realization

3-D case (example)

Data generated from $\mathbf{H}(x_1, x_2, x_3) = \mathbf{H}(s, t, p) = (s + pt)/(p^2 + s + t)$ of complexity $(1, 1, 2)$

$$\begin{array}{rcl} \mathbf{c}_3^\top & = & \left[\begin{array}{ccc|ccc|ccc|ccc|c} \frac{1}{2} & -\frac{39}{28} & \frac{13}{14} & -\frac{15}{28} & \frac{41}{28} & -\frac{27}{28} & -\frac{15}{28} & \frac{41}{28} & -\frac{27}{28} & \frac{4}{7} & -\frac{43}{28} & 1 \end{array}\right] \\ \mathbb{W}_3 & = & \left[\begin{array}{ccc|ccc|ccc|ccc} \frac{1}{4} & \frac{8}{39} & \frac{9}{52} & \frac{17}{30} & \frac{20}{41} & \frac{23}{54} & \frac{3}{10} & \frac{10}{41} & \frac{11}{54} & \frac{19}{32} & \frac{22}{43} & \frac{25}{56} \end{array}\right] \end{array}$$

Arrangement #1

$(s) - (t, p)$, one obtains a realization of dimension $m = 13$:

$$\kappa = 2 \text{ and } \ell = 2 \times 3$$

$$\mathbf{\Delta}(s) = \mathbb{X}^{\text{Lag}}{}_1(s)$$

$$\mathbf{\Gamma}(t, p) = \mathbb{X}^{\text{Lag}}{}_2(t) \otimes \mathbb{X}^{\text{Lag}}{}_3(p)$$

Arrangement #2

$(s, t) - (p)$, one obtains a realization of dimension $m = 9$:

$$\kappa = 2 \times 2 \text{ and } \ell = 3$$

$$\mathbf{\Delta}(s, t) = \mathbb{X}^{\text{Lag}}{}_1(s) \otimes \mathbb{X}^{\text{Lag}}{}_2(t)$$

$$\mathbf{\Gamma}(p) = \mathbb{X}^{\text{Lag}}{}_3(p)$$

Multi-variate realization

3-D case (example)

Data generated from $\mathbf{H}(x_1, x_2, x_3) = \mathbf{H}(s, t, p) = (s + pt)/(p^2 + s + t)$ of complexity $(1, 1, 2)$

$$\begin{array}{rcl} \mathbf{c}_3^\top & = & \left[\begin{array}{ccc|ccc|ccc|ccc} \frac{1}{2} & -\frac{39}{28} & \frac{13}{14} & -\frac{15}{28} & \frac{41}{28} & -\frac{27}{28} & -\frac{15}{28} & \frac{41}{28} & -\frac{27}{28} & \frac{4}{7} & -\frac{43}{28} & 1 \end{array}\right] \\ \mathbb{W}_3 & = & \left[\begin{array}{ccc|ccc|ccc|ccc} \frac{1}{4} & \frac{8}{39} & \frac{9}{52} & \frac{17}{30} & \frac{20}{41} & \frac{23}{54} & \frac{3}{10} & \frac{10}{41} & \frac{11}{54} & \frac{19}{32} & \frac{22}{43} & \frac{25}{56} \end{array}\right] \end{array}$$

$$\mathbf{\Phi} = \left[\begin{array}{ccc|cc|ccc} (s-2)\ (t-1) & -(s-2)\ (t-3) & -(t-1)\ (s-4) & (s-4)\ (t-3) & 0 & 0 & 0 & 0 & 0 \\ 1-\frac{s}{2} & \frac{s}{2}-1 & \frac{s}{2}-2 & 2-\frac{s}{2} & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2}-\frac{t}{2} & \frac{t}{2}-\frac{3}{2} & \frac{t}{2}-\frac{1}{2} & \frac{3}{2}-\frac{t}{2} & 0 & 0 & 0 & 0 & 0 \\ \hline \frac{1}{2} & -\frac{15}{28} & -\frac{15}{28} & \frac{4}{7} & p-5 & p-5 & 0 & 0 & 0 \\ -\frac{39}{28} & \frac{41}{28} & \frac{41}{28} & -\frac{43}{28} & 6-p & 0 & 0 & 0 & 0 \\ \frac{13}{14} & -\frac{27}{28} & -\frac{27}{28} & 1 & 0 & 7-p & 0 & 0 & 0 \\ \hline \frac{1}{8} & -\frac{17}{56} & -\frac{9}{56} & \frac{19}{56} & 0 & 0 & p-5 & p-5 & \frac{1}{2} \\ -\frac{2}{7} & \frac{5}{7} & \frac{5}{14} & -\frac{11}{14} & 0 & 0 & 6-p & 0 & -1 \\ \frac{9}{56} & -\frac{23}{56} & -\frac{11}{56} & \frac{25}{56} & 0 & 0 & 0 & 7-p & \frac{1}{2} \end{array}\right]$$

$$\mathbf{W} = -\mathbf{e}_9^\top \text{ and } \mathbf{G}^\top = \left[\begin{array}{c|ccc|c} \mathbf{0}_{1,3} & 1/2 & -1 & 1/2 & \mathbf{0}_{1,3} \end{array}\right]$$

Multi-variate realization

3-D case (example)

Data generated from $\mathbf{H}(x_1, x_2, x_3) = \mathbf{H}(s, t, p) = (s + pt)/(p^2 + s + t)$ of complexity $(1, 1, 2)$

$$\begin{array}{rcl} \mathbf{c}_3^\top & = & \left[\begin{array}{ccc|ccc|ccc|ccc} \frac{1}{2} & -\frac{39}{28} & \frac{13}{14} & -\frac{15}{28} & \frac{41}{28} & -\frac{27}{28} & -\frac{15}{28} & \frac{41}{28} & -\frac{27}{28} & \frac{4}{7} & -\frac{43}{28} & 1 \end{array}\right] \\ \mathbb{W}_3 & = & \left[\begin{array}{ccc|ccc|ccc|ccc} \frac{1}{4} & \frac{8}{39} & \frac{9}{52} & \frac{17}{30} & \frac{20}{41} & \frac{23}{54} & \frac{3}{10} & \frac{10}{41} & \frac{11}{54} & \frac{19}{32} & \frac{22}{43} & \frac{25}{56} \end{array}\right] \end{array}$$

$$\mathbf{\Phi}_\mathsf{c} = \left[\begin{array}{cccc|cc} (s-2)\,(t-1) & -(s-2)\,(t-3) & -(t-1)\,(s-4) & (s-4)\,(t-3) & 0 & 0 \\ 1-\frac{s}{2} & \frac{s}{2}-1 & \frac{s}{2}-2 & 2-\frac{s}{2} & 0 & 0 \\ \frac{1}{2}-\frac{t}{2} & \frac{t}{2}-\frac{3}{2} & \frac{t}{2}-\frac{1}{2} & \frac{3}{2}-\frac{t}{2} & 0 & 0 \\ \hline \frac{1}{2} & -\frac{15}{28} & -\frac{15}{28} & \frac{4}{7} & p-5 & p-5 \\ -\frac{39}{28} & \frac{41}{28} & \frac{41}{28} & -\frac{43}{28} & 6-p & 0 \\ \frac{13}{14} & -\frac{27}{28} & -\frac{27}{28} & 1 & 0 & 7-p \end{array}\right]$$

$$\mathbf{W}_\mathsf{c}(p) = \left[\begin{array}{ccc|cc} \frac{p}{28}+\frac{1}{14} & -\frac{3\,p}{28}-\frac{1}{14} & -\frac{p}{28}-\frac{1}{7} & \frac{3\,p}{28}+\frac{1}{7} & 0 & 0 \end{array}\right] \text{ and } \mathbf{G}_\mathsf{c}^\top = \left[\begin{array}{c|ccc} \mathbf{0}_{1,3} & 1/2 & -1 & 1/2 \end{array}\right]$$

Multi-variate realization

(Compressed) generalized n-D Lagrangian realization

$$\mathbf{g}(x_1, x_2, \cdots, x_n) = \frac{\sum_{j_1=1}^{k_1} \sum_{j_2=1}^{k_2} \cdots \sum_{j_n=1}^{k_n} \frac{c(j_1,j_2,\cdots,j_n)\mathbf{w}(j_1,j_2,\cdots,j_n)}{(x_1-\lambda_1(j_1))(x_2-\lambda_2(j_2))\cdots(x_n-\lambda_n(j_n))}}{\sum_{j_1=1}^{k_1} \sum_{j_2=1}^{k_2} \cdots \sum_{j_n=1}^{k_n} \frac{c(j_1,j_2,\cdots,j_n)}{(x_1-\lambda_1(j_1))(x_2-\lambda_2(j_2))\cdots(x_n-\lambda_n(j_n))}},$$

Theorem: n-D Lagrangian compressed realization

A $\ell+\kappa-1=m$-th order realization $(\hat{\mathbf{G}}_\mathsf{c}, \hat{\mathbf{\Phi}}_\mathsf{c}, \hat{\mathbf{W}}_\mathsf{c})$ of the multi-variate function $\hat{\mathbf{H}}$ in barycentric form, satisfying $\mathbf{H}(x_1,\cdots,x_n) = \hat{\mathbf{W}}_\mathsf{c}\hat{\mathbf{\Phi}}_\mathsf{c}(x_1,x_2,\cdots,x_n)^{-1}\hat{\mathbf{G}}_\mathsf{c}(x_{k+1},\cdots,x_n)$, is given by,

$$\begin{aligned}
\hat{\mathbf{\Phi}}_\mathsf{c}(x_1,\cdots,x_n) &= \left[\begin{array}{c|c} \mathbf{\Gamma}(1:\kappa-1,:) & \mathbf{0}_{\kappa-1,\ell-1} \\ \hline \mathbb{A}^{\mathrm{Lag}} & \mathbf{\Delta}(1:\ell-1,:)^\top \end{array}\right] \in \mathbb{C}^{m\times m} \\
\hat{\mathbf{G}}_\mathsf{c}(x_{k+1},\cdots,x_n) &= \left[\begin{array}{c} \mathbf{0}_{\kappa-1,1} \\ \hline \mathbf{\Delta}(\ell,:)^\top \end{array}\right] \in \mathbb{C}^{m\times 1} \\
\hat{\mathbf{W}}_\mathsf{c} &= \mathbf{e}_\ell^\top \mathbf{\Delta}^{-\top} \left[\begin{array}{c|c} \mathbb{B}^{\mathrm{Lag}} & \mathbf{0}_{\ell,\ell-1} \end{array}\right] \in \mathbb{C}^{1\times m}
\end{aligned}$$

where $\mathbb{A}^{\mathrm{Lag}}, \mathbb{B}^{\mathrm{Lag}} \in \mathbb{C}^{\ell\times\kappa}$ are appropriately chosen, according to the chosen pseudo-companion basis.

Multi-variate realization

Generalized n-D monomial realization

$$\mathbf{g}_{\text{lag}}(x_1, x_2, \cdots, x_n) = \frac{\sum_{j_1=1}^{k_1}\sum_{j_2=1}^{k_2}\cdots\sum_{j_n=1}^{k_n} \frac{c(j_1,j_2,\cdots,j_n)\mathbf{w}(j_1,j_2,\cdots,j_n)}{(x_1-\lambda_1(j_1)))(x_2-\lambda_2(j_2))\cdots(x_n-\lambda_n(j_n))}}{\sum_{j_1=1}^{k_1}\sum_{j_2=1}^{k_2}\cdots\sum_{j_n=1}^{k_n} \frac{c(j_1,j_2,\cdots,j_n)}{(x_1-\lambda_1(j_1))(x_2-\lambda_2(j_2))\cdots(x_n-\lambda_n(j_n))}},$$

$$\mathbf{g}_{\text{mon}}(x_1, x_2, \cdots, x_n) = \frac{\sum_{j_1=1}^{k_1}\sum_{j_2=1}^{k_2}\cdots\sum_{j_n=1}^{k_n} \mathbf{n}(j_1, j_2, \cdots, j_n)(x_1^{(j_1-1)}\cdots x_n^{(j_n-1)})}{\sum_{j_1=1}^{k_1}\sum_{j_2=1}^{k_2}\cdots\sum_{j_n=1}^{k_n} \mathbf{d}(j_1, j_2, \cdots, j_n)(x_1^{(j_1-1)}\cdots x_n^{(j_n-1)})},$$

Projection with Vandermonde matrices

Introducing

$$\mathbf{V} = \mathbf{V}_{x_1} \otimes \mathbf{V}_{x_2} \otimes \cdots \otimes \mathbf{V}_{x_n}$$

where $\mathbf{V}_{x_l}$ is the Vandermonde associated to the l-th variable. Monomial coefficients for the numerator $\mathbf{n}(j_1, j_2, \cdots, j_n) = \mathbf{V}(\mathbf{w} \odot \mathbf{c})$ and denominator as $\mathbf{d}(j_1, j_2, \cdots, j_n) = \mathbf{V}\mathbf{c}$

W. Gander, *"Change of basis in polynomial interpolation"*, Numerical Linear Algebra with Applications 12(8), 2005.

Multi-variate realization

3-D case (example)

Data generated from $\mathbf{H}(x_1, x_2, x_3) = \mathbf{H}(s, t, p) = (s + pt)/(p^2 + s + t)$ of complexity $(1, 1, 2)$

$$\begin{array}{rcl} \mathbf{c}_3^\top & = & \left[\begin{array}{ccc|ccc|ccc|ccc} \frac{1}{2} & -\frac{39}{28} & \frac{13}{14} & -\frac{15}{28} & \frac{41}{28} & -\frac{27}{28} & -\frac{15}{28} & \frac{41}{28} & -\frac{27}{28} & \frac{4}{7} & -\frac{43}{28} & 1 \end{array}\right] \\ \mathbb{W}_3 & = & \left[\begin{array}{ccc|ccc|ccc|ccc} \frac{1}{4} & \frac{8}{39} & \frac{9}{52} & \frac{17}{30} & \frac{20}{41} & \frac{23}{54} & \frac{3}{10} & \frac{10}{41} & \frac{11}{54} & \frac{19}{32} & \frac{22}{43} & \frac{25}{56} \end{array}\right] \end{array}$$

Arrangement #1

$(s) - (t, p)$, one obtains a realization of dimension $m = 13$:

$$\kappa = 2 \text{ and } \ell = 2 \times 3$$

$$\mathbf{\Delta}(s) = \mathbb{X}^{\mathrm{Lag}}{}_1(s)$$

$$\mathbf{\Gamma}(t, p) = \mathbb{X}^{\mathrm{Lag}}{}_2(t) \otimes \mathbb{X}^{\mathrm{Lag}}{}_3(p)$$

Arrangement #2

$(s, t) - (p)$, one obtains a realization of dimension $m = 9$:

$$\kappa = 2 \times 2 \text{ and } \ell = 3$$

$$\mathbf{\Delta}(s, t) = \mathbb{X}^{\mathrm{Lag}}{}_1(s) \otimes \mathbb{X}^{\mathrm{Lag}}{}_2(t)$$

$$\mathbf{\Gamma}(p) = \mathbb{X}^{\mathrm{Lag}}{}_3(p)$$

Multi-variate realization

3-D case (example)

Data generated from $\mathbf{H}(x_1, x_2, x_3) = \mathbf{H}(s, t, p) = (s + pt)/(p^2 + s + t)$ of complexity $(1, 1, 2)$

$$
\begin{array}{rcl}
\mathbf{c}_3^\top & = & \left[\begin{array}{ccc|ccc|ccc|ccc} \frac{1}{2} & -\frac{39}{28} & \frac{13}{14} & -\frac{15}{28} & \frac{41}{28} & -\frac{27}{28} & -\frac{15}{28} & \frac{41}{28} & -\frac{27}{28} & \frac{4}{7} & -\frac{43}{28} & 1 \end{array}\right] \\
\mathbb{W}_3 & = & \left[\begin{array}{ccc|ccc|ccc|ccc} \frac{1}{4} & \frac{8}{39} & \frac{9}{52} & \frac{17}{30} & \frac{20}{41} & \frac{23}{54} & \frac{3}{10} & \frac{10}{41} & \frac{11}{54} & \frac{19}{32} & \frac{22}{43} & \frac{25}{56} \end{array}\right]
\end{array}
$$

$$
\mathbf{\Phi} = \left[\begin{array}{ccccc|cc|ccc}
(s-2)\,(t-1) & -(s-2)\,(t-3) & -(t-1)\,(s-4) & (s-4)\,(t-3) & & 0 & 0 & 0 & 0 & 0 \\
1-\frac{s}{2} & \frac{s}{2}-1 & \frac{s}{2}-2 & 2-\frac{s}{2} & & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{2}-\frac{t}{2} & \frac{t}{2}-\frac{3}{2} & \frac{t}{2}-\frac{1}{2} & \frac{3}{2}-\frac{t}{2} & & 0 & 0 & 0 & 0 & 0 \\
\hline
\frac{1}{2} & -\frac{15}{28} & -\frac{15}{28} & \frac{4}{7} & & p-5 & p-5 & 0 & 0 & 0 \\
-\frac{39}{28} & \frac{41}{28} & \frac{41}{28} & -\frac{43}{28} & & 6-p & 0 & 0 & 0 & 0 \\
\frac{13}{14} & -\frac{27}{28} & -\frac{27}{28} & 1 & & 0 & 7-p & 0 & 0 & 0 \\
\hline
\frac{1}{8} & -\frac{17}{56} & -\frac{9}{56} & \frac{19}{56} & & 0 & 0 & p-5 & p-5 & \frac{1}{2} \\
-\frac{2}{7} & \frac{5}{7} & \frac{5}{14} & -\frac{11}{14} & & 0 & 0 & 6-p & 0 & -1 \\
\frac{9}{56} & -\frac{23}{56} & -\frac{11}{56} & \frac{25}{56} & & 0 & 0 & 0 & 7-p & \frac{1}{2}
\end{array}\right]
$$

$$
\mathbf{W} = -\mathbf{e}_9^\top \text{ and } \mathbf{G}^\top = \left[\begin{array}{c|ccc|c} \mathbf{0}_{1,3} & 1/2 & -1 & 1/2 & \mathbf{0}_{1,3} \end{array}\right]
$$

Multi-variate realization

3-D case (example)

Data generated from $\mathbf{H}(x_1, x_2, x_3) = \mathbf{H}(s, t, p) = (s + pt)/(p^2 + s + t)$ of complexity $(1, 1, 2)$

$$\begin{array}{rcl} \mathbf{c}_3^\top & = & \left[\begin{array}{ccc|ccc|ccc|ccc} \frac{1}{2} & -\frac{39}{28} & \frac{13}{14} & -\frac{15}{28} & \frac{41}{28} & -\frac{27}{28} & -\frac{15}{28} & \frac{41}{28} & -\frac{27}{28} & \frac{4}{7} & -\frac{43}{28} & 1 \end{array}\right] \\ \mathbb{W}_3 & = & \left[\begin{array}{ccc|ccc|ccc|ccc} \frac{1}{4} & \frac{8}{39} & \frac{9}{52} & \frac{17}{30} & \frac{20}{41} & \frac{23}{54} & \frac{3}{10} & \frac{10}{41} & \frac{11}{54} & \frac{19}{32} & \frac{22}{43} & \frac{25}{56} \end{array}\right] \end{array}$$

$$\mathbf{\Phi}_{\mathsf{c}} = \left[\begin{array}{cccc|cc} (s-2)\,(t-1) & -(s-2)\,(t-3) & -(t-1)\,(s-4) & (s-4)\,(t-3) & 0 & 0 \\ 1-\frac{s}{2} & \frac{s}{2}-1 & \frac{s}{2}-2 & 2-\frac{s}{2} & 0 & 0 \\ \frac{1}{2}-\frac{t}{2} & \frac{t}{2}-\frac{3}{2} & \frac{t}{2}-\frac{1}{2} & \frac{3}{2}-\frac{t}{2} & 0 & 0 \\ \hline \frac{1}{2} & -\frac{15}{28} & -\frac{15}{28} & \frac{4}{7} & p-5 & p-5 \\ -\frac{39}{28} & \frac{41}{28} & \frac{41}{28} & -\frac{43}{28} & 6-p & 0 \\ \frac{13}{14} & -\frac{27}{28} & -\frac{27}{28} & 1 & 0 & 7-p \end{array}\right]$$

$$\mathbf{W}_{\mathsf{c}}(p) = \left[\begin{array}{ccc|cc} \frac{p}{28}+\frac{1}{14} & -\frac{3\,p}{28}-\frac{1}{14} & -\frac{p}{28}-\frac{1}{7} & \frac{3\,p}{28}+\frac{1}{7} & 0 & 0 \end{array}\right] \text{ and } \mathbf{G}_{\mathsf{c}}^\top = \left[\begin{array}{c|ccc} \mathbf{0}_{1,3} & 1/2 & -1 & 1/2 \end{array}\right]$$

Content

Forewords

Multi-variate data, function & Loewner matrix

Taming the curse of dimensionality

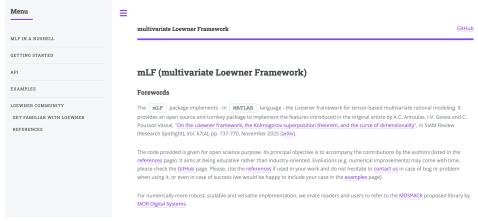
Variables decoupling, KST and KANs

Comparisons

Multi-variate realization

The MATLAB `mLF` package

Conclusion



The MATLAB `mLF` package

GitHub page

https://cpoussot.github.io/mlf/index.html

```
%%% Chose model
[H,infoCas] = mlf.examples(34);
n           = infoCas.n;
p_c         = infoCas.p_c;
p_r         = infoCas.p_r;
%%% Data tensor/rand
[y,x,dim]   = mlf.make_tab_vec(H,p_c,p_r);
tab         = mlf.vec2mat(y,dim);
%%% Alg. 1: direct pLoe [A/G/P-V, 2025]
opt.ord_tol      = 1e-9;
opt.method_null  = 'svd0';
opt.method       = 'rec';
opt.ord_obj      = [];
opt.ord_N        = 10;
opt.ord_show     = false;
opt.data_min     = true;
[g,iloe1]        = mlf.alg1(tab,p_c,p_r,opt);
```

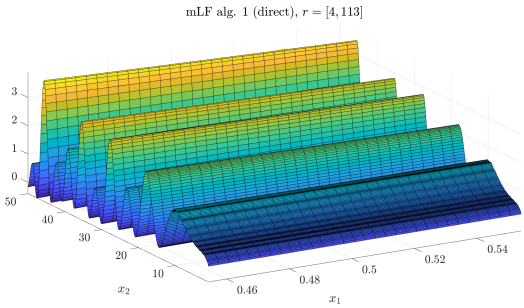


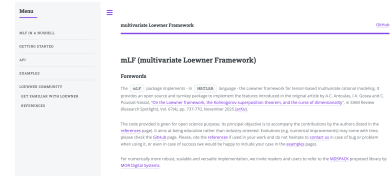
- Generic algorithm `mlf.alg1`
- Specific steps `mlf.compute_order`, `mlf.point_selection`, `mlf.loewner_null_rec`, `mlf.tf_lagrangian`, ...

The MATLAB `mLF` package

GitHub page

`https://cpoussot.github.io/mlf/index.html`





- Generic algorithm `mlf.alg1`
- Specific steps `mlf.compute_order`, `mlf.point_selection`, `mlf.loewner_null_rec`, `mlf.tf_lagrangian`, ...

The MATLAB mLF package

Detailed example 2-D (define the problem)

```
%%% Define a multivariate problem
syms x1 x2;
Hsym    = x1*x2^3+2*x1*x2-1;
Hf      = matlabFunction(Hsym);
H       = @(x) Hf(x(:,1),x(:,2));
n       = 2; % number of variables
xbnd    = [-1 1]; % bounds of both variables
Nip     = 10; % number of interpolation points
X1      = linspace(xbnd(1),xbnd(2),Nip);
X2      = linspace(xbnd(1),xbnd(2),Nip);
```

$$\mathbf{H}(x_1, x_2) = x_1 \, {x_2}^3 + 2\, x_1 \, x_2 - 1$$

- $n = 2$, bounds $\mathcal{X}_l := \begin{bmatrix} -1 & 1 \end{bmatrix}$, discretized with $N_l = 10$, $l = \{1, 2\}$
- $\mathbf{x}_l = \begin{bmatrix} -1 & -\frac{7}{9} & -\frac{5}{9} & -\frac{1}{3} & -\frac{1}{9} & \frac{1}{9} & \frac{1}{3} & \frac{5}{9} & \frac{7}{9} & 1 \end{bmatrix} \in \mathcal{X}_l^{10}$

The MATLAB mLF package

Detailed example 2-D (select interpolation points and construct the tensor)

```
%%% Interpolation points
p_c{1}  = X1(2:2:end);
p_r{1}  = X1(1:2:end);
p_c{2}  = X2(2:2:end);
p_r{2}  = X2(1:2:end);
```

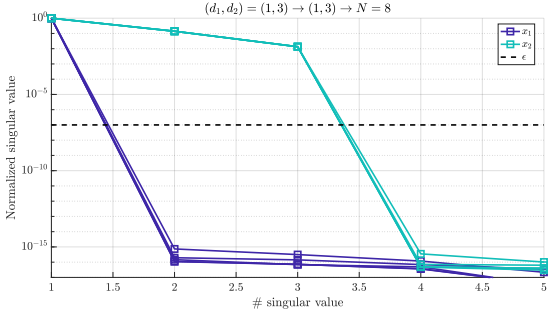
```
%%% Construct tensor
% Check that column/row IP are disjoint
ok          = mlf.check(p_c,p_r);
% Construct tab_n
[y,x,dim]   = mlf.make_tab_vec(H,p_c,p_r);
tab         = mlf.vec2mat(y,dim);
```

- columns $\lambda_l(j_l)$ (denoted p_c{1} and p_c{2}) and
- rows $\mu_l(i_l)$ (denoted p_r{1} and p_r{2})
- we chose equal dimensions (i.e. $k_l = 5$ and $q_l = 5$)
- IP are disjoints, and 2-D tensor $\mathcal{T}_2^{\otimes} \in \mathbb{R}^{(q_1+q_2)\times(k_1+k_2)}$

The MATLAB mLF package

Detailed example 2-D (estimate the orders)

```
%%% Estimate orders along each variables and select a subset of IP
tol                 = 1e-7;
ord                 = mlf.compute_order(p_c, p_r, tab, tol, [], 5, true);
[pc, pr, W, V, tabr] = mlf.points_selection(p_c, p_r, tab, ord, true);
w                   = mlf.mat2vec(W);
```



The MATLAB mLF package

Detailed example 2-D (estimate the orders)

```
%%% Estimate orders along each variables and select a subset of IP
tol                 = 1e-7;
ord                 = mlf.compute_order(p_c,p_r,tab,tol,[],5,true);
[pc,pr,W,V,tabr]    = mlf.points_selection(p_c,p_r,tab,ord,true);
w                   = mlf.mat2vec(W);
```

- $(k_1, k_2) = (2, 4)$ column IP
- Barycentric dimension $2 \times 4 = 8 = N$
- $\lambda_1(j_1) = \begin{bmatrix} -\frac{7}{9} & 1 \end{bmatrix}$ (pc{1})
- $\lambda_2(j_2) = \begin{bmatrix} -\frac{7}{9} & -\frac{1}{3} & \frac{1}{9} & 1 \end{bmatrix}$ (pc{2})
- $\mu_1(i_1) = \begin{bmatrix} -1 & \frac{7}{9} \end{bmatrix}$ (pr{1})
- $\mu_2(i_2) = \begin{bmatrix} -1 & -\frac{5}{9} & -\frac{1}{9} & \frac{7}{9} \end{bmatrix}$ (pr{2})

The MATLAB mLF package

Detailed example 2-D (compute null space)

```
%%% Compute rec. LL null
[c,lag] = mlf.loewner_null_rec(pc,pr,tabr,'svd0');
```

- Recursive: 136 flop and 16 Bytes
- Full: 512 flop and 64 Bytes
- $\mathbf{W}_{2,4}^{\otimes} = \begin{bmatrix} \frac{3778}{6561} & -\frac{110}{243} & -\frac{7702}{6561} & -\frac{10}{3} \\ -\frac{2206}{729} & -\frac{46}{27} & -\frac{566}{729} & 2 \end{bmatrix}$
- $\mathbf{C}_{2,4}^{\otimes} = \begin{bmatrix} 3 & -8 & 6 & -1 \\ -3 & 8 & -6 & 1 \end{bmatrix}$

The MATLAB `mLF` package

Detailed example 2-D (Lagrange form)

```
%%% Transfer function in Lagrangian form
[glag, ilag] = mlf.tf_lagrangian(pc,w,c,false);
```

$$\left(\begin{array}{c|c|c} \mathcal{B}_{\text{lag}}(x_1,x_2) & \mathbf{N}_{\text{lag}} & \mathbf{D}_{\text{lag}} \end{array} \right)$$

$$\left(\begin{array}{c|c|c} \left(x_1+\frac{7}{9}\right)\left(x_2+\frac{7}{9}\right) & \frac{3778}{2187} & 3 \\ \left(x_2+\frac{1}{3}\right)\left(x_1+\frac{7}{9}\right) & \frac{880}{243} & -8 \\ \left(x_2-\frac{1}{9}\right)\left(x_1+\frac{7}{9}\right) & -\frac{15404}{2187} & 6 \\ (x_2-1)\left(x_1+\frac{7}{9}\right) & \frac{10}{3} & -1 \\ (x_1-1)\left(x_2+\frac{7}{9}\right) & \frac{2206}{243} & -3 \\ (x_1-1)\left(x_2+\frac{1}{3}\right) & -\frac{368}{27} & 8 \\ (x_1-1)\left(x_2-\frac{1}{9}\right) & \frac{1132}{243} & -6 \\ (x_1-1)\,(x_2-1) & 2 & 1 \end{array} \right)$$

$$\mathbf{G}_{\text{lag}}(x_1,x_2) = \frac{\sum_{j_1=1}^{2}\sum_{j_2=1}^{4}\frac{\mathbf{C}_{2,4}^{\otimes}(j_1,j_2)\mathbf{W}_{2,4}^{\otimes}(j_1,j_2)}{(x_1-\lambda_1(j_1))(x_2-\lambda_2(j_2))}}{\sum_{j_1=1}^{2}\sum_{j_2=1}^{4}\frac{\mathbf{C}_{2,4}^{\otimes}(j_1,j_2)}{(x_1-\lambda_1(j_1))(x_2-\lambda_2(j_2))}},$$

where

$$\mathbf{W}_{2,4}^{\otimes} = \begin{bmatrix} \frac{3778}{6561} & -\frac{110}{243} & -\frac{7702}{6561} & -\frac{10}{3} \\ -\frac{2206}{729} & -\frac{46}{27} & -\frac{566}{729} & 2 \end{bmatrix}$$

$$\mathbf{C}_{2,4}^{\otimes} = \begin{bmatrix} 3 & -8 & 6 & -1 \\ -3 & 8 & -6 & 1 \end{bmatrix}$$

The MATLAB mLF package

Detailed example 2-D (Monomial form)

```
%%% Transfer function in Monomial form
[gmon,imon] = mlf.tf_monomial(pc,w,c,false);
```

$$\left(\begin{array}{c|c|c} \mathcal{B}_{\mathrm{mon}}(x_1, x_2) & \mathbf{N}_{\mathrm{mon}} & \mathbf{D}_{\mathrm{mon}} \end{array} \right)$$

$$\left(\begin{array}{c|c|c} x_1\, {x_2}^3 & 1 & 0 \\ x_1\, {x_2}^2 & 0 & 0 \\ x_1\, x_2 & 2 & 0 \\ x_1 & 0 & 0 \\ {x_2}^3 & 0 & 0 \\ {x_2}^2 & 0 & 0 \\ x_2 & 0 & 0 \\ 1 & -1 & 1 \end{array} \right)$$

$$\mathbf{G}_{\mathrm{mon}}(x_1, x_2) = \frac{\sum_{j_1=1}^{2} \sum_{j_2=1}^{4} \mathbf{N}_{2,4}^{\otimes}(j_1, j_2) \left(x_1^{(j_1-1)} x_2^{(j_2-1)} \right)}{\sum_{j_1=1}^{2} \sum_{j_2=1}^{4} \mathbf{D}_{2,4}^{\otimes}(j_1, j_2) \left(x_1^{(j_1-1)} x_2^{(j_2-1)} \right)},$$

where

$$\mathbf{W}_{2,4}^{\otimes} = \begin{bmatrix} 1 & 0 & 2 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

$$\mathbf{C}_{2,4}^{\otimes} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The MATLAB mLF package

Detailed example 2-D (KST form)

```
%%% Decoupling toward KST
lag_struct.pc   = pc;
lag_struct.lag  = lag;
[Bary,Lag,Cx]   = mlf.decoupling(lag_struct);
```

$$\begin{bmatrix} \mathbf{Bary}(x_1) & \mathbf{Bary}(x_2) \end{bmatrix} = \begin{bmatrix} -1 & -3 \\ -1 & 8 \\ -1 & -6 \\ -1 & 1 \\ 1 & -3 \\ 1 & 8 \\ 1 & -6 \\ 1 & 1 \end{bmatrix}$$

$$\begin{bmatrix} \mathbf{Lag}(x_1) & \mathbf{Lag}(x_2) \end{bmatrix} = \begin{bmatrix} \frac{1}{x_1+\frac{7}{9}} & \frac{1}{x_2+\frac{7}{9}} \\ \frac{1}{x_1+\frac{7}{9}} & \frac{1}{x_2+\frac{1}{3}} \\ \frac{1}{x_1+\frac{7}{9}} & \frac{1}{x_2-\frac{1}{9}} \\ \frac{1}{x_1+\frac{7}{9}} & \frac{1}{x_2-1} \\ \frac{1}{x_1-1} & \frac{1}{x_2+\frac{7}{9}} \\ \frac{1}{x_1-1} & \frac{1}{x_2+\frac{1}{3}} \\ \frac{1}{x_1-1} & \frac{1}{x_2-\frac{1}{9}} \\ \frac{1}{x_1-1} & \frac{1}{x_2-1} \end{bmatrix}$$

Univariate vector functions $\mathbf{\Phi}_1(x_1)$ and $\mathbf{\Phi}_2(x_2)$

$$\begin{cases} \mathbf{Bary}(x_1) \odot \mathbf{Lag}(x_1) \\ \mathbf{Bary}(x_2) \odot \mathbf{Lag}(x_2) \end{cases}$$

KST rational interpolant

$$\mathbf{H}_{\mathrm{kst}}(x_1, x_2) = \frac{\mathbf{n}_{\mathrm{kst}}(x_1, x_2)}{\mathbf{d}_{\mathrm{kst}}(x_1, x_2)}$$

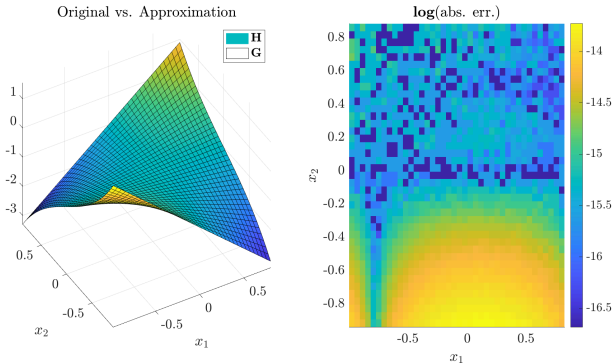
where numerator and denominator are

$$\begin{cases} \sum_{\text{rows}} \mathbf{w} \odot \mathbf{\Phi}_1(x_1) \odot \mathbf{\Phi}_2(x_2) \\ \sum_{\text{rows}} \mathbf{\Phi}_1(x_1) \odot \mathbf{\Phi}_2(x_2) \end{cases}$$

The MATLAB mLF package

Detailed example 2-D (KST form)

$$\mathbf{H}(x_1, x_2) = x_1 \, {x_2}^3 + 2\, x_1 \, x_2 - 1$$



The MATLAB mLF package

Detailed example 2-D (define the problem)

```
%%% Define a multivariate problem
syms x1 x2;
Hsym    = (sin(.01*x2+pi)+1)/(tanh(x1)*x2+3);
Hf      = matlabFunction(Hsym);
H       = @(x) Hf(x(:,1),x(:,2));
n       = 2; % number of variables
xbnd    = [-2 2]; % bounds of both variables
Nip     = 12; % number of interpolation points
X1      = linspace(xbnd(1),xbnd(2),Nip);
X2      = linspace(xbnd(1),xbnd(2),Nip);
```

$$\mathbf{H}(x_1, x_2) = -\frac{\sin\left(\frac{x_2}{100}\right) - 1}{x_2 \tanh(x_1) + 3}$$

- $n = 2$, bounds $\mathcal{X}_l := \begin{bmatrix} -2.0 & 2.0 \end{bmatrix}$, discretized with $N_l = 12$, $l = \{1, 2\}$
- $\mathbf{x}_l = \begin{bmatrix} -2.0 & -1.64 & -1.27 & -0.909 & -0.545 & -0.182 & 0.182 & 0.545 & 0.909 & 1.27 & 1.64 & 2.0 \end{bmatrix} \in \mathcal{X}_l^{12}$

The MATLAB mLF package

Detailed example 2-D (select interpolation points and construct the tensor)

```
%%% Interpolation points
p_c{1}  = X1(2:2:end);
p_r{1}  = X1(1:2:end);
p_c{2}  = X2(2:2:end);
p_r{2}  = X2(1:2:end);
```

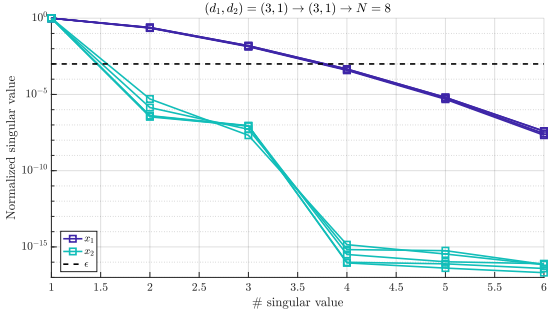
```
%%% Construct tensor
% Check that column/row IP are disjoint
ok          = mlf.check(p_c,p_r);
% Construct tab_n
[y,x,dim]   = mlf.make_tab_vec(H,p_c,p_r);
tab         = mlf.vec2mat(y,dim);
```

- columns $\lambda_l(j_l)$ (denoted p_c{1} and p_c{2}) and
- rows $\mu_l(i_l)$ (denoted p_r{1} and p_r{2})
- we chose equal dimensions (i.e. $k_l = 6$ and $q_l = 6$)
- IP are disjoints, and 2-D tensor $\mathcal{T}_2^{\otimes} \in \mathbb{R}^{(q_1+q_2)\times(k_1+k_2)}$

The MATLAB mLF package

Detailed example 2-D (estimate the orders)

```
%%% Estimate orders along each variables and select a subset of IP
tol                 = 1e-3;
ord                 = mlf.compute_order(p_c,p_r,tab,tol,[],5,true);
[pc,pr,W,V,tabr]    = mlf.points_selection(p_c,p_r,tab,ord,true);
w                   = mlf.mat2vec(W);
```



The MATLAB mLF package

Detailed example 2-D (estimate the orders)

```
%%% Estimate orders along each variables and select a subset of IP
tol                 = 1e-3;
ord                 = mlf.compute_order(p_c,p_r,tab,tol,[],5,true);
[pc,pr,W,V,tabr]    = mlf.points_selection(p_c,p_r,tab,ord,true);
w                   = mlf.mat2vec(W);
```

- $(k_1, k_2) = (4, 2)$ column IP
- Barycentric dimension $4 \times 2 = 8 = N$
- $\lambda_1(j_1) = \begin{bmatrix} -1.64 & -0.909 & 0.545 & 2.0 \end{bmatrix}$ (pc{1})
- $\lambda_2(j_2) = \begin{bmatrix} -1.64 & 2.0 \end{bmatrix}$ (pc{2})
- $\mu_1(i_1) = \begin{bmatrix} -2.0 & -1.27 & 0.182 & 1.64 \end{bmatrix}$ (pr{1})
- $\mu_2(i_2) = \begin{bmatrix} -2.0 & 1.64 \end{bmatrix}$ (pr{2})

The MATLAB mLF package

Detailed example 2-D (compute null space)

```
%%% Compute rec. LL null
[c,lag] = mlf.loewner_null_rec(pc,pr,tabr,'svd0');
```

- Recursive: 96 flop and 16.0 Bytes
- Full: 512 flop and 64.0 Bytes
- $\mathbf{W}_{4,2}^{\otimes} = \begin{bmatrix} 0.225 & 0.855 \\ 0.243 & 0.629 \\ 0.465 & 0.245 \\ 0.714 & 0.199 \end{bmatrix}$
- $\mathbf{C}_{4,2}^{\otimes} = \begin{bmatrix} -1.79 & 0.453 \\ 1.97 & -0.735 \\ -0.547 & 1.0 \\ 0.233 & -0.808 \end{bmatrix}$

The MATLAB mLF package

Detailed example 2-D (Lagrange form)

```
%%% Transfer function in Lagrangian form
[glag, ilag] = mlf.tf_lagrangian(pc,w,c,false);
```

$$\left(\begin{array}{c|c|c} \mathcal{B}_{\text{lag}}(x_1, x_2) & \mathbf{N}_{\text{lag}} & \mathbf{D}_{\text{lag}} \end{array} \right)$$

$$\left(\begin{array}{c|c|c} (x_1 + 1.64)\ (x_2 + 1.64) & -0.402 & -1.79 \\ (x_2 - 2.0)\ (x_1 + 1.64) & 0.387 & 0.453 \\ (x_2 + 1.64)\ (x_1 + 0.909) & 0.479 & 1.97 \\ (x_2 - 2.0)\ (x_1 + 0.909) & -0.462 & -0.735 \\ (x_1 - 0.545)\ (x_2 + 1.64) & -0.254 & -0.547 \\ (x_2 - 2.0)\ (x_1 - 0.545) & 0.245 & 1.0 \\ (x_1 - 2.0)\ (x_2 + 1.64) & 0.167 & 0.233 \\ (x_1 - 2.0)\ (x_2 - 2.0) & -0.161 & -0.808 \end{array} \right)$$

$$\mathbf{G}_{\text{lag}}(x_1, x_2) = \frac{\sum_{j_1=1}^{4} \sum_{j_2=1}^{2} \frac{\mathbf{C}_{4,2}^{\otimes}(j_1,j_2)\mathbf{W}_{4,2}^{\otimes}(j_1,j_2)}{(x_1-\lambda_1(j_1))(x_2-\lambda_2(j_2))}}{\sum_{j_1=1}^{4} \sum_{j_2=1}^{2} \frac{\mathbf{C}_{4,2}^{\otimes}(j_1,j_2)}{(x_1-\lambda_1(j_1))(x_2-\lambda_2(j_2))}},$$

where

$$\mathbf{W}_{4,2}^{\otimes} = \begin{bmatrix} 0.225 & 0.855 \\ 0.243 & 0.629 \\ 0.465 & 0.245 \\ 0.714 & 0.199 \end{bmatrix}$$

$$\mathbf{C}_{4,2}^{\otimes} = \begin{bmatrix} -1.79 & 0.453 \\ 1.97 & -0.735 \\ -0.547 & 1.0 \\ 0.233 & -0.808 \end{bmatrix}$$

The MATLAB mLF package

Detailed example 2-D (Monomial form)

```
%%% Transfer function in Monomial form
[gmon,imon] = mlf.tf_monomial(pc,w,c,false);
```

$$\left(\begin{array}{c|c|c} \mathcal{B}_{\mathrm{mon}}(x_1,x_2) & \mathbf{N}_{\mathrm{mon}} & \mathbf{D}_{\mathrm{mon}} \end{array} \right)$$

$$\left(\begin{array}{c|c|c} {x_1}^3\,x_2 & -3.27e-5 & -0.019 \\ {x_1}^3 & 0.00328 & 0.00983 \\ {x_1}^2\,x_2 & 0.00129 & 0.0134 \\ {x_1}^2 & -0.129 & -0.386 \\ x_1\,x_2 & -1.48e-4 & -0.333 \\ x_1 & 0.0148 & 0.0444 \\ x_2 & 0.00333 & 2.87e-4 \\ 1.0 & -0.333 & -1.0 \end{array} \right)$$

$$\mathbf{G}_{\mathrm{mon}}(x_1,x_2) = \frac{\sum_{j1=1}^{4}\sum_{j2=1}^{2}\mathbf{N}_{4,2}^{\otimes}(j_1,j_2)\left(x_1^{(j_1-1)}x_2^{(j_2-1)}\right)}{\sum_{j1=1}^{4}\sum_{j2=1}^{2}\mathbf{D}_{4,2}^{\otimes}(j_1,j_2)\left(x_1^{(j_1-1)}x_2^{(j_2-1)}\right)},$$

where

$$\mathbf{W}_{4,2}^{\otimes} = \begin{bmatrix} -3.27e-5 & 0.00328 \\ 0.00129 & -0.129 \\ -1.48e-4 & 0.0148 \\ 0.00333 & -0.333 \end{bmatrix}$$

$$\mathbf{C}_{4,2}^{\otimes} = \begin{bmatrix} -0.019 & 0.00983 \\ 0.0134 & -0.386 \\ -0.333 & 0.0444 \\ 2.87e-4 & -1.0 \end{bmatrix}$$

The MATLAB mLF package

Detailed example 2-D (KST form)

```
%%% Decoupling toward KST
lag_struct.pc   = pc;
lag_struct.lag  = lag;
[Bary,Lag,Cx]   = mlf.decoupling(lag_struct);
```

$$\begin{bmatrix} \mathbf{Bary}(x_1) & \mathbf{Bary}(x_2) \end{bmatrix} = \begin{bmatrix} 0.453 & -3.94 \\ 0.453 & 1.0 \\ -0.735 & -2.68 \\ -0.735 & 1.0 \\ 1.0 & -0.547 \\ 1.0 & 1.0 \\ -0.808 & -0.289 \\ -0.808 & 1.0 \end{bmatrix}$$

$$\begin{bmatrix} \mathbf{Lag}(x_1) & \mathbf{Lag}(x_2) \end{bmatrix} = \begin{bmatrix} \frac{1}{x_1+1.64} & \frac{1}{x_2+1.64} \\ \frac{1}{x_1+1.64} & \frac{1}{x_2-2.0} \\ \frac{1}{x_1+0.909} & \frac{1}{x_2+1.64} \\ \frac{1}{x_1+0.909} & \frac{1}{x_2-2.0} \\ \frac{1}{x_1-0.545} & \frac{1}{x_2+1.64} \\ \frac{1}{x_1-0.545} & \frac{1}{x_2-2.0} \\ \frac{1}{x_1-2.0} & \frac{1}{x_2+1.64} \\ \frac{1}{x_1-2.0} & \frac{1}{x_2-2.0} \end{bmatrix}$$

Univariate vector functions $\mathbf{\Phi}_1(x_1)$ and $\mathbf{\Phi}_2(x_2)$

$$\begin{cases} \mathbf{Bary}(x_1) \odot \mathbf{Lag}(x_1) \\ \mathbf{Bary}(x_2) \odot \mathbf{Lag}(x_2) \end{cases}$$

KST rational interpolant

$$\mathbf{H}_{\mathrm{kst}}(x_1, x_2) = \frac{\mathbf{n}_{\mathrm{kst}}(x_1, x_2)}{\mathbf{d}_{\mathrm{kst}}(x_1, x_2)}$$

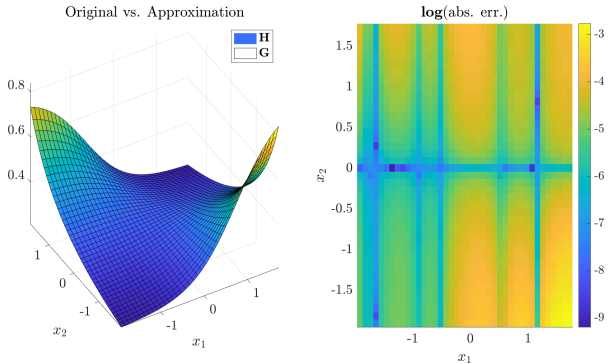
where numerator and denominator are

$$\begin{cases} \sum_{\text{rows}} \mathbf{w} \odot \mathbf{\Phi}_1(x_1) \odot \mathbf{\Phi}_2(x_2) \\ \sum_{\text{rows}} \mathbf{\Phi}_1(x_1) \odot \mathbf{\Phi}_2(x_2) \end{cases}$$

The MATLAB mLF package

Detailed example 2-D (KST form)

$$\mathbf{H}(x_1, x_2) = -\frac{\sin\left(\frac{x_2}{100}\right) - 1}{x_2 \tanh(x_1) + 3}$$



The MATLAB mLF package

Detailed example 3-D (define the problem)

```
%%% Define a multivariate problem
syms x1 x2 x3;
Hsym    = (x1+x2^2*x3)/(x2*x1-10);
Hf      = matlabFunction(Hsym);
H       = @(x) Hf(x(:,1),x(:,2),x(:,3));
n       = 3; % number of variables
xbnd    = [-2 2]; % bounds of both variables
Nip     = 6; % number of interpolation points
X1      = linspace(xbnd(1),xbnd(2),Nip);
X2      = linspace(xbnd(1),xbnd(2),Nip);
X3      = linspace(xbnd(1),xbnd(2),Nip);
```

$$\mathbf{H}(x_1, x_2, x_3) = \frac{x_3 {x_2}^2 + x_1}{x_1 x_2 - 10}$$

- $n = 3$, bounds $\mathcal{X}_l := \begin{bmatrix} -2.0 & 2.0 \end{bmatrix}$, discretized with $N_l = 6$, $l = \{1, 2\}$
- $\mathbf{x}_l = \begin{bmatrix} -2.0 & -1.2 & -0.4 & 0.4 & 1.2 & 2.0 \end{bmatrix} \in \mathcal{X}_l^6$

The MATLAB mLF package

Detailed example 3-D (select interpolation points and construct the tensor)

```
%%% Interpolation points
p_c{1}  = X1(2:2:end);
p_r{1}  = X1(1:2:end);
p_c{2}  = X2(2:2:end);
p_r{2}  = X2(1:2:end);
p_c{3}  = X3(2:2:end);
p_r{3}  = X3(1:2:end);
```

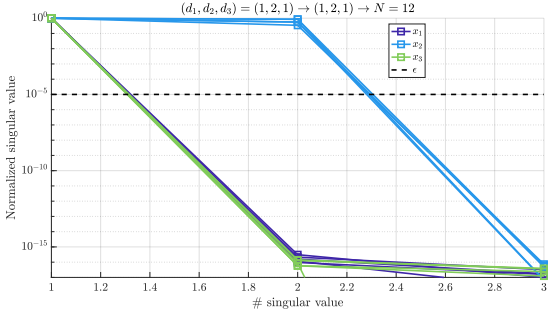
```
%%% Construct tensor
% Check that column/row IP are disjoint
ok          = mlf.check(p_c,p_r);
% Construct tab_n
[y,x,dim]   = mlf.make_tab_vec(H,p_c,p_r);
tab         = mlf.vec2mat(y,dim);
```

- columns $\lambda_l(j_l)$ (denoted p_c{1}, p_c{2} and p_c{3}) and
- rows $\mu_l(i_l)$ (denoted p_r{1}, p_r{2} and p_r{3})
- we chose equal dimensions (i.e. $k_l = 3$ and $q_l = 3$)
- IP are disjoints, and 2-D tensor $\mathcal{T}_2^{\otimes} \in \mathbb{R}^{(q_1+q_2+q_3)\times(k_1+k_2+k_3)}$

The MATLAB mLF package

Detailed example 3-D (estimate the orders)

```
%%% Estimate orders along each variables and select a subset of IP
tol                 = 1e-5;
ord                 = mlf.compute_order(p_c,p_r,tab,tol,[],5,true);
[pc,pr,W,V,tabr]    = mlf.points_selection(p_c,p_r,tab,ord,true);
w                   = mlf.mat2vec(W);
```



The MATLAB mLF package

Detailed example 3-D (estimate the orders)

```
%%% Estimate orders along each variables and select a subset of IP
tol                 = 1e-5;
ord                 = mlf.compute_order(p_c, p_r, tab, tol, [], 5, true);
[pc, pr, W, V, tabr] = mlf.points_selection(p_c, p_r, tab, ord, true);
w                   = mlf.mat2vec(W);
```

- $(k_1, k_2, k_3) = (2, 3, 2)$ column IP
- Barycentric dimension $2 \times 3 \times 2 = 12 = N$
- $\lambda_1(j_1) = \begin{bmatrix} -1.2 & 2.0 \end{bmatrix}$ (pc{1})
- $\lambda_2(j_2) = \begin{bmatrix} -1.2 & 0.4 & 2.0 \end{bmatrix}$ (pc{2})
- $\lambda_3(j_3) = \begin{bmatrix} -1.2 & 2.0 \end{bmatrix}$ (pc{3})
- $\mu_1(i_1) = \begin{bmatrix} -2.0 & 1.2 \end{bmatrix}$ (pr{1})
- $\mu_2(i_2) = \begin{bmatrix} -2.0 & -0.4 & 1.2 \end{bmatrix}$ (pr{2})
- $\mu_3(i_3) = \begin{bmatrix} -2.0 & 1.2 \end{bmatrix}$ (pr{3})

The MATLAB mLF package

Detailed example 3-D (compute null space)

```
%%% Compute rec. LL null
[c,lag] = mlf.loewner_null_rec(pc,pr,tabr,'svd0');
```

- Recursive: 110 flop and 9.0 Bytes
- Full: 1728 flop and 144.0 Bytes
- $\mathbf{W}_{2,3,2}^{\otimes} = \begin{bmatrix} 0.342 & 0.133 & 0.484 \\ -0.0219 & -0.197 & 0.467 \end{bmatrix}, \begin{bmatrix} -0.196 & 0.084 & -0.548 \\ -0.394 & -0.252 & -1.67 \end{bmatrix}$
- $\mathbf{C}_{2,3,2}^{\otimes} = \begin{bmatrix} -0.69 & 1.69 & -1.0 \\ 1.0 & -1.48 & 0.484 \end{bmatrix}, \begin{bmatrix} 0.69 & -1.69 & 1.0 \\ -1.0 & 1.48 & -0.484 \end{bmatrix}$

The MATLAB mLF package

Detailed example 3-D (Lagrange form)

```
%%% Transfer function in Lagrangian form
[glag, ilag] = mlf.tf_lagrangian(pc,w,c,false);
```

$$\left(\begin{array}{c|c|c} \mathcal{B}_{\text{lag}}(x_1,x_2,x_3) & \mathbf{N}_{\text{lag}} & \mathbf{D}_{\text{lag}} \end{array} \right)$$

$$\left(\begin{array}{c|c|c} (x_1+1.2)\ (x_2+1.2)\ (x_3+1.2) & -0.236 & -0.69 \\ (x_3-2.0)\ (x_1+1.2)\ (x_2+1.2) & -0.135 & 0.69 \\ (x_1+1.2)\ (x_3+1.2)\ (x_2-0.4) & 0.225 & 1.69 \\ (x_3-2.0)\ (x_1+1.2)\ (x_2-0.4) & -0.142 & -1.69 \\ (x_2-2.0)\ (x_1+1.2)\ (x_3+1.2) & -0.484 & -1.0 \\ (x_2-2.0)\ (x_3-2.0)\ (x_1+1.2) & -0.548 & 1.0 \\ (x_1-2.0)\ (x_2+1.2)\ (x_3+1.2) & -0.0219 & 1.0 \\ (x_1-2.0)\ (x_3-2.0)\ (x_2+1.2) & 0.394 & -1.0 \\ (x_1-2.0)\ (x_3+1.2)\ (x_2-0.4) & 0.292 & -1.48 \\ (x_1-2.0)\ (x_3-2.0)\ (x_2-0.4) & -0.374 & 1.48 \\ (x_1-2.0)\ (x_2-2.0)\ (x_3+1.2) & 0.226 & 0.484 \\ (x_1-2.0)\ (x_2-2.0)\ (x_3-2.0) & 0.806 & -0.484 \end{array} \right)$$

$$\mathbf{G}_{\text{lag}}(x_1,x_2,x_3) = \frac{\sum_{j_1=1}^{2}\sum_{j_2=1}^{3}\sum_{j_3=1}^{2} \frac{\mathbf{C}^{\otimes}_{2,3,2}(j_1,j_2,j3)\mathbf{W}^{\otimes}_{2,3,2}(j_1,j_2,j_3)}{(x_1-\lambda_1(j_1))\cdots(x_3-\lambda_3(j_3))}}{\sum_{j_1=1}^{2}\sum_{j_2=1}^{3}\sum_{j_3=1}^{2} \frac{\mathbf{C}^{\otimes}_{2,3,2}(j_1,j_2,j_3)}{(x_1-\lambda_1(j_1))\cdots(x_3-\lambda_3(j_3))}},$$

The MATLAB mLF package

Detailed example 3-D (Monomial form)

```
%%% Transfer function in Monomial form
[gmon,imon] = mlf.tf_monomial(pc,w,c,false);
```

$$\left(\begin{array}{c|c|c} \mathcal{B}_{\mathrm{mon}}(x_1, x_2, x_3) & \mathbf{N}_{\mathrm{mon}} & \mathbf{D}_{\mathrm{mon}} \end{array} \right)$$

$$\left(\begin{array}{c|c|c} x_1 {x_2}^2 x_3 & 0 & 0 \\ x_1 {x_2}^2 & 0 & 0 \\ x_1 x_2 x_3 & 0 & 0 \\ x_1 x_2 & 0 & 0.1 \\ x_1 x_3 & 0 & 0 \\ x_1 & 0.1 & 0 \\ {x_2}^2 x_3 & 0.1 & 0 \\ {x_2}^2 & 0 & 0 \\ x_2 x_3 & 0 & 0 \\ x_2 & 0 & 0 \\ x_3 & 0 & 0 \\ 1.0 & 0 & -1.0 \end{array} \right)$$

$$\mathbf{G}_{\mathrm{mon}}(x_1, x_2, x_3) = \frac{\sum_{j1=1}^{2} \sum_{j2=1}^{3} \sum_{j3=1}^{2} \mathbf{N}^{\otimes}_{2,3,2}(j_1, j_2, j_3) \left(x_1^{(j_1-1)} \cdots x_3^{(j_3-1)} \right.}{\sum_{j1=1}^{2} \sum_{j2=1}^{3} \sum_{j3=1}^{2} \mathbf{D}^{\otimes}_{2,3,2}(j_1, j_2, j_3) \left(x_1^{(j_1-1)} \cdots x_3^{(j_3-1)} \right.}$$

The MATLAB mLF package

Detailed example 3-D (KST form)

```
%%% Decoupling toward KST
lag_struct.pc   = pc;
lag_struct.lag  = lag;
[Bary,Lag,Cx]   = mlf.decoupling(lag_struct);
```

$[\mathbf{Bary}(x_1), \mathbf{Bary}(x_2), \mathbf{Bary}(x_3)] =$

$$\begin{bmatrix}
1.0 & 0.69 & -1.0 \\
1.0 & 0.69 & 1.0 \\
1.0 & -1.69 & -1.0 \\
1.0 & -1.69 & 1.0 \\
1.0 & 1.0 & -1.0 \\
1.0 & 1.0 & 1.0 \\
-0.484 & 2.07 & -1.0 \\
-0.484 & 2.07 & 1.0 \\
-0.484 & -3.07 & -1.0 \\
-0.484 & -3.07 & 1.0 \\
-0.484 & 1.0 & -1.0 \\
-0.484 & 1.0 & 1.0
\end{bmatrix}$$

$[\mathbf{Lag}(x_1), \mathbf{Lag}(x_2), \mathbf{Lag}(x_3)] =$

$$\begin{bmatrix}
\frac{1}{x_1+1.2} & \frac{1}{x_2+1.2} & \frac{1}{x_3+1.2} \\
\frac{1}{x_1+1.2} & \frac{1}{x_2+1.2} & \frac{1}{x_3-2.0} \\
\frac{1}{x_1+1.2} & \frac{1}{x_2-0.4} & \frac{1}{x_3+1.2} \\
\frac{1}{x_1+1.2} & \frac{1}{x_2-0.4} & \frac{1}{x_3-2.0} \\
\frac{1}{x_1+1.2} & \frac{1}{x_2-2.0} & \frac{1}{x_3+1.2} \\
\frac{1}{x_1+1.2} & \frac{1}{x_2-2.0} & \frac{1}{x_3-2.0} \\
\frac{1}{x_1-2.0} & \frac{1}{x_2+1.2} & \frac{1}{x_3+1.2} \\
\frac{1}{x_1-2.0} & \frac{1}{x_2+1.2} & \frac{1}{x_3-2.0} \\
\frac{1}{x_1-2.0} & \frac{1}{x_2-0.4} & \frac{1}{x_3+1.2} \\
\frac{1}{x_1-2.0} & \frac{1}{x_2-0.4} & \frac{1}{x_3-2.0} \\
\frac{1}{x_1-2.0} & \frac{1}{x_2-2.0} & \frac{1}{x_3+1.2} \\
\frac{1}{x_1-2.0} & \frac{1}{x_2-2.0} & \frac{1}{x_3-2.0}
\end{bmatrix}$$

Univariate vector functions $\mathbf{\Phi}_1(x_1)$ $\ldots$ $\mathbf{\Phi}_3(x_3)$

$$\begin{cases}
\mathbf{Bary}(x_1) \odot \mathbf{Lag}(x_1) \\
\mathbf{Bary}(x_2) \odot \mathbf{Lag}(x_2) \\
\mathbf{Bary}(x_3) \odot \mathbf{Lag}(x_3)
\end{cases}$$

KST rational interpolant

$$\mathbf{H}_{\text{kst}}(x_1, x_2, x_3) = \frac{\mathbf{n}_{\text{kst}}(x_1, x_2, x_3)}{\mathbf{d}_{\text{kst}}(x_1, x_2, x_3)}$$

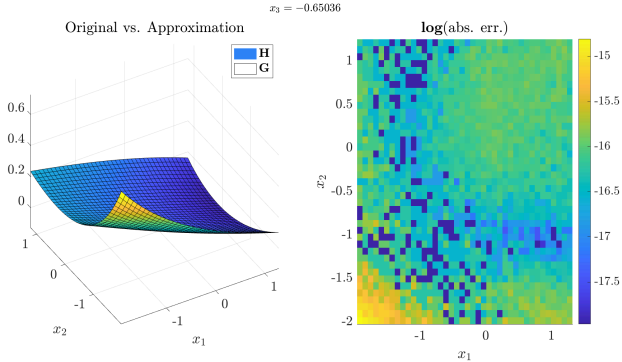
where numerator and denominator are

$$\begin{cases}
\sum_{\text{rows}} \mathbf{w} \odot \mathbf{\Phi}_1(x_1) \odot \mathbf{\Phi}_2(x_2) \odot \mathbf{\Phi}_2(x_3 \\
\sum_{\text{rows}} \mathbf{\Phi}_1(x_1) \odot \mathbf{\Phi}_2(x_2) \odot \mathbf{\Phi}_2(x_3)
\end{cases}$$

The MATLAB mLF package

Detailed example 3-D (KST form)

$$\mathbf{H}(x_1, x_2) = \frac{x_3\, {x_2}^2 + x_1}{x_1\, x_2 - 10}$$



Content

Forewords

Multi-variate data, function & Loewner matrix

Taming the curse of dimensionnality

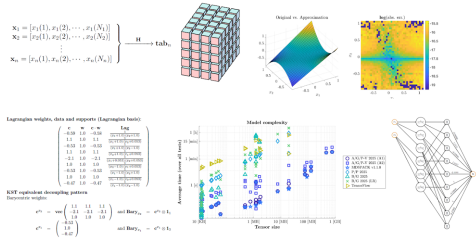
Variables decoupling, KST and KANs

Comparisons

Multi-variate realization

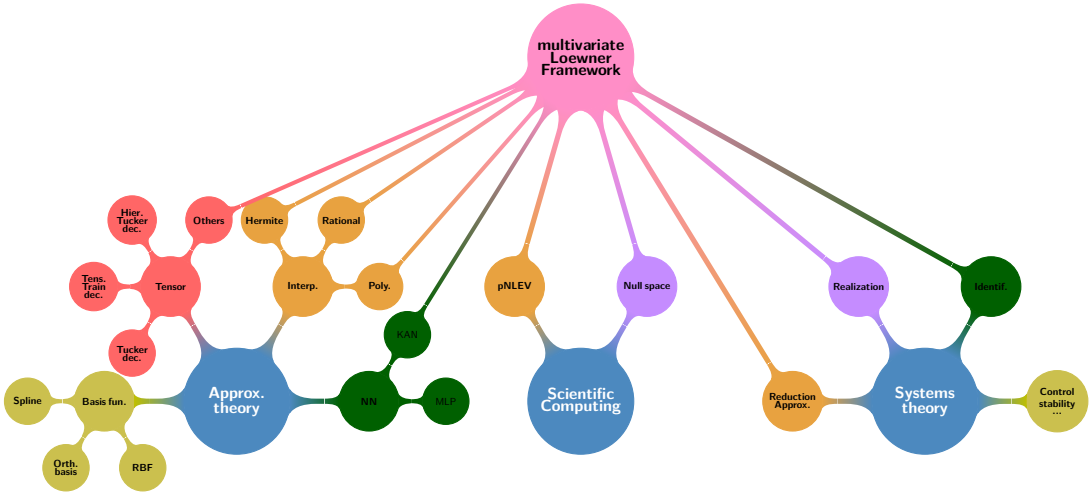
The MATLAB mLF package

Conclusion



Conclusion

Multivariate Loewner



Conclusion

Take home message

Main contributions (from n-D tensor)

- Transfer function in barycentric form
- Realization with controlled complexity
- Tame the computational complexity
- Two algorithms (direct & iterative)
- Solution of rational KST
- Connection with KAN

Side effects

[Sci. con.] Tensor approximation
[Sci. con.] pNEVP multi-linearization
[Sci. con.] Exact (Loewner) matrix null space
[Dyn. sys.] Parametric realization
[Dyn. sys.] LFR connections
[Dyn. sys.] Uncertainty propagation

Collaboration with
A.C. Antoulas [Rice Univ.]
I.V. Goșea [MPI]
P. Vuillemin [ONERA]
`https://cpoussot.github.io`

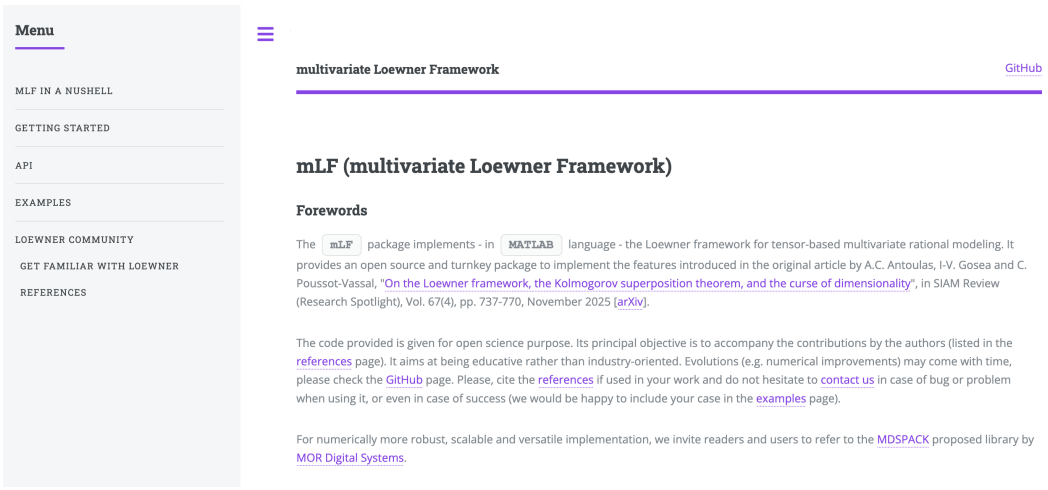
Theory (paper & code)
`https://doi.org/10.1137/24M1656657`
`https://cpoussot.github.io/mlf/`

Benchmark (paper & code)
`https://arxiv.org/abs/2506.04791`
`https://github.com/cpoussot/benchmark_tensor`



Conclusion

https://cpoussot.github.io/mlf/



Conclusion

In parting... if enough time, 20-D example

The unknown

$$\mathbf{H}(x_1, x_2, \ldots, x_{20}) = \frac{3 \cdot x_1^3 + 4 \cdot x_8 + x_{12} + x_{13} \cdot x_{14} + x_{15}}{x_1 + x_2^2 \cdot x_3 + x_4 + x_5 + x_6 + x_7 \cdot x_8 + x_9 \cdot x_{10} \cdot x_{11} + x_{13} + x_{13}^3 \cdot \pi + x_{17} + x_{18} \cdot x_{19} - x_{20}}$$

Statistics

- 20-D tensor of dimension (≥ 48 TBytes in real double precision)
- Complexity: $(3, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 1, 1, 1, 1)$
- Memory:
  Full: $6,291,456^2 \rightarrow 288$ TBytes in real double precision
  Recursive: $4^2 \rightarrow 0.12$ KBytes in real double precision
- Computation load:
  Full : $2.49 \cdot 10^{20}$ `flop`
  Recursive: $5.43 \cdot 10^7$ `flop`
- error $\approx 10^{-11}$

Conclusion

In parting... if enough time, 20-D example

The unknown

$$\mathbf{H}(x_1, x_2, \ldots, x_{20}) = \frac{3 \cdot x_1^3 + 4 \cdot x_8 + x_{12} + x_{13} \cdot x_{14} + x_{15}}{x_1 + x_2^2 \cdot x_3 + x_4 + x_5 + x_6 + x_7 \cdot x_8 + x_9 \cdot x_{10} \cdot x_{11} + x_{13} + x_{13}^3 \cdot \pi + x_{17} + x_{18} \cdot x_{19} - x_{20}}$$

Statistics

- 20-D tensor of dimension (≥ 48 TBytes in real double precision)
- Complexity: $(\mathbf{3}, \mathbf{3}, \mathbf{2}, \mathbf{1}, \mathbf{1}, \mathbf{1}, \mathbf{1}, \mathbf{1}, \mathbf{1}, \mathbf{1}, \mathbf{1}, \mathbf{1}, \mathbf{1}, \mathbf{1}, \mathbf{1}, \mathbf{1}, \mathbf{1}, \mathbf{1}, \mathbf{1}, \mathbf{1})$
- Memory:
  Full: $6,291,456^2 \rightarrow 288$ TBytes in real double precision
  Recursive: $4^2 \rightarrow 0.12$ KBytes in real double precision
- Computation load:
  Full : $2.49 \cdot 10^{20}$ `flop`
  Recursive: $\mathbf{5.03 \cdot 10^7}$ `flop`
- error $\approx 10^{-11}$

The multivariate Loewner framework

The Kolmogorov superposition theorem, the curse of dimensionality, tensors & more

C. Poussot-Vassal, in coll. with A.C. Antoulas [Rice Univ.], I.V. Gosea [MPI] and P. Vuillemin [ONERA]
November 24, 2026

`https://doi.org/10.1137/24M1656657` [in SIAM Review - Research Spotlight]
`https://arxiv.org/abs/2506.04791` [extensive benchmark]
`https://cpoussot.github.io/mlf/` [mLF code & documentation]

