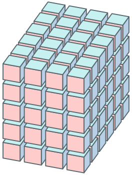
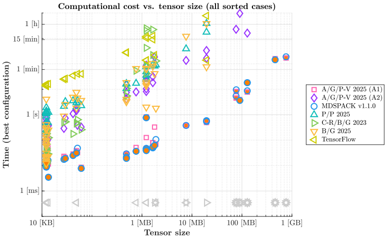
The Loewner framework, in the eye of the tensor...

The Kolmogorov superposition theorem, the curse of dimensionality & benchmark

C. Poussot-Vassal, in coll. with A.C. Antoulas [Rice Univ.], I.V. Goșea [MPI] and P. Vuillemin [ONERA]
February 17, 2026

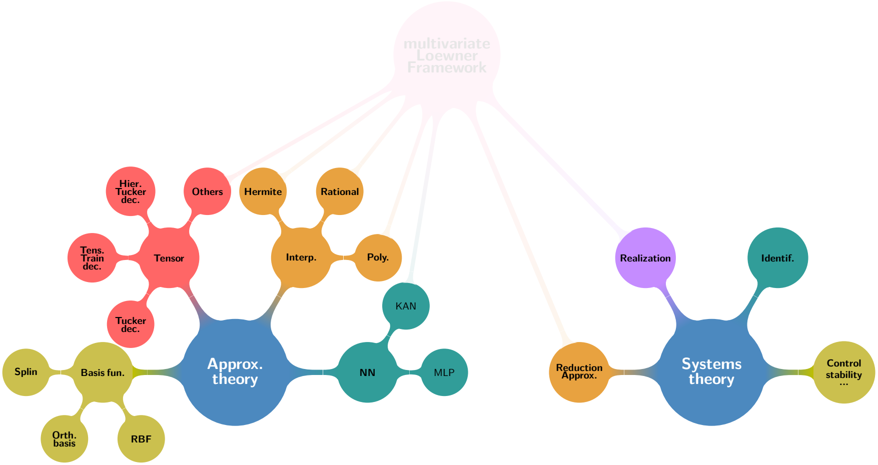
https://doi.org/10.1137/24M1656657 [in SIAM Review - Research Spotlight]
https://arxiv.org/abs/2506.04791 [extensive benchmark]





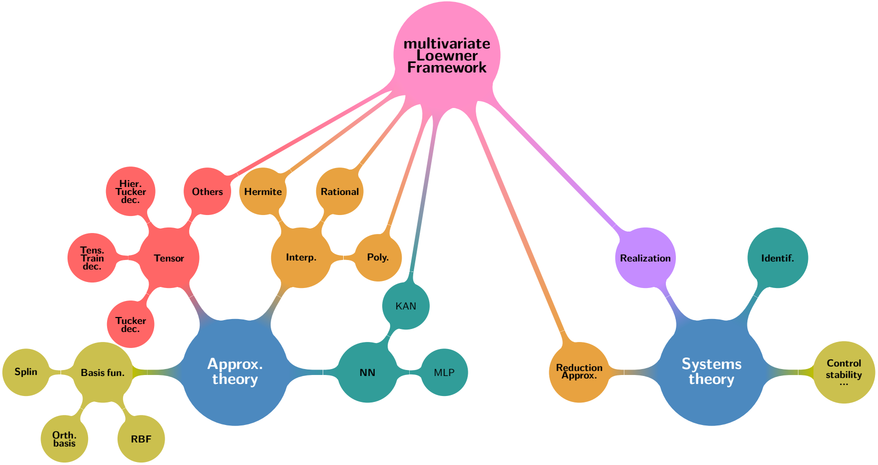
Forewords

Approximation & systems theory... where we stand

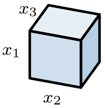


Forewords

Approximation & systems theory... where we stand



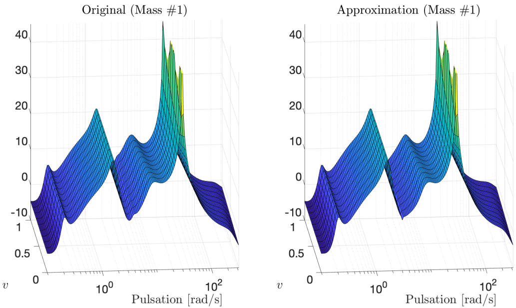
$$\mathbf{\Sigma}(x_1,x_2,x_3)=\mathbf{\Sigma}(s,m,v):s^2M(m)\mathbf{x}(s)+sB(m)\mathbf{x}(s)+K(m)\mathbf{x}(s)-G(s,v)=\mathbf{u}(s),\ \mathbf{y}(s)=C\mathbf{x}(s)$$



$$\begin{array}{ccccc} x_1 & \times & x_2 & \times & x_3 \\ \imath[10,35] & \times & [\underline{m},\overline{m}] & \times & [\underline{v},\overline{v}] \end{array}$$

$$\mathcal{T}_3^{\otimes}\in\mathbb{C}^{300\times10\times10}$$

$\approx$`468.75 Ko ('complex')`

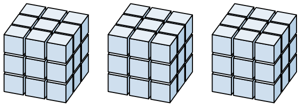


A. dos Reis de Souza et al., *"Aircraft flutter suppression: from a parametric model to robust control"*, ECC, 2023.

Forewords

Starting (motivating) examples - static function #30 (Borehole)

$$\mathbf{H}(x_1,\cdots,x_8)=\mathbf{H}(r_w,r,T_u,H_u,T_l,H_l,L,K_w)=\frac{2\pi T_u(H_u-H_l)}{\ln\left(\frac{r}{r_w}\right)\left(1+\frac{2LT_u}{\ln(r/r_w)r_w^2K_w}\right)+\frac{T_u}{T_l}}$$



$$\begin{array}{ccccc} x_1 & \times & \cdots & \times & x_8 \\ [\underline{r_w},\overline{r_w}] & \times & \cdots & \times & [\underline{K_w},\overline{K_w}] \end{array}$$

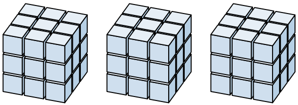
$$\mathcal{T}_8^{\otimes}\in\mathbb{C}^{8\times 8\times\cdots\times 8}$$

≈130 Mo ('real')

$r_w \in [0.05, 0.15]$	radius of borehole (m)
$r \in [100, 50\,000]$	radius of influence (m)
$T_u \in [63\,070, 115\,600]$	transmissivity of upper aquifer (m^2/yr)
$H_u \in [990, 1110]$	potentiometric head of upper aquifer (m)
$T_l \in [63.1, 116]$	transmissivity of lower aquifer (m^2/yr)
$H_l \in [700, 820]$	potentiometric head of lower aquifer (m)
$L \in [1120, 1680]$	length of borehole (m)
$K_w \in [9855, 12\,045]$	hydraulic conductivity of borehole (m/yr)

S. Surjanovic, *"Borehole function"*, `https://www.sfu.ca/~ssurjano/borehole.html`.

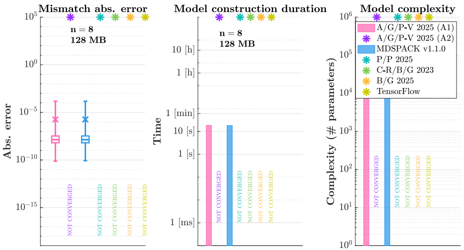
$$\mathbf{H}(x_1,\cdots,x_8)=\mathbf{H}(r_w,r,T_u,H_u,T_l,H_l,L,K_w)=\frac{2\pi T_u(H_u-H_l)}{\ln\left(\frac{r}{r_w}\right)\left(1+\frac{2LT_u}{\ln(r/r_w)r_w^2K_w}\right)+\frac{T_u}{T_l}}$$



$$\begin{array}{ccccc} x_1 & \times & \cdots & \times & x_8 \\ [\underline{r_w},\overline{r_w}] & \times & \cdots & \times & [\underline{K_w},\overline{K_w}] \end{array}$$

$$\mathcal{T}_8^{\otimes}\in\mathbb{C}^{8\times 8\times\cdots\times 8}$$

≈130 Mo ('real')



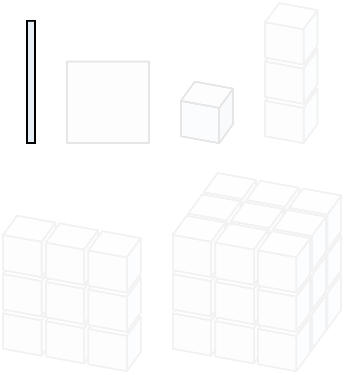
S. Surjanovic, *"Borehole function"*, `https://www.sfu.ca/~ssurjano/borehole.html`.

Column / Row data

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

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

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



Forewords

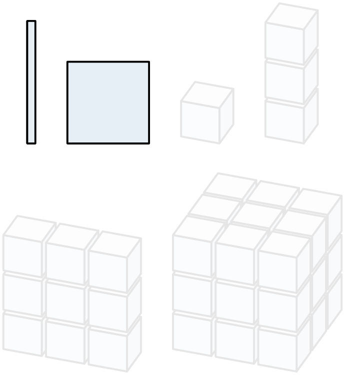
Data (and tensors)

Column / Row data

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

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

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

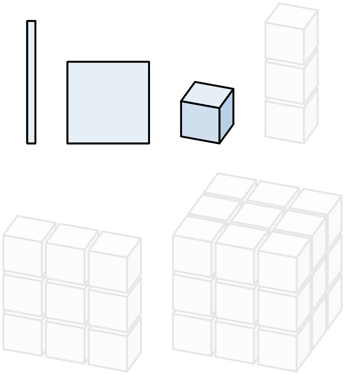
$x_3 = \lambda_3(1,\cdots,k_3)$

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) $\mathcal{T}_3^{\otimes}$



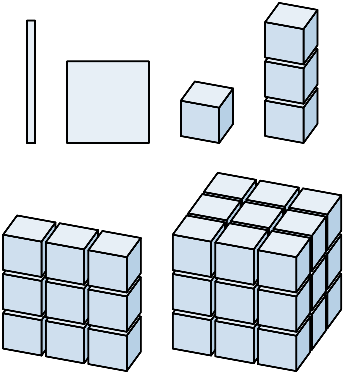
Forewords

Data (and tensors)

Column / Row data

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

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

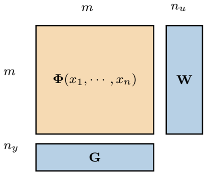


Data-driven model approximation

Being given a n-dimensional tensor (data), we seek a multi-variate **rational function** $\hat{\mathbf{H}}$ and **realization** $(\mathbf{G}, \mathbf{\Phi}, \mathbf{W})$

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

interpolating the data.



Connection to standard dynamical system realization

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

$$\begin{array}{rcl} \mathbf{E}(\mathcal{S})\dot{\mathbf{x}}(t;\mathcal{S}) & = & \mathbf{A}(\mathcal{S})\mathbf{x}(t;\mathcal{S}) + \mathbf{B}\mathbf{u}(t) \\ \mathbf{y}(t;\mathcal{S}) & = & \mathbf{C}\mathbf{x}(t;\mathcal{S}) \end{array}$$

equivalently

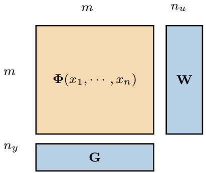
$$\hat{\mathbf{H}}(x_1, x_2, \ldots, x_n) = \mathbf{C}(\mathcal{S})\left[x_1\mathbf{E}(\mathcal{S}) - \mathbf{A}(\mathcal{S})\right]^{-1}\mathbf{B} \in \mathbb{C}.$$

Data-driven model approximation

Being given a n-dimensional tensor (data), we seek a multi-variate **rational function** $\hat{\mathbf{H}}$ and **realization** $(\mathbf{G}, \mathbf{\Phi}, \mathbf{W})$

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

interpolating the data.



Connection to standard dynamical system realization

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

$$\begin{array}{rcl} \mathbf{E}(\mathcal{S})\dot{\mathbf{x}}(t;\mathcal{S}) & = & \mathbf{A}(\mathcal{S})\mathbf{x}(t;\mathcal{S}) + \mathbf{B}\mathbf{u}(t) \\ \mathbf{y}(t;\mathcal{S}) & = & \mathbf{C}\mathbf{x}(t;\mathcal{S}) \end{array}$$

equivalently

$$\hat{\mathbf{H}}(x_1, x_2, \ldots, x_n) = \mathbf{C}(\mathcal{S})\left[x_1\mathbf{E}(\mathcal{S}) - \mathbf{A}(\mathcal{S})\right]^{-1}\mathbf{B} \in \mathbb{C}.$$

Forewords

Where we stand (some references)

1-D Two-sided Loewner
$\Rightarrow$ (interpolation) barycentric form
$\Rightarrow$ realization minimality
$\Rightarrow$ direct algorithm

1-D One-sided Loewner
$\Rightarrow$ (interpolation) barycentric form
$\Rightarrow$ direct algorithm

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

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

3-D Parametric AAA
$\Rightarrow$ (mixed interpolation LS) barycentric form
$\Rightarrow$ 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
- n-variable generalized realization
- Taming the curse of dimensionality (`flop`/`Bytes`/accuracy)
- n-variable **decoupling**
- **KST** formulation for rational functions
- Comparison with **MLP, KAN, pAAA**

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, November, 2025 (`https://arxiv.org/abs/2405.00495`).

A.C. Antoulas, I-V. Gosea, C. P-V. and P. Vuillemin, *"Tensor-based multivariate function approximation: methods benchmarking and comparison"*, arXiv, June, 2025 (`https://arxiv.org/abs/2506.04791`).

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

Content

Multi-variate data, function & Loewner matrix

1-D case

$$\left\{\begin{array}{lll} P_c^{(1)} & := & \{(\lambda_1(j_1); \mathbf{w}_{j_1}),\ j_1 = 1, \dots, k_1\} \\ P_r^{(1)} & := & \{(\mu_1(i_1); \mathbf{v}_{i_1}),\ i_1 = 1, \dots, q_1\} \end{array}\right.$$

Loewner matrix

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

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

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

Lagrangian form

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

Null space

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

$$\mathbf{c}_1 = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_{k_1} \end{bmatrix} \in \mathbb{C}^{k_1}$$

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

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

Loewner matrix

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

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

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

Lagrangian form

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

Null space

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

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

Multi-variate data, function & Loewner matrix

2-D case (example)

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

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

Loewner matrix

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

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

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

Lagrangian form

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

Null space

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

$$\mathbf{c}_n = \left[\begin{array}{c} c_{1,\cdots,1} \\ \vdots \\ c_{1,\cdots,k_n} \\ \hline \vdots \\ \hline c_{k_1,\cdots,1} \\ \vdots \\ c_{k_1,\cdots,k_n} \end{array}\right] \in \mathbb{C}^{k_1 \cdots k_n}$$

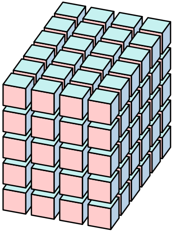
Content

Taming the curse of dimensionality

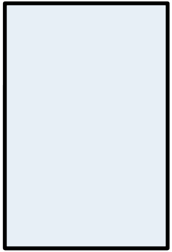
n-variable Loewner matrix operator[1]

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

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



matrix $\mathbb{L}_n$

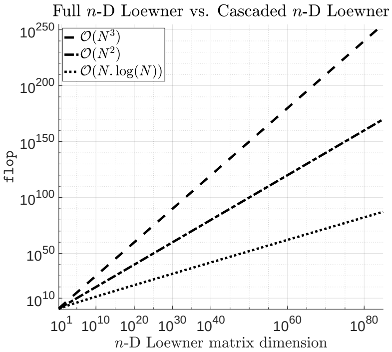


[1] where $Q = q_1 q_2 \cdots q_n$ and $K = k_1 k_2 \cdots k_n$.

Taming the curse of dimensionality

Null space flop and memory issues

log-log scale



(rows) $Q = q_1 q_2 \dots q_n$ and
(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 $\mathbf{H}(x_1,x_2)=\mathbf{H}(s,t)=(s^2t)/(s-t+1)$ of complexity $(2,1)$

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

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

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

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 $(2,1)$

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

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

- 1 $\mathbb{L}_1$ along x_1, for $x_2 = \lambda_2(2) = -3$

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

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

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

Taming the curse of dimensionality

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

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

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

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

- 1 $\mathbb{L}_1$ along x_1, for $x_2=\lambda_2(2)=-3$
- 3 $\mathbb{L}_1$ along x_2 for $x_1=\{\lambda_1(1),\lambda_1(2),\lambda_1(3)\}$

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

- Scaled null space $\mathbf{c}_2^\top=$

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

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 $(2,1)$

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

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

- 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 $\mathbf{c}_2^\top =$
  $\left[\ \mathbf{c}_1^{\lambda_1(1)\top} \cdot [\mathbf{c}_1^{\lambda_2(2)}]_1 \quad \mathbf{c}_1^{\lambda_1(2)\top} \cdot [\mathbf{c}_1^{\lambda_2(2)}]_2 \quad \mathbf{c}_1^{\lambda_1(3)\top} \cdot [\mathbf{c}_1^{\lambda_2(2)}]_3\ \right]^\top$

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

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 Loenwer matrix $\mathbb{L}_2$, is spanned by

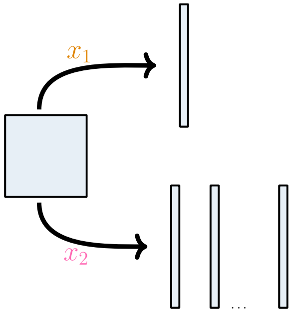
$$\mathbf{c}_2 = \mathcal{N}(\mathbb{L}_2) = \mathbf{vec}\ \left[\mathbf{c}_1^{\lambda_2(1)} \cdot \left[\mathbf{c}_1^{\lambda_1(k_1)} \right]_1, \cdots, \mathbf{c}_1^{\lambda_2(k_2)} \cdot \left[\mathbf{c}_1^{\lambda_1(k_1)} \right]_{k_2} \right],$$

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 $x_2 = \{\lambda_2(1), \cdots, \lambda_2(k_2)\}$.

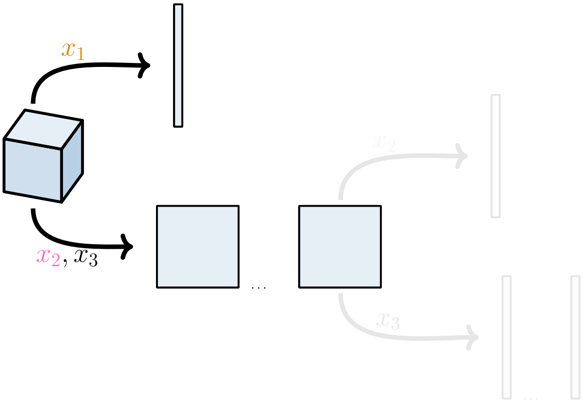
Taming the curse of dimensionality

2-D case

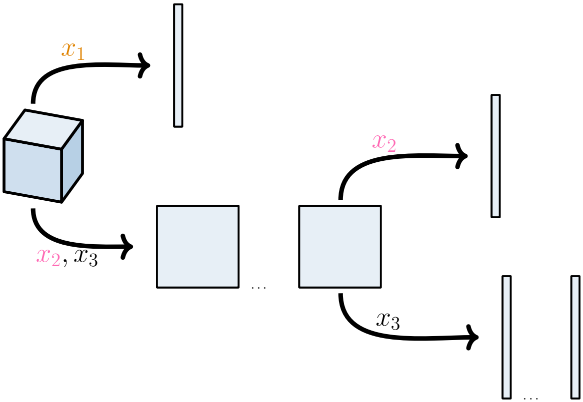


Taming the curse of dimensionality

3-D case



Taming the curse of dimensionality
3-D case



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 of the corresponding n-D Loenwer matrix $\mathbb{L}_n$, is spanned by

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

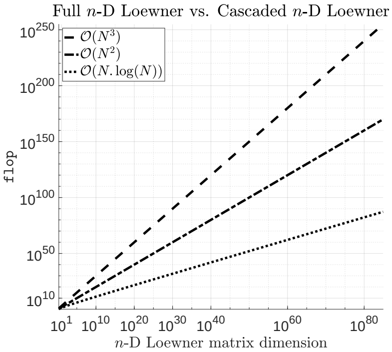
where

- $\mathbf{c}_1^{(\lambda_2(k_2), \lambda_3(k_3), \cdots, \lambda_n(k_n))}$ 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

Null space - `flop` complexity

log-log scale



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

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

Computational issue

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

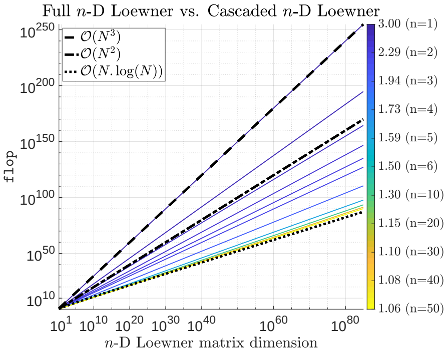
- QK^2 (if $Q > K$)
- N^3 (if $Q = K = N$)

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

Taming the curse of dimensionality

Null space - `flop` complexity

log-log scale



Theorem: Recursive complexity

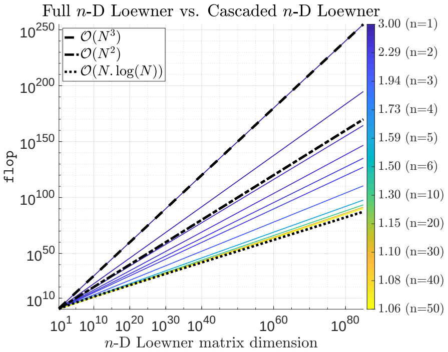
$$\texttt{flop}_1(n) = \sum_{j=1}^{n} \left(k_j^3 \prod_{l=1}^{j} k_{l-1} \right) \text{ where } k_0 = 1.$$

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

Taming the curse of dimensionality

Null space - `flop` complexity

log-log scale



Theorem: Worst case complexity

k interpolation points per variables.

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

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

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

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

Taming the curse of dimensionality

Null space - memory and storage

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

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

Full n-D Loewner

Construction of

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

where $N = k_1k_2\cdots k_n$, needs

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

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

Taming the curse of dimensionality

Null space - memory and storage

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

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

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

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

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

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

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

Content

Kolmogorov, Arnold, Kahane, Lorentz and Sprecher

For every continuous function $\mathbf{f} : \mathbb{I}^n \mapsto \mathbb{R}$ and any $n \in \mathbb{N}$, $n \geq 2$, there exist

- real numbers $\lambda_1, \ldots, \lambda_n$;
- continuous functions $\phi_k : \mathbb{I} \mapsto \mathbb{R}$, $k = 1, \ldots, 2n+1$;
- a continuous function $\mathbf{\Phi} : \mathbb{R} \mapsto \mathbb{R}$;

such that $\forall (x_1, \cdots, x_n) \in \mathbb{I}^n$:

$$\mathbf{f}(x_1, \cdots, x_n) = \sum_{q=1}^{2n+1} \mathbf{\Phi} \left(\sum_{p=1}^{n} \phi_{q,p}(x_p) \right) = \sum_{k=1}^{2n+1} \mathbf{\Phi}(\lambda_1 \phi_k(x_1) + \cdots + \lambda_n \phi_k(x_n))$$

"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, pp. 11-25.

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 ($\mathbf{H} = x_1 \cdot x_2$)

$$
\begin{array}{rcl}
\lambda_1(j_1) & = & \begin{pmatrix} -1 & 1 \end{pmatrix} \\
\lambda_2(j_2) & = & \begin{pmatrix} -1 & 1 \end{pmatrix} \\
\mu_1(i_1) & = & \lambda_1(j_1) + 0.5 \\
\mu_2(i_2) & = & \lambda_2(j_2) + 0.5
\end{array}
$$

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

$$
\begin{pmatrix}
\mathbf{c} & \mathbf{w} & \mathbf{c} \odot \mathbf{w} & \mathbf{Lag} \\
1.0 & 1.0 & 1.0 & \frac{1}{(x_1+1.0)(x_2+1.0)} \\
-1.0 & -1.0 & 1.0 & \frac{1}{(x_1+1.0)(x_2-1.0)} \\
-1.0 & -1.0 & 1.0 & \frac{1}{(x_1-1.0)(x_2+1.0)} \\
1.0 & 1.0 & 1.0 & \frac{1}{(x_1-1.0)(x_2-1.0)}
\end{pmatrix}
$$

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

where $\mathbf{C}_{2,2}^{\otimes}$ and $\mathbf{W}_{2,2}^{\otimes}$ are the tensorized forms of $\mathbf{c}$ and $\mathbf{w}$.

Variables decoupling, KST and KANs

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

$$
\begin{aligned}
\lambda_1(j_1) &= \begin{pmatrix} -1 & 1 \end{pmatrix}\\
\lambda_2(j_2) &= \begin{pmatrix} -1 & 1 \end{pmatrix}\\
\mu_1(i_1) &= \lambda_1(j_1) + 0.5\\
\mu_2(i_2) &= \lambda_2(j_2) + 0.5
\end{aligned}
$$

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

$$
\begin{pmatrix}
\mathbf{c} & \mathbf{w} & \mathbf{c} \odot \mathbf{w} & \mathbf{Lag}\\
1.0 & 1.0 & 1.0 & \frac{1}{(x_1+1.0)(x_2+1.0)}\\
-1.0 & -1.0 & 1.0 & \frac{1}{(x_1+1.0)(x_2-1.0)}\\
-1.0 & -1.0 & 1.0 & \frac{1}{(x_1-1.0)(x_2+1.0)}\\
1.0 & 1.0 & 1.0 & \frac{1}{(x_1-1.0)(x_2-1.0)}
\end{pmatrix}
$$

Decoupling theorem

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

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

Variables decoupling, KST and KANs

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

$$
\begin{aligned}
\lambda_1(j_1) &= \begin{pmatrix} -1 & 1 \end{pmatrix}\\
\lambda_2(j_2) &= \begin{pmatrix} -1 & 1 \end{pmatrix}\\
\mu_1(i_1) &= \lambda_1(j_1) + 0.5\\
\mu_2(i_2) &= \lambda_2(j_2) + 0.5
\end{aligned}
$$

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

$$
\begin{pmatrix}
\mathbf{c} & \mathbf{w} & \mathbf{c} \odot \mathbf{w} & \textbf{Lag} \\
1.0 & 1.0 & 1.0 & \frac{1}{(x_1+1.0)(x_2+1.0)} \\
-1.0 & -1.0 & 1.0 & \frac{1}{(x_1+1.0)(x_2-1.0)} \\
-1.0 & -1.0 & 1.0 & \frac{1}{(x_1-1.0)(x_2+1.0)} \\
1.0 & 1.0 & 1.0 & \frac{1}{(x_1-1.0)(x_2-1.0)}
\end{pmatrix}
$$

Denominator / Numerator

$$
\mathbf{D} = \begin{pmatrix}
\overbrace{\textbf{Bary}(x_1) \cdot \textbf{Lag}(x_1)}^{\Phi_1(x_1)} & \overbrace{\textbf{Bary}(x_2) \cdot \textbf{Lag}(x_2)}^{\Phi_2(x_2)} \\
-\frac{1.0}{x_1+1.0} & -\frac{1.0}{x_2+1.0} \\
-\frac{1.0}{x_1+1.0} & \frac{1}{x_2-1.0} \\
\frac{1}{x_1-1.0} & -\frac{1.0}{x_2+1.0} \\
\frac{1}{x_1-1.0} & \frac{1}{x_2-1.0}
\end{pmatrix}
$$

Equivalent denominator and numerator read:

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

Variables decoupling, KST and KANs

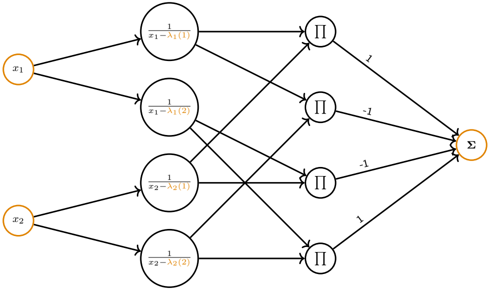
Decoupling, KST and KANs via Loewner with rational activation functions ($\mathbf{H} = x_1 \cdot x_2$)

$$
\begin{aligned}
\lambda_1(j_1) &= \begin{pmatrix} -1 & 1 \end{pmatrix} \\
\lambda_2(j_2) &= \begin{pmatrix} -1 & 1 \end{pmatrix} \\
\mu_1(i_1) &= \lambda_1(j_1) + 0.5 \\
\mu_2(i_2) &= \lambda_2(j_2) + 0.5
\end{aligned}
$$

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

$$
\begin{pmatrix}
\mathbf{c} & \mathbf{w} & \mathbf{c} \odot \mathbf{w} & \mathbf{Lag} \\
1.0 & 1.0 & 1.0 & \frac{1}{(x_1+1.0)(x_2+1.0)} \\
-1.0 & -1.0 & 1.0 & \frac{1}{(x_1+1.0)(x_2-1.0)} \\
-1.0 & -1.0 & 1.0 & \frac{1}{(x_1-1.0)(x_2+1.0)} \\
1.0 & 1.0 & 1.0 & \frac{1}{(x_1-1.0)(x_2-1.0)}
\end{pmatrix}
$$

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

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

$$\Updownarrow$$

$$\begin{aligned}
\mathbf{H}_{\text{kst}}(x_1, x_2) &= \frac{\sum_{\text{rows}} \mathbf{w} \odot \Phi_1(x_1) \odot \Phi_2(x_2)}{\sum_{\text{rows}} \Phi_1(x_1) \odot \Phi_2(x_2)} \\
&= \frac{\sum_{\text{rows}} \exp\left(\log(\mathbf{w}) + \log\left(\Phi_1(x_1)\right) + \log\left(\Phi_2(x_2)\right)\right)}{\sum_{\text{rows}} \exp\left(\log\left(\Phi_1(x_1)\right) + \log\left(\Phi_2(x_2)\right)\right)}
\end{aligned}$$

This is the solution of KST for rational forms !

Content

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
- Multi Layer Perceptron
- Dense connected / ReLU / ADAM / 1000 it. / rand. init.

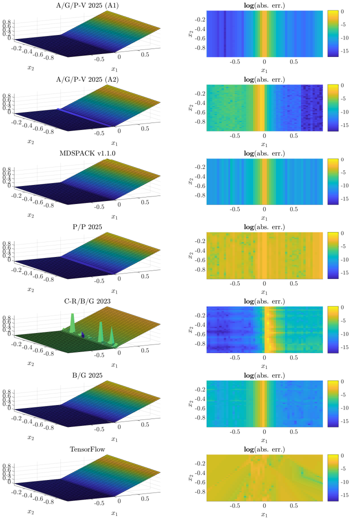
L. Balicki and S. Gugercin, *"Multivariate Rational Approximation via Low-Rank Tensors and the p-AAA Algorithm"*, SISC, 2025.

M. Poluektov and A. Polar, *"Construction of the Kolmogorov-Arnold representation using the Newton-Kaczmarz method"*, `https://arxiv.org/abs/2305.08194`.

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

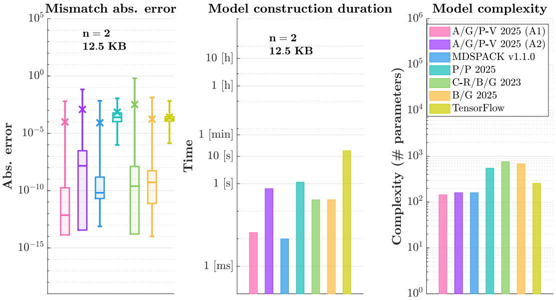
Comparisons

Function #1



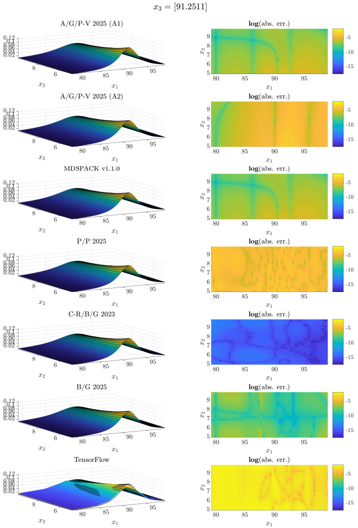
$$\mathrm{ReLU}(x_1) + \frac{1}{100} x_2$$

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



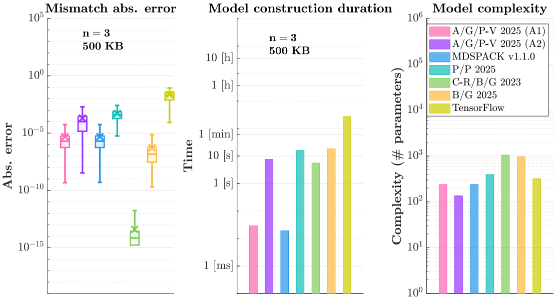
Comparisons

Function #20

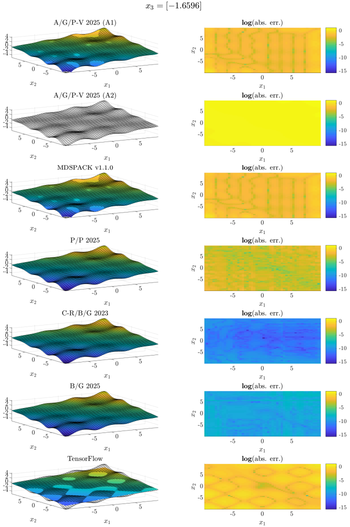


Breit Wigner function

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

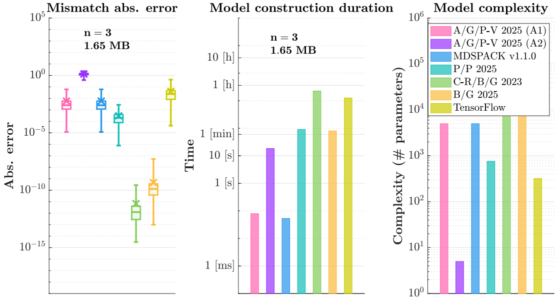


Comparisons
Function #26

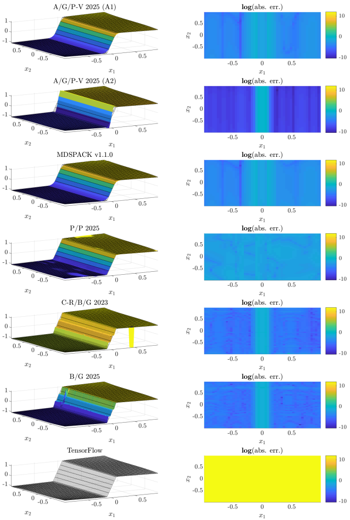


$$\frac{x_1+x_2+x_3}{6+\cos(x_1)+\cos(x_2)+\cos(x_3)}$$

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

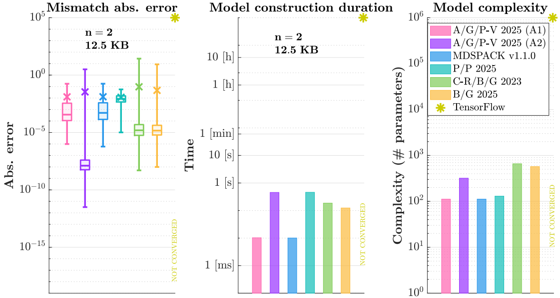


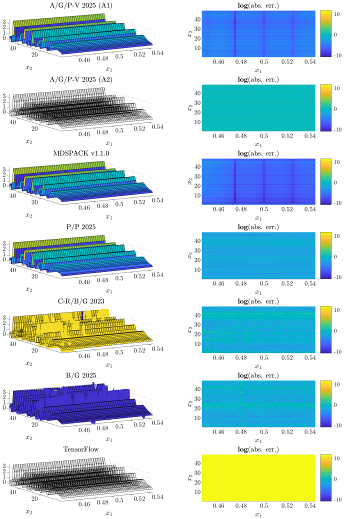
Comparisons
Function #29



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

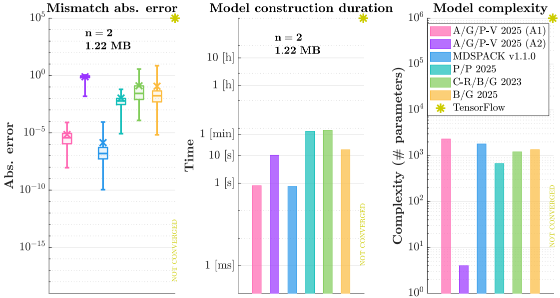
- Reference: Personal communication
- Tensor size: **12.5 KB**





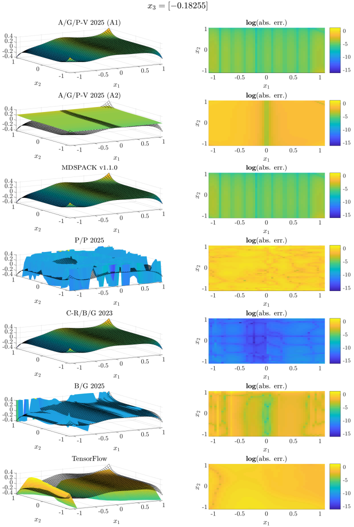
Riemann ζ function (real part)

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



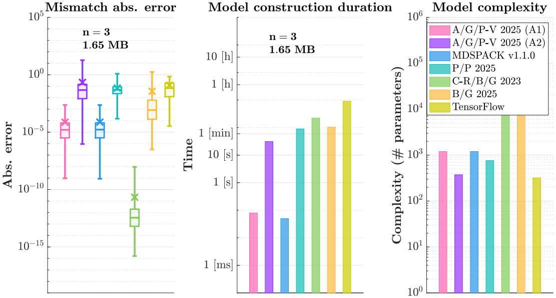
Comparisons

Function #38



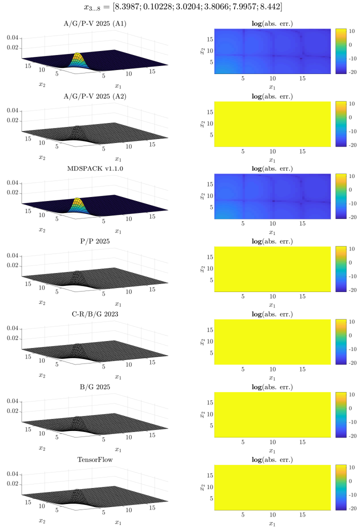
$$\frac{x_1^9 x_2^7 + x_1^3 + 5x_3^2}{5x_1^4 + 4x_1^2 + x_3 x_2^3 + 1}$$

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



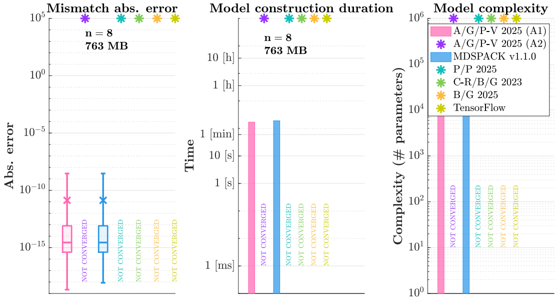
Comparisons

Function #44

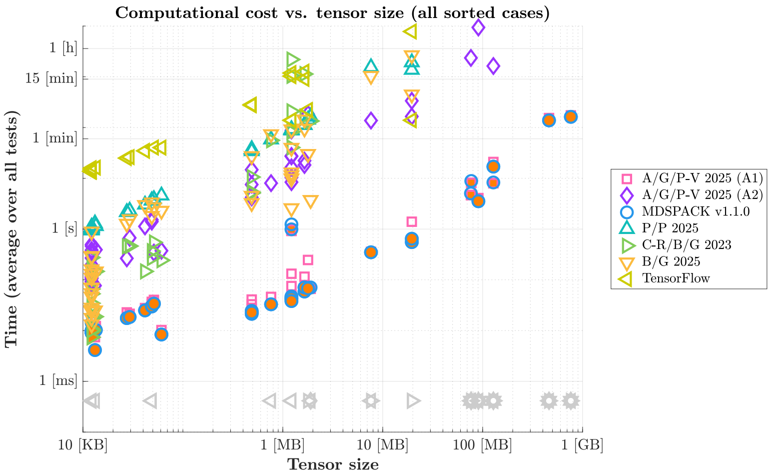


$$\frac{1}{x_1^4+x_2^2x_3+x_4^2+x_5+x_6+x_7+x_8}$$

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

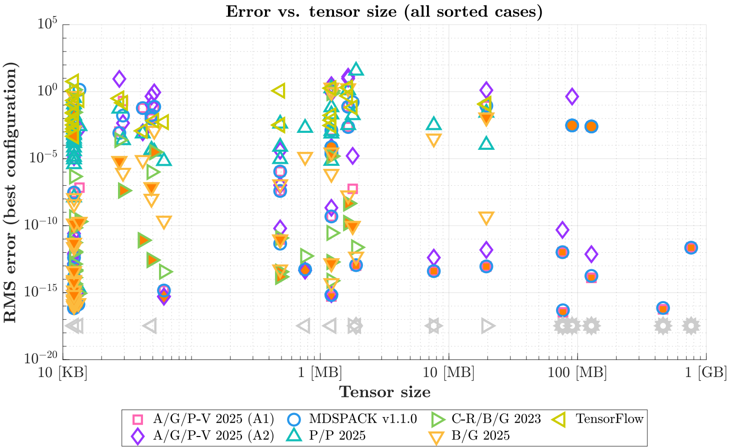


Comparisons
Size vs. computation time



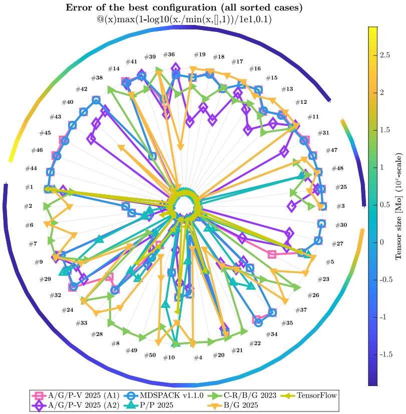
Comparisons

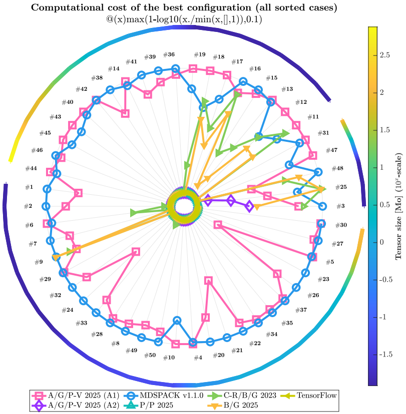
Size vs. error



Comparisons

Radar error & computational time





Content

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

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

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

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

Data generated from $\mathbf{H}(x_1,x_2) = \mathbf{H}(s,t) = (s^2t)/(s-t+1)$ 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 $\hat{\mathbf{H}}(s,t) = \mathbf{W}\mathbf{\Phi}(s,t)^{-1}\mathbf{G}$

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

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

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

$\rightarrow$ (3,3) block is unimodular !

Multi-variate realization

2-D case (example cont'd)

Data generated from $\mathbf{H}(x_1,x_2)=\mathbf{H}(s,t)=(s^2t)/(s-t+1)$ 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 $\hat{\mathbf{H}}_{\mathsf{c}}(s,t)=\mathbf{W}_{\mathsf{c}}(t)\mathbf{\Phi}_{\mathsf{c}}(s,t)^{-1}\mathbf{G}_{\mathsf{c}}$

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

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

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

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,

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

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 Lagrangian realization (focus on left / right variable sets)

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

$$\begin{aligned}
\boldsymbol{\Gamma} &= {\mathbb{X}^{\mathrm{Lag}}}_1 \otimes {\mathbb{X}^{\mathrm{Lag}}}_2 \otimes \cdots \otimes {\mathbb{X}^{\mathrm{Lag}}}_k \in \mathbb{C}^{\kappa\times\kappa}[x_1,\cdots,x_k] \\
\boldsymbol{\Delta} &= {\mathbb{X}^{\mathrm{Lag}}}_{k+1} \otimes {\mathbb{X}^{\mathrm{Lag}}}_{k+2} \otimes \cdots \otimes {\mathbb{X}^{\mathrm{Lag}}}_n \in \mathbb{C}^{\ell\times\ell}[x_{k+1},\cdots,x_n] \\
{\mathbb{X}^{\mathrm{Lag}}}_j &= \left[\begin{array}{ccccc} \mathbf{x}_{1j} & -\mathbf{x}_{2j} & 0 & \cdots & 0 \\ \mathbf{x}_{1j} & 0 & -\mathbf{x}_{3j} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{x}_{1j} & 0 & \cdots & 0 & -\mathbf{x}_{n_j j} \\ \hline q_{1j} & q_{2j} & \cdots & q_{n_j-1\,j} & q_{n_j j} \end{array}\right] \in \mathbb{C}^{n_j\times n_j} \\
\mathbf{x}_{ij} &= x_j - \lambda_j(i)
\end{aligned}$$

Facts

- Left / right variables splitting

$$\Gamma \text{ and } \Delta$$

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

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

- ...so are Γ and Δ

Multi-variate realization

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

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

$$\begin{aligned} \mathbb{A}^{\text{Lag}} &= \begin{bmatrix} \alpha_{1,1} & \alpha_{1,2} & \cdots & \alpha_{1,m+1} \\ \alpha_{2,1} & \alpha_{2,2} & \cdots & \alpha_{2,m+1} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{n+1,1} & \alpha_{n+1,2} & \cdots & \alpha_{n+1,m+1} \end{bmatrix} \\ \mathbf{c}_n &= \text{vec}\left(\mathbb{A}^{\text{Lag}}\right) \end{aligned}$$

Facts

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

Multi-variate realization

Generalized n-D Lagrangian realization (control the complexity)

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

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

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 $(1,1,2)$

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

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 $(1,1,2)$

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

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

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

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

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

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

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

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.

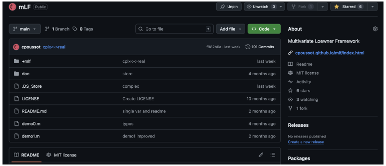
Content

The MATLAB `mLF` package

GitHub page

https://github.com/cpoussot/mLF

```matlab
%% 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);
```

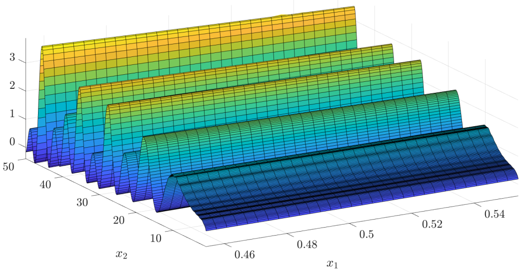


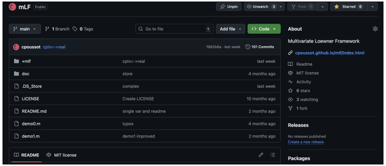
- Generic algorithm
  `mlf.alg1`
  `mlf.alg2`
- Specific steps
  `mlf.compute_order`,
  `mlf.point_selection`,
  `mlf.loewner_null_rec`,
  `mlf.tf_lagrangian`, ...

The MATLAB mLF package
GitHub page

`https://github.com/cpoussot/mLF`

mLF alg. 1 (direct), $r = [4, 113]$





- Generic algorithm
  `mlf.alg1`
  `mlf.alg2`
- 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);
```

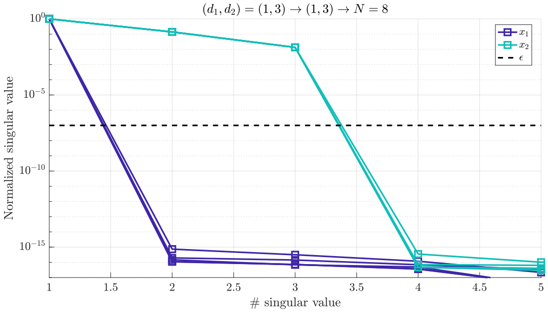
```
%%% Construct tensor
% Check that column/row IP are disjoint
ok          = mlf.check(p_c,p_r);
% Construct tab_n
[y,x,dim]   = mlf.make_tab_vec(H,p_c,p_r);
tab         = mlf.vec2mat(y,dim);
```

- 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);
```

$$
\left(\ \mathcal{B}_{\text{lag}}(x_1,x_2)\ \middle\|\ \mathbf{N}_{\text{lag}}\ \middle|\ \mathbf{D}_{\text{lag}}\ \right)
$$

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

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

where

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

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

The MATLAB `mLF` package

Detailed example 2-D (Monomial form)

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

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

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

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

where

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

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

The MATLAB `mLF` package

Detailed example 2-D (KST form)

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

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

$$\begin{bmatrix} \textbf{Lag}(x_1) & \textbf{Lag}(x_2) \end{bmatrix} = \begin{bmatrix} \frac{1}{s_1+\frac{7}{9}} & \frac{1}{s_2+\frac{7}{9}} \\ \frac{1}{s_1+\frac{7}{9}} & \frac{1}{s_2+\frac{1}{3}} \\ \frac{1}{s_1+\frac{7}{9}} & \frac{1}{s_2-\frac{1}{9}} \\ \frac{1}{s_1+\frac{7}{9}} & \frac{1}{s_2-1} \\ \frac{1}{s_1-1} & \frac{1}{s_2+\frac{7}{9}} \\ \frac{1}{s_1-1} & \frac{1}{s_2+\frac{1}{3}} \\ \frac{1}{s_1-1} & \frac{1}{s_2-\frac{1}{9}} \\ \frac{1}{s_1-1} & \frac{1}{s_2-1} \end{bmatrix}$$

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

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

KST rational interpolant

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

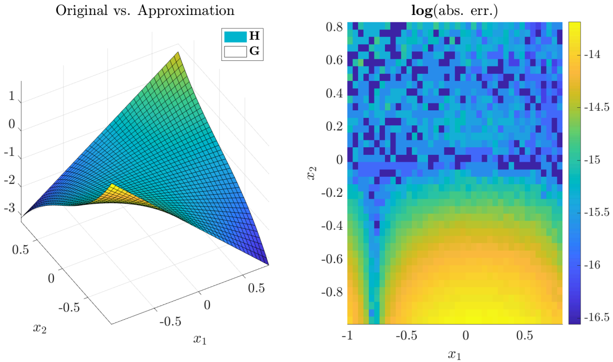
where numerator and denominator are

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

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);
```

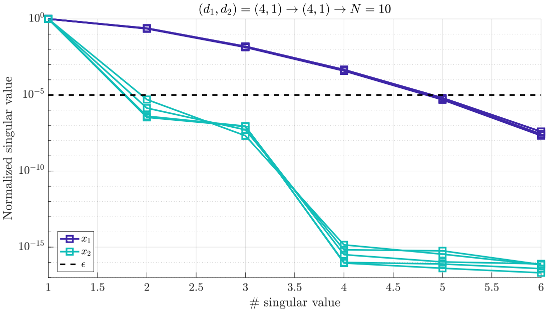
```
%%% Construct tensor
% Check that column/row IP are disjoint
ok           = mlf.check(p_c,p_r);
% Construct tab_n
[y,x,dim]    = mlf.make_tab_vec(H,p_c,p_r);
tab          = mlf.vec2mat(y,dim);
```

- 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-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 2-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);
```

- $(k_1, k_2) = (5, 2)$ column IP
- Barycentric dimension $5 \times 2 = 10 = N$
- $\lambda_1(j_1) = \begin{bmatrix} -1.64 & -0.909 & -0.182 & 0.545 & 2.0 \end{bmatrix}$ (pc{1})
- $\lambda_2(j_2) = \begin{bmatrix} -1.64 & 2.0 \end{bmatrix}$ (pc{2})
- $\mu_1(i_1) = \begin{bmatrix} -2.0 & -1.27 & -0.545 & 0.182 & 1.64 \end{bmatrix}$ (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: 165 `flop` and 25.0 `Bytes`
- Full: 1000 `flop` and 100.0 `Bytes`
- $\mathbf{W}^{\otimes}_{5,2} = \begin{bmatrix} 0.225 & 0.855 \\ 0.243 & 0.629 \\ 0.309 & 0.371 \\ 0.465 & 0.245 \\ 0.714 & 0.199 \end{bmatrix}$
- $\mathbf{C}^{\otimes}_{5,2} = \begin{bmatrix} -0.771 & 0.196 \\ 1.62 & -0.605 \\ -1.25 & 1.0 \\ 0.451 & -0.823 \\ -0.0712 & 0.247 \end{bmatrix}$

The MATLAB mLF package

Detailed example 2-D (Lagrange form)

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

$$
\left(\ \mathcal{B}_{\text{lag}}(x_1,x_2)\ \Big\|\ \mathbf{N}_{\text{lag}}\ \Big|\ \mathbf{D}_{\text{lag}}\ \right)
$$

$$
\left(\begin{array}{c||c|c}
(x_1+1.64)\ (x_2+1.64) & -0.174 & -0.771 \\
(x_2-2.0)\ (x_1+1.64) & 0.167 & 0.196 \\
(x_2+1.64)\ (x_1+0.909) & 0.394 & 1.62 \\
(x_2-2.0)\ (x_1+0.909) & -0.38 & -0.605 \\
(x_1+0.182)\ (x_2+1.64) & -0.385 & -1.25 \\
(x_2-2.0)\ (x_1+0.182) & 0.371 & 1.0 \\
(x_1-0.545)\ (x_2+1.64) & 0.21 & 0.451 \\
(x_2-2.0)\ (x_1-0.545) & -0.202 & -0.823 \\
(x_1-2.0)\ (x_2+1.64) & -0.0509 & -0.0712 \\
(x_1-2.0)\ (x_2-2.0) & 0.0491 & 0.247
\end{array}\right)
$$

$$
\mathbf{G}_{\text{lag}}(x_1,x_2) = \frac{\sum_{j_1=1}^{5}\sum_{j_2=1}^{2}\frac{\mathbf{C}^{\otimes}_{5,2}(j_1,j_2)\mathbf{W}^{\otimes}_{5,2}(j_1,j_2)}{(x_1-\lambda_1(j_1))(x_2-\lambda_2(j_2))}}{\sum_{j_1=1}^{5}\sum_{j_2=1}^{2}\frac{\mathbf{C}^{\otimes}_{5,2}(j_1,j_2)}{(x_1-\lambda_1(j_1))(x_2-\lambda_2(j_2))}},
$$

where

$$
\mathbf{W}^{\otimes}_{5,2} = \begin{bmatrix} 0.225 & 0.855 \\ 0.243 & 0.629 \\ 0.309 & 0.371 \\ 0.465 & 0.245 \\ 0.714 & 0.199 \end{bmatrix}
$$

$$
\mathbf{C}^{\otimes}_{5,2} = \begin{bmatrix} -0.771 & 0.196 \\ 1.62 & -0.605 \\ -1.25 & 1.0 \end{bmatrix}
$$

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

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

$$
\left(\begin{array}{c||c|c}
{x_1}^4\, x_2 & -2.9e-5 & -5.31e-4 \\
{x_1}^4 & 0.00288 & 0.00865 \\
{x_1}^3\, x_2 & 3.69e-5 & 0.0308 \\
{x_1}^3 & -0.00369 & -0.0111 \\
{x_1}^2\, x_2 & -0.00142 & -0.00965 \\
{x_1}^2 & 0.142 & 0.426 \\
x_1\, x_2 & 9.69e-5 & 0.333 \\
x_1 & -0.00967 & -0.029 \\
x_2 & -0.00333 & -1.17e-6 \\
1.0 & 0.333 & 1.0
\end{array}\right)
$$

$$
\mathbf{G}_{\mathrm{mon}}(x_1,x_2) = \frac{\sum_{j_1=1}^{5}\sum_{j_2=1}^{2} \mathbf{N}^{\otimes}_{5,2}(j_1,j_2)\left(x_1^{(j_1-1)}x_2^{(j_2-1)}\right)}{\sum_{j_1=1}^{5}\sum_{j_2=1}^{2} \mathbf{D}^{\otimes}_{5,2}(j_1,j_2)\left(x_1^{(j_1-1)}x_2^{(j_2-1)}\right)},
$$

where

$$
\mathbf{W}^{\otimes}_{5,2} = \begin{bmatrix} -2.9e-5 & 0.00288 \\ 3.69e-5 & -0.00369 \\ -0.00142 & 0.142 \\ 9.69e-5 & -0.00967 \\ -0.00333 & 0.333 \end{bmatrix}
$$

$$
\mathbf{C}^{\otimes}_{5,2} = \begin{bmatrix} -5.31e-4 & 0.00865 \\ 0.0308 & -0.0111 \\ -0.00965 & 0.426 \\ 0.333 & -0.029 \end{bmatrix}
$$

The MATLAB `mLF` package

Detailed example 2-D (KST form)

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

$$\begin{bmatrix} \textbf{Bary}(x_1) & \textbf{Bary}(x_2) \end{bmatrix} = \begin{bmatrix} 0.196 & -3.94 \\ 0.196 & 1.0 \\ -0.605 & -2.68 \\ -0.605 & 1.0 \\ 1.0 & -1.25 \\ 1.0 & 1.0 \\ -0.823 & -0.547 \\ -0.823 & 1.0 \\ 0.247 & -0.289 \\ 0.247 & 1.0 \end{bmatrix}$$

$$\begin{bmatrix} \textbf{Lag}(x_1) & \textbf{Lag}(x_2) \end{bmatrix} = \begin{bmatrix} \frac{1}{s_1+1.64} & \frac{1}{s_2+1.64} \\ \frac{1}{s_1+1.64} & \frac{1}{s_2-2.0} \\ \frac{1}{s_1+0.909} & \frac{1}{s_2+1.64} \\ \frac{1}{s_1+0.909} & \frac{1}{s_2-2.0} \\ \frac{1}{s_1+0.182} & \frac{1}{s_2+1.64} \\ \frac{1}{s_1+0.182} & \frac{1}{s_2-2.0} \\ \frac{1}{s_1-0.545} & \frac{1}{s_2+1.64} \\ \frac{1}{s_1-0.545} & \frac{1}{s_2-2.0} \\ \frac{1}{s_1-2.0} & \frac{1}{s_2+1.64} \\ \frac{1}{s_1-2.0} & \frac{1}{s_2-2.0} \end{bmatrix}$$

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

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

KST rational interpolant

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

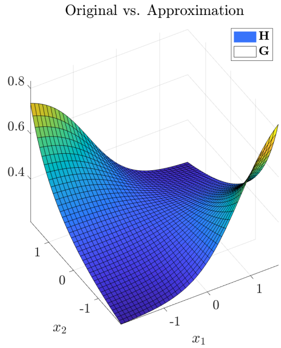
where numerator and denominator are

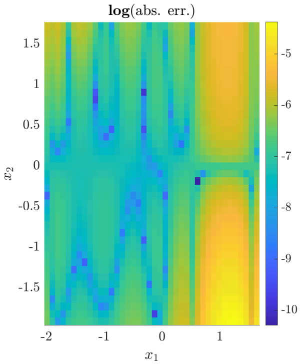
$$\begin{cases} \sum_{\text{rows}} \mathbf{w} \odot \mathbf{\Phi}_1(x_1) \odot \mathbf{\Phi}_2(x_2) \\ \sum_{\text{rows}} \mathbf{\Phi}_1(x_1) \odot \mathbf{\Phi}_2(x_2) \end{cases}$$

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 $\mathcal{X}_l := \left[\begin{array}{cc} -2.0 & 2.0 \end{array}\right]$, discretized with $N_l = 6$, $l = \{1,2\}$
- $\mathbf{x}_l = \left[\begin{array}{cccccc} -2.0 & -1.2 & -0.4 & 0.4 & 1.2 & 2.0 \end{array}\right] \in \mathcal{X}_l^6$

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);
```

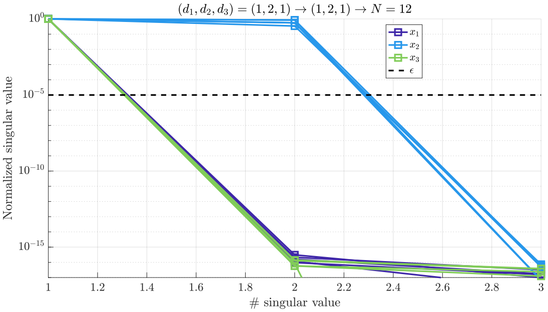
```
%%% Construct tensor
% Check that column/row IP are disjoint
ok          = mlf.check(p_c,p_r);
% Construct tab_n
[y,x,dim]   = mlf.make_tab_vec(H,p_c,p_r);
tab         = mlf.vec2mat(y,dim);
```

- 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`
- $\mathbf{W}^{\otimes}_{2,3,2} = \begin{bmatrix} 0.342 & 0.133 & 0.484 \\ -0.0219 & -0.197 & 0.467 \end{bmatrix}, \begin{bmatrix} -0.196 & 0.084 & -0.548 \\ -0.394 & -0.252 & -1.67 \end{bmatrix}$
- $\mathbf{C}^{\otimes}_{2,3,2} = \begin{bmatrix} -0.69 & 1.69 & -1.0 \\ 1.0 & -1.48 & 0.484 \end{bmatrix}, \begin{bmatrix} 0.69 & -1.69 & 1.0 \\ -1.0 & 1.48 & -0.484 \end{bmatrix}$

The MATLAB `mLF` package

Detailed example 3-D (Lagrange form)

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

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

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

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

The MATLAB mLF package

Detailed example 3-D (Monomial form)

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

$$\left(\mathcal{B}_{\text{mon}}(x_1,x_2,x_3) \;\middle\|\; \mathbf{N}_{\text{mon}} \;\middle|\; \mathbf{D}_{\text{mon}} \right)$$

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

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

The MATLAB mLF package

Detailed example 3-D (KST form)

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

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

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

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

$$\begin{bmatrix} \frac{1}{s_1+1.2} & \frac{1}{s_2+1.2} & \frac{1}{s_3+1.2} \\ \frac{1}{s_1+1.2} & \frac{1}{s_2+1.2} & \frac{1}{s_3-2.0} \\ \frac{1}{s_1+1.2} & \frac{1}{s_2-0.4} & \frac{1}{s_3+1.2} \\ \frac{1}{s_1+1.2} & \frac{1}{s_2-0.4} & \frac{1}{s_3-2.0} \\ \frac{1}{s_1+1.2} & \frac{1}{s_2-2.0} & \frac{1}{s_3+1.2} \\ \frac{1}{s_1+1.2} & \frac{1}{s_2-2.0} & \frac{1}{s_3-2.0} \\ \frac{1}{s_1-2.0} & \frac{1}{s_2+1.2} & \frac{1}{s_3+1.2} \\ \frac{1}{s_1-2.0} & \frac{1}{s_2+1.2} & \frac{1}{s_3-2.0} \\ \frac{1}{s_1-2.0} & \frac{1}{s_2-0.4} & \frac{1}{s_3+1.2} \\ \frac{1}{s_1-2.0} & \frac{1}{s_2-0.4} & \frac{1}{s_3-2.0} \\ \frac{1}{s_1-2.0} & \frac{1}{s_2-2.0} & \frac{1}{s_3+1.2} \\ \frac{1}{s_1-2.0} & \frac{1}{s_2-2.0} & \frac{1}{s_3-2.0} \end{bmatrix}$$

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

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

KST rational interpolant

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

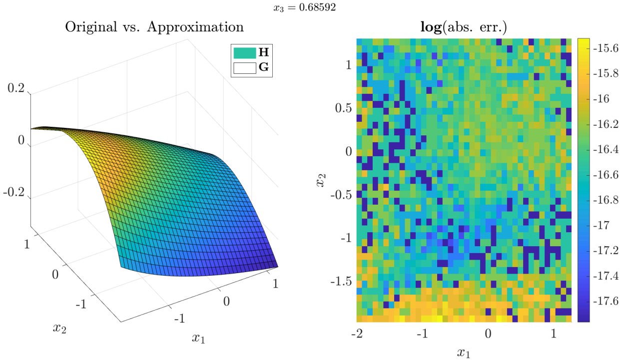
where numerator and denominator are

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

The MATLAB `mLF` package

Detailed example 3-D (KST form)

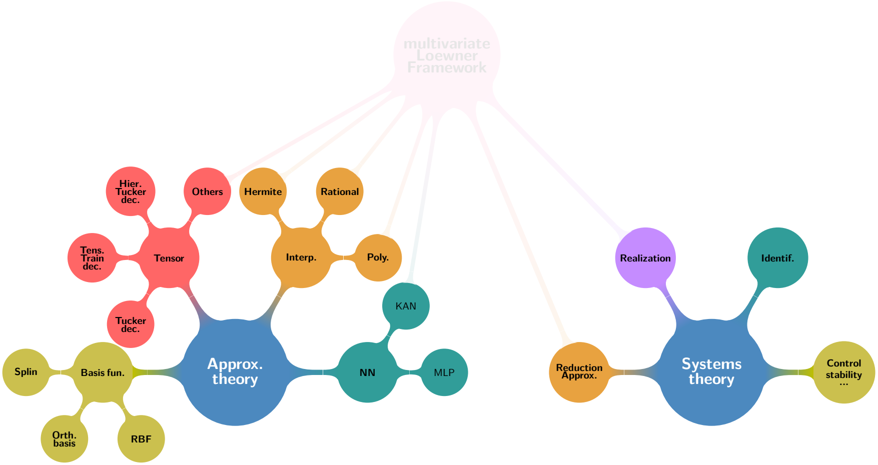
$$\mathbf{H}(x_1, x_2) = \frac{x_3\, {x_2}^2 + x_1}{x_1\, x_2 - 10}$$



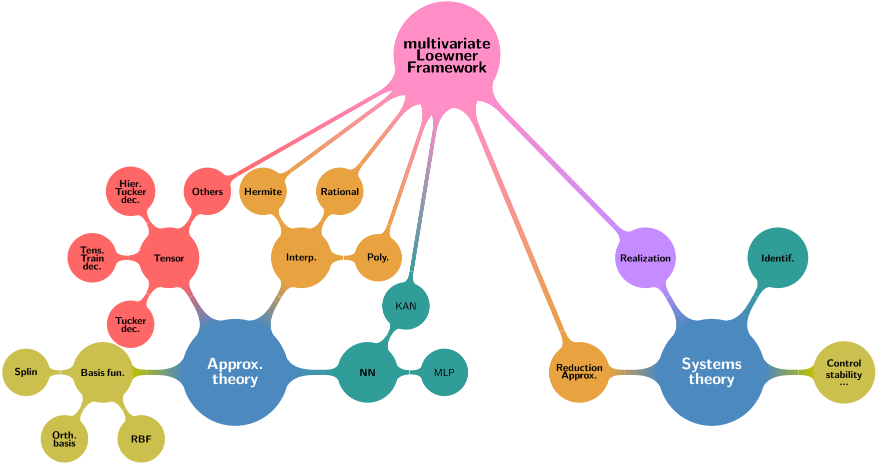
Content

Conclusion

Multivariate Loewner



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

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://github.com/cpoussot/mLF`

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

Content

Require: $\mathcal{T}_n^{\otimes}$

1: Check that interpolation points are disjoints.

2: Compute $d_j = \max_k \textbf{rank}\ \mathbb{L}_1^{(k)}$, the order along variable x_j (k being all possible combinations for frozen the variables $\{x_1, \cdots, x_{k-1}, x_{k+1}, \cdots, x_n\}$).

3: Construct data set, a sub-selection $P_c^{(n)}$ where $(k_1, k_2 \dots, k_n) = (d_1, d_2, \dots, d_n) + 1$; and $P_r^{(n)}$ where $(q_1, q_2 \dots, q_n)$ gather the rest of the data.

4: Compute $\mathbf{c}_n$, the n-D Loewner matrix null space.

5: Construct $\mathbb{A}^{\mathrm{Lag}}$, $\mathbb{B}^{\mathrm{Lag}}$, $\mathbf{\Gamma}$ and $\mathbf{\Delta}$ with any left/right separation.

6: Construct multivariate realization.

Ensure: $\hat{\mathbf{H}}(x_1, \dots, x_n) = \mathbf{W}\mathbf{\Phi}(x_1, x_2, \dots, x_n)^{-1}\mathbf{G}$ interpolates $\mathbf{H}(x_1, x_2, \dots, x_n)$ along $P_c^{(n)}$.

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

Require: $\mathcal{T}_n^{\otimes}$ and tolerance $\texttt{tol} > 0$

Check that interpolation points are disjoints.

while $\texttt{error} > \texttt{tol}$ **do**

  Search the point indexes with maximal $\texttt{error}$ (first iteration: pick any set).

  Add points in $P_c^{(n)}$ and put the remaining ones in $P_r^{(n)}$, obtain data set.

  Compute $\mathbf{c}_n$, the n-D Loewner matrix null space.

  Construct $\mathbb{A}^{\mathrm{Lag}}$, $\mathbb{B}^{\mathrm{Lag}}$, $\mathbf{\Gamma}$ and $\mathbf{\Delta}$ with any left/right separation.

  Construct multivariate realization.

  Evaluate $\texttt{error} = \max ||\widehat{\mathcal{T}_n^{\otimes}} - \mathcal{T}_n^{\otimes}||$ where $\widehat{\mathcal{T}_n^{\otimes}}$ is the evaluation of $\hat{\mathbf{H}}(x_1, \ldots, x_n)$ along the support points.

end while

Ensure: $\hat{\mathbf{H}}(x_1, \ldots, x_n) = \mathbf{W}\mathbf{\Phi}(x_1, x_2, \ldots, x_n)^{-1}\mathbf{G}$ interpolates $\mathbf{H}(x_1, x_2, \ldots, x_n)$ along $P_c^{(n)}$.

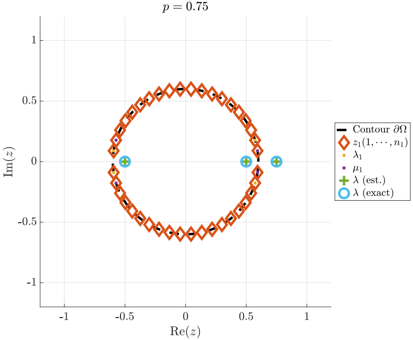
A.C. Rodriguez, L. Balicki and S. Gugercin, *"The p-AAA algorithm for data driven modeling of parametric dynamical systems"*, SICON, 45(3), 2023.

$$\mathbf{H}(\lambda)\mathbf{v} = 0$$

Linear model

$$\mathbf{H}(s,p) = \mathbf{1}^\top \left(sI_3 - \begin{bmatrix} 0 & 1 & 0 \\ 1-p & 0 & 0 \\ 0 & 1 & p \end{bmatrix} \right) \mathbf{1}$$

- `mlf.alg1`
- `mlf.make_realization_lag`



L. Balicki, M. Embree and S. Gugercin, *"A parametric Keldysh decomposition"*, arXiv:2601.01553v1.

$$\mathbf{H}(\lambda)\mathbf{v} = 0$$

Delay model

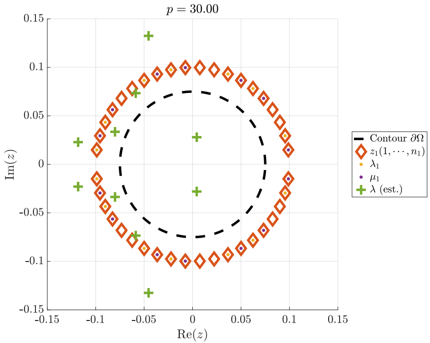
Let $E = \mathrm{diag}([10^{-4}, \cdots, 10^{10}])$, $n = 10$,

$$\mathbf{x} = -E\mathbf{x}(t) - 0.001\mathbf{x}(t-p)$$

or

$$\mathbf{H}(s,p) = \mathbf{1}_n^\top \left((s - 0.001e^{-ps})I - E\right)\mathbf{1}_n$$

- `mlf.alg1`
- `mlf.make_realization_lag`

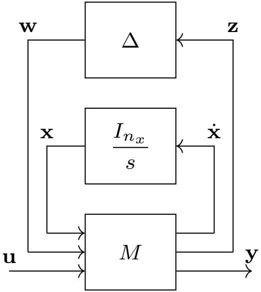


L. Balicki, M. Embree and S. Gugercin, *"A parametric Keldysh decomposition"*, arXiv:2601.01553v1.

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

$$
\left\{\begin{array}{ccl}
\mathbf{\Phi}_0 & = & \mathbf{\Phi}(0,0)\\
\mathbf{\Phi}_{x_1} & = & \mathbf{\Phi}(1,0)-\mathbf{\Phi}_0\\
\mathbf{\Phi}_{x_2} & = & \mathbf{\Phi}(0,1)-\mathbf{\Phi}_0\\
n_x & = & \kappa\\
n_w & = & 2(\ell-1)
\end{array}\right.
$$

$$
\left\{\begin{array}{ccl}
E & = & \mathbf{\Phi}_{x_1}\\
A & = & -\mathbf{\Phi}_0\\
B & = & \left[\begin{array}{c|c} -\mathbf{\Phi}_{x_2}(:,n_x+2:m-1) & \mathbf{G}\end{array}\right]\\
C & = & \left[\begin{array}{cc|c} \mathbf{0}_{n_w,n_x+1} & I_{n_w} & \mathbf{0}_{n_w}\\ \hline \multicolumn{3}{c}{\mathbf{W}}\end{array}\right]\\
D & = & \mathbf{0}\\
\Delta & = & x_2 I_{n_w}
\end{array}\right.
$$



Appendix & open research

On-going

- Parametric and uncertain modeling
- Uncertainty propagation