Contour Integral Methods for Nonlinear Eigenvalue Problems

A Systems Theory Perspective

Mark Embree and Serkan Gugercin · Virginia Tech

Michael Brennan · MIT

E-NLA Seminar · 17 June 2020

Thanks to: US National Science Foundation DMS-1720257

Linear and Nonlinear Eigenvalues Problems: A Simple String

Consider the wave equation on $0 \leq x \leq 1$

$$u_{tt}(x,t) = u_{xx}(x,t)$$

with the boundary conditions

$$u(0,t) = 0, \qquad u(1,t) = u_x(1,t).$$

Linear and Nonlinear Eigenvalues Problems: A Simple String

Consider the wave equation on $0 \le x \le 1$

$$u_{tt}(x,t) = u_{xx}(x,t)$$

with the boundary conditions

$$u(0,t) = 0, \qquad u(1,t) = u_x(1,t).$$

Solve by expanding $u(\cdot, t)$ in eigenfunctions, which must have the form

$$v''(x) = \lambda v(x) \qquad \Longrightarrow \qquad v(x) = A\sin(\lambda x) + B\cos(\lambda x)$$

Left boundary condition:

$$v(0) = 0 \qquad \Longrightarrow \qquad v(x) = A\sin(\lambda x)$$

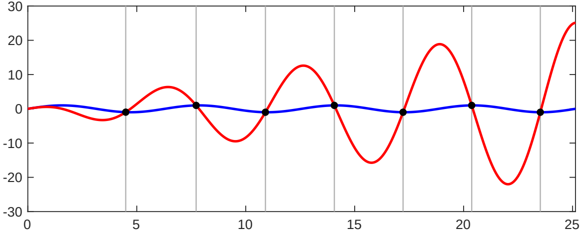
Right boundary condition:

$$v(1) = v'(1) \qquad \Longrightarrow \qquad \sin(\lambda) = \lambda\cos(\lambda)$$

Linear and Nonlinear Eigenvalues Problems: A Simple String

Shooting function: Eigenvalues λ are positive numbers that satisfy

$$\sin(\lambda) = \lambda\cos(\lambda)$$



A *scalar nonlinear eigenvalue problem* ($n = 1$) with infinitely many eigenvalues.

How can we best find these (leftmost) eigenvalues ?

- *Newton's method* for each root
- *Linearization* via polynomial approximation
- Other ideas . . .

Linear and Nonlinear Eigenvalues Problems: A Simple String

$$\sin(\lambda) = \lambda \cos(\lambda)$$

Approximate the nonlinear functions using polynomials, e.g.,

$$\left(\lambda - \frac{\lambda^3}{3!} + \frac{\lambda^5}{5!}\right) \approx \lambda \left(1 - \frac{\lambda^2}{2!} + \frac{\lambda^4}{4!}\right)$$

Find roots of the polynomial equation

$$0 = p(\lambda) = \lambda^5 - c_4\lambda^4 - c_3\lambda^3 - c_2\lambda^2 - c_1\lambda - c_0.$$

Linear and Nonlinear Eigenvalues Problems: A Simple String

$$\sin(\lambda) = \lambda\cos(\lambda)$$

Approximate the nonlinear functions using polynomials, e.g.,

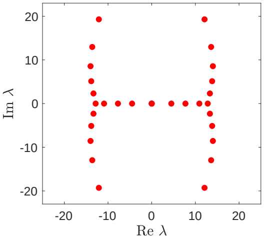
$$\left(\lambda - \frac{\lambda^3}{3!} + \frac{\lambda^5}{5!}\right) \approx \lambda\left(1 - \frac{\lambda^2}{2!} + \frac{\lambda^4}{4!}\right)$$

Find roots of the polynomial equation

$$0 = p(\lambda) = \lambda^5 - c_4\lambda^4 - c_3\lambda^3 - c_2\lambda^2 - c_1\lambda - c_0.$$

Companion "linearization" gives a *linear* eigenvalue problem $\mathbf{Av} = \lambda\mathbf{v}$ with, e.g.,

$$\mathbf{A} = \begin{bmatrix} & 1 & & & \\ & & 1 & & \\ & & & 1 & \\ & & & & 1 \\ c_0 & c_1 & c_2 & c_3 & c_4 \end{bmatrix}$$



$\deg(p) = 31$

Nonlinear Eigenvalue Problems: Stability of Delay Systems

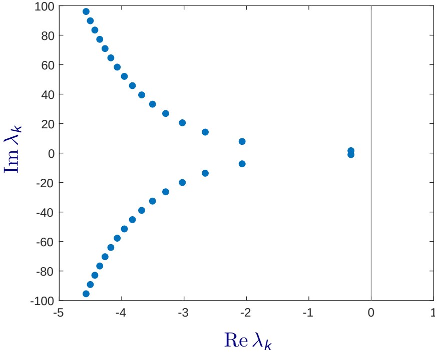
Consider the simple *scalar* delay differential equation

$$x'(t) = -x(t-1).$$

Substituting the ansatz $x(t) = e^{\lambda t}$ yields the *nonlinear eigenvalue problem*

$$T(\lambda) = 1 + \lambda e^{\lambda} = 0.$$

32 (of infinitely many) eigenvalues of T for this *scalar* ($n = 1$) equation:



eigenvalues derived from the Lambert-W function

See, e.g., [Michiels & Niculescu 2007]

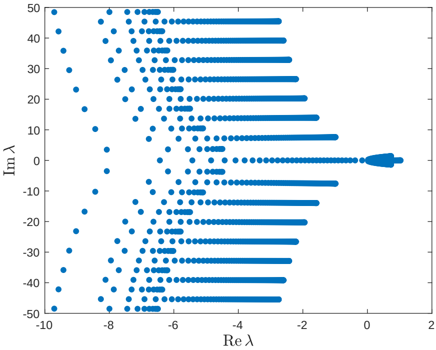
Nonlinear Eigenvalue Problems: Stability of Delay Systems

Consider the *system* of delay differential equations

$$\mathbf{E}\mathbf{x}'(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{x}(t-1).$$

Substituting the ansatz $\mathbf{x}(t) = \mathrm{e}^{\lambda t}\mathbf{v}$ yields the *nonlinear eigenvalue problem*

$$\mathbf{T}(\lambda)\mathbf{v} = \left(\mathbf{A} - \lambda\mathbf{E} + \mathrm{e}^{-\lambda}\mathbf{B}\right)\mathbf{v} = \mathbf{0}.$$



See, e.g., [Michiels & Niculescu 2007]

Nonlinear Eigenvalue Problems (NLEVP)

We seek eigenvalues of the nonlinear eigenvalue problem

$$\boxed{\mathbf{T}(\lambda)\mathbf{v} = \left(\sum_{k=0}^{d} f_k(\lambda)\mathbf{C}_k\right)\mathbf{v} = \mathbf{0}}$$

in some compact region $\Omega \subseteq \mathbb{C}$.

Nonlinear Eigenvalue Problems (NLEVP)

We seek eigenvalues of the nonlinear eigenvalue problem

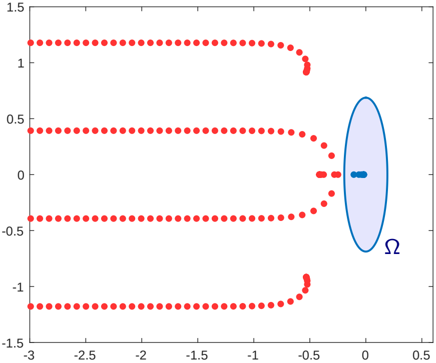
$$\boxed{\mathbf{T}(\lambda)\mathbf{v} = \left(\sum_{k=0}^{d} f_k(\lambda)\mathbf{C}_k\right)\mathbf{v} = \mathbf{0}}$$

in some compact region $\Omega \subseteq \mathbb{C}$.

- Helpful NLEVP surveys:
  Mehrmann & Voss, *GAMM*, [2004]
  Voss, *Handbook of Linear Algebra*, [2014]
  Güttel & Tisseur, *Acta Numerica* survey [2017]
- Software resources include:
  NLEVP test collection [Betcke, Higham, Mehrmann, Schröder, Tisseur 2013]
  SLEPC contains NLEVP algorithm implementations [Roman et al.]
- Many algorithms based on Newton's method, polynomial and rational approximation of the f_k's, projection, contour integration, etc.
  Incomplete list of contributors: Asakura, Bai, Betcke, Beyn, Effenberger, Gavin, Güttel, Ikegami, Jarlebring, Kimura, Kressner, Leitart, Meerbergen, Michiels, Miedlar, Niculescu, Pérez, Polizzi, Sakurai, Tadano, Tisseur, Van Barel, Van Beeumen, Vandereycken, Voss, Yokota, ...

Plan for the Talk

We seek m eigenvalues of $\mathbf{T}(z)$
contained within the bounded region $\Omega \subset \mathbb{C}$.



Here we seek the $m = 11$ eigenvalues in the blue ellipse, Ω.

Plan for the Talk

We seek m eigenvalues of $\mathbf{T}(z)$
contained within the bounded region $\Omega \subset \mathbb{C}$.

- Keldysh's Theorem and contour integration
- Fundamental algorithm:

  contour integration $\Longrightarrow$
  moments $\Longrightarrow$
  Hankel matrices $\Longrightarrow$
  eigenvalues

- Interpretation as *realization of associated dynamical systems*
- Extensions based on Loewner matrix interpolation:

  – Moment matching at a finite $z = \infty$
  – Tangential interpolation at multiple complex points

  Key technology: rational interpolation theory of Mayo and Antoulas [2007]

Hankel Contour Integral Method

for the

Nonlinear Eigenvalue Problems

Keldysh's Theorem: locally, the problem looks linear

Theorem [Keldysh 1951]. Suppose $\mathbf{T}(z)$ has m eigenvalues $\lambda_1, \ldots, \lambda_m$ (counting multiplicity) in the bounded region $\Omega \subset \mathbb{C}$, all semi-simple. Then

$$\mathbf{T}(z)^{-1} = \mathbf{V}(z\mathbf{I} - \mathbf{\Lambda})^{-1}\mathbf{W}^* + \mathbf{N}(z),$$

- $\mathbf{V} = [\mathbf{v}_1 \ \cdots \ \mathbf{v}_m]$, $\mathbf{W} = [\mathbf{w}_1 \ \cdots \ \mathbf{w}_m]$, $\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_m)$, $\mathbf{w}_j^* \mathbf{T}'(\lambda_j)\mathbf{v}_j = 1$;
- $\mathbf{N}(z)$ is analytic in Ω.

$$\mathbf{T}(z)^{-1} = \boxed{\mathbf{V}}\ \boxed{(z\mathbf{I} - \mathbf{\Lambda})^{-1}}\ \boxed{\mathbf{W}^*} + \boxed{\mathbf{N}(z)}$$

$\mathbf{H}(z) := \mathbf{V}(z\mathbf{I} - \mathbf{\Lambda})^{-1}\mathbf{W}^*$

$n \times n$ rational matrix function

m poles in Ω

nonlinear function, but analytic in Ω

Keldysh's Theorem: locally, the problem looks linear

$$\mathbf{T}(z)^{-1} = \mathbf{V}\,(z\mathbf{I}-\mathbf{\Lambda})^{-1}\,\mathbf{W}^* + \mathbf{N}(z)$$

$\mathbf{H}(z) := \mathbf{V}(z\mathbf{I}-\mathbf{\Lambda})^{-1}\mathbf{W}^*$

$n \times n$ rational matrix function

m poles in Ω

nonlinear function,

but analytic in Ω

Goal: Use samples of $\mathbf{T}(z_j)^{-1}$ (or $\mathbf{L}^*\mathbf{T}(z_j)^{-1}\mathbf{R}$) to discover $\mathbf{H}(z)$,

and hence $\mathbf{\Lambda} \in \mathbb{C}^{m\times m}$ and (ideally) eigenvector matrices $\mathbf{V}, \mathbf{W} \in \mathbb{C}^{n\times m}$.

Contour integration exposes the linear part

$$\mathbf{T}(z)^{-1} = \boxed{\mathbf{V}}\;\boxed{(z\mathbf{I}-\mathbf{\Lambda})^{-1}}\;\boxed{\mathbf{W}^*} + \boxed{\mathbf{N}(z)}$$

$\mathbf{H}(z) := \mathbf{V}(z\mathbf{I}-\mathbf{\Lambda})^{-1}\mathbf{W}^*$

analytic in Ω

For any f analytic on Ω,

$$\begin{aligned}
\frac{1}{2\pi \mathrm{i}}\int_{\partial\Omega} f(z)\mathbf{T}(z)^{-1}\,\mathrm{d}z &= \frac{1}{2\pi \mathrm{i}}\int_{\partial\Omega} f(z)\mathbf{V}(z\mathbf{I}-\mathbf{\Lambda})^{-1}\mathbf{W}^*\,\mathrm{d}z + \frac{1}{2\pi \mathrm{i}}\int_{\partial\Omega} f(z)\mathbf{N}(z)\,\mathrm{d}z \\
&= \frac{1}{2\pi \mathrm{i}}\int_{\partial\Omega} f(z)\mathbf{V}(z\mathbf{I}-\mathbf{\Lambda})^{-1}\mathbf{W}^*\,\mathrm{d}z \\
&= \mathbf{V}f(\mathbf{\Lambda})\mathbf{W}^*.
\end{aligned}$$

Expedite calculation via sampling, trapezoid rule

$$\frac{1}{2\pi \mathrm{i}} \int_{\partial\Omega} f(z)\mathbf{T}(z)^{-1}\,\mathrm{d}z = \mathbf{V}f(\mathbf{\Lambda})\mathbf{W}^* \in \mathbb{C}^{n\times n}.$$

- **Reduce dimension via sampling (*sketching*)**

  For $\mathbf{L} \in \mathbb{C}^{n\times \ell}$ and $\mathbf{R} \in \mathbb{C}^{n\times r}$ (e.g., random), compute

$$\frac{1}{2\pi \mathrm{i}} \int_{\partial\Omega} f(z)\,\mathbf{L}^*\mathbf{T}(z)^{-1}\mathbf{R}\,\mathrm{d}z = \mathbf{L}^*\mathbf{V}f(\mathbf{\Lambda})\mathbf{W}^*\mathbf{R} \in \mathbb{C}^{\ell\times r}.$$

- **Approximate the integral using the trapezoid rule**

$$\frac{1}{2\pi \mathrm{i}} \int_{\partial\Omega} f(z)\,\mathbf{L}^*\mathbf{T}(z)^{-1}\mathbf{R}\,\mathrm{d}z \approx \sum_{j=0}^{N} w_j\, f(\zeta_j)\,\mathbf{L}^*\mathbf{T}(\zeta_j)^{-1}\mathbf{R}.$$

  – The terms $\mathbf{L}^*\mathbf{T}(\zeta_j)^{-1}\mathbf{R} \in \mathbb{C}^{\ell\times r}$ can be computed in parallel

  – Once these terms have been computed, easy to use different f.

Motivating idea for the algorithms in this talk

We seek the m eigenvalues of $\mathbf{T}(z)$ in Ω.

- Keldysh's Theorem + contour integration of $\mathbf{T}(z)^{-1} = \mathbf{H}(z) + \mathbf{N}(z)$ gives access to $\mathbf{H}(z)$ (or $\mathbf{L}^*\mathbf{H}(z)\mathbf{R}$)
- $\mathbf{H}(z) : \mathbb{C} \to \mathbb{C}^{n\times n}$ is a rational matrix function of McMillan degree m
- Can we use $O(m)$ samples of $\mathbf{H}(z)$ (or $\mathbf{L}^*\mathbf{H}(z)\mathbf{R}$) to discover the rational function $\mathbf{H}(z)$ (or *realize* an associated dynamical system) via interpolation/approximation ?

Contour integration method for NLEVPs

- [Asakura, Sakurai, Tadano, Ikegami, Kimura 2009] and [Beyn 2012] proposed several influential algorithms based on the choice $f(z) = z^j$, analogous to the Sakurai–Sugiura method for linear eigenvalue problems. Filter function enhancement: [Van Barel, Kravanja 2016] and [Van Barel 2016].
- See [Güttel, Tisseur 2017] for a presentation and numerical experiments.
- These algorithms have potential for development into black-box software for the nonlinear eigenvalue problem; see [Porzio,Tisseur, ICIAM 2019].

The **basic algorithm** uses

$$f(z) \equiv 1, \qquad \mathbf{A}_0 = \frac{1}{2\pi \mathrm{i}} \int_{\partial\Omega} \mathbf{L}^* \mathbf{T}(z)^{-1} \mathbf{R} \, \mathrm{d}z \quad = \; \mathbf{L}^* \mathbf{V} \mathbf{W}^* \mathbf{R},$$

$$f(z) = z, \qquad \mathbf{A}_1 = \frac{1}{2\pi \mathrm{i}} \int_{\partial\Omega} z \, \mathbf{L}^* \mathbf{T}(z)^{-1} \mathbf{R} \, \mathrm{d}z \; = \; \mathbf{L}^* \mathbf{V} \mathbf{\Lambda} \mathbf{W}^* \mathbf{R}.$$

Contour integration method for NLEVPs

- [Asakura, Sakurai, Tadano, Ikegami, Kimura 2009] and [Beyn 2012] proposed several influential algorithms based on the choice $f(z) = z^j$, analogous to the Sakurai–Sugiura method for linear eigenvalue problems. Filter function enhancement: [Van Barel, Kravanja 2016] and [Van Barel 2016].
- See [Güttel, Tisseur 2017] for a presentation and numerical experiments.
- These algorithms have potential for development into black-box software for the nonlinear eigenvalue problem; see [Porzio,Tisseur, ICIAM 2019].

The **basic algorithm** uses

$$f(z) \equiv 1, \qquad \mathbf{A}_0 = \frac{1}{2\pi \mathrm{i}} \int_{\partial\Omega} \mathbf{L}^* \mathbf{T}(z)^{-1} \mathbf{R}\, \mathrm{d}z \quad = \; \mathbf{L}^* \mathbf{V} \mathbf{W}^* \mathbf{R},$$

$$f(z) = z, \qquad \mathbf{A}_1 = \frac{1}{2\pi \mathrm{i}} \int_{\partial\Omega} z\, \mathbf{L}^* \mathbf{T}(z)^{-1} \mathbf{R}\, \mathrm{d}z \; = \; \mathbf{L}^* \mathbf{V} \boldsymbol{\Lambda} \mathbf{W}^* \mathbf{R}.$$

One could then analyze the $\ell \times r$ rectangular matrix pencil

$$z\mathbf{A}_0 - \mathbf{A}_1 = z\mathbf{L}^* \mathbf{V} \mathbf{W}^* \mathbf{R} - \mathbf{L}^* \mathbf{V} \boldsymbol{\Lambda} \mathbf{W}^* \mathbf{R} = \mathbf{L}^* \mathbf{V} (z\mathbf{I} - \boldsymbol{\Lambda}) \mathbf{W}^* \mathbf{R}.$$

Basic contour integration method for NLEVPs

Instead of analyzing the pencil $z\mathbf{A}_0 - \mathbf{A}_1 = \mathbf{L}^*\mathbf{V}(z\mathbf{I} - \mathbf{\Lambda})\mathbf{W}^*\mathbf{R}$,
compress to an $m \times m$ matrix and solve a standard linear eigenvalue problem.

Rank condition

$$\operatorname{rank}(\mathbf{L}^*\mathbf{V}) = \operatorname{rank}(\mathbf{W}^*\mathbf{R}) = \operatorname{rank}(\mathbf{A}_0) = m$$

Reduced SVD of $\mathbf{A}_0$

$$\mathbf{A}_0 = \mathbf{X}\mathbf{\Sigma}\mathbf{Y}^*$$

Rank condition implies $\operatorname{range}(\mathbf{L}^*\mathbf{V}) = \operatorname{range}(\mathbf{X})$, hence invertible $\mathbf{S} \in \mathbb{C}^{m\times m}$ with

$$\mathbf{L}^*\mathbf{V} = \mathbf{X}\mathbf{S} \qquad \Longrightarrow \qquad \mathbf{\Sigma} = \mathbf{X}^*\mathbf{L}^*\mathbf{V}\mathbf{W}^*\mathbf{R}\mathbf{Y} = \mathbf{S}\mathbf{W}^*\mathbf{R}\mathbf{Y}$$

Basic contour integration method for NLEVPs

Instead of analyzing the pencil $z\mathbf{A}_0 - \mathbf{A}_1 = \mathbf{L}^*\mathbf{V}(z\mathbf{I} - \mathbf{\Lambda})\mathbf{W}^*\mathbf{R}$*,*
compress to an $m \times m$ *matrix and solve a standard linear eigenvalue problem.*

Rank condition

$$\mathrm{rank}(\mathbf{L}^*\mathbf{V}) = \mathrm{rank}(\mathbf{W}^*\mathbf{R}) = \mathrm{rank}(\mathbf{A}_0) = m$$

Reduced SVD of $\mathbf{A}_0$

$$\mathbf{A}_0 = \mathbf{X}\mathbf{\Sigma}\mathbf{Y}^*$$

Rank condition implies $\mathrm{range}(\mathbf{L}^*\mathbf{V}) = \mathrm{range}(\mathbf{X})$, hence invertible $\mathbf{S} \in \mathbb{C}^{m\times m}$ with

$$\mathbf{L}^*\mathbf{V} = \mathbf{X}\mathbf{S} \qquad \Longrightarrow \qquad \mathbf{\Sigma} = \mathbf{X}^*\mathbf{L}^*\mathbf{V}\mathbf{W}^*\mathbf{R}\mathbf{Y} = \mathbf{S}\mathbf{W}^*\mathbf{R}\mathbf{Y}$$

Reduction to $m \times m$ matrix eigenvalue problem

$$\begin{aligned}
\mathbf{B} &:= \mathbf{X}^*\mathbf{A}_1\mathbf{Y}\mathbf{\Sigma}^{-1} \\
&= \mathbf{X}^*(\mathbf{L}^*\mathbf{V}\mathbf{\Lambda}\mathbf{W}^*\mathbf{R})\mathbf{Y}\mathbf{\Sigma}^{-1} \\
&= \mathbf{X}^*(\mathbf{L}^*\mathbf{V})\mathbf{\Lambda}(\mathbf{S}^{-1}\mathbf{S})(\mathbf{W}^*\mathbf{R})\mathbf{Y}\mathbf{\Sigma}^{-1} \\
&= \mathbf{X}^*(\mathbf{X}\mathbf{S})\mathbf{\Lambda}\mathbf{S}^{-1}(\mathbf{S}\mathbf{W}^*\mathbf{R}\mathbf{Y})\mathbf{\Sigma}^{-1} \quad = \quad \mathbf{S}\mathbf{\Lambda}\mathbf{S}^{-1}
\end{aligned}$$

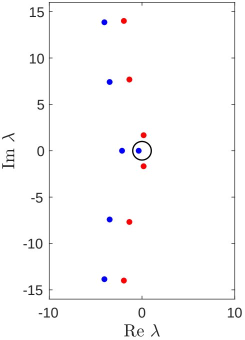
A simple example

The delay differential equation

$$\mathbf{x}'(t) = \begin{bmatrix} -1/4 & 0 \\ 0 & -2 \end{bmatrix} \mathbf{x}(t-1)$$

leads to the NLEVP

$$\mathbf{T}(z) = \begin{bmatrix} -1/4 + ze^z & 0 \\ 0 & -2 + ze^z \end{bmatrix}.$$



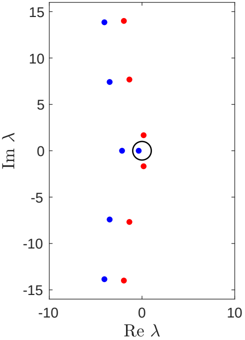
A simple example

The delay differential equation

$$\mathbf{x}'(t) = \begin{bmatrix} -1/4 & 0 \\ 0 & -2 \end{bmatrix} \mathbf{x}(t-1)$$

leads to the NLEVP

$$\mathbf{T}(z) = \begin{bmatrix} -1/4 + ze^z & 0 \\ 0 & -2 + ze^z \end{bmatrix}.$$



Basic algorithm:

$$\mathbf{A}_0 = \mathbf{X\Sigma Y}^*, \qquad \mathbf{B} = \mathbf{X}^*\mathbf{A}_1\mathbf{Y\Sigma}^{-1} = \mathbf{S\Lambda S}^{-1}, \qquad \mathbf{L}^*\mathbf{V} = \mathbf{XS}$$

Integrate around $\partial\Omega =$ unit circle with $\mathbf{L} = \mathbf{R} = \mathbf{I}$ to get

$$\mathbf{A}_0 = \frac{1}{2\pi \mathrm{i}} \int_{\partial\Omega} \begin{bmatrix} \frac{1}{-1/4+ze^z} & 0 \\ 0 & \frac{1}{-2+ze^z} \end{bmatrix} \mathrm{d}z = \begin{bmatrix} 2.22474 & 0 \\ 0 & 0 \end{bmatrix},$$

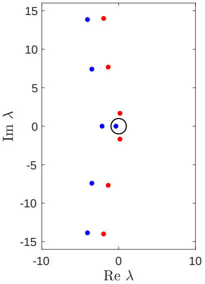
$$\mathbf{A}_1 = \frac{1}{2\pi \mathrm{i}} \int_{\partial\Omega} \begin{bmatrix} \frac{z}{-1/4+ze^z} & 0 \\ 0 & \frac{z}{-2+ze^z} \end{bmatrix} \mathrm{d}z = \begin{bmatrix} -0.79513 & 0 \\ 0 & 0 \end{bmatrix}.$$

A simple example

$$\mathbf{T}(z) = \begin{bmatrix} -1/4 + ze^z & 0 \\ 0 & -2 + ze^z \end{bmatrix}.$$

$$\mathbf{A}_0 = \frac{1}{2\pi\mathrm{i}} \int_{\partial\Omega} \begin{bmatrix} \frac{1}{-1/4+ze^z} & 0 \\ 0 & \frac{1}{-2+ze^z} \end{bmatrix} \mathrm{d}z = \begin{bmatrix} 2.22474 & 0 \\ 0 & 0 \end{bmatrix}$$

$$\mathbf{A}_1 = \frac{1}{2\pi\mathrm{i}} \int_{\partial\Omega} \begin{bmatrix} \frac{z}{-1/4+ze^z} & 0 \\ 0 & \frac{z}{-2+ze^z} \end{bmatrix} \mathrm{d}z = \begin{bmatrix} -0.79513 & 0 \\ 0 & 0 \end{bmatrix}$$



Basic algorithm: $\mathbf{A}_0 = \mathbf{X\Sigma Y}^*, \quad \mathbf{B} = \mathbf{X}^*\mathbf{A}_1\mathbf{Y\Sigma}^{-1} = \mathbf{S\Lambda S}^{-1}, \quad \mathbf{L}^*\mathbf{V} = \mathbf{XS}$

$$\mathbf{A}_0 = \mathbf{X\Sigma Y}^* = \begin{bmatrix} 1 \\ 0 \end{bmatrix} [2.22474] \begin{bmatrix} 1 & 0 \end{bmatrix}$$

$$\mathbf{B} = \mathbf{X}^*\mathbf{A}_1\mathbf{Y\Sigma}^{-1} = \frac{-0.79513}{2.22474} = -0.35740 = \mathbf{S\Lambda S}^{-1}$$

$$\mathbf{V} = \mathbf{L}^*\mathbf{V} = \mathbf{XS} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

Basic contour integration method for NLEVPs

Basic algorithm:

$$\mathbf{A}_0 = \mathbf{X}\mathbf{\Sigma}\mathbf{Y}^*, \qquad \mathbf{B} = \mathbf{X}^*\mathbf{A}_1\mathbf{Y}\mathbf{\Sigma}^{-1} = \mathbf{S}\mathbf{\Lambda}\mathbf{S}^{-1}, \qquad \mathbf{L}^*\mathbf{V} = \mathbf{X}\mathbf{S}$$

If $\mathbf{L} = \mathbf{I}$, recover $\mathbf{V} = \mathbf{X}\mathbf{S}$; otherwise, use inverse iteration for eigenvectors.

What could go wrong?

- Dimension n could be smaller then the number of eigenvalues m.
- Rank condition fails for poor choices of $\mathbf{L}$ and $\mathbf{R}$.
- Rank condition fails for linearly dependent eigenvectors in $\mathbf{V}$ and $\mathbf{W}$, including (but not limited to) derogatory multiple eigenvalues.
- Rank of $\mathbf{A}_0$ might be difficult to detect (complicated by quadrature errors).

Basic contour integration method for NLEVPs

Basic algorithm:

$$\mathbf{A}_0 = \mathbf{X}\boldsymbol{\Sigma}\mathbf{Y}^*, \qquad \mathbf{B} = \mathbf{X}^*\mathbf{A}_1\mathbf{Y}\boldsymbol{\Sigma}^{-1} = \mathbf{S}\boldsymbol{\Lambda}\mathbf{S}^{-1}, \qquad \mathbf{L}^*\mathbf{V} = \mathbf{X}\mathbf{S}$$

If $\mathbf{L} = \mathbf{I}$, recover $\mathbf{V} = \mathbf{X}\mathbf{S}$; otherwise, use inverse iteration for eigenvectors.

What could go wrong?

- Dimension n could be smaller then the number of eigenvalues m.
- Rank condition fails for poor choices of $\mathbf{L}$ and $\mathbf{R}$.
- Rank condition fails for linearly dependent eigenvectors in $\mathbf{V}$ and $\mathbf{W}$, including (but not limited to) derogatory multiple eigenvalues.
- Rank of $\mathbf{A}_0$ might be difficult to detect (complicated by quadrature errors).

Remedy: Incorporate higher moments

$$f(z) = z^k, \qquad \mathbf{A}_k = \frac{1}{2\pi \mathrm{i}} \int_{\partial\Omega} z^k\, \mathbf{L}^*\mathbf{T}(z)^{-1}\mathbf{R}\,\mathrm{d}z \;=\; \mathbf{L}^*\mathbf{V}\boldsymbol{\Lambda}^k\mathbf{W}^*\mathbf{R}$$

These moments are easy to compute from the same data used for $\mathbf{A}_0$ *and* $\mathbf{A}_1$*.*

Embed higher moments in Hankel Matrices

- Let $\mathbf{L} \in \mathbb{C}^{n \times \ell}$ and $\mathbf{R} \in \mathbb{C}^{n \times r}$ for $\ell, r \ll n$.
- Compute $\mathbf{A}_k = \mathbf{L}^* \mathbf{V} \mathbf{\Lambda}^k \mathbf{W}^* \mathbf{R} \in \mathbb{C}^{\ell \times r}$, using the trapezoid rule, $f(z) = z^k$.
- Pick some $K \geq 1$ and form the block Hankel matrices

$$\mathbb{H} = \begin{bmatrix} \mathbf{A}_0 & \mathbf{A}_1 & \cdots & \mathbf{A}_{K-1} \\ \mathbf{A}_1 & \mathbf{A}_2 & \cdots & \mathbf{A}_K \\ \vdots & \vdots & \iddots & \vdots \\ \mathbf{A}_{K-1} & \mathbf{A}_K & \cdots & \mathbf{A}_{2K-2} \end{bmatrix}, \quad \mathbb{H}_s = \begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_2 & \cdots & \mathbf{A}_K \\ \mathbf{A}_2 & \mathbf{A}_3 & \cdots & \mathbf{A}_{K+1} \\ \vdots & \vdots & \iddots & \vdots \\ \mathbf{A}_K & \mathbf{A}_{K+1} & \cdots & \mathbf{A}_{2K-1} \end{bmatrix}.$$

Embed higher moments in Hankel Matrices

- Let $\mathbf{L} \in \mathbb{C}^{n\times\ell}$ and $\mathbf{R} \in \mathbb{C}^{n\times r}$ for $\ell, r \ll n$.
- Compute $\mathbf{A}_k = \mathbf{L}^*\mathbf{V}\mathbf{\Lambda}^k\mathbf{W}^*\mathbf{R} \in \mathbb{C}^{\ell\times r}$, using the trapezoid rule, $f(z) = z^k$.
- Pick some $K \geq 1$ and form the block Hankel matrices

$$\mathbb{H} = \begin{bmatrix} \mathbf{A}_0 & \mathbf{A}_1 & \cdots & \mathbf{A}_{K-1} \\ \mathbf{A}_1 & \mathbf{A}_2 & \cdots & \mathbf{A}_K \\ \vdots & \vdots & \iddots & \vdots \\ \mathbf{A}_{K-1} & \mathbf{A}_K & \cdots & \mathbf{A}_{2K-2} \end{bmatrix}, \quad \mathbb{H}_s = \begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_2 & \cdots & \mathbf{A}_K \\ \mathbf{A}_2 & \mathbf{A}_3 & \cdots & \mathbf{A}_{K+1} \\ \vdots & \vdots & \iddots & \vdots \\ \mathbf{A}_K & \mathbf{A}_{K+1} & \cdots & \mathbf{A}_{2K-1} \end{bmatrix}.$$

The block Hankel matrices can be factored as

$$\mathbb{H} = \mathcal{V}\mathcal{W}^*, \quad \mathbb{H}_s = \mathcal{V}\mathbf{\Lambda}\mathcal{W}^*,$$

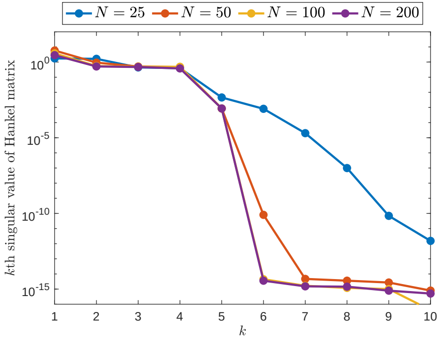
$$\mathcal{V} = \begin{bmatrix} \mathbf{L}^*\mathbf{V} \\ \mathbf{L}^*\mathbf{V}\mathbf{\Lambda} \\ \vdots \\ \mathbf{L}^*\mathbf{V}\mathbf{\Lambda}^{K-1} \end{bmatrix} \in \mathbb{C}^{K\ell\times m}, \quad \mathcal{W}^* = \begin{bmatrix} \mathbf{W}^*\mathbf{R} & \mathbf{\Lambda}\mathbf{W}^*\mathbf{R} & \cdots & \mathbf{\Lambda}^{K-1}\mathbf{W}^*\mathbf{R} \end{bmatrix} \in \mathbb{C}^{m\times Kr}.$$

Rank condition

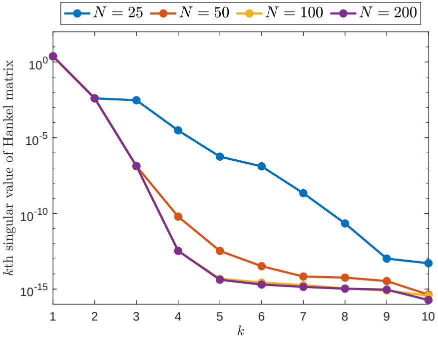
$$\mathrm{rank}(\mathcal{V}) = \mathrm{rank}(\mathcal{W}) = \mathrm{rank}(\mathbb{H}) = m$$

Singular values of $\mathbb{H}$ reveal m...

In all cases here, we seek $m = 5$ eigenvalues in the unit circle.
The parameter N shows the number of trapezoid rule points.
We use $\ell = r = 1$ sampling direction.



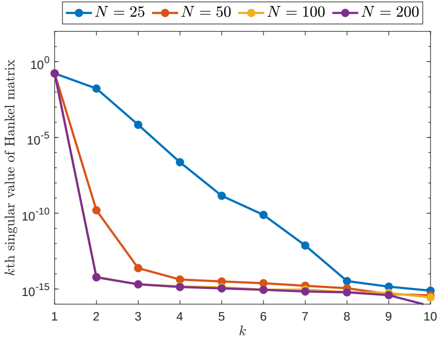
desired eigenvalues are well separated
undesired eigenvalues are far from Ω
rank($\mathbb{H}$) $= 5$ is clear



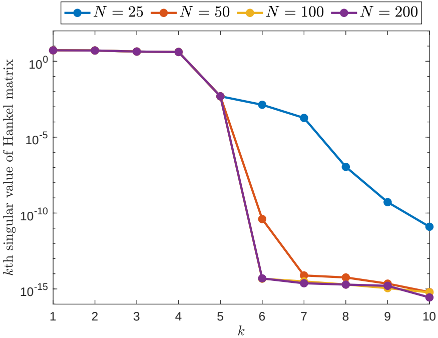
desired eigenvalues are tightly clustered
undesired eigenvalues are far from Ω
rank($\mathbb{H}$) $= 5$ is unclear

Singular values of $\mathbb{H}$ reveal m...

In all cases here, we seek $m = 5$ eigenvalues in the unit circle.
The parameter N shows the number of trapezoid rule points.
We use $\ell = r = 1$ sampling direction.



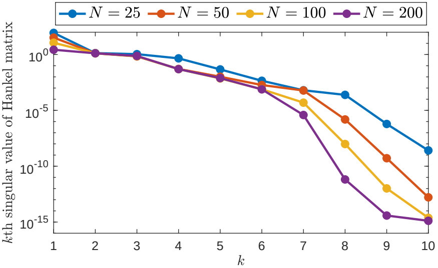
desired eigenvalue is derogatory (multiple)
undesired eigenvalues are far from Ω
eigenvectors missing with $\ell = r = 1$
rank($\mathbb{H}$) = 1 is clear



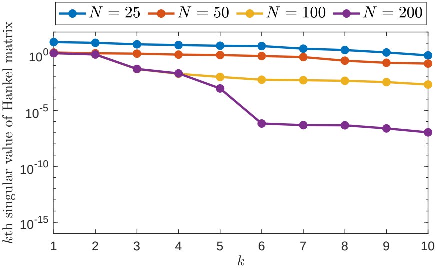
desired eigenvalues for a Jordan block
undesired eigenvalues are far from Ω
Jordan block discovered with $\ell = r = 1$
rank($\mathbb{H}$) = 5 is clear

Singular values of $\mathbb{H}$ reveal m...

In all cases here, we seek $m = 5$ eigenvalues in the unit circle.
The parameter N shows the number of trapezoid rule points.
We use $\ell = r = 1$ sampling direction.



desired eigenvalues are well separated
one undesired eigenvalue near Ω ($\lambda = 1.01$)
trapezoid rule is slow to converge
rank($\mathbb{H}$) = 5 is not clear



desired eigenvalues are well separated
all undesired eigenvalue near Ω ($|\lambda| = 1.1$)
trapezoid rule is slow to converge
rank($\mathbb{H}$) = 5 is not clear

Contour methods for NLEVPs via Hankel matrices

$$\mathbb{H} = \begin{bmatrix} \mathbf{A}_0 & \mathbf{A}_1 & \cdots & \mathbf{A}_{K-1} \\ \mathbf{A}_1 & \mathbf{A}_2 & \cdots & \mathbf{A}_K \\ \vdots & \vdots & \iddots & \vdots \\ \mathbf{A}_{K-1} & \mathbf{A}_K & \cdots & \mathbf{A}_{2K-2} \end{bmatrix}, \quad \mathbb{H}_s = \begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_2 & \cdots & \mathbf{A}_K \\ \mathbf{A}_2 & \mathbf{A}_3 & \cdots & \mathbf{A}_{K+1} \\ \vdots & \vdots & \iddots & \vdots \\ \mathbf{A}_K & \mathbf{A}_{K+1} & \cdots & \mathbf{A}_{2K-1} \end{bmatrix}.$$

Rank condition

$$\mathrm{rank}(\mathcal{V}) = \mathrm{rank}(\mathcal{W}) = \mathrm{rank}(\mathbb{H}) = m$$

Reduced SVD of $\mathbb{H}$

$$\mathbb{H} = \mathbf{X}\boldsymbol{\Sigma}\mathbf{Y}^*$$

Reduction to $m \times m$ matrix eigenvalue problem

$$\mathbf{B} := \mathbf{X}^*\mathbb{H}_s\mathbf{Y}\boldsymbol{\Sigma}^{-1} = \mathbf{S}\boldsymbol{\Lambda}\mathbf{S}^{-1},$$

perfectly generalizing the $K = 1$ case,

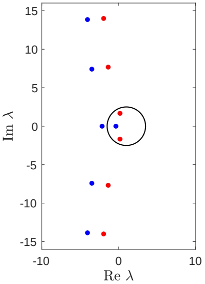
$$\mathbf{B} := \mathbf{X}^*\mathbf{A}_1\mathbf{Y}\boldsymbol{\Sigma}^{-1} = \mathbf{S}\boldsymbol{\Lambda}\mathbf{S}^{-1}.$$

A simple example, revisited

$$\mathbf{T}(z) = \begin{bmatrix} -1/4 + ze^z & 0 \\ 0 & -2 + ze^z \end{bmatrix}.$$

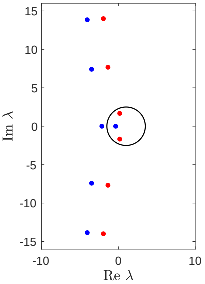
Enlarge Ω to contain $m = 3$ eigenvalues (so $m > n$).

$$\mathbb{H} = \begin{bmatrix} \mathbf{A}_0 & \mathbf{A}_1 \\ \mathbf{A}_1 & \mathbf{A}_2 \end{bmatrix} = \left[\begin{array}{cc|cc} 2.2247 & 0 & -0.7951 & 0 \\ 0 & -0.7192 & 0 & 0.5464 \\ \hline -0.7951 & 0 & 0.2842 & 0 \\ 0 & 0.5464 & 0 & 2.225 \end{array}\right]$$



A simple example, revisited

$$\mathbf{T}(z) = \begin{bmatrix} -1/4 + ze^z & 0 \\ 0 & -2 + ze^z \end{bmatrix}.$$



Enlarge Ω to contain $m = 3$ eigenvalues (so $m > n$).

$$\mathbb{H} = \begin{bmatrix} \mathbf{A}_0 & \mathbf{A}_1 \\ \mathbf{A}_1 & \mathbf{A}_2 \end{bmatrix} = \left[\begin{array}{cc|cc} 2.2247 & 0 & -0.7951 & 0 \\ 0 & -0.7192 & 0 & 0.5464 \\ \hline -0.7951 & 0 & 0.2842 & 0 \\ 0 & 0.5464 & 0 & 2.225 \end{array}\right]$$

Basic algorithm: $\mathbb{H} = \mathbf{X\Sigma Y}^*, \quad \mathbf{B} = \mathbf{X}^*\mathbb{H}_s\mathbf{Y\Sigma}^{-1} = \mathbf{S\Lambda S}^{-1}, \quad \mathcal{V} = \mathbf{XS}$

$$\mathbf{\Sigma} = (\text{singular values of } \mathbb{H}) = \begin{bmatrix} 2.5089 & & \\ & 2.3231 & \\ & & 0.8173 \end{bmatrix}$$

$$\mathbf{\Lambda} = (\text{eigenvalues of } \mathbf{B}) = \begin{bmatrix} -0.3574 & & \\ & 0.1728 + 1.6738i & \\ & & 0.1728 - 1.6738i \end{bmatrix}$$

Hankel NLEVP Algorithm

from a

Systems Theory Perspective

A dynamical system associated with the Hankel method

Consider the dynamical system

$$\boxed{\begin{aligned} \mathbf{x}'(t) &= \boldsymbol{\Lambda}\,\mathbf{x}(t) + \mathbf{W}^*\mathbf{R}\,\mathbf{u}(t) \\ \mathbf{y}(t) &= \mathbf{L}^*\mathbf{V}\,\mathbf{x}(t). \end{aligned}}$$

The Laplace transform gives (for $\mathbf{x}(0) = \mathbf{0}$)

$$\begin{aligned} s(\mathcal{L}\mathbf{x})(s) &= \boldsymbol{\Lambda}(\mathcal{L}\mathbf{x})(s) + \mathbf{W}^*\mathbf{R}(\mathcal{L}\mathbf{u})(s) \\ (\mathcal{L}\mathbf{y})(s) &= \mathbf{L}^*\mathbf{V}(\mathcal{L}\mathbf{x})(s), \end{aligned}$$

which we write as

$$(\mathcal{L}\mathbf{y})(s) = \Big(\mathbf{L}^*\mathbf{V}(s\mathbf{I} - \boldsymbol{\Lambda})^{-1}\mathbf{W}^*\mathbf{R}\Big)(\mathcal{L}\mathbf{u})(s).$$

We call

$$\widetilde{\mathbf{H}}(s) := \mathbf{L}^*\mathbf{V}(s\mathbf{I} - \boldsymbol{\Lambda})^{-1}\mathbf{W}^*\mathbf{R}$$

the *transfer function* for the dynamical system.

A dynamical system associated with the Hankel method

$$\begin{aligned} \mathbf{x}'(t) &= \mathbf{\Lambda}\,\mathbf{x}(t) + \mathbf{W}^*\mathbf{R}\,\mathbf{u}(t) \\ \mathbf{y}(t) &= \mathbf{L}^*\mathbf{V}\,\mathbf{x}(t) \end{aligned}$$

$$\widetilde{\mathbf{H}}(z) = \mathbf{L}^*\mathbf{V}(z\mathbf{I} - \mathbf{\Lambda})^{-1}\mathbf{W}^*\mathbf{R}$$

Expand $\widetilde{\mathbf{H}}(z)$ about $z = \infty$:

$$\widetilde{\mathbf{H}}(z) = \sum_{k=0}^{\infty} \frac{1}{z^{k+1}} \mathbf{A}_k = \sum_{k=0}^{\infty} \frac{1}{z^{k+1}} \mathbf{L}^*\mathbf{V}\mathbf{\Lambda}^k\mathbf{W}^*\mathbf{R}.$$

In the language of systems theory, the Hankel contour integral algorithm seeks to *realize* the dynamical system from a finite set of its moments.

Realization Problem ($z = \infty$) for sampled system

Given the $2K$ moments

$$\mathbf{A}_k = \mathbf{L}^*\mathbf{V}\mathbf{\Lambda}^k\mathbf{W}^*\mathbf{R}, \qquad k = 0, \ldots, 2K-1,$$

recover the system matrices $\mathbf{\Lambda}$, $\mathbf{L}^*\mathbf{V}$, and $\mathbf{W}^*\mathbf{R}$.

A dynamical system associated with the Hankel method

Realization Problem ($z = \infty$) for sampled system

Given the $2K$ moments

$$\mathbf{A}_k = \mathbf{L}^*\mathbf{V}\mathbf{\Lambda}^k\mathbf{W}^*\mathbf{R}, \qquad k = 0, \dots, 2K-1,$$

recover the system matrices $\mathbf{\Lambda}$, $\mathbf{L}^*\mathbf{V}$, and $\mathbf{W}^*\mathbf{R}$.

- The rank condition $\mathrm{rank}(\mathcal{V}) = \mathrm{rank}(\mathcal{W}) = \mathrm{rank}(\mathbb{H}) = m$ amounts to *observability* and *reachability* of the $\widetilde{\mathbf{H}}(z)$ dynamical system.

- The basic algorithm

$$\mathbb{H} = \mathbf{X}\mathbf{\Sigma}\mathbf{Y}^*, \qquad \mathbf{B} = \mathbf{X}^*\mathbb{H}_s\mathbf{Y}\mathbf{\Sigma}^{-1} = \mathbf{S}\mathbf{\Lambda}\mathbf{S}^{-1}, \qquad \mathbf{L}^*\mathbf{V} = \mathbf{X}\mathbf{S}$$

  seeks an order-m *realization* of the dynamical system.

- This method resembles the Ho–Kalman method [1966], [De Schutter 2000] for constructing a *minimum realization* of the system.
  The SVD gives an especially appealing choice of basis transformation.
  (Cf. the Silverman realization algorithm [1971], another Hankel method.)

A dynamical system for the full linear part of $\mathbf{T}(z)$

Alternatively, consider the dynamical system *without sampling*

$$\begin{aligned} \mathbf{x}'(t) &= \boldsymbol{\Lambda}\,\mathbf{x}(t) + \mathbf{W}^*\,\mathbf{u}(t) \\ \mathbf{y}(t) &= \mathbf{V}\,\mathbf{x}(t) \end{aligned}$$

$$\mathbf{H}(z) = \mathbf{V}(z\mathbf{I} - \boldsymbol{\Lambda})^{-1}\mathbf{W}^*.$$

Expand this transfer function in (*unsampled*) moments $\mathbf{M}_k := \mathbf{V}\boldsymbol{\Lambda}^k\mathbf{W}^*$:

$$\mathbf{H}(z) = \sum_{k=0}^{\infty} \frac{1}{z^{k+1}}\mathbf{M}_k = \sum_{k=0}^{\infty} \frac{1}{z^{k+1}}\mathbf{V}\boldsymbol{\Lambda}^k\mathbf{W}^*.$$

Realization Problem ($z = \infty$) for full system

Given $2K$ tangentially sampled moments

$$\mathbf{L}^*\mathbf{M}_k = \mathbf{L}^*\mathbf{V}\boldsymbol{\Lambda}^k\mathbf{W}^*, \qquad \mathbf{M}_k\mathbf{R} = \mathbf{V}\boldsymbol{\Lambda}^k\mathbf{W}^*\mathbf{R},$$

for $k = 0, \ldots, 2K-1$, recover the system matrices $\boldsymbol{\Lambda}$, $\mathbf{V}$, and $\mathbf{R}$.

Realizing the full linear part of $\mathbf{T}(z)$

Realization Problem ($z = \infty$) for full system

Given $2K$ tangentially sampled moments

$$\mathbf{L}^*\mathbf{M}_k = \mathbf{L}^*\mathbf{V}\mathbf{\Lambda}^k\mathbf{W}^*, \qquad \mathbf{M}_k\mathbf{R} = \mathbf{V}\mathbf{\Lambda}^k\mathbf{W}^*\mathbf{R},$$

for $k = 0, \dots, 2K-1$, recover the system matrices $\mathbf{\Lambda}$, $\mathbf{V}$, and $\mathbf{R}$.

- Construct the one-sided data matrices

$$\mathbb{B} := \begin{bmatrix} \mathbf{L}^*\mathbf{M}_0 \\ \mathbf{L}^*\mathbf{M}_1 \\ \vdots \\ \mathbf{L}^*\mathbf{M}_{K-1} \end{bmatrix} \in \mathbb{C}^{\ell K \times n}, \qquad \mathbb{C} := \begin{bmatrix} \mathbf{M}_0\mathbf{R} & \mathbf{M}_1\mathbf{R} & \cdots & \mathbf{M}_{K-1}\mathbf{R} \end{bmatrix} \in \mathbb{C}^{n \times rK}.$$

- Compute the SVD of the usual Hankel matrix $\mathbb{H}$ (blocks $\mathbf{A}_k = \mathbf{L}^*\mathbf{M}_k\mathbf{R}$):

$$\mathbb{H} = \mathbf{X}\mathbf{\Sigma}\mathbf{Y}^*.$$

- Under suitable rank conditions cf. [Mayo, Antoulas 2007]:

$$\mathbf{H}(z) = \mathbb{C}\mathbf{Y}(z\mathbf{\Sigma} - \mathbf{X}^*\mathbb{H}_s\mathbf{Y})^{-1}\mathbf{X}^*\mathbb{B}.$$

Realizing the full linear part of $\mathbf{T}(z)$

$$\mathbb{B} := \begin{bmatrix} \mathbf{L}^*\mathbf{M}_0 \\ \mathbf{L}^*\mathbf{M}_1 \\ \vdots \\ \mathbf{L}^*\mathbf{M}_{K-1} \end{bmatrix} \in \mathbb{C}^{\ell K \times n}, \qquad \mathbb{C} := [\,\mathbf{M}_0\mathbf{R} \quad \mathbf{M}_1\mathbf{R} \quad \cdots \quad \mathbf{M}_{K-1}\mathbf{R}\,] \in \mathbb{C}^{n \times rK}.$$

$$\mathbb{H} = \mathbf{X}\boldsymbol{\Sigma}\mathbf{Y}^*.$$

Using the same diagonalization $\mathbf{X}^*\mathbb{H}_s\mathbf{Y}\boldsymbol{\Sigma}^{-1} = \mathbf{S}\boldsymbol{\Lambda}\mathbf{S}^{-1}$,

$$\begin{aligned} \mathbf{H}(z) &= \mathbb{C}\mathbf{Y}(z\boldsymbol{\Sigma} - \mathbf{X}^*\mathbb{H}_s\mathbf{Y})^{-1}\mathbf{X}^*\mathbb{B} \\ &= \big(\mathbb{C}\mathbf{Y}\boldsymbol{\Sigma}^{-1}\big)\big(z\mathbf{I} - \mathbf{X}^*\mathbb{H}_s\mathbf{Y}\boldsymbol{\Sigma}^{-1}\big)^{-1}\big(\mathbf{X}^*\mathbb{B}\big) \\ &= \big(\mathbb{C}\mathbf{Y}\boldsymbol{\Sigma}^{-1}\mathbf{S}\big)\big(z\mathbf{I} - \boldsymbol{\Lambda}\big)^{-1}\big(\mathbf{S}^{-1}\mathbf{X}^*\mathbb{B}\big) \end{aligned}$$

giving a realization of the full dynamical system with $\mathbf{H}(z) = \mathbf{V}(z - \boldsymbol{\Lambda})^{-1}\mathbf{W}^*$,

and access to the eigenvalues and *left and right eigenvectors* of $\mathbf{T}(z)$.

Rational NLEVP Algorithms

motivated by

System Realization

Realization based on data from a single point

$$\mathbf{H}(z) = \mathbf{V}(z\mathbf{I} - \mathbf{\Lambda})^{-1}\mathbf{W}^*$$

The contour integral method uses the expansion of $\mathbf{H}(z)$ at $z = \infty$:

$$\mathbf{H}(z) = \sum_{k=0}^{\infty} \frac{1}{z^{k+1}} \mathbf{M}_k = \sum_{k=0}^{\infty} \frac{1}{z^{k+1}} \mathbf{V}\mathbf{\Lambda}^k\mathbf{W}^*.$$

What if we prefer to expand about a finite point $z = \sigma \in \mathbb{C}$, $z \notin \Omega$?

$$\mathbf{H}(z) = \sum_{k=0}^{\infty} \left(\frac{1}{k!}\mathbf{H}^{(k)}(\sigma)\right)(z-\sigma)^k = \sum_{k=0}^{\infty} \mathbf{M}_k (z-\sigma)^k$$

Sample the $n \times n$ *moments*

$$\mathbf{M}_k = \frac{1}{k!}\mathbf{H}^{(k)}(\sigma) \in \mathbb{C}^{n\times n}$$

in the $\mathbf{L}$ and $\mathbf{R}$ directions via contour integration with special f; e.g.,

$$\mathbf{M}_k\mathbf{R} = \frac{1}{2\pi \mathrm{i}} \int_{\partial\Omega} \frac{(-1)^k}{(\sigma - z)^{k+1}} \mathbf{T}(z)^{-1}\mathbf{R}\,\mathrm{d}z.$$

Realization based on data from a single point

The rational interpolation/realization theory of Mayo and Antoulas [2007] recovers $\mathbf{H}(z)$. Arrange the data as

$$\mathbb{B} := \begin{bmatrix} \mathbf{L}^*\mathbf{M}_0 \\ \mathbf{L}^*\mathbf{M}_1 \\ \vdots \\ \mathbf{L}^*\mathbf{M}_{K-1} \end{bmatrix} \in \mathbb{C}^{\ell K \times n}, \qquad \mathbb{C} := [\, \mathbf{M}_0\mathbf{R} \;\; \mathbf{M}_1\mathbf{R} \;\; \cdots \;\; \mathbf{M}_{K-1}\mathbf{R} \,] \in \mathbb{C}^{n \times rK},$$

$$\mathbb{L}_0 := \begin{bmatrix} \mathbf{L}^*\mathbf{M}_0\mathbf{R} & \cdots & \mathbf{L}^*\mathbf{M}_{K-1}\mathbf{R} \\ \mathbf{L}^*\mathbf{M}_1\mathbf{R} & \cdots & \mathbf{L}^*\mathbf{M}_K\mathbf{R} \\ \vdots & \iddots & \vdots \\ \mathbf{L}^*\mathbf{M}_{K-1}\mathbf{R} & \cdots & \mathbf{L}^*\mathbf{M}_{2K-2}\mathbf{R} \end{bmatrix}, \qquad \mathbb{L} := \begin{bmatrix} \mathbf{L}^*\mathbf{M}_1\mathbf{R} & \cdots & \mathbf{L}^*\mathbf{M}_K\mathbf{R} \\ \mathbf{L}^*\mathbf{M}_2\mathbf{R} & \cdots & \mathbf{L}^*\mathbf{M}_{K+1}\mathbf{R} \\ \vdots & \iddots & \vdots \\ \mathbf{L}^*\mathbf{M}_K\mathbf{R} & \cdots & \mathbf{L}^*\mathbf{M}_{2K-1}\mathbf{R} \end{bmatrix},$$

and define $\mathbb{L}_s = \sigma\mathbb{L} - \mathbb{L}_0$.

Realization based on data from a single point

The rational interpolation/realization theory of Mayo and Antoulas [2007] recovers $\mathbf{H}(z)$. Arrange the data as

$$\mathbb{B} := \begin{bmatrix} \mathbf{L}^*\mathbf{M}_0 \\ \mathbf{L}^*\mathbf{M}_1 \\ \vdots \\ \mathbf{L}^*\mathbf{M}_{K-1} \end{bmatrix} \in \mathbb{C}^{\ell K \times n}, \qquad \mathbb{C} := [\,\mathbf{M}_0\mathbf{R} \;\; \mathbf{M}_1\mathbf{R} \;\; \cdots \;\; \mathbf{M}_{K-1}\mathbf{R}\,] \in \mathbb{C}^{n \times rK},$$

$$\mathbb{L}_0 := \begin{bmatrix} \mathbf{L}^*\mathbf{M}_0\mathbf{R} & \cdots & \mathbf{L}^*\mathbf{M}_{K-1}\mathbf{R} \\ \mathbf{L}^*\mathbf{M}_1\mathbf{R} & \cdots & \mathbf{L}^*\mathbf{M}_K\mathbf{R} \\ \vdots & \iddots & \vdots \\ \mathbf{L}^*\mathbf{M}_{K-1}\mathbf{R} & \cdots & \mathbf{L}^*\mathbf{M}_{2K-2}\mathbf{R} \end{bmatrix}, \qquad \mathbb{L} := \begin{bmatrix} \mathbf{L}^*\mathbf{M}_1\mathbf{R} & \cdots & \mathbf{L}^*\mathbf{M}_K\mathbf{R} \\ \mathbf{L}^*\mathbf{M}_2\mathbf{R} & \cdots & \mathbf{L}^*\mathbf{M}_{K+1}\mathbf{R} \\ \vdots & \iddots & \vdots \\ \mathbf{L}^*\mathbf{M}_K\mathbf{R} & \cdots & \mathbf{L}^*\mathbf{M}_{2K-1}\mathbf{R} \end{bmatrix},$$

and define $\mathbb{L}_s = \sigma\mathbb{L} - \mathbb{L}_0$.

With reduced SVD $\qquad \mathbb{L}_0 = -\mathbf{X}\mathbf{\Sigma}\mathbf{Y}^*$

we can recover the full transfer function as

$$\mathbf{H}(z) = \mathbb{C}\mathbf{Y}(\mathbf{X}^*\mathbb{L}_s\mathbf{Y} - z\mathbf{X}^*\mathbb{L}\mathbf{Y})^{-1}\mathbf{X}^*\mathbb{B}$$

under suitable rank conditions.

Incorporating tangential data at multiple points

So far we have recovered $\mathbf{H}(z) = \mathbf{V}(z\mathbf{I} - \mathbf{\Lambda})^{-1}\mathbf{W}^*$ using *high-order data* at a *single point* ($z = \infty$ or $z = \sigma$).

We can also incorporate *low-order data from multiple points.*

Incorporating tangential data at multiple points

So far we have recovered $\mathbf{H}(z) = \mathbf{V}(z\mathbf{I} - \mathbf{\Lambda})^{-1}\mathbf{W}^*$ using *high-order data* at a *single point* ($z = \infty$ or $z = \sigma$).

We can also incorporate *low-order data from multiple points.*

- Select r interpolation points and directions:

  right points, directions: $\theta_1, \ldots, \theta_r \in \mathbb{C} \setminus \Omega$, $\quad \mathbf{r}_1, \ldots, \mathbf{r}_r \in \mathbb{C}^n$

  left points, directions: $\mu_1, \ldots, \mu_r \in \mathbb{C} \setminus \Omega$, $\quad \boldsymbol{\ell}_1, \ldots, \boldsymbol{\ell}_r \in \mathbb{C}^n$

- We seek to interpolate left and right tangential data:

  right interpolation data: $\mathbb{f}_1 = \mathbf{H}(\theta_1)\mathbf{r}_1, \quad \ldots, \quad \mathbb{f}_r = \mathbf{H}(\theta_r)\mathbf{r}_r$

  left interpolation data: $\mathbb{g}_1^* = \boldsymbol{\ell}_1^*\mathbf{H}(\mu_1), \quad \ldots, \quad \mathbb{g}_r^* = \boldsymbol{\ell}_r^*\mathbf{H}(\mu_r)$

Realization Problem w/multi-point data for full system

Given $2r$ tangential samples $\{\mathbb{f}_j = \mathbf{H}(\theta_j)\mathbf{r}_j\}_{j=1}^r$ and $\{\mathbb{g}_j^* = \boldsymbol{\ell}_j^*\mathbf{H}(\mu_j)\}_{j=1}^r$, recover the system matrices $\mathbf{\Lambda}$, $\mathbf{V}$, and $\mathbf{W}$.

Compute tangential data via contour integration

Realization Problem w/multi-point data for full system

Given $2r$ tangential samples $\{\mathbb{f}_j = \mathbf{H}(\theta_j)\mathbf{r}_j\}_{j=1}^r$ and $\{\mathbb{g}_j^* = \boldsymbol{\ell}_j^*\mathbf{H}(\mu_j)\}_{j=1}^r$, recover the system matrices $\boldsymbol{\Lambda}$, $\mathbf{V}$, and $\mathbf{W}$.

Contour integration also gives access to this data (via the trapezoid rule):

$$\mathbb{f}_j := \mathbf{H}(\theta_j)\mathbf{r}_j = \frac{1}{2\pi \mathrm{i}} \int_{\partial\Omega} \frac{1}{\theta_j - z}\, \mathbf{T}(z)^{-1}\mathbf{r}_j \,\mathrm{d}z$$

$$\mathbb{g}_j^* := \boldsymbol{\ell}_j^*\mathbf{H}(\mu_j) = \frac{1}{2\pi \mathrm{i}} \int_{\partial\Omega} \frac{1}{\mu_j - z}\, \boldsymbol{\ell}_j^*\mathbf{T}(z)^{-1} \,\mathrm{d}z.$$

Note: $\mathbb{f}_j$ and $\mathbb{g}_j$ are *one-sided* data, like $\mathbf{M}_k\mathbf{R}$ and $\mathbf{L}^*\mathbf{M}_k$ in the Hankel setting.

Organize the data into Loewner matrices

Given	*right points, directions:*	$\theta_1, \ldots, \theta_r \in \mathbb{C}$,	$\mathbf{r}_1, \ldots, \mathbf{r}_r \in \mathbb{C}^n$
	left points, directions:	$\mu_1, \ldots, \mu_r \in \mathbb{C}$,	$\boldsymbol{\ell}_1, \ldots, \boldsymbol{\ell}_r \in \mathbb{C}^n$
Compute	*right interpolation data:*	$\mathbb{f}_1 = \mathbf{H}(\theta_1)\mathbf{r}_1, \quad \ldots,$	$\mathbb{f}_r = \mathbf{H}(\theta_r)\mathbf{r}_r$
	left interpolation data:	$\mathbb{g}_1^* = \boldsymbol{\ell}_1^*\mathbf{H}(\mu_1), \quad \ldots,$	$\mathbb{g}_r^* = \boldsymbol{\ell}_r^*\mathbf{H}(\mu_r)$

$$\mathbb{L} = \begin{bmatrix} \dfrac{\mathbb{g}_1^*\mathbf{r}_1 - \boldsymbol{\ell}_1^*\mathbb{f}_1}{\mu_1 - \theta_1} & \cdots & \dfrac{\mathbb{g}_1^*\mathbf{r}_r - \boldsymbol{\ell}_1^*\mathbb{f}_r}{\mu_1 - \theta_r} \\ \vdots & \ddots & \vdots \\ \dfrac{\mathbb{g}_r^*\mathbf{r}_1 - \boldsymbol{\ell}_r^*\mathbb{f}_1}{\mu_r - \theta_1} & \cdots & \dfrac{\mathbb{g}_r^*\mathbf{r}_r - \boldsymbol{\ell}_r^*\mathbb{f}_r}{\mu_r - \theta_r} \end{bmatrix} \in \mathbb{C}^{r \times r}$$

Loewner matrix

$$\mathbb{L}_s = \begin{bmatrix} \dfrac{\mu_1\mathbb{g}_1^*\mathbf{r}_1 - \theta_1\boldsymbol{\ell}_1^*\mathbb{f}_1}{\mu_1 - \theta_1} & \cdots & \dfrac{\mu_1\mathbb{g}_1^*\mathbf{r}_r - \theta_r\boldsymbol{\ell}_1^*\mathbb{f}_r}{\mu_1 - \theta_r} \\ \vdots & \ddots & \vdots \\ \dfrac{\mu_r\mathbb{g}_r^*\mathbf{r}_1 - \theta_1\boldsymbol{\ell}_r^*\mathbb{f}_1}{\mu_r - \theta_1} & \cdots & \dfrac{\mu_r\mathbb{g}_r^*\mathbf{r}_r - \theta_r\boldsymbol{\ell}_r^*\mathbb{f}_r}{\mu_r - \theta_r} \end{bmatrix} \in \mathbb{C}^{r \times r}$$

shifted Loewner matrix

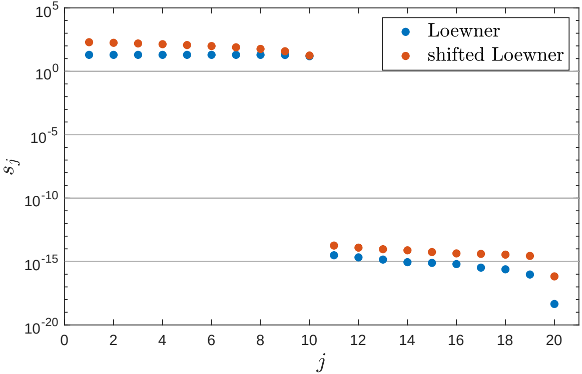
$$\mathbb{F} = [\,\mathbb{f}_1, \ldots, \mathbb{f}_r\,] \in \mathbb{C}^{n \times r}$$

$$\mathbb{G} = [\,\mathbb{g}_1, \ldots, \mathbb{g}_r\,] \in \mathbb{C}^{n \times r}$$

Rank of Loewner matrices reveal m

The rank of $\mathbb{L}$ and $\mathbb{L}_s$ reveal the number of eigenvalues m in Ω (the *McMillan degree* of $\mathbf{H}(z)$), but the singular values of the Loewner matrices depend on the choice of the interpolation points.

Example: system of degree $m = 10$; Loewner matrices have dimension $r = 20$.

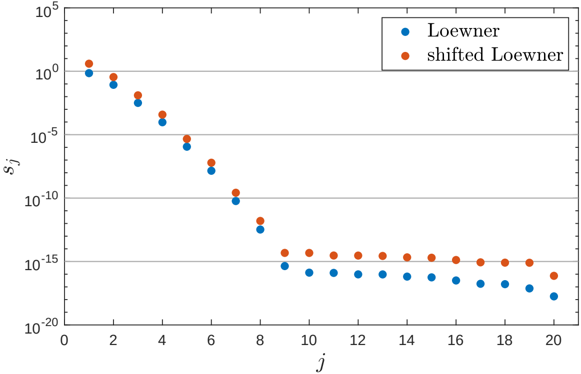


Singular values clearly reveal that the system has order $m = 10$.

Rank of Loewner matrices reveal m

The rank of $\mathbb{L}$ and $\mathbb{L}_s$ reveal the number of eigenvalues m in Ω (the *McMillan degree* of $\mathbf{H}(z)$), but the singular values of the Loewner matrices depend on the choice of the interpolation points.

Example: system of degree $m = 10$; Loewner matrices have dimension $r = 20$.



Same system, only interpolation points differ. What is the rank?

Two key Sylvester equations

$$\boldsymbol{\Theta} = \begin{bmatrix} \theta_1 & & \\ & \ddots & \\ & & \theta_r \end{bmatrix} \in \mathbb{C}^{r\times r}, \quad \mathbf{R} = \begin{bmatrix} \mathbf{r}_1 & \cdots & \mathbf{r}_r \end{bmatrix} \in \mathbb{C}^{n\times r}, \quad \mathbb{F} = \begin{bmatrix} \mathbb{f}_1 & \cdots & \mathbb{f}_r \end{bmatrix} \in \mathbb{C}^{n\times r}$$

$$\mathbf{M} = \begin{bmatrix} \mu_1 & & \\ & \ddots & \\ & & \mu_r \end{bmatrix} \in \mathbb{C}^{r\times r}, \quad \mathbf{L} = \begin{bmatrix} \boldsymbol{\ell}_1 & \cdots & \boldsymbol{\ell}_r \end{bmatrix} \in \mathbb{C}^{n\times r}, \quad \mathbb{G} = \begin{bmatrix} \mathbb{g}_1 & \cdots & \mathbb{g}_r \end{bmatrix} \in \mathbb{C}^{n\times r}$$

By construction, the Loewner matrices $\mathbb{L}$ and $\mathbb{L}_s$ solve the *Sylvester equations*

$$\mathbb{L}\boldsymbol{\Theta} - \mathbf{M}\mathbb{L} = \mathbf{L}^*\mathbb{F} - \mathbb{G}^*\mathbf{R}, \qquad \mathbb{L}_s\boldsymbol{\Theta} - \mathbf{M}\mathbb{L}_s = \mathbf{L}^*\mathbb{F}\boldsymbol{\Theta} - \mathbf{M}\mathbb{G}^*\mathbf{R}$$

which (typically) have right-hand sides with low rank.

$$\mathbb{L} = \begin{bmatrix} \dfrac{\mathbb{g}_1^*\mathbf{r}_1 - \boldsymbol{\ell}_1^*\mathbb{f}_1}{\mu_1 - \theta_1} & \cdots & \dfrac{\mathbb{g}_1^*\mathbf{r}_r - \boldsymbol{\ell}_1^*\mathbb{f}_r}{\mu_1 - \theta_r} \\ \vdots & \ddots & \vdots \\ \dfrac{\mathbb{g}_r^*\mathbf{r}_1 - \boldsymbol{\ell}_r^*\mathbb{f}_1}{\mu_r - \theta_1} & \cdots & \dfrac{\mathbb{g}_r^*\mathbf{r}_r - \boldsymbol{\ell}_r^*\mathbb{f}_r}{\mu_r - \theta_r} \end{bmatrix}, \quad \mathbb{L}_s = \begin{bmatrix} \dfrac{\mu_1\mathbb{g}_1^*\mathbf{r}_1 - \theta_1\boldsymbol{\ell}_1^*\mathbb{f}_1}{\mu_1 - \theta_1} & \cdots & \dfrac{\mu_1\mathbb{g}_1^*\mathbf{r}_r - \theta_r\boldsymbol{\ell}_1^*\mathbb{f}_r}{\mu_1 - \theta_r} \\ \vdots & \ddots & \vdots \\ \dfrac{\mu_r\mathbb{g}_r^*\mathbf{r}_1 - \theta_1\boldsymbol{\ell}_r^*\mathbb{f}_1}{\mu_r - \theta_1} & \cdots & \dfrac{\mu_r\mathbb{g}_r^*\mathbf{r}_r - \theta_r\boldsymbol{\ell}_r^*\mathbb{f}_r}{\mu_r - \theta_r} \end{bmatrix}$$

Two key Sylvester equations

$$\mathbf{\Theta} = \begin{bmatrix} \theta_1 & & \\ & \ddots & \\ & & \theta_r \end{bmatrix} \in \mathbb{C}^{r\times r}, \quad \mathbf{R} = \begin{bmatrix} \mathbf{r}_1 & \cdots & \mathbf{r}_r \end{bmatrix} \in \mathbb{C}^{n\times r}, \quad \mathbb{F} = \begin{bmatrix} \mathbb{f}_1 & \cdots & \mathbb{f}_r \end{bmatrix} \in \mathbb{C}^{n\times r}$$

$$\mathbf{M} = \begin{bmatrix} \mu_1 & & \\ & \ddots & \\ & & \mu_r \end{bmatrix} \in \mathbb{C}^{r\times r}, \quad \mathbf{L} = \begin{bmatrix} \ell_1 & \cdots & \ell_r \end{bmatrix} \in \mathbb{C}^{n\times r}, \quad \mathbb{G} = \begin{bmatrix} \mathbb{g}_1 & \cdots & \mathbb{g}_r \end{bmatrix} \in \mathbb{C}^{n\times r}$$

By construction, the Loewner matrices $\mathbb{L}$ and $\mathbb{L}_s$ solve the *Sylvester equations*

$$\mathbb{L}\mathbf{\Theta} - \mathbf{M}\mathbb{L} = \mathbf{L}^*\mathbb{F} - \mathbb{G}^*\mathbf{R}, \qquad \mathbb{L}_s\mathbf{\Theta} - \mathbf{M}\mathbb{L}_s = \mathbf{L}^*\mathbb{F}\mathbf{\Theta} - \mathbf{M}\mathbb{G}^*\mathbf{R}$$

which (typically) have right-hand sides with low rank.

- Penzl [1999] observed that (for Lyapunov equations)
  low-rank right-hand sides often imply rapid decay of singular values of $\mathbb{L}$.

- Decay bounds depends on the relative location of the spectra of $\mathbf{\Theta}$ and $\mathbf{M}$ [Penzl, 1999], [Antoulas, Sorensen, Zhou, 2002], [Baker, E., Sabino, 2015], [Beckermann & Townsend, 2017], ...

Insufficient data? Construct an interpolant.

Suppose we have $r < m$ measurements, and $\operatorname{rank}(\mathbb{L}) = \operatorname{rank}(\mathbb{L}_s) = r$.

Then construct the reduced model

$$\boxed{\mathbf{H}_r(z) := \mathbb{F}(\mathbb{L}_s - z\mathbb{L})^{-1}\mathbb{G}^* : \mathbb{C} \to \mathbb{C}^{n\times n}.}$$

The resulting degree-r model *interpolates the right and left data*:

$$\mathbf{H}_r(\theta_j)\mathbf{r}_j = \mathbf{H}(\theta_j)\mathbf{r}_j = \mathbb{f}_j, \qquad \boldsymbol{\ell}_j^*\mathbf{H}_r(\mu_j) = \boldsymbol{\ell}_j^*\mathbf{H}(\mu_j) = \mathbb{g}_j^*$$

for $j = 1, \ldots, r$.

While the poles of $\mathbf{H}_r(z)$ will not generally match the eigenvalues of $\mathbf{T}(z)$, they could provide helpful approximations, especially if the directions $\mathbf{r}_j$ and $\boldsymbol{\ell}_j$ approximate eigenvectors.

Sufficient data? Recover $\mathbf{H}(z)$.

Define the *reachability* and *observability* matrices

$$\mathcal{R} = \left[\ (\theta_1\mathbf{I} - \mathbf{\Lambda})^{-1}\mathbf{W}^*\mathbf{r}_1 \quad \cdots \quad (\theta_r\mathbf{I} - \mathbf{\Lambda})^{-1}\mathbf{W}^*\mathbf{r}_r\ \right] \in \mathbb{C}^{n\times r}, \qquad \mathcal{O} = \begin{bmatrix} \boldsymbol{\ell}_1^*\mathbf{V}(\mu_1\mathbf{I} - \mathbf{\Lambda})^{-1} \\ \vdots \\ \boldsymbol{\ell}_r^*\mathbf{V}(\mu_r\mathbf{I} - \mathbf{\Lambda})^{-1} \end{bmatrix} \in \mathbb{C}^{r\times n}$$

Rank condition

$$\operatorname{rank}(\mathcal{R}) = \operatorname{rank}(\mathcal{O}) = \operatorname{rank}(\mathbb{L}) = m \le r$$

Compute two reduced SVDs (rank m)

$$\left[\ \mathbb{L} \quad \mathbb{L}_s\ \right] = \mathbf{X}\mathbf{\Sigma}_1\mathbf{Y}_1^*, \qquad \begin{bmatrix} \mathbb{L} \\ \mathbb{L}_s \end{bmatrix} = \mathbf{X}_2\mathbf{\Sigma}_2\mathbf{Y}^*,$$

for $\mathbf{X}, \mathbf{Y} \in \mathbb{C}^{r\times m}$.

Construct reduced model

$$\widetilde{\mathbb{L}} := \mathbf{X}^*\mathbb{L}\mathbf{Y} \in \mathbb{C}^{m\times m} \qquad \widetilde{\mathbb{L}}_s := \mathbf{X}^*\mathbb{L}_s\mathbf{Y} \in \mathbb{C}^{m\times m}$$

$$\widetilde{\mathbb{F}} := \mathbb{F}\mathbf{Y} \in \mathbb{C}^{n\times m} \qquad \widetilde{\mathbb{G}} := \mathbb{G}\mathbf{X} \in \mathbb{C}^{n\times m}$$

Sufficient data? Recover $\mathbf{H}(z)$.

Rank condition

$$\operatorname{rank}(\mathcal{R}) = \operatorname{rank}(\mathcal{O}) = \operatorname{rank}(\mathbb{L}) = m \le r$$

Compute two reduced SVDs (rank m)

$$\begin{bmatrix} \mathbb{L} & \mathbb{L}_s \end{bmatrix} = \mathbf{X}\mathbf{\Sigma}_1\mathbf{Y}_1^*, \qquad \begin{bmatrix} \mathbb{L} \\ \mathbb{L}_s \end{bmatrix} = \mathbf{X}_2\mathbf{\Sigma}_2\mathbf{Y}^*,$$

for $\mathbf{X}, \mathbf{Y} \in \mathbb{C}^{r\times m}$.

Construct reduced model

$$\widetilde{\mathbb{L}} := \mathbf{X}^*\mathbb{L}\mathbf{Y} \in \mathbb{C}^{m\times m} \qquad \widetilde{\mathbb{L}}_s := \mathbf{X}^*\mathbb{L}_s\mathbf{Y} \in \mathbb{C}^{m\times m}$$

$$\widetilde{\mathbb{F}} := \mathbb{F}\mathbf{Y} \in \mathbb{C}^{n\times m} \qquad \widetilde{\mathbb{G}} := \mathbb{G}\mathbf{X} \in \mathbb{C}^{n\times m}$$

Recover the transfer function

$$\mathbf{H}(z) = \widetilde{\mathbb{F}}(\widetilde{\mathbb{L}}_s - z\widetilde{\mathbb{L}})^{-1}\widetilde{\mathbb{G}}^*$$

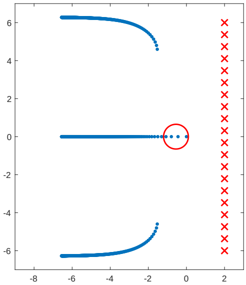
Reduction to $m \times m$ matrix eigenvalue problem

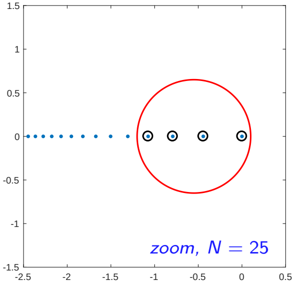
$$\mathbf{B} := \widetilde{\mathbb{L}}_s\widetilde{\mathbb{L}}^{-1} = (\mathbf{X}^*\mathbb{L}_s\mathbf{Y})(\mathbf{X}^*\mathbb{L}_s\mathbf{Y})^{-1} = \mathbf{S}\mathbf{\Lambda}\mathbf{S}^{-1},$$

analogous to the earlier $\mathbf{B} = \mathbf{X}^*\mathbb{H}_s\mathbf{Y}\mathbf{\Sigma}^{-1}$.

Computational examples

Example. $\mathbf{T}(\lambda) = \lambda\mathbf{I} - \mathbf{A} - e^{-\lambda}\mathbf{I}$,
where $\mathbf{A}$ is symmetric with $n = 1000$; eigenvalues of $\mathbf{A} = \{-1, -2, \dots, -n\}$.



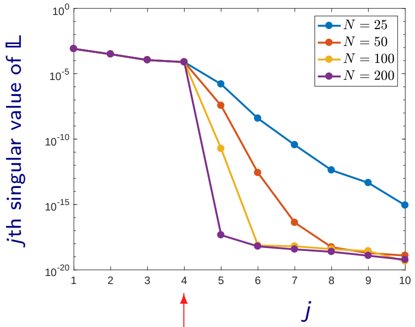


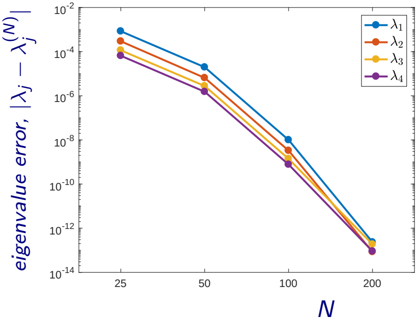
- $\cdot$ Eigenvalues of full $\mathbf{T}(\lambda)$
- $\times$ $r = 10$ left and right interpolation points interlaced in $2 + [-6i, 6i]$
- $\circ$ Eigenvalues of reduced $(m = 4)$ matrix pencil
- — Contour of integration $\partial\Omega$ (circle)

  Trapezoid rule uses $N = 25$, 50, 100, and 200 quadrature points

Computational examples

Example. $\mathbf{T}(\lambda) = \lambda\mathbf{I} - \mathbf{A} - e^{-\lambda}\mathbf{I}$,
where $\mathbf{A}$ is symmetric with $n = 1000$; eigenvalues of $\mathbf{A} = \{-1, -2, \ldots, -n\}$.



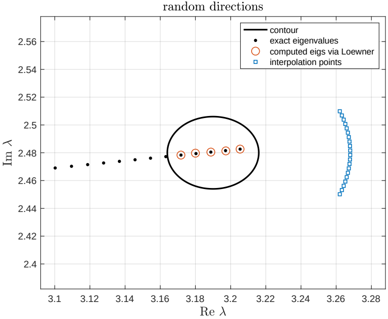


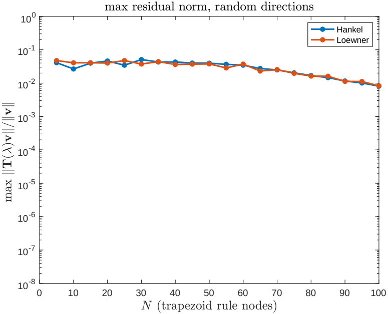
4 eigenvalues in Ω
$\Rightarrow \text{rank}(\mathbb{L}) = 4$

Cf. [Beyn 2012], [Güttel & Tisseur 2017] for $f(z) = z^k$.

Example: $\mathbf{T}(\lambda) = \lambda\mathbf{I} - e^{-\lambda}\mathbf{A}$

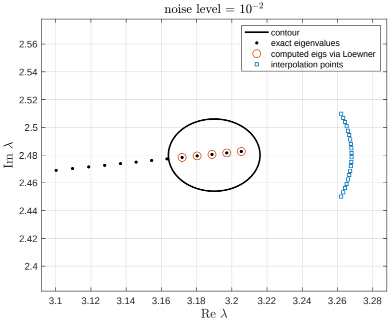
- $\mathbf{A}$ is symmetric with $n = 100$; eigenvalues of $\mathbf{A} = \{-1, -2, \ldots, -n\}$
- Compare $\max_j \|\mathbf{T}(\lambda_j)\mathbf{v}_j\|_2 / \|\mathbf{v}_j\|$ for $j = 1, 2, \ldots, 5$
- $m = 5$ eigenvalues in Ω, $r = 10$ left and right samples
- random search directions $\mathbf{R} = [\mathbf{r}_1, \ldots, \mathbf{r}_{10}]$ and $\mathbf{L} = [\boldsymbol{\ell}_1, \ldots, \boldsymbol{\ell}_{10}]$

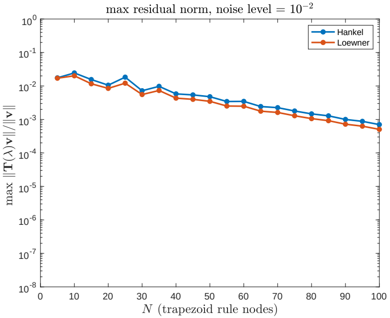




Example: $\mathbf{T}(\lambda) = \lambda\mathbf{I} - e^{-\lambda}\mathbf{A}$

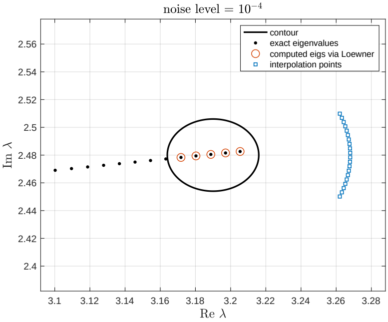
- $\mathbf{A}$ is symmetric with $n = 100$; eigenvalues of $\mathbf{A} = \{-1, -2, \dots, -n\}$
- Compare $\max_j \|\mathbf{T}(\lambda_j)\mathbf{v}_j\|_2 / \|\mathbf{v}_j\|$ for $j = 1, 2, \dots, 5$
- $m = 5$ eigenvalues in Ω, $r = 10$ left and right samples
- approximate eigenvector directions $\mathbf{R} = [\mathbf{r}_1, \dots, \mathbf{r}_{10}]$ and $\mathbf{L} = [\ell_1, \dots, \ell_{10}]$ noise level 10^{-2}

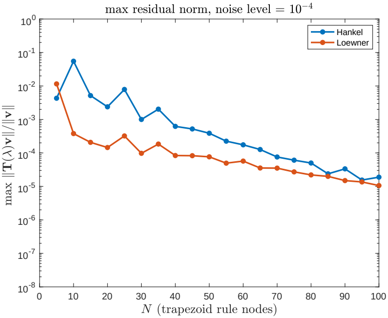




Example: $\mathbf{T}(\lambda) = \lambda\mathbf{I} - e^{-\lambda}\mathbf{A}$

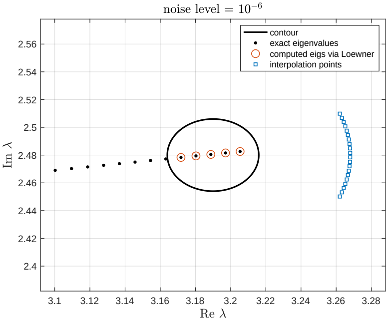
- $\mathbf{A}$ is symmetric with $n = 100$; eigenvalues of $\mathbf{A} = \{-1, -2, \ldots, -n\}$
- Compare $\max_j \|\mathbf{T}(\lambda_j)\mathbf{v}_j\|_2/\|\mathbf{v}_j\|$ for $j = 1, 2, \ldots, 5$
- $m = 5$ eigenvalues in Ω, $r = 10$ left and right samples
- approximate eigenvector directions $\mathbf{R} = [\mathbf{r}_1, \ldots, \mathbf{r}_{10}]$ and $\mathbf{L} = [\boldsymbol{\ell}_1, \ldots, \boldsymbol{\ell}_{10}]$ noise level 10^{-4}

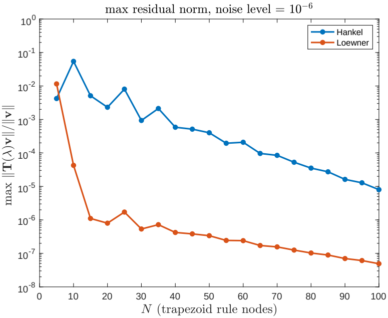




Example: $\mathbf{T}(\lambda) = \lambda\mathbf{I} - e^{-\lambda}\mathbf{A}$

- $\mathbf{A}$ is symmetric with $n = 100$; eigenvalues of $\mathbf{A} = \{-1, -2, \ldots, -n\}$
- Compare $\max_j \|\mathbf{T}(\lambda_j)\mathbf{v}_j\|_2/\|\mathbf{v}_j\|$ for $j = 1, 2, \ldots, 5$
- $m = 5$ eigenvalues in Ω, $r = 10$ left and right samples
- approximate eigenvector directions $\mathbf{R} = [\mathbf{r}_1, \ldots, \mathbf{r}_{10}]$ and $\mathbf{L} = [\ell_1, \ldots, \ell_{10}]$ noise level 10^{-6}





Conclusion

Contour integral algorithms are a rich and promising class of techniques for solving nonlinear eigenvalue problems, especially when one knows a target region Ω containing a modest number of eigenvalues.

We have described how these recent developments in numerical linear algebra connect to adjacent developments in systems realization theory.

This connection suggests several natural ways to expand the class of contour integration methods, suggesting avenues for new algorithm designs.

Conclusion

This connection suggests several natural ways to expand the class of contour integration methods, suggesting avenues for new algorithm designs.

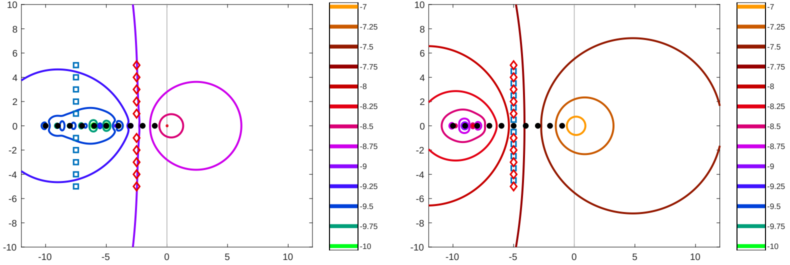
Numerous questions remain to be explored.

- How should we optimally choose the number of sampling directions, r?
- How do the sampling directions $\mathbf{r}_1, \ldots, \mathbf{r}_r$ and $\boldsymbol{\ell}_1, \ldots, \boldsymbol{\ell}_r$ affect convergence?
- How does convergence of the quadrature rule depend on:
  – the choice of interpolation points θ_j and μ_j ?
  – the distance of the contour from eigenvalues outside Ω ?
- How does the numerical rank of $\mathbb{L}$ depend on the quadrature accuracy, interpolation points ?

  This question has close connections to Loewner system realization with noisy measurements. Recent work by [Hokanson], [Drmac & Peherstorfer], [E. & Ionita], [Gosea & Antoulas], ...

Pseudospectra of Loewner pencils

Pole sensitivity depends on the location/ partitioning of interpolation points. . .



[E. & Ionita, arXiv 1910.12153]