Nonlinear Eigenvalue Problems: Interpolatory Algorithms and Transient Dynamics

Mark Embree · Virginia Tech

with
Serkan Gugercin
Jonathan Baker, Michael Brennan, Alexander Grimm
Virginia Tech

SIAM Conference on Applied Linear Algebra
Hong Kong · May 2018

Thanks to: International Linear Algebra Society (ILAS)
US National Science Foundation DMS-1720257
Rational interpolation for nlevps

Rational / Loewner techniques for nonlinear eigenvalue problems, motivated by algorithms from model reduction.

- Structure Preserving Rational Interpolation
- Data-Driven Rational Interpolation Matrix Pencils
- Minimal Realization via Rational Contour Integrals

transients for delay equations

Scalar delay equations: a case-study for how one can apply pseudospectra techniques to analyze the transient behavior of a dynamical system.

- *Finite dimensional nonlinear* problem
  ⇒ *infinite dimensional linear* problem
- Pseudospectral theory applies to the linear problem, *but the choice of norm is important*
<table>
<thead>
<tr>
<th>Problem</th>
<th>Equation</th>
<th># Eigenvalues</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard eigenvalue problem</td>
<td>((A - \lambda I)v = 0)</td>
<td>(n)</td>
</tr>
<tr>
<td>Generalized eigenvalue problem</td>
<td>((A - \lambda E)v = 0)</td>
<td>(n)</td>
</tr>
<tr>
<td>Quadratic eigenvalue problem</td>
<td>((K + \lambda D + \lambda^2 M)v = 0)</td>
<td>(2n)</td>
</tr>
<tr>
<td>Polynomial eigenvalue problem</td>
<td>((\sum_{k=0}^{d} \lambda^k A_k)v = 0)</td>
<td>(dn)</td>
</tr>
<tr>
<td>Nonlinear eigenvalue problem</td>
<td>((\sum_{k=0}^{d} f_k(\lambda)A_k)v = 0)</td>
<td>(\infty)</td>
</tr>
</tbody>
</table>
### Nonlinear Eigenvalue Problems: The Final Frontier?

<table>
<thead>
<tr>
<th>Problem</th>
<th>Equation</th>
<th>Typical # Eigenvalues</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard Eigenvalue Problem</td>
<td>((A - \lambda I)v = 0)</td>
<td>(n)</td>
</tr>
<tr>
<td>Generalized Eigenvalue Problem</td>
<td>((A - \lambda E)v = 0)</td>
<td>(n)</td>
</tr>
<tr>
<td>Quadratic Eigenvalue Problem</td>
<td>((K + \lambda D + \lambda^2 M)v = 0)</td>
<td>(2n)</td>
</tr>
<tr>
<td>Polynomial Eigenvalue Problem</td>
<td>((\sum_{k=0}^{d} \lambda^k A_k)v = 0)</td>
<td>(dn)</td>
</tr>
<tr>
<td>Nonlinear Eigenvalue Problem</td>
<td>((\sum_{k=0}^{d} f_k(\lambda)A_k)v = 0)</td>
<td>(\infty)</td>
</tr>
<tr>
<td>Nonlinear Eigenvector Problem</td>
<td>(F(\lambda, v) = 0)</td>
<td>(\infty)</td>
</tr>
</tbody>
</table>
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:

See, e.g., [Michiels & Niculescu 2007]
Nonlinear eigenvalue problems have classical roots, but now form a fast-moving field with many excellent resources and new algorithms.

- Helpful surveys:
  - Mehrmann & Voss, *GAMM*, [2004]
  - Voss, *Handbook of Linear Algebra*, [2014]

- Software:
  - NLEVP test collection [Betcke, Higham, Mehrmann, Schröder, Tisseur 2013]
  - SLEPC contains NLEVP algorithm implementations [Roman et al.]

- Many algorithms based on Newton’s method, rational approximation, linearization, contour integration, projection, etc.

- Infinite dimensional nonlinear spectral problems are even more subtle:
  - [Appell, De Pascale, Vignoli 2004] give *seven distinct definitions* of the spectrum.
Rational Interpolation

Algorithms

for

Nonlinear Eigenvalue Problems
Rational interpolation problem.

Given points \( \{z_j\}_{j=1}^{2r} \subset \mathbb{C} \) and data \( \{f_j \equiv f(z_j)\}_{j=1}^{2r} \), find a rational function \( R(z) = \frac{p(z)}{q(z)} \) of type \( (r - 1, r - 1) \) such that

\[
R(z_j) = f_j.
\]
Rational interpolation problem.
Given points \( \{z_j\}_{j=1}^{2r} \subset \mathbb{C} \) and data \( \{f_j \equiv f(z_j)\}_{j=1}^{2r} \), find a rational function \( R(z) = \frac{p(z)}{q(z)} \) of type \( (r - 1, r - 1) \) such that
\[
R(z_j) = f_j.
\]

Given Lagrange basis functions \( \ell_j(z) = \prod_{k=1 \atop k \neq j}^{r}(z - z_k) \) and nodal polynomial \( \ell(z) = \prod_{k=1}^{r}(z - z_k) \),
\[
R(z) = \frac{p(z)}{q(z)} = \frac{\sum_{j=1}^{r} \beta_j \ell_j(z)}{\sum_{j=1}^{r} w_j \ell_j(z)} = \frac{\sum_{j=1}^{r} \beta_j \ell_j(z)}{\ell(z)} = \frac{\sum_{j=1}^{r} \beta_j}{\sum_{j=1}^{r} \frac{w_j}{z - z_j}}
\]

barycentric form
Lagrange basis: \( \ell_j(z) = \prod_{k=1 \atop k \neq j}^r (z - z_k) \)

\[
R(z) = \frac{p(z)}{q(z)} = \frac{\sum_{j=1}^r \beta_j \ell_j(z)}{\sum_{j=1}^r w_j \ell_j(z)} = \frac{\sum_{j=1}^r \frac{\beta_j}{z - z_j}}{\sum_{j=1}^r \frac{w_j}{z - z_j}}
\]
rational interpolation: barycentric perspective

Lagrange basis: \( \ell_j(z) = \prod_{k=1}^{r} (z - z_k) \)

\[
R(z) = \frac{p(z)}{q(z)} = \sum_{j=1}^{r} \beta_j \ell_j(z) = \sum_{j=1}^{r} \frac{\beta_j}{z - z_j} = \sum_{j=1}^{r} \frac{w_j}{z - z_j}
\]

▶ Fix \( \{\beta_j = f_j w_j\}_{j=1}^{r} \) to interpolate at \( z_1, \ldots, z_r \): \( R(z_j) = f_j \).
rational interpolation: barycentric perspective

Lagrange basis: $\ell_j(z) = \prod_{k=1 \atop k \neq j}^r (z - z_k)$

$$R(z) = \frac{p(z)}{q(z)} = \frac{\sum_{j=1}^r \beta_j \ell_j(z)}{\sum_{j=1}^r w_j \ell_j(z)} = \frac{\sum_{j=1}^r \beta_j}{\sum_{j=1}^r w_j}$$

- Fix $\{\beta_j = f_j w_j\}_{j=1}^r$ to interpolate at $z_1, \ldots, z_r$: $R(z_j) = f_j$.
- Determine $w_1, \ldots, w_r$ to interpolate at $z_{r+1}, \ldots, z_{2r}$:

$$R(z_k) = \frac{\sum_{j=1}^r \frac{f_j w_j}{z_k - z_j}}{\sum_{j=1}^r \frac{w_j}{z_k - z_j}} = f_k \implies \sum_{j=1}^r \frac{f_j w_j}{z_k - z_j} = \sum_{j=1}^r \frac{f_k w_j}{z_k - z_j}$$
rational interpolation: barycentric perspective

- Fix \( \{\beta_j = f_j w_j\}_{j=1}^r \) to interpolate at \( z_1, \ldots, z_r \): \( r(z_j) = f_j \).
- Determine \( w_1, \ldots, w_r \) to interpolate at \( z_{r+1}, \ldots, z_{2r} \):

\[
R(z_k) = f_k \implies \sum_{j=1}^r \frac{f_j w_j}{z_k - z_j} = \sum_{j=1}^r \frac{f_k w_j}{z_k - z_j} \implies \sum_{j=1}^r \frac{f_j - f_k}{z_j - z_k} w_j = 0.
\]
Fix \( \{ \beta_j = f_j w_j \}_{j=1}^r \) to interpolate at \( z_1, \ldots, z_r \): \( r(z_j) = f_j \).

Determine \( w_1, \ldots, w_r \) to interpolate at \( z_{r+1}, \ldots, z_{2r} \):

\[
R(z_k) = f_k \implies \sum_{j=1}^r \frac{f_j w_j}{z_k - z_j} = \sum_{j=1}^r \frac{f_k w_j}{z_k - z_j} \implies \sum_{j=1}^r \frac{f_j - f_k}{z_j - z_k} w_j = 0.
\]

\[
\begin{bmatrix}
\frac{f_1 - f_{r+1}}{z_1 - z_{r+1}} & \frac{f_2 - f_{r+1}}{z_2 - z_{r+1}} & \cdots & \frac{f_r - f_{r+1}}{z_r - z_{r+1}} \\
\frac{f_1 - f_{r+2}}{z_1 - z_{r+2}} & \frac{f_2 - f_{r+2}}{z_2 - z_{r+2}} & \cdots & \frac{f_r - f_{r+2}}{z_r - z_{r+2}} \\
\vdots & \vdots & \ddots & \vdots \\
\frac{f_1 - f_{2r}}{z_1 - z_{2r}} & \frac{f_2 - f_{2r}}{z_2 - z_{2r}} & \cdots & \frac{f_r - f_{2r}}{z_r - z_{2r}}
\end{bmatrix}
\begin{bmatrix}
w_1 \\
w_2 \\
\vdots \\
w_r
\end{bmatrix}
= \begin{bmatrix}
0 \\
0 \\
\vdots \\
0
\end{bmatrix}
\]

\textit{Loewner matrix, } \mathbb{L}
Fix \( \{ \beta_j = f_j w_j \}_{j=1}^r \) to interpolate at \( z_1, \ldots, z_r \): \( r(z_j) = f_j \).

Determine \( w_1, \ldots, w_r \) to interpolate at \( z_{r+1}, \ldots, z_{2r} \):

\[
R(z_k) = f_k \implies \sum_{j=1}^r \frac{f_j w_j}{z_k - z_j} = \sum_{j=1}^r \frac{f_k w_j}{z_k - z_j} \implies \sum_{j=1}^r \frac{f_j - f_k}{z_j - z_k} w_j = 0.
\]

\[
\begin{bmatrix}
\frac{f_1 - f_{r+1}}{z_1 - z_{r+1}} & \frac{f_2 - f_{r+1}}{z_2 - z_{r+1}} & \ldots & \frac{f_r - f_{r+1}}{z_r - z_{r+1}} \\
\frac{f_1 - f_{r+2}}{z_1 - z_{r+2}} & \frac{f_2 - f_{r+2}}{z_2 - z_{r+2}} & \ldots & \frac{f_r - f_{r+2}}{z_r - z_{r+2}} \\
\vdots & \vdots & \ddots & \vdots \\
\frac{f_1 - f_{2r}}{z_1 - z_{2r}} & \frac{f_2 - f_{2r}}{z_2 - z_{2r}} & \ldots & \frac{f_r - f_{2r}}{z_r - z_{2r}}
\end{bmatrix}
\begin{bmatrix}
w_1 \\
w_2 \\
\vdots \\
w_r
\end{bmatrix}
= \begin{bmatrix}
0 \\
0 \\
\vdots \\
0
\end{bmatrix}
\]

*Loewner matrix, \( L \)*

Barycentric rational interpolation algorithm [Antoulas & Anderson [1986]]

AAA (Adaptive Antoulas–Anderson) Method [Nakatsukasa, Sète, Trefethen, 2016]
The rational interpolant $R(z)$ to $f$ at $z_1, \ldots, z_{2r}$ can also be formulated in state-space form using Loewner matrix techniques.

$$R(z) = c (\mathbb{I}_s - z\mathbb{L})^{-1} b,$$

where $c = [f_{r+1}, \ldots, f_{2r}]$, $b = [f_1, \ldots, f_r]^T$ and

$$\begin{bmatrix}
\frac{z_1 f_1 - z_{r+1} f_{r+1}}{z_1 - z_{r+1}} & \ldots & \frac{z_r f_r - z_{r+1} f_{r+1}}{z_r - z_{r+1}} \\
\vdots & \ddots & \vdots \\
\frac{z_1 f_1 - z_{2r} f_{2r}}{z_1 - z_{2r}} & \ldots & \frac{z_r f_r - z_{2r} f_{2r}}{z_r - z_{2r}}
\end{bmatrix}, \quad \begin{bmatrix}
\frac{f_1 - f_{r+1}}{z_1 - z_{r+1}} & \ldots & \frac{f_r - f_{r+1}}{z_r - z_{r+1}} \\
\vdots & \ddots & \vdots \\
\frac{f_1 - f_{2r}}{z_1 - z_{2r}} & \ldots & \frac{f_r - f_{2r}}{z_r - z_{2r}}
\end{bmatrix}.$$

*shifted Loewner matrix, $\mathbb{I}_s$*  
*Loewner matrix, $\mathbb{L}$*

- State space formulation proposed by Mayo & Antoulas [2007]
- Natural approach for handling *tangential interpolation for vector data*
- For details, applications, and extensions, see [Antoulas, Lefteriu, Ionita 2017]
**Scenario:** \( T(\lambda) \in \mathbb{C}^{n \times n} \) has *large dimension* \( n \).

**Goal:** Reduce dimension of \( T(\lambda) \) *but maintain the nonlinear structure.* Smaller problem will be more amenable to dense nonlinear eigensolvers.

**Method:** Rational tangential interpolation of \( T(\lambda) \)^{-1} at \( r \) points, directions.

**Iteratively Corrected Rational Interpolation method**
approach one: structure preserving rational interpolation

Scenario: \( T(\lambda) \in \mathbb{C}^{n \times n} \) has large dimension \( n \).

Goal: Reduce dimension of \( T(\lambda) \) but maintain the nonlinear structure. Smaller problem will be more amenable to dense nonlinear eigensolvers.

Method: Rational tangential interpolation of \( T(\lambda)^{-1} \) at \( r \) points, directions.

Iteratively Corrected Rational Interpolation method

- Pick \( r \) interpolation points \( \{z_j\}_{j=1}^r \) and interpolation directions \( \{w_j\}_{j=1}^r \).
**Scenario:** \( T(\lambda) \in \mathbb{C}^{n \times n} \) has **large dimension** \( n \).

**Goal:** Reduce dimension of \( T(\lambda) \) **but maintain the nonlinear structure.**
Smaller problem will be more amenable to dense nonlinear eigensolvers.

**Method:** Rational tangential interpolation of \( T(\lambda)^{-1} \) at \( r \) points, directions.

**Iteratively Corrected Rational Interpolation method**

- Pick \( r \) **interpolation points** \( \{z_j\}_{j=1}^r \) and **interpolation directions** \( \{w_j\}_{j=1}^r \).

- Construct a basis for projection (cf. **shift-invert Arnoldi**):

  \[
  U = \text{orth}\left([T(z_1)^{-1}w_1 \ T(z_2)^{-1}w_2 \ \cdots \ T(z_r)^{-1}w_r]\right) \in \mathbb{C}^{n \times r}.
  \]
**Scenario:** \( T(\lambda) \in \mathbb{C}^{n \times n} \) has *large dimension* \( n \).

**Goal:** Reduce dimension of \( T(\lambda) \) *but maintain the nonlinear structure.* Smaller problem will be more amenable to dense nonlinear eigensolvers.

**Method:** Rational tangential interpolation of \( T(\lambda)^{-1} \) at \( r \) points, directions.

**Iteratively Corrected Rational Interpolation method**

- Pick \( r \) *interpolation points* \( \{z_j\}_{j=1}^r \) and *interpolation directions* \( \{w_j\}_{j=1}^r \).

- Construct a basis for projection (cf. *shift-invert Arnoldi*):
  \[
  U = \text{orth}([T(z_1)^{-1}w_1 \quad T(z_2)^{-1}w_2 \quad \cdots \quad T(z_r)^{-1}w_r]) \in \mathbb{C}^{n \times r}.
  \]

- Form the reduced-dimension nonlinear system:
  \[
  T_r(\lambda) := U^*T(\lambda)U \in \mathbb{C}^{r \times r}.
  \]
approach one: structure preserving rational interpolation

Scenario: \( T(\lambda) \in \mathbb{C}^{n \times n} \) has large dimension \( n \).

Goal: Reduce dimension of \( T(\lambda) \) but maintain the nonlinear structure. Smaller problem will be more amenable to dense nonlinear eigensolvers.

Method: Rational tangential interpolation of \( T(\lambda)^{-1} \) at \( r \) points, directions.

Iteratively Corrected Rational Interpolation method

- Pick \( r \) interpolation points \( \{z_j\}_{j=1}^r \) and interpolation directions \( \{w_j\}_{j=1}^r \).

- Construct a basis for projection (cf. shift-invert Arnoldi):
  \[
  U = \text{orth}( [T(z_1)^{-1}w_1 \ T(z_2)^{-1}w_2 \ \cdots \ T(z_r)^{-1}w_r ] ) \in \mathbb{C}^{n \times r}.
  \]

- Form the reduced-dimension nonlinear system:
  \[
  T_r(\lambda) := U^*T(\lambda)U \in \mathbb{C}^{r \times r}.
  \]

- Compute the spectrum of \( T_r(\lambda) \) and use its eigenvalues and eigenvectors to update \( \{z_j\}_{j=1}^r \) and \( \{w_j\}_{j=1}^r \), and repeat.
approach one: structure preserving rational interpolation

The choice of projection subspace $\text{Ran}(U)$ delivers the key *interpolation property*.

**Interpolation Theorem.**

Provided $z_j \notin \sigma(T) \cup \sigma(T_r)$ for all $j = 1, \ldots, r$,

\[ T(z_j)^{-1} w_j = U T_r(z_j)^{-1} U^* w_j. \]

The choice of projection subspace $\text{Ran}(U)$ delivers the key *interpolation property*.

**Interpolation Theorem.**
Provided $z_j \notin \sigma(T) \cup \sigma(T_r)$ for all $j = 1, \ldots, r$,

$$T(z_j)^{-1}w_j = UT_r(z_j)^{-1}U^*w_j.$$ 


**Illustration.** As for all orthogonal projection methods:

$$T(\lambda) = f_0(\lambda) A_0 + f_1(\lambda) A_1 + f_2(\lambda) A_2$$

$$T_r(\lambda) = f_0(\lambda) U^*A_0U + f_1(\lambda) U^*A_1U + f_2(\lambda) U^*A_2U$$
approach one: structure preserving rational interpolation

The choice of projection subspace \( \text{Ran}(U) \) delivers the key interpolation property.

**Interpolation Theorem.**
Provided \( z_j \not\in \sigma(T) \cup \sigma(T_r) \) for all \( j = 1, \ldots, r \),
\[
T(z_j)^{-1}w_j = UT_r(z_j)^{-1}U^*w_j.
\]


**Illustration.** As for all orthogonal projection methods:
\[
\begin{align*}
T(\lambda) &= f_0(\lambda)A_0 + f_1(\lambda)A_1 + f_2(\lambda)A_2 \\
T_r(\lambda) &= f_0(\lambda)U^*A_0U + f_1(\lambda)U^*A_1U + f_2(\lambda)U^*A_2U
\end{align*}
\]

- The nonlinear functions \( f_j \) remain intact: *the structure is preserved.*
- The coefficients \( A_j \in \mathbb{C}^{n \times n} \) are compressed to \( U^*A_jU \in \mathbb{C}^{r \times r} \).
- Contrast: [Lietaert, Pérez, Vandereycken, Meerbergen 2018+] apply AAA approximation to \( f_j(\lambda) \), leave coefficient matrices intact.
approach one: structure preserving rational interpolation

Example 1. \( T(\lambda) = \lambda I - A - e^{-\lambda}I, \)

where \( A \) is symmetric with \( n = 1000; \) eigenvalues of \( A = \{-1, -2, \ldots, -n\}. \)

- Eigenvalues of full \( T(\lambda) \)
  - Interpolation points \( \{z_j\} \)

\( r = 16 \) used at each cycle (new points = real eigenvalues of \( T_r(\lambda) \))

initial \( \{z_j\} \) uniformly distributed on \([-10i, 10i]\), \( \{w_j\} \) selected randomly
Example 1. $T(\lambda) = \lambda I - A - e^{-\lambda} I$,
where $A$ is symmetric with $n = 1000$; eigenvalues of $A = \{-1, -2, \ldots, -n\}$.

- Eigenvalues of full $T(\lambda)$
- Interpolation points $\{z_j\}$
- Eigenvalues of reduced $T_r(\lambda)$
  
  $r = 16$ used at each cycle (new points = real eigenvalues of $T_r(\lambda)$)

initial $\{z_j\}$ uniformly distributed on $[-10i, 10i]$, $\{w_j\}$ selected randomly
Example 1. $T(\lambda) = \lambda I - A - e^{-\lambda}I,$
where $A$ is symmetric with $n = 1000$; eigenvalues of $A = \{-1, -2, \ldots, -n\}$. 

- **Eigenvalues of full $T(\lambda)$**
  - Interpolation points $\{z_j\}$
  - Eigenvalues of reduced $T_r(\lambda)$
    
    $r = 16$ used at each cycle (new points = real eigenvalues of $T_r(\lambda)$)
    
    initial $\{z_j\}$ uniformly distributed on $[-10i, 10i]$, $\{w_j\}$ selected randomly
Example 1. $T(\lambda) = \lambda I - A - e^{-\lambda}I$,
where $A$ is symmetric with $n = 1000$; eigenvalues of $A = \{-1, -2, \ldots, -n\}$.

- Eigenvalues of full $T(\lambda)$
  - Interpolation points $\{z_j\}$
  - Eigenvalues of reduced $T_r(\lambda)$
    
    $r = 16$ used at each cycle (new points $=$ real eigenvalues of $T_r(\lambda)$)
    
    initial $\{z_j\}$ uniformly distributed on $[-10i, 10i]$, $\{w_j\}$ selected randomly
Example 1. $T(\lambda) = \lambda I - A - e^{-\lambda} I$, where $A$ is symmetric with $n = 1000$; eigenvalues of $A = \{-1, -2, \ldots, -n\}$.

- Eigenvalues of full $T(\lambda)$
  - Interpolation points $\{z_j\}$
- Eigenvalues of reduced $T_r(\lambda)$
  - $r = 16$ used at each cycle (new points $=$ real eigenvalues of $T_r(\lambda)$)
  - initial $\{z_j\}$ uniformly distributed on $[-10i, 10i]$, $\{w_j\}$ selected randomly
Example 1. \( T(\lambda) = \lambda I - A - e^{-\lambda I} \),
where \( A \) is symmetric with \( n = 1000 \); eigenvalues of \( A = \{-1, -2, \ldots, -n\} \).

- Eigenvalues of full \( T(\lambda) \)
- Eigenvalues of reduced \( T_r(\lambda) \) at the final cycle
- Final interpolation points \( \{z_j\} \)
  \( r = 16 \) used at each cycle (new points = real eigenvalues of \( T_r(\lambda) \))
  initial \( \{z_j\} \) uniformly distributed on \([-10i, 10i] \), \( \{w_j\} \) selected randomly
Example 1. $\mathbf{T}(\lambda) = \lambda \mathbf{I} - \mathbf{A} - e^{-\lambda} \mathbf{I}$, where $\mathbf{A}$ is symmetric with $n = 1000$; eigenvalues of $\mathbf{A} = \{-1, -2, \ldots, -n\}$.

- Eigenvalues of full $\mathbf{T}(\lambda)$
  - Eigenvalues of reduced $\mathbf{T}_r(\lambda)$ at the final cycle
  - Final interpolation points $\{z_j\}$
    - $r = 16$ used at each cycle (new points = real eigenvalues of $\mathbf{T}_r(\lambda)$)
    - initial $\{z_j\}$ uniformly distributed on $[-10i, 10i]$, $\{w_j\}$ selected randomly
Example 2. $T(\lambda) = \lambda I - A - e^{-\lambda} I$, 
$A$ is symmetric with $n = 1000$; eigenvalues of $A = \{-1/n, -4/n, \ldots, -n\}$.

- Eigenvalues of full $T(\lambda)$
  - Eigenvalues of reduced $T_r(\lambda)$ at the final cycle
  - Final interpolation points $\{z_j\}$
    
    $r = 16$ used at each cycle (new points = real eigenvalues of $T_r(\lambda)$)
    
    initial $\{z_j\}$ uniformly distributed on $[-10i, 10i]$, $\{w_j\}$ selected randomly
Example 2. \( T(\lambda) = \lambda I - A - e^{-\lambda} I \),
\( A \) is symmetric with \( n = 1000 \); eigenvalues of \( A = \{-1/n, -4/n, \ldots, -n\} \).

- Eigenvalues of full \( T(\lambda) \)
- Eigenvalues of reduced \( T_r(\lambda) \) at the final cycle
- Final interpolation points \( \{z_j\} \)
  \( r = 16 \) used at each cycle (new points = real eigenvalues of \( T_r(\lambda) \))
- Initial \( \{z_j\} \) uniformly distributed on \([-10i, 10i]\), \( \{w_j\} \) selected randomly

convergence of \( r/2 \) rightmost \((z_j, w_j)\) pairs
approach two: data-driven rational interpolation

**Scenario:** $T(\lambda) \in \mathbb{C}^{n \times n}$ has large dimension $n$.

**Goal:** Obtain a small *linear matrix pencil* that *interpolates* the nonlinear eigenvalue problem. Smaller problem requires no further linearization.

**Method:** Data-driven rational interpolation of $T(\lambda)^{-1}$.

Data-Driven Rational Interpolation Matrix Pencil method
approach two: data-driven rational interpolation

Scenario: \( T(\lambda) \in \mathbb{C}^{n \times n} \) has large dimension \( n \).

Goal: Obtain a small linear matrix pencil that interpolates the nonlinear eigenvalue problem. Smaller problem requires no further linearization.

Method: Data-driven rational interpolation of \( T(\lambda)^{-1} \).

Data-Driven Rational Interpolation Matrix Pencil method

- Specify interpolation data:
  - left points, directions: \( z_1, \ldots, z_r \in \mathbb{C}, \quad w_1, \ldots, w_r \in \mathbb{C}^n \)
  - right points, directions: \( z_{r+1}, \ldots, z_{2r} \in \mathbb{C}, \quad w_{r+1}, \ldots, w_{2r} \in \mathbb{C}^n \)
approach two: data-driven rational interpolation

Scenario: \( \mathbf{T}(\lambda) \in \mathbb{C}^{n \times n} \) has large dimension \( n \).

Goal: Obtain a small linear matrix pencil that interpolates the nonlinear eigenvalue problem. Smaller problem requires no further linearization.

Method: Data-driven rational interpolation of \( \mathbf{T}(\lambda)^{-1} \).

Data-Driven Rational Interpolation Matrix Pencil method

▶ Specify interpolation data:

left points, directions: \( z_1, \ldots, z_r \in \mathbb{C}, \ w_1, \ldots, w_r \in \mathbb{C}^n \)

right points, directions: \( z_{r+1}, \ldots, z_{2r} \in \mathbb{C}, \ w_{r+1}, \ldots, w_{2r} \in \mathbb{C}^n \)

▶ Construct \( \mathbf{T}_r(\lambda)^{-1} : = \mathbf{C}_r (\mathbf{A}_r - \lambda \mathbf{E}_r)^{-1} \mathbf{B}_r \) to tangentially interpolate \( \mathbf{T}(\lambda)^{-1} \).

Tangential Interpolation Theorem. Provided \( z_j \notin \sigma(\mathbf{T}) \cup \sigma(\mathbf{T}_r) \),

\[
\begin{align*}
\mathbf{w}_j^T \mathbf{T}(z_j)^{-1} &= \mathbf{w}_j^T \mathbf{T}_r(z_j)^{-1}, \quad j = 1, \ldots, r; \\
\mathbf{T}(z_j)^{-1} \mathbf{w}_j &= \mathbf{T}_r(z_j)^{-1} \mathbf{w}_j, \quad j = r + 1, \ldots, 2r.
\end{align*}
\]
Given left points, directions: \( z_1, \ldots, z_r \in \mathbb{C}, \ w_1, \ldots, w_r \in \mathbb{C}^n \)
right points, directions: \( z_{r+1}, \ldots, z_{2r} \in \mathbb{C}, \ w_{r+1}, \ldots, w_{2r} \in \mathbb{C}^n \)

Define left interpolation data: \( f_1 = T(z_1)^{-T}w_1, \ldots, f_r = T(z_r)^{-T}w_r \)
right interpolation data: \( f_{r+1} = T(z_{r+1})^{-1}w_{r+1}, \ldots, f_{2r} = T(z_{2r})^{-1}w_{2r} \)
**approach two: data-driven rational interpolation**

Given  
*left points, directions:*  
\( z_1, \ldots, z_r \in \mathbb{C}, \quad w_1, \ldots, w_r \in \mathbb{C}^n \)

*right points, directions:*  
\( z_{r+1}, \ldots, z_{2r} \in \mathbb{C}, \quad w_{r+1}, \ldots, w_{2r} \in \mathbb{C}^n \)

Define  
*left interpolation data:*  
\( f_1 = T(z_1)^{-T}w_1, \quad \ldots, \quad f_r = T(z_r)^{-T}w_r \)

*right interpolation data:*  
\( f_{r+1} = T(z_{r+1})^{-1}w_{r+1}, \quad \ldots, \quad f_{2r} = T(z_{2r})^{-1}w_{2r} \)

*Order-\( r \) (linear) model:*  
\( T_r(z)^{-1} = C_r(A_r - zE_r)^{-1}B_r \)
### Approach Two: Data-Driven Rational Interpolation

Given **left points, directions:**

\[ z_1, \ldots, z_r \in \mathbb{C}, \quad w_1, \ldots, w_r \in \mathbb{C}^n \]

**right points, directions:**

\[ z_{r+1}, \ldots, z_{2r} \in \mathbb{C}, \quad w_{r+1}, \ldots, w_{2r} \in \mathbb{C}^n \]

Define **left interpolation data:**

\[ f_1 = T(z_1)^{-T}w_1, \quad \ldots, \quad f_r = T(z_r)^{-T}w_r \]

**right interpolation data:**

\[ f_{r+1} = T(z_{r+1})^{-1}w_{r+1}, \quad \ldots, \quad f_{2r} = T(z_{2r})^{-1}w_{2r} \]

**Order-\( r \) (linear) model:**

\[ T_r(z)^{-1} = C_r(A_r - zE_r)^{-1}B_r \]

#### Coefficients

- **\( C_r \):**
  \[
  C_r = \begin{bmatrix} f_{r+1}, \ldots, f_{2r} \end{bmatrix}
  \]

- **\( A_r \):**
  \[
  A_r = \begin{bmatrix}
  \frac{z_1 f_1^T w_{r+1} - z_{r+1}^T f_{r+1}}{z_1 - z_{r+1}} & \ldots & \frac{z_r f_r^T w_{r+1} - z_{r+1}^T f_{r+1}}{z_r - z_{r+1}} \\
  0 & \ddots & 0 \\
  \frac{z_1 f_1^T w_{2r} - z_{2r}^T f_{2r}}{z_1 - z_{2r}} & \ldots & \frac{z_r f_r^T w_{2r} - z_{2r}^T f_{2r}}{z_r - z_{2r}}
  \end{bmatrix}
  \]

- **\( E_r \):**
  \[
  E_r = \begin{bmatrix}
  \frac{f_1^T w_{r+1} - w_1^T f_{r+1}}{z_1 - z_{r+1}} & \ldots & \frac{f_r^T w_{r+1} - w_r^T f_{r+1}}{z_r - z_{r+1}} \\
  0 & \ddots & 0 \\
  \frac{f_1^T w_{2r} - w_1^T f_{2r}}{z_1 - z_{2r}} & \ldots & \frac{f_r^T w_{2r} - w_r^T f_{2r}}{z_r - z_{2r}}
  \end{bmatrix}
  \]

- **\( B_r \):**
  \[
  B_r = \begin{bmatrix} f_1, \ldots, f_r \end{bmatrix}^T
  \]

**Shifted Loewner**

**Loewner**
approach two: data-driven rational interpolation

Given

left points, directions: \( z_1, \ldots, z_r \in \mathbb{C}, \quad w_1, \ldots, w_r \in \mathbb{C}^n \)

right points, directions: \( z_{r+1}, \ldots, z_{2r} \in \mathbb{C}, \quad w_{r+1}, \ldots, w_{2r} \in \mathbb{C}^n \)

Define

left interpolation data: \( f_1 = T(z_1)^{-T}w_1, \quad \ldots, \quad f_r = T(z_r)^{-T}w_r \)

right interpolation data: \( f_{r+1} = T(z_{r+1})^{-1}w_{r+1}, \quad \ldots, \quad f_{2r} = T(z_{2r})^{-1}w_{2r} \)

Rank-\( r \) (linear) model: \( T_r(z)^{-1} = C_r(A_r - zE_r)^{-1}B_r \)

\[
\begin{align*}
T(z)^{-1} & \approx C_r \quad \text{linear matrix pencil} \\
(A_r - zE_r)^{-1} & B_r
\end{align*}
\]
**Example.** $T(\lambda) = \lambda I - A - e^{-\lambda I}$,

where $A$ is symmetric with $n = 1000$; eigenvalues of $A = \{-1, -2, \ldots, -n\}$.

- Eigenvalues of full $T(\lambda)$
  - Eigenvalues of reduced matrix pencil $A_r - zE_r$
    - $r = 40$ interpolation points used, uniform in interval $[-80i, 80i]$  

*Hermite interpolation variant that only uses $r$ distinct interpolation points.*

Interpolation directions from smallest singular values of $T(z_j)$. 

---

*zoom out*
approach three: Loewner realization via contour integration

**Scenario:** Seek all eigenvalues of $T(\lambda) \in \mathbb{C}^{n \times n}$ in a prescribed region $\Omega$ of $\mathbb{C}$.

**Goal:** Use Keldysh’s Theorem to isolate interesting part of $T(\lambda)$ in $\Omega$.

**Method:** Contour integration of $T(\lambda)$ against *rational test functions.*
Loewner matrix will reveal number of eigenvalues in $\Omega$.

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

$$T(z)^{-1} = V(zI - \Lambda)^{-1}U^* + R(z),$$

- $V = [v_1 \cdots v_m]$, $U = [u_1 \cdots u_m]$, $\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_m)$, $u_j^*T'(\lambda_j)v_j = 1$;
- $R(z)$ is analytic in $\Omega$. 

**Scenario:** Seek all eigenvalues of \( T(\lambda) \in \mathbb{C}^{n \times n} \) in a prescribed region \( \Omega \) of \( \mathbb{C} \).

**Goal:** Use Keldysh’s Theorem to isolate interesting part of \( T(\lambda) \) in \( \Omega \).

**Method:** Contour integration of \( T(\lambda) \) against rational test functions. Loewner matrix will reveal number of eigenvalues in \( \Omega \).

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

\[
T(z)^{-1} = V(zI - \Lambda)^{-1}U^* + R(z),
\]

- \( V = [v_1 \cdots v_m] \), \( U = [u_1 \cdots u_m] \), \( \Lambda = \text{diag}(\lambda_1, \ldots, \lambda_m) \), \( u_j^*T'(\lambda_j)v_j = 1 \);
- \( R(z) \) is analytic in \( \Omega \).

\[
T(z)^{-1} = H(z) + R(z)
\]

where \( H(z) := V(zI - \Lambda)^{-1}U^* \) is a transfer function for a linear system.
Theorem [Keldysh 1951]. Suppose $T(z)$ has $m$ eigenvalues $\lambda_1, \ldots, \lambda_m$ (counting multiplicity) in the region $\Omega \subset \mathbb{C}$, all semi-simple. Then

$$T(z)^{-1} = V(zI - \Lambda)^{-1}U^* + R(z),$$

- $V = [v_1 \cdots v_m]$, $U = [U_1 \cdots U_m]$, $\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_m)$, $U^*T'(\lambda_j)v_j = 1$;
- $R(z)$ is analytic in $\Omega$.

\[
\begin{align*}
T(z)^{-1} = & \quad V \\
& (zI - \Lambda)^{-1} \\
& U^* \\
& + \quad R(z)
\end{align*}
\]

$h(z) := V(zI - \Lambda)^{-1}U^*$

$n \times n$ linear system, order $m$:

$m$ poles in $\Omega$
A family of algorithms use the fact that, by the Cauchy integral formula,

$$\frac{1}{2\pi i} \int_{\partial \Omega} f(z) T(z)^{-1} \, dz = V f(\Lambda) U^*,$$

see [Asakura, Sakurai, Tadano, Ikegami, Kimura 2009], [Beyn 2012], [Yokota & Sakurai 2013], etc., building upon contour integral eigensolvers for matrix pencils [Sakurai & Sugiura 2003], [Polizzi 2009], etc.

These algorithms use \( f(z) = z^k \) for \( k = 0, 1, \ldots \) to produce Hankel matrix pencils.
approach three: Loewner realization via contour integration

\[ T(z)^{-1} = H(z) + R(z) \]

where \( H(z) : V(zI - \Lambda)^{-1}U^* \) is a transfer function for a linear system.

A family of algorithms use the fact that, by the Cauchy integral formula,

\[ \frac{1}{2\pi i} \int_{\partial\Omega} f(z) T(z)^{-1} \, dz = V f(\Lambda) U^*; \]

see [Asakura, Sakurai, Tadano, Ikegami, Kimura 2009], [Beyn 2012], [Yokota & Sakurai 2013], etc., building upon contour integral eigensolvers for matrix pencils [Sakurai & Sugiura 2003], [Polizzi 2009], etc.

These algorithms use \( f(z) = z^k \) for \( k = 0, 1, \ldots \) to produce Hankel matrix pencils.

Key observation: If we use \( f(z) = 1/(z_j - z) \) for \( z_j \) exterior to \( \Omega \), we obtain

\[ \frac{1}{2\pi i} \int_{\partial\Omega} \frac{1}{z_j - z} T(z)^{-1} \, dz = V(z_jI - \Lambda)^{-1}U^* = H(z_j). \]

Contour integrals yield measurements of the linear system with the desired eigenvalues.
approach three: Loewner realization via contour integration

Minimal Realization via Rational Contour Integrals for $m$ eigenvalues

Let $r \geq m$, and select interpolation points and directions:

- **left points, directions:** $z_1, \ldots, z_r \in \mathbb{C} \setminus \Omega$, $w_1, \ldots, w_r \in \mathbb{C}^n$
- **right points, directions:** $z_{r+1}, \ldots, z_{2r} \in \mathbb{C} \setminus \Omega$, $w_{r+1}, \ldots, w_{2r} \in \mathbb{C}^n$

Use contour integrals to compute the left and right interpolation data:

- **left interpolation data:** $f_1 = H(z_1)T_{w_1}, \ldots, f_r = H(z_r)T_{w_r}$
- **right interpolation data:** $f_{r+1} = H(z_{r+1})w_{r+1}, \ldots, f_{2r} = H(z_{2r})w_{2r}$

Construct Loewner and shifted Loewner matrices from this data, just as in the Data-Driven Rational Interpolation method:

$C_r = [f_{r+1}, \ldots, f_{2r}]$
$B_r = [f_1, \ldots, f_r]^T$
$A_r = $ shifted Loewner matrix
$E_r = $ Loewner matrix

If $r = m$, then $V(z_I - \Lambda)^{-1}U^* = C_r(A_r - zE_r)^{-1}B_r$: compute eigenvalues!

If $r > m$, use SVD truncation / minimum realization techniques to reduce dimension; cf. [Mayo & Antoulas 2007].
approach three: Loewner realization via contour integration

Minimal Realization via Rational Contour Integrals for $m$ eigenvalues

- Let $r \geq m$, and select interpolation points and directions:

  left points, directions: $z_1, \ldots, z_r \in \mathbb{C} \setminus \Omega$, $w_1, \ldots, w_r \in \mathbb{C}^n$

  right points, directions: $z_{r+1}, \ldots, z_{2r} \in \mathbb{C} \setminus \Omega$, $w_{r+1}, \ldots, w_{2r} \in \mathbb{C}^n$

- Use contour integrals to compute the left and right interpolation data:

  left interpolation data: $f_1 = H(z_1)^T w_1$, $\ldots$, $f_r = H(z_r)^T w_r$

  right interpolation data: $f_{r+1} = H(z_{r+1}) w_{r+1}$, $\ldots$, $f_{2r} = H(z_{2r}) w_{2r}$

  $H(z_j)w_j = \frac{1}{2\pi i} \int_{\partial \Omega} \frac{1}{z_j - z} T(z)^{-1} w_j \, dz$. 
approach three: Loewner realization via contour integration

Minimal Realization via Rational Contour Integrals for $m$ eigenvalues

- Let $r \geq m$, and select interpolation points and directions:
  
  **left points, directions:** $z_1, \ldots, z_r \in \mathbb{C} \setminus \Omega$, \quad $w_1, \ldots, w_r \in \mathbb{C}^n$
  
  **right points, directions:** $z_{r+1}, \ldots, z_{2r} \in \mathbb{C} \setminus \Omega$, \quad $w_{r+1}, \ldots, w_{2r} \in \mathbb{C}^n$

- Use contour integrals to compute the left and right interpolation data:
  
  **left interpolation data:** $f_1 = H(z_1)^T w_1$, \quad \ldots, \quad f_r = H(z_r)^T w_r$
  
  **right interpolation data:** $f_{r+1} = H(z_{r+1}) w_{r+1}$, \quad \ldots, \quad $f_{2r} = H(z_{2r}) w_{2r}$

  $$H(z_j) w_j = \frac{1}{2\pi i} \int_{\partial \Omega} \frac{1}{z_j - z} T(z)^{-1} w_j \, dz.$$ 

- Construct Loewner and shifted Loewner matrices from this data, just as in the Data-Driven Rational Interpolation method:

  $C_r = [f_{r+1}, \ldots, f_{2r}]$ \quad $B_r = [f_1, \ldots, f_r]^T$

  $A_r = \text{shifted Loewner matrix}$ \quad $E_r = \text{Loewner matrix}$

- If $r = m$, then

  $$V(z_I - \Lambda) - 1 U^* = C_r (A_r - z E_r)^{-1} B_r$$

- If $r > m$, use SVD truncation / minimum realization techniques to reduce dimension; cf. [Mayo & Antoulas 2007].
approach three: Loewner realization via contour integration

Minimal Realization via Rational Contour Integrals for $m$ eigenvalues

- Let $r \geq m$, and select interpolation points and directions:
  
  **left points, directions:** $z_1, \ldots, z_r \in \mathbb{C} \setminus \Omega$, $w_1, \ldots, w_r \in \mathbb{C}^n$
  
  **right points, directions:** $z_{r+1}, \ldots, z_{2r} \in \mathbb{C} \setminus \Omega$, $w_{r+1}, \ldots, w_{2r} \in \mathbb{C}^n$

- Use contour integrals to compute the left and right interpolation data:
  
  **left interpolation data:** $f_1 = H(z_1)^T w_1$, $\ldots$, $f_r = H(z_r)^T w_r$
  
  **right interpolation data:** $f_{r+1} = H(z_{r+1}) w_{r+1}$, $\ldots$, $f_{2r} = H(z_{2r}) w_{2r}$

  $H(z_j) w_j = \frac{1}{2\pi i} \int_{\partial \Omega} \frac{1}{z_j - z} T(z)^{-1} w_j \, dz.$

- Construct *Loewner* and *shifted Loewner* matrices from this data, just as in the Data-Driven Rational Interpolation method:

  $C_r = [f_{r+1}, \ldots, f_{2r}]$  \hspace{1cm} $B_r = [f_1, \ldots, f_r]^T$

  $A_r = \text{shifted Loewner matrix}$  \hspace{1cm} $E_r = \text{Loewner matrix}$

- If $r = m$, then $V(zI - \Lambda)^{-1} U^* = C_r (A_r - zE_r)^{-1} B_r$: compute eigenvalues!

  If $r > m$, use SVD truncation / minimum realization techniques to reduce dimension; cf. [Mayo & Antoulas 2007].
Example. \( T(\lambda) = \lambda I - A - e^{-\lambda}I \), where \( A \) is symmetric with \( n = 1000 \); eigenvalues of \( A = \{-1, -2, \ldots, -n\} \).

- Eigenvalues of full \( T(\lambda) \)
- 20 interpolation points in \( 2 + [-6i, 6i] \)
- Eigenvalues of minimal \( (m = 4) \) matrix pencil
- Contour of integration (circle)
  Trapezoid rule uses \( N = 25, 50, 100, \) and 200 interpolation points
Example. $T(\lambda) = \lambda I - A - e^{-\lambda} I$, where $A$ is symmetric with $n = 1000$; eigenvalues of $A = \{-1, -2, \ldots, -n\}$.

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

Cf. [Beyn 2012], [Güttel & Tisseur 2017] for $f(z) = z^k$. 
For rank detection for Loewner matrices, see [Hokanson 2018+].
Transient Dynamics
for
Dynamical Systems
with Delays

a case study of pseudospectral analysis
We often care about *eigenvalues* because we seek insight about *dynamics*. 
We often care about eigenvalues because we seek insight about dynamics.

Start with the simple scalar system

\[ x'(t) = \alpha x(t), \]

with solution

\[ x(t) = e^{t\alpha} x(0). \]

If \( \text{Re} \alpha < 0 \), then \( |x(t)| \to 0 \) monotonically as \( t \to \infty \).
We often care about eigenvalues because we seek insight about dynamics.

Now consider the $n$-dimensional system

$$x'(t) = Ax(t)$$

with solution

$$x(t) = e^{tA}x(0).$$

If $\text{Re}\lambda < 0$ for all $\lambda \in \sigma(A)$, then $\|x(t)\| \to 0$ asymptotically as $t \to \infty$, for some $t^* \in (0, \infty)$. 

\[
A = \begin{bmatrix}
-1 & 0 \\
100 & -2
\end{bmatrix}
\]
We often care about *eigenvalues* because we seek insight about *dynamics*.

Now consider the $n$-dimensional system

$$x'(t) = Ax(t)$$

with solution

$$x(t) = e^{tA}x(0).$$

If $\text{Re}\lambda < 0$ for all $\lambda \in \sigma(A)$, then $\|x(t)\| \to 0$ asymptotically as $t \to \infty$, but it is possible that $\|x(t_*)\| \gg \|x(0)\|$ for some $t_* \in (0, \infty)$. 

\[
A = \begin{bmatrix}
-1 & 0 \\
100 & -2
\end{bmatrix}
\]
why transients matter

Often the linear dynamical system \( \mathbf{x}'(t) = A\mathbf{x}(t) \) arises from linear stability analysis for a fixed point of a nonlinear system

\[
\mathbf{y}'(t) = \mathbf{F}(\mathbf{y}(t), t).
\]

For example,

\[
\mathbf{y}'(t) = A\mathbf{y}(t) + \frac{1}{20}\mathbf{y}(t)^2.
\]

In this example, linear transient growth feeds the nonlinearity. Such behavior can provide a mechanism for transition to turbulence in fluid flows; see, e.g., [Butler & Farrell 1992], [Trefethen et al. 1993].
detecting the potential for transient growth

One can draw insight about transient growth from the numerical range (field of values) and $\varepsilon$-pseudospectra of $A$:

$$\sigma_\varepsilon(A) = \{z \in \mathbb{C} : \|(zI - A)^{-1}\| > 1/\varepsilon\}$$

$$= \{z \in \mathbb{C} : z \in \sigma(A + E) \text{ for some } E \in \mathbb{C}^{n \times n} \text{ with } \|E\| < \varepsilon\}$$

For upper and lower bounds on $\|x(t)\|$, see [Trefethen & E. 2005], e.g.,

$$\sup_{t \geq 0} \|e^{tA}\| \geq \sup_{z \in \sigma_\varepsilon(A)} \frac{\text{Re } z}{\varepsilon}.$$  

If $\sigma_\varepsilon(A)$ extends more than $\varepsilon$ across the imaginary axis, $\|e^{tA}\|$ grows transiently.
detecting the potential for transient growth

One can draw insight about transient growth from the numerical range (field of values) and \( \varepsilon \)-pseudospectra of \( A \):

\[
\sigma_\varepsilon(A) = \{ z \in \mathbb{C} : \|(zI - A)^{-1}\| > 1/\varepsilon \} = \{ z \in \mathbb{C} : z \in \sigma(A + E) \text{ for some } E \in \mathbb{C}^{n \times n} \text{ with } \|E\| < \varepsilon \}
\]

For upper and lower bounds on \( \|x(t)\| \), see [Trefethen & E. 2005], e.g.,

\[
\sup_{t \geq 0} \|e^{tA}\| \geq \sup_{z \in \sigma_\varepsilon(A)} \frac{\text{Re } z}{\varepsilon}.
\]

If \( \sigma_\varepsilon(A) \) extends more than \( \varepsilon \) across the imaginary axis, \( \|e^{tA}\| \) grows transiently.

Pseudospectra can guarantee that some \( x(0) \) induce transient growth.
Two ways to look at pseudospectra

Two equivalent definitions give two distinct perspectives.

**perturbed eigenvalues**

\[ \sigma_\varepsilon(A) = \{ z \in \mathbb{C} : z \in \sigma(A + E) \text{ for some } E \in \mathbb{C}^{n \times n} \text{ with } \|E\| < \varepsilon \} \]

**norms of resolvents**

\[ \sigma_\varepsilon(A) = \{ z \in \mathbb{C} : \| (zI - A)^{-1} \| > 1/\varepsilon \} \]
two ways to look at pseudospectra

Two *equivalent* definitions give two distinct perspectives.

**perturbed eigenvalues**

\[
\sigma_\varepsilon(A) = \{ z \in \mathbb{C} : z \in \sigma(A + E) \text{ for some } E \in \mathbb{C}^{n \times n} \text{ with } \|E\| < \varepsilon \} 
\]

- \( \sigma_\varepsilon(A) \) contains the eigenvalues of all matrices with distance \( \varepsilon \) of \( A \).
- Ideal for assessing *asymptotic stability of uncertain systems*: Is some matrix near \( A \) unstable?
- Why consider all \( E \in \mathbb{C}^{n \times n} \)? *Structured pseudospectra* further restrict \( E \) (real, Toeplitz, etc.).
  [Hinrichsen & Pritchard], [Karow], [Rump]

**norms of resolvents**

\[
\sigma_\varepsilon(A) = \{ z \in \mathbb{C} : \| (zI - A)^{-1} \| > 1/\varepsilon \} 
\]

- \( \sigma_\varepsilon(A) \) is bounded by \( 1/\varepsilon \) level sets of the resolvent norm.
- Ideal for assessing *transient behavior of stable systems*: \( \|e^tA\| > 1 \) or \( \|A^k\| > 1 \)?
- Rooted in semigroup theory: based on the solution operator for the dynamical system; structure of \( A \) plays no role.

These perspectives match for \( x'(t) = Ax(t) \), but not for more complicated systems.
Two ways to look at pseudospectra

Two *equivalent* definitions give two distinct perspectives.

**perturbed eigenvalues**

\[ \sigma_{\varepsilon}(A) = \{ z \in \mathbb{C} : z \in \sigma(A + E) \text{ for some } E \in \mathbb{C}^{n \times n} \text{ with } \|E\| < \varepsilon \} \]

- \( \sigma_{\varepsilon}(A) \) contains the eigenvalues of all matrices with distance \( \varepsilon \) of \( A \).
- Ideal for assessing *asymptotic stability of uncertain systems*: Is some matrix *near* \( A \) unstable?
- Why consider all \( E \in \mathbb{C}^{n \times n} \)? *Structured pseudospectra* further restrict \( E \) (real, Toeplitz, etc.).

**norms of resolvents**

\[ \sigma_{\varepsilon}(A) = \{ z \in \mathbb{C} : \| (zI - A)^{-1} \| > 1/\varepsilon \} \]

- \( \sigma_{\varepsilon}(A) \) is bounded by \( 1/\varepsilon \) level sets of the resolvent norm.
- Ideal for assessing *transient behavior of stable systems*: \( \|e^{tA}\| > 1 \) or \( \|A^k\| > 1 \)?
- Rooted in semigroup theory: based on the solution operator for the dynamical system; structure of \( A \) plays no role.
## Two ways to look at pseudospectra

Two *equivalent* definitions give two distinct perspectives.

### Perturbed Eigenvalues

\[ \sigma_\varepsilon(A) = \{ z \in \mathbb{C} : z \in \sigma(A + E) \text{ for some } E \in \mathbb{C}^{n \times n} \text{ with } \|E\| < \varepsilon \} \]

- \( \sigma_\varepsilon(A) \) contains the eigenvalues of all matrices with distance \( \varepsilon \) of \( A \).
- Ideal for assessing *asymptotic stability of uncertain systems*: Is some matrix near \( A \) unstable?
- Why consider all \( E \in \mathbb{C}^{n \times n} \)? *Structured pseudospectra* further restrict \( E \) (real, Toeplitz, etc.). [Hinrichsen & Pritchard], [Karow], [Rump]

### Norms of Resolvents

\[ \sigma_\varepsilon(A) = \{ z \in \mathbb{C} : \| (zI - A)^{-1} \| > 1/\varepsilon \} \]

- \( \sigma_\varepsilon(A) \) is bounded by \( 1/\varepsilon \) level sets of the resolvent norm.
- Ideal for assessing *transient behavior of stable systems*: \( \| e^{tA} \| > 1 \) or \( \| A^k \| > 1 \)?
- Rooted in semigroup theory: based on the solution operator for the dynamical system; structure of \( A \) plays no role.

*These perspectives match for \( x'(t) = Ax(t) \), but not for more complicated systems.*
scalar delay equations and the nonlinear eigenvalue problem

We shall apply these ideas to explore the potential for *transient growth in solutions to stable delay differential equations.*

Solutions of scalar systems \( x'(t) = \alpha x(t) \) behave monotonically: \(|x(t)| = e^{t \operatorname{Re} \alpha} |x(0)| \). What about scalar delay equations?

\[
x'(t) = \alpha x(t) + \beta x(t - 1)
\]

Using the techniques seen earlier, we associate this system with the NLEVP

\[
(\lambda - \alpha)e^\lambda = \beta,
\]

with infinitely many eigenvalues given by branches of the Lambert-\( W \) function:

\[
\lambda_k = \alpha + W_k(\beta e^{-\alpha}).
\]
We shall apply these ideas to explore the potential for *transient growth in solutions to stable delay differential equations.*

Solutions of scalar systems \( x'(t) = \alpha x(t) \) behave monotonically: \( |x(t)| = e^{t \Re \alpha} |x(0)| \). What about scalar delay equations?

\[
\begin{align*}
\lambda_k &= \alpha + W_k(\beta e^{-\alpha}).
\end{align*}
\]
We shall apply these ideas to explore the potential for transient growth in solutions to stable delay differential equations. Solutions of scalar systems $x'(t) = \alpha x(t)$ behave monotonically: $|x(t)| = e^{t \text{Re} \alpha} |x(0)|$. What about scalar delay equations?

$$\{\lambda_k\} \text{ for } \begin{cases} \alpha = 3/4 \\ \beta = -1 \end{cases}$$

$x'(t) = \alpha x(t) + \beta x(t-1) \implies \lambda_k = \alpha + W_k(\beta e^{-\alpha})$. 
Conventional eigenvalue-based stability analysis reveals the \((\alpha, \beta)\) combinations that yield \textit{asymptotically stable} solutions.

Such \textit{stability charts} are standard tools for studying stability of parameter-dependent delay systems.
pseudospectra for nonlinear eigenvalue problems


Consider the nonlinear eigenvalue problem $T(\lambda)v = 0$ with

$$T(\lambda) = \sum_{j=1}^{m} f_j(\lambda) A_j.$$  

For $p, q \in [1, \infty]$ and weights $w_1, \ldots, w_m \in (0, \infty]$, define the norm

$$\|(E_1, \ldots, E_m)\|_{p,q} = \left\| \begin{bmatrix} w_1\|E_1\|_q & \ldots & \|E_m\|_q \end{bmatrix} \right\|_p.$$

Given this way of measuring a perturbation to $T(\lambda)$, [MGWN 2006] define

$$\sigma_\varepsilon(T) = \left\{ z \in \mathbb{C} : z \in \sigma \left( \sum_{j=1}^{m} f_j(\lambda) (A_j + E_j) \right) \text{ for some } E_1, \ldots, E_m \in \mathbb{C}^{n \times n} \text{ with } \|(E_1, \ldots, E_m)\|_{p,q} < \varepsilon \right\}.$$
pseudospectra for the scalar delay equation

\[ x'(t) = \alpha x(t) + \beta x(t - 1) \]

\[ T(\lambda) = \lambda - \alpha - \beta e^{-\lambda}. \]

MGWN $\varepsilon$-pseudospectra for $\alpha = \frac{3}{4}$ and $\beta = -1$, with perturbation norm given by $q \in [1, \infty]$ and $p = \infty$, and $w_1 = w_2 = 1$. 
pseudospectra for the scalar delay equation

\[ x'(t) = -x(t) + 0 \cdot x(t - 1) \]

\[ T(\lambda) = \lambda + 1 \]

\[ T(\lambda) = \lambda + 1 - 0 e^{-\lambda} \]

MGWN $\varepsilon$-pseudospectra with $p = \infty$: structure affects pseudospectra.
To better understand transient behavior, just integrate the differential equation:

\[
x'(t) = \alpha x(t) + \beta x(t - 1)
\]

history: \( x(t - 1) = u(t) \) for \( t \in [0, 1] \).
To better understand transient behavior, just integrate the differential equation:

\[ x'(t) = \alpha x(t) + \beta x(t - 1) \]

history: \( x(t - 1) = u(t) \) for \( t \in [0, 1] \).

Integrate

\[ x'(t) = \alpha x(t) + \beta u(t) \]

to get, for \( t \in [0, 1] \),

\[ x(t) = e^{t\alpha} x(0) + \beta \int_0^t e^{(t-s)\alpha} u(s) \, ds \]

\[ = e^{t\alpha} u(1) + \beta \int_0^t e^{(t-s)\alpha} u(s) \, ds. \]
To better understand transient behavior, just integrate the differential equation:

\[
x'(t) = \alpha x(t) + \beta x(t - 1)
\]

history: \( x(t - 1) = u(t) \) for \( t \in [0, 1] \).

Integrate

\[
x'(t) = \alpha x(t) + \beta u(t)
\]

to get, for \( t \in [0, 1] \),

\[
x(t) = e^{t\alpha} x(0) + \beta \int_0^t e^{(t-s)\alpha} u(s) \, ds
\]

\[
= e^{t\alpha} u(1) + \beta \int_0^t e^{(t-s)\alpha} u(s) \, ds.
\]

This operation maps the history \( u \) to the solution \( x \) for \( t \in [0, 1] \):

\[
u \in C([0, 1]) \quad \mapsto \quad x \in C([0, 1]).
\]
Define the solution operator $K : C[0, 1] \rightarrow C[0, 1]$ via

$$x(t) = (Ku)(t) = e^{t\alpha} u(1) + \beta \int_0^t e^{(t-s)\alpha} u(s) \, ds, \quad t \in [0, 1].$$
Define the solution operator \( K : C[0, 1] \to C[0, 1] \) via

\[
x(t) = (K u)(t) = e^{t \alpha} u(1) + \beta \int_0^t e^{(t-s) \alpha} u(s) \, ds,
\]
\( t \in [0, 1] \).

Define:

\[
x^{(0)} := u
\]

To advance \( t \) by 1 unit, apply \( K \):

\[
x^{(1)} := K x^{(0)}
\]

To advance \( t \) by 2 units, apply \( K^2 \):

\[
x^{(2)} := K x^{(1)} = K^2 x^{(0)}
\]

\[
\vdots
\]

To advance \( t \) by \( m \) units, apply \( K^m \):

\[
x^{(m)} := K x^{(m-1)} = K^m x^{(0)}
\]

\[
x_m = x(t) \big|_{t \in [m-1, m]}
\]

\( t \in [−1, 0] \)

\( t \in [0, 1] \)

\( t \in [1, 2] \)

\( t \in [m − 1, m] \)
the solution operator

Define the \textit{solution operator} $K : C[0, 1] \to C[0, 1]$ via

\[
x(t) = (Ku)(t) = e^{t\alpha}u(1) + \beta \int_0^t e^{(t-s)\alpha} u(s) \, ds, \quad t \in [0, 1].
\]

Define: \quad $x^{(0)} := u$

to advance $t$ by 1 unit, apply $K$: \quad $x^{(1)} := K x^{(0)}$
to advance $t$ by 2 units, apply $K^2$: \quad $x^{(2)} := K x^{(1)} = K^2 x^{(0)}$

\[\vdots\]
to advance $t$ by $m$ units, apply $K^m$: \quad $x^{(m)} := K x^{(m-1)} = K^m x^{(0)} \quad t \in [m-1, m]$

\[x_m = x(t)_{|t \in [m-1, m]} \]

to $t \in [-1, 0]$
to $t \in [0, 1]$
to $t \in [1, 2]$

View the delay system as a \textit{discrete-time dynamical system} over 1-unit time intervals:

\[x^{(m)} = K^m x^{(0)}.\]
We discretize the solution operator using a Chebyshev pseudospectral method based on [Trefethen 2000]; see [Bueler 2007], [Jarlebring 2008].

\[ x(t_j) \approx x_j := e^{t_j \alpha} u_0 + \sum_{k=0}^{N} \beta w_{j,k} u_k, \quad w_{j,k} := \int_0^{t_j} e^{(t_j-s) \alpha} \ell_k(s) \, ds \]

\[
\begin{bmatrix}
    x_0 \\
    x_1 \\
    \vdots \\
    x_N
\end{bmatrix} =
\begin{bmatrix}
    e^{t_0 \alpha} & 0 & \cdots & 0 \\
    e^{t_1 \alpha} & 0 & \cdots & 0 \\
    \vdots & \vdots & \ddots & \vdots \\
    e^{t_N \alpha} & 0 & \cdots & 0
\end{bmatrix}
\begin{bmatrix}
    u_0 \\
    u_1 \\
    \vdots \\
    u_N
\end{bmatrix}
+ \beta
\begin{bmatrix}
    w_{0,0} & w_{0,1} & \cdots & w_{0,N} \\
    w_{1,0} & w_{1,1} & \cdots & w_{1,N} \\
    \vdots & \vdots & \ddots & \vdots \\
    w_{N,0} & w_{N,1} & \cdots & w_{N,N}
\end{bmatrix}
\begin{bmatrix}
    u_0 \\
    u_1 \\
    \vdots \\
    u_N
\end{bmatrix}
\]

\[ E_N(\alpha) \quad \text{and} \quad W_N(\alpha) \]
We discretize the solution operator using a Chebyshev pseudospectral method based on [Trefethen 2000]; see [Bueler 2007], [Jarlebring 2008].

\[
x(t_j) \approx x_j := e^{t_j \alpha} u_0 + \sum_{k=0}^{N} \beta w_{j,k} u_k,
\]

\[
w_{j,k} := \int_0^{t_j} e^{(t_j-s) \alpha} \ell_k(s) \, ds
\]

\[
\begin{bmatrix}
x_0 \\
x_1 \\
\vdots \\
x_N
\end{bmatrix} =
\begin{bmatrix}
e^{t_0 \alpha} & 0 & \cdots & 0 \\
e^{t_1 \alpha} & 0 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
e^{t_N \alpha} & 0 & \cdots & 0
\end{bmatrix}
\begin{bmatrix}
u_0 \\
u_1 \\
\vdots \\
u_N
\end{bmatrix} + \beta
\begin{bmatrix}
w_{0,0} & w_{0,1} & \cdots & w_{0,N} \\
w_{1,0} & w_{1,1} & \cdots & w_{1,N} \\
\vdots & \vdots & \ddots & \vdots \\
w_{N,0} & w_{N,1} & \cdots & w_{N,N}
\end{bmatrix}
\begin{bmatrix}
u_0 \\
u_1 \\
\vdots \\
u_N
\end{bmatrix}
\]

\[
K_N := E_N(\alpha) + \beta W_N(\alpha)
\]

\[
x^{(1)} := K_N u
\]

\[
x^{(m)} := K_N^m u
\]
approaches to transient analysis of delay equations

- Jacob Stroh [2006], in a master’s thesis advised by Ed Bueler, computes $L^2$-pseudospectra of Chebyshev discretizations of the compact solution operator and considers nonnormality as a function of a time-varying coefficient in the delay term: *our approach follows closely.*

- Green & Wagenknecht [2006], in their paper about perturbation-based pseudospectra for delay equations, describe computing the pseudospectra of the generator for the solution semigroup as a way of gauging transient behavior; for relevant semigroup theory, see, e.g., [Engel & Nagel 2000].


- Solution operator approach converts a *finite dimensional nonlinear problem* into an *infinite dimensional linear problem*, akin to the *infinite Arnoldi algorithm* [Jarlebring, Meerbergen, Michiels 2010, 2012, 2014].
To study convergence, consider $\alpha = 0, \beta = -1$: $x'(t) = -x(t - 1)$.

**$\mu_j^{(N)}$: the $j$th largest magnitude eigenvalue of $K_N$**

**$e^{\lambda_j}$: $\lambda_j$ is the $j$th rightmost eigenvalue of the NLEVP**

We generally use $N = 64$ for our computations throughout what follows.
nonconvergence of the $L^2$ pseudospectra of the solution operator

Eigenvalues converge, but the $L^2[0,1]$ pseudospectra of $K_N$ do not: the departure from normality increases with $N$!
the problem with the $L^2$ norm

Problem: The $L^2(0,1)$ norm does not measure transient growth of $|x(t)|$.

One can easily find $u(x)$ such that $\|u\|_{L^2[0,1]} \ll 1$ but $\|x\|_{L^2[0,1]} \geq 1$.

Let $\alpha = 0$, $\beta = -1$: $x'(t) = -x(t-1) \implies x(t) = u(1) - \int_0^t u(s) \, ds$.
the problem with the $L^2$ norm

Problem: *The $L^2(0,1)$ norm does not measure transient growth of $|x(t)|$.*

One can easily find $u(x)$ such that $\|u\|_{L^2[0,1]} \ll 1$ but $\|x\|_{L^2[0,1]} \geq 1$.

Let $\alpha = 0$, $\beta = -1$: $x'(t) = -x(t - 1) \implies x(t) = u(1) - \int_0^t u(s) \, ds$. 
pseudospectra and transient growth of matrix powers

Since we care about the largest value \( |x(t)| \) can take, we should really study

\[
\|x^{(m)}\|_{L\infty},
\]

and thus the \( \varepsilon \)-pseudospectrum \( \sigma_\varepsilon(K_N) \) defined using the \( \infty \)-norm:

\[
\sigma_\varepsilon(K_N) := \{ z \in \mathbb{C} : \|(zI - K_N)^{-1}\|_\infty > 1/\varepsilon \} \\
:= \{ z \in \mathbb{C} : z \in \sigma(K_N + E) \text{ for some } E \in \mathbb{C}^{n \times n} \text{ with } \|E\|_\infty < \varepsilon \}. 
\]
Since we care about the largest value $|x(t)|$ can take, we should really study

$$\|x^{(m)}\|_{L^\infty},$$

and thus the $\varepsilon$-pseudospectrum $\sigma_\varepsilon(K_N)$ defined using the $\infty$-norm:

$$\sigma_\varepsilon(K_N) := \{z \in \mathbb{C} : \|(zI - K_N)^{-1}\|_\infty > 1/\varepsilon\}$$

$$:= \{z \in \mathbb{C} : z \in \sigma(K_N + E) \text{ for some } E \in \mathbb{C}^{n \times n} \text{ with } \|E\|_\infty < \varepsilon\}.$$

Even in Banach spaces, pseudospectra give lower bounds on transient growth; see, e.g., [Trefethen & E., 2005].

$$\sup_{m \geq 0} \|K^m\| \geq \sup_{z \in \sigma_\varepsilon(K)} \frac{|z| - 1}{\varepsilon}$$

If $\sigma_\varepsilon(K)$ extends more than $\varepsilon$ outside the unit disk, $\|K^m\|$ grows transiently.

Limitations: [Greenbaum & Trefethen 1994], [Ransford et al. 2007, 2009, 2011]
stability versus solution operator norm

\[ x'(t) = \alpha x(t) + \beta x(t - 1) \]

Stable choices of the \((\alpha, \beta)\) parameters

unstable \((\alpha, \beta)\) pairs

stable \((\alpha, \beta)\) pairs
stability versus solution operator norm

\[ x'(t) = \alpha x(t) + \beta x(t - 1) \]

Level sets: \( \rho(K) = 0.1, 0.2, \ldots, 1.0 \)
stability versus solution operator norm

\[ x'(t) = \alpha x(t) + \beta x(t - 1) \]

Level sets: \( \|K\| = 0.5, 1.0, \ldots, 4.5 \)
stability versus solution operator norm

\[ x'(t) = \alpha x(t) + \beta x(t - 1) \]

Superimposed level sets for \( \rho(K) \) and \( \|K\| \)
stability versus solution operator norm

\[ x'(t) = \alpha x(t) + \beta x(t - 1) \]

Superimposed level sets for \( \rho(K) \) and \( \|K\| \)
solution matrix pseudospectra ($\infty$-norm)

\[ x'(t) = \alpha x(t) + \beta x(t - 1) \]

\[ \alpha = 1 \]
\[ \beta = -1 \]
\[ \rho(K) = 1 \]
\[ \|K\|_\infty = 4.43632 \]
solution matrix pseudospectra ($\infty$-norm)

\[ x'(t) = \alpha x(t) + \beta x(t - 1) \]

\[ \alpha = 0.98995 \quad \beta = -0.99000 \quad \rho(K) = 0.99000 \quad \|K\|_\infty = 4.38204 \]
solution matrix pseudospectra ($\infty$-norm)

$$x'(t) = \alpha x(t) + \beta x(t - 1)$$

$\alpha = 0.98995$
$\beta = -0.90000$
$\rho(K) = 0.90000$
$\|K\|_{\infty} = 3.90135$
solution matrix pseudospectra ($\infty$-norm)

\[ x'(t) = \alpha x(t) + \beta x(t - 1) \]

\[ \alpha = 0.98995 \quad \beta = -0.99000 \quad \rho(K) = 0.99000 \quad ||K||_{\infty} = 4.38204 \]

\[ \alpha = 0.98995 \quad \beta = -0.90000 \quad \rho(K) = 0.90000 \quad ||K||_{\infty} = 3.90135 \]
solution operator: transient growth

\[ x'(t) = \alpha x(t) + \beta x(t - 1) \]
solution operator: transient growth

\[ x'(t) = \alpha x(t) + \beta x(t - 1) \]

As \( \alpha \uparrow 1 \) and \( \beta \downarrow -1 \), solutions exhibit arbitrary transient growth, but slowly.
can scalar equations exhibit stronger transients?

Is faster transient growth possible in a scalar equation if we allow \textit{multiple synchronized delays}?

\[
x'(t) = c_0 x(t) + c_1 x(t - 1) + c_2 x(t - 2) + \cdots + c_d x(t - d).
\]
Can scalar equations exhibit stronger transients?

Is faster transient growth possible in a scalar equation if we allow *multiple synchronized delays*?

\[ x'(t) = c_0 x(t) + c_1 x(t - 1) + c_2 x(t - 2) + \cdots + c_d x(t - d). \]

**Key:** Look for solutions of the form \( x(t) = t^d e^{\lambda t} \).
can scalar equations exhibit stronger transients?

Is faster transient growth possible in a scalar equation if we allow *multiple synchronized delays*?

\[ x'(t) = c_0 x(t) + c_1 x(t - 1) + c_2 x(t - 2) + \cdots + c_d x(t - d). \]

**Key:** *Look for solutions of the form* \( x(t) = t^d e^{\lambda t} \).

One can show that \( x(t) = t^d e^{\lambda t} \) is a solution if and only if \( c_0, c_1, \ldots, c_d \) solve the Vandermonde linear system

\[
\begin{pmatrix}
1 & 1 & 1 & \cdots & 1 \\
0 & 1 & 2 & \cdots & d \\
0 & 1 & 4 & \cdots & d^2 \\
& \vdots & \vdots & \ddots & \vdots \\
0 & 1 & 2^d & \cdots & d^d
\end{pmatrix}
\begin{pmatrix}
c_0 \\
e^{-\lambda} c_1 \\
e^{-2\lambda} c_2 \\
\vdots \\
e^{-d\lambda} c_d
\end{pmatrix}
= 
\begin{pmatrix}
\lambda \\
-1 \\
0 \\
\vdots \\
0
\end{pmatrix}.
\]
Commensurate delays can give much larger pseudospectra

\[ x'(t) = c_0 x(t) + c_1 x(t - 1) + \cdots + c_d x(t - d) \]
Commensurate delays can induce strong transients

\[ x'(t) = c_0 x(t) + c_1 x(t - 1) + c_2 x(t - 2) + \cdots + c_d x(t - d) \]

Initial data:
\[ x(t) = -1 + 2e^{10t} \]
for \( t \leq 0 \)

\[ c_0 = 0.8946 \]
\[ c_1 = -0.9000 \]
\[ d = 1 \]

\[ c_0 = 1.3946 \]
\[ c_1 = -1.8000 \]
\[ c_2 = 0.4050 \]
\[ d = 2 \]

\[ c_0 = 1.7280 \]
\[ c_1 = -2.7000 \]
\[ c_2 = 1.2150 \]
\[ c_3 = -0.2430 \]
\[ d = 3 \]

\[ c_0 = 1.9780 \]
\[ c_1 = -3.9600 \]
\[ c_2 = 2.4300 \]
\[ c_3 = -0.9720 \]
\[ c_4 = 0.1640 \]
\[ d = 4 \]
With commensurate delays, solutions to scalar equations can exhibit significant transient growth very quickly in time.

\[ x'(t) = c_0 x(t) + c_1 x(t - 1) + \cdots + c_d x(t - d) \]
rational interpolation for nlevps

Rational / Loewner techniques motivated by algorithms from model reduction

- **Structure Preserving Rational Interpolation**: iteratively improve projection subspaces via interpolation points and directions.
- **Data-Driven Rational Interpolation Matrix Pencils**: reduce nonlinear problem to linear matrix pencil with tangential interpolation property.
- **Minimal Realization via Rational Contour Integrals**: isolates a transfer function for a linear system, recover via Loewner minimal realization techniques.

transients for delay equations

Solutions to *scalar* delay equations can exhibit strong transient growth.

- **Finite dimensional nonlinear problem** ⇒ **infinite dimensional linear problem**
- Pseudospectral theory applies to the linear problem, *but the choice of norm is important*.
- Chebyshev collocation keeps the discretization matrix size small.
- Adding commensurate delays enables a *faster rate* of initial transient growth.