The use of rational approximation for linearization of models that are nonlinear in the frequency

Elke Deckers   Stijn Jonckheere   Karl Meerbergen

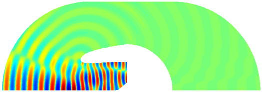
KU LEUVEN   fwo   BE·MATHS·IN

June 21st, 2021

Analysis of vibrations

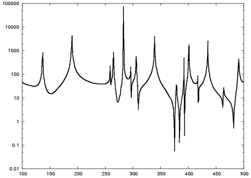
- 'Classical' analysis (frequency domain): Helmholtz equation
- Discretization: FE, BE, Trefftz

$$(K - \omega^2 M)x = f \qquad\qquad (K + \imath\omega C - \omega^2 M)x = f$$



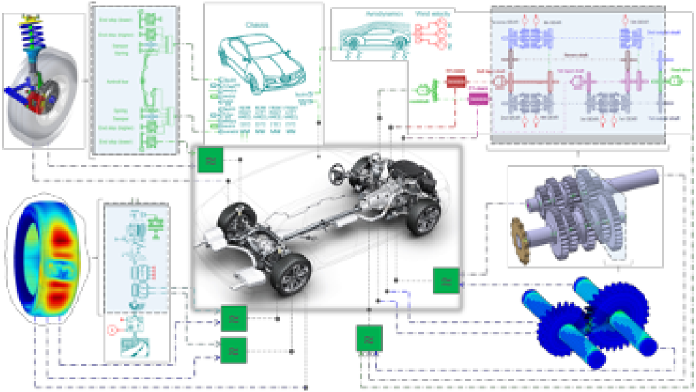
Simple ω dependency $\Longrightarrow$

- Frequency sweeping (computing x for many ω)
- Time stepping (connection between Fourier domain and time domain)
- Eigenvalue computations



Trends in the analysis of vibrations

- Nonlinear frequency dependencies
- Nonlinear time dependent models (mechatronic systems)
- Digital twins, optimization, inverse problems:
  - Time critical: model order reduction and other fast methods
  - Time domain
  - Coupled systems



Polynomial and rational

- Polynomial and rational frequency dependency = linear in the frequency.
- 'Quadratic eigenvalue problem'

$$(K + sC + s^2M)x = f$$

- is 'linearized' to

$$\begin{bmatrix} K & C + sM \\ sI & -I \end{bmatrix} \begin{pmatrix} x \\ sx \end{pmatrix} = \begin{pmatrix} f \\ 0 \end{pmatrix}$$

- Linear in $s = \imath\omega \Longrightarrow$:
  - Time stepping
  - Fast frequency sweeping
  - Eigenvalues

Rational: Plate with poro-elastic damping

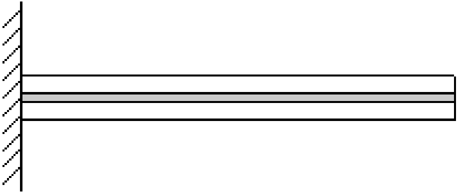
Model:

$$\left(K_e + s^2 M + \left(G_0 + \sum_{j=1}^{p} G_j \frac{s\tau_j}{1 + s\tau_j} \right) K_v \right) x = f$$

with $p = 12$. Problem of size $n = 28,087$ [Lietaert, Deckers, M., 2018]
Linearization:

$$\begin{bmatrix} K_e + G_0 K_v & sM & G_1 K_v & \cdots & G_p K_v \\ sI & -I & & & \\ -s\tau_1 I & & 1 + s\tau_1 & & \\ & \ddots & \ddots & & \\ & & & -s\tau_p & 1 + s\tau_p \end{bmatrix} \begin{pmatrix} x \\ sx \\ \frac{s\tau_1}{1+s\tau_1} x \\ \vdots \\ \frac{s\tau_p}{1+s\tau_p} x \end{pmatrix} = \begin{pmatrix} f \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

Nonlinear damping



- Clamped sandwich beam
- Linear system

$$\left(K_e + \frac{G_0 + G_\infty (s\tau)^\alpha}{1 + (s\tau)^\alpha} K_v + s^2 M\right) x = f$$

with $\alpha = 0.675$ and $\tau = 8.230$.
Parameters $G_0, G_\infty, \alpha, \tau$ are obtained from measurements.

Linearizations of nonlinear frequency dependencies

Two approaches for

$$\begin{aligned} A(s)x &= f \\ y &= c^T x \end{aligned}$$

1. Rational approximation of $y(s)$:
   - Sampling methods (Loewner matrices) [Mayo, Antoulas, 2007]
   - Possibly combined with IRKA (TFIRKA) [Beattie, Gugercin, ...]
   - Used for 'matrix free' BEM [Desmet, Jonckheere, 2016]
   - Computational cost is high.
2. Rational approximation of $A(s)$:
   - Approximate $A(s)$ by a (rational) polynomial
   - Form the linear representation
   - All advantages of linear models
   - ... provided the rational approximation is fast to form

Approach of linearization

$$A(s)x = f$$

We assume the following form (holomorphic decomposition):

$$A(s) = \sum_{i=1}^{m} C_i g_i(s)$$

with g_i *holomorphic* in $\imath\mathbb{R}$.
Two steps

1. polynomial/rational approximation

$$A(s) \approx \sum_{i=1}^{m} C_i \psi_i(s)$$

   with ψ_i (rational) polynomial of degree d, with poles outside $\imath\mathbb{R}$.
2. Linearization

Rational approximation

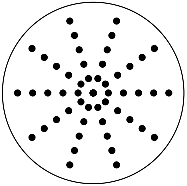
- Padé approximation [Su & Bai, 2011]
- Infinite Arnoldi: Spectral discretization [Trefethen 2000], [Michiels, Niculescu 2007] [Jarlebring, Michiel, M. 2013]
- NLEIGS: potential theory [Güttel, Van Beemen, M. & Michiels, 2014]
- AAA: Adaptive Antoulas Anderson [Nakatsukasa, Sète, Trefethen, 2018] [Lietaert, M., 2018] [Lietaert, M., Perez, Vandereycken, 2020] [Güttel, Negri Porzio and Tisseur, 2020]

AAA approximation

Rational approximation in barycentric form:

$$g(s) \approx r(s) = \sum_{j=1}^{d} \frac{g(z_j)\,\omega_j}{s - z_j} \Big/ \sum_{j=1}^{d} \frac{\omega_j}{s - z_j}$$

Ω



- z_j: support point
- ω_j: weight

Selection of z_j and ω_j: greedy procedure *adaptive Antoulas–Anderson* [Nakatsukasa, Sète, & Trefethen, 2017]

AAA

$$A(s) = A_0 + sB_0 + A_1 g_1(s)$$

$$A(s) \approx R(s) = A_0 + sB_0 + A_1 (a_1^T (E_1 - sF_1)^{-1} b_1$$

and linearization

$$\begin{bmatrix} A_0 + sB_0 & a_1^T \otimes A_1 \\ b_1 \otimes I_n & (E_1 - sF_1) \otimes I_n \end{bmatrix}$$

with

$$\left[\begin{array}{c|c} 0 & a_1^T \\ \hline b_1 & E_1 - sF_1 \end{array}\right] = \left[\begin{array}{c|ccccc} 0 & g_1(z_1) & g_1(z_2) & \cdots & & g_1(z_d) \\ \hline -1 & 1 & 1 & \cdots & & 1 \\ 0 & \omega_2(s - z_1) & \omega_1(z_2 - s) & & & \\ \vdots & & \omega_3(s - z_2) & \ddots & & \\ \vdots & & & \ddots & \omega_{d-2}(z_{d-1} - s) & \\ 0 & & & & \omega_d(s - z_{d-1}) & \omega_{d-1}(z_d - s) \end{array}\right]$$

[Lietaert, M., Pérez, Vandereycken, 2020]

Set valued AAA

$$A(s) = A_0 + sB_0 + \sum_{j=1}^{r} A_j g_j(s)$$

if $r > 1$, then we have to build separate AAA approximations for each g_j and join them together as follows:

$$A(s) = A_0 + sB_0 + \sum_{j=1}^{r} A_j (a_j^T (E_j - sF_j)^{-1} b_j$$

and linearization (for $r = 2$):

$$\begin{bmatrix} A_0 + sB_0 & a_1^T \otimes A_1 & a_2^T \otimes A_2 \\ b_1 \otimes I_n & (E_1 - sF_1) \otimes I_n & 0 \\ b_2 \otimes I_n & 0 & (E_2 - sF_2) \otimes I_n \end{bmatrix}$$

Set valued AAA

Related to [FastAAA by Hochman, 2018]

$$A(s) = A_0 + sB_0 + \sum_{j=1}^{m} A_j g_j(s)$$

Support points and weights are the same for all g_j.

$$A(s) \approx R(s) = A_0 + sB_0 - \sum_{j=1}^{r} (a_j^T \otimes A_j)(b^T (E - sF)^{-1} \otimes I_n)$$

The linearization is

$$\begin{bmatrix} A_0 + sB_0 & \sum_{j=1}^{r} a_j^T \otimes A_j \\ b \otimes I_n & (E - sF) \otimes I_n \end{bmatrix}$$

Set valued AAA

[Elsworth & Güttel, 2018]

Apply AAA to $v^*A(s)u$ for well chosen v and u.

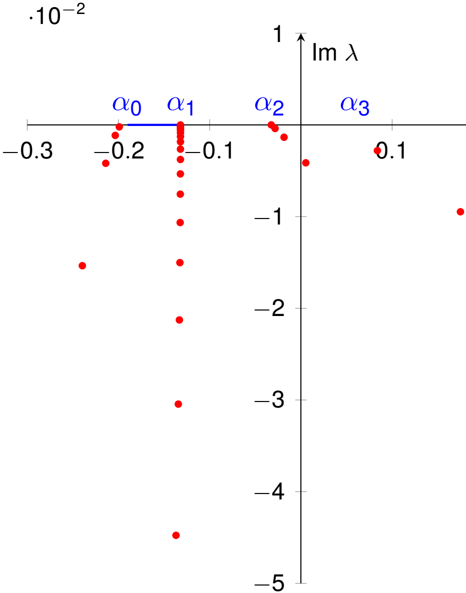
- Use the support points of $v^*A(s)u$ for rational approximation of $A(s)$.
- Good choice when matrices do not have an explicit form $A(s) = \sum_{j=1}^{m} C_j g_j(s)$.
- As matrix vector products $v^*A(s)u$ as test points required.
- We found that g_j are not always well approximated, although the linear combination

$$\sum_{j=1}^{m} (v^* C_j u) g_j(s)$$

is. (Typically lower degree for $v^*A(s)u$.)

Example

1. 2D model of a semiconductor device
2. 81 functions: $g_j = e^{i\sqrt{s-\alpha_j}}$ for $j = 0, \ldots, 80$.
3. interval $[\alpha_0, \alpha_1]$ was discretized with 1000 equidistant interior points.
4. With tolerance 10^{-12} this led to a rational approximation with $d = 45$.

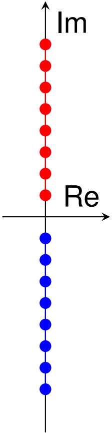


‘Real’ formulation

- Symmetry along the real axis: $g_j(\bar{s}) = \overline{g_j(s)}$.
- Obtain a real valued function for real s.
- Two complex conjugate support points $z_1, z_2 = \overline{z_2}$ (add in pairs):

$$\frac{g(z_1)\omega_1}{s - z_1} + \frac{\overline{g(z_1)}\omega_1}{s - \overline{z_1}} \Big/ \frac{g(z_1)\omega_1}{s - z_1} + \frac{\overline{g(z_1)}\omega_1}{s - \overline{z_1}}$$

- Real weights ω_1 en ω_2.
- Also see [Hochman, 2018], but without linearization.
- Make linearization real valued by linear combination of rows/columns.



Rational Krylov method

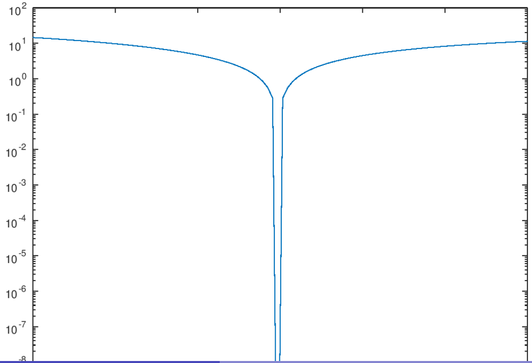
Optimal choice of interpolation points

- IRKA (Iterative Rational Krylov) [Gugercin, Antoulas, Beattie, 2008]
  - Iteratively determine interpolation points that guarantee, on convergence, minimal $\mathcal{H}_2$ error
  - Expensive procedure: each iteration, an order k model has to be constructed
- Greedy optimization [Druskin, Simoncini, 2008] [Druskin, Lieberman, Zaslavsky, 2010]
  - On each iteration, add one interpolation point
  - Choose interpolation point based on an error estimation
  - The easiest is to choose the residual norm of the linear system (cheap and accurate)
  - Does not produce an optimal reduction
- Combination: SPARK [Panzer, Jaensch, Wolf, and Lohmann, 2013].
- Computational improvement: keep the shift during a small number of iterations.

Greedy algorithm

- Build a reduced model of dimension $k \times k$ by projection of the linear model on a subspace
- At iteration k, add state vector $x(\sigma_k)$ with σ_k chosen so that the residual $r(s)$, with $s \in \imath\mathbb{R}$ has largest norm for $s = \sigma_k$.
- Higher order interpolation: build a small Krylov space for the shift:

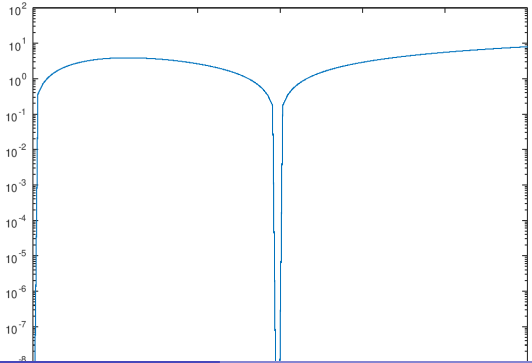
$$(\sigma_k E - A)^{-1} f, \ldots, ((\sigma_k E - A)^{-1} E)^{m-1} (\sigma_k E - A)^{-1} f$$



Greedy algorithm

- Build a reduced model of dimension $k \times k$ by projection of the linear model on a subspace
- At iteration k, add state vector $x(\sigma_k)$ with σ_k chosen so that the residual $r(s)$, with $s \in \imath\mathbb{R}$ has largest norm for $s = \sigma_k$.
- Higher order interpolation: build a small Krylov space for the shift:

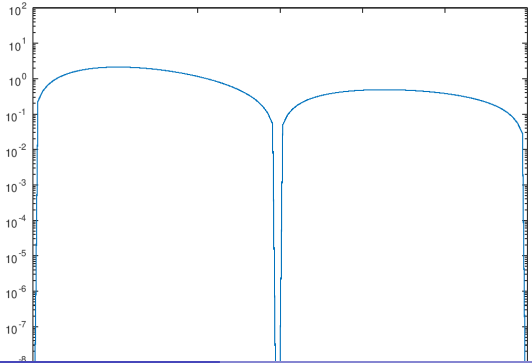
$$(\sigma_k E - A)^{-1} f, \ldots, ((\sigma_k E - A)^{-1} E)^{m-1} (\sigma_k E - A)^{-1} f$$



Greedy algorithm

- Build a reduced model of dimension $k \times k$ by projection of the linear model on a subspace
- At iteration k, add state vector $x(\sigma_k)$ with σ_k chosen so that the residual $r(s)$, with $s \in \imath\mathbb{R}$ has largest norm for $s = \sigma_k$.
- Higher order interpolation: build a small Krylov space for the shift:

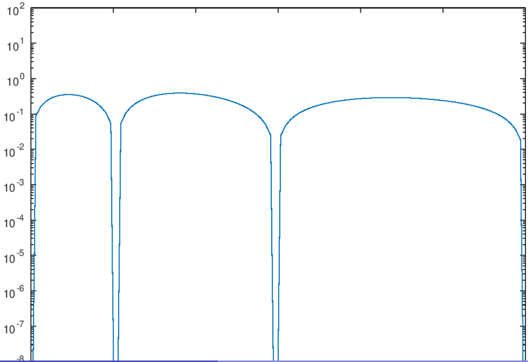
$$(\sigma_k E - A)^{-1} f, \ldots, ((\sigma_k E - A)^{-1} E)^{m-1} (\sigma_k E - A)^{-1} f$$



Greedy algorithm

- Build a reduced model of dimension $k \times k$ by projection of the linear model on a subspace
- At iteration k, add state vector $x(\sigma_k)$ with σ_k chosen so that the residual $r(s)$, with $s \in \imath\mathbb{R}$ has largest norm for $s = \sigma_k$.
- Higher order interpolation: build a small Krylov space for the shift:

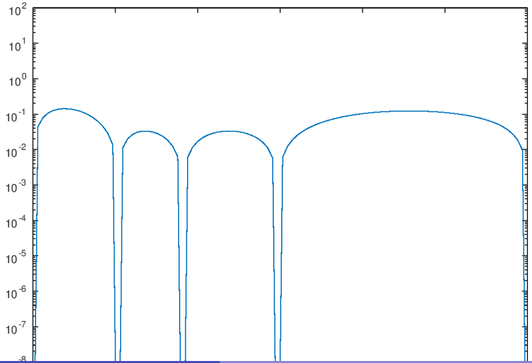
$$(\sigma_k E - A)^{-1} f, \ldots, ((\sigma_k E - A)^{-1} E)^{m-1} (\sigma_k E - A)^{-1} f$$



Greedy algorithm

- Build a reduced model of dimension $k \times k$ by projection of the linear model on a subspace
- At iteration k, add state vector $x(\sigma_k)$ with σ_k chosen so that the residual $r(s)$, with $s \in \imath\mathbb{R}$ has largest norm for $s = \sigma_k$.
- Higher order interpolation: build a small Krylov space for the shift:

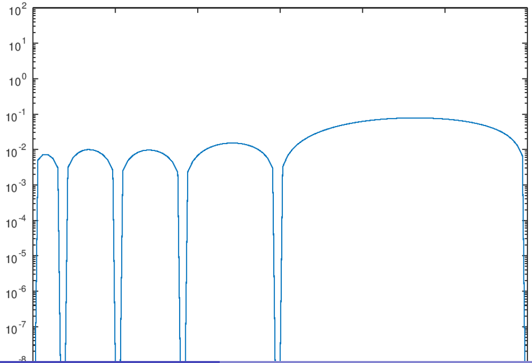
$$(\sigma_k E - A)^{-1} f, \ldots, ((\sigma_k E - A)^{-1} E)^{m-1} (\sigma_k E - A)^{-1} f$$



Greedy algorithm

- Build a reduced model of dimension $k \times k$ by projection of the linear model on a subspace
- At iteration k, add state vector $x(\sigma_k)$ with σ_k chosen so that the residual $r(s)$, with $s \in \imath\mathbb{R}$ has largest norm for $s = \sigma_k$.
- Higher order interpolation: build a small Krylov space for the shift:

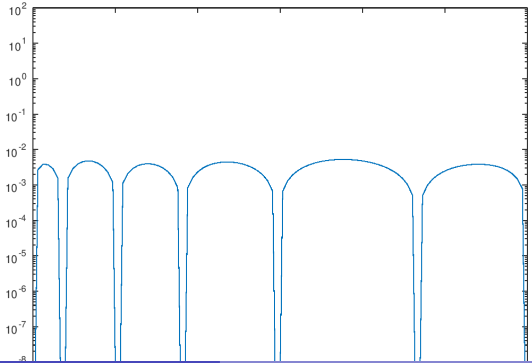
$$(\sigma_k E - A)^{-1} f, \ldots, ((\sigma_k E - A)^{-1} E)^{m-1} (\sigma_k E - A)^{-1} f$$



Greedy algorithm

- Build a reduced model of dimension $k \times k$ by projection of the linear model on a subspace
- At iteration k, add state vector $x(\sigma_k)$ with σ_k chosen so that the residual $r(s)$, with $s \in \imath\mathbb{R}$ has largest norm for $s = \sigma_k$.
- Higher order interpolation: build a small Krylov space for the shift:

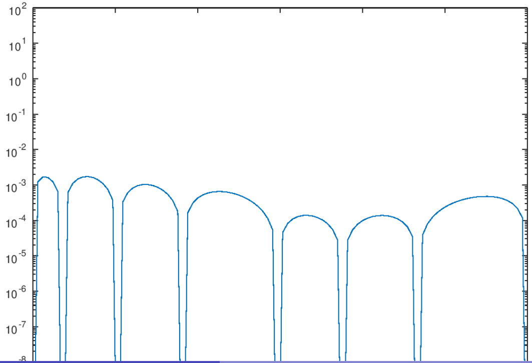
$$(\sigma_k E - A)^{-1} f, \ldots, ((\sigma_k E - A)^{-1} E)^{m-1} (\sigma_k E - A)^{-1} f$$



Greedy algorithm

- Build a reduced model of dimension $k \times k$ by projection of the linear model on a subspace
- At iteration k, add state vector $x(\sigma_k)$ with σ_k chosen so that the residual $r(s)$, with $s \in \imath\mathbb{R}$ has largest norm for $s = \sigma_k$.
- Higher order interpolation: build a small Krylov space for the shift:

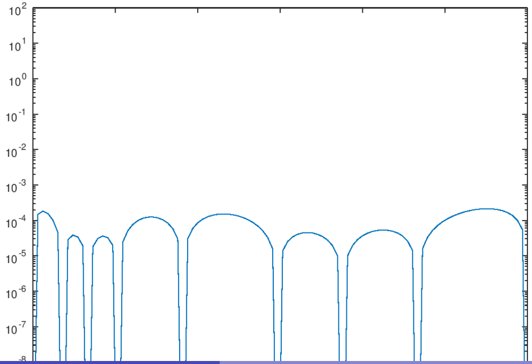
$$(\sigma_k E - A)^{-1} f, \ldots, ((\sigma_k E - A)^{-1} E)^{m-1} (\sigma_k E - A)^{-1} f$$



Greedy algorithm

- Build a reduced model of dimension $k \times k$ by projection of the linear model on a subspace
- At iteration k, add state vector $x(\sigma_k)$ with σ_k chosen so that the residual $r(s)$, with $s \in \imath\mathbb{R}$ has largest norm for $s = \sigma_k$.
- Higher order interpolation: build a small Krylov space for the shift:

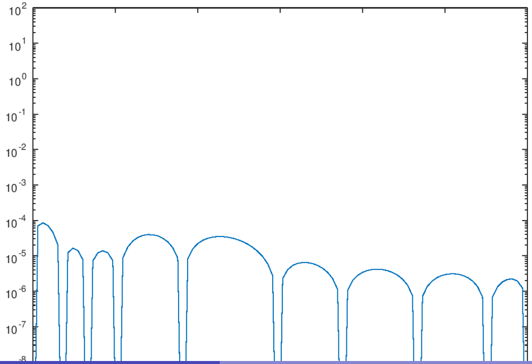
$$(\sigma_k E - A)^{-1} f, \ldots, ((\sigma_k E - A)^{-1} E)^{m-1} (\sigma_k E - A)^{-1} f$$



Greedy algorithm

- Build a reduced model of dimension $k \times k$ by projection of the linear model on a subspace
- At iteration k, add state vector $x(\sigma_k)$ with σ_k chosen so that the residual $r(s)$, with $s \in \imath\mathbb{R}$ has largest norm for $s = \sigma_k$.
- Higher order interpolation: build a small Krylov space for the shift:

$$(\sigma_k E - A)^{-1} f, \ldots, ((\sigma_k E - A)^{-1} E)^{m-1} (\sigma_k E - A)^{-1} f$$



Fast frequency sweeping: plate

- The problem ($n = 28,087$) ($0 - 500$Hz)

$$(K_e + G_v(\omega)K_v - \omega^2 M)x = f$$

$$G_v = G_0 + \sum_{k=1}^{m} G_k \frac{\imath\omega\tau_k}{1 + \imath\omega\tau_k}$$

- Results for Greedy method with multiple shifts:

solves per shift	LU factorizations	subspace dimension	time
1	30	65	251.9
20	4	141	48.1

Time integration

- Linearization in Laplace domain has state vector

$$\begin{pmatrix} x \\ \phi_1 x \\ \vdots \\ \phi_d x \end{pmatrix}$$

  with ϕ_j barycentric rational basis functions.
- In the time domain:

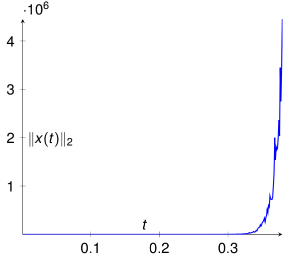
$$\begin{pmatrix} \tilde{x} \\ \tilde{y}_1 \\ \vdots \\ \tilde{y}_d \end{pmatrix} \quad \text{with} \quad \tilde{y}_j(t) = \beta_{j,0}\tilde{x}(t) + \sum_{i=1}^{d} \beta_{j,i} \int_0^\infty \tilde{x}(t-s)e^{\alpha_i s} ds.$$

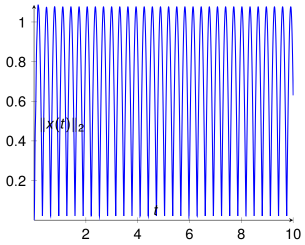
- Initial values: $\tilde{y}_j(0) = 0$ if $x(t) = 0$ for $t \leq 0$ (system in rest position).

Example of the clamped beam

$$\left(K_e + \frac{G_0 + G_\infty (s\tau)^\alpha}{1 + (s\tau)^\alpha} K_v + s^2 M\right) x = 0$$

- right-hand side $f(t) = f_0 \cdot \sin(\omega t)$ with $\omega = 2\pi \cdot 10$.
- Two selections of approximations:
  1. 1000 sample points (log scale) in $[10, 10^3]$Hz: $d = 33$
  2. 1000 sample points (log scale) in $[10, 10^5]$Hz: $d = 47$
- Amplitude corresponds to modulus of transfer function for $s = \imath\omega$.





Conclusions

- Linearizations by AAA leads to fast methods for frequency sweeping and time integration
- Choosing a large frequency range is essential for stability of the ODE/DAE
- Downside: problems with many g_j's to approximate may annihilate the disadvantages.
- Work on going with Simon Dirckx, Daan Huybrechs en Elke Deckers to improve for other situations including BEM.