Recent advances in the reduction of frequency based models for structures and vibrations

Karl Meerbergen

KU Leuven

December 11–13, 2013

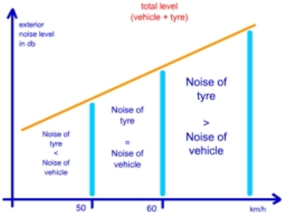
Joint work with Yao Yue, Maryam Saadvandi, Roel Van Beeumen, Wim Michiels, Wim Desmet

Outline

Outline

Examples structures and vibrations

Car tyres

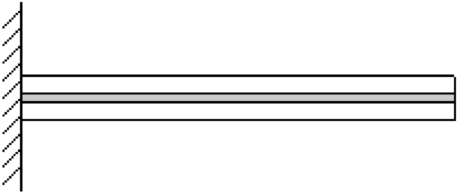


Windscreens

- Structural damping
- Choice of connection (glue) to the car

Planes

Nonlinear damping



- Clamped sandwhich beam
- 168 degrees of freedom (finite elements)
- Linear system

$$(K_e + \frac{G_0 + G_\infty (i\omega\tau)^\alpha}{1 + (i\omega\tau)^\alpha} K_v - \omega^2 M)x = f$$

with $\alpha = 0.675$ and $\tau = 8.230$.
Parameters obtained from measurements.

Lamot footbridge damper optimization

- Lamot bridge finite element model ($n = 25,962$)
- The goal is to determine the optimal stiffness and damping coefficient of four bridge dampers (=8 parameters).

Objective

- Large dynamical system in the frequency domain:

$$\begin{aligned} A_n(\omega)x &= f \cdot u(\omega) \\ y &= c^T x \end{aligned}$$

  with $x \in \mathbb{C}^n$ with n large.
  We call $H = y/u$ the *transfer function* or *frequency response function*.
- Reduce to

$$\begin{aligned} A_r(\omega)\widetilde{x} &= \widetilde{f} \cdot u(\omega) \\ \widetilde{y} &= \widetilde{c}^T \widetilde{x} \end{aligned}$$

  with $\widetilde{x} \in \mathbb{C}^r$ with $r \ll n$
- Subspace methods:
  - Right subspace $\mathrm{Range}(V)$ with $V \in \mathbb{C}^{n\times r}$ (control side)
  - Left subspace $\mathrm{Range}(W)$ with $W \in \mathbb{C}^{n\times r}$ (observation side)

$$\begin{aligned} (W^*AV)\widetilde{x} &= (W^*f) \\ \widehat{y} &= (V^*c)^T \widetilde{x} \end{aligned}$$

Objective

- Krylov methods
- Dominant pole algorithm

both for

- Nonlinear frequency dependence

$$\begin{aligned} (A_0 + i\omega A_1 + e^{-i\omega\tau} A_2)x &= f \\ y &= c^T x \end{aligned}$$

- Parametric models

$$\begin{aligned} (A_0 + i\omega A_1 + \gamma_1 A_2 + i\omega\gamma_2 A_3)x &= f \\ y &= c^T x \end{aligned}$$

Outline

Polynomial models

- Polynomial model

$$\begin{aligned}(A_0 + sA_1 + \cdots + A_N s^N)x &= f \\ y &= c^T x\end{aligned}$$

- Companion linearization

$$\begin{aligned}(\mathcal{A} + s\mathcal{B})\mathbf{x} &= \mathbf{b} \\ y &= (e_1 \otimes c)^T \mathbf{x}\end{aligned}$$

with

$$\mathcal{A}+s\mathcal{B} = \begin{bmatrix} A_0 + sA_1 & sA_2 & \cdots & sA_N \\ -sI & I & & 0 \\ & \ddots & \ddots & \vdots \\ & & -sI & I \end{bmatrix}, \mathbf{x} = \begin{pmatrix} x \\ sx \\ \vdots \\ s^{N-1}x \end{pmatrix}, \mathbf{b} = \begin{pmatrix} f \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

Nonlinear models

- General form:

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

- Approximate by a polynomial or rational polynomial by interpolation in $N+1$ points:

$$\sigma_0, \sigma_1, \ldots, \sigma_N, \ldots$$

  by a (rational) polynomial

Nonlinear models

- Different bases:
  - Newton polynomials:

$$N_{j+1}(\lambda) = N_j(\lambda) \cdot (\lambda - \sigma_j) \quad j = 1, 2, \ldots$$

  - Rational Newton polynomials

$$(\lambda - \xi_j) \cdot N_{j+1}(\lambda) = (\lambda - \sigma_j) \cdot N_j(\lambda)$$

  - barycentric Lagrange polynomials, ...
- Variants
  - Dynamic: choose polynomial coefficients and degree during the iterations
  - Static: fix the polynomial and then reduce

Newton polynomial basis

- Newton polynomials:

$$N_0 := 1, N_1 := (\lambda - \sigma_0), N_2 := (\lambda - \sigma_0)(\lambda - \sigma_1), \ldots$$

$$N_{j+1}(\lambda) = N_j(\lambda) \cdot (\lambda - \sigma_j) \quad j = 1, 2, \ldots$$

- $A(s) = A_0 N_0 + A_1 N_1 + \cdots$ with
- $\sigma_0, \sigma_1, \ldots$ are interpolation points.
- Linearization:

$$\left(\begin{bmatrix} A_0 & A_1 & \cdots & A_N \\ \sigma_0 I & I & & \\ & \ddots & \ddots & \\ & & \sigma_{N-2} I & I \end{bmatrix} - s \begin{bmatrix} 0 & \cdots & \cdots & 0 \\ I & & & \vdots \\ & \ddots & & \vdots \\ & & I & 0 \end{bmatrix} \right) \begin{bmatrix} x \\ N_1(s)x \\ N_2(s)x \\ \vdots \end{bmatrix} = \begin{bmatrix} f \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Linear reduced model

- Build reduced model for the linearization. Vectors:

$$\mathbf{V} = \begin{pmatrix} V \\ \star \\ \vdots \\ \star \end{pmatrix} \qquad \mathbf{W} = \begin{pmatrix} W \\ \star \\ \vdots \\ \star \end{pmatrix}$$

- Reduced model for the linearization:

$$\begin{aligned} (\widetilde{A} + s\widetilde{B})z &= \widetilde{f} \\ \widetilde{H} &= \widetilde{c}^T z \end{aligned}$$

  where $\widetilde{A} = \mathbf{W}^* A_N \mathbf{V}$, $\widetilde{B} = \mathbf{W}^* B_N \mathbf{V}$.
- For example, a delay differential equation can then be solved by an ODE solver
- [Freund 2005], [Michiels, Jarlebring, M. 2012]

Properties of linearized model

- Original equation:

$$A(\lambda)x = f$$

- Linearized equation ($N = 2$)

$$\left(\begin{bmatrix} A_0 & A_1 & A_2 \\ \sigma_0 I & I & \\ & \sigma_1 I & I \end{bmatrix} - s \begin{bmatrix} 0 & 0 & 0 \\ I & 0 & 0 \\ 0 & I & 0 \end{bmatrix} \right) \begin{bmatrix} x \\ N_1(s)x \\ N_2(s)x \end{bmatrix} = \begin{bmatrix} f \\ 0 \\ 0 \end{bmatrix}$$

- Eigenvectors

$$\begin{aligned} A(\lambda)p &= 0 \\ \begin{bmatrix} A_0 & A_1 & A_2 \\ (\sigma_0 - \lambda)I & I & \\ & (\sigma_1 - \lambda)I & I \end{bmatrix} \begin{pmatrix} p \\ N_1(\lambda)p \\ N_2(\lambda)p \end{pmatrix} &= 0 \end{aligned}$$

- Left eigenvectors

$$\begin{aligned} q^* A(\lambda) &= 0 \\ \begin{pmatrix} q \\ -A_1^* q + (\sigma_1 - \lambda) A_2^* q \\ -A_2^* q \end{pmatrix}^* \begin{bmatrix} A_0 & A_1 & A_2 \\ (\sigma_0 - \lambda)I & I & \\ & (\sigma_1 - \lambda)I & I \end{bmatrix} &= 0 \end{aligned}$$

Properties of linearized model

- Moments for simple interpolation points:

$$\sigma_0, \sigma_1, \sigma_2$$

- Nonlinear problems has moments: $m_j = A(\sigma_j)^{-1} f$ for $j = 0, 1, 2$.
- Linearization of the polynomial has moments:

$$\mathbf{m}_0 = \begin{pmatrix} m_0 \\ 0 \\ 0 \end{pmatrix} \quad \mathbf{m}_1 = \begin{pmatrix} m_1 \\ N_1(\sigma_1)m_1 \\ 0 \end{pmatrix} \quad \mathbf{m}_2 = \begin{pmatrix} m_2 \\ N_1(\sigma_2)m_2 \\ N_2(\sigma_2)m_2 \end{pmatrix}$$

- For 2-sided moment matching, see [Michiels, Peeters 2013]

Nonlinear reduced model

- Original matrix:

$$A(s) = \sum_{j=0}^{m} f_j(s) A_j$$

- Krylov vectors V and W of linearization
- Reduced nonlinear model:

$$\begin{aligned} \left(\sum_{j=0}^{m} f_j(s)\widetilde{A}_j\right) z &= \widetilde{f} \\ \widetilde{H} &= \widetilde{c}^T z \end{aligned}$$

  where $\widetilde{A}_j = W^* A_j V$.
- SOAR [Bai, Su 2005]

Algorithms

- Apply a Krylov method to the linearization (moment matching)
  - Krylov methods: two-sided Arnoldi
  - Rational Krylov methods: the poles correspond to the interpolation points σ_j
- Use the dominant pole algorithm:
  - The σ_j are the Ritz values computed by the dominant pole algorithm
  - Linearization supports deflation (which is needed to compute more than one pole)

Rational Krylov method

- Linear system:

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

- Two-sided model reduction
  - Interpolation in $s_0, \ldots, s_N$:

$$\begin{aligned} V &= \mathrm{orth}((A+s_0B)^{-1}f, \ldots, (A+s_NB)^{-1}f) \\ W &= \mathrm{orth}((A+s_0B)^{-*}c, \ldots, (A+s_NB)^{-*}c) \end{aligned}$$

  - Moment matching property:

$$\widetilde{H}(s_j) = H(s_j)$$

Rational Krylov method

- One sided algorithm:

  Let $b = (A + \sigma_0 B)^{-1} f$
  Let $v_1 = b/\|b\|$
  for $j = 1, \ldots, k$ **do**
  Solve
  $$(A + \sigma_j B)t_j = Bv_j$$
  Orthonormalize t_j against V_j and add: V_{j+1}
  end for

- Numerically more stable than accumulating moments.

Rational Krylov

Shift-and-invert step: $(A - \sigma_j B)w = Bv$

1. First step: multiply by B: shift vector downwards

$$
\begin{array}{ccc} B \cdot v & = & w \end{array}
$$

$$
\begin{bmatrix} 0 & \cdots & \cdots & 0 \\ I & & & \vdots \\ & \ddots & & \vdots \\ & & I & 0 \end{bmatrix} \cdot \begin{bmatrix} \star \\ \star \\ \vdots \\ \star \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ \star \\ \star \\ \vdots \\ \star \end{bmatrix}
$$

2. Then solve with A

$$
\begin{bmatrix} A_0 & A_1 & A_2 & A_3 & A_4 & A_5 \\ \sigma_0 I & I & & & & \\ & \sigma_1 I & I & & & \\ & & \sigma_2 I & I & & \\ & & & \sigma_3 I & I & \\ & & & & \sigma_4 I & I \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ \star \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \star \\ \star \\ \star \\ \star \\ \star \\ \star \end{bmatrix}
$$

This is rational Krylov with poles $\sigma_1, \sigma_2, \ldots$

Rational Krylov

Shift-and-invert step: $(A - \sigma_j B)w = Bv$

1. First step: multiply by B: shift vector downwards

$$B \cdot v \quad = \quad w$$

$$\begin{bmatrix} 0 & \cdots & \cdots & 0 \\ I & & & \vdots \\ & \ddots & & \vdots \\ & & I & 0 \end{bmatrix} \cdot \begin{bmatrix} \star \\ 0 \\ \vdots \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ \star \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

2. Then solve with $A - \sigma_1 B$

$$\begin{bmatrix} A_0 & A_1 & A_2 & A_3 & A_4 & A_5 \\ \star & I & & & & \\ & 0 & I & & & \\ & & \star & I & & \\ & & & \star & I & \\ & & & & \star & I \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ \star \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \star \\ \star \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

This is rational Krylov with poles $\sigma_1, \sigma_2, \ldots$

Rational Krylov

Shift-and-invert step: $(A - \sigma_j B)w = Bv$

1. First step: multiply by B: shift vector downwards

$$B \cdot v \quad = \quad w$$

$$\begin{bmatrix} 0 & \cdots & \cdots & 0 \\ I & & & \vdots \\ & \ddots & & \vdots \\ & & I & 0 \end{bmatrix} \cdot \begin{bmatrix} \star \\ \star \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ \star \\ \star \\ 0 \\ 0 \end{bmatrix}$$

2. Then solve with $A - \sigma_2 B$

$$\begin{bmatrix} A_0 & A_1 & A_2 & A_3 & A_4 & A_5 \\ \star & I & & & & \\ & \star & I & & & \\ & & 0 & I & & \\ & & & \star & I & \\ & & & & \star & I \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ \star \\ \star \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \star \\ \star \\ \star \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

This is rational Krylov with poles $\sigma_1, \sigma_2, \ldots$

Rational Krylov

Shift-and-invert step: $(A - \sigma_j B)w = Bv$

1. First step: multiply by B: shift vector downwards

$$B \cdot v = w$$

$$\begin{bmatrix} 0 & \cdots & \cdots & 0 \\ I & & & \vdots \\ & \ddots & & \vdots \\ & & I & 0 \end{bmatrix} \cdot \begin{bmatrix} \star \\ \star \\ \star \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ \star \\ \star \\ \star \\ 0 \end{bmatrix}$$

2. Then solve with $A - \sigma_3 B$

$$\begin{bmatrix} A_0 & A_1 & A_2 & A_3 & A_4 & A_5 \\ \star & I & & & & \\ & \star & I & & & \\ & & \star & I & & \\ & & & 0 & I & \\ & & & & \star & I \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ \star \\ \star \\ \star \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \star \\ \star \\ \star \\ \star \\ 0 \\ 0 \end{bmatrix}$$

This is rational Krylov with poles $\sigma_1, \sigma_2, \ldots$

Rational Krylov

Shift-and-invert step: $(A - \sigma_j B)w = Bv$

1. First step: multiply by B: shift vector downwards

$$
\begin{array}{ccc} B \cdot v & = & w \end{array}
$$

$$
\begin{bmatrix} 0 & \cdots & \cdots & 0 \\ I & & & \vdots \\ & \ddots & & \vdots \\ & & I & 0 \end{bmatrix} \cdot \begin{bmatrix} \star \\ \star \\ \star \\ \star \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ \star \\ \star \\ \star \\ \star \end{bmatrix}
$$

2. Then solve with $A - \sigma_4 B$

$$
\begin{bmatrix} A_0 & A_1 & A_2 & A_3 & A_4 & A_5 \\ \star & I & & & & \\ & \star & I & & & \\ & & \star & I & & \\ & & & \star & I & \\ & & & & 0 & I \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ \star \\ \star \\ \star \\ \star \\ 0 \end{bmatrix} = \begin{bmatrix} \star \\ \star \\ \star \\ \star \\ \star \\ 0 \end{bmatrix}
$$

This is rational Krylov with poles $\sigma_1, \sigma_2, \ldots$

Numerical example

'gun' problem Manchester-Berlin NEVP collection

$$\begin{cases} A(\omega)x(\omega) & = & fu(\omega) \\ y(\omega) & = & cx(\omega) \end{cases},$$

where

$$A(\omega) = K - \omega^2 M + i\sqrt{\omega^2 - \sigma_1^2}W_1 + i\sqrt{\omega^2 - \sigma_2^2}W_2,$$

with

- M, K, W_1, W_2 real symmetric matrices of dimension 9956.
- K positive semidefinite and M positive definite.
- W_1 and W_2 low rank matrices.
- The vectors f and c are

$$f = [1\ 1\ \cdots\ 1]^T/\sqrt{n}, \qquad c = [1\ 1\ \cdots\ 1]^T/\sqrt{n}.$$

Example: 'gun' problem

Reduced order models were constructed

$$\begin{cases} \widehat{A}(\omega)x(\omega) & = & \widehat{f}u(\omega) \\ y(\omega) & = & \widehat{c}^*\widehat{x}(\omega) \end{cases},$$

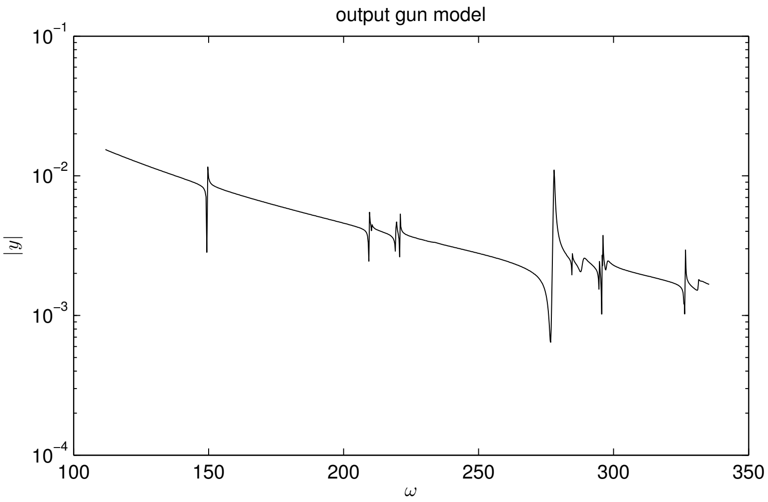
where $\widehat{x}(\omega) \in \mathbb{C}^k$ and

$$\widehat{A}(\omega) = V_k^* K V_k - \omega^2 \; V_k^* M V_k + i\sqrt{\omega^2 - \sigma_1^2} \; V_k^* W_1 V_k$$
$$+ \; i\sqrt{\omega^2 - \sigma_2^2} \; V_k^* W_2 V_k,$$

$$\widehat{f} = V_k^* f,$$

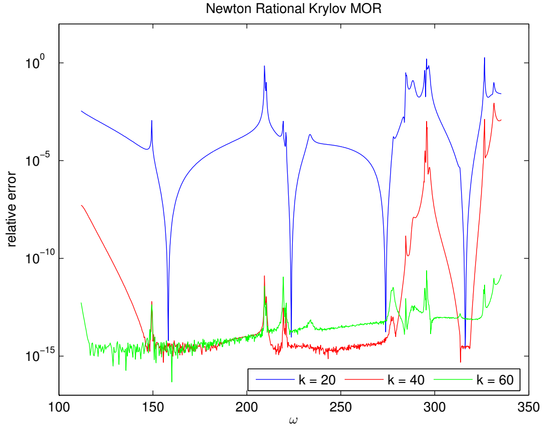
$$\widehat{c} = V_k^* c$$

Example: ‘gun’ problem



Example: 'gun' problem

Hermite interpolation in 4 points:



reduced model	full model
3.8 s ($k = 20$) 4.9 s ($k = 40$) 7.1 s ($k = 60$)	512 s

Example: boundary element problem

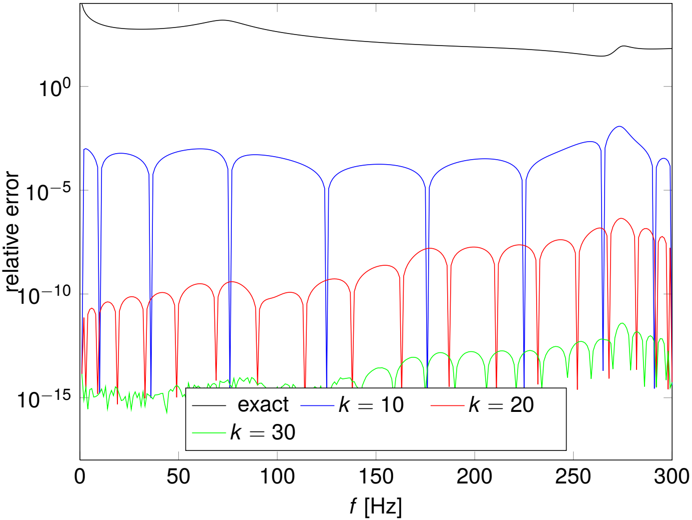
Boundary element formulation for the Helmholtz equation (LMS — Siemens):

- Nonlinear in ω
- Full matrices

Procedure as follows:

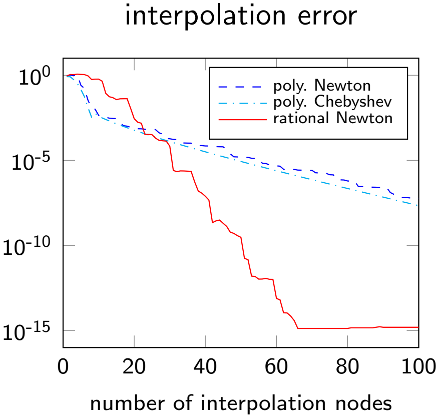
1. Interpolate in k points on the ω axis
2. Build the interpolating polynomial (barycentric Lagrange polynomials)
3. Build reduced polynomial problem of degree $k-1$ with matrices of dimension $k \times k$.

Example: boundary element problem



Other approximations

- [Michiels, Jarlebring and M. 2012] Use spectral discretization for delay equation
- [Güttel, Van Beeumen, M., Michiels 2013] Use rational Newton polynomials



Dominant pole algorithm

- Computes poles of

$$c^*A(s)^{-1}f \approx \sum_j \frac{R_j}{s-\lambda_j}$$

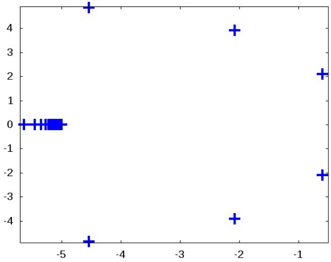
with

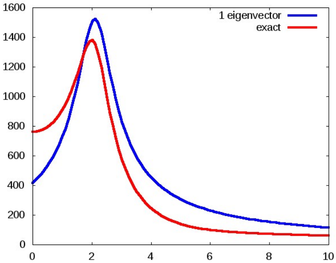
$$R_j = \frac{(q_j^* f)(c^* p_j)}{q_j^* A'(\lambda_j) p_j}$$

and $A(\lambda_j)p_j = 0$ and $q_j^* A(\lambda_j) = 0$.

- DPA = Newton's method applied to $1/(c^*A(s)^{-1}f)$.
- It converges to the eigenvalue with largest

$$\frac{|R_j|}{|\mathrm{Re}(\lambda_j)|}$$





Dominant pole algorithm

- For a linear problem, DPA with subspace acceleration is like rational Krylov/Jacobi-Davidson for the eigenvalue problem:
  1. solve a sequence of linear systems:

  $$\begin{aligned} (A + s_j B)x_j &= f \\ (A + s_j B)^* t_j &= c \end{aligned}$$

  2. Accumulate in subspace: $V_j = [V_{j-1}, x_j]$, $W_j = [W_{j-1}, t_j]$.
  3. Solve projected eigenvalue problem

  $$\det((W_j^* A V_j) + \lambda (W_j^* B V_j)) = 0 \quad \rightarrow \quad \text{new } s_{j+1}$$

  (= two-sided version of rational Krylov)
- For computing more than one eigenvalue, modify the definition of dominance:

  $$f - (Bp)(q^* f)$$

  [Rommes, & Martins 2006, 2008]

Dominant pole algorithm

- For a nonlinear problem, we do the same, but for a polynomial eigenvalue problem interpolating in the s_j's.
  1. solve a sequence of linear systems:

$$\begin{aligned} A(s_j)x_j &= f \\ A(s_j)^* y_j &= c \end{aligned}$$

  2. Solve projected nonlinear eigenvalue problem

$$\det(W_j^* A(\lambda) V_j) = 0 \quad \rightarrow \quad \text{new } s_{j+1}$$

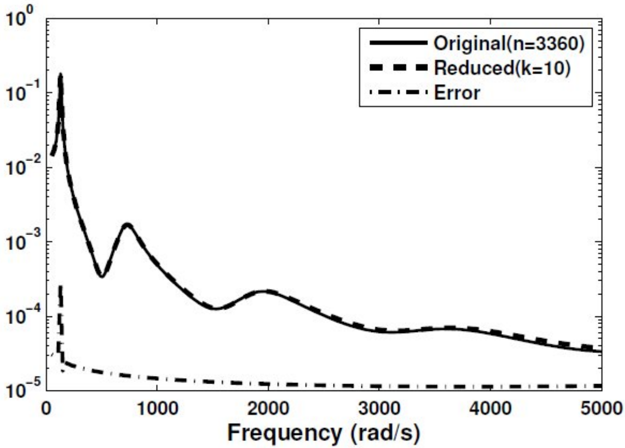
- Deflation can be reformulated as:

$$f - \frac{A(s)p}{(\lambda - s)(q^* A'(\lambda)p)} q^* f$$

(Follows from the use of a 'linearization')
[Saadvandi, M. & Jarlebring 2012]

Example: Sandwhich beam

- Computed five dominant poles
- Small scale (nonlinear) eigenvalue problem is solved with rational Krylov



Outline

Parametric models

- Model:

$$\left((A_0 + \sum_{j=1}^{p} \gamma_j A_j) + s(B_0 + \sum_{j=1}^{p} \gamma_j B_j) \right) x = f$$
$$y = c^T x$$

  with $x \in \mathbb{C}^n$ with n large; $\gamma \in \mathbb{R}^p$ are parameters
- Examples:
  - The mean of the output:

$$z = \int_\Gamma y d\gamma$$

    Numerical integration using cubature rule:

$$z \approx \sum_{j=1}^{N} w_j \cdot y(\gamma_j)$$

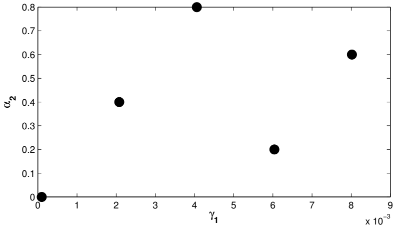
  - Optimization problem:

$$\min_{\gamma \in \Gamma} |y|^2$$

Discretize the parameter space

Discretization of parameter space:

- Cartesian grid (only for small number of parameters)
- Sparse grids
- Lattice rules



$$1, 1, 2, 3, 5, 8, \ldots, Z, N\ .$$

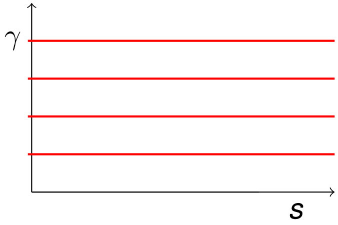
We define the N lattice points as

$$\alpha^{(j)} = \begin{bmatrix} \alpha_1^{(j)} \\ \alpha_2^{(j)} \end{bmatrix} = \frac{\mathrm{mod}\left((j-1)\begin{bmatrix} 1 \\ Z \end{bmatrix}, N\right)}{N}, \qquad j = 1, \ldots, N$$

Parametric models

Interpolatory model reduction:

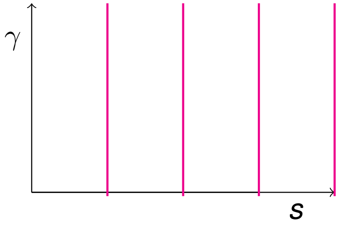
1. Discretize the parameter space
2. Build a reduced model for each node γ_j
3. Merge the spaces



For DPA

Parameter integrated in the algorithm:

- Vectors are functions of γ
- Eigenvalues and moments are functions of γ
- Compute these functions



For moment matching

Hermite interpolation

- For moment matching:
  - $\dfrac{\partial^i \widehat{H}(\sigma_j)}{\partial s^i} = \dfrac{\partial^i H(\sigma_j)}{\partial s^i}$ for $i = 0, \ldots, 2k-2$
  - $\dfrac{\partial^{i+1} \widehat{H}(\sigma_j)}{\partial \gamma \partial s^i} = \dfrac{\partial^{i+1} H(\sigma_j)}{\partial \gamma \partial s^i}$ for $i = 0, \ldots, k-1$

  [Baur, Beattie, Benner, Gugercin 2011] [Yue, M. 2012]
- For dominant poles:
  - $\lambda_j = \widehat{\lambda}_j$
  - $\dfrac{\partial \lambda_j}{\partial \gamma} = \dfrac{\partial \widehat{\lambda}_j}{\partial \gamma}$
  - $\widehat{H}(s) \sim H(s), s \to \lambda$
  - $\dfrac{\partial \widehat{H}(s)}{\partial \gamma} \sim \dfrac{\partial H(s)}{\partial \gamma}, s \to \lambda$

  [Saadvandi, M. & Desmet 2013]

Interpolatory MOR

- Standard algorithm:

  for $j = 1, \ldots, N$ **do**

  Compute $V_k^{(j)}$ and $W_k^{(j)}$ for node $\gamma^{(j)}$

  Merge the subspaces in V, W.

  end for

- Continuation DPA (CDPA)

  Compute V_k and W_k for node $\gamma^{(1)}$

  for $j = 1, \ldots, N$ **do**

  Use V ane W as a starting subspace (good starting values).

  Compute V_k and W_k for node $\gamma^{(j)}$.

  Merge the subspaces in V, W.

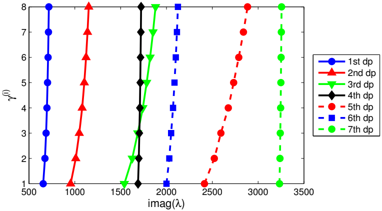
  end for

Numerical example: aluminum plate

Model:

$$(K_0 + \gamma_2 K_{\mathrm{bnd}} + i\omega(C_0 + \gamma_1 C_{\mathrm{bnd}}) - \omega^2 M)x = f$$

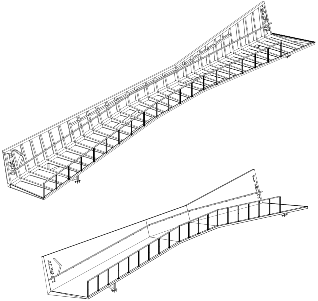
$$\begin{aligned} \gamma_1 &= 10^{-2} + \alpha_1(10^{-2} - 10^{-4}) \\ \gamma_2 &= \frac{-100}{\alpha_2 - 1} - 101 \\ \alpha_{1,2} &\in [0, 1) \end{aligned}$$

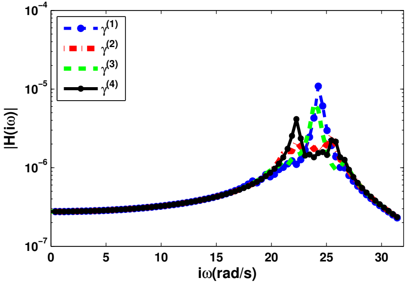


	Iterations
$\gamma^{(1)}$	22
$\gamma^{(2)}$	18
$\gamma^{(3)}$	13
$\gamma^{(4)}$	15
$\gamma^{(5)}$	10

Numerical example: Lamot footbridge

$$\left\{ \begin{array}{rcl} \left((1+0.02i)K_0 + (k_1 + i\omega c_1)K_1 - \omega^2 M\right) x & = & \\ y & = & \end{array} \right.$$





	# iterations	time (min)
$\gamma^{(1)}$	8	0.45
$\gamma^{(2)}$	4	0.25
$\gamma^{(3)}$	7	0.43
$\gamma^{(4)}$	3	0.19
Total	22	1.32

Tensorize the solution

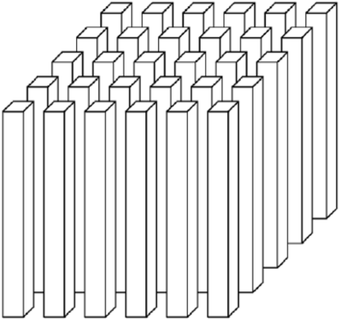
- Discretize the parameters in a grid of quadrature nodes:

$$\gamma^{(i_1,\ldots,i_p)} = (\gamma_1^{(i_1)}, \ldots, \gamma_p^{(i_p)}) \in \Gamma \subset \mathbb{R}^p$$

  with $i_j = 1, \ldots, n_j$, $j = 1, \ldots, p$.

- Then represent the state vector as a tensor:

$$\mathcal{X}_{:,i_1,\ldots,i_p} = x(\gamma^{(i_1,\ldots,i_p)})$$



- System for each grid point:

$$\left((A_0 + sB_0 + \sum_{j=1}^{p} \gamma_j^{(i_1,\ldots,i_p)} (A_j + sB_j)) \right) \mathcal{X}_{:,i_1,\ldots,i_p} = f$$

Tensor-Krylov

- For each grid point, a linear system needs to be solved:

$$\left(A_0 + sB_0 + \sum_{j=1}^{p} \gamma_j^{(i_1,\ldots,i_p)}(A_j + sB_j)\right) \mathcal{X}_{:,i_1,\ldots,i_p} \quad = \quad f$$

  This can be done with a tensor Krylov method [Kressner, Tobler 2011]: $\mathcal{X}$ is Tucker tensor

- Solve preconditioned system iteratively

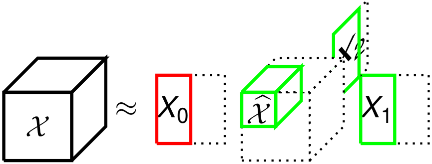
$$M^{-1}\left(A_0 + sB_0 + \sum_{j=1}^{p} \gamma_j^{(i_1,\ldots,i_p)}(A_j + sB_j)\right) x(\gamma^{(i_1,\ldots,i_p)}) = M^{-1} f$$

  The main operation is the matrix-vector product
  $t = M^{-1}(A(\gamma) + sB(\gamma))z$

Implementing preconditioned matrix-vector product

- Represent tensors using Tucker decomposition

$$X = X_1 \widehat{X} X_2^T \qquad \text{(SVD)}$$
$$\mathcal{X} = (X_0, X_1, \ldots, X_p) \bullet \widehat{\mathcal{X}}$$



- The operation $t = M^{-1}(A(\gamma) + sB(\gamma))z$ goes as follows
  1. No need to evaluate $M^{-1}(A(\gamma) + sB(\gamma))z$ for every grid point.
  2. z is represented as a low rank tensor $\mathcal{Z} \approx (Z_0, \ldots, Z_p) \bullet \widehat{\mathcal{Z}}$ with $Z_0 \in \mathbb{C}^{n \times r}$ and $r \ll n$.
  3. Sparse matvec operations: $M^{-1}(A_j + sB_j)Z_0, j = 0, \ldots, p$
  4. Operations on tensors: can be expensive!
- Key point: low rank X_0

Two algorithms

1. Interpolatory MOR
   - Tensor-Krylov with preconditioner M^{-1} with $M \approx A_0 + \sigma B_0$
   - Perform sufficiently large number of iterations so that parametric system is solved accurately enough
2. Small number of iterations:
   - Connection with multivariate Padé

Multivariate Padé

We want to build the parametric Krylov space:

$$(A(\gamma)+\sigma_0 B(\gamma))^{-1}f, (A(\gamma)+\sigma_1 B(\gamma))^{-1}f, \ldots, (A(\gamma)+\sigma_k B(\gamma))^{-1}f$$

where the basis vectors are discretized by tensors.

Theorem

Consider the power sequence ($M = A_0 + \sigma_j B_0$):

$$\begin{aligned} v_j^{(0)}(\gamma) &\equiv M^{-1}f \\ v_j^{(\ell)}(\gamma) &= v_j^{(\ell-1)}(\gamma) + \sum_{j=1}^{p} M^{-1}\gamma_j A_j v_j^{(\ell-1)}(\gamma) \end{aligned}$$

$v_j^{(\ell)}$ *is a moment matching polynomial of degree ℓ in γ:*

$$\|v_j^{(\ell)}(\gamma) - (A(\gamma)+\sigma_j B(\gamma))^{-1}f\|_2 = O(\|\gamma^{\ell+1}\|)$$

Reduced model

- Form a reduced model for the mean problem (using the tensors as state vectors.)

$$\left((A_0 + \sum_{j=1}^{p} \gamma_j^{(i_1,\ldots,i_p)} A_j) + sB_0\right) \mathcal{X}_{:,i_1,\ldots,i_p} = f$$

$$z = \sum_{i_1,\ldots,i_p} w_{i_1,\ldots,i_p}(c^* \mathcal{X}_{:,i_1,\ldots,i_p})$$

- From tensor representation (evaluation for grid points is available)

$$\mathcal{Y} = (c^*)\bullet_0 \mathcal{X}$$

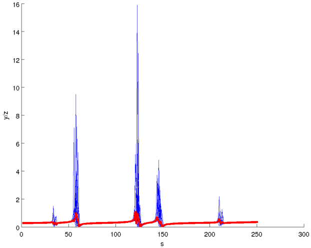
- Classical parametric model using the mode-0 vector space of iteration vectors: columns of $V_{j,0}$ in

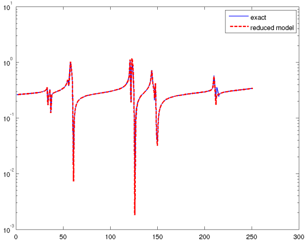
$$\mathcal{V}_j = (V_{j,0}, \ldots, V_{j,p})\bullet_0 \widehat{\mathcal{V}}_j$$

  span the moments.
  Project on the joined $V = \mathrm{orth}([V_{0,0}, \ldots, V_{k,0}])$

Numerical example: acoustic box

3D box with parametric Robin boundary condition on two faces





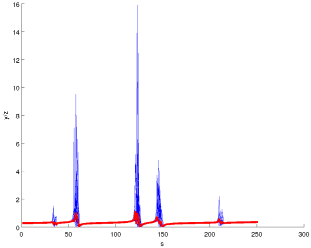
$\ell = 3$ GMRES iterations

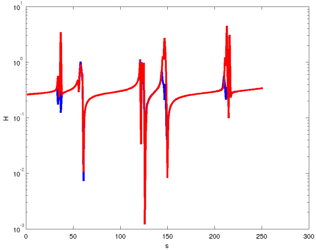
Tensors truncated $\tau = 10^{-8}$

Iter.	Rank	Iter.	Rank	Iter.	Rank
1	16	8	109	14	205
2	27	9	125	15	221
3	38	10	141	16	237
4	51	11	157	17	253
5	64	12	173	18	269
6	79	13	189	19	285
7	94				

Numerical example: acoustic box

3D box with parametric Robin boundary condition on two faces





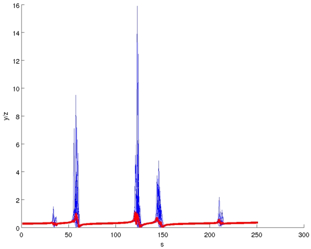
$\ell = 3$ GMRES iterations

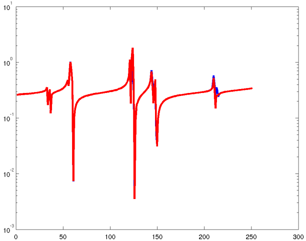
Tensors truncated $\tau = 10^{-4}$

Iter.	Rank	Iter.	Rank	Iter.	Rank
1	7	8	51	14	111
2	11	9	60	15	122
3	16	10	69	16	137
4	22	11	79	17	148
5	28	12	88	18	161
6	35	13	99	19	177
7	43				

Numerical example: acoustic box

3D box with parametric Robin boundary condition on two faces





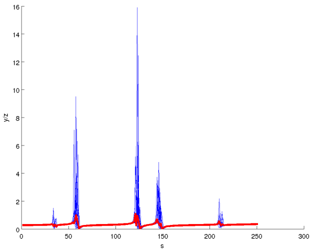
$\ell = 15$ GMRES iterations

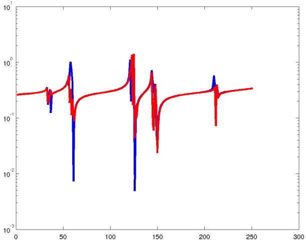
Tensors truncated $\tau = 10^{-4}$

Iter.	Rank	Iter.	Rank	Iter.	Rank
1	17	8	108	14	204
2	25	9	124	15	220
3	34	10	140	16	236
4	50	11	156	17	252
5	63	12	172	18	268
6	77	13	188	19	284
7	93				

Numerical example: acoustic box

3D box with parametric Robin boundary condition on two faces





M computed by ILU
$\ell = 15$ GMRES iterations
Tensors truncated $\tau = 10^{-4}$

Iter.	Rank	Iter.	Rank	Iter.	Rank
1	17	8	111	14	207
2	28	9	127	15	223
3	39	10	143	16	239
4	53	11	159	17	255
5	66	12	175	18	271
6	80	13	191	19	287
7	96				

Numerical example: footbridge

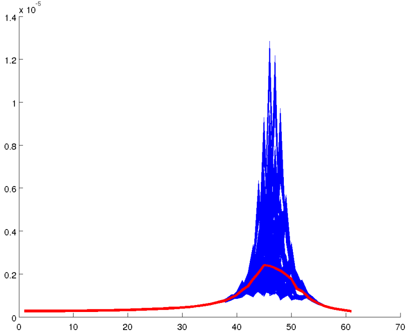
- The matrix dependency on the 8 parameters has rank 8

$$\left((1+0.02i)K_0+\sum_{i=1}^{4}(k_i+i\omega c_i)K_i-\omega^2 M\right)x=f$$

(Matrices $K_1,\ldots,K_4$ each have rank two.)

Ranks of iteration matrices in Arnoldi's method:

$n_i = 3$	$n_i = 5$
9	9
14	14
20	25
28	37
35	51



Results related to block Arnoldi for low rank parametric terms [Yue, M. 2013]

Outline

Conclusions

- Nonlinear in the frequency:
  - Rational Krylov with simple poles is easy
  - Rational Krylov with high order moment matching: possible, but technical (based on polynomial interpolation)
- Parametric:
  - Interpolatory MOR for DPA can enjoy continuation properties of eigenspaces
- Tensors:
  - Related to multivariate interpolation
  - If a low rank tensor approximation does not exist, it is not a practical method
  - Automatic rank reduction