Linearizations for NEPv: nonlinear eigenvalue problems with eigenvector nonlinearity

Karl Meerbergen
Joint work with
Wim Michiels, Roel Van Beeumen, Elias Jarlebring, Rob Claes, Simon Telen, and Parikshit Upadhyaya

KU Leuven

Back to the numerical linear algebra roots of polynomials and nonlinear eigenvalue problems

NEPv: nonlinear eigenvalue problem with vector nonlinearity

Definition (NEPv)

Given a matrix valued function

$$T : \mathbb{C}^n \times \mathbb{C} \to \mathbb{C}^{n\times n}, (x, \lambda) \to T(x, \lambda)$$

of functions in x.
NEPv:

$$T(x, \lambda)x = 0$$

NEPv

$$T(x,\lambda)x = 0 \quad x \neq 0$$

Example:

$$\begin{bmatrix} \frac{x_1}{x_2} + \lambda & 0 \\ 0 & \frac{x_2}{x_1} + 2\lambda \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = 0$$

Solutions ($x_1 \neq 0 \neq x_2$):

$$\left(-1/\sqrt{2}, \begin{pmatrix} 1 \\ \sqrt{2} \end{pmatrix}\right) \quad , \quad \left(1/\sqrt{2}, \begin{pmatrix} 1 \\ -\sqrt{2} \end{pmatrix}\right)$$

- ‘Characteristic equation’ by elimination of x_1, x_2 (function of λ):

$$-1 + 2\lambda^2 = 0.$$

Eigenvector nonlinearities

Most general form:

$$T(V)V = V\Lambda$$

with $V \in \mathbb{C}^{n\times k}$, $\Lambda \in \mathbb{C}^{k\times k}$ and $T : \mathbb{P}^{n-1} \to \mathbb{C}^{n\times n}$

Applications:

- Physics: e.g., Kohn-Sham equations [Hartree, 1928], [Kohn&Sham, 1965].
- Data science: support vector machine classifier. [Bai et al, 2018]
- Data science: spectral clustering of data. [Bühler & Hein, 2009], [Tudisco et al, 2019]

Methods for computing a few eigenvalues (based on Self Consistent Field iteration):

- Fixed point iteration: SCF [Conces&Le Bris, 2000], [Saad et al. 2010], [Levitt, 2012], [Liu et al., 2014, 2015], [Upadhyaya et al., 2021], [Bai et al., 2022], [Saunders&Hillier, 1973], [Martin, 2020]
- Level shifting [Saunders&Hillier, 1973]
- DIIS – LIST [Pulay, 1982] [Garza&Scuseria, 2015]
- Secant SCF [Claes&M.,2022]

Polynomial matrices and linearization

Polynomial matrix:

$$P(\lambda) = P_0 + \lambda P_1 + \lambda^2 P_2 + \cdots + \lambda^d P_d \in \mathbb{C}^{n\times n}$$

Eigenvalue λ: $\det A(\lambda) = 0$ for a regular A.
Companion (Strong) linearization

$$\mathbf{L} = \mathbf{A} - \lambda \mathbf{E} = \begin{bmatrix} P_0 & \cdots & P_{d-2} & P_{d-1} + \lambda P_d \\ -\lambda I & I & & \\ & \ddots & \ddots & \\ & & -sI & I \end{bmatrix} \in \mathbb{C}^{nd\times nd}$$

P and $\mathbf{L}$ have the same eigenvalues (including multiplicities).

Polynomial and rational matrices

- Polynomial matrices

$$P_0 + \lambda P_1 + \lambda^2 P_2 + \cdots + \lambda^d P_d$$

- Rational matrices:

$$P_0 + \lambda P_1 + \frac{1}{\lambda} R_1 + \frac{1}{\lambda - 2} R_2$$

Two classes of reliable methods for large scale and selection of eigenvalues:

- Linearization + Krylov method = CORK
- Contour integration = rational filter

Objective: reliable NEP solver

Linearization:

- Even if T is linear in v, the problem is not known to be written as a linear eigenvalue problem in general.
- Carleman linearization: too large size matrices for practical use: $n + n^2 + \cdots + n^d$ instead of nd for PEP.

Objective: reliable NEP solver

Linearization:

- Even if T is linear in v, the problem is not known to be written as a linear eigenvalue problem in general.
- Carleman linearization: too large size matrices for practical use: $n + n^2 + \cdots + n^d$ instead of nd for PEP.

Ideas:

1. A rational NEPv [Claes, Jarlebring, M. & Upadhaya, 2022]
2. Interpolating Secant-SCF and Muller-SCF [Claes & M., 2023]
3. Contour integration for PEPv, [Claes, M. & Telen, 2023]

All methods compute a selection of eigenvalues.

1. SCF

- Fixed point iteration method for $T(x)x = \lambda x$: solve x_k from

$$T(x_{k-1})x_k = \lambda_k x_k \quad , \quad k = 1, 2, \ldots$$

- Sometimes slow or no convergence
- Level shifted SVF [Bai, Li, Lu, 2022]

$$(T(x) + \sigma x x^T)x = (\lambda - \sigma)x$$

- DIIS [Pulay, 1980], [Garza & Scuseria, 2012]:

$$T(c_1 x_1 + \cdots + c_k x_k)$$

- Using information from previous iterations is a key idea in many subspace methods.

Linearization

Polynomial or rational formulation:

$$A(s) = \sum_{j=0}^{d-1} (A_j - sB_j)\phi_j(s) \in \mathbb{C}^{n \times n}$$

with ϕ_j scalar functions and A_j, B_j constant matrices.

Linearization

Polynomial or rational formulation:

$$A(s) = \sum_{j=0}^{d-1} (A_j - sB_j)\phi_j(s) \in \mathbb{C}^{n\times n}$$

with ϕ_j scalar functions and A_j, B_j constant matrices. Linearization:

$$\begin{aligned} \mathbf{L}(s) &= \mathbf{A} - s\mathbf{B} \\ &= \left[\begin{array}{cccc} A_0 - sB_0 & A_1 - sB_1 & \cdots & A_{d-1} - sB_{d-1} \\ \hline \multicolumn{4}{c}{(M - sN)\otimes I_n} \end{array}\right] \\ &= \left[\begin{array}{c} [A_i - sB_i]_{i=0}^{d-1} \\ (M - sN)\otimes I_n \end{array}\right] \end{aligned}$$

with

$$(M - sN)\Phi = 0 \quad \text{and} \quad \Phi = \begin{pmatrix} \phi_0 \\ \vdots \\ \phi_d \end{pmatrix}$$

Linearization

$$\mathbf{L}(s) = \mathbf{A} - s\mathbf{B}$$

$$= \left[\begin{array}{cccc} A_0 - sB_0 & A_1 - sB_1 & \cdots & A_{d-1} - sB_{d-1} \\ \hline \multicolumn{4}{c}{(M - sN) \otimes I_n} \end{array}\right]$$

$$= \begin{bmatrix} [A_i - sB_i]_{i=0}^{d-1} \\ (M - sN) \otimes I_n \end{bmatrix}$$

Eigenvectors:

$$A(\lambda)x = 0 \quad \Leftrightarrow \quad \mathbf{L}(\lambda) \begin{pmatrix} \phi_0(\lambda)x \\ \vdots \\ \phi_{d-1}(\lambda)x \end{pmatrix} = 0$$

Companion linearization

- Matrix polynomial:

$$A(\lambda) = A_0 + \lambda A_1 + \lambda^2 A_2 + \lambda^3 A_3$$

- Linearization:

$$\mathbf{L}_1 \quad = \quad \mathbf{A} - \lambda \mathbf{B} = \begin{bmatrix} A_0 & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix} - \lambda \begin{bmatrix} -A_1 & -A_2 & -A_3 \\ I & 0 & 0 \\ 0 & I & 0 \end{bmatrix}$$

- Eigenvectors:

$$A(\lambda)x = 0 \quad \Leftrightarrow \quad \mathbf{L}(\lambda) \begin{pmatrix} x \\ \lambda x \\ \lambda^2 x \end{pmatrix} = 0$$

Krylov spaces for Companion linearization

- Linearization:

$$\mathbf{L}_1 = \mathbf{A} - \lambda \mathbf{B} = \begin{bmatrix} A_0 & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix} - \lambda \begin{bmatrix} -A_1 & -A_2 & -A_3 \\ I & 0 & 0 \\ 0 & I & 0 \end{bmatrix}$$

Krylov spaces for Companion linearization

- Linearization:

$$\mathbf{L}_1 \quad = \quad \mathbf{A} - \lambda \mathbf{B} = \begin{bmatrix} A_0 & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix} - \lambda \begin{bmatrix} -A_1 & -A_2 & -A_3 \\ I & 0 & 0 \\ 0 & I & 0 \end{bmatrix}$$

- Krylov space with

$$\mathbf{S} = \mathbf{A}^{-1}\mathbf{B} = \begin{bmatrix} -A_0^{-1}A_1 & -A_0^{-1}A_2 & -A_0^{-1}A_3 \\ I & 0 & 0 \\ 0 & I & 0 \end{bmatrix}$$

$$\begin{pmatrix} v \\ 0 \\ 0 \end{pmatrix} \xrightarrow{\mathbf{S}} \begin{pmatrix} w = -A_0^{-1}A_1 v \\ v \\ 0 \end{pmatrix} \xrightarrow{\mathbf{S}} \begin{pmatrix} t = -A_0^{-1}A_1 w - A^{-1}A_1 v \\ w \\ v \end{pmatrix}$$

Compact rational Krylov decomposition

[Su, Zhang, Bai 2008], [Zhang, Su 2013] & [Kressner, Roman 2013], [Van Beeumen, M. & Michiels, 2015], [Dopico et al., 2019], ...

The iteration vectors:

$$\mathbf{V} = \begin{bmatrix} v_{11} & v_{12} & \cdots & v_{1,j+1} \\ v_{21} & v_{22} & \cdots & v_{2,j+1} \\ \vdots & \vdots & \ddots & \vdots \\ v_{d1} & v_{d2} & \cdots & v_{d,j+1} \end{bmatrix}$$

- Define Q such that

$$\mathrm{span}(Q) = \mathrm{span}\left\{ \begin{bmatrix} v_{11} & \cdots & v_{1,j+1} & \cdots & v_{d1} & \cdots & v_{d,j+1} \end{bmatrix} \right\}$$

Compact rational Krylov decomposition

[Su, Zhang, Bai 2008], [Zhang, Su 2013] & [Kressner, Roman 2013], [Van Beeumen, M. & Michiels, 2015], [Dopico et al., 2019], ...

The iteration vectors:

$$\mathbf{V} = \begin{bmatrix} v_{11} & v_{12} & \cdots & v_{1,j+1} \\ v_{21} & v_{22} & \cdots & v_{2,j+1} \\ \vdots & \vdots & \ddots & \vdots \\ v_{d1} & v_{d2} & \cdots & v_{d,j+1} \end{bmatrix}$$

- Define Q such that

$$\begin{aligned} \operatorname{span}(Q) &= \operatorname{span}\left\{\begin{bmatrix} v_{11} & \cdots & v_{1,j+1} & \cdots & v_{d1} & \cdots & v_{d,j+1} \end{bmatrix}\right\} \\ &= \operatorname{span}\left\{\begin{bmatrix} v_{11} & \cdots & v_{d1} & v_{12} & \cdots & v_{1,j+1} \end{bmatrix}\right\} \end{aligned}$$

Compact rational Krylov decomposition

[Su, Zhang, Bai 2008], [Zhang, Su 2013] & [Kressner, Roman 2013], [Van Beeumen, M. & Michiels, 2015], [Dopico et al., 2019], …

The iteration vectors:

$$\mathbf{V} = \begin{bmatrix} v_{11} & v_{12} & \cdots & v_{1,j+1} \\ v_{21} & v_{22} & \cdots & v_{2,j+1} \\ \vdots & \vdots & \ddots & \vdots \\ v_{d1} & v_{d2} & \cdots & v_{d,j+1} \end{bmatrix}$$

- Define Q such that

$$\begin{aligned} \text{span}(Q) &= \text{span}\left\{\begin{bmatrix} v_{11} & \cdots & v_{1,j+1} & \cdots & v_{d1} & \cdots & v_{d,j+1} \end{bmatrix}\right\} \\ &= \text{span}\left\{\begin{bmatrix} v_{11} & \cdots & v_{d1} & v_{12} & \cdots & v_{1,j+1} \end{bmatrix}\right\} \end{aligned}$$

  with

$$r := \text{rank}(Q) \leq d + j.$$

- Not exploiting structure: $dj + d$ vectors of length n

Compact rational Krylov decomposition

[Su, Zhang, Bai 2008], [Zhang, Su 2013] & [Kressner, Roman 2013], [Van Beeumen, M. & Michiels, 2015], [Dopico et al., 2019], ...

The iteration vectors:

$$\mathbf{V} = \begin{bmatrix} v_{11} & v_{12} & \cdots & v_{1,j+1} \\ v_{21} & v_{22} & \cdots & v_{2,j+1} \\ \vdots & \vdots & \ddots & \vdots \\ v_{d1} & v_{d2} & \cdots & v_{d,j+1} \end{bmatrix} = \begin{bmatrix} Q & & & \\ & Q & & \\ & & \ddots & \\ & & & Q \end{bmatrix} \cdot \begin{bmatrix} u_{1,1} & u_{1,2} & \cdots & u_{1,j+1} \\ u_{2,1} & u_{2,2} & \cdots & u_{2,j+1} \\ \vdots & & & \vdots \\ u_{d,1} & \cdots & \cdots & u_{d,d} \end{bmatrix}$$

- Define Q such that

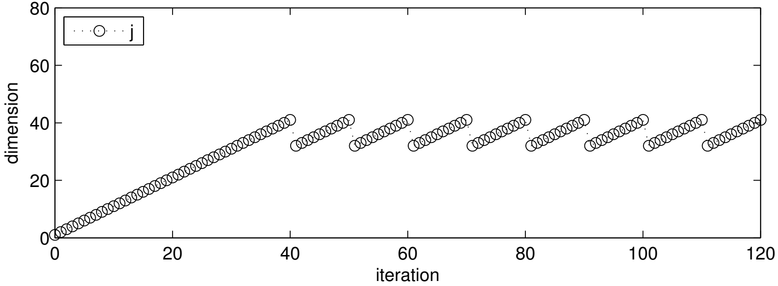
$$\begin{aligned} \text{span}(Q) &= \text{span}\left\{\begin{bmatrix} v_{11} & \cdots & v_{1,j+1} & \cdots & v_{d1} & \cdots & v_{d,j+1} \end{bmatrix}\right\} \\ &= \text{span}\left\{\begin{bmatrix} v_{11} & \cdots & v_{d1} & v_{12} & \cdots & v_{1,j+1} \end{bmatrix}\right\} \end{aligned}$$

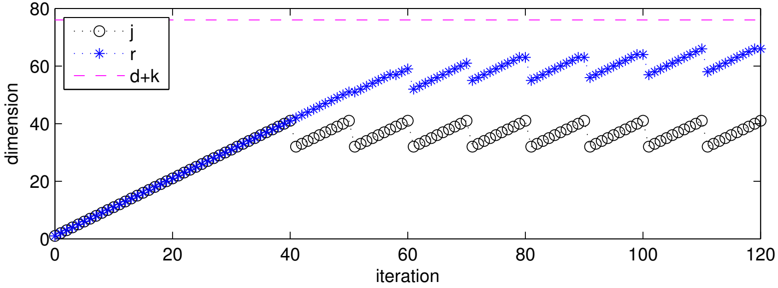
  with

$$r := \text{rank}(Q) \leq d + j.$$

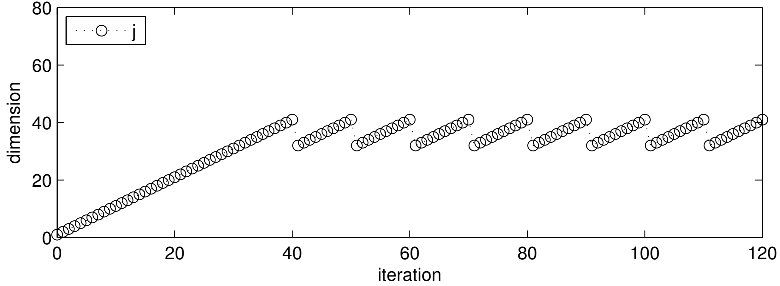
- Not exploiting structure: $dj + d$ vectors of length n

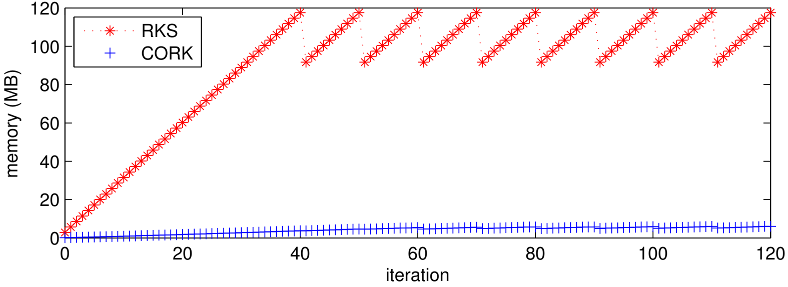
Numerical experiments



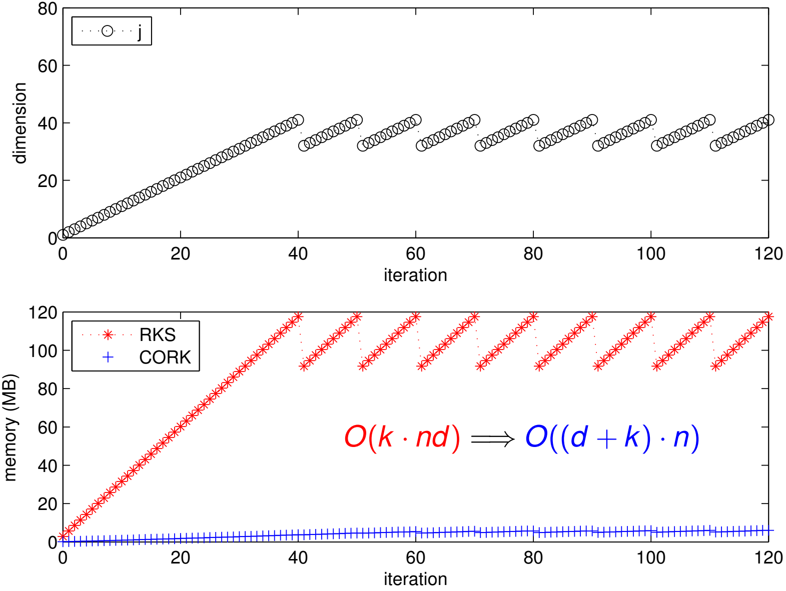


Numerical experiments





Numerical experiments



Approximate Linearizations

- For the NEP $\det(A\lambda)) = 0$ with A holomorphic in λ in some region in $\mathbb{C}$, e.g.,

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

- Approximate $A(\lambda)$ by a rational matrix and then linearize.
- For NEPv, one could assume that the eigenvector is a function of λ, and then approximate

$$T(x(\lambda), \lambda) \approx R(\lambda)$$

- The problem is that $x(\lambda)$ usually is not a function.
- What is possible: local approximation [Claes & M., 2022]:
  - Secant SCF
  - Muller SCF

2. REPv

Rational functions:

$$T(x,\lambda) = A + \lambda B + \frac{r_1^T x}{s_1^T x} T_1 + \cdots + \frac{r_m^T x}{s_m^T x} T_m,$$

Introduction of variables μ_i so that $(s_i^T x)\mu_i = r_i^T x$ turns this a problem in $n+m$ variables x and $\mu = [\mu_1 \ldots, \mu_m]$:

$$\begin{aligned} Ax + \lambda Bx + C_1\mu_1 x + \cdots + C_m\mu_m x &= 0 \\ s_1^T x\mu_1 - r_1^T x &= 0 \\ &\vdots \\ s_m^T x\mu_m - r_m^T x &= 0 \end{aligned}$$

The alternative is to view $\mu_1, \ldots, \mu_m$ as 'eigenvalue parameters' and then solve a linear $(m+1)$ parameter eigenvalue problem

[Claes, Jarlebring, M. and Upadhyaya, 2022]

REPv: linearization

- REPv is transformed to a linear $m+1$-multiparameter eigenvalue problem (MEP).
- Case $m = 1$:

$$Ax + \lambda Bx + C\mu_1 x = 0$$
$$s_1^T x \mu_1 - r_1^T x = 0$$

- Rewrite as:

$$\begin{aligned} Ax + \lambda Bx + \mu_1 Cx &= 0 \\ (A + g_1 r_1^T)x + \lambda Bx + \mu_1 (C - g_1 s_1^T)x &= 0 \end{aligned}$$

- Elimination of μ_1:

$$\Delta_0 = \begin{vmatrix} B & C_1 \\ B & C_1 - g_1 s_1^T \end{vmatrix}_\otimes \quad \Delta_1 = \begin{vmatrix} A & C_1 \\ A + g_1 r_1^T & C_1 - g_1 s_1^T \end{vmatrix}_\otimes \in \mathbb{C}^{n^2 \times n^2}$$

- Operator determinant:

$$\Delta_1 + \lambda \Delta_0 = (A + \lambda B) \otimes (C_1 - g_1 s_1^T) - C_1 \otimes (A + g_1 r_1^T + \lambda B)$$

REPv: linearization

- Eigenpairs: if (λ, x) is an eigenpair of T, i.e., $(x, \lambda) = 0$, then

$$\Delta_1(x \otimes x) = \lambda \Delta_0(x \otimes x)$$

- $x \otimes x$ is called a symmetric eigenvector of $\Delta_1 - \lambda\Delta_0$.
- Any symmetric eigenvector of $\Delta_1 - \lambda\Delta_0$ corresponds to an eigenvector T, provided that $s_1^T x \neq 0$ (otherwise $\mu = \infty$).
- Let $x_1 \otimes x_2$ be a nonsymmetric eigenvector, then x_1 is an eigenvector of T iff $g_1^T y_2 \neq 0$ with $y_1 \otimes y_2$ the left eigenvector of $\Delta_1 - \lambda\Delta_0$.
- In the last case, also $x_1 \otimes x_1$ is an eigenvector of $\Delta_1 - \lambda\Delta_0$.
- Therefore, we only consider symmetric eigenvectors.
- Note that there always can be spurious eigenpairs.

REPv: linearization example

Find (λ, x) in $\mathbb{C} \times \mathbb{C}^2$ such that

$$\left(A + \lambda B + \frac{r^T x}{s^T x} C\right) x = 0,$$

$$A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \quad C = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}, \quad r = \begin{bmatrix} 3 \\ 2 \end{bmatrix}, \quad s = \begin{bmatrix} 4 \\ 3 \end{bmatrix}$$

With $g = \begin{bmatrix} 1 & 3 \end{bmatrix}^T$,

$$\Delta_0 = \begin{vmatrix} B & C \\ B & (C - gs^T) \end{vmatrix}_{\otimes} = \begin{bmatrix} -4 & -7 & -4 & -6 \\ -18 & -16 & -24 & -16 \\ -6 & -9 & -9 & -14 \\ -36 & -24 & -51 & -36 \end{bmatrix},$$

$$\Delta_1 = \begin{vmatrix} -A & C \\ -(A + gr^T) & C - gs^T \end{vmatrix}_{\otimes} = \begin{bmatrix} 10 & 9 & 2 & 3 \\ 30 & 22 & 12 & 8 \\ 0 & 0 & 6 & 6 \\ 0 & 0 & 21 & 15 \end{bmatrix}.$$

REPv: linearization example

The four eigenpairs obtained by solving the GEP:

$$
\begin{aligned}
\lambda_1 &\approx 5.2462, \quad z_1 \approx \begin{bmatrix} 1 \\ -0.69 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ -0.69 \end{bmatrix}, \\
\lambda_2 &\approx -0.4224, \quad z_2 \approx \begin{bmatrix} 1 \\ -1.47 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ -1.47 \end{bmatrix}, \\
\lambda_3 &\approx -0.4367, \quad z_3 \approx \begin{bmatrix} 1 \\ -14.8 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ -14.8 \end{bmatrix}, \\
\lambda_4 &\approx -1.2500, \quad z_4 \approx \begin{bmatrix} 1 \\ -1.88 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ -0.83 \end{bmatrix}.
\end{aligned}
$$

(Note that y_1 and y_2 for λ_4 are orthogonal to g_1.)

REPv: linearization for $m > 1$

$$\Delta_0 = \begin{vmatrix} B & C_1 & \cdots & C_m \\ B & C_1 - g_1 s_1^T & \cdots & C_m \\ \vdots & \vdots & \ddots & \vdots \\ B & C_1 & \cdots & C_m - g_m s_m^T, \end{vmatrix}_\otimes$$

and

$$\Delta_i = \begin{vmatrix} B & C_1 & \cdots & -A & \cdots & C_m \\ B & C_1 - g_1 s_1^T & \cdots & -A - g_1 r_1^T & \cdots & C_m \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ B & C_1 & \cdots & \underbrace{-A - g_m r_m^T}_{\text{column } i} & \cdots & C_m - g_m s_m^T \end{vmatrix}_\otimes$$

Solution of MEP

- Jacobi Davidson method: fixed point iteration method on $\lambda, \mu_1, \ldots, \mu_m, x_1, \ldots, x_m$ [Kosir, Plestenjak, Hochstenbach]
  - Projection method: suitable for large scale.
  - Every iteration: solution of linear systems $A_{1,i}x_1 + \lambda B_i x_i + \mu_1 C_{1,i} x_i + \cdots + C_{m,i} x_i = b_i$ for $i = 1, \ldots, m$
  - Disadvantage: good estimate of $\mu_1, \ldots, \mu_m$ are needed.
- Residual inverse iteration:
  - Variation on Jacobi Davidson. Find eigenvalues nearest $\lambda = \sigma$, $\mu_1 = \tau_1, \ldots, \mu_m = \tau_m$
  - Respects the low rank structure of eigenvector.
- Inverse iteration on Delta matrices:
  - Exploitation of Kronecker structure is possible
  - Exploitation of low rank structure of eigenvector is possible.
  - Only estimate of λ is needed.
- Krylov iteration on Delta matrices:
  - Exploitation of Kronecker structure is possible
  - Exploitation of low rank structure of eigenvector is hard.
  - Only estimate of λ is needed.

REPv: residual inverse iteration

- Quasi Newton method[1]

$$\begin{aligned} (A + \lambda B + \mu_1 C_1)x_1 &= 0, \\ \left(A + g_1 r_1^T + \lambda B + \mu_1 (C_1 - g_1 s_1^T)\right) x_2 &= 0, \\ v_1^T x_1 - 1 &= 0, \\ v_2^T x_2 - 1 &= 0. \end{aligned}$$

using Jacobian for fixed $\lambda = \sigma$ and $\mu_1 = \tau_1$ (shifts).

[1] [Plestenjak, 2016]

REPv: residual inverse iteration

- Given $x_1^{(0)}$ and $x_2^{(0)}$.
- Iteration k:

$$x_1^{(k+1)} = x_1^{(k)} - (A + \sigma B + \tau_1 C_1)^{-1}(Ax_1^{(k)} + \lambda^{(k)} Bx_1^{(k)} + \mu_1^{(k)} Cx_1^{(k)})$$

  simiular equation for $x_2^{(k+1)}$ and $\lambda^{(k+1)}$ and $\mu_1^{(k+1)}$ solved of a 2×2 linear system.
- Linear convergence under mild conditions.
- If $v_1 = v_2 = v$, $x_1^{(0)} = x_2^{(0)}$ then $x_1^{(k)} = x_2^{(k)}$ for $k > 0$.

REPv: inverse iteration

- Eigenvalue problem:

$$(\Delta_1+\lambda\Delta_0)(z\otimes z) = (A+\lambda B)z\otimes(C_1-g_1s_1^T)z-C_1z\otimes(A+g_1r_1^T+\lambda B)z =$$

- Write in matrix form:

$$(C_1 - g_1 s_1^T)Z(A+\lambda B)^T - (A+g_1 r_1^T+\lambda B)ZC_1^T = 0$$

with $Z = zz^T$.

- If the initial vector is symmetric, then all iteration vectors are symmetric (Not proven yet.)
- Under the same conditions, the convergence rate is $|\sigma-\lambda_1|/|\sigma-\lambda_2|$.
- For $m = 1$: solve from Sylvester equation[2]:

$$(A+g_1r_1^T+\sigma B)Z_{k+1}C^T - (C-g_1s_1^T)Z_{k+1}(A+\sigma B)^T = (C-g_1s_1^T)Z_kB^T - BZ_kC^T$$

- For $m > 1$, tensor train subspace method can be used [3]

[2] [M. & Plestenjak, 2015]

[3] [Ruymbeek et al, 2022]

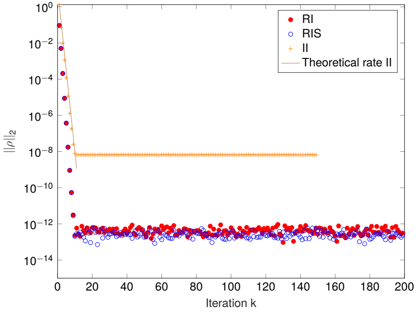
REPv: inverse iteration

- Iterate Z_k is not rank 1: complicates the solution of the Sylvester equation (or rank efficient methods in the tensor case).
- For large scale problems: projection method [M. & Spence, 2010]:
  - Project the matrices on the eigenvectors of Z_k: reduced MEP, solvable by QZ method on the Delta matrices
  - Produces a rank one iterate.
  - Sylvester equation is cheaper to solve.
  - Faster convergence thanks to projection (often 2 or 3 iterations).
- The numerical experiments do not use the projection trick since n is small (projected problem is relatively large).

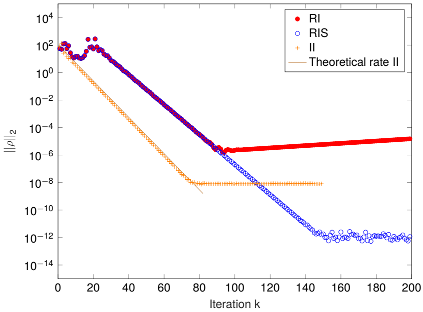
REPv: Numerical example

$$(A + \lambda B + f_1(x)C_1 + f_2(x)C_2))\, x = 0$$

$$f_1(x) = \frac{r_1^T x}{s_1^T x}, f_2(x) = \frac{r_2^T x}{s_2^T x}$$



$\sigma = -1.2769 - 0.0442\imath$



$\sigma = -4 - 3\imath$

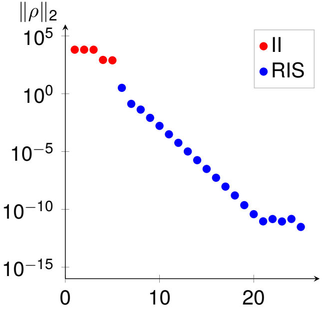
REPv: Numerical example

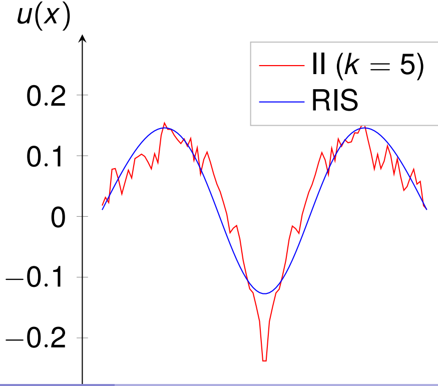
$$u''(x) + \lambda k_1(x)u(x) + f(u)k_2(x)u(x) = 0 \qquad (1)$$

for $x \in [-1, 1]$ with $u(-1) = u(1) = 0$.

$$f(u) = \frac{\int_{-1}^{1} e^{-\gamma x^2} dx(u)}{u'(0)}$$

$k_1(x) = 1 + \frac{1}{2}\tanh(5x)$, $k_2(x) = 1 + \frac{1}{2}\cos(\pi x)$ and $\gamma = 10$.





3. PEPv: polynomial eigenvalue problem with vector nonlinearity

Definition (PEPv)

Given a matrix valued function

$$T : \mathbb{C}^n \times \mathbb{C} \to \mathbb{C}^{n\times n}, (x, \lambda) \to T(x, \lambda)$$

of homogeneous polynomials in x of degree $d > 0$.
PEPv:

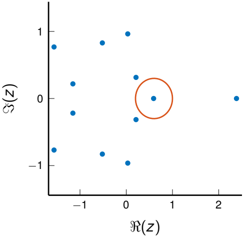
$$T(x, \lambda)x = 0$$

Due to the homogeneity of x, eigenpairs (λ, x) are in $\mathbb{C} \times \mathbb{P}^{n-1}$.

Example ($n = 3, d = 1$)

$$T(x,\lambda)\cdot x = \begin{bmatrix} x_1 + \lambda x_2 & \lambda x_2 + x_3 & x_1 - x_3 \\ x_1 + (1+\lambda)x_2 & (1-\lambda^2)x_2 - \lambda x_3 & x_1 + x_3 \\ (1+\lambda)x_1 + x_2 & x_2 - x_3 & \lambda x_1 + (1-\lambda)x_3 \end{bmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

Contour integration methods



- For NEP $\det A(\lambda) = 0$: Keldysh theorem (in the case of simple eigenvalues $\lambda_1, \ldots, \lambda_\ell$ inside Ω):

$$A(\lambda)^{-1} = \sum_{j=1}^{\ell} \frac{x_j y_j^*}{\lambda - \lambda_j} + R(\lambda).$$

- Sakurai and coworkers: Hankel matrix (linear eigenvalue problem) [Sakurai&Sugiura, 2003], [Van Barel&Kravanja, 2016]
- FEAST [Polizzi, 2009]
- Kind of subspace iteration method [Beyn, 2012]
- All methods require the computation of

$$A_k = \frac{1}{2\pi\imath} \oint_{z \in \partial\Omega} z^k A(z)^{-1} V dz$$

with $V \in \mathbb{C}^{n \times m}$, m vectors of size n.

Contour integration methods

- In practice, use quadrature rule:

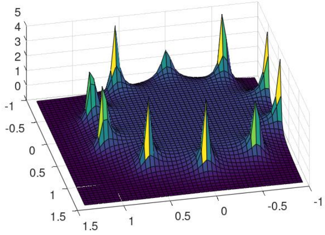
$$A_k = \frac{1}{2\pi\imath}\oint_{z\in\partial\Omega} z^k A(z)^{-1} V dz \approx \frac{1}{2\pi\imath}\sum_{i=1}^{N} z_i^k A(z_i)^{-1} V w_i$$

- Beyn, 2012 suggests to solve rectangular eigenvalue problem

$$A_1 x = \lambda A_0 x \qquad A_1, A_0) \in \mathbb{C}^{n\times m}$$

- Due to their structure they are the eigenvalues of

$$(A_0^T A_1 x = \lambda (A_0^T A_0) x$$



- Quadrature rule acts as a rational filter on the subspace spanned by the columns of V:

$$\sum_{i=1}^{N} \frac{z_i^k}{\lambda - z_i} w_i$$

Polynomial systems of equations

Consider

$$T(x,z)\cdot x = \begin{pmatrix} x_1 + zx_2 & zx_2 + x_3 & x_1 - x_3 \\ x_1 + (1+z)x_2 & (1-z^2)x_2 - zx_3 & x_1 + x_3 \\ (1+z)x_1 + x_2 & x_2 - x_3 & zx_1 + (1-z)x_3 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} =$$

Write the polynomials as a matrix

$$\left[\begin{array}{cccccc} x_1^2 & x_2^2 & x_3^2 & x_1x_2 & x_1x_3 & x_2x_3 \\ \hline 1 & z & -1 & z & 1 & z \\ 1 & 1-z^2 & 1 & 1+z & 1 & -z \\ 1+z & 1 & 1-z & 1 & z & -1 \end{array}\right]$$

Resultants

- For PEPv, take the example of monomials of degree two. Write the polynomials as a matrix

$$\left[\begin{array}{cccccc} x_1^2 & x_2^2 & x_3^2 & x_1x_2 & x_1x_3 & x_2x_3 \\ \hline 1 & z & -1 & z & 1 & z \\ 1 & 1-z^2 & 1 & 1+z & 1 & -z \\ 1+z & 1 & 1-z & 1 & z & -1 \end{array}\right]$$

- From this matrix a 6×6 determinant is deduced which produces the resultant

$$\mathcal{R}(z) = 4z^{12} + 12z^{11} - z^{10} - 53z^9 - 100z^8 - 108z^7 \\ - 78z^6 - 23z^5 + 14z^4 + 22z^3 + 8z^2 - 4z + 3$$

- The resultant $\mathcal{R}(z)$ is a generalization of the determinant for a NEP: $\mathcal{R}(z) = \det A(z)$.

Traces

Definition

Let $V(a)$ be the set of δ solutions to the system $T(x,z)x - a = 0$ with a a polynomial in x and a given $z \in \mathbb{C}$.
The trace (vector) is

$$\mathrm{Tr}(a) = \sum_{v \in V(a)} \mu(v) v$$

with $\mu(v)$ the multiplicity of v.

Theorem

$$Tr(a) = \frac{\mathcal{Q}_a(z)}{\mathcal{R}(z)\mathcal{S}_a(z)}$$

with $\mathcal{S}_a(z)$ a nonzero polynomial.

Traces

$$\text{Tr}(a) = \frac{\mathcal{Q}_a(z)}{\mathcal{R}(z)\mathcal{S}_a(z)}$$

- For the NEPv, $\mathcal{S}_a(z)$ is 'often' a nonzero complex constant. (choice of a)
- Toric eigenvector:
  - A toric eigenvector is an eigenvector x with all $x_i \neq 0$, $i = 1, \ldots, n$.
  - If an eigenvalue λ of $T(z, x)$ has a simple toric eigenvector x, then $\mathcal{R}(\lambda) = 0$ and $\mathcal{Q}_{i,a}(\lambda) \neq 0$ for some $i = 1, \ldots, n$.
  - These eigenvalues are poles of $\text{Tr}(a)$.

Trace example

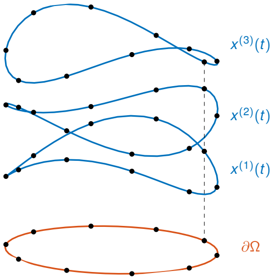
$$T(x,z) = \begin{pmatrix} 1 & z & 1 \\ 2 & 1 & z \\ x_2 & (z+1)x_3 + x_2 & 0 \end{pmatrix}$$

and $a = [a_1, a_2, b_{31}x_1 + b_{32}x_2 + b_{33}x_3]^T$
Then, the first component of the trace vector

$$\mathrm{Tr}_1(a) = \frac{\begin{array}{c} b_{31}z^4 + (a_1 + a_2 - b_{32} - 2b_{33})z^3 + \cdots \\ +(2a_1 + 4a_2 + b_{31} - 2b_{32} - b_{33}) \end{array}}{(z^2 + 2z - 2)(z - 2)}$$

- $\mathcal{R}(z) = z^2 + 2z - 2.$
- $\mathcal{S}_a(z) = z - 2.$

Trace practice



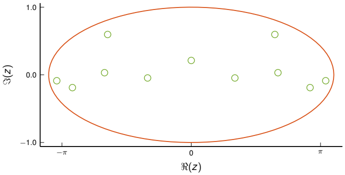
- number of paths for random a: $(d+1)^n - d^n$
- number of paths for monomial a: $(d+1)^{n-1}$

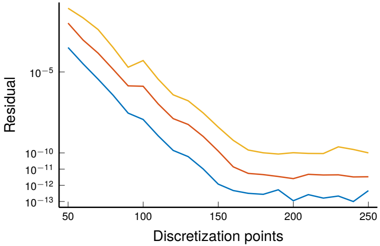
- Use trapezoidal rule with N points on the contour
- Use homotopy continuation for following the path(s) along the contour (Davidenko equation plus Newton iteration)
- Initial set of δ solution determined using polyhedral homotopies [Huber&Sturmfells, 1995]
- Beyn's method with $V \in \mathbb{C}^{n\times n}$, i.e., choose n polynomials a.

$$A(z)^{-1}V = \begin{pmatrix} \mathrm{Tr}(a_1) & \cdots & \mathrm{Tr}(a_n) \end{pmatrix}$$

Numerical example

$$A(v,\lambda) = \begin{bmatrix} v_1^2 v_2 & -2\imath v_1^2 v_2 \cos(\lambda) \\ -v_2^2 \cos(\lambda^2) & 2v_2^2 \sin(3\lambda) \end{bmatrix}$$





- Good selection of a: random polynomials with the same degree as $A(v,\lambda)$.
- Number of paths by good selection of a: 3 instead of 5.

Conclusions

- Difficult computational problem from algebraic geometry.
- Reliability in eigenvalue solvers is very important.
- Globally convergent methods are the goal.
- First steps:
  - Secant SCF (local approximation)
  - REPv of special form
  - Trace function