MANCHESTER 1824

The University of Manchester

M\cr NA Manchester Numerical Analysis

The Nonlinear Eigenvalue Problem Part III

Françoise Tisseur

School of Mathematics
The University of Manchester

ESSAM, Kácov, 2018

Outline

Two lectures:

- Part I: Mathematical properties of nonlinear eigenproblems (NEPs)
  - Definition and historical aspects
  - Examples and applications
  - Solution structure
- Part II and Part III: Numerical methods for NEPs
  - Solvers based on Newton's method
  - Solvers using contour integrals
  - Linear interpolation methods

S. GÜTTEL AND F. TISSEUR, *The nonlinear eigenvalue problem*. Acta Numerica 26:1–94, 2017.

Methods based on linear interpolation

Instead of solving $F(\lambda)v = 0$ directly, we may approximate F by simpler function R_m on $\Sigma \subseteq \mathbb{C}$ and solve $R_m(\lambda)v = 0$.

Methods based on linear interpolation

Instead of solving $F(\lambda)v = 0$ directly, we may approximate F by simpler function R_m on $\Sigma \subseteq \mathbb{C}$ and solve $R_m(\lambda)v = 0$.

Practical approach: Polynomial or rational form

$$R_m(z) = b_0(z)D_0 + b_1(z)D_1 + \cdots + b_m(z)D_m$$

where

- $D_j \in \mathbb{C}^{n\times n}$ are constant coefficient matrices and
- b_j are polynomials or rational functions of type (m, m).

Methods based on linear interpolation

Instead of solving $F(\lambda)v = 0$ directly, we may approximate F by simpler function R_m on $\Sigma \subseteq \mathbb{C}$ and solve $R_m(\lambda)v = 0$.

Practical approach: Polynomial or rational form

$$R_m(z) = b_0(z)D_0 + b_1(z)D_1 + \cdots + b_m(z)D_m$$

where

- $D_j \in \mathbb{C}^{n\times n}$ are constant coefficient matrices and
- b_j are polynomials or rational functions of type (m, m).

It is crucial that $R_m \approx F$ is some sense, for otherwise eigenpairs of F and R_m are not related.

Basic Properties of Approximant R_m

We impose that the approximation R_m on $\Sigma \subseteq \Omega$ to $F \in H(\Omega, \mathbb{C}^{n\times n})$ satisfies

$$\|F - R_m\|_\Sigma := \max_{z\in\Sigma} \|F(z) - R_m(z)\|_2 \leq \varepsilon,$$

for some small ε.

Basic Properties of Approximant R_m

We impose that the approximation R_m on $\Sigma \subseteq \Omega$ to $F \in H(\Omega, \mathbb{C}^{n\times n})$ satisfies

$$\|F - R_m\|_\Sigma := \max_{z\in\Sigma} \|F(z) - R_m(z)\|_2 \le \varepsilon,$$

for some small ε.

Let $\lambda \in \Sigma$ and $v \in \mathbb{C}^n$ s.t. $\|v\|_2 = 1$ and $R_m(\lambda)v = 0$. Then

$$\|F(\lambda)v\|_2 = \|\big(F(\lambda) - R_m(\lambda)\big)v\|_2 \le \|F(\lambda) - R_m(\lambda)\|_2 \le \varepsilon,$$

i.e., bounded residual.

Ideally, R_m should not any ei'vals in Σ that are in the resolvent set of F (i.e., R_m should be free of spurious ei'vals).

Rational Newton Basis Functions

Approximate $F \in H(\Omega, \mathbb{C}^{n\times n})$ by

$$R_m(z) = b_0(z)D_0 + b_1(z)D_1 + \cdots + b_m(z)D_m \in H(\Sigma, \mathbb{C}^{n\times n}),$$

where $\Sigma \subseteq \Omega$, $D_j \in \mathbb{C}^{n\times n}$ and b_j are rational functions.

Particularly useful: (scaled) rational Newton basis

$$b_0(z) \equiv \frac{1}{\beta_0}, \quad b_{j+1}(z) = \frac{z - \sigma_j}{\beta_{j+1}(1 - z/\xi_{j+1})} b_j(z)$$

with interpolation points $\sigma_j \in \Sigma$, poles $\xi_j \in \overline{\mathbb{C}} \setminus \Sigma$, and scaling factors $\beta_j \neq 0$.

Rational Newton Basis Functions

Approximate $F \in H(\Omega, \mathbb{C}^{n \times n})$ by

$$R_m(z) = b_0(z)D_0 + b_1(z)D_1 + \cdots + b_m(z)D_m \in H(\Sigma, \mathbb{C}^{n \times n}),$$

where $\Sigma \subseteq \Omega$, $D_j \in \mathbb{C}^{n \times n}$ and b_j are rational functions.

Particularly useful: (scaled) rational Newton basis

$$b_0(z) \equiv \frac{1}{\beta_0}, \quad b_{j+1}(z) = \frac{z - \sigma_j}{\beta_{j+1}(1 - z/\xi_{j+1})} b_j(z)$$

with interpolation points $\sigma_j \in \Sigma$, poles $\xi_j \in \overline{\mathbb{C}} \setminus \Sigma$, and scaling factors $\beta_j \neq 0$.

- Shifted and scaled monomials: $\sigma_j = \sigma$, $\xi_j = \infty$.
- Scaled Newton polynomials: $\xi_j = \infty$.

Rational Newton Basis Functions

Approximate $F \in H(\Omega, \mathbb{C}^{n\times n})$ by

$$R_m(z) = b_0(z)D_0 + b_1(z)D_1 + \cdots + b_m(z)D_m \in H(\Sigma, \mathbb{C}^{n\times n}),$$

where $\Sigma \subseteq \Omega$, $D_j \in \mathbb{C}^{n\times n}$ and b_j are rational functions.

Particularly useful: (scaled) rational Newton basis

$$b_0(z) \equiv \frac{1}{\beta_0}, \quad b_{j+1}(z) = \frac{z - \sigma_j}{\beta_{j+1}(1 - z/\xi_{j+1})} b_j(z)$$

with interpolation points $\sigma_j \in \Sigma$, poles $\xi_j \in \overline{\mathbb{C}} \setminus \Sigma$, and scaling factors $\beta_j \neq 0$.

- Shifted and scaled monomials: $\sigma_j = \sigma$, $\xi_j = \infty$.
- Scaled Newton polynomials: $\xi_j = \infty$.

Choice of σ_j, ξ_j, β_j by NLEIGS sampling **[Güttel et al 2014].**

NLEIGS sampling

Assume F is holomorphic on $\Omega = \mathbb{C} \setminus \Xi$ and we target the eigenvalues in $\Sigma \subseteq \Omega$.

Assume we have chosen nodes $\sigma_0, \sigma_1, \ldots, \sigma_m \in \Sigma$ and poles $\xi_1, \ldots, \xi_m \in \Xi$. Define $s_m(z) := (z - \sigma_m) b_m(z)$.

By the **Hermite–Walsh formula** we have

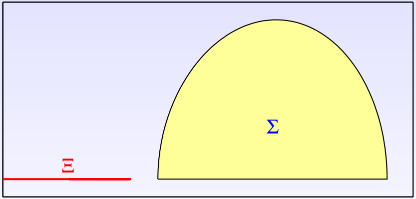
$$F(z) - R_m(z) = \frac{1}{2\pi \mathrm{i}} \int_\Gamma \frac{s_m(z)}{s_m(\zeta)} \frac{F(\zeta)}{\zeta - z} \, \mathrm{d}\zeta,$$

and so the **uniform approximation error on Σ** satisfies

$$\|F - R_m\|_\Sigma := \max_{z \in \Sigma} \|F(z) - R_m(z)\|_2 \leq C \|s_m\|_\Sigma \cdot \|s_m^{-1}\|_\Gamma$$

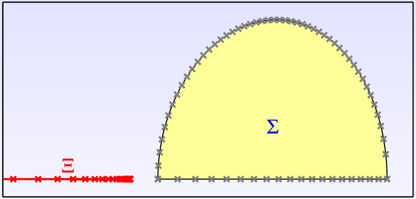
Aim: Make s_m small on Σ and large on Γ.

NLEIGS sampling of F on Σ



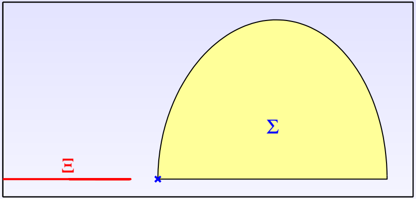
NLEIGS sampling of F on Σ

1. Discretize boundaries of Σ and Ξ sufficiently fine.



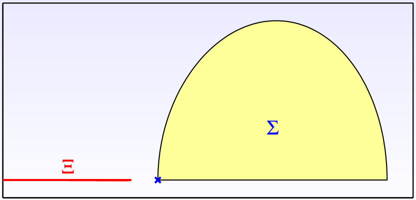
NLEIGS sampling of F on Σ

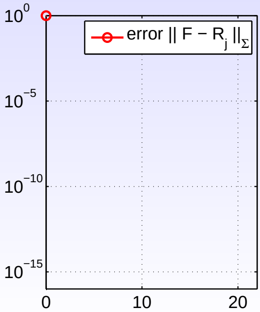
1. Discretize boundaries of Σ and Ξ sufficiently fine.
2. Choose interpolation node $\sigma_0 \in \Sigma$ and set $R_0(z) \equiv F(\sigma_0)$.



NLEIGS sampling of F on Σ

1. Discretize boundaries of Σ and Ξ sufficiently fine.
2. Choose interpolation node $\sigma_0 \in \Sigma$ and set $R_0(z) \equiv F(\sigma_0)$.

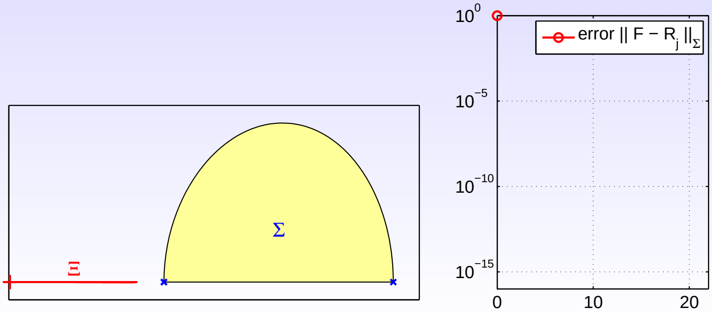




NLEIGS sampling of F on Σ

1. Discretize boundaries of Σ and Ξ sufficiently fine.
2. Choose interpolation node $\sigma_0 \in \Sigma$ and set $R_0(z) \equiv F(\sigma_0)$.
3. For $j = 1, 2, \ldots, m$ choose σ_j and ξ_j such that

$$\max_{z \in \Sigma} |s_{j-1}(z)| = |s_{j-1}(\sigma_j)| \quad \text{and} \quad \min_{z \in \Xi} |s_{j-1}(z)| = |s_{j-1}(\xi_j)|.$$

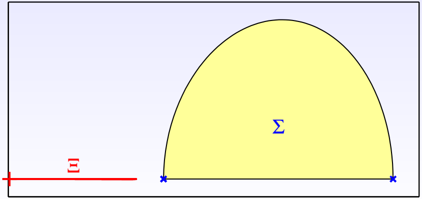


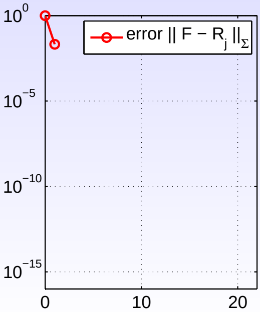
NLEIGS sampling of F on Σ

1. Discretize boundaries of Σ and Ξ sufficiently fine.
2. Choose interpolation node $\sigma_0 \in \Sigma$ and set $R_0(z) \equiv F(\sigma_0)$.
3. For $j = 1, 2, \ldots, m$ choose σ_j and ξ_j such that

$$\max_{z \in \Sigma} |s_{j-1}(z)| = |s_{j-1}(\sigma_j)| \quad \text{and} \quad \min_{z \in \Xi} |s_{j-1}(z)| = |s_{j-1}(\xi_j)|.$$

Choose β_j such that $\|s_j\|_\Sigma = 1$ and set $D_j = \frac{F(\sigma_j) - R_{j-1}(\sigma_j)}{b_j(\sigma_j)}$.



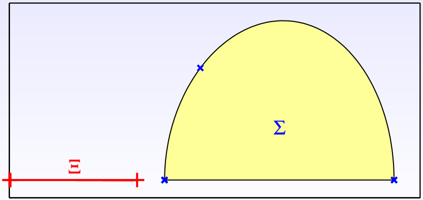


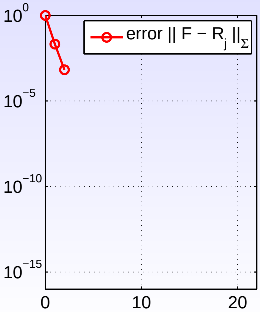
NLEIGS sampling of F on Σ

1. Discretize boundaries of Σ and Ξ sufficiently fine.
2. Choose interpolation node $\sigma_0 \in \Sigma$ and set $R_0(z) \equiv F(\sigma_0)$.
3. For $j = 1, 2, \ldots, m$ choose σ_j and ξ_j such that

$$\max_{z\in\Sigma} |s_{j-1}(z)| = |s_{j-1}(\sigma_j)| \quad \text{and} \quad \min_{z\in\Xi} |s_{j-1}(z)| = |s_{j-1}(\xi_j)|.$$

Choose β_j such that $\|s_j\|_\Sigma = 1$ and set $D_j = \frac{F(\sigma_j) - R_{j-1}(\sigma_j)}{b_j(\sigma_j)}$.



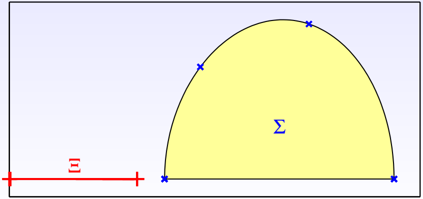


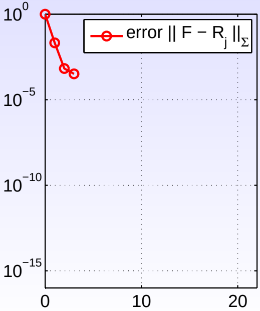
NLEIGS sampling of F on Σ

1. Discretize boundaries of Σ and Ξ sufficiently fine.
2. Choose interpolation node $\sigma_0 \in \Sigma$ and set $R_0(z) \equiv F(\sigma_0)$.
3. For $j = 1, 2, \ldots, m$ choose σ_j and ξ_j such that

$$\max_{z\in\Sigma} |s_{j-1}(z)| = |s_{j-1}(\sigma_j)| \quad \text{and} \quad \min_{z\in\Xi} |s_{j-1}(z)| = |s_{j-1}(\xi_j)|.$$

Choose β_j such that $\|s_j\|_\Sigma = 1$ and set $D_j = \frac{F(\sigma_j)-R_{j-1}(\sigma_j)}{b_j(\sigma_j)}$.



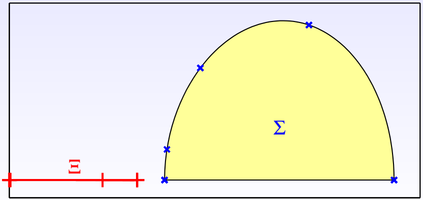


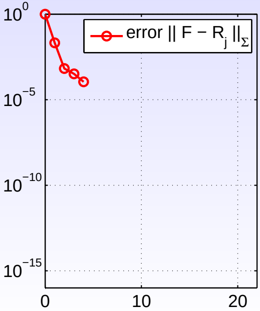
NLEIGS sampling of F on Σ

1. Discretize boundaries of Σ and Ξ sufficiently fine.
2. Choose interpolation node $\sigma_0 \in \Sigma$ and set $R_0(z) \equiv F(\sigma_0)$.
3. For $j = 1, 2, \ldots, m$ choose σ_j and ξ_j such that

$$\max_{z\in\Sigma} |s_{j-1}(z)| = |s_{j-1}(\sigma_j)| \quad \text{and} \quad \min_{z\in\Xi} |s_{j-1}(z)| = |s_{j-1}(\xi_j)|.$$

   Choose β_j such that $\|s_j\|_\Sigma = 1$ and set $D_j = \frac{F(\sigma_j) - R_{j-1}(\sigma_j)}{b_j(\sigma_j)}$.



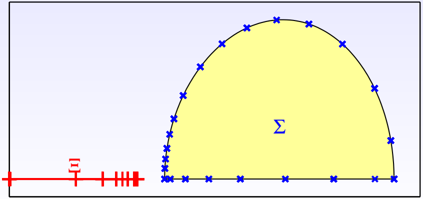


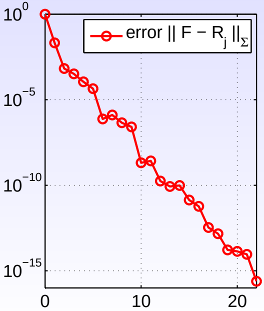
NLEIGS sampling of F on Σ

1. Discretize boundaries of Σ and Ξ sufficiently fine.
2. Choose interpolation node $\sigma_0 \in \Sigma$ and set $R_0(z) \equiv F(\sigma_0)$.
3. For $j = 1, 2, \ldots, m$ choose σ_j and ξ_j such that

$$\max_{z\in\Sigma} |s_{j-1}(z)| = |s_{j-1}(\sigma_j)| \quad \text{and} \quad \min_{z\in\Xi} |s_{j-1}(z)| = |s_{j-1}(\xi_j)|.$$

Choose β_j such that $\|s_j\|_\Sigma = 1$ and set $D_j = \frac{F(\sigma_j) - R_{j-1}(\sigma_j)}{b_j(\sigma_j)}$.



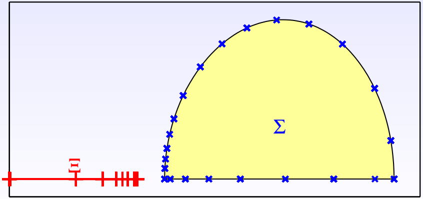


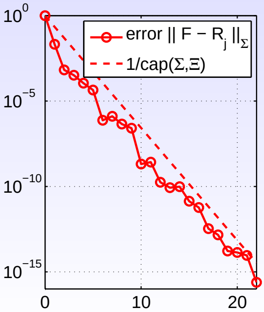
NLEIGS sampling of F on Σ

1. Discretize boundaries of Σ and Ξ sufficiently fine.
2. Choose interpolation node $\sigma_0 \in \Sigma$ and set $R_0(z) \equiv F(\sigma_0)$.
3. For $j = 1, 2, \ldots, m$ choose σ_j and ξ_j such that

$$\max_{z\in\Sigma} |s_{j-1}(z)| = |s_{j-1}(\sigma_j)| \quad \text{and} \quad \min_{z\in\Xi} |s_{j-1}(z)| = |s_{j-1}(\xi_j)|.$$

Choose β_j such that $\|s_j\|_\Sigma = 1$ and set $D_j = \frac{F(\sigma_j) - R_{j-1}(\sigma_j)}{b_j(\sigma_j)}$.





Other Interpolation Techniques

These include

- Chebyshev interpolation: for NEPs with ei'vals located on the real line or on pre-specified curves in $\mathbb{C}$ [Effenberger and Kressner 2012],
- Automatic rational approximation using the adaptive Antoulas-Anderson (AAA) algorithm [Nakatsukasa et al. (2017)], [Lietaert et al. (2018)]:

$$F(\lambda) \approx R_m(\lambda) = P(\lambda) + \sum_{i=1}^{m} (A_i - \lambda B_i) r_i(\lambda),$$

where $P(\lambda)$ is a matrix polynomial and $r_i(\lambda)$ are rational approximants in barycentric form.

Eigenvalue Solution

Interpolation techniques can be combined with

- a linearization of R_m, i.e., $R_m(\lambda)x = 0$ is rewritten as a structured generalized eigenproblem (GEP)

$$\mathcal{A}v = \lambda\mathcal{B}v$$

  of larger dimension.

- Solve the GEP:
  - e.g., QZ algorithm for small dense problems,
  - (rational) Krylov algorithms for large sparse problems.

Polynomial Eigenvalue Problems (PEPs)

Consider $n \times n$ matrix polynomial

$$\begin{aligned} P(\lambda) &= \lambda^d A_d + \cdots + \lambda A_1 + A_0 \text{ (monomial form)} \\ &= \phi_d(\lambda) P_d + \cdots + \phi_1(\lambda) P_1 + \phi_0(\lambda) P_0 \text{ (non-monomial form),} \end{aligned}$$

where $\phi_i(\lambda)$ is a polynomial of degree i.

Polynomial Eigenvalue Problems (PEPs)

Consider $n \times n$ matrix polynomial

$$\begin{aligned} P(\lambda) &= \lambda^d A_d + \cdots + \lambda A_1 + A_0 \text{ (monomial form)} \\ &= \phi_d(\lambda) P_d + \cdots + \phi_1(\lambda) P_1 + \phi_0(\lambda) P_0 \text{ (non-monomial form),} \end{aligned}$$

where $\phi_i(\lambda)$ is a polynomial of degree i.

- Algorithms for dense and large sparse PEPs (monomial form).
- Algorithms for PEPs in non-monomial form.

Condensed Forms

$$P(\lambda) = \sum_{i=0}^{d} \lambda^i A_i, \quad A_i \in \mathbb{C}^{n\times n}, \quad A_d \neq 0.$$

- **Generalized Schur decomposition**: there exist U, V unitary s.t. $U(\lambda A_1 + A_0)V = \lambda T + S$ is upper triangular.
  - $\lambda_j = -s_{jj}/t_{jj}$, $j = 1 : n$.
  - Can be computed by the QZ algorithm.
  - Useful for purging and locking of e'vals, implicit restart, ...
- **No analog of generalized Schur decomposition** when $d > 1$.

Condensed Forms

$$P(\lambda) = \sum_{i=0}^{d} \lambda^i A_i, \quad A_i \in \mathbb{C}^{n\times n}, \quad A_d \neq 0.$$

- **Generalized Schur decomposition**: there exist U, V unitary s.t. $U(\lambda A_1 + A_0)V = \lambda T + S$ is upper triangular.
  - $\lambda_j = -s_{jj}/t_{jj}$, $j = 1: n$.
  - Can be computed by the QZ algorithm.
  - Useful for purging and locking of e'vals, implicit restart, . . .
- **No analog of generalized Schur decomposition** when $d > 1$.

Rewrite $P(\lambda)x = 0$ as a linear eigenproblem $(\mathcal{A} - \lambda\mathcal{B})\xi = 0$ of larger dimension (usually $dn \times dn$).

Standard Solution Process ($d = 2$)

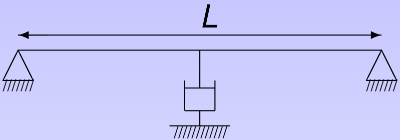
Find all λ and x satisfying $Q(\lambda)x = (\lambda^2 M + \lambda D + K)x = 0$.

- Commonly solved by **linearization**:
  - **Convert** $Q(\lambda)x = 0$ into $(\mathcal{A} - \lambda\mathcal{B})\xi = 0$, e.g.,

$$\mathcal{A} - \lambda\mathcal{B} = \begin{bmatrix} K & 0 \\ 0 & I \end{bmatrix} - \lambda \begin{bmatrix} -D & -M \\ I & 0 \end{bmatrix}, \quad \xi = \begin{bmatrix} x \\ \lambda x \end{bmatrix}.$$

  - **Solve** $(\mathcal{A} - \lambda\mathcal{B})\xi = 0$ with an eigensolver for generalized eigenproblem (e.g., QZ algorithm).
  - **Recover** eigenvectors of $Q(\lambda)$ from those of $\mathcal{A} - \lambda\mathcal{B}$.
- Solution process extend to degree $d > 2$.
- Numerical issues with this process.

Example 2: Beam Problem



Transverse displacement $u(x,t)$

$$\rho A \frac{\partial^2 u}{\partial t^2} + c(x)\frac{\partial u}{\partial t} + EI\frac{\partial^4 u}{\partial x^4} = 0.$$

$$u(0,t) = u''(0,t) = u(L,t) = u''(L,t) = 0.$$

Finite element method leads to

$$Q(\lambda)v = (\lambda^2 M + \lambda D + K)v = 0$$

with symmetric $M, D, K \in \mathbb{R}^{n\times n}$.

- $M > 0$, $K > 0$, $D \geq 0 \;\Rightarrow$ all e'vals have $\mathrm{Re}(\lambda) \leq 0$.
- D is rank 1. Can show n pure imaginary e'vals.

[Higham et al, 2018].

Computed Spectra of C_1, L_1 and L_2

$$C_1(\lambda) = \lambda \begin{bmatrix} M & 0 \\ 0 & I \end{bmatrix} + \begin{bmatrix} D & K \\ -I & 0 \end{bmatrix},$$

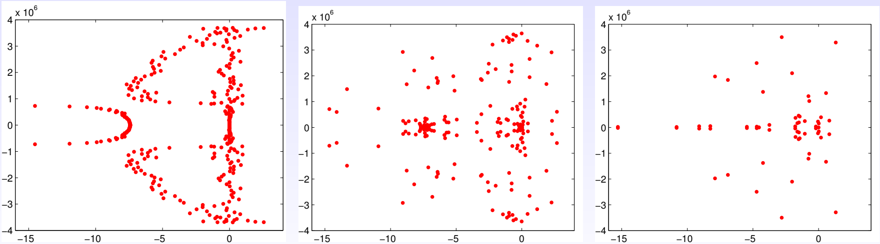
$$L_1(\lambda) = \lambda \begin{bmatrix} M & 0 \\ 0 & -K \end{bmatrix} + \begin{bmatrix} D & K \\ K & 0 \end{bmatrix}, \quad L_2(\lambda) = \lambda \begin{bmatrix} 0 & M \\ M & D \end{bmatrix} + \begin{bmatrix} -M & 0 \\ 0 & K \end{bmatrix}$$

Computed Spectra of C_1, L_1 and L_2

$$C_1(\lambda) = \lambda \begin{bmatrix} M & 0 \\ 0 & I \end{bmatrix} + \begin{bmatrix} D & K \\ -I & 0 \end{bmatrix},$$

$$L_1(\lambda) = \lambda \begin{bmatrix} M & 0 \\ 0 & -K \end{bmatrix} + \begin{bmatrix} D & K \\ K & 0 \end{bmatrix}, \quad L_2(\lambda) = \lambda \begin{bmatrix} 0 & M \\ M & D \end{bmatrix} + \begin{bmatrix} -M & 0 \\ 0 & K \end{bmatrix}$$

$C_1(\lambda)$ $L_1(\lambda)$ $L_2(\lambda)$



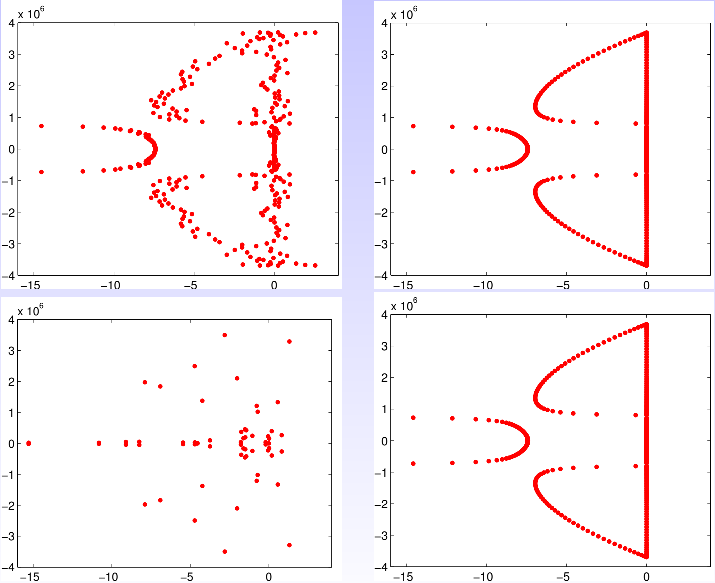
Sensitivity and Stability of Linearizations

- Developed theory concerning the sensitivity and stability of linearizations [Higham, Mackey, T. 06, Higham, Li, T. 07, Grammont, Higham, T., 11].
- Importance of **scaling** QEPs/PEPs before computing e'vals via linearization.
  - Eigenvalue parameter scaling:

$$\lambda = \gamma\mu, \quad \widetilde{Q}(\mu) := \delta Q(\gamma\mu).$$

  - Does not affect sparsity of matrix coeffs.
  - γ, δ chosen to improve growth factors $\kappa_L(\lambda)/\kappa_Q(\lambda)$ (conditioning), $\eta_Q(z_i, \lambda)/\eta_L(z, \lambda)$ (backward error).

Spectrum of C_1, L_2 Before/After Scaling



quadeig for $Q(\lambda) = \lambda^2 M + \lambda D + K$.

- Eigensolver for dense (small to medium size) quadratics—**quadeig**.
- Incorporates:
  - Appropriate choice of linearization: uses $\begin{bmatrix} D & -I \\ K & 0 \end{bmatrix} - \lambda \begin{bmatrix} -M & 0 \\ 0 & -I \end{bmatrix}$.
  - Deflation of 0 and ∞ eigenvalues.
  - Eigenvalue parameter scaling (FLV/tropical).
  - Careful recovery of the eigenvectors.
  - Backward stable when $\|D\| \lesssim (\|M\|\|K\|)^{1/2}$.
- MATLAB and Fortran implementations (NAG, LAPACK) [Hammarling, Munro, T. 2013].

Eigensolvers for Large Sparse QEPs

- Second Order Arnoldi (SOAR) method [Bai & Su, 2005]. Projection applied directly to quadratic.
- Quadratic Arnoldi method [Meerbergen, 2008].
- Two-level Orthogonal Arnoldi method [Lu, Su, Bai, 2016].

Eigensolvers for Large Sparse QEPs

- Second Order Arnoldi (SOAR) method [Bai & Su, 2005]. Projection applied directly to quadratic.
- Quadratic Arnoldi method [Meerbergen, 2008].
- Two-level Orthogonal Arnoldi method [Lu, Su, Bai, 2016].

- See Ivana Šain Glibic 's talk for implicit restart.
- Extend to higher degree matrix polynomials.

PEPs in Non-monomial Form

Let $P(\lambda) = P_0\phi_0(\lambda) + P_1\phi_1(\lambda) + \cdots + P_d\phi_d(\lambda)$, where $P_k \in \mathbb{C}^{n\times n}$ and the $\phi_k(\lambda)$ are rational functions/polynomials.

- shifted and scaled monomials,

$$\phi_0(\lambda) = 1/\beta_0, \quad \phi_j(\lambda) = s - \sigma\phi_{j-1}(\lambda)/\beta_j, \quad j \geq 1,$$

- orthogonal polynomials, $\phi_{-1}(\lambda) = 0$, $\phi_0(\lambda) = 1$,

$$s\phi_j(\lambda) = \alpha_j\phi_{j+1}(\lambda) + \beta_j\phi_j(\lambda) + \gamma_j\phi_{j-1}(\lambda), \quad j \geq 0,$$

- rational Newton basis functions,

$$\phi_0(\lambda) = \frac{1}{\beta_0}, \quad \phi_j(\lambda) = \frac{s - \sigma_j}{\beta_j(1 - s/\zeta_j)}\phi_{j-1}(\lambda), \quad j \geq 1,$$

- ....

Basis Functions Matrix Relation

The linear recurrence relation between the basis functions can be rewritten in matrix form as

$$M_d \, \Phi_d(\lambda) = \lambda N_d \, \Phi_d(\lambda),$$

where

$$\Phi_d(\lambda) = [\phi_0(\lambda), \phi_1(\lambda), \ldots, \phi_d(\lambda)]^T$$

and $M_d, N_d \in \mathbb{C}^{d \times (d+1)}$ are well-defined matrices. E.g., for rational Newton basis,

$$M_d = \begin{bmatrix} \sigma_1 & \beta_1 & & & \\ & \sigma_2 & \beta_2 & & \\ & & \ddots & \ddots & \\ & & & \sigma_d & \beta_d \end{bmatrix}, N_d = \begin{bmatrix} 1 & \beta_1/\zeta_1 & & & \\ & 1 & \beta_2/\zeta_2 & & \\ & & \ddots & \ddots & \\ & & & 1 & \beta_d/\zeta_d \end{bmatrix}.$$

CORK Pencil

Rewrite $P(\lambda)$ as

$$g(\lambda)P(\lambda) = \sum_{j=0}^{d-1}(A_j - \lambda B_j)\phi_j(\lambda) = (\mathbf{A} - \lambda\mathbf{B})(\Phi_{d-1}(\lambda) \otimes I_n),$$

- A_j, B_j depend on coeffs P_j of $P(\lambda)$,
- $g(\lambda) = 1$ for poly. bases, $g(\lambda) = (1 - \frac{\lambda}{\zeta_d})$ for rat. basis,
- $\mathbf{A} - \lambda\mathbf{B} = [\, A_0 - \lambda B_0 \quad \cdots \quad A_{d-1} - \lambda B_{d-1} \,] \in \mathbb{C}^{n \times nd}$,

The **CORK pencil**

$$\mathcal{A} - \lambda\mathcal{B} = \begin{bmatrix} \mathbf{A} - \lambda\mathbf{B} \\ (M_{d-1} - \lambda N_{d-1}) \otimes I_n \end{bmatrix} \in \mathbb{C}^{nd \times nd}$$

satisfies $(\mathcal{A} - \lambda\mathcal{B})(\Phi_{d-1}(\lambda) \otimes I_n) = e_1 \otimes g(\lambda)P(\lambda)$.

[Van Beeumen et al, 2015]

Properties of CORK Pencil

The **CORK pencil**

$$\mathcal{A} - \lambda\mathcal{B} = \begin{bmatrix} \mathbf{A} - \lambda\mathbf{B} \\ (M_{d-1} - \lambda N_{d-1}) \otimes I_n \end{bmatrix} \in \mathbb{C}^{nd\times nd}$$

satisfies right- and left-sided factorizations:

$$\begin{aligned} (\mathcal{A} - \lambda\mathcal{B})(\Phi_{d-1}(\lambda) \otimes I_n) &= e_1 \otimes g(\lambda)P(\lambda), \\ H(\lambda)(\mathcal{A} - \lambda\mathcal{B}) &= e_1^T \otimes g(\lambda)P(\lambda). \end{aligned}$$

Properties of CORK Pencil

The **CORK pencil**

$$\mathcal{A} - \lambda\mathcal{B} = \begin{bmatrix} \mathbf{A} - \lambda\mathbf{B} \\ (M_{d-1} - \lambda N_{d-1}) \otimes I_n \end{bmatrix} \in \mathbb{C}^{nd \times nd}$$

satisfies right- and left-sided factorizations:

$$(\mathcal{A} - \lambda\mathcal{B})(\Phi_{d-1}(\lambda) \otimes I_n) = e_1 \otimes g(\lambda)P(\lambda),$$
$$H(\lambda)(\mathcal{A} - \lambda\mathcal{B}) = e_1^T \otimes g(\lambda)P(\lambda).$$

- If $\phi_{d-1}(\lambda) \neq 0$ at λ then $\Phi_{d-1}(\lambda) \otimes x$ is a right ei'vec of $\mathcal{A} - \lambda\mathcal{B}$ with ei'val λ iff x is a right ei'vec of $P(\lambda)$ with ei'val λ.
- If $H(\lambda)$ has full rank then $\begin{bmatrix} y \\ R_\phi(\lambda)y \end{bmatrix}$ is a left ei'vec of $\mathcal{A} - \lambda\mathcal{B}$ with ei'val λ iff y is a left ei'vec of $P(\lambda)$ with ei'val λ.

Rational Arnoldi Algorithm

Given $\mathcal{A}, \mathcal{B} \in \mathbb{C}^{dn \times dn}$, $\mathbf{v} \in \mathbb{C}^{dn} \setminus \{\mathbf{0}\}$, shifts $(\tau_j)_{j=1}^{k} \subset \mathbb{C}$.
Set $\mathbf{v}_1 := \mathbf{v}/\|\mathbf{v}\|_2$.
for $j = 1, 2, \ldots, k$
  Compute $\mathbf{w} := (\mathcal{A} - \tau_j \mathcal{B})^{-1} \mathcal{B} \mathbf{v}_j$.
  Orthogonalize $\widehat{\mathbf{w}} := \mathbf{w} - \sum_{i=1}^{j} \mu_{i,j} \mathbf{v}_i$, where $\mu_{i,j} = \mathbf{v}_i^* \mathbf{w}$.
  Set $\mu_{j+1,j} = \|\widehat{\mathbf{w}}\|_2$ and normalize $\mathbf{v}_{j+1} := \mathbf{w}/\mu_{j+1,j}$.
end

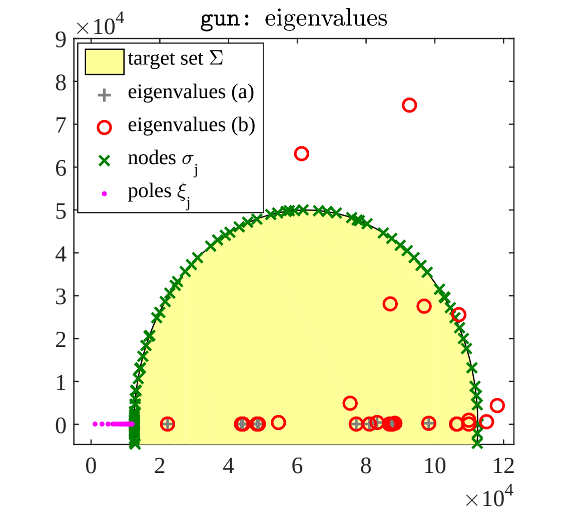
Rational Arnoldi Algorithm

Given $\mathcal{A}, \mathcal{B} \in \mathbb{C}^{dn \times dn}$, $\mathbf{v} \in \mathbb{C}^{dn} \setminus \{\mathbf{0}\}$, shifts $(\tau_j)_{j=1}^{k} \subset \mathbb{C}$.
Set $\mathbf{v}_1 := \mathbf{v}/\|\mathbf{v}\|_2$.
for $j = 1, 2, \ldots, k$
  Compute $\mathbf{w} := (\mathcal{A} - \tau_j \mathcal{B})^{-1} \mathcal{B} \mathbf{v}_j$.
  Orthogonalize $\widehat{\mathbf{w}} := \mathbf{w} - \sum_{i=1}^{j} \mu_{i,j} \mathbf{v}_i$, where $\mu_{i,j} = \mathbf{v}_i^* \mathbf{w}$.
  Set $\mu_{j+1,j} = \|\widehat{\mathbf{w}}\|_2$ and normalize $\mathbf{v}_{j+1} := \mathbf{w}/\mu_{j+1,j}$.
end

If $\mathcal{A} - \lambda \mathcal{B}$ is a CORK pencil then $\mathbf{v}_j = (I_{d+1} \otimes Q_j)\mathbf{u}_j$, with $\mathbf{u}_j$ of smaller length $\mathbf{v}_j$.

Gun Problem

$$F(\lambda)v = \left[K - \lambda M + i(\lambda - \sigma_1^2)^{\frac{1}{2}} W_1 + i(\lambda - \sigma_2^2)^{\frac{1}{2}} W_2\right] v = 0.$$



Krylov solution

- dynamic increase of degree m during Krylov iteration
  $\Longrightarrow$ infinite Arnoldi method [Jarlebring et al 2012]
- compact Krylov basis storage of [Lu, Su, Bai 2016]
  $\Longrightarrow$ CORK [Van Beeumen et al 2015].

Krylov solution

- dynamic increase of degree m during Krylov iteration
  $\Longrightarrow$ infinite Arnoldi method [Jarlebring et al 2012]
- compact Krylov basis storage of [Lu, Su, Bai 2016]
  $\Longrightarrow$ CORK [Van Beeumen et al 2015].

NLEIGS implementations are available in the

- SLEPc library version 3.7 [Campos & Roman 2016]
- Rational Krylov Toolbox [Berljafa & Güttel 2015].

S. GÜTTEL, R. VAN BEEUMEN, K. MEERBERGEN, W. MICHIELS, *NLEIGS: A class of fully rational Krylov methods for nonlinear eigenvalue problems,* SIAM J. Sci. Comput., 2014.

V. HERNANDEZ, J. E. ROMAN, V. VIDAL, *SLEPc: A scalable and flexible toolkit for the solution of eigenvalue problems*, ACM TOMS, 2005.

Concluding Remarks

NEPs have interesting mathematical properties. They arise in many applications and their efficient solution requires ideas from numerical linear algebra, complex analysis, and approximation theory (among other fields).

Concluding Remarks

NEPs have interesting mathematical properties. They arise in many applications and their efficient solution requires ideas from numerical linear algebra, complex analysis, and approximation theory (among other fields).

There is more to be said, e.g.,

- Structured NEPs?
- Higher-order integral moments
- Preconditioning/scaling of linearizations
- Implementation, software packages

References I

Z. Bai and Y. Su.

SOAR: a second-order Arnoldi method for the solution of the quadratic eigenvalue problem.

SIAM J. Matrix Anal. Appl., 26(3):640–659, 2005.

M. Berljafa and S. Güttel.

A Rational Krylov Toolbox for MATLAB.

MIMS EPrint 2014.56, Manchester Institute for Mathematical Sciences, The University of Manchester, UK, 2014.

8 pp.

Available for download at `http://rktoolbox.org/`.

References II

C. Campos and J. E. Roman.

Parallel iterative refinement in polynomial eigenvalue problems.

Numer. Linear Algebra Appl., 23(4):730–745, 2016.

C. Effenberger and D. Kressner.

Chebyshev interpolation for nonlinear eigenvalue problems.

BIT, 52(4):933–951, 2012.

L. Grammont, N. J. Higham, and F. Tisseur.

A framework for analyzing nonlinear eigenproblems and parametrized linear systems.

Linear Algebra Appl., 435(3):623–640, 2011.

References III

S. Güttel and F. Tisseur.

The nonlinear eigenvalue problem.

In *Acta Numerica*, volume 26, pages 1–94. Cambridge University Press, 2017.

S. Güttel, R. Van Beeumen, K. Meerbergen, and W. Michiels.

NLEIGS: A class of fully rational Krylov methods for nonlinear eigenvalue problems.

SIAM J. Sci. Comput., 36(6):A2842–A2864, 2014.

S. Hammarling, C. J. Munro, and F. Tisseur.

An algorithm for the complete solution of quadratic eigenvalue problems.

ACM Trans. Math. Software, 39(3):18:1–18:19, Apr. 2013.

References IV

V. Hernandez, J. E. Roman, and V. Vidal.
SLEPc: A scalable and flexible toolkit for the solution of eigenvalue problems.
ACM Transactions on Mathematical Software (TOMS), 31(3): 351–362, 2005.

N. J. Higham, R.-C. Li, and F. Tisseur.
Backward error of polynomial eigenproblems solved by linearization.
SIAM J. Matrix Anal. Appl., 29(4):1218–1241, 2007.

N. J. Higham, D. S. Mackey, and F. Tisseur.
The conditioning of linearizations of matrix polynomials.
SIAM J. Matrix Anal. Appl., 28(4):1005–1028, 2006.

References V

N. J. Higham, D. S. Mackey, F. Tisseur, and S. D. Garvey.
Scaling, sensitivity and stability in the numerical solution of quadratic eigenvalue problems.
Internat. J. Numer. Methods Eng., 73(3):344–360, 2008.

E. Jarlebring, K. Meerbergen, and W. Michiels.
Computing a partial Schur factorization of nonlinear eigenvalue problems using the infinite Arnoldi method.
SIAM J. Matrix Anal. Appl., 35(2):411–436, 2014.

References VI

P. Lietaertand, J. Pérez, B. Vandereycken, and K. Meerbergen.

Automatic rational approximation and linearization of nonlinear eigenvalue problems.

Technical Report arXiv:1801.08622v2, 2018.

D. Lu, Y. Su, and Z. Bai.

Stability analysis of the two-level orthogonal Arnoldi procedure.

SIAM J. Matrix Anal. Appl., 37(1):195–214, 2016.

K. Meerbergen.

Fast frequency response computation for Rayleigh damping.

Internat. J. Numer. Methods Eng., 73(1):96–106, 2008.

References VII

Y. Nakatsukasa, O. SÃČÅąte, and L. N. Trefethen.
The aaa algorithm for rational approximation.
SIAM J. Sci. Comput., 40(3):A1494–A1522, 2018.

R. Van Beeumen, K. Meerbergen, and W. Michiels.
Compact rational Krylov methods for nonlinear eigenvalue problems.
SIAM J. Matrix Anal. Appl., 36(2):820–838, 2015.