

Advances in Algebraic Nonlinear Eigenvalue Problems

Zhaojun Bai
University of California, Davis

with the assistance of Ding Lu of University of Geneva

Lecture notes prepared for LSEC Summer School, July 24 – August 5, 2017
(Version August 5, 2017)

Outline

- ▶ Part 1. Linear eigenvalue problems
- ▶ Part 2. Nonlinear eigenvalue problems
- ▶ Part 3. Eigenvalue problems with eigenvector nonlinearity

Outline, cont'd

- ▶ Part 1. Linear eigenvalue problems
 1. Accelerated subspace iteration
 2. Steepest descent method
 3. Arnoldi method
 4. Rational Krylov method
 5. Topics of more recent interest
- ▶ Part 2. Nonlinear eigenvalue problems
 1. Essential theory
 2. Methods based on Newton iteration
 3. Methods specially designed for QEP and REP
 4. Methods based on approximation and linearization
 5. Of things not treated
- ▶ Part 3. Eigenvalue problems with eigenvector nonlinearity
 1. Kohn-Sham density functional theory
 2. Sum of trace ratio
 3. Robust Rayleigh quotient optimization
 4. Of things not treated

Part 1: Linear eigenvalue problems

Getting started

1. (Standard) Linear eigenvalue problems

$$Ax = \lambda x,$$

where $A \in \mathbb{C}^{n \times n}$

2. Generalized linear eigenvalue problems

$$Ax = \lambda Bx,$$

where $A, B \in \mathbb{C}^{n \times n}$

3. Generalized Hermitian definite linear eigenvalue problems

$$Ax = \lambda Bx$$

where $A, B \in \mathbb{C}^{n \times n}$ and $A^* = A$ and $B^* = B > 0$

4. Textbooks and monographs, for examples,

- ▶ B. Parlett, *The Symmetric Eigenvalue Problem* (revised edition), SIAM, 1998 (first edition, ~ 1982)
- ▶ Y. Saad, *Numerical Methods for Large Eigenvalue Problems* (revised edition), SIAM, 2011 (first edition, 1992)
- ▶ G. W. Stewart, *Matrix Algorithms, Vol. II: Eigensystems*, SIAM, 2001
- ▶ G. Golub and C. Van Loan, *Matrix Computations* (4th Ed.), John Hopkins University Press, 2013. (Chapters 7, 8, 10).
- ▶ J. Demmel, *Applied Numerical Linear Algebra*, SIAM, 1997 (Chapters 4, 5, 7)
- ▶ G. W. Stewart and J.-G. Sun, *Matrix Perturbation Theory*, Academic Press, 1990.
- ▶ J.-G. Sun, *Matrix Perturbation Analysis* (2nd edition), Science Press, 2001 (in Chinese).

- ▶ Textbooks in Chinese

- ▶ Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. van der Vorst (editors). *Templates for the solution of Algebraic Eigenvalue Problems: A Practical Guide*. SIAM, Philadelphia, 2000. available at <http://web.cs.ucdavis.edu/~bai/ET/contents.html>

Outline of Part 1

1. Accelerated subspace iteration
2. Steepest descent method
3. Arnoldi method
4. Rational Krylov method
5. Topics of more recent interest
 - (a) Computing many eigenpairs of a Hermitian matrix
 - (b) Solving “ill-conditioned” generalized symmetric definite eigenvalue problems

Part 1.1 Accelerated subspace iteration

1. Consider the generalized Hermitian definite eigenvalue problem

$$Ax = \lambda Bx$$

where $A, B \in \mathbb{C}^{n \times n}$, $A^* = A$, and $B^* = B > 0$. (λ, x) is an *eigenpair* of the pencil $A - \lambda B$.

2. **Eigenvalue decomposition**: there exists an $n \times n$ nonsingular matrix X , such that

$$AX = BX\Lambda \quad \text{and} \quad X^*BX = I,$$

where Λ is a real diagonal matrix, and X is called B -orthogonal. Each diagonal entry λ of Λ with its corresponding vector x of X constitute an eigenpair of the matrix pencil $A - \lambda B$.

3. Mathematically, determining eigenpairs for the generalized eigenproblem of (A, B) is equivalent to determining eigenpairs of the single matrix $B^{-1}A$. Define

$$M \stackrel{\text{def}}{=} B^{-1}A = X\Lambda X^{-1} (= X\Lambda X^*B).$$

4. Accelerated subspace iteration with Rayleigh-Ritz projection

- 1: choose vector $Q_0 \in \mathbb{C}^{n \times k}$ ($Q_0^* Q_0 = I$)
- 2: **for** $j = 1, 2, \dots$ **do**
- 3: compute $Y_j = \rho(M) Q_{j-1}$
- 4: compute $\hat{A}_j = Y_j^* A Y_j$ and $\hat{B}_j = Y_j^* B Y_j$
- 5: compute the eigen-decomposition $\hat{A}_j \hat{X}_j = \hat{B}_j \hat{X}_j \hat{\Lambda}_j$ and $\hat{X}_j^* \hat{B}_j \hat{X}_j = I$
- 6: set $Q_j = Y_j \hat{X}_j$
- 7: test for convergence of approximate eigenpairs $(\hat{\Lambda}_j, Q_j)$
- 8: **end for**

5. Acceleration (filter) functions $\rho(\cdot)$:

- (a) *Ideal* accelerator/filter: spectral projector $\rho(M) = X_J X_J^* B$, where X_J is the set of columns from the eigenvector matrix X corresponding to the eigenvectors of interest. (Ex. verify that it converges in one-step!)
- (b) Classical subspace iteration: $\rho(M) = M$
- (c) Multiple-step subspace iteration: $\rho(M) = M^q$ for some integer $q > 1$
- (d) Chebyshev subspace iteration: $\rho(M) = p_q(M)$, where $p_q(\lambda)$ is a scaled Chebyshev polynomial of degree q of the first kind.
- (e) Rational filters \rightsquigarrow contour integral (FEAST)
 - ▶ P. T. P. Tang and E. Polizzi, FEAST as a subspace iteration eigensolver accelerated by approximate spectral projection, SIMAX, 35, pp.354-390, 2014
- (f) ARRABIT (Augmented Rayleigh-Ritz And Block ITERation)
 - ▶ Z. Wen and Y. Zhang, Block algorithms with augmented Rayleigh-Ritz projections for large-scale eigenpair computation, arXiv:1507.06078v1, July 22, 2015.
- (g) Zolotarev's best rational function approximation of the signum function
 - ▶ Y. Li and H. Yang, Spectrum slicing for sparse Hermitian definite matrices based on Zolotarev's functions, arXiv:1701.08935v2, Feb. 28, 2017

6. MATLAB script: `demo_RayleighRitz.m`

Part 1.2 Steepest descent method

1. Consider the generalized Hermitian definite eigenvalue problem $Ax = \lambda Bx$, let eigenvalues $\{\lambda_i\}$ be ordered such that $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$
2. **Rayleigh quotient:** $\rho(x) = \frac{x^* Ax}{x^* Bx}$
3. Courant-Fischer min-max principle:

$$\lambda_i = \min_{\mathcal{X}, \dim(\mathcal{X})=i} \max_{x \in \mathcal{X}} \rho(x)$$

In particular, $\lambda_1 = \min_{x \in \mathbb{C}^n} \rho(x)$

4. Ky-Fan trace-min principle:

$$\sum_{i=1}^k \lambda_i = \min_{\substack{X \in \mathbb{R}^{n \times k} \\ X^* B X = I}} \text{trace}(X^* A X)$$

5. Cauchy interlacing property: Let $\mu_1 \leq \mu_2 \leq \dots \leq \mu_k$ be the eigenvalues of the pencil $W^* A W - \lambda W^* B W$, where $W \in \mathbb{C}^{n \times k}$ and $\text{rank}(W) = k$, then

$$\lambda_j \leq \mu_j \leq \lambda_{j+n-k} \quad \text{for } 1 \leq j \leq k.$$

6. The Steepest Descent (SD) is a general technique to solve $\min_x f(x)$
7. Recall $\lambda_1 = \min_x \rho(x)$
- ▶ Gradient $\nabla \rho(x) = \frac{2}{x^* B x} [Ax - \rho(x) Bx]$
 - ▶ SD direction: $\nabla \rho(x)$ parallels to the residual $r(x) = Ax - \rho(x) Bx$.
 - ▶ Line search (plain SD): $x_+ = x_c + t_* \cdot r(x_c)$, where $t_* = \operatorname{argmin}_t \rho(x_c + t \cdot r(x_c))$
8. The SD method (subspace projection version)
- 1: choose a vector x_0
 - 2: compute $x_0 = x_0 / \|x_0\|_B$, $\rho_0 = x_0^* A x_0$ and $r_0 = Ax_0 - \rho_0 Bx_0$.
 - 3: **for** $j = 0, 1, 2, \dots$ **do**
 - 4: if $\|r_j\|_2 / (\|Ax_j\|_2 + |\rho_j| \|Bx_j\|) \leq \text{tol}$, then break;
 - 5: set $Z = [x_j, r_j]$ (*search space*)
 - 6: compute the smaller eigenvalue μ and corresponding eigenvector v of $Z^* A Z - \lambda Z^* B Z$
 - 7: compute $x_{j+1} = \hat{x} / \|\hat{x}\|_B$, where $\hat{x} = Zv$
 - 8: set $\rho_{j+1} = \mu$ and compute the residue $r_{j+1} = Ax_{j+1} - \rho_{j+1} Bx_{j+1}$.
 - 9: **end for**
 - 10: return (ρ_j, x_j) as an approximate eigenpair to (λ_1, x_1)

9. Extensions:

- ▶ Locally optimal conjugate gradient method (LOCG) of Knyazev (1991)

$$Z = \text{span} \{x_{j-1}, x_j, (A - \rho_j B)x_j\}$$

- ▶ Extended SD method (inverse-free) of Golub and Ye (2002):

$$Z = \text{span} \left\{ x_j, (A - \rho_j B)x_j, \dots, (A - \rho_j B)^{m-1}x_j \right\}$$

for some integer $m > 2$

10. Practical issues

- ▶ Preconditioning, e.g.,

$$Z \rightsquigarrow \tilde{Z} = \text{span} \{x_{j-1}, x_j, K(A - \rho_j B)x_j\} \quad (\text{LOPCG})$$

- ▶ Blocking
- ▶ Deflation

11. MATLAB script demo_SD.m and demo_preconditionedSD.m

12. Further reading

- ▶ R.C. Li, *Rayleigh quotient based numerical methods for eigenvalue problems*, Lecture notes at Gene Golub SIAM Summer School 2013, available at <http://www.siam.org/students/g2s3/2013/course.html> **and references therein**

Part 1.3 Arnoldi method

1. Arnoldi process generates an orthonormal basis V_j of the Krylov subspace

$$\mathcal{K}_j(A, v_1) = \text{span} \{v_1, Av_1, \dots, A^{j-1}v_1\}$$

2. Arnoldi method for computing approximate eigenpairs of A

- 1: choose vector v_1 ($\|v_1\| = 1$)
- 2: **for** $j = 1, 2, \dots$ **do**
- 3: compute: $\hat{v} = Av_j$
- 4: orthogonalize: $\tilde{v} = \hat{v} - V_j h_j$, $h_j = V_j^* \hat{v}$
- 5: get new vector $v_{j+1} = \tilde{v} / h_{j+1,j}$, where $h_{j+1,j} = \|\tilde{v}\|$
- 6: compute the Ritz pairs (λ_i, x_i)
- 7: test for convergence
- 8: **end for**

3. Recurrence relation (Arnoldi decomposition)

$$AV_j = V_j H_j + h_{j+1,j} v_{j+1} e_j^* \equiv V_{j+1} \underline{H}_j$$

4. Ritz pairs (i.e., approximate eigenpairs of A) are given by $(\lambda_i, x_i = V_j y_i)$, where (λ_i, y_i) are eigenpairs of $H_j = V_j^* AV_j$.

5. Practical issues

- (a) Reorthogonalization to treat the loss of orthogonality in finite precision arithmetic
- (b) Restarting: explicit and implicit
- (c) Deflation (*aka locking*)
- (d) Shift-and-invert spectral transformation (known as *the shift-and-invert Arnoldi method*)

6. MATLAB script `demo_eigs.m`

7. Further reading: covered in the most textbooks.

Part 1.4 Rational Krylov method

1. The Rational Krylov (RK) method is a generalization of the shift-and-invert Arnoldi method
2. Starting with a vector v_1 , RK uses a Gram-Schmidt orthogonalization process to construct an orthonormal basis V_m for the subspace

$$\mathcal{Q}_m = \text{span}\{\hat{v}_1, \hat{v}_2, \dots, \hat{v}_m\} \quad \text{with} \quad v_{j+1} = (A - \sigma_j I)^{-1} v_j$$

when shifts $\sigma_j \in \mathbb{C}$.

3. The RK method

- 1: choose vector v_1 , $\|v_1\| = 1$
- 2: **for** $j = 1, 2, \dots$ **do**
- 3: choose shift σ_j
- 4: compute $w := \begin{cases} (A - \sigma_j I)^{-1} v_j, & \sigma_j \neq \infty \\ Av_j, & \sigma_j = \infty \end{cases}$
- 5: orthogonalize: $w := w - V_j h_j$, where $h_j = V_j^* w$
- 6: get new vector $v_{j+1} = w/h_{j+1,j}$, where $h_{j+1,j} = \|w\|$
- 7: compute the Ritz pairs: (λ_i, x_i)
- 8: test for convergence
- 9: **end for**

4. The RK satisfies the recurrence relation

$$AV_m \underline{K}_{m-1} = V_m \underline{L}_{m-1}$$

where the j th columns of $m \times (m - 1)$ upper Hessenberg matrices \underline{K}_{m-1} and \underline{L}_{m-1} are

$$\begin{aligned} \underline{k}_j &= \underline{h}_j, & \underline{l}_j &= \sigma_j \underline{h}_j + \underline{e}_j, & \text{for } \sigma_j &\neq \infty \\ \underline{k}_j &= \underline{e}_j, & \underline{l}_j &= \underline{h}_j, & \text{for } \sigma_j &= \infty \end{aligned}$$

with $\underline{h}_j = [h_j, h_{j+1,j}]^T$, the coefficients of the Gram-Schmidt orthogonalization process, and \underline{e}_j the j th column of the identity matrix I_m extended with a zero at the bottom.

5. Note that by these relations, we have

$$\sigma_j = \frac{l_{j+1,j}}{k_{j+1,j}}, \quad j = 1, \dots, m - 1,$$

where we assume that $h_{j+1,j} \neq 0$ (no breakdown).

6. Ritz pairs (approximate eigenpairs of A):

$$(\lambda_i, x_i = V_m \underline{K}_{m-1} s_i),$$

where

$$L_{m-1} s_i = \lambda_i K_{m-1} s_i.$$

with the residual

$$r_i = Ax_i - \lambda_i x_i = (l_{m,m-1} - \lambda_i k_{m,m-1})(e_{m-1}^T s_i) v_m$$

7. Practical issues

- ▶ implicit restarting
- ▶ deflation

8. MATLAB script `demo_rkm.m`

9. Further reading

- ▶ A. Ruhe, Rational Krylov sequence methods for eigenvalue computation, Lin. Alg. Appl. 58, pp.391-405, 1983.
- ▶ R. Van Beeumen, K. Meerbergen and W. Michiels, Connections between contour integration and rational Krylov methods for eigenvalue problems. TW673, Dept of Computer Science, KU Leuven, Nov. 2016 (and references therein)

Part 1.5 Topics of more recent interest

Part 1.5(a) Computing many eigenpairs of a Hermitian matrix

1. Let (λ_i, u_i) be the eigenpairs of a $n \times n$ Hermitian matrix A with the ordering $\lambda_1 \leq \lambda_2 \leq \dots$
 - ▶ Problem 1: Compute the first m -eigenpairs, where m is large, say $m = n/100$ when $n = O(10^6)$,
OR
 - ▶ Problem 2: Compute all eigenpairs in a given interval $[\alpha, \beta]$, where the interval contains many eigenvalues.
2. Approaches to be discussed
 - A1. Explicit deflation
 - A2. Spectrum slicing

A1. Explicit deflation = Wielandt's deflation

1. Let $AU = UA$ be the eigen-decomposition of A , partition

$$U = [U_k, U_{n-k}] \quad \text{and} \quad A = \begin{bmatrix} \Lambda_k & \\ & \Lambda_{n-k} \end{bmatrix},$$

where U_k consists of eigenvectors corresponding to the **first** k eigenvalues. Define the deflated matrix of the form

$$\tilde{A} = A + \zeta U_k U_k^*$$

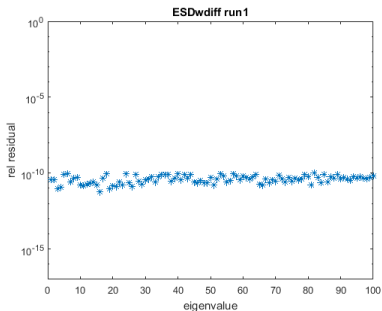
Then

- (a) \tilde{A} and A have the same eigenvectors
 - (b) The eigenvalues of \tilde{A} are $\begin{cases} \lambda_i + \zeta & \text{for } 1 \leq i \leq k \\ \lambda_i & \text{for } k+1 \leq i \leq n \end{cases}$
2. After the first k eigenpairs are computed by an eigensolver, one can pick a sufficient large ζ , and apply the eigensolver to compute the next k eigenpairs.
 3. Eigensolvers EIGIFP and BLEIGIFP (inverse-free preconditioned Krylov subspace method of Golub and Ye, 2002) have implemented such explicit deflation scheme. Can be easily incooperated into the most of existing eigensolvers, say, ARRABIT.

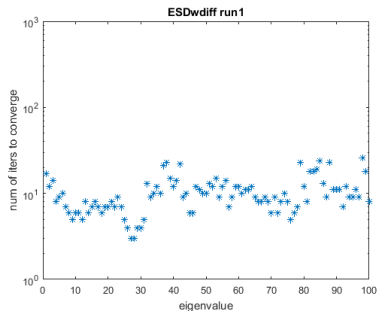
4. Example

- ▶ A Laplacian matrix of a diffusion map in nonparametric modeling of dynamical systems from J. Harlim.
- ▶ Matrix size $n = 10,000$, and compute 100 smallest eigenvalues **one-by-one via explicit deflation**

Residual norms



number of augmented SD iterations



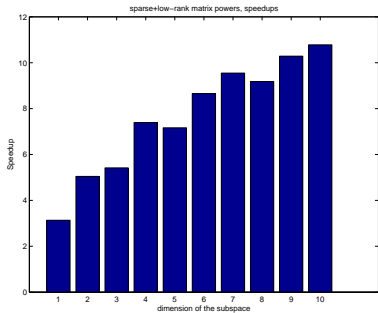
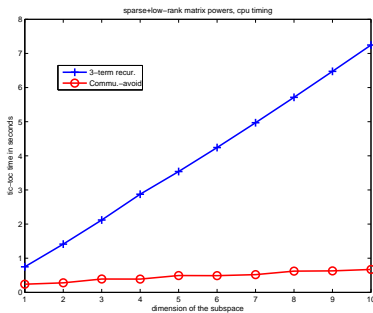
5. Two key open issues:

- numerical stability with the computed \hat{U}_k ,
- increasing the cost of the matrix-vector product $\tilde{A}q = (A + \zeta U_k U_k^T)q$

Communication-avoiding*Difference can even be seen on this laptop with MATLAB*

Example: Sparse plus low-rank matrix powers

$$y = p_j(A + \zeta U_k U_k^T) \cdot x$$



$$n = 250000, k = 500$$

A2. Spectrum slicing

1. **Task:** partition the interval $[\alpha, \beta]$ into subintervals, where each interval contains about the same number of eigenpairs
2. Subtask: computing the number, denoted as $c(\alpha, \beta)$, of eigenvalues of A within the given interval $[\alpha, \beta]$.
3. Approach 1: compute the inertia of $A - \alpha I$ and $A - \beta I$ via the LDLT factorizations of the matrices.
4. Approach 2: “density of states (dos)” [Lin, Saad and Yang’16, SIAM Rev.]
5. Approach 3: a preconditioned iterative method
 - ▶ $c(\alpha, \beta) = n_-(A - \beta I) - n_-(A - \alpha I)$, where $n_-(A - \tau I)$ is the negative inertia, which is the number of negative eigenvalues of $A - \tau I$.
 - ▶ inertia preservation:
 - ▶ $n_-(A - \tau I) = n_-(M(A - \tau I)M^*) \equiv n_-(C)$, where $T = MM^{ast}$ is a preconditioner of $A - \tau I$.
 - ▶ $n_-(C) = \text{trace}(h(C))$, where $h(x)$ is the step function

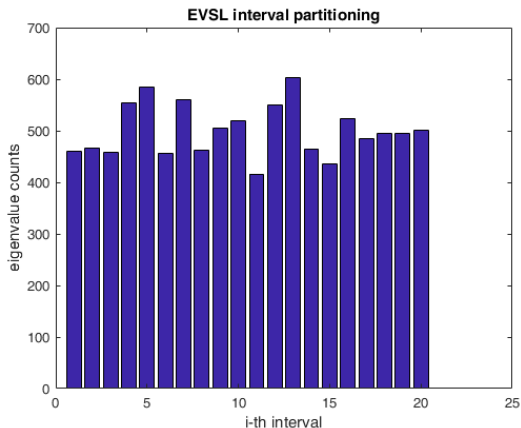
$$h(x) = \begin{cases} 1, & x < 0 \\ 0, & \text{otherwise} \end{cases}$$

- ▶ Monte-Carlo estimation of the trace of function

$$\text{trace}(h(C)) \approx \frac{1}{m} \sum_{j=1}^m v_j^* h(C) v_j \quad \text{for random vectors } v_j.$$

- ▶ Ref. E. Vecharynski and C. Yang, Preconditioned iterative methods for eigenvalue counts, arXiv:1602.02306v1, Feb. 2016.

6. Eigensolvers, such as FEAST, ZOLOEIGS and ChevLanTr in EVSL, need to have an interval of eigenvalues of interest.
7. EVSL (EigenValue Slicing Library)¹. provides a subroutine for the **Task**. In addition, EVSL also provides an implementation of the thick restarted Lanczos method with Chebyshev acceleration for computing many eigenpairs.
8. Example
 - ▶ The Laplacian matrix of a diffusion map
 - ▶ Number of eigenvalues in 20 subintervals (matrix size $n = 10000$)



¹available at <http://www-users.cs.umn.edu/~saad/software/EVSL>

Part 1.5(b) “Ill-conditioned” generalized symmetric-definite eigenvalue problems

1. Generalized symmetric definite eigenvalue problem

$$Ax = \lambda Bx \quad \text{with } A^T = A \text{ and } B^T = B > 0$$

2. LAPACK routines DSYGV[D,X] are based on the following Wilkinson's algorithm:

- 1: compute the Cholesky factorization $B = GG^T$
- 2: compute $C = G^{-1}AG^{-T}$
- 3: compute symmetric eigen-decomposition $Q^T C Q = A$
- 4: set $X = G^{-T}Q$

3. DSYGV[D,X] could be *numerically unstable* if B is ill-conditioned, since for a computed eigenpair $(\hat{\lambda}_i, \hat{x}_i)$:

$$|\hat{\lambda}_i - \lambda_i| \lesssim p(n)(\|B^{-1}\|_2 \|A\|_2 + \text{cond}(B)|\hat{\lambda}_i|) \cdot \epsilon$$

and

$$\theta(\hat{x}_i, x_i) \lesssim p(n) \frac{\|B^{-1}\|_2 \|A\|_2 (\text{cond}(B))^{1/2} + \text{cond}(B)|\hat{\lambda}_i|}{\text{specgap}_i} \cdot \epsilon$$

4. User's choice between the inversion of ill-conditioned Cholesky decomposition and the QZ algorithm that destroys symmetry

5. Existing work to address the ill-conditioning issue for dense matrices:

- ▶ Fix-Heiberger'72: explicit reduction (also see Sec.15.5 of Parlett's book)
- ▶ Chang-Chung Chang'74: SQZ method (QZ by Moler and Stewart'73)
- ▶ Bunse-Gerstner'84: MDR method
- ▶ Chandrasekaran'00: "proper pivoting scheme"
- ▶ Davies-Higham-Tisseur'01: Cholesky+Jacobi
- ▶ Working notes by Kahan'11 and Moler'14

6. Approaches to be discussed

- A1. A LAPACK-style implementation of Fix-Heiberger reduction method
- A2. An algebraic regularization
- A3. A locally accelerated preconditioned steepest descent method

A1. A LAPACK-style implementation of Fix-Heiberger algorithm

1. Given the threshold ε , a LAPACK-style computational routine DSYGVIC determines
 - ▶ $A - \lambda B$ is regular and has k ($0 \leq k \leq n$) ε -stable eigenvalues **OR**
 - ▶ $A - \lambda B$ is singular.
2. Implementation of DSYGVIC is based on Fix-Heiberger's algorithm, and organized in three phases.

3. DSYGVIC Phase I:

- (a) Compute the eigenvalue decomposition of B (DSYEV):

$$B^{(0)} = Q_1^T B Q_1 = \begin{bmatrix} D^{(0)} & \\ & E^{(0)} \end{bmatrix},$$

where diagonal entries of $n_1 \times n_1$ diagonal matrix D : $d_{11} \geq d_{22} \geq \dots \geq d_{n_1 n_1}$, and diagonal elements of $n_2 \times n_2$ diagonal matrix $E^{(0)}$ are smaller than $\varepsilon d_{11}^{(0)}$.

- (b) Set $E^{(0)} = 0$, and update A and $B^{(0)}$ with $R_1 = \text{diag}((D^{(0)})^{-1/2}, I)$:

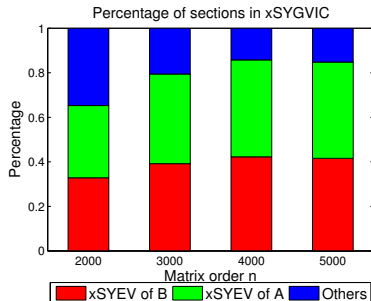
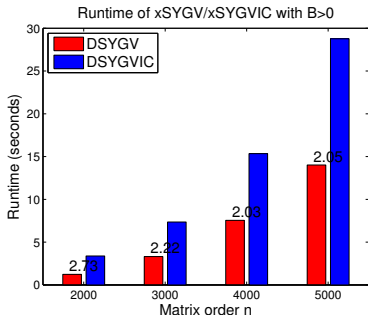
$$A^{(1)} = R_1^T Q_1^T A Q_1 R_1 = \begin{bmatrix} A_{11}^{(1)} & A_{12}^{(1)} \\ A_{12}^{(1)T} & A_{22}^{(1)} \end{bmatrix} \quad \text{and} \quad B^{(1)} = R_1^T B^{(0)} R_1 = \begin{bmatrix} I & \\ & 0 \end{bmatrix}$$

- (c) *Early exit.* If B is ε -well-conditioned (i.e., $n_2 = 0$), then $A - \lambda B$ is regular and has n ε -stable eigenpairs (λ, X) :

- ▶ $A^{(1)}U = UA$ (DSYEV).
- ▶ $X = Q_1 R_1 U$

4. DSYGVIC Phase I: timing profile

- ▶ Test matrices $A = Q_A D_A Q_A^T$ and $B = Q_B D_B Q_B^T$ where
 - ▶ Q_A, Q_B are random orthogonal matrices;
 - ▶ D_A is diagonal with $-1 < D_A(i, i) < 1, i = 1, \dots, n$;
 - ▶ D_B is diagonal with $0 < \varepsilon < D_B(i, i) < 1, i = 1, \dots, n$;
- ▶ 12-core on an Intel "Ivy Bridge" processor (Edison@NERSC)



- ▶ When B is ε -well-conditioned, DSYGVIC is about twice slower than Wilkinson's algorithm.

5. DSYGVIC Phase II

- (a) Compute the eigendecomposition of (2,2)-block $A_{22}^{(1)}$ of $A^{(1)}$ (DSYEV):

$$A_{22}^{(2)} = Q_{22}^{(2)T} A_{22}^{(1)} Q_{22}^{(2)} = \begin{bmatrix} D^{(2)} & \\ & E^{(2)} \end{bmatrix}$$

where eigenvalues are ordered such that $|d_{11}^{(2)}| \geq |d_{22}^{(2)}| \geq \dots$, and elements of $E^{(2)}$ are smaller than $\varepsilon|d_{11}^{(2)}|$.

- (b) Set $E^{(2)} = 0$, and update $A^{(1)}$ and $B^{(1)}$:

$$A^{(2)} = Q_2^T A^{(1)} Q_2 = \begin{bmatrix} A_{11}^{(2)} & A_{12}^{(2)} & A_{13}^{(2)} \\ A_{12}^{(2)T} & D^{(2)} & \\ A_{13}^{(2)T} & & 0 \end{bmatrix}$$
$$B^{(2)} = Q_2^T B^{(1)} Q_2 = \begin{bmatrix} I & & \\ & 0 & \\ & & 0 \end{bmatrix}$$

where $Q_2 = \text{diag}(I, Q_{22}^{(2)})$.

- (c) *Early exit.* When $A_{22}^{(1)}$ is a ε -well-conditioned matrix (i.e., $E^{(2)}$ is empty), $A - \lambda B$ is regular and has n_1 ε -stable eigenpairs (Λ, X) :

- ▶ $A^{(2)}U = B^{(2)}U\Lambda$ (use Schur complement and DSYEV)
- ▶ $X = Q_1 R_1 Q_2 U$.

6. DSYGVIC Phase II: accuracy test

- ▶ If $B \geq 0$ has n_2 zero eigenvalues:
 - ▶ DSYGV stops, the Cholesky factorization of B could not be completed.
 - ▶ DSYGVIC successfully computes $n - n_2$ ε -stable eigenpairs.
- ▶ If B has n_2 small eigenvalues about δ , both DSYGV and DSYGVIC “work”, but produce different quality numerically.

- ▶ $n = 1000, n_2 = 100, \delta = 10^{-13}$ and $\varepsilon = 10^{-12}$.

	Res1	Res2
DSYGV	3.5e-8	1.7e-11
DSYGVIC	9.5e-15	7.1e-12

- ▶ $n = 1000, n_2 = 100, \delta = 10^{-15}$ and $\varepsilon = 10^{-12}$.

	Res1	Res2
DSYGV	3.6e-6	1.8e-10
DSYGVIC	1.3e-16	6.8e-14

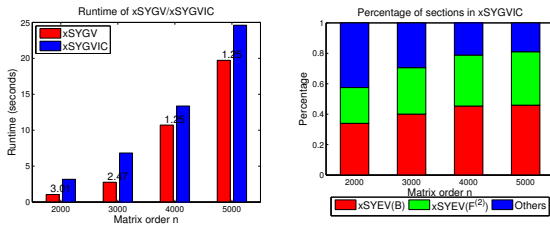
where

$$\text{Res1} = \frac{\|A\hat{X} - B\hat{X}\hat{\Lambda}\|_F}{n\|A\|_F \|\hat{X}\|_F} \quad \text{and} \quad \text{Res2} = \frac{\|\hat{X}^T B \hat{X} - I\|_F}{\|B\|_F \|\hat{X}\|_F^2}.$$

(as defined in LAPACK test routines)

7. DSYGVIC Phase II: timing profile

- ▶ Test matrices $A = Q_A D_A Q_A^T$ and $B = Q_B D_B Q_B^T$ where
 - ▶ Q_A, Q_B are random orthogonal matrices;
 - ▶ D_A is diagonal with $-1 < D_A(i, i) < 1, i = 1, \dots, n$;
 - ▶ D_B is diagonal with $0 < D_B(i, i) < 1, i = 1, \dots, n$ and $n_2/n D_B(i, i) < \varepsilon$.
- ▶ 12-core on an Intel "Ivy Bridge" processor (Edison@NERSC)



- ▶ Note that the performance of DSYGV varies depending on the percentage of "zero" eigenvalues of B . This is why the overhead ratio of DSYGVIC is lower.

8. DSYGVIC Phase III

(a) Recall $A^{(2)}$ and $B^{(2)}$ has the 3 by 3 block structure


$$A^{(2)} = \begin{bmatrix} A_{11}^{(2)} & A_{12}^{(2)} & A_{13}^{(2)} \\ A_{12}^{(2)T} & D^{(2)} & \\ A_{13}^{(2)T} & & 0 \end{bmatrix} \quad \text{and} \quad B^{(2)} = \begin{bmatrix} I & & \\ & 0 & \\ & & 0 \end{bmatrix}$$

(b) Reveal the rank of $A_{13}^{(2)}$ by QR decomposition with pivoting:

$$A_{13}^{(2)} \Pi = Q_{13}^{(3)} \cdot \begin{matrix} n_4 \\ n_1 - n_4 \end{matrix} \begin{bmatrix} R \\ 0 \end{bmatrix}$$

(c) *Final exit.* When $n_1 > n_4$ and R is full rank,² then $A - \lambda B$ is regular and has $n_1 - n_4$ ε -stable eigenpairs (Λ, X) :

- ▶ $A^{(3)}U = B^{(3)}U\Lambda$ (use Schur complement and DSYEV)
- ▶ $X = Q_1 R_1 Q_2 Q_3 U$.

²All the other cases either lead $A - \lambda B$ to be "singular" or "regular but no finite eigenvalues" 

9. DSYGVIC Phase III: accuracy test

- ▶ Consider 8×8 matrices (Fix-Heiberger'72):

$$A = Q^T H Q \quad \text{and} \quad B = Q^T S Q,$$

where Q is an random orthogonal matrix, and

$$H = \begin{bmatrix} 6 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 5 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

and

$$S = \text{diag}[1, 1, 1, 1, \delta, \delta, \delta, \delta]$$

- ▶ As $\delta \rightarrow 0$, $\lambda = 3, 4$ are the only stable eigenvalues of $A - \lambda B$.
- ▶ The computed eigenvalues when $\delta = 10^{-15}$:

λ_i	eig(A,B,'chol')	DSYGV	DSYGVIC($\epsilon = 10^{-12}$)
1	-3.334340289520080e+07	-0.3229260685047438e+08	0.3000000000000001e+01
2	-3.138309114827999e+07	-0.3107213627119420e+08	0.3999999999999999e+01
3	2.999999998949329e+00	0.2957918878610765e+01	
4	3.999999999513074e+00	0.4150528124449937e+01	
5	3.138309673669569e+07	0.3107214204558684e+08	
6	3.334340856015300e+07	0.3229261357421688e+08	
7	1.077763236890488e+15	0.1004773743630529e+16	
8	2.468473375420724e+15	0.2202090698823234e+16	

10. Further reading:

- ▶ http://cmjiang.cs.ucdavis.edu/fh_temp.html and references therein

A2. An algebraic regularization

1. A symmetric semi-definite pencil $A - \lambda B$ is defined by $A^T = A$ and $B^T = B \geq 0$.
2. Canonical form. There exists a nonsingular matrix $W \in \mathbb{R}^{n \times n}$ such that

$$W^T A W = \begin{matrix} & \begin{matrix} 2n_1 & r & n_2 & s \end{matrix} \\ \begin{matrix} 2n_1 \\ r \\ n_2 \\ s \end{matrix} & \begin{bmatrix} \Lambda_1 & & & \\ & \Lambda_2 & & \\ & & \Lambda_3 & \\ & & & 0 \end{bmatrix} \end{matrix}, \quad W^T B W = \begin{matrix} & \begin{matrix} 2n_1 & r & n_2 & s \end{matrix} \\ \begin{matrix} 2n_1 \\ r \\ n_2 \\ s \end{matrix} & \begin{bmatrix} \Omega_1 & & & \\ & I & & \\ & & 0 & \\ & & & 0 \end{bmatrix} \end{matrix}$$

where

$$\Lambda_1 = I_{n_1} \otimes K, \Lambda_2 = \text{diag}(\lambda_i), \Lambda_3 = \text{diag}(\pm 1), \Omega_1 = I_{n_1} \otimes T$$

and

$$K = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad T = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

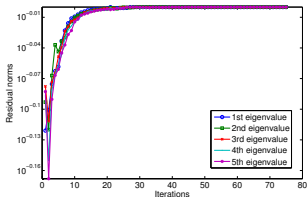
3. The canonical form reveals the structure of finite, infinite and indefinite ("0/0") eigenvalues, and singularity.
4. (a) The pencil $A - \lambda B$ is regular iff $s = 0$.
(b) A and B are *simultaneously diagonalizable* if $n_1 = 0$, which is equivalent to $\mathcal{AN}(B) \cap \mathcal{R}(B) = \{0\}$.

5. **Algebraic regularization:** Suppose that the pencil $A - \lambda B$ is regular and simultaneously diagonalizable. Let

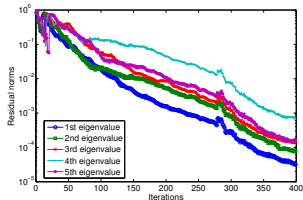
$$K = A + \mu(AZ)H(AZ)^T, \quad M = B + (AZ)H(AZ)^T,$$

where $Z \in \mathbb{R}^{n \times k}$ spans the nullspace of B , and $H \in \mathbb{R}^{k \times k}$ is an arbitrary symmetric positive definite matrix, and $\mu \in \mathbb{R}$. Then

- (1) $M > 0$
 - (2) $\lambda(K, M) = \lambda_f(A, B) \cup \lambda(\mu H + (Z^T A Z)^{-1}, H)$.³
6. By appropriately chosen H and μ , one can compute the k smallest (finite) eigenvalues of $A - \lambda B$ directly, say by LOBPCG.
7. Example: LOBPCG for a structure dynamics eigenvalue problem



LOBPCG(A, B)



LOBPCG(K, M)

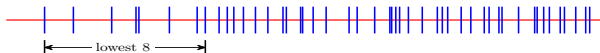
8. Reference: H. Xie, C.-P. Lin and Z. Bai, work in progress

³ $\lambda(A, B)$ denotes the set of eigenvalues of a pencil $A - \lambda B$. $\lambda_f(A, B)$ denotes the set of all finite eigenvalues of $A - \lambda B$.

A3. A locally accelerated preconditioned steepest descent method

1. Ill-conditioned GSEP $Hu_i = \lambda_i Su_i$:

- (a) Matrices H and S are ill-conditioned
e.g., $\text{cond}(H), \text{cond}(S) = \mathcal{O}(10^{10})$
- (b) Share a *near-nullspace* $\text{span}\{V\}$
e.g., $\|HV\| = \|SV\| = \mathcal{O}(10^{-4})$
- (c) No obvious spectrum gap between eigenvalues of interest and the rest
e.g.,



2. Assume the first $i - 1$ eigenpairs $(\lambda_1, u_1), \dots, (\lambda_{i-1}, u_{i-1})$ have been computed, and denote $U_{i-1} = [u_1, u_2, \dots, u_{i-1}]$.
3. PSD*id* (Preconditioned Steepest Descent with implicit deflation) computes the i -th eigenpair (λ_i, u_i)
 - 0 initialize $(\lambda_{i;0}, u_{i;0})$
 - 1 for $j = 0, 1, \dots$ until convergence
 - 2 compute $r_{i;j} = Hu_{i;j} - \lambda_{i;j}Su_{i;j}$
 - 3 precondition $p_{i;j} = -K_{i;j}r_{i;j}$
 - 4 $(\gamma_i, w_i) = \text{RR}(H, S, Z)$, where $Z = [U_{i-1} \ u_{i;j} \ p_{i;j}]$
 - 5 $\lambda_{i;j+1} = \gamma_i, u_{i;j+1} = Zw_i$
4. PSD*id* is an extension of [Faddeev and Faddeeva'63] and [Longsine and McCormick'80] for $K_{i;j} = I$.

5. With the assumptions:

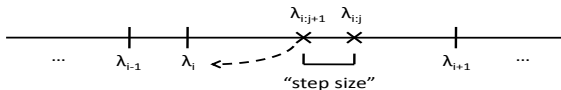
- (a) initialize $u_{i;0}$ such that $U_{i-1}^H S u_{i;0} = 0$ and $\|u_{i;0}\|_S = 1$
- (b) $\lambda_{i;0} = \rho(u_{i;0})$, Rayleigh quotient
- (c) the preconditioners $K_{i;j}$ are *effective positive definite*, namely,

$$K_{i;j}^d \equiv (U_{i-1}^c)^T S K_{i;j} S U_{i-1}^c > 0,$$

where $U_{i-1}^c = [u_i, u_{i+1}, \dots, u_n]$.

Then we can show the following properties of *PSDid*:

- (a) Z is of full column rank
- (b) $U_{i-1}^H S u_{i;j+1} = 0$ and $\|u_{i;j+1}\|_S = 1$
- (c) $\lambda_i \leq \lambda_{i;j+1} < \lambda_{i;j}$
- (d) $\lambda_{i;j} - \lambda_{i;j+1} \geq \sqrt{g^2 + \phi^2} - g = \text{"step size"} > 0,$



- (e) $p_{i;j} = -K_{i;j} r_{i;j}$ is an *ideal search direction* if $p_{i;j}$ satisfies

$$U^T S(u_{i;j} + p_{i;j}) = (\times, \dots, \times, \xi_i, 0, \dots, 0)^T \quad \text{and} \quad \xi_i \neq 0. \quad (1)$$

It implies that $\lambda_{i;j+1} = \lambda_i$.

6. PSD *id* convergence:

If $\lambda_i < \lambda_{i;0} < \lambda_{i+1}$ and $\sup_j \text{cond}(K_{i;j}^d) = q < \infty$, then the sequence $\{\lambda_{i;j}\}_j$ is strictly decreasing and bounded from below by λ_i , i.e.,

$$\lambda_{i;0} > \lambda_{i;1} > \cdots > \lambda_{i;j} > \lambda_{i;j+1} > \cdots \geq \lambda_i$$

and as $j \rightarrow \infty$,

- (a) $\lambda_{i;j} \rightarrow \lambda_i$
- (b) $u_{i;j}$ converges to u_i *directionally*:

$$\|r_{i;j}\|_{S^{-1}} = \|Hu_{i;j} - \lambda_{i;j}Su_{i;j}\|_{S^{-1}} \rightarrow 0$$

7. PSD *id* convergence rate: Let $\epsilon_{i;j} = \lambda_{i;j} - \lambda_i$, then

$$\epsilon_{i;j+1} \leq \left[\frac{\Delta + \tau \sqrt{\theta_{i;j} \epsilon_{i;j}}}{1 - \tau(\sqrt{\theta_{i;j} \epsilon_{i;j}} + \delta_{i;j} \epsilon_{i;j})} \right]^2 \epsilon_{i;j}$$

provided that the i -th approximate eigenvalue $\lambda_{i;j}$ is *localized*, i.e.

$$\tau(\sqrt{\theta_{i;j} \epsilon_{i;j}} + \delta_{i;j} \epsilon_{i;j}) < 1,$$

where

▶ $\Delta = \frac{\Gamma - \gamma}{\Gamma + \gamma}$ and $\tau = \frac{2}{\Gamma + \gamma}$

▶ $\delta_{i;j} = \|S^{\frac{1}{2}} K_{i;j} S^{\frac{1}{2}}\|$ and $\theta_{i;j} = \|S^{\frac{1}{2}} K_{i;j} M K_{i;j} S^{\frac{1}{2}}\|$

Γ and γ are largest and smallest pos. eigenvalues of $K_{i;j} M$ and $M = P_{i-1}^H (H - \lambda_i S) P_{i-1}$ and $P_{i-1} = I - U_{i-1} U_{i-1}^H S$

8. Remarks:

- (a) If $K_{i;j} = I$, the convergence of SD proven in [Faddeev and Faddeeva'63] and [Longside and McCormick'80]
- (b) If $i = 1$ and $K_{1;j} = K > 0$, it is Samokish's theorem (1958), which is first and still sharpest quantitative analysis [Ovtchinnikov'06].
- (c) Asymptotically,

$$\epsilon_{i;j+1} \leq \left[\Delta + \mathcal{O}(\epsilon_{i;j}^{1/2}) \right]^2 \epsilon_{i;j}$$

- (d) Optimal $K_{i;j}$: $\Delta = 0 \rightsquigarrow$ quadratic conv.
- (e) Semi-optimal $K_{i;j}$: $\Delta + \tau \sqrt{\theta_{i;j} \epsilon_{i;j}} \rightarrow 0 \rightsquigarrow$ superlinear conv.
- (f) (Semi-)optimality depends on the eigenvalue distribution of $K_{i;j} M$

9. Locally accelerated preconditioner: Consider the preconditioner

$$\widehat{K}_{i;j} = (H - \beta_{i;j}S)^{-1} \quad \text{with} \quad \beta_{i;j} = \lambda_{i;j} - c\|r_{i;j}\|_{S^{-1}}$$

where c is some constant. If

$$0 < \Delta_{i;j} < \min\left\{\frac{1}{4}\Delta_i^2, \frac{1}{10}\right\} \quad \text{and} \quad c > 3\sqrt{\Delta_{i;j}}.$$

where $\Delta_i = (\lambda_i - \lambda_{i-1})/(\lambda_{i+1} - \lambda_i)$ and $\Delta_{i;j} = (\lambda_{i;j} - \lambda_{i-1})/(\lambda_{i+1} - \lambda_{i;j})$.
Then

- (a) $K_{i;j}$ is effective positive definite
- (b) $\lambda_{i;j}$ is localized
- (c) $\Delta + \tau\sqrt{\theta_{i;j}\epsilon_{i;j}} \rightarrow 0$

Therefore, $\widehat{K}_{i;j}$ is asymptotically optimal

10. PSD $id \rightsquigarrow$ LABPSD = Locally Accelerated Block PSD

```

0  Initialize  $U_{m+\ell;0} = [u_{1;0} \ u_{2;0} \ \dots \ u_{m+\ell;0}]$ 
1   $(\Gamma, W) = \text{RR}(H, S, U_{m+\ell;0})$ 
2  update  $\Lambda_{m+\ell;0} = \Gamma$  and  $U_{m+\ell;0} = U_{m+\ell;0}W$ 
3  for  $j = 0, 1, \dots$ , do
4      compute  $R = HU_{m;j} - SU_{m;j}\Lambda_{m;j} \equiv [r_{1;j} \ r_{2;j} \ \dots \ r_{m;j}]$ 
5      if  $\text{Res}[\Lambda_{m;j}, U_{m;j}] = \max_{1 \leq i \leq m} \text{Res}[\lambda_{i;j}, u_{i;j}] \leq \tau_{\text{eig}}$ , break
6      for  $i = 1, 2, \dots, m$ 
          if  $\lambda_{i;j}$  is localized, then solve  $(H - \lambda_{i;j}S)p_{i;j} = -r_{i;j}$  for  $p_{i;j}$ 
7       $(\Gamma_{m+\ell}, W_{m+\ell}) = \text{RR}(H, S, Z)$ , where  $Z = [U_{m+\ell;j} \ p_{1;j} \ \dots \ p_{m;j}]$ 
8      update  $\Lambda_{m+\ell;j+1} = \Gamma_{m+\ell}$  and  $U_{m+\ell;j+1} = ZW_{m+\ell}$ 
9  end
10 return  $\{(\lambda_{i;j}, u_{i;j})\}_{i=1}^m$ 

```

11. Remarks:

- ▶ $\text{RR}(H, S, U)$ is the Rayleigh-Ritz procedure for the matrix pair (H, S) with the projection subspace $\text{span}(U)$.
- ▶ A "global" preconditioner $\approx (H - \sigma S)^{-1}$ can be used to accelerate the "localization" and convergence of step 6.

12. Example 1. Harmonic1D

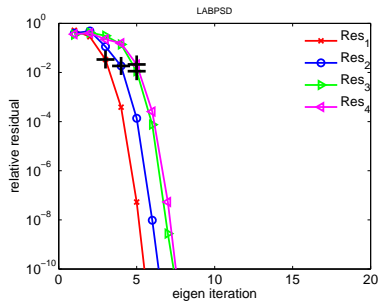
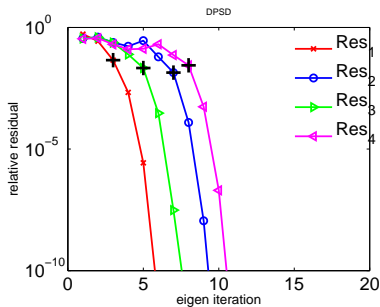
- ▶ PUFE discretization for harmonic oscillator in 1D
- ▶ $n = 112$ for 6-digit accuracy of 4 smallest eigenvalues $\lambda_1, \lambda_2, \lambda_3, \lambda_4$
- ▶ H and S are ill-conditioned

$$\text{cond}(H) = 8.79 \times 10^{10} \quad \text{and} \quad \text{cond}(S) = 2.00 \times 10^{12}$$

- ▶ H and S share a *near-nullspace* $\text{span}\{V\}$

$$\|HV\| = \|SV\| = O(10^{-5}) \quad \text{and} \quad \dim(V) = 17$$

- ▶ All computed $\hat{\lambda}_1, \hat{\lambda}_2, \hat{\lambda}_3, \hat{\lambda}_4$ have 6-digit accuracy.



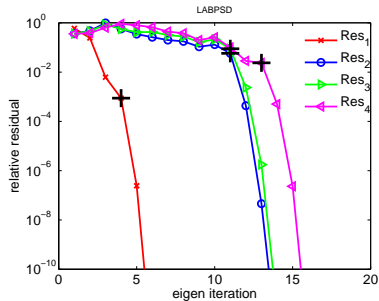
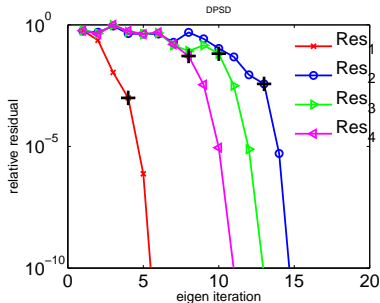
13. Example 2. CeAl-PUFE

- ▶ Metallic, triclinic CeAl, particularly challenging
- ▶ $n = 5336$ from PUFE discretization of the Kohn-Sham equation
- ▶ H and S are ill-conditioned

$$\text{cond}(H) = 1.16 \times 10^{10} \quad \text{and} \quad \text{cond}(S) = 2.57 \times 10^{11}$$

- ▶ H and S share a *near-nullspace* $\text{span}\{V\}$

$$\|HV\| = \|SV\| = O(10^{-4}) \quad \text{and} \quad \dim(V) = 1000$$



14. Further reading:

- ▶ Y. Cai, Z. Bai, J. Pask and N. Sukumar, Convergence analysis of a locally accelerated preconditioned steepest descent method for Hermitian-definite generalized eigenvalue problems, J. Comput. Math., to appear, 2017.
- ▶ Y. Cai, Z. Bai, J. Pask and N. Sukumar, Hybrid preconditioning for iterative diagonalization of ill-conditioned generalized eigenvalue problems in electronic structure calculations, J. of Comput. Phys., 255, pp.16-33, 2013

Recap of Part 1

1. Accelerated subspace iteration

MATLAB script `demo_RayleighRitz.m`

2. Steepest descent method

MATLAB scripts `demo_SD.m`, `demo_PreconditionedSD.m`

3. Arnoldi method

MATLAB script `demo_eigs.m`

4. Rational Krylov method

MATLAB script `demo_rkm.m`

5. Topics of more recent interest

(a) Computing many eigenpairs of a Hermitian matrix

(b) Solving “ill-conditioned” generalized symmetric definite eigenvalue problems

6. Exercises

Part 1(a) – 4 problems

Part 1(b) – 4 problems

Part 2: Nonlinear eigenvalue problems

Outline of Part 2

1. Essential theory
2. Methods based on Newton iterations
 - (a) Newton's method based on scalar functions
 - (b) Newton's method based on vector equations
 - (c) Common issues
3. Methods designed for QEP and REP
 - (a) QEP
 - (b) REP
 - (c) QEP with low-rank damping
4. Methods based on approximation/interpolation and linearization
 - (a) The method of successive linear approximation
 - (b) Linearization of a matrix polynomial
 - (c) PEP in monomial basis and linearization
 - (d) Lagrange interpolation in barycentric form and linearization
 - (e) Newton interpolation and linearization
 - (f) (Rational) Padé approximation and linearization
 - (g) Rational interpolation and linearization
 - (h) Algorithmic framework and software
5. Of things not treated

Part 2.1 Essential theory

1. Nonlinear eigenvalue problem

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

where

- ▶ $T : \Omega \rightarrow \mathbb{C}^{n \times n}$ is a matrix-valued function and $\Omega \subseteq \mathbb{C}$ is a nonempty open set.
- ▶ $\lambda \in \Omega$ is an *eigenvalue*.
- ▶ $x \in \mathbb{C}^n \setminus \{0\}$ is an *eigenvector*.

The set of all eigenvalues is denoted by $\Lambda(T)$ and referred to as the *spectrum* of T , while $\Omega \setminus \Lambda(T)$ is called the *resolvent set* of T .

2. The eigenvalues of $T(\lambda)$ are the roots of the scalar function $f(z) = \det(T(z))$.
3. The *algebraic multiplicity* of an eigenvalue λ is defined as the multiplicity of the root of $f(z)$ at λ . An eigenvalue is called simple if its algebraic multiplicity is one.
4. The *geometric multiplicity* of λ is the dimension of the null space of $T(\lambda)$. An eigenvalue is called semi-simple if its algebraic and geometric multiplicities coincide.

5. Example 1. T is 1×1 scalar function $T : \Omega \rightarrow \mathbb{C}$:

- ▶ no solution at all, e.g., $T(z) = e^z$.
- ▶ finitely many solutions, e.g., $T(z) = z^4 - 1$.
- ▶ infinitely many solutions, e.g., $T(z) = \cos(z)$.

6. Example 2. T is a 2×2 matrix

$$T(z) = \begin{bmatrix} e^{iz^2} & 1 \\ 1 & 1 \end{bmatrix}.$$

It is singular at the points $z \in \mathbb{C}$ where $e^{iz^2} = 1$. Hence the eigenvalues are $\lambda_k = \pm\sqrt{2\pi k}$ for $k = 0, \pm 1, \pm 2, \dots$

7. For LEPs, eigenvectors corresponding to distinct eigenvalues are linearly independent, which is not the case for NEPs. For examples,

- ▶ there is only one eigenvector $x = [1, -1]^T$ for the the previous 2×2 matrix $T(z)$.
- ▶ the quadratic eigenvalue problem

$$T(\lambda)x = \left(\begin{bmatrix} 0 & 1 \\ -2 & 3 \end{bmatrix} + \lambda \begin{bmatrix} 7 & -5 \\ 10 & -8 \end{bmatrix} + \lambda^2 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right) x = 0.$$

The distinct eigenvalues $\lambda = 1$ and $\lambda = 2$ share the eigenvector $x = [1, 2]^T$.

8. It is common to assume that $T : \Omega \rightarrow \mathbb{C}^{n \times n}$ is holomorphic⁴ in the domain $\Omega \subseteq \mathbb{C}$, denoted by $T \in H(\Omega, \mathbb{C}^{n \times n})$.
9. $T \in H(\Omega, \mathbb{C}^{n \times n})$ implies that $\det(T(z)) \in H(\Omega, \mathbb{C})$.
10. T is *regular* if $\det(T(z)) \not\equiv 0$. The condition that T is regular is equivalent to that the resolvent set $\Omega \setminus \Lambda(T)$ is nonempty.
11. If T is regular, every eigenvalue $\lambda \in \Lambda(T)$ is *isolated*, i.e., there exists an open neighborhood $\mathcal{U} \subset \Omega$ such that $\mathcal{U} \cap \Lambda(T) = \{\lambda\}$.
12. A matrix-valued function $T(z)$ is said to be *Hermitian* if $T(z)^* = T(\bar{z})$ for all $z \in \mathbb{C}$.
 - ▶ The eigenvalues of a Hermitian F are either real or they come in pairs $(\lambda, \bar{\lambda})$
 - ▶ $T(\lambda)v = 0$ and $w^*T(\lambda) = 0$ implies that $T(\bar{\lambda})w = 0$ and $v^*T(\bar{\lambda}) = 0$.
 - ▶ If λ is real, the left and right eigenvectors corresponding to λ coincide.
13. A matrix-valued function $T(z)$ is called *genuinely nonlinear* if the problem of finding its eigenvalues is not linearizable. The eigenvalue problem for $T(z)$ is then called *non-linearizable*.

⁴i.e., differentiable in a neighborhood of every point in its domain.

14. References:

- (a) H. Voss. *Nonlinear eigenvalue problems*. Chapter 60 of Handbook of Linear Algebra (2nd edition), L. Hogben editor. Chapman and Hall/CRC, 2013. (A collection of mathematical facts about NEPs)
- (b) S. Güttel and F. Tisseur, *The nonlinear eigenvalue problem*, Acta Numerica, Vol.26, 2017. (Section 2 contains the solution structure of NEPs)
- (c) (i) D. Bindel and A. Hood. *Localization theorems for nonlinear eigenvalue problems*. SIAM Journal on Matrix Analysis and Applications, 34(4):17281749, 2013, (ii) A. Hood. *Localizing the eigenvalues of matrix-valued functions: analysis and Applications*, PhD thesis, Cornell University, Jan. 2017
- (d) T. Betcke, N. J. Higham, V. Mehrmann, C. Schröder, and F. Tisseur. *NLEVP: A collection of nonlinear eigenvalue problems*. ACM Trans. Math. Softw., 39(2):7:17:28, 2013. <http://www.mims.manchester.ac.uk/research/numerical-analysis/nlevp.html>

Part 2.2 Newton's methods

- (a) Newton's method based on scalar functions
- (b) Newton's method based on vector functions
- (c) Common issues

Part 2.2(a) Newton's method based on scalar functions

1. Eigenvalues as roots of determinant:

$$F(\lambda)v = 0 \quad \iff \quad f(\lambda) := \det(F(\lambda)) = 0$$

2. Apply Newton's method for a root of $f(\lambda)$ with initial $\lambda^{(0)}$:

$$\lambda^{(k+1)} = \lambda^{(k)} - \frac{f(\lambda^{(k)})}{f'(\lambda^{(k)})}, \quad k = 0, 1, 2, \dots$$

3. By Jacobi's derivative formula,

$$f'(\lambda) = \det(F(\lambda)) \cdot \operatorname{tr}(F^{-1}(\lambda) F'(\lambda)),$$

Newton's iteration becomes

$$\lambda^{(k+1)} = \lambda^{(k)} - \frac{1}{\operatorname{tr}(F^{-1}(\lambda^{(k)}) F'(\lambda^{(k)}))}.$$

4. To evaluate the trace, we apply the LU factorization of $F(\lambda^{(k)})$ to solve $2n$ triangular systems for all diagonal elements.

5. Remarks

- ▶ Initial value: $\lambda^{(0)}$ is required (the choice is crucial to the convergence).
- ▶ Convergence rate: locally quadratic to a simple root.
- ▶ More than one eigvals: deflate the computed eigvals λ_ℓ , for $\ell = 1:k$, by replacing $f(\lambda)$ with

$$\tilde{f}(\lambda) = \frac{f(\lambda)}{\prod_{\ell=1}^k (\lambda - \lambda_\ell)}.$$

- ▶ Approximate eigvecs v for λ : return the (approximate) right singular vector of $F(\lambda)$ as the corresponding eigenvector v .

6. MATLAB script `Newton_Trace.m`

7. Reference:

- ▶ P. Lancaster, *Lambda-Matrices and Vibrating Systems*, Pergamon Press Inc. 1966

8. Extension I: Implicit determinant

- ▶ Key observation: to avoid working with nearly singular matrices $F(\lambda^{(k)})$, consider the boarded linear system with constant vectors $b, c \in \mathbb{C}^n$:

$$\underbrace{\begin{bmatrix} F(\lambda) & b \\ c^T & 0 \end{bmatrix}}_{G(\lambda)} \begin{bmatrix} x \\ f \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix},$$

the solution is given by (using Cramer's rule and assuming G nonsingular)

$$f(\lambda) = \frac{\det F(\lambda)}{\det G(\lambda)}.$$

- ▶ Derivative of f : by differentiating the linear system w.r.t. λ , we obtain

$$\begin{bmatrix} F(\lambda) & b \\ c^T & 0 \end{bmatrix} \begin{bmatrix} x'(\lambda) \\ f'(\lambda) \end{bmatrix} = \begin{bmatrix} -F'(\lambda)x(\lambda) \\ 0 \end{bmatrix}.$$

- ▶ Newton's method for $f = 0$: both $f(\lambda)$ and $f'(\lambda)$ can be evaluated by solving the above two linear systems with a common coefficient matrix $G(\lambda)$.
- ▶ MATLAB script `Newton_implicit_determinant.m`
- ▶ References:
 - ▶ A. L. Andrew, E. K. Chu, and P. Lancaster. "On the numerical solution of nonlinear eigenvalue problems." *Computing* 55(2), pp.91-111, 1995.
 - ▶ A. Spence and C. Poulton. "Photonic band structure calculations using nonlinear eigenvalue techniques." *Journal of Computational Physics* 204(1), pp.65-81, 2005.

9. Extension II: Newton-QR iteration

- ▶ QR decomposition of $F(\lambda)$ with column pivoting

$$F(\lambda)\Pi(\lambda) = Q(\lambda)R(\lambda) \equiv Q(\lambda) \begin{bmatrix} R_{11}(\lambda) & r_{12}(\lambda) \\ 0 & r_{nn}(\lambda) \end{bmatrix}.$$

where Π is a permutation matrix, Q is an orthogonal matrix and R is upper triangular with decreasing diagonal $r_{11} \geq r_{22} \geq \dots \geq r_{nn}$.

- ▶ Scalar equation:

$$\det F(\lambda) = 0 \quad \iff \quad f(\lambda) := r_{nn}(\lambda) = 0$$

- ▶ Derivative: assume $\Pi(\lambda)$ is constant in a small neighborhood of λ , then

$$r'_{nn}(\lambda) = e_n^T Q(\lambda)^* F'(\lambda) \Pi(\lambda) \begin{bmatrix} -R_{11}^{-1}(\lambda)r_{12}(\lambda) \\ 1 \end{bmatrix}.$$

This leads to the Newton-QR iteration for a root of $r_{nn}(\lambda)$.

- ▶ MATLAB script `Newton_QR.m`

- ▶ References:

- ▶ V. N. Kublanovskaya, "On an approach to the solution of the generalized latent value problem for λ -matrices", *SIAM J. Numer. Anal.* 7, pp.532–537, 1970.
- ▶ C. K. Garrett, Z. Bai, and R.-C. Li, "A nonlinear QR algorithm for banded nonlinear eigenvalue problems." *ACM Transactions on Mathematical Software* 43.1, Article 4, 19 pages, 2016, and references therein.

Part 2.2(b) Newton's method based on vector equations

1. Eigenpairs as the solution of vector equations:

$$\begin{cases} F(\lambda)v = 0 \\ u^*v = 1 \end{cases} \iff \mathcal{N} \begin{bmatrix} v \\ \lambda \end{bmatrix} := \begin{bmatrix} F(\lambda)v \\ u^*v - 1 \end{bmatrix} = 0$$

- ▶ $u \in \mathbb{C}^n$: a constant normalization vector.
- ▶ $\mathcal{N} : \mathbb{C}^{n+1} \rightarrow \mathbb{C}^{n+1}$: a nonlinear functional.

2. Newton's method for $\mathcal{N} = 0$:

$$\begin{bmatrix} v^{(k+1)} \\ \lambda^{(k+1)} \end{bmatrix} = \begin{bmatrix} v^{(k)} \\ \lambda^{(k)} \end{bmatrix} - \left(J_{\mathcal{N}} \begin{bmatrix} v^{(k)} \\ \lambda^{(k)} \end{bmatrix} \right)^{-1} \mathcal{N} \begin{bmatrix} v^{(k)} \\ \lambda^{(k)} \end{bmatrix},$$

where the Jacobian matrix is given by

$$J_{\mathcal{N}} \begin{bmatrix} v \\ \lambda \end{bmatrix} = \begin{bmatrix} F(\lambda) & F'(\lambda)v \\ u^* & 0 \end{bmatrix}.$$

3. Implementation: Nonlinear inverse iteration

- ▶ Newton's iteration: given $\lambda^{(k)}$ and $v^{(k)}$ with $u^* v^{(k)} = 1$,

$$\begin{bmatrix} v^{(k+1)} \\ \lambda^{(k+1)} \end{bmatrix} = \begin{bmatrix} v^{(k)} \\ \lambda^{(k)} \end{bmatrix} - \left(\begin{bmatrix} F(\lambda^{(k)}) & F'(\lambda^{(k)})v^{(k)} \\ u^* & 0 \end{bmatrix} \right)^{-1} \begin{bmatrix} F(\lambda^{(k)})v^{(k)} \\ 0 \end{bmatrix}$$

- ▶ Blockwise form:

$$\begin{cases} F(\lambda^{(k)})v^{(k+1)} = - \underbrace{(\lambda^{(k+1)} - \lambda^{(k)})}_{\text{scalar}} \cdot F'(\lambda^{(k)})v^{(k)}, \\ u^* v^{(k+1)} = 1. \end{cases}$$

- ▶ Nonlinear inverse iteration:

1. Solve $F(\lambda^{(k)})x = F'(\lambda^{(k)})v^{(k)}$ for x .
2. Normalize $v^{(k+1)} = x/\alpha$ with $\alpha = u^* x$.
3. Update $\lambda^{(k+1)} = \lambda^{(k)} - \alpha^{-1}$.

- ▶ Remark: The normalizing u can either be a constant vector (e.g. $u = e_i$, or orthogonal to previously computed eigvecs), or updated in iterations (e.g., $u = F(\lambda^{(k)})w^{(k)}$ and $w^{(k)}$ is an approximate left eigvec).

4. MATLAB script `inverse_iteration.m`

5. Ref.: A. Ruhe, Algorithms for the nonlinear eigenvalue problem. SIAM Numer. Anal. 10, pp.674-689, 1973.

6. Extension I: Two-sided Rayleigh functional iteration

- ▶ Eigenvectors update: Inverse iteration updates the (right) eigenvector (up to normalization) by

$$v^{(k+1)} = F(\lambda^{(k)})^{-1} F'(\lambda^{(k)}) v^{(k)}.$$

We can do the same for $F(\lambda)^*$ to update the left eigenvector

$$w^{(k+1)} = F(\lambda^{(k)})^{-*} F'(\lambda^{(k)})^* w^{(k)}.$$

Remark: For Hermitian NEPs, we have $w^{(k+1)} = v^{(k+1)}$.

- ▶ Eigenvalue update: with both approximate left and right eigenvectors, update the eigenvalue by the Rayleigh functional

$$\lambda^{(k+1)} \longleftarrow \text{the root } \rho \text{ of } w^{(k+1)*} F(\rho) v^{(k+1)} = 0 \text{ closest to } \lambda^{(k)}.$$

- ▶ Two-sided Rayleigh functional iteration: iterate the eigentriple

$$(\lambda^{(k+1)}, v^{(k+1)}, w^{(k+1)}) \longleftarrow (\lambda^{(k)}, v^{(k)}, w^{(k)})$$

- ▶ It is locally cubically convergent for a simple eigentriple (λ, v, w) .
- ▶ Reference: K. Schreiber, *Nonlinear Eigenvalue Problems: Newton-type Methods and Nonlinear Rayleigh Functionals*, Ph.D. thesis, TU Berlin, 2008.

7. Extension II: Residual inverse iteration

- ▶ Quasi-Newton's method: (given $\lambda^{(k)}$ and $v^{(k)}$ with $u^* v^{(k)} = 1$)

$$\begin{bmatrix} v^{(k+1)} \\ \lambda^{(k+1)} \end{bmatrix} = \begin{bmatrix} v^{(k)} \\ \lambda^{(k)} \end{bmatrix} - \tilde{J}_k^{-1} \begin{bmatrix} F(\lambda^{(k)})v^{(k)} \\ 0 \end{bmatrix},$$

where \tilde{J}_k is an *approximate Jacobian* for a prescribed 'shift' σ :

$$\tilde{J}_k := \begin{bmatrix} F(\sigma) & \frac{F(\lambda^{(k+1)}) - F(\lambda^{(k)})}{\lambda^{(k+1)} - \lambda^{(k)}} v^{(k)} \\ u^* & 0 \end{bmatrix} \approx \begin{bmatrix} F(\lambda^{(k)}) & F'(\lambda^{(k)})v^{(k)} \\ u^* & 0 \end{bmatrix}.$$

- ▶ Blockwise form:

$$\begin{cases} v^{(k+1)} = v^{(k)} + F(\sigma)^{-1} F(\lambda^{(k+1)})v^{(k)}, \\ 0 = u^* F(\sigma)^{-1} F(\lambda^{(k+1)})v^{(k)}. \end{cases}$$

Solve the new approximant $\lambda^{(k+1)}$ from the second equation, then update the eigenvector $v^{(k+1)}$.

7. Extension II: Residual inverse iteration, cont'd

- ▶ Residual inverse iteration: given $\lambda^{(k)}$ and $v^{(k)}$
 1. Update $\lambda^{(k+1)}$: the root ρ of $u^* F(\sigma)^{-1} F(\rho) v^{(k)} = 0$ closest to $\lambda^{(k)}$.
 2. Residual inverse step: solve $F(\sigma)x = F(\lambda^{(k+1)})v^{(k)}$ for x .
 3. Normalization: $v^{(k+1)} = \tilde{v}^{(k+1)} / u^* \tilde{v}^{(k+1)}$ with $\tilde{v}^{(k+1)} = v^{(k)} + x$.
- ▶ Remarks:
 - ▶ Precompute the LU fact. of $F(\sigma)$, and reuse it in each residual inverse step.
 - ▶ Update the shift σ when convergence is slow.
 - ▶ The algorithm is *locally linearly* convergent due to the fixed shift σ .
 - ▶ For Hermitian NEPs, one can solve $v^{(k)*} F(\rho) v^{(k)} = 0$ for $\lambda^{(k+1)}$. The algorithm is locally quadratically convergent.
- ▶ References
 - ▶ A. Neumaier, *Residual inverse iteration for the nonlinear eigenvalue problem*. SIAM Numer. Anal. 22(1985): 914-923.
 - ▶ E. Jarlebring, A. Koskela and G. Mele, *Disguised and new quasi-Newton methods for nonlinear eigenvalue problems*, arXiv:1702.08492v1, Feb.27, 2017

Part 2.2(c) Common issues

1. Large sparse NEPs:

- ▶ **Inexact** Newton-type methods:

applying iterative solvers (e.g., GMRES) for the linear systems of each 'inverse' step. For example, in the nonlinear inverse iteration, compute $\tilde{v}^{(k+1)}$ with residual error satisfying

$$\|F'(\lambda^{(k)})v^{(k)} - F(\lambda^{(k)})\tilde{v}^{(k+1)}\| \leq \tau^{(k)} \|F'(\lambda^{(k)})v^{(k)}\|,$$

where $\tau^{(k)} > 0$ is an user provided tolerance.

This is an issue of "*inner-outer iteration*".

- ▶ Reference: D. B. Szyld and F. Xue, Local convergence analysis of several inexact Newton-type algorithms for general nonlinear eigenvalue problems, Numer. Math. 123, pp.333-362, 2013.

2. Deflation of computed eigenvalues and eigenpairs

- ▶ Goal: solve for **several** eigenvalues and/or eigenpairs.
- ▶ Nonequivalence transformation:

$$\tilde{F}(\lambda) = F(\lambda) \prod_{i=1}^{\ell} \left(I - \frac{\lambda - \lambda_i - 1}{\lambda - \lambda_i} y_i x_i^* \right)$$

- ▶ $\lambda_1, \dots, \lambda_\ell$: the computed ℓ simple eigenvalues of F .
 - ▶ $x_i, y_i \in \mathbb{C}^n$: vectors such that $y_i^* x_i = 1$ for $i = 1, \dots, \ell$.
- ▶ Deflation properties:
- ▶ $\tilde{F}(\lambda)$ and $F(\lambda)$ share the same eigenvalues except $\lambda_1, \dots, \lambda_\ell$, since

$$\det \tilde{F}(\lambda) = \det F(\lambda) \prod_{i=1}^{\ell} \frac{1}{\lambda - \lambda_i}.$$

- ▶ Let (λ, \tilde{v}) be an eigenpair of \tilde{F} , then $\lambda \neq \lambda_i$, for $i = 1 : \ell$, is an eigenvalue of F with eigenvector

$$v = \prod_{i=1}^{\ell} \left(I - \frac{\tilde{\lambda} - \lambda_i - 1}{\tilde{\lambda} - \lambda_i} y_i x_i^* \right) \tilde{v}.$$

Part 2.3 Methods specially designed for QEP and REP

- (a) QEP
- (b) REP
- (c) QEP with low-rank damping

Part 2.3(a) QEP

1. The Quadratic Eigenvalue Problem (QEP): Given M , D and $K \in \mathbb{C}^{n \times n}$, find $\lambda \in \mathbb{C}$ and nonzero $x \in \mathbb{C}^n$ satisfying

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

2. The QEP is the most studied (and important) NEP. Mathematical and algorithmic ideas developed for the QEP can often be extended to other linearizable and genuinely nonlinear eigenvalue problems.
3. The recent work on solving dense QEPs can be found in
 - ▶ S. Hammarling, C. J. Munro and F. Tisseur, An algorithm for the complete solution of quadratic eigenvalue problems, ACM Trans. Math. Software, 39(3), pp.913-938, 2013
 - ▶ L. Zeng and Y. Su, A backward stable algorithm for quadratic eigenvalue problems, SIAM J. Matrix Anal. Appl., 35(2), pp.499-516, 2014
 - ▶ MALTLAB script `demo_quadeig.m`

We will be focusing on solving large-scale sparse QEPs.

4. To compute the eigenvalues of the QEP close to a point σ of interest, a standard approach is to first apply the shift spectral transformation $\mu = \lambda - \sigma$ and then solve the QEP

$$(\mu^2 M + \mu D_\sigma + K_\sigma)x = 0,$$

where $D_\sigma = C + 2\sigma M$ and $K_\sigma = \sigma^2 M + \sigma C + K$.

5. By a linearization technique, say in the first companion form, the QEP is equivalent to the linear eigenvalue problem (LEP):

$$\begin{bmatrix} -D_\sigma & -K_\sigma \\ I & 0 \end{bmatrix} \begin{bmatrix} \mu x \\ x \end{bmatrix} = \mu \begin{bmatrix} M & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \mu x \\ x \end{bmatrix},$$

or rewritten as

$$\begin{bmatrix} A & B \\ I & 0 \end{bmatrix} \begin{bmatrix} \mu x \\ x \end{bmatrix} = \mu \begin{bmatrix} \mu x \\ x \end{bmatrix}$$

where

$$A = -M^{-1}D_\sigma \quad \text{and} \quad B = -M^{-1}K_\sigma.$$

6. The task of finding eigenvalues λ of the QEP close to the shift σ becomes one of extracting smallest (in modulus) few eigenvalues μ of the LEP. However, the dimension of the LEP is twice the dimension of the QEP. If without carefully exploiting the structure of the LEP, memory and computational costs are increased substantially. We next discuss memory-efficient QEP algorithms. As a product, we also gain a significant improvement of accuracy.

7. Given $A, B \in \mathbb{R}^{n \times n}$ and $r_{-1}, r_0 \in \mathbb{R}^n$, the second-order Krylov subspace

$$\mathcal{G}_k(A, B; r_{-1}, r_0) := \text{span}\{r_{-1}, r_0, r_1, r_2, \dots, r_{k-1}\} = \text{span}\{Q_k\}$$

where $r_i = Ar_{i-2} + Br_{i-1}$ for $i = 1, 2, \dots$

8. Subspace embedding

$$\begin{aligned} \text{span} \left\{ \begin{bmatrix} r_0 \\ r_{-1} \end{bmatrix}, \begin{bmatrix} r_1 \\ r_0 \end{bmatrix}, \begin{bmatrix} r_2 \\ r_1 \end{bmatrix}, \dots, \begin{bmatrix} r_{k-1} \\ r_{k-2} \end{bmatrix} \right\} &= \mathcal{K}_k(L, v_0) \\ &= \text{span}\{V_k\} = \text{span} \left\{ \begin{bmatrix} V_{1,k} \\ V_{2,k} \end{bmatrix} \right\}, \end{aligned}$$

where

$$\mathcal{K}_k(L, v_0) = \text{span}\{v_0, Lv_0, \dots, L^{k-1}v_0\}, \quad L = \begin{bmatrix} A & B \\ I & 0 \end{bmatrix}, \quad v_0 = \begin{bmatrix} r_0 \\ r_{-1} \end{bmatrix}$$

9. A simple, yet critical observation

$$\text{span}\{Q_k\} = \text{span}\{V_{1,k}, V_{2,k}\}.$$

10. SOAR (Second-Order ARnoldi) is an Arnoldi-type procedure to generate an orthogonal basis matrix Q_k of $\mathcal{G}_k(A, B; r_{-1}, r_0)$. Unfortunately, SOAR procedure can be numerically unstable due to

- ▶ triangular inversion in SOAR could be ill-conditioned
- ▶ SOAR implicitly generates a non-orthogonal basis $V_k = \begin{bmatrix} Q_k \\ Q_k U_{k,2} \end{bmatrix}$ of $\mathcal{K}_k(L, v_0)$.

11. Two-level Orthogonality ARnoldi (TOAR) procedure.

- ▶ Recall the basis matrix connection

$$\text{span}\{Q_k\} = \text{span}\{V_{1,k}, V_{2,k}\}.$$

- ▶ Represent an orthonormal basis V_k of $\mathcal{K}_k(L, v_0)$ by a two-level orthogonality:

$$V_k = \begin{bmatrix} V_{1,k} \\ V_{2,k} \end{bmatrix} = \begin{bmatrix} Q_k U_{k,1} \\ Q_k U_{k,2} \end{bmatrix} = \begin{bmatrix} Q_k & \\ & Q_k \end{bmatrix} \underbrace{\begin{bmatrix} U_{k,1} \\ U_{k,2} \end{bmatrix}}_{U_k},$$

where $Q_k \in \mathbb{R}^{n \times k}$ and $U_k \in \mathbb{R}^{2k \times k}$ are orthonormal.

- ▶ Compact storage: $nk + 2k^2$ (versus $2nk$ for storing V_k)

12. Arnoldi process

- 1: $v_1 = v_0 / \|v_0\|_2$
- 2: **for** $j = 1 : k - 1$ **do**
- 3: $w = Lv_j$
- 4: $h_j = V_j^T w$
- 5: $w := w - V_j h_j$
- 6: $h_{j+1,j} = \|w\|_2$
- 7: $v_{j+1} = w / h_{j+1,j}$
- 8: **end for**

- Computes an orthonormal basis V_k of $\mathcal{K}(L, v_0)$, governed by the Arnoldi decomposition

$$LV_{k-1} = V_k \underline{H}_k,$$

$V_k = [v_1, v_2, \dots, v_k]$, $\underline{H}_k \in \mathbb{R}^{k \times k-1}$ upper Hessenberg.

13. Let $V_k = \begin{bmatrix} Q_k U_{k,1} \\ Q_k U_{k,2} \end{bmatrix}$ to replace V_k in the Arnoldi decomposition:

$$\underbrace{\begin{bmatrix} A & B \\ I & 0 \end{bmatrix}}_L \cdot \underbrace{\begin{bmatrix} Q_{k-1} U_{k-1,1} \\ Q_{k-1} U_{k-1,2} \end{bmatrix}}_{V_{k-1}} = \underbrace{\begin{bmatrix} Q_k U_{k,1} \\ Q_k U_{k,2} \end{bmatrix}}_{V_k} \underline{H}_k,$$

then we can derive the TOAR procedure.

14. TOAR procedure

```
1: Rank revealing QR:  $\begin{bmatrix} r_{-1} & r_0 \end{bmatrix} = QX$  with  $\eta_1$  being the rank.
2: Initialize  $Q_1 = Q$ ,  $U_{1,1} = X(:, 2)/\gamma$  and  $U_{1,2} = X(:, 1)/\gamma$ .
3: for  $j = 1, 2, \dots, k-1$  do
4:    $r = A(Q_j U_{j,1}(:, j)) + B(Q_j U_{j,2}(:, j))$ 
5:   for  $i = 1, \dots, \eta_j$  do
6:      $s_i = q_i^T r$ 
7:      $r = r - s_i q_i$ 
8:   end for
9:    $\alpha = \|r\|_2$ 
10:  Set  $s = [s_1, \dots, s_{\eta_j}]^T$  and  $u = U_{j,1}(:, j)$ 
11:  for  $i = 1, \dots, j$  do
12:     $h_{ij} = U_{j,1}(:, i)^T s + U_{j,2}(:, i)^T u$ 
13:     $s = s - h_{ij} U_{j,1}(:, i)$ ;  $u = u - h_{ij} U_{j,2}(:, i)$ 
14:  end for
15:   $h_{j+1,j} = (\alpha^2 + \|s\|_2^2 + \|u\|_2^2)^{1/2}$ 
16:  if  $h_{j+1,j} = 0$  then
17:    stop (breakdown)
18:  end if
19:  if  $\alpha = 0$  then
20:     $\eta_{j+1} = \eta_j$  (deflation)
21:     $Q_{j+1} = Q_j$ ;  $U_{j+1,1} = [U_{j,1}, s/h_{j+1,j}]$ ;  $U_{j+1,2} = [U_{j,2}, u/h_{j+1,j}]$ 
22:  else
23:     $\eta_{j+1} = \eta_j + 1$ 
24:     $Q_{j+1} = [Q_j, r/\alpha]$ ;  $U_{j+1,1} = \begin{bmatrix} U_{j,1} & s/h_{j+1,j} \\ 0 & \alpha/h_{j+1,j} \end{bmatrix}$ ;  $U_{j+1,2} = \begin{bmatrix} U_{j,2} & u/h_{j+1,j} \\ 0 & 0 \end{bmatrix}$ 
25:  end if
26: end for
```

15. Arnoldi decomposition in exact arithmetic:

$$\underbrace{\begin{bmatrix} A & B \\ I & 0 \end{bmatrix}}_L \cdot \underbrace{\begin{bmatrix} Q_{k-1} \hat{U}_{k-1,1} \\ Q_{k-1} \hat{U}_{k-1,2} \end{bmatrix}}_{V_{k-1}} = \underbrace{\begin{bmatrix} Q_k U_{k,1} \\ Q_k U_{k,2} \end{bmatrix}}_{V_k} \underline{H}_k.$$

16. In floating point arithmetic, computed quantities satisfy the perturbed Arnoldi decomposition

$$\underbrace{\left(\begin{bmatrix} A & B \\ I & 0 \end{bmatrix} + \Delta L \right)}_L \begin{bmatrix} \hat{Q}_{k-1} \hat{U}_{k-1,1} \\ \hat{Q}_{k-1} \hat{U}_{k-1,2} \end{bmatrix} = \begin{bmatrix} \hat{Q}_k \hat{U}_{k,1} \\ \hat{Q}_k \hat{U}_{k,2} \end{bmatrix} \hat{H}_k.$$

17. Theorem: Suppose \hat{Q}_k and \hat{U}_k have full column rank. Then the relative backward error satisfies

$$\frac{\|\Delta L\|_F}{\|L\|_F} \leq \varphi \kappa^4 \varepsilon + \mathcal{O}(\varepsilon^2),$$

where $\varphi = 4k(2n + 1)$ and $\kappa = \max\{\kappa_2(\hat{Q}_k), \kappa_2(\hat{U}_k)\}$.

18. Remark: when \hat{Q}_k and \hat{U}_k are generated by the modified Gram-Schmidt process with the partial reorthogonalization, we have $\kappa = 1 + \mathcal{O}(\varepsilon)$. Consequently, $\varphi \kappa^4 \approx \varphi$. Thus the theorem implies that the TOAR procedure is relatively backward stable.

19. Structure-preserving Rayleigh-Ritz projection method for the QEP

- 1: compute an orthonormal basis Q_k of $\mathcal{G}_k(-M^{-1}D, -M^{-1}K; r_{-1}, r_0)$ by SOAR/TOAR procedure
- 2: reduce the dimension of QEP by projection

$$(\lambda^2 M_k + \lambda D_k + K_k)w = 0,$$

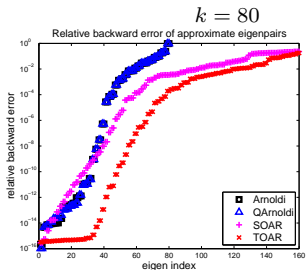
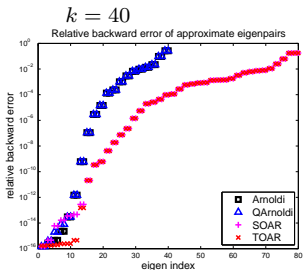
where $M_k = Q_k^H M Q_k$, $D_k = Q_k^H D Q_k$, and $K_k = Q_k^H K Q_k$.

- 3: compute the eigenpairs $(\hat{\lambda}, w)$ of the reduced QEP by linearization.
- 4: check the accuracy of the approximate eigenpairs $(\hat{\lambda}, \hat{x} = Q_k w)$ from the relative backward error:

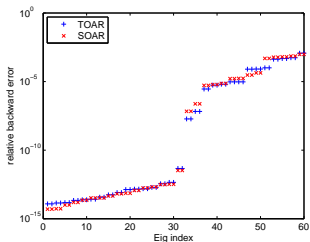
$$r = \frac{\|(\hat{\lambda}^2 M + \hat{\lambda} D + K)\hat{x}\|}{(|\hat{\lambda}|^2 \|M\| + |\hat{\lambda}| \|D\| + \|K\|)\|\hat{x}\|}.$$

20. MATLAB script `demo_toar.m`

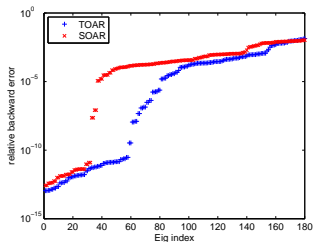
21. Example: vibration of wiresaw manufacturing process, $n = 800$.



22. Example: Butterfly Gyroscope, $n = 17,361$



$k = 30$



$k = 90$

23. Remarks

- ▶ TOAR is a numerically stable procedure for computing an orthonormal basis of the second-order Krylov subspace and is extensible to higher-order Krylov subspace.
- ▶ TOAR is a memory-efficient Arnoldi process for linearized QEPs and is a promising process to become routinely applied for polynomial and linearizable eigenvalue computations, see Part 2.4 of this lecture notes.
- ▶ Open problems: structured backward error bound of TOAR in terms of A and B , respectively? stability of implicit restarted TOAR?

24. Further reading

- ▶ 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), pp.640-659, 2005
- ▶ D. Lu, Y. Su and Z. Bai, Stability analysis of two-level orthogonal Arnoldi procedure, SIAM J. Matrix Anal. Appl. 37(1), pp.192-214, 2016
Software and data available at <http://www.unige.ch/~dlu/toar.html>
- ▶ K. Meerbergen and J. Pérez, Mixed forward-backward stability of the two-level orthogonality Arnoldi method for quadratic problems, arXiv:1707.00930v1, July 4, 2017.

Part 2.3(b) REP

1. Rational Eigenvalue Problem (REP)

$$R(\lambda)x = 0, \quad (2)$$

where $R(\lambda)$ is an $n \times n$ matrix rational function of the form

$$R(\lambda) = P(\lambda) - \sum_{i=1}^k \frac{s_i(\lambda)}{q_i(\lambda)} E_i, \quad (3)$$

$P(\lambda)$ is an $n \times n$ matrix polynomial in λ of degree d , $s_i(\lambda)$ and $q_i(\lambda)$ are scalar polynomials of degrees n_i and d_i , respectively, and $E_i = L_i U_i^T$, $L_i, U_i \in \mathbb{R}^{n \times r_i}$ of full column rank $r_i \ll n$.

2. Assume that $s_i(\lambda)$ and $q_i(\lambda)$ are *coprime* (i.e., no common factors), and $\frac{s_i(\lambda)}{q_i(\lambda)}$ are *proper* (i.e., $s_i(\lambda)$ having smaller degree than $q_i(\lambda)$). Then we have

$$\frac{s_i(\lambda)}{q_i(\lambda)} = a_i^T (C_i - \lambda D_i)^{-1} b_i,$$

for some matrix $C_i \in \mathbb{R}^{d_i \times d_i}$ and nonsingular matrix $D_i \in \mathbb{R}^{d_i \times d_i}$, and vectors $a_i, b_i \in \mathbb{R}^{d_i \times 1}$. The quadruple (C_i, D_i, a_i, b_i) is called a *minimal realization* in the theory of control system.

3. By the realizations of $s_i(\lambda)/q_i(\lambda)$ and the factorizations of E_i , the rational terms of $R(\lambda)$ can be rewritten as

$$\begin{aligned} \sum_{i=1}^k \frac{s_i(\lambda)}{q_i(\lambda)} E_i &= \sum_{i=1}^k a_i^T (C_i - \lambda D_i)^{-1} b_i L_i U_i^T = \sum_{i=1}^k L_i \left[a_i^T (C_i - \lambda D_i)^{-1} b_i \cdot I_{r_i} \right] U_i^T \\ &= \sum_{i=1}^k L_i (I_{r_i} \otimes a_i)^T (I_{r_i} \otimes C_i - \lambda I_{r_i} \otimes D_i)^{-1} (I_{r_i} \otimes b_i) U_i^T, \end{aligned}$$

where \otimes is the Kronecker product.

4. The REP (2) can be equivalently written in the realization (compact) form

$$R(\lambda)x = \left[P(\lambda) - L(C - \lambda D)^{-1} U^T \right] x = 0, \quad (4)$$

where

$$\begin{aligned} C &= \text{diag}(I_{r_1} \otimes C_1, I_{r_2} \otimes C_2, \dots, I_{r_k} \otimes C_k), \\ D &= \text{diag}(I_{r_1} \otimes D_1, I_{r_2} \otimes D_2, \dots, I_{r_k} \otimes D_k), \\ L &= \left[\begin{array}{cccc} L_1(I_{r_1} \otimes a_1)^T & L_2(I_{r_2} \otimes a_2)^T & \cdots & L_k(I_{r_k} \otimes a_k)^T \end{array} \right], \\ U &= \left[\begin{array}{cccc} U_1(I_{r_1} \otimes b_1)^T & U_2(I_{r_2} \otimes b_2)^T & \cdots & U_k(I_{r_k} \otimes b_k)^T \end{array} \right], \end{aligned}$$

Note that the size of C and D is $m \times m$, the size of L and U is $n \times m$, and $m = r_1 d_1 + r_2 d_2 + \dots + r_k d_k$. Furthermore, D is nonsingular since the matrices D_i are nonsingular. The eigenvalues of $C - \lambda D$ are the poles of $R(\lambda)$.

5. Example. Let $A, B \in \mathbb{C}^{n \times n}$, $c, d \in \mathbb{C}^n$, and

$$\begin{aligned}
 R(\lambda) &= A - \lambda B + \frac{\lambda}{\lambda - \sigma_1} cc^T + \frac{\lambda}{\lambda - \sigma_2} dd^T \\
 &= A - \lambda B + \left(1 - \frac{\sigma_1}{\sigma_1 - \lambda}\right) cc^T + \left(1 - \frac{\sigma_2}{\sigma_2 - \lambda}\right) dd^T \\
 &= A + cc^T + dd^T - \lambda B - \frac{\sigma_1}{\sigma_1 - \lambda} cc^T - \frac{\sigma_2}{\sigma_2 - \lambda} dd^T \\
 &= A + cc^T + dd^T - \lambda B - c(1 - \lambda\sigma_1^{-1})c^T - d(1 - \lambda\sigma_2^{-1})d^T \\
 &= \underline{A + cc^T + dd^T - \lambda B} - [c \ d] (I - \lambda\Sigma)^{-1} [c \ d]^T \\
 &\equiv P(\lambda) - L(C - \lambda D)^{-1} L^T
 \end{aligned}$$

where

$$\Sigma = \begin{bmatrix} \sigma_1^{-1} & \\ & \sigma_2^{-1} \end{bmatrix}$$

6. If $P(\lambda)$ is linear, $P(\lambda) = A - \lambda B$, then the REP (4) is of the form

$$\left[A - \lambda B - L(C - \lambda D)^{-1}U^T \right] x = 0. \quad (5)$$

By introducing the auxiliary vector

$$y = -(C - \lambda D)^{-1}U^T x,$$

the equation (4) can be written as the following LEP:

$$(\mathbf{A} - \lambda \mathbf{B})\mathbf{z} = 0, \quad (6)$$

where

$$\mathbf{A} = \begin{bmatrix} A & L \\ U^T & C \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} B & \\ & D \end{bmatrix}, \quad \mathbf{z} = \begin{bmatrix} x \\ y \end{bmatrix}.$$

7. If the matrix polynomial $P(\lambda)$ is of degree d , we can first write the REP (4) as a "PEP" of the form

$$\left(\lambda^d A_d + \lambda^{d-1} A_{d-1} + \cdots + \lambda A_1 + \tilde{A}_0(\lambda) \right) x = 0, \quad (7)$$

where $\tilde{A}_0(\lambda) \triangleq A_0 - L(C - \lambda D)^{-1} U^T$. Then by symbolically applying the companion form linearization to (7), we have

$$\left(\begin{bmatrix} A_{d-1} & A_{d-2} & \cdots & \tilde{A}_0(\lambda) \\ -I & 0 & \cdots & 0 \\ & \ddots & \ddots & \vdots \\ & & -I & 0 \end{bmatrix} - \lambda \begin{bmatrix} A_d & & & \\ & I & & \\ & & \ddots & \\ & & & I \end{bmatrix} \right) \begin{bmatrix} \lambda^{d-1} x \\ \lambda^{d-2} x \\ \vdots \\ x \end{bmatrix} = 0, \quad (8)$$

8. It can be equivalently written as

$$\left(\begin{bmatrix} A_{d-1} & A_{d-2} & \cdots & A_0 \\ -I & 0 & \cdots & 0 \\ & \ddots & \ddots & \vdots \\ & & -I & 0 \end{bmatrix} - \lambda \begin{bmatrix} A_d & & & \\ & I & & \\ & & \ddots & \\ & & & I \end{bmatrix} \right) \begin{bmatrix} L \\ 0 \\ \vdots \\ 0 \end{bmatrix} (C - \lambda D)^{-1} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ U \end{bmatrix}^T \begin{bmatrix} \lambda^{d-1} x \\ \lambda^{d-2} x \\ \vdots \\ x \end{bmatrix} = 0.$$

9. The above equation is of the same form as (5). Therefore, by introducing the variable

$$y = -(C - \lambda D)^{-1} \begin{bmatrix} 0 & 0 & \dots & U^T \end{bmatrix} \begin{bmatrix} \lambda^{d-1} x \\ \lambda^{d-2} x \\ \vdots \\ x \end{bmatrix} = -(C - \lambda D)^{-1} U^T x,$$

we derive the following linearization of the REP (2):

$$(\mathbf{A} - \lambda \mathbf{B})\mathbf{z} = 0, \quad (9)$$

where \mathbf{A} and \mathbf{B} are $nd + m$, and $m = r_1 d_1 + r_2 d_2 + \dots + r_k d_k$:

$$\mathbf{A} = \left[\begin{array}{cccc|c} A_{d-1} & A_{d-2} & \dots & A_0 & L \\ -I & 0 & \dots & 0 & \\ & & \ddots & \vdots & \\ & & & -I & 0 \\ \hline & & & & U^T \\ & & & & C \end{array} \right], \quad \mathbf{B} = - \left[\begin{array}{ccc|c} A_d & & & \\ & I & & \\ & & \ddots & \\ & & & I \\ \hline & & & -D \end{array} \right], \quad \mathbf{z} = \begin{bmatrix} \lambda^{d-1} x \\ \lambda^{d-2} x \\ \vdots \\ x \\ \hline y \end{bmatrix}.$$

The size of \mathbf{A} and \mathbf{B} is $nd + m$, and $m = r_1 d_1 + r_2 d_2 + \dots + r_k d_k$.

10. In the case that all the coefficient matrices E_i are of full rank, i.e., $r_i = n$, the LEP (9) is of the size nd_* , where $d_* = d + d_1 + \dots + d_k$. This is the same size as the one derived by the brute-force approach. However, it is typical that $r_i \ll n$ in practice, then $nd + m \ll nd_*$. The LEP (9) is a **trimmed linearization** of the REP (2).
11. Note that under the assumption of nonsingularity of A_d , \mathbf{B} is nonsingular. Therefore all eigenvalues of the LEP (9) are finite. There is no infinite eigenvalue.

12. Connection between eigenvalues of the REP (2) and the LEP (9):

- (a) If λ is an eigenvalue of the REP (2), then it is an eigenvalue of the LEP (9).
- (b) Let λ be an eigenvalue of the LEP (9) and be not a pole of $R(\lambda)$,
 $\mathbf{z} = [z_1^T, z_2^T, \dots, z_d^T, y^T]^T$ be the corresponding eigenvector, where z_i are vectors of length n . Then $z_d \neq 0$ and $R(\lambda)z_d = 0$, namely, λ is an eigenvalue of the REP (2) and z_d is the corresponding eigenvector. Moreover, the algebraic and geometric multiplicities of λ for REP (2) and LEP (9) are the same.

13. Remarks:

- ▶ The condition that λ is not a pole of the $R(\lambda)$ in the theorem is necessary. Consider

$$\left(\lambda I_2 - \frac{1}{\lambda} e_2 e_2^T \right) x = 0, \quad (10)$$

where e_2 is the second column of I_2 . Since $\det(R(\lambda)) = \lambda(\lambda - 1/\lambda)$, the REP (10) has two eigenvalues 1 and -1 . Moreover, $\lambda = 0$ is a pole. Let $y = \lambda^{-1} e_2^T x$, then the corresponding LEP is given by

$$(\mathbf{A} - \lambda \mathbf{B})\mathbf{z} = \left(\left[\begin{array}{cc} 0 & e_2 \\ e_2^T & 0 \end{array} \right] - \lambda \left[\begin{array}{cc} I_2 & \\ & 1 \end{array} \right] \right) \begin{bmatrix} x \\ y \end{bmatrix} = 0.$$

It has three eigenvalues $\{0, \pm 1\}$. But $\lambda = 0$ is not an eigenvalue of the REP (10).

- ▶ The realization of a rational function can be represented in different forms.
- ▶ There are many different ways of linearization for the matrix polynomials, see [Part 2.4](#).

14. Example.

- ▶ REP from mechanical vibration of a fluid-solid structure:

$$\left(A - \lambda B + \sum_{i=1}^k \frac{\lambda}{\lambda - \sigma_i} C_i C_i^T \right) x = 0, \quad (11)$$

where A and B are symmetric positive definite, and for $i = 1, 2, \dots, k$, the poles σ_i are positive, $C_i \in \mathbb{R}^{n \times r_i}$ has rank r_i .

We are interested in finding/determining the number of eigenvalues of the REP in a given interval (α, β) .

- ▶ We first write the rational terms of (11) in the proper form

$$\left(A + \sum_{i=1}^k C_i C_i^T - \lambda B - \sum_{i=1}^k \frac{\sigma_i}{\sigma_i - \lambda} C_i C_i^T \right) x = 0. \quad (12)$$

- ▶ Let $C = [C_1 \ C_2 \ \dots \ C_k]$ and $\Sigma = \text{diag}(\sigma_1 I_{r_1}, \dots, \sigma_k I_{r_k})$, Then the equation (12) can be written as

$$\left[A + C C^T - \lambda B - C(I - \lambda \Sigma^{-1})^{-1} C^T \right] x = 0.$$

- ▶ By introducing the variable $y = -(I - \lambda \Sigma^{-1})^{-1} C^T x$, we have the following LEP:

$$(\mathbf{A} - \lambda \mathbf{B}) \begin{bmatrix} x \\ y \end{bmatrix} = 0, \quad (13)$$

where

$$\mathbf{A} = \begin{bmatrix} A + C C^T & C \\ C^T & I \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} B & \\ & \Sigma^{-1} \end{bmatrix}$$

14. Example, cont'd

- ▶ Note that the matrix \mathbf{A} is symmetric, and \mathbf{B} is symmetric positive definite.
- ▶ Numerical experiments:
 - ▶ Problem size $n = 36046$, rational terms $k = 9$. and $\text{rank}(C_i) = 2$. The pole $\sigma_i = i$ for $i = 1, 2, \dots, k$.
 - ▶ Compute all eigenvalues in the interval $(\alpha, \beta) = (1, 2)$
 - ▶ It only takes about 13.3% extra time to solve the full REP than the simple eigenvalue problem of the pencil $A - \lambda B$.

15. Further reading:

- ▶ Y. Su and Z. Bai, Solving rational eigenvalue problems via linearization, SIAM J. of Matrix Analysis and Applications, Vol.32, No.1, pp.201-216, 2011
- ▶ R. Alam, N. Behera, Linearizations for rational matrix functions and Rosenbrock system polynomials, SIAM J. Matrix Anal. Appl., 37, pp.354-380, 2016.
- ▶ A. Amparan, F. M. Dopico, S. Marcaida and I. Zaballa, Strong linearizations of rational matrices, MIMS EPrints 2016.51, Univ. of Manchester, Oct. 2016.
- ▶ F. M. Dopico and J. Gonzalez-Pizarro, A compact rational Krylov method for large-scale rational eigenvalue problems, arXiv:1705.06982v1, May 19, 2017
- ▶ C. Engström, H. Langer and C. Tretter, Non-linear eigenvalue problems and applications to photonic crystals, arXiv:1607.06381v1, July 23, 2015

Part 2.3(c) QEP with low-rank damping

1. The QEP with low-rank damping:

▶ QEP:

$$Q(\lambda)x = (\lambda^2 M + \lambda C + K)x = 0,$$

▶ Low-rank property

$$\text{rank}(C) = r \ll n$$

2. The low-rank property is observed in **all** practical QEPs we have encountered.

3. Standard approach: to find eigenvalues around the shift σ , by the spectral transformation $\mu = \lambda - \sigma$, QEP is equivalent to the LEP, say in the first companion form:

$$\begin{bmatrix} -C_\sigma & -K_\sigma \\ I_n & 0 \end{bmatrix} \begin{bmatrix} \mu x \\ x \end{bmatrix} = \mu \begin{bmatrix} M & 0 \\ 0 & I_n \end{bmatrix} \begin{bmatrix} \mu x \\ x \end{bmatrix},$$

where $C_\sigma = C + 2\sigma M$ and $K_\sigma = \sigma^2 M + \sigma C + K$.

4. Alternatively, use the following structure-preserving subspace projection method:

- 1: construction of a proper projection subspace V_k by TOAR
- 2: compute the structure-preserving projection

$$\left(\theta^2 V_k^T M V_k + \theta V_k^T C V_k + V_k^T K V_k \right) z = 0,$$

- 2: Solve the reduced QEP and test convergence

5. All these approaches ignore the low-rank property of C !

6. To exploit the low-rank property, consider the following rather *unusual* four-step approach, called “Padé Approximate Linearization (PAL)”:

- 1: QEP converted to NEP via a “quadratic” spectral transformation
- 2: REP approximation of NEP via a Padé approximant
- 3: Trimmed linearization of REP
- 4: A scaling strategy to minimize the backward error.

7. Step 1. QEP to NEP via “quadratic” spectral transformations

- ▶ Spectral transformations:

$$g: \mathbb{S}_\sigma \longrightarrow \mathbb{C}$$

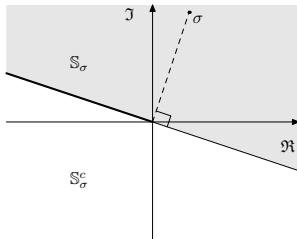
$$\lambda \longrightarrow \mu = \frac{\lambda^2}{\sigma^2} - 1$$

$$f: \mathbb{C} \longrightarrow \mathbb{S}_\sigma$$

$$\mu \longrightarrow \lambda = \sigma \sqrt{\mu + 1}$$

where

$$\mathbb{S}_\sigma \equiv \left\{ w \mid \arg \left(\frac{w}{\sigma} \right) \in \left(-\frac{\pi}{2}, \frac{\pi}{2} \right) \right\}$$



- ▶ QEP \Leftrightarrow NEP

$$Q(\lambda)x \equiv (\lambda^2 M + \lambda C + K)x = 0$$

$$\Updownarrow$$

$$T(\mu)x = [K_\sigma - \mu M_\sigma + f(\mu)C]x = 0$$

where $M_\sigma = -\sigma^2 M$ and $K_\sigma = K + \sigma^2 M$.

- ▶ (1) If (λ, x) is an eigenpair of $Q(\lambda)$ and $\lambda \in \mathbb{S}_\sigma$, then $(\mu = g(\lambda), x)$ is an eigenpair of $T(\mu)$.
- ▶ (2) If (μ, x) is an eigenpair of $T(\mu)$, then $(\lambda = f(\mu), x)$ is an eigenpair of $Q(\lambda)$.

8. Step 2. REP approximation of NEP via Padé approximant

- ▶ Padé approximation:

$$\begin{aligned} f(\mu) &= \sigma \sqrt{\mu + 1} = \underline{\sigma p_{mm}(\mu)} + \sigma e(\mu) \\ &= \underline{-\sigma a^T (I_m - \mu D)^{-1} a + \sigma d} + \mathcal{O}(\sigma \mu^{2m+1}) \end{aligned}$$

- ▶ The NEP becomes

$$T(\mu)x = \left(K_\sigma - \mu M_\sigma + \sigma p_{mm}(\mu)C + \sigma e(\mu)C \right) x = 0$$

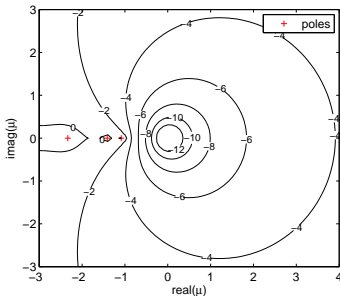
- ▶ REP approximation of the NEP

$$R(\mu)x \equiv \left(K_\sigma - \mu M_\sigma + \sigma p_{mm}(\mu)C \right) x = 0$$

and the poles of $R(\mu)$ are in the interval $(-\infty, -1)$,

- ▶ Assumption/justification:

- Corresponding to the interested eigenvalues λ of the QEP close to σ , $\mu = \frac{\lambda^2}{\sigma^2} - 1 \notin (-\infty, -1)$.
- $\|\sigma e(\mu)C\|$ is relatively small due to high accurate Padé approximation



9. Step 3. Trimmed linearization of REP

- ▶ LEP by a trimmed linearization of REP (assuming $C = FF^T$):

$$L(\mu)x_L = (\mathbf{A} - \mu\mathbf{B})x_L = 0$$

where

$$\mathbf{A} = \begin{bmatrix} K_\sigma + \sigma dC & F_\sigma \\ F_\sigma^T & I_{rm} \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} M_\sigma & 0 \\ 0 & I_r \otimes D \end{bmatrix}, \quad x_L = H(\mu)x$$

and

$$M_\sigma = -\sigma^2 M, \quad K_\sigma = K + \sigma^2 M,$$

$$F_\sigma = \sigma^{1/2} F(I_r \otimes a^T), \quad H(\mu) = \begin{bmatrix} I_n \\ -(I_{rm} - \mu I_r \otimes D)^{-1} F_\sigma^T \end{bmatrix}.$$

- ▶ **Theorem:** REP and LEP are equivalent.

10. Features of PAL

- ▶ By exploiting the low-rank property, the size of the LEP

$$n_L = n + rm$$

where $r = \text{rank}(C)$, and m is the order of the Padé approximant

- ▶ For the matrix-vector multiplication $v = \mathbf{A}^{-1}\mathbf{B}u$, we have

$$\mathbf{A}^{-1} = \begin{bmatrix} I_n & \\ -F_\sigma^T & I_{rm} \end{bmatrix} \begin{bmatrix} Q(\sigma)^{-1} & \\ & I_{rm} \end{bmatrix} \begin{bmatrix} I_n & -F_\sigma \\ & I_{rm} \end{bmatrix}.$$

where $Q(\sigma) = \sigma^2 M + \sigma C + K$. This is similar to the standard linearization.

- ▶ More can be exploited on the structure and properties of (\mathbf{A}, \mathbf{B}) , depending on (M, C, K) and the choice of σ .

11. *A posteriori* error bound

- ▶ Let $(\hat{\mu}, \hat{x}_L)$ be a computed eigenpair of the LEP. Then for the approximate eigenpair

$$(\hat{\lambda}, \hat{x}) = (f(\hat{\mu}), \hat{x}_{L(1:n)})$$

of the QEP, we have the following *a posteriori* error bound

$$\eta_Q(\hat{\lambda}, \hat{x}) \leq \alpha \cdot \eta_L(\hat{\mu}, \hat{x}_L) + \beta$$

where

- ▶ η_Q and η_L are normalized backward errors of QEP and LEP:

$$\eta_Q(\hat{\lambda}, \hat{x}) = \frac{\|Q(\hat{\lambda})\hat{x}\|}{\rho(\hat{\lambda})\|\hat{x}\|}, \quad \eta_L(\hat{\mu}, \hat{x}_L) = \frac{\|L(\hat{\mu})\hat{x}_L\|}{\varphi(\hat{\mu})\|\hat{x}_L\|},$$

and $\rho(\hat{\lambda}) = |\hat{\lambda}|^2 \|M\| + |\hat{\lambda}| \|C\| + \|K\|$, $\varphi(\hat{\mu}) = \|A\| + |\hat{\mu}| \|B\|$.

- ▶ α and β are given by

$$\alpha = \frac{\varphi(\hat{\mu})}{\rho(\hat{\lambda})} \|H(\hat{\mu})\|^2 \quad \text{and} \quad \beta = \frac{|\sigma e(\hat{\mu})|}{\rho(\hat{\lambda})} \frac{\|C\hat{x}\|}{\|\hat{x}\|}.$$

12. Step 4. Minimizing the backward error

- ▶ α is an error growth amplifier from the LEP to QEP. Under mild assumptions, with an “optimal” scaling of the QEP, we have

$$\alpha \lesssim 2(1 + 4m)$$

where m is the order of diagonal Padé approximant.

- ▶ β is largely determined by

$$\frac{|\sigma e(\hat{\mu})|}{\rho(\hat{\lambda})} = \mathcal{O}(\text{Padé truncation error})$$

However, we might have “*extra accuracy bonus*”

$$\frac{\|C\hat{x}\|}{\|\hat{x}\|} = \text{tiny}$$

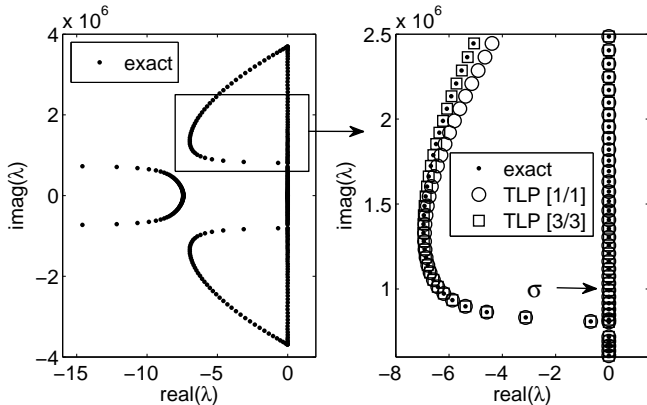
due to the closeness of eigenspaces of undamped and damped QEPs.

13. PAL algorithm

- 1: select a shift $\sigma \neq 0$ and the order m of Padé approximant
- 2: compute the scaling factor s
- 3: form the scaled LEP $L_s(\mu)x_L = (\mathbf{A}_s - \mu\mathbf{B}_s)x_L = 0$ *implicitly*
- 4: compute the k smallest (in modulus) eigenpairs $(\hat{\mu}, \hat{x}_L)$ of L_s with the backward errors satisfying $\eta_{L_s}(\hat{\mu}, \hat{x}_L) \leq \text{tol}$
- 5: return k approximate eigenpairs $(\hat{\lambda}, \hat{x}) = (f(\hat{\mu}), \hat{x}_{L(1:n)})$ of QEP and the corresponding backward errors $\eta_Q(\hat{\lambda}, \hat{x})$.

14. Example 1. damped_beam in NLEVP collection

- ▶ $n = 200$, $\text{rank}(C) = 1$.
shift $\sigma = 10^6 i$, orders of Padé : $m = 1, 3$
PAL leads to LEPs of orders $n_L = n + rm$: 201, 203
- ▶ Accuracy



14. Example 1, cont'd

- ▶ Effectiveness of the error bound

$$\eta_Q(\hat{\lambda}, \hat{x}) \leq \alpha \cdot \eta_L(\hat{\mu}, \hat{x}_L) + \beta$$

#	α	η_{L_s}	β	$\alpha \cdot \eta_{L_s} + \beta$	η_Q
1	1.061	6.64E-16	4.78E-24	7.04E-16	6.93E-16
2	1.019	5.00E-16	7.26E-19	5.10E-16	5.02E-16
3	1.012	5.31E-16	2.22E-18	5.40E-16	5.35E-16
4	1.038	1.01E-15	8.55E-14	8.65E-14	8.55E-14
5	1.020	5.94E-16	1.71E-9	1.71E-9	1.71E-9
6	1.013	5.75E-16	4.06E-9	4.06E-9	4.06E-9

15. Computational efficiency for a C++ implementation of PAL.

▶ Example 1:

- ▶ Acoustic-wave-2d in NLEVP collection, $n = 249500$, $\text{rank}(C) = 499$, $m = 3$.
- ▶ CPU timing (in seconds) of computing $k = 300$ eigenpairs:

	SpMVs	GS	EvComp	Updating	Subtotal
DLIN	168.95	1130.44	314.65	304.00	1918.04
PAL	162.26	562.37	137.66	156.10	1018.39

- ▶ the SpMV costs for the two linearizations are almost the same.
 - ▶ The bulk of computational time lies in the Gram-Schmidt orthogonalization process, where PAL reduces the cost by almost half.
 - ▶ PAL runs 47% faster than DLIN (Direct LINearization).
- ### ▶ Example 2:

- ▶ Frequency responses of a car body from SIEMENS, $n = 655812$, $\text{rank}(C) = 126$, $m = 3$
- ▶ CPU timing of computing $k = 300$ eigenpairs:

	SpMV	GS	EvComp	Update	Subtotal
DLIN	1305.94	2277.01	791.25	393.81	4768.01
PAL	1297.92	1146.89	394.37	202.43	3459.99

- ▶ The SpMV cost is high in this example, but is almost the same for PAL and DLIN.
- ▶ The bulk of computational time still lies in the Gram-Schmidt orthogonalization process and PAL reduces it by almost half.
- ▶ PAL runs 33.4% faster than DLIN.

16. PAL summary

- ▶ PAL is an efficient approach for large-scale QEPs with low-rank damping.
- ▶ Low-rank structure is common in matrix computations How to exploit the low-rank structure in eigenvalue computations remains widely open (vs. linear system solvers).
more to see in Part 2.4(g) and 2.4(h)

17. Further reading:

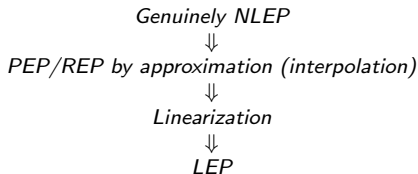
- ▶ D. Lu, X. Huang, Z. Bai and Y. Su, A Padé approximate linearization algorithm for solving the quadratic eigenvalue problem with low-rank damping, Int. J. Numer. Meth. Engng, Vol.103, pp.840-858, 2015
Software and data available at <http://www.unige.ch/~dlu/palm.html>
- ▶ For the treatment of the dense QEP with low-rank damping, see
L. Taslaman, An algorithm for quadratic eigenproblems with low rank damping, MIMS Eprint 2014.21, The University of Manchester, UK.

Part 2.4 Methods based approximation and linearization

Outline:

- (a) The method of successive linear approximation
- (b) Linearization
- (c) PEP in monomial basis and linearization
- (d) Lagrange interpolation in barycentric form and linearization
- (e) Newton interpolation and linearization
- (f) (Rational) Padé approximation and linearization
- (g) Rational interpolation and linearization
- (h) Algorithmic framework and software

Algorithmic framework:



Part 2.4(a) The method of successive linear approximation

1. By the Taylor expansion

$$T(\lambda + h) = T(\lambda) + hT'(\lambda) + \frac{1}{2}h^2R(\lambda, h)$$

by discarding R , the Successive Linear Approximation Method (SLAM):

- 1: choose λ_0 appropriately
 - 2: solve LEVP: $-T(\lambda_s)x_{s+1} = \mu_s T'(\lambda_s)x_{s+1}$
 - 3: $\lambda_{s+1} = \lambda_s + \mu_s$
2. Case study: the SLAM on the nonlinear low-rank modification of a symmetric eigenvalue problem

$$(A + s(\lambda)FF^T)x = \lambda x$$

from vibration of mechanical structures with elastically attached loads and propagation modes in optical fiber.

- ▶ Existence and uniqueness of eigenvalues, interlacing properties
 - ▶ The global convergence of the SLAM
 - ▶ Numerical experiments illustrate the robustness of SLAM.
3. References:
 - ▶ A. Ruhe, Algorithms for the nonlinear eigenvalue problem, SIAM J. Numer. Anal. 10, pp.674-689, 1973
 - ▶ H. Voss, K. Yildiztekin and X, Huang, Nonlinear low-rank modification of a symmetric eigenvalue problem, SIAM J. Matrix Anal. Appl. 32, pp515-535, 2011.
($A + \phi(\lambda)H$) $x = \lambda x$).

Part 2.4(b) Linearization

1. Matrix polynomials of degree d

$$P(\lambda) = \sum_{i=1}^d A_i \lambda^i, \quad \text{where } A_i \in \mathbb{C}^{n \times n} \text{ and } A_d \neq 0.$$

and $P(\lambda)$ is regular.

2. $P(\lambda)$ is uniquely determined by $n + 1$ samples: $P_j = P(z_j)$ where the points z_0, z_1, \dots, z_n are distinct.
3. A linear pencil $L(\lambda) = \mathbf{A} - \lambda \mathbf{B}$ is called a **(weak) linearization** of $P(\lambda)$ if there exist unimodular⁵ matrix polynomials $E(\lambda)$ and $F(\lambda)$ such that

$$E(\lambda)L(\lambda)F(\lambda) = \begin{bmatrix} P(\lambda) & 0 \\ 0 & I_{(d-1)n} \end{bmatrix}$$

4. The “reverse” polynomial of $P(\lambda)$: $P^\sharp(\lambda) = \lambda^d P(\lambda^{-1})$.
 - ▶ The nonzero finite eigenvalues of $P^\sharp(\lambda)$ are the reciprocals of those of $P(\lambda)$ and that an eigenvalue at ∞ of $P(\lambda)$ corresponds to an eigenvalue 0 of the reversal polynomial $P^\sharp(\lambda)$.

⁵A matrix polynomial $E(\lambda)$ is unimodular if $\det(E(\lambda)) = \text{constant}$

5. If $L(\lambda)$ is a linearization of $P(\lambda)$ and $L^\sharp(\lambda)$ is a linearization of $P^\sharp(\lambda)$, then $L(\lambda)$ is a **strong linearization** of $P(\lambda)$.

- ▶ The strong linearization ensures that for regular matrix polynomial the Jordan structure of the eigenvalues ∞ is preserved.

6. References

- ▶ I. Gohberg, P. Lancaster, and L. Rodman, Matrix Polynomials, Academic Press, New York, 1982.
- ▶ I. Gohberg, M. A. Kaashoek, and P. Lancaster, General theory of regular matrix polynomials and band Toeplitz operators, Integral Equations Operator Theory, 11, pp.776–882, 1988

Part 2.4(c) PEP in monomial basis and linearization

1. Consider the matrix polynomial in monomial basis

$$P(\lambda) = \sum_{i=1}^d A_i \lambda^i, \quad \text{where } A_i \in \mathbb{C}^{n \times n} \text{ and } A_d \neq 0.$$

Then the $dn \times dn$ linear (companion) pencil $L(\lambda) = \mathbf{A} - \lambda \mathbf{B}$ is a strong linearization of $P(\lambda)$, where

$$\mathbf{A} = \begin{bmatrix} A_0 & A_1 & \cdots & A_{d-1} \\ & I & & \\ & & \ddots & \\ & & & I \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 & 0 & \cdots & -A_d \\ I & 0 & & \\ & \ddots & \ddots & \\ & & I & 0 \end{bmatrix}.$$

2. $L(\lambda)$ is also called the companion pencil of $P(\lambda)$.
3. A useful identity

$$L(\lambda)(A(\lambda) \otimes I_n) = e_1 \otimes P(\lambda)$$

where

$$A(\lambda) = \begin{bmatrix} 1 \\ \lambda \\ \vdots \\ \lambda^{d-1} \end{bmatrix}.$$

4. Connection of eigenvalues and eigenvectors

- (a) If (λ_*, x) is an eigenpair of $P(\lambda)$, then $(\lambda_*, \Lambda(\lambda_*) \otimes x)$ is an eigenpair of $L(\lambda)$.
- (b) If (λ_*, \mathbf{x}) is an eigenpair of $L(\lambda)$, then there exists a vector x such that $\mathbf{x} = \Lambda(\lambda_*) \otimes x$ and the pair (λ_*, x) is an eigenpair of $P(\lambda)$.

5. Reference

- ▶ I. Gohberg, M. A. Kaashoek, and P. Lancaster, General theory of regular matrix polynomials and band Toeplitz operators, *Integral Equations Operator Theory*, 11, pp.776–882, 1988

Part 2.4(d) Lagrange interpolation in barycentric form and Linearization

1. Lagrange interpolation of scalar function $f(\lambda)$ in **classical form**

$$p(\lambda) = \sum_{i=0}^d f(\sigma_i) \ell_i(\lambda) \quad \text{with} \quad \ell_i(\lambda) = \frac{\prod_{k=0, k \neq i}^d (\lambda - \sigma_k)}{\prod_{k=0, k \neq i}^d (\sigma_i - \sigma_k)},$$

where σ_i are distinct interpolation points.

2. Shortcomings of the classical form:

- ▶ adding a new interpolation point requires computations from scratch
- ▶ computation is numerically unstable

3. Lagrange interpolation in **barycentric form**

$$p(\lambda) = \sum_{i=0}^d f(\sigma_i) b_i(\lambda)$$

where for $i = 0, 1, \dots, d$,

$$b_i(\lambda) = \frac{1}{b(\lambda)} \frac{w_i}{\lambda - \sigma_i},$$

with

$$b(\lambda) = \sum_{i=0}^d \frac{w_i}{\lambda - \sigma_i}, \quad w_i = \frac{1}{\prod_{k \neq i} (\sigma_i - \sigma_k)}$$

4. Advantages of the barycentric form

- ▶ Updating the weights w_i in $O(d)$ flops incorporate a new data pair (σ_{d+1}, f_{d+1})
- ▶ Computation is forward stable under mild assumption.

5. References:

- ▶ J.-P. Berrut and L. N. Trefethen, Barycentric Lagrange interpolation, SIAM Rev., 46, pp.501–517. 2004.
- ▶ N. J. Higham, The numerical stability of barycentric Lagrange interpolation, IMA J. Numer. Anal., 24, pp.547–556, 2004

6. Lagrange matrix interpolation of $T(\lambda)$ in barycentric form

$$P(\lambda) = \sum_{i=0}^d A_i b_i(\lambda) \quad \text{with } A_i = T(\sigma_i).$$

7. The $dn \times dn$ linear pencil $L(\lambda) = \mathbf{A} - \lambda\mathbf{B}$ is a strong linearization of $P(\lambda)$, where

$$\mathbf{A} = \begin{bmatrix} \sigma_1 A_0 & \sigma_2 A_1 & \cdots & \sigma_{d-1} A_{d-2} & \tilde{A}_d \\ \sigma_0 I & -\sigma_2 \theta_1 I & & & \\ & \ddots & \ddots & & \\ & & \ddots & -\sigma_{d-1} \theta_{d-2} I & \\ & & & \sigma_{d-2} I & -\sigma_d \theta_{d-1} I \end{bmatrix},$$

$$\mathbf{B} = \begin{bmatrix} A_0 & A_1 & \cdots & A_{d-2} & \tilde{B}_d \\ I & -\theta_1 I & & & \\ & \ddots & \ddots & & \\ & & \ddots & -\theta_{d-2} I & \\ & & & I & -\theta_{d-1} I \end{bmatrix}$$

and $\theta_i = w_{i-1}/w_i$ for $i = 1, 2, \dots, d$. $\tilde{A}_d = \sigma_d A_{d-1} + \sigma_{d-1} \theta_d^{-1} A_d$
 $\tilde{B}_d = A_{d-1} + \theta_d^{-1} A_d$.

8. A useful identity

$$L(\lambda)(\tilde{A}(\lambda) \otimes I) = e_1 \otimes P(\lambda).$$

where

$$\tilde{A}(\lambda) = \begin{bmatrix} \tilde{\ell}_0(\lambda) \\ \tilde{\ell}_1(\lambda) \\ \vdots \\ \tilde{\ell}_{n-1}(\lambda) \end{bmatrix} \quad \text{and} \quad \tilde{\ell}_i(\lambda) = \frac{\ell_i(\lambda)}{\lambda - \sigma_{i+1}}.$$

9. Connections between eigenvalues and eigenvectors

(a) If the pair (λ_*, x) is an eigenpair of $P(\lambda)$, then the pair $(\lambda_*, \tilde{A}(\lambda_*) \otimes x)$ is an eigenpair of $L(\lambda)$.

(b) If the pair (λ_*, \mathbf{x}) is an eigenpair of $L(\lambda)$, then there exists a vector x such that $\mathbf{x} = \tilde{A}(\lambda_*) \otimes x$ and the pair (λ_*, x) is an eigenpair of $P(\lambda)$.

10. Reference

- ▶ R. Van Beeumen, W. Michiels, and K. Meerbergen, Linearization of Lagrange and Hermite interpolating matrix polynomials, IMA J. Numer. Anal., 35, pp.909-930, 2014.

Part 2.4(e) Newton interpolation and Linearization

1. Scalar Newton interpolation of $f(\lambda)$:

$$p(\lambda) = \sum_{i=0}^d \alpha_i n_i(\lambda)$$

where $n_i(\lambda)$ are Newton basis functions:

$$n_0(\lambda) = 1, \quad n_i(\lambda) = \prod_{k=1}^i (\lambda - \sigma_{k-1}) \quad \text{for } i = 1, 2, \dots$$

and σ_i are distinct nodes, and α_i are the divided differences.

2. Opitz's method. The divided difference coefficients α_i are the elements in the first row of $f(M)$, where

$$M = \begin{bmatrix} \sigma_0 & 1 & & & & \\ & \sigma_1 & 1 & & & \\ & & \ddots & \ddots & & \\ & & & \ddots & \ddots & \\ & & & & \ddots & 1 \\ & & & & & \sigma_d \end{bmatrix}.$$

3. Matrix Newton polynomial interpolation of $T(\lambda)$

$$P(\lambda) = \sum_{i=0}^d A_i n_i(\lambda)$$

and the discussion on computing the divided difference matrices A_i .

4. Then the $dn \times dn$ linear pencil $L(\lambda) = \mathbf{A} - \lambda \mathbf{B}$ is a strong linearization of $P(\lambda)$, where

$$\mathbf{A} = \begin{bmatrix} A_0 & A_1 & \cdots & A_{d-2} & \tilde{A}_{d-1} \\ \sigma_0 I & I & & & \\ & \ddots & \ddots & & \\ & & \ddots & I & \\ & & & \sigma_{d-2} I & I \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 & 0 & \cdots & 0 & -A_d \\ I & 0 & & & \\ & \ddots & \ddots & & \\ & & \ddots & 0 & \\ & & & I & 0 \end{bmatrix}$$

and $\tilde{A}_{d-1} = A_{d-1} - \sigma_{d-1} A_d$

5. (a) If the pair (λ_*, x) is an eigenpair of $P(\lambda)$, then the pair $(\lambda_*, \tilde{n}(\lambda_*) \otimes x)$ is an

eigenpair of $L(\lambda)$, where $\tilde{n}(\lambda) = \begin{bmatrix} n_0(\lambda) \\ \vdots \\ n_{n-1}(\lambda) \end{bmatrix}$.

- (b) If the pair (λ_*, \mathbf{x}) is an eigenpair of $L(\lambda)$, then there exists a vector x such that $\mathbf{x} = \tilde{n}(\lambda_*) \otimes x$ and the pair (λ_*, x) is an eigenpair of $P(\lambda)$.

6. Dynamic polynomial interpolation

7. Reference

- ▶ A. Amiraslani, R. M. Corless and P. Lancaster, Linearization of matrix polynomials expressed in polynomial bases, IMA J. Numer. Anal. 29(1), pp.141-157, 2009

Part 2.4(f) (Rational) Padé approximation and linearization

1. Consider the NEP given by

$$T(\lambda)x = \left[K - \lambda M + \sum_{j=1}^q f_j(\lambda) C_j \right] x = 0 \quad (14)$$

where $K, M, C_j \in \mathbb{C}^{n \times n}$ for all j and each $f_j : \mathbb{C} \rightarrow \mathbb{C}$ is assumed to be analytic. Each C_j is of rank $r_j \ll n$ with rank-revealing factorizations $C_j = E_j F_j^T$ and $E_j, F_j \in \mathbb{C}^{n \times r_j}$.

2. Assume that the target eigenvalues of $T(\lambda)$ lie near the origin. Since a substitution allows us to shift the origin to any point in \mathbb{C} , there is no loss of generality.

3. Given a function $f(z)$, the order $[m, n]$ -Padé approximant of $f(z)$:

$$r_n^m(z) \equiv \frac{p_m(z)}{q_n(z)} = \frac{a_0 + a_1z + \cdots + a_mz^m}{1 + b_1z + \cdots + b_nz^n}$$

whose Taylor series agrees with that of f to the highest possible order, namely,

$$f(z) - r_n^m(z) = O(z^{m+n+1})$$

4. For simplicity, we focus on the case $m = n$, namely “diagonal Padé”, and denote $r_m(z) = r_n^m(z)$.
5. The order- (m, m) Padé approximant can be written in the matrix-vector form

$$r_m(z) = -a_m^T(I_m - zD_m)^{-1}b_m + d_m \quad (15)$$

where $a_m, b_m \in \mathbb{R}^m$, $D_m \in \mathbb{R}^{m \times m}$, and $d_m \in \mathbb{R}$. It is called a *minimal realization* in control theory.

6. References

- ▶ G. A. Baker, Jr. and P. R. Graves-Morris, Padé Approximants, 2nd ed., Cambridge University Press, Cambridge, UK, 1996
- ▶ P. Gonnet, S. Güttel and L. N. Trefethen, Robust Padé approximation via SVD, SIAM Review, 55(5), pp.101-117, 2013

7. Example. The order- (m, m) Padé approximation of $\sqrt{z+1}$

- In the realization form,

$$r_m(z) = -a_m^T (I_m - zD_m)^{-1} a_m + d_m \quad (16)$$

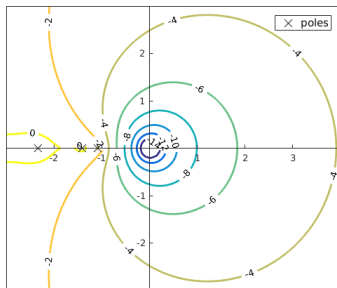
where

$$a_m = [(\gamma_1/\xi_1)^{1/2}, (\gamma_2/\xi_2)^{1/2}, \dots, (\gamma_m/\xi_m)^{1/2},$$

$$D_m = -\text{diag}(\xi_1, \xi_2, \dots, \xi_m),$$

and $d_m = 2m + 1$, and $\gamma_j = \frac{2}{2m+1} \sin^2 \frac{j\pi}{2m+1}$ and $\xi_j = \cos^2 \frac{j\pi}{2m+1}$.

- The poles are the values $-1/\xi_j$ for $j = 1, \dots, m$.
- Contour plot of $\log_{10} |e(z)|$ with $e(z) = \sqrt{z+1} - r_5(z)$. Note that two poles approximately -5.7948 and -49.3742 are outside of the range of the image.



8. Example. The order- (m, m) Padé approximation of $\exp(z)$

- ▶ The order- (m, m) Padé approximation of $\exp(z)$:

$$h_m(z) = \frac{\gamma_0 + \cdots + \gamma_{m-1}z^{m-1} + \gamma_m z^m}{\xi_0 + \cdots + \xi_{m-1}z^{m-1} + \xi_m z^m} \quad (17)$$

where for $j = 1, \dots, m$,

$$\gamma_j = \frac{(2m-j)!m!}{(2m)!j!(m-j)!}, \quad \xi_j = \frac{(-1)^j(2m-j)!m!}{(2m)!j!(m-j)!}$$

- ▶ To write in the realization form, we first rewrite (17) as

$$\begin{aligned} h_m(z) &= \frac{(\gamma_0/\xi_m) + \cdots + (\gamma_{m-1}/\xi_m)z^{m-1} + (\gamma_m/\xi_m)z^m}{(\xi_0/\xi_m) + \cdots + (\xi_{m-1}/\xi_m)z^{m-1} + z^m} \\ &\equiv \frac{a_0 + \cdots + a_{m-1}z^{m-1}}{b_0 + \cdots + b_{m-1}z^{m-1} + z^m} + d_m \end{aligned}$$

where the coefficients $a_j = (\gamma_j - (-1)^m \xi_j)/\xi_m$, $b_j = \xi_j/\xi_m$ and $d_m = \gamma_m/\xi_m = (-1)^m$ are obtained through polynomial long division.

8. Example. The order- (m, m) Padé approximation of $\exp(z)$, cont'd

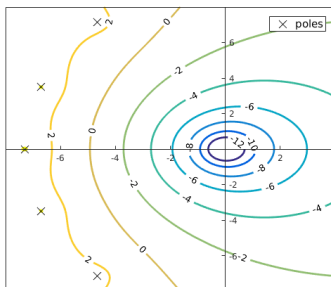
- ▶ In the realization form,

$$h_m(z) = -u_m^T (I_m - zD_m)^{-1} v_m + d_m$$

where

$$u_m = \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{m-1} \end{bmatrix}, \quad D_m = \begin{bmatrix} -b_1/b_0 & \cdots & -b_{m-1}/b_0 & -1/b_0 \\ 1 & 0 & & \\ & \ddots & \ddots & \\ & & 1 & 0 \end{bmatrix}, \quad v_m = \begin{bmatrix} -1/b_0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

- ▶ The contour plot of $\log_{10} |e(z)|$ where $e(z) = \exp(z) - r_5(z)$.



9. Applying the Padé approximants (15) to the functions $f_j(\lambda)$, we have that

$$T(\lambda) = K - \lambda M + \sum_{j=1}^q (r_{m_j}(\lambda) + e_j(\lambda)) C_j$$

Truncating the errors $e_j(\lambda)$ yields the REP

$$R(\lambda)x = \left[K - \lambda M + \sum_{j=1}^q r_{m_j}(\lambda) C_j \right] x = 0. \quad (18)$$

10. By the minimal realization of the Padé approximants, we now show that the REP can be written in a compact form.

- ▶ Each rational term of the REP (18) can be written

$$\begin{aligned}
 r_{m_j}(\lambda)C_j &= -a_{m_j}^T(I_{m_j} - \lambda D_{m_j})^{-1}b_j \cdot C_j + d_{m_j}C_j \\
 &= -E_j(I_{r_j} \cdot a_{m_j}^T(I_{m_j} - \lambda D_{m_j})^{-1}b_{m_j})F_j^T + d_{m_j}C_j \\
 &= -E_j(I_{r_j} \otimes a_{m_j}^T)(I_{r_j} \otimes I_{m_j} - \lambda I_{r_j} \otimes D_{m_j})^{-1}(I_{r_j} \otimes b_{m_j})F_j^T + d_{m_j}C_j.
 \end{aligned}$$

- ▶ Define

$$\begin{aligned}
 E &= \left[E_1(I_{r_1} \otimes a_{m_1}^T) \quad E_2(I_{r_2} \otimes a_{m_2}^T) \quad \dots \quad E_q(I_{r_q} \otimes a_{m_q}^T) \right] \\
 I_p &= \text{diag}(I_{r_1} \otimes I_{m_1}, I_{r_2} \otimes I_{m_2}, \dots, I_{r_q} \otimes I_{m_q}) \\
 D &= \text{diag}(I_{r_1} \otimes D_{m_1}, I_{r_2} \otimes D_{m_2}, \dots, I_{r_q} \otimes D_{m_q}) \\
 F &= \left[F_1(I_{r_1} \otimes b_{m_1}^T) \quad F_2(I_{r_2} \otimes b_{m_2}^T) \quad \dots \quad F_q(I_{r_q} \otimes b_{m_q}^T) \right],
 \end{aligned}$$

- ▶ The REP (18) can be written in the compact form

$$R(\lambda)x \equiv \left[\widehat{K} - \lambda M - E(I_p - \lambda D)^{-1}F^T \right] x = 0 \quad (19)$$

where $\widehat{K} = K + \sum_{j=1}^q d_{m_j}C_j$, and $p = r_1m_1 + r_2m_2 + \dots + r_qm_q$.

11. Applying the trimmed linearization technique discussed in Part 2.3(b), the REP (18) is converted to the following LEP of dimension $N = n + p$:

$$L(\lambda)\mathbf{v} \equiv (\mathbf{A} - \lambda\mathbf{B})\mathbf{v} = 0 \quad (20)$$

where

$$\mathbf{A} = \begin{bmatrix} \widehat{K} & E \\ F^T & I_p \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} M & \\ & D \end{bmatrix}, \quad \text{and} \quad \mathbf{v} = \begin{bmatrix} x \\ -(I_p - \lambda D)^{-1} F^T x \end{bmatrix}.$$

12. Theorem.

- (a) If λ is an eigenvalue of the REP (18), then it is also an eigenvalue of the LEP (20).
- (b) If (λ, \mathbf{v}) is an eigenpair of of the LEP (20), λ is not a pole of the REP (18), and $\mathbf{v}(1 : n) \neq 0$, then $(\lambda, \mathbf{v}(1 : n))$ is an eigenpair of the REP (18).

13. To solve the LEP, it is necessary to provide a method for computing the product

$$v = \mathbf{A}^{-1} \mathbf{B}u.$$

First note that we can write

$$\mathbf{A} = \begin{bmatrix} I_n & E \\ & I_p \end{bmatrix} \begin{bmatrix} \widehat{K} - EF^T & \\ & I_p \end{bmatrix} \begin{bmatrix} I_n & \\ F^T & I_p \end{bmatrix}$$

Consequently, if we let $v = [v_1^T \ v_2^T]^T$ and $u = [u_1^T \ u_2^T]^T$ then

$$v_1 = R(0)^{-1}(Mu_1 - EDu_2)$$

$$v_2 = Du_2 - F^T R(0)^{-1}(Mu_1 - EDu_2) = Du_2 - F^T v_1.$$

where

$$R(0) = \widehat{K} - EF^T.$$

14. PAL algorithm for the genuinely NEP (14)

- 1: initialize the number n_{eig} of desired eigenpairs and orders m_j of the Padé approximants of $f_j(\lambda)$
- 2: form the LEP (20)
- 3: compute the LU factorization of $R(0)$
- 4: compute the n_{eig} smallest (in modulus) eigenpairs (λ, \mathbf{v}) of the LEP.
- 5: remove any values λ which fall near the poles of any r_{m_j}
- 6: compute the approximate eigenpairs $(\lambda, \mathbf{x}) = (\lambda, \mathbf{v}(1:n))$ of the NEP (14) and the relative residual errors.

15. Case study I.

- ▶ SLAC NEP: Maxwell's equations with nonlinear waveguide boundary conditions for cavity design of a linear accelerator.
- ▶ NEP is of the form

$$T(\lambda)v \equiv \left(K - \lambda M + i \sum_{j=1}^p \sqrt{\lambda - \sigma_j^2} W_j \right) x = 0 \quad (21)$$

where $K, M, W_j \in \mathbb{C}^{n \times n}$ are constant matrices, $i = \sqrt{-1}$, and $\sigma_j \in \mathbb{R}$. Furthermore each W_j is assumed to be low rank with $W_j = E_j F_j^T$ where $E_j, F_j \in \mathbb{C}^{n \times l_j}$ have rank $l_j \ll n$.

- ▶ Suppose we wish to find eigenvalues of (21) near the shift α . Then, letting $\mu = \lambda - \alpha$ yields the shifted NEP

$$\hat{T}(\mu)x \equiv \left(K_\alpha - \mu M + i \sum_{j=1}^p \beta_j^{1/2} \sqrt{\mu/\beta_j + 1} W_j \right) x = 0 \quad (22)$$

where $K_\alpha = K - \alpha M$ and $\beta_j = \alpha - \sigma_j^2$.

- ▶ Numerical results: The 'gun' problem takes the form

$$T(\lambda) \equiv K - \lambda M + i \sqrt{\lambda - \sigma_1^2} W_1 + i \sqrt{\lambda - \sigma_2^2} W_2$$

where $\sigma_1 = 0$, $\sigma_2 = 108.8774$, $\text{rank}(W_1) = 19$, $\text{rank}(W_2) = 65$, and $n = 9,956$. The target set Σ is the upper half disk centered at 250^2 with radius $300^2 - 200^2$.

15. Case study I, cont'd

- ▶ Ref. B.-S. Liao, Z. Bai, L.-Q. Lee and K. Ko, Nonlinear Rayleigh-Ritz iterative method for solving large scale nonlinear eigenvalue problems, Taiwanese J. Math., Vol.14, No.3A, pp.869-883, 2010 and the data is available in the NLEVP collection.
- ▶ We implemented the PAL algorithm with shift $\alpha = 250^2$, and Padé orders $m_1 = 8$ and $m_2 = 10$ yielding an LEP size of $N = 9,956 + 152 + 650 = 10,758$. For comparison, we use NLEIGS (to be presented in Part 2.4(h)) on the same data set with target set Σ .
- ▶ MATLAB script `pal_slac` by J. Johnson
- ▶ The following table presents eigenvalues and residual norms found by the `pal_slac` and NLEIGS:

No.	$\text{Re}(\sqrt{\lambda})$	$\text{Im}(\sqrt{\lambda})$	PAL Res	NLEIGS Res	$ \lambda - \alpha $
1	1.494828×10^2	2.157434×10^{-3}	9.64×10^{-15}	6.31×10^{-17}	4.02×10^4
2	2.094221×10^2	4.900518×10^{-2}	4.51×10^{-16}	1.94×10^{-16}	1.86×10^4
3	2.103792×10^2	8.498907×10^{-3}	1.12×10^{-15}	1.27×10^{-16}	1.82×10^4
4	2.194130×10^2	9.546291×10^{-2}	2.34×10^{-15}	1.76×10^{-16}	1.44×10^4
5	2.208817×10^2	1.431522×10^{-2}	1.43×10^{-15}	6.67×10^{-17}	1.37×10^4
6	2.335618×10^2	9.837165×10^{-1}	7.72×10^{-16}	5.29×10^{-17}	7.96×10^3
7	2.747434×10^2	9.005400×10^0	1.36×10^{-15}	8.34×10^{-16}	1.38×10^4

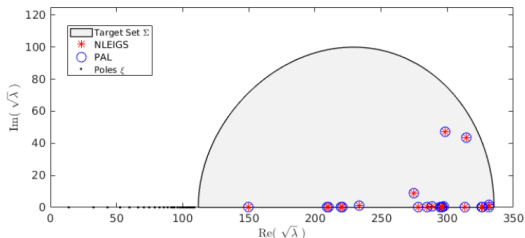
where the residuals are computed via

$$\text{Res}(\lambda, x) = \frac{\|T(\lambda)x\|_2 / \|x\|_2}{\|K\|_1 + |\lambda| \|M\|_1 + \sqrt{|\lambda - \sigma_1^2|} \|W_1\|_1 + \sqrt{|\lambda - \sigma_2^2|} \|W_2\|_1}$$

Note: software NLEIGS is to be discussed in Part 2.4(h).

15. Case study I, cont'd

- ▶ Region of interest and computed eigenvalues



- ▶ CPU timing

PAL Timing

Low Rank Decomp.	LU Decomp	eigs	Total
0.00339 s	0.31411 s	1.4391 s	1.8320

	PAL	NLEIGS			
		Variant P	Variant R_1	Variant R_2	Variant S
Total Time	1.83 s	6.62 s	6.90 s	20.01 s	5.51 s
Conv. Eigs	21	18	21	21	21

16. Case study II.

- ▶ Consider the delay differential equation

$$\frac{\partial v(x, t)}{\partial t} = \frac{\partial^2 v(x, t)}{\partial x^2} + a_0(x)v + a_1(x)v(\pi - x, t - \tau)$$

with $a_0(x) = -2 \sin(x)$, $a_1(x) = 2 \sin(x)$, and $v_x(0, t) = v_x(\pi, t) = 0$.

- ▶ After finite-difference discretizing, it yields the following so-called delay nonlinear eigenvalue problem:

$$T(\lambda)x \equiv \left(A_0 - \lambda I + A_1 e^{-\tau\lambda} \right) x = 0 \quad (23)$$

where $A_0, A_1 \in \mathbb{C}^{n \times n}$, I is the $n \times n$ identity, and $\tau \in \mathbb{R}$. The matrix A_1 is low rank with rank-revealing decomposition $A_1 = E_1 F_1^T$ and $E_1, F_1 \in \mathbb{C}^{n \times r}$.

- ▶ E. Jarlebring, K. Meerbergen, and W. Michiels, A Krylov method for the delay eigenvalue problem, SIAM J. Sci. Comput., 32, pp.3278–3300, 2010.
- ▶ To compute eigenvalues of the NEP (23) near the shift α , we set $\mu = \lambda - \alpha$ and convert (23) to the shifted NEP

$$\widehat{T}(\mu)x \equiv \left((A_0 - \alpha I) - \mu I + \widehat{A}_1 e^{-\tau\mu} \right) x = 0 \quad (24)$$

where $\widehat{A}_1 = A_1 e^{-\tau\alpha}$.

- ▶ Use the Padé approximation of $\exp(z)$ of order- $[m, m]$, the NEP (24) is approximated by the REP

$$R(\mu)x \equiv \left((A_0 - \alpha I) - \mu I + \widehat{A}_1 h_m(-\tau\mu) \right) x = 0 \quad (25)$$

16. Case study II, cont'd

- Observe that we may write

$$\begin{aligned}
 \widehat{A}_1 h_m(-\tau\mu) &= \widehat{A}_1 \cdot (-u_m^T (I_m - (-\tau\mu)D_m)^{-1} v_m + d_m) \\
 &= -E_1 \cdot \left(u_m^T (I_m - \mu D_m^\tau)^{-1} v_m \right) \cdot F_1^T + d_m \widehat{A}_1 \\
 &= -E_1 (I_r \otimes u_m^T) (I_r \otimes I_m - \mu I_r \otimes D_m^\tau)^{-1} (I_r \otimes v_m) F_1^T + d_m \widehat{A}_1 \\
 &= -E (I_P - \mu D)^{-1} F^T + d_m \widehat{A}_1
 \end{aligned}$$

where $D_m^\tau = -\tau D_m$, $E = E_1 (I_r \otimes u_m^T)$, $I_P = I_r \otimes I_m$, $D = I_r \otimes D_m^\tau$, and $F = (I_r \otimes v_m^T) F_1$.

- The REP (25) can be rewritten in the form

$$R(\mu)x \equiv \left(\widehat{A}_0 - \mu I - E (I_P - \mu D)^{-1} F^T \right) x = 0 \quad (26)$$

where $\widehat{A}_0 = A_0 - \alpha I + d_m \widehat{A}_1$.

- By the trimmed linearization technique, the REP (26) leads to the LEP

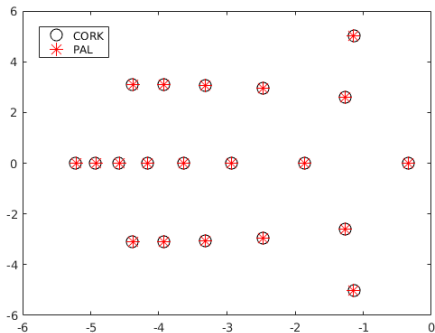
$$L(\mu)\mathbf{v} \equiv (\mathbf{A} - \mu\mathbf{B})\mathbf{v} = 0 \quad (27)$$

with

$$\mathbf{A} = \begin{bmatrix} \widehat{A}_0 & E \\ F^T & I_P \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} I & \\ & D \end{bmatrix}, \quad \mathbf{v} = \begin{bmatrix} x \\ -(I_P - \mu D)^{-1} F^T x \end{bmatrix}.$$

16. Case study II, cont'd

- ▶ Numerical results: $n = 5000$, the shift $\alpha = 0$ and Padé degree $m = 5$.
- ▶ MATLAB script `pal_delay` by J. Johnson
- ▶ The following plot depicts the 20 eigenvalues returned upon completion of `pal_delay` and CORK (to be presented in [Part 2.4\(h\)](#)):



16. Case study II, cont'd

- ▶ Residual errors:

$$\text{Res}(\lambda, x) = \frac{\|T(\lambda)x\|_2}{\|A_0\|_1 + |\lambda| + |e^{-\tau\lambda}|\|A_1\|}$$

	λ	PAL Res	CORK Res
1.	-3.3121e-01 + 0.0000e+00i	7.7129e-14	1.1979e-16
2.	-1.8663e+00 + 0.0000e+00i	2.1834e-13	3.9228e-16
3.	-1.2608e+00 + 2.5925e+00i	5.1430e-12	1.2635e-15
4.	-1.2608e+00 - 2.5925e+00i	5.1430e-12	1.2635e-15
5.	-2.9274e+00 + 0.0000e+00i	4.0764e-11	2.3917e-15
6.	-3.6387e+00 + 0.0000e+00i	9.9101e-10	5.6059e-15
7.	-2.4756e+00 + 2.9444e+00i	4.1293e-10	2.3270e-15
8.	-2.4756e+00 - 2.9444e+00i	4.1293e-10	2.3270e-15
9.	-4.1658e+00 + 0.0000e+00i	8.0378e-09	9.6437e-15
10.	-3.3174e+00 + 3.0500e+00i	5.9206e-09	4.1271e-15
11.	-3.3174e+00 - 3.0500e+00i	5.9206e-09	4.1271e-15
12.	-4.5831e+00 + 0.0000e+00i	3.7308e-08	1.3577e-14
13.	-4.9282e+00 + 0.0000e+00i	1.2388e-07	2.4985e-14
14.	-3.9207e+00 + 3.0894e+00i	3.6052e-08	8.5138e-15
15.	-3.9207e+00 - 3.0894e+00i	3.6052e-08	8.5138e-15
16.	-1.1371e+00 + 5.0365e+00i	2.0603e-09	1.4386e-14
17.	-1.1371e+00 - 5.0365e+00i	2.0603e-09	1.4386e-14
18.	-5.2224e+00 + 0.0000e+00i	3.2887e-07	2.6630e-13
19.	-4.3858e+00 + 3.1080e+00i	1.3792e-07	2.8538e-14
20.	-4.3858e+00 - 3.1080e+00i	1.3792e-07	2.8538e-14

Note: low-accuracy of the PAL algorithm in general.

- ▶ CPU timing

pal_delay	4.4456 s
CORK	22.6467 s

Part 2.4(g) Rational interpolation and linearization

1. Rational interpolants exhibit much faster convergence than polynomials
2. Let the interpolation nodes be $\sigma_0, \sigma_1, \dots, \sigma_d$ and nonzero poles be $\xi_1, \xi_1, \dots, \xi_d$, we define *rational basis functions*:

$$b_0(\lambda) = \frac{1}{\beta_0}, \quad b_i(\lambda) = \frac{\lambda - \sigma_{i-1}}{\beta_i(1 - \lambda/\xi_i)} b_{i-1}(\lambda), \quad i = 1, 2, \dots, d,$$

where β_0, \dots, β_d are nonzero scaling parameters.

3. Then the rational matrix function

$$Q(\lambda) = \sum_{i=0}^d D_i b_i(\lambda), \quad D_i \in \mathbb{C}^{n \times n}$$

interpolates $T(\lambda)$ in the interpolation nodes $\{\sigma_i\}$.

4. Computation of the rational divided-difference matrices D_i by a generalization of Opitz's method.
5. Remarks:
 - (a) Assume that the poles are all distinct from the nodes.
 - (b) If all ξ_i are at infinity, $b_i(\lambda)$ reduces to the Newton basis functions $n_i(\lambda)$.
 - (c) If $T(\lambda)$ is a rational of type $[d, d]$ with poles $\{\xi_i\}$, we have the $Q(\lambda) \equiv T(\lambda)$ for all λ .

6. The REP $Q(\lambda)x = 0$ can be linearized to the LEP

$$L(\lambda)\mathbf{x} = (\mathbf{A} - \lambda\mathbf{B})\mathbf{x},$$

where

$$\mathbf{A} = \begin{bmatrix} D_0 & D_1 & \cdots & D_{d-2} & \widetilde{D}_{d-1} \\ \sigma_0 I & \beta_1 I & & & \\ & \ddots & \ddots & & \\ & & & \beta_{d-2} I & \\ & & & \sigma_{d-2} I & \beta_{d-1} I \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \frac{D_0}{\xi_d} & \frac{D_1}{\xi_d} & \cdots & \frac{D_{d-2}}{\xi_d} & B_d \\ I & \frac{\beta_1}{\xi_1} I & & & \\ & \ddots & \ddots & & \\ & & & \frac{\beta_{d-2}}{\xi_{d-2}} I & \\ & & & I & \frac{\beta_{d-1}}{\xi_{d-1}} I \end{bmatrix},$$

and

$$\mathbf{x} = \begin{bmatrix} b_0(\lambda)x \\ b_1(\lambda)x \\ \vdots \\ b_{d-1}(\lambda)x \end{bmatrix}$$

and $\widetilde{D}_{d-1} = D_{d-1} - \sigma_{d-1}D_d/\beta_d$, $B_d = D_{d-1}\xi_d - D_d/\beta_d$.

- Note that the special role of the last pole ξ_d . For convenience, picking the last pole $\xi_d = \infty$ yields that the linearization has the same structure as in the Newton interpolation.
- Like the Newton interpolation, rational interpolation is dynamic, the rational Krylov method can be applied to a “growing” pencil.

9. Rational Krylov method

- 1: choose node σ_0 , scaling $\beta_0 = 1$ and the starting vector $v_1 \in \mathbb{C}^n$
- 2: **for** $j = 1, 2, \dots$ **do**
- 3: EXPANSION PHASE
- 4: choose σ_j , ξ_j , and β_j
- 5: compute rational divided difference D_j
- 6: expand \mathbf{A}_j , \mathbf{B}_j and V_j
- 7: RATIONAL KRYLOV PHASE
- 8: set continuation combination vector t_j
- 9: compute $\widehat{v} = (\mathbf{A}_j - \sigma_j \mathbf{B}_j)^{-1} \mathbf{B}_j V_j t_j$
- 10: orthogonalize $\widetilde{v} = \widehat{v} - V_j h_j$, where $h_j = V_j^* \widehat{v}$.
- 11: get new vector $v_{j+1} = \widetilde{v} / h_{j+1,j}$ where $h_{j+1,j} = \|\widetilde{v}\|$
- 12: compute Ritz pairs $(\lambda_i, \mathbf{x}_i)$
- 13: test for the convergence for the nonlinear eigenpair $(\lambda_i, x_i^{[1]})$
- 14: **end for**

10. Recurrence relation and Ritz pairs are generalizations of the Rational Krylov method discussed in [Part 1.4](#).

11. Choice of parameters $\{\sigma_i\}$, $\{\xi_i\}$, and $\{\beta_i\}$

12. Reference

- ▶ 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, pp. A2842–A2864, 2014

Part 2.4(h) Algorithmic framework and software

1. A unified representation of linearization pencils

$$L(\lambda) = \underbrace{\left[\frac{A_0 \ A_1 \ \cdots \ A_{d-1}}{M \otimes I_n} \right]}_{\mathbf{A}} - \lambda \underbrace{\left[\frac{B_0 \ B_1 \ \cdots \ B_{d-1}}{N \otimes I_n} \right]}_{\mathbf{B}}$$

for proper defined matrices $\{A_i\}$, $\{B_i\}$, M and N , see

- ▶ Part 2.4(c): PEP in monomial basis
 - ▶ Part 2.4(d): Lagrange interpolation in barycentric form
 - ▶ Part 2.4(e): Newton interpolation
 - ▶ Part 2.4(g): Rational interpolation
2. CORK: a rational Krylov method for the linear pencil $L(\lambda)$ with a compact representation (two-level orthogonality) of basis vectors:
 - ▶ R. Van Beeumen, K. Meerbergen and W. Michiels, Compact rational Krylov method for nonlinear eigenvalue problems, SIAM J. Matrix Anal. Appl. 36(2), pp.820-838, 2015
 - ▶ R. Van Beeumen, Rational krylov methods for nonlinear eigenvalue problems, PhD thesis, KU Leuven, 2015
 3. CORK and NLEIGS toolboxes for a class of linearization methods, include
 - ▶ polynomial or rational interpolation/approximation in different bases, such as discussed in [Parts 2.4\(c\)](#), [2.4\(d\)](#), [2.4\(e\)](#) and [2.4\(g\)](#)
 - ▶ interpolatio nodes and poles
 - ▶ shifts in the Rationak Krylov method
 - ▶ Dynamic/static/hybrid

available at <http://www.roelvanbeeumen.be>

4. Revisit MATLAB scripts `pal_slac` and `pal_delay`
5. Recent work on compact representations of basis vectors in the subspace projection methods:
 - ▶ The QEP and PEP in monomial basis – [Part 2.3\(a\)](#)
 - ▶ Y. Su, J. Zhang and Z. Bai, A compact Arnoldi algorithm for polynomial eigenvalue problems, conference presentation, 2008
 - ▶ D. Lu, Y. Su and Z. Bai, Stability analysis of two-level orthogonal Arnoldi procedure, *SIAM J. Matrix Anal. Appl.* 37(1), pp.192-214, 2016
 - ▶ The PEPs in orthogonal basis:
 - ▶ D. Kressner and J. E. Roman, Memory-efficient Arnoldi algorithms for linearizations of matrix polynomials in Chebyshev basis, *Numer. Linear Algebra Appl.*, 21, pp. 569–588, 2014.
 - ▶ C. Campos and J. E. Roman, Parallel Krylov solvers for the polynomial eigenvalue problem in SLEPc, *SIAM J. Sci. Comput.* 38(5), pp.S385-S411, 2016
 - ▶ The REP by trimmed linearization – [Part 2.3\(b\)](#)
 - ▶ F. M. Dopico and J. Gonzalez-Pizarro, A compact rational Krylov method for large-scale rational eigenvalue problems, arXiv:1705.06982v1, May 19, 2017
 - ▶ The waveguide eigenvalue problem
 - ▶ E. Jarlebring, G. Mele and O. Runborg, The waveguide eigenvalue problem and the tensor infinite Arnoldi method, arXiv:1503.02096v2, Mar. 20, 2015

Part 2.5 Of things not treated

1. Matrix polynomial interpolation in orthogonal bases and linearization
 - ▶ C. Effenberger, D. Kressner, O. Steinbach, and G. Unger. Interpolation-based solution of a nonlinear eigenvalue problem in fluid-structure interaction. In Proceedings in Applied Mathematics and Mechanics, volume 12, pp. 633 - 634, 2012.
 - ▶ C. Effenberger, Robust solution methods for nonlinear eigenvalue problems, PhD thesis, EPFL, 2013
2. Infinite Arnoldi (IAR), TensorIAR and Infinite Bi-Lanczos methods
 - ▶ E. Jarlebring, W. Michiels, K. Meerbergen, A linear eigenvalue algorithm for the nonlinear eigenvalue problem, Numer. Math. 122 (1), pp.169-195, 2012.
 - ▶ E. Jarlebring, G. Mele and O. Runborg, The waveguide eigenvalue problem and the tensor infinite Arnoldi method, arXiv:1503.02096v2, Mar. 20, 2015
 - ▶ S. W. Gaff and E. Jarlebring, The infinite bi-Lanczos method for nonlinear eigenvalue problems, arXiv:1607.03454v1, July 12, 2016
3. Methods based on contour integrals
 - ▶ J. Asakura, T. Sakurai, H. Tadano, T. Ikegami, K. Kimura, A numerical method for nonlinear eigenvalue problems using contour integrals, JSIAM Lett. 1, pp.52-55, 2009.
 - ▶ W.-J. Beyn, An integral method for solving nonlinear eigenvalue problems, Lin.Alg.Appl. 436, pp.3839-3863, 2012
 - ▶ M. Van Barel and P. Kravanja, Nonlinear eigenvalue problems and contour integrals, J. Comp. Appl. Math. 292, pp.526-540, 2016
 - ▶ J. Xiao, C. Zhang, T.-M. Huang and T. Sakurai, Solving large-scale nonlinear eigenvalue problems by rational interpolation approach and resolvent sampling based Rayleigh-Ritz method, arXiv:1605.07951v1, May 25, 2016

Part 3: Eigenvalue problems with eigenvector nonlinearity

Outline of Part 3

1. Kohn-Sham density functional theory
2. Sum of trace ratio
3. Robust Rayleigh quotient optimization
4. Of things not treated

Part 3.1 KS DFT

1. The discretized Kohn-Sham eigenvalue problem with eigenvector nonlinearity:

$$\begin{aligned}H(X)X &= X\Lambda \\ X^T X &= I,\end{aligned}\tag{28}$$

where $X \in \mathbb{R}^{n \times k}$, $H(X) \in \mathbb{R}^{n \times n}$ is a symmetric matrix function of X , $\Lambda \in \mathbb{R}^{k \times k}$ is a diagonal matrix consisting of k smallest eigenvalues of $H(X)$.

2. It is often (assumed) that

$$H(X) = H(XQ) \quad \text{for any } k \times k \text{ orthogonal matrix } Q.$$

This implies that $H(X)$ is a matrix function of k -dimensional subspaces of $\mathbb{R}^{n \times n}$. Namely, if X is a solution, then so is XQ for any $k \times k$ orthogonal matrix Q . The solution to (28) is **unique** in terms of that for two solutions X_1, X_2 , $\text{span}\{X_1\} = \text{span}\{X_2\}$, or equivalently, $X_1 X_1^T = X_2 X_2^T$.

3. $H(X)$ is a discretized Hamiltonian from the the density functional theory (DFT) for electronic structure calculations.

- ▶ A simple single-particle Hamiltonian model

$$H(X) = L + \alpha \text{Diag}(L^{-1}\rho(X)) \quad (29)$$

where L is a discrete Laplacian, and $\rho(X) = \text{diag}(XX^T)$.⁶

- ▶ C. Yang, W. Gao and J. Meza, On convergence of the self-consistent field iteration for a class of nonlinear eigenvalue problems, *SIAM. J. Matrix Anal. Appl.*, 30, pp.1773-1788, 2009.
- ▶ An extended model

$$H(X) = \frac{1}{2}L + V_{\text{ion}} + \sum_{\ell} w_{\ell}w_{\ell}^T + \text{Diag}(L^{\dagger}\rho(X)) + \text{Diag}(\mu_{\text{xc}}(\rho(X))^T e) \quad (30)$$

see, for examples, and references therein

- ▶ C. Yang, J. Meza and L. Wang, A trust-region direct constrained minimization algorithm for the Kohn-Sham equation, *SIAM J. Sci. Comput.*, 29, pp.1854-1875, 2007.
- ▶ C. Yang, J. Meza, B. Lee and L.-W. Wang, KSSOLV – A MATLAB toolbox for solving the Kohn-Sham equations, *ACM Trans. Math. Software*, 46, pp.1-35, 2009
- ▶ X. Liu, X. Wang, Z. Wen and Y. Yuan, On the convergence of the self-consistent field iteration in Kohn-Sham density functional theory, *SIAM J. Matrix Anal. Appl.*, 35, pp.546-558, 2014.
- ▶ Models in physics and chemistry literature, see for example,
 - ▶ R. M. Martin, *Electronic Structure: Basic Theory and Practical Methods*, Cambridge University Press, Cambridge, 2004.
- ▶ Survey articles, for example, see
 - ▶ Y. Saad, J. R. Chelikowsky, and S. M. Shontz. Numerical methods for electronic structure calculations of materials. *SIAM Rev.*, 52(1):3–54, 2010.
- ▶ Part III of Professor N. Sukumar's lecture at this summer school.

⁶Diag(x) denotes the diagonal matrix with the vector x on its diagonal. $\text{diag}(A)$ denotes the vector containing the diagonal elements of the matrix A .

4. Optimization point of view

- ▶ The model (29) is equivalent to the constrained minimization problem

$$\begin{aligned} \min \quad & E(X) := \frac{1}{2} \text{tr}(X^T L X) + \frac{\alpha}{4} \rho(X)^T L^{-1} \rho(X) \\ \text{s.t.} \quad & X^T X = I_k \end{aligned} \tag{31}$$

in the sense that the solution of the eigenvalue problem (28) is a global minimizer. (A *nontrivial exercise to verify the statement!*)

- ▶ The extended model (30) has also be shown to be equivalent to a constrained minimization problem of the form (31), see
 - ▶ X. Liu, X. Wang, Z. Wen and Y. Yuan, On the convergence of the self-consistent field iteration in Kohn-Sham density functional theory, *SIAM J. Matrix Anal. Appl.*, 35, pp.546-558, 2014.
- ▶ Solvers for the constrained minimization problem (31):
 - ▶ M. C. Payne, M. P. Teter, D. C. Allen, T. A. Arias, and J. D. Joannopoulos, Iterative minimization techniques for ab initio total energy calculation: Molecular dynamics and conjugate gradients, *Rev. Modern Phys.*, 64 (1992), pp. 1045-1097
 - ▶ ...
 - ▶ X. Zhang, J. Zhu, Z. Wen and A. Zhou, Gradient type optimization methods for electronic structure calculations. *SIAM J. Sci. Comput.* 36, C265–C289, 2014.
 - ▶ Z. Zhao, Z.-J. Bai and X. Jin, A Riemannian Newton algorithm for nonlinear eigenvalue problems, *SIAM J. Matrix Anal. Appl.*, 36, pp.752-774, 2015.
 - ▶ X. Dai, Z. Liu, L. Zhang and A. Zhou, A conjugate gradient method for electronic structure calculations, arXiv:1601.07676v4, Jan. 13, 2017

5. Additional notations

- ▶ $\mathbb{X}_k = \{X \mid X \in \mathbb{R}^{n \times k} \text{ and } X^T X = I\}$
- ▶ $\Theta(X, Y)$ denotes the principal angles between subspaces $\text{span}(X)$ and $\text{span}(Y)$.
- ▶ It is known that if $[X, X_c]$ and $[Y, Y_c]$ are two orthogonal matrices with $X, Y \in \mathbb{R}^{n \times k}$, then

$$\|\sin \Theta(X, Y)\|_2 = \|X X^T - Y Y^T\|_2 = \|X_c^T Y\|_2 = \|X^T Y_c\|_2.$$

see

- ▶ G. W. Stewart and J.-G. Sun, *Matrix Perturbation Theory*, Academic Press, 1990.
- ▶ J.-G. Sun, *Matrix Perturbation Analysis* (2nd edition), Science Press, 2001 (in Chinese).

6. Existence and uniqueness of the eigenvalue problem (28):

Assume that there exists a positive constant ξ such that

$$\|H(X_1) - H(X_2)\|_2 \leq \xi \|\sin \Theta(X_1, X_2)\|_2, \quad (32)$$

where X_1, X_2 are arbitrary matrices in \mathbb{X}_k . In addition, assume that there exists a positive constant δ_v such that

$$\lambda_{k+1}(H(X)) - \lambda_k(H(X)) \geq \delta_v \quad \text{for any } X \in \mathbb{X}_k. \quad (33)$$

Then if $\delta_v > 2\xi$, the eigenvalue problem (28) has a unique solution.

Remarks:

- ▶ The condition is a Lipschitz-like condition.
- ▶ The condition (33) is known as “uniformly well posed” in DFT calculations

7. A plain SCF (Self-Consistent-Field) iteration for solving the eigenvalue problem (28)

- 1: choose initial guess X^0
- 2: **for** $j = 1, 2, \dots$ until convergence **do**
- 3: construct $H^i = H(X^{i-1})$
- 4: compute the partial eigenpairs (Λ^i, X^i) the LEP X^i such that $H^i X^i = X^i \Lambda^i$, where Λ^i contains the k smallest eigenvalues of H^i
- 5: **end for**

8. Example.

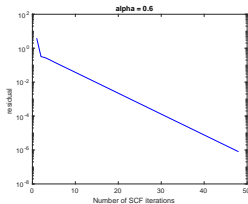
- ▶ Consider the simple model problem (29):

$$H(X) = L + \alpha \text{Diag}(L^{-1}\rho(X))$$

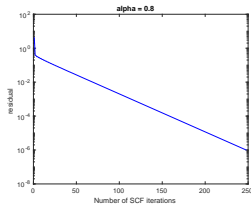
where L is a 1-D Laplacian, i.e.,

$$L = \begin{bmatrix} 2 & -1 & & & & & \\ -1 & 2 & -1 & & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & -1 & 2 & -1 & \\ & & & & -1 & 2 & \\ & & & & & & 2 \end{bmatrix}$$

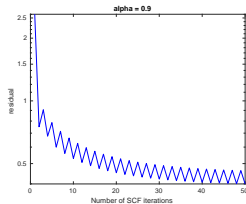
- ▶ Let $n = 10$ and $k = 2$, convergence behavior of the SCF for different values of α :



fast conv.



slow conv.



non-conv.

- ▶ Open question: What is the optimal bound for α for the convergence of the SCF iteration? (see Table 1 in [Yang-Gao-Meza'09])

9. Studies of the convergence of SCF iterations, see for examples, .
- ▶ E. Cancès and C. L. Bris, On the convergence of SCF algorithms for the HartreeFock equations, *Math. Model. Numer. Anal.*, 34, pp.749-774, 2000.
 - ▶ C. Yang, W. Gao, and J. Meza, On the convergence of the self-consistent field iteration for a class of nonlinear eigenvalue problems, *SIAM J. Matrix Anal. Appl.*, 30, pp. 1773-1788, 2009.
 - ▶ X. Liu, X. Wang, Z. Wen and Y. Yuan, On the convergence of the self-consistent field iteration in Kohn-Sham density functional theory, *SIAM J. Matrix Anal. Appl.*, 35, pp.546-558, 2014.
10. In this lecture, we just focus on the results of convergence analysis of the plain SCF for the algebraic eigenvalue problem (28), recently driven by [Yunfeng Cai](#), [Ren-Cang Li](#) and [B.](#)

11. Global convergence of the plain SCF

Let the Lipschitz-like condition (6) hold. In addition, assume that there exists a positive constant δ such that

$$\lambda_{k+1}^i - \lambda_k^i \geq \delta \quad \text{for all } i = 1, 2, \dots \quad (34)$$

Denote $\theta_i = \|\sin \Theta(X^{i-1}, X^i)\|_2$. Then if

$$\delta > (1 + \theta_1)\xi,$$

then we have

$$\theta_{i+1} \leq \tau \theta_i \quad \text{with} \quad \tau = \frac{\xi}{\delta - \xi \theta_1} < 1.$$

This implies that the SCF is globally linearly convergent, i.e., $\theta_i \rightarrow 0$ as $i \rightarrow \infty$.

Remark:

- ▶ The condition (34) is a “uniformly well posed” in the SCF iteration

12. Local convergence of the SCF

Let X^* be a solution to the eigenvalue problem (28), and

$$\delta_* = \lambda_{k+1} - \lambda_k > 0, \quad (35)$$

where λ_k and λ_{k+1} are the k -th and $(k+1)$ -th eigenvalues of $H(X^*)$.

Assume that

- A1. $H(X)$ is continuous at X^* ;
- A2. There exists a constant $\chi > 0$ such that

$$\limsup_{\|\sin \Theta(X, X^*)\|_2 \rightarrow 0} \frac{\|(I - P^*)[H(X) - H(X^*)]P^*\|_2}{\|\sin \Theta(X, X^*)\|_2} \leq \chi. \quad (36)$$

where $P^* = XX^*$.

Then if $\chi < \delta_*$, and X^0 is sufficiently close to X^* (i.e., $\|\sin \Theta(X^0, X^*)\|_2$ is sufficiently small), then there exists a positive constant $\tau < 1$ such that

$$\|\sin \Theta(X^i, X^*)\|_2 \leq \tau \|\sin \Theta(X^{i-1}, X^*)\|_2 \quad \text{for } i = 1, 2, \dots$$

This implies that the SCF is locally linearly convergent.

13. Remarks on assumption A2.

- ▶ at the solution X^* ,

$$[X^*, X_c^*]^T H(X^*) [X^*, X_c^*] = \begin{bmatrix} (X^*)^T H(X^*) X^* & 0 \\ 0 & (X_c^*)^T H(X^*) X_c^* \end{bmatrix}.$$

- ▶ note that

$$\begin{aligned} \|(I - P^*)[H(X) - H(X^*)]P^*\|_2 &= \|(X_c^*)^T [H(X) - H(X^*)] X^*\|_2 \\ &= \|(X_c^*)^T H(X) X^*\|_2 \end{aligned}$$

- ▶ Therefore, the assumption A2 implies that the (2, 1) block of

$$[X^*, X_c^*]^T H(X) [X^{(*)}, X_c^{(*)}] = \begin{bmatrix} (X^*)^T H(X) X^* & (X^*)^T H(X) X_c^* \\ (X_c^*)^T H(X) X^* & (X_c^*)^T H(X) X_c^* \end{bmatrix}$$

is close to be the zero.

- ▶ The assumption A2 is weaker than the Lipschitz-like condition (6), i.e., if the Lipschitz-like condition (6) holds, one can pick $\chi = \xi$ to satisfy the assumption A2.

14. Consider the KS model (30):

$$H(X) = \frac{1}{2}L + V_{\text{ion}} + \sum_{\ell} w_{\ell} w_{\ell}^T + \text{Diag}(L^{\dagger} \rho(X)) + \text{Diag}(\mu_{\text{xc}}(\rho(X))^T e)$$

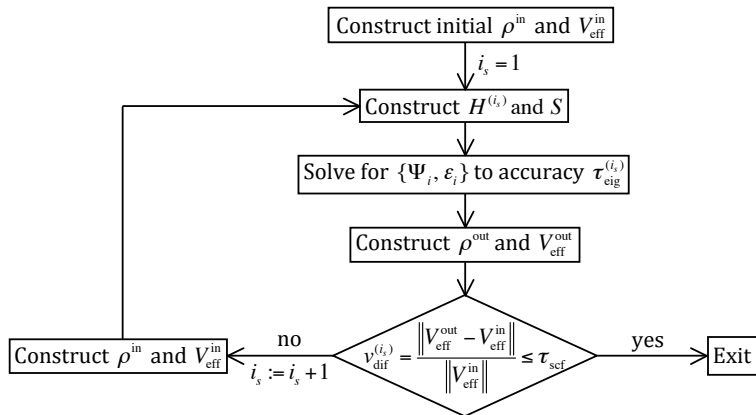
Assume that there exists a positive constant σ such that

$$\|\text{Diag}(\mu_{\text{xc}}(\rho(X_1))^T e) - \text{Diag}(\mu_{\text{xc}}(\rho(X_2))^T e)\|_{\infty} \leq \sigma \|\rho(X_1) - \rho(X_2)\|_{\infty},$$

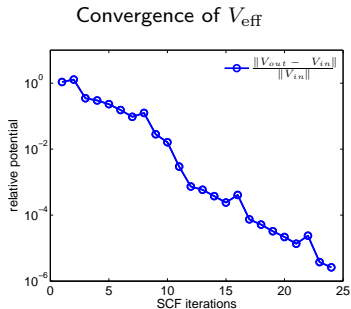
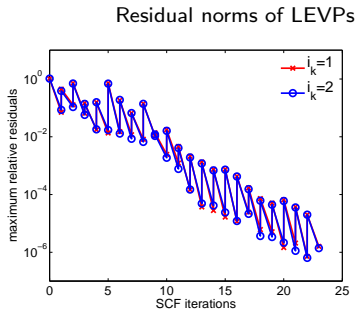
for all $X_1, X_2 \in \mathbb{X}_k$. Then by the previous convergence results, we have

- (i) If $\lambda_{k+1}^{(i)} - \lambda_k^{(i)} > (1 + \theta_1)(\|L^{\dagger}\|_1 + \sigma)$ for all i , then the plain SCF is globally linearly convergent.
- (ii) If $\lambda_{k+1} - \lambda_k > \|L^{\dagger}\|_1 + \sigma$, then the plain SCF is locally linearly convergent.

15. SCF in KS-DFT calculations



16. Iterative diagonalization within SCF



17. Many studies for iterative diagonalization, for examples, see and the references therein

- ▶ Y. Zhou, Y. Saad, M. L. Tiago and J. R. Chelikowsky, Self-consistent-field calculations using Chebyshev-filtered subspace iteration, J. Comput. Phys. 219:172-184, 2006
- ▶ Y. Cai, Z. Bai, J. Pask and N. Sukumar, Hybrid preconditioning for iterative diagonalization of ill-conditioned generalized eigenvalue problems in electronic structure calculations, J. of Comput. Phys., 255, pp.16-33, 2013
- ▶ Y. Zhou, Z. Wang and A. Zhou, Accelerating large partial EVD/SVD calculations by filtered block Davidson methods, Sci. China Math. 50(8), pp.1635-1662, 2016

Part 3.2 Sum of Trace Ratios

1. The sum of trace ratios problem:

$$\max_{\substack{V \in \mathbb{R}^{n \times \ell} \\ V^T V = I_\ell}} \sum_{i=1}^k \frac{\text{tr}(V^T A_i V)}{\text{tr}(V^T B_i V)},$$

where $A_i = A_i^T$, $B_i = B_i^T \in \mathbb{R}^{n \times n}$, $B_i > 0$ and $\ell < n$.

2. Applications:

- ▶ $k = 1$, $\ell = 1$: the eigenvalue problem for $A_1 - \lambda B_1$
- ▶ $k = 1$, $\ell > 1$: Fisher discriminant analysis in pattern recognition [Ngo, Bellalij, & Saad (2010); Zhang, Liao, & Ng (2010)]
- ▶ $k > 1$, $\ell = 1$: balancing individual capacities in a multi-user MIMO downlink channel in ratio transmission [Primolevo, Simeone, & Spagnolini (2006); Zhang (2013)].
- ▶ The general case $k > 1$ and $\ell > 1$ in Fisher-like discriminant analysis for classifying two or more sets in certain balanced way.

3. We focus on the following Sum-of-Two-Trace-Ratios (S2TR) problem

$$\max_{V^T V = I_\ell} \left\{ \frac{\text{tr}(V^T A V)}{\text{tr}(V^T B V)} + \text{tr}(V^T C V) \right\}$$

where $A = A^T$, $B = B^T$, $C = C^T \in \mathbb{R}^{n \times n}$ and $B > 0$.

4. The S2TR problem is an optimization problem on Stiefel manifold

$$\mathbb{O}^{n \times \ell} := \{V \in \mathbb{R}^{n \times \ell} : V^T V = I_\ell\}.$$

5. Existing generic Stiefel manifold-based optimization methods:

- ▶ `sg_min` by R. Lippert and A. Edelman⁷
Fletcher-Reeves CG, Polak-Ribière CG, Newton, and dog-leg Newton
- ▶ `OptM` by Z. Wen and W. Yin.⁸
Barzilai-Borwein conjugate gradient with Crank-Nicolson-like updating to preserve orthogonality constraints and curvilinear search
- ▶ `ROPTIB` by Wen Huang⁹
An object-oriented C++ library for optimization on Riemannian manifolds.

6. The method discussed here is an approximate 1st order method specially designed for the S2TR problem.

⁷Section 9.4 in [Bai, Demmel, Dongarra, Ruhe and van der Vorst (editors). *Templates for the solution of Algebraic Eigenvalue Problems: A Practical Guide*. SIAM, Philadelphia, 2000.

⁸Z. Wen and W. Yin. A feasible method for optimization with orthogonality constraints. *Math. Programming*, 142(1-2):397–434, 2013.

⁹available at http://www.math.fsu.edu/~whuang2/Indices/index_ROPTLIB.html

7. First order condition

- ▶ Let $\phi_H(V) := \text{tr}(V^T H V)$ for $V \in \mathbb{O}^{n \times \ell}$, then

$$f(V) := \frac{\text{tr}(V^T A V)}{\text{tr}(V^T B V)} + \text{tr}(V^T C V) \equiv \frac{\phi_A(V)}{\phi_B(V)} + \phi_C(V).$$

- ▶ Ignore $V^T V = I_\ell$ and use

$$f(V + \delta V) = f(V) + \left\langle \delta V, \frac{\partial f(V)}{\partial V} \right\rangle + O(\|\delta V\|^2)$$

to see

$$\frac{\partial f(V)}{\partial V} = 2 \underbrace{\left[A \frac{1}{\phi_B(V)} - B \frac{\phi_A(V)}{[\phi_B(V)]^2} + C \right]}_{=: E(V)} V,$$

where $\langle X, Y \rangle = \text{tr}(X^T Y)$.

- ▶ But it has to be projected onto the tangent space at $V \in \mathbb{O}^{n \times \ell}$ for the gradient of $f(V)$.

7. First order condition, cont'd

- ▶ Tangent space at $V \in \mathbb{O}^{n \times \ell}$:

$$\begin{aligned}\mathcal{T}_V \mathbb{O}^{n \times \ell} &:= \{X \in \mathbb{R}^{n \times \ell} : X^T V + V^T X = 0\} \\ &= \{X = VK + (I - VV^T)J : K = -K^T \in \mathbb{R}^{\ell \times \ell}, J \in \mathbb{R}^{n \times \ell}\}.\end{aligned}$$

- ▶ Orthogonal projection:

$$\Pi_{\mathcal{T}} : Z \in \mathbb{R}^{n \times \ell} \rightarrow \Pi_{\mathcal{T}}(Z) \in \mathcal{T}_V \mathbb{O}^{n \times \ell},$$

where

$$\Pi_{\mathcal{T}}(Z) := V \left(\frac{V^T Z - Z^T V}{2} \right) + (I_m - VV^T)Z = Z - V \operatorname{sym}(V^T Z).$$

and

$$\operatorname{sym}(V^T Z) = \frac{V^T Z + Z^T V}{2}.$$

7. First order condition, cont'd

- ▶ Gradient of $f(V)$:

$$\text{grad}f|_{\mathbb{O}^{n \times \ell}}(V) = \Pi_{\mathcal{T}} \left(\frac{\partial f(V)}{\partial V} \right) = 2\{E(V)V - VM(V)\}$$

where

$$M(V) = \text{sym}(V^T E(V)V) = \frac{V^T AV}{\phi_B(V)} - \frac{\phi_A(V)}{[\phi_B(V)]^2} V^T BV + V^T CV.$$

- ▶ First order optimality (KKT) condition: If $V \in \mathbb{O}^{n \times \ell}$ is a local maximizer, then

$$E(V)V = VM(V).$$

- ▶ It implies that $\text{eig}(M(V)) \subset \text{eig}(E(V))$, V is an orthonormal eigenbasis of $E(V)$ associated with its eigenvalues given by $\text{eig}(M(V))$.

8. Second order condition

- ▶ Riemannian Hessian:

$$\begin{aligned} \text{hess}f|_{\mathbb{O}^{n \times \ell}}(V) &: \mathcal{T}_V \mathbb{O}^{n \times \ell} \rightarrow \mathcal{T}_V \mathbb{O}^{n \times \ell}, \\ X &\mapsto \Pi_{\mathcal{T}}(\mathcal{D}g(V)[X]), \end{aligned}$$

where $g(V) := \text{grad}f|_{\mathbb{O}^{n \times \ell}}(V) = 2\{E(V)V - VM(V)\}$ and $\mathcal{D}g(V)[X]$ represents the classical directional derivative at $V \in \mathbb{O}^{n \times \ell}$ along X .

- ▶ Calculation leads to

$$\begin{aligned} \text{hess}f|_{\mathbb{O}^{n \times \ell}}(V)[X] &= 2\left[E(V)X - V\text{sym}(V^T E(V)X) - XV^T E(V)V \right. \\ &\quad \left. + V\text{sym}(V^T XV^T E(V)V) + G(V, X)V - VV^T G(V, X)V\right], \end{aligned}$$

where

$$G(V, X) := 4 \frac{\text{tr}(V^T AV)\text{tr}(X^T BV)}{[\text{tr}(V^T BV)]^3} B - 2 \frac{\text{tr}(X^T BV)A + \text{tr}(X^T AV)B}{[\text{tr}(V^T BV)]^2}.$$

- ▶ Given $V \in \mathbb{O}^{n \times \ell}$, a critical (aka KKT) point, the second-order optimality condition at V is about

$$\text{hess}f|_{\mathbb{O}^{n \times \ell}}(V)[X, X] = \langle \text{hess}f|_{\mathbb{O}^{n \times \ell}}(V)[X], X \rangle \quad \text{for } X \in \mathcal{T}_V \mathbb{O}^{n \times \ell}$$

which, after nontrivial simplifications, leads to

$$\langle \text{hess}f|_{\mathbb{O}^{n \times \ell}}(V)[X], X \rangle = 2\langle X, E(V)X \rangle - 2\langle X, XV^T E(V)V + G(V, X)V \rangle.$$

8. Second order condition, cont'd

- ▶ Second order optimality condition: If $V \in \mathbb{O}^{n \times \ell}$ is a local maximizer, then for any $X \in \mathcal{T}_V \mathbb{O}^{n \times \ell}$

$$\text{tr}(X^T E(V)X) - \text{tr}(XM(V)X^T) - \text{tr}(X^T G(V, X)V) \leq 0.$$

If it is a strict inequality for $X \neq 0$, then V is a strict local maximizer.

- ▶ Since any $X \in \mathcal{T}_V \mathbb{O}^{n \times \ell}$ takes the form

$$X = VK + (I_m - VV^T)J \quad \text{for } K^T = -K \text{ and arbitrary } J,$$

an (equivalent) second order optimality condition:

If $V \in \mathbb{O}^{n \times \ell}$ is a local maximizer, then for all $J \in \mathbb{R}^{n \times \ell}$

$$\begin{aligned} & \text{tr}(J^T E(V)J) + \text{tr}(V^T JM(V)J^T V) \\ & - \text{tr}(J^T VM(V)V^T J) - \text{tr}(JM(V)J^T) \\ & + 4 \frac{\text{tr}(J^T [I_m - VV^T]BV) \text{tr}(J^T [I_m - VV^T]CV)}{\phi_B(V)} \leq 0. \end{aligned}$$

If it is a strict inequality for $J \neq 0$, then V is a strict local maximizer.

- ▶ Although more complicated, this one turns out to be more useful in deriving necessary conditions.

9. Necessary condition for local maximizer

- ▶ Suppose V is a local maximizer of $E(V)V = VM(V)$ and let $\text{eig}(E(V)) = \{\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n\}$.
- ▶ $E(V)V = VM(V) \Rightarrow \text{eig}(M(V)) \subset \text{eig}(E(V))$, i.e.,

$$\text{eig}(M(V)) = \{\lambda_{\pi_i}, i = 1, 2, \dots, \ell\},$$

where $1 \leq \pi_1 < \pi_2 < \dots < \pi_\ell \leq n$.

- ▶ Necessary condition:

$$\lambda_{\pi_1} \geq \lambda_{2\ell}.$$

10. Necessary condition for global maximizer

- ▶ Without loss of generality, assume $C > 0$, because

$$\frac{\operatorname{tr}(V^T AV)}{\operatorname{tr}(V^T BV)} + \operatorname{tr}(V^T [C + \xi I] V) = \frac{\operatorname{tr}(V^T AV)}{\operatorname{tr}(V^T BV)} + \operatorname{tr}(V^T CV) + \ell \xi.$$

- ▶ Suppose V is a global maximizer of $E(V)V = VM(V)$ and let $\operatorname{eig}(E(V)) = \{\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n\}$. Then

$$\operatorname{eig}(M(V)) = \{\lambda_{\pi_i}, i = 1, 2, \dots, \ell\}.$$

- ▶ Necessary condition

$$\pi_i = i \quad \text{for } 1 \leq i \leq \ell,$$

i.e., V corresponds to the ℓ largest eigenvalues λ_i for $1 \leq i \leq \ell$.

11. Self-Consistent-Field (SCF) iteration

- ▶ The necessary condition for a global maximizer has an important numerical implication – leading to the following SCF for $E(V)V = VM(V)$:
 - 1: choose $V_0 \in \mathbb{O}^{n \times \ell}$ and a tolerance `tol`
 - 2: **for** $k = 0, 1, \dots$ **do**
 - 3: compute an orthonormal eigenbasis V_{k+1} of $E(V_k)$ associated with its ℓ largest eigenvalues;
 - 4: **if** $r_k := \|E(V_{k+1})V_{k+1} - V_{k+1}M(V_{k+1})\|_2 \leq \text{tol}$ **then**
 - 5: **break**;
 - 6: **end if**
 - 7: **end for**
 - 8: return V_{k+1} as an approximate maximizer.
- ▶ It is an approximate first-order method.

12. Remarks:

- ▶ Reminiscent of SCF for the Kohn-Sham equations in electronic structure calculations.
- ▶ If the sequence $\{V_k\}$ converges to V_* , then not only V_* is a KKT point, but also satisfies the necessary condition for a global maximizer. This is one of the major advantages of SCF over optimization-based methods which primarily concern monotonic change of the objective value. The converged KKT points may or may not satisfy the necessary condition.
- ▶ Major computational cost lies at Line 3 – finding dominant orthonormal eigenbasis of $E(V_k)$. For large scale ones, iterative methods should be used.

13. Local convergence

- ▶ Suppose $E(\hat{V})\hat{V} = \hat{V}M(\hat{V})$, i.e., \hat{V} is a stationary point and

$$\text{eig}(E(\hat{V})) = \{\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n\}, \quad \text{eig}(M(\hat{V})) = \{\lambda_i, i = 1, 2, \dots, \ell\}.$$

- ▶ Set

$$R_A := A\hat{V} - \hat{V}(\hat{V}^\top A\hat{V})$$

$$R_B := B\hat{V} - \hat{V}(\hat{V}^\top B\hat{V})$$

$$\delta := \lambda_\ell - \lambda_{\ell+1}$$

Suppose $\delta > 0$.

- ▶ If $\|\sin \Theta(V_0, \hat{V})\|_2$ is sufficiently small, then $\|\sin \Theta(V_i, \hat{V})\|_2$ goes to 0 at least linearly.
- ▶ Suppose $R_B = 0$. If $\|\sin \Theta(V_0, \hat{V})\|_2$ is sufficiently small, then $\|\sin \Theta(V_i, \hat{V})\|_2$ goes to 0 quadratically.
- ▶ Suppose $R_A = R_B = 0$. If $\|\sin \Theta(V_0, \hat{V})\|_2$ is sufficiently small, the convergence is instant, i.e., $\text{span}(V_1) = \text{span}(\hat{V})$.
- ▶ Proofs rely on complicated estimates. Linear convergence rate depends on $\|R_B\|$ and goes to 0 as R_B does.

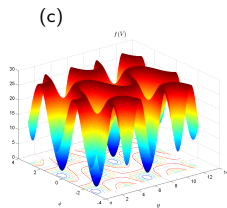
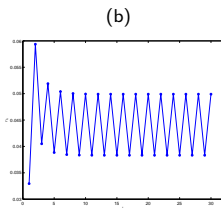
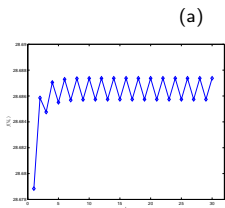
14. Example. SCF has no global convergence in general

- ▶ Consider $n = 3$, $\ell = 2$, and

$$A = \begin{bmatrix} 11 & 5 & 8 \\ 5 & 10 & 9 \\ 8 & 9 & 5 \end{bmatrix}, \quad B = \begin{bmatrix} 7 & 7 & 7 \\ 7 & 10 & 8 \\ 7 & 8 & 8 \end{bmatrix} \quad \text{and} \quad C = \begin{bmatrix} 15 & 10 & 9 \\ 10 & 7 & 6 \\ 9 & 6 & 6 \end{bmatrix}.$$

- ▶ $V_0 = [e_1, e_2]$, the first two columns of I_3 .
- ▶ $\{f(V_k)\}$ and $\{r_k\}$ generated by SCF oscillate after a few iterations. They behave similarly with random V_0 , too.
- ▶ Plots:

- (a) $f(V_k)$
- (b) $r_k = \|E(V_{k+1})V_{k+1} - V_{k+1}M(V_{k+1})\|_2$
- (c) contour and Mesh of $f(V)$



15. Numerical results for random matrices

- ▶ Test matrices

$$A = \text{randn}(n, n); \quad A = A' * A; \quad B = \text{randn}(n, n); \quad B = B' * B;$$

$$C = \text{randn}(n, n); \quad C = C' * C; \quad V_0 = \text{orth}(\text{randn}(n, \ell), 0).$$

- ▶ Note that all are positive definite but do not lose any generality because

$$\frac{\text{tr}(V^T [A + \xi_1 B] V)}{\text{tr}(V^T B V)} + \text{tr}(V^T [C + \xi_2 I] V) = \xi_1 + \ell \xi_2 + \frac{\text{tr}(V^T A V)}{\text{tr}(V^T B V)} + \text{tr}(V^T C V).$$

- ▶ Tested methods

- ▶ `OptM` of [Wen & Yin] on $-f(V)$ (optman.blogs.rice.edu/)
- ▶ `sg_min` of [Ruppert & Edelman] on $-f(V)$ (web.mit.edu/~ripper/www/sgmin.html)

Methods	Identifier
Fletcher-Reeves CG	<code>frcg</code>
Polak-Ribière CG	<code>prcg</code>
Newton	<code>newton</code>
dog-leg Newton	<code>dog</code>

- ▶ The SCF for the eigenvalue problem with eigenvector nonlinearity

15. Numerical results for random matrices, cont'd

- ▶ # of outer iterations averaged over 20 random tests

ℓ	n	SCF	sg_min				OptM
			frcg	prcg	Newton	dog	
3	100	5.20	51.80	252.10	9.20	12.10	72.10
	200	5.00	61.10	297.60	9.80	13.20	81.20
	500	4.70	111.12	653.90	11.30	15.70	97.70
	1000	4.60	110.90	875.60	11.30	16.70	123.80
	2000	4.00	110.50	601.50	11.60	16.90	137.70
5	100	5.20	44.40	231.50	9.00	11.70	61.80
	200	4.90	62.60	384.20	9.80	14.00	83.50
	500	4.50	86.60	448.90	10.60	15.10	108.80
	1000	4.00	152.30	1183.50	11.60	17.10	150.20
	2000	4.00	124.80	737.20	11.90	17.40	151.60

- ▶ Residuals r_k averaged over 20 random tests

ℓ	n	SCF	sg_min				OptM
			frcg	prcg	Newton	dog	
3	100	2.09e-09	9.61e-5	1.02e-4	7.23e-5	1.01e-4	3.94e-5
	200	2.85e-09	2.01e-4	2.10e-4	1.50e-4	2.15e-4	3.20e-4
	500	1.24e-09	5.15e-4	5.17e-4	3.96e-4	5.93e-4	6.78e-4
	1000	2.90e-09	1.04e-3	1.02e-3	8.15e-4	1.26e-3	1.77e-4
	2000	1.88e-09	2.04e-3	2.04e-3	1.74e-3	2.61e-3	4.80e-4
5	100	3.53e-09	1.03e-4	1.10e-4	7.54e-5	1.24e-4	5.65e-5
	200	9.92e-10	2.38e-4	2.18e-4	1.74e-4	2.56e-4	2.15e-4
	500	3.62e-09	5.78e-4	5.36e-4	5.03e-4	7.85e-4	8.03e-4
	1000	2.17e-09	1.20e-3	1.10e-3	8.95e-3	1.64e-3	1.59e-4
	2000	4.14e-10	2.42e-3	2.09e-3	1.75e-3	3.31e-3	4.25e-4

16. Summary

- ▶ Studied the S2TR problem.
- ▶ Obtained 1st (KKT) and 2nd order conditions, necessary condition for local maximizer. *Most importantly, necessary condition for global maximizer – naturally leading to SCF method.*
- ▶ *In general, no unconditional global convergence possible – an counterexample.*
- ▶ Performed random numerical tests, demonstrating SCF is very efficient and superior to generic Stiefel manifold-based optimization methods: `OptM` and `sg_min`, when it works. No divergence is encountered in all random testing.
- ▶ Idea may be extensible to sums of more than two trace ratios (SCF easy, but various conditions need to be carefully examined)

17. Further reading

- ▶ L. H. Zhang and R.-C. Li, Maximization of the sum of the trace ratio on the stiefel manifold. I: Theory & II: Computation, *SCIENCE CHINA Math.*, 57(2014), 2495-2508, & 58(2015), 1549-1566.
- ▶ P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms On Matrix Manifolds*. Princeton University Press, Princeton, NJ, 2008.

Part 3.3 Robust eigenvector classifiers

1. Data classification

▶ Training data points

Class A: $a_1, a_2, \dots, a_m \in \mathbb{R}^n$.

Class B: $b_1, b_2, \dots, b_p \in \mathbb{R}^n$.

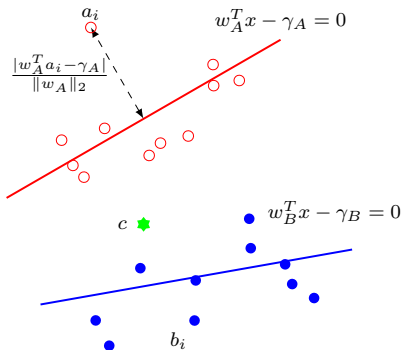
▶ Find hyperplane \mathcal{L}_A : $w_A^T x - \gamma_A = 0$ for $x \in \mathbb{R}^n$ to minimize the ratio

$$\frac{\text{dist}(\text{class A}, \mathcal{L}_A)}{\text{dist}(\text{class B}, \mathcal{L}_A)} = \frac{\sum_{i=1}^m |w_A^T a_i - \gamma_A|^2}{\sum_{j=1}^p |w_A^T b_j - \gamma_A|^2}$$

▶ In analogy, define hyperplane \mathcal{L}_B for class B: $w_B^T x - \gamma_B = 0$.

▶ For a testing point $c \in \mathbb{R}^n$, classify

$$\text{class}(c) = \underset{\ell \in \{A, B\}}{\text{argmin}} \frac{|w_\ell^T c - \gamma_\ell|}{\|w_\ell\|_2}.$$



2. Generalized eigenvalue problem

- ▶ The optimal hyperplane $z = \begin{bmatrix} w_A \\ \gamma_A \end{bmatrix}$ minimizes the Rayleigh quotient

$$\min_z \frac{\sum_{i=1}^m |w_A^T a_i - \gamma_A|^2}{\sum_{j=1}^p |w_A^T b_j - \gamma_A|^2} = \min_z \frac{z^T ([A, e]^T [A, e]) z}{z^T ([B, e]^T [B, e]) z}$$

where $A^T = [a_1, \dots, a_m] \in \mathbb{R}^{n \times m}$, $B^T = [b_1, \dots, b_p] \in \mathbb{R}^{n \times p}$, and e is a vector of ones.

- ▶ The optimal solution z by solving the following Hermitian generalized eigenvalue problem for the smallest eigenvalue λ :

$$Gz = \lambda Hz,$$

where $G = [A, e]^T [A, e]$ and $H = [B, e]^T [B, e]$.

3. Ref. O. Mangasarian and W. Wild, Multisurface proximal support vector machine classification via generalized eigenvalues, IEEE Trans. Pattern Analysis and Machine Intelligence, 27(2), pp.1-6, 2005

4. Data uncertainty

- ▶ Due to data uncertainty, the data points may vary in the ellipsoids

$$\hat{a}_i \in S_i^{(A)} := \{a_i + \delta : \delta^T \Sigma_i^{(A)} \delta \leq 1\}$$

$$\hat{b}_j \in S_j^{(B)} := \{b_j + \delta : \delta^T \Sigma_j^{(B)} \delta \leq 1\}$$

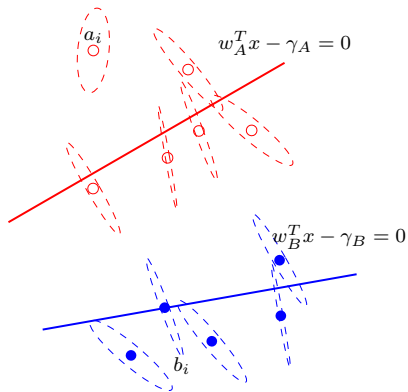
where

- ▶ $(a_i)_{i=1}^m$ and $(b_j)_{j=1}^p$ are given ellipsoids center.
 - ▶ $\Sigma_i^{(A)}$ and $\Sigma_j^{(B)}$ are symmetric positive definite matrices.
- ▶ Robust minimization (worst-case)

$$\min_{z = \begin{bmatrix} w_A \\ \gamma_A \end{bmatrix}} \left(\max \frac{\sum_{i=1}^m |w_A^T \hat{a}_i - \gamma_A|^2}{\sum_{j=1}^p |w_A^T \hat{b}_j - \gamma_A|^2} \right)$$

$$\text{s.t. } \hat{a}_i \in S_i^{(A)} \text{ for } i = 1: m,$$

$$\hat{b}_j \in S_j^{(B)} \text{ for } j = 1: p.$$



5. The inner max problem in the robust min:

$$\left(\begin{array}{l} \max \frac{\sum_{i=1}^m |w_A^T \hat{a}_i - \gamma_A|^2}{\sum_{j=1}^p |w_A^T \hat{b}_j - \gamma_A|^2} \\ \text{s.t. } \hat{a}_i \in S_i^{(A)} \text{ for } i = 1: m, \\ \hat{b}_j \in S_j^{(B)} \text{ for } j = 1: p \end{array} \right) = \frac{\sum_{i=1}^m \max_{\hat{a}_i \in S_i^{(A)}} |w_A^T \hat{a}_i - \gamma_A|^2}{\sum_{j=1}^p \min_{\hat{b}_j \in S_j^{(B)}} |w_A^T \hat{b}_j - \gamma_A|^2}$$

- ▶ Each 'max' sub problem is solved by using the fact

$$\max_{\delta^T \Sigma \delta \leq 1} |w^T (a + \delta) - \gamma|^2 = \left(w^T (a + \delta_*) - \gamma \right)^2,$$

where

$$\delta_* = \frac{\text{sgn}(w^T a - \gamma)}{\sqrt{w^T \Sigma^{-1} w}} \Sigma^{-1} w.$$

- ▶ Each 'min' sub problem is solved by exploiting

$$\min_{\delta^T \Sigma \delta \leq 1} |w^T (b + \delta) - \gamma|^2 = \left(w^T (b + \delta_*) - \gamma \right)^2,$$

where

$$\delta_* = \frac{\text{sgn}(\gamma - w^T b)}{\sqrt{w^T \Sigma^{-1} w}} \Sigma^{-1} w.$$

Note that it is assumed that $w^T (b + \delta) - \gamma \neq 0, \forall \delta$ with $\delta^T \Sigma \delta \leq 1$, otherwise $\min = 0$.

6. By analytic solutions of the inner optimal problems, we obtain the following nonlinear Rayleigh quotient minimization:

$$\min_{z = \begin{bmatrix} w_A \\ \gamma_A \end{bmatrix}} \rho(z) := \frac{z^T G(z) z}{z^T H(z) z},$$

where

$$G(z) = [A + \Delta A(z), -e]^T [A + \Delta A(z), -e],$$

$$H(z) = [B + \Delta B(z), -e]^T [B + \Delta B(z), -e],$$

and

$$\Delta A(z) = \begin{bmatrix} \frac{\text{sgn}(w^T a_1 - \gamma)}{\sqrt{w^T \Sigma_1^{(A)-1} w}} \cdot w^T \Sigma_1^{(A)-1} \\ \frac{\text{sgn}(w^T a_2 - \gamma)}{\sqrt{w^T \Sigma_2^{(A)-1} w}} \cdot w^T \Sigma_2^{(A)-1} \\ \vdots \\ \frac{\text{sgn}(w^T a_m - \gamma)}{\sqrt{w^T \Sigma_m^{(A)-1} w}} \cdot w^T \Sigma_m^{(A)-1} \end{bmatrix}, \quad \Delta B(z) = \begin{bmatrix} \frac{\text{sgn}(\gamma - w^T b_1)}{\sqrt{w^T \Sigma_1^{(B)-1} w}} \cdot w^T \Sigma_1^{(B)-1} \\ \frac{\text{sgn}(\gamma - w^T b_2)}{\sqrt{w^T \Sigma_2^{(B)-1} w}} \cdot w^T \Sigma_2^{(B)-1} \\ \vdots \\ \frac{\text{sgn}(\gamma - w^T b_p)}{\sqrt{w^T \Sigma_p^{(B)-1} w}} \cdot w^T \Sigma_p^{(B)-1} \end{bmatrix}.$$

Remark: in the derivation, we assumed the hyperplane \mathcal{L}_A does not intersect the ellipsoids $S_j^{(B)}$ for $j = 1, 2, \dots, p$.)

7. Since $G(z) \equiv G(\alpha z)$ and $H(z) \equiv H(\alpha z)$, for $\alpha \neq 0$, are homogeneous in z , we can rewrite the nonlinear Rayleigh quotient problem as

$$\min_{z \in \mathbb{R}^n} z^T G(z) z \quad \text{s.t.} \quad z^T H(z) z = 1.$$

8. Define the Lagrangian function with multiplier λ

$$L(z, \lambda) = z^T G(z) z - \lambda(z^T H(z) z - 1).$$

9. By straightforward derivation, we obtain the gradient of the Lagrangian

$$\nabla_z L(z, \lambda) = 2 \left(G(z) + \tilde{G}(z) - \lambda(H(z) + \tilde{H}(z)) \right) z$$

$$\nabla_\lambda L(z, \lambda) = z^T H(z) z - 1.$$

where

$$\tilde{G}(z) = \begin{bmatrix} z^T \frac{\partial G(z)}{\partial z_1} \\ \vdots \\ z^T \frac{\partial G(z)}{\partial z_{n+1}} \end{bmatrix} \quad \text{and} \quad \tilde{H}(z) = \begin{bmatrix} z^T \frac{\partial H(z)}{\partial z_1} \\ \vdots \\ z^T \frac{\partial H(z)}{\partial z_{n+1}} \end{bmatrix}.$$

10. Take the derivative with respect to z again, we obtain Hessian matrix

$$\nabla_{zz}L(z, \lambda) = 2\left(G(z) + \tilde{G}(z) - \lambda(H(z) + \tilde{H}(z))\right) \equiv 2\left(\mathcal{G}(z) - \lambda\mathcal{H}(z)\right),$$

where we used the facts $\tilde{G}(z)z = 0$ and $\tilde{H}(z)z = 0$.

11. The first-order optimality conditions for the constrained opt prob are given by

$$\nabla_z L(z, \lambda) = 0 \quad \text{and} \quad z^T H(z)z = 1.$$

This leads to a (local) optimizer z_* is an eigenvector of the eigenvalue problem:

$$\mathcal{G}(z)z = \lambda\mathcal{H}(z)z. \tag{37}$$

12. The second-order optimality condition¹⁰ is

$$s^T \nabla_{zz}L(z, \lambda)s \geq 0$$

for all $s \neq 0$ and $s^T H(z)z = 0$. This immediately leads to the condition

$$\mathcal{G}(z) - \lambda\mathcal{H}(z) \succeq 0. \tag{38}$$

13. In order to satisfy the semi-positive definite condition (38), the corresponding eigenvalue λ_* at the local minimizer z_* must be the **least positive eigenvalue** of the matrix pair $(\mathcal{G}(z_*), \mathcal{H}(z_*))$

¹⁰Reference to opt cond of constrained opt prob: J. Nocedal and S. Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

14. Since $\tilde{G}(z)z = 0$ and $\tilde{H}(z)z = 0$,

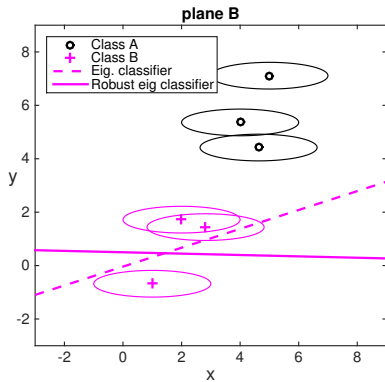
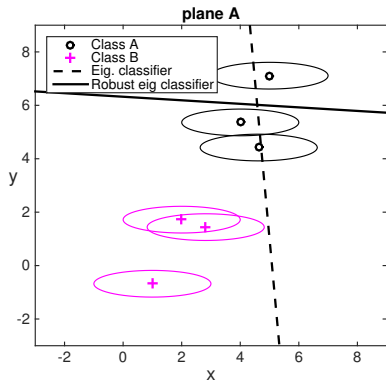
$$(\mathcal{G}(z) - \lambda\mathcal{H}(z))z \equiv (G(z) - \lambda H(z))z = 0,$$

Therefore, the local optimizer z_* is also an eigenvector of “simplified” problem

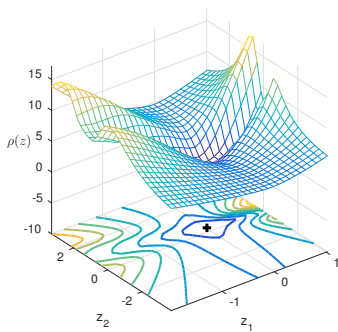
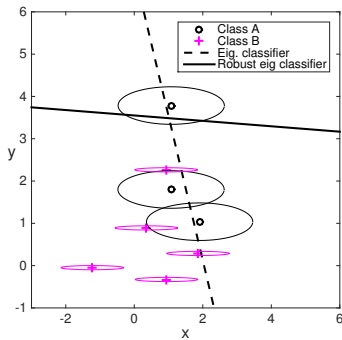
$$G(z)z = \lambda H(z)z. \tag{39}$$

However, as an example shown later, the corresponding eigenvalue λ_* may *not* be the smallest eigenvalue of the matrix pair $(G(z_*), H(z_*))$.

15. A simple example with $n = 2$



16. A simple example with $n = 2$



- ▶ Apply SCF to solve the second-order NEP $\mathcal{G}(z)z = \lambda\mathcal{H}(z)z$ for the least positive eigval, we obtain the eigvec \hat{z} (marked + in the contour plot of $\rho(z)$ on the right.)

16. A simple example $n = 2$, cont'd

- ▶ Eigenvalues at the local minimizer z_* :

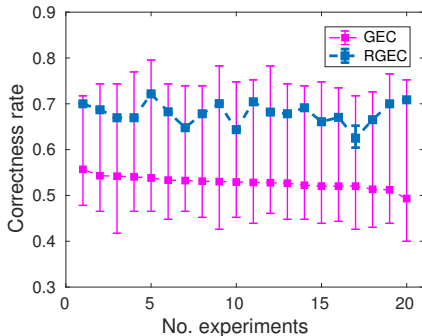
$$\begin{array}{lll} \text{1st-order NEP} & G(z_*)z = \lambda H(z_*)z & \\ & \lambda_1 = 0.0747100 & \underline{\lambda_2 = 0.2991528} \quad \lambda_3 = 3.5632835 \end{array}$$

$$\begin{array}{lll} \text{2nd-order NEP} & \mathcal{G}(z_*)z = \lambda \mathcal{H}(z_*)z & \\ & \lambda_1 = -0.4960373 & \underline{\lambda_2 = 0.2991528} \quad \lambda_3 = 3.3250210 \end{array}$$

- ▶ Note that the corresponding “optimal” eigenvalue λ_* is *not* the smallest eigenvalue of the matrix pair $(G(z_*), H(z_*))$. **In general, the correspondence is unknown.**
- ▶ The corresponding “optimal” eigenvalue λ_* is the **least positive eigenvalue** of the matrix pair $(\mathcal{G}(z_*), \mathcal{H}(z_*))$

17. Example.

- ▶ Classification experiments for the “Pima Indians Diabetes dataset” from UCI Machine Learning Repository, available at <http://archive.ics.uci.edu/ml>.
- ▶ Total instances = 768 (sample 70% for training, 30% for testing).
- ▶ Number of features $n = 8$.
- ▶ Perturbation ellipsoids: $\Sigma^{-1} = \text{diag}(\alpha_1^2 \bar{x}_1^2, \dots, \alpha_n^2 \bar{x}_n^2)$
 - ▶ \bar{x}_i = average value of i -th feature (of all instances).
 - ▶ α_i = perturbation level.
 - $\alpha_i = 0.001$ for feature 1 (pregnant times) and feature 8 (age),
 - $\alpha_i = 0.2$ for the rest.
- ▶ Testing data points perturbation: $x = x + \delta x$ with $\delta x \sim \mathcal{N}(0, \Sigma)$.
- ▶ Correctness rates and variances of GEC (generalized eigenvector classifier) vs. RGEC (robust generalized eigenvector classifier)



- ▶ Significant improvements and much smaller variance of the correctness rate.

Part 3.4 Of things not treated

1. Gross-Pitaevskii type equations

$$-\Delta u + \dots + \zeta|u|^2 u = \lambda u$$

- ▶ W. Bao and Q. Du, Computing the ground state solution of Bose-Einstein condensates by a normalized gradient flow. *SIAM J Sci Comput*, 25, pp.1674-1697, 2004
- ▶ S.H. Jia, H.-H. Xie, M.-T. Xie and F. Xu, A full multigrid method for nonlinear eigenvalue problems, *Sci. China Math.* 59(10).pp.2037-2048, 2016
- ▶ E. Jarlebring, S. Kvaal and W. Michiels, An inverse iteration method for eigenvalue problems with eigenvector nonlinearities, *SIAM J. Sci. Comput.* 36(4), pp.A1978-A2001, 2014 (Inverse iteration for $A(v)v = \lambda v$)

2. Spectral graph theory

- ▶ M. Hein and T. Bühler, An inverse power method for nonlinear eigenproblems with applications in 1-Spectral clustering and sparse PCA, *NIPS*, pp.847-855, 2010
- ▶ L. Jost, S. Setzer and M. Hein, Nonlinear eigenproblems in data analysis – balanced graph cuts and the RatioDCA-Prox, arXiv:1312.5192v2, Mar. 2014. $(\min_f \frac{R(f)}{S(f)})$

Review and some random-thoughts

- ▶ Part 1. Linear eigenvalue problems " $Ax = \lambda Bx$ "
 1. Accelerated subspace iteration
 - ▶ Polynomial and rational approximations of *ideal/optimal accelerator* of symmetric LEPs
 - ▶ Notion of *optimal accelerator/preconditioners* for nonsymmetric LEPs?
 2. Steepest descent method
 - ▶ Eigenvalue problems \leftrightarrow Optimization of Rayleigh quotient (variational form)
 3. Arnoldi method
 - ▶ Krylov subspace and MATLAB's `eigs.m`
 4. Rational Krylov method
 - ▶ Multi-shift-invert Krylov subspace method
 5. Topics of more recent interest
 - ▶ Computing many eigenpairs – deflations vs. spectrum slicing?
 - ▶ Notion of a numerically "ill-conditioned" eigenproblems?

Review and some random-thoughts, cont'd

- ▶ Part 2. Nonlinear eigenvalue problems " $T(\lambda)x = 0$ "
 1. Essential theory
 - ▶ sensitivity/perturbations of (structured) NEPs?
 2. Methods based on Newton iteration
 - ▶ select proper objective function for root-finding
 - ▶ exploit structures/properties of a specific problem
 3. Methods specially designed for QEP and REP
 - ▶ QEP: essential idea of compact representation (two-level-orthogonality) of the basis vectors of the Krylov subspace
 - ▶ REP: how to exploit the structure of the trimmed linearization?
 - ▶ PAL: exploit the low-rank property of the QEP, and unconventional three-stage approach.
 4. Methods based on approximation and linearization
 - ▶ local vs. "global" approximations?
 - ▶ error analysis of approximations?
 - ▶ compact representation of projection subspace for (much-large) linearized eigenproblems
 5. Of things not treated
 - ▶ Contour Integral methods for NEPs?

Review and some random-thoughts, cont'd

▶ Part 3. Eigenvalue problems with eigenvector nonlinearity

1. Kohn-Sham eigenvalue problem " $H(X)X = X\Lambda$ "

- ▶ Convergence analysis of plain and "improved" SCF iterations
- ▶ An open contest: optimal bound for α for the SCF convergence of the "simple" model in [Yang-Gao-Meza'09]

2. Sum of trace ratio " $E(V)V = VM(V)$ "

- ▶ optimization on matrix manifolds \leftrightarrow eigenvalue problems

3. Robust Rayleigh quotient optimization " $G(z)z = \lambda H(z)z$ "

- ▶ generalized Rayleigh quotient $\frac{z^T G(z)z}{z^T H(z)z}$, optimization, eigenvalue ordering

4. Of things not treated

- ▶ Eigenvalue problems from "non-smooth" optimizations [Hein et al'10]