



MAX PLANCK INSTITUTE
FOR DYNAMICS OF COMPLEX
TECHNICAL SYSTEMS
MAGDEBURG



COMPUTATIONAL METHODS IN
SYSTEMS AND CONTROL THEORY

Novel Algorithms for Solving Structured Eigenvalue Problems Arising in the Description of Electronic Excitations

Carolín Penke, Forschungszentrum Jülich

Nov 17, 2022

CECAM flagship workshop: Challenges and Advances in Solving
Eigenproblems for Electronic-Structure Theory

Partners:



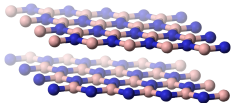
MAX PLANCK COMPUTING & DATA FACILITY
RECHENZENTRUM GARCHING DER MAX-PLANCK-GESELLSCHAFT



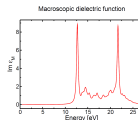
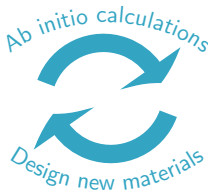
Inter-University Research Institute Corporation /
Research Organization of Information and Systems
National Institute of Informatics



HPC algorithms



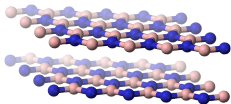
Physical structure of material



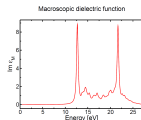
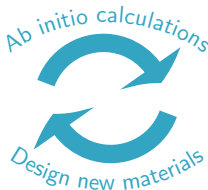
Optical properties of material



HPC algorithms



Physical structure of material



Optical properties of material



Improved features
for applications





- Bethe-Salpeter equation determines polarizability propagator

$$\chi(\omega) \approx \left(\omega \begin{bmatrix} I & 0 \\ 0 & -I \end{bmatrix} - H \right)^{-1}.$$

- Specific matrix H is determined by physical system and choice of basis functions.



Eigenpairs of

$$\mathcal{H} = \begin{bmatrix} A & B \\ -B^H & -A^T \end{bmatrix},$$
$$A = A^H, B = B^T \in \mathbb{C}^{n \times n}$$

give *optical properties* of material.



$$\mathcal{H} = \begin{bmatrix} A & B \\ -B^H & -A^T \end{bmatrix},$$
$$A = A^H, B = B^T \in \mathbb{C}^{n \times n}$$

$\Updownarrow$ [BENNER, PENKE '21]

2 symmetries:

$$J := \begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix}$$

$$J\mathcal{H} = -(J\mathcal{H})^T$$

“Hamiltonian”

$$\Sigma := \begin{bmatrix} I & 0 \\ 0 & -I \end{bmatrix}$$

$$\Sigma\mathcal{H} = (\Sigma\mathcal{H})^H$$

“Pseudosymmetric”



$$\mathcal{H} = \begin{bmatrix} A & B \\ -B^H & -A^T \end{bmatrix},$$
$$A = A^H, B = B^T \in \mathbb{C}^{n \times n}$$

$\Updownarrow$ [BENNER, PENKE '21]

2 symmetries:

$$J := \begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix}$$

$$J\mathcal{H} = -(J\mathcal{H})^T$$

“Hamiltonian”

$$\Sigma := \begin{bmatrix} I & 0 \\ 0 & -I \end{bmatrix}$$

$$\Sigma\mathcal{H} = (\Sigma\mathcal{H})^H$$

“Pseudosymmetric”

1st approach: transformation to skew-symmetric

ELPA

2nd approach: pseudosymmetric spectral divide-and-conquer



$$\mathcal{H} = \begin{bmatrix} A & B \\ -B^H & -A^T \end{bmatrix},$$
$$A = A^H, B = B^T \in \mathbb{C}^{n \times n}$$

$\Updownarrow$ [BENNER, PENKE '21]

2 symmetries:

$$J := \begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix}$$

$$J\mathcal{H} = -(J\mathcal{H})^T$$

"Hamiltonian"

1st approach: transformation to skew-symmetric

ELPA

$$\Sigma := \begin{bmatrix} I & 0 \\ 0 & -I \end{bmatrix}$$

$$\Sigma\mathcal{H} = (\Sigma\mathcal{H})^H$$

"Pseudosymmetric"

2nd approach: pseudosymmetric spectral divide-and-conquer



Definition

Let $J = \begin{bmatrix} 0 & I_n \\ -I_n & 0 \end{bmatrix}$. A matrix $H \in \mathbb{R}^{2n \times 2n}$ is called Hamiltonian if

$$JH = (JH)^T.$$

- Also important in control theory.
- If $-JH > 0$, the problem can be recast using a Cholesky decomposition $-JH = LL^T$, skew-symmetric

$$Hx = \lambda x \quad \Leftrightarrow \quad \underbrace{L^T J L}_{\text{skew-symmetric}} \hat{x} = \lambda \hat{x} \quad \text{with } \hat{x} = L^T x.$$



Definition

Let $J = \begin{bmatrix} 0 & I_n \\ -I_n & 0 \end{bmatrix}$. A matrix $H \in \mathbb{R}^{2n \times 2n}$ is called Hamiltonian if

$$JH = (JH)^T.$$

- Also important in control theory.
- If $-JH > 0$, the problem can be recast using a Cholesky decomposition $-JH = LL^T$, skew-symmetric

$$Hx = \lambda x \quad \Leftrightarrow \quad \underbrace{L^T J L}_{\text{skew-symmetric}} \hat{x} = \lambda \hat{x} \quad \text{with } \hat{x} = L^T x.$$



Definition

Let $J = \begin{bmatrix} 0 & I_n \\ -I_n & 0 \end{bmatrix}$. A matrix $H \in \mathbb{R}^{2n \times 2n}$ is called Hamiltonian if

$$JH = (JH)^T.$$

- Also important in control theory.
- If $-JH > 0$, the problem can be recast using a Cholesky decomposition $-JH = LL^T$, skew-symmetric

$$Hx = \lambda x \quad \Leftrightarrow \quad \underbrace{L^T J L}_{\text{skew-symmetric}} \hat{x} = \lambda \hat{x} \quad \text{with } \hat{x} = L^T x.$$

- Principle applied on Bethe-Salpeter EVP by [SHAO, DA JORNADA, YANG, DESLIPPE, LOUIE '15].
- We developed HPC eigenvalue solver for large, dense skew-symmetric matrices!



- Very popular library for solving symmetric eigenvalue problems in computational quantum physics:

The ELPA Project

Eigenvalue SoLvers for Petaflop-Appl*ications*

*The publicly available ELPA library provides **highly efficient** and **highly scalable** direct eigensolvers for **symmetric matrices**. Though especially designed for use for PetaFlop/s applications solving large problem sizes on **massively parallel supercomputers**, ELPA eigensolvers have proven to be also very efficient for smaller matrices.*

- Mainly developed at *Max Planck Computing and Data Facility* (MPCDF) in Garching.



The eigenvalue decomposition

$$A = Q^T \Lambda Q$$

is computed in three steps.



The eigenvalue decomposition

$$A = Q^T \Lambda Q$$

is computed in three steps.

1. Reduce A to tridiagonal form

$$A_{trd} = Q_{trd}^T A Q_{trd} = \begin{bmatrix} \diagdown \\ \diagup \end{bmatrix}.$$



The eigenvalue decomposition

$$A = Q^T \Lambda Q$$

is computed in three steps.

1. Reduce A to tridiagonal form

$$A_{trd} = Q_{trd}^T A Q_{trd} = \begin{bmatrix} \diagdown & & \\ & \diagdown & \\ & & \diagdown \end{bmatrix}.$$

2. Solve the tridiagonal eigenvalue problem

$$\Lambda = Q_{diag}^T A_{trd} Q_{diag} = \begin{bmatrix} \diagdown & & \\ & \diagdown & \\ & & \diagdown \end{bmatrix}.$$



The eigenvalue decomposition

$$A = Q^T \Lambda Q$$

is computed in three steps.

1. Reduce A to tridiagonal form

$$A_{trd} = Q_{trd}^T A Q_{trd} = \begin{bmatrix} \diagdown \end{bmatrix}.$$

2. Solve the tridiagonal eigenvalue problem

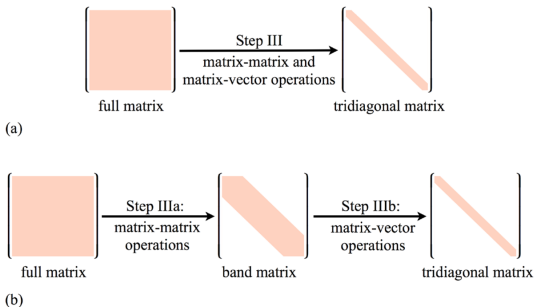
$$\Lambda = Q_{diag}^T A_{trd} Q_{diag} = \begin{bmatrix} \diagdown \end{bmatrix}.$$

3. **Optional:** Transform the desired eigenvectors back,

$$Q = Q_{trd} Q_{diag}.$$



- The reduction to tridiagonal form can happen in one (ELPA1) or two steps (ELPA2).



- Two step-reduction:

Pro: Faster due to higher amount of BLAS3 operations.

Con: Additional step in the eigenvector back-transformation.

→ Better runtime when not all eigenvectors are required.



Lemma

[WARD, GRAY '78]

With the unitary matrix $D = \text{diag}\{1, i, i^2, \dots, i^{n-1}\}$, where i denotes the imaginary unit, $\alpha_j \in \mathbb{R}$, it holds

$$-iD^H \begin{bmatrix} 0 & \alpha_1 & & & \\ -\alpha_1 & 0 & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & \alpha_{n-1} \\ & & & & & 0 \end{bmatrix} D = \begin{bmatrix} 0 & \alpha_1 & & & \\ \alpha_1 & 0 & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & \alpha_{n-1} \\ & & & & & 0 \end{bmatrix}.$$



Lemma

[WARD, GRAY '78]

With the unitary matrix $D = \text{diag}\{1, i, i^2, \dots, i^{n-1}\}$, where i denotes the imaginary unit, $\alpha_j \in \mathbb{R}$, it holds

$$-iD^H \begin{bmatrix} 0 & \alpha_1 & & & \\ -\alpha_1 & 0 & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & \alpha_{n-1} \\ & & & & & 0 \end{bmatrix} D = \begin{bmatrix} 0 & \alpha_1 & & & \\ \alpha_1 & 0 & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & \alpha_{n-1} \\ & & & & & 0 \end{bmatrix}.$$

→ Instead of solving skew-symmetric eigenvalue problem we can solve a symmetric eigenvalue problem.



Lemma

[WARD, GRAY '78]

With the unitary matrix $D = \text{diag}\{1, i, i^2, \dots, i^{n-1}\}$, where i denotes the imaginary unit, $\alpha_j \in \mathbb{R}$, it holds

$$-iD^H \begin{bmatrix} 0 & \alpha_1 & & & \\ -\alpha_1 & 0 & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & \alpha_{n-1} & 0 \\ & & & & -\alpha_{n-1} & 0 \end{bmatrix} D = \begin{bmatrix} 0 & \alpha_1 & & & \\ \alpha_1 & 0 & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & \alpha_{n-1} & 0 \\ & & & & \alpha_{n-1} & 0 \end{bmatrix}.$$

- Instead of solving skew-symmetric eigenvalue problem we can solve a symmetric eigenvalue problem.
- Any tridiagonalization method for symmetric matrices based on Householder transformations also works for skew-symmetric matrices.



Specific changes in the code:

1. Tridiagonalization (ELPA1 and ELPA2):

- Identify points in code where symmetry is implicitly assumed for more efficient computations.
- Change signs or employ skew-symmetric BLAS routines such that skew-symmetry is assumed.



Specific changes in the code:

1. Tridiagonalization (ELPA1 and ELPA2):

- Identify points in code where symmetry is implicitly assumed for more efficient computations.
- Change signs or employ skew-symmetric BLAS routines such that skew-symmetry is assumed.

2. Eigenvector back transformation:

- Multiplication with matrix D as additional step.
- Introduces complex values.



Specific changes in the code:

1. Tridiagonalization (ELPA1 and ELPA2):

- Identify points in code where symmetry is implicitly assumed for more efficient computations.
- Change signs or employ skew-symmetric BLAS routines such that skew-symmetry is assumed.

2. Eigenvector back transformation:

- Multiplication with matrix D as additional step.
- Introduces complex values.

3. The eigenvalues of a skew-symmetric matrix come in purely imaginary pairs $\pm i\lambda$, $\lambda \in \mathbb{R}$. $\Rightarrow$ Only half the eigenpairs need to be computed.



Specific changes in the code:

1. Tridiagonalization (ELPA1 and ELPA2):
 - Identify points in code where symmetry is implicitly assumed for more efficient computations.
 - Change signs or employ skew-symmetric BLAS routines such that skew-symmetry is assumed.
2. Eigenvector back transformation:
 - Multiplication with matrix D as additional step.
 - Introduces complex values.
3. The eigenvalues of a skew-symmetric matrix come in purely imaginary pairs $\pm i\lambda$, $\lambda \in \mathbb{R}$. $\Rightarrow$ Only half the eigenpairs need to be computed.

Stays the same:

- Tridiagonal solve.
- Transformation of real and imaginary parts of eigenvectors.

**Algorithm 1:** Solution of a Skew-symmetric Eigenvalue Problem

[BENNER, P., MAREK '19]

Input: $A = -A^T \in \mathbb{R}^{n \times n}$ **Output:** Unitary eigenvectors $Q \in \mathbb{C}^{n \times n}$, $\lambda_1, \dots, \lambda_n \in \mathbb{R}$ s.t. $Q^H A Q = \text{diag}\{\lambda_1 i, \dots, \lambda_n i\}$.1: Reduce A to tridiagonal form, i.e. generate Q_{trd} s.t.

$$Q_{trd}^T A Q_{trd} = A_{trd} = \begin{bmatrix} 0 & \alpha_1 & & & \\ -\alpha_1 & 0 & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & & -\alpha_{n-1} & 0 \\ & & & & \alpha_{n-1} \end{bmatrix}.$$

2: Solve the eigenvalue problem for the symmetric tridiagonal matrix $-iD^H A_{trd} D$, where $D = \text{diag}\{1, i, i^2, \dots, i^n\}$, i.e. generate Q_{diag} s.t.

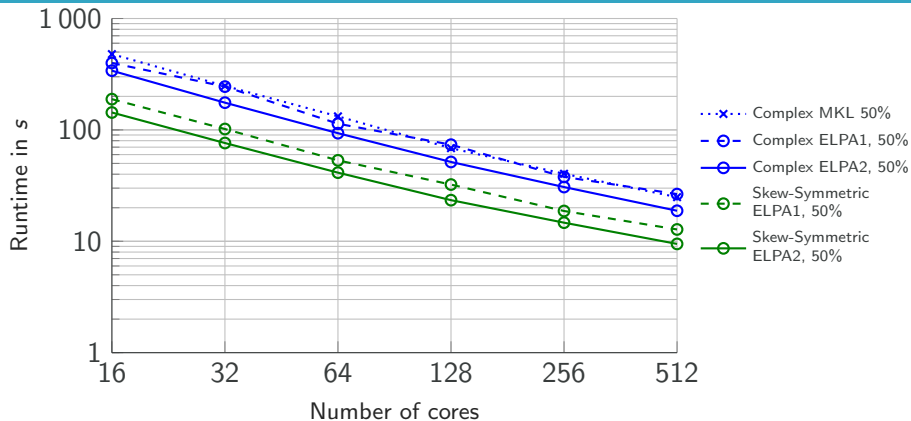
$$Q_{diag}^T \begin{bmatrix} 0 & \alpha_1 & & & \\ \alpha_1 & 0 & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & & -\alpha_{n-1} & 0 \\ & & & & \alpha_{n-1} \end{bmatrix} Q_{diag} = \begin{bmatrix} \lambda_1 & & & & \\ & \lambda_2 & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & \lambda_n \end{bmatrix}.$$

3: Back transformation corresponding to symmetrization, i.e. compute

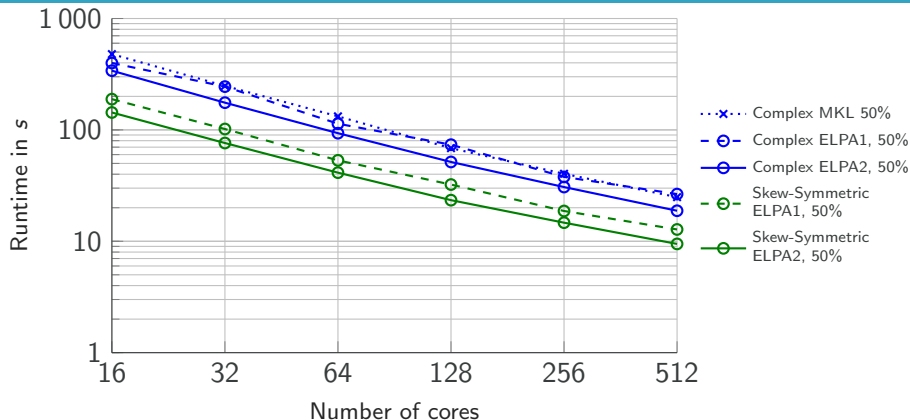
$$Q \leftarrow D Q_{diag} \in \mathbb{C}^{n \times n}.$$

4: Back transformation corresponding to band-to-tridiagonal reduction, i.e. compute

$$Q \leftarrow Q_{trd} Q.$$



Scaling of the ELPA solver for skew-symmetric matrices of size $n = 20\,000$ on Mechthild compute cluster at MPI Magdeburg.



Scaling of the ELPA solver for skew-symmetric matrices of size $n = 20\,000$ on Mechthild compute cluster at MPI Magdeburg.

→ Skew-Symmetric ELPA is **3x faster** than Intel MKL!



$$\mathcal{H} = \begin{bmatrix} A & B \\ -B^H & -A^T \end{bmatrix},$$
$$A = A^H, B = B^T \in \mathbb{C}^{n \times n}$$

$\Updownarrow$ [BENNER, P. '21]

2 symmetries:

$$J := \begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix}$$

$$J\mathcal{H} = -(J\mathcal{H})^T$$

“Hamiltonian”

1st approach: transformation to skew-symmetric

ELPA

$$\Sigma := \begin{bmatrix} I & 0 \\ 0 & -I \end{bmatrix}$$

$$\Sigma\mathcal{H} = (\Sigma\mathcal{H})^H$$

“Pseudosymmetric”

2nd approach: pseudosymmetric spectral divide-and-conquer



Idea

- Generalize a method for symmetric eigenvalue problems to work for pseudosymmetric matrices!
- Method of choice:
spectral divide-and-conquer.
- Used tools need to be generalized:
 - Polar decomposition
 - QR decomposition



Photo by Thiago Matos
[www.pexels.com/photo/mirror-fragments-on-gray-surface
-with-the-reflection-of-a-person-s-hand-3022456/](https://www.pexels.com/photo/mirror-fragments-on-gray-surface-with-the-reflection-of-a-person-s-hand-3022456/)



Goal: Diagonalize matrix A !

- An invariant subspace basis V_1 of a matrix A and a complementing basis V_2 can be used to transform a matrix into two smaller blocks:

$$V = [V_1 \quad V_2], \quad V^{-1}AV = \begin{bmatrix} A_{11} & 0 \\ 0 & A_{22} \end{bmatrix}$$

→ Solve two smaller eigenvalue problems instead of a big one. 😊

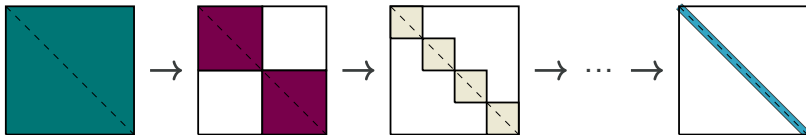


Goal: Diagonalize matrix A !

- An invariant subspace basis V_1 of a matrix A and a complementing basis V_2 can be used to transform a matrix into two smaller blocks:

$$V = [V_1 \quad V_2], \quad V^{-1}AV = \begin{bmatrix} A_{11} & 0 \\ 0 & A_{22} \end{bmatrix}$$

- Solve two smaller eigenvalue problems instead of a big one. 😊
- Repeat until diagonalization!



Symmetric structure is preserved when V is orthogonal, i.e. $V^{-1} = V^T$.



- $A \in \mathbb{R}^{n \times n}$ is called *pseudosymmetric* if it becomes symmetric when multiplied with a diagonal matrix Σ with $+1$ or -1 on the diagonal.
- Σ is called *signature matrix*.

Example:

$$A = \begin{bmatrix} 4.7 & -2.3 & -6.5 & 0.7 \\ 2.3 & -2.1 & 5.5 & 9.2 \\ 6.5 & 5.5 & 7.4 & 1.3 \\ 0.7 & -9.2 & -1.3 & 3.5 \end{bmatrix} \text{ is pseudosymmetric with } \Sigma = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$



- $A \in \mathbb{R}^{n \times n}$ is called *pseudosymmetric* if it becomes symmetric when multiplied with a diagonal matrix Σ with $+1$ or -1 on the diagonal.
- Σ is called *signature matrix*.

Example:

$$A = \begin{bmatrix} 4.7 & -2.3 & -6.5 & 0.7 \\ 2.3 & -2.1 & 5.5 & 9.2 \\ 6.5 & 5.5 & 7.4 & 1.3 \\ 0.7 & -9.2 & -1.3 & 3.5 \end{bmatrix} \text{ is pseudosymmetric with } \Sigma = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Structure	Symmetric $A^T = A$	Pseudosymmetric $A^T = \Sigma A \Sigma$
preserved by	orthogonal $V^{-1} = V^T$	$(\Sigma, \hat{\Sigma})$ -orthogonal $V^{-1} = \hat{\Sigma} V^T \Sigma$

When **another signature matrix** $\hat{\Sigma} \neq \Sigma$ is used, the transformed matrix is pseudosymmetric with respect to $\hat{\Sigma}$!



1. **Question:** How to compute invariant subspace?

■ **Answer:** Matrix sign function

The matrix sign function

[ROBERTS '71]

Let A have a Jordan decomposition

$$A = Z \begin{bmatrix} J_1 & 0 \\ 0 & J_2 \end{bmatrix} Z^{-1}, \quad \begin{array}{l} J_1 \text{ contains Jordan blocks for } \operatorname{Re}(\lambda) < 0, \\ J_2 \text{ contains Jordan blocks for } \operatorname{Re}(\lambda) > 0. \end{array}$$

$$\operatorname{sign}(A) := Z \begin{bmatrix} -I_p & 0 \\ 0 & I_q \end{bmatrix} Z^{-1}.$$



1. Question: How to compute invariant subspace?

- Answer: Matrix sign function

The matrix sign function

[ROBERTS '71]

Let A have a Jordan decomposition

$$A = Z \begin{bmatrix} J_1 & 0 \\ 0 & J_2 \end{bmatrix} Z^{-1}, \quad \begin{array}{l} J_1 \text{ contains Jordan blocks for } \operatorname{Re}(\lambda) < 0, \\ J_2 \text{ contains Jordan blocks for } \operatorname{Re}(\lambda) > 0. \end{array}$$

$$\operatorname{sign}(A) := Z \begin{bmatrix} -I_p & 0 \\ 0 & I_q \end{bmatrix} Z^{-1}.$$

- $P_+ = \frac{1}{2}(I + \operatorname{sign}(A))$ and $P_- = \frac{1}{2}(I - \operatorname{sign}(A))$ are projectors onto the invariant subspaces associated with eigenvalues on the open right and open left half-plane respectively.



1. Question: How to compute invariant subspace?

- Answer: Matrix sign function

The matrix sign function

[ROBERTS '71]

Let A have a Jordan decomposition

$$A = Z \begin{bmatrix} J_1 & 0 \\ 0 & J_2 \end{bmatrix} Z^{-1}, \quad \begin{array}{l} J_1 \text{ contains Jordan blocks for } \operatorname{Re}(\lambda) < 0, \\ J_2 \text{ contains Jordan blocks for } \operatorname{Re}(\lambda) > 0. \end{array}$$

$$\operatorname{sign}(A) := Z \begin{bmatrix} -I_p & 0 \\ 0 & I_q \end{bmatrix} Z^{-1}.$$

- $P_+ = \frac{1}{2}(I + \operatorname{sign}(A))$ and $P_- = \frac{1}{2}(I - \operatorname{sign}(A))$ are projectors onto the invariant subspaces associated with eigenvalues on the open right and open left half-plane respectively.

2. Question: How to find $(\Sigma, \hat{\Sigma})$ -orthogonal representation of that subspace?

- Answer: Hyperbolic QR decomposition.

Algorithm 2: Pseudosymmetric spectral divide-and-conquer for block diagonalization.

Data: Signature matrix Σ , pseudosymmetric A with respect to Σ , i.e. $\Sigma A = (\Sigma A)^T$.

Result: Signature matrix $\hat{\Sigma}$, and $(\Sigma, \hat{\Sigma})$ -orthogonal V such that $V^{-1}AV = D$ is block diagonal with blocks no larger than 2×2 .

- 1 Stop if A is of size 1×1 or 2×2 with a complex conjugate pair of eigenvalues.
- 2 Find shift σ such that $A - \sigma I$ has eigenvalues with positive and negative real part and no eigenvalues with zero real part.

- 3 Compute $S = \text{sign}(A - \sigma I)$ via an iteration. How?

- 4 Get a basis V_+ of range($S + I$) and V_- such that $V_0 = [V_+ \ V_-]$ is (Σ, Σ_0) -orthogonal with $\Sigma_0 = \begin{bmatrix} \Sigma_+ & \\ & \Sigma_- \end{bmatrix}$. Then

$$V_0^{-1}AV_0 = \begin{bmatrix} A_{11} & 0 \\ 0 & A_{22} \end{bmatrix}, \quad \Sigma_+ A_{11} = (\Sigma_+ A_{11})^T, \quad \Sigma_- A_{22} = (\Sigma_- A_{22})^T.$$

- 5 Repeat spectral divide-and-conquer for A_{11} with $\Sigma := \Sigma_+$, i.e. find (Σ_+, Σ_1) -orthogonal V_1 such that $V_1^{-1}A_{11}V_1 = D_{11}$ is diagonal.
 - 6 Repeat spectral divide-and-conquer for A_{22} with $\Sigma := \Sigma_-$, i.e. find (Σ_-, Σ_2) -orthogonal V_2 such that $V_2^{-1}A_{22}V_2 = D_{22}$ is diagonal.
 - 7 $V \leftarrow V_0 \begin{bmatrix} V_1 & 0 \\ 0 & V_2 \end{bmatrix}$, $\hat{\Sigma} \leftarrow \begin{bmatrix} \Sigma_1 & \\ & \Sigma_2 \end{bmatrix}$, $D \leftarrow \begin{bmatrix} D_{11} & 0 \\ 0 & D_{22} \end{bmatrix}$.
-



Simplest way to compute $\text{sign}(A)$:

Newton iteration

[ROBERTS '71]

$$X_{k+1} = \frac{1}{2}(X_k + X_k^{-1}), \quad X_0 = A.$$

Challenge: Method to compute matrix sign function, that

- converges in few iterations,
- exploits pseudosymmetric structure,
- is stable (avoids inverses),
- is suitable for high performance computing.

**Polar decomposition**

[AUTONNE 1902]

Polar decomposition of $A \in \mathbb{C}^{m \times n}$, $m \geq n$:

$$A = UH$$

$U \in \mathbb{C}^{m \times n}$: Orthogonal columns, unique if A has full column rank,

$H \in \mathbb{C}^{n \times n}$: Symmetric positive (semi-)definite, unique.

For **symmetric matrix** A :

$$\text{sign}(A) = U.$$



Generalized polar decomposition (for signature matrix Σ)

[HIGHAM, MEHL, TISSEUR '10]

Σ -Polar decomposition of $A \in \mathbb{C}^{m \times n}$, $m \geq n$:

$$A = WS$$

$W \in \mathbb{C}^{m \times n}$: (Σ, Σ) -Orthogonal columns, unique if A has full column rank,

$S \in \mathbb{C}^{n \times n}$: Pseudosymmetric, $\Lambda(S) \subseteq \mathbb{C}^+ \cup \{0\}$, unique.

Existence in general not guaranteed

For **pseudosymmetric matrix** A :

$$\text{sign}(A) = W.$$

→ Motivation to generalize iteration methods (QDWH and Zolotarev) for polar decomposition.



- The Σ -generalized polar factor of A can be computed via **scaling** iterations
- **Scaling factors** determined by (estimated) bounds on singular values.

1. Suboptimal scaled Newton

$$X_{k+1} = \frac{1}{2}(\mu_k X_k + \mu_k^{-1} \Sigma X_k^{-*} \Sigma), \quad X_0 = sA.$$

- Convergence in 9 steps*.

* Under certain definiteness assumptions



2. Dynamically Weighted Halley (DWH) iteration:

$$X_{k+1} = X_k(a_k I + b_k \Sigma X_k^* \Sigma X_k)(I + c_k \Sigma X_k^* \Sigma X_k)^{-1}, \quad X_0 = sA.$$

- Convergence in 6 steps*.

3. Zolotarev iteration of rank r :

$$X_{k+1} = \hat{C}\left(X_k + \sum_{j=1}^r a_{j,k} X_k (X_k^{*M} X_k + c_{2j-1,k} I)^{-1}\right) \quad X_0 = sA.$$

- Convergence in 2 steps*
- More work each, but parallelizable!

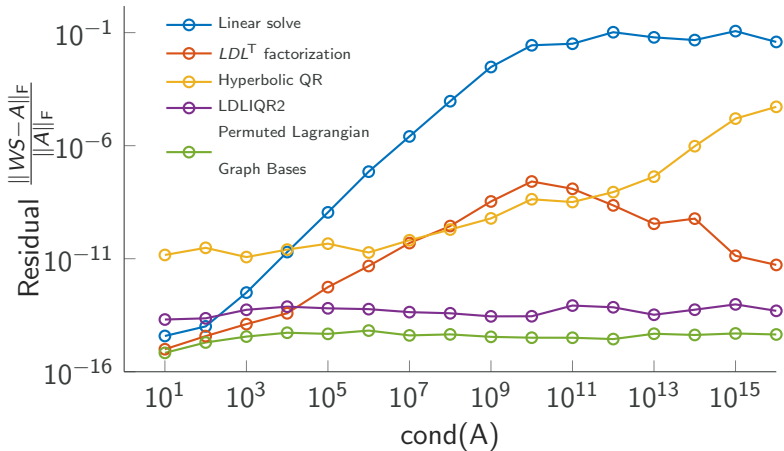
Trade-off stability vs. flop count

We developed various reformulations to avoid e.g. computing matrix inverses and improve stability.

* Under certain definiteness assumptions



Figure: Residuals for different iterations for computing the generalized polar decomposition of pseudosymmetric matrices.





- We extended and developed approaches to exploit *two symmetries* individually.

Hamiltonian



- Transform to skew-symmetric and use ELPA.
- HPC implementation.

Pseudosymmetric



- Generalize spectral divide-and-conquer.
- Mathematically more complex.
- Many algorithmic choices.
- Prototype implementations.

Thank you for your attention!



P. Benner, Y. Nakatsukasa, and C. Penke.

A structure-preserving divide-and-conquer method for pseudosymmetric matrices.

e-print 2203.08900, arXiv, 2022.

Handed in.



P. Benner, Y. Nakatsukasa, and C. Penke.

Stable and efficient computation of generalized polar decompositions.

SIAM J. Matrix Anal. Appl., 2022.



P. Benner and C. Penke.

GR decompositions and their relations to Cholesky-like factorizations.

Proc. Appl. Math. Mech., 20(1):e202000065, 2021.



P. Benner and C. Penke.

Efficient and accurate algorithms for solving the Bethe-Salpeter eigenvalue problem for crystalline systems.

J. Comput. Appl. Math., 2021.



C. Penke, A. Marek, C. Vorwerk, C. Draxl, and P. Benner.

High performance solution of skew-symmetric eigenvalue problems with applications in solving the Bethe-Salpeter eigenvalue problem.

Par. Comp., 96:102639, 2020.



P. Benner, A. Marek, and C. Penke.

Improving the performance of numerical algorithms for the Bethe-Salpeter eigenvalue problem.

Proc. Appl. Math. Mech., 18(1):e201800255, 2018.

