A Structure-Preserving Divide-and-Conquer Method for Pseudosymmetric Matrices

Peter Benner* Yuji Nakatsukasa† Carolin Penke‡

*Max Planck Institute for Dynamics of Complex Technical Systems,
Sandtorstr. 1, 39106 Magdeburg, Germany.
Email: peter.benner@mpi-magdeburg.mpg.de, ORCID: 0000-0003-3362-4103
†Mathematical Institute, University of Oxford,
Oxford, OX2 6GG, UK.
Email: nakatsukasa@maths.ox.ac.uk, ORCID: 0000-0001-7911-1501
‡Max Planck Institute for Dynamics of Complex Technical Systems,
Sandtorstr. 1, 39106 Magdeburg, Germany.
ORCID: 0000-0002-4043-3885

Abstract: We devise a spectral divide-and-conquer scheme for matrices that are self-adjoint with respect to a given indefinite scalar product (i.e. *pseudosymmetic* matrices). The pseudosymmetric structure of the matrix is preserved in the spectral division, such that the method can be applied recursively to achieve full diagonalization. The method is well-suited for structured matrices that come up in computational quantum physics and chemistry. In this application context, additional definiteness properties guarantee a convergence of the matrix sign function iteration within two steps when Zolotarev functions are used. The steps are easily parallelizable. Furthermore, it is shown that the matrix decouples into symmetric definite eigenvalue problems after just one step of spectral division.

Keywords: Matrix Sign Function, Polar Decomposition, Eigenvalue Problem, Structure Preservation, Divide-and-Conquer, Pseudosymmetry

AMS subject classifications: 15A18, 65F15

Novelty statement: The spectral divide-and-conquer methodology is extended such that the structure of pseudosymmetric matrices is preserved. Further results are given regarding the computation of the matrix sign function and subspace bases, in particular when a certain definiteness property holds.

1 Introduction

Given a diagonalizable matrix $A \in \mathbb{K}^{n\times n}$, where $\mathbb{K} \in \{\mathbb{R}, \mathbb{C}\}$, we are interested in full diagonalization, i.e. finding $V \in \mathbb{K}^{n\times n}$, such that

$$V^{-1}AV = D. \tag{1}$$

For $\mathbb{K} = \mathbb{C}$, the matrix D is diagonal and contains the eigenvalues of A as diagonal values. For $\mathbb{K} = \mathbb{R}$, D is block diagonal with blocks of size 1×1, corresponding to real eigenvalues, or 2×2, corresponding to a pair of complex conjugate eigenvalues. The well-established standard approach for solving (1) starts by computing the Schur decomposition of A

$$Q^{\mathsf{T}}AQ = T,$$

where Q is orthogonal (or unitary) and T is (block) upper triangular, via the QR algorithm [24]. The eigenvectors of T are computed via backward substitution or the eigenvectors of A are recovered via inverse iteration [4]. The QR algorithm, however, has proven difficult to parallelize and is not well-suited for computing only parts of the eigenvalue spectrum [5]. This is why spectral divide-and-conquer algorithms were explored as an alternative [5–7, 34]. They are based on the idea of spectral division. A matrix V is found such that

$$V^{-1}AV = \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix}. \tag{2}$$

This is achieved when the first columns of V form a basis of an invariant subspace of A and the remaining columns complement them to form a basis of $\mathbb{K}^n$. Now, the eigenvalue problems of the smaller matrices A_{11} and A_{22} are considered. Repeating this method recursively leads to a *spectral divide-and-conquer* scheme for the triangularization of a matrix.

The required subspace bases are acquired by employing the matrix sign function, which is computed via an iteration. In general, the operation count of spectral divide-and-conquer methods is higher than that of QR based algorithms. This is why optimized implementations that exploit available parallelism are needed.

One direction towards more efficient implementations is to take the given structure of a matrix into account. For example, it is clear that symmetry must be exploited when available. In spectral division (2) exploiting symmetric structure is achieved by finding an orthogonal matrix V. This way, a block-diagonalization is realized instead of the block-triangularization.

This is done in the spectral divide-and-conquer approach presented in [39]. For symmetric matrices, the computation of the matrix sign function can be parallelized particularly well [12, 13, 37], making it competitive with standard approaches in a high performance setting [31, 32]. An important aspect is that spectral divide-and-conquer methods require less communication than QR based approaches. In recent years many efforts have been directed to finding communication-avoiding implementations of essential tools in numerical linear algebra [9]. Spectral divide-and-conquer methods can be implemented using these available building blocks [8]. On more advanced architectures, avoiding communication is more important than avoiding FLOPs in order to minimize the runtime.

In the present work, we extend the spectral divide-and-conquer approach to solve eigenvalue problems of matrices with a more general structure, called pseudosymmetry.

A pseudosymmetric matrix is symmetric up to sign changes of rows or columns. Symmetric matrices are a subset of pseudosymmetric matrices. The complex analogue is called a pseudo-Hermitian matrix. In the following, statements are formulated for (pseudo-)symmetric matrices, but also hold for (pseudo-)Hermitian matrices.

Efforts to exploit pseudosymmetric structure led to the development of the HR algorithm [17, 20, 21]. It generalizes the symmetric QR algorithm and is motivated by the following observation. A generalized eigenvalue problem with symmetric matrices

$$Ax = \lambda Bx, \qquad A = A^{\mathsf{T}}, \quad B = B^{\mathsf{T}}, \tag{3}$$

can be cast into a pseudosymmetric standard eigenvalue problem. Neither A nor B need to be positive definite. If B is nonsingular, it has a decomposition $B = R^{\mathsf{T}}\Sigma R$, where Σ is a diagonal matrix with 1 or -1 as diagonal values. Then (3) is equivalent to

$$\Sigma R^{-\mathsf{T}} A R^{-1} y = \lambda y, \qquad y = Rx.$$

$\Sigma R^{-\mathsf{T}} A R^{-1}$ is clearly a pseudosymmetric matrix. Diagonal matrices with ± 1 as diagonal values are called *signature matrices* in the following.

The QR algorithm computes its results with great accuracy because only (implicit) orthogonal transformations are involved. This is not true for the HR algorithm, which uses $(\Sigma, \hat{\Sigma})$-orthogonal matrices instead (also called pseudoorthogonal) [53]. A $(\Sigma, \hat{\Sigma})$-orthogonal matrix H, where Σ and $\hat{\Sigma}$ are two signature matrices, fulfills $H^{\mathsf{T}}\Sigma H = \hat{\Sigma}$. They are used to transform a matrix to upper triangular form, similar to the QR decomposition.

On top of that, the HR algorithm suffers from the same drawbacks as the QR algorithm in a high-performance environment: It is hard to parallelize and not communication-avoiding, as explained above.

The spectral divide-and-conquer method developed in this work presents a promising alternative. It can be parallelized and only relies on building blocks for which communication-avoiding implementations exist. It can be used to compute only parts of the spectrum with reduced computational effort. Stability concerns are addressed by employing alternatives to the HR decomposition in the computation of the matrix sign function, presented in [14].

Our main motivation stems from computational quantum science. Time-dependent density functional theory in the linear-response regime (TDDFT) and the Bethe-Salpeter approach are two competing methods for computing excited states of solids or molecules in the context of a perturbed induced density matrix [40, 44].

The Bethe-Salpeter equation derived from many body perturbation theory [11, 40, 46] and the Casida equation derived from TDDFT for molecules [23] lead to pseudosymmetric eigenvalue problems in their discretized form: The matrices become symmetric when multiplied with $\Sigma = \operatorname{diag}(I, -I)$, where I denotes the identity matrix.

Due to physical constraints, these matrices have another property, which is exploited in our proposed algorithm. The symmetric matrix resulting from multiplication with Σ is positive definite. It was shown in [14] that for these matrices the proposed iterations have the same favorable convergence properties as in the symmetric setting. Furthermore, we prove in this work that the first round of spectral division decouples the problem into a positive and a negative definite symmetric matrix.

Pseudosymmetric matrices with these definiteness properties also play a role in describing damped oscillations of linear systems [49].

The remainder of the paper is structured as follows. Section 2 introduces scalar products and related notation which form basic concepts used throughout the paper. This refers to a generalization of symmetry and orthogonality with respect to a scalar product defined by signature matrices. $(\Sigma, \hat{\Sigma})$-orthogonal matrices ensure the preservation of structure in the spectral division for pseudosymmetric matrices. Furthermore, the matrix sign function is introduced as the central tool for spectral division. Section 3 explains the idea of spectral divide-and-conquer methods and presents a generalization of this approach for pseudosymmetric matrices. The acquisition of $(\Sigma, \hat{\Sigma})$-orthogonal representations of invariant subspaces is essential for structure preservation in the spectral division. In Section 4, we point out a link between QR decompositions of symmetric projection matrices (describing orthogonal projections) and Cholesky factorizations. This link exists analogously for pseudosymmetric projection matrices and the LDL^{T} factorization. We use this insight to compute required basis representations via the LDL^{T} factorization. Matrices arising in the application of electronic excitation have additional definiteness properties. Section 5 shows how these are exploited in the presented algorithms. The computation of proper basis representations simplifies to a partial Cholesky factorization (Section 5.2) and the computation of the matrix sign function can be accelerated using Zolotarev functions (Section 5.3). Section 6 presents the results of numerical experiments regarding the new method. Conclusions and further research directions are given in Section 7.

2 Preliminaries

Following [27] and [33], we give some basic results regarding non-Euclidian scalar products. A nonsingular matrix M defines a scalar product on $\mathbb{K}^n$, where $\mathbb{K} \in \{\mathbb{C}, \mathbb{R}\}$, that is a bilinear or sesquilinear form $\langle \cdot, \cdot \rangle_M$, given by

$$\langle x, y \rangle_M = \begin{cases} x^{\mathsf{T}} M y \text{ for bilinear forms,} \\ x^H M y \text{ for sesquilinear forms,} \end{cases}$$

for $x, y \in \mathbb{K}^n$. We use $\cdot^*$ throughout the paper to indicate transposition $\cdot^{\mathsf{T}}$ or conjugated transposition $\cdot^{\mathsf{H}}$, depending on whether a bilinear or sesquilinear form is given.

For a matrix $A \in \mathbb{K}^{n\times n}$, $A^{\star_M} \in \mathbb{K}^{n\times n}$ denotes the adjoint with respect to the scalar product defined by M. This is a uniquely defined matrix satisfying the identity

$$\langle Ax, y \rangle_M = \langle x, A^{\star_M} y \rangle_M$$

for all $x, y \in \mathbb{K}^n$. We call $A^{\star_M}$ the M-adjoint of A and it holds

$$A^{\star_M} = M^{-1}A^*M. \tag{4}$$

A matrix S is called (M-)self-adjoint (with respect to the scalar product induced by M) if $S = S^{\star_M}$.

Similar concepts are available for rectangular matrices. As two vector spaces of different dimensions now play a role, two distinct scalar products are considered. We give some clarifying notation following [28]. For a matrix $A \in \mathbb{K}^{m\times n}$, $A^{\star_{M,N}} \in \mathbb{K}^{n\times m}$ denotes the adjoint with respect to the two scalar products defined by the nonsingular matrices $M \in \mathbb{K}^{m\times m}$, $N \in \mathbb{K}^{n\times n}$. This matrix is uniquely defined by the identity

$$\langle Ax, y\rangle_M = \langle x, A^{\star_{M,N}}y\rangle_N$$

for all $x \in \mathbb{K}^n$, $y \in \mathbb{K}^m$. We call $A^{\star_{M,N}}$ the (M, N)-adjoint of A and it holds

$$A^{\star_{M,N}} = N^{-1}A^*M.$$

A matrix $H \in \mathbb{K}^{m\times n}$ is called (M, N)-orthogonal when

$$H^{\star_{M,N}}H = I_n.$$

A prerequisite for this is $n \le m$, which is the case for the applications discussed later. For (M, N)-orthogonal matrices, we define

$$H^\dagger := H^{\star_{M,N}} = N^{-1}H^*M. \tag{5}$$

Here, (5) gives the (M, N)-Moore-Penrose pseudoinverse discussed in [28]. This notion generalizes the well-known Moore-Penrose pseudoinverse, which is achieved by setting $M = I_m$ and $N = I_n$.

Our proposed methods rely on the matrix sign function [26, 30, 43]. Let $A \in \mathbb{K}^{n\times n}$ be a nonsingular matrix with no imaginary eigenvalues with Jordan canonical form

$$A = Z\begin{bmatrix} J_- & \\ & J_+ \end{bmatrix}Z^{-1},$$

where $J_- \in \mathbb{K}^{m\times m}$ contains the Jordan blocks associated with the eigenvalues having a negative real part and $J_+ \in \mathbb{K}^{p\times p}$ contains the Jordan blocks associated with the eigenvalues having a positive real part. Then the matrix sign function of A is defined as

$$\operatorname{sign}(A) := Z\begin{bmatrix} -I_m & \\ & I_p \end{bmatrix}Z^{-1}. \tag{6}$$

It follows from (6) that the matrix sign function can be used to acquire projectors onto invariant subspaces associated with the subset of eigenvalues that have positive and negative real parts, respectively.

Lemma 1. *Let $A \in \mathbb{K}^{n\times n}$ be nonsingular with no imaginary eigenvalues and $S = \operatorname{sign}(A)$. Then $P_+ = \frac{1}{2}(I_n + S)$ and $P_- = \frac{1}{2}(I_n - S)$ are projectors onto the invariant subspaces associated with eigenvalues in the open right and open left half-plane, respectively.*

In order to acquire projections onto invariant subspaces associated with other eigenvalue subsets, we can use the matrix sign function of a shifted $A + \sigma I$. Another possibility is to transform A before computing the matrix sign function in order to acquire subspaces associated with almost arbitrary regions of the eigenvalue spectrum [7]. What makes the matrix sign function useful is that there exist iterative methods for its computation [26, 29]. Among the simplest is Newton's iteration to find the roots of $f(x) = x^2 - 1$,

$$X_{k+1} = \frac{1}{2}(X_k + X_k^{-1}), \quad X_0 = A. \tag{7}$$

Our iteration of choice is based on Zolotarev functions and discussed in Section 5.3.

3 Structure preserving divide-and-conquer methods

The property of the matrix sign function to acquire invariant subspaces was used in the original paper [43] to solve algebraic Riccati equations. Later, it was used as a building block to devise parallelizable methods for eigenvalue computations of nonsymmetric matrices [5, 13, 48]. In [39] a spectral divide-and-conquer algorithm for symmetric matrices is formulated, based on the relation between the matrix sign function and the polar decomposition. In this section, we generalize this approach to pseudosymmetric matrices. They are defined using signature matrices, which are diagonal matrices $\Sigma = \operatorname{diag}(\sigma_1, \ldots, \sigma_n)$, where $\sigma_i \in \{1, -1\}$ for $i = 1, \ldots, n$.

Definition 1. *A matrix $A \in \mathbb{K}^{n\times n}$ is called* pseudosymmetric (pseudo-Hermitian) *if there exists a signature matrix Σ, such that A is self-adjoint with respect to the bilinear form (sesquilinear form) induced by Σ.*

Definition 1 is equivalent to ΣA (or $A\Sigma$) being symmetric. Essentially, a pseudosymmetric matrix is symmetric up to sign changes of certain rows (or columns). This definition is slightly different than the one given, e.g., in [33], as we allow any signature matrix and not just $\Sigma_{p,q} = \begin{bmatrix} I_p & \\ & -I_q \end{bmatrix}$.

In Section 3.1 we outline the general idea of spectral division, which reduces a large eigenvalue problem to two smaller ones. Recursively applying this technique yields parallelizable methods for acquiring all eigenvalues and eigenvectors. Section 3.2 recounts how a symmetric structure can be preserved in this context. The same line of argument is applied to pseudosymmetric matrices in Section 3.3.

3.1 General spectral divide-and-conquer

It is a well-known concept to use invariant subspaces of a matrix to block-triangularize it with a similarity transformation. In the following we focus on real matrices, but everything extends to complex matrices. For real matrices we end up with 2×2 matrix blocks on the diagonal for complex eigenvalues, whereas for complex matrices, this is unnecessary.

Theorem 2 ([48])**.** *Let $A \in \mathbb{R}^{n\times n}$ and $V_1 \in \mathbb{R}^{n\times k}$ be a basis for an invariant subspace of A and $V = \begin{bmatrix} V_1 & V_2 \end{bmatrix} \in \mathbb{R}^{n\times n}$ have full rank. Then*

$$V^{-1}AV = \begin{bmatrix} A_{11} & A_{21} \\ 0 & A_{22} \end{bmatrix}, \qquad A_{11} \in \mathbb{R}^{k\times k},\ A_{22} \in \mathbb{R}^{(n-k)\times(n-k)}.$$

Recursively applying the idea of Theorem 2 with shifts leads to a divide-and-conquer scheme, given in Algorithm 1.

This algorithm serves as a prototype for structure preserving methods developed in the next subsections. The key idea is to choose the subspace basis in Step 4 in a way that preserves the structure in the spectral division.

3.2 Symmetric spectral divide-and-conquer

In this section we consider the symmetric eigenvalue problem, i.e. $A = A^\mathsf{T}$. A structure-preserving method requires the spectral division $V^{-1}AV$ to be symmetric. This is fulfilled by orthogonal matrices, i.e. for matrices fulfilling $V^{-1} = V^\mathsf{T}$. A structure-preserving variant of Theorem 2 for symmetric matrices is given in the following.

Theorem 3. *Let $A = A^\mathsf{T} \in \mathbb{R}^{n\times n}$ and $V_1 \in \mathbb{R}^{n\times k}$ be a basis of an invariant subspace of A and $V = \begin{bmatrix} V_1 & V_2 \end{bmatrix}$ be orthogonal. Then*

$$V^{-1}AV = V^\mathsf{T}AV = \begin{bmatrix} A_{11} & 0 \\ 0 & A_{22} \end{bmatrix}, \qquad A_{11} = A_{11}^\mathsf{T} \in \mathbb{R}^{k\times k},\ A_{22} = A_{22}^\mathsf{T} \in \mathbb{R}^{(n-k)\times(n-k)}.$$

The symmetric version of Algorithm 1 follows immediately as Algorithm 2.

Due to the symmetry of A and by restricting the subspace basis to be orthogonal, this can become a highly viable method. For symmetric A, $\operatorname{sign}(A)$ can be computed in a stable way via

Algorithm 1 Unstructured spectral divide-and-conquer

Input: $A \in \mathbb{R}^{n\times n}$

Output: V, T such that $V^{-1}AV = T$ is block-upper triangular with blocks no larger than 2×2 on the diagonal.

1: Stop if A is of size 1×1 or 2×2 with a complex 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 $S = \operatorname{sign}(A - \sigma I)$ via an iteration.
4: Compute a basis V_+ of $\operatorname{range}(S + I)$ and V_- such that $V_0 = \begin{bmatrix} V_+ & V_- \end{bmatrix}$ has full rank. Then

$$V_0^{-1} A V_0 = \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix}.$$

5: Repeat spectral divide-and-conquer for A_{11}, i.e. find V_1 such that $V_1^{-1} A_{11} V_1 = T_{11}$ is block-upper triangular.
6: Repeat spectral divide-and-conquer for A_{22}, i.e. find V_2 such that $V_2^{-1} A_{22} V_2 = T_{22}$ is block-upper triangular.
7: $V \leftarrow V \begin{bmatrix} V_1 & 0 \\ 0 & V_2 \end{bmatrix}$, $T \leftarrow \begin{bmatrix} T_{11} & V_1^{-1} A_{12} V_2 \\ 0 & T_{22} \end{bmatrix}$.

Algorithm 2 Symmetric spectral divide-and-conquer

Input: $A = A^\mathsf{T} \in \mathbb{R}^{n\times n}$

Output: Orthogonal V, diagonal D such that $V^\mathsf{T} A V = D$.

1: Stop if A is of size 1×1.
2: Find shift σ such that $A - \sigma I$ has positive and negative eigenvalues and no zero eigenvalues.
3: Compute $S = \operatorname{sign}(A - \sigma I)$ via an iteration.
4: Compute a basis V_+ of $\operatorname{range}(S + I)$ and V_- such that $V_0 = \begin{bmatrix} V_+ & V_- \end{bmatrix}$ is orthogonal. Then

$$V_0^\mathsf{T} A V_0 = \begin{bmatrix} A_{11} & 0 \\ 0 & A_{22} \end{bmatrix}, \quad A_{11} = A_{11}^\mathsf{T},\ A_{22} = A_{22}^\mathsf{T}.$$

5: Repeat spectral divide-and-conquer for A_{11}, i.e. find V_1 such that $V_1^\mathsf{T} A_{11} V_1 = D_{11}$ is diagonal.
6: Repeat spectral divide-and-conquer for A_{22}, i.e. find V_2 such that $V_2^\mathsf{T} A_{22} V_2 = D_{22}$ is diagonal.
7: $V \leftarrow V_0 \begin{bmatrix} V_1 & 0 \\ 0 & V_2 \end{bmatrix}$, $D \leftarrow \begin{bmatrix} D_{11} & 0 \\ 0 & D_{22} \end{bmatrix}$.

the QDWH iteration [36, 38] or the Zolotarev iteration [37]. The basis extraction can be done by performing a rank-revealing QR decomposition or a subspace iteration [39], if pivoting is considered too expensive.

3.3 Pseudosymmetric spectral divide-and-conquer

We now extend Section 3.2 to pseudosymmetric matrices. The role of structure-preserving similarity transformations was played by orthogonal matrices in Section 3.2. For pseudosymmetric matrices this role is played by $(\Sigma, \hat{\Sigma})$-orthogonal matrices.

Lemma 4. *If $V \in \mathbb{R}^{m\times n}$, $m \geq n$, is a $(\Sigma, \hat{\Sigma})$-orthogonal matrix and $A \in \mathbb{R}^{m\times m}$ is pseudosymmetric with respect to Σ, i.e. $\Sigma A = A^\mathsf{T}\Sigma$. Then $\hat{A} = V^\dagger A V$ is pseudosymmetric with respect to $\hat{\Sigma}$, i.e. $\hat{\Sigma}\hat{A} = \hat{A}^\mathsf{T}\hat{\Sigma}$.*

Proof. With $V^\dagger = \hat{\Sigma} V^\mathsf{T} \Sigma$, $\Sigma V = (V^\dagger)^\mathsf{T}\hat{\Sigma}$, $\Sigma^2 = I_m$ and $\hat{\Sigma}^2 = I_n$ we have

$$\hat{\Sigma}(V^\dagger A V) = V^\mathsf{T}\Sigma A V = V^\mathsf{T} A^\mathsf{T}\Sigma V = V^\mathsf{T} A^\mathsf{T}(V^\dagger)^\mathsf{T}\hat{\Sigma} = (V^\dagger A V)^\mathsf{T}\hat{\Sigma}.$$

□

Algorithm 3 Pseudosymmetric spectral divide-and-conquer

Input: Signature matrix Σ, pseudosymmetric A with respect to Σ, i.e. $\Sigma A = (\Sigma A)^\mathsf{T}$.
Output: Signature matrix $\hat{\Sigma}$, and $(\Sigma, \hat{\Sigma})$-orthogonal V such that $V^\dagger A V = D$ 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 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 $S = \operatorname{sign}(A - \sigma I)$ via an iteration.
4: Compute a basis V_+ of range$(S + I)$ and V_- such that $V_0 = \begin{bmatrix} V_+ & V_- \end{bmatrix}$ is (Σ, Σ_0)-orthogonal with $\Sigma_0 = \begin{bmatrix} \Sigma_+ & \\ & \Sigma_- \end{bmatrix}$. Then

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

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

What $(\Sigma, \hat{\Sigma})$-orthogonal matrices have in common with orthogonal matrices is that their (pseudo-) inverses can be easily computed via (5) in the form of

$$V^\dagger = \hat{\Sigma} V^\mathsf{T} \Sigma.$$

For square matrices it holds $V^{-1} = V^\dagger$ and $V^\dagger A V$ constitutes a similarity transformation.

Methods for computing these matrices include the HR decomposition [19] and methods described in [14]. They prescribe Σ and yield $\hat{\Sigma}$ and the $(\Sigma, \hat{\Sigma})$-orthogonal matrix H. We do not actually care about how $\hat{\Sigma}$ looks exactly, as long as it is a signature matrix. This way, pseudosymmetry as we defined it in Definition 1, not being bound to a specific Σ, is preserved. These kind of matrices, i.e. $(\Sigma, \hat{\Sigma})$-orthogonal matrices, where $\hat{\Sigma}$ does not matter, are sometimes called "hyperexchange" (e.g. in [51, 52]).

These observations can be used to formulate a pseudosymmetric variant of Algorithm 1, given in Algorithm 3. In Step 4, we apply spectral division (Theorem 2) and preserve pseudosymmetric structure (Lemma 2). The result is a block-triangular matrix that is also pseudosymmetric and therefore block-diagonal. The property preserved in the spectral division is the pseudosymmetry. This means that Σ does not stay fixed, but is permuted and truncated in each division step.

4 Computing $(\Sigma, \hat{\Sigma})$-orthogonal representations of subspaces

Symmetric spectral divide-and-conquer methods rely on variants of the QR decomposition. The natural generalization in the indefinite context is the hyperbolic QR decomposition.

Theorem 5 (The hyperbolic QR decomposition [19])**.** *Let $\Sigma \in \mathbb{R}^{m \times m}$ be a signature matrix, $A \in \mathbb{R}^{m \times n}$, $m \geq n$. Suppose all the leading principal submatrices of $A^\mathsf{T} \Sigma A$ are nonsingular. Then there exists a permutation P, a signature matrix $\hat{\Sigma} = P^\mathsf{T} \Sigma P$, a $(\Sigma, \hat{\Sigma})$-orthogonal matrix $H \in \mathbb{R}^{m \times n}$ (i.e. $H^\mathsf{T} \Sigma H = \hat{\Sigma}$), and an upper triangular matrix $R \in \mathbb{R}^{n \times n}$, such that*

$$A = H \begin{bmatrix} R \\ 0 \end{bmatrix}.$$

Similar to the orthogonal QR decomposition, it can be computed by applying transformations that introduce zeros below the diagonal, column by column. Details can e.g. be found in [53].

In [47], the indefinite QR decomposition is presented, which improves stability by allowing 2×2 blocks on the diagonal of R and additional pivoting. This variant can also be computed via the (pivoted) LDL^T decomposition of $A^\mathsf{T}\Sigma A$; a link which was exploited in [14] and [16]. There, the stability is improved by applying this method twice.

In the context of this work we aim to compute the indefinite QR decomposition of a pseudosymmetric projection matrix. We will see that in this special case, an indefinite QR decomposition can be computed via the LDL^T decomposition without the need to form $A^\mathsf{T}\Sigma A$. We do not make any statement about the stability of the proposed computations, as these considerations go beyond the scope of this paper, but make empirical observations in the numerical experiments presented in Section 6.

We start with an observation regarding the symmetric divide-and-conquer method. Here, the matrix sign function computes a symmetric projection matrix, representing an orthogonal projection.

Lemma 6. *Let $P \in \mathbb{R}^{n\times n}$ be an orthogonal projection matrix, i.e. $P^2 = P$ and $P = P^\mathsf{T}$, with rank r. Let $R^\mathsf{T}R = P$, where $R \in \mathbb{R}^{r\times n}$ has full row rank. Then R^T has orthogonal columns, i.e. $RR^\mathsf{T} = I_r$.*

Proof. Because P is positive semi-definite, a low-rank factorization $R^\mathsf{T}R = P$ exists. From $P = P^2$ follows $R^\mathsf{T}R = R^\mathsf{T}RR^\mathsf{T}R$ and therefore $RR^\mathsf{T} = I_r$. □

R can for example be chosen to be upper triangular, i.e. forming a low-rank Cholesky decomposition of P. In that case, $R^\mathsf{T}R = P$ is a thin QR decomposition. Then Lemma 6 states that for projection matrices attained via the matrix sign function, the low-rank Cholesky and the thin QR decomposition are equivalent.

Let $P_+ = \frac{1}{2}(I_n + \operatorname{sign}(A))$ be the projection on the subspace of A associated with positive eigenvalues. The advantage of computing the (full) QR decomposition $\begin{bmatrix} Q_+ & Q_- \end{bmatrix} \begin{bmatrix} R \\ 0 \end{bmatrix}$ is that we immediately get a basis Q_- for the complementing subspace, associated with negative eigenvalues. The Cholesky factorization applied in the sense of Lemma 6 can only yield a thin QR decomposition. However, the same procedure can be applied to $P_- = \frac{1}{2}(I_n - \operatorname{sign}(A))$. The two thin QR decompositions can be combined to form a full one. Indeed, let Q_+ and Q_- be acquired from P_+ and P_- via Lemma 6. The identities $Q_+^\mathsf{T}Q_+ = I$ and $Q_-^\mathsf{T}Q_- = I$ follow immediately from the orthogonality proven in the lemma. From $P_+ = Q_+Q_+^\mathsf{T}$ follows ${Q_+}^\mathsf{T} = Q_+^\mathsf{T}P_+$ and from $P_- = Q_-Q_-^\mathsf{T}$ follows $Q_- = P_-Q_-$. From the definition of the projectors in Lemma 1 we have $P_+P_- = 0$ and therefore $Q_+^\mathsf{T}Q_- = Q_+P_+^\mathsf{T}P_-Q_- = 0$.

Algorithm 4 shows how Lemma 6 can be used to compute an orthogonal representation of an invariant subspace of a symmetric matrix. In Step 3 we use the trace of a projection matrix to determine its rank. For badly conditioned matrices, pivoting could be included in the computation of the Cholesky factorization. Lemma 6 does not assume the triangular shape of R to show that its rows are orthogonal.

In the symmetric context, computing the QR decomposition like this does not have an obvious benefit over computing a QR decomposition the standard way. However, it can be generalized to the indefinite case. Here, an LDL^T decomposition can be used instead of a hyperbolic QR decomposition, which is much more widely used. Established algorithms and highly-optimized implementations are available and ready to use, e.g. in MATLAB as `ldl` command. Details are given in the following theorem.

Theorem 7. *Σ is a given signature matrix, $P \in \mathbb{R}^{n\times n}$ is a projection matrix and pseudosymmetric with respect to Σ, i.e. $P^2 = P$ and $\Sigma P\Sigma = P^\mathsf{T}$, with rank r. Let $R^\mathsf{T}\hat{\Sigma}R = \Sigma P$, where $R \in \mathbb{R}^{r\times n}$, be a scaled low-rank LDL^T factorization, where R has full row rank and $\hat{\Sigma} \in \mathbb{R}^{r\times r}$ is another signature matrix. Then R^T is $(\Sigma, \hat{\Sigma})$-orthogonal,, i.e. $R\Sigma R^\mathsf{T} = \hat{\Sigma}$, and $HR = P$ with $H = \Sigma R^\mathsf{T}\hat{\Sigma}$ is a decomposition of P, where H is $(\Sigma, \hat{\Sigma})$-orthogonal.*

Proof. With $P = \Sigma R^\mathsf{T}\hat{\Sigma}R$ and $P = P^2$ it follows $\Sigma R^\mathsf{T}\hat{\Sigma}R = \Sigma R^\mathsf{T}\hat{\Sigma}R\Sigma R^\mathsf{T}\hat{\Sigma}R$ and therefore $\hat{\Sigma} = \hat{\Sigma}R\Sigma R^\mathsf{T}\hat{\Sigma}$ and finally

$$\hat{\Sigma} = R\Sigma R^\mathsf{T}. \tag{8}$$

Algorithm 4 Compute orthogonal invariant subspace representations of a symmetric matrix via Cholesky.

Input: $A = A^{\mathsf{T}} \in \mathbb{R}^{n\times n}$ nonsingular.

Output: An orthogonal basis $Q = \begin{bmatrix} Q_+ & Q_- \end{bmatrix}$, where Q_+ is a basis of the invariant subspace of A associated with positive eigenvalues, Q_- is a basis of the invariant subspace of A associated with negative eigenvalues.

1: $S \leftarrow \operatorname{sign}(A)$.
2: $P_+ \leftarrow \frac{1}{2}(I_n + S)$.
3: Compute $\operatorname{rank}(P_+) =: r_+ \leftarrow \operatorname{tr}(P_+)$.
4: $Q_{1,+} \leftarrow \texttt{chol}(P_+(1:r_+, 1:r_+)$.
5: $Q_+ \leftarrow \begin{bmatrix} Q_{1,+} \\ P_+(r_+ + 1:n, 1:r_+)Q_{1,+}^{-\mathsf{T}} \end{bmatrix}$.
6: $P_- \leftarrow \frac{1}{2}(I_n - S)$.
7: Compute $\operatorname{rank}(P_-) =: r_- \leftarrow n - r_+$.
8: $Q_{1,-} \leftarrow \texttt{chol}(P_-(1:r_-, 1:r_-)$.
9: $Q_- \leftarrow \begin{bmatrix} Q_{1,-} \\ P_+(r_- + 1:n, 1:r_-)Q_{1,-}^{-\mathsf{T}} \end{bmatrix}$.

We used $\hat{\Sigma}^2 = I_r$. Equation (8) is equivalent to $H := R^{\dagger} = \Sigma R^{\mathsf{T}} \hat{\Sigma}$ being $(\Sigma, \hat{\Sigma})$-orthogonal: $H^{\mathsf{T}} \Sigma H = \hat{\Sigma}$. We therefore have a decomposition $P = \Sigma R^{\mathsf{T}} \hat{\Sigma} R = HR$. □

If R in Theorem 7 is computed with the Bunch-Kaufman algorithm [18] (e.g. MATLAB `ldl`), it can be a permuted block-triangular matrix and stability can be improved. Then $P = HR$ is not a hyperbolic QR decomposition in the strict sense given in Theorem 5. This is not important here, as we are only interested in the subspace given by H. The indefinite variant of Algorithm 4 is given in Algorithm 5.

In contrast to the MATLAB function `chol`, `ldl` is not affected by singular matrices, such as the given projectors. This is why steps 5 and 9 in Algorithm 4 do not have a correspondence in Algorithm 5. The Cholesky-based algorithm (Algorithm 4) computes the Cholesky factorization of the upper left block and expands it in order to get a low-rank version. The LDL^{T}-based algorithm (Algorithm 5) on the other hand computes an LDL^{T} decomposition of the whole matrix, which we then truncate in Steps 6 and 11.

5 Definite pseudosymmetric matrices

In this section we consider pseudosymmetric matrices with an additional property. We call a pseudosymmetric matrix A with respect to a signature matrix Σ *definite* if ΣA is positive definite.

The Bethe-Salpeter equation (BSE) approach is a state-of-the art method for computing optical properties of materials and molecules, derived from many-body perturbation theory. After appropriate discretization, eigenvalues and eigenvectors of a complex structured matrix

$$H_{\mathrm{BSE}} = \begin{bmatrix} A_{\mathrm{BSE}} & B_{\mathrm{BSE}} \\ -B_{\mathrm{BSE}}^{\mathsf{H}} & -A_{\mathrm{BSE}}^{\mathsf{T}} \end{bmatrix}, \qquad A_{\mathrm{BSE}} = A_{\mathrm{BSE}}^{\mathsf{H}}, \quad B_{\mathrm{BSE}} = B_{\mathrm{BSE}}^{\mathsf{T}} \tag{9}$$

are sought [40]. A similar eigenvalue problem arises when molecules are considered within time-dependent density functional theory in the linear response regime. Here, the Casida equation can be recast into an eigenvalue problem of the real matrix

$$H_{\mathrm{Cas}} = \begin{bmatrix} A_{\mathrm{Cas}} & B_{\mathrm{Cas}} \\ -B_{\mathrm{Cas}} & -A_{\mathrm{Cas}} \end{bmatrix}, \qquad A_{\mathrm{Cas}} = A_{\mathrm{Cas}}^{\mathsf{T}}, \quad B_{\mathrm{Cas}} = B_{\mathrm{Cas}}^{\mathsf{T}}. \tag{10}$$

Considering a Bethe-Salpeter approach within Hartree-Fock theory for molecules leads to a matrix with the same structure [11].

Algorithm 5 Compute hyperbolic invariant subspace representations of a pseudosymmetric projection matrix via LDL^{T}

Input: Signature matrix Σ, $A = \Sigma A^{\mathsf{T}}\Sigma \in \mathbb{R}^n$ without purely imaginary eigenvalues.
Output: A signature matrix $\hat{\Sigma}$, which is a permuted variant of Σ, a $(\Sigma, \hat{\Sigma})$-orthogonal basis $Q = \begin{bmatrix} Q_+ & Q_- \end{bmatrix}$, i.e. $Q^{\mathsf{T}}\Sigma Q = \hat{\Sigma}$, where Q_+ is a basis of the invariant subspace of A associated with eigenvalues with positive real part, Q_- is a basis of the invariant subspace of A associated with eigenvalues with negative real part.

1: $S \leftarrow \operatorname{sign}(A)$.
2: $P_+ \leftarrow \frac{1}{2}(I_n + S)$.
3: Compute $\operatorname{rank}(P_+) =: r_+ \leftarrow \operatorname{tr}(P_+)$.
4: $[L_+, D_+] \leftarrow \texttt{ldl}(\Sigma P_+)$.
5: Diagonalize D_+ if it has blocks on the diagonal: $[V_+, D_+] \leftarrow \texttt{eig}(D_+)$, such that $D_+(1:r_+, 1:r_+)$ contains the nonzero diagonal values of D_+.
6: $R_+ \leftarrow (L_+V_+(:, 1:r_+)D_+(1:r_+, 1:r_+)^{\frac{1}{2}})^{\mathsf{T}}$, $\quad \hat{\Sigma}_+ \leftarrow \operatorname{sign}(D_+(1:r_+, 1:r_+))$.
7: $P_- \leftarrow \frac{1}{2}(I_n - S)$.
8: Compute $\operatorname{rank}(P_-) =: r_- \leftarrow n - r_+$.
9: $[L_-, D_-] \leftarrow \texttt{ldl}(\Sigma P_-)$.
10: Diagonalize D_- if it has blocks on the diagonal: $[V_-, D_-] \leftarrow \texttt{eig}(D_-)$, such that $D_-(1:r_-, 1:r_-)$ contains the nonzero diagonal values of D_-.
11: $R_- \leftarrow (L_-V_-(:, 1:r_-)D_-(1:r_-, 1:r_-)^{\frac{1}{2}})^{\mathsf{T}}$, $\quad \hat{\Sigma}_- \leftarrow \operatorname{sign}(D_-(1:r_-, 1:r_-))$.
12: $\hat{\Sigma} \leftarrow \operatorname{diag}(\Sigma_+, \Sigma_-)$.
13: $Q_+ \leftarrow \Sigma R_+^{\mathsf{T}}\hat{\Sigma}$, $\quad Q_- \leftarrow \Sigma R_-^{\mathsf{T}}\hat{\Sigma}$.

For crystalline solids, the periodic structure can be exploited and with a proper choice of basis functions the resulting matrix has the form [45]

$$H_{\mathrm{BSE},2} = \begin{bmatrix} A_{\mathrm{BSE},2} & B_{\mathrm{BSE},2} \\ -B_{\mathrm{BSE},2} & -A_{\mathrm{BSE},2} \end{bmatrix}, \qquad A_{\mathrm{BSE},2} = A_{\mathrm{BSE},2}^{\mathsf{H}}, \quad B_{\mathrm{BSE},2} = B_{\mathrm{BSE},2}^{\mathsf{H}}. \tag{11}$$

The setup (11) is essentially a complex version of (10). A more detailed analysis of the special structure in (9) and (11) is given in [16].

All of these matrices are obviously pseudosymmetric with respect to $\Sigma = \operatorname{diag}(I, -I)$. Furthermore, they are typically definite, i.e. ΣH is positive definite for any H defined in (9), (10) or (11).

5.1 Decoupling the indefinite eigenvalue problem into two symmetric definite problems

In the following, we explain how the spectral divide-and-conquer algorithm described in Section 3.3 simplifies greatly for definite pseudosymmetric matrices. Essentially, the problem can be reduced to two Hermitian positive definite eigenvalue problems after just one spectral division step.

As a first result, we present the following theorem, clarifying the spectral structure of definite pseudosymmetric matrices. It is an extension of Theorem 5 in [16], additionally clarifying the structure of the eigenvectors, and a more general variant of Theorem 3 in [46]. Our version is independent of the additional structure of Bethe-Salpeter matrices given in (9). It can be proven in a similar fashion relying on the simultaneous diagonalization of ΣA and Σ.

Theorem 8. *Let $A \in \mathbb{K}^{n\times n}$ be a definite pseudosymmetric matrix with respect to Σ, where Σ has p positive and $n-p$ negative diagonal entries. Then A has only real, nonzero eigenvalues, of which p are positive and $n-p$ are negative. There is an eigenvalue decomposition $AV = V\Lambda$, $\Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_n)$, where $\lambda_1, \ldots, \lambda_p > 0$, $\lambda_{p+1}, \ldots, \lambda_n < 0$, such that*

$$V^*\Sigma V = \begin{bmatrix} I_p & \\ & -I_{n-p} \end{bmatrix}. \tag{12}$$

Proof. As ΣA is positive definite, and Σ is symmetric, they can be diagonalized simultaneously (see [24], Corollary 8.7.2), i.e. there is a nonsingular $X \in \mathbb{C}^{n\times n}$ s.t. $X^{\mathsf{H}}\Sigma A X = I_n$, and $X^{\mathsf{H}}\Sigma X = \Lambda^{-1} \in \mathbb{R}^{n\times n}$, where $\Lambda^{-1} = \operatorname{diag}\left(\lambda_1^{-1}, \ldots, \lambda_n^{-1}\right)$ gives the eigenvalues of the matrix pencil $\Sigma - \lambda\Sigma A$. It follows from Sylvester's law of inertia that Λ^{-1} has p positive and $n-p$ negative values. We have $X^{-1}AX = \Lambda$, i.e. A is diagonalizable and Λ^{-1} contains the eigenvalues of A. The columns of X can be arranged, such that the positive eigenvalues are given in the upper left part of Λ and the negative ones are given in the lower right part. X can be scaled in form of $V := X|\Lambda|^{-\frac{1}{2}}$, where $|\cdot|$ denotes the entry-wise absolute value, such that (12) holds. □

For pseudosymmetric matrices that are definite, the structure-preserving spectral divide-and-conquer algorithm (Algorithm 3) shows a special behaviour that can be exploited algorithmically. Generally, after one step of spectral division, we get two smaller matrices that are pseudosymmetric with respect to two submatrices of the original signature matrix Σ, denoted Σ_+ and Σ_- in Algorithm 3. The p positive and the $n-p$ negative values on the diagonal of Σ split up in an unpredictable way. For definite matrices they split up neatly: The positive values gather in $\Sigma_+ = I_p$ and the negative values gather in $\Sigma_- = -I_{n-p}$. After spectral division, the upper left block A_{11} is definite pseudosymmetric with respect to I_p, i.e. symmetric positive definite. The lower right block A_{22} is definite pseudosymmetric with respect to $-I_{n-p}$, i.e. symmetric negative definite. This behavior is explained in the following theorem.

Theorem 9. *Let $A \in \mathbb{K}^{n\times n}$ be a definite pseudosymmetric matrix with respect to Σ with p positive and $n-p$ negative diagonal values. Let H be a basis of the invariant subspace of A associated with the p positive (respectively $n-p$ negative) eigenvalues, such that $H^*\Sigma H = \hat{\Sigma}$, where $\hat{\Sigma}$ is another signature matrix. Then $H^\dagger AH$ is Hermitian positive (respectively negative) definite and $\hat{\Sigma} = I_p$ (respectively $\hat{\Sigma} = -I_{n-p}$).*

Proof. Let $AV = V\Lambda$ be the eigenvalue decomposition given in Theorem 8. Let $V_p = \begin{bmatrix} v_1 & \ldots v_p \end{bmatrix}$ denote the first p columns of V, associated with the positive eigenvalues $\Lambda_+ = \operatorname{diag}(\lambda_1, \ldots, \lambda_p)$. Then $AV_+ = V_+\Lambda_+$ and

$$V_+^*\Sigma V_+ = I_p. \tag{13}$$

As H spans the same subspace as V_+, there must be $X \in \mathbb{K}^{p\times p}$ such that $H = V_+X$. Then $H^*\Sigma H = X^*V_+^*\Sigma V_+X = X^*X$ is positive definite. The only signature matrix with this property is the identity, showing $\hat{\Sigma} = I_p$.

Then it holds $H^\dagger = H^*\Sigma$ and therefore

$$H^\dagger AH = H^*\Sigma AH$$

is Hermitian positive definite, as ΣA is Hermitian positive definite. Concerning the negative eigenvalues it can be shown that $\hat{\Sigma} = -I_{n-p}$ and therefore

$$H^\dagger AH = -H^*\Sigma AH$$

is Hermitian negative definite. □

Theorem 9 greatly simplifies the divide-and-conquer method for definite pseudosymmetric matrices (Algorithm 3). We only need one spectral division step and can then fall back on existing algorithms for symmetric positive definite matrices. They can be of the divide-and-conquer variety, e.g. developed in [39], but do not have to be. In a high-performance setting, parallelized algorithms implemented in libraries such as ELPA [35] can be used.

5.2 Computing $(\Sigma, \hat{\Sigma})$-orthogonal representations

The computation of pseudoorthogonal subspace representations described in Section 4 also simplifies. In Step 6 of Algorithm 5, the smaller signature matrix Σ_+ related to the subspace associated with positive eigenvalues is computed by taking the signs of the diagonal matrix D of the previously computed LDL^{T} decomposition. Because of Theorem 9 we know that $\Sigma_+ = I_p$. The

Algorithm 6 Compute $(\Sigma, \hat{\Sigma})$-orthogonal invariant subspace representations of a definite pseudosymmetric projection matrix via Cholesky.

Input: Signature matrix Σ with r_+ positive and r_- negative diagonal values, $A \in \mathbb{R}^n$, such that ΣA is symmetric positive definite.

Output: A$(\Sigma, \hat{\Sigma})$-orthogonal basis $Q = \begin{bmatrix} Q_+ & Q_- \end{bmatrix}$, where $\hat{\Sigma} = \operatorname{diag}(I_{r_+}, -I_{r_-})$, i.e. $Q^{\mathsf{T}} \Sigma Q = \hat{\Sigma}$, where Q_+ is a basis of the invariant subspace of A associated with positive eigenvalues, Q_- is a basis of the invariant subspace of A associated with negative eigenvalues.

1: $S \leftarrow \operatorname{sign}(A)$.
2: $P_+ \leftarrow \frac{1}{2}(I_n + S)$.
3: $Q_{1,+} \leftarrow \texttt{chol}(\Sigma(1:r_+, 1:r_+) P_+(1:r_+, 1:r_+))$.
4: $Q_+ \leftarrow \begin{bmatrix} \Sigma(1:r_+, 1:r_+) Q_{1,+}^{\mathsf{H}} \\ P_+(r_+ + 1:n, 1:r_+) Q_{1,+}^{-1} \end{bmatrix}$.
5: $P_- \leftarrow \frac{1}{2}(I_n - S)$.
6: $Q_{1,-} \leftarrow \texttt{chol}(-\Sigma(1:r_-, 1:r_-) P_-(1:r_-, 1:r_-))$.
7: $Q_- \leftarrow \begin{bmatrix} \Sigma(1:r_-, 1:r_-) Q_{1,-}^{\mathsf{H}} \\ P_-(r_- + 1:n, 1:r_-) Q_{1,-}^{-1} \end{bmatrix}$.

Algorithm 7 Robust computation of $(\Sigma, \hat{\Sigma})$-orthogonal invariant subspace representations of a definite pseudosymmetric projection matrix via LDL^{T}

Input: Signature matrix Σ with r_+ positive and r_- negative diagonal values, $A \in \mathbb{R}^n$, such that ΣA is symmetric positive definite.

Output: A$(\Sigma, \hat{\Sigma})$-orthogonal basis $Q = \begin{bmatrix} Q_+ & Q_- \end{bmatrix}$, where $\hat{\Sigma} = \operatorname{diag}(I_{r_+}, -I_{r_-})$, i.e. $Q^{\mathsf{T}} \Sigma Q = \hat{\Sigma}$, where Q_+ is a basis of the invariant subspace of A associated with positive eigenvalues, Q_- is a basis of the invariant subspace of A associated with negative eigenvalues.

1: $S \leftarrow \operatorname{sign}(A)$.
2: $P_+ \leftarrow \frac{1}{2}(I_n + S)$.
3: $[L_+, D_+] \leftarrow \texttt{ldl}(\Sigma P_+)$.
4: Diagonalize D_+ if it has blocks on the diagonal: $[V_+, D_+] \leftarrow \texttt{eig}(D_+)$, such that the diagonal entries of D_+ are given in descending order.
5: $Q_+ \leftarrow \Sigma L_+ V_+(:, 1:r_+) D_+^{\frac{1}{2}}(1:r_+, 1:r_+)$.
6: $P_- \leftarrow \frac{1}{2}(I_n - S)$.
7: $[L_-, D_-] \leftarrow \texttt{ldl}(-\Sigma P_-)$.
8: Diagonalize D_- if it has blocks on the diagonal: $[V_-, D_-] \leftarrow \texttt{eig}(D_-)$, such that the diagonal entries of D_- are given in descending order.
9: $Q_- \leftarrow \Sigma L_- V_-(:, 1:r_-) D_-^{\frac{1}{2}}(1:r_-, 1:r_-)$.

LDL^{T} decomposition was taken of ΣP_+, which hence must be positive semidefinite. Therefore, the LDL^{T} decomposition can be substituted by a low-rank Cholesky factorization, similar to the symmetric case described in Algorithm 4. The computation of the rank (Step 3 in Algorithm 5) is omitted because we know that A has as many positive eigenvalues as Σ has positive diagonal values according to Theorem 8.

Numerical experiments (in particular examples from electronic structure theory, presented in Section 6.2) show that Algorithm 6 can break down due to numerical errors in floating point arithmetic. This happens when numerical errors lead to ΣP_+ having negative eigenvalues or ΣP_- having positive eigenvalues, such that the Cholesky decomposition breaks down. In order to avoid this case, we implement a more robust variant based on a truncated LDL^{T} decompositions, which includes pivoting, given in Algorithm 7.

5.3 Using Zolotarev functions to accelerate the matrix sign iteration

It was observed in [14] that the matrix sign function of a self-adjoint matrix A is given as the first factor of its generalized polar decomposition, offering a new perspective for its computation. A matrix $A \in \mathbb{K}^{n \times n}$ (under certain assumptions, see [27]) admits a generalized polar decomposition

with respect to a scalar product induced by a nonsingular matrix M

$$A = WS,$$

where W is a partial M-isometry and S is M-self-adjoint with no eigenvalues on the negative real axis. Canonical generalized polar decompositions can be defined for rectangular matrices [28]. We only consider the square case relevant to the application discussed in this work.

Iterations of a certain form that compute the generalized polar decomposition $A = WS$ work as a scalar iteration on the eigenvalues of S, pushing them closer to 1 in the course of the iteration. The following lemma clarifies this idea and is a slightly altered variant of Corollary 5.2 in [14].

Lemma 10. *Let g be a scalar function of the form*

$$g(x) = xh(x^2), \tag{14}$$

where h is a scalar function that is analytic in a domain containing the spectrum of the input matrix. Let $A \in \mathbb{R}^{n\times n}$ be a matrix with a generalized polar decomposition $A = WS$ for a given scalar product induced by $M \in \mathbb{R}^{n\times n}$. Let

$$G(X) := Xh(X^{\star_M} X) \tag{15}$$

be a matrix function. Then it holds

$$G(A) = WG(S) = Wg(S).$$

Proof. Observe

$$G(A) = G(WS) = WSh(S^{\star_M} W^{\star_M} WS) = WSh(S^{\star_M} S) = WG(S) = Wg(S).$$

We used $W^{\star_M} WS = S$, which holds according to Lemma 3.7 in [28]. The last equality holds because S is self-adjoint and $S^{\star_M} S = S^2$. □

Given an iteration of the form

$$X_{k+1} = G(X_k), \tag{16}$$

with G from (15), Lemma 10 states that it acts as the function g from (14) on the eigenvalues of the self-adjoint factor S. With the Jordan decomposition $S = ZJZ^{-1}$, $J = \operatorname{diag}(J_k)$ we see

$$X_{k+1} = WZ \operatorname{diag}(g(J_k))\, Z^{-1}, \quad k \in \mathbb{N}. \tag{17}$$

We have already seen in Theorem 9 that definite pseudosymmetric matrices have a special spectral structure. The following lemma shows that as a consequence, the eigenvalues of the self-adjoint factor S, on which iterations of the form (16) act, are real.

Lemma 11. *Let $A \in \mathbb{K}^{n\times n}$ be a definite pseudosymmetric matrix with respect to Σ. Then the generalized polar decomposition of A with respect to Σ,*

$$A = WS,$$

exists. The eigenvalues of S are positive real and constitute the absolute values of the eigenvalues of A.

Proof. For pseudosymmetric matrices it holds $A^{\star_\Sigma} A = \Sigma A^* \Sigma A = A^2$. As A has only real nonzero eigenvalues (following from Theorem 8), A^2 has only real positive eigenvalues. Hence the generalized polar decomposition exists. The self-adjoint factor of the polar decomposition is defined as $S = (A^{\star_\Sigma} A)^{\frac{1}{2}}$ and has only real eigenvalues, as the square roots of real positive values are real. They are the absolute values of the eigenvalues of A. □

Let A be scaled, such that its eigenvalues lie in $[-1,1]$, and let $0 < \ell < |\lambda|$ for all $\lambda \in \Lambda(A)$. Then the eigenvalues of S lie in $(\ell,1]$. A rational function $g(x) = xh(x^2)$ which maps them close to 1, i.e. approximates the scalar sign function on the interval $(\ell,1]$, can be used in an iteration (16). We see from (17) that the result will be an approximation to the polar factor W, which in our setting coincides with the matrix sign function. Luckily, explicit formulas for rational best-approximations of the sign function with form (14) were found by Zolotarev in 1877 [54]. In [37], Zolotarev functions are used to devise an iteration which computes the polar decomposition in just two steps. The algorithmic cost of the steps is increased compared to other iterative techniques, but the additional computations can be performed completely in parallel. We extend this approach for computing the polar decomposition of definite pseudosymmetric matrices.

We call the unique rational function of degree $(2r+1, 2r)$ solving

$$\min_{R \in \mathcal{R}_{2r+1,2r}} \max_{x \in [-1,-\ell] \cup [\ell,1]} |\operatorname{sign}(x) - R(x)|$$

for a given $0 < \ell < 1$ and an integer r, the *type* $(2r+1, 2r)$ *Zolotarev function*. Here, $\mathcal{R}_{2r+1,2r}$ denotes the set of rational functions, where the numerator is a polynomial of degree $2r+1$ and the denominator is a polynomial of degree $2r$. The Zolotarev function is given explicitly in the form of

$$Z_{2r+1}(x;\ell) := Cx \prod_{j=1}^{r} \frac{x^2 + c_{2j}}{x^2 + c_{2j-1}}. \tag{18}$$

The coefficients $c_1, \dots, c_{2r}$ are determined via the Jacobi elliptic functions $\operatorname{sn}(u;\ell)$ and $\operatorname{cn}(u;\ell)$ (see e.g. [1, Chapter 17], [2, Chapter 5]) as

$$c_i = \ell^2 \frac{\operatorname{sn}^2(\frac{iK'}{2r+1};\ell')}{\operatorname{cn}^2(\frac{iK'}{2r+1};\ell')}, \qquad i = 1, \dots 2r, \tag{19}$$

where $\ell' = \sqrt{1-\ell^2}$ and $K' = \int_0^{\pi/2} (1 - (\ell')^2 \sin^2(\theta))^{-1/2} d\theta$ are familiar quantities in the context of Jacobi elliptic functions. Details on the stable computation of the coefficients can be found in [37]. The constant $C > 0$ is uniquely determined, which will later be substituted by a normalization constant $\hat{C}$. We use implementations provided as MATLAB functions in [37] for their computation.

Zolotarev also showed (see [3, Chapter 9], [41, Chapter 4]) that $Z_{2r+1}(x;\ell)$ solves

$$\max_{P,Q \in \mathcal{P}_r} \min_{\ell \le x \le 1} x \frac{P(x^2)}{Q(x^2)},$$

where $\mathcal{P}_r$ denotes the set of polynomials of rank r. For $r = 1$, this optimization problem was solved in [36], leading to the dynamically weighted Halley (DWH) iteration. This iteration was used in [14] to compute the generalized polar decomposition. An iteration based on Zolotarev functions therefore generalizes the DWH approach in terms of higher-degree rational functions.

A key oberservation in [37] is that the composition of Zolotarev functions is again a Zolotarev function. More precisely, it holds

$$\hat{Z}_{2r+1}(\hat{Z}_{2r+1}(x;\ell);\ell_1) = \hat{Z}_{(2r+1)^2}(x;\ell),$$

where

$$\hat{Z}_{2r+1}(x;\ell) = \frac{Z_{2r+1}(x;\ell)}{Z_{2r+1}(1;\ell)} = \hat{C}x \prod_{j=1}^{r} \frac{x^2 + c_{2j}}{x^2 + c_{2j-1}}, \text{ with } \hat{C} = \prod_{j=1}^{r} \frac{1 + c_{2j-1}}{1 + c_{2j}}, \tag{20}$$

is a scaled Zolotarev function and $\ell_1 = \hat{Z}_{2r+1}(\ell;\ell)$. It can be verified that with $r := 8$, $\ell \ge 10^{-16}$, we have $Z_{(2r+1)^2}([\ell,1],\ell]) \subseteq [1-10^{-15},1]$. Consequently, employing Lemma 11 twice on a matrix $A = WS$ with $g(x) = \hat{Z}_{2r+1}(x;\ell)$, we see that the eigenvalues of $g(g(S))$ will be in the interval $[1-10^{-15},1]$, under the condition that all eigenvalues of S are in $[\ell,1]$ with $\ell \ge 10^{-16}$. In this sense, $G(G(A)) \approx W$ has converged to the polar factor W, after two iterations of Iteration (16). Choosing a higher r, algorithms can be devised that converge in just one step. It was argued in

[37] that a 2-step approach is a sensible choice to acquire a robust algorithm. This way, potential instabilities, e.g. in the computation of the Zolotarev coefficients c_i, are suppressed.

The scaled Zolotarev function can be represented in a partial fraction decomposition

$$\hat{Z}_{2r+1}(x;\ell) = \hat{C}x\left(1+\sum_{j=1}^{r}\frac{a_j}{x^2+c_{2j-1}}\right), \tag{21}$$

$$a_j = -\left(\prod_{k=1}^{r}(c_{2j-1}-c_{2k})\right)\cdot\left(\prod_{k=1,k\neq j}^{r}(c_{2j-1}-c_{2k-1})\right). \tag{22}$$

An iteration (16) derived from (21) takes the form

$$X_{k+1} = \hat{C}(X_k + \sum_{j=1}^{r} a_{j,k}X_k(X_k^{\star_M}X_k + c_{2j-1,k}I)^{-1}). \tag{23}$$

With $M=\Sigma$ as a signature matrix, (23) becomes

$$X_{k+1} = \hat{C}(X_k + \sum_{j=1}^{r} a_{j,k}X_k(X_k^{*}\Sigma X + c_{2j-1,k}\Sigma)^{-1}\Sigma). \tag{24}$$

Computing the inverse via an LDL^{T} decomposition leads to a first practical iteration.

$$\begin{cases} Z_{2j-1,k} = (X_k^{*}\Sigma X_k + c_{2j-1,k}\Sigma), \quad [L_j, D_j, P_j] = \mathrm{ldl}(Z_{2j-1,k}), \\ X_{k+1} = \hat{C}(X_k + \sum_{j=1}^{r} a_j X_k P_j L_j^{-*} D_j^{-1} L_j^{-1} P_j^{\mathsf{T}}\Sigma) \end{cases} \tag{25}$$

The first line of (25) means that in iteration k, the LDL^{T} decomposition $Z_{2j-1,k} = P_jL_jD_jL_j^{*}P_j^{\mathsf{T}}$ is computed for each $Z_{2j-1,k}$, $j=1,\dots,r$. P_j is a permutation matrix, L_j is lower triangular, and D_j is block-diagonal with 1×1 or 2×2 blocks. In [14], the special case for $r=1$ is derived. There, the iteration is rewritten such that it becomes inverse-free and a hyperbolic QR decomposition is employed instead.

Lemma 12. *Let Σ be a signature matrix, $\eta\in\mathbb{R}$. For $X\in\mathbb{K}^{n\times n}$, let $\begin{bmatrix}\eta X\\ I\end{bmatrix} = HR$, $H=\begin{bmatrix}H_1\\H_2\end{bmatrix}\in\mathbb{K}^{2n\times n}$, $R\in\mathbb{K}^{n\times n}$ be a decomposition, such that $H^{*}\begin{bmatrix}\Sigma & \\ & \Sigma\end{bmatrix}H=\hat{\Sigma}$, where $\hat{\Sigma}\in\mathbb{R}^{n\times n}$ is another signature matrix. Then*

$$\eta X(I+\eta^2 X^{\star_\Sigma}X)^{-1} = H_1\hat{\Sigma}H_2^{*}\Sigma.$$

Proof. Applying Lemma 5.4 in [14] with $V:=H$ and $M:=N:=\Sigma$ yields

$$\eta X(I+\eta^2X^{\star_\Sigma}X)^{-1} = H_1\left(\Sigma H^{*}\begin{bmatrix}\Sigma & \\ & \Sigma\end{bmatrix}H\right)^{-1}\Sigma H_2^{*}\Sigma = H_1\hat{\Sigma}H_2^{*}\Sigma,$$

where in the last equality we have used $H^{*}\begin{bmatrix}\Sigma & \\ & \Sigma\end{bmatrix}H=\hat{\Sigma}$.

□

Using Lemma 12 with $\eta=\frac{1}{\sqrt{c_{2j-1,k}}}$, (24) can be rewritten as

$$\begin{cases} \begin{bmatrix}X_k\\ \sqrt{c_{2j-1,k}}I\end{bmatrix} = \begin{bmatrix}H_{1,j}\\H_{2,j}\end{bmatrix}R_j, \text{ where } \begin{bmatrix}H_{1,j}\\H_{2,j}\end{bmatrix}^{*}\begin{bmatrix}\Sigma & \\ & \Sigma\end{bmatrix}\begin{bmatrix}H_{1,j}\\H_{2,j}\end{bmatrix} = \hat{\Sigma} \\ X_{k+1} = \hat{C}(X_k + \sum_{j=1}^{r}\frac{a_j}{\sqrt{c_{2j-1}}}H_{1,j}\hat{\Sigma}H_{2,j}^{*}\Sigma). \end{cases} \tag{26}$$

Algorithm 8 Hyperbolic Zolo-PD for definite pseudosymmetric matrices

Input: Signature matrix Σ with p positive and $n-p$ negative values on the diagonal, $A \in \mathbb{C}^{n\times n}$ such that ΣA is Hermitian positive definite, Zolotarev rank r.

Output: $S = \operatorname{sign}(A)$.

1: Estimate $\alpha \gtrsim \max\{|\lambda| : \lambda \in \Lambda(A)\}$, $\beta \lesssim \min\{|\lambda| : \lambda \in \Lambda(A)\}$.

2: $X_0 \leftarrow \frac{1}{\alpha}A$, $\ell \leftarrow \frac{\beta}{\alpha}$.

First iteration:

3: **for** $j = 1, \dots, 2r$ **do**

4: $\quad c_j \leftarrow \ell^2 \mathrm{sn}^2(\frac{\mathrm{i}K'}{2r+1}; \ell')/\mathrm{cn}^2(\frac{\mathrm{i}K'}{2r+1}; \ell')$. ▷ See (19)

5: **end for**

6: **for** $j = 1, \dots, r$ **do**

7: $\quad a_j \leftarrow -\left(\prod_{k=1}^{r}(c_{2j-1} - c_{2k})\right) \cdot \left(\prod_{k=1,k\neq j}^{r}(c_{2j-1} - c_{2k-1})\right)$. ▷ See (22)

8: **end for**

9: $\hat{C} \leftarrow \prod_j^r \frac{1+c_{2j-1}}{1+c_{2j}}$ ▷ See (20)

10: Compute X_1 according to (26), using `LDLIQR2` algorithm in [14]:

$$\begin{cases} \begin{bmatrix} X_0 \\ \sqrt{c_{2j-1}}I \end{bmatrix} = \begin{bmatrix} H_{1,j} \\ H_{2,j} \end{bmatrix} R_j, \text{ where } \begin{bmatrix} H_{1,j} \\ H_{2,j} \end{bmatrix}^* \begin{bmatrix} \Sigma & \\ & \Sigma \end{bmatrix} \begin{bmatrix} H_{1,j} \\ H_{2,j} \end{bmatrix} = \hat{\Sigma} \\ \\ X_1 \leftarrow \hat{C}(X_k + \sum_{j=1}^{r} \frac{a_j}{\sqrt{c_{2j-1}}} H_{1,j}\hat{\Sigma}H_{2,j}^*\Sigma). \end{cases}$$

11: $\ell \leftarrow \hat{C}\ell \prod_{j=1}^{r}(\ell^2 + c_{2j})/(\ell^2 + c_{2j-1})$.

12: Repeat Step 3 to Step 9 to update c_j for $j = 1, \dots, 2r$ and a_j for $j = 1, \dots, r$ and $\hat{C}$.

Second iteration:

13: Compute X_2 according to (25):

$$\begin{cases} Z_{2j-1,k} = (X_k^*\Sigma X_k + c_{2j-1,k}\Sigma), \quad [L_j, D_j, P_j] = \mathrm{ldl}(Z_{2j-1,k}), \\ \\ X_{k+1} \leftarrow \hat{C}(X_k + \sum_{j=1}^{r} a_j X_k P_j L_j^{-*} D_j^{-1} L_j^{-1} P_j^{\mathsf{T}}\Sigma). \end{cases}$$

14: **if** $\|X_2 - X_1\|_F / \|X_2\|_F \leq u^{1/(2r+1)}$ **then**

15: $\quad S \leftarrow X_2$.

16: **else**

17: $\quad A \leftarrow X_2$, return to Step 1.

18: **end if**

As in iteration (25), the first line refers to the computation of a total of r independent decompositions $\begin{bmatrix} X_k \\ \sqrt{c_{2j-1,k}}I \end{bmatrix} = H_j R_j$ for $j = 1, \dots, r$, per iteration step. One way of computing the needed matrix H is the hyperbolic QR decomposition, which we introduced in Theorem 5. Computing it via a column-elimination approach is notoriously unstable. This is why [14, 15] exploit a link to the LDL^{T} factorization and introduce the `LDLIQR2` algorithm.

Algorithm 8 is the pseudocode of a Zolotarev-based computation of the generalized polar factor. We assume that convergence is reached after just two steps, which are explicitly written in the Algorithm. For the computation of iterate X_1, iteration (26) is employed. For the computation of the H matrices, we use the `LDLIQR2` algorithm from [14], which showed a better numerical stability than column-elimination based approaches. The second iterate X_2 can safely be computed using the LDL^{T}-based iteration (25) for the same reasoning given in [37]. The parameter estimation and the scaling of A (Steps 1 and 2) is needed to make sure that the eigenvalues of the self-adjoint factor lie in the interval $[\ell, 1]$ (see Lemma 11 and the discussion following). In our implementation, these are bounded using the MATLAB functions `normest` and `condest`.

Algorithm 8 converges even for badly conditioned matrices. As explained in [37], for well-conditioned A, it is possible to skip the first iteration or choose a lower Zolotarev rank $r < 8$.

In exact arithmetic, the algorithm converges in 2 steps, as argued above. As a safeguard for

numerical errors we adopt the stopping criterion from [37], $\|X_2 - X_1\|_F / \|X_2\|_F \leq u^{1/(2r+1)}$, to guarantee convergence, using the known convergence rate of $2r+1$. We assume calculations are carried out in IEEE double precision with unit roundoff $u = 2^{-53} \approx 1.1 \times 10^{-16}$.

6 Numerical experiments

In this section we apply one step of spectral divide-and-conquer (Algorithm 3) on definite pseudosymmetric matrices. The matrix sign function is computed by the hyperbolic Zolo-PD algorithm (Algorithm 8), algorithms based on the ΣDWH iteration presented in [14], or a scaled Newton iteration with suboptimal scaling presented in [22]. We expect these algorithms to show the same convergence properties as in the symmetric case, due to Lemma 10. Zolo-PD should converge in 2 steps, ΣDWH in 6 steps and Newton in 9 steps. We use Algorithm 6 or 7 to compute $(\Sigma, \hat{\Sigma})$-orthogonal subspace representations used in the spectral division. All experiments are performed in MATLAB R2017a using double-precision arithmetic running on Ubuntu 18.04.5, using an Intel(R) Core™ i7-8550U CPU with 4 cores, 8 threads, and a clock rate of 1.80 GHz. Random matrices are generated with a seed defined by `rng(0)`. We choose r according to Table 3.1 in [37], i.e. the condition number is estimated and r is chosen such that a convergence within two steps is guaranteed as outlined in Section 5.3.

6.1 Random pseudosymmetric matrices

The goal of our first numerical experiment is to determine the achieved accuracy with different methods for computing the matrix sign function.

Example 1 Σ is a signature matrix, where the diagonal values are chosen to be 1 or -1 with equal probability. Given a number $\kappa = \mathrm{cond}(A)$, we generate real 250×250 matrices as $A = \Sigma Q D Q^{\mathsf{T}}$. D is a diagonal matrix containing equally spaced values between 1 and κ. Q is a random orthogonal matrix (`Q=orth(rand(n,n))` in MATLAB). We perform 10 runs for different randomly generated matrices and compare the backward error represented by $\|Q_+^{\mathsf{T}} \Sigma A Q_-\|_{\mathrm{F}} / \|A\|_{\mathrm{F}}$ that is achieved by the different methods described in this work, [14] and [22].

The averaged results are given in Figure 1. All methods yield backward errors smaller than 10^{-9}, even for badly conditioned matrices. All show a similar behavior. Hyperbolic Zolo-PD exhibits the highest backward error. Compared to the DWH-based iteration this is expected, because Zolotarev functions of higher order are used. The direct application of Zolotarev functions of high degree is known to be unstable [37]. In the indefinite setting, this phenomenon seems to appear sooner than in the setting described in [37]. The accuracy of DWH can be improved by employing permuted Lagrangian graph (PLG) bases. This way, the accuracy is comparable to a Newton approach [22]. Permuted Lagrangian graph bases can also be employed for Zolotarev iterations of higher order but go beyond the scope of this work.

Figure 2 displays the data of the individual runs of the same experiment. Here we see that even badly conditioned matrices often achieve a backward error of 10^{-14}, but some outliers increase the average. Further investigations are required in order to answer the question of what backward error can be achieved for a given matrix. The red crosses denote the matrices of a given κ for which hyperbolic Zolo-PD performed worst. We see that for the same matrices ΣDWH with `LDLIQR2` and the Newton iteration also perform worse than on other matrices generated in the same way. The quality therefore seems innate to the considered matrix. When PLG bases are employed, this relation can not be observed as clearly but is still noticeable.

The second example provides first insights on the performance which can be expected by using different methods.

Example 2 A random matrix of size $5\,000 \times 5\,000$ is generated as in Example 1. We measure the number of iterations and the runtime using different methods to compute the matrix sign function. We measure the runtime of the sequential implementation of Zolo-PD, as well as the runtime resulting from its critical path. This means that we only take the runtime of one of the r independent steps in each iteration, i.e. the first lines in iterations (25) and (26), into account.

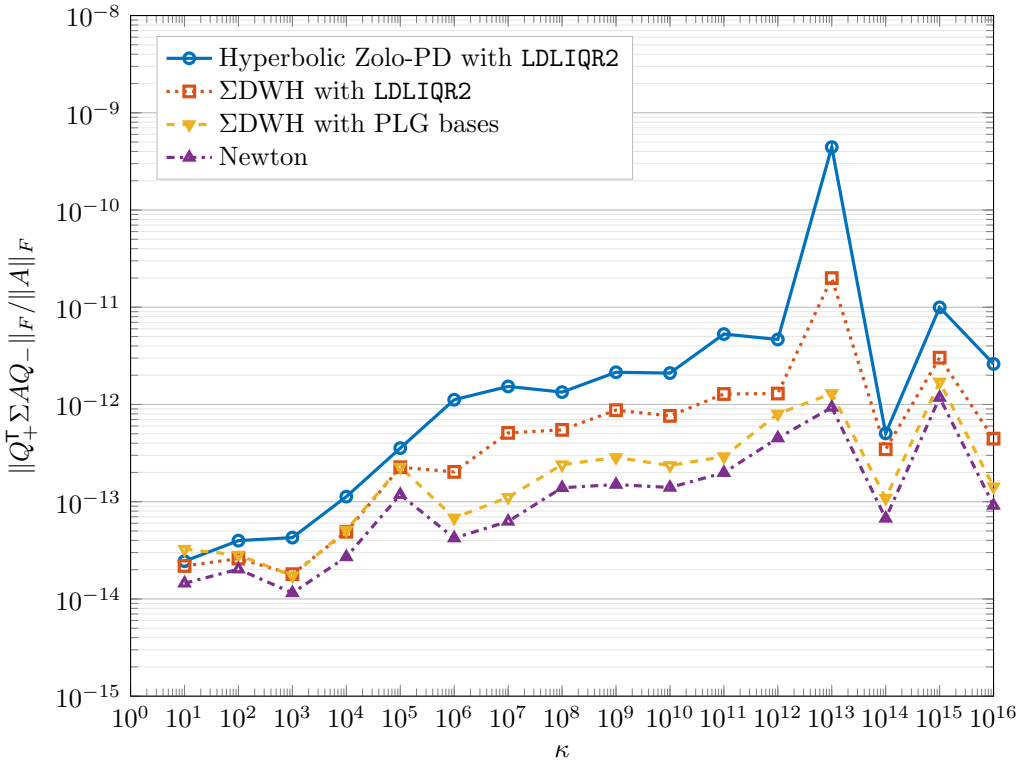
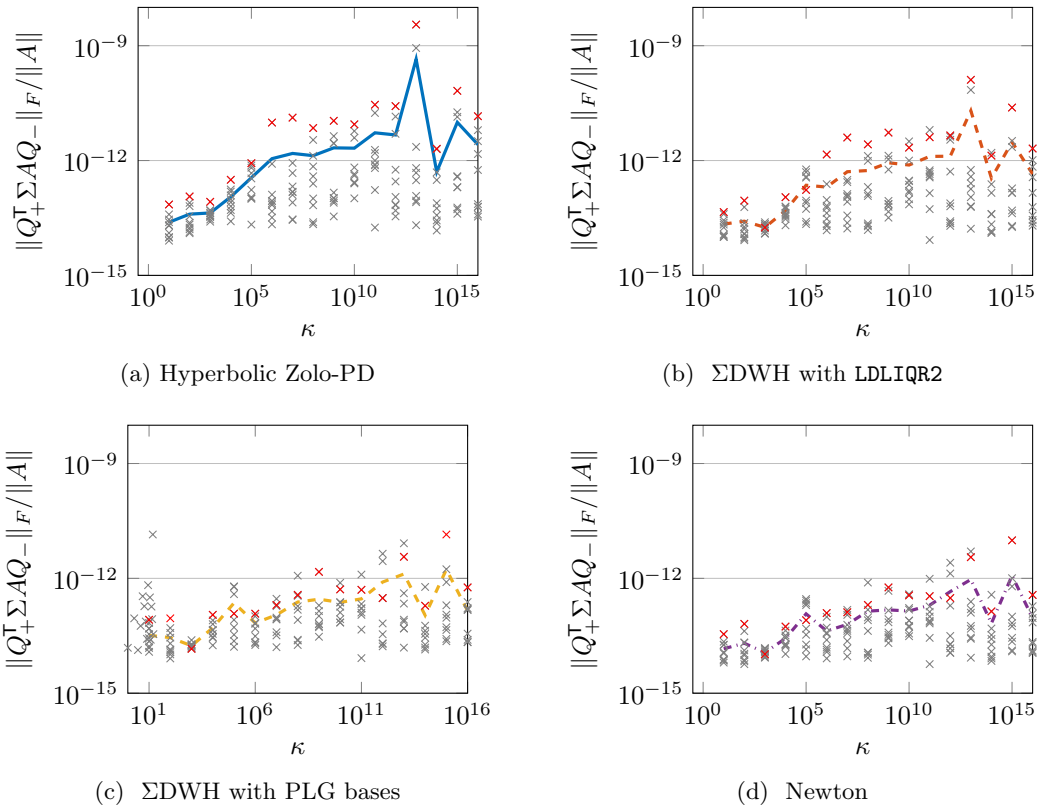


Figure 1: *Example 1*: Average residual after one spectral divide-and-conquer step, for 10 random matrices of size 250×250 with certain condition numbers. Different methods are used for computing the matrix sign function.

This runtime reflects a performance which can be achieved when these independent computations are implemented in parallel. We compare it to runtimes achieved by ΣDWH based on $LDLIQR2$ and ΣDWH based on LDL^T factorizations [16], and the Newton iteration [22]. The computation of PLG bases is not yet suited for large-scale performance-critical algorithms, which is why it is not included in the comparison. The results are found in Table 1.

	κ	10^2	10^8	10^{12}
# iterations	Hyperbolic Zolo-PD	2	2	2
	ΣDWH with LDLIQR2	5	6	6
	ΣDWH with LDL^T	5	6	x
	Newton	7	9	9
runtime	Hyperbolic Zolo-PD	941.70	1136.91	1240.86
	(critical path)	(298.66)	(255.86)	(257.70)
	ΣDWH with LDLIQR2	883.79	988.39	1067.43
	ΣDWH with LDL^T	281.95	304.27	x
	Newton	355.05	379.38	416.19
backward error $\frac{\|Q_+^T \Sigma A Q_-\|_F}{\|A\|_F}$	Hyperbolic Zolo-PD	7.42e-14	1.05e-11	1.78e-13
	ΣDWH with LDLIQR2	7.88e-14	2.81e-12	3.29e-13
	ΣDWH with LDL^T	8.50e-14	1.38e-11	x
	Newton	1.70e-13	6.66e-13	1.01e-13

Table 1: *Example 2:* Number of iterations, runtimes and error for different methods of spectral division for a matrix of size $5\,000 \times 5\,000$. ΣDWH with LDL^T did not converge for matrices with $\kappa = 10^{12}$.



(a) Hyperbolic Zolo-PD

(b) ΣDWH with `LDLIQR2`

(c) ΣDWH with PLG bases

(d) Newton

Figure 2: *Example 1*: Residuals after one step of spectral divide-and-conquer for 10 runs with randomly generated matrices of certain condition numbers.

The methods converge as expected and all except ΣDWH with LDL^T show good accuracy. ΣDWH with LDL^T is known to be unstable for badly conditioned matrices [14]. However, if it converges, it is the fastest among the measured methods. The computational effort of one ΣDWH iteration based on LDL^T is comparable to the effort of one Newton iteration that is also based on an LDL^T factorization. ΣDWH converges in up to 6 steps, and Newton uses up to 9 steps. If `LDLIQR2` is employed instead of LDL^T in ΣDWH, the computational effort doubles, as a second LDL^T decomposition is used for "reorthogonalization". This makes it slower than the Newton iteration.

Measuring the runtime of the critical path of Hyperbolic Zolo-PD, i.e. including only the computation of one addend in Equation (25), gives an idea of the wall time that could be achieved in a parallel setting. In this sense we achieve a runtime reduction of 38% for badly conditioned matrices compared to Newton when using Hyperbolic Zolo-PD. The main building blocks of all iterations are LDL^T decompositions and matrix multiplications, so they would benefit equally by further optimizations of these routines.

Hyperbolic Zolo-PD could be accelerated at the cost of stability, when one LDL^T decomposition is used instead of `LDLIQR2` in each iteration step.

6.2 Applications in electronic structure computations

We now apply the developed method to two motivating examples concerning electronic excitations in solids and molecules.

Example 3 The `exciting` package [25, 50] implements various ab initio methods for computing excited states of solids or molecules, based on (linearized) augmented planewave + local orbital ((L)APW+lo) methods. It can be used to compute the optical scattering spectrum of Lithium Fluoride based on the Bethe-Salpeter equation. The main computational effort in this example is

to compute eigenvalues and eigenvectors of a matrix of the form

$$H_{LF} = \begin{bmatrix} A_{LF} & B_{LF} \\ -B_{LF} & -A_{LF} \end{bmatrix} \in \mathbb{C}^{2560\times 2560},\ A_{LF} = A_{LF}^{\mathsf{H}},\ B_{LF} = B_{LF}^{\mathsf{H}}. \tag{27}$$

H_{LF} is obviously pseudo-Hermitian with respect to $\Sigma = \operatorname{diag}(I_n, -I_n)$. Due to the additional structure, the eigenvalues are known to come in pairs of $\pm\lambda$ [16]. One step of spectral division results in a positive definite matrix, from which all eigenvalues and eigenvectors can be reconstructed. We extracted the matrix from the FORTRAN-based `exciting` code as a test example for our MATLAB-based prototype.

	Hyperbolic Zolo-PD	ΣDWH with `LDLIQR2`	ΣDWH with LDL^{T}	Newton
# iterations	2	5	5	7
Zolotarev rank	4	1	1	not applicable
backward error (Chol, Alg. 6)	1.02e-10	7.42e-11	9.62e-11	2.48e-10
backward error (LDL, Alg. 7)	6.93e-18	7.26e-18	6.99e-18	1.48e-17

Table 2: *Example 3:* Results for Bethe-Salpeter matrix computed for Lithium Fluoride.

The results in Table 2 show that convergence is achieved in a limited number of iterations for all methods, as expected. The reported backward error $\frac{\|Q_+^{\mathsf{T}}\Sigma A Q_-\|_F}{\|A\|_F}$ depends largely on the chosen method for computing a hyperbolic subspace representation. The Cholesky-based method does not work well. The eigenvalues smallest in modulus easily "pass over", such that the computed quantities ΣP_+ or $-\Sigma P_-$ have negative eigenvalues. The Cholesky-based method in Algorithm 6 does not accurately capture this behavior, while the LDL^{T}-based method alleviates the effect through pivoting.

All methods for computing the matrix sign function work equally well concerning accuracy because H_{LF} is well conditioned ($\operatorname{cond}(H_{LF}) \approx 10$).

Example 4 In [10, 11] a Bethe-Salpeter approach is explored in the context of tensor-structured Hartree-Fock theory for molecules [42]. We consider the N_2H_4 example in [11]. With real-valued orbitals the derivation arrives at a structured eigenvalue problem similar to Example 3, but with real values.

$$H_{N_2H_4} = \begin{bmatrix} A_{N_2H_4} & B_{N_2H_4} \\ -B_{N_2H_4}^{\mathsf{T}} & -A_{N_2H_4}^{\mathsf{T}} \end{bmatrix} \in \mathbb{R}^{1314\times 1314},\ A_{N_2H_4} = A_{N_2H_4}^{\mathsf{T}},\ B_{N_2H_4} \approx B_{N_2H_4}^{\mathsf{T}}. \tag{28}$$

While the original derivation in [42] yields a symmetric off-diagonal block B, in the construction in [11], this property is lost. The property of pseudosymmetry, however, is not affected, making our developed method applicable.

	Hyperbolic Zolo-PD	ΣDWH with `LDLIQR2`	ΣDWH with LDL^{T}	Newton
# iterations	2	5	5	7
Zolotarev rank	5	1	1	not applicable
backward error (Chol, Alg. 6)	1.19e-18	9.23e-19	1.46e-17	2.04e-18
backward error (LDL, Alg. 7)	1.22e-18	9.62e-19	1.46e-17	2.13e-18

Table 3: *Example 4:* Results for Bethe-Salpeter matrix computed for N_2H_4.

Numerical results of the spectral division are found in Table 3. All methods yield good results ($\mathrm{cond}(H_{N_2H_4}) \approx 5$). In contrast to Example 3, no problem occurs when the Cholesky decomposition is used for computing hyperbolic subspace representations. An explanation is probably linked to the fact that real matrices instead of complex ones are considered but requires further investigation.

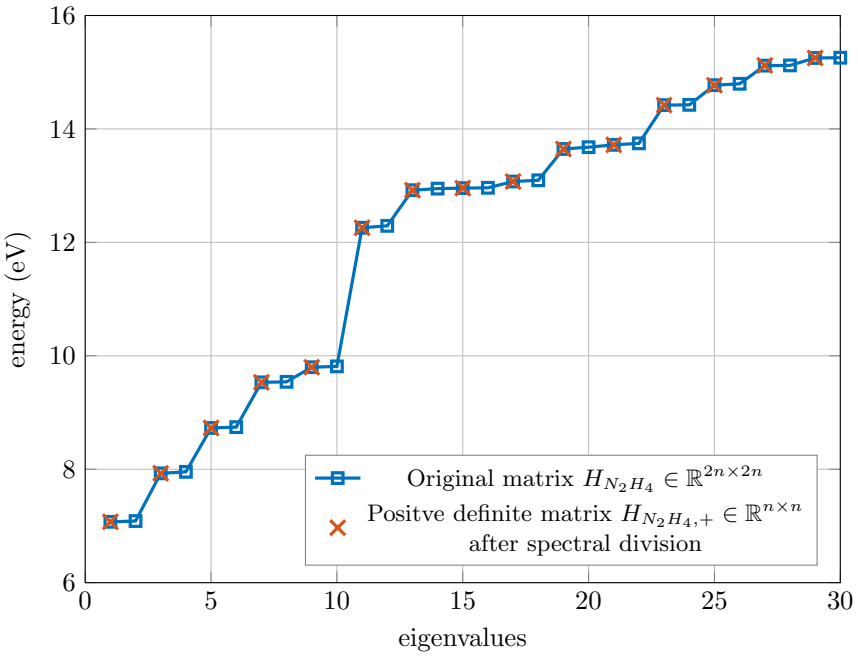


Figure 3: *Example 4*: Absolute values of eigenvalues corresponding to N_2H_4.

Figure 3 corresponds to Figure 2 in [11] and displays absolute values of the eigenvalues of $H_{N_2H_4}$. The red crosses denote the eigenvalues of the positive definite matrix resulting after one step of spectral division (A_{11} in Algorithm 3). The remaining eigenvalues have (approximately) equal modulus, but opposite sign and are found as the eigenvalues of the negative definite matrix (A_{22} in Algorithm 3).

7 Conclusions

We presented a generalization of the well-known spectral divide-and-conquer approach for the computation of eigenvalues and eigenvectors of pseudosymmetric matrices. In particular, when matrices with additional definiteness properties are considered, many parallels to the symmetric divide-and-conquer method become apparent. These parallels allow a computation of the matrix sign function, the key element for spectral division approaches, in just two iterations, using Zolotarev functions. Furthermore, the eigenvalue problem is decoupled into two smaller symmetric eigenvalue problems that can be solved with existing techniques. The presented algorithm is a promising new approach in the field of computing electronic excitations.

As we presented a completely new approach for structured eigenvalue computations, naturally, many possible future research directions open up as a consequence of this work. It is possible to use permuted Lagrangian graph bases, as presented in [16], to further improve the accuracy of the Zolotarev iteration for computing the matrix sign function. This should go hand in hand with a well-founded analysis of the stability of the proposed methods. In the same vein, the numerical behavior of the subspace computations (Algorithms 6 and 7) is not yet fully understood, as the examples presented in Section 6.2 show. Regarding the applications concerning electron excitation, the matrices ((9) to (11)) show even more structure than has been exploited in the presented methods. Making the proposed iterations aware of these structures, such that they operate directly on the matrix blocks A and B, is a promising direction towards even more efficient

methods.

References

[1] M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions: With Formulas, Graphs, and Mathematical Tables*, vol. 55 of National Bureau of Standards Applied Mathematics Series, U.S. Government Printing Office, Washington, D.C., 1972.

[2] N. I. Akhiezer, *Elements of the Theory of Elliptic Functions*, vol. 79 of Translations of Mathematical Monographs, American Mathematical Society, Providence, RI, 1990. Translated from the second Russian edition by H. H. McFaden.

[3] N. I. Akhiezer, *Theory of Approximation*, Dover, New York, 1992.

[4] E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen, *LAPACK Users' Guide*, SIAM, Philadelphia, PA, third ed., 1999.

[5] Z. Bai and J. Demmel, *Design of a parallel nonsymmetric eigenroutine toolbox, part i*, Tech. Rep. UCB/CSD-92-718, EECS Department, University of California, Berkeley, Feb 1993.

[6] Z. Bai and J. Demmel, *Design of a parallel nonsymmetric eigenroutine toolbox, Part II*, tech. rep., Computer Science Division, University of California, Berkeley, CA 94720, 1994.

[7] Z. Bai, J. Demmel, and M. Gu, *An inverse free parallel spectral divide and conquer algorithm for nonsymmetric eigenproblems*, Numer. Math., 76 (1997), pp. 279–308.

[8] G. Ballard, J. Demmel, and I. Dumitriu, *Minimizing communication for eigenproblems and the singular value decomposition*, 2010.

[9] G. Ballard, J. Demmel, O. Holtz, and O. Schwartz, *Minimizing communication in numerical linear algebra*, SIAM J. Matrix Anal. Appl., 32 (2011), pp. 866–901.

[10] P. Benner, S. Dolgov, V. Khoromskaia, and B. N. Khoromskij, *Fast iterative solution of the Bethe-Salpeter eigenvalue problem using low-rank and QTT tensor approximation*, J. Comput. Phys., 334 (2017), pp. 221–239.

[11] P. Benner, V. Khoromskaia, and B. N. Khoromskij, *A reduced basis approach for calculation of the Bethe-Salpeter excitation energies using low-rank tensor factorizations*, Mol. Phys., 114 (2016), pp. 1148–1161.

[12] P. Benner, M. Köhler, and J. Saak, *A cache-aware implementation of the spectral divide-and-conquer approach for the non-symmetric generalized eigenvalue problem*, Proc. Appl. Math. Mech., 14 (2014), pp. 819–820.

[13] ——, *Fast approximate solution of the non-symmetric generalized eigenvalue problem on multicore architectures*, in Parallel Computing: Accelerating Computational Science and Engineering (CSE), M. Bader, A. Bodeand, H.-J. Bungartz, M. Gerndt, G. R. Joubert, and F. Peters, eds., vol. 25 of Advances in Parallel Computing, IOS Press, 2014, pp. 143–152.

[14] P. Benner, Y. Nakatsukasa, and C. Penke, *Stable and efficient computation of generalized polar decompositions*, SIAM J. Matrix Anal. Appl., 43 (2022), pp. 1058–1083.

[15] P. Benner and C. Penke, *GR decompositions and their relations to Cholesky-like factorizations*, Proc. Appl. Math. Mech., 20 (2021), p. e202000065.

[16] ——, *Efficient and accurate algorithms for solving the Bethe-Salpeter eigenvalue problem for crystalline systems*, J. Comput. Appl. Math., 400 (2022), p. 113650.

[17] M. Brebner and J. Grad, *Eigenvalues of $Ax = \lambda Bx$ for real symmetric matrices A and B computed by reduction to a pseudosymmetric form and the HR process*, Linear Algebra Appl., 43 (1982), pp. 99–118.

[18] J. R. Bunch and L. Kaufman, *Some stable methods for calculating inertia and solving symmetric linear systems*, Math. Comp., 31 (1977), pp. 163–179.

[19] W. Bunse and A. Bunse-Gerstner, *Numerische Lineare Algebra*, Teubner, Stuttgart, 1985.

[20] A. Bunse-Gerstner, *Berechnung der Eigenwerte einer Matrix mit dem HR-Verfahren*, in Numerische Behandlung von Eigenwertaufgaben, Band 2 (Tagung, Tech. Univ. Clausthal, Clausthal, 1978), vol. 43 of Internat. Schriftenreihe Numer. Math., Birkhäuser, Basel-Boston, Mass., 1979, pp. 26–39.

[21] A. Bunse-Gerstner, *An analysis of the HR algorithm for computing the eigenvalues of a matrix*, Linear Algebra Appl., 35 (1981), pp. 155–173.

[22] R. Byers and H. Xu, *A new scaling for Newton's iteration for the polar decomposition and its backward stability*, SIAM J. Matrix Anal. Appl., 30 (2008), pp. 822–843.

[23] M. Casida, *Time-dependent density functional response theory for molecules*, in Recent Advances in Density Functional Methods, World Scientific, 1995, pp. 155–192.

[24] G. H. Golub and C. F. Van Loan, *Matrix Computations*, Johns Hopkins Studies in the Mathematical Sciences, Johns Hopkins University Press, Baltimore, fourth ed., 2013.

[25] A. Gulans, S. Kontur, C. Meisenbichler, D. Nabok, P. Pavone, S. Rigamonti, S. Sagmeister, U. Werner, and C. Draxl, *exciting: a full-potential all-electron package implementing density-functional theory and many-body perturbation theory*, Journal of Physics: Condensed Matter, 26 (2014), p. 363202.

[26] N. J. Higham, *Functions of Matrices: Theory and Computation*, Applied Mathematics, SIAM, Philadelphia, PA, 2008.

[27] N. J. Higham, D. Mackey, N. Mackey, and F. Tisseur, *Functions preserving matrix groups and iterations for the matrix square root*, SIAM J. Matrix Anal. Appl., 26 (2005), pp. 849–877.

[28] N. J. Higham, C. Mehl, and F. Tisseur, *The canonical generalized polar decomposition*, SIAM J. Matrix Anal. Appl., 31 (2010), pp. 2163–2180.

[29] C. Kenney and A. J. Laub, *Rational iterative methods for the matrix sign function*, SIAM J. Matrix Anal. Appl., 12 (1991), pp. 273–291.

[30] ——, *The matrix sign function*, IEEE Trans. Autom. Control, 40 (1995), pp. 1330–1348.

[31] D. Keyes, H. Ltaief, Y. Nakatsukasa, and D. Sukkari, *High-performance partial spectrum computation for symmetric eigenvalue problems and the SVD*, 2021.

[32] H. Ltaief, D. Sukkari, A. Esposito, Y. Nakatsukasa, and D. Keyes, *Massively parallel polar decomposition on distributed-memory systems*, ACM Trans. Parallel Comput., 6 (2019).

[33] D. S. Mackey, N. Mackey, and F. Tisseur, *Structured factorizations in scalar product spaces*, SIAM J. Matrix Anal. Appl., 27 (2005), pp. 821–850.

[34] A. N. Malyshev, *Parallel algorithm for solving some spectral problems of linear algebra*, Linear Algebra Appl., 188/189 (1993), pp. 489–520.

[35] A. Marek, V. Blum, R. Johanni, V. Havu, B. Lang, T. Auckenthaler, A. Heinecke, H.-J. Bungartz, and H. Lederer, *The ELPA library: scalable parallel eigenvalue solutions for electronic structure theory and computational science*, J. Phys. Condens. Matter, 26 (2014), p. 213201.

[36] Y. Nakatsukasa, Z. Bai, and F. Gygi, *Optimizing Halley's iteration for computing the matrix polar decomposition*, SIAM J. Matrix Anal. Appl., 31 (2010), pp. 2700–2720.

[37] Y. Nakatsukasa and R. W. Freund, *Computing fundamental matrix decompositions accurately via the matrix sign function in two iterations: the power of Zolotarev's functions*, SIAM Rev., 58 (2016), pp. 461–493.

[38] Y. Nakatsukasa and N. J. Higham, *Backward stability of iterations for computing the polar decomposition*, SIAM J. Matrix Anal. Appl., 33 (2012), pp. 460–479.

[39] ——, *Stable and efficient spectral divide and conquer algorithms for the symmetric eigenvalue decomposition and the SVD*, SIAM J. Sci. Comp, 35 (2013), pp. A1325–A1349.

[40] G. Onida, L. Reining, and A. Rubio, *Electronic excitations: density-functional versus many-body Green's-function approaches*, Rev. Mod. Phys., 74 (2002), pp. 601–659.

[41] P. P. Petrushev and V. A. Popov, *Rational Approximation of Real Functions*, vol. 28 of Encyclopedia of Mathematics and its Applications, Cambridge University Press, Cambridge, 1987.

[42] E. Rebolini, J. Toulouse, and A. Savin, *Electronic excitation energies of molecular systems from the Bethe-Salpeter equation: Example of the H2 molecule*, in Concepts and Methods in Modern Theoretical Chemistry, S. Ghosh and P. K. Chattaraj, eds., CRC Press, Boca Raton, 2013, ch. 18, pp. 367–389.

[43] J. D. Roberts, *Linear model reduction and solution of the algebraic Riccati equation by use of the sign function*, Internat. J. Control, 32 (1980), pp. 677–687.

[44] S. Sagmeister and C. Ambrosch-Draxl, *Time-dependent density functional theory versus Bethe-Salpeter equation: an all-electron study*, Phys. Chem. Chem. Phys., 11 (2009), pp. 4451–4457.

[45] T. Sander, E. Maggio, and G. Kresse, *Beyond the Tamm-Dancoff approximation for extended systems using exact diagonalization*, Phys. Rev. B, 92 (2015), p. 045209.

[46] M. Shao, F. H. da Jornada, C. Yang, J. Deslippe, and S. G. Louie, *Structure preserving parallel algorithms for solving the Bethe-Salpeter eigenvalue problem*, Linear Algebra Appl., 488 (2016), pp. 148–167.

[47] S. Singer and S. Singer, *Rounding-error and perturbation bounds for the indefinite QR factorization*, Linear Algebra Appl., 309 (2000), pp. 103–119.

[48] X. Sun and E. S. Quintana-Ortí, *Spectral division methods for block generalized Schur decompositions*, Math. Comp., 73 (2004), pp. 1827–1847.

[49] K. Veselić, *Damped Oscillations of Linear Systems*, vol. 2023 of Lecture Notes in Mathematics, Springer-Verlag, Berlin Heidelberg, 2011.

[50] C. Vorwerk, B. Aurich, C. Cocchi, and C. Draxl, *Bethe–Salpeter equation for absorption and scattering spectroscopy: implementation in the exciting code*, Electron. Struct., 1 (2019), p. 037001.

[51] V. Šego, *Two-sided hyperbolic SVD*, Linear Algebra Appl., 433 (2010), pp. 1265–1275.

[52] ——, *The hyperbolic Schur decomposition*, Linear Algebra Appl., 440 (2014), pp. 90–110.

[53] D. Watkins, *The Matrix Eigenvalue Problem*, SIAM, 2007.

[54] I. Zolotarev, *Application of elliptic functions to questions of functions deviating least and most from zero*, Zap. Imp. Akad. Nauk. St. Petersburg, 30 (1877), pp. 1–59. Reprinted in his Collected Works.