

# UC Berkeley

## UC Berkeley Previously Published Works

### Title

Structure preserving parallel algorithms for solving the Bethe–Salpeter eigenvalue problem

### Permalink

<https://escholarship.org/uc/item/2f45g6hj>

### Authors

Shao, Meiyue  
da Jornada, Felipe H  
Yang, Chao  
et al.

### Publication Date

2016

### DOI

10.1016/j.laa.2015.09.036

Peer reviewed

# Structure Preserving Parallel Algorithms for Solving the Bethe–Salpeter Eigenvalue Problem

Meiyue Shao<sup>1</sup>, Felipe H. da Jornada<sup>2,3</sup>, Chao Yang<sup>1</sup>, Jack Deslippe<sup>4</sup>, and Steven G. Louie<sup>2,3</sup>

<sup>1</sup>*Computational Research Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720*

<sup>2</sup>*Department of Physics, University of California, Berkeley, CA 94720*

<sup>3</sup>*Materials Sciences Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720*

<sup>4</sup>*NERSC, Lawrence Berkeley National Laboratory, Berkeley, CA 94720*

September 22, 2015

## Abstract

The Bethe–Salpeter eigenvalue problem is a dense structured eigenvalue problem arising from discretized Bethe–Salpeter equation in the context of computing exciton energies and states. A computational challenge is that at least half of the eigenvalues and the associated eigenvectors are desired in practice. We establish the equivalence between Bethe–Salpeter eigenvalue problems and real Hamiltonian eigenvalue problems. Based on theoretical analysis, structure preserving algorithms for a class of Bethe–Salpeter eigenvalue problems are proposed. We also show that for this class of problems all eigenvalues obtained from the Tamm–Dancoff approximation are overestimated. In order to solve large scale problems of practical interest, we discuss parallel implementations of our algorithms targeting distributed memory systems. Several numerical examples are presented to demonstrate the efficiency and accuracy of our algorithms.

**Keywords:** Bethe–Salpeter equation, Tamm–Dancoff approximation, Hamiltonian eigenvalue problems, structure preserving algorithms, parallel algorithms

## 1 Introduction

The absorption of a photon by a molecular system or solid can promote an electron in an occupied single-particle state (or orbital) to an unoccupied state while keeping the charge neutrality. In the physics community, this process is often described as the simultaneous creation of a negatively charged quasielectron (or simply electron) and a positively charged quasihole (or hole) in the material that was originally in the lowest energy electronic configuration (the ground state). Upon absorbing a photon, the entire molecular or extended system is in an excited state that contains a correlated electron–hole pair, which is referred to as an *exciton*. The amount of energy required to trigger this excitation gives an important characterization of the material. In many-body physics, a two-particle collective excitation is often described by a two-particle Green’s function, with the

excitation energy level being a pole of this function. It has been shown that the two-particle Green's function satisfies an equation often known as the *Bethe–Salpeter equation* (BSE) [28].

The poles of the two-particle Green's function can be obtained by computing the eigenvalues of a Hamiltonian operator  $\mathcal{H}$  associated with this Green's function. It can be shown that, with an appropriate discretization scheme, the finite dimensional representation of the Bethe–Salpeter Hamiltonian has the following block structure

$$H = \begin{bmatrix} A & B \\ -\bar{B} & -\bar{A} \end{bmatrix}, \quad (1)$$

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

$$A = A^H, \quad B = B^T. \quad (2)$$

Here we use  $A^H$  to denote the conjugate transpose of  $A$  and  $B^T$  to denote the transpose of  $B$ . We will refer to an eigenvalue problem of the form (1) with the additional symmetry given by Equation (2) as a *Bethe–Salpeter eigenvalue problem*.

In principle, we are interested in all possible excitation energies, although some excitations are more likely to occur than others. Such likelihood can often be measured in term of what is known as the *spectral density* or *density of states* of  $H$ , which is defined to be the number of eigenvalues per unit energy interval [19], that is,

$$\phi(\omega) = \frac{1}{2n} \sum_{j=1}^{2n} \delta(\omega - \lambda_j). \quad (3)$$

where  $\lambda_j$ 's denote the eigenvalues of  $H$ . This formulation requires all eigenvalues of  $H$  to be real, which is the case for most physical systems. In addition, the optical absorption spectrum

$$\epsilon^+(\omega) = \sum_{j=1}^n \frac{(d_r^H x_j)(y_j^H d_l)}{y_j^H x_j} \delta(\omega - \lambda_j), \quad (4)$$

which can be measured in optical absorption experiments, is also of practical interest. Here  $d_r$  and  $d_l$  are *dipole vectors*,  $x_j$  and  $y_j$  are right and left eigenvectors, respectively, corresponding to the *positive* eigenvalue  $\lambda_j$ . To obtain highly accurate representations of (3) and (4), the computation of all eigenpairs is required. In addition to the optical absorption spectrum defined in (4), the individual pairs of left and right eigenvectors are often desired, since they describe the character of each two-particle excited state.

In general, both  $A$  and  $B$  are dense. The dimension of these matrices is proportional to the product of the number of occupied and unoccupied states, both of which are proportional to the number of electrons  $n_e$  in the system. Hence it can become quite large for large systems that contain many atoms (and electrons).

We are interested in efficient and reliable parallel algorithms for computing all eigenvalues and the corresponding eigenvectors of (1). Because  $H$  is a non-Hermitian matrix, we need to compute both the left and the right eigenvectors.

Although it is possible to treat  $H$  as a general non-Hermitian matrix and use existing parallel algorithms [12, 15] implemented in ScaLAPACK [8] to solve such a problem, this approach does not take advantage of the special structure of the Bethe–Salpeter Hamiltonian and is thus not

efficient. Nor does this approach preserve some desirable properties of the eigenvalues and eigenvectors. Moreover, the current release of ScaLAPACK only has a subroutine that performs a Schur decomposition of a complex non-Hermitian matrix [12].

In the following section, we show that  $H$  belongs to a class of matrices known as *J-symmetric matrices* whose eigenvalues satisfy a special symmetry property. Although several algorithms have been developed for computing the eigenvalues and eigenvectors of this class of matrices, efficient parallel implementations of these algorithms are not available, and are not easy to develop.

In this paper, we develop a special algorithm that can leverage existing parallel computational kernels available in the ScaLAPACK software package. Our algorithm is based on the observation that computing the eigenpairs of (1) is equivalent to computing the eigenpairs of a real Hamiltonian matrix of the form

$$H_r = \begin{bmatrix} \text{Im}(A + B) & -\text{Re}(A - B) \\ \text{Re}(A + B) & \text{Im}(A - B) \end{bmatrix} \quad (5)$$

where  $\text{Re}(A)$  and  $\text{Im}(A)$  denote the real and imaginary parts of  $A$ , respectively. Furthermore, when  $A$  and  $B$  satisfy the property

$$\begin{bmatrix} A & B \\ \bar{B} & \bar{A} \end{bmatrix} \succ 0,^1 \quad (6)$$

which often holds in practice [35], it can be shown that all eigenvalues of  $iH_r$ , and thus of  $H$  are real, and they come in positive and negative pairs [16, 17]. We present an efficient parallel algorithm for computing the positive eigenvalues and the corresponding right eigenvectors. The algorithm makes use of existing kernel in ScaLAPACK as well as a new kernel we developed for computing eigenpairs of a skew symmetric tridiagonal matrix. A simple transformation can be used to obtain the right eigenvectors associated with the negative eigenvalues, as well as all left eigenvectors.

When  $H$  is real, which is the case for systems with real-space inversion symmetry, the Bethe–Salpeter eigenvalue problem can be transformed into a product eigenvalue problem. We propose an efficient and accurate parallel algorithm for solving the product eigenvalue problem.

When facing the challenge of computing the eigenpairs of the non-Hermitian matrix (1), many researchers in the physics community choose to drop the off-diagonal blocks,  $B$  and  $-\bar{B}$ , and compute eigenpairs of the Hermitian matrix  $A$  only. This approach is often known as the *Tamm–Dancoff approximation* (TDA) [10, 27, 29]. We show that when the condition (6) holds, each eigenvalue of  $A$  is an upper bound of the corresponding positive eigenvalue of  $H$  when all eigenvalues of  $A$  and  $H$  are sorted. Our numerical experiment shows that TDA can introduce a non-negligible shift of the spectral density of  $H$ , which is consistent with what has been reported in the physics literature [16, 25, 35].

The rest of the paper is organized as follows. In Section 2, we discuss some basic properties of the Bethe–Salpeter eigenvalue problem. Then in Section 3, we develop structure preserving algorithms built on these properties as well as the additional assumption (6). Finally, we demonstrate the efficiency and accuracy of our proposed algorithms in Section 4 by several examples from physical models.

## 2 Properties of Bethe–Salpeter eigenvalue problems

In this section, we examine the special properties of the Bethe–Salpeter Hamiltonian that allow us to develop efficient algorithms for computing its eigenpairs.

---

<sup>1</sup> $X \succ Y$  means that  $X - Y$  is Hermitian positive definite.

## 2.1 Relation to Hamiltonian eigenvalue problems

We first show that  $H$  belongs to a class of matrices known as  $J$ -symmetric matrices. Let  $J_n$  be the  $2n \times 2n$  skew-symmetric matrix

$$J_n = \begin{bmatrix} 0 & I_n \\ -I_n & 0 \end{bmatrix}. \quad (7)$$

A matrix  $X \in \mathbb{C}^{2n \times 2n}$  is called a  $J$ -symmetric matrix if  $(XJ_n)^\top = XJ_n$  [21, 22]. When  $X$  is real, it is also called a *Hamiltonian* matrix. By definition,  $X$  is  $J$ -symmetric if and only if  $X$  admits the block structure

$$X = \begin{bmatrix} X_{11} & X_{12} \\ X_{21} & -X_{11}^\top \end{bmatrix}$$

with  $X_{12}$  and  $X_{21}$  complex symmetric. It can also be verified that this  $J$ -symmetric structure is preserved under what is called a *complex symplectic similarity transformation*

$$\hat{H} = S^{-1}HS,$$

where  $S \in \mathbb{C}^{2n \times 2n}$  is a *complex symplectic matrix* that satisfies  $S^\top J_n S = J_n$  [22]. These properties are key to the development of several numerical algorithms for computing the eigenvalues of dense  $J$ -symmetric matrices. Examples of these algorithms include the Hamiltonian–Jacobi algorithms [9, 24] and SR-like algorithms [13].

The Bethe–Salpeter Hamiltonian matrix defined in (1) is clearly a  $J$ -symmetric matrix because

$$H = \begin{bmatrix} A & B \\ -\bar{B} & -\bar{A} \end{bmatrix} = \begin{bmatrix} A & B \\ -\bar{B} & -A^\top \end{bmatrix}.$$

Although the algorithms for  $J$ -symmetric matrices can be used to solve the Bethe–Salpeter eigenvalue problems, they do not take advantage of the additional symmetry relationship between the (1,2) and (2,1) blocks in  $H$ . This symmetry leads to the symmetry of  $\Lambda(H)$  as stated in the following theorem.

**Theorem 1** ([4]). *Let  $H$  be of the form (1) with  $A$  Hermitian and  $B$  symmetric. If  $\lambda$  is an eigenvalue of  $H$ , then  $-\lambda$ ,  $\bar{\lambda}$ ,  $-\bar{\lambda}$  are also eigenvalues of  $H$  with the same multiplicity.*

Unfortunately, complex symplectic transformations in general do *not* preserve the structure of Bethe–Salpeter Hamiltonian matrices. To seek a class of fully structure preserving similarity transformations, we show in the following theorem that solving a Bethe–Salpeter eigenvalue problem is equivalent to solving a real Hamiltonian eigenvalue problem. This is the main theoretical result of this paper.

**Theorem 2.** *A Bethe–Salpeter eigenvalue problem can be reduced to a real Hamiltonian eigenvalue problem, and vice versa.*

*Proof.* Let

$$Q = \frac{1}{\sqrt{2}} \begin{bmatrix} I_n & -iI_n \\ I_n & iI_n \end{bmatrix}.$$

Then

$$Q^H \begin{bmatrix} A & B \\ -\bar{B} & -\bar{A} \end{bmatrix} Q = i \begin{bmatrix} \operatorname{Im}(A+B) & -\operatorname{Re}(A-B) \\ \operatorname{Re}(A+B) & \operatorname{Im}(A-B) \end{bmatrix} = -iJ_n M,$$

where

$$M = \begin{bmatrix} \operatorname{Re}(A + B) & \operatorname{Im}(A - B) \\ -\operatorname{Im}(A + B) & \operatorname{Re}(A - B) \end{bmatrix} \quad (8)$$

is a real symmetric matrix. Therefore, any Bethe–Salpeter eigenvalue problem can be converted to a real Hamiltonian eigenvalue problem.

On the other hand, let  $H_r$  be a real  $2n \times 2n$  Hamiltonian matrix of the form

$$H_r = \begin{bmatrix} H_{11} & H_{12} \\ H_{21} & -H_{11}^\top \end{bmatrix},$$

where  $H_{12}^\top = H_{12}$ ,  $H_{21}^\top = H_{21}$ . We set

$$A = \frac{H_{12} - H_{21}}{2} + i \frac{H_{11}^\top - H_{11}}{2}, \quad B = -\frac{H_{12} + H_{21}}{2} - i \frac{H_{11}^\top + H_{11}}{2}.$$

It can be verified that  $A^\mathsf{H} = A$ ,  $B^\top = B$ , and

$$QH_r Q^\mathsf{H} = i \begin{bmatrix} A & B \\ -\bar{B} & -\bar{A} \end{bmatrix}.$$

Therefore, we can convert a real Hamiltonian eigenvalue problem to a Bethe–Salpeter eigenvalue problem. This completes the proof.  $\square$

Theorem 2 fully characterizes the spectral properties of general Bethe–Salpeter eigenvalue problems. Theorem 1 is a direct consequence of Theorem 2. As a result, several existing algorithms (see [5]) for solving Hamiltonian eigenvalue problems can be applied to the matrix  $H_r$  defined in (5).

The primary interest of this paper is drawn to the case when the property (6) holds. In this case, the Bethe–Salpeter eigenvalue problem is essentially a symmetric eigenvalue problem because

$$H = \begin{bmatrix} I_n & 0 \\ 0 & -I_n \end{bmatrix} \begin{bmatrix} A & B \\ \bar{B} & \bar{A} \end{bmatrix} \quad (9)$$

is the product of two Hermitian matrices, and in addition, one of them is positive definite [16]. Therefore,  $H$  is diagonalizable and has real spectrum. In addition, the eigenvectors of  $H$  also admit special structures. These properties are summarized in the following theorem.

**Theorem 3.** *Let  $H$  be of the form (1) satisfying (2) and (6). Then there exist  $X_1, X_2 \in \mathbb{C}^{n \times n}$  and positive numbers  $\lambda_1, \lambda_2, \dots, \lambda_n \in \mathbb{R}$  such that*

$$HX = X \begin{bmatrix} \Lambda_+ & \\ & -\Lambda_+ \end{bmatrix}, \quad Y^\mathsf{H} H = \begin{bmatrix} \Lambda_+ & \\ & -\Lambda_+ \end{bmatrix} Y^\mathsf{H}, \quad Y^\mathsf{H} X = I_{2n},$$

where

$$X = \begin{bmatrix} X_1 & \bar{X}_2 \\ X_2 & \bar{X}_1 \end{bmatrix}, \quad Y = \begin{bmatrix} X_1 & -\bar{X}_2 \\ -X_2 & \bar{X}_1 \end{bmatrix},$$

and  $\Lambda_+ = \operatorname{diag} \{\lambda_1, \dots, \lambda_n\}$ .

*Proof.* Let

$$S = \begin{bmatrix} I_n & 0 \\ 0 & -I_n \end{bmatrix}, \quad \Omega = \begin{bmatrix} A & B \\ \bar{B} & \bar{A} \end{bmatrix}.$$

It follows from (9) that the eigenvalue problem  $Hx = \lambda x$  is equivalent to  $Sx = \lambda^{-1}\Omega x$ , which is a generalized Hermitian-definite eigenvalue problem. It is well known (see, for example, [14, Section 8.7]) that  $S$  and  $\Omega$  are simultaneously diagonalizable by congruence transformations,<sup>2</sup> and hence  $H$  is diagonalizable and has real eigenvalues. According to Theorem 1, there exist positive numbers  $\lambda_1, \lambda_2, \dots, \lambda_n$  such that  $H$  is similar to  $\text{diag}\{\Lambda_+, -\Lambda_+\}$ , where  $\Lambda_+ = \text{diag}\{\lambda_1, \dots, \lambda_n\}$ .

To determine the structure of the eigenvectors of  $H$ , we start with any nonsingular matrix

$$C = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} \in \mathbb{C}^{2n \times 2n}$$

satisfying  $C^H \Omega C = I_{2n}$  and  $C^H S C = \text{diag}\{\Lambda_+, -\Lambda_+\}^{-1}$ . Then  $C^{-1} H C = \text{diag}\{\Lambda_+, -\Lambda_+\}$ . By setting  $X_1 = C_{11} \Lambda_+^{1/2}$  and  $X_2 = C_{21} \Lambda_+^{1/2}$ , we obtain that

$$H \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} \Lambda_+ \quad (10)$$

and

$$X_1^H X_1 - X_2^H X_2 = I_n, \quad (11)$$

It is straightforward to verify that the following equations are equivalent to (10):

$$H \begin{bmatrix} \bar{X}_2 \\ \bar{X}_1 \end{bmatrix} = - \begin{bmatrix} \bar{X}_2 \\ \bar{X}_1 \end{bmatrix} \Lambda_+, \quad (12)$$

$$[X_1^H, -X_2^H] H = \Lambda_+ [X_1^H, -X_2^H], \quad (13)$$

$$[-\bar{X}_2^H, \bar{X}_1^H] H = -\Lambda_+ [-\bar{X}_2^H, \bar{X}_1^H]. \quad (14)$$

Thus we have obtained all right and left eigenvectors of  $H$ . Finally, it follows from (10) and (14) that

$$(\bar{X}_1^H X_2 - \bar{X}_2^H X_1) \Lambda_+ = [-\bar{X}_2^H, \bar{X}_1^H] H \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = -\Lambda_+ (\bar{X}_1^H X_2 - \bar{X}_2^H X_1).$$

Since  $\Lambda_+$  and  $-\Lambda_+$  have no common eigenvalue, the homogeneous Sylvester equation  $Z \Lambda_+ = -\Lambda_+ Z$  has a unique solution  $Z = 0$ . Hence  $\bar{X}_1^H X_2 - \bar{X}_2^H X_1 = 0$ . Combining this result with (11), we conclude that  $Y^H X = I_{2n}$ . This completes the proof.  $\square$

We now consider a relatively simple case in which  $H$  is a real matrix. In this case (1) simplifies to

$$H = \begin{bmatrix} A & B \\ -B & -A \end{bmatrix}, \quad (15)$$

where both  $A$  and  $B$  are  $n \times n$  real symmetric matrices. By definition  $H$  is a real Hamiltonian matrix. Performing a symplectic orthogonal similarity transformation yields a block cyclic form

$$\frac{1}{\sqrt{2}} \begin{bmatrix} I_n & I_n \\ -I_n & I_n \end{bmatrix}^\top \cdot H \cdot \frac{1}{\sqrt{2}} \begin{bmatrix} I_n & I_n \\ -I_n & I_n \end{bmatrix} = \begin{bmatrix} 0 & A+B \\ A-B & 0 \end{bmatrix}.$$

---

<sup>2</sup>By congruence transformation, we mean a linear map on  $\mathbb{C}^{m \times m}$  of the form  $X \mapsto C^H X C$  where  $C \in \mathbb{C}^{m \times m}$  is nonsingular.

This suggests that the eigenvalues of  $H$  are the square roots of the eigenvalues of  $(A + B)(A - B)$ . Since both  $A$  and  $B$  are real, (6) simplifies to that

$$\begin{bmatrix} A & B \\ B & A \end{bmatrix} \succ 0,$$

or equivalently, *both  $A + B$  and  $A - B$  are positive definite*. Under this condition, the real Bethe–Salpeter eigenvalue problem is also known as a *linear response eigenvalue problem* which recently has attracted a lot of attention (see, for example, [2, 3]). In contrast to many recent developments [3, 26] in linear response eigenvalue problems that focus on large sparse eigensolvers, we develop dense eigensolvers for this eigenvalue problem in the next section.

## 2.2 Tamm–Dancoff approximation

When the off-diagonal blocks of the Bethe–Salpeter Hamiltonian are set to zero, known in the physics community as the Tamm–Dancoff approximation [10, 27, 29], the Bethe–Salpeter eigenvalue problem reduces to a Hermitian eigenvalue problem. One can use efficient algorithms available in ScaLAPACK to compute eigenpairs of  $A$ . In many cases, the results are found to be sufficiently close to the eigenvalues of the full Bethe–Salpeter Hamiltonian and explain experiment. However, in general, this simplification can lead to noticeable difference in the computed spectrum.

In this subsection, we show that Tamm–Dancoff approximation consistently overestimate the positive eigenvalues when the property (6) holds. More precisely, we have

$$\lambda_j(H) \leq \lambda_j(A)$$

for  $j = 1, 2, \dots, n$ , where  $\lambda_j(\cdot)$  denote the  $j$ th *largest* eigenvalue. That is, *every* positive eigenvalue obtained by TDA is greater than or equal to the corresponding one of  $H$ . This theoretical result is consistent with several computational experiments reported in [16, 25]. However, we have not been able to find a rigorous proof of such a result in physics literature. To the best of our knowledge, this result is not well known in the numerical linear algebra community, and its proof is not entirely trivial.

We provide a proof of this important property which we state clearly in Theorem 4. Our proof makes use of the following lemma which appeared relatively recently in [6].

**Lemma 1** ([6]). *Let  $A_1, A_2 \in \mathbb{C}^{n \times n}$  be Hermitian positive definite. Then*

$$\sqrt{\lambda_j(A_1 A_2)} \leq \lambda_j\left(\frac{A_1 + A_2}{2}\right),$$

for  $j = 1, 2, \dots, n$ .

With the help of this arithmetic–geometric inequality on eigenvalues, we prove the claim we have made in the following theorem.

**Theorem 4.** *Let  $H$  be as defined in (1). Then under the conditions (2) and (6), we have*

$$\lambda_j(H) \leq \lambda_j(A)$$

for  $j = 1, 2, \dots, n$ .



*Proof.* Let

$$\tilde{A} = \begin{bmatrix} \operatorname{Re}(A) & \operatorname{Im}(A) \\ -\operatorname{Im}(A) & \operatorname{Re}(A) \end{bmatrix}, \quad \tilde{B} = \begin{bmatrix} \operatorname{Re}(B) & -\operatorname{Im}(B) \\ -\operatorname{Im}(B) & -\operatorname{Re}(B) \end{bmatrix}. \quad (16)$$

Then

$$\frac{1}{\sqrt{2}} \begin{bmatrix} I_n & -iI_n \\ I_n & iI_n \end{bmatrix}^H \cdot \begin{bmatrix} A & B \\ \bar{B} & \bar{A} \end{bmatrix} \cdot \frac{1}{\sqrt{2}} \begin{bmatrix} I_n & -iI_n \\ I_n & iI_n \end{bmatrix} = \tilde{A} + \tilde{B},$$

indicating that (6) is equivalent to  $\tilde{A} + \tilde{B} \succ 0$ . Notice that

$$\tilde{A} + \tilde{B} = J_n^T (\tilde{A} - \tilde{B}) J_n.$$

Therefore,  $\tilde{A} - \tilde{B}$  is also positive definite. In the proof of Theorem 2, we have shown that  $H$  is unitarily similar to  $-iJ_n(\tilde{A} + \tilde{B})$ . Since the eigenvalues of  $H$  are real and appear in pairs  $\{\lambda_j(H), -\lambda_j(H)\}$ , we obtain

$$0 < \lambda_j(H) = \lambda_j(-iJ_n(\tilde{A} + \tilde{B})) = \sqrt{\lambda_{2j}([-iJ_n(\tilde{A} + \tilde{B})]^2)} = \sqrt{\lambda_{2j}((\tilde{A} - \tilde{B})(\tilde{A} + \tilde{B}))}.$$

Applying Lemma 1 yields

$$\sqrt{\lambda_{2j}((\tilde{A} - \tilde{B})(\tilde{A} + \tilde{B}))} \leq \lambda_{2j}(\tilde{A}).$$

Finally, since  $\Lambda(\tilde{A}) = \Lambda(A) \cup \Lambda(\bar{A}) = \Lambda(A) \cup \Lambda(A)$ , that is, the eigenvalues of  $\tilde{A}$  are the same as those of  $A$  with doubled multiplicity, we arrive at

$$\lambda_{2j}(\tilde{A}) = \lambda_j(A).$$

The theorem is thus proved.  $\square$

### 3 Algorithms and implementations

In this section, we present structure preserving algorithms for solving the Bethe–Salpeter eigenvalue problem. As we have shown in Theorem 2, Bethe–Salpeter eigenvalue problems are equivalent to real Hamiltonian eigenvalue problems. Thus any Hamiltonian eigensolver (see, for example, [5]) can always be used to solve this type of eigenvalue problem. However, when (6) holds, a more efficient algorithm can be used to solve the Bethe–Salpeter eigenvalue problem. Throughout this section the condition (6) is assumed to be satisfied. In most cases, only the positive eigenvalues and the corresponding eigenvectors are required for studying the properties of materials. We will demonstrate how to compute them efficiently.

#### 3.1 Complex Bethe–Salpeter eigenvalue problems

From (9), we can write

$$Hx = \lambda x \quad \iff \quad \begin{bmatrix} I_n & 0 \\ 0 & -I_n \end{bmatrix} x = \frac{1}{\lambda} \begin{bmatrix} A & B \\ \bar{B} & \bar{A} \end{bmatrix} x.$$

A straightforward approach is to feed this problem to a generalized Hermitian–definite eigensolver (for example, ZHEGV in LAPACK [1]). However, this approach is in general *not* structure preserving,

meaning that the computed eigenvalues and eigenvectors may not have the properties described in Theorem 3. To see this, we analyze the algorithm implemented in LAPACK's ZHEGV, which first computes the Cholesky factorization

$$\begin{bmatrix} A & B \\ \bar{B} & \bar{A} \end{bmatrix} = LL^H$$

and then applies a standard Hermitian eigensolver to the transformed problem

$$L \begin{bmatrix} I_n & 0 \\ 0 & -I_n \end{bmatrix} L^H.$$

Let  $L + \Delta L$  be the computed Cholesky factor. Then the backward error of  $H$  is

$$\begin{bmatrix} \Delta_{11} & -\Delta_{21}^H \\ \Delta_{21} & \Delta_{22} \end{bmatrix} := \begin{bmatrix} I_n & 0 \\ 0 & -I_n \end{bmatrix} \left( (L + \Delta L)(L + \Delta L)^H - \begin{bmatrix} A & B \\ \bar{B} & \bar{A} \end{bmatrix} \right),$$

which does not necessarily satisfy  $\Delta_{21}^T = \Delta_{21}$  and  $\Delta_{22} = \bar{\Delta}_{11}$ . Therefore, the structure of  $H$  is destroyed in the sense that the error introduced in the computed Cholesky factorization cannot be interpreted as a structured backward error of  $H$ . Consequently, the properties given in Theorem 3 are lost. For instance, the eigenvalues of  $(L + \Delta L) \text{diag} \{I_n, -I_n\} (L + \Delta L)$  do not necessarily appear in positive and negative pairs.

To develop a structure preserving approach, we make use of the observation we made in Section 2. We have observed that  $Q^H H Q = -iJ_n M$ , where

$$Q = \frac{1}{\sqrt{2}} \begin{bmatrix} I_n & -iI_n \\ I_n & iI_n \end{bmatrix}, \quad M = \begin{bmatrix} \text{Re}(A + B) & \text{Im}(A - B) \\ -\text{Im}(A + B) & \text{Re}(A - B) \end{bmatrix} \succ 0.$$

Notice that both  $J_n$  and  $M$  are real matrices. Thus by working with these matrices, we can avoid the use of complex arithmetic. The basic steps of our algorithm are outlined in Algorithm 1, in which we only compute the positive eigenvalues and the corresponding eigenvectors. Once the matrix  $M$  has been constructed, we perform a Cholesky factorization,  $M = LL^T$  to transform the non-Hermitian matrix,  $-iJ_n LL^T$ , into the Hermitian matrix,  $-iL^T J_n L$ . Let  $\Delta L$  be the error in the computed Cholesky factor  $L$ . Then

$$\begin{bmatrix} \Delta_{11} & \Delta_{21}^T \\ \Delta_{21} & \Delta_{22} \end{bmatrix} := (L + \Delta L)(L + \Delta L)^T - M$$

is the backward error of  $M$ , and is real and symmetric. Similar to the proof of Theorem 2, we set

$$\Delta A = \frac{\Delta_{11} + \Delta_{22}}{2} - i \frac{\Delta_{21} - \Delta_{21}^T}{2}, \quad \Delta B = \frac{\Delta_{11} - \Delta_{22}}{2} - i \frac{\Delta_{21} + \Delta_{21}^T}{2}.$$

Then we have  $(\Delta A)^H = \Delta A$ ,  $(\Delta B)^T = \Delta B$ , and

$$\begin{bmatrix} \Delta_{11} & \Delta_{21}^T \\ \Delta_{21} & \Delta_{22} \end{bmatrix} = \begin{bmatrix} \text{Re}(\Delta A + \Delta B) & \text{Im}(\Delta A - \Delta B) \\ -\text{Im}(\Delta A + \Delta B) & \text{Re}(\Delta A - \Delta B) \end{bmatrix},$$

indicating that the backward error in the computed Cholesky factorization of  $M$  can be converted to a structured backward error of  $H$ . We remark that this analysis is valid even if  $\Delta L$  is *not* lower triangular.

---

**Algorithm 1** Algorithm for the complex Bethe–Salpeter eigenvalue problem

---

**Input:**  $A = A^H, B = B^T \in \mathbb{C}^{n \times n}$  such that (6) is satisfied.

**Output:**  $X_1, X_2 \in \mathbb{C}^{n \times n}$  and  $\Lambda_+ = \text{diag} \{\lambda_1, \dots, \lambda_n\}$  satisfying

$$H \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} \Lambda_+, \quad X_1^H X_1 - X_2^H X_2 = I_n,$$

and  $\lambda_i > 0$  for  $i = 1, \dots, n$ .

1: Construct

$$M = \begin{bmatrix} \text{Re}(A + B) & \text{Im}(A - B) \\ -\text{Im}(A + B) & \text{Re}(A - B) \end{bmatrix}.$$

2: Compute the Cholesky factorization  $M = LL^T$ .

3: Construct  $W = L^T J_n L$ , where  $J_n$  is defined in (7).

4: Compute the spectral decomposition  $-iW = [Z_+, Z_-] \text{diag} \{\Lambda_+, -\Lambda_+\} [Z_+, Z_-]^H$ .

5: Set

$$\begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \begin{bmatrix} I_n & 0 \\ 0 & -I_n \end{bmatrix} QLZ_+ \Lambda_+^{-1/2}.$$


---

In the next step we form  $L^T J_n L$ , which is a real skew-symmetric matrix whose spectrum is symmetric with respect to the origin. Applying a skew-symmetric eigensolver (for example, the one in [30, 31]) to this matrix preserves the structure of  $\Lambda(H)$  and avoids the explicit use of complex arithmetic. We will discuss the use of a skew-symmetric eigensolver in detail in Section 3.3. Since any  $2n \times 2n$  real skew-symmetric matrix is of the form  $C^T J_n C$ , where  $C \in \mathbb{R}^{2n \times 2n}$ , the skew-symmetric backward errors introduced in the construction of  $L^T J_n L$  and in the skew-symmetric eigensolver can be interpreted as a backward error (which does not need to be lower triangular) of  $L$ . We have shown that the error in  $L$  can be converted to a structured backward error of  $H$ . Thus this approach is structure preserving. As a remark, we mention that there exists an alternative approach that computes an SVD-like decomposition of  $L$  and avoids the explicit construction of  $L^T J_n L$ , see [33, 34] for details.

Once the eigenvalues and eigenvectors of  $L^T J_n L$  have been computed, the eigenvectors of  $H$  can be recovered by simply accumulating all similarity transformations. Complex arithmetic can also be avoided in this step by carefully manipulating the real and imaginary parts of the eigenvectors. As we have shown in Theorem 3, the left eigenvectors, as well as the eigenvectors corresponding to negative eigenvalues, can be restored with negligible effort according to (12)–(14) if needed.

### 3.2 Real Bethe–Salpeter eigenvalue problems

As we have seen in Section 2, the real Bethe–Salpeter eigenvalue problem can be reduced to a product eigenvalue problem. Directly feeding  $A + B$  and  $A - B$  to a symmetric–definite eigensolver squares the eigenvalues and can potentially spoil the accuracy of the eigenvalues and eigenvectors. When the accuracy requirement is high, an alternative approach is needed. Suppose that  $A + B = L_1 L_1^T$ ,  $A - B = L_2 L_2^T$ . Then  $(A + B)(A - B)$  is similar to  $(L_2^T L_1)(L_2^T L_1)^T$ . Notice that the spectral decomposition of  $(L_2^T L_1)(L_2^T L_1)^T$  can be obtained from the singular value decomposition of  $L_2^T L_1$ . Theorem 5 summarizes this observation.

---

**Algorithm 2** Algorithm for the real Bethe–Salpeter eigenvalue problem

---

**Input:**  $A = A^\top, B = B^\top \in \mathbb{R}^{n \times n}$  such that  $A + B \succ 0$  and  $A - B \succ 0$ .

**Output:**  $X_1, X_2 \in \mathbb{R}^{n \times n}$  and  $\Lambda_+ = \text{diag}\{\lambda_1, \dots, \lambda_n\}$  satisfying

$$H \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} \Lambda_+, \quad X_1^\top X_1 - X_2^\top X_2 = I_n,$$

and  $\lambda_i > 0$  for  $i = 1, \dots, n$ .

- 1: Compute the Cholesky factorizations  $A + B = L_1 L_1^\top, A - B = L_2 L_2^\top$ .
  - 2: Compute the singular value decomposition  $L_2^\top L_1 = U \Lambda_+ V^\top$ .
  - 3: Set  $X_1 = L_2 U + L_1 V, X_2 = L_2 U - L_1 V$ .
- 

**Theorem 5.** Let  $H$  be defined by (15) satisfying that  $A + B \succ 0, A - B \succ 0$ . Suppose that  $L_2^\top L_1 = U \Lambda_+ V^\top$  is the singular value decomposition of  $L_2^\top L_1$ , where  $L_1, L_2 \in \mathbb{R}^{n \times n}$  satisfy that  $L_1 L_1^\top = A + B$  and  $L_2 L_2^\top = A - B$ . Then the spectral decomposition of  $H$  is given by

$$H = X \begin{bmatrix} \Lambda_+ & 0 \\ 0 & -\Lambda_+ \end{bmatrix} Y^\top,$$

where

$$X = \frac{1}{2} \begin{bmatrix} L_2 U + L_1 V & L_2 U - L_1 V \\ L_2 U - L_1 V & L_2 U + L_1 V \end{bmatrix} \begin{bmatrix} \Lambda_+^{-1/2} & 0 \\ 0 & \Lambda_+^{-1/2} \end{bmatrix}$$

and

$$Y = X^{-\top} = \frac{1}{2} \begin{bmatrix} L_1 V + L_2 U & L_1 V - L_2 U \\ L_1 V - L_2 U & L_1 V + L_2 U \end{bmatrix} \begin{bmatrix} \Lambda_+^{-1/2} & 0 \\ 0 & \Lambda_+^{-1/2} \end{bmatrix}.$$

*Proof.* It suffices to verify that  $HX = X \text{diag}\{\Lambda_+, -\Lambda_+\}$  and  $Y^\top X = I_{2n}$ , which only require simple algebraic manipulations.  $\square$

Based on this theorem, we propose Algorithm 2 as a real Bethe–Salpeter eigensolver. We remark that such an eigensolver can also be used as a dense kernel within structure preserving projection methods for linear response eigenvalue problems [3, 18]. Just like the complex case, the negative eigenvalues and the corresponding eigenvectors, if needed, can be easily constructed from the outputs of Algorithm 2 according to Theorems 3 and 5.

### 3.3 Parallel implementations

To solve large scale Bethe–Salpeter eigenvalue problems arising from quantum physics, parallelization of Algorithms 1 and 2 on distributed memory systems is required in practical computations. All  $O(n^3)$  operations in Algorithm 2 consist of basic linear algebra operations, and can be accomplished by calling linear algebra libraries such as BLAS/LAPACK. There also exist ScaLAPACK subroutines for these operations, which allow us to parallelize the algorithm in a straightforward manner. So we will mainly discuss implementation issues for Algorithm 1.

The main obstacle to efficiently implementing Algorithm 1 is the lack of a skew-symmetric eigensolver in LAPACK/ScaLAPACK. The algorithm described in [30, 31] is based on level 1 BLAS operations, hence is not efficient on modern architectures with memory hierarchy. Therefore, we

---

**Algorithm 3** Parallel implementation of steps 4 and 5 in Algorithm 1
 

---

- 1: Tridiagonal reduction:  $W = UTU^T$ .
- 2: Compute the positive spectral decomposition  $-iD^H T D = [V_+, V_-] \text{diag} \{\Lambda_+, -\Lambda_+\} [V_+, V_-]^T$ , where  $D = \text{diag} \{1, i, i^2, \dots, i^{2n}\}$ .
- 3: Construct  $\Phi = LU$  by applying a sequence of Householder reflections.
- 4: Construct  $\hat{Y} = Q\Phi D$ .
- 5: Construct  $Y_+ = \hat{Y}V_+$  using PBLAS.
- 6: Set

$$\begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \begin{bmatrix} I_n & 0 \\ 0 & -I_n \end{bmatrix} Y_+.$$


---

have to implement step 4 in Algorithm 1 by ourselves. To make use of ScaLAPACK as much as possible, we propose the following strategy shown in Algorithm 3. Several remarks on the implementation are in order:

1. Tridiagonal reduction of a skew-symmetric matrix can be accomplished by applying a sequence of Householder reflections. This is in fact slightly simpler compared to reducing a symmetric matrix to tridiagonal form. We implemented a modified version of ScaLAPACK's PDSYTRD to achieve this goal. Several BLAS/PBLAS-like subroutines for skew-symmetric matrix operations are also implemented.
2. Suppose that

$$T = \text{tridiag} \begin{Bmatrix} \alpha_1 & & \cdots & & \alpha_{2n-1} & \\ 0 & & \cdots & & & 0 \\ & -\alpha_1 & & \cdots & & \\ & & & \cdots & & \\ & & & & -\alpha_{2n-1} & \end{Bmatrix}.$$

Then

$$-iD^H T D = \text{tridiag} \begin{Bmatrix} \alpha_1 & & \cdots & & \alpha_{2n-1} & \\ 0 & & \cdots & & & 0 \\ & \alpha_1 & & \cdots & & \\ & & & \cdots & & \\ & & & & \alpha_{2n-1} & \end{Bmatrix}$$

is a real symmetric tridiagonal matrix, whose spectral decomposition can be easily computed by calling ScaLAPACK. This technique is essentially the same as the one in [31]. We then use the bisection method (PDSTEBZ/PDSTEIN) to compute the positive eigenvalues and the corresponding eigenvectors. The eigenvectors are reorthogonalized when the accuracy requirement is high.

3. Step 4 in Algorithm 3 is *not* performed by PBLAS. Because  $Q$  and  $D$  are both known, the application of these unitary transformations can be accomplished with  $O(n^2)$  operations.
4. In Step 5 of Algorithm 3, we separate the computation for real and imaginary parts:  $\text{Re}(X_+) = \text{Re}(\hat{X})V_+$ ,  $\text{Im}(X_+) = \text{Im}(\hat{X})V_+$ . This is based on the fact that  $V_+$  is real, and one PZGEMM call is about twice as expensive as two PDGEMM calls.

Finally, we remark that our parallel algorithms/implementations are just proof-of-concept. There is certainly room for improvement. For example, our tridiagonal reduction step is modified from ScaLAPACK's PDSYTRD, which is relatively simple, but is not state-of-the-art. Many

modern implementations of symmetric tridiagonal reduction are based on successive band reduction [7, 20, 23]. The successive band reduction technique extends naturally to skew-symmetric matrices. There are also many alternative tridiagonal eigensolvers other than the bisection method, for example, the MRRR method [11, 32]. But improvements in these directions exceed the scope of this paper.

## 4 Numerical experiments

In this section, we present numerical examples for three matrices obtained from discretized Bethe–Salpeter equations. The numerical experiments are performed on the Cray XE6 machine, Hopper, at the National Energy Research Scientific Computing Center (NERSC).<sup>3</sup> Each Hopper node consists of two twelve-core AMD ‘MagnyCours’ 2.1-GHz processors, and has 32 GB DDR3 1333-MHz memory. Each core has its own L1 and L2 caches, with 64 KB and 512 KB, respectively. Hopper’s compute nodes are connected via a 3D torus Cray Gemini network with a maximum bandwidth of 8.3 gigabytes per second. The internode latency ranges from 1.27 microseconds to 1.38 microseconds. The latency between cores is less than 1 microsecond.

The examples we use here correspond to discretized Bethe–Salpeter Hamiltonians associated with naphthalene, gallium arsenide (GaAs), and boron nitride (BN), respectively. The dimensions of these  $2n \times 2n$  Hamiltonians are  $64 \times 64$ ,  $256 \times 256$ , and  $4608 \times 4608$ , respectively. We implemented Algorithm 1 in Fortran 90, using the message passing interface (MPI) for parallelization. We used ScaLAPACK to perform some basic parallel matrix computations. No multithreading features such as OpenMP or threaded linear algebra libraries are used. To make fair comparisons, all eigenvalues and eigenvectors are calculated.

We first compare our implementation of Algorithm 1 with LAPACK’s non-Hermitian eigensolver ZGEEV. Test runs are performed using a single core so that both solvers are sequential. From Table 1 we see that both solvers produce solutions with small residuals, and Algorithm 1 is in general more accurate. We also compute the eigenvalues of  $A$  using LAPACK’s Hermitian eigensolver ZHEEV. In Figure 1, we plot the spectral density of the computed eigenvalues with and without TDA. The delta function in (3) is approximated using the Gaussian function, that is,

$$\delta(t) \approx \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{t^2}{2\sigma^2}\right),$$

with standard deviation  $\sigma = 5 \times 10^{-4}$ . The figure illustrates the difference between  $\Lambda(A)$  and  $\Lambda(H)$  for the naphthalene example. We observe up to 12% relative differences in eigenvalues in this example when TDA is used. We also see from the figure that  $\Lambda(A)$  is always to the right of  $\Lambda(H)$ . For the other two examples, the error introduced by TDA are up to 1.2% and 0.93%, respectively. Computational results confirm that all eigenvalues obtained by TDA are larger than the true eigenvalues as predicted by Theorem 4. We remark that ZGEEV produces quite accurate eigenvalues in these examples, despite the fact that all computed eigenvalues are not real.

Table 2 contains the execution time of different approaches. Algorithm 1 is about five times faster than the non-Hermitian solver ZGEEV. An interesting observation is that TDA is *not* much faster than our full-BSE solver, especially when  $n$  gets large. This is not a surprising result. In fact, the major cost of Algorithm 1 is diagonalizing a real  $2n \times 2n$  skew-symmetric matrix, which is comparable to the cost of diagonalizing a complex  $n \times n$  Hermitian matrix.

<sup>3</sup><https://www.nersc.gov/users/computational-systems/hopper/>

Table 1: Comparison between ZGEEV and Algorithm 1.

		naphthalene	GaAs	BN
		$n = 32$	$n = 128$	$n = 2304$
ZGEEV	$\ Y^H H X - \Lambda\ _F / \ H\ _F$	$3.8 \times 10^{-15}$	$8.5 \times 10^{-15}$	$2.0 \times 10^{-14}$
	$\ Y^H X - I_{2n}\ _F / \sqrt{2n}$	$2.8 \times 10^{-15}$	$9.0 \times 10^{-15}$	$2.1 \times 10^{-14}$
Algorithm 1	$\ Y^H H X - \Lambda\ _F / \ H\ _F$	$1.5 \times 10^{-15}$	$3.3 \times 10^{-15}$	$5.4 \times 10^{-15}$
	$\ Y^H X - I_{2n}\ _F / \sqrt{2n}$	$1.1 \times 10^{-15}$	$3.1 \times 10^{-15}$	$4.3 \times 10^{-15}$

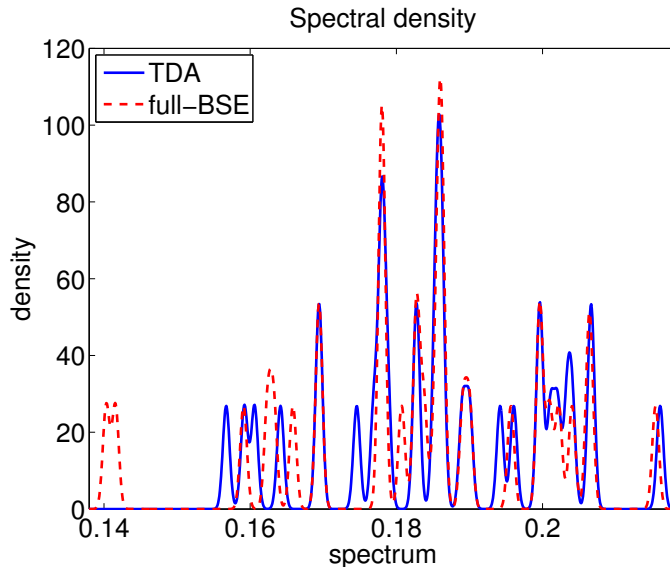


Figure 1: Comparison of the spectral density, defined by Equation (3), for the full Bethe–Salpeter eigenvalue problem and the BSE constructed without the off-diagonal blocks, with the Tamm–Dancoff approximation.

Table 2: Execution time of different approaches.

	naphthalene	GaAs	BN
	$n = 32$	$n = 128$	$n = 2304$
ZGEEV	$1.8 \times 10^{-2}$	$4.8 \times 10^{-1}$	$1.2 \times 10^3$
Algorithm 1	$4.1 \times 10^{-3}$	$7.6 \times 10^{-2}$	$1.6 \times 10^2$
ZHEEV (TDA)	$1.6 \times 10^{-3}$	$5.2 \times 10^{-2}$	$2.5 \times 10^2$

Finally, we perform a simple scalability test on our parallel implementation of Algorithm 1. We use the BN example since it is of moderate size. Test runs are performed on  $1 \times 1$ ,  $2 \times 2$ ,  $\dots$ ,  $9 \times 9$ ,  $10 \times 10$  processor grids, with block factor  $n_b = 64$  for the 2D block cyclic data layout. In Figure 2, we illustrate the overall execution time, as well as the performance profile for each component of the algorithm. We can see from the figure that all components scale similarly in this example. This indicates that Algorithm 1 scales reasonably well as the overall parallel scalability is close to that

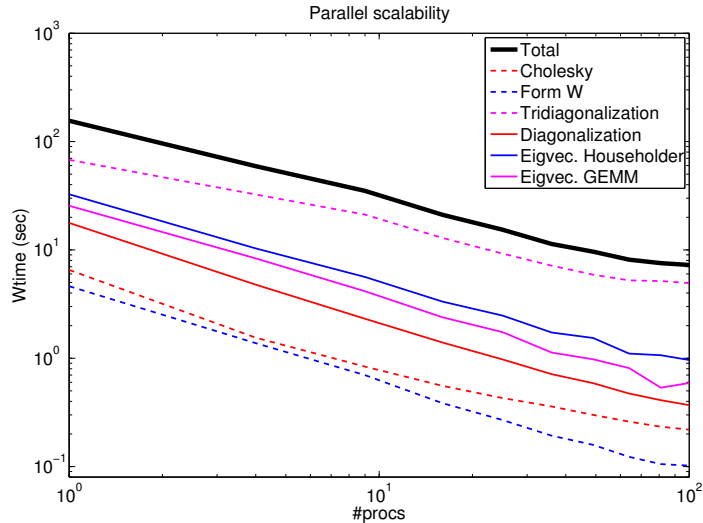


Figure 2: Parallel scalability of Algorithm 1 for BN ( $H \in \mathbb{C}^{4608 \times 4608}$ ).

of the PDGEMM component.

## 5 Conclusions

We showed, in this paper, that the Bethe–Salpeter eigenvalue problem is equivalent to a real Hamiltonian eigenvalue problem and can always be solved by an efficient Hamiltonian eigensolver. For complex Bethe–Salpeter Hamiltonians that satisfy the additional property (6), which almost always holds in practice, it is possible to compute all of its eigenpairs by a structure preserving algorithm we developed in this paper. When the Hamiltonian is real, we can turn the eigenvalue problem into a product eigenvalue problem. We presented an efficient and reliable way to solve this eigenvalue problem.

When the Tamm–Dancoff approximation is used, the Bethe–Salpeter eigenvalue problem reduces to a Hermitian eigenvalue problem that can be solved by existing tools in the ScaLAPACK library. We showed that Tamm–Dancoff approximation always overestimate all eigenvalues.

We presented numerical algorithms that demonstrated the accuracy and efficiency of our algorithm. However, we should point out that our parallel implementation is preliminary, and there is plenty of room for improvements.

## Acknowledgments

The authors thank Zhaojun Bai, Peter Benner, Fabien Bruneval, Heike Faßbender, Daniel Kressner, and Hongguo Xu for fruitful discussions. The authors are also grateful to anonymous referees for careful reading and providing valuable comments. This material is based upon work supported by the Scientific Discovery through Advanced Computing (SciDAC) Program on Excited State



Phenomena in Energy Materials funded by the U.S. Department of Energy, Office of Science, under SciDAC program, the Offices of Advanced Scientific Computing Research and Basic Energy Sciences contract number DE-AC02-05CH11231 at the Lawrence Berkeley National Laboratory. This research used resources of the National Energy Research Scientific Computing Center, a DOE Office of Science User Facility supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231.

## References

- [1] E. Anderson, Z. Bai, C. H. Bischof, L. S. Blackford, J. W. Demmel, J. J. Dongarra, J. J. Du Croz, A. Greenbaum, S. J. Hammarling, A. McKenney, and D. C. Sorensen. *LAPACK User's Guide*. SIAM, Philadelphia, PA, USA, 3rd edition, 1999.
- [2] Z. Bai and R.-C. Li. Minimization principles for the linear response eigenvalue problem I: Theory. *SIAM J. Matrix Anal. Appl.*, 33(4):1075–1100, 2012.
- [3] Z. Bai and R.-C. Li. Minimization principles for the linear response eigenvalue problem II: Computation. *SIAM J. Matrix Anal. Appl.*, 34(2):392–416, 2013.
- [4] P. Benner, H. Fassbender, and C. Yang. Some remarks on the complex  $J$ -symmetric eigenproblem. Preprint MPIMD/15-12, Max Planck Institute Magdeburg, 2015. Available from <http://www.mpi-magdeburg.mpg.de/preprints/>.
- [5] P. Benner, D. Kressner, and V. Mehrmann. Skew-Hamiltonian and Hamiltonian eigenvalue problems: Theory, algorithms and applications. In Z. Drmač, M. Marušić, and Z. Tutek, editors, *Proceedings of the Conference on Applied Mathematics and Scientific Computing*, pages 3–39. Springer Netherlands, 2005.
- [6] R. Bhatia and F. Kittaneh. Notes on matrix arithmetic–geometric mean inequalities. *Linear Algebra Appl.*, 308:203–211, 2000.
- [7] C. H. Bischof, B. Lang, and X. Sun. Algorithm 807: The SBR toolbox—software for successive band reduction. *ACM Trans. Math. Software*, 26(4):602–616, 2000.
- [8] L. S. Blackford, J. Choi, A. Cleary, E. D’Azevedo, J. W. Demmel, I. Dhillon, J. J. Dongarra, S. J. Hammarling, G. Henry, A. Petitet, K. Stanley, D. W. Walker, and R. C. Whaley. *ScaLAPACK User's Guide*. SIAM, Philadelphia, PA, USA, 1997.
- [9] R. Byers. A Hamiltonian–Jacobi algorithm. *IEEE Trans. Automat. Control*, 35(5):566–570, 1990.
- [10] S. M. Dancoff. Non-adiabatic meson theory of nuclear forces. *Phys. Rev.*, 78(4):382–385, 1950.
- [11] I. Dhillon, B. N. Parlett, and C. Vömel. The design and implementation of the MRRR algorithm. *ACM Trans. Math. Software*, 32(4):533–560, 2006.
- [12] M. R. Fahey. Algorithm 826: A parallel eigenvalue routine for complex Hessenberg matrices. *ACM Trans. Math. Software*, 29(3):326–336, 2002.

- [13] H. Faßbender. The parameterized SR algorithm for Hamiltonian matrices. *Electron. Trans. Numer. Anal.*, 26:121–145, 2007.
- [14] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, MD, USA, 3rd edition, 1996.
- [15] R. Granat, B. Kågström, D. Kressner, and M. Shao. Algorithm xxx: Parallel library software for the multishift QR algorithm with aggressive early deflation. *ACM Trans. Math. Software*, (to appear).
- [16] M. Grüning, A. Marini, and X. Gonze. Exciton–plasmon states in nanoscale materials: Breakdown of the Tamm–Dancoff approximation. *Nano Lett.*, 9(8):2820–2824, 2009.
- [17] M. Grüning, A. Marini, and X. Gonze. Implementation and testing of Lanczos-based algorithms for random-phase approximation eigenproblems. *Comput. Mater. Sci.*, 50:2148–2156, 2011.
- [18] D. Kressner, M. Miloloža Pandur, and M. Shao. An indefinite variant of LOBPCG for definite matrix pencils. *Numer. Algor.*, 66(4):681–703, 2014.
- [19] L. Lin, Y. Saad, and C. Yang. Approximating spectral densities of large matrices. *SIAM Rev.*, (to appear). Also available from <http://arxiv.org/abs/1308.5467>.
- [20] P. Luszczek, H. Ltaief, and J. Dongarra. Two-stage tridiagonal reduction for dense symmetric matrices using tile algorithms on multicore architectures. In *IPDPS 2011: IEEE International Parallel and Distributed Processing Symposium*, pages 944–955. IEEE, 2011.
- [21] D. S. Mackey, N. Mackey, and F. Tisseur. Structured tools for structured matrices. *Electron. J. Linear Algebra*, 10:106–145, 2003.
- [22] D. S. Mackey, N. Mackey, and F. Tisseur. Structured factorizations in scalar product spaces. *SIAM J. Matrix Anal. Appl.*, 27(3):821–850, 2006.
- [23] 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(21):213201, 2014.
- [24] C. Mehl. On asymptotic convergence of nonsymmetric Jacobi algorithms. *SIAM J. Matrix Anal. Appl.*, 30(1):291–331, 2008.
- [25] P. Puschnig, C. Meisenbichler, and C. Ambrosch-Draxl. Excited state properties of organic semiconductors: Breakdown of the Tamm–Dancoff approximation, 2013. arXiv:1306:3790.
- [26] D. Rocca, Z. Bai, R.-C. Li, and G. Galli. A block variational procedure for the iterative diagonalization of non-Hermitian random-phase approximation matrices. *J. Chem. Phys.*, 136:034111(1–8), 2012.
- [27] M. Rohlfing and S. G. Louie. Electron–hole excitations and optical spectra from first principles. *Phys. Rev. B*, 62(8):4927–4944, 2000.
- [28] E. E. Salpeter and H. A. Bethe. A relativistic equation for bounded-state problems. *Phys. Rev.*, 84(6):1232–1242, 1951.

- [29] I. Y. Tamm. Relativistic interaction of elementary particles. *J. Phys. (USSR)*, 9:449–460, 1945.
- [30] R. C. Ward and L. J. Gray. Algorithm 530: Eigensystem computation for skew-symmetric matrices and a class of symmetric matrices [F2]. *ACM Trans. Math. Software*, 4(3):286–289, 1978.
- [31] R. C. Ward and L. J. Gray. Eigensystem computation for skew-symmetric matrices and a class of symmetric matrices. *ACM Trans. Math. Software*, 4(3):278–285, 1978.
- [32] P. R. Willems and B. Lang. A framework for the MR<sup>3</sup> algorithm: Theory and implementation. *SIAM J. Sci. Comput.*, 35(2):A740–A766, 2013.
- [33] H. Xu. An SVD-like matrix decomposition and its applications. *Linear Algebra Appl.*, 368:1–24, 2003.
- [34] H. Xu. A numerical method for computing an SVD-like decomposition. *SIAM J. Matrix Anal. Appl.*, 26(4):1058–1082, 2005.
- [35] R. Zimmermann. Influence of the non-Hermitian splitting terms on excitonic spectra. *Phys. Stat. Sol. (b)*, 41:23–43, 1970.