

Lawrence Berkeley National Laboratory

LBL Publications

Title

A greedy algorithm for computing eigenvalues of a symmetric matrix

Permalink

<https://escholarship.org/uc/item/8sh8v6jh>

Journal

Numerical Linear Algebra with Applications, 28(2)

ISSN

1070-5325

Authors

Hernandez, Taylor M

Van Beeumen, Roel

Caprio, Mark A

et al.

Publication Date

2021-03-01

DOI

10.1002/nla.2341

A greedy algorithm for computing eigenvalues of a symmetric matrix

Taylor M. Hernandez¹, Roel Van Beeumen², Mark A. Caprio¹, Chao Yang^{2,*}

¹Department of Physics, University of Notre Dame, IN, United States

²Computational Research Division, Lawrence Berkeley National Laboratory, CA, United States

Abstract

We present a greedy algorithm for computing selected eigenpairs of a large sparse matrix H that can exploit localization features of the eigenvector. When the eigenvector to be computed is localized, meaning only a small number of its components have large magnitudes, the proposed algorithm identifies the location of these components in a greedy manner, and obtains approximations to the desired eigenpairs of H by computing eigenpairs of a submatrix extracted from the corresponding rows and columns of H . Even when the eigenvector is not completely localized, the approximate eigenvectors obtained by the greedy algorithm can be used as good starting guesses to accelerate the convergence of an iterative eigensolver applied to H . We discuss a few possibilities for selecting important rows and columns of H and techniques for constructing good initial guesses for an iterative eigensolver using the approximate eigenvectors returned from the greedy algorithm. We demonstrate the effectiveness of this approach with examples from nuclear quantum many-body calculations and many-body localization studies of quantum spin chains.

Keywords large-scale eigenvalue problem, eigenvector localization, greedy algorithm, perturbation analysis

1 Introduction

Large-scale eigenvalue problems arise from quantum many-body calculations. In this type of calculation, we are typically interested in a few algebraically smallest eigenvalues of a large sparse symmetric matrix. The dimension of the matrix depends on the number of particles and approximation model parameters. It can grow rapidly with respect to the size of the problem and accuracy requirement. Because only a small fraction of the matrix elements are nonzero, and since only a small number of eigenpairs are desired, iterative methods are often used to solve this type of problem. The dominant cost of these methods is in performing a sparse matrix vector multiplication at each step of the iterative solver. For large problems, performing this calculation efficiently on a high performance computer is a challenging task. Not only do we need to choose an appropriate data structure to represent the sparse matrix, we also need to develop efficient schemes to distribute the matrix and vectors on multiple nodes or processors to overcome the single node memory limitation and enable the computation to be performed in parallel.

It is well known that, for some problems, the eigenvector to be computed has localization properties, i.e., many elements of the desired eigenvector are negligibly small [9, 14]. Physically,

*Lawrence Berkeley National Laboratory, 1 Cyclotron Road, Berkeley, CA 94720. Email: CYang@lbl.gov.

localization means that the many-body operator of interest can be represented by a few many-body basis functions in a small configuration space. This feature of the problem implies that only the rows and columns associated with the large elements of the eigenvectors are important. We can then effectively work with a much smaller matrix by excluding rows and columns associated with small elements in the eigenvector.

However, in general, we do not know which elements of the eigenvector are small (in magnitude) in advance. In some cases, there are efficient numerical procedures that can be used to identify these elements, e.g., the latest work by Arnold et al. [2, 1]. But these techniques generally only work for low dimensional problems in practice. There are sometimes physical intuitions we may use to infer which rows/columns are more important than others. For example, in a configuration interaction approach for quantum many-body problems, the matrix to be diagonalized is the representation of the Hamiltonian in a many-body basis that consists of antisymmetric products (Slater determinants) of a set of single-particle basis functions, e.g., eigenfunctions of a quantum harmonic oscillator. Many-body basis functions defined by single-particle functions associated with lower single-particle energies tend to be more important than others, although this is not always true.

In this paper, we describe a greedy algorithm to incrementally probe large components of a localized eigenvector to be computed. The matrix rows and columns corresponding to these components are extracted to construct a much smaller matrix. The eigenvector of this small matrix is then used to obtain an approximate eigenvector of the original matrix to be diagonalized. If the approximate eigenpair is not sufficiently accurate (the metric for measuring accuracy will be described below), we select some additional rows and columns of the original matrix and solve a slightly larger problem using the solution of the previous problem as the starting guess. This procedure can be repeated recursively until the computed eigenpair is sufficiently accurate.

For problems that are not strictly localized, i.e., many eigenvector components are small but not zero, this approach does not completely eliminate the need to use an iterative method to compute the desired eigenpair of the original matrix. However, the number of iterations required to reach convergence can be significantly reduced if a good starting guess can be constructed from the greedy scheme. If the submatrices selected by the greedy algorithm are relatively small, the cost of computing the desired eigenpairs of these smaller matrices is relatively low. Consequently, the overall cost of the computation can be reduced.

We should note that the greedy algorithm proposed in this paper is different from the hierarchical algorithm presented by Shao et al. [18]. Instead of using a predefined set of hierarchical configuration spaces to construct a sequence of submatrices from which approximate eigenpairs are computed, the greedy algorithm constructs these submatrices dynamically using the previous approximate eigenvector to guide such a construction.

The greedy strategy used to construct a sequence of submatrices from which approximate eigenpairs are computed is similar to the so-called selected configuration interaction approach used in quantum chemistry [22, 12, 23, 17, 7, 5, 19, 20, 6] and the importance truncation scheme used in nuclear physics [16]. But we would like to emphasize that the techniques discussed here are more general. They are not restricted to problems arising from quantum chemistry or physics. Moreover, we describe greedy strategies in terms of matrices and vectors instead of many-body configurations and Hamiltonians. As a result, these strategies can potentially be applied to other applications such as sparse principal component analysis [24].

This paper is organized as follows. In section 2, we discuss the implication of eigenvector localization on the development of an efficient iterative method for computing such an eigenvector, and outline the general strategy for developing such an algorithm. In section 3, we discuss several greedy strategies for selecting rows and columns of the original matrix to construct a submatrix from which approximate eigenpairs are computed and used as a starting guess for computing the eigenpairs of the original problem. Techniques for improving the starting guess are discussed in section 4. In section 5, we present some numerical examples to demonstrate the efficiency of the greedy algorithm. Additional improvement of the algorithm is discussed in section 6.

2 Eigenvector localization and a hierarchical method for computing localized eigenvectors

Let $H \in \mathbb{R}^{n \times n}$ be the symmetric matrix to be diagonalized. To simplify our discussion, let us focus on computing the algebraically smallest eigenvalue λ of H and its corresponding eigenvector x . If the desired eigenvector x is localized, i.e., only a subset of its elements are nonzero, we can reorder the elements of the eigenvector to have all nonzero elements appear in the leading n_1 rows, i.e.,

$$Px = \begin{bmatrix} x_1 \\ 0 \end{bmatrix},$$

where $x_1 \in \mathbb{R}^{n_1}$ and P the permutation matrix associated with such a reordering. Consequently, we can reorder the rows and columns of the matrix H so that

$$(PHP^T)(Px) = \begin{bmatrix} H_1 & B \\ B^T & C \end{bmatrix} \begin{bmatrix} x_1 \\ 0 \end{bmatrix} = \lambda \begin{bmatrix} x_1 \\ 0 \end{bmatrix} \quad (1)$$

holds. To obtain x_1 , we only need to solve the eigenvalue problem

$$H_1 x_1 = \lambda x_1. \quad (2)$$

Even when x is not strictly localized, i.e., the magnitude of the elements in x_1 are significantly larger than the other elements that are small but not necessarily zero, the solution of (2) can be used to construct a good initial guess of x that can be used to accelerate the convergence of an iterative method applied to compute the desired eigenpair of H .

However, since we do not know how large the elements of x are in advance, we do not have the permutation P that allows us to pick rows and columns of H to form H_1 .

The algorithm presented in this paper seeks to identify the permutation P that allows us to construct H_1 incrementally so that successively more accurate approximations to the desired eigenpair can be obtained efficiently. The basic algorithm we use to achieve this goal can be described as follows.

1. We select a subset of the indices $1, 2, \dots, n$ denoted by \mathcal{S} that corresponds to “important” rows and columns of H . In the configuration interaction method for solving quantum many-body eigenvalue problems, this subset may correspond to a set of many-body basis functions produced from some type of basis truncation scheme.
2. Let H_1 be a submatrix of H that consists of rows and columns of H defined by \mathcal{S} . Assuming the size of \mathcal{S} is small relative to n , we can easily compute the desired eigenpairs (λ_1, x_1) of H_1 , i.e., $H_1 x_1 = \lambda_1 x_1$.
3. We take λ_1 to be the approximation to the smallest eigenvalue of H . The approximation to the eigenvector of H is constructed as $\hat{x} = P^T \begin{bmatrix} x_1^T & 0 \end{bmatrix}^T$. To assess the accuracy of the computed eigenpair (λ_1, \hat{x}) , we compute the full residual $r = H\hat{x} - \lambda_1 \hat{x}$.
4. If the norm of r is sufficiently small, we terminate the computation and return (λ_1, \hat{x}) as the approximate solution. Otherwise, we select some additional rows and columns of H to augment H_1 and repeat steps 2–4 again.

If the eigenvector to be computed is localized, this procedure should terminate before the dimension of H_1 becomes very large, assuming that we can identify the most important rows and columns of H in some way.

3 Greedy algorithms for detecting localization

Without loss of generality, we take \mathcal{S} to be the leading n_1 rows and columns of H so that we can partition H as

$$H = \begin{bmatrix} H_1 & B \\ B^T & C \end{bmatrix}. \quad (3)$$

We now discuss how to select additional “important” rows and columns outside of the subset \mathcal{S} to obtain a more accurate approximation of the desired eigenvector of H .

3.1 Residual based approach

Suppose (λ_1, x_1) is the computed eigenpair of the submatrix H_1 that serve as an approximation to the desired eigenpair (λ, x) . By padding x_1 with zeros to form

$$\hat{x} = \begin{bmatrix} x_1 \\ 0 \end{bmatrix}, \quad (4)$$

we can assess the accuracy of the approximate eigenvector \hat{x} in the full space by computing its residual

$$r = H\hat{x} - \lambda_1\hat{x} = \begin{bmatrix} 0 \\ B^T x_1 \end{bmatrix} \equiv \begin{bmatrix} 0 \\ r' \end{bmatrix}. \quad (5)$$

A first greedy scheme for improving the accuracy of x_1 is to select k row and column indices in $\{1, 2, \dots, n\} \setminus \mathcal{S}$ that correspond to components of $r' = B^T x_1$ with the largest magnitude. These indices, along with \mathcal{S} , yield an augmented H_1 from which a more accurate approximation to (λ, x) can be obtained.

3.2 Perturbation analysis based approach

It is possible that a component of r' is large in magnitude even though the magnitude of the corresponding component in the eigenvector x is relatively small, or *vice versa*. Therefore, instead of selecting row and column indices that correspond to the components of the largest magnitude within r' , it may be that a better selection can be made by estimating the magnitude of the components of x whose corresponding indices are outside of \mathcal{S} , and then selecting the row and column indices that correspond to these estimated largest elements.

To do that, let us modify the j th component of the zero block of \hat{x} in (4) and assume the vector

$$\tilde{x} = \begin{bmatrix} x_1 \\ \gamma e_j \end{bmatrix} \quad (6)$$

is a better approximation to the eigenvector x than \hat{x} defined in (4), with the corresponding eigenvalue approximation $\tilde{\lambda} = \lambda_1 + \delta$, where δ is the correction to the eigenvalue, and then e_j is the j th column of the $(n - n_1) \times (n - n_1)$ identity matrix.

Substituting (6) into $Hx = \lambda x$ and examining the $(n_1 + j)$ th row of the equation yields

$$e_j^T B^T x_1 + \gamma e_j^T C e_j = (\lambda_1 + \delta)\gamma. \quad (7)$$

If we drop the second order correction term $\delta\gamma$ and rearrange the equation, we obtain

$$(\lambda_1 - e_j^T C e_j)\gamma \approx e_j^T B^T x_1. \quad (8)$$

As a result, the $(n_1 + j)$ th component of x can be estimated to be

$$\gamma \approx \frac{e_j^T B^T x_1}{\lambda_1 - C_{j,j}}, \quad (9)$$

where $C_{j,j} = e_j^T C e_j$ is the j diagonal element of the matrix C .

The magnitude of this quantity γ in (9), the perturbation analysis estimate for the $(n_1 + j)$ th eigenvector component, is then taken to provide an estimate for the importance of the corresponding row and column of H in a greedy selection approach. If we compare with the corresponding component $e_j^T r' = e_j^T B^T x_1$ of the residual vector r , calculated above in (5) to provide an estimate of the importance of this row and column of H in the residual based greedy selection approach, we see that the quantities used to estimate the importance of a row and column in the two approaches only differ by a scaling factor $|\lambda_1 - C_{j,j}|^{-1}$.

In (6), we limit the perturbation to exactly one component in the zero block of \hat{x} . This is the approach taken in references [6, 22]. We will refer to this type of perturbation as *componentwise perturbation*.

It is conceivable that perturbing several components in this block may result in a better approximation of x . In the extreme case, all components of the zero block can be perturbed to yield a better approximation. In that case, we can express the perturbed approximation to the desired eigenvector as

$$\tilde{x} = \begin{bmatrix} x_1 \\ z \end{bmatrix}. \quad (10)$$

Substituting (10) into $Hx = \lambda x$ and examine the second block of the equation yields

$$B^T x_1 + Cz = (\lambda_1 + \delta)z. \quad (11)$$

Again, if we drop the second order correction term δz and rearrange the equation, we obtain

$$z \approx (\lambda_1 I - C)^{-1} B^T x_1. \quad (12)$$

From (12) we can see that a full correction of the zero component of \hat{x} requires solving a linear equation with the shifted matrix $\lambda_1 I - C$ as the coefficient. This is likely to be prohibitively expensive because the dimension of C is assumed to be much larger than the dimension of H_1 . However, because all we need is the magnitudes of the components of z relative to each other, which we will use to select the next set of rows and columns of B and C to be included in H_1 , we do not necessarily need to solve the linear equation accurately. We will refer to this type of perturbation as *full perturbation*.

There are a number of options to obtain an approximate solution to (11). One possibility is to use an iterative solver such as the minimum residual (MINRES) algorithm [15], and perform a few iterations to obtain an approximation to z . Another possibility is to approximate the matrix C by another matrix that is much easier to invert. For example, if C is diagonally dominant, we can replace C with a diagonal matrix D that contains the diagonal of C . This approach will yield the same selection criterion as that provided by (9). When C is not diagonal dominant, we may also include a few subdiagonal and superdiagonal bands to form a banded matrix approximation to C . Another possibility is to replace C with a block diagonal matrix G with relatively small diagonal blocks. This approach corresponds to perturbing a few rows of the zero block of \hat{x} at a time. In this approach, it is important to block rows and columns of C in such a way that $C = G + E$ for some matrix E that is relatively small (in a matrix norm).

4 Updating the Eigenvector Approximation

Once new row and column indices have been selected using the criteria discussed in the previous section, we update H_1 by including the additional rows and columns of B and C specified by the new row and column indices. We then compute the desired eigenvalue and the corresponding eigenvector of the updated H_1 .

Since we already have the approximate eigenvector x_1 associated with the previous H_1 , we hope to obtain the new approximation quickly by using an iterative method that can take advantage of a good starting guess of the desired eigenvector.

In this paper, we consider both the Lanczos method [10], which extracts approximate eigenpairs from the Krylov subspace

$$\mathcal{K}(H_1, v_0) = \text{span} \{v_0, H_1 v_0, H_1^2 v_0, \dots, H_1^{m-1} v_0\},$$

where v_0 is the starting guess of the desired eigenvector, and the locally optimal block preconditioned conjugate gradient (LOBPCG) method [8]. In the LOBPCG method, the approximate eigenvector $x^{(j)}$ is updated successively according to the following updating formula

$$x^{(j+1)} = \alpha x^{(j)} + \beta P r^{(j)} + \eta x^{(j-1)},$$

where $r^{(j)} = H_1 x^{(j)} - \lambda^{(j)} x^{(j)}$ is the residual associated with the approximate eigenpair $(\lambda^{(j)}, x^{(j)})$, P is a properly chosen preconditioner, and the scalars α , β , and η are chosen to minimize the Rayleigh quotient $\langle x^{(j+1)}, H_1 x^{(j+1)} \rangle$, subject to the normalization constraint $\langle x^{(j+1)}, x^{(j+1)} \rangle = 1$. In addition to its ability to accelerate convergence by incorporating a preconditioner P when one is available, the LOBPCG method can also take advantage of approximations to several eigenvectors simultaneously. However, in this paper, we will focus on computing the lowest eigenvalue of H and its corresponding eigenvector.

There are a number of ways to choose the starting guess for both the Lanczos method and the LOBPCG method. The simplest approach is to construct the starting guess by padding x_1 with additional zeros. Another possibility is to pad x_1 with the largest components (in magnitude) of the approximate solution z defined by (12), especially if (12) is used to select the new rows and columns of B and C to be included in H_1 . This approach may work well if components of x_1 are already very close to the corresponding components in the exact eigenvector x . However, if that is not the case, we need to correct x_1 as well by defining \tilde{x} as

$$\tilde{x} = \begin{bmatrix} x_1 + z_1 \\ z_2 \end{bmatrix}, \quad x_1^T z_1 = 0. \quad (13)$$

Substituting \tilde{x} and $\lambda = \lambda_1 - \delta$ into $Hx = \lambda x$, enforcing the $x_1^T z_1 = 0$ constraint, and dropping the second order perturbation term yields

$$\begin{bmatrix} H_1 & B & x_1 \\ B^T & C - \lambda_1 I & 0 \\ x_1^T & 0 & 0 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ \delta \end{bmatrix} = \begin{bmatrix} 0 \\ -B^T x_1 \\ 0 \end{bmatrix}. \quad (14)$$

Eliminating δ from (14) and applying the projector

$$\begin{bmatrix} I - x_1 x_1^T & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

to both sides of the equation results in

$$\begin{bmatrix} \hat{H}_1 & \hat{B} \\ \hat{B}^T & C - \lambda_1 I \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} 0 \\ -B^T x_1 \end{bmatrix}, \quad (15)$$

where $\hat{H}_1 = (I - x_1 x_1^T)(H_1 - \lambda_1 I)(I - x_1 x_1^T)$ and $\hat{B} = (I - x_1 x_1^T)B$.

We can solve (15) by using an iterative solver such as the MINRES algorithm. Instead of adding z_1 and z_2 directly to $[x_1^T \ 0]^T$ as shown in (13), we can project H into a two-dimensional subspace spanned by

$$Q = \left\{ \begin{bmatrix} x_1 \\ 0 \end{bmatrix}, \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} \right\},$$

and solving a 2×2 eigenvalue problem. If g_1 is the eigenvector associated with the smallest eigenvalue of the projected matrix, the starting guess of the desired eigenvector of H can be chosen as

$$\tilde{x} = Qg_1.$$

We will refer to this approach of preparing the starting guess as the *Newton correction*.

5 Numerical examples

In this section, we demonstrate the effectiveness of the greedy algorithm for computing the lowest eigenvalue of H for two different applications. The first one arises from nuclear structure calculations. The second is concerned with computing the localized eigenvector of a model many-body Hamiltonian that includes local interactions and a disordered potential term.

Before presenting the results of the numerical experiments, we first describe 2 reference calculations, the 3 different variants of the greedy algorithm to be compared, and the Newton correction approach below:

- **original**: reference calculation by directly solving the full problem and using a random vector as initial guess.
- **hierarchy**: reference calculation by exploiting the hierarchical structure of the matrix H , i.e., first solving the small problem, next padding the obtained small eigenvector with zeros and using it as initial guess for the full problem.
- **greedy-res**: greedy algorithm which uses the residual based approach for selecting the row and column indices to augment H with.
- **greedy-pert**: greedy algorithm which uses the componentwise perturbation analysis based approach (9) for selecting the row and column indices to augment H with.
- **greedy-pert-full**: greedy algorithm which uses the full perturbation analysis based approach (12) for selecting the row and column indices to augment H with.
- **newton-corr**: Newton correction approach (13) for updating the eigenvector and initial guess.

5.1 Nuclear Configuration Interaction

The matrix H to be diagonalized in this example is the nuclear many-body Schrödinger Hamiltonian for the nucleus of a lithium atom, in particular, of the isotope ${}^6\text{Li}$, for which the nucleus consists of 3 protons and 3 neutrons. The matrix approximation to the nuclear Schrödinger Hamiltonian operator is constructed on the so-called configuration interaction space, spanned by a set of many-body basis functions.

Each of these many-body basis functions is a Slater determinant of a set of eigenfunctions of a 3D harmonic oscillator. These single-particle eigenfunctions are indexed by a set of quantum numbers $\{n(a), l(a), j(a), m(a)\}$, for each nucleon a , and is associated with number $N(a) = 2n(a) + l(a)$ of oscillator quanta [21]. In the nuclear physics applications, the selection of Slater determinants for the configuration space is often done by specifying a limit on the sum of the oscillator quanta $N_{\text{tot}} = \sum_a [2n(a) + l(a)]$ (some additional constraints are imposed on the quantum numbers to ensure appropriate symmetry properties) [3]. This limit on the oscillator quanta is often expressed in terms of a cutoff parameter N_{max} indicating the limit on the number of quanta permitted *above* the minimal number N_0 possible (that is, consistent with the Pauli principle, or antisymmetry of Slater determinants) for that nucleus: then the many-body basis function is restricted to $N_{\text{tot}} \leq N_0 + N_{\text{max}}$. This constraint defines a truncation relative to the full configuration space, defined by all possible Slater determinants that can be generated, from a given set of harmonic oscillator eigenfunctions. The larger the N_{max} , the larger the dimension of the matrix approximation H to the Hamiltonian, and the higher the cost to obtain the desired eigenpairs of H .

Because we consider only two-body potential interactions, the finite dimensional Hamiltonian constructed from a truncated configuration space is sparse. Figure 1(a) shows the nonzero matrix element pattern of H for the $N_{\text{max}} = 6$ truncation level. The dimension of this matrix is 197,882. The leading 800×800 principal submatrix of H corresponds to the Hamiltonian truncated with $N_{\text{max}} = 2$.

As a reference, we use the LOBPCG algorithm to compute the lowest eigenvalue and its eigenvector of H for $N_{\text{max}} = 6$. We plot the magnitude of its components in Figure 1(b). As we can see, many of these components are small.

We first compare the perturbation analysis based greedy algorithm (**greedy-pert**) to the standard LOBPCG algorithm (**original**) and the hierarchical approach (**hierarchy**). It is clear from Figure 1(b) that the largest components (in magnitude) of the eigenvector appear in the leading portion of the vector that correspond to the configuration space defined by a small N_{max} truncation level. Therefore, we take the leading 800×800 principal submatrix ($N_{\text{max}} = 2$) as the

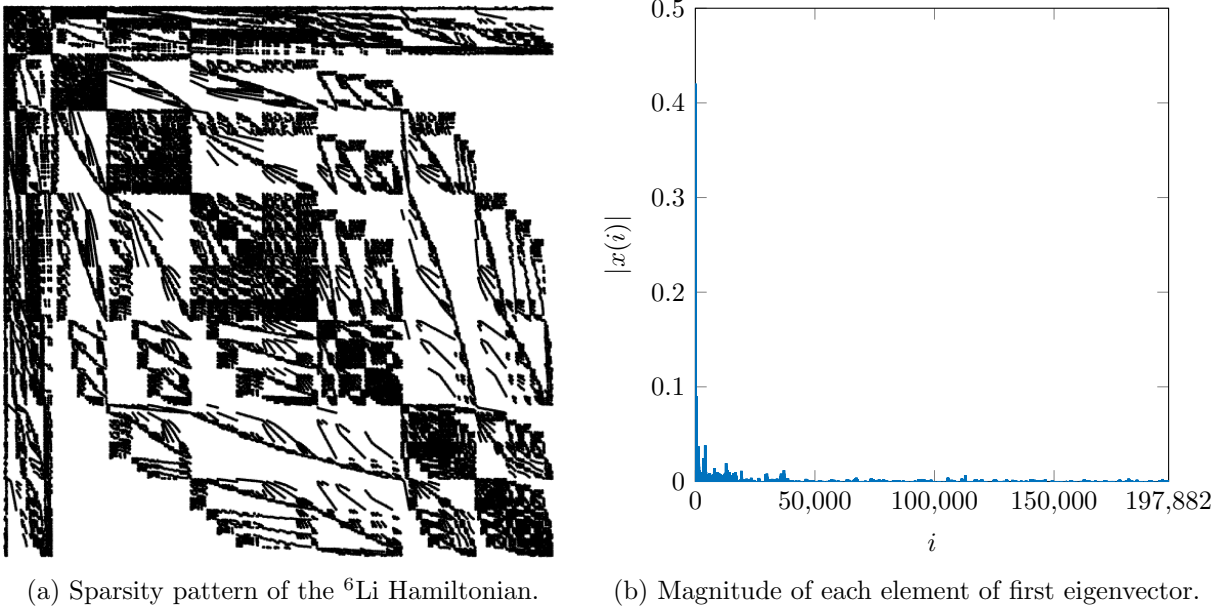


Figure 1: The sparsity structures of the ${}^6\text{Li}$ Hamiltonian in the $N_{\max} = 6$ configuration space and its first eigenvector.

starting point of both the hierarchical and greedy algorithm and compute its smallest eigenvalue and corresponding eigenvector using the LOBPCG algorithm.

For the greedy algorithm, we then use the γ value defined in (9) to select additional rows and columns to augment the matrix H_1 . We first select all rows and columns with $|\gamma|$ greater than a threshold of $\tau = 5 \times 10^{-3}$. The total number of selected rows (and columns) is 62. Next, we compute the lowest eigenvalue and corresponding eigenvector of this 862×862 matrix H_1 using the zero padded eigenvector of the previous H_1 as the starting guess, and perform the perturbation analysis again to select additional rows and columns. Using the threshold value of $\tau = 5 \times 10^{-4}$ yields an augmented matrix H_1 of dimension 8004. Although it is possible to continue this process by using a lower threshold to select additional rows and columns to further augment H_1 , a slightly lower threshold actually results in a significant increase in the number of new rows and columns to be included in H_1 . This makes it costly to compute the desired eigenpair of H_1 even when a zero padded eigenvector of the previous H_1 is used as the starting guess. We believe this is because the eigenvector associated with the smallest eigenvalue of H is not completely localized, since more than 51% of the components of the eigenvector have magnitude less than 10^{-4} and less than 10% of them are less than 10^{-5} in magnitude. Therefore, we stop the greedy selection of additional rows and columns when the dimension of H_1 reaches 8004, and use the eigenvector associated with the smallest eigenvalue of this problem as the starting guess to compute the ground state of H , after it is padded with zeros.

Figure 2 shows the convergence history of the LOBPCG algorithm applied to H using as starting guess a random starting vector (**original**), the small eigenvector of size 800 padded by zeros (**hierarch**), and the zero padded eigenvector obtained by the greedy approach from the 8004×8004 H_1 (**greedy-pert**). We plot the relative residual norm defined as

$$\|Hx^{(k)} - \theta^{(k)}\|/|\theta^{(k)}|, \quad (16)$$

where k is the iteration number, and $(\theta^{(k)}, x^{(k)})$ are the approximate eigenvalues and corresponding eigenvectors obtained at the k th iteration. We can see from Figure 2 that the starting vector constructed from the greedy approach enables the LOBPCG algorithm to converge in less than half of the number of iterations required in either the “original” approach and “hierarchical” approach. In terms of the total wall clock time, which includes the time required

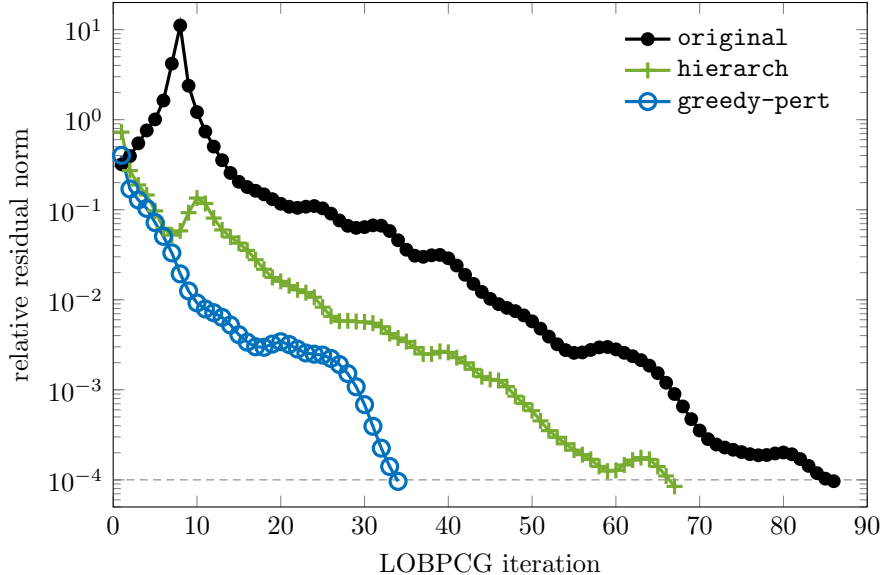


Figure 2: The convergence of the LOBPCG algorithm for computing the ground state of the ${}^6\text{Li}$ Hamiltonian at the $N_{\max} = 6$ truncation level when the initial approximation to the eigenvectors is prepared with a greedy algorithm, a hierarchical scheme and a random vector.

to compute eigenpairs of the sequence of H_1 matrices, the greedy algorithm is 2.5 times faster than the “original” approach, and 1.9 times faster than the “hierarchical” approach.

We now compare the perturbation analysis based greedy approach (**greedy-pert**) to the residual based (**greedy-res**) and full perturbation analysis based (**greedy-pert-full**) greedy approaches. Instead of using the componentwise perturbation analysis and selecting rows and columns to be included in H_1 by examining the magnitude of γ , we can examine the magnitude of the residual r' defined in (5) and choose rows and columns of H associated with elements of r' that are sufficiently large in magnitude. By setting the threshold τ to 5×10^{-1} and 10^{-1} respectively, we generate H_1 matrices of similar dimensions compared to those generated from the perturbation analysis based approach. Using the eigenvector computed from the larger H_1 matrix, we are able to obtain the desired eigenpair of H with nearly the same number of LOBPCG iterations as used by the perturbation based approach as we can see in Figure 3.

As discussed in section 3, the most costly linear perturbation analysis requires (approximately) solving a linear equation of the form given in (12) to produce the vector z that can be used for selecting additional rows and columns. In this example, we solve (12) by running 10 iterations of the MINRES algorithm. By setting the threshold τ to 5×10^{-3} and 10^{-3} respectively, we obtain H_1 matrices of similar dimensions compared to those generated from the componentwise perturbation analysis based approach. Using the eigenvector computed from the larger H_1 matrix as the starting vector, we are able to obtain the desired eigenpair of H with a slightly fewer iterations as we can see in Figure 3. However, since we need to solve (12), the overall cost of this approach is actually slightly higher.

We suggested in section 4 that it may be more beneficial to correct the eigenvector obtained from the small configuration space by performing a Newton correction which requires solving (15). Figure 4 shows that such a starting vector yields a noticeable reduction in the number of LOBPCG iterations compared to the approach that simply constructs the initial guess by padding x_1 with zeros. In this example, equation (15) is solved by running 5 MINRES iterations. If we take into account the cost required to solve (15), the overall cost of the Newton correction approach is comparable to that used by the zero padding approach.

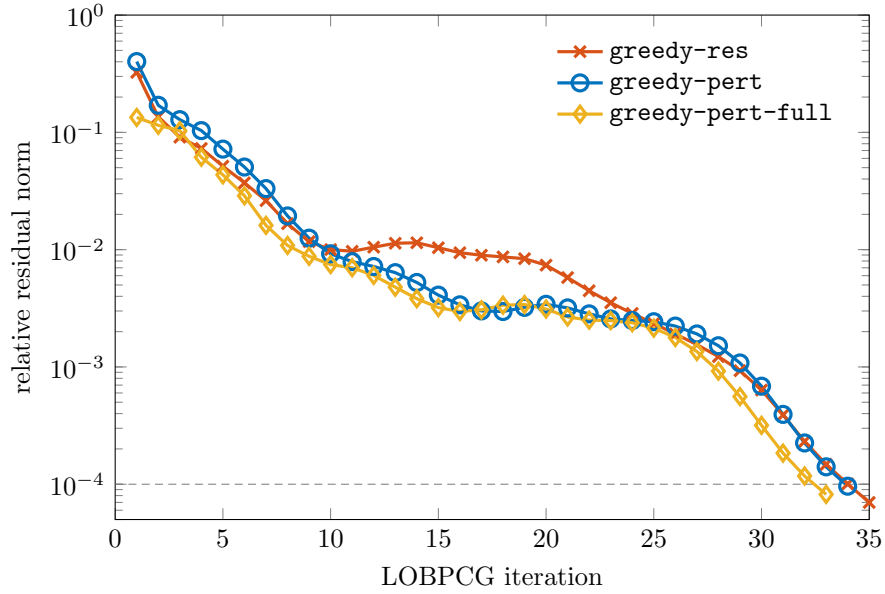


Figure 3: A comparison of the convergence of the LOBPCG algorithm when it is applied to H with starting vectors obtained from greedy algorithms that use residual and perturbation analysis respectively to select rows and columns. All 3 methods (`greedy-res`, `greedy-pert`, `greedy-pert-full`) use zero padded starting vectors.

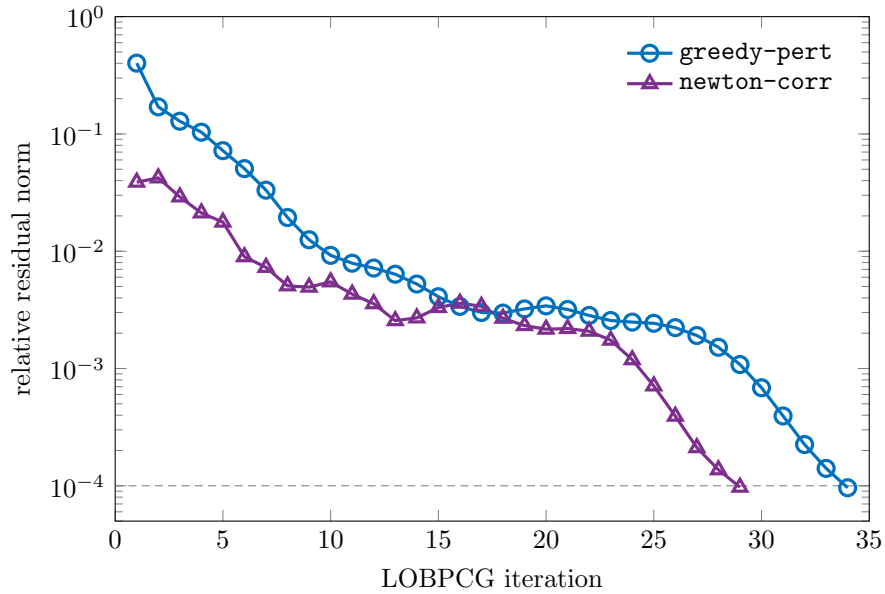


Figure 4: A comparison of the convergence of the LOBPCG algorithm when it is applied to H with starting vectors produced by the greedy algorithm padded with zeros (`greedy-pert`) or corrected by Newton's method (`newton-corr`).

5.2 Many-Body Localization

In this section, we give another example that illustrates the effectiveness of the greedy algorithm. The matrix of interest is a many-body Hamiltonian (Heisenberg spin-1/2 Hamiltonian) associated with a disordered quantum spin chain with $L = 20$ spins and nearest neighbor interactions. The Hamiltonian has the form

$$H = \sum_{i=1}^{L-1} I \otimes \cdots \otimes I \otimes H_{i,i+1} \otimes I \otimes \cdots \otimes I + \sum_{i=1}^L I \otimes \cdots \otimes I \otimes h_i S_i^z \otimes I \otimes \cdots \otimes I, \quad (17)$$

where the parameters h_i are randomly generated and represent the disorder, I is the 2-by-2 identity matrix, and

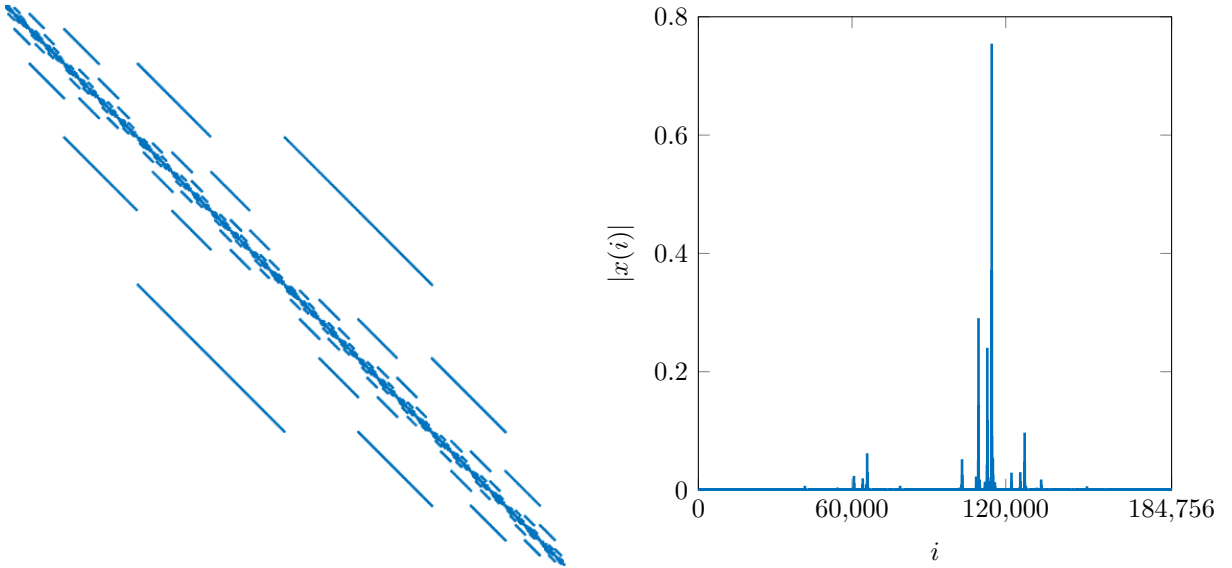
$$H_{i,i+1} = S_i^x \otimes S_{i+1}^x + S_i^y \otimes S_{i+1}^y + S_i^z \otimes S_{i+1}^z$$

is a 4-by-4 real matrix, with S^x , S^y , and S^z being spin matrices (related to the Pauli matrices by a factor of 1/2), defined as

$$S^x = \frac{1}{2} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad S^y = \frac{1}{2} \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad S^z = \frac{1}{2} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix},$$

respectively. Note that the matrices $H_{i,i+1}$ are identical for all i , their subscripts simply indicating the overlapping positions in each Kronecker product.

The matrix H (17) can be permuted into a block diagonal form. We are interested in the lowest eigenvalue of the largest diagonal block, which corresponds to half the spins being up and the other half down. The sparsity structure of this matrix, which has a dimension of 184,756, is shown in Figure 5(a). When the disorder h is sufficiently large, the eigenvectors of H exhibit a localized feature [13, 4]. Figure 5(b) shows the eigenvector associated with the lowest eigenvalue.



(a) Sparsity pattern of the many-body Hamiltonian. (b) Magnitude of each element of first eigenvector.

Figure 5: The sparsity structures of the many-body Hamiltonian associated with a Heisenberg spin chain of length 20, and its eigenvector associated with the lowest eigenvalue.

When applying the greedy algorithm (`greedy-pert`) to H , we first randomly pick 200 rows and columns of the H matrix and compute the lowest eigenvalue, i.e., the ground state, and the corresponding eigenvector of H_1 using the `eigs` function in MATLAB, which implements the implicitly restarted Lanczos method [11]. We then use the first order componentwise perturbation method to seek additional rows and columns of H to add to the submatrix H_1 to be

threshold (τ)	$\ r\ / \theta $	$\dim(H_1)$	wall clock time (sec)
10^{-3}	4.7×10^{-3}	986	2.7×10^{-3}
10^{-4}	1.0×10^{-3}	2,546	2.9×10^{-3}
10^{-5}	1.7×10^{-4}	4,316	6.0×10^{-3}
10^{-6}	2.3×10^{-5}	7,558	8.5×10^{-3}
10^{-7}	3.0×10^{-6}	12,451	1.1×10^{-2}
10^{-8}	4.2×10^{-7}	18,442	1.7×10^{-2}

Table 1: The relative residual norms of the approximate eigenpairs obtained from H_1 matrices associated with different selection thresholds τ , the corresponding dimension of H_1 , and the wall clock time required to compute these approximations for the Heisenberg spin-1/2 Hamiltonian with $L = 20$ spins.

diagonalized. To be specific, we choose rows (and columns) whose corresponding γ value defined in (9) is less than a threshold of $\tau = 10^{-3}$ and add these to the submatrix H_1 . We compute the lowest eigenvalue and its corresponding eigenvector of the augmented matrix again.

Then, rather than simply using the result to provide a starting guess vector for a diagonalization of the full matrix H , we choose to repeatedly iterate the perturbation based greedy selection process. If no additional rows (and columns) can be selected with the current threshold, we lower the threshold by a factor 10. We terminate the computation when the relative residual norm is below 10^{-7} . Because this accuracy is already sufficient, we do not need to follow up with a calculation on the full Hamiltonian, that is, using a zero padded starting guess, as we have previously described doing at the end of the greedy selection procedure.

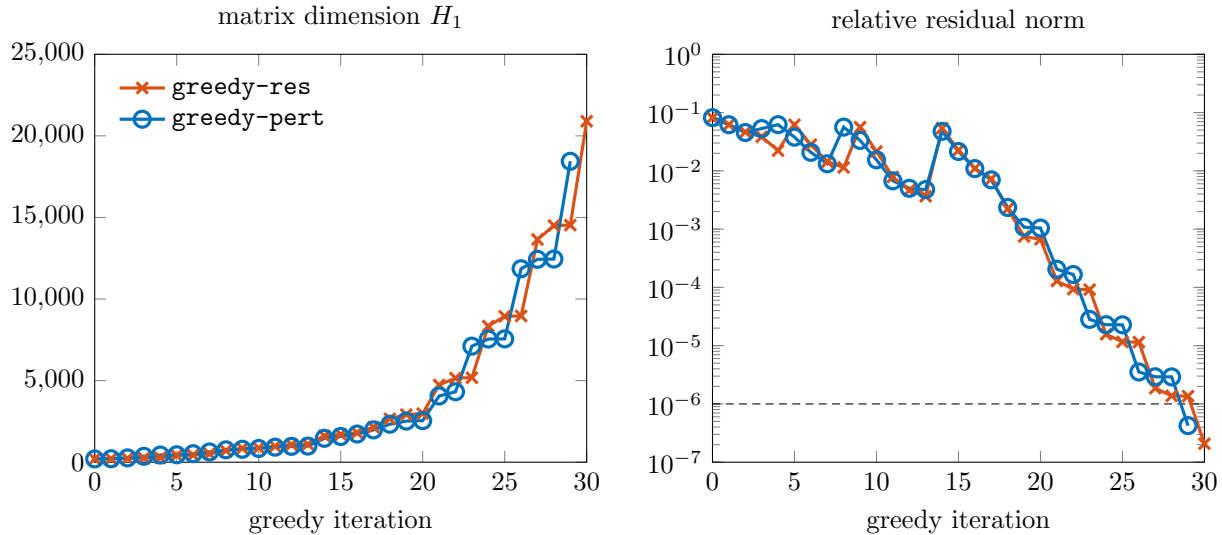
Table 1 shows the relative residual norm of the approximate eigenpair, $\|r\|/|\theta|$, where r is defined by (5), for each of the thresholds τ used in the greedy procedure. For each τ value, we also show the dimension of the augmented H_1 right before the threshold is lowered, and the wall clock time used to compute the desired eigenpairs for successively augmented H_1 matrices generated for that particular τ threshold.

When the greedy algorithm terminates, the dimension of H_1 becomes 18,442, which is less than 10% of the dimension of H , which is 184,756 for $L = 20$ spins. To illustrate the overall efficiency of this algorithm, we use the `eigs` function to compute the lowest eigenvalue and the corresponding eigenvector of the full H directly, using a random vector as the starting guess. The total wall clock time used in this full calculation is more than seven times of that used by the greedy algorithm.

We then compare the perturbation analysis based greedy approach (`greedy-pert`) to the residual based (`greedy-res`) approach. Although it may be seen, in Figure 6, that the relative residual norm of the approximate eigenpairs and the dimension of the submatrix H_1 evolve similarly with the number of greedy iterations in the two approaches, the small differences in the dimension of H_1 yield slightly different time to solution.

6 Conclusions

We presented a greedy algorithm for computing the algebraically smallest eigenpairs of a symmetric matrix H that has localization properties. The key feature of the algorithm is to select rows and columns of H to be included in a submatrix H_1 that can be easily diagonalized. The eigenvectors thus obtained for this submatrix are then used to generate more efficient starting guesses for iterative diagonalization of the full matrix H . We discussed a number of greedy strategies and criteria for such a selection, and presented numerical examples using Hamiltonian matrices arising in two types of quantum many-body eigenvalue problems. For both problems, we found that the residual based selection approach is almost as good as the strategy based on perturbation analysis. Both approaches require computing Bx_1 , where the submatrix B is defined in (3). For large problems, accessing the entirety of B may be prohibitively expensive,



(a) The change in matrix dimension of H_1 .

(b) The change in relative residual norm.

Figure 6: A comparison of two versions of the greedy algorithm that use different metric to select rows and columns.

even though this submatrix is typically sparse. Algorithms based on choosing and evaluating selected rows of B may be more efficient.

Acknowledgments

This work was supported in part by the U.S. Department of Energy, Office of Science, Office of Workforce Development for Teachers and Scientists (WDTS) under the Science Undergraduate Laboratory Internship (SULI) program, the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Scientific Discovery through Advanced Computing (SciDAC) program, and the U.S. Department of Energy, Office of Science, Office of Nuclear Physics, under Award Number DE-FG02-95ER-40934.

References

- [1] Douglas N. Arnold, Guy David, Marcel Filoche, David Jerison, and Svitlana Mayboroda. Computing spectra without solving eigenvalue problems. *SIAM Journal on Scientific Computing*, 41(1):B69–B92, 2019.
- [2] Douglas N. Arnold, Guy David, Marcel Filoche, David Jerison, and Svitlana Mayboroda. Localization of eigenfunctions via an effective potential. *Communications in Partial Differential Equations*, 2019. To appear. arXiv preprint 1712.02419.
- [3] Bruce R. Barrett, Petr Navrátil, and James P. Vary. *Ab initio* no core shell model. *Prog. Part. Nucl. Phys.*, 69:131, 2013.
- [4] R. Van Beeumen, G. Meyer, N. Yao, and C. Yang. A scalable matrix-free iterative eigensolver for studying many-body localization. In *Proc. HPC Asia*, to appear, 2020.
- [5] D. Cleland, G. H. Booth, and A. Alavi. Communications: Survival of the fittest: Accelerating convergence in full configuration-interaction quantum monte carlo. *J. Chem. Phys.*, 132:041103, 2010.

- [6] Robert J. Harrison. Approximating full configuration interaction with selected configuration interaction and perturbation theory. *The Journal of Chemical Physics*, 94(7):5021–5031, 1991.
- [7] A. A. Holmes, N. M. Tubman, and C. J. Umrigar. Heat-bath configuration interaction: An efficient selected configuration interaction algorithm inspired by heat-bath sampling. *J. Chem. Theory Comput.*, 12:3674–3680, 2016.
- [8] A. V. Knyazev. Toward the optimal preconditioned eigensolver: Locally optimal block preconditioned conjugate gradient method. *SIAM Journal on Scientific Computing*, 23(2):517–541, 2001.
- [9] A. Lagendijk, B. van Tiggelen, and D. S. Wiersma. Fifty years of anderson localization. *Physics Today*, 62(8):24, 2009.
- [10] C. Lanczos. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *J. Res. Nat. Bur. Stand.*, 45:255–282, 1950.
- [11] R. B. Lehoucq, D. C. Sorensen, and C. Yang. *ARPACK Users’ Guide*. Society for Industrial and Applied Mathematics, 1998.
- [12] Y. Li, J. Lu, and Z. Wang. Coordinatewise descent methods for leading eigenvalue problem. *SIAM J. Sci. Comp.*, 41:A2681–A2716, 2019.
- [13] David J. Luitz, Nicolas Laflorencie, and Fabien Alet. Many-body localization edge in the random-field heisenberg chain. *Phys. Rev. B*, 91:081103, Feb 2015.
- [14] R. Nandkishore and D. A. Huse. Many-Body Localization and Thermalization in Quantum Statistical Mechanics. *Annual Review of Condensed Matter Physics*, 6(1):15, 2015.
- [15] C. C. Paige and M. A. Saunders. Solution of sparse indefinite systems of linear equations. *SIAM J. Numerical Analysis*, 12:617–629, 1975.
- [16] Robert Roth. Importance truncation for large-scale configuration interaction approaches. *Physical Review C - Nuclear Physics*, 79(6):1–18, 2009.
- [17] J. B. Schriber and F. A. Evangelista. Adaptive configuration interaction for computing challenging electronic excited states with tunable accuracy. *J. Chem. Theory Comput.*, 13:5354–5366, 2017.
- [18] M. Shao, H. M. Aktulga, C. Yang, E. G. Ng, P. Maris, and J. P. Vary. Accelerating nuclear configuration interaction calculations through a preconditioned block iterative eigensolver. *Comput. Phys. Commun.*, 222:1–13, 2018.
- [19] S. Sharma, A. A. Holmes, G. Jeanmairet, A. Alavi, and C. J. Umrigar. Semistochastic heat-bath configuration interaction method: Selected configuration interaction with semistochastic perturbation theory. *J. Chem. Theory Comput.*, 13:1595–1604, 2017.
- [20] C. D. Sherrill and H. F. Schaefer. The configuration interaction method: Advances in highly correlated approaches. In P.-O. Lowdin, editor, *Advances in Quantum Chemistry*, volume 34, pages 143–269, New York, 1999. Academic Press.
- [21] J. Suhonen. *From Nucleons to Nucleus*. Springer-Verlag, Berlin, 2007.
- [22] Norm M. Tubman, Joonho Lee, Tyler Y. Takeshita, Martin Head-Gordon, and K. Birgitta Whaley. A deterministic alternative to the full configuration interaction quantum Monte Carlo method. *Journal of Chemical Physics*, 145(4), 2016.
- [23] Z. Wang, Y. Li, and J. Lu. Coordinate descent full configuration interaction. *J. Chem. Theory Comput.*, 15:3558–3569, 2019.
- [24] H. Zou and T. Hastie. Sparse principal component analysis. *Journal of Computational and Graphical Statistics*, 15:262–286, 2006.