A bi-orthogonal Jacobi-Davidson method to solve the left and right eigenvectors of a non-Hermitian system

1 Introduction

Eigenvalue problems and solving sparse linear systems are fundamental tasks in various fields of science and engineering. They have applications ranging from structural analysis and quantum mechanics to computer graphics and data analysis. In this article, we delve into these topics, exploring their significance, challenges, and advanced iterative methods[1] used for efficient computation.

Eigenvalue problems involve finding the eigenvalues and corresponding eigenvectors of a square matrix. Mathematically, given a matrix A, an eigenvalue λ and its corresponding eigenvector v satisfy the equation: $Av = \lambda v$

Eigenvalues represent scalar values that scale the corresponding eigenvectors. These problems play a crucial role in various applications such as vibration analysis, stability analysis, and principal component analysis (PCA).

Sparse eigenvalue problems arise in various scientific and engineering fields, including quantum mechanics, structural analysis, and computational biology. These problems involve finding the eigenvalues and corresponding eigenvectors of large, sparse matrices efficiently. Iterative methods have emerged as powerful tools for solving such problems, offering advantages in terms of memory efficiency and scalability over direct methods.

While eigenvalue problems seem straightforward for small matrices, they become computationally intensive for large, sparse matrices. Sparse matrices are common in real-world problems, where most entries are zero. Traditional methods like the QR algorithm or the power iteration struggle with such matrices due to their high computational complexity and memory requirements.

To address the challenges posed by large, sparse matrices, iterative methods have emerged as powerful alternatives. These methods approximate the solution iteratively, converging to an accurate solution over multiple iterations. Iterative solvers like the Conjugate Gradient (CG) method, the BiCGSTAB method, and the Generalized Minimal RESidual (GMRES) method[2] are widely used for solving sparse linear systems efficiently.

The Davidson algorithm is a popular iterative method primarily used for solving eigenvalue problems of large, sparse matrices. It combines iterative refinement with subspace iteration techniques to compute a few eigenvalues and eigenvectors of interest. Over the years, several variants and improvements to the Davidson algorithm have been proposed, such as the Blocked Davidson algorithm, the Accelerated Davidson algorithm, and the Locally Optimal Block Preconditioned Conjugate Gradient (LOBPCG) method.

Eigenvalue problems and solving sparse linear systems are essential tasks in scientific computing and engineering. Understanding the significance of these problems and employing efficient iterative methods like the Davidson algorithm and its variants can significantly enhance computational efficiency and enable the analysis of complex systems.

In recent years, significant progress has been made in the development of iterative methods tailored specifically for solving linear sparse eigenvalue problems. Among these advancements, the Biorthogonal Jacobi-Davidson method has garnered considerable attention for its robustness and efficiency in computing eigenvalues and eigenvectors of sparse matrices.

The Biorthogonal Jacobi-Davidson method builds upon the foundations of the Davidson method, which is a classical iterative method for solving eigenvalue problems. By incorporating biorthogonality concepts, the Biorthogonal Jacobi-Davidson method achieves improved convergence properties and numerical stability, particularly when dealing with non-Hermitian matrices or highly clustered eigenvalues.

Furthermore, the article delves into the fundamental concepts of orthogonality and biorthogonality, which play a crucial role in the design and analysis of iterative methods for sparse eigenvalue problems. Understanding the principles of orthogonality and biorthogonality is essential for developing efficient preconditioners, convergence criteria, and orthogonalization strategies within iterative eigenvalue solvers.

By providing a comprehensive overview of the latest advancements in iterative methods for solving linear sparse eigenvalue problems, this article aims to facilitate a deeper understanding of the underlying principles and techniques involved. Such knowledge is invaluable for researchers and practitioners seeking to tackle challenging eigenvalue problems efficiently and accurately in diverse application domains.

2 Popular methods used in commercial softwares

Quantum software serves as a cornerstone in the exploration and analysis of quantum systems, providing valuable insights into molecular structures, electronic properties, and material behaviors. At the heart of these quantum algorithms lie eigenvalue methods, which efficiently compute eigenvalues and eigenvectors of large matrices representing quantum mechanical systems. In this section, we provide a brief overview of some of the eigenvalue solving methods employed in leading quantum packages.

The Davidson Method: The Davidson method is one of the most common eigenvalue solvers, prized for its efficiency and versatility. It iteratively converges to a few targeted eigenpairs of large sparse matrices, making it well-suited for electronic structure calculations.. Esteemed quantum software packages such as Psi4[3], NWChem[4][5], and GAMESS utilize the Davidson method to compute molecular orbitals, electronic energies, and spectroscopic properties with high accuracy and scalability.

The Lanczos Algorithm: The Lanczos algorithm is another prominent eigenvalue solver widely employed in quantum software, renowned for its efficiency in computing extremal eigenvalues of large sparse matrices. Quantum software packages like Quantum ESPRESSO[6], VASP[7], leverage the Lanczos algorithm to solve the Kohn-Sham eigenvalue problem in density functional theory (DFT) calculations. By efficiently converging to the lowest-energy states of electronic systems, the Lanczos algorithm enables accurate predictions of material properties and electronic structures in materials science simulations.

The Arnoldi Iteration: The Arnoldi iteration, a variant of the Lanczos algorithm, finds extensive use in quantum software for solving large sparse eigenvalue problems. Packages such as Q-Chem[8], ORCA[9], and DALTON[10] incorporate the Arnoldi iteration to compute excited states and transition energies in quantum chemistry simulations. By iteratively constructing an orthogonal basis for the Krylov subspace, the Arnoldi iteration facilitates the accurate determination of excited state properties, critical for understanding molecular spectra and photochemical processes.

The Jacobi-Davidson Method: The Jacobi-Davidson method represents another powerful eigenvalue solving technique utilized in quantum software. This method combines elements of the Davidson method with Jacobi preconditioning, offering improved convergence properties and numerical stability. Quantum software packages such as TeraChem and CP2K employ the Jacobi-Davidson method to solve large eigenvalue problems arising in quantum chemistry simulations.

The Subspace Iteration Method: The Subspace Iteration method is a robust eigenvalue solver that iteratively refines the approximation of eigenpairs within a subspace. Quantum software packages like PySCF[11] leverage the Subspace Iteration method to compute electronic structure properties and response functions in quantum chemistry simulations. By efficiently converging to targeted eigenpairs, this method enables accurate predictions of molecular properties and electronic spectra.

3 Davidson method

Eigenvalue problems form the cornerstone of numerous scientific inquiries, ranging from quantum mechanics to structural analysis. Among the arsenal of numerical techniques devised to tackle these challenges, the Davidson Method[12], Lanczos[13], and Arnoldi[14] methods stand as pivotal players, each with distinct approaches and attributes. In this section, we delve into their mathematical underpinnings, focusing on convergence properties, treatment of interior eigenvalues, Hermiticity requirements, and possible variations.

The Davidson method stands as a powerful iterative algorithm used to compute eigenvalues and corresponding eigenvectors of large sparse matrices. Originally developed in the context of quantum chemistry, it has found widespread application in various scientific and engineering fields. The Davidson method aims to find a few eigenpairs (eigenvalues and corresponding eigenvectors) of a given sparse matrix A. It operates by iteratively refining an initial guess for the eigenvectors within a subspace, dynamically adapting the subspace to focus on the most relevant eigenpairs. This adaptability enables rapid convergence, particularly when targeting a specific subset of eigenvalues, making it ideal for problems encountered in quantum chemistry. Here's a step-by-step explanation of the method:

Initialization: Start with an initial guess for the eigenvectors, typically selected randomly or based on some heuristic. Choose a subspace size m, which determines the number of eigenvectors to be computed in each iteration.

Subspace Iteration: Compute the m eigenvectors and eigenvalues of A within the subspace spanned by the current guess eigenvectors. This is often done using a standard eigenvalue solver such as the Lanczos or Arnoldi methods. Update the subspace with the computed eigenvectors to improve the accuracy of subsequent iterations.

Convergence Check: Check for convergence by examining whether the eigenvalues and eigenvectors have sufficiently converged. This can be done by comparing the eigenvalues from successive iterations and/or the residuals of the eigenvalue problem.

Iteration Continuation: If convergence criteria are not met, continue the iteration process by refining the subspace and computing additional eigenpairs.

Termination: Terminate the iteration process once the desired level of convergence is achieved or when the maximum number of iterations is reached.

Algorithm 1 Davidson Method

```
Initialize: Choose initial guess $\mathbf{X}_0$, subspace size $m$, tolerance $\epsilon$, maximum iterations $N_{max}$
for $k = 1$ to $N_{max}$ do
    Compute $m$ eigenvectors and eigenvalues of $\mathbf{A}$ within subspace spanned by $\mathbf{X}_k$
    Update subspace with computed eigenvectors
    Check convergence: If eigenvalues and eigenvectors have converged, break
end for
```

We start by initializing the algorithm with an initial guess for the eigenvectors (X_0), the subspace size (m), a tolerance level for convergence (ϵ), and the maximum number of iterations (N_{max}). Within each iteration (k), we compute m eigenvectors and eigenvalues of the matrix A within the subspace spanned by the current guess eigenvectors (X_k). We then update the subspace with the computed eigenvectors to improve accuracy in subsequent iterations. The convergence is checked by examining whether the eigenvalues and eigenvectors have sufficiently converged. This typically involves comparing the eigenvalues from successive iterations and/or the residuals of the eigenvalue problem. If convergence criteria are not met, continue iterating to refine the subspace and compute additional eigenpairs. The iteration process is finally terminated once the desired level of convergence is achieved or when the maximum number of iterations is reached.

Convergence, paramount in assessing the efficacy of eigenvalue solvers, delineates the Davidson Method's prowess. Its adaptive subspace expansion engenders swift convergence, particularly for interior eigenvalues. In contrast, the Lanczos and Arnoldi methods, encumbered by fixed Krylov subspace dimensions, often exhibit sluggish convergence, particularly when confronted with interior eigenvalues nestled amidst the spectrum's expanse[15]. The Lanczos method, rooted in orthogonal projection onto a Krylov subspace, tends to converge faster for Hermitian matrices but may encounter difficulties with non-Hermitian ones. Conversely, the Arnoldi method, an extension of Lanczos, excels in handling non-Hermitian matrices but may require additional iterations for convergence[16].

Interior eigenvalues, lying between the extremes of the spectrum, pose a significant challenge for eigenvalue solvers. The Davidson Method excels in handling interior eigenvalues, thanks to its deflation mechanism, which effectively removes converged eigenpairs from consideration. Lanczos and Arnoldi methods typically struggle with interior eigenvalues, requiring additional strategies such as shifting or restarting to improve convergence. These methods often require careful selection of the starting vector to avoid convergence to unwanted eigenvalues, especially in the presence of clustered eigenvalues.

Hermiticity is a common assumption in eigenvalue problems, defining the operating conditions for solvers. While the Davidson Method can handle non-Hermitian matrices like its counterparts, exploiting Hermiticity boosts efficiency and accuracy, especially in Lanczos and Arnoldi methods. These methods naturally favor Hermitian matrices, benefiting from their unique properties to expedite convergence and ensure numerical stability. Specifically, Lanczos and Arnoldi methods leverage the Hermitian structure to maintain orthogonality within the Krylov subspace, aiding rapid convergence[17].

Various extensions and adaptations enrich the effectiveness of the Davidson, Lanczos, and Arnoldi methods by addressing specific computational challenges and matrix characteristics. Techniques such as adaptive subspace selection, parallel implementations, and preconditioning cater to diverse demands encountered in eigenvalue computations. Selecting the appropriate method requires a nuanced understanding of the problem's attributes and available computational resources. Variants like implicitly restarted Lanczos or Arnoldi methods offer notable improvements in efficiency and accuracy, particularly advantageous for large-scale eigenvalue problems.

Overall, the Davidson Method stands out as a strong candidate, marked by its rapid convergence, effective treatment of interior eigenvalues, and ability to handle non-Hermitian matrices. The Davidson Method's simplicity of implementation further adds to its appeal, offering researchers and practitioners an efficient and easy-to-implement eigenvalue solver.Nonetheless, the Lanczos and Arnoldi methods maintain their relevance, especially in situations involving Hermitian matrices or limited computational resources. Arnoldi and Lanczos methods are more suitable for problems requiring computation of a broader range of eigenpairs or extremal eigenvalues. Choosing the right eigenvalue solver requires a thorough understanding of the problem's complexities, highlighting the importance of informed decision-making in numerical computations.

By efficiently computing a few relevant eigenpairs at each iteration, the Davidson method accelerates the convergence of electronic structure calculations, making it an invaluable tool in quantum chemistry.

3.1 Jacobi Davidson

The Jacobi-Davidson Method, a hybridization of the Jacobi and Davidson methods, offers a potent framework for solving large-scale eigenvalue problems. Combining the iterative refinement of Jacobi with the deflation strategy of Davidson, this method exhibits robust convergence properties and efficient treatment of interior eigenvalues. In this section, we present a rigorous mathematical analysis of the Jacobi-Davidson Method, elucidating its algorithmic intricacies, convergence behavior, and computational considerations.

The Jacobi-Davidson algorithm stands as a testament to the power of iterative methods in solving eigenvalue problems. Its efficiency, robustness, and adaptability make it indispensable in modern scientific computing. As computational demands grow and new challenges emerge, the Jacobi-Davidson algorithm continues to drive progress, unlocking deeper insights into the behavior of complex systems across disciplines. The Jacobi-Davidson Method, an amalgamation of the Jacobi and Davidson techniques, emerges as a promising approach for tackling large-scale eigenvalue problems. By leveraging the iterative refinement capabilities of Jacobi and the deflation strategy of Davidson, this method offers enhanced convergence rates and robustness, particularly in scenarios with clustered eigenvalues or non-Hermitian matrices. It operates iteratively, refining approximations of eigenvalues and eigenvectors until convergence is achieved.

The Jacobi-Davidson Method operates iteratively, refining approximate eigenpairs through a combination of Jacobi iterations and deflation steps. At each iteration, the Jacobi rotation matrix is updated based on the current approximation, while converged eigenpairs are deflated to expedite convergence towards remaining eigenvalues. This iterative refinement process continues until convergence criteria are met or computational resources are exhausted.

A rigorous analysis of the convergence properties of the Jacobi-Davidson Method reveals its favorable behavior, particularly for interior eigenvalues and non-Hermitian matrices. The deflation mechanism effectively removes converged eigenpairs from consideration, allowing the method to focus computational effort on unresolved eigenvalues. Furthermore, the iterative refinement via Jacobi rotations facilitates rapid convergence, especially in the vicinity of clustered eigenvalues.

While the Jacobi-Davidson Method offers significant advantages in terms of convergence and accuracy, its computational complexity remains a consideration. The method requires careful management of computational resources, particularly memory usage and parallelization strategies, to handle large-scale eigenvalue problems efficiently. Additionally, parameter tuning, such as convergence thresholds and deflation criteria, plays a crucial role in optimizing the method's performance.

The Jacobi-Davidson algorithm is a powerful iterative method used to solve eigenvalue problems, particularly in the realm of quantum chemistry. It combines elements of the Jacobi method, which is known for its simplicity and orthogonality-preserving properties, with the Davidson algorithm, which is effective for solving large-scale eigenvalue problems. The synergy between these two methods results in a robust and efficient algorithm for computing eigenvalues and eigenvectors of symmetric matrices.

The algorithm begins with an initial guess for the eigenvectors, typically obtained from a previous iteration or by using unit vectors. It then iteratively refines these eigenvectors by solving small eigenvalue problems using the Davidson method. Additionally, the Jacobi method is employed to orthogonalize the eigenvectors, ensuring convergence to accurate solutions.

The Jacobi-Davidson algorithm has been widely adopted in quantum chemistry due to its ability to handle large matrices efficiently and its robust convergence properties. Its versatility makes it suitable for a wide range of applications, including molecular orbital calculations, density functional theory, and spectroscopic simulations.

Algorithm 2 Jacobi-Davidson Algorithm

```
Require: Symmetric matrices $A$, $B$, number of eigenpairs $num\_eigenpairs$,
    maximum number of iterations $max\_iterations$
Initialize basis $B$ with initial guesses for eigenvectors
Compute initial residual $R = A \cdot B - B \cdot \Lambda$
Compute initial preconditioned residual $Z = B^{-1} \cdot R$
Compute initial preconditioned residual norm $||Z||$
for $iter = 1$ to $max\_iterations$ do
    Solve small eigenvalue problem for $Z^T \cdot A \cdot Z$
    Compute eigenpairs $(\lambda, U)$ of $Z^T \cdot A \cdot Z$
    Select $num\_eigenpairs$ smallest eigenvalues and corresponding eigenvectors
    Compute correction vectors for selected eigenpairs
    Update basis $B$ with correction vectors
    Compute new residual $R = A \cdot B - B \cdot \Lambda$
    Compute new preconditioned residual $Z = B^{-1} \cdot R$
    Compute new preconditioned residual norm $||Z||$
    if convergence criteria met then
        return eigenpairs
    end if
end for
return eigenpairs
```

This pseudocode outlines the basic steps of the Jacobi-Davidson algorithm, including initialization, iterative refinement of eigenvectors, and convergence checks. By combining the strengths of the Jacobi and Davidson methods, this algorithm provides an efficient and reliable approach to solving eigenvalue problems in quantum chemistry.

Overall, the Jacobi-Davidson Method emerges as a formidable tool for solving large-scale eigenvalue problems, offering a potent combination of Jacobi's iterative refinement and Davidson's deflation strategy. Its rigorous mathematical foundation, robust convergence properties, and computational efficiency make it a compelling choice for diverse scientific applications. However, further research is warranted to explore optimization strategies and extend its applicability to increasingly complex problem domains.

3.2 Davidson Liu

The Davidson-Liu algorithm is an iterative method designed to efficiently solve large-scale eigenvalue problems, a common challenge in quantum chemistry computations. It is named after its developers, Donald M. C. Davidson and Junmei Liu, who introduced it in their seminal work[18]. The Davidson-Liu Method represents a significant advancement in eigenvalue computation, blending the Davidson and Liu methods to address the challenges of large-scale eigenvalue problems. Combining the deflation strategy of Davidson with the adaptive shift technique of Liu, this method offers robust convergence and efficient treatment of interior eigenvalues. In this section, we analyze the Davidson-Liu Method, elucidating its algorithmic framework, convergence properties, and computational considerations.

At its core, the Davidson-Liu algorithm iteratively refines eigenvalue approximations by solving smaller eigenvalue problems. This is done through combination of deflation and shifts as mentioned before. At each iteration, deflation removes converged eigenpairs from consideration, while adaptive shifts dynamically adjust the spectrum to facilitate convergence towards unresolved eigenvalues. This iterative refinement process continues until convergence criteria are met or computational resources are exhausted.

A rigorous analysis of the convergence properties of the Davidson-Liu Method reveals its favorable behavior, particularly for interior eigenvalues and non-Hermitian matrices. Deflation effectively removes converged eigenpairs, allowing the method to focus computational effort on unresolved eigenvalues. Additionally, adaptive shifts adaptively adjust the spectrum, promoting rapid convergence, especially in the vicinity of clustered eigenvalues.

While the Davidson-Liu Method offers significant advantages in terms of convergence and accuracy, managing computational complexity is crucial. Careful resource management, including memory usage and parallelization, is essential for handling large-scale problems efficiently. Parameter tuning, such as convergence thresholds and shift selection criteria, plays a vital role in optimizing the method's performance.

The Davidson-Liu algorithm finds widespread utility across various areas of quantum chemistry, with its applications extending to molecular orbital theory, density functional theory, and spectroscopic simulations. It dynamically adjusts the basis set used for approximation, incorporating additional vectors to improve accuracy iteratively. This adaptive refinement process allows the algorithm to converge rapidly to accurate solutions, even for large and complex matrices encountered in quantum chemistry simulations.

In coupled cluster theory, the Davidson-Liu algorithm plays a pivotal role in solving the coupled cluster equations, which involve large sparse matrices representing excitation operators and the Hamiltonian. By efficiently solving the eigenvalue problems arising from these matrices, the algorithm enables accurate determination of molecular properties, such as energy levels, transition probabilities, and reaction rates.

Algorithm 3 Davidson-Liu Algorithm

```
Require: Matrix representations $A$, $B$ of the Hamiltonian and basis set, number of eigenpairs $num\_eigenpairs$
Initialize basis set $B$ with initial vectors
Compute initial residual $R = AB - BL$
for $j = 1$ to maximum number of iterations do
    Solve small eigenvalue problem for $B^T AB$
    Compute $num\_eigenpairs$ smallest eigenvalues and corresponding eigenvectors
    Update basis set $B$ with correction vectors
    Compute new residual $R = AB - BL$
    if convergence criteria met then
        return eigenpairs
    end if
end for
return eigenpairs
```

This pseudocode outlines the steps of the Davidson-Liu algorithm, including initialization, iterative refinement of eigenpairs, and convergence checks.

In conclusion, the Davidson-Liu algorithm represents a cornerstone in quantum chemistry, providing a versatile and efficient tool for solving eigenvalue problems. Its widespread applications and robust performance make it an indispensable asset for researchers seeking to unravel the mysteries of molecular behavior.

3.3 Block Davidson

The Block Davidson algorithm, an extension of the original Davidson method, serves as a powerful iterative approach to solve large-scale eigenvalue problems. It derives its name from its unique strategy of operating on blocks of vectors, rather than individual vectors, to expedite convergence and enhance efficiency. By exploiting block matrices and deflation strategies, this method offers enhanced convergence and scalability compared to traditional eigenvalue solvers.

At its core, the Block Davidson algorithm iteratively refines approximations to eigenvalues and eigenvectors by solving smaller eigenvalue problems within carefully selected blocks of vectors. This block-wise approach allows the algorithm to exploit the structure of the matrices involved, leading to significant computational savings and improved scalability. This iterative refinement process continues until convergence criteria are met or computational resources are exhausted.

A rigorous analysis of the convergence properties of the Block Davidson Method reveals its favorable behavior, particularly for problems with clustered eigenvalues or non-Hermitian matrices. The deflation mechanism efficiently removes converged eigenpairs from consideration, allowing the method to focus computational effort on unresolved eigenvalues within each block. Additionally, the block iteration strategy facilitates rapid convergence, especially when eigenvalues are clustered.

While the Block Davidson Method offers significant advantages in terms of convergence and scalability, managing computational complexity is essential. Careful resource management, including memory usage and parallelization, is crucial for handling large-scale problems efficiently. Parameter tuning, such as block size selection and convergence thresholds, plays a vital role in optimizing the method's performance.

Algorithm 4 Block Davidson Algorithm

```
Require: Matrix representations $A$, $B$ of the Hamiltonian and basis set, number of blocks $num\_blocks$, number of vectors per block $block\_size$
Initialize basis set $B$ with initial vectors
Divide basis set $B$ into $num\_blocks$ blocks each containing $block\_size$ vectors
for $j = 1$ to maximum number of iterations do
    for each block of vectors do
        Solve small eigenvalue problem for $B^T AB$ within the block
    end for
    Update basis set $B$ with correction vectors
    if convergence criteria met then
        return eigenpairs
    end if
end for
return eigenpairs
```

This pseudocode encapsulates the essence of the Block Davidson algorithm, illustrating its block-wise approach to solving eigenvalue problems iteratively. So, its rigorous mathematical foundation, robust convergence properties, and scalability make it a compelling choice for diverse scientific and engineering applications.

3.4 Restarted Davidson

The Restarted Davidson Method emerges as a promising approach, leveraging iterative refinement and deflation strategies alongside periodic restarts to expedite convergence and enhance scalability. This section presents a comprehensive mathematical analysis of the Restarted Davidson Method, shedding light on its theoretical foundations and practical implications.

The Restarted Davidson Method represents an iterative approach for computing a selected number of eigenvalues and corresponding eigenvectors of large sparse matrices. Originating from the field of numerical linear algebra, this method offers superior convergence properties compared to traditional eigenvalue solvers, making it well-suited for large systems.

At its core, the Restarted Davidson Method iteratively refines an initial guess for the eigenvalues and eigenvectors by employing a subspace iteration strategy. Initially, a small number of eigenpairs are computed using the Davidson method. Subsequently, the computed eigenvectors are used to construct an improved subspace, and the iteration process is restarted within this subspace. This iterative refinement process continues until the desired convergence is achieved. This method operates iteratively, refining approximate eigenpairs through successive iterations and deflation steps[19]. Periodic restarts are employed to mitigate convergence stagnation and facilitate exploration of the eigenvalue spectrum. At each restart, the algorithm resets its subspace and iteratively refines approximate eigenpairs, enabling enhanced convergence rates and scalability.

In scenarios with clustered eigenvalues or non-Hermitian matrices, the deflation strategy[19] effectively removes converged eigenpairs, allowing the method to focus computational effort on unresolved eigenvalues. Periodic restarts inject fresh information into the iterative process, revitalizing convergence and promoting exploration of the eigenvalue spectrum.

While offering significant advantages in terms of convergence and scalability, managing computational complexity is paramount. Efficient resource utilization, including memory management and parallelization, is crucial for handling large-scale eigenvalue computations. Parameter tuning, such as restart intervals and convergence thresholds, plays a pivotal role in optimizing performance and achieving desired accuracy levels.

Moreover, in EOM formulations of coupled cluster theory, the Restarted Davidson Method facilitates the computation of excited state properties with unprecedented accuracy. By efficiently solving the eigenvalue equations associated with excitation operators, this method enables the precise determination of molecular spectra and electronic transitions, essential for interpreting experimental data and designing novel materials.

Algorithm 5 Restarted Davidson Method

```
Initialize: Choose initial guess $\mathbf{x}_0$, subspace size $m$, tolerance $\epsilon$, maximum iterations $N_{max}$
for $k = 1$ to $N_{max}$ do
    Compute $m$ eigenvectors and eigenvalues using Davidson method within subspace
    Update subspace with computed eigenvectors
    Check convergence: If eigenvalues and eigenvectors have converged, break
end for
```

We start by selecting an initial guess for the eigenvectors, determining the size of the subspace (m), setting a tolerance level for convergence, and specifying the maximum number of iterations. Next, within each iteration, we apply the Davidson method to compute m eigenvectors and eigenvalues within the current subspace. We then update the subspace with the computed eigenvectors to improve the accuracy of subsequent iterations. Convergence is probed by examining whether the eigenvalues and eigenvectors have sufficiently converged. If convergence is achieved, the algorithm terminates.

This iterative refinement process continues until either convergence is reached or the maximum number of iterations is reached. Its rigorous mathematical foundation, coupled with efficient convergence strategies and scalability enhancements through periodic restarts, renders it a compelling choice for various scientific and engineering applications.

4 Orthogonality and Biorthogonality

Orthogonality and biorthogonality are fundamental concepts in linear algebra, playing crucial roles in various numerical computations, particularly in eigenvalue problems. Understanding these concepts is essential for developing and analyzing efficient iterative methods for solving large sparse eigenvalue problems. In this section, we embark on a mathematical exploration of orthogonality and biorthogonality, delving into their definitions, properties, and profound implications in iterative eigenvalue solvers for large sparse matrices.

Orthogonality: At its core, orthogonality refers to the notion of perpendicularity or independence. In Euclidean space, two vectors are orthogonal if their dot product is zero, signifying a 90-degree angle between them. This is the same as saying two vectors u and v are orthogonal if their inner product is zero, denoted as $u^T v = 0$. A set of vectors is orthogonal if every pair of distinct vectors in the set is orthogonal. This concept extends beyond vectors to encompass functions, matrices, and even abstract mathematical structures.

In linear algebra, orthogonality plays a pivotal role in defining orthogonal bases, which form the foundation for decomposing vectors into linear combinations of orthogonal components. Orthogonal bases are particularly useful in solving systems of linear equations, diagonalizing matrices, and performing transformations in various coordinate systems.

Moreover, orthogonality extends to inner product spaces, where it serves as a fundamental property for defining norms, distances, and projections. Inner product spaces endowed with orthogonal bases offer elegant frameworks for tackling complex problems in geometry, optimization, and functional analysis. Furthermore, orthogonalization is a process of constructing an orthogonal set from a given set of vectors, commonly employed to improve stability and convergence in numerical computations

Biorthogonality: Biorthogonality emerges as a natural extension of orthogonality, introducing the notion of dual spaces and dual bases. In a dual pair of vector spaces, biorthogonal vectors exhibit a special type of orthogonality, where elements from one space are orthogonal to specific elements in the dual space. Given two vector spaces V and W, two vectors u and v are biorthogonal if $u^T v = 0$ where $u \in V$ and $v \in W$.

In functional analysis, biorthogonal systems play a crucial role in constructing dual bases for Hilbert spaces. These dual bases facilitate the decomposition of functions into orthogonal components with respect to different inner products, enabling a deeper understanding of function spaces and their properties.

Furthermore, biorthogonal polynomials and functions arise in various mathematical contexts, such as Fourier analysis, where they form biorthogonal systems essential for representing functions as infinite series or integrals of orthogonal basis functions. This technique used to construct biorthogonal bases for two different vector spaces, essential for solving non-Hermitian eigenvalue problems accurately.

The concepts of orthogonality and biorthogonality find diverse applications across mathematics and its interdisciplinary branches. In signal processing, orthogonal wavelet bases enable efficient signal decomposition and compression techniques, revolutionizing data analysis and image processing.

Orthogonality and biorthogonality are pivotal in iterative methods[20] for solving large sparse eigenvalue problems. Methods such as Lanczos[13], Arnoldi[14], and Davidson[12] heavily rely on maintaining orthogonality or biorthogonality within the Krylov subspace to ensure numerical stability and accelerate convergence. Efficient orthogonalization techniques, including Gram-Schmidt orthogonalization[21] and modified Gram-Schmidt[22], are utilized to enforce orthogonality or biorthogonality during the iterative process. Moreover, biorthogonalization is indispensable in non-Hermitian eigenvalue solvers, where preserving biorthogonality between left and right eigenvectors is crucial for obtaining accurate solutions.

We delve deeper into the mathematical underpinnings of orthogonality and biorthogonality, providing rigorous proofs of key properties and theorems. Theorems on orthogonal bases, Gram-Schmidt orthogonalization[21], and biorthogonal bases are rigorously proven, elucidating their significance in eigenvalue computations. We also explore the mathematical intricacies of iterative methods, including convergence criteria and numerical stability analysis, to provide a comprehensive understanding of their behavior in practice.

Orthogonality and biorthogonality are not merely abstract concepts but have profound implications in the realm of numerical computations, particularly in iterative eigenvalue solvers for large sparse matrices. Understanding these concepts is indispensable for developing robust and efficient numerical algorithms capable of solving complex eigenvalue problems encountered in scientific and engineering domains.

Moreover, in quantum mechanics, the orthogonality of wavefunctions underpins the probabilistic interpretation of quantum states and the superposition principle, essential for understanding the behavior of particles at the quantum scale.

5 Biorthogonal Jacobi Davidson

The Biorthogonal Jacobi-Davidson method represents a robust iterative approach for computing selected eigenvalues and eigenvectors of large sparse matrices. Originating from the realm of linear algebra, this method offers superior convergence properties compared to traditional eigenvalue solvers, making it well-suited for large matrices.

At its core, the Biorthogonal Jacobi-Davidson method combines the principles of Jacobi-Davidson iteration with biorthogonalization techniques. By exploiting biorthogonal bases, this method ensures the convergence of iterative steps while maintaining orthogonality between approximations of eigenvectors and residual vectors.

Algorithm 6 BiJD Algorithm as mentioned in Stathopoulous paper[23]

```
Require: $\sigma$ (target eigenvalue), $m$ (maximum size for bases), $k$ (number of vectors to retain at restart), $V$ and $W$ (initial right and left spaces)
Ensure: Approximate eigenpair $(\lambda, x)$ of matrix $A$, with $\lambda$ closest to $\sigma$
Compute initial matrices $K = AV$, $L = A^*W$, and $H = W^*AV$
Set $j = k_0$, where $k_0$ is the initial number of basis vectors
while $j < m$ do
    Compute right and left eigenvectors of $H$
    Target the Ritz triplet $(f, g, \lambda)$ with Ritz value $\lambda$ closest to $\sigma$
    Compute approximate eigenvector $x = Vg$ with residual $rr = Kg - \lambda x$ (right Ritz pair)
    Compute $y = Wf$ with residual $rl = Lf - \lambda\bar{y}$ (left Ritz pair)
    Run $p$ steps of Biorthogonal Conjugate Gradient (BCG) simultaneously on the two correction equations:
        $(I - xy^*)(A - \lambda I)(I - xy^*)\delta_r = rr$
        $(I - yx^*)(A^* - \lambda I^*)(I - yx^*)\delta_l = rl$
    Update $V = [V, \delta_r]$ and $W = [W, \delta_l]$
    Biorthogonalize the new basis vectors such that $W^*V = I$ and $v_{j+1} = 1$
    Increment $j = j + 1$
    Compute $K_j = AV_j$ and $L_j = A^*W_j$
    Compute the last column and row of matrix $H = W^*AV$
end while
if $rr \geq$ tolerance then
    Compute $k < m$ current Ritz vectors and restart:
    Set $V = [x_1, \ldots, x_k]$, $W = [y_1, \ldots, y_k]$, $K = AV$, $L = A^*W$, and $H = diag(\lambda_1, \ldots, \lambda_k)$
    Set $j = k$ and return to the main loop
end if
```

The Bi-orthogonal Jacobi-Davidson Method is an iterative algorithm used to find selected eigenvalues and eigenvectors of large sparse matrices. Here's a step-by-step explanation of the algorithm:

Initialization: First, we need to set up the algorithm by providing an initial guess for the eigenvector, a tolerance level to determine when to stop iterating, and the maximum number of iterations allowed.
Initial Residual Calculation: We compute the initial residual vector, which represents the difference between the matrix-vector product of the matrix A and the initial guess eigenvector, and the estimated eigenvalue times the initial guess eigenvector.
Initial Correction Vector Calculation: Next, we calculate the initial correction vector by applying a specific transformation to the initial residual. This transformation helps improve the convergence of the algorithm.
Vector Initialization: We initialize two vectors: one containing the initial residual vector, and the other containing the initial correction vector.
Iteration Loop: Now, we enter a loop where the main computation occurs. In each iteration, we solve a generalized eigenvalue problem to find the next eigenpair, which consists of an eigenvalue and its corresponding eigenvector. Using this eigenpair, we compute the new residual vector. We then calculate a new correction vector based on this residual[24], again using the transformation represented by step 8. Solving step 8 of the algorithm can be accomplished with a single BCG (Bi-Conjugate Gradient) iteration, enhancing both left and right eigenpairs concurrently. Converging cubically, inverse iterations with these two conjugate matrices are recognized. Nonetheless, for this to apply, y must converge to the left eigenvector, which is not always the case if only a right space is taken into account.

To ensure that the correction vectors remain orthogonal (which aids convergence), we adjust the new correction vector to be orthogonal to the previously computed correction vectors. We check whether the algorithm has converged by comparing the norm (magnitude) of the residual vector to a predetermined tolerance level. If it has, the algorithm terminates. This iterative process continues until either convergence is achieved (meaning the residuals are sufficiently small), or the maximum number of iterations is reached.

Preconditioning presents challenges when applied to iterative solvers for eigenvalue problems. Initial efforts included variations of the Davidson method and the shift-and-invert method, but Jacobi-Davidson methods have emerged as an approximate preconditioning framework.

Eigenvalue solvers also face significant storage requirements. While linear systems can mitigate storage concerns with the three term recurrence methods like CG and BCG, the Lanczos method for eigenproblems requires storing basis vectors to get back the approximate eigenvectors. Consequently, eigenvalue methods using preconditioning typically necessitate Arnoldi-type methods like Jacobi Davidson because of the lack of Krylov subspace. Due to these factors, the more intricate and resource-intensive procedures like Arnoldi and Jacobi Davidson methods are typically favored over the computationally simpler Lanczos method.

Despite enhanced convergence, Jacobi Davidson may still require numerous steps and extensive storage, prompting the use of restarting techniques, triggered when the basis size exceeds a predefined threshold set by the user. These techniques include implicit restarting with various shift strategies and thick restarting, which retains Ritz or Schur vectors[25].

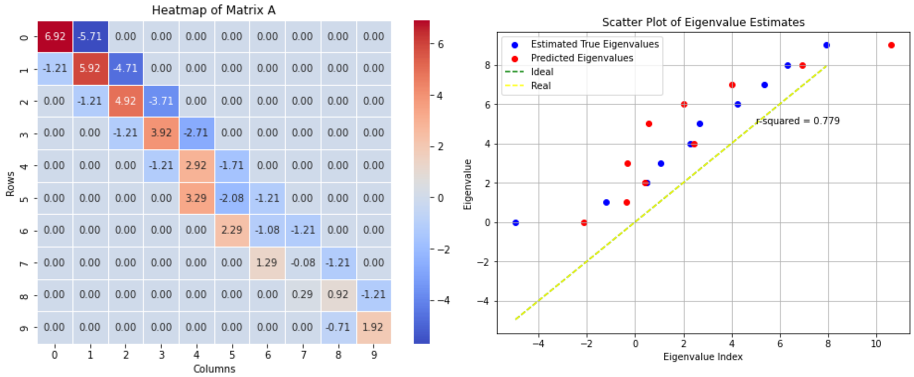
The non-symmetric Lanczos method[26] offers appealing characteristics that could enhance a Jacobi Davidson framework, notably its maintenance of both left and right bi-orthogonal bases generated by A^* and A respectively. The biJD algorithm combines Lanczos two-sided iterations with solving the correction equation for both left and right Ritz pairs, offering faster convergence and an effective restarting scheme.

The Davidson method, initially a diagonally preconditioned variant of the Lanczos method for symmetric eigenproblems, focuses on finding eigenpairs closest to σ. Extensions to finding more eigenpairs are feasible, with converged eigenpairs either remaining in the basis or being locked out. This is also implemented in our program.

BiJD, however, is computationally more demanding than Jacobi Davidson, requiring matrix-vector multiplication with A^*. While it uses twice as much storage due to the left space W and its image $L = A^*W$, savings can be made by computing residuals explicitly.

One notable advantage of biJD is its ability to obtain left eigenvectors almost effortlessly and with similar accuracy as right ones. These left eigenvectors can be valuable, aiding in deflating converged eigenpairs[19] and estimating the condition number of required eigenvalues.

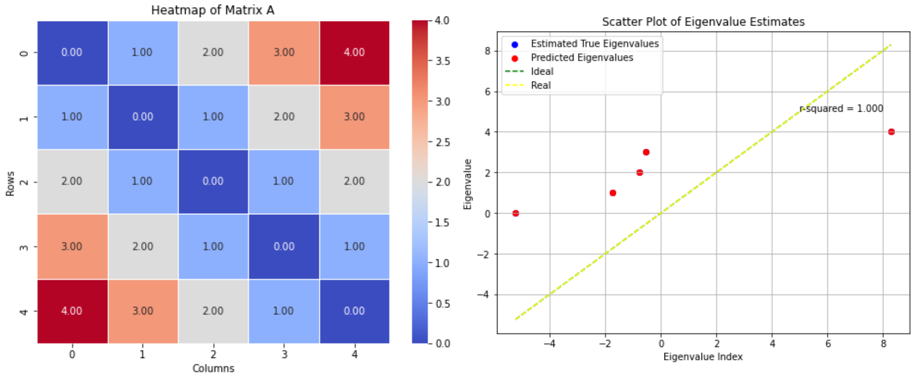
The Rayleigh quotient, easily computed in biJD, serves as a valuable means of assessing eigenvalue convergence. BiJD inherits several characteristics advantageous over conventional JD, including explicit biorthogonalization and flexibility in accommodating various restarting techniques.



	0	1	2	3	4	5	6	7	8	9
0	6.92	-5.71	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1	-1.21	5.92	-4.71	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	0.00	-1.21	4.92	-3.71	0.00	0.00	0.00	0.00	0.00	0.00
3	0.00	0.00	-1.21	3.92	-2.71	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	-1.21	2.92	-1.71	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	3.29	-2.08	-1.21	0.00	0.00	0.00
6	0.00	0.00	0.00	0.00	0.00	2.29	-1.08	-1.21	0.00	0.00
7	0.00	0.00	0.00	0.00	0.00	0.00	1.29	-0.08	-1.21	0.00
8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.29	0.92	-1.21
9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	-0.71	1.92

Figure 1: Sparsity map and scatter plot of true vs eigenvalues predicted by biJD algorithm for a diagonally dominant, ill conditioned, tridiagonal dorr matrix. The r^2 score is also mentioned in the inset.

The biJD algorithm offers the capability to employ spectral projection for deflating converged eigenpairs. By utilizing both left and right eigenpairs, it provides an estimate of the condition number of the target eigenvalue, indicating



	0	1	2	3	4
0	0.00	1.00	2.00	3.00	4.00
1	1.00	0.00	1.00	2.00	3.00
2	2.00	1.00	0.00	1.00	2.00
3	3.00	2.00	1.00	0.00	1.00
4	4.00	3.00	2.00	1.00	0.00

Figure 2: Sparsity map and scatter plot of true vs eigenvalues predicted by biJD algorithm for a symmetric fiedler matrix. The r^2 score is also mentioned in the inset.

the reliability of the computed eigenpair. Even before reaching convergence, identifying an ill-conditioned eigenvalue can potentially accelerate the correction equation by approximately mitigating this ill-conditioning through a similarity transformation.

The Rayleigh quotient of a vector can closely approximate the desired eigenvalues, even if the vector is a linear combination of unrelated eigenvectors. This occurrence is common for normal matrices when σ lies within the spectrum's interior, and for non-normal matrices, it's also feasible for exterior eigenvalues. Given that the Rayleigh quotient computation is computationally inexpensive within the biJD framework, it serves as an effective means of evaluating eigenvalue convergence.

Overall, while biJD's use of twice the storage and its computational demands pose challenges, its ability to obtain left eigenvectors and flexibility in restarting techniques make it a promising approach for certain problems.

6 Results

The biJD algorithm implemented using the details mentioned in the previous section was used to test the convergence and behaviour of the program with respect to the left and right residuals and the correctness of the eigenvalues predicted by the program, especially in ill-conditioned and non-symmetric cases. The matrices used fro running test cases were taken from The Test Matrix Toolkit developed by Prof Higham[27].

The program was explicitly tested on matrices which include[28],

1. A matrix whose columns repeat cyclically.
2. A diagonally dominant, ill conditioned and tridiagonal matrix.
3. A symmetric matrix.
4. A matrix with ill conditioned eigenvalues.
5. A singular matrix that is sparse.
6. A sparse block tridiagonal matrix from the poisson problem.
7. An explicitly symmetric and anti-symmetric matrix constructed using the random function in python.

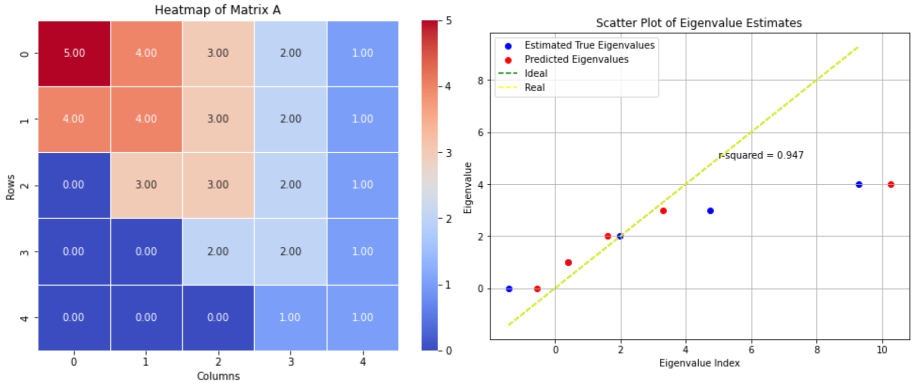


Figure 3: Sparsity map and scatter plot of true vs eigenvalues predicted by biJD algorithm for a matrix with ill conditioned eigenvalues (frank matrix). The r^2 score is also mentioned in the inset.

We observe that the eigenvalues predicted by the program and the true eigenvalues match perfectly if the matrix input is symmetric, else there is mismatch between these values. These are qualitatively described by the scatter plot that plots both of these arrays and also computes the R^2 value describes the closeness of those values.

The heat maps of the matrix inputs are also shown to visualize the inherent symmetry/antisymmetry in these structures and how the level of antisymmetry affects the deviation of computed eigenvalues to the tru eigenvalues of the matrix inputs.

When i tried plotting the convergence of residuals, the graph was showing a harmonic behaviour. It is hence difficult to prove the claims as mentioned in

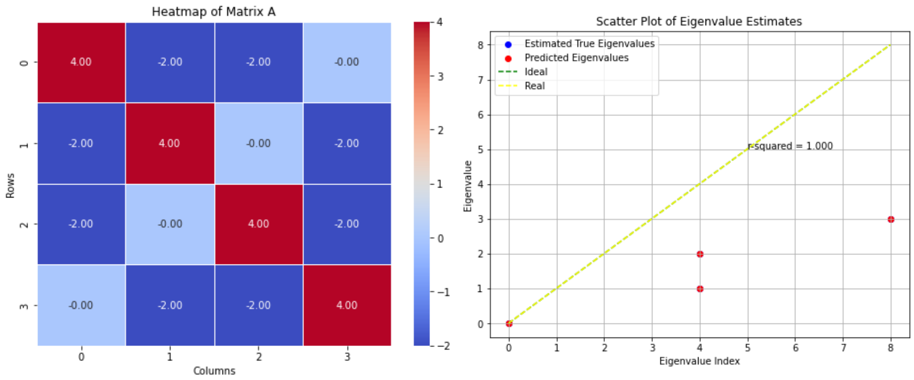


Figure 4: Sparsity map and scatter plot of true vs eigenvalues predicted by biJD algorithm for a sparse neumann matrix. The r^2 score is also mentioned in the inset.

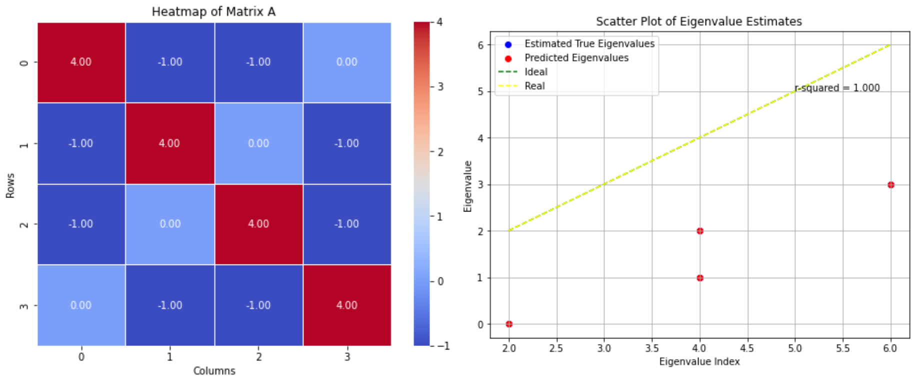


Figure 5: Sparsity map and scatter plot of true vs eigenvalues predicted by biJD algorithm for a block tridiagonal sparse poisson matrix. The r^2 score is also mentioned in the inset.

the Stathopoulos paper regarding the differential convergence of left and right residuals at present. Future work should be done to fix this disparity and obtain convergence graphs consistent with the theory mentioned in the previous section. All the codes used are available at $https://github.com/aksharma412/Biorthogonal-Jacobi-Davidson-algorithm.git$.

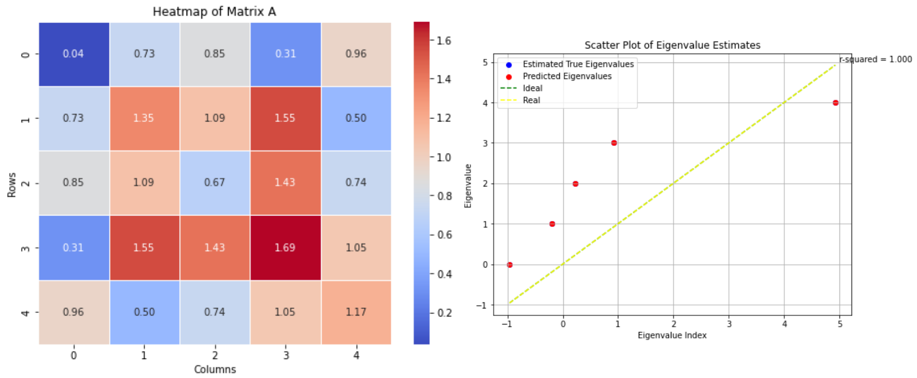


Figure 6: Sparsity map and scatter plot of true vs eigenvalues predicted by biJD algorithm for an explicitly symmetric matrix using numpy.rand(). The r^2 score is also mentioned in the inset.

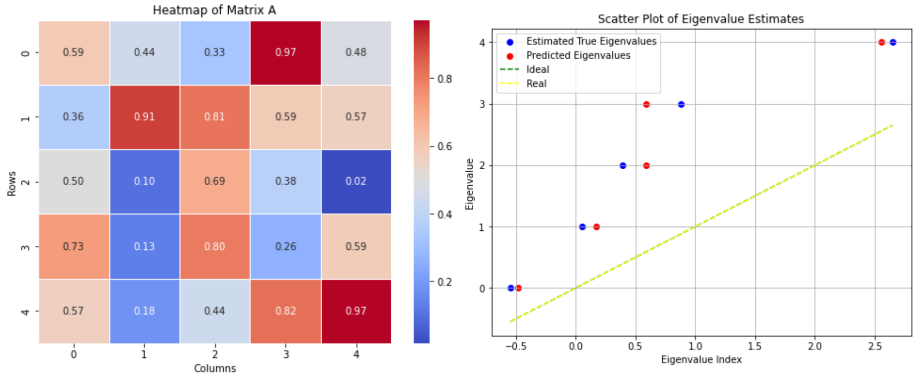


Figure 7: Sparsity map and scatter plot of true vs eigenvalues predicted by biJD algorithm for an explicitly antisymmetric matrix using numpy.rand(). The r^2 score is also mentioned in the inset.

References

[1] Owe Axelsson. *Iterative solution methods*. Cambridge university press, 1996.

[2] Ronald B Morgan. “A restarted GMRES method augmented with eigenvectors”. In: *SIAM Journal on Matrix Analysis and Applications* 16.4 (1995), pp. 1154–1171.

[3] Justin M Turney et al. “Psi4: an open-source ab initio electronic structure program”. In: *Wiley Interdisciplinary Reviews: Computational Molecular Science* 2.4 (2012), pp. 556–565.

[4] E Apra et al. “NWChem”. In: (2020).

[5] Yuri Alexeev et al. “GAMESS as a free quantum-mechanical platform for drug research”. In: *Current topics in medicinal chemistry* 12.18 (2012), pp. 2013–2033.

[6] Paolo Giannozzi et al. “QUANTUM ESPRESSO: a modular and open-source software project for quantum simulations of materials”. In: *Journal of physics: Condensed matter* 21.39 (2009), p. 395502.

[7] Johan H van Lenthe and Peter Pulay. “A space-saving modification of Davidson’s eigenvector algorithm”. In: *Journal of computational chemistry* 11.10 (1990), pp. 1164–1168.

[8] Yihan Shao et al. “Advances in molecular quantum chemistry contained in the Q-Chem 4 program package”. In: *Molecular Physics* 113.2 (2015), pp. 184–215.

[9] Frank Neese. “The ORCA program system”. In: *Wiley Interdisciplinary Reviews: Computational Molecular Science* 2.1 (2012), pp. 73–78.

[10] Kestutis Aidas et al. “The D alton quantum chemistry program system”. In: *Wiley Interdisciplinary Reviews: Computational Molecular Science* 4.3 (2014), pp. 269–284.

[11] Qiming Sun et al. “PySCF: the Python-based simulations of chemistry framework”. In: *Wiley Interdisciplinary Reviews: Computational Molecular Science* 8.1 (2018), e1340.

[12] Michel Crouzeix, Bernard Philippe, and Miloud Sadkane. “The davidson method”. In: *SIAM Journal on Scientific Computing* 15.1 (1994), pp. 62–76.

[13] Cornelius Lanczos. “An iteration method for the solution of the eigenvalue problem of linear differential and integral operators”. In: (1950).

[14] Walter Edwin Arnoldi. “The principle of minimized iterations in the solution of the matrix eigenvalue problem”. In: *Quarterly of applied mathematics* 9.1 (1951), pp. 17–29.

[15] Kensuke Aishima. “Global convergence of the restarted Lanczos and Jacobi–Davidson methods for symmetric eigenvalue problems”. In: *Numerische Mathematik* 131.3 (2015), pp. 405–423.

[16] Ronald Morgan. “On restarting the Arnoldi method for large nonsymmetric eigenvalue problems”. In: *Mathematics of Computation* 65.215 (1996), pp. 1213–1230.

[17] Danny C Sorensen. “Implicitly restarted Arnoldi/Lanczos methods for large scale eigenvalue calculations”. In: *Parallel Numerical Algorithms*. Springer, 1997, pp. 119–165.

[18] Robert M Parrish, Edward G Hohenstein, and Todd J Martínez. ““Balancing” the block Davidson–Liu algorithm”. In: *Journal of Chemical Theory and Computation* 12.7 (2016), pp. 3003–3007.

[19] Andrew Chapman and Yousef Saad. “Deflated and augmented Krylov subspace techniques”. In: *Numerical linear algebra with applications* 4.1 (1997), pp. 43–66.

[20] Yousef Saad. *Iterative methods for sparse linear systems.* SIAM, 2003.

[21] Steven J Leon, Åke Björck, and Walter Gander. “Gram-Schmidt orthogonalization: 100 years and more”. In: *Numerical Linear Algebra with Applications* 20.3 (2013), pp. 492–532.

[22] Mårten Gulliksson. “On the modified Gram-Schmidt algorithm for weighted and constrained linear least squares problems”. In: *BIT Numerical Mathematics* 35 (1995), pp. 453–468.

[23] Andreas Stathopoulos. “A case for a biorthogonal Jacobi–Davidson method: Restarting and correction equation”. In: *SIAM Journal on Matrix Analysis and Applications* 24.1 (2002), pp. 238–259.

[24] Menno Genseberger and Gerard LG Sleijpen. “Alternative correction equations in the Jacobi-Davidson method”. In: *Numerical linear algebra with applications* 6.3 (1999), pp. 235–253.

[25] Christopher Beattie. “Harmonic Ritz and Lehmann bounds”. In: *Electron. Trans. Numer. Anal* 7 (1998), pp. 18–39.

[26] Beresford N Parlett, Derek R Taylor, and Zhishun A Liu. “A look-ahead Lanczos algorithm for unsymmetric matrices”. In: *Mathematics of computation* 44.169 (1985), pp. 105–124.

[27] Nicholas J Higham. *The matrix computation toolbox for MATLAB (version 1.0).* Citeseer, 2002.

[28] Zhaojun Bai et al. “A test matrix collection for non-Hermitian eigenvalue problems”. In: *Prof. Z. Bai, Dept. of Mathematics* 751 (1996), pp. 40506–0027.