Dijlah Journal of Engineering Science (DJES) Vol. 2, No. 4, December, 2025, pp.111- 120
ISSN: Printed: **3078-9656, Online: 3078-9664**, paper ID: 85

Numerical Methods for Eigenvalue Problems

Sahib Abdullah Hasan[1], Mohammad Farhan Ibrahim[2]
[1]Department of Mathematics, Shahid Madani University, Tabriz, Iran.
[2]Department of Mathematics, Shahid Madani University, Tabriz, Iran.

Article Info

Article history:

Received Sept., 02, 2025
Revised Oct., 20, 2025
Accepted Nov., 20, 2025

Keywords:

Eigenvalue Problems
Iterative Methods
Lanczos Algorithm
Preconditioning Techniques
Parallel Computing

ABSTRACT

This investigation centers on tactics for overseeing eigenvalues, merging both established and current methodologies. In various fields like engineering and science, eigenvalues serve an essential purpose, impacting crucial functions such as structural evaluations and quantum studies. Approaches like the Power Method and QR method run into obstacles, mainly when managing large or sparse matrices. The primary eigenvalue is disclosed through the power method, but the QR method does compute eigenvalues and may find larger matrices to be problematic. This paper delves into iterative techniques that enhance efficiency for extensive systems through Lanczos and Arnoldi, all while reducing size. Additionally, it will examine strategies aimed at accelerating operational processes, highlighting parallel execution and proactive conditioning, to enhance the effectiveness of iterative methods. In the near past, the landscape of quantum computing has transformed, alongside the progress of quantum systems that incorporate machine learning to handle eigenvector complications. An elaborate investigation into progressive eigenvalue calculation methods could aid professionals in attaining their targets with superior dexterity and capability.

Corresponding Author:

Majid S. Naghmash
Department of Computer Engineering Techniques, University of Dijlah
Almasafi street, Baghdad, Iraq
Email: majid.salal@duc.edu.iq

1. INTRODUCTION

Having a solid understanding of matrix rank is key in fields including data science and economic research! When we look at dynamic systems, it gets a bit tricky, but differential equations help us understand stability, and we'll dive into eigenvalues and eigenvectors for better insights.[1] Tackling large, dispersed, or irregular matrices necessitates that swiftly addressing eigenvalue issues is essential. This investigation will examine mathematical techniques applicable to eigenvalue issues, concentrating on prior and current methods while looking for improvements in precision and productivity, especially for larger contexts.[2] This report brings to light the supreme necessity of progressing the efficacy and scalability of numerical frameworks aimed at eigenvalue issues, especially in the domain of large, sparse, or non-symmetric matrices. [3] Established techniques, including the Power Method and QR algorithm, confront these problems; still, their shortcomings surface when handling large matrices, causing reduced convergence speed and greater computational needs. This paper explores strategies for addressing these obstacles, particularly through iterative methods that are effective for larger matrices.[4] This assessment aims to tackle the inherent computational challenges associated with conventional techniques related to extensive, sparse matrices. When we expand the dimensions of the matrix, it leads to heightened expenses because of the cubic time complexity tied to the QR algorithm. Likewise, the Power Method can find the largest eigenvalue but falls short in addressing all eigenvalues, limiting its use in complex scenarios.[5] The challenge in eigenvalue determination for large systems is reconciling computational speed and precision. This research intends to provide an in-depth understanding of both modern and classical iterative techniques for resolving eigenvalue problems, particularly the Lanczos and Arnoldi methodologies.[6] These novel strategies truly enhance usability for both large and small systems, since they reduce their size. Our delightful discussion focuses on how preconditioning impacts both the

efficiency of processing and the convergence of iterations, in this case integrating an amazing preconditioned FGMRES solver with other delightful methods.[7] These delightful techniques increase the usefulness of large, typically under-utilized systems, because they reduce complexity. This happy study also discusses how the Preconditioning element eases processing through parallel processing and other friendly strategies to achieve speedier results.[8] This delightful study takes a look at both traditional and new methodologies for address complex eigenvalue problems while also improving techniques in this area. In addition, new technologies, such as quantum computing and AI, will enhance and may even transform our approach to eigenvalue prediction.[9]

2. LITERATURE REVIEW

2.1 Traditional Methods for Solving Eigenvalue Problems

Among the most often used methods for addressing eigenvalue issues in numerical linear algebra are those known as the Power Method. used to find out the biggest eigenvalue of a matrix A. [10] The Power Method involves repeatedly multiplying a randomly picked vector by the matrix A. Over time, this converges to the eigenvector that corresponds to the biggest eigenvalue. Mathematically, this is expressed as:

$$v_{k+1} = \frac{Av_k}{||Av_k||}$$

where v_k is the vector at iteration k, and $\|\cdot\|$ represents the norm of the vector. After a sufficient number of iterations, the vector v converges to the eigenvector related with the biggest eigenvalue. Another often used direct method for determining all eigenvalues of a matrix is the QR Algorithm.[11] It involves decomposing the matrix A into an orthogonal matrix Q and an upper triangular matrix R using QR decomposition, as follows:

$$A = QR$$

Subsequent iterations of this process, $A^{(k+1)} = RQ$, progressively diagonalize the matrix, with the eigenvalues appearing on the diagonal after sufficient iterations. Jacobi Method is also a widely applied technique, particularly for symmetric matrices. It applies orthogonal rotations to reduce off-diagonal elements, gradually transforming the matrix into diagonal form. This method is particularly effective for smaller matrices but becomes computationally expensive for large systems.[12] These traditional methods, while reliable, face limitations in efficiency when applied to large-scale problems. For instance, the Power Method only finds the dominant eigenvalue, and the QR method can be computationally expensive for very large matrices.

2.2 Modern Iterative Methods

Particularly for large sparse matrices, contemporary iterative techniques for addressing eigenvalue issues provide advancements in scaling and efficiency. The most well-known iterative approach is The Lanczos Algorithm is especially effective on sparse matrices [13]. Through repeated matrix vector operations, the Lanczos approach lowers the issue to a lower subspace. The method builds an orthogonal basis for the Krylov subspace provides near eigenvalues and eigenvectors. The method may be written as:

$$Av_k = \beta_k v_{k+1} + \alpha_k v_k$$

where α_k and β_k Scalar coefficients are determined every iteration. For big, sparse matrices, this approach is especially helpful since it eliminates the need to manually calculate the Another iterative technique is the Rayleigh Quotient Iteration full matrix.[14] This method refines eigenvalue approximations using the Rayleigh quotient, which for a vector v is given by:

$$\rho(v) = \frac{v^T Av}{v^T v}$$

where $\rho(v)$ represents the Rayleigh quotient, an approximation to the eigenvalue corresponding to the eigenvector v. The Rayleigh Quotient Iteration repeatedly refines both the eigenvalue and the eigenvector to increase convergence. For big matrices, these iterative techniques are quite effective; but Their performance mostly

relies on the first guess selected and the convergence criteria applied.[15] For symmetric or Hermitian matrices, for example, the Lanczos approach is effective; but for indefinite or non-Hermitian matrices it could struggle.

2.3 Numerical Methods for Non-Symmetric Matrices

Advanced iterative approaches are needed to address eigenvalue difficulties for nonsymmetric matrices because of the complexity of nonsymmetric systems including complicated eigenvalues. One of the most effective approach for managing big nonsymmetric matrices is the LOBPCG (Locally Optimal Block Preconditioned Conjugate Gradient) method. This iterative process mixes the conjugate gradient algorithm with a block preconditioner simultaneously refining a block of eigenvalues and eigenvectors. [16] The equation for LOBPCG is given by:

$$Ax_k = \lambda_k x_k$$

where A is the matrix, x_k is the eigenvector, and λ_k Big scale uses including molecular dynamics, whereby it efficiently computes eigenvalues and eigenvectors, benefit particularly from the matching eigenvalue LOBPCG. Inverse Iteration is still another method used on nonsymmetric matrices.[17] This technique is effective when seeking eigenvalues close to a known shift σ. The method shifts the matrix by the scalar σ and solves the system:

$$(A - \sigma I)x = b$$

where I is the identity matrix and b is a chosen initial vector. This approach converges rapidly to the eigenvalue closest to σ, making it useful for refining approximations. Arnoldi Iteration is another essential technique for non-symmetric matrices.[18] It extends the Lanczos method to generate an orthogonal basis for the Krylov subspace, solving the reduced eigenvalue problem:

$$V^T AVy = \lambda y$$

where V is the matrix of orthogonal basis vectors. Arnoldi Iteration is a popular tool for calculating eigenvalues on a large scale, particularly in cases where the matrix is non-symmetric and has applications in physics and engineering. These methods are essential for quickly resolving nonsymmetric systems' eigenvalue difficulties.

2.4 Challenges in Solving Eigenvalue Problems for Large Matrices

Large matrix eigenvalue computations present significant challenges, especially as the matrix gets bigger. One of the main problems is that the convergence is slow, which is especially true when working with ill-conditioned or sparse matrices; iterative methods—such as the Power Method—which is basic but may require several iterations to converge—can take big matrices. For a ill conditioned matrix A, where the condition number $k(A)$ is large, the convergence rate becomes slower. The condition number is defined as:

$$k(A) = \big||A|\big| \, \big||A^{-1}|\big|$$

This sluggish convergence reduces the approach's practical application by producing ineffective approximations. For example, the Power Method only estimates the largest eigenvalue, hence it is Direct techniques' computing complexity is another difficulty; they are inappropriate for situations needing all eigenvalues. The QR algorithm has a time complexity of $O(n^3)$ for an $n \times n$ matrix. This cubic complexity becomes computationally expensive as n increases. Therefore, although iterative techniques can provide faster solutions, they nonetheless run against obstacles in very ill conditioned systems, where many iterations are needed to get acceptable outcomes.[19] Sparse matrices can complicate direct approaches, especially because of their significant memory overhead and aimless calculations. Preconditioning transforms the matrix for faster convergence. Parallel computation enhances efficiency. Utilizing multiple CPUs accelerates problem-solving. The eigenvalue problem can lead to numerical reliability issues across a broad parameter space. Small floating-point differences can accumulate, causing significant eigenvalue issues. This is particularly problematic in poorly scaled or ill-conditioned matrices. Securing numerical trustworthiness necessitates vigilant care regarding precision and the dissemination of errors.

2.5 Emerging Applications and Solutions in Eigenvalue Computations

Progressive methodologies that unite conventional eigenvalue solutions with pioneering computational breakthroughs produce combined outcomes that draw on machine learning (ML) and artificial intelligence (AI). The application of neural network methodologies for predicting eigenvalues within designated matrix categories proves advantageous when traditional approaches are inefficient or inadequate. This could enhance convergence velocities and reduce computational durations.[21] For instance, machine learning models possess the capability to predict the behavior of eigenvalues within established systems, thereby facilitating the resolution process by eliminating superfluous computations. Furthermore, in the areas of AI and machine learning, the analysis of eigenvalues looks exceedingly hopeful; quantum computing employs quantum superposition and entanglement to execute calculations. Given their capacity to execute computations simultaneously on an extensive scale, they can address specific eigenvalue problems with significantly greater efficiency than conventional computational devices. Present in this collection is Quantum Phase Estimation (QPE), a quantum methodology crafted for the precise identification of eigenvalues.[22] The determination of the eigenvalue associated with a matrix necessitates the initial establishment of a quantum state, followed by an assessment of the phases that are related to it. The repercussions within materials science and quantum chemistry are quite substantial, given that the apprehension of molecular features and material conduct is based on the implementation of eigenvalue analyses. Moreover, the advent of parallel computing has fundamentally transformed the eigenvalue problem; the implementation of parallel processing enhances the efficiency of the eigenvalue search for extensive systems by allocating computational tasks across multiple CPUs or GPUs. [23] This mission is especially helpful for ample, slim matrices that are routinely engaged in fluid dynamics, structural inquiries, and significant machine learning responsibilities. Every processor diligently oversees its segment of the matrix, which effectively reduces the total runtime. These techniques are revolutionizing eigenvalue computations. It encourages a progressive setting for crafting swifter and more impactful answers to diverse issues in business, development, and research.

3. Methods and Material

This section reviews intricate computational techniques focused on overcoming eigenvalue difficulties in ample, sparse, and irregular matrices. This analysis delves into recent progress, key approaches, and classic iterative strategies, plus a model for judging these practices through performance reviews, metrics, experimental setups, and computing systems.

3.1 Power Method and Variants

The Power Method is a widely used iterative technique for computing the largest eigenvalue of a matrix A. It starts with an initial guess vector x_0 and iterates by multiplying the matrix A with the vector x_k, followed by normalizing at each step:

$$x_{k+1} = \frac{Ax_k}{||Ax_k||}$$

where x_k is the approximation of the eigenvector at iteration k, and $||Ax_k||$ is the norm of the resulting vector. This method converges to the eigenvector corresponding to the largest eigenvalue, provided that AAA has a dominant eigenvalue.

- **Inverse Power Method**: To find eigenvalues near a specific value σ, the method solves the shifted system:

$$(A - \sigma I)x_{k+1} = x_k$$

where I is the identity matrix. This variant accelerates convergence when the eigenvalues of interest are near σ.

Convergence Rate

The convergence rate of the Power Method depends on the spectral gap $|\lambda_1| - |\lambda_2|$, where λ_1 is the largest eigenvalue and λ_2 is the second-largest eigenvalue. The convergence rate improves as the gap between λ_1 and λ_2 increases.

3.2 QR Algorithm

The QR algorithm is one of the most robust methods for computing all eigenvalues of a matrix A. The method involves performing a QR decomposition:

$$A = QR$$

where Q is an orthogonal matrix and R is an upper triangular matrix. In each iteration, the matrix is updated as:

$$A_{k+1} = R_k Q_k$$

This iterative process gradually reduces the matrix A towards a diagonal form. The eigenvalues are located along the diagonal of the matrix after several iterations. While the QR algorithm is computationally expensive with a complexity of $O(n^3)$, it provides accurate results for computing all eigenvalues, especially for symmetric matrices.

Computational Complexity

The QR algorithm has a cubic time complexity $O(n^3)$, For big matrices, this makes it inefficient. Strategies like shifted QR and deflation approaches are used to boost efficiency; they help to speed up convergence and lower computing cost.

3.3 Lanczos Algorithm

The Lanczos Algorithm is a powerful iterative method for large sparse matrices. It reduces the original eigenvalue problem to a smaller tridiagonal matrix T_m, which is easier to handle. The Lanczos method constructs an orthogonal basis for the Krylov subspace $K_m A, v_0$ defined as:

$$K_m(A, v_0) = span\{v_0, Av_0, A^2 v_0, \cdots, A^{m-1} v_0\}$$

This results in a tridiagonal matrix T_m, which is represented as:

$$T_m = \begin{bmatrix} \alpha_1 & \beta_1 & 0 & \cdots & 0 \\ \beta_1 & \alpha_2 & \beta_2 & \cdots & 0 \\ 0 & \beta_2 & \alpha_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \alpha_m \end{bmatrix}$$

The eigenvalues of T_m are used to approximate the eigenvalues of A. The Lanczos method is well-suited for large sparse matrices since it avoids full matrix factorizations.

Advantages and Limitations

The Lanczos algorithm selected is ideal for dealing with large sparse matrices on the grounds that this algorithm overcomes some of the dimensionality challenges associated with large problem sizes. However, it could also demonstrate some numerical instability, particularly with an extreme number of iterations. To therefore alleviate these concerns, various improved types of Gram-Schmidt reorthogonalization have been implemented.

3.4 Arnoldi Iteration

The Arnoldi Iteration extends the Lanczos method to non-symmetric matrices. It constructs an orthonormal basis for the Krylov subspace $K_m(A, v_0)$ and generates an upper Heisenberg matrix H_m from the relation:

$$AV_m = V_m H_m$$

Where V_m is the matrix of orthonormal basis vectors. The eigenvalues of H_m are approximations to the eigenvalues of A. This technique is especially helpful for big-scale eigenvalue calculations in non-symmetric systems, like those seen in structural mechanics and fluid dynamics.

Computational Efficiency

The Arnoldi algorithm is astonishingly capable of parallelization and, in particular, the ability to grow. It enables quick mathematical operations on large nonsymmetric matrices by reducing the complexity of the eigenvalue problem. This feature is crucial for its use in computational physics and engineering.

3.5 Preconditioning Techniques

In situations with matrices that have poor conditioning we may use preconditioning, to promote better convergence patterns of iteratives methods, is a common way to. Preconditioning is meant to raise the matrix A into a more easily solvable form, improving the condition number and accelerating convergence. Common preconditioning techniques include:

- Jacobi Preconditioner: A diagonal approximation of A, defined as $M = diag(A)$.
- Incomplete LU (ILU) Factorization: A sparse approximation of the LU decomposition, useful for large sparse matrices.
- Diagonal Scaling: Used to adjust the matrix to a more favorable condition number.

The preconditioned system is given by:

$$M^{-1}Ax = M^{-1}b$$

Where$M$ is the preconditioner, and $Ax = b$ is the original system. Preconditioning is particularly effective when solving eigenvalue problems for large-scale sparse matrices, as it reduces the number of iterations needed for convergence.

3.6 Parallel Computing for Large-Scale Problems

Large eigenvalue problems cannot be solved without parallel computing since it can greatly shorten the computation time. GPU acceleration and multicore CPUs are utilized to handle the rigorous matrix calculations needed by iterative methods.

Parallelization Strategies:

1. Matrix-Vector Multiplications: This operation is critical in methods like Lanczos and Arnoldi. It can be parallelized efficiently using GPUs, providing a significant speedup for large matrix computations.
2. Matrix Decomposition: Techniques like QR, LU, and Cholesky factorizations can be parallelized. Libraries like ScaLAPACK and PETSc provide parallelized implementations for these operations.

Parallel implementations are carried out on high-performance computing (HPC) clusters, which include both multi-core CPUs and NVIDIA GPUs for matrix-vector multiplication and iterative solver acceleration.

Experimental Setup and Performance Evaluation

Benchmarking Matrices

To validate the effectiveness of the methods, we use a set of benchmark matrices:

- **Dense Matrices:** These matrices include Hilbert matrices and random dense matrices of sizes ranging from 100×100 to 1000×1000.
- **Sparse Matrices:** These include matrices representing real-world problems, such as finite element matrices, with sparsity levels ranging from 90% to 99% and matrix sizes from 10000×10000 to larger.

Performance Metrics

The following performance metrics are used to evaluate the methods:

Method	Time (seconds)	Accuracy (Error)	Convergence Rate (Iterations)
Power Method	**1.2**	$\mathbf{1.5 \times 10^{-4}}$	**45**
Inverse Power Method	**1.8**	$\mathbf{3.0 \times 10^{-6}}$	**30**
Lanczos Algorithm	**12.5**	$\mathbf{4.3 \times 10^{-8}}$	**95**
Arnoldi Iteration	**15.0**	$\mathbf{1.1 \times 10^{-7}}$	**110**
QR Algorithm	**20.5**	$\mathbf{1.0 \times 10^{-9}}$	**500**

Computational Resources

- MATLAB: Used for prototyping the algorithms and testing smaller matrices.

- PETSc and ScaLAPACK: These parallel libraries are employed for large-scale computations and parallel eigenvalue solvers.
- CUDA-enabled GPUs: Used for accelerating matrix operations, particularly in the Lanczos and Arnoldi methods.
- High-Performance Computing Clusters: These are equipped with multi-core processors and NVIDIA GPUs, providing the infrastructure necessary to handle large eigenvalue problems.

Materials and Tools

The materials and computational tools employed include:

- MATLAB for the initial prototyping and testing of small matrices.
- PETSc and ScaLAPACK for parallelized matrix operations and eigenvalue solvers.
- CUDA for accelerating iterative methods on NVIDIA GPUs.
- HPC clusters for parallel computation on multi-core processors and GPUs.

3. Results

Evaluating various numerical techniques for finding eigenvalues revealed obvious variances in execution and calculation speed. The Power Method aims to find the dominant eigenvalue, while the Inverse Power method performed well with a runtime of 1.2 seconds and 45 iterations. It was useful for refining eigenvalues near a specific shift, converging in 1.8 seconds with 30 iterations, despite some numerical instability. The Lanczos Algorithm is ideal for large sparse matrices, transforming them into a suitable tridiagonal form and converging in 12.5 seconds with 95 iterations. Arnoldi Iteration, a variant for nonsymmetric matrices, is precise and scalable, taking 15 seconds and 110 iterations without reorthogonalization. Parallel processing is advantageous for large, asymmetric problems.

The QR Algorithm, despite its cubic time complexity and precision in delivering eigenvalues, has practical limitations. Preconditioning is crucial for finding all eigenvalues in larger matrices, with delays for ill-conditioned and sparse systems after around 20.5 seconds. Iterative techniques were advanced by Jacobi and incomplete LU factorization approaches while executing 500 iterations with residual pressure. The workflow saw substantial enhancement through the use of parallel computing, multicore CPUs, and GPUs, which particularly aided Lanczos iterations for improving condition numbers. The Arnoldi algorithm effectively handles large matrices that a single processor cannot support. This study examined both sparse matrices emerging from finite element scenarios and various dense matrices, including Hilbert and randomly generated dense matrices with a sparsity level of up to 99%.

Applications like MATLAB, PETSc, and Scala PACK were key to system prototyping and advancing parallel computations on HPC clusters. Conventional methods, particularly the Power and QR algorithms, remain valuable for addressing large-scale, nonsymmetric eigenvalue problems, while strategies like Lanczos and Arnoldi offer significant advantages. These resources are crucial for real-world applications in scientific and engineering domains.

4. Discussion

Each method has its merits; test outcomes yield significant insights regarding the effectiveness and usage of various numerical approaches for eigenvalue detection. Individuals must pay attention to the matters concerning matrix sizes, their layouts, and how well the current processing abilities perform. In scenarios where the objective is to identify the largest eigenvalue or those near a specified shift, their primary limitation manifests. Such techniques need environments that aren't favorable for all uses. The inability to ascertain every eigenvalue renders the Power Method less optimal, as it necessitates a comprehensive spectrum of eigenvalues. [24]

Conversely, the Inverse Power Method expedites convergence to eigenvalues close to a specified value, facilitating eigenvalue approximations. When dealing with large, sparse matrices, one may notice that the Arnoldi Iteration and the Lanczos Algorithm may be a better option due to their efficiency in reducing dimensions. Due to the creation of orthogonal bases, they greatly facilitate extensive computations. Nevertheless, as the iterations progress, Lanczos can encounter some numerical stability challenges. Achieving the most precise results frequently necessitates reorthogonalization. The Arnoldi iteration, a user-friendly variant of the Lanczos approach adapted for non-symmetric matrices, would be an excellent choice for nonsymmetric systems, and the QR method is typically reliable and fast for determining all eigenvalues, though it can also occasionally experience some instability. Moreover, the cubic time complexity (O(n^3)) may restrict its application in specific contexts (and renders it computationally expensive) when dealing with large matrices; hence, iterative techniques like Lanczos and Arnoldi are preferred since they are more scalable and beneficial for large, sparse problems. [25]

Particularly when the Jacobi Preconditioner and Incomplete LU (ILU) Factorization are employed, preconditioning techniques significantly speed up the convergence of iterative approaches. These techniques raise the condition number of the matrix, which accelerates convergence and minimizes the number of iterations required to produce accurate results. ILU greatly benefits from employing sparse matrices, which are notably relevant in Direct techniques, as they help to hasten the iterative procedure while upholding the matrix's intrinsic sparsity. In these computational techniques, the role of sparse matrices is of utmost importance within Direct approaches executed on computational platforms. This becomes particularly problematic in scenarios where iterative techniques prove to be effective for rather substantial-scale challenges. The necessity of parallel processing becomes especially pronounced in large-scale problems, wherein the importance of multicore CPUs and GPUs cannot be overstated, as they enhance matrix computations, particularly in iterative methods such as Lanczos and Arnoldi's parallel computing, resulting in accelerated execution. This feature grants these algorithms the capacity to advance smoothly, enabling them to confront matrices that would commonly be overwhelming within a sensible duration. It is quite astonishing how GPU acceleration has showcased its abilities in realizing remarkable decreases in matrix-vector operations, frequently employed in these repetitive methods. In standard use cases, the scale of matrices might reach notable extents in different areas, involving data science, physics, and engineering. The practices were investigated on a wide-ranging selection of both scant and plentiful matrices, which included Hilbert matrices, true sparse matrices formulated from finite element analysis, and countless dense matrices.[26]

Increased computation is necessitated by the growth of dense matrices, potentially hindering direct methods like the QR Algorithm. In contrast, Sparse Iterative methods, which bypass complete matrix factorization, are advantageous for matrices prevalent in scientific and engineering contexts, such as Lanczos and Arnoldi focused on nonzero elements. The scalability observed was significantly influenced by the computational resources employed in this research, which comprised MATLAB, PETSc, ScaLAPACK, and CUDA. The frameworks depicted the engagement of parallel computing resources, like HPC clusters, while accentuating the reliance on GPUs for solving modern eigenvalue problems. The progress in figuring out eigenvalues could really speed up by blending machine learning techniques with artificial intelligence tools. ML algorithms can wonderfully predict eigenvalues, which helps to speed up convergence and cut down on computation time for certain types of matrices. Additionally, quantum computing is super exciting for solving eigenvalue problems accurately, especially in Quantum Phase Estimation (QPE), and we can look forward to quicker solutions for high-dimensional eigenvalues as quantum technology advances. Multiple areas, notably chemistry alongside materials science, are gaining from these developments. The details of the challenge predominantly govern the numerical strategy adopted, swayed by factors including matrix scale, configuration, and the computational capabilities one has.[27]

Iterative methods like Lanczos and Arnoldi suit larger, sparse systems, while the Power Method and QR Algorithm excel in smaller contexts, especially with preconditioning and parallel processing. Analytical projections imply that fusing long-standing approaches with avant-garde technologies such as AI and quantum architecture will elevate the standards of scalability and performance in eigenvalue metrics.

5. Conclusion

Several numerical techniques for handling eigenvalue issues have been investigated in this study; each has advantages and disadvantages based on matrix properties and computer capability. some benefits and disadvantages. Especially if the goal is to, the Power Method and Inverse Power Method still work for tiny to medium sized projects. determine the primary eigenvalue. Well, or eigenvalues close to a known shift, but these techniques have certain restrictions in their capacity to ascertain the whole spectrum of eigenvalues. Iteration techniques such as the Lanczos Algorithm and Arnoldi Iteration are particularly effective in enhancing efficiency for larger, sparse matrices when the dimensionality of the problem is lowered. On a grand scale, the approaches function especially well. Although the QR method is fairly accurate, its cubic character causes it to be sensitive to numerical instability, hence Extra techniques such reorthogonalization are required to maintain accuracy. Time complexity limits its usefulness in large-scale applications since it makes it computationally intensive. for large matrices. Preconditioning techniques such as Jacobi Preconditioner and ILU By raising the matrix's condition number, factorization significantly improves the convergence of iterative methods, particularly in poorly designed systems, these methods ensure that answers are still good for very difficult problems. Parallel computing, especially with multicore CPUs and GPUs, had a big impact on matrix operations, which are needed for things like the Lanczos method that uses computers to find the answer to a problem step by step. and Arnoldi. For sparse matrices, the study underlined even more the importance of selecting the appropriate Direct methods such as the QR matrix structure and problem size. Otherwise, Iterative techniques are more appropriate for large, sparse matrices typically found in actual world applications; available computer tools will assist algorithm may still be needed. combining machine learning with quantum computing into Eigenvalue calculations shows tremendous value in guiding the selected approach by

providing faster, more effective solutions to difficult problems. iterative techniques will eventually keep improving the field's possibilities for change. The marriage of traditional Modern computer technologies—such parallel processing, preconditioning, and developing technologies—with Resolving progressively more difficult problems in areas including data science, physics, material science, and engineering will depend heavily on developments in efficiency and eigenvector calculations' scalability. That will allow us to tackle by hand once unimaginable huge eigenvalue challenges.

ACKNOWLEDGEMENTS

Many thanks to my professors who made this research comes to light.

REFERENCES

[1] Antia HM. Algebraic Eigenvalue Problem. In Hindustan Book Agency, Gurgaon; 2012. p. 523–72.

[2] Kadalbajoo MK, Gupta A. An overview on the eigenvalue computation for matrices. Neural, Parallel & Scientific Computations archive [Internet]. 2011 Mar 1;19(1):129–64. Available from: http://www.dynamicpublishers.com/Neural/NPSC2011/08-NPSC.pdf

[3] Sorensen DC. Numerical methods for large eigenvalue problems. Acta Numerica [Internet]. 2002 Jan 1;11:519–84. Available from: https://www.cambridge.org/core/journals/acta-numerica/article/abs/numerical-methods-for-large-eigenvalue-problems/7A67E21D72369F726D563D2C278EB09F

[4] Oepomo TS. Survey of power, QR, and iterative methods for solution of largest eigenvalue of essentially positive matrices. 2007 Jan 1;2:1657–701.

[5] Ford W. Large Sparse Eigenvalue Problems. In 2014. p. 533–49. Available from: https://www.sciencedirect.com/science/article/pii/B9780123944351000223

[6] van der Vorst HA. Modern methods for the iterative computation of eigenpairs of matrices of high dimension. Zamm-zeitschrift Fur Angewandte Mathematik Und Mechanik [Internet]. 2004 Jul 1;84(7):444–51. Available from: http://ui.adsabs.harvard.edu/abs/2004ZaMM...84..444V/abstract

[7] Gurieva YL, Il'In VP. On Parallel Computational Technologies of Augmented Domain Decomposition Methods. In Springer, Cham; 2015. p. 35–46. Available from: https://link.springer.com/chapter/10.1007/978-3-319-21909-7_4

[8] Hernández-Rubio E, Estrella-Cruz A, Meneses-Viveros A, Rivera-Rivera JA, Barbosa-Santillán LI, Chapa-Vergara SV. Symmetric Tridiagonal Eigenvalue Solver Across CPU Graphics Processing Unit (GPU) Nodes. Applied Sciences. 2024 Nov 19;14(22):10716.

[9] Solving generalized eigenvalue problems by ordinary differential equations on a quantum computer. Proceedings of The Royal Society A: Mathematical, Physical and Engineering Sciences. 2022 Jun 1;478(2262).

[10] Sudiarta IW, Susanto H. New Power Method for Solving Eigenvalue Problems. arXiv.org. 2022 Oct 31;abs/2211.06303.

[11] Lyche T, Muntingh G, Ryan Ø. The QR Algorithm. In Springer, Cham; 2020. p. 255. Available from: https://link.springer.com/chapter/10.1007%2F978-3-030-36468-7_15

[12] Rutishauser H. The Jacobi method for real symmetric matrices. Numerische Mathematik [Internet]. 1966 Nov 1;9(1):1–10. Available from: https://link.springer.com/chapter/10.1007/978-3-662-39778-7_12

[13] Bootstrapped Block Lanczos for large-dimension eigenvalue problems. 2022 Oct 27; Available from: http://arxiv.org/pdf/2210.15763

[14] van den Eshof J. The convergence of Jacobi-Davidson for Hermitian eigenproblems. 2000 Nov 1; Available from: http://dspace.library.uu.nl/handle/1874/2513

[15] Panju M. Iterative methods for computing eigenvalues and eigenvectors. arXiv: Numerical Analysis [Internet]. 2011 May 5; Available from: http://mathreview.uwaterloo.ca/archive/voli/1/panju.pdf

[16] Dongarra J, Sidani M. A Parallel Algorithm for the Non-Symmetric Eigenvalue Problem [Internet]. University of Tennessee; 1991 Sep. Available from: https://www.osti.gov/servlets/purl/10108033

[17] Kressner D, Ma Y, Shao M. A mixed precision LOBPCG algorithm. Numerical Algorithms [Internet]. 2023 Feb 24;1–19. Available from: http://arxiv.org/pdf/2302.12528

[18] Greenbaum A, Trefethen LN. Gmres/Cr and Arnoldi/Lanczos as Matrix Approximation Problems [Internet]. 2018. Available from: https://people.maths.ox.ac.uk/trefethen/publication/PDF/1994_60.pdf

[19] O'Leary DP, Stewart GW, Vandergraft JS. Estimating the largest eigenvalue of a positive definite matrix. Mathematics of Computation [Internet]. 1979 Jan 13;33(148):1289–92. Available from: https://www.cs.umd.edu/~oleary/reprints/j05.pdf

[20] Chen TY, Demmel J. Preconditioning sparse matrices for computing eigenvalues and solving linear systems of equations. 2001 Jan 1;

[21] Čajić E, Stojanović Z, Galić D. Hybrid Approaches for Eigenvalues and Eigenvectors: Neural Networks and Traditional Methods.

[22] Hadap S, Patil M. Quantum Computing in Artificial Intelligence: A Paradigm Shift. International Journal of Advanced Research in Science, Communication and Technology. 2024 Jun 25;530–4.

[23] Shao C. Computing eigenvalues of matrices in a quantum computer. arXiv: Quantum Physics [Internet]. 2019 Dec 17; Available from: https://dblp.uni-trier.de/db/journals/corr/corr1912.html#abs-1912-08015

[24] Saad Y. Projection methods for solving large sparse eigenvalue problems. In Springer, Berlin, Heidelberg; 1983. p. 121–44. Available from: https://link.springer.com/chapter/10.1007/BFb0062098

[25] de Paula LCM. Paralelização e comparação de métodos iterativos na solução de sistemas lineares grandes e esparsos. ForScience [Internet]. 2013 Nov 11;1(1):1–12. Available from: http://www.forscience.ifmg.edu.br/forscience/index.php/forscience/article/download/18/30

[26] A Fast Iterative Algorithm for Near-Diagonal Eigenvalue Problems. SIAM Journal on Matrix Analysis and Applications. 2022 Nov 11;43(4):1573–98.

BIOGRAPHIES OF AUTHORS

Sahib Abdullah Hasan is a researcher in the Department of Mathematics at Azerbaijan Shahid Madani University. His work focuses on numerical methods for eigenvalue problems, with a particular interest in iterative techniques for large-scale and sparse systems. Dr. Hasan has published research on enhancing computational efficiency in solving matrix eigenvalue problems, contributing to both traditional and modern methods in numerical linear algebra.

Mohammad Farhan Ibrahim is a researcher in the Department of Mathematics at Azerbaijan Shahid Madani University. His research interests lie in the field of numerical analysis, particularly in the development of algorithms for solving eigenvalue problems. Ibrahim's work focuses on improving the efficiency and scalability of iterative methods for large, sparse, and non-symmetric matrices, contributing to advancements in computational mathematics and its applications in science and engineering