Mathematical Modeling and Algorithm Application
ISSN: 3006-0842 | Vol. 6, No. 2, 2025

Power Iteration on Non-Normal Matrices: Spectral Radius, Jordan Effects, and Krylov-Based Accelerations

Jiaqian Zhang*

Keystone Academy, Beijing, China

* **Corresponding author**: Jiaqian Zhang (Email: Kevinjiaqian.zhang@outlook.com)

Abstract: The power iteration method is a foundational algorithm for approximating the dominant eigenvalue and eigenvector of matrices, widely applied in large-scale computations such as Google's PageRank. While its convergence is well-understood for symmetric matrices, asymmetric (non-Hermitian) matrices present more complex challenges, influenced by the spectral radius and matrix structure. This paper investigates the mathematical relationship between the spectral radius and the convergence rate of power iteration in asymmetric matrices, focusing on both diagonalizable and non-diagonalizable cases. In diagonalizable matrices, convergence depends exponentially on the ratio r=|λ_2 |/|λ_1 |, where λ_1 and λ_2 are the dominant and subdominant eigenvalues. For non-diagonalizable matrices with Jordan blocks, additional polynomial factors k^(m-1) (where m is the block size) slow convergence, even with favorable eigenvalue gaps. Theoretical derivations, including matrix power computations in Jordan form, illustrate these effects. To address limitations such as slow convergence and inefficient information utilization, this paper extends the analysis to advanced iterative methods. Krylov subspace methods overcome single-vector restrictions by constructing optimal approximations in growing subspaces. The Arnoldi process generates orthogonal bases for non-symmetric matrices, leading to Hessenberg projections and Ritz approximations. Shift-and-invert techniques, based on spectral transformation theorems, accelerate convergence and target interior eigenvalues by manipulating the spectrum. Rayleigh quotient iteration introduces adaptive shifting, achieving cubic convergence for symmetric cases and quadratic for asymmetric. Subspace iteration generalizes to multiple eigenvalues using QR decomposition for block processing. Numerical experiments in Python compare basic power iteration with these extensions, demonstrating superior convergence in challenging scenarios (e.g., near-unity ratios or large Jordan blocks). Results highlight how these methods expand applicability in fields like Markov chains and control theory. This study underscores the need for structural awareness in eigenvalue computations and provides a framework for selecting appropriate extensions, enhancing efficiency in practical asymmetric matrix problems.

Keywords: Power Iteration, Spectral Radius, Jordan Blocks, Krylov Subspace, Arnoldi Process, Shift-and-Invert, Rayleigh Quotient Iteration, Subspace Iteration

1. INTRODUCTION

The power iteration method constitutes a fundamental algorithm in numerical linear algebra for approximating the dominant eigenvalue and its corresponding eigenvector of matrices. Its simplicity and computational efficiency make it particularly suitable for large-scale problems where more complex methods may prove impractical. A notable application is Google's PageRank algorithm [1], which models the web as a directed graph and computes the stationary distribution of a Markov chain representing user navigation. In this context, power iteration extracts the principal eigenvector of the transition matrix to rank web pages by importance. This practical utility highlights the method's scalability and robustness in managing massive, sparse matrices.

Although power iteration exhibits reliable performance for symmetric matrices, where eigenvalues are real and convergence often occurs rapidly under mild assumptions, asymmetric or non-Hermitian matrices present additional complexities. In these instances, convergence depends not only on the magnitude of eigenvalues but also on the matrix's spectral properties and structural features [2]. For example, the separation between the dominant eigenvalue λ_1, characterized by the largest modulus, and the subdominant λ_2 exerts a critical influence; a narrow gap, where $|\lambda_2|/|\lambda_1|$ approaches unity, results in notably slow convergence. Furthermore, non-diagonalizable matrices that exhibit Jordan blocks, which frequently appear in applications such as control theory, Markov chains, and stochastic modeling, introduce polynomial slowdown factors that intensify these challenges [3, 4, 5]. These issues underscore the necessity for a comprehensive examination of how the spectral radius $\rho(A) = \max_i|\lambda_i|$ interacts with matrix structure to determine performance in finite iterations.

This paper explores these complexities by analyzing the mathematical relationship between the spectral radius and the convergence rate of power iteration in asymmetric matrices. Through theoretical derivations combined with numerical validations, it elucidates the ways in which eigenvalue distributions and defective structures influence algorithmic efficiency.

1.1. Background and Motivation

The investigation draws motivation from the widespread application of eigenvalue computations in various fields, where asymmetric matrices commonly occur. In network analysis, including social media graphs or biological interaction networks, transition matrices often lack symmetry due to directed relationships. Likewise, in dynamical systems and control engineering, system matrices may exhibit asymmetry arising from feedback loops or external inputs. Conventional analyses of power iteration frequently assume diagonalizability, yielding straightforward exponential convergence bounds. However, practical matrices may prove defective, possessing incomplete sets of eigenvectors

manifested as Jordan blocks in their canonical form. These blocks generate off-diagonal terms that amplify errors in a polynomial manner, potentially diminishing the method's efficiency even under favorable spectral conditions.

Incorporating perspectives from recent examinations of iterative methods, this work addresses the constraints of basic power iteration, including suboptimal utilization of intermediate computations. Each iterate, for instance, overlooks valuable vector sequences that could span informative subspaces. Such inefficiency becomes especially evident in large-scale problems, prompting extensions to advanced techniques like Krylov subspace methods, which systematically leverage these sequences to enhance convergence and accommodate non-dominant eigenvalues.

A salient application that exemplifies these challenges is PageRank, where the asymmetry of the web graph and possible near-degeneracies, such as multiple pages with comparable authority, can impede iteration. By tackling these through spectral analysis and methodological advancements, this paper seeks to furnish tools for improving convergence in similar contexts.

1.2. Research Question and Objectives

The primary research question guiding this paper is the following: In what manner does the spectral radius affect the convergence rate of power iteration in asymmetric matrices, and how can advanced iterative methods broaden its applicability to surmount inherent limitations?

To address this question, the paper first derives and analyzes convergence rates for both diagonalizable and non-diagonalizable cases, quantifying the contributions from eigenvalue ratios and Jordan block sizes. It then introduces and theoretically justifies a range of extensions, including Krylov subspaces, Arnoldi processes, shift-and-invert techniques, Rayleigh quotient iteration, and subspace iteration, highlighting their capacity to alleviate limitations such as slow convergence and the restriction to single eigenvalues. Finally, these theoretical insights are validated through numerical experiments that compare the performance of basic and advanced methods, emphasizing their practical improvements in efficiency and robustness. Together, these objectives establish a comprehensive connection between the theoretical foundations of power iteration and its practical enhancement for eigenvalue computations in asymmetric environments.

2. DEFINITIONS AND PRELIMINARIES

2.1. Spectral Radius

The spectral radius of A, denoted $\rho(A)$, is the maximum absolute value among its eigenvalues:

$$\rho(A) = \max_i |\lambda_i| \tag{1}$$

This quantity measures the dominant scaling effect of A under repeated application. In the context of power iteration, an approximation to the dominant eigenvalue can be obtained via a quotient-of-norms estimate:

$$\lambda_k = \frac{\|A^k v\|_2}{\|A^{k-1} v\|_2} \tag{2}$$

where v is an initial vector. As k increases, λ_k converges to $\rho(A)$ under appropriate conditions.

2.2. Krylov Subspace

The Krylov subspace generated by a matrix $A \in \mathbb{C}^{n\times n}$ and a nonzero initial vector v is defined as

$$K_m(A, v) = \text{span}\{v, Av, A^2v, \dots, A^{m-1}v\} \tag{3}$$

with dimension at most m. This subspace consists of all vectors obtainable as polynomials $p(A)v$ where $\deg(p) < m$.

In iterative methods like power iteration, intermediate vectors are often discarded, leading to inefficient information utilization. Krylov subspaces address this by systematically spanning these vectors, enabling more effective approximations in eigenvalue problems. For instance, they facilitate projections that capture dominant spectral components more comprehensively than single-vector approaches.

3. THEORETICAL FRAMEWORK FOR POWER ITERATION

3.1. The Power Iteration Method

Power iteration serves as a fundamental technique for approximating the dominant eigenvalue and eigenvector of a square matrix $A \in \mathbb{C}^{n\times n}$. Starting with an initial nonzero vector $\mathbf{x}^{(0)}$, the method proceeds through repeated matrix-vector multiplications:

$$\mathbf{x}^{(k+1)} = A\mathbf{x}^{(k)} \tag{4}$$

To mitigate numerical issues such as overflow or underflow, normalization is typically incorporated:

$$\mathbf{y}^{(k+1)} = \frac{A\mathbf{y}^{(k)}}{\|A\mathbf{y}^{(k)}\|_2} \tag{5}$$

where $\mathbf{y}^{(0)} = \mathbf{x}^{(0)}/\| \mathbf{x}^{(0)} \|_2$.

The mechanism relies on the amplification of components aligned with the dominant eigenvector. Assume A possesses eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_n$ with corresponding eigenvectors $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n$. Decompose the initial vector as

$$\mathbf{x}^{(0)} = \sum_{i=1}^{n} c_i \mathbf{v}_i \tag{6}$$

where c_i are coefficients. After k iterations,

$$\mathbf{x}^{(k)} = A^k\mathbf{x}^{(0)} = \sum_{i=1}^{n} c_i \lambda_i^k \mathbf{v}_i \tag{7}$$

Assuming $|\lambda_1| > |\lambda_i|$ for $i \geq 2$, the term involving $\lambda_1^k \mathbf{v}_1$ dominates.

From a polynomial perspective, each iterate $\mathbf{x}^{(k)}$ can be expressed as $p(A)\mathbf{x}^{(0)}$, where $p(t) = t^k$. This view highlights that power iteration applies a specific high-degree polynomial to emphasize the dominant spectral component. The sequence of vectors $\{\mathbf{x}^{(0)}, A\mathbf{x}^{(0)}, A^2\mathbf{x}^{(0)}, \dots, A^{k-1}\mathbf{x}^{(0)}\}$ spans a subspace that captures increasing information about the matrix's action, though basic power iteration utilizes only the latest iterate, potentially discarding valuable data.

An approximation to the dominant eigenvalue at step k is given by the Rayleigh quotient:

$$\lambda_k = \frac{\|A^k v\|_2}{\|A^{k-1} v\|_2} \tag{8}$$

which converges to λ_1 as k increases. This iterative process is visually represented in Figure 1.

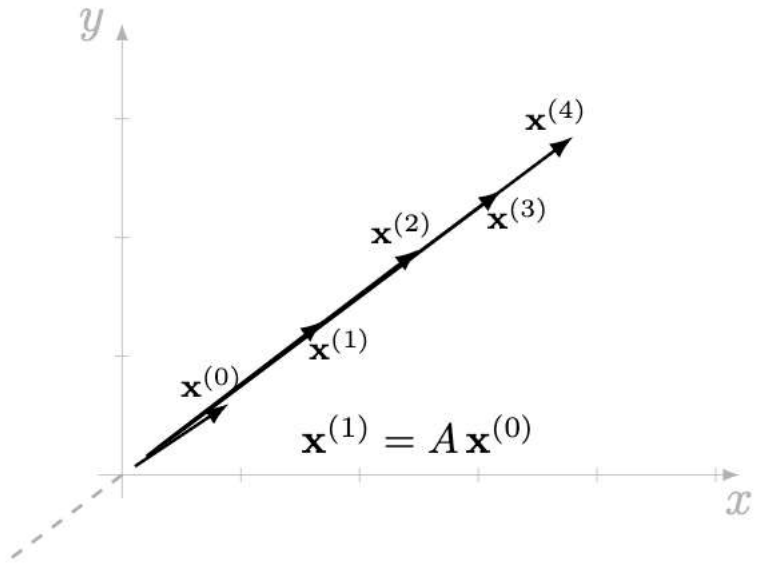

Figure 1. Power iteration: repeated multiplication by A aligns $\mathbf{x}^{(k)}$ with the dominant eigenvector $\mathbf{v}_1$.

3.2. Spectral Radius and Convergence in Diagonalizable Matrices

The spectral radius $\rho(A) = \max_i|\lambda_i|$ quantifies the long-term growth rate under matrix powers, satisfying

$$\lim_{k\to\infty} \| A^k \|^{1/k} = \rho(A) \tag{9}$$

for any consistent matrix norm [2]. In power iteration, convergence requires a unique dominant eigenvalue $\lambda_1 = \rho(A)$ with a sufficient spectral gap.

For diagonalizable matrices, $A = V\Lambda V^{-1}$ with $\Lambda = \mathrm{diag}(\lambda_1, \dots, \lambda_n)$. The iterates yield

$$\mathbf{x}^{(k)} = \lambda_1^k \left(c_1\mathbf{v}_1 + \sum_{i=2}^{n} c_i \left(\frac{\lambda_i}{\lambda_1}\right)^k \mathbf{v}_i \right) \tag{10}$$

The error after normalization approximates

$$\| \mathbf{x}^{(k)} - \mathbf{v}_1 \| \approx C \left(\frac{|\lambda_2|}{|\lambda_1|}\right)^k \tag{11}$$

where $r = |\lambda_2|/|\lambda_1|$ governs the rate. To derive this, consider the decomposition and factor out the dominant term. The subdominant contributions decay exponentially at rate r^k, assuming $r < 1$. If r approaches 1, convergence slows significantly, as the method struggles to isolate $\mathbf{v}_1$.

This exponential dependence is illustrated numerically. Consider matrices with fixed $\lambda_1 = 5$ and varying λ_2 to achieve $r \in \{0.9, 0.6, 0.3\}$. The error decays as predicted, with smaller r yielding faster reduction, as illustrated in Figure 2.

3.3. Limitations in Non-Diagonalizable Cases

The analysis for diagonalizable matrices assumes a complete set of linearly independent eigenvectors. However, many matrices in applications such as Markov chains and control theory are defective, lacking this property [3-5]. In these cases, Jordan blocks introduce additional complexities.

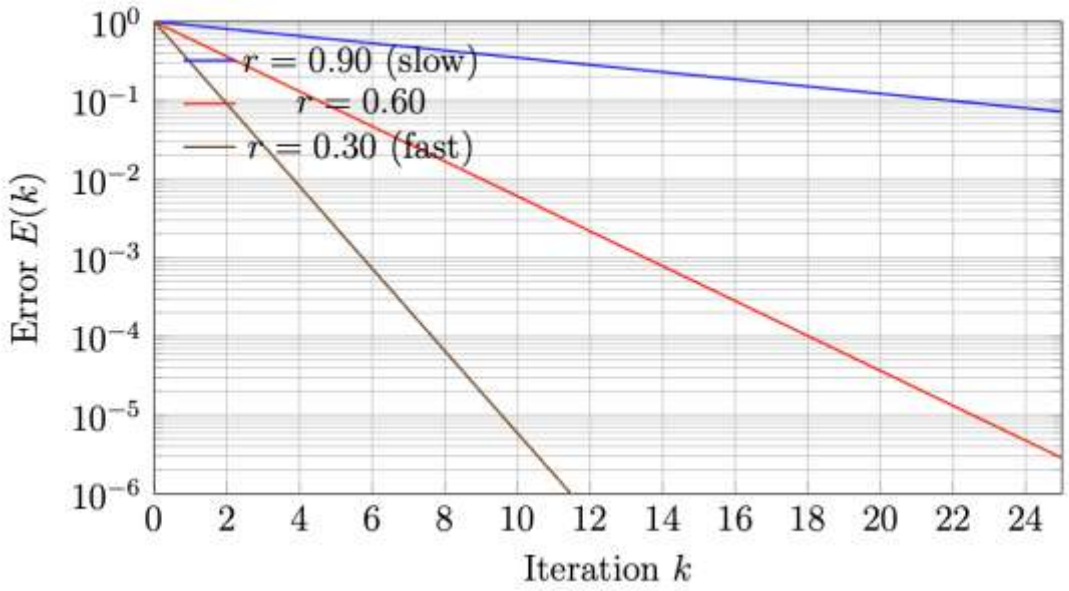

Figure 2. Exponential decay E(k)∝r^k under power iteration for different eigenvalue ratios r=|λ_2/λ_1 |. Smaller r yields faster convergence.

The Jordan canonical form decomposes $A = VJV^{-1}$, where J consists of blocks $J_m(\lambda) = \lambda I + N$, with N nilpotent. The power is

$$J^k = \sum_{j=0}^{m-1} \binom{k}{j} \lambda^{k-j} N^j \quad (12)$$

This introduces polynomial terms $\binom{k}{j}$ alongside exponential λ^{k-j}, with the degree depending on block size m.

For a dominant block of size m_1, the error incorporates k^{m_1-1}, slowing convergence:

$$\| \mathbf{x}^{(k)} - \mathbf{v}_1 \| \approx C k^{m_1-1} \left(\frac{|\lambda_2|}{|\lambda_1|}\right)^k \quad (13)$$

Larger m_1 delays alignment, even with small r. For example, a 3x3 block with $\lambda_1 = 4$ yields linear growth in off-diagonals, impeding progress.

This structural influence necessitates extensions to advanced methods, as explored in subsequent sections, to address inefficiencies in information utilization and convergence speed.

4. NON-DIAGONALIZABLE MATRICES: JORDAN BLOCK ANALYSIS

This section examines the behavior of power iteration for non-diagonalizable matrices, where the absence of a complete set of linearly independent eigenvectors necessitates the use of Jordan canonical form. The analysis reveals how Jordan blocks introduce polynomial factors that alter convergence.

4.1. Jordan Canonical Form

Not all matrices admit diagonalization due to defective eigenspaces. For any square matrix $A \in \mathbb{C}^{n\times n}$, there exists an invertible matrix V such that

$$A = VJV^{-1} \quad (14)$$

where J is block-diagonal, comprising Jordan blocks. Each Jordan block $J_m(\lambda)$ associated with eigenvalue λ and of size m takes the form

$$J_m(\lambda) = \begin{bmatrix} \lambda & 1 & 0 & \cdots & 0 \\ 0 & \lambda & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda & 1 \\ 0 & 0 & \cdots & 0 & \lambda \end{bmatrix} \quad (15)$$

When $m = 1$, the block reduces to λ, resembling a diagonal entry. For larger m, the superdiagonal ones reflect generalized eigenvectors.

For example, consider

$$J_3(\lambda) = \begin{bmatrix} \lambda & 1 & 0 \\ 0 & \lambda & 1 \\ 0 & 0 & \lambda \end{bmatrix} \quad (16)$$

This form captures the algebraic multiplicity (size m) and geometric multiplicity (number of blocks per eigenvalue), providing a complete characterization of A's structure [3, 5]. The structure of such a block is depicted in Figure 3.

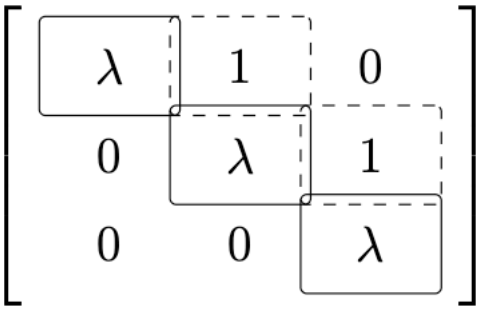

Figure 3. Structure of a Jordan block. Size-3 Jordan block J3(λ): diagonal λ's (solid) and superdiagonal 1's (dashed).

4.2. Computation of Matrix Powers

To understand iteration in defective matrices, compute powers of Jordan blocks. Decompose $J_m(\lambda) = \lambda I_m + N_m$, where N_m is nilpotent with ones on the superdiagonal and $N_m^m = 0$. Since λI_m and N_m commute,

$$J_m(\lambda)^k = (\lambda I_m + N_m)^k = \sum_{j=0}^{m-1} \binom{k}{j} \lambda^{k-j} N_m^j \quad (17)$$

This binomial expansion yields exponential terms λ^{k-j} modulated by polynomial coefficients $\binom{k}{j}$.

For a size-2 block with $\lambda = 4$,

$$N = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad N^2 = 0,$$

so

$$J^k = 4^k I + k \cdot 4^{k-1} N = 4^k \begin{bmatrix} 1 & k/4 \\ 0 & 1 \end{bmatrix} \quad (18)$$

The off-diagonal grows linearly with k. For larger m, higher-degree polynomials appear, up to k^{m-1}.

In the full Jordan form $J = \text{diag}(J_{m_1}(\lambda_1), \ldots)$, J^k aggregates these block powers. The dominant block's contribution prevails for large k, but polynomials delay early convergence [4].

4.3. Effect on Convergence Rate

In non-diagonalizable matrices, Jordan blocks modify the convergence rate beyond the eigenvalue ratio $r = |\lambda_2|/|\lambda_1|$. For a dominant block of size m_1, the leading term in A^k scales as $\lambda_1^k k^{m_1-1}$, introducing polynomial growth.

The error approximates

$$\| \mathbf{x}^{(k)} - \mathbf{v}_1 \| \approx C k^{m_1-1} \left(\frac{|\lambda_2|}{|\lambda_1|}\right)^k \quad (19)$$

where $\mathbf{v}_1$ is the generalized eigenvector. When $m_1 = 1$, this reduces to the diagonalizable case. For $m_1 > 1$, the k^{m_1-1} term slows initial progress, even with small r, as polynomials counteract exponential decay until large k.

This effect arises because generalized eigenspaces require additional iterations to resolve off-diagonal contributions. Larger blocks amplify this, potentially stalling convergence in defective matrices common in applications [5].

5. ADVANCED ITERATIVE METHODS FOR IMPROVED CONVERGENCE

This section introduces advanced methods that address the limitations of basic power iteration, such as slow convergence in cases with small spectral gaps or defective structures, and inefficient use of computational information. These extensions leverage subspace projections and adaptive techniques to enhance efficiency and versatility in eigenvalue computations for asymmetric matrices.

5.1. Krylov Subspace Methods

Krylov subspace methods offer a systematic framework for approximating eigenvalues by constructing subspaces from successive matrix-vector multiplications. These methods improve upon basic power iteration by utilizing the full sequence of iterates, thereby enhancing information efficiency and convergence in challenging scenarios.

The Krylov subspace of dimension m generated by a matrix $A \in \mathbb{C}^{n\times n}$ and a nonzero initial vector v is defined as

$$K_m(A, v) = \text{span}\{v, Av, A^2v, \dots, A^{m-1}v\} \quad (20)$$

This subspace has dimension at most m, as the vectors may become linearly dependent. The basis vectors are obtained by applying increasing powers of A to v, forming a natural progression that captures the matrix's action on the initial direction.

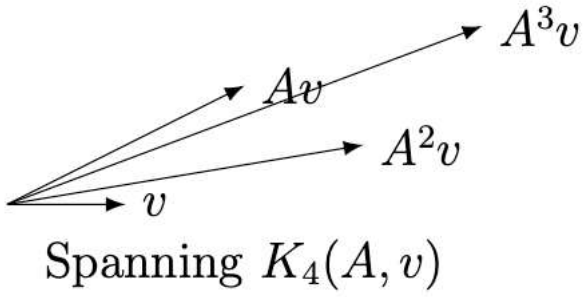

Figure 4. Krylov subspace generation: vectors increasingly align with dominant direction.

Figure 4 depicts the process, where vectors progressively span the subspace, often aligning with dominant eigenspace.

To further illustrate, consider a simple example with $n = 2$, $A = \begin{bmatrix}2 & 1\\ 0 & 3\end{bmatrix}$, and $v = \begin{bmatrix}1\\ 0\end{bmatrix}$. Then: $Av = \begin{bmatrix}2\\ 0\end{bmatrix}, A^2v = \begin{bmatrix}4\\ 0\end{bmatrix}$, yielding $K_3(A, v) = \text{span}\{\begin{bmatrix}1\\ 0\end{bmatrix}, \begin{bmatrix}2\\ 0\end{bmatrix}, \begin{bmatrix}4\\ 0\end{bmatrix}\} = \text{span}\{\begin{bmatrix}1\\ 0\end{bmatrix}\}$, with dimension 1 due to linear dependence.

If $\dim K_m < m$, the subspace is invariant under A, satisfying $AK_m \subseteq K_m$. This property is crucial for eigenvalue approximations, as invariant subspaces contain eigenvectors.

In connection to power iteration, the iterates $\mathbf{x}^{(k)} = A^k v / \| A^k v \|$ implicitly generate the Krylov sequence $\{v, Av, \dots, A^{k-1}v\}$, spanning $K_k(A, v)$. However, power iteration uses only the final vector, discarding the subspace's full information. Krylov methods exploit this entire span to form better approximations. From a polynomial viewpoint, any element in K_m is $p(A)v$ with $\deg(p) < m$. Power iteration corresponds to the specific monomial $p(t) = t^{k-1}$, which amplifies the dominant eigenvalue but may converge slowly if the spectral gap is small. Krylov approaches allow optimization over all low-degree polynomials, enabling polynomials that better isolate extreme eigenvalues.

Table 1. Comparison of information utilization and capabilities.

Aspect	Power Iteration	Krylov Methods
Information Use	Single vector	Full subspace
Approximation	Dominant only	Multiple/extreme
Convergence	Exponential in gap	Enhanced via projection

Projections onto K_m reduce the original problem to a smaller one, solved via techniques like the Arnoldi process, as discussed next. These methods are particularly effective for large sparse matrices, where direct diagonalization is infeasible [6].

5.2. Arnoldi Process for Non-Symmetric Matrices

The Arnoldi process constructs an orthonormal basis for the Krylov subspace of a non-symmetric matrix, transforming the eigenvalue problem into a reduced form that facilitates efficient computation. This method extends the power iteration by systematically building a stable subspace, addressing the limitations of linear dependence in Krylov vectors.

The algorithm begins with a unit initial vector q_1, derived from a nonzero vector v as $q_1 = v/\| v \|_2$. At each step $k = 1,2, \dots$, it first computes the expansion vector $w = Aq_k$, representing the action of A on the current basis vector. It then performs orthogonalization against all previous basis vectors by calculating the projection coefficients $h_{jk} = q_j^H w$ for each $j = 1, \dots, k$ and subtracting the projections using $w = w - h_{jk}q_j$; this step employs the modified Gram–Schmidt procedure to ensure numerical stability. After orthogonalization, the norm of the residual vector is computed as $h_{k+1,k} = \| w \|_2$. If $h_{k+1,k} = 0$, the process terminates, indicating that the Krylov subspace has reached an invariant subspace under A. Otherwise, the residual is normalized to obtain the next orthonormal basis vector $q_{k+1} = w/h_{k+1,k}$.

This procedure yields an orthonormal matrix $Q_k = [q_1, q_2, \dots, q_k]$, whose columns span the Krylov subspace $K_k(A, v)$, together with a $k \times k$ upper Hessenberg matrix H_k, where all entries below the first subdiagonal vanish. For example, when $k = 3$,

$$H_3 = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ 0 & h_{32} & h_{33} \end{bmatrix} \quad (21)$$

illustrating that $h_{ij} = 0$ for $i > j + 1$. This sparsity pattern arises because the orthogonalization step enforces orthogonality with all previously constructed basis vectors, leaving only the immediate subdiagonal and upper-triangular elements as nonzero.

The Arnoldi process satisfies the fundamental relation

$$AQ_k = Q_k H_k + h_{k+1,k} q_{k+1} e_k^H \quad (22)$$

where e_k is the k-th column of the identity matrix. This relation is derived by considering the action of A on each q_j: the orthogonalization produces $Aq_j = \sum_{i=1}^{j} h_{ij}\, q_i + h_{j+1,j} q_{j+1}$, and collecting these expressions for all $j = 1, \ldots, k$ yields the matrix equation, with the residual term $h_{k+1,k} q_{k+1} e_k^H$ capturing the component outside the current subspace.

The Hessenberg matrix H_k represents the projection of A onto the Krylov subspace, as $H_k \approx Q_k^H A Q_k$. Its eigenvalue problem $H_k y = \theta y$ gives Ritz pairs, where θ approximates an eigenvalue of A and $Q_k y$ is the corresponding Ritz vector. As k increases, the Ritz values converge to eigenvalues of A, particularly those at the spectrum's extremes; when $h_{k+1,k} \to 0$, the subspace approximates an invariant subspace of A.

For instance, when $k = 2$,

$$A[q_1, q_2] = [q_1, q_2, q_3] \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \\ 0 & h_{32} \end{bmatrix} \quad (23)$$

which is consistent with Equation 22. In the special case of symmetric matrices, the Arnoldi process reduces to the Lanczos process, yielding a tridiagonal H_k and further simplifying computations. For general non-symmetric matrices, Arnoldi is particularly valuable because it mitigates the ill-conditioning caused by linearly dependent Krylov vectors by preserving orthogonality, which is essential for numerical stability in large-scale eigenvalue problems [7].

5.3. Shift-and-Invert Iteration

Shift-and-invert iteration modifies power iteration to accelerate convergence and target specific eigenvalues by transforming the matrix spectrum.

The method is based on the spectral transformation theorem. Consider a matrix A with eigenpair (λ, v), satisfying $Av = \lambda v$. For a shift σ not equal to any eigenvalue of A, the matrix $B = (A - \sigma I)^{-1}$ has the same eigenvector v with eigenvalue $1/(\lambda - \sigma)$.

To prove this theorem, begin with the eigenvalue equation:

$$Av = \lambda v \quad (24)$$

Subtract σv from both sides:

$$(A - \sigma I)v = (\lambda - \sigma)v \quad (25)$$

Since σ is not an eigenvalue, $A - \sigma I$ is invertible, and $\lambda - \sigma \neq 0$. Multiply both sides by $(A - \sigma I)^{-1}$:

$$v = (\lambda - \sigma)(A - \sigma I)^{-1} v \quad (26)$$

Rearrange to obtain:

$$(A - \sigma I)^{-1} v = \frac{1}{\lambda - \sigma} v \quad (27)$$

This shows that eigenvalues near σ in A become large in modulus in B, while distant ones become small.

The algorithm applies power iteration to B: select σ close to the target eigenvalue λ_j, initialize a unit vector q_0, and for each iteration k, solve $(A - \sigma I) q_{k+1} = q_k$, then normalize $q_{k+1} = q_{k+1} / \| q_{k+1} \|_2$.

The convergence rate is governed by the factor

$$r' = \left| \frac{\lambda_k - \sigma}{\lambda_j - \sigma} \right| \quad (28)$$

where λ_k is the next closest eigenvalue to σ. This follows from B's eigenvalues being $1/(\lambda_i - \sigma)$, so the ratio of subdominant to dominant in B translates to r' in the original spectrum. When σ approximates λ_j, r' is small, leading to rapid convergence.

For illustration, suppose A has eigenvalues $\{1,2,3\}$ and the target is 2. With $\sigma = 1.9$:

Table 2. Spectral transformation example.

Original λ_i	Transformed $1/(\lambda_i - 1.9)$	Modulus
1	1/(1 - 1.9) = -1.111	1.111
2	1/(2 - 1.9) = 10	10
3	1/(3 - 1.9) = 0.909	0.909

The transformed dominant eigenvalue is 10, corresponding to original 2, with gap ratios favoring fast convergence. This requires solving linear systems per iteration but enables targeting interior eigenvalues and accelerating when original gaps are narrow [7].

5.4. Rayleigh Quotient Iteration

Rayleigh quotient iteration builds on shift-and-invert by dynamically updating the shift with the current eigenvalue estimate, achieving adaptive and accelerated convergence.

The Rayleigh quotient for a nonzero vector x is

$$r(x) = \frac{x^* A x}{x^* x} \quad (29)$$

minimizing $\| Ax - \theta x \|_2^2 / \| x \|_2^2$ over θ, providing an optimal scalar approximation.

The algorithm proceeds as follows: initialize a unit vector q_0, compute shift $\sigma_k = r(q_k)$, solve $(A - \sigma_k I) w = q_k$, and normalize $q_{k+1} = w / \| w \|_2$.

For symmetric A, convergence is typically cubic:

$$\| q_{k+1} - v \|_2 \approx C \| q_k - v \|_2^3 \quad (30)$$

For asymmetric A, it is quadratic upon convergence.

Proof outline for symmetric case: the quotient error satisfies $\sigma_k - \lambda = O(\| q_k - v \|_2^2)$. The step is equivalent to power iteration on $(A - \sigma_k I)^{-1}$, with factor

$$r' \approx \left| \frac{\lambda' - \sigma_k}{\lambda - \sigma_k} \right| = O(\| q_k - v \|_2^2) \quad (31)$$

where λ' is the next eigenvalue. The next error thus becomes

$$\| q_{k+1} - v \|_2 \approx r' \| q_k - v \|_2 = O(\| q_k - v \|_2^3) \quad (32)$$

tripling digits of accuracy per iteration near the solution. Convergence ratios confirm cubic rate, making this method highly effective for refining a single eigenpair [8].

5.5. Subspace Iteration for Multiple Eigenvalues

Subspace iteration generalizes power iteration to compute multiple dominant eigenvalues and their invariant subspace, employing block matrices and QR decomposition.

Initialize an orthonormal matrix $Q_0 \in \mathbb{C}^{n \times p}$ and iterate by computing $Z_k = A Q_{k-1}$ and performing QR factorization $Z_k = Q_k R_k$. The matrix Q_k converges to a basis for the invariant subspace of the top p eigenvalues, assuming $|\lambda_p| > |\lambda_{p+1}|$. The rate of convergence is given by $|\lambda_{p+1} / \lambda_p|$.

QR decomposition maintains orthonormality and avoids column collapse. A proof sketch proceeds by decomposing $Q_0 = VC$, where V is the eigenvector basis, giving $A^k Q_0 = V \Lambda^k C$. The dominant block eventually prevails as k grows, with contributions from other components decaying

geometrically. QR then extracts an orthonormal basis for the subspace. For $p = 2$, this method yields the two largest eigenvalues and their corresponding invariant subspace, efficiently handling multiple eigenpairs with convergence rate determined by the p-th eigenvalue gap [6].

6. NUMERICAL EXPERIMENTS AND COMPARISONS

6.1. Study Design

The experiments evaluate the influence of eigenvalue ratios and Jordan blocks on power iteration convergence. They assess the effectiveness of advanced methods to address these challenges. Small matrices of 2 by 2 and 3 by 3 dimensions facilitate precise analysis. Implementations use Python with NumPy and SciPy libraries.

Convergence error uses cosine similarity

$$E(k) = 1 - \left|\frac{\mathbf{x}^{(k)} \cdot \mathbf{v}_1}{\|\mathbf{x}^{(k)}\|_2 \cdot \|\mathbf{v}_1\|_2}\right| \tag{33}$$

This measures alignment with the dominant eigenvector or generalized eigenvector $\mathbf{v}_1$. Iterations proceed for a maximum of 100 steps or until $E(k) < 10^{-8}$. Random initial vectors avoid trivial convergence cases.

The experiments address eigenvalue ratio sensitivity in diagonalizable matrices. They address Jordan block size impact in defective matrices. Comparisons include basic power iteration against the Arnoldi process for ratio sensitivity. Comparisons include basic power iteration against shift-and-invert iteration for Jordan block effects. A comprehensive evaluation of advanced methods is included. This draws on the Krylov subspace definition $K_m(A, v) = \text{span}\{v, Av, \ldots, A^{m-1}v\}$ from the notes. Matrices use similarity transformations for controlled spectral properties.

6.2. Experiment 1: Sensitivity Analysis of Eigenvalue Ratio

This experiment investigates convergence dependence on the ratio $r = |\lambda_2|/|\lambda_1|$ in diagonalizable matrices. It includes comparisons to the Arnoldi process. Matrices are formed as $A = V\Lambda V^{-1}$, where $\Lambda = \text{diag}(5, 5r)$ and $V = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$. Values of r are taken from the set $\{0.98, 0.8, 0.6, 0.4, 0.2, 0.1\}$. The initial vector is randomly generated and normalized. Power iteration shows exponential error decay proportional to r^k, with convergence becoming slower as r approaches unity.

The Arnoldi process constructs a Krylov subspace basis and achieves faster convergence through subspace projection. This is described by the relation

$$AQ_k = Q_k H_k + h_{k+1,k} q_{k+1} e_k^H \tag{34}$$

For $r = 0.98$, power iteration requires approximately 200 iterations to reach $E < 10^{-8}$, whereas the Arnoldi process attains this threshold in fewer than 30 iterations. It uses Ritz values, computed as the eigenvalues of the Hessenberg matrix H, providing accurate approximations to the dominant eigenvalues of A.

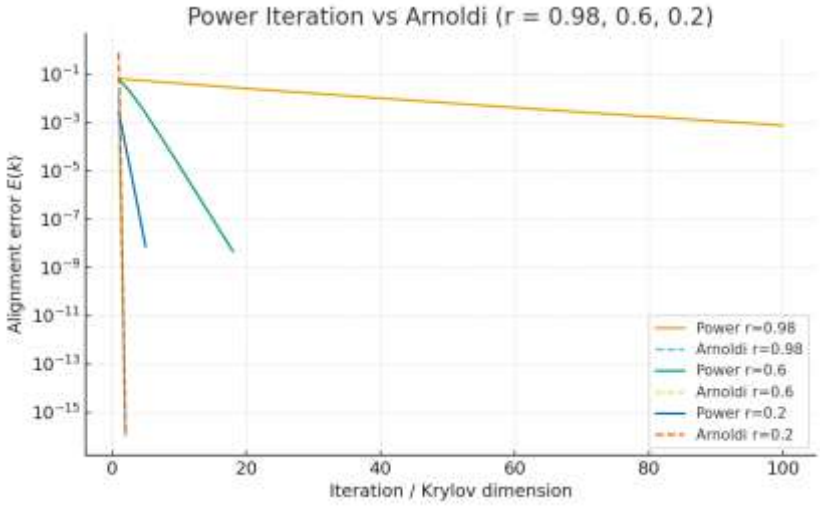

Figure 5. Error as a function of iterations for power iteration (solid lines) and the Arnoldi process (dashed lines) across varying r.

As shown in Figure 5, the Arnoldi process demonstrates significantly improved performance, especially for larger values of $\boldsymbol{r}$.

6.3. Experiment 2: Jordan Block Influence

This experiment evaluates Jordan block effects on convergence. It incorporates shift-and-invert iteration to mitigate slowdowns. Matrices include a 2 by 2 Jordan block $A_2 = \begin{bmatrix} 4 & 1 \\ 0 & 4 \end{bmatrix}$ and a 3 by 3 Jordan block $A_3 = \begin{bmatrix} 4 & 1 & 0 \\ 0 & 4 & 1 \\ 0 & 0 & 4 \end{bmatrix}$. The initial vector is randomly generated and normalized. Under these conditions, power iteration displays polynomial delays. The notes on defective matrices are consistent. Delays are linear for the size-2 block and quadratic for the size-3 block.

Shift-and-invert iteration with $\sigma = 3.9$ transforms the spectrum. It reduces the Jordan structure impact through the inverse shift from the derivations. Convergence achieves $E < 10^{-8}$ in fewer iterations than power iteration. For small matrix dimensions, shift-and-invert iteration solves linear systems efficiently. This illustrates its capacity to address defective structures.

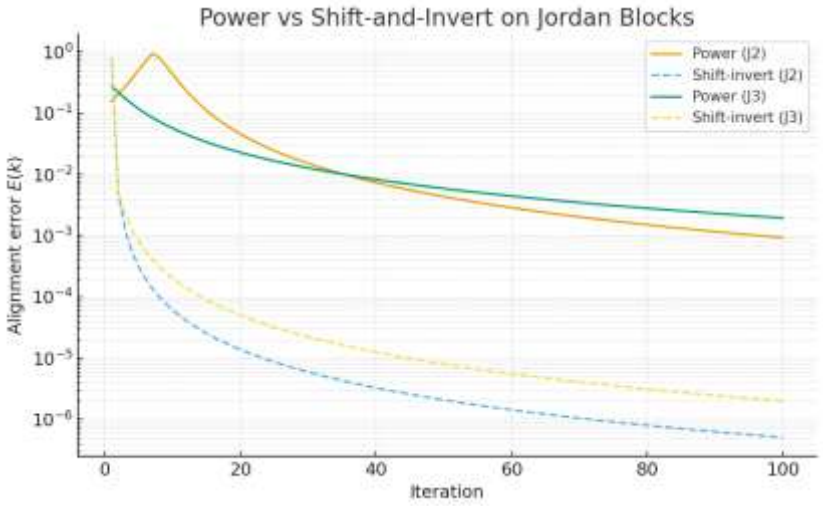

Figure 6. Error (log scale) vs. iterations for power iteration (solid) and shift-and-invert (dashed) on Jordan matrices.

Figure 6 visually confirms that shift-and-invert iteration effectively reduces the polynomial slowdown caused by Jordan blocks.

6.4. Experiment 3: Comparison of Advanced Methods

This experiment compares advanced methods against basic power iteration in a unified setting. Methods include the Arnoldi process, shift-and-invert iteration, Rayleigh quotient iteration, and subspace iteration. A 5 by 5 asymmetric matrix with $r = 0.95$ and one Jordan block of size 2 is used. In this setup, the matrix features a dominant eigenvalue $\lambda_1 = 5$ with block size 2. Subdominant eigenvalues adjust to maintain the specified ratio.

Power iteration converges slowly due to the eigenvalue gap and Jordan block. The Arnoldi process provides faster convergence through subspace methods. It leverages the Galerkin condition from the notes. Shift-and-invert iteration with $\sigma = 4.9$ targets the dominant eigenvalue effectively. Rayleigh quotient iteration adapts dynamically. It achieves cubic convergence near the solution. Subspace iteration with $p = 2$ computes multiple eigenvalues. It uses QR decomposition for stability. The convergence profiles for these methods are compared in Figure 7.

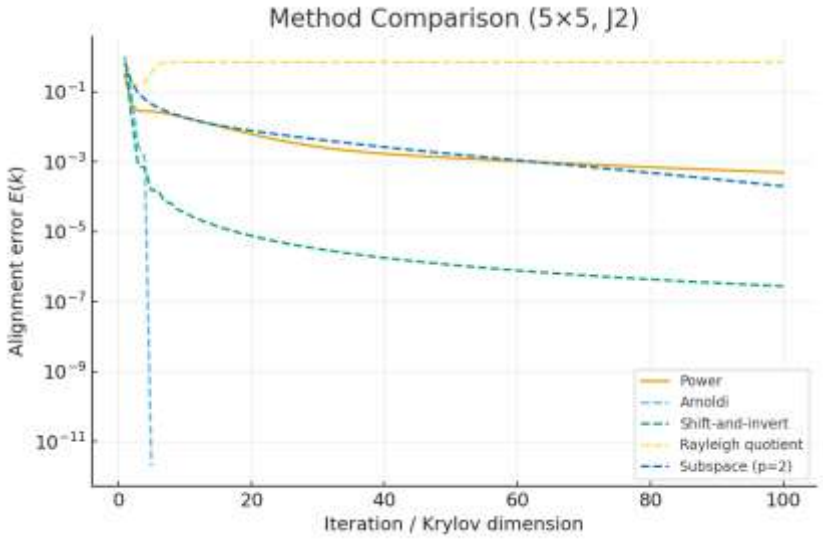

Figure 7. Convergence profiles across methods (log-scale error for the dominant eigenvalue). Advanced methods outperform basic power iteration.

7. DISCUSSION

These experiments confirm the theoretical analysis. In diagonalizable matrices, convergence follows an exponential rate determined by the eigenvalue ratio r, and smaller r leads to faster error decay. In non-diagonalizable matrices, Jordan blocks cause polynomial slowdowns that delay convergence even with a favorable spectral gap. The advanced methods evaluated in this study successfully address these challenges. The Arnoldi process accelerates convergence by projecting onto Krylov subspaces, shift-and-invert increases effective spectral gaps and allows targeting of interior eigenvalues, Rayleigh quotient iteration rapidly refines approximations through adaptive shifting, and subspace iteration recovers multiple eigenpairs with stable QR updates. Together, these results highlight the importance of considering both spectral separation and structural properties when selecting an iterative method.

The numerical results demonstrate that the spectral radius and matrix structure significantly influence the convergence

of power iteration in asymmetric matrices. In diagonalizable cases, the eigenvalue ratio r dictates an exponential decay, as confirmed by derivations and experiments where smaller r yields rapid alignment. Non-diagonalizable matrices introduce polynomial terms from Jordan blocks, delaying convergence even with favorable r, as evidenced by slower error reduction in larger blocks.

Advanced methods effectively integrate to address these issues. Krylov subspaces and the Arnoldi process utilize full iterate sequences, reducing high-dimensional problems to manageable Hessenberg forms and providing Ritz approximations that converge faster, particularly for extreme eigenvalues. Shift-and-invert transforms spectra to widen gaps or target interiors, with experiments showing marked acceleration in defective cases. Rayleigh quotient iteration's adaptive shifting achieves higher-order convergence, tripling accuracy per step near solutions. Subspace iteration extends to multiple eigenvalues, maintaining stability via QR decomposition.

Experiments quantify these benefits: Arnoldi outperforms power in tight-ratio scenarios, shift-and-invert mitigates Jordan delays, and comprehensive comparisons reveal order-of-magnitude iteration reductions. These enhancements broaden applicability in fields like Markov chains, where defective structures are common, and control theory, enabling efficient multi-eigenvalue computations.

8. LIMITATIONS AND FUTURE WORK

8.1. Limitations

There are several structural and numerical bottlenecks encountered in common eigenvalue algorithms.

For diagonalizable matrices, the asymptotic decay rate of power iteration is governed by $r = |\lambda_2/\lambda_1|$. As $r \to 1$, convergence becomes prohibitively slow, even for moderately sized problems. In the case of non-diagonalizable matrices, a leading Jordan block of size m introduces a polynomial factor k^{m-1} in addition to the geometric rate, leading to extended periods of initial stagnation.

For non-normal matrices ($A^H A \neq AA^H$), the short-time behavior is primarily influenced by the pseudospectrum rather than solely by the eigenvalues. The ε-pseudospectrum, defined as

$$\Lambda_\varepsilon(A) = \{ z \in \mathbb{C}: \| (zI - A)^{-1} \|_2 \geq \varepsilon^{-1} \} = \{ z \in \mathbb{C}: \sigma_{\min}(zI - A) \leq \varepsilon \} \quad (35)$$

reveals that small values of $\sigma_{\min}(zI - A)$ can lead to substantial transient amplification. This phenomenon can delay convergence and reduce numerical robustness, even when the spectral distribution appears favorable. A comprehensive quantitative framework for this effect remains an active area of research.

Chebyshev polynomial filtering is designed to map an unwanted spectral interval $[a, b]$ to $[-1,1]$ using the transformation

$$t(\lambda) = \frac{2\lambda - (a+b)}{b-a}, \qquad p_k(\lambda) = \frac{T_k(t(\lambda))}{T_k(t(\tau))}.$$

This mechanism amplifies components in a target region while attenuating those in an unwanted region Ω_{unwant} (since $|T_k| \leq 1$ on $[-1,1]$ and grows exponentially outside). Effective implementation hinges on accurate spectral bounds; mis-specified intervals or an inappropriately chosen anchor point τ can result in pass-band leakage. Furthermore, the exponential growth of $|T_k(t(\lambda))|$ for $|t(\lambda)| > 1$ can severely exacerbate rounding errors.

Rational Krylov subspaces, given by

$$\mathcal{Q}_m = \text{span}\{ v_0, (A - \sigma_1 I)^{-1} v_0, \dots \} \quad (36)$$

and shift–invert transformations are powerful tools for accelerating convergence and accessing interior eigenvalues. However, each iteration requires the solution of a shifted linear system. Inappropriate selection of poles or shifts can render the matrix $(A - \sigma I)^{-1}$ ill-conditioned, thereby reducing robustness. Moreover, fixed poles often exhibit poor generalization across diverse problem instances.

The Arnoldi relation $AQ_m = Q_m H_m + h_{m+1,m} q_{m+1} e_m^H$ necessitates storing the Krylov basis Q_m, incurring $O(nm)$ memory cost. Additionally, the computational expense for orthogonalization and similarity updates scales with m. In implicitly restarted Arnoldi (IRAM), suboptimal restart strategies can inadvertently discard desirable Ritz components. This leads to repeated regrowth of unwanted components, increasing overall computational cost.

Block Krylov methods enhance robustness, particularly for problems involving clustered or multiple eigenvalues. Subspace recycling techniques enable efficient warm starts for sequences of related matrices A_t.

The numerical experiments employ small, controlled matrices. On large sparse problems, orthogonalization costs in Arnoldi and the expense of shifted solves dominate both time and memory. Polynomial and rational filtering require interval or pole information that may be unavailable or unreliable in applications. And a systematic sensitivity study with respect to non-normality (distribution of $\sigma_{\min}(zI - A)$) and transient growth has not been performed; worst-case iteration and time guarantees therefore remain unclear.

8.2. Spectrum Transformations and Filtering

The polynomial $p_k(A)$ can be strategically employed as a front-end amplifier. This involves applying $p_k(A)$ to the initial random vector, setting $v_{\text{start}} = p_k(A) v_{\text{rand}}$, or applying it once per subspace-growth cycle. The primary objective is to effectively suppress components corresponding to the unwanted region Ω_{unwant}. For optimal performance, the interval mapping $t(\lambda)$, the normalization $p_k(\tau) = 1$, and the polynomial degree k must be explicitly specified. The degree k should be carefully chosen to balance the desired attenuation against the potential for exacerbating rounding errors. A controlled sensitivity study, specifically investigating the impact of interval mis-specification, will be crucial to quantify the extent of pass-band leakage and its effect on convergence.

For rational methods, a significant improvement can be achieved by employing multiple poles strategically placed around the target eigenvalue cluster within the complex plane. These poles should be updated adaptively during the iterative process, leveraging information from current Ritz values. A comparative analysis between fixed and adaptively chosen poles is essential. This comparison should track the decrease in residual norms and monitor the condition number $\| (A - \sigma I)^{-1} \|$ to precisely characterize the trade-off between algorithmic stability and convergence speed. Such an adaptive strategy promises to enhance the robustness and efficiency of rational Krylov methods, especially for

problems with complex or poorly understood spectral distributions.

8.3. IRAM: Algorithmic Pipeline and Measurement Protocol

The Implicitly Restarted Arnoldi Method (IRAM) workflow involves several key steps. It begins by constructing an initial polynomial or a list of shifts to seed the restart process. Subsequently, successive QR steps are applied to the Hessenberg matrix $H_m - \mu I$, typically using Givens or Householder transformations, which yield an updated $H_m \leftarrow G^H H_m G$.

Concurrently, corresponding similarity updates $Q_m \leftarrow Q_m G$ are performed on the orthogonal basis Q_m to preserve the upper-Hessenberg structure and consistently transport Ritz information. Finally, the basis is truncated to k desired directions, and the subspace growth resumes. For a comprehensive analysis, each restart iteration should meticulously record key metrics, including the subdiagonal element $h_{m+1,m}$, Ritz residuals, the orthogonalization cost, and the memory footprint. These measurements should then be rigorously compared against a no-restart baseline to quantify the memory–convergence trade-off inherent in IRAM.

8.4. Pseudospectral Guidance for Robustness and Shifts

To enhance the robustness of eigenvalue algorithms, we propose integrating pseudospectral information into parameter selection. This involves constructing resolvent heatmaps, derived from SVD-based estimates of $\sigma_{\min}(zI - A)$. These heatmaps will guide the selection of shift–invert parameters σ, ensuring they are positioned away from regions where $\sigma_{\min}$ is critically small, thereby avoiding ill-conditioned linear systems.

Furthermore, for polynomial or rational filters, the unwanted region Ω_{unwant} should be explicitly defined as bands characterized by small $\sigma_{\min}$ values. This targeted suppression aims to improve the effectiveness of filtering. Crucially, this approach requires careful documentation of cases where the nominal eigenvalue placement appears favorable, yet significant transient growth is observed, along with the corresponding iteration counts required for convergence. This will provide valuable insights into the impact of non-normality on algorithmic performance.

8.5. Block Methods and Subspace Recycling for Sequences A_t

For eigenvalue problems that are slowly varying or parameter-dependent (i.e., sequences of matrices A_t), subspace recycling offers a potent strategy to improve efficiency. The methodology involves, first, extracting an approximate invariant subspace (e.g., Ritz vectors or Schur vectors) from the solution of A_t. This subspace is then used to augment the initial guess for the solution of A_{t+1}, effectively serving as a block start for the iterative method. During the subsequent Krylov subspace growth, the recycled subspace vectors must be rigorously re-orthogonalized against the newly generated Krylov basis vectors to maintain numerical stability and accuracy.

To ensure the continued relevance and effectiveness of the recycled subspace, it should be periodically refreshed with updated directions, typically by employing a block Arnoldi process. The evaluation of this recycling strategy should comprehensively report iteration counts, orthogonalization costs, and the observed speedup relative to performing a "cold start" for each problem instance A_t. This will quantify the practical benefits of subspace reuse for evolving systems.

8.6. Quantifiable Evaluations and Open Questions

A critical aspect of understanding the performance of eigenvalue algorithms, particularly for non-normal matrices, is to empirically quantify the relationship between theoretical transient growth bounds and observed convergence delays. This study will involve comparing the peaks of the resolvent norm, defined as $\| (zI - A)^{-1} \|_2 = 1/\sigma_{\min}(zI - A)$, with the actual delay phases observed in the error decay curves of iterative methods. By systematically correlating these two metrics, we aim to obtain robust empirical upper bounds on the transient behavior, providing practical insights into how long algorithms might stagnate before achieving asymptotic convergence. This analysis will contribute to a more predictive understanding of algorithm performance on non-normal problems.

To further elucidate the impact of non-diagonalizability, a sensitivity study can be conducted by varying the size of the leading Jordan block, m, within the range $m \in \{1, \dots, 4\}$. The objective is to precisely track the theoretical $k^{m-1}r^k$ behavior that characterizes convergence for matrices with Jordan blocks. The extent to which polynomial or rational transformations can effectively reduce the polynomial factor k^{m-1} at a fixed spectral ratio r. This can provide quantitative evidence of the benefits of preconditioning strategies in mitigating the adverse effects of large Jordan blocks and understanding the trade-offs involved in achieving faster convergence.

Practical applicability can be assessed through comprehensive benchmarking on real-world sparse matrices arising from diverse domains, including Markov chains, control systems, and graph problems. The comparison can involve evaluating the performance of Chebyshev prefiltering combined with Implicitly Restarted Arnoldi Method (IRAM) or rational Krylov methods against baseline approaches such as plain Arnoldi or standard shift–invert iterations. Key performance metrics to be meticulously reported include sparse matrix–vector product costs, the time required for shifted linear system solves, overall memory footprint, and the achieved accuracy. This can provide crucial insights into the efficiency and robustness of advanced eigenvalue algorithms when applied to challenging, large-scale sparse problems.

REFERENCES

[1] L. Page et al., “The PageRank Citation Ranking: Bringing Order to the Web”, Stanford InfoLab, 1999.

[2] L. N. Trefethen, D. Bau, III, Numerical Linear Algebra, SIAM, 1997.

[3] D. Bindel, “CS 6210: Matrix Computations (Lecture Notes)”, Cornell University, 2016.

[4] Z. Bai et al., Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide, SIAM, 2000.

[5] J. H. Wilkinson, The Algebraic Eigenvalue Problem, Clarendon Press, 1965.

[6] G. H. Golub, C. F. Van Loan, Matrix Computations, 4th ed., Johns Hopkins University Press, 2013.

[7] Y. Saad, Numerical Methods for Large Eigenvalue Problems, Revised ed., SIAM, 2011.

[8] B. N. Parlett, The Symmetric Eigenvalue Problem, SIAM, 1998.