Chapter 03: Eigenvalue Problems

Numerical Methods in Computational Physics

Computational Physics - Numerical Methods

December 4, 2025

Contents

1 Introduction and Physical Motivation

The eigenvalue problem is one of the most fundamental in mathematical and computational physics. It appears naturally in:

- **Quantum mechanics**: The time-independent Schrödinger equation $\hat{H}\psi = E\psi$
- **Mechanical vibrations**: Normal modes of oscillation
- **Stability of dynamical systems**: Jacobian analysis
- **Principal component analysis**: Dimensionality reduction
- **Differential equations**: Separation of variables in PDEs

Definition 1.1 (Eigenvalue Problem)**.** *Given a matrix $A \in \mathbb{R}^{n\times n}$ (or $\mathbb{C}^{n\times n}$), the* eigenvalue problem *consists of finding scalars λ (eigenvalues) and nonzero vectors $\mathbf{v}$ (eigenvectors) such that:*

$$A\mathbf{v} = \lambda\mathbf{v} \tag{1}$$

Equivalently, $(\lambda, \mathbf{v})$ is an eigenpair if and only if $(A-\lambda I)\mathbf{v} = \mathbf{0}$, which requires $\det(A - \lambda I) = 0$.

Definition 1.2 (Characteristic Polynomial)**.** *The* characteristic polynomial *of A is:*

$$p(\lambda) = \det(A - \lambda I) = (-1)^n\lambda^n + c_{n-1}\lambda^{n-1} + \cdots + c_1\lambda + c_0 \tag{2}$$

where the eigenvalues are the roots of $p(\lambda) = 0$.

Remark 1.3. *Although the characteristic polynomial exactly defines the eigenvalues, it is **never** computed explicitly in numerical practice because:*

1. *Computing determinants is $\mathcal{O}(n^3)$ and numerically unstable*
2. *Finding roots of high-degree polynomials is an ill-conditioned problem*
3. *The Abel-Ruffini theorem: no closed formula exists for $n \geq 5$*

2 Fundamental Properties

2.1 Spectrum and Spectral Decomposition

Definition 2.1 (Spectrum)**.** *The* spectrum *of A, denoted $\sigma(A)$, is the set of all its eigenvalues:*

$$\sigma(A) = \{\lambda \in \mathbb{C} : \det(A - \lambda I) = 0\} \tag{3}$$

Theorem 2.2 (Spectral Decomposition - Symmetric Matrices)**.** *If $A \in \mathbb{R}^{n\times n}$ is symmetric ($A = A^T$), then:*

1. *All eigenvalues are real: $\lambda_i \in \mathbb{R}$*
2. *There exists an orthonormal basis of eigenvectors*
3. *A can be decomposed as:*

$$A = Q\Lambda Q^T = \sum_{i=1}^{n} \lambda_i \mathbf{q}_i\mathbf{q}_i^T \tag{4}$$

where $Q = [\mathbf{q}_1|\cdots|\mathbf{q}_n]$ is orthogonal ($Q^TQ = I$) and $\Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_n)$

Proof (real eigenvalues). Let λ be an eigenvalue with eigenvector $\mathbf{v} \neq \mathbf{0}$. Then:

$$\mathbf{v}^*A\mathbf{v} = \mathbf{v}^*(\lambda\mathbf{v}) = \lambda\|\mathbf{v}\|^2 \tag{5}$$

$$\mathbf{v}^*A\mathbf{v} = \mathbf{v}^*A^T\mathbf{v} = (A\mathbf{v})^*\mathbf{v} = (\lambda\mathbf{v})^*\mathbf{v} = \bar{\lambda}\|\mathbf{v}\|^2 \tag{6}$$

Therefore $\lambda = \bar{\lambda}$, which implies $\lambda \in \mathbb{R}$. □

2.2 Spectral Radius and Condition Number

Definition 2.3 (Spectral Radius). *The* spectral radius *of A is:*

$$\rho(A) = \max_{\lambda \in \sigma(A)} |\lambda| \tag{7}$$

Definition 2.4 (Spectral Condition Number). *For symmetric positive definite matrices:*

$$\kappa(A) = \text{cond}_2(A) = \frac{\lambda_{\max}}{\lambda_{\min}} = \frac{|\lambda_{\max}|}{|\lambda_{\min}|} \tag{8}$$

The condition number measures the "difficulty" of the eigenvalue problem and affects the convergence of iterative methods.

3 Rayleigh Quotient

The Rayleigh quotient is the fundamental tool for approximating eigenvalues.

Definition 3.1 (Rayleigh Quotient). *For symmetric A and $\mathbf{x} \neq \mathbf{0}$, the* Rayleigh quotient *is:*

$$r(\mathbf{x}) = \frac{\mathbf{x}^T A \mathbf{x}}{\mathbf{x}^T \mathbf{x}} = \frac{\langle \mathbf{x}, A\mathbf{x} \rangle}{\|\mathbf{x}\|^2} \tag{9}$$

Theorem 3.2 (Properties of the Rayleigh Quotient). *Let A be symmetric with eigenvalues $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$. Then:*

1. ***Bounds**: $\lambda_1 \leq r(\mathbf{x}) \leq \lambda_n$ for all $\mathbf{x} \neq \mathbf{0}$*
2. ***Variational characterization**:*

$$\lambda_1 = \min_{\mathbf{x} \neq \mathbf{0}} r(\mathbf{x}) \tag{10}$$

$$\lambda_n = \max_{\mathbf{x} \neq \mathbf{0}} r(\mathbf{x}) \tag{11}$$

3. ***Stationary points**: $\nabla r(\mathbf{x}) = \mathbf{0}$ if and only if $\mathbf{x}$ is an eigenvector*
4. ***Quadratic approximation**: If $\mathbf{x} = \mathbf{q}_i + \epsilon \mathbf{w}$ with $\mathbf{w} \perp \mathbf{q}_i$:*

$$r(\mathbf{x}) = \lambda_i + \mathcal{O}(\epsilon^2) \tag{12}$$

Proof (quadratic approximation). Writing $\mathbf{x}$ in the eigenvector basis: $\mathbf{x} = \sum_j c_j \mathbf{q}_j$ with $c_i \approx 1$ and $|c_j| = \mathcal{O}(\epsilon)$ for $j \neq i$:

$$r(\mathbf{x}) = \frac{\sum_j c_j^2 \lambda_j}{\sum_j c_j^2} = \frac{\lambda_i + \mathcal{O}(\epsilon^2)}{1 + \mathcal{O}(\epsilon^2)} = \lambda_i + \mathcal{O}(\epsilon^2) \tag{13}$$

□

Remark 3.3 (Importance of the Quadratic Approximation). *This property is crucial: the Rayleigh quotient approximates the eigenvalue with $\mathcal{O}(\epsilon^2)$ error even though the eigenvector has $\mathcal{O}(\epsilon)$ error. This means that **eigenvalues converge faster than eigenvectors**.*

4 Power Iteration

4.1 Basic Algorithm

Power iteration is the simplest algorithm for finding the dominant eigenvalue.

Algorithm 1 Power Iteration

Require: Matrix A, initial vector $\mathbf{v}_0$, tolerance ϵ
Ensure: Dominant eigenvalue λ, eigenvector $\mathbf{v}$

```
v ← v_0/‖v_0‖
for k = 1, 2, ... do
    w ← Av
    λ ← v^T w                          ▷ Rayleigh quotient
    v ← w/‖w‖
    if convergence reached then
        return λ, v
    end if
end for
```

4.2 Convergence Analysis

Theorem 4.1 (Convergence of Power Iteration)**.** *Let A be diagonalizable with eigenvalues $|\lambda_1| > |\lambda_2| \geq \cdots \geq |\lambda_n|$. If $\mathbf{v}_0$ has a nonzero component in the direction of $\mathbf{q}_1$, then:*

$$\mathbf{v}_k = \frac{A^k \mathbf{v}_0}{\|A^k \mathbf{v}_0\|} \to \pm \mathbf{q}_1 \quad \text{as } k \to \infty \tag{14}$$

with convergence rate:

$$\|\mathbf{v}_k - \pm \mathbf{q}_1\| = \mathcal{O}\left(\left|\frac{\lambda_2}{\lambda_1}\right|^k\right) \tag{15}$$

Proof. Expanding $\mathbf{v}_0$ in the eigenvector basis: $\mathbf{v}_0 = \sum_{i=1}^{n} c_i \mathbf{q}_i$ with $c_1 \neq 0$:

$$A^k \mathbf{v}_0 = \sum_{i=1}^{n} c_i \lambda_i^k \mathbf{q}_i = \lambda_1^k \left[c_1 \mathbf{q}_1 + \sum_{i=2}^{n} c_i \left(\frac{\lambda_i}{\lambda_1}\right)^k \mathbf{q}_i \right] \tag{16}$$

Since $|\lambda_i/\lambda_1| < 1$ for $i \geq 2$, the terms $(\lambda_i/\lambda_1)^k \to 0$ exponentially. □

Remark 4.2 (Limitations)**.**

- *Slow convergence if $|\lambda_2/\lambda_1| \approx 1$*
- *Only finds the* dominant *eigenvalue*
- *Fails if $|\lambda_1| = |\lambda_2|$ (eigenvalues of equal magnitude)*

5 Inverse Iteration

5.1 Core Idea

To find the eigenvalue closest to a *shift* σ, we apply power iteration to $(A - \sigma I)^{-1}$.

Proposition 5.1. *If λ is an eigenvalue of A with eigenvector $\mathbf{v}$, then:*

$$(A - \sigma I)^{-1} \mathbf{v} = \frac{1}{\lambda - \sigma} \mathbf{v} \tag{17}$$

That is, $(A - \sigma I)^{-1}$ has eigenvalues $1/(\lambda_i - \sigma)$ with the same eigenvectors.

The eigenvalue of $(A - \sigma I)^{-1}$ with largest magnitude corresponds to the λ_i closest to σ.

Algorithm 2 Inverse Iteration

Require: Matrix A, shift σ, initial vector $\mathbf{v}_0$
Ensure: Eigenvalue λ closest to σ, eigenvector $\mathbf{v}$

```
v ← v_0/‖v_0‖
Factorize A − σI = LU                          ▷ Only once
for k = 1, 2, ... do
    Solve (A − σI)w = v                        ▷ Using LU
    v ← w/‖w‖
    λ ← σ + 1/(v^T w)                          ▷ Or use Rayleigh
end for
```

5.2 Convergence Analysis

Theorem 5.2 (Convergence of Inverse Iteration)**.** *Let λ_j be the eigenvalue closest to σ. The convergence is:*

$$error_k = \mathcal{O}\left(\left|\frac{\lambda_j - \sigma}{\lambda_i - \sigma}\right|^k\right) \tag{18}$$

where λ_i is the second closest eigenvalue to σ.

Remark 5.3. *If σ is very close to an eigenvalue, convergence is very fast. In the limit $\sigma \to \lambda_j$, convergence occurs in a single iteration (but $(A - \sigma I)$ becomes singular).*

6 Rayleigh Quotient Iteration

Rayleigh quotient iteration combines inverse iteration with an *adaptive* shift.

Algorithm 3 Rayleigh Quotient Iteration

Require: Symmetric matrix A, initial vector $\mathbf{v}_0$, initial shift σ_0
Ensure: Eigenpair $(\lambda, \mathbf{v})$

```
v ← v_0/‖v_0‖
σ ← σ_0
for k = 1, 2, ... do
    Solve (A − σI)w = v
    v ← w/‖w‖
    σ ← v^T A v                                ▷ Update shift with Rayleigh
end for
return σ, v
```

Theorem 6.1 (Cubic Convergence)**.** *For symmetric matrices, Rayleigh quotient iteration has* ***cubic*** *convergence:*

$$|\sigma_{k+1} - \lambda| = \mathcal{O}(|\sigma_k - \lambda|^3) \tag{19}$$

Proof sketch. Let $\epsilon_k = \sigma_k - \lambda$. By the quadratic approximation of the Rayleigh quotient:

$$\sigma_{k+1} = r(\mathbf{v}_{k+1}) = \lambda + \mathcal{O}(\|\mathbf{v}_{k+1} - \mathbf{q}\|^2) \tag{20}$$

Furthermore, inverse iteration with shift σ_k gives:

$$\|\mathbf{v}_{k+1} - \mathbf{q}\| = \mathcal{O}(|\epsilon_k|) \cdot \|\mathbf{v}_k - \mathbf{q}\| \tag{21}$$

Combining: the error in σ is squared twice, yielding cubic convergence. □

Example 6.2. *With* $\epsilon_0 = 10^{-1}$*:*

- *Iteration 1:* $\epsilon_1 \approx 10^{-3}$
- *Iteration 2:* $\epsilon_2 \approx 10^{-9}$
- *Iteration 3:* $\epsilon_3 \approx 10^{-27}$ *(machine precision)*

Convergence in 2-3 iterations!

7 QR Iteration

7.1 QR Factorization

Definition 7.1 (QR Factorization)**.** *Every matrix* $A \in \mathbb{R}^{n\times n}$ *admits a factorization:*

$$A = QR \tag{22}$$

where Q *is orthogonal* ($Q^TQ = I$) *and* R *is upper triangular.*

7.2 Basic QR Algorithm

Algorithm 4 QR Iteration (basic)

Require: Matrix $A_0 = A$
Ensure: Diagonal/triangular matrix with eigenvalues on the diagonal
1: **for** $k = 0, 1, 2, \ldots$ **do**
2: $A_k = Q_kR_k$ ▷ QR factorization
3: $A_{k+1} = R_kQ_k$ ▷ Multiply in reverse order
4: **end for**

Theorem 7.2 (Similarity Property)**.** *All matrices* A_k *are similar:*

$$A_{k+1} = R_kQ_k = Q_k^T(Q_kR_k)Q_k = Q_k^TA_kQ_k \tag{23}$$

Therefore, they share the same eigenvalues.

Theorem 7.3 (Convergence of the QR Algorithm)**.** *For matrices with eigenvalues of distinct magnitudes* $|\lambda_1| > |\lambda_2| > \cdots > |\lambda_n|$*:*

$$A_k \to \begin{pmatrix} \lambda_1 & * & \cdots & * \\ 0 & \lambda_2 & \cdots & * \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & \lambda_n \end{pmatrix} \tag{24}$$

with elements (i, j) *for* $i > j$ *decaying as* $\mathcal{O}(|\lambda_i/\lambda_j|^k)$*.*

7.3 Relation to Power Iteration

Proposition 7.4. *The QR algorithm can be viewed as* simultaneous *power iteration over all eigenvectors:*

$$\tilde{Q}_k = Q_0Q_1 \cdots Q_{k-1} \tag{25}$$

converges to the eigenvector matrix, and

$$A^k = \tilde{Q}_k\tilde{R}_k \tag{26}$$

where $\tilde{R}_k = R_{k-1}R_{k-2} \cdots R_0$*.*

8 Lanczos Method

8.1 Motivation

For large sparse matrices ($n \gg 1000$), the previous methods are impractical:

- QR iteration: $\mathcal{O}(n^3)$ per iteration
- Full storage: $\mathcal{O}(n^2)$

The Lanczos method is optimal for **symmetric sparse** matrices, with:

- Complexity $\mathcal{O}(kn)$ for k eigenvalues
- Only requires matrix-vector products

8.2 Krylov Subspace

Definition 8.1 (Krylov Subspace)**.** *Given A and an initial vector $\mathbf{v}$, the* Krylov subspace *of order k is:*

$$\mathcal{K}_k(A, \mathbf{v}) = \text{span}\{\mathbf{v}, A\mathbf{v}, A^2\mathbf{v}, \ldots, A^{k-1}\mathbf{v}\} \tag{27}$$

Remark 8.2. *Power iteration works in $\mathcal{K}_k$ but only extracts information from the vector $A^{k-1}\mathbf{v}$. Lanczos leverages* all *the information in the subspace.*

8.3 Lanczos Algorithm

The algorithm constructs an orthonormal basis $\{\mathbf{q}_1, \ldots, \mathbf{q}_k\}$ of $\mathcal{K}_k$ such that:

$$Q_k^T A Q_k = T_k \tag{28}$$

where T_k is **symmetric tridiagonal**:

$$T_k = \begin{pmatrix} \alpha_1 & \beta_1 & & \\ \beta_1 & \alpha_2 & \beta_2 & \\ & \ddots & \ddots & \ddots \\ & & \beta_{k-1} & \alpha_k \end{pmatrix} \tag{29}$$

Algorithm 5 Lanczos Iteration

Require: Symmetric matrix A, initial vector $\mathbf{q}_1$ with $\|\mathbf{q}_1\| = 1$
Ensure: Coefficients α_j, β_j of the tridiagonal matrix T

$\beta_0 \leftarrow 0$, $\mathbf{q}_0 \leftarrow \mathbf{0}$
for $j = 1, 2, \ldots, k$ **do**
  $\mathbf{w} \leftarrow A\mathbf{q}_j$
  $\alpha_j \leftarrow \mathbf{q}_j^T \mathbf{w}$
  $\mathbf{w} \leftarrow \mathbf{w} - \alpha_j \mathbf{q}_j - \beta_{j-1}\mathbf{q}_{j-1}$
  $\beta_j \leftarrow \|\mathbf{w}\|$
  if $\beta_j < \epsilon$ **then**
    break ▷ Invariant subspace found
  end if
  $\mathbf{q}_{j+1} \leftarrow \mathbf{w}/\beta_j$
end for

8.4 Ritz Values

Definition 8.3 (Ritz Values and Vectors). *The eigenvalues of T_k are called* Ritz values *and approximate the eigenvalues of A. If $T_k\mathbf{s} = \theta\mathbf{s}$, then $\mathbf{y} = Q_k\mathbf{s}$ is the corresponding* Ritz vector.

Theorem 8.4 (Lanczos Convergence). *The extreme Ritz values (largest and smallest) converge exponentially fast toward the extreme eigenvalues of A. For matrices with well-separated eigenvalues, convergence is:*

$$|\theta_1^{(k)} - \lambda_1| = \mathcal{O}\left(\left(\frac{\lambda_2 - \lambda_n}{\lambda_1 - \lambda_n}\right)^{2k}\right) \tag{30}$$

8.5 Loss of Orthogonality

Remark 8.5 (Numerical Issue). *In finite arithmetic, Lanczos vectors gradually lose orthogonality due to rounding errors. This causes:*

- *Appearance of "ghost" eigenvalues (spurious duplicates)*
- *Delayed convergence*

***Solution**: Reorthogonalization (partial or complete).*

9 Computational Complexity

Table 1: Computational complexity comparison

Method	Cost/iter	Storage	Eigenvalues
Power Iteration	$\mathcal{O}(n^2)$	$\mathcal{O}(n)$	1 (dominant)
Inverse Iteration	$\mathcal{O}(n^3) + \mathcal{O}(n^2)$	$\mathcal{O}(n^2)$	1 (closest to σ)
Rayleigh Quotient	$\mathcal{O}(n^3)$	$\mathcal{O}(n^2)$	1 (any)
QR Iteration	$\mathcal{O}(n^3)$	$\mathcal{O}(n^2)$	All
Lanczos	$\mathcal{O}(mn)$	$\mathcal{O}(kn)$	k extreme

Where m is the number of nonzero elements of A (for sparse matrices).

10 Stability and Conditioning Analysis

10.1 Eigenvalue Sensitivity

Theorem 10.1 (Bauer-Fike Theorem). *Let A be diagonalizable with $A = X\Lambda X^{-1}$. If μ is an eigenvalue of $A + E$, then:*

$$\min_{\lambda \in \sigma(A)} |\mu - \lambda| \leq \kappa(X)\|E\| \tag{31}$$

where $\kappa(X) = \|X\|\|X^{-1}\|$.

Corollary 10.2 (Normal Matrices). *For normal matrices ($AA^* = A^*A$, including symmetric), $\kappa(X) = 1$, and eigenvalues are perfectly conditioned:*

$$|\mu - \lambda| \leq \|E\| \tag{32}$$

10.2 Eigenvector Sensitivity

Eigenvectors are more sensitive than eigenvalues:

Theorem 10.3. *For simple eigenvalues of symmetric matrices, if $\tilde{\mathbf{v}}$ is the perturbed eigenvector:*

$$\sin \angle(\mathbf{v}, \tilde{\mathbf{v}}) \leq \frac{\|E\|}{\min_{j \neq i} |\lambda_i - \lambda_j|} \tag{33}$$

Remark 10.4. *Eigenvectors corresponding to nearby eigenvalues ("clusters") are inherently ill-conditioned.*

11 Method Summary

Table 2: Method selection guide

Situation	Recommended Method
Dominant eigenvalue	Power Iteration
Specific eigenvalue	Inverse Iteration with shift
Maximum precision, one eigenvalue	Rayleigh Quotient Iteration
All eigenvalues, small n	QR Iteration (or NumPy/SciPy)
Large sparse matrices	Lanczos (SA/LA)
Large non-symmetric matrices	Arnoldi (not covered here)

12 Conclusions

Iterative methods for eigenvalue problems are based on fundamental ideas:

1. **Rayleigh Quotient**: Approximates eigenvalues with quadratic error
2. **Power Iteration**: Linear convergence toward the dominant eigenvalue
3. **Shift-and-Invert**: Transforms the problem to converge to any eigenvalue
4. **Krylov Subspaces**: Exploit structure for sparse matrices

Method selection depends on:

- Matrix size (n)
- Structure (dense vs sparse)
- Number of eigenvalues required
- Special properties (symmetry, positive definiteness)

References

[1] Trefethen, L.N. and Bau, D. *Numerical Linear Algebra*. SIAM, 1997.

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

[3] Saad, Y. *Numerical Methods for Large Eigenvalue Problems*, 2nd ed. SIAM, 2011.

[4] Parlett, B.N. *The Symmetric Eigenvalue Problem*. SIAM Classics in Applied Mathematics, 1998.

[5] Watkins, D.S. *The Matrix Eigenvalue Problem: GR and Krylov Subspace Methods*. SIAM, 2007.