Eigenvalue Calculation

Arjun Pavanje: EE24BTECH11005

November 18, 2024

1 Introduction

Eigenvalue computation is a cornerstone of numerical linear algebra with widespread applications across sciences and engineering. This report delves into the implementation of eigenvalue computation using Householder transformations and Givens rotations.

2 Definition

Firstly, what are eigenvalues? Eigenvalues of a matrix A are those values $'\lambda'$ that satisfy the relation $A\mathbf{v} = \lambda\mathbf{v}$.

It is important to understand the intuitive sense behind them. The $A\mathbf{v}$ represents a linear transformation (rotation/scaling/shear/translation) being applied on $\mathbf{v}$. $\lambda\mathbf{v}$ represents the vector $\mathbf{v}$ being scaled by a factor of λ. Both of them being equal implies that for a matrix A, there exist some special vectors $\mathbf{v}$ for which applying the linear transformation represented by the matrix just has a scaling effect on it, i.e. $\mathbf{v}$ lies in its own span after going through the linear transformation represented by $A\mathbf{v}$.

Now, to define it a little more rigorously,
Eigenvalues are defined as scalars $\lambda \in \mathbb{C}$ for which there exists a non-zero vector ($\mathbf{v} \in \mathbb{C}^n, v \neq 0$) such that:

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

where A is an $(n \times n)$ matrix. The scalar (λ) is the factor by which the vector $\mathbf{v}$, called the *eigenvector*, is scaled under the linear transformation represented by A.

3 Applications

Eigenvalue computation has widespread applications in various branches of science and engineering.

4 Theory

4.1 Why Not Solve $\det(A - \lambda I) = 0$ Directly?

The characteristic polynomial $\det(A - \lambda I) = 0$ theoretically provides all eigenvalues of A. However, solving it directly is impractical for several reasons:

1. **No closed form solution:** According to Abel-Ruffini theorem, there is no general solution for a polynomial of degree n beyond n greater than 4. What it means is, like how there exists a general solution of a quadratic equation

$$\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

, similar formulae (but much more complicated) exist for cubic and biquadratic polynomials. But no such generalized formula to find all the roots exist for quintic polynomials and beyond.

2. **Numerical instability:** The polynomial roots are highly sensitive to perturbations, leading to numerical instability.

3. **Computational Expense:** For large matrices, the degree of the polynomial is high, making root-finding algorithms computationally expensive.

4.2 Schur Form

The Schur form is a useful matrix decomposition where any square matrix $A \in \mathbb{C}^{n \times n}$ can be written as $A = QTQ^H$, where Q, is a unitary matrix and T is an upper triangular matrix. The eigenvalues of A are the diagonal elements of T, making the Schur form desirable for finding eigenvalues. It simplifies computations greatly. To find it, QR algorithm is implemented.

4.3 QR Decomposition

QR decomposition is a matrix decomposition technique where a given matrix A is decomposed into a product of two matrices (one orthogonal, one upper triangular):

$$A = QR,$$

where:

- Q is an orthogonal matrix, meaning $(Q^\top Q = I)$,
- R is an upper triangular matrix.

QR decomposition is widely used in solving linear systems, eigenvalue problems, etc. This is due to numerical stability of QR algorithms as compared to others.

4.4 Similar Matrices

Two square matrices A and B of size $n \times n$ are said to be *similar* if there exists an invertible matrix P such that:

$$B = P^{-1}AP.$$

4.4.1 Properties of Similar Matrices:

1. **Same Eigenvalues:** Similar matrices share the same eigenvalues, including their algebraic multiplicities.
2. **Determinant and Trace:** The determinant and trace of similar matrices are equal.
3. **Same Rank:** Similarity transformations do not change the rank of a matrix.

4.4.2 Intuition:

Similarity captures the idea that A and B represent the same linear transformation but in different bases. The matrix P provides the change of basis.

4.4.3 Proof that Similar Matrices Have the Same Eigenvalues

Given that A, B are similar matrices, and P is an invertible matrix such that $B = P^{-1}AP$ By definition of eigenvalues,

$$\begin{aligned}\det(B - \lambda I) &= \det(P^{-1}AP - \lambda I) \\ &= \det(P^1AP - \lambda P^{-1}P) \\ &= \det(P^{-1})\det(A - \lambda I)\det(P) \\ &= \det(P^{-1}P)\det(A - \lambda I) \\ &= \det(A - \lambda I)\end{aligned}$$

4.5 Eigenvalue preservation and Convergence

The QR decomposition of a square matrix A is the factorization:

$$A = QR,$$

where Q is an orthogonal matrix ($Q^\top Q = I$) and R is an upper triangular matrix. The QR algorithm iteratively computes:

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

where $A_k = Q_k R_k$ is the QR decomposition of A_k. Eventually repeating this process will lead to the matrix A_n being an upper triangular matrix, which is what we desire. When the matrix is upper triangular, we can just note its principal diagonal elements, as they will be the eigenvalues of the matrix.

4.5.1 Eigenvalue Conservation

To see why eigenvalues are conserved during the QR iteration, consider the similarity transformation performed at each step:

$$A_{k+1} = R_k Q_k = Q_k^\top A_k Q_k.$$

Given, $A_k = Q_k R_k$ and $A_{k+1} = R_k Q_k$. Consider the definition of eigenvalue

$$\det(A_{k+1} - \lambda I) = \det(R_k Q_k - \lambda I).$$

Writing I as $Q_k^\top Q_k$ and R_k as $Q_k^\top A_k$

$$\begin{aligned}\det(R_k Q_k - \lambda I) &= \det(Q_k^\top A_k Q_k - \lambda Q_k^\top Q_k) \\ &= \det(Q_k^\top)\det(A_k - \lambda I)\det(Q_k) \\ &= \det(Q_k^\top Q_k)\det(A_k - \lambda I)\det(Q_k) \\ &= \det(A_k - \lambda I) \qquad (\because Q_k^\top Q_k = I)\end{aligned}$$

So, A_{k+1}, A_k are similar, i.e. eigenvalues are conserved in the transformation of A_k to A_{k+1}

4.5.2 Convergence to Upper Triangular Form

The QR iteration gradually reduces A_k to an upper triangular form. When the matrix is upper triangular, we can just note its principal diagonal elements, as they will be the eigenvalues of the matrix. The eigenvalues of a matrix are conserved in the QR algorithm because the process relies on similarity transformations. Successive iterations lead to con- vergence towards an upper triangular form, with the eigenvalues appearing on the diagonal. For faster convergence, shift strategies such as Rayleigh's quotient are employed, ensuring numerical stability and efficiency.

4.6 Hessenberg Form via Householder Transformation

A square matrix A of order $n \times n$ is said to be in upper Hessenberg form if all the entries below the first subdiagonal are zero. To define it a little more mathematically, if the elements of a matrix are represented as $[a_{ij}]$, it is said to be in Upper Hessenberg iff

$$a_{ij} = 0, \forall i > j + 1$$

For example:

$$H = \begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & 0 & \times & \times \end{bmatrix}.$$

Applying QR decomposition to reach Schur-form to the matrix after it is in Upper Hessenberg form greatly accelerates rate of convergence.
Householder transformations are used to reduce a general matrix A to Hessenberg form. A Householder reflector is an orthogonal matrix defined as:

$$P = I - 2\mathbf{u}\mathbf{u}^\top$$

where $\|\mathbf{u}\| = 1$
If the entries of the matrix are complex, transpose is to replaced with conjugate transpose. Vector $\mathbf{u}$ must be carefully chosen such that the resultant matrix P obtained from it must zero out all elements below the first subdiagonal for that particular column while maintaining similarity to preserve eigenvalues.

4.6.1 Choosing the Vector u

For a given column vector $x \in \mathbb{R}^n$, the vector u is chosen as:

$$\mathbf{u} = \frac{\mathbf{x} - \|\mathbf{x}\|\rho\mathbf{e_1}}{\|\mathbf{x} - \|\mathbf{x}\|\rho\mathbf{e_1}\|}$$

where:

- $\mathbf{e_1}$ is impulse vector of appropriate dimensions, first element 1
- ρ is something we have a degree of freedom in choosing as long as $|\rho| = 1$

Usually, $\rho = -sign(x_1)$, but here for ease of calculation $\rho = -e^{j\phi}$ where $x_1 = |x_1|e^{j\phi}$

We then construct Householder reflector matrix P_1 as,

$$P_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times & \times \end{bmatrix} = \begin{bmatrix} 1 & \mathbf{0}^\top \\ \mathbf{0} & I_4 - 2\mathbf{u_1}\mathbf{u_1}^\top \end{bmatrix}$$

For a general case P_i (for i^{th} column) would be to create $I_{n-i} - 2u_i u_i^\top$ and expand it from the top left such that it is like an $n \times n$ identity matrix with it as the bottom right submatrix

$$P_i = \begin{bmatrix} 1 & \mathbf{0}^\top \\ \mathbf{0} & I_4 - 2\mathbf{u_i}\mathbf{u_i}^\top \end{bmatrix}$$

As mentioned above, if enteries are complex, conjugate transpose must be taken instead of transpose.
The multiplication of P_1 from the left inserts the desired zeros in first column of A. The multiplication from the right is necessary in order to have similarity. Because of the nonzero structure of P_1 the first column of P_1A is not affected. Hence, the zeros stay there.

4.6.2 Example: Reducing a Matrix to Hessenberg Form

We start with a general square matrix A:

$$A = \begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \end{bmatrix}.$$

The goal is to apply successive Householder transformations to eliminate elements below the first subdiagonal.

$$\begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \end{bmatrix} \xrightarrow{P_1} \begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \end{bmatrix} \xrightarrow{P_2} \begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & 0 & \times & \times \end{bmatrix}.$$

After applying these transformations, the matrix is reduced to Hessenberg form.

4.6.3 Matrix Representation of Transformations

Each Householder transformation P_k is applied as:

$$P_k A P_k^\top.$$

The sequence of transformations for $k = 1, 2, \ldots, n-2$ results in the Hessenberg matrix H:

$$H = (P_{n-2} \cdots P_2 P_1) A (P_1^\top P_2^\top \cdots P_{n-2}^\top).$$

All the Householder reflector matrices P_i are all orthogonal (because of how we construct them). So $(P_1 P_2 \cdots P_{n-2})$ is orthogonal i.e. H matrix can be represented as,

$$H = P^\top A P$$

where $P = (P_{n-2} \cdots P_2 P_1)$ and $P^\top = P^{-1}$
So, H and A are similar matrices, i.e. eigenvalues are conserved. Householder reflection is a similarity transform.

Algorithm 1 Householder Reflection for Upper Hessenberg Form

Require: A square matrix A of size $n \times n$
Ensure: A is reduced to upper Hessenberg form

1: **for** $k = 1$ to $n - 2$ **do**
2: Extract the sub-vector $\mathbf{x}$ from column k, consisting of elements $a_{k+1,k}, a_{k+2,k}, \ldots, a_{n,k}$.
3: Compute the 2-norm of $\mathbf{x}$, $\|\mathbf{x}\|_2$.
4: Let $\rho = -e^{j\phi}$, where $e^{j\phi}$ is derived from the first element of $\mathbf{x}$.
5: Define the vector $\mathbf{u}$ as:

$$\mathbf{u} = \frac{\mathbf{x} - \rho\|\mathbf{x}\|_2 \mathbf{e}_1}{\|\mathbf{x} - \rho\|\mathbf{x}\|_2 \mathbf{e}_1\|_2},$$

where $\mathbf{e}_1$ is the first impulse vector of appropriate size.
6: Form the Householder reflector:

$$H_k = I - 2\mathbf{u}\mathbf{u}^T,$$

where I is the identity matrix of size $(n-k) \times (n-k)$.
7: Expand H_k to construct P, where P is an identity matrix of size $n \times n$ with H_k as its bottom-right submatrix.
8: Update A using the similarity transformation:

$$A \leftarrow PAP.$$

9: **end for**=0

4.7 QR Decomposition by Givens Rotation

Givens rotations are used to eliminate specific elements of a matrix while preserving its eigenvalues. This approach is particularly efficient for structured matrices like Hessenberg matrices, where only a few elements need to be zeroed.

4.7.1 Givens Rotation and Its Matrix Form

A Givens rotation matrix ($G(i, j, \theta)$ zeroes out the element a_{ij} by rotating in the (i, j)-plane. It is defined as:

$$G(i,j,\theta) = \begin{bmatrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & c & \cdots & s & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & -\overline{s} & \cdots & \overline{c} & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{bmatrix}$$

where $\cos\theta$ and $\sin\theta$ are chosen such that the target element is eliminated.
To choose the values of c and s for the Givens rotation in QR decomposition, let a_j be the element we wish to null out (i.e. make 0). Pick an arbitrary non-zero pivot element a_i (on a different row). Usually, if we wish to null a particular sub-diagonal element, we pick the principal diagonal element above it as a pivot.

$$c = \frac{\overline{a_i}}{\sqrt{a_i^2 + a_j^2}}, \quad s = \frac{-\overline{a_j}}{\sqrt{a_i^2 + a_j^2}}$$

Givens rotation essentially rotates the two rows that a_i and a_j are on such that $a_j = 0$ after rotation, other rows remain unaffected.

These values rotate the vector formed by a_i and a_j to eliminate a_j while maintaining the orthogonality of the rotation matrix. The choice of c and s ensures that the resulting transformed matrix has zeros below the diagonal in the desired locations.

4.7.2 Transforming a Hessenberg Matrix to Upper Triangular

Starting with a Hessenberg matrix A:

$$A = \begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & 0 & \times & \times \end{bmatrix},$$

we apply a sequence of Givens rotations to eliminate the subdiagonal elements.

$$\begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & 0 & \times & \times \end{bmatrix} \xrightarrow{G(3,2,\theta_1)} \begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ 0 & 0 & \times & \times \\ 0 & 0 & \times & \times \end{bmatrix} \xrightarrow{G(4,3,\theta_2)} \begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ 0 & 0 & \times & \times \\ 0 & 0 & 0 & \times \end{bmatrix}.$$

After all Givens rotations, the resulting matrix is upper triangular:

$$R = \begin{bmatrix} \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & 0 & \times & \times \\ 0 & 0 & 0 & \times \end{bmatrix}.$$

4.7.3 Givens Rotations and QR Decomposition

The sequence of Givens rotations $G_1, G_2, \ldots, G_m$ satisfies:

$$G_m \cdots G_2 G_1 A = R,$$

where R is upper triangular. The QR decomposition is obtained by combining the transposes of the Givens rotations into Q:

$$A = QR, \quad Q = G_1^\top G_2^\top \cdots G_m^\top.$$

$$\begin{aligned} A_{k+1} &= R_k Q_k \\ &= (G_n \ldots G_2 G_1) A_k (G_1^\top G_2^\top \ldots G_n^\top) \\ &= (G_n \ldots G_2 G_1) A_k (G_n \ldots G_2 G_1)^\top \end{aligned}$$

$(G_n \ldots G_2 G_1)$ is an orthogonal matrix, so by the definition of similar matrices, A_{k+1} and A_k are similar, i.e. they have the same eigenvalues

Algorithm 2 QR Factorization of an Upper Hessenberg Matrix using Givens Rotations

Require: An upper Hessenberg matrix $H \in \mathbb{C}^{n \times n}$

Ensure: Updated $H' = RQ$, where $H = QR$

1: **for** $k = 1$ to $n - 1$ **do**

2: Identify the pivot element:

$$h_i = H(k, k), \quad h_j = H(k, k+1).$$

3: Compute the rotation parameters:

$$r = \sqrt{|h_i|^2 + |h_j|^2}, \quad c = \frac{\overline{h_i}}{r}, \quad s = \frac{\overline{h_j}}{r}.$$

4: Apply Givens rotation to rows k and $k+1$ on the right:

$$\begin{aligned} H(k, :) &\leftarrow cH(k, :) - sH(k+1, :), \\ H(k+1, :) &\leftarrow \overline{c} H(k+1, :) + \overline{s} H(k, :). \end{aligned}$$

5: Apply the reverse Givens rotation to columns k and $k+1$ on the left:

$$\begin{aligned} H(:, k) &\leftarrow \overline{c} H(:, k) - \overline{s} H(:, k+1), \\ H(:, k+1) &\leftarrow cH(:, k+1) + sH(:, k). \end{aligned}$$

6: **end for**

=0

4.8 Improving Convergence Rate by Rayleigh's Quotient

Introducing a shift to the matrix can drastically improve its convergence rate. If σ is the shift amount, before applying QR decomposition on a matrix shift the matrix by σ, perform QR on shifted matrix, shift the matrix back by σ

$$A'_k = A - \sigma I$$
$$A'_{k+1} = (Q'_k R'_k) + \sigma I$$

to calculate σ, follow the following algorithm.

Algorithm 3 Hessenberg QR Algorithm with Rayleigh Quotient Shift

Require: Upper Hessenberg matrix $H \in \mathbb{C}^{n \times n}$

$k \leftarrow 0$
for $m = n, n-1, ..., 2$ **do**
  $k \leftarrow k + 1$
  repeat
    $\sigma_k \leftarrow H_{m-1,m-1}$
    $H_k - \sigma_k I = Q_k R_k$ Decomposition
    $H_{k+1} \leftarrow R_k Q_k + \sigma_k I$
  until $|H_{m,m-1}|$ is sufficiently small (like say 10^{-10})
end for=0

4.9 Jordan Blocks and Eigenvalues

Jordan blocks are used to represent non-diagonalizable matrices. A Jordan block occurs when the Matrix on which eigenvalue alorithm is applied does not converge to Upper-Triangular. One example is when the enteries of the matrix are real, but they some of the eigenvalues are complex. Presence of a Jordan block indicates the presence of a pair of complex conjugate eigenvalues.

$$\begin{bmatrix} \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & 0 & \times \end{bmatrix}$$

4.9.1 Accounting for Jordan Blocks

To handle Jordan blocks during eigenvalue computation:

- Identify where the subdiagonal element is non zero.
- Consider the Jordan block as shown in the image, solve the obtained 2×2 matrix for its eigenvalues.
- The obtained complex conjugate roots are eigenvalues of the matrix

5 Chosen Algorithm

Algorithm 4 Eigenvalue Calculation

Require: A square matrix $A \in \mathbb{C}^{n \times n}$
Ensure: Eigenvalues of A
To Upper Hessenberg form (via Householder reflection):
for $k = 1$ TO $n - 2$ **do**
  Compute Householder reflector P_k to zero out subdiagonal elements.
  Update A: $A \leftarrow P_k A P_k^T$
end for
QR Decomposition (via Givens Rotation):
while not converged **do**
  QR Decomposition:
  for $k = 1$ TO $n - 1$ **do**
    Compute Givens rotation matrix G_k to zero out the $(k + 1, k)$ element.
    Apply G_k from the left and right to A.
  end for
  Form $A \leftarrow RQ$
end while
Extract Eigenvalues from Quasi-Triangular Matrix
for $i = 1$ TO n **do**
  if $|A_{i+1,i}| < \epsilon$ **then**
    Eigenvalue: $\lambda_i = A_{i,i}$
  else
    Compute eigenvalues of the 2×2 submatrix by solving quadratic:

$$\begin{bmatrix} A_{i,i} & A_{i,i+1} \\ A_{i+1,i} & A_{i+1,i+1} \end{bmatrix}$$

  end if
end for
return Eigenvalues $\lambda_1, \lambda_2, ..., \lambda_n$ =0

The eigenvalue computation algorithm combines:

1. **Householder Transformations:** To reduce the matrix to Hessenberg form, simplifying subsequent computations.

2. **QR Iterations with Givens Rotations:** To iteratively compute an upper triangular matrix, where the diagonal elements converge to the eigenvalues.

6 Time Complexity Analysis

1. **Householder Transformations** Reducing an $(n \times n)$matrix to Hessenberg form takes $O(n^3)$ operations.

2. **QR Iterations** Each QR decomposition requires $O(n^2)$ operations, on $n - 1$ rows. Overall time complexity is $O(n^3)$

3. **Overall Complexity** The total time complexity of the combined approach is $O(n^3)$, making it one of the better eigenvalue computation methods.

7 Comparison of Algorithms

Effectiveness of eigenvalue algorithms is relative. Each algorithms have their own pros and cons. For instance, while an algorithm like Jacobi will work best, but it is limited to real, symmetric matrices. Below are a few popular algorithms, with their time complexities, pros, cons, conditions.

Algorithm	**Pros**	**Cons**	**Time Complexity**	**Conditions**
Householder + Givens	Stable, precise; preserves sparsity.	Complex preprocessing for Hessenberg form.	$O(n^3)$	Any square matrix.
Jacobi Method	Accurate for symmetric matrices; simple.	Slow for non-symmetric matrices; inefficient for large ones.	$O(n^3)$ (dense)	Symmetric (real/complex).
Lanczos Algorithm	Efficient for sparse symmetric matrices.	Sensitive to rounding; limited to Hermitian matrices.	$O(kn^2)$	Sparse, symmetric/Hermitian.
QR Algorithm	Robust for eigenvalue computation.	High cost for large dense matrices.	$O(n^3)$	Any square matrix.
Arnoldi Algorithm	Handles non-symmetric sparse matrices.	Requires reorthogonalization; less stable.	$O(kn^2)$	Sparse, any matrix.
Power Iteration	Simple, finds dominant eigenvalue.	Only computes the largest eigenvalue; slow for close eigenvalues.	$O(kn^2)$	Any square matrix.

Why Givens + Householder is Preferred

Householder Transformations:

- Efficient for reducing matrices to a simpler form (upper Hessenberg or tridiagonal).
- Provides numerical stability.
- Reduces the complexity of subsequent steps, like QR decomposition.

Givens Rotations:

- Ideal for zeroing specific matrix elements without affecting others.
- Well-suited for sparse matrices.
- Simple to implement and numerically stable.

Advantages:

- This algorithm has no limitation in terms of what the enteries of the matrix can be (i.e. complex/real enteries), or on what the matrix has to be (ex: matrix does not have to be only symmetric/hermitian for it to work).
- The chosen algorithm of Givens rotations and Householder reflections balances numerical stability and efficiency.
- By reducing a matrix to Hessenberg form using Householder transformations and iteratively zeroing out elements with Givens rotations, it preserves orthogonality and sparsity while accelerating rate of convergence.
- Implementing a shift like Rayleigh quotient, rate of convergence drastically increases.

8 Conclusion

The combination of Householder transformations and Givens rotations provides a robust and efficient method for eigenvalue computation, especially for dense matrices. While other methods may be tailored for specific matrix types, this approach balances numerical stability, computational efficiency, and broad applicability.

9 References

References

[1] ETH Zurich *QR Algorithms*. Available at: `https://people.inf.ethz.ch/arbenz/ewp/Lnotes/chapter4.pdf`.

[2] Blue1Brown. *Essence of Linear Algebra Playlist*. Available at: `https://youtube.com/playlist?list=PLZHQObOWTQDPD3MizzM2xVFitgF8hE_ab&si=oxNwAmrJMTRl0K5V`.

[3] Axler, S. (2024). *Linear Algebra Done Right (4th ed.)*. Available at: `https://www.linear.axler.net/LADR4e.pdf`.

[4] Wikipedia contributors. (2024). *Eigenvalue algorithm*. Available at: `https://en.wikipedia.org/wiki/Eigenvalue_algorithm#Hessenberg_and_tridiagonal_matrices`.