Report

Ellanti Rohith-EE24BTECH11020

November 18, 2024

Abstract

This report is about my work on process of selecting an appropriate algorithm for eigenvalue computation, and explaining about it. The study includes a comparison of various methods and advantages of the Algorithm and overview of it's implmentation.

Contents

1 Introduction

After analysing a few algorithms to calculate eigen values. The QR Algorithm is selected due to its robustness and versatility. Algorithms like Rayleigh Quotient Iteration, Power Iteration were analyzed to understand their pros and cons.

The algorithms explored during this study include:

1. Power Iteration
2. Inverse Iteration
3. Rayleigh Quotient Iteration
4. QR Algorithm

2 Methodology

To select an appropriate algorithm, several methodologies were explored:

1. Understanding eigenvalue computation algorithms through theoretical analysis.
2. Comparing algorithms such as Rayleigh Quotient Iteration and the QR Algorithm based on factors like robustness, and computational cost.
3. Selecting the QR Algorithm for its ability to handle both real and complex eigenvalues.

3 Explanation of Algorithm

3.1 QR Decomposition

The QR algorithm is an iterative method to compute the eigenvalues of a square matrix A. The idea is to factorize A into an orthogonal matrix Q and an upper triangular matrix R.And the multiply those two matrices in reverse to get a new matrix which has same eigen values as A

Steps in the QR Algorithm:

1. Start with a square matrix $A_0 = A$.
2. Perform the QR decomposition:

$$A_k = Q_k R_k$$

   where Q_k is orthogonal ($Q_k^T Q_k = I$), and R_k is upper triangular.

3. Form a new matrix by multiplying R_k and Q_k:

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

4. Repeat steps until A_k converges to an upper triangular matrix .

After some iterations depending on matrix size, the matrix A_k converges, and its diagonal entries approximate the eigenvalues of the original matrix A.

3.2 QR Factorization

We can do QR factorization in many ways

1. Gram-Schmidt Algorithm

2. Householder Reflections for QR factorization

Householder Transformation is chosen to compute the QR decomposition by systematically zeroing out elements below the diagonal of the matrix. due to its Numerical stabilty,Precision, Robustness.
Definition: A Householder reflection is an orthogonal transformation defined as:

$$H = I - 2\frac{vv^T}{v^Tv},$$

where v is a vector chosen to zero out specific components of a column of A.

Steps in Householder Reflections:

1. For a given column j of A, construct a vector v such that:

$$H_jA$$

2. Multiply H_j to A from left to produce a new matrix:

$$A' = H_jA.$$

3. Repeat this process for each column until A is transformed into an upper triangular matrix R.

4. Combine all Householder matrices $H_1, H_2, \dots, H_{n-1}$ to form Q:

$$Q = H_1H_2 \dots H_{n-1}.$$

Advantages of Householder Reflections:

1. They are numerically stable, making them suitable for large matrices.

2. They are efficient for dense matrix computations.

4 Computational Complexity of the QR Algorithm with Householder Transformations

The computational complexity of the algorithm is as follows:

4.1 Matrix Multiplication

Multiplying two $n \times n$ matrices requires $O(n^3)$ operations. Since the algorithm repeatedly multiplies matrices during the iterative process, this operation dominates the runtime.

4.2 Householder Transformation

The Householder transformation involves the following steps:

1. **Vector Norms:** Each norm calculation requires $O(n)$.
2. **Matrix-Vector Multiplications:** Each transformation requires $O(n^2)$.
3. **Generating the Householder Matrix:** This requires $O(n^2)$.
4. **Multiplying the Householder Matrix with the Input Matrix:** This takes $O(n^3)$.

Householder transformations are performed $n-1$ times in the decomposition, so the total cost per iteration is $O(n^3)$.

4.3 Conjugate Transpose

Taking the conjugate transpose of a matrix involves iterating over half the elements of the $n \times n$ matrix. This step has a complexity of $O(n^2)$.

4.4 Iterative QR Algorithm

In the main iterative loop:

1. The matrix is multiplied with Q^* and then decomposed again into Q and R using the Householder procedure.
2. The off-diagonal elements are checked to ensure convergence.

Each iteration involves:

1. **Matrix Multiplication:** $O(n^3)$.
2. **Householder Decomposition:** $O(n^3)$.
3. **Convergence Check:** $O(n^2)$.

Thus, the cost per iteration is $O(n^3)$.

5. Convergence

The QR algorithm typically requires $O(n)$ iterations to converge for well-conditioned matrices. In the worst case, convergence can take up to $O(n^2)$ iterations for highly ill-conditioned matrices.

Overall Complexity

Combining the above factors:

1. Per iteration cost: $O(n^3)$.
2. Number of iterations: $O(n)$ (best case) to $O(n^2)$ (worst case).

5 Advantages of QR Algorithm

1. All eigen values(complex and real) can be computed.
2. Maintains accuracy and robustness
3. Supports complex matrices, making it versatile. .
4. Efficient for small to dense matrices
5. Effective stopping criteria ensure convergence to eigenvalues.
6. High accuracy with customizable tolerance levels.

6 Comparison with other Algorithms

This section compares the QR Algorithm with other Algorithms.

Advantages of QR Algorithm over others

1. Power Iteration method Finds only the largest eigenvalue.
2. It Converges slowly, especially when eigenvalues are close in magnitude.
3. Inverse Iteration Requires solving a linear system at each iteration, which increases Complexity.
4. Inverse Iteration highjly depends on shift selection; poor shifts may lead to divergence.
5. Rayleigh Quotient Iteration (RQI) is sensitive to the choice of the initial guess.
6. It focuses on a single eigenvalue at a time.
7. Matrix inversion incurs high computational cost.

QR Algorithm

Advantages:

1. Computes all eigenvalues simultaneously.
2. Robust for various matrix types, including symmetric, non-symmetric, and complex matrices.
3. Handles both real and complex eigenvalues automatically.
4. Convergence is linear to quadratic, depending on implementation.
5. Efficient for small to medium dense matrices.
6. Householder reflections and orthogonalization techniques improve numerical stability.

7. Requires no initial guess or manual intervention.

Limitations:

1. Computational cost is $O(n^3)$, which can be prohibitive for very large matrices.
2. Requires more memory than Power Iteration and Inverse Iteration.

Why QR Algorithm is the Best?

The QR Algorithm is the most versatile and robust choice for eigenvalue computation due to its ability to compute all eigenvalues in a single run, handle complex matrices, and ensure numerical stability. While other methods excel in specific scenarios (e.g., Rayleigh Quotient Iteration for single eigenvalue refinement), the QR Algorithm is a universal approach suitable for general eigenvalue problems, particularly for small to medium-sized dense matrices.

7 Description of the Program

The program is implemented in the C programming language. It operates on matrices through a custom structure that stores the dimensions of the matrix and a pointer to its elements. This design facilitates flexible and efficient handling of matrix operations.

7.1 Matrix Operations

All matrix-related operations are implemented as standalone functions, ensuring reusability. These functions include basic operations such as matrix multiplication, transposition, and inverse, as well as more specialized routines.

7.2 QR Decomposition Using Householder Reflections

A function decomposes the input matrix into Q and R matrices using the method of Householder reflections. This step is crucial for performing the QR algorithm.

7.3 Iterative Eigenvalue Computation

The main function iterates continuously, applying the QR decomposition until the matrix converges to an upper triangular form. In the case of triangular matrices that converge partially (some elements below diagnol will not be zero), which occur for some complex eigenvalue scenarios, the eigenvalues are calculated using the `printeigenvalues` function.

7.4 Program Features

1. The program allows the user to specify the tolerance value for convergence and the maximum number of iterations.
2. Eigenvalues, including complex ones, are computed and displayed based on the matrix's final form.

References

1. Wikipedia
2. Introduction to Applied Linear Algebra – Vectors, Matrices, and Least Squares Stephen Boyd and Lieven Vandenberghe.
3. Other online Sources like Youtube, Google etc.