MATRIX THEORY - EE1030 Software Assignment

ee24btech11058 - P.Shiny Diavajna

Aim : To compute the eigenvalues of a given matrix using C code.

Chosen Algorithm:

QR Algorithm with Gram Schmidt Orthogonalization

- The QR algorithm is an iterative method for calculating eigenvalues of a matrix. It repeatedly factors a given matrix A into a product of an orthogonal matrix Q and an upper triangular matrix R, then updates A as $A = RQ$. Over iterations, A converges to a quasi-diagonal matrix, where the diagonal elements approximate the eigenvalues.
- **Gram-Schmidt Orthogonalization** is used to compute the orthogonal matrix Q during QR factorization. It ensures numerical stability by orthogonalizing the columns of A explicitly.

Time Complexity:

- **QR Factorization with Gram-Schmidt:** For an $n \times n$matrix, the Gram-Schmidt process has a complexity of $O(n^3)$.
- **Overall QR Algorithm :** Since the QR factorization is performed iteratively ,the total complexity depends on the number of iterations k. For k iterations, the complexity is $O(kn^3)$. For matrices that converge quickly (e.g., symmetric matrices), k is often small.

Other Aspects :

1) **Memory Usage:**The QR algorithm requires storing the matrices Q and R, making memory usage approximately $O(n^2)$.

2) **Convergence Rate :** The convergence of the QR algorithm depends on the type of matrix
   - The algorithm converges rapidly , often quadratically, without additional enhancements.
   - Convergence can be slower and may require shifts or deflation techniques to accelerate. The rate also depends on how well-separated the eigenvalues requiring more iterations.

3) **Suitability:**
   - **Symmetric Matrices:** Excellent due to faster convergence and numerical stability.
   - **Sparse Matrices:** the algorithm may not preserve sparsity, leading to increased computational costs.
   - **Large Matrices:** High computational cost per iteration makes it less practical for very large matrices. However,it can handle moderately large matrices effectively.

Comparison with other algorithms:

Algorithm	**Time Complexity**	**Accuracy**	**Suitability**
QR Algorithm	$O(kn^3)$	High for all matrices	Best for dense, symmetric matrices
Power Iteration	$O(n^2k)$	Good for dominant eigenvalue	Best for largest eigenvalue only
Jacobi Method	$O(n^3 \log n)$	High for symmetric matrices	Best for symmetric matrices
Lanczos Method	$O(kn^2)$	Moderate	Best for sparse symmetric matrices
Divide and Conquer	$O(n^3)$	High	Suitable for symmetric matrices

Conclusion:

The QR algorithm with Gram-Schmidt orthogonalization is a robust and accurate method for eigenvalue computation, particularly for symmetric and dense matrices. However, its $O(kn^3)$ complexity makes it less efficient for very large or sparse matrices, where specialized methods like Lanczos are more suitable. For practical applications, matrix properties and computational resources dictate the choice of algorithm.