

Lab Session 8

Instruction: Write a report for the question 2 in a pdf file.

1. Write a function program `[iter, lambda] = Powermethod(A, x, k)` that performs `k` iterations of the Power Method with $A \in \mathbb{C}^{n \times n}$ and initial vector $x \in \mathbb{C}^n$ and returns an $n \times k$ matrix `iter` whose j th column is the j th iterate q_j and a scalar `lambda` which is the dominant eigenvalue.
2. Run `[iter, lambda] = Powermethod(A, x, k)` with $x = [1 \ 1 \ 1]^T$ and the following matrices

$$(i) A = \begin{bmatrix} 1 & 1 & 1 \\ -1 & 9 & 2 \\ 0 & -1 & 2 \end{bmatrix} \quad (ii) A = \begin{bmatrix} 1 & 1 & 1 \\ -1 & 9 & 2 \\ -4 & -1 & 2 \end{bmatrix} \quad (iii) A = \begin{bmatrix} 1 & 1 & 1 \\ -1 & 3 & 2 \\ -4 & -1 & 2 \end{bmatrix}$$

In each case check if for large enough values of j , $\|\text{iter}(:, j+1) - v\| / \|\text{iter}(:, j) - v\|$ agrees with the theoretical convergence rate of $|\lambda_2|/|\lambda_1|$ where λ_1 and λ_2 are the largest and second largest eigenvalues of A in magnitude and v is an eigenvector corresponding to λ_1 . Try to explain your observations. (Type `format long e` to see more digits and use the `[V, D] = eig(A)` command to find λ_1, λ_2 and v .)

3. Write a function program `[iter, lambda] = Shiftinv(A, x, s, k)` that *efficiently* performs `k` iterations of Shift and Invert Method using $A \in \mathbb{C}^{n \times n}$, $x \in \mathbb{C}^n$ and shift `s` and returns an $n \times k$ matrix `iter` whose j th column is the j th iterate q_j and a scalar `lambda` is the the eigenvalue of A closest to `s`.
4. Write a function program `[iter, lambda] = Rayleigh(A, x, k)` that *efficiently* performs `k` iterations of inverse iterations with Rayleigh quotient shifts using $A \in \mathbb{C}^{n \times n}$ and $x \in \mathbb{C}^n$ and returns an $n \times k$ matrix `iter` whose j th column is the j th iterate q_j and a scalar `lambda` is the eigenvalue of A to which the Rayleigh quotient shifts converge.

Note. Use `[Q, H] = hess(A)` to find an upper Hessenberg matrix H and a unitary matrix Q such that $Q^* A Q = H$ and use H in place of A in the iterations. This will reduce the flop count in each iteration from $O(n^3)$ to $O(n^2)$. However, if the program will converges, it will be to an eigenvector v corresponding to H from which an eigenvector corresponding to A will have to be found by computing Qv .

5. Write a function program `B = FrancisQRS(A)` to perform a *single iteration* of Francis's single shift (or degree one) Implicit QR algorithm via Householder reflectors on an upper Hessenberg matrix A .