Alakbarzade Yusif - 317297
Numerical Methods 2 - Mini-Project Task 2
Project task 10 **- Finding all roots of a polynomial**

Program Description

Program consists of 2 files: findRoots.m and test.m. Code defines a function in MATLAB called "findRoots" which takes a single input argument "w", which is assumed to be a vector representing the coefficients of a polynomial. The function finds the roots of the polynomial represented by the input vector "w".

The function first calls another function called "findRootsPart" which is defined within the function, passing it the input vector.

The first line of the function calculates the degree of the polynomial, which is equal to the length of the input vector minus one. If the polynomial has 2 non-zero coefficients, it then calls another function called "findTwoNonZeroValues" to find the position of the two non-zero coefficients. Then it uses these positions to calculate the roots of the polynomial and return them.

If the degree of the polynomial is greater than one, the function proceeds to create a companion matrix, C, that is used in the power method. The companion matrix is a square matrix with ones along the sub-diagonal and the negative of the coefficients of the polynomial in the first row, divided by the first coefficient of the polynomial.

The function then uses a random initial vector x and performs the power method to find the largest eigenvalue of the companion matrix. The power method is an iterative algorithm that calculates the eigenvalue of the matrix by repeatedly multiplying the matrix with an initial vector. The loop continues until the difference between the current eigenvalue and the next eigenvalue is smaller than a specified tolerance or the maximum number of iterations is reached.

Once the eigenvalue is found, the function uses the vector x that is corresponding to the eigenvalue to find the roots of the polynomial. The function then calls a subfunction, 'polydeflate' which deflates the polynomial using Horner method by subtracting the roots from the polynomial. After this, the function checks if the deflated polynomial has any other roots and if there are any it calls the 'findRootsPart' function again with the deflated polynomial as an input.

Finally, the function returns the roots of the polynomial in a vector and removes similar values with specified tolerance.

The code is using the eigenvalues of the companion matrix to approximate the roots of the polynomial. However, this method may fail to find all roots or may return approximate values for certain roots, especially for polynomials with complex roots or multiple roots

It's also worth noting that there are other methods to find the roots of a polynomial, such as the bisection method, the Newton-Raphson method, or the Durand–Kerner method. These methods are more robust and can handle a wider range of polynomials.

Numerical Examples

We got 7 functions

1. 1x^3+3x^2-4x = 0
2. 1x^2-9 = 0
3. $1x\hat{} - 37x\hat{}5 + 337x\hat{}4 - 331x\hat{}3 - 7706x\hat{}2 + 31400x - 33600 = 0$
4. 2x^3+4x^2-30x = 0
5. 1x^2+5x-6 = 0
6. 1x^6-21x^5+175x^4-735x^3+1624x^2-1764x+720 = 0
7. 3x^2-27 = 0

I chose this function because it was different to show that the program works.

Numerical Tests

We got many inputs to test the correctness of the function.
I tested the functions with matlab's built-in roots function.

Functions	Roots of My function	Built-in roots function	Error
1x^3 + 3x^2 − 4x = 0	x1= -4 x2= 3.2e-09 x3= 1	x1= -4 x2= 0 x3= 1	x1= 0 x2= 3.2e-09 x3= 0
1x^2-9 = 0	x1= -3 x2= 3	x1= -3 x2= 3	x1= 0 x2= 0
1x^ − 37x^5 + 337x^4 − 331x^3 − 7706x^2 + 31400x − 33600 =	x1= -5 x2= 2 x3= 4 x4= 5 x5= 7 x6= 24	x1= -5 x2= 2 x3= 4 x4= 5 x5= 7 x6= 24	x1= 0 x2= 0 x3= 0 x4= 0 x5= 0 x6= 0
2x^3+4x^2-30x = 0	x1= -5 x2= 2.51e-09 x3= 3	x1= -5 x2= 0 x3= 3	x1= 0 x2= 2.51e-09 x3= 3
1x^2+5x-6 = 0	x1= -6 x2= 1	x1= -6 x2= 1	x1= 0 x2= 0
1x^6-21x^5+175x^4-735x^3+1624x^2-1764x+720 = 0	x1= 1 x2= 2 x3= 3 x4= 4 x5= 5 x6= 6	x1= 1 x2= 2 x3= 3 x4= 4 x5= 5 x6= 6	x1= 0 x2= 0 x3= 0 x4= 0 x5= 0 x6= 0
3x^2-27 = 0	x1= -3 x2= 3	x1= -3 x2= 3	x1= -0 x2= 0

$$E = |\text{Built-in roots - My function}|$$

As shown on the table the results of a given roots are correct in most cases.To conclude we can say that the function works good for finding the roots of a given polynomial.