CE 33 A and B
2nd Semester, 2019 - 2020

Project 1: Roots of Polynomials

Project due: March 16, 2020 (Monday)
Project documents due: March 20, 2020 (Friday, p.m. session)

1. Specifications

Your main objective is to implement algorithms for finding all roots of polynomials with real-valued coefficients.

2. Detailed Specifications

Develop an application that obtains all the roots of a polynomial whose coefficients are given in a file specified by the user.

The application obtains information about the polynomial from a file specified by the user. The filename may be specified on the command line when the function is invoked from the command line. If no file is specified on the command line, the application asks for the filename.

E.g. Assuming the name of the application is *polyroot.exe*, the following options should be available:

Here the file containing data is specified when the application is invoked:

```
(command prompt)> polyroot polytest.txt
```

If no file is specified, the application asks for the filename:

```
(command prompt)> polyroot
filename: polytest.txt
```

Print all results according to their accuracy. If a root is accurate up to 10 significant figures, print it showing 10 significant figures.

The format of files encoding polynomials is as follows:

N a0 a1 a2 ... aN misc_info

with *N* a positive integer specifying the degree of the polynomial, *a0* the constant coefficient, *a1* the coefficient of the 1st power of the variable, *a2* the coefficient of the 2nd power of the variable, and so on up to *aN* , the coefficient of the *N*th power of the variable. *misc_info* is any information that may come after the coefficients and may just be ignored by the root finding algorithm. A user may put in such information merely to document where the polynomial originated, for instance. Items in the file are separated from each other by one or more white spaces.

The application must echo back to the user the polynomial coefficients obtained from the file.

Once all the roots have been determined, the applications lists all roots found as a list of complex numbers, with each root on a separate line. Show all roots in the form
(*real part*, *imaginary part*).

After reporting the roots, the application shows the results of evaluating the polynomial at each of the roots identified. Do NOT use `pow()` in evaluating the polynomial. Show the results of evaluating each claimed root on a separate line. Display the results using scientific notation.

Example:
The text file *polytest.txt* may contain the following:

"5 1.25 -3.875 2.125 2.75 -3.5 1
This is the 5th degree polynomial x^5 - 3.5 x^4 + 2.75 x^3 + 2.125 x^2 - 3.875x +1.25
roots are 2, -1, 1 +/- 0.5, 0.5 "

Assuming the application provides 6 significant figure accuracy in its output and works perfectly well the output produced should be

```
Polynomial:
1.00000 x^5
- 3.50000 x^4
2.75000 x^3
2.12500 x^2
- 3.87500 x^1
1.25 x^0

Roots:
(2.00000, 0.00000)
(-1.00000, 0.00000)
(1.00000, 0.500000)
(1.00000, -0.500000)
(0.500000, 0.00000)

Evaluating polynomial at identified roots:
f (2.00000, 0.00000) = 0.000000e+000
f (-1.00000, 0.00000) = 0.000000e+000
f (1.00000, 0.500000) = 0.000000e+000
f (1.00000, -0.500000) = 0.000000e+000
f (0.500000, 0.00000) = 0.000000e+000
```

This project must be done in C++.

3. Submission of Requirements

Project due: start of class, March 16, 2020 (Monday)
Submit the following:

Project CD as described in the syllabus (submitted by group)

Project documents due: start of class, March 20, 2020 (Friday, p.m. session)
Submit the following:

1. Hard copies of all source code, printed on A4 bond paper (submitted by group)
2. A filled up Peer Evaluation Form for Group Projects (submitted individually)
3. Project documentation (prepared and submitted individually)

Requirements that are not ready for submission at the start of class are subject to penalties. A 20-pt penalty applies to projects that are not submitted at the start of class. A separate 20-pt penalty is imposed on projects where documentation is not submitted promptly.

Project documentation are prepared and submitted individually. Project documentation that are unreasonably similar to each other shall be considered as a single submission, with the points earned from that submission divided among the number of students who submitted similar reports.

These requirements must be submitted directly to the course instructor.

4. Documentation

Project documentation should specifically target the most significant functions implementing numerical methods for determining roots of polynomials and verifying results.

Include a diagram or flow chart showing the sequence in which functions are called and showing what functions are called upon by other functions.

For each significant function, document the action of the function on its parameters and/or on data of the class to which the function belongs. Relate the function to a description of the process being implemented and show that the function is an efficient implementation. Include a description of how the function uses results from other functions, how the results of a function are subsequently used, and what other functions are called upon by a function.

Show the results of testing your work on test files provided.

5. Project Defense

A project defense may be required on a case-to-case basis.