

ICOM 4036 – Programming Languages

Programming Assignment 1¹

Due on Thursday September 25, 2003 by 1:30 PM

Objectives

- Gain experience with using selection structures and repetitive structures in FORTRAN
- Gain experience with file processing and user defined formatting in FORTRAN 77
- Gain experience with FUNCTION subprograms in FORTRAN
- Gain experience in program development following programming standards
- Gain an appreciation of the virtues and limitations of FORTRAN as a programming languages compared to more modern languages.

Problem Statement

In many applications, it is necessary to find a **zero** or **root** of a function f , that is, to solve the equation $f(x) = 0$. For some functions, it may be very difficult or even impossible to find this solution exactly. For such functions, an iterative numerical method may be used to find an approximate zero. One method of finding an approximate zero of a function is called **bisection method**. In this method, we begin with two numbers a and b where the function values $f(a)$ and $f(b)$ have opposite signs. If f is continuous between $x = a$ and $x = b$, that is, if there is no break in the graph of $y = f(x)$ between these two values, then the graph of f must cross the x -axis at least once between $x = a$ and $x = b$, and thus there must be at least one solution of the equation $f(x) = 0$ between a and b . To locate one of these solutions, we first bisect the interval $[a, b]$ and determine in which half f changes sign, thereby locating a smaller subinterval containing a solution of the equation. We bisect this subinterval and determine in which half f changes sign; this gives a still smaller subinterval containing a solution. Repeating this process yields a sequence of subintervals, each of which contains a solution of the equation and has a length one-half that of the preceding interval. Note that at each step, the midpoint of a subinterval of length L is within $L/2$ of the exact solution. This value or epsilon should be used as criteria for stopping the iteration when the solution is precise enough since the iteration may go on infinitely, a maximum number of iterations can also be specified.

Write a program to implement the **bisection method**, and use it to find a solution of the equation $x^3 + x - 5 = 0$. While computing the root of the function $x^3 + x - 5 = 0$, write intermediate results to an output file named "youraccount-pa1.out", which should be as similar as possible to the following **sample output**:

¹ This programming assignment was originally used for a Computer Science course at Mississippi State University taught by Maria A. Teran

N	A (N)	F (A (N))	B (N)	F (B (N))	M (N)	F (M (N))
0	-3.00000	-0.35000E+02	3.00000	0.25000E+02	0.00000	-0.50000E+01
1	-3.00000	-0.35000E+02	3.00000	0.25000E+02	0.00000	-0.50000E+01
2	0.00000	-0.50000E+01	3.00000	0.25000E+02	1.50000	-0.12500E+00
3	1.50000	-0.12500E+00	3.00000	0.25000E+02	2.25000	0.86406E+01
4	1.50000	-0.12500E+00	2.25000	0.86406E+01	1.87500	0.34668E+01
5	1.50000	-0.12500E+00	1.87500	0.34668E+01	1.68750	0.14929E+01
6	1.50000	-0.12500E+00	1.68750	0.14929E+01	1.59375	0.64194E+00
7	1.50000	-0.12500E+00	1.59375	0.64194E+00	1.54688	0.24827E+00
8	1.50000	-0.12500E+00	1.54688	0.24827E+00	1.52344	0.59125E-01
9	1.50000	-0.12500E+00	1.52344	0.59125E-01	1.51172	-0.33560E-01
10	1.51172	-0.33560E-01	1.52344	0.59125E-01	1.51758	0.12627E-01
11	1.51172	-0.33560E-01	1.51758	0.12627E-01	1.51465	-0.10506E-01
12	1.51465	-0.10506E-01	1.51758	0.12627E-01	1.51611	0.10505E-02
13	1.51465	-0.10506E-01	1.51611	0.10505E-02	1.51538	-0.47302E-02
14	1.51538	-0.47302E-02	1.51611	0.10505E-02	1.51575	-0.18406E-02
15	1.51575	-0.18406E-02	1.51611	0.10505E-02	1.51593	-0.39530E-03
16	1.51593	-0.39530E-03	1.51611	0.10505E-02	1.51602	0.32806E-03
17	1.51593	-0.39530E-03	1.51602	0.32806E-03	1.51598	-0.33379E-04

Where,

N stands for the number of iteration,

A(N) stands for the value of a (lower end of the interval) at iteration N ,

F(A(N)) stands for the value of $f(a)$ at iteration N ,

B(N) stands for the value of b (higher end of the interval) at iteration N ,

F(B(N)) stands for the value of $f(b)$ at iteration N ,

M(N) stands for the value of the midpoint m between interval a and b at iteration N ,

F(M(N)) stands for the value of $f(m)$ at iteration N , $m = (a + b)/2$.

The above results is based on $\epsilon = 1E-4$, maximum iteration $\# = 20$, initial $a = -3$, initial $b = +3$. For meanings of these input variables, see **program requirements** below. Please note that a , b , and m change from one iteration to another as $f(m)$ approaches to zero.

Required Program Skeleton

A template for your program will be available at the course website. Your program must include the PROGRAM statement, followed by a block of comments in the form shown in the template. Include all items indicated. The **Program Purpose** should be a short phrase that summarizes the purpose of the program. The **Program Description** should include more details about how the program accomplishes its goal.

Additional Program Requirements

- Name your source code file **youraccount-pal.f**
- Document your code well using comments (begin with * in column 1).
- Use meaningful variable names
- Use **DATA** statement to initialize variables

- Use **INQUIRE** to test for the existence of the output data file “youraccount-pa1.out” before deciding to create or overwrite the output file.
- Must use **FORMAT** statements to print the output table. Use list-directed input statements (READ) to read in values from user’s input.
- Your program must generate an output file that contains the table with the name "youraccount-pa1.out", whose **format** (not necessary its content) should look **exactly** as the **sample output**.
- Use **DO WHILE** loop as the iterative control structure, stop the loop if **any** of the following two condition is true:
 - 1. The **absolute** value of $f(m)$ as shown in the **F(M(N))** column in the **sample output**, is less than **epsilon**, which is specified by the user.
 - 2. The number of iteration exceeds the **maximum number of iterations**, which is also specified by the user
- Use a **FUNCTION** subprogram to compute the value of $f(x)$.
- Use a **FUNCTION** subprogram to compute the root of $f(x)$ between a and b for any argument function f. The function may assume that a root exists.
- Briefly describe the purpose of the function at the beginning of each **FUNCTION** statement.
- Ask the user to input **epsilon**, **maximum number of iterations**, **initial a** and **initial b**. Display **initial a** and **initial b** as **A(N)** and **B(N)** at iteration 0.

Software Development Platform

You will carry out all your software development using the tools available at the Linux Academic Computing Lab (Amadeus). You may work on your personal Linux-based PC, but you must make sure that your code works with the GNU g77 compiler version available at Amadeus as this is the compiler that the staff will use to grade your assignment.

Remember from the prontuario that your programming assignments will be graded according to the following late penalty policy:

Days Late	Percent Deduction
1 day late	25%
2 days late	50%
3 days late	100%

Programming assignments will be graded for both correctness and quality according to the following weights:

Criteria	Weight (%)
Correctness	60%
Design	20%
Efficiency	10%
Style & Documentation	10%