Computer Programming & Numerical Analysis

Version 2.01
Last modified on April 25, 2021

Contents

Note to the Reader **2**

1 Introduction **5**
1.1 General Introduction . . . 5
1.1.1 Operating System . . . 5
1.1.1.1 Linux Operating System . . . 6
1.1.2 Compiler . . . 9
1.1.3 Editors . . . 10
1.1.3.1 Edit/Write your own program . . . 10
1.1.4 Programming Language . . . 11
1.2 Introduction to GNU C . . . 11
1.2.1 Create your first C program . . . 11
1.2.2 Compile/Run/Execute your first C program . . . 12
1.2.2.1 Run your first C program . . . 12
1.2.3 Variables in C . . . 17
1.2.4 Loops in C . . . 20
1.3 Functions in C . . . 24
1.3.1 Decision Making in C . . . 29
1.3.2 Function Prototypes . . . 35
1.4 Operators in C . . . 36
1.4.1 Arithmetic Operators . . . 37
1.4.2 Assignment Operators . . . 37
1.4.3 Relational Operators . . . 38
1.4.4 Logical Operators . . . 38
1.5 Mathematical Functions in C . . . 39
1.6 Storing the results of the program in a data file . . . 40
1.7 Examples . . . 41
1.8 Advanced Topic: Pointers . . . 47
1.9 QUESTIONS . . . 51
1.10 PROBLEMS . . . 51

2 Graphics Using GNUPLOT **53**
2.1 Introduction . . . 53
2.2 Plotting with inbuilt functions of GNUPLOT . . . 53
2.2.1 Interactive plotting . . . 54
2.3 Saving Plots . . . 56
2.4 Plotting using script . . . 56
2.4.1 Customization . . . 57
2.5 Plotting using data from a file . . . 58
2.6 Periodic Function . . . 60
2.7 PROBLEMS . . . 63

3 Finite & Infinite Series **66**
3.1 Introduction . . . 66
3.2 Finite Series . . . 66
3.3 Infinite Series . . . 70
3.4 Questions . . . 75
3.5 Problems . . . 76

4 Root Finding **78**
4.1 Introduction . . . 78
4.2 Bisection Method . . . 79
4.3 Secant Method . . . 82
4.4 Newton-Raphson Method . . . 84
4.5 Questions . . . 87
4.6 Problems . . . 87

5 Ordinary Differential Equations **90**
5.1 Introduction . . . 90
5.2 Euler's Formula . . . 91
5.3 Runge-Kutta methods . . . 91
5.4 Simultaneous Equations of First-Order . . . 95
5.5 Problems . . . 99

6 Integration **102**
6.1 Introduction . . . 102
6.2 Methods Based on Intervals of Equal Width . . . 102
6.2.1 Trapezoidal Rule . . . 102
6.2.2 Simpson's Rule . . . 104
6.3 Methods Based on Intervals of Unequal Width . . . 106
6.3.1 Gauss Quadrature . . . 106
6.3.2 Gauss-Laguerre Quadrature . . . 109
6.3.3 Gauss-Hermite Quadrature . . . 109
6.3.4 General Considerations & Programming Tips . . . 109
6.4 Problems . . . 112

7 Matrices **114**
7.1 Introduction . . . 114
7.2 Arrays in C . . . 114
7.2.1 Declaration of Arrays . . . 115
7.2.2 Initializing & Using Arrays . . . 115
7.3 The Function matalloc . . . 118
7.4 Problems . . . 121

A Note to the Reader

This Manual is intended for use in the Computer Programming & Numerical Analysis class. **The Manual does not assume any familiarity with either the C programming language which is used or the techniques of numerical analysis**. However, if the reader is familiar with these then it would be of help.

The book is organised into 7 Chapters and they are best gone through in sequence. **Chapter 1** is an introduction to programming in general and to the C programming language in particular. This Chapter may be skipped by readers who already are familiar with the language though it may help to go through it once to familiarize themselves with the notation etc. It also has some sample programs to illustrate the use of various C progamming constructs like loops and conditional statements etc.

Chapter 2 is an introduction to **Gnuplot**, an open source graphics package which we use to generate graphics in this course. Once again it is self contained and does not assume any previous familiarity with **Gnuplot**.

Chapter 3 uses the programming and graphics fundamentals from the earlier Chapters to discuss techniques to numerically find the sum of finite and infinite series.

Chapter 4 discusses various numerical methods to find roots of equations. The methods discussed are the Bisection method, Secant method and the Newton-Raphson method.

Chapter 5 is an introduction to numerically solving ordinary differential equations using Euler's formula and 4^{th} order Runge-Kutta method.

Chapter 6 discusses integration by various methods. Methods using intervals of equal and unequal width are discussed here. The Trapezoidal and Simpson's rule are discussed as examples of methods using intervals of equal width and Gauss-Legendre, Gauss-Hermite and Gauss-Lagauerre quadratures as examples of methods which use unequal intervals.

Chapter 7 is a short discussion of matrices and how to manipulate and use them in C. Here we use arrays and pointers to do various matrix operations.

Each Chapter has several sample programs so that the reader can get an idea of how to write his/her own programs. In addition, each Chapter has some questions and problems at the end which the reader

should certainly attempt to answer and solve.

For students who own computers at home, the chances are that you are using a Windows OS based machine. This programming language used in this course is C and the compiler used is **gcc** which comes with a Linux OS. In addition, the graphics package Gnuplot also comes with the Linux OS. For Windows users, there is a simple way to emulate a Linux environment.

Go to **https://www.cygwin.com/** and download and install the program Cygwin. This will come in two versions- Cygwin32 and Cygwin64. If you have a 64-bit computer then install Cygwin64, else install Cygwin32. Once installed, run the program as you run any windows program, that is by double clicking on the icon. This will open a small window. In this type **startx**. This will open a terminal on your screen, exactly like the one you see in the laboratory on the Linux machines. It is as if, your Windows machine has turned into a Linux machine. You can run all the Linux OS programs like **Gnuplot, gcc, emacs** etc. in this terminal. Once you are finished, just type exit and you will return to your Windows environment.

We would very much like to get your suggestions regarding how to improve this Manual. In addition, if there are any errors or misprints that are spotted in the Manual, we would like to hear from you. Please send a mail with the suggestions/errors etc. to **shobhit.mahajan@gmail.com** making sure you quote the version number of the Manual as well as the Modification date of the Manual you are using. The version number and date are on the title page of the Manual.

Finally, programming is like cycling or swimming- no amount of reading can ever teach you how to cycle or swim. The only way to learn is to actually do it. Yes, you will lose balance and fall initially but ultimately you will learn how to ride a bicycle. It is the same with programming- unless you actually write your programs, you will never ever learn how to program. Here too, you will initially make mistakes which later on you will think are silly. But everyone who learns programming does this. So there is nothing to be ashamed or scared of. Just start writing and running your own programs and you will see how quickly you will become good at it. The important part will be to develop the ability to debug a program- that is to say, when a program has errors or is not doing what you intended it to do, then to be able to spot the reason for it. This is what we hope you will learn after this course.