Eastern Economy Edition

Numerical Methods with Computer Programs in C++

Pallab Ghosh

PHI

CD INCLUDED

Numerical Methods with Computer Programs in C++

PALLAB GHOSH
Assistant Professor
Department of Chemical Engineering
Indian Institute of Technology Guwahati

PHI Learning Private Limited
Delhi-110 001
2009

NUMERICAL METHODS WITH COMPUTER PROGRAMS IN C++ (with CD-ROM)
Pallab Ghosh

ISBN-978-81-203-2987-4

Third Printing ... ... **August, 2009**

Published by Asoke K. Ghosh, PHI Learning Private Limited, Rimjhim House, 111, Patparganj Industrial Estate, Delhi-110092 and Printed by Rajkamal Electric Press, Plot No. 2, Phase IV, HSIDC, Kundli-131028, Sonepat, Haryana.

Contents

Preface

This book on *Numerical Methods with Computer Programs in C++* takes into account the tremendous role that computers play to help solve scientific and engineering problems. Apart from their enormous speed, they ensure accuracy, finesse and versatility.

There are many books available in India on numerical methods. These books give a fair idea about numerical methods and their application to problems. So the question obviously arises: Do we need yet another book on the subject? When I told some of my colleagues of my plan to write a book on Numerical Methods, they exclaimed: What! Another Book! I firmly believe that a book on Numerical Methods, which provides computer programs and clearly demonstrates how scientific and engineering problems can be solved, is certainly needed.

A variety of reasons exist for such a line of thinking. Chief among them are that, in recent years, C++ has been the choice of the software developers because of its many attractive features, and many science and engineering students after their graduation choose a career in software development.

There are very few books on numerical methods available in the market, which have used this powerful language. This book presents a small tutorial on C++ in the first chapter. Nowadays, C programming language is taught at the school level. In most engineering institutes, advanced C and C++ are taught to the fresh engineering students. Therefore, it is expected that the language will not be a barrier in the computer implementation of the numerical methods.

Today, a course on numerical methods is typically composed of two or three hours of lecture and a three-hour computational laboratory session per week. The laboratory session helps the students to apply the algorithms on a computer. This book is designed to cover such a curriculum. It is primarily intended to be used in an undergraduate course in engineering. It may not be possible to cover all the 15 chapters in one semester. The instructor may use part of them in one semester depending on the requirements of the students who take the course. This book will also be helpful to professionals who need computer implementation of the numerical methods.

The subject comprises three main parts: (i) mathematical foundation (which involves theorems and their corollaries); (ii) scientific and engineering applications of the methods; and (iii) computer implementation. All three parts are rarely included in a single book covering the

entire undergraduate syllabus. There are many good books available in the market, which present the theories involved in numerical techniques. This book presents numerical techniques from an **application perspective.** Simple problems are solved manually, but the emphasis is on the use of computers. Many solved problems — manual as well as computer-oriented — are presented in all chapters to facilitate understanding of the concepts. The problems given at the end of every chapter are expected to provide a good practice for the student. The books listed under *Suggested Further Reading* after every chapter should be consulted for more detailed information on the topics covered in that chapter. References to many good books are provided at the end of the book. Students are advised to consult them.

After learning how a program works and the methods to develop such a program, you will be in a strong position to understand how a numerical algorithm works. Probably you can even design a smarter program at this point. Shortly after this, you can begin to use commercial software such as MATLAB, Mathematica, Polymath, NAG and IMSL routines.

I wish to thank the Director and Deputy Director of IIT Guwahati for their encouragement and for allowing me in bringing out this book. My students always wanted a book with computer programs that work. They have given valuable feedback and suggestions for which I am grateful. I was encouraged throughout the preparation of this book by my sister Kakali and my parents. My colleagues have been very enthusiastic, kind and cooperative. I am indebted to all of them. My special thanks are due to my former colleague, Dr. Anupam Shukla who suggested me to write a book on the subject. Finally, I wish to thank the publishers, PHI Learning, in particular, the management as well as the editorial and production team, for their fine collaboration in bringing out this book.

If this book rouses any interest in you on numerical methods and their computer implementation, I will be most happy. Any constructive comments for improving the contents will be warmly welcomed.

The **CD-ROM** accompanying this book contains many programs, all written in simple C++ language.

A CD-ROM containing the **Solutions Manual** is separately available. The instructors may obtain it from the publishers.

Pallab Ghosh
IIT Guwahati

Chapter 1

C++ and Object-Oriented Programming

Charles Babbage

Charles Babbage was born in London in 1791. He had a passion for Mathematics since his childhood. Babbage originated the concept of modern computer. At Cambridge University, he dreamt of logarithms calculated by a machine. In 1822, he developed a small **Difference Engine.** This machine performed computation only through addition. For this invention, Babbage received a gold medal from the Royal Astronomical Society, London. He received grants for further development of his machine. Babbage prepared the drawings of his **Analytical Engine** in 1834. This was more sophisticated than his Difference Engine and quite similar to the modern computer. However, he never built it. Babbage died in London in 1871.

CONTENTS

1.1 INTRODUCTION

The C++ language is a better version of the C language. One of its main features is that it supports object-oriented programming (OOP). However, C++ is not the only language that supports OOP. Languages like Smalltalk, Ruby and Eiffel also support OPP. But, C++ has become more popular than all of these languages for a variety of reasons. When C++ was created, it was looked upon as a 'superset of C'. There was a very big group of users of C who wrote programs for a variety of applications. Most of these programmers shifted to C++, taking advantage of its advanced and easy-to-use features. Traditionally, programs on engineering applications were written in FORTRAN. However, C++ offered the object-oriented approach and easy portability of programs. In addition, there were features for generating attractive graphics. This made software development a fun.

C++ is used for scientific computing as well as system programming. Nowadays, very good compilers are available, such as Microsoft, Borland, GNU and Intel C++ compilers. Programmers find these as exciting tools for developing software. The programs presented in this book were developed using the Microsoft Visual C++ compiler (version 6.0) running on Windows (98, 2000 and XP) operating systems. However, all these programs will also run on Unix with minor modifications.

C++ is a vast language. It has evolved over C which is itself immense in contents. We will not attempt to make you learn the entire C++ language in this chapter. This chapter is an introduction to C++ for the engineer who intends to write programs on numerical methods using this language. It is not a substitute for the numerous good books available on C++. Some of these books are listed at the end of this chapter under "Further Reading". Here, a part of the language is introduced and illustrated with examples. This will enable the reader to understand all programs presented in this book. A preliminary knowledge of the computer is assumed, i.e., the reader can start the environment of the development software, compile and run a program using the menus of the software. Knowledge of C language is not a prerequisite to learn C++. However, if the reader already knows C, he will find many familiar terms in C++.

1.2 ELEMENTS OF A C++ PROGRAM

An object-oriented C++ program will have a user-defined class (where data and member functions are placed), object(s) of the class, and a main function. Here is our first C++ program.

PROGRAM P1.1

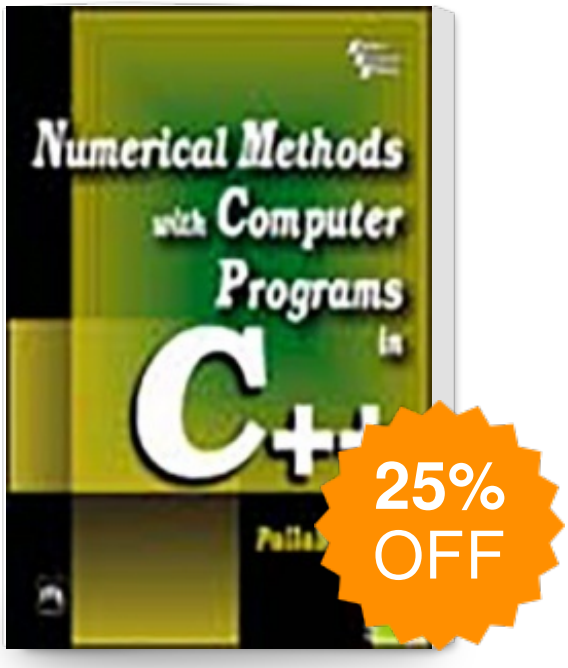
```
//P1_1.CPP

# include <iostream.h>
# include <math.h>

class Program1
{
  private:
        double a, result;
```

```
  public:
        void square_root ()
        {
              cout<<"\nEnter a number ";
              cin>>a;
              result = sqrt (a);
              cout<<"\nSquare root of the number is
              "<<result<<endl;
        }
};

//main function

void main ()
{
  Program1 P1;

  P1.square_root ();
}
```

The program starts with the line, //P1_1.CPP. The double slash // indicates that this line contains *comments*. Any line beginning with a double slash will be taken as comments and such a line in the program will not be executed. The compiler ignores comments, which do not increase the size of the program or execution time. You can write anything (of course, meaningful statements) as comments. Next comes the two *#include* directives. These *preprocessor directives* tell the compiler to add the contents of the two files indicated in the <> brackets (i.e., *iostream.h* and *math.h*) in the program. Why is it done? Because the program we have just developed requires some declarations (required for input, output and mathematical operations), which are written in these files. These files are called *header files*, and they have the suffix .h as a matter of convention.

The next statement in the program specifies a class called "Program1". This specification contains the keyword **class,** followed by the name of the class, *Program1*. The curly brackets that follow this line delimit the body of the class. The class contains two keywords, viz. **private** and **public.** A semicolon follows each of these words. The data or functions, which need to be inaccessible from outside the class (but accessible from within the class), are put under *private*. On the other hand, data or functions that may be accessed from outside the class are put under *public*. This approach is intended for accidental misuse of data in programming. In programming parlance, it is called **data hiding.** The data items are called *data members of the class*. Similarly, the functions are called member *functions of the class*. The main idea of object-oriented programming is to place data and functions together in a single entity (i.e., the class). The description of a class is shown in Fig. 1.1.

Numerical Methods With Computer Programs In C++



Publisher : PHI Learning ISBN : 9788120329874 Author : GHOSH, PALLAB