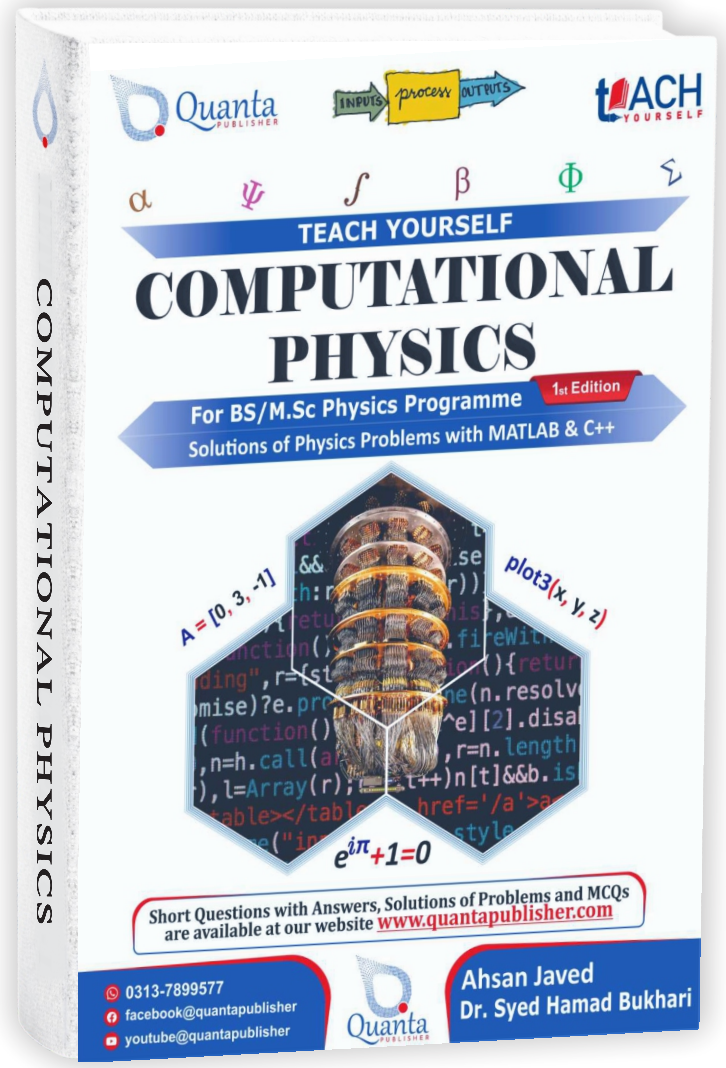
Quanta
PUBLISHER

BS Physics

SAMPLE PAGES



TEACH YOURSELF

COMPUTATIONAL PHYSICS (C++ & MATLAB)

1st Edition

For **BS/M.Sc Physics** students of all Pakistani Universities/Colleges

Ahsan Javed
PhD Scholar
Lecturer, Department of Physics
Govt. Associate College Kalyana, Pakpattan

&

Dr. Syed Hamad Bukhari
Assistant Professor
Department of Physics
G.C. University Faisalabad, Sub-Campus, Layyah

Quanta Publisher, 2660/6C Raza Abad, Shah Shamas, Multan.

Contents

Chapter 1

Programming:
The Art of Digital Creation

> *Whether you want to uncover the secrets of the universe, or you just want to pursue a career in the 21st century, basic computer programming is an essential skill to learn.*
> *Stephen Hawking (1942 2018)*

Computational Physics is a branch of physics that uses numerical methods and algorithms to solve problems. This can include simulating physical systems, solving dierential equations, and analyzing data. It is used in a wide range of areas, such as condensed matter physics, astrophysics, and particle physics. It is an interdisciplinary eld that combines physics, applied mathematics and computer science. Computational physics has led us to many breakthroughs i.e. discovery of dark matter[1], imaging of black hole, etc.

[1] In the early 20^{th} century, Swiss astronomer Fritz Zwicky observed that the observed mass of galaxy clusters was not enough to account for the gravitational forces holding them together. He proposed the existence of **Dark Matter** to account for the discrepancy. Dark matter is a mysterious form of matter that is believed to make up about 85% of the universes mass. The nature and properties of dark matter are not well understood, and it cannot be directly detected with current technology. However, it wasnt until the late 20^{th} century that technology advanced enough to allow for detailed simulations of the universe to be run on supercomputers. These simulations showed that the distribution of matter in the universe matched observations much better when an additional, invisible form of matter was included in the calculations. This provided strong evidence for the existence of dark matter, and has since been conrmed through various other observations and experiments. This has led to the conclusion that dark matter must exist in order to account for the gravitational forces holding galaxy clusters together and the large-scale structure of the universe. The story illustrates the power of computational physics in helping to solve complex problems and make important discoveries in the eld of physics. It also shows how simulations can be used to make predictions to be conrmed through experiments.

Chapter 2

Exploring the World of C++

C++ is a valuable asset for physicists, allowing us to model, simulate,
and analyze systems with unprecedented accuracy and precision.
C++ creates a strong programming base i.e. if you are good in this,
you will be good in other programming languages too.

2.1 Getting Started with IDE

C++ was developed in the early 1980s. It was designed to extend the C programming language and add object-oriented features. C++ has a syntax similar to C but has added features, allowing for more complex and exible program design. It is regarded as one of the most ecient and widely-used programming languages and is often used for developing complex, high-performance applications.

There are several Integrated Development Environments (IDEs[1]) available for C++ i.e., Dev-C++, Microsoft Visual Studio in Windows Operating System (OS), GCC in Linux OS[2], etc. For this book, we will be using Dev-C++ in **WINDOWS OS.**

1 **IDE** is a software application that provides comprehensive facilities to computer programmers for software development. An IDE consists of a source code editor, a compiler or interpreter, build automation tools, and a debugger. The purpose of an IDE is to streamline the development process by providing all the necessary tools and features in one place, allowing developers to write, test, and debug code eciently.

2 **Appendix C** to learn how to run a C++ program in LINUX OS

Chapter 3

Customizing Code: User-Dened Functions and Arrays

3.1 Functions in C++: Built-in and User-dened

Built-in functions

Built-in functions in C++ are functions that are part of the C++ standard library and are available for use without the need for additional code or libraries. These functions are already built into the compiler and can be called directly from within the code.

sqrt(x): Returns the square root of a number x.
pow(x, y): Returns x raised to the power of y.
abs(x): Returns the absolute value of a number x.
ceil(x): Returns the smallest integer greater than or equal to x.
max(x, y): Returns the larger of two values x and y.
min(x, y): Returns the smaller of two values x and y.
rand(): Returns a random integer.
sin(x): Computes the sine of an angle x (in **radians**).
log(x): Computes the natural logarithm (base e) of a number x.
log10(x): Computes the base-10 logarithm of a number x.
toupper(): Converts a lowercase character to uppercase.
tolower(): Converts an uppercase character to lowercase.

Chapter 4

Numerical Ninjas: Solving Problems with Code

Numerical analysis is a branch of mathematics that solves continuous problems using numeric approximation. It involves designing methods that give approximate but accurate numeric solutions, which is useful in cases where the exact solution is impossible or prohibitively expensive to calculate. Dierent numerical techniques are:

1. Numerical Integration Techniques
2. Numerical Dierentiation Techniques
3. Numerical Methods for Ordinary Dierential Equations (ODEs)
4. Numerical Methods for Solution of Non Linear Equations
5. Numerical Methods for Solution of Linear Equations
6. Numerical Solutions of Partial Dierential Equations

4.0.1 Numerical Errors and Instabilities

Error in numerical analysis refers to the dierence between the exact (theoretical) value and the approximated value obtained using a numerical method. The error can arise from various sources, such as rounding errors, truncation errors, or numerical instability.

1. **Rounding errors** One common source of approximation errors is the use of nite precision arithmetic, which can cause rounding errors when a oating-point number is approximated by a nite number of binary digits. This can lead to inaccuracies in

Chapter 5

MAT$_{rix}$ LAB$_{oratory}$

MATLAB (Matrix Laboratory) is a high-level programming language that is used for technical computing, data analysis and visualization. MATLAB was developed by MathWorks and was rst released in the 1980s. It is widely used in elds such as engineering, physics, nance and is also popular in education for teaching mathematical and scientic concepts. Advantages of MATLAB include its ease of use, its extensive library of built-in functions, and its ability to handle complex computations.

5.1 MATLAB Interface

To get started with MATLAB, you will need to download and install the software on your computer, and then launch the application.[1] Once you have the application open, you can begin writing code in the MATLAB. The MATLAB Desktop Environment is the graphical user interface (GUI) used to interact with the software.
The major components of the MATLAB interface (as shown in gure 5.1) include:

Command Window: This is the primary interface to MATLAB, where users can enter commands and run code. The Command Window displays the results of commands and allows users to interactively experiment with MATLAB.
Workspace: The Workspace displays variables that are currently in memory. Users can view and manipulate variables in the Workspace.

[1] **MATLAB Mobile** is a companion app that allows users to run MATLAB on their mobile devices, such as smartphones or tablets. The app provides a lightweight version, allowing users to access and run MATLAB code on their mobile devices, as well as view and edit les stored in the MATLAB Drive cloud storage.

Chapter 6

MATLAB Control Structures (MCS)

MATLAB provides various control structures that can be used to control the ow of execution of a program, such as:

1. **Selection** (if, if-else, if-elseif-else)
2. **Repetition** (for, while)

6.1 Selection

Selection refers to the ability to conditionally execute code based on certain conditions.[1].

6.1.1 Syntax for If

```
if condition
    % code to be executed if the condition is true
end
```

Heres an example of how you could use if to check if a variable x is greater than 0

```
x = 5;
if x > 0
    disp( x is positive );
end
```

[1] For detail theoretical discussion on Selection/Repetition theory, revisit **Section 2.5**

Chapter 7

Seeing is Believing: Data Visualization

Visualization in MATLAB refers to the process of plotting data in form of graphs, charts, and other types of visual representations. MATLAB is a powerful tool for data visualization, with a variety of built-in functions and options for customizing the plots.

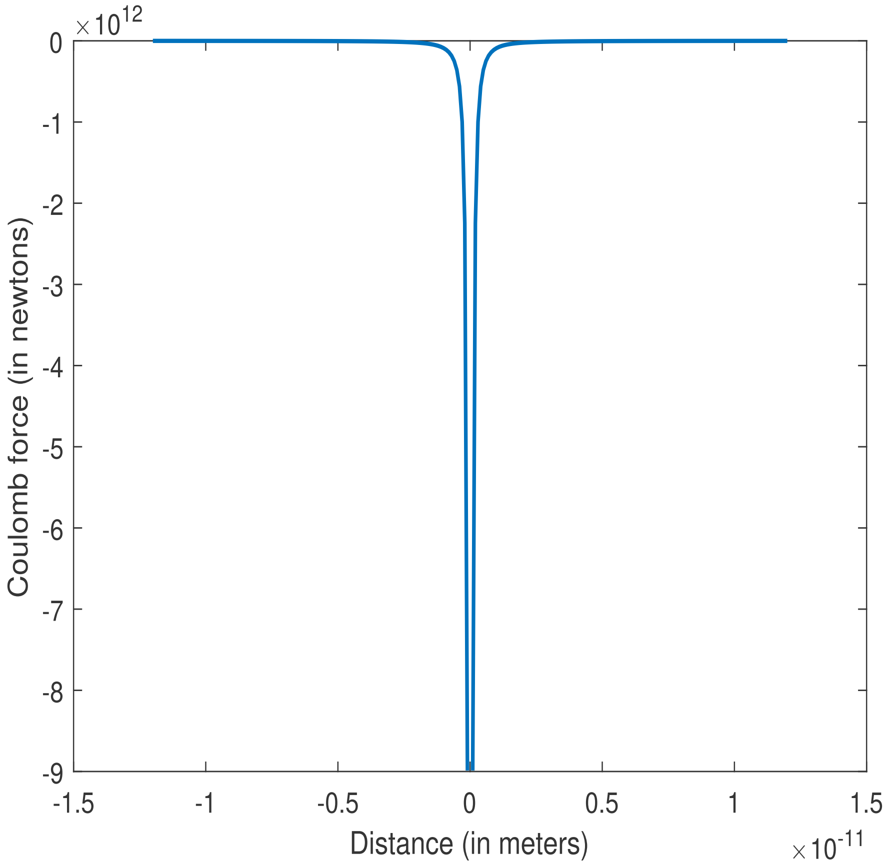


Fig. 7.1. Electrostatic (Coulomb) force as inverse square law

Chapter 8

Exploring the Physical World through MATLAB Simulations

MATLAB can be used to solve real-world physics problems: Many physical systems are too complex to understand through simple mathematical models or experiments alone. `Simulations` allow physicists to model these systems in great detail, taking into account all the relevant factors that contribute to their behavior. This allows them to gain a better understanding of how these systems work and how they might be manipulated.

8.1 Importance of Simulations in Physics

Predicting Behavior: Simulations can be used to make predictions about the behavior of physical phenomena under dierent conditions. For example, a simulation of the behavior of particles in a magnetic eld could be used to predict how these particles would behave in a particle accelerator. This allows physicists to design experiments that test these predictions and further rene their understanding of physical laws that govern behavior of these systems.

Designing experiments: Simulations can also be used to design experiments that are more ecient and eective. By simulating dierent experimental setups, physicists can optimize their designs to achieve the best possible results. This can save time and resources, and lead to more accurate and useful data.


Quanta
TEACH YOURSELF
COMPUTATIONAL PHYSICS
For BS/M.Sc Physics Programme
Solutions of Physics Problems with MATLAB & C++
Ahsan Javed
Dr. Syed Hamad Bukhari
Rs:400/-

Books by Quanta Publisher for BS Physics Students