

## **DEPARTMENT OF PHYSICS**

### **BSc. (Hons.) Physics**

#### *Category-I*

### **DISCIPLINE SPECIFIC CORE COURSE – 1 (DSC-1) Mathematical Physics I**

#### **CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE**

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course
		Lecture	Tutorial	Practical/ Practice		
Mathematical Physics I	4	3	0	1	Class XII pass with Physics and Mathematics as main subjects	Physics and Mathematics syllabus of class XII

#### **Learning Objectives**

The emphasis of the course is on applications in solving problems of interest to physicists. The course will teach the students to model a physics problem mathematically and then solve those numerically using computational methods. The course will expose the students to fundamental computational physics skills enabling them to solve a wide range of physics problems. The skills developed during course will prepare them not only for doing fundamental and applied research but also for a wide variety of careers.

#### **Learning Outcomes**

After completing this course, student will be able to,

- Draw and interpret graphs of various elementary functions and their combinations.
- Understand the vector quantities as entities with Cartesian components which satisfy appropriate rules of transformation under rotation of the axes.
- Use index notation to write the product of vectors in compact form easily applicable in computational work.
- Solve first and second order differential equations and apply these to physics problems.
- Understand the functions of more than one variable and concept of partial derivatives.
- Understand the concept of scalar field, vector field, gradient of scalar field and divergence and curl of vector fields.
- Perform line, surface and volume integration and apply Green's, Stokes' and Gauss's theorems to compute these integrals and apply these to physics problems
- Understand the properties of discrete and continuous distribution functions.

In the laboratory course, the students will learn to,

- Prepare algorithms and flowcharts for solving a problem.
- Design, code and test simple programs in Python/C++ to solve various problems.

- Perform various operations of 1-d and 2-d arrays.
- Visualize data and functions graphically using Matplotlib/Gnuplot

## SYLLABUS OF DSC – 1

### THEORY COMPONENT

#### Unit 1 (18 Hours)

**Functions:** Plotting elementary functions and their combinations, Interpreting graphs of functions using the concepts of calculus, Taylor's series expansion for elementary functions.

**Ordinary Differential Equations:** First order differential equations of degree one and those reducible to this form, Exact and Inexact equations, Integrating Factor, Applications to physics problems

Higher order linear homogeneous differential equations with constant coefficients, Wronskian and linearly independent functions. Non-homogeneous second order linear differential equations with constant coefficients, complimentary function, particular integral and general solution, Determination of particular integral using method of undetermined coefficients and method of variation of parameters, Cauchy-Euler equation, Initial value problems. Applications to physics problems

#### Unit 2 (12 Hours)

**Vector Algebra:** Transformation of Cartesian components of vectors under rotation of the axes, Introduction to index notation and summation convention. Product of vectors - scalar and vector product of two, three and four vectors in index notation using  $\delta_{ij}$  and  $\epsilon_{ijk}$  (as symbols only – no rigorous proof of properties). Invariance of scalar product under rotation transformation.

**Vector Differential Calculus:** Functions of more than one variable, Partial derivatives, chain rule for partial derivatives. Scalar and vector fields, concept of directional derivative, the vector differential operator  $\vec{\nabla}$ , gradient of a scalar field and its geometrical interpretation. Divergence and curl of a vector field and their physical interpretation. Laplacian operator. Vector identities.

#### Unit 3 (15 Hours)

**Vector Integral Calculus:** Integrals of vector-valued functions of single scalar variable. Multiple integrals, Jacobian, Notion of infinitesimal line, surface and volume elements. Line, surface and volume integrals of vector fields. Flux of a vector field. Gauss divergence theorem, Green's and Stokes' Theorems (no proofs) and their applications

**Probability Distributions:** Discrete and continuous random variables, Probability distribution functions, Binomial, Poisson and Gaussian distributions, Mean and variance of these distributions.

## PRACTICAL COMPONENT (Hours)

(30

The aim of this lab is not just to teach computer programming and numerical analysis but to emphasize its role in solving problems in Physics. The course will consist of practical sessions and lectures on the related theoretical aspects of the laboratory. Assessment is to be done not only on the programming but also on the basis of formulating the problem.

- Every student must perform at least 6 programs covering each unit.
- The list of recommended programs is suggestive only. Students should be encouraged to do more practice. Emphasis should be given to assess student's ability to formulate a physics problem as mathematical one and solve by computational methods.
- The implementation can be either in Python or C++. Accordingly, the instructor can choose section A or B respectively from Unit 1 and 2. The list of programs is common for both sections. If C++ is used, then for all plotting programs, Gnuplot has to be used.

Basics of scientific computing (Mandatory):

- (a) Binary and decimal arithmetic, Floating point numbers, single and double precision arithmetic, underflow and overflow, numerical errors of elementary floating point operations, round off and truncation errors with examples.
- (b) Introduction to Algorithms and Flow charts. Branching with examples of conditional statements, for and while loops.

### Unit 1

Section A:

**Basic Elements of Python:** The Python interpreter, the print statement, comments, Python as simple calculator, objects and expressions, variables (numeric, character and sequence types) and assignments, mathematical operators. Strings, Lists, Tuples and Dictionaries, type conversions, input statement, list methods. List mutability, Formatting in the print statement.

**Control Structures:** Conditional operations, if, if-else, if-elif-else, while and for loops, indentation, break and continue, List comprehension. Simple programs for practice like solving quadratic equations, temperature conversion etc.

**Functions:** Inbuilt functions, user-defined functions, local and global variables, passing functions, modules, importing modules, math module, making new modules. Writing functions to perform simple operations like finding largest of three numbers, listing prime numbers, etc., Generating pseudo random numbers.

OR

**Section B:**

**Introduction to C++:** Basic idea of Compilers. Structured programming. Idea of Headers, Data Types, Enumerated Data, Conversion and casting, constants and variables, Mathematical, Relational, Logical and Bit wise Operators. Precedence of Operators, Expressions and Statements, Scope and Visibility of Data, block, Local and Global variables, Auto, static and External variables. Input and output statements. I/O

manipulations, iostream and cmath header files, using namespace.

**Control Statements:** The if-statement, if-else statement, Nested if Structure, If - Else if – else block, Ternary operator, Goto statement, switch statement, Unconditional and Conditional looping, While loop, Do-while loop, For loop, nested loops, break and continue statements. Simple programs for practice like solving quadratic equations, temperature conversion etc.

**Functions:** Inbuilt functions. User-defined functions, function declaration, function definition, function prototype, void functions and function arguments, return statement. Local and global variables. The main function. Passing parameter by value and by reference. Inline functions. Function overloading. Writing functions to perform simple operations like finding largest of three numbers, listing prime numbers etc., Generating pseudo random numbers.

#### Recommended List of Programs (At least Two)

- (a) Make a function that takes a number N as input and returns the value of factorial of N. Use this function to print the number of ways a set of m red and n blue balls can be arranged.
- (b) Generate random numbers (integers and floats) in a given range and calculate area and volume of regular shapes with random dimensions.
- (c) Generate data for coordinates of a projectile and plot the trajectory. Determine the range, maximum height and time of flight for a projectile motion.

## Unit 2

### Section A:

**NumPy Fundamentals:** Importing Numpy, Difference between List and NumPy array, Adding, removing and sorting elements, creating arrays using ones(), zeros(), random(), arange(), linspace(). Basic array operations (sum, max, min, mean, variance), 2-d arrays, matrix operations, reshaping and transposing arrays, savetxt() and loadtxt().

**Plotting with Matplotlib:** matplotlib.pyplot functions, Plotting of functions given in closed form as well as in the form of discrete data and making histograms.

OR

### Section B:

**Arrays:** Array definition, passing arrays to functions, Finding sum, maximum, minimum, mean and variance of given array. 2-d arrays, matrix operations (sum, product, transpose etc). Saving data generated by a C++ program in a file.

**Gnuplot:** Introduction to Gnuplot. Visualization of discrete data and plotting functions given in closed form and data for graphical visualization. Plotting data from the output file created by a C++ program, making histogram.

#### Recommended List of Programs (At least Three)

- (a) To plot the displacement-time and velocity-time graph for the un-damped, under-damped

critically damped and over-damped oscillator using matplotlib (or Gnuplot) using given formulae.

- (b) To compute the left, right and central approximations for derivative of a function given in closed form. Plot both the function and derivative on the same graph. Plot (using matplotlib/Gnuplot) the error as a function of step size on a log-log graph, study the behaviour of the plot as step size decreases and hence discuss the effect of round off error.
- (c) To generate array of N random numbers drawn from a given distribution (uniform, binomial, poisson and gaussian) and plot them using matplotlib/Gnuplot for increasing N to verify the distribution. Verify the central limit theorem.
- (d) To implement the transformation of physical observables under Galilean, Lorentz and Rotation transformation

### Unit 3

#### **Recommended List of Programs (At least one)**

- (a) To find value of  $\pi$  and to integrate a given function using acceptance-rejection method.
- (b) To perform linear fitting of data using the inbuilt function `scipy.stats.linregress` in Python or using Gnuplot. Plot the data points and the fitted line on the same graph.

#### References (for Laboratory Work):

- 1) Documentation at the Python home page (<https://docs.python.org/3/>) and the tutorials there (<https://docs.python.org/3/tutorial/>).
- 2) Documentation of NumPy and Matplotlib : <https://numpy.org/doc/stable/user/> and <https://matplotlib.org/stable/tutorials/>
- 3) Schaum's Outline of Programming with C++, J. Hubbard, 2000, McGraw-Hill Education.
- 4) C++ How to Program, Paul J. Deitel and Harvey Deitel, Pearson (2016).
- 5) Computational Physics, Darren Walker, 1st Edn., Scientific International Pvt. Ltd (2015).
- 6) Elementary Numerical Analysis, K. E. Atkinson, 3rd Edn., 2007, Wiley India Edition.
- 7) An Introduction to Computational Physics, T. Pang, Cambridge University Press (2010).
- 8) Introduction to Numerical Analysis, S. S. Sastry, 5th Edn., 2012, PHI Learning Pvt. Ltd.
- 9) Applied numerical analysis, Cutis F. Gerald and P. O. Wheatley, Pearson Education, India (2007).

#### **Essential/Recommended Readings**

#### **REFERENCES FOR THEORY COMPONENT**

- 1) An introduction to ordinary differential equations, E.A. Coddington, 2009, PHI learning.
- 2) Differential Equations, George F. Simmons, 2007, McGraw Hill.
- 3) Mathematical methods for Scientists and Engineers, D.A. McQuarrie, 2003, Viva Book.
- 4) Advanced Engineering Mathematics, D.G. Zill and W.S. Wright, 5 Ed., 2012, Jones and Bartlett Learning.

- 5) Advanced Engineering Mathematics, Erwin Kreyszig, 2008, Wiley India.
- 6) Probability and Statistics, Murray R Spiegel, John J Schiller and R Alu Srinivasan, 2018, McGraw Hill Education Private Limited.
- 7) Essential Mathematical Methods, K.F.Riley and M.P.Hobson, 2011, Cambridge Univ. Press.
- 8) Vector Analysis and Cartesian Tensors, D.E. Bourne and P.C. Kendall, 3 Ed. , 2017, CRC Press.
- 9) Vector Analysis, Murray Spiegel, 2 Ed., 2017, Schaum's outlines series.
- 10) John E. Freund's Mathematical Statistics with Applications, I. Miller and M. Miller, 7<sup>th</sup> Ed., 2003, Pearson Education, Asia.

**.Suggestive readings:**

- 1) Mathematical Methods for Physicists, G.B. Arfken, H.J. Weber, F.E. Harris, 7 Ed., 2013, Elsevier.
- 2) Introduction to Electrodynamics, Chapter 1, David J. Griffiths, 4 Ed., 2017, Cambridge University Press.
- 3) The Feynman Lectures on Physics, Volume II, Feynman, Leighton and Sands, 2008, Narosa Publishing House.
- 4) Introduction to Vector Analysis, Davis and Snider, 6 Ed., 1990, McGraw Hill.
- 5) Differential Equations, R. Bronson and G.B. Costa, Schaum's outline series.
- 6) Mathematical Physics, A.K. Ghatak, I.C. Goyal and S.J. Chua, Laxmi Publications Private Limited (2017)
- 7) Mathematical Tools for Physics, James Nearing, 2010, Dover Publications.

**DISCIPLINE SPECIFIC CORE COURSE – 2 (DSC - 2) MECHANICS**

**Credit distribution, Eligibility and Prerequisites of the Course**

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course
		Lecture	Tutorial	Practical/ Practice		
<b>Mechanics</b> <b>DSC – 2</b>	<b>4</b>	<b>3</b>	<b>0</b>	<b>1</b>	Class XII pass with Physics and Mathematics as main subjects	Physics and Mathematics syllabus of class XII

**Learning Objectives**

This course reviews the concepts of mechanics learnt at school from a more advanced perspective and goes on to build new concepts. It begins with Newton's Laws of Motion and ends with the Fictitious Forces and Special Theory of Relativity. The students will learn the collisions in the centre of mass frame, rotational motion and central forces. They will be able to apply the concepts learnt to several real world problems. In the laboratory part of the course, the students will learn to use various instruments, estimate the error for