Module: Introduction to Computational Physics

Module Code: Phys-M3111;	EtCTS of Course: 5 ;	Course Status:	Compulsory
Course Title :	**Introduction to Computational Physics**		
Course Code:	Phys3111	Credits Hours:	3
Mode of delivery:	Full Semester;	Weeks required:	16
Prerequisite(s):	Comp	Co-requisite(s):	
Academic Year:	20_____/_____	Year/Semester:	III/ II
Students' College/Faculty:	_____	Department:	Physics
Program:	Undergraduate	Enrollment:	_____
Instructor's Name (Coordinator)	__________		
	Address:	Block No._____	Rm. No. _____
Class Hours:	__________		

Module/Course Rationale

Computational Physics is a problem-solving technique, that is, the measure of a student's progress is demonstrated by the ability to solve numerical problems in physics. While the very nature of physics is to express relationships between physical quantities in mathematical terms, an analytic solution of the resulting formulas is often not available. Instead, numerical solutions based on computer programs are required to obtain concrete results for real problems. Computation has led to important conceptual advances and new ways of thinking about physical systems. Computation can support three-dimensional visualizations of abstract quantities, offer opportunities to construct symbolic rather than numeric solutions to problems, and provide experience with the use of vectors as coordinate-free entities. Computation can also allow students to explore models in a way not possible using the analytical tools available to students. Computer simulation is considered to be the third option for solving physical problems. Upon completion of this module, the student will posses the basic knowledge of numerical modeling that may be required for graduate school or in a position at a technical corporation. In addition, almost all undergraduate students who take physics module will use computational tools in their future careers even if they do not become practicing physicists.

Learning Outcomes

The aim of the course is to show how the power of computers enables a computational approach to solving physics problems to be adopted, which is distinct from, and complimentary to, traditional experimental and theoretical approaches. The material covered will found useful in any project or problem solving work that contains a strong computational or data analysis element. The module is designed such that a significant fraction of the students' time is spent actually programming specific physical problems rather than learning abstract techniques.

Upon completion of this course students should be able to:

- gain experience on writing manuscripts in a scientific journal style using the LATEX,
- have the ability to use computers to solve physical problems,
- write a moderate-sized computer program to model a given physical process.
- use the program to investigate the underlying physics of the given process.
- discretize a differential equation using grid and basis set methods,
- improved confidence in developing and writing computer programs for scientific applications
- develop awareness of the value and also of the limitations of numerical methods in the simulation of physical systems
- demonstrate knowledge in essential methods and techniques for numerical computation in physics

- use appropriate numerical method to solve the differential equations governing the dynamics in physical systems
- design and implement computer programs to solve physical problems by using Fortran, C++ or other software
- outline the essential features of each of the simulation techniques introduced and give examples of their use in contemporary science,
- develop computer simulation for science problems, and investigate the problems using statistical, graphical and numerical packages,
  They will have gained a deeper understanding of the physical processes and principles underlying the particular system they have modelled.

Module Competency

Up on completion of this module, the students:

- Have knowledge of different computer working environments and use the appropriate one for their purpose.
- Can write computer program specific to scientific problem they want to solve numerically.
- Can generate data from the program they develop and demonstrate the results graphically.
- Can analyze the numerical results against experimental and analytical results.

Module Description

This module is designed to cover Introductory Computational Physics and techniques used in modeling physical systems numerically. It is designed to help the students in the selection of an operating system (Windows versus Unix/Linux), and programming language (some of the more popular in science include Fortran, C, C++, MatLab, Mathematica, and Visual Basic) that best meet the requirements needed to solve the problem. Techniques will be developed to data fitting and to numerically differentiate and integrate, and to solve systems of linear equations, ordinary differential equations (ODE), trajectory and orbit problems with numerical methods, and finally Fourier analysis. Molecular dynamics, Monte-Carlo techniques and Ising Model will also be discussed as modern applications to the technique.

Course Outline (5 ECTS)

No	Topic	Lecture (hrs)	Lab (hrs)	Home (hrs)	T.Load (hrs)
1)	**Introduction**	4	4	7	15
2)	**Methods of data fitting**	2	3	4	9
3)	**Root finding**	2	3	4	9
4)	**Methods of differentiation and integration**	2	3	4	9
5)	**Function optimization**	2	3	4	9
6)	**Matrices and systems of linear equations**	3	5	7	15
7)	**Numerical solutions to ordinary differential equations**	3	5	7	15
8)	**Trajectories and orbits**	2	3	4	9
9)	**Fourier analysis and oscillations**	2	3	4	9
10)	**Molecular dynamics**	2	3	4	9
11)	**Monte Carlo methods**	2	3	4	9
12)	**2-D and 3-D numerical problems**	2	3	4	9
13)	**The Ising model**	2	3	4	9
	Total	30	44	61	135

Method of Teaching

The module hase two structures:

1 The Lecture Component (2 hrs per week) and which covers:

- physics background of the projects
- introduction to numerical algorithms
- basics of data analysis

2 The Laboratory Component, (3 hrs per week) and which covers
Students:

- develp computer code
- run simulations
- analyze and visualize data

Instructor:

- guides setup of simulations
- helps with coding and debugging
- provides input for data analysis

Tentative Time Breakdown of Lecture Topics

Date	Topics	Pedagogical Approaches	Teachers' Tasks/Activities	Students' Tasks/Activities
Week 1	Introduction	Lecture Questioning and Answering	explain experiment and theory as problem solving techniques in physics Define Computational Physics assign reading assignments	Listen to presentation take notes home reading
	Operating systems Unix, Latex, Postscript, pdf	Group discussion, Questioning and Answering, Learning by doing	Question and answer Explain use of Unix, Latex, Postscript and pdf lab demonstration	Ask and Answer questions take notes, lab practice
Week 2	Scientific programming	Group discussion and Questioning and answering	Ask quation Introduce programming languages	Listen to a lesson and take short notes, Asking and answering questions, Doing Home work questions
	Error analysis and uncertainties	Group discussion Questioning and answering Learning by doing	Ask quations on errors and uncertainties Tutoring practice	Group discussion Questioning and answering Lab practice
Week 3	Methods of data fitting	Lecture Lecture Learning by doing	Questioning and answering Present the lesson Leading lab session **Provide home work**	Questioning and answering Take notes Practice exercise at home
Week 4	Root finding	Lecture Lecture Lab work	Questioning and answering Present the lesson Leading lab session **provide home work**	Questioning and answering Take notes Practice exercise at home
Week 5	Methods of differentiation and integration	Questioning and answering Lecture	Asking questions Lecturing Tutoring	Questioning and answering taking notes lab practice exercise at home
Facility dependent individual/group project assignment				
Week 6	Function Optimization	Questioning and answering	Questions and Lecturing	Questioning, answering and take notes
Test I				

Date	Topics	Pedagogical Approaches	Teachers' Tasks/Activities	Students' Tasks/Activities
Week 7	Matrices and Systems of Linear Equations	Questioning and answering Learning by doing	Questioning, answering and lecturing lab demonstration **provide home work**	Questioning, answering take notes lab practice perform home work
Week 8	Matrices and Systems of Linear Equations	Questioning and answering Learning by doing	Questioning, answering and lecturing lab demonstration provide home work	Questioning, answering take notes lab practice perform home work
	Numerical Sulutions to ordinary differential equations	Lecture and Learning by doing	Questioning, answering lecturing Leading lab session	Questioning, answering and take notes lab practice
	Group Assignment			
Week 9	Numerical Sulutions to ordinary differential equations	Lecture and Learning by doing	Questioning, answering lecturing Leading lab session Provide group assignment	Questioning, answering and take notes lab practice
	Test II			
Week 10	Trajectories and orbits	Lecture, discussion, questioning and answering	questioning, answering and lecturing give examples **provide home work**	questioning, answering and taking notes home exercise
Week 11	Fourier Analysis and Oscillations	Lecture	Lecturing provide class and home activities	taking notes perform activities
Week 12	Molecular dynamics	Lecture Demonstraion	questioning, answering Topic lecturing Show simulation and tutoring	questioning, answering and taking notes observe simulation and lab practice
	Individual Assignment			
Week 13	Monte Carlo Methods	Lecture and Demonstration Learning by doing	Lecturing and Demonstrating the method Tutoring	Observe demonstration and take notes lab practice
Week 14	2-D and 3-D numerical problems	Lecture and Demonstration learning by doing	Demonstrate 2-D and 3-D objects and lecturing Tutoring **provide home work**	Observe demonstration and take notes lab practice
Week 15	The Ising Model	Lecture	Lecturing with examples, questioning, answering	Taking notes, questioning and answering
Week 15	**Revision**			
Week 16	**Written and Practical final examination**			

Assessment

No	Type of Assessment	Time	Weight
1	Test I	Week 6	10%
	Test II	Week 9	10%
2	Group Assignment I	Week 5	10%
3	Group project	Week 8	15%
4	Individual assignment	Week 12	15%
5	Class activity and home work	All weeks	10%
6	Practical final exam	Final week	15%
7	Written final exam	Final week	15%
		Total	100%

Recommended References

1. S.S. Sastry, *Introductory Methods of Numerical Analysis*, 3^{rd}*ed.*, Prentice Hall of India, New Delhi (2003).
2. Tao Pang, *An Introduction to Computational Physics*,Cambridge University Press, (1997)
3. R. Fitzpatrick, *Computational Physics: Computer based learning unit*, University of Leads, (1996).
4. H Gould, et al, *An Introduction to computer simulation methods: Application to Physical System*, 2^{nd} ed., (1995).
5. R. Fitzpatrick, *Introduction to Computational Physics*, University of Texas.
6. V. Rajaraman, *Computer Programming in Fortran 90 and 95*; Prentice Hall of India, New Delhi (2002).
7. S. Rahtz, *A LATEX Survival Guide for Unix Systems*; 1994 and the Internet.