MSE3114: COMPUTATIONAL METHODS FOR PHYSICISTS AND MATERIALS ENGINEERS

Effective Term

Semester B 2023/24

Part I Course Overview

Course Title

Computational Methods for Physicists and Materials Engineers

Subject Code

MSE - Materials Science and Engineering

Course Number

3114

Academic Unit

Materials Science and Engineering (MSE)

College/School

College of Engineering (EG)

Course Duration

One Semester

Credit Units

3

Level

B1, B2, B3, B4 - Bachelor's Degree

Medium of Instruction

English

Medium of Assessment

English

Prerequisites

MA2158 Linear Algebra and Calculus or
MA2170 Linear Algebra and Multi-variable Calculus or
MA2172 Applied Statistics for Sciences and Engineering or
MA2177 Engineering Mathematics and Statistics or
MA2181 Mathematical Methods for Engineering

Precursors

Nil

Equivalent Courses

AP3114 Computational Methods for Physicists and Materials Engineers

Exclusive Courses

PHY3115 Introduction to Computational Physics

Part II Course Details

Abstract

Computational Science concentrates on solving scientific problem by computers. Emphasis is placed upon using software packages and programs to solve problems in Physics and Materials Engineering. This course consists of formulating and analyzing problems, simulations and modeling, mathematical and numerical analysis, visualisation through graphics, and preliminary programming.

This course introduces scientific problem-solving using Python (or MATLAB) programming. However, no previous programming experience in programming is needed. The focus is problem-solving rather than programming. Scientific problems from Physics and Materials Engineering will be targeted with the aim of introducing the use of computers in science for students who may need such skills in the pursuit of their Major of studies in Applied Physics and Materials Engineering.

Course Intended Learning Outcomes (CILOs)

	CILOs	Weighting (if app.)	DEC-A1	DEC-A2	DEC-A3
1	To analyse and formulate the mathematical models for typical problems in Physics and Materials Engineering.		x		
2	To attain a basic level of competency in computational tools, e.g. NumPy and SciPy, including the use of variables, arrays, matrices, and control structures involving logical statements, and to write Python code.		x	x	
3	To implement basic numerical methods, for example, procedures for numerical root finding, solution of ordinary differential equations, and Fourier transform, and to apply such techniques to solve the mathematical models of typical problems in Physics and Materials Engineering.		x	x	x
4	To understand some of the ways in which computation may lead to misleading results, including a model being invalid and numerical errors such as round-off error.		x		

A1: Attitude
Develop an attitude of discovery/innovation/creativity, as demonstrated by students possessing a strong sense of curiosity, asking questions actively, challenging assumptions or engaging in inquiry together with teachers.

A2: Ability
Develop the ability/skill needed to discover/innovate/create, as demonstrated by students possessing critical thinking skills to assess ideas, acquiring research skills, synthesizing knowledge across disciplines or applying academic knowledge to real-life problems.

A3: Accomplishments
Demonstrate accomplishment of discovery/innovation/creativity through producing /constructing creative works/new artefacts, effective solutions to real-life problems or new processes.

Teaching and Learning Activities (TLAs)

	TLAs	Brief Description	CILO No.	Hours/week (if applicable)
1	Lecture	Explain key concepts such as theories related to numerical scheme	1, 2, 3, 4	3 hrs/wk
2	Tutorials	Demonstrate how to program, help students to write scripts and debug their code	1, 2, 3, 4	1 hr/wk

Assessment Tasks / Activities (ATs)

	ATs	CILO No.	Weighting (%)	Remarks (e.g. Parameter for GenAI use)
1	Assignments (At least 5)	1, 2, 3, 4	20	
2	Practical Tests	1, 2, 3, 4	30	
3	Examination	1, 2, 3, 4		Duration: 2 hours

Continuous Assessment (%)

50

Examination (%)

50

Examination Duration (Hours)

2

Assessment Rubrics (AR)

Assessment Task

1. Assignments

Criterion

1.1 Capacity for understanding concepts and program logic
1.2 Ability to design codes to solve scientific problems

Excellent (A+, A, A-)

High

Good (B+, B, B-)

Significant

Fair (C+, C, C-)

Moderate

Marginal (D)

Basic

Failure (F)

Not even reaching marginal levels

Assessment Task

2. Practical Tests

Criterion
2.1 Ability to solve physics and engineering problems independently

Excellent (A+, A, A-)
High

Good (B+, B, B-)
Significant

Fair (C+, C, C-)
Moderate

Marginal (D)
Basic

Failure (F)
Not even reaching marginal levels

Assessment Task
3. Examination

Criterion
3.1 Capacity for understanding concepts and program logic
3.2 Ability to solve physics and engineering problems independently

Excellent (A+, A, A-)
High

Good (B+, B, B-)
Significant

Fair (C+, C, C-)
Moderate

Marginal (D)
Basic

Failure (F)
Not even reaching marginal levels

Part III Other Information

Keyword Syllabus
Introduction:
- How to solve a problem?

Basic programming:
- Python, linear algebra by NumPy, plotting by matplotlib, and symbolic calculation by SymPy

Systems of linear equations:
- Scientific problems
- Direct method: Gaussian elimination, LU decomposition, QR decomposition

Iterative method: Jacobi method, Gauss-Seidel method, relaxation method, generalized minimal residual, conjugate gradient method

Least squares fitting, machine learning

Systems of nonlinear equations:

Fixed-point problem: Banach theorem

Root-finding problem: Newton’s method, damped Newton method, Quasi-Newton method

Ordinary differential equations:

Initial value problem: forward/backward Euler method, Heun method, Runge-Kutta method

Fourier analysis:

Continuous/discrete Fourier transform, fast Fourier transform

Parallel computing

Reading List

Compulsory Readings

	Title
1	Qingkai Kong, Timmy Siauw, and Alexandre Bayen (2020). Python Programming and Numerical Methods: A Guide for Engineers and Scientists. Academic Press

Additional Readings

	Title
1	Numerical Python: Scientific Computing and Data Science Applications with Numpy, SciPy and Matplotlib, 2nd ed. 2019, Apress, by Robert Johansson.
2	Fundamentals of Numerical Computation, 2018, Society for Industrial and Applied Mathematics, by Tobin A. Driscoll, and Richard J. Braun
3	Numerical Analysis, 1998, Springer, by Rainer Kress