

COURSE TITLE	PYTHON FOR SCIENTIFIC COMPUTING IN MECHANICAL ENGINEERING
COURSE CODE	01ME1509
COURSE CREDITS	2

Objective:

- 1 To equip students with practical skills in Python programming, numerical methods, and visualization for solving, analyzing, and simulating problems in mechanical engineering.

Course Outcomes: After completion of this course, student will be able to:

- 1 Apply Python programming and scientific libraries for solving engineering problems.
- 2 Implement numerical methods for algebraic, differential, and integral equations.
- 3 Visualize and interpret engineering data using 2D/3D plots and animations.
- 4 Apply Python programming to simulate and analyze real-world problems in core areas of Mechanical Engineering.
- 5 Develop and present mini-projects using Python to solve practical problems in mechanical and industrial applications.

Pre-requisite of course:NIL

Teaching and Examination Scheme

Theory Hours	Tutorial Hours	Practical Hours	ESE	IA	CSE	Viva	Term Work
0	0	4	0	0	0	50	50
Contents : Unit	Topics						Contact Hours
Total Hours							

Suggested List of Experiments:

Contents : Unit	Topics	Contact Hours
1	Python Programming on Python Fundamentals for Engineers • Basics: syntax, variables, conditionals, loops, functions, file I/O. • Python math operators and scientific constants. • Working with lists, tuples, dictionaries for data handling. • Application: Efficiency of heat engine, entropy change calculation, unit conversions.	10
2	Python Programming on Data Visualization with Matplotlib – Basics • 2D plotting: line, bar, scatter, multiple curves. • Adding titles, legends, labels, and grid. • Application: PV diagram of an isothermal & adiabatic process.	4

Suggested List of Experiments:

Contents : Unit	Topics	Contact Hours
3	Python Programming on Advanced Plotting & 3D Visualization • Subplots, contour plots, surface plots, heatmaps. • 3D plotting using Axes3D. • Simulation • Application: Temperature distribution (dummy data) on a plate.	3
4	Python Programming on Matrix Operations with Numpy • Matrix addition, multiplication, transpose, inverse. • Eigenvalues and eigenvectors, LU decomposition. • Application: Force analysis in 2-bar truss, stiffness matrix method.	3
5	Python Programming on Solution of Linear Systems • Direct methods: Gauss elimination, Gauss-Jordan. • Iterative methods: Jacobi, Gauss-Seidel. • Application: 1D steady-state heat conduction with 3 nodes.	3
6	Python Programming on Root Finding Methods • Bisection, Regula-Falsi, Newton-Raphson. • Error estimation and convergence. • Application: Solve for Mach number in compressible flow.	3
7	Python Programming on Numerical Differentiation • Forward, backward, central difference. • Higher-order finite differences. • Application: Velocity and acceleration from displacement-time data.	3
8	Python Programming on Numerical Integration methods • Trapezoidal, Simpson's 1/3 and 3/8 rule. • Gaussian quadrature basics. • Application: Work done in PV diagram, flowrate in open channel.	3
9	Python Programming on Interpolation and Curve Fitting methods • Newton forward & backward interpolation. • Lagrange interpolation, least squares regression. • Application: Stress-strain data curve fitting, predicting yield stress.	3
10	Python Programming on Solving ODEs (Initial Value Problems) • Euler, Modified Euler, Runge-Kutta 2nd and 4th order. • Solving single and coupled ODEs with scipy.integrate.odeint. • Application: Mass-spring-damper vibration, population growth analogy.	4
11	Python Programming on Boundary Value Problems & PDEs • Shooting method for boundary value ODEs. • Finite difference method for PDEs. • Application: Heat conduction in a fin with convective boundary.	4
12	Python Programming on Welding & Heat Transfer Simulation • Gaussian heat source modeling. • Transient temperature distribution using finite difference. • Contour, surface, and animated plots of heat distribution. • Application: TIG welding plate simulation.	5

Suggested List of Experiments:

Contents : Unit	Topics	Contact Hours
13	Mini Project Requirements • Technology: Python (mandatory) + Generative AI framework/API • Relevance: Must solve or simulate a problem, or create a tool/game for mechanical engineering (thermodynamics, fluids, SOM, manufacturing, robotics, welding). Deliverables: • Documented Python code • Working demo (software/web app/game/CLI) • Final report (problem, method, results, limitations, scope) Team Size: 2–3 students	0
Total Hours		48

Textbook :

- 1 A Primer on Scientific Programming with Python, Hans Petter Langtangen, Springer, 2016
- 2 Numerical Python: A Practical Techniques Approach for Industry, Robert Johansson, Apress, 2019
- 3 Python Programming and Numerical Methods: A Guide for Engineers and Scientists, Qingkai Kong, Timmy Siau, Alexandre Bayen, Academic Press, 2020

References:

- 1 Numerical Methods for Engineers, Numerical Methods for Engineers, Steven C. Chapra and Raymond P. Canale, McGraw-Hill, 2015
- 2 Applied Numerical Methods with Python for Engineers and Scientists, Applied Numerical Methods with Python for Engineers and Scientists, Steven C. Chapra, McGraw-Hill, 2021
- 3 Engineering Problem Solving with Python, Engineering Problem Solving with Python, Delores M. Etter, Pearson, 2018
- 4 Computational Fluid Mechanics and Heat Transfer, Computational Fluid Mechanics and Heat Transfer, Richard H. Pletcher, John C. Tannehill, Dale Anderson, CRC Press, 2013
- 5 An Introduction to Computational Science: Modeling and Simulation for the Sciences, An Introduction to Computational Science: Modeling and Simulation for the Sciences, Angela B. Shiflet and George W. Shiflet, Princeton University Press, 2014
- 6 Finite Element Analysis with Python, Finite Element Analysis with Python, Jasbir S. Arora, CRC Press, 2022

Suggested Theory Distribution:

The suggested theory distribution as per Bloom's taxonomy is as follows. This distribution serves as guidelines for teachers and students to achieve effective teaching-learning process

Distribution of Theory for course delivery and evaluation					
Remember / Knowledge	Understand	Apply	Analyze	Evaluate	Higher order Thinking / Creative
10.00	10.00	30.00	30.00	20.00	0.00

Instructional Method:

- 1 Online Platform

Supplementary Resources:

- 1 https://onlinecourses.nptel.ac.in/noc24_cs57/preview