CS 535: Introduction to Scientific Computing
Spring 2016

CBCT (UG/PG)
Department of Computer Science & Engineering,
Tezpur University

Course Instructor: Zubin Bhuyan
zubin@tezu.ernet.in

L-T-P: 2-0-1

1 Course Overview

CS 535 introduces students to the various concepts and practices used for creating computational solutions to STEM problems. This will include modelling, simulation, and visualization, rather than just numerical analysis.

2 Course prerequisites

Students willing to take this course must have knowledge of calculus and vector/matrix operations. Although it is not mandatory, it will be helpful if the students have a working knowledge of C, C++, Python, or any such programming language. ***A significant time commitment will be required from students with minimal or no previous programming experience.***

3 Learning Objectives

	Aspect	Learning Objectives
1	Programming Basics	To obtain an understanding of the tools and methods of programming; to be able to write and execute programs for solving scientific problems.
2	Modelling	To learn how to model physical systems, and transform continuous/ statistical models into programs.
3	Simulation	To learn how to solve linear and non-linear systems, interpolation and approximation, and numerical differential equations.

4 Lecture Plan

Lecture (tentative)	Topic(s)
1	Motivation: What is Scientific Computing and why do we need it
2	Introduction to software and programming tools; “Hello World” program
3	Understanding *Python 3* and IDLE: Installation demo
4	Simple tasks in Python: Class Demo
5-8	Branching & Looping constructs, Lists
9-10	Functions
11-15	Classes and Modules
11	Random and pseudo-random number generation
12-14	Discrete and continuous modelling
15-16	Interpolation methods
17	Bisection, Newton–Raphson method
18	Integration: Midpoint, Trapezoidal rule
19-20	Differential equations: Euler method, Runge-Kutta methods
21-22	Linear Algebra; Gaussian Elimination *Comparision with Cramer’s rule*
23	Simple and Damped Harmonic Motion
24	Scientific Visualization: Visualizing datasets
25-27	Overflow and Underflow; Errors in numerical computing: Floating point arithmetic, Relative and absolute errors, truncation and round-off
28-30	Clustering analysis, algorithms, implementing K-means
31-34	Statistical classification techniques and approaches, SVM
35-38	Hands-on ‘Cloudy’: simulating interstellar environments!

5 Outcome

After completion of this course, students will be able to write programs with a better understanding of the language constructs, common abstraction mechanisms, and efficiency considerations. They will be able to perform practical implementation of solutions to scientific and engineering problems.

6 Evaluation & Grading Scheme

Test no.	Marks	Duration
I	25	30 minutes
II	25	30 minutes
III	40 (Major 1)	60 minutes
IV	25 (Assignments: 5 x 5)	-
V	25	30 minutes
VI	60 (Major 2)	120 minutes
Total	200	

6.1 Assignment submission & Extensions

All assignments must be submitted by the due date and time. Deadline extensions will be granted only for university approved excuses.

6.2 Make-up Policy

Make-ups for quizzes, exams and assignments will be given only for situations which are beyond the student’s control. Circumstances necessitating the absence have to be established with proper documentation, and have to be approved by the DAC.

6.3 Plagiarism

The Oxford English Dictionary defines plagiarism as the wrongful appropriation and publication as one's own, of the ideas, or the expression of the ideas of another. In simple words, it means **stealing and publication of another author's language, thoughts, ideas, or expressions.** Such scholastic dishonesty will be dealt with severe penalty.