CS2008: Numerical Computing

Introduction, Course Overview, Policies

Muhammad Almas Khan
muhammad.almas@nu.nu.pk

FAST-NUCES

January 10, 2025

Course Outline

What is Numerical Computing?

- Understanding and implementing numerical methods.
- Solving mathematical problems using computational techniques.
- Applications in science, engineering, and business.

Why Study This Course?

- **Purpose:**
  - Equip students with computational tools to solve real-world mathematical problems.
  - Prepare for careers in data science, engineering, and research.
- **Importance:**
  - Numerical computing is essential for solving complex problems where analytical solutions are impractical.
  - Bridges the gap between theory and practice in mathematical modeling.

Rules of the Game

- **Punctuality:** Be on time and silence your phone.
- **Focus:** Engage with the lecture, not your friends.
- **Participation:** Ask questions and respect opinions.
- **Attendance:** Attendance will be recorded at the start of class. After class, ensure your presence is correctly marked on Flex and address any discrepancies promptly, don't expect any correction after class time.
- **Evaluation:** Address discrepancies within one day after update on Flex.
- **Group Work:** Disjoint groups of 3 or 4 students.

What Will You Learn?

- Fundamental concepts in numerical computing.
- Hands-on experience with tools like Python and numerical libraries.
- Real-world applications of numerical methods in various fields.

Tools You Will Use

- **Python Programming Language:** The primary language for this course.
- **NumPy:** A library for numerical computing and array manipulations.
- **SciPy:** A library for advanced mathematical and numerical functions.
- **Matplotlib and Seaborn:** Libraries for data visualization.

Why Python?

- **Ease of Use:** Simple and readable syntax for quick implementation.
- **Powerful Libraries:** Libraries like NumPy, SciPy, and Matplotlib make it ideal for numerical tasks.
- **Versatility:** Widely used in academia, industry, and research.
- **Community Support:** Active community with extensive resources and documentation.

About This Course

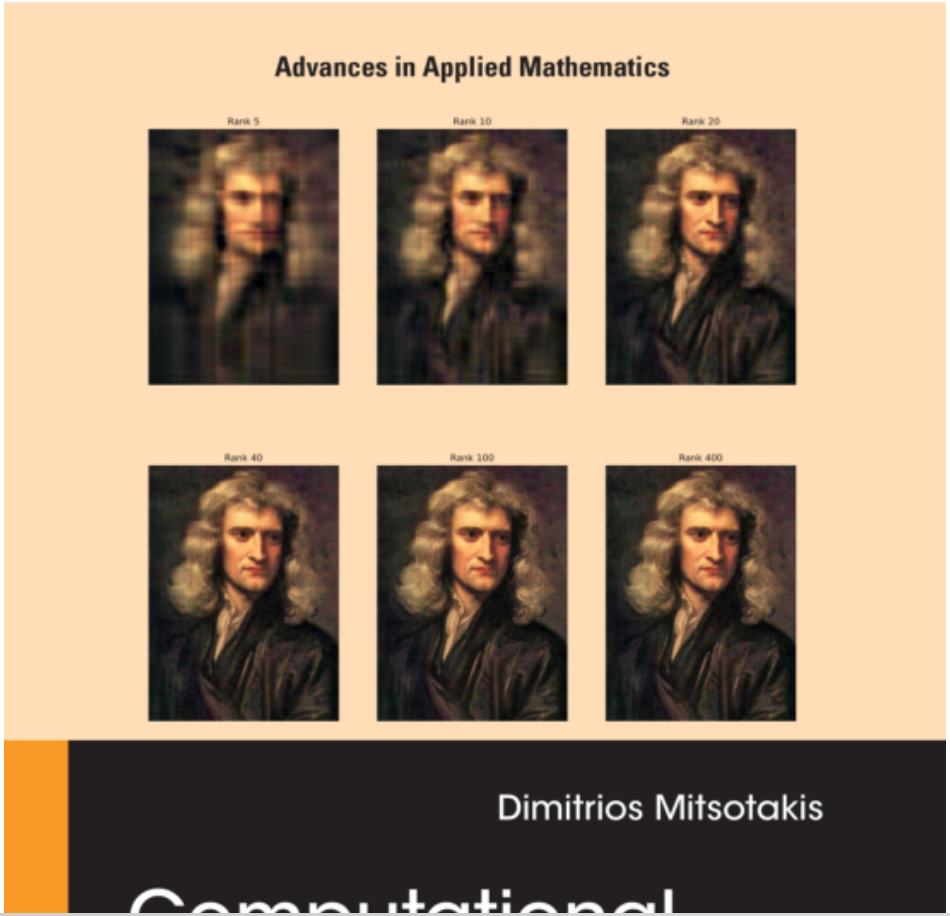
- **What is this course for?**
  - Developing computational skills for solving numerical problems.
- **What will you learn?**
  - Numerical methods for solving algebraic, differential, and optimization problems.
- **Our Approach:**
  - Quizzes: 4 to 8.
  - Assignments: 4 to 8.
  - 2 sessional exams and 1 final exam.

How Will You Be Graded?

- Homework Assignments: 10%
- Quizzes: 10%
- Sessional: (15+15=30)%
- Final Exam: 50%

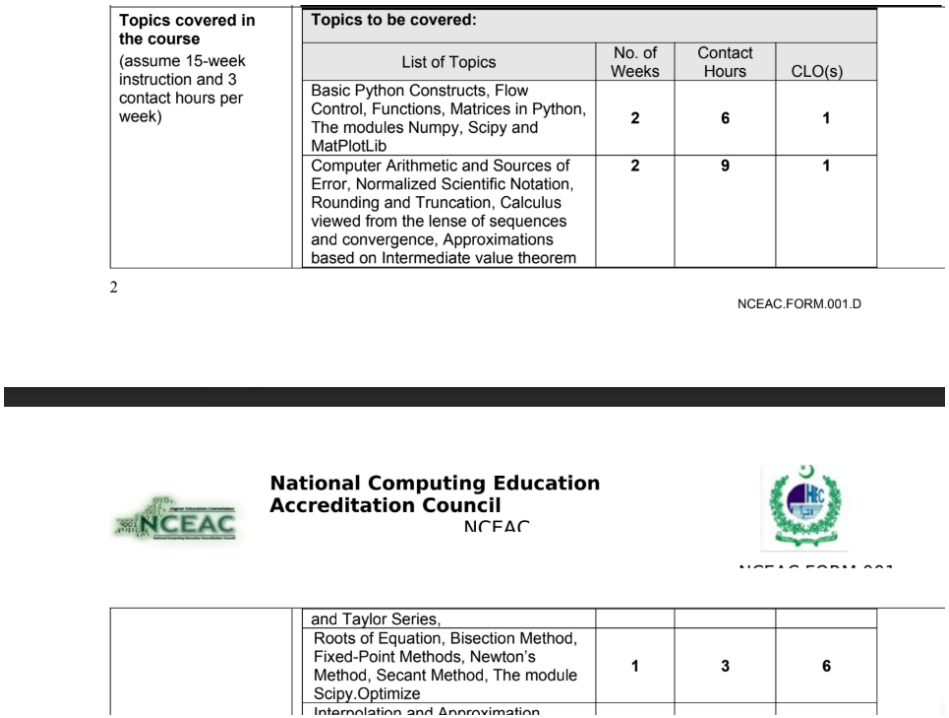
Resources for Success

- Textbook: *Numerical Methods for Engineers*.
- Python (with libraries for numerical computing).
- Office Hours: Will share on GCR.



GitHub Repository:
`https://github.com/dmitsot/`

Course Outline (availible on GCR)



Topics covered in the course (assume 15-week instruction and 3 contact hours per week)	Topics to be covered:			
	List of Topics	No. of Weeks	Contact Hours	CLO(s)
	Basic Python Constructs, Flow Control, Functions, Matrices in Python, The modules Numpy, Scipy and MatPlotLib	2	6	1
	Computer Arithmetic and Sources of Error, Normalized Scientific Notation, Rounding and Truncation, Calculus viewed from the lense of sequences and convergence, Approximations based on Intermediate value theorem	2	9	1

2

NCEAC.FORM.001.D

National Computing Education Accreditation Council

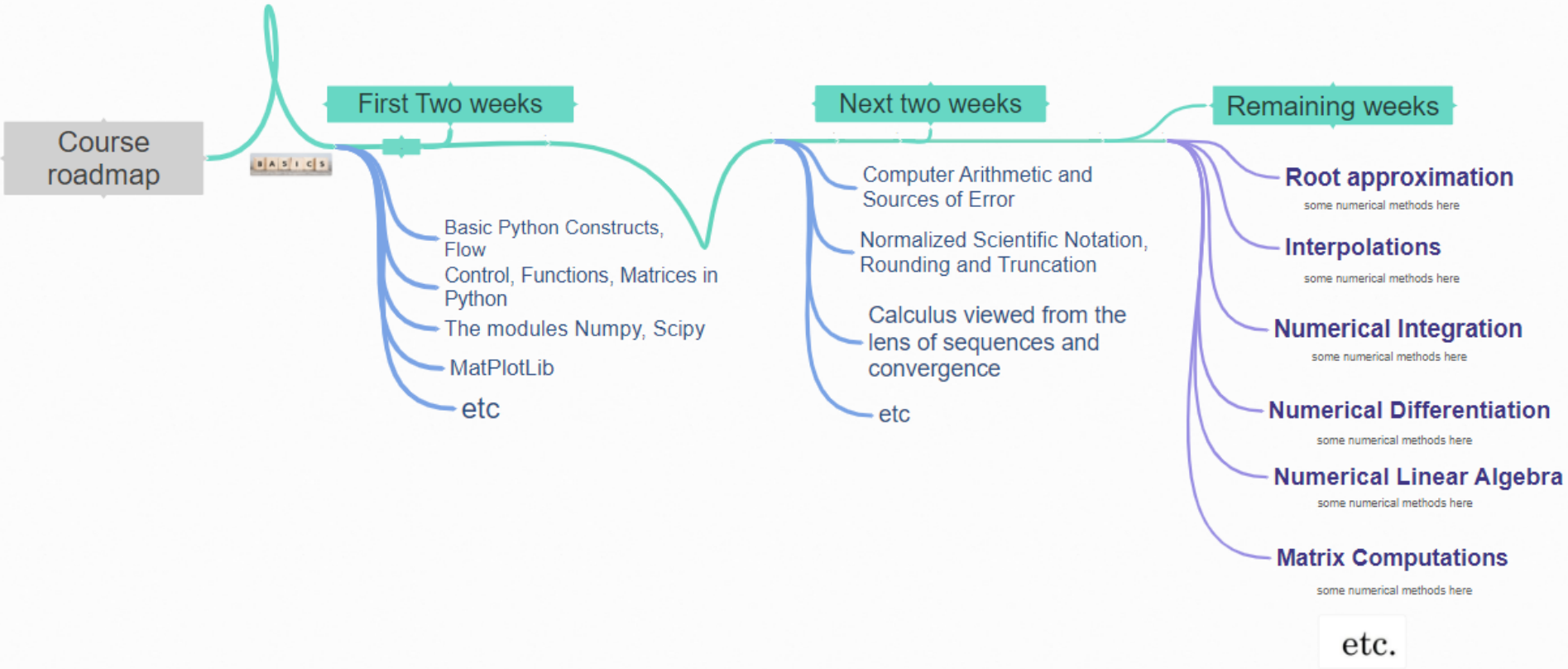
NCEAC

	and Taylor Series,			
	Roots of Equation, Bisection Method, Fixed-Point Methods, Newton's Method, Secant Method, The module Scipy.Optimize	1	3	6
	Interpolation and Approximation			

Course Outline Details:

- Week 1-5: Introduction to Numerical Computing, Methods for Solving Equations, Numerical Integration and Differentiation.
- Week 7-13: Linear Algebra, Eigenvalues and Eigenvectors, Optimization Methods.
- Week 14-16: Case Studies and Project Work.

Course Outline (availible on GCR)



Let's Get Started with Numerical Methods!

Questions? Feel free to ask.

Welcome to CS2008: Numerical Computing!