The Hong Kong University of Science and Technology

UG Course Syllabus

Course Title: Numerical Methods in Engineering

Course Code: MECH 4740

No. of Credits: 3

Pre-requisites: MATH2111 or MATH 2121 or MATH 2350 or MATH2351

Name: Xiaotian Zhang

Email: xiaotian@ust.hk

Course Description

Numerical methods are at the core of modern engineering. Many real systems cannot be solved analytically, and reliable computation is often the only way to predict performance, explore designs, and make decisions from models and data. In the era of rapidly advancing computing and AI, engineers are expected not only to use simulation tools, but also to understand the numerical principles behind accuracy, stability, and efficiency. This course provides that foundation and connects it to real engineering practice in both academia and industry.

The course covers a broad range of essential topics for numerical solutions to engineering problems, including numerical error analysis, root finding, systems of linear algebraic equations, numerical differentiation and integration, and numerical methods for ordinary and partial differential equations. Optimization techniques are also introduced to connect numerical computation to engineering design. Throughout the course, students will learn both the theory behind each method and the approaches for effective implementation.

This course is intended for senior undergraduate students and early graduate students in engineering. It has three main objectives: (1) to teach the basic theories and fundamentals of numerical methods, (2) to help students gain skills to implement these methods for computer-based solutions, and (3) to provide an environment where students can familiarize themselves with many today's popular commercial software systems, such as MATLAB or Python, and their use in solving real engineering problems.

The course includes two weekly lectures and one tutorial session. In addition to exams, students will complete a group final project where they can select a topic of interest, ranging from real world engineering problems to research oriented explorations.

Intended Learning Outcomes (ILOs)

By the end of this course, students should be able to:

1. Demonstrate a thorough understanding of the fundamental mathematical principles, numerical theories, and computational algorithms underlying modern numerical methods for engineering analysis and scientific computing.
2. Develop MATLAB and/or Python programs to implement representative numerical methods and computational algorithms for solving engineering problems.

3. Use advanced engineering computing tools and built-in computational functions in MATLAB and/or Python to analyze engineering problems, perform numerical computations, and interpret computational results effectively.

Assessment and Grading

This course will be assessed using criterion-referencing and grades will not be assigned using a curve. Detailed rubrics for each assignment are provided below, outlining the criteria used for evaluation.

Assessments:

Assessment Task	Contribution to Overall Course grade (%)	Due date
Homework	20%	Throughout the term
Mid-Term	20%	Around week 7-8
Final Project	30%	The week before final examination
Final examination	30%	Final examination period

Assessment marks for individual assessed tasks will be released within two weeks of the due date.

Mapping of Course ILOs to Assessment Tasks

Assessed Task	Mapped ILOs	Explanation
Homework	ILO1, ILO2, ILO3	Homework assesses students' understanding of fundamental numerical methods and computational algorithms (ILO1), their ability to develop MATLAB/Python programs to implement these methods (ILO2), and their proficiency in applying engineering computing tools to analyze numerical results and solve practical engineering problems (ILO3).
Mid-Term	ILO1, ILO2	The mid-term examination assesses students' understanding of the mathematical foundations and theoretical concepts of numerical methods (ILO1), as well as their ability to interpret, analyze, and perform numerical computations using given algorithm implementations (ILO2).
Final Project	ILO1, ILO2, ILO3	The final project assesses students' ability to integrate numerical theories (ILO1), develop computational solutions through programming (ILO2), and employ advanced engineering computing tools to analyze, interpret, and solve a realistic engineering problem individually or as part of a team (ILO3).
Final examination	ILO1, ILO2	The final examination assesses students' understanding of the theoretical foundations of numerical

		methods (ILO1) and their ability to analyze, evaluate, and perform engineering computations based on representative numerical algorithm implementations (ILO2).

Grading Rubrics

Grading policy for Homework, Mid-Term Examination, and Final Examination:

Homework, the mid-term examination, and the final examination are graded based on the total marks earned for each question. Marks are awarded according to the correctness of the mathematical formulation or derivation, understanding and application of numerical methods, analysis of numerical algorithms and computational procedures, computational implementation (where applicable), and interpretation of numerical results. Partial credit may be awarded for demonstrating correct reasoning, appropriate solution strategies, and key intermediate steps, even if the final answer is incomplete or incorrect.

Rubric for Final Project: 10-point final project assessment with five equally weighted categories.

Assessment Criterion	Marks	Assessment Description
Problem Background and Motivation	2	Clearly introduces the engineering or scientific problem, articulates its significance and motivation, and establishes the objectives of the project.
Problem Formulation and Numerical Methodology	2	Presents a rigorous problem formulation, including appropriate assumptions, mathematical models, and sufficient details of the selected numerical methods and computational approach.
Numerical Implementation	2	Demonstrates correct and effective implementation of numerical algorithms using appropriate engineering computing tools. The implementation reflects concepts and techniques covered in the course and is well organized and documented.
Results and Discussion	2	Presents complete computational results and critically evaluates the performance of the numerical method through appropriate analyses, such as convergence behavior, error analysis, numerical stability, computational efficiency, and, where applicable, the effects of stiffness or other numerical challenges. Discusses the validity of the results, limitations of the approach, and potential future improvements.
Presentation and Q&A	2	Delivers a clear, well-structured, and professional presentation. Demonstrates effective

		communication, answers questions accurately and confidently, and exhibits a strong understanding of the project and underlying numerical methods.

Final Grade Descriptors:

Grades	Short Description	Elaboration on subject grading description
A	Excellent Performance	Demonstrates a comprehensive understanding of numerical methods and their mathematical foundations. Develops accurate, efficient, and well-structured computational solutions, applies advanced engineering computing tools proficiently, and analyzes numerical results with excellent technical judgment. Consistently exceeds the expected learning outcomes through rigorous reasoning, creativity, and independent problem-solving.
B	Good Performance	Demonstrates solid understanding of the fundamental concepts and algorithms of numerical methods. Develops correct computational implementations with minor shortcomings and effectively applies engineering computing tools to solve engineering problems. Shows good analytical ability, sound technical judgment, and consistent achievement of the intended learning outcomes.
C	Satisfactory Performance	Demonstrates adequate understanding of the core concepts and numerical algorithms. Produces computational solutions that correctly address familiar engineering problems with some guidance and performs basic analysis and interpretation of numerical results. Achieves the essential intended learning outcomes at an acceptable standard.
D	Marginal Pass	Demonstrates only a basic understanding of fundamental numerical methods and limited ability to implement or apply computational techniques. Can solve simple engineering problems with significant guidance but exhibits weaknesses in programming, analysis, or interpretation. Meets only the minimum requirements for passing the course.
F	Fail	Demonstrates insufficient understanding of the mathematical foundations and numerical methods, is unable to implement or apply computational techniques appropriately, and shows inadequate ability to analyze engineering problems or interpret computational results. Does not meet the minimum intended learning outcomes of the course.

Course AI Policy

The use of generative artificial intelligence (AI) tools (e.g., ChatGPT, GitHub Copilot, or similar systems) is permitted in this course as a learning aid to support conceptual understanding, brainstorming, debugging, programming, code optimization, and improving the clarity of written work. However, students remain fully responsible for the accuracy, originality, and integrity of all submitted work.

For homework and project assignments, students may use AI tools to assist with developing, explaining, or refining computer programs. However, students must ensure that the final submitted code and analysis reflect their own understanding and judgment. AI-generated code should be carefully verified, tested, and, where appropriate, modified by the student before submission. Students should be able to explain and justify every part of their submitted work, regardless of whether AI tools were used in its development. Any substantive use of AI that contributes to submitted work should be acknowledged where appropriate.

The use of AI tools is not permitted during any examinations.

Communication and Feedback

Assessment marks for individual assessed tasks will be communicated via Canvas within two weeks of submission whenever practicable. Feedback on assignments will include comments on the correctness of the answers to written problems, numerical method selections and programming implementations, computational results, and areas for improvement. Students who have questions regarding the feedback or awarded marks should consult the instructor within five working days after the feedback is released.

Homework solutions, examination solutions, and discussions of project ideas, implementation strategies, and programming approaches will be provided during the weekly tutorial sessions led by the Teaching Assistant. These sessions are intended to reinforce key course concepts, address common difficulties, and provide guidance on the application of numerical methods to engineering problems

Late Submission Policy

Homework submitted after the specified deadline will be accepted but will incur a late penalty. The final awarded mark will be 80% of the earned mark (i.e., a 20% deduction). For example, a homework that earns 90 marks will receive a final score of 72 marks if submitted late. The late penalty may be waived if prior approval has been granted by the instructor or valid supporting documentation is provided.

Required Texts and Materials

The main textbook for this course is:

- *Numerical Methods for Engineers* by S.C. Chapra and R.P. Canale, 6th edition.

Other materials, including lecture notes, homework assignments and answers, exam answers and project instructions will be provided via Canvas.

Academic Integrity

Students are expected to adhere to the university's academic integrity policy. Students are expected to uphold HKUST's Academic Honor Code and to maintain the highest standards of academic integrity. The University has zero tolerance of academic misconduct.

Students are encouraged to discuss course materials, programming approaches, and project ideas with classmates, the instructor, and teaching assistants. However, all submitted assignments, programs, reports, and examinations must represent the student's own work unless the task is explicitly designated as group work (e.g., the Final Project).

Students should refer to Academic Integrity | HKUST – Academic Registry for the University's definition of plagiarism and ways to avoid cheating and plagiarism.

Course Outline

- **Part One: Modeling, Computers, and Error Analysis**
  Topic 1: Mathematical Modeling and Engineering
  Topic 2: Programming and Software
  Topic 3: Approximations and Round-Off Errors

Topic 4: Truncation Errors and the Taylor Series

- **Part Two: Roots of Equations**
  Topic 5: Bracketing Methods
  Topic 6: Open Methods
  Topic 7: Roots of Polynomials
  Topic 8: Engineering Applications: Roots of Equations
- **Part Three: Linear Algebraic Equations**
  Topic 9: Gauss Elimination
  Topic 10: LU Decomposition and Matrix Inversion
  Topic 11: Special Matrices and Gauss-Seidel
  Topic 12: Engineering Applications: Linear Algebraic
- **Part Four: Numerical Differentiation and Integration**
  Topic 13: Newton-Cotes Integration Formulas
  Topic 14: Integration of Equations
  Topic 15: Numerical Differentiation
  Topic 16: Engineering Applications: Numerical Integration and Differentiation
- **Part Five: Ordinary Differential Equations**
  Topic 17: Runge-Kutta Methods
  Topic 18: Stiffness and Multistep Methods
  Topic 19: Boundary-Value and Eigenvalue Problems
  Topic 20: Engineering Applications: Ordinary Differential Equations
- **Part Six: Partial Differential Equations**
  Topic 21: Finite Difference: Elliptic Equations
  Topic 22: Finite Difference: Parabolic Equations
  Topic 23: Finite-Element Method
  Topic 24: Engineering Applications: Partial Differential Equations
- **Part Seven: Optimization**
  Topic 25: One-Dimensional Unconstrained Optimization
  Topic 26: Multidimensional Unconstrained Optimization
  Topic 27: Constrained Optimization
  Topic 28: Engineering Applications: Optimization
- **Part Eight: Curve Fitting**
  Topic 29: Least-Squares Regression
  Topic 30: Interpolation
  Topic 31: Fourier Approximation
  Topic 32: Engineering Applications: Curve Fitting