COURSE OUTLINE / PLAN DE COURS

LaurentianUniversity
UniversitéLaurentienne

CPSC-5506EL-01

Introduction to Computational Science

Spring
2023

Calendar Description This course introduces Computational Science from the point of view of applied mathematics, computer science, and science and engineering. Topics may include finite-difference methods, fast-Fourier transform, compression methods and Monte-Carlo simulations.

– Laurentian University Online Course and Program Catalogue (2023)

The program and the university reserve the right to modify elements of the course during the term. The university may change the dates and deadlines for any or all courses in extreme circumstances. If modifications become necessary, reasonable notice and communication with the students will be given. Students will be provided with an explanation and an opportunity to comment.

Prerequisites: Admission to the M.Sc. in Computational Sciences or permission of the instructor

Instructor:
Wenfeng Chen
Sessional Lecturer, Laurentian University

Office: FA 383
Telephone: (705) 675-1151 Extension 3342
Email: wchen@laurentian.ca

Lectures: Th 4:00 PM -6:50 PM C-206

Tutorials: 2:50 PM – 3:50 PM C-205

Labs: All of the computational examples will be presented using **MATLAB**. This software is available on the computers in the computer lab. In addition, Laurentian University has purchased a campus-wide license of MATLAB and Simulink for all students and faculty members. Students can come here https://www.mathworks.com/login?uri=%2Flicensecenter%2Flicenses%2Fadd%3Fs_v1%3Delg-chk-yes-1
to create an account and then follow the instructions to download MATLAB and install on your own PC or Laptop. It is recommended that students grasp basic MATLAB programming skills.

Attendance:

1. Students must attend classes regularly. A student must sign the attendance sheet after each class is dismissed;
2. A student, who is unable to write midterm test because of illness or other circumstances beyond his/her control and wants to write the make-up, should bring the instructor with an appropriate document such as a medical doctor’s note. The instructor will arrange an alternative test for the student within a short time after the normal test.
3. A student, who is unable to write final exam because of illness or other circumstances beyond his/her control, should look up the relevant rules of Laurentian University
4. A student with disability should inform the instructor so that the accommodation for the student to write midterm tests and final exam can be arranged

Website/ page web: ***Course materials and announcements will be posted on D2L (Desire 2 Learn). It is the student's responsibility to check the D2L page for course information throughout the semester***

Textbook: Textbooks/Reference Books

1. W. Cheney and D. Kincaid, *Numerical Mathematics and Computing*, 7th edition, Brooks/Cole (2013)
2. R. Burden and J.D. Faires, *Numerical Analysis*, 9th edition, Brooks/Cole (2011)
3. A.Kharab and R.B. Guenther, *An Introduction to Numerical Methods: MATLAB Approach,* CRC Press, 4th edition, 2018;
4. Uri M. Ascher and Chen Greif, *A First Course in Numerical Methods*, SIAM, 2011
5. C.V. Loan, *Computational Frameworks for the Fast Fourier Transform*, SIAM, 1992.
6. E.O. Brigham, *The Fast Fourier Transform*, Prentice-Hall, 1974.
7. E.K.P. Chong and S.H. Zak, *An Introduction to Optimization*, Wiley, 2001.
8. W. Sun and Y-X Yuan, *Optimization Theory and Methods*, Springer, 2006.

Office Hours: M/W 5:50 PM-6:50 PM, T: 4:20 PM-5:20 PM, Th: 2:50 PM-3:50 PM C-205; email for appointment other time

Assessment:

Assessment Type	Number	Overall Weight (%)
Midterm Tests	2	25
Homework	5	25
Lab	3	15
Final Exam	1	30
Class Attendance and Lecture Notes Taking		5

Final Grades: **The final grades will be consistent with the University Grading System.**
For information regarding appeals of final grades or other academic matters please consult the university's intranet site.

Add any program-specific requirements for this course here

LU Grading Scheme/ Système de notation

Grade Appeal Policy / Politique d'appel de note

Course Content: **Part 1: Finite Difference Methods and Applications**

(1). Definition of finite difference from the discretization of a continuous function and general definition and properties of finite difference for discrete functions.

(2). Application of finite difference in numerical differentiation and MATLAB programming.
Forward-, backward- and central-divided difference approximation to the derivative of a differentiable function. Richardson's extrapolation technique for improving truncation error in the finite difference approximation.

(3). Application of finite difference in numerical integration and MATLAB programming.
Review on Newton's divided-difference interpolating polynomial and truncation error analysis, and Lagrange interpolating polynomial; Closed and open Newton-Cotes formulas and error analysis; Trapezoidal rule, Simpson's rule and middle point rule; Romberg algorithm for improving truncation error and Gaussian quadrature.

(4). Application of finite difference in solving initial value problems of ordinary differential equations and systems of fist-order differential equations and MATLAB programming.
Introducing various numerical methods including one-step methods such as Euler's method, higher-order Taylor methods and Runge-Kutta methods, and multistep methods such as Adams-Bashforth methods, Adams-Moulton methods and predictor-corrector methods. Discussion on numerical stability and error analysis.

(5). Application of finite difference in solving boundary value problems of ordinary differential equations and MATLAB programming.
Shooting methods for linear and nonlinear problems; finite difference methods for linear and nonlinear problems; Eigenvalue method and *the Rayleigh-Ritz variation method
(6). Application of finite difference in solving partial differential equations and MATLAB programming. Apply finite difference to solve elliptic, hyperbolic and hyperbolic partial linear differential equations *and a number of nonlinear partial differential equations such as Burger's equation, reaction-diffusion equation. Porous media equation and Hamilton-Jacobi-Bellman equation etc. *Introduction to finite-element method.

Part 2: Fourier Transform, Discrete Fourier Transform, Faster Fourier Transform and Applications
(1). Review on Fourier series, continuous Fourier transform and inverse Fourier transform and properties as well as MATLAB implementation.
(2). Definition, two typical features including reciprocity and periodicity, properties such as linearity and shifting property, matrix forms and MATLAB coding of discrete Fourier transform and inverse Discrete transform;
(3). Basic idea of fast Fourier transform. Definition and algorithm of performing radix-2 FFT, matrix form and MATLAB implementation; signal flow graph and bit reversal graph of the general radix-2 algorithm; Definition and algorithms of performing radix-3 FFT, matrix forms and MATLAB implementations. The Cooley-Tukey algorithm for radix-2 FFT and MATLAB coding. General radix frameworks including mixed-radix factorization, radix-4 and radix-8 frameworks and MATLAB implementation.

Part 3: Monte Carlo Methods and Simulation
(1). Definition of random numbers; random number generating algorithm and MATLAB coding, characteristic of random number algorithms
(2). Numerical computation on double and triple integrals by the Monte Carlo method and further application in estimating the area and volume by using random sequence of points, and MATLAB coding.
(3). Simulation on some typical problems by the Monte Carlo method including loaded die problem, Buffon's needle problem, two dice problem and neutron shielding problem etc.
*(4) Random walks and the Metropolis algorithm
*(5). Application of the Monte Carlo methods in statistical physics and quantum physics.

***Part 4: Unconstrained and constrained nonlinear optimization**
(1). Introduction to optimization theory; Review on mathematical notions and theorem involved in optimization including minimizer, stationary points, gradient and Hessian matrix of a function, sequence, convex set and convex function etc.; Optimality conditions including necessary and sufficient conditions at both first- and second-order for unconstrained optimization;
(2). Some necessary notions in constrained optimization theory such as feasible set, feasible directions, descent directions, active and inactive constraints etc.; Optimality conditions including first-order necessary condition (KKT condition) and second-order sufficient condition for constrained optimization;
(3). Introduction to descent iteration algorithm; Line search methods including Golden Section Method, Newton's method, and Quadratic Interpolation Method etc.;
(4). Classical methods for solving unconstrained optimization problems including Steepest Descent Method, Newton's Method, Conjugate Gradient Method and Quasi-Newton Methods;
(5). Classical methods for solving constrained optimization problems including Penalty Function Methods and Augmented Lagrange Method

Assignments:

1. Weekly Homework will be assigned weekly after Thursday class and should be submitted **biweekly**, i.e., submitted on **every odd number Thursday class starting from 3rd week,** i.e., submitted on 3rd, 5th, 7th, 9th and 11th week. It will be **marked and recorded as a part of evaluation**. Late submission will not be accepted except that a reasonable excuse is provided.

2. Lab Homework Assignment will be assigned triweekly and should be submitted **triweekly**, That is, the assignment on 3rd , 6th , 9th week and students submit on Friday **every 4th number week (**4th, 8th , 11th week**)** . It will be **marked and recorded as a part of evaluation**. Late submission will not be accepted except that a reasonable excuse is provided

Exams:

1. All exam/tests will be closed book. Students will be only allowed a black pen or pencil and an ordinary scientific calculator, the exams will be similar (but not necessarily identical) to the homework and classroom examples, and will include problems with different degrees of difficulty, from basic calculations to challenging problems. The solutions of all exam problems must be presented in details and in a legible and neat way. Sloppy and/or illegible presentation will result in reduction of points.
2. A student, who is unable to write midterm test because of illness or other circumstances beyond his/her control and wants to write the make-up, should bring the instructor with an appropriate document such as a medical doctor's note. The instructor will arrange an alternative test for the student within a short time after the regular test.
3. A student, who is unable to write final exam because of illness or other circumstances beyond his/her control, should look up the relevant rules of Laurentian University
4. A student with disability should inform the instructor so that the accommodation for the student to write midterm tests and final exam can be arranged

Student Conduct: **Students will be expected to abide by the Laurentian University Code of Conduct.**
Code of Student Rights and Responsibilities / Énoncé des droits et responsabilités des étudiantes et étudiants

Academic Integrity: **In this course, students are expected to submit their own individual work for academic credit, properly cite the work of others, and to follow the rules for examinations. Academic misconduct, including plagiarism and cheating, will not be tolerated. Copying of assignments and lab reports is considered academic misconduct. Students are responsible for understanding and following the Laurentian University Policy on Academic Integrity.**

Student Academic Integrity / Intégrité intellectuelle de la population étudiante, politique et procédure

Safety: **The Faculty of Science, Engineering and Architecture takes safety very seriously. Students are expected to work in a safe manner, follow all safety instructions, and use any personal protective equipment required or provided. Students failing to observe the safety rules in any laboratory will be asked to leave.**

Learning Outcomes: **By the end of this course, students will be able to:**

Upon completion of the course a student will be able to

1. Know the application of finite difference methods and MATLAB implementation in doing numerical differentiation and how to improve the truncation error
2. Know how to apply the closed and open Newton-Cotes formulas and Gaussian quadrature to do numerical integration and use Romberg algorithm to improve truncation error and further, how to do MATLAB coding.
3. Know various numerical methods including one-step methods such as Euler's method, higher-order Taylor methods and Runge-Kutta methods, and multistep methods such as Adams-Bashforth methods, Adams-Moulton methods and predictor-corrector methods to solve initial value problems of ordinary differential equations and then program with MATLAB

4. Know how apply some numerical methods such as shooting methods finite difference methods to solve boundary–value problems for linear and nonlinear ordinary differential equations and how to do MATLAB coding.
5. Know how to apply finite difference to solve elliptic, hyperbolic and hyperbolic partial linear differential equations and a number of nonlinear partial differential equations numerically with computer programs.
6. Know the origin of discrete Fourier transform and fast Fourier transform, the radix-2 frameworks, especially the Cooley-Tuckey algorithm for the fast Fourier transform and general radix frameworks for the fast Fourier transform and the corresponding MATLAB implementation.
7. Understand the basic idea of the Monte Carl methods and know its application in approximating multidimensional integrals and simulating some typical problems in mathematics and physics.
8. *Understand basic ideas of unconstrained and constrained optimizations and various algorithms of unconstrained and constrained optimizations and the corresponding MATLAB programming.