



National University of Engineering (UNI)
School of Computer Science
Syllabus 2027-I

1. COURSE

MA202. Numerical Methods (Mandatory)

2. GENERAL INFORMATION

2.1 Course	: MA202. Numerical Methods
2.2 Semester	: 4 th Semester.
2.3 Credits	: 3
2.4 Hours	: 2 HT; 2 HP;
2.5 Duration of the period	: 16 weeks
2.6 Type of course	: Mandatory
2.7 Learning modality	: Face to face
2.8 Prerequisites	: BMA103. Integral Calculus. (2 nd Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

Numerical methods are a fundamental tool shared by computer science and engineering disciplines for approximating solutions to mathematical problems that cannot be solved analytically. This course provides an introduction to the most common numerical methods, including error analysis, solution of nonlinear equations, interpolation, numerical integration, and solution of differential equations, in scientific and engineering contexts.

5. GOALS

- Understand the importance of numerical methods in solving scientific and engineering problems.
- Apply different numerical methods to approximate solutions to mathematical problems.
- Analyze the accuracy and efficiency of the numerical methods used.

6. COMPETENCES

AG-C08) Problem Analysis: Identifies, formulates, and analyzes complex computing problems. (Usage)

AG-C09) Design and Development of Solutions: Designs, implements, and evaluates solutions for complex computing problems. (Usage)

AG-C11) Use of Tools: Applies modern computing tools in problem solving. (Usage)

7. TOPICS

Unit 1: Error Analysis and Floating-Point Arithmetic (6 hours)	
Competences Expected: AG-C08,AG-C11	
Topics	Learning Outcomes
• IEEE 754 floating-point representation: machine epsilon, overflow, underflow, and rounding modes • Rounding errors, catastrophic cancellation, and loss of significance • Forward and backward error analysis; condition number of a problem • Numerical stability: stable vs. unstable algorithms; Wilkinson's backward error analysis • Error propagation in arithmetic operations and function evaluation	• Explain the IEEE 754 floating-point standard and identify the sources of rounding and cancellation errors [Familiarity] • Estimate the condition number of a problem and predict the accuracy of a numerical result [Usage] • Analyze an algorithm for numerical stability using forward and backward error bounds [Assessment]
Readings : [ChapraCanale2015], [BurdenFaires2010]	

Unit 2: Root-Finding for Nonlinear Equations (10 hours)	
Competences Expected: AG-C08,AG-C11	
Topics	Learning Outcomes
• Bisection method: interval bracketing, stopping criteria, and error bound • Newton-Raphson method: derivation, quadratic convergence, and failure cases • Secant method as an alternative to Newton's method without an explicit derivative • Order of convergence and efficiency comparison among iterative methods • Extension of Newton's method to systems of nonlinear equations via the Jacobian matrix, the basis of solvers used in machine learning	• Compare the convergence guarantees of the bisection, Newton-Raphson, and secant methods [Familiarity] • Implement the Newton-Raphson method to approximate roots of a nonlinear function [Usage] • Analyze the order of convergence of an iterative method and justify its choice based on computational cost [Assessment]
Readings : [ChapraCanale2015], [BurdenFaires2010]	

Unit 3: Approximation Theory and Interpolation (10 hours)	
Competences Expected: AG-C08,AG-C11	
Topics	Learning Outcomes
• Polynomial interpolation: Lagrange, Newton divided differences, and Runge's phenomenon • Spline interpolation: cubic splines, B-splines, and piecewise polynomial methods • Best approximation in normed spaces: Chebyshev (minimax) and least squares • Chebyshev polynomials: properties, orthogonality, and spectral convergence • Trigonometric approximation and the fast Fourier transform (FFT)	• Explain Runge's phenomenon and justify the choice of Chebyshev nodes to mitigate it [Familiarity] • Construct cubic spline and Chebyshev interpolants for given data and estimate the interpolation error [Usage] • Apply the FFT to efficiently compute trigonometric approximations of a sampled function [Assessment]
Readings : [ChapraCanale2015], [BurdenFaires2010]	

Unit 4: Numerical Integration and Quadrature (10 hours)	
Competences Expected: AG-C08,AG-C11	
Topics	Learning Outcomes
• Newton-Cotes rules: trapezoidal, Simpson's, and composite rules; error analysis • Adaptive quadrature and automatic error control • Gaussian quadrature: optimal nodes and weights, orthogonal polynomial connection • Numerical treatment of improper and singular integrals • Monte Carlo integration and quasi-Monte Carlo methods for high-dimensional integrals	• Derive the error formula for composite Simpson's rule and identify its order of accuracy [Familiarity] • Select and apply an appropriate quadrature rule (Gaussian, adaptive) based on integrand regularity [Usage] • Apply Monte Carlo integration to estimate high-dimensional integrals and quantify the statistical error [Assessment]
Readings : [ChapraCanale2015], [BurdenFaires2010]	

Unit 5: Numerical Methods for Differential Equations (12 hours)	
Competences Expected: AG-C08,AG-C11	
Topics	Learning Outcomes
• Runge-Kutta methods: Euler, RK4, and embedded methods for error control • Linear multistep methods: Adams-Bashforth, Adams-Moulton, and BDF methods • Stability analysis: zero-stability, absolute stability regions, and stiff ODEs • Finite difference methods for parabolic and elliptic PDEs: stability and convergence	• Compare explicit and implicit ODE solvers and explain their relative stability for stiff problems [Familiarity] • Implement a Runge-Kutta method with adaptive step control and apply it to a system of ODEs [Usage] • Analyze the stability and convergence of a finite difference scheme for a parabolic PDE [Assessment]
Readings : [ChapraCanale2015], [BurdenFaires2010]	

Unit 6: Numerical Linear Algebra (12 hours)	
Competences Expected: AG-C08,AG-C11	
Topics	Learning Outcomes
• Gaussian elimination with partial pivoting, LU factorization, and complexity analysis • QR factorization via Householder and Gram-Schmidt; least squares problems • Eigenvalue algorithms: power iteration, QR algorithm, and the Lanczos method • Singular value decomposition: computation, truncation, and applications (PCA, pseudoinverse) • Iterative methods for large sparse systems: CG, GMRES, and preconditioning	• Compare direct and iterative solvers and select the appropriate method for a given matrix structure [Familiarity] • Apply QR factorization and SVD to solve least squares problems and compute low-rank approximations [Usage] • Analyze the numerical stability and computational cost of eigenvalue algorithms [Assessment]
Readings : [BurdenFaires2010], [TrefethenBau1997]	

Unit 7: Finite Element Method (8 hours)	
Competences Expected: AG-C08,AG-C09	
Topics	Learning Outcomes
• Weak (variational) formulation of boundary value problems and Sobolev spaces • Galerkin method: finite-dimensional approximation spaces and the stiffness matrix • Triangular and quadrilateral finite element spaces: Lagrange elements and conformity • A priori error estimates: Céa's lemma and interpolation error bounds • A posteriori error estimates and adaptive mesh refinement	• Derive the weak formulation of an elliptic BVP and show its equivalence to the strong form [Familiarity] • Assemble the global stiffness matrix and load vector for a linear finite element discretization [Usage] • Estimate the H^1 error of a finite element solution using Céa's lemma and interpolation theory [Assessment]
Readings : [ReddyFEM2019], [BurdenFaires2010]	

Unit 8: Optimization Algorithms (4 hours)	
Competences Expected: AG-C09,AG-C11	
Topics	Learning Outcomes
• Gradient descent and line search methods: Armijo-Wolfe conditions and convergence rates • Newton's method and quasi-Newton methods (BFGS, L-BFGS) • Constrained optimization: KKT conditions, penalty methods, and sequential quadratic programming • Convex optimization algorithms: interior-point methods and ADMM • Stochastic gradient descent (SGD), variance reduction, and Adam optimizer	• Explain the convergence guarantees of gradient descent for smooth convex functions [Familiarity] • Apply Newton and quasi-Newton methods to unconstrained optimization problems [Usage] • Formulate a constrained optimization problem as a KKT system and apply an interior-point method [Assessment]
Readings : [NocedalWright2006]	

Unit 9: Parallel and High-Performance Computing (4 hours)	
Competences Expected: AG-C11	
Topics	Learning Outcomes
• Parallel architectures: shared memory (OpenMP), distributed memory (MPI), and GPU (CUDA) • Performance models: Amdahl's law, roofline model, and communication complexity • Parallel algorithms for dense and sparse linear algebra (ScaLAPACK, PETSc) • Domain decomposition and parallel PDE solvers • Automatic differentiation (forward and reverse mode) and its role in scientific ML	• Explain Amdahl's law and identify the bottlenecks limiting parallel speedup in a given algorithm [Familiarity] • Implement a parallel linear algebra routine using MPI or OpenMP and measure parallel efficiency [Usage] • Apply automatic differentiation to compute gradients of complex scientific functions for optimization [Assessment]
Readings : [PachecoMalensek2022]	

Unit 10: Graph Theory (4 hours)	
Competences Expected: AG-C08	
Topics	Learning Outcomes
• Graphs: definitions, isomorphism, degree sequences, trees, and spanning trees • Connectivity, Menger's theorem, and network flows (max-flow min-cut) • Matchings (Hall's theorem), graph colorings, and the chromatic polynomial • Planar graphs, Euler's formula, Kuratowski's theorem, and the four-color theorem • Adjacency and Laplacian matrices, eigenvalues, and expander graphs	• Identify structural properties of graphs (connectivity, planarity, bipartiteness) and apply Euler's formula [Familiarity] • Apply Hall's marriage theorem and network flow algorithms to matching and routing problems [Usage] • Analyze a graph's spectrum to bound its chromatic number and connectivity properties [Assessment]
Readings : [West2001]	

Unit 11: Time Series Analysis (2 hours)	
Competences Expected: AG-C08,AG-C11	
Topics	Learning Outcomes
• Stationarity, autocovariance function, and the autocorrelation function (ACF) • ARMA models: identification, estimation (Yule-Walker, MLE), and diagnostics • Spectral density, the periodogram, and Wiener-Khinchin theorem • State-space models and the Kalman filter • ARIMA models, seasonal adjustment, and multi-step forecasting	• Identify stationarity and determine the order of an ARMA model from ACF and PACF plots [Familiarity] • Fit ARIMA models to time series data, validate residuals, and produce forecasts [Usage] • Apply the Kalman filter to estimate hidden states in a linear Gaussian state-space model [Assessment]
Readings : [ShumwayStoffer2017]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. EVALUATION SYSTEM

Theory Sessions:

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

Lab Sessions:

In order to verify their competences, several activities including active learning and roleplay will be developed during lab sessions.

Oral Presentations:

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

Reading:

Throughout the course different readings are provided, which are evaluated. The average of the notes in the readings is considered as the mark of a qualified practice. The use of our virtual campus allows each student to access the course information, and interact outside the classroom with the teacher and with the other students.

The final grade NF depends on the following grades:

- PE : Partial Exam.
- FE : Final Exam.
- P : Permanent Evaluation (Average of practical work and graded assignments).

The final grade NF is calculated as follows:

$$NF = \frac{PE + FE + P}{3}$$

10. BASIC BIBLIOGRAPHY