Anekant Education Society's

Tuljaram Chaturchand College of Arts, Science & Commerce, Baramati

(*Empowered Autonomous*)

Three/Four Year Honours/Honours with Research B.Sc. (Comp. Sci.)
Degree Program in Computer Science
(Faculty of Science)

CBCS Syllabus
S.Y.B.Sc. (Comp. Sci.) Mathematics
For Department of Mathematics

NEP-2.0
Choice Based Credit System Syllabus
(2024 Pattern)

(As Per NEP-2020)

To be implemented from Academic Year 2025-26

CBCS Syllabus as per NEP 2020 for S.Y.B.Sc. (Comp. Sci.) Mathematics (2024 Pattern)

Name of the Programme	**:** B.Sc. (Comp. Sci.)
Program Code	**:** USCOS
Class	**:** S.Y.B.Sc. (Comp. Sci.)
Semester	**:** IV
Course Type	**:** Minor
Course Name	**:** Numerical Analysis
Course Code	**:** COS-256-MN (B)
No. of Teaching Hours	**:** 30
No. of Credits	**:** 2

Course Objectives:

1. To understand types of errors and their significance in numerical computations.
2. To solve algebraic and transcendental equations using iterative and approximation methods.
3. To study finite differences, operators, and their applications in interpolation.
4. To apply Newton's and Lagrange's interpolation formulas for estimating unknown values.
5. To learn and implement numerical integration techniques such as Trapezoidal and Simpson's rules.
6. To analyze and solve ordinary differential equations using Euler and Runge-Kutta methods.
7. To develop computational skills for applying numerical methods to real-world mathematical problems.

Course Outcomes:

CO1: Students will be able to apply numerical methods to solve algebraic and transcendental equations using iterative techniques.

CO2: Students will demonstrate the use of finite difference operators and their relations in numerical computation.

CO3: Students will be able to construct interpolation polynomial using Newton's interpolation formulas.

CO4: Students will be able to evaluate definite integrals numerically using quadrature formulas and standard rules.

CO5: Students will implement Euler's and Modified Euler's methods to approximate solutions of ordinary differential equations.

CO6: Students will be able to apply Runge-Kutta methods of various orders for solving initial value problems numerically.

CO7: Students will be able to analyze accuracy, errors, and convergence of different numerical methods in problem-solving.

Topics and Learning Points

	Teaching Hours
Unit 1: Algebraic and Transcendental Equations	**04**
1.1 Errors	
1.2 Algebraic and Transcendental Equations	
1.3 False Position Method/Regula Falsi Method	
1.4 Newton-Raphson's Method	
Unit 2: Calculus of Finite Differences and Interpolation	**10**
2.1 Differences	
2.2 Properties of Operators	
2.3 Relation between Operators	
2.4 Newton's Gregory Formula for forward Difference Interpolation	
2.5 Newton's Gregory Formula for Backward Difference Interpolation	
2.6 Lagrange's Interpolation Formula	
Unit 3: Numerical Integration	**08**
3.1 General Quadrature Formula	
3.2 Trapezoidal Rule	
3.3 Simpson's one-third Rule	
3.4 Simpson's three-eight Rule	
Unit 4: Numerical Solution of Ordinary Differential Equation	**08**
4.1 Euler's Method	
4.2 Euler's Modified Method	
4.3 Runge-Kutta's Second-order Method	
4.4 Runge-Kutta's Fourth order Method	

Text Book:

S. S. Sastry, *Introductory Methods of Numerical Analysis*, Printice Hall of India (3rd Edition), 1999.

Reference Books:

1. H.C. Saxena, *Finite differences and Numerical Analysis*, S. Chand and Company.
2. K.E. Atkinson, *An Introduction to Numerical Analysis*, Wiley Publications.
3. Balguruswamy, *Numerical Analysis*.
4. A. K. Jaiswal and Anju Khandelwal, *A Textbook of Computer Based Numerical and Statistical Techniques*, New Age International Publishers.

CO-PO Mapping

Weightage: 1 – weak or low relation, 2 – moderate or partial relation, 3 – strong or direct relation

Programme Outcomes	Course Outcomes						
	CO1	CO2	CO3	CO4	CO5	CO6	CO7
PO01	3	3	3	3	3	3	3
PO02	2	3	3	3	3	3	3
PO03	1	1	1	2	2	2	2
PO04	2	3	3	3	3	3	3
PO05	3	3	3	3	3	3	3
PO06	1	1	1	2	2	2	2
PO07	2	2	2	3	3	3	3
PO08	2	2	2	2	3	3	3
PO09	2	3	3	3	3	3	3
PO10	1	1	1	1	1	1	1
PO11	1	1	1	1	1	1	1
PO12	2	2	2	2	3	3	3
PO13	1	1	1	1	1	1	1

Justification for the mapping

PO1: ***Comprehensive Knowledge and Understanding*** - This course develops a deep conceptual understanding of various numerical methods and their mathematical foundations. Students gain comprehensive knowledge of iterative techniques, interpolation, numerical integration, and differential equation solvers, enabling them to understand both the theory and application of these methods in scientific computation.

PO2: ***Practical, Professional, and Procedural Knowledge*** - Through solving practical computational problems using Newton-Raphson, Lagrange's interpolation, and Runge-Kutta methods, students acquire procedural proficiency and professional competence. The hands-on practice strengthens their ability to apply algorithms systematically to real-world mathematical and engineering problems.

PO3: ***Entrepreneurial Mindset and Knowledge*** **-** Although indirectly related, the course cultivates analytical and innovative thinking, which are essential components of an entrepreneurial mindset. Understanding numerical modeling and problem-solving nurtures decision-making skills and fosters a mindset for developing computational tools and solutions in technology-driven startups.

PO4: ***Specialized skills and competencies*** - Students develop specialized computational skills, such as formulating numerical models, implementing algorithms, and analyzing convergence and stability. These competencies are directly applicable in scientific research, engineering simulations, and data analysis, enhancing their technical expertise in computational mathematics.

PO5: ***Capacity for Application, Problem-Solving, and Analytical Reasoning*** - This course primarily focuses on analytical and problem-solving abilities. Students learn to select appropriate numerical methods, evaluate their accuracy, and analyse computational errors. They apply reasoning and logic to determine the most efficient methods for various mathematical models, demonstrating a high level of analytical proficiency.

PO6: ***Communication skills and collaboration*** - Students interpret numerical results, prepare tabulated data, and present outcomes using graphical or computational software. Group assignments or collaborative lab work foster teamwork, presentation, and communication skills in the context of mathematical problem-solving.

PO7: ***Research-related skills*** - By analyzing errors, convergence, and stability of algorithms, students engage in research-oriented thinking. The course encourages critical evaluation and comparison of different numerical techniques, laying the foundation for further research in applied mathematics, data science, and computational modeling.

PO8: ***Learning how to learn skills*** - Students continuously adapt to new problem-solving situations by learning and implementing diverse numerical methods. The iterative nature of computation promotes self-learning, reflection, and adaptability — essential for lifelong learning and professional development.

PO9: ***Digital and technological skills*** - The use of digital tools, software (e.g., Python, MATLAB, or GeoGebra), and computational platforms for implementing numerical algorithms directly enhances students' technological literacy and digital problem-solving skills, aligning the course strongly with this outcome.

PO10: ***Multicultural competence, inclusive spirit, and empathy*** - The course indirectly supports multicultural competence through collaborative learning and teamwork among students from diverse backgrounds. However, the course content itself remains technical, contributing minimally to this outcome.

PO11: ***Value inculcation and environmental awareness*** - While not directly related to environmental issues, the course encourages ethical practices in computation, such as accuracy, data integrity, and responsible use of digital resources, promoting scientific integrity and awareness of sustainable practices.

PO12: ***Autonomy, responsibility, and accountability*** - Assignments, computational projects, and independent algorithm implementation develop self-reliance, accountability, and responsibility. Students learn to validate their results and take ownership of their computational decisions and outcomes.

PO13: ***Community engagement and service*** - The direct connection is minimal, but the analytical and computational skills acquired may indirectly contribute to community development through applications such as modelling population growth, environmental data, or resource optimization in future projects.

CBCS Syllabus as per NEP 2020 for S.Y.B.Sc. (Comp. Sci.) Mathematics (2024 Pattern)

Name of the Programme : B.Sc. (Comp. Sci.)
Program Code : USCOS
Class : S.Y.B.Sc. (Comp. Sci.)
Semester : IV
Course Type : Minor
Course Name : Numerical Analysis Practical using Scilab Software
Course Code : COS-257-MN (B)
No. of Teaching Hours : 60
No. of Credits : 2

Course Objectives:

1. To understand and apply algorithms to determine connectivity in graphs.
2. To implement and utilize algorithms for finding shortest paths in connected graphs.
3. To apply tree structures to problem-solving in different contexts.
4. To apply matrix representations to solve graph-related problems.
5. To develop C functions for matrix operations relevant to graph representations.
6. To apply graph coloring algorithms to different types of graphs.
7. To Implement algorithms to find cycles and determine the shortest path in directed graphs.

Course Outcomes:

CO1: Student will be able to implement algorithms to find shortest paths in connected graphs.
CO2: Student will be able to understand and describe tree structures in the context of graph theory.
CO3: Student will be able to analyze and interpret properties of adjacency matrix and incidence matrix.
CO4: Write C programs for basic graph representation using an adjacency matrix.
CO5: Student will be able to develop C programs for implementing DFS and BFS algorithms for tree traversal.
CO6: Student will be able to implement C functions for matrix operations applicable to graph representations.
CO7: Student will be able to develop C functions to perform operations on directed graphs, including cycle detection and shortest path determination.

Topics and Learning Points

Teaching Hours

List of Practical: **60**

1) **Introduction to Scilab**: Installation of the software Scilab, syntax, some important commands on Scilab.
2) **Operations on Matrices**: Matrix construction, Algebraic operations on Matrices, Accessing rows and columns, Determinant and inverse of a matrix, Reduced row echelon form, Rank of a matrix.
3) **Solving systems of linear equations using Scilab.**
4) **Eigenvalues and Eigenvectors using Scilab.**
5) **Polynomial Operations in Scilab and User-defined functions.**
6) **Plotting graphs using Scilab**: 2-D graph, 3-D graph.
7) **Iterations & conditional statements in Scilab:** for statement, while statement, if statement.
8) **Finding roots of an equation using Scilab:** False Position (Regula Falsi) Method.
9) **Finding roots of an equation using Scilab:** Newton-Raphson Method.
10) **Numerical Integration using Scilab:** Trapezoidal Rule, Simpson's 1/3rd Rule, Simpson's 3/8th Rule.
11) **Numerical Differentiation using Scilab:** Euler’s Method.
12) **Numerical Differentiation using Scilab:** Runge-Kutta Method.

CO-PO Mapping

Weightage: 1 – weak or low relation, 2 – moderate or partial relation, 3 – strong or direct relation

Programme Outcomes	Course Outcomes						
	CO1	CO2	CO3	CO4	CO5	CO6	CO7
PO01	3	3	3	2	2	2	3
PO02	2	2	2	3	3	3	3
PO03	1	1	1	1	1	2	2
PO04	2	2	3	3	3	3	3
PO05	3	2	2	2	3	3	3
PO06	1	1	1	2	2	2	2
PO07	2	1	2	1	2	2	3
PO08	2	2	2	2	2	2	3
PO09	2	2	3	3	3	3	3
PO10	1	1	1	1	1	1	1
PO11	1	1	1	1	1	1	1
PO12	2	2	2	2	3	3	3
PO13	1	1	1	1	1	1	1

Justification for the mapping

PO1: ***Comprehensive Knowledge and Understanding*** **-** Strong correlation (3) with CO1–CO3 & CO7, as students gain theoretical and conceptual understanding of graph theory and shortest path algorithms. Moderate (2) with coding- based CO4–CO6 as these extend the theoretical base into implementation.

PO2: ***Practical, Professional, and Procedural Knowledge*** **-** Directly related (3) to CO4–CO7 where C programming skills and standard algorithmic procedures (DFS, BFS, matrix representation) are applied. Moderate relation (2) to CO1– CO3 where theoretical understanding supports application.

PO3: ***Entrepreneurial Mindset and Knowledge*** **-** Weak relation (1–2) as course content is largely theoretical/technical; however, algorithmic problem-solving (CO6–CO7) may foster innovative computational thinking useful in start-up or technical project contexts.

PO4: ***Specialized skills and competencies*** - Strong (3) for CO3–CO7 since students develop domain-specific computational and programming skills in graph analysis, traversal, and matrix computation.

PO5: ***Capacity for Application, Problem-Solving, and Analytical Reasoning*** - Strong (3) relation for CO1, CO5–CO7 as they involve problem-solving in graphs (shortest paths, traversals, cycle detection). Moderate (2) for CO2–CO4 where analysis precedes implementation.

PO6: ***Communication skills and collaboration*** - Weak to moderate (1–2) as the course involves documentation and collaborative coding but not major presentation components.

PO7: ***Research-related skills*** - Moderate (2–3) linkage with CO3, CO6, CO7 as students analyze data structures, evaluate algorithms, and may extend them—skills foundational to research.

PO8: ***Learning how to learn skills*** **-** Moderate to strong (2–3), as students explore algorithmic efficiency and new graph techniques independently through programming and analysis tasks.

PO9: ***Digital and technological skills*** - Strong (3) in all coding-related COs (CO3–CO7) as students use digital tools, compilers, and algorithmic design in C.
PO10: ***Multicultural competence, inclusive spirit, and empathy*** - Weak (1) since the course focuses on technical graph concepts, though group projects may foster cooperation.
PO11: ***Value inculcation and environmental awareness*** - Weak (1) because the course does not directly address ethical or environmental issues, except indirectly via responsible coding practices.
PO12: ***Autonomy, responsibility, and accountability*** - Strong (3) for implementation-based CO5–CO7, where independent programming, debugging, and algorithm testing require responsibility.
PO13: ***Community engagement and service*** - Weak (1) as the course outcomes are technical; minimal direct connection to community service.