Syllabus Of Mathematics (Computer Science)

MTC -241MN : Mathematics for Computer Science-I

(Numerical Techniques)

Course type: Minor **No. of Credits: 02(T)**

Course Outcomes: Students will able to

1. Understand and apply numerical methods to solve algebraic and transcendental equations, including bisection, false position, and Newton-Raphson methods, with an emphasis on error analysis and convergence.
2. Develop a strong foundation in finite difference concepts, including forward, backward, central, and other difference operators, and understand their roles in numerical approximation.
3. Apply interpolation techniques for estimating intermediate values, using Newton's Gregory formulas, Lagrange's interpolation, and divided differences.
4. Utilize numerical integration methods such as the trapezoidal rule, Simpson's one-third rule, and Simpson's three-eighth rule to approximate definite integrals.
5. Solve ordinary differential equations numerically using Euler's method, Euler's modified method, and Runge-Kutta methods, understanding their accuracy and applicability.
6. Analyze the efficiency, accuracy, and limitations of various numerical methods, enabling selection of appropriate techniques for solving real-world mathematical problems.

Course Content

Unit 1: Algebraic and Transcendental Equation (06 Hours)

1.1 Errors

1.2 Bisection Mehtod

1.3 False Position Method

1.4 Newton-Raphson Method

Unit 2: Calculus of Finite Differences and Interpolation (10 Hours)

1.1 Differences

- 1.1.1 Forward Differences
- 1.1.2 Backward Differences
- 1.1.3 Central Differences
- 1.1.4 Other Differences

2.2 Relation between Operators

2.3 Newton's Gregory Formula for Forward Interpolation

2.4 Newton's Gregory Formula for Backward Interpolation

2.5 Lagrange's Interpolation Formula

2.6 Divided Difference

2.7 Newton's Divided Difference Formula

Unit 3: Numerical Integration (06 Hours)

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 Hours)

4.1 Euler's Method

4.2 Euler's Modified Method

4.3 Runge-Kutta Second Order Method

4.4 Runge-Kutta Fourth Order Method (Without Proof)

Text Book:

1. A Textbook of Computer Based Numerical and Statistical Techniques, by A. K. Jaiswal and Anju Khandelwal, New Age International Publishers.

   Chapter 1: 2.1,2.4, 2.5, 2.7

   Chapter 2: 3.1, 3.2, 3.4, 3.5,4.1, 4.2, 4.3, 5.1, 5.2, 5.4, 5.5

   Chapter 3: 6.1, 6.3, 6.4, 6.5, 6.6, 6.7

   Chapter 4: 7.1, 7.4, 7.5, 7.6

Reference Books:

1. S. S. Sastry; Introductory Methods of Numerical Analysis, 3rd edition, Prentice Hall of India, 1999.
2. H. C. Saxena; Finite differences and Numerical Analysis, S. Chand and Company.
3. K. E. Atkinson; An Introduction to Numerical Analysis, Wiley Publications.
4. Balguruswamy; Numerical Analysis.

******

MTC-242MNP: Practical on Mathematics for Computer Science-I

Course type: Minor No. of Credits: 02(P)

---

1) Plotting of the graph of function y= f(x) against x. Understanding geometric meaning of root of equation f(x) = 0 by plotting graph of function f(x).

2) i. Writing python program to solve Algebraic and Transcendental Equations:

Bisection method

ii. Plotting of the graph of function y= f(x) against x and plotting roots in the given interval graphically to understand the bisection method

3) i. Writing python program to solve Algebraic and Transcendental Equations:

using Regula Falsi method.

ii Writing python program to solve Algebraic and Transcendental Equations:

Newton-Raphson Method

4) Writing python programs to prepare difference table .

i. Newton's Forward Difference table

ii. Newton's Backward Difference table

5) Writing python program to prepare difference table .

i. Divided Difference table

6) Writing python programs for

i. Newton's Forward interpolation formula

ii. Newton's Backward interpolation formula

7) Writing python program for Newton's divided difference formula.

8) Writing python program for Lagrange's interpolation for unequal interval.

9) Writing python program for generating Lagrange's polynomial.

10) Writing python programs for Numerical Integration using

i. Trapezoidal Rule.

ii. Simpson's (1/3)rd rule.

iii. Simpson's (3/8)th rule.

11) Writing python programs for Numerical Solution of Ordinary Differential Equation

i. Euler's Method

ii. Euler's Modified Method

12) Writing python programs for Numerical Solution of Ordinary Differential Equation

i. Runge-Kutta Method 2nd order

ii. Runge-Kutta Method 4th order

MTC-291MN: Mathematics for Computer Science-II (Computational Geometry)

Course type: Minor **No. of Credits: 02(T)**

Course Outcomes: Students will able to

1. Understand the fundamental concepts and mathematical representations of two-dimensional transformations, including translation, rotation, scaling, reflection, and shearing.
2. Apply transformation matrices to perform and combine 2D geometric transformations on points, lines, and simple shapes using homogeneous coordinates.
3. Analyze three-dimensional transformations such as scaling, shearing, reflection, and ro- tation about coordinate axes and planes, and their application in object manipulation.
4. Construct and interpret different types of projections, including orthographic, axonomet- ric, and oblique projections, for visualizing 3D objects on 2D planes.
5. Develop parametric representations of common plane curves such as circles and hyperbo- las, and generate these curves through mathematical methods.
6. Demonstrate the ability to integrate multiple transformation techniques and projections to solve basic computer graphics problems involving geometric modeling.

Course Content

Unit 1: Two dimensional transformations **(10 Hours)**

1.1 Introduction

1.2 Representation of points

1.3 Transformations and matrices.

1.4 Transformation of points.

1.5 Transformation of straight lines

1.6 Midpoint Transformation

1.7 Transformation of parallel lines

1.8 Transformation of intersecting lines

1.5 Transformation: rotations, reflections, scaling, shearing

1.6 Combined transformations

1.7 Transformation of a unit square.

1.8 Solid body transformations

1.9 Translations and homogeneous coordinates

Unit 2: Three dimensional transformations (07 Hours)

2.1 Introduction

2.2 Three dimensional Scaling, shearing, rotation, reflection, translation

2.3 Multiple transformations

2.4 Rotation about an axis parallel to coordinate axes,

2.5 Reflection through coordinate planes

Unit 3: Projection (07 Hours)

3.1 Orthographic projections.

3.2 Axonometric projections.

3.3 Oblique projections.

3.4 Application of projection

Unit 4: Plane Curves (06 Hours)

4.1 Introduction

4.2 Curve representation

4.3 Parametric curves

4.4 Parametric representation of a circle and generation of circle

4.5 Parametric representation of an Parabola and generation of Parabola

Text Book:

1. D. F. Rogers, J. A. Adams, Mathematical elements for Computer graphics, Mc Graw Hill Intnl Edition.

   Chapter 1: 2-1 to 2.14

   Chapter 2: 3.1 to 3.7,

   Chapter 3: 3.12 to 3.12

   Chapter 4: 4.1, 4.2, 4.6

Reference books:

1. Schaum Series, Computer Graphics. .
2. M. E. Mortenson, Computer Graphics Handbook, Industrial Pres Inc

MTC-292MNP: Practical on Mathematics for Computer Science-II

Course type: Minor **No. of Credits: 02(P)**

---

1) Plotting 2D bar graphs, histographs, pie charts and subplots etc.
2) Plotting 3D Surface Plots, Wireframes plots and Surface Plots.
3) Using sympy generation of 2D geometrical objects like points, line segments, lines, triangles , other polygons and regular polygons. Transformations of straight lines .
4) Finding area, perimeter, midpoint of line segment or centroid of 2D objects. Point of intersections of two objects and angles between them etc.
5) Plotting of 2D geometrical objects like points, line segments, triangles and other polygons.
6) Finding rotations, reflections, scaling, shearing and translation of given 2D objects.
7) Plotting original 2D object and transformed object after applying any of the 2D transformations such as rotation, reflection, scaling, shearing and translation.
8) Using combined transformations on 2D object and finding transformed figure. Plotting both.
9) Using sympy and/or transformation matrices finding 3D rotations, reflections, scaling, shearing and translation of given 3D objects such as points or line segments etc.
10) Generation and plotting of uniformly spaced n- points on circumference of standard Circle $x^2 + y^2 = r^2$ and on arc of a circle.
11) Generation and plotting of uniformly spaced n- points on parabola when x range or y range is given