NUMERICAL METHODS
ENSH 252

Lecture : 3   **Year : II**
Tutorial : 1   **Part : II**
Practical : 3

Course Objectives:

The objective of this course is to equip students with a thorough understanding of numerical methods, focusing on their application in obtaining approximate solutions to complex mathematical problems commonly encountered in science and engineering. Emphasizing algorithm development, programming, and visualization techniques, the course enables students to apply computational approaches effectively, enhancing their problem-solving capabilities in real-world applications.

1   Solution of Non-Linear Equations   (7 hours)

    1.1   Errors and accuracy in numerical computations
    1.2   Bisection method
    1.3   Regula Falsi method and secant method
    1.4   Newton Raphson method
    1.5   Fixed point iteration method
    1.6   Comparison of the methods (Bracketing vs open-ended methods and rates of convergence)
    1.7   Solution of system of non-linear equations
        1.7.1   Direct approach
        1.7.2   Newton Raphson method

2   Solution of System of Linear Algebraic Equations   (8 hours)

    2.1   Direct methods
        2.1.1   Gauss Jordan method
        2.1.2   Gauss elimination method, pivoting strategies (Partial and complete)
        2.1.3   Matrix inverse using Gauss Jordan and Gauss elimination methods
        2.1.4   Factorization methods (Do-Little's method and Crout's method)
    2.2   Iterative methods
        2.2.1   Jacobi's method
        2.2.2   Gauss-Seidal method
    2.3   Determination of largest and smallest Eigen values and corresponding vectors using the power method

3   Interpolation   **(9 hours)**

   3.1   Polynomial Interpolation
       3.1.1   Finite differences (Forward, backward, central and divided differences)
       3.1.2   Interpolation with equally spaced intervals: Newton's forward and backward difference interpolation, Stirling's and Bessel's central difference interpolation
       3.1.3   Interpolation with unequally spaced intervals: Newton's divided difference interpolation, Lagrange interpolation
   3.2   Least square method of curve fitting
       3.2.1   Linear form and forms reducible to linear form
       3.2.2   Quadratic form and forms reducible to quadratic form
       3.2.3   Higher degree polynomials
   3.3   Cubic spline interpolation
       3.3.1   Equally spaced interval
       3.3.2   Unequally spaced interval

4   Numerical Differentiation and Integration   **(6 hours)**

   4.1   Numerical differentiation
       4.1.1   Differentiation using polynomial interpolation formulae for equally spaced intervals
       4.1.2   Local maxima and minima from equally spaced data
   4.2   Numerical integration
       4.2.1   Newton Cote's general quadrature formula
       4.2.2   Trapezoidal rule, Simpson's 1/3 and 3/8 rules, Boole's rule, Weddle's rule
       4.2.3   Romberg integration
       4.2.4   Gauss-Legendre integration (up to 3-point formula)

5   Solution of Ordinary Differential Equations (ODE)   **(8 hours)**

   5.1   Initial value problems
       5.1.1   Solution of first order equations: Taylor's series method, Euler's method, Runge-Kutta methods (Second and fourth order)
       5.1.2   Solution of system of first order ODEs via Runge-Kutta methods
       5.1.3   Solution of second order ODEs via Runge-Kutta methods
   5.2   Two-point boundary value problems
       5.2.1   Shooting method
       5.2.2   Finite difference method

6   Solution of Partial Differential Equations   (7 hours)

    6.1   Introduction and classification
    6.2   Finite difference approximations of partial derivatives
    6.3   Solution of elliptic equations
        6.3.1   Laplace equation
        6.3.2   Poisson's equation
    6.4   Solution of parabolic and hyperbolic equations
        6.4.1   One-dimensional heat equation: Bendre-Schmidt method,
                Crank-Nicolson method
        6.4.2   Solution of wave equation

Tutorial   (15 hours)

    1.   Solution of non-linear equations
    2.   Solution of system of linear algebraic equations
    3.   Polynomial interpolation
    4.   Least square method of curve fitting
    5.   Cubic spline interpolation
    6.   Numerical differentiation
    7.   Numerical Integration
    8.   Solution of ordinary differential equations (Initial value problems)
    9.   Solution of ordinary differential equations (Boundary value problems)
    10. Solution of partial differential equations

Practical   (45 hours)
Programming language to be used: Python
Results to be visualized graphically wherever possible
Practical report contents: Working principle, Pseudocode, Source code, Test Cases

    1.   Basics of programming in Python:
          Basic input/output
          Basic data types and data structures
          Control flow
          Functions and modules
          Basic numerical and scientific computation
          Graphical visualization
    2.   Solution of Non-linear equations:
          Bisection method
          Secant method
          Newton-Raphson
          System of non-linear equations using Newton-Raphson method
    3.   System of linear algebraic equations:
          Gauss Jordan Method
          Gauss elimination method with partial pivoting
          Gauss-Seidal method
          Power method

4. Interpolation
    Newton's forward difference interpolation
    Lagrange interpolation
    Least square method for linear, exponential and polynomial curve fitting
    Cubic spline interpolation
5. Numerical Integration
    Trapezoidal rule
    Simpson's 1/3 rule or Simpson's 3/8 rule
    Boole's Rule or Weddle's Rule
    Gauss-Legendre integration
6. Solution of Ordinary Differential Equations:
    Runge-Kutta fourth order method for first order ODE
    Runge-Kutta fourth order method for system of ODEs / 2nd order ODE
    Solution of two-point boundary value problem using Shooting method
    Solution of two-point boundary value problem using finite difference method
7. Solution of partial differential equations using finite difference approach:
    Laplace equation using Gauss-Seidal iteration
    Poisson's equation using Gauss-Seidal iteration
    One-dimensional heat equation using Bendre-Schmidt method
    One-dimensional heat equation using Crank-Nicholson method

Final Exam

The questions will cover all the chapters in the syllabus. The evaluation scheme will be as indicated in the table below:

Chapter	Hours	Marks distribution*
1	7	10
2	8	10
3	9	10
4	6	10
5	8	10
6	7	10
Total	**45**	**60**

* There may be minor deviation in marks distribution.

References

1. Chapra, S. C., Canale, R. P. (2010). Numerical Methods for Engineers (6th edition). McGraw-Hill.
2. Kiusalaas, J. (2013). Numerical Methods in Engineering with Python 3 (3rd edition). Cambridge University Press.
3. Grewal, B. S. (2017). Numerical Methods in Engineering & Science (11th edition). India: Khanna Publishers.
4. Yakowitz, S., Szidarovszky, F. (1986). An Introduction to Numerical Computations (2nd edition). Macmillan Publishing.
5. Kong, Q., Siauw T., Bayen A. (2020). Python Programming and Numerical Methods. Academic Press.