NUMERICAL ANALYSIS

PROF. S. BASKAR
Department of Mathematics
IIT Bombay

PREREQUISITES: Calculus, Linear Algebra at UG level. Some experience in programming in any language may be desirable.

COURSE OUTLINE :

Numerical analysis is a branch of mathematics that deals with the construction and analysis of numerical methods to various mathematical problems. For a given suitable mathematical problem, the aim is

- to develop numerical methods; and
- to develop mathematical tools to study the error involved in the numerical solution when compared to the exact solution of the mathematical problem.

In this course, we learn to meet the above two goals through some simple and familiar mathematical problems. We will also learn to implement some simple numerical methods as computer code.

ABOUT INSTRUCTOR :

Prof. S. Baskar has 16 years of teaching experience in the Department of Mathematics, IIT Bombay. He has been teaching several UG and PG courses in mathematics. Especially he has taught numerical analysis more than 15 times and also taught other mathematics courses like ordinary differential equations, partial differential equations, multivariable calculus, probability theory, and derivative pricing. Baskar received the IIT Bombay excellence in teaching award for the year 2020 and the departmental excellence in teaching award in 2018.

COURSE PLAN :

Week 1: Motivations, Preliminaries, Order of convergence.
Error Analysis: Floating-point approximations, Significant digits, Stability in computation.
Week 2: Tutorial Session for Week 1.
Introducing Python coding.
Direct Methods for Linear System of Equations: Gaussian elimination method, Partial pivoting, LU factorization, Operation counting.
Week 3: Matrix Norms: Subordinate matrix norms, Condition number of an invertible matrix.
Iterative Methods for Linear Systems: Jacobi Method
Tutorial Session for Week 2 and Python implementation.
Week 4: Iterative Methods (continued): Gauss-Seidel method, Residual corrector method, Conjugate gradient method.
Tutorial Session for Week 3 and Python implementation.
Week 5: Methods for Computing Eigenvalue and Eigenvectors: Power method, inverse power method, Gerschgorin's theorem.Tutorial Session for Week 4 and Python implementation.
Week 6: Nonlinear Equations: Bisection method, Regula-falsi method, Secant method, Newton-Raphson method, Stopping criteria. Convergence analysis ;Tutorial Session for Week 5 and Python implementation.
Week 7: Nonlinear Equations (Continued): Fixed-point iteration method, Newton's method for system of nonlinear equations ;Tutorial Session for Week 6 and 7, and Python implementation.
Week 8: Polynomial Interpolation: Lagrange form, Newton's form, divided differences, error analysis.
Week 9: Piecewise Interpolation: Piecewise polynomial interpolation, Hermite interpolation, Cubic spline interpolations.
Tutorial Session for Week 8 and 9, and Python implementation.
Week 10: Numerical Integration: Rectangle rule, Mid-point rule, Trapezoidal rule, Simpson's rule, Gaussian quadrature rule, error analysis ;Numerical Differentiation: Method of undetermined coefficients, error analysis.
Week 11: Tutorial Session for Week 10 and Python implementation.
Numerical Methods for ODEs: Initial Value Problems for First Order ODEs - Euler methods, Modified Euler methods, error analysis, Runge-Kutta method of order 2 and 4.
Week 12: Numerical Methods for ODEs: Numerical methods for two-point boundary value problems.
Numerical Methods for PDEs: Numerical methods for linear advection equation.Stability analysis ;Tutorial Session for Week 11 and 12, and Python implementation.