B. Sc. Physical Sciences with Electronics as one of the Core Disciplines

DISCIPLINE SPECIFIC ELECTIVE COURSE – DSE 7-1: NUMERICAL ANALYSIS

Course Title & Code	Credits	Credit distribution of the course			Pre-requisite of the course
		Lecture	Tutorial	Practical	
Numerical Analysis DSE 7-1	4	2	0	2	

COURSE OBJECTIVES

The main objective of this course is to introduce the students to the field of numerical analysis, enabling them to solve a wide range of physics problems. The skills developed during the course will prepare them not only for doing fundamental and applied research but also for a wide variety of careers.

LEARNING OUTCOMES

After completing this course, student will be able to,

- Analyse a physics problem, establish the mathematical model and determine the appropriate numerical techniques to solve it.
- Derive numerical methods for various mathematical tasks such as solution of non-linear algebraic and transcendental equations, system of linear equations, interpolation, least square fitting, numerical differentiation, numerical integration, eigen value problems and solution of initial value and boundary value problems.
- Analyse and evaluate the accuracy of the numerical methods learned.
- In the laboratory course, the students will learn to implement these numerical methods in Python/C++/Scilab and develop codes to solve various physics problems and analyze the results.

SYLLABUS OF DSE 7-1
THEORY COMPONENT
(Hours: 30)

Unit I **(3 Hours)**

Approximation and Errors in computing: Introduction to numerical computation, Taylor's expansion and mean value theorem. Floating Point Computation, overflow and underflow. Single and double precision arithmetic. Rounding and truncation error, absolute and relative error, error propagation.

Unit II **(8 Hours)**

Linear Systems: Solution of linear systems by Gaussian elimination method, partial and complete pivoting, LU decomposition, norms and errors, condition numbers, Gauss-Seidel method, diagonally dominant matrix and convergence of iteration methods. Solution of Tridiagonal systems; Eigenvalue Problem: Power method, inverse power method.

Unit III **(12 Hours)**

Interpolation: Lagrange and Newton's methods (divided difference) for polynomial interpolation, theoretical error of interpolation. Inverse Interpolation. Optimal points for interpolation and Chebyshev Polynomials. Minimax Theorem (Statement only). Numerical Integration: Newton Cotes quadrature methods. Derivation of Trapezoidal and Simpson (1/3 and 3/8) rules from Lagrange interpolating polynomial. Error and degree of precision of a quadrature formula. Composite formulae for Trapezoidal and Simpson methods.
Gauss Quadrature methods. Legendre, Lagaurre and Hermite quadrature methods.

Unit IV **(7 Hours)**

Initial and Boundary Value Problems: Solution of initial value problems by Euler, modified Euler and Runge Kutta (RK) methods. Local and global errors, comparison of errors in the Euler and RK methods.
Finite difference and shooting method for solving two-point linear boundary value problems.

PRACTICAL COMPONENT: NUMERICAL ANALYSIS
(Hours: 60)

The aim of this lab is not just to teach computer programming and numerical analysis but to emphasize its role in solving problems in Physics. Assessment is to be done not only on the programming but also on the basis of formulating the problem. The list of recommended programs is suggestive only. Students should be encouraged to do more physics applications. Emphasis should be given to formulate a physics problem as mathematical one and solve by computational methods. The students should be encouraged to develop and present an independent project. At least 10 programs must be

attempted (taking at least two from each topics). The implementation can be either in Python/ C++/Scilab.

Linear Systems

a) Solve a system of linear equations using Gauss Elimination method with pivoting (application to electric networks).
b) Solve a system of linear equations using Gauss-Seidel method and study the convergence (application to spring mass system).
c) Determine the inverse of a square matrix using Gauss-Jordan method.
d) Solve a tri-diagonal system of linear equations.
e) Study an example of ill-conditioned systematic
f) Find the LU equivalent of a matrix.
g) Determine the largest and smallest eigenvalues using Power and inverse power methods. Consider a case where power method fails.

Interpolation

a) Given a dataset (x, y) with equidistant x values, prepare the Newton's forward difference, backward difference and divided difference tables.
b) Given a dataset (x, y) corresponding to a physics problem, use Lagrange and Newton's forms of interpolating polynomials and compare. Determine the value of y at an intermediate value of x not included in the data set. This may be done with equally spaced and non-equally spaced x-values.
c) Given a tabulated data for an elementary function, approximate it by a polynomial and compare with the true function.
d) Compare the interpolating polynomial for a given dataset (following a known form e.g. exponential) with the approximation obtained by least square fitting.
e) Compare the interpolating polynomial approximating a given function in a given range obtained with uniformly spaced points and by Chebyshev points.
f) Compare the Chebyshev and Maclaurin series expansions of an exponential or sinusoidal function.

Integration

a) Use integral definition of error function to compute and plot erf(x) in a given range. Use Trapezoidal, Simpson and Gauss Legendre methods and compare the results for small and large values of x.
b) Use the definition of erf(x) and numerically take the limit x going to infinity to get the value of Gaussian integral using Simpson method. Compare the result with the value obtained by Gauss Hermite and Gauss Lagaurre methods.
c) Verify the degree of precision of each quadrature rule.
d) Use Simpson methods to compute a double integral over a rectangular region.

Initial Value Problems (IVP)

a) Compare the errors in Euler, RK2 and RK4 by solving a first order IVP with known solution. Reduce the step size to a point where the round off errors takes over.
b) Solve a system of n first order differential equations by Euler and RK methods. Use it to solve an nth order IVP. Solve a damped free and forced harmonic oscillator problem using this.
c) Solve a physics problem like free fall with air drag or parachte problem using

RK method.

d) Solve a compound spring system (3 springs) by solving a system of differential equations using Euler and RK for a given set of i1nitial conditions.
e) Obtain the current flowing in a series LCR circuit with constant voltage for a given set of initial conditions.

Boundary value problems (BVP)

a) Solve a linear BVP using shooting and finite difference method and compare the results.
b) Solve a non-linear BVP using the finite difference and shooting method and compare the results.
c) Determine the temperature distribution along a rod made of two dissimilar materials (of different thermal conductivities) welded together when temperatures at two ends are maintained at given temperatures.
d) Design a physics problem that can be modelled by a BVP and solve it by any method.

REFERENCES

Essential Readings for Theory Component

1) Applied numerical analysis, Cutis F. Gerald and P. O. Wheatley, Pearson Education, India, 2007
2) Advanced Engineering Mathematics, Erwin Kreyszig, Wiley India, 2008
3) Introduction to Numerical Analysis, S. S. Sastry, 5th Edition, PHI Learning Pvt. Ltd, 2012
4) Elementary Numerical Analysis, K. E. Atkinson, 3rd Edition, Wiley India Edition, 2007

Additional Readings for Theory Component

1) Numerical Recipes: The art of scientific computing, William H. Press, Saul A. Teukolsky and William Vetterling, Cambridge University Press; 3rd Edition, 2007
2) Numerical methods for scientific and engineering computation, M. K. Jain, S. R. K. Iyenge

References for Laboratory Work

1) Documentation at the Python home page (https://docs.python.org/3/) and the tutorials there (https://docs.python.org/3/tutorial/).
2) Documentation of NumPy and Matplotlib: https://numpy.org/doc/stable/user/ and https://matplotlib.org/stable/tutorials/
3) Computational Physics, Darren Walker, 1st Edition, Scientific International Pvt. Ltd, 2015
4) An Introduction to Computational Physics, T. Pang, Cambridge University Press, 2010
5) Computational Problems for Physics, R. H. Landau and M. J. Páez, CRC Press, 201