<table>
<tr><th colspan="6">Linear Algebra</th></tr>
<tr><td colspan="2">Pre-requisite: None</td><td>L</td><td>T</td><td>P</td><td>C</td></tr>
<tr><td colspan="2">Total hours: 42</td><td>3</td><td>0</td><td>0</td><td>3</td></tr>
<tr><td colspan="5">Course Content</td><td>Hrs.</td></tr>
<tr><td>Unit 1</td><td colspan="4">Introduction: Algebra in ancient India. Review of set theory and fields. Vector spaces and subspaces, linear independence and dependence, spanning set, basis and its existence, sums and direct sum of subspaces.</td><td>10</td></tr>
<tr><td>Unit 2</td><td colspan="4">Linear transformations: Linear transformations and their basic properties, Rank-Nullity theorem and its consequences, ordered basis, coordinate matrix, change of basis, matrix representation of a linear transformation, vector space of linear transformations, linear functional.</td><td>8</td></tr>
<tr><td>Unit 3</td><td colspan="4">Inner product spaces: Cauchy-Schwartz inequality, orthogonality, Gram-Schmidt orthogonalization, orthonormal basis, QR-factorization, least square approximation, best approximation, orthogonal complement, Adjoint of an operator, Hermitian, unitary and normal operators.</td><td>12</td></tr>
<tr><td>Unit 4</td><td colspan="4">Eigen-values and Eigen-vectors of a linear operator: Characteristic polynomials, minimal polynomials, Cayley Hamilton’s theorem for operators, Schur’s theorem (Statement Only), diagonalizability, Spectral theorem for normal operators, Singular Value Decomposition.</td><td>12</td></tr>
<tr><td colspan="6">Course Outcomes
Understand the concepts of vector spaces, subspaces, bases, dimension and their properties.
Relate matrices and linear transformations, compute Eigen values and Eigen vectors of linear transformations and use these notions in diagonalization.
Learn properties of inner product spaces and properties of some special operators on such spaces.
Learn to apply geometric and algebraic notions of linear algebra to determine best approximations in vector spaces.</td></tr>
<tr><td colspan="6">Text Books</td></tr>
<tr><td>1.</td><td colspan="5">Axler S., Linear Algebra Done Right, Springer UTM, 1997.</td></tr>
<tr><td>2.</td><td colspan="5">Anton H., Rorres C., Elementary Linear Algebra: Applications, 11th edition, Wiley, 2013.</td></tr>
<tr><td colspan="6">Reference Books</td></tr>
</table>

1.	Strang G., Linear Algebra and Its Applications, 4th edition, Cengage Learning, 2006.
2.	Hoffman K. and Kunze R., Linear Algebra, 2nd edition, PHI Learning, 2009.
3.	Lang S., Introduction to Linear Algebra, 2nd Edition, Springer India, 2005.
4.	Kumaresan S., Linear Algebra: A Geometric Approach, PHI Learning, 2000.
5.	VaradarajanV. S., Algebra in ancient and modern times, American Mathematical Soc., 1998

Graphs and Matrices					
Prerequisite: Mathematics II		L	T	P	C
Total hours: 42		3	0	0	3
Course Content					Hrs.
Unit 1	Graphs: Vertex degrees, isolated/pendant vertices, and the Handshaking Lemma, walks, trails, Paths, Cycles, connectivity, special types of graphs, subgraphs, isomorphism, trees and their properties, Spanning tree and minimum number of spanning trees, Eulerian and Hamiltonian graphs				12
Unit 2	Matrices: Eigenvalues of symmetric matrices, Generalized Inverse, Moore Penrose Inverse, Schur Complement, Interlacing Inequality, Weyl's Inequality, Positive Definite and Positive Semi-Definite Matrices				6
Unit 3	Representation of graphs: Incidence matrix, Matchings in bipartite graphs, Adjacency Matrix, Eigenvalues of some special graphs, Determinant, Bounds, Energy of a graph, Anti-Adjacency matrix of a directed graph, Non-singular trees				8
Unit 4	Laplacian matrix: Basic properties, Computing Laplacian Eigenvalues, Matrix-tree theorem, Bounds of Laplacian spectral radius, Signless Laplacian matrix, Regular graphs, strongly regular graphs, and the friendship theorem, Graph with maximum Energy				8
Unit 5	Algebraic connectivity: Classification of trees, Monotonicity properties of Fielder vector, Bounds of algebraic connectivity. Distance matrix of a graph, Distance and Laplacian matrix of a tree.				8
Text Books					
1.	R. Balakrishnan and K. Ranganathan, A Text Book of Graph Theory, Springer, 2000.				
2.	R. A. Horn and C. R. Johnson, Matrix Analysis, Second Edition, Cambridge University Press, 2012.				
3.	R. Bapat, Graphs and Matrices, Text and Readings in Mathematics, Second Edition, Hindustan Book Agency, New Delhi, 2014.				
Reference books					
1.	J. M. Aldous. Graphs and Applications. Springer, LPE, 2007.				
2.	J. A. Bondy and U. S. R. Murty. Graph Theory with Applications. North-Holland, 1976.				
3.	West D. B., *Introduction to Graph Theory*, Second Edition, Pearson, 2001.				
4.	R. J. Wilson, *Introduction to Graph Theory,* Fourth Edition, Prentice Hall, 1996.				
5.	D. R. Cvetkovic, S. Simic, An introduction to the theory of Graph Spectra, First Edition, Cambridge University press, 2010.				

6.	N. Biggs, Algebraic Graph Theory, Second Edition, Cambridge University press, 1993.

Computational Linear Algebra Lab					
Prerequisite: Python programming		L	T	P	C
Total hours: 24		0	0	2	1
Lab	**Lab Session Topic**				Hrs.
1	Python/NumPy/SciPy setup; matrix operations, norms, solving Ax = b; condition number experiments				2
2	Floating-point arithmetic: cancellation, overflow, underflow; ill-conditioning effects on regression accuracy				2
3	LU factorization and pivoting from scratch; solving OLS normal equations; comparison with np.linalg.solve				2
4	Cholesky factorization; ridge regression implementation and regularization path; polar decomposition via SVD				2
5	Householder QR from scratch; orthogonality verification; numerical stability vs Gram–Schmidt				2
6	Givens rotations: implementation, selective zeroing, comparison with Householder; sparse matrix demo				2
7	Least squares regression: polynomial and multi-feature fitting; optimization model view; comparison with sklearn				2
8	Weighted least squares and ridge regression; bias-variance tradeoff on noisy datasets; CPU timing experiments				2
9	Eigen decomposition of covariance matrices; PCA from scratch on MNIST or Iris				2
10	ICA with FastICA: comparison of PCA and ICA on mixed signal data				2
11	SVD; Lanczos algorithm; image compression using truncated SVD; reconstruction error vs rank				2
12	Jacobi/Gauss–Seidel solvers; convergence study; Conjugate Gradient on sparse SPD systems; GMRES demo				2

Course Outcomes:

• Demonstrate proficiency in implementing computational linear algebra algorithms using Python, NumPy, and SciPy for solving matrix-based problems.

• Analyze the numerical stability, conditioning, and computational efficiency of direct matrix factorization techniques such as LU, Cholesky, QR, and SVD.

• Apply least squares, ridge regression, PCA, and ICA techniques to solve data analysis, optimization, and dimensionality reduction problems.

• Utilize iterative methods such as Jacobi, Gauss–Seidel, Conjugate Gradient, and GMRES to solve large-scale linear systems and evaluate their convergence and accuracy.

Reference Books

1	Golub G. H., and Van Loan C. F., Matrix Computations, 3rd Edition, Hindustan Book Agency, 2007.
2	Gourlay A. R., and Watson G. A., Computational Methods for Matrix Eigenproblems, John Wiley & Sons, New York, 1973.
3	Hager W. W., Applied Numerical Linear Algebra, Prentice-Hall, Englewood Cliffs, New Jersey, 1988.
4	Lindfield G., and Penny J., Numerical Methods Using MATLAB, 4th edition, Academic Press, 2019.
5	Watkins D. S., Fundamentals of Matrix Computations, John Wiley & Sons, New York, 1991.
6	Van Loan C. F., Introduction to Scientific Computing: A Matrix-Vector Approach Using MATLAB, Prentice-Hall, Upper Saddle River, New Jersey, 1997.

Numerical Methods and Computations				
Prerequisite: Mathematics - II, Real Analysis	L	T	P	C
Total hours: 42	3	0	0	3

Course Content		Hrs.
Unit 1	Errors in Numerical Computations: Representation of numbers in computers and their accuracy, floating point arithmetic, Machine epsilon, errors in computations, types of errors, propagation of errors, conditioning and stability of numerical computations, approximations of functions: Taylor's series.	6
Unit 2	Solution of Simultaneous Linear Equations: Direct methods: Gauss and Gauss-Jordan elimination method, Pivoting strategies, LU-decomposition, Operations count. Iterative methods: Jacobi and Gauss-Seidel iteration methods. Review of vector and matrix norms, condition number, and convergence analysis.	8
Unit 3	Numerical Solution of Algebraic and Transcendental Equations: Iterative algorithms: Ancient Indian Algorithms for Square Root Computation, Bisection method, Secant method, Newton-Raphson method, and Fixed-point iteration, Convergence and error analysis, Multiple Roots. Solution of a system of nonlinear equations using Newton's method.	8
Unit 4	Numerical Interpolation: Forward, backward and central differences operators, Second difference approximations in Ancient Indian Astronomy, Newton's forward and backward difference interpolation formulae. Central difference formulae: Gauss, Stirling, and Bessel's formula. Lagrange's and Newton's divided difference interpolation formulae, Error in polynomial interpolation. Hermite and Cubic-spline interpolation.	11
Unit 5	Numerical Differentiation and Integration: Numerical differentiation through an interpolating polynomial, the Method of Undetermined Coefficients. Numerical integration: Newton-Cotes formulae, Error estimates. Romberg integration and Gaussian quadrature. Extension to multiple integrals.	9

Course Outcomes:

- Understanding of fundamental numerical methods for solving mathematical problems (linear and nonlinear equations) approximately.
- Demonstrate proficiency in applying numerical techniques to analyze and approximate tabulated functions.

• Ability to examine the accuracy and efficiency of numerical algorithms and computations. • Utilize numerical methods to evaluate numerical differentiation and integration of functions	
Text Books	
1.	Burden R. L. and Faires J. D., Numerical Analysis, 9th Edition, Cengage Learning, 2011.
2.	Jain M.K., Iyengar S.R.K., and Jain R.K., Numerical Methods for Scientific and Engineering Computation, Wiley Eastern Limited, 2012.
Reference books	
1.	Gerald C.F and Wheatly P.O., Applied Numerical Analysis, Seventh Edition, Pearson Addison-Wesley Pub. Co, 1985.
2.	Kincaid D.R. and Cheney W., Numerical Analysis: Mathematics of Scientific Computing, Third Edition, AMS, Providence, Rhode Island, 2002.
3.	Sastry S.S., Introductory Methods of Numerical Analysis, Prentice Hall of India, 2012.
4.	Sharma J.N., Numerical methods for Engineers and Scientists, 2nd edition, Narosa Publishing House, New Delhi, 2008.
5.	Divakaran P. P., The Mathematics of India: Concepts, Methods and Connections, Springer (with Hindustan Book Agency, New Delhi), 2018.

Numerical Methods and Computations Lab

Prerequisite: Mathematics-II, Real Analysis	L	T	P	C
Total hours: 24	0	0	2	1

Course Content		**Hrs.**
Exp. 1	Introduction to Programming Language/Package.	4
Exp. 2	Comparing root-finding algorithms through programming.	4
Exp. 3	Rounding-off errors in direct solvers for linear system of equations.	2
Exp. 4	Exploring convergence of iterative methods for solving linear systems.	2
Exp. 5	Analyzing interpolation through finite differences.	2
Exp. 6	Analyzing interpolation through finite differences: Equi-spaced data.	2
Exp. 7	Cubic spline interpolation.	4
Exp. 8	Numerical approximation of derivatives using finite differences.	2
Exp. 9	Numerical methods for approximating integrals.	2

Note: The course coordinator may decide upon programming language/package and may provide a list of experiments at the start of the semester based on the above list.

Course Outcomes:

- Demonstrate proficiency in implementing and testing numerical algorithms using computational software.
- Analyze numerical errors and assess the accuracy of computational solutions.
- Utilize iterative techniques to solve linear and nonlinear equations.
- Apply numerical methods to solve a variety of mathematical problems, including interpolation, numerical differentiation and integration

Reference Books

1.	Chapra S., Applied Numerical Methods with MATLAB for Engineers and Scientists, 7th edition, McGraw Hill publications, 2019.
2.	Vetterling W.T., Flannery B.P., Press W.H., Teukolsky S., Numerical Recipes in C: The Art of Scientific Computing, 3rd Edition, Cambridge University Press, 2007.
3.	Pratap R., Getting Started with MATLAB: A Quick Introduction for Scientists and Engineers, 7th edition, 2016.
4.	Lindfield G., and Penny J., Numerical Methods Using MATLAB, 4th edition, Academic Press, 2019.
5.	Bayen A., Kong Q., and Siauw T., Python Programming and Numerical Methods:

	A Guide for Engineers and Scientists,
6.	Solomon J., Numerical Algorithms: Methods for Computer Vision, Machine Learning, and Graphics, CRC Press, 2015.

Design and Analysis of Algorithms					
Prerequisite: Data Structures and Algorithms		L	T	P	C
Total hours: 42		3	0	0	3
Course Content					Hrs.
Unit 1	Algorithm Analysis: Asymptotic notation, model of computation, time and space complexities, average and worst-case analysis, Master's Theorem, solving recurrence equations- iteration method, substitution, recursion tree, master method. Amortised Analysis. Linear Search, Insertion Sort, Euclid's Algorithm for finding GCD (Lame's Theorem): Correctness, Best-Case, Average-Case and the Worst-Case Running Time Analysis. Permutation Model for Average-Case Analysis of an Algorithm for Finding the Maximum Element in an Array.				12
Unit 2	Divide and Conquer: General recurrence and methods for obtaining bounds on given recurrence. Binary Search, Merge Sort, and Maximum Subarray Sum Problem. Quick-sort: Correctness, Running Time Analysis, Order statistics - finding median and Worst-case Linear Time Algorithm for Selection Problem. Max-Min problem, Strassen's Algorithm for Matrix Multiplication, Karatsuba`s Algorithm for Large Integer Multiplication.				6
Unit 3	Dynamic Programming Approach: Introduction to dynamic programming - principal of optimality, Optimal substructure. Matrix Chain Multiplication Problem, Optimal Binary Search Tree Problem, Longest Common Subsequence Problem, 0/1 Knapsack Problem. Greedy Approach: Elements of Greedy Strategy - Greedy choice property, optimal substructure. Example Problems – Activity Selection Problem, Fractional Knapsack Problem, Huffman codes, Travelling Salesman Problem.				8
Unit 4	Graph Algorithms: Graph Traversal Algorithms (BFS, DFS), Shortest path algorithms (Bellman-Ford, Dijkstra's, Transitive-Closure, Floyd-Warshall), minimum spanning tree algorithms Kruskal, Prim), Network-flow (Ford-Fulkerson), applications of DFS:- biconnectivity, topological sort, strongly-connected components, Articulation point.				8
Unit 5	Backtracking: Introduction to Backtracking, Enumerating Independent Sets of a Graph, Graph Colouring Problem and N-Queens Problem. Complexity Classes: P, NP, NP-Hard and NP-Complete. NP-Complete. Examples with Reductions, Satisfiability, Clique, Independent Set, Vertex Cover, Graph Colouring, Dominating Set.				6
Course Outcomes:					

• Develop a comprehensive understanding of algorithm design paradigms such as divide and conquer, dynamic programming, and greedy algorithms. • Analyze the time and space complexity of algorithms using mathematical techniques such as running time analysis. • Apply algorithm design techniques to solve computational problems across various domains. • Design and implement algorithms for optimization problems, graph algorithms, and backtracking.	
Text Books	
1.	Cormen T.H., Leiserson C.E., Ronald L. Rivest and Clifford Stein, Introduction to Algorithms, Third Edition, PHI, 2009.
2.	Horowitz E., Sahni S. and Rajasekaran S., Fundamentals of Computer Algorithms, Second Edition, Universities Press, 2011.
Reference books	
1.	Goodrich M.T. and Tamassia R., Algorithm Design: Foundations, Analysis and Internet Examples, Second Edition, Wiley-India, 2006.
2.	Michael R. and David S., Computers and Intractability: A Guide the theory of NP Incompleteness, W.H. Freeman & Co., 1979.
3.	Herbert S. Wilf, Algorithms and Complexity, AK Peters Ltd., 2003.
4.	Kleinberg J. and Tardos E., Algorithm Design. Addison-Wesley Longman Publishing Co., Inc., USA, 2005.

Design and Analysis of Algorithms Lab				
Pre-requisite: Data Structures and Algorithms	L	T	P	C
Total hours: 40	0	0	4	2

Course Content

1.	Implementation of various sorting and searching algorithms (Revision), Implement quick sort with three different positions of pivot element- first, last, random
2.	Implement Tree traversal, and graph traversal (recursive algorithms), Implement deterministic and randomized selection problem
3.	Implement maximum subarray sum problem
4.	Implement Karatsuba`s Algorithm for Large Integer Multiplication
5.	Implement matrix chain multiplication, longest common sub-sequences, 0/1 knapsack
6.	A program to obtain the topological ordering of vertices in a given digraph.
7.	Implement travelling salesman problem.
8.	Print all the nodes reachable from a given starting node in a digraph using BFS method.
9.	Check whether a given graph is connected or not using DFS method.
10.	Find minimum cost spanning tree of a given undirected path using a Prim's algorithm.
11.	From a given vertex in a weighted connected graph, find shortest paths to other vertices using Dijkstra's algorithm.
12.	
13.	

Course Outcomes

- Gain practical experience in designing and analyzing algorithms through laboratory exercises and programming assignments.
- Implement and evaluate algorithmic solutions to various computational problems using programming languages.
- Analyze the time and space complexity of algorithms through empirical analysis and theoretical calculations, and compare different algorithmic approaches.
- Collaborate with peers to design and implement algorithmic solutions to programming challenges and real-world problems, and communicate findings and insights effectively.

Text Books

1.	Cormen T.H., Leiserson E., Rivest R.L. and Stein C., Introduction to Algorithms, Third Edition, PHI, 2009.
2.	Horowitz E., Sahni S. and Rajasekaran S., Fundamentals of Computer Algorithms, Second Edition, Universities Press, 2011.
Reference Books	
1.	Goodrich M.T. and Tamassia R., Algorithm Design: Foundations, Analysis and Internet Examples, Second Edition, Wiley-India, 2006.

Digital Systems and Computer Architecture					
Pre-requisite: None		L	T	P	C
Total hours: 40		3	1	0	4
Course Content					Hrs.
Unit 1	Boolean Algebra and Logic Gates: Basic definition, Axiomatic Definition, Basic theorem and Properties of Boolean Algebra, Minterms and Maxterms, Logic Operations, Digital logic gates IC, digital logic families.				6
Unit 2	Simplification of Boolean functions: Different types map method, product of sum simplification, NAND or NOR implementation, Don't care condition, Tabulation method, Adder, subtractor, Code Conversion, Universal Gates.				6
Unit 3	Sequential Logic: Flip-flops, Triggering of Flip-flops, Analysis of clocked sequential circuits, State reduction and Assignment, Flip-flop excitation, Design of counters, Design with state equations.				6
Unit 4	Basic Computer Organization and Design: Instruction codes, Computer registers, Computer instructions, Addressing modes, Timing and Control, Instruction cycle, Memory-Reference Instructions, Input-output and interrupt. Design of Basic computer, Design of Accumulator Unit.				8
Unit 5	Programming the Basic Computer: Introduction, Machine Language, Assembly Language, the Assembler. Programming Arithmetic and logic operations. Subroutines. I-O Programming.				7
Unit 6	Central Processing Unit: Introduction, GeneralRegister Organization, Stack Organization, Instruction format. Addressing Modes, Data transfer and manipulation. Program Control, Reduced Instruction Set Computer (RISC). Pipelining, Parallel Processing.				7

Course Outcomes

- Understand the fundamental principles of digital systems and computer architecture, including sequential logic, Boolean algebra, and basic of computer organization and design.
- Analyze and design digital systems using combinational and sequential logic components, and understand their role in modern computer architecture.
- Explore advanced topics in computer architecture, such as pipelining, addressing modes, and parallel processing, and their impact on system performance and efficiency.

• Design and implement computer architecture components such as processors, memory subsystems, and I/O programming.	
Text Books	
1.	Mano, Morris M., Computer system architecture. Prentice-Hall, Inc., 1993.
2.	Mano, Morris M., Digital logic and computer design, PHI, 2004
Reference Books	
1.	Tanenbaum A.S., Structured Computer Organization, 2016.
2.	Stallings, Computer Organization, Computer Organization & Architecture 7e, 2008.
3.	Hayes J., Computer Architecture and Organization, 2007.
4.	Hwang K., Advanced Computer Architecture, 2003.