Proposed Syllabus for M.Sc. (Scientific Computing) programme
Department of Scientific Computing, Modeling & Simulation
Savitribai Phule Pune University
Pune 411007
INDIA

B. W. Gore

August 21, 2023

1 Preamble

MSc in Scientific Computing formerly known as M.C.S. (Master of Computer science with specialization in scientific computing). is being offered since 1994. It is a unique programme of its kind in the country and one of the few in the world. M.Sc. (Scientific Computing) programme attempts to strike a balance between training in Sciences and Computer Science. Scientific Computing (SC) including modeling has grown into an academic discipline, in its own right, over the past few decades. It is concerned with harnessing the power of modern computers to carry out simulations relevant to science and engineering. By its very nature, SC is a fundamentally multidisciplinary subject and the various application areas give rise to mathematical models of the scientific phenomena being studied.

With the introduction of the New Education Policy (2020), it has become necessary to revise the programme structure to comply with *Circular No. 122/2023 :- Regarding implement of New Education Policy 2020 as per guidelines from Academic year 2023-24*

The various sections of this syllabus were contributed by the faculty members (current and former) of this department *viz.* Drs., Smita Bedekar, Vaishali Shah, Mihir Arjunwadkar, Sukratu Barve, Bhalchandra Gore, Bhalchandra Pujari, Ankita Katre and Snehal Shekatkar. The contributions from Dr. Abhijat Vichare and Dr. V. K. Jayaraman as well as numerous other experts is also acknowledged.

The proposed structure of the programme is summarised in the table below.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	**Level**	**Year**	**Semester**	**Type**	**Name**	**Theory**	**Practical**	**Credits**			**Core**	**Elective**	**RM**	**RP**	**FP**
2															
3	6	1	1	Core	Principles of Programming Languages-1	2	2	4	C Programming + Algorithms + Data structures		4	0	0	0	0
4	6	1	1	Core	Linear Algebra	2	0	2	Vectors + Matrices		2	0	0	0	0
5	6	1	1	Core	Mathematics for Scientific Computing	4	0	4	Real Analysis, Geometry, Calculus, Complex Analysis		4	0	0	0	0
6	6	1	1	Core	Computing Lab		4	4			4	0	0	0	0
7	6	1	1	RM	Research Methodology for Scientific Computing	4	0	4	Stats Modeling of data and Research communication		0	0	4	0	0
8	6	1	1	Elective	Software Engineering	4	0	4			0	4	0	0	0
9						**16**	**6**	**22**			**14**	**4**	**4**	**0**	**0**
10															
11															
12	6	1	2	Core	Principles of Programming Languages-2	2	2	4	C++ and Haskell		4	0	0	0	0
13	6	1	2	Core	Numerical Computing 1	2	2	4	Finite precision arithmetic, root finding, functional approximations & interpolation-extrapolation, Gaussian elimination		4	0	0	0	0
14	6	2	2	FP	Field Project On-Job Training		4	4	Hands-On In-house field project on Operating Systems		0	0	0	0	4
15	6	1	2	Elective	CBCS Elective	4	0	4			0	4	0	0	0
16	6	1	2	Core	Computing Lab		4	4			4	0	0	0	0
17	6	1	2	Core	Operating System Concepts	2		2			2	0	0	0	0
18						**10**	**12**	**22**			**14**	**4**	**0**	**0**	**4**
19															
20	6.5	2	3	Core	Numerical Computing 2	2	2	4	Eigenvalues, Givens & Householder Transformations, GS orthogonalization, Numerical differentiation & integration, numerical method for ODE-PDE solving		4	0	0	0	0
21	6.5	2	3	Core	Optimization	2	2	4	Deterministic + Stochastic Otpimization		4	0	0	0	0
22	6.5	2	3	Core	Python For Scientific Computing	0	2	2	Optimiztion codes for Nelder-Mead, Golden ratio search, Steepest descent, conjugate gradient, Fletcher-Reeves, Powerl, BFGS etc.		2	0	0	0	0
23	6.5	2	3	Core	Network Concepts	2	2	4			4	0	0	0	0
24	6.5	2	3	Elective	CBCS Elective	4	0	4			0	4	0	0	0
25	6.5	2	3	Core	Grid/HPC Computing concepts	2	2	4			4	0	0	0	0
26						**12**	**10**	**22**			**18**	**4**	**0**	**0**	**0**
27															
28	6.5	2	4	RM	MOOC Course	4		4			0	4		0	0
29	6.5	2	4	RP	Research Project		18	18	C401 Internship/Industrial Project Evaluation: Viva, Presentation, Poster and Report		0	0	0	18	0
30						**4**	**18**	**22**			0	4	0	18	0
31										**88**	**46**	**16**	**4**	**18**	**4**
32										**Total**	**Core**	**Elective**	**RM**	**RP**	**FP**
33									SPPU Specs	88	54	16	4	10	4

The following sections of this document detail out the syllabus with semester-wise breakup. The courses are coded as **SC-xxx**

2 Semester-I

Total Number of Credits	22
Theory	16
Practicals	6
Core	18
Elective	4

Course summary for Semester-1 :

Course Code	Type	Course Name	Theory Credits	Practical Credits	Total Credits
SC 501 MJ	Core	Principles of Programming Languages-1	2	2	4
SC 502 MJ	Core	Mathematics For Scientific Computing	4	0	4
SC 503 MJ	Core	Linear Algebra	2	0	2
SC 504 MJ	Core	Computational Lab-1	0	4	4
SC 505 MJ	RM	Research Methodology	4	0	4
SC 510 MJ	Elective	Software Engineering Concepts	4	0	4

2.1 SC 501 MJ : Principles of Programming Languages-1 (Core, 2T+2P=4CR)

Expected Course Outcome: Upon successful completion of this course, the student will be able to

1. Understand basics of procedural programming
2. Design an algorithm to solve given problems of various kinds
3. Analyse the given algorithm/code
4. Debug the code to spot logical errors, exceptions *etc.*
5. Write reasonably complex C code for solving various problems in scientific computing, modeling and simulation

Topics to be covered:

1. Introduction and Motivation
   (a) Introduction to Algorithms: Role of algorithms in computing, analysing and designing algorithms, explanation through examples
   (b) Introduction to some notations: Big Omicron, Big Theta, Big Omega, Small theta, Small omega. Notions of space and time efficiency and motivation for algorithm design methods, demonstration of algorithm analysis for some suitable example algorithm, say merge sort.
   (c) Study and comparison of sorting algorithms: merge sort, bubble sort, heap sort, quick sort *etc.*

2. Some useful Data Structures: Linked lists, stacks, queues, trees, binary trees, hash tables, graphs, minimum spanning trees

3. Introduction to programming: ANSI C syntax, data types, concept of void, variables, operators, expressions and statements, arrays, memory layout of multidimensional arrays, character input and output, console input and output, inclusion of standard header files

4. Control flows: If-else, for, while, do-while, switch-case, break and continue, blocks

5. pre-processor directives

6. Functions: Basics of functions, return statement, recursion, blocks, static variables

7. Program compilation and debugging techniques: introducing tools like gcc, gdb, gnu make *etc.*

8. Pointers: concept of pointers, pointer arithmetic, pointers *vs.* arrays, array of pointers, use of `malloc` and `free` functions

9. Structures, self referential structures, unions, bit fields, typedef

10. Files and I/O in ANSI C

The practical session should bring out typical scientific computing phenomena like finite precision arithmetic, truncation and round-off errors, dynamic memory allocation, series computations, recurrence relations *etc.*

Suggested texts and references:

1. Cormen, Leiserson, Rivest, and Stein: Introduction to Algorithms, MIT press; ISBN: 9788120340077

2. Aho, Hopcroft, and Ullman: Design and Analysis of Algorithms, Pearson; ISBN: 9788131702055

3. Kruse, Leung and Tondo, and Mogalla: Data Structures And Program Design In C, Pearson (2006); ISBN: 9788177584233

4. Kernighan and Ritchie: The C Programming Language, Prentice Hall, India; ISBN: 9788120305960

2.2 SC 502 MJ : Mathematics For Scientific Computing (Core, 4T=4CR)

Expected Course Outcome: A practical hands-on understanding of this fundamental area of mathematics is essential for dealing with the mathematical complexity of many kinds in Scientific computing, modeling & Simulation. This course is aimed at understanding conceptually and practically (through problem solving)

1. real-valued sets, sequences, series;

2. functions: properties and visualization;

3. convergence, limits, continuity;

4. differentiation and (multiple) integration, their interrelation, and interpretation.

5. Differential Equations

Topics to be covered:

1. Function: Definition, examples, graphs of functions
2. Limit of a function: Definition, right hand & left hand limits examples.
3. Continuity: Definition, examples and properties of continuous functions, Types of discontinuity.
4. Derivatives: Definition, geometrical interpretation, Derivatives of elementary functions by first principle. Problems on velocity and acceleration product, quotient and chain rule, implicit differentiation, derivative of inverse function.
5. Applications of derivatives: Concavity and points of inflation, Maxima and minima of a function, Related rates, Roll's theorem and Mean value Theorem, L'Hospitals rule
6. Integration: Introduction, elementary integration formulae, indefinite and definite integrals.
7. Integration Methods: Substitution, Integration by parts, Integration of product and power's of trigonometric functions, partial fractions.
8. Applications of integration: Area under the curve, length of a curve, volumes and surface areas of solids of revolution.
9. Infinite Series: Sequences and series of numbers, Limit of sequence, convergence criteria for series, Tests of convergence for series of positive numbers, Power series – region of convergence, tests of convergence, Term by term integration of power series, Fourier series.
10. Functions of two or more variables: Definition, limits and continuity, partial derivatives, Directional derivatives, gradients and tangent planes, Second derivative, maxima, minima, saddle points
11. Differential Equations: First order ODE, variables separable form, Solution of first order linear equation, Second and higher order equations, Solution of constant coefficient second order equation.

Suggested texts and references:

1. Thomas and Finney: Calculus and Analytical Geometry (Ed. 9), Pearson Education
2. James Stewart: Calculus (Ed. 5)
3. Erwin Kreyszig: Advanced Engineering Mathematics (8th Edition); John Willey and Sons
4. Simmons: Differentials Equations with applications and Historical notes.

2.3 SC 503 MJ : Linear Algebra (Core, 2T=2CR)

Expected Course Outcome: This course aims at

1. familiarizing students with abstract concept of vectors, vector spaces on fields, and bases;
2. developing an understanding of linear transformations and their matrix representation on vector spaces;

3. developing the ability to invert matrices and solving system of linear equations;
4. developing the ability to diagonalize matrices and find eigenvalues and eigenvectors.

Topics to be covered:

1. Vectors: Introduction, scalar and vector product, Equation of line and plane using vectors, Projections, Arc length of space curve, Curvature and normal vectors, applications such as displacement, velocity *etc.*
2. Vector spaces: Introduction to Vector subspaces, linear independence, span, bases, dimension and its uniqueness, direct sums, Transformation of bases.
3. Matrices: definition and interpretation, Rank of matrix, determinant, Solutions of linear equations, Eigenvalues and eigenvectors, Similar matrices and diagonalization
4. Linear operators: Definition and properties. Null space and range. Transformation of operator matrices according to basis transformations Representation of linear operators as matrices (at least rotation, translation and scaling).

Suggested texts and references:

1. Hoffman and Kunz: Linear Algebra (Ed. 2); Prentice Hall International
2. Peter D. Lax: Linear Algebra
3. Gilbert Strang: Linear Algebra and Its Applications; Cengage India Private Limited

---

2.4 SC 504 MJ : Computational Lab-1 (Core, 4P=4CR)

Expected Course Outcome: This course aims at making students familiar to working on projects. They can typically take up projects in groups of 2 or 3 students and implement a project based on some scientific computing theme with the following aim:

1. Analysis of the project
2. Design of the project (using any design model)
3. Coding
4. Testing
5. Preparing a report

Experts from industry/research organisation will guide these projects.

2.5 SC 505 MJ : Research Methodology (RM, 4CR)

Expected Course Outcome: This course aims at making students aware and familiar with the standard methods of scientific research

An outline of Topics to be covered:

1. History of research, research methodology

2. Literature search, selection of research topic (case study based), maintaining records (case study based).
3. Ethical considerations, effective verbal and non verbal communication,
4. Data collection, data safety, data integrity
5. Implementing research methodology in scientific computing, modeling and simulation
6. Statistical analysis: The module will consist of case studies of the research performed in various subjects using statistical methods, Error and noise analysis, curve fitting, regression analysis *etc*.
7. Qualitative and Quantitative Research
8. Writing research paper and/or project report, making a presentation (seminar/poster)

Suggested texts and references:

1. Ranjeet Kumar: Research Methodology-A Step by Step Guide for beginners
2. Martin Shuttleworth: History of the Scientific Methods
   https://explorable.com/history-of-the-scientific-method.
3. John Mandel: The Statistical Analysis of Experimental Data ISBN: 9780486646664

2.6 SC 510 MJ Software Engineering Concepts (Elective, 4CR)

This elective will beoffred by the department. The students desiring to take up any other elective instead of this, may have to choose from the electives offered elsewhere.
Aim of the course: To make students familiar with Software Engineering Concepts

Topics to be covered:

1. Introduction to software engineering:
   (a) A generic view of process
   (b) Process models
   (c) An agile view of process
2. Software engineering practice:
   (a) Software Engineering Practices: Planning practice, communication practice, construction practice, modeling practice
   (b) System Engineering: System engineering hierarchy, system modeling
   (c) Requirement Engineering: Requirements engineering tasks, initiating the requirement process, finding requirements, building analysis model, developing use cases
   (d) Design Engineering: Design process and design quality, design concepts, design models,
   (e) Performing user interface design: The Golden rules, User interface analysis and design, interface analysis,
   (f) Testing strategies: Test strategies for conventional software, validation testing, system testing, debugging, different testing types like white box, black box etc.

(g) Project scheduling: Basic principles, finding task set, defining task network, scheduling

(h) Risk management: Reactive Vs. proactive risk, risk identification, projection and refinement. RMMM plan

3. Case study: Initiating, analyzing, designing and testing of a project

4. Advance topics in software engineering

Suggested texts and references:

1. Roger S. Pressman: Software Engineering a Practitioner's Approach; McGraw–Hill

2. Richard Fairley: Software Engineering, Tata McGraw Hill

3. David Gustafson: Software Engineering

4. Meilier Page: Practical Guide in Structured System Design

---

3 Semester-II

Total Number of Credits	22
Theory	10
Practicals	12
Core	14
Elective	4
Field Project	4

Course summary for Semester-2 :

Course Code	Type	Course Name	Theory Credits	Practical Credits	Total Credits
SC 551 MJ	Core	Principles of Programming Languages-2	2	2	4
SC 552 MJ	Core	Numerical Computing-1	2	2	4
SC 553 MJ	Core	Operating System Concepts	2	0	2
SC 554 MJ	Core	Computational Lab-2	0	4	4
SC 581 MJ	FP	Field Project / On-Job Training	0	4	4
SC 5xx MJ	Elective	CBCS Elective-1	4	0	4

3.1 SC 551 MJ : Principles of Programming Languages-2 (Core, 2T+2P=4CR)

Course Outcomes: Upon successful completion of this course, the student will

1. Understand basics of object-oriented programming (OOP), syntax, semantics using C++ language

2. Design an algorithm to solve problems of various kinds in scientific computing and implement using C++ programming language.

3. Debug the code to spot logical errors, exceptions etc.

4. Write reasonably complex C++ code for various problems in scientific computing, modeling and simulation.

5. Understand functional programming using Haskell

The weightings recommended are: C++ 80 to 90% and Haskell 10 to 20%. The actual distribution can be left to the discretion of the instructor.

Topics to be covered:

1. Object Oriented Programming Fundamentals
   (a) What is object?
   (b) How to form an object?
   (c) What is Object Oriented programming?
   (d) Case study to identify objects

2. Object Oriented Analysis and Design: Object modeling, Inheritance, encapsulation, communication between objects

3. Working with classes: Concept of Classes, Types of Classes, Encapsulation, Conversions, type Promotion, Default Arguments And Type Casts.

4. Pointers and arrays, Dynamic memory, Expressions and statements, Various Types of Functions (Inline, Friend etc), Namespases and Exceptions

5. Class Inheritance: Operator Overloading, Inheritance, Virtual Functions, Templates, Exception Handling.

6. Introduction to functions and their evaluation, variable binding and substitution, effective computability, Turing machine concept. Haskell as a functional programming language

7. Using Haskell to solve problems in scientific computing: factorial, symbolic calculations, determinants, series computations *etc*.

Suggested texts and references:

1. Stroustroup: The C++ Programming Language; Pearson Education

2. Lippman: C++ Primer; Addison-Wesley

3. S. Prata: C++ Primer Plus, Galgotia Publications

4. Hutton: Programming in Haskell; Cambridge University Press

5. https://www.haskell.org/

3.2 SC 552 MJ : Numerical Computing-1 (Core, 2T+2P=4CR)

Expected Course Outcome: By numerical computing, we mean numerical analysis and numerical mathematics with a strong hands-on computing component. This course is intended to cover topics of practical importance. Specific objectives of this course include the following:

1. developing an understanding of finite precision arithmetic;

2. To develop ability to understand and implement numerical algorithms;

3. To be able to make an informed choice of an appropriate numerical method to solve a given problem;

4. To be able to estimate rounding error, run time, memory and other computational requirements.

Topics to be covered:

1. Roots, zeros, and nonlinear equations in one variable. Are there any roots anywhere? Examples of root-finding methods. Fixed point iteration, bracketing methods such as bisection, regula falsi. Slope methods: Newton-Raphson, Secant. Accelerated Convergence Methods: Aitken's process, Steffensen's and Muller's method.

2. Interpolation. Concept of interpolation. Polynomial approximation. The interpolation problem and the vandermonde determinant. The Lagrange form of the interpolation polynomial. The error in polynomial interpolation. Newton's form of the interpolation polynomial. Divided differences, Newton-Gregory forward and backward differences. Piece-wise interpolation: spline interpolation and cubic splines.

3. Approximations. The Minimax approximation problem. Construction of the minimax polynomial. Least-squares and weighted least squares approximations. Solving the least-squares problem: direct and orthogonal polynomial methods.

4. Solving linear systems of equations. Gaussian elimination. Pivoting. Ill-conditioning. Gauss-Jordan method. Matrix inversion. Triangular factorization (LU). Permutation matrices. Cholesky factorization. Iterative methods for linear systems. Diagonally dominant matrices. Jacobi iteration. Gauss-Seidel iteration.

Suggested texts and references:

1. David Goldberg: What Every Computer Scientist Should Know About Floating-Point Numbers. Computing Surveys, March 1991.

   http://docs.sun.com/source/806-3568/ncg_goldberg.html

2. Doron Levy: Introduction to Numerical Analysis. Unpublished, 2010.

   http://www.math.umd.edu/~dlevy/books/na.pdf

3. M. T. Heath: Scientific Computing: An Introductory Survey. McGraw-Hill, 2002.

   http://heath.cs.illinois.edu/scicomp/

4. Steven C. Chapra and Raymond P. Canale: Numerical Methods for Engineers. Tata McGraw-Hill, third edition, 2000.

5. H. M. Antia: Numerical Methods for Scientists and Engineers. Hindusthan Book Agency, second edition, 2002.

6. Kendall E. Atkinson: An Introduction To Numerical Analysis. Wiley India, second edition, 2008.

3.3 SC 553 MJ : Operating System Concepts (Core, 2T=2CR)

Expected Course Outcome: To understand the nature of modern operating systems and the intricacies thereof. The Unix operating system can be used as a case study. The knowledge of this will be a pre-cursor for the course 3.5 (Field Project/On-Job Training).

1. To understand need of operating system
2. Overview of complications in handling multi-user, multi-tasking, multi-processor systems
3. To understand process model
4. File system designs.

Topics to be covered:

1. Introduction to UNIX.
2. Implementation of buffer cache: Structure and Philosophy of the cache implementation. Algorithms used by the buffer cache. Also, touch upon inode cache
3. File system: The file system switchers table, VFS architecture, File systems implementation on disk, File system handling kernel algorithms, Issues for file system handling, System calls for file system manipulations.
4. Process: State transitions, Process structure and layout, Multiprocessing details-Context, context switches, memory management concepts, System calls.
5. Process Scheduler: Class specific implementation, Priorities, system calls.
6. Memory Management Techniques: Swapping, Demand paging, Hybrid, Virtual Memory.
7. Time and Clock
8. I/O Subsystems

Suggested texts and references:

1. Bach: Design of the Unix System; Prentice Hall
2. Peterson, Silberschatz, Galvin: Operating System Concepts; Addision Wesley
3. Andrew Tanenbaum: Modern Operating System; Pearson Education.
4. William Stallings: Operating System Internals and Design Principles

---

3.4 SC 554 MJ : Computational Lab-2 (Core, 4P=4CR)

This course may be thought of as an extension of Computational Lab-1 project work. The student groups will continue working on their projects from the previous semester. Alternatively, if their first semester projects would have been completed by the end of the first semester, the instructor may assign fresh projects to the group of students. Experts from industry/research organisation will guide these projects.

3.5 SC 581 MJ : Field Project/On-Job Training (FP/OJT, 4P=4CR)

Expected Course Outcome: This field project is supposed to give students the opportunity to implement their knowledge gathered so far from the various courses to complete a computing project. The project may be from any filed from the scientific computing syllabus, *e.g.* Operating Systems (3.3) where a student may choose to implement the working of processor control thread or scheduling algorithm or memory swap operation.
The course is expected to let students

1. gather necessary information, data, algorithms to implement the project
2. develop the design and implementation
3. plan the testing phase
4. demonstrate the working computing model of the problem chosen.

3.6 SC 5xx MJ : CBCS Elective (Elective, 4CR)

Each student will choose one elective from the list of elective courses given at the end of this document. The 'xx' in the course code is between the range 60 to 79. This may be the first part of the elective courses that run in pairs, such as CFD-1, DSIP-1 *etc.*

---

4 Semester-III

Total Number of Credits	22
Theory	12
Practicals	10
Core	18
Elective	4

Course summary for Semester-3 :

Course Code	Type	Course Name	Theory Credits	Practical Credits	Total Credits
SC 601 MJ	Core	Network Concepts	2	2	4
SC 602 MJ	Core	Numerical Computing-2	2	2	4
SC 603 MJ	Core	Optimization Concepts	2	2	4
SC 604 MJ	Core	Grid/HPC Computing concepts	2	2	4
SC 605 MJ	Core	Python For Scientific Computing	0	2	2
SC 6yy MJ	Elective	CBCS Elective-2	4	0	4

4.1 SC 601 MJ : Network Concepts (Core, 2T+2P=4CR)

Expected Course Outcome: This course aims at making the students understand the nitty-gritty of computer networks. The topics mentioned here are not elaborated upon in order to address the very volatile and fast changing nature of the networking technology. The course teacher will be the best judge of what details to cover under each topic. The department committee (DC) will approve the plan of the teacher.

Topics to be covered:

1. Computer Networking Fundamentals; using twisted pair of wires, ethernet cable, coaxial cable, optical fibre, wireless media for communication
2. Networking Standards and the OSI Model
3. TCP, UDP, IP protocol; IPv4 and IPv6; DNS, Telnet and FTP.
4. Network Mail Protocols: SMTP, POP3, IMAP
5. Network Topologies
6. Ethernet
7. Wide Area Networks
8. Wireless Networking
9. Virtual Networks including VPN
10. Network Security

Suggested texts and references:

1. William Stallings: Data and Computer Communication; Pearson Education India
2. Richard Stevens: TCP / IP Illustrated Vol . I / II / III; Addison-Wesley
3. Rich Seifert: The Switch Book: The Complete Guide to LAN Switching Technology; Wiley
4. Andrew S. Tanenbaum: Computer Networks; Pearson Education India
5. Prakash C. Gupta: Data Communication and Computer Networks; PHI Learning

---

4.2 SC 602 MJ : Numerical Computing-2 (Core, 2T+2P=4CR)

Course Rationale: This course being a sequel of 3.2, the expected outcome remains the same. The topics covered here are in continuation with the course 3.2

Topics to be covered:

1. Numerical differentiation: Basic concepts. Differentiation via interpolation. The method of undetermined coefficients. Numerical derivatives using forward difference, backward difference and central difference. Richardson extrapolation. Differentiation using Lagrange Polynomial, Newton Polynomial.
2. Numerical integration. Basic concepts. Integration via interpolation. Composite integration rules. Additional integration techniques. The method of undetermined coefficients. Change of an interval. General integration formulas. Simpson integration. The quadrature error. Composite Simpson rule. Gaussian quadrature. (Optional) Romberg integration. Adaptive quadrature basics.
3. Numerical solutions of ODEs. Euler method, accuracy and stability, stepsize control. Runge Kutta methods, Heun's method. Discussion about stiffness and adaptive-stepsize solvers. Introduction and overview of predictor-corrector methods.

4. Eigenvalues and eigenvectors. Homogeneous systems, Power Method, Jacobi's method, Given's, Householder's transformation and Lanczos transformation to tridiagonal form, LR, QL/QR transformation for eigenvalues of tridiagonal matrices, determinants of tridiagonal matrices, symmetric matrices, band matrices

5. (Optional) Numerical linear algebra software: working introduction to BLAS/LAPACK.

Suggested texts and references:

1. H. M. Antia: Numerical Methods for Scientists and Engineers. Hindusthan Book Agency, second edition, 2002.

2. Doron Levy, Introduction to Numerical Analysis. Unpublished, 2010.

   http://www.math.umd.edu/~dlevy/books/na.pdf

3. M. T. Heath: Scientific Computing: An Introductory Survey. McGraw-Hill, 2002.

   http://heath.cs.illinois.edu/scicomp/36

4. Curtis F. Gerald and Patrick O. Wheatley: Applied Numerical Analysis. Addison-Wesley, fifth edition 1998.

5. Kendall E. Atkinson: An Introduction To Numerical Analysis. Wiley India, second edition, 2008.

4.3 SC 603 MJ : Optimization Concepts (Core, 2T+2P=4CR)

Expected Course Outcome: This course is intended to build a foundation for deterministic and stochastic optimization methods. The course is based on the use of key concepts in linear algebra and calculus to develop understanding of optimization methods. Stochastic optimization methods are useful in situations involving multiple local or global minima, when there is combinatorial complexity in the optimization problem, when the goal is to locate a global minimum. A background in these methods should be an essential part of scientific computing repertoire. By completing this course, the student will be able to

1. 1. understand the need for optimization 2. choose and implement appropriate optimization method to solve problem at hand 3. differentiate between deterministic and stochastic; unconstrained and constrained optimization problems

Topics to be covered:

1. Preliminaries. The need for optimization: A survey of problems and their modeling contexts (e.g., problems in domains like engineering/economic/agricultural industry domains, time and cost minimization, efficiency enhancement etc.); the objective function, domain of the objective function, applicable optimization methods. a minimizer, local and global minima, gradient based distinction of maxima, minima and saddle points. constrained and unconstrained optimization, convexity. Visualization: contours, surfaces, surfaces, normals, orthogonality; The optimization and visualization software tools available.

2. Optimization in one dimension. Numerical methods without derivatives: two-point bracketing and bisection, golden section search, parabolic interpolation and Brent's method.

   Numerical methods with derivatives: Newton's method, Davidon's method.

3. Unconstrained minimization in more than one dimension: Generalization of Newton method for multiple dimensional problems. Concepts of Jacobian and Hessian. Limitations of Newton method.

   Steepest descent method.

   conjugate direction methods.

   quasi-Newton methods: Approximating inverse Hessian, rank one correction and algorithm; rank two correction and DFP, BFGS algorithms.

4. Constrained Minimization. Equality and inequality constraints: general theory. Lagrange multipliers, Karush-Kuhn-Tucker (KKT) method.

5. Simplex method. Nelder-Mead Algorithm.

6. A brief introduction to random number generators and Monte Carlo methods.

7. Introduction to stochastic optimization. Formal problem statement. Stochastic vs. deterministic optimization. Principles of stochastic optimization. Local vs. global minimization. An overview of problems involving multiple minima, local or global.

8. Random search methods. General properties of direct random search. A few specific algorithms for random search.

9. Simulated annealing. The analogy between optimization and free-energy minimization by a physical system. The travelling salesman problem and SA.

10. Genetic algorithms. Introduction. Chromosome coding and the basic GA operations. The core genetic algorithm. Implementation aspects. Some perspective on the theory for GAs.

Suggested texts and references:

1. E.K.P. Chong and S.H. Zak: An Introduction To Optimization; Wiley, India (2016).

2. A. Ravindran, K.M. Ragsdel, and G.V. Reklaitis: Engineering Optimization: Methods and Applications; Wiley, India (2006).

3. M. T. Heath: Scientific Computing: An Introductory Survey; McGraw-Hill (2002). http://heath.cs.illinois.edu/scicomp/

4. John H. Mathews: Numerical Methods for Mathematics, Science and Engineering; Wiley.

5. James C. Spall: Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control; Wiley (2003).

6. Michell: An Introduction to Genetic Algorithms; MIT Press (1996).

7. P. J. M. Van Laarhoven and E. H. L. Aarts: Simulated Annealing: Theory and Applications; Kluwer Academic Publishers (1987).

4.4 SC 604 MJ : Grid/HPC Computing concepts (Core, 2T+2P=4CR)

Topics to be covered:

1. HPC programming platforms: Implicit Parallelism: trends in microprocessor architectures, memory system performance limitations, control structure of HPC platforms, communication model, physical organisation, architecture of parallel computer, interconnection networks, network topologies, static, dynamic interconnection networks, cache coherence in multiprocessor systems, communication costs, message passing costs

2. Programming using message-passing: Principles of message passing programming, send and receive operations, blocking and non-blocking message passing, Message passing interface, introduction to MPI routines, data types, concept of communicators, communication domain, communicator groups, creating topologies using MPI, overlapping communication with computation, MPI syntax for frequently used communication calls related to send, receive, barrier, broadcast, reduction, prefix, gather, scatter, all-to-all communication. programs with MPI for addition/multiplication of list of random numbers, matrix-matrix multiplication, bubble sort, shell sort, quick sort, bucket sort, sample sort etc.

3. Parallel algorithm design: Decomposition, tasks, dependency graphs, granularity, concurrency, task interaction, processes and mapping, decomposition techniques: recursive, data, speculative, hybrid, characteristics of tasks and intertask interactions, mapping techniques for load balancing, static mapping, dynamic mapping, methods for containing interaction overheads, parallel algorithm models: data parallel, task-graph, work pool, master-slave, producer-consumer or pipeline model

4. Basic communication operations: Personalized Communication, Collective Communication, Collective communication operation algorithms on ring, mesh, hypercube topologies and their cost analysis, improving speed of communication operations

5. Parallel algorithms for linear algebra: Matrix-vector multiplication with 1-D, 2-D partitioning, matrix-matrix multiplication: simple algorithm (1-D partitioning), Cannon's algorithm (2-D partitioning), DNS algorithm (3-D partitioning), solving system of linear equations with simple Gaussian elimination algorithm (1-D partitioning), 1-D partitioning with pipelined communication and computation, 2-D partitioning with pipelined communication and computation, Gaussian elimination with partial pivoting, solving a triangular system with back substitution, parallel algorithm for Jacobi's iterative method and Gauss-Seidel iterative method for solving system of linear equations.

6. Analytical modeling of parallel programs: Overheads in parallel programs, performance metrics such as execution time, total parallel overhead, speedup, efficiency, cost, effect of granularity on performance, scalability, scaling characteristics, isoefficiency metric of scalability, cost-optimality, isoefficiency function, degree of concurrency and isoefficiency function, minimum execution time, minimum cost-optimal execution time, asymptotic analysis of parallel programs, other scalability metrics Cost analysis of parallel programs developed in the course work.

7. (Optional) Introduction to GPUs: CUDA programming concepts

Suggested texts and references:

1. Ananth Grama, Anshul Gupta, George Karypis and Vipin Kumar: Introduction to Parallel Computing; Pearson Education (2004).

2. V. Rajaraman and C. Siva Ram Murthy: Parallel Computers: Architecture and Programming; Prentice-Hall India (2000).

3. Ian Foster: Designing and Building Parallel Algorithms; Addison-Wesley (1995)

4. V. Rajaraman: Elements of Parallel Computing; Prentice Hall (1990).

5. Barry Wilkinson and Michael Allen: Parallel Programming: Techniques and Applications Using Networked Workstations and Parallel Computers; Pearson India.

6. Victor Eijkhout, Edmond Chow, Robert van de Geijn: Introduction to High Performance Scientific Computing; MIT Press (2016).

4.5 SC 605 MJ : Python For Scientific Computing (Core, 2P=2CR)

Expected Course Outcome: Python is a new programming medium for scientific computing It has strong object-oriented framework and in-built libraries. At the end of the course students are expected to develop Python programs to implement advanced level scientific computing projects. Ideally the aim of the course could be to have one Python project at the the end of the term, and necessary syntax and other tools are taught as the requirement along the development.

Topics to be covered:

1. Data and control structures Data types and objects. Introduction to conditionals (`if`) and loops (`for` and `while`). Loading packages.

2. Functions Defining and using functions. Using built-in functions. Recursion.

3. Numpy/Scipy Introduction and usage of numpy and scipy modules. Generation of arrays, indexing, slicing, binary operations, linear algebra etc.

4. Dataframe Introduction of pandas. Generation of dataframes, I/O of dataframes, selections, indexing and other operations. Examples using real world problems.

5. Plots Advance 2D and 3D plotting

6. Classes User defined classes, objects, methods, and functions.

Suggested texts and references:

1. David Beazley, Brian K. Jones: Python Cookbook; O'Reilly Media

2. Mark Lutz: Programming Python; O'Reilly Media

3. The Official Python Documentation (https://docs.python.org/)

High Performance Computing (HPC) hardware architecture, Parallelization strategies, task parallelism, data parallelism, and work sharing techniques, Shared memory parallelism, Parallel programming with OpenMP and (POSIX) threads, Message passing with MPI, Execution profiling, timing techniques, and benchmarking for modern single-core and multi-core processors, High performance algorithms, Parallel scientific computing, Parallel data science algorithms, HPC libraries, Introduction to GPUs

4.6 SC 6yy MJ : CBCS Elective (Elective, 4CR)

Each student will choose one elective from the list of elective courses given at the end of this document. The 'yy' in the course code is between the range 10 to 29. This may be the seocnd part of the elective courses that run in pairs, such as CFD-2, DSIP-2 *etc.*

---

5 Semester-IV

Total Number of Credits	22
Theory	4
Practicals	18
Core	18
Elective	4

Course summary for Semester-3 :

Course Code	Type	Course Name	Theory Credits	Practical Credits	Total Credits
SC 651 MJ	Core	Industrial Project	0	18	18
SC 6zz MJ	Elective	MOOC course	4	0	4

5.1 SC 651 MJ : R&D/Industrial Project (Core, 18CR)

Course Rationale: R&D/Industrial Project is the pinnacle of this programme. The purpose of the course is for the students to get in-depth exposure and experience in addressing challenging, real-life problems through scientific computing methods applied in industry and/or R&D projects.

5.2 SC 6zz MJ : (Elective, 4CR)

Course Rationale: This is a course which the students can use to learn the desired skills and/or additional knowledge required for their project 5.1. Alternatively, this course may be used to acquire any additional knowledge. The approval of the DC will be required to undergo the online course chosen. The 'zz' in the course code is between the range 60 to 79.

---

6 Choice-Based Credit System: Elective courses

The elective courses offered are divided in two catagories: streams and stand-alone. The streams consist of a pair courses that can be offred at basic and advanced level or general and particular specialization. The following subsections list these courses and the summary table is presented in each of them to have a view of available choices at a glance.

6.1 Choice-Based Credit System: Elective Streams

Code	Name	Cr
SC 560/610/660 MJ	Computational Fluid Dynamics-1	4
SC 561/611/661 MJ	Computational Fluid Dynamics-2	4
SC 562/612/662 MJ	Machine Learning-1	4
SC 563/613/663 MJ	Machine Learning-2	4
SC 564/614/664 MJ	Digital Signal and Image Processing-1	4
SC 565/615/665 MJ	Digital Signal and Image Processing-2	4

6.1.1 SC 560/610/660 MJ Computational Fluid Dynamics-1

Course Goal: Develop intermediate-level understanding and hands-on skills in the domain of computational fluid dynamics.

Topics to be covered:

1. Elementary concepts: Background space, coordinate systems. Fields, scalars, vectors, tensors, transformations, distance metric. Concepts of vector calculus (flux, line integral, Gauss and Stokes theorems). Index notation and Einstein convention. Total derivative, integral curves, velocity field and co-moving derivative.
2. Balance equations: Equation of continuity. Jacobians and their rates of change. Lagrangian coordinates. Reynolds theorem. Surface forces and traction vector. Cauchy theorem and concept of stress tensor. Cauchy equation of momentum balance. Angular momentum balance equation. Heat flux density, internal energy density, energy balance equation.
3. Constitutive relations: Introduction. Thermodynamic stimulus and response, rate of response. Darcy's, Fourier's, Ohm's and Fick's laws, Hooke's law, Newton's law of viscosity. Shear, rotation and dilation of velocity field, Navier-Stokes equation, boundary conditions and their importance.
4. Examples of flow: Hagen-Poisseuille flow, Couette flow and other special cases.

Suggested texts and references:

1. T. J. Chung: Computational Fluid Dynamics; Cambridge University Press (2002)

6.1.2 SC 561/611/661 MJ Computational Fluid Dynamics-2

Course Goal: Develop intermediate-level understanding and hands-on skills in the domain of computational fluid dynamics.

Topics to be covered:

1. FEM techniques: Finite elements. Shape functions. Finite element interpolation functions. Weighted residual approach. Assembly of element equation. Finite element formulation for advection equation.
2. Finite volume approach: Finite volume method. Finite volume discretization. Face area and cell volume. Finite volume via finite difference. Finite volume via finite element method. Comparison of finite difference, finite element, and finite volume methods.
3. Grid generation: Structured grid generation. Unstructured grid generation. Mesh adaptation. Automatic grid generation for complex geometry problems. Computing techniques.
4. Application to multiphase flows.
5. Higher-order methods for CFD.
6. Optimization through CFD: Optimization problem associated with evaluation of first derivative. Optimization problem associated with evaluation of second derivatives.
7. Advanced fluid dynamics: Intermediate structures like vortices, boundary layers, shocks, waves and caustics, stream filaments.
8. Numerical methods: Grid generation techniques for structures and unstructured grids.
9. Hands-on problem-solving through CFD: Implementation of codes for CFD. Computational environments for CFD such as `OpenFOAM, CFDExpert`. `OpenFOAM` architecture, solvers cases and utilities; writing cases and solvers. `CFDEXpert` problems.

Suggested texts and references:

1. T. J. Chung: Computational Fluid Dynamics; Cambridge University Press (2002)

6.1.3 SC 562/612/662 MJ : ML1

Course Rationale: In the current era of data explosion, scientists and engineers are looking for methods to extract relevant information through automation. Machine learning methods enable development of algorithms to "learn" from the data and make relevant inferences and predictions. This course is designed to introduce students with the subject of machine learning via a variety of statistical tools like classification, clustering, etc. At the end of the course, students are expected to understand the basics of classification and clustering, develop necessary codes, and make correct inferences from given data.

Topics to be covered:

1. Introduction and background: What is machine learning? Overview and survey of applications. Problem of induction and statistical inference: Input-output functions, Boolean functions, parametric and non-parametric inference. Probability, certainty, and Bayes theorem. Introduction to typical learning tasks: regression, pattern recognition, feature selection, classification, clustering, rule induction (association). Model validation techniques: cross validation, leave-one-out, majority, etc. Measures of performance of a classifier: confusion matrix, sensitivity, specificity, ROC curves and the AUC, etc.
2. Computational environments for machine learning: Brief introduction to ML frameworks such as `Weka`, packages or modules in `R`, `python`, etc., I/O formats, basic introduction to interfaces.
3. Supervised learning: Additive models, generative models and discriminative models, logistic regression, Naïve Bayes classifier, linear discriminant analysis, neural networks and support vector machines.

4. Unsupervised learning fundamentals:

   Clustering: k-means, hierarchical, self-organizing map. Feature selection via principal component analysis.

5. Laboratory: Models using `Weka` or `R` on UCI benchmark data sets. Writing interfaces for a classifier as derived from a learner. Regression Models. k-means clustering, writing interface for a clusterer.

Suggested texts and references:

1. T. Hastie, R. Tibshirani, J. H. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2013.
2. Tom Mitchell, *Machine Learning*. McGraw-Hill, 1997.
3. Peter Flach, *Machine Learning: The Art and Science of Algorithms that Make Sense of Data*. Cambridge University Press, 2012.
4. Carl Edward Rasmussen and Christopher K. I. Williams, *Gaussian Processes for Machine Learning*. MIT Press, 2005.
5. Daphne Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.
6. Christopher Bishop, *Pattern Recognition and Machine Learning*. Springer, 2007.
7. Kevin P. Murphy, *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.
8. Larry Wasserman, *All of Statistics: A Concise Course in Statistical Inference*. Springer, 2004.
9. David MacKay, *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, 2003.
10. Y. S. Abu-Mostafa, Malik Magdon-Ismail, and Hsuan-Tien Lin, *Learning From Data*. AMLBook, 2012.
11. Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar, *Foundations of Machine Learning*. MIT Press, 2012.

6.1.4 SC 563/613/663 MJ : ML2

Course Rationale: Students are expected to be familiar with basic tools of machine learning by now. This course should be pitched at more advanced level than the prequel 6.1.3. At the end of this course, students are expected to be able to develop new hybrid algorithms for data analysis and predictions. Students may be introduced to standalone software packages that employ machine learning tools.

Topics to be covered:

1. Formulation of the learning problem: Learning as a statistical problem: estimation of probability measure and basic problems of statistics, learning as density estimation, risk, empirical risk and structural risk, introduction to ill-posed problems and regularization. Learning as an algebraic problem. Learning as a computational problem: learnability, PAC learning, bounds on data, algorithmic learning theory basics. Laboratory: linear models in `R`, writing basic interface for a learner.
2. Reinforcement learning.

3. Ensemble methods, boosting and bagging.
4. Intelligent agents.
5. Advanced clustering methods: Gaussian/Normal mixture models and the EM algorithm. Fuzzy clustering.
6. Feature selection: Feature selection using singular value decomposition (SVD). Various filter and wrapper methods.
7. Advanced applications, machine learning algorithms, and case studies: The machine learning approach to time series analysis. Fundamentals and application of clustering/classification to image analysis. Text analytics and natural language processing fundamentals with applications. Social media analytics. Financial analytics.
8. Laboratory: Case studies using real-life public-domain data, based on the topics covered.

Suggested texts and references:
See the relevant section for the sister course 6.1.3.

6.1.5 SC 564/614/664 MJ : DP1

Course Rationale: This course introduces the student to signal modeling and analysis formalisms, tools, and techniques, and their applications to solve real-life problems in various fields like electronic communication, signal detection, satellite imaging, medical diagnostic signals and images, video analysis, etc., with the view of making the student aware of (a) basic theory and mathematical, statistical and algorithmic tools (b) perspective on modeling by way of real-life contexts, examples and applications, and (c) relevant numerical methods.

Specific objectives of this course include:

1. Understanding the need for signal processing;
2. understanding the correspondence between actual devices, their operations, and their mathematical-statistical representations; learning mathematical, statistical, and algorithmic tools used to improve the quality of a signal and extract useful information from signal;
3. being able to design simple systems for signal processing (like digital filters, digital spectrum analyzer, etc.) using tools such as transforms (Fourier, Z, wavelet), difference equations, pole-zero and frequency plots, mean, median, variance, histogram, probability distributions, etc.;
4. being able to analyze given signal using appropriate tools and infer about quality, content, etc.

Topics to be covered:

1. Discrete time signals: sequences; representation of signals on orthogonal basis; sampling and quantization; reconstruction of signals; Nyquist's theorem; analog $\rightleftharpoons$ digital signal conversions.
2. Discrete systems: attributes, classifications, analysis of LTI systems, representation of discrete time systems using difference equations, implementation of discrete time systems; correlation of discrete time signals.
3. Z-transform, frequency analysis, discrete Fourier transform (DFT), fast Fourier transform algorithm, frequency response of a system, spectra at output of LTI system; convolution-deconvolution concepts.

4. LTI system as frequency domain filters; filter characterization, inverse systems.
5. Design of FIR and IIR filters; Gaussian, Butterworth, Chebyshev approximations; low-pass, high-pass, notch, bandpass and band-reject filters; effect of quantization on filters–round-off effects.
6. Adaptive filters; power spectrum estimation.
7. Applications of DSP to speech/music and radar/radio-telescope signal processing.

Suggested texts and references:

1. J. G. Proakis and D. G. Manolakis: Digital Signal Processing Principles, Algorithms, and Applications; Pearson (2012).
2. Oppenheim and Schafer: Digital Signal Processing; PHI Learning (2008).

6.1.6 SC 565/615/665 MJ : DP2

Course Rationale: See purpose/outlook/rationale/goals for the prequel 6.1.5. The focus of this follow-up course is more on digital image processing; e.g., electronic communication, satellite imaging, medical diagnostic images, video analysis, image segmentation, etc. Specific objectives of this course are:

1. Understanding the need for image processing;
2. understanding the correspondence between actual devices (camera, X-ray, tomographic devices, etc.), their operations, and their mathematical-statistical representations;
3. learning mathematical, statistical, and algorithmic tools used to improve quality of image and extract useful information from image; being able to design simple systems for image processing (like image smoothing, removal of noise, color spectrum analysis etc.) using tools like mean, median, variance, histogram, probability distribution, spatial and temporal filters, etc.;
4. being able to analyze given image using appropriate tools and infer about quality, content, etc.; learning to segment the given image to isolate different objects from it.

Topics to be covered:

1. What is digital image processing? Its origin, overview of fields of applications, fundamental steps in digital image processing,components of an image processing system.
2. Digital image fundamentals: Human eye and image formation; EM spectrum, image sensing and acquisition, sampling and quantization, basic relationships among pixels-neighbours, connectivity, regions, boundaries, distance measures.
3. Tools used for digital image processing: matrices and vectors, linear vs. non-linear operations, set and logical operators, image transforms, probabilistic methods.
4. Intensity transformations. Basic functions: negation, log, gamma transformation, histogram processing.
5. Spatial filtering: fundamentals, smoothing and sharpening spatial filters, unsharp masking and high boost filtering, use of gradients for non-linear image sharpening, Laplacian operator, using fuzzy techniques for spatial filtering and for intensity transformations

6. Filtering in frequency domain: Fourier series and transform, DFT, FFT, generalization for 2D image space, basic filtering in frequency domain, correspondence between spatial and frequency domain filtering, image smoothing and sharpening using frequency filters, low-pass, high-pass and band-pass filters, notch and band-reject filters, Gaussian, Butterworth, Laplacian filters.
7. Image restoration and reconstruction: Image degradation model, modeling noise, noise reduction using spatial filtering, periodic noise reduction using frequency domain filtering, estimating image degradation function using observation, experimentation and modeling; inverse filtering, statistical (Wiener's mean square error filtering, constrained least square filtering, mean filtering (arithmetic, geometric and harmonic); image restoration from projections (fundamentals of tomography).
8. Colour image processing; colour models: RGB, CMY, CMYK, HSI, relations between them; colour transformations, colour smoothing and sharpening.
9. Wavelets: Haar, Daubechies; DWT, 2D generalization, significance of wavelet coefficients, compression using wavelets.
10. Image compression techniques: spatial and temporal redundancy in images, fidelity criteria, image compression models, some image compression methods (using Huffman, Golomb, arithmetic, LZW, run-length, bit-plane and block-transform coding), relation between block-transform coding and wavelets.
11. Image segmentation fundamentals; point, line and edge detection, thresholding, region-based segmentation.
12. Overview of topics like Video analysis, morphology, watermarking, object/pattern recognition, compressed sensing, etc.

Suggested texts and references:

1. R. C. Gonzalez and R. E. Woods: Digital Image Processing, Third Edition; Pearson (2013).
2. Bose and Tamal: Digital Signal and Image Processing; Wiley India (2008).

6.2 Choice-Based Credit System: Standalone Electives

6.2.1 SC 566/616/666 MJ : APPLICATIONS OF COMPUTERS TO CHEMISTRY

Topics to be covered:

1. Computational Chemistry. [1 hrs] Why learn Computational Chemistry? Applications areas: fundamental understanding. Predictions, design, structure of biomolecules, polymer design, catalyst and drug design.

2. Fundamentals of Chemistry. [2 hrs] Concepts in chemistry like valency, hybridization, electronegativity, covalent bond, ionic bond, hydrogen bond, co-ordinate bond. Geometries of molecules like linear, angular, tetrahedral, etc.

3. Molecular Representations and Search [4 hrs] Connectivity matrix, Adjacency matrix, SMILES notation, substructure search

4. Molecular Graphics and fitting. [4 hrs] 3-dimensional structures, steric pictures, CPK models, molecular dimensions and van der Waals volume.

5. Force Field (FF) Methods [4 hrs] Molecular mechanics expressions for bond stretch, bond angle, torsion, improper torsion, Van der Waals, electrostatics and cross terms. Types of force fields, computational aspects in FF, parameterization in FF. Evaluation of number of energy terms for a given molecule

6. Classical energy minimization techniques [3 hrs] Energy minimization by simplex, steepest descent, conjugate gradient and Newton-Raphson methods.

7. Conformational Analysis [4 hrs] What is a conformation? Systematic, Monte Carlo and genetic algorithm based conformational analysis. Polling method of conformational analysis, simulated annealing method

8. Semi-empirical QM calculations. [6 hrs] Cluster model calculations for the electronic structure of extended systems, Prospect and pitfalls in the usage.

9. Molecular Docking [4 hrs] Concepts in docking. Parameterization in docking, Rigid docking, flexible docking, virtual screening, Scoring functions

10. Molecular Descriptors [4 hrs] Molecular connectivity indices, topological indices, electrotopological indices, information theory indices, etc.

11. Quantitative Structure Activity Relationship [6 hrs] Generation of training and test set methods, variable selection methods like stepwise forward, backward, etc. Regression methods like multiple regression, principal component regression.

12. Futuristic modeling techniques. [2 hrs] Expert systems, Neural networks, Artificial Intelligence and virtual reality.

Suggested texts and references:

1. An Introduction to Chemoinformatics, A.R. Leach; V.J. Gillet, Kluwer Academic Publishers, The Netherlands, 2003.

2. Introduction to Computational Chemistry Frank Jensen Chichester, Wiley, 2006.

3. Molecular Modeling: Principals and Applications A.R. Leach Pearson Education Limited, Essex, 2001
4. Essentials of Computational Chemistry: theories and models Christopher J. Cramer John Wiley, 2004
5. Optimization in Computational Chemistry and Molecular Biology: local and global Approaches M. Panos, Christodoulos A. Pardalos Floudas - Science – 2000
6. Chemoinformatics: A Textbook J. (Johann) Gasteiger Thomas Engel – 2003

6.2.2 SC 567/617/667 MJ : SCIENTIFIC VISUALIZATION

Topics to be covered:

1. Introduction to Computer Graphics. [5 hrs] Examples of Graphics applications. Key journals in Graphics input and output graphics devices, world coordinate systems viewports and world to viewport mapping.
2. Raster graphics technique [5 hrs] Line drawing algorithms scan converting circles and ellipses, polygon filling with solid colors and filling patterns, half toning and dithering techniques.
3. Vectors and their use in graphics [2 hrs] Operation with vectors, adding, scaling, subtracting, linear spaces, dot product, cross Product, Scalar triple product, application of dot product, cross product and scalar triple product, application of vectors to polygons.
4. Transformation of pictures [3 hrs] 2D Affine transformations, use of Homogenous coordinates, 3D affine transformations.
5. 3D viewing [6 hrs] Synthetic camera approach, describing objects in view coordinates, perspective and parallel Projections, 3D clipping.
6. 3D graphics, Write frame models [5 hrs] Marching cube algorithms for contour generation and surface generation from a given data over 2D/3D grid.
7. Hidden line and surface removal, backface culling [7 hrs]
8. Light and shading models, rendering polygonal masks Flat, Gouraud, phone shading. [8 hrs]
9. Ray Tracing [11 hrs] Overview, intersecting ray with plane, square, cylinder, cone, cube and sphere, drawing shaded pictures of scenes, reflections and transparency.
10. ASSIGNMENTS:
    (a) 2D transformations
    (b) 3D transformations
    (c) Projections, Clipping, Shading
    (d) Contour, Marching cube
    (e) Creating a graphics package using above assignments.

Suggested texts and references:

1. Computer Graphics: A programming approach, S. Harrington (McGraw Hill, 1986)
2. Computer Graphics – Principles and Practice Foley, Van Dam
3. Computer Graphics, F. S. Hill Jr. (Macmillan, 1990)
4. Procedural Elements for Computer Graphics. D.F. Rogers (McGraw Hill, 1995)
5. Principles of Interactive Computer Graphics William Newman, Robert Sproull

---

6.2.3 SC 568/618/668 MJ : STATISTICAL COMPUTING

Topics to be covered:

1. Introduction [3 hrs] Computation of averages and measures of central tendency, skewness and kurtosis. Preparation of frequency tables, computation of Pearson and rank correlation coefficients.
2. Statistical Distributions [6 hrs] Evaluation of standard probability mass functions, cumulative density functions, and quantile functions.
3. Monte Carlo [4 hrs] Random number generation including Markov Chain Monte Carlo (MCMC). Statistical simulation and Monte Carlo studies in statistics.
4. Numerical optimization and root finding methods, including the Newton-Raphson method and the EM algorithm. [5 hrs]
5. Resampling techniques, including the permutation test and the bootstrap. [5 hrs]
6. Classification using discriminant functions, rough sets, artificial neural networks, and decision trees. [12 hrs]
7. Clustering techniques [10 hrs]
8. Multiple regression Analysis [6 hrs]

Suggested texts and references:

1. Simulation Modeling and analysis Averill M. Law and W. David Kelton
2. Applied Multivariate Statistical Analysis Richard A. Johnson and Dean W. Wichern
3. Data Mining: Concepts and Techniques Jiawei Han and Micheline Kamber
4. Data Mining : Multimedia, Soft Computing, Bioinformatics Sushmita Mitra and Tinku Acharya
5. Computational Statistics Geof H. Given and Jennifer A. Hoeting

---

6.2.4 SC 569/619/669 MJ : Applications of Computer to Physics

Topics to be covered:

1. Introduction to Mathematical modeling

   Any two of the following four topics :

2. Modeling Projectile Motion:

   Model trajectories using Euler's method for ODE's and Runge-Kutta Method. Problems related with trajectory of cannon ball, motion of batted ball, effect of air resistance, drag coefficient on the trajectories.

3. Oscillatory Motion Simple Harmonic Motion, Driven non-linear pendulum, chaos in non-linear pendulum

4. Solar System Planetary Motion, inverse square law and stability of planetary orbits. Three body problem and its effect on one planet

5. Waves Waves on a string with free ends/ without free ends, frequency spectrum of waves on a string, vibrating strings with frictional loss

   Any two of the following five topics :

6. Random Systems Random number generation, distribution functions, Monte Carlo methods, sampling, integration, Metropolis algorithm

   Random walks, self-avoiding walks, random walks and diffusion, cluster growth model, fractals, percolation Radioactive decay, discrete decay, continuous decay, decay simulation using Monte-Carlo

7. Molecular Dynamics Dilute gas with L-J potential, Methods for many body systems, Verlet algorithm, structure of atomic clusters, elementary ideas of ensembles, constant pressure, constant temperature, simulated annealing, melting

8. Dynamical Systems Linear, nonlinear models, Nonlinear growth - Logistic map, fixed points, period doublings, attractors, bifurcation diagram, Figenbaum constant,Henon Map, Lorenz Map, Chaos

9. Fractals Self-similarity, fractal dimension, L-systems, Self affine fractals,diffusion limited aggregation, cellular automata and its applications.

10. Ising Model Ising Chain – analytic solution, numerical solution, approach to thermal equilibrium, beyond nearest neighbors.

Suggested texts and references:

1. Computational Physics, Nicholas J. Giordano Prentice Hall (1997)

2. An Introduction to Computational Physics Tao Pang Cambridge University Press, (2006)

3. Projects in Scientific Computation Richard E. Crandall Springer (2000)

4. Introductory Statistics and Random Phenomena Manfred Denker, W. A. Woyczynski Birkhauser (2005)

5. Computational Physics Steven Koonin and Dawn Meredith Westview Press (1998)

6. Computational Physics J. M. Thissen Cambridge University Press (1999)

7. Computational Physics: Problem Solving with Computers Rubin. H. Landau, Manuel J. Paez, Cristian C Bordeianu Wiley-VCH (2007)

Assignments

1. Generate and plot trajectory of Cannon ball with and without air drag. 2. Effect of spin on trajectory of a thrown ball 3. Use Euler method for damped, nonlinear, driven pendulum. Plot trajectories. 4. Period-doubling and route to chaos in pendulum 5. Phase space plot of Lorenz model 6. Planetary trajectories with Euler-Cromer method. 7. Elliptical orbits and stability of orbits 8. Orbits of two planets under the influence of third planet 9. Signal from vibrating string. 10. Spectrum analysis of waves on string (fixed and free ends) 11. Random walk in one dimension 12. Two dimensional self-avoiding walk 13. Diffusion equation in two or three dimension 14. Diffusion limited aggregation in clusters with fractal dimensionality 15. Generate fractal curves recursively and obtain their dimensionality 16. Calculate M, E for the Ising model 17. Using Lennard-Jones potential, simulate a dilute gas, cluster of 10-20atoms. 18. Simulate cluster of atoms at a particular temperature. Plot the motion of a test particle, temperature of system as a function of time, pair separation and mean square displacement as a function of time.

6.2.5 SC 570/620/670 MJ : BIOLOGICAL SEQUENCE ANALYSIS

Topics to be covered:

1. Analysis of DNA and Protein sequence-distribution, frequency statistics, pattern and motif searches-randomization etc.-sequence segmentation. [10 hrs]

2. Sequence alignment – scoring matrices- PAM and BLOSUM-Local and global alignment concepts- dynamic programming methodology- Needleman and Wunsch algorithm, Smith –Waterman algorithm –statistics of alignment score-Multiple sequence alignment –Progressive Alignment. Database searches for homologous sequences – Fasta and Blast versions. [15 hrs]

3. Fragment assembly, Genome sequence assembly – Gene finding methods:- content and signal methods- background of transform techniques – Fourier transformation and gene prediction – analysis and predictions of regulatory regions. [10 hrs]

4. Neural network concepts and secondary structure prediction Probabilistic models: Markov chain-random walk-Hidden Markov models. Gene identification and other applications. [10 hrs]

5. Evolutionary analysis: distances-clustering methods – rooted and unrooted tree representation – Bootstrapping strategies [5 hrs]

Suggested texts and references:

1. Bioinformatics: a practical guide to the analysis of genes and proteins. A. Baxevanis and F.B.F. Ouellette(Eds.) John Wiley, New York (1998).

2. Introduction to computational biology: maps, sequences, and genomes. M.S. Waterman, Chapman and Hall, London (1995).

3. Proteome research: new frontiers in functional genomic M.R. Wilkins, K.L. Williams, R.D. Appel and D.H. Hochstrasser (Eds.), Springer, Berlin (1997).

6.2.6 SC 571/621/671 MJ : MODELLING OF BIOLOGICAL SYSTEM

Topics to be covered:

1. Concepts and principles of modeling. Limitations of models. [20 hrs] Identifying the components of a process / system – variables, controlled variables, rate constants, relationships between variables. Writing a set of equations, describing a process or a system. Types of solutions – integration, equilibrium solutions, numerical solutions. Models involving space: spatial simulations.

2. Models of behavior. [15 hrs] Foraging theory, Decision making – dynamic models, Game theory.

3. Modeling in Epidemiology and Public Health SIR models; Stochastic models and Spatial models. [10 hrs]

Suggested texts and references:

1. Models in Biology. B. Brown and P. Rothary , John Wiley and Sons, New York.

2. Evolutionary Genetics J. M. Smith,. Oxford University Press, Oxford (1989).

6.2.7 SC 572/622/672 MJ : Artificial Intelligence

Topics to be covered:

1. Conventional AI - Reasoning and Belief Systems a) Logical Inference [3 hrs] Reasoning Patterns in Propositional Logic, Propositional inference, Predicate calculus, Predicate and arguments, ISA hierarchy, Frame notation, resolution, Natural deduction etc. b) Reasoning under Uncertainty [3 hrs] Belief and uncertainty handling mechanisms, certainty, possibility and probability, Dempster Schaeffer theory , fuzzy inference, structure knowledge representation, semantic net, Frames, Script, Conceptual dependency etc. c) Goal Driven Intelligence(Planning, Search and Perception) [6 hrs] i) Planning: Formulation of Planning Problem, decomposition, representation of states, goals and actions, action schema, partial order planning, planning graphs Block world, strips, Implementation using goal stack, Non linear Planning with goal stacks, Hierarchical planning, List Commitment strategy. ii) Game Playing and Search: Heuristic search techniques. Best first search, mean and end analysis, A* and AO* Algorithm, Minimize search procedure, Alpha beta cutoffs, waiting for Quiescence, Secondary search, Perception - Action, Robot Architecture, Vision, Texture and images, representing and recognizing scenes, waltz algorithm, Constraint determination, Trihedral and non trihedral figures labeling d)Expert systems [3 hrs] Utilization and functionality, architecture of expert system, knowledge and rule bases, rule chaining strategies, conflict resolution, RETE algorithm, uncertainty handling in expert systems

2. Intelligent Agents and Computational Intelligence a) Agent Oriented Programming and Intelligent agents [3 hrs] Agent oriented programming as a paradigm, Agent orientation vs. object orientation, autonomous and intelligent agents, "Agency" and Intelligence, logical agents, multi agent systems, planning, search and cooperation using agents. b) Evolutionary Algorithms [3 hrs] Evolutionary paradigms, genetic algorithms and genetic programming, Ant colonies and optimization, evolutionary search strategies. c) Agents

, internet and Softbots [3 hrs] Interface agents and reactive systems , Softbots and info agents, the three layer model , process automation and agents, d) AI paradigms from biological, physical and social sciences [3 hrs] Swarms and collective intelligence, programming with swarms, fault tolerant systems, spin glasses and neural networks, self organizing systems, cellular automata and amorphous computing.

3. Statistical Learning Theory a) Learning Theory [3 hrs] Formulation of learning as a statistical problem, estimation of probability measure, empirical and structural risk minimization, Linear methods, supervised and unsupervised learning, regularization and kernel methods, model selection, inference and averaging, boosting and additive methods b) Applications and algorithms [3 hrs] Perceptrons and Neural Networks, Support vector machines, Classification and regression trees, nearest neighbors and EM clustering, Kohonen maps. c) Text Mining and Natural language processing [2 hrs] Sentence, syntax and semantic analysis, document classification, sentiment perception, theme and association mining

4. Hybrid Systems [6 hrs] Integration of data driven and concept driven methodologies, Neural Networks and Expert Systems hybrid, neural networks and game tree search hybrid, evolutionary systems and supervised learning hybrid, neuro-fuzzy systems, genetic programming for rule induction

Suggested texts and references:

1. AI: a modern approach Russell and Norvig:

2. AI Winston

3. Mathematical Methods in Artificial Intelligence Bender

4. Reasoning about Intelligent Agents Woolbridge

5. Artificial Intelligence. Elaine Rich and Kerin Knight:

6. Artificial Neural Network Kishen Mehrotra, Sanjay Rawika, K Mohan;.

6.2.8 SC 573/623/673 MJ : Quality Assurance and Software Testing

Topics to be covered:

1. Introduction to Software Quality Management Principles [2 hrs]

2. Software Quality Assurance & Quality Control [3 hrs] What and how Software Quality, Quality Goals for Software, Process Quality Goal, Product Quality Goal, Quality policy and Quality Objectives: Linkage and control

3. Quality Models [3 hrs] ISO 9001 – 2000 model, CMMI models, IS27000

4. Software Verification, Validation & Testing (VV & T) [3 hrs] Understanding verification, validation & testing, Quality improvement through activities (like Reviews, Inspection, Walkthroughs, Testing), Process improvement through VV & T

5. Software Testing Principles and Concepts [5 hrs] Software Testing Vocabulary, Testing and Quality, Who should Test, Independence in Testing, When Should Testing start?, Static versus Dynamic Testing, Testing and Debugging, The Cost of Quality, General Testing Process - Test Planning, Test Case Design, Test Case Execution, Test Analysis and Defect Reporting, Test Closure

6. Life Cycle Testing [5 hrs] Various Software Development Models, Levels of Testing (Unit Testing, Integration Testing, System Testing, User Acceptance Testing), OO-oriented Testing, Model Based Testing, The "V" Model of Testing, Early Testing, Verification and Validation, Retesting and Regression Testing

7. Testing Techniques [6 hrs] Static Testing Techniques (Reviews, Informal review, Walk-through, Technical review, Inspection), Functional /Specification based Testing Techniques, Structural Testing Techniques, Experienced based techniques, Choosing test techniques, Test Oracle, Building Test Cases, Process for Building Test Cases, Test Case Execution, Recording Test Results, Problem Deviation, Problem Effect, Problem Cause, Use of Test Results

8. Test Reporting Process [6 hrs] Prerequisites to Test Reporting, Define and Collect Test Status Data, Define Test Metrics used in Reporting, Define Effective Test Metrics, Test Tools used to Build Test Reports, Pareto charts, Cause and Effect Diagrams, Check Sheets, Histograms, Run Charts, Scatter Plot Diagrams, Regression Analysis, Multivariate Analysis, Control Charts, Test Tools used to Enhance Test Reporting, Benchmarking, Quality Function Deployment, Reporting Test Results, Current Status Test Reports, Final Test Reports, Guidelines for Report Writing

9. Test Management [5 hrs] Testing in an Organization, Test management documentation, Test plan documentation, Test estimation and scheduling of Test Planning, Analyzing Testing metrics, Test progress monitoring and control, Testing and risk, Risk management, Software Configuration Management, Change Management

10. Test tools [3 hrs] Types of Testing Tools and their use, Tool selection and implementation

Suggested texts and references:

1. Software Testing Techniques: Finding the Defects that Matter (Programming Series) Michael Shannon, Geoffrey Miller, Richard, Jr. Prewitt,

2. Software Testing Fundamentals: Methods and Metrics M. Hutcheson

3. "Software Testing: A Craftsman's Approach, Second Edition," Paul C Jorgensen, CRC Press, June 26, 2002.

4. "The Art of Software Testing," 2nd ed., Glenford J. Myers, John Wiley & Sons, Inc.

5. "Lessons Learned in Software Testing: a Context-Driven Approach," Cem Kaner, James Bach, and Bret Pettichord John Wiley & Sons, Inc.

6.2.9 SC 574/624/674 MJ : Soft Computing

Topics to be covered:

1. Neural Networks [15 hrs] a) Characterization b) The brain, neural networks and computers c) Neural networks and artificial intelligence Background, Applications, Neural network software, Learning paradigms - Supervised learning, Unsupervised learning, Reinforcement learning, Learning algorithms d) Neural networks and neuroscience Types of models, Current research e) History of the neural network analogy

2. Fuzzy Systems [15 hrs] a) Antilock brakes b) Fuzzy sets Fuzzy control in detail, Building a fuzzy controller c) History & applications d) Logical interpretation of Fuzzy control

3. Evolutionary Computing [15 hrs] a) Concept of Population genetics, probability, evolution principle b) Genetic Algorithms or Evolutionary Strategies i) Genetic Algorithms - General mechanism and terminologies, Selection, Crossover, Mutation ii) Evolutionary Strategies - Two-membered Evolutionary strategy, Multi-member Evolutionary strategy, Recombination c) Swarm Intelligence (one of the following three)

   Ant colony optimization, Particle swarm optimization, Stochastic diffusion search Applications

   ASSIGNMENT: To implement at least two of these major computational methods.

Suggested texts and references:

1. Neural Networks, A Classroom Approach Satish Kumar Tata McGraw-Hill Publishing Company Limited

2. Artificial Neural Networks Kishan Mehrotra, Chilkuri K. Mohan, Sanjay Ranka Penram International Publishing (India)

3. Neural Networks, A Comprehensive Foundation Simon Haykin Pearson Education

4. Genetic Algorithms, in Search, Optimization & Machine Learning David E. Goldberg Pearson Education

5. Artificial Intelligence and Intelligent Systems N P Padhy OXFORD University Press

---

6.2.10 SC 575/625/675 MJ : Design concepts and modeling

Topics to be covered:

1. Introduction to design process. [6 hrs] Building models suitable for the stages of a software development project. Introduction to UML.

2. Inception phase. [8 hrs] Structured analysis, scenario structures.

3. Elaboration phase. [10 hrs] Object modeling. Interfaces and abstraction. Information hiding.

4. Construction phase. [10 hrs] Coupling and object interaction. Responsibilities, defensive programming and exceptions. Functional decomposition, module and code layout. Variable roles, object state, verification and assertions. Design patterns.

5. Transition phase. [10 hrs] Inspections, walkthroughs, testing, debugging. Iterative development, prototyping and refactoring. Optimization.

Suggested texts and references:

1. Code complete: a practical handbook of software construction. McConnell, S. (1993) Microsoft Press.

2. UML distilled. Addison-Wesley (2nd ed.). Fowler, M. (2000).

6.2.11 SC 576/626/676 MJ : Business Analysis

Topics to be covered:

1. Introduction to DWH and OLAP [8 hrs]
   A. Decision Support System: Introduction to Decision Support System (DSS), DSS Components, Decision Types; Data warehouse (DWH): Need, Definition, Advantages of DWH, OLTP Vs DWH, 3-tier Architecture, DWH Design Process, ETL Process, DWH Backend Tools and Utilities, Metadata Repository, Models of DWH: Enterprise Warehouse, Data Mart, Virtual Warehouse, Comparison; OLAP: Data Cube and OLAP, Concept Hierarchies, OLAP Operations: Drill-Down, Roll-Up and Extreme Roll-Up, Slice-Dice and Pivot, OLAP Types, OLAP Query Processing, Computation of Data Cube. B. ETL Tools, Commercial DWH Vendors/ Tools and their Comparison, Project Failure Reasons, Data Analytics, Business Intelligence, SAS Software.

2. Dimensional Modeling [8 hrs]
   A. Dimensional Modeling: Dimensional Model Vs ER Model, DWH Schemas: Star, Snowflake, Fact Constellation, their Comparison, Techniques to Handle Changing Dimensions, Aggregation, Families of Fact Tables, Fact Less Fact Tables; Data Warehouse Indexing: Factors used to select an Indexing Technique, Properties of a Good Indexing Technique for DWH, Indexing Techniques: Projection Index, Bitmap Index (Pure and Encoded), Join Index and their Comparison. B. Case Studies of Data Warehouse Applications in various Industry Segments.

3. Data Mining and Functionalities [8 hrs]
   A. Introduction: Need of Data Mining, Knowledge Discovery in Database (KDD), Architecture of Data Mining System, Data Mining on Different kind of Data, Data Mining Functionalities; Data Preprocessing: Need, Cleaning, Integration, Transformation, Reduction, Discretization, Concept Hierarchy Generation; Cluster Analysis: Categories of Clustering methods, Partitioning methods: k-Means, k-Medoids ; Prediction: Numerical Prediction, Linear, Non-Linear Regression; Outlier Analysis: Applications, Techniques. B. Data Mining Task Primitives, Query Language, System Classification, Data Mining Issues.

4. Classification [9 hrs]
   A. Classification: Decision Tree Classifier, Rule Based Classification, Bayesian Classification, Neural Network Classification: Back Propagation Algorithm, Lazy Learner: kNN Classifier, Case-Based Reasoning, Other: Fuzzy Set Approach, Classifier Accuracy Measures, Techniques for Evaluating Classifier Accuracy; Frequent Itemset Mining: Interesting Item Set Mining: Market Basket Analysis, APriori Algorithm, Generating Association Rules, Types of Association Rules, Correlation Analysis. B. Support Vector Machine, Associative Classification, other Classification Techniques: Genetic Algorithm, Rough Set, Constraints Based Association Mining.

5. Data Mining on different Databases [7 hrs]
   A. Multimedia Data Mining, Web Mining, Text Mining, Spatial Data Mining, Mining on Social Networks, Multirelational Data Mining. B. Data Mining Applications, Trends/ Challenges of Data Mining, Mining Sequence Patterns in Transactional Database, Graph Mining, Data Mining Tools- Dbminer/ WEKA/ Oracle DM Tools/ OLE DB/ Ida.

Suggested texts and references:

1. The Data Warehouse Lifecycle Toolkit Kimball, Reeves, Ross, Thornthwaite John Wiley
2. Data Mining: Concepts and Techniques Jiawei Han and Micheline Kamber, Morgan Kaufman

---

6.2.12 SC 577/627/677 MJ : Applied Cryptography

Expected Course Outcome: This is an introductory course on modern cryptography, providing essential mathematical background knowledge required to understand cryptographic principles. The course will cover both symmetric key and asymmetric key algorithms, also covering their cryptanalysis. Practical implementations of these algorithms will be demonstrated. The curriculum includes algorithms for digital signatures. In addition to the fundamental concepts, the course will touch upon advanced topics such as zero-knowledge proof, homomorphic encryption, and post-quantum cryptography, etc. Case studies will be presented to showcase the real-world applications of cryptography and how it plays a crucial role in various well-known scenarios. By the end of this course, students will have a comprehensive understanding of modern cryptography, equipped with the necessary skills to apply cryptographic techniques securely.

Prerequisites: An undergraduate-level understanding of C/C++ or Python.

Topics to be covered:

1. **Mathematical background**
   1. **Number theory** - Primes, Euclids algorithm, Primality testing, Chinese Remainder Theorem
   2. **Abstract algebra** - Preliminaries of - Groups, Rings and Fields for cryptography
   3. **Theory of elliptic curves** - Preliminaries of - Points on a curve, algorithms for point addition and doubling, paring in elliptic curve

2. **Algorithms**
   1. Introduction to Cryptography : Need for cryptography, Classical cryptography, Shift Cipher, Vigenere Cipher, Hill Cipher
   2. Symmetric key algorithms : Advanced Encryption Standard (AES), Blowfish, Twofish
   3, Asymmetric key algorithms : Key Exchange : Diffie Hellman, ECDH. Encryption: RSA, ElGamal, ECDLP. Digital signature : RSA, ElGamal signature, Elliptic curve digital signature algorithm
   4. Cryptanalysis : Known attacks on RSA, Known attacks on DLP, Known attacks on ECDLP

3. **Advanced Topics**
   Zero-knowledge proof, Homomorphic encryption, Post-quantum cryptography, Side-Channel Attacks, etc.

4. **Case studies**
   1. WhatsApp : How end-to-end encryption is implemented ?
   2. Google Pay / PhonePe : How UPI transactions are secured ?
   3. Crypto-currencies / Blockchain : How do BitCoin and Ethereum decide ownership of a coin? **4.TLS/SSL** : Transport Layer Sercurity protocol to protects internet communications

5. **Industrial applications** [Optional]

6. **Assignments**
   All the algorithms from 2 above are to be implemented using **C++** and **NTL** (Number theory library - Victor Shoup). Alternatively, **SageMath** can be used. For the cryptanalysis of public key algorithms, openMPI can be used.

   1. Implement : Shift cipher, Vigenere Cipher, Hill Cipher, AES, BlowFish, TwoFish, Diffie Hellman, ECDH (key exchange), RSA, ElGamal, ECC algorithms.
   2. Cryptanalysis of : RSA, ElGamal, ECDLP algorithms
   3. Demonstrate : Wireshark, OpenSSL, PGP, etc.

Suggested texts and references:

1. Stinson, D. R. **Cryptography: Theory and Practice (3rd Edition)**. CRC press.
2. Schneier, B. **Applied cryptography: protocols, algorithms, and source code in C (2nd Edition)**. John Wiley & Sons.
3. Menezes, A. J., Van Oorschot, P. C., & Vanstone, S. **A. Handbook of applied cryptography**. CRC press.
4. Cohen, H., Frey, G., Avanzi, R., Doche, C., Lange, T., Nguyen, K., & Vercauteren, F. (Eds.). **Handbook of elliptic and hyperelliptic curve cryptography**. CRC press.
5. Burton, D. **Elementary Number Theory**. McGraw Hill.
6. Niven, I., Zuckerman, H. S., & Montgomery, H. L. **An introduction to the theory of numbers**. John Wiley & Sons.
7. Das, A. **Computational number theory**. CRC Press.
8. Silverman, J. H., Pipher, J., & Hoffstein, J. **An introduction to mathematical cryptography**. Springer New York.

6.2.13 SC 578/628/678 MJ : Cloud Solution Architect

Topics to be covered:

1. Cloud Fundamentals
2. Cloud Services - AWS as a case study–

```
1. Security, Identity and Compliance
  a. Identity and access management (IAM)
  b. Key management service (KMS)
  c. Single Sign ON
  d. Cognito, etc.

2. Compute
  a. Elastic Cloud Compute (EC2)
  b. Server less - Lambdaa
  c. BeanStack, etc.

3. Storage
  a. Simple Storatge Service (S3)
  b. Elastic File Storage/Elastic Block Storage
  c. Simple Storatge Service - Glacier
  d. FSx
  e. Storage Gateway, etc.

4. Networking and Content Delivery
  a. Virtual Private Cloud (VPC)
  b. Cloud Front
  c. Route-53
  d. API Gateway
  e. Direct Connect, etc.

5. Managment and Governance
  a. Cloion
  b. Cloudwatch Format
  c. Cloud Trail
  d. Trusted Advisor, etc.

6. Database
  a. Relation Database Service (RDS)
  b. DynamoDB
  c. Elastic Cache
  d. Redshift, etc.

7. Miscellaneous Services
  a. Kinesis
  b. Athena/Macie
  c. HPC in the Cloud
  d. Auto Scaling
```

3. Architectures

- Design Resilient Architectures
- Design Performant Architectures
- Specify Secure Applications and Architectures
- Design Cost-Optimized Architectures
- Define Operationally-excellent Architectures

4. Infrastructure as a Code
5. Introduction to cloud platforms - Microsoft Azure, Google Cloud Platform.
