Semester VI

DISCIPLINE SPECIFIC CORE COURSE – 16 (DSC-16)
VI.1. Numerical Methods for Computational Mathematics

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
Numerical Methods for Computational Mathematics DSC 16, VI.1	**4**	**2**	**0**	**2**	**12th Pass with Mathematics**	**Linear Algebra, Programming Language, Ordinary Differential Equations**

Learning Objectives

In practical scenarios, the governing mathematical models are usually too complex to be solved analytically and numerical techniques become the only way out to approximate the solutions. This paper aims to teach the student solutions of nonlinear equations in one variable with error analysis, interpolation and approximation, numerical differentiation and integration, direct methods for solving linear systems, numerical solution of ordinary differential equations. By the end of this paper, students should have the ability to compare the computational methods for advantages and drawbacks and choose the suitable computational method among several existing methods for underlying physical problems. In this paper, students will write codes in MATLAB/C/C++ for implementation of numerical methods.

Learning outcomes

After completing this course, student should be able to;

- Understand the need of numerical techniques and their importance in real situations
- Learn different techniques of solving non-linear equations such as Bisection method, Newton Raphson method, Regula Falsi method, Secant method & Iterative methods

- Analyze errors associated with these methods and their rate of convergence
- Learn Gauss elimination, Gauss seidel, LU decomposition methods for solving system of linear equations with pivoting concepts
- Learn polynomial interpolation, linear and cubic spline interpolations, analyze errors of interpolation
- Conceptualize numerical integration and errors associated with it.
- Learn Euler's method and Runge-Kutta method for numerical solution of differential equations
- Write programs of all these numerical methods in MATLAB/C/C++

SYLLABUS

Unit I: Solving Nonlinear Equations - Graphical method - Bracketing and Non-bracketing approach. - Bisection, Method of false position, Iterative method, Newton-Raphson method and Secant method - Errors and rate of convergence **(8 Hours)**

Unit II: Matrix notation of a system of linear equations - Gaussian elimination and Gauss-seidel method – Pivoting - Row-echelon form - LU factorization **(6 Hours)**

Unit III: Polynomial interpolation - Forward, Backward and Divided differences - Piecewise linear and Cubic Spline interpolation - Errors in interpolation **(6 Hours)**

Unit IV: Newton-Cotes Integration Formula - Trapezoidal and Simpson's rules - Gaussian quadrature, Euler, Modified Euler and Runge-Kutta methods for solution of differential equations - Power method, QR method for Eigen Value problems **(10 Hours)**

Practicals- (60 Hours)

- Writing MATLAB/C/C++programs for finding root of the equations using Bisection,Newton- Raphson, Iterative and Secant methods
- Writing MATLAB/C/C++programs for solving system of linear equations (Gaussian Eliminations, Gauss Jacobi & Gauss Seidel Method)
- Writing MATLAB/C/C++ programs for interpolation, forward, backward and divided difference
- Writing MATLAB/C/C++ programs for methods of numerical integration
- Writing MATLAB/C/C++ programs for Euler and Runge-Kutta methods.

Essential/recommended readings

- Applied Numerical Analysis, C. F. Gerald and P. O. Wheatly, Pearson Education India, 2007.
- Introduction to Applied Numerical Analysis, R. W. Hamming, Dover Publications, 2012.

- Elementary Numerical Analysis- An Algorithmic Approach, S. D. Conte and Carl de Boor, McGraw-Hill, 1980.
- Numerical Recipes: The Art of ScientificComputing,3rd Edition, William H.Press, Saul A.Teukolsky,William T. Vetterling, Brian P. Flannery, Cambridge University Press, 2007

DISCIPLINE SPECIFIC CORE COURSE – 17 (DSC-17)
VI.2. Information Security

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical / Practice		
Information Security, DSC-17, VI.2.	**4**	**3**	**0**	**1**	**12^{th} Pass with Mathematics**	**Programming Language**

Learning Objectives

This course will discuss the fundamentals of cryptography and its application to network security. Understand network security threats, security services, and countermeasures. Understand vulnerability analysis of network security. Acquire background on hash functions; authentication; firewalls; intrusion detection techniques. Gain hands-on experience with programming and simulation techniques for security protocols. Understand the tradeoffs and criteria/concerns for security countermeasure development. Apply methods for authentication, access control, intrusion detection and prevention. Identify and mitigate software security vulnerabilities in existing systems.

Learning outcomes

After completing this course, student should be able to;

- Understand and explain the risks faced by computer systems and networks.
- Identify and analyze security problems in computer systems and networks.
- Explain how standard security mechanisms work.
- Develop security mechanisms to protect computer systems and networks.
- Write programs that are more secure.
- Use cryptography algorithms and protocols to achieve computer security.

SYLLABUS

Unit I: Introduction to Information Security, CIA, Conventional Cryptographic Techniques: Substitution and transposition ciphers, One Time Pad. **(12 Hours)**

Unit II: Block cipher and Stream Cipher, Steganography**:** Symmetric and Asymmetric Cryptographic Techniques: DES, AES, RSA algorithms, Authentication and Digital Signatures, Secure Hash function.
(13 Hours)

Unit III: Program Security: Nonmalicious Program errors – Buffer overflow, Incomplete mediation, Time-of-check to Time-of- use Errors, Viruses, Trapdoors, Salami attack, Man-in-the- middle attacks, **(10 Hours)**

Unit IV: Threats in networks, Network Security Controls – Architecture, Wireless Security, Honeypots, Traffic flow security, Firewalls, Types of Firewalls, Personal Firewalls, IDS, Email Security – PGP,S/MIME **(10 Hours)**

Practicals: (30 Hours)

- Implementing trans-positional ciphers.
- Implementing substitution ciphers
- Using block and stream ciphers from various available libraries
- Implementing Al-Gamal Key sharing algorithm
- Implementing and using AES/DES and RSA algorithm
- Understanding authentication practically
- Simulating attacks on system

Essential/recommended readings

- Security in Computing, Fourth Edition, by Charles P. Pfleeger, Pearson Education
- Cryptography And Network Security Principles And Practice, Fourth or Fifth Edition, William Stallings, Pearson
- Modern Cryptography: Theory and Practice, by Wenbo Mao, Prentice Hall
- Network Security Essentials: Applications and Standards, by William Stallings. Prentice Hall

DISCIPLINE SPECIFIC COURSE -18 (DSC 18)
VI. 3: Artificial Intelligence

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course
		Lecture	Tutorial	Practical		

				/ Practice		(if any)
Artificial Intelligence DSC 18, VI.3	**4**	**3**	**1**	**0**	**12th Pass with Mathematics**	**Programming, Data Structure, Design and Analysis of Algorithms**

Learning Objectives

The objective is to introduce the basic principles and techniques of Artificial Intelligence. The course provides a theoretical foundation for variety of concepts in the field of artificial intelligence. To enhance the practical understanding of AI concepts, the course expects problem-solving projects as a 'hands-on' approach and avenue for exploration and creativity.

Learning outcomes

Upon completion of this course, the student will be able to:

- Learn the fundamentals of artificial intelligence.
- Learn to problematize the problems and solve them.
- Understand and implement search and adversarial (game) algorithms.
- Understand mathematical models such as belief networks and apply them to a range of AI problems.
- Have a glance at machine learning algorithms and extracting knowledge models from data.
- Understand the fundamentals of Machine learning and Reinforcement learning.

SYLLABUS

Unit I: Philosophy of Artificial Intelligence, Intelligent Agents **(9 Hours)**

UNIT II: Problem-solving, Search techniques, Constraint satisfaction, Game playing, Automated Planning **(9 Hours)**

UNIT III: Knowledge Representation and Reasoning through Logic, Bayesian Networks, Markov Decision Processes **(12 Hours)**

UNIT IV: Machine Learning, and Reinforcement Learning **(15 Hours)**

Practicals- (30 Hours)

- Implementation of problem-solving and search techniques.
- Implementation of hill climbing and its variations.
- Implementation of genetic algorithm search.
- Implementation of heuristics search techniques.
- Implementation of machine learning algorithms and their applications.
- Development of artificial intelligence projects.

Essential/recommended readings

- Russell, S., & Norvig, P. (2021). Artificial intelligence: A modern approach, global edition 4th. Foundations, 19, 23.

- Poole, D. L., & Mackworth, A. K. (2010). Artificial Intelligence: foundations of computational agents. Cambridge University Press.
- Kulkarni, P., & Joshi, P. (2015). Artificial intelligence: Building intelligent systems. PHI Learning Pvt. Ltd.
- Artificial Intelligence, 3rd Edition. R. Elaine, K. Knight, S. Nair, Tata McGraw-Hill, 2009.
- Bishop, C. M., & Nasrabadi, N. M. (2006). *Pattern recognition and machine learning* (Vol. 4, No. 4, p. 738). New York: Springer.
- Winston, Patrick Henry, Artificial Intelligence. 3rd ed. Addison-Wesley, 1992.
- Kevin P. Murphy and Robert R. Reitano, Machine Learning: A Probabilistic Perspective, MIT Press, 2012.