M. Tech.
Modeling & Simulation

Savitraibai Phule Pune University

Revised curriculum document to be implemented from AY 2025-26.

https://scms.unipune.ac.in

Contents

I

Chapter

Preamble

The **M.Tech. in Modeling & Simulation** program at Savitribai Phule Pune University is designed to equip students with advanced knowledge, technical expertise, and research capabilities in the rapidly evolving field of computational modeling and simulation. This interdisciplinary program bridges mathematics, computer science, and domain-specific applications, fostering innovation and problem-solving skills applicable to a wide range of industries and research areas.

The revised curriculum emphasizes a multidisciplinary approach, research-driven learning, and industry-oriented training. The program aims to prepare students for challenges in areas such as artificial intelligence, scientific computing, optimization, and data-driven modeling, while also encouraging creativity, critical thinking, and ethical practices.

The restructured syllabus integrates core theoretical concepts, elective specializations, hands-on laboratory training, and project-based learning. With a balanced credit distribution of foundational courses, advanced electives, and research components, the program ensures holistic development and prepares graduates for roles in academia, R&D organizations, and the tech industry.

This document outlines the credit framework, semester-wise structure, and key components of the M.Tech. program, reflecting the university's commitment to fostering academic excellence and addressing the demands of emerging technologies and societal needs.

2

Chapter

About

2.1 About this document

This is the curriculum document for the M.Tech Programme in Modeling & Simulation. The programme, designed by a core group of people associated with the Department of Scientific Computing, Modeling and Simulation [1], Savitribai Phule Pune University. The M. Tech Programme was approved by the University in 2007, and came into existence in the academic year 2008-09. The present document outlines a university-approved revision of the curriculum of the programme.

2.2 What changed?

This document is grounded in the foundational vision articulated in the original M.Tech. program framework [1], as well as its subsequent revisions, most notably the revision made in AY2022 [2]. While preserving the core objectives of its predecessors, this revised version introduces substantive updates to cope up with external demands. Among the principal modifications are:

- the provision for exit and re-entry pathways,
- the incorporation of on-the-job training opportunities,
- a comprehensive restructuring of the credit structure, and
- enhanced curricular flexibility to support individualized student learning trajectories.

[1] Formerly known as 'Centre for Modeling and Simulation'

2.2.1 Citing this document

Core Curriculum Team and Contributors, Master of Technology (M.Tech.) Programme in Modeling and Simulation 2025 . Public Document CMS-PD-20220913 of the Department of Scientific Computing Modeling and Simulation, Savitribai Phule Pune University, Pune 411007, India (2025); available at http://scms.unipune.ac.in/reports/.

2.2.2 Credits and Acknowledgments

Core Curriculum Team: Bhalchandra Pujari and Mihir Arjunwadkar.

Contributors: Listed under each individual syllabus separately.

Writing, Collation, Editing: Bhalchandra Pujari and Mihir Arjunwadkar. The LaTeX template is based on yLaTeX[3] repository. The authors acknowledges the use of Large Language Models[4, 5] for assistance in editing, and refining parts of this document, specifically in LaTeX customization.

Organizational Support: Arun Banpurkar, as Head, Department of Scientific Computing, Modeling and Simulation; Leelavati Narlikar, and Abhijat Vichare as members of the *ad hoc* Board of Studies in Scientific Computing, Modeling & Simulation.

2.3 About the department

The Centre for Modeling and Simulation, Savitribai Phule Pune University, was established in August 2003 with a vision to promote modeling and simulation methodologies and, in keeping with worldwide trends of modern times, to encourage, facilitate, and support highly interdisciplinary approaches to basic and applied research that transcend traditional boundaries separating individual knowledge disciplines. In 2020, the Centre was merged into Department of Scientific Computing, Modeling and Simulation, Savitribai Phule Pune University. This academic programme is now inherited by **Department of Scientific Computing, Modeling and Simulation**. For more information, visit https://scms.unipune.ac.in/

2.4 About the degree programme

Title: M. Tech. in Modeling & Simulation.

Board of Studies: Scientific Computing, Modeling & Simulation, under Faculty of Science and Technology.

Mode: Full time, autonomous, in-person.

Medium: English

Intake: Regular Admissions: 40, and as per the prevailing Savitribai Phule Pune University policies.

Duration: 2 (Two) years (with the provision of Early Exit)

Early Exit: Students completing one year (with all due 44 credits) can opt out of the course, and will be awarded *Post Graduate Diploma in Modeling & Simulation* as per NEP2020.

Admissions:

Default entry: Admissions to first year through entrance test, and as per the prevailing Savitribai Phule Pune University policies.

Lateral Entry: Admissions to second year based on the grades of Post Graduate Diploma in Modeling & Simulation. Number of seats may be limited by the prevailing university policies.

Eligibility:

Default entry: B.E./B.Tech. any branch OR M.Sc. in any subject with mathematics/statistics equivalent to at least second year of B. Sc.

Lateral Entry: Admission to second year with Post Graduate Diploma in Modeling & Simulation.

Fees: As per the prevailing Savitribai Phule Pune University policies.

Examinations:

- A student will have to complete a total of 88 credits, details of which are given in the enclosed course structure, to obtain the degree.
- Each credit will be for 25 marks, 50% of the marks for continuous assessment and 50% marks for term-end examination.
- Continuous assessment may be through Seminars, Assignments, Oral test, Written test, or any other mean that faculty deem fit.
- The term-end examination for theory courses will be in the form of written examination for the whole course
- The continuous assessment for a laboratory course may involve completing and submitting desired set of assignments throughout the semester. The term-end examination may involve completing an assessment/viva-voce/written test or seminar based on the assignments.
- For courses designated as hybrid, the credits may be split in theory and laboratory modules, with each module having its separate examination.
- As per the existing rules, the student has to obtain 40% marks in the combined Continuous Assessment and Semester-End Assessment with a minimum passing of 30% in both these separately.
- The rules of examination prescribed by the University will be applicable.

Programme outcome

Graduates of the **M.Tech. in Modeling & Simulation** program are expected to:

$\mathcal{P}1$: Develop advanced expertise in mathematical modeling, simulation techniques, and computational tools.

$\mathcal{P}2$: Conduct independent research and innovate solutions to real-world challenges.

$\mathcal{P}3$: Solve complex interdisciplinary problems using a multi-domain approach.

$\mathcal{P}4$: Communicate technical ideas effectively and work collaboratively in teams.

$\mathcal{P}5$: Adhere to ethical practices and contribute to sustainable development.

$\mathcal{P}6$: Pursue leadership roles in academia, industry, or research institutions with lifelong learning abilities.

3
Chapter

Credit Framework

3.1 Framework

National Credit Framework

The framework draws its inspiration from the guidelines provided by the ‘the High-Level Inter-Ministerial Committee on National Credit Accumulation & Transfer Framework’ in **National credit framework**[6]. The given framework tries to embody the spirit of “multi-disciplinary, holistic education, allowing imaginative and need based curricular structures and enabling creative combinations of subjects and disciplines”.

Table 3.1 Summary of Credit Distribution

Component	**Credits**
Major Core Courses	32
Major Elective Courses	22
Research Methodology	4
On Job Training	4
Research Project/Internship	26
Credits per Semester	**22**
Total Credits (2 years)	**88**

3.2 Semester-wise structure

Table 3.2 Credit Distribution by Semester. The rows in **grey** indicate elective courses.

Course code	Course	Credits	Scheme
SEMESTER 1 (Core:20, Elective:2, Project:0)			
MNS-501-MJ	Foundations of Mathematics	4	Theory
MNS-502-MJ	Linear Algebra	4	Theory
MNS-503-MJ	Probability Theory	4	Hybrid
MNS-504-MJP	Fundamentals of Computing	4	Laboratory
MNS-505-MJP	Research Methodology in Modeling & Simulation	4	Laboratory
MNS-506-MJ	Colloquium in M&S	2	Seminar
Total: 22 Credits			
SEMESTER 2 (Core:10, Elective:8, Project:4)			
MNS-551-MJP	Numerical Computing 1	2	Laboratory
MNS-552-MJ	Optimization 1	2	Theory
MNS-553-MJ	Statistical Inference	4	Theory
MNS-554-MJP	Overview of Modeling & Simulation	2	Laboratory
MNS-555-OJT	On Job Training	4	Internship
Elective	Choice from Set 1 (Sec. 5.1)	4	
Elective	Choice from Set 1 (Sec. 5.1)	4	
Total: 22 Credits			
SEMESTER 3 (Core:6, Elective:12, Project:4)			
MNS-601-MJP	Numerical Computing 2	2	Laboratory
MNS-602-MJ	Optimization 2	4	Hybrid
MNS-603-MJ	MOOC course	4	Hybrid
MNS-604-RP	M&S Project	4	Project
Elective	Choice from Set 2 (Sec. 5.2)	4	
Elective	Choice from Set 2 (Sec. 5.2)	4	
Total: 22 Credits			
SEMESTER 4(Core:0, Elective:0, Project:22)			
MNS-651-RP	Dissertation	22	Internship
Total: 22 Credits			

3.3 Achieving Programme Outcomes

In designing the curriculum for this program, individual courses were carefully aligned to support one or more of the program outcomes described in Section

2.4. Each course is intended to play a specific role in advancing particular competencies or areas of understanding, ensuring that students receive a coherent and comprehensive education.

While section 2.4 provides a description of the program outcomes, Table 3.4 presents a mapping of the individual courses to these outcomes, highlighting the specific contributions each course makes. This mapping not only ensures transparency in curriculum design but also aids in evaluating how effectively the program achieves its intended objectives. Instructors are advised to go through this table before offering a course.

Course	$\mathcal{P}$**1**	$\mathcal{P}$**2**	$\mathcal{P}$**3**	$\mathcal{P}$**4**	$\mathcal{P}$**5**	$\mathcal{P}$**6**
MNS-501-MJ	■		■			
MNS-502-MJ	■		■			
MNS-503-MJ	■		■			
MNS-504-MJP	■		■			
MNS-505-MJP	■	■	■			
MNS-506-MJ		■		■	■	■
MNS-551-MJP	■		■			
MNS-552-MJ	■		■			
MNS-553-MJ	■		■			
MNS-554-MJP	■		■			
MNS-555-OJT		■	■	■	■	■
MNS-601-MJP	■		■			
MNS-602-MJ	■		■			
MNS-603-MJ	■	■	■		■	
Electives (Set1/2)	■	■	■	■		
MNS-651-RP	■	■	■	■	■	■

Table 3.4 Targeted programme outcomes (see Sec 2.4) by each course.

3.4 Mandatory Extra credits

UGC recommends following Extra credit courses (Total credits 12, Compulsory, not counted towards final grades). These courses are typically conducted in online mode through the University or the UGC. There are subjected to change as per the UGC directives.

Subject Code	Subject Title	Credits
CBCS-CS	Cyber Security	4
SDC01	Skill Development Courses	4
CBCS-HR	Human Rights Education	2
CBCS-IIC	Introduction to the Indian Constitution	2

Table 3.5 Mandatory Extra Courses as per the recommendations of UGC and the University.

4

Chapter

Major Core Credits

4.1 Semester 1

The first semester of the M.Tech program in Modeling and Simulation lays a comprehensive foundation in both theoretical and practical aspects of mathematical and computational modeling. Foundations of Mathematics and Linear Algebra develop essential skills in logical reasoning, abstraction, and algebraic structures—tools critical for formulating and analyzing models. Probability Theory and its companion course, Practical Probability, introduce students to the mathematics of uncertainty along with hands-on experience in applying probabilistic methods to real-world problems. Fundamentals of Computing equips students with programming and algorithmic skills necessary for implementing simulations and working with data. Complementing these technical courses, Research Methodology guides students in the principles of scientific inquiry, literature review, and structured research design—preparing them to engage with complex modeling challenges in both academic and applied contexts.

Common Prerequisites (CP)

> All course prerequisites are interpreted as indicative of the minimum background necessary to assimilate the course content meaningfully.
>
> 1. Proficiency in Mathematics at 12+2-level science (i.e., S.Y.B.Sc.) and engineering (i.e., M1+M2+M3) programmes of Savitribai Phule Pune University.

4.1.1 MNS-501-MJ: Foundations of Mathematics

Credits: 4 **Category:** Theory **Prerequisites.** CP

Course Outcomes

A practical hands-on understanding of this fundamental area of mathematics is essential for dealing with the mathematical complexity of many kinds of mathematical model. This foundational course is intended to bring the student at an acceptable level of understanding of basics of mathematics. Specifically, this course intends to enable the student to

1. analyze vector functions;
2. find derivatives, line/surface/volume integrals, arc lengths, and curvatures;
3. understand and apply gradient, divergence, and curl operators; Gauss and Stokes theorems.
4. understand real-valued sets, sequences, series;
5. understand functions: properties and visualization; convergence, limits, continuity;
6. interpret differentiation and (multiple) integration, their interrelation

Syllabus.
Module I: Real Analysis & Calculus

1. Sets. Basics of set theory. Relations and functions. Open and closed sets. Countability.
2. Real numbers. Real numbers, real sequences, infinite series, convergence and tests of convergence.
3. Real functions. Real functions of single and several real variables, plotting graphs of such functions, limits, continuity.
4. Real functions of one real variable. Derivative, Rolle's and Lagrange mean value theorems, Taylor's theorem, order notation, extreme values and indeterminate forms.
5. Real functions of several real variables. Differentiability, Young and Schwarz theorems, partial derivatives, Taylor's theorem and extreme values, homogeneous functions and Euler's theorem, implicit functions, Jacobians.
6. Integration. Revision of integration of functions of one variable, definition, standard results and methods of integration, interpretation as area under graph, infinitesimals and Riemann sums.

7. Multiple integration. Methods of multiple integrals and their interpretation, Green's Theorem, Fubini's theorem, change of variables.

Module II: Vector Calculus

1. Scalar and vector fields, surfaces and curves in space and their parametric equations.
2. Continuity and differentiability of vector and scalar fields. Partial derivatives of vector and scalar fields, the operator ∇ in Cartesian, cylindrical and spherical coordinate systems.
3. Gradient of a scalar field, level/equipotential surfaces, directional derivative and interpretation of gradient, tangent plane and normal to level surfaces.
4. Divergence and curl. Important identities relating gradient, divergence and curl.
5. Flux of a vector field through a surface, vector line/surface integrals.
6. Gauss divergence theorem. Interpretation of divergence in terms of flux.
7. Stokes' theorem. Interpretation of curl in terms of vector line integrals.

Suggested Texts/References.

1. S. C. Malik and Savita Arora, *Mathematical Analysis*. New Age Publishers, 2009.
2. James Stewart, *Essential Calculus: Early Transcendentals*. Thomson/Brooks-Cole, 2007.
3. Open calculus texts at https://openstax.org/.
4. Richard Courant and Fritz John, *Introduction to Calculus and Analysis, Vol 1 and Vol 2*. Springer, 1998.
5. Erwin Kreyszig, *Advanced Engineering Mathematics*. Wiley India, 2014.
6. Michael Greenberg, *Advanced Engineering Mathematics*. Pearson, 2002.
7. A. R. Vasishtha and Kiran Vasishtha, *Vector Calculus*. Krishna Prakashan Media, 2007.
8. Anil Kumar Sharma, *A Textbook of Vector Calculus*. Discovery Publishing House, 2006.
9. Shanti Narayan and P. K. Mitta, *A Textbook of Vector Calculus*. S. Chand, 1987.

Notes on Pedagogy. Depending upon the capacity of the batch of students, previous orientation and training, the teacher can adjust the depth of delivery so as to best meet the objective. The content can also be tuned accordingly. The content could even be ordered and modified according to the presentation in the prescribed text book/s.

Contributor/s.

- Bhalchandra Gore (https://scms.unipune.ac.in/~bwgore)

- Snehal Shekatkar (http://inferred.co/)
- Sukratu Barve (https://scms.unipune.ac.in/~sukratu)

4.1.2 MNS-502-MJ: Linear Algebra

Credits: 4 **Category:** Theory **Prerequisites.** CP

Course Outcomes

This fundamental branch of mathematics has ramifications almost everywhere where mathematics is used, including mathematical modeling, statistics, and computing. 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.

Syllabus.

1. Vector spaces. Introduction to fields. Definition of vectors and vector space over a field. Vector subspaces, linear independence, span, bases, dimension and its uniqueness. Linear Transformations, change of basis, similarity transformation.
2. Linear operators & Matrices Definition and properties. Null space and range. Matrices and matrix operations. Determinants: Definitions, Properties, Inverses
3. Systems of linear equations. Row echelon form and reduced row echelon form (Gaussian elimination). Rank of a matrix, Finding the inverse of a matrix. Solving a system of linear equations (Illustrative Examples)
4. Eigenvalues, eigenvectors and diagonalization. Definition of eigenvectors and eigenvalues of linear operators. Diagonalization using a particular similarity transformation (explicit computations), application in linear equations. (Optional) : Linear ODEs, normal matrices and diagonalizability.
5. Orthogonality & Inner product spaces. Dot product and norm, Orthogonal & orthonormal sets, Gram-Schmidt orthogonalization process
6. More on matrices Symmetric and skew-symmetric matrices orthogonal matrices and their properties, positive definite and semidefinite matrices. Singular value decomposition (SVD) introduction

7. Singular Value Decomposition (SVD) Definition and computation of singular values, Properties of SVD, Applications of SVD in data compression and signal processing

Suggested Texts/References.

1. G. Strang, *Linear Algebra and Its Applications*. Cengage Learning (4th Edition), 2006.
2. Paul Halmos and John L. Kelley, *Finite Dimensional Vector Spaces*. Literary Licensing, LLC, 2013.
3. Kanti Bhushan Datta, *Matrix and Linear Algebra*. Prentice Hall India, 2008.
4. S.K. Mapa, *Higher Algebra: Abstract and Linear*. Levant Books, 2011.
5. Seymour Lipschutz and Marc Lipson, *Linear Algebra (Schaum Series)*. McGraw-Hill India, 2005.
6. Otto Bretscher, *Linear Algebra with Applications*. Pearson, 2008.
7. Georgi Shilov, *Introduction to the Theory of Linear Spaces*. Martino Fine Books, 2013.

Notes on Pedagogy. Linear algebra is an abstract subject and students find it difficult to comprehend. Nonetheless it provides vital background for advances courses to follow. It is advisable not to trivialize the concept of vectors by restricting to 3D coordinate space.

Contributor/s.

- Varsha Gejji (https://in.linkedin.com/in/varsha-gejji-8b675238),
- Bhalchandra Pujari (https://scms.unipune.ac.in/~bspujari)

4.1.3 MNS-503-MJ: Probability Theory

Credits: 4 **Category:** Hybrid **Prerequisites.** CP

Suggested credit distribution: Theory: 3, Lab: 1

Course Outcomes

Probability is the mathematical language for quantifying uncertainty or ignorance, and is the foundation of statistical inference and all probability-based modeling. This course attempts to develop

1. perspective on modeling using probability by way of real-life contexts and examples;
2. solid understanding of probability theory as basis of statistical inference;
3. familiarity with basic theory and pertinent mathematical results;
4. outlook of exploring formal mathematical concepts and results using simulation.

This course places equal emphasis on conceptual understanding as well as hands-on explorations using computation. See also Notes on Pedagogy below.

Syllabus.

1. Randomness. Qualitative discussion about ubiquity of randomness and uncertainty in the real-world. Relationship between randomness, uncertainty, ignorance, pattern, predictability, complexity, lack of apparent pattern, etc. Randomness as a modeling assumption. Probability as a quantifier of uncertainty and randomness. Probability theory as mathematical model for phenomena involving randomness and uncertainty. Pseudo-random-number generators.

   Laboratory suggestions. Computational exploration of the statement “(frequentist) probability of an event is the long-run ($n \to \infty$) limit of the proportion of the number of times the event of interest occurred in n trials”.

2. Elements of probability. Modeling experiments whose outcomes are uncertain. Experiment, sample space and events. Sets, Venn diagrams and the algebra of events. Axioms of probability. Sample spaces having equally likely

outcomes. Conditional probability. Bayes' formula/theorem. Probabilistic independence, independent events, information aspect of independence.

Laboratory suggestions. Simulations of coin tossing, dice throwing, etc. and probability as emergent from their long-run behaviour. Computational exploration of $P(A|B)$ as the (long-run) proportion of times that A occurred amongst those when B occurred.

3. Random variables and expectation. Random variables (RV). RVs as measured quantities related to the experiment whose outcomes are uncertain. Types of RV. Jointly distributed RVs. Independent RVs. Indicator RVs. Conditional distributions. Pair of RVs; bivariate joint distributions (PDF, PMF, CDF); conditional distributions; independence. Qualitative discussion about transformations of RVs and how they transform the distribution.

   Laboratory suggestions. How do we understand the claim that a (pseudo-)random-number generator generates numbers which follow the claimed distribution? Summarizing a heap of numbers using commonly-used descriptive statistics such as mean, variance, standard deviation, median, quantiles, etc. Visualizing a heap of numbers using common devices such as boxplots and histograms. Qualitative discussion: boxplot as a 5-number summary, outliers, histogram as piecewise flat estimator of probability density and the importance of using an optimal bin width (such as Freedman-Diaconis), EDF as estimator of the CDF, quantile-quantile plots, etc. Bivariate scatter as indicative of the structure of the underlying bivariate PDF. How transformation of a RV transforms the PDF.

4. Expectation. Definition and meaning. Properties of the expected value. Expected value of sums of random variables. Probability as expectation of an indicator random variable. Variance: Definition and meaning. Covariance and variance of sums of random variables. Correlation. Covariance and correlation as measuring the extent of linear dependence between two random variables. Zero correlation does not imply independence; e.g., $Y = X^2$ with $X \sim N(0, 1)$ where there is a clear but nonlinear dependence but the covariance can be shown to be 0. Moment and moment generating function. Chebyshev's inequality and the weak law of large numbers (WLLN). Conceptual understanding of WLLN as convergence in probability.

   Laboratory suggestions. Bivariate scatters of different shapes and their connection with correlation. Computational exploration of the WLLN: Concentration behaviour as seen through boxplots. Exploring the full mathematical structure of the WLLN through simulation: Estimating probability as function of n and ϵ, representing it as a table and visualizing it as a heatmap. Qualitative discussion of concentration behaviour as indicative of convergence in probability.

5. Special random variables. The Bernoulli and binomial random variables. The Geometric random variable. The Poisson random variable. The hypergeometric random variable. The uniform random variable. Normal random variables. Exponential random variables. Distributions arising from the normal. The χ^2 distribution. The t-distribution. The F-distribution. The logistic distribution.

   Laboratory suggestions. Distribution-related functions in `R`, `python`, etc.

   Seminar suggestion. Students (individually or in groups) may be asked to prepare and present the sections about individual distributions.

6. Distributions of sampling statistics Introduction. The sample mean $\overline{X}_n$. The central limit theorem (CLT). Approximate distribution of the sample mean. How large a sample is needed?. The sample variance S_n^2. Sampling distributions from a normal population. Distribution of the sample mean. Joint distribution of $\overline{X}_n$ and S_n^2. Sampling from a finite population.

   Laboratory suggestions. Computational exploration of the CLT. Qualitative discussion of the CLT behaviour as indicative of convergence in distribution.

Suggested Texts/References.

1. Sheldon M. Ross, *Introduction to Probability and Statistics for Engineers and Scientists*. Academic Press, 2020.
2. Morris deGroot and Mark Schervish, *Probability and Statistics*. Addison-Wesley, 2012.
3. Larry Wasserman, *All of Statistics*. Springer-Verlag, 2004 (Part 1 of the book).

Notes on Pedagogy. Computing platforms such as `R`, `Python`, `julia` or similar are to be used liberally to illustrate (by the instructor) and explore (by the student) probability-related concepts and results such as the weak law of large numbers and the central limit theorem. The syllabus above is based on chapters 3-6 of Ross (3). However, the pedagogic approach of putting equal emphasis conceptual rigour (considered distinct from mathematical rigour) and on being able to put the conceptual machinery in practice via computational exploration is indigenous. deGroot and Schervish (2) is an excellent resource accessible to the students. Wasserman (3) Part I, though terse for most students, provides interesting insights and examples; it should be consulted by the instructor.

Contributor/s. Mihir Arjunwadkar (`https://scms.unipune.ac.in/~mihir`)

4.1.4 MNS-504-MJP: Fundamentals of Computing

Credits: 4 **Category:** Laboratory **Prerequisites.** CP

Course Outcomes

This course is intended to be an introduction to computing and algorithms for a non-computer-science graduate student. In principle, any programming language (C, `Python`, `Haskel`, `LISP`, etc.) can be used for illustrating algorithms, at the discretion of the instructor. This course is intended for

1. creating awareness about computing as a problem-solving approach;
2. developing an understanding of finite precision arithmetic;
3. developing an understanding of algorithm development and analysis; and
4. introducing basic types of algorithms used for problem solving.

Syllabus.

1. Digital computer fundamentals. Introduction to Linux Operating System. (Optional) simple overview of Bash scripts.
2. Installation. How to install the interpreter/IDE or compiler (depending upon the choice of programming language). In case of interpreter, an introduction to console environment. Simple 'Hello-world' program.
3. Variables and Expression. Various types of variables available in the choice of programming language. For example in `R`, Strings, Boolean etc. Expressions like `if-then-else` etc.
4. Loops. Introductions to various types of looping in the language.
5. Functions. Writing and calling `functions/subroutines`. Generous time is expected to be spent on mastering the technique of writing functions.
6. I/O Various ways of getting the input from the user and writing (formatted) output. This includes command-line I/O as well as file I/O.
7. Advanced topic Depending on the choice of language more advance topic like in case of Python `sets, dictionaries, tuple` or in case of C `pointers`
8. Plotting. Generation of simple plots using the data generated by the program. In case of languages like C, one may use external tools like `Gnuplot`.

Suggested Texts/References.

1. V. Rajaraman, T. Radhakrishnan, *An Introduction to Digital Computer Design*. PHI, 2007.
2. T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, *Introduction to Algorithms*. PHI Learning, 2009.
3. D. E. Knuth, *The Art of Computer Programming, Vol. 1*. Addison Wesley, 2011.
4. A. V. Aho, J. E. Hopcroft, J. D. Ullman, *Design and Analysis of Algorithms*. Pearson Education, 2011.
5. E. Horowitz, S. Sahni, *Fundamentals of Computer Algorithms*. Universities Press, 2008.

Notes on Pedagogy.

1. The course if expected to be hands-on with significant attention to improve the programming abilities by numerous examples rather than simply covering the topic.
2. Instructor can choose the programming language based on the need of the time, and appropriate tweaking of the syllabus is expected based on the composition of the class.

Contributor/s.

1. Snehal Shekatkar (`http://inferred.in/`)
2. Bhalchandra Pujari (`http://scms.unipune.ac.in/~bspujari`)
3. Ankita Katre (`https://ankitamkatre.wixsite.com/my-research`)

4.1.5 MNS-505-MJP: Research Methodology in Modeling & Simulation

Credits: 4 **Category:** Laboratory **Prerequisites.** CP

Course Outcomes

Mathematical Modeling is the ability to observe a situation around us, consider its constituent elements, describe it in words and then in symbols, such that we can bring to bear the symbolic techniques of mathematics to improve our understanding the situation. Why is mathematical modeling valuable? The answer is: given a model, we can predict real life consequences should circumstances or situations change. The changing part of our model is a parameter of the model; be it time, energy, population, force, or what you may. The model then acts as a super-calculator, enabling us to explore possible futures.

The end result of courses in mathematical modeling should be students who can clarify and simplify situations and extract the essential nugget that needs to be modeled. Specifically, they should be able answer questions such as: what aspect/s of the reality does the model capture? What aspect/s does it *not* capture? What questions can the model answer and what questions can it not answer? Why? Is the model capable of simulation? What questions need a simulation using the model to be answered? Are they different from questions that the model can answer without simulation? If so, in what way?

They then need to know how to look for, learn, and deploy the mathematical "technology" which is most suitable for their needs. While mathematical "technology" can be taught in conventional mathematics courses, the core sequence of mathematical modeling courses in this programme tries to teach the *style of thinking* that allows a modeler translate real-world scenarios to mathematics.

See also: sequels MNS-554-MJP and MNS-604-RP.

Syllabus. The following is not intended as a fixed course outline to be followed rigidly, but as suggestive of what should be covered in this course.

1. Historical context with Indian Knowledge System, as illustrating mathematical modeling through the works of Eratosthenes, Aryabhatta, Bhaskaracharya and Mercator for astronomy and cartography; Galileo and Newton for astron-

omy and mechanics, Euler and Fourier for dynamic phenomena; etc.

2. A selection of mathematical modeling examples from various problem domains, such as biology, chemistry, economics, physics, psychology, sociology, etc.
3. Modeling change. Difference equations and ordinary differential equations as models of change: examples; elementary analytical methods for solving ODE; qualitative analysis of ODE and visualization; software tools for analysis, solution, and visualization.

Suggested Texts/References.

1. William P. Fox, Frank Giordano, and Steven Horton, *A First Course in Mathematical Modeling*. Brooks/Cole, 2013.
2. John H. Holland, *Emergence: From Chaos to Order*. Helix Books, 1998.
3. Stephen Wolfram, *A New Kind of Science*. Wolfram Media, Inc, 2002.
4. Roger Penrose, *The Road to Reality: A Complete Guide to the Laws of the Universe*. Jonathan Cape, London, 2004.

Notes on Pedagogy.

1. Teaching to translate a real-world scenarios into mathematics is a difficult task. In this course, we have hoped that these translation skills would develop, to some extent at least, by studying the works of past masters of the trade.
2. Through the examples discussed, an overview of the modeling process and principles should emerge, including: Simplification, abstraction, mathematization; geometric similarity and proportionality as models; principle of parsimony AKA Occam's razor; model validation and modeling life cycle.
3. Relating to the point 1 in the syllabus: The usual sort of school courses that mention these scientists focus on their solutions, while we focus on how they reached those solutions in the context of the problems they were trying to understand or solve. The power of mathematization can be brought out through seemingly simple ideas from precalculus that get us as far as the size of the solar system, the creation of calculus to understand orbits, the use of geometry and mechanics to understand design of gear profiles, the use differential equations in biology, etc. At the end of the course, students should find that the mathematical methods they have seen in the past are now anchored to phenomena that they have experienced. Mathematics is thus seen as a tool for solving problems rather than as a set of rituals to be followed.
4. Relating to the point 3 in the syllabus: This is the only core course which includes a discussion on differential equations in this curriculum. It is, therefore, important to cover this topic in this course.

Contributor/s.

1. Aamod Sane (https://www.flame.edu.in/about-flame/faculty/sane-aamod)
2. Bhalchandra Pujari (https://scms.unipune.ac.in/~bspujari)
3. Ankita Katre (https://ankitamkatre.wixsite.com/my-research)
4. Bhalchandra Gore (https://scms.unipune.ac.in/~bwgore)
5. Mihir Arjunwadkar (https://scms.unipune.ac.in/~mihir)
6. Abhijat Vichare (https://www.linkedin.com/pub/abhijat-vichare/2/822/828)
7. Snehal Shekatkar (http://inferred.co/)

4.1.6 MNS-506-MJ: Colloquium in M&S

Credits: 2 **Category:** Seminar **Prerequisites.** CP

Course Outcomes

- Familiarize students with recent trends and key literature in modeling and simulation.
- Improve technical communication through seminars and reports.
- Develop critical thinking and analytical reading of scientific papers.
- Encourage interdisciplinary integration through topic selection.

Syllabus. The Seminar course is designed to develop the student's ability to understand, synthesize, and communicate recent advances in modeling and simulation. It encourages engagement with contemporary research, fosters interdisciplinary awareness, and builds presentation and academic writing skills. There is no fixed syllabus.

Suggested weekly Structure

Week 1 Introduction and course guidelines. How to give a seminar.

Week 2–3 Topic selection and approval by faculty mentor.

Week 4–7 Literature review, outline preparation, one-on-one guidance.

Week 8 Draft presentation and peer feedback sessions.

Week 9–12 Final seminar preparation, presentations with Q&A.

Week 13–14 Written report submission and instructor feedback.

Suggested Seminar Topics Please refer to Table 4.1 for example topics. These topics are illustrative and students are highly encouraged to go beyond these topics and investigate the current trends. The relevant topics from Indian Knowledge Systems are also welcomed.

Suggested Texts/References. No prescribed books. But latest literature review of good quality, peer-review articles is expected.

Topic	Modeling Focus
Agent-based modeling in epidemiology	Simulating disease spread using individual agents and stochastic interactions
Monte Carlo simulations in finance and physics	Probabilistic modeling and numerical integration of complex systems
Tumor growth modeling	Differential equation models of cancer cell proliferation and treatment response
ODE models of biological networks	Systems of ordinary differential equations to describe biological interactions
Machine learning in simulation-based inference	Surrogate modeling, emulation, and Bayesian inference techniques
Finite element methods in structural analysis	Numerical modeling of stress, strain, and mechanical behavior
Ecological system modeling	Predator-prey dynamics, population modeling, and resource flow
Multiscale modeling in materials science	Linking atomistic to continuum scales using hybrid simulations
Mathematical Models in Ancient Indian Astronomy (Jyotisha)	Epicyclic modeling of planetary motion, solar-lunar cycles
Ayurvedic Modeling of Body Systems (Tridosha Theory)	Abstract dynamical system representing physiological states
Water Management Systems in Ancient India	Simulation of flow and storage dynamics in traditional irrigation systems
Ecological Knowledge in Indian Agriculture	Systems modeling of crop planning, monsoon dependency, intercropping
Mathematical Structures in Indian Classical Music (Tala System)	Algorithmic modeling and simulation of rhythmic cycles

Table 4.1 Seminar Topics with Relevance to Modeling and Simulation (Including Indian Knowledge System)

Notes on Pedagogy. This course adopts a student-centered and interdisciplinary pedagogical approach. Students are guided through a structured yet flexible process involving topic selection, literature review, seminar preparation, and report writing. Emphasis is placed on fostering independent exploration while ensuring regular faculty mentorship. Peer interaction is encouraged through discussions, feedback sessions, and active seminar participation. Attention is given to developing both oral and written scientific communication skills, with guidance on presentation design and academic writing practices. The course promotes critical thinking, respectful engagement with diverse knowledge systems including Indian Knowledge Systems (IKS), and the effective use of digital tools.

Contributor/s.

1. Bhalchandra Pujari (`https://scms.unipune.ac.in/~bspujari`)

4.2 Semester 2

4.2.1 MNS-551-MJP: Numerical Computing 1

Credits: 2 **Category:** Laboratory **Prerequisites.** MNS-501-MJ, MNS-502-MJ, MNS-504-MJP

Course Outcomes

Many modeling formalisms lead to situations that involve numerical computing. By *numerical computing*, we mean numerical analysis and numerical mathematics with a strong hands-on computing component. This course and its sister course MNS-601-MJP (Sec. 4.3.1) are intended to cover topics of practical importance that are not covered elsewhere in the curriculum. Specific objectives of this course include the following:

1. To develop ability to understand and implement numerical algorithms.
2. To be able to make an informed choice of an appropriate numerical method to solve a given problem.
3. To be able to estimate rounding error, run time, memory and other computational requirements.

Syllabus.

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/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.

Notes on Pedagogy. This is not intended to be a course on formal numerical analysis per se. The hands-on computing component needs to be emphasized, a point-of-view that is consistent with the “concept-over-rigour” viewpoint that is at the heart of this programme. Exercises should involve a mix of paper-and-pencil and computing exercises using any programming language (e.g., C together with GSL) or computing environment that students are familiar with (e.g., matlab/scilab, python, R, etc.). Modeling contexts in which these numerical methods find their way are left to the discretion of the (expert) instructor.

Contributor/s.

1. Ankita Katre (https://ankitamkatre.wixsite.com/my-research)
2. Bhalchandra Gore (https://scms.unipune.ac.in/~bwgore)
3. Vaishali Shah (https://www.researchgate.net/profile/Vaishali_Shah10)

4.2.2 MNS-552-MJ: Optimization 1

Credits: 2 **Category:** Theory **Prerequisites.** MNS-501-MJ, MNS-502-MJ, MNS-504-MJP

Course Outcomes

This course is intended to build a foundation for deterministic optimization methods. The course is based on the use of key concepts in linear algebra and calculus to develop understanding of optimization methods. By completing this course, the student will be able to

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

Syllabus.

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, isosurfaces, 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.

Suggested Texts/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. R. L. Burden and J. D. Faires, *Numerical Analysis*. Brooks Cole, 2004.

Notes on Pedagogy. Hands-on work using either C+GSL or through computing platforms such as MATLAB/Scilab/Python/R would be beneficial for the students to understand intricacies of optimization problems.

Contributor/s.

1. Bhalchandra Gore (https://scms.unipune.ac.in/~bwgore)
2. Vaishali Shah (https://www.researchgate.net/profile/Vaishali_Shah10)

4.2.3 MNS-553-MJ: Statistical Inference

Credits: 4 **Category:** Theory **Prerequisites.** MNS-503-MJ, MNS-504-MJP

Suggested credit distribution: Theory: 2, Lab: 2

Course Outcomes

Statistical inference is a formalism for reasoning under uncertainty. It is crucial for modeling noisy data, analyzing it, and making inferences from it. In an age where almost every human endeavour is getting data-rich, knowledge of the basics of statistical inference will give an edge to the student. This course, in particular, focuses on the fundamentals of frequentist statistical inference. At the end of the course, the student is expected to have developed

1. solid conceptual understanding of the fundamentals of statistical inference;
2. ability to understand and illustrate formal concepts using simulation;
3. ability to apply them conceptually and computationally.

Syllabus.

1. Statistical models, inference and learning. Examples of questions that culminate into inference problems. Statistical Inference as reasoning under uncertainty. Statistical inference as inverse of probability theory. Modeling data as random variables. Parametric and nonparametric statistical models. Population and sample. Data categories and levels of measurement.

   Laboratory suggestions. Inspecting, understanding, visualizing and summarizing data. Mean, sd, var, median, MAD, IQR, quantiles, correlation, etc. Outliers. Visualization: Scatterplot, boxplot, QQ plot, histogram with optimal bin width, empirical distribution function, etc. Why visualization is important: Anscombe, Datasaurus Dozen, Iris, etc.

2. Point estimation. The frequentist viewpoint that the (unknown) parameters have well-defined, fixed values. Estimation as making the best guess under uncertainty. Estimation, estimate, estimator. Estimator as a random variable. Sampling distribution. Common examples of estimators, such as: mean (sample mean), median (sample median), standard deviation (standard error), IQR, MAD, etc. Assessing the quality of an estimator: Bias, variance, mean squared error (MSE). The bias-variance trade-off (BVT). Consistency and

(asymptotic) normality of an estimator.

Laboratory suggestions. Computational explorations of the BVT: (a) in histogram as estimator of the (unknown) PDF; and (b) in regressogram as estimator of the (unknown) regression function. Understanding consistency and (asymptotic) normality through computation/simulation.

3. Interval estimation. Definition of confidence interval (CI) and related terminology (coverage, “trapping the true value”, etc.). Interpretations of CIs: (a) interpretation based on replication of data-generating experiment, and (b) interpretation that does not require replication of data-generating experiment. CI as a probability statement about uncertainty in the location of the true but unknown parameter. Examples of analytical CIs: CI for normal mean; CI for the Bernoulli p parameter based on Hoeffding inequality; CI for normal mean when the variance is unknown; CI for the variance of a normal distribution; CIs based on approximate or asymptotic normality; etc. Prediction and prediction intervals.

   Laboratory suggestions. The frequency interpretation of a CI based on replication of the data-generating experiment. Comparison of coverage of normal-based and Hoeffding-based CIs for Bernoulli p.

4. Empirical distribution function (EDF), statistical functionals, bootstrap. The EDF, properties, confidence band, etc. EDF as a nonparametric piecewise flat estimator for the CDF that puts equal weights on the data points. Functionals, statistical functionals, linear statistical functionals. Plug-in estimators of linear statistical functionals. Nonparametric bootstrap philosophy and its basis in the EDF. Bootstrap estimation of variance. Bootstrap confidence intervals: normal-based, pivotal and percentile.

   Laboratory suggestions. Estimation of variance using bootstrap: E.g., using the plug-in estimator of skewness. Computing CIs using bootstrap for, say, skewness, kurtosis, etc.

5. Statistical correlation. Qualitative understanding of the correlation (strong vs. weak, linear vs. nonlinear) via bivariate scatterplots. Pearson correlation ρ and its plug-in estimator. Relationship between ρ and the slope in linear regression. Why ρ measures only linear dependence. Qualitative understanding of homo- and hetero-skedasticity. Spearman rank correlation. Difference between correlation and causation.

   Laboratory suggestions. Generate bivariate data with different structures, visualize them as scatterplots, compute the two correlation coefficients, and interpret the correlation values. Use bootstrap to compute confidence intervals for correlation.

6. Parametric inference. Parametric statistical models, parameter of interest, nuisance parameter. Method of moments (MoM) and the large-sample properties of MoM estimators (consistency, asymptotic normality). Maximum likelihood (ML) estimation and and the large-sample properties of ML estimators (consistency, asymptotic normality, equivariance, optimality/efficiency, delta method). ML for multiparameter models and Fisher information. The parametric bootstrap.

   Laboratory suggestions. Simple 1- and 2-parameter examples illustrating the consistency and asymptotic normality of MoM estimators. Simple 1- and 2-

parameter examples illustrating the consistency, asymptotic normality, and efficiency of ML estimators. Illustration of parametric bootstrap.

7. Hypothesis testing. Hypothesis testing as the problem of trying to decide whether a parameter θ lies in one subset of the parameter space or in its complement. Conceptual machinery and terminology: Parameter space, null and alternate hypotheses, data/sample space, test statistic, critical value / rejection threshold, decision rule, critical / rejection region, type-I and type-II errors, power function, level/size, p-values, simple and composite hypotheses, one- and two-sided tests. The opposite behaviour of the two errors as function of the critical value, and balancing the two errors via the power function. Scientific significance versus statistical significance. Relationship/equivalence between hypothesis tests and confidence intervals/sets. Common tests; e.g., Wald test and its connection with confidence interval, Pearson's χ^2 test for multinomial data, permutation test, likelihood ratio test, goodness-of-fit tests, etc. A basic-level introduction to the multiple testing problem.

   Laboratory suggestions. Computational exploration of the opposite behaviour of the two errors for a pair of simple hypotheses. Explorations illustrating power functions for simple tests. Computational application / exploration of the tests discussed with reference to specific data. Exploration of finite- and large-sample properties of tests using artificial data via replication.

8. (Optional) Bayesian inference basics. Probability as a measure of degree of belief and the Bayesian philosophy. Bayesian inference as "learning": the prior as encoding the analyst's state of knowledge prior to observing data; the likelihood as encoding the analyst's view of the data-generating experiment; the posterior as encoding the analyst's updated state of knowledge. Flat, improper, and noninformative priors. Conjugate priors. The prior as regularization, role of prior in avoiding overfitting. Large-sample properties of Bayes procedures and their connection to frequentist likelihood-based inference. Strengths and weaknesses of Bayesian inference *vis a vis* the frequentist/classical inference, and the subjectivity-objectivity debate.

   Laboratory suggestions. Plotting likelihood functions for simple one-parameter examples such as Bernoulli(p); plotting posterior distributions for the same likelihood and various priors; exploring how these posteriors change with the data size n.

Suggested Texts/References.

1. Irwin Miller & Marylees Miller, *John E. Freund's Mathematical Statistics with Applications*. Pearson Education, 2014.
2. Morris deGroot & Mark Schervish, *Probability and Statistics*. Addison-Wesley, 2002.
3. Sheldon M. Ross, *Introduction to Probability and Statistics for Engineers and Scientists*. Academic Press, 2020.
4. Larry Wasserman, *All of Statistics*. Springer-Verlag, 2004.

Notes on Pedagogy. The emphasis of the course should be on (a) understanding concepts well rather than on mathematical rigour, (b) being able to interpret formal results and visualize formal constructs, and (c) being able to apply these concepts and methods to real problems. That said, formal reasoning and analysis should be an integral part of the course wherever it helps understand or illustrate concepts better and develop insights. The course should also develop a perspective on real-life and simulated data modeling contexts where statistical inference plays a crucial role. Computing platforms such as `R`, `Python`, `julia` or similar are to be used liberally to illustrate (by the instructor) and explore (by the student) concepts, methods, and applications.

Contributor/s.

1. Mihir Arjunwadkar (`https://scms.unipune.ac.in/~mihir`)
2. Snehal Shekatkar (`https://inferred.in`)

4.2.4 MNS-554-MJP: Overview of Modeling & Simulation

Credits: 2 **Category:** Laboratory **Prerequisites.** MNS-505-MJP, MNS-504-MJP

Course Outcomes

In this course, we consider simulation as an enhancement of mathematical modeling. Classical modeling, as illustrated in the previous Research Methodology course MNS-505-MJP, has been mostly analytical; but with the emergence of computers, non-analytical, algorithmic, discretized or discrete, numerical, and computer algebra systems have come into their full power. This course thus focuses on agent-based simulation as perhaps the most general form of a computational model of a real-world phenomenon. See also: sister courses MNS-505-MJP and MNS-554-MJP.

Syllabus. The following is not intended as a fixed course outline to be followed rigidly, but as suggestive of what should be covered in this course.

1. Introduction to the agent-based paradigm.
2. Introduction to an agent-based software platform. E.g., NetLogo.
3. Agent-based M&S hands-on. Examples such as those discussed in the Rationale section above may be worked out in the agent paradigm using the agent-based platform introduced.

Suggested Texts/References.

1. John H. Holland, *Emergence: From Chaos to Order*. Helix Books, 1998.
2. Stephen Wolfram, *A New Kind of Science*. Wolfram Media, Inc, 2002.
3. Roger Penrose, *The Road to Reality: A Complete Guide to the Laws of the Universe*. Jonathan Cape, London, 2004.

Notes on Pedagogy. Agent systems require the modeler to think of the world we see as *emergent*, arising from local interactions rather than from a global, omniscient view that usual analytical models seem to prefer. In the previous course, students have seen that calculus is in fact made of "summations of infinite small effects", but the local style of thinking used by Newton, for instance, in his justification of Kepler's laws does not quite sink in because calculus generates a global view as its end result. With agent-based modeling, this distinction now sinks in in the minds of the students. We connect the two courses by asking students to solve very similar problems, but this time with agents rather than with equations. Other forms of simulation, like discrete-event simulations, Monte Carlo methods, etc., can be explicitly discussed if time allows, or can be programmed via the randomization facilities available in agent-based modeling systems like NetLogo.

Contributor/s.

1. Aamod Sane (https://www.flame.edu.in/about-flame/faculty/sane-aamod)
2. Bhalchandra Pujari (https://scms.unipune.ac.in/~bspujari)
3. Ankita Katre (https://ankitamkatre.wixsite.com/my-research)
4. Bhalchandra Gore (https://scms.unipune.ac.in/~bwgore)
5. Mihir Arjunwadkar (https://scms.unipune.ac.in/~mihir)
6. Abhijat Vichare (https://www.linkedin.com/pub/abhijat-vichare/2/822/828)
7. Snehal Shekatkar (http://inferred.co/)

4.2.5 MNS-555-OJT: On Job Training

Credits: 4 **Category:** Internship **Prerequisites.** All courses so far

Course Outcomes

By the end of the OJT, students will be able to:

- Apply modeling and simulation techniques to solve real-world problems in academic, industrial or governance/decision-support settings.
- Integrate theoretical knowledge with hands-on experience in computational tools and data analysis.
- Communicate technical results effectively through written reports or presentations.
- Demonstrate professional skills such as teamwork, problem-solving, and time management.

Syllabus. **No fixed syllabus.** On-the-job training (OJT) in the M.Tech Modeling and Simulation program is strategically designed to prepare students for both academic research and industry-oriented careers. Through structured internships/projects in research laboratories, academic institutes, or industrial settings, students engage directly with real-world modeling challenges. This practical exposure not only enhances their technical and analytical skills but also to foster their critical thinking, collaboration, and communication—key competencies for both scholarly work and professional environments. Although students are expected to work in external organizations, they may also choose to carry out literature survey of published research on a selected topic related to the current field in M&S. The internship should be carried out under a supervisor.

Suggested Texts/References. No prescribed texts.

Notes on Pedagogy. Evaluation for this course is to be done using a combination of

1. progress of the student as gauged by the external advisor/s and the internal mentor/s;
2. one or more mid-term presentation/s and/or viva voce;
3. a final presentation and/or a comprehensive poster; and

4. a final written report.

Contributor/s.

1. Bhalchandra Pujari (https://scms.unipune.ac.in/~bspujari)
2. Mihir Arjunwadkar (https://scms.unipune.ac.in/~mihir)

4.3 Semester 3

4.3.1 MNS-601-MJP: Numerical Computing 2

Credits: 2 **Category:** Laboratory **Prerequisites.** MNS-551-MJP

Course Outcomes

Many modeling formalisms lead to situations that involve numerical computing. By *numerical computing*, we mean numerical analysis and numerical mathematics with a strong hands-on computing component. This course and its sister course MNS-551-MJP are intended to cover topics of practical importance that are not covered elsewhere in the curriculum. Specific objectives of this course include the following:

1. To develop ability to understand and implement numerical algorithms.
2. To be able to make an informed choice of an appropriate numerical method to solve a given problem.
3. To be able to estimate rounding error, run time, memory and other computational requirements.

Syllabus.

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. (Optional) Introduction to predictor-corrector method.
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. Quick working introduction to BLAS/LAPACK, and interfaces in GSL, MATLAB/Scilab, etc.

Suggested Texts/References.

1. H. M. Antia, *Numerical Methods for Scientists and Engineers*. Hindusthan Book Agency, second edition, 2002.
2. Doron Levy, *Introduction to Numerical Analysis*. Unpublised, 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. D. V. Griffiths and I. M. Smith, *Numerical Methods for Engineers*. Chapman and Hall/CRC, second edition 2011.
5. Steven C. Chapra and Raymond P. Canale, *Numerical Methods for Engineers*. Tata McGraw-Hill, second edition 2000.
6. Curtis F. Gerald and Patrick O. Wheatley, *Applied Numerical Analysis*. Addison-Wesley, fifth edition 1998.
7. Kendall E. Atkinson, *An Introduction To Numerical Analysis*. Wiley India, second edition, 2008.

Notes on Pedagogy. This is not intended to be a course on formal numerical analysis per se; the hands-on, computing component needs to be emphasized slightly more, a point-of-view that is consistent with the "concept-over-rigour" viewpoint that is at the heart of this programme. Exercises should involve a mix of paper-and-pencil and computing exercises using any programming language (e.g., C) or computing environment (e.g., matlab/Scilab, R, etc.) that students are familiar with. Modeling contexts in which these numerical methods find their way are left to the discretion of the (expert) instructor. Coding numerical methods oneself helps most students understand these methods better.

Contributor/s.

1. Ankita Katre (https://ankitamkatre.wixsite.com/my-research)
2. Bhalchandra Gore (https://scms.unipune.ac.in/~bwgore)
3. Vaishali Shah (https://www.researchgate.net/profile/Vaishali_Shah10)

4.3.2 MNS-602-MJ: Optimization 2

Credits: 4 **Category:** Hybrid **Prerequisites.** MNS-503-MJ, MNS-504-MJP, MNS-552-MJ

Suggested credit distribution: Theory: 2, Lab: 1

Course Outcomes

Stochastic optimization methods are core to training machine learning models, especially when gradients are noisy, the parameter space is high-dimensional, or objective functions are non-convex. These methods are also crucial for hyperparameter tuning, policy search in reinforcement learning, and black-box optimization. This course introduces essential algorithms and ideas that directly support modeling and learning tasks in modern AI. A background in these methods should be an essential part of a modeler's toolkit.

Syllabus.

1. Random number generation and Monte Carlo simulation. Brief introduction to pseudo-random number generators. Sampling strategies and Monte Carlo integration in high dimensions.
2. Introduction to stochastic optimization. Problem formulation. Comparison with deterministic optimization. Principles, motivations, and application scenarios. Use cases in ML, control, and decision-making under uncertainty.
3. Random search algorithms. Theory and practical aspects of direct random search. Performance tradeoffs and hybrid strategies.
4. Simulated annealing. Thermodynamic analogy. Application to combinatorial problems (e.g., TSP). Cooling schedules and convergence behavior.
5. Genetic algorithms and evolutionary strategies. Encoding schemes, operators, selection, crossover, mutation. Applications in optimization and learning. Parameter tuning and hybrid methods.
6. Stochastic approximation methods. Robbins–Monro algorithms. Finite-difference and simultaneous perturbation approaches. Application in adaptive control and machine learning.
7. Recent applications in data science and machine learning. Stochastic Gradient Descent (SGD) and Variants. Hyperparameter optimization. Bayesian optimization overview. Connections to reinforcement learning and black-box optimization.

8. (Optional) Other bio-inspired and population-based methods. Particle swarm optimization, ant colony methods. Brief exposure to differential evolution and estimation of distribution algorithms.

Suggested Texts/References.

1. James C. Spall, *Stochastic Optimization*, in J. Gentle, W. Härdle, and Y. Mori, eds., *Handbook of Computational Statistics*. Springer, 2004.
2. James C. Spall, *Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control*. Wiley, 2003.
3. M. Michell, *An Introduction to Genetic Algorithms*. MIT Press, 1996.
4. P. J. M. Van Laarhoven and E. H. L. Aarts, *Simulated Annealing: Theory and Applications*. Kluwer Academic Publishers, 1987.
5. Ian Goodfellow, Yoshua Bengio, Aaron Courville, *Deep Learning*. MIT Press, 2016.

Notes on Pedagogy. The syllabus outline above is partially based on the first review article above. A qualified instructor experienced in stochastic optimization methods may alter the sequence or topics without altering the overall focus of the course. A detailed exposition of more stochastic optimization methods methods may be included at the instructor's discretion.

Contributor/s.

1. VK Jayaraman (https://scholar.google.co.in/citations?user=GRv1gLQAAAAJ&hl=en)
2. Mihir Arjunwadkar (https://scms.unipune.ac.in/~mihir)

4.3.3 MNS-603-MJ: MOOC course

Credits: 4 **Category:** Hybrid **Prerequisites.** Varies

Course Outcomes

MOOC stands for *Massive Open Online Courses* which are the courses offered by various reputed orgnizations. This course is designed to enable students to:

1. Identify and engage with contemporary topics in modeling and simulation through high-quality online resources.
2. Develop specialized knowledge in emerging areas.
3. Cultivate the ability to independently acquire, assess, and apply new tools and techniques relevant to their academic and research goals.
4. Bridge the gap between core M&S training and domain-specific advancements across science and engineering.

Syllabus. No prescribed syllabus.
Suggested mode of conduct:

- Students will **select a MOOC** from platforms such as Swayam, Coursera, edX, NPTEL, FutureLearn, MITx, or similar.
- The MOOC must be **relevant to Modeling and Simulation**, broadly interpreted. Examples include courses in:
  - Machine Learning / Deep Learning / AI for Scientific Modeling
  - Advanced Computational Fluid Dynamics (CFD)
  - Mathematical Modeling of Biological Systems
  - Operations Research / Optimization Methods
  - Scientific Computing / High-Performance Computing
  - Materials Modeling / Data-Driven Materials Science
  - Climate and Earth System Modeling
- Selection must be **approved by the faculty coordinator**.
- Students are expected to **complete the MOOC and submit a reflection report / certificate** .

- Faculty coordinator may choose to independently assess the students via suitable mechnanism.

Note 1: Students may take the MOOC course in during or before semester, including during vacations.
Note 2: In case of difficulty, students may choose to opt a course offered by other departments with due approvals.
Note 3: If deemed fit, the department may choose to offer an elective in place of the MOOC course.

Suggested Texts/References. No prescribed texts.

Notes on Pedagogy. *Role of the faculty coordinator*

- Guide the student in choosing an appropriate MOOC.
- Approve the course based on alignment with the M&S curriculum.
- Provide checkpoints or optional discussion sessions.
- Choose the mode of assessment.
- Review the final report and presentation.

Contributor/s.

1. Bhalchandra Pujari (`https://scms.unipune.ac.in/~bspujari`)

4.3.4 MNS-604-RP: M&S Project

Credits: 4 **Category:** Project **Prerequisites.** MNS-555-OJT

Course Outcomes

This last course in the M&S stream of courses tries to brings together the ideas learn in the previous courses MNS-505-MJP and MNS-555-OJT by asking students to solve a single problem in depth. They may at the same time be required to learn and use new techniques (e.g., game theory) as tools in addition to ones they use. This course lets them see the tools they have learnt in action, going beyond the week-long scale problems they have seen in prior courses, and instead asking them to solve 4-month long problems with a lot of self-driven work. It also serves as preparation for the industrial internship that are required to do during their degree.

This is an individualized course where a student is to work closely with a mentor. The emphasis of this course is on developing problem-solving, self-learning/self-study, and presentation (written and oral) skills. With the help of the mentor, the student should carry out hands-on work related to the broad focus of the programme, and culminating into a written report and a presentation. With their internship MNS-651-RP on the horizon, this course is intended to prepare students to develop appropriate skills including but not limited to literature search, resourcefulness, hands-on problem-solving related to a topic not studied before, etc. See pedagogic notes below.

Syllabus.

1. Hands-on M&S miniproject and/or internship preparation. Faculty mentor to decide the best strategy to achieve aims and objectives of this course for each student separately. For students who have decided their area of internship may be given M&S hands-on work appropriate for that area.
2. An overview of the modeling process. Simplification, abstraction, mathematization. Principle of parsimony AKA Occam's razor. Distinctions: Stochastic vs deterministic, linear vs nonlinear, static vs dynamic, etc. All models are false, some are useful: Modeling as an on-going process of knowledge refinement.

Suggested Texts/References. No prescribed texts. The mentor and the student can use any individually-chosen text or reference depending on the topic of study.

Notes on Pedagogy. The role of the faculty mentor is critical for the success of this course. If a student has already decided her/his place of internship MNS-651-RP, advisor, topic, etc., then the mentor should make sure, in collaboration with the organization or advisor for the internship, that (s)he spends her/his time in developing skills and background necessary for the internship. For other students, and until the details of their internship are not finalized, mentor should work on developing students' self-study, presentation (written and oral), and any other skills that may not be covered adequately elsewhere during the programme, or those in which the student may not be adequately trained. Assigning a modeling project can be one possible strategy to achieve the goals of this course. The mentor and the students need to meet on regular basis to ensure good and regular progress.

Contributor/s.

1. Snehal Shekatkar (http://inferred.co/)
2. Bhalchandra Pujari (https://scms.unipune.ac.in/~bwpujari)
3. Ankita Katre (https://ankitamkatre.wixsite.com/my-research)
4. Bhalchandra Gore (https://scms.unipune.ac.in/~bwgore)
5. Mihir Arjunwadkar (https://scms.unipune.ac.in/~mihir)
6. Aamod Sane (https://www.flame.edu.in/about-flame/faculty/sane-aamod)
7. Abhijat Vichare (https://www.linkedin.com/pub/abhijat-vichare/2/822/828)

4.4 Semester 4

4.4.1 MNS-651-RP: Dissertation

Credits: 22 **Category:** Internship **Prerequisites.** All

Course Outcomes

M. Tech. Dissertation is the pinnacle of the degree programme. The course is completed via an internship. The purpose of the internship is for the students to get in-depth exposure and experience in addressing challenging, real-life problems through M&S methods.

Syllabus. No fixed syllabus. Internship advisor/s and internal mentor/s at the Centre to decide the best strategy to achieve the aims and objectives of the internship for each student separately.

Suggested Texts/References. No prescribed texts.

Notes on Pedagogy. Internships are intended to be individual, and spanning a complete semester. In the best interest of the student, internships in settings external to the Centre (such as industry, research or academic institutes, NGOs, etc., depending on the interests of the student) are recommended, although possibilities of in-house internships are not ruled out. The internship topic/project may span any breadth of the M&S enterprise in any problem domain, from translating a domain-specific problem into an appropriate mathematical model, attempting to get analytical insights into the behaviour of the model, to exploring the behaviour of the model through computing/simulation. While the internship may have a substantial computing/coding component, it is not intended to be a pure software/coding project. The Centre's faculty committee is the supreme authority of all matters relating to the approval of internship topics/projects. The M&S context of the topic/project should be well-understood by the student, and should be brought out clearly in the report and presentations. The student, the external advisor/s, and the student's mentor/s at the Centre should ensure this, and have clarity as to where the internship fits into the M&S enterprise. Evaluation for this course is to be done using a combination of

1. progress of the student as gauged by the external advisor/s and the internal mentor/s;
2. one or more mid-term presentation/s and/or viva voce;
3. a final presentation and/or a comprehensive poster; and
4. a final written report.

Contributor/s.

1. Ankita Katre (https://ankitamkatre.wixsite.com/my-research)
2. Snehal Shekatkar (http://inferred.co/)
3. Bhalchandra Pujari (http://scms.unipune.ac.in/~bspujari)
4. Bhalchandra Gore (http://scms.unipune.ac.in/~bwgore)
5. Mihir Arjunwadkar (http://scms.unipune.ac.in/~mihir)

5

Chapter

Major Elective Credits

Choice-based electives listed below can be offered during any semester provided their prerequisites are satisfied. Prerequisites listed are suggestive/indicative; they can be decided/redefined by the course instructor with approval from the Departmental committee. Sufficient mastery over the content of the listed prerequisites, and over an appropriate programming language is assumed for all elective courses.

Students are required to complete elective courses totaling eight credits each in Semesters 2 and 3. Electives listed in Set 1 (Section 5.1) and Set 2 (Section 5.2) are designed as two-part course sequences, with one part offered per semester. While it is strongly recommended that students complete both parts of a sequence for a coherent learning experience, they may, in principle, opt for standalone electives from Set 3 (Section 5.3).

The offering of a course is contingent upon the availability of a qualified instructor and sufficient student interest. The final decision to offer any course rests with the department.

5.1 Elective Set 1

Table 5.1 Available courses under **Elective Set 1**

Course code	Course	Credits
MNS-560-MJ	Astrostatistics 1	4
MNS-561-MJ	Complex Networks 1	4
MNS-562-MJ	Computational Fluid Dynamics 1	4
MNS-563-MJ	Operations Research 1	4
MNS-564-MJ	Digital Signal and Image Processing 1	4
MNS-565-MJ	Artificial Intelligence 1	4

5.1.1 MNS-560-MJ: Astrostatistics 1

Credits: 4 **Category:** Hybrid **Prerequisites.** MNS-553-MJ,MNS-554-MJP

Course Outcomes

Outlook and rationale for this course can be summarized as follows:

> With data rates, sizes, and complexities soaring up high over the coming decades, meaningful investigation into a scientific question will require fresh ways of identifying patterns and structure in the data using sophisticated statistical and computational methodologies. (Indeed, the 21st century science has been aptly described as large datasets, complex questions science (Efron, 2011).) Technology development, which is essential for progress of science, also necessitates methodological development for its efficient and effective use. It is important to remember that it is the nature of the data and scientific questions being addressed which should dictate the method, and not vice versa (Arjunwadkar, Kashikar, and Bagchi, J. Astrophys. Astr. (2016) 37:28).

This course aims at introducing students with adequate background in statistics and computation – but not necessarily in astronomy and astrophysics (A&A) – to the field of astrostatistics. See the Notes on Pedagogy section of this syllabus document for more details.

Syllabus.

1. A crisp introduction to astronomy and astrophysics (outreach+ / undergraduate− level). Overview/survey of astronomical phenomena and their physics at a very basic level, essential terminology, and historical perspective as appropriate.
   (a) Basics of Astronomy: Observing the sky (naked eye, telescopes, and other instruments), measuring distances, distance scales, units, etc.
   (b) Basics of Astrophysics: blackbody radiation, stars, galaxies and their distribution, scaling relations.
   (c) Astronomical data: Nature of data-gathering and measurement processes together with discussion of the underlying statistical assumptions. Sources of errors in measurement and calibration.
2. Density estimation.

   (a) Histograms, bias-variance trade-off, optimal bin width, confidence bands.
   (b) Kernel Density Estimation: Univariate and multivariate, optimal bandwidth via cross-validation, confidence bands.
   (c) Adaptive smoothing: Adaptive kernel estimators, nearest-neighbour estimators.
   (d) Density estimation via normal mixtures: estimation, model selection, and bootstrap confidence intervals on parameters.
   (e) Nonparametric density estimation via orthogonal functions, and confidence sets.

3. Regression and model selection.

   (a) Generalities: Regression and regression function, quadratic prediction risk and $r(x) = \mathsf{E}(Y|X = x)$ as its minimizer, bias-variance decompositions.
   (b) Simple and multiple linear regression, least squares: unweighted and weighted, maximum likelihood, confidence intervals and bands, prediction bands, hat matrix, simple tests for coefficients, diagnostics, collinearity, transformations as a way to conform to assumptions better, linearity as linearity in the coefficients.
   (c) Variable/model selection: Training risk, Mallow's C_p, leave-one-out cross-validation, AIC, BIC, MDL, variable selection vs. hypothesis testing. An overview of regularization (ridge regression, LASSO and sparsity). Identifiability.
   (d) Nonparametric regression: Regressogram, running mean / local average smoother, linear smoothers, smoothing (hat) matrix. Choosing the smoothing parameter: Bias-variance trade-off, predictive risk and its estimators: training risk, leave-one-out-cross-validation risk, generalized cross-validation risk, etc. Kernel regression. Nonparametric regression via orthogonal functions, and confidence sets. Local polynomials. Regularization and splines.
   (e) Robust regression. Nonlinear regression. Handling (measurement) error in the covariates.

4. Clustering and classification. A crisp overview of supervised, unsupervised, and reinforcement learning, and a subset of topics in the following broad area: Generative and discriminative classifiers. Naive Bayes, Gaussian mixtures, generative adversarial classifiers, K-NN, logistic regression neural networks, decision tree and random forest classifiers. Deep neural networks. K-means, hierarchical and Gaussian mixtures for clustering. Time series classification and clustering. Hypothesis testing for classification. Attribute selection for classification.

Suggested Texts/References.

1. Fiegelson & Babu, *Modern Statistical Methods for Astronomy with R Applications*. Cambridge, 2012.
2. Wall & Jenkins, *Practical Statistics for Astronomers*. Cambridge, 2003.

3. AK Chattopadhyaya, *Statistical Methods for Astronomical Data Analysis*. Springer, 2014.
4. Starck & Murtagh, *Astronomical Image and Data Analysis*. Springer, 2006.
5. Mike Inglis, *Astrophysics is Easy! An Introduction for the Amateur Astronomer*. Springer, 2007.
6. Hastie, Tibshirani, & Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2009.
7. Efron & Hastie, *Computer Age Statistical Inference: Algorithms, Evidence, and Data Science*. Cambridge, 2016.

Notes on Pedagogy.

1. This course assumes that the audience have sufficient background in the statistics and computing, but none in A&A.
2. This course in its current form is organized around topics in statistics. Apt and appropriate applications and examples are to be chosen by the instructors from A&A.
3. Depending on the instructors' fields of expertise and the maturity level of the audience, this and/or a follow-up advanced course may also be run entirely in the problem-centric mode which weaves statistics around concrete astronomical problems and their associated datasets. In either mode, instantiations of this course should have a substantial hands-on component (textbook as well as non-textbook) analyzing actual astronomical data sets – from exploratory data analysis all the way upto trying to answer the science questions for which the data was collected.
4. Choice of programming language, computing environment, or software is left to the discretion of the instructors.
5. The above syllabus should be taken as indicative, and not as set in stone. Topics, subtopics, and finer content may be tweaked by the instructors to suite the audience's prior training and know-how. Except for #1 and #2 in the syllabus above, other topics may be presented/discussed in the class in any order. Given the highly multidisciplinary nature of the course, it may be tweaked to match the instructors' areas of expertise in statistics, A&A, astrostatistics, etc., and new developments in these fields.
6. Given the multidisciplinary nature of the course, ideally it should be interspersed with colloquia by astronomers, astrophysicists, statisticians, and data scientists. Such colloquia may be focused on case studies, specific applications, broad overviews – In short, anything that would help broaden the students' outlook.

Contributor/s.

1. Mihir Arjunwadkar (https://scms.unipune.ac.in/~mihir)
2. Akanksha Kashikar (https://www.researchgate.net/profile/Akanksha_Kashikar)

3. TV Ramanathan (https://www.researchgate.net/profile/T_Ramanathan)
4. VK Jayaraman (https://scholar.google.co.in/citations?user=GRv1gLQAAAAJ)
5. Kaustubh Vaghmare (https://www.researchgate.net/profile/Kaustubh_Vaghmare)
6. Dhruba J Saikia (http://mutha.ncra.tifr.res.in/ncra/people/academic/ncra-faculty/djs)
7. Somak Raychaudhuri (https://www.researchgate.net/profile/Somak_Raychaudhury)
8. Dipanjan Mitra (http://www.ncra.tifr.res.in/ncra/people/academic/ncra-faculty/dmitra)

5.1.2 MNS-561-MJ: Complex Networks 1

Credits: 4 **Category:** Hybrid **Prerequisites.** All SEM-I courses

Course Outcomes

A large number of real-world systems like Facebook, air-transport, metabolic reactions inside a living cell, and the Internet can be modelled as networks. Also several phenomena like traffic jams, rumour spreading and genetic regulations can be modelled as processes on networks making it an indispensable tool to study 'complex systems'. The theory of complex networks builds on methods borrowed from computer science, physics, statistics, social sciences and many others, and so is highly interdisciplinary. In this first part of this course stream, the student will gain understanding of

1. the abstraction called networks or graphs;
2. networks as a paradigm for certain real-world systems and their applicability;
3. quantification and analysis of network data;
4. important computer algorithms used to analyze networks;
5. insights gained about the real-world systems using the network paradigm.

Syllabus.

1. Introduction to networks. Networks as mathematical abstractions, usefulness and inappropriateness of networks while modeling the real-world systems, Empirical networks: Technological, Social, Information, Biological.
2. Mathematics of networks.
   (a) Mathematical representation, Adjacency matrix.
   (b) Weighted, directed, bipartite, multilayer, temporal/dynamic networks.
   (c) Degree, walks and paths, independent paths, cut sets, Menger's theorem
   (d) Trees, planar graphs.
   (e) Components in undirected and directed graphs
   (f) (Optional) Directed acyclic graphs
3. Network quantification.
   (a) Centrality values and distributions: degree, eigenvector, katz, pagerank, closeness, betweenness.

   (b) Cliques and k-cores, k-components
   (c) Transitivity and the global clustering coefficient, local clustering coefficient
   (d) Signed edges and structural balance
   (e) Vertex similarity: structural vs regular
   (f) Degree distributions in undirected and directed networks
   (g) Homophily and assortative mixing

4. Network algorithms.

   (a) Basics of time complexity of algorithms.
   (b) Storing network data: adjacency matrix, adjacency list, edge list.
   (c) Algorithms for degree distributions, clustering coefficients.
   (d) Breadth-first search (BFS), Dijkstra's algorithm.
   (e) Calculation of betweenness centrality using BFS

5. The structure of real-world networks.

   (a) Qualitative discussion of the types of errors and their sources in network data
   (b) Components, shortest paths and the small-world effect
   (c) Degree distributions, Scale-free networks, detection of power-laws.
   (d) Clustering coefficients, assortative mixing

Suggested Texts/References.

1. Mark Newman, *Networks: An introduction*. Oxford University Press (New York), 2018 or latest.
2. Harry Crane, *Probabilistic Foundations of Statistical Network Analysis*. Chapman and Hall/CRC, 2018.
3. Albert-László Barabasi, *Network Science*. Cambridge University Press, 2016.
4. Guido Caldarelli, *Scale-Free Networks: Complex Webs in Nature and Technology*. Oxford University Press UK, 2013.
5. S.N. Dorogovtsev and J.F.F. Mendes, *Evolution of Networks: From Biological Nets to the Internet and WWW*. Oxford University Press (Oxford), 2013.

Notes on Pedagogy. Knowledge of at least one programming language (`Python`, `R`, `C/C++`, etc.) or computational environment (`Matlab/SciLab`, `Mathematica`, etc.) is required.

Contributor/s. Snehal Shekatkar (`https://inferred.co/`)

5.1.3 MNS-562-MJ: Computational Fluid Dynamics 1

Credits: 4 **Category:** Hybrid **Prerequisites.** MNS-501-MJ, MNS-502-MJ

Course Outcomes

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

Syllabus.

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 comoving 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/References.

1. T. J. Chung, *Computational Fluid Dynamics*. Cambridge University Press, 2002.

Notes on Pedagogy.

Contributor/s. Sukratu Barve (`https://scms.unipune.ac.in/~sukratu`)

5.1.4 MNS-563-MJ: Operations Research 1

Credits: 4 **Category:** Hybrid **Prerequisites.** MNS-552-MJ

Course Outcomes

1. To make students aware of the problem domain of operations research (OR).
2. To introduce various models and simulation methods in OR.
3. Introducing software tools used in OR practices.
4. To make students aware of how the methods are used to solve real life problems.

Syllabus.

1. Linear programming. Simplex algorithm and simplex method, artificial variable methods. Degeneracy, duality in linear programming, duality theorems, dual simplex method with justification. Integer linear programming problem: pure and mixed integer programming problem, Gomory's all integer programming method. Fractional cut method, all integer and mixed integer linear programming problem, branch-and-bound method. Sensitivity analysis.
2. Transportation and assignment problems. Balance and degeneracy in transportation and trans-shipment problems. Duality theory of testing optimality of solution in transportation and trans-shipment problems. Hungarian method of assignment. Maximization, prohibition and other variations of assignment problems.
3. Dynamic programming.
4. Networking models. Graph theory concepts such as spanning trees, and algorithms for shortest path, network flow, maximum flows, etc. Transportation, trans-shipment, and assignment problems as networking problems. Network scheduling by PERT/CPM technique. Resource analysis in network scheduling.

Suggested Texts/References.

1. Kambo, N. S., *Mathematical Programming Techniques*. Affiliated East-West Press, 2002.
2. Taha, H. A., *Operations research*. Pearson, 2014 (9ed).
3. Sierksma, *Linear and Integer Programming*. CRC Press, 2002.

4. Kantiswaroop, P. K., Gupta, M. M., *Operation Research: An Introduction to Management Science*. S. Chand & Co., 2014.
5. Sharma, J. K., *Operations Research Theory and Applications*. Macmillan Publisher India, 2013 (5ed).
6. N. S. Kambo, *Mathematical Programming Techniques*. Affiliated East-West Press, 2002.
7. Ahuja, Magneti, Orline, *Network flows: Theory, Algorithm, and Applications*. Pearson, 2014.
8. T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, *Introduction to Algorithms*. PHI Learning, 2009.

Notes on Pedagogy. Theory in this course should be complemented with computational exercises based on software tools such as Lingo, AMPL, Gurobi, CPLEX and/or toolboxes/packages in languages such as Python and R.

Contributor/s. Padma Pingle

5.1.5 MNS-564-MJ: Digital Signal and Image Processing 1

Credits: 4 **Category:** Hybrid **Prerequisites.** MNS-501-MJ, MNS-502-MJ, MNS-505-MJP

Course Outcomes

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.

Syllabus.

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/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.

Notes on Pedagogy. At the discretion of the instructor, `matlab/octave/scilab` can be used for computing. This has the advantage of letting a student focus more on the domain context rather than on programming.

Contributor/s. Bhalchandra Gore (https://scms.unipune.ac.in/~bwgore)

5.1.6 MNS-565-MJ: Artificial Intelligence 1

Credits: 4 **Category:** Hybrid **Prerequisites.** MNS-503-MJ,MNS-504-MJP,MNS-552-MJ

Course Outcomes

In the current era of data explosion, scientists and engineers are looking for methods to extract relevant information through automation. The field variously known as artificial intelligence (AI), neural networks (NN), machine learning (ML) and statistical learning (SL) offers algorithms to "learn" from the data and make predictions. This and the sister course MNS-615-MJ are intended to introduce students to the basics of ML/SL/NN/AI. The emphasis and conduct of these courses should be on making the student as independent as possible towards understanding and engaging with the ever-increasing set of ML/SL/NN/AI methods and their applications – without treating the methods as black boxes, and with the ability to interpret and diagnose their results. Open-source toolboxes/libraries in the R and Python domains are currently deemed ideal for both pedagogic and production purposes.

Syllabus.

1. Learning, statistical learning, machine learning. (ISL chapters 1 & 2)
2. Understanding data, population and sample, data categories and levels of measurement, exploratory data analysis. Bias-variance trade-off. Maximum likelihood method. Overfitting and underfitting. Model and variable/feature selection.
3. Unsupervised learning. Principle component analysis and dimension reduction. Clustering: k-means and hierarchical. (ISL chapter 10)
4. Regression generalities. Linear and multiple linear regression. Accuracy and other diagnostics of the fit. Categorical predictors. Subset/model selection. Regularization/shrinkage methods. Dimension reduction methods. Nonlinear regression using polynomials, step functions and some basis. (ISL chapters 3, 6 and 7)
5. Resampling methods: Bootstrap and cross-validation (ISL chapter 5)
6. Classification. Generalities, Bayes rule for binary and multiclass classification, "regression function" in the context of classification, taxonomy of classification methods based on the regression function. Logistic regression: The model, fitting the model, and fitted-model diagnostics. Linear and quadratic discriminant analysis: The model, fitting the model, and fitted-model diagnostics. Naive Bayes. (ISL chapter 4)

7. (Optional) Classification and regression using trees (ISL chapter 8). Problems with high-dimensions (ISL chapter 6). Spline regression (ISL chapter 7).

Suggested Texts/References.

1. (ISL) G. James, D. Witten, T. Hastie, R. Tibshirani, *An Introduction to Statistical Learning with Applications in* R. Springer, 202+. https://www.statlearning.com/ (python instead of R if suitable)

Notes on Pedagogy. Crowd-sourcing platforms for data science such as http://kaggle.com/ offer rich data sets and challenges for exercising ML/AI skills and expertise. Instructors may consider encouraging and guiding students to participate in such competitions. Adequate emphasis on hands-on work (e.g., the lab sections in the suggested text above) and the required programming skills needs to be maintained throughout the course. Students may be encouraged to build a "machine learning portfolio" through somewhat extended miniprojects during this and the sister course MNS-615-MJ, focusing on

1. understanding the domain context of the chosen data;
2. understanding and visualizing the chosen data;
3. formulating interesting questions that can be investigated using the chosen data;
4. applying a range of methods to arrive at answers to these questions;
5. presenting the results of the analysis together with diagnostics; and
6. assessing the merit of the methods applied with reference to the data and the questions.

Contributor/s. Mihir Arjunwadkar (http://scms.unipune.ac.in/~mihir) and Bhalchandra Pujari (http://scms.unipune.ac.in/~bspujari)

5.2 Elective Set 2

Table 5.2 Available courses under **Elective Set 2**

Course code	Course	Credits
MNS-610-MJ	Astrostatistics 2	4
MNS-611-MJ	Complex Networks 2	4
MNS-612-MJ	Computational Fluid Dynamics 2	4
MNS-613-MJ	Operations Research 2	4
MNS-614-MJ	Digital Signal and Image Processing 2	4
MNS-615-MJ	Artificial Intelligence 2	4

5.2.1 MNS-610-MJ: Astrostatistics 2

Credits: 4 **Category:** Hybrid **Prerequisites.** MNS-560-MJ (Sec. 5.1.1)

Course Outcomes

This course is a sequel to MNS-560-MJ (Sec. 5.1.1), and introduces a few more areas of statistics which are heavily used in statistical data analysis in Astronomy & Astrophysics.

Syllabus.

1. Outliers. A crisp overview of considerations, core concepts, methods, and practice, broadly along the lines of chapters 1 and 2 of #1 in Suggested Texts/References.
2. Missing data. A crisp overview of considerations, core concepts, methods, and practice, broadly along the lines of chapters 1 and 2 of #2 in Suggested Texts/References.
3. Time-series analysis. A crisp overview of considerations, core concepts, methods, and practice, broadly along the lines of chapters 1–6 of #3 in Suggested Texts/References.
4. Spatial statistics. A crisp overview of considerations, core concepts, methods, and practice, broadly along the lines of chapters 1–3 of #6 in Suggested Texts/References.
5. Bayesian inference. Review of axioms of probability theory and Bayes Theorem for conditional probabilities. The Bayesian inferential philosophy: Classical/frequentist vs. subjective probabilities. Bayes theorem and its inferential interpretation as flow of information from prior to posterior via likelihood. Standard examples, such as Bernoulli likelihood with Beta prior, normal likelihood with normal prior; etc. Conjugate priors. Prior distributions: Proper, improper, flat, noninformative. Role of priors in making ill-defined estimation/inference problems well-behaved: E.g., Non-identifiable Gaussian mixture models with too many components (and when data is sparse), etc. Relative importance of prior and likelihood with respect to data size. Posterior distribution as the prime inferential object. Exploring the posterior through simulation. Credible intervals. Bayesian testing. Bayesian model selection. Bayesian vs. classical: Strengths and weaknesses.

Suggested Texts/References.

1. Charu C. Aggarwal, *Outlier Analysis*. Springer, 2013.
2. Little & Rubin, *Statistical Analysis with Missing Data*. Wiley, 2002.
3. Brockwell & Davis, *Introduction to Time Series and Forecasting*. Springer, 2002.
4. Gelman et al., *Bayesian Data Analysis*. CRC Press, 2013.
5. Joseph B. Kadane, *Principles of Uncertainty*. CRC Press, 2011.
6. Cressie, *Statistics for Spatial Data*. Wiley, 1993.
7. Bivand, Pebesma, & Gómez-Rubio, *Applied Spatial Data Analysis With R*. Springer, 2008.

Notes on Pedagogy. See this section for the the prequel course MNS-560-MJ (Sec. 5.1.1).

Contributor/s.

1. Mihir Arjunwadkar (`https://scms.unipune.ac.in/~mihir`)
2. Akanksha Kashikar (`https://www.researchgate.net/profile/Akanksha_Kashikar`)
3. TV Ramanathan (`https://www.researchgate.net/profile/T_Ramanathan`)

5.2.2 MNS-611-MJ: Complex Networks 2

Credits: 4 **Category:** Hybrid **Prerequisites.** MNS-561-MJ

Course Outcomes

This sequel of MNS-561-MJ (Sec. 5.1.2) deals with several intermediate- to advanced-level topics in the theory of complex networks. Mastering the material presented here will prepare the students to handle several real-world problems involving networks. The course is expected to lead the student to understand

1. the concept of random graphs, and corresponding mathematical methods;
2. possible mechanisms behind the growth of real-world networks;
3. community structure in networks, and elementary methods to detect it;
4. simple spreading processes and percolation on networks.

Syllabus.

1. Random graphs
   (a) Concept of random graphs
   (b) Erdős-Rényi (ER) graph: degree distribution, clustering coefficient
   (c) Concept of the giant component, giant component and path lengths in the ER graph
   (d) (Optional) Small components in ER graph
   (e) The configuration model, excess degree distribution and the friendship paradox, clustering coefficient
   (f) Locally-tree-like nature of the configuration model, the number of second neighbours, the giant component
   (g) The small components in the configuration model, configuration model with power-law degree distribution, diameter
   (h) Stochastic block model, the small-world model
   (i) (Optional) Generating function methods
2. Models of network formation
   (a) Preferential attachment and Price's model, Barabási-Albert model (BA model), the first mover effect

   (b) Qualitative discussion of the extensions of the preferential attachment models
   (c) Vertex copying models, network optimization model

3. Community detection
   (a) Graph partitioning and graph Laplacian, The Kernighan-Lin algorithm, spectral partitioning
   (b) Community detection, modularity maximization
   (c) Kernighan-Lin-Newman algorithm, spectral modularity maximization, division into more than two groups
   (d) Overview of the Louvain algorithm, methods based on statistical inference and the Newman-Girvan algorithm
   (e) Measuring community detection algorithm performance
   (f) (Optional) Resolution limit for modularity maximization, degeneracy problem

4. Spreading processes and percolation on networks.
   (a) Diffusion on networks, random walks, cascading
   (b) Introduction to the edge and vertex percolations
   (c) Computer algorithms for network percolation
   (d) Computer simulations for vertex percolation on ER and BA networks, interpretations of the results

Suggested Texts/References.

1. Mark Newman, *Networks: An introduction*. Oxford University Press (New York), 2018 or latest.
2. Albert-László Barabasi, *Network Science*. Cambridge University Press, 2016.
3. Guido Caldarelli, *Scale-Free Networks: Complex Webs in Nature and Technology*. Oxford University Press UK, 2013.
4. S.N. Dorogovtsev and J.F.F. Mendes, *Evolution of Networks: From Biological Nets to the Internet and WWW*. Oxford University Press (Oxford), 2013.
5. Alain Barrat, Marc Barthélemy and Alessandro Vespignani, *Dynamical Processes on Complex Networks*. Cambridge University Press, 2012.

Notes on Pedagogy. Knowledge of at least one programming language (`Python`, `R`, `C/C++`, etc.) or computational environment (`Matlab/SciLab`, `Mathematica`, etc.) is required.

Contributor/s. Snehal Shekatkar (`https://inferred.co/`)

5.2.3 MNS-612-MJ: Computational Fluid Dynamics 2

Credits: 4 **Category:** Hybrid **Prerequisites.** MNS-562-MJ

Course Outcomes

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

Syllabus.

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/References.

1. T. J. Chung, *Computational Fluid Dynamics*. Cambridge University Press, 2002.

Notes on Pedagogy.

Contributor/s. Sukratu Barve (https://scms.unipune.ac.in/~sukratu)

5.2.4 MNS-613-MJ: Operations Research 2

Credits: 4 **Category:** Hybrid **Prerequisites.** MNS-563-MJ

Course Outcomes

1. To make students aware of the advanced techniques in Operations Research (OR).
2. Introducing various models & theories used in OR.
3. To make students able to apply their knowledge for solving real life problems in OR.

Syllabus.

1. Advanced linear programming. Advanced techniques: Revised Simplex Method, Simplex Method versus Revised Simplex Method, Bounded variable technique, Parametric Linear Programming, Linear Fractional Programming and their applications, Karmarkar algorithm.

   (Optional) Sequential problems. Basic terms used in sequencing, n-jobs two machines sequencing problems, processing two jobs through K machines sequencing problems.
2. Queuing theory. Markovian and Non-Markovian queuing models (i.e., $(M/M/1)$, $(M/M/s)$, $(M/E_k/1)$, $(M,G,1)$, steady-state probabilities and their characteristics, cost profit models of $(M/M/1)$ queuing systems. Simulation, event type simulation, simulation of a queuing system.
3. Applications of OR. Modeling and OR analysis of real-life problems such as inventory models, vehicle routing, job shop scheduling, production planning, facility location, etc.
4. (Optional) Game theory. Two-person Zero-sum games, Maximin-Minimax principle. Games without saddle point. Graphical solution of $2 \times n$ and $m \times 2$ games, Dominance property, Arithmetic methods for $n \times n$ games, General solution of $m \times n$ Rectangular games, Limitations and Extensions.

Suggested Texts/References.

1. Kambo, N. S., *Mathematical Programming Techniques*. Affiliated East-West Press, 2002.
2. Taha, H. A., *Operations research*. Pearson, 2014 (9ed).
3. Sierksma, *Linear and Integer Programming*. CRC Press, 2002.

4. Kantiswaroop, P. K., Gupta, M. M., *Operation Research: An Introduction to Management Science*. S. Chand & Co., 2014.
5. Sharma, J. K., *Operations Research Theory and Applications*. Macmillan Publisher India, 2013 (5ed).

Notes on Pedagogy. The focus of this course is on advanced OR techniques, applications to real-life problems, large scale optimization, and on learning new algorithms and solution methodologies.

Contributor/s. Padma Pingle

5.2.5 MNS-614-MJ: Digital Signal and Image Processing 2

Credits: 4 **Category:** Hybrid **Prerequisites.** MNS-564-MJ

Course Outcomes

See purpose/outlook/rationale/goals for the prequel MNS-564-MJ. 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.

Syllabus.

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. nonlinear 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/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.

Notes on Pedagogy. At the discretion of the instructor, the computer vision library `OpenCV` can be introduced and used profusely in addition to `matlab/octave/scilab` to ease the programming effort, allowing a student to focus on the formalism and the domain context.

Contributor/s. Bhalchandra Gore (https://scms.unipune.ac.in/~bwgore)

5.2.6 MNS-615-MJ: Artificial Intelligence 2

Credits: 4 **Category:** Hybrid **Prerequisites.** MNS-565-MJ

Course Outcomes

In the current era of data explosion, scientists and engineers are looking for methods to extract relevant information through automation. The field variously known as artificial intelligence (AI), neural networks (NN), machine learning (ML) and statistical learning (SL) offers algorithms to "learn" from the data and make predictions. This and the sister course MNS-565-MJ are intended to introduce students to the basics of ML/SL/NN/AI. The emphasis and conduct of these courses should be on making the student as independent as possible towards understanding and engaging with the ever-increasing set of ML/SL/NN/AI methods and their applications – without treating the methods as black boxes, and with the ability to interpret and diagnose their results. Open-source toolboxes/libraries in the R and Python domains are currently deemed ideal for both pedagogic and production purposes.

Syllabus.

1. A detailed hands-on introduction to deep learning using neural networks (NN) (ISL chapter 10). Perceptron. Single-layer and multi-layer feed-forward NNs. Convolutional NNs and data augmentation. Featurization and encoding. Sequential data and recurrent NNs. When to use deep learning? When not to? Fitting a neural network: Back-propagation, regularization, stochastic gradient, dropout learning, double descent, etc.
2. Broad overviews of current topics such as: Neural networks as universal approximators; Neural architecture search (NAS); Natural-language processing (NLP); Generative AI; Encoders, transformers, attention and large language models (LLM); etc.

Suggested Texts/References.

1. (ISL) G. James, D. Witten, T. Hastie, R. Tibshirani, *An Introduction to Statistical Learning with Applications in* R. Springer, 202+. https://www.statlearning.com/ (python instead of R if suitable)

Notes on Pedagogy. Crowd-sourcing platforms for data science such as http://kaggle.com/ offer rich data sets and challenges for exercising ML/AI skills and expertise. Instructors may consider encouraging and guiding students to participate in such competitions. Adequate emphasis on hands-on work (e.g., the lab sections in the suggested text above) and the required programming skills needs to be maintained throughout the course. Students may be encouraged to build a "machine learning portfolio" through somewhat extended miniprojects during this and the sister course MNS-565-MJ, focusing on

1. understanding the domain context of the chosen data;
2. understanding and visualizing the chosen data;
3. formulating interesting questions that can be investigated using the chosen data;
4. applying a range of methods to arrive at answers to these questions;
5. presenting the results of the analysis together with diagnostics; and
6. assessing the merit of the methods applied with reference to the data and the questions.

Contributor/s.

1. Mihir Arjunwadkar (`http://scms.unipune.ac.in/~mihir`)
2. Bhalchandra Pujari (`http://scms.unipune.ac.in/~bspujari`)

5.3 Elective Set 3

Table 5.3 Available courses under **Elective Set 3**

Course code	Course	Credits
MNS-570-MJ	Concurrent Computing	4
MNS-571-MJ	High-Performance Computing	4
MNS-572-MJ	Theory of Computation	4
MNS-573-MJ	Functional Programming	2
22-E005	Computing with Java	2
MNS-575-MJP	Advance Python programming	2
MNS-576-MJ	Computing with R	2
MNS-577-MJP	Computing with MATLAB/Scilab	2
MNS-578-MJP	Computing with C	2
MNS-579-MJP	Statistical Models and Methods	4
MNS-581-MJ	Stochastic Simulation	2
MNS-582-MJ	Data Visualization	2
MNS-583-MJ	Difference Equations	2
MNS-584-MJ	Ordinary Differential Equations	2
MNS-585-MJ	Partial Differential Equations	2
MNS-586-MJ	Transforms	2
MNS-588-MJP	Advanced Statistics Laboratory 1	2
MNS-623-MJP	Advanced Statistics Laboratory 2	2
MNS-620-MJP	Computational Fluid Dynamics Laboratory	2
MNS-621-MJP	Machine Learning Laboratory	2
MNS-622-MJ	Advanced Data Analysis	2

5.3.1 MNS-570-MJ: Concurrent Computing

Credits: 4 **Category:** Hybrid **Prerequisites.** MNS-504-MJP

Course Outcomes

Concurrency is ubiquitous, and not only a part of OS courses. It can also be a program modularization technique whereby applications can be organized as a set of interacting concurrent components. The word “concurrency” not only alludes to “occurring at the same time”, but also has other connotations like “agreeing on some thing”, or “coming together for a task” etc. While concurrency in the first sense is often conflated with parallel computing, concurrency is broader and more pervasive. This course aims to bring out the basic issues and techniques needed to deal with concurrency.

1. Understanding of concurrency in parallel and distributed computing
2. Designing and modularising programs using concurrent tasks
3. Synchronizing and communicating between tasks in a concurrent program
4. (Optional) Formal modeling of concurrent systems

Syllabus.

1. Introduction. Introducing concurrency with simple problems like Readers-Writers, Producer-Consumer, Bounded Buffer, etc., as illustrations. Challenges in concurrency: synchronization, mutual exclusion, deadlock, livelock, starvation, non-determinism. Distinction between concurrent and distributed systems. Relation between timing and concurrency. Concurrency in algorithms and physical concurrency. Example of concurrent systems: operating systems, database systems, web servers.
2. Types of concurrency. (a) Program level – Introduction to the variety of control flows: sequential, coroutines etc. Program execution approaches: single tasking, multitasking, multiprocessing, multicomputing and distributed. (b) Data level – data structures and their concurrency, parallel computing.
3. Inter-process communication. Communication between components in a concurrent program: shared memory, message passing. Communication/-Data exchange/Interaction between processes on: time shared systems, client-server systems, distributed systems. Components of a concurrent program versus interaction between processes; similarities and differences.

   Techniques of handling concurrency. Locking, Time stamp ordering, Semaphores, Monitors. Goals: Correctness or fault tolerance.

4. Distributed systems. timing and clock synchronization, time stamps, Chandy-Lamport time ordering. Global snapshot on distributed systems. Examples: from distributed snapshots, Synchronous and asynchronous systems.
5. (Sample: See note 1). Introduction to Android as a programming platform. The basic "Hello World" application on Android. Implement algorithms on Android – three sort algorithms as independent programs. IPC on Android – A pair of sort algorithms on shared data set (Scenarios: e.g., (a) One works in place, and the other on a copy – both run concurrently, (b) Both work in place and concurrently, etc. Choose a scenario).
6. (Optional) See note 2. Abstract representation of concurrent (esp. distributed) systems: Introduction to a Process Algebra, e.g., CCS. The Calculus of Communicating Systems: Syntax of the CCS, Operational semantics of the CCS (non-determinism etc.). Worked examples for the CCS.

Suggested Texts/References.

1. Clay Breshears, *The Art of Concurrency: A Thread Monkey's Guide to Writing Parallel Applications*. O'Reilly, 2009.
2. Robin Milner, *Communicating and Mobile Systems: The π Calculus*. Cambridge University Press, 1999.
3. Michel Raynal, *Concurrent Programming: Algorithms, Principles, and Foundations*. Springer, 2013.
4. A. Roscoe, *Theory and Practice of Concurrency*. Prentice Hall, 1997.
5. G. Blake Meike, *Programming Android*. Shroff, 2012.
6. Reto Meier, *Professional Android 4 Application Development (Wrox)*. Wiley, 2012.

Notes on Pedagogy.

1. This course requires programming exercises. This syllabus illustrates the concepts discussed using Android. However, any suitable system could be used based on the familiarity of the students and the instructor. For example, a Unix/Linux based system would bring out shared memory (`shm*`) system calls, message passing (`msg*`) system calls, and possibly simple socket based IPC programming. Yet another possibility is to use MPI based programming assignments with similar detailing. This component is therefore labeled as "Sample" to suggest that the *means* used to bring out concurrency may be decided by the instructor depending on the circumstances.
2. The formal aspects are truly optional and at the discretion of the instructor. The main goal of including this treatment should be to illustrate the *mathematical modeling* component of the course. The suggested syllabus uses CCS – Calculus of Communicating Systems – by Robin Milner. However, any other useful system – e.g., Petri Nets, CSP (Communicating Sequential Processes – C.A.R. Hoare) etc. – can be used.

Contributor/s. Abhijat Vichare (https://www.linkedin.com/pub/abhijat-vichare/2/822/828)

5.3.2 MNS-571-MJ: High-Performance Computing

Credits: 4 **Category:** Hybrid **Prerequisites.** MNS-504-MJP

Course Outcomes

1. Understand the architecture and memory hierarchy of HPC systems.
2. Develop parallel programs using MPI for common numerical and sorting problems.
3. Design and analyze parallel algorithms using decomposition and mapping strategies.
4. Implement efficient communication operations across various topologies.
5. Apply parallel computing techniques to linear algebra problems such as matrix operations and solving systems of equations.

Syllabus.

1. HPC programming platforms. Implicit parallelism: trends in microprocessor architectures, memory system performance limitations, control structure of HPC platforms, communication model, physical organization, 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 inter-task 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. (Optional) 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.

Suggested Texts/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. Michael T. Heath, *Scientific Computing*. Tata McGraw-Hill, .
7. Michael Quinn, *Parallel Programming in C with MPI and OpenMP*. Tata McGraw-Hill, .

Notes on Pedagogy.

Contributor/s. Vaishali Shah (https://www.researchgate.net/profile/Vaishali_Shah10)

5.3.3 MNS-572-MJ: Theory of Computation

Credits: 4 **Category:** Hybrid **Prerequisites.** MNS-505-MJP

Course Outcomes

This is an introduction to the formal theory of computation and its modeling applications.

Syllabus.

1. Introduction to languages. Symbols, strings, words and languages. Symbolic dynamics, dynamics as language. Examples of languages. Finite representation of languages. String induction principles.
2. Finite automata. Functions as tables: introduction to theory of automata. Regular expressions and languages. Equivalence and simplification of regular expressions. Finite automata and labeled paths. Isomorphism of finite automata. Algorithms for checking acceptance and finding accepting paths. Simplification of finite automata. Proving the correctness of finite automata. Empty-string finite automata. Nondeterministic finite automata. Deterministic finite automata. Closure properties of regular languages. Transfer matrices and finite automata. Solving real-life problems with finite automata.
3. Context-free grammars. Examples of languages which are not regular. State minimization. The pumping lemma for regular languages. Context-free grammars, parse trees, stacks and queues. Dynamical systems generating context-free languages. Functions with "internal memory" and push-down automata. Isomorphism of grammars. Derivations, Converting between parse trees and derivations. Simplification and ambiguity of grammars. Determinism and parsing. Pumping lemma for context free grammars. Chomsky normal form. A parsing algorithm.
4. Turing machines. Examples which are not context free. Chomsky hierarchy. Another look at symbolic dynamics and coding theory for examples of dynamical systems in various levels of Chomsky hierarchy. Computing with dynamical systems. Functions with "external memory" and Turing machines. Computing with Turing machines. Extensions of Turing machines. Random access Turing machine. Non-deterministic Turing machines. Chaotic systems as Turing machines.
5. Universal Turing machines, complexity, computability. Universal Turing machines. Church-Turing thesis. Halting problem. Undecidable problems. Tiling problem and the Potts model. Computability and complexity theory. Classes P and NP. Cook's theorem and P-NP completeness theorems.

6. (Optional) Formal computation as a modeling paradigm. A judicious selection of topics such as: symbolic dynamics of dynamical systems, biological sequences and stochastic grammars, computational musicology, computational linguistics, natural language processing, L-systems, automata in game theory, etc.

Suggested Texts/References.

1. H. Lewis and C. Papadimitrion, *Elements of Theory of Computation*. Prentice-Hall, 1998.
2. V. E. Krishnamurthy, *Introductory Theory of Computer Science*. Springer-Verlag, 1985.

Notes on Pedagogy. Formal development should be coupled with adequate emphasis on modeling applications, and preferably some hands-on work in any form such class projects, etc.

Contributor/s.

1. Ashutosh Ashutosh (https://www.linkedin.com/profile/view?id=198572337)
2. Abhijat Vichare (https://www.linkedin.com/pub/abhijat-vichare/2/822/828)

5.3.4 MNS-573-MJ: Functional Programming

Credits: 2 **Category:** Hybrid **Prerequisites.** MNS-505-MJP

Course Outcomes

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

1. Understand basics of functional programming.
2. Design an algorithm to solve problems of various kinds in modeling and simulation and implement using functional language.
3. Debug the code to spot logical errors, exceptions etc.
4. Write reasonably complex code for solving various problems in modeling and simulation.

Syllabus.

1. Introduction to theoretical framework for describing functions and their evaluation, variable binding and substitution, effective computability, Turing machine and Lambda calculus.
2. Dynamically and Statically Typed Functional Languages.
3. Other domain specific programming languages based on/implementing functional programming concepts; e.g, R, SQL, etc.
4. How does the chosen programming language (LISP/Haskell) implement the above concepts?
5. First-class and higher-order functions, pure functions, recursion.
6. Comparison with other programming paradigms.
7. Declarative *vs.* imperative, using pipelines, Computation by expression evaluation.
8. Reliability.
9. Syntax, coding style and other aspects of programming in the chosen functional language.

Suggested Texts/References. As recommended by the course instructor.

Notes on Pedagogy. Any functional programming language like such as `LISP`, `Scheme`, or `Haskell` can be chosen for this course. The concepts of functional programming, how they are used in the chosen language and programming in that language for M&S problem-solving should be clearly brought out by the instructor.

Contributor/s. Bhalchandra Gore (`https://scms.unipune.ac.in/~bwgore`), Charulata Patil (`charulata@cms.unipune.ac.in`)

5.3.5 22-E005: Computing with Java

Credits: 2 **Category:** Hybrid **Prerequisites.** MNS-505-MJP

Course Outcomes

Upon successful completion of this course, the student will

1. Understand basics of object-oriented programming (OOP), syntax, semantics of Java.
2. Design an algorithm to solve problems of various kinds in modeling and simulation and implement using Java programming language.
3. Debug the code to spot logical errors, exceptions etc.
4. Write reasonably complex Java code/applets for various problems in modeling and simulation.

Syllabus.

1. Overview of Java programming. Compiling and Running a Java program. Coding Standards. Introduction to Java programming language. Data types, operators and control structures. Primitive data classes.
2. OOAD. Class, object, inheritance, Class Design and Implementation, class responsibilities, data encapsulation, Polymorphism.
3. Exception handling. Exception handling in Java.
4. Java generics. generic programming, ready made classes and generic data structures. Generic Algorithms. Enumerations, autoboxing, and annotations. The Collections Framework
5. Threads. Thread handling in Java.
6. GUI. AWT library. Window, Controls, events, callback, event handlers, frames, graphics, image handling.
7. Applets. Application development and animations.

Suggested Texts/References.

1. Herb Schildt, *Java: The Complete Reference*. McGraw Hill Education (India) Private Limited, 2013.
2. Various web resources; especially, http://docs.oracle.com/javase/7/docs/api/.

Notes on Pedagogy.

Contributor/s. Bhalchandra Gore (https://scms.unipune.ac.in/~bwgore)

5.3.6 MNS-575-MJP: Advance Python programming

Credits: 2 **Category:** Laboratory **Prerequisites.** MNS-505-MJP

Course Outcomes

Python offers a powerful means for M&S via its intuitive object-oriented framework and in-built libraries. The goal of this course is not the master Python after its introduction in MNS-505-MJP, and demonstration of how Python can be used in mathematical modeling. At the end of the course students are expected to develop Python programs to implement intermediate to advance level of mathematical models.

Syllabus.

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/References.

1. David Beazley, Brian K. Jones, *Python Cookbook*. O'Reilly Media, 2013.
2. Mark Lutz, *Programming Python*. O'Reilly Media, 2010.
3. The Official Python Documentation (https://docs.python.org/).

Notes on Pedagogy. Since the focus of the course is using Python for modeling, the syntax could be introduced as a necessity to solve the given problem. 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.

Contributor/s. Bhalchandra Pujari (http://scms.unipune.ac.in/~bspujari)

5.3.7 MNS-576-MJ: Computing with R

Credits: 2 **Category:** Laboratory **Prerequisites.** MNS-505-MJP

Course Outcomes

The R (http://cran.r-project.org/) statistical computing environment, built around the S programming language, is rich in computational statistics primitives. It is open-source and supported by an ever-growing community of users and contributors. It allows a variety of programming styles from quick-and-dirty explorations to elaborate imperative, procedural, object-oriented, and functional coding. It is ideally suited for statistical modeling and data analysis, graphics and visualization, as well as a platform for teaching/learning probability and statistics through hands-on exploration. As such, R is a must for any broad-based M&S curriculum with a statistical modeling/data analysis component. Goals: proficiency in computational problem-solving using R; specifically, decent algorithmic, coding, and scripting skills.

Syllabus.

1. Overview of R and S. History of R. Why use R? When not to use R? GUIs for R. Invoking and exiting the R interpreter environment. Getting help and finding information. `demo()`. The six atomic types. Assignment operators. Standard arithmetic and logical operators. Comments. Conditionals and loops. Parenthesis and braces. Expressions. Every expression has a value. Common composite data types: `vector`, `list`, `matrix`, and `data.frame`. The elementwise operations rule for `vector` and related container types. `functions`. Writing and executing R scripts: `source()` and `Rscript`.
2. Case studies illustrating R capabilities, in-built functions, and common packages. Overview of R graphics. Probability distributions and random number generators. Creating numerical and graphical data summaries, and exploratory data analysis. Complex numbers, numerical methods, etc. Character strings. Set operations. Interface to the operating system shell. Data input and output.
3. Installing R and R packages locally into a linux user account. Installing R from source: `configure` – `make` – `make install` sequence. Installing packages: `install.packages()` and the `R CMD INSTALL` mechanism.
4. Migrating from C to R. Automatic type identification in an assignment vs. explicit declaration of data type. `;` and `\n` as expression terminators. Explicit loops vs. vectorization.

5. Getting performance from R codes. Coding style guidelines. Explicit loops vs. vectorization. The `compiler` package. Debugging and profiling tools. Interfacing with `C`, `C++`, `fortran`.
6. Hands-on explorations using R. Any reasonable set of hands-on problems designed to enhance computational problem-solving and algorithmic abilities. Such problems may be related to M&S in general, or specifically to topics from other courses (e.g., probability theory, statistical inference) in the programme or the instructor's field of expertise.

Suggested Texts/References.

1. W. N. Venables, D. M. Smith, and the R Development Core Team, *An Introduction to R*. The R Project, latest available edition. http://cran.r-project.org/doc/manuals/R-intro.html
2. John Verzani, *Using R for Introductory Statistics*. Chapman & Hall/CRC, 2005.
3. Daniel Navarro, *Learning Statistics with R: A Tutorial for Psychology Students and Other Beginners*. Self-published, 2013. http://learningstatisticswithr.com/
4. Paul Murrell, *R Graphics*. Chapman & Hall/CRC, 2011.
5. Patrick Burns, *The R Inferno*. http://www.lulu.com/, 2012. Available at http://www.burns-stat.com/documents/books/the-r-inferno/.
6. W. N. Venebles and B. D. Ripley, *Modern Applied Statistics with* `S-Plus`. Springer, 2002.
7. R. G. Dromey, *How to Solve It By Computer*. Prentice-Hall, 1982.

Notes on Pedagogy. This syllabus is based on an outline for a longer course that was refined over several course deliveries by the course contributor. Depending on the background and capabilities of the students, this outline may need to be diluted or intensified – without compromising upon the essentials and goals for the course. Apart from familiarizing a student with R, a major emphasis of this course is on tinkering and exploration, on computational problem-solving, and on translating a problem into a computational framework leading to a solution and/or a better understanding of the problem, and on how R can be used as a M&S tool, and for exploring/visualizing probability and statistics concepts. Assignments often consist of problems that are exploratory in nature (e.g., illustrating formal results that may be difficult to grasp, such as the central limit theorem; see MNS-503-MJ), or require a student to understand an algorithm from its plain-English or pseudocode description (e.g., generating permutations in the lexicographic order). Examinations may consist of problems not necessarily discussed in the class: Here, adequate information about the method of solution or algorithm is provided.

Contributor/s. Mihir Arjunwadkar (https://scms.unipune.ac.in/~mihir)

5.3.8 MNS-577-MJP: Computing with MATLAB/Scilab

Credits: 2 **Category:** Laboratory **Prerequisites.** CP

Course Outcomes

MATLAB and its open-source parallel Scilab are popular, powerful, and flexible platforms for numerical and symbolic computation, visualization and graphics, etc., and are rich in computational primitives for diverse fields from digital signal processing to statistics. This course aims at developing an intermediate skill level in writing scripts, performing calculations, using the command line, importing data from files, plotting data, and integrating with other programming languages such as C.

Syllabus.

1. Introduction. Environment. Workspaces. General syntax.
2. Numerics. Creating matrices. Matrix operations. Sub-matrices. Statistical operations. Polynomials, differential equations.
3. Plots. Plotting graphs for 2D, 3D functions. Various types of plots.
4. Programming. Functions, Scilab/MATLAB programming language, Script files and function files.
5. I/O. Reading, writing data in various formats.
6. Interfacing with programming languages such as C.

Suggested Texts/References.

1. Amos Gilat, *MATLAB: An Introduction with Applications*. Wiley, 2008.
2. Mathews and Fink, *Numerical Methods Using MATLAB*. Pearson, 2004.
3. J. C. Polking and D. Arnold, *Ordinary Differential Equations using MATLAB*. Pearson, 2003.
4. An extensive comparison of MATLAB and Scilab: http://www.professores.uff.br/controledeprocessos-eq/images/stories/Comparative-Study-of-Matlab-and-Scilab.pdf

Notes on Pedagogy. Case studies and problems used for introducing MATLAB/Scilab can be derived from other courses running concurrently.

Contributor/s. Bhalchandra Pujari (https://scms.unipune.ac.in/~bspujari)

5.3.9 MNS-578-MJP: Computing with C

Credits: 2 **Category:** Laboratory **Prerequisites.** CP

Course Outcomes

Upon successful completion of this course, the student is expected to be able to

1. Understand basics of procedural/functional programming, syntax, semantics of C.
2. Design an algorithm to solve problems of various kinds in modeling and simulation and implement using C programming language.
3. Debug the code to spot logical errors, exceptions etc.
4. Write reasonably complex C code for solving various problems in modeling and simulation.

Syllabus.

1. ANSI C. Syntax, data types, concept of void, variables, operators, expressions and statements, character input and output, console input and output, inclusion of standard header files, pre-processor directives.
2. Control flows. If-else, for, while, do-while, switch-case, break and continue, code blocks and nesting of blocks.
3. Functions. Basics of functions, return statement, recursion, function blocks, static variables
4. Memory management. Dynamic versus static memory allocation, freeing memory, arrays, memory layout of multidimensional arrays.
5. Pointers. Concept of pointers, pointer arithmetic, pointers versus arrays, array of pointers.
6. Program compilation and debugging techniques. Introduction to tools like `gdb` together with `ddd`, `GNU make` and profiler `gprof`. Code organization across files. Version/revision control using `svn` or `git`.
7. Structures and Unions. Structures and unions, bit fields, typedef, self referential structures, their use in link-list, queue, stack *etc.*
8. Input and output in C. Files, file operations.

Suggested Texts/References.

1. Kernighan and Ritchie, *The C Programming Language*. PHI, 1990.
2. R. L. Kruse, B. P. Leung, C. L. Tondo, *Data Structures And Program Design In C*. Pearson Education, 2007.

Notes on Pedagogy. The student may be exposed here to the bare-basics of finite-precision representations and arithmetic if time permits, and at the discretion of the instructor.

Contributor/s. Bhalchandra Gore (`https://scms.unipune.ac.in/~bwgore`)

5.3.10 MNS-579-MJP: Statistical Models and Methods

Credits: 4 **Category:** Laboratory **Prerequisites.** MNS-503-MJ

Course Outcomes

At the end of the course students be well versed with developing the models based on various basic and advanced statittical techniques.

Syllabus.

1. Linear and logistic regression. Simple Linear Regression. Least Squares and Maximum Likelihood. Properties of the Least Squares Estimators. Prediction. Multiple Regression. Model Selection. Logistic Regression.
2. Multivariate models. Random vectors. Estimating the correlation. Multivariate normal. Multinomial.
3. Inference about independence. Two binary variables. Two discrete variables. Two continuous variables. One continuous variable and one discrete variables.
4. Causal inference. The counterfactual model. Beyond binary treatments. Observational studies and confounding. Simpson's paradox.
5. Directed graphs and conditional independence. Conditional independence. Directed graphs. Probability and directed graphs. More independence relations. Estimation for directed graphs.
6. Undirected graphs. Undirected graphs. Probability and graphs. Cliques and potentials. Fitting graphs to data.
7. Log-linear models. The log-linear model. Graphical log-linear models. Hierarchical log-linear models. Model generators. Fitting log-linear models to data.
8. Nonparametric curve estimation. The bias-variance trade-off. Histograms. Kernel density estimation. Nonparametric regression.
9. Smoothing using orthogonal functions. Orthogonal functions and L_2 spaces. Density estimation. Regression. Wavelets.
10. Classification. Error rates and the Bayes classifier. Gaussian and linear classifiers. Linear regression and logistic regression. Relationship between logistic regression and linear discrimination analysis. Density estimation and naive Bayes. Trees. Assessing error rates and choosing a good classifier. Support vector machines. Kernelization. Other classifiers.
11. Time series analysis. Overview of the Box-Jenkins and Bayesian approaches. Principles of nonlinear and chaotic time series analysis.

Suggested Texts/References.

1. Larry Wasserman, *All of Statistics*. Springer-Verlag, 2004.
2. Larry Wasserman, *All of Nonparametric Statistics*. Springer-Verlag, 2006.

Notes on Pedagogy. The suggested syllabus is primarily the content of Part III of the first recommended reference book. Appropriate balance of theory and practice is recommended. Listed above in the syllabus section is a representative selection of topics for this course rated at 1 credit each. With approval from the Centre, a qualified instructor may introduce additional 1-credit topics. Any selection of topics adding up to the requisite number of credits may be offered.

Contributor/s. Mihir Arjunwadkar (`https://scms.unipune.ac.in/~mihir`)

5.3.11 MNS-581-MJ: Stochastic Simulation

Credits: 2 **Category:** Hybrid **Prerequisites.** MNS-503-MJ (Sec. 4.1.3)

Course Outcomes

To familiarize the student with simulation methods (together with their modeling contexts) involving the use of randomness, including methods for sampling from probability distributions, Monte Carlo integration, Markov chain Monte Carlo methods, etc.

Syllabus.

1. Randomness. Randomness in natural processes: decaying nuclei, chaotic oscillators, leaky faucets, cosmic ray showers, etc. Randomness as complexity, non-compressibility of information, unpredictability, ignorance, statistical independence. Randomness as a modeling assumption. Randomness and entropy.
2. Pseudo-random number generators. Generating deterministic sequences of numbers that appear random. Uniform pseudo-random number generators their properties. Breaking correlations via shuffling. Mersenne Twister and other state-of-the-art generators: an overview. Simple transformations from the Uniform. Other distributions as transformations from the Uniform: exponential, Cauchy, Beta, etc. $N(0,1)$ using Box-Müller and other methods. Arbitrary distributions and acceptance-rejection sampling. Testing for randomness: how random is random enough? `DIEHARD` and other test batteries.
3. (Optional) Correlated random numbers. Normal random numbers with prespecified correlations. Other representative correlated situations and methods.
4. Monte Carlo integration. Estimating π using a dartboard. Estimating one-dimensional integrals: basic MC integration. Importance sampling for better estimators and tighter errorbars. Deterministic vs. Stochastic: Behaviour of the error as function of the number of dimensions.
5. Sampling and integration in more than one dimension. Markov chains, their properties, and limit theorems. Metropolis, Metropolis-Hastings and Gibbs sampling. Master equation, detailed balance, and why Metropolis-Hastings works. Relationship between Metropolis-Hastings, Metropolis, and Gibbs. Relationship between Metropolis and rejection sampling. A survey of illustrative problems involving high-dimensional distributions, integration/expectation, and simulations. Practical considerations: the adjustable step length

parameter, behaviour of Markov chain Monte Carlo methods when the distribution being sampled is multimodal, burn-in or equilibration behaviour, detecting equilibration/convergence of the Markov chain, convergence diagnostics, correlations and error bars on estimates, etc.

6. (Optional) (Specialized (M&)S methods involving randomness. Reaction kinetics, epidemiology, and population dynamics: The Gillespie method. Agent-based stochastic models in epidemiology and other fields. Tutorial on stochastic differential equations. Discrete versus continuous, stochastic versus deterministic: What is more appropriate/useful for given problem?

Suggested Texts/References.

1. Charles M. Grinstead and J. Laurie Snell, *Introduction to Probability*. American Mathematical Society, 1997 or later. https://math.dartmouth.edu/~prob/prob/prob.pdf
2. Hoel, Port, and Stone, *Introduction to Stochastic Processes*. Houghton Mifflin, 1972.
3. George Casella and Edward I. George, *Explaining the Gibbs Sampler*, The American Statistician **46**(3) 167–174 (1992).
4. Brooks, Gelman, Jones, and Meng (eds.), *Handbook of Markov Chain Monte Carlo*. Chapman and Hall/CRC, 2011.
5. Liang, Liu, and Carrol, *Advanced Markov Chain Monte Carlo Methods: Learning from Past Samples*. Wiley, 2010.
6. Gilks, Richardson, and Spiegelhalter, *Markov Chain Monte Carlo Methods in Practice*. Chapman and Hall, 1996.

Notes on Pedagogy. This syllabus is based on an outline for a longer course that was refined over several course deliveries by the contributor (see below). Depending on the background and capabilities of the students, this outline may need to be somewhat diluted or intensified – without compromising upon the essential content and the goals for the course. This course needs sufficient level of hands-on activities, and students require adequate computing skills. What is not mentioned explicitly in the syllabus is the modeling contexts in which stochastic methods can be useful. Exposing the student to such modeling contexts is a must. Specific modeling contexts can be chosen by the instructor according to her/his field of specialization.

Contributor/s. Mihir Arjunwadkar (https://scms.unipune.ac.in/~mihir)

5.3.12 MNS-582-MJ: Data Visualization

Credits: 2 **Category:** Hybrid **Prerequisites.** MNS-504-MJP

Course Outcomes

The purpose of this course is to introduce the audience to the field of visual design so as to create awareness and literacy about visual design. The intended audience of this course is assumed to be a lay audience as far as this area is concerned. The expected outcome is awareness and improved understanding of visual display of quantitative information. The design awareness created through introductory lectures should be reinforced through hands-on visualization work using available tools, packages, APIs, libraries, etc.

Syllabus.

1. Introduction to visual design. A guided tour through the world of visual/graphic design so as to create a sense of visual design awareness and literacy. A crisp overview of Edward Tufte's framework for envisioning information.
2. Data visualization. Starting from the all-familiar scatterplots, pair plots, boxplots, histograms, wiremesh plots, contour plots, etc., introduce principles and strategies for visualizing complex / hierarchical / high-dimensional data. A pedagogic approach to data visualization (such as that in Tamara Munzner's book mentioned below) is recommended for this part.
3. Hands-on work. Group and individual (mini)projects. Students to survey and choose appropriate tools, libraries, APIs, etc., best-suited for their chosen topic, under the guidance of the instructor/s.

Suggested Texts/References.

1. Robin Williams, *The Non-Designer's Design Book*. Peachpit Press, 2014.
2. Tamara Munzner, *Visualization Analysis and Design*. CRC Press, 2014.
3. Edward Tufte, *Envisioning Information*. Graphics Press, 1990. Edward Tufte, *The Visual Display of Quantitative Information*. Graphics Press, 2001.
4. David McCandless, *Information is Beautiful*. Graphics Press, 2009.
5. Nathan Yau, *Visualize This: The FlowingData Guide to Design, Visualization, and Statistics*. Wiley, 2011.
6. Manuel Lima, *Visual Complexity: Mapping Patterns of Information*. Princeton Architectural Press, 2013.

7. Stephen Few, *Now You See It: Simple Visualization Techniques for Quantitative Analysis*. Analytics Press, 2009.
8. Chun-houh Chen, Wolfgang Härdle, and Antony Unwin, *Handbook of Data Visualization*. Springer, 2008.
9. Phil Simon, *The Visual Organization: Data Visualization, Big Data, and the Quest for Better Decisions*. Wiley, 2014.
10. Kieran Healy, *Data Visualization: A Practical Introduction*. Princeton University Press, 2018.
11. Jack Dougherty & Ilya Ilyankou, *Data Visualization for All*. https://datavizforall.org/, 2019.

Notes on Pedagogy. While this course does not assume any design literacy on part of the audience, the audience may have some prior exposure to design principles from fields such as engineering design, software design, etc. Any appropriate set of books, internet resources, and software such as ggplot2, D3, etc., may be used as per the instructor's preferences.

Contributor/s.

1. Jitendra Pawagi (https://landconcern.blogspot.com/)
2. Jayant Gadgil (http://sciencepark.unipune.ac.in/about-us.html)
3. Girish Dalvi (http://www.idc.iitb.ac.in/girish/)
4. Bhalchandra Pujari (https://scms.unipune.ac.in/~bspujari)
5. Mihir Arjunwadkar (https://scms.unipune.ac.in/~mihir)

5.3.13 MNS-583-MJ: Difference Equations

Credits: 2 **Category:** Hybrid **Prerequisites.** MNS-501-MJ

Course Outcomes

At the end of the course students should be able to classify and interpret different types of differential equations (ODEs/PDEs) and identify appropriate methods of solution.

Syllabus.

1. Definitions of difference equations (ordinary and partial) dependent and independent variables, order and degree, types (linear, quasilinear) elliptic, hyperbolic and parabolic difference equations. Types of side conditions for each.
2. Translation operator and algebraic methods to solve degree one homogeneous and inhomogeneous difference equations. Similarity with ODE methods. Methods using the Z transform.
3. Numerical methods of solution of ordinary difference equations. Examples of usage of data structures and creation of algorithms in this context. Difference equations that result from discretization of differential equations (two examples like Euler stepping and Runge Kutta method).
4. Translation operators for partial difference equations and algebraic methods for solving basic forms of partial difference equations analytically.
5. Numerical methods for partial difference equations, classification as sweeping and stepping methods. Examples of usage of data structures and creation of algorithms in this context. Difference equations that result from discretization of partial difference equations (examples of finite difference method only).
6. Application of difference equations in population dynamics and finance.
7. Introduction to theoretical concepts of stability, oscillation and asymptotic behaviour of difference equations and their solutions.

Suggested Texts/References.

1. Saber N. Elaydi, *Introduction to Difference Equations*. Springer, 1999.
2. Ronald E. Mickens, *Difference Equations*. CRC Press, 1991.

3. Walter C. Kelley and Allan C. Peterson, *Difference Equations: An Introduction with Applications*. Academic Press, 2001.
4. *Mathematical Modelling Through Difference Equations*, Chapter 5, in: J. N. Kapur, *Mathematical Modelling*. New Age International Publishers, 2008.
5. Sui Cun Cheng, *Partial Difference Equations*. Taylor and Francis, 2003.

Notes on Pedagogy.

Contributor/s. Sukratu Barve (https://scms.unipune.ac.in/~sukratu)

5.3.14 MNS-584-MJ: Ordinary Differential Equations

Credits: 2 **Category:** Theory **Prerequisites.** MNS-501-MJ

Course Outcomes

At the end of the course students should be well versed with solution and interpreation of ordinary differential equations.

Syllabus.

1. Definition of an ordinary differential equation, order and degree along with examples, Definition of solution. General particular and singular solutions, homogeneous functions.
2. First order ODEs having homogeneous functions. Shift of origin change of variables for converting to homogeneous form. Exact first order ODEs. Standard examples of exact ODEs. Integrating factors. Standard examples of integrating factors in various categories of first order ODEs. Linear first order ODEs and integrating factor, Bernoulli's ODE First order ODEs with higher degree and methods of solution (solvable for x, y or dy/dx) Clairaut's form of first order ODE.
3. Second order ODEs (homogeneous and non-homogeneous).
4. ODEs with constant coefficients, (optional examples of analysis of spring-mass-dashpot system) differential operator and its polynomial, complementary and particular integrals, general procedure of obtaining solution.
5. ODEs with variable coefficients, method of variation of parameters (only in case of second order ODEs)
6. Revision of sequences, series and convergence. Series of functions, Power series, ratio test, radius of convergence, series solutions of ODEs, method exemplified in particular cases (second order ODEs) Bessel, Hermite and Legendre ODEs and their series solutions. Brief outline of special functions and their properties.
7. Side conditions of ODEs and their illustration in all the above techniques.
8. Conversion of higher order ODEs into first order ODEs with several dependent variables. Linear ODEs with several dependent variables, matrix formulation, interpretation of characteristic vectors and characteristic values, stability. Applications of this in perturbation of ODEs. Discussion of examples of 6 DoF analysis, control systems stability criteria, predator-prey and chemical reaction ODEs.
9. (Optional) Existence and uniqueness of solutions of first order ODEs, normed vactor space techniques, uniform continuity, Lifshitz functions and outline of

Picard's theorem. Linear ordinary differential operators and resolvants. Examples of resolvants in common linear ODEs. Relation of side conditions to resolvants. Qualitative analysis of ODEs: Limit sets, fixed points, limit cycles, basins of attractors, Poincare Bendixson theorem (without proof) Lienard's theorem (without proof).

Suggested Texts/References.

1. S. Balachandra Rao and H. R. Anuradha, *Differential Equations with Applications and Programs*. Universities Press, 1996.
2. E. Rukmangadachari, *Differential Equations*. Dorling Kindersley India, 2012.
3. A. Chakrabarti, *Elements of Ordinary Differential Equations and Special Functions*. New Age International, 1990.
4. E. A. Coddington and N. Levinson, *Theory of ordinary Differential Equations*. Tata-McGraw Hill, 1972.
5. G. F. Simmons, *Differential Equations with Applications and Historical Notes*. Tata-McGraw Hill, 1991.
6. G. F. Simmons and S. G. Krantz, *Differential Equations: Theory, Techniques and Practice*. Tata-McGraw Hill, 2007.

Notes on Pedagogy.

Contributor/s. Sukratu Barve (https://scms.unipune.ac.in/~sukratu)

5.3.15 MNS-585-MJ: Partial Differential Equations

Credits: 2 **Category:** Theory **Prerequisites.** MNS-584-MJ

Course Outcomes

At the end of the course students should be well versed with solution and interpreation of partial differential equations.

Syllabus.

1. Definition of partial differential equation. Order, dependent and independent variables, themes of classification and standard categories, first and second order PDEs; Laplace, heat and wave equations as basic examples of linear second order PDEs. Examples of higher order and non linear PDEs.
2. Cauchy Problems for First Order Hyperbolic Equations. Method of characteristics, Monge cone.
3. Classification of Partial Differential Equations. Normal forms and characteristics for second order PDEs. Principal symbol and quasilinear PDEs, classification of quasilinear PDEs, types of side conditions and principal symbols. General types of side conditions occurring in applications of hyperbolic, parabolic and elliptic PDEs.
4. Initial and Boundary Value Problems. Lagrange-Green's identity and uniqueness by energy methods.
5. (Optional) Stability theory. Energy conservation and dispersion.
6. Laplace equation. Mean value property, weak and strong maximum principle, Green's function, Poisson's formula, Dirichlet's principle, existence of solution using Perron's method (without proof).
7. Heat equation. Initial value problem, fundamental solution, weak and strong maximum principle and uniqueness results (outline of proofs and emphasis on interpretations)
8. Wave equation. Uniqueness, D'Alembert's method, method of spherical means, Duhamel's principle: outline of proofs with emphasis on interpretation.
9. Methods of separation of variables for heat, Laplace and wave equations. Various other methods of solution of PDEs and brief descriptions.
10. (Optional) (Numerical methods. Finite difference method as numerical methods for PDEs. Finite Difference Operators, Finite Difference methods, FDM for 1D heat and wave equations, implicit and explicit methods of solution,method of lines,Jacobi, Gauss Seidel and Relaxation methods (for 2D

Laplace and Poisson equations) von Neumann stability for difference equations and applications to 2D heat and wave equations. Stability and convergence of matrix difference methods.

Suggested Texts/References.

1. Erich Zauderer, *Partial Differential Equations of Applied Mathematics*. Wiley, 2006.
2. K. Sankara Rao, *Introduction to Partial Differential Equations*. PHI Learning, 2010.
3. Phoolan Prasad and Renuka Ravindran, *Partial Differential Equations*. New Age Publishers, 2012.
4. Lokenath Debnath, *Nonlinear Partial Differential Equations for Scientists and Engineers*. Birkhäuser, 2011.
5. Lawrence C. Evans, *Partial Differential Equations*. American Mathematical Society, 2010.

Notes on Pedagogy.

Contributor/s. Sukratu Barve (https://scms.unipune.ac.in/~sukratu)

5.3.16 MNS-586-MJ: Transforms

Credits: 2 **Category:** Theory **Prerequisites.** MNS-501-MJ

Course Outcomes

Integral and discrete transforms are often useful in transforming a complex problem into a simpler one. Moreover, insights about the system can be obtained through transforms. Needless to say that integral and discrete transforms are vital tools in the modeling and simulation premises. The goal of this course is to introduce students to a few commonly used transforms with substantial emphasis on Fourier transform. A significant computing aspect is also expected. Students should be able to write codes for some of the transforms. Students are also expected to learn to analyze the results of transformed signals through computational platforms such as `matlab/scilab/R`.

Syllabus.

1. Introduction and background. Brief introduction to vector spaces, function spaces and basis sets. Special functions. Function parity. Concepts in complex analysis. Kernel of an integral transform.
2. Fourier series. Periodic functions. Fourier series in trigonometric as well as complex exponent representation. Functions with arbitrary period. Solving differential equations with Fourier series.
3. Fourier transform. Fourier integrals. Fourier sine/cosine transforms. Fourier transform. Inverse Fourier transform. Properties of Fourier transform. Convolution. Applications. Solving differential equations using Fourier transform, Power spectrum and its interpretation.
4. Discrete Fourier transform and fast Fourier transform. Discretization. Sampling. Nyquist sampling. Discrete Fourier transform. Properties. Matrix representation of Discrete Fourier transform. Fast Fourier transform. Comparison.
5. Laplace transform. Laplace transform. Properties. Inverse transform using partial fractions, convolution and complex integration. Applications. Solution of differential equations.
6. (Optional) Z-transform. Definition of Z-transform. Properties. Inverse Z-translations using complex integral methods. Difference equations.
7. (Optional) Wavelet transform. Limitations of Fourier transform. Introduction to wavelets and family of wavelets. Translations and scaling. Continuous and Discrete wavelet transform. Haar scaling and wavelet functions. Functions spaces. Decomposition using Haar bases. General wavelet system.

Daubechies wavelets. Multiresolution analysis. Analysis of output of wavelet transforms.

Suggested Texts/References.

1. L. C. Andrews and B. K. Shivamoggi, *Integral Transforms for Engineers*. Prentice-Hall of India, 2003.
2. Erwin Kreyszig, *Advanced Engineering Mathematics*. Wiley India, 2014.
3. R. N. Bracewell, *The Fourier transform and its applications*. Tata McGraw-Hill, 2003.
4. L. Devnath and D. Bhatta, *Integral transforms and their applications*. Chapman and Hall/CRC, 2010.
5. K. P. Soman and K. I. Ramchandran, *Insights into wavelets - From theory to practice*. Prentice-Hall India, 2005.
6. Online coursework such as Brad Osgood (Stanford), Alan Oppenheim (MIT), etc.

Notes on Pedagogy. An instructor may choose to alter the order to introduce the transforms. However it is logical to start with Fourier series for periodic function, followed by Fourier transform for non-periodic function. The limitation of Fourier transform to resolve the function in both frequency and time domains leads to use of Wavelet and on the other hand Laplace transform can be viewed as 'generalized' Fourier transform, which takes into account the real part of frequency. Similarly Z-transform can be seen as generalized discrete Fourier transform.

Modules like Laplace transforms may require separate introduction to Complex numbers and analysis.

A significant emphasize is expected to be on students developing their own codes for various transforms.

Contributor/s. Bhalchandra Pujari (`https://scms.unipune.ac.in/~bspujari`)

5.3.17 MNS-588-MJP: Advanced Statistics Laboratory 1

Credits: 2 **Category:** Laboratory **Prerequisites.** MNS-503-MJ, MNS-553-MJ

Course Outcomes

This course intends to augments the theoretical knowledge gained in Probability and Statistical Inference with hands-on explorations.

Syllabus.

1. `Numpy` and `matplotlib` or a similar free and open-source environment suitable for data analysis (e.g., R).
2. Computational exercises related to probability theory. Simulation of problems like Drunkard's walk, St. Petersburg paradox, Monty Hall problem. More hands-on examples: probabilistic simulations of models such as SIR and Schelling.
3. Drawing random numbers from various distributions and plotting their histograms along with theoretical curves. See the open course material for this course at https://apps.scms.unipune.ac.in/moodle/course/view.php?id=190 Computing and plotting histograms for sum and maximum of several random variables. Computational verification of law of large numbers and central limit Theorem.
4. Computing Pearson's and Spearman's correlation coefficients for a given dataset. Finding associated confidence intervals using resampling methods like Bootstrap or Jackknife.

Suggested Texts/References.

Notes on Pedagogy.

Contributor/s. Snehal Shekatkar (http://inferred.in)

5.3.18 MNS-623-MJP: Advanced Statistics Laboratory 2

Credits: 2 **Category:** Laboratory **Prerequisites.** MNS-503-MJ, MNS-553-MJ

Course Outcomes

This course intends to augments the theoretical knowledge gained in Probability and Statistical Inference with hands-on explorations.

Syllabus.

1. Computing sample mean, sampling distribution, and normal-based confidence interval for different parameters (e.g., mean) of various distributions (Uniform, Normal, Exponential) when standard deviation is known and when it is unknown. Computationally observing failure of sample mean for Cauchy distribution. Similar computations when sample is small and comparing Z-based vs. t-based confidence intervals.
2. Computing empirical distribution function and 95% confidence band for some real-world data (e.g., Old Faithful data). Estimating the mean/median using Bootstrap. Computing correlation coefficient and associated confidence interval using Bootstrap or Jacknife.
3. Using hypothesis testing to decide if the sample means of two moderate sized datasets are same. Repeating the same when dataset sizes are small using Student's t-test.

Suggested Texts/References.

Notes on Pedagogy.

Contributor/s. Snehal Shekatkar (`http://inferred.in`)

5.3.19 MNS-620-MJP: Computational Fluid Dynamics Laboratory

Credits: 2 **Category:** Laboratory **Prerequisites.** MNS-562-MJ

Course Outcomes

The purpose of this course is to introduce commonly used (open-source) libraries, tools, packages, and platforms for computational fluid dynamics, so as enhance the industry-preparedness of students. Such tools include, for example, `CFD Python`, `PyCFD`, etc.; meshing tools such as `Gmsh`, etc.; `SWIG`, `PyPar`, `PySPH`, and `SciPy Weave` for development of efficient parallel codes; as also `SU2` and/or `OpenFoam` at a greater depth than what is covered in MNS-562-MJ. Not all the tools mentioned above need be covered in the course. The selection of particular tool/s may change from course instance to course instance, depending on (a) the instructor's expertise and preference, (b) industry trends and developments in this field, and (c) a balance between depth and breadth.

Syllabus.

1. A broad survey of commonly used and available tools such as those mentioned above.
2. A crisp introduction to one or two such tools of instructor's choice.
3. Individual or group miniprojects. Students may survey and choose any other appropriate tools, libraries, APIs, etc., best-suited for their chosen topic, under the guidance of the instructor/s.

Suggested Texts/References. Appropriate set of internet and other resources recommended by the instructor.

Notes on Pedagogy. This course is intended to complement the computational fluid dynamics courses MNS-562-MJ and MNS-612-MJ through hands-on case studies, explorations and applications, with focus on (open-source and free) tools. The instructor may consider an open-laboratory, tinkering / experimentation-based approach where different students / groups may explore different (open-source) tools through individual or group mini-projects.

Contributor/s.

1. Disha Patil (https://www.linkedin.com/in/dishapatil/)
2. Bhalchandra Pujari (https://scms.unipune.ac.in/~bspujari)
3. Bhalchandra Gore (https://scms.unipune.ac.in/~bwgore)
4. Mihir Arjunwadkar (https://scms.unipune.ac.in/~mihir)

5.3.20 MNS-621-MJP: Machine Learning Laboratory

Credits: 2 **Category:** Laboratory **Prerequisites.** MNS-565-MJ

Course Outcomes

The purpose of this course is to introduce commonly used (open-source) libraries, tools, packages, and platforms for machine learning and artificial intelligence, so as enhance the industry-preparedness of students. Such tools include, for example, `TensorFlow`, `Scikit-learn`, `Keras`, `PyTorch`, `Theano`, `GenSim`, `Caffe`, `Chainer`, `Statsmodels`, `Neon`, `Nilearn` in the `Python` world, similar tools in the `R` world, and APIs such as `Apache Spark`, etc. Not all the tools mentioned above need be covered in the course. The selection of particular tool/s may change from one course instance to another, depending on

1. the instructor's expertise and preference,
2. industry trends and developments in this field, and
3. a balance between depth and breadth.

Syllabus.

1. A broad survey of commonly used and available tools such as those mentioned above.
2. A crisp introduction to one or two such tools of instructor's choice.
3. Individual or group miniprojects. Students may survey and choose any other appropriate tools, libraries, APIs, etc., best-suited for their chosen topic, under the guidance of the instructor/s.

Suggested Texts/References. Appropriate set of internet and other resources recommended by the instructor.

Notes on Pedagogy. This course should complement the machine learning courses MNS-565-MJ (Sec. 5.1.6) and MNS-615-MJ (Sec. 5.2.6). It is intended to be a hands-on course involving case studies and applications, with focus on (open-source) tools. The instructor may consider an open-laboratory, tinkering / experimentation-based approach where different students / groups may explore different (open-source) tools through individual or group mini-projects.

Contributor/s.

1. V. K. Jayaraman (https://in.linkedin.com/in/jayaraman-valadi-a7916925)
2. Bhalchandra Pujari (https://scms.unipune.ac.in/~bspujari)
3. Bhalchandra Gore (https://scms.unipune.ac.in/~bwgore)
4. Mihir Arjunwadkar (https://scms.unipune.ac.in/~mihir)

5.3.21 MNS-622-MJ: Advanced Data Analysis

Credits: 2 **Category:** Hybrid **Prerequisites.** MNS-553-MJ

Course Outcomes

Given the basic background in statistical modeling developed in prior courses, this course intends to expose the student to data analysis in a hands-on and problem-centric manner, so as to develop a feel for the challenges involved. At the end of this course, the student is expected to develop understanding about

1. formulating the problem in the light of the (scientific) question being explored;
2. choosing a statistical method that is most appropriate for the problem;
3. adapting or developing computational tools; and
4. evaluating results of the analysis and the statistical models involved critically.

This course is inspired by similarly-spirited courses such as http://www.stat.cmu.edu/~cshalizi/uADA/12/ and http://stat.ethz.ch/education/semesters/ss2010/CompStat.

Syllabus. An assortment of data analysis problems from any selection of fields at the discretion of the instructor/s. Problems should be chosen to reflect the diversity of (scientific) questions probed as well as that of statistical models and methods, such as (but not limited to): simple and multiple linear regression; model selection; finite mixture models; nonparametric methods for regression, density estimation, and classification (kernel methods, smoothing splines, classification and regression trees, additive models, etc.); resampling, bootstrap, and cross-validation methods.

Suggested Texts/References.

1. Cosma Rohilla Shalizi, *Advanced Data Analysis from an Elementary Point of View*. Unpublished, 2014. Available as http://www.stat.cmu.edu/~cshalizi/uADA/12/lectures/ADAfaEPoV.pdf.
2. Peter Bühlmann and Martin Mächler, *Computational Statistics*. Unpublished, 2008. Available as https://stat.ethz.ch/education/semesters/ss2010/CompStat.

Notes on Pedagogy. An apt name that conveys the spirit of this course better could have been *Delving into Data Dungeons*. This is intended to be a highly hands-on, interactive course involving an appropriate number of individual and group projects by the students. Apart from the projects, additional case studies may be presented by the instructor or guest speakers so as to develop perspective on the challenges involved in real-life data analysis situations.

Contributor/s. Mihir Arjunwadkar (`https://scms.unipune.ac.in/~mihir`)

Bibliography

[1] Mihir Arjunwadkar, Abhay Parvate, Sukratu Barve, P. M. Gade, and Dilip G. Kanhere. Master of technology (m.tech.) programme in modeling and simulation. https://scms.unipune.ac.in/reports/pd-20070223/, 2007. Accessed on: January 16, 2025.

[2] Snehal M. Shekatkar, Bhalchandra Pujari, and Mihir Arjunwadkar. Master of technology (m.tech.) programme in modeling and simulation. https://scms.unipune.ac.in/reports/pd-20220913/, 2022. Accessed on: January 16, 2025.

[3] Harvey Sheppard and Yves Zumbach. ylatex. https://github.com/HarveySheppard/yLaTeX, 2022. MIT License. Accessed on: January 16, 2025.

[4] OpenAI. Chatgpt. https://chat.openai.com, 2024. Accessed on June 04, 2025.

[5] Google DeepMind. Gemini ai. https://gemini.google.com, 2024. Accessed on June 04, 2025.

[6] HLCNCATF. Final report on national credit framework for public consultation. https://www.aicte-india.org/sites/default/files/report/Final%20report%20NCrF%20for%20public%20consultation.pdf, 2020. Accessed on January 16, 2025.