cms.unipune.ernet.in

Centre for Modeling and Simulation
university of pune • pune 411 007 • india

+91.(20).2569.1140 • +91.(20).2569.0842
office@cms.unipune.ernet.in

Master of Technology (M.Tech.) Programme in Modeling and Simulation

Approved by the University of Pune
Board of Studies: Scientific Computing

July 12, 2007 • Version 2.0
Minor Corrections: June 1, 2008

About This Document

This document outlines an academic programme, the *Master of Technology (M.Tech.) Programme in Modeling and Simulation*, designed by the people at the Centre for Modeling and Simulation, University of Pune. This academic programme has been approved by the University of Pune.

Feedback Requested. The utility of modeling and simulation as a methodology is extensive, and the community that uses it, both from academics and from industry, is very diverse. We would thus highly appreciate your feedback and suggestions on any aspect of this programme – We believe that your feedback will help us make this programme better.

Citing This Document

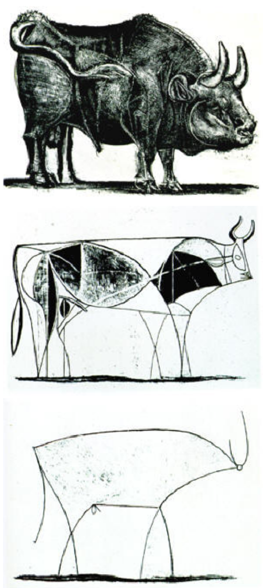
This is a public document available at `http://cms.unipune.ernet.in/reports/`. Complete citation for this document is as follows:

Mihir Arjunwadkar, Abhay Parvate, Sukratu Barve, P. M. Gade, and D. G. Kanhere, *Master of Technology (M.Tech.) Programme in Modeling and Simulation*. Public Document CMS-PD-20070223 of the Centre for Modeling and Simulation, University of Pune, Pune 411 007, INDIA (2007); available at `http://cms.unipune.ernet.in/reports/`.

Acknowledgments

Many people have contributed to the creation of this this programme and this document. Their time and effort, contributions and insights, and comments and suggestions have all gone a long way towards bringing this programme to its present form. The Centre values, appreciates and acknowledges all these forces that shaped this document and this programme, and would like to extend profound and sincere thanks to all of them. A comprehensive list of our contributors and reviewers can be found in Sec. A.

About the Centre



The spirit of scientific modeling as illustrated, perhaps co-incidentally, in Picasso's Stieren *series.*

The Centre for Modeling and Simulation, University of Pune was established in August 2003 with a vision to promote awareness about 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 scientific disciplines.

As a part of fulfilling this vision, the Centre designed a novel and highly interdisciplinary academic programme called the Advanced Diploma Programme in Modeling and Simulation; this became operational beginning with academic year 2005-06. The M.Tech. Programme in Modeling and Simulation outlined in this document incorporates insights gained from the operation of this diploma programme.

Collectively, the Centre possesses expertise in the areas of mathematical and computational physics, materials modeling, statistical data modeling and analysis, mathematical modeling, scientific computing, and numerics. The Centre also possesses excellent computing facilities, some of which are managed and maintained by the Centre as a campus-wide computing resource. The greatest strength of the Centre is its competent and highly motivated staff, both academic as well as non-academic. The resulting academic, work, and intellectual environment at the Centre is an open, vibrant, lively, and non-hierarchical one.

For more information, visit our website `http://cms.unipune.ernet.in/`.

Administrative Summary of the **M.Tech.(M&S) Programme**

Title of the Programme	Master of Technology (M.Tech.) Programme in Modeling and Simulation. *M.Tech.(M&S) Programme for short.*
Designed By	Centre for Modeling and Simulation, University of Pune, Pune 411 007, INDIA.
Status of the Programme	The M.Tech.(M&S) programme will be run as an **Autonomous Programme** under the **Faculty of Science**, University of Pune, by the Centre for Modeling and Simulation, University of Pune.
Board of Studies	Interdisciplinary School of Scientific Computing, University of Pune.
Modes of Operation	Full-time, part-time, and distance-learning modes. *The Centre plans to deploy the part-time and the distance-learning modes upon availability of sufficient faculty and other resources.*
Duration of the Programme	Full-time mode: 2 years. Part-time mode: minimum 3 years (10CH per week). Distance-learning mode: as per the progress of the student.
Total Number of Credits	100.
Credit Breakup	Year 1: • 11 core (45CR) and 1 elective (5CR) courses. Year 2: • 4 core (20CR) and 1 elective (5CR) courses. • minimum 6-month full-time project (25CR).
Structure and Syllabus	Enclosed (as Sec. 4 of this document).
Eligibility	1. B.E. or equivalent in any branch of engineering **OR** Master's degree in any science/arts/commerce discipline. **AND** 2. Background in mathematics equivalent to the University of Pune F.Y.B.Sc. mathematics syllabus as ascertained via the applicable selection criterion below.
Selection Criteria	*Regular Admissions* 1. Academic record, **AND** 2. Performance in an entrance test, with the selection threshold on performance set appropriately to match the vigorous and intensive nature of the programme, and applied uniformly across all candidates. **AND** 3. A statement of purpose, if necessary. *Industry-Sponsored Candidates* To be decided in consultation with the interested industry or organisation on a case-by-case basis. *Foreign Candidates* To be decided on a case-by-case basis.
Number of Seats	Not more than 30.
Fees	As per University rules, policies, and norms.

The **M.Tech.(M&S) Programme** at a Glance

Aims and Objectives. The Master of Technology (M.Tech.) Programme in Modeling and Simulation, also called the M.Tech.(M&S) Programme, is a unique, fast-paced, and vigorous academic training programme that aims at creating a breed of problem-solvers

- who have a breadth and perspective on mathematical modeling, a solid training in simulation methods, impeccable computational skills, and the ability to generate reasonable solutions, algorithmic or otherwise, for problems not necessarily encountered earlier;
- who are familiar with the current state of relevant technologies, and from familiar to skilled in a variety of relevant software tools and methodologies; and
- who, outside of their native knowledge domain, have sufficiently broad background and skills to interface between domain experts and coders in a multidisciplinary team.

Academic Structure. This is a highly interdisciplinary programme that focusses on mathematical modeling formalisms and simulation methodologies by integrating applied mathematics, statistics, and computing in a coherent package. *This is not a programme in the traditional domain of computer science.* This programme may, however, be thought of as a *computational science* programme.

The M.Tech.(M&S) Programme consists of core courses, elective courses, and a project. In the full-time mode, the duration of the M.Tech.(M&S) Programme is 2 years (4 semesters). Each semester is broken up into 18 weeks of instruction, 1 week for preparation, and 1 week for actual end-semester examinations. The credit total of the entire programme is 100 credits[1]: The first year is devoted to coursework consisting of 11 core courses (45 credits) and 1 specialized elective (5 credits); the second year is devoted to 4 core courses (20 credits), 1 specialized elective (5 credits), and a minimum 6-month full-time project/internship/industrial training (25 credits). Evaluation is based on (a) continuous assessment throughout a semester, and (b) an end-semester examination. Evaluation of the project is based on continuous assessment, a project report, and a presentation cum open defense.

Part-Time and Distance-Learning Modes. The academic structure of the M.Tech.(M&S) Programme is flexible enough so that it could in principle be run in full-time, part-time, and distance-learning modes. *In the foreseeable future, however, the Centre plans to run the M.Tech.(M&S) Programme only in the full-time mode.* With additional faculty, manpower, infrastructure, and resources, it may be possible to run the programme in a part-time or distance-learning mode. The part-time and distance-learning modes will make the programme most attractive to working individuals in the industrial sector and R&D organisations.

Historical Underpinnings. The Centre for Modeling and Simulation, University of Pune, has gathered considerable experience in designing and running academic programmes in modeling and simulation from the operation of its *Advanced Diploma Programme in Modeling and Simulation*. The M.Tech.(M&S) Programme expands upon, and incorporates academic, logistic, and operational insights obtained from, this diploma programme.

The M.Tech.(M&S) Programme may be considered a thoroughly revamped and modernized version of an existing University of Pune M.Tech. Programme in Modeling and Simulation. This existing M.Tech. programme used to be run in collaboration with the Institute for Armament Technology (IAT). IAT, now renamed DIAT, the Defense Institute for Advanced Technology, is now a deemed University.

[1]The current University of Pune credit system defines one credit as one clock hour of contact between faculty and students per week for 15 consecutive weeks.

Contents

1 Introduction

This document outlines a multidisciplinary academic programme, the Master of Technology (M.Tech.) Programme in Modeling and Simulation, designed by the Centre for Modeling and Simulation, University of Pune. For the purpose this document, this programme may be referred to as the M.Tech.(M&S) Programme or, simply, as *the Programme*. The Centre for Modeling and Simulation, University of Pune, may similarly be referred to as *the Centre*.

In this section, we first present (Sec. 1.1) our perspective and outlook on the modeling and simulation enterprise which forms the basis for the structure of the M.Tech.(M&S) Programme, followed by a comprehensive discussion on design considerations and possibilities for a programme of this kind (Sec. 1.3) and the need for such a programme (Sec. 1.2), and finally, a historical perspective on the programme (Sec. 1.4). An outline of the rest of this document is presented in Sec. 1.5.

1.1 The Modeling and Simulation Enterprise

Model, *n.* **1** a usually miniature representation of something; also : a pattern of something to be made **2** an example for imitation or emulation **3** archetype **4** a description or analogy used to help visualize something (as an atom) that cannot be directly observed **5** a system of postulates, data, and inferences presented as a mathematical description of an entity or state of affairs.

source: www.merriam-webster.com

Simulation, *n.* **1** The imitative representation of the functioning of one system or process by means of the functioning of another <a computer simulation of an industrial process> **2** Examination of a problem often not subject to direct experimentation by means of a simulating device.

source: www.merriam-webster.com

Simulation, *n.* **1** Imitation or representation, as of a potential situation or in experimental testing **2** Representation of the operation or features of one process or system through the use of another: computer simulation of an in-flight emergency **3** Attempting to predict aspects of the behaviour of some system by creating an approximate (mathematical) model of it. This can be done by physical modeling, by writing a special-purpose computer program or using a more general simulation package, probably still aimed at a particular kind of simulation (e.g., structural engineering, fluid flow). Typical examples are aircraft flight simulators or electronic circuit simulators. A great many simulation languages exist; e.g., Simula.

source: www.dictionary.com

A *model* tries to capture the essential features of a system under scrutiny. A *simulation*, on the other hand, attempts to represent a model of the system under study using some other well-understood system, the *simulation system*, wherein features of interest of the system under study are represented using properties of the simulation system. The correctness of representation of features of interest embodied by the model, and whether one system could at all be simulated by another should be the principal concerns of a "theory" of simulation. Assuming that the model did capture essential ingredients of the system being studied, and that the simulation system is capable of representing the model to sufficient accuracy, the corresponding simulation could be expected to mimic the behaviour of the underlying real-life system.

Most often, mathematics is used to model the system under study. Usually, the need to understand the system in a *quantitative* fashion and the ability to make *quantitative* predictions about the system are the key reasons for using mathematics in this fashion. From a purist point-of-view, patterns of behaviour of a system are oftentimes perceived as having some sort of inherent mathematical structure: Either deciphering that structure and expressing it in the most concise fashion, or developing a mathematical structure inspired by the observed behaviour of the system, is of great interest to some. Indeed, all scientific theories can be thought of as models representing aspects of "reality" to within their own respective domains of applicability.

Conceivably, one could attempt a naïve classification of mathematical models into two very broad classes, namely, probabilistic vs. non-probabilistic. Probabilistic (or stochastic) models, which are based on the formalism of probability theory, are perhaps the only known way to model situations where noise, randomness, complexity, uncertainty, or ignorance dominate either the behaviour of the system or the observation process. Non-probabilistic (sometimes called deterministic) models are based on the assumption of absence of these confounds.

The most challenging modeling and simulation problems arise when the system under study is neither well-understood nor, possibly, mathematised. For example, in comparison with physical systems, complex phenomena such as human social behaviour are neither as well-understood, nor are as mathematised, in microscopic detail, as physical theories. Construction of a simulation system for such phenomena may have somewhat nebulous boundaries between an art and a science. Furthermore, the bounds of validity of the mathematical model for a system have a direct influence on the reliability of the simulation system. The degree of identity (or similarity) between the behaviour of the real system and the simulated system needs to be determined so as to enable making of valid inferences based on observations of the simulated system.

A mathematical model typically extracts essential features of the system under study from the *knowledge domain*. For example, the mathematical model of air flight would have to incorporate fluid dynamical statements about the properties of air as a fluid system. A typical modern scientific team working on a challenging real-life problem consists of domain experts (e.g., experts in fluid dynamics), mathematical modeling experts, and experts from the field that is being used to construct the simulation system. In case modern digital computers are used to perform a simulation, the field of expertise for constructing a simulation system would be computer science and engineering. Other cases, for example, a full-scale flight simulation system, could involve expertise from almost every branch of engineering and robotics just to construct the simulation system.

A simulation is thus built using these three principal components; namely, (a) domain expertise, (b) mathematical modeling strategies, and (c) methodologies and technologies specific to the simulation system.

In modern times, digital computers have emerged as the preferred simulation system to perform simulations on. Usually, a mathematical model of the system under scrutiny is programmed into a computer and then run, and the behaviour of the model as observed in this simulation is used to make inferences about the real system that is being modeled and simulated. It appears that the use of computers saves costs because employing them in place of full-scale physical simulation systems reduces the engineering overheads (although the scientific principles at the base still need to be incorporated).

If we choose to use a computer to perform simulations, then its behaviour and properties as a simulation system must be "well-understood": this implies that an academic programme designed around the use of computers as simulation systems of choice must have sufficient theoretical and practical content to ensure strong foundations in computation.

1.2 Need for a Programme in Modeling and Simulation

The rapid pace of advances in computer and computation-related technologies over last few decades and the ever-increasing availability of comparatively inexpensive raw computing power have encouraged the use of computation and simulation methods (via mathematical modeling) in all domains of the human enterprise where quantitative reasoning has a significant role to play. This includes all branches of the scientific and technological endeavour.

Typical degree programmes (such as a two-year Masters programme in physical sciences mathematics and statistics, or Bachelors/Masters programmes in an engineering discipline) often include an introductory course on programming, and perhaps another one on domain-specific computational methods. On the other hand, typical degree programmes in disciplines such as computer science and software engineering that are expected to develop good programming skills and computing expertise, do not focus on the mathematical foundations of modeling and simulation. As a result, a sound and solid foundation in all three aspects of modeling and simulation is usually impossible to develop during such degree programmes given their focus, workload, and constraints.

The M.Tech.(M&S) Programme, designed by the Centre, is geared towards creating a breed of problem-solvers with a breadth and perspective on modeling, solid training in simulation methods, good problem-solving skills and versatility and, finally, the ability to generate reasonable solutions for problems not necessarily encountered previously. Specifically, the Programme is expected to fill the gaps between, e.g.,

- a conventional degree programme and industrial R&D work requiring substantial background in modeling and simulation, and
- a conventional degree programme and a research (Ph.D.) programme involving extensive computational and simulational research.

We would like to emphasize that the M.Tech.(M&S) Programme is a highly interdisciplinary programme that focusses on mathematical modeling formalisms and simulation methodologies by integrating applied mathematics, applied statistics, and applied computing in a coherent package. *As such, this is not a programme in the traditional arena of computer science.* This programme may, however, be thought of as a *computational science* programme.

1.3 Programme Design Considerations

1.3.1 From an Academic Perspective

As we saw earlier, a simulation is built on

- domain expertise,
- mathematical modeling formalisms and strategies, and
- methodologies and technologies specific to the simulation system.

A Master of Technology (M.Tech.) Programme such as the present one may be designed such that at the successful completion of training, a student may be one of the following:

1. **Conversant** with the basic principles in all the three components, or
2. **Conversant** with basic principles in the mathematical and simulational components, but unaware of any domain expertise, or
3. **Conversant** with only the simulational components, but unaware of domain expertise and mathematical methods, or

4. Be **skilled**, as opposed to **conversant**, with the basic principles of all the three components, but be **aware/conversant** with a few advanced techniques from the three, or
5. Be **skilled**, as opposed to **conversant**, with the basic principles of mathematical and simulational components, but be **aware/conversant** with a few techniques from the first, or
6. Be **skilled**, as opposed to **conversant**, with the basic principles of simulational components, but be **aware/conversant** with a few techniques from the first and the second[2].

All possibilities other than those above are intuitively judged to be more intensive and hence most likely to be out of the time bound of a two-year programme with 1–1.5 year course work. (1)–(3) have been included because given that the programme is a post-graduate degree programme, it may find favour to be licensed to outside entities for a wider spread. (4)–(6) are a set of levels of decreasing intensity, but admissible for a University post-graduate degree programme spanning 12-24 months. The alternative (4) is the most intense one, requiring active involvement of experts in *all* the three components, thus unsuitable for operation by the Centre in its current state.

Alternative (5) is the most practical alternative for the M.Tech.(M&S) Programme; we have thus designed a curriculum based on this alternative. Nevertheless, we would like to build provisions into the programme structure to offer specialized expert-level courses as and when domain experts become available. The reason for this design choice is threefold:

1. Although domain expertise is ultimately what is necessary for generating the best possible solution to a problem, specific domain expertise is very difficult to develop in a short span of 1.5 years (3 semesters) of coursework *in addition to* methodological training in the basics of mathematical modeling and computational simulation methods.
2. A prerequisite for admission in this programme is either a Master's degree in any discipline or a Bachelor's degree in Engineering, plus an adequate mathematical background, problem-solving and algorithmic thinking skills. We expect that students admitted to the programme will have some domain expertise in the area of their prior academic training programme.
3. We believe in developing strong problem-solving, algorithmic thinking and computational skills, and an interdisciplinary and problem-centric approach in our students which, coupled with broad background and training in mathematical modeling and computational simulation, should enable them to become *versatile*. By *versatility*, we mean
   - The ability to learn, and to develop domain-specific expertise in a new field as and when required; and
   - The ability to generate reasonable solutions (algorithmic/computational or otherwise) to a problem not necessarily encountered before.
   - The ability to make connections across domains so as to generate useful solutions.

[2] This last alternative is most suitable for an institution like the Department of Computer Science, where the focus is quite different, and simulation is one part of the many aspects that need to be covered. We disregard this alternative for the M.Tech.(M&S) Programme.

1.3.2 From a Student's Perspective

Most candidates tend to evaluate degree programmes from the point of view of job prospects and value addition. Employment opportunities for a graduate of the M.Tech.(M&S) Programme are most likely to come from:

- R&D centres and analysis organisations (in the industrial, defense, academic, government, and other sectors), that use modern computational and simulation methodologies in their design, development, and research/analysis initiatives. Such initiatives include all areas of engineering, science and technology including (but not limited to) materials science and engineering, nanotechnology, bioinformatics and biotechnology, computational fluid dynamics, molecular modeling and drug design, process engineering, etc.
- Research- and computing-oriented support positions in research-and-analysis organisations.
- Research programmes leading to an advanced degree such as Ph.D.

Though some of our students may find careers in the IT-related and conventional software industry, that is not the focus of the M.Tech.(M&S) Programme. However, we envisage that the skill-set being developed through the M.Tech.(M&S) Programme, specifically, problem-solving and programming skills, and versatility, will enable a student to migrate easily to such careers.

1.4 Historical Underpinnings

The M.Tech.(M&S) Programme, as outlined in this document, has evolved from and incorporates considerable academic, logistic, and operational expertise and insights accumulated from the operation of the Centre's existing academic programme, the *Advanced Diploma Programme in Modeling and Simulation*.

From another perspective, the M.Tech.(M&S) Programme may be considered a thoroughly revamped and modernized version of an existing University of Pune M.Tech. Programme in Modeling and Simulation. This existing M.Tech. programme used to be run in collaboration with the Institute of Armament Technology (IAT), Pune. The IAT, which is now called Defense Institute of Advanced Technology (DIAT), is a Deemed University. The Centre for Modeling and Simulation, University of Pune, has modified, broadened, and thoroughly modernized the scope this somewhat dated programme so as to keep it in pace with modern developments in computing technologies and computational methodologies, and to promote an interdisciplinary computational approach to science, technology, and other fields of the human endeavour where quantitative reasoning plays any role.

1.5 Outline of This Document

The rest of this document is organised as follows. Sec. 2 states the specific objectives and the expected outcome of the Programme. Sec. 3 is devoted to eligibility and selection criteria. Sec. 4 presents the academic structure and curriculum of the M.Tech.(M&S) Programme. Sec. A is devoted to acknowledgments and credits. Sec. I is devoted to implementation issues, recommendations, and guidelines. Finally, individual modulewise syllabi are included in Sec. M in alphabetical order.

2 Specific Objectives and Outcomes

Assuming design alternative (5) in Section 1.3.1, we expect a graduate of the M.Tech.(M&S) Programme to be the following after undergoing the training of the programme:

- She or he is a problem solver with a breadth and perspective on modeling, good training in computation and simulation methods, and the ability to generate reasonable solutions for problems not necessarily encountered earlier.
- Outside of his or her knowledge domain, (s)he is able to create a computer representation of a specific detailed description of the problem domain given that a domain expert has distilled the mathematical essence for him or her.
- She or he is up-to-date with the current state of relevant technologies, and from familiar to skilled in a variety of relevant software tools and methodologies.

3 The Prospective Student

Structure of the M.Tech.(M&S) Programme is built around the assumption that a prospective student satisfies these three principal prerequisites:

- Domain expertise at least at the level of a B.E., B.Tech. or equivalent degree in any technological knowledge domain, or a Masters degree in a science, commerce or arts discipline in which quantitative reasoning plays a major role;
- A demonstrable background in mathematics at the level of the current University of Pune F.Y.B.Sc. mathematics syllabus. This is to be assessed via an entrance examination and the academic record (see Section 3.2 below); and
- Basic computer usage skills preferably with some programming experience and algorithmic skills. This is to be assessed via an entrance examination and the academic record (see Section 3.2 below).

Furthermore, it is crucial to select candidates with the right background and aptitude so that they are able to cope up with the vigorous and intensive nature of this programme. We thus propose the following eligibility and selection criteria for a prospective candidate who desires admission in the programme.

3.1 Eligibility

1. B.E. or equivalent in any branch of engineering **OR** Master's degree in any science, arts, commerce discipline.

AND

2. Background in mathematics equivalent to the University of Pune F.Y.B.Sc. mathematics syllabus as ascertained via the applicable selection criterion below.

3.2 Selection

3.2.1 Regular Admissions

Selection Criteria. Candidates will be selected for regular admission in the programme on the basis of

1. Their academic record,

2. An entrance examination, and
3. A statement of purpose, if necessary.

The purpose of the entrance examination is to assess basic analytical, reasoning, and algorithmic skills, and the required mathematics background and computer literacy. Given the intensive and vigorous nature of the programme, it is essential to set the selection threshold on the entrance examination score appropriately, and to apply it to all candidates uniformly.

Number of Seats. Consistent with University of Pune rules, norms and policies on reservation, the Centre will select not more than 30 best or top-scoring candidates.

3.2.2 Industry-Sponsored Candidates

The Centre will welcome a small number of candidates sponsored by an industry or organisation every academic year. Interests of the sponsoring organisation in sponsoring a candidate could vary; however, we envisage the following as the most likely scenarios:

1. Primary interest of the sponsoring organisation and the candidate is in outsourcing the core training of the first 1.5 years (the first 3 semesters) of the programme, and the sponsored candidate is not necessarily interested in obtaining a degree certificate from University of Pune. In this case, a simple certificate of completion that includes the grade-point average (or some assessment of the performance of the candidate) issued by the Centre at the end of the first 1.5 years (the first 3 semesters) should suffice.
2. A sponsored candidate is indeed interested in completing the full programme and in obtaining a degree certificate from University of Pune at the end of the programme, and the sponsoring organisation is willing to support the candidate to that extent.

Therefore, the details of such admissions, including eligibility and selection, fee structure, *modus operandi* for the minimum 6-month project, etc., will need to be decided in consultation with the interested industry or organisation on a case-by-case basis without compromising on the spirit, intensity, and depth of the programme.

3.2.3 Foreign Candidates

The Centre will welcome a small number of applications from foreign candidates every academic year, consistent with the policies, norms, and guidelines of the University of Pune. Given that a centralized evaluation mechanism (similar to GRE, AGRE, etc.) for admission into Indian Universities does not exist, we propose to make a decision about admission of a foreign applicant in the programme on a case-by-case basis, so as to ensure that the background and abilities of the applicant are appropriate, and that the spirit, intensity, and depth of the programme is not compromised upon.

4 The Curriculum

Programme Content. As discussed in Sec. 1.3, design alternative 5 is the most suitable alternative for this programme. As a consequence, the curriculum of the M.Tech.(M&S) Programme focusses almost exclusively on methodologies from the mathematical modeling and simulation domains. Broadly speaking, practical and often-used mathematical modeling and simulation methodologies have been developed in the field of Applied Mathematics and Applied Statistics. Constructing a simulation on a modern digital computer, on the other hand, needs a detailed knowledge of computing technologies, computation in general, and, to make things work, programming. The M.Tech.(M&S) Programme curriculum can be thought of, to a good approximation, as a mixture in equal parts of Applied Mathematics, Applied Statistics, and Computing.

Timeline. The first 3 semesters (initial 1.5 years) of this programme are devoted to coursework consisting of 15 core courses and 2 specialized electives. In the fourth semester (second year) of the programme, a student is expected to undergo hands-on, rigorous, and application-oriented training involving substantial independent work in the form of a project (same as internship or industrial training). The required duration for a project is six months. However, more extensive and longer-than-six-months projects are allowed.

Curriculum Organisation. We have envisioned a hierarchical organisation of academic curricula in which an *academic programme* consists of *courses* which in turn consist of *modules*. A *module* is an instructional unit that consists of subject matter related to a single topic at the intended level for the target audience. In other words, a module is a logical entity of subject matter organisation that cannot be broken up further into smaller pieces in a meaningful fashion Modules can thus be shared, in principle, across academic programmes targeted at similar audiences. A *course* consists of one or more modules organised around a coherent theme. Course codes used in this document are of the form CMS-MT-yy-nnn, where yy stands for the last two digits of the academic year, and nnn is a unique number assigned to a course.

Organisation of this Section. This section presents a crisp overview of the academic structure of the M.Tech.(M&S) Programme in the full-time mode. In Sec. 4.1–4.4, we present a semesterwise summary of the curriculum: To highlight the structure, we have relegated detailed modulewise syllabi to Sec. M. The last column of the semesterwise tabular summaries indicates section numbers for individual modulewise syllabi and rationales. In Sec. 4.5, we present the rationale and outlook on each course-theme individually.

4.1 Structure at a Glance: Semester 1 (Year 1)

Course Name (Course Code) Module Name	**Credits** Contact Hours	**Rationale Section** Syllabus Section
Fundamentals (CMS-MT-yy-11)	**5CR**	**4.5.1**
1. Basics of Analysis	15CH	M.1.3
2. Vector Analysis	15CH	M.1.30
3. Complex Analysis	15CH	M.1.4
4. Linear Algebra	30CH	M.1.9
Computing I (CMS-MT-yy-12)	**5CR**	**4.5.2**
1. Programming for Modelers I	50CH	M.1.21
2. Practical Computing I	20CH	M.1.17
3. Finite-Precision Arithmetic	05CH	M.1.5
Perspectives on Mathematical Modeling I (CMS-MT-yy-13)	**5CR**	**4.5.3**
1. Introduction to `Matlab/Octave`	05CH	M.1.7
2. Ordinary Differential Equations	35CH	M.1.13
3. Partial Differential Equations	35CH	M.1.16
Perspectives on Probability Modeling I (CMS-MT-yy-14)	**5CR**	**4.5.4**
1. Introduction to `R`	05CH	M.1.8
2. Probability Theory	35CH	M.1.20
3. Reasoning Under Uncertainty	35CH	M.1.23
Technical Communication I (CMS-MT-yy-15)	**2CR**	**4.5.5**
1. Technical Reading, Writing, and Presentation	30CH	M.1.27
Overview of M&S I (CMS-MT-yy-16)	**3CR**	**4.5.6**
1. Overview of M&S I	45CH	M.1.14

4.2 Structure at a Glance: Semester 2 (Year 1)

Course Name (Course Code) / Module Name	Credits / Contact Hours	Rationale Section / Syllabus Section
Elective I (CMS-MT-yy-21)	**5CR**	**4.5.8**
1. Elective I	75CH	M.2
Computing II (CMS-MT-yy-22)	**5CR**	**4.5.2**
1. Programming for Modelers II	50CH	M.1.22
2. Practical Computing II	25CH	M.1.18
Perspectives on Mathematical Modeling II (CMS-MT-yy-23)	**5CR**	**4.5.3**
1. A Survey of Numerical Mathematics	37CH	M.1.2
2. Optimization	37CH	M.1.12
Perspectives on Probability Modeling II (CMS-MT-yy-24)	**5CR**	**4.5.4**
1. Stochastic Simulation	37CH	M.1.26
2. Statistical Models and Methods	38CH	M.1.24
Technical Communication II (CMS-MT-yy-25)	**2CR**	**4.5.5**
1. Term Paper	30CH	M.1.28
Overview of M&S II (CMS-MT-yy-26)	**3CR**	**4.5.6**
1. Overview of M&S II	45CH	M.1.15

4.3 Structure at a Glance: Semester 3 (Year 2)

Course Name (Course Code) Module Name	Credits Contact Hours	Rationale Section Syllabus Section
Elective II (CMS-MT-yy-31)	**5CR**	**4.5.8**
1. Elective II	75CH	M.2
Computing III (CMS-MT-yy-32)	**5CR**	**4.5.2**
1. Practical Computing III	50CH	M.1.19
2. High-Performance Computing	25CH	M.1.6
Perspectives on Mathematical Modeling III (CMS-MT-yy-33)	**5CR**	**4.5.3**
1. Numerical Linear Algebra	38CH	M.1.11
2. Transforms	38CH	M.1.29
Perspectives on Probability Modeling III (CMS-MT-yy-34)	**5CR**	**4.5.4**
1. Advanced Probability	37CH	M.1.1
2. Stochastic Optimization and Evolutionary Computing	38CH	M.1.25
M&S Hands-On (CMS-MT-yy-35)	**5CR**	**4.5.7**
1. M&S Hands-On	75CH	M.1.10

4.4 Structure at a Glance: Semester 4 (Year 2)

Course Name (Course Code) Module Name	**Credits** Contact Hours	**Rationale Section** Syllabus Section
Project (Internship/Industrial Training) (CMS-MT-yy-41) 1. Project	**25CR** 375CH	**4.5.9** M.3

4.5 Rationale for Individual Course Themes

4.5.1 Fundamentals

Course Code/s: CMS-MT-yy-11.

Rationale and Outlook. This group of modules serves to (a) to establish a common denominator for mathematical training keeping in mind the students' diverse and nonuniform mathematical training, (b) to equip the student to read through, and to be able to grasp, elementary mathematical treatment of specialized topics, (c) to cover common mathematics prerequisites of other modules, and (d) to generate a broad perspective on mathematical modeling and computational simulation.

4.5.2 Computing

Course Code/s: CMS-MT-yy-12, CMS-MT-yy-22, CMS-MT-yy-32.

Rationale and Outlook. A solid background in basic computing methodologies and software technologies, together with ample hands-on experience, is essential to be able to work with one's own hands and effectively utilize the training imparted in the rest of the Programme. Sec. I.2.4 contains an extensive discussion of pedagogic issues involved in introducing programming to novices; our pedagogic outlook is based on this discussion.

4.5.3 Perspectives on Mathematical Modeling

Course Code/s: CMS-MT-yy-13, CMS-MT-yy-23, CMS-MT-yy-33.

Rationale and Outlook. The utility of Applied Mathematics as a formalism for modeling needs no justification. This group of modules is intended to cover areas of Applied Mathematics most commonly encountered in modeling and simulation contexts.

In order to emphasize the modeling perspective and maintain focus on applications, we suggest a three-component structure for each of these modules; this structure consists of: (a) basic theory and pertinent mathematical results, (b) perspective on modeling by way of real-life contexts, examples and applications, and (c) relevant numerical methods.

At the discretion of the instructor, a tutorial on `matlab/octave` could be included in this course. Use of a computing environment such as `matlab/octave` has the advantage of letting a student without refined programming skills to focus on the mathematics and the modeling context of a topic. However, we recommend that at least a part of the computational exercises should be geared towards enhancing the student's programming skills in a low-level, imperative language such as `C`.

4.5.4 Perspectives on Probability Modeling

Course Code/s: CMS-MT-yy-14, CMS-MT-yy-24, CMS-MT-yy-34.

Rationale and Outlook. Probability is the mathematical language for quantifying uncertainty or ignorance, and is the foundation of statistical inference and all probability-based modeling.

In order to emphasize the modeling perspective and maintain focus on applications, we suggest a three-component structure for each of these modules; this structure consists of: (a) basic theory and pertinent mathematical results, (b) perspective on modeling by way of real-life contexts, examples and applications, and (c) relevant numerical methods.

At the discretion of the instructor, a tutorial on R could be included in this course. Use of a computing environment such as R has the advantage of letting a student without refined programming skills to focus on the mathematics and the modeling context of a topic.

4.5.5 Technical Communication

Course Code/s: CMS-MT-yy-15, CMS-MT-yy-25.

Rationale and Outlook. This group of courses aims at developing

1. resourcefulness, independence, self-study and time-management skills,
2. the ability to read technical and research literature, and finally,
3. proficiency in making effective technical presentations.

During the second course, a student is expected to take on an independent study of a topic that is not studied by him or her before, learning enough about it to be able to write a technical report (term paper) and make an oral presentation.

4.5.6 Overview of M&S

Course Code/s: CMS-MT-yy-16, CMS-MT-yy-26.

Rationale and Outlook. Since the rest of the curriculum focusses primarily on methodological aspects of M&S, it is highly desirable to include courses that attempt at developing a perspective on

1. the wide world of M&S as a whole,
2. the process of going from a problem to a simulation/computation via a model,
3. common pitfalls, sanity checks, etc., and
4. the "art" aspects of mathematical modeling by way of examples

These two courses are expected to strike a balance between generalities and specific examples, between stochastic and deterministic models, and between describing finished work and actual hands-on modeling on the part of a student. These two courses can be thought of as the heart of this programme, as they attempt to put the diversity of topics in the Programme curriculum in a unified perspective. Given the diversity of the M&S enterprise, these courses are expected to be pedagogically most demanding and a challenge to the instructor.

4.5.7 M&S Hands-On

Course Code/s: CMS-MT-yy-35.

Rationale and Outlook. This full 5-credit course is intended as a hands-on preparation for the 4-th semester project/internship/industrial training. Here, a student is expected to exercise his or her M&S skills by taking up a sufficiently complex real-life problem, making appropriate analysis and modeling decisions to come up with a of useful model of the underlying system, and eventually extract relevant information about the system possibly through the use of computation and simulation. This course is intended as a testing ground where qualities that matter for a modeler, namely, independence, resourcefulness, thoroughness, innovativeness in building (mathematical) models, and computational skills, are put to a thorough and rigorous test.

This course may be thought of as an in-house project. However, a project in close collaboration with an external organisation may be allowed with the approval of the Centre's academic team, after a realistic assessment of the feasibility of such a project.

4.5.8 Elective

Course Code/s: CMS-MT-yy-21, CMS-MT-yy-31.

Rationale and Outlook. Given the core training of the first two semesters, a student is expected to be in a position to assimilate material from advanced, specialized, or domain-specific elective modules of his or her choice. The choice and syllabi of elective modules offered may vary from year to year depending on the availability of specialist instructors. Elective modules could be organised around a variety of loosely-defined themes:

- **Methodology-Oriented Electives.** The focus here is on advanced methodologies or technologies. Examples are:
  1. High-Performance Computing.
  2. Machine Learning.
- **Formalism-Oriented Electives.** A formalism-oriented elective attempts to develop somewhat deeper understanding of a specific mathematical formalism. Examples of this kind are
  1. Advanced Probability Theory.
  2. Theory of Computation.
  3. Advanced Numerical Methods.
  4. Game Theory and Applications.
- **Domain-Oriented Electives.** A domain-oriented elective is one that develops domain-specific formalisms for modeling and simulation, and illustrates them via real-life problems, models, and applications. Examples of this kind are
  1. Computational Fluid Dynamics.
  2. Computational Materials Modeling.
  3. Modeling and Simulation of Biological Systems.
  4. Monte Carlo Methods in Finance.
  5. Evolutionary and Agent-Based Modeling in Social Sciences.
  6. Introduction to M&S for Complex Systems.
  7. Six Degrees of Separation: an Introduction to Networks.

Sample syllabi by for some of the above electives are included in Appendix M.2.

Policy on Failure in an Elective. Instructors for electives are likely to come from other departments on the campus, from non-University organisations, or from the industry. As such, instructors for electives may not be available for re-examinations in case a student fails in an elective. In general, such a student needs to opt afresh for an elective offered during the next academic year. We, however, suggest that (s)he may be given one chance, if possible, to be evaluated on the same elective (in which (s)he failed) during the upcoming semester.

4.5.9 Project (Internship/Industrial Training)

Course Code/s: CMS-MT-yy-41.

Rationale and Outlook. Given the highly applied nature of the M.Tech.(M&S) Programme, it is natural to emphasize substantial hands-on, independent work in the form of a project or industrial training. We have thus included a heavy, full-time project component of minimum 6-month duration. There is no fixed syllabus for this module; what is expected is substantial hands-on work in consultation with a project advisor on the chosen project topic.

Modus Operandi. Project advisor and the project topic needs to be approved by the academic team at the Centre for Modeling and Simulation, University of Pune. The project duration may be extended beyond the formal requirement of the M.Tech.(M&S) Programme (i.e., 6 months minimum) by mutual agreement between a student and his/her project advisor, and subject to approval by the Centre's academic team. Ideally, a student should begin his or her search for a project (and a suitable project advisor) in the second semester (first year) of the programme. Networking with the industry on part of the Centre needs to be initiated sufficiently early on. Project work should begin on day one of semester 4 (year 2) with full gusto.

Prerequisites. A student is allowed to go for a project in the fourth semester (second year) provided (s)he has at least a passing grade in at least 30CR worth of course work of the first year.

Evaluation. Evaluation of a project is based on (a) a written report, (b) a presentation cum open defense, and (c) continuous evaluation by the project advisor and the Centre's academic team throughout the project duration.

A Acknowledgments and Credits

• The structure of this Programme emerged out of close interactions, many informal discussions, unique individual perspectives, and a few formal meetings between a core group that consisted of Abhay Parvate, Sukratu Barve, and Mihir Arjunwadkar. Credit for the design, compilation, and the actual writing of most of this document goes to Mihir Arjunwadkar and Abhay Parvate. Dr. Prashant Gade and Prof. Dilip G. Kanhere, as accomplished senior members of the Centre's academic team, offered their viewpoints from time to time.

• We would like to thank Dr. William B. Sawyer (Chakraborty Software, Zürich, Switzerland) who spent part of his sabbatical as a Visiting Faculty at the Centre for Modeling and Simulation, University of Pune, during August–December 2006, for close interaction and discussions, and for sharing with us his fresh outlook on modeling, simulation, computation, and pedagogy. Some of his suggestions have been incorporated in the structure of the Programme.

• Historically, Dr. Abhijat Vichare (CFDVS, IIT Powai) deserves credit for defining our outlook on simulation and modeling. We would also like to thank him for permitting us to incorporate his writings *almost* verbatim in this document (as Sec. 1.1 and 1.3).

• We are deeply indebted to Prof. Anil P. Gore (Department of Statistics, University of Pune) for giving us a first-rate education in the art of statistical modeling. We would also like to thank Prof. S. G. Kunte, and Prof. Uttara V. Naik-Nimbalkar, Prof. Manohar B. Rajarshi, all at the Department of Statistics, University of Pune, for close interactions, discussions, and support.

• We would like to thank Dr. Ashutosh (Persistent Systems, Pune) and Dr. Mandaar Pande (Tech Mahindra, Pune) for their early feedback on the structure of this Programme from an industry perspective. Their suggestions have been incorporated in this document.

• We would like to thank all those who contributed individual modulewise syllabi (Appendices M.1 and M.2) for topics of their speciality or choice; their names appear at the end of their respective module/s, and short profiles in Sec. A.1.

• While designing individual modulewise syllabi, we have consulted numerous resources on the world-wide web. Oftentimes, we find that all the resources that we consulted are too numerous to acknowledge individually. We would like to thank their authors collectively for their spirit of openness. In return, we proliferate this spirit of openness by publishing this very document over the web.

• Last, but by no means the least, we are deeply indebted to John Gardner for his eloquence and depth of thought as reflected in the following quote. Our enthusiasm for detailed, meticulous work in the pedagogic domain has been sustained in part by this and similar insights from many great thinkers.

> *The society which scorns excellence in plumbing because plumbing is a humble activity and tolerates shoddiness in philosophy because it is an exalted activity will have neither good plumbing nor good philosophy. Neither its pipes nor its theories will hold water.*
>
> *John Gardner, Excellence, 1961*

A.1 Contributor Profiles

This section contains short profiles of those who contributed syllabi (Sec. M) for topics of their speciality or choice.

A.1.1 Mihir Arjunwadar

Reader at the Centre. Ph.D. in Physics (Computational Condensed Matter; 1996, University of Pune). After a brief postdoctoral stint at the University of Pune, Mihir held a research computing position for several years at the Statistics Department, Carnegie Mellon University, an experience that opened up his mind to the fascinating world of signal plus noise. Mihir's current research interests include development of computational methodologies for challenging problems, statistical data modeling and analysis, computational condensed matter and statistical physics, complex systems, and computational/systems biology.

A.1.2 Ashutosh

Ph.D. in Physics (Nonlinear Dynamics and Turbulence; 2001, University of Pune). Ashutosh is currently an Associate Technical Manager (Life Science Informatics), Persistent systems Pvt. Ltd. His research interests include evolution- and immune-system-based algorithms and their applications to life science problems.

A.1.3 Sukratu Barve

Lecturer at the Centre. M.Tech. in Materials Science (1995, IIT Mumbai), Ph.D. in Physics (General-Relativistic Quantum Field Theory; 2002, TIFR, Mumbai). Prior to joining the Centre as a faculty in 2006, Sukratu worked as a faculty at the Birla Institute of Technology and Science (Goa Campus), as a visiting scientist at the Albert Einstein Institüt Max Plank Institüt für Gravitationsphysik, Golm, Germany, and as a postdoctoral researcher at the Institute of Mathematical Sciences, Chennai, and the Physical Research Laboratory, Ahmedabad. Highly valued by us for his analytical and mathematical skills and his passion for teaching, he is the only oddball at the Centre who has resisted, so far, the urge to resort to computation using computers. Sukratu's research interests include general theory of relativity and quantum field theory on curved spacetime (areas in which he has 8 research publications in prestigious journals), nonequilibrium statistical physics, diffusion in zeolites, and granulation processes. Sukratu has an enormous interest in history (especially that of the city of Pune) and human geography.

A.1.4 Abhay Parvate

Lecturer and Programme Coordinator at the Centre. M.Sc. in Physics (1997, University of Pune). Abhay is a mathematical physicist in the making and a graduate student at the Department of Physics, University of Pune, close to finishing his Ph.D. thesis in the area of calculus on fractals. Abhay is the strong backbone that keeps the everyday operations related to our *Advanced Diploma Programme in Modeling and Simulation* on the right track. A first-class programmer by habit, Abhay's passions include system administration, software engineering, organisation of anything and everything that he happens to run into (people, disc and office space, computing resources, and knowledge), and singing in his office.

A.1.5 William B. Sawyer

Consultant to Chakraborty Software, Zürich, Switzerland, and Visiting Scientist at the Centre during August–December 2006. Dipl. Inf. Ing. in Computer Science Engineering (1987, ETH Zürich); M.S. in Scientific Computing/Computational Mathematics (1993, Stanford University); Doctor of Sciences in Mathematics (2006, ETH Zürich). After his M.S. degree, Will worked in the field of algorithm development for high-performance computers at the Swiss Center for Scientific Computing (CSCS, Manno, Switzerland) and at NASA Goddard Space Flight Center (Greenbelt, MD, USA). He returned part time to the ETH Zürich for doctoral studies in 2001, and completed his thesis "Efficient Numerical Methods for the Shallow Water Equations on the Sphere" in 2006. His main research interests are in modeling global atmospheric dynamics on supercomputers, and software frameworks for Earth Science applications. He has 11 peer-reviewed publications in journals such as Numerische Mathematik and International Journal for High Performance Computer Applications.

A.1.6 K. C. Sharma

After retiring as Chairman for Applied Mathematics, Defense Institute of Advanced Technology (formerly, the Institute of Armament Technology), Pune, Prof. K. C. Sharma is now a Professor Emeritus at the Bhaskaracharya Pratishthan, Pune, and a Visiting Faculty at the the Institute of Defence Scientists and Technologists, Pune, the Department of Space Sciences and the Centre for Modeling and Simulation, University of Pune. His research areas include fluid mechanics and flight dynamics, missile guidance, numerical mathematics, and computational fluid dynamics.

A.1.7 V. Sundararajan

Group Coordinator, Scientific and Engineering Computing Group, Centre for Development of Advanced Computing (C-DAC), Pune. Ph.D. in Physics (Computational Condensed Matter; 1989, Anna University, Chennai). After a brief stint of post-doctoral work at University of Pune, he worked as a lecturer there till March 1994. During this tenure, he visited International Centre for Theoretical Physics, Trieste, Italy for about 14 months between 1992-93. Since April 1994 he has been working at Centre for Development of Advanced Computing (C-DAC), Pune. Currently, he is the coordinator of the scientific and Engineering Computing Group which spearheads the high-performance computing applications on the PARAM machine developed by C-DAC. His major areas of research interest are protein structure prediction multiscale modeling and simulation, applied evolutionary computing, and high-performance computing. He has published about 40 research articles in well-respected journals.

A.1.8 Abhijat Vichare

Research Scientist at the Center for Formal Design and Verification of Software (CFDVS), Indian Institute of Technology (IIT), Mumbai. Ph.D. in Physics (Computational Materials; 2002, University of Pune). Deeply interested in theoretical computer science, programming languages,and compilers, Abhijat joined the Department of Computer Science, University of Pune, as a Lecturer after completing his Ph.D. research, where he came to be considered, by his students and colleagues alike, a highly skilled teacher with a great passion for pedagogy. At CFDVS, IIT, Mumbai, he is engaged in formulating a formal verification framework for the GNU Compiler Collection (GCC).

I Implementation: Issues, Ideas, Suggestions

I.1 Ground Realities of Indian Undergraduate Education

In our collective experience of past two decades or so as teachers, we observe that the great Indian undergraduate education system, on the average, serves to effectively curb independent thinking, self-study skills, resourcefulness, intellectual maturity, academic confidence, and the very motivation to learn with excellence. Academic excellence is often identified, wrongly, with performance in examinations that tend to assess mostly memorization skills of a student, and the true measures of academic excellence such as depth of understanding, originality, authenticity, creativity, and perseverance are systematically discouraged.

While the reasons for this deplorable situation could be traced all the way back to Indian primary education, it is generally agreed that the immediate cause for the situate is the disproportionate importance that end-semester or end-year examinations have come to attain, starting right from the secondary and higher secondary school certificate examinations. This is seen to encourage, on the average, unhealthy learning styles on part of the students. For example, mindless memorization instead of an understanding of the underlying principles of a topic is usually perceived by students as the most successful strategy for "getting through" an examination. To make the situation worse, students coming typically from rural areas unnecessarily feel hampered by their self-perceived lack of proficiency in English.

As a thin silver lining to this dark cloud, a small respectable minority of students somehow survives through the damages inflicted by the great Indian undergraduate education system. The reasons for this are not entirely clear to us, but very likely they are related to strong family values or a local culture/environment that encourage excellence, and perhaps because of excellent, dedicated teachers at various levels who took their job seriously.

By the time a student enters a post-graduate programme in a University, he or she is usually under the additional pressure of finding either a career or an assured well-paying job (the latter expectation is further inflated in times of economic boom such as the present one). To make things worse, it turns out that many students have never been exposed to the notions of excellence, authenticity, and commitment to quality, and are generally confused about their goals.

The purpose of a University is "to give the society what it *needs*, and not what it *wants*"[3]. While it could be argued that the effort to institute remedial measures for this social malady should ideally exist in the very structure and ethos of a University, we believe that, in its absence, independent effort at all levels should be encouraged.

On a completely different note, we have witnessed a considerable rise over the past decade in the proportion of students with stringent family commitments–children, in particular. This is clearly a reflection of the transitions in the urban social structure in the modern times. A common characteristic of such students is their strong motivation to learn and improve their qualification while trying to balance their personal lives, without much support from their extended families or the society at large. A University should ideally provide a positive support and moral reinforcement to such students. This will very likely require fundamental reforms in the very structure, ethos, and outlook of an average Indian University.

The Programme as outlined in this document, is a vigorous and demanding programme if implemented in its true spirit. We see the realities outlined above as the greatest confounds in the effective deployment of the Programme. In this section, we present ideas that emerged out of our own experiments at the Centre for Modeling and Simulation, University of Pune.

[3] Edsgar W. Dijkstra, *The Strengths of an Academic Enterprise.* Emphasis ours.

I.2 Pedagogy

I.2.1 General Outlook

Given the highly multidisciplinary, vigorous, and compact nature of the M.Tech.(M&S) Programme, we strongly recommend adopting the following outlook on pedagogy.

1. **Rigour vs. Clarity** • Rigour and clarity are not always synonymous: concepts need to be emphasized from a practical and application standpoint in place of tedious derivations and rigorous proofs[4].
2. **Effective Instruction** • The role of dedicated instructors with an outlook to excel in pedagogy in addition to their own specialized field of expertise, well-designed instruction and self-study material, effective presentation methods, and judicious use of modern educational technologies to assist pedagogy cannot be overemphasized.
3. **Classroom Instruction vs. Independent Work** • Organised and focussed work on part of a student is as necessary for effective learning as the pedagogic skills of the instructor. We recommend that a student invests, on an average, 1.5–2 hours of self-study (including assignments) for every hour of instruction, on the same material. Thus, e.g., assuming an average of 4 one-hour lectures per day, we recommend that a student puts in 6–8 hours of independent work every day, on an average.
4. **Education Vs. Training** • There is a need to distinguish between *education* and *training*: While *education* has a universal character, *training* is necessarily geared towards a specific niche, and is, by its very nature, of a spatio-temporally limited utility. There is no doubt that the *training* component of any academic programme is most important from a practical and short-term point-of-view. However, any programme that fails to propagate lasting or durable values to its students is bound to deviate from larger goals of *education*. Specifically, it is important to learn how to learn–especially so in the technology-driven fast-changing modern world. This point of view is most eloquently expressed by a modern visionary as follows:

   > *We don't even know what skills may be needed in the years ahead. That is why we must train our young people in the fundamental fields of knowledge, and equip them to understand and cope with change. That is why we must give them the critical qualities of mind and durable qualities of character that will serve them in circumstances we cannot now even predict.*
   >
   > *John Gardner, Excellence, 1961*

I.2.2 Student Evaluation

We strongly recommend the following for fair and judicious evaluation of students, and for maximizing the pedagogic impact of the curriculum:

1. Adequate weightage to both "theory" and "practice" aspects.
2. Periodic examinations designed to strongly discourage mindless memorizing, to encourage independence, resourcefulness, authenticity, original thought, and to bring about the real knowledge, abilities, and skills of the student.
3. Continuous evaluation that aims at accurately assessing the knowledge, abilities and skills acquired through a course module. This may include, at the discretion of the instructor, class participation.

[4] Quote adopted from the introduction of *All of Statistics* by Larry Wasserman (Springer-Verlag, 2004).

4. Encouraging regular and consistent work throughout a semester. Weightages to different evaluational entities such as class participation, assignments, mid-semester examinations, end-semester examinations, etc., should be such that this goal is achieved. The end-semester examination alone should not attain a disproportionately high importance.
5. Declaring module- and instructor-wise policies on grading and evaluation at the very beginning of a course module. Assuming high levels of domain and pedagogic expertise and depth on part of the instructor, we strongly believe in the freedom of the instructor to formulate his/her own evaluation policy as a means of encouraging innovation in teaching practices. Consistent with the learning goals of a module, a well-formulated grading and evaluation policy helps maximize the pedagogic impact of the module. Numerous resources on formulating best grading and evaluation policies are available on the web (see, e.g., http://depts.washington.edu/grading/plan/needs.html).

I.2.3 M&S Colloquia and Case-Study Sessions

The M.Tech.(M&S) Programme curriculum, by design, focusses almost exclusively on methodological aspects of modeling and simulation. It is highly essential to impart to the students a perspective on modeling and simulation that will bring together this diverse set of concepts into a unified view, both from conceptual and practical points of view. Domain knowledge at some level is assumed on part of the student and, depending on the elective, may be enhanced to some extent in the course of the M.Tech.(M&S) Programme. What is missed upon in the M.Tech.(M&S) Programme curriculum is a perspective on modeling and simulation field as a whole (and, specifically, the “art” aspects of modeling and simulation) that will bring together the diverse set of concepts, methodological or otherwise, into a unified view, both from conceptual and practical points of view.

To remedy this situation, we strongly recommend arranging colloquia, seminars, and case-study sessions (approximately once every two weeks) by experts from industrial or academic research, with the intention of generating a perspective on current real-life applications of modeling and simulation as a methodology. By a *seminar*, we mean a specialized and focussed presentation on a research topic. A *colloquium* is a relatively extended presentation that is aimed at a lay, non-expert, or mixed audience, is usually biased towards the pedagogic end (as opposed to the “research” end) of the spectrum, and does not assume–generally–any domain-specific expertise on part of the audience. A *case-study* is a focussed and somewhat longer (3-6 hours) session with a strong hands-on component, presented and coordinated by an expert. The purpose of the M&S colloquia, seminars, and case-study sessions is twofold:

1. To illustrate the art of (mathematical) modeling in practice.
2. To generate a perspective on, and increase awareness about, modeling and simulation as a methodology.

In our experience, such a perspective could be generated in two distinct ways:

1. By exposing the audience to a variety of applications and examples from diverse areas of science and technology that employ modeling and simulation as a methodology, and by imparting a clear understanding, at an appropriate level of detail or description, of
    - the underlying scientific or technological system or problem, and the complexities in it;
    - the modeling decisions that went into building an appropriate mathematical model for this system or problem;

- the (mathematical) complexity of this model, and possible ways of extracting useful information from the model;
- the need to resort to computation and simulation, and the complexities involved therein;
- what is learnt about the system or problem through modeling and simulation, and how well does the model corroborate with real life.

2. By presenting an overview, at an appropriate level of detail or description that depends on the audience, of one or more *open* problems in a field where modeling and simulation methodologies are likely to help.

In addition, a focussed tutorial on a specific computational tool or environment (e.g., the `ns-2` network simulator, the `OpenFOAM` environment for computational fluid dynamics, etc.) could also serve as a case-study session.

I.2.4 Introducing Programming to Novices

The number of programming languages that have evolved over the history runs in hundreds (the `Google` programming languages directory lists about 150 languages). Each language has its own purpose, domain, paradigm, conventions, etc. The number of modern programming concepts and paradigms that these languages contribute together is quite large, and a student of programming needs to know (or at least be aware of) many such concepts right from the beginning to become an effective and efficient programmer.

One cannot obviously teach a multitude of languages in order to cover these important concepts and paradigms in programming. Further, since programming and languages is a rapidly growing and changing technology, what is popular in the industry or academics today need not be so tomorrow. The choice of a programming language is also seen to vary widely across and within technological, scientific, and industrial sectors. Therefore, the first language needs to be chosen carefully, not only considering the popularity in a particular sector, but the extent to which it allows various programming concepts and paradigms to be implemented, illustrated, and explored.

This implies that choosing, as the first language, a programming language that insists on a particular paradigm too rigidly could lead to unnecessary dogmatism on part of the students, and a student may find it difficult to switch to other paradigms later on. Such a choice will also make it difficult to do justice to many other desirable programming concepts.

Other characteristics of programming languages that prove to be hurdles in learning common programming paradigms and useful programming concepts effectively and in enhancing algorithmic thinking skills in the students are, e.g., mundane details of manual dynamic memory management and pointers, lack of abstractions such as higher-order functions, and abstract data types or classes.

Further, in an academic programme that does not wish to alienate itself from the outside world at large, and has a limited time (e.g., 1–1.5 years) for the coursework, the look and feel of the first programming language should ideally be similar to the "popular" languages, so that so that learning such "popular" languages becomes easier for a student as and when needed.

Languages such as `C, C++, fortran, java, Eiffel, Haskell, Scheme` all have some of the problems mentioned above. On the other hand, it turns out that `Python`, with its small, clean, and clarity-enforcing syntax and expressive semantics, satisfies *most* of the criteria mentioned above, and avoids pitfalls associated with the choice of a single-paradigm language as the first language. Therefore, we have recommended `Python` as the first language (module *Programming for Modelers I*, Sec. M.1.21).

We are aware that no language is perfect. Specifically, it may be argued from a certain point of view that `Python` too has its own set of pitfalls such as lack of type declarations or an otherwise static type system and the interpreted nature of `Python`. However, the concepts which are covered in *Programming for Modellers I* (Sec. M.1.21) include many modern concepts from a wide spectrum of paradigms, and all of them can be satisfactorily illustrated through `Python`.

We would like to note that this is not a complete answer to the problem of teaching programming. Various people have various solutions, and even those who try to standardize on computing education leave the dilemma without any satisfactory resolution. For further discussion on this, please refer to, e.g.,

1. Linda Grandell, Mia Peltomäki, Ralph-Johan Back, and Tapio Salakoski, *Why Complicate Things? Introducing Programming in High School Using Python*, in: Denise Tolhurst and Samuel Mann (Ed.), Conferences in Research and Practice in Information Technology, **52** (Australian Computer Society, 2006). Associated website: `http://www.tucs.fi/publications/attachment.php?fname=inpGrPeBaSa06a.pdf`
2. `http://www.acm.org/education/curricula.html`.
3. `http://www.cs.washington.edu/homes/dickey/curricula/`.
4. *Computing Curricula 2001*, Journal on Educational Resources in Computing **1** (2001).

I.3 Student Support

While we have incorporated several mechanisms in the selection process that ensure that students admitted in the Programme have the right combination of background and capabilities, keeping up their morale, enhancing their motivation, and encouraging excellence requires some systematic effort and a work environment that is flexible yet serious.

I.3.1 24/7 Access to Facilities

We strongly recommend ready and extended availability of academic and computational resources on a 24/7 basis. This includes access to the internet, study, and library facilities.

I.3.2 Orientation

In our experience, it helps to organise an orientation session at the beginning of an academic year to give the freshers an idea about what to expect during the Programme, important milestones and the timeline, and our outlook on pedagogy.

I.3.3 Continuous Monitoring and Academic Counselling

Continuous monitoring of progress and academic counselling of every student on an individual basis helps to ensure quality of the outcome of the Programme. Individual problems identified early on are often much easier to deal with. An implementation of this recommendation requires considerable faculty involvement and time on a regular basis.

I.3.4 The Project

In our experience, planning for the fourth-semester project needs to begin fairly early on; i.e., during the second semester of the first year. For those who wish to take up a job after completing the programme, a project often serves as a convenient mechanism to evaluate a potential job candidate (from the perspective of the employer) or to evaluate the work environment (from the perspective of the job seeker). In any case, finding an appropriate project and an adviser to suit individual interests, capabilities, and preferences needs considerable effort on part of a student and considerable logistic support on part of the organisation as a whole.

I.3.5 The Logistics of Time Management

In our experience, it helps to decide the academic calendar for an academic year, in particular a schedule of judiciously selected holidays and academic breaks, well before the beginning of an academic year. We strongly recommend a five-day academic week and a schedule of not more than 20 hours of classroom instruction per week so as to avoid saturation on part of a student. Deployment of course modules needs to take into account prerequisites. The logistics of scheduling classes especially for guest and visiting teachers needs considerable faculty involvement and time.

I.3.6 Value Addition

On the technical side, a general perspective on the art of mathematical problem solving and plausible reasoning[5] will help a student in the programme immensely. This is best accomplished in the form of a focussed workshop conducted by a qualified mathematician with excellent pedagogic skills. We further recommend similar workshops or mini-courses on a variety of themes of enduring or current interest, such as algorithmic thinking skills, grid computing, scientific visualization, etc. We also strongly recommend encouragement for soft skills; namely, creative thinking, innovation, and creativity in problem-solving. This is again best achieved by organising a workshop conducted by a qualified psychologist with excellent pedagogic, communication, and people skills. Workshops or mini-courses on communication skills, time management, and job hunt are also recommended.

[5] In the sense of G. Polya's famous two-volume book, *Mathematics and Plausible Reasoning*.

M Modulewise Syllabi

M.1 Core Modules

M.1.1 Advanced Probability

Contact Hours: 37 **Prerequisites: M.1.20**

Syllabus

1. Probability as Measure. σ-algebras and Borel functions. Measures. Probability spaces, random variables and distribution functions. Integration theory. Expectation in a probability space. Characteristic functions. Independence. Convergence of random variables. The strong law of large numbers.
2. Stochastic Processes. *Please refer to the suggested text (Hoel, Port, and Stone) below.*
3. Information Theory. Shannon entropy and its many interpretations: entropy as uncertainty; axiomatic denition and Khinchin axioms; Shannon entropy as thermodynamic entropy; Shannon entropy as average surprise; entropy and yes-no questions; entropy and coding; applications. Joint and conditional entropy. Mutual information. Entropy of continuous variables: continuous entropy $\leftrightarrow$ discrete entropy; Kullback-Leibler divergence; Non-extensive information measures (Rényi, Tsallis, etc.).
4. Game Theory Basics. *Please refer to the the two game theory texts suggested below.*

Notes

- This syllabus as outlined above should be considered indicative of the focus of the module. Selection of topics and emphasis may be varied at the discretion of a competent instructor.

Suggested Texts and References

- G. R. Grimmett and D. R. Stirzaker, *Probability and Random Processes*. Oxford University Press (2001).
- P. G. Hoel, S. C. Port, and C. J. Stone, *An Introduction to Stochastic Processes*. Houghton Mifflin (1972).
  Associated website: http://www.waveland.com/Titles/Hoel-et-al.htm
- David P. Feldman, *Information Theory, Excess Entropy and Statistical Complexity: Discovering and Quantifying Statistical Structure*. Web-published (1997). Associated website: http://hornacek.coa.edu/dave/Tutorial/
- Chapter 1 of: David J. C. MacKay, *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press (2003). Available as a public web resource at http://www.inference.phy.cam.ac.uk/mackay/itprnn/book.html
- Stef Tijs, *Introduction to Game Theory*. Hindustan Book Agency (2003).
- Roger A. McCain, *Game Theory: A Non-Technical Introduction to the Analysis of Strategy*. South-Western College Publishers (2004).

Syllabus contributed by Mihir Arjunwadkar

M.1.2 A Survey of Numerical Mathematics

Contact Hours: 37 **Prerequisites: M.1.3, M.1.30, M.1.21, M.1.5**

Syllabus

1. Miscellaneous Topics in Numerical Mathematics. Topics of practical importance that are not covered elsewhere in the curriculum, such as: function approximation, interpolation, integration, nonlinear algebraic equations, integral equations.

Notes

- This syllabus as outlined above should be considered indicative of the focus of the module. Selection of topics and emphasis may be varied at the discretion of a competent instructor.

Suggested Texts and References

- H. M. Antia, *Numerical Methods for Scientists and Engineers*. Hindusthan Book Agency (second edition, 2002).
- William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling, *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press (second edition, 1992). Indian edition/reprint available. Available as a public web resource at `http://lib-www.lanl.gov/numerical/`

Syllabus contributed by Mihir Arjunwadkar

M.1.3 Basics of Analysis

Contact Hours: 15 **Prerequisites: None**

Syllabus

1. Introduction. Continuum of numbers. Concept of function. Elementary functions. Sequences. Mathematical induction. Limits.
2. Integral and Differential Calculus. Integral. Examples of integration. Rules of integration. Indefinite integral. Logarithm, exponential function, powers. Derivative. Fundamental theorems of calculus.
3. Techniques. Simplest rules. Derivatives of: inverse functions, composite functions, exponential, hyperbolic functions. Maxima and minima. Order of magnitude of functions. Method of substitution. Integration by parts. Integration of rational functions. Integration of some other classes of functions. Extension of the concept of integral. Differential equations of the trigonometric functions.
4. Taylor's Expansion. Introduction: power series. Expansion of the logarithm and the inverse tangent. Taylor's theorem. Expression and estimates for the remainder. Expansions of the elementary functions. Geometrical applications.
5. Infinite Sums and Products. The concepts of convergence and divergence. Tests for absolute convergence and divergence. Sequences of functions. Uniform and nonuniform convergence. Power series. Expansion of functions in power series; Method of undetermined coefficients. Power series with complex terms.

Suggested Texts and References

- Courant and John, *Introduction to Calculus and Analysis: Volume I*. Springer (1989).
- Arfken and Weber, *Mathematical Methods for Physicists*. Elsevier (2005).
- Erwin Kreyszig, *Advanced Engineering Mathematics*. Wiley (ninth edition, 2006).

Syllabus contributed by Abhay Parvate

M.1.4 Complex Analysis

Contact Hours: 15 **Prerequisites: M.1.3**

Syllabus

1. Complex Analytic Functions. Complex Numbers. Polar form of complex numbers, triangle inequality. Curves and regions in the complex plane. Complex function, limit, continuity, derivative. Analytic function. Cauchy-Riemann equations. Laplace's equation. Rational functions, roots, exponential function, trigonometric and hyperbolic functions, logarithm, general power.
2. Complex Integrals. Line integral in the complex plane. Basic properties of the complex line integral. Cauchy's integral theorem. Evaluation of line integrals by indefinite integration. Cauchy's integral formula. Derivatives of an analytic function.
3. Laurent Series. Review of power series and Taylor Series. Convergence. Uniform convergence. Laurent series, analyticity at infinity, zeros and singularities.
4. Complex Integration by Method of Residues. Analytic functions and singularities. Residues, poles, and essential singularities. The residue theorem. Contour integration and Cauchy residue theorem as techniques for real integration. Principal values of integrals. Branch cuts and integration. Illustrations using, e.g., Laplace, Fourier, and Mellin transforms and their inverses.

Suggested Texts and References

- Tristan Needham, *Visual Complex Analysis*. Oxford University Press (1999).
- Arfken and Weber, *Mathematical Methods for Physicists*. Elsevier (2005).
- Erwin Kreyszig, *Advanced Engineering Mathematics*. Wiley (ninth edition, 2006).

Syllabus contributed by Mihir Arjunwadkar

M.1.5 Finite-Precision Arithmetic

Contact Hours: 5 **Prerequisites: None**

Syllabus

1. Computer Representation of Numbers. Introduction, principles and issues. Representation of integers and properties of the representation. Overflow. Endianness.
2. Representation of Real Numbers. Rational numbers. Fixed-point and floating-point representations. Properties of floating-point numbers. Number and distribution of representable floating-point numbers. Overflow and underflow.
3. Finite-Precision Arithmetic with Floating-Point Numbers. How elementary arithmetic operations are performed. Subtraction and loss of precision. Illustrative examples.
4. A User's Perspective on IEEE 754. The IEEE 754 specification. All about `NaN`s and `Inf`s. Tracking floating-point exceptions. Outlook on IEEE 754.

Suggested Texts and References

- H. M. Antia, *Numerical Methods for Scientists and Engineers*. Hindusthan Book Agency (second edition, 2002).

- William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling, *Numerical Recipes in C: The Art of Scientific Computing.* Cambridge University Press (second edition, 1992). Indian edition/reprint available. Available as a public web resource at http://lib-www.lanl.gov/numerical/
- David Goldberg, *What Every Computer Scientist Should Know About Floating-Point Numbers.* Computing Surveys (March, 1991). Available as a public web resource at http://docs.sun.com/source/806-3568/ncg_goldberg.html

Syllabus contributed by Mihir Arjunwadkar

M.1.6 High-Performance Computing

Contact Hours: 25 **Prerequisites: M.1.22**

Syllabus

1. High-Performance Sequential Computing. Effects of the memory hierarchy. Out-of-order execution. Superscalar processors.
2. Supercomputer Architecture Basics and Parallel Computing Terminology. Vector machines. Parallel processors. Data parallel processors. Single-instruction-multiple-data. Multiple-instruction-multiple-data. Pipelining. Vectorization. Parallelization. Comparison of serial, parallel and vector architectures.
3. Programming for Parallel Computing. Shared memory parallel computers and programming with OpenMP. Distributed memory parallel computers and message passing basics with MPI. Advanced message passing with MPI.
4. Performance Measures and Models. Debugging and measuring parallel applications. Profiling and development tools. Speed-up limitations. Sustained versus peak performance. Amdahl's Law and extensions. Scaled speed-up. Pipeline speed-up.
5. Parallel Algorithms and Techniques. Parallelization of algorithms. Loop optimizations. Principal of locality. Caches and buffers. Data distribution in parallel applications, data dependency reduction, data flow, loop reordering, massively data parallel algorithms. Techniques for data-intensive and numerically-intensive applications. Purely parallel algorithms. Block decomposition methods. High-performance libraries and parallel programming packages.
6. Emerging Technologies: Grid Computing. Overview and history of grid computing. Major grids around the world. Computational grids, data grids. Grid clients and middleware: standardization, security, and related issues. Porting applications for the grid.

Notes

- Ideally, upon successful completion of this module a student should be able to: design a parallel algorithm for distributed or shared memory machines, implement a parallel algorithm on a collection of networked workstations using MPI, implement a parallel algorithm on supercomputers using MPI (distributed memory) and OpenMP (shared memory), analyze the performance of parallel algorithms using a variety of performance measures/tools, describe conceptually the various paradigms of parallel computing, and report on the diverse applications of parallel and high performance computing in diverse fields such as science, engineering, technology, finance, etc.

- This syllabus as outlined above should be considered indicative of the focus of the module. Selection of topics and emphasis may be varied at the discretion of a competent instructor.

Suggested Texts and References

- Hesham El-Rewini and Ted Lewis, *Distributed and Parallel Computing.* Manning (1998).
- Lloyd Fosdick, Liz Jessup, et al., *An Introduction to High Performance Scientific Computing.* MIT Press (1996).
- Peter Pacheco, *Parallel Programming with MPI.* Morgan Kaufmann (1997).
- Michael Quinn, *Parallel Computing: Theory and Practice.* McGraw-Hill (1994).
- Charles Severance and Kevin Dowd, *High Performance Computing.* O'Reilly (second edition, 1998).

Syllabus contributed by Mihir Arjunwadkar

M.1.7 Introduction to `Matlab/Octave`

Contact Hours: 5 **Prerequisites: None**

Syllabus

- The `matlab/octave` programming language and the `matlab/octave` computing environment.

Suggested Texts and References

- A large number of resources available over the internet.

Syllabus contributed by Mihir Arjunwadkar

M.1.8 Introduction to `R`

Contact Hours: 5 **Prerequisites: None**

Syllabus

- The `S` programming language and the `R` computing environment.

Suggested Texts and References

- A large number of resources available over the internet.
- W. N. Venables and B. D. Ripley, *Modern Applied Statistics with S-PLUS.* Springer (Fourth Edition, 2002). Associated website: `http://www.stats.ox.ac.uk/pub/MASS4/`

Syllabus contributed by Mihir Arjunwadkar

M.1.9 Linear Algebra

Contact Hours: 36 **Prerequisites: None**

Syllabus

1. Linear Equations. Introduction. Gaussian elimination and matrices. Gauss-Jordan method. Two-point boundary-value problems. Making gaussian elimination work. Ill-conditioned systems.
2. Rectangular Systems and Echelon Forms. Row echelon form and rank. The reduced row echelon form. Consistency of linear systems. Homogeneous systems. Nonhomogeneous systems. Electrical circuits.
3. Matrix Algebra. Addition, scalar multiplication, and transposition. Linearity. Matrix multiplication. Properties of matrix multiplication. Matrix inversion. Inverses of sums and sensitivity. Elementary matrices and equivalence. The LU factorization.
4. Vector Spaces. Spaces and subspaces. Four fundamental subspaces. Linear independence. Basis and dimension. More about rank. Classical least squares. Linear transformations. Change of basis and similarity. Invariant subspaces.
5. Norms, Inner Products, and Orthogonality. Vector norms. Matrix norms. Inner product spaces. Orthogonal vectors. Gram-Schmidt procedure. Unitary and orthogonal matrices. Orthogonal reduction. The discrete fourier transform. Complementary subspaces. Range-nullspace decomposition. Orthogonal decomposition. Singular value decomposition. Orthogonal projection. Why least squares? Angles between subspaces.
6. Determinants. Properties of determinants.
7. Eigenvalues and Eigenvectors. Elementary properties of eigensystems. Diagonalization by similarity transformations. Functions of diagonalizable matrices. Systems of differential equations. Normal matrices. Positive definite matrices. Nilpotent matrices and Jordan structure. The Jordan form. Functions of non-diagonalizable matrices. Difference equations, limits, and summability. Minimum polynomials and Krylov methods.
8. Perron-Frobenius Theory of Nonnegative Matrices. Introduction. Positive matrices. Nonnegative matrices. Stochastic matrices and Markov chains.
9. Function Spaces. Function spaces as vector spaces. Norms, inner products, orthogonality, eigenvalues and eigenvectors in function spaces. Differential and integral operators as linear operators. Special functions as bases. Applications in transforms and differential equations.

Suggested Texts and References

- Carl D. Meyer, *Matrix Analysis and Applied Linear Algebra*. Society for Industrial and Applied Mathematics (SIAM) (2000). Available as a public web resource at http://www.matrixanalysis.com/
- Gilbert Strang, *Introduction to Linear Algebra*. Wellesley-Cambridge Press (third edition, 1998).

Syllabus contributed by Mihir Arjunwadkar

M.1.10 M&S Hands-On

Contact Hours: 75 **Prerequisites: As decided by the Instructor/s**

Syllabus

- No fixed syllabus: Please refer to Sec. 4.5.7 for the outlook on this course.

Syllabus contributed by Mihir Arjunwadkar

M.1.11 Numerical Linear Algebra

Contact Hours: 38 **Prerequisites: M.1.3, M.1.30, M.1.9, M.1.21, M.1.5**

Syllabus

1. Matrix Multiplication Problems. Basic algorithms. Exploiting structure: algorithms for band and sparse matrices. Vectorization and reuse issues.
2. The Basics. Review of basic ideas from linear algebra. Vector and matrix norms. Finite precision issues. Orthogonality and the SVD. Projections and the CS decomposition. Sensitivity of square linear systems.
3. General Linear Systems. Triangular systems. LU factorization. Round-off analysis of Gaussian elimination. Pivoting. Improving and estimating accuracy.
4. Special Linear Systems. LDM^T and LDL^T factorizations. Positive definite systems. Banded systems. Symmetric indefinite systems. Block systems. Vandermonde systems and the FFT. Toeplitz and related systems.
5. Orthogonalization and Least Squares. Householder and Givens matrices. QR factorization. Full-rank LS problem. Other orthogonal factorizations. Rank-deficient LS problem. Weighing and iterative improvement. Square and underdetermined systems.
6. Parallel Matrix Computations. Basic concepts. Matrix multiplication. Factorizations.
7. The Unsymmetric Eigenvalue Problem. Properties and decompositions. Perturbation theory. Power iterations. Hessenberg and Real Schur forms. The practical QR algorithm. Invariant subspace computations. The QZ method for $Ax = \lambda Bx$.
8. The Symmetric Eigenvalue Problem. Properties and decompositions. Power iterations. The symmetric QR algorithm. Jacobi methods. Tridiagonal methods. Computing the SVD. Generalized eigenvalue problems.
9. Lanczos Methods. Derivation and convergence properties. Practical Lanczos procedures. Applications to $Ax = b$ and least squares. Arnoldi and unsymmetric Lanczos.
10. Iterative Methods for Linear Systems. Standard iterations. Conjugate gradient method. Preconditioned conjugate gradients. Other Krylov subspace methods.
11. Functions of Matrices. Eigenvalue methods. Approximation methods. The matrix exponential.
12. Special Topics. Constrained least squares. Subset selection using SVD. Total least squares. Computing subspaces using SVD. Updating matrix factorizations. Modified/structured eigenproblems.

Notes

- This syllabus as outlined above should be considered indicative of the focus of the module. Selection of topics and emphasis may be varied at the discretion of a competent instructor.

Suggested Texts and References

- Gene H. Golub and Charles F. van Loan, *Matrix Computations.* The Johns Hopkins University Press (third edition, 1996).
- Jonathan Richard Shewchuck, *An Introduction to the Conjugate Gradient Method Without the Agonizing Pain.* Carnegie Mellon School of Computer Science Technical Report CS-94-125 (1994).
  Associated website: `http://historical.ncstrl.org/tr/ps/cmucs/CMU-CS-94-125.ps`

Syllabus contributed by Mihir Arjunwadkar

M.1.12 Optimization

Contact Hours: 37 **Prerequisites: M.1.3, M.1.30, M.1.9, M.1.22**

Syllabus

1. Computing Guidelines. Coding style recommendations for efficient large-scale numerics: code organization, modularity and reusability; readability and self-explanatory documentation; making assumptions explicit; printing floating-point numbers. `GSL, BLAS` and `LAPACK`. Visualization in 2D and 3D: contours, surfaces, isosurfaces; visualization tools.
2. Preliminaries. Why optimize? A survey of optimization problems and their modeling contexts; system/behaviour $\rightarrow$ model or formulation as an optimization problem, the objective function, domain of the objective function, applicable optimization methods. Some terminology (with lots of pictures): a minimizer, local and global minima, constrained and unconstrained minimization,
3. One Dimension. Derivatives, conditions for extrema, and the Taylor series. Numerical methods without derivatives: general structure, parabolic interpolation (and its connection with the secant method), golden section search, two-point bracketing and bisection, Brent's method. Numerical methods that use derivative information: Newton-Raphson, Davidon's method, Brent with bracketing. How close to a minimum is numerically close enough? Comparison of, and perspective on, 1D methods.
4. Unconstrained Minimization in More Than One Dimension: Generalities. Partial derivatives and conditions for order-independence. Gradient, Hessian, and directional derivative: properties and interpretation. Taylor expansions in $d = 2$ and in $d = n$. Extrema in N dimensions: necessary conditions for an extremum, extreme values and Taylor expansions, quadratic models, geometry of symmetric bilinear forms, Hessian eigenvalues and eigendirections.
5. Unconstrained Minimization in More Than One Dimension: Methods. Ad hoc methods: a simplex-based method, method of alternating variables. General structure of commonly-used methods. Steepest descent: rationale, algorithm, and convergence behaviour. Using second-derivative information: Newton method and its convergence behaviour; where and why Newton fails; quasi-Newton methods: rationale and generalities; BFGS and DFP. Direction set methods: basic ideas, Powell's method, conjugate directions and quadratic termination, conjugate gradient method.
6. Constrained Minimization. Linear programming and the simplex method. Equality and bound constraints: general theory. A survey of constrained minimization methods for nonlinear problems.

7. Deterministic Global Minimization. Survey of global minimization problems and deterministic global minimization methods. Comparison with stochastic methods.

Suggested Texts and References

- R. Fletcher, *Practical Methods of Optimization*. John Wiley & Sons (second edition, 2004).
- H. M. Antia, *Numerical Methods for Scientists and Engineers*. Hindusthan Book Agency (second edition, 2002).
- William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling, *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press (second edition, 1992).. Indian edition/reprint available. Available as a public web resource at http://lib-www.lanl.gov/numerical/
- Jacob Barhen, Vladimir Protopopsecu, and David Reister, *TRUST: A Deterministic Algorithm for Global Optimization.*. Science **276**, 1094–1097 (1997). Associated website: http://dx.doi.org/10.1126/science.276.5315.1094

Syllabus contributed by Mihir Arjunwadkar

M.1.13 Ordinary Differential Equations

Contact Hours: 35 **Prerequisites: M.1.3, M.1.30, M.1.4, M.1.5, M.1.7**

Syllabus

1. A brief introduction to Functional Analysis.
2. Classification of ODEs. Order. Linearity. Variable/constant coefficients.
3. General analytic solutions for first order differential equations with variable coefficients, and higher-order linear ODEs with constant coefficients.
4. Higher-order differential equations in brief. Their connection to first-order systems of ODEs.
5. The stability of 2-D systems of first order linear ODEs linearization of non-linear systems. Liapunov functions and their use in determining stability of general ODEs.
6. The numerical initial value problem for n-dimensional systems. Numerical methods. Classification into single-/multi-step, explicit/implicit. Definition of the order of the method. Definition of the method's stability region. Predictor/corrector methods and the use of the error predictor inherent to these methods. Adams-Moulton schemes (implicit and explicit) Runge-Kutta schemes along with their derivation for second and third order. Advantages and disadvantages of these methods. Stiff ODEs and their inherent problems: stiffness, definition of absolute and stiff stability, several methods particularly suited for stiff ODEs such as certain extrapolation methods (e.g. Gear's method).
7. The definition of the ODE boundary value problem. Numerical methods suited for its solution. The shooting method (especially for non-linear problems) and finite difference methods (predominantly for linear problems).
8. A brief introduction to expansion methods (finite element and spectral methods).

Suggested Texts and References

- H. M. Antia, *Numerical Methods for Scientists and Engineers*. Hindusthan Book Agency (second edition, 2002).

- Earl A. Coddington, *An Introduction to Ordinary Differential Equations*. Prentice-Hall India (1968).

Syllabus contributed by William B. Sawyer

M.1.14 Overview of M&S I

Contact Hours: 45 **Prerequisites: None**

Syllabus

1. Describing Reality Through Models. Deduction, induction, abduction (See Haefner in Suggested Texts and References below). Reality and model: the proverbial six blind men and the elephant. Models as simplified descriptions of reality relevant to the problem being addressed. Quantitative reasoning and mathematical modeling. Mathematical modeling and the scientific method: falsifiability, physical theories as models, parsimony and Occam's razor. Mathematical modeling and the practice of engineering (See Bender in Suggested Texts and References below).
2. The Life Cycle of a Model. Problem = a question about a system waiting to be answered. The modeling life cycle: problem → model → infer behaviour of the model → validation against reality → refined model → ... Formulation of a model: abstraction, qualitative vs. quantitative, stochastic vs. deterministic, etc. Deciphering the behaviour of the model: analytical reasoning and mathematical analysis, approximation, numerical computation, simulation: computational and non-computational. Handy tools for mathematical modeling: dimensional analysis, scaling, conservation laws and balance principles, ... Parsimony and Occam's razor: choosing the best from multiple models for the same problem. The dark side: misuses of models (See Bender in Suggested Texts and References below).
3. Case Studies in Modeling. A wide range of case-studies from diverse fields of science and technology that illustrate (a) the spirit of mathematical modeling and the "art" aspects of the modeling process, (b) the use of a number of formalisms of applied mathematics and applied statistics (deterministic and probabilistic), and (c) the problem-oriented outlook and multidisciplinary approach.

Notes

1. Please see Sec. 4.5.6 for the rationale of this and the companion course Overview of M&S II. This duo is expected to be pedagogically the most challenging pair of courses in this programme. As such, these courses would need to be handled in the most delicate fashion by the instructor. Substantial preparation would be needed given that these two courses are expected to convey the *spirit* of mathematical modeling in the most *coherent* manner, through a proper mix of discussion on principles with aptly chosen case-studies from diverse fields, thus putting the methodological training of the rest curriculum in a unified perspective.
2. The syllabus above is only *indicative* of the topics to be discussed in these two courses. Arguably, two instructors with distinct backgrounds may wish to accomplish the goals of these two courses following very different pedagogic paths.
3. A huge amount of useful literature on M&S is available in the form of books, lecture notes, tutorials, etc. The list of Suggested Texts and References below is again only indicative of its diversity.
4. The division of course material across these two courses is again left to the instructor.

Suggested Texts and References

- Clive L. Dym, *Principles of Mathematical Modeling*. Elsevier (second edition, 2004). Indian edition/reprint available.
- Edward A. Bender, *An Introduction to Mathematical Modeling*. Dover Publications (2000).
- Walter J. Meyer, *Concepts of Mathematical Modeling*. Dover Publications (2004).
- Rutherform Aris, *Mathematical Modeling Techniques*. Dover Publications (1994).
- Reinhard Illner, C. Sean Bohun, Samantha McCollum, Thea van Roode, *Mathematical Modelling: A Case Studies Approach*. American Mathematical Society (2005).
- Anil P. Gore, *Lecture Notes for the module* Statistical Models and Methods, available at http://cms.unipune.ernet.in/programmes/2005-06/modules/06-1-Statistical-Models-and-Methods/resources/
- K. Borovkov, *Elements of Stochastic Modelling*. World Scientific (2003).
- Sheldon M. Ross, *Introduction to Probability Models*. Academic Press (ninth edition, 2003).
- Samuel Karlin and Howard M. Taylor, *An Introduction to Stochastic Modeling*. Academic Press (third edition, 1998).
- James W. Haefner, *Modeling Biological Systems: Principles and Applications*. Springer (second edition, 2005).

Syllabus contributed by Mihir Arjunwadkar

M.1.15 Overview of M&S II

Contact Hours: 45 **Prerequisites: M.1.14**

Notes

1. See the syllabus for the sister course Overview of M&S I (Sec. M.1.14) for a detailed discussion of this pair of courses. The division of this suggested course material across these two courses is again left to the instructor.

Syllabus contributed by Mihir Arjunwadkar

M.1.16 Partial Differential Equations

Contact Hours: 35 **Prerequisites: M.1.13**

Syllabus

1. Classification of PDEs: hyperbolic, parabolic, elliptic; stationary, non-stationary; Eulerian, Semi-Lagrangian formulations; initial- and boundary-value problems.
2. Hyperbolic problems, one-dimensional advection equation. Simple finite-difference methods, stability and convergence using Von Neumann's (Fourier's) method.
3. First order, second order, and higher order schemes in space and time.
4. Solving parabolic and elliptic problems using finite differences. The linear wave, heat diffusion, and Laplace equations. Various alternative techniques, such as the Method of Lines.
5. A brief introduction to finite volume methods, conservative methods, the shallow water and Euler equations.

6. Analytic solutions to linear PDEs using the method of separation of variables and Fourier analysis.

Notes

- This module deals primarily with the numerical treatment of partial differential equations (PDEs).

Suggested Texts and References

- The course text "Numerical Solution of Partial Differential Equations", given in the Fourth Year of Honours Mathematics at the University of Glasgow, contains much of the material given in this module. In particular, there is a good treatment of the Fourier Method for stability analysis (referred to in this module as the Von Neumann Method).
- An good discussion of Finite Difference methods for PDEs can be found in Chapter 3 of the excellent on-line book by Nick Trefethen: "Finite Difference Methods". This text has not yet been completed by the author, so there are many omissions.
- The lectures on the numerical treatment of PDEs can be based on Chapter 13 of H. M. Antia, *Numerical Methods for Scientists and Engineers*. Hindusthan Book Agency (second edition, 2002).
- The lectures on analytic solutions to PDEs can be based on Richard Haberman, *Elementary Applied Partial Differential Equations*. Prentice-Hall, Inc. (Second Edition, 1987). This is an outstanding introduction to the method of Separation of Variables, Fourier Analysis, the treatment of linear PDEs for various domains, Green's functions, etc.

Syllabus contributed by William B. Sawyer

M.1.17 Practical Computing I

Contact Hours: 20 **Prerequisites: None**

Syllabus

1. `Linux` Preliminaries. Logging in and out. Passwords. Files and directories; path and filename conventions. Controlling access to files and directories. Shell and shell facilities. Environment variables. Emails and the internet; etiquette and guidelines on acceptable use. Various editors; in particular `vi` and `emacs`. Shell scripting: creating your own commands. Tasks of an operating system.

2. `Linux` Tools. Inventory of commands. Manual pages, info pages, and other resources. Finding the right command/tool for the given job. Standard input, standard output, and standard error. Redirecting and piping.

Suggested Texts and References

- A variety of online resources such as Unix/Linux `man` and `info` pages, internet resources, etc.

Syllabus contributed by Abhay Parvate

M.1.18 Practical Computing II

Contact Hours: 25 **Prerequisites: M.1.17**

Syllabus

1. Programming in the `Linux` Environment. Compilation and interpretation: Stages, units (source files, object files, etc.), libraries. Distinction between standards and implementations. Running programs in background, nice values. Scripting Tools and Languages (`awk, sed`, etc.).
2. `Linux` Processes. Programs vs. processes. Permissions and access control. `fork` and `exec`. Foreground and background. Signals. Boot-up sequence, kernel, and `init`.

Suggested Texts and References

- A variety of online resources such as Unix/Linux `man` and `info` pages, internet resources, etc.

Syllabus contributed by Abhay Parvate

M.1.19 Practical Computing III

Contact Hours: 50 **Prerequisites: M.1.22**

Syllabus

- `Java` for Modelers.
- The `Eclipse` Development Environment.
- The `.NET` Framework. `.NET` and `Mono`, `.NET` vs. `Java`.

Suggested Texts and References

- Texts/references to be recommended by the instructor.
- `Eclipse` project website: http://www.eclipse.org/
- `Mono` project website: http://www.mono-project.com/
- `.NET` developer's centre: http://msdn.microsoft.com/netframework/

Syllabus contributed by Abhay Parvate

M.1.20 Probability Theory

Contact Hours: 35 **Prerequisites: M.1.3, M.1.8**

Syllabus

1. Probability. Sample spaces and events. Probability on finite sample spaces. Independent events. Conditional probability. Bayes' theorem.
2. Random Variables. Distribution functions and probability functions. Important discrete and continuous random variables. Bivariate and multivariate distributions. Independent random variables. Conditional distributions. Important multivariate distributions. Transformations on one or more random variables.
3. Expectation. Properties. Variance and covariance. Expectation and variance for important random variables. Conditional expectation. Moment generating functions.

4. Inequalities for Probabilities and Expectations.
5. Convergence and Limit Theorems. Notion of convergence for random variables. Types of convergence. Law of large numbers, central limit theorem, the delta method.

Suggested Texts and References

- David Stirzacker, *Elementary Probability*. Cambridge University Press (1994). Indian edition/reprint available.
- Geoffrey R. Grimmett and David R. Stirzacker, *Probability and Random Processes*. Oxford University Press (2001).
- Larry Wasserman, *All of Statistics*. Springer-Verlag (2004).

Syllabus contributed by Mihir Arjunwadkar

M.1.21 Programming for Modelers I

Contact Hours: 50 **Prerequisites: None**

Syllabus

1. Introduction to `Python`. The interpreter. Simple expressions and calculations. More complex data types (e.g. Strings, Tuples, Lists, Dictionaries). Expressions involving complex data types. List comprehension. Interconversions between various data types.
2. The Functional Paradigm. Mathematical Basics: Sets, Relations and Functions. Relations and Functions as Subsets of Cartesian Products. Correspondence between Sets and Types. Function Definitions corresponding to Mathematical Functions. Recursion: Tail and otherwise. Iteration and Recursion. Functions as Types. Higher Order Functions: Functions as Parameters and Return Values. Type Analysis. Developing Functions to handle complex operations or data types. Proofs and Verifications of Correctness. Induction Technique.
3. The Imperative Paradigm (IP). History and need. Computation as a composition of Actions and Functions. Need to separate the two (Command-Query-Separation). Control Structures: Conditionals, Loops, etc. Structured Programming: Procedures (as different from Functions). Emphasis on Divide-and-Rule.
4. Data Structures and Algorithms. Lists, Dictionaries, Stacks, Queues, Priority Queues, Heaps. Specification, Implementations, and Common Applications. The Asymptotic (O) Notation. Search Trees. Sorting. Greedy Methods, Divide-and-Conquer, Dynamic Programming. Graphs. Network Flow and Matching. NP-Completeness.
5. IP Continued: Modular Programming. The Problem of Software Engineering: Combining small units into a large one. Concept of a Module. Namespaces. Scope Rules. Private and Public Data. Separation of Implementation from the Interface. Application Programming Interface (API).
6. IP Continued: Object Oriented Programming. Abstract Data Types (ADT) as mathematical Specifications. Interface-Implementation Separation as applied to ADT. Implementing ADT using classes. Analogy between Sets and Classes, and between Elements and Objects. Classes as Types. Software as a Collection of Interacting Objects.
7. More on Object Oriented Programming. Finding commonalities in various classes. The concepts of Inheritance and Polymorphism. Derived Classes as Subsets. Code Reuse; Avoiding Redundancy. Exceptions. Genericity. Design by Contract.

Suggested Texts and References

- Online `Python` documentation at http://www.python.org/doc/
- Abelson and Sussman, *Structure and Interpretation of Computer Programs*. The MIT Press (1996).
- Aho, Hopcroft, and Ullman, *Design and Analysis of Algorithms*. Addison-Wesley (1976).
- Meyer, *Object-Oriented Software Construction*. Prentice-Hall (second edition, 1997).

Syllabus contributed by Abhay Parvate

M.1.22 Programming for Modelers II

Contact Hours: 50 **Prerequisites: M.1.21, M.1.5**

Syllabus

- A programming language and/or environment suitable for modeling and simulation, and commonly used in industry or academics. Choice during a particular academic year depends upon current industry standards, preferences of the students, and availability of instructors. A few possible choices are `C`, `C++`, `C#`, and `Fortran`.

Suggested Texts and References

- As recommended by the instructor for this module and the specific language.

Syllabus contributed by Abhay Parvate and Mihir Arjunwadkar

M.1.23 Reasoning Under Uncertainty

Contact Hours: 35 **Prerequisites: M.1.20**

Syllabus

1. Models, Statistical Inference and Learning. Parametric and nonparametric models. Fundamental concepts in inference: point estimation, confidence sets, hypothesis testing.
2. Estimating the CDF and Statistical Functionals. The empirical distribution function. Statistical functionals.
3. The Bootstrap. Introduction. Simulation strategy. Bootstrap variance estimation. Bootstrap confidence intervals. Jackknife. Justification for the percentile interval.
4. Parametric Inference. Parameter of interest. The method of moments. Maximum likelihood. Properties of maximum likelihood estimators. Consistency of maximum likelihood estimators. Equivariance of the MLE. Asymptotic normality. Optimality. The delta method. Multiparameter models. The parametric bootstrap. Checking assumptions. Computing maximum likelihood estimates and the EM algorithm.
5. Hypothesis Testing and p-Values. The Wald test. p-values. The χ^2 distribution. Pearson's χ^2 test for multinomial data. The permutation test. The likelihood ratio test. Multiple testing. Goodness-of-fit tests. The Neyman-Pearson lemma. The t-test.
6. Bayesian Inference. The Bayesian philosophy. The Bayesian method. Functions of parameters. Simulation. Large sample properties of Bayes procedures. Flat priors, improper priors and "noninformative" priors. Multiparameter problems. Bayesian testing. Strengths and weaknesses of Bayesian inference.

7. Statistical Decision Theory. Comparing risk functions. Bayes estimators. Minimax rules. Maximum likelihood, minimax and Bayes. Admissibility. Stein's paradox.

Suggested Texts and References

- Larry Wasserman, *All of Statistics*. Springer-Verlag (2004).
- Morris deGroot and Mark Schervish, *Probability and Statistics*. Addison-Wesley (third edition, 2002).
- John E. Freund, *Mathematical Statistics*. Prentice-Hall of India (fifth Indian reprint, 1998).. Indian edition/reprint available.

Syllabus contributed by Mihir Arjunwadkar

M.1.24 Statistical Models and Methods

Contact Hours: 38 **Prerequisites: M.1.23**

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 tradeoff. 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 and References

- Larry Wasserman, *All of Statistics*. Springer-Verlag (2004).

- Morris deGroot and Mark Schervish, *Probability and Statistics*. Addison-Wesley (third edition, 2002).
- John E. Freund, *Mathematical Statistics*. Prentice-Hall of India (fifth Indian reprint, 1998).. Indian edition/reprint available.
- G. E. Box, G. M. Jenkins and G. Reinsel, *Time Series Analysis: Forcasting and Control*. Prentice-Hall (third edition, 1994).
- H. Kantz and T. Schreiber, *Nonlinear Time Series Analysis*. Cambridge University Press (second edition, 2004).

Syllabus contributed by Mihir Arjunwadkar

M.1.25 Stochastic Optimization and Evolutionary Computing

Contact Hours: 38 **Prerequisites: M.1.20, M.1.22, M.1.26**

Syllabus

1. Introduction. What is optimization? Problem categories. Comparison of different methods. Discussion of two example problems in optimization; viz. structure optimization of atomic clusters, and the travelling salesman problem (TSP).
2. Monte Carlo Methods. Detailed discussion of random walk, simple Monte Carlo, Metropolis Monte Carlo, and simulated annealing for the two problems. Comparison of these methods in terms of dynamics and acceptance criteria.
3. Evolutionary Strategies. Concepts of population, evolution, mutation. Two-membered, multi-membered populations. Methods with recombination.
4. Genetic Algorithms. Schema theorem. Building-block hypothesis. A simple genetic algorithm with selection, crossover and mutation. Parallelism and other variations.
5. Ant-Colony Optimization. Ants finding the shortest path and pheromone trails. Ant-colony algorithms. Application to TSP.

Notes

- This syllabus as outlined above should be considered indicative of the focus of the module. Selection of topics and emphasis may be varied at the discretion of a competent instructor.

Suggested Texts and References

- J. C. Spall, *Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control*. John Wiley and Sons (2003). http://www.jhuapl.edu/ISSO/
- P. J. M. Van Laarhoven and E. H. L. Aarts, *Simulated Annealing: Theory and Applications*. Kluwer Academic Publishers (1987).
- S. Kirkpatrick, C. D. Gelatt Jr, and M. P. Vecchi, *Optimization by Simulated Annealing*. Science **220**, 671–680 (1983).
- Maarten H. van der Vlerk, *Stochastic Programming Bibliography* (1996-2003). Associated website: http://mally.eco.rug.nl/spbib.html
- Swarm Intelligence Resources: http://www.swarmintelligence.org/, http://clerc.maurice.free.fr/pso/.

Syllabus contributed by V. Sundararajan

M.1.26 Stochastic Simulation

Contact Hours: 37 **Prerequisites: M.1.20, M.1.21, M.1.5**

Syllabus

1. A Tutorial on R.
2. Randomness. When do we perceive a sequence of objects as being random? Randomness, complexity, unpredictability, and ignorance. Cognitive biases. Randomness in natural processes: decaying nuclei, chaotic oscillators, leaky faucets, cosmic ray showers, . . .
3. Simulating Randomness. Generating sequences of numbers that appear random. Uniform pseudo-RNGs: properties, breaking correlations by shuffling, the Mersenne Twister, simple transformations from the Uniform. Other distributions as transformations from the Uniform: transformations of random variables, strictly monotone transformations of the Uniform, graphical illustration of the transformation method, exponential, Cauchy, ..., $N(0, 1)$ using the Box-Müller method, χ^2_p from $N(0, 1)$. Arbitrary distributions and acceptance-rejection sampling: sampling the area below a curve uniformly, $N(0, 1)$ using Cauchy as the envelope. Testing for randomness: how random is random enough? RNGs based on natural processes. Correlated random numbers: normal random numbers with prespecified correlations, Nataf transformation.
4. Monte Carlo Integration. Estimating π and $\log(x)$ using a dartboard. Estimating one-dimensional integrals: basic MC integration. Importance sampling: better estimators, tighter errorbars.
5. Deterministic vs. Stochastic. Behaviour of the error in d-dimensions: deterministic vs. stochastic. Estimating volume of a d-dimensional hypersphere.
6. More Than One Dimension. A survey of problems involving high-dimensional distributions, expectation values, and simulations. An operational introduction to Gibbs and Metropolis-Hastings Algorithms: an example/illustration of Gibbs sampling, Cauchy RNs Using $N(\cdot, b^2)$ as the proposal density. Markov Chains, their properties, and limit theorems. Why Metropolis-Hastings works: master equation and detailed balance, Metropolis-Hastings satisfies detailed balance, relationship between Metropolis-Hastings, Metropolis, and Gibbs, relationship between Metropolis and rejection sampling. Practical issues: the adjustable step length parameter, burn-in or equilibration behaviour, correlations and error bars on estimates. Optimization by simulated annealing.

Suggested Texts and References

- Hoel, Port, and Stone, *Introduction to Stochastic Processes*. Houghton Mifflin (1972).
- Gilks, Richardson and Spiegelhalter, *Markov Chain Monte Carlo Methods in Practice*. Chapman and Hall (1996).

Syllabus contributed by Mihir Arjunwadkar

M.1.27 Technical Reading, Writing, and Presentation

Contact Hours: 30 **Prerequisites: None**

Syllabus

1. Making Effective Presentations. Importance of being confident, of getting over stage fright, of eye contact. Clarity of expression, need for good language skills. Audience analysis. Anticipating and handling questions from audience. Listening to the audience. Primer on information design and its cognitive impact. Presentation slides.
2. English as a Second Language. Exercises designed to develop a functional understanding of the grammar and correct usage of English; e.g., on adjectives and adverbs, articles (*a*, *an*, *the*), appositives, prepositions, pronouns, relative pronouns, verb tenses, irregular verbs, two-part (phrasal) verbs (idioms), verbals: gerunds, participles, and infinitives; punctuation, capital letters, numbers; sentence clarity, sentence fragments, independent and dependent clauses, dangling modifiers, sentence punctuation patterns, subject/verb agreement, etc.; on pronunciation, spelling conventions, diction, vocabulary, and etymology; on text-level as against sentence-level grammar.
3. Resource Hunt and Technical Reading. Evaluating sources of information, resources for documenting sources in the disciplines, searching the world wide web, documenting conventional and electronic sources of information. Fast reading and comprehension of key ideas. Making sense out of technical documents (research papers, technical reports, manuals, etc.).
4. Academic, Scientific, Technical, and Professional Writing. Audience analysis: tailoring documents for a specific audience. Creating a thesis statement, developing an outline, pre-writing, refinement. Overcoming writer's block/writer's anxiety. Adding emphasis in writing. Annotated bibliographies. Avoiding plagiarism. Conciseness, clarity, and flow. Establishing arguments. Paragraphs and paragraphing. Quoting, paraphrasing, and summarizing. Sentence variety. Using appropriate language. Proofreading your writing; parallels in proofreading and debugging. Writing white paper/technical report, research paper, report abstracts, manuals and documentation. Effective workplace writing: accentuating the positives, prioritizing your concerns for effective business writing; memo writing, email etiquette, revision in business writing, tone in business writing, model letters for various purposes.
5. Job Search. Action verbs to describe skills, jobs, and accomplishments in employment documents. Accentuating the positives. Prioritizing your concerns for effective job search writing. Audience analysis: tailoring employment documents for a specific audience. Resume design: introduction to resumes, resume structure, when to use two pages or more, scannable resumes. Cover letters: quick tips, preparing to write a cover letter, writing your cover letter. Academic and business cover letters. Writing a job acceptance letter. Writing the curriculum vitae. Writing the personal statement/statement of purpose. Interview skills.

Notes

- This syllabus as outlined above should be considered indicative of the focus of the module. Selection of topics and emphasis may be varied at the discretion of a competent instructor.
- This is basically a remedial and need-based module to enhance the communication skills and proficiency in English of an average Indian student. Thus, e.g., learning grammatical terminology is not important. What is important is, e.g., learning correct usage of English, developing good communication skills, and moreover, developing an attitude and outlook that will help a student improve him(her)self.

Suggested Texts and References

- The Online Writing Labs (OWL) family of websites `http://owl.english.purdue.edu/`; specifically, `http://owl.english.purdue.edu/workshops/hypertext/`
- Hardy Hoover, *Essentials for the Scientific and Technical Writer.* Dover (1980).
- Steven G. Krantz, *A Primer of Mathematical Writing.* University Press (1997).
- Meenakshi Raman and Sangeeta Sharma, *Technical Communication: Principles and Practice.* Oxford University Press (2004).
- Sharon J. Gerson and Steven M. Gerson, *Technical Writing: Process and Product.* Pearson Education (2000).
- Lynne Truss, *Eats, Shoots and Leaves: The Zero Tolerance Guide to Punctuation.* Gotham (2004). Associated website: `http://eatsshootsandleaves.com/`
- Books by Edward Tufte; specifically, *Visual Explanations*, *Envisioning Information*, *Beautiful Evidence*, and *Visual Display of Quantitative Information.* See also: *The Cognitive Style of Powerpoint: Pitching Out Corrupts Within.* Associated website: `http://www.edwardtufte.com/`

Syllabus contributed by Abhay Parvate and Mihir Arjunwadkar

M.1.28 Term Paper

Contact Hours: 30 **Prerequisites: M.1.27**

Syllabus

- There is no fixed syllabus for this module. The Term Paper is an exercise in (a) developing independence, resourcefulness, and literature survey skills, (b) developing perspective on a topic not studied before and understanding the state-of-the-art in the chosen term paper topic through literature survey and internet search, and (c) developing presentation skills, in consultation with an in-house term-paper adviser (and possibly an external expert) on the chosen topic. Evaluation to be based on a written report plus a presentation cum open defense. Please refer to Sec. 4.5.5 for a discussion on the term paper.

Suggested Texts and References

- No Texts. Resource hunt and reading work as recommended by the project adviser.

Syllabus contributed by Mihir Arjunwadkar

M.1.29 Transforms

Contact Hours: 38 **Prerequisites: M.1.3, M.1.30, M.1.4, M.1.9, M.1.22**

Syllabus

1. Background and Overview. Function spaces in engineering and science. Integral transforms as linear operators on function spaces and as metric products on basis vectors. Domain and co-domain of transform, integral limits. Inverse transform and its limits. Special functions and transforms involving orthogonal functions.
2. Fourier Series, Fourier Integral, and Fourier Transforms. Periodic functions, trigonometric series. Fourier series, Euler formulae. Functions with arbitrary period. Even and odd functions. Half-range expansions. The Fourier integral. Fourier transform and its properties. Wiener-Khinchin theorem. The fast Fourier transform (FFT) algorithm. Applications involving Fourier transforms (along with their modeling contexts).

3. The Laplace Transform. Laplace transform, inverse transform, linearity. Laplace transforms of derivatives and integrals. Shifting the s- and t-axes; unit step function. Differentiation and integration of transforms. Convolution. Partial fractions. Applications along with their modeling contexts; e.g., periodic functions, LCR circuit equations, spring-dashpot model for polymers, etc.
4. Solving Differential and Integral Equations Using Integral Transforms. Boundary and initial value problems, and their relation to transforms. Integral equations and integral transforms.
5. A Plethora of Other Integral Transforms. Hartley transform as modified Fourier transform. Mellin transform and its Scaling and exponential sampling properties; Mellin transforms in image and audio processing. Bessel functions and the Hankel transform, and its applications in cylindrical symmetry and axisymmetry, cylindrical electromagnetic waves, small angle X-ray scattering, etc. Abel transform and axial symmetry; geometrical interpretation as a projection of a 3D image on a plane. Hough transform in image analysis for identifying objects like lines, circles, etc. Radon transform as integration along a line, applications in cross-sectional images using CAT scans, hyperbolic differential equations, tomographic reconstruction, projection slice theorem, inverse radon transform and its sensitivity to noise. Hilbert transform and signal processing, filtering out negative frequency modes, amplitude modulation and demodulation, amplitude envelope followers and Hilbert transforms.
6. Wavelet Transforms. Why wavelets? Bases for L_2: preliminaries, orthonormal bases generated by a single function: the Balian-Low theorem, smooth projections on L_2, local sine and cosine bases and the construction of some wavelets, unitary folding operators and smooth projections. Multiresolution analysis and the construction of wavelets. Band-Limited Wavelets: orthonormality, completeness, the Lemarié-Meyer wavelet. Other Constructions: Franklin wavelets on the real line, spline wavelets on the real line, orthonormal bases of piecewise linear continuous functions for L_2, orthonormal bases of periodic splines, periodization of wavelets defined on the real line. Representation of functions by wavelets. Applications along with their modeling contexts. The fast wavelet transform (FWT) algorithm.
7. Transforms in Perspective. Areas of application of various transforms. Classification by symmetry. Nature of PDEs, boundary conditions, geometric interpretation, etc. Comparison of properties of transforms. Scope for improvising transforms and developing new ones.

Notes

- This syllabus as outlined above should be considered indicative of the focus of the module. Selection of topics and emphasis may be varied at the discretion of a competent instructor.

Suggested Texts and References

- L. C. Andrews and B. K. Shivamoggi, *Integral Transforms for Engineers*. Prentice-Hall of India (2003). Indian edition/reprint available.
- Paul L. DeVries, *A First Course in Computational Physics*. Wiley (1993).
- J. W. Brown and R. V. Churchill, *Fourier Series and Boundary Value Problems*. McGraw-Hill (sixth edition, 2001).
- David F. Walnut, *An Introduction to Wavelets Analysis*. Birkhauser (2001).

- Boggess and Narcowich, *A First Course in Wavelets with Fourier Analysis*. Prentice-Hall (2001).
- Eugenio Hernandez and Guido Weiss, *A First Course on Wavelets*. CRC Press (1996).

Syllabus contributed by Sukratu Barve and Mihir Arjunawdkar

M.1.30 Vector Analysis

Contact Hours: 15 **Prerequisites: M.1.3, M.1.9**

Syllabus

1. Preliminary Concepts and Motivating Examples. Vector functions. Explicitly, parametrically, and implicitly defined functions. Level sets. Vector fields. Arc length. Line integrals.
2. Vector Calculus. Partial derivatives. Order-independence. Vector partial derivatives and their geometric significance, flows, and trajectories. Limits and continuity. Differentials and the Jacobian. Directional derivative. Gradient and its geometric significance. Composition of functions and the chain rule. Inverse and implicit function theorems. Surfaces and tangents.
3. Real-Valued Functions. Extreme values and critical points. Lagrange multiplier method. Quadratic polynomials. Taylor expansions. Hessian and its geometric significance. Uniform convergence. Orthogonal functions.
4. Multiple Integration and Vector Fields. Iterated integrals. Multiple integrals. Change of variables. Improper integrals. Scalar product, vector product, triple vector and scalar product. The operators $\nabla, \nabla\cdot$, and $\nabla\times$. Successive applications of ∇. Green's theorem. Conservative vector fields. Surface integrals. Stokes' theorem. Gauss' theorem. Potential theory, Gauss' law, Poisson's equation, Helmholtz's theorem. Differential forms. The exterior derivative.
5. Vector Analysis in Curved Coordinates and Tensors. Curvilinear coordinates. Orthogonal coordinates in $\mathbf{R}^3$. Differential vector operators. Special coordinate systems. Circular cylinder coordinates; spherical polar coordinates. Tensor analysis. Contraction, direct product. Quotient rule. Pseudotensors, dual tensors. General tensors. Tensor derivative operators.

Suggested Texts and References

- Williamson, Crowell, and Trotter, *Calculus of Vector Functions*. Prentice-Hall (1972).
- Courant and John, *Introduction to Calculus and Analysis: Volume I*. Springer (1989).
- Arfken and Weber, *Mathematical Methods for Physicists*. Elsevier (2005).
- Erwin Kreyszig, *Advanced Engineering Mathematics*. Wiley (ninth edition, 2006).

Syllabus contributed by Abhay Parvate and Mihir Arjunwadkar

M.2 Elective Modules: Sample Syllabi

M.2.1 Computational Fluid Dynamics

Contact Hours: 75 **Prerequisites: As decided by the instructor/s**

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 co-moving derivative.
2. Balance Equations. Equation of continuity. Jacobians and their rates of change. Lagrangian coordinates. Reynolds theorem. Surface forces and traction vector. Cauchy theorem and concept of stress tensor. Cauchy equation of momentum balance. Angular momentum balance equation. Heat flux density, internal energy density, energy balance equation.
3. Constitutive Relations. Introduction. Thermodynamic stimulus and response, rate of response. Darcy's, Fourier's, Ohm's and Fick's laws, Hooke's law, Newton's law of viscosity. Shear, rotation and dilation of velocity field, Navier-Stokes equation, boundary conditions and their importance.
4. Examples of Flow. Hagen-Poisseuille flow, Couette flow and other special cases.
5. FEM Techniques. Finite elements. Shape functions. Finite element interpolation functions. Weighted residual approach. Assembly of element equation. Finite element formulation for advection equation.
6. Finite Volume Approach. Finite volume method. Finite volume discretisation. 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.
7. Grid Generation. Structured grid generation. Unstructured grid generation. Mesh adaptation. Automatic grid generation for complex geometry problems. Computing techniques.
8. Application to Multiphase Flows.
9. Higher-Order Methods for CFD.
10. Optimization Through CFD. Optimization problem associated with evaluation of first derivative. Optimization problem associated with evaluation of second derivatives.
11. Problem-Solving Through CFD. Implementation of codes in CFD. Computational environments for CFD (such as `Phoenix`).

Suggested Texts and References

- T. J. Chung, *Computational Fluid Dynamics*. Cambridge University Press (2002).

Syllabus contributed by Sukratu Barve and Prof. K. C. Sharma

M.2.2 Machine Learning

Contact Hours: 75 **Prerequisites: As decided by the instructor/s**

Syllabus

1. Introduction and Background. What is machine learning–overview and survey of applications. Problem of induction and statistical inference: Input-output functions, Boolean functions, Parametric and nonparametric inference. Probability, certainty and Bayes theorem. Introduction to typical learning tasks: regression, pattern recognition, feature selection, classification, clustering, rule induction (association). Model validation techniques: cross-validation, leave-one-out, majority, etc., Measures of performance (sensitivity, specificity, ROC curves, etc.)
2. Computational Environments for Machine Learning. Setting up of modeling frameworks (`Weka, Orange and R`), I/O formats, basic introduction to interfaces.
3. Formulation of the Learning Problem. Learning as a statistical problem: estimation of probability measure and basic problems of statistics, learning as density estimation, risk, empirical risk and structural risk, introduction to ill-posed problems and regularization. Learning as an algebraic problem. Learning as a computational problem: learnability, PAC learning, bounds on data, algorithmic learning theory basics. Laboratory: linear models in `R`, writing basic interface for a learner.
4. Supervised Learning. Additive model: logistic regression. Generative model: naïve Bayes classifier. Discriminative model: neural network (NN) and support vector machine (SVM). Laboratory: models using `Weka` or `Orange` on UCI benchmark data sets. Writing interfaces for a classifier as derived from a learner.
5. Unsupervised Learning. Clustering: k-means, hierarchical, self-organizing map, EM algorithm. Feature selection: principal component analysis. Laboratory: k-means clustering, writing interface for a clusterer.
6. Reinforcement Learning.
7. Ensemble Methods. Boosting and bagging.
8. Intelligent Agents.

Suggested Texts and References

- T. Hastie, R. Tibshirani and J. H. Friedman, *The Elements of Statistical Learning.* Springer (2003).
- Tom Mitchell, *Machine Learning.* McGraw-Hill (1997).

Syllabus contributed by Ashutosh

M.2.3 Theory of Computation

Contact Hours: 75 **Prerequisites: As decided by the instructor/s**

Syllabus

1. Theory of Computation as a Modeling Formalism. Symbolic dynamics of dynamical systems, biological sequence analysis and stochastic grammars, computational musicology, etc.
2. Introduction to Languages. Symbols, strings, words and languages. Symbolic dynamics, dynamics as language. Examples of languages. Finite representation of languages. String induction principles.
3. 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.

4. 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.

5. 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.

6. 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.

Suggested Texts and References

- H. Lewis and C. Papadimitrion, *Elements of Theory of Computation*. Prentice-Hall (second edition, 1998). Indian edition/reprint available.
- V. E. Krishnamurthy, *Introductory Theory of Computer Science*. Springer-Verlag (1985). Indian edition/reprint available.

Syllabus contributed by Ashutosh and Abhijat Vichare

M.3 The Project (Internship/Industrial Training)

M.3.1 Project

Contact Hours: 375

Prerequisites: Passing grade in at least 30CR worth of course work of Year 1

Syllabus

- Please refer to Sec. 4.5.9 for a discussion on the project.

Suggested Texts and References

- No Texts. Resource hunt and reading work as recommended by the project advisor.

Syllabus contributed by Mihir Arjunwadkar