MATH3311/MATH5335 – Course Outline

Information about the course

Course Authority: A/Prof Rob Womersley, RC-3062, email R.Womersley@unsw.edu.au

Consultation: Check “Course Contacts” in UNSW Blackboard or by email

Credit, Prerequisites, Exclusions:
This course counts for 6 Units of Credit (6UOC).
The prerequisites for MATH3311 are

- MATH2120 Mathematical Methods for Differential Equations (or the higher version MATH2130)
- MATH2501 Linear Algebra (or the higher version MATH2601)
- 6 UOC of Level 2 Statistics.

In addition, it is beneficial to have prior experience with Matlab and with simple procedural programming, such as is provided in MATH2301 Mathematical Computing or MATH1151/1251 Mathematics for Actuarial Studies and Finance.

There is no higher version of MATH3311.
The graduate course MATH5335 has no formal prerequisites but students need to have taken courses roughly equivalent to those mentioned above.
The undergraduate course MATH3311 and the postgraduate course MATH5335 exclude each other.

Classes: Combined Lectures (2 hours) for MATH3311 and MATH5335 will run

| Thursday 4:00–6:00pm | RC-4082 | Weeks 1–5, 7–13 |

MATH3311 has two hours of laboratory/tutorial classes which run

| Wednesday 9:00–11:00am | RC-G012C | Weeks 2–9, 11–13 |

In Week 10, the laboratory test will be held in place of the normal tutorial.

MATH5335 has a one hour laboratory/tutorial class immediately after lectures:

| Thursday 6:00–7:00pm | RC-M020 | Weeks 2–9, 11–13 |

Students in the graduate course MATH5335 are expected to display much more independence, especially in working through the laboratory exercises.

UNSW Blackboard: Further information, skeleton lecture notes, sample programs and other material will be provided via UNSW Blackboard. You should log on to the MATH3311/MATH5335 course module regularly to check for announcements.
Course aims

In the end finance is concerned with making definite numerical recommendations which frequently can only be obtained by analyzing sophisticated models using high-speed computers. This course studies the design, implementation and use of computer programs to solve practical mathematical problems of relevance to finance, insurance and risk management.

Relation to other mathematics courses

The most closely related courses are MATH2301 Mathematical Computing and MATH3101 Computational Mathematics (with its postgraduate version MATH5305). Some aspects are also related to the Numerical Methods component of MATH2089 Numerical Methods and Statistics, which is only available to students in Engineering programs.

Student Learning Outcomes

Practical solution of problems of relevance to finance using Matlab is an essential component of this course. This prepares you for the finance/insurance workforce where software like Matlab or Excel/VBA is used for research, investigation and proof of concept before it is passed on for professional programming, often in C/C++, integrating with the company’s front end systems, databases and risk management processes.

Relation to graduate attributes

This course will help develop your quantitative skills and ability to reason logically and mathematically and apply these skills to problems of relevance to finance, insurance and risk management. It will emphasize the importance of understanding the underlying mathematics, computational techniques and problems solving skills. It will help establish what confidence you should have in the computed answers and what are the risks associated with a given computational problem.

You are expected to always think about your results and their consequences, not just blindly believe what is produced by a computer.

The above outcomes are related to the development of the Science Faculty Graduate Attributes, in particular:

1. Research, inquiry and analytical thinking abilities,
2. Communication,
3. Information literacy
Teaching strategies underpinning the course

The lectures will introduce a wide range of computational methods used in a variety of financial applications. During the lab classes, you will use Matlab to study examples involving these methods. The examples will illustrate key features of the methods, including any significant limitations.

The key to success in MATH3311/MATH5335 is to work through as many of the lab exercises as possible.

Assessment

The course has four in semester assessment tasks plus a final exam, weighted as follows:

<table>
<thead>
<tr>
<th>Assessment</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Online Quiz 1 (week 5)</td>
<td>5%</td>
</tr>
<tr>
<td>Assignment (week 7)</td>
<td>20%</td>
</tr>
<tr>
<td>Lab Test (week 10)</td>
<td>20%</td>
</tr>
<tr>
<td>Online Quiz 2 (week 11)</td>
<td>5%</td>
</tr>
<tr>
<td>Final Exam (2 hours)</td>
<td>50%</td>
</tr>
</tbody>
</table>

The assignment must be YOUR OWN WORK, or severe penalties will be incurred. You should consult the University web page on plagiarism.

http://www.lc.unsw.edu.au/plagiarism

Maple TA quizzes

The online quizzes will be conducted in MapleTA based on material covered in lectures and tutorial/laboratories to that stage. They are intended to be both a learning activity and an assessment of your knowledge at that stage.

There is a link to Maple TA in the UNSW Blackboard course, as well as information on how to use Maple TA.

There are MATLAB self-paced lessons available through UNSW Blackboard and associated MapleTA quizzes. If you are not familiar with MATLAB, have not used MATLAB for some time, then these lessons should be done as soon as possible. The Maple TA quizzes on MATLAB do not count towards your grade, but can be used both to check your knowledge of MATLAB and to become familiar with MapleTA.

Knowledge and abilities assessed

The assignment due in week 7 will have a written and a computing component, and emphasize basic Matlab programming and will also provide a preview of topics to be covered during lectures later in the course. This task will test your skill at synthesizing mathematical theory and practical computation, and at interpreting and presenting numerical results.

The laboratory test in week 10 will take place in the computer lab under exam conditions. You will have to write several short Matlab scripts and functions to solve simpler versions of the lab exercises.
The Final Exam will mainly test your mathematical understanding of the numerical methods discussed in lectures, and of their application in a financial context.

**Assessment criteria**

For computing tasks, the main criteria will be correctness, efficiency and clarity. In other words, does the program produce the correct answers? Does the program run in a reasonable time with reasonable memory requirements? Can a human readily understand (from reading the source code) what the program does? For written tasks, you should set out your working clearly and in a logical sequence, with adequate justification for each step. Aim for the most direct answer possible.

Students in the postgraduate version will be expected to complete additional/harder sections in the assignment and final exam.

**Additional resources and support**

**UNSW Blackboard**

All course materials will be available on UNSW Blackboard at

[http://lms-blackboard.telt.unsw.edu.au/](http://lms-blackboard.telt.unsw.edu.au/)

You should check regularly for new materials, as well as for announcements about assessment tasks etc.

**Tutorial and Laboratory Exercises**

Exercise sets for the laboratories/tutorials will be provided. In the labs, you will work independently at your own pace, but I will be available to help when necessary. Take advantage of this help: it may save you a great deal of time.

During the lecture I will outline which lab exercises I expect you to cover over the coming week. Before each lab, you should prepare by reading the relevant exercises and at least making a start on some of them.

**Lecture notes**

Lecture notes, including laboratory exercises, and also notes summarizing key prerequisite knowledge about Matlab and statistical concepts, will be provided through UNSW Blackboard.

**Software**

You can purchase the Matlab & Simulink Student Version R2012a from the University Bookshop for $119. See
Doing so is not essential because you can use the software in the Mathematics and Statistics computer labs, but owning a personal copy of Matlab will certainly be convenient, especially if you study part-time and are not often on campus. The student edition includes several toolboxes, including the symbolic math, optimization and statistics toolboxes which are very useful for financial applications.

A free software package called Octave is largely compatible with Matlab, but has a less sophisticated user interface. You can download Octave from


**Computer laboratories**

See the School of Mathematics and Statistics web page for information about opening times etc. **Note that the three downstairs labs in RC-G012 will be unavailable for week 10 when they are reserved for running tests.** The labs are also likely be very busy during week 9.

**Remote access Linux Lab PC**

On UNSW Blackboard you will find information on using the NX secure communication software to access `sigma.maths.unsw.edu.au`, which can be used to run mathematical and statistical software such as MATLAB, Maple and R.

**References**

There is no textbook which covers all aspects of this course. Use the library catalogue to search the library for books on *Computational Finance, Mathematical Computing, Mathematical Software*, Matlab, etc. A book which follows the themes of this course is *Numerical Methods in Finance: A Matlab based introduction* by Paolo Brandimarte [2]. This book P 332.0151/10 has been paced in the UNSW Library high use collection. Standard texts on financial derivatives are Hull [10] and Wilmott [21, 23], while Glasserman [5] and Jäckel [11] are good for Monte-Carlo based simulation methods. [23, 13, 22] are more general texts on computational aspects of finance..

**General References**

Numerical methods for partial differential equations [14, 17, 21] are a key part of the subject. Methods based on Fourier analysis [3] are also important. A general reference on applied mathematics covering many aspects of this subject is Strang [19]. General references on scientific computing include [4, 6, 12]. Nash [16] covers the history of scientific computing.
Linear Algebra

An elementary treatment of linear algebra can be found in Strang [20], while a more advanced reference is Golub and Van Loan [7]. The LAPACK manual [1] contains both descriptions of the methods and available software.

Mathematical Software

Solving practical problems typically requires programming, either in a high level numerical language like Matlab [9], a symbolic package like Maple, or a lower level language like Fortran or C. If you are using Fortran or C the routines available in the Numerical Recipes [18], the LAPACK library [11] and the NAG Library [15] may be very useful.

References


Course Evaluation and Development

The School of Mathematics and Statistics evaluates each course each time it is run. We consider the student responses and their implications for course development.

Administrative matters

It is the student’s responsibility to be familiar with UNSW and School of Mathematics and Statistics policies.

School Rules and Regulations

Details of the general rules regarding attendance, release of marks, special consideration etc are available via the School of Mathematics and Statistics Web page at http://www.maths.unsw.edu.au/currentstudents/assessment-policies.

Additional Assessment


Plagiarism and academic honesty

Plagiarism is the presentation of the thoughts or work of another as one’s own. Issues you must be aware of regarding plagiarism and the university’s policies on academic honesty and plagiarism can be found at http://www.lc.unsw.edu.au/plagiarism and http://www.lc.unsw.edu.au/plagiarism/plagiarism_STUDENTBOOK.pdf.
Detailed course schedule

The table below provides a week-by-week overview of the lecture content.

<table>
<thead>
<tr>
<th>Week</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Computing: hardware, software, accuracy and efficiency</td>
</tr>
<tr>
<td>2</td>
<td>Vector and matrix norms, condition numbers, special matrices</td>
</tr>
<tr>
<td>3</td>
<td>Linear systems and matrix decompositions</td>
</tr>
<tr>
<td>4</td>
<td>Data fitting: least squares, $\ell_1$ and splines; Model calibration</td>
</tr>
<tr>
<td>5</td>
<td>Fourier analysis: discrete Fourier transform and FFT</td>
</tr>
<tr>
<td>6</td>
<td><strong>No lectures this week</strong> but tutorials will be held.</td>
</tr>
<tr>
<td>7</td>
<td>Sparse matrices: banded matrices and general sparse matrices</td>
</tr>
<tr>
<td>8</td>
<td>Nonlinear equations: implied volatility, nonlinear systems</td>
</tr>
<tr>
<td>9</td>
<td>Numerical integration: expected values, one dimensional, multi-dimensional</td>
</tr>
<tr>
<td>10</td>
<td>Monte-Carlo methods: pseudo random numbers, QMC methods</td>
</tr>
<tr>
<td>11</td>
<td>Weiner processes, stochastic differential equations, stochastic integrals</td>
</tr>
<tr>
<td>12</td>
<td>Option pricing: Black–Scholes formula, PDE</td>
</tr>
<tr>
<td>13</td>
<td>Black–Scholes PDE: Finite difference methods</td>
</tr>
</tbody>
</table>