THE UNIVERSITY OF NEW SOUTH WALES

FACULTY OF SCIENCE
SCHOOL OF MATHEMATICS AND STATISTICS

MATH3311 Mathematical Computing for Finance

MATH5335 Computational Methods for Finance

Session 2, 2009

MATH3311/MATH5335 – Course Outline

Information about the course

Course Authority: Dr. W. McLean RC-2085, email `w.mclean unsw.edu.au`

Consultation: see the timetable outside my office door (after week 1)

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 in Level 2 Statistics.

In addition, you should 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: Lectures will run in weeks 1–12, and lab classes in weeks 2–13. Both courses share 2 lectures per week

Wednesday	4:00–6:00PM	ME-304

The undergraduate course MATH3311 has two lab classes per week

Thursday	2:00–3:00PM	RC-G012A
Friday	1:00–2:00PM	RC-G012A

and the postgraduate course MATH5335 has one lab class

Wednesday	6:00–7:00PM	RC-G012A

Depending on enrolments, we may merge one of the undergraduate lab classes with the postgraduate one.

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

WebCT: Further information, skeleton lecture notes, and other material will be provided via WebCT. You should log on to the 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 analysing 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).

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 to professional programming, often in C/C++, integrating with the company's front end systems 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**, 4. **Communication**, 6. **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 MATH3311/5335 is to work through as many of the lab exercises as possible.

Assessment

The course has three assessment tasks, weighted as follows:

Assignment (week 5)	20%
Lab Test (week 10)	30%
Final Exam (2 hours)	50%

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

`www.lc.unsw.edu.au/plagiarism`

Knowledge and abilities assessed: The assignment due in week 5 will provide early feedback on your progress in the course. This task will emphasise basic Matlab programming and will also provide a preview of topics to be coveried during lectures later in the course. The test in week 9 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? And 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.

Additional resources and support

webCT

All course materials will be available on webCT at

`http://vista.elearning.unsw.edu.au`

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

Tutorial and Laboratory Exercises

I will provide exercise sets for the tutorials and labs. 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

The course has a detailed set of lecture notes, written by Assoc. Prof. Rob Womersley. These notes include the lab exercises, and also a couple of appendices summarizing key prerequisite knowledge about Matlab and statistical concepts.

Software

You can purchase the Student Edition of Matlab (Version 2009a) from the University Bookshop for $119. Doing so is not essential because you can use the software in the Maths and Stats 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.

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

`http://www.gnu.org/software/octave/`

Computer laboratories

See the School of Maths and Stats web page for information about opening times etc. **Note that the three downstairs labs will be unavailable for the whole of week 10 which is reserved for running tests.** The labs are also likely be very busy during week 9.

Reference books

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]. Other particularly useful references are [5, 10, 22, 12, 21].

General References

General references on mathematical finance include [23, 24, 9]. Numerical methods for partial differential equations [13, 16, 20] 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 [18]. General references on scientific computing include [4, 6, 11]. Nash [15] covers the history of scientific computing.

Linear Algebra

An elementary treatment of linear algebra can be found in Strang [19], 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 [8], 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 [17], the LAPACK library [1] and the NAG Library [14] may be very useful.

References

[1] E. Anderson, Z. Bai, C. Bischoff, J. Demmel, J. J. Dongarra, J. D. Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov, and D. C. Sorensen, *LAPACK Users' Guide*, SIAM, Philadelphia, second ed., 1995.

[2] P. Brandimarte, *Numerical Methods in Finance: A MATLAB based introudction*, John Wiley, New York, 2002.

[3] W. L. Briggs and V. E. Hansen, *The DFT: An Owners Manual for the Discrete Fourier Transform*, SIAM, Philadelphia, 1995.

[4] L. D. Fosdick, E. R. Jessup, C. J. C. Schauble, and G. Domik, *An Introduction to High-Performance Scientific Computing*, MIT Press, 1996.

[5] P. Glasserman, *Monte Carlo Methods in Financial Engineering*, Springer, 2005.

[6] G. H. Golub and J. M. Ortega, *Scientific Computing: An Introduction to Parallel Computing*, Academic Press, New York and London, 1993.

[7] G. H. Golub and C. F. Van Loan, *Matrix Computations*, John Hopkins University Press, Baltimore and London, third ed., 1996.

[8] D. J. Higham and N. J. Higham, *MATLAB Guide*, SIAM, Philadelphia, 2000.

[9] J. Hull, *Options, Futures, and other Derivative Decurities*, Prentice Hall, Englewood Cliffs, NJ, 3rd ed., 1997.

[10] P. Jäckel, *Monte Carlo Methods in Finance*, John Wiley, Chichester, 2002.

[11] N. Knockler, *Numerical Methods and Scientific Computing: Using Software Libraries for Problem Solving*, Clarendon Press, Oxford, 1991.

[12] B. Lapeyra, A. Sulem, and D. Talay, *Understanding Numerical Analysis for Option Pricing*, Cambridge University Press, Cambridge, UK, 2000.

[13] K. W. Morton and D. F. Mayers, *Numerical Solution of Partial Differential Equations*, Cambridge University Press, Cambridge, UK, 1991.

[14] *NAG numerical Libraries*, Numerical Algorithm Group, Oxford, U.K. http://www.nag.co.uk/numeric.html.

[15] S. G. Nash, *A History of Scientific Computing*, ACM Press, New York, 1990.

[16] J. M. Ortega and G. H. Golub, *Scientific Computing and Differential Equations: An Introduction Numerical Methods*, Academic Press, New York and London, 1992.

[17] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in Fortran: The Art of Scientific Computing*, Cambridge University Press, Cambridge, UK, second ed., 1992. Second edition is also available in C.

[18] G. Strang, *Introduction to Applied Mathematics*, Wellesley-Cambridge Press, Cambridge, MA, 1986.

[19] ——, *Linear Algebra and its Applications*, Harcourt Brace Jovanovich, San Diego, 3 ed., 1988.

[20] J. C. Striwerda, *Finite Difference Schemes for Partial Differential Equations*, Wadsworth and Brooks/Cole, 1989.

[21] D. Tavella, *Quantitative Methods in Derivative Pricing: An introudcution to Computational Finance*, John Wiley, Chichester, 2002.

[22] D. Tavella and K. Randall, *Pricing Financial Instruments: the finite difference method*, John Wiley, New York, 2000.

[23] P. Wilmott, *Derivatives: The Theory and Practice of Financial Engineering*, John Wiley, 1998.

[24] P. Wilmott, S. D. Howison, and J. Dewynne, *The Mathematics of Financial Derivatives: A Student Introduction*, Cambridge University Press, Cambridge, UK, 1995.

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

Additional Assessment

Information about additional assessment for Maths and Stats courses is posted at `http://www.maths.unsw.edu.au/students/current/examinfo.html`.

School Rules and Regulations

Fuller 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/students/current/policies/studentpolicy.html`.

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.

Week	Topic
1	Estimating computational cost; floating-point arithmetic
2	Special matrices
3	Matrix decompositions
4	Data fitting
5	Fourier analysis
6	Sparse matrices
7	Nonlinear equations
8	Numerical integration
9	Monte-Carlo methods
10	Lab Test
11	The Black–Scholes model for option pricing
12	Finite difference methods