COSC 2001 Computational Science  
COSC 2901 Computational Science (Advanced)  
First Semester 2004  

UNIT OF STUDY OUTLINE

Welcome to COSC

COSC 2001/2901 provides an Intermediate treatment of the solving of scientific problems using both the C and MATLAB programming languages. The scientific problems addressed will be drawn from a wide variety of disciplines, and only a basic understanding of C and MATLAB is expected. The emphasis is on problem solving rather than programming, with the aim of developing the use of computers in science for students with diverse backgrounds and interests.

In the natural sciences the most commonly used programming languages are Fortran, C/C++ and MATLAB. For the first half of the semester MATLAB will be used to analyse and interpret scientific data. In the second half of the semester, C has been chosen to solve scientific problems. You will not be expected to master advanced aspects of either language, but you will be expected to learn to use C and MATLAB to solve realistic problems arising in the natural sciences. Examples of the kind of techniques that you will learn to use include: evaluating integrals numerically, solving differential equations, applying linear algebra and determining the stability of systems. You will be able to use libraries of mathematical routines, and be able to recognise when a numerical approach is required in a problem, and solve such problems. Whilst the unit of study is not a complete, formal course in programming, you will develop basic programming skills, including an understanding of structured programming and the importance of documentation.

COSC 2001 students will share the same lecture and practical sessions as COSC 2901 students. However, COSC 2901 students will be given some alternative problems in the practical sessions and will be expected to gain a greater understanding of the mathematical methods used in obtaining numerical solutions.

Background

Computational Science is a new Science major at the University of Sydney, focusing on scientific problem solving using computers. It covers the formulation and analysis of problems, the use of software packages and programs to solve these problems computationally, simulations and modelling, mathematical and numerical analysis, high-performance supercomputing, graphics, visualization, and programming. A student majoring in Computational Science can choose from a wide range of electives offered by various Departments and Schools across the Faculty. The major in Computational Science is defined by the Senior (3rd year) Core and Elective courses; units at Junior and Intermediate (1st and 2nd year) level are provided for student interest and experience.

Summary

| Credit points | 6 |
| Offered       | First semester |
| Assumed Knowledge | A basic understanding of both C and MATLAB |
| Prerequisites  | COSC 2001: 12 credit points chosen from junior mathematics or junior computational science units.  
                 COSC 2901: 12 points at a credit level chosen from junior mathematics or junior computational science units. |
| Prohibition    | COSC 2001 may not be counted with COSC 2901 |
| Classes        | 2 hours of lectures and 3 hours of practical sessions per week |
| Assessment     | 2 practical exams and a final written examination |
Unit of study goals

The general goals of COSC 2001/COSC 2901 are to enable you

- to develop tools to solve scientific problems at an Intermediate level using computational methods, and
- to develop insight into the structure of computational problems.

COSC 2001/COSC 2901 also aims to help you acquire some of the generic attributes expected of all graduates of the University (http://policy.rms.usyd.edu.au/0000050.pdf). These attributes include knowledge skills (in programming and numerical methods, ability to communicate results), thinking skills (model building, analysis of problems and interpretation of the results), personal skills and attributes (the ability to work independently or as a member of a group), and practical skills (the use of computers).

Your learning commitments

Students enrolled in any 6-credit point unit of study offered by the Faculty of Science should consider spending up to 9 hours per week on that unit during the 13 teaching weeks and the 2 study vacation weeks. In COSC 2001/COSC 2901 this involves the following.

- 25 1-hour lectures. Lectures will include computer simulations, demonstrations and class discussions.
- 12 3-hour practical sessions in a computational science laboratory, based on and supporting the lecture modules. You work alone or in pairs at a computer on a selection of qualitative and quantitative questions and problems. Practical notes with problems will be provided, and tutors are present to assist you.
- 4 hours per week of independent study. You are expected to use this time to read through and understand your notes, practical notes and relevant sections of the recommended references and to study for the written and practical examinations.

If you attend classes regularly and involve yourself in all of these learning experiences, you will gain a good understanding of the course work. This will have a considerable impact on your exam preparation and performance.

This is an Intermediate course and basic knowledge of both C and MATLAB is assumed. Mathematical techniques such as algebra and calculus will be used during the course, and hence 1st year mathematics is a prerequisite. If you are experiencing difficulties with mathematics during the unit please inform your lecturer and seek assistance from the Mathematics Learning Centre (Carslaw building).

Class timetabling

There is a single lecture stream for all students in COSC 2001/COSC 2901.

<table>
<thead>
<tr>
<th>Session</th>
<th>Location</th>
<th>Times</th>
<th>Starting Date</th>
<th>Ending Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lecture 1</td>
<td>Physics Lecture Theatre 1</td>
<td>Monday 11 am</td>
<td>8 March</td>
<td>7 June</td>
</tr>
<tr>
<td>Lecture 2</td>
<td>Physics Lecture Theatre 1</td>
<td>Tuesday 12 pm</td>
<td>9 March</td>
<td>8 June</td>
</tr>
<tr>
<td>Practical Session</td>
<td>Physics Room 359, Computational Science Lab</td>
<td>Tuesday 2-5 pm</td>
<td>9 March</td>
<td>8 June</td>
</tr>
</tbody>
</table>

Note: there will be no classes over the mid-semester break.

Assessment

Assessments are intended to allow you to demonstrate what you have learned related to the goals of this unit. They also serve to encourage you to interact with the material, but it should not dominate your approach. See it as another learning activity to complement those listed earlier.

Assessments for this unit of study are based upon practical work, two practical exams of 90 minutes each, and a 2-hour theory exam. The practical exams will be held during week 7 and after the final practical session following week 13.

<table>
<thead>
<tr>
<th>Date</th>
<th>Assessment Task</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tuesday 27 April</td>
<td>Prac exam MATLAB</td>
<td>25%</td>
</tr>
<tr>
<td>TBA</td>
<td>Prac exam C</td>
<td>25%</td>
</tr>
<tr>
<td>Prac sessions weeks 1-6, 8-13</td>
<td>Prac session attendance</td>
<td>10%</td>
</tr>
<tr>
<td>TBA</td>
<td>Written Exam</td>
<td>40%</td>
</tr>
</tbody>
</table>
A two-hour examination covering the material included in the unit of study is held at the end of the semester. You will be asked to write descriptive answers to questions, to explain physical principles and to answer quantitative questions.

Assessment in the practical sessions is based on successful completion of a set of exercises. For each laboratory session, you are awarded marks based on your record of completion of the exercises in a logbook.

Students in science must be able to express themselves accurately by clear, efficient use of the English language in their written work. Spelling, grammar, punctuation and correct use of language will be taken into account when examination work is assessed.

**Study resources**

**Recommended References**
There is no text book as such. However, the following are recommended references.


B. Gottfried, *Schaum’s Outlines - Programming with C*, (McGraw-Hill 1996), which is available from the Co-op Bookshop and introduces the basics of the C programming language.


**Web Resources**

The ‘Student Information’ link on the School of Physics web page (http://www.physics.usyd.edu.au) provides resources to help you with your studies. Please spend time getting acquainted with this site, and the specific page relative to your unit of study. Unit webpages are provided under the University’s WebCT environment, which can be accessed from the Junior Physics webpages (http://www.physics.usyd.edu.au/ugrad/jpc.html) or the Student Intranet site (http://intranet.usyd.edu.au/student/). Access to some Intranet pages requires a Unikey username and password that is issued with your confirmation of enrolment. The University provides computer facilities in the Access Centres (http://www.usyd.edu.au/su/is/labs/).

A brief introduction to help you with web access is available on the Junior Physics web page.

**Where to go for help**

If you need help, you can

- go to the Physics Student Office, Room 202 in the Physics building, or phone 9351 3037
- ask your lecturer or tutor
- ask other students using the Discussion forum provided under the Communication link on the unit WebCT page.
- consult one of the many services provided by the University, such as the Maths Learning Centre. These can be found by choosing Junior Physics Resources and Links from the unit web page or through the Student Intranet site (http://intranet.usyd.edu.au/student/).

**Please note:** The University provides you with email access based on your username. We may use this email address to provide you with important information regarding this unit of study. **We expect you to periodically read this email account or to forward mail from it to an email account you do read (eg. a hotmail account).** See the ITS helpdesk (http://helpdesk.usyd.edu.au) for information on how to forward your email.
Special Consideration

If your academic performance is adversely affected by illness or some other serious event, such as an accident, you should notify the Faculty of Science Student Information Office (level 2 of the Carslaw building) within 7 days, by completing an Application for Special Consideration with accompanying documentation. This is especially important if you miss an examination.

You should not submit an Application for Special Consideration if you have a reasonable opportunity to make up any work you missed. If you miss an assignment, Special Consideration is appropriate to allow late submission, but we do expect the assignment to be submitted. Sometimes, catching up may be impossible, in which case we will consider a pro-rata adjustment of your marks on the basis of an Application for Special Consideration.

The Special Consideration procedures have changed considerably in 2004. To submit an application you should:

Obtain a Special Consideration Application pack (containing the Application for Special Consideration Form, the Professional Practitioners Certificate and the Academic Judgment Form) from the Student Information Office of the Faculty of Science, the Faculty website at http://www.science.usyd.edu.au, or the Physics Student Office.

Complete the forms and obtain whatever original documentary evidence is appropriate. Note especially that the Professional Practitioner's Certificate is essential for Special Consideration on grounds of serious illness - Medical Certificates will NOT be accepted.

Take the original copy of all forms and documents, plus sufficient copies for each unit of study affected and yourself, to the Faculty of Science Student Information Office. They will sign/stamp both the original application form and the copies. In the case of Physics units, one copy of the documentation must then be submitted to the Physics Student Office. Keep one copy yourself. A formal decision on your application will be sent to your university email address within 7 days.

Further details on University policy regarding Special Considerations can be found at http://policy.rms.usyd.edu.au/00000ag.pdf.

Students with special problems should consult the Intermediate Physics Coordinator. If you have problems which are affecting your ability to study, do not hesitate to consult the relevant person; often they are able to help directly or suggest ways of overcoming the problems. But they cannot help you unless they know about your problem.

Lecture and practical session outlines

There is a outline for each of the 13 weeks of study which list specific objectives that define what you should learn and understand. Understanding a term or concept means that you should be able to:

• explain its meaning in writing and give examples;
• interpret it correctly when you read or hear it;
• use it correctly in your writing; and
• apply it correctly to examples and problems.

The outline also lists what sections of the recommended references are relevant.

MATLAB component 2001/2901

Week: 1 – Introduction and review of MATLAB

Specific objectives – after this week you should be able to:

• Use the basic commands of MATLAB to carry out scalar and matrix operations, plot graphs in 2 and 3 dimensions, write script files for simple problems.
• Use the intrinsic facilities and documents of MATLAB to get help on any aspect of MATLAB.

Week: 2 – Matrices and linear algebra
Relevant parts of recommended references: Using MATLAB, chapter 10.

Specific objectives – after this week you should be able to:

• Create and manipulate matrices, e.g., find transpose, inverse, norm, determinant, and eigenvalues.
• Solve linear systems of equations, e.g., coupled harmonic oscillators.
• Solve eigenvalue equations, e.g., normal modes of molecular vibrations.
Week: 3 – Data analysis and statistics
**Relevant parts of recommended references:** Using MATLAB, chapter 12.

**Specific objectives – after this week you should be able to:**
- Load a set of data and analyse it using the MATLAB functions.
- Fit the data to a given functional form and determine the best-fit values for the parameters.

Week: 4 – Fourier series
**Relevant parts of recommended references:** Using MATLAB, chapter 12.

**Specific objectives – after this week you should be able to:**
- Use MATLAB in solution of physics problems involving Fourier series.
- Use the Fast Fourier Transform (FFT) functions in signal analysis.

Weeks: 5 and 6 – Analysis and interpretation of large data sets
**Relevant parts of recommended references:** Using MATLAB, chapter 6.

**Specific objectives – after this week you should be able to:**
- Load large data sets generated during computer simulations of atomic systems.
- Use MATLAB to visualize, animate and analyse the data to provide physical insights and interpretation.

---

**C component 2001/2901**

Weeks: 7 – Introduction to C
**Relevant parts of recommended references:** Programming with C (Shaum’s) chapters 1-6.

**Specific objectives – after this week you should be able to:**
- Understand the basics of programming and the C programming language.

Week: 8 – Stability and chaos
**Relevant parts of recommended references:** Programming with C (Shaum’s) chapters 6-9.

**Specific objectives – after this week you should be able to:**
- Understand numerical stability with regards to non-linear finite difference equations.
- Write code to simulate the stability of non-linear finite difference equations.
- Write codes to recreate both the Julia and Mandelbrot fractal sets.
- Identify and communicate areas where fractals are seen in nature.

Weeks: 9 – Integration and probability
**Relevant parts of recommended references:** Scientific Computing, chapter 8.

**Specific objectives – after this week you should be able to:**
- Understand random events and the distributions that model them (Binomial, Poisson and Gaussian).
- Understand and be able to implement both the trapezoidal and Simpson’s methods of numerical integration.
- Understand the truncation error associated with each of these methods of integration.
- Create libraries in C that perform numerical integration and generate probability distributions.
- Apply numerical integration to problems involving random events.

Weeks: 10 – Brownian motion and random numbers
**Relevant parts of recommended references:** Scientific Computing, chapter 13.

**Specific objectives – after this week you should be able to:**
- Understand and be able to create pseudo-random numbers.
- Understand the basic theory and characteristics of Brownian motion.
- Write code to simulate Brownian motion.
- Identify and communicate applications of Brownian motion.
**Weeks: 11 - 13 – Ordinary differential equations (ODEs)**

**Relevant parts of recommended references:** Scientific Computing, chapters 9 & 10.

**Specific objectives – after this week you should be able to:**

- Recognise ordinary differential equations.
- Understand and solve initial value problems for ODEs.
- Understand and solve boundary value problems for ODEs.
- Understand and solve problems involving stiff ODEs.
- Understand and be able to implement Euler’s method to solve first order ODEs.
- Write codes to model scientific problems involving first order ODEs.
- Write codes to more accurate integrate ODEs than Euler’s method.
- Understand the method of treatment of higher order ODEs.
- Write codes to model scientific problems involving higher order ODEs.
- Understand the importance of numerical accuracy and stability in the solution of ODEs.
- (Advanced) Investigate problems associated with non-linear ODEs.