PHYS703

Computational Science

S1 Day 2014

Physics and Astronomy

Contents

General Information

Unit convenor and teaching staff
Other Staff
Annabelle McIver
annabelle.mciver@mq.edu.au
Contact via annabelle.mciver@mq.edu.au
Wednesday 3pm--4pm; other times by appointment.

Unit Convenor
Jason Twamley
jason.twamley@mq.edu.au
Contact via jason.twamley@mq.edu.au
E6B 2.612
Thursdays 10--11am; other times by appointment.

Credit points
4

Prerequisites
Admission to MRes

Corequisites

Co-badged status

Unit description
Computational techniques are a critical aspect of modern physics, science and engineering. They sit apart from theoretical and experimental physics but borrow characteristics from both. The aim is to turn a computer into a virtual laboratory for research, that allows breakthroughs and insights from what would otherwise be intractable problems by analytical methods. This unit focuses specifically on the computational techniques for solving problems in physics, engineering and science in general. It is not a course in programming though a low level of programming ability will be required to practice the techniques. Topics to be covered: o Introduction to Python and the Python scientific environment. o ODEs: Euler, Runge-Kutta and adaptive techniques, examining accuracy and stability with examples drawn from planetary science and chaotic systems. o Spectral methods: systems of linear equations, spectral analysis and analysis of normal modes. o PDEs: Initial and boundary conditions, discretisation. Relaxation and implicit schemes. Examples of Poisson, diffusion and wave equations. o Monte Carlo methods: random numbers, Monte Carlo integration, random walks, Metropolis algorithm. Examples of Ising model and phase transitions. o Convex optimisation: convex sets and functions, optimisation problems, linear and quadratic programming, duality.

Important Academic Dates

Information about important academic dates including deadlines for withdrawing from units are available at https://www.mq.edu.au/study/calendar-of-dates

Learning Outcomes

On successful completion of this unit, you will be able to:

This unit is designed to give the students a tool set of computational techniques that will be useful in their research and future careers. Students will be able to:
understand the framework of techniques and which techniques would be most applicable to a given problem.
demonstrate an understanding of the methods covered.
apply their knowledge to solve computational problems, creating and testing the necessary routines.
understand how to analyse and evaluate the accuracy of a computational routine.
develop skills in basic programming, code organisation
understand at an introductory level key concepts in modern scientific programming - classes, parallel techniques

Assessment Tasks

Name	Weighting	Due
Assignments	25%	TBA
In-Lab Assessment	25%	Weekly
Project	25%	Week 12
Final Examination	25%	Exam Weeks

Assignments

Due: **TBA**
Weighting: **25%**

The assignments will comprise of 3-4 questions designed to engage the students with

the material as it's covered. The difficulty of the questions will be set so that the assignment would take on

average around 7 hours to complete ; mix of theory and programming work

On successful completion you will be able to:

- This unit is designed to give the students a tool set of computational techniques that will be useful in their research and future careers. Students will be able to:
- understand the framework of techniques and which techniques would be most applicable to a given problem.
- demonstrate an understanding of the methods covered.
- apply their knowledge to solve computational problems, creating and testing the necessary routines.
- understand how to analyse and evaluate the accuracy of a computational routine.
- develop skills in basic programming, code organisation
- understand at an introductory level key concepts in modern scientific programming - classes, parallel techniques

In-Lab Assessment

Due: **Weekly**
Weighting: **25%**

Laboratory exercises (computer laboratory; based on structured class work sheets and note books)

On successful completion you will be able to:

- This unit is designed to give the students a tool set of computational techniques that will be useful in their research and future careers. Students will be able to:
- demonstrate an understanding of the methods covered.
- apply their knowledge to solve computational problems, creating and testing the necessary routines.
- understand how to analyse and evaluate the accuracy of a computational routine.
- develop skills in basic programming, code organisation
- understand at an introductory level key concepts in modern scientific programming - classes, parallel techniques

Project

Due: **Week 12**
Weighting: **25%**

Individual projects - 4 weeks at the end of the semester with close supervision and a final report/ in class presentation

On successful completion you will be able to:

- This unit is designed to give the students a tool set of computational techniques that will be useful in their research and future careers. Students will be able to:
- understand the framework of techniques and which techniques would be most applicable to a given problem.
- demonstrate an understanding of the methods covered.
- apply their knowledge to solve computational problems, creating and testing the necessary routines.
- understand how to analyse and evaluate the accuracy of a computational routine.
- develop skills in basic programming, code organisation

Final Examination

Due: **Exam Weeks**
Weighting: **25%**

Final exam based on a combination of written and oral examination

On successful completion you will be able to:

- This unit is designed to give the students a tool set of computational techniques that will be useful in their research and future careers. Students will be able to:
- understand the framework of techniques and which techniques would be most applicable to a given problem.
- demonstrate an understanding of the methods covered.
- apply their knowledge to solve computational problems, creating and testing the necessary routines.
- understand how to analyse and evaluate the accuracy of a computational routine.
- develop skills in basic programming, code organisation

Delivery and Resources

- New and Improved Unit Description available at http://physics.mq.edu.au/current/mres/phys703/
- Textbook requires is called "Introduction to Computational Science", Angela B. Shiflet and George W. Shiflet and is available in the on-campus bookstore
- All the material will be available via ilearn
- Course will use Matlab and Vensim (Systems Dynamics simulation: free for educational use)

Unit Schedule

Wed: 14:00-16:00 Location: EMC G230 (Note this is a lecture in the EMC building)

Monday 14:00-17:00 Location EMC G230 (Labs)

The Lecture will start in Week 1 of S1 but the Labs will start in Week 2

Policies and Procedures

Macquarie University policies and procedures are accessible from Policy Central. Students should be aware of the following policies in particular with regard to Learning and Teaching:

Academic Honesty Policy http://mq.edu.au/policy/docs/academic_honesty/policy.html

Assessment Policy http://mq.edu.au/policy/docs/assessment/policy.html

Grading Policy http://mq.edu.au/policy/docs/grading/policy.html

Grade Appeal Policy http://mq.edu.au/policy/docs/gradeappeal/policy.html

Grievance Management Policy http://mq.edu.au/policy/docs/grievance_management/policy.html

Disruption to Studies Policy http://www.mq.edu.au/policy/docs/disruption_studies/policy.html *The Disruption to Studies Policy is effective from March 3 2014 and replaces the Special Consideration Policy.*

In addition, a number of other policies can be found in the Learning and Teaching Category of Policy Central.

Student Code of Conduct

Macquarie University students have a responsibility to be familiar with the Student Code of Conduct: https://students.mq.edu.au/support/student_conduct/

Student Support

Macquarie University provides a range of support services for students. For details, visit http://students.mq.edu.au/support/

Learning Skills

Learning Skills (mq.edu.au/learningskills) provides academic writing resources and study strategies to improve your marks and take control of your study.

- Workshops
- StudyWise
- Academic Integrity Module for Students

- Ask a Learning Adviser

Student Services and Support

Students with a disability are encouraged to contact the Disability Service who can provide appropriate help with any issues that arise during their studies.

Student Enquiries

For all student enquiries, visit Student Connect at ask.mq.edu.au

IT Help

For help with University computer systems and technology, visit http://informatics.mq.edu.au/help/.

When using the University's IT, you must adhere to the Acceptable Use Policy. The policy applies to all who connect to the MQ network including students.