CITS4403 Computational Modelling
Lectures for Semester 1, 2017
Overview and Administration

Unit coordinator: Mark Reynolds

March 6, 2017
This unit explores current research topics in computational modelling. Students develop skills to identify problems, formulate solutions and conduct further research in open questions in this domain.
On completion of this unit, students will be able to:

- present computational models to address a given research hypothesis and qualify the limitations of these models;
- identify and discuss current open research topics in the field of computational modelling;
- explain the concepts and technologies used in the field of computational modelling; and
- design, apply and analyse relevant technologies to solve problems in the field of computational modelling.
<table>
<thead>
<tr>
<th>Week</th>
<th>Thursday</th>
<th>Lecture Thursday 3pm</th>
<th>Anything Due</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Mar 2</td>
<td>Introduction to computational modelling</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Mar 9</td>
<td>Complexity Science Overview</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Mar 16</td>
<td>Graphs</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Mar 23</td>
<td>Analysis of Algorithms</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Mar 30</td>
<td>Small World Graphs</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Apr 6</td>
<td>Scale-free Networks</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Apr 13</td>
<td>Cellular Automata</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Apr 20</td>
<td>NO LECTURE (mid-semester study break)</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Apr 27</td>
<td>Game of Life</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>May 4</td>
<td>Agent-based Modelling</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>May 11</td>
<td>Self-organised Criticality</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>May 18</td>
<td>Modelling Complex Systems: Guest Lecture</td>
<td>Assignment Due Wednesday (24/5)</td>
</tr>
<tr>
<td>13</td>
<td>May 25</td>
<td>Student Presentations</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>June 1</td>
<td>Exam Information and Sample Exam</td>
<td></td>
</tr>
<tr>
<td></td>
<td>June 8</td>
<td>Study Break</td>
<td>Notebook Due (5/9)</td>
</tr>
</tbody>
</table>
Answers to everything:
People and Places

Unit coordinator: Professor Mark Reynolds
Consultation: Thursdays 11am to 12 noon in CSSE Rm 1.31B (via CS reception)

Other Lecturer: Dr John McCabe-Dansted
Consultation: TBA

Students must attend one two hour lecture each week:

<table>
<thead>
<tr>
<th>Type</th>
<th>Time</th>
<th>Day</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lecture</td>
<td>9am</td>
<td>Thursday</td>
<td>ENCM105</td>
</tr>
</tbody>
</table>
Free Online Textbook

*Think Complexity: Complexity Science and Computational Modeling* by Allen B. Downey Published: February 23, 2012

Python

Programming and examples for this unit will be undertaken in Python.
It is recommended to use the latest 2.X version to be consistent with the text book examples. (Not 3.X).
We will also use the Jupyter notebooks as a programming and presentation environment.
Coursework

Your work in this course will include:

- Reading from the textbook and popular books on related topics.
- Exercises from the book.
- Explorations and experiments.
- Diagnostic quizzes.
- Writing about the books you read and experiments you perform (in your assignment and exam).
- Brief class presentations of relevant book chapter reviews.
- A final report (assignment) in the form of a book chapter.
- Submit notebooks explaining and exploring the topics covered in lectures.
<table>
<thead>
<tr>
<th>Type</th>
<th>Percent</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assignment</td>
<td>30%</td>
<td>Wednesday 24th May (at any time)</td>
</tr>
<tr>
<td>Notebook</td>
<td>20%</td>
<td>Monday 5th June (at any time)</td>
</tr>
<tr>
<td>Exam</td>
<td>50%</td>
<td>June exam period</td>
</tr>
</tbody>
</table>

The *Assignment* is a Case Study, and it may be done in pairs. More details will be given in week 5. Students work for 6 weeks on a series of experiments and present them in the form of a short report. The final output should be similar to a case-study chapter from the textbook.

The *Exam* is a 2 hour written exam. More details in week 13.
The Jupyter Notebook http://jupyter.org/ is a web application that allows you to create and share documents that contain live code, equations, visualizations and explanatory text. For each lecture, students are required to read the corresponding chapter, work through and extend the code examples. Students should prepare a brief synopsis of the chapter, link to relevant literature they have found and include their code in a Jupyter notebook. The summaries can be of any 2 of the standard lectures (weeks 3-11).

Each summary will be marked on:

- Presentation (20%)
- Code (40%)
- Explanation (20%)
- Investigation (20%)
Plagiarism

Do not submit the work of other people.

Penalties are severe.

If you are unsure of a particular situation, please check http://www.student.uwa.edu.au/learning/studysmarter/online/plagiarism
Link to Help4403 on web page.

https://secure.csse.uwa.edu.au/run/help4403

Or see lecturer.
This course and the text book are about complexity science, data structures and algorithms, intermediate programming in Python, and the philosophy of science.

Note: much of the material in lectures is from the text book and the associated course material.
A data structure is a collection that contains data elements organized in a way that supports particular operations. For example, a dictionary organizes key-value pairs in a way that provides fast mapping from keys to values, but mapping from values to keys is generally slower.

An algorithm is a mechanical process for performing a computation. Designing efficient programs often involves the co-evolution of data structures and the algorithms that use them. For example, the first few chapters are about graphs, a data structure that is a good implementation of a graph—nested dictionaries—and several graph algorithms that use this data structure.
We assume that you have or will quickly catch up with basic knowledge of Python. The course will try to emphasize fundamental ideas that apply to programming in many languages, but along the way you will learn some useful features that are specific to Python.
Philosophy of science:

The models and results in this book raise a number of questions relevant to the philosophy of science, including the nature of scientific laws, theory choice, realism and instrumentalism, holism and reductionism, and Bayesian epistemology.
A model is a simplified description of a system that is useful for simulation or analysis. Computational models are designed to take advantage of cheap, fast computation.

This book focuses on discrete models, which include graphs, cellular automata, and agent-based models. They are often characterized by structure, rules and transitions rather than by equations. They tend to be more abstract than continuous models; in some cases there is no direct correspondence between the model and a physical system.
Complexity science

...is an interdisciplinary field—at the intersection of mathematics, computer science and physics—that focuses on these kinds of models.

That’s what this course and the text book is about.
For next week

Obtain a copy of the text book and read the preface and Chapter 1.

Set up Python (2.X) on your computer. Write a few simple programs to be familiar with the basic syntax.

Set up Jupyter on your computer, and have a look at some interesting Jupyter Notebooks:

Look at the reading list (on web site), obtain/borrow one book from the list and read one to two chapters. Write a summary and prepare to give a two minute presentation on it.
And that’s all for the first lecture.

Enjoy!