Course information

PH-204 is about *Physics Simulation*. Broadly, this means using one physical process (that we can control) to simulate another (in which we're interested).

Historically, mechanical systems were used (e.g. Tide-predicting machines). We now have computers, which are *general purpose* machines which can be *programmed* to simulate any physical system. A notable gap is *quantum mechanics*, for which no known *classical* algorithm exists for efficient simulation. Using quantum systems which we can control (e.g. cold atoms) to simulate other quantum systems in which we're interested (e.g. solid state) is an active area of research. A *general purpose quantum computer* is the obvious next step.

Approach

This course is about using computers to solve physics problems. It is not about computer science, user-interface design, or software engineering. There is not enough time to teach programming thoroughly.

We cover fundamentals and the idea that you are writing instructions which externalize your thought process.

We then focus on recipes which will bootstrap your ability to use programming as a problem solving tool, and hopefully spark an interest.

The course is 100% continuously assessed, and time is extraordinarily tight to set a meaningful first assignment.

We will cover sufficient material before an assignment is set; you may use any additional material learned during the course.

Python

Python is a real language used by science and industry. It is not just a teaching aid. Concepts learned here are applicable to other programming languages.

Python is remarkably flexible: it matters what something can **do**, not what something **is**.

You will see examples of typical use. Please follow through, step-by-step. Appreciate that there are different ways to achieve the same thing, so be creative and try things.

Assessment

Assessment is by in-class competence checks and assessments with the following weighting:

Component	C0	A1a	C1	A1b	C2	A2	C3	A3	C4	Total
Weighting	0	5	5	25	5	25	5	25	5	100

Competence checks

Competence checks ensure that you know the building blocks necessary for the assignment.

If you struggle, we can tackle issues **before** they impact your ability to do the assignment.

Competence checks are intended to be straightforward and take no more than 30 minutes.

Course notes and recommended texts are permitted. **No discussion**. **No Googling**.

Upload to Blackboard as a **working** Python3 script. No Jupyter. No cut-and-paste from the terminal.

Once complete, if you are satisfied, please start the assignment.

If you had difficulty, please discuss with teaching staff over the next 60 minutes.

Assignments

While you are encouraged to discuss problems, as you would any other problem sheet, work which you submit **must be your own**.

If you are struggling, please contact me.

Upload to Blackboard as **working** Python3 scripts. No Jupyter. No cut-and-paste from the terminal.

Copy-and-paste from the notes is **encouraged**: you are not expected to remember all the boilerplate, but you must know enough to use it.

Copy-and-paste from your colleagues is **forbidden**: while you are encouraged to discuss problems, work you submit must be your own.

Broad outline

Part 0 is about Python fundamentals

Part 1 is about thinking algorithmically, functions, and flow control.

Part 2 is about root finding, numerical integrals, interpolation, and fitting to experimental data.

Part 3 is about matrices, randomness, and solving Ordinary Differential Equations.

Part 4 is about symbolic computing, and exploring other useful libraries

Reading material

These webpages are tutorials and useful references. They may be consulted during Competence checks.

During Competence Checks, you may also consult the course notes, your own notes, and the following, but please **no other on-line resources**.

Tutorial; NumPy; SciPy; matplotlib; PH-113 Python Primer; SciPy Lectures; SciPy cookbook

Timetable

Scheduled contact time including Competence Tests (C0,C1,C2,C3,C4) and Assignments (A1,A2,A3).

Attendance is mandatory at all workshops, even when there is no scheduled test.

Assignment deadlines are for illustration only; please refer to individual sheets for exact deadlines.

Week	Date	Event	Description	Assign
1	2019-09-30 Mon	Lecture 0	Welcome; review PH113; fundamentals	
1	2019-10-01 Tue	Workshop 0	Competence 0	
2	2019-10-07 Mon	Lecture 1	imports; root finding	1a out
2	2019-10-08 Tue	Workshop 1	Competence 1	1b out
3	2019-10-14 Mon	Lecture 2.1	arrays; plotting; integration	
3	2019-10-16 Wed	Workshop 2.1	*Practice*	
4	2019-10-21 Mon	Lecture 2.2	interpolation; fitting to data	
4	2019-10-22 Tue	Workshop 2.2	*Practice*	1a due
5	2019-10-28 Mon	Lecture 2.3	loading and saving data	1b due
5	2019-10-29 Tue	Workshop 2.3	Competence 2	2 out
6	2019-11-04 Mon	Lecture 3.1	vectors; matrices; randomness	
6	2019-11-05 Tue	Workshop 3.1	*Practice*	
7	2019-11-11 Mon	Lecture 3.2	ODEs: companion form; solvers	
7	2019-11-12 Tue	Workshop 3.2	*Practice*	
8	2019-11-18 Mon	Lecture 3.3	solvers and accuracy	2 due
8	2019-11-19 Tue	Workshop 3.3	Competence 3	3 out
9	2019-11-25 Mon	Lecture 4.1	Other DEs; multiprocessing; objects	
9	2019-11-26 Tue	Workshop 4.1	*Practice*	
10	2019-12-02 Mon	Lecture 4.2	Introduction to symbolic	
11	2019-12-03 Tue	Workshop 4.2	*Practice*	
11	2019-12-09 Mon	Lecture 4.3	Ising model; more symbolic computing	3 due
11	2019-12-10 Tue	Workshop 4.3	Competence 4	

Contact

Dr James Bateman

613 Vivian Tower

j.e.bateman@swansea.ac.uk

Office Hours: Wednesday 9:00–11:00