CH40208 Topics in Computational Chemistry Semester 1, 2019/20

*Drs B. J. Morgan & A. R. McCluskey*

**Synopsis and Assessment**

Combined lecture & practical session in PC room (CB 4.17) each week (10:15 – 13:05)

2 Oct BJM overview What is computational chemistry? Course philosophy. Outline.

ARM lecture 1 Introduction to Python: Variable types; arithmetic; print and input,

flow control using if, elif, else; logical operators.

Worksheet 1Introduction to Python programming in the Jupyter Notebook environment:  
• Practice exercises: variables, arithmetic, printing, etc.  
• Program to convert °F to °C to K.  
• Program to calculate an equilibrium constant from free energies.

9 Oct ARM lecture 2 Working with data: lists; loops, numpy arrays; plotting; importing data.

Worksheet 2 • Write a program to calculate distances between atoms in a molecule.  
• Revise this to use NumPy arrays for handling your data.

16 Oct ARM lecture 3 Functions and Modular code.

Worksheet 3  Program to calculate Lennard-Jones interaction energies;

• Program to optimise interatomic distances.

23 Oct ARM lecture 4 Debugging: Reading error messages and finding help.

Worksheet 4 • Molecular rotations and rotation matrices.

30 Oct ARM lecture 5 Writing legible and maintainable code: docstrings, testing, and  
“test-driven development”.

Worksheet 5 • Writing code using tests.

6 Nov Workshop 1 Hands-on practice.

13 Nov Workshop 2 Hands-on practice.

20 Nov BJM lecture 1 Introduction to linear algebra; vectors and matrices.  
Revisiting molecular rotations; moments of inertia; finding principal axes  
of rotation (eigenvalues and matrix diagonalization).

27 Nov BJM lecture 2Finding normal modes of vibration and their frequencies.

Solving the Schrödinger equation → application to Hückel theory.

4 Dec Test 1 10:15 – 13:05 (CB 4.17) TDD exercises (50% total mark)

11 Dec Test 2 10:15 – 13:05 (CB 4.17) Single multistep problem (50%)

***Assessment***

Formative assessment:

* Lecture weeks (1–5, 8, 9): Includes time to work through small exercises, and one or more larger programming exercises to do (demonstrators available to answer questions and discuss).
* Workshop weeks (6 & 7): No lecture. Instead you will be given bigger problems to solve than in the previous weeks and asked to think about how to solve these / write the code (with demonstrators / advice / help available).

Summative assessment:

* Test 1 (week 10): Open-book test; working through a series of small exercises   
  (“test-driven development”)   
  → 50% of the total mark
* Test 2 (week 11): Open-book test; single multistep problem to work through.   
  → 50% of the total mark