Course Title: Computational Methods (using Python)
Course Lecturer: Mr. Mayhew Steyn
Course credits: 2
Lecturer contact hours: 48
Tutorial/practical hours: 12

1) Course overview:
In this course we will cover the basics of Python, a few popular packages relevant to astronomy and physics, and general numerical methods used in science. The focus of this course is on the practical application of this programming knowledge, and most of the contact time will be dedicated to working through exercise and tutorials.

Each lecture will consist of a pre-reading component from the course notes, an in-class revision of this, and time to work on exercises related to the readings (not assessed). The tutorial sessions will be hosted by the tutors and present an opportunity for you to work on the weekly tutorials (assessed) and any other outstanding work.

The other assessments in this module are 2 projects. Each of these consists of a problem that you will be tasked with solving using Python programming. Submissions for these projects include both Python code and a write-up typeset using LaTeX. Project1 will be released half-way through the semester, and Project 2 will be released during the examination period.

2) Course breakdown/syllabus:
- Use of the Python Standard Library:
    - Variables, operators, data types and data structures.
    - Control flow: If Statements and Loops
    - Functions
    - File input/output
- Scientific Python Packages:
    - Numpy (arrays, special functions, and random numbers)
    - Matplotlib (plotting)
    - Scipy (curve fitting, numerical algorithms)
    - Astropy (astronomy focused, dimensional analysis, image processing, etc)
- Numerical Methods:
    - Linear regression and curve fitting with Scipy
    - Solving ordinary-differential equations (Euler and Runge-Kutta methods)
    - Integration (midpoint, trapezoidal and Simpson`s rules)
    - Root finding (bisection, secant and Newton-Raphson methods)

3) Resources:
Course notes (website and PDF formats) and all other resources provided.

4) Breakdown of practicals/tutorials:
In the tutorial sessions you will be given open time to work on the tutorials and exercises (which cover the lecture content) under the guidance of the tutors.

5) Additional skills to be developed during the course:
Python programming (writing original code), scientific writing/reporting (using LaTeX), numerical problem solving (transforming physics into code).

6) Assessment

- 10 tutorials: 25%
- Project 1: 25%
- Project 2: 50%