PH30110 Computational Astrophysics SR Davies

Coursework 1. (Submit by 11:00am, Wednesday, 3rd March, via Moodle page)

- These exercises should help to **improve your programming ability** as well as **assess** it.
- Your coursework submission should take the form of a **report**. It should include printouts of your **Python programs** and their results and discussion of the approach/methods used to **design** and **test** your programs.
- **60% of the marks** will be given for **a satisfactory report, with sound, working programs**.
- **Additional marks** will be given for **good program design and programming style** (10%), for **more sophisticated programs** (10%) and for better **presentation and discussion of their results** (20%).
- Always make sure that you get a good, straightforward program working first, before you try to get it to do something more complicated. Any program that partly carries out the given task will be worth plenty of marks, even if it does not fully do the job.
- **Electronic copies of the report and the main programs** should be submitted. Details will be posted on the Moodle page.
- **Note. Your report and your programs must be your own work**. University regulations require that all your submitted work will be tested with anti-plagiarism software and anyone suspected of plagiarism or collusion will undergo the University's formal disciplinary process.

NOTE: This is the first half of Coursework 1, which you should work on prior to PC Lab 4. The second part of Coursework 1 will be added to this document after completing Lab 4.

Q1. Total 25 Marks

(a). This is based upon Ex. 6 from PC Lab 2.

Consider the Sun, of mass *M*, to be stationary at the origin, with an orbiting comet, of mass *m*, at position vector $\vec{\boldsymbol{r}}$. If we orient our coordinate system so that the comet is in the *z* = 0 plane, then we can ignore *z* and let $r = \sqrt{x^2 + y^2}$. The force between the Sun and the comet will be in the direction $-\vec{\boldsymbol{r}}/r$, i.e. towards the Sun and hence

$$m\,\frac{d^2\vec{\boldsymbol{r}}}{dt^2} = -\left(\frac{GMm}{r^2}\right)\frac{\vec{\boldsymbol{r}}}{r}$$

At its largest distance from the Sun (aphelion), Halley's comet is at a distance of 5.2 billion km with a speed of 880 $m.s^{-1}$. Develop a Python program to calculate the orbit of the comet. Test your program and discuss the results in your report.

(15 marks)

(b). This is a more open-ended exercise. There are several ways in which your basic program could be improved, some of which have been discussed in class and many of which have not. One such improvement would be to introduce an adaptive time step, so that the time intervals are more closely spaced for the faster-moving portions of the orbit. Other improvements are available. Develop a more sophisticated version of your program from the baseline version resulting from part (a), by incorporating at least one such improvement.

(10 marks)

NOTE: This is the second half of Coursework 1, which you should work on after PC Lab 4.

Q2. Total 25 Marks

(a). This is based upon Ex. 3 from PC Lab 4.

Set up a computational model of a simple, 3-body problem consisting of two objects orbiting around a fixed, central mass. The central object is a star of mass *M* (one solar mass), orbited by two planets with masses given by

$$m_1/M = 10^{-3}\,, \qquad m_2/M = 4 \times 10^{-2}$$

Model the orbits of the planets using a numerical algorithm (e.g. RK4). Start with circular orbits at values a_1 = 2.52 AU and a_2 = 5.24 AU. You will need to work out suitable initial values for the RK4 routine. Investigate what the trajectories look like.

Now model the orbits (numerically) of Jupiter and Saturn (see PC Lab 4 for the planetary data) and investigate what the trajectories look like.

(15 marks)

The final part of the coursework is more challenging, but only worth 20% of the total marks. Make sure you have done a good job on the previous parts of the coursework (worth 80% of the marks) before spending much time on this part.

(b).

Try to transform your coordinate system to a frame based upon the centre-of-mass of the system. One approach was discussed in Lecture 4. A similar approach would be to think about the position and velocity vectors for the centre-of-mass:

$$\vec{r}_{cm}(t) = \sum_i \frac{m_i}{M_{total}} \vec{r}_i(t) \qquad \text{and} \qquad \vec{v}_{cm}(t) = \sum_i \frac{m_i}{M_{total}} \dot{\vec{r}}_i(t)$$

and how to subtract them from $\vec{r}_i(t)$ and $\dot{\vec{r}}_i(t)$. If you succeed in implementing this, you should be able to simulate more complicated systems (one example would be a binary star system with an orbiting exoplanet).

(10 marks)