PHAS0007 Computing Final Assignment 2022-23: Simulating Planetary Orbits

Dr Ben Waugh, Dr Rebecca Chislett, Dr Louise Dash

Revision date: December 16, 2022

1 Introduction

You will take the programming skills you have developed throughout the course and use them, combined with your knowledge of classical mechanics, to create animations of planets orbiting a star.

The assignment is divided into two parts:

- You can ask questions and get advice from us on part A in the drop-in sessions in the last week of term. However, the work you submit must be your own, and you must not collaborate with anyone else.
- Part B must be completed independently. You must work on this entirely on your own, without the assistance of other students, course staff or anyone else.

Everything in your submission must be either completely your own work, or have the source explicitly acknowledged and suitably referenced. We will be using several methods to check for similarities between work submitted by different students, and investigating any indications of plagiarism or collusion.

2 The physics

2.1 Newton's law of universal gravitation

Newton's law of universal gravitation states that the gravitational force between any two objects is proportional to their masses, and inversely proportional to the square of the distance between them:

$$F = \frac{Gm_1m_2}{r^2}$$

In SI units, the gravitational constant $G = 6.674 \times 10^{-11}$ N m^2 kg^{-2}. This is very small: gravity is a very weak force compared to electromagnetic and nuclear forces.

Figure 1 illustrates Newton's law of universal gravitation. A point mass m_1 attracts another point mass m_2 by a force F_2 pointing along the line intersecting both points. In the same way, m_2 attracts m_1 with an equal force F_1 pointing in the opposite direction.

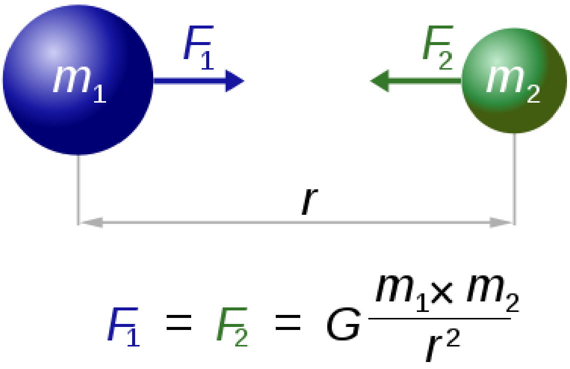

Figure 1: Illustration of Newton's law of gravitation. Created by Dennis Nilsson.

Consider two masses m_1 and m_2, as shown in the figure. The gravitational force exerted on m_1 due to m_2 will be

$$\mathbf{F}_{12} = -G\frac{m_1 m_2}{|\mathbf{r}_{21}|^2}\hat{\mathbf{r}}_{21},$$

where $\mathbf{r}_{21}$ is the position vector from m_2 to m_1, i.e. $\mathbf{r}_1 - \mathbf{r}_2$
Similarly, the force exerted on m_2 due to m_1 will be

$$\mathbf{F}_{21} = -G\frac{m_1 m_2}{|\mathbf{r}_{12}|^2}\hat{\mathbf{r}}_{12}$$

and clearly $\mathbf{F}_{12} = -\mathbf{F}_{21}$: the forces are equal and opposite, as they should be according to Newton's third law.

From Newton's second law, we also have $\mathbf{F} = m\mathbf{a}$, so therefore we can see that the accelerations experienced by m_1 due to m_2 and vice versa will be

$$\mathbf{a}_1 = -\frac{Gm_2}{r_{12}^2}\hat{\mathbf{r}}_{12}$$

$$\mathbf{a}_2 = -\frac{Gm_1}{r_{21}^2}\hat{\mathbf{r}}_{21}$$

These accelerations will not be equal in magnitude unless the objects have equal masses.

2.2 Converting Newton's laws to a numerical model

Let's consider a system with two objects: a star with mass M and a planet with mass m. In a solar system like ours $M \gg m$ and so the acceleration experienced by the star due to the planet will be small compared to the acceleration experienced by the planet due to the star. The gravitational force on the planet due to the star is then

$$\mathbf{F}_{mM} = -GMm\frac{\hat{\mathbf{r}}_{Mm}}{|\mathbf{r}_{Mm}|^2}$$

To keep things simple, we will assume the acceleration on the star due to the planet is small enough that we can neglect it, and put the star in a fixed position at the origin of our co-ordinate system. Thus $\mathbf{r}_{Mm} = \mathbf{r}_m - \mathbf{r}_M$ becomes $\mathbf{r}_m$. We'll drop the now unnecessary subscript on $\mathbf{r}_m$, and replace the unit vector $\hat{\mathbf{r}}$ with $\mathbf{r}/|\mathbf{r}|$, to give a simpler-looking expression:

$$\mathbf{F}_{mM} = -GMm\frac{\mathbf{r}}{|\mathbf{r}|^3}.$$

We can work out the motion of the planet straight from this, just by using the laws of motion:

$$\mathbf{F}_{mM} = m\mathbf{a} = m\frac{d\mathbf{v}}{dt} = m\frac{d^2\mathbf{r}}{dt^2}$$

In order to calculate the motion of the planet around the star, we need to take the initial position and velocity of the planet as the starting positions, and integrate in order to calculate the force. Here we will not do this exactly but use a numerical method to get an approximate result. This approach is particularly useful in cases where it is not possible to find an exact solution, e.g. when we are taking general relativity into account or dealing with more than two bodies. Here we use Euler's method, as in unit 9. This is the simplest, most naive method and definitely not the most accurate, as you will learn in later computing courses.

We replace the differentials to small finite differences and rewrite the force as

$$\mathbf{F}_{mM} = m\frac{d\mathbf{v}}{dt} \approx m\frac{\delta\mathbf{v}}{\delta t}$$

and then rearrange this to give the change in velocity of the planet in the time δt as

$$\delta\mathbf{v} = \frac{\mathbf{F}_{Mm}}{m}\delta t = -GM\frac{\mathbf{r}}{|\mathbf{r}|^3}\delta t$$

where $\delta\mathbf{v}$ and δt are very small finite quantities. We can define δt (usually known as the timestep) in our code as a parameter.

Similarly, we can write the change in the planet's position over the same δt as

$$\delta\mathbf{r} = \mathbf{v}\delta t$$

Given our starting point of the initial position and velocity of the planet, and a fixed timestep δt, we can calculate how $\mathbf{r}$ and $\mathbf{v}$ change:

$$\mathbf{r}(t+\delta t) = \mathbf{r}(t) + \delta\mathbf{r}$$

$$= \mathbf{r}(t) + \mathbf{v}\delta t$$

$$\mathbf{v}(t+\delta t) = \mathbf{v}(t) + \delta\mathbf{v}$$

$$= \mathbf{v}(t) - \frac{GM\mathbf{r}}{|\mathbf{r}|^{\mathbf{3}}}\delta t = \mathbf{v}(t) - \frac{GM\hat{\mathbf{r}}}{|\mathbf{r}|^{\mathbf{2}}}\delta t$$

We can use these equations to calculate the path of the planet, and create an animation of this, just as we simulated the path of a projectile with air resistance in unit 9.

3 Task instructions

3.1 Part A (40% of marks)

- Download the template notebook provided.
- Complete and edit this template to create your own self-contained notebook, completing the tasks described.
- The template includes code cells and functions for you to complete to
  - calculate the gravitational force between two objects;
  - update the position of the planet in each time step;
  - loop over a number of steps to calculate the trajectory of the planet;
  - display an animation of a planet orbiting a star.
- You will also need to complete the introduction text cell, and add further code and text cells in the "investigation" section.
- The template also includes some additional code to help you ensure each function is working correctly. You should leave the code in these cells unchanged. You may add your own tests if you wish, but these should be in separate cells.

3.2 Part B

3.2.1 Two non-interacting planets (10% of marks)

- By copying and adapting code from the supplementary notebook, create an animation of two planets orbiting a single star. Use the following parameters:
  - mass of star: 2.5×10^{30} kg
  - initial position of first planet: $(2, 0) \times 10^{11}$ m
  - initial velocity of first planet: $(0, 2.89) \times 10^{4}$ m
  - initial position of second planet: $(1.56, 1.56) \times 10^{11}$ m
  - initial velocity of second planet: $(-1.95, 1.95) \times 10^{4}$ m
- Describe and comment on the behaviour in a text cell.

3.2.2 Two planets with gravitational interactions (35% of marks)

- Create a new function, similar to `move_planet`, but this time updating the positions of two planets and taking into account the gravitational attraction between the planets as well as between each planet and the star.

In addition to the positions and velocities of the two planets, this function will need to take two more arguments to represent the masses of the planets. It will need to return the updated position and velocity of both planets.

Your function should call the force function from part A: you do not need to repeat or change this function.

- Create a new function, similar to `trajectory`, but this time calculating the trajectories of two planets, using the function you have written in the previous section to update the position and velocity of the planets taking into account the gravitational attraction between the planets.

- Using these functions, and any others you need, create an animation showing the trajectories of two planets with the same starting parameters as those in section 3.2.1, and using a mass of 6×10^{24} kg for each planet.

- Create another animation showing the trajectories of two planets with the same starting parameters, but this time increasing the mass of the second planet by a factor of 100.

3.2.3 Further investigation (15% of marks)

Extend your notebook to simulate some further scenarios and investigate the resulting behaviour. Add a few relevant animations and text cells to describe and explain what you observe, along with a conclusion.

There are many aspects of orbits you could look into, but you should pick one or two to investigate in detail. Some ideas that you could consider, of varying levels of difficulty, are:

- vary the masses of the star and planets;

- vary the starting positions and velocities of the planets;

- calculate the required velocity to achieve a stable circular orbit with a given radius, and see how close or how massive the planets have to be to have a significant effect on each other's orbits;

- turn one of the planets into a moon orbiting the other, by careful choice of masses and initial conditions;

- simulate a solar system with more than two planets;

- use a different procedure to calculate the trajectories, instead of Euler's method.

4 Strategies and hints

You are recommended to break each problem down into small steps. It is easier to get a small, simple piece of code working and then build on it than to write a larger section in one go and then have to debug it. In part A we have helped by splitting the overall task into three functions for you to complete one at a time. In part B it is up to you how you manage the process.

Make sure you explain all choices you make in your implementation, especially in the comments. If you use an unusual coding approach, you **must** explain it clearly in the comments, and state why you have chosen this approach, otherwise you may not receive full credit for it. You must also ensure that if you reuse or repurpose any code from another source (including your own coursework submissions) you include a reference to this in both the comments and the bibliography.

As usual, you are recommended to comment your code as you go along, as this forms an essential part of the debugging procedure. You may also find it useful to enable line numbering in your code cells (press ESC-L in a code cell).

Before you submit your code, make sure that you have tested it fully.

You are **strongly advised** not to attempt to complete this assignment in a single coding session. Plan your working time sensibly to allow enough time for completion before the published deadline.

5 Assessment

Everything in your submission must be either **completely your own work**, or have the source **explicitly acknowledged and suitably referenced**. We will be using several methods to check for any plagiarised or copied assignments.

5.1 Anonymity

As for the mid-term assignment, this assignment will be graded anonymously. Please ensure that

- your name does not appear anywhere in your submission;
- if you need to include a reference to one of your own submissions in your notebook, you should replace your name with your student number in the reference.

It is your responsibility to ensure that you submit the correct file. After uploading your notebook, you should download it again and run the downloaded version right through to ensure it works as expected.

Your work will be graded using a mark scheme based on the following assessment criteria:

- whether your calculations are correct;
- the quality of your code, including structure, layout, style, and comments;
- the quality and presentation of your notebook including referencing, discussion, and any conclusions you have drawn from your results;
- the appearance of your animations;
- any improvements or extensions that are relevant to the task but go beyond what is required (up to 5%).

Remember that this is a formal assignment: your document should be clearly laid out in sections, with complete, grammatically correct sentences and paragraphs rather than bullet point lists.

You must a bibliography, using the Vancouver referencing style, as for the mid-term assignment. As a minimum, this should include a reference to this script, but you should also include any and all external sources you referred to in order to complete this assignment.