A Spacecraft Voyage, Part 2: From the Earth to the Moon

OBJECTIVES

In this program you will model the motion of a spacecraft traveling from the Earth to the Moon. To do this you will iteratively (repeatedly):

- calculate the gravitational force on the spacecraft by the Earth and the Moon
- apply the Momentum Principle to update the momentum of the spacecraft
- update the position of the spacecraft

You will use your working program to

- explore the effect of the spacecraft's initial velocity on its trajectory
- study the effect of your choice of Δt (deltat) on the accuracy of your predictions

This activity is a practical example of the use of a computer program to do repeated calculations. This problem (three gravitationally interacting objects) cannot be solved any other way — it is possible to write down a set of calculus equations, but they will not have a general symbolic solution!

Resources you will find useful:

- Start from a copy of your previous program to predict the motion of a spacecraft due just to the Earth
- The VPython reference manual (Help menu, choose Visual)

Organization of these instructions:

The general goal of each section will be stated at the beginning of the section. Hints and detailed suggestions for implementation are given in following sub-sections.

Data

Mass of Earth: 6e24 kg	Radius of Earth: 6.4e6 m
Mass of spacecraft: 15e3 kg	Radius of spacecraft: very small (exaggerated in program)
Mass of Moon: 7e22 kg	Radius of Moon: 1.75e6 m
Distance from Earth to Moon: 4e8 m	G=6.7e-11 N·m^2/kg^2

1 Including the effect of the Moon

In the real world, the net force on a spacecraft is rarely due to only one other object. By adding the Moon to your program, you will get to observe the kind of complex motion that the interaction of three or more objects can produce.

Create a sphere to represent the Moon:

Object Name	**Kind of Object**	**Initial Location**	**Radius**	**Color**
`Moon`	`sphere`	4e8 m to the right of the Earth	1.75e6 m	white

Inside your `while` loop, add a calculation of the gravitational force that the Moon exerts on the spacecraft, and calculate the net force due to the force of the Earth and the force of the Moon. Use the net force to update the momentum of the spacecraft.

- **Inside your `while` loop, add a calculation of the gravitational force that the Moon exerts on the spacecraft.**
- **Calculate the *net* force on the spacecraft, and use the *net* force to update the momentum.**
- **Add a check for crashing on the Moon, like your check for crashing on the Earth.**
- **Lengthen the loop time to 60 days, to follow the more complicated orbits that can occur.**
- **Initially, use a step size of 10 seconds.**
- **Find an initial speed that leads to crashing on the Moon.**
- **Find an initial speed that yields a "figure-8" orbit that loops around the Moon before returning to Earth. Make a note of it as a comment in your program.**
- **How sensitive is this to the initial velocity? How much can you change the initial speed and still get a figure-8 orbit?**
- **Play around with the initial speed and see what other kinds of orbits you can find.**

When the spacecraft interacted solely with the Earth (a "two-body" interaction), there were only a few kinds of orbits possible, and they were quite simple curves (circle, ellipse, parabola, hyperbola, straight line). Small changes in the initial velocity typically led to small changes in the orbit. For example, an ellipse merely became a longer ellipse with a larger initial speed.

In the "three-body" interaction of Earth, Moon, and spacecraft, there is a much larger variety of orbits. If just three bodies show complex behavior, no wonder a macroscopic system such as you is highly complex, since you contain an astronomical number of interacting atoms! Also, an extremely slight change in the initial velocity can make a major change in the motion: the orbit is highly sensitive to the initial velocity. The rich variety of types of motion, and the high sensitivity to initial velocity, are typical of complex systems. This is the subject of the relatively new science of "chaos." See the discussion of these matters in the textbook.

2 Answer these questions about a particularly interesting trajectory

- **Give the spacecraft a speed of 3.27 km/s (3.27e3 m/s), headed in the +y direction. What happens?**
- **Why does coming nearly to a stop lead to retracing the path?**
  Explain in terms of the Momentum Principle and the gravitational force law.
- **Discuss your results and answers with someone else in the class.**

3 Accuracy

If you use a very large Δt, the calculation is inaccurate, because during that large time interval the force changes a lot, making the momentum update inaccurate, and the momentum changes a lot, making the position update inaccurate. On the other hand, if you use a very small Δt for high accuracy, the program runs very slowly. In computational modeling, there is a trade-off between accuracy and speed.

How can you tell whether an orbit is accurate? There's a simple test: Make Δt smaller and see whether the motion changes. That is, see whether the orbit changes shape. (Obviously the program will run more slowly).

- You've been using a step size of 10 seconds. Try a step size one-fifth as large (2 seconds). With an initial speed of 3.27 km/s, is the orbit the same? If the orbit is the same using this smaller step size, that implies that a step size less than or equal to 10 seconds is adequately short to give accurate results.
- To see the effects of Δt being too large, make Δt 100 times as large as you had been using (1000 seconds).

4 Answer these questions about the effect of step size on accuracy

- **Does the 10 second step size give an accurate orbit? How do you know?**
- ***Why* does the large step size (1000 seconds) give an inaccurate orbit?**

Verify your program's behavior and results with someone else in the class.

Before turning in your program, restore it to the following state:
10 second step size
initial speed of spacecraft 3.27e3 m/s in the +y direction

5 Extra Credit Challenges: Playing Around

Shoot the spacecraft straight toward the Moon from near the surface of the Earth, starting at location $\langle 1.01 * \text{Earth.radius}, 0, 0\rangle$, with an initial speed of 11 km/s. What happens? Why? What happens if you increase or decrease the initial speed? It's interesting to watch the momentum vector and think about the effects of the forces on the momentum during this motion.

You can make a more realistic model if you let the Moon and Earth orbit each other while the spacecraft is moving. Calculate all the forces among the three bodies, then update all of the momenta before updating any of the positions. That way you calculate the forces in a consistent way at a particular instant. You will have to experiment a bit with the initial velocity to get the nearly circular orbit of the Moon around the Earth (the Earth follows a nearly circular orbit, too, but it is only a small wobble because the Earth's mass is much larger than the Moon's mass.)

You might even try to model the Sun-Earth-Moon system, with or without the spacecraft. A practical difficulty with visualizing the Sun-Earth-Moon system is that the Earth-Moon system is very small compared to the great distance to the Sun, so it's hard to see the details of the Moon's motion. However, you can continually update the center of the scene by resetting `scene.center` inside the loop to follow the Earth-Moon system, and zoom in on this region. Alternatively, you can set `scene.autocenter = True`, which makes the centering automatic.