A spacecraft voyage, part 1: Spacecraft and Earth

OBJECTIVES

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

- calculate gravitational force on the spacecraft by the Earth
- 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.

TIME

You should plan to finish this activity in 45 minutes or less.

GROUP ROLES

Before you begin, you should agree on the responsibilities of the Manager, the Recorder, and the Skeptic for this activity.

Data for this and subsequent programs:

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

You may find it useful to refer to previous programs in which you calculated gravitational forces, and predicted motion iteratively.

PROGRAM SHELL

A program shell (a skeleton program) has been provided in WebAssign, and you can copy this code and paste it into an IDLE window, and save it in My Documents. Remember to give it the extension ".py".

Read through the program shell together. Make sure everyone in the group understands the function of each VPython instructon. Answer the following question before going on:

Q1: Given the initial location and momentum of the spacecraft, what should the path of the spacecraft be?
Q2: How many years will the loop run?

1. Iterative calculations

- **In order to get the spacecraft to move, you must calculate the gravitational force, update momentum, and update position. Because this must be done after each time step, you need to put these calculations inside the loop.**

- **Use the symbolic names defined in the program shell. You may need to add other symbolic names in your calculations.**

- **In your calculations, for readability and ease of making changes, use the position of the Earth in the form `Earth.pos` rather than assuming this position never changes. This makes it easier to modify the program should you wish to let the Earth move.**

- **Do this, and run your program. Is the behavior what you expected?**

2. Detecting a collision

If your spacecraft collides with the Earth, the program should stop. Add code similar to this inside your loop (using the name you defined for the distance between the spacecraft and the center of the Earth):

```
if rmag < Earth.radius:
     break
```

This code tells VPython to get out of the loop if the spacecraft touches the Earth.

3. Answer these questions about the effect of initial velocity on the motion

- **Give the spacecraft a speed of 0.6 km/s (0.6e3 m/s), headed in the +*y* direction. What happens?**
- **Increase the initial momentum slightly until the spacecraft just misses the Earth. Approximately (to 3 significant figures), what is the minimum initial speed required to miss the Earth? What does the orbit shape look like?**
- **Approximately, what initial speed is required to make a circular orbit around the Earth? You may wish to zoom out to examine the orbit more closely.**
- **Approximately, what minimum initial speed is required so that the spacecraft "escapes" and never comes back? You may have to zoom out to see whether the spacecraft shows signs of coming back. You may also have to extend the time in the while statement.**

- **Compare your results with those of a nearby group.**

4. Visualize the momentum vector with an arrow

In a previous program you used arrows to visualize gravitational force vectors. In this program you will use an arrow to visualize the momentum of the spacecraft.

You will create an **`arrow`** before the **`while`** loop and update its position and axis inside the loop, as the spacecraft's momentum changes.. You need to know the approximate magnitude of the momentum in order to be able to scale the arrow to fit into the scene, so just before the loop print the momentum vector:

```
print “p=”, pcraft
```

As an example, you know that the distance between the center of the Earth and the Spacecraft is about 10 Earth radii, or 10*6e6 m = 6e7 m. With your fingers, estimate about how long you'd like the momentum arrow to be in the scene. How long is that distance in the units of your virtual world? What scalar factor would you need to multiply the momentum by to change its magnitude to the number you want? That is the value of your scale factor.

In the #constants section of your program, add this scale factor

```
pscale = ??       ## the value you decided on
```

Also insert the following statement **in the #OBJECTS section of your program (NOT inside the loop)**:

```
parr = arrow(color=color.green)
```

This statement creates an arrow object with default position and axis attributes. You will set these attributes inside the loop.

- **Comment out any print statements which are inside your loop because they slow down plotting.**
- **Inside your loop, update the pos attribute of the parr arrow object to be at the center of the spacecraft, and update its axis attribute to represent the current vector value of the momentum of the spacecraft (multiplied by pscale).**
- **Run the program with an initial velocity that produces an elliptical orbit.**
- **You may have to adjust the scale factor once you have seen the full orbit.**

5. Answer these questions about the changes in the spacecraft's momentum

- **For this elliptical orbit, what is the direction of the spacecraft's momentum vector? Tangential? Radial? Something else?**
- **What happens to the momentum as the spacecraft moves away from the Earth?**
- **As it moves toward the Earth?**
- **Why? Explain these changes in momentum in terms of the Momentum Principle. Be careful: The Momentum Principle does NOT say "big force -> big momentum".**

- **Compare your answers to those of another group.**