# More Physics Simulations: Astronomical Orbits

## I. INTRODUCTION

You have made a simulation of the one-dimensional motion of a mass on a spring — now you are going to make simulations of an astronomical orbits. This will require that you be more careful with the vector nature of the position and velocity of the objects, and the vector nature of the force.

We will start by modeling the orbit of the earth around the sun, and we will make the simplifying assumption that the sun is infinitely massive and doesn't move at all as the earth orbits the sun. The magnitude of the force is

$$F_{\text{grav}} = \frac{G m_{\text{sun}} m_{\text{earth}}}{r^2}, \tag{1}$$

where $r$ is the distance between the sun and the earth. The force is directed from the earth toward the sun. Your Python program must determine a vector that expresses this magnitude and direction so that it can determine a vector acceleration, so that it can update the vector velocity.

**Challenge:** Figure out a way to express the *vector* force in terms of the position *vector* of the earth, and then translate this into Python.

## II. PROGRAM DETAILS

You should not have to change much in your numerical integration `while` loop. Here are some values (in MKS units) that you will need to use in your program:

- $G = 6.67 \times 10^{-11}\,\text{N}\cdot\text{m}^2/\text{kg}^2$
- $m_{\text{earth}} = 5.97 \times 10^{24}\,\text{kg}$

- $m_{\text{sun}} = 1.99 \times 10^{30}\,\text{kg}$
- Radius of earth's (nearly circular) orbit: $1\,\text{AU} = 1.5 \times 10^{11}\,\text{m}$

Here are some programming suggestions:

- Use a time step $\Delta t$ of $1\,\text{day} = 24 \times 60 \times 60\,\text{s}$.
- Make sure that you choose the sizes for your sun and earth to be large enough to make them visible. Remember that your simulation will display a *very* large area.
- Put the sun at the origin.
- Start with the earth on the $x$-axis.
- Give the earth an initial velocity that is in the $y$-direction. Choose a magnitude that is consistent with what you know about the earth's motion around the sun.
- Make sure that your force function returns a vector.
- You can make the earth "leave a trail." If you defined an object with the name `earth`, you can turn on the trail feature with the command:
  `earth.trail = curve(color=earth.color)`
- The size of the field of view is usually set automatically; it is determined by the graphical objects you place in the scene. You might want to set the scene size yourself with the command
  `scene = display(width = 500, height = 500, range = NUMBER HERE)`

## III. FORCE FUNCTION

Here's one way to code a gravitational force function:

```
def force(r):
    f = -G*mS*mE*r/dot(r,r)**1.5
    return f
```