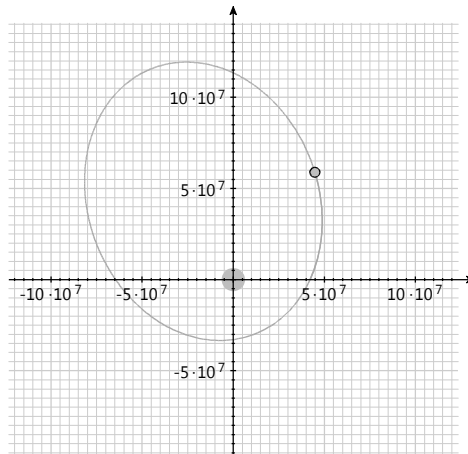


## Exercise Sheet 8

### Satellite



A satellite is moving near Earth. Gravity forces the satellite into an orbit around the earth. It is assumed that the earth and satellite are point shaped masses. The Earth is in the origin of the coordinate system.

The mass of the earth is  $m_E = 5.974 \cdot 10^{24} \text{ kg}$  the mass of the satellite is  $m_S = 100 \text{ kg}$ . Gravity produces a force of attraction between  $m_E$  and  $m_S$ . This force is:

$$F_G = G \frac{m_E m_S}{r^2}$$

As the earth has a very large mass in comparison to the satellite, the earth experiences only a very, very small acceleration due to this force. Simplifyingly, it is supposed that the earth is located in the origin of the coordinate system and does not change its position.

#### Preparation

As in exercise sheet 6, the class *MechanicsTVG* shall be used to visualize the satellite's trajectory. Include the library *mechanics.jar* in your project, if you have not already done so. The file *mechanics.jar* can be found on the website in the area of this exercise.

Copy the following program code:

```

import static java.lang.Math.*;
import mechanics.tvg.MechanicsTVG;
import de.physolator.usr.*;
import de.physolator.usr.tvg.Shape;
import de.physolator.usr.util.Colors;

public class Satellit extends PhysicalSystem {

    public void initGraphicsComponents(GraphicsComponents g, Structure s, Recorder r,
        SimulationParameters sp) {
        MechanicsTVG t = new MechanicsTVG(this, s, r) {
            public void paint() {
                super.paint();
                style.fillColor = Colors.grey;
                drawCircle(0,0,6.370e6, Shape.POLYGON);
            }
        };
        double p = 2e8;
        t.geometry.setUserArea(-p, p, -p, p);
        t.showPaths = true;
        t.showVelocity = false;
        t.showAcceleration = false;
        t.showLabels = false;
        t.addPointMass("x", "y", "vx", "vy", "ax", "ay");
        g.addTVG(t);
    }

    public void initSimulationParameters(SimulationParameters s) {
        s.fastMotionFactor = 20000;
    }
}

```

### Explanation of the program code

This program code uses the graphics component *MechanicsTVG*. *MechanicsTVG* will be used to display the satellite's orbit on screen. Analogous to exercise sheet 5  $(x, y)$  is the position of the satellite, its velocity is  $(v_x, v_y)$  and its acceleration  $(a_x, a_y)$ . The program code does not yet contain any physical variables. The formulas describing the behavior of the satellite are still missing (method *f*). However, the method call *addPointMass* already specifies that the graphics component *MechanicsTVG* is to access the variables  $x, y, v_x, v_y, a_x$  and  $a_y$  in the physical systems in order to display this point-shaped mass graphically.

By overwriting the *initSimulationParameter* method, one can change parameters that determine how the Physolator performs the simulation. Satellites take hours, days or even weeks to orbit the earth. The assignment

```
s.fastMotionFactor = 20000;
```

causes the simulation to run in fast motion – ten thousand times faster than in reality.

### Exercise 1

Let the initial position of the satellite be  $\begin{pmatrix} 4 \cdot 10^7 \\ 0 \end{pmatrix} m$  its initial speed  $\begin{pmatrix} 0 \\ 4000 \end{pmatrix} \frac{m}{s}$ .

Program this physical system analogous to exercise sheet 1 by adding declarations for the physical variables, specifying the initial values of the physical variables and defining the derivation relationships between the physical variables. Use the variable names  $x, y, v_x, v_y, a_x$ , and  $a_y$ , where  $(x, y)$  stands for

the position of the satellite,  $(v_x, v_y)$  for its speed, and  $(a_x, a_y)$  for the acceleration experienced by the satellite. Program a method  $f$  with the formulas to determine the dependent variable values.

### Exercise 2

Determine how long the satellite will take for one complete orbit.

Use the *ThresholdTrigger* class. Find the explanations on the *ThresholdTrigger* class in the exercise 3 from exercise sheet 6.

### Exercise 3

Launch the satellite from different positions and at different initial speeds. What shapes do the satellite's trajectories have? How can the shape be varied?

### Exercise 4

Mathematically derive the initial position and velocity for a geostationary satellite. Then use this starting position and speed for a physical simulation. Verify the result! Use the simulation result to check whether your calculations of starting position and starting speed are correct!

#### Explanations

Geostationary satellites fly at a speed that is just large enough to circle the earth on a sidereal day. They orbit the earth with the same angular speed at which the earth rotates. For this reason, an observer standing on the earth's surface at the equator has the impression that the satellite is immobile.

A sidereal day is the time it takes the earth to turn around its own axis. The reference point is the fixed starry sky, the universe. A sidereal day lasts 23 hours, 56 minutes, 4.091 seconds, i. e. a total of 86164.091s. The sidereal day is a little shorter than an "ordinary" day with 24 hours. The ordinary day with 24 hours describes a rotation of the earth in relation to the sun.

#### Hints

The angular velocity at which the geostationary satellite flies can be obtained directly from the duration of the sidereal day. To get to the right distance between satellite and earth, the gravitational force must equal centrifugal force. With the angular speed and the distance you get to the trajectory speed, which is also the initial velocity.