Gliwice, 15.12.2024

Mathematical modelling and computer simulations in theory and practice

Documentation of laboratory task no 3

Title: **Trajectory of a Comet**

Author: Aleksandra Stachecka
Field of studies: Informatics sem.V

1 Project Objective

The goal of the project is to visualize the trajectory of a celestial body, such as comet, moving around the Sun based on its eccentricity. The eccentricity determines the shape and type of the orbit.

2 Description

The program consists of a function `trajectory[eccentricity]` which takes one input: *eccentricity*. For value in range from 0 to 1 the function will visualize the trajectory of the comet around the Sun, for the value 1 and above the function will return only the information that the trajectory of the comet cannot be plotted, which I will explain later in this section.

```
  trajectory[eccentricity_] := Module[{v, sol, x, y, t, tt
     },

(* checking eccentricity value to determine trajectory
   type *)
trajectoryType = Which[
  eccentricity == 0, "Circular",
  0 < eccentricity < 1, "Elliptical",
  eccentricity == 1, "Parabolic",
  eccentricity > 1, "Hyperbolic"
  ];

If[eccentricity >= 1,
 Print["For ", trajectoryType,
```

```
    "␣orbits␣(eccentricity␣\[GreaterEqual]␣1),␣numerical␣
        issues␣may␣\
arise␣due␣to␣singularities."];
   Return[
    "This␣trajectory␣cannot␣be␣plotted␣accurately␣with␣the␣
        given␣\
equations."];
   ];
```

2.1 Eccentricity value less than 1

When the eccentricity value is 0, the trajectory will be circular, and when the eccentricity lies between 0 and 1 the trajectory will be elliptical.
To visualize the trajectory of the comet, we first need to solve the velocity differential equation. This equation describes how the velocity of the object changes over time based on the eccentricity of its orbit. It is derived from Kepler's laws of planetary motion and is given as:

$$v'(t) = \frac{(1 + \text{eccentricity})^2}{(1 + \text{eccentricity} \cdot \cos(v(t)))^2} \tag{1}$$

Here $v(t)$ represents the velocity as a function of time, and $v'(t)$ is its time derivative. We will use `NDSolve` to solve this equation numerically.

```
    sol = v /.
   NDSolve[{v'[
        t] == (1 + eccentricity)^2/(1 + eccentricity Cos[v[t
            ]])^2,
      v[0] == 0}, v, {t, -1000, 1000}][[1]];
```

The next step is to compute the trajectory of the comet by determining its position at each point in time. The x and y coordinates are calculated using the solution for the velocity, which we obtained earlier.

```
    x[tt_] := (1 + eccentricity) Cos[
     sol[tt]]/(1 + eccentricity Cos[sol[tt]]);
y[tt_] := (1 + eccentricity) Sin[
     sol[tt]]/(1 + eccentricity Cos[sol[tt]]);
```

2.2 Eccentricity greater than or equal to 1

As mentioned in the beginning, when eccentricity value is greater than or equal to 1, the function will only return an information to the user and will not visualize the trajectory of the comet. That is because of a few reasons:

- For eccentricities $e >= 1$, the denominator in the equations used to calculate the x and y coordinates can approach zero or become negative, causing mathematical singularities, which will make the trajectory undefined or numerically unstable.

- Parabolic ($e = 1$) and hyperbolic ($e > 1$) orbits are unbounded, meaning the object will not return to its starting point or remain within fixed range. This creates difficulties in plotting the trajectory.

So the function detects when $e >= 1$ and provides a warning to the user, without attempting to compute and visualize the comet's path. This helps preventing errors.

2.3 Visualization of the trajectory

After all the calculations we can move to the visualizing the trajectory. The animation is generated using the `Animate` function.
The Sun is represented as a yellow disk at the origin, which remains stationary throughout the animation. The comet's position is plotted as a series of points along its path. We can clearly see how the comet moves over time in response to its eccentricity and the nature of orbit (whether it is circular, elliptical, parabolic or hyperbolic - the last two is only an information to the user).

```
    Animate[Graphics[
    {
     (* the sun *)
     {Yellow, Disk[{0, 0}, 0.4]},
     PointSize[0.005],
     Point[Table[{x[tt], y[tt]}, {tt, -t, t, 1}]] (*
     trajectory points *)
     },
    Axes -> True, AxesLabel -> {"x", "y"},
    PlotRange -> {{-3, 3}, {-3, 3}},
    ImageSize -> {400, 350}
    ],
   {t, 1, 1000}, AnimationRate -> 10]];
```

These will be our results for eccentricities like 0, 0.3, 0.6, 0.9, 1, 1.2:

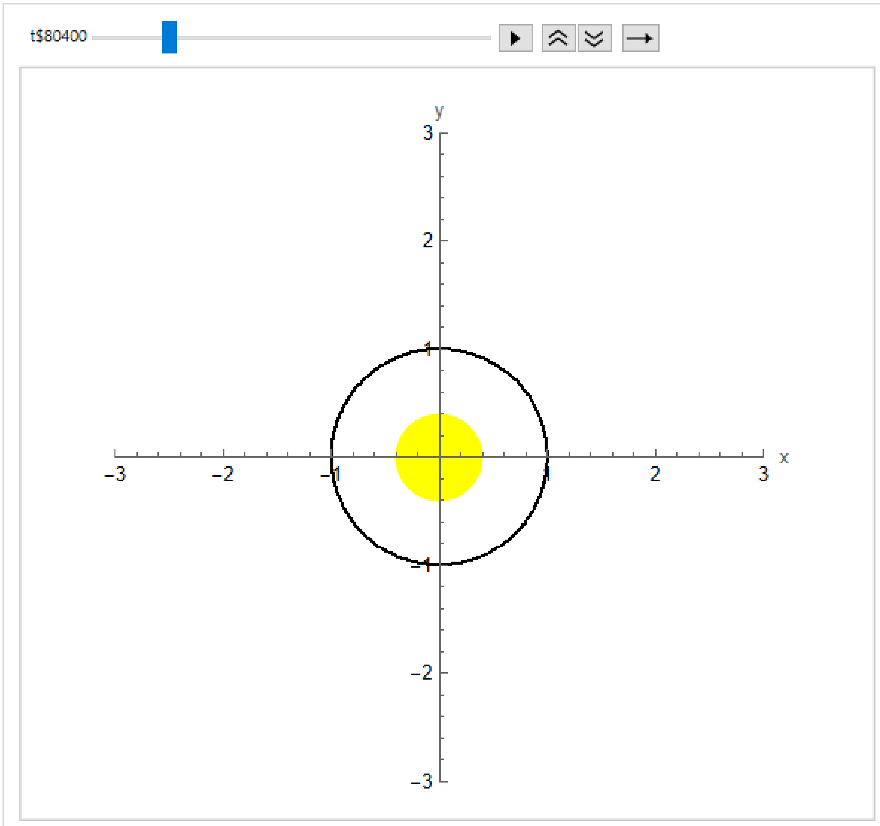

Figure 1: e = 0

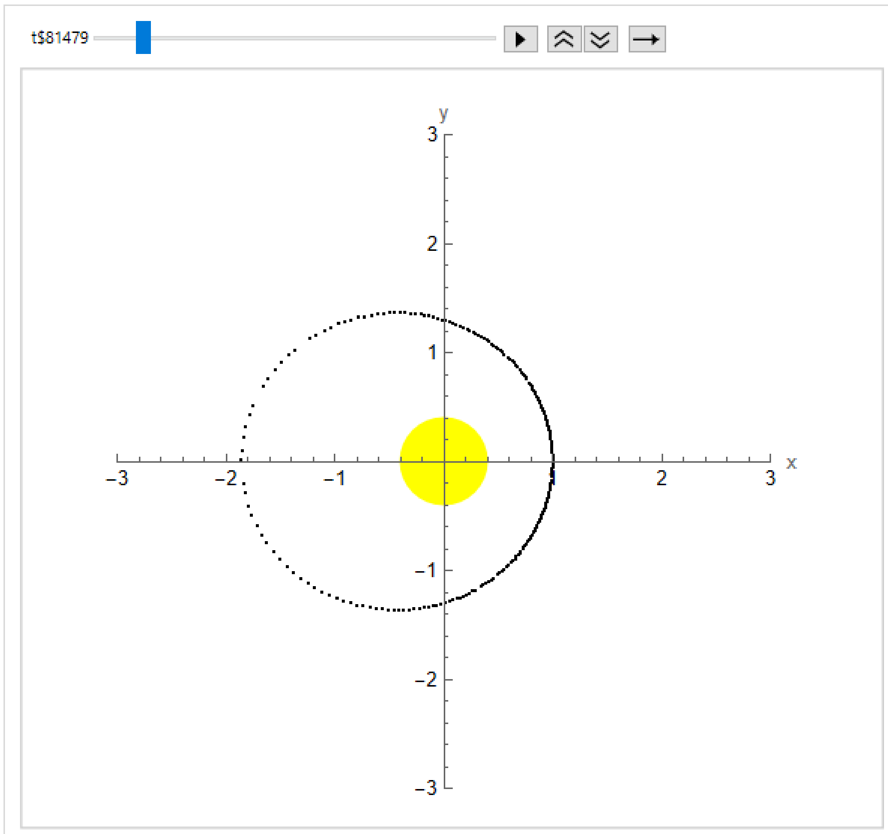

Figure 2: e = 0.3

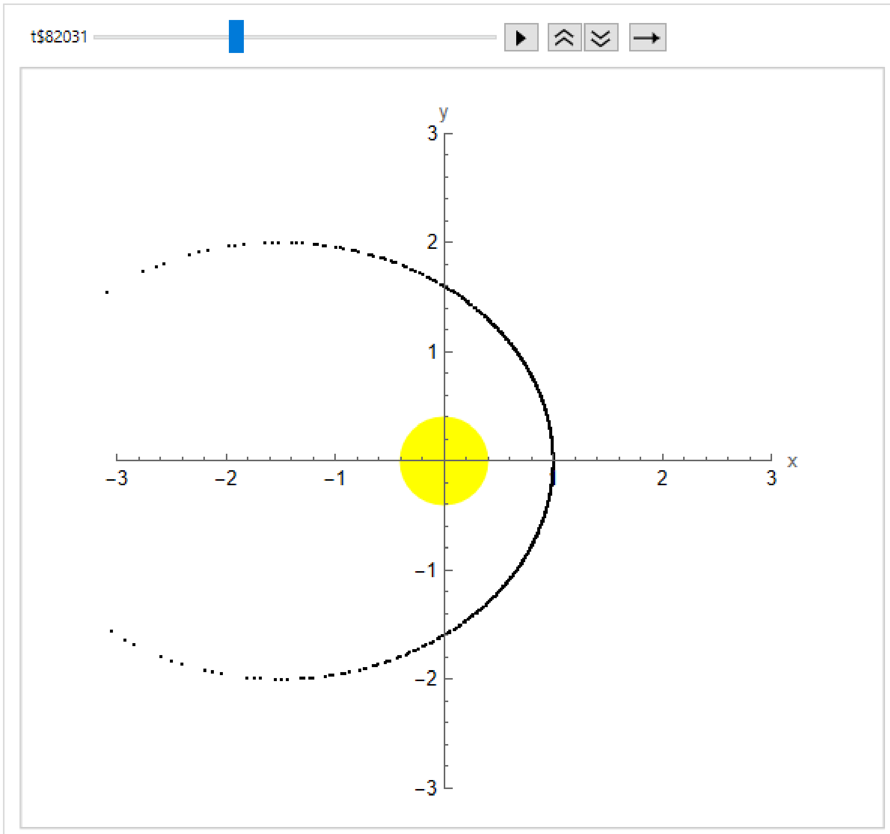

Figure 3: e = 0.6

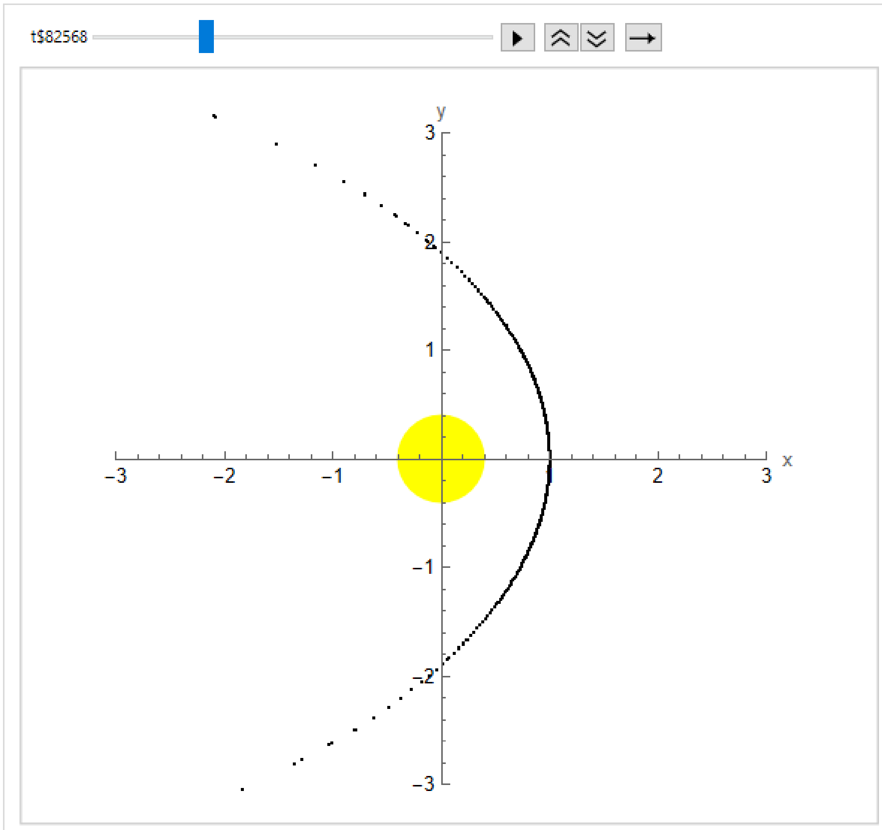

Figure 4: e = 0.9

```
trajectory[1]
For Parabolic
  orbits (eccentricity ≥ 1), numerical issues may arise due to singularities.
This trajectory cannot be plotted accurately with the given equations.

trajectory[1.2]
For Hyperbolic
  orbits (eccentricity ≥ 1), numerical issues may arise due to singularities.
This trajectory cannot be plotted accurately with the given equations.
```

Figure 5: e ¿= 1 - information for user

3 Enclosures

"STACHECKA Aleksandra Project 5.nb"