Cursor Tutorial: Simulating a Spacecraft Orbit

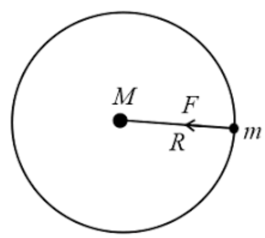
Installation Guide

- Account required - free trial is limited to 2 weeks :C
- Lots of alternative tools:
  - trae.ai
  - trypear.ai

```
https://github.com/jamesmurdza/awesome-ai-devtools
```

https://huggingface.co/datasets/fka/awesome-chatgpt-prompts/viewer/default/train?views[]=train&row=72

Physics setup:



We are going to code the orbit of the Kepler spacecraft around the Sun!

Tip: Imagine the Sun at (0,0) and Kepler moving in 2D (x, y). The force always pulls Kepler toward the Sun.

To simulate Kepler's orbit, we calculate the gravitational force and update the spacecraft's motion step-by-step:

1. F = G M m / ||**position**|| ^2
2. Direction of the force: - **position** / ||**position**||
3. F = m **a** → calculate acceleration **a** (2d array in direction of force)
4. **v** = integral (**a**) , **position** = integral (**v**)
5. We are going to update velocity and position iteratively:

   v = dt * **a**

   p = dt * **v**

Objective

Use Cursor's tab-completion and AI tools to:

- Set up basic orbital parameters and initial values for Kepler
- Simulate gravitational acceleration and orbital motion using Euler integration
- Visualize the orbit with `matplotlib`

We are going to do the same task in two different ways:

1. Coding with minimal help (tab completion)
2. Plan with cursor, refine plan, then ask cursor to execute the plan.

General Cursor steps:

Cheatsheet:

AI panel ctrl+alt+b

Change a chunk of code: ctrl+k

Accept changes ctrl+enter

Set rules for cursor inside settings: https://cursor.directory/rules

Change the choice of AI.

Cursor can see your a directory and can understand how software is structured - it can be useful for understanding a codebase and you can ask the AI chat Q i.e. 'what part of the code does x'.

Version 1: Coding with minimal AI

1. Type out the physics steps described above with tab completion. Use comments in your code to guide it.
2. Ask the AI chat to fill in some of the orbital parameters for the Kepler spacecraft.
3. Ask the AI to get the initial values of velocity and position based on eccentricity (eccentricity = 0 is a circular orbit).

Initialize Parameters

Cursor can auto-suggest known constants:

```
import numpy as np
import matplotlib.pyplot as plt

G = 6.67430e-11  # gravitational constant
M = 1.989e30     # mass of the Sun
m = 1052         # mass of Kepler spacecraft
dt = 3600        # time step (1 hour)
```

3. Initial Conditions

Define initial position & velocity (based on circular or elliptical orbit):

```
positions = np.array([1.5e11, 0])  # initial position (e.g., 1 AU)
v = np.array([0, 30000])   # tangential velocity (adjust for eccentricity)
```

4. Main Loop with Euler Integration

```
positions_history = np.zeros((1000,2))
for i in range(10000):
    a = -G * M * positions / np.linalg.norm(positions)**3
    v += a * dt
    positions += v * dt
      positions_history[i] = positions
```

5. Plot the Orbit

```
plt.plot(positions_history[:,0], positions_history[:,1])
plt.axis("equal")
plt.show()
```

Version 2: Cursor as your AI assistant

Set the tone in the AI chat (or rules)

```
I prefer code in golf style, no comments, very short code and we are going to do some planning first before you code.
```

Ask the AI to come up with a plan

```
Simulation of orbit of a spacecraft around a larger mass, with sun parameters for larger mass, kepler spacecraft orbit for smaller mas
s. Start with the gravity equation. Simplest integration with dt. Plan first no code.
```

Ask the AI to initialize the velocity and position based on eccentricity

```
add in an eccentricity parameter, set velocity and position based on this
```

Visualize

```
visualize keplers law, equal areas swept in equal times, with a sliding bar (time)
```
