

利用 Python 进行天体运行的模拟

崔士强 PB22151743

2023 年 6 月 26 日

1 原理

本程序通过设置步长再逐步计算的方式进行模拟。设步长为 dt ，两个天体的位置坐标分别为 $\mathbf{p}_i, \mathbf{p}_j$ 每一步需要进行的计算为：

$$\mathbf{F}_{ij,t} = \frac{Gm_i m_j}{d^2} \mathbf{e}_{ij}$$

$$\mathbf{a}_{ij,t} = \frac{\mathbf{F}_{ij,t}}{m_i}$$

$$\mathbf{v}_{i,t+1} = \mathbf{v}_{i,t} + \mathbf{a}_{i,t} dt$$

可以计算两个天体经过一小段时间的位置变化（近似值）。

程序还涉及添加环绕天体。结合向心力公式

$$F = \frac{mv^2}{R}$$

可以得到

$$v = \sqrt{\frac{GM}{R}}$$

其中 M 为中心天体质量。添加环绕天体时同时赋予该速度。

2 设计方案

定义一个类 `CelestialBody` 来表示天体。属性包括名称 `name`，质量 `mass`，位置 `position`，速度 `velocity`，两个方法 `update_velocity` 和 `update_position` 分别计算速度和位置的变化。

对于每一个输入的质量及长度，提供几个可选单位，程序中用一个字典存储这些单位，输入后由函数 `convert_mass_to_kg` 以及 `convert_distance_to_m` 换算成国际单位制。

如果添加环绕天体，程序会将其放置在中心天体 x 负方向，距离（即环绕半径）由用户输入。程序根据中心天体质量以及环绕半径计算速度。

如果发射一个天体，用户需要输入发射起点，方向以及速度。程序会计算三个方向上的分速度。

添加天体完成后可以选择进行模拟。需要输入步长以及步数。程序逐步计算，每一步的结果储存在二维列表 `positions` 中。每一步计算完成后检查是否出现相撞。判定标准为两个天体

小于用户输入的阈值即为相撞，根据动量守恒计算相撞后运动情况，这里假设撞后两个天体合二为一。

模拟完成后可以绘图表示结果。程序根据列表 `positions` 的信息绘制散点图，图中的点即为每一步各个天体的位置

3 创新性描述

1. 程序为质量和长度提供了多个单位，方便输入
2. 通过局部近似成匀速直线运动进行模拟，避免过于复杂的计算

4 运行示例

两个质量均为 1 倍太阳质量的天体 a,b. a 静止于原点, b 于 $(0.5AU, 0.5AU, 0.5AU)$ 的位置, 以 $20km/s^2$ 的速度向 z 轴负方向射出, 模拟步长为 5000s, 模拟 1000 步, 结果如下图所示:

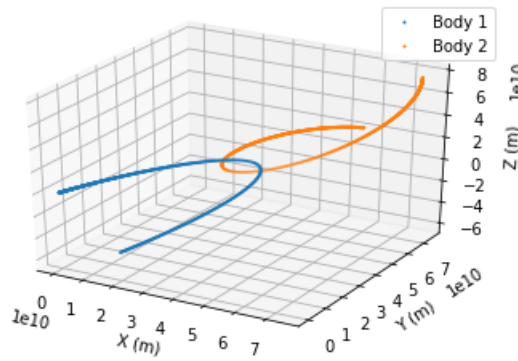


图 1: 示例结果 1

如果将步数增加至 4000 步, 结果如下图所示:

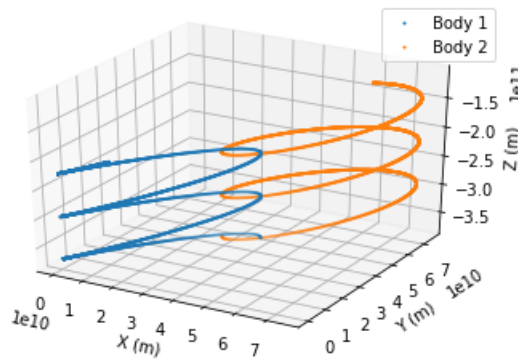


图 2: 示例结果 2

5 学习心得

本课程是我接触的第二门计算机类课程，使我了解了又一种计算机语言 Python。相比于 C 语言，Python 丰富的科学计算库，更高的开发效率使得其在多数场景下更适合科学计算。通过本课程，我获取了辅助日常学习以及今后科研工作的一项重要技能，也接触到了面向对象编程的思想。

参考文献

- [1] 罗奇鸣. Python 科学计算基础.
- [2] Ian Morison. Introduction to Astronomy and Cosmology. West Sussex: John Wiley & Sons Ltd, 2008.