Material Modeling Homework 2 Report

Gengyao Qiu

September 26, 2025

1 Introduction

In this project, I'm focusing on writing Python code to solve two classic physical questions by using numerical method called Euler Method. During the project, I met some problems both on coding and mathematical concepts. In coming details, I provide more information about the problems I met and the methods I solved them. Also the contribution in group are also demonstrated.

2 The problems I met and how I solved

2.1 Calculate the τ

In the Carbon dating section, I have been given the half-life of carbon and need to calculate the decay constant τ.

We know the analytical solution of decay function is:

$$N(t) = N_0 \mathrm{e}^{\frac{-t}{\tau}} \tag{1}$$

Assume the half-life is $T_{1/2}$, then according to definitions:

$$\begin{gathered} N(t) = 2N(t + T_{1/2}) \\ 2 = \frac{N(t)}{N(t + T_{1/2})} = \frac{\mathrm{e}^{-t/\tau}}{\mathrm{e}^{(-t-T_{1/2})/\tau}} = \mathrm{e}^{\frac{-t}{\tau} + \frac{t+T_{1/2}}{\tau}} \end{gathered} \tag{2}$$

Then we got the relation between τ and $T_{1/2}$ is

$$T_{1/2} = \tau \ln 2 \tag{3}$$

```
The deviation in width=10 is -0.08432282690637188%
The deviation in width=100 is -0.8461957745584046%
The deviation in width=1000 is -8.59000771328155%
```

Figure 1: Percents of deviation in different width.

2.2 Deviation Analysis

We know in Euler Method, the precision of answer is related to the step width. Bigger step width can make computing faster but will loss precision, too small step width will also lead to lower computational efficiency. Hence, I should make out what step width is better. In the project I need to calculate the percent of deviation between numerical answer and analytical answer, and also need to figure out if the deviation is as large as I could expect from the neglected second-order term.

By coding, I know the percents of deviation in different width, which can be found in Figure 1. Based on it, we could find the deviation up to around 8.6% in width=1000. Usually, this deviation is not acceptable. After this, I can test the scale of this answer by neglected second-order term. Consider the Taylor expansion as follow:

$$f(x + \Delta x) = f(x) + f'(x)\Delta x + \frac{1}{2}f''(x)\Delta t^2 + ... \tag{4}$$

In Euler Method we directly ignore all the terms higher than the first term, which cause the deviation. So the deviation in one step is:

$$E_{step} \sim \frac{1}{2\tau^2}N(t)\Delta t^2 \tag{5}$$

The deviation for all steps is:

$$E \sim \frac{t}{\Delta t}E_{step} = \frac{tN(t)\Delta t}{2\tau^2} \tag{6}$$

Then the deviation is:

$$\frac{E}{N(t)} \sim \frac{t\Delta t}{2\tau^2} \tag{7}$$

After this analysis, we know the deviation is around 8%, and this is similar to program's result.

3 Results and discussion

3.1 Carbon dating

In carbon section we can get the decay rate curves by run `make carbon` or `make carbon WIDTH=10,100,1000` where `WIDTH` is the time steps you want to show on the figure, an example has been shown in Figure 2.

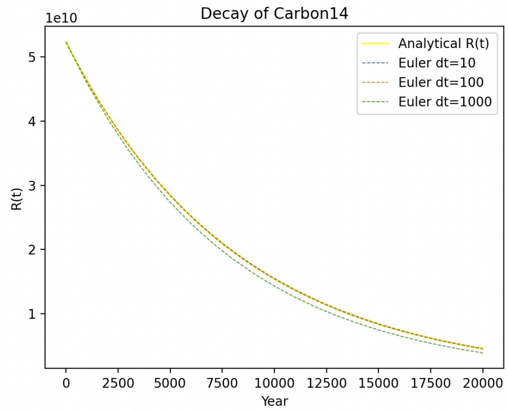
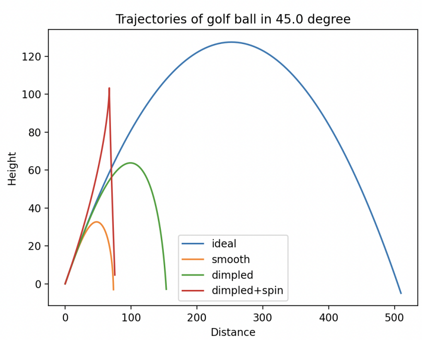


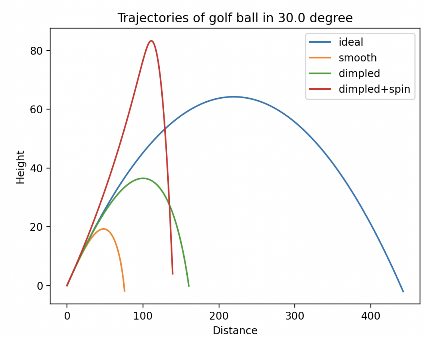
Figure 2: Carbon dating example

3.2 Golf

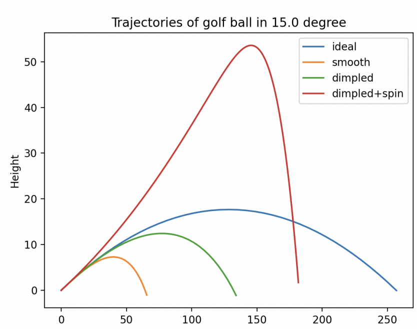
In golf section we can draw the trajectories of a golf ball in different conditions by run command line `make golf` or `make golf THETA=9` where `THETA` is the angle for initial velocity, the results in $45°$,$30°$,$15°$ and $9°$ have been shown in Figure 3.



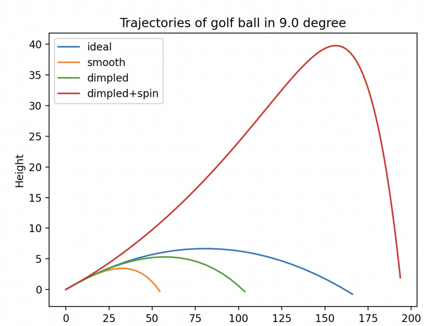
(a) Trajectories in 45 degree



(b) Trajectories in 30 degree



(c) Trajectories in 15 degree



(d) Trajectories in 9 degree

Figure 3: The trajectories of golf balls in different conditions.

Based on Figure 3 we find that while the θ is smaller, the ball with dimpled and spin will have a better performance. And while we make the ball dimpled which will go further than smooth one.

4 Contributions

In this project, I engaged in group discussion and help my group members to debug the Python code. During the project, I discussed the possibilities that may cause the difference between our output. But we finished our homework include coding and formula derivation independently.