# Homework No.7

Ruqi Feng

November 2022

# 1 Simple Pendulum

## 1.1 Description

Write a program to numerically solve a simple pendulum’s motion using Euler’s midpoint method, RK4, Euler-trapezoidal method (implement these methods by yourself). Plot the angle and total energy as a function of time. Explain the results.

## 1.2 Solution

考虑一个 $n$ 阶 ODE

$$c_n f^{(n)}(x) + c_{n-1} f^{(n-1)}(x) + c_{n-2} f^{(n-2)}(x) + c_1 f'(x) + c_0 f(x) = 0 \tag{1}$$

可以通过换元 $f^{(k)}(x) \to y_k (k \leq n_{max} - 1)$ 被转化为一阶 ODE 组：

$$\frac{dy_n}{dx} = -\frac{c_{n-1}y_{n-1} + c_{n-2}y_{n-2} + c_1 y_1 + c_0 y_0}{c_n} \tag{2}$$

$$\frac{dy_{n-1}}{dx} = y_n \tag{3}$$

$$... \tag{4}$$

$$\frac{dy_k}{dx} = y_{k+1} \tag{5}$$

$$... \tag{6}$$

$$\frac{dy_0}{dx} = y_1 \tag{7}$$

写成向量形式，就是

$$\frac{d\vec{y}}{dx} = \vec{f}(\vec{y}, x) \tag{8}$$

对于单摆，$\ddot{\theta} = -k\sin\theta$, 其中 $k = \frac{g}{l}$ Eq.8中的 $\vec{f}$ 不显含时，于是有

$$\frac{dy_0}{dx} = y_1 \tag{9}$$

$$\frac{dy_1}{dx} = -k\sin\theta \tag{10}$$

其中 $y_0 = \theta$, $y_1 = \dot{\theta}$ 下面我们使用四种方法解这个 ODE：

### 1.2.1 Euler

把 $\vec{y}$ 关于 $x$ 展开到第 1 阶

$$\vec{y}(x + \delta x) = \vec{y}(x) + \frac{d\vec{y}}{dx}\delta x + O(\delta x) \tag{11}$$

将 Eq.8 and Eq.9代入得

$$\vec{y}_{i+1} = \vec{y}_i + \vec{f}(\vec{y}_i, x_i)\delta x + O(\delta x) \tag{12}$$

其中 $\delta x$ 为步长。只要输入初始的 $\vec{y}_0, x_0$ 和 $\vec{f}(\vec{y}_0, x_0)$，迭代即可求解。

### 1.2.2 Midpoint

为了得到更准确的导数，用 Euler 法得到的 $y_{i+1}$ 估计中点，再利用中点的导数 $\vec{f}(\vec{y} + \Delta y_{attempt}/2, x + h/2)$ 进行迭代。其中

$$\Delta y_{attempt} = \vec{f}(\vec{y}_i, x_i)\delta x \tag{13}$$

### 1.2.3 RK-4

假设 $\vec{y}_{i+1}$ 的变化量是不同位置 $(x_j, y_j)$ 的导数的线性组合

$$\vec{y}(x_{i+1}) = \vec{y}(x_i) + \sum_{j=1}^{n} a_{ij}\vec{f}_{ij}\delta x \tag{14}$$

其中

$$\vec{f}_{ij} = \vec{f}(y_i + \sum_{k=1}^{j} b_{jk}\vec{f}_{ik}\delta x, x_i + c_{ij}\delta x) \tag{15}$$

于是对于某一个求解步 $i$ 我们获得了 $a_{ij}$, $b_{jk}$, $c_{ij}$ 额外的自由度。于是我们可以调整这些参数让更高阶的 Taylor 公式被满足。4 阶 RK 法中，有 4 个 $\vec{f}$ 被求解出，在常数时间复杂度内让局部精度达到了 $o(h^4)$。对应的参数为

$$k_1 = \vec{f}(x_i, y_i) \tag{16}$$

$$k_2 = \vec{f}(x_i + \frac{1}{2}\delta x, y_i + \frac{1}{2}k_1\delta x) \tag{17}$$

$$k_3 = \vec{f}(x_i + \frac{1}{2}\delta x, y_i + \frac{1}{2}k_2\delta x) \tag{18}$$

$$k_4 = \vec{f}(x_i + \delta x, y_i + k_3\delta x) \tag{19}$$

#### 1.2.4 Euler-trapazoidal

为了在函数变化较平缓的地方减少计算开销，我们希望在这些地方增加步长，影响精度较小。E-T 方法就是一个变步长的方法。它的基本思想是在某一个求解步的位置上迭代计算直到 $\vec{f}$ 收敛到希望的精度。

首先用 Euler 法得到初始迭代

$$\vec{y}_{i+1,0} = \vec{y}_i + \vec{f}_{i,0}\delta x \tag{20}$$

得到 $\vec{f}_{i+1,0} = \vec{f}(x_)$ 迭代计算导数

$$\vec{y}_{i+1,j} = \vec{y}_i + (\vec{f}_{i,0} + \vec{f}_{i+1,j})\frac{\delta x}{2} \tag{21}$$

其中

$$\vec{f}_{i+1,j+1} = \vec{f}(x + \delta x, \vec{y}_{i+1,j}) \tag{22}$$

### 1.3 Input and Outputs

用四种方法求解并画出 $\theta$ 和 $\omega$ 对时间的图像。步长均为 $1e-3$s(ET 法收敛判断条件为 $1e-5$)，求解 $10 \to 20$s 内的运动。可以看出，Euler 法的角度振幅逐渐变大，而其他方法没有明显的变化。单摆的方程是能量守恒的，所以为了研究各个数值方法的精度，将能量对时间画出。除了 Euler 法中，能量非周期性的偏移主导外，其他各个方法中能量周期性震荡，而非周期性的偏移与之相比很小。这代表着虽然这些方法并不保证能量守恒，但能量的总体偏移也较小。其中 RK-4 在周期中能量波动的相对误差很小，达到了 $1e-11$。z

程序使用了 numpy, matplotlib 库。在安装了依赖库的环境中运行 q1_pendulum.py 即可。下面是程序运行的截图。

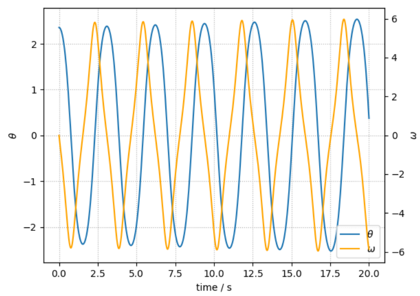


图 1: Euler 法

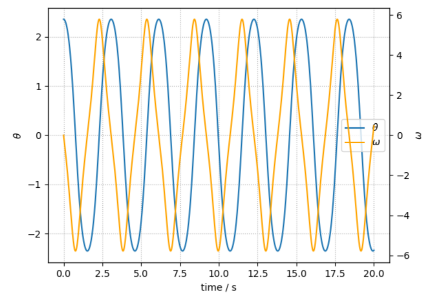


图 2: midpoint 法

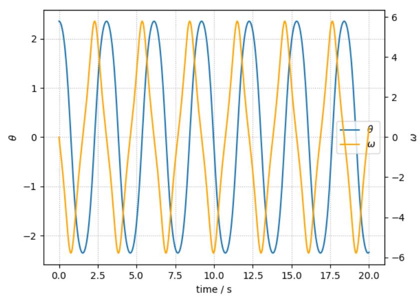


图 3: Runge-Kutta - 4 order 法

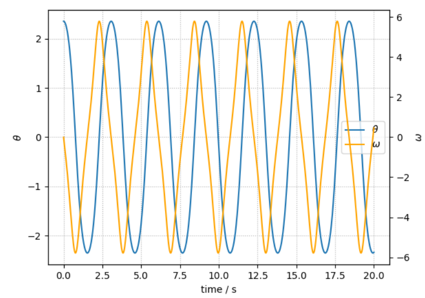


图 4: Euler-Trapezoidal 法

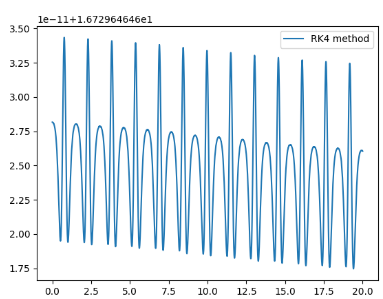


图 5: RK-4 法，能量变化

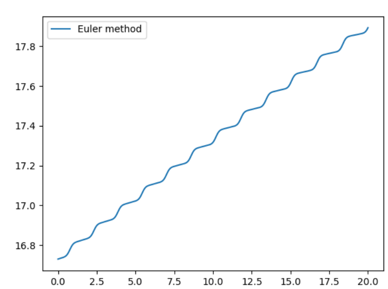


图 6: Euler 法, 能量变化

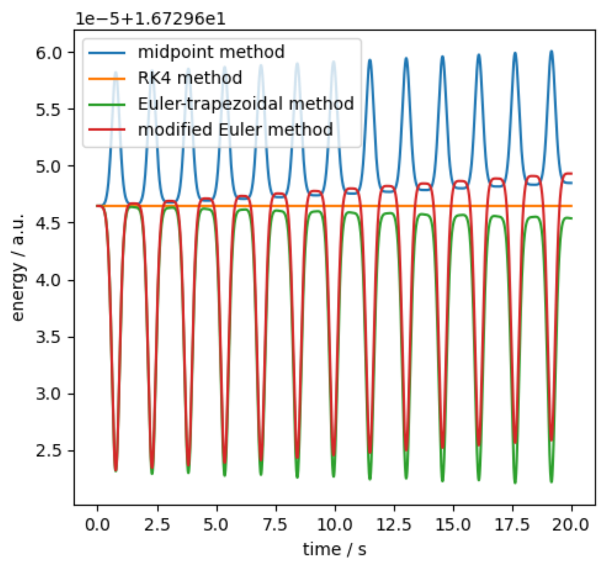


图 7: midpoint method, Euler-Trapezoidal method, modified Euler method 的能量变化。可以看出 RK-4 法比其他更精确。

### 1.4 Pseudocode

**Algorithm 1** Simple Pendulum: Euler

**Input:** $k$, or $\frac{g}{l}$, $\theta_0$, $\dot{\theta}_0$ $h$, $t_m$

**Output:** $E(t)$, $\theta(t)$

$t \leftarrow 0$
$\omega[0] \leftarrow \dot{\theta}_0$
$\theta[0] \leftarrow \theta_0$
**while** $t < t_m$ **do**
    $\theta[i+1] \leftarrow \theta[i] + \omega h$
    $\omega[i+1] \leftarrow \omega[i] - k \sin \theta[i] h$
    $t \leftarrow t + h$
**end while**
$\vec{E} \leftarrow \frac{1}{2} l^2 \omega^2 + g^2 l^2 (1 - \cos \theta)$ ▷ set m as 1
**return** $\vec{E}, \vec{\theta}$

**Algorithm 2** Simple Pendulum: midpoint

**Input:** $k$, or $\frac{g}{l}$, $\theta_0$, $\dot{\theta}_0$ $h$, $t_m$

**Output:** $E(t)$, $\theta(t)$

$t \leftarrow 0$
$\omega[0] \leftarrow \dot{\theta}_0$
$\theta[0] \leftarrow \theta_0$
**while** $t < t_m$ **do**
    $\theta_{mid} \leftarrow \theta[i] + \omega h$
    $\omega_{mid} \leftarrow \omega[i] - k \sin \theta[i] h$
    $\theta[i+1] \leftarrow \theta[i] + \frac{\omega[i] + \omega_{mid}}{2} h$
    $\omega[i+1] \leftarrow \omega[i] - k \sin(\frac{\theta[i] + \theta_{mid}}{2}) h$
    $t \leftarrow t + h$
    $i \leftarrow i + 1$
**end while**
$\vec{E} \leftarrow \frac{1}{2} l^2 \omega^2 + g^2 l^2 (1 - \cos \theta)$ ▷ set m as 1
**return** $\vec{E}, \vec{\theta}$

**Algorithm 3** Simple Pendulum: E-T

**Input:** $k$, or $\frac{g}{l}$, $\theta_0$, $\dot{\theta}_0$ $h$, $t_m$

**Output:** $E(t)$, $\theta(t)$

$t \leftarrow 0$
$\omega[0] \leftarrow \dot{\theta}_0$
$\theta[0] \leftarrow \theta_0$
**while** $t < t_m$ **do**
  $\theta_{last} \leftarrow \theta[i] + \omega h + 1$
  $\omega_{last} \leftarrow \omega[i] - k \sin \theta[i] h + 1$
  $\theta_{tmp} \leftarrow \theta[i] + \omega h$
  $\omega_{tmp} \leftarrow \omega[i] - k \sin \theta[i] h$
  **while** $(\theta_{tmp} - \theta_{last})^2 + (\omega_j - \omega_{last})^2$<1e-5 **do**
    $\theta_{last} \leftarrow \theta_{tmp}$
    $\omega_{last} \leftarrow \omega_{tmp}$
    $\theta_{tmp} \leftarrow \theta[i] + \frac{\omega[i]+\omega_{last}}{2} h$
    $\omega_{tmp} \leftarrow \omega[i] - k(\frac{\sin \theta[i]}{2} + \frac{\sin \theta_{last}}{2}) h$
  **end while**
  $\theta[i+1] \leftarrow \theta_{tmp}$
  $\omega[i+1] \leftarrow \omega_{tmp}$
  $t \leftarrow t + h$
  $i \leftarrow i + 1$
**end while**
$\vec{E} \leftarrow \frac{1}{2} l^2 \omega^2 + g^2 l^2 (1 - \cos \theta)$ ▷ set m as 1
**return** $\vec{E}, \vec{\theta}$

**Algorithm 4** Simple Pendulum: RK-4

**Input:** $k$, or $\frac{g}{l}$, $\theta_0$, $\dot{\theta}_0$ $h$, $t_m$

**Output:** $E(t)$, $\theta(t)$

$t \leftarrow 0$
$\omega[0] \leftarrow \dot{\theta}_0$
$\theta[0] \leftarrow \theta_0$
**while** $t < t_m$ **do**
    $\theta_{mid} \leftarrow \theta[i] + \omega h$
    $\omega_{mid} \leftarrow \omega[i] - k \sin \theta[i] h$
    $\theta[i+1] \leftarrow \theta[i] + \frac{\omega[i]+\omega_{mid}}{2} h$
    $\omega[i+1] \leftarrow \omega[i] - k \sin(\frac{\theta[i]+\theta_{mid}}{2}) h$
    $t \leftarrow t + h$
    $i \leftarrow i + 1$
**end while**
$\vec{E} \leftarrow \frac{1}{2} l^2 \omega^2 + g^2 l^2 (1 - \cos\theta)$ ▷ set m as 1
**return** $\vec{E}, \vec{\theta}$

# 2 Radial Schrodinger Equation

## 2.1 Description

Write a program to numerically solve radial Schrödinger equation (see detais in the slides) for

- $V(r) = -1/r$ (hydrogen atom)
- $V_{loc}(r)$

Compute and plot the first three eigenstates.

## 2.2 Solution

我们知道径向 Schrodinger 方程可以写成

$$(\frac{\hbar^2}{2m}\frac{d^2}{dr^2} + E - V(r) - \frac{\hbar^2 l(l+1)}{2mr^2})u = 0 \tag{23}$$

其中

$$u(r) = r\psi(r) \tag{24}$$

是归一化的径向波函数。因此和 Sec.1一致，我们可以将此方程转化为 2 元的 1 阶 ODE 组。但这里 $E$ 未知，是一个本征值方程。

注意到边界条件

$$u(0) = 0 \tag{25}$$

$$u(\infty) = 0 \tag{26}$$

所以在不考虑归一化时，边界导数可以任取，相当于 $u \to \lambda u$ 一定仍然满足 Eq.23。于是本征值问题可以转化为解方程

$$u_\infty(E) = 0 \tag{27}$$

其中 $u_\infty$ 在给定 $E$ 时可以由初始条件 $u(0)$ 解得。等价地，也可以从 $r = \infty$ 作为解 ODE 的初始条件，寻找使得 $u(0) = 0$ 的 $E$，这被称为 shooting 法。我采用了 bisection 方法解 Eq.27，并用 RK-4 解 ODE 组。为了确定各个解的初始区间，我先在在不同 l 下足够大的区域内画出 $u(\infty)$ 关于 $E$ 的图像，从而找到 3 个最低的本征能量的大致范围。

## 2.3 Input and Outputs

取 $\frac{\hbar^2}{m} = 1$ 以及 $\frac{e^2}{4\pi\varepsilon_0} = 1$。

### 2.3.1 hydrogen atom

可以注意到, 取 $r_\infty$ 为约 100 以上时，$u_\infty$ 零点对应的能量已经基本收敛。所以取 $r_\infty = 100$。由 Fig.8,9, 10可以看出，在势能 $V_1(r) = \frac{1}{r}$ 时，$l$ 增大 1 会使得最小的本征能量被去除；而某个 $l$ 的所有本征能量都包含在 $l-1$ 的本征能量中。这对应着某个 $n$ 能级的能量是 $n(n+1)$ 度简并的。

最小 3 个能量的本征态可以是 $l = 0$ 的最小三个能量。分别为约 $0.5, 0.125, 0.5555...$。用 bisection 法可以解得能量为

| $n$ | $E/\alpha^2 m_e c^2$ |
|---|---|
| 1 | -0.4999992370605469 |
| 2 | -0.12499923706054687 |
| 3 | -0.05555603027343749 |

值得注意的是，从 $u_\infty$ 开始求解和 $u_0$ 开始求解给出了完全一**致**的结果。这说明从两个方向求解 ODE 的过程中差别不大。但在画出本征态的过程中，可以看出11中的虚线，即从 0 开始求解时，$u_\infty$ 并不能满足，而从 0 开始求解却可以。我认为，这是由于

$$\|\frac{\partial u_\infty}{\partial E}\| \gg \|\frac{\partial u_0}{\partial E}\| \tag{28}$$

即使求解 ODE 的过程中没有误差，$u_\infty$ 也会因为 $E$ 受限于浮点数精度不是精确值而出现大幅度偏离。类似地，$u_\infty$ 也会受到 RK-4 求解误差的影响，这也可能是 $u_\infty$ 发散而 $u_0$ 不发散的原因。

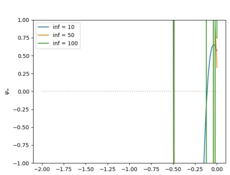


图 8: $u_\infty(E)$ 在 $l=0$

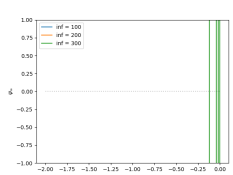


图 9: $u_\infty(E)$ 在 $l=1$

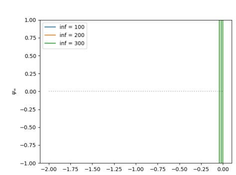


图 10: $u_\infty(E)$ 在 $l=2$

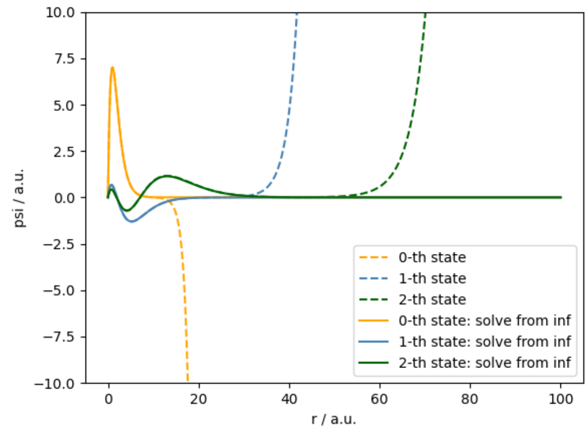


图 11: 从两个方向代入 E 求解 $u(r)$，从 0 开始求解时，在一段时间后 $u_\infty$ 会发散

### 2.3.2 pseudo potential

取势能为2.1并再次画出 $u_\infty(E)$ 的图像，如 Fig.12, 13, 14所示。可以看出最小的能量约为 $-4$ 在 $l=0$，第二小约 $-1.1$ 在 $l=1$, 第三小在 $-1.1$ 约 $l=0$. 而且可以明显看出不同 $l$ 的简并解除了。可以求解出三个最小能量是

| $n$ | $E/\alpha^2 m_e c^2$ |
|---|---|
| 1 | -4.458247184753418 |
| 2 | -1.122279167175293 |
| 3 | -1.1154321670532226 |

画出对应的本征态如图15所示，可以发现 $l=0$ 的最小、第二小能量本征态分别有 0,1 个零点; $l=1$ 的最小能量本征态有 0 个零点。这再次确认了我们已经找到了最小的能量。从 $u_\infty$ 开始求解的发散原因与 Sec.2.3.1中的讨论相同。

### 2.3.3 screenshot

程序使用了 numpy 和 matplotlib 和 scipy 库。在安装了依赖库的环境下运行 q2_radial_Schrodinger.py 即可。

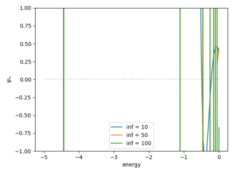

图 12:

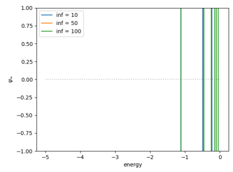

图 13:

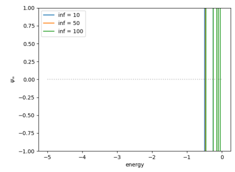

图 14:

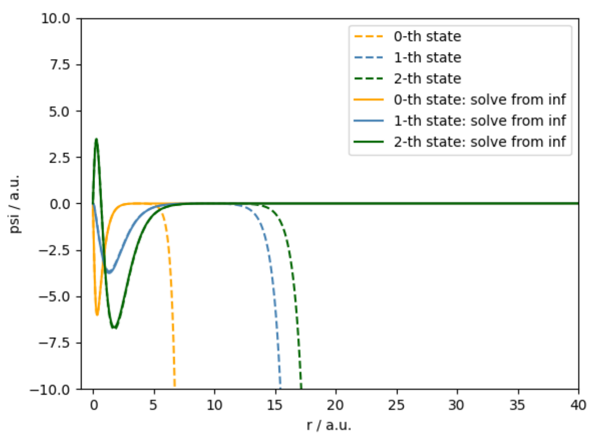


图 15: 代入 $E$ 求解 $u$。节点数分别是 $0, 0, 1$, 对应 $l = 0$ 的第 1, 2 小能量和 $l = 1$ 的第一小能量。

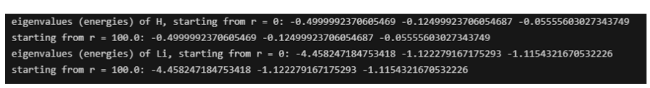


图 16: 程序运行截图

## 2.4 Pseudocode

**Algorithm 5** Radial Schrodinger

**Input**: potential $V(r)$, energy range $el, er$ where $u_\infty(el) < 0$ and $u_\infty(er) > 0$

**Output**: energy eigenvalue $E$

$u_0 \leftarrow 0$

$u_0' \leftarrow 1$ ▷ arbitrary derivative, I chose 1

$tol \leftarrow 1e - 5$

**while** abs$(el - er) > tol$ **do**

  Solve $u_\infty$ with energy $E = \frac{el+er}{2}$ using RK-4 method ▷ see Algo.4

  **if** $u_infty < 0$ **then**

    $el \leftarrow \frac{el+er}{2}$

  **else**

    $er \leftarrow \frac{el+er}{2}$

  **end if**

**end while**

$E \leftarrow \frac{el+er}{2}$

**Return**: $E$