CompPhys Assignment 07

李尚坤　物理学系　20307130215

1 Simple Pendulum

1.1 题目描述

Write a code to numerically solves the motion of a simple pendulum using Euler's method, 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 解决方案描述

考虑一个质量为 m，摆长为 l 的单摆在重力场中的运动，设重力加速度为 g，单摆与竖直方向夹角为 θ，其动力学方程为：

$$\frac{d^2\theta}{dt^2} + \frac{g}{l}\sin\theta = 0 \tag{1.1}$$

令 $y_1 = \theta, y_2 = \frac{d\theta}{dt}$，则可将上述二阶方程转化为一阶线性微分方程组：

$$\begin{cases} \frac{dy_1}{dt} = y_2 \\ \frac{dy_2}{d_t} = -\frac{g}{l}\sin y_1 \end{cases} \tag{1.2}$$

最后再利用公式 $E = \frac{1}{2}ml^2\frac{d\theta}{dt}^2 + mgl(1-\cos\theta)$ 计算出能量随时间的变化关系。接下来我们分别利用四种方法对该问题进行求解。

1.2.1 Euler's Method

取每一小区间长度为 h，用区间起始点的斜率和线性估计估算区间末端函数值，不断迭代即可得最终函数值：

$$\begin{cases} y_1(t_{i+1}) = y_1(t_i) + h\frac{dy_1(t_i)}{dt} = y_1(t_i) + hy_2(t_i) \\ y_2(t_{i+1}) = y_2(t_i) + h\frac{dy_2(t_i)}{dt} = y_2(t_i) - h\frac{g}{l}\sin y_1(t_i) \end{cases} \tag{1.3}$$

1.2.2 Midpoint Method

取每一小区间长度为 h，先用区间起始点的斜率和线性估计估算区间末端函数值，再由初末端函数值取平均估算出中点的函数值

$$\begin{cases} y_1(t_{i+\frac{1}{2}}) = y_1(t_i) + \frac{h}{2}\frac{dy_1(t_i)}{dt} = y_1(t_i) + \frac{h}{2}y_2(t_i) \\ y_2(t_{i+\frac{1}{2}}) = y_2(t_i) + \frac{h}{2}\frac{dy_2(t_i)}{dt} = y_2(t_i) - \frac{h}{2}\frac{g}{l}\sin y_1(t_i) \end{cases} \tag{1.4}$$

由中点函数值计算区间斜率，再算出末端函数值:

$$\begin{cases} y_1(t_{i+1}) = y_1(t_i) + h\frac{dy_1(t_{i+1/2})}{dt} = y_1(t_i) + hy_2(t_{i+\frac{1}{2}}) \\ y_2(t_{i+1}) = y_2(t_i) + h\frac{dy_2(t_{i+1/2})}{dt} = y_2(t_i) - h\frac{g}{l}\sin y_1(t_{i+\frac{1}{2}}) \end{cases} \tag{1.5}$$

由此不断迭代就可以算出最终函数值

1.2.3 RK4

对于区间 $[t_i, t_{i+1}]$，先计算出 t_i 处斜率

$$\begin{cases} S_1(y_1) = y_2(t_i) \\ S_1(y_2) = -\frac{g}{l}\sin y_1(t_i) \end{cases} \tag{1.6}$$

由此计算出区间中点的函数值

$$\begin{cases} y_1(t_{i+\frac{1}{2}}) = y_1(t_i) + \frac{h}{2}S_1(y_1) \\ y_2(t_{i+\frac{1}{2}}) = y_2(t_i) + \frac{h}{2}S_1(y_2) \end{cases}$$

随后由区间中点初函数值计算中点处斜率

$$\begin{cases} S_2(y_1) = y_2(t_{i+\frac{1}{2}}) \\ S_2(y_2) = -\frac{g}{l}\sin y_1(t_{i+\frac{1}{2}}) \end{cases} \tag{1.7}$$

由此斜率重新计算区间中点函数值

$$\begin{cases} y_1'(t_{i+\frac{1}{2}}) = y_1(t_i) + \frac{h}{2}S_2(y_1) \\ y_2'(t_{i+\frac{1}{2}}) = y_2(t_i) + \frac{h}{2}S_2(y_2) \end{cases}$$

然后重新计算区间中点的斜率值

$$\begin{cases} S_3(y_1) = y_2'(t_{i+\frac{1}{2}}) \\ S_3(y_2) = -\frac{g}{l}\sin y_1'(t_{i+\frac{1}{2}}) \end{cases} \tag{1.8}$$

由此斜率计算出区间末端函数的估计值

$$\begin{cases} y_1(t_{i+1}) = y_1(t_i) + hS_2(y_1) \\ y_2(t_{i+2}) = y_2(t_i) + hS_2(y_2) \end{cases}$$

算出区间末端的斜率值

$$\begin{cases} S_4(y_1) = y_2(t_{i+1}) \\ S_4(y_2) = -\frac{g}{l}\sin y_1(t_{i+1}) \end{cases} \tag{1.9}$$

最后给出区间末端的函数值

$$\begin{cases} y_1(t_{i+1}) = y_1(t_i) + \frac{h}{6}(S_1(y_1) + 2S_2(y_1) + 2S_3(y_1) + S_4(y_1)) \\ y_2(t_{i+1}) = y_2(t_i) + \frac{h}{6}(S_1(y_2) + 2S_2(y_2) + 2S_3(y_2) + S_4(y_2)) \end{cases} \tag{1.10}$$

1.2.4 Euler-Trapezodal Method

主要是利用如下的迭代规则进行计算，直到满足精度要求

$$\begin{cases} y_{1,j+1}(t_{i+1}) = y_1(t_i) + \frac{h}{2}(y_2(t_i) + y_{2,j}(t_i)) \\ y_{2,j+1}(t_{i+1}) = y_2(t_i) - \frac{h}{2}\frac{g}{l}(\sin y_1(t_i) + \sin y_{1,j}(t_i)) \end{cases} \tag{1.11}$$

1.3 伪代码

Algorithm 1: Euler's Method

Input: Initail angle θ_0, ω_0, h, N

Output: Plot the angle-t and energy-t figure

$y_1[0] \leftarrow \theta_0,\ y_2[0] \leftarrow \omega_0$
for $i \leftarrow 0$ **to** N **do**
  $y_1[i+1] \leftarrow y_1[i] + hy_2[i]$
  $y_2[i+1] \leftarrow y_2[i] - h\frac{g}{l}\sin y_1[i]$
end
return y_1, y_2

Algorithm 2: Midpoint Method

Input: Initail angle θ_0, ω_0, h, N

Output: Plot the angle-t and energy-t figure

$y_1[0] \leftarrow \theta_0,\ y_2[0] \leftarrow \omega_0$
for $i \leftarrow 0$ **to** N **do**
  $\Delta y_1 \leftarrow hy_2[i]$
  $\Delta y_2 \leftarrow -h\frac{g}{l}\sin y_1[i]$
  $y_1[i+1] \leftarrow y_1[i] + h(y_2[i] + \Delta y_2/2)$
  $y_2[i+1] \leftarrow y_2[i] - h\frac{g}{l}\sin(y_1[i] + \Delta y_1/2)$
end
return y_1, y_2

Algorithm 3: RK4

Input: Initail angle θ_0, ω_0, h, N

Output: Plot the angle-t and energy-t figure

$y_1[0] \leftarrow \theta_0,\ y_2[0] \leftarrow \omega_0$
for $i \leftarrow 0$ **to** N **do**
  $S_1(y_1) \leftarrow y_2[i],\ S_1(y_2) \leftarrow -\frac{g}{l}\sin(y_1[i+1])$
  $S_2(y_1) \leftarrow y_2[i] + \frac{h}{2}S_1(y_2),\ S_2(y_2) \leftarrow -\frac{g}{l}\sin(y_1[i] + \frac{h}{2}S_1(y_1))$
  $S_3(y_1) \leftarrow y_2[i] + \frac{h}{2}S_2(y_2),\ S_3(y_2) \leftarrow -\frac{g}{l}\sin(y_1[i] + \frac{h}{2}S_2(y_1))$
  $S_4(y_1) \leftarrow y_2[i] + hS_3(y_2),\ S_4(y_2) \leftarrow -\frac{g}{l}\sin(y_1[i] + hS_3(y_1))$
  $y_1[i+1] \leftarrow y_1[i] + \frac{h}{6}[S_1(y_1) + 2S_2(y_1) + 2S_3(y_1) + S_4(y_1)]$
  $y_2[i+1] \leftarrow y_2[i] + \frac{h}{6}[S_1(y_2) + 2S_2(y_2) + 2S_3(y_2) + S_4(y_2)]$
end
return y_1, y_2

Algorithm 4: Euler-Trapezodal Method

Input: Initail angle θ_0, ω_0, h, N

Output: Plot the angle-t and energy-t figure

$y_1[0] \leftarrow \theta_0,\ y_2[0] \leftarrow \omega_0$
for $i \leftarrow 0$ **to** N **do**
  $y_{1,init} \leftarrow y_1[i],\ y_{2,init} \leftarrow y_2[i]$ `// initial` y_1, y_2
  $y_{1,adju} \leftarrow y_1[i] + hy_2[i],\ y_{2,adju} \leftarrow y_2[i] - \frac{g}{l}\sin(y_1[i])$ `// adjusted` y_1,y_2
  while $ABS(y_{1,init} - y_{1,adju}) > 10^{-5}$ *or* $ABS(y_{2,init} - y_{2,adju}) > 10^{-5}$ **do**
    $y_{1,init} \leftarrow y_{1,adju},\ y_{2,init} \leftarrow y_{2,adju}$
    $y_{1,adju} \leftarrow y_1[i] + \frac{h}{2}(y_2[i] + y_{2,init})$
    $y_{2,adju} \leftarrow y_2[i] - \frac{h}{2}\frac{g}{l}(\sin(y_1[i]) + \sin(y_{1,init}))$
  end
  $y_1[i+1] \leftarrow y_{1,adju}$
  $y_2[i+1] \leftarrow y_{2,adju}$
end
return y_1, y_2

1.4 输入输出示例

本题中我们设定 $\theta_0 = 0.5rad,\ \omega_0 = 0$，小球质量 $m = 1kg$，绳长 $l = 1m$，重力加速度 $g = 9.8m/s^2$。并取区间间隔 $h = 0.01, N = 1000$，绘制出用四种不同方法计算得到的 $\theta \sim t$ 关系图如图 1 所示。

其中 Euler's Method 计算的结果（图 1 中蓝色线）精度较低，计算结果会发散，即振幅会逐渐增大。其余三种算法计算得到的结果精度较高，因此三条曲线重合在了一起，只显示为 1 条绿色曲线。

将四种方法计算得到的能量 E 与时间 t 的关系绘制在图 2∼5 中

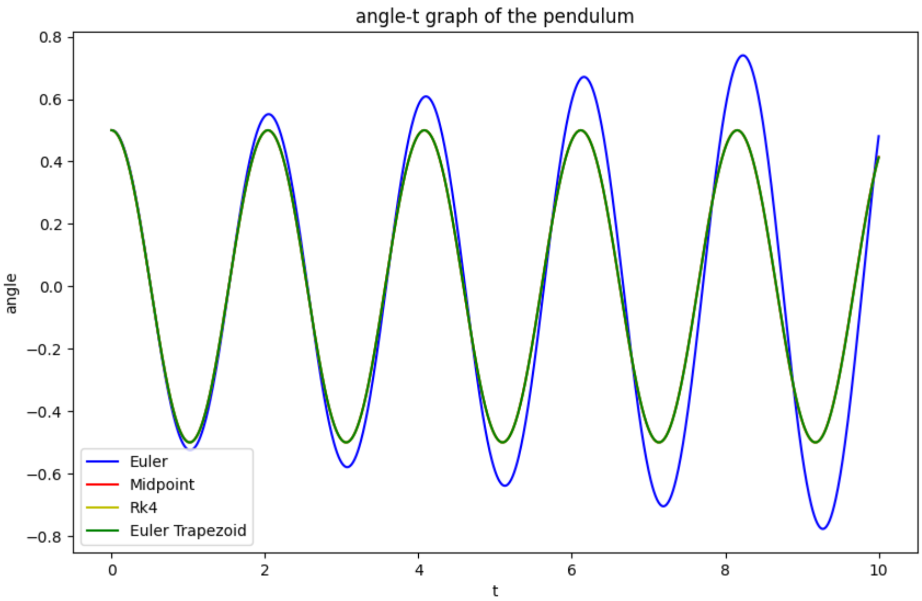

图 1: 不同算法得到的 $\theta \sim t$ 关系图

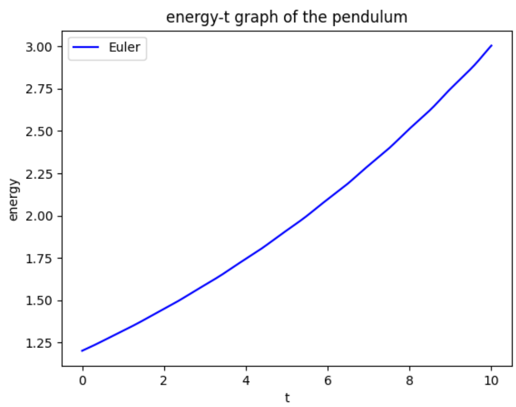

图 2: Euler's Method

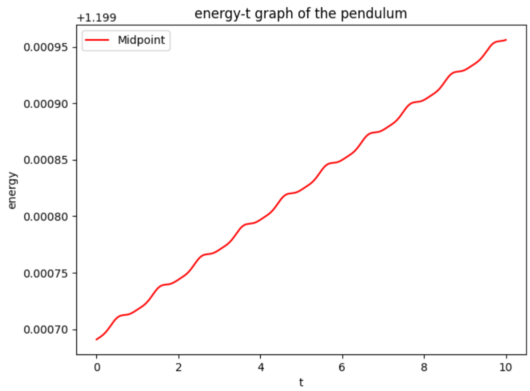

图 3: Midpoint Method

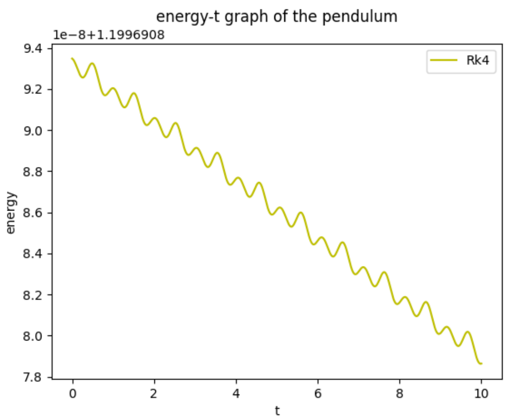

图 4: RK4 Method

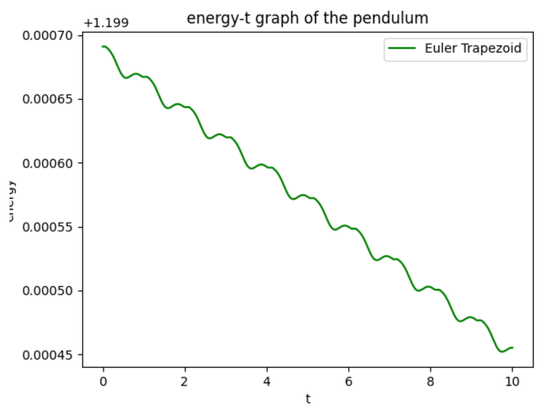

图 5: Euler-Trapezoidal Method

从上面四张图可以看出**精度最高的是 RK4 Method**，其次是 Euler-Trapezoidal Method，再次是 Midpoint Method，**精度最差的是 Euler's Method**.

1.5 用户手册

1. 本程序的源程序为 Pendulum.py
2. 在执行源程序之前，应当先安装 numpy, matplotlib, math 库
3. 本程序分别利用 Euler's Method, Midpoint Method, RK4 Method 和 Euler-Trapezoidal Method 求解单摆的周期性运动
4. 运行程序后，终端将提示"Please input the times you want to calculate:"，提示您输入想要计算的次数，即 N 的值
5. 输入完毕后按下回车键，将依次绘制出 $\theta \sim t$ 关系图和四种算法下的 $E \sim t$ 关系图

2 Radial Schrodinger Equation

2.1 题目描述

Write a code to numerically solves radial Schrödinger equation for:

$$\left[-\frac{1}{2}\nabla^2 + V(\mathbf{r})\right]\psi(\mathbf{r}) = E\psi(\mathbf{r}), V(\mathbf{r}) = V(r) \tag{2.1}$$

(1) $V(r) = -\frac{1}{r}$ (hydrogen atom)

(2) $Z_{ion} = 0.4000000, C_1 = -14.0093922, C_2 = 9.5099073, C_3 = -1.7532723, C_4 = 0.0834586$

$$\begin{aligned} V(r) = &-\frac{Z_{ion}}{r}erf\left(\frac{r}{\sqrt{2}r_{loc}}\right) + exp\left[-\frac{1}{2}\left(\frac{r}{r_{loc}}\right)^2\right] \\ &\times\left[C_1 + C_2\left(\frac{r}{r_{loc}}\right)^2 + C_3\left(\frac{r}{r_{loc}}\right)^4 + C_4\left(\frac{r}{r_{loc}}\right)^6\right] \end{aligned}$$

Compute and plot the first three eigenstates.

2.2 解决方案描述

首先我们进行分分离变量求解：$\psi(\mathbf{r}) = R_l(r)Y_l^m(\theta,\phi) = \left[\frac{u_l(r)}{r}\right]Y_l^m(\theta,\phi)$，由于 $Y_l^m(\theta,\phi)$ 为球函数，是已知的，因此我们只需要求解关于 r 的方程

$$\left\{\frac{\hbar^2}{2m}\frac{d^2}{dr^2} + \left[E - V(r) - \frac{\hbar^2 l(l+1)}{2mr^2}\right]\right\}u_l(r) = 0,\ u_l(0) = 0 \tag{2.2}$$

本题采用打靶法和割线求根法相结合的方式对该问题进行求解。该问题为一个边值问题，对应的边界条件为:

$$u(0) = 0,\ , u(r_{max}) = 0 \tag{2.3}$$

其中 r_{max} 一般取为 $30a$ 其中 a 为玻尔半径。因此，本题一共分两个步骤来实现:

1. 给定初始的 E, $u(r_{max}) = 0$, $u'(r_{max}) = 1$（可以证明在边界处导数值对最终结果没有影响，因此这个值任意）。然后利用 RK4 方法计算出 $u(r = 0)$ 处的函数值。为了减少迭代次数，我们做变量代换：$r(t) = r_p(e^t - 1), t \in [0, t_{max}]$，并令 $y_1 = u, y_2 = \frac{du}{dt}$，得到如下的微分方程组:

$$\begin{cases} \frac{dy_1}{dt} = y_2 \\ \frac{dy_2}{dt} = y_2 - \frac{2mr_p^2e^{2t}y_1}{\hbar^2}\left[E - V(r(t)) - \frac{\hbar^2 l(l+1)}{2mr^2(t)}\right] \end{cases} \tag{2.4}$$

2. 将上一步计算得到的 $u(0)$ 与 0 进行比较，如果 $u(0) \neq 0$，则通过割线法调整 E 的值，再次计算 $u(0)$，直到满足精度要求为止，其中割线法调整 E 的迭代公式如下:

$$E_{k+1} = E_k - u(r = 0, E_k)\frac{(E_k - E_{k-1})}{u(r = 0, E_k) - u(r = 0, E_{k-1})} \tag{2.5}$$

2.3 伪代码

Algorithm 5: Radial Schrodinger Equation

Input: Initial energy E, l, N, h

Output: Plot the first three eigenstates function

def *RK4*(E, y_1, y_2, l)*:*
  def *deri_y2*(y_1, y_2, t)*:*
    return $y_2 - \frac{2mr_p^2 e^{2t} y_1}{\hbar^2}\left[E - V(r(t)) - \frac{\hbar^2 l(l+1)}{2mr^2(t)}\right]$
  end
  for $i \leftarrow N$ **to** 1 **do**
    $S_1(y_1) \leftarrow y_2[i],\ S_1(y_2) \leftarrow deri\_y2(y_1[i], y_2[i], t[i])$
    $S_2(y_1) \leftarrow y_2[i] - \frac{h}{2}S_1(y_2)$
    $S_2(y_2) \leftarrow deri\_y2\left(y_1[i] - \frac{h}{2}S_1(y_1), y_2[i] - \frac{h}{2}S_1(y_2), t[i]\right)$
    $S_3(y_1) \leftarrow y_2[i] - \frac{h}{2}S_2(y_2)$
    $S_3(y_2) \leftarrow deri\_y2\left(y_1[i] - \frac{h}{2}S_2(y_1), y_2[i] - \frac{h}{2}S_2(y_2), t[i]\right)$
    $S_4(y_1) \leftarrow y_2[i] - hS_3(y_2)$
    $S_4(y_2) \leftarrow deri\_y2\left(y_1[i] - hS_2(y_1), y_2[i] - hS_2(y_2), t[i-1]\right)$
    $y_1[i+1] \leftarrow y_1[i] - \frac{h}{6}[S_1(y_1) + 2S_2(y_1) + 2S_3(y_1) + S_4(y_1)]$
    $y_2[i+1] \leftarrow y_2[i] - \frac{h}{6}[S_1(y_2) + 2S_2(y_2) + 2S_3(y_2) + S_4(y_2)]$
  end
  return y_1, y_2
end
def *shoot*(E, l)
  $y_1[N] \leftarrow 0,\ y_2[N] \leftarrow 1$
  $E_{init} \leftarrow E,\ E_{adju} \leftarrow E - eps$
  $y_{1,init}(0) \leftarrow \mathrm{RK4}(E_{init}, y_1, y_2)$
  while *ABS(RK4*(E_{adju}, y_1, y_2)*))* $> 10^{-5}$ **do**
    $y_{1,adju}(0) \leftarrow \mathrm{RK4}(E_{adju}, y_1, y_2)$
    $tmp \leftarrow E_{adju}$
    $E_{adju} \leftarrow y_{1,init}(0)\frac{E_{adju} - E_{init}}{y_{1,adju}(0) - y_{1,init}(0)}$
    $E_{init} \leftarrow tmp,\ y_{1,init}(0) \leftarrow y_{1,adju}(0)$
  end
  return $E_{adju},\ y_1$
end
$E,\ u \leftarrow \mathrm{shoot}(E, l)$ `// adjust E to obtain different eigenstates`
return E
$\mathrm{PLOT}(r, \frac{u}{r})$

2.4 输入/输出示例

本题目中将为简便计算取 $\hbar = 1, m = 1, a = 1$

2.4.1 Hydrogen Atom

对氢原子而言，由于初始时不知道氢原子的能量本征值位于哪个区间内，因此我们先取一系列 E 的值，利用 RK4 方法算出 u 在 $r = 0$ 处的值，作出 $u(r = 0) \sim E$ 的关系图，分别取 $l = 0$ 与 $l = 1$，关系图如下：

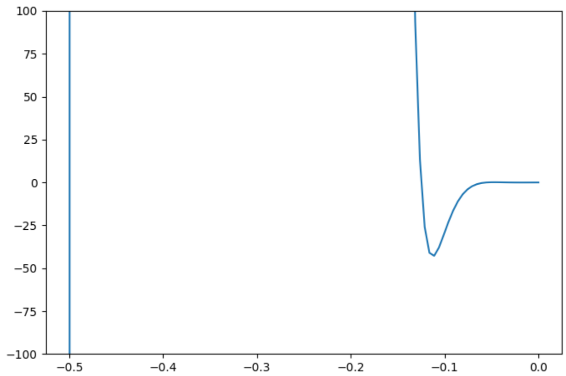

图 6: 氢原子 $l = 0$，$u(0) \sim E$ 关系图

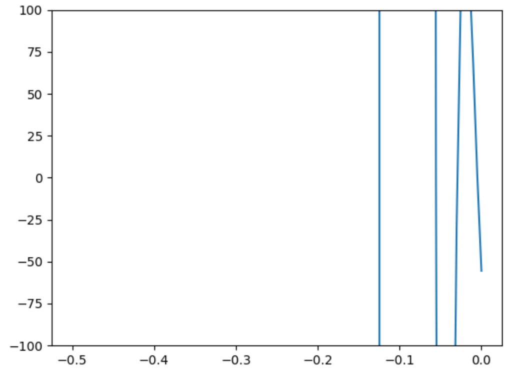

图 7: 氢原子 $l = 1$，$u(0) \sim E$ 关系图

从图中我们可以看出，对于 $l = 0$，E 的最小两个取值分别在 -0.5 和 -0.15 附近，对 $l = 1$，E 的最小取值在 -0.15 附近，将其作为 E 的初始值输入程序中，就可得到氢原子前三个最小能量对应的能量本征值和波函数，分别为 $1s, 2s, 2p$ 三个状态。

输入输出结果如下表，最终求得的三个状态波函数图像如下图。

Input		Output	Note
$E_{initial}$	l	E	
-0.8	0	-0.49965213872799520	1s 态
-0.3	0	-0.12495637986041683	2s 态
-0.3	1	-0.12499973448495807	2p 态

表 1: 氢原子三个最小能量态

从能量的结果中我们可以看出，这三个能量分别约等于 $-\frac{1}{2}\frac{1}{1^2}, -\frac{1}{2}\frac{1}{2^2}, -\frac{1}{2}\frac{1}{2^2}$，与氢原子能级 $E_n = -\frac{1}{2}\frac{1}{n^2}$ 一致。同时查阅资料，发现本程序计算得到的 $R(r)$ 波函数图像也与实际结果一致，只是本程序中没有对波函数进行归一化处理。

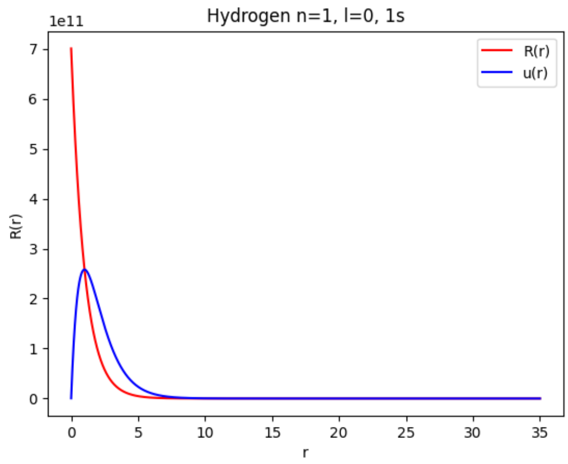

图 8: 氢原子 $1s$ 态，$u(r)$ 与 $R(r)$ 图

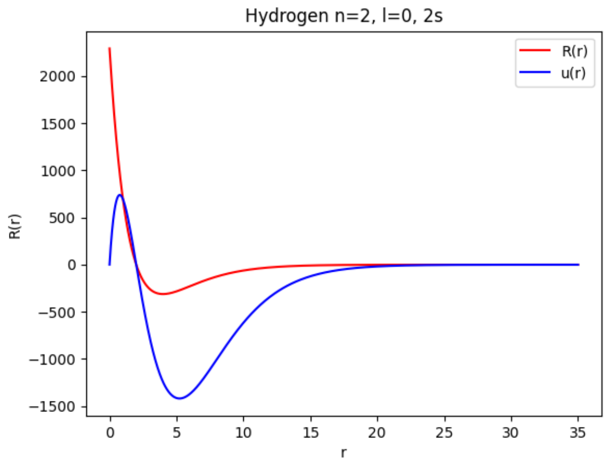

图 9: 氢原子 $2s$ 态，$u(r)$ 与 $R(r)$ 图

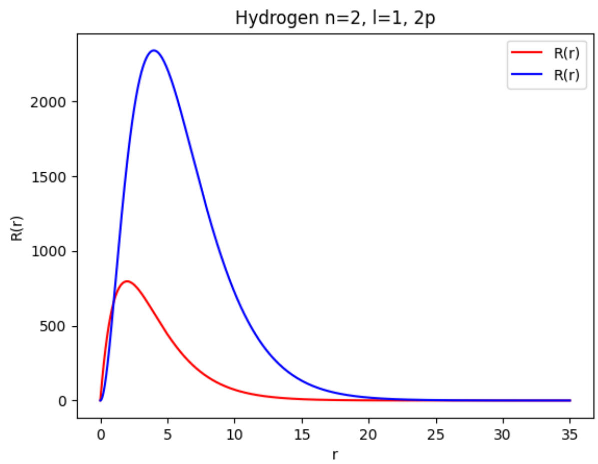

图 10: 氢原子 $2p$ 态，$u(r)$ 与 $R(r)$ 图

2.4.2 Li

类似的，我们先取一系列 E 的值，利用 RK4 方法算出 u 在 $r=0$ 处的值，作出 $u(r=0)\sim E$ 的关系图，分别取 $l=0$ 与 $l=1$，关系图如下图所示。

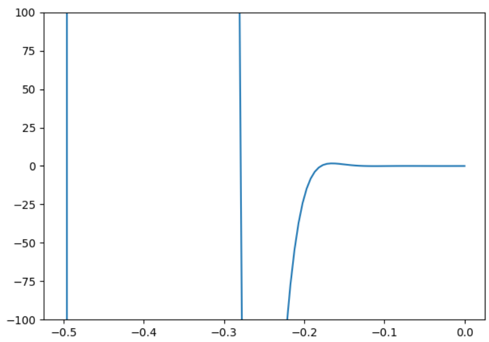

图 11: 锂原子 $l=0$，$u(0)\sim E$ 关系图

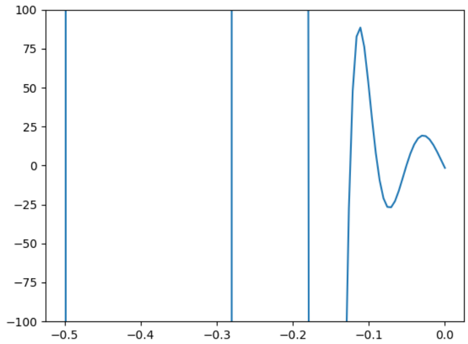

图 12: 锂原子 $l=1$，$u(0)\sim E$ 关系图

对于 $l=0$，E 的最小两个取值分别在 -0.5 和 -0.3 附近，对 $l=1$，E 的最小取值在 -0.5 和 -0.3 附近，将其作为 E 的初始值输入程序中，就可得到锂原子前三个最小能量对应的能量本征值和波函数。

输入输出结果如下表，最终求得的三个状态波函数图像如下图。

Input		Output	Note
$E_{initial}$	l	E	
-1	0	-0.4961297798159532	
-0.4	0	-0.2806395973910577	
-1	1	-0.4986783392501938	

表 2: 锂原子三个最小能量态

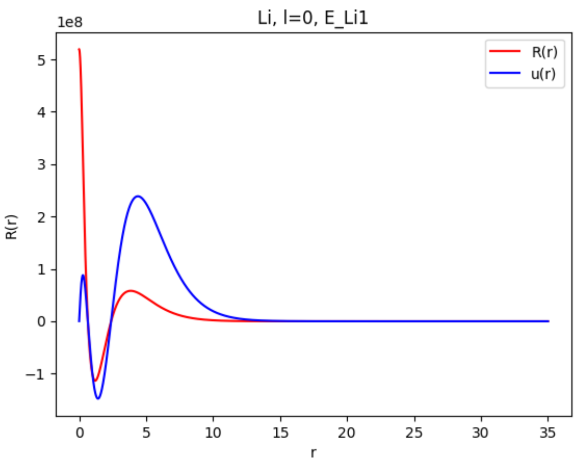

图 13: 锂原子 $l=0, E=-0.496$ 态，$u(r)$ 与 $R(r)$ 图

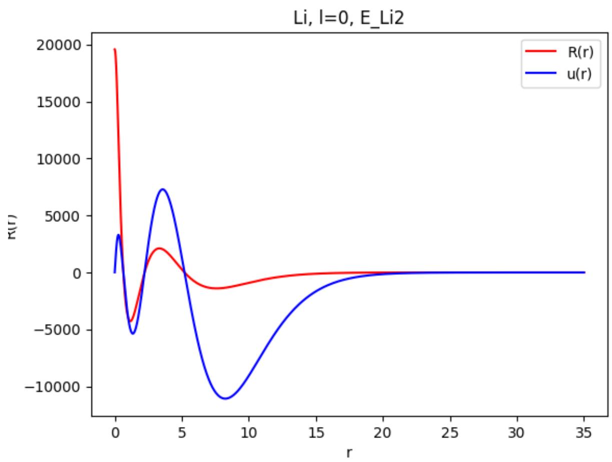

图 14: 锂原子 $l=0, E=-0.280$ 态，$u(r)$ 与 $R(r)$ 图

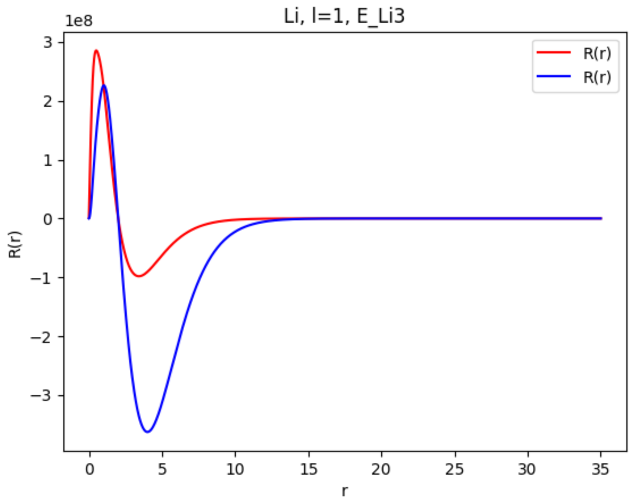

图 15: 锂原子 $l = 1, E = -0.499$ 态，$u(r)$ 与 $R(r)$ 图

2.5 用户手册

1. 本程序的源程序为 Radial_Schro.py
2. 在执行源程序之前，应当先安装 numpy, matplotlib, math 库
3. 本程序分别利用 RK4 方法和割线求根法求解氢原子和锂原子波函数
4. 运行程序后，将分别输出氢原子和锂原子最小三个能量本征值对应的 R 波函数（共六张图），并在终端输出其能量值