第七次作业

尹朝阳 物理学系

1 题目 1

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 程序描述

本题目要求用以下 4 种解法对于单摆的运动进行求解，绘制角度随时间的变化函数以及能量随时间的变化函数。

首先，单摆的运动方程为

$$\frac{d^2\theta}{dt^2} = -\frac{g}{l}\sin\theta$$

这是一个二阶微分方程，我们将其改造为如下形式：

$$\begin{cases} \dfrac{d\theta}{dt} = \omega \\ \dfrac{d\omega}{dt} = -\dfrac{g}{l}\sin\theta \end{cases}$$

认为 $[\theta,\omega]$ 为一个向量 X，则有

$$\frac{dX}{dt} = f(X(t),t)$$

其中 $f(X(t),t)$ 满足如上的迭代关系。

1. Euler’s method

将 $X(t)$ 在 t_0 处展开，有 $X(t) = X(t_0) + \frac{dX}{dt}(t-t_0) + O(t^2)$，又因为 $\frac{dX}{dt} = f(X(t),t)$，则有 $X(t) = X(t_0) + f(X(t_0),t_0)(t-t_0) + O(t^2)$，则得到迭代关系：

$$X[t_{i+1}] = X[t_i] + f(X[t_i],t_i) * (t_{i+1} - t_i)$$

2. midpoint method

由于 Euler’s method 采用 t_i 处的导数值进行 t_{i+1} 的迭代，误差较大，采用 midpoint method 进行估计。midpoint method 的迭代公式如下：

$$\Delta X[t_i] = \Delta t * f(X[t_i],t_i)$$
$$\Delta X[t_{i+0.5}] = \Delta t * f(X[t_i] + \frac{\Delta X[t_i]}{2}, t_i + \frac{\Delta t}{2})$$
$$X[t_{i+1}] = X[t_i] + \Delta X[t_{i+0.5}]$$

最终可以整合为

$$X[t_{i+1}] = X[t_i] + \Delta t * f(X[t_i] + \frac{\Delta t * f(X[t_i], t_i)}{2}, t_i + \frac{\Delta t}{2})$$

3. RK4 method

RK4 method 的迭代公式如下：

$$\begin{aligned} K_1 &= f(X[t_i], t_i) \\ K_2 &= f(X[t_i] + \frac{\Delta t}{2} * K_1, t_i + \frac{\Delta t}{2}) \\ K_3 &= f(X[t_i] + \frac{\Delta t}{2} * K_2, t_i + \frac{\Delta t}{2}) \\ K_4 &= f(X[t_i] + \Delta t * K_3, t_i + \Delta t) \\ X[t_{i+1}] &= X[t_i] + \frac{\Delta t}{6} * (K_1 + 2K_2 + 2K_3 + K_4) \end{aligned}$$

4. Euler-trapezoidal method

Euler-trapezoidal method 在 Euler method 的基础上，加入了对于每一步 $X[t_{i+1}]$ 的优化。此时 X 是一个矩阵，我们关心的也是最终要取的是这个矩阵的每一列中（第 i 列代表 $X[t_i]$）的最终的迭代值。

其迭代公式如下：

$$\begin{aligned} X[t_{i+1}]_0 &= X[t_i]_m + \Delta t * f(X[t_i]_m, t_i) \\ X[t_{i+1}]_{j+1} &= X[t_i]_m + \frac{\Delta t}{2}[f(X[t_i]_m, t_i) + f(X[t_{i+1}]_j, t_{i+1}) \end{aligned}$$

每一个 $X[t_i]$ 都是一个列向量，式中的 m 代表 t_i 时刻的 X 经过迭代之后的值其列序号为 m，其中迭代完成的标志为

$$|X[t_i]_m - X[t_i]_{m-1}| < \epsilon$$

这里 ϵ 为求解常微分方程所要求的精度。

最后，将 $X(t)$ 从实现函数中传递出来后，分别得到 $\theta(t)$ 和 $\omega(t)$，则能量有如下公式给出：

$$E(t) = mgl(1 - \cos\theta) + \frac{1}{2}ml^2\omega^2$$

本题目的源程序位于 Solve_motion_of_simple_pendulum 文件夹中，导入的第三方库有 numpy 和 matplotlib。

1.3 伪代码

本题目的伪代码将分别给出映射关系 $f(X(t), t)$ 以及 4 种方法的伪代码实现，一共五个伪代码块。在将 $\theta(t)$ 和 $\omega(t)$ 对应上相应的函数接口传递出的 $X(t)$ 后，可以直接代入能量公式解出能量，较为简单，不再赘述。

Algorithm 1.1 mapping of X onto $\frac{dX}{dt} : f(X(t), t)$

function $\textsc{f}(X, t)$
    $X_{forward}[0] \leftarrow X[1]$
    $X_{forward}[1] \leftarrow -\frac{g}{l} * \sin(X[0])$
    return $X_{forward}$
end function

Algorithm 1.2 Euler's method

$X_1 \leftarrow [\theta_0, \omega_0]$
for $i \leftarrow 1$ to $n - 1$ **do** ▷ 1 to n as the time span
    $X_{i+1} \leftarrow X_i + f(X_i, t_i) * \Delta t$ ▷ Δt as the step
end for
return X

Algorithm 1.3 midpoint method

$X_1 \leftarrow [\theta_0, \omega_0]$
for $i \leftarrow 1$ to $n - 1$ **do** ▷ 1 to n as the time span
    $X_{i+1} \leftarrow X_i + \Delta t * f(X_i + \frac{\Delta t}{2} * f(X_i, t_i), t_i + \frac{\Delta t}{2})$ ▷ Δt as the step
end for
return X

Algorithm 1.4 RK4 method

$X_1 \leftarrow [\theta_0, \omega_0]$
for $i \leftarrow 1$ to $n-1$ **do** ▷ 1 to n as the time span
    $K_1 \leftarrow f(X_i, t_i)$
    $K_2 \leftarrow f(X_i + \frac{\Delta t}{2} * K_1, t_i + \frac{\Delta t}{2})$ ▷ Δt as the step
    $K_3 \leftarrow f(X_i + \frac{\Delta t}{2} * K_2, t_i + \frac{\Delta t}{2})$
    $K_4 \leftarrow f(X_i + \Delta t * K_3, t_i + \Delta t)$
    $X_{i+1} \leftarrow X_i + \frac{\Delta t}{6} * (K_1 + 2K_2 + 2K_3 + K_4)$
end for
return X

Algorithm 1.5 Euler-trapezoidal method

$X_{00} \leftarrow [\theta_0, \omega_0]$
$m_0 \leftarrow 0$
for $i \leftarrow 0$ to $n-1$ **do** ▷ 0 to n as the time span
    $j \leftarrow 0$
    $X_{i+1,0} \leftarrow X_{i,m_i} + \Delta t * f(X_{i,m_i}, t_i)$ ▷ Δt as the step
    $X_{i+1,1} \leftarrow X_{i,m_i} + \frac{\Delta t}{2}(f(X_{i,m_i}, t_i) + f(X_{i+1,0}, t_{i+1}))$
    while $|X_{i+1,j+1} - X_{i+1,j}| > \epsilon$ **do**
        $j \leftarrow j + 1$
        $X_{i+1,j+1} \leftarrow X_{i,m_i} + \frac{\Delta t}{2}(f(X_{i,m_i}, t_i) + f(X_{i+1,j}, t_{i+1}))$
    end while
    $m_{i+1} \leftarrow j + 1$
end for
return X, m
for $i \leftarrow 0$ to n **do**
    $\theta_i \leftarrow X_{0,m_i}$
    $\omega_i \leftarrow X_{1,m_i}$
end for

1.4 测试用例

取 $m = 1kg, l = 20m, g = 9.80m/s^2$，时间跨度为 $0 - 60$ 秒，步长为 0.1 秒，初值为——角度为 $\pi/6$、角速度为 0。4 种方法的最终结果如下（其中 RK4 method 的能量为能量与初值的偏差）：

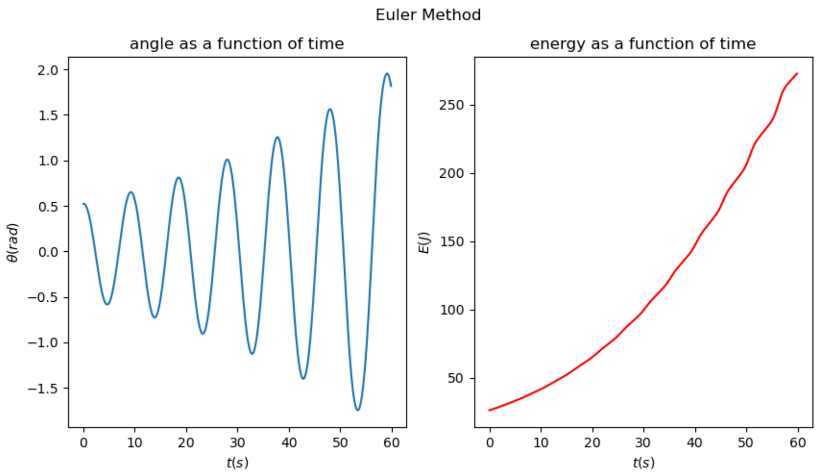

图 1: Euler method

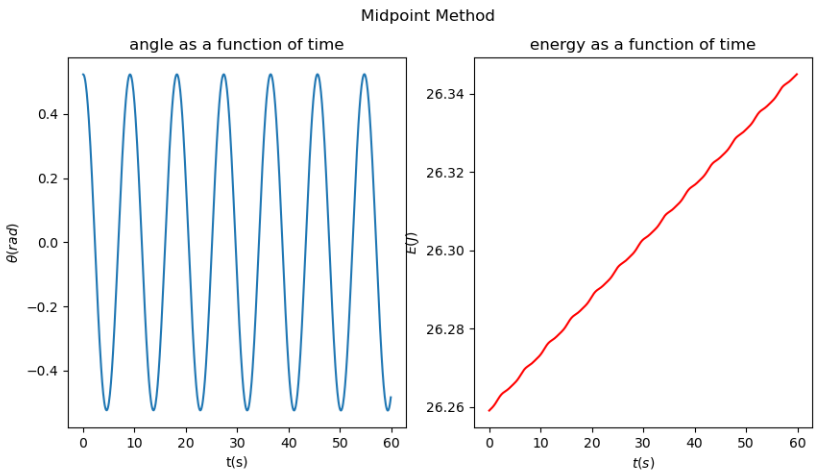

图 2: Midpoint method

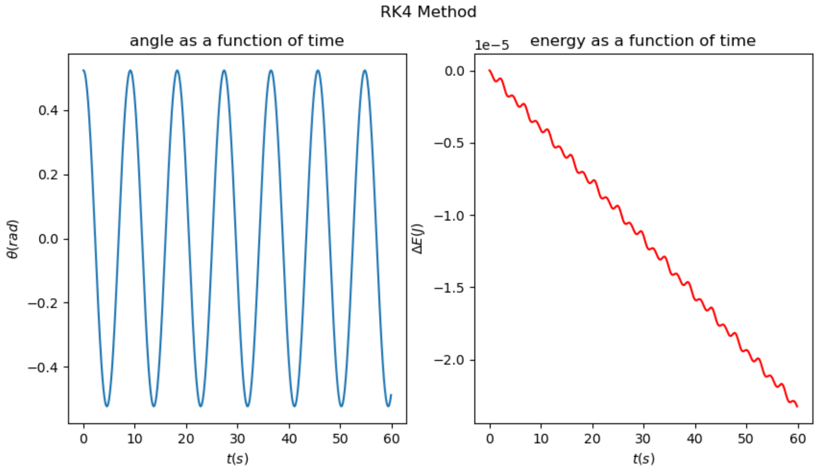

图 3: RK4 method

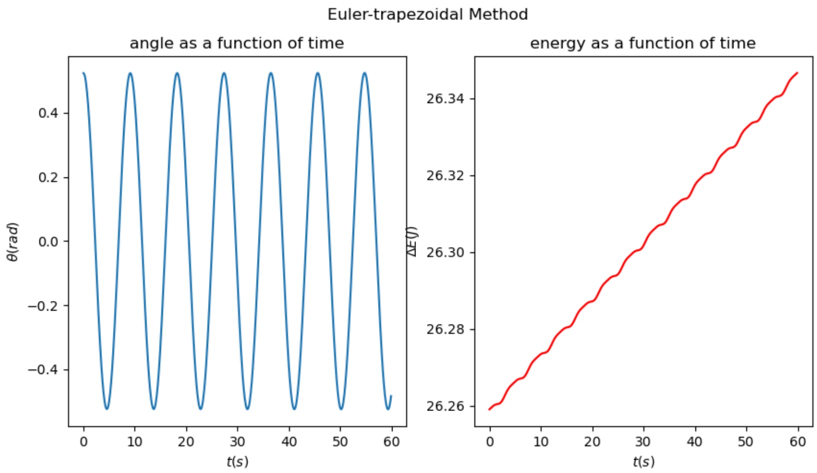

图 4: Euler-trapezoidal method

可以发现，前三个中，Euler method 的结果最差，Midpoint method 的结果较好，RK4 method 的结果最好。这是因为 Euler method 的局部误差为 h^2（h 为步长），总体误差为 h；Midpoint method 的局部误差为 h^3，总体误差为 h^2；RK4 method 的局部误差为 h^5，总体误差为 h^4，故在选定步长的条件下，有 Euler method 的精度小于 Midpoint method 的精度小于 RK4 method 的精度。而对于第四个 Euler-trapezoidal method，虽然其采用 Euler method 作为向前演化的算法，但是对于每一个时间 t 处的值，该方法也用 ϵ 进行了精度控制，故其不只受步长的影响，也受人为选取的 ϵ 的调控。

2 题目 2

2.1 题目描述

Write a code to numerically solves radial Schrödinger equation for

$$[-\frac{1}{2}\nabla^2 + V(\vec{r})]\phi(\vec{r}) = E\phi(\vec{r})$$

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

(2) $V_{loc}(r) = \frac{-Z_{ion}}{r} erf(\frac{r}{\sqrt{2}r_{loc}}) + e^{-\frac{1}{2}(\frac{r}{r_{loc}})^2} \times [C_1 + C_2(\frac{r}{r_{loc}})^2 + C_3(\frac{r}{r_{loc}})^4 + C_4(\frac{r}{r_{loc}})^6]$

2.2 程序描述

本题目要求解薛定谔方程的径向方程。

首先，由径向函数的性质，存在一个 r_{max}，使得 $R(r > r_{max}) = 0$。故我们只需要关注 $0 - r_{max}$ 处的情况。

将 $0 - r_{max}$ 均分为 $x_0, x_1, ..., x_M$ 个点，其中 $x_0 = 0, x_M = r_{max}$，满足 $u(x_0) = 0, u(x_M) = 0$。设该点列的步长为 s，则利用中心差分方法，有

$$\frac{d^2u(x)}{dx^2} = \frac{u(x+s) + u(x-s) - 2u(x)}{s^2}$$

则薛定谔方程变为

$$-\frac{1}{2s^2}\phi(x_{n-1}) + [\frac{1}{s^2} + V(x_n)]\phi(x_n) - \frac{1}{2s^2}\phi(x_{n+1}) = E * \phi(x_n)$$

对于 $x_1, x_2, ..., x_{M-1}$ 个点。一共有 $M-1$ 个线性方程，构成线性方程组，可以转化为矩阵求解。该矩阵满足：主元为 $H_{ii} = \frac{1}{s^2} + V(x_{i-1})$ 对于 $i = 0, 1, 2, ..., M-2$ 成立，辅元为 $H_{i+1,i} = H_{i,i+1} = -\frac{1}{2s^2}$ 对于 $i = 0, 1, 2, ..., M-3$ 成立。

最终通过求解本征值和本征向量，可以得出一系列本征能量（本征能量就是求解出来的本征值）以及相应的径向函数。

本题目的源程序位于 Solve_radial_Schrodinger_equation 文件夹中，potential_of_hydrogen_atom.py 为第一问，potential_2.py 为第二问。导入的第三方库有 numpy 和 matplotlib。

2.3 伪代码

Algorithm 2.1 Numerically solve the radial Schrodinger equation with the potential $V = -1/r$

function POTENTIAL(r)
    return $-1/r$
end function

$x \leftarrow [0 : n : h]$ ▷ h as the step
$n \leftarrow$ the shape of the vector x
for $i \leftarrow 1$ to $n - 1$ **do**
    for $j \leftarrow 1$ to $n - 1$ **do**
        if $i = j$ **then**
            $H_{ij} \leftarrow 1/h^2 + Potential(x_i)$
        else if $i = j + 1$ **or** $i = j - 1$ **then**
            $H_{ij} \leftarrow -1/(2h^2)$
        end if
    end for
end for
$eigenvalues \leftarrow$ the eigenvalues of H matrix
$eigenvectors \leftarrow$ the eigenvectors of H matrix
$index \leftarrow$ the indexes of the 3 smallest values in $eigenvalues$
for $i \leftarrow 1$ to 3 **do**
    $u_i[0] \leftarrow 0$
    $u_i[n] \leftarrow 0$
    for $j \leftarrow 1$ to $n - 1$ **do**
        $u_i[j] \leftarrow eigenvectors[j][index[i]]$
    end for
    plot u_i as the function of x with its energy $eigenvalues[index[i]]$
end for

Algorithm 2.2 Numerically solve the radial Schrodinger equation with the effective potential

function POTENTIAL(r)
    return $-Z/r * erf(r/(\sqrt{2}r_{loc})) + e^{-0.5*(r/r_{loc})^2} * [C_1 + C_2(r/r_{loc})^2 + C_3(r/r_{loc})^4 + C_4(r/r_{loc})^6]$
end function

$x \leftarrow [0 : n : h]$ ▷ h as the step
$n \leftarrow$ the shape of the vector x
for $i \leftarrow 1$ to $n - 1$ **do**
    for $j \leftarrow 1$ to $n - 1$ **do**
        if $i = j$ **then**
            $H_{ij} \leftarrow 1/h^2 + Potential(x_i)$
        else if $i = j + 1$ **or** $i = j - 1$ **then**
            $H_{ij} \leftarrow -1/(2h^2)$
        end if
    end for
end for
$eigenvalues \leftarrow$ the eigenvalues of H matrix
$eigenvectors \leftarrow$ the eigenvectors of H matrix
$index \leftarrow$ the indexes of the 3 smallest values in $eigenvalues$
for $i \leftarrow 1$ to 3 **do**
    $u_i[0] \leftarrow 0$
    $u_i[n] \leftarrow 0$
    for $j \leftarrow 1$ to $n - 1$ **do**
        $u_i[j] \leftarrow eigenvectors[j][index[i]]$
    end for
    plot u_i as the function of x with its energy $eigenvalues[index[i]]$
end for

2.4 测试用例

最终解得的最低的三个本征能量及其对应的径向函数 $u(r)$（注：$R(r)=u(r)/r$）如下图所示。

第一问：

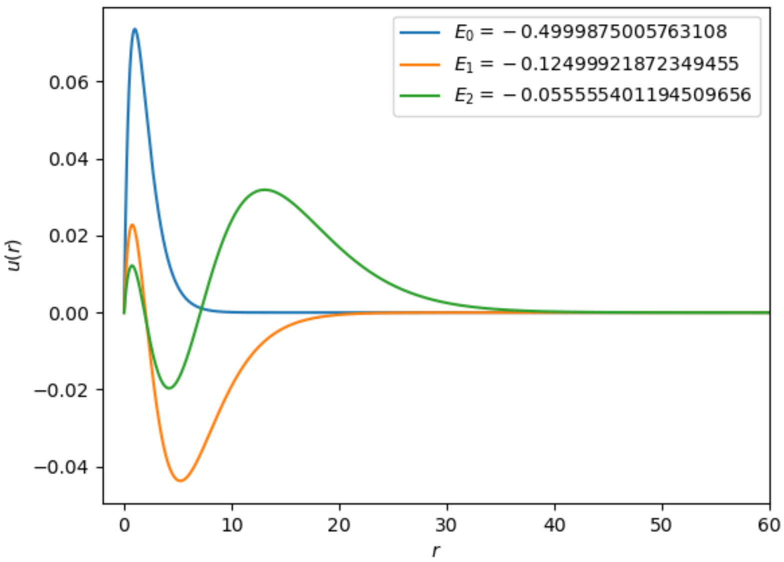

图 5: hydrogen atom

第二问：

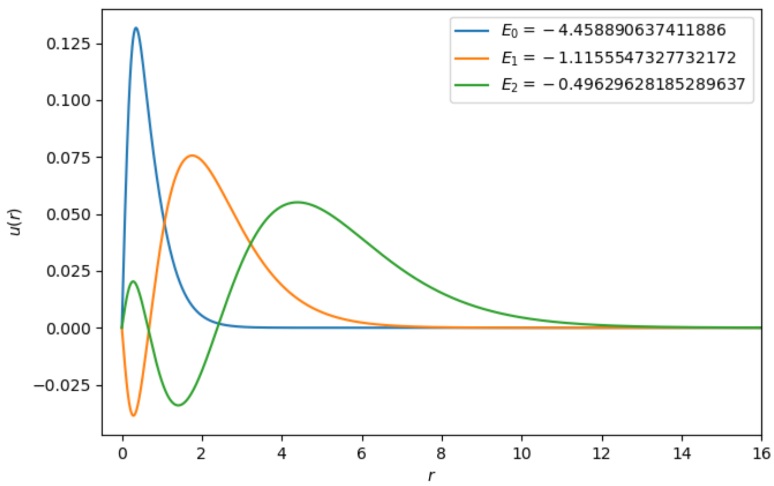

图 6: effective potential