Homework7

Zric

January 8, 2026

1 Problem1: Simple Pendulum

1.1 problem description

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 algorithm description

The equation of motion of a simple pendulum is given by:

$$\frac{d^2}{dt^2}\theta(t) + \frac{g}{l}\sin\theta(t) = 0 \tag{1}$$

Where $\theta(t)$ is the angle of the pendulum at time t, m is the mass of the pendulum, g is the acceleration due to gravity, and l is the length of the pendulum.

For numerical solution, we can convert this second-order differential equation into a system of 2 first-order differential equations by introducing the angular velocity $\omega(t)$:

$$\frac{d\theta(t)}{dt} = \omega(t) \tag{2}$$

$$\frac{d\omega(t)}{dt} = -\frac{g}{l}\sin\theta(t) \tag{3}$$

So that the state vector can be defined as:

$$\mathbf{y}(t) = \begin{bmatrix}\theta(t)\\ \omega(t)\end{bmatrix} \tag{4}$$

and equation(2)(3) can be written in first-order ODE form as:

$$\frac{d\mathbf{y}(t)}{dt} = \begin{bmatrix}\omega(t)\\ -\frac{g}{l}\sin\theta(t)\end{bmatrix} = \mathbf{f}(\mathbf{y}(t), t) \tag{5}$$

We can then apply the 4 numerical methods in first-order ODE to solve this system of equations.Total energy of the simple pendulum is given by:

$$E(t) = \frac{1}{2}ml^2\omega(t)^2 + mgl(1-\cos\theta(t)) \tag{6}$$

In my code, I set time interval $t \in [0, 20]$ seconds, time step $\Delta t = 0.01$ seconds, and compute the angle $\theta(t)$ and total energy $E(t)$ at each time step using the 4 numerical methods. You can choose different initial conditions for $\theta(0)$ and $\omega(0)$ to see the results.

1.3 output

run `problem1.py`

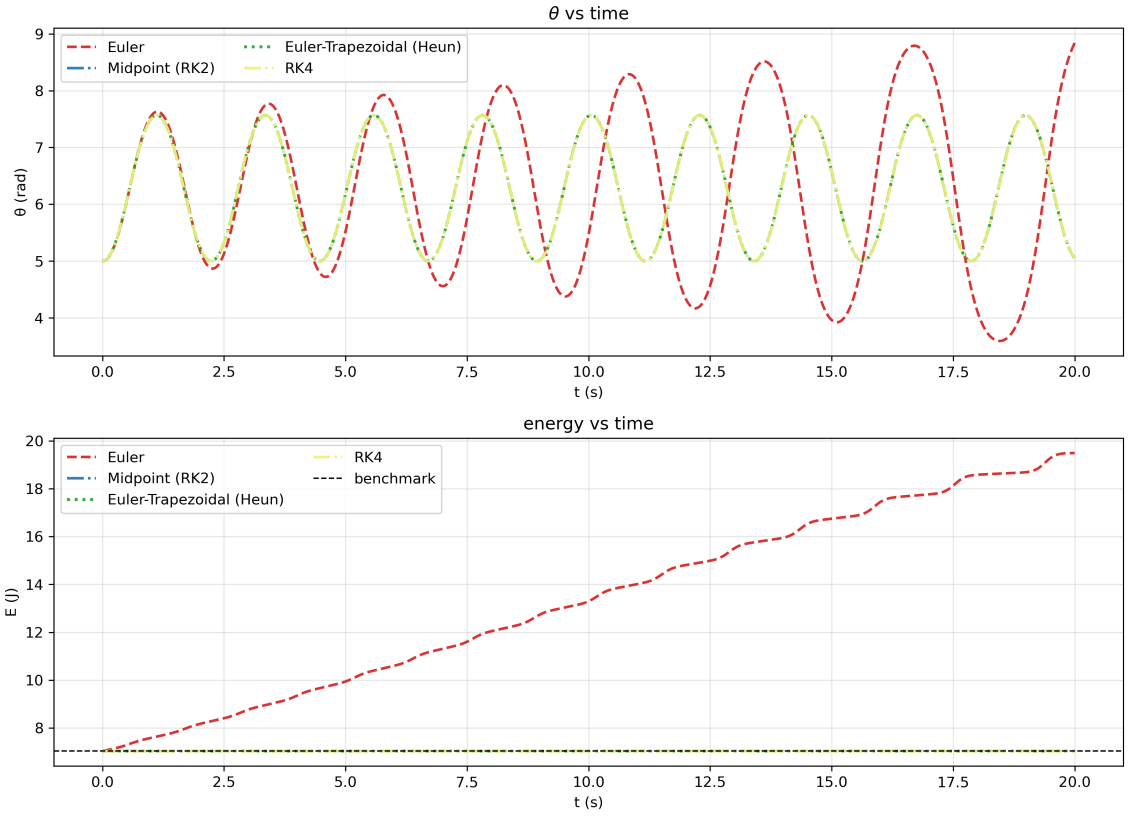

Figure 1: θ and energy vs Time for Simple Pendulum using Different Numerical Methods

Analysis:

In Figure 1, we can see that Euler's method gives a diverging solution for both angle and energy. Its angle and energy will drift away in a fixed direction from the true solution over time, depending on initial conditions. This is because Euler's method is not symplectic and does not conserve energy, leading to numerical instability over long time simulations. Explain in equation:

$$E_{i+1} - E_i = \frac{1}{2}ml^2(\omega_{i+1}^2 - \omega_i^2) - mgl(\cos\theta_{i+1} - \cos\theta_i) \tag{7}$$

$$= \frac{1}{2}\left(\frac{1}{2}ml^2\frac{d^2}{dt^2}\omega_i^2 - mgl\frac{d^2}{dt^2}\cos\theta_i\right)\Delta t^2 + O(\Delta t^3) \tag{8}$$

$$= \frac{1}{2}\left[ml^2(\dot{\omega}^2 + \omega\ddot{\omega}) + mgl(\cos\theta\omega^2 + \sin\theta\dot{\omega})\right]\Delta t^2 + O(\Delta t^3) \tag{9}$$

$$= mgl\cos\theta_i\omega_i^2\Delta t^2 + O(\Delta t^3) \tag{10}$$

In Euler's method the error per step is $O(h^2)$, at small θ_i the second-order error term $mgl\cos\theta_i\omega_i^2 > 0$, remaining positive leading to energy increasing.

Here the midpoint method, Euler-trapezoidal method and RK4 method provide the same and stable solutions for both angle and energy.

Compared with Euler's method, this is because their error per step is $O(h^3)$ or higher, $O(h^3)$ for midpoint method and Euler-trapezoidal method, $O(h^5)$ for RK4 method, and

the coefficients of the leading error term do not remain positive or negative all the time, thus the energy does not drift away in a fixed direction over time. Among these three methods, RK4 method should give the most accurate solution, as it has the highest order of accuracy.

2 Problem2: Radial Schrödinger Equation

2.1 problem description

Write a code to numerically solves radial Schrödinger equation for $[-\frac{1}{2}\nabla^2 + V(r)]R(r) = ER(r)$. Considering the following potentials:

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

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

where erf is the error function. And for element Li, you could set :

$Z_{ion} = 3; r_{loc} = 0.4; C_1 = -14.0093922; C_2 = 9.5099073; C_3 = -1.7532723; C_4 = 0.0834586$

Compute and plot the first three eigenstates.

2.2 algorithm description

First we transform $R(r) = \frac{u(r)}{r}$ so the radial Schrödinger equation becomes:

$$-\frac{1}{2}\frac{d^2}{dr^2}u(r) + V(r)u(r) = Eu(r) \tag{11}$$

Then we use finite difference method to discretize the radial Schrödinger equation. The radial Schrödinger equation can be written as:

$$-\frac{1}{2}\frac{d^2}{dr^2}u(r) + V(r)u(r) = Eu(r) \rightarrow -u_{i-1} + \left(2s^2V_i + 2\right) - u_{i+1} = 2s^2Eu_i \tag{12}$$

Here i is the value of the transformed wavefunction at the i-th grid point, V_i is the potential at the i-th grid point, s is the grid spacing,so in matrix form, we have:

$$HU_n = E_nU_n \tag{13}$$

Where H is a tridiagonal matrix, U_n is the eigenvector corresponding to the eigenvalue E_n. Here the boundary condition is $u(0) = 0$ and $u(r_{max}) = 0$. In my code I use `scipy.linalg.eigh_tridiagonal` to solve the eigenvalue of the tridiagonal matrix.

2.3 output

run `problem2.py`

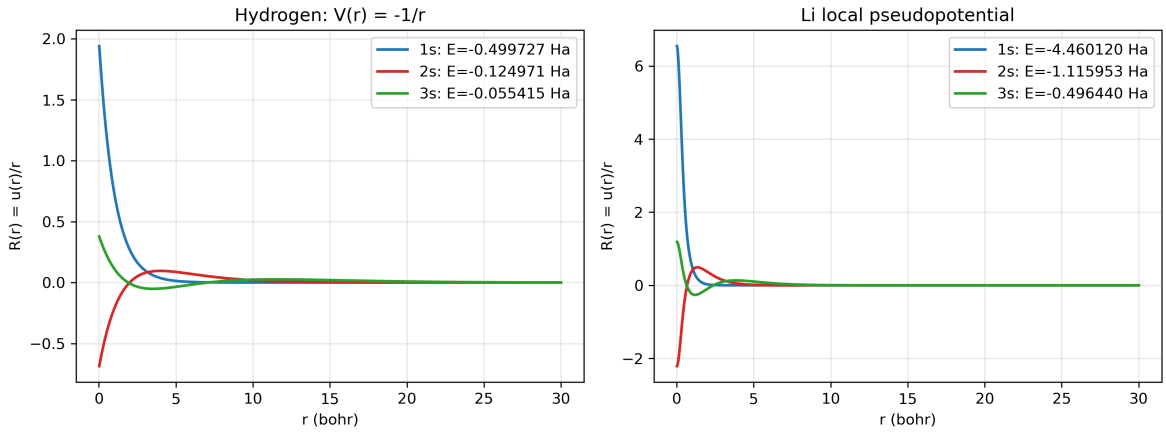

Figure 2: First three eigenstates (l=0)

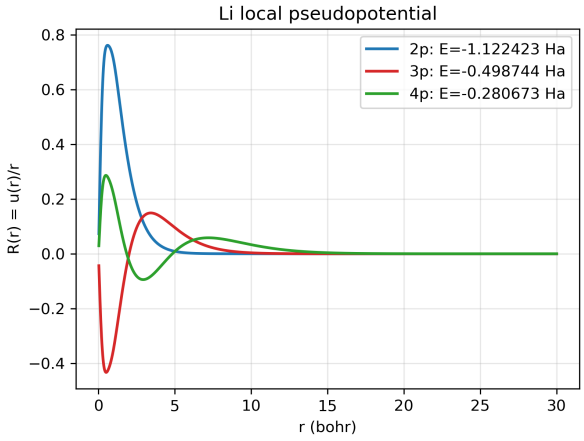

Figure 3: First three eigenstates of Li (l=1)

There is one point that need to be specified, for potential $V(r) = \frac{1}{r}$, its symmetry is higher than arbitrary spherically potential, because of the existence of conservation Runge-Lentz vector, so the eigenvalues only depend on the principal quantum number n.

For arbitrary spherically symmetric potential, the eigenvalues depend on both principal quantum number n and angular momentum quantum number l, we only know different angular momentum quantum number l has degenerate eigenvalues.

Here I choose the angular momentum quantum number $l = 0, 1$ for Li pseudo-potential, since different l has degenerate eigenvalues in spherically symmetric potential.

I set the radial distance range $r \in [0, 30]$ Bohr, with 1200 uniform sample points, the energy unit is Hartree.

```
Li local pseudopotential, l=0, first 3 states (Hartree):
1s: E = -4.46012035
2s: E = -1.11595330
3s: E = -0.49643966
```

Figure 4: energy output (l=0)

```
Li local pseudopotential, l=1, first 3 states (Hartree):
2p: E = -1.12242250
3p: E = -0.49874446
4p: E = -0.28067348
```

Figure 5: energy output (l=1)

So the first three eigenvalues for Li is 1s, 2p, 2s states.