単振り子および二重振り子の運動方程式に対する Runge-Kutta 法の適用

計数工学科 J4190147 小野悠太

2021 年 2 月 9 日

目次

1　目的

このレポートでは，二重振り子の運動方程式に対して Runge-Kutta 法を適用し数値解を求め，運動を可視化したうえで系の全エネルギーがどれほどの精度で保存されるのかを確かめることを目的とする．

このような目的を定めた理由としては，1A セメスターで受講した「振動波動論」で二重振り子のカオスが紹介されておりその複雑な運動に興味を持ったことや今セメスターの「数学 1D」で解析力学を習い，二重振り子の運動方程式を立てやすくなったことが挙げられる．また，カオスという複雑で非周期的な運動に対し Runge-Kutta 法を適用しても，期待される精度が出るのか気になったことも理由の一つである．

具体的な設定や実験方法については後の章で説明する．また，ここで扱う二重振り子のモデルはそれぞれの振り子の腕の先端に質点が存在するモデル（単振り子を連結したもの）とする．

2　単振り子の運動

この章では二重振り子ではなく単振り子を扱う．この運動はカオスではない．まず単振り子に対して Runge-Kutta 法を適用して系のエネルギーの変化を求めることで，カオスである場合とそうでない場合の比較を可能とすることを目的としている．
また，今回扱う二重振り子のモデルは単振り子を 2 つ直列に連結したものであるから，これらの結果に類似性が見られることも期待される．

2.1　単振り子の運動方程式

ここでは，振り子の腕の一方が原点 O に固定されており，他方の端点には質量 m_1 の質点が取り付けられてあるとする．
θ は鉛直下方向と振り子の腕がなす角とし，腕の長さを l とする．これは極めて一般的な振り子であり，その運動方程式は以下のようにあらわされる．

$$\ddot{\theta} = -\frac{g}{l}\sin\theta \tag{1}$$

$\dot{\theta} = \omega$ とするとこの運動方程式は，一階の微分方程式を連立させたものとなる．

$$\begin{cases} \dot{\theta} = \omega & \text{(2a)} \\ \dot{\omega} = -\dfrac{g}{l}\sin\theta & \text{(2b)} \end{cases}$$

2.2　単振り子の運動方程式に対する Runge-Kutta 法の適用

以下では式 (2) に対して Runge-Kutta 法を適用することを考える．

$$f(\theta) = -\frac{g}{l}\sin\theta \tag{3}$$

として，

$$k_1 = f(\theta^{(m)}) \tag{4}$$
$$n_1 = \omega^{(m)} \tag{5}$$
$$k_2 = f\left(\theta^{(m)} + \frac{n_1}{2}\Delta t\right) \tag{6}$$
$$n_2 = \omega^{(m)} + \frac{k_1}{2}\Delta t \tag{7}$$
$$k_3 = f\left(\theta^{(m)} + \frac{n_2}{2}\Delta t\right) \tag{8}$$
$$n_3 = \omega^{(m)} + \frac{k_2}{2}\Delta t \tag{9}$$
$$k_4 = f\left(\theta^{(m)} + n_3\Delta t\right) \tag{10}$$
$$n_4 = \omega^{(m)} + k_3\Delta t \tag{11}$$

と定義すると，

$$\omega^{(m+1)} = \omega^{(m)} + \left(\frac{1}{6}k_1 + \frac{1}{3}k_2 + \frac{1}{3}k_3 + \frac{1}{6}k_4\right) \tag{12}$$

$$\theta^{(m+1)} = \theta^{(m)} + \left(\frac{1}{6}n_1 + \frac{1}{3}n_2 + \frac{1}{3}n_3 + \frac{1}{6}n_4\right) \tag{13}$$

と表せる．

今回はこれを Python により実装し，運動の可視化およびそのエネルギー変化を可視化した．運動の可視化のソースコードは付録中のソースコード 1，エネルギーの時間変化のグラフを作成するソースコードは**??**，時間幅の取り方によるエネルギーの誤差の変化を可視化するソースコードは 2 に示す．

2.3　結果

2.3.1　単振り子の運動のアニメーション

2.3.2　エネルギー誤差の時間変化

2.3.3　時間幅の取り方による誤差の変化

3.4　考察

4　付録

4.1　二重振り子の運動方程式の導出

4.2 ソースコード

ソースコード 1　単振り子のアニメーション作成

```
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.animation as anim
from collections import deque
from datetime import datetime


# constants
g = 9.80665 # Standard gravity
dt_now = datetime.now()


# parameters
l1 = 1
m1 = 1
theta1 = np.pi/4
w1 = 0
t_start = 0
t_end = 10
steps = 1000


# calculated
dt = (t_end - t_start)/steps


# lists
theta_series = deque([theta1])
w_series = deque([w1])
xs = deque([l1*np.sin(theta1)])
ys = deque([-l1*np.cos(theta1)])
t_series = np.linspace(t_start, t_end, steps+1)
T_series = deque([m1*(l1**2)*(w1**2)/2])
U_series = deque([-m1*g*l1*np.cos(theta1)])
H_series = deque([m1*(l1**2)*(w1**2)/2-m1*g*l1*np.cos(theta1)])


fig, ax = plt.subplots()
```

```
ax.set_xlabel(R'$x\␣/m$', fontsize=14)
ax.set_ylabel(R'$y\␣/m$', fontsize=14)


def f(theta):
    return -g*np.sin(theta)/l1


def RungeKutta41(theta, w):
    k1 = f(theta) # w
    n1 = w # theta
    k2 = f(theta + n1*dt/2)
    n2 = w + k1*dt/2
    k3 = f(theta + n2*dt/2)
    n3 = w + k2*dt/2
    k4 = f(theta + n3*dt)
    n4 = w + k3*dt
    w = w + (k1/6 + k2/3 + k3/3 + k4/6)*dt
    theta = theta + (n1/6 + n2/3 + n3/3 + n4/6)*dt

    w_series.append(w)
    theta_series.append(theta)
    xs.append(l1*np.sin(theta))
    ys.append(-l1*np.cos(theta))
    T = m1*(l1**2)*(w**2)/2
    U = - m1*g*l1*np.cos(theta)
    H = T + U
    T_series.append(T)
    U_series.append(U)
    H_series.append(H)


for i in range(steps):
    theta = theta_series[-1]
    w = w_series[-1]
    RungeKutta41(theta, w)

images = []

for i in range(steps):
    x = [0, xs[i]]
    y = [0, ys[i]]
```

```
81      image = ax.plot(x, y, 'o-', lw=2, c="black", label="pendulum")
82      ax.grid(True)
83      ax.axis('equal')
84      ax.set_title("simple␣pendulum", fontsize=18)
85      images.append(image)
86
87  ani = anim.ArtistAnimation(fig, images, interval=10)
88  plt.tight_layout()
89
90  ani.save('./figure/RK41/animation/{0}-{1}-{2}-{3}{4}{5}.gif'.format(
91      dt_now.year, dt_now.month, dt_now.day,
92      dt_now.hour, dt_now.minute, dt_now.second), writer='pillow', fps=50)
93  # plt.show()
```

ソースコード 2　単振り子における RK4 の精度確認

```
import numpy as np
import matplotlib.pyplot as plt
from collections import deque
from datetime import datetime


# constants
g = 9.80665 # Standard gravity
dt_now = datetime.now()

# parameters
l1 = 1
m1 = 1
w0 = 0
t_start = 0
t_end = 10

# lists
T_series = deque([])
U_series = deque([])
H_series = deque([])


def f(theta):
    return -g*np.sin(theta)/l1


def RungeKutta41(theta, w, dt):
    k1 = f(theta) # w
    n1 = w # theta
    k2 = f(theta + n1*dt/2)
    n2 = w + k1*dt/2
    k3 = f(theta + n2*dt/2)
    n3 = w + k2*dt/2
    k4 = f(theta + n3*dt)
    n4 = w + k3*dt
    w = w + (k1/6 + k2/3 + k3/3 + k4/6)*dt
    theta = theta + (n1/6 + n2/3 + n3/3 + n4/6)*dt

    w_series.append(w)
    theta_series.append(theta)
```

```
    T = m1*(l1**2)*(w**2)/2
    U = - m1*g*l1*np.cos(theta)
    H = T + U
    T_series.append(T)
    U_series.append(U)
    H_series.append(H)


thetas = [np.pi/6, np.pi/4, np.pi/3, np.pi/2]
thetas_str = ["pi6", "pi4", "pi3", "pi2"]
thetas_tex = [R"\pi/6", R"\pi/4", R"\pi/3", R"\pi/2"]
dts = np.logspace(-4, 0, num=17, base=10.0)


for theta0, theta0_str, theta0_tex in zip(thetas, thetas_str, thetas_tex):
    fig, ax = plt.subplots()
    exact_H = m1*(l1**2)*(w0**2)/2 - m1*g*l1*np.cos(theta0)
    y = []
    for dt in dts:
        steps = int((t_end - t_start) / dt)
        theta_series = deque([theta0])
        w_series = deque([w0])
        for i in range(steps):
            theta = theta_series[-1]
            w = w_series[-1]
            RungeKutta41(theta, w, dt)
        y.append(np.abs(exact_H - H_series[-1]))
    ax.plot(dts, y, label=R"$\theta_0␣={}$".format(theta0_tex))
    ax.semilogx()
    ax.semilogy()
    ax.grid()
    ax.legend(fontsize="14")
    ax.set_xlabel(R"$␣\Delta␣t$", fontsize="14")
    ax.set_ylabel("Error", fontsize="14")
    fig.suptitle('Evaluation␣of␣RK4', fontsize="20")
    fig.savefig('./figure/RK41/evaluation1/{6}_{0}-{1}-{2}-{3}{4}{5}.jpeg'
                .format(dt_now.year, dt_now.month, dt_now.day, dt_now.hour,
                        dt_now.minute, dt_now.second, theta0_str))
```

4.3 計算機環境

4.4 参考文献