

2020

Pendulum

S.Matoi

目次

第 1 章	単振子	2
1.1	モデルの定式化	2
1.2	Python で模擬実験	3
第 2 章	2 重平面振子	5
2.1	モデルの定式化	5
2.2	Python による模擬実験	7
第 3 章	支点が水平方向に可動な単振子	9
3.1	モデルの定式化	9
3.2	Python による模擬実験	10
第 4 章	支点が円周上を周回している単振子	13
4.1	モデルの定式化	13
4.2	Python による模擬実験	16
第 5 章	支点が鉛直に振動している単振子	20
5.1	モデルの定式化	20
5.2	Python による模擬実験	21
第 6 章	支点が水平に振動している単振子	24
6.1	モデルの定式化	24
6.2	Python による模擬実験	25
謝辞		28
参考文献		29

第1章

単振り子

1.1 モデルの定式化

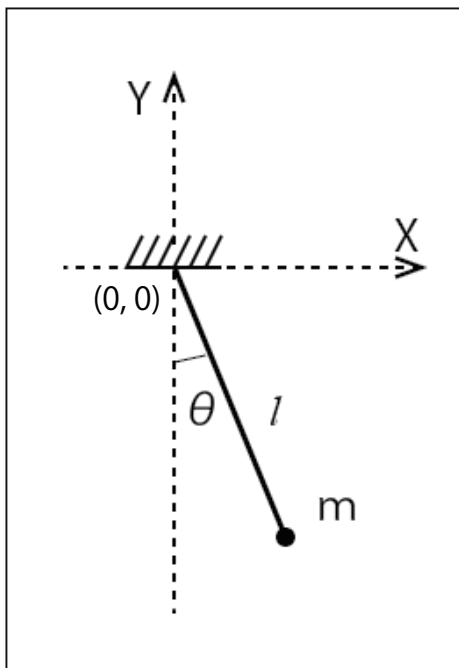


図 1.1 単振り子

一方の端が固定され、質量を無視できる長さ l の棒の他端に、質量 m の質点が拘束されている。棒が鉛直となす角は θ 、重力加速度は g である。

$$T = \frac{1}{2}m(l\dot{\theta})^2, \quad U = -mgl \cos \theta$$

より, Lagrangian $L = T - U$ は,

$$L = \frac{1}{2}ml^2\dot{\theta}^2 + mgl \cos \theta$$

Euler-Lagrange eq. は、

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}} - \frac{\partial L}{\partial \theta} = 0$$

により次の通り.

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

1.2 Python で模擬実験

これを次のように1階まで分解し, 連立にして関数 ode を定義する.

$$\frac{d\theta}{dt} = \dot{\theta}$$

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

常微分方程式 (Ordinary Differential Equation; ODE) の積分は, `scipy.integrate` の `odeint` を使う. 関数 `ode` の戻り値は, $[\theta, \dot{\theta}]$ である. Python のアニメーションには, `FuncAnimation` を使う.

ソースコード 1.1 単振り子

```

1  import matplotlib.pyplot as plt
2  import numpy as np
3  from numpy import sin, cos, pi
4  from matplotlib.animation import FuncAnimation
5  from scipy.integrate import odeint
6  #
7  # Constants
8  #
9  G = 9.8 # [m/s^2] acceleration of gravity
10 THETA0 = pi/4. # [rad] initial angle
11 V0 = 1. # [m/s] initial velocity
12 L = 1. # [m] length of the pendulum
13 DURATION = 10. # [s] duration time
14 INTERVAL = 0.05 # [s] interval time
15 #
16 # Differential Equation
17 #
18 def ode(f, t):
19     theta, dtheta = f
20     dfdt = [dtheta, -(G/L) * sin(theta)]
21     return dfdt
22 #
23 # Initial condition
24 #
25 f0 = [THETA0, V0/L] # [theta, v] at t = 0
26 t = np.arange(0, DURATION + INTERVAL, INTERVAL) # domain of definition
27 #
28 # Solve the equation
29 #
30 sol = odeint(ode, f0, t)

```

```

31 theta = sol[:, 0]
32 x = L * sin(theta)
33 y = - L * cos(theta)    # coordinates of the mass point
34 #
35 # Prepare the Screen to display
36 #
37 fig = plt.figure()
38 ax = fig.add_subplot(111, aspect='equal', autoscale_on=False,
39                      xlim=(-L, L), ylim=(-L, L))
40 ax.grid()
41 markers_on = [1]
42 line, = plt.plot([], [], 'ro-', markevery=markers_on, animated=True)
43 time_template = 'time = %.1fs'
44 time_text = ax.text(0.05, 0.9, '', transform=ax.transAxes)
45 #
46 # Animate the simulated results
47 #
48 def init():
49     time_text.set_text('')
50     return line, time_text
51
52 def update(i):
53     next_x = [0, x[i]]
54     next_y = [0, y[i]]
55     line.set_data(next_x, next_y)
56     time_text.set_text(time_template % (i*INTERVAL))
57     return line, time_text
58
59 FRAME_INTERVAL = 1000 * INTERVAL # [msec] interval between frames
60 ani = FuncAnimation(fig, update, frames=np.arange(0, len(t)),
61                    interval=FRAME_INTERVAL, init_func=init, blit=True)
62 #
63 # Show on the screen and Save the results
64 #
65 plt.show()
66 FPS = 1000/FRAME_INTERVAL # frames per second
67 #ani.save('single_pendulum.mp4', fps=FPS, extra_args=['-vcodec', 'libx264'])
68 ani.save('single_pendulum.gif', writer='imagemagick', fps=FPS)

```

第2章

2重平面振子

2.1 モデルの定式化

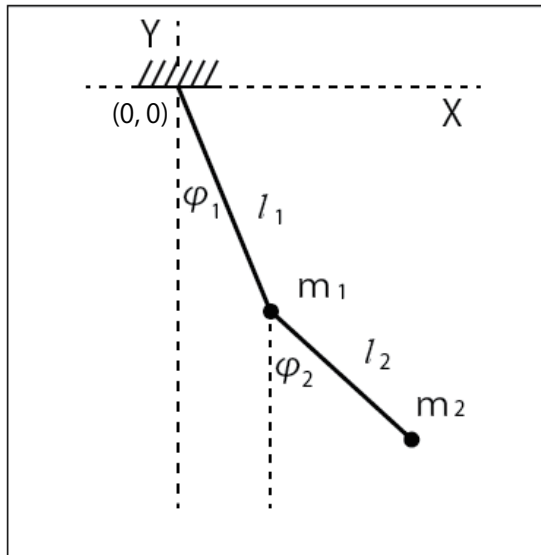


図 2.1 2重平面振子

棒 l_1 及び l_2 が鉛直となす角を φ_1, φ_2 とする. 座標原点を支点におくことにすると, 質点の座標はそれぞれ

$$(l_1 \sin \varphi_1, -l_1 \cos \varphi_1), \quad (l_1 \sin \varphi_1 + l_2 \sin \varphi_2, -l_1 \cos \varphi_1 - l_2 \cos \varphi_2)$$

質点 m_1 に対する運動エネルギー T_1 , 及びポテンシャル・エネルギー U_1 は,

$$T_1 = \frac{1}{2} m_1 (l_1 \dot{\varphi}_1)^2, \quad U_1 = -m_1 g l_1 \cos \varphi_1$$

($T_1 = \frac{1}{2} m_1 (\dot{x}_1^2 + \dot{y}_1^2)$ を計算しても, 同じ結果になる.)

一方質点 m_2 について, そのデカルト座標 (x_2, y_2) から,

$$\begin{aligned} x_2 &= l_1 \sin \varphi_1 + l_2 \sin \varphi_2 \quad , \quad y_2 = -(l_1 \cos \varphi_1 + l_2 \cos \varphi_2) \\ \dot{x}_2 &= l_1 \dot{\varphi}_1 \cos \varphi_1 + l_2 \dot{\varphi}_2 \cos \varphi_2 \quad , \quad \dot{y}_2 = l_1 \dot{\varphi}_1 \sin \varphi_1 + l_2 \dot{\varphi}_2 \sin \varphi_1 \\ \dot{x}_2^2 &= l_1^2 \dot{\varphi}_1^2 \cos^2 \varphi_1 + 2l_1 l_2 \dot{\varphi}_1 \dot{\varphi}_2 \cos \varphi_1 \cos \varphi_2 + l_2^2 \dot{\varphi}_2^2 \cos^2 \varphi_2 \quad , \\ \dot{y}_2^2 &= l_1^2 \dot{\varphi}_1^2 \sin^2 \varphi_1 + 2l_1 l_2 \dot{\varphi}_1 \dot{\varphi}_2 \sin \varphi_1 \sin \varphi_2 + l_2^2 \dot{\varphi}_2^2 \sin^2 \varphi_2 \end{aligned}$$

運動エネルギーとポテンシャル・エネルギーは,

$\cos(\alpha - \beta) = \cos \alpha \cos \beta + \sin \alpha \sin \beta$, また $\sin^2 \alpha + \cos^2 \alpha = 1$ だから

$$T_2 = \frac{1}{2} m_2 (\dot{x}_2^2 + \dot{y}_2^2) = \frac{m_2}{2} (l_1^2 \dot{\varphi}_1^2 + l_2^2 \dot{\varphi}_2^2 + 2l_1 l_2 \cos(\varphi_1 - \varphi_2) \dot{\varphi}_1 \dot{\varphi}_2)$$

$$U_2 = -m_2 g l_1 \cos \varphi_1 - m_2 g l_2 \cos \varphi_2$$

従って, Lagrangian $L = T_1 + T_2 - U_1 - U_2$ は,

$$\begin{aligned} L &= \frac{m_1 + m_2}{2} l_1^2 \dot{\varphi}_1^2 + \frac{m_2}{2} l_2^2 \dot{\varphi}_2^2 + m_2 l_1 l_2 \dot{\varphi}_1 \dot{\varphi}_2 \cos(\varphi_1 - \varphi_2) \\ &\quad + (m_1 + m_2) g l_1 \cos \varphi_1 + m_2 g l_2 \cos \varphi_2 \end{aligned}$$

になる.

$$\frac{\partial L}{\partial \varphi_1} = -m_2 l_1 l_2 \sin(\varphi_1 - \varphi_2) \dot{\varphi}_1 \dot{\varphi}_2 - (m_1 + m_2) g l_1 \sin \varphi_1 \quad ,$$

$$\frac{\partial L}{\partial \dot{\varphi}_1} = (m_1 + m_2) l_1^2 \dot{\varphi}_1 + m_2 l_1 l_2 \cos(\varphi_1 - \varphi_2) \dot{\varphi}_2 \quad ,$$

$$\frac{\partial L}{\partial \varphi_2} = m_2 l_1 l_2 \sin(\varphi_1 - \varphi_2) \dot{\varphi}_1 \dot{\varphi}_2 - m_2 g l_2 \sin \varphi_2 \quad ,$$

$$\frac{\partial L}{\partial \dot{\varphi}_2} = m_2 l_2^2 \dot{\varphi}_2 + m_2 l_1 l_2 \cos(\varphi_1 - \varphi_2) \dot{\varphi}_1$$

より, Euler-Lagrange eq.

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{\varphi}_i} - \frac{\partial L}{\partial \varphi_i} = 0$$

は,

$$(m_1 + m_2) l_1^2 \ddot{\varphi}_1 + m_2 l_1 l_2 \cos(\varphi_1 - \varphi_2) \ddot{\varphi}_2 + m_2 l_1 l_2 \sin(\varphi_1 - \varphi_2) \dot{\varphi}_1 \dot{\varphi}_2 + (m_1 + m_2) g l_1 \sin \varphi_1 = 0,$$

$$m_2 l_2^2 \ddot{\varphi}_2 + m_2 l_1 l_2 \cos(\varphi_1 - \varphi_2) \ddot{\varphi}_1 - m_2 l_1 l_2 \sin(\varphi_1 - \varphi_2) \dot{\varphi}_1 \dot{\varphi}_2 + m_2 g l_2 \sin \varphi_2 = 0$$

2.2 Python による模擬実験

連立の常微分方程式にして、関数 ode を定義する.

$$\frac{d\varphi_1}{dt} = \dot{\varphi}_1$$

$$\frac{d\varphi_2}{dt} = \dot{\varphi}_2$$

$$\frac{d\dot{\varphi}_1}{dt} = \frac{Mg \sin \varphi_1 + m_2 \sin H(l_1 \cos H + l_2)\dot{\varphi}_1\dot{\varphi}_2 - m_2g \cos H \sin \varphi_2}{-(Ml_1 - m_2l_1 \cos^2 H)}$$

$$\frac{d\dot{\varphi}_2}{dt} = \frac{Mg \cos H \sin \varphi_1 + \sin H(Ml_1 + m_2l_2 \cos H)\dot{\varphi}_1\dot{\varphi}_2 - Mg \sin \varphi_2}{Ml_2 - m_2l_2 \cos^2 H}$$

ここで, $H = \varphi_1 - \varphi_2$, $M = m_1 + m_2$ である.

単振子との違いは質点が2個あるので、関数 ode の戻り値は、単振子のときに $[\varphi, \dot{\varphi}]$ としていたのを、 $[\varphi_1, \dot{\varphi}_1, \varphi_2, \dot{\varphi}_2]$ とする.

積分は, `scipy.integrate` の `odeint` を、アニメーションは, `matplotlib.animation` の `FuncAnimation` を使う.

ソースコード 2.1 2重平面振子

```

1  from numpy import sin, cos, pi
2  import matplotlib.pyplot as plt
3  from matplotlib.animation import FuncAnimation
4  import numpy as np
5  from scipy.integrate import odeint
6  #
7  # Constants
8  #
9  G = 9.8                # [m/s^2] acceleration of gravity
10 PHI1_0 = pi            # [rad] initial theta
11 PHI2_0 = -pi/6.        # [rad] initial theta
12 V1_0 = 0.              # [m/s] initial velocity
13 V2_0 = 0.              # [m/s] initial velocity
14 L1 = 1.                 # [m] length of pendulum
15 L2 = 1.                 # [m] length of pendulum
16 M1 = 1.                 # [kg] mass
17 M2 = 0.4                # [kg] mass
18 DURATION = 15.          # [s] duration time
19 INTERVAL = 0.05         # [s] interval time
20 #
21 # Differential Equation
22 #
23 def ode(f, t):
24     phi1, dphi1, phi2, dphi2 = f
25     M, H = M1 + M2, phi1 - phi2
26     dphi1dt = (M*G*sin(phi1)\
27               + M2*(L2+L1*cos(H))*sin(H)*dphi1*dphi2\

```



```

28         - M2*G*cos(H)*sin(phi2))\
29         /(-M*L1+M2*L1*cos(H)*cos(H))
30     dphi2dt = (M*G*cos(H)*sin(phi1)
31               + (M2*L2*cos(H)+M*L1)*sin(H)*dphi1*dphi2\
32               - M*G*sin(phi2))\
33               / (M*L2-M2*L2*(cos(H))**2)
34     return [dphi1, dphi1dt, dphi2, dphi2dt]
35 #
36 # Initial Condition
37 #
38 f_0 = [PHI1_0, V1_0/L1, PHI2_0, V2_0/L2]
39 t = np.arange(0, DURATION + INTERVAL, INTERVAL)
40 #
41 # Solve the Equation
42 #
43 sol = odeint(ode, f_0, t)
44 phi1, phi2 = sol[:, 0], sol[:, 2]
45 x1 = L1 * sin(phi1)
46 y1 = - L1 * cos(phi1)
47 x2 = x1 + L2 * sin(phi2)
48 y2 = y1 - L2 * cos(phi2)
49 #
50 # Prepare the Screen to display
51 #
52 fig = plt.figure()
53 ax = fig.add_subplot(111, aspect='equal', autoscale_on=False,
54                     xlim = [-L1 - L2, L1 + L2], ylim = [-L1 - L2, L1 + L2])
55 ax.grid()
56 markers_on = [1,2]
57 line, = plt.plot([], [], 'ro-', markevery=markers_on, animated = True)
58 time_template = 'time = %.1fs'
59 time_text = ax.text(0.05, 0.9, '', transform=ax.transAxes)
60 #
61 # Simulate
62 #
63 def init():
64     time_text.set_text('')
65     return line, time_text
66
67 def update(i):
68     next_x = [0, x1[i], x2[i]]
69     next_y = [0, y1[i], y2[i]]
70     line.set_data(next_x, next_y)
71     time_text.set_text(time_template % (i * INTERVAL))
72     return line, time_text
73
74 FRAME_INTERVAL = 1000 * INTERVAL # [msec] interval time between frames
75 ani = FuncAnimation(fig, update, frames=np.arange(0, len(t)),
76                   interval=FRAME_INTERVAL, init_func=init, blit=True)
77 #
78 # Show and Save the results
79 #
80 plt.show()
81 FPS = 1000/FRAME_INTERVAL # frames per second
82 #ani.save('double_pendulum.mp4', fps=FPS, extra_args=['-vcodec', 'libx264'])
83 ani.save('double_pendulum.gif', writer='imagemagick', fps=FPS)

```

第3章

支点が水平方向に可動な単振子

3.1 モデルの定式化

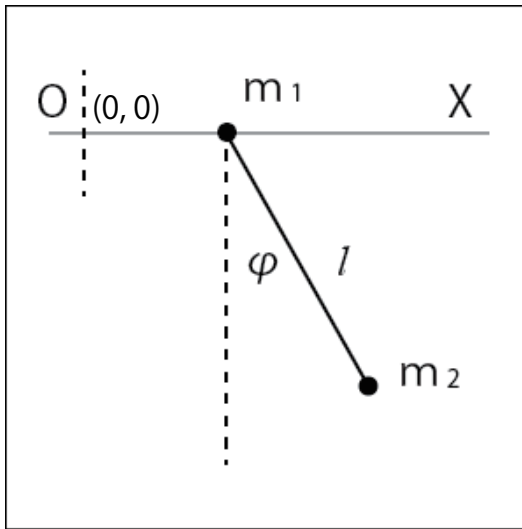


図 3.1 質量 m_1 の支点到質量 m_2 の単振子

質量 m_2 の単振子. その支点である質量 m_1 の質点が水平方向に運動できる.
水平に運動している質点 m_1 の運動エネルギーは、

$$T_1 = \frac{m_1}{2} \dot{x}^2$$

質点 m_2 のデカルト座標 (x_2, y_2) は、

$$\begin{aligned} x_2 &= x + l \sin \varphi, & y_2 &= -l \cos \varphi \\ \dot{x}_2 &= \dot{x} + l \dot{\varphi} \cos \varphi, & \dot{y}_2 &= l \dot{\varphi} \sin \varphi \\ \dot{x}_2^2 &= \dot{x}^2 + 2\dot{x}l\dot{\varphi} \cos \varphi + l^2 \dot{\varphi}^2 \cos^2 \varphi, & \dot{y}_2^2 &= l^2 \dot{\varphi}^2 \sin^2 \varphi \end{aligned}$$

質点 m_2 の運動エネルギーは、 $\sin^2 \alpha + \cos^2 \alpha = 1$ を思い出して、

$$T_2 = \frac{m_2}{2} (\dot{x}_2^2 + \dot{y}_2^2) = \frac{m_2}{2} (\dot{x}^2 + 2\dot{x}l\dot{\varphi} \cos \varphi + l^2 \dot{\varphi}^2)$$

一方、ポテンシャル・エネルギー U は、

$$U = -m_2 gl \cos \varphi$$

従って、Lagrangian $L = T_1 + T_2 - U$ は、

$$L = \frac{m_1 + m_2}{2} \dot{x}^2 + \frac{m_2}{2} (l^2 \dot{\varphi}^2 + 2l\dot{x}\dot{\varphi} \cos \varphi) + m_2 gl \cos \varphi$$

Euler-Lagrange eq. は、次の計算をして、

$$\frac{\partial L}{\partial \dot{\varphi}} = m_2 l^2 \dot{\varphi} + m_2 l \dot{x} \cos \varphi, \quad \frac{\partial L}{\partial \varphi} = -m_2 l \dot{x} \dot{\varphi} \sin \varphi - m_2 gl \sin \varphi$$

$$\frac{\partial L}{\partial \dot{x}} = (m_1 + m_2) \dot{x} + m_2 l \dot{\varphi} \cos \varphi, \quad \frac{\partial L}{\partial x} = 0$$

従って、

$$m_2 l^2 \ddot{\varphi} - m_2 l \dot{x} \sin \varphi = -m_2 l \dot{x} \dot{\varphi} \sin \varphi - m_2 gl \sin \varphi$$

$$\therefore \ddot{\varphi} = \frac{\sin \varphi}{l} \dot{x} + \frac{-\sin \varphi}{l} \dot{x} \dot{\varphi} + \frac{-g}{l} \sin \varphi$$

もう一方の式は、

$$(m_1 + m_2) \ddot{x} + m_2 l \dot{\varphi} \cos \varphi = 0$$

$$\therefore \ddot{x} = \frac{-m_2 l \dot{\varphi} \cos \varphi}{m_1 + m_2}$$

3.2 Python による模擬実験

これらを連立の一階微分方程式に直すと

$$\frac{d\varphi}{dt} = \dot{\varphi}$$

$$\frac{dx}{dt} = \dot{x}$$

$$\frac{d\dot{\varphi}}{dt} = \frac{\sin \varphi}{l} \dot{x} + \frac{-\sin \varphi}{l} \dot{x} \dot{\varphi} + \frac{-g}{l} \sin \varphi$$

$$\frac{d\dot{x}}{dt} = \frac{-m_2 l \dot{\varphi} \cos \varphi}{m_1 + m_2}$$

2つの質点の座標 $(x, y), (x_2, y_2)$ については、座標 x はラグランジアンに陽に含まれていないことから循環的な座標である。(ランダウ「力学」p.37) 循環座標が存在する場合の運動方程式の積分は簡単化できて、 x を含まない L を x で微分しても、その結果は定数(ゼロ)になる。

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{x}} = \frac{\partial L}{\partial x} = 0$$

時間微分 $\frac{d}{dt}$ の結果がゼロなので、 $\frac{\partial L}{\partial \dot{x}}$ は定数.

$$\frac{\partial L}{\partial \dot{x}} = (m_1 + m_2)\dot{x} + m_2 l \dot{\varphi} \cos \varphi = \text{const.}$$

これは系の運動量が保存することを表している. この式の積分は、 $(m_1 + m_2)x + m_2 l \sin \varphi = \text{const.}$, $\text{const} = 0$ に対して x が次の通り求まる. (ランダウ「力学」p.42 問3)

$$x = \frac{-m_2 l \sin \varphi}{m_1 + m_2}$$

$$y = 0$$

$$x_2 = x + l \sin \varphi$$

$$y_2 = -y - l \cos \varphi$$

支点の質量 m_1 を大きくして模擬実験を試すと、固定支点の状態に近づいてくる. ($m_1 \rightarrow \infty$ で $x \rightarrow 0$, これは通常の単振り子だ.)

ソースコード 3.1 支点が水平方向に可動な単振り子

```

1  from numpy import sin, cos, pi
2  import matplotlib.pyplot as plt
3  from matplotlib.animation import FuncAnimation
4  import numpy as np
5  from scipy.integrate import odeint
6  #
7  # Constants
8  #
9  L = 1          # [m]          length of the pendulum
10 PHIO = -pi/6   # [rad]        initial phi
11 V0 = 0.0       # [m/s]        initial velocity
12 X0 = 0         # [m]          initial location
13 G = 9.8        # [m/s^2]      acceleration of gravity
14 M1 = 1         # [kg]         mass
15 M2 = 1         # [kg]         mass
16 DURATION = 10  # [s]          duration time
17 INTERVAL = 0.05 # [s]          interval time
18 #
19 # Differential equation
20 #
21 def ode(f, t):
22     phi, dphi, x, dx = f
23     dpdt = sin(phi)/L*dx - sin(phi)/L*dx*dphi - G/L*sin(phi)
24     dxdt = -M2*L*dphi*cos(phi)/(M1+M2)
25     return [dphi, dpdt, dx, dxdt]
26 #
27 # Initial condition
28 #
29 f_0 = [PHIO, V0/L, X0, V0/L]
30 t = np.arange(0, DURATION + INTERVAL, INTERVAL)
31 #
32 # Solve the equation

```

```

33 #
34 sol = odeint(ode, f_0, t)
35 phi, x = sol[:, 0], sol[:, 2]
36 x1 = M2*L*sin(phi)/(M1+M2)
37 y1 = 0*t
38 x2 = x1 + L * sin(phi)
39 y2 = -y1 - L * cos(phi)
40 #
41 # Prepare the screen to display the results
42 #
43 fig = plt.figure()
44 ax = fig.add_subplot(111, aspect='equal', autoscale_on=False,\
45                     xlim = [-L - L, L + L], ylim = [-L - L, L + L])
46 ax.grid()
47 ax.set_title('M1={}, M2={}, L={}'.format(M1,M2,L))
48 line, = plt.plot([], [], 'ro-', animated = True)
49 time_template = 'time = %.1fs'
50 time_text = ax.text(0.05, 0.9, '', transform=ax.transAxes)
51 #
52 # Simulate with animation
53 #
54 def init():
55     time_text.set_text('')
56     return line, time_text
57
58 def update(i):
59     next_x = [x1[i], x2[i]]
60     next_y = [y1[i], y2[i]]
61     line.set_data(next_x, next_y)
62     time_text.set_text(time_template % (i * INTERVAL))
63     return line, time_text
64
65 FRAME_INTERVAL = 1000 * INTERVAL # [msec] interval time between frames
66 ani = FuncAnimation(fig, update, frames=np.arange(0, len(t)),
67                     interval=FRAME_INTERVAL, init_func=init, blit=True)
68 #
69 # Show and save the results
70 #
71 FPS = 1000/FRAME_INTERVAL # frames per second
72 plt.show()
73 #ani.save('pendulum3.mp4', fps=FPS, extra_args=['-vcodec', 'libx264'])
74 ani.save('pendulum3.gif', writer='imagemagick', fps=FPS)

```

第4章

支点が円周上を周回している単振子

4.1 モデルの定式化

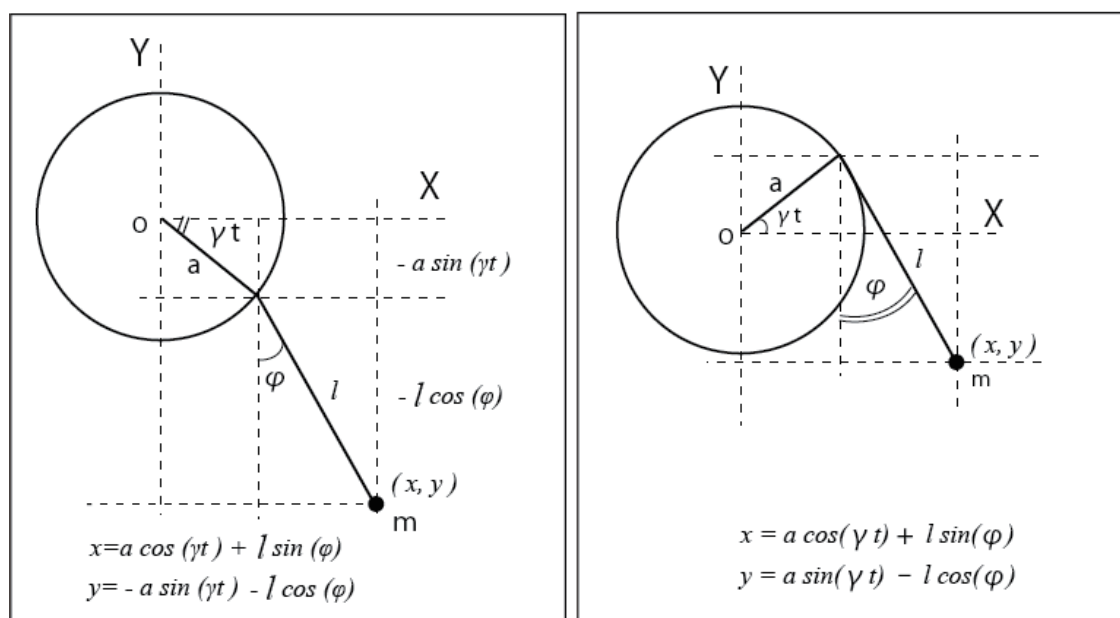


図 4.1 支点が円周上を周回する単振子

支点が鉛直平面内の円周上を一定の角速度 γ で一様に動いている単振子. ($\gamma < 0$ で CW, $\gamma > 0$ で CCW, γ が 0 なら支点は動かない)

円周上の点に注目して ($\sin(-\gamma t) = -\sin(\gamma t)$ だから) CW と CCW の違い

右の図は, CCW 方向に回転する場合のモデル化になっている. 質点 m の位置 (x, y) は,

$$x = a \cos \gamma t + l \sin \varphi$$

$$y = a \sin \gamma t - l \cos \varphi$$

質点の速度, その自乗を作って,

$$\begin{aligned}
\dot{x} &= -a\gamma \sin \gamma t + l\dot{\varphi} \cos \varphi \\
\dot{y} &= a\gamma \cos \gamma t + l\dot{\varphi} \sin \varphi \\
\dot{x}^2 &= a^2\gamma^2 \sin^2 \gamma t + l^2\dot{\varphi}^2 \cos^2 \varphi - 2al\gamma\dot{\varphi} \sin \gamma t \cos \varphi \\
\dot{y}^2 &= a^2\gamma^2 \cos^2 \gamma t + l^2\dot{\varphi}^2 \sin^2 \varphi + 2al\gamma\dot{\varphi} \sin \gamma t \cos \varphi
\end{aligned}$$

運動エネルギー T とポテンシャル・エネルギー $U = mgy$ は,

$$\begin{aligned}
T &= \frac{m}{2} (\dot{x}^2 + \dot{y}^2) \\
&= \frac{m}{2} \{a^2\gamma^2 + l^2\dot{\varphi}^2 + 2al\gamma\dot{\varphi} (\sin \varphi \cos \gamma t - \cos \varphi \sin \gamma t)\} \\
&= \frac{m}{2} \{a^2\gamma^2 + l^2\dot{\varphi}^2 + 2al\gamma\dot{\varphi} \sin(\varphi - \gamma t)\} \\
U &= mgy = mga \sin \gamma t - mgl \cos \varphi
\end{aligned}$$

Lagrangian $L = T - U$ は,

$$\begin{aligned}
L &= T - U \\
&= \frac{m}{2} a^2 \gamma^2 + \frac{m}{2} l^2 \dot{\varphi}^2 + alm\gamma\dot{\varphi} \sin(\varphi - \gamma t) - mga \sin \gamma t + mgl \cos \varphi \\
&= \frac{m}{2} l^2 \dot{\varphi}^2 + alm\gamma\dot{\varphi} \sin(\varphi - \gamma t) + mgl \cos \varphi + \left(\frac{m}{2} a^2 \gamma^2 - mga \sin \gamma t \right)
\end{aligned}$$

右辺の最終項は、次の関数 $f(t)$ の時間に関する完全導関数になっている。

$$f(t) = \frac{m}{2} a^2 \gamma^2 t + mga \frac{1}{\gamma} \cos \gamma t, \quad \frac{df(t)}{dt} = \frac{m}{2} a^2 \gamma^2 - mga \sin \gamma t$$

時間だけに依存する最終項を除けば, Lagrangian は最終的に次の様になる。(＊座標と時間の任意の関数 f の, 時間に関する完全導関数 \dot{f} を Lagrangian の中に含んでも, 作用積分によってその変分は消えてしまう項になるので, 取り除いても運動方程式としては変わらない. → ランダウ「力学」 p.5)

$$\therefore L = \frac{m}{2} l^2 \dot{\varphi}^2 + alm\gamma\dot{\varphi} \sin(\varphi - \gamma t) + mgl \cos \varphi$$

次の計算をして,

$$\begin{aligned}
\frac{\partial L}{\partial \dot{\varphi}} &= ml^2 \dot{\varphi} + alm\gamma \sin(\varphi - \gamma t) \\
\frac{\partial L}{\partial \varphi} &= alm\gamma\dot{\varphi} \cos(\varphi - \gamma t) - mgl \sin \varphi
\end{aligned}$$

もう少し準備して,

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{\varphi}} = ml^2 \ddot{\varphi} - alm\gamma^2 \cos(\varphi - \gamma t)$$

Euler-Langange eq.

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{\varphi}} - \frac{\partial L}{\partial \varphi} = 0$$

は,

$$\therefore \ddot{\varphi} = \frac{a\gamma^2}{l} \cos(\varphi - \gamma t) + \frac{a\gamma}{l} \dot{\varphi} \cos(\varphi - \gamma t) - \frac{g}{l} \sin \varphi$$

一方, 左の図は CW 方向に回転する場合のモデル化になっている。(実際, 右の図のモデルで γ を $-\gamma$ で置き換えても同じ事になる.) 質点 m の位置 (x, y) は,

$$\begin{aligned} x &= a \cos \gamma t + l \sin \varphi \\ y &= -a \sin \gamma t - l \cos \varphi \end{aligned}$$

質点の速度, その自乗と計算していくと,

$$\begin{aligned} \dot{x} &= -a\gamma \sin \gamma t + l\dot{\varphi} \cos \varphi \\ \dot{y} &= -a\gamma \cos \gamma t + l\dot{\varphi} \sin \varphi \\ \dot{x}^2 &= a^2\gamma^2 \sin^2 \gamma t + l^2\dot{\varphi}^2 \cos^2 \varphi - 2al\gamma\dot{\varphi} \sin \gamma t \cos \varphi \\ \dot{y}^2 &= a^2\gamma^2 \cos^2 \gamma t + l^2\dot{\varphi}^2 \sin^2 \varphi - 2al\gamma\dot{\varphi} \sin \varphi \cos \gamma t \end{aligned}$$

運動エネルギー T とポテンシャル・エネルギー U は,

$$\begin{aligned} T &= \frac{m}{2} (\dot{x}^2 + \dot{y}^2) \\ &= \frac{m}{2} \{a^2\gamma^2 + l^2\dot{\varphi}^2 - 2al\gamma\dot{\varphi} (\sin \varphi \cos \gamma t + \cos \varphi \sin \gamma t)\} \\ &= \frac{m}{2} \{a^2\gamma^2 + l^2\dot{\varphi}^2 - 2al\gamma\dot{\varphi} \sin(\varphi + \gamma t)\} \\ U &= mgy = -mga \sin \gamma t - mgl \cos \varphi \end{aligned}$$

Lagrangian $L = T - U$ は,

$$\begin{aligned} L &= T - U \\ &= \frac{m}{2} a^2 \gamma^2 + \frac{m}{2} l^2 \dot{\varphi}^2 - alm\gamma\dot{\varphi} \sin(\varphi + \gamma t) + mga \sin \gamma t + mgl \cos \varphi \\ &= \frac{m}{2} l^2 \dot{\varphi}^2 - alm\gamma\dot{\varphi} \sin(\varphi + \gamma t) + mgl \cos \varphi + \left(\frac{m}{2} a^2 \gamma^2 + mga \sin \gamma t \right) \end{aligned}$$

右辺の最終項は, 次の関数 $f(t)$ の時間に関する完全導関数になっている.

$$f(t) = \frac{m}{2} a^2 \gamma^2 t - mga \frac{1}{\gamma} \cos \gamma t, \quad \frac{df(t)}{dt} = \frac{m}{2} a^2 \gamma^2 + mga \sin \gamma t$$

(*座標と時間の任意の関数 f の, 時間に関する完全導関数 \dot{f} を Lagrangian の中に含んでも, 作用積分によってその変分は消えてしまう項になるので, 取り除いても運動方程式としては変わらない. → ランダウ「力学」p.5) 時間だけに依存する最終項を除けば, Lagrangian は最終的に次の様になる.

$$\therefore L = \frac{m}{2} l^2 \dot{\varphi}^2 - alm\gamma\dot{\varphi} \sin(\varphi + \gamma t) + mgl \cos \varphi$$

次の計算をして,

$$\begin{aligned}\frac{\partial L}{\partial \dot{\varphi}} &= ml^2 \dot{\varphi} - alm\gamma \sin(\varphi + \gamma t) \\ \frac{\partial L}{\partial \varphi} &= -alm\gamma \dot{\varphi} \cos(\varphi + \gamma t) - mgl \sin \varphi\end{aligned}$$

もう少し準備して,

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{\varphi}} = ml^2 \ddot{\varphi} - alm\gamma^2 \cos(\varphi + \gamma t)$$

Eular-Langange eq.

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{\varphi}} - \frac{\partial L}{\partial \varphi} = 0$$

は,

$$\therefore \ddot{\varphi} = \frac{a\gamma^2}{l} \cos(\varphi + \gamma t) - \frac{a\gamma}{l} \dot{\varphi} \cos(\varphi + \gamma t) - \frac{g}{l} \sin \varphi$$

この CW の式は, CCW の場合の Euler-Lgrange eq. の γ を $-\gamma$ で置き換えて得られる式と同じ格好をしている.

4.2 Python による模擬実験

連立の一階微分方程式に直して, 関数 ode 定義する. 右の図 (CCW モデル) の場合の Euler-Lagrange eq. からは,

$$\frac{d\varphi}{dt} = \dot{\varphi}$$

$$\frac{d\dot{\varphi}}{dt} = \frac{a\gamma^2}{l} \cos(\varphi - \gamma t) + \frac{a\gamma}{l} \dot{\varphi} \cos(\varphi - \gamma t) - \frac{g}{l} \sin \varphi$$

円周上の点の位置 (x_0, y_0) は,

$$x_0 = a \cos \gamma t \quad , \quad y_0 = a \sin \gamma t$$

であり, 質点の位置 (x, y) は,

$$x = x_0 + l \sin \varphi \quad , \quad y = y_0 - l \cos \varphi$$

である.

左の図 (CW モデル) の場合の Euler-Lagrange eq. からは,


```

33         - (G/L)*sin(phi)]
34     return dfdt
35 #
36 # Initial condition
37 #
38 f0 = [PHI0, VO/L]      # [theta, v] at t = 0
39 t = np.arange(0, DURATION + INTERVAL, INTERVAL)
40 #
41 # Solve the equation
42 #
43 sol = odeint(ode, f0, t)
44 phi = sol[:, 0]
45 if CCW:
46     # CCW at the right sided figure
47     circ_x = A * cos(GAMMA*t)
48     circ_y = A * sin(GAMMA*t)
49     x = circ_x + L * sin(phi)
50     y = circ_y - L * cos(phi)
51 else:
52     # CW at the left sided figure
53     circ_x = A * cos(GAMMA*t)
54     circ_y = - A * sin(GAMMA*t)
55     x = circ_x + L * sin(phi)
56     y = circ_y - L * cos(phi)
57 #
58 # Prepare the screen to display
59 #
60 fig = plt.figure()
61 ax = fig.add_subplot(111, aspect='equal', autoscale_on=False,\
62                     xlim=(-2*L, 2*L), ylim=(-2*L, 2*L))
63 ax.grid()
64 markers_on = [2]
65 line, = plt.plot([], [], 'ro-', markevery=markers_on, animated = True)
66 time_template = 'time = %.1fs'
67 time_text = ax.text(0.05, 0.9, '', transform=ax.transAxes)
68 #
69 # Simulate
70 #
71 def init():
72     time_text.set_text('')
73     return line, time_text
74
75 def update(i):
76     next_x = [0, circ_x[i], x[i]]
77     next_y = [0, circ_y[i], y[i]]
78     line.set_data(next_x, next_y)
79     time_text.set_text(time_template % (i*INTERVAL))
80     return line, time_text
81
82 FRAME_INTERVAL = 1000 * INTERVAL # [msec] interval between frames
83 ani = FuncAnimation(fig, update, frames=np.arange(0, len(t)),
84                   interval=FRAME_INTERVAL, init_func=init, blit=True)
85 #
86 # Show and save the results
87 #
88 plt.show()
89 FPS = 1000/FRAME_INTERVAL # frames per second
90 if CCW:
91     fname = 'pendulum4CCW'

```

```
92 | else:
93 |     fname = 'pendulum4CW'
94 |     #ani.save(fname+'.mp4', fps=FPS, extra_args=['-vcodec', 'libx264'])
95 |     ani.save(fname+'.gif', writer='imagemagick', fps=FPS)
```

第5章

支点が鉛直に振動している単振子

5.1 モデルの定式化

支点が鉛直方向に $a \cos \gamma t$ にしたがって振動している単振子

単振子の座標は、 $(l \sin \varphi, l \cos \varphi)$ なので、質点 m の座標は、 $x = l \sin \varphi, y = a \cos \gamma t + l \cos \varphi$
次の準備をしておいて

$$\begin{aligned}\dot{x} &= l\dot{\varphi} \cos \varphi, & \dot{y} &= -a\gamma \sin \gamma t - l\dot{\varphi} \sin \varphi \\ \dot{x}^2 &= l^2 \dot{\varphi}^2 \cos^2 \varphi, & \dot{y}^2 &= a^2 \gamma^2 \sin^2 \gamma t + 2a\gamma l \dot{\varphi} \sin \gamma t \sin \varphi + l^2 \dot{\varphi}^2 \sin^2 \varphi\end{aligned}$$

運動エネルギー T とポテンシャル・エネルギー U は次の通り ($\sin^2 \alpha + \cos^2 \alpha = 1$)

$$\begin{aligned}T &= \frac{m}{2} (\dot{x}^2 + \dot{y}^2) \\ &= \frac{m}{2} (l^2 \dot{\varphi}^2 (\sin^2 \varphi + \cos^2 \varphi) + a^2 \gamma^2 \sin^2 \gamma t + 2a\gamma l \dot{\varphi} \sin \gamma t \sin \varphi) \\ &= \frac{m}{2} l^2 \dot{\varphi}^2 + ma\gamma l \dot{\varphi} \sin \gamma t \sin \varphi + \frac{m}{2} a^2 \gamma^2 \sin^2 \gamma t \\ U &= -mgy \\ &= -mg(a \cos \gamma t + l \cos \varphi) \\ &= -mga \cos \gamma t - mgl \cos \varphi\end{aligned}$$

ラグランジアン $L = T - U$ を求める.

$$\begin{aligned}L &= T - U \\ &= \frac{m}{2} l^2 \dot{\varphi}^2 + ma\gamma l \dot{\varphi} \sin \gamma t \sin \varphi + \frac{m}{2} a^2 \gamma^2 \sin^2 \gamma t + mga \cos \gamma t + mgl \cos \varphi \\ &= \frac{m}{2} l^2 \dot{\varphi}^2 + \left(ma\gamma^2 l \cos \gamma t \cos \varphi - ma\gamma l \frac{d}{dt} (\cos \varphi \sin \gamma t) \right) \\ &\quad + \frac{m}{2} a^2 \gamma^2 \sin^2 \gamma t + mga \cos \gamma t + mgl \cos \varphi \\ &= \frac{m}{2} l^2 \dot{\varphi}^2 + ma\gamma^2 l \cos \gamma t \cos \varphi + mgl \cos \varphi \\ &\quad + \left(-ma\gamma l \frac{d}{dt} (\cos \varphi \sin \gamma t) + \frac{m}{2} a^2 \gamma^2 \sin^2 \gamma t + mga \cos \gamma t \right)\end{aligned}$$

ここで、次の関係を使っている.

$$\begin{aligned} -ma\gamma l \frac{d}{dt}(\cos \varphi \sin \gamma t) &= -ma\gamma l (-\dot{\varphi} \sin \varphi \sin \gamma t + \gamma \cos \varphi \cos \gamma t) \\ &= ma\gamma l \dot{\varphi} \sin \gamma t \sin \varphi - ma\gamma^2 l \cos \gamma t \cos \varphi \end{aligned}$$

更に、次の関数 $f(\varphi, t)$ を用意すると、

$$f(\varphi, t) = -ma\gamma l \cos \varphi \sin \gamma t + \frac{1}{4}ma^2\gamma^2 t - \frac{1}{8}ma^2\gamma \sin 2\gamma t + mga \frac{1}{\gamma} \sin \gamma t$$

ラグランジアン最後の括弧内の部分は、関数 $f(\varphi, t)$ の時間に関する完全導関数になっている.
($\cos 2\alpha = 1 - 2\sin^2 \alpha$)

$$\begin{aligned} \frac{df(\varphi, t)}{dt} &= -ma\gamma l(-\dot{\varphi} \sin \varphi \sin \gamma t + \gamma \cos \varphi \cos \gamma t) + \frac{1}{4}ma^2\gamma^2 - \frac{1}{8}ma^2\gamma 2\gamma \cos 2\gamma t + mga \frac{1}{\gamma} \gamma \cos \gamma t \\ &= -ma\gamma l \frac{d}{dt}(\cos \varphi \sin \gamma t) + \frac{1}{4}ma^2\gamma^2 - \frac{1}{4}ma^2\gamma^2(1 - 2\sin^2 \gamma t) + mga \cos \gamma t \\ &= -ma\gamma l \frac{d}{dt}(\cos \varphi \sin \gamma t) + \frac{1}{4}ma^2\gamma^2 2\sin^2 \gamma t + mga \cos \gamma t \\ &= -ma\gamma l \frac{d}{dt}(\cos \varphi \sin \gamma t) + \frac{m}{2}a^2\gamma^2 \sin^2 \gamma t + mga \cos \gamma t \end{aligned}$$

従って、最終的な Lagrangian は、(時間に関する完全導関数の部分を除いて)

$$L = \frac{ml^2}{2}\dot{\varphi}^2 + ma\gamma^2 l \cos \gamma t \cos \varphi + mgl \cos \varphi$$

次の計算から

$$\begin{aligned} \frac{\partial L}{\partial \dot{\varphi}} &= ml^2 \dot{\varphi} \\ \frac{\partial L}{\partial \varphi} &= -mla\gamma^2 \cos \gamma t \sin \varphi - mgl \sin \varphi \end{aligned}$$

Euler-Lagrange eq. は

$$\ddot{\varphi} = -\frac{a\gamma^2}{l} \cos \gamma t \sin \varphi - \frac{g}{l} \sin \varphi$$

5.2 Python による模擬実験

これを連立の一階微分方程式に直すと

$$\begin{aligned} \frac{d\varphi}{dt} &= \dot{\varphi} \\ \frac{d\dot{\varphi}}{dt} &= -\frac{a\gamma^2}{l} \cos \gamma t \sin \varphi - \frac{g}{l} \sin \varphi \end{aligned}$$

質点の座標は

$$x = l \sin \varphi \quad , \quad y = a \cos \gamma t + l \cos \varphi$$

ソースコード 5.1 支点が鉛直に振動している単振り子

```

1  import matplotlib.pyplot as plt
2  import numpy as np
3  from numpy import sin, cos, pi
4  from matplotlib.animation import FuncAnimation
5  from scipy.integrate import odeint
6  #
7  # Constants
8  #
9  A = 1.0          # [m]
10 F = 0.1          # [Hz]
11 GAMMA = 2*pi*F
12 G = 9.8          # [m/s^2] gravitational acceleration
13 THETA0 = pi/4    # [rad] initial angle
14 V0 = 0.05        # [m/s] initial velocity
15 L = 1            # [m] length of the pendulum
16 DURATION = 20    # [s] duration time
17 INTERVAL = 0.05  # [s] interval time
18 #
19 # Differential equation
20 #
21 def ode(f, t):
22     theta, dtheta = f
23     dfdt = [dtheta, -1*(A*GAMMA*GAMMA/L) * cos(GAMMA * t) * sin(theta) - (G/L) *
24             sin(theta)]
25     return dfdt
26 #
27 # Initial condition
28 #
29 f0 = [THETA0, V0/L] # [theta, v] at t = 0
30 t = np.arange(0, DURATION + INTERVAL, INTERVAL)
31 #
32 # Solve the equation
33 #
34 sol = odeint(ode, f0, t)
35 theta = sol[:, 0]
36 x = L * sin(theta)
37 y = - L * cos(theta) - A * cos(GAMMA*t)
38 #
39 # Prepare the screen to display the results
40 #
41 fig = plt.figure()
42 ax = fig.add_subplot(111, aspect='equal', autoscale_on=False, xlim=(-2*L, 2*L),
43                     ylim=(-2*L, 2*L))
44 ax.grid()
45 markers_on = [1]
46 line, = plt.plot([], [], 'r-o', markevery=markers_on, animated=True)
47 time_template = 'time = %.1fs'
48 time_text = ax.text(0.05, 0.9, '', transform=ax.transAxes)
49 #
50 # Simulate the results
51 #
52 def init():

```

```
51     time_text.set_text('')
52     return line, time_text
53
54 def update(i):
55     circ_x = 0*t
56     circ_y = -A * cos(GAMMA*t)
57     next_x = [circ_x[i], x[i]]
58     next_y = [circ_y[i], y[i]]
59     line.set_data(next_x, next_y)
60     time_text.set_text(time_template % (i*INTERVAL))
61     return line, time_text
62
63 FRAME_INTERVAL = 1000 * INTERVAL # [msec] interval between frames
64
65 ani = FuncAnimation(fig, update, frames=np.arange(0, len(t)),
66                     interval=FRAME_INTERVAL, init_func=init, blit=True)
67
68 #
69 # Show and save the results
70 #
71 plt.show()
72 FPS = 1000/FRAME_INTERVAL # frames per second
73 #ani.save('pendulum5.mp4', fps=FPS, extra_args=['-vcodec', 'libx264'])
74 # using the Pillow instead of the imagemagic for MovieWriter
75 ani.save('pendulum5.gif', writer='Pillow', fps=FPS)
```


第6章

支点が水平に振動している単振子

6.1 モデルの定式化

単振子の座標は、 $(l \sin \varphi, l \cos \varphi)$ なので、質点 m の座標は、

$$x = a \cos \gamma t + l \sin \varphi, y = l \cos \varphi$$

予め次の計算の準備をしておいて、

$$\begin{aligned} \dot{x} &= -a\gamma \sin \gamma t + l\dot{\varphi} \cos \varphi, \quad \dot{y} = -l\dot{\varphi} \sin \varphi \\ \dot{x}^2 &= a^2\gamma^2 \sin^2 \gamma t - 2a\gamma l\dot{\varphi} \sin \gamma t \cos \varphi + l^2\dot{\varphi}^2 \cos^2 \varphi, \quad \dot{y}^2 = l^2\dot{\varphi}^2 \sin^2 \varphi \end{aligned}$$

運動エネルギー T とポテンシャル・エネルギー U は次のようになる ($\sin^2 \alpha + \cos^2 \alpha = 1$ を使って)

$$\begin{aligned} T &= \frac{m}{2}(\dot{x}^2 + \dot{y}^2) \\ &= \frac{m}{2}l^2\dot{\varphi}^2 (\sin^2 \varphi + \cos^2 \varphi) - ma\gamma l\dot{\varphi} \sin \gamma t \cos \varphi + \frac{m}{2}a^2\gamma^2 \sin^2 \gamma t \\ &= \frac{m}{2}l^2\dot{\varphi}^2 - ma\gamma l\dot{\varphi} \sin \gamma t \cos \varphi + \frac{m}{2}a^2\gamma^2 \sin^2 \gamma t \\ U &= -mgy = -mgl \cos \varphi \end{aligned}$$

ラグランジアン $L = T - U$ を求めていく.

$$\begin{aligned} L &= T - U \\ &= \frac{m}{2}l^2\dot{\varphi}^2 - ma\gamma l\dot{\varphi} \sin \gamma t \cos \varphi + \frac{m}{2}a^2\gamma^2 \sin^2 \gamma t + mgl \cos \varphi \\ &= \frac{m}{2}l^2\dot{\varphi}^2 + \left(ma\gamma^2 l \cos \gamma t \sin \varphi - ma\gamma l \frac{d}{dt}(\sin \varphi \sin \gamma t) \right) + \\ &\quad \frac{m}{2}a^2\gamma^2 \sin^2 \gamma t + mgl \cos \varphi \\ &= \frac{m}{2}l^2\dot{\varphi}^2 + ma\gamma^2 l \cos \gamma t \sin \varphi + \\ &\quad \left(-ma\gamma l \frac{d}{dt}(\sin \varphi \sin \gamma t) \frac{m}{2}a^2\gamma^2 \sin^2 \gamma t \right) + mgl \cos \varphi \end{aligned}$$

ここで、次の関係を使っている.

$$\begin{aligned}
-ma\gamma l \frac{d}{dt}(\sin \varphi \sin \gamma t) &= -ma\gamma l (\gamma \sin \varphi \cos \gamma t + \dot{\varphi} \cos \varphi \sin \gamma t) \\
&= -ma\gamma^2 l \sin \varphi \cos \gamma t - ma\gamma l \dot{\varphi} \cos \varphi \sin \gamma t
\end{aligned}$$

更に、次の関数 $f(\varphi, t)$ を用意する、

$$f(\varphi, t) = -ma\gamma l \sin \varphi \sin \gamma t + \frac{1}{4}ma^2\gamma^2 t - \frac{1}{8}ma^2\gamma \sin 2\gamma t$$

ラグランジアン最後の括弧内の部分は、関数 $f(\varphi, t)$ の時間に関する完全導関数になっている。
($\cos 2\alpha = 1 - 2\sin^2 \alpha$ を使って)

$$\begin{aligned}
\frac{df(\varphi, t)}{dt} &= -ma\gamma l \frac{d}{dt}(\sin \varphi \sin \gamma t) + \frac{1}{4}ma^2\gamma^2 - \frac{1}{8}ma^2\gamma(2\gamma) \cos 2\gamma t \\
&= -ma\gamma l \frac{d}{dt}(\sin \varphi \sin \gamma t) + \frac{1}{4}ma^2\gamma^2 - \frac{1}{4}ma^2\gamma^2(1 - 2\sin^2 \gamma t) \\
&= -ma\gamma l \frac{d}{dt}(\sin \varphi \sin \gamma t) + \frac{1}{4}ma^2\gamma^2(2\sin^2 \gamma t) \\
&= -ma\gamma l \frac{d}{dt}(\sin \varphi \sin \gamma t) + \frac{m}{2}a^2\gamma^2 \sin^2 \gamma t
\end{aligned}$$

従って、最終的な Lagrangian は、(時間に関する完全導関数を除いて)

$$L = \frac{ml^2}{2}\dot{\varphi}^2 + mla\gamma^2 \cos \gamma t \sin \varphi + mgl \cos \varphi$$

次の計算から、

$$\begin{aligned}
\frac{\partial L}{\partial \dot{\varphi}} &= ml^2 \dot{\varphi} \\
\frac{\partial L}{\partial \varphi} &= mla\gamma^2 \cos \gamma t \cos \varphi - mgl \sin \varphi
\end{aligned}$$

Euler-Lagrange eq. は

$$\ddot{\varphi} = \frac{a\gamma^2}{l} \cos \gamma t \cos \varphi - \frac{g}{l} \sin \varphi$$

6.2 Python による模擬実験

これを連立の一階微分方程式に直すと

$$\begin{aligned}
\frac{d\varphi}{dt} &= \dot{\varphi} \\
\frac{d\dot{\varphi}}{dt} &= \frac{a\gamma^2}{l} \cos \gamma t \cos \varphi - \frac{g}{l} \sin \varphi
\end{aligned}$$

質点の座標は

$$x = a \cos \gamma t + l \sin \varphi \quad , \quad y = l \cos \varphi$$

ソースコード 6.1 支点が水平に振動している単振子

```

1  import matplotlib.pyplot as plt
2  import numpy as np
3  from numpy import sin, cos, pi
4  from matplotlib.animation import FuncAnimation
5  from scipy.integrate import odeint
6
7  #
8  # Constants
9  #
10 A = 1.0 # [m]
11 F = 0.1 # [Hz]
12 GAMMA = 2*pi*F
13 G = 9.8 # [m/s^2] gravitational acceleration
14 THETA0 = pi/4 # [rad] initial angle
15 V0 = 0.05 # [m/s] initial velocity
16 L = 1 # [m] length of the pendulum
17 DURATION = 20 # [s] duration time
18 INTERVAL = 0.05 # [s] interval time
19 #
20 # Differential equation
21 #
22 def ode(f, t):
23     theta, dtheta = f
24     dfdt = [dtheta, (A*GAMMA*GAMMA/L) * cos(GAMMA * t) * cos(theta) - (G/L) * sin(
25         theta)]
26     return dfdt
27 #
28 # Initial condition
29 #
30 f0 = [THETA0, V0/L] # [theta, v] at t = 0
31 t = np.arange(0, DURATION + INTERVAL, INTERVAL)
32 #
33 # Solve the equation
34 #
35 sol = odeint(ode, f0, t)
36 theta = sol[:, 0]
37 x = L * sin(theta) + A * cos(GAMMA*t)
38 y = -L * cos(theta)
39 #
40 # Prepare the screen to display the results
41 #
42 fig = plt.figure()
43 ax = fig.add_subplot(111, aspect='equal', autoscale_on=False, xlim=(-2*L, 2*L),
44     ylim=(-2*L, 2*L))
45 ax.grid()
46 markers_on = [1]
47 line, = plt.plot([], [], 'ro-', markevery=markers_on, animated = True)
48 time_template = 'time = %.1fs'
49 time_text = ax.text(0.05, 0.9, '', transform=ax.transAxes)
50 #
51 # Simulate with animation
52 #
53 def init():
54     time_text.set_text('')
55     return line, time_text

```

```
55 def update(i):
56     circ_x = A * cos(GAMMA*t)
57     circ_y = 0*t
58     next_x = [circ_x[i], x[i]]
59     next_y = [circ_y[i], y[i]]
60     line.set_data(next_x, next_y)
61     time_text.set_text(time_template % (i*INTERVAL))
62     return line, time_text
63
64 FRAME_INTERVAL = 1000 * INTERVAL # [msec] interval between frames
65
66 ani = FuncAnimation(fig, update, frames=np.arange(0, len(t)),
67                     interval=FRAME_INTERVAL, init_func=init, blit=True)
68 #
69 # Show and save the results
70 #
71 FPS = 1000/FRAME_INTERVAL # frames per second
72 plt.show()
73 #ani.save('pendulum4.mp4', fps=FPS, extra_args=['-vcodec', 'libx264'])
74 # using Pillow instead of imagemagic for MovieWriter
75 ani.save('pendulum4.gif', writer='pillow', fps=FPS)
```

謝辞

参考文献

[1]