

# 計算機実習課題 問題 6.14 標準写像

早稲田大学先進理工学部物理学科 B4 藤本将太郎

2014/05/20

## 1 シミュレーションの目的

標準写像

$$\theta_{n+1} = (\theta_n + p_n) \quad \text{modulo } 2\pi \quad (1)$$

$$p_{n+1} = p_n + k \sin \theta_{n+1} \quad (2)$$

は、ハミルトン系の中でも、ある初期状態では規則的で可積分であるが他の条件ではカオス的になるという、混合系ともいえるクラスの単純なモデルである。物理的な実体としては、図 1 に示されているような回転子を考えている。慣性モーメント  $I$ 、長さ  $L$  の棒の一端が摩擦のない支点に固定されていて、他端には強さ  $k/L$  の周期的な撃力が、時刻  $t = 0, \tau, 2\tau, \dots$  に鉛直方向に加えられる。回転子の運動は角度  $\theta$  とそれに対応する角運動量  $p_\theta$  で記述される。この系のハミルトニアンは

$$H(\theta, p_\theta, t) = \frac{p_\theta^2}{2I} + k \cos \theta \sum_n \delta(t - n\tau) \quad (3)$$

と書かれる。したがって、対応する運動方程式は、正準方程式を用いて

$$\frac{dp_\theta}{dt} = k \sin \theta \sum_n \delta(t - n\tau) \quad (4)$$

$$\frac{d\theta}{dt} = \frac{p_\theta}{I} \quad (5)$$

となる。式 (4)、(5) から、 $p_\theta$  は撃力とつぎの撃力の間では一定 (重力はないと仮定している) であるが、撃力が加わるときに不連続的に変化することがわかる。角度  $\theta$  は撃力と撃力の間で  $t$  の 1 次で変化し、撃力が加わる瞬間では連続である。撃力を加えた直後の  $\theta$  と  $p_\theta$  の値を求めると都合がよい。 $\varepsilon > 0$  を微小量として  $t = n\tau + \varepsilon$  における  $\theta(t)$  および  $p_\theta(t)$  の値をそれぞれ  $\theta_n$  および  $p_n$  とする。式 (4) を  $t = (n+1)\tau - \varepsilon$  から  $t = (n+1)\tau + \varepsilon$  まで積分すると、

$$p_{n+1} - p_n = k \sin \theta_{n+1} \quad (6)$$

を得る ( $p$  は撃力と撃力の間で一定であり、 $t = (n+1)\tau$  のときにのみデルタ関数が積分に寄与することに注意が必要である)。式 (5) からは

$$\theta_{n+1} - \theta_n = (\tau/I)p_n \quad (7)$$

が得られる。 $\tau/I = 1$  となるように単位を選ぶと、標準写像 (1),(2) が得られる。式 (1) には角度  $\theta$  の値が 0 と  $2\pi$  の間に制限されるという条件 (modulo  $2\pi$ ) を付け加えてある。

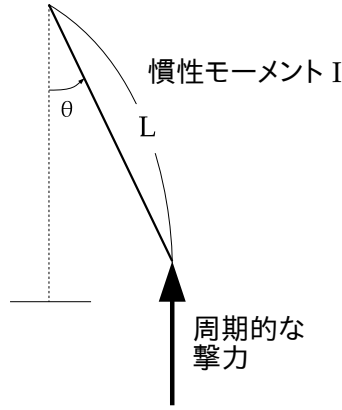


図1 撃力が加えられた慣性モーメント  $I$  の棒からなる回転子。重力と支点での摩擦は無視する。

式 (1),(2) の反復計算を行う前に、これらがハミルトン系、つまり  $n$  を大きくしていても  $q$ - $p$  空間における面積が一定であることを確かめておく (ここでは  $q$  は  $\theta$  を指している)。長さ  $dq_n$  と  $dp_n$  の長方形から始めることを考えると、1 回の反復の後、この長方形は  $dq_{n+1}$  と  $dp_{n+1}$  の辺を持つ平行四辺形に変形される。すなわち、

$$dq_{n+1} = dq_n + dp_n \quad (8)$$

$$dp_{n+1} = dp_n + k \cos q_{n+1} dq_{n+1} \quad (9)$$

である。式 (8) に (9) を代入すると、

$$dp_{n+1} = (1 + k \cos q_{n+1}) dp_n + k \cos q_{n+1} dq_n \quad (10)$$

を得る。ベクトル  $d\mathbf{q}_{n+1} = (dq_n, dp_n)$  と  $d\mathbf{p}_{n+1} = (1 + k \cos q_n, k \cos q_n dq_n)$  のベクトル積の大きさを計算して平行四辺形の面積を求めると、 $dq_n dp_n$  となり、位相空間内の面積は不変であり、標準写像は保存系であることがわかる。標準写像の定性的性質については問題 6.16 で調べることであるが、それらは次のように要約することができる。 $k = 0$  のとき、棒は運動量  $p_n = p_0 = \text{一定}$  によって定まる一定の角運動量で回転する。 $p_0$  が  $2\pi$  の有理数倍のときには、位相空間の軌跡は、水平な線上にある孤立した点列から構成される (共鳴トーラス)。 $p_0$  が  $2\pi$  の有理数倍でないか、コンピュータに十分な精度がないときには、十分に長い時間の後に軌跡は位相空間内の水平な曲線へと変化する。そして共有トーラスに対応する孤立した点列は閉曲線へと変化する。初期条件によっては、過渡的な振る舞いが消えるくらいに十分長い反復の後に、軌跡はカオス的になるだろう。

## 2 作成したプログラム

本シミュレーションで作成・使用したプログラムを以下に示す。

### 2.1 パラメータの設定ウィンドウ (SetParameter.py)

問題 6.14 で使用したものと同一である。

```
1  #!/usr/bin/env python
2  # -*- coding:utf-8 -*-
3  #
4  # written by Shotaro Fujimoto, Mgy 2014.
5
6  from Tkinter import *
7
8  class SetParameter():
9
10     def show_setting_window(self, parameters, commands):
11         """ Show a parameter setting window.
12
13         parameters: A list of dictionaries {'parameter name': default_value}
14         commands: A list of dictionary {'name of button': command}
15         """
16         self.root = Tk()
17         self.root.title('Parameter')
18
19         frame1 = Frame(self.root, padx=5, pady=5)
20         frame1.pack(side='top')
21         self.entry = []
22         for i, parameter in enumerate(parameters):
23             label = Label(frame1, text=parameter.items()[0][0] + ' = ')
24             label.grid(row=i, column=0, sticky=E)
25             self.entry.append(Entry(frame1, width=10))
26             self.entry[i].grid(row=i, column=1)
27             self.entry[i].delete(0, END)
28             self.entry[i].insert(0, parameter.items()[0][1])
29         self.entry[0].focus_set()
30
31         frame2 = Frame(self.root, padx=5, pady=5)
32         frame2.pack(side='bottom')
```

```

33         self.button = []
34         for i, command in enumerate(commands):
35             self.button.append(Button(frame2, text=command.items()[0][0],
36                                     command=command.items()[0][1]
37                                     )
38                                     )
39             self.button[i].grid(row=0, column=i)
40
41         self.root.mainloop()
42
43     def quit(self):
44         self.root.destroy()
45

```

## 2.2 標準写像の計算・描画プログラム (6-16\_standard\_map.py)

このプログラムでは、初めの `ntransient` 回の計算結果を捨てて、`nplot` 回の反復計算の結果をグラフに表示するプログラムである。プログラムを実行すると、`SetParameter` の `show_setting_window` が呼び出され、パラメータの設定が行える。OK ボタンを押すと、関数 `static` を介してパラメータの代入とグラフの描画を行う。ここでのカウンタは `scatter` メソッドが標準で色のループを行わないため、自前で色の制御を行うために追加したものである。実際に写像の計算を行っているのは 50 行目から 63 行目であり、あとはグラフの体裁を整えるためのものである。

```

1  #!/usr/bin/env python
2  # -*- coding:utf-8 -*-
3  #
4  # written by Shotaro Fujimoto, May 2014.
5
6  from math import sin, pi
7  from SetParameter import SetParameter
8  from array import array
9  import matplotlib.pyplot as plt
10 import numpy as np
11 import sys
12
13 # --- init ---
14 counter = 0
15 colors = ['blue', 'green', 'red', 'purple', 'black']
16 ntransient = 1000
17 nplot = 10000

```

```

18 nmax = ntransient + nplot
19 max_p = 0
20 min_p = 0
21
22 # --- prepare figure and axes ---
23 fig = plt.figure(figsize=(12, 6), dpi=80)
24 ax = fig.add_subplot(111)
25 plt.subplots_adjust(bottom=0.14)
26 plt.subplots_adjust(right=0.68)
27 ax.grid()
28
29 def static():
30     global counter
31     assignment()
32
33     color = colors[counter]
34     counter += 1
35     if counter >= len(colors):
36         counter = counter - len(colors)
37
38     plot_static(theta, ang_mom, color)
39
40 def assignment():
41     """ Assign the values to variables and plot the map.
42     """
43     global theta, ang_mom, k, color
44     theta = float(window.entry[0].get())
45     ang_mom = float(window.entry[1].get())
46     k = float(window.entry[2].get())
47
48 def plot_static(q, p, color):
49     """ Show an standard map.
50     """
51     global max_p, min_p
52     # --- Caluculating ---
53     ary_q = array('d')
54     ary_p = array('d')
55     ary_q.append(q)
56     ary_p.append(p)
57

```

```

58     for i in range(nmax+1):
59         _q = ary_q[i] + ary_p[i]
60         # 0 <= theta <= 2*pi
61         while _q > 2*pi:
62             _q = _q - 2*pi
63         while _q < 0.:
64             _q = _q + 2*pi
65         ary_q.append(_q)
66         ary_p.append(ary_p[i] + k*sin(ary_q[i] + ary_p[i]))
67
68     # --- Set labels and ticks. ---
69     plt.xlabel(r'$\theta$', fontsize=16)
70     plt.ylabel(r'$p_{\theta}$', fontsize=16)
71     plt.title('Standard map')
72     unit = 0.5
73     q_tick = np.arange(0, 2.0+unit, unit)
74     q_label = [r"$0$", r"$\frac{1}{2}\pi$", r"$\pi$", r"$\frac{3}{2}\pi$", r"$2\pi$"]
75     ax.set_xticks(q_tick*pi)
76     ax.set_xticklabels(q_label, fontsize=18)
77     if max(ary_p[ntransient:]) > max_p:
78         max_p = max(ary_p[ntransient:])
79     if min(ary_p[ntransient:]) < min_p:
80         min_p = min(ary_p[ntransient:])
81     margin = (max_p-min_p)*0.05
82     plt.ylim(min_p-margin, max_p+margin)
83
84     # --- Plot the standard map. ---
85     plt.scatter(ary_q[ntransient:], ary_p[ntransient:],
86                 s=2.0, marker='o', color=color,
87                 label=r'$\theta_{0}$=' + str(theta) + ' : '
88                 + r'$p_{0}$=' + str(ang_mom) + ' : '
89                 + r'$k$=' + repr(k)
90                 )
91     ax.legend(bbox_to_anchor=(1.05, 1.0), loc='upper left', borderaxespad=0)
92
93     plt.show()
94
95 if __name__ == '__main__':
96
97     window = SetParameter()

```

```

98     default_params = [{'theta':1.5}, {'p':0.2}, {'k':0.1}]
99     commands = [{'OK':static}, {'EXIT':sys.exit}]
100     window.show_setting_window(default_params, commands)
101

```

### 3 実習課題

- a. 標準写像の反復を実行するプログラムを書いて、位相空間の軌跡を描け。 $k$  を変えたときの挙動を調べよ。特に、位相空間内の軌道の変化に着目せよ。

まず、単純な問題として  $k = 0$  として固定したとき、初期条件  $p_0$  の値をさまざまに変えたときの軌跡の様子を図 2 に示す。この図は、周期的な撃力がないときには、棒は初めの角運動量を保ったまま回転を続けるという事実を再現している。

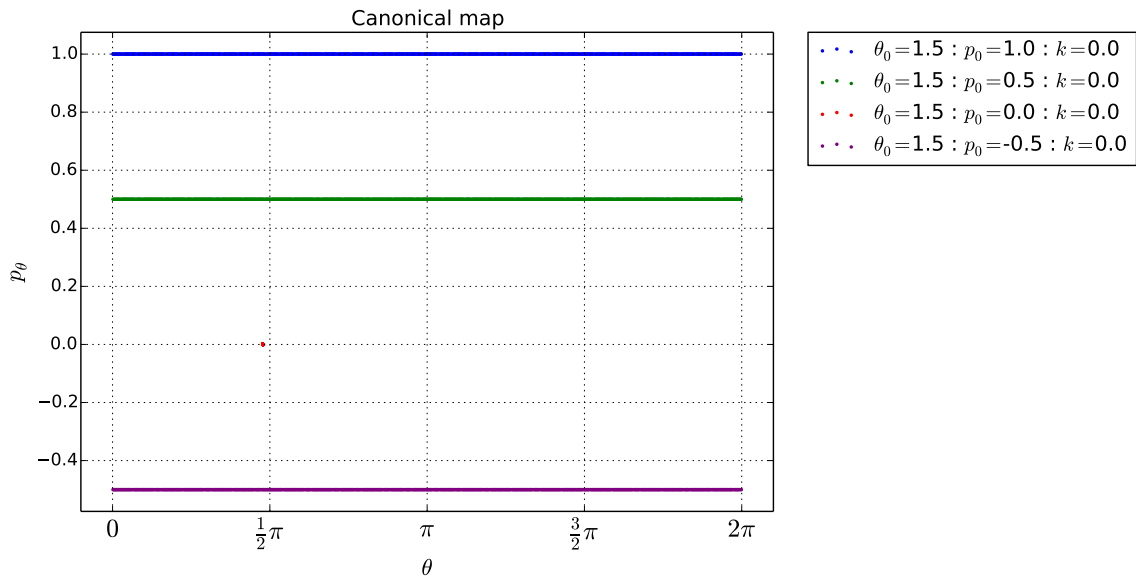


図 2  $k = 0$  のときの位相空間の軌跡

次に、 $\theta_0$  と  $p_0$  を固定して、 $k$  を  $k = 0.1, 1.0, 1.7, 1.99694994, 1.99694995, 2.0$  としたときの位相空間の軌跡を図 3、4 に示す。このグラフから、 $k$  が大きくなるにつれて曲線や閉曲線であった軌跡が分岐していくようになり、トーラスの島と呼ばれる環状の軌跡が現れる様子も観察される。また、 $k$  がある値  $k = k_c \simeq 1.99694995$  以上になると、その曲線からさらに離れた位置を取るように なって、カオス的な振る舞いを示すことも分かる。この  $k_c$  の値は、初期値によって変わることも確かめることができた。

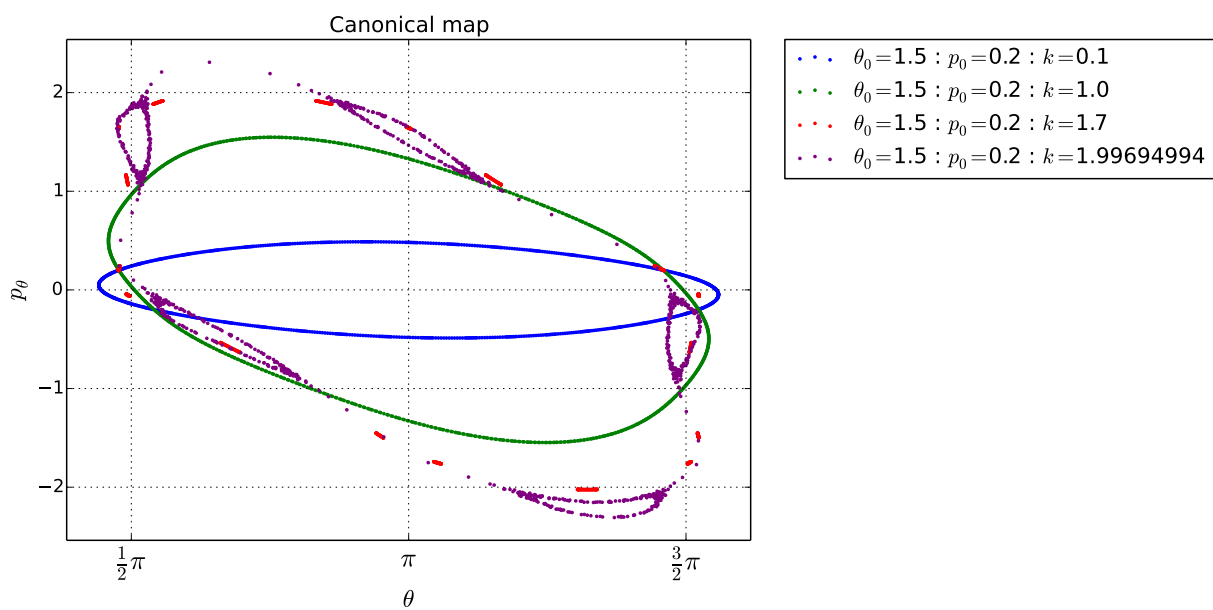


図3  $\theta_0 = 1.5$ 、 $p_0 = 0.2$  としたとき、 $k$  による位相空間での軌跡の変化

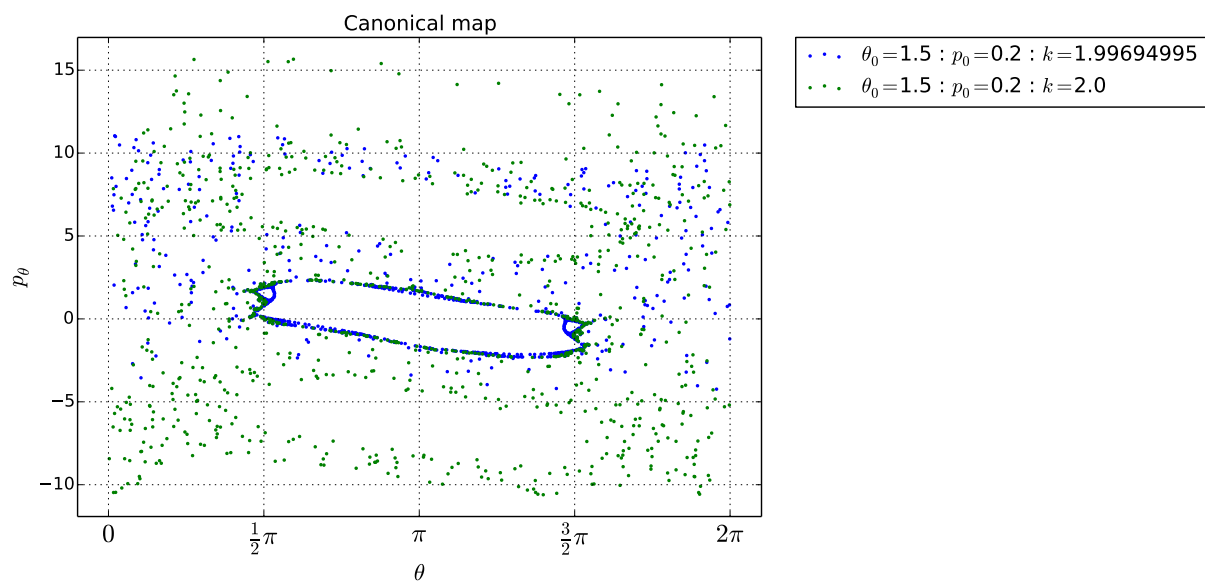


図4  $\theta_0 = 1.5$ 、 $p_0 = 0.2$  としたとき、 $k$  による位相空間での軌跡の変化



## 4 まとめ

ハミルトン系の単純な例である標準写像について、 $k$  による位相空間での軌跡の変化を観察することができた。

## 5 参考文献

- ハーベイ・ゴールド, ジャン・トポチニク, 石川正勝・宮島佐介訳『計算物理学入門』, ピアソン・エデュケーション, 2000.