

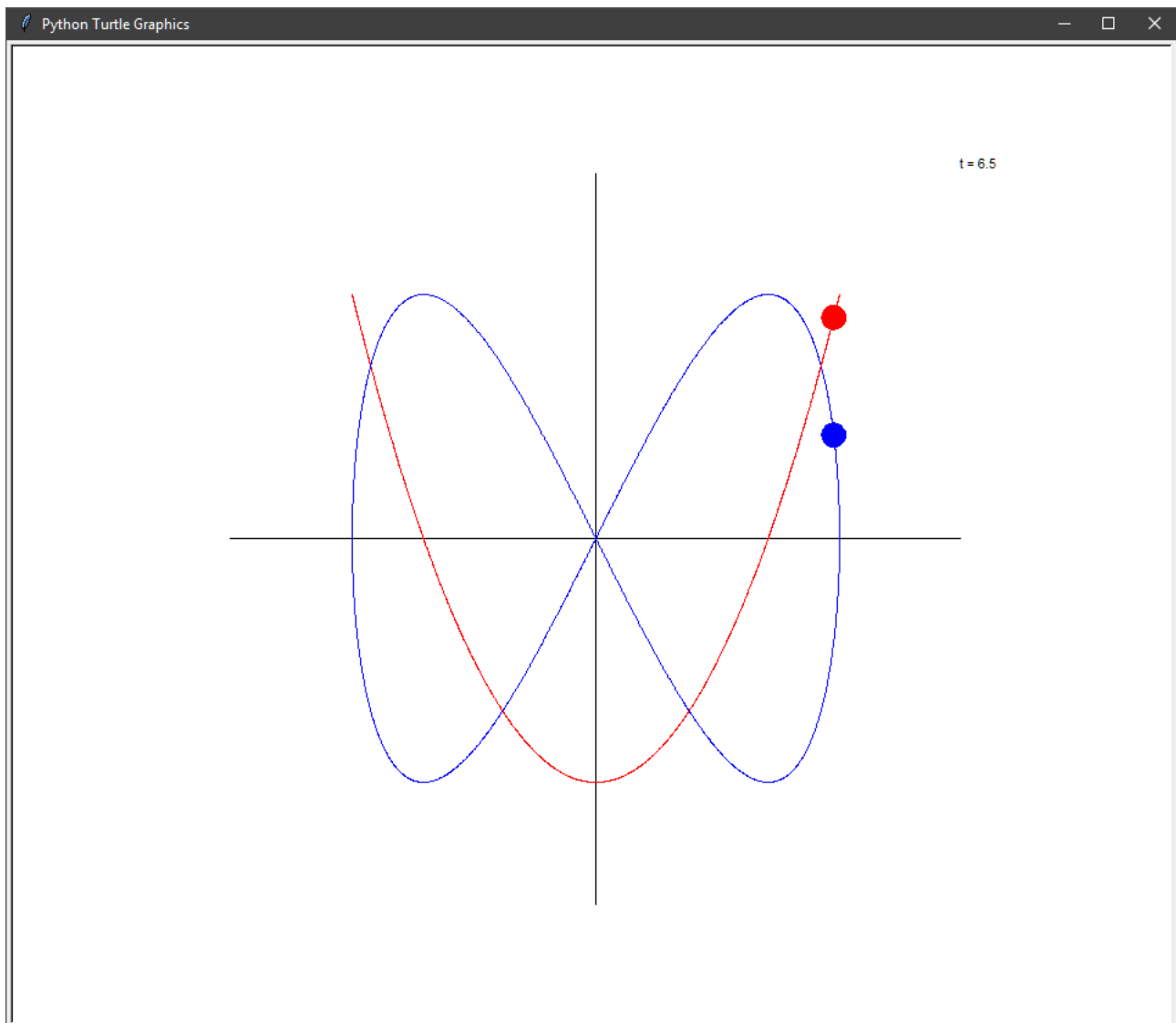
Фігури Ліссажу

Реалізація за допомогою Python

1. Що таке фігури Ліссажу?

Фігури Ліссажу – замкнуті траєкторії, які прокреслюються точкою, що здійснює одночасно два гармонійних коливання у двох взаємно перпендикулярних напрямках. Вперше вивчені французьким науковцем Ж. Ліссажу (фр. J. Lissajous; 1822—1880). Вид фігур залежить від співвідношення між періодами (частотами), фазами і амплітудами обох коливань. У найпростішому випадку (за рівності обох періодів) фігури являють собою еліпси, які при різниці фаз 0 або π вироджуються у відрізки прямих, а при різниці фаз $\pi/2$ і рівності амплітуд перетворюються в коло. Якщо періоди обох коливань не точно збігаються, то різниця фаз весь час змінюється, внаслідок чого еліпс весь час деформується. При істотно різних періодах фігури Ліссажу не спостерігаються, оскільки еліпс деформується швидко, картина розмивається. Однак, якщо періоди відносяться як цілі числа, то через проміжок часу, рівний найменшому кратному обох періодів, точка, що рухається, знову повертається в те ж положення — виходять фігури Ліссажу складнішої форми. Фігури Ліссажу вписуються в прямокутник, центр якого збігається з початком координат, а сторони паралельні осям координат і розташовані по обидва боки від них на відстанях, рівних амплітудами коливань. При цьому кількість дотиків фігури до сторін прямокутника, в який вона вписана дає відношення періодів двох коливань.

2. Реалізація алгоритму



```
1  # -*- coding: utf-8 -*-
2
3  from turtle import Turtle, Screen
4  from math import pi, cos
5  from timeit import default_timer as timer
6
7  # %% class
8  class Body(Turtle):
9      """
10     Oscilating bodies.
11     Amplitude - Object Amplitude
12     omega      - Object Angular Velocity
13     phi        - Phase Shift
14     k1         - x Coordinate Angular Velocity amplifier
15     k2         - y Coordinate Angular Velocity amplifier
16     """
```

```

17     def __init__(self, Amplitude = 100, omega = 1, k1 = 1, k2 = 1, phi = pi/2):
18         Turtle.__init__(self)
19         self.Amplitude = Amplitude
20         self.omega = omega
21         self.k1 = k1
22         self.k2 = k2
23         self.phi = phi
24
25 timer_t = Turtle(visible=False)
26 timer_t.penup()
27 timer_t.setposition(300,300)
28
29 global screen
30 screen = Screen()
31 screen.tracer(0)
32
33 # %% Loop function
34 def loop(bodies):
35     """
36     The movement of bodies in accordance with a sinusoidal law
37     """
38     for body in bodies:
39         body.penup()
40         body.showturtle()
41         body.shape('circle')
42         body.speed(1)
43
44     t0 = timer() # strange nonzeroth starting value of time
45     while True:
46         time = timer() - t0
47         for body in bodies:
48             body.x = body.Amplitude * cos(body.k1*body.omega * time) # x
49             ↪ Coordinate law of motion
50             body.y = body.Amplitude * cos(body.k2*body.omega * time - body.phi) # y
51             ↪ Coordinate law of motion
52             body.goto(body.x, body.y)
53             body.pendown()
54
55             timer_t.write('t = ' + str(round(time, 2)))
56             screen.update()
57             timer_t.undo()
58
59 # %% Axis Draw
60 def AxisDraw(axis, maxval):
61     """
62     Drawing Cartesian coordinates
63     """
64     axis.speed(0)

```

```

65     axis.setposition(0,-maxval)
66     axis.goto(0,maxval)
67     axis.penup()
68     axis.setposition(-maxval,0)
69     axis.pendown()
70     axis.goto(maxval,0)
71
72     # %% Main fumnction
73
74     def main():
75         global screen
76         screen = Screen()
77         axis = Turtle(visible=False)
78         AxisDraw(axis, 300)
79
80         body1 = Body(Amplitude=200, omega=1, k1=1, k2=2, phi=0)
81         body1.color('red')
82
83         body2 = Body(Amplitude=200, omega=1, k1=1, k2=2, phi=pi/2)
84         body2.color('blue')
85
86         loop([body1, body2])
87
88     if __name__ == '__main__':
89         main()
90         screen.mainloop()

```
