

# 问题描述

Hénon maps是一种可以产生混沌现象的离散时间动态系统。

经典的 Hénon 映射公式为：

$$x_{n+1} = 1 - ax_n^2 + y_n$$

$$y_{n+1} = bx_n$$

期末作业第一部分要求使用Python语言，编写代码以满足如下需求：

1. 计算Hénon map的函数：输入任意函数系数 `a`、`b`、初始值 `u0` 以及轨迹长度 `n`，输出 Hénon map的轨迹 `[u0, u1, u2, ..., uN]`。
2. 利用编写的函数计算经典Hénon map的轨迹，参数取值为 `a=1.4`，`b=0.3`，`u0=(0, 0)`，探索 `n` 的取值，求解得到的轨迹，并绘制轨迹图（`x` 为横坐标，`y` 为纵坐标）。
3. 利用编写的函数计算Hénon map的orbit digram，即固定 `b=0.3`，改变 `a` 后获得一系列 Hénon map的轨迹，然后以 `a` 为横轴，`x` 为纵轴绘制orbit digram图。
4. 分析上述画出的orbit digram，找到Hénon map可以收敛到一条周期性轨道的 `a` 值，计算该 `a` 值对应的Hénon map的轨迹并绘图。

## 问题解答思路

### 第1，2小问：

事实上，第1小问和第2小问是同一个问题，只是第2小问只有 `n` 是需要主动输入的，且增加了绘图需求。因此，我把第1和第2小问放在了同一个文件中，即[期末part1\\_12.py](#)，具体代码如下。

```
1  # -*- coding: utf-8 -*-
2
3  # 定义 Henon 映射函数
4  def HM(a, b, x0, y0, n):
5      # 定义 Hénon 映射的 x 更新公式
6      def HMX(a, x0, y0):
7          x = 1 - a * x0**2 + y0 # 经典的 Hénon 映射公式
8          return x
9
10     # 定义 Hénon 映射的 y 更新公式
11     def HMY(b, x0):
12         y = b * x0 # 经典的 Hénon 映射公式
13         return y
14
15     # 存储轨迹的输出列表
16     output = []
17
18     # 迭代计算 n 次
19     for i in range(n):
20         x = HMX(a, x0, y0) # 计算当前的 x 值
21         y = HMY(b, x0)     # 计算当前的 y 值
22         output.append((x, y)) # 将 (x, y) 添加到输出列表中
23         x0, y0 = x, y      # 更新 x0 和 y0 为当前的 x 和 y，以便进行下一次迭代
```

```

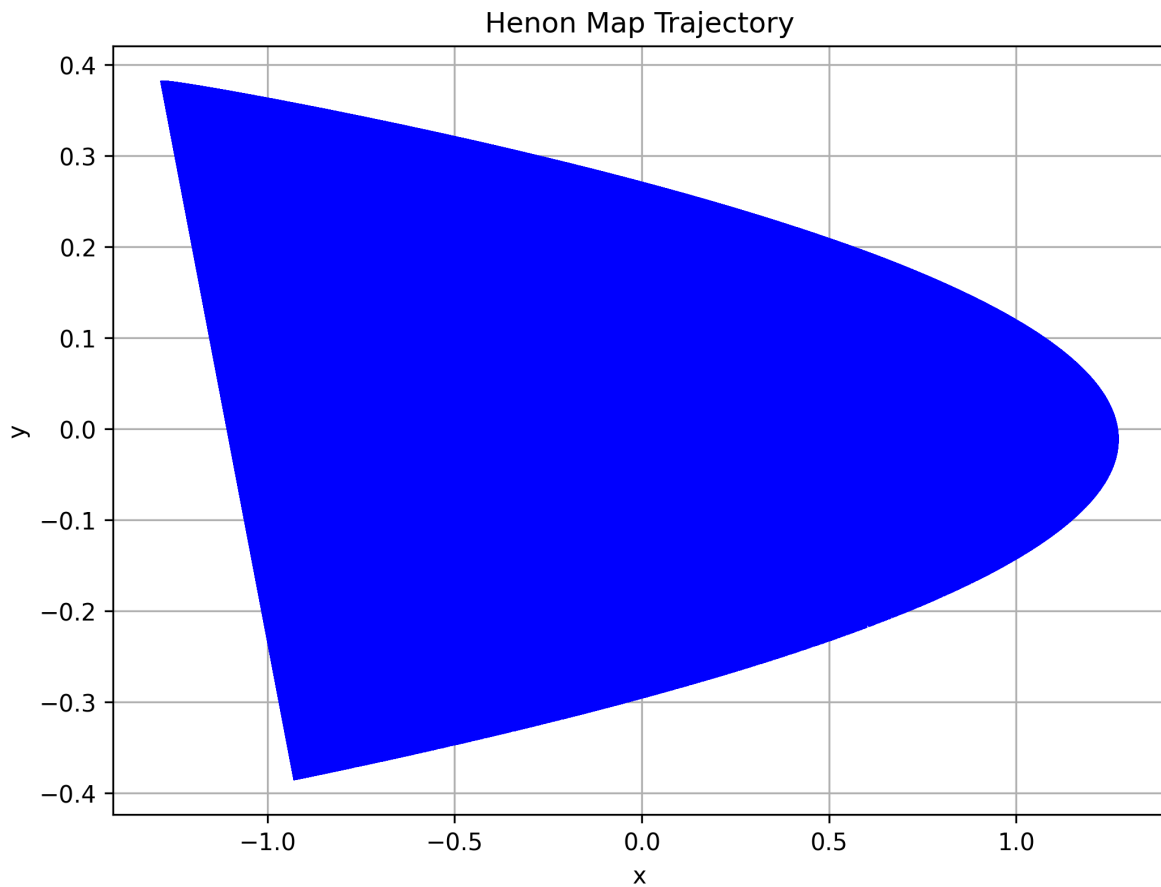
24
25     return output # 返回轨迹数据
26
27 # a = float(input('please input a:'))
28 # b = float(input('please input b:'))
29 # x0 = float(input('please input x0:'))
30 # y0 = float(input('please input y0:'))
31 # n = int(input('please input n:'))
32 ##这里被注释掉的代码可以自由输入初始的a,b,x0,y0,n, 需要时可以取消注释（同时注释掉下面的5
   行代码）
33
34 # 初始化 Henon 映射的参数
35 a = 1.4
36 b = 0.3
37 x0 = 0
38 y0 = 0
39 n = 50000 #迭代次数
40
41 # 打印 Henon 映射的轨迹
42 print(HM(a, b, x0, y0, n))
43
44 # 导入 matplotlib 用于绘制轨迹图
45 import matplotlib.pyplot as plt
46
47 # 获取 Henon 映射的轨迹
48 trajectory = HM(a, b, x0, y0, n)
49
50 # 提取轨迹中的 x 和 y 值
51 x_values = [point[0] for point in trajectory] # 提取 x 值
52 y_values = [point[1] for point in trajectory] # 提取 y 值
53
54 # 绘制轨迹图
55 plt.figure(figsize=(8, 6)) # 设置图像的大小
56 plt.plot(x_values, y_values, color='blue', linewidth=0.5) # 绘制轨迹线, 颜色为
   蓝色, 线宽为0.5
57 plt.title("Henon Map Trajectory") # 设置图像标题
58 plt.xlabel("x") # 设置 x 轴标签
59 plt.ylabel("y") # 设置 y 轴标签
60 plt.grid(True) # 显示网格线
61 plt.show() # 显示图像
62

```

以上是第1和第2小问的代码。首先，Hénon maps公式其实已经在**问题描述**部分给了出来，代码中定义的HMX和HMY两个函数只是将计算 $x_n$ 和 $y_n$ 的公式编写成了Python语言形式。其次，整个代码通过第19行的主循环实现 $n$ 次迭代计算。最后，HMX、HMY以及主循环被封装进了主函数HM中，这个函数输出的是最终得到的轨迹坐标列表。如，当 $a$ ,  $b$ ,  $x_0$ ,  $y_0$ ,  $n$ 分别为1, 2, 3, 4, 5时，打印出的轨迹坐标列表为 $[(-4.0, 6.0), (-9.0, -8.0), (-88.0, -18.0), (-7761.0, -176.0), (-60233296.0, -15522.0)]$ 。

在第1小问中，要求可以“输入任意函数系数 $a$ ,  $b$ 、初始值 $u_0$ 以及轨迹长度 $n$ ”，可以通过将27~31行代码取消注释，同时将第35~39行代码注释，来灵活输入 $a$ ,  $b$ 、初始值 $u_0$ 以及轨迹长度 $n$ ；第2小问固定 $a=1.4$ ,  $b=0.3$ ,  $u_0=(0, 0)$ ，则可以通过相反的操作，手动改变第39行中 $n$ 的值，来观察在不同的 $n$ 的取值下，轨迹图的变化。

下图是当固定 $a=1.4$ ,  $b=0.3$ ,  $u_0=(0, 0)$ ，且 $n$ 为50000时，Hénon maps的轨迹图：



### 第3小问:

相对于前两小问，第3小问要求以  $a$  为横轴， $x$  为纵轴绘制orbit digram图，因此单独放在了另一个文件中，即[期末part1\\_3.py](#)，具体代码如下。

```
1  # -*- coding: utf-8 -*-
2  import matplotlib.pyplot as plt
3
4  # Henon map函数
5  def HM(a, b, x0, y0, n):
6      def HMX(a, x0, y0):
7          x = 1 - a * x0**2 + y0
8          return x
9
10     def HMY(b, x0):
11         y = b * x0
12         return y
13
14     output = []
15     for i in range(n):
16         x = HMX(a, x0, y0) # 计算当前的 x 值
17         y = HMY(b, x0)     # 计算当前的 y 值
18         output.append((x, y))
19         x0, y0 = x, y      # 更新 x0 和 y0 为当前的 x 和 y
20
21     return output
22
23 # 固定参数
24 b = 0.3
25 x0 = 0
```

```

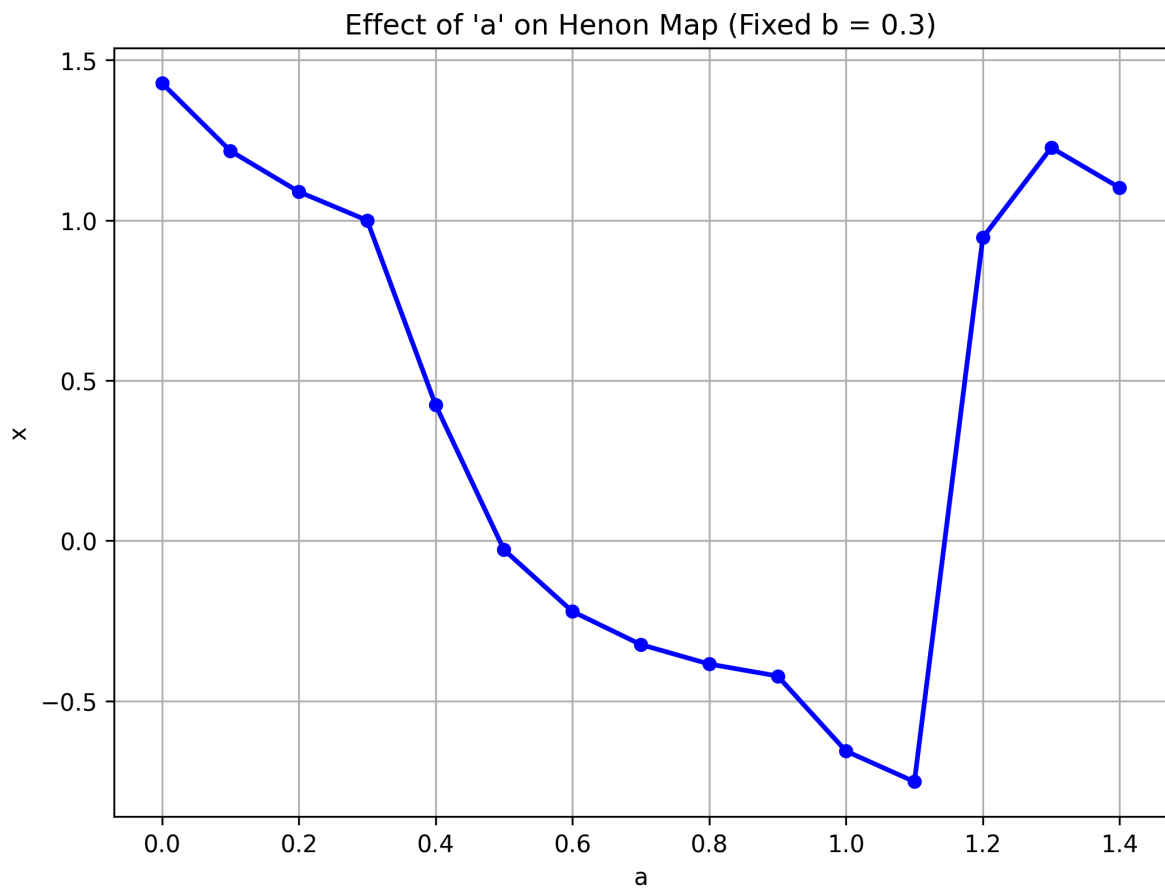
26 y0 = 0
27 n = 50000 # 迭代次数
28
29 # 创建a值的范围，设置不同的a值
30 a_values = [i * 0.1 for i in range(15)] # 生成a的不同值，0.0到1.4之间，步长为0.1
31
32 # 存储对应的x值
33 x_results = []
34
35 # 对每个a值计算Henon映射轨迹的最后一个x值（收敛到稳定状态的x值）
36 for a in a_values:
37     trajectory = HM(a, b, x0, y0, n)
38     x_results.append(trajectory[-1][0]) # 获取轨迹的最后一个x值
39
40 # 绘制a值与x的关系图
41 plt.figure(figsize=(8, 6))
42 plt.plot(a_values, x_results, color='blue', marker='o', linestyle='-',
43         linewidth=2, markersize=5)
44 plt.title("Effect of 'a' on Henon Map (Fixed b = 0.3)")
45 plt.xlabel("a")
46 plt.ylabel("x")
47 plt.grid(True)
48 plt.show()

```

在第30行中，`a` 的范围被设置在0.0到1.4之间，步长为0.1。在0.1的步长下，如果 `a` 的值大于0.14，则Python会报错 `OverflowError: (34, 'Result too large')`，经过查询，这是由于浮点数的计算精度导致的问题，暂时没有找到很好的解决办法，因此，第三小问中 `a` 的范围被限制在了0.0到1.4之间，步长为0.1。

同时，在同一个 `a` 值下，由于迭代次数 `n` 的不同，会得到一系列的 `x` 值，即  $[x_1, x_2, \dots, x_{n+1}]$ 。第三小问要求以 `a` 为横轴，`x` 为纵轴，挥着轨迹图，我取的 `x` 是迭代 `n` 次之后的最后一个 `x`，即第38行代码中的 `trajectory[-1][0]`。

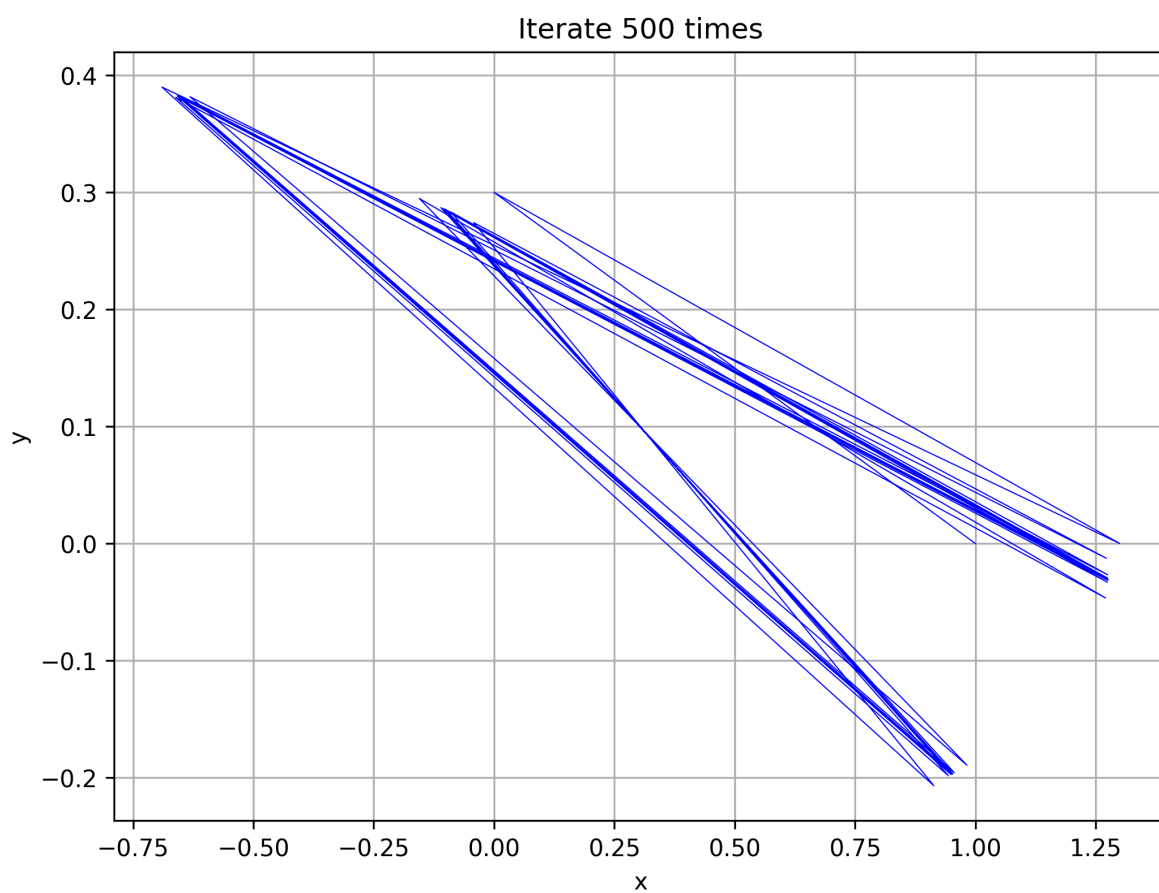
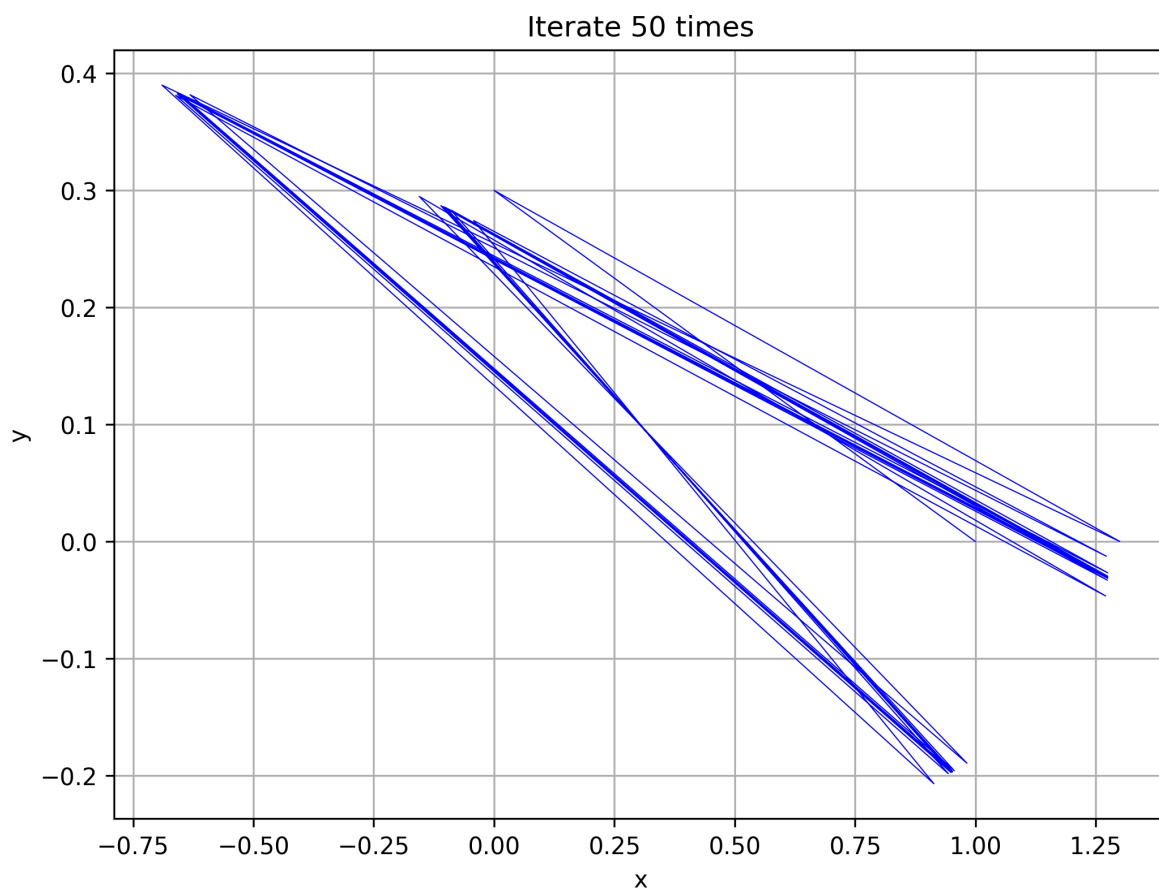
这里是固定 `b` 为0.3，`x0`，`y0` 均为0，`n` 为50000，`a` 为横轴，最后一次迭代得到的 `x` 为纵轴，绘制的轨迹图：

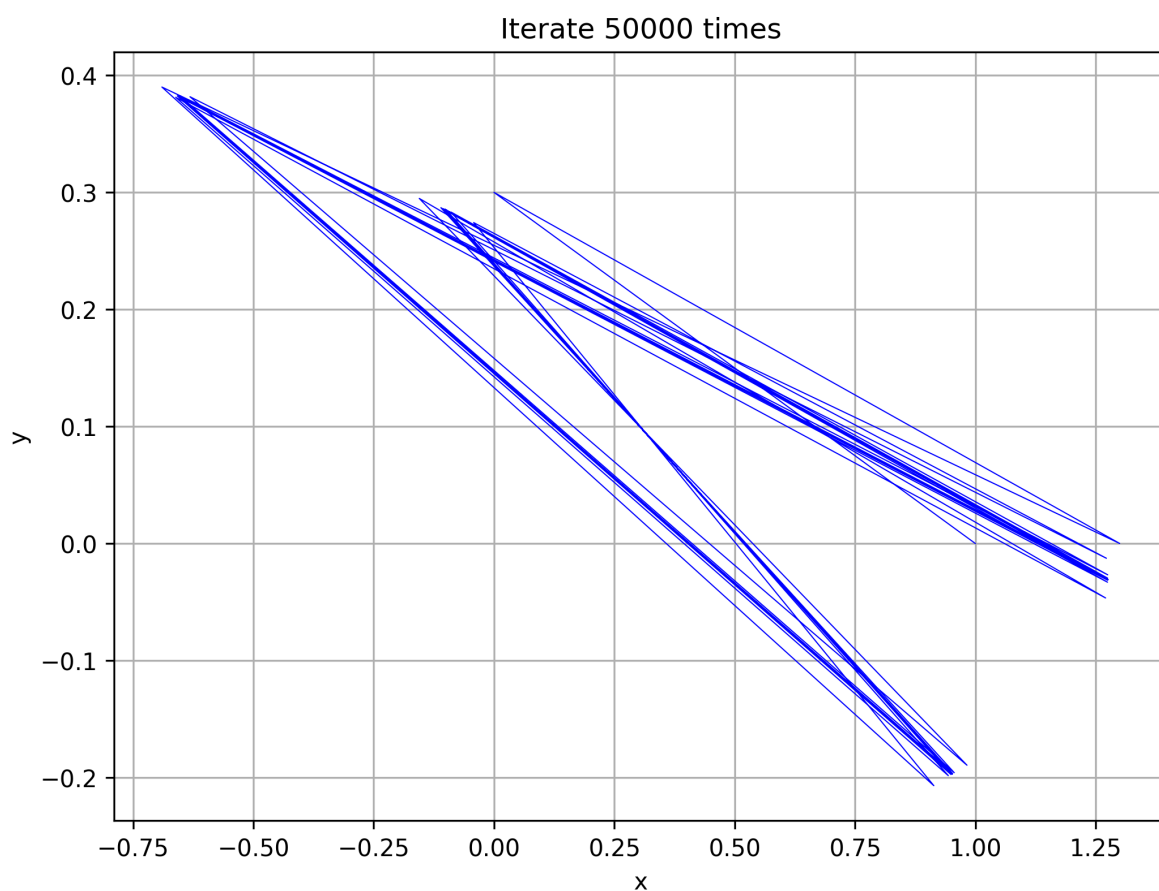
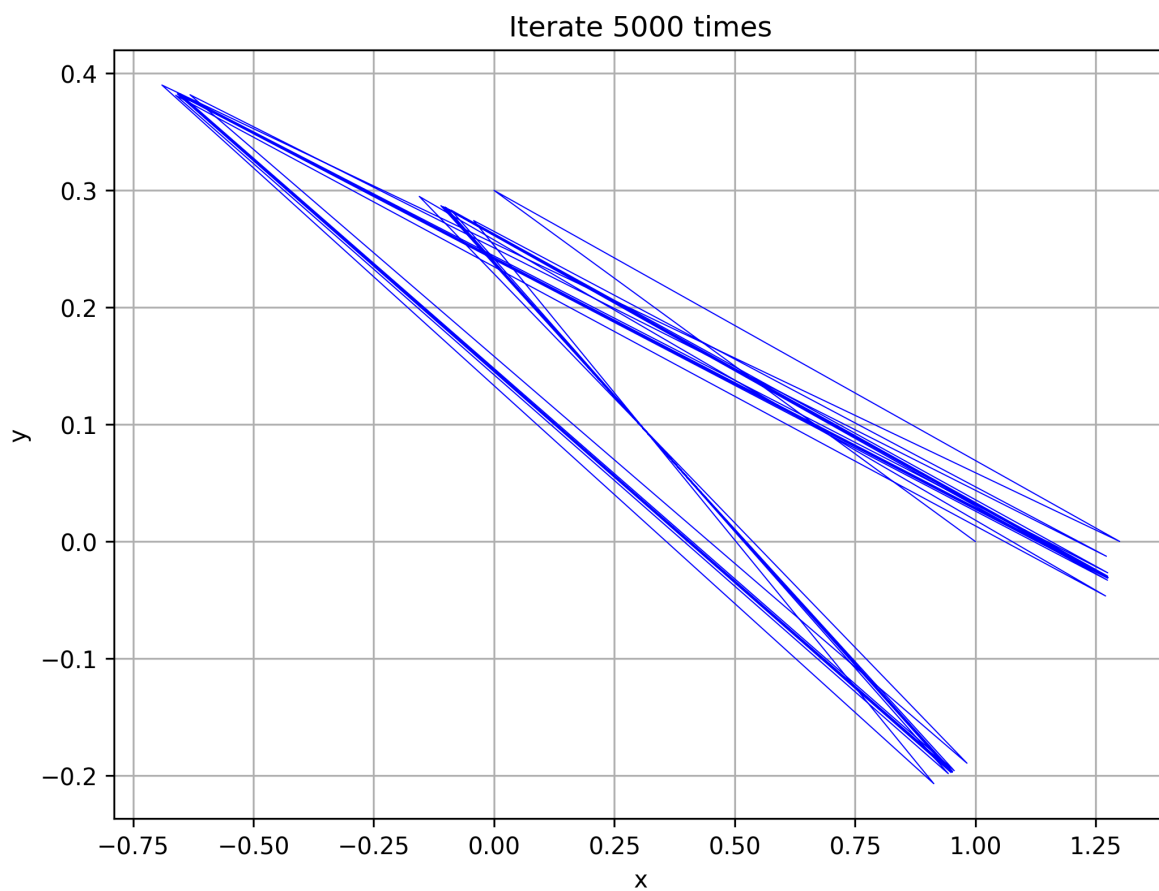


## 第4小问

第4小问要求找到Hénon map可以收敛到一条周期性轨道的  $a$  值，我试图在第3小问绘制出的  $a$  与  $x$  的关系图中找到  $x$  呈周期性变化的趋势，但是受限于  $a$  的取值，这个方法失效了。

但是经过手动修改  $a$  的值，我发现，在固定  $b$  为0.3， $x_0, y_0$  均为0， $n$  为50000的情况下，当  $a$  取值为1，绘制出的  $x$  和  $y$  的轨迹图最终收敛到一个周期性轨道里，以下分别是迭代50，500，5000，50000次的轨迹图：





迭代次数的增加没有带来轨迹图的变化（至少肉眼看来如此），我认为  $a$  取1是一个可以使Hénon map可以收敛到一条周期性轨道的值。当  $a$  取1时，这个周期可能很长（在迭代500次的轨迹坐标列表中，我没有发现重复的坐标）。

#### 第4小问附录：迭代500次时的轨迹坐标列表如下：

[(1, 0.0), (0.0, 0.3), (1.3, 0.0), (-0.6900000000000002, 0.39), (0.9138999999999998,  
-0.20700000000000005), (-0.04221320999999975, 0.2741699999999999), (1.2723880449014957,  
-0.012663962999999926), (-0.6316352998082505, 0.3817164134704487), (0.9827532615065901,  
-0.18949058994247514), (-0.15529456294431554, 0.29482597845197706), (1.270709577171911,  
-0.04658836888329466), (-0.6612911983997116, 0.3812128731515733), (0.9439068240706465,  
-0.1983873595199135), (-0.08934745204704797, 0.28317204722119393), (1.2751890800338943,  
-0.02680423561411439), (-0.6529114254518041, 0.38255672401016827), (0.9562633945246615,  
-0.19587342763554125), (-0.1103131073433696, 0.28687901835739843), (1.2747100367056488,  
-0.03309393220301088), (-0.6579796098811272, 0.3824130110116946), (0.9494758439923743,  
-0.19739388296433816), (-0.09889826128936965, 0.2848427531977123), (1.275061887111652,  
-0.029669478386810895), (-0.6554522943515378, 0.38251856613349555), (0.9529008559628005,  
-0.19663568830546135), (-0.10465572960009922, 0.28587025678884015), (1.274917435050711,  
-0.03139671888002977), (-0.656811185076314, 0.3824752305152133), (0.9510742976738613,  
-0.19704335552289418), (-0.10158567521872272, 0.28532228930215836), (1.2750026398925145,  
-0.030475702565616814), (-0.6561074342984978, 0.38250079196775433), (0.9520238266259966,  
-0.19683223028954935), (-0.103181596753155, 0.28560714798779896), (1.2749607060792683,  
-0.0309544790259465), (-0.6564792810720929, 0.38248821182378046), (0.9515231653468486,  
-0.19694378432162785), (-0.10234011851331393, 0.28545694960405454), (1.2749834497467354,  
-0.030702035553994176), (-0.6562848326820803, 0.3824950349240206), (0.9517852533154745,  
-0.19688544980462408), (-0.10278061823342599, 0.28553557599464235), (1.2749717205101971,  
-0.030834185470027795), (-0.65638707357076, 0.38249151615305915), (0.9516475258022727,  
-0.196916122071228), (-0.10254913543681532, 0.2854942577406818), (1.2749779325618436,  
-0.030764740631044594), (-0.6563334691507176, 0.3824933797685531), (0.9517197570411371,  
-0.19690004074521525), (-0.10267053668765622, 0.28551592711234114), (1.2749746880086097,  
-0.030801161006296863), (-0.6563616160689485, 0.3824924064025829), (0.9516818353539411,  
-0.19690848482068454), (-0.10260680056333046, 0.28550455060618235), (1.2749763950843394,  
-0.030782040168999138), (-0.6563468481912564, 0.3824929185253018), (0.9517017333947055,  
-0.19690405445737694), (-0.10264024380386408, 0.28551052001841165), (1.274975500370295,  
-0.030792073141159223), (-0.6563545996856431, 0.38249265011108846), (0.9516912895825876,  
-0.19690637990569293), (-0.10262269057306161, 0.2855073868747763), (1.274975970254322,  
-0.030786807171918482), (-0.6563505318978683, 0.3824927910762966), (0.9516967703536818,  
-0.19690515956936047), (-0.10263190227098903, 0.28550903110610454), (1.2749757237423427,  
-0.03078957068129671), (-0.6563526668136074, 0.3824927171227028), (0.9516938938893684,  
-0.1969058000440822), (-0.10262706771039065, 0.2855081681668105), (1.2749758531399773,  
-0.030788120313117194), (-0.6563515464031301, 0.38249275594199317), (0.951695403476213,  
-0.19690546392093902), (-0.1026296049186908, 0.2855086210428639), (1.2749757852370973,  
-0.030788881475607238), (-0.6563521344165602, 0.38249273557112917), (0.951694611217955,  
-0.19690564032496805), (-0.10262827334626246, 0.2855083833653865), (1.2749758208753512,  
-0.030788482003878737), (-0.6563518258206543, 0.3824927462626054), (0.9516950270044988,  
-0.19690554774619629), (-0.10262897217128997, 0.28550850810134965), (1.2749758021724142,  
-0.03078869165138699), (-0.6563519877765781, 0.3824927406517243), (0.9516948087934589,  
-0.19690559633297341), (-0.10262860541739169, 0.28550844263803765), (1.274975811988119,  
-0.030788581625217505), (-0.6563519027799806, 0.38249274359643565), (0.9516949233135346,  
-0.19690557083399415), (-0.10262879789474863, 0.28550847699406035), (1.2749758068367392,  
-0.03078863936842459), (-0.6563519473874185, 0.38249274205102174), (0.9516948632117652,  
-0.19690558421622556), (-0.10262869687988602, 0.28550845896352955), (1.2749758095402661,  
-0.030788609063965803), (-0.6563519239768228, 0.38249274286207985), (0.9516948947540029,  
-0.19690557719304685), (-0.10262874989387955, 0.28550846842620087), (1.2749758081214204,



-0.030788624968163862), (-0.6563519362630329, 0.3824927424364261), (0.9516948782001937,  
-0.19690558087890986), (-0.1026287220713914, 0.2855084634600581), (1.2749758088660512,  
-0.03078861662141742), (-0.656351929815059, 0.38249274265981537), (0.9516948868878633,  
-0.19690557894451768), (-0.10262873667302055, 0.285508466066359), (1.2749758084752587,  
-0.030788621001906165), (-0.6563519331990457, 0.3824927425425776), (0.951694882328453,  
-0.19690557995971372), (-0.10262872900988176, 0.2855084646985359), (1.2749758086803522,  
-0.030788618702964527), (-0.6563519314230827, 0.38249274260410565), (0.9516948847212945,  
-0.1969055794269248), (-0.10262873303160283, 0.28550846541638836), (1.2749758085727163,  
-0.030788619909480848), (-0.6563519323551326, 0.38249274257181487), (0.9516948834654984,  
-0.19690557970653977), (-0.10262873092094826, 0.2855084650396495), (1.2749758086292051,  
-0.030788619276284478), (-0.6563519318659801, 0.38249274258876154), (0.9516948841245574,  
-0.19690557955979401), (-0.10262873202864872, 0.2855084652373672), (1.274975808599559,  
-0.030788619608594615), (-0.6563519321226939, 0.3824927425798677), (0.9516948837786743,  
-0.1969055796368082), (-0.10262873144731252, 0.2855084651336023), (1.2749758086151177,  
-0.030788619434193754), (-0.6563519319879672, 0.3824927425845353), (0.9516948839601982,  
-0.19690557959639013), (-0.1026287317524052, 0.28550846518805945), (1.2749758086069523,  
-0.030788619525721556), (-0.6563519320586734, 0.38249274258208565), (0.9516948838649323,  
-0.196905579617602), (-0.102628731592289, 0.2855084651594797), (1.2749758086112375,  
-0.0307886194776867), (-0.6563519320215656, 0.38249274258337124), (0.9516948839149293,  
-0.19690557960646968), (-0.10262873167632044, 0.28550846517447875), (1.2749758086089886,  
-0.03078861950289613), (-0.6563519320410405, 0.38249274258269655), (0.9516948838886898,  
-0.19690557961231214), (-0.10262873163221897, 0.28550846516660694), (1.274975808610169,  
-0.03078861948966569), (-0.6563519320308201, 0.3824927425830507), (0.9516948839024604,  
-0.196905579609246), (-0.10262873165536354, 0.2855084651707381), (1.2749758086095495,  
-0.030788619496609063), (-0.6563519320361836, 0.38249274258286486), (0.9516948838952339,  
-0.19690557961085506), (-0.10262873164321781, 0.28550846516857015), (1.2749758086098746,  
-0.030788619492965342), (-0.656351932033369, 0.38249274258296234), (0.9516948838990262,  
-0.19690557961001068), (-0.10262873164959166, 0.28550846516970785), (1.274975808609704,  
-0.030788619494877496), (-0.6563519320348462, 0.3824927425829112), (0.9516948838970358,  
-0.19690557961045385), (-0.10262873164624631, 0.2855084651691107), (1.2749758086097935,  
-0.03078861949387389), (-0.6563519320340708, 0.38249274258293803), (0.9516948838980805,  
-0.19690557961022123), (-0.10262873164800221, 0.28550846516942413), (1.2749758086097465,  
-0.030788619494400662), (-0.6563519320344775, 0.38249274258292393), (0.9516948838975325,  
-0.19690557961034325), (-0.10262873164708117, 0.28550846516925976), (1.274975808609771,  
-0.03078861949412435), (-0.656351932034264, 0.3824927425829313), (0.9516948838978202,  
-0.19690557961027919), (-0.10262873164756467, 0.285508465169346), (1.2749758086097582,  
-0.0307886194942694), (-0.6563519320343761, 0.3824927425829275), (0.9516948838976691,  
-0.19690557961031283), (-0.10262873164731065, 0.28550846516930073), (1.274975808609765,  
-0.030788619494193193), (-0.6563519320343171, 0.3824927425829295), (0.9516948838977486,  
-0.19690557961029512), (-0.10262873164744427, 0.28550846516932454), (1.2749758086097613,  
-0.03078861949423328), (-0.656351932034348, 0.38249274258292837), (0.9516948838977068,  
-0.19690557961030442), (-0.10262873164737407, 0.28550846516931205), (1.2749758086097633,  
-0.03078861949421222), (-0.6563519320343321, 0.382492742582929), (0.9516948838977286,  
-0.1969055796102996), (-0.10262873164741079, 0.28550846516931855), (1.2749758086097622,  
-0.030788619494223235), (-0.6563519320343403, 0.38249274258292865), (0.9516948838977174,  
-0.19690557961030208), (-0.10262873164739184, 0.2855084651693152), (1.274975808609763,  
-0.03078861949421755), (-0.6563519320343363, 0.38249274258292887), (0.9516948838977229,  
-0.1969055796103009), (-0.10262873164740119, 0.2855084651693169), (1.2749758086097627,  
-0.030788619494220355), (-0.6563519320343385, 0.3824927425829288), (0.95169488389772,  
-0.19690557961030156), (-0.10262873164739619, 0.285508465169316), (1.2749758086097627,

[illegible]

[illegible]

[illegible]

-0.19690557961030125), (-0.10262873164739844, 0.2855084651693164), (1.2749758086097627,  
-0.03078861949421953), (-0.6563519320343376, 0.3824927425829288)]