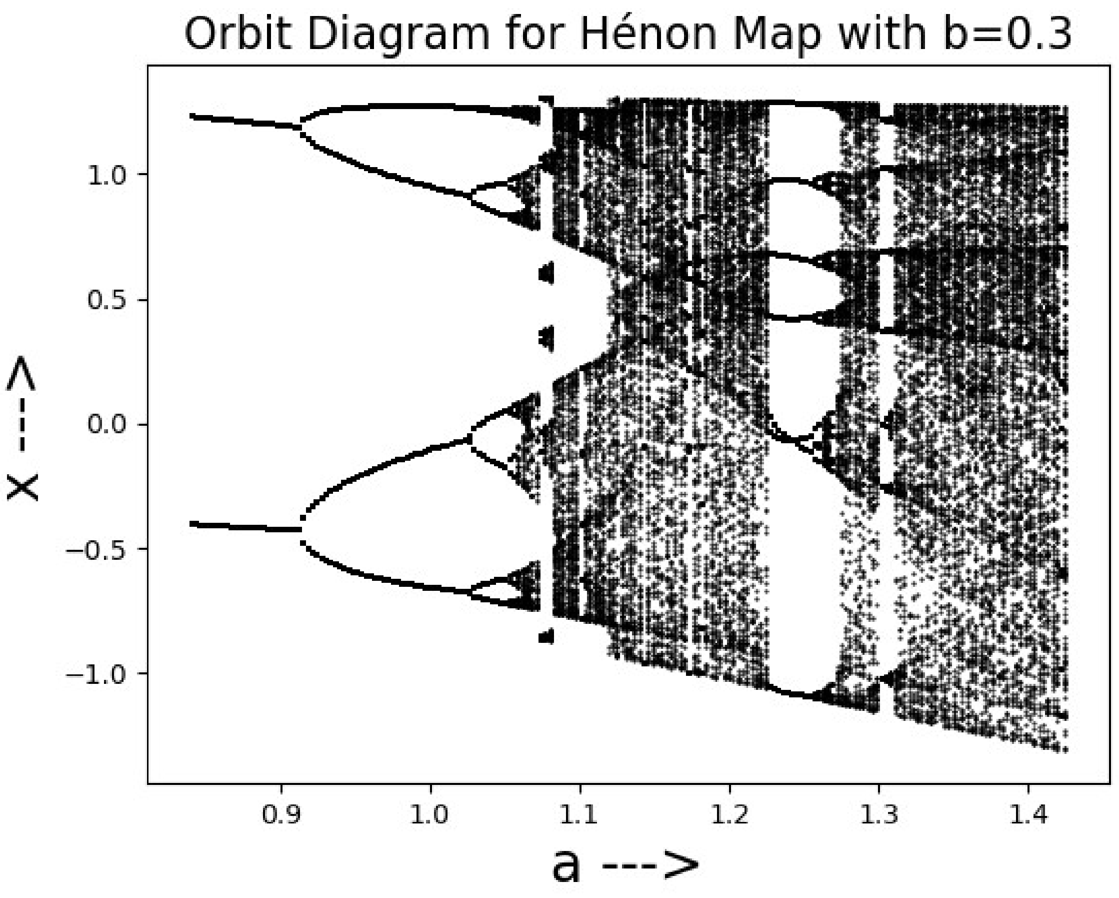
Below is the Orbit Diagram for Henon Map. Here we have two parameters in Henon Map, these are conventionally taken as 'a', 'b'. Keeping the 'b' value fixed we will vary 'a' so that the bifurcation is shown.

```python
# importing necessary libraries
import numpy
import matplotlib.pyplot as plt

b=0.3 # value of 'b' parameter
def henon_attractor(x,y,a): # Henon map is defined through this
function
    xn=1-a*x**2+y
    yn=b*x
    return xn,yn

# starting from the initial point the map will be applied 50000 times
and last 250 points will be stored for graph draw
steps=50000
Points=250
def H(a):
    X=numpy.zeros(steps+1)
    Y=numpy.zeros(steps+1)
    X[0],Y[0]=0.07,0.04
    # initial point taken (0.07, 0.04)
    for i in range(steps):
        x_next,y_next=henon_attractor(X[i],Y[i],a)
        X[i+1]=x_next
        Y[i+1]=y_next
    return X[steps+1-Points:steps+1]

A=numpy.arange(0.84,1.44,0.003) # A is an array storing the values of
'a' parameter
for a in A:
    avalues=a*numpy.ones(Points)
    Xv=H(a)
    plt.plot(avalues,Xv,'.',c='black',ms=1) # plotting of x
coordinates of the point for each 'a' value
plt.xlabel('a --->',fontsize=20)
plt.ylabel('x --->',fontsize=20)
plt.title('Orbit Diagram for Hénon Map with b=0.3',fontsize=16)
plt.show()
```

```
C:\Users\USER\AppData\Local\Temp\ipykernel_5888\1802244353.py:7:
RuntimeWarning: overflow encountered in scalar power
  xn=1-a*x**2+y
```

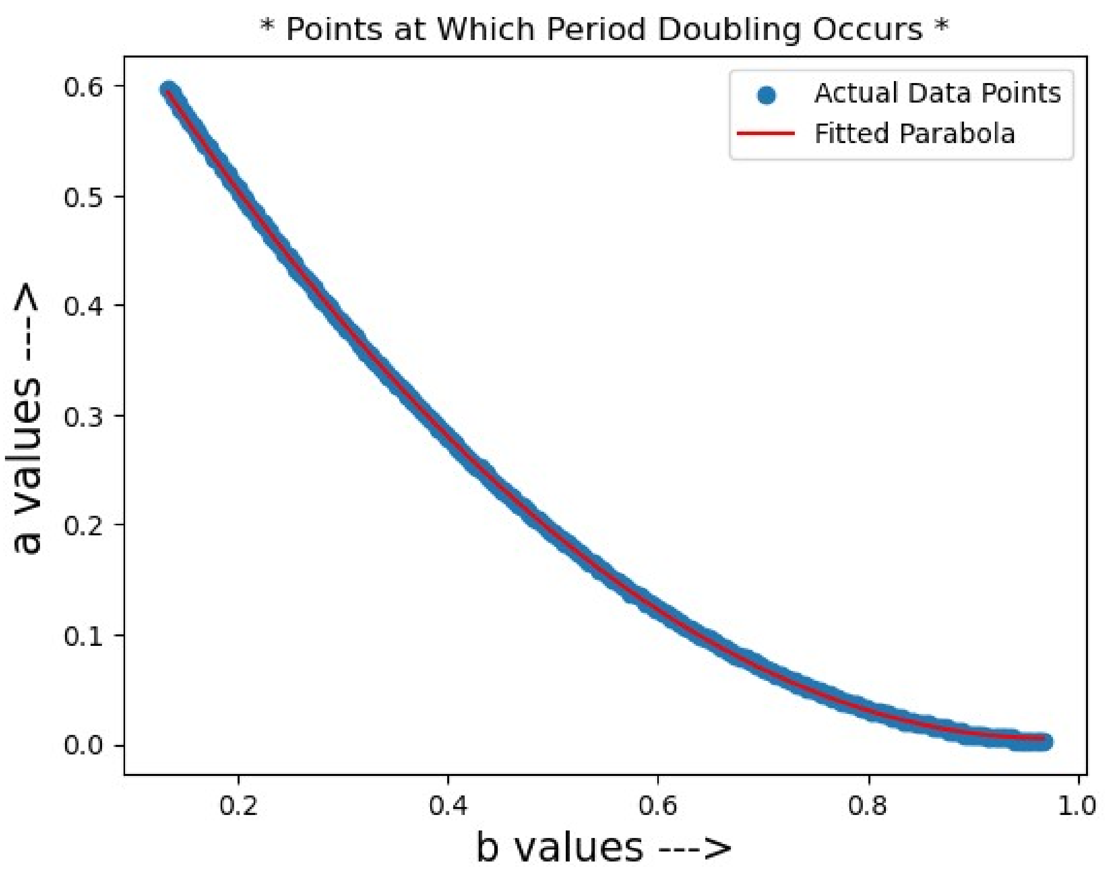


From stability and fixed point analysis it can be easily shown that for a < -0.25(1-b)^2 there will be NO fixed points.

For -0.25(1-b)^2 < a < 0.75(1-b)^2 there will be two fixed points- one attracting and other saddle

For a > 0.75(1-b)^2 there will be two attracting fixed points.

```
# So it's clear that period doubling occurs for a = 0.75(1-b)^2
# We shall verify that computationally

# importing necessary libraries
import numpy
import matplotlib.pyplot as plt

B=numpy.arange(0.01,0.97,0.003) # 'b' parameter is varied from 0.01 to 
0.97
def henon_attractor(x,y,a,b): # Henon Map is defined
    xn=1-a*x**2+y
    yn=b*x
    return xn,yn

steps=1000 # Henon Map is applied for 1000 times
def H(a,b):
    X=numpy.zeros(steps+1)
    Y=numpy.zeros(steps+1)
    X[0],Y[0]=0.07,0.04 # initial starting value

    for i in range(steps):
        x_next,y_next=henon_attractor(X[i],Y[i],a,b)
        X[i+1]=x_next
        Y[i+1]=y_next

    if abs(X[-2]-X[-1])>=0.6:
        return 1 # if period doubling has already occured then the
function returns 1
    else:
        return 0 # # otherwise the function returns 0

A=numpy.arange(0,0.6,0.003) # 'a' is varied
avalues,bvalues=[],[]
for b in B:
    for a in A:
        if H(a,b)==1:
            avalues.append(a)
            bvalues.append(b) # if period doubling occurs then 'a' and
'b' parameters will be appended
            break

a=numpy.array(avalues)
b=numpy.array(bvalues)
coefficients=numpy.polyfit(b,a,2) # quadratic fit keeping 'a' along +y
axis and 'b' along +x axis
polynomial=numpy.poly1d(coefficients)
# the coefficient of b^2 is close to 0.75
y_fit=polynomial(b)
plt.scatter(b,a,label='Actual Data Points')
plt.plot(b,y_fit,c='r',label='Fitted Parabola')
plt.xlabel('b values --->',fontsize=15)
plt.ylabel('a values --->',fontsize=16)
plt.legend()
plt.title('* Points at Which Period Doubling Occurs *')
plt.show()
```

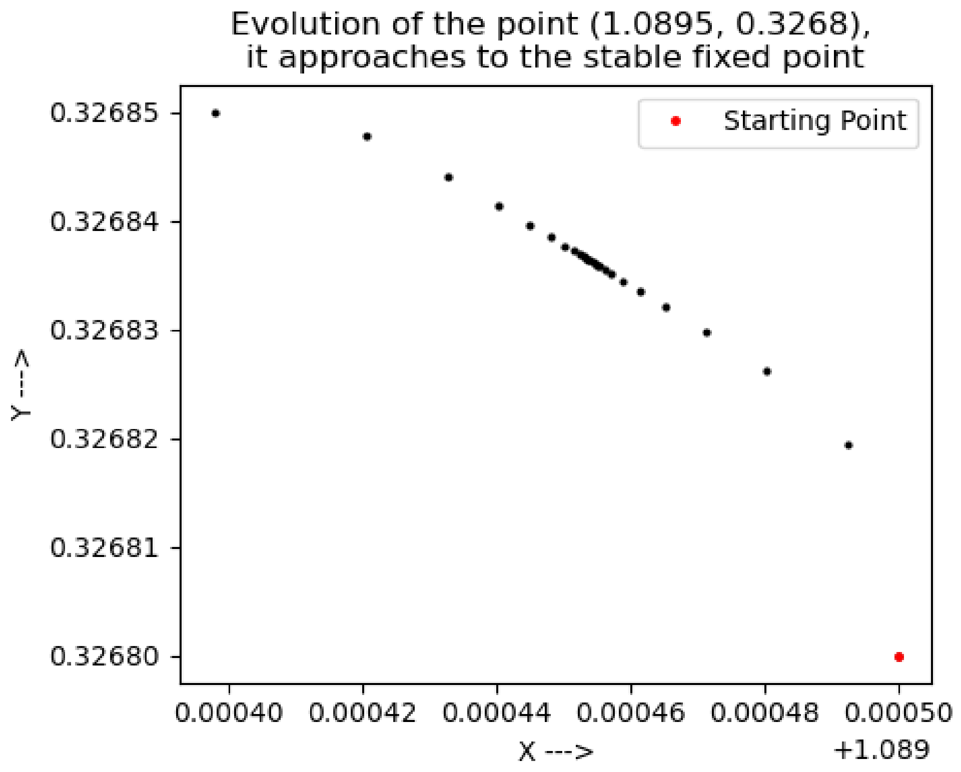


We can find the fixed points analytically for -0.25(1-b)^2 < a < 0.75(1-b)^2. We shall take two points which will be very close to the fixed points.

For the stable fixed point the point chosen close to it will approach to the stable fixed points iteratively.

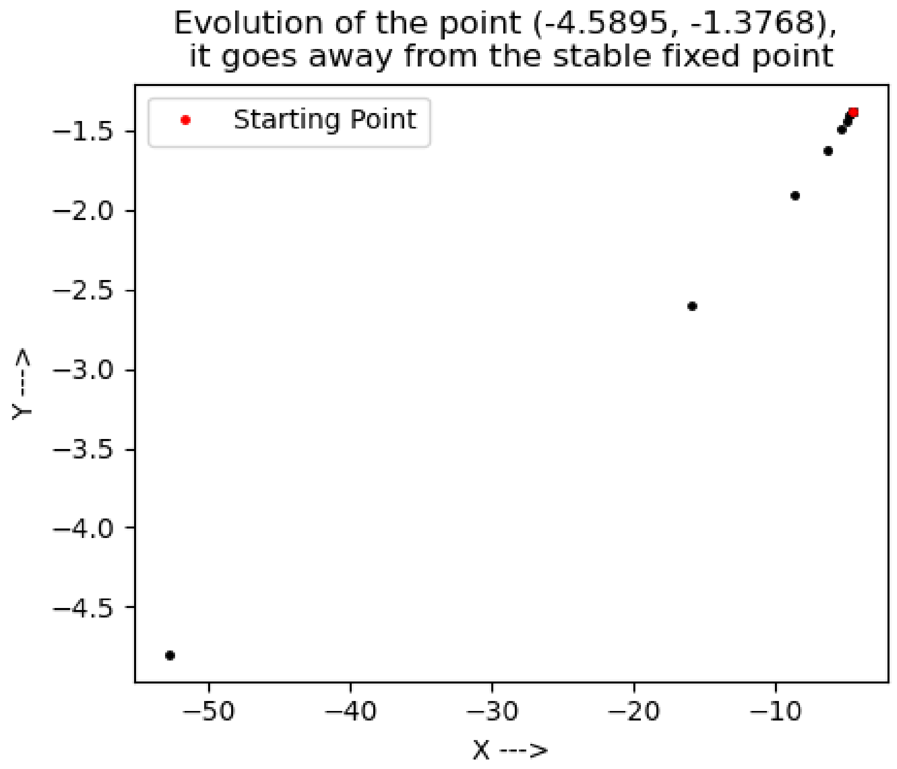
On the other hand the point close to the saddle point will go far away from the fixed point.

```
# importing libraries
import numpy
import matplotlib.pyplot as plt
b,a=0.3,0.2 # two values of 'a' and 'b' for which we have two fixed
poinys only
def henon_attractor(x,y): # defining Henon Map
    xn=1-a*x**2+y
    yn=b*x
    return xn,yn
steps=25
X=numpy.zeros(steps+1)
Y=numpy.zeros(steps+1)
X[0],Y[0]=1.0895,0.3268 # point (1.0895, 0.3268) is very close to the
stable fixed point
for i in range(steps):
    x_next,y_next=henon_attractor(X[i],Y[i])
    X[i+1]=x_next
    Y[i+1]=y_next

# plotting
plt.figure(figsize=(5,4))
plt.plot(X,Y, '.',c='black',ms=4)
plt.plot(X[0],Y[0],'.',c='red',ms=5,label='Starting Point')
plt.legend()
plt.xlabel('X --->')
plt.ylabel('Y --->')
plt.title('Evolution of the point (1.0895, 0.3268), \nit approaches to
the stable fixed point',fontsize=12)
plt.show()
```



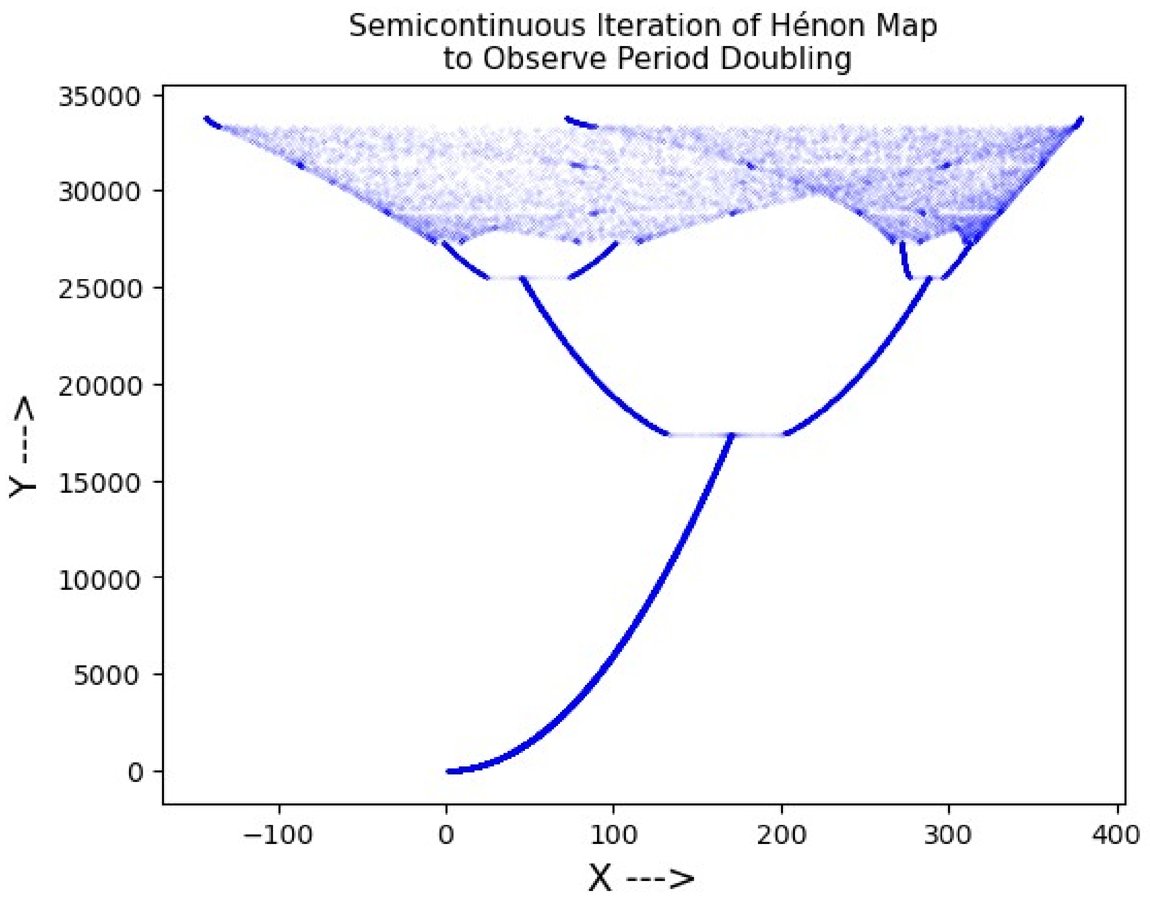
```
b,a=0.3,0.2 # two values of 'a' and 'b' for which we have two fixed
poinys only
def henon_attractor(x,y): # defining Henon Map
    xn=1-a*x**2+y
    yn=b*x
    return xn,yn
steps=19 # datapoints 19
X=numpy.zeros(steps+1)
Y=numpy.zeros(steps+1)
X[0],Y[0]=-4.5895,-1.3768 # point (-4.5895, -1.3768) is very close to
```

```
the stable fixed point
for i in range(steps):
    x_next,y_next=henon_attractor(X[i],Y[i])
    X[i+1]=x_next
    Y[i+1]=y_next
# plotting
plt.figure(figsize=(5,4))
plt.plot(X,Y, '.',c='black',ms=5)
plt.plot(X[0],Y[0],'.',c='red',ms=5,label='Starting Point')
plt.legend()
plt.xlabel('X --->')
plt.ylabel('Y --->')
plt.title('Evolution of the point (-4.5895, -1.3768), \nit goes away
from the stable fixed point',fontsize=12)
plt.show()
```



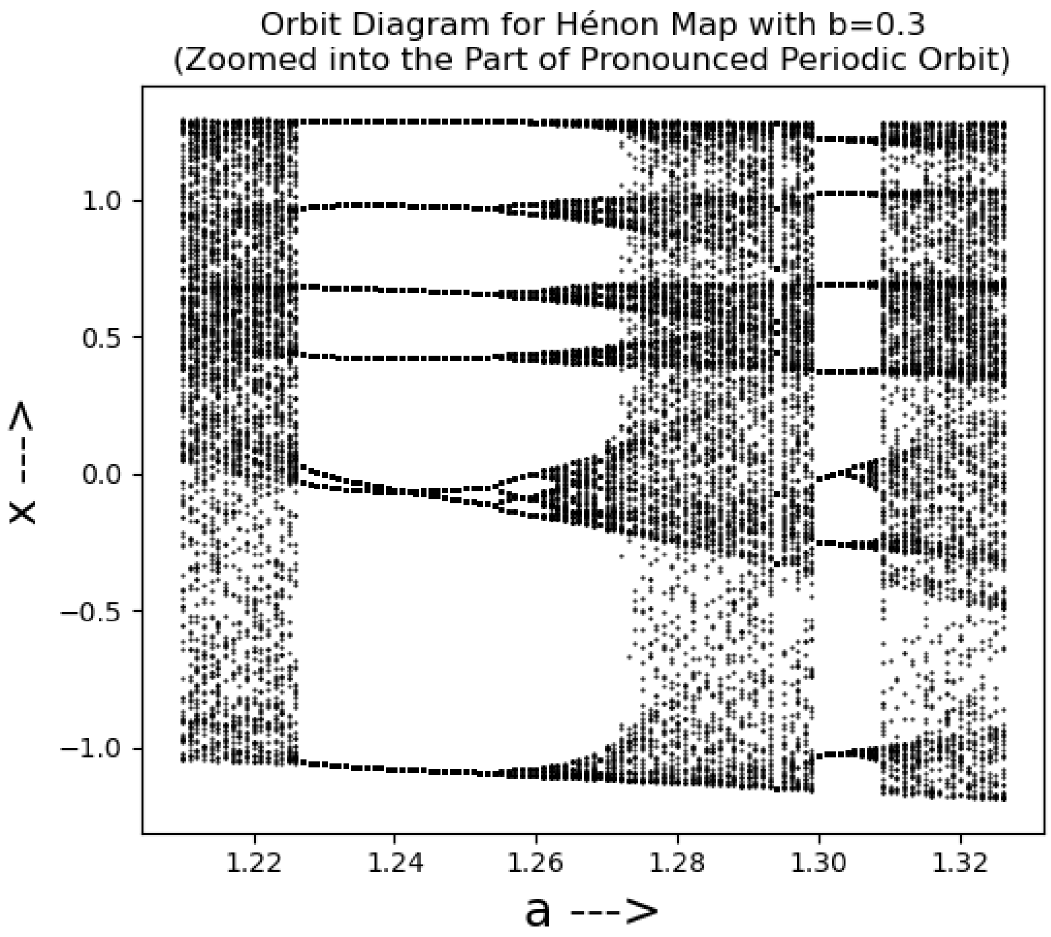
Below is the Semicontinuous Iteration of Henon Map

```
# importing necessary libraries
import numpy
import matplotlib.pyplot as plt

def henon_attractor(x,y,a,b): # define Henon Map
    xn=1-a*x**2+y
    yn=b*x
    return xn,yn

X=numpy.zeros(100000+1)
Y=numpy.zeros(100000+1)
# starting point (1, 1)
X[0]=1
Y[0]=1
dt=0.01
# semicontinuous iteration of Henon Map with time step 0.01
for I in range(100000):
    Xn,Yn=henon_attractor(X[I],Y[I],0.6,0.3)
    X[I+1]=X[I]+Xn*dt
    Y[I+1]=Y[I]+Yn*dt
plt.plot(X,Y,'.',ms=0.1,c='blue')
plt.xlabel('X --->',fontsize=14)
plt.ylabel('Y --->',fontsize=14)
plt.title('Semicontinuous Iteration of Hénon Map\n to Observe Period
Doubling',fontsize=11)
plt.show()
```



There's a periodic window which we want to address. So let's zoom into the part of the orbit diagram where the pronounced periodic window exists.

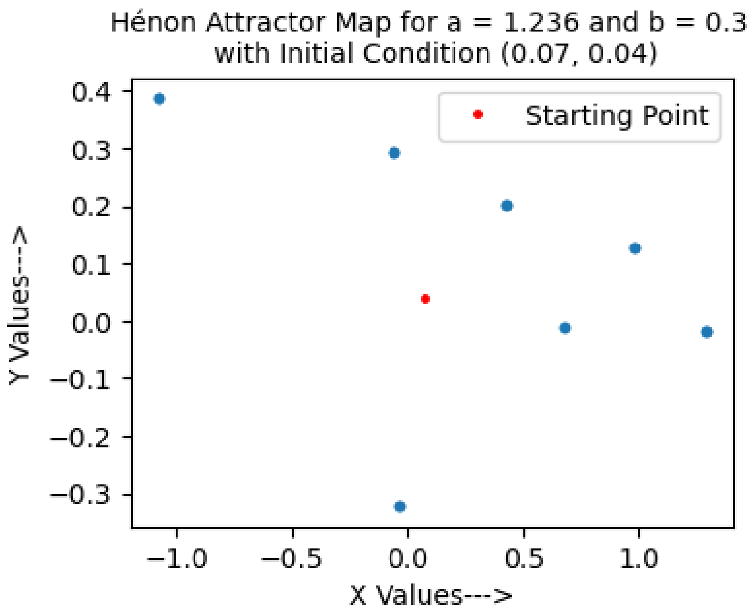
```python
b=0.3
def henon_attractor(x,y,a): # Henon map defining
    xn=1-a*x**2+y
    yn=b*x
    return xn,yn
# similar to the orbit diagram done previously
steps=50000
Points=250
def H(a):
    X=numpy.zeros(steps+1)
    Y=numpy.zeros(steps+1)
    X[0],Y[0]=0.07,0.04

    for i in range(steps):
        x_next,y_next=henon_attractor(X[i],Y[i],a)
        X[i+1]=x_next
        Y[i+1]=y_next
    return X[steps+1-Points:steps+1]

A=numpy.arange(1.21,1.327,0.001)
plt.figure(figsize=(6,5))
for a in A:
    avalues=a*numpy.ones(Points)
    Xv=H(a)
    plt.plot(avalues,Xv,'.',c='black',ms=1)
plt.xlabel('a --->',fontsize=18)
plt.ylabel('x --->',fontsize=17)
plt.title('Orbit Diagram for Hénon Map with b=0.3\n(Zoomed into the
Part of Pronounced Periodic Orbit)')
plt.show()
```



We want to capture the 7-periodic orbit that causes the emergence of periodic window. We shall plot the points where the initial point subjected to Henon Map settles down as Henon Map is applied on it iteratively.

```
# importing libraries
import numpy
import matplotlib.pyplot as plt

b,a=0.3,1.236 # we want to capture the 7-periodic orbit that causes
the emergence of periodic window
# defining Henon map
def henon_map(x,y):
    f=1-a*x**2+y
    g=b*x
    return f,g

x0,y0=0.07,0.04 # initial x coordinate 0.07 and initial y coordinate
0.04
xvalues,yvalues=[x0],[y0]
plt.figure(figsize=(4,3))
# plotting the starting point
```

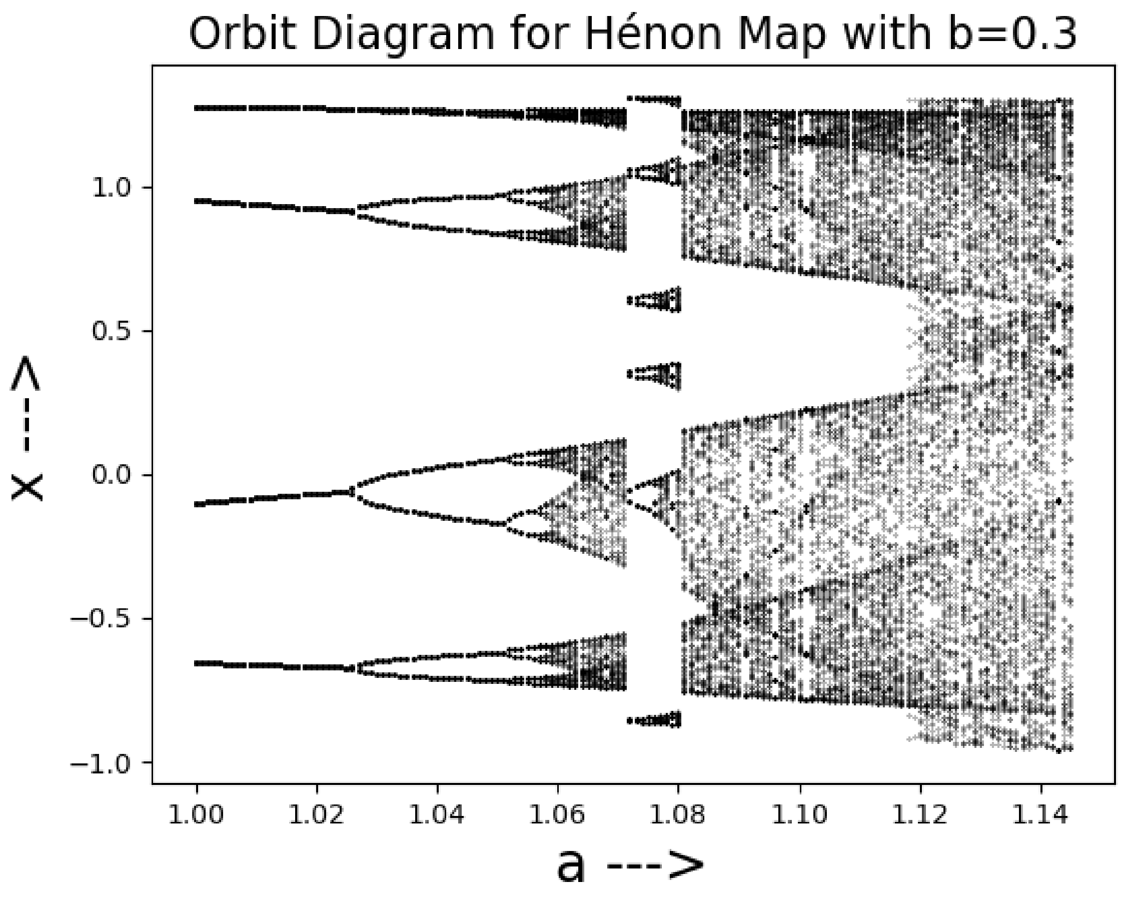
```python
plt.plot(xvalues,yvalues,'.',c='red',ms=5,label='Starting Point')
plt.legend()
iterations=1000 # the Henon map is applied 1000 times
for i in range(iterations):
    xvalues.append(henon_map(x0,y0)[0])
    yvalues.append(henon_map(x0,y0)[1])
    x0,y0=xvalues[i+1],yvalues[i+1]
# plotting of the 7-periodic fixed poits as the Henon map is applied
iteratively on (0.07, 0.04)
plt.plot(xvalues[-20:],yvalues[-20:],'.')
plt.xlabel('X Values--->')
plt.ylabel('Y Values--->')
plt.title('Hénon Attractor Map for a = 1.236 and b = 0.3 \n with
Initial Condition (0.07, 0.04)',fontsize=10)
plt.show()
```



In the orbit diagram, at around a = 1.075 a region of periodic and chaotic behaviour is noticed. Let's zoom into that region

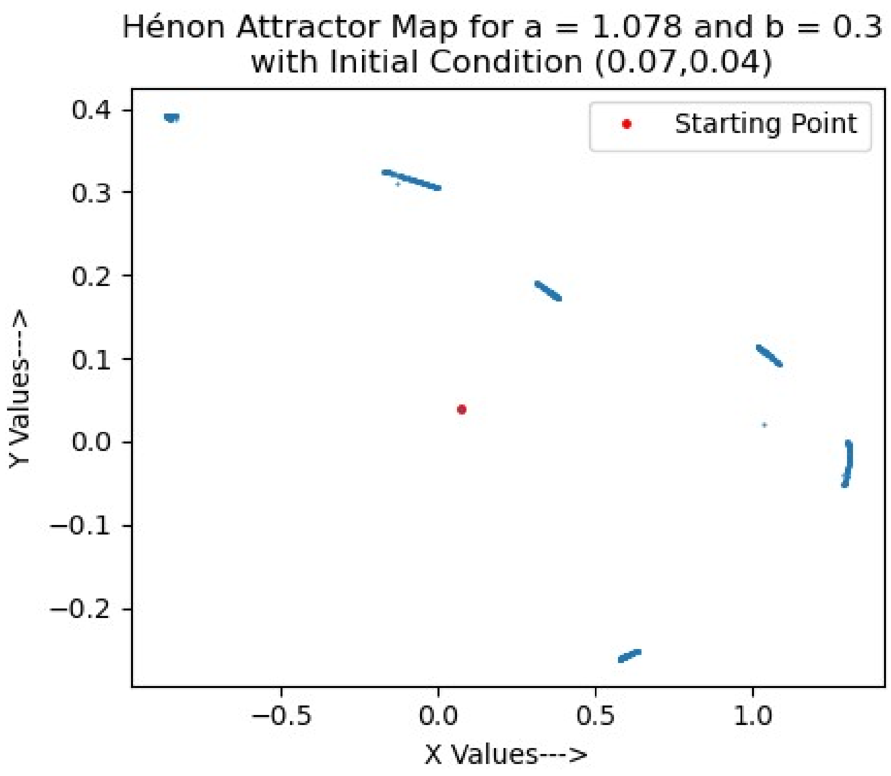
```python
# importing libraries
import numpy
import matplotlib.pyplot as plt

b=0.3 # parameter 'b' value is taken 0.3
def henon_attractor(x,y,a): # defining Henon map
    xn=1-a*x**2+y
    yn=b*x
    return xn,yn
# number of iteration is 50000
steps=50000
```

```
Points=250 # the last 250 points will be taken for each 'a' value to
draw graph

# below a function is defined which takes 'a' value and iterates 50000
times to return the last 250 values of iteration
def H(a):
    X=numpy.zeros(steps+1)
    Y=numpy.zeros(steps+1)
    X[0],Y[0]=0.07,0.04 # initial point of start

    for i in range(steps):
        x_next,y_next=henon_attractor(X[i],Y[i],a)
        X[i+1]=x_next
        Y[i+1]=y_next
    return X[steps+1-Points:steps+1]

A=numpy.arange(1,1.145,0.001) # this array stores the value of 'a'
parameter
for a in A:
    avalues=a*numpy.ones(Points)
    Xv=H(a)
    plt.plot(avalues,Xv,'.',c='black',ms=0.5) # plotting
plt.xlabel('a --->',fontsize=20)
plt.ylabel('x --->',fontsize=20)
plt.title('Orbit Diagram for Hénon Map with b=0.3',fontsize=16)
plt.show()
```



Henon attractor region in the chaotic and periodic zone ('a' value is 1.078)

```
b,a=0.3,1.078 # two parameters
def henon_map(x,y): # defining Henon map
    f=1-a*x**2+y
    g=b*x
    return f,g

x0,y0=0.07,0.04 # initial point (0.07, 0.04)
xvalues,yvalues=[x0],[y0]
plt.figure(figsize=(5,4))
# plotting the starting point
plt.plot(xvalues,yvalues,'.',c='red',ms=5,label='Starting Point')
plt.legend()
iterations=1000 # number of iteration 1000
for i in range(iterations):
    xvalues.append(henon_map(x0,y0)[0])
    yvalues.append(henon_map(x0,y0)[1])
    x0,y0=xvalues[i+1],yvalues[i+1]
plt.plot(xvalues,yvalues,'.',ms=1)
plt.xlabel('X Values--->')
plt.ylabel('Y Values--->')
plt.title('Hénon Attractor Map for a = 1.078 and b = 0.3 \n with
Initial Condition (0.07,0.04)')
plt.show()
```


Hénon Attractor Map for a = 1.078 and b = 0.3
with Initial Condition (0.07,0.04)
Starting Point
Y Values--->
X Values--->