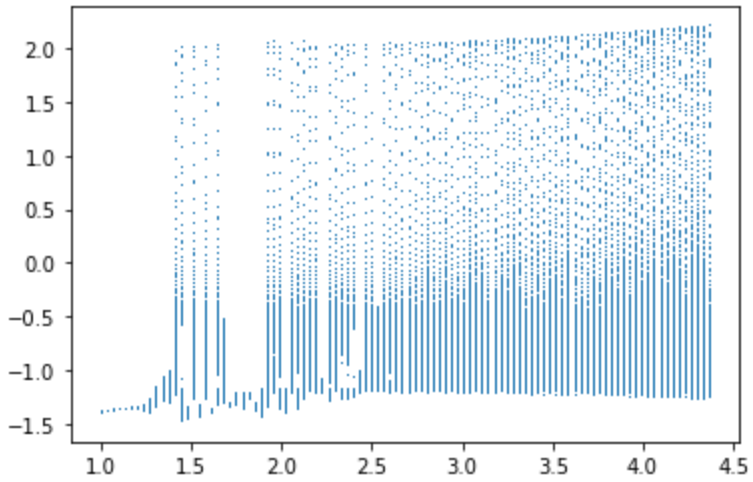
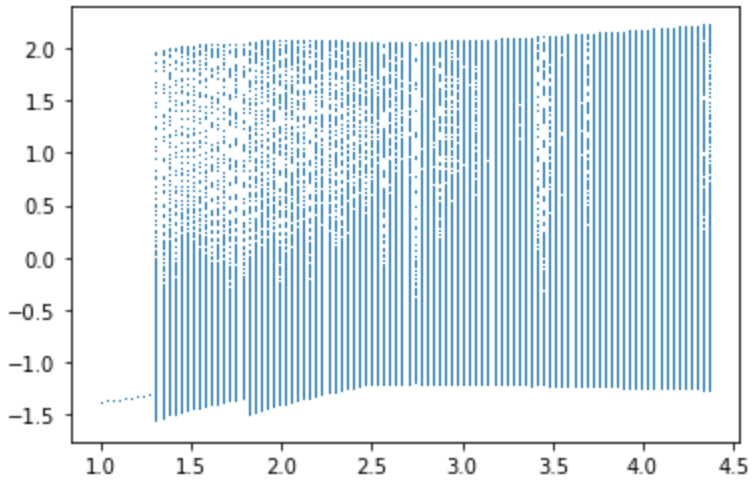
In [21]:
```
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.pyplot as plt
import math
a = 1.0
b = 3.0
c = 1.0
d = 5.0
s = 4.0
alf = - 1.6
r = 0.003
eps = 0.1
x = -0.7797753083855133
y = 0.1963653397308617
z = 0.43101317810390494
x1 = 0
y1 = 0
z1 = 0
ii = []
xx = []
def f1 (funx, funy, funz):
    return funy - a*funx*funx*funx + b*funx*funx - funz + par_i
def f2(funx, funy, funz):
    return c - d*funx*funx - funy
def f3(funx, funy, funz):
    return r*(s*(funx - alf) - funz)
i_min = 1.0
i_max = 4.4
hi = (i_max - i_min)/100
for ia in range(100):
    par_i = i_min + ia*hi
    for j in range(1, 4000):
        x1 = x + eps*f1(x,y,z)
        y1 = y + eps*f2(x,y,z)
        z1 = z + eps*f3(x,y,z)
        x=x1
        y=y1
        z=z1
        if j > 3500:
            ii.append(par_i)
            xx.append(x)
plt.plot(ii, xx, ',')
plt.show()
```



In [22]:
```
import numpy as np
import matplotlib.pyplot as plt
```

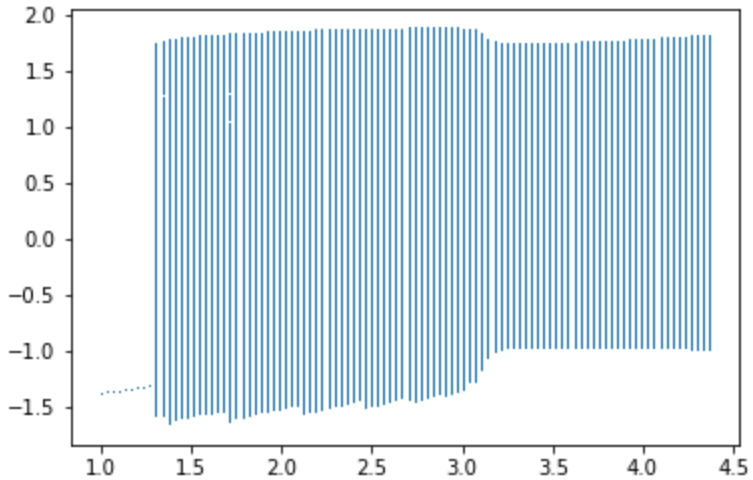
```
import matplotlib.pyplot as plt
import math
a = 1.0
b = 3.0
c = 1.0
d = 5.0
s = 4.0
alf = - 1.6
r = 0.003
eps = 0.1
x = -0.7797753083855133
y = 0.1963653397308617
z = 0.43101317810390494
x1 = 0
y1 = 0
z1 = 0
ii = []
xx = []
def f1 (funx, funy, funz):
    return funy - a*funx*funx*funx + b*funx*funx - funz + par_i
def f2(funx, funy, funz):
    return c - d*funx*funx - funy
def f3(funx, funy, funz):
    return r*(s*(funx - alf) - funz)
i_min = 1.0
i_max = 4.4
hi = (i_max - i_min)/100
for ia in range(100):
    par_i = i_min + ia*hi
    for j in range(1, 90000):
        x1 = x + eps*f1(x,y,z)
        y1 = y + eps*f2(x,y,z)
        z1 = z + eps*f3(x,y,z)
        x=x1
        y=y1
        z=z1
        if j > 85000:
            ii.append(par_i)
            xx.append(x)
plt.plot(ii, xx, ',')
plt.show()
```



In [19]:

```
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.pyplot as plt
import math
a = 1.0
b = 3.0
```

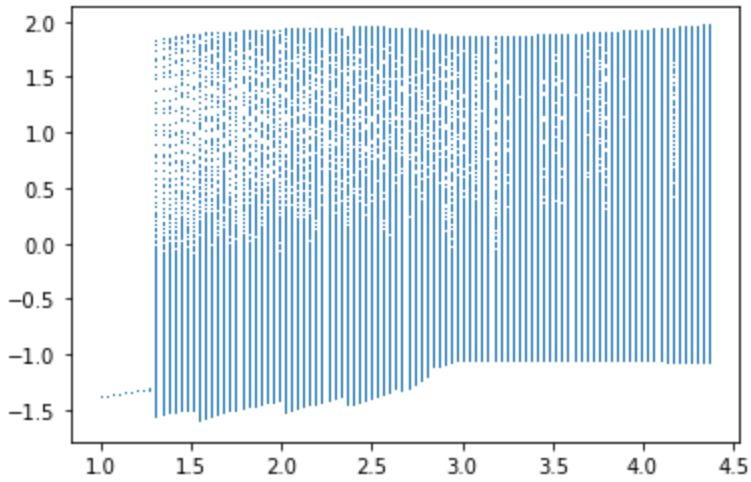
```
c = 1.0
d = 5.0
s = 4.0
alf = - 1.6
r = 0.003
eps = 0.02
x = -0.7797753083855133
y = 0.1963653397308617
z = 0.43101317810390494
x1 = 0
y1 = 0
z1 = 0
ii = []
xx = []
def f1 (funx, funy, funz):
    return funy - a*funx*funx*funx + b*funx*funx - funz + par_i
def f2(funx, funy, funz):
    return c - d*funx*funx - funy
def f3(funx, funy, funz):
    return r*(s*(funx - alf) - funz)
i_min = 1.0
i_max = 4.4
hi = (i_max - i_min)/100
for ia in range(100):
    par_i = i_min + ia*hi
    for j in range(1, 400000):
        x1 = x + eps*f1(x,y,z)
        y1 = y + eps*f2(x,y,z)
        z1 = z + eps*f3(x,y,z)
        x=x1
        y=y1
        z=z1
        if j > 350000:
            ii.append(par_i)
            xx.append(x)
plt.plot(ii, xx, ',')
plt.show()
```



In [23]:

```
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.pyplot as plt
import math
a = 1.0
b = 3.0
c = 1.0
d = 5.0
s = 4.0
alf = - 1.6
```

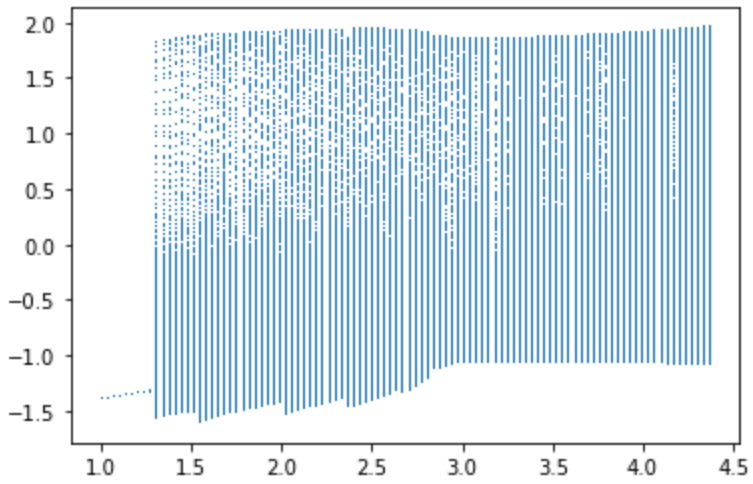
```
r = 0.003
eps = 0.05
x = -0.7797753083855133
y = 0.1963653397308617
z = 0.43101317810390494
x1 = 0
y1 = 0
z1 = 0
ii = []
xx = []
def f1 (funx, funy, funz):
    return funy - a*funx*funx*funx + b*funx*funx - funz + par_i
def f2(funx, funy, funz):
    return c - d*funx*funx - funy
def f3(funx, funy, funz):
    return r*(s*(funx - alf) - funz)
i_min = 1.0
i_max = 4.4
hi = (i_max - i_min)/100
for ia in range(100):
    par_i = i_min + ia*hi
    for j in range(1, 40000):
        x1 = x + eps*f1(x,y,z)
        y1 = y + eps*f2(x,y,z)
        z1 = z + eps*f3(x,y,z)
        x=x1
        y=y1
        z=z1
        if j > 35000:
            ii.append(par_i)
            xx.append(x)
plt.plot(ii, xx, ',')
plt.show()
```



In [20]:

```
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.pyplot as plt
import math
a = 1.0
b = 3.0
c = 1.0
d = 5.0
s = 4.0
alf = - 1.6
r = 0.003
eps = 0.05
x = -0.7797753083855133
y = 0.1963653397308617
```

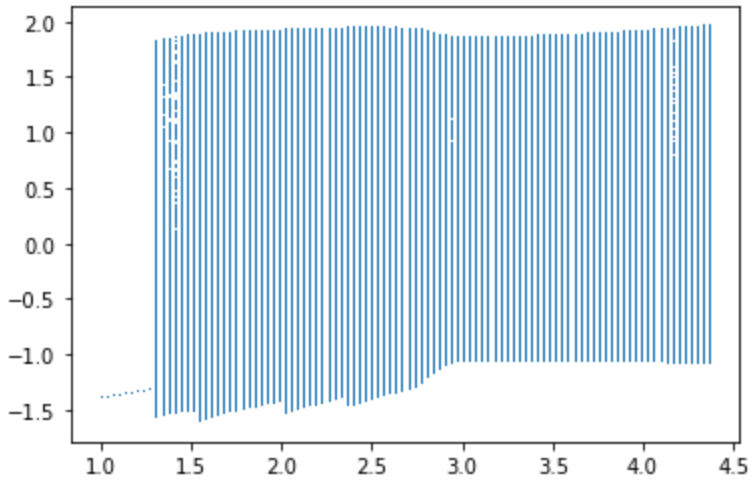
```
z = 0.43101317810390494
x1 = 0
y1 = 0
z1 = 0
ii = []
xx = []
def f1 (funx, funy, funz):
    return funy - a*funx*funx*funx + b*funx*funx - funz + par_i
def f2(funx, funy, funz):
    return c - d*funx*funx - funy
def f3(funx, funy, funz):
    return r*(s*(funx - alf) - funz)
i_min = 1.0
i_max = 4.4
hi = (i_max - i_min)/100
for ia in range(100):
    par_i = i_min + ia*hi
    for j in range(1, 40000):
        x1 = x + eps*f1(x,y,z)
        y1 = y + eps*f2(x,y,z)
        z1 = z + eps*f3(x,y,z)
        x=x1
        y=y1
        z=z1
        if j > 35000:
            ii.append(par_i)
            xx.append(x)
plt.plot(ii, xx, ',')
plt.show()
```



In [24]:

```
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.pyplot as plt
import math
a = 1.0
b = 3.0
c = 1.0
d = 5.0
s = 4.0
alf = - 1.6
r = 0.003
eps = 0.05
x = -0.7797753083855133
y = 0.1963653397308617
z = 0.43101317810390494
x1 = 0
y1 = 0
z1 = 0
```

```
ii = []
xx = []
def f1 (funx, funy, funz):
    return funy - a*funx*funx*funx + b*funx*funx - funz + par_i
def f2(funx, funy, funz):
    return c - d*funx*funx - funy
def f3(funx, funy, funz):
    return r*(s*(funx - alf) - funz)
i_min = 1.0
i_max = 4.4
hi = (i_max - i_min)/100
for ia in range(100):
    par_i = i_min + ia*hi
    for j in range(1, 400000):
        x1 = x + eps*f1(x,y,z)
        y1 = y + eps*f2(x,y,z)
        z1 = z + eps*f3(x,y,z)
        x=x1
        y=y1
        z=z1
        if j > 350000:
            ii.append(par_i)
            xx.append(x)
plt.plot(ii, xx, ',')
plt.show()
```



In [ ]: