

▼ Cobwebbing of first order difference equations

This program takes as input a first-order recurrence relation and an initial condition, and draws out an appropriate cobwebbing diagram.

```
import matplotlib.pyplot as plt
import numpy as np

def f(x):
    return -x**2/3 + 3

fig, ax = plt.subplots()
ax.set_xlim(0,5)
ax.set_ylim(0,5)
ax.set_xlabel("X-axis")
ax.set_ylabel("Y-axis")

ax.plot(np.arange(100)/10, np.arange(100)/10)
ax.plot(np.arange(100)/10, [f(x) for x in np.arange(100)/10])

num_iter = 8
x_0 = 1
point_list = []
point_list.append((x_0, 0))

for i in range(num_iter):
    x, y = point_list[-1]
    y = f(x)
    point_list.append((x, y))
    x = y
    point_list.append((x, y))
point_list
point_array = np.array(point_list)
X, Y = point_array.T
ax.plot(X, Y)

--INSERT--
```

[<matplotlib.lines.Line2D at 0x7f3c6e548470>]



