Fathan Ananta Nur
236-D8749

44135 Applied Information Theory
Assignment 1 (Revision)

In these experiments, we will use the parameters below:

```
Length (N) = 1000000
Initial value = {0.623521, 0.623522} //set the slight differences on those values
to observe the sensitivity of initial values
Parameter c = {0.2, 0.3, 0.4, 0.5} //0.5 value is used to observe bit-shift problem
Iteration n = 60
```

1. Generate chaotic real-valued sequences by the following chaotic maps and draw the orbits $0 \leq n \leq 60$. Also, confirm the sensitive dependence on initial conditions by slightly changing the initial values.

1.1. Logistic map (p.7)

On logistic map experiment, initial value 0.623521 and 0.623522 will be set, and the differences will be compared. Parameter c is not used on this map, so there is no possibility on bit-shift problem as the graph plot will use the same function f(x).

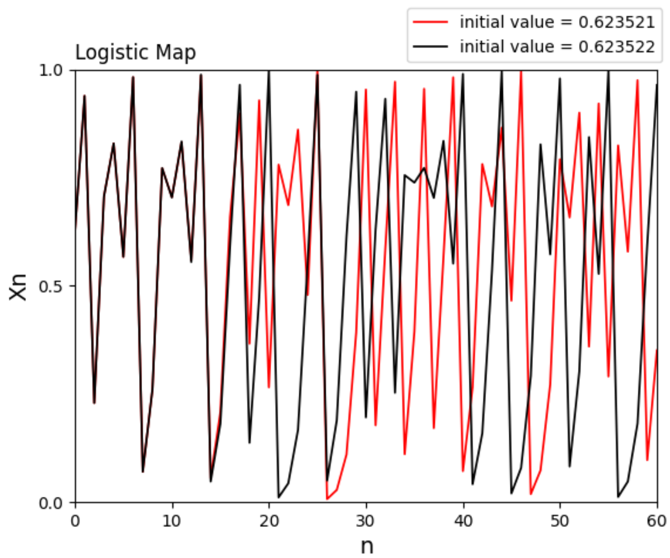


Figure 1. Logistic Map with Initial Values 0.623521 and 0.623522

On Figure 1, we can see that the graph from those initial values will differ significantly on around n = 18. It can be happened because of the differences of initial value, even the changes are very slightly different (0.623521 and 0.623522). This is the proof of "chaos theory" (or "butterfly effect") that the sensitive dependence on initial conditions in which a small change in one state of a deterministic nonlinear system can result in large differences in a later state.

1.2. Skew Bernoulli map (p.13) for some values of c (Confirm "bit-shift problem" for $c = 0.5$)

On Skew Bernoulli map, not only initial values that will be set, but also parameter c to observe the more significant difference on graph and bit-shift problem. We will use the same initial values as used on logistic map and parameter c 0.2, 0.3, 0.4, and 0.5.

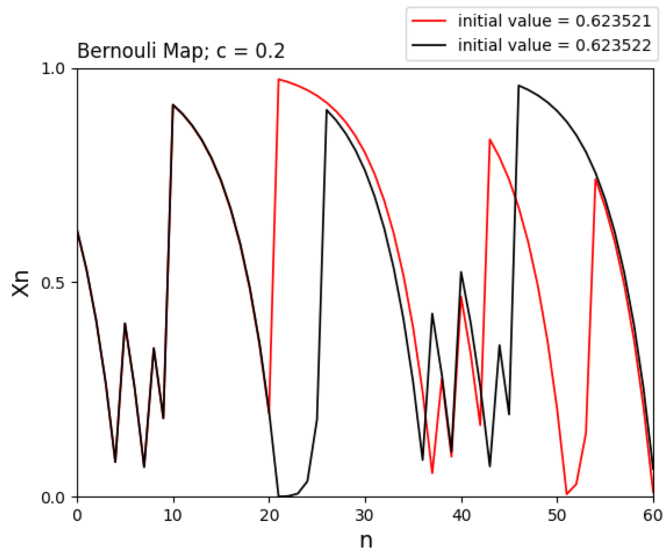


Figure 2. Bernoulli Map Parameter c = 0.2, with Initial Values 0.623521 and 0.623522

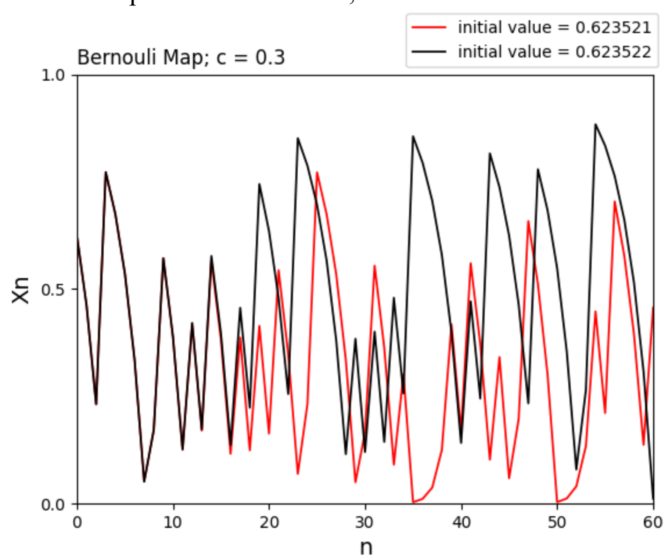


Figure 3. Bernoulli Map Parameter c = 0.3, with Initial Values 0.623521 and 0.623522

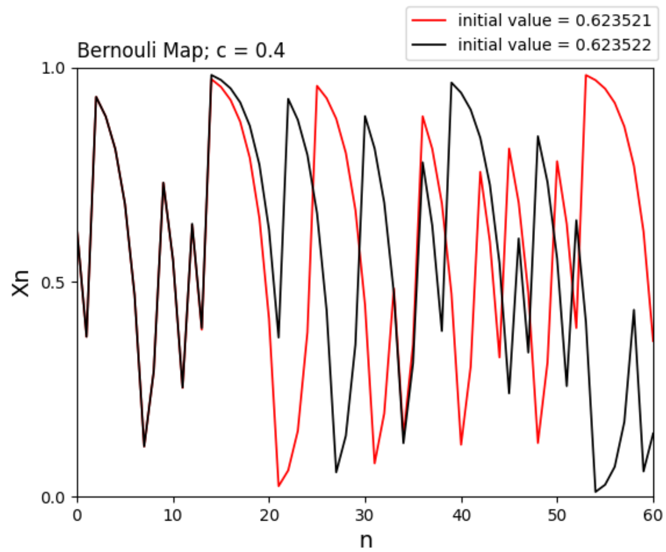


Figure 4. Bernoulli Map Parameter c = 0.4, with Initial Values 0.623521 and 0.623522

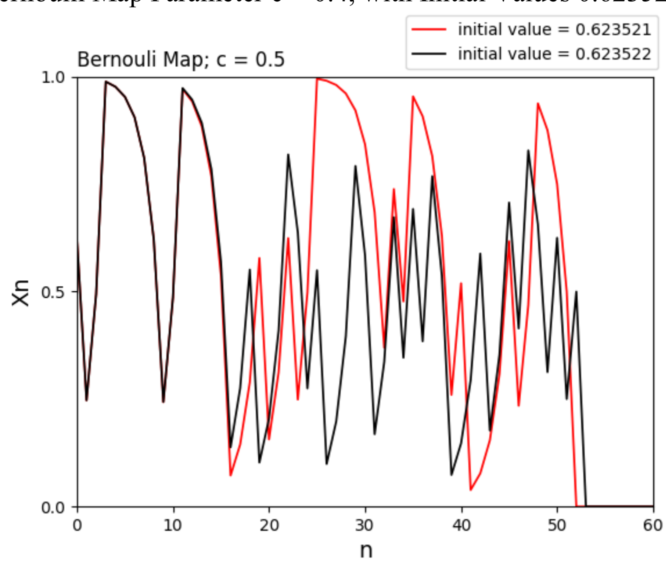


Figure 5. Bernoulli Map Parameter c = 0.5, with Initial Values 0.623521 and 0.623522

Figure 2, Figure 3, Figure 4, and Figure 5 show the graph plot using parameter c 0.2, 0.3, 0.4, and 0.5 respectively. Each graph uses the slightly different initial values resulting in different graph plot starting on around n = 10. On Figure 3, we can observe **the bit-shift problem on around n = 50** as both lines will convergent to Xn = 0. The bit-shift problem occurs because the Bernoulli map with c = 0.5 causes values to be doubled, which is a bit-shift operation in binary representation. This can lead to numerical instability and precision issues, especially for values near critical points like 0.5 (which is 0.4999…99 and 0.500…01)

2. Compute the distributions of the real-valued sequences of the above chaotic maps (1-1) and (1-2). For skew Bernoulli maps, confirm that they are uniform for every value of *c*

2.1. Logistic map

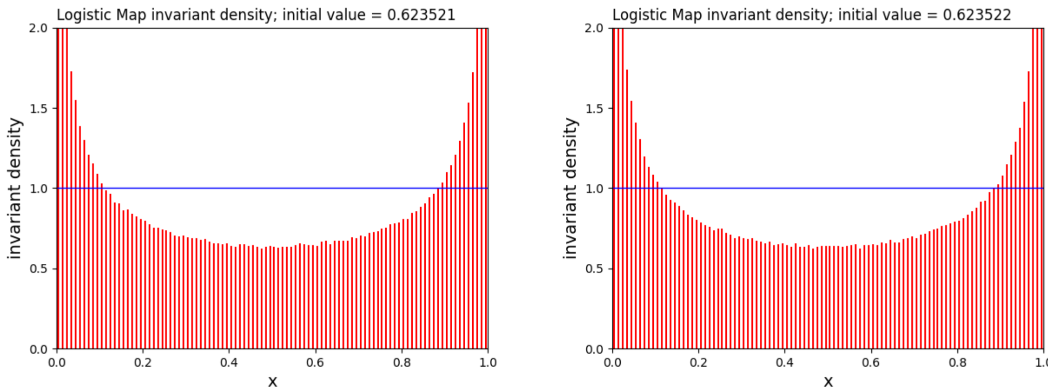
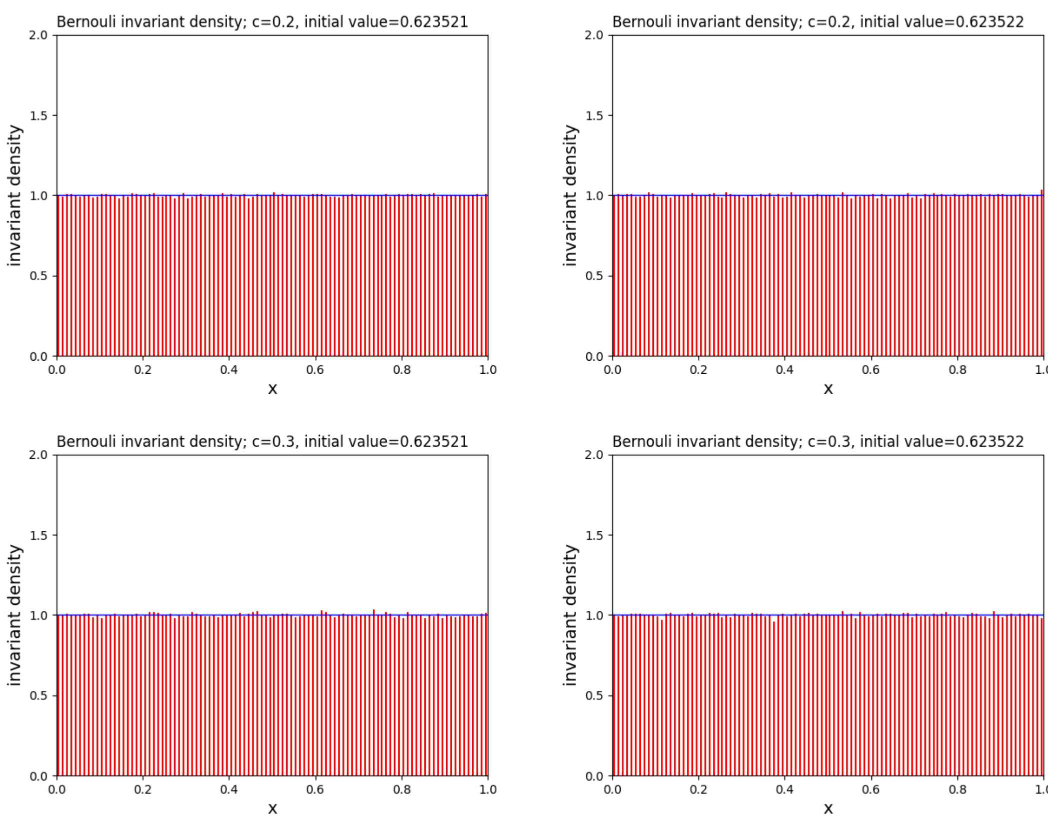


Figure 6. Logistic Map Invariant Density with Initial Values 0.623521 and 0.623522

Figure 6. shows us the distributions of the real-valued sequences on logistic map with the slight difference on initial values. The distribution of the real number sequence on both graph is roughly similar. The distribution is called non-uniform distribution as the value not always nearly same. This means that the probabilities of sequence x assigned to different outcomes are not the same. From that fact, we can believe that some outcomes are more likely to occur than others.

2.2. Bernoulli map



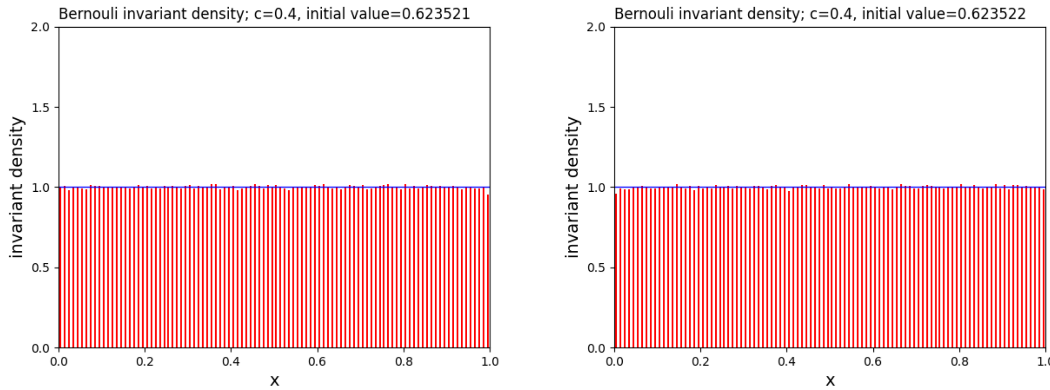


Figure 7. Bernoulli Map (Parameter c = 0.2, 0.3, and 0.4) Invariant Density with Initial Values 0.623521 and 0.623522

Figure 7. shows us the distributions of the real-valued sequences on bernoulli map with parameter c = 0.2, 0.3, and 0.4 with the slight difference on initial values. The distribution of the real number sequence on both graph is roughly similar. The distribution is called uniform distribution, as we can see that every sequence x is always approaching 1 on invarient density value.

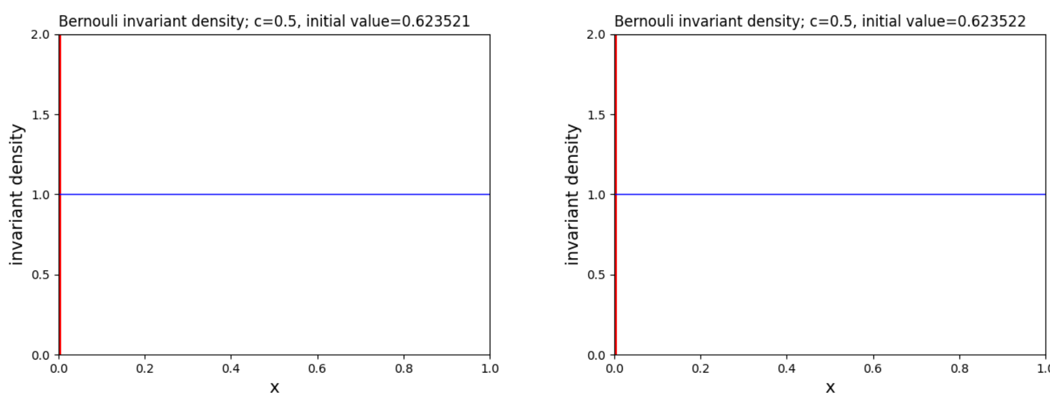


Figure 8. Bernoulli Map (Parameter c = 0.5) Invariant Density with Initial Values 0.59999 and 0.6

Figure 8. shows us that the graph is failed to be plotted by matplotlib.pyplot because of bit-shift error.

Appendix

1. Logistic Map Function (p.7)

$$\tau(x) = 4x(1-x)$$

2. Bernoulli Map Function (p.13)

$$\tau(x,c) = \begin{cases} \dfrac{x}{c} & 0 \le x < c \\ \dfrac{x-c}{1-c} & c \le x \le 1 \end{cases}$$

3. Source code

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

# Variables
c_params = [0.2, 0.3, 0.4, 0.5] # parameter c
ivs = [0.623521, 0.623522] # initial values
l = 1000000 # length (N)
n = 60 # iteration
x = np.zeros((len(ivs), l)) # sequence for bernouli
y = np.zeros((len(ivs), l)) # sequence for logistic

def skew_bernouli_map(x, c): # Bernoulli function
    if x < c:
        return (x/c)
    else:
        return (x-c)/(1-c)

def logistic_map(x): # Logistic function
    return 4*x*(1-x)

def main():
    for c in c_params:
        for idx, iv in enumerate(ivs):
            x[idx, 0] = iv
            for i in range(1, x.shape[1]):
                x[idx, i] = skew_bernouli_map(x[idx, i-1], c)

        # 1-2 Bernoulli Skew Map
        plt.figure()
        plt.plot(x[0, :n+1], color='r', label=f"initial value = {x[0, 0]}", linewidth=1.25)
        plt.plot(x[1, :n+1], color='k', label=f"initial value = {x[1, 0]}", linewidth=1.25)
        plt.legend(loc='upper center', bbox_to_anchor=(0.79, 1.16))
        plt.xlim(0, n)
        plt.ylim(0, 1)
```

```python
    plt.yticks([0, 0.5, 1])
    plt.xlabel("n", fontsize=14)
    plt.ylabel("Xn", fontsize=14)
    plt.title(f'Bernouli Map; c = {c}', loc = "left")
    plt.savefig(f"result/1-2_bernouli_{c}.png")

    # 2-2 Bernoulli Invariant
    for idx, iv in enumerate(ivs):
        plt.figure()
        plt.hist(x[idx, :], bins=100, rwidth=0.4, color='r', density=True)
        plt.xlim(0, 1)
        plt.ylim(0, 2)
        plt.yticks([0, 0.5, 1, 1.5, 2])
        plt.hlines(1, 0, 1, color='b', linewidth=1)
        plt.xlabel("x", fontsize=14)
        plt.ylabel("invariant density", fontsize=14)
        plt.title(f'Bernouli invariant density; c={c}, initial value={iv}', loc = "left")
        plt.savefig(f"result/2-2_bernouli_{c}_{iv}.png")

for idx, iv in enumerate(ivs):
    y[idx, 0] = iv
    for i in range(1, x.shape[1]):
        y[idx, i] = logistic_map(y[idx, i-1])

    # 1-1 Logistic Map
    plt.figure()
    plt.plot(y[0, :n+1], color='r', label=f"initial value = {y[0, 0]}", linewidth=1.25)
    plt.plot(y[1, :n+1], color='k', label=f"initial value = {y[1, 0]}", linewidth=1.25)
    plt.legend(loc='upper center', bbox_to_anchor=(0.79, 1.16))
    plt.xlim(0, n)
    plt.ylim(0, 1)
    plt.yticks([0, 0.5, 1])
    plt.xlabel("n", fontsize=14)
    plt.ylabel("Xn", fontsize=14)
    plt.title(f'Logistic Map', loc = "left")
    plt.savefig(f"result/1-1_logistic.png")

    # 2-1 Logistic Invarant
    for idx, iv in enumerate(ivs):
        plt.figure()
        plt.hist(y[idx, :], bins=100, rwidth=0.4, color='r', density=True)
        plt.xlim(0, 1)
        plt.ylim(0, 2)
        plt.yticks([0, 0.5, 1, 1.5, 2])
        plt.hlines(1, 0, 1, color='b', linewidth=1)
        plt.xlabel("x", fontsize=14)
        plt.ylabel("invariant density", fontsize=14)
        plt.title(f'Logistic Map invariant density; initial value = {iv}', loc = "left")
```

```
            plt.savefig(f"result/2-1_logistic_{iv}.png")

if __name__ == "__main__":
    os.makedirs('result', exist_ok=True)
    main()
```