M7001_quiz4_24000114067

1 Quiz 4

LIS MASc
Engaging Complexity

Access this notebook on GitHub
PDF generated using nbconvert

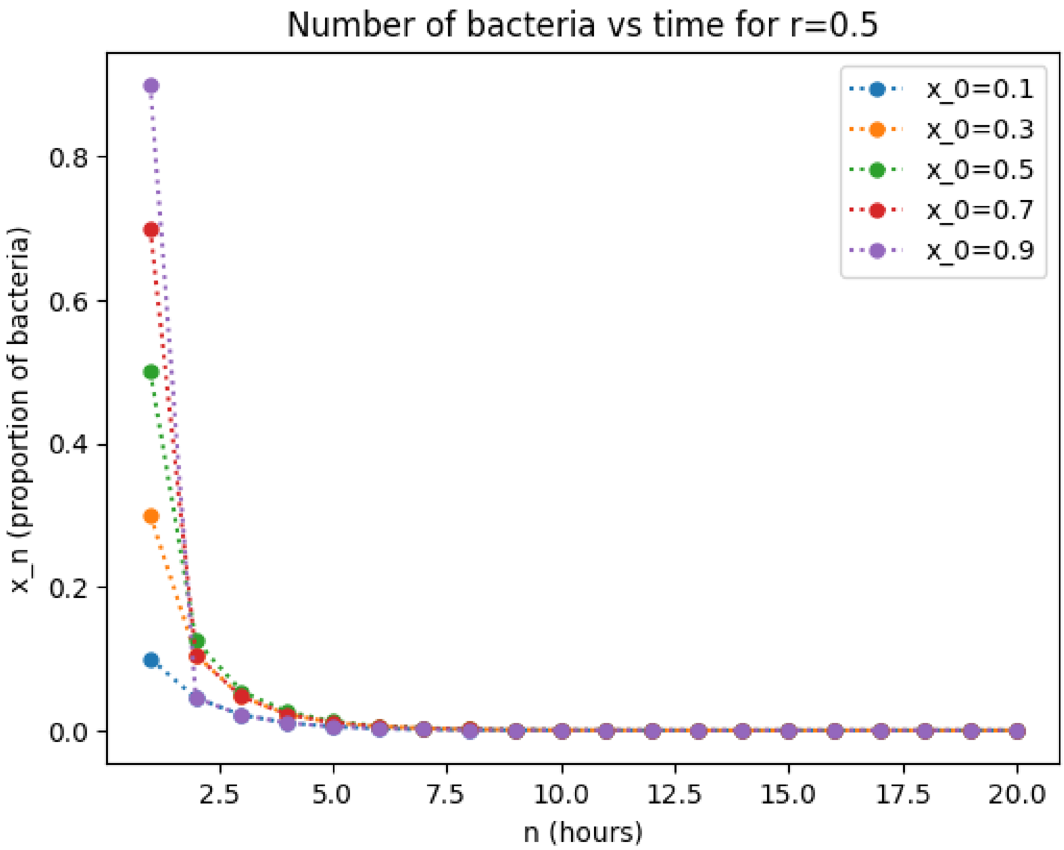
[1]:
```
import numpy as np
import matplotlib.pyplot as plt
```

1.1 1

1.1.1 a)

[2]:
```
def logistic_map(r, max_n, x0_vector):
    n = np.arange(1,max_n+1)
    xn = np.zeros(max_n)

    for x0 in x0_vector:
        xn[0] = x0

        for i in np.arange(0,max_n-1):
            xn[i+1] = r * xn[i] * (1-xn[i])

        plt.plot(n, xn, 'o:', markersize=5, label='x_0='+str(x0))


    plt.title('Number of bacteria vs time for r='+str(r))
    plt.ylabel('x_n (proportion of bacteria)')
    plt.xlabel('n (hours)')
    plt.legend();
```

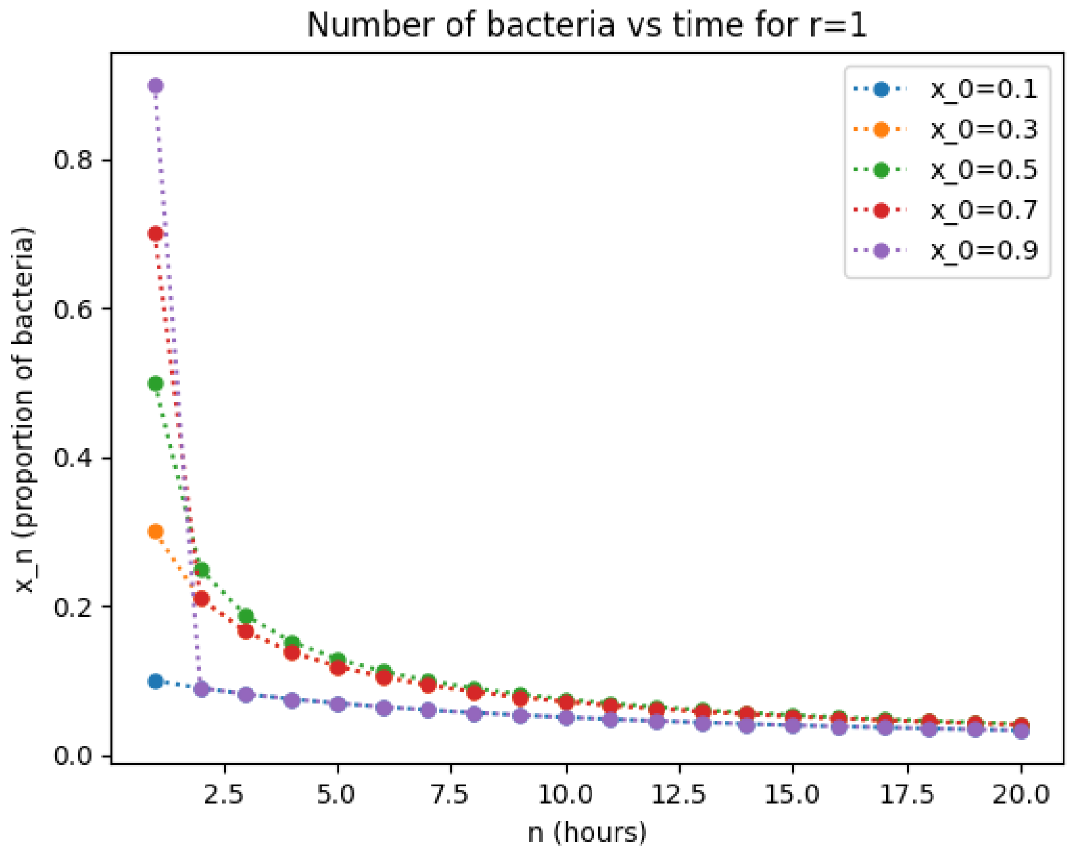
[3]:
```
logistic_map(r=0.5, max_n=20, x0_vector=[0.1,0.3,0.5,0.7,0.9])
```



1.1.2 (b)

[4]:

```
logistic_map(r=1, max_n=20, x0_vector=[0.1,0.3,0.5,0.7,0.9])
```



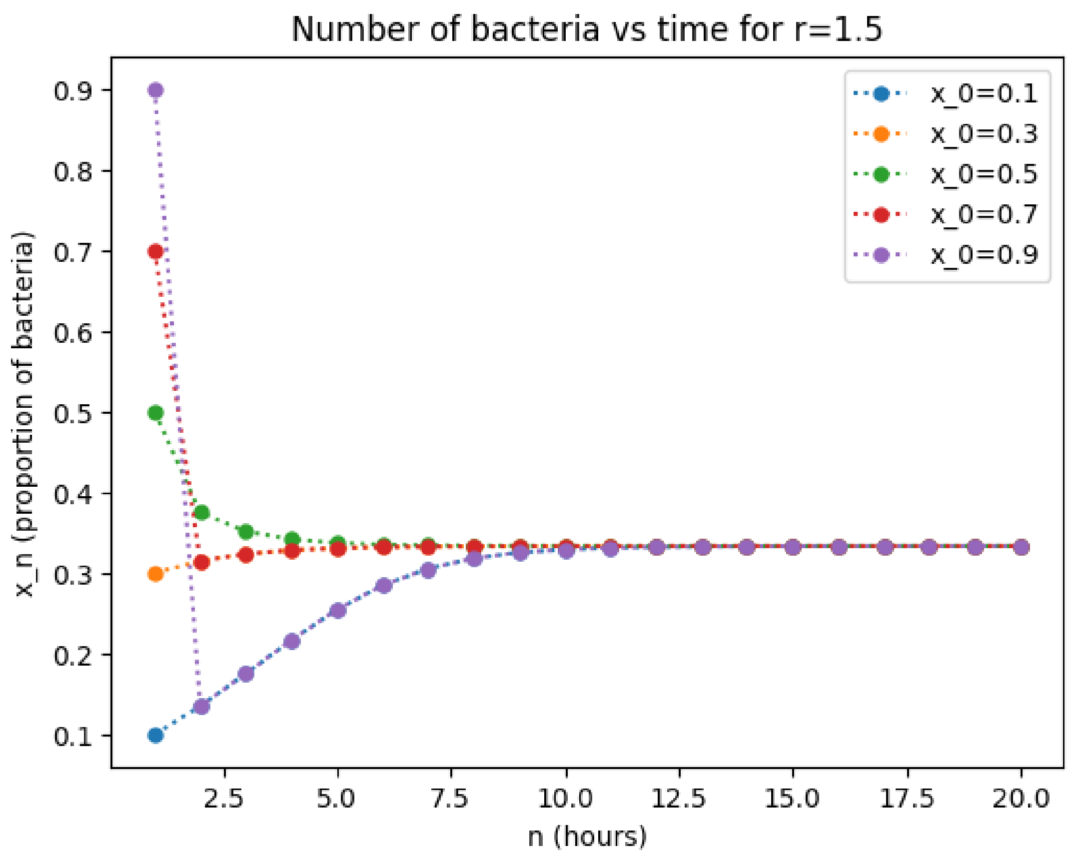
1.1.3 (c)

As n increases, all values of x_n converge independently from initial conditions (set by the x_0 vector). In this case, they approach zero - the only equilibrium point. This is notable because, given enough time, the behaviour is independent of x_0, depending only on r.

1.2 2

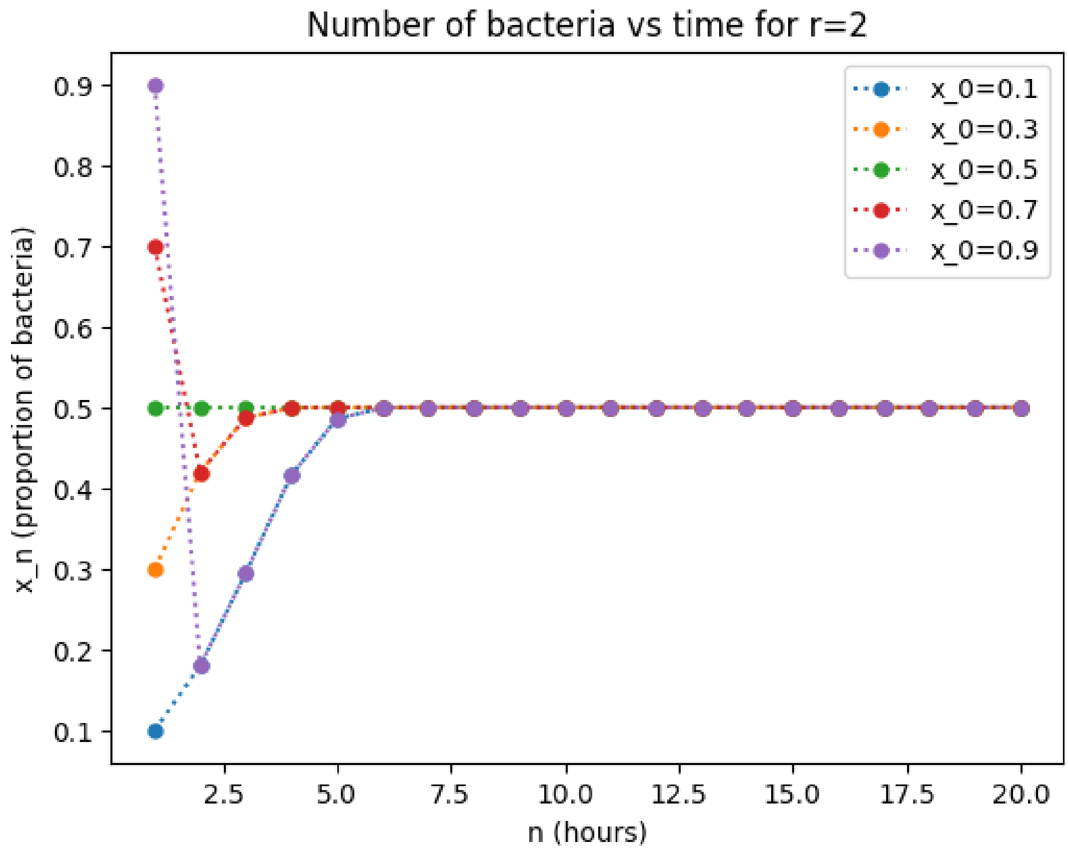
1.2.1 (a)

[5]:
```
logistic_map(r=1.5, max_n=20, x0_vector=[0.1,0.3,0.5,0.7,0.9])
```

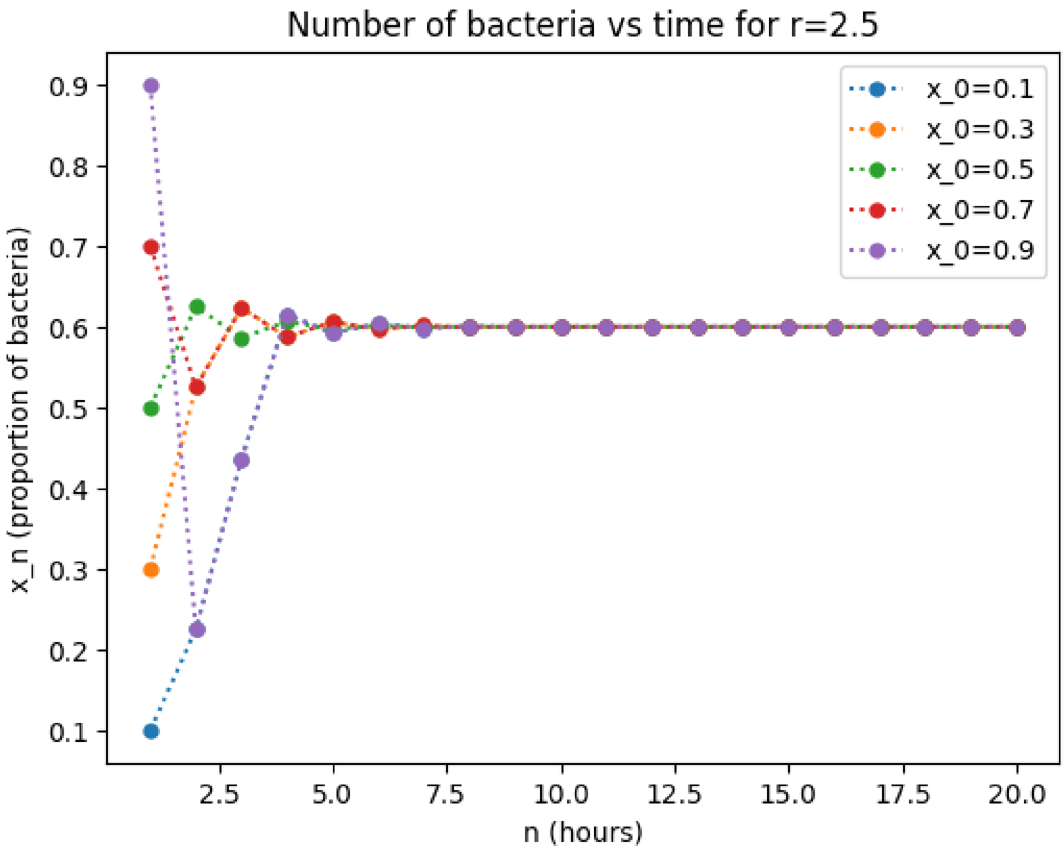


[6]:

```
logistic_map(r=2, max_n=20, x0_vector=[0.1,0.3,0.5,0.7,0.9])
```



[7]:
```
logistic_map(r=2.5, max_n=20, x0_vector=[0.1,0.3,0.5,0.7,0.9])
```



1.2.2 (b)

As n increases, all values of x_n converge independently from initial conditions (set by the x_0 vector). In these cases, they each approach one equilibrium point (approximately 0.33, 0.5 and 0.6, respectively). This is notable because, given enough time, the behaviour is independent of x_0, depending only on r.

1.2.3 (c)

$x_{n+1} = r \cdot x_n(1 - x_n)$

The equilibrium point happens when the current and previous iteration achieve the same value. This can be represented as $x_{n+1} = x_n = x$

Considering this,

$x = r \cdot x(1 - x)$

$-rx^2 + rx - x = 0$

$-rx^2 + rx - x = 0$

$-x(rx - r + 1) = 0$

So either $-x = 0$ or $rx - r + 1$.

Solving for x:

$-x = 0$

$$x = 0$$

or

$rx - r + 1 = 0$

$rx = r - 1$

$$x = \frac{r-1}{r}$$

The equilibrium points for the given equation are, therefore,

$$x = \boxed{0} \quad \text{or} \quad x = \boxed{\frac{r-1}{r}}$$

1.2.4 (d)

Given $x = \frac{r-1}{r}$, we can deduce that

- for values of r between 0 and 1, x will be negative. This means that the population will have a negative growth rate, decreasing until it stabilizes at zero.
- for values of r between 1 and 2, x will be between 0 and 0.5.

Calculating the equilibrium points for the previous solutions, we can see that they follow this exact behaviour:

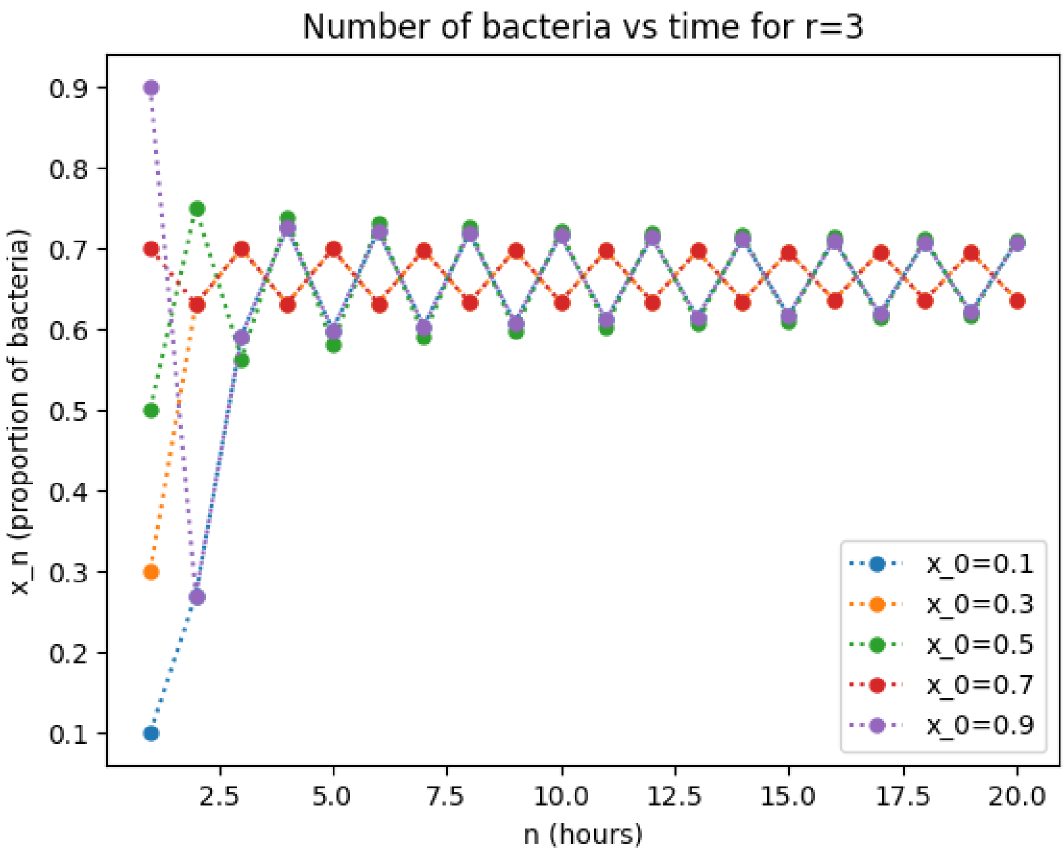
[8]:
```
def logistic_equilibria(r):
    return (r - 1) / r

rs = [0.5, 1, 1.5, 2, 2.5]

for r in rs:
    print(f'if r = {r}:\tthe nonzero equilibrium point is␣
 ↪{logistic_equilibria(r)}')
```

```
if r = 0.5:     the nonzero equilibrium point is -1.0
if r = 1:       the nonzero equilibrium point is 0.0
if r = 1.5:     the nonzero equilibrium point is 0.3333333333333333
if r = 2:       the nonzero equilibrium point is 0.5
if r = 2.5:     the nonzero equilibrium point is 0.6
```

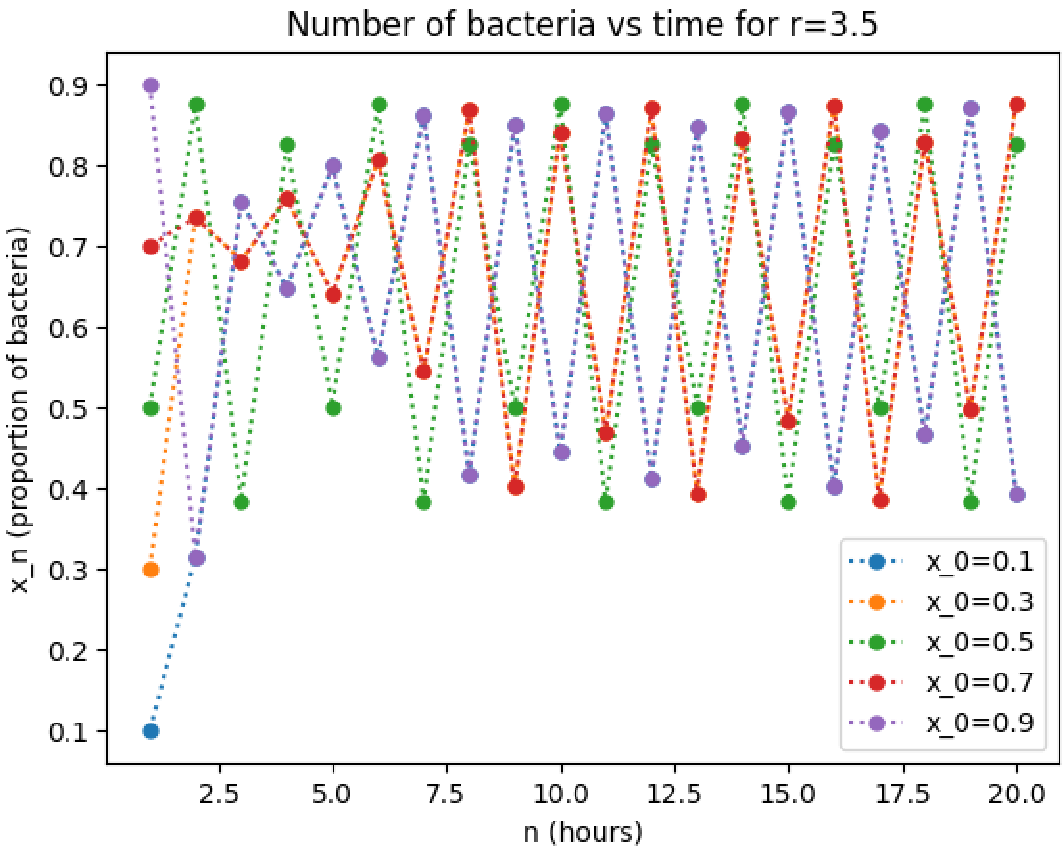
1.3 3

1.3.1 (a)

[9]:
```
logistic_map(r=3, max_n=20, x0_vector=[0.1,0.3,0.5,0.7,0.9])
```



[10]:
```
logistic_map(r=3.5, max_n=20, x0_vector=[0.1,0.3,0.5,0.7,0.9])
```



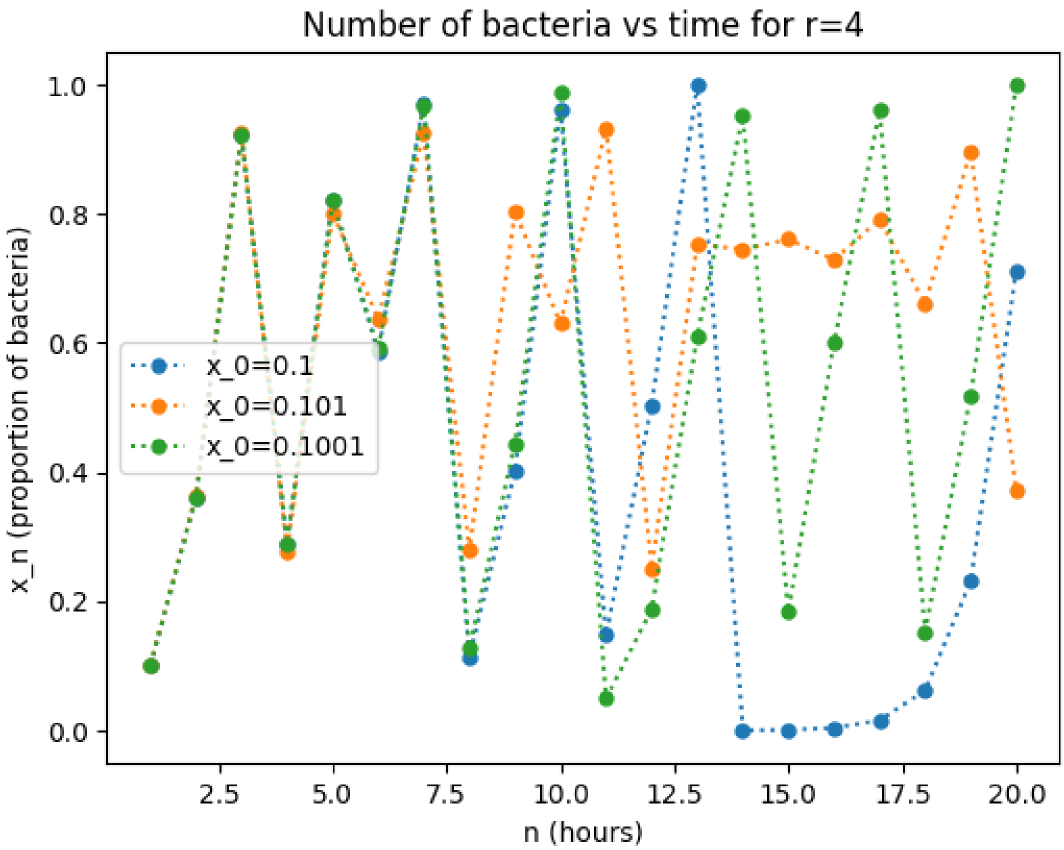
1.3.2 (b)

As n increases, values of x_n enter into a state of oscillation. Depending on the value of r, this oscillation has different numbers of attraction points (in the previous cases, 2 and 4).

1.4 4

1.4.1 (a)

[11]:
```
logistic_map(r=4, max_n=20, x0_vector=[0.1,0.101,0.1001])
```



1.4.2 (b)

With $r = 4$, the behaviour of x does not converge anymore. Instead, it is chaotic. This means that it is very sensitive to initial conditions (x_n).

1.5 5

Between 0 and 1, x_n converges to zero. Between 1 and 3, x_n has one attracting point. After that, the points bifurcate over and over again until its behaviour becomes chaotic. Sometimes, the behaviour inside the chaotic region becomes simple for a period, until it resettles into chaos again. While the chaotic behaviour might look random because of its sensitivity to initial conditions, it is, in fact, deterministic.

1.6 6

1.6.1 (a)

As we saw in questions 3, 4 and 5, the behaviour of the logistic map changes dramatically as r increases.

1.6.2 (b)

Comparing the results from questions 1, 2 and 3, we can observe the emergence of surprising behaviour from simple rules. This is particularly the case with the oscillation between attraction points and the decline into chaos - nothing about the simple equations immediately suggests this complex behaviour.

1.6.3 (c)

While seemingly unpredictable, we can reproduce exactly the previous findings, because even though the behaviour is surprising, it is deterministic. This displays the duality between order and chaos present in the logistic map. This is also suggested by the streaks in the bifurcation diagram - while it looks random, in fact it follows a movement that precisely determined by the equation.

1.6.4 (d)

Given that the logistic map is a system with simple rules that displays unexpected behaviour that changes over time, we conclude that it is a complex system.

[ ]:
```
!jupyter nbconvert M7001_quiz4_24000114067.ipynb --to pdf --LatexPreprocessor.
 ↪date ""
```