

# 知能情報実験II(第13回):ロジスティックモデル

175751C 宮城孝明

平成31年2月1日

## 目 次

<b>1</b>	<b>リターンマップの <math>r</math> 依存性</b>	<b>2</b>
1.1	プログラムのソースコード (python) . . . . .	2
1.2	プログラムのソースコード (C) . . . . .	2
1.3	結果と考察 . . . . .	3
<b>2</b>	<b>初期値鋭敏性</b>	<b>4</b>
2.1	プログラムのソースコード (python) . . . . .	4
2.2	プログラムのソースコード (C) . . . . .	4
2.3	結果と考察 . . . . .	5

## 1 リターンマップのr依存性

### 1.1 プログラムのソースコード (python)

Listing 1: python\_code

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3 x=[]
4 y=[]
5 init=0.7
6 r=3.80
7 for i in range(0,101):
8     if i%2==0:
9         x.append(init)
10        init=r*(1-init)*init
11        y.append(init)
12    else:
13        x.append(init)
14        y.append(init)
```

### 1.2 プログラムのソースコード (C)

Listing 2: C 言語\_code

```
1 #include <stdio.h>
2 #define MAX 100
3 float r = 3.80;
4 float x_list[MAX];
5 float y_list[MAX];
6
7 int main(void){
8     FILE *file;
9     file = fopen("c_returnmap.txt","w");
10    float init = 0.7;
11    for(int i=0; i<MAX; i++){
12        if (i%2==0){
13            x_list[i]=init;
14            init = r * (1 - init) * init;
15            y_list[i]=init;
16        }else{
17            x_list[i]=y_list[i-1];
18            y_list[i]=y_list[i-1];
19        }
20    }
21    for(int j=0; j<MAX; j++){
22        fprintf(file,"%0.3f",x_list[j]);
23        fprintf(file,",");
24        fprintf(file,"%0.3f",y_list[j]);
25        fprintf(file,"\n");
26    }
```

```

27 | fclose(file);
28 | printf("終了しました");
29 |     return 0;
30 | }

```

### 1.3 結果と考察

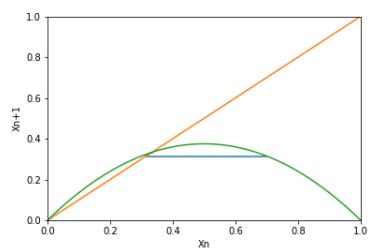


図 1:  $r = 1.50$  のグラフ.

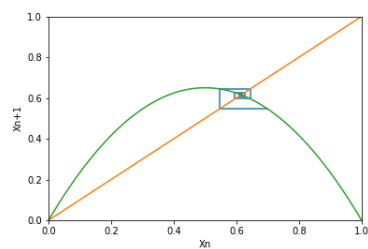


図 2:  $r = 2.60$  のグラフ.

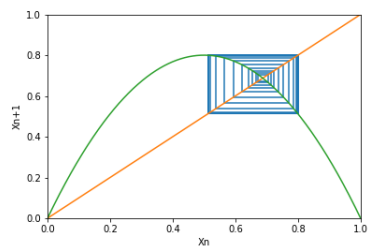


図 3:  $r = 3.20$  のグラフ.

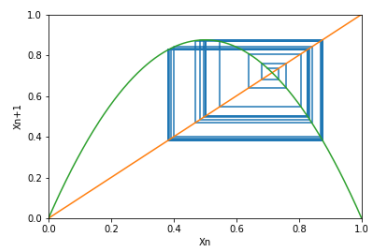


図 4:  $r = 3.50$  のグラフ.

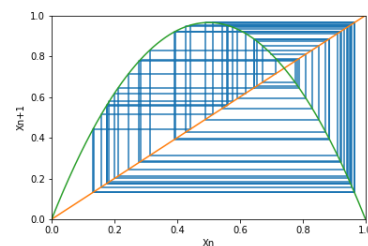


図 5:  $r = 3.86$  のグラフ.

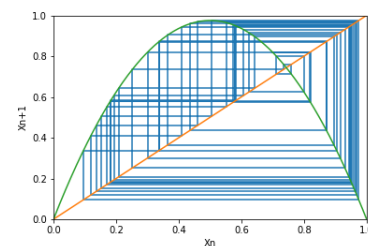


図 6:  $r = 3.90$  のグラフ.

今回求めたリターンマップモデル図は、ある空間内に存在する生物の個体数変動の特徴を現している。

図 1 では、青の直線が個体数の変動を示している。それからわかる事としては、子世代の個体数には変動が見られず、親世代は約 0.3 から約 0.7 の間

に動いている。つまり、最大増加率が1.5以下であれば大きな変動が見られないと思われる。

次に、図5と6を見てみる。図4の値と比べて、最大増加率が3.8を超えていると変動する幅も大きいとわかる。しかし、青線は決まった空間内を越えることがないため、最大増加率が4以上なっても変わらないことが考えられる。

## 2 初期値鋭敏性

### 2.1 プログラムのソースコード (python)

Listing 3: python.code

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3 x=[]
4 y=[]
5 r=1.50
6 for init in np.arange(0,1,0.001):
7     init_n=init
8     for i in range(0,201):
9         if 150<=i:
10             x.append(init)
11             y.append(init_n)
12             init_n = r*(1-init_n)*init_n
```

### 2.2 プログラムのソースコード (C)

Listing 4: C 言語.code

```
1 #include <stdio.h>
2 #define MAX 200
3 float r = 1.50;
4 float x_list[MAX];
5 float y_list[MAX];
6
7 int main(void){
8     FILE *file;
9     file = fopen("c_sensitivity.txt","w");
10    double init = 0;
11    for(int i=0; i<1000; i++){
12        double init_n = init;
13        for(int j=0; j<MAX; j++){
14            if (150<=j){
15                fprintf(file,"%0.3f",init);
16                fprintf(file,",");
17                fprintf(file,"%0.3f",init_n);
```

```

18         fprintf(file, "\n");
19     }
20     init_n = r*(1-init_n)*init_n;
21 }
22     init+=0.001;
23 }
24 fclose(file);
25 printf("終了しました");
26     return 0;
27 }

```

## 2.3 結果と考察

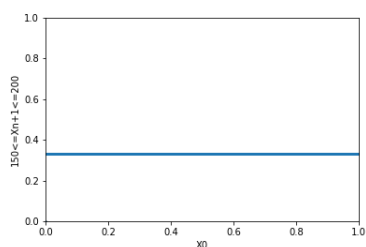


図 7:  $r = 1.50$  のグラフ.

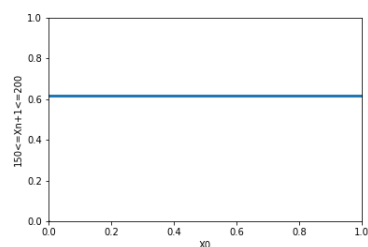


図 8:  $r = 2.60$  のグラフ.

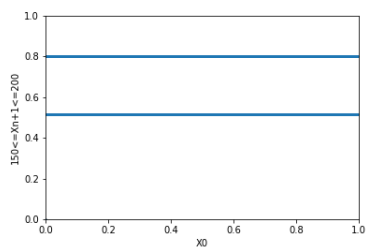


図 9:  $r = 3.20$  のグラフ.

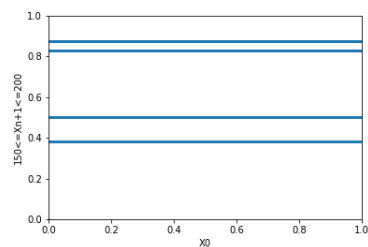


図 10:  $r = 3.50$  のグラフ.

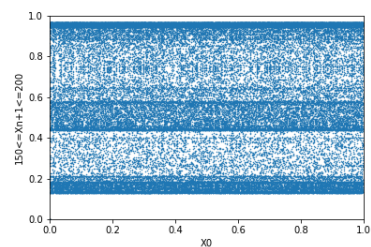


図 11:  $r = 3.86$  のグラフ.

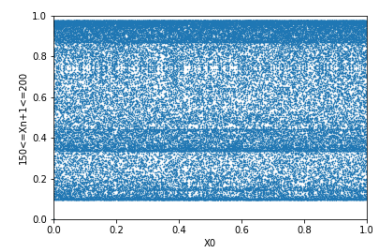


図 12:  $r = 3.90$  のグラフ.

初期値を 0.001 と細かくくぎりにしていき、初期値鋭敏性の性質を視覚的にわかりやすくするために、グラフを作成し求めた。そのグラフからある値に収束する図があれば、どの値にも収束せず色々な値に分散してしまう図がある。

図 1 から 4 までは、1 本か 2 本、4 本にしか値の変動が見られないが、最大増加率が 3.80 以上になると初期値の値によって、値が大きく変わっている。つまり、最大増加率が 3.8 以上になるとある空間内でも生物の増加や減少が激しくなると予測できる。