

密度効果が働く離散時間モデル

個体密度 N_t の時間変化: $N_{t+1} = rN_t$

r が定数の場合、個体密度は指数的に変化する

$r > 1$ の時、個体密度が発散してしまい現実的ではない

$$\lim_{t \rightarrow \infty} N_t = \infty$$

現実の生物集団では、 r は一定で有り続けることはなく、餌の不足や環境条件の悪化などにより、個体密度が高くなると r は低下すると思われる。

増殖率が個体密度に依存して変化する効果を**密度効果**という

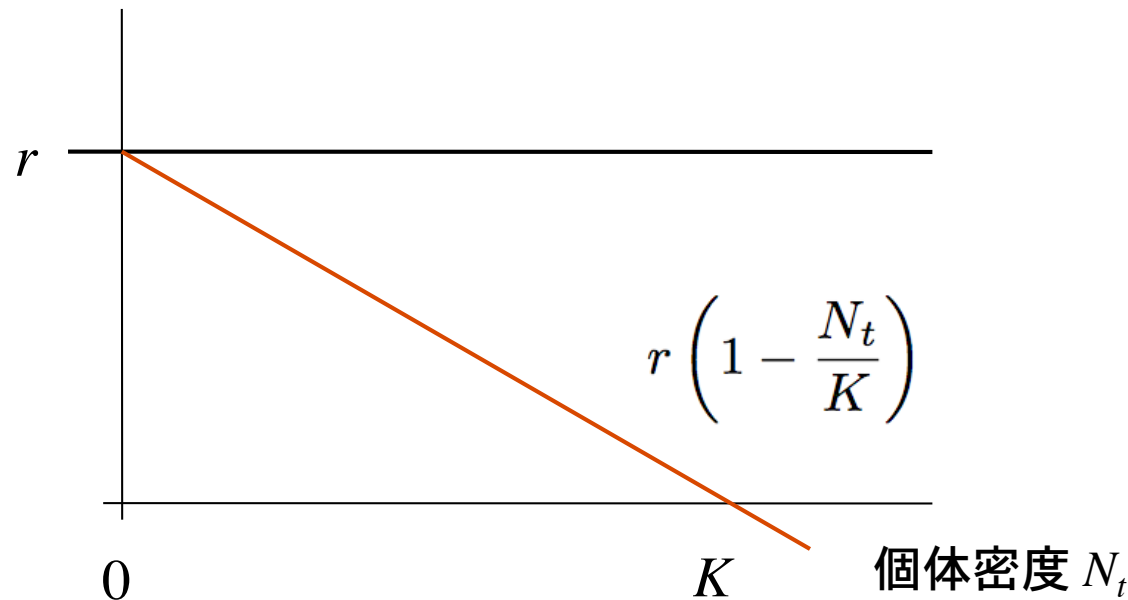
密度効果をモデルに組み込む

$$N_{t+1} = r N_t$$

↑

1 個体あたりの増殖率 r が定数ではなく、個体密度 N_t に比例して単調減少する場合を考える

増殖率



$$N_{t+1} = r \left(1 - \frac{N_t}{K} \right) N_t$$

離散時間の
ロジスティックモデル

個体密度が極めて小さければ ($N_t \ll 1$) 増殖率はほぼ r だが、密度の増加につれて増殖率が減少し、 K に達したときゼロとなるモデル。こういう生物が実際に存在するかは別問題。あくまでモデルである。

ロジスティックモデルの解析

$$N_{t+1} = r \left(1 - \frac{N_t}{K} \right) N_t \quad \begin{array}{l} \text{右辺が非線形であり解けない。} \\ \text{2つのパラメータ } r, K \end{array}$$

両辺を K で割って、 N_t / K を新たに x_t と書くと、

$$\frac{N_{t+1}}{K} = r \left(1 - \frac{N_t}{K} \right) \frac{N_t}{K} \quad \longrightarrow \quad x_{t+1} = r(1 - x_t)x_t$$

x_t の振る舞い(数列 $\{x_0, x_1, x_2, x_3, \dots\}$) が分かれば、これを K 倍したものが N_t の振る舞いとなる。今後は x_t に関する差分式に注目する。

パラメータ K の値は、 N_t の振る舞いに定性的な影響を与えない。
解析を簡易にするためにパラメータの数を減らすことを**無次元化**という。

Cobwebbing の方法

1 変数の差分方程式の振る舞いをグラフを用いて視覚的に解析する方法

Ricker diagram ともいう

一般に $x_{t+1} = f(x_t)$ の場合を考える。

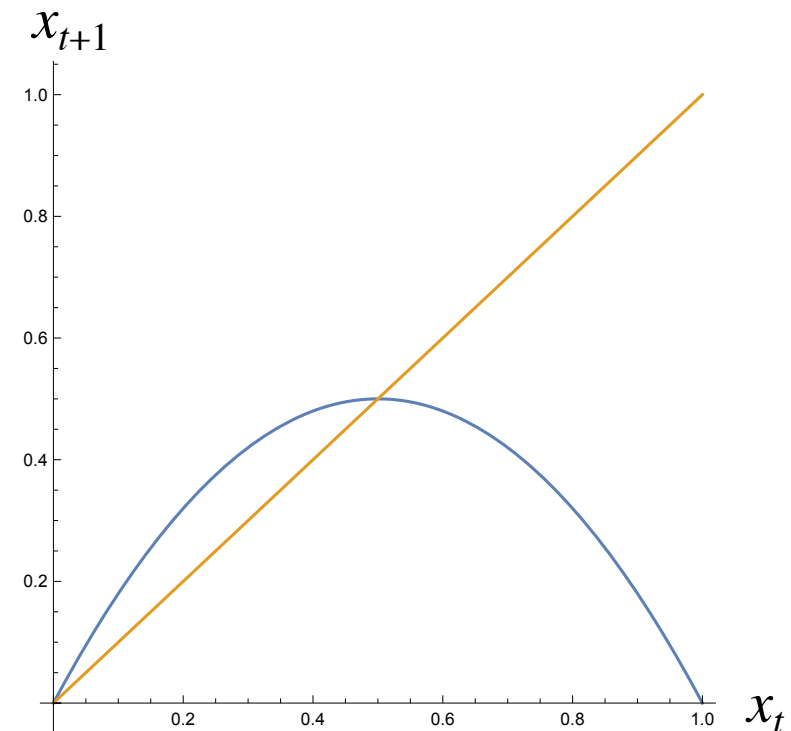
ロジスティックモデルの場合

$$f(x_t) = r(1 - x_t)x_t$$

Co-webbing の下準備

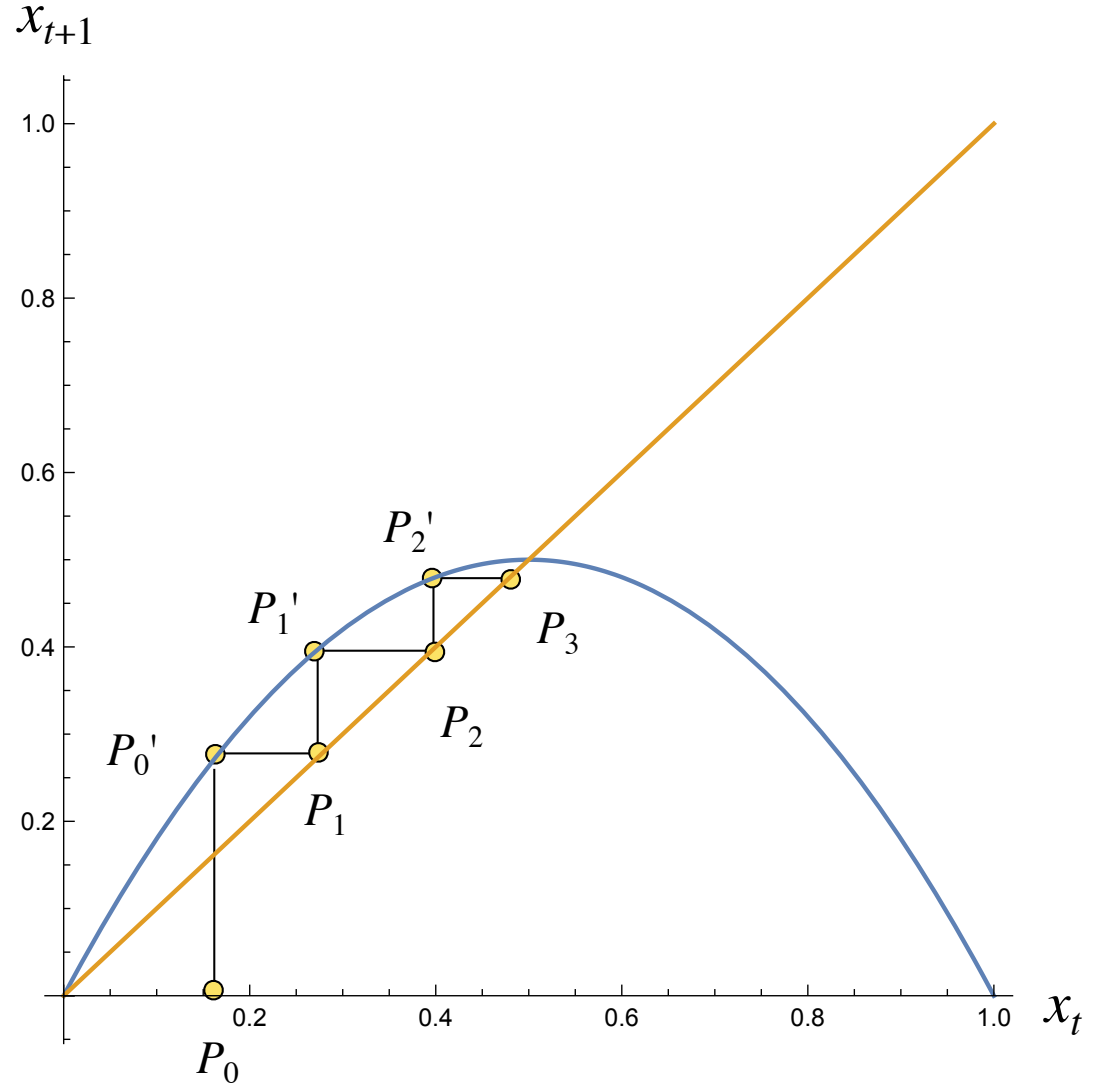
1) 横軸に x_t 、縦軸に $x_{t+1} = f(x_t)$ の グラフを描く

2) 原点を通り傾き 1 の直線 $x_{t+1} = x_t$ を描く



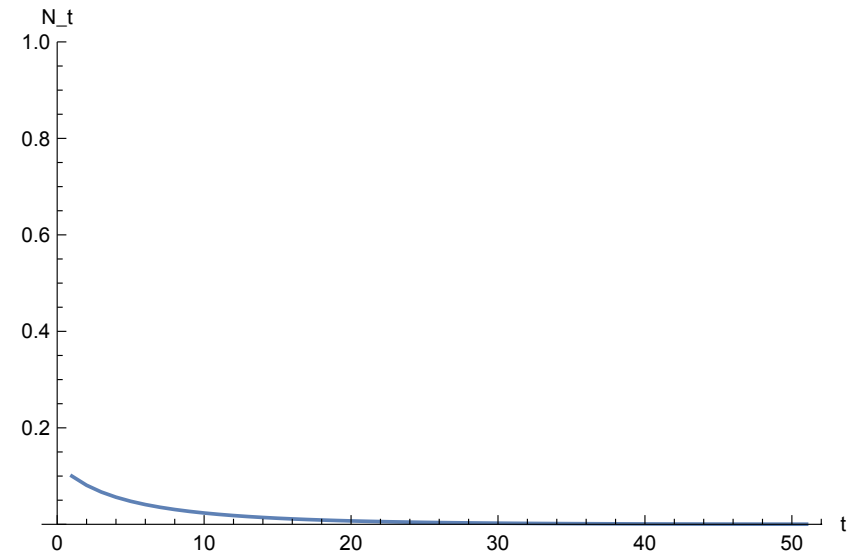
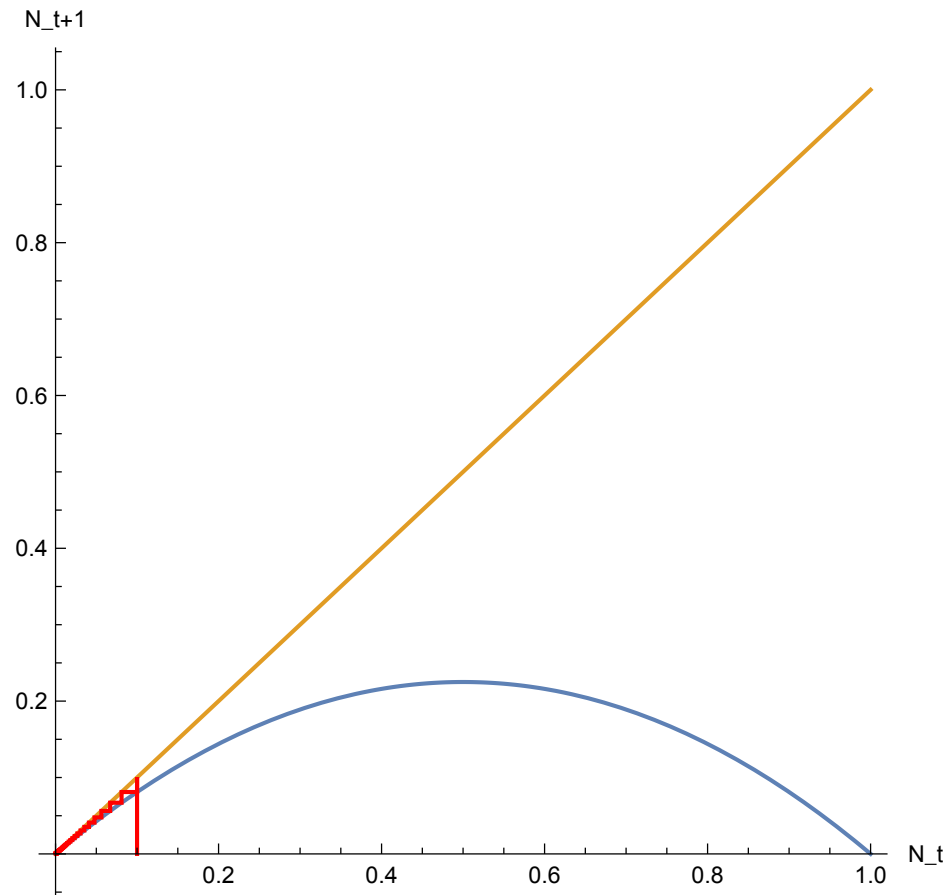
Cobwebbing 手順

- 1) 横軸上に初期値 x_0 を取り、点 P_0 とする
- 2) 点 P_0 から垂直線を延ばし、
関数 $f(x_t)$ との交点を点 P_0' とする
点 P_0' の y 座標は $x_1 = f(x_0)$ である
- 3) 点 P_0' から水平線を延ばし
傾き1の直線との交点を P_1 とする
点 P_1 の x 座標は $x_1 = f(x_0)$ である
- 4) P_1 を新たな始点として上記の
手順1)～3)を繰り返す



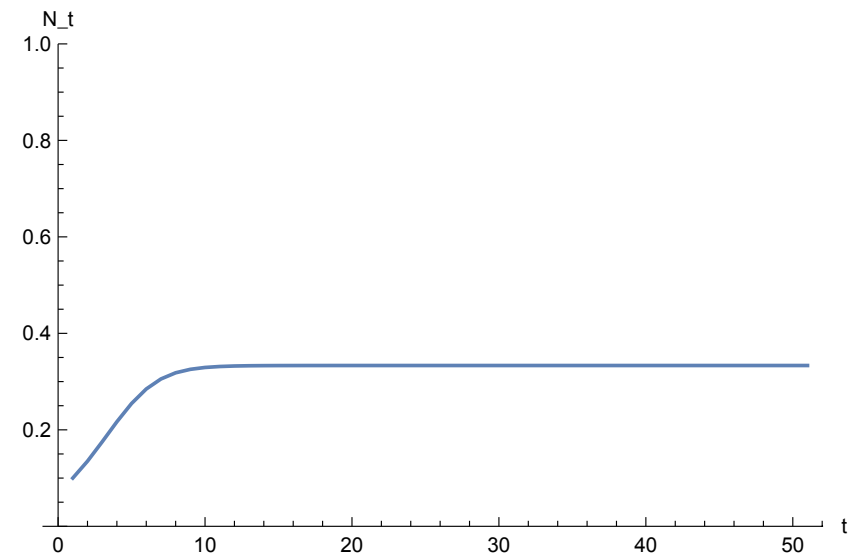
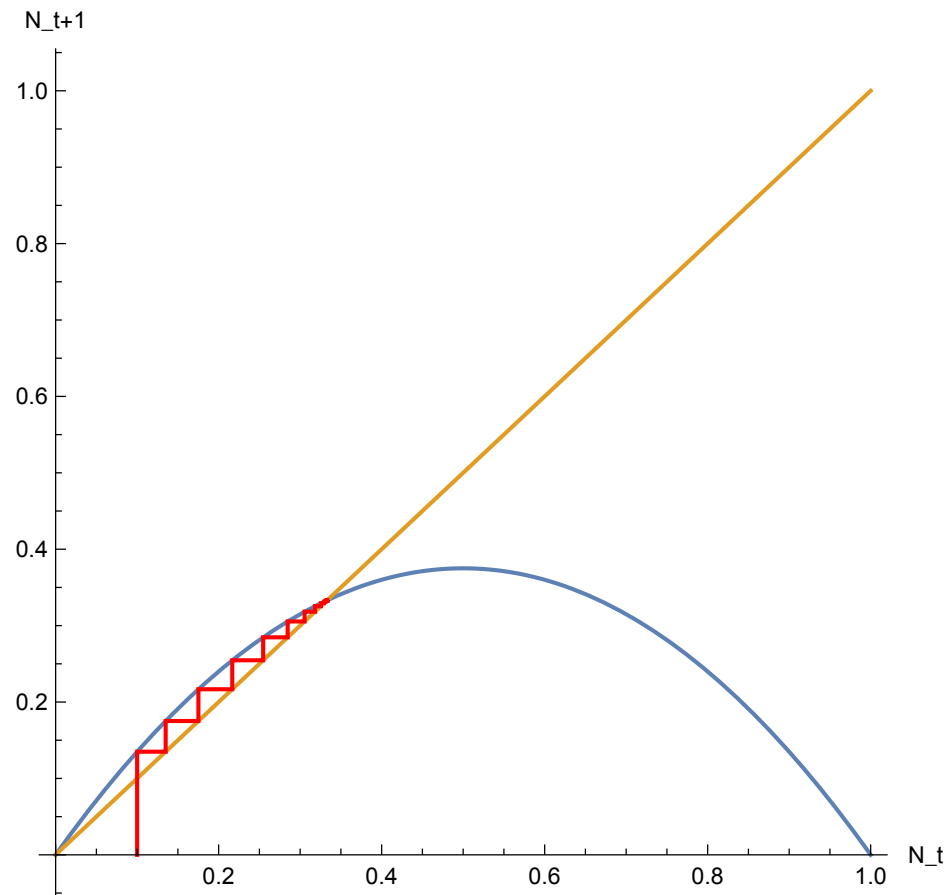
数値計算例 0

$$x_{t+1} = r(1 - x_t)x_t \quad r = 0.9$$



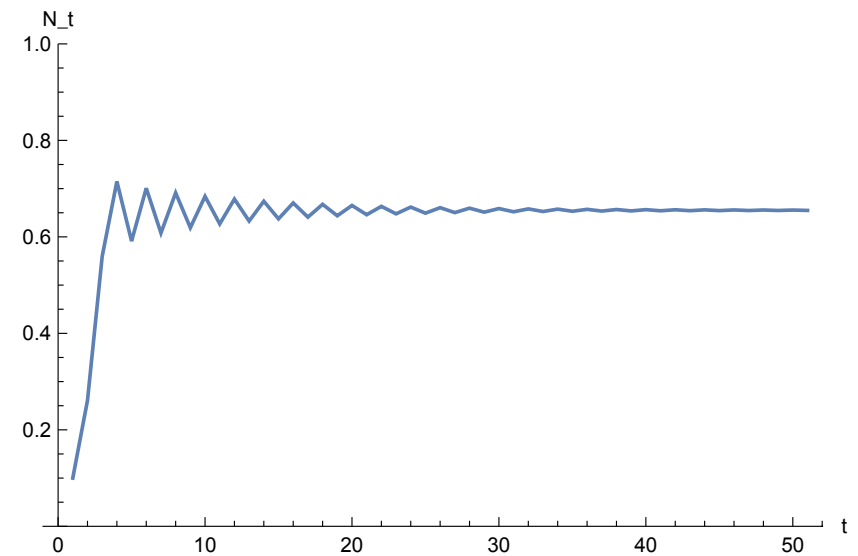
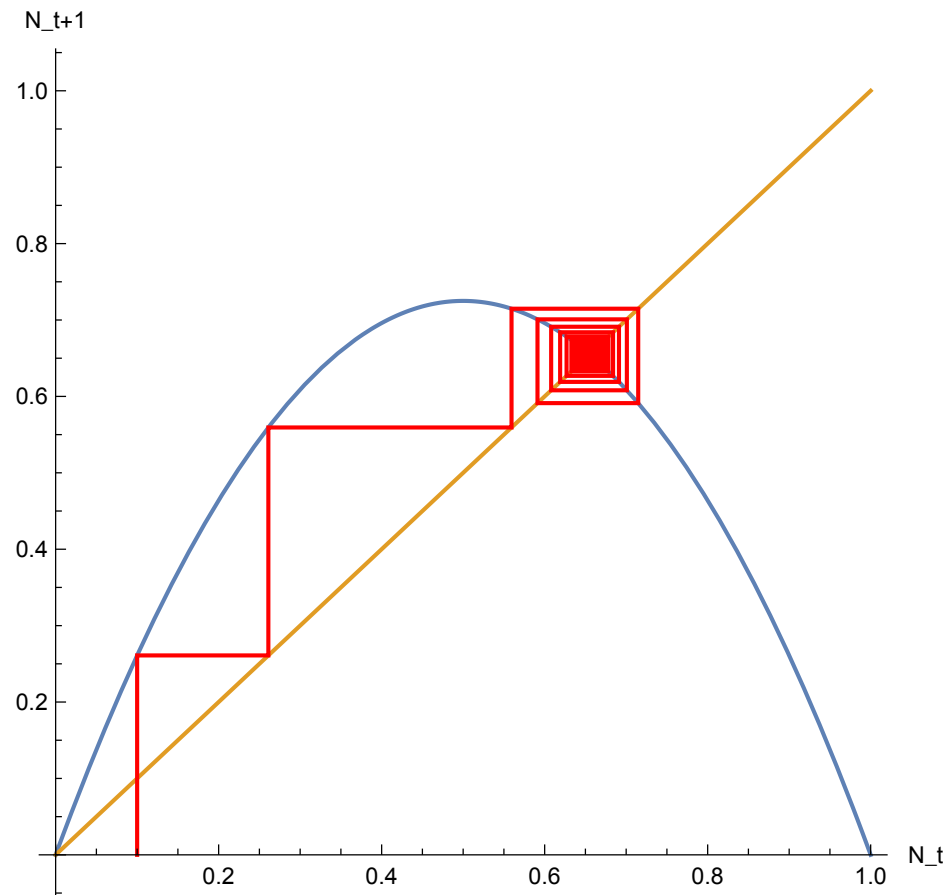
数値計算例 1

$$x_{t+1} = r(1 - x_t)x_t \quad r = 1.5$$



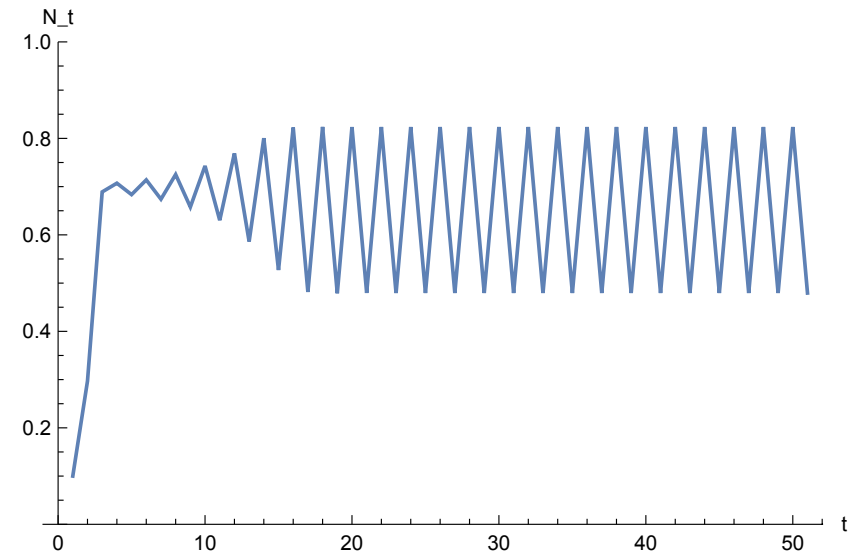
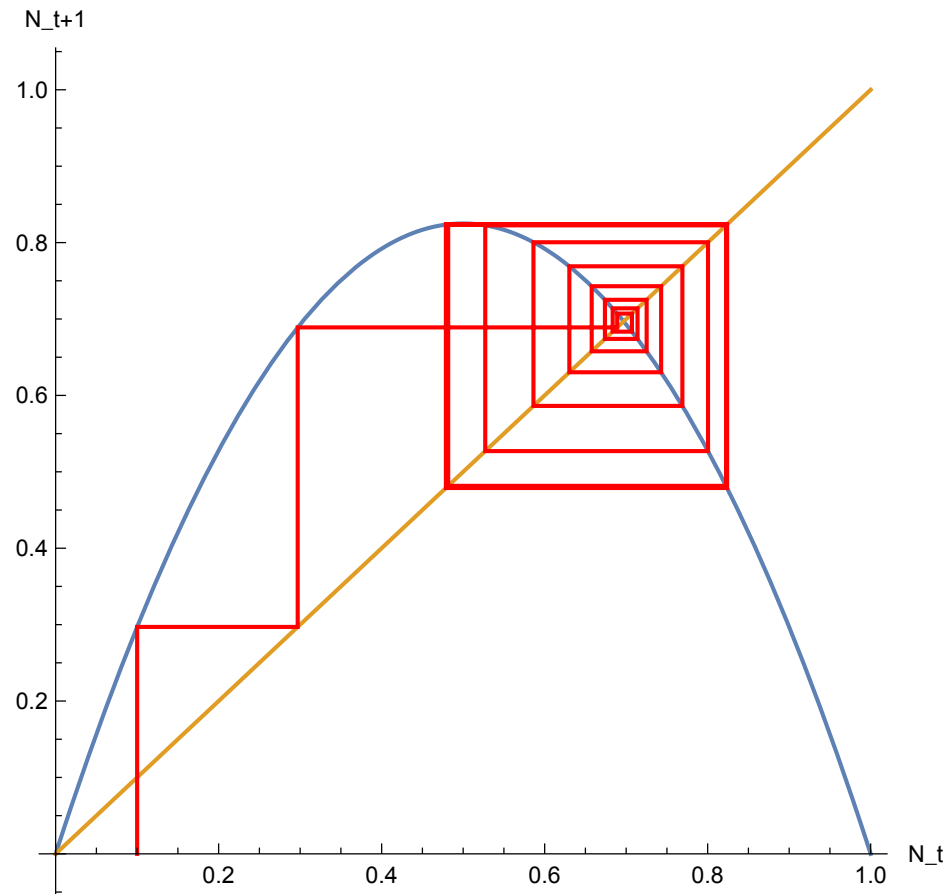
数値計算例 2

$$x_{t+1} = r(1 - x_t)x_t \quad r = 2.9$$



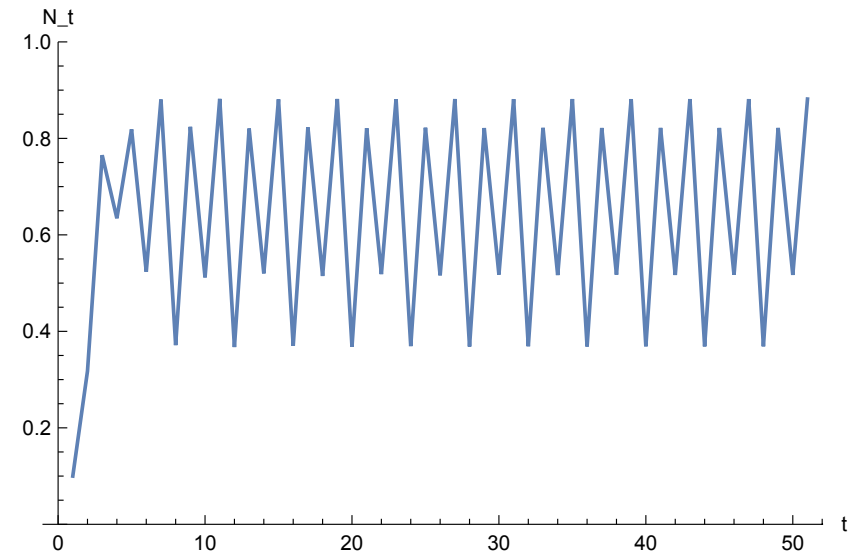
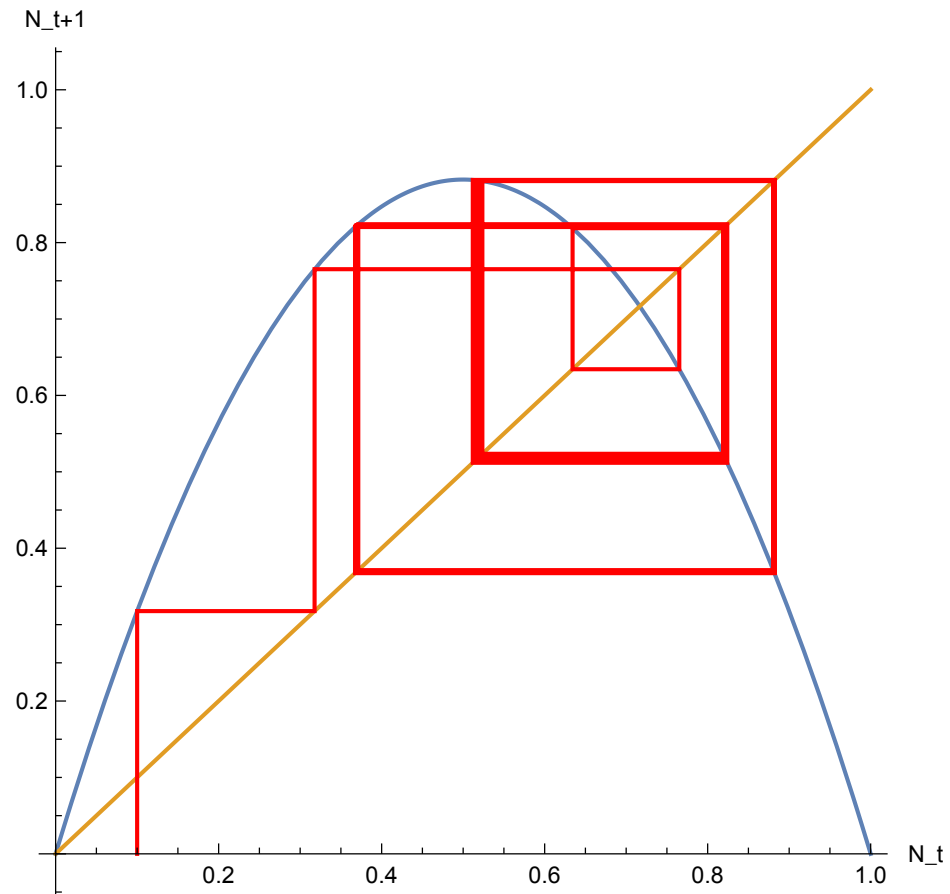
数値計算例 3

$$x_{t+1} = r(1 - x_t)x_t \quad r = 3.3$$



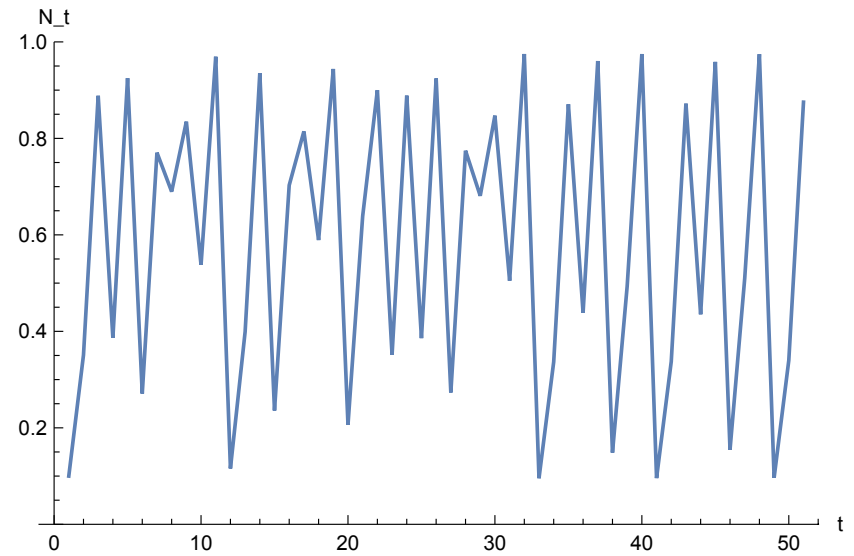
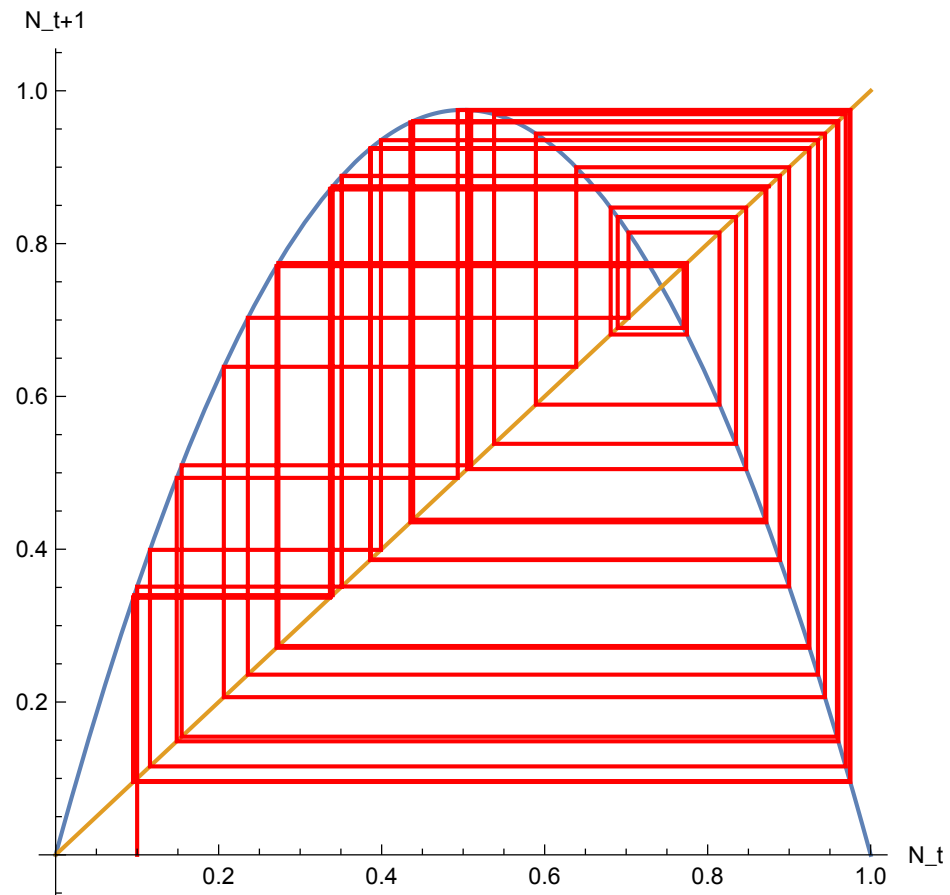
数値計算例 4

$$x_{t+1} = r(1 - x_t)x_t \quad r = 3.53$$



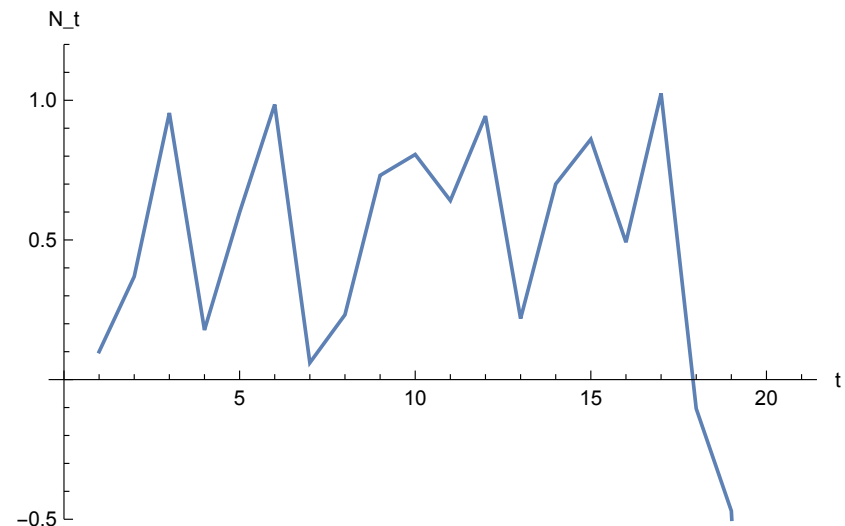
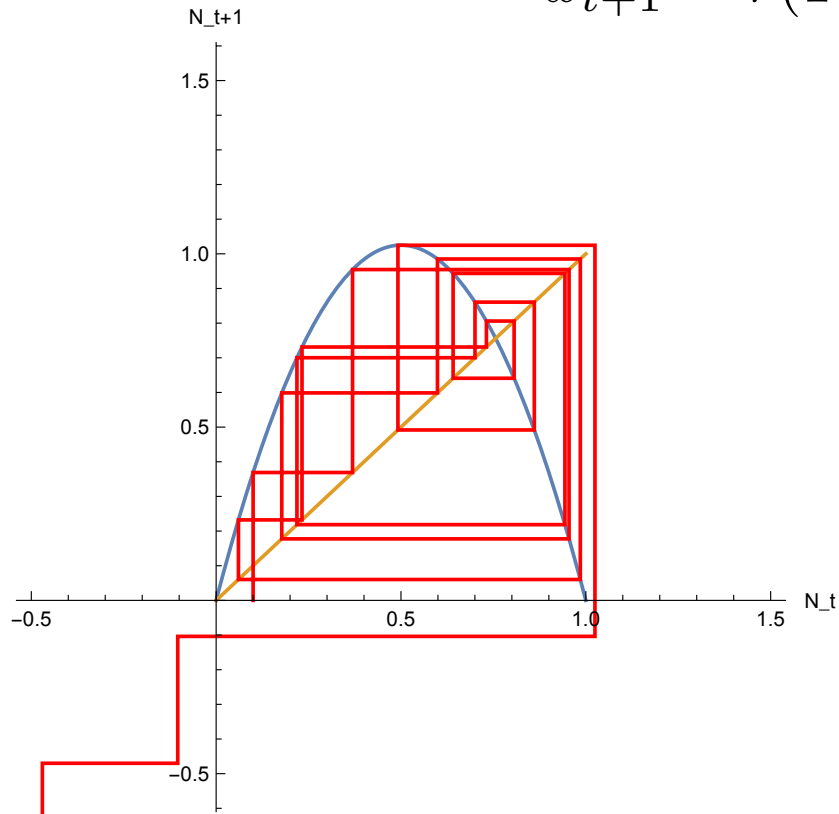
数値計算例 5

$$x_{t+1} = r(1 - x_t)x_t \quad r = 3.9$$



数値計算例 6

$$x_{t+1} = r(1 - x_t)x_t \quad r = 4.1$$



2 次関数 $r(1 - x_t)x_t$ の最大値は $r/4$ 。 $r > 4$ の時、最大値が 1 を越える。

次時刻密度 x_{t+1} が生物学的に意味を持つ(非負)為には

$0 \leq x_t \leq 1$ でなければならないが $r > 4$ の時、この条件は成立しない

平衡点

差分式 $x_{t+1} = f(x_t)$ において、
時間的に変化しない点のことを**平衡点**と呼ぶ。

平衡点 x^* は $x_{t+1} = x_t = x^*$ より、次の条件を満たす。

$$x^* = f(x^*)$$

離散ロジスティックモデルの場合、平衡点 x^* は次式の解で与えられる

$$x^* = f(x^*) = r(1 - x^*)x^*$$

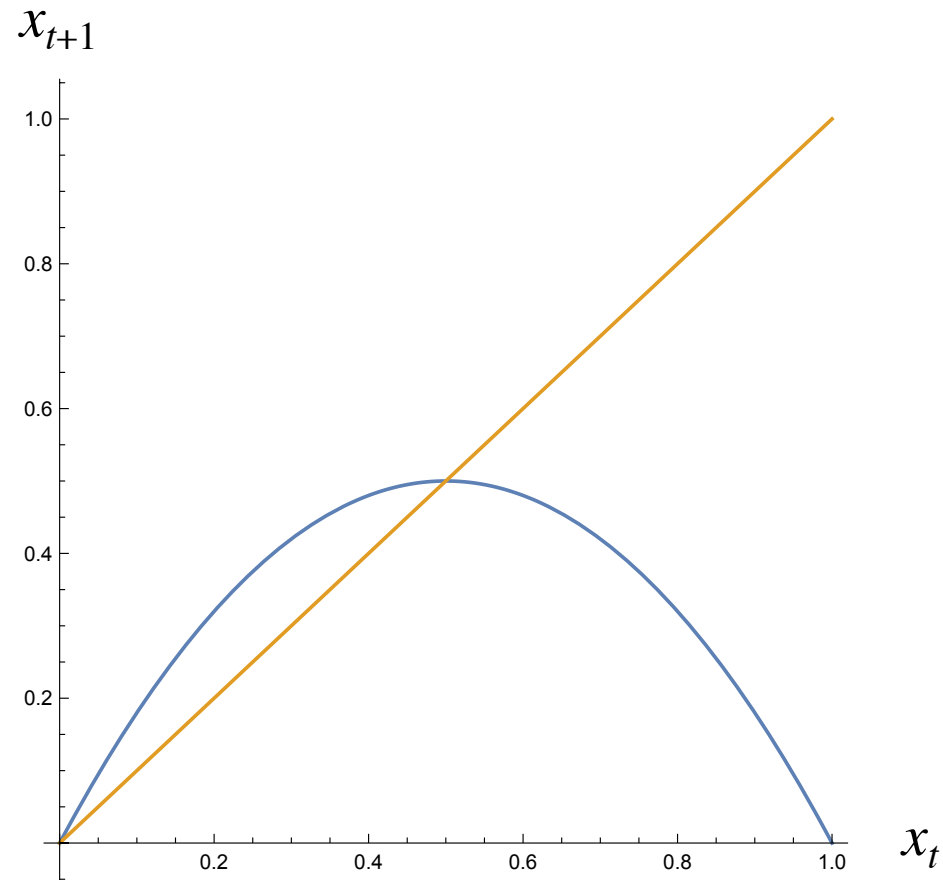
つまり、

$$x^* = 0, (r - 1)/r$$

平衡点と Cobwebbing

平衡点 x^* は Ricker diagram において関数 $x_{t+1} = f(x_t)$ と傾き 1 の直線 $x_{t+1} = x_t$ との交点に相当

離散ロジスティックモデルの平衡点 $x^* = 0, \frac{r-1}{r}$



平衡点の安定性

平衡点とは時間的に変化しない状態。もし初期状態が平衡点 x^* であれば ($x_0 = x^*$)、未来永劫その状態に留まり続ける ($x_t = x^*$ for $t \geq 1$)

$$x_1 = f(x_0) = f(x^*) = x^* \quad \longleftarrow \quad \text{平衡点の定義より } x^* = f(x^*) \text{ である}$$

$$x_2 = f(x_1) = f(x^*) = x^*$$

$$x_3 = f(x_2) = f(x^*) = x^*$$

⋮

平衡点の局所安定性:

平衡点から状態を微小量ずらしたときに、ずれが時間とともにゼロに収束する場合、この平衡点は**局所的に安定**であるという。ずれが時間とともに拡大する場合、**不安定**であるという。

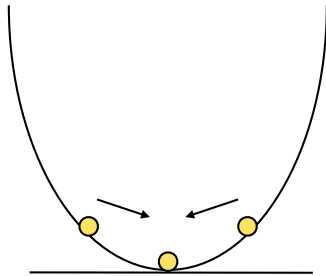
局所安定性

平衡点のイメージ

平らな平面におかれたボールは動かない

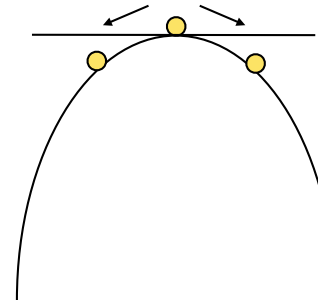


安定な平衡点のイメージ



谷間にあるボールは、位置を少しずらしても元に戻る(平衡点からのずれがゼロに収束)

不安定な平衡点のイメージ



山の頂にあるボールは、位置を少しずらしたただけで元に戻らない(平衡点からのずれが拡大する)

局所安定性解析

差分方程式 $x_{t+1} = f(x_t)$ の平衡点 x^* を考える

平衡点 x^* からの微小なずれを h_t と書くと

$$x_t = x^* + h_t \quad |h_t| \ll 1$$

元の式に代入して、ずれ h_t の式に書き直すと次式を得る

$$x^* + h_{t+1} = f(x^* + h_t)$$

ここで右辺を平衡点の周りでテイラー展開すると

$$f(x^* + h_t) = f(x^*) + f'(x^*)h_t + \frac{1}{2}f''(x^*)h_t^2 + \dots$$

テイラー展開

$f(x)$ が a を含む区間で $n+1$ 回微分可能であれば、この区間内の任意の x に対して次式が成り立つ (a を中心としたテイラー展開)

$$f(x) = f(a) + f'(a)(x-a) + \frac{1}{2}f''(a)(x-a)^2 + \cdots + \frac{1}{n!}f^{(n)}(a)(x-a)^n + \cdots$$

平衡点からのずれ h_t が微小であれば、微少量 h_t の2次以上の項は無視できる

$$x^* + h_{t+1} = f(x^* + h_t) \approx f(x^*) + f'(x^*) h_t$$

x^* は平衡点であるので $x^* = f(x^*)$ を満たす。結局、ずれの時間変化に関する次式を得る

$$h_{t+1} \approx f'(x^*) h_t \quad \longrightarrow \quad h_t \propto (f'(x^*))^t$$

局所安定性

平衡点 x^* からの微小なずれは指数的に変化 $h_t \propto (f'(x^*))^t$

$0 < f'(x^*) < 1$ ずれは指数的に減少してゼロに収束

$-1 < f'(x^*) < 0$ ずれは符号を変えながら指数的に減少してゼロに収束



平衡点 x^* は局所的に安定 locally stable

$1 < f'(x^*)$ ずれは指数的に拡大

$f'(x^*) < -1$ ずれは符号を変えながら指数的に拡大



平衡点 x^* は不安定 unstable

まとめ

平衡点 x^* の局所安定性は、
 x^* におけるグラフ $x_{t+1} = f(x_t)$ の傾き $f'(x^*)$ で判定できる

離散ロジスティックの場合: $f(x_t) = r(1-x_t)x_t$ $f'(x_t) = r(1-2x_t)$

平衡点 $x^*=0$ については $f'(0) = r$

$0 < r < 1$ の時、局所的に安定

$1 < r$ の時、不安定

平衡点 $x^* = (r-1)/r$ については $f'((r-1)/r) = 2-r$

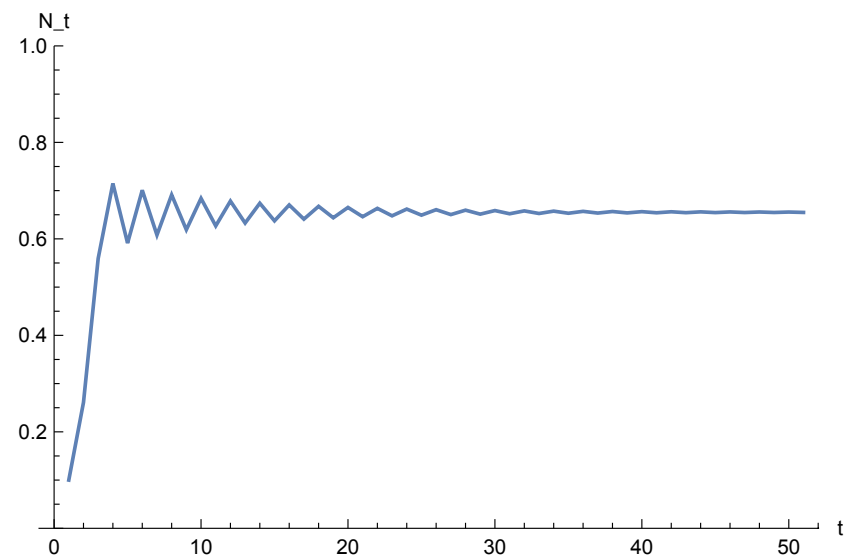
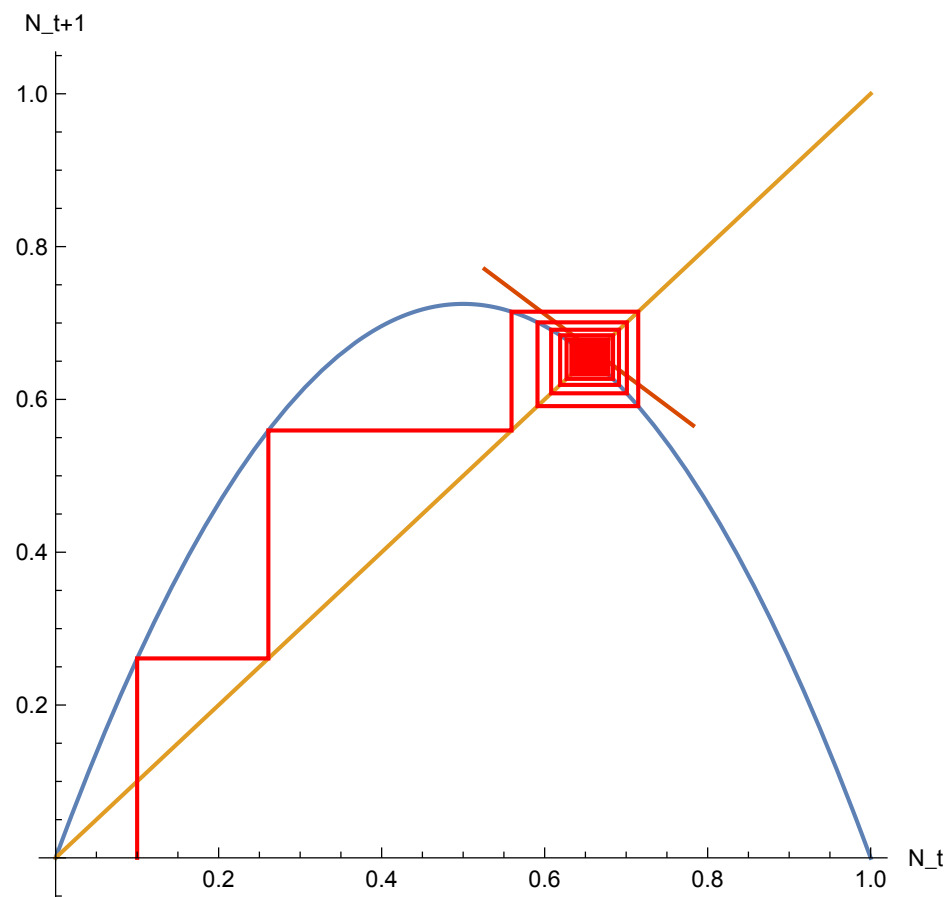
$0 < 2-r < 1$ ($1 < r < 2$) の時、局所的に安定(単調収束)

$-1 < 2-r < 0$ ($2 < r < 3$) の時、局所的に安定(減衰振動収束)

$2-r < -1$ ($3 < r < 4$) の時、不安定(振動して収束しない)

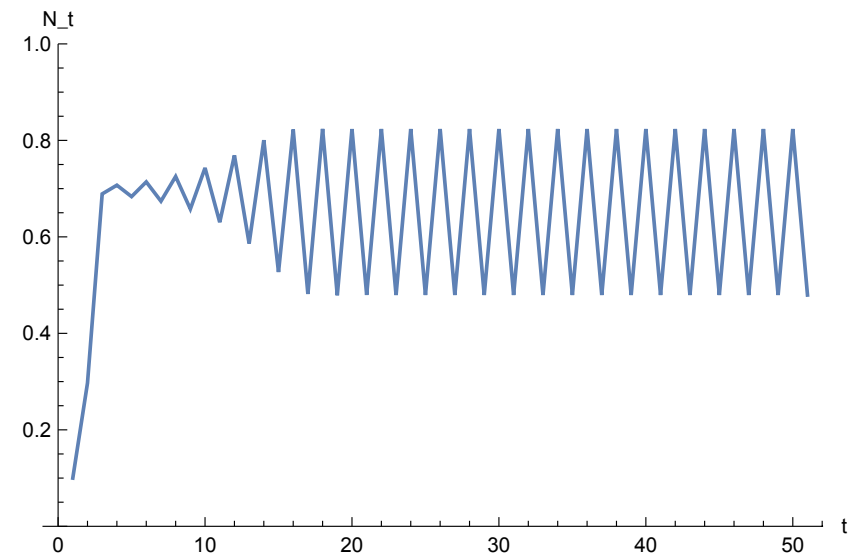
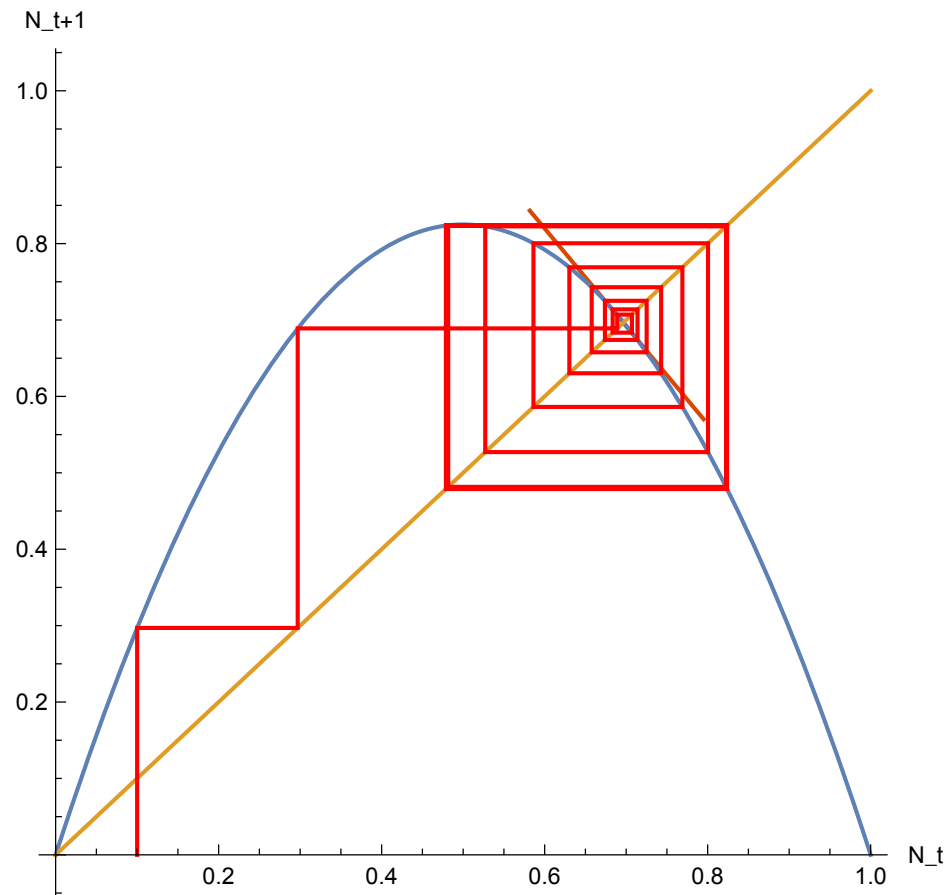
減衰振動して収束

$$x_{t+1} = r(1 - x_t)x_t \quad r = 2.9$$



周期 2 の振動

$$x_{t+1} = r(1 - x_t)x_t \quad r = 3.3$$

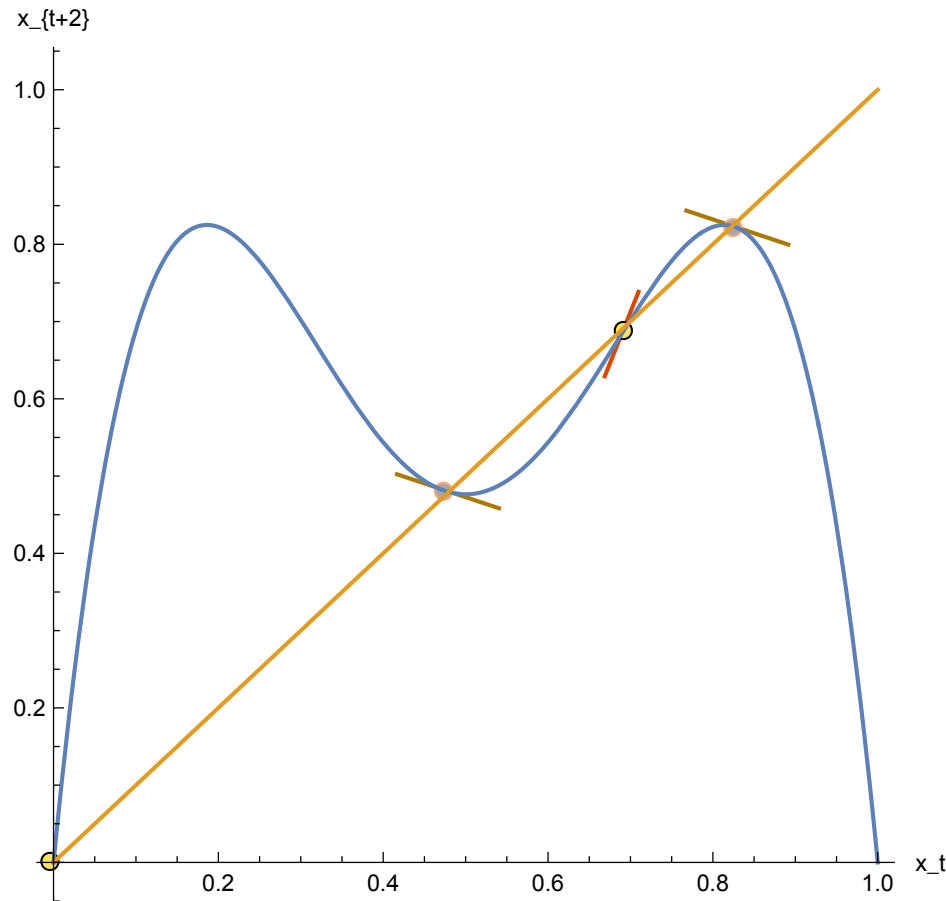


周期 2 の振動解

1 つ飛ばしのダイナミクスを考える

$$x_{t+2} = f(x_{t+1}) = f(f(x_t)) = F(x_t) \quad f(x) = r(1-x)x$$

離散ロジスティックモデルの場合、関数 F は 4 次式



- 周期 1 の平衡点(不安定)
周期 1 の平衡点は周期 2 の平衡点でもある

- 周期 2 の平衡点(安定)

$3 < r < 3.45$ の時、周期 2 の振動解が安定

r が 3.45 を越えると周期 4 の振動解が安定

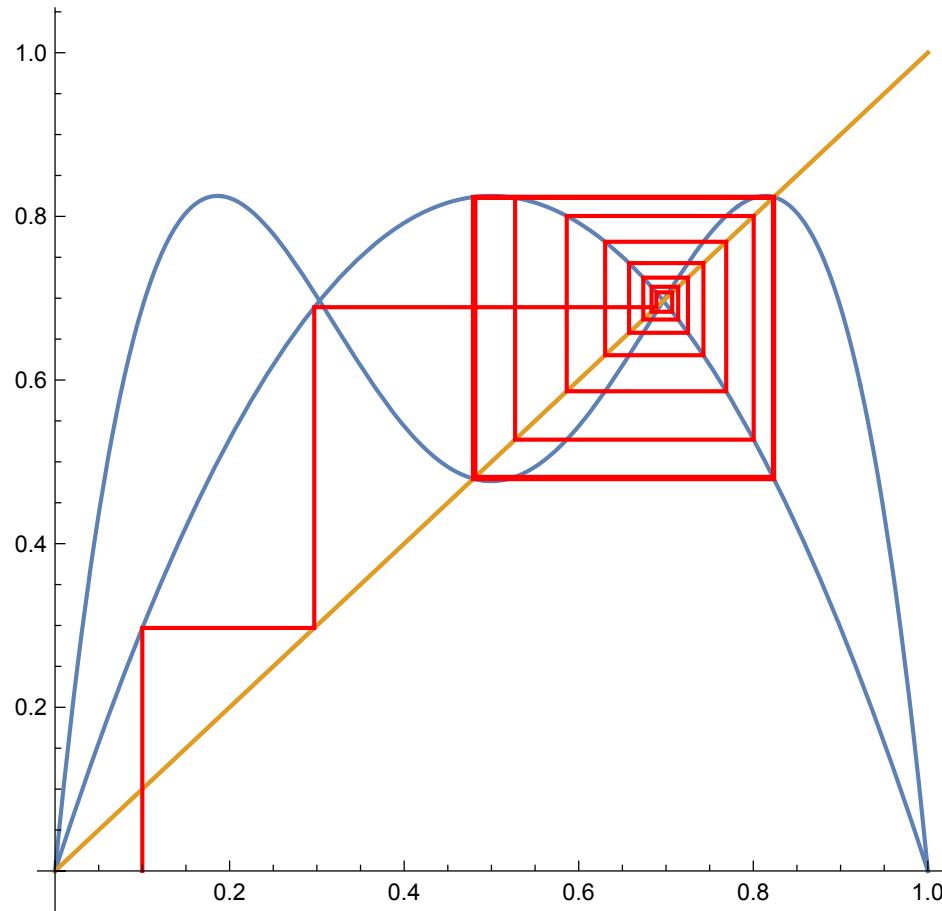
r がさらに大きくなると
周期 4 の振動解は不安定

周期 2 の振動解

1 つ飛ばしのダイナミクスを考える

$$x_{t+2} = f(x_{t+1}) = f(f(x_t)) = F(x_t) \qquad f(x) = r(1-x)x$$

離散ロジスティックモデルの場合、関数 F は 4 次式



- 周期 1 の平衡点(不安定)
周期 1 の平衡点は周期 2 の平衡点でもある
- 周期 2 の平衡点(安定)
 $3 < r < 3.45$ の時、周期 2 の振動解が安定
 r が 3.45 を越えると周期 4 の振動解が安定
 r がさらに大きくなると
周期 4 の振動解は不安定

ロジスティックモデルの r 依存性

$$0 < r < 1$$

x_t は単調に減少して 0 に収束

$$1 < r < 2$$

単調に非自明な平衡点に収束

$$2 < r < 3$$

減衰振動しながら非自明な平衡点に収束

$$3 < r < 3.45$$

周期 2 の振動解

$$3.45 < r < 3.57$$

r が大きくなるにつれて周期 4, 8, 16, 32, .. と周期が 2 倍になる振動解 (Period doubling)

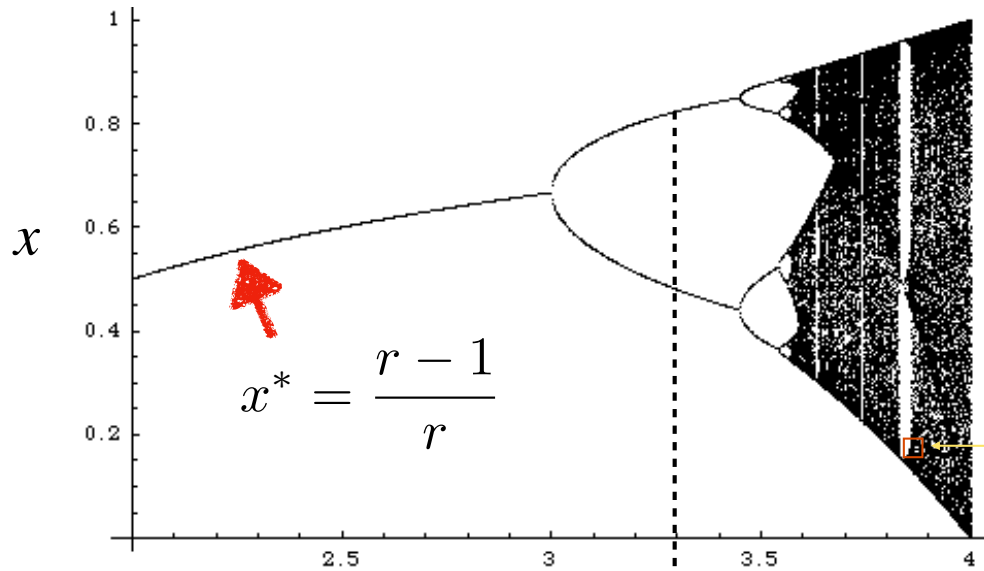
$$3.57 < r < 4$$

あらゆる周期を含む非周期的振る舞い: **カオス** Chaos
初期値のわずかな違いが予測不可能な結果をもたらす

分岐図

パラメータ r と十分時間が経過した後の x の関係。

自己相似形が現れる**フラクタル**

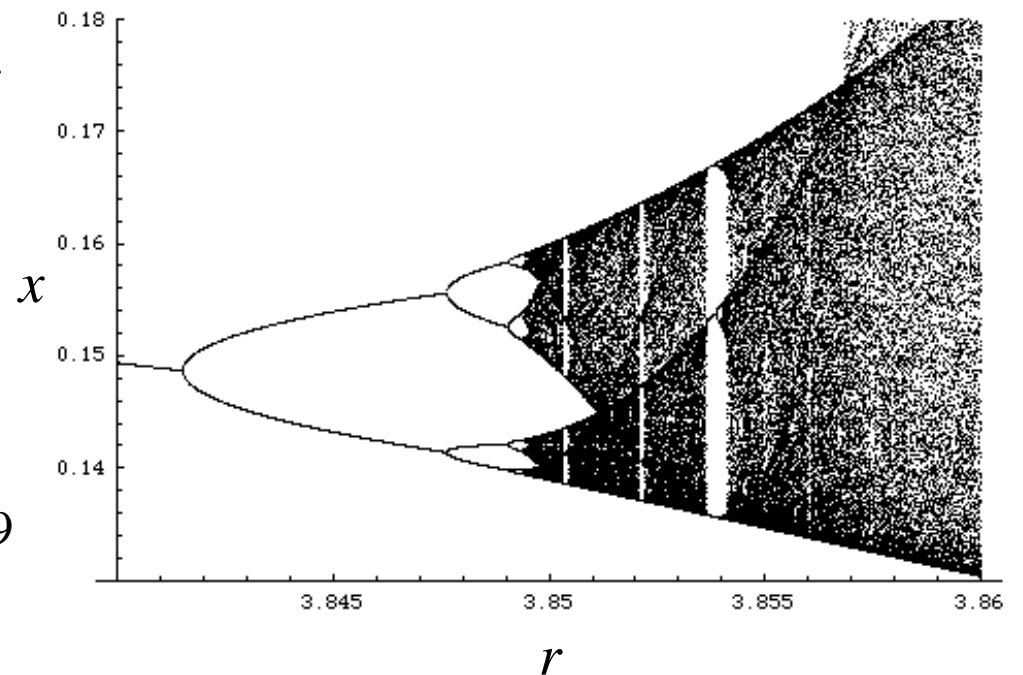


$$x_{t+1} = r(1 - x_t)x_t$$

$1 < r < 3$ の時、
安定平衡解 x^* (周期1)

$3 < r < 3.45$ で周期2解
 $r = 3.3$ の時、0.823 と 0.479

拡大図



カオスの初期値依存性

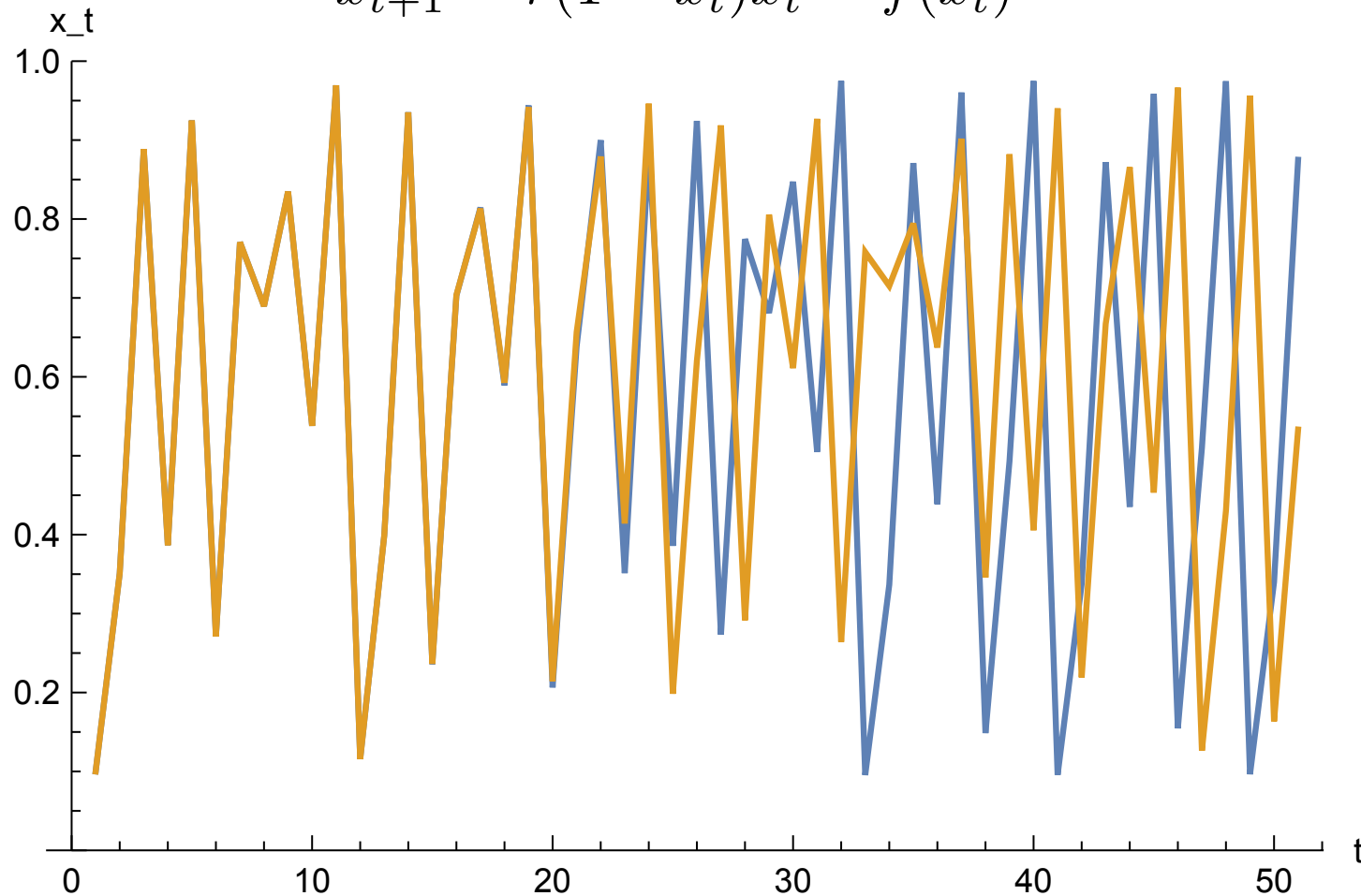
カオスが起こるパラメータ $r = 3.9$

初期値のわずかな違いが最終的に大きな変化をもたらす

$$x_0 = 0.1$$

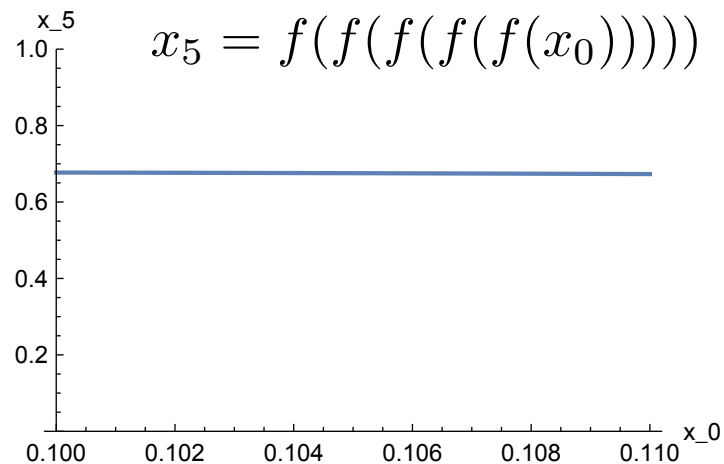
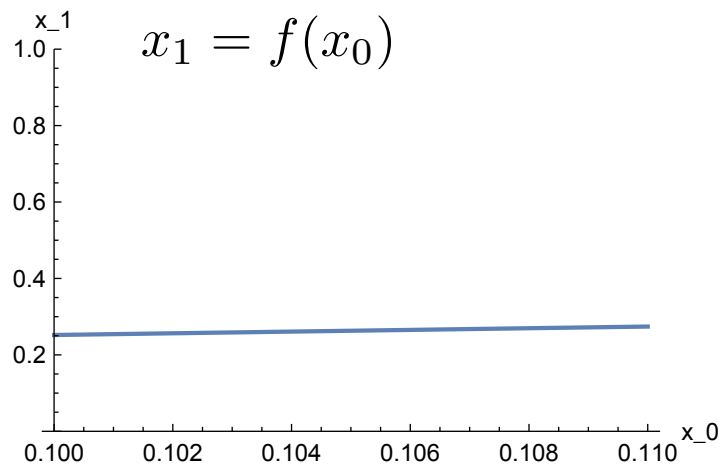
$$x_0 = 0.1000001$$

$$x_{t+1} = r(1 - x_t)x_t = f(x_t)$$

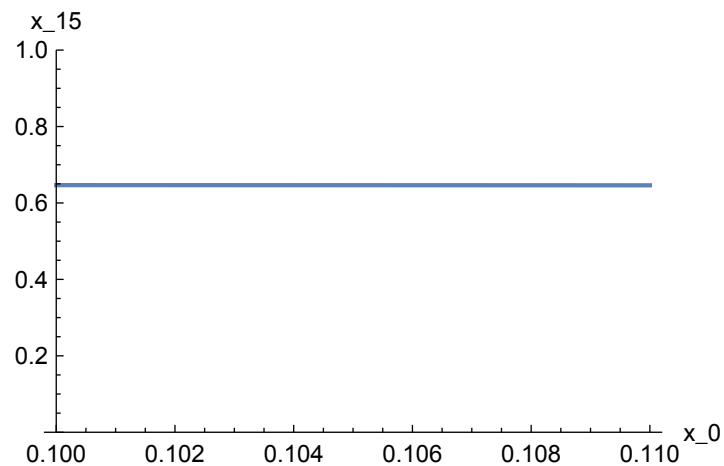
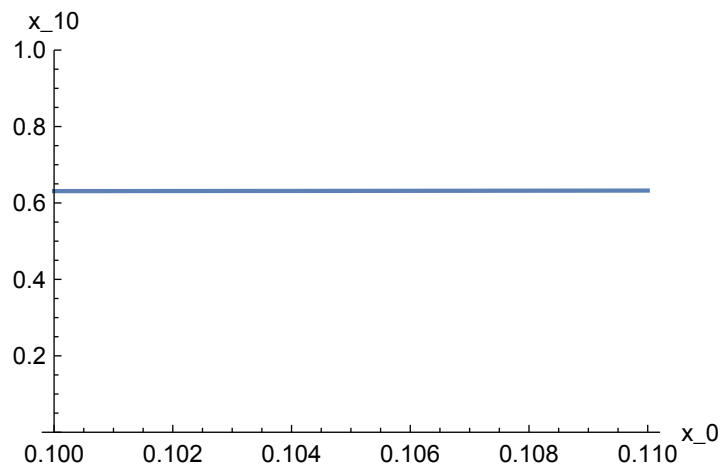


初期値依存性 1

減衰振動して平衡点に収束する場合: $r = 2.8$



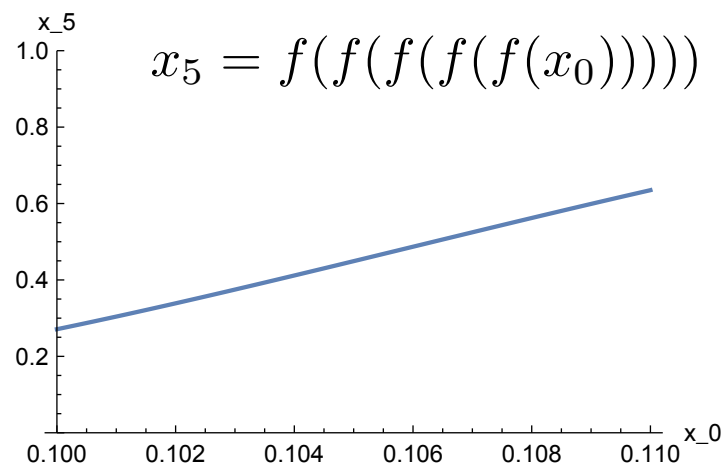
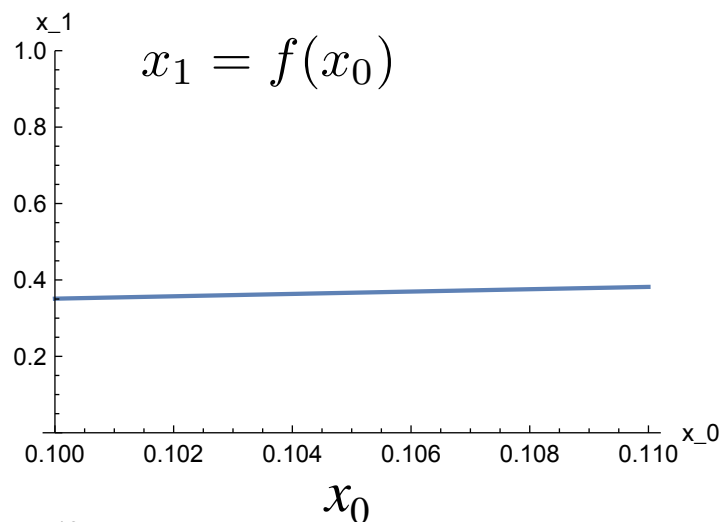
Out[*]=



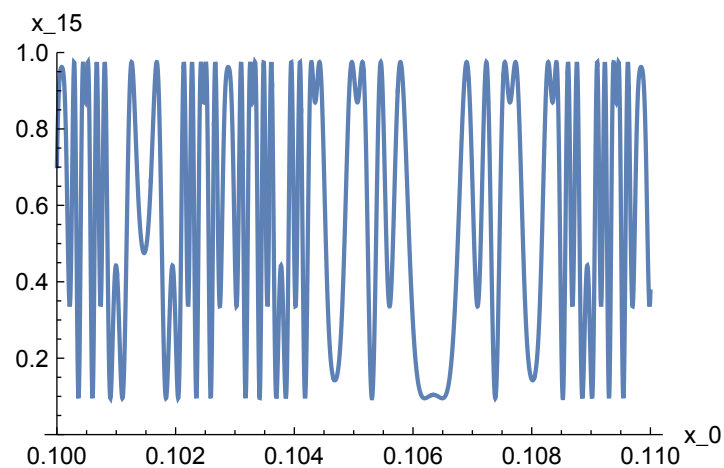
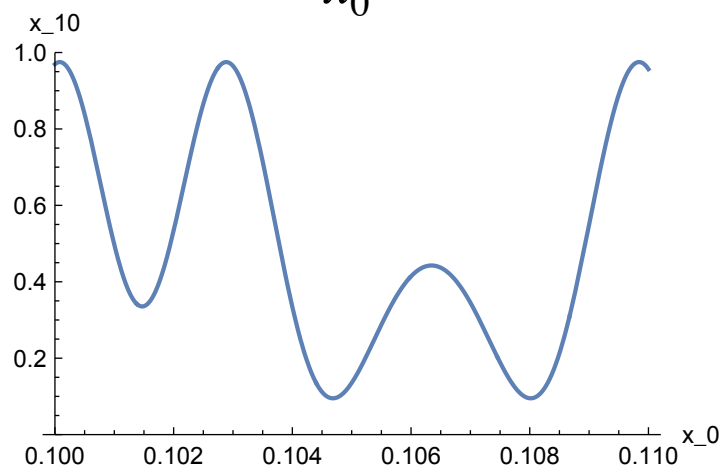
初期値 x_0 を変えても結果 x_1, x_5, x_{10}, x_{15} は大きく変わらない

初期値依存性 2

カオスが起こるパラメータ値: $r = 3.9$



Out[•]=



初期値 x_0 がわずかに変化しただけで時刻 15 後の値 x_{15} は全く変わってしまう!

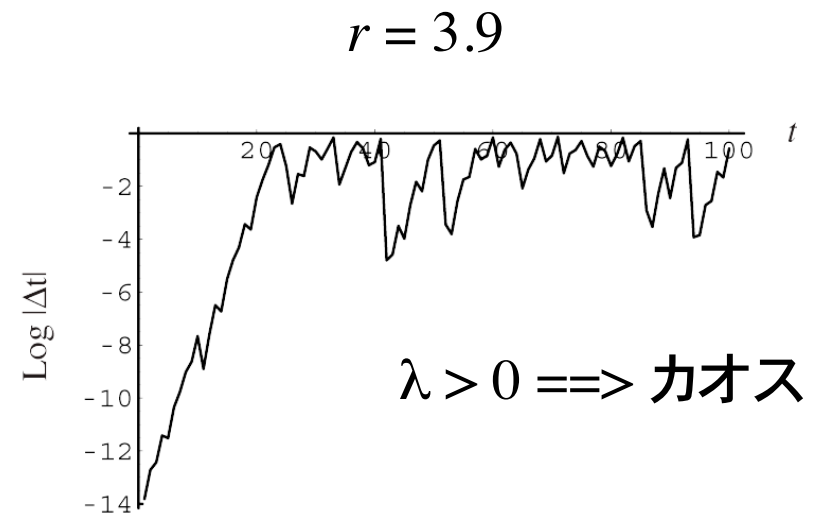
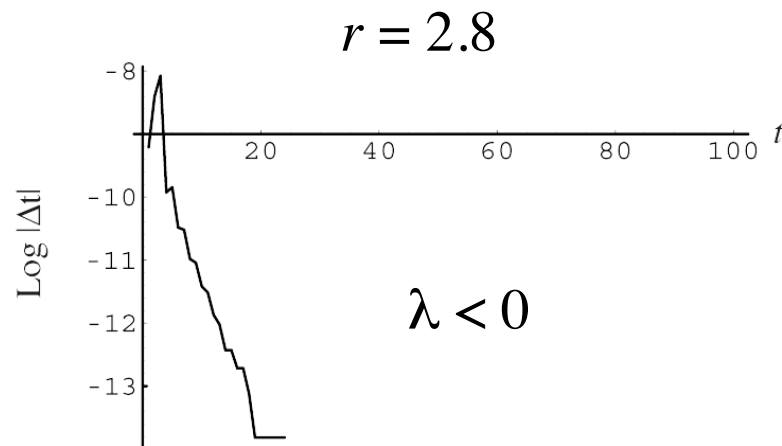
リヤプノフ指数

初期値の差異が時間とともにどのように縮小・拡大するのかを表す指標にリヤプノフ指数 λ がある。

$$|\Delta_t| = |\Delta_0| \exp[\lambda t]$$

$|\Delta_t|$: 時刻 t での時系列データの差異の絶対値

$|\Delta_0|$: 初期値の差異の絶対値



決定論的モデルとカオス

差分方程式 $x_{t+1} = f(x_t)$ は、初期状態 x_0 を決めれば
すべての x_t ($t = 1, 2, 3, \dots$) が一意的に定まる**決定論的モデル**

決定論的モデルに対し、変化が確率的に起こる事象を取り扱うモデルに**確率論的モデル**がある

決定論的モデルであるロジスティックモデル(単純な数式)でも、パラメータ値によってはカオスが起る。

観測精度を高めれば正確な予測が可能か？

長期天気予報は可能か？

バタフライ効果

自然界でカオスが起っているかどうかについては様々な意見がある

離散ロジスティックモデルの問題点

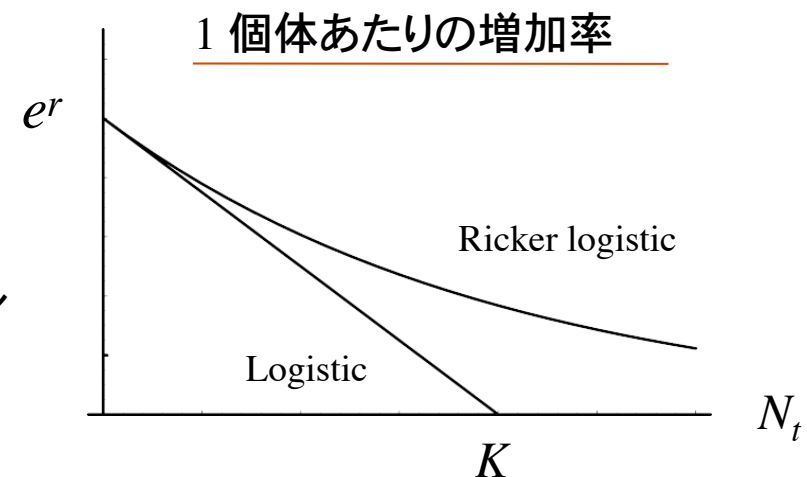
離散ロジスティックモデルは、個体密度 N_t がある時刻で K を越えると ($N_t > K$) 次時刻の個体密度は負となり ($N_{t+1} < 0$)、生物学的に意味をなさない。

$$N_{t+1} = r \left(1 - \frac{N_t}{K} \right) N_t$$

この点を改良したモデルとして、個体密度 N_t が高くなっても増殖率が負にならない次のモデルがある。これを Ricker ロジスティックモデルという。

$$N_{t+1} = \exp \left[r \left(1 - \frac{N_t}{K} \right) \right] N_t$$

ロジスティックモデルの r は Ricker ロジスティックモデルの er に相当する。

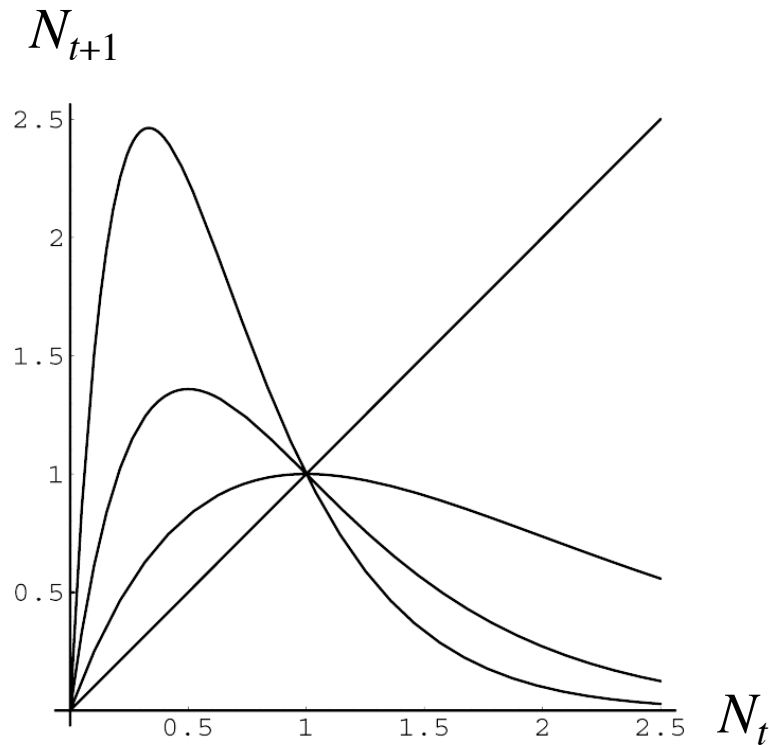


Ricker ロジスティックモデル

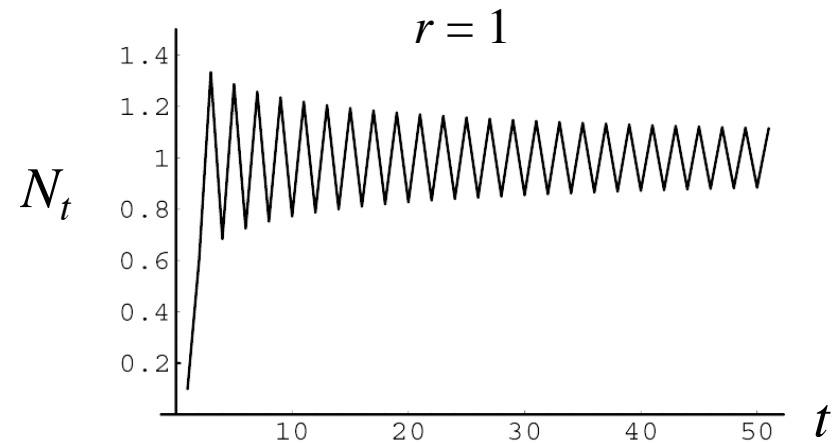
$$N_{t+1} = \exp \left[r \left(1 - \frac{N_t}{K} \right) \right] N_t$$

r の値が大きいと、より鋭いピークを持つ

$N_0 > 0$ なら $N_t > 0$ が保証される



$K = 1, r = 1, 2, 3$



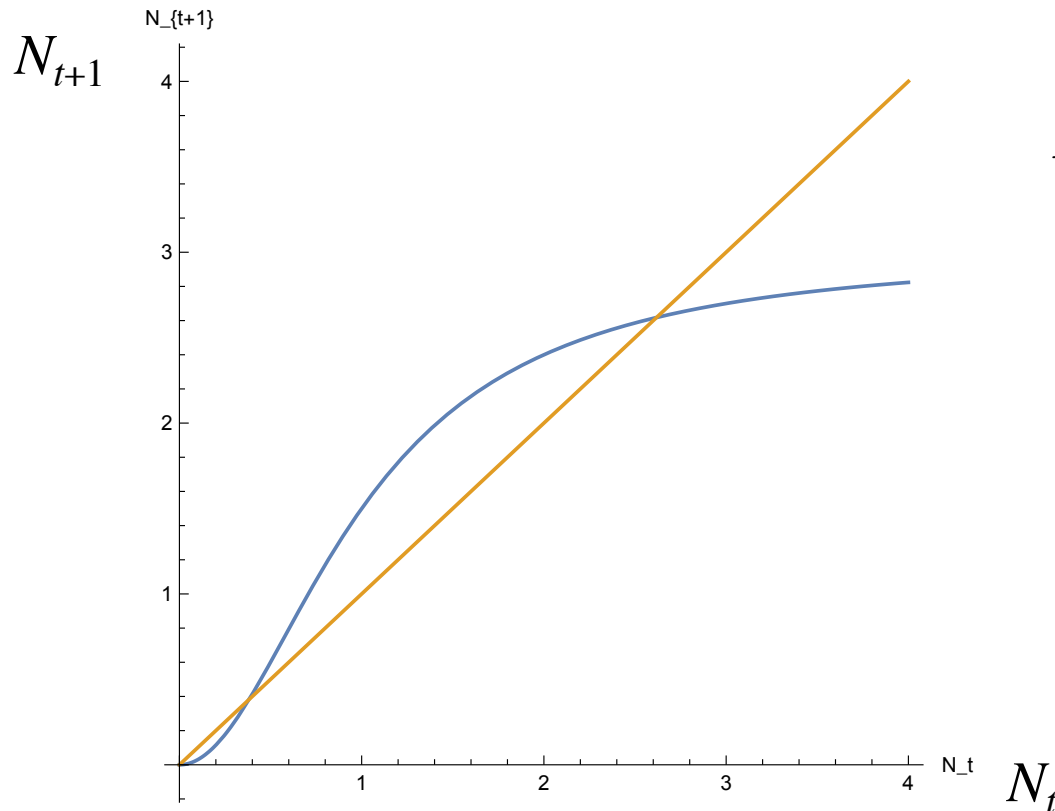
アリー効果

$$N_{t+1} = r N_t$$

↑

個体密度 N_t が低くなると1個体あたりの増殖率 r が1以下に減少する効果をアリー Allee 効果という。密度依存効果の1つ。

個体密度が低くなると、交配相手が見つからず繁殖が困難になる効果。希少動物に当てはまる場合が多い。



$$N_{t+1} = f(N_t) = \frac{aN_t^2}{b + N_t^2}$$

個体密度が小さいとき ($N_t \ll 1$)

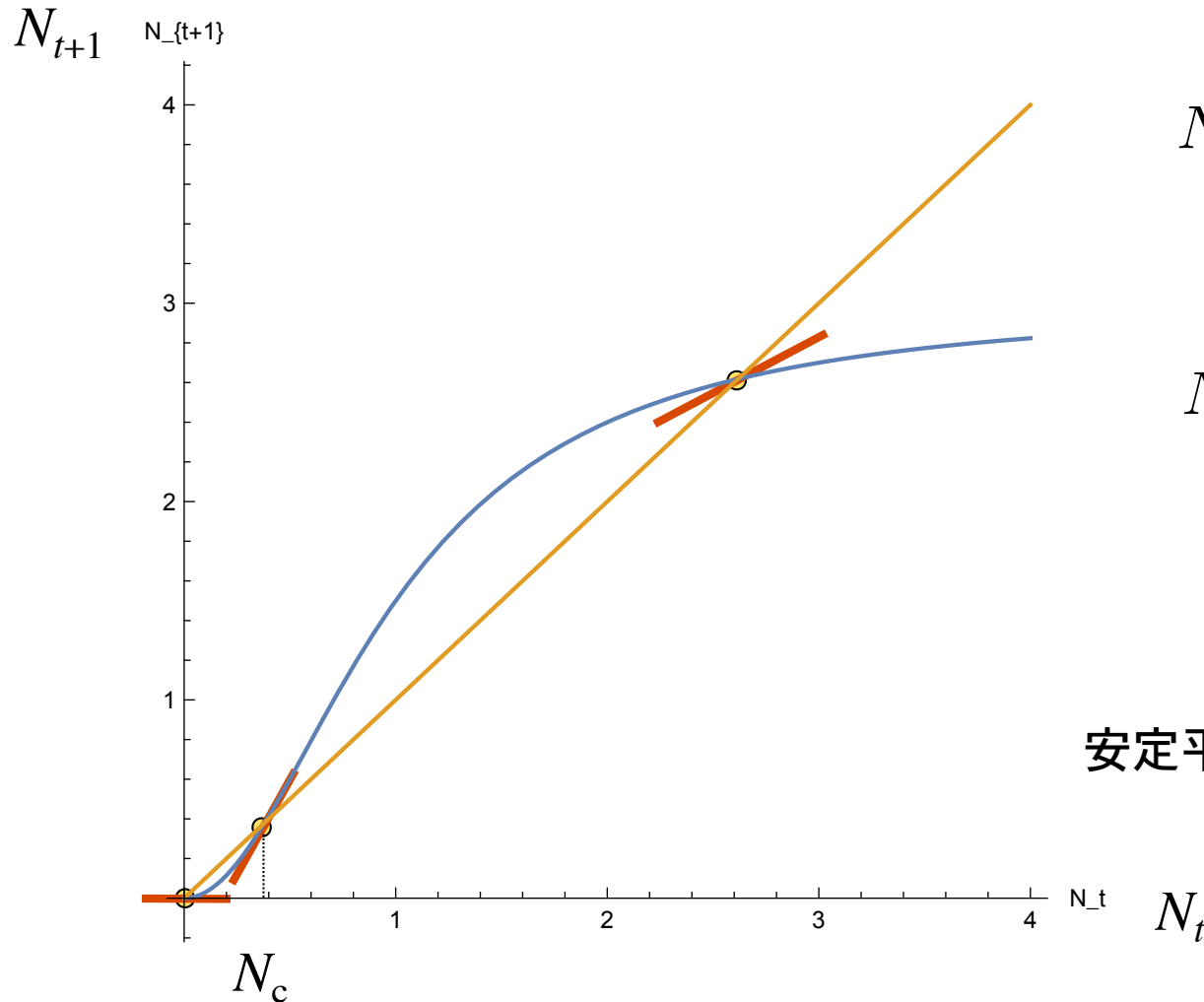
$$N_{t+1} \approx \frac{a}{b} N_t^2$$

個体密度が大きいとき ($N_t \gg 1$)

$$N_{t+1} \approx a$$

アリー効果数値計算例

アリー効果が働く場合、一旦、個体密度が閾値よりも小さくなると集団は絶滅へ向かう



$$N_{t+1} = f(N_t) = \frac{aN_t^2}{b + N_t^2}$$

$$N^* = f(N^*) = \frac{aN^{*2}}{b + N^{*2}}$$

平衡点は3つ

安定平衡点は2つ。残り1つは不安定。

問題

1) Ricker ロジスティックモデルに関して問いに答えよ。

$$N_{t+1} = \exp \left[r \left(1 - \frac{N_t}{K} \right) \right] N_t$$

- パラメータ K は N_t の振る舞いに定性的に影響しないことを示せ
- N_t の振る舞いを調べるために Ricker diagram を作成せよ
- 平衡点 (2 つある) を求めて局所安定性解析を行い、安定性のパラメータ依存性を調べよ
- 平衡点が安定な場合と不安定な場合の2通りの数値計算を行い、視覚化せよ

2) 次式はいずれも N_t に関する離散時間のモデルである。それぞれ、1) 非自明な平衡点を求め、2) 局所安定性解析を行って非自明な平衡点が不安定になるパラメータの条件を求めよ。なおすべてのパラメータ値は正とする ($a, b, K > 0$)。

$$\text{a) } N_{t+1} = \frac{rN_t}{(1 + aN_t)^b}$$

$$\text{b) } N_{t+1} = \frac{rN_t}{1 + (N_t/K)^b}$$

Python による数値計算

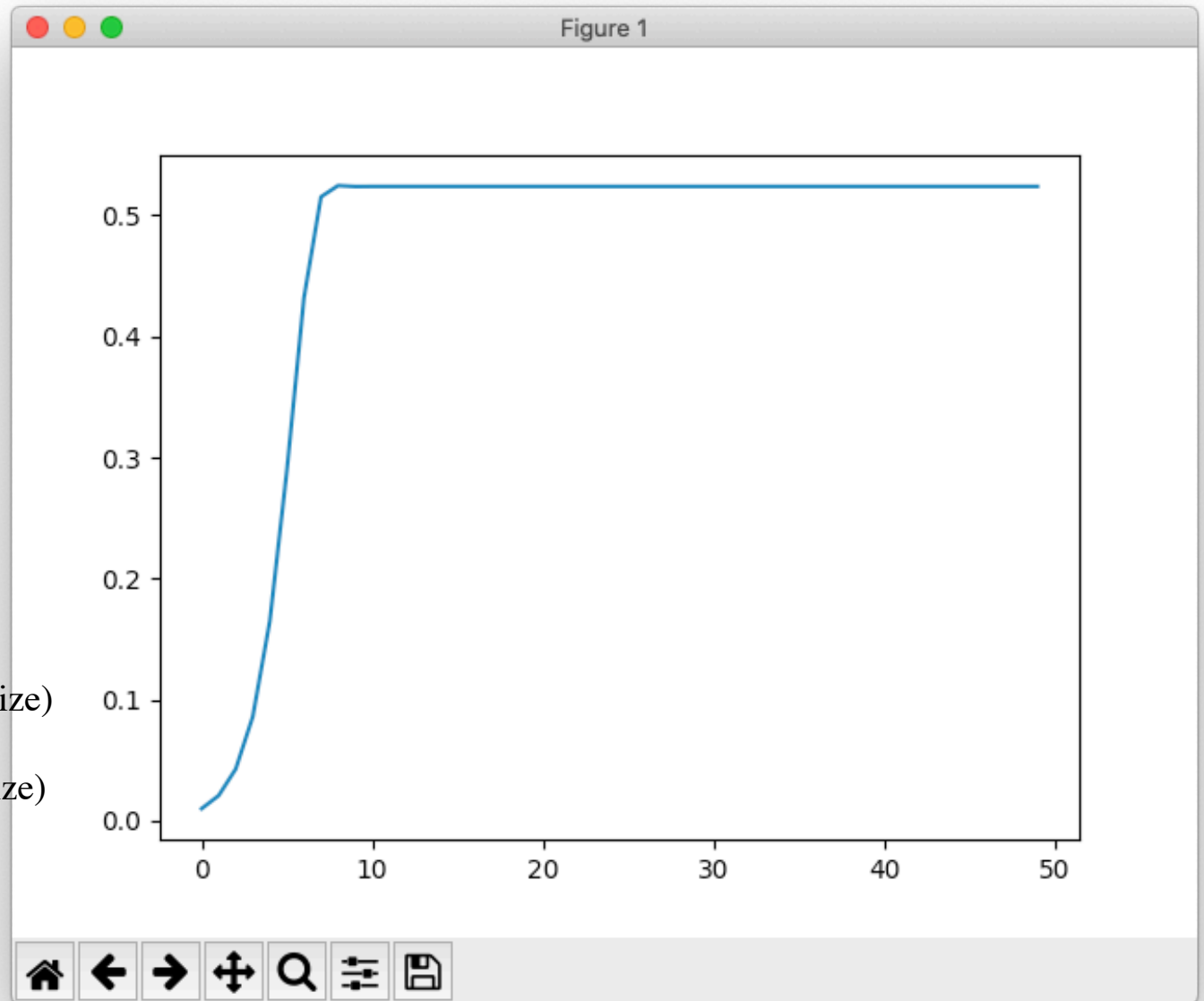
```
import matplotlib.pyplot as plt

def myFunc(n):
    r = 2.1
    tmp = r*(1-n)*n
    return tmp

tEnd = 50
popSize = 0.01
t = 0
seqTime=[]
seqPopSize=[]

while t < tEnd:
    print(t, popSize)
    seqTime.append(t)
    seqPopSize.append(popSize)
    t += 1
    popSize = myFunc(popSize)

# Draw the dynamics
plt.plot(seqTime, seqPopSize)
plt.show()
```



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

```
def myFunc(n):
    tmp = r*(1-n)*n
    return tmp
```

```
r = 3.2
tEnd = 50
x = 0.01
t = 0
seqx = []
seqy = []
seqx.append(x)
seqy.append(0)
```

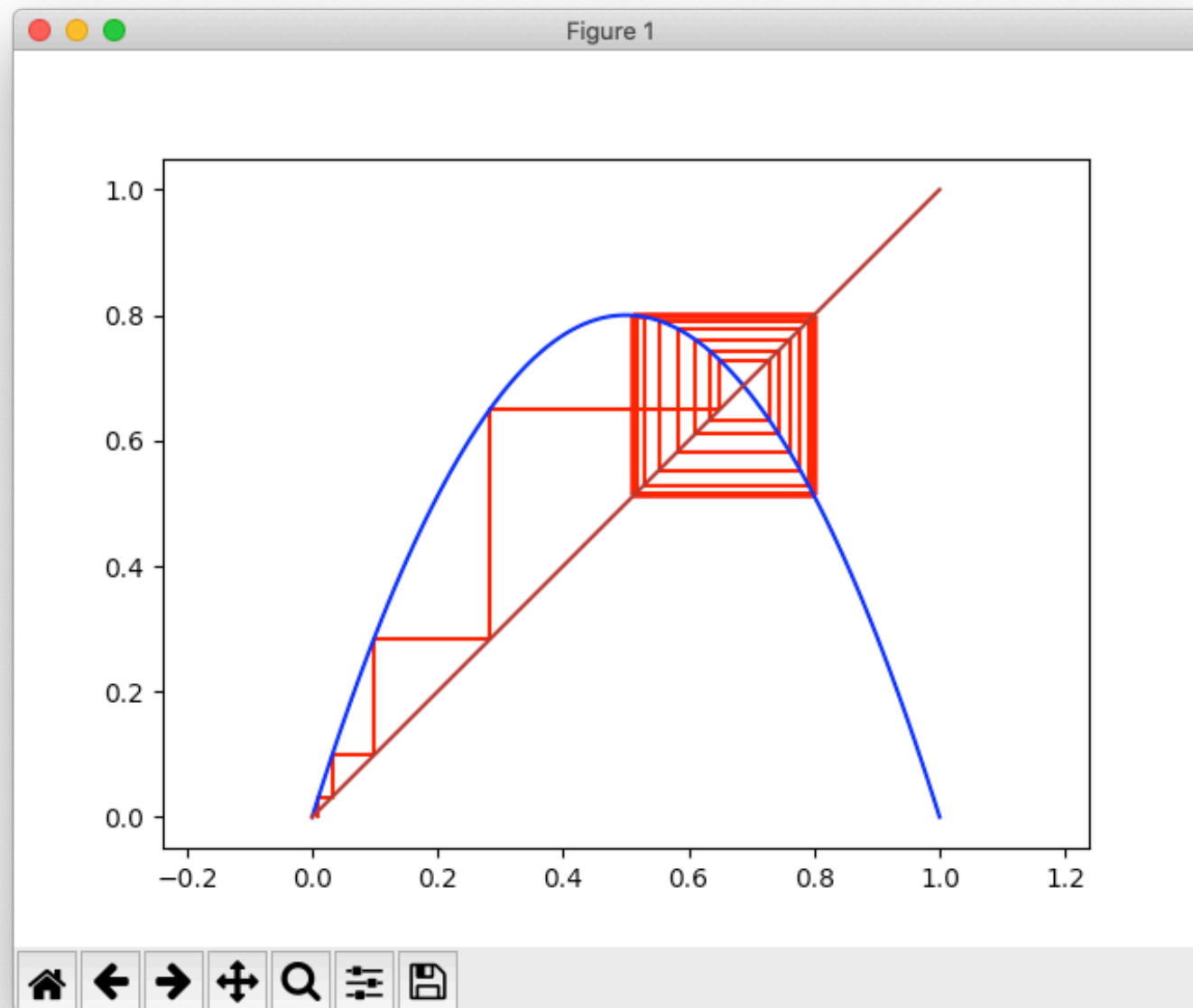
```
while t <= tEnd:
    t += 1
    xd = myFunc(x)
    seqx.append(x)
    seqy.append(xd)
    seqx.append(xd)
    seqy.append(xd)
    x = xd
```

```
plt.plot(seqx, seqy, color="red")
xt = np.linspace(0, 1, 100)
xtd = myFunc(xt)
plt.plot(xt, xtd, color="blue")
```

```
xt2=xt
plt.plot(xt, xt2, color="brown")
```

```
plt.axis('equal')
plt.show()
```

Python による数値計算



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

```
def myFunc(r, x):
    tmp = r*(1-x)*x
    return tmp

def myAttractor(rx):
    tIdle = 2000
    tEnd = 100
    x = 0.1
    seq = []

    for t in range(tIdle):
        x = myFunc(rx, x)

    for t in range(tEnd):
        seq.append(x)
        x = myFunc(rx, x)

    seqList = list(set(seq))
    seqListUnique = np.unique(seqList)
    return seqListUnique
```

```
rStart = 2
rEnd = 3.8
rIntv = 0.002
```

```
rs = np.arange(rStart, rEnd, rIntv)
seqrs = []
seqConvergence = []
```

```
for r in rs:
    print("r = ", r)
    tmp = myAttractor(r)
    for i in range(len(tmp)):
        seqrs.append(r)
        seqConvergence.append(tmp[i])
```

```
plt.scatter(seqrs, seqConvergence, s = 1)
plt.show()
```

Python による数値計算

