発展

数理生物学演習

第５回　個体群動態の数理モデル（３）：
ロトカ-ボルテラ モデル

野下 浩司（Noshita, Koji）

noshita@morphometrics.jp

https://koji.noshita.net

理学研究院 数理生物学研究室

平衡点を数値的に見つける：ロジスティック成長モデル

ロジスティック成長モデル

$$\frac{dx}{dt} = r\left(1 - \frac{x}{K}\right)x$$

解析解 $x_1^* = 0, x_2^* = K$

xにおける接線の傾き

$$f'(x) = r\left(1 - \frac{2}{K}x\right)$$

ニュートン法に利用する漸化式

$$x_{i+1} = \frac{x_i^2}{2x_i - K}$$

```python
# 01-01. ロジスティック成長モデル

# パラメータ設定
K = 100

# 初期値
x = 90

for i in range(1000):
    x = x*x/(2*x-K)

print("解析解: ", 0, ",", K)
print("近似解: ", x)
```

```
# 出力
解析解:  0 , 100
近似解:  100.0
```

初期値に応じて，
平衡点のいずれかが数値的に求まる

平衡点を数値的に見つける：被食-捕食系

被食-捕食系

$$\begin{cases} \dfrac{dx}{dt} = ax - bxy \\ \dfrac{dy}{dt} = cxy - dy \end{cases}$$

解析解

$$(x_1^*, y_1^*) = (0,0), (x_2^*, y_2^*) = \left(\frac{d}{c}, \frac{a}{b}\right)$$

ヤコビ行列

$$\mathbf{J}(x, y) = \begin{pmatrix} \dfrac{\partial f_1}{dx} & \dfrac{\partial f_1}{dy} \\ \dfrac{\partial f_2}{dx} & \dfrac{\partial f_2}{dy} \end{pmatrix} = \begin{pmatrix} a - by & -bx \\ cy & cx - d \end{pmatrix}$$

ニュートン法に利用する漸化式

$$\mathbf{x}_{i+1} = \mathbf{x}_i - \mathbf{J}_{\mathbf{x}_i}^{-1}\mathbf{f}(\mathbf{x}_i)$$

$$\begin{pmatrix} x_{i+1} \\ y_{i+1} \end{pmatrix} = \frac{x_i y_i}{acx_i + bdy_i - ad}\begin{pmatrix} bd \\ ac \end{pmatrix}$$

```
# 01-02. 被食–捕食系

# パラメータ設定
a = 2
b = 3
c = 1
d = 2

# 初期値
x = 1.5
y = 0.5

for i in range(1000):
    x = b*d*x*y/(a*c*x+b*d*y-a*d)
    y = a*c*x*y/(a*c*x+b*d*y-a*d)

print("解析解: ", (0,0), ",", (d/c, a/b))
print("近似解: ", (x, y)
```

```
# 出力
解析解:  (0, 0) , (2.0, 0.6666666666666666)
近似解:  (2.0, 0.6666666666666666)
```


アイソクライン isocline

２種系の場合だと，$\dfrac{dx}{dt} = 0$,　または $\dfrac{dy}{dt} = 0$ を満たす直線や曲線のこと

アイソクラインを境界として個体数の増減のパタンが変わるので，これが相平面上にプロットできると全体のダイナミクスを見通しやすくなる

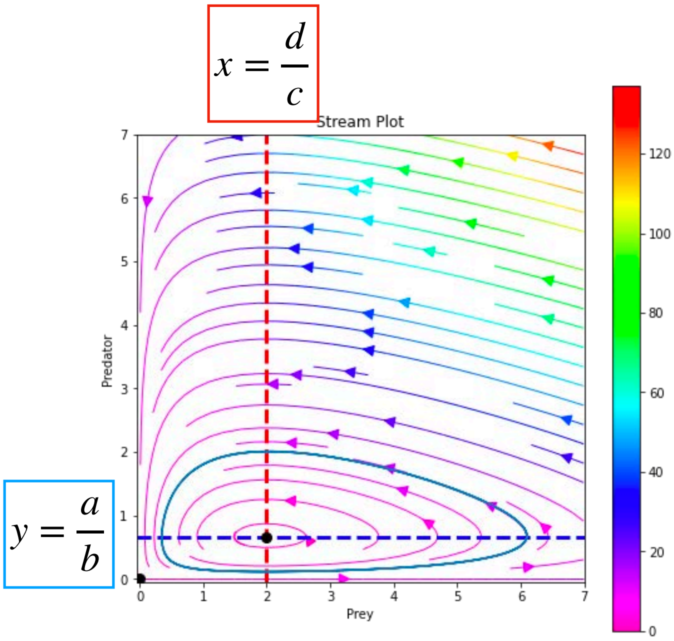
被食-捕食系

$$\begin{cases} \dfrac{dx}{dt} = ax - bxy \\ \dfrac{dy}{dt} = cxy - dy \end{cases}$$

アイソクライン

$$\frac{dx}{dt} = 0 \rightarrow y = \frac{a}{b}$$

$$\frac{dy}{dt} = 0 \rightarrow x = \frac{d}{c}$$



ロトカ-ボルテラ モデルの相図とアイソクラインを使ったダイナミクスの解析

```
# 01-03. 被食−捕食系のダイナミクスとアイソクライン
# モデルのパラメータ
a = 2.0
b = 3.0
c = 1.0
d = 2.0

# 初期値
x = 0.4
y = 0.4
t = 0.0

# 時間の設定
dt = 0.0001
tEnd = 10
iEnd = int(tEnd/dt)+1

t_list = [t]
x_list = [x]
y_list = [y]
for i in range(iEnd):
    t = dt*(i+1)
    x = x + dt*(a-b*y)*x
    y = y + dt*(c*x-d)*y

    t_list.append(t)
    x_list.append(x)
    y_list.append(y)
```

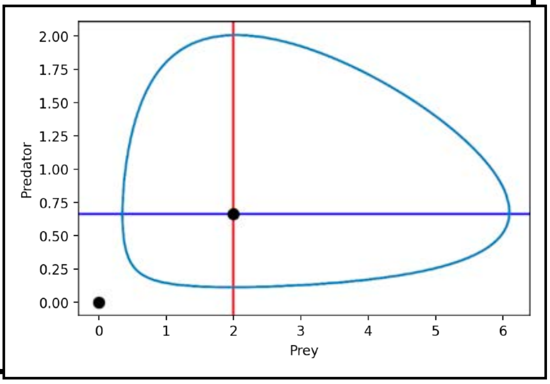
被食−捕食系の場合は，アイソクラインは水平，垂直の直線なのでplotではなくaxhline，axvlineを使っている．

平衡点を黒い丸でプロット，マーカーサイズも指定して視認性を上げている

```
# アイソクライン
plt.axhline(a/b, color = "b")
plt.axvline(d/c, color = "r")

# 平衡点
plt.plot(0,0, "ko",d/c, a/b ,"ko", markersize = 8)

# 相平面上でのダイナミクス
plt.plot(x_list, y_list)
plt.xlabel("Prey")
plt.ylabel("Predator")
```



左半分は，被食−捕食系のシミュレーション（演習資料台５回 #02-01）と（プロットの部分を除き）同じ

matplotlib.pyplot
- axhline(yの値，オプション)：水平な直線をプロット
- axvline(xの値，オプション)：垂直な直線をプロット
  ここでは色のオプションで青（b）と赤（r）を指定