

# 捕食・被食関係のシミュレーション

講師：高安 亮紀

第5回

# 目的

- ロトカ・ボルテラ方程式を用いて、捕食者と被食者の増減関係をシミュレーションする。
- 前回に引き続き、常微分方程式とその解法について学習し、MATLABのODEソルバを用い、プログラミングを行い、検証を行う。

※前回講義の人口変動シミュレーションを理解しておくこと。

# ロトカ・ボルテラモデル

- 捕食者と被食者の増減関係をモデル化し、その増殖速度を表現した非線形微分方程式

$$\left\{ \begin{array}{l} \frac{dH}{dt} = \gamma_H H - aHP \\ \frac{dP}{dt} = -\gamma_P P + bHP \end{array} \right. \quad \begin{array}{l} \text{被食者の個体密度変化} \\ = \text{捕食がない場合の増加} - \text{捕食による減少} \\ \\ \text{捕食者の個体密度変化} \\ = -\text{捕食がない場合の減少} + \text{捕食による増加} \end{array}$$

H: 被食者の個体数密度, P: 捕食者の個体数密度

a: 捕食率, b: 転換率

$\gamma_H$ : 被食者の1個体当たりの増加率

$\gamma_P$ : 捕食者の死亡率

# ロトカ・ボルテラモデル

- 被食者と捕食者が出会う確率は、両者の密度の積  $H \times P$  に比例する。
- 捕食者がいない場合、被食者は指数増加。

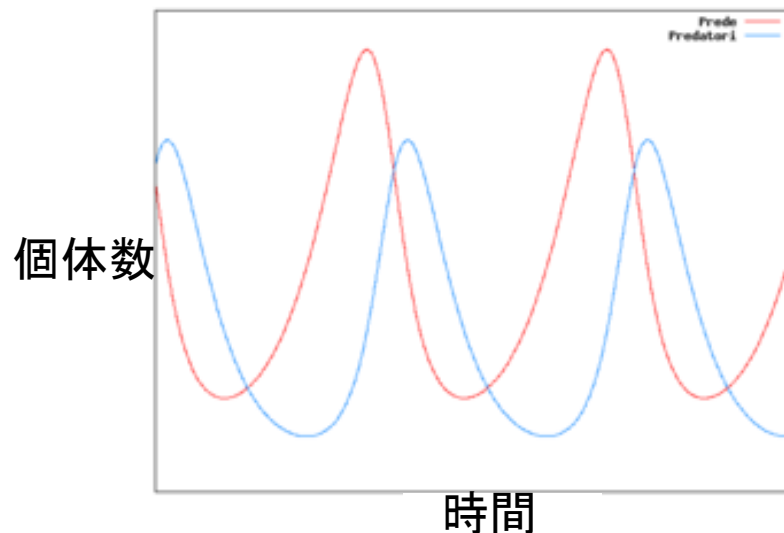
$$\frac{dH}{dt} = \gamma_H H$$

※前回講義の人口変動シミュレーションの式と同様

- 捕食なしでは、捕食者は指数減少。

$$\frac{dP}{dt} = -\gamma_P P$$

・図のように捕食者が増加すると、被食者が減少し、さらに捕食者が減少する、という時間変化を示すことができる。



青: 捕食者  
赤: 被食者

# 常微分方程式の数値解法

- コンピュータを用いて数値的に微分方程式

$$\frac{dy}{dx} = f(x, y)$$

の解を求める。

- 本講義では、MATLABのODEソルバなどの基礎公式になっているルンゲ・クッタ法について説明する。

# ルンゲ・クッタ法

- 微分方程式の数値解法として、4次のルンゲ・クッタ法を使用。
- 基本的な考え方はオイラー法と同じ。

$$k_1 = f(x_i, y_i)$$

h: 刻み幅

$$k_2 = f\left(x_i + \frac{h}{2}, y_i + \frac{h}{2}k_1\right)$$

$$k_3 = f\left(x_i + \frac{h}{2}, y_i + \frac{h}{2}k_2\right)$$

$$k_4 = f(x_i + h, y_i + hk_3)$$

$$y_{i+1} = y_i + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

$$x_{i+1} = x_i + h$$

# ODEソルバ

- MATLABでは、微分方程式の解を求めるものとして、odeソルバが用意されている。ode45では、少し工夫したルンゲ・クッタ法が使われている。

| ソルバ     | 問題のタイプ | 精度の度合い | 使用法                                           |
|---------|--------|--------|-----------------------------------------------|
| ode45   | ノンステッフ | 中      | ほとんどの場合に使います。最初に実行してみるソルバ。                    |
| ode23   | ノンステッフ | 低      | 粗い誤差基準を使ったり、わずかにステッフな問題を解く場合                  |
| ode113  | ノンステッフ | 低から高   | 精密な許容誤差を使ったり、計算の集中したODEファイルを解く場合              |
| ode15s  | ステッフ   | 低から中   | ode45が遅い(ステッフなシステム)、または大きい質量行列がある場合           |
| ode23s  | ステッフ   | 低      | ステッフなシステムを解くために粗い許容誤差を使ったり、または、大きい定数質量行列がある場合 |
| ode23t  | 少しステッフ | 低      | 問題がある程度ステッフなだけで、数値減衰が無い解を必要とする場合              |
| ode23tb | ステッフ   | 低      | ステッフなシステムを解くために、大きな誤差の許容量を使うか、または、質量行列がある場合   |

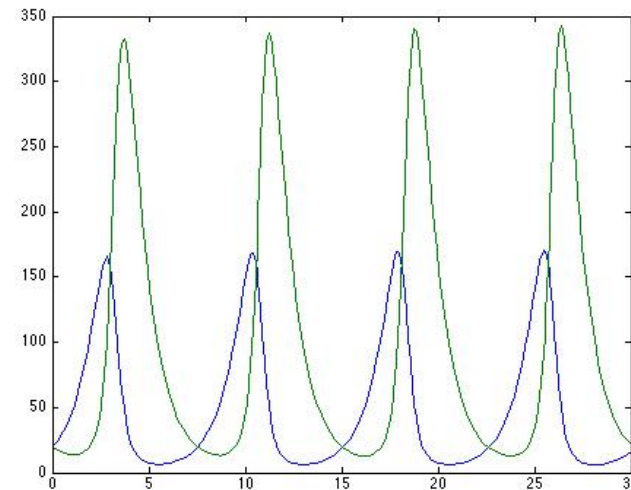
※ODEソルバで使われるアルゴリズムは、精度と設計されたシステムのタイプ(stiffの度合い)により異なる。  
 堅い方程式(stiff equation)とは、解の緩やかな変化と急激な変化が混在する方程式で、刻み幅を小さく設定しなければならない場合がある。

# ODEソルバでロトカ・ボルテラ方程式を解く

lotka1.m

```
function dy = lotka1(t,y)
rH = 1; % 被食者の1個体当たりの増加率
rP = 1; % 捕食者の死亡率
a = 0.01; % 捕食率
b = 0.02; % 転換率
dy = zeros(2,1);
dy(1) = (rH - a*y(2))*y(1);
dy(2) = (-rP + b*y(1))*y(2);
```

```
>> t = [0, 30]; % tの範囲
>> y0 = [20; 20]; % 個体密度初期値
>> [t,y] = ode45('lotka1',t,y0);
>> plot(t,y)
```

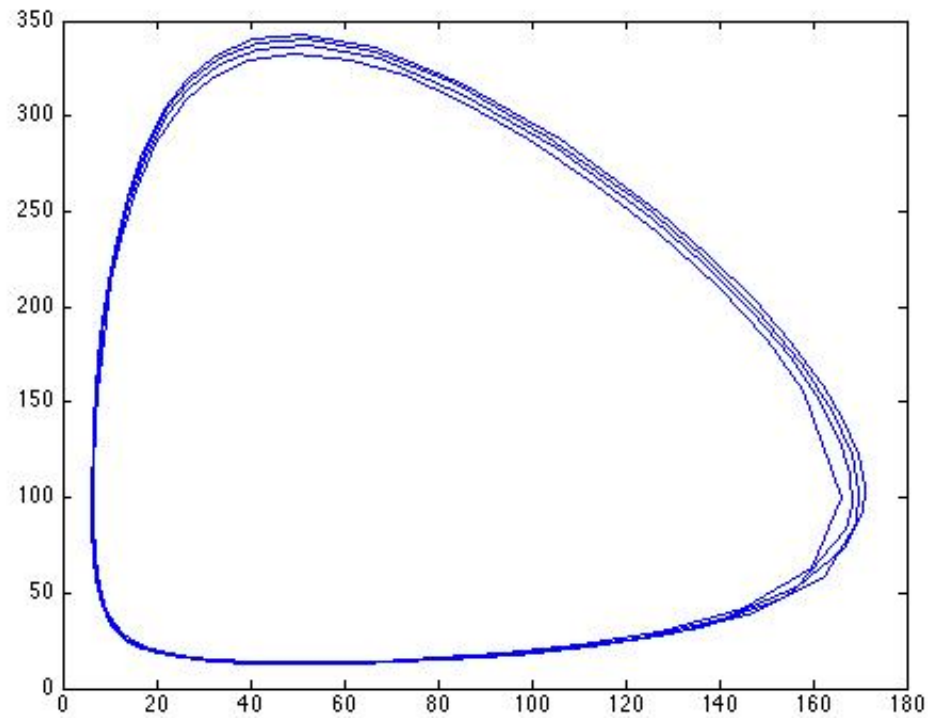




# 解の軌道

- ・必ず反時計回りの閉軌道(周期解)になる。  
→ロトカ・ボルテラモデルの解は必ず周期解になることが証明できる

```
>> plot(y(:,1),y(:,2))
```



# 実習1:うさぎと山猫

- 右の図は、実際のうさぎと山猫の個体データに基づいている。このデータを用いて、個体数の変化をシミュレーションしましょう。
- ヒント
  1. lotka1.mをもとにlotka2.mを作成
  2. 捕食率などのパラメータを適当に設定
  3. ODEソルバで解く

| 年    | うさぎ | 山猫 |
|------|-----|----|
| 1850 | 55  | 2  |
| 1851 | 80  | 1  |
| 1852 | 80  | 1  |
| 1853 | 90  | 1  |
| 1854 | 70  | 5  |
| 1855 | 80  | 13 |
| 1856 | 95  | 16 |
| 1857 | 75  | 25 |
| 1858 | 30  | 14 |
| 1859 | 15  | 8  |
| 1860 | 20  | 3  |

# 実習2

- 捕食・被食に関する実際のデータを調査し、シミュレーションしてみましょう。
- データはインターネットで調べると色々あります。
  - 魚の漁獲量と魚の数
  - メダカとミジンコ等