数理生物学演習

第3回　指数増殖，ロジスティック成長

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

第3回：指数増殖，ロジスティック成長

本日の目標

- 微分方程式を数値的に解く
- 結果を解析解と比較する

指数増殖

ある集団のサイズ（個体数）を x とし，
その増加速度（dx/dt）が集団サイズ $x(t)$ に比例する場合，
ダイナミクスは以下の式で表すことができる．

$$\frac{dx}{dt} = ax$$

初期条件

$$x(0) = x_0$$

a：単位時間あたり一個体あたりの増加率（マルサス係数）

$$x(t) = x_0 e^{at}$$

解いてみよう

ロジスティック成長

指数増殖におけるマルサス係数（a）がr(1-x/K)で置き換えられている．
つまり，個体数が増加するに伴い単位時間あたり一個体あたりの増加率が減少する．

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

初期条件

$$x(0) = x_0$$

r：内的自然増加率．個体群密度が十分に小さい（~0）場合の増加率．
K：環境収容力．ある環境が維持できる個体数．利用できる資源量や
空間サイズなどを反映する．ここではK>0とする．

$$x(t) = \frac{K}{1 + \left(\frac{K}{x_0} - 1\right)e^{-rt}}$$

解いてみよう

実際にプログラムを組んでみよう！

条件分岐 if文

特定の条件下でのみ実行したい処理を書く！

・if文

```
if 条件文:
  文１
  文2
  ⋮
  文n
```

条件文が真ならば,
文１～文nを実行する.
（偽ならば何もしない）

特に注意！！
インデントされた文がif文のブロックとみなされる

```
# if文
for i in range(30):
    if i > 15:
        print(i)
```

出力
16
17
⋮
29

iが15より大きければ
iを画面に表示する

forループとif文が使えればたいていのプログラムが組める！

関係演算子と論理演算子

関係演算子
- > より大きい
- >= 以上
- < より小さい
- <= 以下
- == 等しい
- != 等しくない

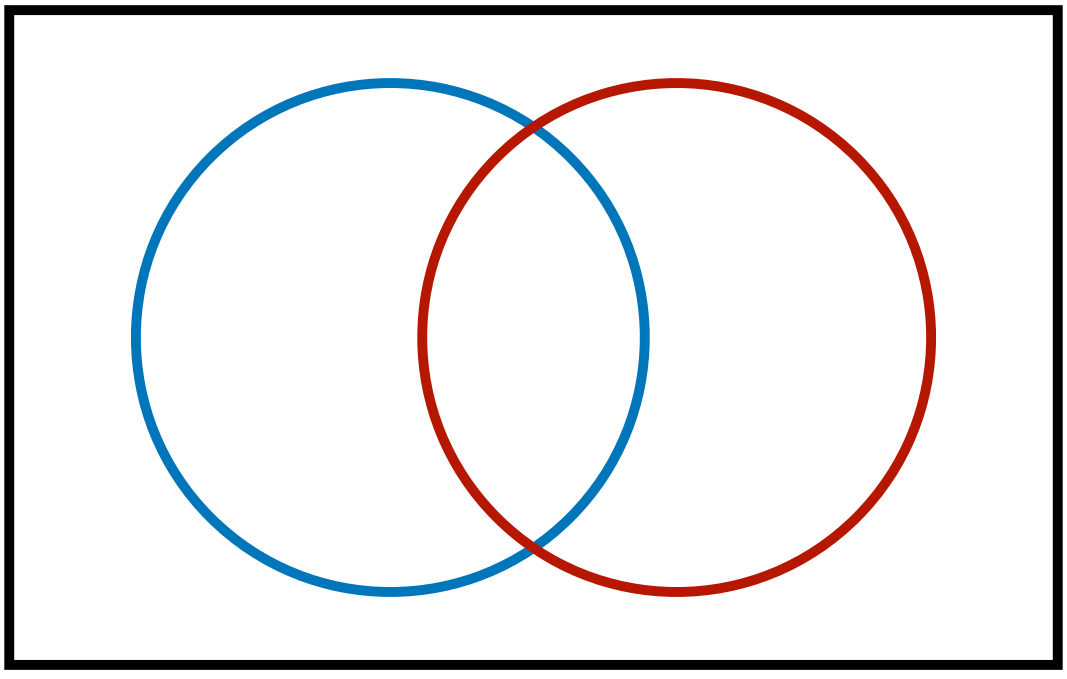
論理演算子
- and かつ（論理積）
- or または（論理和）
- not ではない（否定）

- print(“文1”, “文2”, …,“文n”)
  文1 文2 … 文n を（改行せず）表示する

```
# 関係演算子と論理演算子
for i in range(10):
    print("i=", i)
    if i > 5:
        print("  iは5より大きい")
    if i == 3:
        print("  iは3と等しい")
    if i>=3 and i<=6:
        print("  iは3以上6以下")
    if not (i==1 or i==2):
        print("  iは1または2ではない")
```

出力
```
i= 0
  iは1または2ではない
i= 1
i= 2
i= 3
  iは3と等しい
  iは3以上6以下
  iは1または2ではない
i= 4
  iは3以上6以下
  iは1または2ではない
i= 5
  iは3以上6以下
  iは1または2ではない
i= 6
  iは5より大きい
  iは3以上6以下
  iは1または2ではない
i= 7
  iは5より大きい
  iは1または2ではない
i= 8
  iは5より大きい
  iは1または2ではない
i= 9
  iは5より大きい
  iは1または2ではない
```

関係演算子と論理演算子



悩んだらベン図を描いてみる

if文を鍛える（１）

・if-else文
```
if 条件文:
  文１
  文2
  ⋮
  文m
else:
  文’１
  文’2
  ⋮
  文’n
```

条件文が
真ならば文１～文mを実行
偽ならば文’１～文’nを実行

```
# 偶奇判定
for i in range(50):
    if i%2==0:
        print(”even”)
    else:
        print(”odd”)
```

出力
even
odd
even
odd
・・・
odd

iが偶数ならばevenを
奇数ならばoddを
画面に出力する

注意
elseはifと同じ深さ

if文を鍛える（２）

・if-else-if文
```
if条件文1:
  文１
  文2
  ⋮
  文m
elif 条件文2:
  文’１
  文’2
  ⋮
  文’n
```

```
#3の倍数
for i in range(50):
    if i%3==0:
        print(”3の倍数”)
    elif i%3==1:
        print(”余り1”)
    else:
        print(””)
```

出力
余り1

3の倍数
余り1

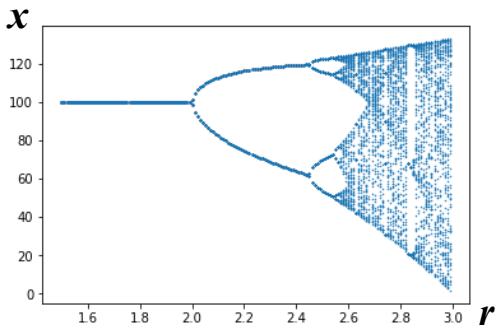
3の倍数
・・・
余り1

iを3で割った余りが
0のとき，「３の倍数」
1のとき，「余り1」
をそれぞれ画面に表示し，
2のとき，何もしない

条件文1が真ならば文１～文mを実行
条件文1が偽かつ条件文2が真ならば文’1～文’nを実行
（条件文1も2も偽ならば何もしない）

分岐図 ifバージョン

```
# 分岐図 ifバージョン
%matplotlib inline
import matplotlib.pyplot as plt

K = 100

rList = []
xList = []

for i in range(150,300):
    x0 = 10
    r = i/100
    x=x0
    for t in range(1000):
        xx =x + r* (1-x/K)*x
        x = xx
        if t>900:
            rList.append(r)
            xList.append(x)

plt.plot(rList, xList, '.', markersize="1")
```



今回は全部で1000ステップ計算する

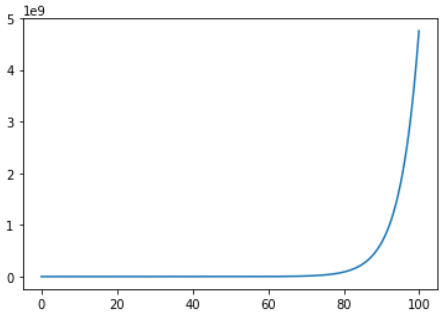
最後の100ステップをリストに格納

matplotlib.pyplot.plotのオプション
- markersize
  float型でマーカーサイズを指定

その他のオプションについては
https://matplotlib.org/api/pyplot_summary.html を参照

指数増殖（解析解）

```
# 指数増殖

%matplotlib inline
import math
import matplotlib.pyplot as plt

a = 0.2
x0 = 10

dt = 0.1
tList = []
xList = []
for i in range(1000):
    t = dt*i
    x = x0*math.exp(a*t)
    tList.append(t)
    xList.append(x)

plt.plot(tList, xList)
```

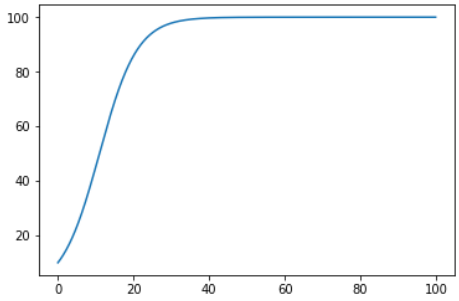


注意：dtを変化させたら，その分iの最大値を変えなければ，観察している時間の長さが変わる.

いろいろなオプションを調整して見やすくしてみよう.

ロジスティック成長（解析解）

```
# ロジスティック成長
指数増殖のプログラムを参考に
自分で考えてみてください
```



こんな感じのがプロットしたい

オイラー法 Euler's method

- 計算機は直接は微分や積分ができない
- 微分方程式を離散化し計算機が扱えるようにする

目的

微分方程式

$$\frac{dx}{dt} = f(x,t) \quad \cdots \quad (1)$$

を数値的に解きたい.

微分の定義から

$$\frac{dx}{dt} = \lim_{\Delta t \to 0} \frac{x(t+\Delta t) - x(t)}{\Delta t}$$

なので，Δt が十分に小さければ，

(1) は近似的に

$$f(x,t) \approx \frac{x(t+\Delta t) - x(t)}{\Delta t}$$

変形すると

$$x(t+\Delta t) \approx x(t) + f(x,t)\Delta t$$

$x(0) = x_0$ とし，$x_1, x_2, \ldots, x_n$ を考えると

$$x_n \approx x_{n-1} + f(x_{n-1}, t_{n-1})\Delta t$$

ただし，$t_n = \Delta t \cdot n$

ここで

$$X_n = X_{n-1} + f(X_{n-1}, t_{n-1})\Delta t$$

として，この X_n を x_n の近似値として採用する.

- 時間方向の刻み幅 Δt を小さくすることである程度誤差を小さくできる
- オイラー法はあまり精度の良い近似法ではない

最終的なプログラムは
前回の差分方程式と似たものになる

指数増殖の離散化

指数増殖

$$\frac{dx}{dt} = ax$$

微分の近似（Δtは十分小さいとする）

$$\frac{dx}{dt} \approx \frac{x(t+\Delta t) - x(t)}{\Delta t}$$

↓

$$ax(t) \approx \frac{x(t+\Delta t) - x(t)}{\Delta t}$$

式を整理

$$x(t+\Delta t) \approx x(t) + ax(t)\Delta t$$

↓ $x(0) = x_0$ とし，$x_1, x_2, \ldots, x_n$
また，$t_n = \Delta t \cdot n$

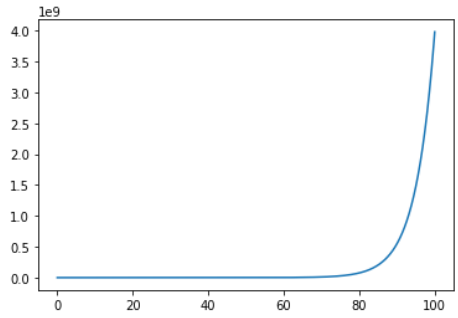
$$x_{n+1} \approx x_n + ax_n\Delta t$$

↓

$$X_{n+1} = X_n + aX_n\Delta t$$

指数増殖

```
# 指数増殖（数値解）
%matplotlib inline
import math
import matplotlib.pyplot as plt

a = 0.2
x0 = 10

dt = 0.1

t = 0
x = x0
xList = [x]
tList = [t]
for i in range(1000):
    t = dt*(i+1)
    xx = x + a*x*dt
    x = xx
    tList.append(t)
    xList.append(x)

plt.plot(tList, xList)
```



注意：dtを変化させたら，その分iの最大値を変えなければ，観察している時間の長さが変わる．

オイラー法による近似

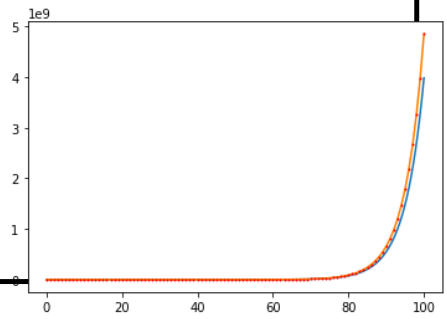
指数増殖

```
# 指数増殖（解析解と数値解の比較）
a = 0.2
x0 = 10

tEnd = 100

#
# 時間幅
#
dtA = 0.1
iEndA = int(tEnd/dtA)+1

dtN = 0.1
iEndN = int(tEnd/dtN)

# 解析解
tListA = []
xListA = []
for i in range(iEndA):
    t = dtA*i
    x = x0*math.exp(a*t)
    tListA.append(t)
    xListA.append(x)
```

dtNを0.1～0.001の範囲で変えて，どの程度解析解と合うかを検討してみる

```
# 数値解
t = 0
x = x0
xListN = [x]
tListN = [t]
for i in range(iEndN):
    t = dtN*(i+1)
    xx = x + a*x*dtN
    x = xx
    tListN.append(t)
    xListN.append(x)
```

出力例



ロジスティック成長

3-7.ロジスティック成長
指数増殖のプログラムを参考に
自分で考えてみてください

本日の課題

注意：氏名，学籍番号，所属を必ず書く！

1. 指数増殖，ロジスティック成長モデルを解析的に解け.
2. ロジスティック成長モデルについて１．で求めた解析解とオイラー法により近似した数値解を一つの図にプロットせよ．（その際のパラメータや初期値も示すこと）
3. 2．の図について，時間方向の刻み幅⊿tを様々に変化させ，その影響を考察せよ.
4. 質問，意見，要望等をどうぞ.

課題をPDFファイルにまとめて，Moodleにて提出すること

PDFファイルでの保存（Word）

https://koji.noshita.net/page/compbio/tips/tipspdfsave/

Webページ参照

ファイルサイズが5MB以下になるように調整すること
（Moodleでサイズの制限をおこなっています）

次回予告

第4回：ロトカ-ボルテラ モデル

５月１３日

（4月２９日，５月６日は休日のため演習はありません）

復習推奨

- ロトカ-ボルテラ モデル
- 力学系の局所安定性解析