

確定的モデルによる シミュレーション

23j1-214

教科書P80-81

この時間の目標

- 確定的モデルによるシミュレーション
- S モデル化とシミュレーションについてよく理解でき、活用方法を考えることができた
- A モデル化とシミュレーションについてよく理解できた
- B モデル化とシミュレーションについて理解できた
- C モデル化とシミュレーションについて理解できなかった

確定的モデルによる シミュレーション

確定的モデルによるシミュレーション

- 条件はあらかじめ決まっている(確定している)
- シミュレーションの結果を見て人間が判断する

問題

- 秒速40m/s(時速144km/h)で真上に投げた物体は何秒後に落ちてくるか。重力加速度を9.8m/s²とする。

下向きにかかる力で1秒間で遅くなる速度

- 0秒後

• 速度:40

位置:0

- 1秒後

• 速度:40-9.8=30.2

位置:0+40

- 2秒後

• 速度:30.2-9.8=20.4

位置:40+30.2=70.2

解法

- 初期条件を入力
- あとは計算させる
 - 規則性を見つけて式で表現
- 1秒後の式を作る
 - 速度は1秒前より9.8減る
 - 位置は1秒前の位置と速度を足す

	A	B	C	D
1	40	9.8		
2				
3	時間 T	速度 V	位置 P	
4	0		0	
5	1	30.2	=B4+C4	
6	2			
7	3			
8	4			
9	5			
10	6			
11	7			
12	8			
13	9			
14	10			
15	11			
16	12			
17				

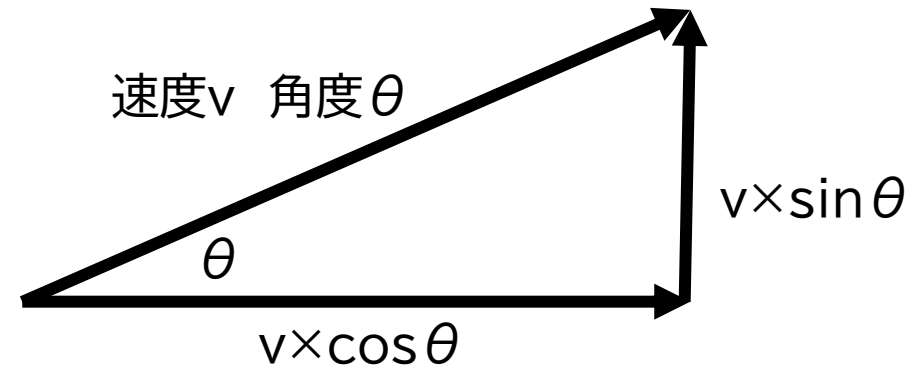
解法

- 初期条件を入力
- あとは計算させる
- 1秒後の式2つをコピー
 - フィルハンドルを利用
- 位置が0になるのはいつ？

	A	B	C	D
1	40	9.8		
2				
3	時間	速度	位置	
4	0	40	0	
5	1	30.2	40	
6	2	20.4	70.2	
7	3	10.6	90.6	
8	4	0.8	101.2	
9	5	-9	102	
10	6	-18.8	93	
11	7	-28.6	74.2	
12	8	-38.4	45.6	
13	9	-48.2	7.2	
14	10	-58	-41	
15	11	-67.8	-99	
16	12	-77.6	-166.8	
17				

何度で投げると遠くに飛ぶのか

- 三角比を利用
- 垂直・水平に分けて考える
 - 水平:一定の速度
 - 垂直:重力の影響
- シミュレーションには数学は重要



確定的モデルによる シミュレーション

PythonとExcelでやってみよう

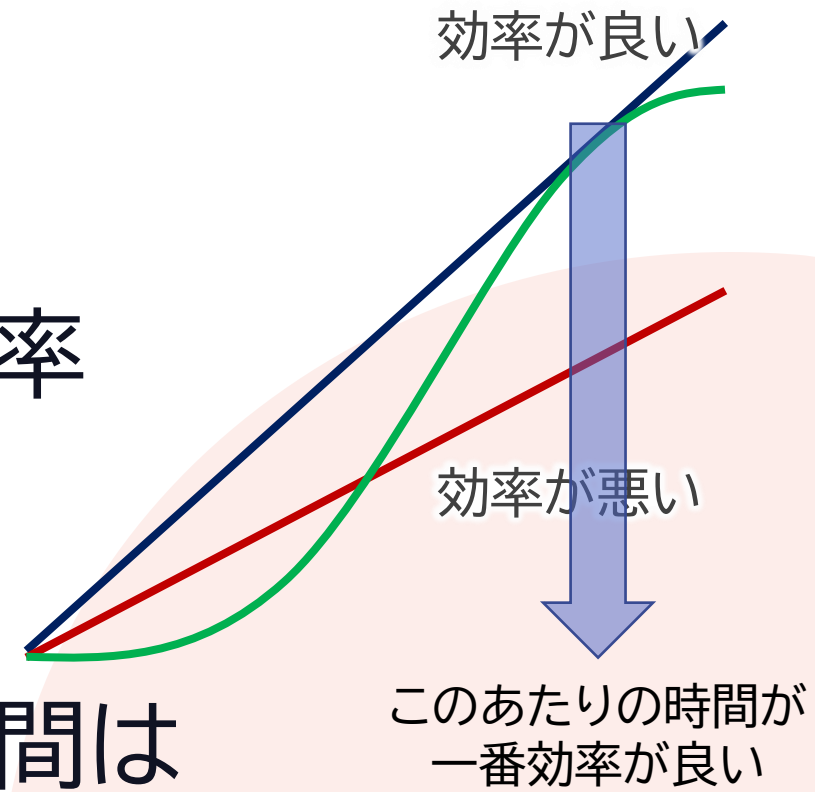
預金の複利計算(モデル)

- 複利法
 - 元金に利息を加算したものを次の期間の元金として利息計算
- 次期の金額 = 現在の金額 + 利息
- 利息 = 現在の金額 × 利率
- 次期の金額 = 現在の金額 + 現在の金額 × 利率

- 知りたいこと 20年後にはいくらになるのか？

生命体の増加シミュレーション(モデル)

- ある空間の中の生命体の増加について
 - 個体が多ければ増加する
 - 個体数が多すぎると減少する
 - 増加数 = 個体数 × 増加率
 - 減少率 = (個体数 / 環境収容数) × 増加率
 - 減少数 = 個体数 × 減少率
- 知りたいこと 一番効率が良い生育時間は



実習の目的

- 実際にシミュレーションしてみる
 - 数値を変えるとどうなるか試してみる
 - どこを変えたら良いかを理解する
- 仕組みがどうなっているか考える
 - 理解すべきスライドには ☆重要 マークが付いている
 - 式の意味や作り方をしっかり読んで考える
- できたらOKの課題ではない
 - 何をしようとしているのか/今の操作で何ができたのか
 - 考えながら/相方と意見交換しながら

今日の進め方

- 複利法のシミュレーション
 - Excelでシミュレーション
 - Pythonでシミュレーション
- 生命体の増加シミュレーション
 - Excelでシミュレーション
 - Pythonでシミュレーション
- 情報 I 第14回[シミュレーション実習1]

少なくとも
Excelと
Pythonを
一つずつ

Pythonは
JupyterNotebook

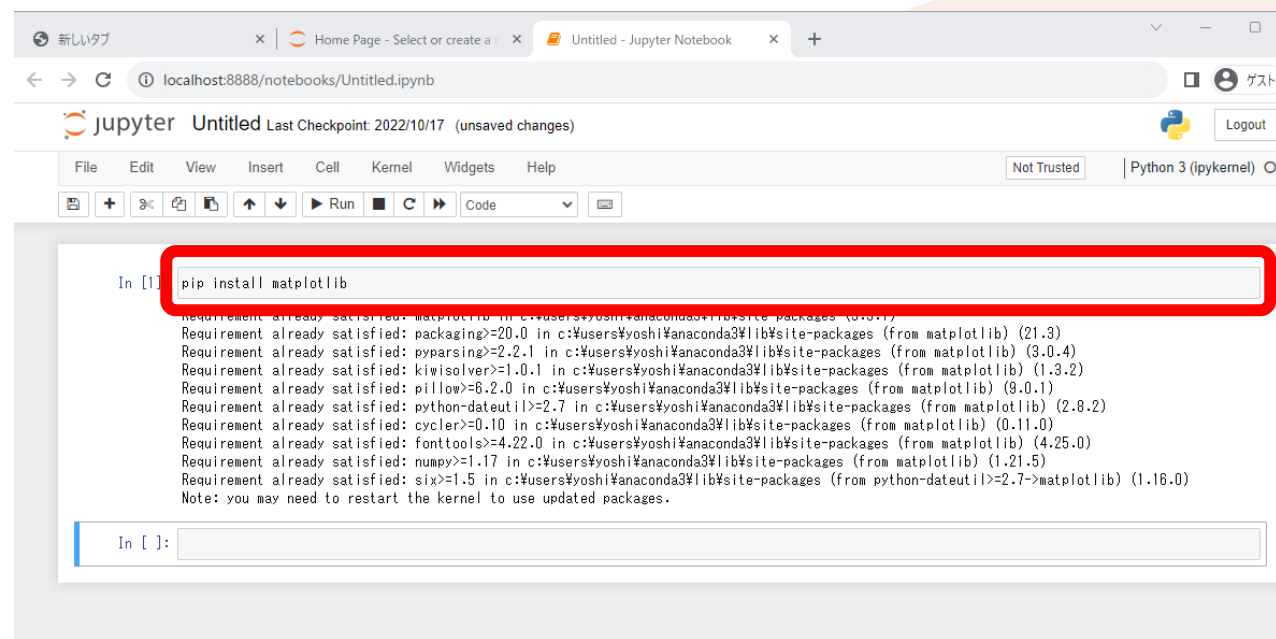
Excelは
スタート→Excel
→[空白のブック]
から始める

Jupyter Notebookにおまじない

1. Jupyter Notebookのセルに以下の文字列を入力

```
pip install matplotlib
```

2. [Shift]+[Enter]



```
In [1]: pip install matplotlib
Requirement already satisfied: matplotlib in c:\users\yoshi\anaconda3\lib\site-packages (3.7.1)
Requirement already satisfied: packaging>=20.0 in c:\users\yoshi\anaconda3\lib\site-packages (from matplotlib) (21.3)
Requirement already satisfied: pyparsing>=2.2.1 in c:\users\yoshi\anaconda3\lib\site-packages (from matplotlib) (3.0.4)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\yoshi\anaconda3\lib\site-packages (from matplotlib) (1.3.2)
Requirement already satisfied: pillow>=6.2.0 in c:\users\yoshi\anaconda3\lib\site-packages (from matplotlib) (9.0.1)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\yoshi\anaconda3\lib\site-packages (from matplotlib) (2.8.2)
Requirement already satisfied: cycler>=0.10 in c:\users\yoshi\anaconda3\lib\site-packages (from matplotlib) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\yoshi\anaconda3\lib\site-packages (from matplotlib) (4.25.0)
Requirement already satisfied: numpy>=1.17 in c:\users\yoshi\anaconda3\lib\site-packages (from matplotlib) (1.21.5)
Requirement already satisfied: six>=1.5 in c:\users\yoshi\anaconda3\lib\site-packages (from python-dateutil>=2.7->matplotlib) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

複利法のシミュレーション

お金を預けるとどうなるのか

借りたお金を放置するとどうなるのか

預金の複利計算(モデル)

- 複利法
 - 元金に利息を加算したものを次の期間の元金として利息計算
- 次期の金額 = 現在の金額 + 利息
- 利息 = 現在の金額 × 利率

例:10万円を年利5%で預けると

- 以下のモデルを使ってシミュレーション
 - 次期の金額 = 現在の金額 + 利息
 - 利息 = 現在の金額 × 利率

• Excelでシミュレーション



• Pythonでシミュレーション



例:10万円を年利5%で預けると

Excelでシミュレーション

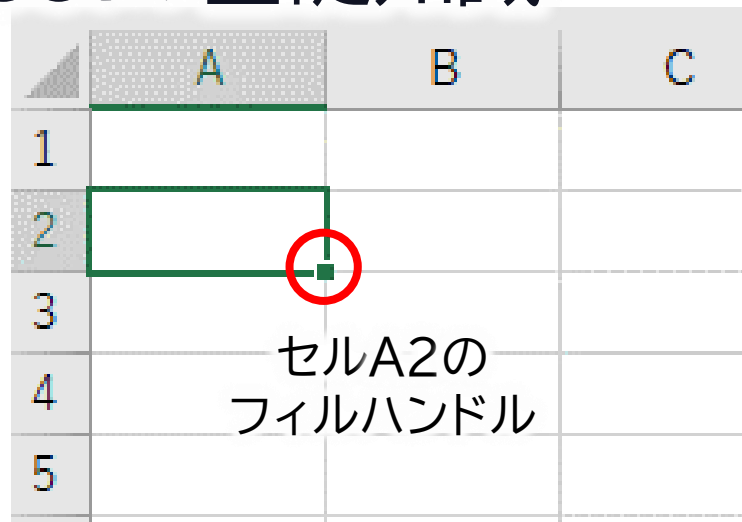


•手順

1. 表の作成
2. 計算式の入力
3. グラフ作成
4. 数値を変えて実行

•手順の説明は次ページへ

Excelの基礎知識



セルA2の
フィルハンドル

おかしいときは直そうとせず
[Esc]→[Ctrl]+[Z]

例:10万円を年利5%で預けると

Excelでシミュレーション



•表の作成

1. セルA1に 利率 と入力
2. セルB2に 金額 と入力
3. セルA3に 預金 と入力
4. セルA4に 1年目 と入力
5. セルA4のフィルハンドルをドラッグし、
20年目まで作る

	A	B	C
1	利率		
2		金額	
3	預金		
4	1年目		
5	2年目		
6	3年目		
7	4年目		
8	5年目		
9	6年目		
10	7年目		
11	8年目		
12	9年目		
13	10年目		
14	11年目		
15	12年目		
16	13年目		
17	14年目		
18	15年目		
19	16年目		
20	17年目		
21	18年目		
22	19年目		
23	20年目		
24			

完成イメージ

例:10万円を年利5%で預けると

Excelでシミュレーション



• 計算式の入力

1. セルB1に 0.05 を入力
2. セルB3に 100000 を入力
3. セルB4に `=B3+INT(B3*B1)` を入力
4. セルB4のフィルハンドルをダブルクリック

入力方法(入力する文字はすべて半角)

1. セルB4に `=` とタイプ
2. セルB3をクリックして `+` をタイプ
3. `int(` とタイプしてからセルB3をクリック
4. `*` をタイプしてからセルB1をクリック
5. `[F4]`キーを押す
6. `)`をタイプしてから`[Enter]`キーを押す



例:10万円を年利5%で預けると

Excelでシミュレーション

- 計算式の入力-計算式の意味
 - 次期の金額 = 現在の金額 + 利息
 - 利息 = 現在の金額 × 利率

$$=B3 + \text{INT}(B3 * \$B\$1)$$

現在の金額

利息

整数化

現在の金額

利率

	A	B	C
1	利率	0.05	
2		金額	
3	預金	100000	
4	1年目	=B3+INT(B3*\$B\$1)	
5	2年目	11 INT(数値)	

例:10万円を年利5%で預けると

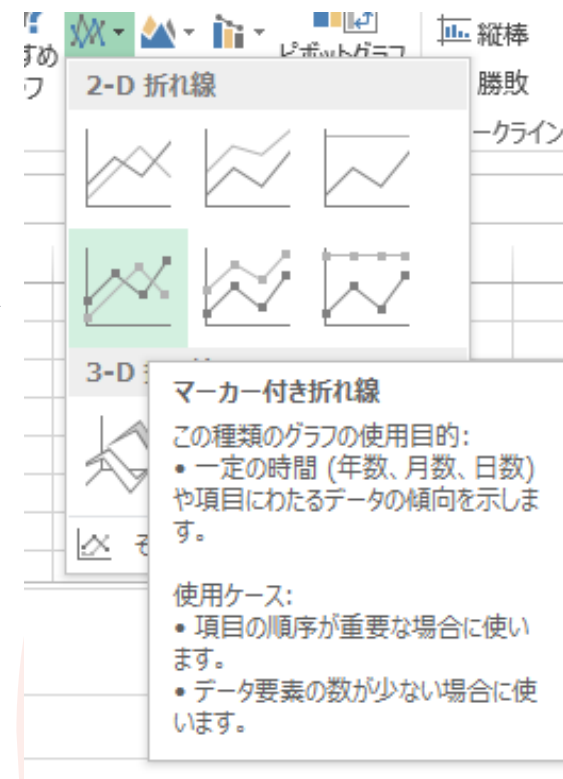
Excelでシミュレーション



• グラフ作成

1. セルA2からセルB23までドラッグ
2. [挿入]→[折れ線グラフ]とクリック
3. [マーカー付折れ線]をクリック

• グラフから150,000円を超えるのは何年後か読み取ってみよう



例:10万円を年利5%で預けると

Excelでシミュレーション



- 数値を変えて実行
 - 現在の実際の利率は0.1%以下です
 - 0.1%の利率で利息部分が1000円を超える(101000になる)のは何年後かシミュレーションする

- 自分のドライブに
fukuri という名前で保存

例:10万円を年利5%で預けると

Pythonでシミュレーション



- 手順

1. プログラムの作成・実行
2. グラフ作成のために改造・実行
3. グラフ表示プログラムの追加・実行
4. 数値を変えて実行

- 手順の説明は次ページへ

例:10万円を年利5%で預けると

Pythonでシミュレーション



- プログラムの作成・実行
 1. 新しいノートブックを用意
 2. 以下のプログラムを入力
 3. 入力できたら実行
 4. 20年目まで表示
するよう改造

```
yokin=100000
riritu=0.05
for i in range(10):
    risoku=int(yokin*riritu)
    yokin=yokin+risoku
    print(i+1, '年目:', yokin)
```



例:10万円を年利5%で預けると

Pythonでシミュレーション

- プログラムの作成・実行-プログラムの意味

```
yokin=100000
riritu=0.05
for i in range(10):
    risoku=int(yokin*riritu)
    yokin=yokin+risoku
    print(i+1,'年目:',yokin)
```

```
#変数 yokin に 100000 を代入
#変数 riritu に 0.05 を代入
#変数 i を0~9に変える
    #変数 risoku に yokin*riritu を整数にして代入
    #変数 yokin に yokin+risoku を代入
    # i+1,'年目:',yokin を表示
```

- 次期の金額 = 現在の金額 + 利息
- 利息 = 現在の金額 × 利率

例:10万円を年利5%で預けると

Pythonでシミュレーション



- グラフ作成のために改造・実行
 - 以下の部分を修正して実行し動作確認

```
import matplotlib.pyplot as plt
yokin=[100000]
riritu=0.05
for i in range(20):
    risoku=int(yokin[i]*riritu)
    yokin.append(yokin[i]+risoku)
    print(i+1, '年目:', yokin[i+1])
```

例:10万円を年利5%で預けると

Pythonでシミュレーション



- グラフ表示プログラムの追加・実行
 - 以下の部分を追加して実行し動作確認

```
plt.title('fukuri')
plt.xlabel('Year')
plt.ylabel('Yokin')
plt.plot(yokin, marker='o')
plt.grid(True)
plt.show()
```

```
import matplotlib.pyplot as plt
yokin=[100000]
riritu=0.05
for i in range(20):
    risoku=int(yokin[i]*riritu)
    yokin.append(yokin[i]+risoku)
    print(i+1, '年目:', yokin[i+1])

plt.title('fukuri')
plt.xlabel('Year')
plt.ylabel('Yokin')
plt.plot(yokin, marker='o')
plt.grid(True)
plt.show()
```

完成イメージ



例:10万円を年利5%で預けると

Pythonでシミュレーション

- 数値を変えて実行
 - 現在の実際の利率は0.1%以下です
 - 0.1%の利率で利息部分が1000円を超える(101000になる)のは何年後かシミュレーションする

生命体の増加 シミュレーション

生命体の増加シミュレーション(モデル)

- ある空間の中の生命体の増加について
 - 個体が多ければ増加する
 - 個体数が多すぎると減少する
- 増加数 = 個体数 × 増加率
- 減少率 = (個体数 / 環境収容数) × 増加率
- 減少数 = 個体数 × 減少率

例：個体数10,増加率0.01,環境収容数1000

- 以下のモデルを使ってシミュレーション
 - 増加数 = 個体数 × 増加率
 - 減少率 = (個体数 / 環境収容数) × 増加率
 - 減少数 = 個体数 × 減少率
- Excelでシミュレーション
- Pythonでシミュレーション



例：個体数10,増加率0.01,環境収容数100

Excelでシミュレーション

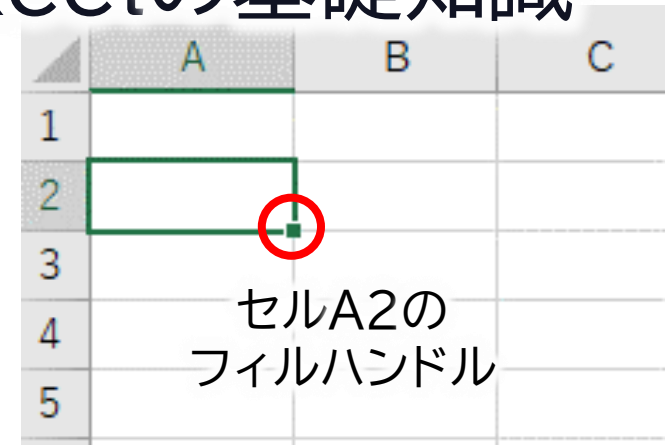


• 手順

1. 表の作成
2. 計算式の入力
3. グラフ作成
4. 効率の良いタイミングを調べる

• 手順の説明は次ページへ

Excelの基礎知識



おかしいときは直そうとせず
[Esc]→[Ctrl]+[Z]

例: 個体数10, 増加率0.01, 環境収容数100

Excelでシミュレーション



• 1. 表の作成

1. セルA1に 環境収容数 と入力
2. セルA2に 増加率 と入力
3. セルA5に 個体数 と入力

	A	B
1	環境収容数	
2	増加率	
3		
4		
5	個体数	
6		
7		
8		
9		

完成イメージ

例: 個体数10, 増加率0.01, 環境収容数1000

Excelでシミュレーション



• 2. 計算式の入力

1. セルB1に 1000 を入力

2. セルB2に 0.01 を入力

3. セルA6に 10 を入力

4. セルA7に $=A6+A6*\$B\$2-A6*(A6/\$B\$1)*\$B\2
を入力

5. セルA7のフィルハンドルを下にドラッグして
セルA1006までドラッグする

例：個体数10,増加率0.01,環境収容数1000

Excelでシミュレーション



- 2. 計算式の入力-計算式の意味
 - 増加数 = 個体数 × 増加率
 - 減少率 = (個体数 / 環境収容数) × 増加率
 - 減少数 = 個体数 × 減少率

$$\bullet = \text{現在の数} + \text{増加数} - \text{減少数}$$

$$\bullet = A6 + A6 * \$B\$2 - A6 * (A6 / \$B\$1) * \$B\$2$$

	A	B	C	D
1	環境収容数	1000		
2	増加率	0.01		
3				
4				
5	個体数			
6	10			
7	=A6+A6*\$B\$2-A6*(A6/\$B\$1)*\$B\$2			
8	10.19897			

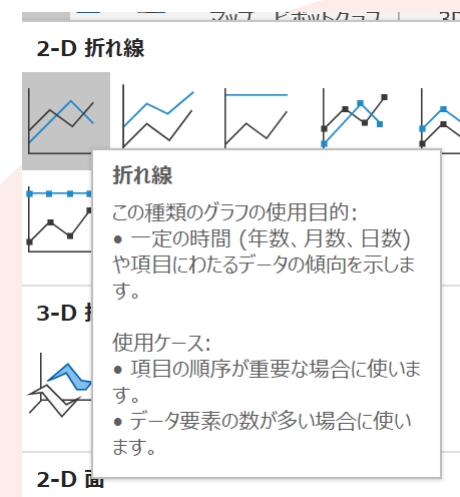
例：個体数10,増加率0.01,環境収容数1000

Excelでシミュレーション



• 3. グラフ作成

1. セルA5からセルA1006までドラッグ
2. [挿入]→[折れ線グラフ]とクリック
3. [折れ線]をクリック



例: 個体数10, 増加率0.01, 環境収容数100

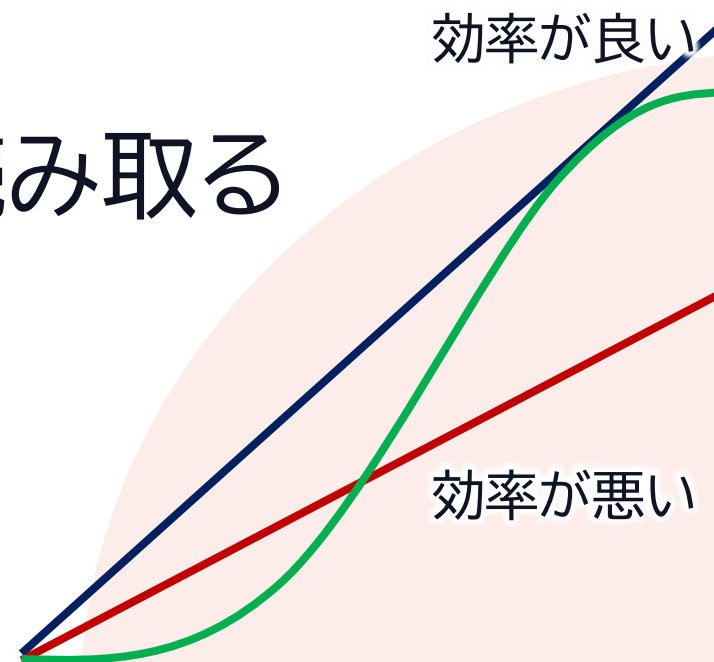
Excelでシミュレーション



- 4. 効率の良いタイミングを調べる
- グラフ上の点と原点を結ぶ直線の傾きが最大
→ 増加の効率が一番高い

- 一番効率が良い時間をグラフから読み取る

- 自分のドライブに
zouka という名前で保存



例: 個体数10, 増加率0.01, 環境収容数100

Pythonでシミュレーション



- 手順

1. プログラムの作成・実行
2. グラフ作成のために改造・実行
3. グラフ表示プログラムの追加・実行
4. 効率の良いタイミングを調べる

- 手順の説明は次ページへ

例: 個体数10, 増加率0.01, 環境収容数1000

Pythonでシミュレーション



- プログラムの作成・実行
 1. 新しいノートブックを用意
 2. 以下のプログラムを入力
 3. 入力できたら実行

```
zouka=0.01
capacity=1000
n=10
for i in range(1000):
    zoukasuu=n*zouka
    gensyousuu=n*(n/capacity)*zouka
    n=(n+(zoukasuu - gensyousuu))
    print(n)
```




例：個体数10,増加率0.01,環境収容数1000

Pythonでシミュレーション

- プログラムの作成・実行-プログラムの意味

```
zouka=0.01
capacity=1000
n=10
for i in range(1000):
    zoukasuu=n*zouka
    gensyousuu=n*(n/capacity)*zouka
    n=(n+(zoukasuu - gensyousuu))
    print(n)
```

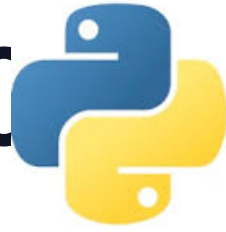
増加率
環境収容数
最初の個体数

増加数
減少数
個体数

- 増加数 = 個体数 × 増加率
- 減少率 = (個体数 / 環境収容数) × 増加率
- 減少数 = 個体数 × 減少率

例: 個体数10, 増加率0.01, 環境収容数1000

Pythonでシミュレーション



- 2. グラフ作成のために改造・実行
 - 以下の部分を修正して実行し動作確認

```
import matplotlib.pyplot as plt
zouka=0.01
capacity=1000
n=[10]
for i in range(1000):
    zoukasuu=n[i]*zouka
    gensyousuu=n[i]*(n[i]/capacity)*zouka
    n.append(n[i]+(zoukasuu - gensyousuu))
    print(n[i+1])
```

例: 個体数10, 増加率0.01, 環境収容数1000

Pythonでシミュレーション



- 3. グラフ表示プログラムの追加・実行
 - 以下の部分を追加して実行し動作確認

```
plt.plot(n)
plt.title("number of life")
plt.xlabel("time")
plt.ylabel("number")
plt.show()
```

```
import matplotlib.pyplot as plt
zouka=0.01
capacity=1000
n=[10]
for i in range(1000):
    zoukasuu=n[i]*zouka
    gensyousuu=n[i]*(n[i]/capacity)*zouka
    n.append(n[i]+(zoukasuu-gensyousuu))
    print(n[i+1])
```

```
plt.plot(n)
plt.title("number of life")
plt.xlabel("time")
plt.ylabel("number")
plt.show()
```

完成イメージ



例: 個体数10, 増加率0.01, 環境収容数100

Pythonでシミュレーション

- 4. 効率の良いタイミングを調べる
- グラフ上の点と原点を結ぶ直線の傾きが最大
→ 増加の効率が一番高い

- 一番効率が良い時間をグラフから読み取る

