④「モデル化とシミュレーション」

私たちの周りで起こる現象は、多くの要因が重なり合って生じており、そのすべてを考慮することは不可能です。そこで、その現象のある一面を抽出し簡略化した形で分析を行う「モデル化」の手法が有効となります。身近な例として「埃が隅に集まる理由」をモデル化し、まず手動シミュレーションでモデルとシミュレーションの考え方を理解しましょう。さらに、効率的かつ大規模にシミュレーションするための手法としてコンピュータシミュレーションを導入し、その有効性を実感しましょう。

ねらい	ランダムウォークを例に、モデル化とシミュレーションの基本的な考え方を理解するとともに、コンピュータシミュレーションの有効性を実感する。
所要時間	３時間
準備するもの	コイン、手動シミュレーション用台紙、パソコン（VBA）

展　開

【手動でシミュレーションを行う】

1　写真を見て埃が隅に多いことに気づかせる。
2　埃が隅に多い理由を班で話し合い、発表する。
3　場所により埃が動く量が違うことを確認し、モデル化する。
4　コインを使って手動シミュレーションを行う。
5　クラス全員の結果を集約し、結果を評価する。

【コンピュータでシミュレーションを行う】

1　手動シミュレーションの利点と欠点を確認する。
2　上記の欠点を補う一手法としてコンピュータシミュレーションを導入する。
3　VBA の基本とプログラムの概要を説明する。
4　１個の埃についてのシミュレーションプログラムを作成し、実行する。
5　100 個の埃についてのシミュレーションプログラムを作成し、実行する。

【振り返り】

1　この授業で、「学んだこと」「気づいたこと」を書く。
2　課題探究のテーマになりそうなことを最低２つは挙げる。

留意点

■モデル化の本質は、現象の背後にある本質的な要素を抽出することにある。何が本質的な要素であるかを見抜くことは困難であることも多いが、まずは現象をしっかりと観察し定性的に把握することが重要となる。
■ここでは、細かなプログラミング上の文法ではなく、モデル化の考え方とそのプログラム上での表現を重視して指導を行う。
■興味のある生徒には、発展的な課題として２次元のランダムウォークへの拡張や Python やＣなど、他言語によるシミュレーションも紹介する。

発　展

【落ち葉の推移モデルとシミュレーション】

地面に落ちている落ち葉を見てみると、多くの場合、裏向きの落ち葉の方が裏向きのものよりも多くなっています。この現象を次のようにモデル化してみます。※1

①落ち葉は風によって一定の割合で反転する

②反転割合は表向きの方が大きい

③新しく落ちてくる落ち葉や出て行く落ち葉はない

※1　ここで紹介している事例は、第 17 回日本物理学会 Jr. セッション 2021 で最優秀賞を受賞した本校生徒の研究『落ち葉に宇宙の神秘を見る』をもとにしています。

これを視覚化したものが、図１の落ち葉の推移モデルです。

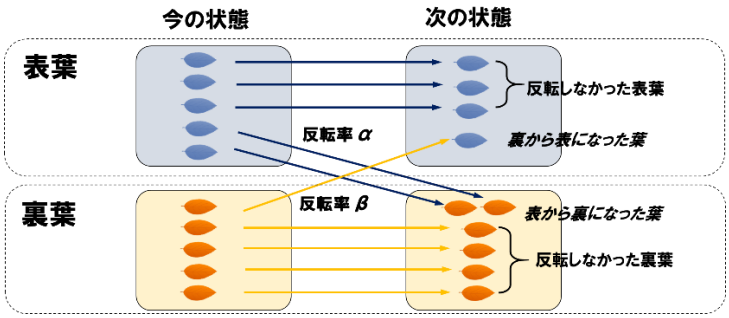

図１　落ち葉の推移モデル

①漸化式によるシミュレーション

今の状態の表葉、裏葉の数をそれぞれ a_n、b_n、次の状態のそれらを a_{n+1}、b_{n+1} とすると、推移モデルは漸化式を用いて、（ｉ）、（ii）式のように表すことができます。

$$\text{表葉}\quad a_{n+1} = (1-\alpha)a_n + \beta b_n \qquad (\text{i})$$

$$\text{裏葉}\quad b_{n+1} = (1-\beta)b_n + \alpha a_n \qquad (\text{ii})$$

最初の状態の表葉、裏葉の数および反転率を設定すれば、表計算ソフトにより表葉、裏葉の数の推移をシミュレーションすることができます。※2

※2　（ｉ）、（ii）式は、$a_n + b_n = N$（一定）なので一般項を求めることができます。

②コンピュータによるシミュレーション

落ち葉の反転は確率的な要因が大きい現象です。確率的要素を取り入れるには、コンピュータシミュレーションが有効です。図２はその中心となる部分のフローチャートです。

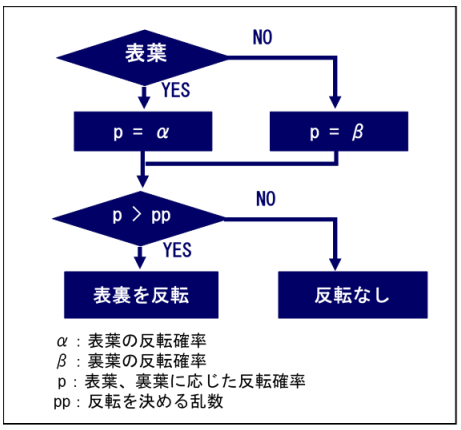

図２　シミュレーションのフローチャート

表１　シミュレーション環境

ＯＳ	Windows10
言語	Python 3.6.9
使用ライブラリ	numpy，matplotlib.pyplot
実行環境	Jupyter Lab 1.0.2

プログラムおよび実行環境は以下のとおりです。

表２ シミュレーションプログラム

```
%matplotlib inline
import numpy as np
from matplotlib import pyplot as plt

alpha = 0.5       # 表葉の反転確率
beta = 0.1        # 裏葉の反転確率

# すべての要素が０の配列の作成
Leaves = np.zeros([121, 100])
# 初期状態（０行目）で偶数列が１（表），奇数列が０（裏）

for j in range(100):
    if j % 2 == 0:
        Leaves[0,j] = 1

# 落ち葉の反転操作
for i in range(120):
    for j in range(100):
        if Leaves[i, j] == 1:
            p = alpha
        else:
            p = beta
        pp = np.random.rand()
        if pp < p:
            Leaves[i+1, j] = 1 - Leaves[i, j]
        else:
            Leaves[i+1, j] = Leaves[i, j]

# 表葉・裏葉の割合
a = np.count_nonzero(Leaves == 1, axis=1)
b = np.count_nonzero(Leaves == 0, axis=1)

# グラフ表示
plt.figure(figsize=(8,5))
plt.grid(color='k', linestyle='--', linewidth=1)
x = np.arange(121)
plt.plot(x, a, lw=3)
plt.plot(x, b, lw=3)
plt.xlabel('time [step]', fontsize=16)
plt.ylabel('percentage [%]', fontsize=16)
plt.xticks(np.arange(0,121,20), fontsize=14)
plt.xlim(0, 120)
plt.yticks(np.arange(0,101,20), fontsize=14)
plt.ylim(0, 100)

# グラフを保存
plt.savefig('Leaves_sim.png')
```

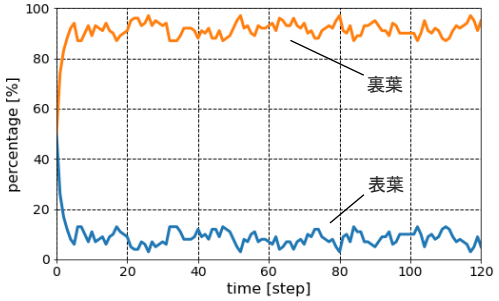

図３ コンピュータシミュレーション結果の一例

1　自然現象の認識　（1）認識の方法

④「モデル化とシミュレーション１」	令和　　年　　月　　日（　　）
【トピック】：ほこりの集まり具合を説明するモデルを考える・検証する	
◎写真を観て気づくこと･･･思いつくことを挙げてみよう！！ 質より量です。	
◎探究の過程を理解する	

モデルをつくってシミュレーションをする

◎廊下のほこりの動きについて

【ほこりの動く原因】

【ほこりの移動の度合い】

【現象のモデル化とシミュレーション】

（１）モデル（右か左かのみ）

①右か左かのみに移動し、どちらに動くかはそれぞれ 1/2 の確率

②廊下中央ほど１回の移動距離が大きい

→ 廊下中央は５、その両脇は３、両端は１の割合で移動

（２）シミュレーション

①スタートは中央

②コインを投げ、表が出たら右、裏が出たら左へ移動

１回の移動距離は、中央が５、その両脇が３、両端が１（壁に当たったらそのまま）

③連続 50 回行い、50 回実行したのちのほこりの位置を座標で記録

50 回後のほこりの位置 ＝ □

④クラス全員のデータ（50 回後の位置）を集約

【振り返り】：何を学び・どのようなことに気づいたか・考えたかを書きましょう

１年　　組　　番　氏名＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿

1　自然現象の認識　（1）認識の方法

④「モデル化とシミュレーション２」	令和　　年　　月　　日（　　）

【トピック】：モデルをもとにコンピュータシミュレーションをしよう

◎プログラムの骨格

①乱数を使って、右に行くか、左に行くかを決める
②ほこりの位置によって、１回に動く距離を変える（中央は大きく、端は小さい）

◎シミュレーションの実行と結果

①ほこりの数を３０個にして、シミュレーションを行う。

②４９回後、各位置にいたほこりの数を集計する。

49回後の位置	−12	−11	−10	−9	−8	−7	−6	−5	−4	−3	−2	−1	0
ほこりの数													
49回後の位置	1	2	3	4	5	6	7	8	9	10	11	12	
ほこりの数													

【振り返り】：何を学び・どのようなことに気づいたか・考えたかを書きましょう

１年　　組　　番　氏名＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿