

## 身近な事象におけるシミュレーションについての指導

### —「モデル化とシミュレーション」実践報告—

東京都立新宿山吹高等学校教諭 綿貫 俊之

#### 1. 学校紹介

新宿山吹高等学校<sup>1)</sup>は、定時制・通信制課程を併設し、必修科目を除けば生徒が科目と時間割を自由に選択できる単位制無学年制の高等学校である。現在定時制課程に設置している専門学科「情報科」では、学習指導要領に記載してある専門教科11科目に加え、学校設定科目を含み計15科目を設置している(表1)。

表1 情報科開設科目

分類名		科目名
共通科目	基礎科目	情報産業と社会 情報と表現 情報実習α(基礎の実習)
	応用科目	モデル化とシミュレーション データ解析
システム系科目	基礎科目	プログラム言語①<C言語> プログラム言語①<VB> データベースシステム
	応用科目	プログラム言語②<C言語> プログラム言語②<VB> 情報システムの開発 ネットワークシステム
コンテンツ系科目	基礎科目	コンピュータデザイン マルチメディア表現
	応用科目	図形と画像の処理 情報コンテンツ
総合科目		課題研究①(必修) 課題研究②(選択)

その中で、以下には共通応用科目で現在実施している「モデル化とシミュレーション」について記すことにする。

#### 2. モデル化とシミュレーションの実施について

「モデル化とシミュレーション」は、普通教科「情報B」の一部単元、また専門教科「情報」では、システム設計・管理分野とマルチメディア分

野の共通分野として、応用選択的科目での1科目として扱われている。

内容として、現象やルールに基づいたモデル化の考え方や方法、実際の問題解決におけるシミュレーションの扱いについて学ぶことで、科学的な理解として有効な内容でもある。

現在専門学科「情報科」にて本科目を実施している全国の学校数は7校あり、多くは3学年での配置である。

#### 3. 確率・統計(サイコロの出た目について)

##### 3.1 サイコロを自ら投げる

「投げ方によりバラツキが大きい」という印象を強めるために、サイコロを何度も振って各目の出た回数を記録するシミュレーションを行うものである。これは、人によって投げ方をどのように行っているかということを検証するものである。投げる手の使い方、腕の使い方などを周囲にて観察しながら行った。ゲームによっては、各自が都合の良いように、たとえば「『1』が出てほしい、『2』が出てほしい…」と願う、出るような投げ方をしているなど、いわゆる「作為」的な投げ方なのか、あるいは正常な投げ方なのかと判断を考えていった。

またこの問題としては、1人10回のサイコロを振り、出た目を調べ10回分の度数と確率を調べていった。さらに、グループにてその各々10回分の合計した結果を書かせた。

確率という点では、度数ごとで増えていく「累積確率」についての求め方の演習も行った。

### 3.2 コンピュータを使用した内容として

#### (度数分布と確率分布)

表計算ソフトウェアには、乱数を発生させる機能を備えているものがある。また乱数の中で基本的なものでは、どのような数値も等しい確率として出てくる一様乱数がある。

これは、不規則な現象のシミュレーションには不可欠であり、乱数を発生させる方法と、それを利用した問題解決の手法を学ぶことを目的とした内容である。

乱数とは、ルーレットのように不規則な並びの数のことをいうが、人間がサイコロを振るなどすると、投げ方によっては、ある結果に偏りが生じる可能性もある。

これより、ここでは単なる計算問題を解くというのではなく、乱数を用いたコンピュータによるサイコロを作成した。それにより、回数限定の結果だけではなく、100回、1000回と行った場合に、どのような結果が生じるのかを検証することができる。

リスト1 サイコロの出た目を出力させるソース

```
<html>
<body>
<script>
y=6*Math.random();
document.write("サイコロの目は");
if (y<1) {
    document.write("1");
} else if (y<2) {
    document.write("2");
}
else if (y<3) {
    document.write("3");
}
else if (y<4) {
    document.write("4");
}
else if (y<5) {
    document.write("5");
}
else {
    document.write("6");
}
document.write("です。");
</script>
</body>
</html>
```

### (1) 乱数を発生させるプログラミング

乱数を発生させるプログラミング「JavaScript」による出力方法である。本授業は、特にプログラミング言語について問うものではないが、予備知識について確認を行った後、リスト1の通り入力を行った。出た目(図1)の結果を調べるものとしては有効である。

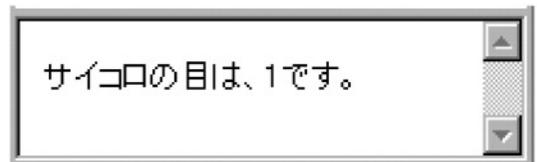


図1 サイコロの出た目を出力した結果の一例

「Math.random();」というように記述があるのは、この数には0以上1未満の数値が代入されることを意味し、値は代入されるたびに異なる数値が出力されることとなる。

また、ソースの中の「else if(y<2)」とある部分で変数を変えることにより、出た目を均等ではなく、作動的に出力させることも可能である。

### (2) 電子サイコロを表計算にて作成

2つ目は、電子サイコロを表計算ソフトウェアにて作成するものである。サイコロの出た目の作成は、一様乱数「0~0.999…」を発生させ、これに6倍すると、「0~5.999…」となる。一様乱数の値は、6未満であることから、さらに、1を加えて「1~6」を作成する。表計算ソフトウェアでの関数上では、「=INT(一様乱数\*6)+1」(図2)として出力させる。関数「INT」は、整数化する関数である。

回数	一様乱数	サイコロの目
1	=RAND( )	=INT(6*左出力セル)+1
2	=RAND( )	=INT(6*左出力セル)+1
⋮	⋮	⋮

図2 ワークシートへの数式の入力

サイコロの出た目の確率は、回数を増やせば増やすほど1/6(≒0.1666...)に近づくことについては、理論的にも数学的にも期待値や独立試

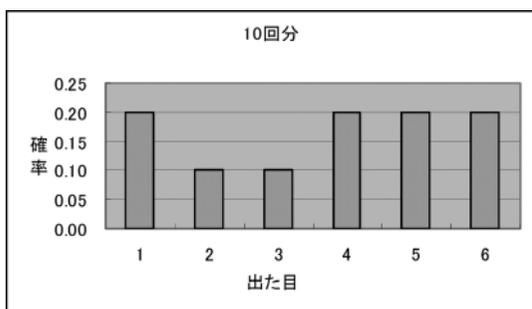


図3 サイコロを10回分振った結果

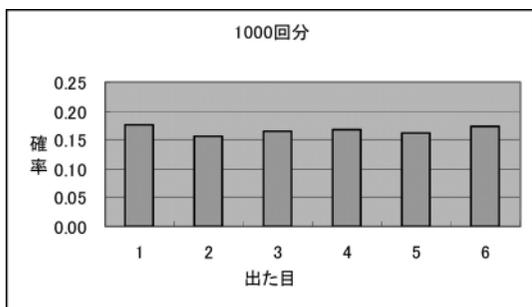


図4 サイコロを1000回分振った結果

行だけでは簡単に説明できない。

これを10回振った場合(図3)だけでなく、1000回(図4)と回数を多くした場合でのシミュレーションを試行する。

ここでは、どうしてシミュレーションを行わなければならないのか、という意義自体も考えさせることにより、コンピュータの存在意義が理解できる。

## 4. 待ち行列

### 4.1 待ち行列が生じる場面について考える

サイコロでの乱数の作成と確率の計算から、これを利用した問題解決の方法として、待ち行列がある。

これらについても、サイコロを自ら振るのと同様、実際にどういう場面で、人の並びが生じるのか考えた上で行った。スーパーマーケットや販売窓口などでサービスを受けるまでに順番待ちが生じる。その待ち時間はどれくらいになるかについて目的を持たせる。

### 4.2 コンピュータを使用した内容として(待ち行列での時刻と時間の関係のシミュレーション)

コンピュータでの実践としては、到着時間やサービス時間を一定にしたり、ランダムにしたり条件を変えてシミュレーションを行うことにより、待ち行列のできるしくみについて理解を深める。

待ち行列がない場合は、客の到着時刻がそのままサービスの開始時刻となるが、待ち行列がある場合は、前の客のサービス終了時刻が次の客のサービス開始時刻となる。

確率を求めた時と同様に、ワークシート上に、過去のデータに基づいた待ち行列のモデルを作成する。次に、作成したモデルから乱数を用いたシミュレーションを行い、どのような待ち行列になるかを分析する。シミュレーションでは、横棒グラフにて、待ちが生じている部分を色で塗りつぶすものとする(図5)。一方、同時に待ち行列の長さを示すグラフでは、時間帯によってどのくらいの待ち人数が生じているのかを表す(図6)。

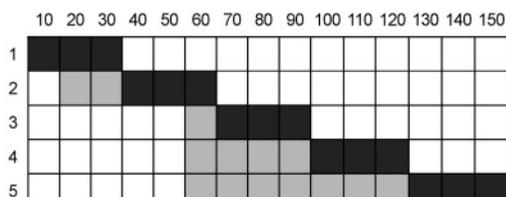


図5 待ち行列のシミュレーション(縦が人目、横軸が秒)を表すグラフ

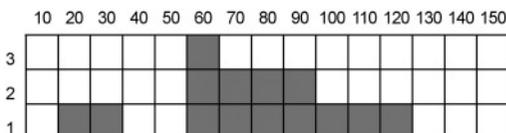


図6 待ち行列の長さ(縦が人目、横軸が秒)を表すグラフ

## 5. モデリングツールによるシミュレーション(システムダイナミクス(SD))

システムダイナミクス(以下、SD)とは、1956年にアメリカのマサチューセッツ工科大学のフォレスター氏により創案された、企業の動きをモデル化し複雑なしくみを表すためのソフトウェアであり、アメリカの高等学校でもSDを使

った授業も行われているという。

しかし、SD本体は高価なため、とても現場ですぐに使用できるソフトウェアとは言い難い。そのため本校の実践では、フリーソフトウェア（Ventana systems社製「Vensim<sup>[2]</sup>」）にて実施した。

モデル化を表すには、図的モデルが適されるが、これと同時に計算までもが一度に行えることが利点である。以下、駐車場におけるモデル化においての一例を示す。

ある駐車場の入出庫において、入庫と出庫の割合を図的モデルと時間間隔とで示した関係をモデル化で表す。

はじめに、現象「駐車場」に対する初期設定（開始時間、終了時間、時間間隔など）を設定し、対象となるフローモデルを作成する。ここでは、「駐車場」自体の現象から、これに対する入力（インフロー）と出力（アウトフロー）を考える（図7）。この時点で入力と出力に対する値を入力後、シミュレーションを実行する。すると、データがセットされ、グラフ（図8）と時間と個数との関係を表す数値（図9）が出力される。

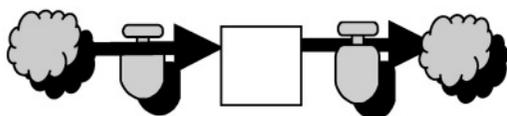


図7 駐車場モデルのストックとフロー（例）

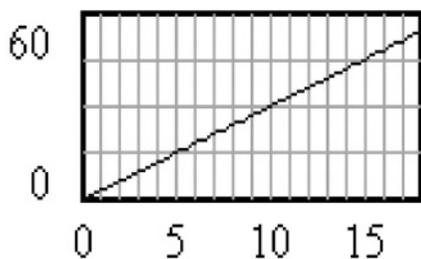


図8 シミュレーション後のグラフ化（例）

時間（時間）	駐車場
0	0
0.5	1.5
1	3
1.5	4.5

図9 シミュレーション後の数値関係（例）

## 6. おわりに

表計算における式の組み立て方、また文章を読み取って図的なモデルを描くことなど、本科目ではある程度の基礎知識を持って望まないといけない問題もある。

学習指導要領では、普通教科、専門教科双方とも同じ意味合いで「簡単にモデル化できる題材を扱い、数理的、技術的な内容に深入りしない」「理論的に深入りしない」とも記載されている。しかし、実際には、アプリケーションソフトウェアの操作やそれに伴う数学的知識が求められている問題（数式モデル）にて表現し、実際のシミュレーションを行っている問題が多い。上記にて、「深入りしない」とはしても、それ以前に学習済みの知識や、予備知識を生かすことが求められている。そのことで、実際の生徒への指導としてどのように考えなくてはならないのか、悩みどころは多い。

「数学的だから難しい」などと敬遠するケースもあるが、生徒自身が今後生活する上で想定される場面でのモデル化、シミュレーションは新学習指導要領でも一部の実施とはいえ、重要な分野であることに違いはない。

また、本実践では「情報=コンピュータ」という概念を持つ生徒に対し、「コンピュータを使わない」、つまり実物で実験（実証）を行うことが大切であるという必要性を前提に行ったことについても加えておく。

### 参考URL

- [1] <http://www.yamabuki-hs.metro.tokyo.jp>
- [2] <http://www.vensim.com/software.html>