

表計算ソフトウェアを用いた教育用シミュレーションの開発

井 寄 幸 平

(受付 2017 年 10 月 31 日)

1. はじめに

シミュレーションとは現実問題から必要な構造と情報を抽出し、モデルを作成して行う模擬実験のことである。機械や建築物の設計、物理現象・社会現象の予測や因果関係の究明など、様々な分野で幅広く利用されている。教育面でもシミュレーションは有用であり、対象とする問題の構成要素やその相互関係を理解・整理してモデル構築を行うという点でシステム科学の、実際に作成したモデルの計算処理や結果表示の手順を論理的に考え、コンピュータ上に実装するという点で情報科学の教育手法として成果が期待できる。

しかしながら実際にシミュレーションを構築・実施するには数学やプログラミングの知識を要することが多く、文系のしかも初学者にはハードルが高く感じられる。プログラミングを視覚的にかつ容易に行えるような専用ソフトウェアも存在するが、導入コストや操作に習熟する手間、応用性などを考えると最適とは言い難い。

そこで本稿では、事実上の業界標準である表計算ソフトウェア Excel を用いて教育用シミュレーションの開発を行う。Excel は多くの高等学校や大学で情報教育の基礎として学ばれており、プログラミング言語や他の専用ソフトウェアと比較すれば文系初学者も取り組みやすいと考えられる。また、Excel が持つ豊富なグラフ機能や計算セルの色づけ機能などは、シミュレーション構築の上で特に専門的な知識を要求されるグラフィカルな結果表示を容易にしてくれる。本来 Excel は非常に高機能なソフトウェアであり、使いこなすにはそれなりの困難が伴うが、本稿では文系初学者を対象とし、マクロや VBA コードなどの複雑な機能を用いずに「作りやすく、結果がグラフや動きで視覚的に確認できる」シミュレーションの構築を目指す。

2. 物理シミュレーション

物体の挙動や変化を取り扱う物理シミュレーションは、現実中存在するものを対象とする

ことが多いため学生にとっても比較的身近でイメージしやすく、また使用する理論や計算式が明確であるためにシミュレーションモデル構築の練習・解説にも適している。ここではごく単純な斜方投射と、複雑な挙動を示す2重振り子のシミュレーションについて述べる。

2.1 斜方投射

ボール等の物体を斜めに投げ上げる斜方投射は、物理計算の初歩であるとともに「時間とともに物体が移動する」ことを視覚的に理解しやすい例題である。シミュレーションモデル構築への導入と興味を喚起することを狙い、斜方投射の軌道や到達点の数値的計算だけでなく「実際にボールが飛んでいるような」シミュレーションの構築をおこなう。

まず初速を v_0 、地面となす角度を θ 、重力加速度を g とすると、投げてから t 秒後の物体の x 、 y 座標は以下のように表される。ここでは物体を質点とみなし、空気抵抗を考えないものとしている。

$$\begin{cases} x = v_0 \cos \theta \cdot t \\ y = v_0 \sin \theta \cdot t - \frac{1}{2} g t^2 \end{cases}$$

この時の物体の位置を時刻に応じて計算するため、微小な時間間隔を設定し、その都度物体の座標を計算してグラフ上に表示させる。具体的には以下のような数式を入力し、Excelの再計算機能を用いて時刻を変化させ、逐次座標を計算して表示する。なお初速、地面との角度、時間間隔は所与として数値を入力する。

	A	B	C	D	E	F
1	初速	角度	時間間隔			
2	10	45	0.05			
3						
4	時刻	x座標	y座標			
5	↑	↑	↑			
6						
7			=A2*SIN(RADIANS(B2))*A5-0.5*9.8*A5*A5			
8						
9		=A2*COS(RADIANS(B2))*A5				
10						
11	=IF(C5<0.0,A5+C2)					
12						

図1. 斜方投射シミュレーションの入力式

時刻の計算の際、自セルの計算に自セルの値を用いるため「循環参照」の警告が表示され

るがいったん無視し、数式の入力完了後に「ファイルメニュー」の「オプション」から「数式」タブを選択し、計算方法を「手動」として「反復計算を行う」にチェックを入れ、最大反復回数を「1」に設定することで繰り返し再計算が可能になる。

こうして得られた x, y 座標を「散布図」を用いてグラフ化することでその時刻における物体の位置がグラフ上に表示される。F9キーによる再計算で時刻が変化するので、キーを押し続ければ物体が飛んでいるように移動して表示される。また、物体の質量と空気抵抗係数を導入すれば、空気の抵抗を受ける場合のシミュレーションを同時に実施して比較することも可能である。

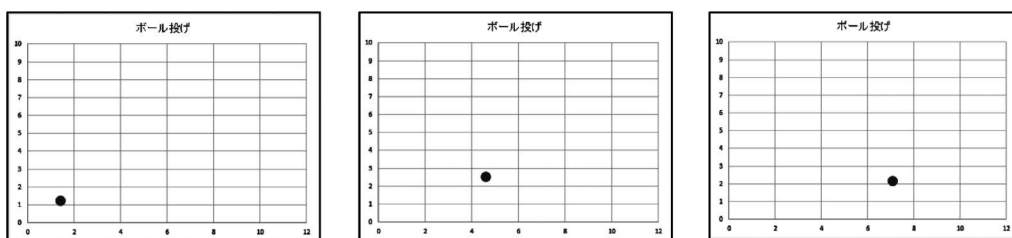


図 2. 斜方投射シミュレーションの挙動

2.2 2重振り子

複数の重りを剛体の腕で接続した多重振り子は、単純な構造ながら非常に複雑な挙動を示すことが知られている。また、その挙動は2階微分方程式を含む運動方程式で記述され、オイラー法やルンゲ・クッタ法等による近似計算の練習にも適している。ここでは2重振り子を対象とし、オイラー法による近似計算を用いて実際に振り子が揺れる様子を観察できるシミュレーションを構築する。

2重振り子の支点を座標 $(0, 0)$ とし、支点到近い方から重りの質量を M_1, M_2 、腕の長さを L_1, L_2 、鉛直方向となす角を θ_1, θ_2 とし、重りの座標を $(x_1, y_1), (x_2, y_2)$ とすると以下のように記述できる。

$$\begin{cases} x_1 = L_1 \sin \theta_1 \\ y_1 = -L_1 \cos \theta_1 \end{cases} \quad \begin{cases} x_2 = L_1 \sin \theta_1 + L_2 \sin \theta_2 \\ y_2 = -L_1 \cos \theta_1 - L_2 \cos \theta_2 \end{cases}$$

これらの式からラグランジュ法によって角度の2階微分を求めると以下ようになる。

$$\begin{cases} \theta_1'' = \frac{A_2 D_1 - B D_2}{A_1 A_2 - B^2} \\ \theta_2'' = \frac{A_1 D_2 - B D_1}{A_1 A_2 - B^2} \end{cases}$$

ここで $A_1 \sim D_2$ は右の通りである。

$$\begin{cases} A_1 = (M_1 + M_2)L_1^2 \\ A_2 = M_2L_2^2 \\ B = M_2L_1L_2 \cos(\theta_1 - \theta_2) \\ D_1 = -M_2L_1L_2\theta_2' \sin(\theta_1 - \theta_2) - (M_1 + M_2)gL_1 \sin \theta_1 \\ D_2 = M_2L_1L_2\theta_1' \sin(\theta_1 - \theta_2) - M_2gL_2 \sin \theta_2 \end{cases}$$

この式をオイラー法で近似すると以下のように2つの重りの角度が得られ、そこから座標が計算できる。

$$\begin{cases} \theta_1'(t + \Delta t) = \theta_1'(t) + \frac{A_2D_1 - BD_2}{A_1A_2 - B^2} \Delta t \\ \theta_2'(t + \Delta t) = \theta_2'(t) + \frac{A_1D_2 - BD_1}{A_1A_2 - B^2} \Delta t \end{cases}$$

$$\begin{cases} \theta_1(t + \Delta t) = \theta_1(t) + \theta_1'(t) \Delta t \\ \theta_2(t + \Delta t) = \theta_2(t) + \theta_2'(t) \Delta t \end{cases}$$

以上のパラメータと数式を以下のように Excel に入力する。

	A	B	C	D	E	F	G
1	M1	M2	L1	L2	$\theta_1(0)$	$\theta_2(0)$	Δt
2	1.2	0.1	1.4	1.2	2.5	2.8	0.05
3	$t + \Delta t$	t					
4	①	②					
5							
6	A1	A2	B	D1	D2		
7	③	④	⑤	⑥	⑦		
8	θ_1'	θ_2'					
9	⑧	⑨					
10	θ_1	θ_2					
11	⑩	⑪					
12							
13		x	y				
14	支点	0	0				
15	1	⑫	⑬				
16	2	⑭	⑮				
17							

① =IF(A4>=30,G2,A4+G2) ② =A4-G2 ③ =(A2+B2)*C2*C2
 ④ =B2*D2*D2 ⑤ =B2*C2*D2*COS(A11-B11)
 ⑥ =-1*B2*C2*D2*B9*SIN(A11-B11)-(A2+B2)*9.8*C2*SIN(A11)
 ⑦ =B2*C2*D2*A9*SIN(A11-B11)-B2*9.8*D2*SIN(B11)
 ⑧ =IF(B4=0,A9+G2*(B7*D7-C7*E7)/(A7*B7-C7*C7))
 ⑨ =IF(B4=0,B9+G2*(A7*E7-C7*D7)/(A7*B7-C7*C7))
 ⑩ =IF(B4=0,E2,A11+A9*G2) ⑪ =IF(B4=0,F2,B11+B9*G2)
 ⑫ =C2*SIN(A11) ⑬ =-1*C2*COS(A11) ⑭ =B15+D2*SIN(B11)
 ⑮ =C15-D2*COS(B11)

図 3. 2重振り子シミュレーションの入力式

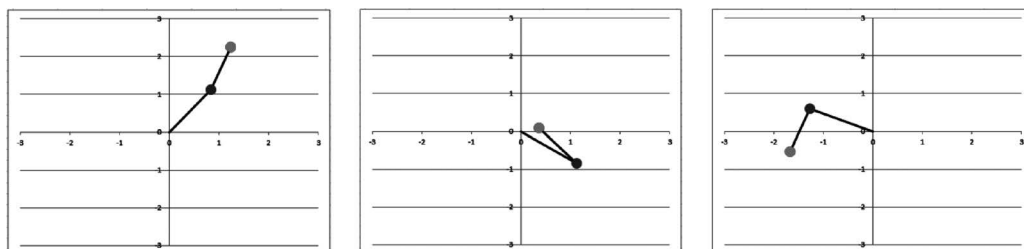


図4. 2重振り子シミュレーションの挙動

ここでも循環参照が発生するため、いったん警告を無視して数式の入力後に反復計算を許可する。その後に重りの座標を選択して「散布図」の「マーカーと直線」を作成すると2重振り子の状態をグラフ上に表示する。斜方投射と同じくF9キーによる再計算をおこなうと、振り子が複雑に揺れ動く様子を観察できる。

3. 経済系シミュレーション

経済科学部の学生にとって、身近な経済問題をモデル化してシミュレーションし、その原理や挙動を学ぶことは重要な意義がある。ここでは仕入れと売上の関係を取り扱う新聞売り子問題と、サービス窓口が発生する待ち行列モデルのシミュレーション構築について述べる。

3.1 新聞売り子問題

新聞売り子問題は変動する客数に対して適切な仕入れ量を決定する問題である。品切れを起こせば得られた利益を逃すことになるが、在庫を翌日に持ち越すことができないため売れ残りもまた損失となる。ここでは以下のようにこの問題をモデル化する。

- * 1部 c 円で仕入れた新聞を a 円で売る。
- * 客は1日平均 m 人やってくるが、毎日の客数 x は変動する。
- * 売れ残りが発生すると仕入れ分の損失となり、品切れを起こすとその分の利益を逃す。
- * 新聞売り子が得る利益を最大にする1日の最適発注部数を求める。

このとき、1日の客数を x 、仕入れ部数を y とすると利益は以下のように表される。

$$\pi = \begin{cases} xa - yc & (x < y \text{ のとき}) \\ y(a - c) & (x \geq y \text{ のとき}) \end{cases}$$

客数 x はポアソン分布に従うと考えることが多いが、Excelの計算式でポアソン分布に従う乱数を発生させるには複雑な手順を踏む必要があるため、平均 m 、標準偏差 \sqrt{m} の正規分

布で近似する。仕入れ値を80，売り値を120，客数の平均を10とし，1～20部仕入れた際の利益を計算するために以下のように Excel に入力する。客数を乱数で発生させるため，繰り返し計算を行って利益の平均値を計算する。なお，図中の①，②は縦，③は横・縦，④，⑥，⑦は横へドラッグしてコピーする。ここでも繰り返し計算のために循環参照が発生するため，数式の入力後に反復計算を許可しておく。

	A	B	C	D	E	F	V	W	X
1	日数	乱数	客数	1	2		19	20	
2	1	①	②	③					
3	2	↓	↓	↓					
4	3	↓	↓	↓					
5	4	↓	↓	↓					
6	5	↓	↓	↓					
7	6	↓	↓	↓					
8	7					...			
9	8								
10	9								
11	10								
12				④					
13									
14	繰り返し回数	⑤	利益合計	⑥					
15			利益平均	⑦					
16									

① =NORMINV(RAND(), 10, SQRT(10)) ② =MAX(INT(B2), 0)
 ③ =IF(\$C2<D\$1, \$C2*120-D\$1*80, D\$1*(120-80))
 ④ =SUM(D2:D11) ⑤ =B14+1 ⑥ =D14+D12 ⑦ =D14/\$B14

図 5. 新聞売り子問題シミュレーションの入力式

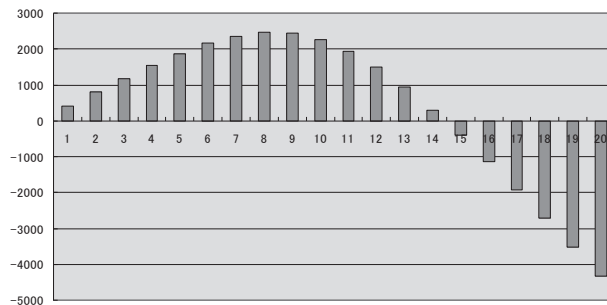


図 6. 新聞売り子問題シミュレーションの結果例

次に「利益平均」の部分を選択し，縦棒グラフを作成することで利益の値を視覚的に確認できる。F9キーによる繰り返し計算を行うと利益の値が変化し，平均してどの程度の利益が得られるかが確認できる。結果として一定数以上の仕入れが大きな損失につながることや，利益幅の大小によって仕入れ数を客数の平均より大きくするべきか小さくするべきかなど，様々な状況をシミュレーションすることができる。

3.2 待ち行列

待ち行列モデルは、例えばスーパーのレジのようにサービスを受ける窓口がある場合に、客が順番に並んでサービスの開始を待つようなモデルである。他にもネットワークに接続されたプリンタに複数の印刷要求が送信された場合など様々な状況を表現でき、システム科学の重要な分析対象となる。ここでは身近でイメージしやすいスーパーのレジを想定し、窓口が1つだけの場合に絞ってシミュレーションを作成する。特に、待ち行列の状態を表すグラフを Excel のセルそのものをグラフ用紙のように扱って作成することで、モデルの内容を理解しながら視覚的に確認できるシミュレーションが構築できる。

ここでは乱数を用いて客の到着間隔とサービスにかかる時間を決定し、そこからサービスの開始時刻と終了時刻を計算する。これらが分かると客の状態（到着前、待ち状態、サービス中、サービス終了）が決定されるため、Excel のセルに状態を表示させてグラフにする。また最大待ち時間、平均待ち時間、最大待ち人数を計算する部分を作成して待ち行列の状態を数値的にも確認する。Excel への入力は以下ようになる。

	A	B	C	D	E	F	G	H	I	J
1	人数	乱数1	到着間隔	乱数2	サービス時間		到着時刻	サービス開始時刻	サービス終了時刻	
2	1	①	0	③	④		0	0	⑤	
3	2	①	②				⑥	⑦	⑧	
4	3									
5	4									
6	5									
7										

① = RAND() ② =IF(B3<0.1,1,IF(B3<0.3,2,IF(B3<0.7,3,IF(B3<0.9,4,5)))
 ③ = RAND() ④ =IF(D2<0.1,1,IF(D2<0.2,2,IF(D2<0.5,3,IF(D2<0.9,4,5)))
 ⑤ =E2 ⑥ =G2+C3 ⑦ =MAX(I2, G3) ⑧ =H3+E3

図 7. 待ち行列シミュレーションの入力式（計算部分）

K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AN	AO	AP			
1																																待ち時間	最大待ち時間	
2																																⑩	⑪	
3																																	平均待ち時間	
4																																	⑬	
5																																	最大待ち人数	
																																	⑭	
5																																		
4																																		
3																																		
2																																		
1																																		
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25								

⑨ =IF(M\$7<\$G2,"",IF(M\$7<\$H2,"待",IF(M\$7<\$I2,"サ","")))
 ⑩ =IF(\$L9<=COUNTIF(M\$2:M\$6,"待"),\$L9,"")
 ⑪ =COUNTIF(M2:AL2,"待") ⑫ =MAX(AN2:AN6)
 ⑬ =AVERAGE(AN2:AN6) ⑭ =MAX(M9:AL13)G2+C3

図 8. 待ち行列シミュレーションの入力式（グラフ部分）

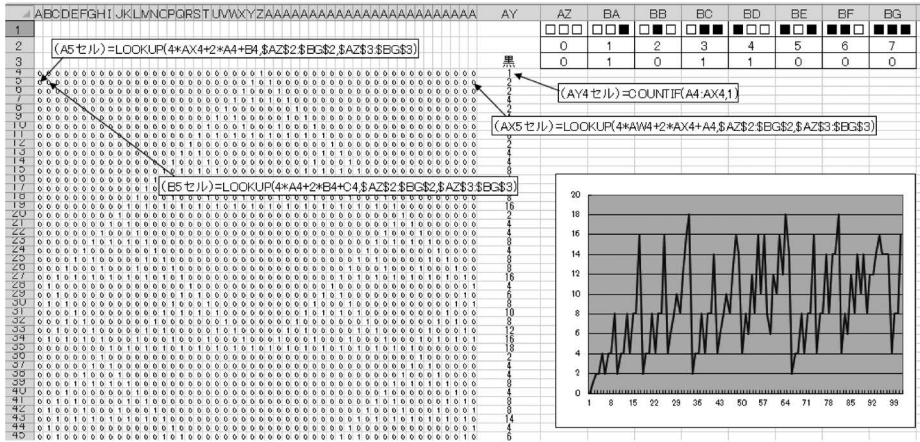


図10. 1次元セルオートマトンの入力式

ここで1～3行目の右側に記述されているものがセルの推移規則であり、1が黒、0が白を表す。実際に色づけするにはExcelの条件付き書式を利用する。推移規則は自由に変更することができ、2状態3近傍のセルオートマトンであれば256通りの推移規則を試することができる。初期値は4行目に数値として自由に入力でき、同じ推移規則であっても初期値によって挙動が変化することが確かめられる。また同時刻における黒いセルの数をカウントし、グラフ化することで全体が安定状態にあるのか、周期的な状態にあるのか、変動を続ける複雑な状態にあるのか等が把握できる。

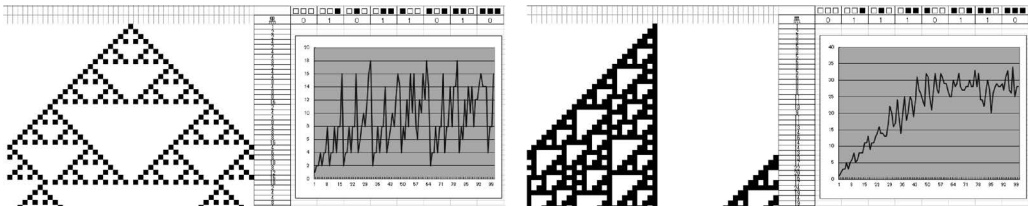


図11. 1次元セルオートマトンの実行例

4.2 ライフゲーム

1次元セルオートマトンを平面上に拡張した2次元セルオートマトンも作成することができる。Conwayが考案したライフゲームは2次元セルオートマトンの代表例であり、非常に複雑で興味深い挙動を示すことが知られている。ライフゲームではセルオートマトンの各セルを生物に見立て、「周囲の状況が過密、もしくは過疎で死亡」「周囲に適度な生存セルがあれば生存」という規則を適用する。具体的には(1)死んでいるセルの周囲に3つの生存セルがあれば次の時刻に誕生し、(2)生存セルの周囲に2つもしくは3つの生存セルがあれば

次の時刻も生き残る。これ以外の場合は全て次の時刻に死亡する。

これを Excel 上で再現するには、「初期状態」「現在のセル状態」「次の時刻のセル状態」の 3 つを分けて作成し、現在のセル状態から次の時刻のセル状態を計算するとともに、その結果を再度現在のセル状態へ反映させるとよい。

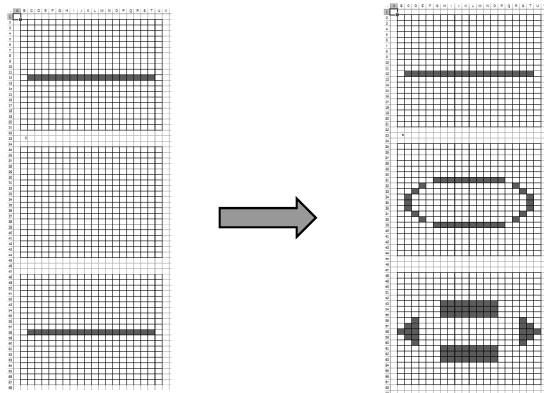


図12. ライフゲームの実行例

一番上の枠内にはすべて「0」を入力し、後で初期状態となるように好きな位置に「1」を入力する

(B23セル) =IF(B23<100,B23+1,0)

(B25セル) =B48

この数式を横、縦にドラッグし、枠内にコピーする

(B48セル) =IF(\$B\$23=0,B2,IF(SUM(A24:C26)=3,1,IF(AND(SUM(A24:C26)=4,B25=1),1,0)))

図13. ライフゲームの入力式

なお上記の計算式では時刻の計算に循環参照が発生するため、入力後に反復計算を許可しておく。また、現在の状態を示す中央部分の枠内を選択し、条件付き書式を用いて色付けをおこなう。初期状態の配置によって、画面上に広がっていくものや繰り返し複雑なパターンを生成するものなど、様々な変化を確認することができる。

5. 強化学習

強化学習は機械学習の一手法であり、試行錯誤を繰り返しながらより良い方策を発見、学習するものである。学習を行う「主体」と「環境」があり、主体は周囲の状態を観測し、行動を選択する。状態と行動はセットとして評価され、行動選択の結果として環境から「報酬」または「罰」が与えられる。この過程を繰り返すことでより環境に適した行動が強化され、適切な行動を学習していくものである。

ここでは応用例として分かりやすい「迷路学習」を作成する。格子状の平面内にエージェントを配置し、迷路となっている空間内をゴールに向かって移動する。エージェントは自分の前後左右を観測して状態を把握でき、行動として前後左右のいずれかに移動できる。壁にぶつかった場合は罰、ゴールに到達した場合は報酬を与えられるものとし、迷路内を壁にぶつかることなく、かつ最短でゴールできる経路を学習する。

強化学習には様々なアルゴリズムが提案されているが、ここでは代表的なアルゴリズムである Q-Learning を使用する。Q-Learning は状態と行動のセットに Q 値と呼ばれる評価値を与え、その値を逐次修正しながら行動を獲得していく手法である。Q 値の更新式は以下のようになる。

$$Q(s_t, a_t) \leftarrow (1 - \alpha)Q(s_t, a_t) + \alpha \left[r_t + \gamma \max_a Q(s_{t+1}, a) \right]$$

ここで s_t , a_t はそれぞれ状態と行動を表す。 r_t は報酬の値（罰の場合は負の値）、 α , γ はそれぞれ学習率、割引率と呼ばれるパラメータである。これらを Excel 上に構築するには、まず迷路となる部分の枠を作成した上で、それぞれの位置（状態）に対応する Q 値を計算する必要がある。行動が上下左右の 4 方向に可能であれば、1 か所につき 4 つの Q 値が計算できるよう、それぞれ枠を作成しておく。また、ライフゲームの時と同様に「現在のエージェントの位置」と「次の時刻のエージェントの位置」を別途計算し、次の時刻の位置を現在の状態に反映させる。

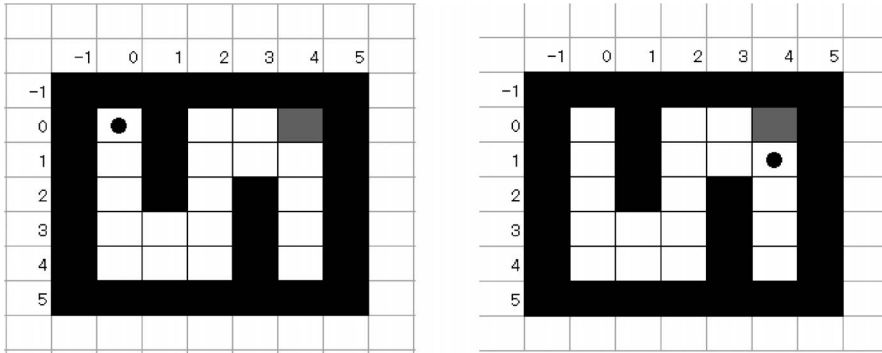


図14. 強化学習による迷路学習例

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC
1																													
2			-1	-1	-1	-1	-1	-1																					
3			-1	0	0	0	0	1	-1																				
4			-1	0	0	0	0	0	-1																				
5			-1	0	0	0	0	0	-1																				
6			-1	0	0	0	0	0	-1																				
7			-1	0	0	0	0	0	-1																				
8			-1	-1	-1	-1	-1	-1	-1																				
9																													
10			0	1	2	3	4			0	1	2	3	4			0	1	2	3	4			0	1	2	3	4	
11		①							②							③							④						
12		1							1						1								1						
13		2							2						2								2						
14		3							3						3								3						
15		4							4						4								4						
16																													
17	①②	③	④	⑤	⑥																								
18	プログラム数	⑦																											

① (C11 セル) =IF(AND(\$B\$20=0,\$B\$21=0),1, IF(AND(\$B\$23=C\$10,\$B\$24=\$B11,\$B\$22=1), 0.5*C11+0.5*(\$B\$29+0.9*\$B\$31),C11))

② (J11 セル) =IF(AND(\$B\$20=0,\$B\$21=0),1, IF(AND(\$B\$23=J\$10,\$B\$24=\$B11,\$B\$22=2), 0.5*J11+0.5*(\$B\$29+0.9*\$B\$31),J11))

③ (Q11 セル) =IF(AND(\$B\$20=0,\$B\$21=0),1, IF(AND(\$B\$23=Q\$10,\$B\$24=\$B11,\$B\$22=3), 0.5*Q11+0.5*(\$B\$29+0.9*\$B\$31),Q11))

④ (X11 セル) =IF(AND(\$B\$20=0,\$B\$21=0),1, IF(AND(\$B\$23=X\$10,\$B\$24=\$B11,\$B\$22=4), 0.5*X11+0.5*(\$B\$29+0.9*\$B\$31),X11))

※①～④はそれぞれの枠内にドラッグしてコピーする

図15. 強化学習の入力式 (1)

ここで2～8行目に作成した枠は迷路の状態とその地点での報酬を表したもので、-1の部分壁、1の部分ゴール、0の部分通路となる。入力値を変化させれば迷路の組み換えやゴール地点の変更が容易に可能である。

	A	B	C	D	E	F	G	H	I	J
19										
20	t	⑩								
21	T	⑪								
22	行動	⑫								
23	x(t)	⑬								
24	y1(t)	⑭								
25	dx	⑮								
26	dy	⑯								
27	x(t+1)	⑰								
28	y(t+1)	⑱								
29	報酬	⑲								
30	Q(t+1)	⑳	㉑	㉒	㉓					
31	Qmax	㉔								
32										
33			-1	0	1	2	3	4	5	
34		-1	⑳							
35		0								
36		1								
37		2								
38		3								
39		4								
40		5								
..										

⑳ =INDIRECT(ADDRESS(B28+11,B27+3))

㉑ =INDIRECT(ADDRESS(B28+11,B27+10))

㉒ =INDIRECT(ADDRESS(B28+11,B27+17))

㉓ =INDIRECT(ADDRESS(B28+11,B27+24))

㉔ =MAX(B30:E30)

㉕ =IF(AND(C\$33=\$B\$27,\$B34=\$B\$28),"●",B2)

※㉕は枠内にドラッグしてコピー

⑩ =IF(B29=1,0,IF(B20<100,B20+1,0))

⑪ =IF(B20=0,B21+1,B21)

⑫ =IF(B18=MAX(B18:E18),1,IF(C18=MAX(B18:E18),2,IF(D18=MAX(B18:E18),3,4)))

⑬ =IF(B20=0,0,B27)

⑭ =IF(B20=0,4,B28)

⑮ =IF(B22=2,1,IF(B22=4,-1,0))

⑯ =IF(B22=1,-1,IF(B22=3,1,0))

⑰ =IF(INDIRECT(ADDRESS(B24+B26+3,B23+B25+3))=-1,B23,B23+B25)

⑱ =IF(INDIRECT(ADDRESS(B24+B26+3,B23+B25+3))=-1,B24,B24+B26)

⑲ =INDIRECT(ADDRESS(B24+B26+3,B23+B25+3))

図16. 強化学習の入力式 (2)

上記の入力式によりエージェントの位置と行動に対応したQ値が逐次計算できる。迷路部分は条件付き書式により色付けを行い、壁部分、ゴール部分を塗っておくとよい。また移動回数のカウントに循環参照を用いているため、計算式の入力後に反復計算を許可しておく。

F9キーによる再計算を行うとエージェントが移動を開始し、様々な試行錯誤を繰り返しながら最終的にゴールへ至る経路を学習する様子が観察できる。この際どの行動が選択され、それに応じてQ値がどう変化したかを同じシート内で全て確認できるため、学習内容の確認や理解にも非常に有用である。

6. おわりに

本稿ではシステム科学、情報科学等の教育に生かすため、表計算ソフトウェア Excel を用いて様々なシミュレーションを開発した。特にマクロや VBA といった複雑な機能を用いず、基本的にセルへの計算式の入力と反復再計算、グラフ機能と条件付き書式によるセルの色付けを利用することで、文系初学者でも取り組みやすいものを目指した。プログラミング言語

のように配列等は使えないため、計算要素が増えるとどうしても多くのセルが必要になること、複数の条件による分岐がある場合は一つの計算式が長くなりがちであることなど、問題点もあるが、視覚的に変化を観察できる面白さに加えて何よりも「自分で作って動かすことができる」点が学生の学習意欲を刺激し、教育効果を高めていると考えている。

今後はより簡潔に作成できるシミュレーションのほか、処理の手順を学生が自ら論理的に考え、自分なりに実装できるような課題や、より高度なモデルやアプリケーションへとステップアップできるような方策を考えていきたい。

参 考 文 献

- [1] Poundstone, W., 有澤 誠 (訳), 「ライフゲームの宇宙」, 日本評論社, 2003.
- [2] Wolfram, S.: *Cellular Automata and Complexity*, Addison-Wesley Publishing Company, 1994.
- [3] 伊藤俊秀, 草薙信照, 「コンピュータシミュレーション」, オーム社, 2006.
- [4] 小高知宏, 「Cによる数値計算とシミュレーション」, オーム社, 2009.
- [5] 小和田正, 沢木勝茂, 加藤 豊, 「OR 入門 意思決定の基礎」, 実教出版, 1984.
- [6] 正司和彦, 高橋参吉, 「最新 モデル化とシミュレーション」, 実教出版, 2006.
- [7] 三上貞芳, 皆川雅章, 「強化学習」, 森北出版, 2000.
- [8] 三井和男, 「Excel コンピュータシミュレーション」, 森北出版, 2007.