

# Excel によるプログラミング教育

— 授業科目「社会ネットワーク分析」における試み —

Programming Education with Excel:  
An Attempt in the Lecture “Social Network Analysis”

牧 野 真 也  
Makino, Shinya

## ABSTRACT

Programming language education is encountering difficulties in many universities. To overcome this problem, the author proposes the lecture “social network analysis” in which the spreadsheet software Excel is used as a tool of programming education. In this paper, we introduce an attempt in this lecture. We expect that Excel is superior to programming language in comprehensibility and synergy effects among various lectures with Excel. As a result of the lecture, we found that Excel was more effective in improving lecture contents, submitted subjects by students and examination results than programming language.

## 1. はじめに

筆者は、本学着任以来、経済学部において情報処理教育を行ってきた。そのうち、プログラミング言語を用いた教育については、旧産業工学科や旧社会システム設計学科における授業科目「プログラミング言語Ⅱ」や旧エキスパートコース向けの「情報基礎演習Ⅲ」、また最近では、2007年度からの「経済プログラミング」などを中心に、主にC言語を用いたプログラミング教育を断続的に行ってきた。

しかし、近年においては、プログラミング言語教育における受講生の反応は決してよいものではなく、授業内容もその理解も低いレベルにとどまっている

状況が続いている。近年開講している授業科目「経済プログラミング」では、簡単な社会シミュレーションを最終的な目的に掲げているものの、その内容のほとんどはプログラミング言語のきわめて基本的な範囲を中心とせざるを得ない状況となっている。また同科目の授業中の課題や期末試験では、基本的な問題を中心に出題しているが、その結果も予想を下回るものとなっている（試験の素点に授業中の課題点を加算して最終的な成績としている）。

一方、他の多くの大学においても、近年では同様の状況にあるようであり、<sup>(1)</sup>大学におけるプログラミング教育、とりわけ文科系におけるプログラミング教育においては、それなりの見直しが迫られているといえる。

確かに、プログラミング言語は、初学者にとってはいくつかの全く新しい概念を含み、一見難しそうな文法をもち、その記述はきわめて厳格で面倒であるといえる。これを短時間で習得し応用するためのモチベーションを文科学部での授業において維持することは困難かもしれない。

しかし、筆者は、経済学部におけるプログラミング教育が不要であるとは全く考えていない。今日の社会科学においては、コンピュータを利用したシミュレーションや分析が研究方法として主要なものの一つになりつつある。また、文科学学生の就職後の実務においても、データ分析や定型的な事務処理の自動化など、本格的ではないかもしれないがそれなりの（広い意味での）プログラミングが求められる場面は多いであろう。そもそも、プログラミングの際に求められる体系的な思考やモデル化の能力、さらには目的に向けて粘り強く努力し達成する能力は、一般的能力として、また昨今その必要性が強調される「就業力」としても不可欠であると考えられる。すなわち、経済学部におけるプログラミング教育の必要性は、むしろ高まりつつあると認識している。

こうした状況において、筆者は、2011年度前期に新規の授業科目「社会

---

(1) たとえば、荻谷（2007）によれば、東京大学の前期課程における情報教育では、「学生の評判はすこぶる悪く……授業内容の改訂を余儀なくされている」とのことであり、とくにプログラミング教育は「受講生の少なさに悩まされている」とのことである。

ネットワーク分析」を開講した。同科目では、これまでのプログラミング言語を中心に据えたプログラミング教育を見直して、表計算ソフト Excel を中心に、場合によってはそれに付属するプログラミング言語 VBA (Visual Basic for Applications) による実装をプログラミングと見なし、社会科学分野に求められるプログラミング教育を Excel の表計算としての機能を最大限に活用して行なうことを目指している。本稿ではその試みを紹介する。

## 2. Excel によるプログラミング教育とは

### (1) プログラミングと Excel

そもそも、プログラミングとは一体何であろうか。さまざまに説明できるが、その本質は、現実に係わる何らかの状況をコンピュータで実施可能にする（実装する）ことであろう。もちろん、すでに現実で行なわれている情報処理が対象になることもあるし、場合によっては、まだ行なわれていない状況や起こっていない現象が対象になることもあろう。

かつては、とりわけ 1990 年代中頃のパソコン普及以前は、こうしたプログラミングのためにプログラミング言語を利用する以外の選択肢がほとんどなかったといえる。したがって、プログラミング教育においてはプログラミング言語を用いざるを得なかった。本学部においても、1990 年代中頃までは UNIX ワークステーションや X 端末を教育用に多数導入し、C 言語によるプログラミング教育を行なっていた。<sup>(2)</sup>

一方、今日では、広い意味でのプログラミングのためのツール（ソフトウェア）は、プログラミング言語に限らずさまざまなものが利用可能である。もちろん、表計算ソフト Excel を用いて問題をモデル化・定式化し実装することも、広い意味ではプログラミングであり、その過程でプログラミングの重要な概念

---

(2) 当時の本学部の産業工学科や社会システム設計学科では、1 年生全員を対象に C 言語教育を行なっていた。また、たとえば文書作成においても TeX の教育が行なわれるなど、あらゆる場面において記述言語を用いる方法が主流であった。

やスキルを習得できるというてよいであろう。たとえば、Excel を用いて所得税の確定申告のワークシートを作成することは、複雑なものでなければ容易に作成できるが、このプロセスはプログラミングと呼んでよいであろう。そして、その過程で入出力、データや処理といったプログラミングの中核となるさまざまな概念を習得することとなる。

とりわけ、今日の Excel はきわめて高機能であり、数百に及ぶ関数、さまざまな分析ツールや可視化機能、データベース機能などを備えている。さらに Excel には VBA (Visual Basic for Applications) という Excel とともに動く Excel における処理を自動化することを主な目的としたプログラミング言語が付属している。なお VBA は BASIC の流れをくむ計算完備なプログラミング言語であり、VBA を用いれば Excel でできることの自動化だけでなく、Excel がない機能も追加できる。すなわち、Excel + VBA で、パソコンでできることのほとんどが実装可能というてよい。

## (2) Excel によるプログラミング教育の形態

以上のような Excel によるプログラミング教育はさまざまに可能であるが、筆者は大きく以下の3つがあると考えている。

まず第1に、VBA を用いずに Excel が提供する機能のみで問題を定式化し実装するものである。もちろん、これだけでも、相当なプログラミング教育が可能である。筆者が2005年度から担当している授業科目「経済情報処理」では、Excel の機能のみを使って、AHP による意思決定（最終的にはユーザインタフェース用のワークシートを作成し、特定の問題の意思決定支援システムとして稼働するようにしている）、株価のテクニカル分析、線形計画問題（Excel のソルバーを用いて変数の数が多い問題についても定式化し解を求める）などを扱っている。後述するように授業科目「社会ネットワーク分析」においても、1次元セルオートマトンを Excel の本来の機能のみで実装している。

第2に、VBA を使って Excel のみでは困難な（不可能な）処理を実装する



教育もできよう。ここでは、とくに操作の繰り返しなどの自動化や、Excel に用意されていない処理を実装することが考えられる。具体的には、授業科目「社会ネットワーク分析」におけるライフゲームのような2次元セルオートマトンの繰り返し処理の自動化と制御や、ネットワークの頂点間の距離を求めるウォーシャル・フロイド法のような、表構造に基づいて繰り返し計算する処理の実装などがあげられる。

そして第3に、VBAによる、アルゴリズムの学習を中心とするプログラミング教育も有効と考えられる。これは、従来のプログラミング教育をVBAによって行なうものといえるが、VBAとExcelワークシート間のデータのやりとりが容易なので、入出力のプログラムの作成が容易(ほぼ不要)となる。また、Excelの通常の機能を用いて、プログラムの結果を検算することなども容易にできる。もちろん、固有价值計算などExcelにない関数の実装などもその内容として有効であろう。さらに、一般プログラミング言語であるVisual Basicのプログラミングに発展されることができれば、Webアプリケーションをはじめとするあらゆる実用レベルのプログラムが開発可能となる。

本稿で紹介する授業科目「社会ネットワーク」では、Excelで可能な処理を中心にVBAで必要な処理を補う第2の形態でのプログラミング教育を目指している。また、将来的には第3の一般プログラミング教育の授業科目の開講も視野に入れていくべきであろう。

### (3) Excelによるプログラミング教育の利点

では、表計算ソフトExcelによるプログラミング教育にはどのような利点があるか。一般プログラミング言語との比較でいくつか考えてみたい。

まず第1に、表計算ソフト自体の使いやすさや理解のしやすさをあげることができる。たとえば、表計算ソフトでは、人間が直感的に理解しやすい表形式のデータを直接操作する。すなわち、表の構成要素であるセルに直接データを入力し、セルに計算式を記入し、その結果が即時に計算され表示される。これ

は、プログラミング言語がプログラムコード中の具体的な値がすぐにはわからない抽象的な変数を厳格な記述によって（文法に従って）操作しなければならないことに比べて格段に理解しやすい。さらに実行のためにコンパイルやロード（あるいは起動）などの別の操作も必要としない。

第2に、とくに文科系大学における情報処理教育全体としての連続性をあげられよう。本学部をはじめとして多くの大学では初期の情報処理教育として表計算ソフト Excel を採りあげている<sup>(3)</sup>。とくに本学部では、1年生前期に全学生が授業科目「情報基礎演習」を受講するようになっていて、そこでは全授業15回中の7回分、すなわち約半分が Excel の習得に充てられている<sup>(4)</sup>。したがって、多くの学生の Excel の習熟度はそれなりのレベルに達していると考えられる。これに対して、一般のプログラミング言語教育では、Excel との連続性はほとんどなく、場合によっては Excel の理解がむしろ有害な場面も指摘されている<sup>(5)</sup>。また、Excel を使用している情報処理以外の経済学・経営学関係の専門科目もいくつかあり、それらの科目との相乗効果も期待できよう。

第3に、上記の連続性と関係するが、仮にプログラミング言語として Excel VBA を用いた場合は、それなりのプログラミング言語特有の概念や文法を理解する必要はあるが、開発環境の習得や入出力のためのプログラミングといった、プログラミング教育（とりわけアルゴリズムの教育）において本質的でない部分はかなり省略可能であることがあげられる。一般にプログラミング言語を用いる場合、そのための開発環境であるツール類の使い方を習得する必要がある。また、入出力のためにそれなりのプログラミングが必要となったり、場

---

(3) 多くの大学のシラバス（多くがインターネットで公開されている）によって確認できる。

また、高等学校の教科「情報」でも場合によって（科目の選択によっては）は採りあげているようである。

(4) 2011年度「情報基礎演習」シラバスによる。

(5) たとえば、プログラミング言語において変数の数式（代入式）を宣言（記述）した場合、該当する変数の値が変更されると、それに関係する他の変数の値が（Excel のように制約指向的に）自動的に再計算されるという誤解が非常に多いと報告されている。たとえば、深町（2010）。

合によっては別のソフトウェアによる実装（たとえば、WWW ブラウザを用いた入出力を HTML で記述するなど）が必要になったりすることもある。一方、Excel VBA を用いれば、入出力は、たとえばワークシート上のセルと VBA 変数の間の代入演算のみによって行なうことができ、また VBE (Visual Basic Editor) という Excel に統合された開発環境を用いることができる。

最後に、Excel は実務において他に比肩するものがないほどよく用いられているソフトウェアであり、Excel を習得する直接的な（経済的な面はもちろん、就職活動などにおいても）価値は高く、学生の学習意欲も高いと考えられる。VBA も Excel における定型処理を自動化するために、一般のオフィスにおいて広く用いられている技術である。これは、一般のプログラミング言語が特別な業務に従事しない限りほとんど用いる機会がないことと対照的である。

### 3. 授業科目「社会ネットワーク分析」

授業科目「社会ネットワーク分析」では、近年注目されている社会ネットワーク分析を最終的な目的としている。社会ネットワーク分析は人間間や人間の集団間の関係を分析しその特徴を研究するものであり、将来的には統計分析と並んで社会科学における主要な分析手法になるものと目されている。

また、授業では、社会ネットワーク分析に先立ってセルオートマトンを扱っている。セルオートマトンは古くからある離散的な計算モデルであり、自然科学分野から社会科学分野の問題まで広く応用されている。また、セルオートマトンで扱うセルが規則的に配置された格子と呼ばれる構造は、社会ネットワーク分析で扱う一般的なネットワークの特殊な形態とみることができ、さらにその格子構造が Excel の表ときわめて親和性が高いため、社会ネットワークの導入部としてセルオートマトンは適切であると考えている。

図 1 に授業のホームページの概形を示す。

以下、セルオートマトンと社会ネットワーク分析のそれぞれの内容について紹介する。



あなたは マキノシンヤ としてログインしています。: 学生 (通常のロールに戻る)

開講科目 ▶ 社会ネットワーク分析

通常のロールに戻る

活動

- フォーラム
- リソース
- 課題

フォーラムの検索

検索オプション

管理

- プロフィール

マイコース

- 実験用コース
- 専門演習 I (担当: 牧野)
- 専門演習 II および卒業論文 (担当: 牧野)
- 社会ネットワーク分析 (担当: 牧野)
- 経済情報処理 (担当: 牧野)

トピックアウトライン

社会ネットワーク分析 (2011年度)

担当教員	対象学年	開講期間/単位数	開講時期	履修室
牧野真也	2・3・4年	前期 2単位	金曜日5限	システム情報学センター・第3演習室

ガイダンス  
お知らせ

以下のフォーラムに、授業に関する質問や意見を自由に投稿してください。

授業に関する質問や意見

メールでの質問 shinya-m@eco.wakayama-u.ac.jp  
授業中の質問 (先生を呼ぶ) ボタン

最新ニュース

04月 22日 12:28  
マキノシンヤ  
Moodleははじめました 詳細...  
過去のピックアップ...

直近イベント

直近のイベントはありません。

カレンダーへ移動する...  
新しいイベント...

最近の活動

2011年 07月 29日 (金曜日) 18:15  
以来の活動  
最近の活動詳細...

最終ログインより更新されたものはありません。

カレンダー

◀ 2011年 07月 ▶

日	月	火	水	木	金	土
					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	26	27	28	29	30
31						

イベントキー

- グローバル
- コース
- グループ
- ユーザ

1

I. 表計算ソフト再入門

期間: 2011/4/15 ~ 5/6

簡単な問題を解くことにより、Excelの復習をします。(テキストは授業科目「経済情報処理」と同じものを用いますが、課題は異なります。)

課題の提出方法は、提出直前の授業で説明します。課題は提出期限(終了日時)までは何度でも出し直せます。(終了日時後も提出できますが、「遅く提出」とマークされます。その場合は評点をそれなりに低くします。)

説明資料  
課題1  
補足資料(説明資料のExcelファイル)

2

II. 社会シミュレーション

期間: 2011/5/13 ~ 6/17

社会を対象にした簡単なシミュレーションについて、特にセルオートマトンをとりあげます。

説明資料  
課題2  
補足説明  
n進数(参考資料)  
ライフゲームのバターン集

3

III. 社会ネットワーク分析

期間: 2011/7/1 ~ 7/22

社会をネットワークとして分析する社会ネットワーク分析について基本的な事項をとりあげます。Excelを拡張したツールも用います。

説明資料  
課題3  
課題3解答例(暫定版)

4

試験

2011/7/29 16時30分 ~ 18時00分

注意

正当な理由なしに、この試験を受けなかった場合は履修放棄とします。

試験は必ずシステム情報学センター第3演習室で受験してください。許可なく別の場所で受験した場合は、履修放棄とします。また、学生証を準備した上で、必ず自分のユーザ名でログインしてください。

試験中は私語をしてはいけません。また、席を立ったり移動したりしてはいけません。プリントアウトもしてはいけません。

持ち込み参照やホームページの閲覧などは自由ですが、他の受験生や外部と情報のやりとりをすることはできません。実際に会話をすることはもちろん、ネットを通じてのメールやチャット、サイトへの投稿などもしてはいけません。これらが発覚した場合は、履修放棄とします。また、酷似した答案があった場合は、当事者にメール等で確認の上適切に対応します。

時間内にできる限り問題を解きましょう。合格点は受験者全体の出来具合に基づいて決定します。終了日時までは何度でも再提出できますが、終了時間を越えた場合提出を受け付けなくなるので、余裕を持って提出するようにしてください。

図 1 授業科目「社会ネットワーク分析」のホームページ (2011 年度前期)

### 3.1 セルオートマトン

セルオートマトンは、規則的に並んだ多数のセルの集合で、それぞれのセルがいくつかの（2つ以上の）内部状態をもち、それらセル同士が部分的に相互作用して次（世代）の内部状態を決定していくというダイナミックなモデルである。<sup>(6)</sup>あるセルが相互作用する範囲はそのセルの近くに存在するセル（近傍）に限定される。このような局所的な相互作用によって出現する（創発する）マクロな特徴（挙動）に基づいた分析がなされる。

セルオートマトンはそのセルが配列される次元によって分類される。授業では直線状の1次元セルオートマトンと碁盤状の2次元セルオートマトンを扱っている。

また、セルオートマトンをはじめとするシミュレーションの実装においては、とりわけ繰り返し制御が重要である。なぜなら、多くの場合そこでは同じ計算（操作）を繰り返すことによって、その世代や状態が進展していくからである。

以下、1次元と2次元のセルオートマトンを授業において Excel でどのように扱っているか簡単に説明する。

#### (1) 1次元セルオートマトン

1次元セルオートマトンでは、Excel のワークシートの1行（列数有限）で1世代分のオートマトンを表現している。内部状態は「生き」と「死に」の2つのみをもち、それぞれを1と0で表現する。また、近傍として両隣の1つずつの2つセルのみを考える。そして、そのオートマトンの次世代をその下の行に作成し、これを繰り返すことによって創発されるマクロなパターンをみる(図2)。

---

(6) 授業では、牧野 (2011a) をテキストに用いている。詳細については同論文を参照のこと。

	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	AD	AE	AF	AG	AH	AI		
1	世代			セルオートマトン																																	
2	1	1	1	0	1	1	0	0	1	0	0	0	0	0	0	0	1	0	1	1	0	0	1	1	0	0	0	1	0	1	1	0	0	1	0	1	
3	2	1	0	1	0	1	1	1	1	0	1	0	1	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	1	1	0	1	0	1	
4	3	0	0	0	0	1	0	0	1	0	0	0	0	0	1	0	0	1	1	0	0	0	0	0	0	1	1	0	0	1	1	1	0	0	0		
5	4	0	0	1	0	1	1	0	1	0	0	1	0	0	1	0	1	1	1	1	0	0	0	0	1	1	1	1	0	1	0	1	1	0	0		
6	5	1	0	1	0	0	1	1	0	0	1	1	0	0	1	0	0	0	0	1	1	0	0	0	1	1	0	0	0	1	0	1	1	1	0		
7	6	1	0	0	1	1	1	1	1	1	1	1	1	1	0	1	0	1	1	1	1	1	1	1	1	0	0	1	0	0	1	0	1	0	0		
8	7	0	1	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	1	1	1	0	1	1	0	0	0	1		
9	8	1	1	0	1	1	0	0	0	0	0	0	1	0	0	0	1	0	1	0	0	0	0	0	1	1	0	1	1	1	0	1	1	0	1		
10	9	1	1	0	1	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	1	1	0	0	0	1	0	1	0	1	0		
11	10	1	1	0	1	0	1	1	0	0	1	0	1	0	0	0	0	0	1	0	1	1	1	0	1	1	0	1	0	1	0	0	0	0	1		
12	11	1	1	0	0	0	1	1	1	1	0	0	0	1	0	0	0	0	1	0	0	1	0	1	0	1	1	0	0	1	0	0	1	1	0		
13	12	0	1	1	0	1	1	0	0	1	1	0	0	1	0	1	0	1	0	1	1	1	0	0	0	0	1	1	1	1	0	1	1	1	0		
14	13	0	1	1	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	1	1	1	0	0	1	1	0	0	1	0	1	0	1	0	1		
15	14	0	1	1	1	0	1	0	0	0	0	1	1	0	0	0	0	0	1	1	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0		
16	15	1	1	1	0	0	1	0	0	0	1	1	1	1	0	0	0	1	1	1	0	1	0	0	0	0	0	0	1	1	0	0	0	0	0	1	
17	16	1	0	1	1	1	1	0	1	1	1	0	0	1	1	0	1	1	0	1	0	0	1	0	0	0	1	1	1	1	0	0	1	0	1	0	
18	17	1	0	1	0	1	0	1	0	1	1	1	1	1	0	1	1	0	0	1	1	0	1	0	1	0	1	1	0	0	1	1	1	1	1	0	
19	18	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		

図2 1次元セルオートマトン（リング状）のExcel ワークシート

図中、C列とAI列はそれぞれ同じ行のAH列とD列を参照するセルである。これは、1次元セルオートマトンをリング状に、すなわち左端の1つ左が右端、右端の1つ右が左端になるようにし、セルの位置による特殊性をなくするためである。また、1世代目の値は乱数を発生させる関数 `randbetween(0,1)` で与え、2世代目以降は1つ前の世代（1行上）を参照し、その世代の生き死にを決定するルールをIF関数などで記述する。セルオートマトンではルールは浸透的なので（すべてのセルに同じものが適用されるので）、このルールを相対参照で記述すれば、それをコピーする分のセルオートマトンの世代ができあがることになる。

つまり、この1次元セルオートマトンでは、行方向へのコピーが次々と新しい世代をつくり出す「繰り返し操作」となる。もっと簡単な例でいえば、セルA1に値1を入れておき、セルA2に「=A1+1」と入れ、セルA2をA3, A4, A5, …とコピーしていけば、それぞれのセルには、2, 3, 4, 5, …の値が入る（表示される）。これは、たとえば、C言語系のプログラムにおける、`for (int i=1, i<=999, i=i+1) print (i);` のような繰り返しに相当するとみてよいであろう。

さらに授業では、近傍が両隣のセル2つである場合の1次元セルオートマト

ンにおいて、あらゆるルールに基づく挙動を簡単に調べるためのワークシートを作成している。近傍が両隣2つの場合のルールは256通りあり<sup>(7)</sup>、これを0～255の整数に対応付け、その整数を与えれば結果を計算し表示するようにする。

そして、それらの結果はルールによって劇的に違ったものとなる。ウルフラム (S. Wolfram) が指摘するように、1次元セルオートマトンはその結果のパターンに基づいていくつかに分けられる。<sup>(8)</sup>すなわち、世代が進むと、①全く変化しなくなる、②周期的変化を繰り返す、③無秩序な変化を繰り返す、④以上のパターンのどれにも当てはまらない複雑なパターンになる、のいずれかになる。④を③から区別することは現実には困難であるが、単純なルールがさまざまなパターンのバリエーションや複雑なパターンを創発することをみる。また、ルールによってはフラクタル図形（たとえば、シェルピンスキーのギャスケット）などが生成される様子もみる。

## (2) 2次元セルオートマトン

2次元セルオートマトンでは、Excelのワークシート上の2次元の表で1世代分のオートマトンを、そのままのイメージで表現する。そして、その次世代は、1次元セルオートマトンと同様に別の表にルールを記述することによって作成される。しかし、Excelは2次元の表を扱うソフトウェアであり、多くの表を関連づけて（たとえば3次的に）扱うことは困難である。また、表を次々と別の場所につくっていくのは、ワークシートの容量の面においても、あるいは表示方法のわかりやすさとしても適切ではない。

そこで、授業では、現世代と次世代の2世代分のみで2次元セルオートマトンの表を確保し、次世代の表のセルに現世代からの生成ルールを記述し、次世

---

(7) 対象セルと近傍の2つのセル（合計3つのセル）の0, 1（生き死に）のパターンは $2^3 = 8$ 通りであり、それぞれに次世代の0, 1を対応させたものがルールに他ならないので、 $2^8 = 256$ 通りとなる。

(8) Wolfram (1984) . ウルフラムはさらに近傍を広げた1次元セルオートマトンに基づいて考察している。

代の計算結果の値を再び現世代にコピーする操作を繰り返すという方法を採用している。Excel ではセルの値が変更されれば、これに影響するセルの値は自動的に再計算されるので、コピーの結果、さらに次の世代が自動的に計算される。このコピー操作を繰り返せば、1つの表の中で世代が進展していくことになる。

図3は、その方法で実装したライフゲームである。ライフゲームは1970年にコンウェイ (J. H. Conway) によって考案された代表的な2次元セルオートマトンである。

ライフゲームの近傍は周囲の8個のセル(ムーア近傍)であり、次世代の生き死に決定のルールは、以下の2つのみである。<sup>(9)</sup>

- ・対象のセルが死んでいるとき、近傍に3個の生きているセルがある場合に次世代に「生き」となり、それ以外は死んだままである。
- ・対象のセルが生きているとき、近傍に2個または3個の生きているセルがある場合に次世代に生きつづけ、それ以外は「死に」となる。

もちろん、このルールは、Excel の IF 関数を使って容易に記述できる。

---

(9) Poundstone (1985) など。



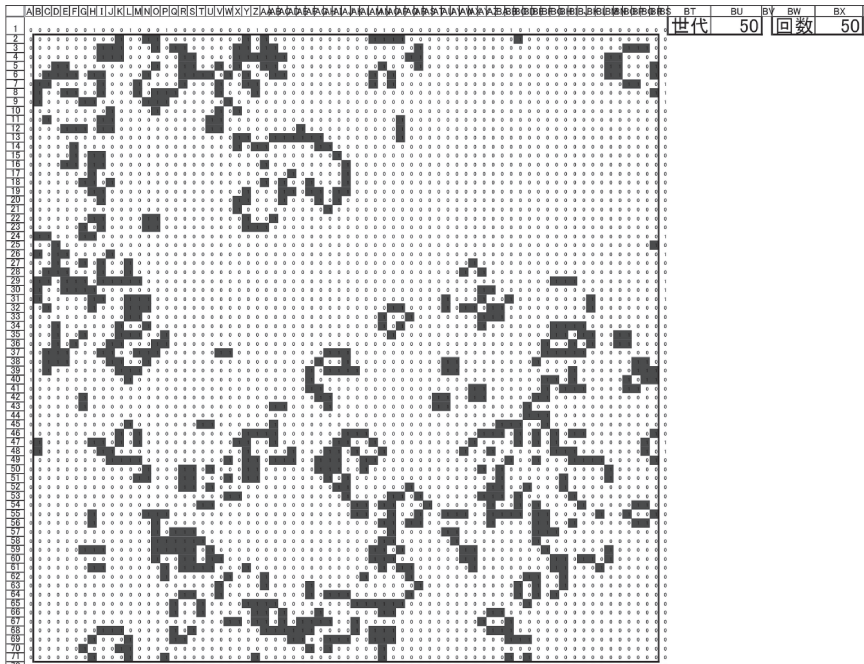


図3 ライフゲーム（トーラス状）のExcelワークシート  
（ランダムな初期値から始めた50世代目）

なお、次世代から現世代への値のコピーは、図4のVBAプログラム4行目の代入文1つで書くことができる。<sup>(10)</sup>ここで、ワークシート名 After のセル範囲“B2:BR71”が次世代のオートマトンであり、ワークシート名 Before の同セル範囲が現世代のオートマトンである。After から Before への代入（コピー）が、For ～ Next ループで times 回繰り返される。繰り返し回数などの変数（パラメータ）の与え方は、ワークシート上の値から代入するなどさまざまに工夫することができる。

(10) もちろん、代入ではなく、セルのコピーと値の貼り付け操作（Copy メソッドと PasteSpecial メソッドの Paste 引数 xlPasteValues）でもできるが、代入の方が記述が簡潔であり実行も高速である。表示の速さを調整するために、Windows の Sleep 関数を用いて処理を一定時間中断させることもできる。

```

1 Sub LifeGame()
2 .....
3   For i = 1 To times 'times: 繰り返し回数が代入されている変数
4     Worksheets("Before").Range("B2:BR71").Value = Worksheets("After").Range("B2:BR71").Value
5   Next i
6
7 End Sub

```

図4 ライフゲームのVBAプログラム

なお、1次元セルオートマトンと同様に、オートマトンの周囲のセルを反対側のセルの値を参照するようにしておけば、端や隅が特殊ではないトーラス状の2次元セルオートマトンを、ルールを変更することなく簡便に作成できる。

また、授業では多数派モデルの<sup>(11)</sup>ような2次元セルオートマトンの社会的な応用も扱っている。

### 3.2 社会ネットワーク分析

社会ネットワーク分析では一般的なネットワーク構造を扱う。授業では単純連結グラフを扱っている。

近年研究が進んでいる「複雑ネットワーク」の知見に基づけば、社会ネットワークをはじめとする現実のネットワークには、スモールワールド性、クラスター性、スケールフリー性という重要な3つの特性がある。<sup>(12)</sup>それぞれ簡単に説明すると、ネットワークが大規模であっても頂点間の距離は小さいという特性、クラスターと呼ばれる頂点間に辺が密に存在する部分が比較的多いという特性、頂点ごとの連結している辺の数の差が著しいという特性である。こうした特性を分析するためには、セルオートマトンの格子のような特殊な構造では不可能であり、一般的なネットワーク構造を扱う必要がある。

社会ネットワーク分析のいくつかはExcelの基本機能でも可能である。また、

(11) 近傍の多数派の影響を受ける2次元セルオートマトンによるモデル。たとえば、ファッションの流行やデファクト・スタンダード競争など、身近な人の影響を強くうける社会現象のシミュレーションに用いられる。たとえば、Gilbert et al. (1999) 邦訳142 - 145 ページ。

(12) たとえば、増田・今野 (2006) など。ただし、これら複雑ネットワークの3つの特性をどう分類するかについては、いくつかの主張がある。

最近では、社会ネットワーク分析のためのツールもいくつか存在する<sup>(13)</sup>。授業では、なるべく Excel を使い、場合によっては VBA で計算し、さらには Excel のテンプレートとして提供されているオープンソースの分析ツール<sup>(14)</sup>を用いている。

以下、一般的なネットワーク構造の Excel での表現と、そのデータ構造を用いた VBA プログラムの例としてのウォーシャル・フロイド法による距離行列の計算について簡単に説明する。<sup>(15)</sup>

(1) ネットワークの Excel での表現

一般的なネットワーク構造を Excel で表現する方法はいくつか考えられる。授業では、辺を構成する 2 頂点のリスト（辺リストと呼んでいる）、頂点ごとの辺で連結されている頂点のリスト（頂点リスト）、頂点間に辺が存在するかどうかを示す行列（隣接行列）の 3 つの方法を用いている。また、それらの間の Excel による変換方法も示している。この 3 つは、ネットワークの表現として非常に一般的であり、多くのネットワーク分析ツールも採用している。

図 5 にそれらの簡単な例を示す。

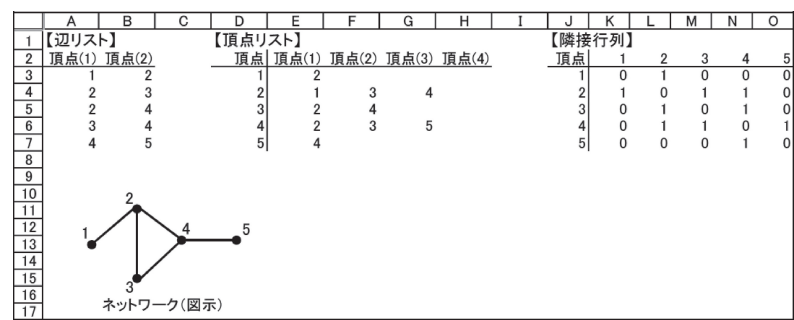


図 5 ネットワーク構造の Excel による表現

(13) 代表的なものとして、Pajek や UCINET のようなネットワーク分析用の専用ソフトウェアや統計ソフト R の sna パッケージなどがある。

(14) NodeXL という Excel テンプレートを用いる。これは、マイクロソフトやワシントン大学、コーネル大学、メリーランド大学などの研究者によって開発されたオープンソースのソフトウェアである。

(15) より詳しくは、授業のテキストとしても用いている牧野（2011b）を参照のこと。

## (2) VBAプログラムの例—距離行列の計算

授業では、こうしたデータを用いて、さまざまなネットワークの特徴量（特性を示す指標）を計算している。いくつかの特徴量は Excel の機能を用いて比較的簡単に計算できる。また、Excel の機能では実現困難あるいは不可能であるが、VBA を用いれば比較的簡単に実現できる特徴量もある。授業では、全頂点間の最短距離を示す行列（距離行列）の計算や、各頂点がクラスターをどのくらいもつかを示す指標（クラスター係数）の計算を VBA でプログラミングしている。その一方で、実現が相当に困難で、とりわけ計算効率に優れたアルゴリズムを考えること自体が研究課題となっているような特徴量もある。<sup>(16)</sup> このような特徴量の計算は、必要に応じて分析ツールによって計算することを推奨している。

ここでは、社会ネットワーク分析において重要であり、さらには一般のネットワーク問題（たとえば輸送問題など）にも広く応用される距離行列の計算について説明する。

そのための簡単で比較的効率のよいアルゴリズムとして、ウォーシャル・フロイド法がよく用いられる。それは、以下のようなアルゴリズムである。<sup>(17)</sup>

①距離行列  $D$  を以下のように初期化する。

- ・ 同一頂点間の距離を 0 にする。
- ・ 隣接頂点間の距離を 1 にする。
- ・ その他の頂点間の距離を無限大にする。

②すべての頂点  $v_k$  について、 $D$  のすべての成分  $d_{ij}$  に対して、 $d_{ik} + d_{kj} < d_{ij}$  であれば  $d_{ij} = d_{ik} + d_{kj}$  とする。

(16) たとえば、媒介中心性をはじめとするネットワークにおける媒介や情報伝達に関する中心性の計算があげられよう。

(17) 古林 (1999)。

このアルゴリズムは非常に簡単なものであるが、VBA を用いないと実装できない。図 6 に距離行列を求めるためのワークシートと VBA プログラム例を示す。プログラム中、距離行列の初期化（4～11 行目）では、隣接行列を修正する形で初期の距離行列を作成するようになっている。

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	【隣接行列】							【距離行列】					
2	頂点	1	2	3	4	5		頂点	1	2	3	4	5
3	1	0	1	0	0	0		1	0	1	2	2	3
4	2	1	0	1	1	0		2	1	0	1	1	2
5	3	0	1	0	1	0		3	2	1	0	1	2
6	4	0	1	1	0	1		4	2	1	1	0	1
7	5	0	0	0	1	0		5	3	2	2	1	0
8													
9	頂点数												
10	5												

```

1 Sub Dist()
2   Dim i As Integer, j As Integer, k As Integer, d As Integer, vn As Integer
3   vn = Range("A10") ' 頂点数が"A10"に格納されている
4   ' -----初期化 "I3"（距離行列の左上）を(1,1)で相対参照
5   For i = 1 To vn
6     For j = 1 To vn
7       If Not ((i = j) Or (Range("I3").Cells(i, j) = 1)) Then
8         Range("I3").Cells(i, j) = 999 ' -----無限大は999で代用
9       End If
10    Next j
11  Next i
12  ' -----ウォーシャル・フロイド法
13  For k = 1 To vn
14    For i = 1 To vn
15      For j = 1 To vn
16        d = Range("I3").Cells(i, k) + Range("I3").Cells(k, j)
17        If d < Range("I3").Cells(i, j) Then
18          Range("I3").Cells(i, j) = d
19        End If
20      Next j
21    Next i
22  Next k
23 End Sub

```

図 6 ウォーシャル・フロイド法による距離行列の計算

この多重ループをプログラミングの経験のない学生が理解することは簡単ではないかもしれない。しかし、一般のプログラム言語の場合は2重配列を用いてこの処理を実装しなければならない。一般プログラミング言語で多重配列の繰り返し処理を理解させることは経験上かなり困難である。

一方、VBAを用いる場合は、2重配列ではなく、実体として認識しやすいExcelワークシート上の表が操作対象となるので、より理解させやすいものと考えられる。たとえば、このプログラムの前段階として、ワークシート上の表にそのセル番地や行番号と列番号の和を代入するプログラムを作成するなど順を追って説明していけば、実際のワークシート上の表との対応関係で具体的に理解できるので、プログラミング言語の2重配列に比べれば相当に理解させやすいものと考えている。

#### 4. おわりに

最後に、2011年度前期に開講した授業科目「社会ネットワーク分析」の評価であるが、同科目の受講生が50人未満であったため、本学部で定められている授業評価自体は実施していない。

しかし、受講生が少ないながらも、授業中の課題提出状況は良好であり、期末試験の成績も、授業内容や問題、受講者数が同じではないので一概にはいえないが、以前のC言語を用いた授業科目「経済プログラミング」と比較して向上した。そして何よりも、プログラミング言語の開発環境や文法の習得に割く時間が少なく済んだために、具体的応用である社会シミュレーションやネットワーク分析自体に多くの時間を割くことができ、それらを受講生自身で実装できたことは大きな成果とみることができる。「経済プログラミング」では1次元セルオートマトンに到達するまでが相当に困難であったことと比較すると大きな前進であると考えられる。

以上のことから、この授業「社会ネットワーク分析」に一応の教育効果があったと考えている。今後さらに検証と改善を続けていくつもりである。また、この授業をベースにさらに高度なプログラミング教育を実施し検証していくこととしたい。

## 参考文献

- 深町修一 (2010) 「文系の学生に対するコンピュータプログラミング教育の一考察」『福岡国際大学紀要』No.23, 39-45 ページ。
- 古林隆 (1999) 「Excel によるネットワーク計画法 (2) —最短路と配達路」『オペレーションズ・リサーチ—経営の科学』Vol.44, No.5, 253-257 ページ。
- Gilbert, N. and Troitzsch, K. G. (1999), *Simulation for the Social Scientist*, Open University Press. (井庭崇・岩村拓哉・高部陽平訳『社会シミュレーションの技法—政治・経済・社会をめぐる思考技術のフロンティア』日本評論社, 2003。)
- 萩谷昌己 (2007) 「Excel でプログラムを書く」<http://lecture.ecc.u-tokyo.ac.jp/~shagiya/excel.pdf>
- 牧野真也 (2011a) 「Excel による社会シミュレーション I—セル・オートマトン—」 *Working Paper Series* (和歌山大学経済学部), 11-02。
- 牧野真也 (2011b) 「Excel による社会ネットワーク分析 I—ネットワークの基本特徴量—」 *Working Paper Series* (和歌山大学経済学部), 11-03。
- 増田直紀・今野紀雄 (2006) 『「複雑ネットワーク」とは何か—複雑な関係を読み解く新しいアプローチ—』講談社。
- Poundstone, W. (1985), *The Recursive Universe: Cosmic Complexity and the Limits of Scientific Knowledge*, Contemporary Books. (有澤誠訳『ライフゲームの宇宙・新装版』日本評論社, 2003。)
- Wolfram, S. (1984), “Universality and Complexity in Cellular Automata,” *Physica*, Vol. D10, pp.1-35.