

<研究ノート>

数値計算ツールと経済学学習

—商用ソフトからフリーソフトウェア Octave へ—

河 合 勝 彦

目 次

- 1 はじめに
- 2 数値計算ツールと経済学
- 3 Octave による経済シミュレーション分析
 - 3.1 Octave とは
 - 3.2 クールノー寡占モデル
 - 3.3 応用一般均衡
 - 3.4 動学的マクロモデル
 - 3.5 エージェントベース・シミュレーション
 - 3.6 その他
- 4 まとめと課題
- A Octave の入手およびインストール方法
 - A.1 入手方法
 - A.2 動作環境およびインストール方法
- B ソースリスト
 - B.1 クールノー寡占モデル
 - B.2 2部門応用一般均衡
 - B.3 動学的マクロモデル
 - B.4 ライフゲーム

はじめに

本稿は数値計算ツールの経済学分野での活用法について概観する。ただし、

キーワード：数値計算，経済学教育，経済シミュレーション，フリーソフトウェア

経済学研究のフロンティアにおいて、いかに数値計算ツールを利用するかといった問題は本稿のレベルを越えている。むしろ、経済学を学ぼうとする学部生、および経済系大学院修士レベルの学生が、どのように数値計算ツールを効率的に活用できるかについてを、いくつかの実用的な例題を解くことによって考察したい。

さて、従来の経済学においては、統計的な分析を主目的とする計量経済学分野を除き、コンピュータによる数値計算が必要とされる場面は非常少なかった。しかしながら、経済理論に基づいた現実的な政策提言をおこなう場合、経済モデルによる質的 (what, how) な分析に加え、量的 (how much) な分析も必要であることは言うまでもない。さらに、経済 (政策) モデルを精緻化¹⁾ しようと、非線形性や不確実性などをモデルに導入した場合、陽的な理論解 (closed form) を導出することが非常に困難になっていくという問題を抱えていた。そこで必要とされたのが、数値解析的なシミュレーション分析である。

幸いなことに、昨今のコンピュータの飛躍的な性能向上と、プログラミングの手間を軽減するユーザフレンドリーなツールの利用によって、必ずしも数値計算の専門家ではない社会学者であっても、数値解析モデルによるシミュレーション分析が可能になってきている。

ただし、『Linux 数値計算ツール』[30]のなかで大石が、

初心者が数値計算の本を読んで、C言語やC++などで数値計算のプログラムを作ったとしても、プロの作った数値計算のプログラムの100倍は実行速度が遅く、また、安定性にも欠けたものしかできないといわれている。

と述べているように、初心者が数値計算アルゴリズムを実用的な形でコーディングするためには、非常に多くの手間と熟練を必要とすることだろう。したがって、本稿でも、いたずらに高級言語による数値計算²⁾を紹介するこ

1) 必ずしもモデルの大規模化を意味しなさい。

2) この分野における定評のある参考書としては、Press, et al.[31]がある。

とはせず、専門家の利用にもたえうる多くの実用的な組込み関数を提供し、かつ初心者にもやさしい言語体系をもつ数値計算ツール、GNU Octave[7]を利用することにする³⁾。Octaveによる独自の数値計算アルゴリズム開発や、アルゴリズム間のパフォーマンスの比較検討については、また別の機会を設けたい⁴⁾。

本稿の構成は以下の通りである。まず第2章では、経済学における数値計算ツール利用の意義を概観する。さらに第3章では、フリーソフトウェア Octave の簡単な紹介と、Octaveを利用して実際にどのように経済学的な分析をおこなうことができるかということの説明をする。最終章ではまとめと展望をおこなう。さらに、付録では Octave の入手方法とインストール方法について簡単に解説する。例題を解く Octave スクリプトの全ソースリストも付録に掲載する。

2 数値計算ツールと経済学

Computational Economics 分野における、最新かつ最先端の研究書である *Applied Computational Economics and Finance* において、Miranda and Fackler [26, p.xv]は以下のように述べている。

Many interesting economic models cannot be solved analytically using the standard mathematical techniques of algebra and calculus. Models that cannot be solved in this way are often applied economic models that attempt to capture the complexities inherent in real-world economic behavior.

つまり、通常の数式展開だけでは解くことができない経済分析モデルは数

3) Octave は無料で入手可能なフリーソフトウェアである。ただし、フリー（自由）とは、ソフトウェアのソースが公開されており、誰でも自由に改変することができるという意味で使用している。必ずしもそのソフトウェアが無料（for free）ということとは意味していないが、ネットワーク接続料を無視すれば、Octave は無料で入手することが可能である。第3章を参照のこと。

4) 本文中でも引用した、Judd[13]およびMiranda and Fackler[26]は、Computational Economics 分野におけるアルゴリズム開発の最先端の研究書物である。両著とも本稿作成のために非常に参考になった。

多く存在しており、複雑な現実経済を対象とする応用経済学分野において、そのようなモデルは特に多いというのである。こうした分野においては、今後、数値解析的分析手法は非常に有用なものとなっていくであろう。

例えば、Shoven and Whalley[34]を代表とする応用一般均衡分析やDynamic Fiscal Policy と呼ばれる Auerback and Kotlikoff[1]を代表とする世代重複モデルによる動学シミュレーションなどは、こうした数値解析的応用経済学分野の範疇にある。

その一方、*Numerical Methods in Economics* において、Computational Economics 分野における第一人者 Judd[13, p.3]は以下のように指摘する。

Computers are changing economic science, offering economists the opportunity to examine economic issues with far greater generality and precision. (中略) More important and radical are the new ways economists are using computation to explore economic theories in far greater detail than pure theory alone can, to determine which problems are important and which are not, and to examine new models of agent behavior and social interactions.

要するに、コンピュータの利用が、純理論的なモデル単独ではなしえない、経済理論の一般化と精緻化を可能にするというのである。例えば、最近の大学院レベルの標準的なマクロ教科書のひとつである Ljungqvist and Sargent [33]は、再帰的動学モデルを用いてマクロ分析をおこなっている。しかしながら、その多くのモデルについて陽的な理論解を得ることが非常に困難であり、数値計算ツールを併用した数値解析的分析が多数なされている。

さらに、理論モデルをベースとしながらも、数値解析的シミュレーションによって実際のモデル解析をおこなっている例としては、学部レベルの中級マクロ教科書、Miller and Upton[25]およびAuerback and Kotlikoff[2]や、一般均衡モデルの概説書、Dinwiddy and Teal[5]をあげることができるだろう。

以下、本稿では、上に述べた問題意識および既存の研究成果のもとに、経済学教育における数値計算ツール (Octave) の活用について考えていきたい。

3 Octave による経済シミュレーション分析

3.1 Octaveとは

ウィスコンシン大学の James B. Rawings とテキサス大学の John G. Ekerdt によって開発が着手された Octave は、当初、化学分野⁵⁾の学部生向け教科書の付録ソフトにしか過ぎなかったという。しかしながら、GNU GPL[9]にそったフリーソフトウェアとしてリリースされるとともに、その開発のリーダーとなった John W. Eaton を中心にオープンな開発が地道に継続されることになり、現在では、化学分野以外でも利用可能な汎用数値計算ツールとして確固たる地位を築きつつある[7, p.3]。

Octave は、

Octave is a high-level interactive language, primarily intended for numerical computations that is mostly compatible with MATLAB.

と、そのオフィシャルサイトの説明にもあるように、数ある商用数値計算ツール（言語）のなかでも最も有名なもののひとつである MATLAB[24]を参考にして開発されている。例えば、先に参考として挙げた大石は、MATLAB と Octave の特徴を表 1, 2 のようにまとめている。

Octaveで利用可能な数値計算の機能を一覧にすると、表 3 のようになる。経済学でもよく利用される数値計算機能のほぼ全てがどカバーされていることがわかるだろう。

技術的観点からまとめると、MATLAB や Octaveは、線形計算の有名な FORTRAN サブルーチン集である LINPACK および EISPACK をまとめた LAPACK⁶⁾を基礎にして開発された数値計算インタプリタである[30]。もっとも、プログラミングのプロではない社会学者にとって、FORTRAN や C で LAPACK のライブラリを呼び出すプログラムを作成することは並大抵な

5) **chemical reactor design**

6) これらの数値計算ソフトを含めた数学ソフトウェア一般については、Netlib[27]が非常に参考になる。

表1 MATLAB の特徴 (大石[30, p.2])

- 欧米では教育, 研究の場で盛んに用いられている
- インタプリタ言語である
- 高速な命令があり, 実用的である
- つねにバージョンアップされている
- 命令が標準になっている

表2 Octave の特徴 (大石[30, p.3])

- MATLABとほぼ同様の仕様である
- 高速で高品質の組込み関数を持つ
- フリーで出来がよい

表3 Octave の数値計算機能

- 行列演算
- 確率, 統計
- 非線型同時方程式
- (不) 定積分
- 微分方程式
- 最適化問題
- 基本的な数式処理

ことではない。多少の速度は犠牲にしても堅固でバグの少ない信頼性の高いプログラミングを組むことが必要になってくる。そして、それを可能にするのがMATLABやOctaveというインタプリタ言語なわけである⁷⁾。

しかしながら、河合[17]でも指摘したように、高価な商用ソフトを利用することは、現在の経済学教育の体系上、コスト的にも非常に難しい。したがって、本稿の経済シミュレーションにおいては、全面的にOctaveを採用することにする⁸⁾。以下、数値計算が必要な代表的な例題を取り上げ、Octave

7) なお、OctaveはCやPascalにも見られるような構造体によってデータを扱うことは可能であるが、MATLABには既に実装されているオブジェクト言語の体裁は、未だ整えられていない。

8) MATLABとOctaveには微妙な文法上の違いがあるが、互換性をもたせるためのプログラム修正はさほど難しいものではない。ソフト付属マニュアルや公式Web

によって実際に数値計算をおこなってみたい。

3.2 クールノー寡占モデル

ある同質な財を生産する企業が複数(N)存在するとする。当該財に対する逆需要関数は、

$$p(Q) = Q^{-\frac{1}{\eta}}, \quad \text{where } Q = q_1 + q_2 + \dots + q_N \quad (1)$$

によってあらわされるものとしよう。なお、上式から容易にわかるように、需要の価格弾力性は、すべての生産量において一定の値(η)をとるものとする⁹⁾。

企業 i は、他企業の供給量 Q を所与として自己の利潤(π)

$$\pi_i(Q) = P(Q)q_i - TC_i(q_i), \quad \text{for } i=1,2,\dots,N \quad (2)$$

を最大化する。

ここで、企業の費用関数(TC)は、2次形式の以下のものを仮定する。

$$TC_i(q_i) = \frac{1}{2}c_i q_i^2, \quad \text{for } i=1,2,\dots,N \quad (3)$$

よって、その利潤最大化条件は、

$$\frac{\partial \pi_i}{\partial q_i} = P(Q) + P'(Q)q_i - C'_i(q_i) \quad \text{for } i=1,2,\dots,N \quad (4)$$

を0とおいた、以下の式となる。

$$Q^{-\frac{1}{\eta}} - \left(\frac{1}{\eta}\right)Q^{-\frac{1}{\eta}-1} q_i - c_i q_i = 0, \quad \text{for } i=1,2,\dots,N \quad (5)$$

式(5)において陽的な解を求めることは明らかに不可能である。したがって、通常のマクロ経済学中級教科書[22, 36]では、線形の逆需要関数および線形の費用関数を利用することによって、この難題を解消している。一方、本稿ではこれを数値計算ツールによって明示的に解くことを課題とする。

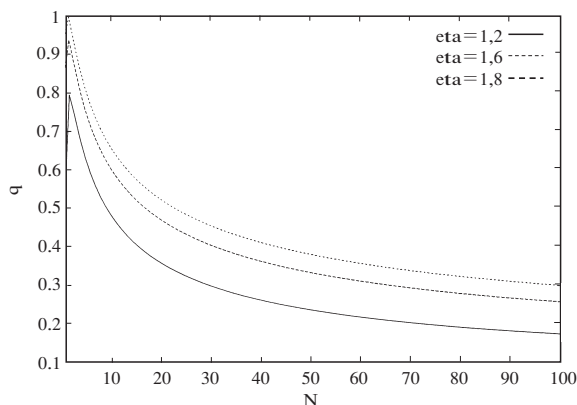
では、Octave でこの問題を解いてみよう¹⁰⁾。この問題は非線形の同時方

サイトを参考にしてもらいたい。

9) 需要の価格弾力性の導出には、式(1)の両辺の対数をとって微分すればよい。

10) 基本的なプログラミングの枠組みはMiranda and Fackler[26, pp.35-36]を参考に

図1 クールノー寡占モデル



程式として定式化されており，これを解くためには，Octave の組み込み関数 `fsolve`¹¹⁾ が利用できる。需要の価格弾力性 η と参入企業の数 N を変化しながら，連続してこの方程式体系(5)を解いてみる。結果は，Octave による PostScript 出力によりグラフ化した¹²⁾。図1を参照されたい。なお，これを解くOctave スクリプトは付録 B1 に掲載した。

3.3 応用一般均衡

ここでは，Dixon 他[6]にしたがって，以下のような2部門一般均衡モデル (Stylized Johansen Model) を構築する¹³⁾。

このモデルにおける，変数およびインデックスの意味は，表4,5の通りである。

家計の行動の定式化

家計は与えられた所得制約，

した。

- 11) `fsolve` は MINPACK[27] の `hybrd` サブルーチンをもとに作られている。
- 12) 実際のグラフ描画機能は，`gnuplot`[11]との連携機能により実現される。
- 13) このモデルの詳細については河合[14]を参照のこと。

表4 変数の定義

内生変数	定義	変数の数
Y	家計総収入	1
PC_i	生産物価格	S
PF_f	要素価格	F
$XCOM_i$	総生産物需要(供給)量	S
XH_i	家計生産物需要	S
XF_{fj}	要素需要量	$F \times S$
XC_{ij}	中間生産物需要量	$S \times S$
変数総数		$3S + F + FS + SS + 1$
外生変数	定義	変数の数
$XFAC_f$	総要素需要量	F

表5 インデックスの定義

$S = \{ \text{sector1}, \text{sector2} \}$
$F = \{ \text{labor}, \text{capital} \}$

$$\sum_{j \in S} PC_j XH_i = Y \tag{6}$$

のもとで、その消費レベルを以下のように定式されたコブ・ダグラス型効用関数

$$\max U = \sum_{j \in S} XH_i^{\alpha_j^h}, \quad \sum_{j \in S} \alpha_j^h = 1 \tag{7}$$

を最大化するように決定する。

以上の定式化から、家計の需要関数として、

$$XH_i = \frac{\alpha_i^h Y}{PC_i}, \quad i \in S \tag{8}$$

$$Y = \sum_{f \in F} PF_f XFAC_f \tag{9}$$

が導出される。

企業の行動の定式化

企業は、所与の技術（コブ・ダグラス型）のもとで、

$$XCOM_j = A_j \prod_{i \in S} XC_{ij}^{\alpha_{ij}^c} \prod_{i \in F} XF_{ij}^{\alpha_{ij}^f}, \quad \sum_{j \in S} \alpha_{ij}^c + \sum_{j \in F} \alpha_{ij}^f = 1 \quad (10)$$

生産費用を最小化するすように、

$$\min C_i = \sum_{i \in S} PC_i XC_{ij} + \sum_{i \in F} PF_{ij} XF_{ij} \quad (11)$$

生産をおこなう。

以上の定式化から、企業の要素需要関数として、

$$XF_{fj} = \frac{\alpha_{fj}^f Q_j XCOM_j W_j}{PF_f}, \quad f \in F, j \in S \quad (12)$$

ただし、

$$Q_j = \frac{\prod_{i \in S} (\alpha_{ij}^c)^{-\alpha_{ij}^c} \prod_{i \in F} (\alpha_{ij}^f)^{-\alpha_{ij}^f}}{A_j} \quad (13)$$

$$W_j = \prod_{i \in S} PC_j^{\alpha_{ij}^c} \prod_{i \in F} PF_j^{\alpha_{ij}^f}, \quad i, j \in S \quad (14)$$

が導出され、さらに、中間消費財の需要関数として、

$$XC_{ij} = \frac{\alpha_{ij}^c Q_j XCOM_j W_j}{PC_i}, \quad i, j \in S \quad (15)$$

が導出される。

均衡条件の導出と価格体系

なお、上の最適化条件は、以下の均衡条件、

1. 財および生産要素の需給均等（Market Clearing Conditions）：

$$XCOM_j = XH_j + \sum_{j \in S} XC_{ij}, \quad i \in S \quad (16)$$

$$XFAC_f = \sum_{j \in S} XF_{fj}, \quad f \in F \quad (17)$$

2. すべての部門で超過利潤がゼロ（Zero-Profit Conditions）：

$$\begin{aligned}
 C_j &= PC_j XCOM_j \\
 &= \sum_{i \in S} PC_i X C_{ij} + \sum_{f \in F} PF_f X F_{fj}, \quad j \in S
 \end{aligned}
 \tag{18}$$

を満たさなければならぬ、

さらに、一般均衡モデルでは、体系内の相対価格しか決定することができないため、以下のように、

$$PC_{sector1} = 1
 \tag{19}$$

ニューメラルとして sector1 の産出財価格を 1 に固定する。

データベース

モデルのデータベースとしては、対象とする経済をスナップショットでとらえた、社会会計表を利用する。社会会計表は、従来の産業連関表を、よりミクロデータに整合的になるように拡張したものである。

表 6 は、本モデルのために作成した仮想的データである。この社会会計表の各々の行と列は、対応する経済主体の（ある一定期間のフローでとらえた）受取りと支払いをあらわす。例えば、sector1 は、中間財需要として sector2 に 2 単位の支払いをおこない、生産要素需要として労働（labor）と資本（capital）に各々 1 単位の支払いをおこなう。

カリブレーション

モデルのパラメータは、基準となるデータベース（表 6）を一般均衡体系のもとで再現するように、以下のように、

$$\alpha_i^h = \frac{XH_i}{Y}, \quad i \in S
 \tag{20}$$

$$\alpha_{fj}^f = \frac{XF_{fj}}{XCOM_j}, \quad j \in S, f \in F
 \tag{21}$$

$$\alpha_{ij}^c = \frac{XC_{ij}}{XCOM_j}, \quad i, j \in S
 \tag{22}$$

設定される。

表6 社会会計表

	sector1	sector2	労働	資本	家計	総収入
sector1	4	2			2	8
sector2	2	6			4	12
労働	1	3				4
資本	1	1				2
家計			4	2		6
総支出	8	12	4	2	6	

Octave によるシミュレーション

このモデルにおいて考えられる最も単純なシミュレーションとして、外生変数（労働，資本）へのショックを考えてみよう。具体的には，労働投入（供給）量を10%減少させ，その一方，資本投入（供給）量を10%増加させてみる。

表7 シミュレーション結果

変数	パーセント変化
$XCOM_{sector1}$	-2.478
$XCOM_{sector2}$	-4.415
$PC_{sector1}$	0.000
$PC_{sector2}$	2.027
PF_{labor}	8.357
$PF_{capital}$	-11.344

本モデルも前のクールノー寡占モデルと同じく，非線型同時方程式体系であるので，Octave の `fsolve` 関数を使って解くことができる。この問題を解く Octave スクリプトは付録 B.2 を参照されたい。

シミュレーションの結果は表7に示す。例えば，その投入量が削減された労働の価格（ PF_{labor} ）は約8.4%上昇し，一方，その投入量が増加した資本の価格（ $PF_{capital}$ ）は約11.3%下落している。

3.4 動学的マクロモデル

Ljungqvist and Sargent[33, pp. 641-643]の定式化を参考に、以下のソーシャル・プランニングモデルを考える。

ソーシャル・プランナーは以下の効用

$$\sum_{t=t_0}^{\infty} \beta^t \left\{ u_1 c_t - \frac{u_2}{2} c_t^2 \right\}, \quad 0 < \beta < 1, \quad u_1 > 0, \quad u_2 > 0 \quad (23)$$

を最大化するように消費量 (c_t) を決定する。ただし、 β はディスカウント因子、 u_1 および u_2 は定数とする。

異時点間を結ぶ所得制約式は、

$$k_{t+1} = (1+r)(k_t + y_t - c_t) \quad (24)$$

である。ここで、 k_t は資本量、 r は利子率、 y_t は労働所得をあらわす。

労働所得の流れは、

$$y_{t+1} = \lambda_0 + \lambda_1 y_t \quad (25)$$

と定式化する。

この問題を二次形式の線形最適制御問題 (linear quadratic optimal control problem)¹⁴⁾ のフレームワークで解くために、

$$\tilde{k}_t = \beta^{\frac{t}{2}} k_t \quad (26)$$

$$\tilde{y}_t = \beta^{\frac{t}{2}} y_t \quad (27)$$

$$\tilde{c}_t = \beta^{\frac{t}{2}} c_t \quad (28)$$

のような変数の変換をおこそう。

すると、この問題は、以下の典型的なstate-spaceフォーム：

$$\max_{\{\tilde{u}_t\}} \sum_{t=t_0}^{\infty} \{ \tilde{x}_t R \tilde{x}_t + 2 \tilde{x}_t W \tilde{u}_t + \tilde{u}_t Q \tilde{u}_t \} \quad (29)$$

制約条件：

$$\tilde{x}_{t+1} = A \tilde{x}_t + B \tilde{u}_t \quad (30)$$

14) 最適制御理論のより一般的な経済学への適用を論じたものにKendrick[19]がある。同書には、一般非線形最適制御問題の線形2次形式問題への近似方法の解説もある。また、最適制御分野におけるMATLABプログラミングを解説した良書に、Ogata[29]がある。

によってあらわすことが可能になる。ただし、変数の内容は以下の通りである。

$$\tilde{x}_t = \begin{bmatrix} \tilde{k}_t \\ \tilde{y}_t \\ \sqrt{\beta^{\frac{1}{2}}} \end{bmatrix}, \quad \tilde{u}_t = \tilde{c}_t \quad (31)$$

$$R = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad W' = \left(0 \quad 0 \quad \frac{u_1}{2} \right) \quad (32)$$

$$Q = -\frac{u_2}{2}, \quad A = \begin{bmatrix} (1+r) & (1+r) & 0 \\ 0 & \lambda_1 & \lambda_0 \\ 0 & 0 & 1 \end{bmatrix} \beta^{\frac{1}{2}}, \quad B = \begin{bmatrix} -(1+r) \\ 0 \\ 0 \end{bmatrix} \beta^{\frac{1}{2}} \quad (33)$$

この問題における最適政策関数 (optimal policy function) は、

$$u_t = -(Q + B'PB)^{-1}(B'PA + W)x_t \quad (34)$$

とあらわすことができる。ただし、 P は、

$$P = R + A'PA - (A'PB + W')(Q + B'PB)^{-1}(B'PA + W) \quad (35)$$

という Riccati 方程式を解いたものである¹⁵⁾。

この問題を Octave によって数値解析的に解いてみる。そのためには、Octaveの組込み関数 `dlqr` を利用することができる。組込み関数を使わない場合には、Kendrick [19, ch.2] や Ogata [29, ch.4] の方法にしたがい、上の式 (35) を再帰的に解くことも可能である。幸いなことに、Octave は行列表示をそのままの形で入力することが可能である。本例題を解く Octave スクリプトは、付録 B.3 に掲載する。

3.5 エージェントベース・シミュレーション

筆者は、河合 [15, 16] において、社会科学におけるエージェントベース・シミュレーションの基礎的な概念説明、およびオブジェクト指向言語によるそのプログラミング手法の解説をおこなった。しかしながら、学部生レベル

15) さらに一般化を含めた詳細な解法については、Kendrick [19] 等を参照のこと。

表8 構造体 (structures) の例

```
agent(1).id = 1;
agent(1).age = 3;
agent(1).sex = 'm';
agent(1).grades = [15 30 20 23];

agent(2).id = 2;
agent(2).age = 5;
agent(2).sex = 'f';
agent(2).grades = [20 40 22 13];
```

の学習にとって、高級言語を利用したプログラミングは非常にハードルが高いものになることは否めない。

ただし、Thorngate[35]が指摘するように、Octaveのような簡易言語でも、簡単なエージェントベースモデルの作成は可能である。具体的なそのプログラミング手法は、構造体 (structures) によりエージェントのデータを構築することである (表8を参照)。さらに、構造体をより拡張した cell 関数を利用することも可能である。

しかしながら、本格的なエージェントベース・シミュレーションをおこなうことは、本稿のレベルを越えている。したがって、エージェントモデルのなかでも最もシンプルなセルラ・オートマタによるモデル、Conway のライフ・ゲーム¹⁶⁾をOctaveによってモデル化してみた。

具体的には、Octave のデータビジュアル化機能 (spy 関数) を使い、セルの遷移過程を追うこともできるようなスクリプトを作成した。そのプログラムは Octave のデータ構造 (行列) を利用することによって、非常に単純明快に作成可能である。C 等の高級言語を使った場合と比べ、1/3 以下のステップ数でプログラムが可能であるといっても過言ではなからう。このライフゲームのソースリストは付録 B.4 を参照されたい。

16) より詳細な説明およびC言語によるそのモデル化例については、Flake[8, ch. 15]を参照のこと。

3.6 その他

応用経済学の範疇に入る数値解析モデルは、以上にあげた分野以外にも数多く存在する。Octave（もしくは同等なツール）によるモデル化が可能かつ効率的であろう経済学関連の研究例（分野）としては、Martinez and Martinez[23]（確率モデル）、Mangel and Clark[20]（確率的ダイナミックプログラミング）、Marimon and Scott[21]（確率的動学マクロ）、Haupt and Haupt[12]（遺伝的アルゴリズム）、Prisman[32]（Financial Optimization）、Brandimarte[3]（Financial Optimization）等があるが、詳細な検討はまたの機会に譲りたい。

4 まとめと課題

本稿は、数値計算ツールの経済学学習における利用法について概観した。さらに、オープンソースソフトウェアの数値計算ツール、Octave を取り上げ、その経済学学習への利用可能性について考察した。

本質的ではない冗長な計算をコンピュータにまかせたり、自分の数値計算とコンピュータの結果をくらべたりと、数値計算ソフトの経済学学習における活用により、学習者は、問題の定式化、および導出された命題の現実妥当性（反証可能性）に、より注意を払うことが可能となる。それは、結果として、経済学の本質の理解へと学習者を導くだろう。

しかしながら、数値計算ソフトの価格高騰、コストパフォーマンスの欠如は、その一般的普及を妨げている。筆者は、そうした問題に対処するために、無料のオープンソースソフトウェアを利用してみることも一案足りうることを本稿で示した。ここで紹介した Octave は、用途により多少の機能的拡充が求められるものの、商用ソフトウェアにも勝るとも劣らない実用性と機能を持つことが理解されたであろう。

今後の筆者の個人的課題として、この Octave を利用した、経済学分野の例題集、経済モデルテンプレートを開発していくことを計画している。もちろんこうした教材は、紙の媒体のみならず、電子的媒体を提供することによ

ってユーザの利便性を向上させることが必要である。筆者は大学内部、外部からのサポートを募りながら、こうして豊富にそろえた教材、研究素材を、コンピュータ経済学 (Computational Economics[18]) ポータルサイトの開設によって、広く提供していくことを計画している。識者からの広いアドバイスを希望する。

付録A Octave の入手およびインストール方法

A.1 入手方法

GNU Octave は、その開発および配布のオフィシャルサルト：

- <http://www.octave.org/>

から無料で入手することができる。

A.2 動作環境およびインストール方法

Octave は、Linux や FreeBSD といった Unix-like システムや、OS/2 および Microsoft Windows で動作させることができる。ただし、MS Windows 上で使用する場合には、Cygwin¹⁷⁾ 等の Unix 環境をインストールしておく必要があり、多少インストールが煩雑になる。

幸いなことに、Cygwin + gnuplot + Octave をオールインワンにパッケージ化したインストーラが、2002年9月にリリースされた。興味のある読者は試してみることをお勧めする。

なお、筆者が本稿のために利用した OS 環境は、Windows 2000, Vine 2.6, Red Hat 7.3, および Debian GNU/Linux 3.0 (Woody) である。このいずれにおいても、Octave のバイナリーパッケージの入手が可能である。ただし、最新の Octave ソースをコンパイルして利用するためには、GNU GCC (C++) [10]の環境が必要となる。

本稿の例題ために使用した Octave のバージョンは、表9の通りである。

17) Red Hatにより開発されている Cygwin は、その配布サイト[4]から無料で入手可能である。

表9 利用バージョン

オペレーティングシステム	Octave バージョン	ソースコンパイル/ バイナリーパッケージ
Vine 2.6	2.1.36	コンパイル
Debian 3.0	2.1.4 + octave-forge	パッケージ
Redhat 7.3	2.1.39 + octave-forge	コンパイル
Windows 2000	2.1.36 + octave-forge	パッケージ

このいずれの環境でも Octave は問題なく動作した。

付録B ソースリスト

B.1 クールノー寡占モデル

```

1  #!/usr/bin/octave -qf
2  #
3  # (c) 2002 Katz Kawai
4  # katz@andrew.ac.jp
5  # $Id: c7.m,v 1.1.1.1 2002/12/22 04:56:31 katz Exp $
6
7  disp ("Solving a cournot model.")
8
9  function y = cn (q)
10   global c
11   global eta
12   ie = -1/eta;
13   y = sum (q) ^ie + ie*sum (q) ^ (ie-1) *q-diag (c) *q;
14 endfunction
15
16 global c
17 global eta # elasticity
18
19 N = 100; # the number of firms
20 disp ("The number of firms is:") , disp (N)

```

```
21 x1 = zeros (N,1) ;
22 x2 = zeros (N,1) ;
23 x3 = zeros (N,1) ;
24 disp ("Please wait ...")
25
26 j = 1;
27 eta = 1.2
28 for i = 1:N
29     c = 0.5 * ones (1,i) ;
30     xx = fsolve ("cn", c) ;
31     j = j + 1;
32     x1 (j) = xx (1) ;
33 end
34
35 j = 1;
36 eta = 1.6
37 for i = 1:N
38     c = 0.5 * ones (1,i) ;
39     xx = fsolve ("cn", c) ;
40     j = j + 1;
41     x2 (j) = xx (1) ;
42 end
43
44 j = 1;
45 eta = 1.8
46 for i = 1:N
47     c = 0.5 * ones (1,i) ;
48     xx = fsolve ("cn", c) ;
49     j = j + 1;
50     x3 (j) = xx (1) ;
51 end
```

```
52
53 # plotting the results
54
55 xlabel ('N') ; ylabel ('q')
56 hold on
57 gplot [1:N] x1 title "eta=1.2"
58 gplot x2 title "eta=1.6"
59 gplot x3 title "eta=1.8"
60 gset term postscript
61 gset output "./cn.ps"
62 pause
63 replot
```

B.2 2 部門応用一般均衡

```
1 #!/usr/bin/octave -qf
2 #
3 # (c) 2002 Katz Kawai
4 # katz@andrew.ac.jp
5 # $Id: k4.m,v 1.5 2002/12/22 14:36:29 katz Exp $
6
7 disp ("solving two sector general equilibrium")
8
9 global sam
10 global o_y
11 global o_pc
12 global o_pf
13 global o_xcom
14 global o_xh
15 global o_xf
16 global o_xc
17 global o_w
```

```
18
19 # social accounting matrix
20 sam = [0.4 0.2 0.0 0.0 0.2 0.8
21 0.2 0.6 0.0 0.0 0.4 1.2
22 0.1 0.3 0.0 0.0 0.0 0.4
23 0.1 0.1 0.0 0.0 0.0 0.2
24 0.0 0.0 0.4 0.2 0.0 0.6
25 0.8 1.2 0.4 0.2 0.6 0.0]
26
27 o_y = sam (5,6)
28 o_pc = [1 1]
29 o_pf = [1 1]
30 o_xcom = sam (6,1:2)
31 o_xh = sam (1:2,5) '
32 o_xf = sam (3:4,1:2)
33 o_xc = sam (1:2,1:2)
34 o_w = [1 1]
35
36 #####
37
38 function [eq] = func (p)
39
40 global sam
41 global o_y
42 global o_pc
43 global o_pf
44 global o_xcom
45 global o_xh
46 global o_xf
47 global o_xc
48 global o_w
```

```

49
50 global xfac
51
52 # calibration
53 a_h = o_xh ./ o_y # (21)
54
55 # a_f (f,j) = o_xf (f,j) / o_xcom (j) # (22)
56 a_f = zeros (2,2)
57 a_f (:,1) = o_xf (:,1) ./ o_xcom (1)
58 a_f (:,2) = o_xf (:,2) ./ o_xcom (2)
59
60 # a_c (i,j) = o_xc (i,j) / o_xcom (j) # (23)
61 a_c = zeros (2,2)
62 a_c (:,1) = o_xc (:,1) ./ o_xcom (1)
63 a_c (:,2) = o_xc (:,2) ./ o_xcom (2)
64
65 # A (j) = o_xcom (j) / prod (o_xc (i,j) ^a_c (i,j) ,i)
    * prod (o_xf (i,j) ^a_f (i,j) ,i) # (10)
66
67 A = zeros (1,2)
68
69 A (1:2) = o_xcom (1:2) ./ (o_xc (1,1:2) ^a_c (1,1:2)
    .* o_xc (2,1:2) ^a_c (2,1:2) ...
70 .* o_xf (1,1:2) ^a_f (1,1:2) .* o_xf (2,1:2) ^a_f (2,1:2) )
71
72 # set variables
73 y = p (1)
74 pc= p (2:3)
75 pf = p (4:5)
76 xcom = p (6:7)
77 xh = p (8:9)

```

```

78  xf = [p (10:11) p (12:13) ]
79  xc = [p (14:15) p (16:17) ]
80  w = p (18:19)
81
82  # set parameters
83
84  q (1:2) = (a_c (1,1:2) .^ (-a_c (1,1:2) ) .* a_c (2,1:2)
            .^ (-a_c (2,1:2) ) ...
85  .* a_f (1,1:2) .^ (-a_f (1,1:2) ) .* a_f (2,1:2)
            .^ (-a_f (2,1:2) ) ) ./ A (1:2) # (13)
86
87  # equations
88
89  eq (1:2) = xh (1:2) ' .- (a_h (1:2) .* y) ./ pc (1:2) ' # (8)
90
91  # eq () = y - pf * xfac' # (9) removing by warlas law
92
93  eq (3:4) = w (1:2) ' .- pc (1) .^a_c (1,1:2) .* pc (2)
            .^a_c (2,1:2) ...
94  .* pf (1) .^a_f (1,1:2) .* pf (2) .^a_f (2,1:2) # (11)
95
96  eq (5:6) = xf (1,1:2) .- (a_f (1,1:2) .* q (1:2) .* xcom (1:2) '
            .* w (1:2) ' ) ...
97  ./ pf (1) # (12)
98
99  eq (7:8) = xf (2,1:2) .- (a_f (2,1:2) .* q (1:2) .* xcom (1:2) '
            .* w (1:2) ' ) ...
100 ./ pf (2) # (12)
101
102 eq (9:10) = xc (1,1:2) .- (a_c (1,1:2) .* q (1:2) .* xcom (1:2)
    ,

```

```

.* w (1:2) ' ) ...
103 ./ pc (1) # (15)
104
105 eq (11:12) = xc (2,1:2) .- (a_c (2,1:2) .* q (1:2) .* xcom (1:2) '
.* w (1:2) ' ) ...
106 ./ pc (2) # (15)
107
108 eq (13:14) = xcom (1:2) .- (xh (1:2) .+ xc (1:2,1)
.+ xc (1:2,2) ) # (16)
109
110 eq (15:16) = xfac (1:2) ' .- (xf (1:2,1) .+ xf (1:2,2) ) # (17)
111
112 eq (17:18) = pc (1:2) .* xcom (1:2) .- ...
113 ( pc (1) .* xc (1,1:2) ' .+ pc (2) .* xc (2,1:2) ' ...
114 + pf (1) .* xf (1,1:2) ' + pf (2) .* xf (2,1:2) ' ) # (18)
115
116 eq (19) = pc (1) - 1 # (19)
117
118 endfunction
119
120 global xfac # exogenous variables
121
122 # giving initial conditions.
123 s (1) = o_y
124 s (2:3) = o_pc
125 s (4:5) = o_pf
126 s (6:7) = o_xcom
127 s (8:9) = o_xh
128 s (10:13) = o_xf
129 s (14:17) = o_xc
130 s (18:19) = o_w

```



```
131
132 xfac = [sam (3,6) sam (4,6) ]
133
134 [x, info] = fsolve ("func", s)
135
136 # giving a shock
137 xfac = [sam (3,6) *0.9 sam (4,6) *1.1]
138
139 [xx, info] = fsolve ("func", x)
140
141 x
142 xx
143
144 # compute the percentage change
145 (xx - x) ./ x
```

B.3 動学的マクロモデル

```
1 #!/usr/bin/octave -qf
2 #
3 # (c) 2002 Katz Kawai
4 # katz@andrew.ac.jp
5 # $Id: s2.m,v 1.1.1.1 2002/12/23 10:19:30 katz Exp $
6
7 u1 = 0.0005
8 u2 = 0.000002
9 r = 0.08
10 b = 0.95
11 l1 = 0.5
12 l0 = 0.9
13
14 R = -[0 0 0
```

```

15    0 0 0
16    0 0 0]
17
18    W = -[0; 0; u1/2]
19    Q = -[u2/2]
20
21    A = [ (1+r)   (1+r)  0
22          0 11 10
23          0 0 1] * b^ (1/2)
24
25    B = [- (1+r) ; 0; 0] * b^ (1/2)
26
27    [k, p, e] = dlqr (A, B, R, Q, W)

```

B.4 ライフゲーム

```

1  #!/usr/bin/octave -qf
2  #
3  # (c) 2002 Katz Kawai
4  # katz@andrew.ac.jp
5  # $Id: l3.m,v 1.1 2002/12/22 12:10:55 katz Exp $
6
7  disp ("Playing a Life Game")
8
9  function nei = life (x)
10   n = [x (2:length (x) ,:) ; x (1,:) ];
11   s = [x (length (x) ,:) ; x (1:length (x) -1,:) ];
12   e = [x (:,length (x) ) x (:,1:length (x) -1) ];
13   w = [x (:,2:length (x) ) x (:,1) ];
14
15   ne = [n (:,length (n) ) n (:,1:length (n) -1) ];
16   nw = [n (:,2:length (n) ) n (:,1) ];

```

```
17 se = [s (:,length (s) ) s (:,1:length (s) -1) ];
18 sw = [s (:,2:length (s) ) s (:,1) ];
19
20 nei = n + ne + e + se + s + sw + w + nw
21 endfunction
22
23 world = rand (80,80) > 0.3
24 for i=1:300
25     spy (world)
26     pause (2)
27
28     world = (world & (life (world) == 2 | life (world) == 3)) ...
29     | life (world) == 3
30
31     endfor
```

参考文献

- [1] Auerback, Alan J., and Laurence J. Kotlikoff, *Dynamic Fiscal Policy*, Cambridge University Press, 1987.
- [2] Auerback, Alan J., and Laurence J. Kotlikoff, *Macroeconomics-An Integrated Approach 2nd ed.*, MIT Press, 1997.
- [3] Brandimarte, Paolo, *Numerical Methods in Finance*, WILEY INTER-SCIENCE, 2002.
- [4] Cygwin, <http://www.cygwin.com>
- [5] Dinwiddy, C. L., and F. J. Teal, *The Two-Sector General Equilibrium Model: A New Approach*, St. Martin's Press, 1988.
- [6] Dixon, Peter B. et al, *Notes and Problems in Applied General Equilibrium Economics*, North-Holland, 1992.
- [7] Eaton, John W. *GNU Octave Manual*, Network Theory Limited, 1997.
- [8] Flake, Gary Willam, *The Computational Beauty of Nature*, MIT Press, 1998.
- [9] The Free Software Definition, <http://www.gnu.org/philosophy/free-sw.html>
- [10] GCC HomePage, <http://gcc.gnu.org/>
- [11] Gnuplot Central, <http://www.gnuplot.info/>

- [12] Haupt, Randy L., and Sue Ellen Haupt, *Practical Genetic Algorithms*, JOHN WILEY & SONS, 1998.
- [13] Judd, Kenneth L., *Numerical Methods in Economics*, MIT Press, 1998.
- [14] 河合勝彦, “経済シミュレーション技法 (1) 応用一般均衡分析の基礎,” 1997年7月, *Journal of Economics and Management*, Vol.42, No.1, pp.99-103.
- [15] 河合勝彦, “経済シミュレーション技法 (2) エージェントベースモデルの基礎,” *Review of Economics and Information Studies* III, Gifu Shotoku Gakuin University, 2002年3月 (印刷中)。
- [16] 河合勝彦, “経済シミュレーション技法 (3) オブジェクト指向とエージェントベースモデル,” 『経済経営論集』 第44巻, 第2号, 23-52ページ, 桃山学院大学経済経営学会, 2002年10月。
- [17] 河合勝彦, “数式処理ソフトと経済学学習,” 『経済経営論集』 桃山学院大学経済経営学会, 2003年1月 (印刷中)。
- [18] Kendrick, David A., “Research Opportunity in Computational Economics- Report of the NSF Workshop on Research Opportunities in Computational Economics,” Center for Economic Research, Department of Economics, The University of Texas at Austin, 1991.
- [19] Kendrick, David A., *Stochastic Control for Economic Models*, McGraw-Hill, 1981.
- [20] Mangel, Marc, and Colin W. Clark, *Dynamic Modeling in Behavioral Ecology*, 1988.
- [21] Marion, Ramon, and Andrew Scott, ed., *Computational Methods for the Study of Dynamic Economies*, OXFORD UNIVERSITY PRESS, 1999.
- [22] 丸山雅祥・成生達彦『現代のミクロ経済学--情報とゲームの応用ミクロ』1997年。
- [23] Martinez, Wendy L., and Angel R. Martinez, *Computational Statistics Handbook with MATLAB*, CHAPMAN & HALL/CRC, 2002.
- [24] The MathWorks, <http://www.mathworks.com/>
- [25] Miller, Merton H., and Charles W. Upton, *Macroeconomics-A Neoclassical Introduction*, Chicago University Press, 1986.
- [26] Miranda, Mario J., and Paul L. Fackler, *Applied Computational Economics and Finance*, MIT Press, 2002.
- [27] The Netlib, <http://www.netlib.org/>
- [28] Octave HomePage, <http://www.octave.org/>
- [29] Ogata, Katsuhiko *Designing Linear Control System with MATLAB*, PRENTICE HALL, 1994.
- [30] 大石進一 『Linux 数値計算ツール』 コロナ社 2000年。
- [31] Press, William H., et al, *Numerical Recipes in C: The Art of Scientific Computing 2nd*

- edition*, Cambridge University Press, 1993.
- [32] Prisman, Eliezer Z. *Pricing Derivative Securities-An Interactive Dynamic Environment with Maple V and Matlab*, Academic Press, 2000.
- [33] Ljungqvist, Lars, and Thomas Sargent, *Recursive Macroeconomic Theory*, MIT Press, 2000.
- [34] Shoven, John B., and John Whalley, *Applying General Equilibrium*, Cambridge University Press, 1992.
- [35] Thorngate, Warren, "Teaching Social Simulation with Matlab," *Journal of Artificial Societies and Social Simulation* vol. 3, no. 1, <http://www.soc.surrey.ac.uk/JASSS/3/1/forum/1.html>
- [36] Shy, Oz, *Industrial Organization*, MIT Press, 1995.

(かわい・かつひこ／経済学部助教授／2002年12月24日受理)