

# 『定量的マクロ経済学と数値計算』 第1回：イントロダクション

## 1 はじめに

本連載は、経済学、特にマクロ経済学で用いられる数値計算の手法と、それを用いることによってどのような分析が可能になるのかを解説することを目的としている<sup>1</sup>。もともと政策志向が強いマクロ経済学では、データに基づく実証的裏付けがあり、なおかつ経済理論に基づいた政策評価にも耐えうる定量的な分析に対する需要が強い。

マクロ経済学においてデータに基づく実証分析というと、学部レベルの教科書でもお馴染みの IS-LM モデルと計量経済学を組み合わせたマクロ計量モデルを思い浮かべる人も多いだろう。また、計量経済学を学んだ読者であれば、VAR (vector autoregression) や構造 VAR といった時系列分析をイメージする人もいるかもしれない。マクロ計量モデルやその基礎になっている IS-LM モデルは、背後にある個人や企業の意思決定を明示的に考慮していないことから、1970 年代以降、ルーカス批判の対象となった (Lucas (1976))。現在のマクロ経済学はそういった問題点を改良すべく、マクロ経済学のミクロ的基礎付けに取り組みながら進歩してきた。

現在、各国の中央銀行などで政策議論にも用いられるニューケインジアン DSGE モデルはその一つの成果である。DSGE は Dynamic Stochastic General Equilibrium の略で、日本語だと (なぜか確率的を意味する Stochastic の部分が省かれて) 動学的一般均衡と呼ばれている。ニューケインジアン DSGE モデルは、その名の通りミクロ経済学の一般均衡理論に基づいていることから、家計や企業の意思決定をベースとしながら、ケインズ経済学的な価格硬直性を取り込んだマクロ経済モデルである。

リーマンショックに端を発した大不況以降、危機を予見出来なかった、あるいは世界大不況のような大きなショックの場合には使えないという評判から DSGE モデルは批判の対象になっている。しかし、現在でも DSGE モデルがマクロ経済学のワークホースモデルの一つであることは間違いない。また、市場の不完全性は必ずしも価格硬直性に集約されるわけではなく、現在のマクロ経済学は様々な市場の不完全性や行動経済学に基づくバイアスなども取り込みながら、経済政策の効果の分析に取り組んでいる<sup>2</sup>。

様々な市場の不完全性といった現実的な要素を取り込みつつ、データとの整合性が取れたモデル

---

<sup>1</sup> 数値計算は、英語の numerical method の日本語訳で、数値解析 (numerical analysis) と呼ばれることもある。理系の書籍では科学技術計算 (scientific computing) と書かれていることもあるが、本連載ではすべて同義として使い分けしない。例えば、クアルテローニ他 (2014) を参照。

<sup>2</sup> 例えば、Gabaix (2016) などがある。

を構築して、現場の政策議論にまでつながる分析を行うためには、伝統的な紙と鉛筆のアプローチだけではどうしても限界がある<sup>3</sup>。そこで、コンピュータを利用して、シミュレーション分析や反事実的 (counter-factual) 実験などを用いて分析を行う数値計算が台頭してきた。数値計算を多用するマクロ経済学は、**quantitative macroeconomics** と呼ばれている。著者たちの知る限り、まだ定訳が存在していないため定量的マクロ経済学と呼称しておくが、定量的マクロ経済学は実証的分野と理論的分野の中間領域として位置づけることが出来る。

## 2 定量的マクロ経済学とは？

定量的マクロ経済学は、実物的景気循環 (Real Business Cycle; RBC) 理論に端を発するといっても良い。景気循環における実物サイドの役割を強調する RBC 理論は、日本では未だに嫌悪感を持つ人も多いようである。しかし、現在のマクロ経済学における RBC 理論は加藤涼氏 (日本銀行) が言うところのピザの生地であり、RBC モデルのフレームワークをベースとして様々な市場の不完全性 (食材) を導入することによって政策分析につなげてきた経緯がある (加藤 (2007))。すなわち、RBC モデルは、文字通りの意味で「実物サイドのみが重要である」というメッセージよりも、現代のマクロ経済学に共通する骨組みとみるべきである。

Kydland and Prescott (1982) が提示した古典的な RBC モデルや上級レベルのマクロ経済学の教科書で紹介されている最初のモデルは、現在のマクロ経済学からすれば非常に (極端に) シンプルであり、そのまま政策的議論につなげることは困難である。しかし、RBC モデルには拡張性があり、それをベースに様々な市場の不完全性を取り込む事で、それぞれの関心に応じた分析につなげることが出来る。例えば、景気循環に限らず、労働や人的資本、各種社会保障制度、財政・金融政策から所得・資産格差まで、応用範囲は実に様々である。

とはいえ、より現実在即した市場の不完全性を導入するとモデルが複雑になりすぎたり (それでも現実経済の複雑さと比較すれば極端に簡単化されているが)、解析的解 (closed-form solution) が存在しなくなり、たちまち分析することが困難になる。そのときに、私たちが取ることが出来る手段は2つある。第1に紙と鉛筆で解けるまで、モデルを簡略化する事である。センスが良い理論家であれば、現実世界を驚くほど簡単化しながらもエッセンスだけは損なわない見事な分析を披露してくれる。第2の手段は、手で解く事を諦めて、コンピュータを使って近似的にモデルの性質を理解しようというアプローチである。

---

<sup>3</sup> 著者たちは決して紙と鉛筆のアプローチを批判しているわけではない。実際、本連載で紹介していくモデルはしっかりとした理論的基礎があり、その基礎を作ってきたのはミクロ経済学を中心とした経済理論家たちである。

後者は前者ほどエレガントではないかもしれないが、複雑なモデルを出来る限りそのまま分析する事を可能にしてくれる。ただし、そのためには数値計算という新しい分析手法を学ばないといけない。従来の経済学教育 (特に大学院レベルのコースワーク) では、線形代数や微分、最適化理論を用いるマクロ経済学・ミクロ経済学と統計学に基づく計量経済学を習得する必要があったが、そこに数値計算というこれまでの経済学で培ってきた手法とは若干毛色が違う方法が加えられる事になったのである。

数値計算は、モデルの定性的性質だけでなく定量的性質も表現出来る事から、実証研究との相性が良いこともあり、このアプローチは相乗効果をもたらすことになった。モデルを数値的に解けば、当然そこから得られる解は定性的 (上がるか下がるか) な情報だけではなく、どの程度上昇するかという定量的側面を持つ。もしモデルが現実のある側面をうまく描写しているのであれば、そこから得られた政策的含意はこれまでよりも強い定量的メッセージを持つことになる。すなわち、「○○という政策」によって「産出量が上昇する」という結果だけではなく、「 $\Delta$  % の GDP の上昇をもたらす」というより現実に即した政策的含意を得られるのである。

実際、多くの論文はモデルのベンチマーク (基準となる状態) を現実経済のある時点にマッチするようにパラメータを設定して (これをカリブレーションと呼ぶ)、そこから政策分析を行うことによって、これまで以上に豊かな政策的含意を得ることに成功している。また、モデルが現実をうまく描写していないとしても、どの部分にどの程度の乖離があるのかも定量的に測ることが出来る。そのため、現在のマクロ経済学は完成にはほど遠いかもしれないが、データと照らし合わせながら理論の検証を繰り返すことが可能であるため、我々の経済への理解を着実に深めていくことが可能になる。

### 3 経済学における数値計算の位置付け

経済学における数値計算アプローチが経済学者の道具箱の中に仲間入りしてかなりの年月が経過しているが、数値計算の用いられ方も研究者によって様々である。これまで、政策指向が強い定量的マクロ経済学の視点から話をしてきたが、例えば連載の中で紹介する予定の動的計画法 (**dynamic programming**) の研究を行っている Bona and Santos (1997) は、数値計算を「理論モデルの理解を深めるための装置」とみなしている。

It is worth emphasizing that the numerical model is not generally meant to directly emulate an economic situation. This model is simply a supporting device (an algo-

rithm of some sort) aimed at the simulation of the true behavior of the mathematical model. One should have a good idea of how well the behavior and stability properties of the numerical model mimic those of the mathematical model under consideration.

一方で、同じ動的計画法を家計や企業の行動を記述するために利用して、構造推定 (**structural estimation**) を行うことによって、公的年金制度や医療制度と行った現実的な問題の解決に用いようという研究も数多く存在している<sup>4</sup>。

この違いを数値計算に対するスタンスの違い、あるいは研究手法に関する哲学の違いとみなすことも出来るが、もっとシンプルに自分の研究目的にあった便利なツールの領域が拡大してきていると考えるべきだろう。

## 4 数値計算を学ぶ

これまで説明をしてきたとおり、定量的マクロ経済学と数値計算は現在のマクロ経済分析では欠かせない分析ツールの一つとなっているが、幸 (既に分析ツールを身につけた人にとって、複雑な分析ツールの習得は参入コストになるためアカデミックな競争相手が減る) か不幸かその敷居は決して低くはない<sup>5</sup>。大学院レベルのマクロ経済学を学習しようとする、従来の上級マクロ経済学及び計量経済学 (時系列分析やミクロ計量分析) に加えて、数値計算の手法も習得しないといけないのである。

すでに数値計算は経済学の中でも確立された手法で、その歴史も 30 年以上は経過している。経済学の専門的研究を掲載しているトップジャーナルにも数えきれない程、大量の論文がすでに掲載されている。そのため、現在では数値計算の経済学への応用を扱った書籍もいくつか出版されているが、全てが英語である<sup>6</sup>。

上級マクロ経済学のテキストも定評のあるものは英文が主流であることを考えれば、本来、数値計算の本を英語で読むことが特別ハードルが高いわけではないはずである。しかし、残念ながら日本における経済学教育の中で、数値計算分野は遅れていると著者たちは考えている。

より正確に言うと、ニューケインジアン DSGE モデルのような研究が盛んでしっかり教育がで

---

<sup>4</sup> 例えば、French (2005) を参照。

<sup>5</sup> “Unfortunately for Ph.D students and fortunately for those who have worked with DSGE models for a long time, the barriers to entry into the DSGE literature are quite high. (Fernández-Villaverde et al. (2016))”

<sup>6</sup> 理系分野であれば、優れた数値計算のテキストや訳書は存在している。例えば、前出のクアルテローニ他 (2014) は大学院のコースワークレベルの数学を苦にしない読者であれば、読み通せるはずである。ただし、扱っている事例が毛細血管のネットワークといった経済学を学んでいる人からはピンとこないものばかりになってしまうという問題がある。

きる (=自分でそのツールを使って論文を書いている) 分野がある一方で、そうではない本連載で扱う予定の分野については、教育する人も少ない事から手薄となっているのである。もし手薄なのがニッチでマイナーな分野であれば、それもある程度は仕方がないことと思われる。しかし、本連載が扱う範囲は現在のマクロ経済学のメインストリームの一つといっても過言ではなく、ミクロ計量モデルなどと接続して労働経済学や社会保障分野に波及したり、より“ハイテク”なニューケインジアン DSGE モデルにも展開されている。連載を通じて、現代のマクロ経済分析に必要な新たな道具を身につけるサポートをすることが我々の目的である。

具体的には、動的計画法と射影法 (**projection method**) という動学的マクロ経済分析を行う際によく使われる 2 種類の数値計算手法について、実際に手を動かしながら説明をしていく。どちらも、家計や企業の意思決定問題を解く際に必要になる、ベルマン方程式 (**Bellman equation**) あるいはオイラー方程式 (**Euler equation**) を近似的に解く手法である。

連載の後半では、それまでに学んだ数値計算手法を応用する事によって、どのような経済問題を実際に分析できるのかを紹介していく。上級レベルのマクロ経済学を学ぶとまず初めに出てくるのが、ラムゼイモデルあるいは最適成長モデルと呼ばれる個人の消費・貯蓄に関する意思決定問題を解くモデルである。ラムゼイモデルでは、代表的個人 (**representative agent**) の異時点間の消費と貯蓄に関する意思決定問題を解いて、そこからマクロ経済の動態を探っていく。実際、RBC モデルやニューケインジアン DSGE モデルの根幹をなしているのも、この代表的個人の意思決定問題である。

ここでよくある批判が代表的個人、すなわち一人の経済主体の意思決定行動に単純化しても良いのかという問いである。代表的個人の仮定に基づいて、すでに様々な有益な分析結果を得ていることを考えると、代表的個人は現実的ではないから無用であるという批判は的はずれであろう。しかし、代表的個人の仮定を置いてしまうと扱えない経済問題もたしかに存在している。

その最たるものが経済格差問題である。いうまでもないが経済にロビンソン・クルーソー一人しかいないのであれば、格差問題や再分配政策は考えようがない。しかし、様々な側面の異質性を持つ経済主体が混在する経済を考えようとする、途端にモデルが複雑過ぎて解けなくなってしまうという問題がある。

そこで力を発揮するのが数値計算である。本連載の特徴の一つは、代表的個人モデルから離れて、異質な個人 (**heterogeneous agent**) モデルを取り扱う事にある。異質な個人モデルを学ぶことによって、これまでとは違った側面からマクロ経済の現実問題に切り込む力を得ることが出来る。

## 5 プログラミング言語

[R]eading a book on computational techniques without actually using the computer is as foolish as reading a cook-book without ever entering the kitchen. (Marimon and Scott (1999), page 2.)

本連載では、具体的な計算手法やアルゴリズムについて紙面で説明をしていく。しかし、マクロ経済学・ミクロ経済学を学習する際には計算結果を逐一自分で確認しながら理解をしていくように、数値計算もコンピュータ上で自分で手を動かしてみないと身につかない。

実際に多くの読者が関心があるのは、例えば、「財政・金融政策が GDP に与えるインパクトの大きさ」といった数値結果を自力で計算することであろう。そのためには、コンピュータとソフトウェアを自分で用意して (あるいは学校のコンピュータを利用して)、手を動かして試行錯誤をしながら学んでいく必要がある。プログラミングはまさに「習うより慣れよ」の世界である。

ここに数値計算を学習する際のハードルがある。数値計算を実際に学ぶために何を準備すればよいのだろうか？エクセルなどの表計算ソフトでは力不足なので、自力で**プログラミング (programming)** が出来る環境を揃えないといけない。高額なソフトウェアが必要になるのではないかという不安を持つ人も多いのかも知れない。

確かに、専門家が使う数値計算用のソフトウェアやライブラリと呼ばれる便利なツールの中には数 10 万円するものもある。幸いなことに、数値計算のソサイエティ (あるいはプログラマのソサイエティ) の中には、ソフトウェアはタダであるべきだという信念・哲学の元で、ボランティアで開発を進めて公開している人達がいる<sup>7</sup>。フリーだから商用ソフトウェアより劣っているとは必ずしも限らない。例えば、最近流行している (機械学習や深層学習を用いる) 人工知能の分野では、Python という言語がよく用いられているが、Python 本体と機械学習に用いられる TensorFlow のようなツールは基本的にフリーである。また、学生向けに安価なバージョンが提供されていたり、大学に設置された PC に既にインストールされているプログラム言語もある。まずは、自分が数値計算を学習するための環境を整えよう。

本連載では、主に商用ソフトウェアである Matlab を使用するが、必要に応じて Fortran や Python といった言語も用いる予定である。また、連載中に教育目的で使ったソースコードについては適宜 Github などで公開していく。是非、読者自身で数値計算結果を再現したり、実際に手

---

<sup>7</sup> 彼らの活動に共感した人は、開発を援助するために自発的に募金をすることも出来るし、自分がある程度ソフトウェアを使いこなすことが出来るようになったら、開発のお手伝いをする事も出来る。

を動かしてみしてほしい。既に何らかのプログラミング言語に習熟しているのであれば、必要が出てこない限り、そのまま“お気に入り”の言語を使っていけば良いが、まったくの未経験で何から手をつけたら良いかわからないという読者のためにいくつかお勧めのプログラミング言語を紹介しよう<sup>8</sup>。

「そんなマニアックな論文があるのか」と思うかも知れないが、Aruoba and Fernández-Villaverde (2015) は、同じ確率的新古典派成長モデル (パイの生地!) を同じ手法で解くことによってプログラミング言語毎の計算速度の比較を行っている。表 1 は、その結果の一部を抜粋したもので、Time と書かれた列が実際に計算にかかった時間である。実際の速度はコードを動かす PC (新しいか古いか、ノート PC かデスクトップか等) によって変わってくるが、重要なのはその隣の相対的な速さ (Rel. Time) である。見ての通り、同じモデルを同じ手法で解いた場合でも、数倍から数百倍の差があることに驚かされる。

一番早い言語を身につければよいのかというと、そこまで話は単純ではない。表が示すとおり、C++ や Fortran といった言語は計算速度で大きなアドバンテージがある<sup>9</sup>。一方で、C++ はここで挙げられている言語の中でもっとも学習難易度が高く、プログラムを書くのにも手間がかかる。数値計算を実際に行う際には、一瞬で計算結果が得られる場合もあれば、一晩ずっとコンピュータを動かし続けるという複雑なモデルもある。後者であればより高速な言語を学習する必要性は高いが、初学者がいきなり複雑なモデルにチャレンジするのは無謀だし、むしろ経済学と関係のないプログラミング言語の習得で挫折してしまい、本末転倒になりかねない。

C++ や Fortran を利用してみたいという人は中・上級者であると考えられるので、ここでは割愛し、初心者向けのプログラミング言語を簡単に紹介していこう。

■Matlab 前述の通り、本連載では主に MathWorks 社が開発した Matlab という言語を用いる。学習の容易さやユーザインターフェイスの使いやすさなど、有料ではあるが、その分だけ完成度が高いソフトウェアである。日本だけでなく世界中で多くの経済学者が利用しているため、様々なコードをネットから入手しやすいといった利便性も高い。安価な学生版も提供されている。Matlab とほぼ同等の機能をフリーで利用できる Scilab や Octave も有用である。ただし、これらは Matlab と完全互換を保証しているわけではない。

---

<sup>8</sup> 紙面の都合から、特定の言語のインストール方法やプログラム言語の解説は一切しない。

<sup>9</sup> 著者たちが学術論文で大型モデルを解く場合には、Fortran などを使用することが多い。

表 1: 数値計算速度の比較: Aruoba and Fernández-Villaverde (2018)

	Version/Compiler	Time	Rel. Time
C++	GCC-7.3.0	1.60	1.00
	Intel C++ 18.0.2	1.67	1.04
Fortran	GCC-7.3.0	1.61	1.01
	Intel Fortran 18.0.2	1.74	1.09
Julia	0.7.0	2.35	1.47
Matlab	2018a	4.80	3.00
Python	CPython 3.6.4	166.75	104.22
R	3.4.3	57.06	35.66
Matlab, Mex	2018a	2.01	1.26
Python	Numba 0.37.9	2.31	1.44
	Cython	2.13	1.33

ここから入手

- Matlab: <https://jp.mathworks.com/products/matlab.html>
- Scilab: <https://www.scilab.org/>
- Octave: <https://www.gnu.org/software/octave/>

■Python フリーですべての開発環境を揃えたいのであれば、現在流行中の Python を使うのもお勧めである。ただし、Python は Matlab とは異なり、数値計算やシミュレーション分析に特化した言語ではない。Web デザインなどにも使える汎用性がある一方で、例えば逆行列を計算するといった機能はデフォルトで備わっておらず、必要に応じてライブラリを追加しないといけない。

数値計算目的であれば、Python に科学技術演算用のライブラリ、データ分析ツールなどがセットになった Anaconda(あるいは軽量版の Miniconda) をインストールすれば、必要はものは一通り揃う。また、実際にコードを書く際にはテキストエディタを使っても良いが、PyCharm や Visual Studio Code といったソフトウェア (こちらもフリー) をインストールすると、Matlab のように快適に数値計算を行う開発環境が整備出来る。なお、Thomas Sargent 教授と John Stachurski 教授が Web 上で数値計算講義を行っている QuantEcon は Python をメインに使っている。



ここから入手

- Python(公式): <https://www.python.org/>
- Anaconda: <https://www.anaconda.com/>
- Pycharm: <https://www.jetbrains.com/pycharm/>
- Visual Studio Code: <https://azure.microsoft.com/ja-jp/products/visual-studio-code/>
- QuantEcon: <https://lectures.quantecon.org/>

■R/Julia 統計分析を行った経験がある読者であれば、Rを使ったことがあるかも知れない。表1の通り、Rは計算速度の面で他の言語に劣るが、入門レベルであればまったく問題ないので、Rの使い心地が気に入っているのであれば、ひとまずRを使うことも可能である。

Julia という言語は聞き馴染みがないかも知れないが、科学技術演算をターゲットとした新しい言語である。後発なので開発環境や便利なツールといった面ではまだまだな部分もあるが、表1のように計算速度に優れており、学習難易度もMatlabやPythonと同程度であるため、将来性があると考えられる。

どちらも、RStudio、JuliaPro という必要な開発環境が一通り揃ったセット (もちろんフリー) があるので、初心者はそちらを利用するのがお勧めである。

ここから入手

- R: <https://www.r-project.org/>
- Julia: <https://julialang.org/>
- RStudio: <https://www.rstudio.com/>
- JuliaPro: <https://juliacomputing.com/products/juliapro.html>

## 6 おわりに

Matlabを除けば、上で紹介した全てのソフトウェアはフリーなので、まずは試しに色々とインストールをしてみてほしい。気に入らなければアンインストールをしまえば良いのである。繰り返しになるが、プログラミングは「習うより慣れよ」なので、自分の手を動かして試しながら学習していくのが上達の近道である。次回は、シンプルな2期間モデルを用いながら数値計算の勘所について説明をしていく。

## 参考文献

- Aruoba, S. Boragan and Jesus Fernández-Villaverde (2015) "A Comparison of Programming Languages in Economics," *Journal of Economic Dynamics and Control*, Vol. 58, pp. 265-273.
- (2018) "A Comparison of Programming Languages in Economics: An Update," Mimeograph.
- Bona, Jerry L. and Manuel S. Santos (1997) "On the Role of Computation in Economic Theory," *Journal of Economic Theory*, Vol. 72, No. 2, pp. 241-281.
- Fernández-Villaverde, Jesús, Juan F. Rubio-Ramírez, and Frank Schorfheide (2016) "Solution and Estimation Methods for DSGE Models," NBER Working Paper No. 21862.
- French, Eric (2005) "The Effects of Health, Wealth, and Wages on Labour Supply and Retirement Behavior," *Review of Economic Studies*, Vol. 72, No. 2, pp. 395-427.
- Gabaix, Xavier (2016) "A Behavioral New Keynesian Model," NBER Working Paper No. 22954.
- Kydland, Finn E. and Edward C. Prescott (1982) "Time to Build and Aggregate Fluctuations," *Econometrica*, Vol. 50, No. 6, pp. 1345-1370.
- Lucas, Robert E. (1976) "Econometric Policy Evaluation: A Critique," *Carnegie-Rochester Conference Series on Public Policy*, Vol. 1, pp. 19-46.
- Marimon, Ramon and Andrew Scott eds. (1999) *Computational Methods for the Study of Dynamic Economies*: Oxford University Press.
- 加藤涼 (2007) 『現代マクロ経済学講義動学的一般均衡モデル入門』, 東洋経済新報社.
- クアルテローニ, A.・F. サレリ・P. ジェルヴァシオ (2014) 『MATLAB と Octave による科学技術計算』, 丸善出版.