はじめに

対象

この講義ノートでは，時間を通じて変化するマクロ経済の様子を分析するための考え方を紹介する。マクロ経済学という分野を整理して理解する上では，「経済成長」と「経済変動」という 2 つの視点から経済環境を眺めることが有用である。すなわち，次のように考える：マクロ経済（＝国の経済全体）は，長期的には経済環境が好転していくという一定の傾向（トレンド）に従っているように見える一方で，短期の視野で見れば景気の浮き沈みが人々の暮らし向きを大きく左右している。長期的な「経済成長」と，短期的な「経済変動」の 2 つを合成することで，現実経済の複雑な振る舞いの理解に近づこうというのがマクロ経済分析の基本方針だと言える。

成長と変動のいずれに注目して分析するにしても，「時間」という要素が欠かせないことがすぐに分かるだろう。時間の止まった世界では，成長も景気の浮沈も起こりようがないからだ。したがって，マクロ経済学の重要な結論をきちんと理解するためには，時間を通じた変化を扱うための数学的・経済学的なモデルを構築・分析する手法を学ぶ必要がある。

時間を扱うための数学は微分方程式や差分方程式である。本講義では，経済変数が離散的な時間軸上で定義されていると考える差分方程式モデルを主に用いる。どちらかが正解とか簡単ということではないが，念のため選択を正当化しておこう。現実の経済がどのような振る舞いをするかについてのシンプルな理解を得たいというのが私たちの目標なのだけれども，複雑な経済現象をうまく表現するために巧妙な数理モデルを操作した結果，難解な数学的表現をまとった答えから先に進めなくなってしまう，ということ

がしばしば起こる。そのようなときには，コンピュータ・シミュレーションを用いたモデルの定量的な評価や可視化が大いに役に立つ。そして，シミュレーションの観点から言えば，離散時間のモデルは数学的な表現とコンピュータのコードとの対応関係が見つけやすいという意味でやや易しいと言えるかもしれない。これが本書で離散時間のモデルを用いる理由である。それ以上のものではない。

この講義で扱う内容は次の 3 つのポイントに集約される。

- 時間を考慮したマクロ経済モデル（数理モデル）を構築する方法
- 構築した数理モデルを解く方法
- 解いた結果をコンピュータで可視化する方法

この一連の操作に習熟すれば，マクロ経済モデルの結果を解釈・評価することができるようになる。

マクロ経済学が長期の成長と短期の変動の二本柱からなると言っても，実は最近のマクロ経済モデルの多くは，Ramsey モデルや世代重複モデルといった成長モデルを基礎に組み立てられている。これらのモデルは，単純な 2 期間の最適消費モデル（Fisher の異時点間の最適消費モデル）と大きくは変わらないので，恐れずにチャレンジしてほしい。

想定読者

想定する読者像は，標準的な学部中級レベルのミクロ経済学・マクロ経済学を履修済みの学部学生・大学院生である。中級のマクロ経済学を復習しながら，シミュレーションの手法を学んで，より上級のマクロ経済学に進むための橋渡しをすることを目的としている。

マネジメント学類開講科目のうち，

- マクロ経済学入門，マクロ経済学 I，マクロ経済学 II
- ミクロ経済学 I，ミクロ経済学 II

で扱う内容を習得していれば十分であろう。自習をするとすれば，以下の 3 冊を一通り読んでおくとよい。

- ハル・ヴァリアン（佐藤隆三・監訳）『入門 ミクロ経済学』勁草書房
- N・グレゴリー・マンキュー（足立英之他・訳）『マクロ経済学 I 入門編』東洋経済新報社
- N・グレゴリー・マンキュー（足立英之他・訳）『マクロ経済学 II 応用編』東洋経済新報社

高校数学や大学初年次で学ぶ微分積分学や線形代数学を超える数学を使うことはほとんどないはずだが，必要になった場合には詳しく説明することを心がけるので，根気よく式変形を追ってほしい。不安な読者は，制約条件付き最適化問題を解くためのラグランジュ未定乗数法などといった応用数学のテクニックを先に身につけておくと，スムーズに読むことができる。

特徴

この講義では読者が中級レベルまでのミクロ経済学およびマクロ経済学を履修済みであることを想定し，その知識の土台の上に，より高度な分析手法やコンピュータ・シミュレーションの技法を身につけることを目標としている。シミュレーションに用いる Python 3 の使用経験は問わないという方針のもと，講義で扱うトピックの順序は経済学的に自然であることをある程度は意識しつつも多くの例外が発生する。プログラミングのテクニックという観点から難易度が徐々に難しくなるように配慮したというのがその理由である。もし経済学に関する説明が直線的でないことに困難を感じる読者は，まず，上で紹介したミクロ・マクロの教科書をじっくり読むことをすすめる。

各章は，概要・理論・プログラミングの 3 つの部分から成る。理論，プログラミングについては複数の節にまたがることがある。

- 「概要」には，各回で理解してもらいたいトピックを列挙している。他の章との対応関係はこの部分に書くようにするで，基礎知識が足りないと感じた場合は，

概要に記載されているパートを読んでみてほしい。この講義ノートは前から順番に読むようには書かれていないので，自由に動き回ってもらってよい。章の全体を一通り読み終えたら，概要に書かれている内容を自分で説明できるようになっているかを時間をとって考えてみてほしい。

- 「理論」には，数学的・経済学的な分析手法の紹介を書く。数学が苦手な読者は躓いても気にせずに読み進めて，プログラミングの分析まで進んでほしい。
- 「プログラミング」には，「理論」で紹介した分析をコンピュータ上で再現・確認するための方法を書いている。すべてのコードを読者が自分で打ち込んで実行することが期待されている。冗談かと思うかもしれないが，**プログラミングは指の筋肉を使って学ぶもの**である。相当な上級者になるまでは，コードを読むだけで理解しようなどとは考えないほうがいい。数字を変えたり，条件を変えたりしつつコードを実行しながら学ぶように心がけてほしい。

Python 3 の経験があるか他のプログラミング言語にかなり習熟しているが経済学の経験が少ない読者にとっては，標準的な経済学の教程に従った順序で読む方がスムーズに読み進められるかもしれない。そのように感じた読者は，「概要」を読んで，先にどの章を読むべきかを判断してほしい。

コンピュータ・シミュレーションの原理はプログラミング言語によらず共通のものであるが，特定の言語を固定することなく，この共通原理について説明することは大変に困難な仕事である。したがって，この講義では多くの講義と同様に 1 つの言語（すなわち，Python 3）を固定することにした。

プログラミング言語に Python を選んだのに特別な理由はない。しいて言えば，Python はフリーで人気があるプログラミング言語だからだ。Python にこだわらなくても，以下の機能を持つ言語であればなんでもよい。もっと言えば，1 つの言語に限定しなくても複数の環境を組み合わせることもできる。

- ベクトル計算を実行できること。
- 線形代数計算ができること。
- データ可視化が容易にできること。

- 関数の最大化・最小化をする方法を備えていること。
- 代数的に微分ができること。

Python 本体ではこのようなことはできないが，**NumPy**,**SciPy**,**Matplotlib**,**SymPy** というサードパーティ製のライブラリを用いて実現できる。**Pandas** という人気のあるデータ処理用のライブラリを用いれば，生データ，シミュレーション結果の処理を効率的に実行できる。本書では，**Pandas** の使い方についても簡単に紹介する。

構成

本講義ノートは 15 のセッションからなる講義のために書かれている。講義の構成は下表の通りである。

開発環境

この書籍のコードは，表 2 に示される環境で確認されている。必ずしもバージョンをすべて揃える必要はないが，バージョンが異なることで生じる問題は原則的に読者の責任で解決していただきたい。

表 1: 講義の構成

講	理論	プログラミング
1	複利計算と成長	Python を関数電卓として使う
2	物価指数とインフレーション	ベクトルの表現と演算，簡単な可視化
3	GDP の成長と格差	繰り返し計算，制御構文
4	時系列データのモデリングと行列計算	線形システムのシミュレーションと AR モデル
5	成長と変動	Pandas による時系列データ取得，可視化
6	ソロー・モデルと成長会計	ユーザー定義関数，非線形システムのシミュレーション
7	2 期間最適消費モデル	SymPy による代数計算と数値最適化
8	ラムゼー・モデル (1)	非線形システムの線形化。フォワードルッキング・モデルのシミュレーション (1)
9	ラムゼー・モデル (2)	ダイナミック・プログラミング
10	リアル・ビジネスサイクル・モデル	フォワードルッキング・モデルのシミュレーション (2)
11	動学的 AD-AS モデル	
12	ニュー・ケインジアン・フィリップス曲線	フォワードルッキング・モデルのシミュレーション (3)
13	ニュー・ケインジアン・モデル	フォワードルッキング・モデルのシミュレーション (4)

表 2: Python およびライブラリのバージョン

	Version
Python	3.8.5
NumPy	1.19.2
SciPy	1.5.2
SymPy	1.6.2
Pandas	1.1.3
Matplotlib	3.3.2
statsmodels	0.12.0
pandas-datareader	0.9.0

謝辞

本文中で紹介した Python およびサードパーティライブラリ（numpy, scipy, matplotlib, pandas）の他にも多くのオープンソースプロジェクトのお世話になった。LaTeX 2$_\varepsilon$ を用いたタイプセットに LyX を使用し，Python コードと実行結果を本文中に埋め込むために R(R Core Team, 2019),**knitr**(Xie, 2015, 2019) および **reticulate**(Ushey et al., 2019) を用いた。Python, R, LaTeX を取り巻くコミュニティに感謝する。

参考文献

R Core Team (2019) *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, URL: `https://www.R-project.org/`.

Ushey, Kevin, JJ Allaire, and Yuan Tang (2019) *reticulate: Interface to 'Python'*, URL: `https://CRAN.R-project.org/package=reticulate`, R package version 1.12.

Xie, Yihui (2015) *Dynamic Documents with R and knitr*, Boca Raton, Florida: Chapman and Hall/CRC, 2nd edition, URL: `https://yihui.name/knitr/`, ISBN 978-1498716963.

——— (2019) *knitr: A General-Purpose Package for Dynamic Report Generation in R*, URL: `https://yihui.name/knitr/`, R package version 1.23.