

【研究ノート】

動的計画法による 確率的経済成長モデルの数値解析

井上 智夫*

1 動機

動学的マクロ経済モデルの複雑化に伴って、近年ではこれらを数値解析する手法の理解と解析環境の整備が不可欠になっている。特に、異質な経済主体におけるミクロレベルの意思決定問題がマクロレベルの市場均衡に与える影響を分析するには、この種の解析手法の利用が必須である。そこで本稿では、もっとも基本的な動学マクロ経済モデルである「確率的成長モデル」を材料として、数値解析の手順やシミュレーションの方法などを整理する。Extended Path 法や Parameterized Expectations 法などさまざまな数値解析方法が存在するが、ここではその中でも広汎な問題に応用可能な Value Function Iteration 法（価値関数反復法）とその派生である Modified Policy Function Iteration 法（修正政策関数反復法）についてのみ扱う。¹ また、汎用プログラミング言語 Python で解析用プログラムを作成し、計算所用時間や計算精度などを比較検討する。

2 基本モデルの構造

無限期間生存する計画者が、消費 C から得る期待効用の現在割引価値の総和（生涯効用）を最大化する問題に直面していると仮定する。

$$\max_{\{C_0, C_1, \dots\}} U_0 = E_0 \left[\sum_{t=0}^{\infty} \beta^t u(C_t) \right] \quad (1)$$

$$\text{s.t. } C_t \geq 0, K_t \geq 0 \quad (2)$$

$$C_t + (K_{t+1} - \psi K_t) = Z_t f(K_t) \quad (3)$$

* 本稿は、成蹊大学長期研修制度（2015-2016年度）の研究課題のひとつ「動学マクロモデルの計量分析手法の研究」の成果の一部であり、国際基督教大学 社会科学研究所（研修受入先機関）に客員研究員として滞在中に作成した。本研究は成蹊大学研究助成(B)（2012-2014年度）の助成を一部受けている。本研究は、本学経済学部教授 中神康博氏およびシンガポール国立大学教授 清水千弘氏との共同研究、また国際基督教大学教授 海蔵寺大成氏との共同研究がベースになっている。また、数値演算への取組み方について、本学理工学部教授 池上敦子氏から貴重な助言を頂いた。記して感謝したい。本稿の数値演算とグラフ作成に利用した汎用プログラミング言語 Python に関して、井上裕太氏から多大なる協力を得たのでここに謝辞を表す。あり得べき誤りはすべて著者に属する。

¹ 確率的経済成長モデルを異なる解析方法で比較検討した論文集については、*Journal of Business & Economic Statistics*（1990年1月号の特集）を参照。また本稿で扱うモデルの原典である Heer and Mausnner（2009）にも解析方法が整理されている。

ここで、 $E_0[\]$ は 0 時点の情報に基づいた条件付き期待値オペレータ、 β は割引因子で $\beta \in (0, 1)$ 、 ψ は $\psi \in (0, 1)$ で資本減耗率 δ をもちいれば $\psi = 1 - \delta$ であり、 C_t は消費財の t 時点における消費、 K_t は t 期期首時点での資本ストックである。効用関数には異時点間の分離可能性と $u' > 0$, $u'' < 0$ を、また生産関数は $f' > 0$, $f'' < 0$ を仮定する。生産性 Z_t は外生的な確率変数で、このシステムにおける唯一のかく乱要因である。

計画者は、(1)式で示す生涯効用を最大化するような C の将来流列 $\{C_t\}_{t=0}^{\infty}$ を、(2)式の消費・資本ストックにかかわる非負条件と(3)式の資源制約条件のもとで選択する問題に直面している。ただし、 K_0 は所与とする。

3 ベルマン方程式

今期の消費を C 、今期期首の資本ストックを K 、今期の生産性を Z とする。来期の値についてはプライムをつけて K' 、 Z' とする。ここで今期の状態変数を K 、 Z 、来期の状態変数を K' 、 Z' とすると、式(1),(2),(3)で定義した最大化問題に対応するベルマン方程式は次のようになる。

$$V(K, Z) = \max_{0 \leq K' \leq Zf(K) + \psi K} u(C) + \beta E[V(K', Z')|Z] \quad (4)$$

消費 C は資源制約式から

$$C = Zf(K) + \psi K - K'$$

となるので、上記のベルマン方程式は次のように状態変数のみで表記できる。

$$V(K, Z) = \max_{0 \leq K' \leq Zf(K) + \psi K} u(Zf(K) + \psi K - K') + \beta E[V(K', Z')|Z] \quad (5)$$

ここで $V(K, Z)$ は価値関数 (Value Function) とよばれる。以下では、離散状態空間法 (Discrete State-Space 法、または Finite State Approximation 法。以下では DSS 法と略記) をもちいて最適化問題の解として価値関数を数値的に求める。DSS 法では、効用関数と生産関数については通常この種の分析でもちいられる CRRA 型やコブ=ダグラス型のパラメトリックな関数形を想定するが、価値関数はノンパラメトリック的に推定する。また最適化問題の解として、最適な消費行動や資本蓄積行動をあらわす政策関数 (Policy Function) $C(K, Z)$ と $K'(K, Z)$ も推定する。

4 基本モデルの数値解

マクロ経済学の動学的確率モデル分析において、確率的経済成長モデルはもっとも基本的なモデルであり、その数値解析の例は既に多数存在する (*Journal of Business & Economic Statistics* 1990 年 1 月号の特集など)。それらは効用関数や生産関数に利用するパラメトリック

クモデルの選択や、割引因子・資本減耗率などのパラメータの値の設定などの点で異なっている。特にパラメータの値は現実の経済変数を参考に決められるため、分析対象国によっては異なった設定になりうる。本稿の主たる目的は数値計算法そのものの特徴の把握であって現実経済の分析は主たる関心事ではないため、本節と次節では Heer and Maussner (2009) の 4 章で紹介された確率的経済成長モデルの設定をもとにする。

4.1 事前の設定

4.1.1 関数形の設定

数値解析のため、本稿も先行研究にならって効用関数には相対的危険回避度一定 (Constant Relative Risk Aversion。以下では CRRA と略記) 型を想定する。

$$u(C) = \frac{C^{1-\eta} - 1}{1-\eta}, \quad \eta > 0$$

この関数 $u(C)$ は $u' > 0, u'' < 0$ を満たす。ここで η は Arrow-Pratt の相対的リスク回避度の指標である。他方、生産関数にはコブ=ダグラス型を想定する。

$$f(K) = K^\alpha, \quad \alpha \in (0, 1)$$

この関数も生産関数が満たすべき性質 $f' > 0, f'' < 0$ を満たす。

4.1.2 パラメータの値の設定

生産関数や効用関数のパラメータ、資本減耗率や割引因子の値については表 1 のように設定する。資本減耗率 δ が 0.011 なので、 $\psi = 0.989$ である。

表 1 パラメータの設定

変数名	記号	値
生産の資本に対する弾性値	α	0.27
割引因子	β	0.994
危険回避度	η	2.0
資本減耗率	δ	0.011

表 1 のもとで得た定常状態における資本ストックの均衡値の解析解は、最適化の一階条件 $K' = \alpha\beta ZK^\alpha$ から、 $Z = 1$ のもとで次式で与えられる。

$$K^* = \left[\frac{1 - \beta\psi}{\alpha\beta} \right]^{\frac{1}{\alpha-1}}$$

K^* が得られれば、 Y^* と C^* についても順次以下の式で求めることができる。

$$Y^* = (K^*)^\alpha, \quad C^* = Y^* - \delta K^*$$

これらの値は表2の通り。

表2 定常状態における内生変数の均衡値

変数名	記号	値
資本ストック	K^*	44.0375
生産量	Y^*	2.7786
消費量	C^*	2.2942

4.1.3 外生的状態変数 Z の離散化について

生産性ショック Z はこのシステムにとって唯一の外生的なかく乱要因であり、その動学的な挙動は次の1次自己回帰モデル(AR(1))で表現できるものとする。

$$\ln Z_t = \rho \ln Z_{t-1} + \epsilon_t \quad \epsilon_t \sim N(0, \sigma_\epsilon^2)$$

ここで、誤差項 ϵ_t は平均ゼロ、分散が σ_ϵ^2 の正規分布に従うと仮定する。² 本稿では、 $\rho = 0.90$ 、 $\sigma_\epsilon = 0.05$ を仮定する。

生産性 Z' は(4)式や(5)式の右辺の条件付き期待値の一部、次期の価値関数 $V(K', Z')$ 、に含まれている。DSS法では、価値関数はノンパラメトリック的に推計する。よって、 Z' の実現値が連続型のままでは扱いにくい。離散化によって近似を行う。非線形動学マクロモデルの数値解析では、AR過程に従う外生的な状態変数の挙動を、状態数が有限なマルコフ連鎖で近似する方法がとられる。代表的な手法は Tauchen(1986)および Tauchen and Hussey(1991)であり、本稿でも Tauchen(1986)の方法を利用する。

Tauchen(1986)は、外生的状態変数が存在する空間を m 個のグリッド

$$\{\ln Z_1, \ln Z_2, \dots, \ln Z_m\}$$

をもちいて $m-1$ 区間に等間隔で分割する。³ ここで $\ln Z_1$ は $\ln Z$ が現実的にとりうる区間の下限値、 $\ln Z_m$ は区間全体の上限値で、これらは以下のように決定する。

$$\ln Z_m = -\ln Z_1 = \Omega \sigma_z$$

なお、 Ω は正の実数、 σ_z は離散化の対象である連続型AR(1)過程の条件なし標準偏差である。⁴

² 生産性水準の平均値 $E(Z)$ を1に標準化しているため、 $\ln Z$ のAR(1)の定数項を0にしている。

³ ここでは Z に対数正規性を想定する。その結果、 $\ln Z$ は左右対称な正規分布に従う。

⁴ 誤差項の分散が σ_ϵ^2 でAR(1)の係数が ρ なので、 $\ln Z$ の(条件なし)標準偏差 σ_z は $\sigma_z = \sqrt{\sigma_\epsilon^2 / (1 - \rho^2)}$ 。 $\rho = 0.90$ 、 $\sigma_\epsilon = 0.05$ のとき、 $\sigma_z = 0.1147$ になる。

Ω の選択については, Tauchen(1986)は $\Omega = 3$, また Flodén(2008)は $\Omega = 1.2 \ln(m)$ としている。本稿では Heer and Maussner(2009)にならって $\Omega = 4.5$ とした。⁵ したがって $\ln Z$ の上下限値は $\pm 4.5 \times 0.1147$, 対応する Z の上下限値は $Z_1 = 0.5968$ と $Z_m = 1.6756$ になる。

現在時点の状態をインデックス i で, 次期の状態をインデックス j であらわす。このとき, $i = 1, \dots, N$ から j へのマルコフ連鎖の遷移確率は, 標準正規分布の確率分布関数 Φ をもちいて次のようにあらわすことができる。

$$p_{i,j} = \begin{cases} \Phi\left(\frac{Z_1 - \rho Z_i + h/2}{\sigma_\epsilon}\right), & j = 1 \text{ の場合} \\ \Phi\left(\frac{Z_j - \rho Z_i + h/2}{\sigma_\epsilon}\right) - \Phi\left(\frac{Z_j - \rho Z_i - h/2}{\sigma_\epsilon}\right), & j = 2, \dots, m-1 \text{ の場合} \\ \Phi\left(\frac{Z_m - \rho Z_i + h/2}{\sigma_\epsilon}\right), & j = m \text{ の場合} \end{cases}$$

ここで h はグリッド点のステップ幅であり

$$h = \frac{\Omega \sigma_z}{m-1}$$

なので, Ω の設定に依存することになる。

ベルマン方程式を DSS 法で数値的に解く際には, 生産性水準の対数値 $\ln Z$ ではなく生産性水準 Z をもちいるので, ここで外生的状態変数 Z がとりうる離散値の集合を次のように定義する。

$$\mathcal{Z} = \{Z_1, Z_2, \dots, Z_m\}$$

本稿では $m = 7, 15, 31$ の 3 通りを検証する。例えば, $m = 7$ の場合の生産性 Z のグリッドの値は以下ようになる。

$$\mathcal{Z} = [0.5968 \quad 0.7088 \quad 0.8419 \quad 1.0000 \quad 1.1878 \quad 1.4108 \quad 1.6756] \quad (6)$$

もともと $\ln Z$ について等間隔なグリッドを設定したため, 対数の逆変換を行った Z のグリッド幅は等間隔にはなっていない。また, 各グリッド間の状態遷移確率は次のようになる。

$$(p_{i,j}) = \begin{bmatrix} 0.7544 & 0.2456 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0080 & 0.8410 & 0.1509 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0195 & 0.8962 & 0.0843 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0427 & 0.9147 & 0.0427 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.0843 & 0.8962 & 0.0195 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.1509 & 0.8410 & 0.0080 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.2456 & 0.7544 \end{bmatrix} \quad (7)$$

⁵ Heer and Maussner (2009) は 4.2 節の数値例では $\Omega = 5.5$ に, 4.3.1 節の例では $\Omega = 4.5$ に設定している。

正の確率は行列の対角線上近辺に集中していることから、来期の状態は今期の状態のままか、あるいは遷移する場合でも隣接する状態に限定されることになる。本稿の数値例では $\rho = 0.9$ を想定しているが、既存研究のなかには ρ がより 1 に近い値をとるものもある。そのような場合、状態遷移確率はさらに対角線上に集中することになる。

本稿では、 m について 3 種類のグリッド ($m = 7, 15, 31$) を設定して、計算結果を推定精度や計算所用時間などの観点から比較する。図 1 は、それらの遷移確率行列を图示したものである。グラフからは明確ではないが、生産性の状態数 m を増やした場合であっても、状態の変遷は隣接する状態にかなり偏っていることが伺われる。⁶

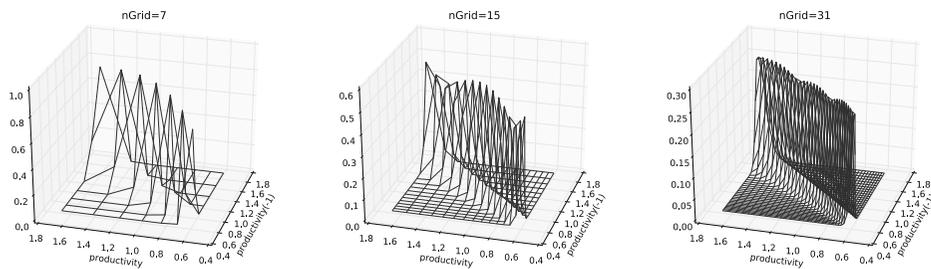


図 1 遷移確率の状態

注) $\rho = 0.9, \sigma_\epsilon = 0.05, \Omega = 4.5$ で算出。(PlotTauchenAR.py)

4.1.4 内生的状態変数 K の離散化について

最後に、価値関数 V に含まれるもう一つの要因、資本ストック K についても離散化を行う。その際に重要なことは、下限値と上限値の設定である。外生的な状態変数である生産性 Z の場合は、経験則から得た Ω の値と AR(1) モデルの誤差分散から、 Z が取りうる下限値と上限値を設定した。一方、内生的状態変数の資本ストックについては、外生的に与えられる生産

⁶ 外生的状態変数 Z の動学過程の離散化については近年さまざまな研究が報告されている。持続性が高い AR 過程 (あるいは固有値が 1 に近い値をとる過程) については Tauchen (1986) 法が効果的でないことが報告されている (Tauchen, 1986, Tauchen and Hussey, 1991, および Flodén, 2008)。もちろん、持続性が高い AR 過程についても状態変数の離散値の個数を多くすることで近似精度を高めることができるが、この対応では外生状態数の増加に伴い実行性が乏しくなってしまう。このような場合、状態変数の離散値を増やさずに近似精度を高める方法としては Rouwenhorst (1995) や Adda and Cooper (2003) がある。Kopecky and Suen (2010) は Tauchen から Adda and Cooper までの 5 種類の方法を数値的に相互比較した結果として Rouwenhorst 法の優位性を報告している。本稿で扱う外生的状態変数は生産性 Z のみだが、モデルによっては状態変数が複数になる場合もある。Gospodinov and Lkhagvasuren (2013) は複数の外生状態変数の相互依存等の挙動を VAR で表現し、これを離散化する方法として Rouwenhorst 法を発展させた Moment-matching 法を提案している。また複数の方法を比較した論文としては Scheuring and Jonen (2013) がある。Lkhagvasuren (2012) は Rouwenhorst 法の主要なモーメントの計算式を提示している。最近の動向については Farmer and Toda (2015) を参照。

性の実現可能な上限値と下限値をもとに設定することとする。具体的には、生産性の下限値（あるいは上限値）の状態が持続した場合の資本ストック水準を、最適化の一階条件

$$1 = \beta (1 - \delta + Z_j f'(K'_j))$$

から計算し、この値を K の下限値 K_1 と上限値 K_n とする。ここでグリッド数を n 個とすると、資本ストックのグリッドの集合は $K = \{K_1, K_2, \dots, K_n\}$ になる。

表 3 K の上下限値の設定

	生産性 Z の上下限値	資本ストック K の上下限値
下限値	0.5968	21.7136
上限値	1.6756	89.3128

基本モデルの数値解析では、4 種類のグリッド数 ($n = 200, 1000, 5000, 25000$) について比較検討する。なお、上下限値は常に固定する。また基本モデルの分析においては、計算の途中で粗いグリッドから細かなグリッドへ移行させることはせずにグリッド数は固定し、その状況下での計算所要時間や計算精度を比較することとした。

4.2 モデルの解法：Value Function Iteration

DSS 法の計算手法にはいくつかのバリエーションが存在するが、まずは基本的な手法である Value Function Iteration（価値関数反復法）で計算を行う。手順は以下の通り。

1. 資本ストックと生産性について、それぞれ離散値の集合 $K = \{K_1, K_2, \dots, K_n\}$ と $Z = \{Z_1, Z_2, \dots, Z_m\}$ を設定。
2. 価値関数行列 V^0 の初期値を設定し、 $V^\ell = V^0$ とする。（ここでは計算速度の単純比較のために、ゼロ行列を初期値とした）
3. (K_i, Z_j) のすべての組合せ ($i = 1, \dots, n; j = 1, \dots, m$) に対して写像 U を計算する。

$$V_{i,j}^{\ell+1} = \max_{K^+} u \left[Z_j f(K_i) + \psi K_i - K^+ \right] + \beta \sum_{l=1}^{N_Z} p_{jl} V^\ell(K^+, Z^l) \equiv UV_{i,j}^\ell$$

4. 収束の判定を行う。 $\sup \|V^\ell - V^{\ell+1}\| < 10^{-6}$ ならば収束したと判断する。
5. 収束するまで、手順 3 を繰り返す。

なお、政策関数の単調性を利用して K^+ の探索範囲を限定することで、効率的な計算を行った。

4.3 価値関数と政策関数の数値解

本稿の計算はすべて汎用プログラミング言語 Python をもちいた。⁷ 計算過程では適宜、Numpy や Scipy を利用し、描画ライブラリとして Matplotlib 中の Pyplot をもちいた。また計算を高速化するために Numba も利用した。⁸

表 4 状態変数のグリッド数設定と計算時間

生産性Zの グリッド (数)	資本ストックKの グリッド (数)	収束までの 更新回数	計算時間 (秒)
31	25,000	2,196	451.2676
31	5,000	2,196	97.1198
31	1,000	2,196	16.1473
31	200	2,196	3.5027
15	25,000	2,196	161.2325
15	5,000	2,196	36.1708
15	1,000	2,196	7.1734
15	200	2,196	1.9050
7	25,000	2,196	76.3602
7	5,000	2,196	16.8794
7	1,000	2,196	3.4512
7	200	2,196	1.0626

注) 計算に使用した Macbook Pro のスペックは、プロセッサが 2.4 GHz Intel Core i5, メモリが 8 GB 1333 MHz DDR3, グラフィックスが Intel HD Graphics 3000 512 MB。計算時間は複数の試行の平均値ではないので、あくまで参考数値。(RBC_PyNumba_benchmark.py)

生産性 Z については 3 種類のグリッド数 (7 個, 15 個, 31 個の 3 種類), 資本ストック K については 4 種類のグリッド数 (200 個, 1000 個, 5000 個, 25000 個の 4 種類) を想定し、計 12 通りの組合せについて計算し、計算所用時間や精度を比較する。

表 4 は、Value Function Iteration での計算結果をグリッド数の組合せごとにまとめたものである。比較のため、価値関数行列の初期値は全要素をゼロにした。収束までの更新回数はグリッド数の設定にかかわらずすべて 2196 回だが、当然のことながらグリッドが細かなほど収束までの所要時間が長くなる。もっとも粗い設定 ($m = 7, n = 200$) では収束まで 1 秒しかかからないが、もっとも細かな設定 ($m = 31, n = 25000$) では 7 分半を要している。

図 2 は、生産性に 31 グリッド、資本ストックに 25000 グリッドを設定して計算した価値関数と政策関数である。図は横軸が生産性 Z 、縦軸が資本ストック K で、これらの状態変数の

⁷ コーディングに際して、McKinney (2013), 石本 (2014), Python サポーターズ (2013) などを参考にした。

⁸ Aruoba and Fernández-Villaverde (2014) は、確率的新古典派経済成長モデルの均衡解を価値関数反復法をもちいて求めており、異なるプログラム言語、具体的には C++, Fortran, Java, Julia, Python, Matlab, Mathematica および R について、計算所用時間の比較を行っている。

もとでの達成可能な価値関数の値を左図に、また最適な消費額 C の水準を右図に、それぞれ等高線をもちいてあらわしている。生産性 Z が高いほど（水平方向に右側になるほど）実現可能な生産量が増え、また同様に資本ストック水準が高い場合（垂直方向に上側に位置する場合）ほど実現可能な生産量が増えるため、価値関数と達成可能な最適消費額は増加しており、その様子が図からも明らかである。

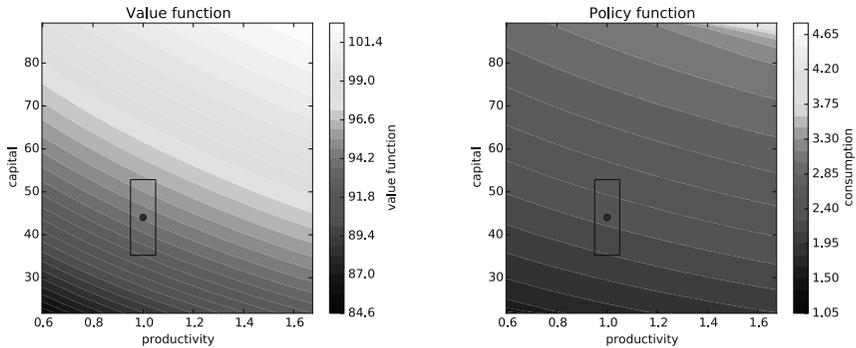


図 2 生産性が 31 グリッド、資本ストックが 25,000 グリッドの場合

注) 図は横軸が生産性 (Z)、縦軸が資本ストック (K) で、これらの状態変数のもとでの同時点の価値関数の値 (左図) と、最適な消費額 (C) の水準 (右図) を等高線であらわしたもの。生産性が高いほど（水平方向に右側になるほど）、また資本ストック水準が高いほど（垂直方向に上側になるほど）、価値関数と達成可能な最適消費額が増加するさまがわかる。図中の点は生産性と資本ストックの定常状態における均衡値 ($Z^* = 1.0, K^* = 44.03$)。それを囲む四角の領域は生産性と資本ストックがそれぞれ $Z^* \pm 5\%$ と $K^* \pm 20\%$ の範囲。以下では計算精度をこの領域に限定して吟味する。(PlotContour.py)

動的的マクロモデルの数値分析では、擬似的に内生変数をシミュレートしてその分布特性を調べ、モデルの妥当性を検討することがある。そこで、上述の数値例をもちいてシミュレーションを実施してみよう。最初に、外生的状態変数の生産性水準 Z を次式を使って生成する。

$$\ln Z_{t+1} = \rho \ln Z_t + \sigma_e e_{t+1}, \quad \text{ただし } e_t \sim i.i.d.N(0, 1) \quad (8)$$

ここで $\rho = 0.9, \sigma_e = 0.05$ に設定する。⁹生成した Z を所与とし、生産量、消費量、および資本ストックを生成する。なお、次期資本ストックは、数値的にもとめた資本ストックの政策関数で決定した。また、以下での比較を容易にするために、初期時点 (0 時点) の状態変数の

⁹ なお、このシステムでの唯一のかく乱要因は Z なので、相互に比較が容易になるように、以下で提示するシミュレーションの数値例ではすべて同一の乱数をもちいている。

値については $K_0 = K^*$, $Z_0 = 1$ で統一する。シミュレートしたデータの例を図3に示す。なお、図中の点線は定常状態における均衡値である。

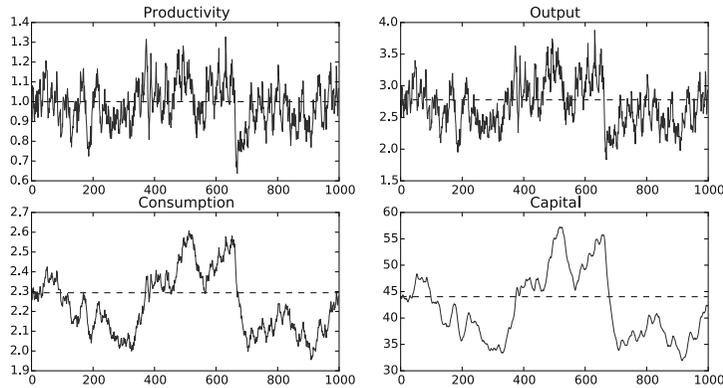


図3 基本モデルのもとでシミュレートした 1,000 個のデータの分布

注) 図中の点線は定常状態における均衡値で、生産量(Y^*)は 2.7786、資本ストック(K^*)は 44.0375、消費量(C^*)は 2.2942 である。生産性水準(Z^*)については 1 を想定する。ここでは均衡値をシミュレーションの初期値に設定した。(SimEndogVarsLine.py)

4.4 計算精度の測定

グリッド数を細かくしても計算精度に大差がないならば、不必要なレベルまで細かなグリッドを設定するべきではないだろう。では、グリッド数の設定は価値関数と政策関数の数値解にどの程度の影響を与えるのであろうか。

計算精度の比較には、一般的に、Euler Equation Residual (以下、EER と略記) が利用されている。EER は、オイラー方程式

$$C^{-\eta} = \beta E \left\{ \left[(C')^{-\eta} \left(1 - \delta + \alpha (e^{\rho \ln Z + \sigma \epsilon'}) (K')^{\alpha-1} \right) \right] \middle| Z \right\}$$

の両辺の消費 C , C' から誤差 $C'/C - 1$ を定義し、ある範囲の消費に関してこの誤算の平均値や最大値を計算し評価指標とするものである。図2の点は、生産性と資本ストックの定常状態における均衡値 ($Z^* = 1.0$, $K^* = 44.03$) であり、それを囲む四角の領域は生産性と資本ストックがそれぞれ $[0.95Z^*, 1.05Z^*]$ と $[0.8K^*, 1.2K^*]$ の範囲を示している。以下では、この領域について EER を計算し、計算精度の評価指標とする。なお本稿では、オイラー方程式の右辺の条件付き期待値の評価にあたり、ガウス・エルミート法とシミュレーションの2種類の方法を比較した。

第一の方法、ガウス・エルミート法は、来期の生産性 Z' の分布をガウス・エルミート求積

法 (Gauss-Hermite quadrature) を利用して数値積分で評価する方法である。EER の測定範囲は、図 2 の枠で囲った定常均衡値の周辺とする。この四角い領域を、縦横 200×200 のグリッドに細分化し、その各点、総数 4 万地点において EER を評価し、その最大値を表 5 の EER (1) の列にまとめた。

EER の値は、最大値がもっとも粗いグリッドでの値 1.9198×10^{-1} 、最小値がもっとも細かいグリッドでの値 1.3269×10^{-3} で、100 倍程度の格差がある。ただしこの差は、生産性グリッド数よりも資本ストックのグリッド数の設定による影響の方が大きいようである。またその効果も $n=200$ から $n=1000$ への増加時が最大で、その後は逓減的である。

第二の方法は、シミュレーションを利用する方法である。この方法は、4.3 節で紹介した数値シミュレーションをもちいて Y, C, K の系列を長期間 (例えば 10 万期間) 生成し、この値をもちいて EER を計算する方法である。ガウス・エルミート法と同様に、EER の最大値を表 5 の EER (2) の列にまとめた。評価領域は、 $(Z, K) = [0.95Z^*, 1.05Z^*] \times [0.8K^*, 1.2K^*]$ の領域である。資本ストックのグリッド数と最大 EER との関係は、ガウス・エルミート法で確認したものとほぼ同じ傾向になっている。

表 5 Euler Equation Residuals の計算結果

生産性グリッド(数)	資本ストックグリッド(数)	EER(1)	EER(2)	(比率)
31	25,000	1.3269e-03	3.5284e-02	(28.38)
31	5,000	5.7468e-03	3.5353e-02	(27.77)
31	1,000	2.3816e-02	5.5232e-02	(23.43)
31	200	1.0319e-01	1.6565e-01	(0.45)
15	25,000	1.2753e-03	3.4430e-02	(28.39)
15	5,000	7.7105e-03	3.6538e-02	(27.78)
15	1,000	2.7457e-02	5.3650e-02	(23.49)
15	200	9.4985e-02	1.9124e-01	(0.40)
7	25,000	1.7163e-03	4.6311e-02	(28.87)
7	5,000	7.0931e-03	4.8652e-02	(28.18)
7	1,000	3.3469e-02	5.4116e-02	(23.59)
7	200	1.9198e-01	1.2359e-01	(0.67)

注) 最右列の比率は、生成した 10 万個の乱数のうちで Euler Residuals の評価領域に該当する値の比率(%)であり、ERR (2) はこれらから計算している。(EulerResiduals.py)

推計精度にかかわる評価については、今回の数値例では結果的にガウス・エルミート法とシミュレーションにもとづく方法のどちらでも大差は確認できなかった。しかしながら、評価領域を均等に総当たりで評価するガウス・エルミート法と比べて、シミュレーションは生成する乱数に影響される点は注意すべきだろう。

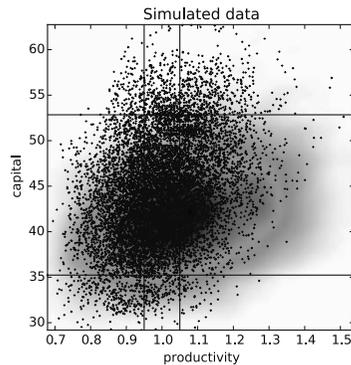


図4 シミュレートした 10,000 個のデータの分布

注) 図中の線は生産性が $0.95Z^*$ と $1.05Z^*$ 、資本ストックが $0.8K^*$ と $1.2K^*$ に対応している。ただし、生産性と資本ストックの定常状態における均衡値は $Z^* = 1.0$, $K^* = 44.03$ 。以下では計算精度をこの領域に限定して吟味する。なお、シミュレーションの初期値には K^* を使用。(SimEndogVarsScat.py)

図4は1万個のシミュレーションデータの分布を図示したものである。散布図は、横軸が生産性、縦軸が資本ストック、点は生成されたデータである。この図では生成した1万個のデータの最小値と最大値をもとに散布図の横軸と縦軸の範囲を決めている。したがって、1万件のデータを生成した際でも、この範囲外の値をとるデータは生成されなかったことを意味している。

表6 シミュレートした 10,000 個のデータのエリアごとの分布状況 (単位: %)

1.41	3.20	4.66
29.39	30.65	25.66
2.96	1.45	0.62

注) 図4の散布図において、領域を9分割した際、各領域に所属するデータの割合を計算したもの。(SimEndogVarsScat.py)

散布図中の縦線2本と横線2本は、EERの評価領域を示している。このことから明らかなのは、生成されたデータの一部しかEERの評価領域に入っていないということである。表6は、図中の線で9分割された領域ごとに位置するデータの個数から計算した分布状況である。表の中央の30.65%がEERの評価領域 $(Z, K) = [0.95Z^*, 1.05Z^*] \times [0.8K^*, 1.2K^*]$ に落ちたデータであり、よってこの方法ではシミュレートされたデータのたかだか3割程度しか評価に利用されていないことになる。

表5の最右列にある“比率”は、EERの評価のために生成したシミュレーションデータ10

万個のうち、評価領域に入ったデータの比率である。資本ストックのグリッド数が 5000 以上の場合はおおむね 3 割弱程度の水準になっている。他方、グリッド数が 200 のときは 1% にも満たないため、この方法での推定精度の評価は適していないことになる。

最後に、1 万個のシミュレーションデータから描いた生産性と資本ストックの分布（図 5）についても補足したい。本モデルにおける根源的なく乱要素は、生産性が確率変数であることにある。生産性の動学は(8)式で定義した AR(1)モデルに従うと仮定しているが、ベルマン方程式を Value Function Iteration で解く場合や、政策関数もちいて数値シミュレーションするには、 $\ln Z$ ではなく Z をもちいている。そのため、生産性の分布は若干右に裾が長い非対称な分布になる。しかしながら、図中に示したように平均値とメディアンがともに 1 になっており、 $E(Z) = 1$ の想定に合致した結果になっている。また資本ストックについては、その平均値が 44.5 でこれも定常状態の均衡値 $K^* = 44.03$ にきわめて近い値をとっている。

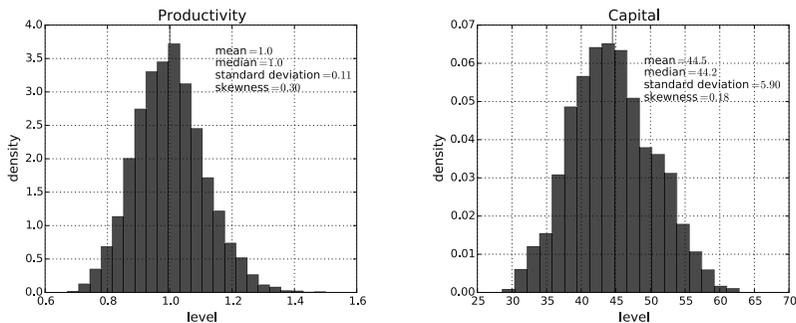


図 5 シミュレートした 10,000 個のデータの分布

注) 図 4 の作図にもちいたシミュレーションデータをもとに作成。(SimEndogVarsScat.py)

本稿の数値解析では、生産性と資本ストックのグリッドの上下限値を、それぞれ $[0.5968, 1.6756]$ と $[21.7136, 89.3128]$ としている（表 3 を参照）。このヒストグラムから、これらの上下限値が Z と K が取得しうる領域を適切にカバーしていることが確認できる。むしろ資本ストックの上限値については若干過大なレベルまでカバーしているため、領域を狭めれば僅かながら計算コストの低減に寄与するかも知れない。

5 基本モデルに投資の非負制約を追加した場合

次に、確率的経済成長モデルに投資の非負制約条件を加えたケースを考慮してみることとする。¹⁰

¹⁰ なお、本節の数値例も Heer and Maussner (2009) を参考にしている。

5.1 追加的な制約条件の影響

ここでは2節の最大化問題を次のように変更する。

$$\begin{aligned} \max_{\{C_0, C_1, \dots\}} U_0 &= E_0 \left[\sum_{t=0}^{\infty} \beta^t u(C_t) \right] \\ \text{s.t. } C_t &\geq 0, \quad K_t \geq 0 \\ C_t + (K_{t+1} - \psi K_t) &= Z_t f(K_t) \\ K_{t+1} - \psi K_t &\geq 0 \end{aligned}$$

これにともない、ベルマン方程式も次のように変更される。

$$V(K, Z) = \max_{\psi K \leq K' \leq Zf(K) + \psi K} u(Zf(K) + \psi K - K') + \beta E[V(K', Z')|Z]$$

基本モデルからの唯一の変更点は、 K' の下限値が 0 ではなく ψK になった点である。 t 期期首の資本ストック K は資本減耗により $t+1$ 期期首には ψK になる。ここで t 期中にプラスの投資を実行すれば $t+1$ 期期首の資本ストック K' は ψK よりも大きな値になる。他方で、 t 期中に新たな投資を実行しなければ $K' = \psi K$ 、生産性の落ち込みが原因で消費の平滑化のために資本ストックの一部を消費にまわすような事態が発生すると $K' < \psi K$ になってしまう。基本モデルではこれらの可能性を排除しなかったが、本節では投資に非負制約を課すこととする。この制約に伴い、生産性が落ち込む場合には消費の平滑化が実現できない事態が発生する。

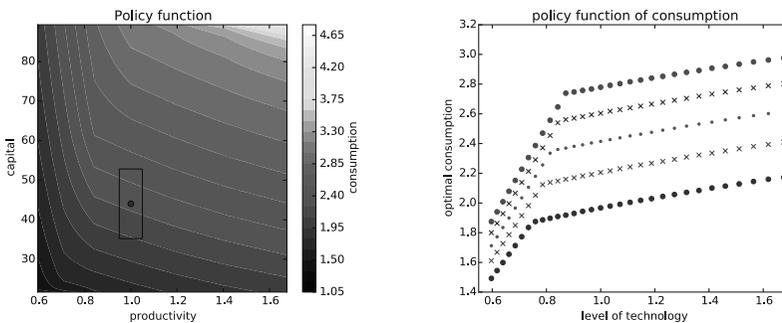


図6 投資に非負制約を課したもとの消費の政策関数の計算結果 ($m = 31, n = 25000$ の場合)

注) 左図は横軸が生産性 (Z)、縦軸が資本ストック (K) で、これらの状態変数のもとでの同時点での最適な消費額 (C) の水準が等高線であらわしたもの。生産性が高いほど (水平方向に右側になるほど)、また資本ストック水準が高いほど (垂直方向に上側になるほど)、達成可能な最適消費額が増加するさまがわかる。また、右図は、5通りの資本ストック水準について、実現した生産性と達成可能な消費額との関係を示した図。左図の鳥瞰図を、 $K = 30, 40, 50, 60, 70$ において輪切りの断面図に相当。生産性水準が低い領域では、消費額が減少しているさまがみえる。(PlotContour.py と PlotPolicy2d.py)

図6は投資に非負制約を課した場合の消費の政策関数の推計結果である。この推計にも Value Function Iteration をもちいた。図から、消費の政策関数平面が屈曲する箇所が図の左側に等高線の尾根として確認できる。図2と比較すると、尾根の右側の領域では図2と図6に差がなく、よって消費額は非負制約の影響を受けていないことが確認できる。他方で尾根の左側の領域では、非負制約下での消費額は制約無しの場合に比べ明らかに減少している。

5.2 基本モデルとのシミュレーションデータの比較

実現した生産性水準がきわめて低かった場合、達成可能な生産量が定常状態の均衡値を大きく下回るため、最適な消費水準を維持するためには資本ストックの一部を消費にまわさなくてはならない。しかし、投資の非負性がこの可能性を排除するため、結果として消費自体が減少することになるはずである。この状況をシミュレーションデータで確認してみよう。

図7は投資の非負制約下での数値シミュレーションの結果である。なお、前節の数値例との比較を容易にするため、図3を作成したときと同じ生産性ショックの乱数をもちいている。¹¹ 一見では図7と図3の消費の系列が同一のように見える。これは、非負制約の影響を受けていない状態では消費行動に差が生じないためである。図4の散布図からも類推できるように、実際、非負制約条件下で生成したデータのほとんどは制約を受けていない。

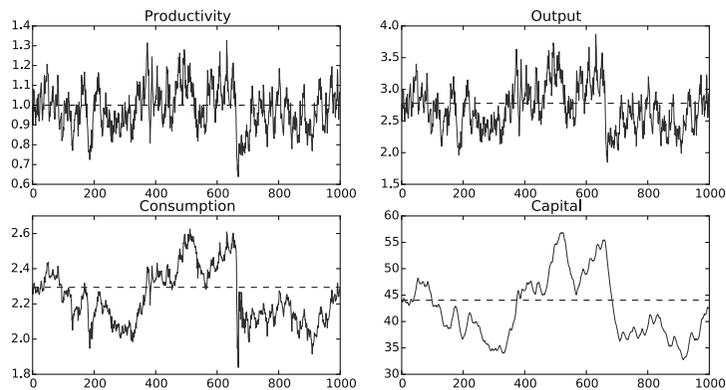


図7 投資の非負制約条件のもとでのシミュレーション結果 ($m = 31, n = 25000$ の場合)

注) 図中の点線は定常状態における均衡値で、生産量(Y^*)は2.7786、資本ストック(K^*)は44.0375、消費量(C^*)は2.2942である。生産性水準(Z^*)については1を想定する。ここでは均衡値をシミュレーションの初期値に設定した。(SimEndogVarsLine.py)

しかし、実現した生産性 Z の値がきわめて低かった場合には消費が制約を受けている。具

¹¹ したがって、図3と図7の Productivity (左上のパネル) は同じ系列。

体的にはシミュレーションしたデータの 190 番目と 670 番目付近の 2 カ所で、実現した生産性が 0.7 近辺の低い値をとっており、これら 2 時点の生産と消費が大きく落ち込んでいることが図 7 からわかる。

5.3 資本ストックのグリッド数 n の選択と計算時間との関係

本節のモデルは、2 節で提示した基本モデルに投資の非負制約を一行追加したにすぎない。しかし、4.2 節で解説した Value Function Iteration に非負制約を追加するだけの修正では、計算に必要な時間が膨大になり実用向きではなかった。

表 4 に要約されるこれまでの検討結果から、1) 資本ストックのグリッド数 n の増加は更新回数に影響しないこと¹²、および 2) 計算所用時間は資本ストックのグリッド数 n の増加に伴って $O(n)$ のオーダーで増加すること、が判明している。したがって、1) 価値関数行列の初期値の設定が真の値に近いほど収束までに要する更新処理は少ない回数で済むこと、また、2) n の値が小さいほど計算は短時間で済むこと、の 2 点が予想できる。

表 7 生産性のグリッド数と追加的な計算所要時間 (秒)

n	$m = 31$		$m = 7$		K のグリッド幅
	更新回数	計算時間(秒)	更新回数	計算時間(秒)	
200	2199	5.304	2199	1.869	0.3380
1,000	1130	42.115	1194	10.202	0.0676
5,000	643	511.268	714	126.611	0.0135
25,000	364	19535.590	364	1594.152	0.0027

注) K の最小値と最大値は [21.7136, 89.3128]。表の最右列の K のグリッド幅は、 K の範囲を n で割った値。計算に使用した Macbook Pro のスペックは、プロセッサが 2.4 GHz Intel Core i5、メモリが 8 GB 1333 MHz DDR3、グラフィックスが Intel HD Graphics 3000 512 MB。計算時間は複数回の試行にもとづく値ではないので、あくまで参考数値。(RBC_PyNumba_nonNegInvest_Iterative.py)

この予想をふまえて計算方法を次のように変更した。まず最初に $n = 200$ の粗いグリッドで反復計算し、いったん価値関数と政策関数の値を取得する。表 7 にまとめたように、 $n = 200$ の場合は $m = 31$ でも 5.304 秒で収束している。次に、 $n = 200$ で得た価値関数行列の値を $n = 1000$ の状態空間で線形補間 (あるいは多項式補間) し、これを $n = 1000$ の反復計算の初期値とする。価値関数行列としての精度は低いものの、ゼロ行列よりは有用なので、この工夫によって (価値関数の初期値がゼロ行列の場合であれば) 収束まで 2199 回を要した (n, m)

¹² n の値にかかわらず更新回数が同じなのは、すべてのケースにおいて価値関数行列の初期値がゼロ行列であるためである。

= (1000, 31) の更新回数¹³が、約半分の 1130 回にまで減少、追加的な計算所用時間も 42.115 秒に留まっている (表 7 を参照)。同様の処置をより細かなグリッド ($n = 5000, 25000$) へ順次適応した結果、それぞれの更新回数は 643 回と 364 回にまで減少した。

しかしながら、 $n = 5000$ と $n = 25000$ のときの追加的な計算時間はそれぞれ 511.268 秒と約 5 時間半 (19535.590 秒) となり、特に $n = 25000$ の場合は未だに実用的とはいえない結果になった。¹³

5.4 状態変数のグリッド数の選択と計算精度との関係

制約条件下での最大化問題は頻繁に直面する問題であるため、この状況で計算時間を短縮できなければ実用的な計算方法とは言えない。そこで、一案としてグリッド数削減の可能性を検討した。表 7 の右側は生産性に粗いグリッド ($m = 7$) を想定した場合の計算時間である。生産性グリッドを $m = 31$ から $m = 7$ へ減らせば、 $n = 25000$ の場合でも全行程を 30 分弱で計算できる。

では、生産性グリッドの設定は非負条件下での計算精度や政策関数の数値解、内生変数のシミュレーションにどの程度の意味を持つのであろうか。まず計算精度についてみてみよう。表 8 に Euler Equation Residuals の最大値をまとめた。表 5 の計算結果と比較するため、ここでの評価領域は定常均衡解の近傍で図 6 の枠内とした。したがって今回の数値例では、評価領域には非負制約の影響で消費が制約を受けている領域は含めていない。表 5 と表 8 の比較からは、同じグリッド数の組合せでは両者に明確な違いは見当たらない結果になった。

表 8 Euler Equation Residuals の計算結果

生産性グリッド (数) m	資本ストックグリッド (数) n	EER
31	25,000	1.2445e-03
31	5,000	6.1351e-03
31	1,000	2.1169e-02
31	200	1.0328e-01
7	25,000	2.9680e-03
7	5,000	7.6106e-03
7	1,000	3.6531e-02
7	200	2.0477e-01

注) EERはガウス・エルミート法にもとづいて計算。評価領域には前掲の表と同じ範囲を利用。
(EulerResiduals.py)

次に政策関数の形状についてはどうだろうか。図 8 は、 $m = 31$ と $m = 7$ の場合の消費の政

¹³ 図 7 は $m = 31, n = 25000$ で計算した結果を利用。

策関数の断面図である。いずれも $n = 25000$ で計算した。当然だが、生産性グリッド数が多い方が、非負制約による屈曲点を正確に捕捉できている。図8の左図 ($m = 31$ の場合) では、消費の政策関数の屈曲点は、生産性水準が低い場合には約 0.75, 高い場合には約 0.85 になっており、屈曲点も明確である。他方、右図 ($m = 7$ の場合) では、屈曲点はかならずしも明確ではない。しかし今回の例では、グリッドが 0.7088 と 0.8419 にあり、偶然にも屈曲点と合致していたため粗めのグリッドではあるが $m = 7$ でも屈曲点の位置は大筋で捉えられているようでもある ((6)式を参照)。

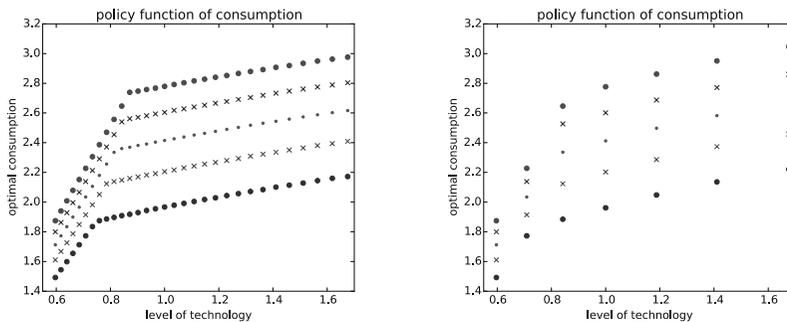


図8 投資に非負制約を課したもとの消費の政策関数の計算結果 ($n = 25000$)

注) 左図は生産性のグリッド数が 31 ($m = 31$) の場合、右図はグリッド数が 7 ($m = 7$) の場合。両図とも 5 通りの資本ストック水準 ($K = 30, 40, 50, 60, 70$) において、実現した生産性と達成可能な消費額との関係を图示。(PlotPolicy2d.py)

最後に、シミュレートした内生変数の動向をみてみよう。 $m = 7, n = 5000$ についても図7と同様のグラフを作図したが生成した内生変数に見た目上の差は存在しなかった。そこで、制約を課さない状態の基本モデルでシミュレートした消費に対して非負制約下の消費がどの程度乖離するかを算出し、これを同時点の生産性水準についてプロットした。シミュレートしたデータは 1000 個。図9の左図は $m = 31, n = 25000$ の場合、右図は $m = 7, n = 5000$ の場合で、両図とも縦軸は非負制約下での消費から基本モデルの消費を差引いた値、横軸は生産性 Z である。基本モデルの数値は $m = 31, n = 25000$ で計算した。

図9の左図からは、 $Z < 0.85$ の領域では制約下の消費は無制約の場合に比べて明かに少なく、また生産性が低いほどその傾向が顕著になっている。シミュレートしたデータ数 (1000 個) に比べてこの領域に存在するデータが限定的なことは、5.2 節で述べた特徴を裏付けるものである。実際に非負制約を受けるほど生産性が大きく落ち込んだのは 1000 個の生産性ショック Z のシミュレーションデータのうちでも 2 回程度であり、これらの点はその影響を受けたものとみられる。一方、生産性水準が $Z > 0.85$ の領域では、非負制約下の消費は制約

なし状態と同一水準（ゼロの場合）あるいは若干多め（プラスの場合）に消費している。詳細な確認はしていないが、生産性の落ち込みに伴い、いったん最適な消費経路から乖離した後、調整期間を経て最適経路へ回帰するまでの僅かな過剰消費がプラスの乖離に現れているものと思われる。

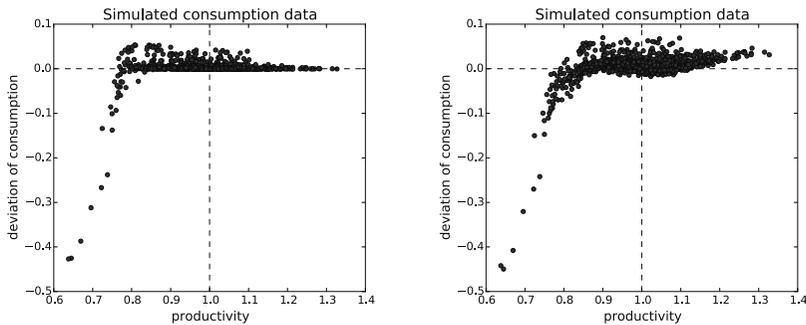


図9 シミュレートした消費データの比較（生産性との関係）

注) シミュレートしたデータ数は 1000 個。左図はグリッド数が $m = 31, n = 25000$ の場合、右図はグリッド数が $m = 7, n = 5000$ の場合。両図とも横軸は生産性 Z 。縦軸の値は、非負制約下での消費から基本モデルの消費を差引いた値。生産性が低い領域では、 m の値にかかわらず、消費の落ち込みが激しい様子が明らか。その一方で、左図では $Z > 0.85$ の領域では消費の差は非負であるのに対し、右図では Z がゼロ近辺で微小ながらマイナスになっている。(SimConsScat.py)

これに対して、図9の右図は状態変数のグリッドを粗くした場合 ($m = 7$ の場合) である。 $m = 31$ に比べて、誤差の幅が大きいだけでなく、屈曲点も不明瞭になっている。さらには生産性水準が高い領域では上振れする傾向がみられる。これはシミュレーションの際に、政策関数を 3 次多項式補間することに起因している可能性がある。これらの結果は、グリッド数の設定方法だけでなく、グリッド間の補間方法についても慎重に選択する必要があることを示唆している。

5.5 モデルの解法の変更：Modified Policy Function Iteration

最後に、さらなる計算時間の短縮のため、Value Function Iteration にかかわって Modified Policy Function Iteration（修正政策関数反復法）の利用を検討する。Modified Policy Function Iteration の計算手順は次の通りである。

1. 資本ストックと生産性について、それぞれ離散値の集合 $K = \{K_1, K_2, \dots, K_n\}$ と $Z = \{Z_1, Z_2, \dots, Z_m\}$ を設定。

2. V^0 の初期値を設定し, $V^\ell = V^0$ とする。
3. (K_i, Z_j) のすべての組合せ ($i = 1, \dots, n; j = 1, \dots, m$) に対して次のように定義された写像 U を計算し, 同時に資本ストックの政策関数 $U_{i,j}^{\ell+1}$ を計算する。

$$U_{i,j}^{\ell+1} = \max_{K^+} u \left[Z_j f(K_i) + \psi K_i - K^+ \right] + \beta \sum_{l=1}^{N_z} p_{jl} V^\ell(K^+, Z^l) \equiv U V_{i,j}^\ell$$

4. $W^0 = V^\ell$ とし, (K_i, Z_j) のすべての組合せについて, $s = 1, \dots, S$ 回, 次の計算を実行する。(本稿では $S = 30$ とする)

$$W^{s+1}(K_i, Z_j) = u \left[Z_j f(K_i) + \psi K_i - U_{i,j}^{\ell+1} \right] + \beta \sum_{l=1}^{N_z} p_{jl} W^s(U_{i,j}^{\ell+1}, Z^l)$$

5. 収束の判定を行う。 $\sup \|V^\ell - V^{\ell+1}\| < 10^{-6}$ ならば収束したと判断する。
6. 収束するまで, 手順 3 と 4 を繰り返す。

計算所用時間の結果は表 9 の通り。計算では, 5.3 節で有用性を確認した「段階的にグリッド数を増やす方法」を採用した。

Modified Policy Function Iteration の有効性は, 今回の数値解析で最も細かなグリッドである $m = 31, n = 25000$ の場合でも, 価値関数行列の初期値がゼロ行列の状態から全行程が 4 分弱で計算が完了したことからも明確である。同様の有効性は, 生産性グリッドを $m = 15$ に減少した場合でも認められた。その一方で, $m = 7$ の場合は, $n = 200$ から $n = 5000$ までは順調に計算処理を進めたが, $n = 25000$ のステップに入ったところで収束を判断する値が 0.2 から 1.2 近辺の値を巡回しはじめ, 10 分を経過し繰り返し回数が 200 回を超えても収束しなかった。

表 9 生産性のグリッド数と追加的な計算所要時間 (秒)

n	$m = 31$		$m = 15$		$m = 7$	
	更新回数	計算時間(秒)	更新回数	計算時間(秒)	更新回数	計算時間(秒)
200	71	1.965	71	1.348	71	1.161
1,000	37	4.140	37	1.985	39	0.973
5,000	21	27.317	22	14.314	24	6.748
25,000	9	192.332	10	101.822	na	na

注) K の最小値と最大値は [21.7136, 89.3128]。計算に使用した Macbook Pro のスペックは, プロセッサが 2.4 GHz Intel Core i5, メモリが 8 GB 1333 MHz DDR3, グラフィックスが Intel HD Graphics 3000 512 MB。計算時間は複数回の試行にもとづく値ではないので, あくまで参考数値。なお, $m = 7, n = 25000$ の場合の値が “na” になっているのは未収束のため。(RBC_PyNumba_nonNegInvest_Iterative.py)

6 結語

動学的意思決定を含むマクロ経済モデルの複雑化に伴って, 近年ではこれらを数値解析し

た研究論文が多数発表されており、解析手法の理解と同時に解析に利用するプログラムの整備が不可欠になっている。そこで本稿では、動学マクロ経済モデルのなかでもっとも基本的な「確率的成長モデル」を材料として、数値解析の手順やシミュレーションの方法などを整理した。検討した計算方法は DSS (Discrete State-Space) 法で、具体的には Value Function Iteration と Modified Policy Function Iteration の 2 種類である。

本稿での試行を通じて、1) 非負条件等の制約を含む動学的最大化問題の数値解析方法として DSS 法が有効であること; 2) 設定する状態変数のグリッド数については、グリッド数が多いほど計算精度は向上するが、同時に所要時間も増加するため、計算精度やシミュレーション等の分析目的にあわせて設定すべきであること; 3) グリッド数が多い場合には Modified Policy Function Iteration を利用することで計算時間を大幅に短縮できること; 4) Modified Policy Function Iteration では反復計算が収束しない事態も発生するので、大規模な反復計算処理を実行する前にグリッド数の設定に関して試行錯誤をすべきこと、などが確認できた。

なお、本稿での数値解析とグラフの作図はすべて汎用プログラミング言語 Python で行った。Python は、近年、大手のソフトウェア企業やインターネット関連企業においても広く利用される言語であるが、同時にプログラミングを専門としない者にとってもコードの可読性や計算処理の効率性、データハンドリングの容易性など多くの点で親しみやすい言語である。本稿で扱った動学的最適化問題は、確率的成長モデルのようなマクロ経済理論のみならず、最近では実証産業組織論の分野でも頻繁に利用されている。この種の分析手法が実証分析の分野で一般化するにつれ、本プログラミング言語の有用性はさらに認識されるだろうと思われる。

(成蹊大学経済学部教授)

参考文献

- [1] 石本敦夫 (2014). 『Python 文法詳解』, オライリー・ジャパン
- [2] McKinney, W. (2013). 『Python によるデータ分析入門 NumPy, pandas を使ったデータ処理』, (小林・鈴木・瀬戸山・滝口・野上 訳), オライリー・ジャパン
- [3] Python サポーターズ (2013). 『パーフェクト Python』, 技術評論社
- [4] Adda, J., and R. Cooper (2003). *Dynamic Economics*. Cambridge, MA: MIT Press
- [5] Aruoba, S.B., and J. Fernández-Villaverde (2014). “A Comparison of Programming Languages in Economics,” *NBER Working Paper*, No.20263.
- [6] Farmer, L.E., and A.A. Toda (2015). “Discretizing Stochastic Processes with Exact Conditional Moments,” *Mimeo*
- [7] Flodén, M. (2008). “A Note on the Accuracy of Markov-chain Approximations to Highly

- Persistent AR(1) Processes,” *Economics Letters*, 99: 516-520.
- [8] Gospodinov, N., and D. Lkhagvasuren (2013). “A Moment-matching method for Approximating Vector Autoregressive Processes by Finite-state Markov Chains,” *Federal Reserve Bank of Atlanta Working Paper Series*, 2013-5.
- [9] Heer, B. and A. Mausnner (2009). *Dynamic General Equilibrium Modeling: Computational Methods and Applications*, Berlin Heidelberg: Springer, 2nd ed.
- [10] Kopecky, K.A., and R.M.H. Suen (2010). “Finite state Markov-chain approximations to highly persistent processes,” *Review of Economic Dynamics*, Vol.13, Issue 3, 701-714.
- [11] Lkhagvasuren, D. (2012). “Key Moments in the Rouwenhorst Method,” *Mimeo*, <http://core.ac.uk/download/pdf/6254042.pdf>
- [12] Rouwenhorst, G.K. (1995). “Asset Pricing Implications of Equilibrium Business Cycle Models,” in Cooley, T. (ed.) *Frontiers of Business Cycle Research*. Princeton: Princeton University Press, 294-330.
- [13] Scheuring, S. and B. Jonen (2013). “Multivariate Markov Chain Approximations,” Available at SSRN: <http://ssrn.com/abstract=2231225> or <http://dx.doi.org/10.2139/ssrn.2231225>
- [14] Tauchen, G. (1986). “Finite State Markov-chain Approximations to Univariate and Vector Autoregressions,” *Economics Letters*, 20: 177-181.
- [15] _____, and R. Hussey (1991). “Quadrature-based Methods for Obtaining Approximate Solutions to Linear Asset Pricing Models,” *Econometrica*, 59: 371-396.