DSGE モデルと MATLAB

寺井　晃*

要旨

本論文は、計算ソフト MATLAB を用いた動学的一般均衡（DSGE）モデルの解説である。様々な DSGE モデルが開発・報告されているが、そうしたモデルで利用されている手法の見通しを良くするために、最も基本的な Ramsey-Cass-Koopmans モデルを例に、MATLAB で解法コードを記述した。

キーワード : DSGE モデル、MATLAB、Ramsey-Cass-Koopmans モデル、対数線形近似、シミュレーション

１．はじめに

本論文は、計算ソフト MATLAB を用いた動学的一般均衡モデル（Dynamic Stochastic General Equilibrium Model: 以下、DSGE モデル）の解法について説明する。

DSGE モデルは、消費者の効用最大化、企業の利潤最大化などの経済主体の最適化を含む、マクロ経済の一般均衡モデルである。この一般均衡問題を、同時点間の最適化問題、異時点間の最適化問題として動学的に解く。その際、解となる変数間の関係を得る際に、モデルを線形化し、行列を用いた計算を必要とすることが多い。また、数値例を用いて繰り返し計算をすることにより、モデルの挙動を示す必要もある。

こうした計算をするにあたっては、手で計算するのは極めて煩雑であるため、PC ソフトの出番である。特に、行列計算に強みを発揮する計算ソフト『MATLAB』は、頻繁に利用されている。

そこで本論文は、MATLAB を用いた DSGE モデルの解法を、最も単純な Ramsey-Cass-Koopmans モデルを例として論じていく。紙と鉛筆で解くことができるモデルを、MATLAB のコードと対応させることにより、PC ソフトを用いて DSGE モデルを解く際の見通しを良くする。このことが、本論文の目的の一つである。

* 京都産業大学経済学部。本論文を執筆するにあたり、JSPS 科研費 26380336 の助成を受けた。

２．先行研究

DSGE モデルの論文は、最近は日本語でも増えてきている。加藤（2007）や矢野（2008）の紹介による影響が、指摘できるだろう。加藤（2007）は、DSGE モデルに関するテキストである。リアルビジネスサイクルモデル（RBC モデル）を「ピザの生地」、独占的競争や名目価格の硬直性などの分析者の導入する仮定を「具」として、様々な DSGE モデルを紹介している。また、本論文と同様に、MATLAB コードの紹介もしていることに特徴がある。矢野（2008）は、DSGE モデルを MATLAB 拡張ソフトの 1 つである Dynare で解く手法を解説した論文である。

こうした分析は、主に政策の分析を示すことを意図して、政府・中央銀行が報告することが多い[1]。

まず、日本銀行はスタッフが「JEM (Japanese Economic Model)」と呼ばれる大規模マクロ経済モデルを開発し、その成果を公表してきた（Fujiwara, Hara, Hirose and Teranishi (2005)）。また、「Q-JEM (Quarterly-Japanese Economic Model)」と呼ばれる大規模マクロ経済モデルも開発している。このモデルの成果を用いて「経済・物価情勢の展望」が書かれており、金融政策の参考としていることがうかがわれる（一上・北村・小島・代田・中村・原（2009）、福永・原・小島・上野・米山（2011））。また、「M-JEM (Medium-scale Japanese Economic Model)」と呼ばれる中規模の動学的一般均衡モデルも開発し、利用していることをうかがわせる（笛木・福永（2011））。

Sugo and Ueda (2007)は、日本経済の中規模 DSGE モデルを、先行研究で提示されたモデルを修正・比較する形で推定し、金融政策ショックに対するインフレ率の反応を導いている。Ichiue, Kurozumi and Sunakawa (2008)はベイズ DSGE モデル推定により、インフレ率の変動と労働投入の調整の関係を議論している。Kitamura (2010)はゼロ金利下での金融緩和政策について、パーティクル・フィルターを応用して提案している。Fueki, Fukunaga, Ichiue and Shirota (2010)は潜在成長率と GDP ギャップを算出している。Sudo (2012)は指標金利の役割をニューケインジアン DSGE モデルで分析している。

また、政府では、内閣府経済社会総合研究所が「次世代短期マクロ計量モデル（DSGE モデル）の開発」を掲げている。Yano (2009)は、 DSGE モデル推定方法について時変係数アプローチによる手法を提案し、流動性の罠の下でのニューケインジアン DSGE モデルを推定した。Iwata (2009)は財政当局を明示したニューケインジアン DSGE モデルを推定し、政策シミュレーションをした。Adjemian and Juillard (2009)は DSGE モデルにおけるトレンド処理を議論し、日本経済のデータに応用した。

このように、DSGE モデルはマクロ経済分析や政策評価において必携ツールともいえる。DSGE モデルには批判も多いが、その分析構造を理解しておくことは、その批判が的を射たものか判断するためにも有用であろう。

---

[1] 日本のみならず各国の DSGE モデル利用については、佐藤（2009）が詳しい。

３．Ramsey-Cass-Koopmans モデル

最も単純な DSGE モデルとして、Ramsey-Cass-Koopmans モデルについて取り上げよう。

まず、消費者は次の効用を最大にするように、各期tの消費$\{c_t\}_{t=0}^{\infty}$を選ぶ。

$$\max_{c_t} E_0 \sum_{t=0}^{\infty} \beta^t \ln c_t \qquad (1)$$

単純化のため、各時点の効用関数は対数関数である。βは割引率であり、0 と 1 の間の数である。E_0は$t=0$時点での期待値オペレータである。

資本k_tの蓄積は、i_tを各期の投資として、単純化のために 100%の減価償却を仮定する。

$$k_{t+1} = i_t \qquad (2)$$

また、$t=0$以前の資本k_{-1}は所与である。

各期の生産y_tは、次の生産関数に基づいて行われる。

$$y_t = A_t k_t^{\alpha} \qquad (3)$$

ここで、Aは生産性を表すパラメータ、αは 0 と 1 の間の数である。生産性は、以下のようなショック付きの AR(1)過程に従うとする。

$$\ln(A_{t+1}) = \rho \ln(A_t) + \epsilon_t \qquad (4)$$

ϵ_tは iid のホワイトノイズである。

そして、資源制約として、生産されたものは消費か投資にしか廻らない。

$$y_t = c_t + i_t \qquad (5)$$

以上が、Ramsey-Cass-Koopmans モデルの設定である。資本蓄積、技術、資源の各制約を前提に、消費者は効用が最大になる消費を選ぶのである。主体（消費者）が最適な値を選ぶという意味で、c_tは control variable とも呼ばれる[2]。また、各期において、主体が所与とする変数を state variable と呼ぶ。このモデルの場合は、k_t、A_tが state variable である。State variable は、主体が環境を変えられる変数と変えられない変数に分けられる。前者を endogenous state variable という場合もあり、このモデルでは資本蓄積を通して変更されるk_tである。後者は exogenous state variable という場合もあり、このモデルの場合はA_tである。

このモデルは、多くの場合、次のラグランジアンを設定して解く。

$$\mathcal{L} = E_0 \sum_{t=0}^{\infty} \beta^t \{\ln c_t + \lambda_t (A_t k_t^{\alpha} - c_t - k_{t+1})\} \qquad (6)$$

λ_tはラグランジュ乗数である。ここから、消費者の選択する変数c_t、消費者の選択次第で変動する変数k_{t+1}、ラグランジュ乗数λ_tについて一階条件（FOC: First Order Condition）を求めると、次の通りとなる。

$$c_t : \frac{1}{c_t} = \lambda_t \qquad (7)$$

[2] jump variable と呼ぶ場合もある。

$$k_{t+1}: \lambda_t = E_t \beta \lambda_{t+1} \alpha A_t k_{t+1}^{\alpha-1} \quad (8)$$

$$\lambda_t: k_{t+1} = A_t k_t^{\alpha} - c_t \quad (9)$$

まずは、このような最適化条件に従った定常状態を求めよう。(4) 式から、$\bar{A} = 1$となる[3]。これを (8) 式に代入すれば、

$$\bar{k} = (\alpha\beta)^{\frac{1}{1-\alpha}} \quad (10)$$

となる。これを (9) 式に代入すれば、

$$\bar{c} = (\alpha\beta)^{\frac{1}{1-\alpha}} \left(\frac{1}{\alpha\beta} - 1 \right) \quad (11)$$

となる。

このモデルが解けるというのは、policy function を見つけ出すことである。つまり、現在の state をインプットとして、control をアウトプットとして出す関数を求めることである。数式で表せば、(7) (8) (9) 式を満たす$c_t = f(k_t、\ A_t)$というfを探すことである。

詳細は省くとして、このモデルの policy function は

$$c_t = (1 - \alpha\beta) A_t k_t^{\alpha} \quad (12)$$

となる[4]。

4．対数線形近似

上の設定の Ramsey-Cass-Koopmans モデルは、policy function を closed form で求めることが可能である。しかし、(2) 式のような資本蓄積でない場合、つまり、減価償却率が 100%でない場合、解を closed form で得ることは難しい[5]。

従って、そのようなモデルを扱う場合、定常状態周りで解がどのような振る舞いなのか、対数線形近似して求めるという手法が用いられる。1 次の近似を行えば、変数同士の関係を線形で表すことができるので、行列による差分方程式の表現が可能となる。そして、この行列による表現を操作すれば、policy function を求めることができるのである。

上の設定の Ramsey-Cass-Koopmans モデルの場合、モデルを解くのに必要な条件は、FOC である (7) (8) 式、制約条件 (9) 式、技術の条件 (4) 式である。これらの式を定常状態周辺で対数線形近似すると、次のような式となる[6]。

---

[3] 各変数の定常状態は、各変数の上にバーを付けた記号で表す。

[4] 詳しくは、Ljungqvist and Sargent (2004)などを参照のこと。このモデルの場合は、減価償却がないという点で関数形がかなり特殊なので、陽表的に解くことが可能である。解を得る手法として、Ljungqvist and Sargent (2004)は 3 つの方法を提示している。1. Value function iteration、 2. Guess and Verify、 3. Howard's Improvement algorithm。

[5] そのため、(7) (8) 式からc_tが一定の条件、(9) 式からk_tが一定の条件を求め、c, k平面にそれぞれの条件を記し、それぞれの変数が定常状態にどのように向かうかという図 (位相図 : phase diagram) による分析に終わることが多い。この経路は鞍点経路 (saddle path) と呼ばれる。policy function は、この鞍点経路を陽表的に得るものである。

[6] ある変数X_tの定常状態周りでの 1 次の対数線形近似は、次の式によって求められる。

$$-\hat{c}_t = \hat{\lambda}_t \tag{13}$$

$$\hat{\lambda}_t = E_t\hat{\lambda}_{t+1} + E_t\hat{a}_t + (\alpha - 1)E_t\hat{k}_{t+1} \tag{14}$$

$$\bar{c}\hat{c}_t + \bar{k}\hat{k}_{t+1} = \bar{k}^{\alpha}\hat{a}_t + \alpha\bar{k}^{\alpha}\hat{k}_{t+1} \tag{15}$$

$$\hat{a}_{t+1} = \rho\hat{a}_t + \epsilon_{t+1} \tag{16}$$

（13）式は同時点間、（14）（15）式は異時点間の関係を表している。ここで、同時点間の式を以下のような行列形式に書き換える。

$$(-1)(c_t) = (0 \quad 1)\begin{pmatrix} k_t \\ \lambda_t \end{pmatrix} + (0)(a_t) \tag{17}$$

左辺に control variable のベクトル(c_t)を作成し、右辺に endogenous state のベクトル$(k_t、\ \lambda_t)'$と exogenous state のベクトル(a_t)を作成して書き換えたものである。

異時点間の関係は、次のように書き換えられる。

$$\begin{pmatrix} \alpha - 1 & 1 \\ \bar{k} & 0 \end{pmatrix}\begin{pmatrix} E_t k_{t+1} \\ E_t\lambda_{t+1} \end{pmatrix} + \begin{pmatrix} 0 & -1 \\ -\alpha\bar{k}^{\alpha} & 0 \end{pmatrix}\begin{pmatrix} k_t \\ \lambda_t \end{pmatrix} =$$

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix}(E_t c_{t+1}) + \begin{pmatrix} 0 \\ -\bar{c} \end{pmatrix}(c_t) + \begin{pmatrix} -1 \\ 0 \end{pmatrix}(E_t a_{t+1}) + \begin{pmatrix} 0 \\ \bar{k}^{\alpha} \end{pmatrix}(a_t) \tag{18}$$

行列やベクトルに適当な名前を振ると、（17）式は以下のように簡略化される[7]。

$$M_{cc}u_t = M_{cs}\begin{pmatrix} x_t \\ \lambda_t \end{pmatrix} + M_{ce}z_t \tag{19}$$

但し、u_tは control variable のベクトル、x_tは endogenous state variable のベクトル、λ_tはラグランジュ乗数（co-state variable）のベクトル、z_tは exogenous state variable のベクトルを表している。（18）式は次のようになる。

$$M_{ss}^0 E_t\begin{pmatrix} x_{t+1} \\ \lambda_{t+1} \end{pmatrix} + M_{ss}^1\begin{pmatrix} x_t \\ \lambda_t \end{pmatrix} = M_{sc}^0 E_t u_{t+1} + M_{sc}^1 u_t + M_{se}^0 E_t z_{t+1} + M_{se}^1 z_t \tag{20}$$

モデルを解くことは、$u_t = f(x_t、\ z_t)$となる関係を求めることである。これは policy function と呼ばれる。また、同時に$x_{t+1} = g(x_t、\ z_t)$となる関係も求めれば、変数の動学的な振る舞いも判明する。

$$f(X_t) \cong f(\bar{X}) + f'(\bar{X})\bar{X}\hat{x}_t$$

ここで、$\bar{X}$はX_tの定常状態の値、$\hat{x}_t$はX_tが定常状態の値から何パーセント離れているかを示したものである。こうした近似を行う際は、両辺とも行うことがほとんどで、近似の第 1 項は両辺で打ち消しあう。

例えば、$Y_t = X_t^{\alpha}$を近似する場合、左辺は$\bar{Y} + \bar{Y}\hat{Y}_t$、右辺は$\bar{X}^{\alpha} + \alpha\bar{X}^{\alpha-1}\bar{X}\hat{X}_t$となる。この近似後の右辺と左辺も等しく、定常状態では$\bar{Y} = \bar{X}^{\alpha}$なのだから、

$$\bar{Y} + \bar{Y}\hat{Y}_t = \bar{X}^{\alpha} + \alpha\bar{X}^{\alpha}\hat{X}_t$$
$$\Leftrightarrow \quad \bar{Y}\hat{Y}_t = \alpha\bar{X}^{\alpha}\hat{X}_t$$
$$\Leftrightarrow \quad \hat{Y}_t = \alpha\hat{X}_t$$

である。

[7] ここでは、Burnside (1999)のノテーションをアレンジして用いる。Burnside (1999)は、King, Plosser and Rebelo (1988)のノテーションを使っているとしている。

５．解法

モデルを解く際に扱いが大事になるのが、endogenous state variable である。これらは差分方程式の形になっている上、通常の経済モデルを考えている限りにおいて、この変数が発散しないことが条件となる。

一般に、k_tは過去からの蓄積があるため、backward に解かれる。λ_tは横断面条件を満たすために、forward に解かれる。これらを上手く識別し、それぞれの変数についての比較的簡単な解法を与えたのが Blanchard and Kahn (1980)である。（19）式を control variable について解き、それを（20）式に代入し、係数行列に適当な文字を割り振ると、次のような式となる。

$$E_t\begin{pmatrix}x_{t+1}\\ \lambda_{t+1}\end{pmatrix} = W\begin{pmatrix}x_t\\ \lambda_t\end{pmatrix} + RE_t z_{t+1} + Qz_t \tag{21}$$

（21）式は、このモデルの形に限らないかなり一般的な形式である。この式のWを固有値分解し、固有値の絶対値が 1 より大きいか小さいかにより、$(k_t、\ \lambda_t)'$に含まれる変数を forward に解くか backward に解くか区別する。一般に、絶対値が 1 より小さい固有値の数と、過去からの蓄積によって求まる変数（predetermined variable という）の数が等しくないと、解の経路が発散したり複数均衡となる。通常のモデルを想定している場合、predetermined variable の数と絶対値が 1 より小さい固有値の数は一致することが要請される[8]。

これを、本論文で関心の Ramsey-Cass-Koopmans モデルの形に特化して、解法を大雑把に見ていこう。（18）に（17）を代入すると、control variable の部分を消去できる。

$$\begin{aligned}E_t\begin{pmatrix}k_{t+1}\\ \lambda_{t+1}\end{pmatrix} &= \begin{pmatrix}\alpha-1 & 1\\ k & 0\end{pmatrix}^{-1}\begin{pmatrix}0 & 1\\ \alpha k^{\alpha} & c\end{pmatrix}\begin{pmatrix}k_t\\ \lambda_t\end{pmatrix} + \begin{pmatrix}\alpha-1 & 1\\ k & 0\end{pmatrix}^{-1}\begin{pmatrix}-1\\ 0\end{pmatrix}a_{t+1}\\ &\quad + \begin{pmatrix}\alpha-1 & 1\\ k & 0\end{pmatrix}^{-1}\begin{pmatrix}0\\ k^{\alpha}\end{pmatrix}a_t\end{aligned} \tag{22}$$

$$= W\begin{pmatrix}k_t\\ \lambda_t\end{pmatrix} + \begin{pmatrix}0\\ -1\end{pmatrix}a_{t+1} + \begin{pmatrix}k^{\alpha-1}\\ (1-\alpha)k^{\alpha-1}\end{pmatrix}a_t \tag{23}$$

ここで、Wを固有値分解した結果が、$W = P\Lambda P^{-1}$であるとする。また、$P^{-1}(k_t、\ \lambda_t)' \equiv (\tilde{k}_t、\ \tilde{\lambda}_t)$とする。また、各行列の要素を以下のような文字で指定する。

$$W = \begin{pmatrix}W^{11} & W^{12}\\ W^{21} & W^{22}\end{pmatrix} \tag{24}$$

$$P^{-1} = \begin{pmatrix}P^{11} & P^{12}\\ P^{21} & P^{22}\end{pmatrix} \tag{25}$$

$$\Lambda = \begin{pmatrix}\Lambda_1 & 0\\ 0 & \Lambda_2\end{pmatrix} \tag{26}$$

[8] 加藤（2007）は、この条件を「Blanchard-Kahn の定理」と呼んでいる。

$$R = \begin{pmatrix} R_x \\ R_l \end{pmatrix} \tag{27}$$

$$Q = \begin{pmatrix} Q_x \\ Q_l \end{pmatrix} \tag{28}$$

ここで、Λは固有値の絶対値の小さい順に並んでおり、$|\Lambda_1| < 1$、$|\Lambda_2| > 1$とする。

すると、(23) 式はP^{-1}を前からかけることによって次のように書き換えられる。

$$E_t \begin{pmatrix} \tilde{k}_{t+1} \\ \tilde{\lambda}_{t+1} \end{pmatrix} = \Lambda \begin{pmatrix} \tilde{k}_t \\ \tilde{\lambda}_t \end{pmatrix} + P^{-1} R a_{t+1} + P^{-1} Q a_t \tag{29}$$

(29) 式の$\tilde{\lambda}_{t+1}$に関する式を抜き出すと、

$$\tilde{\lambda}_{t+1} = \Lambda_2 \tilde{\lambda}_t + (P^{21} R_x + P^{22} R_l) a_{t+1} + (P^{21} Q_x + P^{22} Q_l) a_t \tag{30}$$

$|\Lambda_2| > 1$なので、$\tilde{\lambda}_t$は forward に解かれる。$E_t a_{t+1} = \rho a_t$に注意すると、

$$\tilde{\lambda}_t = \frac{1}{\Lambda_2} \tilde{\lambda}_{t+1} - \frac{1}{\Lambda_2} [P^{21} R_x \rho + P^{22} R_l \rho + P^{21} Q_x + P^{22} Q_l] a_t \tag{31}$$

$$= -\frac{1}{\Lambda_2 - \rho} [P^{21} R_x \rho + P^{22} R_l \rho + P^{21} Q_x + P^{22} Q_l] a_t \tag{32}$$

ところで、$\tilde{\lambda}_t = P^{21} k_t + P^{22} a_t$なのだから、

$$\lambda_t = -(P^{22})^{-1} P^{21} k_t + (P^{22})^{-1} \tilde{\lambda}_t \tag{33}$$

(23) 式のk_{t+1}に関する式を抜き出し、(33) 式を代入すると、

$$k_{t+1} = W^{11} k_t - W^{12} (P^{22})^{-1} P^{21} k_t + W^{12} (P^{22})^{-1} \tilde{\lambda}_t + (\rho R_x + Q_x) a_t \tag{34}$$

この$\tilde{\lambda}_t$に (32) 式を代入すると、

$$\begin{aligned} k_{t+1} = {} & (W^{11} - W^{12} (P^{22})^{-1} P^{21}) k_t \\ & - W^{12} (P^{22})^{-1} \frac{1}{\Lambda_2 - \rho} [P^{21} R_x \rho + P^{22} R_l \rho + P^{21} Q_x + P^{22} Q_l] a_t \\ & + (\rho R_x + Q_x) a_t \end{aligned} \tag{35}$$

$$= \Upsilon_{xx} k_t + \Upsilon_{xz} a_t \tag{36}$$

こうして、時点tの state variable のみで翌期の endogenous state variable を決定する式が求まった。これを (15) 式に代入すれば、

$$c_t = \Upsilon_{ux} k_t + \Upsilon_{uz} a_t \tag{37}$$

という policy function が求まる。

6．インパルスレスポンス関数

(12) 式で示された policy function を再掲すると、

$$c_t = (1 - \alpha\beta) A_t k_t^{\alpha} \tag{38}$$

これを、定常状態周りで対数線形近似すると、

$$\bar{c} \hat{c}_t = \alpha (1 - \alpha\beta) \bar{k}^{\alpha} \hat{k}_t + \alpha (1 - \alpha\beta) \bar{k}^{\alpha} \hat{a}_t \tag{39}$$

ここで、定常状態のcとkの値に注目すると、$\frac{\bar{k}^{\alpha}}{\bar{c}} = \frac{1}{1-\alpha\beta}$であるので、

$$\hat{c}_t = \alpha\hat{k}_t + \hat{a}_t \tag{40}$$

一方で、こうした最適な消費を選ぶ時のkの振る舞いは（15）式によって与えられ、

$$\bar{c}\hat{c}_t + \bar{k}\hat{k}_{t+1} = \alpha\bar{k}^{\alpha}\hat{k}_t + \bar{k}^{\alpha}\hat{a}_t \tag{41}$$

$$\hat{k}_{t+1} = \alpha\hat{k}_t + \hat{a}_t \tag{42}$$

となる。（40）（42）式が、このモデルの解である。そして、この式にショックを与えた際のモデルの挙動を描くことが可能である。例えば、$\alpha = 0.3$、$\beta = 0.9$、$\rho = 0.9$とする。また、$t = 0$時点で定常状態だった。ここで、$t = 1$で$\hat{a}_1 = 1$というショックがあったとする。

$t = 1$において、前期は定常状態なのだから、（42）式により、$\hat{k}_1$に変化はない。したがって、$\hat{k}_1 = 0$である。$\hat{k}_1 = 0$と$\hat{a}_1 = 1$から、（40）式によって、$\hat{c}_1 = 0.3\hat{k}_1 + \hat{a}_1 = 1$が求まる。また、（42）式により、$\hat{k}_2 = 0.3\hat{k}_1 + \hat{a}_1 = 1$が求まる。

$t = 2$において、技術はAR(1)に従い、その係数は$\rho = 0.9$なのだから、$\hat{a}_2 = 0.9\hat{a}_1 = 0.9$である。$\hat{k}_2 = 1$が前期に求まっているのだから、これを所与とした$\hat{c}_2 = 0.3\hat{k}_2 + \hat{a}_2 = 0.3 \cdot 1 + 0.9 = 1.2$が求まる。また、（42）式により、$\hat{k}_3 = 0.3\hat{k}_2 + \hat{a}_2 = 1.2$が求まる。

$t = 3$において、$\hat{a}_3 = 0.81$である。$\hat{k}_3 = 1.2$が前期に求まっているのだから、これを所与とした$\hat{c}_3 = 0.3\hat{k}_3 + \hat{a}_3 = 0.3 \cdot 1.2 + 0.81 = 1.17$が求まる。また、（42）式により、$\hat{k}_4 = 0.3\hat{k}_3 + \hat{a}_3 = 1.17$が求まる。

このように繰り返していき、$\{\hat{c}_t\}$を求める。これを図に描いたものが、インパルスレスポンス関数となる。

7．MATLAB コード

本節では、本論文でここまで描写してきたRamsey-Cass-Koopmansモデルを MATLAB コードで示す。

本コードは、大きく分けて以下の通りで構成されている。

1. 変数の数のチェック
2. パラメータの数値設定
3. 定常状態の計算
4. 対数線形化したモデルの記述
5. モデルの計算
6. 解の表示
7. インパルスレスポンス関数によるシミュレーション

以下ではこれらを順に見ていく。以下、Courier New フォントになっている箇所が、MATLAB コードの箇所である[9]。ここで示したコードを実行する際は、インデントを与え

[9] MATLAB コードにおいて、%はコメントアウトを示し、それに続く文字列をコードと解釈しない。

ている Courier New フォントの箇所を一続きにすればよい。

７－１．変数の数のチェック

```
clear all;
nc=1;       % c(t)
ns=1;       % k(t)
nex=1;      % a(t)
ncs=1;      % lambda(t)
nb=ns;      % # of backward
nt=ns+ncs;  % # of total state variables
```

まず、clear all;でワークスペースからアイテムを削除し、システムメモリを解放する。nc は control variable の数である。ns は endogenous state variable の数である。nex は exogenous state variable の数である。ncs は co-state variable の数である[10]。nb はバックワードに決まる変数の数で、この場合は ns の数と同じである。また、nt は state variable の総数である。

７－２．パラメータの数値設定

```
alpha=0.3;   % Cobb-Douglas parameter、 capital share
beta=0.9;    % time preference
rho=0.9;     % AR(1) coefficient
disp('parameter values [alpha beta rho]')
disp(' ');
disp([alpha beta rho]);
```

モデルではギリシャ文字で表現してきた各パラメータについて、それぞれ対応する文字列で変数名を割り当てる。この場合、$\alpha = 0.3$、$\beta = 0.9$、$\rho = 0.9$である。

disp('XXX')は、コマンドウインドウ内に対し、括弧内に記したものを表示する。アポストロフィで括った箇所は文字列として、括られた中の文字がそのまま表示される。括らなければ、MATLAB のコードのルールに従った結果を表示する。すなわち、disp('parameter values [alpha beta rho]')により

```
parameter values [alpha beta rho]
```

とそのままコマンドウインドウに表示され、disp([alpha beta rho])により

```
0.3000    0.9000    0.9000
```

[10] それぞれ、number of control、 number of state、 number of exogenous、 number of co-state の頭文字である。

と表示する。これらを利用して、パラメータにどの値を与えたか、コード実行時に確認する。

７－３．定常状態の計算

```
kstar=(alpha*beta)^(1/(1-alpha));  % Capital
cstar=kstar^alpha-kstar;           % Consumption
lambdastar=1/cstar;                % Shadow Price
ystar=kstar^alpha;                 % Output
disp('SS values [kstar cstar]')
disp(' ');
disp([kstar cstar]);
```

ここでは、モデルの計算結果に従って、各変数の定常状態の値を計算している。これらの計算は、第 3 節に対応している。

７－４．対数線形化したモデルの記述

```
Mcc=[-1];
Mcs=[0 1];
Mce=[0];
Mss0=[alpha-1 1
      kstar 0];
Mss1=[0 -1
    -alpha*kstar^alpha 0];
Msc0=[0
    0];
Msc1=[0
    -cstar];
Mse0=[-1
     0];
Mse1=[0
     kstar^alpha];
PI=[rho];
```

ここでは、(19)（20）式（あるいはこのモデルに特化した場合、(17)（18）式）に対応した行列の指定をしている。これらのコードは第 4 節に対応している。

７－５．モデルの計算

```
Mbarss= Mss0-Msc0*inv(Mcc)*Mcs;
W=-inv(Mbarss)*(Mss1-Msc1*inv(Mcc)*Mcs);
R=inv(Mbarss)*(Msc0*inv(Mcc)*Mce+Mse0);
Q=inv(Mbarss)*(Msc1*inv(Mcc)*Mce+Mse1);
```

Mbarss は、(21) 式導出のための途中経過の行列の係数である。W は (21) 式のWに対応する。R、 Q も同様、(21) 式のR、 Qに対応する。

```
[pr、lambr] = eig(W);
alamb=abs(diag(lambr))';
[lambs、lambz] = sort(alamb);
lambda=lambr(lambz、 lambz);
p=pr(:、lambz);
ps=inv(p);
```

eig は固有値を取り出すコマンドである。取り出した固有値について、固有ベクトルを pr、固有値の対角行列を lambr に割り当てる。diag は行列の対角成分を取り出すコマンドである。abs でその絶対値を取り、これを alamb という変数名に割り当てる。sort は小さい順に並べ替えるものであり、lambs は並べ替えた結果、lambz は元のベクトルの成分が何番目かを示す情報である。pr(:、lambz)は、固有値を lambz の順に並べ替えることを意味し、その結果を ps という変数名に割り当てる。

```
p11=p(1:nb, 1:nb);
p12=p(1:nb, nb+1:nt);
p21=p(nb+1:nt, 1:nb);
p22=p(nb+1:nt, nb+1:nt);
```

これは、p で割り当てられた固有ベクトルから、p11 がバックワードに決まる変数の数の正方行列となるように、左上の要素として抜き出す操作である。

```
ps11=ps(1:nb, 1:nb);
ps12=ps(1:nb, nb+1:nt);
ps21=ps(nb+1:nt, 1:nb);
ps22=ps(nb+1:nt, nb+1:nt);
```

同様に、ps で割り当てられた固有ベクトルから、ps11 がバックワードに決まる変数の数の正方行列となるように、左上の要素として抜き出す操作である。この操作は（25）式に対応する。

```
Qx=Q(1:nb, 1:nex);
Qlambda=Q(nb+1:nt, 1:nex);
Rx=R(1:nb, 1:nex);
Rlambda=R(nb+1:nt, 1:nex);
```

これも同様に、Q、 R と指定されたベクトルから、上からバックワードに決まる変数の数の分を抜き出してきたのを Rx、 Qx、それ以外を Qlambda、 Rlambda と指定する操作である。これらは、(30）式を導出するための準備である。

```
lamb1=lambda(1:nb, 1:nb);
lamb2=lambda(nb+1:nt, nb+1:nt);
```

固有値を、上からバックワードに決まる変数の数の分だけ抜き出したのを lamb1、それ以外を lamb2 と指定する。lamb2 は（30）式に出てくるΛ_2である。

```
phi0=ps21*Rx+ps22*Rlambda;
phi1=ps21*Qx+ps22*Qlambda;
```

これらは、(30）式導出のための計算である。

```
ww=W(:, lambz);
w11=ww(1:nb, 1:nb);
w12=ww(1:nb, nb+1:nt);
w21=ww(nb+1:nt, 1:nb);
w22=ww(nb+1:nt, nb+1:nt);
```

W について、固有値の大きさの順にソートしなおしたのを ww と指定する。そして、ww から w11 がバックワードに決まる変数の数の正方行列となるように、左上の要素として抜き出す。

```
AA=ps21*Rx*rho+ps22*Rlambda*rho+ps21*Qx+ps22*Qlambda;
```

これは、(31) 式導出のための計算である。

```
upsilonxz=rho*Rx+Qx-w12*inv(ps22)*1/(lamb2-rho) *AA;
upsilonxx = w11-w12*inv(ps22)*ps21;
```

これは、(36) 式導出のための計算である。upsilonxz がΥ_{xz}、upsilonxx がΥ_{xx}に対応する。

```
upsilonux = (alpha*kstar^alpha - kstar*upsilonxx)/cstar;
upsilonuz = (kstar^alpha - kstar * upsilonxz)/cstar;
```

これは、(37) 式導出のための計算である。Upsilonux がΥ_{ux}、upsilonuz がΥ_{uz}に対応する。但し、この計算箇所は本論文で扱っているモデルに則って計算しているため、他のモデルを解く際は修正する必要がある。

７－６．解の表示

```
solx = [upsilonxx upsilonxz];
solc = [upsilonux upsilonuz];
```

solx という形で、state variable の推移の式の変数をまとめておく。同様に、solc という形で、policy function の変数をまとめておく。

```
disp('policy function coefficients, state(t) to control(t)')
disp(' ');
disp(upsilonux);

disp('policy function coefficients, exogenous(t) to control(t)')
disp(' ');
disp(upsilonuz);
```

disp コマンドにより、policy function の係数を確認する。

７－７．インパルスレスポンス関数によるシミュレーション

```
t=60;
```

まず、シミュレーションする期間を指定する。この場合、60 期間分の結果を表示する。

```
k=zeros(t,1);
c=zeros(t,1);
a=zeros(t,1);
y=zeros(t,1);
```

シミュレーションの変数を作成する。これらは、取りあえずすべての要素が 0 の 60 期間分の縦ベクトルを作成したものである。

```
sho=1;
```

与えるショックの大きさを指定する。この場合は、1 単位である。

```
a(1,1)=[sho];
k(1,1)=solx*[0;0];
c(1,1)=solc*[0;sho];
y(1,1)=a(1,1)+alpha*k(1,1);
```

第 1 期の計算をする。a にショックを与える。そのショックに基づき、第 1 期のそれぞれの値を計算している。k の箇所は state variable の推移の式であり、c の箇所は policy function である。y は、(3) 式を対数線形近似したものである。

```
for i=2:t
 k(i,1)=solx*[k(i-1,1);a(i-1,1)];
 a(i,1)=rho*a(i-1,1);
 c(i,1)=solc*[k(i,1);a(i,1)];
 y(i,1)=a(i,1)+alpha*k(i,1);
end
```

for ループにより、繰り返し次の期の計算を行っている。

```
figure(1)
subplot(3,1,1);
plot ( c );
ylabel ( 'consumption' )
xlabel('time');
```

```
subplot(3,1,2);
plot ( k );
ylabel( 'capital' );
xlabel( 'time');
subplot(3,1,3);
plot ( y );
ylabel( 'GDP' );
xlabel( 'time');
```

figure(1)により、1つ目の図であることを示す。subplot(3,1,1)は、図を3つ並べる内の1つ目であることを示す。plot (c)でcの推移を示し、ylabel ('consumption')でy軸に消費とラベルを付け、xlabel('time')でx軸に時間の推移のラベルを付ける。subplot(3,1,2)は、図を3つ並べる内の2つ目であることを示す。あとは同様に資本の推移、GDPの推移を描く。

```
figure(2)
plot(k,c,'-+');
ylabel( 'consumption' );
xlabel('capital');
```

2つ目の図として、位相図を描こう。figure(2)により、2つ目の図であることを示す。plot(k,c,'-+')は、x軸にk、y軸にcをプロットし、順に折れ線で結ぶものである。'-+'は折れ線の形状を指定するもので、-は実線をつなぎ、+は各地点を+マークで示す。

以上で本コードは終了である。

8．おわりに

本論文では、MATLABを用いたDSGEモデルの解法の一例を示した。読者はモデルをアレンジし、それに合わせてコードをアレンジすればDSGEモデルを実践することができるだろう。しかし、本論文で用いたMATLABは汎用性は高いものの、DSGEモデルを解くに際して幾つかの制約を挙げなければならないだろう。

MATLABはあくまで計算ソフトであり、DSGEモデルを解くのに特化した、あるいは意識した計算ソフトではない。従って、DSGEモデルを解く際はそれに合わせてコードをアレンジしなければならない。但し、MATLABの特徴の一つとして、関数M-ファイルの呼び出しというのがある。これは、一定の手続きを記したものを、実行するファイルとは別ファイルに作成し、実行するファイルに関数M-ファイルの読み込みコードを記して実行すれば、関数M-ファイルに書いた手続きに従って結果が表示されるというものである。関数

M-ファイルに定義した文字は実行ファイルと重複しないので、一定のアルゴリズムを関数M-ファイルにしておけば省力化が図れる。

例えば、本節で問題とした Ramsey-Cass-Koopmans モデルは、パラメータの値や定常状態の値、FOC などの挙動はこのモデルに特有のものである。(17)、(18) の形で記されるのは、このモデル特有の形式といえる。しかし、その後の行列表現や固有値を求める手続き、policy function に相当する行列の算出等は、比較的一般的な形式である。従って、一旦この手続きを関数 M-ファイルに記しておけば、この箇所のコードを繰り返し記す手間が省ける。同種のモデルを解く際に、使いまわしが利くのである。

従って、対数線形近似によるモデルの導出は、モデル特有の作業となるので、手作業なりモデルに合わせてコードを書くなりの作業となることが多い。行列の要素を指定するといった作業も、モデルの特徴を指定することに他ならないので、省力化することは難しい。その後の一般的な計算手続きは、一旦関数 M-ファイルを作成すれば省力化できるかもしれない。また、本論文で述べた解法は、対数線形近似を 2 次以降行うことを想定していない。2 次以上の近似によりモデルを解く場合、また別の手順を記した関数 M-ファイルを作成しなくてはならない。

この辺りの作業について、DSGE モデルを解くことを目指したソフトが Dynare である[11]。Dynare は単体では利用できず、計算については MATLAB などを利用するため[12]、ある意味 MATLAB へのアドインである。パラメータの記述、モデルの記述を Dynare コードに従って適切にできれば、本論文で述べた計算を一つ一つ行うことなく、容易に結果を得ることができる。マルコフ連鎖モンテカルロ法でパラメータ推定ができることや、カルマンフィルターの利用など、オプションも豊富である。但し、Dynare コードを実行するのみでは、実際にどのような計算を行っているのかがブラックボックスとなってしまうので、解法に興味のある者には向かない。

DSGE モデルの構造を理解したうえで Dynare を利用することは非常に有益だが、本論文で行ったように、モデル構造を MATLAB コードに記して計算を実行することは、相変わらず教育・研究効果が高いものと思われる。

参考文献

Adjemian, Stéphane and Michel Juillard (2009), "Dealing with trends in DSGE models. An application to the Japanese economy," ESRI Discussion Paper Series No.224.

Blanchard, Olivier J. and Charles M. Kahn (1980), "The Solution of Linear Difference Models under Rational Expectations," Econometrica, 48, pp.1305-1311.

11 以下のサイトで配布されている。
http://www.dynare.org/

12 MATLAB 以外では、Octave 版も配布されている。

Burnside, Craig (1999), "Real Business Cycle Models: Linear Approximation and GMM Estimation," unpublished manuscript.

Fueki, Takuji, Fukunaga Ichiro, Ichiue Hibiki and Shirota Toyoichiro (2010), "Measuring Potential Growth with an Estimated DSGE Model of Japan's Economy," 日本銀行ワーキングペーパーシリーズ 10-E-13.

Fujiwara, Ippei, Hara Naoko, Hirose Yasuo and Teranishi Yuki (2005), "The Japanese Economic Model (JEM)," Monetary and Economic Studies, 23 (2), pp.61-142.

Ichiue, Hibiki, Kurozumi Takashi and Sunakawa Takeki (2008), "Inflation Dynamics and Labor Adjustments in Japan: A Bayesian DSGE Approach," 日本銀行ワーキングペーパーシリーズ 08-E-9.

Iwata, Yasuharu (2009), "Fiscal Policy in an Estimated DSGE Model of the Japanese Economy: Do Non-Ricardian Households Explain All?" ESRI Discussion Paper Series No.216.

Ljungqvist, Lars and Thomas Sargent (2004), Recursive Macroeconomic Theory, The MIT Press.

King, Robert G., Charles I. Plosser and Sergio T. Rebelo (1988), "Production, Growth and Business Cycles: I & II," Journal of Monetary Economics, 21, pp.195-232, 309-341.

Kitamura, Tomiyuki (2010), "Measuring Monetary Policy Under Zero Interest Rates With a Dynamic Stochastic General Equilibrium Model: An Application of a Particle Filter," 日本銀行ワーキングペーパーシリーズ 10-E-10.

Sudo, Nao (2012), "Financial Markets, Monetary Policy and Reference Rates: Assessments in DSGE Framework," 日本銀行ワーキングペーパーシリーズ 12-E-12.

Sugo, Tomohiro and Kozo Ueda (2007), "Estimating a DSGE Model for Japan: Evaluating and Modifying a CEE/SW/LOWW Model," 日本銀行ワーキングペーパーシリーズ 07-E-2.

Yano, Koiti (2009), "Dynamic Stochastic General Equilibrium Models Under a Liquidity Trap and Self-organizing State Space Modeling," ESRI Discussion Paper Series No.206.

一上響・北村冨行・小島早都子・代田豊一郎・中村康治・原尚子（2009），「ハイブリッド型日本経済モデル：Quarterly-Japanese Economic Model (Q-JEM)」，日本銀行ワーキングペーパーシリーズ 09-J-6.

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.

佐藤綾野（2009），「各国中央銀行のマクロ計量モデルサーベイ～FPS と JEM の比較を中心として」，ESRI Discussion Paper Series No.211.

笛木琢治・福永一郎（2011），「Medium-scale Japanese Economic Model (M-JEM)：中規模動学的一般均衡モデルの開発状況と活用例」，日本銀行ワーキングペーパーシリーズ 11-J-8.

福永一郎・原尚子・小島早都子・上野陽一・米山俊一（2011），「Quarterly Japanese Economic Model (Q-JEM)：2011 年バージョン」，日本銀行ワーキングペーパーシリーズ 11-E-11.

矢野浩一（2008），「DYNARE による動学的確率的一般均衡シミュレーション～新ケインズ派マクロ経済モデルへの応用～」，ESRI Discussion Paper Series No.203.