

## 経済変動の動学的確率的一般均衡モデル(2)

—Octave/Dynare によるカリブレーション—

増山 幸一

### 1. 始めに

近年、動学的確率的一般均衡モデル (Dynamic Stochastic General Equilibrium model : DSGE モデル) のカリブレーションに基づいて現実経済の動的変動を説明しようとする研究が急進展してきた。DSGE モデルは、Kydland and Prescott(1982) や Hansen(1985) などの研究で代表される実物的景気循環モデル (Real Business Cycle Models) を基礎としている。また、DSGE モデルの数学的手法は動的計画法と呼ばれる最適化手法に大きく依存している<sup>(1)</sup>。

また、新ケインズ派の経済学者たちは、実物的景気循環モデルに市場の失敗や名目変数の硬直性を導入した DSGE モデルを用いて、経済変動の要因分析や経済政策の効果を分析するようになった。代表的な研究として、Galí(1999) , Galí, Gertler and Lopez-Salido(2001), Woodford(2003), Dotsey and King(2005), Christiano, Eichenbaum and Evans(2005), Smets and Wouters(2007), Gertler and Leahy(2008) などがある。DSGE モデルを用いて金融政策の効果等を分析した研究として、Carlstrom and Fuerst(1997) や Gertler, Gilchrist and Natalucci(2007), Edge, et al.(2008) 等がある。マクロ経済動学や金融論における研究の最先端フロンティアでは、DSGE モデルを自由自在に活用することが不可欠となってきた<sup>(2)</sup>。

DSGE モデルのカリブレーションの計算では、従来、有料ソフトである MATLAB などが主として活用されてきた。例えば、Ljungqvist and Sargent(2004), McCandless(2008), Walsh(2010) などの大学院生向けテキストでは、MATLAB のコードで書かれたプログラムを用いて計算されたカリブレーション結果が示されており、使用された MATLAB プログラム・ファイルがホームページ上で公開されていて、誰でも自由に活用できるようになっている。また、DSGE モデルのカリブレーションの計算で必要不可欠な非線形モデルの線形化を自動的に処理してくれるソフトである Dynare が、フランスの研究機関 (CEPREMAP) で開発されている。このソフトを用いれば、経済モデルの線形化計算が自動的に処理され、MATLAB 上で線形化したモデルを自動的にカリブレーションできる。しかし、商業用ソフトは維持費が高価なこともあり、

---

(1) 動的計画法を用いた経済モデルの構築は Sargent(1987), Stokey and Lucas(1989), Ljungqvist and Sargent(2004) などで詳細に説明されている。

(2) 日本のマクロ経済学者の間では、DSGE モデルを積極的に活用した研究は不活発で少ないが、加藤(2007) や矢野(2008) などは数少ない代表例である。

最近では、MATLAB と代替可能なフリー・ソフトである GNU Octave が活用されている<sup>(3)</sup>。

本稿は、DSGE モデルのカリブレーションをフリー・ソフトの Octave を用いて行うことを説明する。第 2 節で、Octave を活用するために最低限要求されるプログラミングの方法を説明する。実際の例としてソロー・タイプの経済成長モデルを取り上げ、外的攪乱を伴う経済成長のシミュレーションを実行してみる。第 3 節で、DSGE モデルのカリブレーションに必要な対数線形化の方法と連立差分方程式の解法について説明する。特に、Uhlig (1999) によって提案された対数線形化の手法を詳細に説明する。第 4 節では、DSGE モデルの出発点となった Hansen モデルを取り上げて、Octave プログラムを具体的にコード化し、カリブレーションを実行してみる。第 5 節で、Dynare の利用法について、説明し、Dynare を具体的に使用したマクロ経済モデルのカリブレーションを取り上げる。第 6 節で、幾つかの重要な金融モデルのカリブレーションを取り上げる。

## 2. GNU Octave を用いた数値計算

DSGE モデルのカリブレーションの計算とその結果の表示にフリー・ソフト（オープン・ソース）の GNU Octave を採用するので、Octave の利用法について、必要最小限の範囲で説明する。Octave のソースコードは、<http://www.gnu.org/software/octave/> からダウンロードできる。Windows 上でインストールするときは、Windows 用インストーラー、例えば、Octave-3.2.4\_i686-pc-mingw32\_gcc-4.4.0\_setup をダウンロードする<sup>(4)</sup>。通常、Octave は default で、C:\program files\Octave3.2.4 にインストールされるが、ファイル構成の便宜上、著者は C:\Octave にインストールした。Octave の実行ファイルは C:\Octave\3.2.4\_gcc-4.4.0 というディレクトリーの下に置かれている。

### 2.1 Octave の起動とコマンド入力

Octave を起動すると、

GNU Octave, version 3.2.4

(3) Octave の WEB サイトでは、以下のように説明されている。GNU Octave is a high-level interpreted language, primarily intended for numerical computations. It provides capabilities for the numerical solution of linear and nonlinear problems, and for performing other numerical experiments. It also provides extensive graphics capabilities for data visualization and manipulation. Octave is normally used through its interactive command line interface, but it can also be used to write noninteractive programs. The Octave language is quite similar to Matlab so that most programs are easily portable. MATLAB を前提として開発された Dynare を GNU Octave 上で走らせることができる。モデルのカリブレーションの実行には何も問題は伴わないが、カリブレーション結果を示すグラフを EPS ファイルに出力する命令が正しく実行されないで、若干の注意が必要である。

(4) 2011年5月現在での最新 version は Octave-3.2.4 である。

```

Copyright (C) 2009 John W. Eaton and others.
This is free software; see the source code for copying conditions.
There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or
FITNESS FOR A PARTICULAR PURPOSE. For details, type 'warranty'.
Octave was configured for " i686-pc-mingw32".
Additional information about Octave is available at http://www.octave.org.
Please contribute if you find this software useful.
For more information, visit http://www.octave.org/help-wanted.html
Report bugs to <bug@octave.org> (but first, please read
http://www.octave.org/bugs.html to learn how to write a helpful report).
For information about changes from previous versions, type 'news'.

octave-3.2.4.exe:1>

```

という画面が表示される。最終行は Octave プロンプトと呼ばれるもので、Octave のコマンド類を入力するところである。この画面上のプロンプト (>) の後に命令、関数および文などを記述して実行する。

変数名は英文字と数字から構成されるが、アンダースコアで英文字をつなげた変数名も利用できる。変数名の長さに制限はない。変数名に文字列や数値を割り当てることができる。簡単な加減乗除では、加算、引き算、掛け算、割り算、累乗は

```
x+y, x-y, x * y, x/y, x^y
```

のように記述する。画面上で、プロンプト > に続けて、

```
octave-3.2.4.exe:1> x = 1.25
```

と入力すると、

```
x = 1.2500
```

と返される。

```
octave-3.2.4.exe:2> x
```

と入力すると、次行に

```
x=1.2500
```

と x の値がプリントされて、返される。

```
octave-3.2.4.exe:3> y=1+2-3*4/5
```

と入力すると

```
y=0.60000
```

と答えが返る。プロンプト画面上に計算の結果を毎回表示する必要がないときは、コマンドの最後にセミコロンを入力する。例えば、

```
octave-3.2.4.exe:4 > z=y^2;
```

と入力すると、 $z$ の値は計算されて保存するが、計算結果を画面上に表示しない。

Octave (MATLAB) には、通常の数学的関数、例えば、三角関数、指数・対数関数、多項式の解法・微分、統計計算、などのパッケージが組み込まれていて、簡単に利用できる<sup>(5)</sup>。例えば、 $\sin(\pi/2)$  の計算は

```
octave-3.2.4.exe:4> z=sin(pi/2);
```

とすれば、 $\sin(\pi/2)$  が計算されて、変数  $z$  の値として保存される。コマンドラインで  $\pi$  と入力すると、 $\pi$  の値が自動的に呼び出される。

さらに、線形代数、確率分布、高速フーリエ変換、微分方程式の解、最適化計算、信号処理などのモジュールも組み込まれている。

## 2.2 行列の計算

行列計算に必要ないくつかの主要な方法について説明する。行列の積などの計算をするためには、数値をマトリックスの形式で表現する必要がある。2 × 2 行列、例えば、行列

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

を入力しよう。Octave のプロンプトに続けて

```
octave-3.2.4.exe:1> A=[1,2;3,4]
```

と各行ごとにセミコロンで区切って入力すると、

$$A = \\ 1 \quad 2 \\ 3 \quad 4$$

と 2 × 2 行列が作成できる。MATLAB と同じように、各行の要素をコンマの代わりにスペースを用いて区切ることもできる。

```
octave-3.2.4.exe:1> A=[1 2; 3 4]
```

と入力しても同じ結果となる。1 回の入力ごとに結果を画面に印字させたくないときは、入力文の最後にセミコロンを付けて

```
octave-3.2.4.exe:2> B=[1, 1;0, 1];
```

```
octave-3.2.4.exe:3> C=[2, 0; 3, 4];
```

と入力すると、結果は画面に表示されない。行列 B, C は

---

(5) 三角関数としては、(引数がラジアン表示で)  $\sin$ ,  $\cos$ ,  $\tan$ ; 逆三角関数としては、 $\text{asin}$ ,  $\text{acos}$ ,  $\text{atan}$  がある。自然対数関数は  $\log$ , 常用対数は  $\log_{10}$ , 指数関数は  $\exp$ , 絶対値は  $\text{abs}$  を用いて計算できる。 $\pi$  は  $\text{pi}$ , 自然対数の底は  $e$  (Euler's number), 複素数表示は  $i$  また  $j (= \sqrt{-1})$ , 無限大は  $\text{inf}$  と表現される。

$$B = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \quad C = \begin{bmatrix} 2 & 0 \\ 3 & 4 \end{bmatrix}$$

となっている。行列の和は

```
octave-3.24.exe:4> B + C
```

と入力して計算すると、この答えは、

$$\begin{bmatrix} 3 & 1 \\ 3 & 5 \end{bmatrix}$$

となる。行列の積を計算することも簡単にできる。例えば、

```
octave-3.24.exe:5> B * C
```

と入力すると、答えは

$$\begin{bmatrix} 5 & 4 \\ 3 & 4 \end{bmatrix}$$

となる。例えば、

```
octave-3.24.exe:6> D = [1, 2, 3; 5, 6, 7];
```

```
octave-3.24.exe:7> E = [1, 2, 3; 5, 6, 7; 1, 2, 3];
```

と入力する。 $D$ 、 $E$ は以下の行列として定義されている。

$$D = \begin{bmatrix} 1 & 2 & 3 \\ 5 & 6 & 7 \end{bmatrix} \quad E = \begin{bmatrix} 1 & 2 & 3 \\ 5 & 6 & 7 \\ 1 & 2 & 3 \end{bmatrix}.$$

行列 $D$ と行列 $E$ の積を計算するために

```
octave-3.24.exe:8> F=D * E
```

と入力する。この結果、

$$F = \begin{bmatrix} 14 & 20 & 26 \\ 42 & 60 & 78 \end{bmatrix}$$

と正しく計算できていることが分かる。ベクトルは行列の特殊ケースであるから、例えば、

```
octave-3.24.exe:9 > y = [1,2,3];
```

とすれば、1行3列マトリックス、つまり、横ベクトル $y$ が定義できる。縦ベクトルで定義するときは、

```
octave-3.24.exe:10 > y = [1;2;3];
```

とすれば、3行1列のマトリックスが作成できる。また、1から10までの整数の集合を作成す

るには,

```
octave-3.2.4.exe:11> z = 1:10;
```

と入力すればよい。初項 1 で定差 2 の等差数列を定義したいとき、例えば、最終項が 21 になるようにするには

```
octave-3.2.4.exe:12> z = 1:2:21;
```

と入力する。文字列を定義したいときには,

```
octave-3.2.4.exe:13> w = 'Hello World';
```

というように、引用符で囲んで定義する。

共役転置行列の計算も簡単にできる。行列 B の転置を計算するときは

```
octave-3.2.4.exe:12> B'
```

と入力すると、転置された行列

$$\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$$

が返される<sup>(6)</sup>。  $Ax = u$  という連立方程式を解くことを考える。

```
octave-3.2.4.exe:1> A=pascal(3);
```

と入力すると,

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \\ 1 & 3 & 6 \end{bmatrix}$$

となっている。さらに,

```
octave-3.2.4.exe:2> u=[3;1;4];
```

とおく。こう定義された連立方程式の解は  $x = A^{-1}u$  なので,

```
octave-3.2.4.exe:3> x=inv(A) * u
```

と入力すると,

$$x = \begin{bmatrix} 10 \\ -12 \\ 5 \end{bmatrix}$$

という解が得られる。ここで、 $\text{inv}(A)$  はマトリックス  $A$  の逆行列を生成する組込み関数である。

(6) Octave 利用のためのマニュアル類は、<http://www.gnu.org/software/octave/docs.html> で提供されており、部分的にはあられ、日本語での解説もあります。ビギナーにとっては、[http://en.wikibooks.org/wiki/Octave\\_Programming\\_Tutorial](http://en.wikibooks.org/wiki/Octave_Programming_Tutorial) にある getting started が参考になる。英語のマニュアルは、Octave をインストールすると、Octave の下のディレクトリー内に収められる。MATLAB のマニュアルも参考になる。本稿の付録にも簡単な組込み関数の一覧表を載せている。

逆行列を使用しないで解を求めるときは

```
octave-3.2.4.exe:3> x=A\u
```

とおくと、上と同じ計算をして、解が得られる。固有値を計算する組み込み関数は `eig(A)` であるが、固有ベクトルも表示させたいときは、

```
octave-3.2.4.exe:4> [v,lambda]=eig(A)
```

と入力すると、`v` に固有ベクトル、`lambda` に固有値を返してくれる。

### 2.3 スクリプト・ファイル

プログラム言語を用いて何かの処理をしたいとき、毎回の処理ごとに画面上のプロンプトにコマンドを入力していたのでは、大変な労力が必要となる。これを避けるために、Octave の文法ルールに従って、多数の命令の集まりからなるプログラムを作成し、これを保存して、必要となるごとに呼び出して利用する。Octave 上で実行できるプログラムが書かれたソースコードをスクリプト・ファイルという。このスクリプト・ファイルを MATLAB 用の拡張子 (`.m`) を付けたファイルとして保存して、それを呼び出して実行する。テキスト・エディターでスクリプト・ファイルを記述して、それを拡張子 `.m` を付けて保存する。スクリプト・ファイルの保存先は任意であるが、Octave のホームディレクトリの下、あるいはユーザーのドキュメント・ファイルの中に保存しておくのが便利であろう。ここでは、スクリプト・ファイルはユーザーのドキュメント・ファイル

`C:\Users\Koichi\Documents\Octave` (ここで、Koichi は私のユーザー名です) の中に保存しておくとする。保存されたスクリプト・ファイルを実行するには、Octave を起動して、スクリプト・ファイル名を入力する。例えば、上記のフォルダー内に保存された `sample` という名前のスクリプト・ファイルを実行したいときは、起動画面上で

```
octave-3.2.4.exe:1> sample
```

と入力する。Octave が自動的に計算結果を返してくれる。終了するときには、`quit` と入力する。

ところで、Octave でスクリプト・ファイルを実行するためには、以下のようなパスの設定をする必要がある。すなわち、スクリプト・ファイルが置かれているディレクトリーにパスを通しておかなければならない。スクリプト・ファイルはユーザーのドキュメント・ファイル `C:\Users\Koichi\Documents\Octave` に保存する予定なので、Octave の `octaverc` ファイル (startup file) 内に、

```
*****
addpath C:\Users\Koichi\Documents\Octave
*****
```

と、1行を追加して書く必要がある<sup>(7)</sup>。なお、Octave を実行中に異なるホルダー、例えば、`C:\Octave\DSGE_holder` に保存しているファイル `DSGE1.m` を実行する必要がある場合、Octave のシェル・プロンプト上で、

```
octave-3.2.4.exe:1> cd C:\Octave\DSGE_holder
```

と入力し、カレント・ディレクトリを変更してから、

```
octave-3.2.4.exe:2> DSGE1
```

と入力して、スクリプト・ファイルを実行させる。このようにカレント・ディレクトリを変更させれば、任意のホルダーに保存されている Octave のスクリプト・ファイルを実行することができる。

ユーザー自身が記述した関数を、組込み関数と同じように、複数のスクリプト・ファイルの中で使用することができる。関数をスクリプト・ファイルで作成し、それを関数名と同一名のスクリプト・ファイルとして保存する。関数名 `function_name` の関数を定義するには

```
*****
```

```
function x = function_name(arg-list)
```

```
function-body
```

```
*****
```

のように記述する。function-body には、関数を定義する文が記述され、`x` の値として返すような手続きが記述される。この関数を用いた計算をするスクリプト・ファイルからは、引数リスト (`arg-list`) の値を具体的に与えて関数名 (`function_name`) を呼び出すと、引数値に対応する関数値 `x` が返される。以下にあるようなインパルス応答関数を計算する関数スクリプト・ファイルをテキスト・エディターを用いて作成する。

```
*****
```

```
function y=impulse(A,B,n,m,tn)
```

```
% computing the impulse response function for the dynamical system:
```

```
%  $y(t) = A y(t-1) + B z(t)$ 
```

```
% A:  $n \times n$  matrix, B:  $n \times m$  matrix (set  $m=1$ )
```

```
% the shock  $z(t)$  is i.i.d.
```

```
% tn=the length of time for the impulse function
```

```
y=zeros(n,tn);
```

```
z=[1.0 zeros(1,tn-1)]';
```

```
for i=1:n
```

```
  y(i,1)=B(i,1) * z(1);
```

```
end
```

```
for i=1:tn-1
```

- 
- (7) スクリプト・ファイルはディレクトリー C:\octave の下のホルダーに保存しても便利です。この場合は、パスをこのディレクトリーに通してください。octaverc file は、C:\octave\3.2.4\_gcc-4.4.0\share\octave\3.2.4\m\startup というホルダー内にあります。好みのテキスト・エディターを用いて、新しい 1 行を書き込んだのち、上書き保存してください。

```
y(:,i+1)=A * y(:,i)+ B(:,1) * z(i+1);
end
```

\*\*\*\*\*

(A,B,n,m,tn) が関数 impulse の値を計算するための引数である。zeros(n,tn) はすべての要素がゼロである n 行 tn 列マトリックスを作成する組込み関数である。% はコメントのための記号 (# 記号を用いてもよい) で、その行内に書かれた内容はスキップされて、次の行から実行される。for i=1:n ... end 文は、for から end までの間に記述された命令あるいは関数等を、i が 1 から n になるまで繰り返し実行させる条件付き命令文である。この関数スクリプトを、ファイル名 impulse.m を付けて、C:\Users\Koichi\Documents\Octave に保存して、以下のようなメイン・スクリプトから呼び出して使用することができる。

\*\*\*\*\*

```
% main program computing the impulse response function
% call impulse(A,B,n,m,timpulse)
% A: n×n matrix, B: n×m matrix(m=1)
% the exogenous shock is one dimensional i.i.d.
% timpulse is the length of time for computation
A=[0.9418, 1.9667; 0, .95];
B=[0; 1];
n=2;
m=1;
timpulse=100;
y=impulse(A,B,n,m,timpulse);
xx=[1:timpulse];
plot(xx,y(1,:),'b+')
axis([0 timpulse -Inf Inf]);
title('Impulse Response')
```

\*\*\*\*\*

このメイン・スクリプトを example.m というファイル名で同じように、C:\Users\Koichi\Documents\Octave に保存する。次に Octave の起動画面上で

```
octave-3.24.exe:1> example
```

と入力して、このプログラムを実行させると次ページにあるグラフが描かれる。グラフ表示の方法に関しては、後で詳細に説明する。

メイン・スクリプトの中で関数を定義して利用するときは、

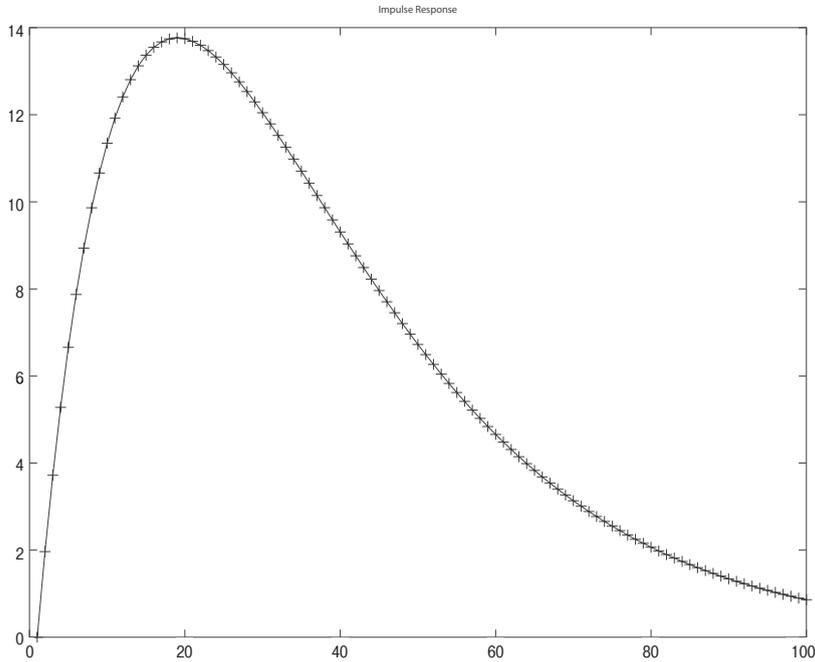
\*\*\*\*\*

```
function y = function_name(arg-list)
```

```
function-body
endfunction
```

```
*****
```

のように定義分の最後に endfunction 文を付ける。この関数定義文をプログラムの最初に置いておけば、この後に記述されているプログラム中で繰り返し使用することができる。



インパルス応答関数のグラフ

#### 2.4 グラフィックスの取り扱い

Octave のすべてのプロット関数はフリーのグラフィックス・ソフトである Gnuplot を用いている。Octave のインストールと同時に Gnuplot もインストールされる。したがって、Octave の起動と同時に、Gnuplot の関数も利用可能となる。2次元プロットの例として、例えば、sin 関数の波形をプロットしてみよう。コマンド linspace(start, stop, length) は数直線区間 [start, stop] を等間隔に分割してできる数値からなるベクトルを作成する。length はこの数値の総数を定めるが、指定しなければ default で100個にする。例えば、

```
octave:1> x = linspace(0, 2 * pi, 200);
```

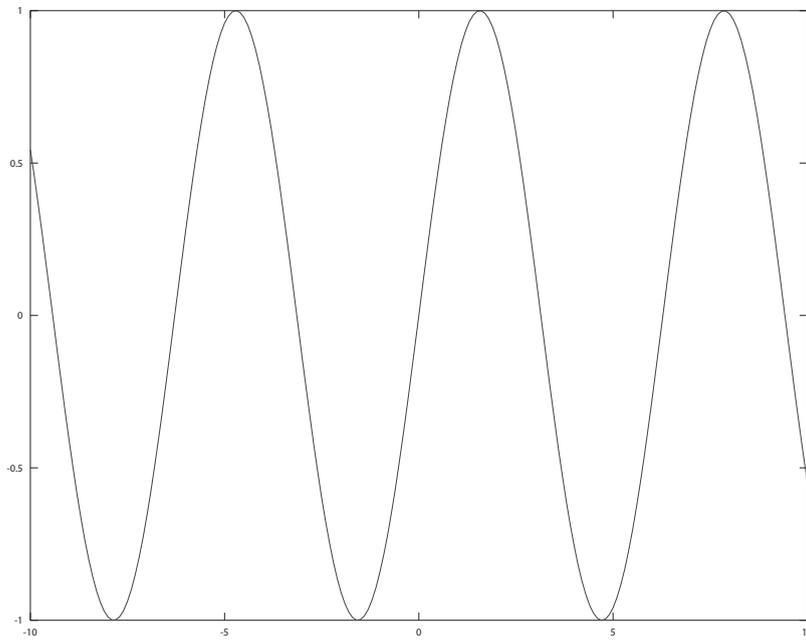
とすれば、変数 x は区間  $[0, 2\pi]$  を均等に分割した200個の数値からなるベクトルである。また、以下のような操作もできる。

```
octave:2> x = -10:0.1:10;
```

と入力すると、変数 x は、区間  $[-10, 10]$  を増分0.1で分割した時の値からなるベクトルを定める。

```
octave:3> plot(x, sin(x))
```

と入力すると、横軸に x、縦軸に  $\sin(x)$  の値をプロットしたグラフが描ける。



Gnuplot によるグラフの表示

描いたグラフをフルカラーの eps 画像ファイルとして保存したいとき

```
print -depsc2 fig.eps
```

と入力すると、fig という名前の eps ファイルが Octave の current directory 内に作成される<sup>(8)</sup>。

-depsc2は画像ファイルを eps 画像形式で、フルカラーで保存するためのオプションである。

以下のようなスクリプト・ファイルを作成する。

```
*****
```

```
# 三角関数のグラフ
```

```
x = linspace(0, 2 * pi);
```

```
a = cos(2 * x);
```

```
b = sin(4 * x);
```

```
c = 2 * sin(x);
```

```
figure;
```

```
plot(x, a);
```

```
hold on;
```

```
plot(x, b);
```

```
plot(x, c) ;
```

```
*****
```

---

(8) current directory は実行中のスクリプト・ファイルが置かれているディレクトリーになっている。

このスクリプト・ファイルを実行すると、同一平面上に、 $a, b, c$  の 3 曲線を描ける。figure というコマンドは、新しいウィンドウを作成して、グラフを描かせる命令である。既に描いたグラフのウィンドウ上の座標平面に曲線等を重ねて描かせたい場合には、hold on コマンドを使用する。既に描いたグラフを消して、重ね描きをさせないためには、hold off コマンドを使用する。同じグラフは、以下のように入力しても描ける。

```
octave:2> plot(x, a, x, b, x, c);
```

ただし、後者のケースでは、 $a, b, c$  の曲線は異なる色で描かれている。

デスクトップの画面上に表示するための命令として、disp 関数、fdisp 関数、および printf 関数が利用可能である。使用例として、

```
octave:3> s = disp(log(10))
```

とコマンド入力すると、

```
s = 2.30258509299405
```

と画面に表示される。あるいは

```
octave:4> x = 10.1;
```

```
octave:5> y = 5.5;
```

```
octave:6> z = 'test';
```

```
octave:7> printf('An integer: %i. A real: %f. This is a %s.\n', x, y, z);
```

と入力すると

```
An integer: 10. A real: 5.500000. This is a test.
```

と画面に印字される。

## 2. 5 経済モデルの例

ここで、経済モデルのシミュレーションの例を取り上げる。ソロータイプの経済成長モデルを考えよう。

生産関数は

$$Y_t = A_t F(K_t, H_t),$$

で与えられる。ここで、 $Y_t$  は時刻  $t$  における経済全体の総生産量、 $A_t$  は時刻  $t$  での生産技術水準、 $K_t$  は時刻  $t$  での資本ストック、 $H_t$  は生産に投入された労働量を表現する。生産関数は一次同次であると仮定する。総生産量を労働者一人当たりに変換すると、

$$y_t = \frac{Y_t}{H_t} = A_t F\left(\frac{K_t}{H_t}, \frac{H_t}{H_t}\right) = A_t F(k_t, 1) \equiv A_t f(k_t),$$

となる。ここで、 $k_t$  は一人当たり資本ストックである。労働力が一定の成長率  $n$  で拡大すると仮定するならば、 $H_{t+1} = (1+n)H_t$  が成立する。また、資本ストックは蓄積方程式

$$K_{t+1} = (1 - \delta)K_t + I_t$$

に従って成長する。  $\delta$  は減価償却率,  $I_t$  は時刻  $t$  での粗投資額である。この式から, 一人当たり資本の蓄積は

$$k_{t+1} = \frac{(1 - \delta)k_t + i_t}{1 + n}$$

に従う。ここで,  $i_t$  は労働者一人当たり粗投資である。貯蓄が生産量の一定割合で行われると仮定できるとする。つまり,  $s_t = \sigma y_t$  と仮定する。  $\sigma$  は労働者一人当たりの貯蓄率と理解される。  $i_t = s_t$  を資本の蓄積方程式に代入すると,

$$k_{t+1} = \frac{(1 - \delta)k_t + \sigma A_t f(k_t)}{1 + n}$$

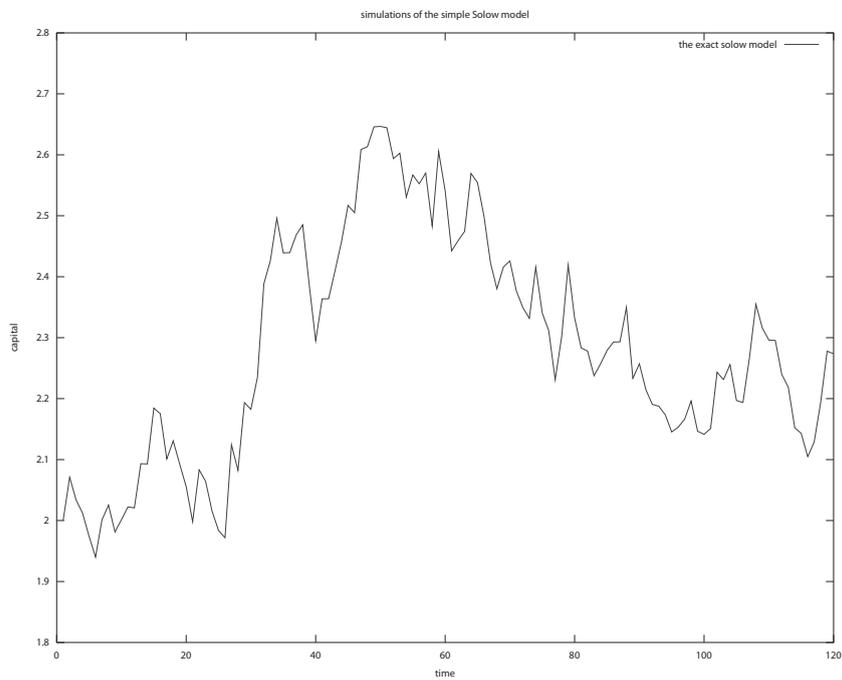
が得られる。

生産関数を  $f(k_t) = k_t^\theta$  と簡単化する。生産技術へのショックは

$$A_t = \bar{A}e^{\epsilon_t}$$

で表現される。ここで,  $\epsilon_t$  は平均値ゼロの正規分布に従う。このモデルをシミュレーションする。パラメータ値として,  $\bar{A} = 1, n = 0.02, \delta = 0.1, \theta = 0.36, \sigma = 0.2$ を採用して, Octaveでシミュレーションした。シミュレーションのためのスクリプト・ファイルは以下のとおりである。

```
*****
% The simulations of the simple Solow model
% the production function f(K)=A K^ theta
% the expression for the technology shock = exp(epsilon_t): the standard error =volat
% the rate of population growth = n
% the depreciation rate = delta
% the saving rate = sigma
T=120;
delta=0.1;
A=1.0;
sigma=0.2;
n=0.02;
theta=0.36;
volat=0.2;
B=(1+theta * n-delta * (1-theta))/(1+n);
```



ソロー・モデルのシミュレーション

```

C=(delta+n)/(1+n);
X_ss=((n+delta)/(sigma * A))^(1/(theta-1));
K(1)=2.0;
X(1)=2.0;
k(1)=0.0;
for t=1:T-1
    R(t)=volat * randn(1);
    K(t+1)=((1-delta) * K(t)+sigma * A * exp(R(t)) * K(t)^theta)/(1+n);
    k(t+1)=B * k(t)+C * R(t);
    X(t+1)=X_ss * exp(k(t+1));
end
hold off;
plot(K)
xlabel('time')
ylabel('capital')
title('simulations of the simple Solow model')
legend('the exact solow model')

```

\*\*\*\*\*

% ... は通常用いられているスクリプトの書き出しである。% はコメントが開始されることを示す記号である。

シミュレーション時間は120期間で、 $\epsilon_t$ の標準偏差は0.2である。定常状態での生産量は2.221となっている。

### 3. Uhlig による対数線形化手法と線形モデルの解法

Uhlig(1999) が導入した対数線形化の方法を説明する<sup>(9)</sup>。確率過程  $X_t$  の対数線形化を考える。 $X_t$  の定常過程を  $\bar{X}$  とするとき、

$$x_t = \ln X_t - \ln \bar{X}$$

と定義される  $x_t$  は  $X_t$  の定常過程  $\bar{X}$  からの対数乖離（差分）の大きさを表す。表現を変えれば、

$$X_t = \bar{X} e^{x_t}$$

と書き直すことができる。 $e^{x_t}$  を原点  $x_t = 0$  の回りでテイラー級数展開すると、

$$X_t = \bar{X} [e^0 + x_t + \frac{1}{2}(x_t)^2 + \dots] = \bar{X} (1 + x_t + \frac{x_t^2}{2} + \dots)$$

が得られる。 $x_t$  が非常に小さい場合を想定しているので、通常、第2次以降の項は無視して、第1次の項までで近似する。

$$X_t \approx \bar{X} (1 + x_t)$$

と表現できる。

$$W_t = e^{ax_t} \text{ と表現される場合,}$$

$$w_t = \ln W_t - \ln \bar{W} = a(X_t - \bar{X})$$

なので、 $x_t = X_t - \bar{X}$ 、 $\bar{W} = e^{a\bar{X}}$  とおくと

$$W_t = \bar{W} e^{ax_t} \approx \bar{W} [1 + ax_t]$$

となる。

$$W_t = X_t(1 + aY_t) \text{ となっている場合, } X_t = \bar{X} e^{x_t}, Y_t = \bar{Y} e^{y_t} \text{ とおけば,}$$

---

(9) Herald Uhlig, A toolkit for analysing nonlinear stochastic models easily, in *Computational Methods for Study of Dynamic Economies*, edited by Ramon Marion and Andrew Scott, 1999, Oxford University Press.

$$W_t = \bar{X}e^{x_t}(1 + a\bar{Y}e^{y_t}) = \bar{X}e^{x_t} + a\bar{X}\bar{Y}e^{x_t+y_t} \approx \bar{X}(1 + x_t) + a\bar{X}\bar{Y}(1 + x_t + y_t)$$

と近似される。さらに、

$$\frac{X_t Y_t^\alpha}{Z_t^\beta}$$

を対数乖離変数を用いて線形近似すると、

$$\frac{X_t Y_t^\alpha}{Z_t^\beta} = \frac{\bar{X}e^{x_t}\bar{Y}^\alpha e^{\alpha y_t}}{\bar{Z}^\beta e^{\beta z_t}} = \frac{\bar{X}\bar{Y}^\alpha}{\bar{Z}^\beta} e^{x_t + \alpha y_t - \beta z_t} \approx \frac{\bar{X}\bar{Y}^\alpha}{\bar{Z}^\beta} (1 + x_t + \alpha y_t - \beta z_t)$$

となる。ここで、 $z_t = \bar{Z}e^{z_t}$ とおいた。

### 例 3.1

コブダグラス型生産関数  $Y_t = \Lambda_t K_t^\theta H_t^{1-\theta}$  を考える。ここで、 $\Lambda_t = \bar{\Lambda}e^{\lambda t}$ ,  $\bar{\lambda} = 0$  とする。上記の対数線形化の手法を用いると、

$$\bar{Y}(1 + y_t) \approx \bar{\Lambda}\bar{K}^\theta \bar{H}^{1-\theta} [1 + \lambda_t + \theta k_t + (1 - \theta)h_t]$$

となる。だから、生産量の変動は

$$y_t = \lambda_t + \theta k_t + (1 - \theta)h_t$$

と表現できることになる。なお、 $K_t = \bar{K}e^{k_t}$ ,  $H_t = \bar{H}e^{h_t}$  とおいている。

対数線形化された動的経済システムは、 $x_t$  を ( $m$  次元) 内生変数ベクトル、 $z_t$  を確率的外生変数ベクトルとして、一般的に以下のような形式で表現できる。

$$E_t[Fx_{t+1} + Gx_t + Hx_{t-1} + Lz_{t+1} + Mz_t] = 0, \quad (1)$$

$$z_{t+1} = Nz_t + \epsilon_{t+1}, \quad (2)$$

ただし、 $E_t$  は時刻  $t$  での条件付き期待値オペレータ、 $F, G, H, L, M, N$  は行列係数である。 $\epsilon_{t+1}$  は外的ショックを表す。このシステムの解を、回帰的均衡則

$$x_t = Px_{t-1} + Qz_t$$

として求めることを考える。この表現を(1)に代入すると、

$$[(FP + G)P + H]x_{t-1} + [(FQ + L)N + (FP + G)Q + M]z_t = 0$$

が得られる。 $x_{t-1}$  の係数がゼロとなる条件から

$$FP^2 + GP + H = 0$$

が得られる。さらに、 $z_t$  の係数がゼロとなる条件から、 $Q$  は以下の条件を満たす行列として計算できる：

$$VQ = -\text{vec}(LN + M), \quad (3)$$

$$V = N' \otimes F + I_m \otimes (FP + G). \quad (4)$$

ここで、 $N'$  は  $N$  の転置行列であり、 $\otimes$  はクロネッカー積、 $\text{vec}(A)$  は行列  $A$  を列ごとにタテベクトル化したベクトル、 $I_m$  は内生変数ベクトルの次元 ( $m$ ) をもつ単位行列である。

回帰的均衡則の行列  $P$  の固有値の絶対値が 1 よりも小さければ、このシステムは安定である。また、行列  $V$  が逆行列をもつならば、行列  $Q$  はユニークに定まる。Uhlig(1999) はこの回帰的均衡則の求め方をより容易にする方法を導入した。内生変数を以下のように 2 種類に分割する。 $x_t$  を内生状態変数 ( $m$  次元ベクトル)、 $y_t$  をジャンプ変数 ( $n$  次元ベクトル) とし、 $z_t$  を確率的  
外生変数 ( $k$  次元) とするとき、経済モデルの均衡条件式を

$$Ax_t + Bx_{t-1} + Cy_t + Dz_t = 0, \quad (5)$$

$$E_t[Fx_{t+1} + Gx_t + Hx_{t-1} + Jy_{t+1} + Ky_t + Lz_{t+1} + Mz_t] = 0, \quad (6)$$

$$z_{t+1} = Nz_t + \epsilon_{t+1}, \quad (7)$$

と表現できると仮定する。ここで、 $A$  と  $B$  は  $l \times m$  行列、 $C$  は  $l \times n$  行列 ( $l \geq n$ )、 $\text{rank}(F) = n$ 、 $F$  は  $(m+n-l) \times n$  行列、 $N$  は安定な固有値のみをもつ行列とする。式(5)は  $l$  個の差分方程式から構成されている。

動学的経済モデルの解を以下のような動的な均衡則 (dynamic equilibrium law)

$$x_t = Px_{t-1} + Qz_t, \quad (8)$$

$$y_t = Rx_{t-1} + Sy_{t-1}, \quad (9)$$

として求めることを考える。Uhlig(1999) は  $l = n$  の場合、以下の定理が成立することを証明した。

### 定理 3.1

$l = n$  とする。(5)、(6)、(7)で表現されている経済モデルの解が(8)、(9)で表現される動学的回帰的均衡則として得られるならば、その係数行列は以下のように計算できる。

(1) $P$  は 2 次方程式 (リカッチ方程式)

$$(F - JC^{-1}A)P^2 - (JC^{-1}B - G + KC^{-1}A)P - KC^{-1}B + H = 0 \quad (10)$$

を満たす。 $P$ の固有値の絶対値が1よりも小さいときにのみ、(8), (9)で記述されるシステムは安定である。

(2) $R$ は

$$R = -C^{-1}(AP + B)$$

で与えられる。

(3) $Q$ は

$$\begin{aligned} [N' \otimes (F - JC^{-1}A) + I_k \otimes (JR + FP + G - KC^{-1}A)] \text{vec}(Q) \\ = \text{vec}[(JC^{-1}D - L)N + KC^{-1} - M] \end{aligned} \quad (11)$$

を満たす。

(4) $S$ は

$$S = -C^{-1}(AQ + D)$$

によって与えられる。

次に、 $P$ の2次方程式(10)の解法について説明する。(10)式の係数を

$$\Psi = F - JC^{-1}A, \quad (12)$$

$$\Gamma = JC^{-1}B - G + KC^{-1}A, \quad (13)$$

$$\Theta = KC^{-1}B - H \quad (14)$$

と置き換えると、行列 $P$ の2次行列方程式が

$$\Psi P^2 - \Gamma P - \Theta = 0 \quad (15)$$

で与えられる。係数行列はすべて $m \times m$ 次元の行列である。このとき、(15)式の解は $P = \Omega \Lambda \Omega^{-1}$ という形式で表現できることを説明する。ここで、行列 $\Lambda$ は以下で導出する固有値を対角線上にもつ行列、 $\Omega$ はそれに対応する固有ベクトルを列とする行列である。新しい $2m \times 2m$ 行列

$$\Xi = \begin{bmatrix} \Gamma & \Theta \\ I_m & 0_m \end{bmatrix} \quad (16)$$

$$\Delta = \begin{bmatrix} \Psi & 0_m \\ 0_m & I_m \end{bmatrix} \quad (17)$$

を定義する。ここで、 $I_m$ は次数 $m$ の単位行列、 $0_m$ は次数 $m$ のゼロ行列である。

**定理 3.2**

(1)  $\Delta$ に関する $\Xi$ の一般化固有値問題に対する固有ベクトルが $s$ ，対応する固有値が $\lambda$ であるならば，固有ベクトル $s$ は

$$s' = (\lambda x', x'), \text{ for some } x \in R^m$$

なる表現形式をもつ<sup>(10)</sup>。ここで， $s'$  はベクトル $s$ の転置ベクトルであり， $2m$ 次元ヨコ・ベクトルとなっている。

(2) この一般化固有値問題の解として， $m$ 個の固有値 $\lambda_1, \lambda_2, \dots, \lambda_m$ と対応する $m$ 個の固有ベクトル

$$s'_1 = (\lambda_1 x'_1, x'_1), s'_2 = (\lambda_2 x'_2, x'_2), \dots, s'_m = (\lambda_m x'_m, x'_m)$$

が存在するならば，そして，これらの $m$ 個の(固有)ベクトルは線形独立であるならば， $P = \Omega \Lambda \Omega^{-1}$ が2次方程式(15)の解となる。 $x'_j = (x_{j1}, x_{j2}, \dots, x_{jm})'$ ， $j = 1, 2, \dots, m$ と表記する。ただし，

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 & 0 \\ 0 & \lambda_2 & 0 & 0 & 0 \\ \vdots & 0 & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \lambda_{m-1} & 0 \\ 0 & \dots & \dots & 0 & \lambda_m \end{bmatrix}$$

$$\Omega = \begin{bmatrix} x_{11} & x_{21} & \dots & x_{m-1,1} & x_{m,1} \\ x_{12} & x_{22} & \dots & x_{m-1,2} & x_{m,2} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{1,m} & x_{2,m} & \dots & x_{m-1,m} & x_{m,m} \end{bmatrix}$$

である。また，すべての固有値の絶対値が1よりも小さければ，経済システムの解は安定となる。

一般化固有値問題の固有値 $\lambda$ は通常， $2m$ 個存在するので， $2m$ 個の固有値の中から $m$ 個の固有値をどのように選択すべきかという問題が生じる。ここで，上記の $\Lambda, \Omega$ の求め方について説明する。一般化固有値問題の解は

$$\Xi s_j = \Delta s_j \lambda_j, j = 1, 2, \dots, 2m$$

を満たす。この固有値問題に対する固有値は $2m$ 個存在するので，これらを $\lambda_j, j = 1, 2, \dots, 2m$ と表記し， $s_j, j = 1, 2, \dots, 2m$ はそれに対応する固有ベクトルである。ここで，重要な操作は，最初の $m$ 個の固有値が安定であるように，固有値の順番を並び替えることである。このように

(10) この一般化固有値問題の解は $\Xi s_j = \Delta s_j \lambda_j; j = 1, 2, \dots, 2m$ を満たす $2m$ 個の固有値 $\lambda_j$ と固有ベクトル $s_j$ となる。固有ベクトルは次数 $2m \times 1$ のタテ・ベクトルとなる。 $\Delta = I_{2m}$ のとき，通常の固有値問題になる。

並べ替えられた固有値に対応する固有ベクトルを列とする行列を行列  $X$  と表記する。

$$X = \begin{bmatrix} s_{11} & s_{21} & \cdots & s_{2m-1,1} & s_{2m,1} \\ s_{12} & s_{22} & \cdots & s_{2m-1,2} & s_{2m,2} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ s_{1,2m} & s_{2,2m} & \cdots & s_{2m-1,2m} & s_{2m,2m} \end{bmatrix} = \begin{bmatrix} X_{11} & X_{21} \\ X_{12} & X_{22} \end{bmatrix}$$

このとき、行列方程式 (15) の解は

$$P = X_{12} \Lambda X_{12}^{-1} \quad (18)$$

で与えられることが知られている。ただし、 $\Lambda$  は安定な固有値  $\lambda_j, j = 1, 2, \dots, m$  のみを対角要素とする行列

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 & 0 \\ 0 & \lambda_2 & 0 & 0 & 0 \\ \vdots & 0 & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \lambda_{m-1} & 0 \\ 0 & \cdots & \cdots & 0 & \lambda_m \end{bmatrix}$$

であり、 $\Omega \equiv X_{12}$  となっている。

### 定理 3.3

$\Delta$  と  $\Xi$  に関する QZ 分解 (一般化 Schur 分解), つまり,

$$Y' \Sigma Z = \Delta$$

$$Y' \Phi Z = \Xi$$

を満たすユニタリー行列  $Y, Z$  と上半三角行列  $\Sigma, \Phi$  を求めることができるとする。 $Y' \cdot Y = I_{2m}, Z' \cdot Z = I_{2m}$  が成立する。

ここで,

$$Z = \begin{bmatrix} Z_{11} & Z_{21} \\ Z_{12} & Z_{22} \end{bmatrix}$$

と表記されており、 $Z_{ij}$  は  $m \times m$  行列である。ただし、 $Z_{21}$  と  $Z_{22}$  は逆行列が存在するとする。

(i) 行列

$$P = -Z_{21}^{-1} Z_{22}$$

は 2 次行列方程式(15)の解となる。

(2)  $\Phi$  の  $(i, j)$  要素を  $\phi_{ij}$  と表記し、同様に、 $\Sigma$  の  $(i, j)$  要素を  $\sigma_{ij}$  と表記する。比率  $\phi_{ii}/\sigma_{ii}$  が  $i$  と共に単調増加列となっているとする。さらに、 $\phi_{mm}/\sigma_{mm} < 1$  と仮定する。このとき、 $P$  は安定である。つまり、

$$P^n x \rightarrow 0 \quad \text{as } n \rightarrow \infty \quad \text{for any } x \in R^m$$

である。

この QZ 分解を用いて計算する方が一般化固有値問題を解いて  $P$  を求める方法よりも一般性・汎用性をもつ。MATLAB/Octave 等ではこの QZ 分解の組込み関数が利用できる。QZ 分解を用いた行列  $P$  を計算する MATLAB コードが Uhlig によって提供されている。Uhlig のホームページにある Toolkit4 という名称のファイルには、QZ 分解によるリカッチ方程式の解法、インパルス応答関数、スペクトル密度等を計算するスクリプト・ファイルがパッケージになっており、これらのモジュール化された計算コードを利用した 7 種類の経済モデルが提供されている。計算コードは MATLAB 向けに書かれているが、Octave 上でもスムーズに実行できる<sup>(11)</sup>。

#### 4. 簡単な DSGE モデルのカリブレーション

本節では、Octave を用いて、DSGE モデルの出発点となっている Hansen モデルのシミュレーションを実際に行う。代表的家計は生涯にわたる効用

$$E_0[\sum_{t=0}^{\infty} \beta^t u(C_t, H_t)]$$

を最大にするような消費  $\{C_t\}$  と労働供給  $\{H_t\}$  を選択する。レジャー  $L_t$  は  $L_t = 1 - H_t$  で与えられる。言い換えると、家計の有効な使用可能時間を 1 に規格化している。ここで、変数、例えば、 $C_t$  の表現における下添え字  $t$  は時刻を表す。 $E_0$  は初期時点での条件付き期待値オペレータである。生産関数を  $f(\lambda_t, K_t, H_t)$  で表現する。 $\lambda_t$  は時刻  $t$  における生産技術に対する外的なショックの大きさを表し、 $K_t$  は資本投入量、 $H_t$  は労働投入量である。資本ストックは蓄積方程式

$$K_{t+1} = (1 - \delta)K_t + I_t$$

に従う。ここで、 $I_t$  は時刻  $t$  での投資、 $\delta$  は資本の減価償却率である。生産された価値は消費と投資に配分されるので、

(11) [http://www2.wiwi.hu-berlin.de/wpol/html/toolkit/version4\\_1.html](http://www2.wiwi.hu-berlin.de/wpol/html/toolkit/version4_1.html) からダウンロード可能となっている。MATLAB/Octave での QZ 分解では、 $\Sigma = Y' \Delta$ 、 $Z$ 、 $\Phi = Y' \Xi Z$  という形式で計算することから、この計算に基づいて  $P = -(Z')_{21}^{-1} (Z')_{22}$  と計算する必要があるため、注意が必要である。

$$f(\lambda_t, K_t, H_t) = C_t + I_t$$

が成立する。

時刻  $t$  における価値関数を  $V(K_t, \lambda_t)$  と定義すると、Bellman 方程式は

$$V(K_t, \lambda_t) = \max_{\{C_t, H_t\}} [u(C_t, H_t) + \beta E_t[V(K_{t+1}, \lambda_{t+1}) | \lambda_t]]$$

となる。ここで、関係式

$$C_t = f(\lambda_t, K_t, H_t) + (1 - \delta)K_t - K_{t+1}$$

を代入すると、第1階の条件は

$$\begin{aligned} \frac{\partial V(K_t, \lambda_t)}{\partial K_{t+1}} &= -\frac{\partial u(C_t, H_t)}{\partial C_t} + \beta E_t\left[\frac{\partial V(K_{t+1}, \lambda_{t+1})}{\partial K_{t+1}} | \lambda_t\right] = 0, \\ \frac{\partial V(K_t, \lambda_t)}{\partial H_t} &= \frac{\partial u(C_t, H_t)}{\partial C_t} \frac{\partial f(\lambda_t, K_t, H_t)}{\partial H_t} + \frac{\partial u(C_t, H_t)}{\partial H_t} = 0 \end{aligned}$$

である。envelope 定理から、

$$\frac{\partial V(K_t, \lambda_t)}{\partial K_t} = \frac{\partial u(C_t, H_t)}{\partial C_t} \left\{ \frac{\partial f(\lambda_t, K_t, H_t)}{\partial K_t} + (1 - \delta) \right\}$$

が成立する。この関係式を上第1階の条件に代入すると、

$$\frac{\partial u(C_t, H_t)}{\partial C_t} = \beta E_t\left[\frac{\partial u(C_{t+1}, H_{t+1})}{\partial C_{t+1}} \left( \frac{\partial f(\lambda_{t+1}, K_{t+1}, H_{t+1})}{\partial K_{t+1}} + 1 - \delta \right) | \lambda_t\right], \quad (19)$$

$$\frac{\partial u(C_t, H_t)}{\partial H_t} = -\frac{\partial u(C_t, H_t)}{\partial C_t} \frac{\partial f(\lambda_t, K_t, H_t)}{\partial H_t} \quad (20)$$

なる関係式が得られる。式(19)はオイラー方程式と呼ばれているマクロ経済学での基本方程式である。

モデルの数値的な挙動を調べるに当たって

$$Y_t = f(\lambda_t, K_t, H_t) = \lambda_t K_t^\theta H_t^{1-\theta}, \quad u(C_t, H_t) = \ln C_t + A \ln(1 - H_t)$$

と具体的な関数形を与えると、第1階条件は

$$\frac{1}{C_t} = \beta E_t\left[\frac{\theta \lambda_{t+1} K_{t+1}^{\theta-1} H_{t+1}^{1-\theta} + 1 - \delta}{C_{t+1}} | \lambda_t\right], \quad (21)$$

$$A C_t = (1 - \theta)(1 - H_t) \lambda_t k_t^\theta H_t^{-\theta} \quad (22)$$

と簡単化される。また、

$$C_t = \lambda_t K_t^\theta H_t^{1-\theta} + (1 - \delta)K_t - K_{t+1}$$

が成立している。

要素市場が完全競争市場であれば、資本の限界生産は資本のレンタル価格（利子率） $r$  に等しく、労働の限界生産は賃金率  $w$  に等しくなる。

$$r_t = \theta \lambda_t K_t^{\theta-1} H_t^{1-\theta} = \theta \frac{Y_t}{K_t}, \quad w_t = (1 - \theta) \lambda_t K_t^\theta H_t^{-\theta} = (1 - \theta) \frac{Y_t}{H_t}.$$

これらの条件を用いると第1階条件は

$$\frac{1}{C_t} = \beta E_t \left[ \frac{r_{t+1} + (1 - \delta)}{C_{t+1}} \mid \lambda_t \right],$$

$$AC_t = (1 - H_t)(1 - \theta) \frac{Y_t}{H_t}$$

と表現できる。ただし、

$$C_t = Y_t + (1 - \delta)K_t - K_{t+1},$$

$$Y_t = \lambda_t K_t^\theta H_t^{1-\theta},$$

$$r_t = \theta \frac{Y_t}{K_t}$$

である。以上が Hansen モデルの動学システムを記述している。 $\lambda_t = \bar{\lambda} e^{z_t}$  とすると、確率的外的攪乱過程は

$$z_{t+1} = \gamma z_t + \epsilon_{t+1}$$

で表現される。ここで、 $\epsilon_t$  は外生的ショック確率過程である。

定常状態では、 $K_t = K_{t+1} = \bar{K}$ ,  $C_t = C_{t+1} = \bar{C}$ ,  $\lambda_t = \lambda_{t+1} = \bar{\lambda}$  とおけるので、労働供給量と資本ストックの定常状態が計算できる。定常状態における労働供給量  $\bar{H}$  と資本ストック  $\bar{K}$  は

$$\bar{H} = \frac{1}{1 + \frac{A}{1-\theta} \left[ 1 - \frac{\beta \delta \theta}{1 - \beta(1-\delta)} \right]}, \quad \bar{K} = \bar{H} \left[ \frac{\theta \bar{\lambda}}{\frac{1}{\beta} - (1 - \delta)} \right]^{\frac{1}{1-\theta}}$$

と与えられる。また、 $1/\beta = \bar{r} + 1 - \delta$  が成立している。

Uhlig(1999) が提案した方法に従って対数線形化を行う。各変数を

$$C_t = \bar{C} e^{c_t}, \quad Y_t = \bar{Y} e^{y_t}, \quad H_t = \bar{H} e^{h_t}, \quad K_t = \bar{K} e^{k_t}, \quad r_t = \bar{r} e^{r_t}, \quad \lambda_t = \bar{\lambda} e^{z_t}$$

とおくことにする。これらの関係式を用いて、Hansen モデルを対数線形化すれば

$$c_t - E_t[c_{t+1}] + \beta \bar{r} E_t[\tilde{r}_{t+1}] = 0,$$

$$y_t - \frac{1}{1 - \bar{H}} h_t - c_t = 0,$$

$$\bar{Y} y_t - \bar{C} c_t + \bar{K} [(1 - \delta)k_t - k_{t+1}] = 0,$$

$$\begin{aligned} z_t + \theta k_t + (1 - \theta)h_t - y_t &= 0, \\ y_t - k_t - \tilde{r}_t &= 0 \end{aligned}$$

となる。 $\bar{\lambda} = 1$  とおいた。モデルの内生変数は  $[k_{t+1}, y_t, c_t, h_t, \tilde{r}_t]$  の5個であり、外生変数は  $z_t$  である。 $x_t = k_{t+1}$  を状態変数として、 $y_t = [y_t, c_t, h_t, \tilde{r}_t]$  をジャンプ変数とすると、

$$\begin{aligned} Ax_t + Bx_{t-1} + Cy_t + Dz_t &= 0, \\ E_t[Fx_{t+1} + Gx_t + Hx_{t-1} + Jy_{t+1} + Ky_t + Lz_{t+1} + Mz_t] &= 0, \\ z_{t+1} &= Nz_t + \varepsilon_{t+1}, \end{aligned}$$

という表現ができる。ここで、 $A$  と  $B$  は  $4 \times 1$  行列、 $C$  は  $4 \times 4$  行列となっている。

$$\begin{aligned} A &= [0, -\bar{K}, 0, 0]' \\ B &= [0, (1 - \delta)\bar{K}, \theta, -1]' \\ C &= \begin{bmatrix} 1 & -1 & -\frac{1}{1-H} & 0 \\ \bar{Y} & -\bar{C} & 0 & 0 \\ -1 & 0 & 1 - \theta & 0 \\ 1 & 0 & 0 & -1 \end{bmatrix} \\ D &= [0, 0, 1, 0]' \\ F &= [0], G = [0], H = [0] \\ J &= [0, -1, 0, \beta\bar{r}]' \\ K &= [0, 1, 0, 0]' \\ L &= [0], M = [0], N = [\gamma] \end{aligned}$$

以上の Hansen モデルにおいて、

$$\beta = 0.99, \delta = 0.025, \theta = 0.36, \bar{H} = \frac{1}{3}, \gamma = 0.95$$

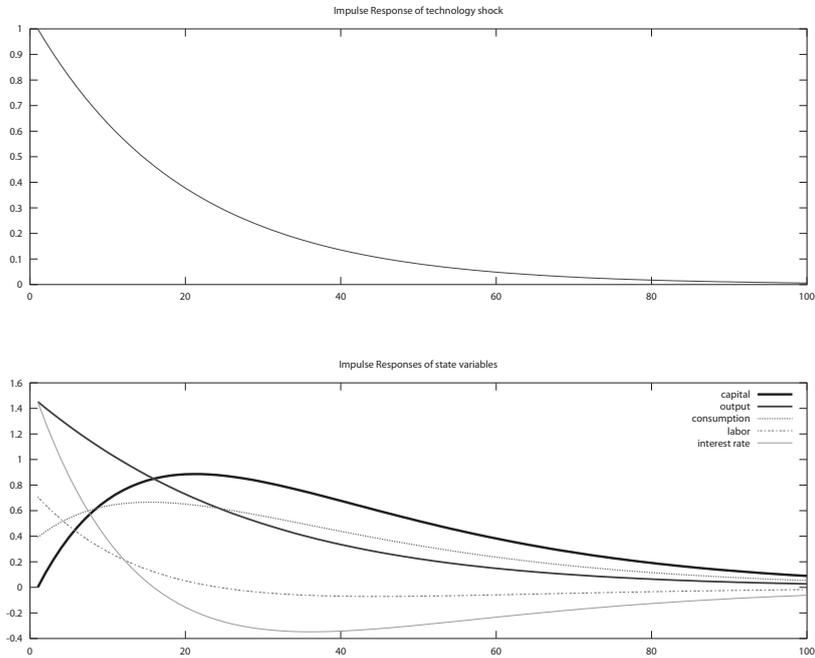
と与える。 $\bar{H} = 1/3$  に対応する  $A$  の値と定常状態での資本ストックの値は

$$A = 1.7214, \bar{K} = 12.670$$

となる。動学的経済モデルの解を動的な均衡則

$$\begin{aligned} x_t &= Px_{t-1} + Qz_t, \\ y_t &= Rx_{t-1} + Sy_{t-1}, \end{aligned}$$

として求める。ただし、 $S = -C^{-1}(AQ + D)$  である。安定な均衡則を求めるとき、得られた結果は



Hansen モデルのシミュレーション

$$P = 0.95367, Q = 0.11320, R = [0.20440, 0.56909, -0.24312, -0.79560]',$$

$$S = [1.45247, 0.39200, 0.70698, 1.45247]'$$

である。この数値のもとでモデルをシミュレーションする。外的な技術ショックが時刻ゼロで起こり、その後のショックはないとして、 $z_0 = 1.0, z_t = 0, t = 1, 2, \dots$  で、シミュレーション期間が100期間のときの結果は上のグラフに示されている<sup>(12)</sup>。

技術ショックに対する各変数の標準偏差の比率は、生産量、消費、労働、利子率のそれぞれについて

$$\sigma_Y = 5.4839, \sigma_C = 4.0633, \sigma_H = 1.6403, \sigma_r = 3.4908$$

となっている。Hansen によれば、米国経済の産出量の標準偏差が0.0176と報告されているので、この産出量の標準偏差になるためには、技術ショックの標準偏差は $0.0176/5.4839=0.0032094$  と予想される。しかし、現実経済で推計された値は区間  $[0.007, 0.01]$  の間にあると指摘されているので、現実の経済でのデータに比べて小さすぎると言える。

(12) 上のグラフ上では、初期時刻は  $t = 1$  に対応する。

## 5. Dynare を用いたマクロ経済モデルのカリブレーション

Dynare を使用するためには、Dynare の Web サイト <http://www.dynare.org/download> からインストール用 (Windows 向け) のファイル `dynare-4.2.0-win.exe` をダウンロードする。Dynare の諸ファイルは、ここでは、`C:\dynare` というディレクトリに保存した。Dynare を Octave 上で利用するためには、`octaverc` file に

```
*****
```

```
addpath C:\dynare\4.2.0\matlab
```

```
*****
```

と 1 行追加して、パスを確保しなければならない。さらに、dynare 用のプログラム・ファイルを実行するために以下の設定をする必要がある。すなわち、プログラム実行の current directory を dynare 用プログラム・ファイルを置いているディレクトリに移動させる必要がある。ディレクトリ `C:\User\Koichi\Documents\Octave\Dynare` に dynare 用のプログラム・ファイル (拡張子 `.mod` 付きのファイル) を置くことにすると、`octaverc` file に

```
*****
```

```
cd C:\User\Koichi\Documents\Octave \Dynare
```

```
*****
```

と 1 行追加する必要がある<sup>(13)</sup>。dynare 用プログラム・ファイル (mod ファイル) を実行するためには、まず、Octave を起動して、dynare filename と入力する。例えば、`sample1` という練習用モデルのシミュレーションを実行させるためには、

```
octave-3.2.4.exe:1> dynare sample1
```

と入力する。計算結果の数値が shell 画面上に表示されると同時に、インパルス応答関数のグラフがデスク・トップ画面に表示される。

Hansen モデルを Dynare を用いてシミュレーションしてみよう。Hansen モデルの動学システムは

$$\frac{1}{C_t} = \beta E_t \left[ \frac{r_{t+1} + 1 - \delta}{C_{t+1}} | \lambda_t \right],$$

$$AC_t = (1 - H_t)(1 - \theta) \frac{Y_t}{H_t}$$

---

(13) Dynare の mod ファイル、例えば、`an example` を実行させるためには、the current directory of Octave を `an example` が置かれているディレクトリへ移動させる必要がある。`an example` を `c:\dynare\examples` に保存したとすれば、`octaverc` ファイルに `cd c:\dynare\examples` と 1 行追加する必要がある。そうしないと、Octave を起動させるたびに、shell 画面上で、毎回このコマンドを入力しなければいけない。

$$\begin{aligned} C_t &= Y_t + (1 - \delta)K_t - K_{t+1}, \\ Y_t &= \lambda_t K_t^\theta H_t^{1-\theta}, \\ r_t &= \theta \frac{Y_t}{K_t} \end{aligned}$$

と表現できる。確率的外的攪乱過程は

$$\log \lambda_{t+1} = \gamma \log \lambda_t + \epsilon_{t+1}$$

で表現される。ここで、 $\epsilon_t$ は外生的ショックの確率過程である。Hansen モデルにおいて、

$$\beta = 0.99, \delta = 0.025, \theta = 0.36, A = 1.72, \gamma = 0.95$$

と与える。このとき、Octave あるいは Dynare を用いて、定常状態の値を計算させることができる。外生変数  $\lambda = 1$  とおくと、以下のような値がえられる。

$$\bar{K} = 12.6698, \bar{H} = 0.33, \bar{C} = 0.918594, \bar{Y} = 1.23534, \bar{w} = 2.3706, \bar{r} = 0.035101,$$

Dynare で実行しようとするモデルの記述は以下のような5つのブロック

Preamble ブロック Model ブロック Initval ブロック Shocks ブロック Computation ブロック

に分けられる<sup>(14)</sup>。Preamble で、変数（内生変数 var，外生変数 varexo）の定義とパラメータ名（parameters）とその値を与える。例えば、

---

```
// comment
var y, c, k, h, r, z;
varexo e;

parameters beta, delta, theta, gamma, AA, hbar, kbar, ybar, cbar, rbar;

beta = 0.99;
delta = 0.025;
theta = 0.36;
```

---

(14) Dynare のマニュアル類やチュートリアルは Dynare の web サイト <http://www.dynare.org/documentation-and-support> で提供されている。さらに、Dynare を用いた様々なマクロモデルの比較研究も行われている。研究プロジェクトのホームページは <http://www.macromodelbase.com/> である。この web サイトでは、Volker Wieland, et. al. (2009) を始めとした幾つかの研究論文が紹介されている。日本での Dynare を用いた代表例は、矢野浩一 (2008) である。

```

gamma = 0.95;
AA = 1.72;
rbar = 1/beta -1 + delta;
hbar = (1-theta) * rbar / (AA * (rbar - delta * theta) + (1-theta) * rbar);
kbar = (theta/rbar) ^ (1/(1-theta)) * hbar;
ybar = kbar ^ theta * hbar ^ (1-theta);
cbar = ybar - delta * kbar;

```

---

と記述する。// はコメントのラインを示す記号である。Model のブロックでは、モデルの方程式を記述する。若干記述の手続きが特殊である。現時点での変数値を  $x$  とすると、1 期間先の変数値は  $x(+1)$  のように、1 期間前の変数値は  $x(-1)$  のように記述する。前節で導出した Hansen モデルの対数線形化モデルを記述すると、

---

```

/*
The following section describes a log-linearized Hansen model.
*/
model(linear);
c = c(+1) - beta * rbar * r(+1);
y = h / (1 - hbar) + c;
kbar * k = ybar * y - cbar * c + kbar * (1 - delta) * k(-1);
y = z + theta * k(-1) + (1 - theta) * h;
y = k(-1) + r;
z = gamma * z(-1) + e;
end;

```

---

となる。/\* .... \*/ はコメントの始まりと終わりを示す記号で、その間にコメントが記述されている。\*/ 以下がモデルの記述となっている。モデルの記述は model; 文から始まり、その最後には必ず、end; 文を書く。初期値の付与は、Initval のブロックで与える。このブロックは initval 文から始まり、end; 文で終わる。

---

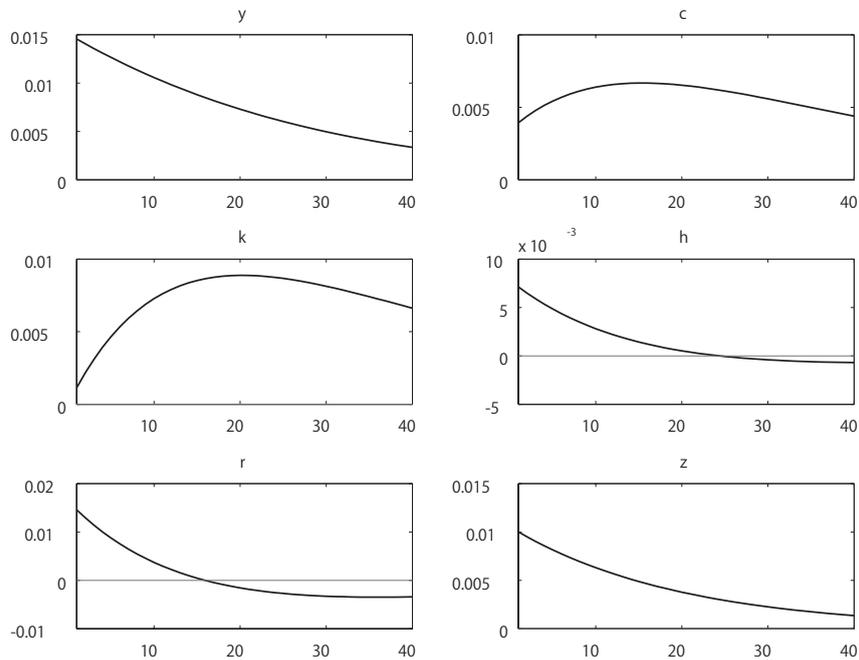
```

initval;
y = 0;

```

---

- (15) 非線形モデルのまま記述して、定常状態の値を求めたいときは、このブロックの下に、steady; という文を挿入する。この場合、初期値として適当な値を付与しておく。steady; 文はモデルの定常状態での値を計算する。



Dynare による Hansen モデルのインパルス応答関数

```
c = 0;
h = 0;
k = 0;
r = 0;
z = 0;
e = 0;
end;
```

線形化しているので、定常状態ではすべての変数値はゼロである<sup>(15)</sup>。外生的ショックの特徴は、shocks ブロックで与える。例えば、

```
shocks;
var e = 0.01^2;
end;
```

(16) GNU Octave のバージョンに依存して、dynare から Octave 経由で出力される EPS ファイルは壊れているので、保存されたグラフを GSview で見るができない。画面上に表示されたグラフを保存する必要がある。この点で、注意が必要である。例として示したグラフは、Latex に組み込むために、Dynare を MATLAB で実行させて、出力した eps ファイルを利用した。

これは、標準偏差が0.01の外生的ショックを記述している。この条件下で、シミュレーションを行いたい時は、

---

```
stoch_simul(periods=2100);
```

---

と書く。これがこの例での computation ブロックを構成する。以上のスクリプト全体を一つのファイルとして、テキスト・エディターで書いて、拡張子 .mod を付けて保存する。このファイルを Dynare で実行すると、前頁に描かれたインパルス応答関数のグラフが得られる<sup>(16)</sup>。

Dynare でのプログラムの記述は非常に簡単であるので、Dynare を用いたカリブレーションは便利である。反面、グラフ等の出力が自動化されているので、ユーザーごとに異なる自由なグラフ出力ができない。また、非線形方程式の線形化を自動的に処理するシステムとなっているが、必ずしもこの線形化の処理がスムーズにいかず、エラーがでることも多い<sup>(17)</sup>。

## 6. 金融論の基礎モデル

### 6.1 MIU モデル

本節では、金融論の基礎モデルとして活用されている cash-in-advance(CIA) モデルと money-in-the-utility function(MIU) モデルを取り上げる。最初に、money-in-the-utility function モデルを取り上げる。このモデルでは、実質貨幣残高の増加が家計の効用水準を上昇させることが前提とされている。実質貨幣残高の増加が家計の効用水準を上昇させる理由として、十分な貨幣を保有することが財やサービスの市場取引にかかる費用を低下させる事実をあげている。時刻  $t$  での家計  $i$  の効用関数は

$$u(c_t^i, \frac{m_t^i}{P_t}, l_t^i) = u(c_t^i, \frac{m_t^i}{P_t}, 1 - h_t^i)$$

で与えられる。ここで、 $c_t^i$  は消費量、 $m_t^i$  は貨幣保有量、 $P_t$  は財価格、 $h_t^i$  は労働供給量である。有効時間を 1 に規格化しているので、レジャー時間は  $1 - h_t^i$  である。各家計を添え字  $i$  でインデックス付ける。マクロ経済を構成する各家計は区間  $[0, 1]$  の間に一様に分布しており、家計の総計は 1 である。よって、 $i \in [0, 1]$  である。各家計  $i$  は期待効用関数の現在割引価値

---

(17) 事実、前節で定式化した Hansen モデルを非線形の形で dynare に記述して実行すると、定常状態の値を計算するまではスムーズにいくが、計算されたヤコビアン行列に基づく計算過程でエラーがでてしまう。

$$E_0[\sum_{t=0}^{\infty} \beta^t u(c_t^i, \frac{m_t^i}{P_t}, 1 - h_t^i)]$$

を最大にするように、消費量  $c_t^i$ 、貨幣保有量  $m_t^i$ 、および、労働供給量  $h_t^i$  を選択する。

マクロ経済の生産関数を

$$y_t = f(\lambda_t, K_t, H_t)$$

と仮定する。 $K_t$  は資本量の経済全体での総計、 $H_t$  は総家計の労働供給量なので、

$$K_t = \int_0^1 k_t^i di, \quad H_t = \int_0^1 h_t^i di$$

と定義される。 $\lambda_t$  は生産技術の水準を表す。要素市場が完全競争市場であれば、資本の限界生産は資本のレンタル価格（利子率） $r$  に等しく、労働の限界生産は賃金率  $w$  に等しくなる。

$$r_t = \frac{\partial f(\lambda_t, K_t, H_t)}{\partial K_t}, \quad w_t = \frac{\partial f(\lambda_t, K_t, H_t)}{\partial H_t}.$$

家計  $i$  は前期から手持ちの貨幣量  $m_{t-1}^i$  を持ち越し、当該期の始めに政府から貨幣の移転を受け取るとする。この政府からの貨幣移転を  $g_t M_{t-1}$  とする。家計  $i$  は期間  $t$  の期首に金融資産として貨幣と債券を保有している。期間  $t-1$  から持ち越した債券  $b_{t-1}^i$  は名目利子率  $i_{t-1}$  で表現される収益を生む。よって、家計  $i$  が期間  $t$  の期首に持ち越す金融資産を  $a_t$  と実質値で表記すると

$$a_t^i = \{m_{t-1}^i + (1 + i_{t-1})b_{t-1}^i + g_t M_{t-1}\} / P_t$$

である。家計は金融資産  $a_t^i$  を保有するだけでなく、前期に蓄積した資本量  $k_t^i$  を持ち越す。期間  $t$  における家計のフローの所得は賃金所得  $w_t h_t^i$  と資本のレンタル料  $r_t k_t^i$  である。よって、家計の予算制約式は

$$c_t^i + k_{t+1}^i + \frac{m_t^i + b_t^i}{P_t} = w_t h_t^i + r_t k_t^i + (1 - \delta)k_t^i + a_t^i$$

と表現できることになる。ここで、期間  $t+1$  に持ち越す金融資産は

$$a_{t+1}^i = \{m_t^i + (1 + i_t)b_t^i + g_{t+1} M_t\} / P_{t+1}$$

である。状態変数を  $a_t^i, k_t^i$  とできるので、価値関数を  $V(a_t^i, k_t^i)$  と表記するとき、動的計画法の Bellman 方程式は

$$V(a_t^i, k_t^i) = \max_{c_t^i, m_t^i, h_t^i} [u(c_t^i, \frac{m_t^i}{P_t}, 1 - h_t^i) + E_t V(a_{t+1}^i, k_{t+1}^i)]$$

となる。以下の計算では、家計のインデックス  $i$  を省略する。最適解の第1階の条件は

$$u_c(c_t, \frac{m_t}{P_t}, 1 - h_t) = \beta E_t [V_k(a_{t+1}, k_{t+1})], \quad (23)$$

$$-u_h(c_t, \frac{m_t}{P_t}, 1 - h_t) = \beta E_t [V_k(a_{t+1}, k_{t+1})w_t], \quad (24)$$

$$u_m(c_t, \frac{m_t}{P_t}, 1 - h_t) = -\beta E_t [V_a(a_{t+1}, k_{t+1})/(1 + \pi_{t+1})] + \beta E_t [V_k(a_{t+1}, k_{t+1})], \quad (25)$$

$$\beta E_t [V_a(a_{t+1}, k_{t+1}) \frac{1 + i_t}{1 + \pi_{t+1}}] = \beta E_t [V_k(a_{t+1}, k_{t+1})] \quad (26)$$

である。ここで、 $V_c$  は関数  $V$  の  $C_t$  に関する偏微分、 $V_h$  は  $(1 - h_t)$  に関する偏微分、 $V_m$  は  $m_t/P_t$  に関する偏微分を表わす。 $\pi_t$  は  $\pi_t = P_t/P_{t-1} - 1$  と定義されるインフレ率である。envelope 定理から

$$V_a(a_t, k_t) = \beta E_t [V_k(a_{t+1}, k_{t+1})], \quad (27)$$

$$V_k(a_t, k_t) = \beta E_t [V_a(a_{t+1}, k_{t+1})(r_t + 1 - \delta)] \quad (28)$$

が成立している。

以上の条件を整理すると、経済の動的変動を表現する以下のような動的方程式が得られる。

$$u_m(c_t, \frac{m_t}{P_t}, 1 - h_t) = \frac{i_t}{1 + i_t} u_c(c_t, \frac{m_t}{P_t}, 1 - h_t), \quad (29)$$

$$-u_h(c_t, \frac{m_t}{P_t}, 1 - h_t) = u_c(c_t, \frac{m_t}{P_t}, 1 - h_t)w_t, \quad (30)$$

$$u_c(c_t, \frac{m_t}{P_t}, 1 - h_t) = \beta E_t [(r_{t+1} + 1 - \delta)u_c(c_{t+1}, \frac{m_{t+1}}{P_{t+1}}, 1 - h_{t+1})], \quad (31)$$

$$\frac{1 + i_t}{1 + \pi_{t+1}} = r_{t+1} + 1 - \delta \quad (32)$$

これらの方程式の経済学的な含意は金融論では常識となっているので、ここでは説明を要しない。各家計が同一であると仮定するとき、

$$K_t = k_t^i, H_t = h_t^i, C_t = c_t^i, M_t = m_t^i, B_t = b_t^i$$

となっている。以下ではこれを仮定する。 $\hat{m}_t = M_t/P_t$  および  $\hat{b}_t = B_t/P_t$  と表記する。実物市場での均衡条件式は、

$$K_{t+1} = f(\lambda_t, K_t, H_t) - C_t + (1 - \delta)K_t$$

であり、資本の蓄積方程式となっている。

経済への外的な攪乱は生産技術へのショックと貨幣供給へのショックの2種類となっている。

$$\lambda_t = \bar{\lambda} e^{z_t^p},$$

$$g_t = \bar{g} + z_t^g$$

と表現できて、攪乱過程がそれぞれ以下のような AR(1) 過程

$$z_{t+1}^p = \rho_p z_t^p + \epsilon_{t+1}^p, \tag{33}$$

$$z_{t+1}^g = \rho_g z_t^g + \epsilon_{t+1}^g \tag{34}$$

と表現できるとする。ここで、 $\epsilon_t^p$  と  $\epsilon_t^g$  は互いに i.i.d. 確率過程とする。貨幣供給量は

$$\frac{M_t}{P_t} = (1 + g_t) \frac{M_{t-1}}{P_t} = \frac{1 + g_t}{1 + \pi_t} \hat{m}_{t-1}$$

となっており、 $\hat{m}_t$  は実質貨幣残高  $M_t/P_t$  である。

モデルの定常状態での特徴を分析する。実質値が一定となる状態を定常状態と定義する。したがって、定常状態では、

$$c_t = c_{t+1} = c^{ss}, k_t = k_{t+1} = k^{ss}, h_t = h_{t+1} = h^{ss}, \hat{m}_t = \hat{m}_{t+1} = \hat{m}^{ss}$$

となっている。上添え字  $ss$  で定常状態における値を示す。消費のオイラー方程式から、

$$r^{ss} + 1 - \delta = \beta^{-1}$$

が成立する。よって、

$$f_k(\bar{\lambda}, k^{ss}, h^{ss}) = \beta^{-1} - 1 + \delta$$

である。この式は、生産技術が規模に関して収穫一定である限り、資本／労働比率  $k^{ss}/h^{ss} = \hat{k}^{ss}$  が割引率と減価償却率にのみ依存し、貨幣量の大小には依存しないことを意味する。資源制約式から、

$$c^{ss} = f(\bar{\lambda}, k^{ss}, h^{ss})/h^{ss} - \delta k^{ss}/h^{ss} = \bar{\phi} h^{ss}$$

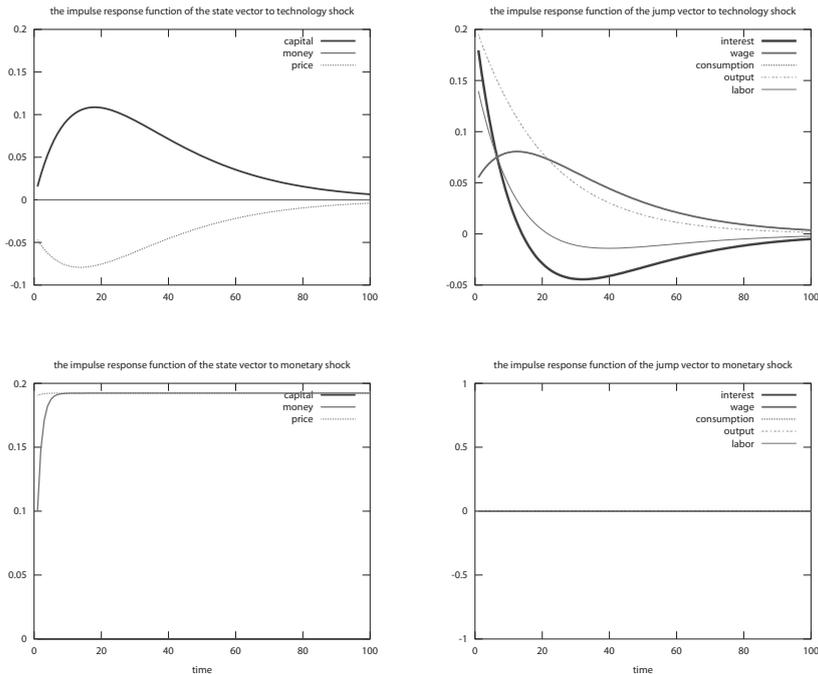
が成立する。ただし、 $\phi(k/h) = f(\bar{\lambda}, k, h)/h$ ,  $\bar{\phi} = \phi(k^{ss}/h^{ss}) - \delta k^{ss}/h^{ss}$  と定義した。労働に関する第1階条件(30)式より、

$$\frac{-u_h(\bar{\phi}h^{ss}, \hat{m}^{ss}, 1 - h^{ss})}{u_c(\bar{\phi}h^{ss}, \hat{m}^{ss}, 1 - h^{ss})} = f_h(\bar{\lambda}, k^{ss}, h^{ss}) = \phi(\hat{k}^{ss}) - \hat{k}^{ss}\phi'(\hat{k}^{ss})$$

である。効用関数が貨幣に関して分離可能であるとき、 $u_c$  および  $u_h$  は貨幣量に依存しない。したがって、労働供給  $h^{ss}$  は貨幣量に依存しないで、決定される。よって、消費量も貨幣量に依存しない。貨幣のスーパー中立性が成立している。しかし、効用関数が貨幣に関して分離可能でない場合、貨幣のスーパー中立性は成立しない。貨幣供給量の拡大率は連動してインフレ率の増大を引き起こし、債券の名目利率を増加させ、貨幣保有の機会費用を上昇させる。貨幣保有の第1階条件式から

$$\frac{u_m(\bar{\phi}h^{ss}, \hat{m}^{ss}, 1 - h^{ss})}{u_c(\bar{\phi}h^{ss}, \hat{m}^{ss}, 1 - h^{ss})} = \frac{i^{ss}}{1 + i^{ss}} = \frac{1 + \pi^{ss} - \beta}{1 + \pi^{ss}}$$

が成立している。労働に関する第1階条件と貨幣保有に関する第1階条件式を連立すると、労働供給  $h^{ss}$  と実質貨幣残高  $\hat{m}^{ss}$  の値が決定される。 $\pi^{ss} = \bar{g}$  なので、労働供給  $h^{ss}$  と実質貨幣残高  $\hat{m}^{ss}$  の値は  $\bar{g}$  の大きさに依存する。貨幣成長率の増加が労働供給をどのように変化させるかは効用関数の性質に依存する。



MIU モデルのカリブレーション：生産技術および貨幣供給へのショック

モデルを解析的に解いて、動学的な挙動を分析することは極めて困難であるので、パラメータ値を変更させつつ、モデルのカリブレーションを実行させて、その結果を考察するという手続きが必要となる。

McCandless(2008) に従って、家計の効用関数を

$$u(c_t^i, h_t^i) = \ln c_t^i + D \ln \frac{m_t^i}{P_t} + B h_t^i$$

と仮定し、生産関数を

$$f(\lambda_t, K_t, H_t) = \lambda_t K_t^\theta H_t^{(1-\theta)}$$

と仮定する。各パラメータに以下のような値を設定する。

$$\beta = 0.99, \delta = 0.025, \theta = 0.36, \bar{\lambda} = 1.0, B = -2.5805, D = 0.01, \bar{g} = 0, \rho_p = 0.95, \rho_g = 0.48$$

この条件の下で、MIU モデルを対数線形化して、この線形化したモデルのシミュレーションを行う。生産技術および貨幣供給へのショックに対するインパルス応答関数は前頁のようなグラフとなる。

このグラフ上で利子率は実質値表示である。明らかに、貨幣ショックは名目値のみを変化させ、実質値に何の影響も与えない。

Walsh(2010) も同様なカリブレーションを行っている。彼が用いた効用関数は、

$$u(c_t, m_t/P_t, h_t) = \frac{[a c_t^{1-b} + (1-a)(m_t/P_t)^{1-b}]^{(1-\Psi)/(1-b)}}{1-\Psi} + \psi \frac{(1-h_t)^{1-\eta}}{1-\eta}$$

という消費と貨幣保有が分離不可となっている CES 関数である。各パラメータ値は

$$\beta = 0.989, \delta = 0.019, \theta = 0.36, a = 0.99, b = 3.0, \Psi = 2.0, \psi = 1.34, \eta = 1., \bar{g} = 0.01, \rho_p = 0.95, \rho_g = 0.75$$

と設定されている<sup>(18)</sup>。

## 6.2 CIA モデル

Hansen and Cooly(1989) によって提案された cash-in-advance モデルを展開する。マクロ経済を構成する各家計は区間  $[0, 1]$  の間に一様に分布しており、家計の総計は 1 である。各家計を添え字  $i$  でインデックス付ける。 $i \in [0, 1]$  である。各家計  $i$  は期待効用関数の現在割引価値

(18) 詳細は Walsh(2010) の第 2 章を参照のこと。Walsh は MATLAB/Dynare を用いてシミュレーションを実行している。このシミュレーションの MATLAB コードは Walsh のホームページ <http://people.ucsc.edu/walshc> からダウンロード可能である。

$$E_0\left[\sum_{t=0}^{\infty} \beta^t u(c_t^i, 1 - h_t^i)\right]$$

を最大するように、消費量  $c_t^i$  と労働供給量  $h_t^i$  を選択する。有効時間を 1 に規格化しているの  
で、レジャー時間は  $1 - h_t^i$  である。マクロ経済の生産関数を

$$y_t = f(\lambda_t, K_t, H_t)$$

と仮定する。 $K_t$  は資本量の経済全体での総計、 $H_t$  は総家計の労働供給量なので、

$$K_t = \int_0^1 k_t^i di, \quad H_t = \int_0^1 h_t^i di$$

と定義される。 $\lambda_t$  は生産技術の水準を表す。要素市場が完全競争市場であれば、資本の限界生  
産は資本のレンタル価格（利率） $r$  に等しく、労働の限界生産は賃金率  $w$  に等しくなる。

$$r_t = \frac{\partial f(\lambda_t, K_t, H_t)}{\partial K_t}, \quad w_t = \frac{\partial f(\lambda_t, K_t, H_t)}{\partial H_t}.$$

次に、各家計の予算制約を考える。家計  $i$  は前期から手持ちの貨幣量  $m_{t-1}^i$  を持ち越し、当該期  
の始めに政府から貨幣の移転を受け取るとする。この政府からの貨幣移転を  $g_t M_{t-1}$  とする。こ  
こで、 $M_{t-1}$  は前期末（期間  $t-1$  末）の一人当たり総貨幣量とする。 $g_t$  は貨幣供給の成長率であ  
る。 $g_t > 0$  であるとき、貨幣供給量は増加する。期間  $t$  における家計  $i$  の cash-in-advance 制約は

$$P_t c_t^i \leq m_{t-1}^i + g_t M_{t-1} \tag{35}$$

となる。ここで、 $P_t$  は期間  $t$  における価格である。この仮定は、期間  $t$  が始まると、まず、財市  
場での取引が開始され、その後に資産市場が開かれて、債券の取引が行われると想定している<sup>(19)</sup>。  
また、通常は、この制約式は等式で成立すると想定できる。上記の貨幣供給に関する仮定では、  
各家計の貨幣保有量とは無関係に政府からの貨幣移転量が決まる。政府からの移転が各家計の  
貨幣保有量に依存するときは、仮定を変更しなければならない。貨幣供給の成長率は割引率よ  
りも大きいと仮定する、 $g_t \geq \beta$ 。もしこの仮定が成り立たないと、消費が時間と共に  $\beta$  よりも  
大きな率で拡大し続けるという問題が生じる。

家計  $i$  は期間  $t$  の期首に金融資産  $P_t a_t^i = m_{t-1}^i + (1 + i_{t-1}) b_{t-1}^i$  を保有するだけでなく、前期に蓄積  
した資本量  $k_t^i$  を持ち越す。期間  $t$  における家計のフローの所得は賃金所得  $w_t h_t^i$  と資本のレンタ  
ル料  $r_t k_t^i$  である。よって、家計の予算制約式は

(19) 期間  $t$  が始まると最初に資産市場で取引が開始され、その後に財の取引が行われるというケースに対  
応するモデルは Walsh(2010) で展開されている。

$$c_t^i + k_{t+1}^i + \frac{m_t^i + b_t^i}{P_t} = w_t h_t^i + r_t k_t^i + (1 - \delta)k_t^i + \frac{m_{t-1}^i + g_t M_{t-1} + (1 + i_{t-1})b_{t-1}^i}{P_t} \quad (36)$$

となる。キャッシュ制約(35)と予算制約式(36)の下に生涯にわたる効用関数最大化問題をラグランジュ乗数を用いて解くと、以下のような第1階の条件式が成立する。

$$-u_h(c_t, 1 - h_t) = \beta u_c(c_t, 1 - h_t) \frac{w_t}{1 + \pi_{t+1}}, \quad (37)$$

$$u_c(c_t, 1 - h_t) = E_t[(r_{t+1} + 1 - \delta) \frac{1 + \pi_t}{1 + \pi_{t+1}} u_c(c_{t+1}, 1 - h_{t+1})], \quad (38)$$

$$\frac{1 + i_{t-1}}{1 + \pi_t} = r_t + 1 - \delta \quad (39)$$

各家計が同一であると仮定するとき、

$$K_t = k_t^i, H_t = h_t^i, C_t = c_t^i, \hat{M}_t = \hat{m}_t^i (M_t = m_t^i)$$

となっている。 $\hat{m}_t = M_t/P_t$  および  $\hat{b}_t = B_t/P_t$  と表記する。実物市場での均衡条件式は、

$$K_{t+1} = f(\lambda_t, K_t, H_t) - C_t + (1 - \delta)K_t$$

であり、資本の蓄積方程式となっている。

技術進歩に関して、 $\lambda_t = \bar{\lambda} e^{z_t^p}$  とおいて、

$$z_{t+1}^p = \gamma z_t^p + \epsilon_{t+1}^p$$

と想定する。 $\epsilon_{t+1}^p$  は i.i.d. 確率過程であるとする。外的攪乱項は生産技術へのショックと貨幣供給へのショックからなる。貨幣供給へのショックを  $g_t = \bar{g} + z_t^g$  とおいて、 $z_t^g$  が

$$z_{t+1}^g = \pi z_t^g + \epsilon_{t+1}^g$$

に従うとする。 $\epsilon_{t+1}^g$  は i.i.d. 確率過程であるとする。利子率および賃金率は

$$r_t = \theta \lambda_t K_t^{\theta-1} H_t^{1-\theta} = \theta \frac{Y_t}{K_t}, \quad w_t = (1 - \theta) \lambda_t K_t^\theta H_t^{-\theta} = (1 - \theta) \frac{Y_t}{H_t}.$$

で与えられる。

定常状態での性質を分析する。消費のオイラー方程式から

$$\beta^{-1} = r^{ss} + 1 - \delta$$

であるから、MIU モデルと同じく、資本/労働比率  $\hat{k}^{ss}$  は割引率と減価償却率にのみ依存し、貨幣成長率には依存しない。よって、賃金率  $w^{ss}$  も貨幣成長率  $\bar{g}$  には依存しない。貨幣の供給ルールから、 $\hat{m}^{ss} = (1 + \bar{g}) / (1 + \pi^{ss}) \hat{m}^{ss}$  が成立するので、インフレ率は貨幣成長率に等しい。また、

CIA 制約から,  $c^{ss} = \hat{m}^{ss}$  が成り立つ。資源制約式から  $c^{ss} = \bar{\phi} h^{ss}$  なので, 消費/労働比率も貨幣成長率に依存しない。労働供給に関する第1階の条件から,

$$\frac{-u_h(\bar{\phi} h^{ss}, 1 - h^{ss})}{u_c(\bar{\phi} h^{ss}, 1 - h^{ss})} = \beta \frac{w^{ss}}{1 + \pi^{ss}}$$

が得られる。この式は労働供給量  $h^{ss}$  がインフレ率, 言い換えると, 貨幣成長率に依存して変化することを意味する。労働と消費が分離可能で, レジャーの限界効用が逓減するならば, 貨幣成長率の上昇は労働供給を減少させる。貨幣はスーパー中立的ではない。

McCandless(2008) に従って, 家計の効用関数を

$$u(c_t^i, h_t^i) = \ln c_t^i + [A \frac{\ln(1 - h_0)}{h_0}] h_t^i$$

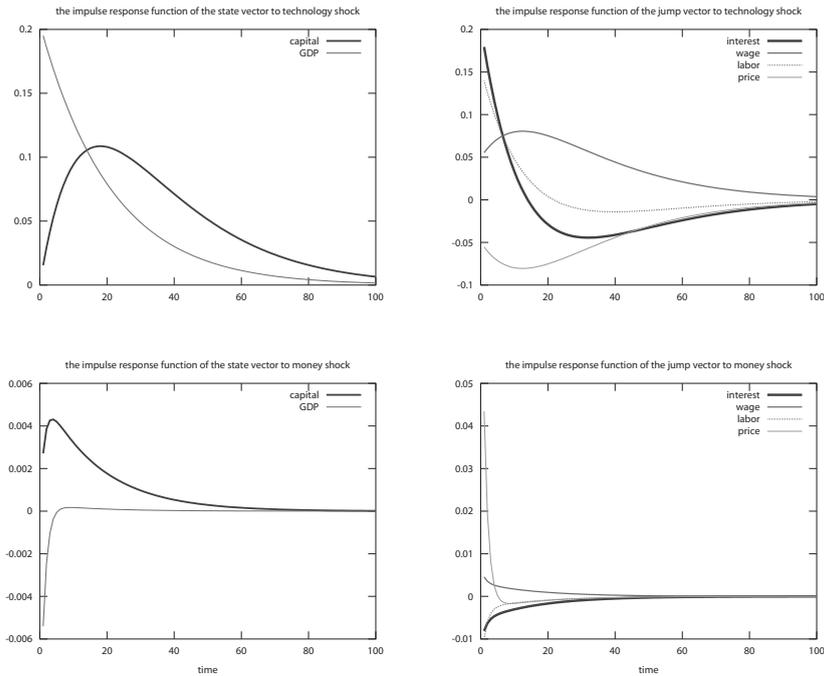
と仮定する。生産関数を

$$f(\lambda_t, K_t, H_t) = \lambda_t K_t^\theta H_t^{(1-\theta)}$$

と仮定する。各パラメータに以下のような値を設定する。

$$\beta = 0.99, \delta = 0.025, \theta = 0.36, \bar{\lambda} = 1.0, A = 1.72, \bar{g} = 0, \rho_p = 0.95, \rho_g = 0.48$$

この条件の下で, CIA モデルを対数線形化して, この線形化したモデルのシミュレーションを



CIA モデルのカリブレーション: 生産技術および貨幣供給へのショック

行う。生産技術および貨幣供給へのショックに対するインパルス応答関数は前頁のようなグラフとなる。

貨幣供給の増大は価格を上昇させるので、名目消費や名目 GDP は増加するが、実質で見ると一時的に減少が起きている。

Walsh(2010) も CIA モデルのカリブレーションを行っている。彼が用いた効用関数は、

$$u(c_t, m_t/P_t, h_t) = \frac{[ac_t^{1-b} + (1-a)(m_t/P_t)^{1-b}]^{(1-\Psi)/(1-b)}}{1-\Psi} + \psi \frac{(1-h_t)^{1-\eta}}{1-\eta}$$

であり、各パラメータ値は

$$\beta = 0.989, \delta = 0.019, \theta = 0.36, a = 0.99, b = 3.0, \Psi = 2.0, \psi = 1.34, \eta = 1.0, \bar{g} = 0.01, \\ \rho_p = 0.95, \rho_g = 0.75$$

と設定されている。CIA モデルの詳細は Walsh(2010) の第 3 章に記述されている。線形化したモデルを演繹する過程の詳細は、Walsh のホームページで提供されている PDF ファイルを参照のこと。Walsh は、MATLAB 上で実行可能なコード、MATLAB から Dynare 使用するプログラム・コードを提供している。MALTAB 向けのプログラムは、Uhlig が開発したモジュールを用いるプログラムと、Paul.Soderlind が開発したモジュールを使用して実行するプログラムの 2 種類になっている。このうち、Uhlig が開発したモジュールを用いるプログラムは Octave 上で実行可能であるが、他のプログラムは Octave では実行できない。その理由は、Paul.Soderlind が開発したモジュールで使用される ordqz という組み込み関数が Octave 内には用意されていないからである<sup>(20)</sup>。

## 7. 終わりに

以上で、新ケインズ派が金融政策や経済政策の効果を分析する目的で活用してきた DSGE モデルを GNU Octave を用いてシミュレーションする具体的な手続きを説明してきた。線形動的方程式のフィードバック行列を計算するためのプログラムを分析者自身で独自に書くことは大変な労力が必要とされるので、この労力を省力化するためには、Uhlig が提供しているモジュールを活用することが便利である。そこで最後に、本稿での DSGE モデルのシミュレーションでも実際に用いた、Uhlig によって提供されたモジュールの使用方法を説明する。一般的に経済モデルが (5, 6, 7) 式と同じく

$$AAx_t + BBx_{t-1} + CCy_t + DDz_t = 0, \quad (40)$$

$$E_t[FFx_{t+1} + GGx_t + HHx_{t-1} + JJy_{t+1} + KKy_t + LLz_{t+1} + MMz_t] = 0, \quad (41)$$

(20) 本稿中で用いたプログラム、Walsh が提供したプログラムおよび Uhlig の Toolkit4.1 パッケージは、著者のホームページ <http://www.meijigakuin.ac.jp/~mashiyam/studies.htm> からダウンロード可能です。

$$z_{t+1} = NNz_t + \epsilon_{t+1}, \quad (42)$$

で与えられるとする。動学的経済モデルの解を (8, 9) と同じく、以下のような動的な均衡則

$$x_t = PPx_{t-1} + QQz_t, \quad (43)$$

$$y_t = RRx_{t-1} + SSy_{t-1}, \quad (44)$$

として求めることを考える。リカッチ方程式を満たすフィードバック行列  $PP$  を見出し、行列  $QQ, RR, SS$  を計算するプログラム・モジュールは以下の 6 ケのスクリプトファイルからなるサブルーチン群

```
solve.m
solve_qz.m
qzdiv.m
qzswitch.m
calc_qrs.m
sol_out.m
```

から構成される。必要とされるオプションを与えて、`solve.m` を実行すると、行列  $P$  および行列  $Q, R, S$  が計算される<sup>(21)</sup>。その後、`sol_out.m` を実行すると、これらの値が表示される。

Uhlig のモジュールを利用するためには、当然、モデルのスクリプト内で、動的方程式の係数行列を数値で与えなければいけない。具体的にシミュレーションしたいモデルの係数行列を与えた後、モデルの次数を

```
% set the dimensions for the state, jump, and exogenous variables.
```

```
[l_eqn, m_states] = size(AA);
```

```
[l_eqn, n_endog] = size(CC);
```

```
[l_eqn, k_exog] = size(DD);
```

と与える。 $x, y, z$  に対して使用する変数名は

```
VARNAMES = ['capital ',
'money',
'price',
'interest',
'wage',
'technology shock',
'monetary shock'];
```

---

(21) これらのサブルーチン群は Octave のカレント・ディレクトリーあるいはパスを通じたディレクトリーにインストールしておいてください。

のように  $x, y, z$  の順番に従って定義する。

`solve.m` を実行する前に設定しなければいけないオプションは以下のとおりである。

```
TOL=0.000001;
DO_QZ=1;
MANUAL_ROOTS=0;
IGNORE_VV_SING=1;
DISPLAY_IMMEDIATELY=0;
DISPLAY_ROOTS = 1;
message = ' ';
warnings = [ ];
% Starting the calculations for matrices PP, QQ, RR, SS.
solve;
sol_out;
```

と書けば、行列  $PP$  および行列  $QQ, RR, SS$  が計算され、結果が表示される。ここで、 $TOL=0.000001$  はゼロと判断する数値を  $0.000001$  以下と指定することを示す。 $DO\_QZ=1$  は  $QZ$  分解を用いて、リカッチ方程式を解くことを指定する<sup>(22)</sup>。このオプションを変更する必要性がないので、上記のとおり書いておけば十分である。

## 付録：Octave の組み込み関数と条件文

以下に、基本的な組み込み関数および条件文を一覧表で示す。

### Arithmetic and Increment Operators

$x + y$	addition
$x - y$	subtraction
$x * y$	matrix multiplication
$x . * y$	element by element multiplication
$x / y$	right division, conceptually equivalent to $(\text{inverse}(y') * x)'$
$x ./ y$	element by element right division
$x \setminus y$	left division, conceptually equivalent to $\text{inverse}(x) * y$
$x . \setminus y$	element by element left division
$x ^ y$	power operator
$x . ^ y$	element by element power operator
$- x$	negation

(22) 一般化固有値問題を解いて行列  $PP$  を計算したいときは、 $DO\_QZ = 0$  と設定する。計算結果を観察してから、手動で 1 以下の固有値を選択したい場合、 $MANUAL\_ROOTS = 1$  と設定する。

<code>+ x</code>	unary plus(a no-op)
<code>x'</code>	complex conjugate transpose
<code>x.'</code>	transpose
<code>++ x(- x)</code>	increment(decrement), return new value

### Basic Matrix Manipulations

<code>rows(a)</code>	return number of rows of a
<code>columns(a)</code>	return number of columns of a
<code>all(a)</code>	check if all elements of a nonzero
<code>any(a)</code>	check if any elements of a nonzero
<code>find(a)</code>	return indices of nonzero elements
<code>sort(a)</code>	order elements in each column of a
<code>sum(a)</code>	sum elements in columns of a
<code>prod(a)</code>	product of elements in columns of a
<code>min(args)</code>	find minimum values
<code>max(args)</code>	find maximum values
<code>rem(x, y)</code>	find remainder of x/y
<code>reshape(a, m, n)</code>	reformat a to be m by n
<code>diag(v, k)</code>	create diagonal matrices
<code>linspace(b, l, n)</code>	create vector of linearly-spaced elements
<code>logspace(b, l, n)</code>	create vector of log-spaced elements
<code>eye(n, m)</code>	create n by m identity matrix
<code>ones(n, m)</code>	create n by m matrix of ones
<code>zeros(n, m)</code>	create n by m matrix of zeros
<code>rand(n, m)</code>	create n by m matrix of random values
<code>randn(n,m)</code>	create n by m matrix of normal distribution random values

### Linear Algebra

<code>chol(a)</code>	Cholesky factorization
<code>det(a)</code>	compute the determinant of a matrix
<code>eig(a)</code>	eigenvalues and eigenvectors
<code>expm(a)</code>	compute the exponential of a matrix
<code>hess(a)</code>	compute Hessenberg decomposition
<code>inverse(a), inv(a)</code>	invert a square matrix
<code>norm(a, p)</code>	compute the p-norm of a matrix
<code>pinv(a)</code>	compute pseudoinverse of a
<code>qr(a)</code>	compute the QR factorization of a matrix

<code>rank(a)</code>	matrix rank
<code>sprank(a)</code>	structural matrix rank
<code>schur(a)</code>	Schur decomposition of a matrix
<code>svd(a)</code>	singular value decomposition
<code>syl(a, b, c)</code>	solve the Sylvester equation
<b>Plotting Functions</b>	
<code>plot(args)</code>	2D plot with linear axes
<code>plot3(args)</code>	3D plot with linear axes
<code>line(args)</code>	2D or 3D line
<code>patch(args)</code>	2D patch
<code>semilogx(args)</code>	2D plot with logarithmic x-axis
<code>semilogy(args)</code>	2D plot with logarithmic y-axis
<code>loglog(args)</code>	2D plot with logarithmic axes
<code>bar(args)</code>	plot bar charts
<code>stairs(x, y)</code>	plot stairsteps
<code>stem(x, it y)</code>	plot a stem graph
<code>hist(y, x)</code>	plot histograms
<code>contour(x, y, z)</code>	contour plot
<code>title(string)</code>	set plot title
<code>axis(limits)</code>	set axis ranges
<code>xlabel(string)</code>	set x-axis label
<code>ylabel(string)</code>	set y-axis label
<code>zlabel(string)</code>	set z-axis label
<code>text(x, y, str)</code>	add text to a plot
<code>legend(string)</code>	set label in plot key
<code>grid[on—off]</code>	set grid state
<code>hold[on—off]</code>	set hold state
<code>ishold</code>	return 1 if hold is on, 0 otherwise
<code>mesh(x, y, z)</code>	3D surface
<code>meshgrid(x, y)</code>	create mesh coordinate matrices
<b>Statistics</b>	
<code>corrcoef(x, y)</code>	correlation coefficient
<code>cov(x, y)</code>	covariance
<code>mean(a)</code>	mean value
<code>median(a)</code>	median value

std(a)                    standard deviation

var(a)                    variance

### Statements

for 文 : i の値が 1 から nn になるまで, for-body の計算を実行する

for i = 1:nn

  for-body

end

if 文 : condition が真実であるならば, if-body を実行し, そうでなければ else-body を実行する

if condition

  if-body

  els

  els-body

end

### Reference

1. Ben S. Bernanke, Mark Gertler and Simon Gilchrist(1999), The Financial Accelerator in a Quantitative Framework, *in Handbook of Macroeconomics*, vol.2, edited by J. B. Taylor and M. Woodford, Elsevier Science.
2. Charles T. Carlstrom and Timothy S. Fuerst(1997), Agency Cost, Net Worth, and Business Fluctuations: A Computable General Equilibrium Analysis, *American Economic Review*, 87(5).
3. Lawrence J. Christiano, Martin Eichenbaum and Charles Evans(2005), Nominal Rigidities and the Dynamic Effects of a Shock to Monetary Policy, *Journal of Political Economy*, 113(1).
4. Michael Dotsey and Robert G. King(2005), Implications of State-Dependent Pricing for Dynamic Macroeconomic Models, *Journal of Monetary Economics*, 52.
5. Rochelle M. Edge, Michael T. Kiley, Jean-Philippe Laforte(2008), Natural rate measures in an estimated DSGE model of the U.S. economy, *Journal of Economic Dynamics and Control*, 32.
6. Jordi Galí, Mark Gertler and D. Lopez-Salido(2001), European Inflation Dynamics, *European Economic Review*, 45(7).
7. Jordi Galí(1999), Technology, Employment, and the Business Cycle: Do technology Shocks Explain Aggregate Fluctuations?, *American Economic Review*, 89(1).
8. Jordi Galí(2008), *Monetary Policy, Inflation, and the Business Cycle: An Introduction to the New Keynesian Framework*, Princeton University Press.
9. Mark Gertler, Simon Gilchrist and Fabio M. Natalucci(2004), External Constraints on Monetary Policy and the Financial Accelerator, *Journal of Money, Credit and Banking*, 39(2-3).
10. Mark Gertler and John Leahy(2008), A Phillips Curve with an Ss Foundation, *Journal of Political Economy*, 116(3).
11. Gary Hansen(1985), Indivisible Labor and the Business Cycle, *Journal of Monetary Economics*, 16.
12. Finn Kydland and Edward C. Prescott(1982), Time to Build and Aggregate Fluctuations, *Econo-*

- metrica*, 50.
13. Lars Ljungqvist and Thomas J. Sargent(2004), *Recursive Macroeconomic Theory*, Second edition, The MIT Press.
  14. George McCandless(2008), *The ABCs of RBCs: An introduction to Dynamic Macroeconomic Models*, Harvard University Press.
  15. Thomas J. Sargent(1987), *Dynamic Macroeconomic Theory*, Harvard University Press.
  16. Frank Smets and Rafael Wouters(2007), Shocks and Frictions in US Business Cycles: A Bayesian DSGE Approach, *American Economic Review*, 97(3).
  17. John Stachurski(2009), *Economic Dynamics: Theory and Computation*, The MIT Press.
  18. Nancy Stokey and Robert Lucas(1989), *Recursive Methods in Economic Dynamics*, Harvard University Press.
  19. Herald Uhlig(1999), A Toolkit for Analysing Nonlinear Stochastic Models Easily, in *Computational Methods for Study of Dynamic Economies*, edited by Ramon Marion and Andrew Scott, Oxford University Press.
  20. Volker Wieland, et. al.(2009), A New Comparative Approach to Macroeconomic Modeling and Policy Analysis, Working Paper, Goethe University of Frankfurt.
  21. Carl E. Walsh(2010), *Monetary Theory and Policy*, Third edition, The MIT Press.
  22. Michael Woodford(2003), *Interest and Prices: Foundations of a Theory of Monetary Policy*, Princeton University Press.
  23. 加藤涼 (2007), 『現代マクロ経済学講義』, 東洋経済新報社.
  24. 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～」, 内閣府経済社会総合研究所, ESRI Discussion Paper Series No.203.