新型コロナウイルス危機のマクロ経済分析：補論

この補論では、本文で扱った非行動 SIR モデルと行動 SIR モデルについて、モデルの詳細とシミュレーション手法について解説を行う。動学的一般均衡モデルについてある程度の知識を持つ、もしくは下記の参考文献とともに読み進めていくことを前提とする。

ソフトウェアについて

Matlab または Octave を用いた。前者は有償、後者は無償だが Matlab と高い互換性を持つ。どちらのソフトウェアでも、この補論の同じプログラムがそのまま動くことを確認している。また、行動 SIR モデルについては、Matlab と Octave 両方の上で動く Dynare というパッケージを利用している。2021 年 3 月現在、Dynare 最新版 4.6.3 が、Octave 最新版 6.2.0 をサポートしていない。筆者は Mac 版の古い Octave 4.4.1 をダウンロードして使った。動学的一般均衡モデルの解説と、ソフトウェアの使い方などについては、以下を参照。

- 加藤涼 (2006)『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
  - ソフトウェアの使い方などサンプルページ
  - `http://www.ryokato.org/genmac/`
- 蓮見亮 (2020)『動学マクロ経済学へのいざない』日本評論社
  - ソフトウェアの使い方などサンプルページ
  - `https://www.rhasumi.net/wiki/wiki.cgi?page=dynamicmodels`
- 仲田泰祐 (2020,2021)「ゼロ金利制約下の金融政策」経済セミナー（日本評論社）連載中
  - ソフトウェアの使い方などサンプルページ
  - `https://sites.google.com/view/keisemi-zlbfrb/`
- Octave の Mac 用旧版ダウンロード
  - `https://octave-app.org/Download.html`
- 行動 SIR モデルで参考にした Krueger et al. (2020) のコード
  - `https://github.com/tjxie/KUX_PandemicMacro`

非行動 SIR モデルについて

本文の式 (1) から (4) と、新規感染者の式 (5), (7), (8) のうちどれかを使う。第 1 期に外生的に人口の 0.1% が感染すると仮定して、それ以降の感染拡大は繰り返し将来に向かって for ループで 100 期間計算していく。

行動 SIR モデルの理論

行動 SIR モデルは理論とシミュレーションにわけてそれぞれ詳細に解説する。本文では Eichenbaum, Rebelo, and Trabandt (2020a) を簡単化したモデルを用いている。

まず、SIRD の式は本文の式 (1) から (4) と同様である。新規感染者数については本文 (9) 式に従い、感染が S と I に依存する非行動 SIR と同様の経路と、消費に依存する経路の 2 つが入っている。このモデルでは

ギリシャ文字 β を割引因子として用いるので、代わりに以下のように π と π_c を使って書き直す。

$$T_t = \pi S_t I_t + \pi_c \left(S_t C_t^s\right)\left(I_t C_t^i\right) \tag{A.1}$$

ここで、C_t^s は感受性者の一人当たり平均消費、C_t^i は感染性者の一人当たり平均消費を表している。下添字 t は週を表す。

感受性者は、今週の消費や労働という経済活動を増加させることと、将来の感染確率上昇についてトレードオフを持っている。U_t^s を感受性者の t 期における期待効用の割引現在価値とする。これは、今週、来週、再来週と続いて得られていく効用の期待値を足し合わせたものとなっており、価値関数 (value function) と呼ばれている。U_t^i を同様に感染性者の期待効用の割引現在価値とする。感受性者が確率 τ_t で感染する場合、以下のような式で表される。

$$U_t^s = u(c_t^s, n_t^s) + \beta\left[(1-\tau_t)U_{t+1}^s + \tau_t U_{t+1}^i\right] \tag{A.2}$$

まず、感受性者は消費 c_t^s と労働 n_t^s から、効用 $u(c_t^s, n_t^s)$ を今週得る。次に、将来の価値は $\beta < 1$ の係数で割り引かれて評価される。来週以降将来にわたる割引現在価値は、確率 $(1-\tau_t)$ で感染することはなく U_{t+1}^s、また確率 τ_t で感染してしまって U_{t+1}^i の価値となる。効用関数については以下のように特定化しよう。

$$u(c, n) = \ln c - \left(\frac{\theta}{2}\right) n^2$$

ここで ln は自然対数 (natural log) であり、またこの効用関数は感染状態 S, I, R によらず共通である。(A.2) 式において c_t^s と n_t^s が小文字で表されているのは、個人の消費量の選択を表すためである。自分以外の大多数の人々が (C_t^s, N_t^s) を選択しているのを所与として、経済全体ではほぼ影響をもたらさない一個人として (c_t^s, n_t^s) を決定しているということを強調して示している。ただ均衡では各感受性者は対象的なので、$c_t^s = C_t^s$ と $n_t^s = N_t^s$ となる。この個人の消費量が c_t^s だった場合の感染確率は、この対称性の上で式 (A.1) より、

$$\tau_t = \frac{T_t}{S_t} = \pi I_t + \pi_c c_t^s \left(I_t C_t^i\right) \tag{A.3}$$

と書ける。この個人は消費 c_t^s を行うために労働所得を得る必要がある。このモデルでは企業は線形の生産関数を持っており、N の労働を投入すると AN だけの商品を生産することができる。このとき、限界生産性と賃金が等しくなることから、賃金は常に A となる。このため感受性者は n_t^s 時間だけ労働を行う An_t^s の労働所得を得るので、予算制約式は

$$c_t^s = An_t^s \tag{A.4}$$

となる。

もう一度感受性者のトレードオフを繰り返すと、感受性者は労働所得を得て消費を行う、つまり (c_t^s, n_t^s) を増やすインセンティブを持つが、しかし感染確率 τ_t が上昇してしまって来週以降の期待効用割引現在価値が U_t^s から U_t^i に減ってしまうことを恐れる、というトレードオフを持つ。この問題は以下のような最適化問題になる。

$$\begin{aligned}
\max_{c_t^s, n_t^s}\ & U_t^s = \ln c_t^s - \left(\frac{\theta}{2}\right)(n_t^s)^2 + \beta\left[(1-\tau_t)U_{t+1}^s + \tau_t U_{t+1}^i\right] \\
\text{s.t.}\ & c_t^s = An_t^s, \\
& \tau_t = \pi I_t + \pi_c c_t^s \left(I_t C_t^i\right)
\end{aligned}$$

この最適化問題は、t 期にも定義されるし、また $t+1$ 期にも同様に定義される。このような再起的に定義される動学的最適化問題の式をベルマン方程式と呼ぶ。最適解は一階条件として与えられる。制約式を代入して計算すると、

$$\frac{1}{c_t^s} = \left(\frac{\theta}{A}\right) n_t^s + \beta(U_{t+1}^s - U_{t+1}^i)\pi_c \left(I_t C_t^i\right) \tag{A.5}$$

となる。

次に、感染者の問題を考えよう。感染者はもう将来の感染を恐れることはない。治癒者の期待効用割引現在価値を U^r_{t+1} とする。感染者が確率 γ_r で治癒した場合には U^r_{t+1} を得るが、確率 γ_d で死亡してしまった場合は割引現在価値がゼロになってしまうと仮定する。本文では統計的生命価値 (VSL) について議論したが、ここでは「個人」の価値観として死の価値を U^r_{t+1} と置いていることになる。ただし、規範的意味合いを含んだ社会的価値とは異なる可能性がある。

$$\max_{c^i_t, n^i_t} \ U^i_t = \ln c^i_t - \left(\frac{\theta}{2}\right)(n^i_t)^2 + \beta\left[(1-\pi_r-\pi_d)U^i_{t+1} + \pi_r U^r_{t+1} + \pi_d \times 0\right], \tag{A.6}$$
$$\text{s.t.} \ c^i_t = An^i_t$$

現在の行動は感染状態の移行には関わらないため、最適化の条件は現在の変数のみで決まる。このため、最適な消費と労働は

$$n^i_t = \frac{1}{\sqrt{\theta}} \tag{A.7}$$
$$c^i_t = \frac{A}{\sqrt{\theta}} \tag{A.8}$$

と非常に簡単に決まる。

最後に治癒者の問題は以下のようになる。

$$\max_{c^r_t, n^r_t} \ U^r_t = \ln c^r_t - \left(\frac{\theta}{2}\right)(n^r_t)^2 + \beta U^r_{t+1}, \tag{A.9}$$
$$\text{s.t.} \ c^r_t = An^r_t$$

治癒者は将来のステータスがもう変わることはない。一回条件を計算すると、式 (A.7) と式 (A.8) と完全に一致して、$n^r_t = 1/\sqrt{\theta}$, $c^r_t = A/\sqrt{\theta}$ となることが分かる。

ここまででシミュレーションに必要な全ての式が求まった。市場均衡条件も必要と思われるが、競争的な労働市場で賃金が A に決まるため、仮に計算しても常に成立しているため、考慮する必要がない。このモデルは差分方程式体系となっており、変数は 11 個

$$S_t, I_t, R_t, D_t, T_t, \tau_t, C^s_t, N^s_t, U^s_t, U^i_t, U^r_t$$

となっている。これに応じて以下のように式も 11 本ありので解くことができる。

- SIRD について、本文の式 (1) から (4) が 4 本
- 予算制約式 (A.4)
- 感染確率の式 (A.3) に式 (A.8) を代入したもの
$$\tau_t = \frac{T_t}{S_t} = \pi I_t + \pi_c c^s_t I_t \left(\frac{A}{\sqrt{\theta}}\right)$$
- 新規感染者の式 (A.1) は $T_t = S_t\tau_t$ と書ける
- 感受性者の一回条件 (A.5) に式 (A.8) を代入したもの
$$\frac{1}{c^s_t} = \left(\frac{\theta}{A}\right) n^s_t + \beta(U^s_{t+1} - U^i_{t+1})\pi_c I_t \left(\frac{A}{\sqrt{\theta}}\right)$$
- 3 本のベルマン方程式。感染性者と治癒者は式 (A.7) と式 (A.8) を代入
$$U^s_t = \ln c^s_t - (\theta/2)(n^s_t)^2 + \beta\left[(1-\tau_t)U^s_{t+1} + \tau_t U^i_{t+1}\right]$$
$$U^i_t = \ln\left(\frac{A}{\sqrt{\theta}}\right) - \frac{1}{2} + \beta\left[(1-\pi_r-\pi_d)U^i_{t+1} + \pi_r U^r_{t+1}\right]$$
$$U^r_t = \ln\left(\frac{A}{\sqrt{\theta}}\right) - \frac{1}{2} + \beta U^r_{t+1}$$

行動 SIR モデルのシミュレーション

このモデルは「完全予見」という仮定の元で解かれる。このモデルでは、初期に 0.1% の人が突然感染した後は確率的な要素は何も入っていない。人々は合理的に現在から将来にわたる感染状態の動き (S_t, I_t, R_t, D_t) や経済状態 (C_t^s, N_t^s) を正しく予想していること前提とする。その上で、人々は感染確率 τ_t や割引現在価値 (U_t^s, U_t^i, U_t^r) も正しく把握している。これらの市場や市中感染の状況を前提として、最適化行動を行って消費や労働を決定する。すると、最適化して決めた (c_t^s, n_t^s) が確かに (C_t^s, N_t^s) と等しくなり、またその上で決まる感染推移も予測と等しくなる、という合理的期待均衡を考えている。

この均衡は Dynare 上でシミュレーションされるわけだが、これは動学的一般均衡モデルをすでに学習したことがある方にとっても少しとっつきにくい。Dynare でよく行われるのはモデルの定常均衡周りで近似してインパルス応答関数を出すものだが、このモデルではこれは感染者の動きが非線形であるためこの手法が使えない。ただ確率的な要素がない完全予見下での deterministic simultion と呼ばれる非線形動学については、Dynare の方で専用のパッケージが用意されている。簡単に言えば、初期点と終点を定常均衡として計算した上で、このモデルではその間を 11 本の非線形差分方程式が成立するように解かれる。シミュレーションでは、11 変数、11 式を 100 期間で考えているが、要は $1100 = 11 \times 100$ 個の変数について 1100 本の非線型方程式体系を解くだけである。しかし、行動 SIR モデルの解法は大きな問題がある。通常のマクロモデルでは定常状態が一意に求まるが、このモデルでは終点の S や R、D はそれまでに感染経路に依存するために内生的に決まる。このため厳密には終点が求められないのであるが、サンプルプログラムでは、S が 1、R と D が 0 という感染前の定常状態にモデルことを仮定して解いている。このため、最終期とその 1 期前でこれらの変数が突然ジャンプして元に戻ってしまう経路をたどり、数値計算上の誤差が生じうる。しかし、最終期の 1 期前はほぼ感染が収束しておりこの効果は無視できること、またこのモデルでは経済主体が貯蓄や資本蓄積など、感染に関わる以外の動学的な意思決定は行わないことから、本モデルではこの誤差はほぼ無視できると考えられる。Eichenbaum et al. (2020b) では変数の差分を取ってシミュレーションすることでこの問題を回避している。

このシミュレーションは他にもう一つ問題がある。行動 SIR モデルの方程式体系は通常のマクロモデルに比べて非常に非線形性が高く、そのままパラメーターを代入してもうまく解を求めることができない。このため、通常の Dynare で完全予見動学を解く場合に比べて少しテクニックがいる。ここで homotopy という手法が使われている。まずは消費経路の感染が完全にない $(\pi_c = 0)$ の場合のモデルを解く。次に、その結果を初期値として、消費経路の感染がほんの少しの場合を解く。さらにその結果を次の初期値として、消費経路の感染がもう少し大きい場合を解き、というプロセスを繰り返して最終的には本当のパラメーターの解を出すというものである。Dynare macro language と呼ばれるものを使い、$\pi_c = 0$ をゼロから始めてループで徐々に増加させつつ何度もモデルを解くという手法を用いる。

```
% 久保田荘（2021）「新型コロナウイルス危機のマクロ経済分析」『医療経済学研究』
% 非行動 SIR モデルのシミュレーション

% Mac 版の Octave 4.4.1 と Matlab R2020a で動作確認
% 適当なファイル（NB_SIR.m とか）を作って、全体をコピペして使える。

% 初期化：変数消去、プロットウィンドウ閉じる、コマンドライン消去
clear;close all;clc;

%
% ここからパラメーター設定
%

% シミュレーションは 100 期間
periods=100;

% 感染パラメーターの設定
gam_d = (7/18)*0.005; % 感染は平均で 18 日続き、0.5% が死亡する。週次にするため 7 で割る。
gam_r = (7/18)*(1-0.005); % 感染は平均で 18 日続き、99.5% が回復する。
beta = 0.5852;
% 感染確率 beta について：これは、Eichenbaum et al. (2020a) で
% 消費や労働がなかった場合に対応するように設定している。
% この論文では、N=28; A=39.835; C=A*N; pi_s1=7.8408e-08; pi_s2=1.2442e-04; pi_s3=0.3901;
% 経済のないこのプログラムでは beta が全ての感染経路とになる。これは以下で計算。
% beta = 0.5852 = pi_s1*C^2+pi_s2*N^2+pi_s3;

% ロックダウンや感染者隔離は 100 期間分のベクトルに記述。
% 本文では別々に記述しているが、L に書けばロックダウンとして全員の行動が、
% Q に書けば感染者のみの行動を止めることができる。両方入れることもできる。
L=zeros(periods,1); % 100 期間、ロックダウンのベクトル
Q=zeros(periods,1); % 100 期間、感染者隔離のベクトル

% ロックダウンの場合、20 期から 40 期まで、20% の人の行動を止める。
L(20:40) = 0.2;

% 感染者隔離の場合、ロックダウンをコメントアウトして、こちらははずす。
% 20 期から 40 期まで、40% の感染者を止める。
%Q(20:40) = 0.4;



%
% シミュレーション用の結果を入れるベクトルの準備
%
```

```
% S,I,R,D のそれぞれの結果を格納する 100 期間のベクトル
S=ones(periods,1);
I=zeros(periods,1);
R=zeros(periods,1);
D=zeros(periods,1);
% 実効再生産数
Rt=zeros(periods,1);
Rt(1) = beta*S(1)/(gam_r+gam_d);  % 初期値だけ入れておく
% 経済損失
Y=zeros(periods,1);
Y(1) = L(1)*(S(1)+R(1)) + (L(1)*(1-Q(1))+Q(1))*I(1) + D(1);

%
% シミュレーション本体
%

I(1) = 0.001; % 初めに人口の 0.1% の人が突然感染。これがないとモデルが動き始めない

% シミュレーションは 1 期から繰り返して計算する
for t=1:periods-1
    % 新規感染者の式は全体ロックダウン L と一部感染者隔離 Q の両方入れてある。
    T = beta * (1-L(t))*S(t) * (1-L(t))*(1-Q(t))*I(t);
    % 定義に従って計算する
    S(t+1) = S(t) - T;
    I(t+1) = I(t) + T - gam_r*I(t) - gam_d*I(t);
    R(t+1) = R(t) + gam_r*I(t);
    D(t+1) = D(t) + gam_d*I(t);
    % 経済損失は第 1 項がロックダウンによるもの、第 2 項感染者隔離のもの、
    % 第 3 項が死者数分労働人口が減る効果。第 2 項はロックダウンとの重複を調整。
    Y(t+1) = L(t+1)*(S(t+1)+R(t+1)) + (L(t+1)*(1-Q(t+1))+Q(t+1))*I(t+1) + D(t+1);
    Rt(t+1) = beta*S(t+1)/(gam_r+gam_d);
end

%
% 結果のプロット
%

% 6 つ図を書く。結果をベクトルにまとめる。
results = [I,S,R,D,Rt,Y];
% それぞれのタイトルを決める。ベクトルにまとめてある。
titles = {'I 感染性人口', 'S 感受性人口', 'R 治癒人口', 'D 累積死亡者数', 'Rt 実効再生産数', 'Y 経済損失'};
```

```
% 新規の描画用ウィンドウを作成
fig = figure;
% 6 回ループしてプロット
for x = 1:6
    subplot(2,3,x); % 縦 2 列、横 3 列
    plot(results(:,x), 'linewidth', 2); % x 番目の図を書く
    title(titles{x}); % x 番目の図にタイトルをつける
end
```

```
// 久保田荘（2021）「新型コロナウイルス危機のマクロ経済分析」『医療経済学研究』
// 行動 SIR モデルのシミュレーション、Dynare プログラム
// Eichnbaum et al. (2020a) の簡単版
// 簡単化：労働経由の感染を削る、感染による生産性低下を削る。
// Krueger-Uhlig-Xie のコードを元にしている。
// https://github.com/tjxie/KUX_PandemicMacro

// 適当な Dynare ファイル（例えば B_SIR.mod）を作って全体をコピペして使える。

// Matlab の場合
// Mac 版の Dynare 4.6.3、Matlab R2020a で動作確認。

// Octave の場合、2021 年 3 月時点の最新版 6.2.0 を Dynare がサポートしていない。
// 以下から Mac 版 Octave 4.4.1 をダウンロードして使える。
// https://octave-app.org/Download.html
// さらに statistics package を入れる必要がある。
// pkg install -forge io statistics
// インストール中 locale についてのエラーがいっぱい出るが待つ。

// パスの追加、win のときと、mac のとき。以下をコメントアウト
// addpath c:/dynare/4.6.3/matlab
// addpath /Applications/Dynare/4.6.3/matlab
// 初期設定にしておいた方が楽。Matlab なら Home-->Set Path
// Octave なら、home directory の.octaverc に書いておく。



// 0 初期化 //////////////////////////////////////////
close all;clc;
// clear は、繰り返し計算を消してしまうため、入れてはいけない。



// 1 マクロループ用パラメーター設定 /////////////////////

// このプログラムは homotopy を利用している。
// モデルが非線形のため、消費経由感染のパラメータ pi_c について、
// 最初から真の値を入れるとうまく解をみつけられない。
// 感染のパラメーター pi_c を最初はゼロから始めて、
// この値を徐々に上げつつ何度も繰り返してシミュレーション。
// 毎回、前回の値を初期値に使うので、少しずつ計算していく。
// 最終的に真の値 pi_target まで引き上げる。
// このループのため、dynare macro を使っている。
```

```
// @#define で真の値を定義。
// @#for で繰り返し計算。

// 真の値を定義。2 通り考える。
// 感染のパラメーターは、非行動 SIR では beta だったが行動 SIR では pi に変更している。

// 経済が 100% の場合、
@#define pi_c_target = 4.7038e-07 // 消費経由の感染
@#define pi_definition  = 0 // S と I の人口だけで決まる感染

// 経済が 1/3 の場合、こちらのコメントを外す。
//@#define pi_c_target = 1.5682e-07 // 消費経由の感染
//@#define pi_definition = 0.3901 // S と I の人口だけで決まる感染

// パラメータ設定について
// もともとの Eichenbaum et al. (2020) では消費 + 労働=1/3 だった。
// N=28; A=39.835; C=A*N; pi_s1=7.8408e-08; pi_s2=1.2442e-04; pi_s3=0.3901;
// 労働が除かれていてるので、全部消費に。1.5682e-07 = (pi_s1*C^2+pi_s2*N^2)/C^2;
// 経済 100% の場合は以下で計算。
// 4.7038e-07=(pi_s1*C^2+pi_s2*N^2+pi_s3)/C^2



// 2 変数 ///////////////////////////////////////////

// 内生変数
var S // 感受性人口
    I // 感染性人口
    R // 治癒人口
    D // 累積死亡者数
    T // 新規感染者数
    tau // S の感染確率
    cs // S の消費
    ns // S の労働
    Us // S の value function
    Ui // I の value function
    Ur; // R の value function、ここで内生変数最後なので";"を書く

// 外生変数。最初に I = 0.001 で 0.1% の人口が突然感染
varexo eps; // 初期感染は外生ショック
```

```
// 3 パラメーター ////////////////////////////////////////////

// パラメーターの名称定義
parameters  pi_c // 消費経由感染の係数
            pi // S と I の直接感染の係数
            gam_r // I が治癒する確率
            gam_d // I が死ぬ確率
            theta // 労働不効用のウェイト
            A // 生産性
            betta; // 割引因子、最後なので";"をつける。

// パラメーターに値を入れる。
gam_d = (7/18)*0.005; // 感染は平均で 18 日続き、0.5% が死亡する。週次にするため 7/18。
gam_r = (7/18)*(1-0.005); // 感染は平均で 18 日続き、99.5% が回復する。
betta = 0.96^(1/52); // 割引因子は１年で 0.96、52 週ある。
A = 39.835;
theta = 0.001275;
pi = @{pi_definition}; // Dynare マクロで定義された直接感染のパラメータを入れる。
pi_c  = 0; // 消費経由の感染はゼロから始めて、少しずつ上げていく。



// 4 モデル ///////////////////////////////////////////////

model;

// SIR の式。
S = S(-1) - T(-1);
I = T(-1) + (1 - gam_r - gam_d) * I(-1) + eps; // eps が 0.1% 初期感染
R = R(-1) + gam_r*I(-1);
D = D(-1) + gam_d*I(-1);

// 感受性者の一階条件
(1/cs) = (theta/A)*ns + betta * (Us(+1) - Ui(+1)) * pi_c * (A/sqrt(theta)) * I;
// 予算制約
cs = A*ns;
// 感受性者の感染確率
tau = pi * I + pi_c * cs * (A/sqrt(theta)) * I;
// 新規感染者数
T = S * tau;

// value functions
Us = log(cs) - (theta/2)*ns^2 + betta*((1 - tau)*Us(+1) + tau*Ui(+1));
```

```
Ui = log(A/sqrt(theta)) - theta/2*(1/sqrt(theta))^2 + betta*((1 - gam_d - gam_r) * Ui(+1) + gam_r * 
Ur = log(A/sqrt(theta)) - theta/2*(1/sqrt(theta))^2 + betta*Ur(+1);

end;




// 5 定常状態 ////////////////////////////////////////////////

// ここで、初期の定常状態のみを定義。
// 最後の定常状態は、シミュレーションでは初期値に戻る。
// このとき、例えば R は正なのに、最終期にゼロにジャンプしてしまう。
// シミュレーション上は少し誤差が出るはずだが、
// このモデルは貯蓄が入っておらず dynamic な行動変化が少ないのでほぼ大丈夫。

steady_state_model; // 定常状態定義

ns  = 1/sqrt(theta);  // R の一階条件より。感染がなければ全員これ。
cs  = A*ns;
tau = 0;
I   = 0;
T   = 0;
S   = 1;
R   = 0;
D   = 0;
// 定常状態では毎期 cs と ns を選択する。betta で割引くので、総和は 1/(1-betta)
Us  = (log(cs)-theta/2*ns^2)/(1-betta);
Ur  = Us; // 感染発生前の定常状態では R 存在しないが、もしいるなら同じ
// I も定常状態では存在しない。仮に測度ゼロで存在したら死ぬ可能性も考える
Ui  = (log(cs)-theta/2*ns^2+betta*gam_r*Ur)/(1-betta*(1-gam_d-gam_r));

end;

steady; // 定常状態作ってチェック
check;




// 6 シミュレーション ////////////////////////////////////////////////

// 初期に 0.1% 感染を入れると、どんどん広まる。
shocks;
    var     eps;
```

```
    periods 1:1;
    values  0.001;
end;

// 完全予見シミュレーション。100 期
perfect_foresight_setup(periods = 100);

// pi_c をゼロから始めて真の値 pi_c_target まで繰り返す。
// value が徐々に上がっていく。value をモデルの中の pi_c に代入して解く。
@#for value in 0:2e-8:pi_c_target
    pi_c = @{value}; // value を pi_c に代入。
    perfect_foresight_solver;  // 前回の結果を初期値に使う。
@#endfor



%% 7 その他の変数を計算 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% ここから dynare プログラムではなく、Matlab/Octave のプログラム
% コメントも % で書いている。

% 総消費、人口比かけて足しあげる。I と R は A/sqrt(theta) で定数。
C = cs.*S+(I+R)*(A/sqrt(theta));

% 次は総消費の減少率。感染前は A/sqrt(theta) なのでそれで割っている。
% 生産関数が線形なので、GDP 減少率も同じ。
C_loss = -(C-A/sqrt(theta))/(A/sqrt(theta));

% 実効再生産数。シミュレーション結果は、1 期に最初に定常状態、
% 2-101 期に 100 期間シミュレーション、102 期に最後の定常状態
% Rt は I=0 の定常状態では定義できないので、真ん中だけで定義。
Rt = zeros(length(cs),1); % length(cs) は 102 のこと。
% 最初と最後を除くと、2:length(Rt)-1 になる。
Rt(2:length(Rt)-1) = (T(2:length(Rt)-1)./I(2:length(Rt)-1))/(gam_r+gam_d);

% 全ての結果をまとめる。
results = [I S R D Rt C_loss];



%% 8 プロット %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% それぞれのタイトルを決める。ベクトルにまとめてある。
```

```
titles = {'I 感染性人口', 'S 感受性人口', 'R 治癒人口', 'D 累積死亡者数', 'Rt 実効再生産数', '経済損失'};
% 新規の描画用ウィンドウを作成
fig = figure;
% 6 回ループしてプロット
for x = 1:6
    subplot(2,3,x); % 縦 2 列、横 3 列
    % 全部の結果をプロットすると、最初と最後の定常状態が入って変なことになる。
    % 最初と最後以外で、2:length(cs)-1 を取り出す。
    plot(results(2:length(cs)-1,x), 'linewidth', 2); % x 番目の図を書く
    title(titles{x}); % x 番目の図にタイトルをつける
end
```