

金融経済論 I 講義ノート RBCコードの解説

2010年11月16日

塩路

rbc2.mの解説

パラメーター値の設定

- clear all; **これまで作った変数などは全て忘れなさい**
- %% ----- [1] Parameter proc -----
-
- global sigma alpha myu beta delta lamda phi ;
この行は重要である。ここではどの変数がGlobal変数かを宣言している。Global変数とは、Matlabがこのファイルの外に出て「関数ファイル」(後述)内で計算するときにも、このファイルで設定したのと同じ値を使いなさい、ということの意味している。その重要性はおいおい明らかとなるであろう。たとえば、上の行でsigmaという変数をGlobal変数に指定しているが、次の行で...
- sigma = 1.5 ; % CRRA
としている。よって、sigmaの値はこのファイルの外でも1.5。

より詳しくは、Matlab上で”help global”と打ってみてください！

パラメーター値の設定、続き

- $\sigma = 1.5$; % CRRA 消費に関する効用関数の曲がり具合、この値が大きいほど、異時点間の消費の変動を嫌う = consumption smoothing motiveが強い
- $\alpha = 0.3$; % Cobb-Doug 生産関数における資本ストックのシェア
- $\mu = 1$ % labor-consumption supply 効用関数における労働供給の(消費と比べた)重要性
- $\beta = 0.99$; % discount factor 割引因子
- $\delta = 0.025$; % depreciation 資本減耗率
- $\lambda = 2$; % labor supply elasticity >1 労働供給に関する効用関数の曲がり具合、この値が大きいほど、異時点間の労働供給の変動を嫌う = 労働供給の異時点間の代替効果が小さくなる
- $\phi = 0.8$; % AR(1) in tech 技術ショックの持続性

(非確率的)定常状態の計算

- %% ----- [2] Steady State proc >> -----

- % SS capital & ss labor

- % (1) real rate (By SS euler)

- $kls = (((1/\beta)+\delta-1)/\alpha)^{1/(\alpha-1)}$; 資本・労働比率

最適成長モデルとその系譜に連なるモデルにおいては、定常状態における実質利子率は割引因子ベータで決まる。つまり、人々が将来の効用をどのくらいの率で割り引くかで決まる。ただし、定常状態における技術進歩率がゼロと異なる場合には、これも実質利子率に影響する。

均衡においては実質利子率は資本ストックの収益率と一致する。

この収益率は資本の実質レンタル料と関係している。

実質レンタル料はMPKで決まる。

生産関数が1次同次であるとき、MPKはK/Lで決まる。

以上から、まずK/Lが決定される。

- % (2) wage

- $wstar = (1-\alpha)*(kls)^\alpha$;

1次同次生産関数の性質より、MPL、ひいては実質賃金もK/Lの関数である。

定常状態続き

- % (3) Labor and goods market clear
- $clstar = kls^{\alpha} - \delta * kls$; 消費・労働比率、この式は (消費) = (生産) - (投資) から求められる。
- $lstar = ((wstar/myu) * (clstar^{(-\sigma)}))^{1/(\lambda + \sigma)}$; 労働供給と消費の間の限界代替率 = 実質賃金
- $kstar = kls * lstar$; 資本ストック
- $cstar = clstar * lstar$; 消費
- $vstar = 1$; 技術水準
- $Ystar = (kstar^{\alpha}) * (lstar^{(1-\alpha)})$; 生産
-
- $ssCKoLY = [cstar \ kstar ; lstar \ Ystar]$ % show SS values 結果をちょっとプリントアウトしてみる

内生変数のリストを作っておく

- %% -----[2] MODEL proc-----%%
- % 内生変数のリスト ('a' denotes t+1 values)
- % t+1期のジャンプ変数、t+1期の先決変数、t期のジャンプ変数、t期のジャンプ変数の順
- % このモデルではジャンプ変数はlとcで先決変数はkとv
- % バージョン1ではこの順序が間違っていた。
- % la ca ka va lt ct kt vt

ここのところは何も計算していないが、自分でプログラミングするときには変数の順番に関するルールを忘れてはいけない、という戒めである。

lは労働、cは消費、kは資本ストック、vは技術の項
aは来期(t+1期)の変数、tは今期(t期)の変数を表す。

いよいよモデルの線形近似へ

- %% ----- [3] Linearization proc -----%%
- % DIFFERENTIATION OF THE MODEL
- % Specify the point at which the derivatives are taken
- xss=[lstar cstar kstar vstar lstar cstar kstar vstar] ;
モデル式を微分するポイントをあらかじめ指定する。
ここで、前ページで決めた変数の順番を守って指定する。
- % Differentiation
- coef1 = fdjac_japan('rbc2_model',xss);
“rbc2_model”という名前の関数を、xssというポイントで微分して、計算結果をcoef1という行列に入れなさい。

ここで、“fdjac_japan”は関数を(数値的に)微分するためのコマンドである。
ではrbc2_modelというファイルの中をみよう→[クリック!](#)。

GDPの決定式も微分しておく

- % DIFFERENTIATION OF THE GDP EQUATION
- % Specify the point at which the derivatives are taken
- `yss=[lstar kstar vstar]`; 関数へのインプット
- % Differentiation
- `coy1 = fdjac_japan('rbc2_others',yss);`
- “rbc2_others”のファイルに書かれている関数を、“yss”のポイントで微分しなさい。→[クリック!](#)

線形近似から対数線形近似へ

- % TRANSFORMATION INTO A LOG LINEARIZED MODEL
- % In terms of % deviations from ss
- $vo = [lstar \ cstar \ kstar \ vstar]$; 各変数の定常値のベクトル(行ベクトル)
- $TW = [vo ; vo ; vo ; vo]$; 行ベクトルをたてにスタック
- 次にこのTWを(微)係数行列にかけることで、普通の線形近似から対数線形近似に変えている。

連立差分方程式モデルの完成

- さて、ここまででcoef1という(微)係数行列が求まった。モデルの変数は8個、式は4本、coef1は4かける8行列である。
- この時点でモデルは $\text{coef1} * z = 0$ という形をしている。詳しく書くと、次ページのようにになっている。

つづき

$$\left[-\tilde{B} \quad | \quad \tilde{C} \right] \cdot \begin{bmatrix} x_{t+1} \\ \text{---} \\ x_t \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

ただしBティルダとCティルダはともに4かける4行列。
これを次のページの式のように直す。

つづき

$$\tilde{B} \cdot x_{t+1} = \tilde{C} \cdot x_t$$

$$x_t = \begin{bmatrix} l_t - l^* \\ c_t - c^* \\ k_t - k^* \\ v_t - v^* \end{bmatrix}$$

次ページのようにxの定常状態の値を各係数にかけると対数線形近似のための微係数行列となる。

つづき

- $B = [-\text{coef1}(:, 1:4)] .* \text{TW} ;$
- $C = [\text{coef1}(:, 5:8)] .* \text{TW} ;$

$$B \cdot \hat{x}_{t+1} = C \cdot \hat{x}_t,$$

$$\hat{x}_t = \begin{bmatrix} \log(l_t) - \log(l^*) \\ \log(c_t) - \log(c^*) \\ \log(k_t) - \log(k^*) \\ \log(v_t) - \log(v^*) \end{bmatrix}$$

つづき

- $A = \text{inv}(C) * B$ %(Linearized reduced form)

$$C^{-1} \cdot B \cdot x_{t+1} = x_t$$

$$\Rightarrow A \cdot x_{t+1} = x_t$$

rbc2_modelの解説

関数ファイルについて

- このファイルは、いわゆる「関数ファイル」である。
- 関数ファイルとは、Matlab上で自分で好きなように関数を定義することを可能にするものである。
- 同じような作業を繰り返し行いたいようなときに重宝する。
- ここでは、我々のRBCモデル全体を一つの関数として定義している。

関数ファイルの開始

- `function optcon=f1(x);`
- まず、“function”によってこのファイルが関数ファイルであることを宣言している。
- インプットを“x”、アウトプットを“optcon”と呼ぶことにする。“f1”はいわばこのファイル内における関数の名前であるが、これは実はなんでもよい。

つづき

- インプットである”x”はこのファイルの外から与えられる。
- 我々の例でいうと、”rbc2.m”のファイル内で、

```
coef1 = fdjac_japan('rbc2_model',xss);
```

と書いた。これは、このrbc2_model.mの関数ファイルにおいて、インプット”x”のところにベクトル”xss”を代入して計算(具体的には微分)をこなさい、という意味である。

計算の開始

- $la=x(1)$; $ca=x(2)$; $ka=x(3)$; $va=x(4)$;
 $lt=x(5)$; $ct=x(6)$; $kt=x(7)$; $vt=x(8)$;
- ベクトルxは8つの要素からなっている。それぞれに名前をつける。これらの名前は、特に指定しない限り、このファイル内でのみ有効である。

グローバル変数の指定

- global sigma alpha myu beta delta lamda phi ;
- これらのパラメーターについては、もとファイル”rbc2.m”で指定したのと同じ値を使いなさい、ということ。
- もとファイルと関数ファイルの両方でこのグローバル変数の指定をしてやる必要。

あとはひたすらモデルを書きこむ

- % Eliminate Price
- $ra = (va * \alpha * (ka/la)^{(\alpha-1)}) ;$
- $wt = (1-\alpha) * vt * (kt/lt)^{\alpha};$
- まず資本の実質レンタル料(t+1期)と実質賃金を計算しておく。

微分の対象となる関数を書き入れる

- % Optimal Conditions & state transition
 - labor = $l^{\lambda} w / (\mu c^{\sigma})$; % LS = LD
 - euler = $c^{(-\sigma)} - (c^{(-\sigma)})^{\beta} (1 + r - \delta)$; % C-Euler
 - capital = $k a - (1 - \delta) k - v t (k^{\alpha}) (l^{(1-\alpha)}) + c$; % K-trans
 - tech = $v a - \phi v t$;
-
- 労働市場の均衡条件、消費のオイラー方程式、資本ストックの蓄積を表す式、技術項の推移を表す式
 - すべての式を、「 $\dots = 0$ 」の形に直して、その左辺のみを書いていることに注意。
 - それぞれの式に、labor, euler, capital, techという名前を付けている。

最後に、関数の指定

- `optcon = [labor ; euler ; capital ; tech];`
- このファイルのアウトプットである`optcon`は、これら4本の式からなっている、ということをごここで指定して終わる。

rbc2_othersの解説

- `function gdpcon=f1(x);`
- `lt=x(1) ; kt=x(2) ; vt=x(3) ;`
- `global alpha ;` もとファイルで指定したアルファの値を用いる
- `Yt = vt*(kt^alpha)*(lt^(1-alpha));` 生産関数
- `gdpcon = [Yt];` アウトプットの指定、ここではGDPだけ