確率的動学一般均衡理論と数値計算

山田知明
立正大学経済学部

2008年8月

はじめに

- 動学的一般均衡理論は、分野を問わずに幅広く応用
- 例えば、
    - 景気循環・経済成長 (RBC)
    - 金融・財政政策 (New Keynesian Model)
    - 企業の参入・退出や雇用創出・喪失
    - 異質な経済主体
    - 社会保障問題
    - 情報の非対称性、モラルハザードやコミットメント制約
- 何故、数値計算が必要になるのか?
    1. 解析的な解 (Analytical Solution) が存在しない
    2. モデルの数量的含意 (Quantitative Implications) が知りたい
- 統計パッケージのようにコマンド一つでは出来ない
    - この講義の目的 $\Rightarrow$ 数値計算を「使える」ようにする

はじめに

- **統計パッケージのようにコマンド一つでは出来ない**
  - **この講義の目的** $\Rightarrow$ **数値計算を「使える」ようにする**

[R]eading a book on computation techniques without actually using the compter is as foolish as reading a cook-book without ever entering the kitchen.(Marimon and Scott (1999), p,2)

- **講義ノート& プログラム**
  - http://homepage2.nifty.com/˜tyamada/index.html
  - Matlab & Fortran (Password:hit-u_lecture)

数値計算は理論か?実証か?

- **両方の知識が必要**
  - **理論モデルが必要になる**
  - **モデルのパラメターには実証的基礎が必要**
- Bona and Santos (1997,JET)

"It is worth emphasizing that the numerial model is not generally meant to directly emulate an economic situation. This model is simply a supporting device (an algorithm of some sort) aimed at the simulation of the true behavior of the mathematical model. One should have a good idea of how well the behavior and stability properties of the numerical model mimic those of the mathematical model under consideration."

実際に数値計算を始めるにあたって

- **テキスト選択**
  - Judd (1998)：**数値解析に関する一番基本的なテキスト**
  - Miranda and Fackler (2002)：Matlab **との親和性が高い初心者向けテキスト。読みやすいが、**Judd (1998) **と比べると厳密性に欠ける**
  - Adda and Cooper (2003)：**動学的マクロ経済学全般に関するテキストで、理論モデル、推計と数値計算を一つの本の中にコンパクトにまとめている**
  - Marimon and Scott (1999)：**各トピックについて、その分野のスペシャリスト達が書いている。トピックがやや古い箇所もあるかも**

数値計算のための数学

- Quarteroni, Sacco and Salei (2000)、Atkinson (1989)：数値計算に必要な数学的基礎をまとめた本。必ずしも全て読む必要はないが、手元においておくととても便利なリファレンス。Quarteroni et al. (2000) はMatlabコードでの解説もある。
- Press, Flannery, Teukolsky and Vetterling (1992): ニュートン法等、数値計算に必要なアルゴリズムの実際的な解説とコードを書いてくれているので、文字通り、レシピ感覚で使える。C、C++、FORTAN 77、Fortran 90/95用がある。数値計算の専門家から見ると問題もあるらしいので、少し注意する必要があるかもしれない。

ソフトウェア選択

- 実際に手を動かして数値計算を行う前に、ソフトウェアの選択を避けて通る事は出来ない
- ソフトウェアは、大雑把に分けて、次の3種類の選択肢がある

1. C言語やFortranのように数値計算専用ではない言語
2. MatlabやGaussのような数値計算(科学技術演算)用のソフトウェア
3. MapleやMathematicaのような数式処理ソフト

ソフトウェア選択

Fortran[FORTRAN]

- 数値計算に関しては一番優れた環境が整う
  - ただし、Matlab 等に比べると手間がかかる
  - I/O は極めて貧弱なので、グラフなどは別のツールが必要

1. Intel Visual Fortran (+IMSL)：統合環境。数値計算パッケージである IMSL がついている。
2. NAG Fortran Builder (NAG)：学習用もあるのでお手ごろ感がある。
3. FTN 95：商用目的でない限りフリーのコンパイラ
4. GCC/G95：GNU プロジェクトによるコンパイラ
5. IMSL/NAG：有料のパッケージ(サブルーチンの集まり)
6. LAPACK/BLAS/MINPACK/HOMPACK：フリーのサブルーチン達(netlib)

ソフトウェア選択

Matlab/Scilab/Octave

- Fortran や C 言語と比べると敷居が低く、入門者用のソフトウェア達
- 扱いやすいが計算のスピード面では問題がある
- 大きなモデルを解きたい場合、プログラミングの手間がかかっても Fortran や C 言語で書いたほうが時間の節約に
- グラフィック面では、Matlab の方が優れている
- Matlab は正規版を購入すると高額だが、Matlab と似たようなな機能を持つフリーのソフトウェアも開発されている

1. Scilab: http://www.scilab.org/
2. Octave: http://www.gnu.org/software/octave/
3. Fortress: http://projectfortress.sun.com/Projects/Community

ソフトウェア選択

並列処理

- 最近のPCに搭載されているCoreDuo(Core2Duo)は、マルチタスクをこなせる
- 二つ(以上)の仕事をうまく割り振れば、計算効率は高まる
  - 並列処理

1. Open MP：並列処理用のソフトウェア(フリー)。IVFならば始めから対応。
   - 牛島 (2006)、安田・小林・飯塚・阿部・青柳(2006)
2. MPI (Message Passing Iterface)：Open MPよりも汎用性が高いが、パラレル・コンピュータが必要。Open MPはメモリは共有しているが、MPIの場合にはメモリが別々になる。詳しくは下記のHPを参照。
   - Makoto Nakajima's HP: http://www.compmacro.com/makoto/

Matlab から Fortran への移行：その(1)

チューニング技法入門:
http://accc.riken.jp/HPC/training/text.html

- 命令の対応関係を覚えれば、一週間もあれば移行できる
- コンパイル形式とインタープリタ形式では実行速度はかなり違う
- Fortran では、「コンパイル」をして「リンク」をするという作業が必要になる
- が、統合環境だとワンクリックで済むので、あまり意識する必要はない
- 互換性に注意
- コンパイルオプションを使いこなせるようになると、計算速度が急激に上昇する(20分が5分になったりする)
- デバッグ用とリリースは全然違うので、最終版はリリースにする(ただし、最適化をしすぎると計算結果が変わる場合も)

Matlab から Fortran への移行：その (2)

- チューニングはとても大事。例えば、Fortran のループは左の添え字から先に動くので、数学的な行列の演算とは異なる点に注意すると計算スピードがアップする。
- Subroutine、Function、Module/Use の使い方を覚える (Common は読む必要はあるが、自分ではあまり使わない)
- フラグ処理もうまく使いこなせると便利だが、使いすぎると全体構造が理解しづらくなる
- ポインターやオブジェクト指向はあまり使わないと思うが、構造体はわりと便利

Matlab から Fortran への移行：その(3)

- 構造化プログラミングを覚えて、読みやすいコードを書くように心掛ける
- 変数の宣言や単精度と倍精度の違いに注意をはらう
- Matlab では勝手にやっているが、きちんと管理したほうがミスが少なくなる
- グローバル変数とローカル変数の違い、変数のスコープを理解する。変数がクリアされているかを理解していないと、大きなエラーにつながる
- メモリーの動的割り当ては、大型のプログラムを書くときには必ず必要になる
- コメントは必ずつける!
- プログラムはアート

コード置場

- 下記の HP では、実際にいろいろな論文で使用されたコードが置いてある。
- 「講義用に作られたコード」と実際に「研究で使用するコード」にはかなりの隔たりがあるので、くじけない程度に眺めてみると良い。
    - QM&RBC: http://dge.repec.org/codes.html

数値計算誤差：その(1)

- 数値計算には計算誤差が常についてくる。
- まったく知らないと思わぬミスにつながるので、あまり面白い話ではないが、確認しておく必要がある。

1. 丸め誤差 (Round-Off-Errors): 例えば、1/3はコンピュータでは近似的にしか表現できない。
2. 打切り誤差 (Truncation Error): 無限回の演算を実際に行う事は出来ないので、通常は有限回で打ち切るが、その時には当然、誤差が生じている。
3. 測定誤差 (Gross Error):

- Curse of Dimensionaliy

数値計算誤差：その(2)

- 経済理論と数値計算結果の間には少し距離がある
- 経済理論との整合性や数値計算結果の解釈をきちんと考える必要がある

1. 単純にコンピューテーション上の誤差であり、オイラー方程式は実際に満たされている。
2. 合理性の限界と「解釈」。人間の計算可能性からすれば、コンピュータが計算可能な精度は十分に良い。ただし、合理的期待均衡の定義とこのような解釈は整合的?
3. Kubler and Schmedders (2005) は、近似均衡が必ずしも真の均衡の近傍にない事を明らかに。パラメター空間のある経済の ϵ 近傍にある別の経済の均衡、つまり ϵ-均衡として、近似均衡を解釈。

ツールボックスに加えておきたい方法

- **経済モデルを解く時に一般的に「最適な方法」はない**
  - **いくつかの方法を知っておいて、ケースに合わせて使う**
  - Taylor and Uhlig (1990,JBES) **以降の研究**

1. **線形(二次)近似**：Blanchard and Kahn (1980,ECTA)、Juillard (1996)、Sims (2000,2001)、Schmitt-Grohé and Uribe (2005)
2. **摂動法**(Pertubation Method)：Gasper and Judd (1993)、Judd and Guu (1993)
3. **動的計画法**：Benitez-Silva, et al. (2000)、Rust (1996)、Santos (1999)、Santos and Vigo-Aguiar (1998)、Aruoba, et al. (2006,JEDC)
4. **射影法**(Projection Method)：Judd (1992)、Judd, et al. (2003)
5. **構造推計**(Structural Estimation)：Fernández-Villaverde and Rubio-Ramírez (2004,2006,2007)

基本になるモデル(1)

- **一部門最適成長モデル**(One Sector Optimal Growth Model；OSGM)

$$\max \sum_{t=0}^{\infty} \beta^t u(c_t),$$

subject to

$$c_t + k_{t+1} = f(k_t) + (1-\delta)k_t, \;\; k_0 \text{ given}, \; 0 < \beta < 1$$

基本になるモデル(2)

- ベルマン方程式

$$v(k) = \max\{u(c) + \beta v(k')\},$$
$$\text{subject to}$$
$$c + k' = f(k) + (1-\delta)k,$$

  - $v(k)$：価値関数
  - k：状態変数(State Variable)
  - c：操作変数(Control Variable)

基本になるモデル(3)

- **一階条件**

$$u'(f(k)+(1-\delta)k-g(k)) = \beta v'(g(k)),$$

$$v'(k) = (1-\delta+f'(k))u'(f(k)+(1-\delta)k-g(k)),$$

  - $g(k)$ **:政策関数**(Policy Function)
- **何が知りたいのか**?
  - **価値関数より政策関数の場合が多い**
  - **経済主体の意思決定は状態変数にのみ依存し、過去の意思決定の流列には影響されない**(**最適性原理**)

基本になるモデル(4)

- **解析的解**
  - $u(c) = \log c$、$f(k) = k^{\alpha}$、$\delta = 1$

    $$v(k_0) = A + B \log k_0,$$
    $$A = (1-\beta)^{-1}\left[\ln(1-\alpha\beta) + \frac{\alpha\beta}{1-\alpha\beta}\ln\beta\alpha\right], \quad B = \frac{\alpha}{1-\alpha\beta}$$
    $$k_{t+1} = \beta\alpha k_t^{\alpha}.$$
  - See Stokey, Lucas and Prescott (1989)

カリブレーション?

- **モデルを解くためには、関数形とパラメターが必要**
  - **「カリブレーション」**
  - $\alpha \in \{0.25, 0.3\}$
  - $\beta \in \{0.9, 0.99\}$
  - $\gamma \in \{1, 2, 5\}$
  - $\delta \in \{1.0, 0.025\}$
- **もっと知りたい場合、**Cooley (1995)**を参照**

価値関数繰り返し法(Value Function Iteration; VFI)

VFI を選択する理由は、

- **価値関数を繰り返し計算するときに縮小写像の性質**(Contraction Mapping Property)**があるので、**
  - **安全**(Safe)
  - **信頼性**(Reliable)
  - **収束**(Convergence Property)
  - DP **の形にすれば、何にでも応用可能**
  - Santos and Vigo (1998) **を参照**
- **価値関数は本当に収束するのか**?
  1. **数値計算では定義域は必ず有限だから問題なし**(Stokey, et al., 1989)
  2. **同次関数であれば大丈夫**(Alvarez and Stokey, 1998)

Value Function Iteratoin (2)

動的最適化問題を解く際に用いられる方法は、大きく分けて二種類の方法

1. **状態変数や操作変数を離散化して解く方法**(Discretization Approach)
2. **解きたい関数(政策関数)を有限個のパラメータで表現する方法**(Parametric Approximation Approach)

- **モデルを理論的に解く場合、通常、変数に連続性を要請**
- **連続区間をコンピュータで扱うのは不可能**
- **何らかの方法で有限個に離散化する必要がある**

Discretized Approach

状態空間と操作変数が共に離散的な場合の動的計画法の解法

- 非常に遅くコンピュータのメモリも大量に消費
- 安定しており、強い非線形性があるモデルでもOK

1. 状態変数の上限$\overline{k}$及び下限$\underline{k}$を設定
2. 状態変数kおよび操作変数k'をそれぞれ離散n個に区切る $(\{k_i, k'_j\}_{i,j=1}^{n})$
   - 離散個に区切った状態変数の各点の事を、グリッド(Grid)、ノード(Node)あるいはメッシュ(Mesh)などと呼ぶ
3. 資本の状態は離散個に区切った値しかとらない

ベルマン方程式は、

$$v(k_i) = \max\{u(k_i^{\alpha} + (1-\delta)k_i - k_j) + \beta v(k_j)\}.$$

Discretized Approach：アルゴリズム

1. グリッド生成：状態空間及び操作変数を有限個のグリッドに区切る。この有限個のグリッド上における価値関数 v^i の初期値を推測 (Initial Guess) する。
    - 価値関数は収束するので、始めは 0 からスタートしても問題ない
2. 収束の基準：収束の基準になるパラメター ε を設定
3. 効用関数：効用関数 $u(k_i^\alpha + (1-\delta)k_i - k_j)$ を各グリッド上で評価する
    - $n \times n$ の行列になる

Discretized Approach：アルゴリズム(続き)

4. **価値関数を繰り返し計算：**
   1. **各 k_i について**

      $$v(k_i) = \{u(k_i^\alpha + (1-\delta)k_i - k_j) + \beta\hat{v}(k_j'),$$

      を計算。右辺第2項の $v(k_j')$ は、初期値では上で与えたもの(例えばゼロ行列)を使う
   2. **各 k_i について、$v(k_i)$ を最大にする k_j を探す(最適政策)**
   3. **古い価値関数 $\hat{v}$ と新しい価値関数 v が十分に近ければストップ**
   4. **そうでなければ、$v(k_i)$ を $\hat{v}(k_j)$ に代入して、同じ計算を繰り返す**
   5. **繰り返していくと、縮小写像の性質があるため、いずれ収束**

[Matlab Code 1]

Parametric Approximation Approach

政策関数を多項式などによってパラメトリックに表現できれば、ある状態変数 k の下での k' を、任意の $k \in [\underline{k}, \overline{k}]$ で計算できる

- **状態空間の評価点**(Evaluation Point)**を離散個** $\{k_i\}$ **だけ選んできて価値関数及び政策関数** $\{v(k_i), g(k_i)\}$ **を計算**
- **評価点以外の個所については、何らかの方法で近似**
  - **線形近似**
  - **多項式近似**(Chebyshev etc.)
  - **スプライン近似**(Cubic-Spline, Shape-Preserving Spline)
  - etc.

PAA：アルゴリズム

Johnson et al. (1993) や Judd and Solnick (1994) を参照

1. グリッド生成：状態空間を有限個のグリッドに区切る
   - グリッドは等分である必要はなく、流動性制約に直面しやすいゼロ近辺を多めにする方が精度が高くなる
   - 上と同様に、この有限個のグリッド上における価値関数の値 $\hat{v}(k_j')$ の初期値を推測
2. 収束の基準：収束の基準になるパラメター ε を設定
3. 近似・評価：近似点 k_i 上にない価値関数の値については、線形近似や多項式近似、スプライン補間などを使って近似する必要があるため、その係数を計算
   - $\hat{v}(k, \mathbf{b})$ をパラメター $\mathbf{b}$ を使って近似した時の、k 上での価値関数の値とする

PAA：アルゴリズム(続き)

4. 最適化：各 k_i について、次の式を計算

$$v\left(k_i\right)=\max\left\{u\left(k_i^{\alpha}+(1-\delta)k_i-k'\right)+\beta\hat{v}\left(k';\mathbf{b}\right)\right\},$$

   - 最適解を得るためには、ニュートン法などの手法が必要
   - ここをどう工夫するかで計算精度やスピード、安定性が変わってくる
   - このステップで新しい価値関数 $\{v\left(k_i\right)\}$ 及び政策関数 $\{g\left(k_i\right)\}$ を得る

5. 新しいデータ $\{v\}$ を使って、価値関数を求めるためにステップ3に戻る
   - ステップ3と4を繰り返していくことによって価値関数は収束

- 計算速度を速める方法はいろいろある!

[Matlab Code 2]

オイラー方程式を使ったアプローチ

最適化モデルの一階条件であるオイラー方程式を利用

- **代表的なものは、政策関数を多項式などで近似して残差をする射影法**(Projection Method)
- **射影法は**2**種類に分かれる**
  1. **スペクトラル法**(Spectoral Method)**：**Judd (1992)
  2. **有限要素法**(Finite Element Method; FEM)**：**McGrattan (1996,1999)

オイラー方程式を使ったアプローチ(続き)

- 基本的なアイディアは、オイラー方程式の誤差を最小にする意思決定関数を探す事
- オイラー方程式は、

$$u'(h(k)) = (1 - \delta + f'(k'))\beta u'(h(k')),$$

  - 先行研究にならって貯蓄関数 $k' = g(k)$ ではなく消費関数 $c = h(c)$ を政策関数としている

スペクトラル法(1)

- 我々が知りたいのは消費関数である $h(k)$
- 状態空間 $k \in [\underline{k}, \overline{k}]$ 上のスムーズな関数 $h(k)$ を何らかのパラメーター $\{a_i\}$ で近似

$$\hat{h}(k; a) = \sum_{i=0}^{n} a_i \phi_i(k)$$

  - a は近似のパラメーターである多項式の係数
  - $\phi_i(k)$ は基底 (Basis)

スペクトラル法(2)

- 最終的な目標はオイラー方程式の残差関数(Residual Function)をゼロにすること

$$R(k;a) = \beta \frac{u'(f(k) + (1-\delta)k - \hat{h}(k,a))}{u'(\hat{h}(k,a))}(1-\delta+f'(k')) - 1,$$

  - 残差関数は状態空間$[\underline{k}, \overline{k}]$上で定義
  - 連続空間上のあらゆる点で残差がゼロになるのが理論的には正しい
  - 数値計算ではそのような表現は出来ない
  - 「最小にする」事の数値計算上の定義が必要になる

スペクトラル法(3)

- コロケーション法：評価点 $\{k_i\}$ を任意に導出して、

$$R(k_i; a) = 0, \ \ i = 1, \dots n,$$

  - 有限個であるため計算が容易
  - 評価点には関数近似のためのグリッドを使えばよい

- ガラーキン法：残差に重み (Weight)$\Psi(k)$ を付けて

$$\int_{\underline{k}}^{\overline{k}} \Psi(k) R(k; a) dk = 0,$$

  - 重みには、多項式の基底などを用いる
  - 数値計算において積分をそのまま計算する事は出来ない
  - 積分をさらに何らかの方法で近似をする必要がある

スペクトラル法：アルゴリズム(1)

1. 関数を近似する際に必要になる基底を選択する
   - ここでは直交多項式である Chebyshev 多項式を選択
2. 近似の程度 n を選択
   - ここでは $n = 10$
   - Chebyshev 多項式で近似をしたいが、そのためにはまず「どこで評価するか」を決定しなくてはならない
   - コロケーション法における Chebyshev 多項式の評価点、あるいは積分で必要になる評価点 (Quadrature Point) の数を選択し、評価点を設定
   - Quadrature Point は、CompEcon の関数 `qnwlege` を使って Gauss-Legendre Quadrature に基づいて計算

スペクトラル法：アルゴリズム(続き)

2. 続き
   - 評価点が得られたので、n 次の Chevyshev 多項式の基底を各評価点上で計算
   - Chebyshev 多項式の基底は次のようになる

$$\phi(k)=\begin{pmatrix} 1 & \cos(\arccos k_1) & \cdots & \cos((n-1)\times\arccos k_1) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \cos(\arccos k_n) & \cdots & \cos((n-1)\times\arccos k_n) \end{pmatrix}$$

   Chebyshev 多項式を作る場合、k の各要素は必ず $[-1,1]$ 区間内
   - 例えば、単純な多項式の場合

$$\phi(k)=\begin{pmatrix} 1 & k_1 & \cdots & k_1^{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & k_m & \cdots & k_m^{n-1} \end{pmatrix}$$

   - 初期値 a_0 を決定する。どうやって決定するかが大きな問題!

スペクトラル法：アルゴリズム(続き)

- 残差関数(Residual Function)を作る
  - 後でRoot-Findingをするため、残差関数を求めたい変数a_0のみの関数にしたい

4. コロケーション法(選点法)では各Chebyshevノードにおいてゼロになるようなa_0を求める
   - よって残差関数は、

$$
\begin{aligned}
\texttt{residual} &= \texttt{resid\_collocation(a0)} \\
&= \begin{bmatrix} R\left(k_1; a_0\right) \\ \vdots \\ R\left(k_n; a_0\right) \end{bmatrix} \sim n \times 1
\end{aligned}
$$

スペクトラル法：アルゴリズム(続き)

4. **求めたいのは、**

$$\int_{\underline{k}}^{\overline{k}} \psi_i(k)\, R(k; a_0)\, dk, \;\; i = 1, \ldots n,$$

ガラーキン法では重みとして Chebyshev **多項式の基底である** $\psi_i(k) = \phi_i(k)\ (i = 1, \ldots, n)$ **を使用**

- **数値積分をする計算ために、** Gauss-Legendre Quadrature **に基づいて近似**

$$\int_{\underline{k}}^{\overline{k}} \psi_i(k)\, R(k; a_0)\, dk \approx \begin{bmatrix} \sum_{l=1}^{qn} \omega_l \cdot \phi_{1,l}(k) \cdot R(k_l; a_0) \\ \vdots \\ \sum_{l=1}^{qn} \omega_l \cdot \phi_{n,l}(k) \cdot R(k_l; a_0) \end{bmatrix} \sim n \times 1$$

[Matlab Code 3]

Endogenous Gridpoint Method

- Carroll (2006,EL)：**安定性& 計算速度**
  - **一番のポイントは、次期の資本 a' の方を有限個に区切る点**
  - **最適化は必要ない**
  - **近似法は必要**
- **家計問題を考える**

$$
\begin{aligned}
v(a) &= \max\{u(c)+\beta v(a')\}, \\
\text{s.t.}\;\; c+a' &= (1+r)a+w
\end{aligned}
$$

Endogenous Gridpoint Method(続き)

- 価値関数の右辺第2項を次のように定義

$$\Omega\left(a'\right) = \beta v(a'),$$

  - 現金保有高 $x \equiv (1+r)a + w$ 上の消費関数を $\tilde{h}(x)$ とする
- 包絡線定理から

$$\begin{aligned} c^{-\gamma} = \Omega'\left(a'\right) &= \beta(1+r)\left(c'\right)^{-\gamma}, \\ &= \beta(1+r)\tilde{h}(x')^{-\gamma}. \end{aligned}$$

  - 消費関数 $\tilde{h}(x)$ の初期値を (例えば線形ルールで) 与えれば、各資産水準 $\{a^1, \dots, a^n\}$ 上で $\Omega'\left(a'\right)$ を容易に計算出来

Endogenous Gridpoint Method (続き)

- CRRA 型の限界効用関数は逆関数を計算する事が可能
- $c^i = [\Omega'(a^i)]^{-\frac{1}{\gamma}}$ が各グリッド a^i 上で計算出来る
- 現金保有高は $x_t^i = c_t^i + a^i$ で求められるので、現金保有高上の新しい政策関数 $\tilde{h}(x)$ を導出する事が可能
- 繰り返していき、収束したものがオイラー方程式を満たす政策関数
    - Carroll (2006) が使用した Matlab(&Mathematica) コードは Carroll の HP に置いてある
    - 次章の Bewley モデルは EGM で計算 (Fortran 90/95)

Pertubation Method

- 線形近似や二次近似で基本になるテイラー展開の考え方を更に発展させたもので、五次近似とかまで応用可能。
- ただし、手で計算できないので、Mathematica が必要になる (Matlab の数式処理は二階微分までなので注意)。
- 詳細は Judd (1998) を参照
- DSGE モデルの推計に Fernandez-Villaverde and Ramirez (2007,NBER Macro Annual) が使っている
  - Fortran&Mathematica

数値計算の精度について

- 結果の頑健性をテストするために、計算精度について気にする必要がある
- 数値計算精度の測り方及び実際に測った際の誤差の程度について、いろいろな研究が存在
  - Judd (1992,JET)：オイラー方程式ベース
  - den Haan and Marcet (1994,REStud)：シミュレーション
  - Santos (2000,ECTA)、Santos and Peralta-Alva (2003)：
  - Aruoba, et al.(2006)：

労働供給を内生化

- Aruoba, Fernandez-Villaverde, and Rubio-Ramirez (2006)
  - Pertubation Method(First-Order, Second-Oreder, Log)
  - Spectoral Method(Chebyshev)
  - Finite Element Method
  - Value Function Iteration
- **ベンチマークモデル**

$$U = E_0 \sum_{t=1}^{\infty} \beta^{t-1} \frac{\left(c_t^{\theta}(1-l_t)^{1-\theta}\right)^{1-\tau}}{1-\tau},$$

s.t.

$$c_t + i_t = y_t, \; k_{t+1} = (1-\delta)k_t + i_t,$$

$$y_t = e^{z_t} k_t^{\alpha} l_t^{1-\alpha}, \; z_t = \rho z_{t-1} + \varepsilon_t, \; \varepsilon_t \sim \mathcal{N}(0, \sigma^2).$$

労働供給を内生化：主要結果

[Aruoba et al., 2006, JEDC]

- 線形近似& 二次近似：定常状態近傍での近似はOK
- 摂動法：
- FEM：コードを書くのが大変だが精度とスピードの両面で高パフォーマンス
- VFI：精度は高いものの収束までに時間がかかるのが問題
- Aruoba, et al. (2006) で使用されたコードは下記のHPからダウンロード可能
    - http://www.econ.upenn.edu/~jesusfv/companion.htm
    - 摂動法はMathematica、線形近似はMatlab、それ以外はFortranで書かれている

Ramsey から Bewley へ

- 一部門最適成長モデルにおいて代表的個人を仮定できるのは、保険市場の完備性を仮定しているため
  - 社会計画者が解く配分問題と市場において達成された配分は一致
- 経済活動をしている主体(家計や企業)は固有のリスク(Idiosyncratic Risk)に直面しているかもしれない
  - 不完備市場の一般均衡
  - Bewley モデル (Bewley, 1977,1980a,1980b,1986)
  - 経済格差や景気循環(Imrohoroglu, 1989, JPE)のコストを考える際のベンチマーク

Bewley モデル：目的関数

- 基本的なセットアップは、Aiyagari (1994) や Huggett (1993,1997)
  - 無限期間の生産経済
  - 経済主体は測度 1 で $[0,1]$ 区間の連続体上に無限人存在
  - 0 期には同質的
- 各家計の目的関数

$$E_0 \sum_{t=0}^{\infty} \beta^t \frac{c_t^{1-\gamma}}{1-\gamma}$$

  - 不確実性があるので期待値を最大化

予算制約

- 資産 a_t と労働所得 y_t を所与
- 予算制約

$$c_t + a_{t+1} \leq (1 + r_t) a_t + y_t$$

  - r_t は利子率
  - 流動性制約：$a_t \geq 0$
  - 借入制約：$a_t \geq \varphi$

固有リスク(Idiosyncratic Risk)

- 家計は無数に存在
  - 推移確率(マルコフ環)：$\pi\left(e^j | e^i\right) > 0$
  - 人口割合の推移も表している
  - 一意的な定常分布：π^*
- 動労所得：$y_t = w_t e_t$
  - 労働保有量(Labor Endowment)：$\left\{e^1, \ldots, e^J\right\}$
  - 賃金：w_t
- 自然な借入制約(Natural Borrowing Limit)

$$a_t \geq -\frac{we^1}{r} = \varphi$$

Income Fluctuation Problem

- Chamberlain and Wilson (2000,RED)
- **家計の最適化問題**

$$v(a,e) = \max\left\{\frac{c^{1-\gamma}}{1-\gamma} + \beta\sum_{e'}\pi(e'|e)\,v(a',e')\right\},$$

$$\text{s.t.}\ \ c + a' \leq (1+r)\,a + we,\ \ a' \geq 0.$$

  - **政策関数**：$g(a,e)$
- FOC：

$$u'(c) \geq \beta(1+r)\sum\pi(e'|e)\,u'(c'),$$

いくつかの基本的な性質

- $g(a, e)$ は a に関する連続・増加関数
- $\beta(1+r) < 1$ とする。このとき、ある $\tilde{x} = we + (1+r)\tilde{a} \geq we_1$ が存在して、あらゆる $x \leq \tilde{x}$ で、$h(a, e) = x$ 及び $g(a, e) = 0$ が成立する。また、状態空間 $[0, \tilde{a}) \times \{e\}$ 上でオイラー方程式は強い不等式で成立する
- ある $a_{\max}$ が存在し、$a_{\max} > a_0$ という初期値を持つ $\{a_t\}_{t=0}^{\infty}$ は、あらゆる t について $a_{\max} > a_t$ である
- $\beta(1+r) < 1$ とする。このとき、あらゆる $a \geq 0$ について、ある $e \in E$ が存在して、$g(a, e) < a$ が成り立つ
- 予備的貯蓄 (Precautionary Saving)：$\beta(1+r) < 1$

推移確率と分布関数

- $x = (a, e)$
- 現在の状態が $x \in \hat{\mathcal{X}}$ である家計が次期に $B \in \mathcal{B}(\hat{\mathcal{X}})$ が実現する確率を推移関数 $Q : \hat{\mathcal{X}} \times \mathcal{B}(\hat{\mathcal{X}}) \to [0, 1]$ と書く

$$
\begin{aligned}
Q(x, B) &= \pi\left(\left\{e' \in \mathcal{E} \middle| \left(a(a, e), e'\right) \in B\right\} \middle| e\right), \\
&= \sum_{e' \in B} \begin{cases} \pi(e'|e) & \text{if } a(a, e) \in B, \\ 0 & \text{otherwise.} \end{cases} .
\end{aligned}
$$

- $\Psi(B)$：$B \in \mathcal{B}(\mathcal{X})$ 上で定義された確率測度
  - Ψ は家計の状態が (a, e) となる割合
  - あらゆる $B \in \mathcal{B}(\mathcal{X})$ について、

$$
\Psi_{t+1}(B) = \int_{\hat{\mathcal{X}}} Q(x, B)\, d\Psi_t \equiv \Gamma(\Psi(B))
$$

マクロ経済(集計経済)

- 総資本供給

$$K = \int g(a, e)\, d\Psi$$

- 総労働供給：

$$N = \sum e\pi^*(e)$$

  - $\pi^*(e)$ は定常分布での e の実現確率
- 企業の生産関数

$$Y = K^\theta N^{1-\theta}$$

競争均衡の定義

- 定常再帰的競争均衡とは、以下を満たす価値関数 v、政策関数 g、利子率 r、賃金 w 及び分布関数 Ψ^* である。

1. 利子率 r 及び賃金 w を所与としたとき、v は家計のベルマン方程式を満たし、$g(\cdot)$ はそれに伴なう政策関数になる。
2. (企業は利潤を最大化している。即ち、

$$r = \theta K^{\theta-1} N^{1-\theta} - \delta, \; w = (1-\theta) K^{\theta} N^{-\theta}.$$

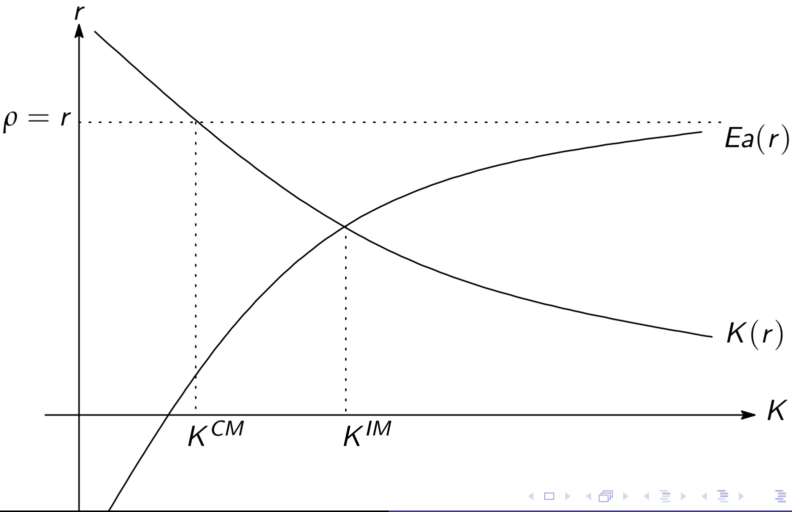
3. 財、労働及び資本市場は均衡している。
4. 資産分布は時間を通じて不変である。

均衡価格を見つける

- **証明のイメージ**

1. **総需要関数** $K(r)$ **が利子率に関して連続で、単調に右下がりになる。**
2. **各利子率** r **の下で定常測度** Ψ_r **が存在して、総供給関数** $Ea(r) = \int ad\Psi$ **が**Well-Defined**となる。すなわち、積分がきちんと定義できることを証明。更に、**r**に関して連続であることを証明する。**
   - Theorem 2：Hopenhayn and Prescott (1992,ECTA)
3. $Ea(r)$ **が右上がりになる事を示し、均衡利子率が存在することを証明する。**

競争均衡の存在



Bewley モデルの数値計算

1. パラメターを設定する。このとき、所得リスクについても特定化
2. 利子率 r を任意に与える。総労働供給 N は決まっているため、企業の一階条件から (w, K) を逆算する事が出来る
3. 要素価格 (r, w) が一つ与えられたので、家計の最適化問題を解く事が出来る
4. 各家計の意思決定関数が得られたので、推移関数に基づいて家計分布の推移を計算し、定常分布 Ψ_r^* を導出する
5. 定常分布 Ψ_r^* を使って家計行動を積分して、総資本供給 $Ea(r)$ を計算
6. 総資本需要 $K(r)$ と総資本供給 $Ea(r)$ が一致する r を見つけるまで、$2-5$ を繰り返す。

所得リスクの近似

- **ちょっと寄り道をして、AR(1)過程の近似する方法を説明**
  - **所得リスクや生産性過程などはAR(1)で推計**
  - **そのままだと、期待値を計算する場合、積分の扱いが面倒**
  - Tauchen (1986,EL)、Tauchen and Hussey (1991,ECTA)
  - **資産価格決定モデルにも使える**
- AR(1)**過程**
  - $y_t = \log d_t$

$$y_t = \lambda y_{t-1} + \tilde{\varepsilon}_t, \tilde{\varepsilon}_t \sim \mathcal{N}\left(0, \sigma_\varepsilon^2\right)$$

$$\log d_t = \lambda \log d_{t-1} + \sigma_y \left(1 - \lambda^2\right)^{\frac{1}{2}} \varepsilon_t,\ \varepsilon_t \sim \mathcal{N}(0, 1)$$

Tauchen 法(1)

- Tauchen 近似
    - 有限マルコフ環で近似
    - 例えば、3σ で両端を決め、7 個の状態とする
    - $\bar{y}^N = 3\sigma_y$、$\bar{y}^1 = -3\sigma_y$ で状態空間は $Y^{\log} = \{-3\sigma_y, -2\sigma_y, -\sigma_y, 0, \sigma_y, 2\sigma_y, 3\sigma_y\}$
    - 7 つの区間を次のように定義

$$I_1 = [3\sigma_y, \frac{5}{2}\sigma_y), I_2 = [-\frac{5}{2}\sigma_y, -\frac{3}{2}\sigma_y),$$
$$I_3 = [-\frac{3}{2}\sigma_y, -\frac{1}{2}\sigma_y), I_4 = [-\frac{1}{2}\sigma_y, \frac{1}{2}\sigma_y),$$
$$I_5 = [\frac{1}{2}\sigma_y, \frac{3}{2}\sigma_y), I_6 = [\frac{3}{2}\sigma_y, \frac{5}{2}\sigma_y),$$
$$I_7 = [\frac{5}{2}\sigma_y, 3\sigma_y)$$

Tauchen 法(2)

- **各状態間の推移確率を計算**
  - **現在、ある状態** $\bar{y}^i = \log d \in Y^{\log}$ **が実現**
  - **次期に** $\bar{y}^j = \log d' \in Y^{\log}$ **が実現する確率**

$$\pi_{ij} = \Pr\left(\log d' = \bar{y}^j \,|\, \log d = \bar{y}^i\right) = \int_{I_j} \frac{1}{\sqrt{2\pi}\sigma_\varepsilon} e^{-\frac{1}{2}\frac{(x-\log \bar{d})^2}{\sigma_\varepsilon}} dx$$

  - **各状態の間隔を** $w = \bar{y}^k - \bar{y}^{k-1}$ **とする**
  - **各** i **について、もし** $j \in [2, N-1]$ **ならば、**

$$\begin{aligned}
\pi_{ij} &= \Pr\left[\bar{y}^j - \frac{w}{2} \le \bar{y}^j \le \bar{y}^j + \frac{w}{2}\right] \\
&= \Pr\left[\bar{y}^j - \frac{w}{2} \le \lambda \bar{y}^i + \varepsilon_t \le \bar{y}^j + \frac{w}{2}\right] \\
&= F\left(\frac{\bar{y}^j - \lambda \bar{y}^i + \frac{w}{2}}{\sigma_\varepsilon}\right) - F\left(\frac{\bar{y}^j - \lambda \bar{y}^i - \frac{w}{2}}{\sigma_\varepsilon}\right)
\end{aligned}$$

Tauchen 法 (3)

- 最後に対数を取っていた AR(1) 過程 $\{\log d_t\}$ を $\{d_t\}$ に戻し、計算のし易さのための基準化
  - 対数での状態空間 $Y^{\log}$ を指数を取って元に戻す
  - 期待値 $N = \sum_{d \in D} d\pi_\infty(d)$ が 1 になるように $Y^{\exp}$ を基準化すると便利
  - N で基準化すると

$$
\begin{aligned}
E &= \{d_1, \ldots, d_7\} \\
&= \left\{ \frac{e^{-3\sigma_y}}{N}, \frac{e^{-2\sigma_y}}{N}, \frac{e^{-\sigma_y}}{N}, \frac{1}{N}, \frac{e^{\sigma_y}}{N}, \frac{e^{2\sigma_y}}{N}, \frac{e^{3\sigma_\varepsilon}}{N} \right\}
\end{aligned}
$$

- Floden (2008,EL)：近似精度について分析
  - HP に Matlab コードあり

分布関数の計算(1)：シミュレーションベース

- Monte Carlo シミュレーションを使って計算
  - 政策関数 $g(a,e)$ は既知なので、$\{a_t^i\}_{t=1}^T$ をシミュレーション
  - 例えば、5000 人分の第 i 家計について、11000 期間 $(T = 11000)$ 計算す
  - $a_{t+1}^i = g(a_t^i, e_t^i)$、$(t = 1, \ldots, 11000)$
  - 5000 人分を集計して平均値を計算すれば、総資本になる
- 特徴
  - 統計データと同じように扱えるので、平均だけでなく分散やジニ係数も簡単に計算
  - シミュレーション誤差が大きい

分布関数の計算(2)：前向きに解く

- 分布関数を離散的に近似：グリッドを設定する
- 分布関数 $\psi_0\left(a^i, e\right)$ の初期値を設定
  - 分布は収束することが知られているため、一様分布などからスタートしてOK
- 現時点で (a, e) である家計が次期に $\hat{a}'$ だけ貯蓄
  - しかし、必ずしも $\hat{a}'$ がグリッド上にあるとは限らない
  - 定数 ω を次のように設定
  - $[a_\ell, a_h]$ は $\hat{a}'$ を挟むグリッド値

$$\omega = \frac{\hat{a}' - a_\ell}{a_h - a_\ell}, \ \hat{a}' \in [a_\ell, a_h]$$

- (a, e) である家計は次期に a_ℓ と a_h に振り分けられる

$$\begin{cases} \pi(e'|e)(1-\omega)\psi_0\left(a, e\right) \\ \pi(e'|e)\omega\psi_0\left(a, e\right) \end{cases}$$

分布関数の計算(3)：逆関数を使う

- 政策関数の逆関数を計算

$$\tilde{a} = g^{-1}(a_l, e'),$$

- 累積密度関数を計算

$$\Psi_1(a_l, e_j) = \Psi_1(a_l, e_j) + \pi(e_j|e_i)\Psi_0(\tilde{a}, e_i),$$

- 積分を計算するには、一工夫が必要
    - Marimon and Scott (1999) 内の Rios-Rull (1999) を参照

数値計算結果

- 数値計算結果
  - 政策関数& 分布関数
  - 計算時間
  - ジニ係数
- コード(Matlab&Fortran)

[smodel2]
[bewley]

- "State of the Art"
  - Chatterjee, Corbae, Nakajima and Rios-Rull (2007,ECTA)

何が問題なのか?

- Bewley モデルにマクロショックを加えたい
- 家計の意思決定に必要な次期の利子率 r_{t+1} にも不確実性

$$u'(c_t) \geq \beta E_t (1 + r_{t+1}) u'(c_{t+1})$$

- 家計は次期の利子率を予測する必要がある
- 利子率 r_{t+1} は総資本 K_{t+1} の関数であり、総資本は分布関数 $K_{t+1} = \int a d\Psi_{t+1}$ によって決定
  - $t+1$ 期の分布関数 Ψ_{t+1} は現在の分布関数 Ψ_t と推移法則から決定
- 次期の利子率を計算するためには各家計が現在の資産分布 Ψ_t に関する情報を知っている必要がある
  - 状態変数に資産分布を入れる?

Approximate Aggregation

- 状態変数に資産分布を含めることは無理!
  - 理論的(数学的)には正しい
  - 無限次元の状態変数
  - 本当に必要な状態変数は?(Diffie, et al. 1994, ECTA)
- 基本的なアイディア
  - 分布関数そのものを状態変数に含めるのではなく、分布のモーメント情報のみを状態変数として、近似的な予測関数を用いる
  - Krusell and Smith (1998)、den Haan (1997)
- Bewley モデルにマクロショックを加えたモデルの数値計算は今でも数値計算研究のフロンティア
  - den Haan, Judd and Juillard (2005)

Krusell and Smith モデル(1)

- マクロショックと固有ショック
- マクロ生産性に関する集計ショック：$A_t \in \{A^g, A^b\}$
  - 好景気 $A_t = A^g$ と不況 $A_t = A^b$

$$\Pi_A = \begin{pmatrix} \pi_{gg'} & \pi_{gb'} \\ \pi_{bg'} & \pi_{bb'} \end{pmatrix}.$$

- 固有リスクの状態も 2 種類：雇用と失業
- (e, A) の推移確率行列

$$\Pi = \Pi_e \otimes \Pi_A = \begin{pmatrix} \pi_{gg'11'} & \pi_{gb'11'} & \pi_{gg'10'} & \pi_{gb'10'} \\ \pi_{bg'11'} & \pi_{bb'11'} & \pi_{bg'10'} & \pi_{bb'10'} \\ \pi_{gg'01'} & \pi_{gb'01'} & \pi_{gg'00'} & \pi_{gb'00'} \\ \pi_{bg'01'} & \pi_{bb'01'} & \pi_{bg'00'} & \pi_{bb'00'} \end{pmatrix}$$

Krusell and Smith モデル(2)

- **追加的条件：その**(1)

$$\pi_{AA'00'} + \pi_{AA'01'} = \pi_{AA'10'} + \pi_{AA'11'} = \pi_{AA'}$$
$$u_A \frac{\pi_{AA'00'}}{\pi_{AA'}} + (1 - u_A)\frac{\pi_{AA'10'}}{\pi_{AA'}} = u_{A'}.$$

- **追加的条件：その**(2)

$$\pi_{bb'00'} = 0.75 \times \pi_{bg'00'},$$
$$\pi_{gg'00'} = 1.25 \times \pi_{gb'00'}$$

Krusell and Smith モデル(3)

- パラメーターは全て Krusell and Smith (1998) に準拠
  - $\beta = 0.99$, $\delta = 0.025$.
  - $\gamma = \{1, 2, 3, 4, 5\}$
  - $\alpha = 0.36$
  - 好況の平均持続期間 $= 8$ 期間
  - 不況の平均持続期間 $= 8$ 期間
  - 好況時の生産性 $= 1.01$
  - 不況時の生産性 $= 0.99$
  - 好況時の失業率 $= 0.04$
  - 不況時の失業率 $= 0.10$
  - 好況時の平均失業持続期間 $= 1.5$
  - 不況時の平均失業持続期間 $= 2.5$

Krusell and Smith アルゴリズム(1)

1. **予測関数を特定化し、パラメターを決める**
   - Krusell and Smith (1998) **の場合、**

$$\ln K' = b_0(A) + b_1(A)\ln K$$

2. **予測関数を所与として、家計の最適化問題を解く**
   - **ベルマン方程式及びオイラー不等式は、状態変数に総資本** K **と** A **が含まれる以外は前と同じ**

$$v(a, e; K, A) = \max\left\{\frac{c^{1-\gamma}}{1-\gamma} + \beta E v\left(a', e'; K', A\right)\right\},$$
$$\text{s.t.}\ \ c + a' \leq (1+r)a + we,\ \ a' \geq 0.$$

   - (a, K) **上のグリッド数：** $\#a = 100$、$\#K = 20$
   - K **の平均値(既知)の周辺で** K **のグリッドを取る**
   - VFI **でも** EGM **でも何でも** OK!

Krusell and Smith アルゴリズム(2)

3. 政策関数 $a' = g(a, e; K, A)$ が求められたら、N 人を T 期間シミュレーション
    - Krusell and Smith (1998) は $N = 5000$、$T = 11000$ として、定常性を求めるためにシミュレーションの始めの1000期間を切り捨てている。
    - 前向きに分布を計算する事も可能
    - 初期分布に、定常均衡で得られた分布を利用すると精度が高まる

Krusell and Smith アルゴリズム(3)

4. ステップ3で得られたデータを使って、新しい予測関数のパラメーター$(\hat{b}_0, \hat{b}_1)$を推計
5. 新しいパラメーターが古いパラメーターと十分に近くなったらストップ
6. ステップ5までで予測関数のパラメーターが収束したにも関わらずフィット(決定係数: R^2)が良くなければ、予測関数の形状を変えてステップ$2-5$を繰り返す

数値計算結果

- **予測関数：**

$$
\begin{aligned}
\ln K' &= 0.135 + 0.963 \ln K, \quad R^2 = 0.999999, \quad \text{if } A = A^g \\
\ln K' &= 0.123 + 0.965 \ln K, \quad R^2 = 0.999998, \quad \text{if } A = A^b
\end{aligned}
$$

- Young (2002) **は二次以上の情報を加えても精度の向上はそれほどない事を明らかにしている**
- **世代重複モデルにすると、影響が出てくる**
  - **世代の影響を明示的に考えると、意志決定の年齢毎の違いが大きくなるため**

State of the Art

- **マクロショックがある経済**
  - Algon, allais and den Haan (2007)、Reiter (2002,2008)
  - Krusell, Mukoyama, and Sahin (2007)、Mukoyama and Sahin (2005)
  - Storesletten, Telmer, and Yaron (2007,RED)
  - Krueger and Kubler (2004,JEDC)
- **ダウンロード可能なコード**
  - den Haan (1997)：http://weber.ucsd.edu/~wdenhaan/soft.html
  - Krusell and Smith (1998)：http://www.econ.yale.edu/smith/
  - Reiter (2008)：http://elaine.ihs.ac.at/~mreiter/research.html