

第4回：オイラー方程式と多項式近似

初稿・再稿：反映済み

1 イントロダクション

連載第3回では、動的計画法 (dynamic programming) を用いて新古典派成長モデルを数値的に解く方法を示した。そこでは、無人島に住むロビンソン・クルーソー、すなわち社会計画者 (social planner) の問題を扱った。社会計画者の最適化問題では、ベルマン方程式を (現在から将来にわたっての効用の流列の割引現在価値である) 価値関数について解くことで、最適な消費や貯蓄の経路を求めることができた¹。この場合、無人島に住むロビンソン・クルーソーは、消費や貯蓄と同時に生産を行っていた。この問題は社会計画者問題 (social planner's problem) と呼ばれる。

しかし、無人島にロビンソン・クルーソーのほかに、彼に代わって生産を行う別の個人がいるとするとどうだろうか。この場合、もともとのロビンソン・クルーソーの問題を分権経済 (decentralized economy) における問題として書き換えることができる。よく知られた事実として、厚生経済学の第二基本定理 (second fundamental theorem of welfare economics) が成り立つ場合、分権経済における配分 (allocation) と社会計画者 = ロビンソン・クルーソーの最適化問題における配分は一致する²。

ところが、厚生経済学の第二基本定理は、分権経済において価格に歪みをもたらす税 (distortionary tax) がある場合は成り立たない。また、次回でより詳しく扱うように、ニューケインジアン・モデルのような価格の硬直性により相対価格に歪みがあるモデルにおいても、分権経済における配分は、一般には、社会計画者問題における最適な配分と一致しない³。このような場合には、社会計画者問題の代わりに、分権経済の問題を考える必要がある。このときに有用なのが、今回説明するオイラー方程式 (Euler equation) あるいは均衡条件 (equilibrium condition) を用いたアプローチであり、時間反復法と呼ばれる。

第3回ではまた、ベルマン方程式における来期の価値関数を近似する方法についても説明した。

¹ 異質な個人 (heterogeneous agent) の問題においても、このアプローチは適用可能であり、第6回以降で詳しく扱う。

² 厚生経済学の第二基本定理については、例えば、Mas-Colell, Whinston, and Green (1995) ch. 16 を参照。

³ ニューケインジアン・モデルにおいて、社会計画者問題における配分を達成するような金融政策を、最適金融政策 (optimal monetary policy) と呼ぶ。英語では Gali (2008)、Woodford (2003) などに詳しい説明がある。日本語の解説としては加藤 (2006)、木村・藤原・黒住 (2005) などがある。

そこで説明した線形補間やスプライン補間は、いずれも状態空間を分割するグリッドの間を異なる関数で近似するが、今回説明する多項式を用いた補間は、状態空間全体を一つの関数で近似する。前者の方がスムーズでない関数を上手く近似できるのに対して、後者の方がグリッドの数を少なくできる(ことが多い)など、それぞれのアプローチには長所と短所がある⁴。また、異なる近似の方法を用いることで、前回説明したように解の精度も変わってくる。同じ問題に異なる方法を適用して求めた解の比較を行い、理解をより深めることも、今回の目的の一つである。

2 分権経済と時間反復法

2.1 分権経済における無限期間モデル

前回扱った、無限期間モデルにおける社会計画者(ロビンソン・クルーソー)の最適化問題は以下の通りであった⁵。

$$\begin{aligned} & \max_{\{c_t, k_{t+1}\}_{t=0}^{\infty}} \sum_{t=0}^{\infty} \beta^t u(c_t) \\ & \text{subject to} \\ & c_t + k_{t+1} = f(k_t), k_0 \text{ given} \end{aligned}$$

ここで、 c_t は t 期における消費、 k_t は t 期初の資本である。 $u(c_t)$ は効用関数、 $f(k_t)$ は生産関数であり、それぞれ標準的な仮定 ($u'(c_t) > 0$ および $u''(c_t) < 0$ 、 $f'(k_t) > 0$ および $f''(k_t) < 0$) を満たす。前回 は動的計画法を用いてこの問題を解いた(消費または貯蓄を今期の資本の関数として求めた)が、今回は別のアプローチを用いる。ラグランジュ未定乗数法を用いると、動的ラグランジュアン (dynamic Lagrangean) は以下のように書くことができる⁶。

$$\mathcal{L}_0 \equiv \max_{\{c_t, k_{t+1}\}_{t=0}^{\infty}} \sum_{t=0}^{\infty} \beta^t \{u(c_t) + \lambda_t (f(k_t) - k_{t+1} - c_t)\}$$

⁴ 借り入れ制約やゼロ金利制約などによって、状態空間のある点において関数のよじれ (kink) が生じている場合には、一般には、多項式近似の近似の精度はよくない。この場合は、線形補間やスプライン補間を用いて、関数のよじれが生じている点のまわりにグリッドの点を多く配置するなどの工夫が必要になる。

⁵ この節では、簡単化のために、資本の減耗率は $\delta = 1$ とする。前回示したように、この場合はモデルの解析的な解が存在する。

⁶ ラグランジュ未定乗数法については、経済数学の教科書、例えば西村 (1990)、Sundaram (1996) などを参照。

ここで、 λ_t はラグランジュ乗数 (Lagrange multiplier) と呼ばれる。 (c_t, k_{t+1}) のそれぞれについて一階条件を求めると、

$$\begin{aligned}\frac{\partial \mathcal{L}_0}{\partial c_t} &= u'(c_t) - \lambda_t = 0 \\ \frac{\partial \mathcal{L}_0}{\partial k_{t+1}} &= -\lambda_t + \beta \lambda_{t+1} f'(k_{t+1}) = 0\end{aligned}$$

となる。 c_t についての微分から、ラグランジュ乗数は消費の限界効用と等しくなる⁷。これら 2 つの式をまとめると、

$$u'(c_t) = \beta u'(c_{t+1}) f'(k_{t+1})$$

となる。ここで求めた式は、オイラー方程式と呼ばれる。この式は均衡条件とも呼ばれる、解の必要条件である⁸。すなわち、最適化問題の解を $c_t = h(k_t)$ とすると、関数 $h(k_t)$ は以下の式を満たす⁹。

$$\begin{aligned}u'(h(k_t)) &= \beta u'(h(k_{t+1})) f'(k_{t+1}) \\ &= \beta u'(h(f(k_t) - h(k_t))) f'(f(k_t) - h(k_t))\end{aligned}$$

したがって、この式を満たすような未知の関数 $h(k_t)$ を求めればよいことになる。

2.1.1 分権経済

ところで、ロビンソン・クルーソーが住む島に、もう一人仕事を分担する人がいるとしたらどうなるだろうか。仮に、この人はロビンソン・クルーソーから土地と種 (k_t) を価格 r_t で借りて作物を作るとしよう。利潤を π_t としたとき、この個人 (あるいは企業) の利潤最大化問題 (profit maximization problem) は以下のように書ける。

$$\pi_t = \max_{k_t} f(k_t) - r_t k_t$$

この問題の最適化条件は、端点解の場合を除くと、 $r_t = f'(k_t)$ となる。一方で、ロビンソン・クルーソーは、生産をする代わりに、地代の支払いを受けて作物を購入するとしよう¹⁰。すなわち、

⁷ このことから、ラグランジュ乗数は消費の影の価格 (shadow price) とも呼ばれる。すなわち、消費者が 1 円余分に予算を持っていたとき、その 1 円を消費することによる効用の増加分は λ_t にほぼ等しくなる。

⁸ オイラー方程式だけでは解の十分条件を満たさず、横断性条件 (transversality condition) も満たす必要がある。詳しくは上級マクロ経済学の教科書、例えば Romer (2011)、齊藤 (2006)などを参照。

⁹ 前回用いた次期の資本についての政策関数 $k_{t+1} = g(k_t)$ は、この場合 $k_{t+1} = f(k_t) - h(k_t)$ と書ける。

¹⁰ ロビンソン・クルーソーはこの企業を所有し、今期の最適な消費と貯蓄の組み合わせを決めた後に、利潤は支払われるものとする。

以下の問題を解く。

$$\begin{aligned} & \max_{\{c_t, k_{t+1}\}_{t=0}^{\infty}} \sum_{t=0}^{\infty} \beta^t u(c_t) \\ & \text{subject to} \\ & c_t + k_{t+1} = r_t k_t + \pi_t, k_0 \text{ given} \end{aligned}$$

この問題を、同じように動的ラグランジュアンを用いて、 (c_t, k_{t+1}) について一階条件を求めた後にまとめると、

$$\begin{aligned} u'(c_t) &= \beta u'(c_{t+1}) r_{t+1} \\ &= \beta u'(c_{t+1}) f'(k_{t+1}) \end{aligned}$$

となり、同じオイラー方程式が得られる。また、予算制約式も整理してまとめると ($\pi_t = f(k_t) - r_t k_t$ に注意)、社会計画者の資源制約と同じ式 $c_t + k_{t+1} = f(k_t)$ が得られる。

ここで、さらにモデルを拡張して、政府の存在を考えて、ロビンソン・クルーソーは地代による所得の一部を税率 τ で政府に税金として納めるとしたらどうか。また政府は税収を元に移転 TR_t を行う。すなわち、政府の予算制約式は $\tau r_t k_t = TR_t$ として与えられる¹¹。消費者の動学最適化問題は、

$$\begin{aligned} & \max_{\{c_t, k_{t+1}\}_{t=0}^{\infty}} \sum_{t=0}^{\infty} \beta^t u(c_t) \\ & \text{subject to} \\ & c_t + k_{t+1} = (1 - \tau) r_t k_t + \pi_t + TR_t, k_0 \text{ given} \end{aligned}$$

このとき、動的ラグランジュアンを用いた解の必要条件、すなわちオイラー方程式は、

$$\begin{aligned} u'(c_t) &= \beta u'(c_{t+1}) (1 - \tau) r_{t+1} \\ &= \beta u'(c_{t+1}) (1 - \tau) f'(k_{t+1}) \end{aligned}$$

となり、社会計画者問題の場合と異なる。したがって、このオイラー方程式を用いて求めた最適化問題の解 $c_t = h(k_t)$ あるいは消費と資本の流列 $\{c_t, k_{t+1}\}_{t=0}^{\infty}$ も異なることになる。このようなケースでは、社会計画者問題をベルマン方程式を使って解くことはできず、別のアプローチが必要になる¹²。

¹¹ 今期の最適な消費と貯蓄の組み合わせを決めた後に、税収は所得移転として支払われる。

¹² 分権経済でも、動学最適化問題 (上の例では消費者の効用最大化問題) にベルマン方程式と価値関数反復法を適用することは可能であり、このような解の特徴付けは、再帰的競争均衡 (recursive competitive equilibrium) と呼ばれる。

2.2 時間反復法

前節で求めた無限期間モデルのオイラー方程式は、ベルマン方程式のときと同様に時間 t に依存しない。そのため、時間をあらわすサブスクリプト t を省略し、次期の変数、例えば k_{t+1} を単にプライムを付けた k' と置き換えることができる。以下では、簡単化と前回との比較のため、 $\tau = 0$ のケースを考える。次期の政策関数 $c' = h(k')$ と今期の資源制約式 $k' = f(k) - c$ を代入することで、以下の式を得る。

$$\begin{aligned} u'(c) &= \beta u'(c') f'(k') \\ &= \beta u'(h(k')) f'(k') \\ &= \beta u'(h(f(k) - c)) f'(f(k) - c) \end{aligned}$$

オイラー方程式が与えられたとき、それを政策関数 $c = h(k)$ について解くための方法の一つが、時間反復法 (**time iteration : TI**) と呼ばれる方法である。前回学んだ価値関数反復法 (VFI) と同様に、TI は、政策関数の流れを作り最終的に収束した関数を解として求める。すなわち、関数の流れは、任意の推測された関数 $h^{(0)}(k)$ から始まり、 $h^{(n-1)}(k)$ を所与として $h^{(n)}(k)$ を計算することで、 $h^{(n)}(k), n = 1, 2, \dots$ と続く。そして、関数の流れが収束するまで、つまり $h^{(n-1)}(k)$ と $h^{(n)}(k)$ が十分近くなるまで、計算を繰り返す。ここで、 $h^{(n-1)}(k)$ を所与とすると、新しい政策関数 $c = h^{(n)}(k)$ は、以下の式を c について解くことで求められる。

$$u'(c) = \beta u'(h^{(n-1)}(f(k) - c)) f'(f(k) - c)$$

これは、VFI において、 $V^{(n-1)}$ を所与としてベルマン方程式を解くことで $V^{(n)}$ を計算することとよく似ている¹³。ここでは、ベルマン方程式の右辺における最大化問題を解く代わりに、オイラー方程式を満たすような c を求める求根問題 (**root-finding problem**) を解くのである。またそのとき、グリッド上にない $h^{(n-1)}(k')$ の値について近似する必要がある。

る。詳しくは Cooley (1995) を参照。

¹³ VFI では、ベルマン方程式は以下で与えられる

$$V^{(n)}(k) = \max_{k'} \{u(f(k) - k') + \beta V^{(n-1)}(k')\}$$

2.3 時間反復法の数値計算

アルゴリズム

1. グリッド生成 状態空間の評価点を有限個のグリッドに区切る。今回は (前回の VFI と同様に) $N = 21$ 個とする。有限個のグリッド上における政策関数の値 $h^{(0)}(k^i)$ の初期値を当て推量する。VFI と異なり、政策関数の収束は必ずしも保証されていないため、初期値はより重要になる。
2. 収束の基準 収束の基準になるパラメータ ε を与える。
3. 近似・評価 近似点 k^i 上には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。 $h(k; \mathbf{b})$ をパラメータのベクトル \mathbf{b} を使って近似した時の、 k 上での政策関数の値とする。
4. 最適化 古い政策関数 $h^{(n-1)}(k)$ を所与として、各 k^i について、

$$u'(c) = \beta u'(h^{(n-1)}(f(k^i) - c; \mathbf{b})) f'(f(k^i) - c)$$

を c について解く。オイラー方程式の解である c を探すためには、各言語に備わっている最適化関数 (求根関数) を利用する。特に TI では、ここをどう工夫するかで計算精度および収束のスピードや安定性が変わってくる。このステップで新しい政策関数 $c = h^{(n)}(k^i)$ を得る。

5. 収束しているか確認 古い政策関数 $h^{(n-1)}$ と新しい政策関数 $h^{(n)}$ の距離を測る。あらゆる k^i について $\|h^{(n)}(k^i) - h^{(n-1)}(k^i)\| < \varepsilon$ であればストップ。そうでなければ、 $h^{(n)}$ を $h^{(n-1)}$ に代入して、ステップ 3 とステップ 4 を繰り返す。

ここで注意したいのが、前述の通り、VFI と異なり、TI による繰り返し計算が真の政策関数に収束する保証はないことだ¹⁴。このため、TI によって計算した解は、それが適切な解であるかどうか、シミュレーションの結果や計算誤差によってよく吟味する必要がある。

政策関数の初期値 $h^{(0)}(k^i), i = 1, 2, \dots, N$ は、VFI と比べてより重要であり、ここでは資本のグリッド上の値を用いている¹⁵。最適化には、MATLAB の場合、ニュートン法をベースとする関

¹⁴ Coleman (1991) は歪みをもたらす税のある確率的新古典派成長モデルにおいて解の存在を一般的に示した。解の存在や一意性についての議論は、Richter et al. (2014)、Sargent and Stachurski (2019) なども参照。

¹⁵ 例えば、次回紹介する非線形ニューケインジアン・モデルでは、初期値やアルゴリズムの違いによって異なる解に収束することが知られている。より複雑な例では、比較的簡単に解ける、対数線形近似したモデルの解を初期値として用いるなどの工夫が必要なものもある。

数 `fsolve` などが使用できる。ニュートン法では、初期値がゼロ点から離れすぎていると解が求まらないことがあるが、ここでは古い政策関数の値を初期値として用いている (このことから、 $h^{(0)}(k^i)$ の値が重要になる)。

MATLAB 関数

- `fsolve` : 第 2 回で紹介した `fzero` と同様、非線形方程式のゼロ点を探す。1 次元だけではなく多次元 of 非線形方程式系にも対応している。ニュートン法 (Newton's method) をベースとするが、より解が求まりやすいように信頼領域法 (trust-region methods) を用いるなどの工夫がされている。

図 1(a) は、TI によって得られた政策関数と解析的解から計算した真の政策関数である¹⁶。VFI と同様、高い精度で真の関数を近似できている。また、政策関数を収束基準としていることから、収束のスピードは VFI と比べて速い。およそ 10 回で繰り返し計算誤差は $\varepsilon = 10^{-5}$ 以下になる。もっとも、前回も説明したように、VFI においても政策関数を収束基準とすることで収束を速めることはできる。

オイラー方程式における計算誤差は、前回同様、以下の式で得られる:

$$\frac{\beta u'([g(k^i)]^\alpha - g(g(k^i)))f'(g(k^i))}{u'([k^i]^\alpha - g(k^i))}.$$

ここで、 $k' = g(k) = f(k) - h(k)$ である。しかし、TI においては、この計算誤差は繰り返し計算誤差とほぼ一致する。なぜなら、TI の最適化のステップではまさにこの計算誤差が各グリッド上でゼロとなるように c を求めているからである。したがって、ここでは、グリッドをさらに細かく取り、TI で使用したグリッドの間でも計算誤差をみることにする。図 1(b) は、グリッドを 201 個としたときのオイラー方程式の誤差の絶対値である。グラフが波打っているのは、TI で使用したグリッドでは、それ以外のグリッドと比べて、誤差が非常に小さくなる (繰り返し計算誤差とほぼ一致する) ためである。また、資本の量が少ないほど誤差は大きくなるが、これは真の政策関数 (あるいは価値関数) の曲率 (curvature) が資本がゼロに近づくにしたがって大きくなるためである。VFI と比較して、特に資本保有量が多いときには、TI は計算誤差でみても同等の精度を持っている

¹⁶ 関数形とパラメータは以下の通り。 $u(c) = c^{1-\gamma}/(1-\gamma)$, $f(k) = k^\alpha$, $\beta = 0.96$, $\gamma = 1.0$, $\alpha = 0.4$, $\delta = 1.0$. 資本グリッドは $[0.05, 0.5]$ を 21 個の点で等間隔に分割した。近似にはスプライン補間 (MATLAB の場合、関数 `interp1`) を用いた。

ことがわかる。資本の量が少ないと、計算誤差はやや大きくなる。計算誤差の別の見方として、誤差の絶対値の平均値と最大値に 10 を底とする対数をとったもの $(\log_{10} L_1, \log_{10} L_\infty)$ をみると、TI では $(-3.066, -2.011)$ 、VFI では $(-3.252, -2.230)$ となっている¹⁷。すなわち、最大値でみても計算誤差は $10^{-2} = 0.01$ 以下にとどまっており、解の近似の精度は良好であることがうかがえる。

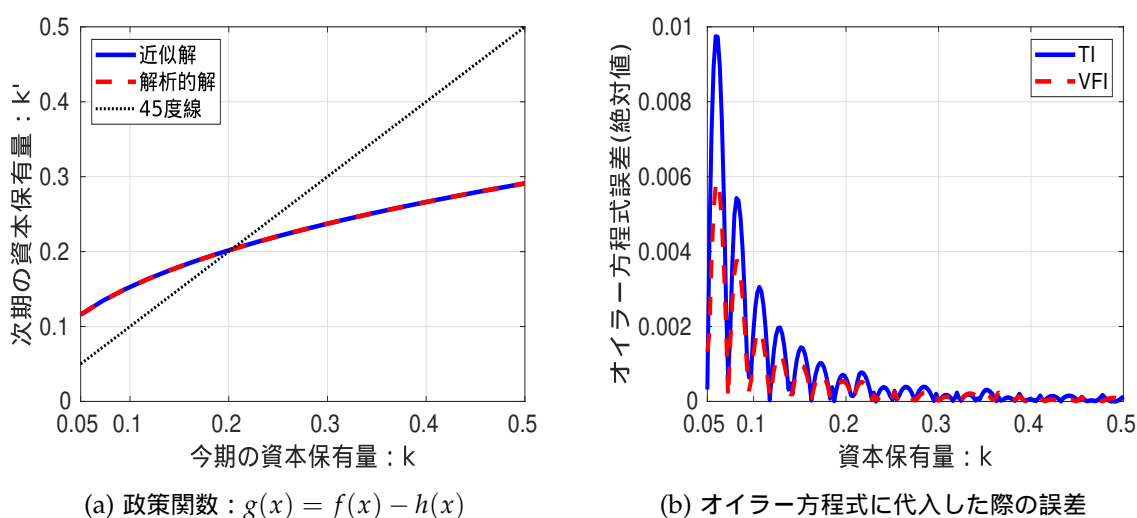


図 1: TI により計算された政策関数とオイラー方程式の誤差

MATLAB Code

- main_ti.m、EulerEq.m

3 多項式近似

これまでの、価値関数反復法において、次期の価値関数を近似するのに、線形補間やスプライン補間を用いた。これらの補間は、いずれも状態空間を分割するグリッドの間を異なる関数で近似するが、今度は状態空間全体を一つの多項式 (**polynomial**) で近似することを考えてみよう。この方法は、グリッドの数を少なくして計算時間を短縮したい場合に便利ことが多い。

いま、

$$y = g(x) = \frac{1}{1+x^2}, x \in [-5, 5]$$

が我々が知りたい真の関数であるとする¹⁸。しかし、我々はその真の姿を知らず、図 2(a) のように

¹⁷ ここで、 L_p はノルム (**norm**) と呼ばれる空間内の距離を測る指標である。すなわち、ベクトル $\mathbf{x} = [x_1, x_2, \dots, x_N]$ に対して、 $L_1 = N^{-1} \sum_{i=1}^N |x_i|$ 、 $L_\infty = \max\{|x_1|, \dots, |x_N|\}$ と定義される。

¹⁸ この関数は、ルンゲ現象 (**Runge's phenomenon**) といって、等間隔のグリッドによる多項式近似が上手くいかない

11 個の点の情報だけを持っているとしよう。我々はもとの関数の形状も知っているため、図 2(a) がスムーズな山型である事は推測できるが、限られた点の情報しか知らない場合、どのように近似すればよいのだろうか。

この 11 個の点を線形補間を使って近似すると図 2(b) の太線のようになる。点と点の間を直線で結んでいるため、データが存在する点の上で屈折しており、微分不可能になっていることがうかがえる。もちろん真の関数がこのような形状であれば問題ないが、もし真の関数が微分可能である場合 (実際、真の関数は微分可能な滑らかな曲線である)、線形近似は好ましくない性質を持つことになる。

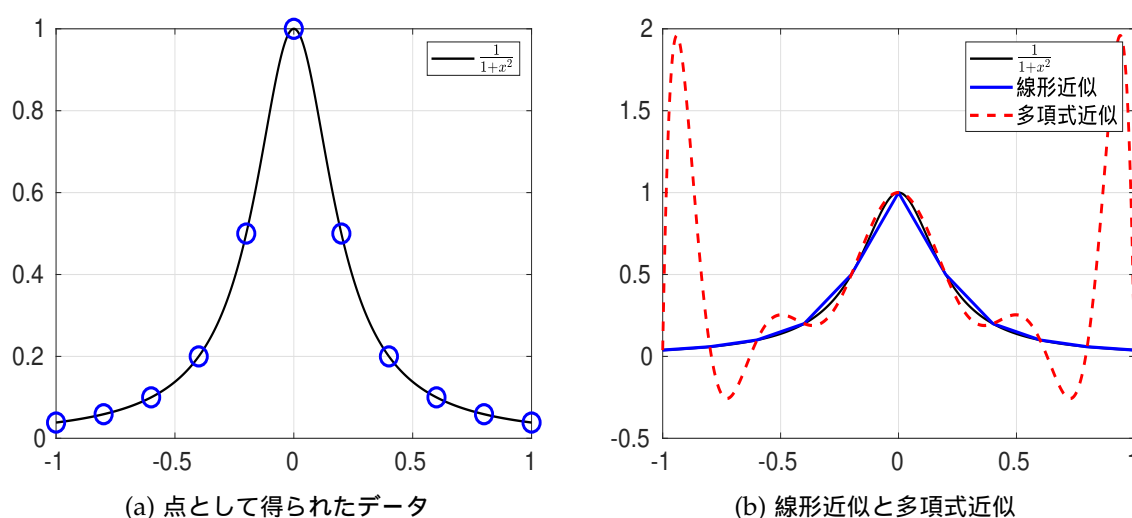


図 2: 線形近似と多項式近似による内挿

一方、図 2(b) の破線は、次のような 10 次の多項式を用いて 11 個の点の間を補間した場合である。

$$g(x; \theta) = \theta_0 + \theta_1 x + \theta_2 x^2 + \cdots + \theta_{10} x^{10}$$

ここで、11 つの未知の係数に対して 11 個の点の情報を用いて (例えば最小二乗法を用いて)、係数の値を求めることができる。しかし、図からわかるように、とくにグラフの両端において多項式による近似の精度はよくない。また、このような通常多項式の推定では、多重共線性 (multicollinearity) から、推定された係数の値は不安定になることがよく知られている。図 3(a) はそれぞれ x^2, x^4, x^8 の値であるが、これらの関数が同じ x に対してカバーする空間が似通っていることがうかがえる。

例として知られている。

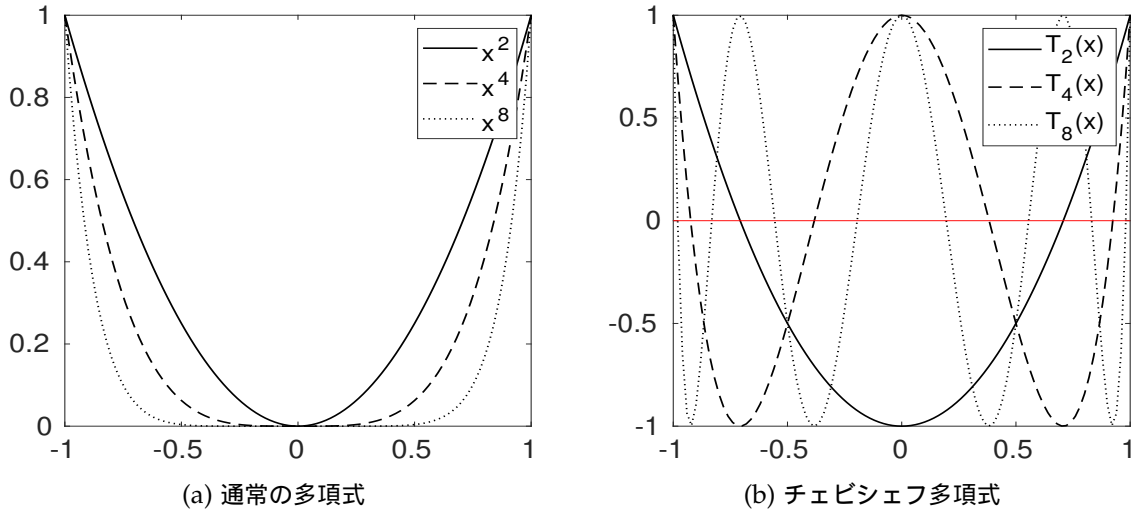


図 3: 多項式の比較

3.1 チェビシエフ多項式

通常の多項式の代わりによく用いられるのが、チェビシエフ多項式 (**Chebyshev polynomial**) と呼ばれる多項式である。 $i (= 1, 2, \dots, N-1)$ 次の基底関数 (**basis functions**) $T_i(x) : [-1, 1] \rightarrow [-1, 1]$ を組み合わせた $N-1$ 次の多項式を用いて、 N 個の点の間を補間することを考えよう:

$$g(x; \theta) = \theta_0 + \theta_1 T_1(x) + \theta_2 T_2(x) + \dots + \theta_{N-1} T_{N-1}(x)$$

チェビシエフ多項式あるいはチェビシエフ基底関数 $T_i(x)$ 、 $i = 0, 1, 2, \dots, N-1$ は次のように再帰的に書ける。

$$\begin{aligned} T_0(x) &= 1, \\ T_1(x) &= x, \\ T_2(x) &= 2x^2 - 1, \\ &\vdots \\ T_{N-1}(x) &= 2xT_{N-2}(x) - T_{N-3}(x) \end{aligned}$$

チェビシエフ多項式をデータにフィットさせてパラメータ $\theta = [\theta_0, \dots, \theta_{N-1}]$ を求めるには、評価点 (**collocation points**) と呼ばれる N 個の点 x_j 、 $j = 0, 1, \dots, N-1$ を用いる。評価点には 2 種類あり、一つはチェビシエフゼロ点 (**Chebyshev zeros**)、もう一つはチェビシエフ極値点

(Chebyshev extrema) と呼ばれ、それぞれ以下のように書ける

チェビシェフゼロ点: $x_0 = 0,$

$$x_j = \cos\left(\frac{(2j-1)\pi}{2(N-1)}\right) \text{ for } j = 1, \dots, N-1$$

チェビシェフ極値点: $x_j = \cos\left(\frac{j\pi}{N-1}\right) \text{ for } j = 0, 1, \dots, N-1$

すなわち、チェビシェフゼロ点は $T_{N-1}(x) = 0$ の解および $x_0 = 0$ であり、チェビシェフ極値点は $|T_{N-1}(x)| = 1$ の解である。例えば、もし 2 次のチェビシェフ多項式を用いると、評価点はチェビシェフゼロ点の場合 $x = [0, -\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}]$ で、チェビシェフ極値点の場合 $x = [0, -1, 1]$ でそれぞれ与えられる。図 3(b) はそれぞれ 2, 4, 8 次のチェビシェフ多項式 $T_2(x), T_4(x), T_8(x)$ の値をプロットしたものである。通常が多項式と異なり、関数の値が空間を満遍なく覆っていることがわかる。

評価点 x_j とその点における関数の値 $g(x_j)$ がわかると、多項式 $g(x_j; \theta)$ を関数の値 $g(x_j)$ にフィットさせることで、パラメータ θ が得られる。評価点における多項式とその値は、以下のように行列の形にまとめられる。

$$\begin{bmatrix} g(x_0) \\ g(x_1) \\ \vdots \\ g(x_{N-1}) \end{bmatrix} = \begin{bmatrix} 1 & T_1(x_0) & T_2(x_0) & \cdots & T_{N-1}(x_0) \\ 1 & T_1(x_1) & T_2(x_1) & \cdots & T_{N-1}(x_1) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & T_1(x_{N-1}) & T_2(x_{N-1}) & \cdots & T_{N-1}(x_{N-1}) \end{bmatrix} \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_{N-1} \end{bmatrix}$$

または、以下のように表せる。

$$g(x) = T(x)\theta$$

ここで、 $x = [x_0, \dots, x_{N-1}]$ である。もし行列 $T(x)$ が正則 (**nonsingular**) であれば、その行列は逆行列を持ち、 $\theta = T(x)^{-1}g(x)$ が求まる。

一般には、行列 $T(x)$ が非正則に近い悪条件 (**ill-conditioned**) になる場合、逆行列の計算結果に含まれる誤差が大きくなる。しかし、チェビシェフ基底関数とその評価点によって作られる行列は直交条件を満たし、正則になることが知られている¹⁹。そのため、多重共線性はなく、チェビ

¹⁹ Miranda and Fackler (2002) p. 122 を参照。すなわち、 $T(x)$ のそれぞれの列は互いに無相関となり、逆行列に含

シェフ多項式と評価点を用いて推定されたパラメータは通常の多項式に比べてより安定した値になる。また、行列 $T(x)$ は、ひとたび基底関数と評価点を選ぶとその後は固定される。多項式近似を TI(あるいは VFI) に適用してモデルを解くときは、政策関数を近似するために繰り返し $T(x)$ の逆行列を使用するが、逆行列の計算には時間がかかるため、それをアルゴリズムの初期にあらかじめ行っておくことで、計算時間を節約できる²⁰。

チェビシェフ多項式では、 x_j は $[-1, 1]$ の間の値をとる。より一般的な関数では、 k_j の値は $[k_1, k_N]$ の間にあるとすると、そのような関数をチェビシェフ多項式で近似したいときは、まずそれぞれのグリッドで k_j を $x_j \in [-1, 1]$ に変換する²¹。

$$x_j = \varphi(k_j) = \frac{2(k_j - k_1)}{k_N - k_1} - 1$$

そして以下のような多項式を得る。

$$g(k; \theta) = \theta_0 + \theta_1 T_1(\varphi(k)) + \theta_2 T_2(\varphi(k)) + \cdots + \theta_{N-1} T_{N-1}(\varphi(k))$$

図 4(a) は $N = 11$ のときの、チェビシェフ多項式と評価点を用いた近似である。グリッド数は同じだが評価点の位置を変えている。通常の多項式と等間隔のグリッドを用いた近似に比べて、近似の精度が改善しているのがうかがえる。この場合、改善の理由は、主に評価点を関数の端に多く集めたことによるものである (図 2(b) と比較してみよう)。また、図 4(b) のように、評価点を多くすることで (この場合は $N = 21$)、さらに近似の精度を高めることができる。

まれる誤差は非常に小さい。

²⁰ Judd et al. (2014) はこの方法を推奨している。一般には、連立方程式 $Ax = b$ を x について一度だけ解く場合は、 $x = A^{-1}b$ とする (MATLAB の場合、 $x = \text{inv}(A)*b$) よりも、連立方程式そのものを x について解く ($x = A \setminus b$) 方が速い。しかし、前者では繰り返し計算の際に逆行列 $\text{inv}(A)$ をあらかじめ計算しておくことができる。

²¹ この変換の逆は以下で与えられる。 $k_j = \varphi^{-1}(x_j) = k_1 + 0.5(1 + x_j)(k_N - k_1)$

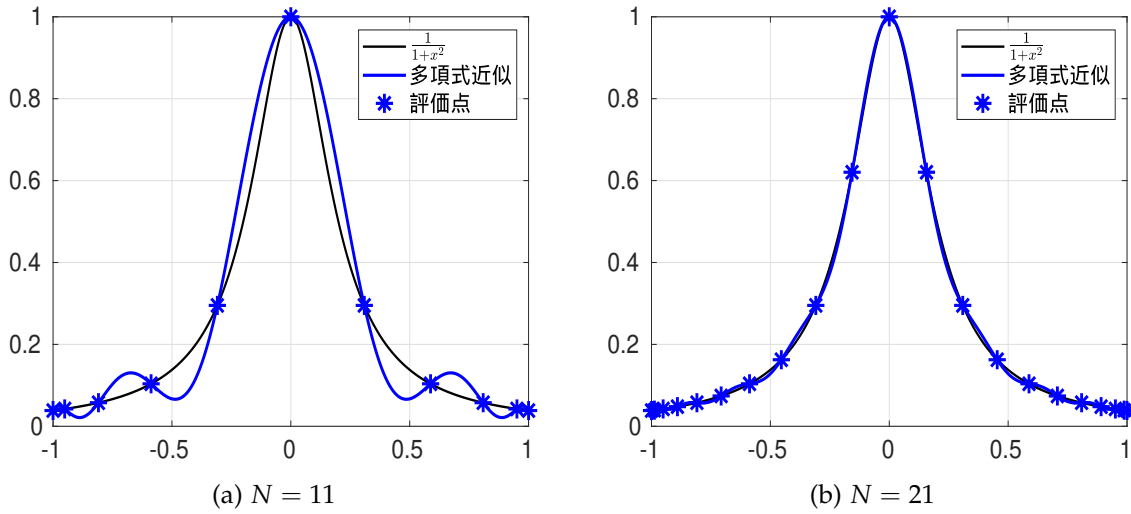


図 4: チェビシェフ多項式と評価点による近似

MATLAB Code

- `how_to_interpolate_cheb.m`, `polygrid.m`, `polybas.m`

3.2 時間反復法への多項式近似の応用

チェビシェフ多項式による関数の近似を、時間反復法に適用してみよう。数値計算のアルゴリズム自体は 2.3 節で説明したものと同一である。実際のコードでもこれまでと異なるのはステップ 3 の近似のところだけである。ここでは、グリッドの数をスプライン補間の場合より少ない $N = 3, 5, 9$ 個とする。このとき、状態空間全体を近似する多項式の次数はそれぞれ $N - 1 = 2, 4, 8$ 次となる。表 1 は、それぞれのグリッドの個数について、計算誤差の絶対値の 10 を底とする対数の平均値と最大値 ($\log_{10} L_1, \log_{10} L_\infty$) をまとめたものである。2 次や 4 次といった低次の多項式でもアルゴリズムは収束するが、真の関数からはやや乖離がみられ、オイラー方程式の計算誤差も悪化する。8 次の多項式を使った場合、計算誤差はやや改善する。

さらに、 k が非常に低い値をとることがありえない場合などでは、グリッドの取り得る値を狭くして、近似誤差が起こりやすい資本の値が少ない場合を避けることで、低次の多項式でも計算誤差を小さくすることができる²²。例えば、資本 k の定常状態 \bar{k} から上下 20 パーセントをグリッドの範囲とした場合 (すなわち、 $k \in [0.8\bar{k}, 1.2\bar{k}]$)。これは、対数線形近似 (**log-linear approximation**)

²² 異質な家計のモデル (Aiyagari (1994) など) では、保有する資本が非常に少なく借入れ制約に直面している家計の存在が重要になるため、このテクニックは適用できない。

で通常カバーする範囲よりも十分に広い)、2 次の多項式を使っても、計算誤差は $10^{-3} \sim 10^{-4}$ 程度に収まる。

また、グリッドの数が少ない方が計算時間は少なくなり、表 1 で計算したグリッドの数や範囲の限りにおいては、収束までの繰り返し回数がほぼ同じであるため、表 2 のように計算時間はグリッドの数にほぼ比例して増えていく。このため、より大規模なモデルで計算時間を短縮したい場合は、グリッドの数を減らす工夫が必要になる²³。

N	$k \in [0.05, 0.5]$		$k \in [0.8\bar{k}, 1.2\bar{k}]$	
	$\log_{10} L_1$	$\log_{10} L_\infty$	$\log_{10} L_1$	$\log_{10} L_\infty$
3	-1.35	-0.94	-3.50	-3.23
5	-2.25	-1.79	-5.80	-5.49
9	-3.76	-3.24	-7.68	-7.68

表 1: オイラー方程式における計算誤差

N	Time (Rel.)
3	0.07 (1.00)
5	0.10 (1.43)
9	0.15 (2.28)

表 2: 計算時間

注：単位は秒。カッコ内は $N = 3$ のケースの値で割った相対的な計算時間。

MATLAB Code

- `main_ti_cheb.m`, `nti_cheb.m`, `EulerEq_cheb.m`

²³ 今回は状態変数が 1 変数のみのケースを扱ったが、外生的なショックや前期の内生変数をモデルに加えることで、状態変数の数は増えていく。グリッドの数は状態変数の数に対して指数的に増加し、これは次元の呪い (curse of dimensionality) と呼ばれる。

4 進んだトピック

今回紹介した時間反復法は、分権経済のモデルを解くときに便利であり、次回紹介するように、名目金利のゼロ制約などの非線形性を明示的に考慮したニューケインジアン・モデルにおいては特によく使われる²⁴。また、今回は状態変数が1変数のみのケースを扱ったが、外生的なショックや前期の内生変数をモデルに加えることで、状態変数の数は増えていく。このようなケースでも、各グリッドでの最適化の方法を改良したり (Christiano and Fisher (2000))、状態空間をグリッドの数が少なくなるよう工夫したり (Judd et al. (2014)、Maliar and Maliar (2015)) することで、時間反復法を使って非線形モデルをより速く解くことができる²⁵。次回も引き続き、時間反復法を用いた非線形ニューケインジアン・モデルへの応用を中心にみていこう。

参考文献

- Aiyagari, S. Rao (1994) "Uninsured Idiosyncratic Risk and Aggregate Saving," *Quarterly Journal of Economics*, Vol. 109, No. 3, pp. 659-684.
- Christiano, Lawrence J. and Jonas D.M. Fisher (2000) "Algorithms for Solving Dynamic Models with Occasionally Binding Constraints," *Journal of Economic Dynamics and Control*, Vol. 24, pp. 1179-1232.
- Coleman, Wilbur John (1991) "Equilibrium in a Production Economy with an Income Tax," *Econometrica*, Vol. 59, No. 4, pp. 1091-1104, July.
- Cooley, Thomas F. (1995) *Frontiers of Business Cycle Research*: Princeton University Press.
- Gali, Jordi (2008) *Monetary Policy, Inflation and the Business Cycle*: Princeton University Press.
- Hirose, Yasuo and Takeki Sunakawa (2019) "Review of Solution and Estimation Methods for Nonlinear DSGE Models with the Zero Lower Bound," *Japanese Economic Review*, Vol. 70, No. 1, pp. 51-104.
- Judd, Kenneth L., Lilia Maliar, Serguei Maliar, and Rafael Valero (2014) "Smolyak Method for Solving Dynamic Economic Models: Lagrange Interpolation, Anisotropic Grid and Adaptive Domain," *Journal of Economic Dynamics and Control*, Vol. 44, pp. 92-123.

²⁴ 次回でやや詳しくみるように、通常使われる対数線形近似では、モデルの非線形性をみることができない。

²⁵ このため、時間反復法はモデルのパラメータ推定にも有用である。より詳しいサーベイについては、Hirose and Sunakawa (2019) を参照。

- Maliar, Lilia and Serguei Maliar (2015) “Merging Simulation and Projection Approaches to Solve High-Dimensional Problems with an Application to a New Keynesian Model,” *Quantitative Economics*, Vol. 6, pp. 1-47.
- Mas-Colell, Andreu, Michael D. Whinston, and Jerry R. Green (1995) *Microeconomic Theory*: Oxford University Press.
- Miranda, Mario J. and Paul L. Fackler (2002) *Applied Computational Economics and Finance*, Cambridge: MIT Press.
- Richter, Alexander W., Nathaniel A. Throckmorton, and Todd B. Walker (2014) “Accuracy, Speed and Robustness of Policy Function Iteration,” *Computational Economics*, Vol. 44, No. 4, pp. 445-476, December.
- Romer, David H. (2011) *Advanced Macroeconomics*: McGraw-Hill, 4th edition.
- Sargent, Thomas J. and John Stachurski (2019) “Optimal Growth II: Time Iteration,” in *Lectures in Quantitative Economics*, <https://lectures.quantecon.org/> (最終アクセス 2019 年 4 月 14 日).
- Sundaram, Rangarajan K. (1996) *A First Course in Optimization Theory*: Cambridge University Press.
- Woodford, Michael (2003) *Interest and Prices: Foundations of a Theory of Monetary Policy*: Princeton University Press.
- 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 .
- 齊藤誠 (2006) 『新しいマクロ経済学—クラシカルとケインジアン邂逅 [新版]』, 有斐閣 .
- 西村清彦 (1990) 『経済学のための最適化理論入門』, 東京大学出版会 .
- 木村武・藤原一平・黒住卓司 (2005) 「社会の経済厚生と金融政策の目的」, 日銀レビュー 2005-J-9 .