

2021年度応用マクロ経済学講義ノート

(8) 動的計画法と数値解法

阿部修人
一橋大学経済研究所

2021年6月

1 確実性下の最大化問題

前回の講義ノートと同様に、家計は T 期間存在し、 T 期末に死亡すると仮定する。単純化のため、時間選好率および金利は共にゼロと仮定し、その代わりに每期変動する所得 Y_t を得る、と仮定しよう。さらに、所得は変動するが、その値は事前にわかっている、すなわち、不確実性は存在しないと仮定する。効用関数は異時点間で分離可能であり、各期の効用関数を $u(c_t)$ 、 $u' > 0$ 、 $u'' < 0$ とする。初期における資産額を A_0 とし、家計は自由に借入れ、貯金が可能であると仮定する。すると、この家計の効用関数と予算制約は下記のように書くことができる。

$$U = \sum_{t=1}^T u(c_t), u' > 0, u'' < 0, \quad (1)$$

$$\sum_{t=1}^T c_t \leq A_0 + \sum_{t=1}^T Y_t. \quad (2)$$

この動学最適化問題の解をラグランジュアンを用いて解くと、ラグランジュアンは

$$L = \sum_{t=1}^T u(c_t) + \lambda \left(A_0 + \sum_{t=1}^T Y_t - \sum_{t=1}^T c_t \right), \quad (3)$$

であり、一階条件は

$$u'(c_t) = \lambda. \quad (4)$$

すなわち、すべての期において、消費の限界効用は一定になる。限界効用は単調減少関数であるから、消費水準も一定となり、

$$c_t = \bar{c} \text{ for all } t. \quad (5)$$

予算制約に代入すると

$$c_t = \frac{1}{T} \left(A_0 + \sum_{t=1}^T Y_t \right). \quad (6)$$

すなわち、生涯所得と初期資産を各期に均等に配分することになる。上式の右辺は恒常所得と呼ばれるものであり、生涯所得の平均値と、初期時点で保有している資産を生存期間に均等に分割したものの和である。ここで重要な性質は、「家計消費は、所得の当期の実現値には依存しないこと」である。たとえば、若年期にほとんど所得がなく、後になってようやく所得が得られる場合、若年期には借金をし、将来の所得で借金を返済することになる。

1.1 不確実性下の最大化問題

次に、もう少し一般化させ、正の割引率を導入し、家計は無限に存在し、所得に不確実性が存在し、 Y_t が確率変数であると仮定しよう。家計が期待効用を最大化させるとし、 β を $0 < \beta < 1$ を満たす時間割引とすると、家計の効用最大化問題は下記のようになる。

$$\max E[U] = E_0 \left[\sum_{t=1}^T \beta^t u(c_t) \right]. \quad (7)$$

家計は、各期に下記のような予算制約に直面しているとする。ただし、金利は r で一定であるとする。

$$\text{s.t. } A_{t+1} \leq (1+r)A_t + Y_t - c_t. \quad (8)$$

なお、資産の値 A_t は正の値を取る場合は貯蓄、負の値を取る場合は負債となる。もしも、最終期の t 期末において、負債を抱えることを許容する、すなわち借金を残したまま死ぬことを許容すると、無限に借金をして大量の消費を行い、負債の返済を行わずに死ぬことが最適となってしまう。そのような行動を除外するため、追加として、No Ponzi Game Condition と呼ばれる下記の条件を課す。

$$A_T \geq 0. \quad (9)$$

この家計の効用最大化問題を解く際には、ラグランジュアンを下記のように設定する。

$$L = E_0 \left[\sum_{t=1}^T \beta^t u(c_t) + \sum_{t=1}^T \beta^t \lambda_t ((1+r)A_t + Y_t - c_t - A_{t+1}) \right] \quad (10)$$

λ_t はラグランジュ乗数であり、各期の所得の実現値により変化する確率変数となっている。この最適化問題の一階条件は、消費に関して微分すると、

$$u'(c_t) = \lambda_t, \quad (11)$$

を得、また資産に関して微分すると、

$$E_t \beta (1+r) [\lambda_{t+1}] = \lambda_t, \quad (12)$$

すなわち、

$$E_t \beta (1+r) [u'(c_{t+1})] = u'(c_t), \quad (13)$$

となる。(13)は消費に関するオイラー方程式と呼ばれるものであり、家計消費分析において最も重要な方程式の一つである。この意味は、不確実性を無視し、下記のような \neq 方程式を考えると明らかである。

$$\frac{u'(c_t)}{\beta u'(c_{t+1})} = (1+r). \quad (14)$$

今期の消費一単位を諦めることによる効用の減少分は $u'(c_t)$ である。この一単位の消費を諦めることにより、来期において、一単位の消費に金利が加わり、 $1+r$ 単位の消費が可能になる。すなわち、 $1+r$ は今期と来期の消費の価格比となる。効用が最大化されるのは、二財の限界代替率が価格比に等しい時であり、来期の一単位の消費が増加する時の限界効用は $\beta u'(c_{t+1})$ であるから、上式は、今期と来期の限界代替率が価格比に等しいことを示している。

各時点の効用関数が CRRA であるとき、オイラー方程式は下記のようになる。

$$E_t \left[\beta \left(\frac{c_t}{c_{t+1}} \right)^\sigma \right] = \frac{1}{(1+r)}. \quad (15)$$

不確実性を無視して、上の式を対数線形近似して整理すると、

$$\begin{aligned} \ln c_{t+1} - \ln c_t &= \frac{1}{\sigma} [\ln(1+r) + \ln \beta], \\ &\simeq \frac{1}{\sigma} (r - \rho). \end{aligned} \quad (16)$$

ただし、 $\beta = 1/(1+\rho)$ である。ここで、 $T = \infty$ とし、無限視野の問題を考えてみよう。 $r = \rho$ のとき、 $\ln c_{t+1} = \ln c_t$ となり、消費は常に一定値をとり、定常状態となる。 $r < (<) \rho$ のとき、消費水準は時間と共に増加(減少していく。このときの変化速度は、 σ の逆数に比例する。既にみたように、 σ は異時点間の代替の弾力性の逆数であり、 σ が大きいほど、異時点間の代替の弾力性は小さくなっている。異時点間の代替の弾力性が小さい時、消費は金利にあまり反応せず、ほぼ一定の値を取り続けることになるが、これは、消費の平滑化 (Consumption Smoothing) と呼ばれ、消費の動学分析のみならず、現代のマクロ動学モデルにおける根幹の動学メカニズムの一つである。 σ が大きいほど、消費の平滑化の強度は大きくなる。逆に、 σ を小さくすると、わずかの金利の変化に対し大幅に消費水準が変化することになる。 σ をゼロに近づけると異時点間の代替の弾力性は無限大となり、消費平滑化が消滅する。このときの各期の効用関数は線形となり、限界効用が逓減しない関数形となっている。

1.2 動的計画法 (Dynamic Programming)

1.2.1 導入

家計消費分析で登場する下記の動学最適化問題、

$$\max E_0 \left[\sum_{t=1}^T \beta^t u(c_t) \right], \quad (17)$$

$$\text{s.t. } A_{t+1} \leq (1+r)A_t + Y_t - c_t \quad \text{for all } t. \quad (18)$$

の解法は、ラグランジュアンを使わずに、動的計画法 (Dynamic Programming, DP) を用いて解くことも可能である。今、所得 Y_t が下記のような一階の自己回帰モデル (AR1) に従っていると仮定しよう¹。

$$Y_{t+1} = \rho Y_t + \varepsilon_{t+1}, \quad 0 < \rho < 1. \quad (19)$$

ただし、 ε_{t+1} は系列相関のない確率変数であり、 t 期における期待値はゼロであり、正規分布に従うと仮定する。すると、この家計は、 T 期首に、前期までの蓄積の結果である資産 $(1+r)A_t$ と今期の所得 Y_t の実現値を観察してから今期の消費額 c_t を決めることになる。今、この家計の最終期における意思決定を考えてみよう。最終期である T 期では、この家計は来期の $T+1$ 期には存在しないので、来期のために資産を残す必要がない。したがって、この家計の最適な消費量は、その時点で保有する全資産と所得を消費に振り分けることになる。すなわち、

$$c_T = (1+r)A_T + Y_T. \quad (20)$$

次に、その1期前、 $T-1$ 期における家計が直面する問題を考えよう。 $T-1$ 期において、家計は $(1+r)A_{T-1}$ の資産と Y_{T-1} の所得を得ている。そこで、 c_{T-1} の消費を行うと、来期には、

$$A_T = ((1+r)A_{T-1} + Y_{T-1} - c_{T-1})(1+r), \quad (21)$$

の資産が残る。家計は来期と今期の効用の現在割引価値の和の期待値を最大化するから、

$$\max_{c_{T-1}} [u(c_{T-1}) + E_{T-1} \beta u(((1+r)A_{T-1} + Y_{T-1} - c_{T-1})(1+r) + Y_T)], \quad (22)$$

を最大化することになる。上の最大化問題は、金利が外生で一定であることを考慮すると、 A_{T-1} と Y_{T-1} を所与として、最適な消費水準 c_{T-1} を選んで

¹AR1 に従うという仮定は下記のアルゴリズムの説明を簡単にするために設けたものであり、より一般的な確率過程を仮定することも可能である。

いることになり、最適な消費水準、および最大化された期待効用の割引現在価値の和もまた A_{T-1} と Y_{T-1} に依存する関数となる²。

上記の最大化問題の解

$$c_{T-1} = c_{T-1}^*(A_{T-1}, Y_{T-1}), \quad (23)$$

は、政策関数 (Policy Function)、政策関数の要素になっている変数を状態変数 (State Variable) と呼ぶ。また、最大化された目的関数

$$\begin{aligned} V_{T-1}(A_{T-1}, Y_{T-1}) = & u(c_{T-1}^*(A_{T-1}, Y_{T-1})) \\ & + E_{T-1}\beta u(((1+r)A_{T-1} + Y_{T-1} - c_{T-1}^*(A_{T-1}, Y_{T-1}))(1+r) + Y_T), \end{aligned} \quad (24)$$

は価値関数 (Value Function) と呼ばれる。(22) を価値関数を用いて書き直すと

$$V_{T-1}(A_{T-1}, Y_{T-1}) = \max_{c_{T-1}} [u(c_{T-1}) + E_{T-1}\beta V_T(A_T, Y_T)], \quad (25)$$

$$s.t. \quad Y_T = \rho Y_{T-1} + \varepsilon_T, \quad (26)$$

$$A_T \leq (1+r)A_{T-1} + Y_{T-1} - c_{T-1}. \quad (27)$$

となる。ちなみに、最終期の価値関数は

$$V_T(A_T, Y_T) = u((1+r)A_T + Y_T), \quad (28)$$

であり、利用可能な資源を全てそのまま消費した時に得られる効用水準に等しい。(25) はベルマン方程式 (Bellman Equation) と呼ばれるものであり、各期の最大化問題は、来期の価値関数と今期の効用の和を最大化させるという、二期間の最大化問題として定式化されている。今は、 $T-1$ 期の問題を考えたが、 $T-1$ 期における価値関数が入手できたら、 $T-2$ 期の問題も同様に定義可能であり、一般に

$$V_t(A_t, Y_t) = \max_{c_t} [u(c_t) + E_t\beta V_{t+1}(A_{t+1}, Y_{t+1})], \quad (29)$$

$$s.t. \quad Y_{t+1} = \rho Y_t + \varepsilon_{t+1}, \quad (30)$$

$$A_{t+1} \leq (1+r)A_t + Y_t - c_t. \quad (31)$$

と書くことができる。このように、最終期における最適解を基に、時間を遡って最適消費経路を求める解法は「後ろ向きの帰納法 (Backward Induction)」と呼ばれ、全期間の最適消費経路を一括して導出するラグランジュアンによる解法と対照的なものである。

²数値的に (22) の最大化問題を解く際には、期待値オペレーターがあるため、積分する必要が生じる。また、 A_{T-1} と Y_{T-1} を所与として、最適な消費水準 c_{T-1} に関する非線形最大化問題を数値的に解く、ということは、 A_{T-1} と Y_{T-1} の可能なコンビネーションに対応する最適な c_{T-1} の対応を全て求めることを意味する。

ラグランジュアンによる解と動的計画法による解は必ずしも一致するとは限らない。また、ベルマン方程式を満たす価値関数が複数存在する可能性もある。両者が一致するための条件は Stokey and Lucas (1988) に詳しく議論されているが、家計消費分析において、両者の乖離が表面化するケースは少ない³。実際にラグランジュアンで解く場合とベルマン方程式を用いる際の最大の違いは、前者は各期の消費の経路を求めるのに対し、後者は各期における政策(消費)関数を求めることにある。コンピューターを用いて、数値的に解を求める場合、動的計画法は(1) 後ろ向きの帰納法 (Backward Induction) か(2) 価値関数反復法 (Value Function Iteration) のどちらかの手法で解くことになる。

1.2.2 後ろ向きの帰納法 (Backward Induction)

後ろ向きの帰納法 (Backward Induction) は、動学最適化問題の最終期における行動が確定している時に主として利用される手法である。家計消費分析では、家計の最後の時点、すなわち経済から撤退する時点が確定している時によく用いられる。この撤退時点は、家計がこれ以上経済に残ることはありえない時点のことであり、例えば人間の寿命が最大で 120 歳であるならば、120 歳が最終時点となる。120 歳よりも前の時点で死亡、あるいは経済からの撤退(消費・貯蓄に関する意思決定をしなくなる)は容易にモデルに組み込むことが可能である。

既に、Backward Induction はベルマン方程式を導出するときに用いているので、ここではより具体的に、Backward Induction を用いた解法を紹介する。

家計の最終期を T 期とし、 $T+1$ 期に資産を残さないということから、(22) が導出された。 $T-1$ 期におけるベルマン方程式は

$$V_{T-1}(A_{T-1}, Y_{T-1}) = \text{Max}_{c_{T-1}} [u(c_{T-1}) + E_{T-1} \beta u(((1+r)A_{T-1} + Y_{T-1} - c_{T-1})(1+r) + Y_T)], \quad (32)$$

であった。上記の解である c_{T-1} は、 A_{T-1} と Y_{T-1} を要素とする関数、 $c_{T-1} = c_{T-1}^*(A_{T-1}, Y_{T-1})$ となる。最適な消費水準 c_{T-1} に関する非線形最大化問題を数値的に解く、ということは、 A_{T-1} と Y_{T-1} の可能なコンビネーションに対応する最適な c_{T-1} の対応を全て求めることを意味する。ここでは、一番単純なグリッド分割による離散近似法を紹介する⁴。

³動的計画法とラグランジュアンによる解が乖離するケースの一つに、時間的整合性 (Time Consistency) がある。ラグランジュアンの解が時間的整合性をみたさないとき、その解は動的計画法の解とはならない。時間的整合性は、様々なケースで発生するが、双曲割引 (Hyperbolic Discount) によるものに関しては Harris and Laibson (2001) を参照せよ。

⁴動学的最適化問題の解法に限らず、経済学における数値解析一般に関しては、Judd (1998) は既に古典としての地位を確立している。Miranda and Fackler (2002) は動的計画法に詳しく、また Matlab のコードを多く掲載しており、より実践的であるが、コードがブラックボックスのままでも計算できてしまうため、数値計算を習得するためには、多少面倒でも Judd (1998) を読むことをお勧めする。

まず、所得過程を特定化する必要がある。以下の例では、単純な所得過程を考える。

(1) 引退前

引退前に t 歳の個人が受け取る所得は定額部分 1 と一定の率 (7%) で成長する通常所得 Y_t 、および変動所得 ω_t から構成されていると仮定する。変動所得は、幸運な時と不運な時の二段階の値をとり、幸運な時は通常所得の 3 割増、不運な時は通常所得の 3 割減の所得を受け取ると仮定する。すなわち、

$$Y_t = 1 + 1.07^{(t-1)} \quad (33)$$

$$Income = Y_t \times \omega_t \quad (34)$$

$$\omega_t = 0.7 \text{ (低い所得のとき)}, \quad (35)$$

$$\omega_t = 1.3 \text{ (高い所得のとき)}, \quad (36)$$

とする。

変動所得 ω_t の状態は、前期の状態に依存しており、前期の幸運なとき、今期も幸運である確率が 0.9、同様に前期不運な時に今期も不運である確率も 0.9 であるとする。今期の状態の確率が前期の状態のみに依存するものはマルチンゲールと呼ばれる。このモデルのように、二つの値しかとまらないものは、前期と今期の状態は下記の推移確率行列で表わすことができる。

$$\begin{array}{cc} & \begin{array}{cc} \text{Low at } t+1 & \text{High at } t+1 \end{array} \\ \begin{array}{c} \text{Low at } t \\ \text{High at } t \end{array} & \begin{pmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{pmatrix} \end{array} \quad (37)$$

第 (1,1) 成分は、 t 期に低い所得で、 $t+1$ 期においてもまた低い所得である確率が 0.9 であることを表わす。このような推移確率行列で状態の推移を記述できる確率過程はマルコフ・チェーンと呼ばれる⁵。

(2) 引退後

引退後の所得は変動も成長もせず、一定の所得をうけとる、すなわち、所得は每期同額で、

$$Income = 0.7 \times Y_{retirement} \quad (38)$$

となる。この家計は 80 年生存し、そのうち最初の 20 年は特に消費活動をせず、21 歳から労働と消費を行い、60 歳で引退し、残りの 20 年を引退期として過ごす⁶と仮定する。

効用関数は CRRA であり、リスク回避度は $\sigma = 3$ 、時間選好 $\beta = 0.97$ 、金利は $R = 1/\beta$ 、初期資産はゼロとする⁶。

⁵マルコフ・チェーンは確率過程論の基本であり、多くの確率論の教科書に解説がある。古くなったが、渡部 (1979) はわかりやすく、また包括的にマルコフ・チェーンの説明を行っている。

⁶負債保有を認めることも勿論可能である。また、Appendix に掲載しているプログラムでは、初期資産が家計により異なり、正規分布に従っている状態を考えている。

この動学問題では、状態変数は資産 A と所得ショック ω の二種類であるが、所得ショック ω の実現値は二種類しかないので、引退前には、High と Low の二種類の状態に関し、それぞれに価値関数を考え、

$$V_t(A_t, High) = \max \left[\frac{c_t^{1-\sigma}}{1-\sigma} + \beta (0.9 * V_{t+1}(A_{t+1}, High) + 0.1 * V_{t+1}(A_{t+1}, Low)) \right], \quad (39)$$

$$V_t(A_t, Low) = \max \left[\frac{c_t^{1-\sigma}}{1-\sigma} + \beta (0.1 * V_{t+1}(A_{t+1}, High) + 0.9 * V_{t+1}(A_{t+1}, Low)) \right], \quad (40)$$

とベルマン方程式を二つに分割する。

状態変数を離散近似する場合は、資産の上限と下限を事前に設定する必要がある。借り入れを認めない場合の資産額の下限はゼロとできるが、上限をどの値に設定するのが適切であるかは、パラメーターや関数形に依存する。したがって、何度か上限を変えてプログラムを走らせ、結果をみながら上限を設定する必要がある。大きな上限値を設定し、細かくグリッドを取ると数値計算に必要な時間とメモリー量が飛躍的に増加するので、動かすコンピューターの性能を考えながらグリッドをとる必要がある。

以下の例では、資産の下限をゼロ、上限を 100 とし、0.1 単位で分割している。したがって、資産は

$$A_T = [0, 0.1, 0.2, \dots, 99.9, 100], \quad (41)$$

と 1001 個のグリッドに分割されている⁷。

まず、最終期における価値関数、

$$V_T(A_T), \quad (42)$$

を計算する。最終期には、家計は t 期首で利用可能な資産総額を全て消費に回すので、最終期の価値関数の値は、 $(RA_T + Y_T)$ と同額の消費から得られる効用と等しくなる。従って、資産グリッド i に関して、価値関数の値を求めると、下記ようになる。

$$V_T(A_T(i)) = \frac{(RA_T(i) + Y_T)^{1-\sigma}}{1-\sigma}. \quad (43)$$

次に、得られた最終期の価値関数を用い、ベルマン方程式の最大値を、 A_{T-1} の各グリッドに関して求める。得られた V_{T-1} を用いて、再度ベルマン方程式の最大値を求めれば、 V_{T-2} を求めることが可能である。繰り返しの作業を行うことで、全期間の価値関数を求めることが可能である。図 1 は、人生最終期 ($t = 80$) とその一期前 ($t = 79$) および二期前 ($t = 78$) の価値関数の形

⁷資産の最小値をゼロとすることで、流動性制約を導入していることになる。資産として負の値を許容することも勿論可能であり、その場合、借り入れを行うことが可能になる。

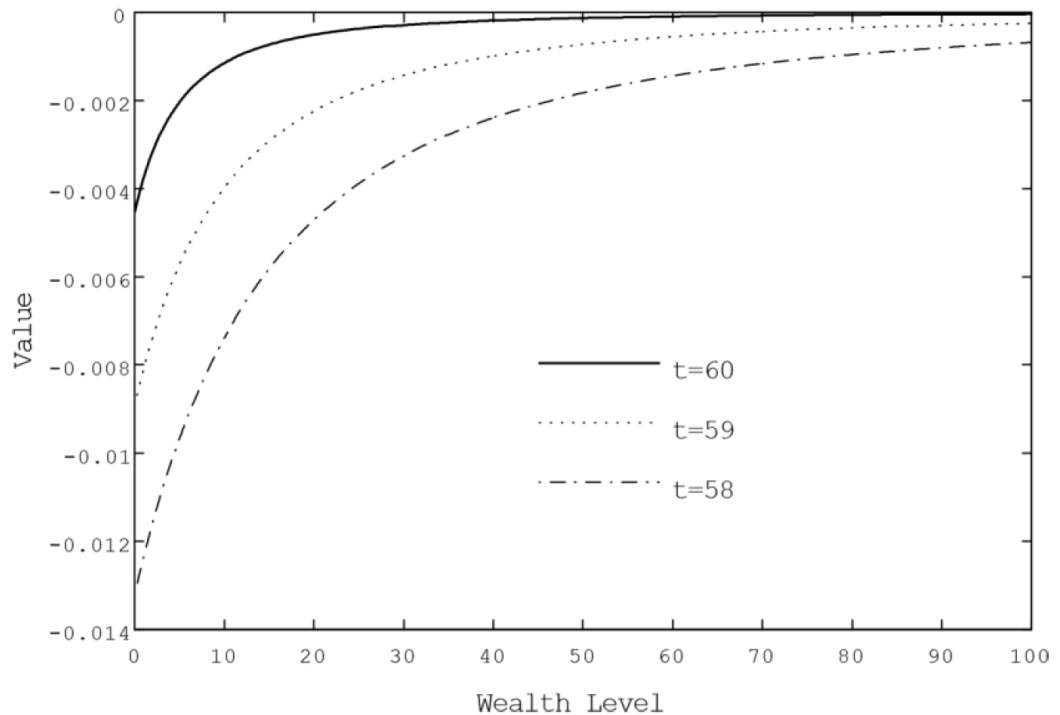
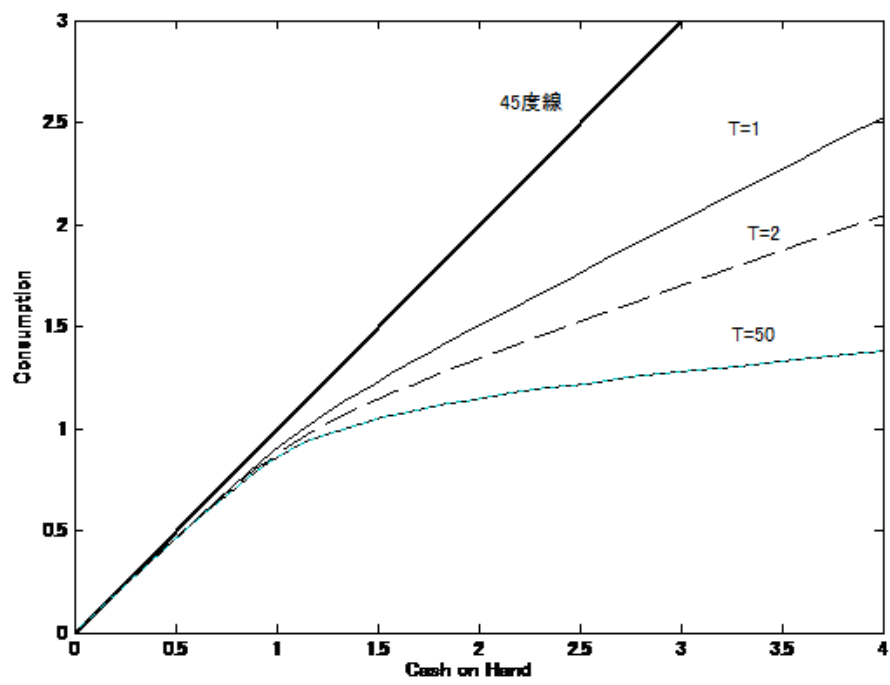


図 1: 価値関数

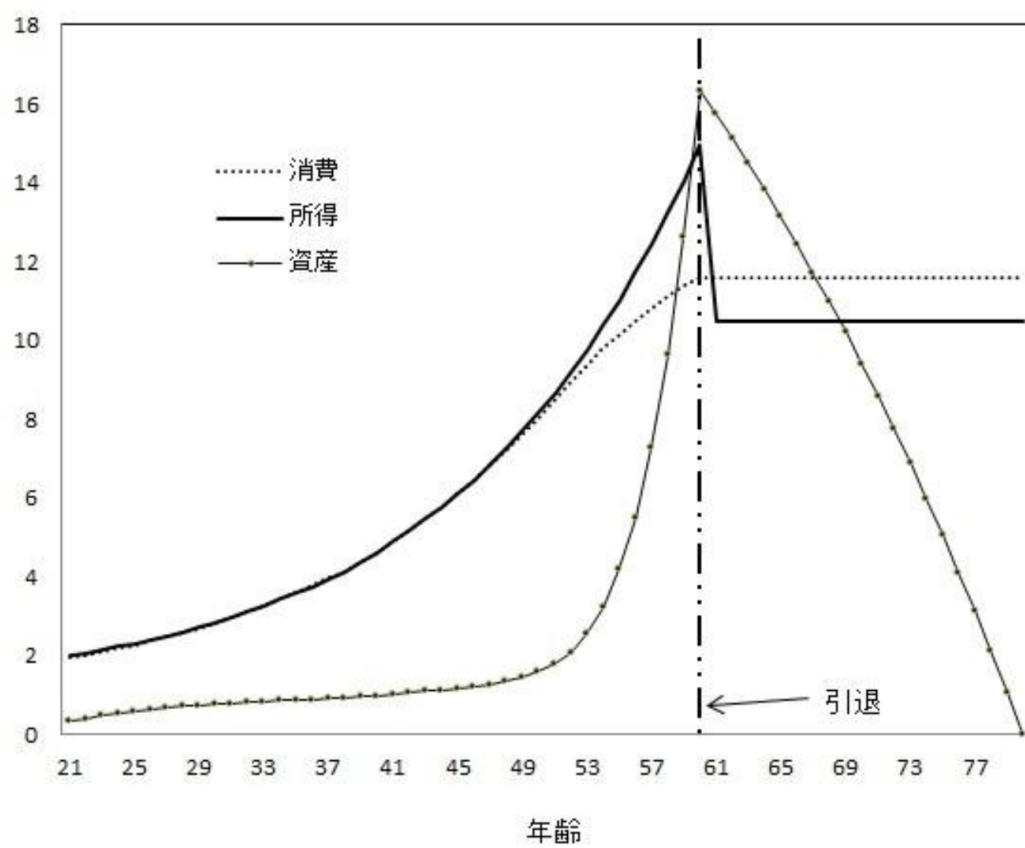
状を描いたものである。価値関数は資産水準に関して増加関数、かつ凹関数になっており、若いほど、低い水準になり、かつ凹性は緩やかになっていく。図 2 は同様に Policy Function を描いたものである。最終期 ($t = 80$) では、保有資産 (正確には資産に金利を乗じ、さらに所得を加えたもの) 全てを消費するが、その一期前では、水準はほぼ半分となっている。これは、残り二期間に消費を分割させている (平滑化させている) ことを示している。図 3 は、所得プロセスをコンピュータ上でシミュレートし、仮想的なライフサイクル行動を描いたものである。このモデルでは、時間選好率と金利が同じ水準にあるため、もしも不確実性がなく、自由に借入れが可能であれば、最適消費経路は一定値をとるはずであるが、(1) 所得に不確実性があり、かつ (2) 負債保有ができない、という二つの理由から、消費は一定ではなくなっている。50 歳までは所得と消費はほぼ一致し、資産蓄積もあまり進まないが、50 歳を過ぎた時点で一気に資産蓄積が進み、引退と同時に資産の取り崩しが始まる。不確実性がなくなり、資産蓄積が進んだ引退期以降の消費は一定となる。

Appendix にある Matlab コードと上記の説明を対応させて、ぜひとも一度はコードを実行し、様々な設定の下で、各年齢の価値関数や Policy Function の形状を自分で描いて欲しい。



☒ 2: Policy Function

年齢プロフィール



1.2.3 価値関数反復法 (Value Function Iteration)

今までの例では、家計のライフサイクルを描写するため、人生の最終期を設定した。この結果、年齢により価値関数の形状が変化し、最適消費計画も年齢により異なるものになった。一方、マクロモデルの場合は、明示的な最終期を設定せず、無限視野を想定することが一般的である。このとき、最終期の存在を前提とした Backward Induction が使えなくなる。その代わり、無限視野の問題では、経済の状況 (分布) が時間に依存しない、定常状態が発生する。定常状態において、価値関数の形状は時間により変化しなくなる。この時、Value Function Iteration と呼ばれるアルゴリズムにより、価値関数の形状を数値的に求めることが可能になる。

単純な無限視野の最適化問題を考える。各期の効用関数を u 、資源制約を g 、状態変数を x 、コントロール変数を c で表し、動的最適化問題を以下のように定義する。

$$\max_{\{c_t\}_{t=0}^{\infty}} \sum_{t=0}^{\infty} \beta^t u(x_t, c_t), \quad (44)$$

$$x_{t+1} = g(x_t, c_t), \quad (45)$$

$$x_0 : \text{given}. \quad (46)$$

上記の問題を解いて得た最適化された生涯効用を V^* で表わすと、

$$V^* = \max_{\{c_t\}_{t=0}^{\infty}} \sum_{t=0}^{\infty} \beta^t u(x_t, c_t) \quad \text{s.t. } x_{t+1} = g(x_t, c_t) \quad \text{and } x_0. \quad (47)$$

そこで、ベルマン方程式は以下のように定義される。

$$V(x) = \max_c (u(x, c) + \beta V(x')) \quad \text{s.t. } x' = g(x, c). \quad (48)$$

ある一定の条件下で、Value Function Iteration とよばれる下記の操作により、ベルマン方程式を満たす価値関数 $V(x_t)$ を得ることが可能であることが知られている。

$$V_{j+1} = \max_c (u(x, c) + \beta V_j(x')) \quad \text{s.t. } x' = g(x, c). \quad j = 1, 2, \dots, \quad (49)$$

まず、適当に初期の価値関数の候補 V_0 を作成する。通常は全ての状態変数のグリッドに関して常にゼロの値となる単純な関数を仮定することが多い。その価値関数の候補を用いて、ベルマン方程式を最大化する第二の価値関数の候補 V_1 を作成する。この作業を繰り返していくと、価値関数は一定の関数形に収束していくのである。この便利な性質は、ベルマン方程式が縮小写像になっていることに起因している。数学的な背景に関しては Stokoy and Lucas (1988)、実践的なアルゴリズムに関しては Ljungqvist and Sargent (2004) および Miranda and Fackler (2002) において詳しく説明されている。

2 Code

```
%% Simple Life Cycle Consumption Model with Uncertainty
%
% 有限視野モデルを Backward Induction で解く
% 各期の効用関数は CRRA
%  $W(t+1)=R*W(t) + Y(t) -C(t)$ 
% 所得ショックには持続性が存在
% 2010年10月
% 阿部修人
clear all
tic
%% Set Parameter Values
gamma=3; % リスク回避度
beta=0.97; % 割引因子
R=1.01; % 金利 (1/beta である必要はない)
maxage=60; % 死亡年齢
retire_age=40; % 引退年齢 (死亡年齢よりも小さい必要あり)
%% 資産グリッドの作成
max_w=100; % 資産の最大値
min_w=0; % 資産の最小値 (不でも可)
inc_w=0.1; % グリッド幅
n= round((max_w-min_w)/inc_w+1); % # グリッド数
% 資産グリッド
for i=1:n
w(i,1)=min_w+inc_w*(i-1);
end
%
%% 所得過程 %%%%%%%%%%%
shockvalue=zeros(1,2); % 所得ショック
shockvalue(1)=0.7; % 低所得実現値
shockvalue(2)=1.3; % 高所得実現値
%
% 推移確率
trprob=[ 0.9 0.1; 0.1 0.9 ];
% 所得過程の非確率部分
Earning=ones(maxage,1); % 所得のフロー
g_young=1.07; % 勤労期における所得成長率
for k=1:maxage
if k<=retire_age
```

```

Earning(k)=1+1*g_young^(k-1);
elseif k>retire_age
Earning(k)=0.7*Earning(retire_age);
end
end
%
%% 初期化 %%%%%%%%%%%
VL_age=zeros(n,maxage);% 年齢プロファイル
VH_age=zeros(n,maxage);
Util=-1000000000000000000*ones(n,n);% 負の消費選択を避ける
Cons=zeros(n,n);
policy_l_age=zeros(n,maxage);% 年齢・Policy プロファイル
policy_h_age=zeros(n,maxage);
policy_l=zeros(n,1);
policy_h=zeros(n,1);
%% 終点作成 %%%%%%%%%%%
%
for i=1:n
for j=1:n
Cons(i,j)=R*w(i)+Earning(maxage);
if Cons(i,j)>0.0
Util(i,j)=(Cons(i,j)^(1-gamma))/(1-gamma);
end
end
end
%
% 終点における価値関数の作成
for i=1:n
VL1(1,i)=Util(i,1);
VH1(1,i)=Util(i,1);
policy_l_age(i,maxage)=R*w(i)+Earning(maxage);
policy_h_age(i,maxage)=R*w(i)+Earning(maxage);
end
%
VL_age(:,maxage)=VL1';
VH_age(:,maxage)=VH1';
%
%% Main Loop: Backward Induction %%%%%%%%%%%
for k=2:maxage

```

```

% k=1 における Policy は既に計算済み
Cl=zeros(n,n);
Ch=zeros(n,n);
Ul = -10000000000000000000*ones(n,n);
Uh = -10000000000000000000*ones(n,n);
for i=1:n
for j=1:n
invretire=maxage-retire_age;
invage=maxage-k+1; % 並べ替え用
if k<=invretire % 引退後
% 引退後は所得は一定であることに注意
Cons(i,j)=R*w(i)+Earning(invage)-w(j);
if Cons(i,j)>0.0
Ul(i,j)=(Cons(i,j)^(1-gamma))/(1-gamma);
Uh(i,j)=Ul(i,j);%
end
end
if k>invretire % 勤労期
Cl(i,j)=R*w(i)+Earning(invage)*shockvalue(1)-w(j);
Ch(i,j)=R*w(i)+Earning(invage)*shockvalue(2)-w(j);
if Cl(i,j)>0
Ul(i,j)=(Cl(i,j)^(1-gamma))/(1-gamma); %
end
if Ch(i,j)>0
Uh(i,j)=(Ch(i,j)^(1-gamma))/(1-gamma); %
end
end
end
end
% ベルマン方程式の最大値を計算
for s=1:n
[VL0(s,1) pl(s)]=max(Ul(s,:)+beta*(trprob(1,1)*VL1+trprob(1,2)*VH1));
[VH0(s,1) ph(s)]=max(Uh(s,:)+beta*(trprob(2,1)*VL1+trprob(2,2)*VH1));
end
VL1=VL0';
VH1=VH0';
VL_age(:,invage)=VL0;
VH_age(:,invage)=VH0;
% Policy functions の導出

```

```

if invage<=retire_age % 勤労期
policy_l(:)=R*w(:)+Earning(invage)*shockvalue(1)- w(pl(:));
policy_h(:)=R*w(:)+Earning(invage)*shockvalue(2)- w(ph(:));
elseif invage>retire_age % 引退期
policy_h(:)=R*w(:)+Earning(invage)- w(ph(:));
policy_l(:)=R*w(:)+Earning(invage)- w(pl(:));
end
end
policy_l_age(:,invage)= policy_l;
policy_h_age(:,invage)= policy_h;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Life Cycle Simulation
%
simnum=5000;% 家計数
sigma=0.5;% 初期資産分布の標準偏差
mhu =0.1;% 初期資産分布の平均値
cons_sim=zeros(simnum, maxage);% シミュレートされた消費
next_wealth=zeros(simnum, maxage);% 同資産
incomedist=zeros(simnum, maxage);% 同所得
wealth=zeros(1,simnum);
wealth2_1=zeros(1,simnum);
%
% 初期資産分布の作成
for i=1:simnum
wealth(i)=round((((randn(1)+mhu)*sigma)-min_w)/inc_w+1);
% Note that the grid number: n=1+(wealth-min_w)/inc_w
if wealth(i)<1
wealth(i)=1;
elseif wealth(i)>n
wealth(i)=n;
end
end
end
%
wealth2=sort(wealth);
%
% 初期所得分布をマルコフチェーンの定常分布と仮定
%
lengthpro=length(trprob);
P=(1/lengthpro)*ones(1,lengthpro);

```



```

for i=1:100
P=P*trprob;
end
incomestate=mnrnd(1,P,simnum);% 所得分布の初期状態
%
% 各最適経路の導出
for i=1:simnum
for k=1:maxage
asset=wealth2(i);
if k<=retire_age
income_now=incomestate(i,:);
nextshock=mnrnd(1, trprob,2);% 所得状態の推移
if income_now(1,1)==1;
cons_sim(i,k)=policy_l_age(asset,k);
next_wealth(i,k)=R*w(asset)+Earning(k)*shockvalue(1)-cons_sim(i,k);
incomedist(i,k)=Earning(k)*shockvalue(1);
incomestate(i,:)=nextshock(1,:);
elseif income_now(1,2)==1
cons_sim(i,k)=policy_h_age(asset,k);
next_wealth(i,k)=R*w(asset)+Earning(k)*shockvalue(2)-cons_sim(i,k);
incomedist(i,k)=Earning(k)*shockvalue(2);
incomestate(i,:)=nextshock(2,:);
end
elseif k>retire_age
cons_sim(i,k)=policy_h_age(asset,k);
next_wealth(i,k)=R*w(asset)+Earning(k)-cons_sim(i,k);
incomedist(i,k)=Earning(k);
end
wealth2_1(i)=round(((next_wealth(i,k))-min_w)/inc_w)+1;
if wealth2_1(i)<1
wealth2_1(i)=1;
elseif wealth2_1(i)>n
wealth2_1(i)=n;
end
wealth2(i)=wealth2_1(i);
end
end
% 年齢プロファイル
teststat=zeros(maxage,6);

```

```

for k=1:maxage
teststat(k,1)=var(log(cons_sim(:,k)));
teststat(k,2)=mean(cons_sim(:,k));
teststat(k,3)=var(log(incomedist(:,k)));
teststat(k,4)=mean(incomedist(:,k));
teststat(k,5)=var(next_wealth(:,k));
teststat(k,6)=mean(next_wealth(:,k));
end
%
%% Plot
%
axe=zeros(maxage,1);
for i=1:maxage
axe(i)=i+20;
end
%
figure
plot( axe,teststat(:,2),'k', 'LineWidth',2 );
hold on
plot( axe,teststat(:,4),'-kx' );
plot(axe,teststat(:,6),'-ko','MarkerSize',3 );
title('Consumption, Earning, and Wealth Profiles');xlabel('age');
legend('消費','所得','資産',2);
hold off
%
toc

```

3 Endogeneous Grid Methods

4 Backward Induction

There are two main numerical methods for solving a dynamic programming, value function iteration and backward induction. To implement the latter, we need information on the terminal point, such as the value of the asset when agents stop the optimization. Compared to the backward induction, the value function iteration can be applied in a more general setting. Unfortunately, the value function iteration tends to be very slow. So, if it is possible, many researchers prefer the backward induction algorithms to the value function iteration.

In this lecture note, we apply the backward induction to the standard consumption model with precautionary saving, known as a buffer saving model by Zeldes and Carroll.

5 Buffer Stock Saving Model

Assume an individual i faces the following maximization problem at time zero. This individual lives for T periods.

$$\max E_0 \sum_{s=0}^T \beta^{t-s} u(C_s^i), \quad (50)$$

s.t.

$$W_{t+1}^i = R [W_t^i + Y_t^i - C_t^i], \quad (51)$$

$$Y_t^i = P_t^i V_t^i, \quad (52)$$

$$P_t^i = G_t P_{t-1}^i N_t^i \text{ for all } t, \quad (53)$$

where, Y : current income, P : permanent income, V : transitive income, G : growth component in the permanent income, N : shock to the permanent income, R : interest rate (given constant). Taking the log of (53) gives,

$$\ln P_t = \ln G_t + \ln P_{t-1} + \ln N_t. \quad (54)$$

Note that G_t is not a random variable, but a deterministic process corresponding to age-wage profile. Thus, if N_t is i.i.d., the log of permanent income follows a Martingale. Define the cash on hand at time t as the sum of the asset at the beginning of the period and current income,

$$X_t = W_t + Y_t, \quad (55)$$

then, we can obtain,

$$X_{t+1} = R[X_t - C_t] + Y_{t+1}. \quad (56)$$

The Bellman equation can be formulated as follows,

$$V_t(X_t, P_t) = \max_{C_t} \left\{ \frac{C_t^{1-\sigma}}{1-\sigma} + \beta E_t V_{t+1}(X_{t+1}, P_{t+1}) \right\}. \quad (57)$$

Note that both X_t and P_t are continuous variables. As we have seen before, the numerical solution of DP with two continuous state variables is difficult to solve. However, if the utility function is CRRA, we can use ‘‘Carroll Trick’’ to avoid the problem. Define the following new variables:

$$c_t = C_t/P_t,$$

$$x_t = X_t/P_t,$$

then, we can rewrite the Euler equation as follows,

$$C_t^{-\gamma} \geq E_t [C_{t+1}^{-\gamma} R\beta], \quad (58)$$

$$\begin{aligned} 1 &\geq E_t [C_t^{-\gamma} C_{t-1}^{\gamma} R\beta] \\ &\geq R\beta E_t \left[\left(\frac{c_{t+1}}{c_t} \right)^{-\gamma} \left(\frac{P_{t+1}}{P_1} \right)^{\gamma} \right], \end{aligned} \quad (59)$$

$$c_t^{-\gamma} \geq R\beta E_t \left[(c_{t+1})^{-\gamma} (G_{t+1}^E N_{t+1}^E)^{-\gamma} \right]. \quad (60)$$

Note that the equality holds when $X_t = 0$.

The budget constraint becomes,

$$\begin{aligned} x_{t+1} &= X_{t+1}/P_{t+1} \\ &= R[X_t - C_t]/P_{t+1} + Y_{t+1}/P_{t+1} \\ &= R[x_t - c_t] P_t/P_{t+1} + P_{t+1}V_{t+1}/P_{t+1} \\ &= R[x_t - c_t]/(G_{t+1}N_{t+1}) + V_{t+1}. \end{aligned} \quad (61)$$

Thus, the normalized Bellman equation can be written as follows:

$$v_t(x_t) = \max_{c_t} \left\{ \frac{c_t^{1-\sigma}}{1-\sigma} + \beta E_t \Lambda_{t+1}^{1-\sigma} v_{t+1}(x_{t+1}) \right\}, \quad (62)$$

$$\text{where} \quad \Lambda_{t+1} \equiv G_{t+1}N_{t+1}. \quad (63)$$

It can be shown that by setting $V_t = P_t^{1-\sigma} v_t$, (57) and (62) provides us with the same policy functions. See Carroll (2006) for detail.

There are several different ways to solve the above model. In this note, we adopt the Euler equation based method, which is said to be faster than the value function based method.

We can rewrite the Euler equation as,

$$\begin{aligned} c_t^{-\sigma} &= R\beta E_t \left[(c_{t+1})^{-\sigma} (G_{t+1}N_{t+1})^{-\sigma} \right] \\ &= R\beta E_t \left[c_{t+1} (R[x_t - c_t]/(G_{t+1}N_{t+1}) + V_{t+1})^{-\sigma} (G_{t+1}N_{t+1})^{-\sigma} \right]. \end{aligned} \quad (64)$$

Let's start from the decision at the terminal point, T . At time T , all the cash on hands are used in consumption (no need to save for future consumption). That is, $c_T(x_T) = x_T$. This implies,

$$c_{T-1}^{-\sigma} = R\beta E_{T-1} \left[(R[x_{T-1} - c_{T-1}]/(G_T N_T) + V_T)^{-\sigma} (G_T N_T)^{-\sigma} \right]. \quad (65)$$

Next, we divide the cash on hands x_{T-1} into several grids. For each grid, we calculate the right hand side of (65). In order to obtain c_{T-1} as a function of x_{T-1} , first, we need to calculate the expected value due to the random variables V_T and N_T , as well as the deterministic function of G_T . Assume the following stochastic process,

$$\begin{aligned} V &= 0 \quad \text{with Probability } p = 0.005, \\ &= Z \quad \text{with Probability } 1 - p. \end{aligned}$$

$$\begin{aligned} \ln Z &\sim TN(\mu_z, \sigma_{\ln Z}^2) : \text{Normal Truncated at } \pm 3\sigma_{\ln Z} + \mu_z. \\ \ln N &\sim TN(\mu_N, \sigma_{\ln N}^2) : \text{Normal Truncated at } \pm 3\sigma_{\ln N} + \mu_N. \end{aligned}$$

$$E_t N_{t+1} = E_t V_{t+1} = 1.$$

In other words, we assume that there is probability 0.5% of zero income. Because of this assumption and the No-Ponzi Game Condition, households cannot borrow, that is, liquidity constraint never binds. Note that precautionary saving occurs regardless of the sign of the third derivatives of utility function. It is also possible to derive precautionary saving motive just by assuming liquidity constraint, that is,

$$c_t \leq x_t. \tag{66}$$

Our current goal is to get the policy function, $c_{T-1}(x_{T-1})$. If we adopt the standard backward induction method, we divide x_{T-1} into several grids and obtain c_{T-1} that satisfies (65). However, c_{T-1} appears in the both sides of (65). Therefore, for each grid, we need to solve nonlinear equation to obtain c_{T-1} . The endogenous grid approach developed by Carroll (2006) is a very powerful and useful tool to avoid this complexity. Carroll proposes that instead of, x_{T-1} , we need to take grids of the end of asset at T-1, $w_{T-1} (= W_{T-1}/P_{T-1})$ ⁸. The end of period asset is a part of the cash on hand that is not consumed, that is,

$$\begin{aligned} W_{T-1} &= X_{T-1} - C_{T-1}, \\ w_{T-1} &= x_{T-1} - c. \end{aligned}$$

Then, we can eliminate c_t or c_{T-1} in (64) and (65),

$$\begin{aligned} c_t^{-\gamma} &= R\beta E_t \left[c_{t+1} (R[w_t] / (G_{t+1}N_{t+1}) + V_{t+1})^{-\gamma} (G_{t+1}N_{t+1})^{-\gamma} \right], \\ c_{T-1}^{-\gamma} &= R\beta E_{T-1} \left[(R[w_{T-1}] / (G_T N_T) + V_T)^{-\gamma} (G_T N_T)^{-\gamma} \right]. \end{aligned}$$

⁸Endogenous grid method have been applied in many applied research. See Barillas and Fernandez-Villaverde (2007) and Hintermaier and Winfried (2010) for more discussion on the endogenous grid method.

When solving the optimization problem at time t , we know the policy function at time $t+1$, $c_{t+1}(x_{t+1})$. Therefore, we do not need to use a nonlinear equation solver to calculate the right hand sides of (59) and (65).

When obtaining c_{T-1}^i for w_{T-1}^i , next, we can obtain the corresponding grid in cash on hand simply by adding the two,

$$x_{T-1}^i = w_{T-1}^i + c_{T-1}^i,$$

which gives us the policy function, $c_{T-1}(x_{T-1})$.

The model we have solved can be summarized as follows,

$$V_t(X_t, P_t) = \max_{C_t} \left\{ \frac{C_t^{1-\gamma}}{1-\gamma} + \beta E_t V_{t+1}(X_{t+1}, P_{t+1}) \right\}, \quad (67)$$

$$s.t. \quad X_{t+1} = R[X_t - C_t] + Y_{t+1},$$

$$W_t = X_t - C_t,$$

$$Y_t = P_t V_t,$$

$$P_t = G_t P_{t-1} N_t$$

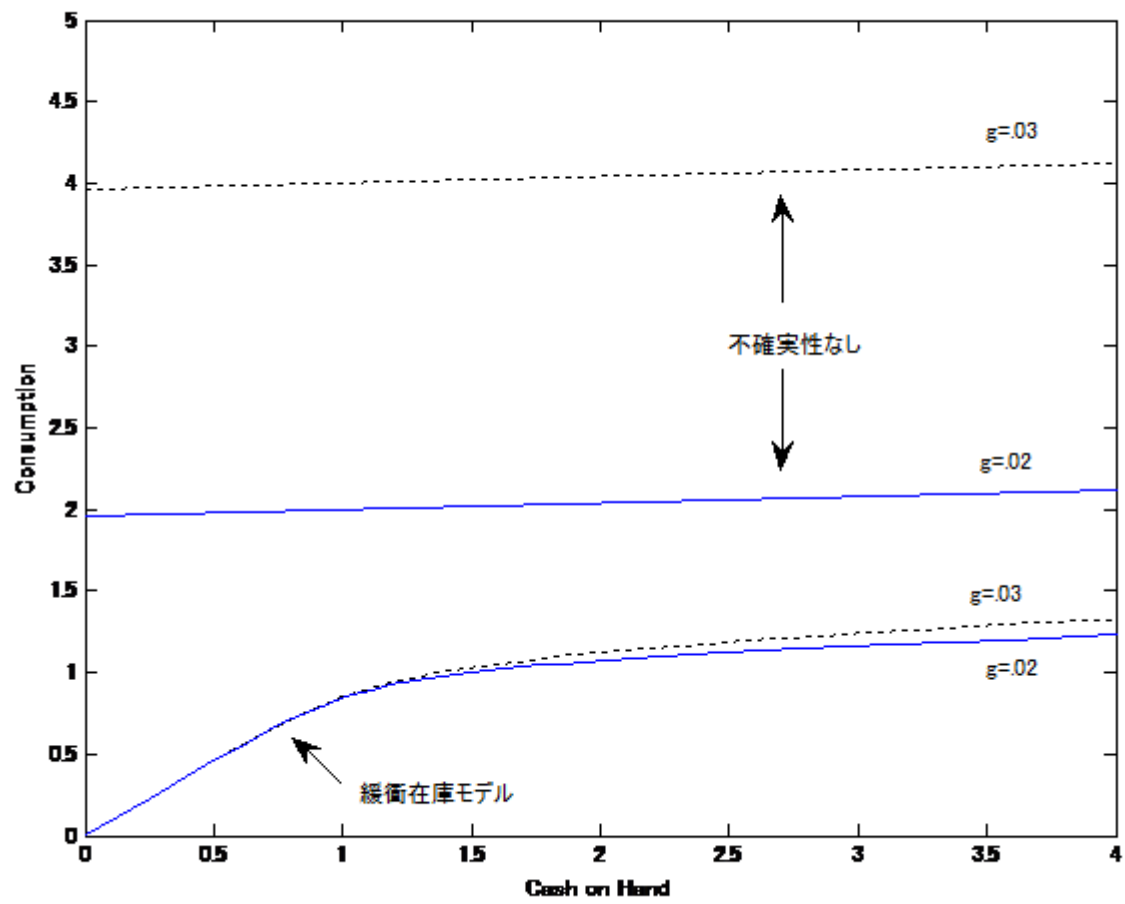
$$P_t = G_t P_{t-1} N_t.$$

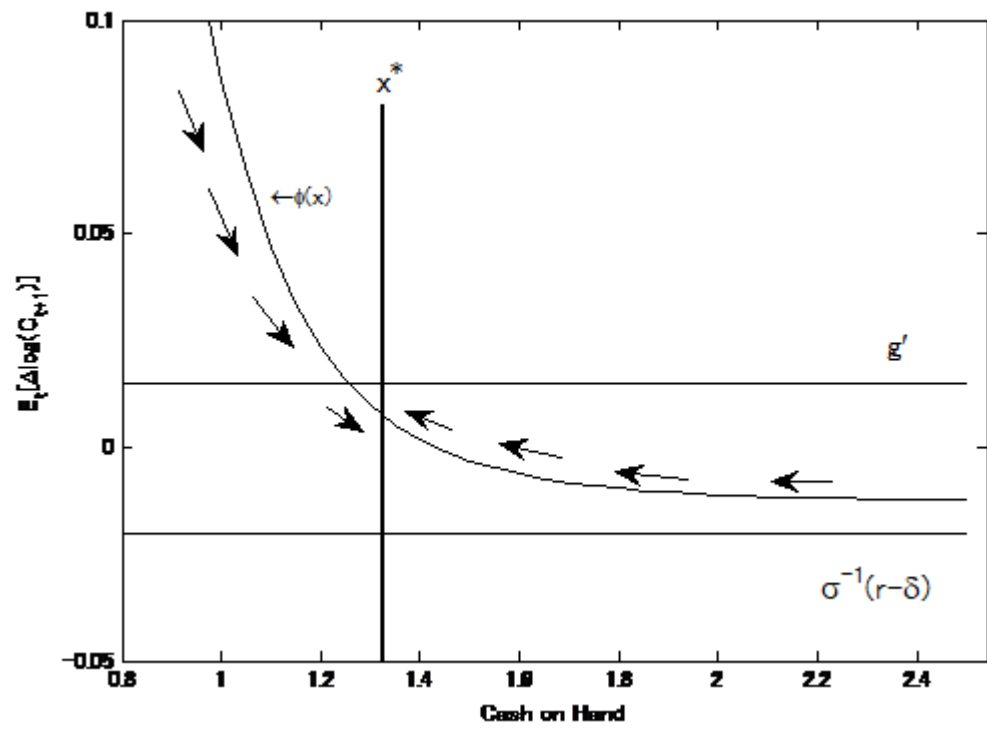
6 Further Topics

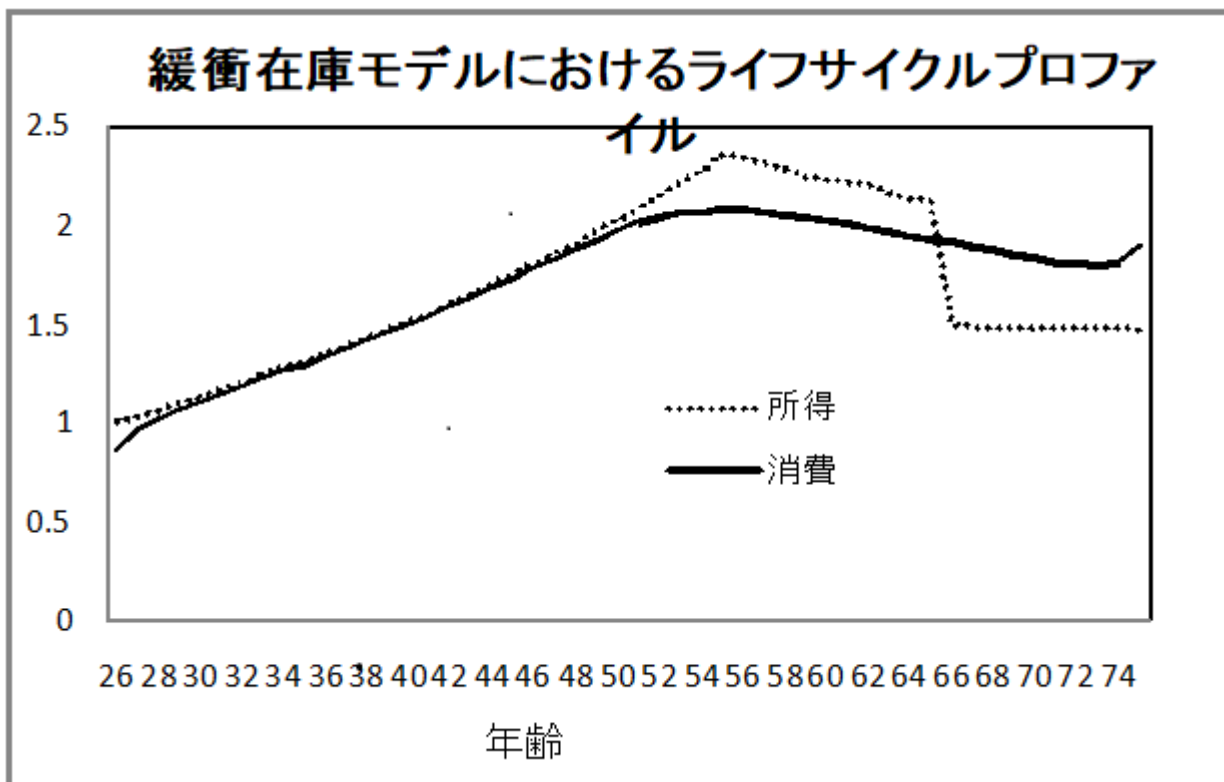
After Carroll's seminal work, endogenous grid methods combined with backward induction have been extended and applied to many fields, including labor supply (Barillas, Francisco & Fernandez-Villaverde, Jesus, (2007) "A generalization of the endogenous grid method," *Journal of Economic Dynamics and Control*, Elsevier, vol. 31(8), pages 2698-2712, August.), (2) aggregate shock and change in wealth distribution in incomplete capital market (Reiter, Michael, (2009) "Solving heterogeneous-agent models by projection and perturbation," *Journal of Economic Dynamics and Control*, Elsevier, vol. 33(3), pages 649-665, March.).

7 Matlab Code

The following matlab program solves the standard buffer stock saving model based on parameters calibrated for Japanese young households. To implement the code, we need two additional codes, (1) compecon tool box (by Miranda and Fackler, available in there web-site) and (2) spline_linear_val.m







(under the GNU LPGL Lincense). See my paper in Keizaikenkyu in 2013 for the detail of calibration.

```

%%%%%%%%%%
%% A simple buffer stock saving model
%
% Solving a finite time model by backward induction
%  $u = (c^{1-\gamma}) / (1-\gamma)$ 
%  $x(t+1) = R(x(t) - c(t)) / (G(t+1) * N(t+1)) + V(t+1)$ 
%
% Needs (1) Compecon toolbox for quadrature, and
% (2) spline_linear_val.m for piecewise linear spline
%
% May, 2014
% Naohito Abe
%
clear all
tic
%
%% Set Parameter Values
alpha=0.1;
gamma=3; % risk aversion
Beta=0.90; % Discount Factor
maxage=70; % the maximum age after age 20 (maxage + 20 =year)
retire_age=40; % Retirement at age 60
gridpoint=100; % the number of spline grid for cash on hands, COH.
Rate=1.02; % The rate of return on non risk assets
%% Deterministic Wage Schedules
agelist=linspace(20,90,71);
% Taken from 平成 23 年版厚生労働白書図 3-3-14
Dataforwage=[
20    1199.40    ;
25    1417.00    ;
30    1646.70    ;
35    1877.60    ;
40    2132.70    ;
45    2313.90    ;
50    2355.20    ;
55    2251.50    ;
60    1723.40    ;

```

```

65 1723.40/2 ;
70 1723.40/2 ;
75 1723.40/2 ;
80 1723.40/2 ;
85 1723.40/2 ;
90 1723.40/2 ];
% Use cubic spline to interpolate the wage-age profile
gspline=spline( Dataforwage(:,1), Dataforwage(:,2));
Growth1=ppval(gspline,agelist);
%
for i=1:70
Growth1(i)=spline.linear_val(length( Dataforwage(:,1)), Dataforwage(:,1)
, Dataforwage(:,2) ,agelist(i));
end
Growth=ones(70,1);
% Growth Rate
for i=2:70
Growth(i-1)=Growth1(i)/Growth1(i-1);
end
%% Permanent and Transitory Shock
% Permanent Shock
numgridforinteg = 10; % the number for integration
pvariance=0.023; % from 21 世紀縱断調査 (阿部 (2013))
pnum=numgridforinteg; % number of cdf points for integration
% transitory shock
tvvariance=0.0116;% from 21 世紀縱断調査 (阿部 (2013))
tnum=numgridforinteg; % number of cdf points for integration
%% For Integration (CDF Points)
minshock=0.00000001;
maxshock=2;
[xpoint,weightquad]=qnwlege(tnum,minshock,maxshock);% quadrature point
and weight
meantshock=sum(weightquad(:).*xpoint(:).*lognpdf(xpoint(:),0,sqrt(tvvariance)));
meanpshock=sum(weightquad(:).*xpoint(:).*lognpdf(xpoint(:),0,sqrt(pvariance)));
%
%% Euler Equation at the Last Period %%%%%%%%%%%
%  $c(T-1)^{-\gamma} = E[RB ( R(aGNV)^{-\gamma} ) (GN)^{-\gamma}]$ 
% Note  $cT=xT$ ,  $aT=0$ 
% Given grid of  $a(T-1)$ , obtaining the RHS by integration.

```

```

% using exp(exp(log(log))) gives more grids for smaller values (near zero)
% than endperiasset=(linspace(0.001, 500, gridpoint))
endperiasset= exp(exp(linspace(0,log(log(50+1)+1),gridpoint))-1)-1;
endperiasset= endperiasset';
forsum=zeros(gridpoint,1);
COH=zeros(gridpoint,1);
prevcons=zeros(gridpoint,1);
for i=1:gridpoint
COH(i)=(Rate*endperiasset(i)/(Growth(maxage)) )+1;
forsum(i)=(COH(i))^(-gamma);
prevcons(i)= (Rate*Beta*forsum(i))^(-1/gamma);
end
%% Backward Induction
COHtable=zeros(gridpoint,maxage);
prevconstable =zeros(gridpoint,maxage);
COHtable(:,maxage)=COH;
prevconstable(:,maxage)=COH;
%% Current Cash on Hands
COH=endperiasset+prevcons;
%% Policy Function at one period before the last
consumptionfull=zeros(gridpoint,1);
% Linear Spline is doing good job than cubic when extrapolating
for i=1:gridpoint
consumptionfull(i) = spline_linear_val(length(COH(:,1)),COH(:,1), prev-
vcons(:,1) ,COH(i));
end
COHtable(:,maxage-1)=COH;
prevconstable(:,maxage-1)=prevcons;
for t=2:maxage-1
invretire=maxage-retire_age;
forsumF=zeros(gridpoint,1);
forsum1=zeros(gridpoint,1);
prevcons1=zeros(gridpoint, 1);
if t<=invretire %after retireme
for i=1:gridpoint
xifull=(Rate*endperiasset(i)/(Growth(maxage-t+1)))+1;
cifull=spline_linear_val(length(COH(:,1)),COH(:,1), prevcons(:,1),xifull);
cifull=min(cifull,xifull);
forsum(i)=((cifull)^(-gamma)) *((Growth(maxage-t+1))^(-gamma));

```

```

prevcons1(i)= (Rate*Beta*forsum(i))^(1/gamma);
prevcons1(i)=max(prevcons1(i),0);
end
% update
for i=1:gridpoint
prevcons(i)= prevcons1(i);
end
COH=endperiassest+prevcons;
COHtable(:,maxage-t)=COH;
prevconstable(:,maxage-t)= prevcons;
end
if t>invretire %before the retirment
% xpoint, weightquad;
% expected marginal utility
for i=1:gridpoint
forsumF(i)=0;
for j=1:pnum
for l=1: tnum
% transform at to xt, and ct
xifull=((Rate*endperiassest(i)/(Growth(maxage-t+1)*xpoint(j)))+xpoint(l));
cifull= spline_linear_val(length(COH(:,1)),COH(:,1), prevcons(:,1) ,xifull);
cifull=min(cifull,xifull);
forsumfullmid=weightquad(l)*weightquad(j)*((cifull)^(-gamma)) *((Growth(maxage-
t+1)*xpoint(j))^(-gamma))*lognpdf(xpoint(j),0,sqrt(pvariance))*lognpdf(xpoint(l),0,sqrt(tvariance));
forsumF(i)=forsumfullmid+forsumF(i);
end
end
%consumption for each grid
forsum1(i)=forsumF(i);
prevcons1(i)= (Rate*Beta*forsum1(i))^(1/gamma);
prevcons1(i)=max(prevcons1(i),0);
end
% update
for i=1:gridpoint
prevcons(i)= prevcons1(i);
end
COH=endperiassest+prevcons;
prevconstable(:,maxage-t)= prevcons;
COHtable(:,maxage-t)=COH;

```

```

end
end
%% Plotting the policy function
gridforplot=(linspace(0,2, 200))';
consumption_plot=zeros(200, maxage);
%
for t=0:maxage-1
for i=1:200
consumption_plot(i,maxage-t)=spline_linear_val(length(COHtable(:,maxage-
t)), COHtable(:,maxage-t),prevconstable(:,maxage-t) ,gridforplot(i));
consumption_plot(i,maxage-t)=min(consumption_plot(i,maxage-t),gridforplot(i));
end
end
figure
plot( gridforplot, consumption_plot(:,1),'k', 'LineWidth',1 );
hold on
plot( gridforplot, consumption_plot(:,30),'kx', 'LineWidth',1 );
plot( gridforplot, consumption_plot(:,60),'o', 'LineWidth',1 );
title('Consumption');xlabel('Cash on Hand');
hold off
%% Life Cycle Simulation %%%%%%%%%%%%%%%
%
simnum=5000;% the number of households
sigma=0.1;% standard deviation of the initial wealth distribution
% mhu =0;% mean of the initial wealth distribution
mhu =1;% mean of the initial wealth distribution
cons_sim=zeros(simnum, maxage);% Simulated Consumption Path
wealth1=zeros(simnum,1);% Simulated Wealth Path
iniwealth=zeros(simnum,1);% Create the initial distribution of the wealth
for i=1:simnum
iniwealth(i)=lognrnd(mhu, sigma);
end
% setting very small initial wealth level
iniwealth=0.001*ones(simnum,1);
consumption=zeros(simnum, maxage);
currentincome=zeros(simnum,maxage);
currentincome2=zeros(simnum,maxage);
endofasset=zeros(simnum, maxage);
Perm=zeros(simnum,1);

```

```

Perm1=zeros(simnum,1);
gappershock=1/meanpshock;
gaptshock=1/meantshock;
for i=1:simnum
wealth1(i)=iniwealth(i);% initial wealth
Perm(i)=1;
t=1;
permshock=lognrnd(0,sqrt(pvariance))*gappershock;
tempshock=lognrnd(0,sqrt(tvariance))*gaptshock;
Perm1(i)=Perm(i)*Growth(t)*permshock;
xtilda=Rate*wealth1(i)*(1/(Growth(t)*permshock))+tempshock;
cons=spline.linear_val(length(COHtable(:,t)), COHtable(:,t),prevconstable(:,t)
,xtilda);
cons=min(cons, xtilda);
capx=xtilda*Perm1(i);
capcons=cons*Perm1(i);
income=xtilda*Perm1(i)- Rate*wealth1(i)*Perm(i)+(Rate-1)*wealth1(i)*Perm(i);
%update
Perm(i)=Perm1(i);
wealth1(i)=xtilda-cons;
consumption(i,t)=capcons;
currentincome(i,t)=income;
endofasset(i,t)=wealth1(i)*Perm(i);
for t=2:maxage
if t<=retire.age
permshock=lognrnd(0,sqrt(pvariance))*gappershock;
tempshock=lognrnd(0,sqrt(tvariance))*gaptshock;
Perm1(i)=Perm(i)*Growth(t)*permshock;
xtilda=Rate*wealth1(i)/(Growth(t)*permshock)+1*tempshock;
cons=spline.linear_val(length(COHtable(:,t)), COHtable(:,t),prevconstable(:,t)
,xtilda);
cons=min(cons, xtilda);
capx=xtilda*Perm1(i);
capcons=cons*Perm1(i);
income=xtilda*Perm1(i)- Rate*wealth1(i)*Perm(i)+(Rate-1)*wealth1(i)*Perm(i);
Perm(i)=Perm1(i);
wealth1(i)=xtilda-cons;
consumption(i,t)=capcons;
currentincome(i,t)=income;

```

```

endofasset(i,t)=wealth1(i)*Perm(i);
end
if t>retire_age
Perm1(i)=Perm(i)*Growth(t);
xtilda=Rate*wealth1(i)/Growth(t)+1;
cons=spline_linear_val(length(COHtable(:,t)), COHtable(:,t),prevconstable(:,t)
,xtilda);
cons=min(cons, xtilda);
capx=xtilda*Perm1(i);
capcons=cons*Perm1(i);
income=xtilda*Perm1(i)- Rate*wealth1(i)*Perm(i)+(Rate-1)*wealth1(i)*Perm(i);
Perm(i)=Perm1(i);
wealth1(i)=xtilda-cons;
consumption(i,t)=capcons;
currentincome(i,t)=income;
endofasset(i,t)=wealth1(i)*Perm(i);
end
end
end
%% Plotting the life cycle profiles
consumption_life=zeros(maxage, 1);
income_life=zeros(maxage, 1);
wealth_life=zeros(maxage, 1);
for t=1:maxage
consumption_life(t)=mean(consumption(:,t));
income_life(t)=mean(currentincome(:,t));
wealth_life(t)=mean(endofasset(:,t));
end
figure
plot(consumption_life,'k');
hold on
plot(income_life,'kx');
plot(wealth_life,'ko');
legend('Consumption ', 'Income',2, 'Asset', 3);
toc

```


8 Appendix

Backward Induction のアルゴリズムを実行する Matlab のコードである。
8GB のメモリーを積んだノート PC で、32 秒程度で計算可能である⁹。

⁹最後の Life Cycle Simulation の際に Statistics Toolbox を用いている。