社会保障とマクロ経済学

— 数値計算入門 —

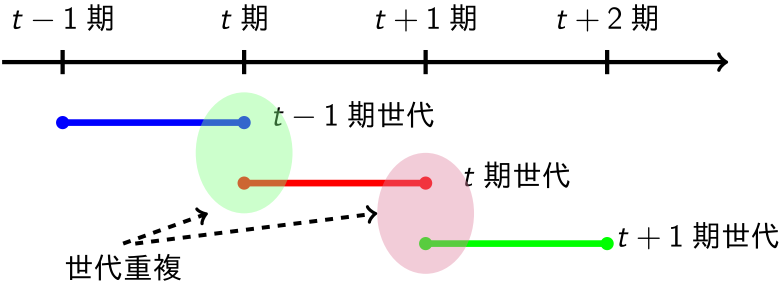
山田知明

明治大学
tyamada@meiji.ac.jp

2018 年 11 月 21 日&28 日@METI(アップデート版)

講義内容

世代重複モデル (復習)



世代重複モデル (復習)

- 世代重複モデルの定式化：t 期世代の目的関数

$$\max u(c_t^Y) + \beta u(c_{t+1}^O)$$

- 生涯予算制約

$$\begin{aligned} c_t^Y + a_{t+1} &= (1-\tau_t)w_t n_t, \\ c_{t+1}^O &= ss_{t+1} + (1+r_{t+1})a_{t+1}. \end{aligned}$$

  - c_t^Y：若年期の消費、c_t^O：老年期の消費、a_{t+1}：貯蓄
  - τ_t：社会保障料率、ss_t：年金

世代重複モデル (復習)

- 生産サイド
  - 総資本：$a_t = K_t$
  - 総労働：$n_t = L_t$
- 生産に投入

$$Y_t = A_t K_t^{\alpha} L_t^{1-\alpha}$$

- 政府の予算制約 (賦課方式)

$$\tau_t w_t n_t \mu_t = ss_t \mu_{t-1}$$

  - μ_t：t 期世代の人口

2 期間モデル

- 世代重複モデルのパーツ

$$\max\left\{u(c_1)+\beta u(c_2)\right\}$$

- 生涯予算制約

$$\begin{aligned} c_1 + a_2 &= y_1 + a_1, \\ c_2 &= y_2 + (1+r)a_2. \end{aligned}$$

  - c_1：第 1 期の消費、c_2：第 2 期の消費
  - a_1：第 1 期に保有している資産 (遺産)、a_2：第 2 期の資産
  - y_1：労働所得、y_2：年金、r：金利、β：割引因子

2 期間モデル (続き)

- 異時点間の最適条件

$$u'(c_1) = \beta(1+r)u'(c_2)$$

  - オイラー方程式 (Euler equation)
- このシンプルな 2 期間モデルで何を "解く"?
  - 言い換えると何が知りたい?
- 政策関数 (policy function)

$$\begin{aligned} c_1 &= f(a_1) \\ a_2 &= g(a_1) \end{aligned}$$

  - 注意：互いに独立ではない

カリブレーション

- 数値計算を行うためには、
  1. 関数型を特定化
  2. 関数のパラメータを設定

  $\Rightarrow$ カリブレーション (calibration)
  - 場合によっては推定 (estimation)
- 2 期間モデルのカリブレーション

$$u(c) = \frac{c^{1-\gamma}}{1-\gamma}$$

  - 相対的危険回避度一定 (CRRA: constant relative risk aversion)
  - γ : 異時点間の代替の弾力性の逆数
  - $\gamma = 1$ だと $u(c) = \ln c$

カリブレーション (続き)

- カリブレーションのターゲット
  - Song et al. (2012)：アメリカ経済
- β：年率で 0.985
  - 2 期間モデルだと 1 期間はおよそ 30 年間：$\beta = 0.985^{30}$
- r：金利 (今は部分均衡モデルなので外生)
  - 年率で 2.5%：$R \equiv 1 + r = 1.025^{30}$
- 労働所得&年金
  - y_1：1 に基準化
  - y_2：0.5(所得代替率より設定)

解析的解の性質

- シンプルなモデルなので解析的に解ける
  - 解析的解 (closed-form solution) が存在
- 知りたいのは貯蓄関数：$a_2 = g(a_1)$

$$a_2 = \frac{a_1 + y_1 - \{\beta(1+r)\}^{-1/\gamma} y_2}{1 + \{\beta(1+r)\}^{-1/\gamma}(1+r)}$$

- 理論的特徴
  - a_1 に関して線形

- 講義で使ったコードは Bitbucket からダウンロード可能
  - Projection method を除く
  - `https://bitbucket.org/TomoakiYamada/meti_slide2/downloads/`

離散近似

- コンピュータは連続 (continuous) を理解できない
  - 消費も貯蓄も本来は実数
  - $c > 0$、$a \in \mathbb{R}$
- 離散化 (discretize) して考える
  - グリッド (grid)、ノード (node)、評価点 (evaluation point)
  - Chatterjee et al. (2007)
- 例 (1)
  - $a_1 \in \{0, 1,000,000, 10,000,000, 100,000,000\}$
- 例 (2)
  - $a_1 \in \{0, 100, 200, ..., 100,000,000\}$
- 本スライド：11 個の評価点
  - $a_{1,(i)} \in \{0, 0.1, 0.2, ..., 1.0\}$

状態変数と操作変数を離散化

- 離散近似で 2 期間モデルを解いてみる
    - ⇒ 基本的な考え方は「総当たり」
- a_1 と a_2 は共に 11 個の値しか選択できない
    - 中間の値 (0.15 とか) は取ることが出来ない
    - $a_{1,(i)} \in \{0, 0.1, 0.2, ..., 1.0\}$
    - $a_{2,(j)} \in \{0, 0.1, 0.2, ..., 1.0\}$

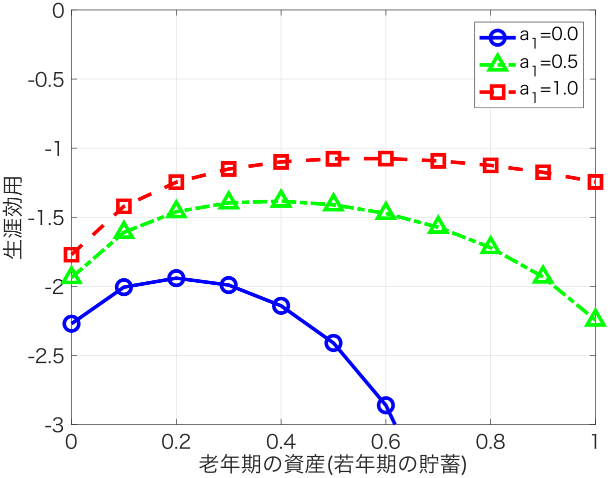
状態変数と操作変数を離散化 (続き)

- 2 期間モデルにおける効用最大化問題
  - c_1 と c_2 に代入

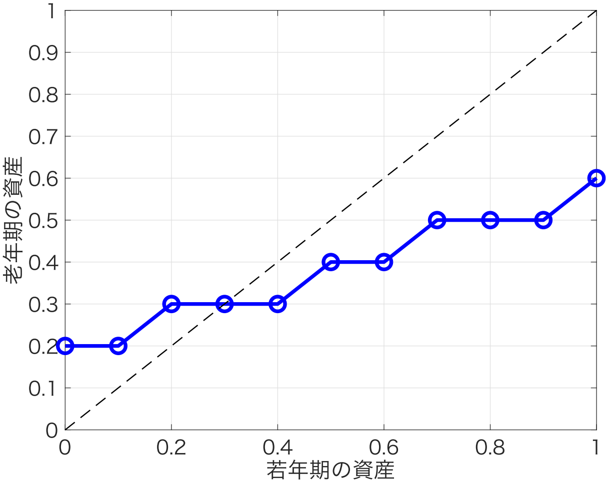
$$\max \frac{[y_1 + a_{1,(i)} - a_{2,(j)}]^{1-\gamma}}{1-\gamma} + \beta \frac{[y_2 + (1+r)a_{2,(j)}]^{1-\gamma}}{1-\gamma}$$

  - ある $a_{1,(i)}$ のもとで、生涯効用を最大にする $a_{2,(j)}$ を選ぶ問題
  - a_1：状態変数 (state variable)⇒11 種類
  - a_2：操作変数 (control variable)⇒11 種類
- 全ての組み合わせで $11 \times 11 = 121$ 種類の計算：総当たり

生涯効用



貯蓄関数



操作変数を連続関数にする

- 離散化の問題点
  1. グリッドの数を増やすと計算時間が指数的に増加
  2. グリッドを節約すると精度が悪い
- もう少し洗練した方法：操作変数は連続的に値を取る
- 2 期間モデルにおける効用最大化問題

$$\max_{a_2} \frac{[y_1 + a_{1,(i)} - a_2]^{1-\gamma}}{1-\gamma} + \beta \frac{[y_2 + (1+r)a_2]^{1-\gamma}}{1-\gamma}$$

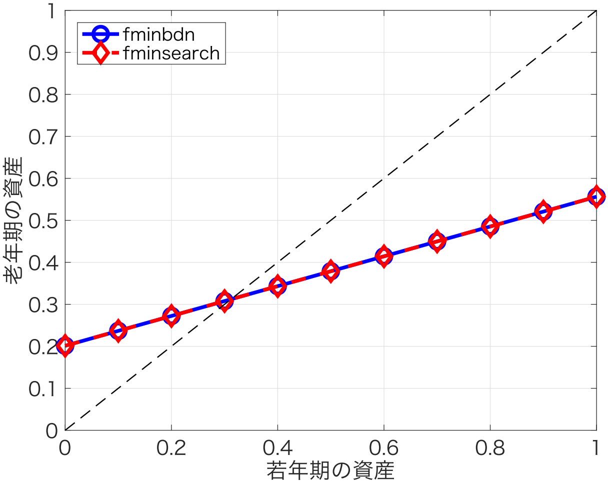
最適化

- どうやって解く?
  $\Rightarrow$ 最適化 (optimization)
- 数値計算ソフトには最適化ライブラリが存在
  - C、C++、Fortran、Matlab、Python、Julia etc.
  - 組み込まれている場合あれば自分で追加の場合もある

アルゴリズム

1. パラメータを設定 (カリブレーション)
2. $a_{1,(i)} \in \{a_{1,(1)}, ..., a_{1,(I)}\}$ を離散化した若年期の資産とする
3. 各 $a_{1,(i)}$ について、p.16 の式を最大にするような a_2 を探し出す
   - 最大値を探すためには、各言語に備わっている (あるいは外部の) 最適化関数を利用
4. 得られた各 $a_{1,(i)}$ と a_2 の組み合わせが貯蓄関数

最適化を使って導出した政策関数



動的計画法 (dynamic programming) はこの考え方を拡張して解く

一階条件を使う

- 多くの経済モデルは一階条件 (オイラー方程式) を使って記述
  - 経済学を学んだ人にとっては馴染みがある
  - 多くの過去のモデル達も一階条件で記述されていた
- 2 期間モデルの一階条件

$$u'(y_1 + a_1 - a_2) = \beta(1 + r)u'(y_2 + (1 + r)a_2)$$

離散化

- 状態変数を離散化：未知の変数は a_2 のみ

$$u'(\underbrace{y_1 + a_{1,(i)}}_{\text{given}} - \underbrace{a_2}_{\text{choice}}) = \underbrace{\beta(1+r)}_{\text{params.}} u'(\underbrace{y_2 + (1+r)a_2}_{\text{known}}))$$

- 知りたいこと

$$R(a_1) \equiv \beta(1+r)\frac{u'(y_2 + (1+r)a_2)}{u'(y_1 + a_{1,(i)} - a_2)} - 1$$

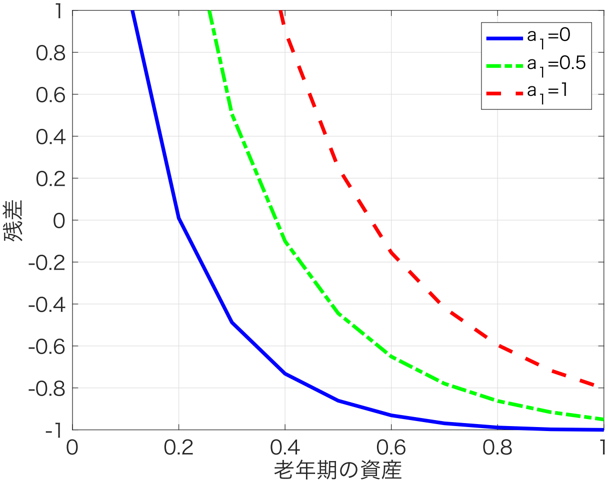
をゼロにするような a_2 は何か?

  - 残差関数 (residual function)$R(a_1) = 0$ を探す問題

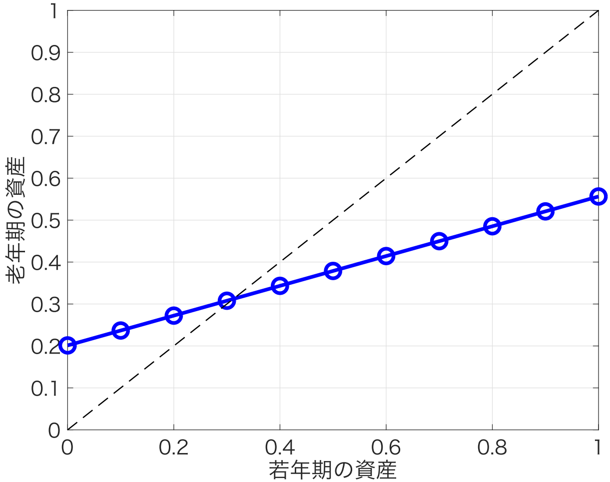
求根法のアルゴリズム

- 残差関数は一般的に非線形 (nonlinear)
- 求根問題 (root-finding problem)⇒nonlinear equation solver
- 最適化と同様、求根アルゴリズムは様々なプログラム言語で書かれている
  - あるプログラム言語が数値計算言語として適切か?= 最適化や求根アルゴリズムのような便利なライブラリが充実しているか?
- 数値計算用の理論を学ぶ必要あり
  - Judd (1998)、Miranda and Fackler (2004)、Heer and Maussner (2009) などを参照

残差関数



求根アルゴリズムを使って導出した政策関数



射影法

- 今までの方法の共通点
  - 状態変数を離散化して、グリッド上の最適な値を計算
- 射影法 (projection method)
  - 関数全体を近似
  - Judd (1992) など
- 貯蓄関数を N 次の多項式 (polynomials) で近似

$$\hat{g}(a;\psi) = \sum_{n=0}^{N} \psi_n a^n$$

$\Rightarrow$ 知りたいのは $\{\psi_n\}_{n=0}^{N}$

射影法 (続き)

- 真の貯蓄関数 (本来は知らない)

$$
\begin{aligned}
a_2 &= \psi_0 + \psi_1 a_1, \\
\psi_0 &= \frac{y_1 - \{\beta(1+r)\}^{-1/\gamma} y_2}{1 + \{\beta(1+r)\}^{-1/\gamma}(1+r)}, \\
\psi_1 &= \frac{1}{1 + \{\beta(1+r)\}^{-1/\gamma}(1+r)}
\end{aligned}
$$

射影法 (続き)

- 近似した政策関数を残差関数に代入

$$R(a;\psi) \equiv \beta(1+r)\frac{u'(y_2+(1+r)\hat{g}(a_1;\psi))}{u'(y_1+a_1-\hat{g}(a_1;\psi))}-1 \approx 0$$

- 残差 “関数” がゼロとは?

$$\int \omega(a)||R(a;\psi)||da = 0$$

  - $\omega(a)$: 重み (weight)
- 実際に計算するのは評価点 $\{a_{1,(i)}\}$ 上の残差

$$\sum_{i=1}^{I} R(a_{(i)};\psi)^2 \approx 0$$

射影法のアルゴリズム

1. パラメータを設定する (カリブレーション)
2. 評価点 $a_{1,(i)} \in \{a_{1,(1)}, ..., a_{1,(I)}\}$ を定める
   - 今回はこれまで使ってきた等分のグリッドと同じで 0 から 1 の区間を 0.1 刻みで 11 個設定
3. 近似したい政策関数の関数形を決める
   - 今回は単項式で 1 次関数 ($\hat{g}(a;\psi) = \psi_0 + \psi_1 a_1$)
4. ある $\{\psi_0, \psi_1\}$ をインプットとして、残差の 2 乗和 $\langle R(a;\psi), R(a;\psi)\rangle$ を計算して返すサブルーチンを書く
5. Matlab などに備わっているゼロ点を探しだすサブルーチン・ライブラリを使って、近似的に残差関数をゼロにするような $\{\psi_0^*, \psi_1^*\}$ を見つける

得られた結果 (Matlab)

- 真の値
  - $\frac{y_1 - \{\beta(1+r)\}^{-1/\gamma} y_2}{1+\{\beta(1+r)\}^{-1/\gamma}(1+r)} = 0.2013$
  - $\frac{1}{1+\{\beta(1+r)\}^{-1/\gamma}(1+r)} = 0.3550$
- 数値計算結果
  - $\psi_0 = 0.2010$
  - $\psi_1 = 0.3556$

内生的格子法

- 最適化や求根法の問題点
  1. 初期値によっては見つけるのに失敗
  2. 時間がかかる
- 計算速度と安定性を高める様々な研究がある
- 内生的格子法 (EGM: endogenous grid method)：Carroll (2006)
  - 一階条件を用いる
  - 状態変数と操作変数を近似するタイミングをずらす
  - 一階条件を導出できれば高速

EGM のアルゴリズム

1. パラメータを設定 (カリブレーション)
2. 老年期の資産 $a_{2,(j)} \in \{a_{2,(1)}, \dots, a_{2,(J)}\}$ を離散化
3. オイラー方程式式の右辺を、
$$\Gamma(a'_{2,(j)}) \equiv \beta(1+r)[y_2 + (1+r)a_{2,(j)}]^{-\gamma}$$
と定義
4. 効用関数が CRRA 型であれば、限界効用関数の逆関数を計算することは可能
$$c_1 = \Gamma(a'_{2,(j)})^{-\frac{1}{\gamma}}$$
5. ステップ 3 を離散化したあらゆる $a_{2,(j)}$ について計算すれば、ある $a_{2,(j)}$ の下でオイラー方程式を満たす $c_{1,(j)}$、すなわち最適な消費と貯蓄の組み合わせを計算したことになる
6. 予算制約 $c_{1,(j)} + a_{2,(j)} = y_1 + a_1 \equiv x_1$ から $a_{1,(j)}$ を逆算

3 期間モデル

- 2 期間モデルは “非現実的” なので 3 期間に拡張
- 目的関数

$$U(c_1, c_2, c_3) = \frac{c_1^{1-\gamma}}{1-\gamma} + \beta \frac{c_2^{1-\gamma}}{1-\gamma} + \beta^2 \frac{c_3^{1-\gamma}}{1-\gamma}$$

- 予算制約

$$\begin{aligned} c_1 + a_2 &= y_1 + a_1, \\ c_2 + a_3 &= y_2 + (1+r)a_2, \\ c_3 &= y_3 + (1+r)a_3 \end{aligned}$$

一階条件

- 一階条件

$$c_1^{-\gamma} = \beta(1+r)c_2^{-\gamma},$$
$$c_2^{-\gamma} = \beta(1+r)c_3^{-\gamma},$$

- どうやって解く?
  - 後ろ向きに解く
    $\Rightarrow$ バックワード・インダクション (backward induction)
- 何が違う?
  - 2–3 期目は前と同じ!
  - これまで説明したどの方法でも OK!

$$u'(y_2 + (1+r)a_{2,(j)} - a_3) = \beta(1+r)u'(y_3 + (1+r)a_3)$$

貯蓄関数を近似する

- 1 期目から 2 期目にかけては?
- 解きたいオイラー方程式

$$\begin{aligned} u'(y_1 + a_{1,(j)} - a_2) &= \beta(1+r)u'(y_2 + (1+r)a_2 - a_3), \\ &= \beta(1+r)u'(y_2 + (1+r)a_2 - g(a_2)) \end{aligned}$$

- これまではオイラー方程式の右辺は “全部消費する” ので簡単に計算が出来た = 次期の消費 (貯蓄) 関数を知っていた
- 3 期間モデルの場合、2 期目の消費関数について知っているのは近似して計算した $\{a_{2,(j)}, c_{2,(j)}\}$ の組み合わせ
  - $a_2 \in \{a_{2,(j)}, a_{2,(j+1)}\}$ だったら $g(a_2)$ をどうやって計算する?

近似

- 数値計算の手法の一つに近似 (interpolation) がある
- 線形補間 (linear interpolation)：多くの人が思いつく近似

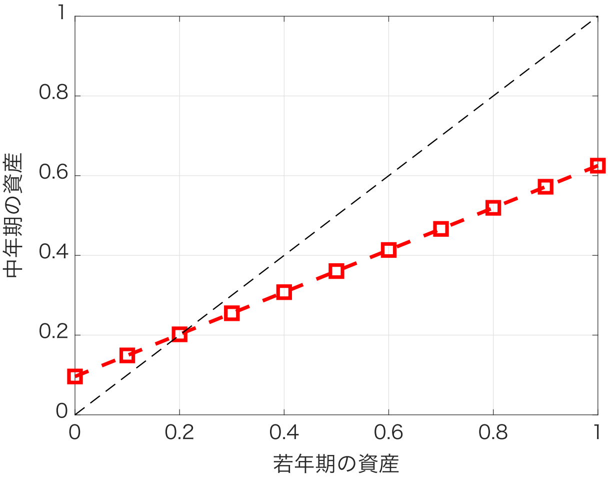
$$\tilde{a}_3 = \tilde{g}\big(a_{2,(j)}\big)\frac{a_{2,(j+1)} - x}{a_{2,(j+1)} - a_{2,(j)}} + g\big(a_{2,(j+1)}\big)\frac{x - a_{2,(j)}}{a_{2,(j+1)} - a_{2,(j)}}$$

- これで十分な場合も多いがより洗練された手法がある
  - 微分可能性：スプライン近似 (spline interpolation)
  - 凹 (凸) 関数：shape-preserving spline
  - チェビシェフ多項式 (Chebyshev polynomial) など

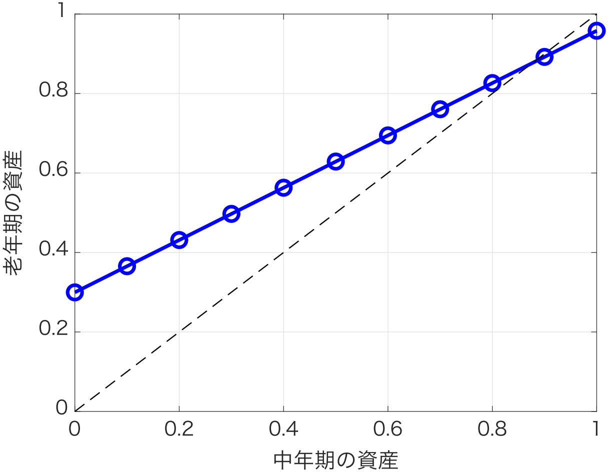
3 期間モデルのアルゴリズム

1. 2 期間モデルから 3 期間モデルに拡張しているため、パラメータを再考
    - モデルの 1 期間を 30 年から 20 年に変更
    - 年功序列型賃金：$y_1 = 1$、$y_2 = 1.2$
2. 離散化した中年期の資産：$a_{2,(j)} \in \{a_{2,(1)}, \dots, a_{2,(J)}\}$
3. 各 $a_{2,(j)}$ について、オイラー方程式を満たす $a_{3,(j)}$ を探す
4. 離散化した若期の資産：$a_{1,(i)} \in \{a_{1,(1)}, \dots, a_{1,(I)}\}$
5. $a_{2,(j)}$ と $a_{3,(j)}$ を近似する内挿法を一つ定める
    - 3 次のスプライン関数などを用いる場合、スプライン係数を計算
6. ステップ 3 で得た中年期の貯蓄関数 $\{a_{2,(j)}, a_{3,(j)}\}$ を所与として、各 $a_{2,(i)}$ について、オイラー方程式を満たす a_2 を探す

若年期の貯蓄関数



老年期の貯蓄関数



次回：多期間ライフサイクルモデルを実際に解く

参考文献

- Carroll, Christopher D. (2006): "The Method of Endogenous Gridpoints for Solving Dynamic Stochastic Optimization Problems," *Economics Letters*, 91, 312–320.
- Chatterjee, Satyajit, Dean Corbae, Makoto Nakajima and Jose-Victor Rios-Rull, (2007): "A Quantitative Theory of Unsecured Consumer Credit with Risk of Default," *Econometrica*, 75, 1525–1589.
- Judd, Kenneth L. (1992): "Projection Methods for Solving Aggregate Growth Models," *Journal of Economic Theory*, 58, 410–452.
- Song, Zheng, Kjetil Storesletten and Fabrizio Zilibotti (2012): "Rotten Parents and Disciplined Children: A Politico-economic Theory of Public Expenditure and Debt," *Econometrica*, 80, 2785–2803.

参考文献

- Heer, Burkhard and Alfred Maussner (2009): *Dynamic General Equilibrium Modeling: Computational Methods and Applications*, Springer.
- Judd, Kenneth L. (1998): *Numerical Methods in Economics*, The MIT Press.
- Miranda, Mario J. and Paul Fackler (2004): *Applied Computational Economics and Finance*, The MIT Press.