Dynamic Macroeconomics

Summer Term 2011

Christian Bayer

Universität Bonn

May 15, 2011

Overview of the Course: My Part

1. Introduction, Basic Theory
2. Stochastic Fluctuations
3. Numerical Tools
4. Estimation
5. Equilibrium
6. Stationary heterogenous agent economies
7. Heterogeneous agent economies with aggregate fluctuations

Literature

Primary reading:

- Heer, B. and A. Maussner (2009), "Dynamic General Equilibrium Modelling", 2nd edition, Springer, Berlin.
  I cover roughly: Ch. 1, 4, 7, 8

Secondary reading:

- Adda, J. and R. Cooper (2004): "Dynamic Economics", MIT Press, Cambridge.
- Ljungqvist, L. und T. Sargent (2004): "Recursive Macroeconomic Theory", MIT press, Cambridge.
- Stockey, N.L. and Lucas, R.E. with E.C. Prescott (1989): "Recursive Methods in Economic Dynamics", Chapters 4 and 9, Havard University Press, Cambridge..

Motivation:

The standard incomplete markets model
Hugget (1993), Aiyagari (1994) and Krussel and Smith (1998)

What we want to model...

- Consider an economy in which households supply labor, consume and accumulate capital.
- They each face idiosyncratic risk as their endowment with efficiency units of labor changes over time (they may be employed or unemployed for example).
- Other than in a standard (i.e. complete markets) macro model, we want to assume that households cannot trade away their idiosyncratic risk on complete asset markets (there is no perfect unemployment insurance).
- We want to assume that the only asset households can trade is a claim on the aggregate stock of capital.
- We want to understand the quantitative implications.

What do we need to model...

- How households make their consumption-savings decisions given prices (forward looking behavior)?
- Heterogeneity of households in wealth
- The effect of aggregate fluctuations on this heterogeneity and its repercussions on prices.

Households

Putting this a little more formally:

- Continuum of households that obtain income from supplying labor n_t and assets a_t at prices w_t and r_t respectively.
- Households enjoy utility from consumption c_t, are expected utility maximizers and are infinitely lived.
- They discount future utility by the discount factor β.
- [Labour supply $n_t \in [n_{\min}, n_{\max}]$ is an exogenous stochastic process.]

Budget and borrowing constraints

- We want to assume (following Aiyagari, 1994) that there are no contingent claims households can trade.
- Households can only self-insure using the asset a.
- Asset holdings must satisfy that the household is able to service eventual debt in any case

$$a' \geq b$$

- b is an exogenous debt limit (necessary to avoid Ponzi schemes if $r < 0$). We will typically set $b = 0$.

Household Planning Problem

Setup

- The planning problem of each households can be summarized in the Bellman equation

$$\max_{<a_t>_{t=1\ldots\infty} a_t \in \mathbb{R}_+} E_0 \sum_{t=0}^{\infty} \beta^t u\left(w_t n_t + a_t\left(1+r_t\right) - a_{t+1}\right)$$

- where the sequence $< a_t >$ is contingent on the history of realizations for $\left(w_t, r_t, n_t, Z_t\right)$
- Z_t is an aggregate state of the economy that determines the probability distributions for n_t, i.e. it determines aggregate employment.

Household Planning Problem

Issues in Solving

- The planning problem

$$\max_{<a_t>_{t=1\ldots\infty} a_t \in R_+} E_0 \sum_{t=0}^{\infty} \beta^t u\left(w_t n_t + a_t\left(1+r_t\right) - a_{t+1}\right)$$

  yields in general first order conditions, with occasionally binding constraints.

$$-\beta^t u'\left(c_t\right) + \beta^{t+1}\left(1+r_t\right) u'\left(c_{t+1}\right) = \lambda_t$$

  where λ_t is the Lagrangian multiplier on the non-negativity constraint for a_t.
- This cannot be solved using local approximations!
- Another difficulty arises in form of the prices (w_t, r_t), which may fluctuate with aggregate employment and households need to form expectations about these price movements.

Part 1

Introduction, Dynamic Programming: Basic Theory and First Numerical Tools

The Theory of Dynamic Programming

1. Back to Consumption Theory 101: Indirect Utility and Value Functions
2. Basic dynamic programming in a finite horizon environment
3. Stationary Problems and Infinite Horizon Dynamic Programming
4. Stochastic Dynamic Programming
5. Analytical Solutions and Numerical Analysis

Numerical Analysis of Dynamic Programming Problems

1. Continuous vs. discrete decision problems
2. Discrete Approximations to a Continuous State Space
3. Value Function Iteration
4. Transition Probability Matrices and Markov Chain
5. Autoregressions and Tauchen's Algorithm

Indirect Utility

Recall from consumer theory the concept of **indirect utility**:

$$V\left(p, I\right) = \max_{c \geq 0} u\left(c\right),\ s.t.\ pc \leq I.$$

Reflects the highest utility level, a household can achieve by consumption choice, given his income I and the price vector p.

- We can think of $\left(p, I\right)$ as indexing the **state of the economy**.
- Therefore we will call $\left(p, I\right)$ the **state vector**.

Profit functions

- Analogously, we have the concept of a (short-run) profit function in production theory:

$$\Pi(p, w, K) = \max_{L} pF(K, L) - wL - rK.$$

- Again, we can think of (p, w, K) as indexing the **state of the economy for the firm**.
- While the concept of indirect utility / reduced form profit functions mainly helps in simplifying some analysis in static consumer theory (Slutsky-decomposition for example), it will turn out to be a very powerful tool in a dynamic analysis.

General formulation, finite time horizon

Now consider a dynamic problem of a generic form:

$$\max \sum_{t=0}^{T} \beta^t u\left(x_t, x_{t+1}\right), \text{ s.t. } x_{t+1} \in \Gamma_t\left(x_t\right), \ x_0 \text{ given.}$$

where $u\left(x_t, x_{t+1}\right)$ is a payoff function that depends on the current state x_t as well as on the future state x_{t+1} chosen by the decision maker.

General formulation, finite time horizon

Denoting the indirect utility obtained as $V(x_t, t)$, we can restate the Problem as a so called **Bellman equation**

$$V(x_t, t) = u(x_t, x_{t+1}) + \beta V(x_{t+1}, t+1), \text{ s.t. } x_{t+1} \in \Gamma_t(x_t).$$

where V exists if Γ is compact valued (Theorem of the Maximum).
The time-index t reflects the fact that it matters **how many remaining periods** there are.
With finite horizon the optimization problem is necessarily **non-stationary,** i.e. changes with time t.

A cake eating example

To fix ideas consider the usage of a depletable resource (cake-eating)

$$\max \sum_{t=0}^{T} \beta^t u\left(c_t\right), \text{ s.t. } W_{t+1} = W_t - c_t, \; c_t \geq 0, \; W_0 \text{ given.}$$

To put this in the general form, expressing the problem only in terms of **state variables $\mathbf{W}_t$** we replace $c_t = W_t - W_{t+1}$

$$\max \sum_{t=0}^{T} \beta^t u\left(W_t - W_{t+1}\right), \text{ s.t. } W_{t+1} \leq W_t.$$

A cake eating example

As formulation in terms of the Bellman equation, we obtain

$$V\left(W_t, t\right) = \max u\left(W_t - W_{t+1}\right) + \beta V\left(W_{t+1}, t+1\right), \text{ s.t. } W_{t+1} \leq W_t.$$

Bellman equations and infinite time horizon

While the dynamic programming (Bellman equation) approach generates somewhat more information about the optimization problem in a finite horizon setup than a direct attack on the problem, it is at the same time more burdensome, since we need to determine V for each t.
It becomes a powerful approach really only when we look at infinite time horizon problems.
These **can be stationary,** i.e. does not change in t, as the remaining time until the end of the decision problem remains always ∞.

Stationary dynamic programming

If the problem is stationary (and a solution does exist), we can state the planning problem as

$$V(x) = \max u(x, y) + \beta V(y) \text{ s.t. } y \in \Gamma(x).$$

- Note however that not all infinite horizon problems are stationary. Sometimes a problem can be reformulated in stationary terms (You know that from econometrics).
- Also note that a solution may not exist.
- The unknown of the Bellman equation is **the function** $V(x)$

An Example

The neo-classical growth model

A social planner wants to maximize the stream of utility from consumption in an economy, where

$$
\begin{aligned}
U &= U(C_t) \\
Y_t &= C_t + I_t;\ \ Y_t = K_t^{\alpha} \\
K_{t+1} &= (1-\delta) K_t + I_t \\
V(K_0) &= \max_{<K_t>_{t=1\ldots\infty}} \sum_{t=0}^{\infty} \beta^t U\left[C\left(K_t, K_{t+1}\right)\right] \\
s.t.\ C &= K_t^{\alpha} + (1-\delta) K_t - K_{t+1} \\
K_{t+1} &\geq 0 \\
K_{t+1} &\leq K_t^{\alpha} + (1-\delta) K_t
\end{aligned}
$$

An Example

The neo-classical growth model

We can rewrite this as

$$
\begin{aligned}
V(K) &= \max_{<K_t>_{t=1\ldots\infty}} \sum_{t=0}^{\infty} \beta^t U\left[C\left(K_t, K_{t+1}\right)\right] \text{ with } K_t = K \\
&= \max_{K_{t+1}} \left\{ u\left(K, K_{t+1}\right) + \max_{<K_{t+1}>_{t=1\ldots\infty}} \beta \sum_{t=0}^{\infty} \beta^t u\left[C\left(K_{t+1}, K_{t+2}\right)\right] \right. \\
&= \max_{K' \in \Gamma(K)} \left\{ u\left(K, K'\right) + \beta V\left(K'\right) \right\} \\
\Gamma(K) &:= \left\{ k \in \mathbb{R}_+ \middle| k \leq K^{\alpha} + (1-\delta) K \right\} \\
u\left(K, K_{t+1}\right) &:= U\left[C\left(K_t, K_{t+1}\right)\right]
\end{aligned}
$$

When does a solution exist?

We can formulate the Bellman equation as a mapping

$$U(x) = T(V(x))$$

$$T[V(x)] = \max_y u(x, y) + \beta V(y) \text{ s.t. } y \in \Gamma(x) \tag{1}$$

that maps function V to a new function U.

Some prerequisits

Definition

A sequence (x_n) is said to be a Cauchy sequence on the metric space (C, d) if for all $\varepsilon > 0$ there exists a $n(\varepsilon)$, such that for all $m, n \geq n(\varepsilon) : d(x_m, x_n) < \varepsilon$.

Lemma

In a complete metric space, all Cauchy sequences converge, i.e. for any Cauchy sequence (x_n), *there is an* $x \in C$, *such that* $d(x_n, x) \longrightarrow 0$ *as* $n \longrightarrow \infty$. *(without proof)*

Some prerequisits

Definition

The mapping T is a contraction on the complete metric space $\{C, d\}$ if for any $x, y \in C \Rightarrow Tx, Ty \in C$ and for some $\rho < 1$: $d\left(Tx, Ty\right) \leq \rho d\left(x, y\right)$.

Lemma

If T is a contraction then $T^n x$ is a Cauchy sequence (to be proof as Excercise 1).

Remark

Note that the set of continuous bounded functions is a complete metric space, with the sup*-norm as metric.*

Exercise 1

Show that if T is a contraction then $T^n x$ is a Cauchy sequence !

Existence of a Solution to the Bellman equation

The contraction mapping theorem

Theorem

If the Bellman equation (1) *defines* T *to be a* **contraction mapping** *on the* **set of continuous bounded functions**, *then a solution to* (1) *exists.*

Existence

Proof.

1.) note that if (1) has a solution, then it is a fixed point of T and vice versa.

2.) if the contraction mapping T has a fixed point, then it is unique: Let

$$\begin{aligned} Tx &= x, Ty = y \Rightarrow \\ d(x,y) &= d(Tx,Ty) \leq \rho d(x,y) \\ &\Rightarrow d(x,y) = 0 \\ &\Rightarrow x = y \end{aligned}$$

$\square$

Existence

Proof.

3.) T is continuous, i.e. $y_n \to y \Rightarrow Ty_n \to Ty$ as

$$\begin{aligned} d\left(Ty_n, Ty\right) &\leq \rho d\left(y_n, y\right) \overset{n\to\infty}{\to} 0 \\ &\Rightarrow Ty_n \to Ty \end{aligned}$$

4.) the limit $\lim_{n\to\infty} T^n x =: x^*$ exists because $T^n x$ is Cauchy (see Exercise 1).

5.) this implies

$$\begin{aligned} Tx^* &= T\left(\lim_{n\to\infty} T^n x\right) \\ &= \lim_{n\to\infty} TT^n x = \lim_{n\to\infty} T^{n+1} x = x^*, \end{aligned}$$

so that x^* is the fixed point of T, which concludes the proof. □

An algorithm to solve the Bellman equation

The proof to show existence of a solution to the Bellman equation is constructive. It tells us how to find a solution to the Bellman equation:

1. Show that T is a contraction.
2. Start with some initial guess V_0 and then construct a sequence $V_n = TV_{n-1}$.
3. After a sufficiently large number of iterations V_n will become arbitrarily close to the solution V.
4. This algorithm is called "Value-Function-Iteration" (VFI).

Blackwell's condition

Theorem

If T fulfills the following conditions, then T is a contraction mapping on the set of bounded and continuous functions:

1. *T preserves boundedness.*
2. *T preserves continuity.*
3. *T is monotonic: $w \geq v \Rightarrow Tw \geq Tv$*
4. *T satisfies discounting, i.e. there is some $0 \leq \beta < 1$, such that for any real valued constant c and any function v we have $T\left(v + c\right) \leq Tv + \beta c$.*

A solution exists in the generic case

Theorem (Existence of the value function)

Assume $u(x, x')$ *is* **real-valued, continuous, and bounded**, $0 < \beta < 1$, *and that the constraint set* $\Gamma(s)$ *is a* **non-empty, compact-valued, and continuous** *correspondence. Then there exists a unique continuous value function* $V(s)$ *that solves the Bellman equation* (1).

Policy function

Theorem (Existence of the policy function)

Assume $u(x, x')$ is real-valued, continuous, strictly **concave** *and bounded, $0 < \beta < 1$, and that the set of potential states is a* **convex** *subset of $\mathbb{R}^k$ and the constraint set $\Gamma(s)$ is a non-empty, compact-valued, continuous,* **and convex** *correspondence. Then the unique value function $V(s)$ is continuous and strictly concave. Moreover the* **optimal policy**

$$\phi(x) := \arg\max_{y \in \Gamma(x)} u(x, y) + \beta V(x)$$

is a continuous (single-valued) function.

A simple optimal capital accumulation model

- To put things into practice, we consider the Ramsey-Cass-Koopmans growth model in a simplified version:
- Consider a household having a utility function

$$u(c_t, n_t) = \ln(c_t)$$

- The consumption good is produced from a depreciating capital good:

$$\begin{aligned} y_t &= zk_t^{\alpha},\ 0 < \alpha < 1, \\ k_{t+1} &= k_t(1-\delta) + i_t,\ 0 \leq \delta \leq 1, \\ y_t &= c_t + i_t. \end{aligned}$$

A simple optimal capital accumulation model

1. How can we put this into the terms of dynamic programming?
2. Consumption is the **policy variable**.
3. Capital is the **state variable.**
4. $$V\left(k_t\right) = \max_{\substack{c_t \leq y_t - i_t \\ y_t = A k_t^{\alpha} \\ k_{t+1} = k_t(1-\delta) + i_t}} \ln\left(c_t\right) + \beta V\left(k_{t+1}\right).$$
5. This problem fulfills all assumptions of our 2 central theorems.

Well this looks like a simple model, ...

- but ...
- You'll need a computer to solve it.
- Unless you fix δ to 1,
- which is what we will do for didactical reasons:
  1. We learn to understand the problem
  2. We can compare approximate numerical solutions to the "true" analytical solution

Full depreciation

- Now the Problem takes the form

$$V\left(k\right)=\max_{k'\leq Ak^{\alpha}+k(1-\delta)}\ln\left(zk^{\alpha}-k'\right)+\beta V\left(k'\right).$$

- Guess that $V\left(k\right)=A+B\ln\left(k\right)$.
- This yields the FOC for the optimal policy

$$-\frac{1}{zk^{\alpha}-k'}+\beta V'\left(k'\right)=0$$

Full depreciation

- Now the Problem takes the form

$$V(k) = \max_{k' \leq Ak^{\alpha} + k(1-\delta)} \ln\left(zk^{\alpha} - k'\right) + \beta V\left(k'\right).$$

- Guess that $V(k) = A + B\ln(k)$.
- Plugging in the guess for V :

$$\left(zk^{\alpha} - k'\right)^{-1} = \beta B k'^{-1}$$

Full depreciation

- Now the Problem takes the form

$$V(k) = \max_{k' \leq Ak^{\alpha} + k(1-\delta)} \ln\left(zk^{\alpha} - k'\right) + \beta V\left(k'\right).$$

- Guess that $V(k) = A + B\ln(k)$.
- Plugging in the guess for V :

$$\beta B\left(zk^{\alpha} - k'\right) = k'$$

Full depreciation

1. Now the Problem takes the form

$$V(k) = \max_{k' \leq Ak^{\alpha} + k(1-\delta)} \ln\left(zk^{\alpha} - k'\right) + \beta V\left(k'\right).$$

2. Guess that $V(k) = A + B\ln(k)$.
3. Optimal policy

$$\frac{\beta B}{(1+\beta B)} zk^{\alpha} = k'$$

Full depreciation

1. Now the Problem takes the form

$$V(k) = \max_{k' \leq Ak^{\alpha} + k(1-\delta)} \ln\left(zk^{\alpha} - k'\right) + \beta V\left(k'\right).$$

2. Guess that $V(k) = A + B\ln(k)$.
3. Optimal policy

$$\frac{\beta B}{(1+\beta B)} y = k'$$

Full depreciation

Plug in optimal policy and $V(k) = A + B\ln(k)$

$$\begin{aligned} A + B\ln(k) &= \ln\left(zk^a\left(1-\frac{\beta B}{1+\beta B}\right)\right) \\ &\quad +\beta\left(A + B\ln\left(\frac{\beta B}{(1+\beta B)}zk^\alpha\right)\right) \end{aligned}$$

which can be restated as

$$\begin{aligned} &A(1-\beta) + \ln(k)\left[B - \alpha(1+\beta B)\right] = \\ &\qquad \ln(z)(1+\beta B) - \ln(1+\beta B)(1+\beta B) + \beta B\ln(\beta B) \end{aligned}$$

Full depreciation

Therefore

$$\begin{aligned} B &= \alpha\left(1+\beta B\right) \\ B &= \frac{\alpha}{1-\alpha\beta} \end{aligned}$$

This means

$$1+\beta B = \frac{1}{1-\alpha\beta}$$

So that

$$\begin{aligned} A\left(1-\beta\right) &= \left(\frac{1}{1-\alpha\beta}\right)\left[\ln\left(z\right)+\ln\left(1-\alpha\beta\right)\right]+\frac{\alpha\beta}{1-\alpha\beta}\ln\left(\frac{\alpha\beta}{1-\alpha\beta}\right) \\ &= \ln\left(1-\alpha\beta\right)+\frac{\ln\left(z\right)+\alpha\beta\ln\left(\alpha\beta\right)}{1-\alpha\beta} \end{aligned}$$

Interpretation

1. Indeed our guess solves the Bellman equation! (It is a fixed point of T)
2. The household saves the fraction $\frac{\beta B}{(1+\beta B)} = \alpha\beta$ of current income $y = zk^{\alpha}$.
3. The analytical solution breaks down as soon as $\delta < 1$.

Extending our model to stochastic environments

- So far we considered only situations in which all state variables were perfectly controlled by the decision maker.
- However, this is unrealistic in most economic situations.
  1. The situation may depend on some variable governed by a stochastic law of motion.
  2. The situation may involve interdependent choices (i.e. a game)
- We'll only look into the first case. The second case is more specialized, but it is basically covered by applying the concept of Markov-perfect equilibria.

Extending our model to stochastic environments

- Thinking of stationary influences in terms of an additionally state variable allows to integrate stochastic elements into the DP setup.
- We define the stochastic version of the Bellman equation as

$$V\left(x,\xi\right)=\max_{y\in\Gamma(x,\xi)}u\left(x,y,\xi\right)+\beta\mathbb{E}_{s'}V\left(y,\xi'\right).$$

Extending our model to stochastic environments

- What is needed is that the process governing s is an ergodic Markov process, i.e.
    1. for a sufficiently rich description of the history of the process captured by vector s, the transition probability is time and history independent
    2. the process is stationary.
- Again, sometimes we may achieve stationarity by reformulating the problem (e.g. consider a "cointegrated." system in which the decision maker wishes to track ξ by adjusting the state x).

Exercise 2

Exercise (2)

Consider the growth model with only capital where productivity z is stochastic and $\ln z$ follows an AR-1 process $\ln z' = \zeta (1 - \rho) + \rho \ln z + \varepsilon$ with $E(\varepsilon) = 0$. Assume $\delta = 1$. Show that the value function can be written as $V(k, z) = A + B \ln k + C \ln z$.

Getting started

The first algorithm we want to study is the **Value Function Iteration** outlined before. It focuses on the Bellman equation, computing the value functions by foward iterations from an initial guess. Independent of the solution of choice, numerical analysis can be thought of as having 4 steps:

1. Choice of functional forms
2. Approximation
    2.1 For methods involving discretization: Discretization of state and control variables
    2.2 For methods involving approximation of the policy function: Choice of parametric family
3. Building the computer code
4. Evaluating numerical results: policy and value functions, simulation, estimation.

Functional Forms

To be able to solve a dynamic programming problem numerically, we need to **specify all functions** involved in the problem and assign values to their parameters. Some functional forms are more common than others:

- Consumption: CRRA functions (log utility if $\gamma = 1$): $u(c) = \frac{c^{1-\gamma}-1}{1-\gamma}$.
- Production: CES production functions (Cobb-Douglas functions if $\rho = 0$) : $f(\vec{x}) = \left(\sum_i \alpha_i x_i^{\rho}\right)^{1/\rho}$
- Aggregators: CES-functions: $C(\vec{c}) = \left(\sum_i \alpha_i c_i^{\rho}\right)^{1/\rho}$
- Cost functions: linear, quadratic, fixed: $C(x) = F + \alpha x + \frac{\beta}{2}(x - \bar{x})^2$

Parameter choice

After choosing the functional forms involved in the dynamic programming problem we need to fix all parameters. There are various ways to do so:

- Take parameters published in other empirical studies
  - Pros: quickest, most orthodox, most likely to be accepted
  - cons: the estimations did not consider the effects you highlight in your study and may be biased
- Perform calibration
  - Pros: Relatively quick, actual data involved in the parameter choice (data guided)
  - Cons: No standard errors for parameters (how reliable are the estimates)
- Estimation
  - Pros: Reliability of parameters is assessed, parameter consistent under the model
  - Cons: Can be very time consuming up to infeasible

Discretization of the state space

- We need to define the space spanned by the state and control variables as well as the space of shocks in case of a stochastic problem.
- In the simplest case, the problem will already be of discrete form (e.g. work - no work).
- In most economic situations, however, the spaces involved in the theoretical problem will be defined on a continuous space. For example:
  - Consumption choice
  - Investment
  - AR(p) processes for shocks
- In all these cases we need a discrete approximation to the state space.

Discretization of the state space

- The discretization typically involves three choices
  1. Bounds to the state space, e.g. take $S = \times_i \left[\underline{S}_i, \overline{S}_i\right]$ for subsets from $\mathbb{R}^k$.
  2. Number of points in each dimension of the state space.
  3. Position of the points on the intervals.
- Each decision is non trivial and if possible should be made theory guided:
- What is the relevant domain?
  - Reflecting boundaries: Are there states s_L, s_H such that

$$\begin{aligned} \phi(s) &< s \forall s \geq s_H \\ \phi(s) &> s \forall s \leq s_L? \end{aligned}$$

  - In a deterministic model: reasonable size
  - In a stochastic model: sufficiently large probability mass, good approximation of **all relevant moments.** (Models with emphasis on higher order terms will need to be more accurate than models emphasizing only first moment.)

Discretization of the state space: Problems

- Not necessarily do reflecting boundaries exist:
  - Cake eating example from the introduction:
  - It can be shown that the household optimally consumes a fraction $(1-\beta)$ of the remaining cake each period
  - $\phi(W) = \beta W$
  - $W_n = \beta W_{n-1} = \beta^n W_0$
  - This converges to zero, which is a point outside the feasible set of cake-sizes if $u(c) = \ln(c)$.
  - Increasing the number of grid-points does not solve the problem: N the numerical solution $V^{(N)} \nrightarrow V$ as $N \rightarrow \infty$.
  - Only approximations, redefining u, as e.g. in Exercise 3.
- Smaller problem: we often do not know the reflecting boundaries even if they exist
- Then do a trial-and-error search

Discretization of the state space: Exercise

Exercise (3)

Solve the cake-eating problem analytically for $u(c) = \ln(c)$*! Then write a programme that solves a numerical approximation to this problem. For this approximation redefine*

$$\hat{u}(c) = \begin{cases} \ln(c) & c > C \\ \ln(C) & c \leq C \end{cases}.$$

Compare the analytical and the numerical solution both graphically as well as in terms of the max*-norm for* $C = \exp(-6.5)$*! What is the optimal grid you should use? How does the approximation quality on* $[0.1, 2]$ *change by altering* C *and by altering* N.

Putting things to work: non-stochastic case

- Value function Iteration loops over

$$V^n = T\left(V^{n-1}\right) = \max_{s' \in \Gamma(s)} u\left(s, s'\right) + \beta V^{n-1}\left(s'\right)$$

  until $\left|V^n - V^{n-1}\right| < crit$
- To implement this, we specify N discrete points in the state space $s_1, s_2, ..., s_N$ and denote $V_i^n = V^n\left(s_i\right)$.
- Note: In the stochastic case, V_i^n includes the expectations operator.
- Let $U\left(i\right) = \left(u(i,1),\ u\left(i,2\right), ...,\ u\left(i,N\right)\right)$ be the vector of all possible instantaneous utility levels obtained by going from state i to j (if impossible, set to $-\infty$).

Putting things to work

- Optimizing starting from state i

$$V_i^n = \max\left\{U(i) + \beta \mathbf{V}^{n-1}\right\}$$

- Denote $\iota_N' = (1, ..., 1)$, and stacking the above expression, we obtain

$$\mathbf{V}^n = \max\left\{\mathbf{U} + \beta \iota_N \mathbf{V}^{n-1}\right\}$$

  where bold typeset indicates linewise stacked variables. Maximum is line-by-line.

- Note: MATLAB max does column-wise max, so you'll need to use $\max(x, 2)$

Putting things to work

- Switch to MATLAB! (Simple growth problem)

Some remarks on efficient programme codes

- At least if you want to do estimation, you'll be quickly running into problems with computation time.
- Therefore efficient programming is essential.
- In the programme codes presented there is some inefficiency.
  - Utility levels that correspond to state-to-state transitions, for example, are calculated each time the value functions is iteratively redetermined.
  - It would be better to calculate them at higher hierarchy and pass them on to *Val_Fu_I.m*.
- Use the **MATLAB Profiler** to find time expensive steps of the calculation. Use **M-Lint** for hints on efficient programming, e.g. pre-allocating growing variables in loops.
- Avoid loops by making use of Matrix-Algebra.
- MATLAB can handle up to 3-dimensional arrays quickly, above that dimensionality performance is worse than loops.

Exercise 4

Throughout this class we will build two computer-based models: (a) an economy with capital accumulation, including frictions to this; (b) a model of optimal savings in an economy with incomplete markets and (endogenous) borrowing constraints.

Exercise (4)

Write a MATLAB code that solves the simplified Ramsey-Cass-Koopmans model for $\delta < 1$ *for fixed productivity* z.

Exercise 5

Exercise (5)

Consider the following model: A household is endowed with a fixed income stream of $y_t = \rho^t y_0$, $\rho < (1+r)^{-1}$ discounts future utility by factor β and its felicity function (instantaneous utility) is a CRRA function

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

The household can save and borrow at the risk-free rate r.

1. *Derive optimal consumption for $\beta(1+r) = 1$ for given current levels of non-human wealth D.*
2. *Write a MATLAB code that solves for optimal consumption given $V(D)$ for $\rho = 1$.*
3. *Solve for V by value function iteration, compare graphically to the analytic solution for the policy function.*

Literature

Primary Reading:

- Heer, B. and A. Maussner (2009), "Dynamic General Equilibrium Modelling", 2nd edition, Springer, Berlin. (Ch. 1)

Secondary reading:

- Adda, J. and R. Cooper (2004): "Dynamic Economics", MIT Press, Cambridge.
- Ljungqvist, L. und T. Sargent (2004): "Recursive Macroeconomic Theory", MIT press, Cambridge.
- Stockey, N.L. and Lucas, R.E. with E.C. Prescott (1989): "Recursive Methods in Economic Dynamics", Chapters 4 and 9, Havard University Press, Cambridge..

Part 2

Stochastic Fluctuations, Simulations and Further Numerical Tools

Programme of the week

1. Discussing the Exercises
2. Modelling Stochastic Fluctuations
   2.1 Markov Chains
   2.2 Tauchen's Algorithm
   2.3 Simulating a Markov Chain
3. Further numerical tools to solve SDP
   3.1 Policy Function Iteration
   3.2 Interpolation
   3.3 Multigrid Algorithms
   3.4 Projection methods

Stochastic Fluctuations and Markov Chains

- We model stochastic fluctuations on a grid of discrete stochastic states.
- If the stochastic fluctuations follow a stationary process, then they can be modelled in terms of a Markov chain.
- A Markov chain is characterized by states $\mathfrak{s}_i$, $i = 1...N$ and a transition probability matrix $P = \left(p_{ij}\right)$ that gives the probability to move from state i to state j.

Stochastic Fluctuations and Markov Chains

- If we denote the stochastic states by $\mathfrak{s}_i$ and the states directly determined by policy choice by x_j, then it is useful to define the value function as a matrix

$$\mathbf{V} = \left(v_{ij}\right)_{\substack{i=1\ldots N \\ j=1\ldots M}} = \left(V\left(\mathfrak{s}_i, x_j\right)\right)_{\substack{i=1\ldots N \\ j=1\ldots M}}.$$

- Then we can calculate the expected value as

$$E_{\mathfrak{s}}\left(V\left(\mathfrak{s}', x'\right)\right) = P\mathbf{V} = \left(\sum_k p_{ik} v_{kj}\right)_{\substack{i=1\ldots N \\ j=1\ldots M}}.$$

Markov Chains

- For details see Sargent/Ljungqvist, Ch. 2, I will be brief and introduce mainly some notation / definitions.
- For a given probability distribution π_0, the recursion

$$\pi_t = P\pi_{t-1}$$

  determines the sequence of probability distributions for the Markov Chain.
- The limit $t \longrightarrow \infty$ of π_t is called the "time invariant", "stationary" or " ergodic" distribution of the Markov Chain.
- Since

$$\begin{aligned} \pi^* &= \lim_{t\to\infty} P^t \pi_0 \\ P\pi^* &= P \lim_{t\to\infty} P^t \pi_0 = \lim_{t\to\infty} P^t \pi_0 = \pi^*. \end{aligned}$$

Markov Chains

- This limit distribution hence solves

$$(I - P)\,\pi^* = 0$$

and corresponds to the unit-eigenvector of P (if the limit exists).

- The limit does exist if for some $k \geq 1$ all elements of $P^k = \underbrace{P \times P \times \cdots \times P}_{k \text{ times}}$ are strictly positive (i.e. it is possible to reach each state from each state after k steps).

- We can calculate it as $\lim_{t\to\infty} P^t e_i$, e_i is the i-th unit-vector $e_i := \left(0 \ldots \underset{i-th \text{ position}}{1} \ldots 0\right)$.

Markov Chains, Two examples

Example

The Markov chain characterised by

$$P = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 1/2 \\ 0 & 1/2 & 1/2 \end{bmatrix}$$

has an ergodic distribution, since

$$P^2 = \begin{bmatrix} \frac{1}{2} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{2} \end{bmatrix}$$

is positive everywhere.

Markov Chains, Two examples

Example

The Markov chain characterised by

$$P = \begin{bmatrix} 1/2 & 1/2 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 \\ 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 1/2 & 1/2 \end{bmatrix}$$

has no ergodic distribution, since the upper and the lower block of states do not "connect".

Markov Chain Approximation of AR(1) Processes

- Economic theory often assumes that the stochastic variable follows a stationary AR(p) process with normal innovations ε_t, particularly $p = 1$ is most common.
- Tauchen (1987) suggests an algorithm to approximate such a process by a Markov chain.
- Consider the following process

$$y_t = \mu(1-\rho) + \rho y_{t-1} + \varepsilon_t,\ \varepsilon_t \sim N\left(0, \sigma^2\left(1-\rho^2\right)\right)$$

- The ergodic distribution of this process is $N\left(\mu, \sigma^2\right)$.

Markov Chain Approximation of AR(1) Processes

An example

Consider the 3-state Markov Chain

$$z_1 = -\sqrt{1.5}\sigma,\ z_2 = 0,\ z_3 = \sqrt{1.5}\sigma$$

with transition probabilities

$$P = \begin{bmatrix} 1-p & p & 0 \\ p/2 & 1-p & p/2 \\ 0 & p & 1-p \end{bmatrix}$$

The ergodic distribution π solves

$$(I-P)\,\pi = 0$$

Markov Chain Approximation of AR(1) Processes

An example

Hence

$$\begin{bmatrix} p & -p & 0 \\ -p/2 & p & -p/2 \\ 0 & -p & p \end{bmatrix} \begin{pmatrix} \pi_1 \\ \pi_2 \\ \pi_3 \end{pmatrix} = 0$$

so that $\pi_i = 1/3$.
This yields

- $E(z) = 0$
- and $E(z^2) = \frac{1}{3}\left(\frac{3}{2}\sigma^2 + 0 + \frac{3}{2}\sigma^2\right) = \sigma^2$
- and finally $E(z_{t+1}z_t) = \frac{1}{3}\left((1-p)\frac{3}{2}\sigma^2 + 0 + (1-p)\frac{3}{2}\sigma^2\right) = (1-p)\sigma^2.$

Exercise 6

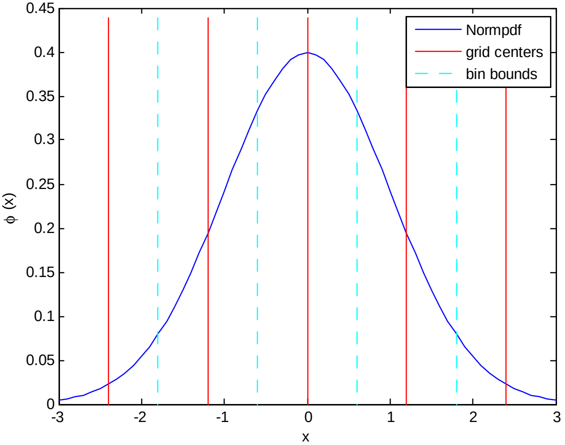
Exercise (6)

Calculate conditional variances $E\left(z_{t+1}^2|z_t = z_i\right) - E\left(z_{t+1}|z_t = z_i\right)^2$, *and unconditional skewness and kurtosis for the above three state Markov Chain. How do these compare to a normally distributed AR-1 process.*

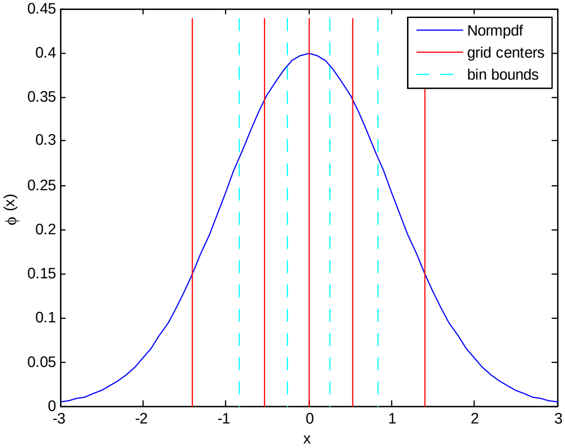
Markov Chain Approximation of AR(1) Processes

- We want to discretize the process over N gridpoints, reflecting N bins.
- Two sensible ways to choose gridpoints:
  1. Take a α % mass interval of the ergodic distribution (e.g. $\mu \pm 2\sigma$) and use equidistant points
  2. Choose the N bins such that each has a probability measure of $\frac{1}{N}$ in the ergodic distribution
- The first option is much easier to calculate. The latter should yield a better approximation - though it not always does.
- For very large N the numerical advantage of the first option makes this the option of choice.
- However, if a high degree of accuracy is necessary Tauchen's method is preferable and a large number of points is necessary for higher order behavior.

First option: equidistant



Second option: equi-likely / importance sampling



Tauchen's Algorithm

The Algorithm goes in three steps

1. Generate N bins to discretize the state space

$$Y = \underbrace{[y_1, y_2)}_{Y_1} \cup \underbrace{[y_2, y_3)}_{Y_2} \cup ... \cup \underbrace{[y_N, y_{N+1}]}_{Y_N}$$

2. Calculate the conditional expectation for each bin. This is to be used as the representative element of the bin in the solution of the numerical model (grid of Y).
3. Calculate transition probabilities $p_{i,j} = P\left(y_{t+1} \in Y_j | y_t \in Y_i\right)$

Generate Bins

To obtain a grid that is a good approximation at points that are reached often, we choose the boundaries of bins Y_i such that

$$P\left(y \in Y_i\right) = \Phi\left(\frac{y_{i+1}-\mu}{\sigma}\right) - \Phi\left(\frac{y_i-\mu}{\sigma}\right) = \frac{1}{N},$$

where Φ is the CDF of a standard normal. That is we make all bins equally likely to occur in the **long run, i.e. in the ergodic distribution of** y.

Therefore, we choose boundaries, such that

$$\Phi\left(\frac{y_{i+1}-\mu}{\sigma}\right) = \frac{i}{N}$$

or

$$y_i = \sigma\Phi^{-1}\left(\frac{i-1}{N}\right) + \mu.$$

Representative element

- The next step is to calculate the "centers" of the bins as

$$z_i = E\left(y \middle| y \in Y_i\right).$$

- Plugging in the distribution function $\phi\left(x\right) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right)$ of a standard normally distributed variable, we obtain

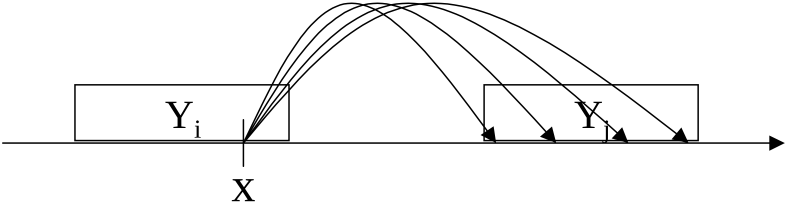
$$z_i = \frac{\int_{\frac{y_i-\mu}{\sigma}}^{\frac{y_{i+1}-\mu}{\sigma}} \left(\sigma x + \mu\right) \phi\left(x\right) dx}{P\left(y \in Y_i\right)}$$

- Observe that $\int x\phi\left(x\right) dx = -\phi\left(x\right) + F$ and $P\left(y \in Y_i\right) = N^{-1}$
- Then, we obtain

$$z_i = N\sigma\left[\phi\left(\frac{y_i - \mu}{\sigma}\right) - \phi\left(\frac{y_{i+1} - \mu}{\sigma}\right)\right] + \mu$$

Calculating Transitions

- The last - and in any terms most complicated - step is to calculate transitions.
- For each $x \in Y_i$ there are many "targets" in Y_j that can be reached



- And then there are many $x \in Y_i$ to start from.

Calculating Transitions

- Fix one $x \in Y_i$. What is the probability to reach an $y \in Y_j$?
- We need

$$y_j \leq \rho x + \mu \left(1 - \rho\right) + \varepsilon$$

and

$$y_{j+1} \geq \rho x + \mu \left(1 - \rho\right) + \varepsilon$$

- To put it differently

$$\varepsilon \in \left[y_j - \rho x - \mu \left(1 - \rho\right), y_{j+1} - \rho x - \mu \left(1 - \rho\right)\right]$$

Calculating Transitions

- With ε being normally $\left(0, \sigma^2\left(1-\rho^2\right)\right)$ distributed, we obtain with $\sigma_\varepsilon^2 = \sigma^2\left(1-\rho^2\right)$

$$P\left(y_{t+1} \in Y_j \cap y_t = x\right) = \Phi\left(\frac{y_{j+1} - \rho x - \mu\left(1-\rho\right)}{\sigma_\varepsilon}\right) - \Phi\left(\frac{y_j - \rho x - \mu\left(1-\rho\right)}{\sigma_\varepsilon}\right)$$

- Integrating over all x (weighted by their density), we obtain

$$P\left(y_{t+1} \in Y_j \cap y_t \in Y_i\right) = \frac{1}{\sqrt{2\pi\sigma}} \int_{y_i}^{y_{i+1}} \exp\left\{\frac{-\left(x-\mu\right)^2}{\sigma^2}\right\} \times \Phi\left(\frac{y_{j+1} - \rho x - \mu\left(1-\rho\right)}{\sigma_\varepsilon}\right) - \Phi\left(\frac{y_j - \rho x - \mu\left(1-\rho\right)}{\sigma_\varepsilon}\right) dx$$

Calculating Transitions

- Finally, making use of $P\left(y_t \in Y_i\right) = \frac{1}{N}$, we obtain

$$p_{i,j} = P\left(y_{t+1} \in Y_j | y_t \in Y_i\right) = \frac{N}{\sqrt{2\pi\sigma}} \int_{y_i}^{y_{i+1}} \exp\left\{\frac{-\left(x-\mu\right)^2}{\sigma^2}\right\} \times$$
$$\Phi\left(\frac{y_{j+1} - \rho x - \mu\left(1-\rho\right)}{\sigma_\varepsilon}\right) - \Phi\left(\frac{y_j - \rho x - \mu\left(1-\rho\right)}{\sigma_\varepsilon}\right) dx$$

- The latter integral has to be determined numerically.
- This is computationally burdensome!

Calculating Transitions

- Therefore an approximation to the Tauchen approximation is

$$p_{i,j} = P\left(y_{t+1} \in Y_j | y_t = z_i\right) = \Phi\left(\frac{y_{j+1} - \rho z_i - \mu\left(1-\rho\right)}{\sigma_\varepsilon}\right) - \Phi\left(\frac{y_j - \rho z_i - \mu\left(1-\rho\right)}{\sigma_\varepsilon}\right)$$

- This can also be used for equi-spaced grids!

Exercises 7

Exercise (7)

Program both the simplified and the full Tauchen (1987) *algorithm!*

Exercises 8

Exercise (8)

Simulate both alternative Tauchen approximations as well as equispaced grid approximations (with $\pm 2\sigma$ grid) to the process

$$y_t = 0.9 y_{t-1} + \varepsilon_t$$

for $N = 5$, $N = 25$, $N = 100$ *points of the grid and* 1000 *time periods. Estimate with OLS*

$$\begin{aligned} y_t &= \mu + \rho_1^1 y_{t-1} + \rho_2^1 y_{t-2} + \beta^1 y_{t-1}^3 + \varepsilon_t. \\ y_t^2 &= \mu + \rho_1^2 y_{t-1} + \rho_2^2 y_{t-2} + \beta^2 y_{t-1}^3 + \varepsilon_t. \\ y_t^3 &= \mu + \rho_1^3 y_{t-1} + \rho_2^3 y_{t-2} + \beta^3 y_{t-1}^3 + \varepsilon_t. \end{aligned}$$

Repeat the simulation and estimation 100 times for randomized y_0*. Compare the estimation results also to a continuous simulation of the process.*

How to simulate a Markov chain

1. If we want to simulate a Markov chain with transition matrix P, we need to start with some initial state $s_0 = i$.
2. We calculate $\Pi = (\pi_{i,j}) = \left(\sum_{k=1}^{j} p_{i,k}\right)$.
3. We start with $t = 0$.
4. Then we draw a uniformly $(0,1)$ distributed variable u_t.
5. and find j such that $\pi_{s_t,j-1} < u_t \leq \pi_{s_t,j}$. This is state $s_{t+1} = j$.
6. We repeat steps 4 and 5 until $t = T$.

Alternative Simulation Procedure

1. In case the Markov chain approximates a continuous data generating process, e.g. an AR-1,
2. then we can alternatively simulate the DGP itself and later replace the realisations by the mean of the bin in which they fall.
3. This means, that we generate first a series $\hat{x}_t, t = 1...T$ from the true DGP.
4. Then we find j, such that $y_j < \hat{x}_t \leq y_{j+1}$.
5. And obtain the series of simulated values as $x_t = z_j$.

A stochastic growth model

- We extend the capital accumulation model to a situation, in which $\ln z$ follows a stationary AR-1 process, with autocorrelation $\rho = 0.8$ and variance $\sigma = 0.5$. (See MATLAB code)

The consumption-savings model

Exercise 9 and 10

Exercise (9)

Extend the consumption model to a situation, in which income y is composed of a log-normal stochastic part x, which follows a stationary AR-1 process, with autocorrelation ρ and a fixed part τ. Note that you need to set $(1+r)<\beta$ because asset holdings drift to infinity otherwise, Moreover, assume households can only borrow up to τ/r (the amount they can credibly promise to repay). Assume log-utility in this and the following exercises.

Exercise (10)

A Bewley Model of Money. *Assume that $\tau = r = 0$, i.e. households cannot borrow and the asset they save in bears no interest (money). Simulate an agent over T=1000 periods of time and calculate the average asset holding of the agent. Do so for various levels of long-run uncertainty and persistence. Plot the results!*

Literature

Primary Reading:

- Heer, B. and A. Maussner (2009), "Dynamic General Equilibrium Modelling", 2nd edition, Springer, Berlin. (Ch. 12)

Secondary reading:

- Adda, J. and R. Cooper (2004): "Dynamic Economics", MIT Press, Cambridge.
- Ljungqvist, L. und T. Sargent (2004): "Recursive Macroeconomic Theory", MIT press, Cambridge.
- Stockey, N.L. and Lucas, R.E. with E.C. Prescott (1989): "Recursive Methods in Economic Dynamics", Chapters 4 and 9, Havard University Press, Cambridge..
- Tauchen, G. (1986): "Finite state Markov-chain approximation to univariate and vector autoregressions", Economic Letters, 20, 177-81.

Part 3

Further Numerical Tools

Programme of this Part

1. Discussing the Exercises
2. Further numerical tools to solve SDP
   2.1 Policy Function Iteration
   2.2 Interpolation
   2.3 Multigrid Algorithms
   2.4 Projection methods

Policy Function Iteration

- While the Value Function Iteration is an intuitive solution algorithm that directly stems from the proof of existence of a solution to the Bellman equation, it is at the same time relatively slow.
- In the worst case, each iteration step reduces the distance between the true solution and the current guess of the value function only by factor β.
- In other words the discount factor β equals rate of convergence of the algorithm.
- For this reason we will discuss algorithms to speed up the solution of the Bellman equation in the following.

Policy Function Iteration

- The first algorithm that we want to go through is **Policy Function Iteration**, aka Howard's Improvement Algorithm.
- While **Value Function Iteration** assumes that the **policy function is applied once**, Policy Function Iteration assumes that the policy function is applied forever:
- For Value Function Iteration, we have

$$\begin{aligned} h_n(s) &= \arg\max u(s, s') + \beta E V_n(s') \\ V_{n+1}(s) &= u(s, h_n(s)) + \beta E V_n(h_n(s)) \end{aligned}$$

- For Policy Function Iteration, we define

$$\begin{aligned} h_n(s) &= \arg\max u(s, s') + \beta E V_n(s') \\ V_{n+1}(s) &= \sum_{t=0}^{\infty} \beta^t u(s_t, h_n(s_t)) \\ s_t &= h_n(s_{t-1}) \end{aligned}$$

Policy Function Iteration

- It is handy to put this into matrix notation.
- First stack the states into a vector, such that $\{s_1, \dots, s_N\}$ are all possible states the value function is defined on.
- Note that not necessarily all states can be reached with certainty by the agent's choice.
- For example, let the state space be productivity $a \in \{a_1, \dots, a_M\}$ (chosen by nature) and capital $k \in \{k_1, \dots, k_H\}$ (chosen by the agent), then $N = HM$ and

$$s_1 = (a_1, k_1), s_2 = (a_2, k_1), \dots, s_{M+1} = (a_1, k_2), \dots, s_N = (a_M, k_H).$$

Policy Function Iteration

- Let $\mathbf{v} = (v_i)_{i=1\ldots N} = [v(s_i)]_{i=1\ldots N}$be the value function writen as a vector that contains the value for any state s_i.
- Let $\mathbf{u} = (u_{i,j})_{\substack{i=1\ldots N \\ j=1\ldots M}} = \left[u\left(s_i, s_j'\right)\right]_{\substack{i=1\ldots N \\ j=1\ldots M}}$ denote the contemporaneous utility obtained by the transition from state s_i to s_j (let it be chosen or stochastic).
- Let $H = \left(h_{i,j}\right)_{\substack{i=1\ldots N \\ j=1\ldots N}}$ denote the stochastic policy function. That is as a transition matrix, which displays the probability that $h(s_i) = s_j$.
- Let T denote the operator defined by the Bellman equation.

Policy Function Iteration

- V satisfies $V = TV$.
- Using the notation from before, we know that

$$\mathbf{v} = H\left(\mathbf{u} + \beta\mathbf{v}\right)$$

where $H\mathbf{u}$ is the vector of instantaneous utility obtained under H in all possible states

Policy Function Iteration

- Putting these consideration into an algorithm, we obtain

$$\mathbf{v}_n = H_n \left(\mathbf{u} + \beta \mathbf{v}_n\right)$$

- and we can solve for v_n given H_n by inverting the relationship

$$\mathbf{v}_n = \left(I - \beta H_n\right)^{-1} H_n \mathbf{u}$$

- The next step is to find the policy function H_{n+1} **that defines the optimal policy given $\mathbf{v}_n$.**
- This means, we search for H_{n+1}, such that

$$\begin{aligned} H_{n+1}\left(\mathbf{u} + \beta \mathbf{v}_n\right) &= T\mathbf{v}_n. \\ H_{n+1}\mathbf{u} + \left(\beta H_{n+1} - I\right)\mathbf{v}_n &= \left(T - I\right)\mathbf{v}_n \end{aligned}$$

Policy Function Iteration: Interpretation

- We can understand policy function iteration as a version of Newton's algorithm to find a zero of $(T - I)$.
- Replacing $H_{n+1}\mathbf{u}$ by $(I_{MN} - \beta H_{n+1})\,\mathbf{v}_{n+1}$, we obtain

$$(I - \beta H_{n+1})\,\mathbf{v}_{n+1} - (I - \beta H_{n+1})\,\mathbf{v}_n = (T - I)\,\mathbf{v}_n$$

- This can be expressed as

$$\mathbf{v}_{n+1} = \mathbf{v}_n + (I - \beta H_{n+1})^{-1}\,(T - I)\,\mathbf{v}_n$$

- and $(\beta H_{n+1} - I)^{-1}$ can be regarded as the gradient of $(T - I)$.
- (Newton's algorithm: solve $G(z) = 0$, $z_{n+1} = z_n - G'(z_n)\,G(z_n)$)

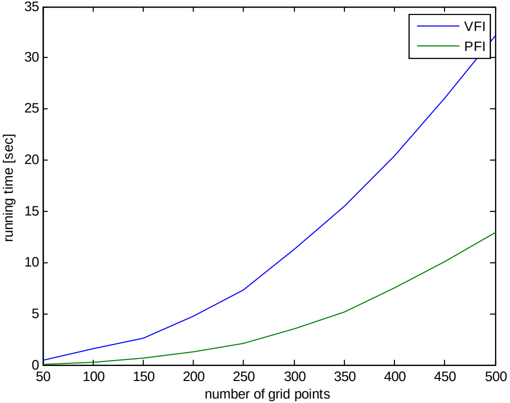
Policy Function Iteration: caveats

- Policy Function Iteration becomes time consuming for larger grid sizes as $L = (I - \beta H_{n+1})$ has to be inverted.
- There are some algorithms to speed this up.
- Inversion of a matrix can be parallelized.
- Alternatively: Apply βH_n just a number of times, making use of

$$(I - \beta H_{n+1})^{-1} = \lim_{S \to \infty} \sum_{s=0}^{S} \beta^s H_{n+1}^s$$

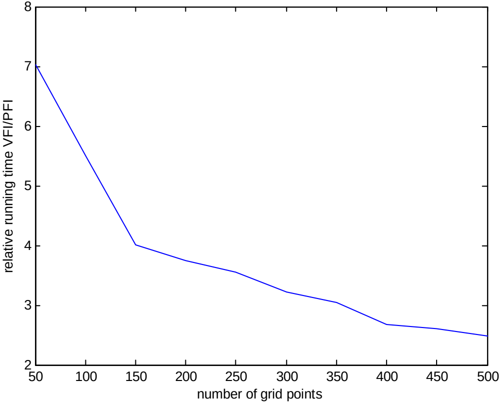
  and approximating this for a small S.

Policy Function Iteration: Comparison of running times



Both algorithms exhibit polynomial behavior in the number of points!

Policy Function Iteration: Relative running times



The larger the number of gridpoints, the smaller the advantage of the PFI!

Policy Function Iteration: Exercise

Exercise (11 in class)

Write a MATLAB code that solves the stochastic growth problem using PFI! Compare running times to VFI for a $N \times N$ *grid for* $N = 10, 50, 150, 250$ *points.*

Exercise (12)

Write a MATLAB code that solves the stochastic consumption-savings problem using PFI! Compare running times to VFI for a $N \times N$ *grid for* $N = 10, 50, 150, 250$ *points.*

Interpolation

- Since the computation costs increase exponentially in the number of points it may be a good alternative to interpolate points between grid points.
- Let x_i, $i = 1...N$ be points at which we know $f_i = f(x_i)$. We want to approximate the function f for off-grid points.
- We'll go through 3 methods: least-squares approximation, linear-interpolation, spline interpolation.

Least squares interpolation

- One method to approximate a function is to estimate the coefficients ψ of a polynomial expression

$$\hat{f}(x) = \sum_{j=1}^{n} \psi_j c_j(x)$$

  where $c_i(x)$ are known baseline functions such as $c_j(x) = x^j$.
- Better than ordinary polynomials are usually Chebyshev polynomials of which the baseline functions are

$$c_j(x) = \cos(j \ \arccos x)$$

- These are orthogonal in function space, and the regressor matrices
- **In practice these methods do not perform very well: Fluctuation, overshooting, ...**
- **The approximation does not necessarily equal the function at** x_i
- Advantage: little Information needs to be stored!

Linear Interpolation

- A computationally easy alternative is to fit a piecewise linear function.
- For each point x at which we want to evaluate our approximation we find the next smaller point about which we have information $x_i < x \leq x_{i+1}$.
- Then we calculate

$$\hat{f}(x) = f_i + \frac{f_{i+1} - f_i}{x_{i+1} - x_i}(x - x_i).$$

- The formula is easily extended to higher order cases.

Linear Interpolation

- It is implemented in MATLAB as default for INTERP1, INTERP2, INTERP3, INTERPN.
- FHAT=INTERP1(XI,FI,X)
- FHAT=INTERP2(XIJ,YIJ,FIJ,X,Y), x,y equal dimensional matrices defining the values of x and y in $f(x, y)$
- MATLAB can also extrapolate using linear interpolation (FHAT=INTERP1(XI,FI,X,LINEAR,EXTRAPVAL))

Spline Interpolation

- While the linear interpolation methos leads to an interpolated function $\hat{f}$ that is continuous, this function typically is non smooth.
- This non-differentiability may translate into non-differentiable policy functions.
- Spline interpolation evades this issue. It approximates the function **locally** by a cubic polynomial

$$\begin{aligned}\hat{f}_i(x) &= f_{i-1} + a_i(x - x_{i-1}) + b_i(x - x_{i-1})^2 + c_i(x - x_{i-1})^3, \\ &\quad x \in [x_{i-1}, x_i], i = 2, ..., N \\ \hat{f}_1(x_1) &= f_1\end{aligned}$$

Spline Interpolation

- This gives $3N-3$ parameters to be determined.
- Imposing continuity on the function and its first two derivatives yields a system of $3N-5$ non-linear equations constraining the choice of a, b, c

$$\begin{aligned} f_{i+1}(x_i) &= f_i(x_i), i = 1, ...N-1 \\ f'_{i+1}(x_i) &= f'_i(x_i), i = 2, ..., N-1 \\ f''_{i+1}(x_i) &= f''_i(x_i), i = 2, ..., N-1 \end{aligned}$$

-

$$\begin{aligned} f_i &= f_{i-1} + a_i(x_i - x_{i-1}) + b_i(x_i - x_{i-1})^2 + c_i(x_i - x_{i-1})^3 \\ a_{i+1} &= a_i + 2b_i(x_i - x_{i-1}) + 3c_i(x_i - x_{i-1})^2 \\ b_{i+1} &= b_i + 3c_i(x_i - x_{i-1}) \end{aligned}$$

Spline Interpolation

- $$\begin{aligned} a_i &= \frac{f_i - f_{i-1}}{(x_i - x_{i-1})} - b_i \left(x_i - x_{i-1}\right) - c_i \left(x_i - x_{i-1}\right)^2 \\ a_{i+1} &= a_i + 2b_i \left(x_i - x_{i-1}\right) + 3c_i \left(x_i - x_{i-1}\right)^2 \\ c_i &= \frac{b_{i+1} - b_i}{3\left(x_i - x_{i-1}\right)} \end{aligned}$$

- $$\begin{aligned} a_i &= \frac{f_i - f_{i-1}}{(x_i - x_{i-1})} - \left(\frac{2}{3} b_i + b_{i+1}\right)\left(x_i - x_{i-1}\right) \\ a_{i+1} &= a_i + \left(b_{i+1} + b_i\right)\left(x_i - x_{i-1}\right) \\ c_i &= \frac{b_{i+1} - b_i}{3\left(x_i - x_{i-1}\right)} \end{aligned}$$

Spline Interpolation

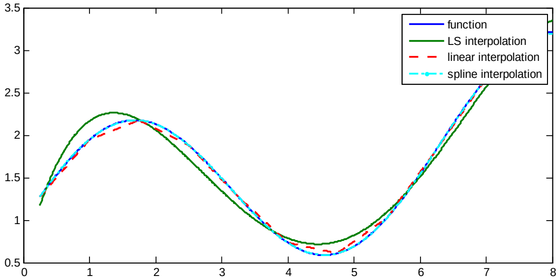
Finally we add $f''(x_1) = f''(x_N) = 0$. and obtain

$$
\begin{aligned}
a_i &= \frac{f_i - f_{i-1}}{(x_i - x_{i-1})} - \left(\frac{2}{3}b_i + b_{i+1}\right)(x_i - x_{i-1}), \;\; i = 2, ..., N \\
a_{i+1} - a_i &= (b_{i+1} + b_i)(x_i - x_{i-1}), \;\; i = 2, ..., N-1 \\
c_i &= \frac{b_{i+1} - b_i}{3(x_i - x_{i-1})}, \;\; i = 2, ...N-1 \\
c_N &= -\frac{b_N}{3(x_N - x_{N-1})} \\
b_2 &= 0
\end{aligned}
$$

Spline Interpolation

While spline interpolation is very accurate, it is very time consuming if N is large.

Comparison of interpolation methods



- Spline interpolation gives best results.
- Linear interpolation is not bad either, yet, derivatives are constant between base points..
- With insufficient degrees of freedom, LS interpolation performs badly.

Off-grid search

- We can use interpolation methods to approximate dynamic programming problems.
- So far we considered approximations to the continuous state-space problem

$$V(s) = \max_{y \in \Gamma(s)} u\left(s, s', \xi\right) + \beta V\left(s'\right).$$

by using a fine grid $\left\{s_1, \ldots, s_N\right\}$ for s and then solving by iterating over the approximated, discretized problem.

$$\mathbf{V}^n = \max\left\{\mathbf{U} + \iota_N \mathbf{V}^{n-1}\right\}.$$

Off-grid search

Alternatively, we can define

$$\hat{V}^n(s) = \text{Interpolation}\left(\left\{V_i^n, s_i\right\}_{i=1\ldots N}, s\right)$$

and update the base points, allowing for policies outside the grid $\left\{s_i\right\}_{i=1\ldots N}$.

$$V_i^{n+1} = \max_{y\in\Gamma(s_i)} u\left(s_i, s', \xi\right) + \beta\hat{V}^n\left(s'\right)$$

Off-grid search

Example

Exercise (13)

Use the non-stochastic growth model with $\delta = 1$.

1. *Define a 10 point grid for k.*
2. *Use spline interpolation to solve for the value function, starting with $V = 0$ as initial guess, save running times.*
3. *Compare to the true value function at the grid points and note average absolute difference.*
4. *Run a sequence of on-grid value function iterations with $N = 10,\ 20,\ 40,\ 80,\ 160,\ 320$ points. Compare average absolute difference to the true model and running times with the off-grid search code.*

Chow and Tsitsiklis' (1991) Multigrid VFI

- We have seen that **computation time for Value Function Iteration** as well as **Policy Function Iteration** grows **polynomially** in the number of grid points. (This can also been shown formally using complexity theory, see Rust (1996))
- Conversely this means that for small problems we can calculate the solution relatively quickly.

Chow and Tsitsiklis' (1991) Multigrid VFI

- Chow and Tsitsiklis (1991) suggest an algorithm that rests on this insight:
  1. First solve the SDP on a sparse grid of points by VFI: This yields $\hat{V}_0^*$
  2. Increase the number of grid points in each dimension by factor 2.
  3. Obtain an initial guess $\hat{V}_1^0$ for the value function on the new grid by interpolation from the solution on the coarser grid.
  4. Perform value function iteration to obtain $\hat{V}_1^*$ for the enlarged grid.
  5. Repeat steps (2) to (4) until the grid is fine enough.
- Ideally the critical value for termination of the VFI decreases by factor 2 in each grid iteration.

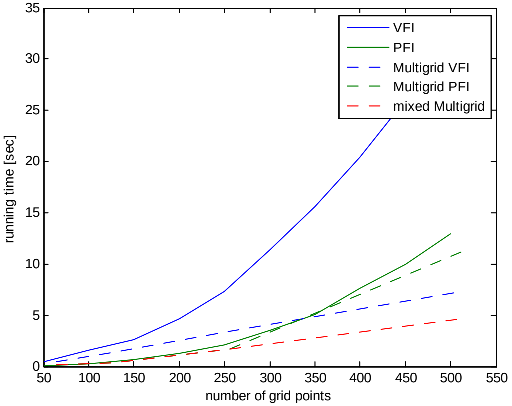
Chow and Tsitsiklis' (1991) Multigrid VFI

- Chow and Tsitsiklis (1991) show that this algorithm almost reaches the efficiency bound (in terms of worst case complexity) for algorithms to solve SDP.
- In practical applications the algorithm is **linear** in the number of gridpoints.
- This does not hold true if VFI is to be replaced by PFI, because complexity in the latter case stems from inverting the Policy matrix.

Chow and Tsitsiklis' (1991) Multigrid VFI

- For practical purposes a mixed algorithm outperforms:
  - Use a multigrid PFI until the total number of gridpoints is about 500.
  - Then switch to VFI.
- One has to trade off grid generation time (e.g. Tauchen) against multigrid time saving.
- A further practical advantage of the multigrid algorithm comes in dealing with outside loops: endogenous contracts, market clearing conditions, stationary equilibria etc.
- The following graphics compares computing times for the stochastic growth model:

Comparison of running times



VFI-Multigrid is linear, PFI-Multigrid is not!

Multigrid VFI: Code

Switch to MATLAB, discuss code!

Multigrid VFI: Exercises

Exercise (14a in class)

Program a multigrid VFI algorithm to solve the stochastic income, consumption-savings problem using the simplified Tauchen procedure to represent the AR-1 income process.

Exercise (14b)

Extend the stochastic growth model such that utility is obtained from an aggregate of current and lagged consumption. Assume $u\left(\bar{C}_t\right) = \ln \bar{C}_t$ *and* $\bar{C}_t = \left(\alpha c_t^{-1} + (1-\alpha)\, c_{t-1}^{-1}\right)^{-1}$. *Use a multigrid algorithm to solve the model. Simulate the model! What effect does the change in preferences have on the persistence of consumption? Compare the results for* $\alpha = 1, \alpha = 0.8, \alpha = 0.5$. *How can you interpret the model?*

Projection Methods

- All the before mentioned algorithms rely on the calculation of the value function to obtain the policy function, which is in most applications the object of interest.
- Projection methods directly solve for the policy function without solving for the value function by invoking the Euler equation.

Euler equation

- Consider a SDP

$$V(s) = \max_{\substack{c \in \Gamma(s) \\ s' = f(s,c)}} u(c) + \beta E V(s')$$

- The first order condition is

$$u'(c) + \beta E V'(s') f'_c(s,c) = 0$$

- Optimality (envelope theorem) implies

$$V'(s) = \beta E V'(s') f'_s(s,c)$$

Euler equation

- Therefore first order condition is

$$u'(c)\,\frac{f_s'(s,c)}{f_c'(s,c)} = -V'(s)$$

- Plugging this back in yields the **Euler equation**

$$u'(c) = \beta E\left(u'(c')\, f_s'(s',c')\right)$$

- The optimal policy trades off marginal utility today and discounted expected marginal utility tomorrow, taking into account the relative marginal effect of c and s on the future state s'.

Projection methods

- Projection methods exploit the **Euler equation**

$$u'(c) - \beta E\left(u'(c') f_s'(s', c')\right) = 0$$

- Define the policy function $c = h(s)$, then we can rewrite the Euler equation as

$$F_h(s) := u'(h(s)) \\ - \beta E\left(u'(h(f(s, h(s)))) f_s'(f(s, h(s)), h(f(s, h(s))))\right) = 0$$

Projection methods

- Projection methods parametrize h, for example by a Chebyshev polynomial.
$$\hat{h}(s) = \sum_{i=1}^{n} \psi_i c_i(x_i)$$
and we need to determine ψ_i, $i = 1, ..., n$.
- Ideally the base functions should look alike the policy function.
- We cannot expect $F_{\hat{h}}(s) = 0$ for all s.
- Therefore we minimize $||F_{\hat{h}}||$ for some appropriate metric $||\cdot||$.
- The metric is from the class $||F|| = \int_A F(a)\, g(a)\, da$, where g is some weighting function

Least square metric

- One possible metric is the least square metric.
- This leads to the minimization problem

$$\min_{\psi} \int \left[F_{\hat{h}}\left(s\right)\right]^2 ds$$

- The integral has to be solved numerically

Collocation method

- An alternative metric uses the mass point function as weighting.
- This function takes value 1 if $x = x_i$ for a pre-specified set of points.
- If one uses n points, the ψ_i become exactly identified and therefore the collocation method forces $F_{\hat{h}}$ to be exactly zero at x_i.
- Ideally one chooses x_i as the roots of the base functions, i.e. for Chebyshev polynomials $x_i = \cos\left(\frac{\pi}{2i}\right)$.
- So that we obtain a system of n equations $F_{\hat{h}}\left(x_i\right) = 0$ which is to be solved for ψ.

Exercise

Exercise (15)

Derive the Euler equation for the consumption model with $u = \frac{c^{1-\gamma}-1}{1-\gamma}$, *fluctuating income and a constant interest rate* R. *Reformulate the Euler equation such that you obtain an expression of the form* $E_t\left(g\left(c_t, c_{t+1}\right)\right) = 0$. *Define* $\varepsilon_{t+1} := g\left(c_t, c_{t+1}\right)$, *what can you tell about the correlation of* ε_{t+1} *and variables that are perfectly observed at time* t?

Literature

Primary Reading:

- Heer, B. and A. Maussner (2009), "Dynamic General Equilibrium Modelling", 2nd edition, Springer, Berlin. (Ch. 11)
- Adda, J. and R. Cooper (2004): "Dynamic Economics", MIT Press, Cambridge (Ch. 3)..

Secondary reading:

- Chow, C.S. and J.N. Tsitsiklis (1991): "An optimal multigrid algorithm for continuous state discrete time stochastic control", IEEE Transactions on Automatic Control, 36(8), 898–914.
- Ljungqvist, L. und T. Sargent (2004): "Recursive Macroeconomic Theory", MIT press, Cambridge.
- Stockey, N.L. and Lucas, R.E. with E.C. Prescott (1989): "Recursive Methods in Economic Dynamics", Chapters 4 and 9, HUP, Cambridge.
- Sundaram R. K. (1996): "A first course in Optimization Theory", Chapters 11 and 12, CUP, Cambridge.
- Tauchen, G. (1986): "Finite state Markov-chain approximation to univariate and vector autoregressions", Economic Letters, 20, 177-81

Part 4

Estimation

Programme of this week

1. Discussing the Exercises
2. Non-Simulation Based Estimation
   - 2.1 Maximum Likelihood
   - 2.2 Generalized Methods of Moments
3. Simulation Based Estimation
   - 3.1 Bootstrap
   - 3.2 Indirect Inference and method of simulated moments

Estimation

- In general we do not directly observe the parameters of the model, but for any numerical analysis we need to fix parameters to certain values.
- In doing so we want to obtain a maximal fit of our model with the data.
- At the same time, we want to evaluate how likely it is that the data which we observe is actually generated by our model.
- First we'll go through some more standard estimation techniques before we discuss particular techniques to estimate numerical models.

Maximum Likelihood

- The most efficient - though not always feasible - *classical* estimation method is maximum likelihood estimation.
- It is a prerequisite for this method that we exactly know the distribution of shocks that drive our model.
- However this is typically not a problem, since we needed to assume some distribution anyway in order to solve the model.

Maximum Likelihood

- Let $\varepsilon_t, t = 1...T$ be a sequence of i.i.d. shocks, X_t be the sequence of states, and Y_t be the sequence of policy variables generated from our model.
- Then we can view the model as some data generating process Φ that generates a sequence

$$[X,\ Y]_t = \Phi\left(X_{t-1}, \varepsilon_t|\theta\right)$$

- Maximum likelihood estimation rests on the inversion of Φ ("backing out the shocks") for given model parameters θ.

Conditional Maximum Likelihood: algorithm

- Typically we start with some X_0, ε_0 and then recursively obtain an error estimate

$$\hat{\varepsilon}_t(\theta) = \Phi^{-1}\left(X_t, Y_t | X_{t-1}, \theta, \hat{\varepsilon}_{t-1}\right)$$

- Then we obtain the likelihood function as

$$L\left(\{\hat{\varepsilon}_t(\theta)\}_{t=1\dots T}\right) = \prod_{t=1}^{T} f\left(\hat{\varepsilon}_t(\theta)\right)$$

- We then maximize L by choosing θ

Maximum Likelihood: 2 issues

- The first problem that we may encounter, is that F is not invertible. Therefore there is no one-to-one mapping that maps shocks to states.
- In other words, under our model one sequence of states of nature may be a result of various sequences of shocks.
- In this case the two series of shocks are said to be **observationally equivalent** and **the model is not identified**.

Maximum Likelihood: 2 issues

- The other issue is less of a problem from a scientific point of view, though of terrible importance to the researcher: the **zero-probability problem**
- It may be that F is invertible, but under all admissible model parameters the sequence $(X \ Y)_t$ cannot be generated by arbitrary shocks.
- In this case the model is simply **wrong.** E.g. assume you suppose $x_t = \frac{1}{1+\theta \exp(\varepsilon_t)}$. If you observe some $x \notin (0, 1)$ then the model must be wrong and is to be rejected.
- Zero probability is a degenerated case of model rejction due to low probability of observing the data even under the most favourable choice of θ.

Maximum Likelihood: An example

- Consider flipping a coin that has probability θ to show heads and tails otherwise.
- Suppose we observe N_1 times heads out of N draws.
- Then the Likelihood function is

$$\begin{aligned} L(x,\theta) &= \theta^{N_1}(1-\theta)^{N-N_1} \\ \ln L &= N_1 \ln\theta + (N-N_1)\ln(1-\theta). \end{aligned}$$

- Maximization leads to

$$\begin{aligned} N_1\frac{1}{\theta} - (N-N_1)\frac{1}{1-\theta} &= 0 \\ \theta^* &= \frac{N_1}{N} \end{aligned}$$

Maximum Likelihood: Estimation of the capital accumulation model

- Suppose we want to estimate the production function parameter α, the shock process parameters ρ and σ from our growth model. We assume a log-normal AR-1 process for productivity.
- Fix $\beta = 0.95$, $\delta = 0.05$ and $\mu = 0$.
- We use German NAICS data on consumption and investment in logs, detrended and demeaned. We assume $Y = C + I$ and use this variable for inference, i.e. we back-out the shocks that exactly reproduce the output series.
- We fix k_0 to the steady state value at mean productivity. And drop the first 10 quarters of the residual to minimize the influence of the initial choice of k.
- Switch to MATLAB!

Maximum Likelihood: Model comparison

- By itself ML Estimation can only perform a weak test of the qualities of the model.
- It only tells us, whether the observed data can be generated by the model and some sequence of shocks even if this sequence has an arbitrary small likelihood.
- Standard approaches to model comparison such as LR tests only work in nested models, e.g. we can test a risk aversion of one against alternative CRRA formulations.
- Test approaches such as Vuong (1989) tests (similar to AIC) are better suited, they allow to compare our theoretical model to an atheoretical descriptions of the data, such as (V)ARs.

GMM Estimation

- A downside of the ML Estimation is usually that we need to know the distribution of the shocks ε to the economic system.
- Generalized method of moments estimation evades this problem by relying only on information on a number of moments generated by ε and our model.
- This also has computational advantages, because "backing out the shocks" can be quite computational intense.
- Note that GMM only helps if we can derive the moment conditions from **theory,** because moment conditions need to hold exactly.

GMM Estimation

- Suppose M_θ is the set of all moments generated by our model under the parameter choice θ. Let $\mu(\theta)$ be a subset of these moments that is used for the estimation.
- Let $\bar{\theta}$ be the true parameters that nature has chosen.
- If our model is correct, then

$$E\left(\hat{\mu}\left(\bar{\theta}\right) - \mu\left(\bar{\theta}\right)\right) = 0$$

  where $\hat{\mu}$ is the sample analogue to μ.
- Define $g(x_t, \theta)$ as the moment generating function and $F(x_t, \theta) = g(x_t, \theta) - \mu(\theta)$.

GMM Estimation

- We can use this observation to construct an estimation procedure.
- Suppose $E\left(F\left(x,\theta\right)\right)=0$ has a unique solution, then this is $\bar{\theta}$.
- If we have as many moments as parameters (**exact identification**), we obtain
$$\hat{\theta}=\mu^{-1}\left(\hat{\mu}\right).$$
- Typically we will choose more moments than parameters, since
  1. this increases efficiency if the moments are informative (i.e. not constant in θ, not collinear to other moments)
  2. this allows us to construct a test of our model

GMM Estimation

- If we have more moments than parameters (**overidentification**), we minimize the quadratic expression

$$\hat{\theta} = \arg\min \left(F_T\left(x,\theta\right)\right) W \left(F_T\left(x,\theta\right)\right)'$$

- Where F_T are the sample means of F and W is a weighting matrix that ideally takes into account the variance-covariance structure of F_T and gives large weight to those moments that are estimated with high precision.

GMM Estimation

- Let f_T be the average derivative of F_T in the sample, i.e $f_T(\theta) = \frac{1}{T}\sum_{t=1}^{T} \frac{\partial F(x_t,\theta)}{\partial \theta}$.
- Then the estimator

$$\hat{\theta} = \arg\min F_T(\theta) W F_T(\theta)'$$

is asymptotically normally distributed and has a variance of

$$\left(f_T' W f_T\right)^{-1} \left(f_T' W V_T W f_T\right) \left(f_T' W f_T\right)^{-1}$$

- where V_T is the **long run** variance of $F_T(\theta)$.

$$V_T = E\left[\left(\sum_{t=1}^{T} F(x_t, \theta)\right)^2\right]$$

GMM Estimation

- This means the optimal choice for W is V_T^{-1}, which confirms our initial guess.
- Calculating V_T^{-1} is complicated in most cases, because this involves θ.
- If moment conditions are formulated such that g does not depend on θ, a single bootstrap procedure provides us with an estimate of V_T.
- Otherwise it is **one** possible algorithm to obtain an estimate of V to start with any initial pds weighting matrix W and then update W iteratively using some appropriate method to estimate V.
- There is a number of alternatives to do so, see Matyas (1999) for example. Examples are HAC, Newey-Whited or Bootstrap estimation procedures.

GMM Estimation: Example 1

- Consider again the coin-flipping example.
- Say heads has value one and tails zero.
- Then the expected value of x is θ.
- Moreover the variance of x is $var\left(x\right)=\theta\left(1-\theta\right)$

(G)MM Estimation: Example 1

- However the second moment is not informative. Whenever the first moment condition is met, then also the second moment condition holds.
- Therefor the MM estimator is

$$\arg\min\left(\frac{N_1}{N}-\theta\right)^2$$

- FOC

$$\begin{aligned} -2\left(\frac{N_1}{N}-\theta\right) &= 0 \\ \theta^* &= \frac{N_1}{N} \end{aligned}$$

- Which is actually the ML estimate

GMM Estimation: Example 2

- Suppose, we have the linear regression model

$$y_t = X_t\beta + u_t$$

- Where $\dim(X_t) = p$, $\text{cov}(X_t, u_t) \neq 0$ but there is some Z_t with $\dim(Z_t) = q$ and $rk\left(E\left(Z'Z\right)\right) = q$ and $\text{cov}(Z_t, u_t) = 0$.
- If further X and Z are not orthogonal, we can formulate as moment condition

$$\begin{aligned} E\left(Z_t'\left(y_t - X_t\beta\right)\right) &= 0 \\ F\left(\beta\right) &= \frac{1}{T}\sum Z_t'\left(y_t - X_t\beta\right) = \frac{1}{T}Z'\left(y - X\beta\right) \end{aligned}$$

GMM Estimation: Example 2

- Then with $\text{cov}(u) = \sigma^2 I$

$$
\begin{aligned}
V &= \frac{1}{T} Z' (y - X\beta) \left(Z' (y - X\beta) \right)' \\
&= \frac{1}{T} Z' u u' Z \\
&= \frac{1}{T} \sum \left(Z_t' u_t u_t Z_t \right) \\
&= \sigma^2 T^{-1} Z' Z
\end{aligned}
$$

- And $f(\beta) = Z' X$

GMM Estimation: Example 2

- Therefore the GMM estimator becomes

$$\begin{aligned}\hat{\beta} &= \arg\min \frac{1}{T} Z' \left(y - X\beta\right) \left(Z'Z\right)^{-1} \left(y' - \beta' X'\right) Z \\ \hat{\beta} &= \left(X'Z \left(Z'Z\right)^{-1} Z'X\right)^{-1} X'Z \left(Z'Z\right)^{-1} Z'y\end{aligned}$$

- which is the IV estimator with more instrumental than model variables.

GMM Estimation: Testing overidentifying restrictions

- To perform a test of the model in the GMM framework, we test how close

$$Q_T = F_T\left(\hat{\theta}\right) W F_T\left(\hat{\theta}\right)'$$

actually gets to zero.

- Under the condition that $W = V_T^{-1}$ Hansen (1982) shows that

$$J_T = TQ_T$$

is χ_{q-p} distributed, where q is the number of linear independent moment restrictions and p is the number of parameters.

Simulation based estimation

- Both Maximum likelihood estimation and the generalized method of moments have serious drawbacks in the estimation of dynamic models.
- For GMM we have to rest on analytical results,
- ML-Estimation quickly becomes numerically infeasible, involving the solution of multiple integrals and so on.
- Simulation based estimation methods are the “natural” way to think about estimation of numerical models.

Simulation based estimation

- We will focus on the method of simulated moments and its extension to indirect inference
- These methods can be thought of as extensions of GMM
- or as "calibration with confidence bounds"

Bootstrap estimates of the variance covariance structure

- Suppose we observe a data sample $x_t, t = 1...T$.
- We are interested in some expression of the form $\gamma(X) \in R^n$ and its distribution, for example sample moments.
- Not always can we determine the covariance structure of γ from analytical grounds without knowing the distribution of X.
- Bootstrapping can help in this situation.

Bootstrap estimates of the variance covariance structure

- We resample N artificial samples $\tilde{X}_i$ from X by drawing T times from X with replacement.
- This gives a sample of $i = 1...N$ realizations γ_i of which we can easily determine the variance structure in sample.
- For $T \to \infty$ and $N \to \infty$ the bootstrap sample variance converges to the theoretical variance.
- For example in GMM Estimation this is useful to estimate the variance matrix of moments.

Exercise

Exercise (16)

Write a programme that

1. *Simulates the VAR*

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix}_t = B \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}_{t-1} + \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \end{pmatrix}_t, \; B = \begin{bmatrix} 0.7 & 0.1 \\ -0.2 & 0.6 \end{bmatrix}$$

   for T*=200 periods where* $\varepsilon \tilde{} N(0, I)$.
2. *Estimates B with OLS*
3. *Draws N=1000 Bootstrap samples*
4. *Re-estimates B with OLS for each of the samples.*
5. *Compares the standard deviation from the estimate of the bootstrap samples and the standard error derived from asymptotic theory of the estimate from the simulated sample.*

Method of simulated moments

- The method of simulated moments (MSM) builds upon the idea of moment-matching in GMM estimation.
- However, what is compared in the MSM case are moment estimates from Monte-Carlo simulations of our economic model and observed data moments.
- Let $y_t, t = 1 \ldots T$ be a sequence of observations which we suppose to be generated by the economic model Φ.
- Let $g\left(y_t, y_{t-1}, \ldots, y_{t-l}\right) \in R^k$ be the moment generating function that exploits information from up to l lags of y.

Method of simulated moments

- We draw R independent (Monte Carlo) simulations of our model to generate simulated datasets $y_t^r(\theta)$, $t = 1 \ldots T$, $r = 1 \ldots R$.
- The length of T is the same as in the sample of observations.
- Typically we need to define a starting state for the model y_0.
  - If possible this is taken as a random draw from the ergodic distribution of y under $\Phi(\theta)$.
  - If this is not feasible, we can choose a sensible starting value $y_{-\tau}$ and then simulate the model for some additional initial periods τ which we then drop.
- We estimate the moments implied by the model as

$$\bar{\mu}(\theta) := R^{-1}(T - l)^{-1} \sum_{r=1}^{R} \sum_{t=l}^{T} g\left(y_t^r(\theta), y_{t-1}^r(\theta), \ldots, y_{t-l}^r(\theta)\right)$$

Method of simulated moments

- The MSM estimate then is

$$\hat{\theta}_{MSM} := \arg\min \left(\hat{\mu} - \bar{\mu}\left(\theta\right)\right) W \left(\hat{\mu} - \bar{\mu}\left(\theta\right)\right)'$$

- where W is some pds matrix.
- Ideally we use $W = V^{-1}$ where V is the $\text{var}(\hat{\mu} - \bar{\mu}\left(\theta\right))$, approximated by $\text{var}(\hat{\mu})$.

Method of simulated moments

- In this case Duffie and Singleton (1993) show that

$$T^{\frac{1}{2}}\left(\hat{\theta}_{MSM}-\theta_0\right) \xrightarrow{d} N\left(0,\left(1+R^{-1}\right)\left(DV^{-1}D\right)^{-1}\right)$$

where

$$D = E\frac{\partial\bar{\mu}\left(\theta_0\right)}{\partial\theta}$$

- This means that the MSM estimator is only by a factor $\left(1+R^{-1}\right)$ less efficient than the GMM estimator.
- Already for relative small values of R, e.g. $R = 5$ or $R = 10$, the loss in precision is mild.

Choice of Moments

- From the inspection of the formula of the asymptotic variance of the MSM estimator

$$avar\left(\hat{\theta}_{MSM}\right) = \left(1 + R^{-1}\right)\left(DV^{-1}D\right)^{-1}$$

  we can see that the precision of the estimate will crucially depend on the sensitivity of moments to model parameters.
- If moments only weakly depend on model parameters, then D is small and the estimate will be imprecise.
- In small samples the choice of overly many moments can lead to a substantial efficiency loss.

Choice of Moments

- This means **from an econometric point of view** we should choose **moments** which are very informative as they **depend strongly on model parameters** and are also estimated with little uncertainty (small V).
- However, if we loosen the econometric point of view somewhat, then we can also understand the choice of the set of moments as a particular point of view from which we analyze the model.
- This second view tends more towards the practice in calibration, where we choose a **set of moments which** we feel **is essential** for the model to match **in the light of its economic content**.

Moments and indirect inference

- The MSM estimation can be extended to the use of regression coefficients as moments.
- For this purpose, we define an auxiliary mispecified model Φ^* to describe the data in reduced form.
- This model has a parameter vector β with $q \geq p$ parameters.
- We can use the parameter estimates $\hat{\beta}$ as moments and use as weighting matrix their variance covariance structure implied by the estimation procedure used to estimate β assuming Φ^* were actually correct.

Indirect inference

- If we estimate β by quasi maximum likelihood, then this is the **indirect inference** proposed by Monfort et al. (1993), aka **simulated minimum distance** estimator.
- Galant and Tauchen (1996) suggest an alternative indirect inference procedure
  - that minimizes a qudratic form of the score functions (derivatives of the log-likelihood) of the auxiliary model,
  - using the simulated data and the parameter estimate from the auxiliary model and the observed data.
  - The optimal weighting matrix is the variance of the scores.
- Smith (1993) suggests a third alternative: simulated quasi maximum likelihood, in which
  - the likelihood of the observed data under the auxiliary model is maximized
  - imposing the ML parameter estimates from the auxiliary model under the simulated data

Efficient method of moments

- A particular advantage of the indirect inference formulation is that the auxiliary model can be chosen data dependent.
- If we increase the complexity of the auxiliary model with T (increasing q, using a semiparametric auxiliary model) the Galant and Tauchen Estimator approaches asymptotically the efficiency bound. (i.e. is equivalent to the infeasible ML of Φ)
- This estimator is termed Efficient method of moments (EMM).
- However, it may behave poorly in small samples.

MSM: Estimation of the capital accumulation model

- Suppose we want to estimate the production function parameter α, risk aversion γ and the shock process parameters ρ and σ from our growth model. We assume a log-normal AR-1 process for productivity.
- We fix $\beta = 0.95$, $\delta = 0.05$ and $\mu = 0$.
- We use German quarterly NAICS data on consumption and investment in logs, detrended and demeaned. We assume $Y = C + I$.
- We fix k_0 to the steady state value at mean productivity and drop the first 10 years of the simulation to minimize the influence of the initial choice of k.
- We use as moments the standard deviations σ_Y, σ_I, and the first order autocorrelations ρ_Y, ρ_I.
- Switch to MATLAB!

Literature

Primary Reading:

- Adda, J. and R. Cooper (2004): "Dynamic Economics", MIT Press, Cambridge (Ch. 4)..

Secondary reading:

- Matyas, L. (1999): "Generalized Method of Moments Estimation", CUP, Cambridge, Ch. 1-3. & 10.
- Gourieroux, C., A. Monfort, and E. Renault (1993): "Indirect inference", Journal of Applied Econometrics, 8, p. 85-118.
- Smith, T. (1993): "Estimating nonlinear time-series models using simulated vector autoregressions", Journal of Applied Econometrics, 8, p. 63-84.
- Vuong, Q. H. (1989): "Likelihood Ratio Tests for Model Selection and Non-Nested Hypotheses," Econometrica, 57, 307 - 333.

Part 5

General equilibrium,
Semi-strategic interactions,
Dynamic Games

Programme of this Part

1. Discussion of Exercises
2. General Equilibrium
3. Incorporating zero profit conditions

General equilibrium

(Decentralization, recursive equilibrium)

- Suppose we model an economy in a dynamic setup, in which some or all actions of economic agents are described as infinite horizon dynamic programming problems.
- If all agents are alike (or of a finite number of types), then we can describe the economy by an aggregate state vector S.
- The agents take the evolution $H(S)$ as given (general equilibrium).
- Prices depend on the aggregate state $P(S)$.

Recursive equilibrium

Definition

A **recursive equilibrium** is comprised of

- price functions $P(S)$,
- individual policy functions $h(s, S)$ that solve the dynamic programming problem,
- a law of motion $H(S)$ for the aggregate economy,

such that

- all agents optimize taking P and H as given
- markets clear
- and the aggregate law of motion and the individual policy functions are consistent, i.e. $H(S) = h(S, S)$.

Example: Capital accumulation

- Suppose we extend the capital accumulation model, such that households and firms are separated.
- Capital is held by households which rent it out to the firm sector that is owned by the households.
- The households planning problem is

$$V(k, A, K) = \max_{k'} u\left(\left(r(K) + (1-\delta)\right)k - k' + \Pi\right) + \beta E V\left(k', \underbrace{A', K'}_{=H(A,K)}\right)$$

- Competitive capital and product markets are characterized by

$$\begin{aligned} r(K)\,k &= \alpha z k^{\alpha} \\ \Pi &= (1-\alpha)\, z k^{\alpha} \end{aligned}$$

Solution strategies

1. Often the recursive equilibrium can be shown to be the solution of a central planner problem, which can be solved using standard techniques.
2. Alternatively, we begin with a guess of H and P and iterate until convergence. Note, however, that the dimensionality of the problem increases and we need an outside loop to find H and P. Finally, convergence is not guaranteed.

Zero Profit Conditions

- Consider a situation in which two agents interact and one agent offers a contract $r(s)$ to the other agent.
- The agent offering the contract is bound by a zero profit condition, but has to take the actions of the other agents into account.
- Generically we can write the situation as

$$V_r(s) = \max_{s' \in \Gamma(s)} u\left(s, s' | r(s)\right) + \beta E V_r\left(s'\right)$$

with ϕ being the associated policy and r being implicitly defined by

$$\pi\left(r(s), \phi_r(s)\right) = 0$$

Zero Profit conditions

- We need to find an optimal policy function (strategies) of the decision maker given r.
- And r has to fulfill the zero-profit condition, given the policy function ϕ.
- Therefore one possible way to solve the problem is to assume a function r_0, then find the optimal ϕ_1 for this guess, then update r to r_1 and so on.
- This does not necessarily converge.

Exercise 17

Debt contracts

Exercise (17)

Extend the consumption model such that the agent can decide to default on his obligations. In the case of default, the agent surrenders all his wealth and future income except for the fixed income τ income to his creditors. There is perfect competition among creditors who themselves borrow at rate R. Assume that in case of bankruptcy, in addition to the loss in the debt claim, the creditor also loses some fixed amount κ to acquire the debtors income.

Strategic interactions

- So far we studied situations with little or no interaction of economic entities.
- Real economic situations often involve strategic interactions.
- These become quickly infeasible to study in terms of DP.
- However, we want to lay out the principles of such analysis.

Strategic interactions

- To be able to model a strategic interaction in a dynamic context, we first need to adapt the problem to a multi-decsionmaker setup.
- Let s be the total state vector whereas s_i is the state vector under control by decision maker i. Let S and S_i be the corresponding choice sets.
- Suppose $p_i(s) = s_i'$ defines the perceived policy function for individual i - perceived by its competitors.
- We can state the individual optimization problem as

$$V_i(s_i, s_{-i}) = \max_{s_i' \in S_i} u\left(s_i, s_{-i}, s_i', p_{-i}(s)\right) + \beta EV_i\left(s_i', p_{-i}(s)\right)$$

Strategic interactions

Definition

A **Markov perfect equilibrium**, now is a set of perceived policy functions p_i, Value Functions V_i and actual Policy Functions P_i such that

1. given p_{-i}, the Value Function V_i and the corresponding Policy Functions P_i solve the agents optimization problem.
2. $P_i = p_i$ for all i.

Remark: The recursive formulation incorporates the Markov property: Policies only condition on the state and not on the entire history of the game. Perfectness is included by defining policies for each possible state, regardless whether this state is reached in equilibrium or not.

Strategic interactions: Solution strategy

- While the formulation looks straightforward, finding a solution becomes quickly infeasible. Moreover it is often not clear in advance whether a solution exists.
- Often the following algorithm succeeds in finding the equilibrium:
  1. Start with some initial guess for the policy functions of p_i^0.
  2. Find the optimal policy p_i^1 for each i under the assumption that $p_{-i} = p_{-i}^0$.
  3. Iterate until convergence.

Strategic interactions: Solution strategy

- While in practice such algorithm may succeed, one can easily see that it does not necessarily do so by considering the static analogue to this algorithm.
- The idea behind this algorithm is to define $T(s_i) = R_i(R_{-i}(s_i))$ and construct a sequence $s_i^n = T^n(s_i)$, where R_i is the best response function.
- If T is a contraction, the algorithm succeeds and finds the unique equilibrium.
- However, there may be more than one equilibrium, or the sequence may actually diverge: Suppose

$$R_i(s_{-i}) = a + bs_{-i}$$

If $b > 1$ then the sequence will diverge.

Part 6

Stationary heterogeneous agent economies

Heterogeneous agent economies

- A further complication arises if economies consist of heterogeneous agents, like households differing in their wealth levels and having non-linear consumption functions. Our definition of a recursive equilibrium relied on the assumption of a single (type) of consumer or firm.
- However, if one takes the statement of heterogeneous agents seriously, an exact analysis becomes infeasible.
- Suppose there is a continuum of consumers, all indexed by their wealth level W. The distribution of wealth across consumers shall be characterized by density $\mu(W)$.

Heterogeneous agent economies: The Problem

- In such economy, aggregate demand will in general depend on the distribution of wealth $\mu(W)$.
- Therefore also prices depend on the distribution of wealth $p = p(\cdot, \mu(W))$.
- This, however, means that the whole distribution μ is a state of the economy.
- Yet, μ is an infinite dimensional object and hence not numerical tractable.

Heterogeneous agent economies: Stationary economies

- One way to circumwent these problems is to look at economies without aggregate fluctuations.
- In such economies, prices are constants.
- Although they depend on μ, this dependence does not matter.
- Next we consider such models.

A particular savings problem

- Suppose income can take m levels $\bar{y}_i$.
- For an individual household y_t evolves according to a Markov Chain with transition probability Π.
- [a special case is two states: unemployed $y_0 = b$ and employed $y_1 = 1$].
- The household can hold assets in amounts given by a grid $A = \{a_1, \ldots, a_n\}$, where $a_{j-1} < a_j$, $0 \in A$.

A particular savings problem

- The household chooses c_t (and thus a_t) in order to maximize

$$\begin{aligned} & E_0 \sum_{t=0}^{\infty} \beta^t u(c_t) \\ s.t.\ c_t + a_{t+1} &= (1+r)\, a_t + y_t;\ a_{t+1} \in A \end{aligned}$$

- We can treat prices as time-fixed, because we just want to look at the steady-state distribution of assets, when the cross-sectional distribution of s and a is its ergodic distribution.
- We assume that $\beta(1+r) < 1$, which is no real restriction since otherwise no stationary equilibrium exists.

A particular savings problem

- The Bellman equation is

$$V\left(a_h, \bar{y}_i\right) = \max_{a' \in A} u\left(\left(1+r\right) a_h + \bar{y}_i - a'\right) + \beta \sum_{j=1}^{m} \Pi\left(i,j\right) v\left(a', \bar{y}_j\right)$$

- We can treat prices as time-fixed, because we just want to look at the steady-state distribution of assets, when the cross-sectional distribution of s and a is its ergodic distribution.
- The dynamic programming problem leads to policy functions $a' = g\left(a, s | r\right)$.

Recursive rule for the distribution

- Now denote $\lambda_t(a, s) = \Pr(a_t = a, s_t = s)$.
- The exogenous Markov chain Π and g induce a law of motion

$$\lambda_{t+1}(a', s') = \sum_{a \in A} \sum_{s \in S} \lambda_t(a, s) \Pi(s'|s) I_{g_r(a,s)=a'}.$$

- If we have stochastic elements (e.g. stochastic depreciation) influencing a' then $I \in \{0, 1\}$ has to be replaced by some probability function taking also values $\in (0, 1)$.
- Writing things a little simpler

$$\lambda_{t+1}(a', s') = \sum_{s \in S} \sum_{\{a | a' = g_{r,w}(a,s)\}} \lambda_t(a, s) \Pi(s'|s).$$

Stationary distribution

The law of motion

$$\lambda_{t+1}\left(a', s'\right) = \sum_{s \in S} \sum_{\{a | a' = g_{r,w}(a,s)\}} \lambda_t\left(a, s\right) \Pi\left(s' | s\right). \tag{L}$$

defines a mapping on λ. A stationary distribution $\bar{\lambda}$ is a fixed point of this mapping.

Stationary distribution

An algorithm

- Given the discrete structure, possible states form a matrix that we can vectorize.
- Let $i = 1 \dots m$, be the index of states of nature (income) and $h = 1 \dots n$ be the index of possible asset choices.
- Now define $j = (i-1)\, m + h$ a joint index. Then

$$\hat{\lambda}_{t+1}(j) = \vartheta(j)'_r \, \hat{\lambda}_t$$

where $\vartheta(j)'$ is the probability vector with element η defining the probability to end in state η :

$$\vartheta_r(j) = \left(\Pi\left(ceil(j/m) | ceil(\eta/m) \right) I_{\hat{g}(\eta) = j - ceil(j/m)} \right)_{\eta = 1 \dots nm}$$

Stationary distribution

- This means we can write

$$\hat{\lambda}_{t+1} = \vartheta_r \hat{\lambda}_t$$

and the stationary distribution $\bar{\lambda}$ corresponds to the unit-eigenvector of ϑ_r.

- There are several ways to solve for this eigenvector.
- One fast procedure is to take any column of

$$\lim_{T \to \infty} \vartheta_r^T$$

- which can be implemented quickly using

$$\vartheta_r^{(n)} = \vartheta_r^{(n-1)} \vartheta_r^{(n-1)}.$$

Stationary distribution

- We can read the matrix ϑ as inducing a Markov chain on λ.
- There is two interpretations of the stationary (ergodic) distribution $\bar{\lambda}$:
  1. $\bar{\lambda}\,(j)$ is the cross-sectional probability of a household being in state j
  2. $\bar{\lambda}\,(j)$ is the fraction of time a given agent spends in state j.

General equilibrium

- We have not yet derived a general equilibrium (as promised), but still treated r as fixed.
- In our example a is credit traded between households.
- For the loan market to clear, we must have

$$\sum_j \lambda(j)\,\alpha\left(\hat{g}(j|r)\right) = 0$$

  where $\alpha(j)$ is the asset corresponding to the j-th state and $\hat{g}$ gives the index of the optimal policy.
- $-\phi = a_1 < 0$ defines a borrowing limit.

Huggett's model

Definition

Given ϕ, a **stationary equilibrium** is an interest rate r, a policy function $g\left(a, s|r\right)$, and a stationary distribution $\lambda\left(a, s|r\right)$ such that

1. The policy functions solves the household's optimization problem given r.
2. The stationary distribution is induced by Π and $g\left(a, s|r\right)$.
3. The loan market clears

$$\sum_{a,s} \lambda\left(a, s\right) g\left(a, s|r\right) = 0$$

Huggett's model

Computation of Equilibrium

The equilibrium computation is relatively straightforward. Any given r induces an optimal policy $g^*(a, s|r)$, which itself induces a stationary distribution $\lambda^*(a, s|r)$. Now define the excess demand function for credit Φ :

$$\Phi(r) := \sum_{a,s} \lambda^*(a, s)\, g^*(a, s|r)$$

of which we need to find a zero. One way to do so is to iteratively, producing a sequence of $r^{(j)}$ where

$$r^{(j+1)} = r^{(j)} - sign\left(\Phi\left(r^{(j)}\right)\right)\left|\frac{r^{(j)} - r^{(j-1)}}{2}\right|$$

setting $r^{(0)} = 0,\ r^{(1)} = \frac{1-\beta}{\beta} - \varepsilon$.

Huggett's model

Exercise

Exercise (18a)

Extend the optimal savings problem to calculate the equilibrium interest rate. Assume income can take two states $y_1 = \tau = 0.5$ (unemployed) $y_2 = 1$ (employed). The transition probabilities are given by

$$\Pi = \begin{pmatrix} .925 & .075 \\ .500 & .500 \end{pmatrix}.$$

Assume agents can borrow up to the natural borrowing limit τ/r. Use a 200-point grid with equidistant points for asset holdings between $-\tau/r$ and $\frac{1}{r}$. Use the multigrid algorithm only on the inside loop.

Huggett's model

Exercise

Exercise (18b)

Same as above, but use a multigrid algorithm also on the outside loop, i.e. solve for $r^*_{(1)}$ *for a sparse grid and use* $r^*_{(1)}$ *as a starting guess for the finer grid for assets obtaining* $r^*_{(2)}$ *and so on.*

Huggett's model

Some remarks

Remark

You may want to solve the household problem for a less fine grid than the grid you want to use for the distribution of asset holdings. Heer and Mausner (2009, pp.340-358) discuss some algorithms.

Remark

This obviously is the case if you allow for off-grid choices.

Remark

If you allow only for ongrid choices, some of the reasons given in Heer and Mausner (2009, pp.340-358) why to use finer grids for the distribution than for the savings problem are obsolete with (a) multigrid algorithms to solve for the value function (b) 64-bit machines that can easily handle large matrices ϑ_r and quickly calculate $\bar{\lambda}_r$.

Aiyagari's model

Capital investment

- Aiyagari (1994) analyses a model where households accumulate capital and rent it out to firms.
- Households are endowed with s_t units of effective labor in each period.
- The endowment of the household follows a Markov Chain with transition probability matrix Π.
- The household budget constraint is given by

$$c_t + k_{t+1} = ws_t + \left(1 + \underbrace{uc - \delta}_{=:r}\right) k_t.$$

Aiyagari's model

Capital investment

- For any pair if given prices (r, w) the household has an optimal capital accumulation plan that induces a stationary distribution $\bar{\lambda}(k, s)$.
- Aggregate capital is given by

$$K = \sum_{k,s} \bar{\lambda}(k, s)\, g(k, s).$$

- Aggregate labor is given by

$$\bar{N} = \pi^{\infty\prime} \bar{s}$$

and is independent of K.

Aiyagari's model

Algorithm

- Equilibrium wages and rental rates fulfill

$$\begin{aligned} F_N\left(K, \bar{N}\right) &= w \\ F_K\left(K, \bar{N}\right) &= uc = r + \delta \end{aligned}$$

- This means, we can write the model in terms of aggregate capital: $(w, r) = \phi\left(K\right)$.
- Now, we obtain the optimal household plan which is a function of K; $g^*\left(k, s|\phi\left(K\right)\right)$.
- This induces a stationary distribution $\bar{\lambda}\left(k, s|\phi\left(K\right)\right)$.
- and implies an aggregate stock of capital

$$K^*\left(K\right) = \sum_{k,s} \bar{\lambda}\left(k, s|\phi\left(K\right)\right) g^*\left(k, s|\phi\left(K\right)\right)$$

- A stationary equilibrium can be calculated by finding a fixed point of K^*

Aiyagari's model

Definition

Given ϕ, a **stationary equilibrium** is an interest rate r and a wage rate w, an average stock of capital K, a policy function $g(k, s|r, w)$, and a stationary distribution $\lambda(k, s|r, w)$ such that

1. Equilibrium wages and rental rates fulfill

$$\begin{aligned} F_N(K, \bar{N}) &= w \\ F_K(K, \bar{N}) &= r + \delta \end{aligned}$$

2. The policy functions solves the household's optimization problem given r, w.
3. The stationary distribution is induced by Π and $g(k, s|r, w)$.
4. The average stock of capital is implied by the households' decision

$$K^*(K) = \sum_{k,s} \lambda(k, s|r, w)\, g(k, s|r, w)$$

Aiyagari's model

Remark

Finding the equilibrium coincides to finding a fixed point on K^*.

Remark

Simple iterative application of K^* *converges slowly, if at all. The algorithm*

$$K^{(n)} = \phi K^* \left(K^{(n-1)}\right) + (1-\phi) K^{(n-1)}$$

for example for $\phi = 0.5$ *works better.*

Aiyagari's model

Exercise

Exercise (19a)

Calculate the equilibrium interest rate in the Aiyagari model. Assume income can take two states $s_1 = 0$ *(unemployed)* $y_2 = 1$ *(employed). The transition probabilities are given by*

$$\Pi = \begin{pmatrix} .925 & .075 \\ .500 & .500 \end{pmatrix}.$$

Let $\delta = .1$, $\beta = 0.95, \alpha = 0.3$ $, \gamma = .3$ *and*

$$F(K, N) = \left(K^{\alpha} N^{1-\alpha}\right)^{1-\gamma}.$$

Use a 400-point grid with equidistant points for asset holdings between 0 *and* 3000*. Use the representative agent economy stock of capital as a starting guess for* K*. Use a multigrid algorithm on the outside loop.*

Aiyagari's model

Exercise

Exercise (19b)

Introduce a lump-sum redistributive labor income tax $\tau = .2$ to the model. Assume lump-sum payments adjust in order to balance the government's budget.

Literature

Primary Reading:

- Heer, B. and A. Maussner (2009), "Dynamic General Equilibrium Modelling", 2nd edition, Springer, Berlin. (Ch. 7)

Secondary reading:

- Ljungqvist, L. und T. Sargent (2004): "Recursive Macroeconomic Theory", MIT press, Cambridge.
- Stockey, N.L. and Lucas, R.E. with E.C. Prescott (1989): "Recursive Methods in Economic Dynamics", Chapters 4 and 9, Havard University Press, Cambridge.
- Huggett, R. G. (1993): "the risk-free rate in heterogeneous-agent incomplete-insurance economies", JEDC, 360-99.
- Aiyagari, R. S. (1994): "Uninsured Idiosyncratic Risk and Aggregate Saving", QJE, 659-84.

Part 7

Heterogeneous agent economies with aggregate fluctuations

Heterogeneous agent economies with aggregate fluctuations

The Problem

- If there is aggregate uncertainty the distribution of heterogeneity, e.g. wealth will change over time.
- Now agents need to take these movements of the distribution into account, i.e. the distribution becomes an aggregate state variable.
- Yet this is infinite dimensional as an object - except for special cases of parametric families.

Heterogeneous agent economies:

The Krusell-Smith algorithm

- Krusell and Smith (1997, 1998) suggest an algorithm to solve the issue by solving an approximation to this economy.
- The idea of this approximation may be summarized as "rational adaptive expectations".
- We approximate the problem by assuming that economic agents forecast prices by using only a finite set of moments m of μ and some parametric model $p = g_p\left(s, m, \beta_p\right)$.
- Similarly households are assumed to use a parametric function $g_m\left(s, m, \beta_m\right)$ to forecast future moments m' of μ.

Heterogeneous agent economies:

An example

- Consider Huggett's model, but with aggregate income risk: The only price is the real rate.
- An households income consists of an aggregate part that affects all households equally and an idiosyncratic part that only affects this household

$$
\begin{aligned}
y_{it} &= \exp\left(x_{it} + Z_t\right) + \tau \\
x_{it} &= \rho x_{it-1} + \sigma_x \varepsilon_{it}^x \\
Z_t &= \rho Z_t + \sigma_z \varepsilon_t^z
\end{aligned}
$$

Heterogeneous agent economies:

An example

- Now the perfectly rational household's problem would be

$$\begin{aligned} V(x, a, Z, F) &= \max_{a \in A} u\left\{a\left[1+r(Z, F)\right]+y(x, Z)-a'\right\} \\ &\quad +\beta E V\left(x', a', Z', G(Z, F)\right) \end{aligned}$$

  where F is the distribution of wealth and $G(Z, F)$ is the perceived aggregate law of motion for F.

- It is infeasible to find a solution to this problem.

Heterogeneous agent economies:

An example

Krussell and Smith propose to assume that the household instead first solves for $\bar{V}$ in

$$\begin{aligned}\bar{V}\left(x,a,Z,m\right) &= \max_{a\in A} u\left\{a+y\left(x,Z\right)-a'\left[1+g_r\left(Z,m\right)\right]^{-1}\right\} \\ &\quad +\beta E\bar{V}\left(x',a',Z',g_m\left(Z,m\right)\right)\end{aligned}$$

where m are summary statistics (moments) of F, g_m is a perceived law of motion for m and g_r is a perceived pricing rule.

Heterogeneous agent economies:

An example

In each actual period of time, the household then acts such that the policy is

$$
\begin{aligned}
h(x, a, Z, m, r) \quad = \quad & \arg\max_{a \in A} u\left\{a + y(x, Z) - a' [1 + r]^{-1}\right\} \\
& + \beta E \bar{V}\left(x', a', Z', g_m(Z, m)\right)
\end{aligned}
$$

where m are summary statistics (moments) of F, g_m is a perceived law of motion for m and g_r is a perceived pricing rule.

Heterogeneous agent economies

Definition of equilibrium

Definition

A boundedly rational (limited information, or Krussell-Smith) equilibrium is: A limited information value function $\bar{V}$, perceived laws of motion g_m and pricing rules g_r and a policy function h, a sequence of distributions of wealth F, a law of motion $G(Z, F)$ for F and prices $r(Z, F)$such that

1. $\bar{V}$ solves the limited information dynamic program, given $g_{r,m}$
2. h solves the household problem given current period prices r and $\bar{V}$
3. G is implied by h
4. r is such that markets clear.
5. $g_{r,m}$ are best predictors for $r(Z, F)$ and the moments m' of $G(Z, F)$ within their parametric families.

Heterogeneous agent economies

Some remarks

Remark

The above definition yields an equilibrium for each parametric family of g. Usually one searches for parametric families and sets of moments, such that the prediction error is small (e.g. high R^2).

Remark

Practically we cannot compute G and r but can only obtain sequences of F, r for a given draw of a sequence Z and some starting values F_0.

Remark

If we use discretization methods to determine $\bar{V}$, note that we can easily define a sparser grid in deriving $\bar{V}$ than for the actual policy h.

Heterogeneous agent economies

Algorithm

1. [Optimize out all intra-period decisions]
2. Choose a set of moments m to characterize the distribution F.
3. Draw a sequence of $\{Z_t\}_{t=1...T}$, choose F_0 (e.g. the ergodic distribution without aggregate risk)
4. Choose a functional form for g (often log linear does well) and guess initial parameters β_0.
5. Solve for $\bar{V}(x, a, Z, m)$ and $h(x, a, Z, m, r)$.
6. Use h and $\{Z_t\}_{t=1...T}$ to generate a sequence of F_t and r_t ensuring market clearing in each period.
7. Update β using LS estimates on $\{m_t, r_t\}_{t=1...T}$, go back to (4.), iterate until convergence
8. Test the goodness of fit of the parametric family g.

Application: Fluctuations in employment risk

Exercise

Calculate the equilibrium interest rate in the Aiyagari model. Assume income can take two states $s_1 = 0$ *(unemployed)* $y_2 = 1$ *(employed). The transition probabilities are given by*

$$\Pi = \begin{pmatrix} .95-\eta & .05+\eta \\ .500 & .500 \end{pmatrix}.$$

Let $\delta = .1$, $\beta = 0.95, \alpha = 0.3$ $,\gamma = .3$ *and*

$$F(K,N) = \left(K^{\alpha} N^{1-\alpha}\right)^{1-\gamma}.$$

Application: Fluctuations in employment risk

Exercise

. . . Use a 400-point grid with equidistant points for asset holdings between 0 *and* 3000*. The aggregate stochastic state is* $\eta \in \{0, 0.05\}$ *and follows a Markov chain with transition probability matrix*

$$\begin{pmatrix} .8 & .2 \\ .2 & .8 \end{pmatrix}$$

Use the stationary distribution from exercise $(19a)$ *as a starting guess. Simulate over* $T = 1500$ *periods dropping the first 100. Use log-linear rules and only mean* $(\ln k)$ *as a moment of* F.