November 23rd 2009 - Lecture Eighteen

Econ205A Macro – 20091123 – Outline – Final Exam Discussion. Quiz. Introducing Stochastic Models. An Arrow Debreu Competitive Equilibrium in Static Stochastic Setting. The Neo Classical Growth Model, Competitive Equilibrium in a Stochastic Setting

Quiz Four
Define a recursive competitive equilibrium of the neo-classical growth model.

Answer – See the end of the previous lecture

Intro - Two more topics to discuss: Stochastic Models – Labour Search.
We are going to have another problem set with dynamic programming in stochastic models plus a question with labour search.

Final Exam Discussion

- ✓ "Problem Set 8 should be considered good practice for the exam."
- ✓ In reply to the question (paraphrased) "what's going to be on the test?" Gorry: I haven't been able to test you on anything after the Mid-Term. The important topics are continuous time models with taxes, growth, recursive problems. Although he hasn't written the final, it is fair to expect a question on recursive stuff.
- ✓ Wilily, refusing to tell us exactly what will be on the test and what will not, Gorry tells us that even the new topics we've learned are built on the old ones. So he can't exclude any topic from the final.
- ✓ He's likely to give us two questions, We'll be given a general problem outline and we have to set up and solve the model. Perhaps set up a Lagrangian and maximize it. Perhaps we'll be given a continuous time program. And perhaps we could be given a continuous time problem to set up and mostly solve.
- ✓ It will be comprehensive by most definitions of the term.
- ✓ Basically, the point of the class is to learn tools that are useful in solving problems. And we can expect to be asked to use those tools to solve problems.

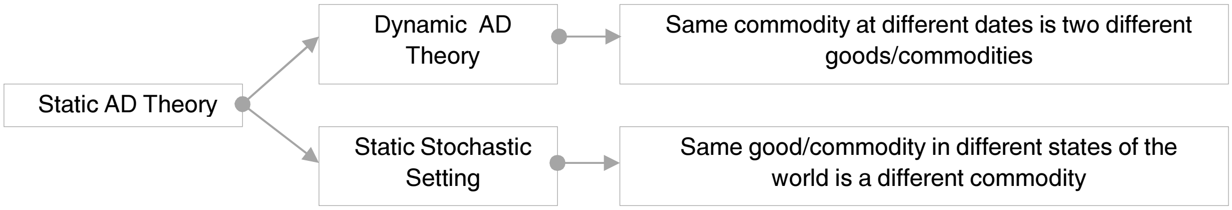
Topics we've talked about – growth model and the initial part of that was the saving's vs investment decision. Labor supply – we've examined labor leisure choices. Taxes – how taxes distort savings and labor supply decisions. More on growth – the standard neo-classical growth model, we have a steady state without long-term growth. How to get long-term growth: changing technology is one route. For homework we've had human capital endogenously generated. We've had more topics to think more deeply and complicatedly about these types of questions.

- ✓ You should be able to pass the final if you know the tools well. Thus, given a problem set-up, we'll need to prove that we can you are able to follow these steps: setting up a problem, taking first order conditions, much of the complicated math. Given the time constraints and stress we aren't expected to do all perfectly but should be able to start the steps to analyze these questions.

Introducing Stochastic Models

First Note - We'll assume risk preferences and learn why later in Micro
Dynamic Programming – remember that we found that dynamic programming reveals the same Euler Equation that the old fashioned discrete time model does. So why are we learning dynamic programming? a complicated method that give the same solution?

1. Dynamic programming provides a method to solve problems numerically. That's great.
2. A Dynamic programming environment can often times simplify problems in stochastic or random settings.



Lecture One – **Static Arrow Debreu**

- Lecture Two: **expanded to dynamic setting. Dynamic AD Theory**

The key innovation that allowed us to think about models over time was that we though of the same good at different dates as different goods (consumption, capital at time $t = \cdots$) – That meant the rules that apply to the static problem now apply to the problem with time.

Same commodity at different dates is two different goods/commodities

- **Extend AD to Static Stochastic Setting.**

Here you have some random variable that come up – not two time periods. If the economy's behavior depends on this random variable then we'll need a way to think about this. The key here is similar to the above problem, if you have different states of the world – the same good in different states of the world can be though of as a different good.

Same good/commodity in different states of the world is a different commodity

One issue – the number of variables that you're dealing with get very large very fast as you add numerous states and time periods. The decision tree might be imagined as a branch ever widening with possible paths – especially given infinite sequences.

Static Stochastic Setting

$I: consumers$

$J: goods \quad made\ by\ L\ producers, perhaps\ use\ inputs\ \&\ outputs$

$S: states\ of\ nature$

$\pi_s: probability\ of\ state\ s \in S$

$$\sum_{s=1}^{S} \Pi_s = 1 \qquad (required\ to\ be\ a\ probability)$$

$$\Pi_s \geq 0 \quad \forall S$$

$i \in I$

$\big(u_i(c_s, s),\ \ \omega_i(s)\big)$

Each consumer i is defined by a pair,
their utility ($u_i(c_i, s)$) which is a function of consumption and the state
& their endowment ($\omega_i(s)$) which is a function of the state.
c_i is potentially a vector, perhaps a scalar.

u_i: $\mathbb{R}_t^J \cdot \{1, \ldots . S\} \to \mathbb{R}$

u_i gives you a function from every consumption vector and in each state. It tells you how you feel about it (from the consumption vector) in every state. It gives you a real number

ω_i: $\{1,2, \ldots S\} \to \mathbb{R}_t^J$

The endowment says, in each state, what do you get?

In this model the timing is going to be that consumer agrees to contingent consumption in each date knowing the probabilities – you know everything up front. You know that whatever state comes up you know what your endowment and you know how you feel about utility in each date. So the idea is that we can just all agree what we're going to consume once the state is realized.

The timing is going be that consumers agree to contingent consumption in each date knowing the probabilities and their potential endowments. That allows us to think about an Arrow Debreu Competitive Equilibrium

■ An Arrow Debreu Competitive Equilibrium is a List

$\{c^*_{ijs}\},\{p^*_{js}\}\ s.t.$

· We're going to need to know everyone's **consumption** of every good, in every state.
· We'll need to characterize a list of consumption for each consumer (i), for each good (j), in each state (s) → c_{ijs}
· But no time – this is a one period model
· We also need **prices** – for each good (j), for each state (s).

· **Consumers are Solve the Problem:**
taking prices as given, each consumer ($\forall i$) wants to maximize their expected utility (doing this by summing over all states and multiply utility by the probability of each state):

1) Taking prices as given, $\{c_{ijs}\}$ solves ($\forall i$)

$$\max_{c_{ijs}} \sum_{s=1}^{S} \Pi_s\, u(c_{is}, s) \qquad c_{is} = (c_{i1s}, c_{i2s}, \dots, c_{iJs})$$

The Arrow Debreu Constraint:

$$s.t. \quad \sum_{j=1}^{J}\sum_{s=1}^{S} p^*_{js}\, c_{ijs} \;\le\; \sum_{j=1}^{J}\sum_{s=1}^{S} p^*_{js}\omega_{ij}(s)$$

· Subject to the Arrow Debreu Constraint. In general AD constraints say the total value of your consumption plan has to be less than or equal to the total value of your production or income over your life.
· This is a one period model. The total value of the consumption is going to be the value of your planned consumption in each state. It must be less than or equal to the value of your endowment in each state.

You can see how having states is kind of annoying in sequence problems.
If you were to add time you'd have four different subscripts.

2) Markets Clear

$$\sum_{i=1}^{I} c^i_{ijs} = \sum_{i=1}^{I} \omega_{ij}(s) \text{ Type equation here.}$$

· For all goods (j) and states (s), summing over I → the amount of consumption is equal to the total endowment.

In Static Stochastic Setting - one period ■
An **Arrow Debreu Competitive Equilibrium** is a List $\{c^*_{ijs}\},\{p^*_{js}\}\ s.t.$

1) Taking prices as given, $\{c_{ijs}\}$ solves ($\forall i$)

$$\max_{c_{ijs}} \sum_{s=1}^{S} \Pi_s\, u(c_{is}, s)$$

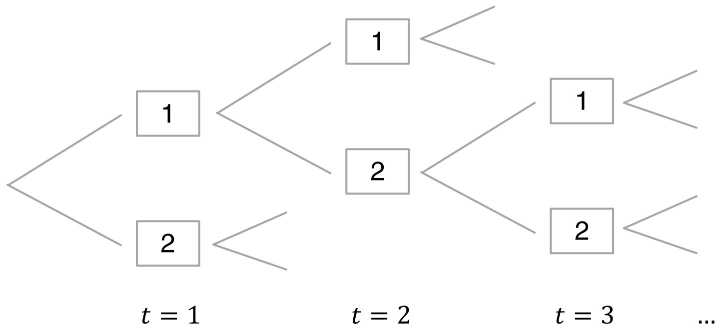
$$s.t. \quad \sum_{j=1}^{J}\sum_{s=1}^{S} p^*_{js}\, c_{ijs} \;\le\; \sum_{j=1}^{J}\sum_{s=1}^{S} p^*_{js}\omega_{ij}(s)$$

2) Markets Clear

$$\sum_{i=1}^{I} c^i_{ijs} = \sum_{i=1}^{I} \omega_{ij}(s)$$

The basic story of this model is there is some number of consumers. They each get different endowments depending on the state (sunny or raining…. Say they are all farmers that produce different crops that perform differently given different weather). They know the probability of all states tomorrow, we know the total output in each case. Before the state is realized we can trade/insure claims on each of the states using our total wealth. Implicitly, the AD framework assumes there are complete markets – this combined budget constraints allows you to trade your total wealth valued at these prices (which include risk) per consumption in any state. Thus even if a farmer only get paid given for one state in ten, they can trade the potential endowment to ensure they end up with some endowment no matter what state actually arrives.

Working States can be very complicated very quickly



This is just for one variable per date state combination. If you have any richness of goods – capital, labor, etc… it gets complicated.

Neo Classical Growth Model – Competitive Equilibrium in a Stochastic Setting

A Competitive Equilibrium is a list of sequences $\{y_{ts}^*\}, \{c_{ts}^*\}, \{x_{ts}^*\}, \{k_{ts}^*\}, \{h_{ts}^*\}, \{w_{ts}^*\}, \{r_{ts}^*\}, \{p_{ts}^*\}$
Since there are various states, the sequence will depend on the time period t & the state s

Consumer's Maximization
1) Taking prices as given, [consumers choose] $\{c_{ts}^*\}, \{x_{ts}^*\}, \{k_{ts}^*\}, \{h_{ts}^*\}$ [that] solve:

$$\max \sum_{t=0}^{\infty} \beta^t \sum_{s=1}^{S} \Pi_t(s) u(c_{ts}, \ 1 - h_{ts}, \ s)$$

$$s.t. \sum_{t=0}^{\infty} \sum_{s=1}^{S} p_{ts}^*(c_{ts} + x_{ts}) \le \sum_{t=0}^{\infty} \sum_{s=1}^{S} (w_{ts}^* h_{ts} + r_{ts}^* k_{ts})$$

$$k_{t+1,s'} = k_{ts}(1-\delta) + x_{ts} \qquad \forall s' \in S_{t+1}$$

Max expected utility, summed over all time, discounted, summed over the time-dependent probability of all possible states. The constraint is summing over all time and across all states. Price times consumption plus investment is the state contingent value of consumption & investment in each period. This must be less than or equal to the state contingent value of total earnings.
$*$ Prices have stars but c, x, h & k don't. Prices p_{ts}^*, w_{ts}^* & r_{ts}^* are already optimized (hence the star) – and we're optimizing over c, x, h & k.

2) Taking prices as given, [firms choose] $\{y_{ts}^*\}, \{k_{ts}^*\}, \{h_{ts}^*\}$ [that] solve

$$\max \sum_{t=0}^{\infty} \sum_{s=1}^{S} (p_{ts}^* y_{ts} - w_{ts}^* h_{ts} - r_{ts}^* k_{ts})$$

$$s.t. \quad y_{ts} = F(k_{ts}, h_{ts}, s)$$

3) Markets Clear

$$\forall \ t, s$$

$$y_{ts}^* = c_{ts}^* + x_{ts}^*$$

Two Points – given the standard sequence set-up, a random shock neoclassical growth model in competitive equilibrium is very complicated. The other point is the randomness has many applications – with an s added to consumption and production, this can be state dependent preferences, meaning you want to consume a lot or a little for a particular state. State specific production (weather). State contingent government policy. State contingent taxes.

The Next Step is trying to solve this type of model. It probably isn't that appetizing in the current set-up. –it would be better to set it up in a dynamic framework where all of the information that you needed to know was embedded in what the current period state is.

Instead of solving for all potential sequences of future states and outcomes, we can define one's current position with a smaller number of variables (say, the current state s, and the current capital stock k). If you can find an optimal decision for whatever current state and capital is – that's equivalent to solving for this whole tree.

In a lot of cases the dynamic programming set-up will be an easier way to solving these types of stochastic problems.

Mathematical Conditions to allow for this to work

We'll need some restrictions on shocks to ensure we can solve the problem.

When we did dynamic programming before (without randomness) we needed the restriction of having a variable that contained all the information required to make decisions about the future (the state variable in previous lecture's context of the work "state").

We need that state variable – plus restrictions on the random processes so that there they can encode enough information to make decisions about the future. Restrictions like – the expectation of the future shock (the distribution of the shock next period) is independent of what's happened in the past.

We also may need to include variables that encode information, say, the probability of future shock.

Markov Process Introduced

In general in order to maintain a recursive structure it's helpful to have a Markov Process. A stochastic process in which the value of the shock or state next period only depends on the state this period.