Topic 2
Solving Representative Agent General Equilibrium Models

Adam Hal Spencer

The University of Nottingham

Applied Computational Economics

Roadmap

Outline

- We've gotten the basic stuff out of the way.
- From here we go:
  - Representative agents in general equilibrium,
  - Heterogeneous agents, (idiosyncratic uncertainty).

Objective

- Last time: we solved a decision problem with a fixed interest rate.
- Let's endogenise it now!
- General equilibrium.

Equilibrium Concepts

- How do we define a dynamic general equilibrium?
- In a static context, we study general equilibrium with a finite number of goods.
- When we start talking about dynamics, we get into the realm of infinite-dimensional spaces...

Roadmap

Equilibrium Concepts

- Let's think of two different types of dynamic market equilibrium concepts.
  - Sequential markets equilibrium,
  - Recursive competitive equilibrium.
- The second is the most useful from a computational perspective.
- These types of decentralised equilibria are interesting when we start introducing distortions, (e.g. taxes).
- Can no longer just use the social planner's problem.
- Let's briefly contrast these equilibrium concepts.

Neoclassical General Equilibrium Model

- Consider the deterministic market version of our favourite problem (where the household owns the capital stock)
- Household's problem is

$$\max_{\{c_t, k_{t+1}\}_{t=0}^{\infty}} \sum_{t=0}^{\infty} \beta^t u(c_t)$$

subject to

$$k_{t+1} + c_t = r_t k_t + (1-\delta)k_t + \pi_t$$
$$c_t, k_{t+1} \geqslant 0$$

with k_0.

- Firm's problem:

$$\max_{\{k_t\}} \ \pi_t = k_t^{\alpha} - r_t k_t$$

Concept (1): Sequence Markets Equilibrium

- A sequential markets equilibrium is a set of
  - Prices $\{r_t^*\}_{t=0}^{\infty}$ (rental rate on capital),
  - Quantities $\{c_t^*, k_{t+1}^*\}_{t=0}^{\infty}$, such that

(1') Sequence $\{c_t^*, k_{t+1}^*\}_{t=0}^{\infty}$ solve the consumer's problem

$$\max_{\{c_t, k_{t+1}\}_{t=0}^{\infty}} \sum_{t=0}^{\infty} \beta^t u(c_t)$$

subject to their time t budget constraint

$$c_t + k_{t+1} = r_t^* k_t + (1-\delta)k_t + \pi_t$$

with k_0 taken as given.

- Notice that the price r_t^* is relative to time t consumption goods.

Concept (1): Sequence Markets Equilibrium

(2') Sequence $\{k_t^*\}_{t=0}^{\infty}$ solves the firm problem

$$\max_{\{k_t\}} k_t^{\alpha} - r_t^* k_t$$

$\forall t$. I.e. $r_t^* = \alpha(k_t^*)^{\alpha-1}$.

(3') Markets clear

$$c_t^* + k_{t+1}^* - (1-\delta)k_t^* = (k_t^*)^{\alpha}$$

$\forall t$.

Sequential Markets Equilibria

- Notice again that this method involves computing infinite sequences.
- Again, computers don't like that.
- Remember we had this same problem with the household's sequence problem?
- Our solution there was the recursive formulation.
- Can something similar save us here?

Concept (2) Recursive Competitive Equilibrium

- We can start by again thinking about the household's recursive formulation.
- There are complications though.
- Recall that we want to endogenise the interest rate

$$r_t = \alpha k_t^{\alpha-1}.$$

Concept (2) Recursive Competitive Equilibrium

- See when it comes to the household's savings decision, next period's interest rate matters

$$r_{t+1} = \alpha k_{t+1}^{\alpha-1}.$$

- In a representative agent context, the agent's savings choice can impact next period's interest rate.
- But it's meant to be a representative agent that takes prices as given.

Concept (2) Recursive Competitive Equilibrium

- Trick: "big-K, little-k".
- Include an additional state for the aggregate capital stock, big K.
- Household takes this as given when making their choices.
- Then make sure that the agent's choice, little k, coincides with the big K as an equilibrium condition.

Concept (2) Recursive Competitive Equilibrium

- The household's recursive formulation is given by

$$v(k, K) = \max_{\{c, k'\}} \ u(c) + \beta v(k', K')$$

subject to

$$\begin{aligned} c + k' &= R(K)k + (1 - \delta)k \\ K' &= G(K) \end{aligned}$$

where notice that the household now has two states — k and K.

- $R(K)$ is the rental rate on capital: determined at the aggregate level from production function K^{α}.

- The second constraint is the law of motion of aggregate capital, which the household takes as given.

Concept (2) Recursive Competitive Equilibrium

- k denotes the household's current capital stock.
- K denotes the aggregate capital stock.
- In equilibrium, they will be the same.
- But the household takes K as given when they make their decisions!
- So we can't allow them to internalise their choices' effect on K.

Concept (2) Recursive Competitive Equilibrium

- A recursive competitive equilibrium is a set of functions
  - Quantities $G(K)$, $g(k, K)$: the law of motion for aggregate capital and the household's policy function respectively.
  - Lifetime utility level $v(k, K)$.
  - Price $R(K)$, all such that

(1") Value function $v(k, K)$ solves the household's recursive formulation and $g(k, K)$ is the associated policy function.

(2") Prices are determined competitively

$$R(K) = \alpha K^{\alpha-1}$$

Concept (2) Recursive Competitive Equilibrium

(3") Consistency is satisfied

$$G(K) = g(K, K) \ \forall K$$

where notice that the requirement that the capital law of motion equal the household's policy function is an equilibrium condition.

- It's not something that we impose until after solving the household's problem with K given.
- This condition says that when the household is endowed with a level of capital equal to that of the aggregate, their behaviour is consistent with the aggregate law of motion.

Concept (2) Recursive Competitive Equilibrium

- Notice that the price $R(K)$ is a function as opposed to a price sequence like it was before.
- The same goes for the control variables: $c(k, K)$ and $k'(k, K)$.
- This is nice: we simplified the infinite-dimensional problem into something recursive and time-invariant that we can solve on a computer!

Roadmap

Randomness

- With randomness, we need to think about states of the world in each period.
- Say we think now about a production function of the form $A_t K_t^{\alpha}$.
- Where A_t follows a Markov process.
- Means that current probabilities are determined by most recent realisations, (A_{t-1} is the state for the stochastic process).
- $A_t \sim Q(A_t | A_{t-1})$.
- Say that the process for A_t is discretised into $|\Omega|$ elements (countably finite), which are invariant across time.
- I.e. $A_t(\omega)$ where $\omega \in \{\omega_1, \omega_2, ..., \omega_{|\Omega|}\}$.

Stochastic Recursive Competitive Equilibrium

- Household's problem will now be given by

$$v(k, K, A) = \max_{\{c, k'\}} \; u(c) + \beta \mathbb{E}_{A'|A}[V(k', K', A')]$$

subject to

$$\begin{aligned} c + k' &= R(K, A)k + (1 - \delta)k \\ K' &= G(K, A) \\ A' &\sim Q(A'|A) \end{aligned}$$

Stochastic Recursive Competitive Equilibrium

- A stochastic recursive competitive equilibrium is a set of functions
  - Quantities $G(K, A)$, $g(k, K, A)$: the law of motion for aggregate capital and the household's policy function respectively.
  - Lifetime utility level $v(k, K, A)$.
  - Price $R(K, A)$, all such that

(1") Value function $v(k, K, A)$ solves the household's recursive formulation and $g(k, K, A)$ is the associated policy function.

(2") Prices are determined competitively

$$R(K, A) = \alpha A K^{\alpha-1}$$

Stochastic Recursive Competitive Equilibrium

(3") Consistency is satisfied

$$G(K, A) = g(K, K, A) \ \forall K$$

- The state can indeed be changing between periods, (through fluctuating A).
- But again, the problem always looks the same in this recursive setup.

Roadmap

Algorithm

- Hopefully you have a decent intuition for the procedure by now.
- Say we are solving the deterministic model from the second section of these slides.

(1) Guess the law of motion for aggregate capital $G(K)$.

(2) Find the return $R(K)$.

(3) Solve the household's problem in the standard way with VFI to get $g(k, K)$.

(4) Update your guess of $G(K)$, (given that the individual and aggregate functions are meant to be "close").

Algorithm

- This process is effectively mapping from a guess of $G(K)$ to an update of it.
- This functional is not necessarily a contraction.
- Iterations on the computer might not converge, if it does, the differences may be highly non-monotonic.
- Update policy functions slowly!

Algorithm

- You can use other methods rather than VFI here, (which can be faster).
- See the appendix for more details.

Roadmap

Summary

- Recursive competitive equilibrium.