Lecture II
Solving Representative Agent General Equilibrium Models

Adam Hal Spencer

The University of Nottingham

Applied Computational Economics 2020

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),
  - Heterogeneous agents, (aggregate uncertainty).

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

- Three predominant approaches we can take when dealing with market economies, (i.e. not social planner's problem)
    - Arrow-Debreu (valuation) equilibrium,
    - Sequential markets equilibrium,
    - Recursive competitive equilibrium.
- The third 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 three 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:

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

- In this setup, consumption at time t is the numerairé.

Concept (1): Arrow-Debreu Equilibrium

- An A-D equilibrium treats this as a static GE problem with an infinite number of goods to allocate across (given that we have infinite time periods).
- The households all trade only at $t = 0$ and deal in irrevocable claims to commodities indexed by time.
- They buy claims to consumption at any arbitrary time $t \geqslant 0$.
- Claim trading closes at the end of $t = 0$. These markets then close forever and then the "world plays-out" until the end of time.
- Firms produce and hand-over their goods, but only in accordance with what was agreed at $t = 0$, (no more negotiations).
- All agents bound to follow contracts set-out at $t = 0$.

Concept (1): Arrow-Debreu Equilibrium

- We take consumption at time $t = 0$ as the numerairé here.
- Price of consumption at time t relative to time $t = 0$ is denoted p_t, (where $p_0 = 1$).

Concept (1): Arrow-Debreu (A-D) Equilibrium

- An A-D equilibrium is a set of
  - Prices $\{p_t^*\}_{t=0}^{\infty}$ (consumption goods), $\{r_t^*\}_{t=0}^{\infty}$ (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 $t = 0$ budget constraint

$$\sum_{t=0}^{\infty} p_t^* [c_t + k_{t+1}] \leqslant \sum_{t=0}^{\infty} p_t^* [r_t^* k_t + (1 - \delta) k_t]$$

with k_0 taken as given.

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

Concept (1): Arrow-Debreu (A-D) Equilibrium

(2) Sequence $\{k_t^*\}_{t=0}^{\infty}$ solves the firm problem

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

(3) Markets clear

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

Concept (2): Sequential Markets Equilibrium

- How about we have trading in assets taking place at every $t \geqslant 0$.
- Each period, markets open all the trading takes place, they close and then open again next period.
- Seems a bit more natural...

Concept (2): 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

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

with k_0 taken as given.

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

Concept (2): Sequence Markets Equilibrium

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

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

$\forall t$.

(3') Markets clear

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

$\forall t$.

A-D and Sequential Markets Equilibria

- Notice again that these two methods involve 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 (3) Recursive Competitive Equilibrium

- We can start by again thinking about the household's recursive formulation.
- There's a slight twist on what we looked at in the partial equilibrium setup last time.
- Factor prices in this market economy are functions of the representative agents' choices, but they're meant to be taking these things as given.
- Trick: "big-K, little-k".

Concept (3) 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 (3) 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 (3) 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 (3) 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 (3) 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

- The notion of a recursive competitive equilibrium is extendible to a stochastic world, (assuming a Markov technology process).
- Again, just build-off the ideas of recursive household representations and consistency.
- Let's assume for now that the household can only save through capital, meaning that markets are incomplete.
- Everything is the same as before except output comes through $A_t K_t^{\alpha}$ where A_t follows a Markov process.

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

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

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

1. Introduction
2. Equilibrium Concepts in Dynamic Models
3. Extending to Stochastic Models
4. **Computing RCE via Value Function Iteration**
5. Transition Dynamics Between RCE: Shooting Algorithm
6. Local Solutions: Perturbation Methods
7. Conclusion

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

Transition Dynamics

- I've been pushing recursive competitive equilibria on you all day.
- We were looking for functions of the state space that were invariant over time.
- What happens in the face of a government policy change though?
- Thing must be changing in the short-run as we transition to a new RCE.

Transition Dynamics

- Consider the social planner's formulation for the neoclassical growth model.
- I.e. let's step away from markets for a second just to keep things simple.
- The concepts are the same when you have a decentralised economy.

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

subject to

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

for some initial k_0.

Transition Dynamics

- Optimality condition for capital investment is

$$u'(c_t) = \beta u'(c_{t+1})[1 - \delta + \alpha k_{t+1}^{\alpha-1}]$$

- Start at point k_0 and the model will eventually converge to a steady state given by

$$k^{ss} = \left\{ \left(\frac{1}{\beta} - (1 - \delta) \right) \frac{1}{\alpha} \right\}^{\frac{1}{\alpha-1}}$$

Transition Dynamics

- Say that the economy was in this steady state until time $t = 0$, (i.e. k_0 is equal to the initial steady state).

- Then the depreciation rate magically (unanticipated) increases to $\delta' > \delta$. Will stay with this depreciation rate forever more.

- The new steady state is given by

$$k^{ss'} = \left\{ \left(\frac{1}{\beta} - (1 - \delta') \right) \frac{1}{\alpha} \right\}^{\frac{1}{\alpha - 1}} < k^{ss}$$

Transition Dynamics

- Ok...fine. How do we reach this new steady state though?
- We have initial and endpoint conditions.
- Also know the sufficient conditions for the optimum (Euler equation and resource constraint).
- Just need to map journey between the two steady states.

Shooting Algorithm

(1) Guess the number of time periods it takes to transition to the new steady state. Call this number $S \in \mathbb{N}$.

(2) Guess your initial value for consumption, c_0.

(3) This then implies your initial investment

$$\Rightarrow k_1 = (k^{ss})^{\alpha} + (1-\delta)k^{ss} - c_0$$

(4) Iterate on your Euler equation to get c_1

$$c_1 = u'^{-1}\left\{u'(c_0)\beta^{-1}[1-\delta+\alpha k_1^{\alpha}]^{-1}\right\}$$

which is just $c_1 = \beta c_0[1-\delta+\alpha k_1^{\alpha-1}]$ if log utility.

Shooting Algorithm

(5) Repeat this procedure until time S.

(6) Will give you a candidate transition path

$$\{c_0, c_1, ..., c_S\} \text{ and } \{k_{ss}, k_1, k_2, ..., k_S\}$$

check if $k_S = k^{SS'}$. Stop if sufficiently close.

(7) If not sufficiently close, update your guess of c_0

- If $k_S < k^{SS'}$ then lower c_0.
- If $k_S > k^{SS'}$ then increase c_0. Return to step (3).

Shooting Algorithm

- Pretty straightforward right?
- Things get more complicated if we don't have closed-form optimality conditions, (e.g. heterogeneous agents, discrete choices, more control variables).
- So we'll revisit this notion of transition dynamics when we get to heterogeneous agents models.

Roadmap

Global v.s. Local Solutions

- The methods we've covered so far yield global solutions to problems.
- Gives you the solution to the problem over the entire domain, (or an approximation to it).
- Local solutions, in contrast, give you solutions in a small neighbourhood of some point.

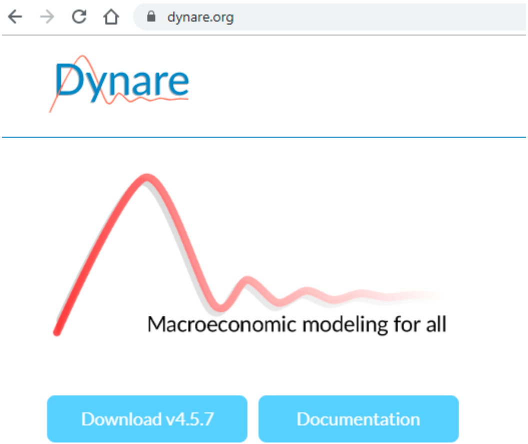
Global v.s. Local Solutions

- When would we want to use a local solution?
- If we're thinking about small (temporary) deviations from a steady state.
- Again, emphasis on small.
- The approach is inaccurate for large deviations.

Dynare

- The good thing about local solutions is that the barrier to implementation is incredibly low.
- Some good citizens (predominantly based in France), developed a software called Dynare.
- Their current slogan says it all...

Dynare


dynare.org
Dynare
Macroeconomic modeling for all
Download v4.5.7
Documentation

Dynare

- This software is a toolbox for Matlab.
- Easy to install, easy to use.
- Solves and estimates dynamic rational expectations models with little-no programming experience.
- Can use maximum likelihood or Bayesian estimation.
- Very popular with people at central banks.
- Used a lot by researchers of the new Keynesian DSGE paradigm.

Dynare

- All you have to do is find the optimality conditions for your problem and type them into a simple script.
- Dynare uses local approximations to these optimality conditions to find your solution.
- You can either find the local approximation yourself, (I'll talk about this in a moment), or even get the software to do it for you.

Dynare

- The software's output depends on whether you're just solving or also estimating a model.
- Solving a model gives output of locally-approximated policy functions.
- Estimation also tells you the parameter values of the model that are internally consistent with the data you provide it with.

Dynare

- I'm all in favour of this software as it does make structural modelling super-accessible.
- To give you an idea: I had three undergraduates last year who studied new Keynesian models with Dynare.
- Really great stuff and a true service to the profession that these developers are contributing.

Controversial Statements

- There's always a *but* with these things though.
- I'm not going to teach you how to use it.
- I've made you aware of it and I'll explain the general idea behind perturbation methods in the slides to come.
- Coming from graduate school at Wisconsin, I can't in all good conscience teach you how to use Dynare here...

Controversial Statements

- My advisor once said:

  > *These heavy computational guys like Victor Rios-Rull would never speak to you again if they found out that you used Dynare (Corbae, 2014).*

- I think he might have also used the words "non-macho", "non-kosher" and Dynare in the same sentence.

- If you want to be a serious quantitative economist, you need to code things up for yourself.

- Dynare's like a black box. Fine for policymakers who want a quick quantitative estimate. Won't get you very far (in terms of journal ranking) if you're writing a paper you want to publish though.

Perturbation Methods

- Say that we're trying to solve a functional equation of the form

$$\mathscr{F}(x, x') = 0$$

Perturbation Methods

- The perturbation approach approximates using

$$x'(x, \tilde{b}) = \sum_{i=0}^{n} \tilde{b}_i (x - x_0)^i$$

where x_0 is a particular point and $\tilde{b}$ is a vector of coefficients.

- The solution is analytic for a neighbourhood around x_0.

Perturbations about Steady State

- The typical thing to do in economics is perturb the model about its non-stochastic steady state.
- Shut-down the shocks and find the point where all the variables are constant.
- Then look for a locally analytic solution.
- Although, in principle, you can approximate about any point.

Perturbations about Steady State

- Consider the stochastic growth model

$$\max_{\{c_t, k_{t+1}\}_{t=0}^{\infty}} \mathbb{E}_0 \sum_{t=0}^{\infty} \beta^t c_t^{1-\sigma}$$

subject to

$$\begin{aligned} c_t + k_{t+1} &= a_t k_t^{\alpha} + (1-\delta) k_t \\ \log(a_t) &= \rho \log(a_{t-1}) + \sigma \epsilon_t, \;\; \epsilon_t \sim N(0,1) \end{aligned}$$

Perturbations about Steady State

- Euler equation for an arbitrary time preiod

$$c_t^{-\sigma} = \beta \mathbb{E}_t \left\{ c_{t+1}^{-\sigma} [\alpha a_{t+1} k_{t+1}^{\alpha-1} + (1-\delta)] \right\}$$

- In the non-stochastic steady state, $\epsilon_t = 0 \ \forall t$. Gives

$$\begin{aligned} \log(a_t) &= \rho \log(a_{t-1}) \\ \Rightarrow a_t &= a^{ss} \\ &= 1 \end{aligned}$$

meaning that

$$\begin{aligned} 1 &= \beta \left\{ \alpha (k^{ss})^{\alpha-1} + (1-\delta) \right\} \\ \Rightarrow k^{ss} &= \left\{ \left(\frac{1}{\beta} - (1-\delta) \right) \frac{1}{\alpha} \right\}^{\frac{1}{\alpha-1}} \end{aligned}$$

just as before.

Perturbations about Steady State

- Also have the steady state resource constraint

$$c^{ss} = (k^{ss})^{\alpha} - \delta k^{ss}$$

Perturbations about Steady State

- Let's pick a perturbation parameter λ.

- In this example, λ is such that

$$\log(a_t) = \rho \log(a_{t-1}) + \lambda \sigma \epsilon_t$$

where $\lambda = 1$ is the stochastic case and $\lambda = 0$ is the deterministic steady state.

- We now search for decision rules of the form

$$c_t = c(k_t, a_t, \lambda)$$
$$k_{t+1} = k(k_t, a_t, \lambda)$$

Perturbations about Steady State

- We now seek a local approximation about the point $(k^{ss}, 1, 0)$ for (k_t, a_t, λ).
- We want to approximate the policy functions $c = c(k, a)$ and $k' = k(k, a)$ locally.
- How? Taylor's theorem.

Taylor's Theorem

- The policy function expansions are of the form

$$\begin{aligned}
c_t &= c(k^{ss}, a_t, 1) \\
&= c(k^{ss}, 1, 0) \\
&+ c_k(k^{ss}, 1, 0)(k_t - k^{ss}) + c_a(k^{ss}, 1, 0)(a_t - 1) + c_\lambda(k^{ss}, 1, 0)\lambda \\
&+ \frac{1}{2}c_{kk}(k^{ss}, 1, 0)(k_t - k^{ss})^2 + \frac{1}{2}c_{ka}(k^{ss}, 1, 0)(a_t - 1)(k_t - k^{ss}) + \ldots
\end{aligned}$$

  look familiar?

- Same idea for the k policy function.

Taylor's Theorem

- Recall we had two equilibrium conditions, which we'll now denote by

$$\vec{F}(k_t, a_t, \lambda) =$$
$$\mathbb{E}_t \left[\begin{array}{c} c(k_t, a_t, \lambda)^{-\sigma} - \beta \left\{ (c(k(k_t, a_t, \lambda), a_{t+1}, \lambda))^{-\sigma} [\alpha a_{t+1} k(k_t, a_t, \lambda)^{\alpha-1} + (1-\delta)] \right\} \\ c(k_t, a_t, \lambda) + k(k_t, a_t, \lambda) - (1-\delta) k_t - a_t k_t^{\alpha} \end{array} \right]$$

where $\vec{F}(k_t, a_t, \lambda) = \vec{0}$ from our FOCs.

- We can also denote this as

$$\vec{F}(k_t, a_t, \lambda) = \vec{\mathscr{F}}(c_t, c_{t+1}, k_t, k_{t+1}, a_t, \lambda)$$

where I write it in this way to make explicit that the dependence through the states come through policy functions for c_t, c_{t+1} and k_{t+1}.

- Denote the derivative of the j^{th} entry of $\vec{\mathscr{F}}$ by $\vec{\mathscr{F}}_j$

Zeroth-Order Expansion

- See that

$$\vec{F}(k^{ss}, 1, 0) = \vec{0}$$
$$\Rightarrow k^{ss} = \left\{ \left(\frac{1}{\beta} - (1 - \delta) \right) \frac{1}{\alpha} \right\}^{\frac{1}{\alpha - 1}}$$
$$\Rightarrow c^{ss} = (k^{ss})^{\alpha} - \delta k^{ss}$$

i.e. just our steady state conditions.

First-Order Expansion

- Take first order derivatives and see that

$$\vec{F}_k(k^{ss}, 1, 0) = 0$$
$$\vec{F}_a(k^{ss}, 1, 0) = 0$$
$$\vec{F}_\lambda(k^{ss}, 1, 0) = 0$$

why the zeros?

First-Order Expansion

- Zeros follow from the fact that $\vec{F}(k_t, a_t, \lambda) = \vec{0}$ always.
- See then that

$$\vec{F}_k(k,1,0) = \vec{\mathscr{F}}_1 c_k + \vec{\mathscr{F}}_2 c_k k_k + \vec{\mathscr{F}}_3 + \vec{\mathscr{F}}_4 k_k = 0$$
$$\vec{F}_a(k,1,0) = \vec{\mathscr{F}}_1 c_a + \vec{\mathscr{F}}_2 \left[c_k k_a + c_a \frac{\partial a_{t+1}}{\partial a_t} \right] + \vec{\mathscr{F}}_4 k_a + \vec{\mathscr{F}}_5 = 0$$

where c and k denote the policy functions for the controls.

First-Order Expansion

- Where

$$\vec{F}_k(k,1,0) = \vec{\mathscr{F}}_1 c_k + \vec{\mathscr{F}}_2 c_k k_k + \vec{\mathscr{F}}_3 + \vec{\mathscr{F}}_4 k_k = 0$$

$$\vec{F}_a(k,1,0) = \vec{\mathscr{F}}_1 c_a + \vec{\mathscr{F}}_2 \left[c_k k_a + c_a \frac{\partial a_{t+1}}{\partial a_t} \right] + \vec{\mathscr{F}}_4 k_a + \vec{\mathscr{F}}_5 = 0$$

  is a quadratic system of 4 unknowns (c_k, c_a, k_k, k_a) with 4 equations (given that we have both the Euler equation and resource constraint).

- Quadratic since we have these coefficients in cross products and the like.

Second-Order Expansion

- For this we then take the second derivatives around $(k, 1, 0)$.

$$F_{kk}(k,1,0) = 0$$
$$F_{ka}(k,1,0) = 0$$
$$F_{k\lambda}(k,1,0) = 0$$
$$F_{aa}(k,1,0) = 0$$
$$F_{a\lambda}(k,1,0) = 0$$
$$F_{\lambda\lambda}(k,1,0) = 0$$

How Many Orders?

- There are some things to note.
- First order approximations miss some things in relation to uncertainty.

How Many Orders?

- Fernandez-Villaverde et al. (2016) point out the following drawbacks of first order approximations
  - Hard to infer the welfare effects of uncertainty,
  - Solution can't generate risk premia for assets,
  - Can't study the consequences of a change in volatility.
- Ok...so go higher...more burdensome computationally though.
- Solving for more and more unknowns.

Roadmap

Summary

- Basically covered two solution techniques.
- Recursive competitive equilibrium.
- Local approximations.
- Both have advantages and drawbacks....the appropriate method really depends on the application you have in mind.