An Advanced Course On The Science and Art of DSGE Modelling

Day 3: Classical Heterogeneous Agent Models

Szabolcs Deák

Kirill Shakhnov

Hyungseok Joo

September 2021

Alternative Solutions Methods and Further Reading

Alternative Solution Methods

- There are several ways to solve heterogeneous agent models.
- The aim here was to give an introduction so we have chosen the simplest solution method: very transparent and easily implementable algorithm
- Also very inaccurate and inefficient.

Alternative Solution Methods

- The key to solving our heterogeneous agent model: the aggregation of the individual decision rules
- Two ingredients:
  - the decision rules, and
  - the cross-sectional distribution (possibly time varying)
- We will review different methods to improve our algorithms to compute these two ingredients: see the *Journal of Economic Dynamics & Control* (Volume 34, Issue 1) and the *Handbook of Computational Economics* (Volume 3, Chapter 6) for a more comprehensive overview
- We will also review two alternative solution algorithms by Reiter (2009) and Achdou et al. (2020)

Alternative Solutions Methods: Decision Rules

Alternative Solution Methods: Decision Rules

- We have used discrete value function iteration: we restricted households to choose a gridpoint
- We discuss three alternatives:
  - continuous value function iteration
  - endogenous gridpoint method
  - perturbation

Continuous Value Function Iteration

- Using a continuous support for state variables: very accurate, but can be slow

- Consider the Aiyagari model with discrete grids:

$$V(k_l, \varepsilon_j) = \max_{k'} U(Rk_l + \varepsilon_j W - k') + Q(k', \varepsilon_j) \tag{58}$$

$$\text{s.t.} \quad Q(k', \varepsilon_j) = \beta E\{V(k', \varepsilon')|\varepsilon\} = \beta \sum_{m=1}^{n_\varepsilon} \mathcal{P}^{\varepsilon}_{mj} V(k', \varepsilon_m) \tag{59}$$

$$k' \geq \underline{k} \tag{60}$$

- In order to compute the continuation value $Q(k', \varepsilon_j)$ in between gridpoints we can use interpolation.

Continuous Value Function Iteration

- Linear interpolation would give

$$Q(k', \varepsilon_j) = \beta \sum_{m=1}^{n_\varepsilon} \mathcal{P}^{\varepsilon}_{mj} \left[\omega V(k_s, \varepsilon_m) + (1-\omega))V(k_{s+1}, \varepsilon_m)\right] \tag{61}$$

where $k_s < k' < k_{s+1}$ and $\omega = (k_{s+1} - k')/(k_{s+1} - k_s)$

- The resulting decision rule
  - is more accurate
  - can be evaluated at any values of k: has implications for the computation of the wealth distribution
- There are many options available (quadratic, cubic, etc.).

Endogenous Gridpoint Method

- Carroll (2006) proposes an algorithm called endogenous gridpoint method (EGM).
- It allows us to use interpolation without the increased computational burden under certain conditions.
- VFI: for each gridpoint k_i we need to find k' and this is where the algorithm typically spends most of the time.
- EGM: for each gridpoint k'_i we need to find k.

Endogenous Gridpoint Method

- Consider again the Aiyagari model with interpolation:

$$V(k_l, \varepsilon_j) = \max_{k'} \log(\underbrace{Rk_l + \varepsilon_j W - k'}_{c}) + \underbrace{\beta[\mathcal{P}^{\varepsilon}_{1j} V(k', \varepsilon_1) + \mathcal{P}^{\varepsilon}_{2j} V(k', \varepsilon_2)]}_{Q(k', \varepsilon_j)} \quad (62)$$

- At each iteration we need to solve a maximization problem for each pair (k_l, ε_j) in order to update the value function.
- This is computationally very costly.

Endogenous Gridpoint Method

- Let us define the household's disposable income at the beginning of a period as:

$$x = Rk + \varepsilon W \tag{63}$$

- Then we can also write the problem as

$$\tilde{V}(x, \varepsilon_j) = \max_{k'} \log(\underbrace{x - k'}_{c}) + \underbrace{\beta[\mathcal{P}^{\varepsilon}_{1j}\tilde{V}(x', \varepsilon_1) + \mathcal{P}^{\varepsilon}_{2j}\tilde{V}(x', \varepsilon_2)]}_{\tilde{Q}} \tag{64}$$

- The two value functions $\tilde{V}(x, \varepsilon)$ and $V(k, \varepsilon)$ are different.
- The key insight of the algorithm: the continuation value $\tilde{Q}$ is a function of k' and ε

$$\tilde{Q}(k', \varepsilon_j) = \beta[\mathcal{P}^{\varepsilon}_{1j}\tilde{V}(Rk' + \varepsilon_1, \varepsilon_1) + \mathcal{P}^{\varepsilon}_{2j}\tilde{V}(Rk' + \varepsilon_2, \varepsilon_2)] \tag{65}$$

Endogenous Gridpoint Method

- So we can write the Bellman equation as

$$\tilde{V}(x, \varepsilon_j) = \max_{k'} \log(x - k') + \tilde{Q}(k', \varepsilon_j) \tag{66}$$

- The corresponding first-order condition has an analytical solution:

$$(x - k')^{-1} = \tilde{Q}_{k'}(k', \varepsilon_j) \quad \Rightarrow \quad c^* = x - k' = \frac{1}{\tilde{Q}_{k'}(k', \varepsilon_j)} \tag{67}$$

- Substituting back we obtain

$$\tilde{V}(x, \varepsilon_j) = \tilde{Q}(k', \varepsilon_j) - \log\left(\tilde{Q}_{k'}(k', \varepsilon_j)\right) \tag{68}$$

- Given grids for k' and ε and an initial guess for $\tilde{Q}(k', \varepsilon)$ we iterate on equations (63), (65) and (68).

Endogenous Gridpoint Method

- How does it work in practice?

  1. Guess the values $\tilde{Q}^{(0)}(k'_l, \varepsilon_j)$.
  2. Given $\tilde{Q}^{(n)}(k'_l, \varepsilon_j)$ compute the partial derivatives $\tilde{Q}^{(n)}_{k'}(k'_l, \varepsilon_j)$ and the values c^*_{lj}.
  3. Compute the values $x^*_{lj} = k'_l + c^*_{lj}$ and use (68) to compute $\tilde{V}(x^*_{lj}, \varepsilon_j)$.
  4. Use (63) to compute $\tilde{Q}^{(n+1)}(k'_l, \varepsilon_j)$

- As you can see no optimization is necessary, just basic matrix algebra, numerical differentiation, and interpolation.

- For implementation details see Barillas and Fernandez-Villaverde (2007).

Perturbation

- Using perturbation to solve for the decision rule: may be even less accurate, but extremely fast.
- Problem: perturbation is not suited for models with occasionally binding constraints (continuity, differentiability, etc.)
- Preston and Roca (2007) and Kim et al. (2010) replace the borrowing constraint with a penalty term in the utility function:
  - households are penalized when capital holdings move close to the lower bound
  - the maximization problem has equality constraints only

Alternative Solution Methods

- The key to solving our heterogeneous agent model: the aggregation of the individual decision rules
- Two ingredients:
  - the decision rules, and
  - the cross-sectional distribution (possibly time varying)
- We will review different methods to improve our algorithms to compute these two ingredients: see the *Journal of Economic Dynamics & Control* (Volume 34, Issue 1) and the *Handbook of Computational Economics* (Volume 3, Chapter 6) for a more comprehensive overview
- We will also review two alternative solution algorithms by Reiter (2009) and Achdou et al. (2020)

Alternative Solutions Methods: Cross-sectional Distribution

Alternative Solution Methods: Cross-sectional Distribution

- As we discussed there are two methods:
  - Simulation: it is subject to noticeable sampling uncertainty even with a large number of agents
  - Non-stochastic simulation: it simulates the distribution directly and no idiosyncratic shocks are generated (hence the name)
- We have used simulation already, so now let us have a look at non-stochastic simulation.

Non-stochastic Simulation

- The decision rule $k' = g(k, \varepsilon)$ together with the law of motion for ε induce the following law of motion for each household in the Aiyagari model:

$$(k, \varepsilon) \xrightarrow[k' = g(k,\varepsilon)]{\text{decision}} (k', \varepsilon) \xrightarrow[Prob(\varepsilon'|\varepsilon)]{\text{shock}} (k', \varepsilon') \xrightarrow[k''_i = g(k',\varepsilon')]{\text{decision}} (k'', \varepsilon') \xrightarrow[Prob(\varepsilon''|\varepsilon')]{\text{shock}} (k'', \varepsilon'')$$

- Within each period there is an independent transition along both dimensions of the unconditional probability distribution:

  1. every household learns its new productivity, followed by
  2. a saving decision

- There are 2 distributions: we are interested in the one after the saving decisions, but before next period's labour productivities become known

Non-stochastic Simulation

- We have discretized the state space:
  - there are n possible asset levels, $\{k_1, \ldots, k_n\}$
  - there are 2 possible productivity levels, $\{\varepsilon_1, \varepsilon_2\}$
- We have $2n$ distinct types of households and we can represent the distribution $\lambda(k, \varepsilon)$ as an $n \times 2$ matrix.
- Given the discrete nature of the probability distribution, its evolution can be written in the form

$$\mathrm{vec}(\lambda') = \Gamma \, \mathrm{vec}(\lambda) \tag{71}$$

- Γ needs to be constructed based on the individual decision rule and the transition matrix for labour productivity.

Non-stochastic Simulation

- The transition probability from, say, (k_l, ε_j) to (k_s, ε_m) can be determined as follows:
  - There are 2 transitions: $(k_l, \varepsilon_j) \to (k_l, \varepsilon_m) \to (k_s, \varepsilon_m)$
  - $(k_l, \varepsilon_j) \to (k_l, \varepsilon_m)$: determined by $\mathcal{P}^{\varepsilon}_{mj}$
  - $(k_l, \varepsilon_m) \to (k_s, \varepsilon_m)$: determined by the decision rule as

$$G^m_{sl} = Prob[(k_l, \varepsilon_m)|(k_s, \varepsilon_m)] = \begin{cases} 1 & \text{if } k_s = g(k_l, \varepsilon_m) \\ 0 & \text{otherwise} \end{cases} \tag{69}$$

- Consequently, the transition probability is given by

$$\begin{aligned} P((k_s, \varepsilon_m)|(k_l, \varepsilon_j)) &= P[(k_s, \varepsilon_m)|(k_l, \varepsilon_m)]P[(k_l, \varepsilon_m)|(k_l, \varepsilon_j)] \\ &= G^m_{sl}\mathcal{P}^{\varepsilon}_{mj} \end{aligned} \tag{70}$$

- It gives the fraction of households transitioning.

Non-stochastic Simulation

- We can collect these probabilities in matrices:

$$\mathrm{vec}(\lambda') = \underbrace{\Gamma_G * \Gamma_{\mathcal{P}}}_{\Gamma} \mathrm{vec}(\lambda) \tag{75}$$

where

$$\Gamma_G = \begin{bmatrix} G^1 & \mathbf{0}_n \\ \mathbf{0}_n & G^2 \end{bmatrix} \qquad \Gamma_{\mathcal{P}} = \mathcal{P}^{\varepsilon} \otimes \mathbf{I}_n \tag{74}$$

- The order of the matrices in (75) is very important.

Non-stochastic Simulation

- We are interested in the stationary (aka ergodic) distribution, which is the time-invariant unconditional distribution that solves

$$\mathrm{vec}(\lambda) = \Gamma\,\mathrm{vec}(\lambda)$$

- The solution is the eigenvector (normalized to satisfy $\sum_i \lambda_i = 1$) associated with a unit eigenvalue of Γ.

Non-stochastic Simulation

1. Use a direct method:
   - You can get all the eigenvalues and eigenvectors this way.
   - Direct methods use some form of matrix decomposition: slow and memory intensive for large matrices
   - Matlab: `eig` command
2. The power iteration algorithm: iterate on

$$\mathrm{vec}(\lambda_{k+1}) = \Gamma\,\mathrm{vec}(\lambda_k) \tag{76}$$

   - Yields the dominant eigenvalue only: exactly what we need
   - Iterative method that avoids matrix decomposition: fast
   - Matlab: `eigs` command
3. See notes.
4. See notes.

Non-stochastic Simulation

- Once we have determined the stationary distribution:

$$\int k' d\lambda(k', \varepsilon) = \sum_{l=1}^{n} \sum_{j=1}^{2} \lambda(k_l, \varepsilon_j) k_l \tag{27}$$

- Essentially we are integrating the grid for k wrt the end-of-period distribution.

- Alternatively we can integrate the decision rule. The market clearing condition can be written as:

$$K' = \int k' d\lambda(k', \varepsilon) = \int g(k, \varepsilon) d\lambda(k, \varepsilon) \tag{83}$$

Non-stochastic Simulation

- We are integrating wrt the beginning-of-period distribution (before saving decisions), which solves:

$$\text{vec}(\lambda') = \underbrace{\Gamma_{\mathcal{P}} * \Gamma_G}_{\Gamma} \text{vec}(\lambda) \tag{84}$$

- Once the distribution is determined:

$$K' = \int g(k, \varepsilon) d\lambda(k, \varepsilon) = \sum_{l=1}^{n} \sum_{j=1}^{2} g(k_l, \varepsilon_j) \lambda(k_l, \varepsilon_j) \tag{85}$$

- Exercise 6 asks you to implement (85) and compare the two integrals.

Non-stochastic Simulation

- We can generalize it to models with aggregate uncertainty.
- The resulting dynamics can be written as:

$$\mathrm{vec}(\lambda_t) = \Gamma_t \, \mathrm{vec}(\lambda_{t-1}) \tag{77}$$

- The individual decision rules vary with the aggregate states: we need to reconstruct Γ_G every period.

Non-stochastic simulation

- Approximating the wealth distribution by a histogram constitutes a piecewise approximation.
- Algan et al. (2010) parametrize the density function to describe the cross-sectional distribution.
- Example: normal distribution can be parametrized by two parameters only
- They use exponential of polynomials as the functional form: it is a very flexible yet accurate tool.

Alternative Solution Methods

- The key to solving our heterogeneous agent model: the aggregation of the individual decision rules
- Two ingredients:
  - the decision rules, and
  - the cross-sectional distribution (possibly time varying)
- We will review different methods to improve our algorithms to compute these two ingredients: see the *Journal of Economic Dynamics & Control* (Volume 34, Issue 1) and the *Handbook of Computational Economics* (Volume 3, Chapter 6) for a more comprehensive overview
- We will also review two alternative solution algorithms by Reiter (2009) and Achdou et al. (2020)

Additional Alternative Solutions Methods

Alternative solution methods

- Reiter (2009) develops a solution algorithm to KS type models that combines projection and perturbation.

- The main idea of the algorithm is very similar to the way DSGE models are solved using perturbation:

  - first we obtain a fixed point of the model (steady state)

  - then we log-linearize the model about this point

  - the steady state of a DSGE model: no uncertainty

  - the steady state of a heterogeneous agent model: no aggregate uncertainty (the stationary equilibrium)

  - we perturb the equilibrium conditions of the model (including the dynamics of the cross-sectional wealth distribution) around the stationary equilibrium to see how they change once aggregate uncertainty is introduced

Alternative solution methods

- Achdou et al. (2020) develop a solution algorithm for heterogeneous agent models in continuous time.

- In continuous time, the equilibrium of heterogeneous agent models can be conveniently characterized as systems of partial differential equations which can be solved easily.

- Consider the first-order condition the of the households' maximization problem in the Aiyagari model:

$$(Rk + \varepsilon W - k'(k, \varepsilon))^{-1} = \beta E V_{k'}(k'(k, \varepsilon), \varepsilon') \tag{87}$$

- We usually need to
  - resort to costly root-finding operations for solving it
  - evaluate the integral on the left-hand side of the equation multiple times

Alternative solution methods

- The continuous-time approach sidesteps both of these difficulties completely.

- In continuous time "tomorrow" is the same thing as "today" and the first-order condition becomes

$$(Rk + \varepsilon W - k'(k, \varepsilon))^{-1} = \beta V_{k'}(k'(k, \varepsilon), \varepsilon) \tag{88}$$

- There is no expectation: never a need for computing integrals

- Like EGM: given a guess for the value function, the first-order condition can be solved analytically.

- Benjamin Moll's website has lots of resources and it is a good point to start learning about this approach.