Lectures 11
Bewley models
Part III

Macroeconomics 4

A.Y. 2014-15

Binning

- The ergodic distribution can be computed using a "binning" approach, as in Young (2010).
- The distribution λ is approx. with a **histogram** over a uniformly dist. grid on $\left[0, \bar{\mathrm{k}}\right] \times \mathcal{S}$, say $\{\mathrm{k}_j\}_{j=1}^{M} \times \mathcal{S}$, where $\mathrm{k}_1 = 0$, $\mathrm{k}_M = \bar{\mathrm{k}}$, and $M \geq m$.
- The histogram is a $M \times n$ matrix $\boldsymbol{\lambda}$, whose element $\lambda_{j,i}$ represents the mass of agents with capital k_j and labor endowment s_i.
- Hence, K can be approx. as $K = K' \approx \sum_{j=1}^{M} \sum_{i=1}^{n} \mathrm{k}'_{j,i} \lambda_{j,i}$, where $\mathrm{k}'_{j,i}$ denotes the optimal k' at node $\{\mathrm{k}_j, \mathrm{s}_i\}$, and is possibly obtained via interpolation (if the two grids do not coincide).

Binning

- Suppose that a strictly positive mass of agents, say υ, saves an amount k' such that $k_z \leq k' \leq k_{z+1}$ for some $z \in \{1, 2, ..., M\}$.
- The key step is to allocate the mass υ to the nodes k_z and k_{z+1} in such a way that K remains unaffected.
- If ω_z denotes the share of mass υ that ends up at node k_z, then the previous requirement boils down to:

$$\omega_z k_z + (1 - \omega_z) k_{z+1} = k'.$$

- Hence, the mass υ is distributed according to:

$$\omega_z(k') = \begin{cases} \frac{k' - k_{z-1}}{k_z - k_{z-1}} & \text{if } k' \in [k_{z-1}, k_z], \\ \frac{k_{z+1} - k'}{k_{z+1} - k_z} & \text{if } k' \in (k_z, k_{z+1}], \\ 0 & \text{otherwise.} \end{cases}$$

Binning

- Note that the two special cases $z = 1$ and $z = M$ have to be taken care separately:
  - If $z = 1$, then:

$$\begin{cases} \omega_1(k') = \frac{k_2 - k'}{k_2 - k_1} & \text{if } k' \in [k_1, k_2], \\ \omega_1(k') = 1 & \text{if } k' < k_1. \end{cases}$$

  - If $z = M$, then:

$$\begin{cases} \omega_M(k') = \frac{k' - k_{M-1}}{k_M - k_{M-1}} & \text{if } k' \in [k_{M-1}, k_M], \\ \omega_M(k') = 1 & \text{if } k' > k_M. \end{cases}$$

- Furthermore, note that $\omega_z(k') \geq 0 \ \forall z$, $\omega_z(k') > 0$ for **at most** two values of z, and:

$$\sum_{j=1}^{M} \omega_z\left(\mathrm{k}'_{j,i}\right) = 1, \ \ \forall z, i.$$

Binning

- The *LoM* for $\boldsymbol{\lambda}$ boils down to

$$\lambda_{z,q} = \sum_{j=1}^{M} \sum_{i=1}^{n} \Pi_{i,q} \omega_z \left(\mathrm{k}'_{j,i} \right) \lambda_{j,i} = \sum_{i=1}^{n} \Pi_{i,q} \sum_{j=1}^{M} \omega_z \left(\mathrm{k}'_{j,i} \right) \lambda_{j,i}.$$

- We can rewrite the *LoM* more compactly as:

$$\lambda_{z,q} = \left[\begin{array}{c|c|c} \Pi_{1,q} \mathbf{g}_{z,1} & \cdots & \Pi_{n,q} \mathbf{g}_{z,n} \end{array} \right] \mathrm{vec} \left(\boldsymbol{\lambda} \right),$$

  where:

$$\mathbf{g}_{z,i} \equiv \left[\begin{array}{cccc} \omega_z \left(\mathrm{k}'_{1,i} \right) & \omega_z \left(\mathrm{k}'_{2,i} \right) & \cdots & \omega_z \left(\mathrm{k}'_{M,i} \right) \end{array} \right].$$

Binning

- By simply stacking $\lambda_{z,q}$ for $z = 1, 2, ..., M$, we get that:

$$\begin{bmatrix} \lambda_{1,q} \\ \vdots \\ \lambda_{M,q} \end{bmatrix} = \begin{bmatrix} \Pi_{1,q}\mathbf{G}_1 & | & \cdots & | \, \Pi_{n,q}\mathbf{G}_n \end{bmatrix} \text{vec}\left(\boldsymbol{\lambda}\right),$$

where:

$$\mathbf{G}_i = \begin{bmatrix} \mathbf{g}_{1,i} \\ \vdots \\ \mathbf{g}_{M,i} \end{bmatrix}.$$

Binning

- Hence, the *LoM* can be written in matrix form as:

$$\text{vec}\,(\boldsymbol{\lambda}) = \mathbf{G}^T\,\text{vec}\,(\boldsymbol{\lambda})\,,$$

where:

$$\mathbf{G} \equiv \begin{bmatrix} \Pi_{1,1}\mathbf{G}_1^T & \Pi_{1,2}\mathbf{G}_1^T & \cdots & \Pi_{1,n}\mathbf{G}_1^T \\ \Pi_{2,1}\mathbf{G}_2^T & \Pi_{2,2}\mathbf{G}_2^T & \cdots & \Pi_{2,n}\mathbf{G}_2^T \\ \vdots & \vdots & \ddots & \vdots \\ \Pi_{n,1}\mathbf{G}_n^T & \Pi_{n,2}\mathbf{G}_n^T & \cdots & \Pi_{n,n}\mathbf{G}_n^T \end{bmatrix} = (\boldsymbol{\Pi}\otimes\mathbf{I}_M)\begin{bmatrix} \mathbf{G}_1^T & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{G}_2^T & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{G}_n^T \end{bmatrix}.$$

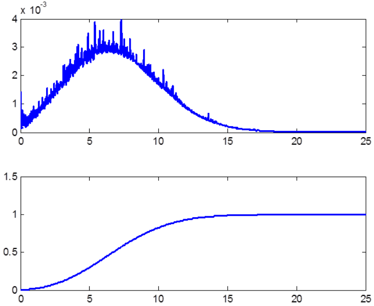
Binning

- The $Mn \times Mn$ matrix $\mathbf{G}$ is *right stochastic*, being $\mathbf{\Pi}$ so by assumption, and sparse.
- Hence, $\text{vec}\left(\boldsymbol{\lambda}\right)$ can be interpreted as the ergodic distribution of a discrete Markov chain characterized by the transition matrix $\mathbf{G}$.
- To compute it, iterate until convergence on the following recursive scheme:

$$\text{vec}\left(\boldsymbol{\lambda}_{k+1}\right) = \mathbf{G}^T \, \text{vec}\left(\boldsymbol{\lambda}_k\right).$$

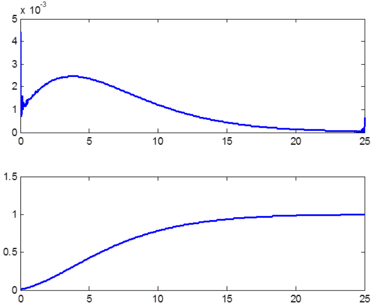
Numerical results

Discretized endo. state +
Value Func. Iter.



Nodes for k and k': 1000
Realizations of shock: 5
Running time: 27.1 seconds

Continuous endo. state +
Fixed Point Iter. on Euler eq.



Nodes for k: 1000
Realizations of shock: 5
Running time: 23.9 seconds

Computing the value funcion

- Once the policy function is available, we can solve, if needed, for the value function by iterating until convergence on:

$$V_{i,z+1}(k) = u\left[c(k, \mathrm{s}_i)\right] + \beta \sum_{j=1}^{n} \Pi_{ij} V_{j,z}\left[k'(k, \mathrm{s}_i)\right],$$

  where:

$$k'(k, \mathrm{s}_i) = (1 - \delta + r)\, k + w\mathrm{s}_i - c(k, \mathrm{s}_i).$$

- Note that, again, $V_{j,z}\left[k'(k, \mathrm{s}_i)\right]$ has to be computed via interpolation.

Transitions

- We will now learn how to compute transition dynamics.
- At $t = 0$, the economy reached a steady state characterized by a pol. function c_0, a couple of values $\{w_0, r_0\}$, and a distribution λ_0.
- *Unexpectedly*, some exogenous elements of the economy change in $t = 1$: this event was completely unexpected by all agents (a zero prob. event)
- This shock (permanent or transitory) drives the economy to a (possibly) new steady state, for $t \to \infty$.
- Assume that it takes a large but finite number of periods, say $T = 1000$, to reach the new steady state.
- The new steady state is characterized by c_T, $\{w_T, r_T\}$, and λ_T.

Transitions

- The goal is to find sequences $\{c_t\}_{t=1}^{T-1}$, $\{w_t, r_t\}_{t=1}^{T-1}$, and $\{\lambda_t\}_{t=1}^{T-1}$ that satisfy the definition of a recursive equilibrium.
- The intuition goes as follows:
  - Given $\{w_t, r_t\}_{t=1}^{T-1}$ and c_T, we can compute the policy functions $\{c_t\}_{t=1}^{T-1}$ by simply solving backwards the Euler equation.
  - Given λ_0 and $\{c_t\}_{t=1}^{T-1}$, we can solve forward for $\{\lambda_t\}_{t=1}^{T-1}$.
  - Given $\{\lambda_t\}_{t=1}^{T-1}$, we can solve for $\{K_t\}_{t=1}^{T-1}$ and $\{L_t\}_{t=1}^{T-1}$.
  - Given $\{K_t\}_{t=1}^{T-1}$ and $\{L_t\}_{t=1}^{T-1}$, we can solve for the implied sequence of factor prices$\{\hat{w}_t, \hat{r}_t\}_{t=1}^{T-1}$.
  - Given $\{\hat{w}_t, \hat{r}_t\}_{t=1}^{T-1}$, we can updated the guess $\{w_t, r_t\}_{t=1}^{T-1}$.
- Iterate the previous steps until convergence.

Transitions

Algorithm: how to compute transitions

1) Compute the initial and final steady states: c_0, $\{w_0, r_0\}$, λ_0, c_T, $\{w_T, r_T\}$, and λ_T.

2) Guess sequences of factor prices $\{w_{t,z}, r_{t,z}\}_{t=1}^{T-1}$, (i.e. guess sequences of K and N), where z denotes the iteration, and:

   a) For $t = T, T-1, ..., 2$, solve the Euler equation **backwards** for $c_{t-1}(\mathbf{x})$ given $c_t(\mathbf{x})$:

$$\begin{cases} u_c\left[c_{t-1}\left(\mathbf{x}_{t-1}\right)\right] = \beta \mathbb{E}\left\{u_c\left[c_t\left(\mathbf{x}_t\right)\right]\left(1-\delta+r_{t,z}\right) \mid \mathbf{x}_{t-1}\right\} & \text{if} \quad k_t > 0 \\ u_c\left[c_{t-1}\left(\mathbf{x}_{t-1}\right)\right] \geq \beta \mathbb{E}\left\{u_c\left[c_t\left(\mathbf{x}_t\right)\right]\left(1-\delta+r_{t,z}\right) \mid \mathbf{x}_{t-1}\right\} & \text{if} \quad k_t = 0 \end{cases},$$

   where:

$$k_t = \left(1-\delta+r_{t-1,z}\right) k_{t-1} + w_{t-1,z} s_{t-1} - c_{t-1}\left(\mathbf{x}_{t-1}\right).$$

   b) ...

Transitions

Algorithm: how to compute transitions

2) ...

   b) Given this sequence of policy functions, $\{c_{t,z}(\mathbf{x})\}_{t=1}^{T-1}$, and the initial dist. λ_0, compute the sequence of distributions $\{\lambda_{t,z}\}_{t=1}^{T-1}$:

$$\lambda_{t+1,z}(\mathbf{x}) = \int_{\mathcal{X}} \mathcal{I}_{t,z}(\mathrm{k}, k, s)\, \Pi(s, \mathrm{s})\, d\lambda_{t,z}, \quad \forall t \in [0, T-1],\ \forall \mathbf{x} \in \mathcal{X}.$$

   where:

$$\mathcal{I}_{t,z}(\mathrm{k}, k, s) = \begin{cases} 1 & \text{if } k'_{t,z} = \mathrm{k} \\ 0 & \text{if } k'_{t,z} \neq \mathrm{k} \end{cases}.$$

   c) ...

Transitions

Algorithm: how to compute transitions

2) ...

   c) Given the sequence of distributions $\{\lambda_{t,z}\}_{t=1}^{T-1}$, compute:

$$K_{t,z} = \int_{\mathcal{X}} k d\lambda_{t,z},$$
$$L_{t,z} = \int_{\mathcal{X}} s d\lambda_{t,z},$$

   for $t = 1, 2, ..., T-1$, and the implied sequences $\{\hat{w}_{t,z}, \hat{r}_{t,z}\}_{t=1}^{T-1}$.

   d) Compare $\{w_{t,z}, r_{t,z}\}_{t=1}^{T-1}$ to $\{\hat{w}_{t,z}, \hat{r}_{t,z}\}_{t=1}^{T-1}$, and update your guess as needed.

3) Iterate steps $(a) - (d)$ until convergence.

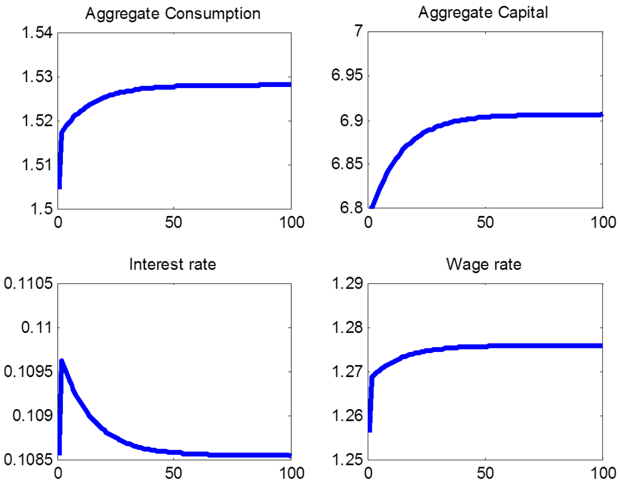
Transitions

- Note that step (a) in the previous algorithm can be performed quite easily:
  - Given $c_t(\mathbf{x})$, one can solve for $c_{t-1}(\mathbf{x})$ using time iteration on the Euler equation, but updating only $c_{t-1}(\mathbf{x})$, and **NOT** $c_t(\mathbf{x})$, at each iteration.
  - This allows also to use the same code (with just small and straightforward modifications) to compute the policy functions in steady state and along transitions.
- Furthermore, note that the "binning" approach lend itself to easily compute the sequence of distributions in step (b); the code used to compute the ergodic distribution can also be used to iterate on:

$$\text{vec}\left(\boldsymbol{\lambda}_{t+1,z}\right) = \mathbf{G}'_{t,z}\,\text{vec}\left(\boldsymbol{\lambda}_{t,z}\right).$$

Numerical results

- Transition after a 1% permanent increase in productivity:



Transitions and welfare

- Once the transition has been solved for, we can also solve for the value functions, $\{V_t(k,s)\}_{t=0}^{T}$.
- Note that:
  - $V_0(k,s)$ is the expected lifetime utility of an agent with states $\{k,s\}$ at time 0, in the initial stationary equilibrium.
  - $V_1(k,s)$ is the expected lifetime utility of an agent with the same states $\{k,s\}$ that has just been informed that there is a permanent structural change in the economy; V_1 takes into account all the transition dynamics through which the agent is going to live.
  - $V_T(k,s)$ is the lifetime utility of an agent with states $\{k,s\}$ born in the new stationary equilibrium (i.e. of an agent that does not live through the transition).

References I

Young, E. R. (2010, January). Solving the incomplete markets model with aggregate uncertainty using the Krusell-Smith algorithm and non-stochastic simulations. *Journal of Economic Dynamics and Control 34*(1), 36–41.