TIME ITERATION

Johannes Brumm

Karlsruhe Institute of Technology (KIT)

ZICE 2017

Agenda

- The (basic) time iteration algorithm
- The endogenous grid method
- Going non-Pareto-optimal: An example with heterogeneous agents and short-sale constraints
- Going high-dimensional: Adaptive simplicial interpolation (on separate slides)

Time Iteration: The Idea

Goes back to Coleman (1990) who calls it "policy-function iteration".
We follow Judd (1998) in calling it time iteration (TI).

- Start with a guess for the policy function 'tomorrow'
- Find policy 'today' that is optimal
  given that policy function 'tomorrow'
- Use this policy as new guess and iterate
- Hope that this procedure converges, i.e. that the policy does (almost) not change anymore
- The final policy (almost) satisfies the Euler equation
  when used 'today' and 'tomorrow'
- Then we have found an (approximate) recursive equilibrium

Time Iteration Algorithm (for Ramsey model)

0. Initial Step *(Set grid, initial policy, and error tolerance)*
   a) Set capital grid $K = [K_1 \; K_2 \; \dots \; K_n] \in \mathbb{R}^n_+, \; K_j < K_{j+1} \; \forall \; j$
   b) Set guess for policy function $p : [K_1, K_n] \to [K_1, K_n]$
   c) Set error tolerance for time iteration $\bar{\epsilon} > 0$

1. Main Step *(Update policy function)*
   a) For all $1 \leq j \leq n$:
   Solve Euler equation

$$u'(\bar{f}(k) - k^+) - \beta \cdot \bar{f}'(k^+) \cdot u'(\bar{f}(k^+) - k^{++}) = 0$$

   for optimal k^+ given $k = K_j$ and $k^{++} = p(k^+)$. Set $K_j^+ = k^+$.
   b) **Approximate** new policy $\tilde{p}$ by $\left\{K_j, K_j^+\right\}_{j=1}^n$.

2. Final Step *(Check error criterion)*
   a) Calculate error: $\epsilon = \|\tilde{p} - p\|_\infty / \|p\|_\infty$
   b) Set $p = \tilde{p}$.
   c) If $\epsilon < \bar{\epsilon}$, then stop and report results; otherwise go to step 1.

Choosing Equation Solver and Function Approximation

To implement policy function iteration, we need to choose:

- A method for solving equations, namely the Euler equation
- A method for approximating functions, namely the policy function

Depending on the problem, the best choices differ dramatically.
It's the art of computational economics to know when to use what!

As a start, we make the simplest choices

- We use Newton's method to solve the Euler equation
- We use piecewise linear interpolation to approximate the policy function

The Stochastic Ramsey Model with Discrete Shocks

Choose $\{c_t, k_{t+1}\}_{t=0}^{\infty}$ to maximize $\mathbb{E}_0\left[U(\{c_t\}_{t=0}^{\infty})\right]$ subject to

$$\forall t \in \mathbb{N}_0 : 0 \leq k_{t+1} \leq \underbrace{z_t f(k_t) + (1-\delta)k_t}_{\equiv \bar{f}(z_t, k_t)} - c_t, 0 \leq c_t, k_0, z_0 \text{ given}$$

where the expectation is over the sequence of stocks $\{z_t\}_{t=1}^{\infty}$ given z_0 which follow a discrete Markov process:

$$z_t \in \mathcal{Z} = \{1, \ldots, Z\}, \mathbb{P}(z_{t+1} = j | z_t = i) = \pi_{ij},$$

where π is the transition matrix. With a slight abuse of notation, we write $\pi(z_t, z_{t+1})$ for $\mathbb{P}(z_{t+1}|z_t)$.

Time Iteration Algorithm With Discrete Shocks

0. Initial Step *(Set grid, initial policy, and error tolerance)*
   a) Set capital grid $K = [K_1 \; K_2 \; \ldots \; K_n] \in \mathbb{R}^n_+, \; K_j < K_{j+1} \; \forall \; j$
   b) Set guess for policy function $p : \mathcal{Z} \times [K_1, K_n] \to [K_1, K_n]$
   c) Set error tolerance for time iteration $\bar{\epsilon}_{it} > 0$
1. Main Step *(Update policy function)*
   a) For all $z \in Z$ and for all $1 \leq j \leq n$:
      Solve Euler equation

$$u'(\bar{f}(k) - k^+) - \beta \sum_{z^+ \in Z} \pi(z, z^+) \cdot \bar{f}'(k^+) \cdot u'(\bar{f}(k^+) - k^{++}(z^+)) = 0$$

      for optimal k^+ given $k = K_j$ and $k^{++}(z^+) = p(z^+, k^+)$. Set $K_j^+ = k^+$.
   b) **Approximate** new policy $\tilde{p}$ by $\left\{K_j, K_j^+(z)\right\}_{1 \leq j \leq n, 1 \leq z \leq Z}$
2. Final Step *(Check error criterion)*
   a) Calculate error: $\epsilon_{it} = \|\tilde{p} - p\|_\infty / \|p\|_\infty$
   b) Set $p = \tilde{p}$
   c) If $\epsilon_{it} < \bar{\epsilon}_{it}$, then stop and report results; otherwise go to step 1

Discussion of Time Iteration

- To get convergence of the policy function under TI, we need same assumptions as for convergence of the value function under value function iteration (VFI), plus ...
  - Strict concavity u, to ensure that also policy functions converge under VFI, not just value functions
  - Differentiability of u and f, to ensure that we can use the Euler equation approach
- Interpolating the **policy function** "corresponds to" interpolating the **first derivative of the value function**. This has two advantages:
  - Interpolating the slope of the value function can give much more accurate result than interpolating the level of the value function.
  - One can use simpler interpolation methods for TI than for VFI: E.g. a piecewise linear interpolation of the slope of the value function corresponds to an interpolation of the value function that is differentiable and concavity preserving.

Endogenous Grid Method (EGM)

- Simple but powerful trick (from Carroll (2006EL)):
  Use a discrete grid for the optimal choice and trace back the state that induces this choice $\Rightarrow$ get an "endogenous" grid for the state from exogenous grid on the choice
- Can be used in combination with
  - VFI (see, e.g., Barillas and Fernandez-Villaverde (2007JECD))
  - TI (see, e.g., Krueger and Ludwig (2006JME))
- To explain the main idea, we discuss an endogenous-grid time-iteration algorithm to solve the deterministic Ramsey model

EGM for Deterministic Ramsey Model

- Recall Euler equation:

$$u'(c) = \beta u'\Big(f(k^+) + (1-\delta)k^+ - k^{++}\Big)\left[f'(k^+) + (1-\delta)\right]$$

- Invert to get c:

$$c = u'^{-1}\left(\beta u'\Big(f(k^+) + (1-\delta)k^+ - k^{++}\Big)\left[f'(k^+) + (1-\delta)\right]\right)$$

- Let $p(k)$ denote the policy function obtained in the previous iteration step of the time iteration procedure:

$$c = u'^{-1}\left(\beta u'\Big(f(k^+) + (1-\delta)k^+ - p(k^+)\Big)\left[f'(k^+) + (1-\delta)\right]\right)$$

EGM procedure

- Combine with resource constraint to get:

$$u'^{-1}\left(\beta u'\Big(f(k^+) + (1-\delta)k^+ - p(k^+)\Big)\left[f'(k^+) + 1 - \delta\right]\right) + k^+$$
$$= f(k) + (1-\delta)k$$

- An EGM time iteration procedure to solve the model is as follows:

1. Choose initial policy function p, set grid for capital choice $K^+ = [K_1^+ \ K_2^+ \ \ldots \ K_n^+] \in \mathbb{R}_+^n, \ K_j^+ < K_{j+1}^+ \ \forall \ j$, and set $\bar{\epsilon} > 0$.
2. For each $k^+ \in K^+$ solve the above equation for k given $p(k^+)$. From the mapping $k^+ \to k$ we get updated policy function $k^+ = \tilde{p}(k)$.
3. If $\|\tilde{p} - p\|_\infty / \|p\|_\infty < \bar{\epsilon}$ then stop and report results; otherwise set $p = \tilde{p}$ and go to step 1.

How to speed things up

- Each time iteration step involves solving a non-linear equation in k:

$$u'^{-1}\left(\beta u'\Big(f(k^+)+(1-\delta)k^+-p(k^+)\Big)\left[f'(k^+)+1-\delta\right]\right)+k^+$$
$$=f(k)+(1-\delta)k$$

- Use trick to avoid the root finding procedure:
  Instead of capital, use the total resources available in the economy as a state variable. Call it *wealth*, and denote it by:

$$w=f(k)+(1-\delta)k$$

- Policy function now maps wealth into capital choices: $k^+=p(w)$

$$k^+ + u'^{-1}\left(\beta u'\Big(w^+-p(w^+)\Big)\left[f'(k^+)+(1-\delta)\right]\right)=w,$$

where $w^+=f(k^+)+(1-\delta)k^+$

Endogenous Grid Time Iteration Algorithm

0. Initial Step *(Set grid, initial policy, and error tolerance)*
   a) Set grid for capital choice $[K_1^+ \; K_2^+ \; \ldots \; K_n^+] \in \mathbb{R}_+^n, \; K_j^+ < K_{j+1}^+ \; \forall \; j$
   b) Make guess for policy function $p : [W_1, W_n] \rightarrow [K_1^+, K_n^+]$ (with $W_1 = f(K_1^+) + (1-\delta)K_1^+, W_n = f(K_n^+) + (1-\delta)K_n^+$)
   c) Set error tolerance for time iteration $\bar{\epsilon} > 0$
1. Main Step *(Update policy function)*
   a) For all $1 \leq j \leq n$, set $k^+ = K_j^+, w^+ = f(k^+) + (1-\delta)k^+$, and define
   $$W_j = k^+ + u'^{-1}\left(\beta u'\left(w^+ - p(w^+)\right)\left[f'(k^+) + (1-\delta)\right]\right),$$
   b) **Approximate** new policy $\tilde{p}$ by $\left\{W_j, K_j^+\right\}_{j=1}^n$.
2. Final Step *(Check error criterion)*
   a) Calculate error: $\epsilon = \|\tilde{p} - p\|_\infty / \|p\|_\infty$
   b) Set $p = \tilde{p}$.
   c) If $\epsilon < \bar{\epsilon}$, then stop and report results; otherwise go to step 1.

Endogenous Grid Method: Discussion

- With an analytic expression for the state (here wealth) as a function of the choice (here capital), we can avoid the numerical root finding procedure in solving the Euler equation
- Moreover, EGM can handle some types of occasionally binding constraints by automatically placing endogenous grid points at values of the state where the policy function exhibits a kink.
- Both are great advantages, but it depends on the model structure whether they are present:
  - Including labor/leisure choice is already a non-trivial issue (Barillas and Fernandez-Villaverde (2007JEDC))
  - Handling occasionally binding constraints among endogenous variables with EGM is only possible for specific functional forms (see Hintermayer and Koeniger (2010JEDC))

Time Iteration for Non-Optimal Economies

- Time iteration can also be applied to non-optimal economies.
- However, convergence (and also existence) results are often not available, except for special cases, see e.g. Coleman (1991).
- We look at a simple heterogeneous agent model with short-sale constraints
- We iterate on a system of equilibrium conditions that includes the Euler equations of all (types of) agents plus market clearing conditions

A Simple Heterogeneous Agent Model: Overview

- Infinite horizon economy with discrete shocks
- There a two (types of) agents in the economy who receive a stochastic (labor) income stream
- There is one asset in unit net supply that pays a stochastic dividend stream (a 'Lucas tree')
- This asset can be traded between agents subject to short-sale constraints $\Rightarrow$ The endogenous state is thus the share of the Lucas tree that agent 1 holds
- By increasing the number of agents, extending the number of assets, and changing the constraints, we can later make the model much more interesting and computationally demanding . . .

A Simple Heterogeneous Agent Model: Assumptions

- Infinite horizon endowment economy, $t \in \mathbb{N}_0$, with discrete shocks, z_t, and Markov transition matrix π, where:
  $z_t \in \mathcal{Z} = \{1, \ldots, Z\}$ and $\pi(z_t, z_{t+1}) = \mathbb{P}(z_{t+1}|z_t)$.
- There a two (types of) agents in the economy with CRRA preferences, $\{\gamma_i, \beta_i\}_{1 \leq i,j \leq 2}$, who receive a stochastic endowment: $e^1(z_t)$, $e^2(z_t)$
- There is a Lucas tree that pays a stochastic dividend stream: $d(z_t)$
- Agents can trade shares, s^i, of the Lucas tree at price p
- The Lucas tree is in unit net supply and there are short-sale constraints on the Lucas tree: $s^1 + s^2 = 1$, $s^1 \geq 0, s^2 \geq 0$
- Aggregate endowments are given by: $\bar{e}(z_t) = e^1(z_t) + e^2(z_t) + d(z_t)$

A Simple Heterogeneous Agent Model: Agents' Problem

Each agent i chooses $\{c_t^i, s_{t+1}^i\}_{t=0}^{\infty}$ to maximize

$$\mathbb{E}_0\left[\ \sum_{t=0}^{t} \beta^t \frac{(c_t^i)^{1-\gamma}}{1-\gamma}\ \right] \quad \text{s.t. } c_t^i + p_t s_{t+1}^i \leq e_t^i + s_t^i(p_t + d_t),$$

$$c_t^i \geq 0,\ \ s_{t+1}^i \geq 0\ \forall t \in \mathbb{N}_0$$

$s_0^i, z_0,$ and $\{p_t(z_t)\}_{t,z^t}$ given.

Denoting the multiplier for the $s_{t+1} \geq 0$ constraint by μ_t,
the FOCs for s_{t+1} are:

$$u'(c_t^i)p_t - \mu_t^i = \beta\mathbb{E}\left[u'(c_{t+1}^i)(p_{t+1} + d_{t+1})\right],$$

$$s_{t+1}^i \geq 0, \mu_t^i \geq 0, s_{t+1}^i \cdot \mu_t^i = 0 \text{ (complementarity condition)}$$

Sequential Equilibrium Conditions

In a sequential equilibrium, the following holds for all (t, z_t):
(below all the z_t are omitted)

- Aggregate resource constraint: $c_t^1 + c_t^2 = e_t^1 + e_t^2 + d_t$
- Market clearing for the Lucas tree: $s_t^1 + s_t^2 = 1$
- First-order optimality conditions for $i = 1, 2$:

$$u'(c_t^i)p_t - \mu_t^i = \beta\mathbb{E}\left[u'(c_{t+1}^i)(p_{t+1} + d_{t+1})\right],$$

$$s_{t+1}^i \geq 0,\ \mu_t^i \geq 0,\ s_t^i \cdot \mu_t^i = 0$$

Recursive Equilibrium

Choose (z, s^1) as recursive state and find policy function $f : \mathcal{Z} \times [0,1] \to \mathbb{R}^7_+$,

$$f(z,s^1) = (c^1(z,s^1), c^2(z,s^1), s^1_+(z,s^1), s^2_+(z,s^1), \mu^1(z,s^1), \mu^2(z,s^1), p(z,s^1))$$

that satisfies for all $(z, s^1) \in \mathcal{Z} \times [0,1]$ the equilibrium conditions (EC):

- Aggregate resource constraint:
  $c^1(z,s^1) + c^2(z,s^1) = e^1(z) + e^2(z) + d(z)$
- Market clearing for the Lucas tree: $s^1_+(z,s^1) + s^2_+(z,s^1) = 1$
- Budget constraints for $i = 1, 2$:
  $c^i(z,s^1) + p(z,s^1)s^i_+(z,s^1) = e^i(z,s^1) + s^i \cdot (p_t(z,s^1) + d_t(z))$
  (note: s^i is the state, $s^2 = 1 - s^1$)
- First-order optimality conditions for $i = 1, 2$:

$$u'(c^i(z,s^1))p(z,s^1) - \mu^i(z,s^1) - \beta\mathbb{E}\left[u'(c^i(z_+,s^1_+)))(p(z_+,s^1_+) + d(z_+)\right] = 0,$$

$$s^i_+(z,s^1) \geq 0,\ \ s^i_+(z,s^1) \cdot \mu^i_t(z,s^1) = 0, \mu^i_t(z,s^1) \geq 0$$

Compute Recursive Equilibrium

- Using this recursive structure, we can apply a time iteration algorithm.
- In each time iteration step, for each state z, and each grid point for s^1, one takes a guess $f : \mathcal{Z} \times [0,1] \rightarrow \mathbb{R}^7_+$ for next-periods policy as given and solves for the new policy satisfying (EC)
- We can restrict attention to three dimensions of f, as the only next-period variables appearing in (EC) are c^1_+, c^2_+, and p_+ (note that s^1_+, s^2_+ are determined this period)
- We cannot simply use a root-finder to solve the system of equilibrium conditions, because they contain the complementarity conditions. There are two ways around that problem:
  - Use a method that can handle the complementarity condition / inequality constraints (see, e.g., ZICE lecture on complementarity conditions, or Miranda and Fackler (2004), pp. 44–50)
  - Transform the inequalities into equations by defining auxiliary variables (e.g. as in Garcia and Zangwill (1981))

TI Algorithm For Heterogeneous Agents

0. Initial Step *(Set grid, initial policy, and error tolerance)*
   a) Set grid $S = [0 \ S_2 \ \ldots \ S_{n-1} \ 1], \ S_j < S_{j+1} \ \forall \ j$
   b) Set guess for policy function $f : \mathcal{Z} \times S \to \mathbb{R}^3_+, \ f(z, s^1) = \left(c^1, c^2, p\right)$
   c) Set error tolerance for time iteration $\bar{\epsilon} > 0$
1. Main Step *(Update policy function)*
   a) For all $z \in Z$ and for all $1 \leq j \leq n$:
      Solve equilibrium conditions (EC) for $(c^1, c^2, s^1_+, s^2_+, \mu^1, \mu^2, p)$
      given $s^1 = S_j$ and $(c^1_+, c^2_+, p_+) = f(z_+, s^1_+)$.
      Set $c^1_j(z) = c^1, c^2_j(z) = c_2, p_j(z) = p$.
   b) **Approximate** new policy $\tilde{f}$ by $\left\{S_j, \left(c^1_j(z), c^2_j(z), p_j(z)\right)\right\}_{1 \leq j \leq n, 1 \leq z \leq Z}$
2. Final Step *(Check error criterion)*
   a) Calculate error: $\epsilon = \|\tilde{f} - f\|_\infty / \|f\|_\infty$
   b) Set $f = \tilde{f}$
   c) If $\epsilon < \bar{\epsilon}$, then stop and report results; otherwise go to step 1

Summary and Outlook

- So far, you have seen the basic TI algorithm, how the EGM can speed it up, how it can be applied to non-optimal economies
- In what follows, you will see how the time iteration algorithm can be applied to problems of higher dimension:
  - Adaptive Simplicial Interpolation (see separate slides)
  - Adaptive Sparse Grids (see Simon's presentation)