Quantitative Dynamic Programming

Seyed Ali Madanizadeh
Sharif U. of Tech.

April 2019

Outline

- Sequential Dynamic Problem
- Dynamic Programming
  - Bellman Equation and Principle of Optimality
  - Contraction Mapping Theorem
  - FOCs
  - Simple Examples
- Numerical Solutions to DP
  - Value Function Iteration
  - Policy Function Iteration
  - Log-Linearization
  - Polynomial Approximation
- DP with Shocks
  - IID shocks
  - State dependent shocks
- Linear Quadratic Approximation

Outline

- Applications
  - Consumption Saving
  - Search Models and Equilibrium Unemployment
  - Investment Adjustment Costs: Convex and Non-Convex
    - Smooth Pasting and Super Contact
  - Monetary Policy and the sS Policies
  - Financial Constraint
    - Households borrowing constraint
    - Firms Borrowing Constraints
- Recursive Competitive Equilibrium
  - Small k, Big K
- Calibration and Estimation of Dynamic Models
- DP in continuous form
  - Speed of Convergence
- Hetereogenous Agent Models

Ramsey Model

- A Simple Neoclassical Growth Model

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

$$\begin{aligned} \text{s.t. } c_t + i_t &= f(k_t) \\ k_{t+1} &= (1-\delta) k_t + i_t \end{aligned}$$

Dynamic Macro problem

- Setup the dynamic problem
  - Determine the exogenous parameters and the endogenous allocations and prices.
- Write First Order Conditions (FOCs) and simplify them
- Solve for the Steady State solution in terms of parameters
- Comparative Statics on the exogenous parameters
- Solve for the transitional dynamics
- Comparative statics for the speed of convergence

First Order Conditions

$$
\begin{aligned}
[c_t] &: \beta^t u_c(c_t) = \lambda_t \\
[k_{t+1}] &: \lambda_t = \lambda_{t+1}\left(1 - \delta + f_{k,t+1}\right)
\end{aligned}
$$

- Taking $U(c) = \log c, f(k) = Ak^{\alpha} \Rightarrow$

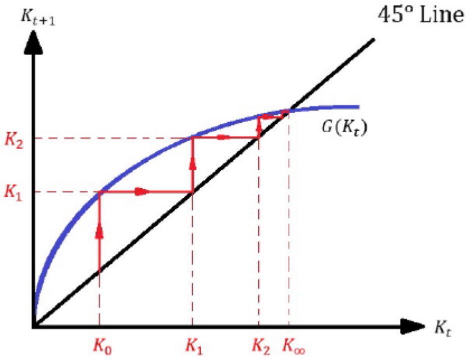
$$
\frac{1}{c_t} = \frac{1 - \delta + \alpha A k_{t+1}^{\alpha-1}}{c_{t+1}}
$$

Steady State:

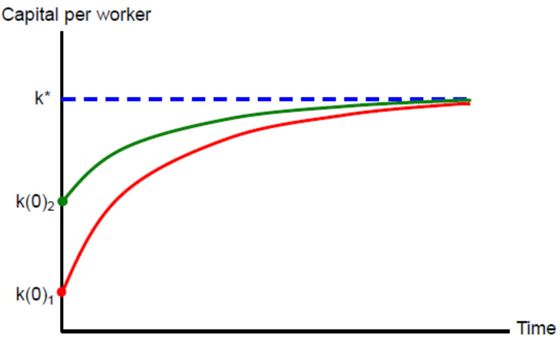
- $c_t = \bar{c}, k_t = \bar{k} \Rightarrow$

$$
\begin{aligned}
\bar{k} &= \left(\frac{\alpha A}{1-\delta}\right)^{\frac{1}{1-\alpha}} \\
\bar{\imath} &= \delta \bar{k} \\
\bar{c} &= A\left(\frac{\alpha A}{1-\delta}\right)^{\frac{\alpha}{1-\alpha}}\left(1-\frac{\alpha\delta}{1-\delta}\right)
\end{aligned}
$$

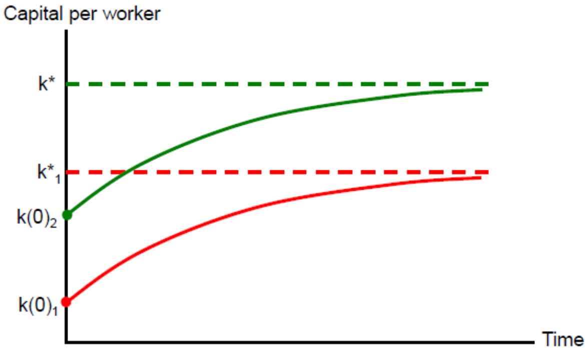
Capital choices and capital dynamics



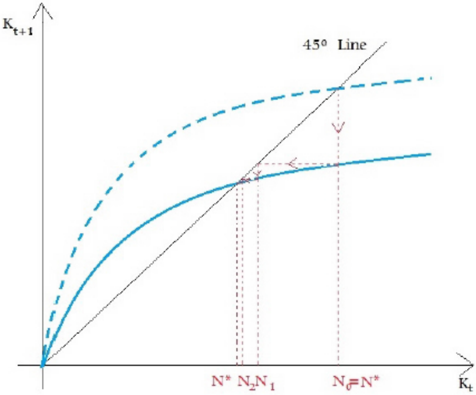
Transition



Transition Paths For Two Economies



Policy Function



Dynamic Programming

- Dynamic Programming version of the Problem

$$V(k) = \max_{\{c,k',i\}} \{U(c) + \beta V(k')\}$$

$$\begin{aligned} \text{s.t. } c + i &= f(k) \\ k' &= (1-\delta)k + i \end{aligned}$$

- Simplified:

$$V(k) = \max_{k'} \{U(f(k) + (1-\delta)k - k') + \beta V(k')\}$$

Dynamic Programming: General Form

- Recursive Problem: Bellman Equation

$$V(x) = \max_{\{y \in \Gamma(x)\}} \{F(x, y) + \beta V(y)\}$$

maximizer of the RHS is maximized by the policy function $g(x) \Rightarrow$

$$V(x) = F(x, g(x)) + \beta V(g(x))$$

- Sequence problem

$$\begin{aligned} V^*(x_0) &= \max_{x_{t+1}} \sum_{t=0}^{\infty} \beta^t F(x_t, x_{t+1}) \\ \text{s.t } x_{t+1} &\in \Gamma(x_t) \quad \text{for all } t \geq 0 \end{aligned}$$

- Principle of Optimality:

$$V(x) = V^*(x) \quad \text{for all } x$$

Dynamic Programming and Contraction mapping

- Definition: Let (S, ρ) be a metric space. Let $T : S \rightarrow S$ be an operator. T is a contraction with modulus $\beta \in (0, 1)$ if

$$\rho\left(Tx, Ty\right) \leq \beta\rho\left(x, y\right)$$

- In our case, S will be the set of continuous and bounded functions from X to R, with the norm sup

- Contraction Mapping (CM) Theorem: If T is a contraction in (S, ρ) with modulus β, then
  1. there is a unique fixed point $s^* \in S$, such that
  $$s^* = Ts^*$$
  2. iterations of T converge to the fixed point
  $$\rho\left(T^n s_0, s^*\right) \leq \beta^n \rho\left(s_0, s^*\right)$$
  for any $s_0 \in S$.

Dynamic Programming

- Define the Bellman operator T as

$$(Tv)(x) = \max_{\{y \in \Gamma(x)\}} \{F(x, y) + \beta V(y)\}$$

  Assume F is bounded and continuous, and that Γ is continuous and has compact range.
- Theorem: T maps the set of continuous and bounded functions S into itself. Moreover T is a contraction.
- and under regular conditions v^* is increasing and concave.

Dynamic Programming: FOCs

- Euler Equation:

$$0 = F_y\left(x, g\left(x\right)\right) + \beta V'\left(g\left(x\right)\right)$$

- Envelope Condition:

$$V'\left(x\right) = F_x\left(x, g\left(x\right)\right)$$

Dynamic Programming: FOCs

- Graphical Rperesentation

Example 1

- For the neoclassical growth model we obtain:

$$\begin{aligned} U'(f(k) - g(k)) &= \beta V'(g(k)) \\ V'(k) &= U'(f(k) - g(k))f'(k) \end{aligned}$$

Example 2

- Linear utility in the neoclassical growth model. Let $U(c) = c$ and

  $$f(k) = F(k, 1) + (1 - \delta)k$$

  where G is a neoclassical production function: strictly increasing and strictly concave in k, satisfying Inada conditions. Assume that $0 \leq k' \leq f\left(k\right)$ then

  $$V\left(k\right) = f\left(k\right) - k^* + \beta \frac{f\left(k^*\right) - k^*}{1 - \beta}$$

Example 3

- Consider the Neoclassical growth model with log utility, Cobb-Douglas production function and 100% depreciation

$$F(x, y) = \log(x^{\alpha} - y)$$
$$\Gamma(x) = [0, x^{\alpha}]$$

- then V is of the form

$$V(x) = a + b \log x$$
$$g(x) = cx^{\alpha}$$

Example 4

- Consider the problem of an agent with wages w that saves with safe gross rate of return $(1+r)$. The budget constraint is

$$x' + c = x(1+r) + w$$

where x is the beginning of period wealth, and x' are savings. Let $\beta(1+r) = 1$, $w > 0$, and U be strictly increasing, bounded, strictly concave, and C^2. Then:

$$\begin{aligned} g(x) &= x \\ c(x) &= w + rx \\ V(x) &= \frac{U(w+rx)}{1-\beta} \end{aligned}$$

Example 5

- Adjustment cost model

$$\begin{aligned} F(x,y) &= -\frac{a}{2}y^2 - \frac{b}{2}(y-x)^2 \\ \Gamma(x) &= R \end{aligned}$$

- Then

$$V(x) = -\frac{c}{2}x^2$$

Computation: Value Function Iteration

- The value function $V(.)$ can be obtained by an iterative technique:
- Value Function Iteration (VFI) - directly computes $V(x)$ and uses it to obtain the optimal policy functions. Usually focuses on solving the Bellman equation directly.
- Theoretical Algorithm

1. Start with a guess— some initial function $w(.)$
2. successively improve it by the *Bellman Operator*

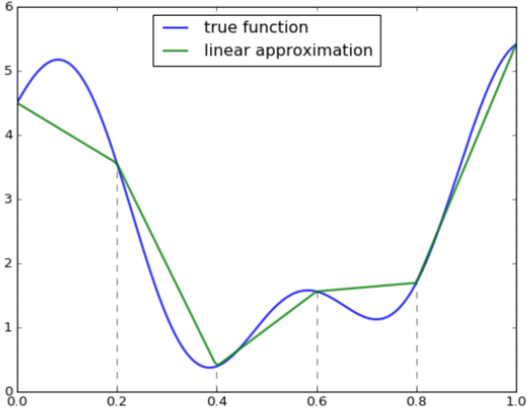
$$(Tw)(x) = \max_{\{y \in \Gamma(x)\}} \{F(x, y) + \beta w(y)\} \tag{1}$$

3. Iteratively applying T from initial condition w produces a sequence of functions w, Tw, $T(Tw) = T^2 w, \ldots$ that converges uniformly to V^*.

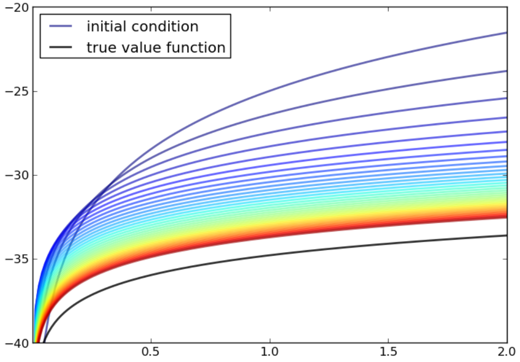
Computation: Value Function Iteration

1. Begin with an array of values $\{w_1, ..., w_I\}$, typically representing the values of some initial function w on the grid points $\{k_1, ..., k_I\}$
2. build a function $\hat{w}$ on the state space R^+ by interpolating the points $\{w_1, ..., w_I\}$.
3. By repeatedly solving (1) obtain and record the value $T\hat{w}(k_i)$ on each grid point k_i
4. Unless some stopping condition is satisfied, set $\{w_1, ..., w_I\} = \{T\hat{w}(k_1), ..., T\hat{w}(k_I)\}$ and go to step 2

Computation: Value Function Iteration



Computation: Value Function Iteration



Computation: Policy Function Iteration

- Policy Function Iteration (PFI) - computes the optimal policies directly.
- Often relies on the first order conditions alone.
- But the additional assumptions of di erentiability and concavity are not always satisfied so we often can not use it.
- It is also usually very sensitive, as it relies on non-linear equation solvers.
- VFI is extremely robust and can solve virtually any (well defined) dynamic programming problem, But it can be slow and subject to a curse of dimensionality.
- It relies on non-linear optimization, usually using discrete grids. The best approach is to frst characterize the problem first and then choose the more suitable method.

Computation: Policy Function Iteration

- Guess a policy function $g^{(0)}(k)$ (Use the M grids on $[0, k^*]$ and set the policy values for $k_j = \frac{j}{M}k^*$ as $g_j^{(0)}$)
- For any $n = 0, 1, ...$ iterate the followings until convergence

1. Construct $V'^{(n)}(k)$ using $V'^{(n)}(k) = U'(f(k) - g^{(n)}(k))f'(k)$
2. Use $V'^{(n)}(k)$ and solve for k' as the solution to $U'(f(k) - k') = \beta V'(k')$
3. Set $g^{(n+1)}(k) = k'(k)$.

Computation: Euler Equation Iteration

- Euler Equation Iteration (EEI): for each k, it calculates the optimal policy by iterating on the Euler equation:

$$0 = F_y\left(x, g\left(x\right)\right) + \beta V'\left(F_x\left(g\left(x\right), g\left(g\left(x\right)\right)\right)\right)$$

- For each x, we search for a value for $x' = g\left(x\right)$ such that the N'th iteration converges to the steady state value.
- By convergence, we mean that it is close enough to the steady state.
- Envelope Condition:

$$V'\left(x\right) = F_x\left(x, g\left(x\right)\right)$$

Algorithm: Euler Equation Iteration

- For each x_0, guess $x_0' = x_1$.
- Define x_2 that satisfies the following eqaution for $n = 0$

$$0 = F_y\left(x_n, x_{n+1}\right) + \beta V'\left(F_x\left(x_{n+1}, x_{n+2}\right)\right)$$

- Then continue this procedure for each $n \leq N$
- check weather $\left|x_N - x_{SS}\right| < \varepsilon$
- If so, algorith terminates and return $x_0' = x_1$.
- If not, use the bisection (or any other search algorithm) to update x_1 and redo the procedure, until convergence.

Adding a Shock

- The first step is to modify the problem to include a production shock.
- The shock sequence will be denoted $\{\zeta_t\}$ and assumed to be IID for simplicity.
- Many treatments include ζ_t as one of the state variables but this can be avoided in the IID case if we choose the timing appropriately.

Stochastic Neoclassical Growth Model

- Consider a simple Consumption-Investment problem of a Social problem with productivity shock
- Timing

1. At the start of period t, current output y_t is observed
2. Consumption c_t is chosen, and the remainder $y_t - c_t$ is used as productive capital.
3. The shock ξ_{t+1} is realized.
4. Production takes place, yielding output $y_{t+1} = f(y_t - c_t)\xi_{t+1}$

- The "current" shock ξ_{t+1} has subscript $t+1$ because it is not in the time t information set
- The production function f is assumed to be continuous
- The shock is multiplicative by assumption—this is not the only possibility
- Depreciation is not made explicit but can be incorporated into the production function

Related Sequential Problem

$$\max_{\{c_t\}} E\left[\sum_{t=0}^{\infty} \beta^t U(c_t)\right]$$

$$\begin{aligned} \text{s.t. } y_{t+1} &= f(y_t - c_t)\,\xi_{t+1} \\ 0 &\leq c_t \leq y_t \end{aligned}$$

Bellman Equation with Uncertainty

- Bellman Equation:

$$V(y) = \max_{\{0 \leq c \leq y\}} \{U(c) + \beta E[V(f(y-c)\xi)]\}$$

Bellman Equation with Uncertainty

- An Example: $\log \xi \tilde{} N(0, \sigma)$ and

$$\begin{aligned} U(c) &= \log c \\ f(k) &= k^{\alpha} \end{aligned}$$

- Close Form Solution

Bellman Equation with IID shocks

- Consider a simple Consumption Saving problem of a HH with uncertain labor endowment s_t that follows an IID Shock

$$\max_{\{c_t\}} E\left[\sum_{t=0}^{\infty} \beta^t U(c_t)\right]$$

$$\begin{aligned} \text{s.t. } c_t + a_{t+1} &= (1+r)\,a_t + ws_t \\ 0 &\leq c_t \\ a_{t+1} &\in A \end{aligned}$$

Bellman Equation with IID shocks

- Define $y = (1+r)\,a + ws$

$$\tilde{V}(y) = \max_{\{a' \leq y\}} \left\{ U\left(y - a'\right) + \beta E\left[\tilde{V}\left((1+r)\,a' + ws'\right)\right] \right\}$$

- Graphical Representation of the solution

Bellman Equation with Uncertainty

$$V(y) = \max_{\{0 \le c \le y\}} \left\{ U(c) + \beta E\left[V\left(f(y-c)\,\xi \right) \right] \right\}$$

- Solution: Define the Bellman Operator

$$Tw(y) = \max_{\{0 \le c \le y\}} \left\{ U(c) + \beta E\left[w\left(f(y-c)\,\xi \right) \right] \right\}$$

- We look for the operator fixed point: $Tv^* = v^*$.
- Value Function Iteration:

$$w_{n+1}(y) = \max_{\{0 \le c \le y\}} \left\{ U(c) + \beta E\left[w_n\left(f(y-c)\,\xi \right) \right] \right\}$$

- w_n converges to V
- We can use Monte Carlo to approximate

$$E\left[w\left(f(y-c)\,\xi \right) \right] \simeq \frac{1}{R} \sum_{r=1}^{R} w\left(f(y-c)\,\xi_r \right)$$

Bellman Equation with State-dependent shocks

- Shocks generally evolve as a Markov Chain
- Markov Chain:
  - Suppose a random process s_t can have m-states in each time t.
  - Suppose P is transition matrix such that the probability of going from state i to state j equals P_{ij}.
  - Then the probability density $\pi_{t+1} = P'\pi_t$
- AR Processes are other samples of Markov Chains

- Bellman Equation with State Dependent shock:

$$\begin{aligned} V(x,\xi) &= \max_{\{u\}} \left\{ r(x,u,\xi) + \beta E\left[V(x',\xi') \,|\xi \right] \right\} \\ x' &= g(x,u,\xi) \end{aligned}$$

- FOC:

$$\frac{\partial r}{\partial u}(x,u,\xi) + \beta E\left[\frac{\partial g}{\partial u}(x,u,\xi) \frac{\partial}{\partial x'} V(x',\xi') \,|\xi \right] = 0$$

- EC:

$$V'(x) = \frac{\partial r}{\partial x}(x,u^*,\xi) + \beta E\left[\frac{\partial g}{\partial x}(x,u^*,\xi) \frac{\partial}{\partial x'} V(x',\xi') \,|\xi \right]$$

where $u^* = h(x,\xi)$.

Bellman Equation with State-dependent shocks

- Stochastic Neoclassical Growth Model

$$V(k,\xi) = \max_{\{0 \le k' \le \xi f(k)\}} \left\{ U\left(\xi f(k) - k'\right) + \beta E\left[V\left(k',\xi'\right)|\xi\right]\right\}$$

where $\xi' = \rho\xi + \varepsilon$

- Then $k' = g(k,\xi)$
- Remember $k_{t+1} = ak_t + b\xi_t$

Bellman Equation with State-dependent shocks

- Consider a simple Consumption Saving problem of a HH with uncertain labor endowment s_t that follows a markov chain

$$\max_{\{c_t\}} E\left[\sum_{t=0}^{\infty} \beta^t U(c_t)\right]$$

$$\begin{aligned} \text{s.t. } c_t + a_{t+1} &= (1+r)\,a_t + ws_t \\ 0 &\leq c_t \\ a_{t+1} &\in A \end{aligned}$$

Bellman Equation with State-dependent shocks

- Consumption Saving problem

$$V(a,s) = \max_{\{0 \leq c\}} \left\{ U\left((1+r)a + ws - a'\right) + \beta E\left[V\left(a', s'\right)|s\right] \right\}$$

- then $a' = g(a,s)$: Show graphically (for each state)
- If shocks are IID, then:

$$V(a,s) = \max_{\{0 \leq c\}} \left\{ U\left((1+r)a + ws - a'\right) + \beta E\left[V\left(a', s'\right)\right] \right\}$$

- Define $y = (1+r)a + ws$

$$\tilde{V}(y) = \max_{\{a' \leq y\}} \left\{ U\left(y - a'\right) + \beta E\left[\tilde{V}\left((1+r)a' + ws'\right)\right] \right\}$$

where $V(a,s) = \tilde{V}((1+r)a + ws)$

Discrete Dynamic Programming

- A discrete DP is a maximization problem with an objective function of the form:

$$E\left[\sum_{t=0}^{\infty}\beta^{t}r\left(s_{t},a_{t}\right)\right]$$

- s_t is the state variable: $s_t \in S$
- a_t is the action: $a_t \in A\left(s_t\right)$
- β is a discount factor
- $r(s_t, a_t)$ is interpreted as a current reward when the state is s_t and the action chosen is a_t.
- Each pair (s_t, a_t) pins down transition probabilities $Q(s_t, a_t, s_{t+1})$ for the next period state s_{t+1}

Discrete Dynamic Programming

- Actions influence not only current rewards but also the future time path of the state
- The essence of dynamic programming problems is to trade off current rewards vs favorable positioning of the future state (modulo randomness)
- Examples:
  - consuming today vs saving and accumulating assets
  - accepting a job offer today vs seeking a better one in the future
  - exercising an option now vs waiting

Discrete Dynamic Programming

- Define

$$(Tv)(s) = \max_{a \in A(s)} \left\{ r(s,a) + \beta \sum_{s' \in S} v(s') \, Q(s,a,s') \right\}$$

- T is monotone and a contraction mapping with module β
- Thus, it has a unique fixed point:

$$v^*(s) = \max_{a \in A(s)} \left\{ r(s,a) + \beta \sum_{s' \in S} v^*(s') \, Q(s,a,s') \right\}$$

Bellman Equation with State-dependent shocks

- Discretize the grids for $A = \{a_1 < ... < a_n\} \Rightarrow$ for $i \in \{1, ..., m\}, h \in \{1, ..., n\}$

$$V(a_h, s_i) = \max_{\{0 \leq c, a' \in A\}} \left\{ U\left((1+r)\,a_h + ws_i - a'\right) + \beta \sum_{j=1}^{m} P_{ij} V\left(a', s_i\right) \right\}$$

- The Curse of Dimentionality

Bellman Equation with State-dependent shocks

- Suppsoe $m = 2$ (two employment state (high and low))
- Define two $n * 1$ vectors v_j where $v_j(i) = v(a_i, s_j)$
- Define two $n * n$ matrices R_j where $R_j(i,h) = U((1+r)a_i + ws_i - a_h)$
- Define an operator $T([v_1, v_2])$ that maps a pair of vectors $[v_1, v_2]$ into a pair of vectors $[Tv_1, Tv_2]$:

$$
\begin{aligned}
Tv_1 &= \max\left\{R_1 + \beta P_{11}\mathbf{1}v_1' + \beta P_{12}\mathbf{1}v_2'\right\} \\
Tv_2 &= \max\left\{R_2 + \beta P_{21}\mathbf{1}v_1' + \beta P_{22}\mathbf{1}v_2'\right\}
\end{aligned}
$$

- Then the Bellman Equation is

$$
[v_1, v_2] = T([v_1, v_2])
$$

- This can be solved by iteration:

$$
[v_1, v_2]_{r+1} = T([v_1, v_2]_r)
$$

Log Linearization

- Finding Steady States
- Log-Linearize around the steady state
- Use State Space Guess and Verify Method

$$\begin{aligned} k_{t+1} &= ak_t + b\xi_t \\ c_t &= dk_t + e\xi_t \end{aligned}$$

$$\begin{aligned} V(x) &= \max_{\{u_t\}} -\sum_{t=0}^{\infty} \left(x_t' R x_t + u_t' Q u_t\right) \\ \text{s.t. } x_{t+1} &= A x_t + B u_t \end{aligned}$$

- or

$$V(x) = \max_{u} -\left\{ \left(x' R x + u' Q u\right) + V\left(Ax + Bu\right) \right\}$$

- Guess:

$$V(x) = -x' P x$$

- Equivalent to:

$$-x' P x = \max_{u} -\left\{ \left(x' R x + u' Q u\right) - \left(Ax + Bu\right)' P \left(Ax + Bu\right) \right\}$$

Solution

$$
\begin{aligned}
V(x) &= -x'Px \\
u &= -Fx
\end{aligned}
$$

$$
\begin{aligned}
F &= \left(Q + B'PB\right)^{-1} B'PA \\
P &= R + A'PA - A'PB\left(Q + B'PB\right)^{-1} B'PA
\end{aligned}
$$

- Called the Algebraic Matrix Riccati Equation

- Value function iteration
  - Start from $P_0 = 0$

$$
\begin{aligned}
P_{j+1} &= R + A' P_j A - A' P_j B \left(Q + B' P_j B\right)^{-1} B' P_j A \\
F_{j+1} &= \left(Q + B' P_j B\right)^{-1} B' P_j A
\end{aligned}
$$

$$\begin{aligned} V(x) &= \max_{\{u_t\}} - \sum_{t=0}^{\infty} \beta^t \left(x_t' R x_t + u_t' Q u_t \right) \\ \text{s.t. } x_{t+1} &= A x_t + B u_t \end{aligned}$$

- or

$$V(x) = \max_u \left\{ \left(x' R x + u' Q u \right) + \beta V (Ax + Bu) \right\}$$

- Solution

$$\begin{aligned} V(x) &= -x' P x \\ u &= -F x \end{aligned}$$

$$\begin{aligned} F &= \beta \left(Q + B' P B \right)^{-1} B' P A \\ P &= R + \beta A' P A - \beta^2 A' P B \left(Q + \beta B' P B \right)^{-1} B' P A \end{aligned}$$

- Policy improvement algorithm
  - Starting from an initial F_0 for which the eigenvalues of $A - BF_0$ are less than $1/\sqrt{\beta}$ in modulus, the algorithm iterates on the two equations:

$$
\begin{aligned}
P_{j+1} &= R + F_j' Q F_j - \beta \left(A - BF_j\right)' P_j \left(A - BF_j\right) \\
F_{j+1} &= \beta \left(Q + \beta B' P_j B\right)^{-1} B' P_j A
\end{aligned}
$$

  - This is an example of a discrete Lyapunov or Sylvester equation

$$
P_j = \sum_{k=0}^{\infty} \beta^k \left(A - BF_j\right)'^k \left(R + F_j' Q F_j\right) \left(A - BF_j\right)^k
$$

  - If the eigenvalues of the matrix $A - BF_j$ are bounded in modulus by $1/\sqrt{\beta}$, then a solution of this equation exists.
  - This algorithm is typically much faster than the algorithm that iterates on the matrix Riccati equation.

$$
\begin{aligned}
V(x) &= \max_{\{u_t\}} -E_0 \sum_{t=0}^{\infty} \beta^t \left(x_t' R x_t + u_t' Q u_t \right) \\
\text{s.t. } x_{t+1} &= A x_t + B u_t + C \varepsilon_{t+1}
\end{aligned}
$$

where ε_{t+1} is an $(n \times 1)$ vector of random variables that is independently and identically distributed according to the normal distribution with mean vector zero and covariance matrix.

$$
E \varepsilon_t \varepsilon_t' = I
$$

Linear Quadratic Dynamic Programming

- Solution

$$
\begin{aligned}
V(x) &= -x'Px - d \\
u &= -Fx
\end{aligned}
$$

$$
\begin{aligned}
F &= \beta\left(Q + B'PB\right)^{-1} B'PA \\
P &= R + \beta A'PA - \beta^2 A'PB\left(Q + \beta B'PB\right)^{-1} B'PA \\
d &= \beta(1-\beta)^{-1} \operatorname{tr}\left(PCC'\right)
\end{aligned}
$$

Theorem

Certainty Equivalence Principle: The feedback rule that solves the stochastic optimal linear regulator■ problem is identical with the rule for the corresponding nonstochastic linear optimal regulator problem.

- Substituting the optimal control $u_t = -Fx_t$ into the law of motion

$$x_{t+1} = (A - BF)x_t$$

- The system is said to be stable if $\lim_{t\to\infty} x_t = 0$ starting from any initial $x_0 \in R^n$.
- Assume that the eigenvalues of $(A - BF)$ are distinct, and use the eigenvalue decomposition $A - BF = D\Lambda D^{-1}$

$$x_t = D\Lambda^t D^{-1} x_0$$

- Evidently, the system is stable for all $x_0 \in R^n$ if and only if the eigenvalues of $A - BF$ are all strictly less than unity in absolute value. Then $(A - BF)$ is said to be a **"stable matrix."**

Definition

The pair (A, B) is said to be stabilizable if there exists a matrix F for which $(A - BF)$ is a stable matrix.

Theorem

If (A, B) is stabilizable and R is positive definite, then under the optimal rule F , $(A - BF)$ is a stable matrix.

Example: Adjustment cost model (1)

- Cost minimization problem with convex adjustment cost

$$\begin{aligned} F(x,y) &= -\frac{a}{2}y^2 - \frac{b}{2}(y-x)^2 \\ \Gamma(x) &= R \end{aligned}$$

- Then

$$V(x) = -\frac{c}{2}x^2$$

Example: Adjustment cost model (2)

- Firm's value maximization problem (increasing marginal cost)

$$V\left(y\right)=\max_{y'}\left\{qy-\frac{k}{2}y^{2}-0.5d\left(y'-y\right)^{2}+\beta V\left(y'\right)\right\}$$

- FOC:

$$\beta V_{y}\left(y'\right)=d\left(y'-y\right)$$

- EC:

$$V_{y}\left(y\right)=q-ky+d\left(y'-y\right)$$

- Guess:

$$\begin{aligned} y' &= a+by \\ V\left(y\right) &= e+fy+0.5gy^{2} \end{aligned}$$

- Solution

$$\beta\left(f+gy'\right) = d\left(y'-y\right)$$
$$y' = \frac{\beta f+dy}{d-\beta g} \Rightarrow a=\frac{\beta f}{d-\beta g}, b=\frac{d}{d-\beta g}$$

-

$$V_y\left(y\right) = f+gy=p-ky+d\left(y'-y\right)=\left(p+da\right)+\left(db-d-k\right)y$$
$$f = p+da$$
$$g = db-d-k=\frac{d^2}{d-\beta g}-d-k$$

- $g'=g/d, k'=k/d$

$$g' = \frac{1}{1-\beta g'}-1-k'$$
$$\Rightarrow \left(1+k'+g'\right)\left(1-\beta g'\right)=1$$
$$0 = k'-\left(1+k'\right)\beta g'+g'-\beta g'^2$$

Example: Adjustment cost model (3)

- Firm's value maximization problem (Constant marginal cost)

$$V(y) = \max_{y'} \left\{ py - 0.5d\left(y' - y\right)^2 + \beta V\left(y'\right) \right\}$$

- FOC:

$$\beta V_y\left(y'\right) = d\left(y' - y\right)$$

- EC:

$$V_y(y) = p + d\left(y' - y\right)$$

- Guess:

$$\begin{aligned} y' &= a + by \\ V(y) &= e + fy + 0.5gy^2 \end{aligned}$$

- Solution

$$\begin{aligned}\beta\left(f+gy'\right) &= d\left(y'-y\right)\\ y' &= \frac{\beta f+dy}{d-\beta g}\Rightarrow a=\frac{\beta f}{d-\beta g}, b=\frac{d}{d-\beta g}\end{aligned}$$

$$\begin{aligned}V_y\left(y\right) &= f+gy=p+d\left(y'-y\right)=\left(p+da\right)+\left(db-d\right)y\\ f &= p+da\\ g &= db-d=\frac{d^2}{d-\beta g}-d\Rightarrow\left(1+\frac{g}{d}\right)\left(1-\beta\frac{g}{d}\right)=1\\ 0 &= \frac{g}{d}\left(1-\beta-\beta\left(\frac{g}{d}\right)\right)\Rightarrow g=d\left(\frac{1-\beta}{\beta}\right),0\end{aligned}$$

$$a=\frac{\beta}{1-\beta}\frac{p}{d}, b=1, g=0, f=\frac{1}{1-\beta}p$$

Example: Adjustment cost model (4)

- Firm's value maximization problem (Constant marginal cost, dynamic states)

$$
\begin{aligned}
V(y,p) &= \max_{y'} \left\{ py - 0.5d\left(y'-y\right)^2 + \beta EV\left(y',p'\right) \right\} \\
p' &= Ap + B\xi
\end{aligned}
$$

- FOC:

$$\beta EV_y\left(y',p'\right) = d\left(y'-y\right)$$

- EC:

$$V_y\left(y,p\right) = p + d\left(y'-y\right)$$

- $$y' = g\left(y,p\right)$$

Dynamic Recursive Equilibrium

- Up to now, we have studied single-agent problems where components of the state vector not under the control of the agent were taken as given.
- Now we describe multiple-agent settings in which some of the components of the state vector that one agent takes as exogenous are determined by the decisions of other agents.
- We study partial equilibrium models of a kind applied in microeconomics
  - Rational expectations or recursive competitive equilibrium
  - Markov perfect equilibrium

Dynamic Recursive Equilibrium

- Start with a simple example: adjustment cost model

$$\max \sum_{t=0}^{\infty} \beta^t R_t$$

$$R_t = p_t y_t - 0.5d \left(y_{t+1} - y_t\right)^2$$

- The firm is a price taker:

$$p_t = A_0 - A_1 Y_t$$

- The firm believes that marketwide output follows the law of motion:

$$Y_{t+1} = H_0 + H_1 Y_t \equiv H(Y_t)$$

Dynamic Recursive Equilibrium

$$\begin{aligned} v\left(y, Y\right) &= \max_{y'} \left\{ A_0 y - A_1 Y y - 0.5 d \left(y' - y\right)^2 + \beta V\left(y', Y'\right) \right\} \\ \text{s.t.} \quad Y' &= H(Y) = H_0 + H_1 Y \end{aligned} \tag{2}$$

- FOC

$$\beta V_y\left(y', Y'\right) = d\left(y' - y\right)$$

- EC:

$$V_y\left(y, Y\right) = A_0 - A_1 Y + d\left(y' - y\right)$$

- The firm's optimal Policy

$$y' = h(y, Y)$$

- n identical firms, setting $Y_t = ny_t$ makes the actual law of motion for output for the market

$$Y' = nh(Y/n, Y) \tag{3}$$

- Thus, when firms believe that the law of motion for marketwide output is equation 2, their optimizing behavior makes the actual law of motion equation 3.
- A recursive competitive equilibrium equates the actual and perceived laws of motion

Dynamic Recursive Equilibrium

Definition

A recursive competitive equilibrium (a rational expectations equilibrium) of the model with adjustment costs is a value function $v(y, Y)$, an optimal policy function $h(y, Y)$, and a law of motion $H(Y)$ such that:
a. Given H, $v(y, Y)$ satisfies the firm's Bellman equation and $h(y, Y)$ is the optimal policy function.
b. The law of motion H satisfies $H(Y) = nh(Y/n, Y)$.

- The firm's optimum problem induces a mapping Φ from a perceived law of motion for capital H to an actual law of motion $\Phi(H)$.
- Try to address this problem by choosing some guess H_0 for the aggregate law of motion and then iterating with Φ.

- NO: We cannot Iterate.
- Unfortunately, the mapping Φ is not a contraction.
- In particular, there is no guarantee that direct iterations on F converge 1
- Fortunately, there is another method that works here
- The method exploits a general connection between equilibrium and Pareto optimality expressed
- in the fundamental theorems of welfare economics (see, e.g, [MCWG95])
- Lucas and Prescott [LP71] used this method to construct a rational expectations equilibrium

Dynamic Recursive Equilibrium

- A planning problem as a solution method
- The solution strategy is to match the Euler equations of the market problem with those for a planning problem that can be solved as a single-agent dynamic programming problem.
- The optimal quantities from the planning problem are then the recursive competitive equilibrium quantities, and the equilibrium price can be coaxed from shadow prices for the planning problem.

$$S_t = S\left(Y_t, Y_{t+1}\right) = \int_0^{Y_t} \left(A_0 - A_1 x\right) dx - 0.5d\left(Y_{t+1} - Y_t\right)^2$$

- The planning problem is to choose a production plan to maximize

$$V\left(Y\right) = \sum_{t=0}^{\infty} \beta^t S\left(Y_{t-1}, Y_t\right) \quad \text{for a given } Y_0$$

Dynamic Recursive Equilibrium

$$V\left(Y\right) = \max_{Y'} \left\{ A_0 - \frac{A_1}{2} Y^2 - 0.5d\left(Y' - Y\right)^2 + \beta V\left(Y'\right) \right\}$$

- FOC:

$$-d\left(Y' - Y\right) + \beta V'\left(Y'\right) = 0$$

- EC:

$$V'\left(Y\right) = A_0 - A_1 Y + d\left(Y' - Y\right)$$

- For $n = 1$, we set $y_t = Y_t$. We get the same equations.

Dynamic Recursive Equilibrium

- Guess:

$$Y' = H_0 + H_1 Y$$

- Solve for H_0, H_1. Then we can solve for $h(y, Y)$.

Recursive Competitive Equilibrium

- Let x be a vector of state variables under the control of a representative agent
- Let X be the vector of those same variables chosen by "the market."
- Let Z be a vector of other state variables chosen by "nature", that is, determined outside the model

$$\begin{aligned} v(x, X, Z) &= \max_u \left\{ R(x, X, Z, u) + \beta v\left(x', X', Z'\right) \right\} \\ \text{s.t. } x' &= g(x, X, Z, u) \\ X' &= G(X, Z) \\ Z' &= \zeta(Z) \end{aligned} \tag{4}$$

- The solution of the representative agent's problem is a decision rule

$$u = h(x, X, Z) \tag{5}$$

Recursive Competitive equilibrium

- To make the representative agent representative, we impose $X = x$, but only "after" we have solved the agent's decision problem.
- Substituting equation 5 and $X = x_t$ into equation 5 gives the actual law of motion

$$X' = G_A(X, Z)$$

where

$$G_A(X, Z) = g(X, X, Z, h(X, X, Z))$$

Recursive Competitive equilibrium

Definition

A recursive competitive equilibrium (rational expectations equilibrium) is a policy function h, an actual aggregate law of motion G_A , and a perceived aggregate law G such that (a) Given G, h solves the representative agent's optimization problem; and (b) h implies that $G_A = G$.

Recursive Competitive equilibrium

- Kydland Prescott (1982) & Mehra and Prescott (1980): Big K, Little k
- Define the economywide capital as K and household's own capital stock k, which it has control on it.
- In Equilibrium $k = K$.
- Household's state variables are (k, K).
- Household chooses consumption and investment (c, x)
- Household perceives that the capital K changes as $K' = (1 - \delta) K + X(K)$

- The representative firm

$$\max_{K,H} F(K,H) - rK - wH$$

- FOC:

$$\begin{aligned} w &= F_h(K,H) \\ r &= F_k(K,H) \end{aligned}$$

Recursive Competitive equilibrium

- HH's problem:

$$
\begin{aligned}
v(k,K) &= \max_{c,x\geq 0}\left\{u(c)+\beta v\left(k',K'\right)\right\} \\
c+x &\leq r(K)\,k+w(K) \\
k' &= (1-\delta)\,k+x \\
K' &= (1-\delta)\,K+X(K)\equiv D(K)
\end{aligned}
$$

- Let $d(k,K)$ be the optimal decision of the Household.
- Take labor supply $h=1$.
- In Equilibrium $k=K$.
- $d(.)$ should be consistent: $d(k,K)=D(K)$

Recursive Competitive equilibrium

- Here: A recursive competitive Equilibrium is a value function $v(k, K)$ and a policy function $d(k, K)$ (which gives decisions on $c(k, K), x(k, K)$) and an aggregate policy function $D(K)$ (which gives aggregte decisions $C(K), X(K)$ and factor prices $r(K), w(K)$ such that these functions satisfy
  1. the HH's problem
  2. the Firm's problem FOC necessary and sufficient conditions
  3. the consistency of individual and aggregate decisions; i.e. $d(K, K) = D(K)$
  4. The Aggregate Resource Constraint: $C(K) + X(K) = Y(K)$

Recursive Competitive equilibrium

- The statement that RCE is pareto optimal implies that $v(K, K)$ and $d(K, K)$ coincides with $V(K)$ and $D(K)$ for the social planner problem.

- We have shcoks z :

$$z' = \rho z + \varepsilon$$
$$e\tilde{\ } N\left(0, \sigma_{\varepsilon}\right)$$

- The representative firm

$$\max_{K,H} e^{z} F\left(K, H\right) - rK - wH$$

- FOC:

$$w = F_h\left(K, H\right)$$
$$r = F_k\left(K, H\right)$$

Recursive Competitive equilibrium: Stochastic

- HH's state variable (z, k, K) , Aggregate state variable (z, K)
- HH's problem:

$$
\begin{aligned}
v(z,k,K) &= \max_{c,x,h\geq 0}\left\{u(c,1-h)+\beta E\left[v\left(z',k',K'\right)|z\right]\right\} \\
c+x &\leq r(z,K)\,k+w(z,K)\,h \\
k' &= (1-\delta)\,k+x \\
K' &= (1-\delta)\,K+X(z,K)\equiv D(z,K) \\
z' &= \rho z+\varepsilon \\
c &\geq 0, 0\leq h\leq 1
\end{aligned}
$$

- Let $d(k,K)$ be the optimal decision of the Household.
- Take labor supply $h=1$.
- In Equilibrium $k=K$.
- $d(.)$ should be consistent: $d(k,K)=D(K)$

Recursive Competitive equilibrium: General form

- RCE for Homogenous Agent models

$$\begin{aligned} v(z,s,S) &= \max_d \left\{ r(z,s,d,S,D) + \beta E\left[v(z',s',S') \,|z \right] \right\} \quad (6) \\ z' &= A(z) + \varepsilon' \\ s' &= B(z,s,d,S,D) \\ S' &= B(z,S,D,S,D) \\ D &= \mathbf{D}(z,S) \end{aligned}$$

- RCE consists of an individual's decision rule $d(.)$, an aggregate rule $D(.)$ and a value function $v(.)$ such that
  1. Given D, the value function $v(.)$ satisfies 6 and $d(.)$ is the associated decision rule.
  2. function D satisfies $\mathbf{D}(z,S) = d(z,s,S)$.

Markov Perfect Equilibrium

- Consider a dynamic model of duopoly.
- A market has two firms.
- Each firm recognizes that its output decision will affect the aggregate output and therefore influence the market price.
- Thus, we drop the assumption of price-taking behavior.
- The one-period return function of firm i is

$$R_{it} = p_{it} y_{it} - 0.5d \left(y_{i,t+1} - y_{it}\right)^2$$

- There is a demand curve:

$$p_t = A_0 - A_1 (y_{1t} + y_{2t}).$$

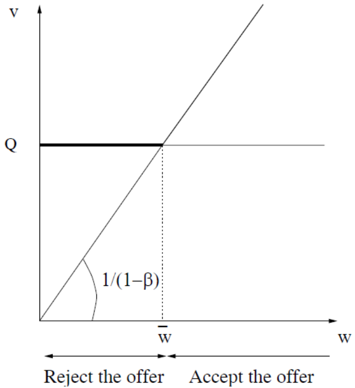
Applications: Job Search Model: Equilibrium Unemployment

- Let $V(w)$ be the total lifetime value accruing to a worker who has offer wage w and should decide whether to accept or reject the offer.
- Here value means the value of the objective function $\sum_{t=0}^{\infty} \beta^t y_t$ when the worker makes optimal decisions now and at all future points in time, where $y_t = w$ if he accepts and $y_t = c$ if he deicdes to be unemployed in period t.
- So

$$V(w) = \max \left\{ \frac{w}{1-\beta}, c + \beta \left[\int_0^B v(w') \, dF(w') \right] \right\}$$

Applications: Job Search Model: Equilibrium Unemployment

$$v(w) = \begin{cases} \frac{\bar{w}}{1-\beta} = c + \beta \left[\int_0^B v(w')\, dF(w') \right] & \text{if } w \leq \bar{w} \\ \frac{w}{1-\beta} & \text{if } w \geq \bar{w} \end{cases}$$



Applications: Job Search Model: Equilibrium Unemployment

- Evaluating $v(w)$ results in:

$$\bar{w} - c = h(\bar{w}) \text{ where } h(w) \equiv \frac{\beta}{1-\beta} \int_{w}^{B} (w' - w)\, dF(w')$$

- $h' < 0$ and $h'' > 0$

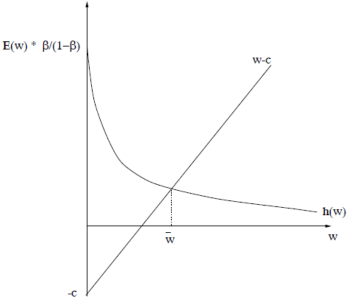


Figure 6.3.2: The reservation wage, $\overline{w}$, that satisfies $\overline{w} - c = [\beta/(1-\beta)] \int_{\overline{w}}^{B} (w' - \overline{w}) dF(w') \equiv h(\overline{w})$.

Applications: Job Search Model: Equilibrium Unemployment

- The McCall Job Search Model
  - If currently employed, the worker consumes his wage w, receiving utility $u(w)$
  - If currently unemployed, he
    - receives and consumes unemployment compensation c
    - receives an offer to start work next period at a wage w' drawn from a known distribution p
  - He can either accept or reject the offer
  - If he accepts the offer, he enters next period employed with wage w'
  - If he rejects the offer, he enters next period unemployed (Note that we do not allow for job search while employed)
  - Job Termination: When employed, he faces a constant probability a of becoming unemployed at the end of the period

Applications: Job Search Model: Equilibrium Unemployment

- Let $V(w)$ be the total lifetime value accruing to a worker who enters the current period employed with wage w
- U be the total lifetime value accruing to a worker who is unemployed this period
- Here value means the value of the objective function $\sum_{t=0}^{\infty} \beta^t u(y_t)$ when the worker makes optimal decisions now and at all future points in time.
- So

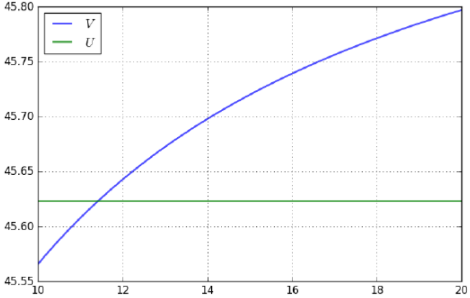
$$
\begin{aligned}
V(w) &= u(w) + \beta \left[(1-\alpha) V(w) + \alpha U \right] \\
U &= u(c) + \beta \sum_{w'} \max \left\{ V(w'), U \right\} p(w')
\end{aligned}
$$

Applications: Job Search Model: Equilibrium Unemployment

- Solution:

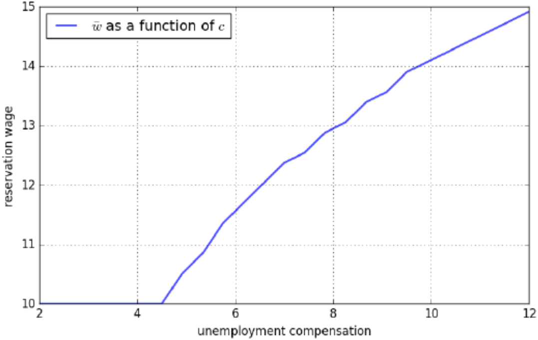
$$
\begin{aligned}
V_{n+1}(w) &= u(w) + \beta\left[(1-\alpha) V_n(w) + \alpha U_n\right] \\
U_{n+1} &= u(c) + \beta \sum_{w'} \max\left\{V_n(w'), U_n\right\} p(w')
\end{aligned}
$$

Applications: Job Search Model: Equilibrium Unemployment

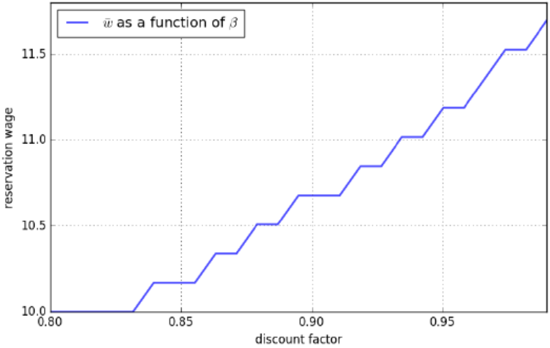


- Reservation Wage $\bar{w}$.

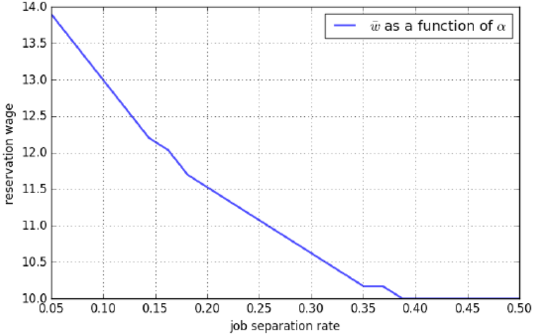
Applications: Job Search Model: Equilibrium Unemployment



Applications: Job Search Model: Equilibrium Unemployment



Applications: Job Search Model: Equilibrium Unemployment



Continuous Problem

- In Continuous form

$$\max_{\{c_t, k_{t+1}, i_t\}} \int_0^\infty e^{-\rho t} U(c_t)\, dt$$

$$\begin{aligned} \text{s.t. } c_t + i_t &= f(k_t) \\ \dot{k}_t &= i_t - \delta k_t \end{aligned}$$

- Write the Hamiltonian:

$$H = U(c_t) + \lambda \left(f(k_t) - \delta k_t - c_t \right)$$

- FOC:

$$\begin{aligned} H_c &= 0 \\ \rho\lambda - \dot{\lambda}_t &= H_k \end{aligned}$$

Continuous Time Bellman Equation

$$
\begin{aligned}
V\left(x_t\right) &= \max_{u_t \in U} \left\{ \Delta h\left(x_t, u_t\right) + \frac{1}{1+\Delta\rho} V\left(x_{t+\Delta}\right) \right\} \\
\text{s.t. } x_{t+\Delta} &= x_t + \Delta g\left(x_t, u_t\right)
\end{aligned}
$$

- In the limit:

$$
\rho V\left(x\right) = \max_{u_t \in U} \left\{ h\left(x, u\right) + V'\left(x\right) g\left(x, u\right) \right\}
$$

FOCs

$$
\begin{aligned}
0 &= h_u\left(x, u^*\left(x\right)\right) + V'\left(x\right) g_u\left(x, u^*\left(x\right)\right) \\
\rho V\left(x\right) &= h\left(x, u^*\left(x\right)\right) + V'\left(x\right) g\left(x, u^*\left(x\right)\right)
\end{aligned}
$$

References

- Cooley and Prescott (1995 in Frontiers of Business Cycle Research edited by Cooley) "Economic Growth and Business Cycles"
- Fernando Alvarez Lecture Notes
- RBC Models - A Toy Example by Ozdagli 2006
- Martin Ellison lecture notes chapter 1,2
- Caplin-Spulber (1987 QJE) "Menu Costs and the Neutrality of Money"
- Caplin-Leahy (1997 Econometrica) "Aggregation and Optimization with State-Dependent Pricing"
- Ljungqvist and Sargent (Recursive Macroeconomic Theory, Chapters 11-12)