Dynamic Programming: Theory and Numerical Analysis[1]

Hosein Joshaghani, Alireza Sepahsalari

Tehran Institute for Advanced Studies, Khatam University

March 2021

[1] We have used Adda and Cooper (2003) to prepare this slide.

Outline

Indirect Utility

- Consumer choice theory focuses on households that solve

$$V(I, p) = \max_{c} u(c)$$

$$\text{subject to} \quad pc = I$$

- FOCs are

$$\frac{u_j(c)}{p_j} = \lambda$$

- What happens if we give the consumer a bit more income?

$$\frac{u_j(c)}{p_j} = \lambda = V_I(I, p) \quad \text{for all } j$$

- It is in this sense that the indirect utility function summarizes the value of the households optimization problem and allows us to determine the marginal value of income without knowing more about consumption functions.

Firms

- Suppose that a firm must choose how many workers to hire at a wage of w given its stock of capital k and product price p.

$$\Pi(w, p, k) = \max_{l} pf(l, k) - wl$$

- Think of $\Pi(w, p, k)$ as an indirect profit function. It completely summarizes the value of the optimization problem of the firm given (w, p, k).
- What is the marginal value of allowing the firm some additional capital?

$$\frac{\partial \Pi(w, p, k)}{\partial k} = pf_k(l, k)$$

Firms

- Suppose that a firm must choose how many workers to hire at a wage of w given its stock of capital k and product price p.

$$\Pi(w, p, k) = \max_{l} pf(l, k) - wl$$

- Think of $\Pi(w, p, k)$ as an indirect profit function. It completely summarizes the value of the optimization problem of the firm given (w, p, k).
- What is the marginal value of allowing the firm some additional capital?

$$\frac{\partial \Pi(w, p, k)}{\partial k} = pf_k(l, k)$$

Firms

- Suppose that a firm must choose how many workers to hire at a wage of w given its stock of capital k and product price p.

$$\Pi(w, p, k) = \max_{l} pf(l, k) - wl$$

- Think of $\Pi(w, p, k)$ as an indirect profit function. It completely summarizes the value of the optimization problem of the firm given (w, p, k).
- What is the marginal value of allowing the firm some additional capital?

$$\frac{\partial \Pi(w, p, k)}{\partial k} = pf_k(l, k)$$

Firms

- Suppose that a firm must choose how many workers to hire at a wage of w given its stock of capital k and product price p.

$$\Pi(w, p, k) = \max_{l} pf(l, k) - wl$$

- Think of $\Pi(w, p, k)$ as an indirect profit function. It completely summarizes the value of the optimization problem of the firm given (w, p, k).
- What is the marginal value of allowing the firm some additional capital?

$$\frac{\partial \Pi(w, p, k)}{\partial k} = pf_k(l, k)$$

Dynamic Optimization: A Cake-Eating Example

- Very simple dynamic optimization problem
- Start with finite horizon and then discuss extensions to the infinite horizon
- Suppose a cake of size W_1.
- At each point of time, $t = 1, 2, \ldots, T$, you can eat some of the cake but must save the rest.
- Let c_t be your consumption in period t, and let $u(c_t)$ represent the flow of utility from this consumption.
- The utility function is not indexed by time: preferences are stationary.
- We can assume that $u(c_t)$ is real valued, differentiable, strictly increasing, and strictly concave.
- Further we can assume

$$\lim_{c \to 0} u'(c) \to \infty$$

Dynamic Optimization: A Cake-Eating Example

- Very simple dynamic optimization problem
- Start with finite horizon and then discuss extensions to the infinite horizon
- Suppose a cake of size W_1.
- At each point of time, $t = 1, 2, \ldots, T$, you can eat some of the cake but must save the rest.
- Let c_t be your consumption in period t, and let $u(c_t)$ represent the flow of utility from this consumption.
- The utility function is not indexed by time: preferences are stationary.
- We can assume that $u(c_t)$ is real valued, differentiable, strictly increasing, and strictly concave.
- Further we can assume

$$\lim_{c \to 0} u'(c) \to \infty$$

Dynamic Optimization: A Cake-Eating Example

- Very simple dynamic optimization problem
- Start with finite horizon and then discuss extensions to the infinite horizon
- Suppose a cake of size W_1.
- At each point of time, $t = 1, 2, \ldots, T$, you can eat some of the cake but must save the rest.
- Let c_t be your consumption in period t, and let $u(c_t)$ represent the flow of utility from this consumption.
- The utility function is not indexed by time: preferences are stationary.
- We can assume that $u(c_t)$ is real valued, differentiable, strictly increasing, and strictly concave.
- Further we can assume

$$\lim_{c \to 0} u'(c) \to \infty$$

Dynamic Optimization: A Cake-Eating Example

- Very simple dynamic optimization problem
- Start with finite horizon and then discuss extensions to the infinite horizon
- Suppose a cake of size W_1.
- At each point of time, $t = 1, 2, \ldots, T$, you can eat some of the cake but must save the rest.
- Let c_t be your consumption in period t, and let $u(c_t)$ represent the flow of utility from this consumption.
- The utility function is not indexed by time: preferences are stationary.
- We can assume that $u(c_t)$ is real valued, differentiable, strictly increasing, and strictly concave.
- Further we can assume

$$\lim_{c \to 0} u'(c) \to \infty$$

Dynamic Optimization: A Cake-Eating Example

- Very simple dynamic optimization problem
- Start with finite horizon and then discuss extensions to the infinite horizon
- Suppose a cake of size W_1.
- At each point of time, $t = 1, 2, \ldots, T$, you can eat some of the cake but must save the rest.
- Let c_t be your consumption in period t, and let $u(c_t)$ represent the flow of utility from this consumption.
- The utility function is not indexed by time: preferences are stationary.
- We can assume that $u(c_t)$ is real valued, differentiable, strictly increasing, and strictly concave.
- Further we can assume

$$\lim_{c \to 0} u'(c) \to \infty$$

Dynamic Optimization: A Cake-Eating Example

- Very simple dynamic optimization problem
- Start with finite horizon and then discuss extensions to the infinite horizon
- Suppose a cake of size W_1.
- At each point of time, $t = 1, 2, \ldots, T$, you can eat some of the cake but must save the rest.
- Let c_t be your consumption in period t, and let $u(c_t)$ represent the flow of utility from this consumption.
- The utility function is not indexed by time: preferences are stationary.
- We can assume that $u(c_t)$ is real valued, differentiable, strictly increasing, and strictly concave.
- Further we can assume

$$\lim_{c \to 0} u'(c) \to \infty$$

Dynamic Optimization: A Cake-Eating Example

- Very simple dynamic optimization problem
- Start with finite horizon and then discuss extensions to the infinite horizon
- Suppose a cake of size W_1.
- At each point of time, $t = 1, 2, \ldots, T$, you can eat some of the cake but must save the rest.
- Let c_t be your consumption in period t, and let $u(c_t)$ represent the flow of utility from this consumption.
- The utility function is not indexed by time: preferences are stationary.
- We can assume that $u(c_t)$ is real valued, differentiable, strictly increasing, and strictly concave.
- Further we can assume

$$\lim_{c \to 0} u'(c) \to \infty$$

Dynamic Optimization: A Cake-Eating Example

- Very simple dynamic optimization problem
- Start with finite horizon and then discuss extensions to the infinite horizon
- Suppose a cake of size W_1.
- At each point of time, $t = 1, 2, \ldots, T$, you can eat some of the cake but must save the rest.
- Let c_t be your consumption in period t, and let $u(c_t)$ represent the flow of utility from this consumption.
- The utility function is not indexed by time: preferences are stationary.
- We can assume that $u(c_t)$ is real valued, differentiable, strictly increasing, and strictly concave.
- Further we can assume

$$\lim_{c \to 0} u'(c) \to \infty$$

Cake-Eating Problem

- We could represent your lifetime utility by:

$$V_T(W_1) = \sum_{t=1}^{T} \beta^{t-1} u(c_t) \tag{1}$$

- For now, assume that the cake does not depreciate (spoil) or grow.
- Hence the evolution of the cake over time is governed by

$$W_{t+1} = W_t - c_t \tag{2}$$

Cake-Eating Problem

- We could represent your lifetime utility by:

$$V_T(W_1) = \sum_{t=1}^{T} \beta^{t-1} u(c_t) \tag{1}$$

- For now, assume that the cake does not depreciate (spoil) or grow.
- Hence the evolution of the cake over time is governed by

$$W_{t+1} = W_t - c_t \tag{2}$$

Cake-Eating Problem

- We could represent your lifetime utility by:

$$V_T(W_1) = \sum_{t=1}^{T} \beta^{t-1} u(c_t) \tag{1}$$

- For now, assume that the cake does not depreciate (spoil) or grow.
- Hence the evolution of the cake over time is governed by

$$W_{t+1} = W_t - c_t \tag{2}$$

Sequence Problem

- Consider the problem of

$$\max_{\{c_t\}_{t=1}^{T},\{W_t\}_{t=2}^{T}} \sum_{t=1}^{T} \beta^{t-1} u(c_t) \tag{3}$$

given W_1 and subject to $W_{T+1} \geq 0$, and the transition equation (2).

- Or alternatively

$$\sum_{t=1}^{T} c_t + W_{T+1} = W_1 \tag{4}$$

- This is a well-behaved problem as the objective is concave and continuous and the constraint set is compact. So there is a solution.
- FOCs are

$$\beta^{t-1} u'(c_t) = \lambda \tag{5}$$

$$\lambda = \phi \tag{6}$$

Sequence Problem

- Consider the problem of

$$\max_{\{c_t\}_{t=1}^T, \{W_t\}_{t=2}^T} \sum_{t=1}^{T} \beta^{t-1} u(c_t) \tag{3}$$

given W_1 and subject to $W_{T+1} \geq 0$, and the transition equation (2).

- Or alternatively

$$\sum_{t=1}^{T} c_t + W_{T+1} = W_1 \tag{4}$$

- This is a well-behaved problem as the objective is concave and continuous and the constraint set is compact. So there is a solution.
- FOCs are

$$\beta^{t-1} u'(c_t) = \lambda \tag{5}$$

$$\lambda = \phi \tag{6}$$

Sequence Problem

- Consider the problem of

$$\max_{\{c_t\}_{t=1}^{T},\{W_t\}_{t=2}^{T}} \sum_{t=1}^{T} \beta^{t-1} u(c_t) \tag{3}$$

given W_1 and subject to $W_{T+1} \geq 0$, and the transition equation (2).

- Or alternatively

$$\sum_{t=1}^{T} c_t + W_{T+1} = W_1 \tag{4}$$

- This is a well-behaved problem as the objective is concave and continuous and the constraint set is compact. So there is a solution.
- FOCs are

$$\beta^{t-1} u'(c_t) = \lambda \tag{5}$$

$$\lambda = \phi \tag{6}$$

Sequence Problem

- Consider the problem of

$$\max_{\{c_t\}_{t=1}^{T},\{W_t\}_{t=2}^{T}} \sum_{t=1}^{T} \beta^{t-1} u(c_t) \tag{3}$$

given W_1 and subject to $W_{T+1} \geq 0$, and the transition equation (2).

- Or alternatively

$$\sum_{t=1}^{T} c_t + W_{T+1} = W_1 \tag{4}$$

- This is a well-behaved problem as the objective is concave and continuous and the constraint set is compact. So there is a solution.
- FOCs are

$$\beta^{t-1} u'(c_t) = \lambda \tag{5}$$

$$\lambda = \phi \tag{6}$$

Euler equation

- Combining the equations, we obtain an expression that links consumption across any two periods:

$$u'(c_t) = \beta u'(c_{t+1}) \tag{7}$$

- Euler equation: A necessary condition of optimality for any t: if it is violated, the agent can do better by adjusting c_t and c_{t+1}.
- Is this enough? Not quite.
- Formally, this involves showing that the nonnegativity constraint on W_{T+1} must bind: $\lambda = \phi > 0$.
- In effect the problem is pinned down by an initial condition (W_1 is given) and by a terminal condition ($W_{T+1} = 0$). The set of $(T - 1)$ Euler equations and (2) then determine the time path of consumption.

Euler equation

- Combining the equations, we obtain an expression that links consumption across any two periods:

$$u'(c_t) = \beta u'(c_{t+1}) \tag{7}$$

- Euler equation: A necessary condition of optimality for any t: if it is violated, the agent can do better by adjusting c_t and c_{t+1}.
- Is this enough? Not quite.
- Formally, this involves showing that the nonnegativity constraint on W_{T+1} must bind: $\lambda = \phi > 0$.
- In effect the problem is pinned down by an initial condition (W_1 is given) and by a terminal condition ($W_{T+1} = 0$). The set of $(T-1)$ Euler equations and (2) then determine the time path of consumption.

Euler equation

- Combining the equations, we obtain an expression that links consumption across any two periods:

$$u'(c_t) = \beta u'(c_{t+1}) \tag{7}$$

- Euler equation: A necessary condition of optimality for any t: if it is violated, the agent can do better by adjusting c_t and c_{t+1}.
- Is this enough? Not quite.
- Formally, this involves showing that the nonnegativity constraint on W_{T+1} must bind: $\lambda = \phi > 0$.
- In effect the problem is pinned down by an initial condition (W_1 is given) and by a terminal condition ($W_{T+1} = 0$). The set of $(T-1)$ Euler equations and (2) then determine the time path of consumption.

Euler equation

- Combining the equations, we obtain an expression that links consumption across any two periods:

$$u'(c_t) = \beta u'(c_{t+1}) \tag{7}$$

- Euler equation: A necessary condition of optimality for any t: if it is violated, the agent can do better by adjusting c_t and c_{t+1}.
- Is this enough? Not quite.
- Formally, this involves showing that the nonnegativity constraint on W_{T+1} must bind: $\lambda = \phi > 0$.
- In effect the problem is pinned down by an initial condition (W_1 is given) and by a terminal condition ($W_{T+1} = 0$). The set of $(T-1)$ Euler equations and (2) then determine the time path of consumption.

Euler equation

- Combining the equations, we obtain an expression that links consumption across any two periods:

$$u'(c_t) = \beta u'(c_{t+1}) \tag{7}$$

- Euler equation: A necessary condition of optimality for any t: if it is violated, the agent can do better by adjusting c_t and c_{t+1}.
- Is this enough? Not quite.
- Formally, this involves showing that the nonnegativity constraint on W_{T+1} must bind: $\lambda = \phi > 0$.
- In effect the problem is pinned down by an initial condition (W_1 is given) and by a terminal condition ($W_{T+1} = 0$). The set of $(T-1)$ Euler equations and (2) then determine the time path of consumption.

Value Function

- Let the solution to this problem be denoted by $V_T(W_1)$, where T is the horizon of the problem and W_1 is the initial size of the cake.
- This is a **value function**.
- A slight increase in the size of the cake leads to an increase in lifetime utility equal to the marginal utility in any period.

$$V'_T(W_1) = \lambda = \beta^{t-1} u'(c_t) \tag{8}$$

Value Function

- Let the solution to this problem be denoted by $V_T(W_1)$, where T is the horizon of the problem and W_1 is the initial size of the cake.
- This is a **value function**.
- A slight increase in the size of the cake leads to an increase in lifetime utility equal to the marginal utility in any period.

$$V'_T(W_1) = \lambda = \beta^{t-1} u'(c_t) \tag{8}$$

Value Function

- Let the solution to this problem be denoted by $V_T(W_1)$, where T is the horizon of the problem and W_1 is the initial size of the cake.
- This is a **value function**.
- A slight increase in the size of the cake leads to an increase in lifetime utility equal to the marginal utility in any period.

$$V'_T(W_1) = \lambda = \beta^{t-1} u'(c_t) \tag{8}$$

Dynamic Programming Approach

- Suppose that we change the problem slightly: we add a period 0 and give an initial cake of size W0.
- But, having done all of the hard work with the T period problem, it would be nice not to have to do it again.
- Dynamic programming approach essentially converts a (arbitrary) T-period problem into a two-period problem with the appropriate rewriting of the objective function.
- Given W_0, consider the problem of

$$\max_{c_0} u_0 + \beta V_T(W1) \tag{9}$$

where $W_1 = W_0 - c_0$; W_0 given.

- Once c_0 and thus W_1 are determined, the value of the problem from then on is given by $V_T(W_1)$.

Dynamic Programming Approach

- Suppose that we change the problem slightly: we add a period 0 and give an initial cake of size W0.
- But, having done all of the hard work with the T period problem, it would be nice not to have to do it again.
- Dynamic programming approach essentially converts a (arbitrary) T-period problem into a two-period problem with the appropriate rewriting of the objective function.
- Given W_0, consider the problem of

$$\max_{c_0} u_0 + \beta V_T(W1) \tag{9}$$

where $W_1 = W_0 - c_0$; W_0 given.

- Once c_0 and thus W_1 are determined, the value of the problem from then on is given by $V_T(W_1)$.

Dynamic Programming Approach

- Suppose that we change the problem slightly: we add a period 0 and give an initial cake of size W0.
- But, having done all of the hard work with the T period problem, it would be nice not to have to do it again.
- Dynamic programming approach essentially converts a (arbitrary) T-period problem into a two-period problem with the appropriate rewriting of the objective function.
- Given W_0, consider the problem of

$$\max_{c_0} u_0 + \beta V_T(W1) \tag{9}$$

where $W_1 = W_0 - c_0$; W_0 given.

- Once c_0 and thus W_1 are determined, the value of the problem from then on is given by $V_T(W_1)$.

Dynamic Programming Approach

- Suppose that we change the problem slightly: we add a period 0 and give an initial cake of size W0.
- But, having done all of the hard work with the T period problem, it would be nice not to have to do it again.
- Dynamic programming approach essentially converts a (arbitrary) T-period problem into a two-period problem with the appropriate rewriting of the objective function.
- Given W_0, consider the problem of

$$\max_{c_0} u_0 + \beta V_T(W1) \tag{9}$$

where $W_1 = W_0 - c_0$; W_0 given.

- Once c_0 and thus W_1 are determined, the value of the problem from then on is given by $V_T(W_1)$.

Dynamic Programming Approach

- Suppose that we change the problem slightly: we add a period 0 and give an initial cake of size W0.
- But, having done all of the hard work with the T period problem, it would be nice not to have to do it again.
- Dynamic programming approach essentially converts a (arbitrary) T-period problem into a two-period problem with the appropriate rewriting of the objective function.
- Given W_0, consider the problem of

$$\max_{c_0} u_0 + \beta V_T(W1) \tag{9}$$

where $W_1 = W_0 - c_0$; W_0 given.

- Once c_0 and thus W_1 are determined, the value of the problem from then on is given by $V_T(W_1)$.

Principle of Optimality

- For the purposes of the dynamic programming problem, it does not matter how the cake will be consumed after the initial period.
- All that is important is that the agent will be acting optimally and thus generating utility given by $V_T(W_1)$.
- This is the principle of optimality, due to Richard Bellman, at work.
- FOC at $t = 0$:

$$u'(c_0) = \beta V'_T(W_1) \tag{10}$$

- From the T-period sequence problem we know that

$$V'_T(W_1) = u'(c_1) = \beta^t u'(c_{t+1}) \tag{11}$$

- Therefore, the necessary condition is the same EE.
- We were able to make the problem look simple by pretending that we actually know $V_T(W_1)$.

Principle of Optimality

- For the purposes of the dynamic programming problem, it does not matter how the cake will be consumed after the initial period.
- All that is important is that the agent will be acting optimally and thus generating utility given by $V_T(W_1)$.
- This is the principle of optimality, due to Richard Bellman, at work.
- FOC at $t = 0$:

$$u'(c_0) = \beta V'_T(W_1) \tag{10}$$

- From the T-period sequence problem we know that

$$V'_T(W_1) = u'(c_1) = \beta^t u'(c_{t+1}) \tag{11}$$

- Therefore, the necessary condition is the same EE.
- We were able to make the problem look simple by pretending that we actually know $V_T(W_1)$.

Principle of Optimality

- For the purposes of the dynamic programming problem, it does not matter how the cake will be consumed after the initial period.
- All that is important is that the agent will be acting optimally and thus generating utility given by $V_T(W_1)$.
- This is the principle of optimality, due to Richard Bellman, at work.
- FOC at $t = 0$:

$$u'(c_0) = \beta V'_T(W_1) \tag{10}$$

- From the T-period sequence problem we know that

$$V'_T(W_1) = u'(c_1) = \beta^t u'(c_{t+1}) \tag{11}$$

- Therefore, the necessary condition is the same EE.
- We were able to make the problem look simple by pretending that we actually know $V_T(W_1)$.

Principle of Optimality

- For the purposes of the dynamic programming problem, it does not matter how the cake will be consumed after the initial period.
- All that is important is that the agent will be acting optimally and thus generating utility given by $V_T(W_1)$.
- This is the principle of optimality, due to Richard Bellman, at work.
- FOC at $t = 0$:

$$u'(c_0) = \beta V'_T(W_1) \tag{10}$$

- From the T-period sequence problem we know that

$$V'_T(W_1) = u'(c_1) = \beta^t u'(c_{t+1}) \tag{11}$$

- Therefore, the necessary condition is the same EE.
- We were able to make the problem look simple by pretending that we actually know $V_T(W_1)$.

Principle of Optimality

- For the purposes of the dynamic programming problem, it does not matter how the cake will be consumed after the initial period.
- All that is important is that the agent will be acting optimally and thus generating utility given by $V_T(W_1)$.
- This is the principle of optimality, due to Richard Bellman, at work.
- FOC at $t = 0$:

$$u'(c_0) = \beta V'_T(W_1) \tag{10}$$

- From the T-period sequence problem we know that

$$V'_T(W_1) = u'(c_1) = \beta^t u'(c_{t+1}) \tag{11}$$

- Therefore, the necessary condition is the same EE.
- We were able to make the problem look simple by pretending that we actually know $V_T(W_1)$.

Principle of Optimality

- For the purposes of the dynamic programming problem, it does not matter how the cake will be consumed after the initial period.
- All that is important is that the agent will be acting optimally and thus generating utility given by $V_T(W_1)$.
- This is the principle of optimality, due to Richard Bellman, at work.
- FOC at $t = 0$:

$$u'(c_0) = \beta V_T'(W_1) \tag{10}$$

- From the T-period sequence problem we know that

$$V_T'(W_1) = u'(c_1) = \beta^t u'(c_{t+1}) \tag{11}$$

- Therefore, the necessary condition is the same EE.
- We were able to make the problem look simple by pretending that we actually know $V_T(W_1)$.

Principle of Optimality

- For the purposes of the dynamic programming problem, it does not matter how the cake will be consumed after the initial period.
- All that is important is that the agent will be acting optimally and thus generating utility given by $V_T(W_1)$.
- This is the principle of optimality, due to Richard Bellman, at work.
- FOC at $t = 0$:

$$u'(c_0) = \beta V'_T(W_1) \tag{10}$$

- From the T-period sequence problem we know that

$$V'_T(W_1) = u'(c_1) = \beta^t u'(c_{t+1}) \tag{11}$$

- Therefore, the necessary condition is the same EE.
- We were able to make the problem look simple by pretending that we actually know $V_T(W_1)$.

Example

- Asssume $u(C) = \ln(c)$ then find $V_T(W_1)$. Hint: Start from $V_1(W_1)$ then find $V_2(W_1)$ and go ahead.

Infinite Horizon

- As before, one can consider solving the infinite horizon sequence problem given by

$$\max_{\{c_t\}_{t=1}^{\infty},\{W_t\}_{t=2}^{\infty}} \sum_{t=1}^{\infty} \beta^{t-1} u(c_t) \tag{12}$$

along with similar transition equations $W_{t+1} = W_t - c_t$.

- In specifying this as a dynamic programming problem, we write

$$V(W) = \max_{c \in [0,W]} u(c) + \beta V(W - c) \tag{13}$$

- The **state** completely summarizes all information from the past that is needed for the forward-looking problem: size of the cake, W.
- The **control** variable is the one that is being chosen: consumption, c.
- **Transition equation** pins down the relation between state tomorrow, state today and the control today, $W' = W - c$.

Infinite Horizon

- As before, one can consider solving the infinite horizon sequence problem given by

$$\max_{\{c_t\}_{t=1}^{\infty},\{W_t\}_{t=2}^{\infty}} \sum_{t=1}^{\infty} \beta^{t-1} u(c_t) \tag{12}$$

along with similar transition equations $W_{t+1} = W_t - c_t$.

- In specifying this as a dynamic programming problem, we write

$$V(W) = \max_{c \in [0,W]} u(c) + \beta V(W - c) \tag{13}$$

- The **state** completely summarizes all information from the past that is needed for the forward-looking problem: size of the cake, W.
- The **control** variable is the one that is being chosen: consumption, c.
- **Transition equation** pins down the relation between state tomorrow, state today and the control today, $W' = W - c$.

Infinite Horizon

- As before, one can consider solving the infinite horizon sequence problem given by

$$\max_{\{c_t\}_{t=1}^{\infty},\{W_t\}_{t=2}^{\infty}} \sum_{t=1}^{\infty} \beta^{t-1} u(c_t) \tag{12}$$

along with similar transition equations $W_{t+1} = W_t - c_t$.

- In specifying this as a dynamic programming problem, we write

$$V(W) = \max_{c \in [0,W]} u(c) + \beta V(W - c) \tag{13}$$

- The **state** completely summarizes all information from the past that is needed for the forward-looking problem: size of the cake, W.
- The **control** variable is the one that is being chosen: consumption, c.
- **Transition equation** pins down the relation between state tomorrow, state today and the control today, $W' = W - c$.

Infinite Horizon

- As before, one can consider solving the infinite horizon sequence problem given by

$$\max_{\{c_t\}_{t=1}^{\infty}, \{W_t\}_{t=2}^{\infty}} \sum_{t=1}^{\infty} \beta^{t-1} u(c_t) \tag{12}$$

  along with similar transition equations $W_{t+1} = W_t - c_t$.
- In specifying this as a dynamic programming problem, we write

$$V(W) = \max_{c \in [0, W]} u(c) + \beta V(W - c) \tag{13}$$

- The **state** completely summarizes all information from the past that is needed for the forward-looking problem: size of the cake, W.
- The **control** variable is the one that is being chosen: consumption, c.
- **Transition equation** pins down the relation between state tomorrow, state today and the control today, $W' = W - c$.

Infinite Horizon

- As before, one can consider solving the infinite horizon sequence problem given by

$$\max_{\{c_t\}_{t=1}^{\infty},\{W_t\}_{t=2}^{\infty}} \sum_{t=1}^{\infty} \beta^{t-1} u(c_t) \tag{12}$$

along with similar transition equations $W_{t+1} = W_t - c_t$.

- In specifying this as a dynamic programming problem, we write

$$V(W) = \max_{c \in [0,W]} u(c) + \beta V(W - c) \tag{13}$$

- The **state** completely summarizes all information from the past that is needed for the forward-looking problem: size of the cake, W.
- The **control** variable is the one that is being chosen: consumption, c.
- **Transition equation** pins down the relation between state tomorrow, state today and the control today, $W' = W - c$.

Functional Equation

- Alternatively, we can specify the problem so that instead of choosing today's consumption we choose tomorrow's state:

$$V(W) = \max_{W' \in [0,W]} u(W - W') + \beta V(W') \tag{14}$$

- Note that time itself does not enter into Bellman's equation: we can express all relations without an indication of time. This is the essence of **stationarity**.
- All information about the past that bears on current and future decisions is summarized by W.
- It is an art to pick your state variable and set up your BE.
- FOC is

$$u'(c) = \beta V'(W') \tag{15}$$

- What is the derivative of the value function? $V'(W) = u'(c)$. Why?

Functional Equation

- Alternatively, we can specify the problem so that instead of choosing today's consumption we choose tomorrow's state:

$$V(W) = \max_{W' \in [0,W]} u(W - W') + \beta V(W') \tag{14}$$

- Note that time itself does not enter into Bellman's equation: we can express all relations without an indication of time. This is the essence of **stationarity**.
- All information about the past that bears on current and future decisions is summarized by W.
- It is an art to pick your state variable and set up your BE.
- FOC is

$$u'(c) = \beta V'(W') \tag{15}$$

- What is the derivative of the value function? $V'(W) = u'(c)$. Why?

Functional Equation

- Alternatively, we can specify the problem so that instead of choosing today's consumption we choose tomorrow's state:

$$V(W) = \max_{W' \in [0,W]} u(W - W') + \beta V(W') \tag{14}$$

- Note that time itself does not enter into Bellman's equation: we can express all relations without an indication of time. This is the essence of **stationarity**.
- All information about the past that bears on current and future decisions is summarized by W.
- It is an art to pick your state variable and set up your BE.
- FOC is

$$u'(c) = \beta V'(W') \tag{15}$$

- What is the derivative of the value function? $V'(W) = u'(c)$. Why?

Functional Equation

- Alternatively, we can specify the problem so that instead of choosing today's consumption we choose tomorrow's state:

$$V(W) = \max_{W' \in [0,W]} u(W - W') + \beta V(W') \tag{14}$$

- Note that time itself does not enter into Bellman's equation: we can express all relations without an indication of time. This is the essence of **stationarity**.
- All information about the past that bears on current and future decisions is summarized by W.
- It is an art to pick your state variable and set up your BE.
- FOC is

$$u'(c) = \beta V'(W') \tag{15}$$

- What is the derivative of the value function? $V'(W) = u'(c)$. Why?

Functional Equation

- Alternatively, we can specify the problem so that instead of choosing today's consumption we choose tomorrow's state:

$$V(W) = \max_{W' \in [0,W]} u(W - W') + \beta V(W') \tag{14}$$

- Note that time itself does not enter into Bellman's equation: we can express all relations without an indication of time. This is the essence of **stationarity**.
- All information about the past that bears on current and future decisions is summarized by W.
- It is an art to pick your state variable and set up your BE.
- FOC is

$$u'(c) = \beta V'(W') \tag{15}$$

- What is the derivative of the value function? $V'(W) = u'(c)$. Why?

Functional Equation

- Alternatively, we can specify the problem so that instead of choosing today's consumption we choose tomorrow's state:

$$V(W) = \max_{W' \in [0,W]} u(W - W') + \beta V(W') \tag{14}$$

- Note that time itself does not enter into Bellman's equation: we can express all relations without an indication of time. This is the essence of **stationarity**.
- All information about the past that bears on current and future decisions is summarized by W.
- It is an art to pick your state variable and set up your BE.
- FOC is

$$u'(c) = \beta V'(W') \tag{15}$$

- What is the derivative of the value function? $V'(W) = u'(c)$. Why?

Functional Equation

- Alternatively, we can specify the problem so that instead of choosing today's consumption we choose tomorrow's state:

$$V(W) = \max_{W' \in [0, W]} u(W - W') + \beta V(W') \tag{14}$$

- Note that time itself does not enter into Bellman's equation: we can express all relations without an indication of time. This is the essence of **stationarity**.
- All information about the past that bears on current and future decisions is summarized by W.
- It is an art to pick your state variable and set up your BE.
- FOC is

$$u'(c) = \beta V'(W') \tag{15}$$

- What is the derivative of the value function? $V'(W) = u'(c)$. Why?

Functional Equation

- Alternatively, we can specify the problem so that instead of choosing today's consumption we choose tomorrow's state:

$$V(W) = \max_{W' \in [0, W]} u(W - W') + \beta V(W') \tag{14}$$

- Note that time itself does not enter into Bellman's equation: we can express all relations without an indication of time. This is the essence of **stationarity**.
- All information about the past that bears on current and future decisions is summarized by W.
- It is an art to pick your state variable and set up your BE.
- FOC is

$$u'(c) = \beta V'(W') \tag{15}$$

- What is the derivative of the value function? $V'(W) = u'(c)$. Why?

Policy Function

- Using the FOC, (15), and the reuslt of the envelope theorm we have:

$$u'(c) = \beta u'(c') \tag{16}$$

  the similar Euler equaiton necessary equation.

- The link from the level of consumption and next period's cake (the controls from the different formulations) to the size of the cake (the state) is given by the policy function:

$$\begin{aligned} c &= \phi(W) \\ W' &= W - \phi(W) \end{aligned}$$

- Substituting these into the Euler equation we have:

$$u'(\phi(W)) = \beta u'(\phi(W - \phi(W))) \tag{17}$$

Policy Function

- Using the FOC, (15), and the reuslt of the envelope theorm we have:

$$u'(c) = \beta u'(c') \tag{16}$$

  the similar Euler equaiton necessary equation.

- The link from the level of consumption and next period's cake (the controls from the different formulations) to the size of the cake (the state) is given by the policy function:

$$\begin{aligned} c &= \phi(W) \\ W' &= W - \phi(W) \end{aligned}$$

- Substituting these into the Euler equation we have:

$$u'(\phi(W)) = \beta u'(\phi(W - \phi(W))) \tag{17}$$

Policy Function

- Using the FOC, (15), and the reuslt of the envelope theorm we have:

$$u'(c) = \beta u'(c') \tag{16}$$

  the similar Euler equaiton necessary equation.

- The link from the level of consumption and next period's cake (the controls from the different formulations) to the size of the cake (the state) is given by the policy function:

$$\begin{aligned} c &= \phi(W) \\ W' &= W - \phi(W) \end{aligned}$$

- Substituting these into the Euler equation we have:

$$u'(\phi(W)) = \beta u'(\phi(W - \phi(W))) \tag{17}$$

Example with Closed Form Solution

- In general, it is not actually possible to find closed form solutions for the value function and the resulting policy functions.
- Assume that $u(C) = \ln(c)$ then find $V(W)$ and $\phi(W)$. Hint: Guess that $V(W) = A + B\ln(W)$.
- Solution:

$$
\begin{aligned}
c =& \phi(W) = W(1-\beta) \\
W =& \beta W
\end{aligned}
$$

Example with Closed Form Solution

- In general, it is not actually possible to find closed form solutions for the value function and the resulting policy functions.
- Assume that $u(C) = \ln(c)$ then find $V(W)$ and $\phi(W)$. Hint: Guess that $V(W) = A + B\ln(W)$.
- Solution:

$$
\begin{aligned}
c &= \phi(W) = W(1-\beta) \\
W' &= \beta W
\end{aligned}
$$

Example with Closed Form Solution

- In general, it is not actually possible to find closed form solutions for the value function and the resulting policy functions.
- Assume that $u(C) = \ln(c)$ then find $V(W)$ and $\phi(W)$. Hint: Guess that $V(W) = A + B\ln(W)$.
- Solution:

$$c = \phi(W) = W(1-\beta)$$
$$W' = \beta W$$

Example with Closed Form Solution

- In general, it is not actually possible to find closed form solutions for the value function and the resulting policy functions.
- Assume that $u(C) = \ln(c)$ then find $V(W)$ and $\phi(W)$. Hint: Guess that $V(W) = A + B\ln(W)$.
- Solution:

$$c = \phi(W) = W(1-\beta)$$
$$W' = \beta W$$

Example with Closed Form Solution

- In general, it is not actually possible to find closed form solutions for the value function and the resulting policy functions.
- Assume that $u(C) = \ln(c)$ then find $V(W)$ and $\phi(W)$. Hint: Guess that $V(W) = A + B\ln(W)$.
- Solution:

$$\begin{aligned} c =& \phi(W) = W(1-\beta) \\ W' =& \beta W \end{aligned}$$

Cake Produces More Cake

- Now assume that the transition equation is modified so that

$$W_{t+1} = \rho W_t - c_t$$

.

- where $\rho > 0$ represents a return from holding cake inventories.
- Solve the T-period problem with this storage technology.
- Interpret the first-order conditions.
- How would you formulate the Bellman equation for the infinite horizon version of this problem?
- Does the size of ρ matter in this discussion? Explain.

Cake Produces More Cake

- Now assume that the transition equation is modified so that

$$W_{t+1} = \rho W_t - c_t$$

.

- where $\rho > 0$ represents a return from holding cake inventories.
- Solve the T-period problem with this storage technology.
- Interpret the first-order conditions.
- How would you formulate the Bellman equation for the infinite horizon version of this problem?
- Does the size of ρ matter in this discussion? Explain.

Cake Produces More Cake

- Now assume that the transition equation is modified so that

$$W_{t+1} = \rho W_t - c_t$$

.

- where $\rho > 0$ represents a return from holding cake inventories.
- Solve the T-period problem with this storage technology.
- Interpret the first-order conditions.
- How would you formulate the Bellman equation for the infinite horizon version of this problem?
- Does the size of ρ matter in this discussion? Explain.

Cake Produces More Cake

- Now assume that the transition equation is modified so that

$$W_{t+1} = \rho W_t - c_t$$

.

- where $\rho > 0$ represents a return from holding cake inventories.
- Solve the T-period problem with this storage technology.
- Interpret the first-order conditions.
- How would you formulate the Bellman equation for the infinite horizon version of this problem?
- Does the size of ρ matter in this discussion? Explain.

Cake Produces More Cake

- Now assume that the transition equation is modified so that

$$W_{t+1} = \rho W_t - c_t$$

.

- where $\rho > 0$ represents a return from holding cake inventories.
- Solve the T-period problem with this storage technology.
- Interpret the first-order conditions.
- How would you formulate the Bellman equation for the infinite horizon version of this problem?
- Does the size of ρ matter in this discussion? Explain.

Cake Produces More Cake

- Now assume that the transition equation is modified so that

$$W_{t+1} = \rho W_t - c_t$$

.

- where $\rho > 0$ represents a return from holding cake inventories.
- Solve the T-period problem with this storage technology.
- Interpret the first-order conditions.
- How would you formulate the Bellman equation for the infinite horizon version of this problem?
- Does the size of ρ matter in this discussion? Explain.

Taste Shocks

- To allow for variations of appetite, suppose that utility over consumption is given by

$$\epsilon u(c)$$

- In problems with stochastic elements, it is critical to be precise about the timing of events.
- Does the optimizing agent know the current shocks when making a decision?
- Assume that the taste shock takes on only two values:

$$\epsilon \in \{\epsilon_l, \epsilon_h\}$$

- Further we can assume that the taste shock follows a first-order Markov process:

$$\pi_{ij} = Prob(\epsilon' = \epsilon_j | \epsilon = \epsilon_i)$$

This matrix is called a **transition matrix**.

Taste Shocks

- To allow for variations of appetite, suppose that utility over consumption is given by

$$\epsilon u(c)$$

- In problems with stochastic elements, it is critical to be precise about the timing of events.
- Does the optimizing agent know the current shocks when making a decision?
- Assume that the taste shock takes on only two values:

$$\epsilon \in \{\epsilon_l, \epsilon_h\}$$

- Further we can assume that the taste shock follows a first-order Markov process:

$$\pi_{ij} = Prob(\epsilon' = \epsilon_j | \epsilon = \epsilon_i)$$

This matrix is called a **transition matrix**.

Taste Shocks

- To allow for variations of appetite, suppose that utility over consumption is given by

$$\epsilon u(c)$$

- In problems with stochastic elements, it is critical to be precise about the timing of events.
- Does the optimizing agent know the current shocks when making a decision?
- Assume that the taste shock takes on only two values:

$$\epsilon \in \{\epsilon_l, \epsilon_h\}$$

- Further we can assume that the taste shock follows a first-order Markov process:

$$\pi_{ij} = Prob(\epsilon' = \epsilon_j | \epsilon = \epsilon_i)$$

This matrix is called a **transition matrix**.

Taste Shocks

- To allow for variations of appetite, suppose that utility over consumption is given by

$$\epsilon u(c)$$

- In problems with stochastic elements, it is critical to be precise about the timing of events.
- Does the optimizing agent know the current shocks when making a decision?
- Assume that the taste shock takes on only two values:

$$\epsilon \in \{\epsilon_l, \epsilon_h\}$$

- Further we can assume that the taste shock follows a first-order Markov process:

$$\pi_{ij} = Prob(\epsilon' = \epsilon_j | \epsilon = \epsilon_i)$$

This matrix is called a **transition matrix**.

Taste Shocks

- To allow for variations of appetite, suppose that utility over consumption is given by

$$\epsilon u(c)$$

- In problems with stochastic elements, it is critical to be precise about the timing of events.
- Does the optimizing agent know the current shocks when making a decision?
- Assume that the taste shock takes on only two values:

$$\epsilon \in \{\epsilon_l, \epsilon_h\}$$

- Further we can assume that the taste shock follows a first-order Markov process:

$$\pi_{ij} = Prob(\epsilon' = \epsilon_j | \epsilon = \epsilon_i)$$

This matrix is called a **transition matrix**.

BE with Taste Shocks

- In the nonstochastic problem, the state was simply the size of the cake: it provided all the information the agent needed to make a choice.
- When taste shocks are introduced, the agent needs to take this factor into account as well.
- The BE is written:

$$V(W.\epsilon) = \max_{W'} \epsilon u(W' - W) + \beta\, E_{\epsilon'|\epsilon}\, V(W', \epsilon')$$

- The FOC for this problem is

$$\begin{aligned} \epsilon u'(W - W') &= \beta\, E_{\epsilon'|\epsilon}\, V_1(W', \epsilon') \\ &= \beta\, E_{\epsilon'|\epsilon}\, u'(W' - W'') \end{aligned}$$

which is the **stochastic Euler equation**.

BE with Taste Shocks

- In the nonstochastic problem, the state was simply the size of the cake: it provided all the information the agent needed to make a choice.
- When taste shocks are introduced, the agent needs to take this factor into account as well.
- The BE is written:

$$V(W.\epsilon) = \max_{W'} \epsilon u(W' - W) + \beta \, E_{\epsilon'|\epsilon} V(W', \epsilon')$$

- The FOC for this problem is

$$\begin{aligned} \epsilon u'(W - W') &= \beta \, E_{\epsilon'|\epsilon} V_1(W', \epsilon') \\ &= \beta \, E_{\epsilon'|\epsilon} u'(W' - W'') \end{aligned}$$

which is the **stochastic Euler equation**.

BE with Taste Shocks

- In the nonstochastic problem, the state was simply the size of the cake: it provided all the information the agent needed to make a choice.
- When taste shocks are introduced, the agent needs to take this factor into account as well.
- The BE is written:

$$V(W.\epsilon) = \max_{W'} \epsilon u(W' - W) + \beta E_{\epsilon'|\epsilon} V(W', \epsilon')$$

- The FOC for this problem is

$$\begin{aligned} \epsilon u'(W - W') &= \beta E_{\epsilon'|\epsilon} V_1(W', \epsilon') \\ &= \beta E_{\epsilon'|\epsilon} u'(W' - W'') \end{aligned}$$

which is the **stochastic Euler equation**.

BE with Taste Shocks

- In the nonstochastic problem, the state was simply the size of the cake: it provided all the information the agent needed to make a choice.
- When taste shocks are introduced, the agent needs to take this factor into account as well.
- The BE is written:

$$V(W.\epsilon) = \max_{W'} \epsilon u(W' - W) + \beta\, E_{\epsilon'|\epsilon}\, V(W', \epsilon')$$

- The FOC for this problem is

$$\begin{aligned} \epsilon u'(W - W') =& \beta\, E_{\epsilon'|\epsilon}\, V_1(W', \epsilon') \\ =& \beta\, E_{\epsilon'|\epsilon}\, u'(W' - W'') \end{aligned}$$

which is the **stochastic Euler equation**.

Optimal Stopping Problems

-

Outline

Indirect

- Link between the basic theory of dynamic programming and the empirical analysis of dynamic optimization problems.
- Value Function Iteration
- Policy Function Iterations
- Projection Methods

Indirect

- Link between the basic theory of dynamic programming and the empirical analysis of dynamic optimization problems.
- Value Function Iteration
- Policy Function Iterations
- Projection Methods

Indirect

- Link between the basic theory of dynamic programming and the empirical analysis of dynamic optimization problems.
- Value Function Iteration
- Policy Function Iterations
- Projection Methods

Indirect

- Link between the basic theory of dynamic programming and the empirical analysis of dynamic optimization problems.
- Value Function Iteration
- Policy Function Iterations
- Projection Methods

Stochastic Cake-Eating Problem

- Start with the stochastic cake-eating problem defined by

$$V(W, y) = \max_{0 \leq c \leq W+y} u(c) + \beta\, E_{y'|y}\, V(W', y') \tag{18}$$

  with $W' = R(W - c + y)$
- State variables:
  1. Size of the cake brought into the current period, W.
  2. The stochastic endowment of additional cake, y.
- For simplicity, endowment is iid: the shock today does not give any information on the shock tomorrow.
- In this case only $X = W + y$ matters. Why? Hence the problem can be simplified to

$$V(X) = \max_{0 \leq c \leq X} u(c) + \beta\, E_{y'}\, V(X') \tag{19}$$

  with $X' = R(X - c) + y'$.

Stochastic Cake-Eating Problem

- Start with the stochastic cake-eating problem defined by

$$V(W, y) = \max_{0 \le c \le W+y} u(c) + \beta\, E_{y'|y}\, V(W', y') \tag{18}$$

with $W' = R(W - c + y)$

- State variables:
  1. Size of the cake brought into the current period, W.
  2. The stochastic endowment of additional cake, y.
- For simplicity, endowment is iid: the shock today does not give any information on the shock tomorrow.
- In this case only $X = W + y$ matters. Why? Hence the problem can be simplified to

$$V(X) = \max_{0 \le c \le X} u(c) + \beta\, E_{y'}\, V(X') \tag{19}$$

with $X' = R(X - c) + y'$.

Stochastic Cake-Eating Problem

- Start with the stochastic cake-eating problem defined by

$$V(W, y) = \max_{0 \leq c \leq W+y} u(c) + \beta\, E_{y'|y}\, V(W', y') \tag{18}$$

with $W' = R(W - c + y)$

- State variables:
  1. Size of the cake brought into the current period, W.
  2. The stochastic endowment of additional cake, y.
- For simplicity, endowment is iid: the shock today does not give any information on the shock tomorrow.
- In this case only $X = W + y$ matters. Why? Hence the problem can be simplified to

$$V(X) = \max_{0 \leq c \leq X} u(c) + \beta\, E_{y'}\, V(X') \tag{19}$$

with $X' = R(X - c) + y'$.

Stochastic Cake-Eating Problem

- Start with the stochastic cake-eating problem defined by

$$V(W, y) = \max_{0 \le c \le W+y} u(c) + \beta\, E_{y'|y}\, V(W', y') \tag{18}$$

with $W' = R(W - c + y)$

- State variables:
  1. Size of the cake brought into the current period, W.
  2. The stochastic endowment of additional cake, y.
- For simplicity, endowment is iid: the shock today does not give any information on the shock tomorrow.
- In this case only $X = W + y$ matters. Why? Hence the problem can be simplified to

$$V(X) = \max_{0 \le c \le X} u(c) + \beta\, E_{y'}\, V(X') \tag{19}$$

with $X' = R(X - c) + y'$.

Value Function Iterations

In order to program value function iteration, there are several important steps:

1. Choosing a functional form for the utility function.
2. Discretizing the state and control variable.
3. Building a computer code to perform value function iteration
4. Evaluating the value and the policy function.

Value Function Iterations

In order to program value function iteration, there are several important steps:

1. Choosing a functional form for the utility function.
2. Discretizing the state and control variable.
3. Building a computer code to perform value function iteration
4. Evaluating the value and the policy function.

Value Function Iterations

In order to program value function iteration, there are several important steps:

1. Choosing a functional form for the utility function.
2. Discretizing the state and control variable.
3. Building a computer code to perform value function iteration
4. Evaluating the value and the policy function.

Value Function Iterations

In order to program value function iteration, there are several important steps:

1. Choosing a functional form for the utility function.
2. Discretizing the state and control variable.
3. Building a computer code to perform value function iteration
4. Evaluating the value and the policy function.

Functional Form and Parameterization

- We need to specify the primitive funciton, here utility.
- Constant relative risk aversion (CRRA) function

$$u(c) = \frac{c^{1-\gamma}}{1-\gamma} \tag{20}$$

- The vector θ will represent the parameters. For the cake-eating problem (γ, β) are both included in θ.
- To solve for the value function, we need to assign particular values to these parameters as well as the exogenous return R.
- For now we assume that $\beta R = 1$ so that the growth in the cake is exactly offset by the consumers discounting of the future.

Functional Form and Parameterization

- We need to specify the primitive funciton, here utility.
- Constant relative risk aversion (CRRA) function

$$u(c) = \frac{c^{1-\gamma}}{1-\gamma} \tag{20}$$

- The vector θ will represent the parameters. For the cake-eating problem (γ, β) are both included in θ.
- To solve for the value function, we need to assign particular values to these parameters as well as the exogenous return R.
- For now we assume that $\beta R = 1$ so that the growth in the cake is exactly offset by the consumers discounting of the future.

Functional Form and Parameterization

- We need to specify the primitive funciton, here utility.
- Constant relative risk aversion (CRRA) function

$$u(c) = \frac{c^{1-\gamma}}{1-\gamma} \tag{20}$$

- The vector θ will represent the parameters. For the cake-eating problem (γ, β) are both included in θ.
- To solve for the value function, we need to assign particular values to these parameters as well as the exogenous return R.
- For now we assume that $\beta R = 1$ so that the growth in the cake is exactly offset by the consumers discounting of the future.

Functional Form and Parameterization

- We need to specify the primitive funciton, here utility.
- Constant relative risk aversion (CRRA) function

$$u(c) = \frac{c^{1-\gamma}}{1-\gamma} \tag{20}$$

- The vector θ will represent the parameters. For the cake-eating problem (γ, β) are both included in θ.
- To solve for the value function, we need to assign particular values to these parameters as well as the exogenous return R.
- For now we assume that $\beta R = 1$ so that the growth in the cake is exactly offset by the consumers discounting of the future.

Functional Form and Parameterization

- We need to specify the primitive funciton, here utility.
- Constant relative risk aversion (CRRA) function

$$u(c) = \frac{c^{1-\gamma}}{1-\gamma} \tag{20}$$

- The vector θ will represent the parameters. For the cake-eating problem (γ, β) are both included in θ.
- To solve for the value function, we need to assign particular values to these parameters as well as the exogenous return R.
- For now we assume that $\beta R = 1$ so that the growth in the cake is exactly offset by the consumers discounting of the future.

State and Control Space

- We have to define the space spanned by the state and the control variables as well as the space for the endowment shocks.
- Trade off
- Two iid income shocks: y_l, y_h with transition matrix

$$\pi = \begin{pmatrix} \pi_L & \pi_H \\ \pi_L & \pi_H \end{pmatrix} \tag{21}$$

- Then the Bellman equation can be simply rewritten as

$$V(X) = \max_{0 \leq c \leq X} u(c) + \beta \sum_{i=L,H} \pi V(R(X-c) + y_i) \tag{22}$$

- For this problem it turns out that the natural state space is given by $[X_l, X_H]$.

State and Control Space

- We have to define the space spanned by the state and the control variables as well as the space for the endowment shocks.
- Trade off
- Two iid income shocks: y_l, y_h with transition matrix

$$\pi = \begin{pmatrix} \pi_L & \pi_H \\ \pi_L & \pi_H \end{pmatrix} \tag{21}$$

- Then the Bellman equation can be simply rewritten as

$$V(X) = \max_{0 \le c \le X} u(c) + \beta \sum_{i=L,H} \pi V(R(X-c) + y_i) \tag{22}$$

- For this problem it turns out that the natural state space is given by $[X_l, X_H]$.

State and Control Space

- We have to define the space spanned by the state and the control variables as well as the space for the endowment shocks.
- Trade off
- Two iid income shocks: y_l, y_h with transition matrix

$$\pi = \begin{pmatrix} \pi_L & \pi_H \\ \pi_L & \pi_H \end{pmatrix} \tag{21}$$

- Then the Bellman equation can be simply rewritten as

$$V(X) = \max_{0 \leq c \leq X} u(c) + \beta \sum_{i=L,H} \pi V(R(X-c) + y_i) \tag{22}$$

- For this problem it turns out that the natural state space is given by $[X_l, X_H]$.

State and Control Space

- We have to define the space spanned by the state and the control variables as well as the space for the endowment shocks.
- Trade off
- Two iid income shocks: y_l, y_h with transition matrix

$$\pi = \begin{pmatrix} \pi_L & \pi_H \\ \pi_L & \pi_H \end{pmatrix} \tag{21}$$

- Then the Bellman equation can be simply rewritten as

$$V(X) = \max_{0 \le c \le X} u(c) + \beta \sum_{i=L,H} \pi V(R(X-c) + y_i) \tag{22}$$

- For this problem it turns out that the natural state space is given by $[X_l, X_H]$.

State and Control Space

- We have to define the space spanned by the state and the control variables as well as the space for the endowment shocks.
- Trade off
- Two iid income shocks: y_l, y_h with transition matrix

$$\pi = \begin{pmatrix} \pi_L & \pi_H \\ \pi_L & \pi_H \end{pmatrix} \tag{21}$$

- Then the Bellman equation can be simply rewritten as

$$V(X) = \max_{0 \leq c \leq X} u(c) + \beta \sum_{i=L,H} \pi V(R(X-c) + y_i) \tag{22}$$

- For this problem it turns out that the natural state space is given by $[X_l, X_H]$.

State and Control Space

- Let n_s be the number of elements in the state space.
- Call the state space

$$\Psi_s = \{X^{i_s}\}_{i_s=1}^{n_s}$$

- The control variable c takes values in $[X_L, X_H]$. These are the extreme levels of consumption given the state space for X.
- We discretize this space into a grid of size n_c, and call the control space

$$\Psi_c = \{c^{i_c}\}_{i_c=1}^{n_c}$$

State and Control Space

- Let n_s be the number of elements in the state space.
- Call the state space
$$\Psi_s = \{X^{i_s}\}_{i_s=1}^{n_s}$$
- The control variable c takes values in $[X_L, X_H]$. These are the extreme levels of consumption given the state space for X.
- We discretize this space into a grid of size n_c, and call the control space
$$\Psi_c = \{c^{i_c}\}_{i_c=1}^{n_c}$$

State and Control Space

- Let n_s be the number of elements in the state space.
- Call the state space

$$\Psi_s = \{X^{i_s}\}_{i_s=1}^{n_s}$$

- The control variable c takes values in $[X_L, X_H]$. These are the extreme levels of consumption given the state space for X.
- We discretize this space into a grid of size n_c, and call the control space

$$\Psi_c = \{c^{i_c}\}_{i_c=1}^{n_c}$$

State and Control Space

- Let n_s be the number of elements in the state space.
- Call the state space

$$\Psi_s = \{X^{i_s}\}_{i_s=1}^{n_s}$$

- The control variable c takes values in $[X_L, X_H]$. These are the extreme levels of consumption given the state space for X.
- We discretize this space into a grid of size n_c, and call the control space

$$\Psi_c = \{c^{i_c}\}_{i_c=1}^{n_c}$$

Value Function Iteration and Policy Function

- Define the mapping $T(v(X))$, defined as

$$T(v(X)) = \max_c u(c) + \beta \sum_{i=L,H} \pi v_j(R(X-c)+y_i) \tag{23}$$

- In this expression $v(X)$ represents a candidate value function that is a proposed solution to (19).
- If $T(v(X)) = v(X)$, then indeed $v(X)$ is the unique solution to (19).
- Thus the solution to the dynamic programming problem is reduced to finding a fixed point of the mapping $T(v(X))$.

Value Function Iteration and Policy Function

- Define the mapping $T(v(X))$, defined as

$$T(v(X)) = \max_{c} u(c) + \beta \sum_{i=L,H} \pi v_j(R(X - c) + y_i) \tag{23}$$

- In this expression $v(X)$ represents a candidate value function that is a proposed solution to (19).
- If $T(v(X)) = v(X)$, then indeed $v(X)$ is the unique solution to (19).
- Thus the solution to the dynamic programming problem is reduced to finding a fixed point of the mapping $T(v(X))$.

Value Function Iteration and Policy Function

- Define the mapping $T(v(X))$, defined as

$$T(v(X)) = \max_{c} u(c) + \beta \sum_{i=L,H} \pi v_j(R(X-c) + y_i) \tag{23}$$

- In this expression $v(X)$ represents a candidate value function that is a proposed solution to (19).
- If $T(v(X)) = v(X)$, then indeed $v(X)$ is the unique solution to (19).
- Thus the solution to the dynamic programming problem is reduced to finding a fixed point of the mapping $T(v(X))$.

Value Function Iteration and Policy Function

- Define the mapping $T(v(X))$, defined as

$$T(v(X)) = \max_c u(c) + \beta \sum_{i=L,H} \pi v_j(R(X-c) + y_i) \tag{23}$$

- In this expression $v(X)$ represents a candidate value function that is a proposed solution to (19).
- If $T(v(X)) = v(X)$, then indeed $v(X)$ is the unique solution to (19).
- Thus the solution to the dynamic programming problem is reduced to finding a fixed point of the mapping $T(v(X))$.

Value Function Iteration

1. Starting with an initial guess $v_0(X)$ and compute a sequence of value functions $v_j(X)$:

$$v_{j+1}(X) = T(v_j(X)) = \max_c u(c) + \beta \sum_{i=L,H} \pi v_j(R(X-c)+y_i) \tag{24}$$

2. The iterations are stopped when $|v_{j+1}(X) - v_{j+1}(X)| < \epsilon$ for all is, where ϵ is a small number.

Value Function Iteration

1. Starting with an initial guess $v_0(X)$ and compute a sequence of value functions $v_j(X)$:

$$v_{j+1}(X) = T(v_j(X)) = \max_c u(c) + \beta \sum_{i=L,H} \pi v_j(R(X-c) + y_i) \tag{24}$$

2. The iterations are stopped when $|v_{j+1}(X) - v_{j+1}(X)| < \epsilon$ for all is, where ϵ is a small number.

Value Function Iteration in Practice

- At each iteration, the values $v_j(X)$ are stored in a ns $n_s \times 1$ matrix:

$$\mathbb{V}_j = \begin{bmatrix} v_j(X^1) & \dots & v_j(X^{i_s}) \dots & v_j(X^{n_s}) \end{bmatrix}'$$

1. Start by choosing a particular size for the cake at the start of the period, X^{i_s}.
2. Search among all the points in the control space Ψ_c for the point where $u + \beta E\, v_j(X')$ is maximized. Denote this point $c^{i_c^*}$ /
3. Once we have calculated the new value for $v_{j+1}(X^{i_s})$, we can proceed to compute similarly the value $v_{j+1}(.)$ for other sizes of the cake.
4. These new values are stacked in $\mathbb{V}_j$

Value Function Iteration in Practice

- At each iteration, the values $v_j(X)$ are stored in a ns $n_s \times 1$ matrix:

$$\mathbb{V}_j = \begin{bmatrix} v_j(X^1) & \dots & v_j(X^{i_s}) \dots & v_j(X^{n_s}) \end{bmatrix}'$$

1. Start by choosing a particular size for the cake at the start of the period, X^{i_s}.
2. Search among all the points in the control space Ψ_c for the point where $u + \beta\, E\, v_j(X')$ is maximized. Denote this point $c^{i_c^*}$ /
3. Once we have calculated the new value for $v_{j+1}(X^{i_s})$, we can proceed to compute similarly the value $v_{j+1}(.)$ for other sizes of the cake.
4. These new values are stacked in $\mathbb{V}_j$

Value Function Iteration in Practice

- At each iteration, the values $v_j(X)$ are stored in a ns $n_s \times 1$ matrix:

$$\mathbb{V}_j = \begin{bmatrix} v_j(X^1) & \ldots & v_j(X^{i_s}) \ldots & v_j(X^{n_s}) \end{bmatrix}'$$

1. Start by choosing a particular size for the cake at the start of the period, X^{i_s}.
2. Search among all the points in the control space Ψ_c for the point where $u + \beta\, E\, v_j(X')$ is maximized. Denote this point $c^{i_c^*}$ /
3. Once we have calculated the new value for $v_{j+1}(X^{i_s})$, we can proceed to compute similarly the value $v_{j+1}(.)$ for other sizes of the cake.
4. These new values are stacked in $\mathbb{V}_j$

Value Function Iteration in Practice

- At each iteration, the values $v_j(X)$ are stored in a ns $n_s \times 1$ matrix:

$$\mathbb{V}_j = \begin{bmatrix} v_j(X^1) & \dots & v_j(X^{i_s}) \dots & v_j(X^{n_s}) \end{bmatrix}'$$

1. Start by choosing a particular size for the cake at the start of the period, X^{i_s}.
2. Search among all the points in the control space Ψ_c for the point where $u + \beta\, E\, v_j(X')$ is maximized. Denote this point $c^{i_c^*}$ /
3. Once we have calculated the new value for $v_{j+1}(X^{i_s})$, we can proceed to compute similarly the value $v_{j+1}(.)$ for other sizes of the cake.
4. These new values are stacked in $\mathbb{V}_j$

Value Function Iteration in Practice

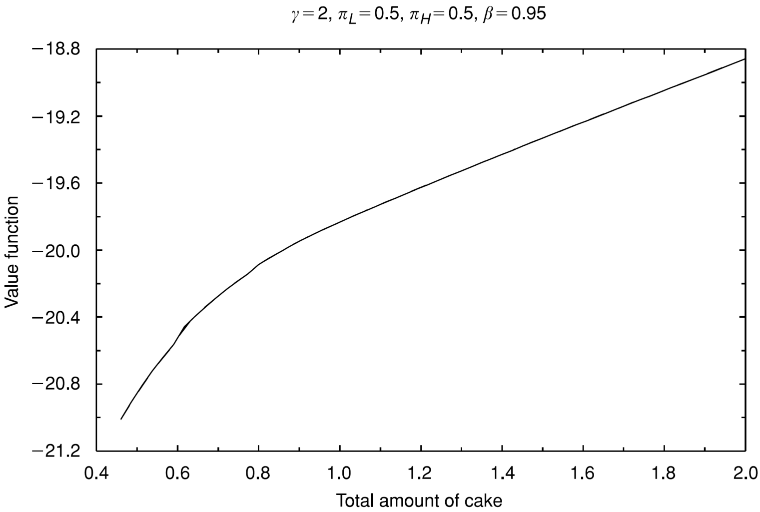
- At each iteration, the values $v_j(X)$ are stored in a ns $n_s \times 1$ matrix:

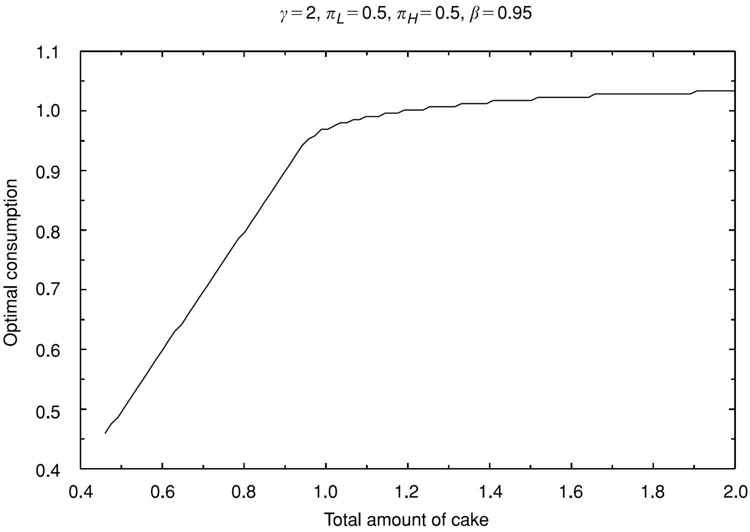
$$\mathbb{V}_j = \left[v_j(X^1) \quad \dots \quad v_j(X^{i_s}) \dots \quad v_j(X^{n_s}) \right]'$$

1. Start by choosing a particular size for the cake at the start of the period, X^{i_s}.
2. Search among all the points in the control space Ψ_c for the point where $u + \beta\, E\, v_j(X')$ is maximized. Denote this point $c^{i_c^*}$ /
3. Once we have calculated the new value for $v_{j+1}(X^{i_s})$, we can proceed to compute similarly the value $v_{j+1}(.)$ for other sizes of the cake.
4. These new values are stacked in $\mathbb{V}_j$

```
i_s=1
do until i_s>n_s                 * Loop over all sizes of the
                                   total amount of cake X *
  c_L=X_L                        * Min value for consumption *
  c_H=X[i_s]                     * Max value for consumption *
  i_c=1
    do until i_c>n_c             * Loop over all consumption
                                   levels *
    c=c_L+(c_H-c_L)/n_c*(i_c-1)
    i_y=1
    EnextV=0                     * Initialize the next value
                                   to zero *
    do until i_y>n_y             * Loop over all possible
                                   realizations of the future
                                   endowment *
    nextX=R*(X[i_s]-c)+Y[i_y]    * Next period amount of
                                   cake *
    nextV=V(nextX)               * Here we use interpolation
                                   to find the next value
                                   function *

    EnextV=EnextV+nextV*Pi[i_y]  * Store the expected future
                                   value using the transition
                                   matrix *
    i_y=i_y+1
    endo                         * End of loop over
                                   endowment *
    aux[i_c]=u(c)+beta*EnextV    * Stores the value of a given
                                   consumption level *
    i_c=i_c+1
    endo                         * End of loop over
                                   consumption *
  newV[i_s,i_y]=max(aux)         * Take the max over all
                                   consumption levels *
  i_s=i_s+1
  endo                           * End of loop over size of
                                   cake *
V=newV                           * Update the new value
                                   function *
```


$\gamma = 2$, $\pi_L = 0.5$, $\pi_H = 0.5$, $\beta = 0.95$
−18.8
−19.2
−19.6
−20.0
−20.4
−20.8
−21.2
Value function
0.4
0.6
0.8
1.0
1.2
1.4
1.6
1.8
2.0
Total amount of cake



$\gamma = 2$, $\pi_L = 0.5$, $\pi_H = 0.5$, $\beta = 0.95$
Optimal consumption
Total amount of cake
1.1
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.4
0.6
0.8
1.0
1.2
1.4
1.6
1.8
2.0