Introduction to Dynamic Programming

ScPo Graduate Labor

February 21, 2018

Table of contents

Dynamic Programming
- Dynamic Programming: Piece of Cake
- Dynamic Programming Theory
- Stochastic Dynamic Programming

Introduction

- This lecture will introduce you to a powerful technique called *dynamic programming (DP)*.
- This set of slides is very similar to the one from your grad macro course. (same teacher.)
- We will repeat much of that material today.
- Next time will talk about a paper that uses DP to solve a dynamic lifecycle model.

References

- Before we start, some useful references on DP:
  1. Adda and Cooper (2003): Dynamic Economics.
  2. Ljungqvist and Sargent (2012) (LS): Recursive Macroeconomic Theory.
  3. Lucas and Stokey (1989): Recursive Methods in Economics Dynamics.
- They are ordered in increasing level of mathematical rigor. Adda and Cooper is a good overview, LS is rather short.

Cake Eating

- You are given a cake of size W_1 and need to decide how much of it to consume in each period $t = 1, 2, 3, \ldots$
- Cake consumption valued as $u(c)$, u is concave, increasing, differentiable and $\lim_{c \to 0} u'(c) = \infty$.
- Lifetime utility is

$$U = \sum_{t=1}^{T} \beta^{t-1} u(c_t), \beta \in [0, 1] \tag{1}$$

- Let's assume the cake does not depreciate/perish, s.t. the *law of motion* of cake is

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

  i.e. cake in $t + 1$ is this cake in t minus whatever you have of it in t.
- How to decide on the optimal consumption sequence $\{c_t\}_{t=1}^{T}$?

A sequential Problem

- This problem can be written as

$$\begin{aligned} v(W_1) &= \max_{\{W_{t+1},c_t\}_{t=1}^{T}} \sum_{t=1}^{T} \beta^{t-1} u(c_t) \\ & \quad s.t. \\ W_{t+1} &= W_t - c_t \\ c_t, W_{t+1} &\geq 0 \text{ and } W_1 \text{ given.} \end{aligned} \tag{3}$$

- Notice that the law of motion (2) implies that

$$\begin{aligned} W_1 &= W_2 + c_1 \\ &= (W_3 + c_2) + c_1 \\ &= \dots \\ &= W_{T+1} + \sum_{t=1}^{T} c_t \end{aligned} \tag{4}$$

Solving the sequential Problem

- Formulate and solve the Lagrangian for (3) with (4):

Solving the sequential Problem

- Formulate and solve the Lagrangian for (3) with (4):

$$L = \sum_{t=1}^{T} \beta^{t-1} u(c_t) + \lambda \left[W_1 - W_{T+1} - \sum_{t=1}^{T} c_t \right] + \phi \left[W_{T+1} \right]$$

- First order conditions:

$$\frac{\partial L}{\partial c_t} = 0 \quad \Longrightarrow \quad \beta^{t-1} u'(c_t) = \lambda \; \forall t \tag{5}$$

$$\frac{\partial L}{\partial W_{T+1}} = 0 \quad \Longrightarrow \quad \lambda = \phi \tag{6}$$

  - ϕ is lagrange multiplier on non-negativity constraint for W_{T+1}.
  - we ignore the constraint $c_t \geq 0$ by the Inada assumption.

Interpreting the sequential solution

- From (5) we know that $\beta^{t-1}u'(c_t) = \lambda$ holds in each t.
- Therefore

$$\begin{aligned} \beta^{t-1}u'(c_t) &= \lambda \\ &= \beta^{(t+1)-1}u'(c_{t+t}) \end{aligned}$$

  i.e. we get the Euler Equation

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

- along an optimal sequence $\{c_t^*\}_{t=1}^T$, each adjacent period $t, t+1$ must satisfy (7).
- If (7) holds, one cannot increase utility by moving some c_t to c_{t+1}.
- What about deviation from $\{c_t^*\}_{t=1}^T$ between t and $t+2$?

Is the Euler Equation enough?

- Is the Euler Equation sufficient for optimality?

Is the Euler Equation enough?

- Is the Euler Equation sufficient for optimality?
- No! We could satisfy (7), but have $W_T > c_T$, i.e. there is some cake left.
- What does this remind you of?
- Discuss how this relates to the value of multipliers λ, ϕ.
- Solution is given by initial condition (W_1), terminal condition $W_{T+1} = 0$ and path in EE.
- Call this solution the **value function**

$$v(W_1)$$

- $v(W_1)$ is the maximal utility flow over T periods given initial cake W_1.

Digression: Power Utility Functions

- We'll look at a specific class of U functions: **Power Utility**, or *isoelastic* utility functions.
- This class includes the **hyperbolic** or **constant** relative risk aversion functions.
- It's defined as

$$u(c) = \begin{cases} \frac{c^{1-\gamma}}{1-\gamma} & \text{if } \gamma \neq 1 \\ \ln(c) & \text{if } \gamma = 1. \end{cases}$$

- The *coefficient of relative risk aversion* is γ i.e. a **constant**.
- Your risk aversion does not depend on your level of wealth.

Code for CRRA utility function

```
# this is julia
function u(c,gamma)
    if gamma==1
        return log(c)
    else
        return (1/(1-gamma)) * c^(1-gamma)
    end
end
```

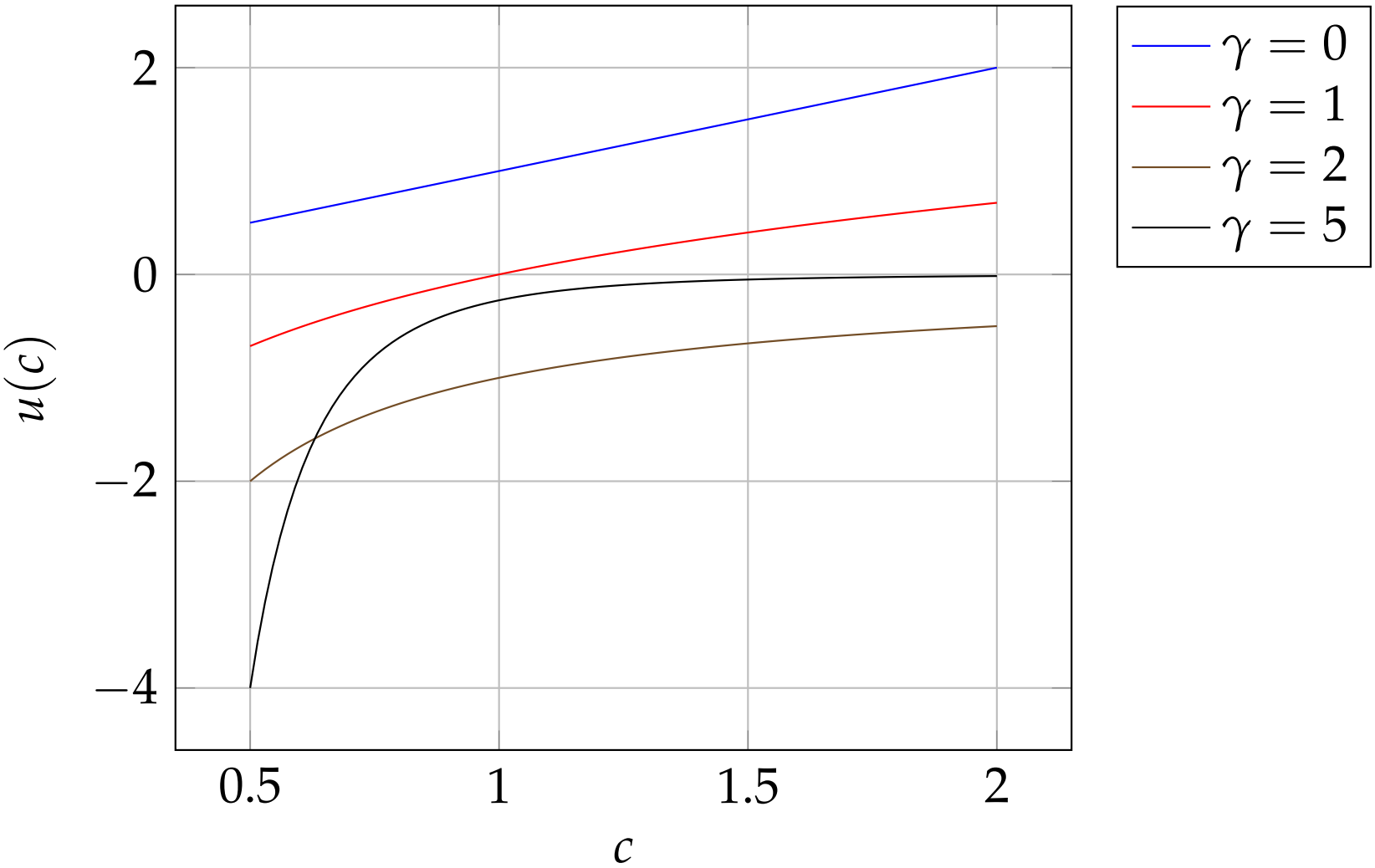
Code for plot

```
using PGFPlots
using LaTeXStrings

p=Axis([
Plots.Linear(x->u(x,0),(0.5,2),legendentry=L"$\gamma=0$"),
Plots.Linear(x->u(x,1),(0.5,2),legendentry=L"$\gamma=1$"),
Plots.Linear(x->u(x,2),(0.5,2),legendentry=L"$\gamma=2$"),
Plots.Linear(x->u(x,5),(0.5,2),legendentry=L"$\gamma=5$")
],xlabel=L"$c$",ylabel=L"$u(c)$",style="grid=both")
p.legendStyle = "{at={(1.05,1.0)},anchor=north west}"
save("images/dp/CRRA.tex",p,include_preamble=false)

# then, next slide just has \input{images/dp/CRRA}
```

CRRA functions



CRRA utility Properties

The next 5 slides were contributed by [click!]Cormac O'Dea @ Yale

- We had:

$$u(c) = \begin{cases} \frac{c^{1-\gamma}}{1-\gamma} & \text{if } \gamma \neq 1 \\ \ln(c) & \text{if } \gamma = 1. \end{cases}$$

  γ^{-1} is the elasticity of intertemporal substitution (IES)

- IES is defined as the *percent change in consumption growth per percent increase in the net interest rate*.

CRRA utility Properties

The next 5 slides were contributed by [click!]Cormac O'Dea @ Yale

- We had:

$$u(c) = \begin{cases} \frac{c^{1-\gamma}}{1-\gamma} & \text{if } \gamma \neq 1 \\ \ln(c) & \text{if } \gamma = 1. \end{cases}$$

  γ^{-1} is the elasticity of intertemporal substitution (IES)

- IES is defined as the *percent change in consumption growth per percent increase in the net interest rate*.

- It is generally accepted that $\gamma \geq 1$, in which case, for $c \in \mathbb{R}^+$

$$u(c) < 0, \quad \lim_{c\to 0} u(c) = -\infty, \quad \lim_{c\to+\infty} u(c) = 0$$
$$u'(c) > 0, \quad \lim_{c\to 0} u'(c) = +\infty, \quad \lim_{c\to+\infty} u'(c) = 0$$

CRRA utility: solution I

- Let's modify our cake eating problem.
- $W_t \Rightarrow a_t$, and we introduce gross interest $R = 1 + r$. (for non-growing cake just take $r = 0$).

$$\max_{(c_1,\ldots,c_T)\in(\mathbb{R}^+)^T} \sum_{t=1}^{T} \beta^{t-1} \frac{c_t^{1-\gamma}}{1-\gamma} \qquad \text{s.t} \quad \sum_{t=1}^{T} R^{1-t} c_t \leq a_1$$

CRRA utility: solution I

- Let's modify our cake eating problem.
- $W_t \Rightarrow a_t$, and we introduce gross interest $R = 1 + r$. (for non-growing cake just take $r = 0$).

$$\max_{(c_1,\ldots,c_T)\in(\mathbb{R}^+)^T} \sum_{t=1}^{T} \beta^{t-1} \frac{c_t^{1-\gamma}}{1-\gamma} \qquad \text{s.t} \quad \sum_{t=1}^{T} R^{1-t} c_t \leq a_1$$

- Euler equations are necessary for interior solutions:

$$c_t^{-\gamma} = \beta R c_{t+1}^{-\gamma} \quad \Rightarrow \quad c_t = (\beta R)^{-\frac{1}{\gamma}} c_{t+1} \quad \text{for } t = 1, \ldots, T-1$$

CRRA utility: solution I

- Let's modify our cake eating problem.
- $W_t \Rightarrow a_t$, and we introduce gross interest $R = 1 + r$. (for non-growing cake just take $r = 0$).

$$\max_{(c_1,\ldots,c_T)\in(\mathbb{R}^+)^T} \sum_{t=1}^{T} \beta^{t-1} \frac{c_t^{1-\gamma}}{1-\gamma} \qquad \text{s.t} \quad \sum_{t=1}^{T} R^{1-t} c_t \leq a_1$$

- Euler equations are necessary for interior solutions:

$$c_t^{-\gamma} = \beta R c_{t+1}^{-\gamma} \quad \Rightarrow \quad c_t = (\beta R)^{-\frac{1}{\gamma}} c_{t+1} \quad \text{for } t = 1, \ldots, T-1$$

- By successive substitution:

$$c_t = (\beta R)^{\frac{t-1}{\gamma}} c_1$$

CRRA utility: solution II

- The budget constraint and optimality condition imply

$$\begin{aligned} a_1 &= \sum_{t=1,\ldots,T} R^{1-t} c_t \\ &= c_1 \sum_{t=1,\ldots,T} \left(\beta^{\frac{1}{\gamma}} R^{\frac{1-\gamma}{\gamma}} \right)^{t-1} \\ &= c_1 \sum_{t=1,\ldots,T} \alpha^{t-1} \end{aligned}$$

CRRA utility: solution II

- The budget constraint and optimality condition imply

$$\begin{aligned} a_1 &= \sum_{t=1,\ldots,T} R^{1-t} c_t \\ &= c_1 \sum_{t=1,\ldots,T} \left(\beta^{\frac{1}{\gamma}} R^{\frac{1-\gamma}{\gamma}}\right)^{t-1} \\ &= c_1 \sum_{t=1,\ldots,T} \alpha^{t-1} \end{aligned}$$

- The solution for $t = 1, \ldots, T$:

$$c_1 = \frac{1-\alpha}{1-\alpha^T} a_1 \quad \text{and} \quad c_t = \frac{1-\alpha}{1-\alpha^T} (\beta R)^{\frac{t-1}{\gamma}} a_1$$

CRRA utility: solution III

In general, if the optimisation problem starts at time t as follows

$$\max_{(c_t,\ldots,c_T)\in(\mathbb{R}^+)^{T-t+1}} \sum_{\tau=t}^{T} \beta^{\tau-t} \frac{c_\tau^{1-\gamma}}{1-\gamma} \qquad \text{s.t} \quad \sum_{\tau=t}^{T} R^{\tau-t} c_\tau \leq a_\tau$$

the solution for c_t is

$$c_t = \frac{1-\alpha}{1-\alpha^{T-t+1}} a_t$$

This is the *consumption function*, a linear function of assets if utility is CRRA

CRRA consumption: $c_t = \frac{1-\alpha}{1-\alpha^T}(\beta R)^{\frac{t-1}{\gamma}} a_1$

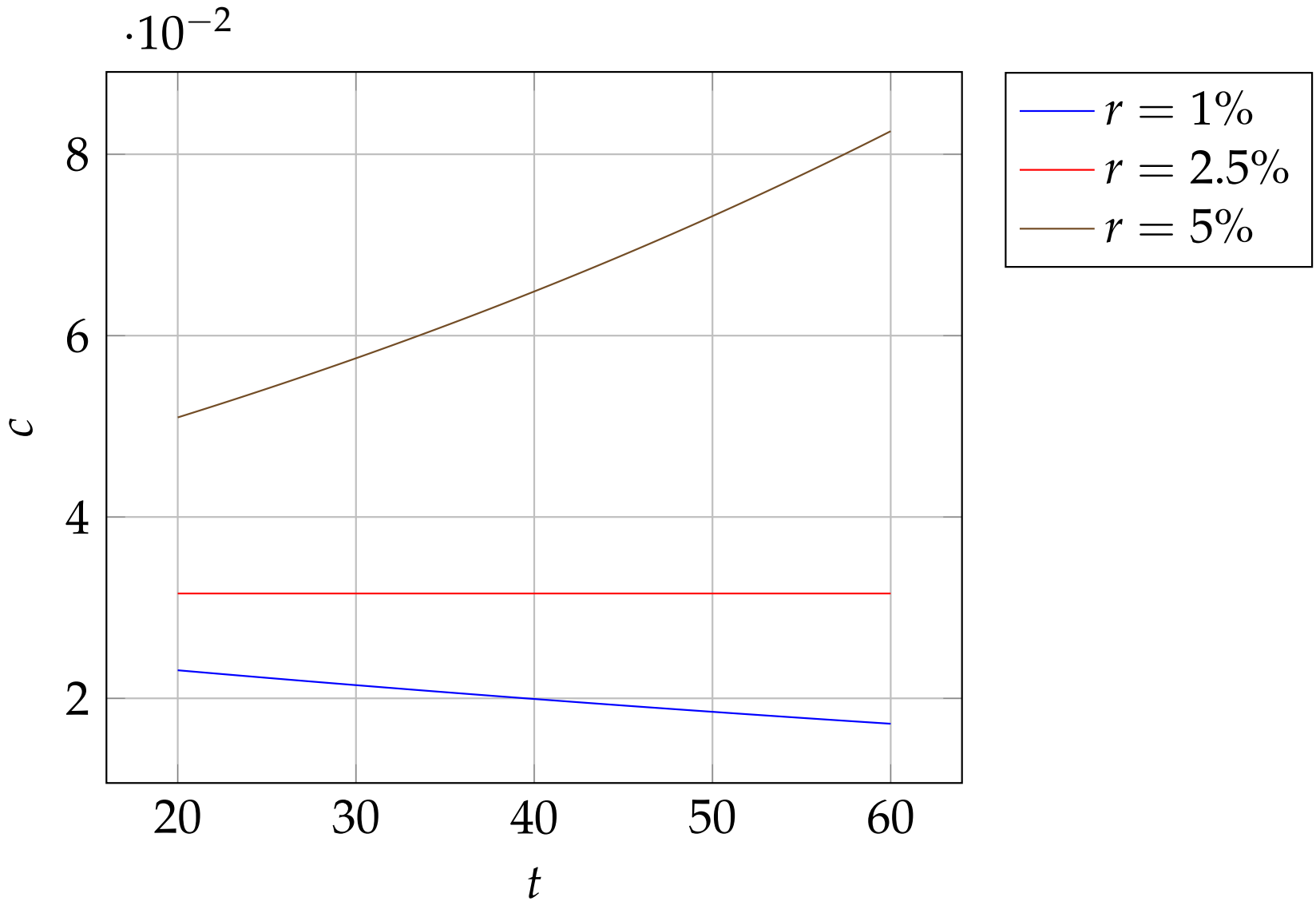


Figure: βR determines the profile of the solution.
$\beta = 1.025^{-1}, \gamma = 2, a_1 = 20$

The Dynamic Programming approach with $T = \infty$

- Let's consider the case $T = \infty$.
- In other words

$$\max_{\{W_{t+1}, c_t\}_{t=1}^{\infty}} \sum_{t=1}^{\infty} \beta^{t-1} u(c_t) \tag{8}$$

$$s.t.$$

$$W_{t+1} = W_t - c_t \tag{9}$$

- Under some conditions, this can be written as

$$v(W_t) = \max_{c_t \in [0, W_t]} u(c_t) + \beta v(W_t - c_t) \tag{10}$$

- Some Definitions:
  - Call W the **state variable,**
  - and c the **control variable**.
  - (9) is the **law of motion** or **transition equation.**

The Dynamic Programming approach with $T = \infty$

- Note that t is irrelevant in (10). Only W matters.
- Substituting $c = W - W'$, where x' is next period's value of x

$$v(W) = \max_{W' \in [0,W]} u\left(W - W'\right) + \beta v(W') \tag{11}$$

- This is the **Bellman Equation** after Richard Bellman.
- It is a functional equation (v is on both sides!).
- Our problem has changed from finding $\{W_{t+1}, c_t\}_{t=1}^{\infty}$ to finding the function v.

This is called a fixed point problem:

Find a function v such that plugging in W on the RHS and doing the maximization, we end up with *the same* v on the LHS.

Value Function and Policy Function

- Great! We have reduced an infinite-length sequential problem to a one-dimensional maximization problem.
- But we have to find 2(!) unknown functions! Why two?
- The maximizer of the RHS of (11) is the **policy function**, $g(W) = c^*$.
- This function gives the optimal value of the control variable, given the state.
- It satisfies

$$v(W) = u(g(W)) + \beta v(g(W)) \tag{12}$$

(you can see that the $\max$ operator vanished, because $g(W)$ is the optimal choice)

- In practice, finding value and policy function is the one operation.

Using Dynamic Programming to solve the Cake problem

- Let's pretend that we knew v for now:

$$v(W) = \max_{W' \in [0,W]} u\left(W - W'\right) + \beta v(W')$$

- Assuming v is differentiable, the FOC wrt W'

$$u'(c) = \beta v'\left(W'\right) \tag{13}$$

- Taking the partial derivative w.r.t. the state W, we get the *envelope condition*

$$v'(W) = u'(c) \tag{14}$$

- This needs to hold in each period. Therefore

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

Using Dynamic Programming to solve the Cake problem

- Combining (13) with (15)

$$\begin{aligned} u'(c) &\overset{(13)}{=} \beta v'\left(W'\right) \\ &\overset{(15)}{=} \beta u'(c') \end{aligned}$$

  we obtain the usual euler equation.
- Any solution $v$will satisfy this necessary condition, as in the sequential case.

Using Dynamic Programming to solve the Cake problem

- Combining (13) with (15)

$$u'(c) \overset{(13)}{=} \beta v'\left(W'\right) \overset{(15)}{=} \beta u'(c')$$

  we obtain the usual euler equation.
- Any solution $v$will satisfy this necessary condition, as in the sequential case.

- So far, so good. But we still don't know v!

Finding v

- Finding the Bellman equation v and associated policy function g is not easy.
- In general, it is impossible to find an analytic expression, i.e. to do it by hand.
- Most of times you will use a computer to solve for it.
- **preview:** The rationale for why we can find it has to do with the fixed point nature of the problem. We will see that under some conditions we can **always** find that fixed point.
- We will look at a particular example now, that we can solve by hand.

Finding v: an example with closed form solution

- Let's assume that $u(c) = \ln c$ in (11).
- Also, let's conjecture that the value function has the form

$$v(W) = A + B \ln W \tag{16}$$

- We have to find A, B such that (16) satisfies (11).
- Plug into (11):

$$A + B \ln W = \max_{W'} \ln \left(W - W'\right) + \beta \left(A + B \ln W'\right) \tag{17}$$

Finding v: an example with closed form solution

- Let's assume that $u(c) = \ln c$ in (11).
- Also, let's conjecture that the value function has the form

$$v(W) = A + B \ln W \tag{16}$$

- We have to find A, B such that (16) satisfies (11).
- Plug into (11):

$$A + B \ln W = \max_{W'} \ln \left(W - W'\right) + \beta \left(A + B \ln W'\right) \tag{17}$$

- FOC wrt W':

$$\begin{aligned} \frac{1}{W - W'} &= \frac{\beta B}{W'} \\ W' &= \beta B (W - W') \\ W' &= \frac{\beta B}{1 + \beta B} W \\ &\equiv g(W) \end{aligned}$$

Finding v: an example with closed form solution

- Let's use this policy function in (17):

$$\begin{aligned} v(W) &= \ln\left(W - g(W)\right) + \beta\left(A + B\ln g(W)\right) \\ &= \ln\frac{W}{1+\beta B} + \beta\left(A + B\ln\left[\frac{\beta B}{1+\beta B}W\right]\right) \end{aligned}$$

- Now we collect all terms $\ln W$ on the RHS, and put all else into the constant A:

$$\begin{aligned} v(W) &= A + \ln W + \beta B \ln W \\ &= A + (1+\beta B)\ln W \end{aligned}$$

- We conjectured that $v(W) = A + B\ln W$. Hence

$$\begin{aligned} B &= (1+\beta B) \\ B &= \frac{1}{1-\beta} \end{aligned}$$

- Policy function: $g(W) = \beta W$

The Guess-and-Verify method

- Note that we guessed a functional form for v.
- And then we verified that it consitutues a solution to the functional equation.
- This method (guess and verify) would in principle always work, but it's not very practical.

Solving the Cake problem with $T < \infty$

- When time is finite, solving this DP is fairly simple.
- If we know the value in the final period, we can simply go backwards in time.
- In period T there is no point setting $W' > 0$. Therefore

$$v_T(W) = u(W) \tag{18}$$

- Notice that we index the value function with time in this case:
  - it's not the same to have W in period 1 as it is to have W in period T. Right?
- But if we know v_T for all values of W, we can construct v_{T-1}!

Backward Induction and the Cake Problem

- We know that

$$\begin{aligned} v_{T-1}(W_{T-1}) &= \max_{W_T \in [0, W_{T-1}]} u(W_{T-1} - W_T) + \beta v_T(W_T) \\ &= \max_{W_T \in [0, W_{T-1}]} u(W_{T-1} - W_T) + \beta u(W_T) \\ &= \max_{W_T \in [0, W_{T-1}]} \ln(W_{T-1} - W_T) + \beta \ln W_T \end{aligned}$$

- FOC wrt W_T:

$$\begin{aligned} \frac{1}{W_{T-1} - W_T} &= \frac{\beta}{W_T} \\ W_T &= \frac{\beta}{1+\beta} W_{T-1} \end{aligned}$$

- Thus the value function in $T-1$ is

$$v_{T-1}(W_{T-1}) = \ln\left(\frac{W_{T-1}}{\beta}\right) + \beta \ln\left(\frac{\beta}{1+\beta} W_{T-1}\right)$$

Backward Induction and the Cake Problem

- Correspondingly, in $T-2$:

$$\begin{aligned} v_{T-2}\left(W_{T-2}\right) &= \max_{W_{T-1}\in[0,W_{T-2}]} u\left(W_{T-2}-W_{T-1}\right)+\beta v_{T-1}\left(W_{T-1}\right) \\ &= \max_{W_{T-1}\in[0,W_{T-2}]} u\left(W_{T-2}-W_{T-1}\right) \\ &+ \beta\left[\ln\left(\frac{W_{T-1}}{\beta}\right)+\beta\ln\left(\frac{\beta}{1+\beta}W_{T-1}\right)\right] \end{aligned}$$

- FOC wrt W_{T-2}.
- and so on until $t=1$.
- Again, without log utility, this quickly get intractable. But your computer would proceed in the same backwards iterating fashion.
- Notice that with T finite, there is *no fixed point problem* if we do backwards induction.

Dynamic Programming Theory

- Let's go back to the infinite horizon problem.
- Let's define a general DP as follows.
- Payoffs over time are

$$U = \sum_{t=1}^{\infty} \beta^t \tilde{u}\left(s_t, c_t\right)$$

  where $\beta < 1$ is a discount factor, s_t is the state, c_t is the control.
- The state (vector) evolves as $s_{t+1} = h(s_t, c_t)$.
- All past decisions are contained in s_t.

DP Theory: more assumptions

- Let $c_t \in C(s_t), s_t \in S$ and assume $\tilde{u}$ is bounded in $(c, s) \in C \times S$.
- Stationarity: neither payoff $\tilde{u}$ nor transition h depend on time.
- Modify $\tilde{u}$ to u s.t. in terms of s' (as in cake: $c = W - W'$):

$$v(s) = \max_{s' \in \Gamma(s)} u(s, s') + \beta v(s') \tag{19}$$

- $\Gamma(s)$ is the constraint set (or feasible set) for s' when the current state is s:
  - before that was $\Gamma(W) = [0, W]$
- We will work towards one possible set of sufficient conditions for the existence to the functional equation. Please consult Stokey and Lucas for greater detail.

Proof of Existence

Theorem

Assume that $u(s, s')$ is real-valued, continuous, and bounded, that $\beta \in (0, 1)$, and that the constraint set $\Gamma(s)$ is nonempty, compact, and continuous. Then there exists a unique function $v(s)$ that solves (19).

Proof.

Stokey and Lucas (1989, theoreom 4.6). □

The Bellman Operator $T(W)$

- Define an *operator* T*on function* W *as* $T(W)$*:*

$$T(W)(s) = \max_{s' \in \Gamma(s)} u(s, s') + \beta W(s') \tag{20}$$

- The Bellman operator takes a guess of current value function W, performs the maximization, and returns the next value function.
- Any $v(s) = T(v)(s)$ is a solution to (19).
- So we need to find a fixed point of $T(W)$.
- This argument proceeds by showing that $T(W)$ is a **contraction**.
- Info: This relies on the *Banach (or contraction) mapping theorem*.
- There are two sufficiency conditions we can check: Monotonicity, and Discounting.

The Blackwell (1965) sufficiency conditions: Monotonicity

- Need to check Monotonicity and Discounting of the operator $T(W)$.
- **Monotonicity** means that

$$W(s) \geq Q(s) \quad \Longrightarrow \quad T(W)(s) \geq T(Q)(s), \forall s$$

- Let $\phi_Q(s)$ be the policy function of

$$Q(s) = \max_{s' \in \Gamma(s)} u(s, s') + \beta Q(s')$$

and assume $W(s) \geq Q(s)$. Then

$$\begin{aligned} T(W)(s) &= \max_{s' \in \Gamma(s)} u(s, s') + \beta W(s') \geq u(s, \phi_Q(s)) + \beta W(\phi_Q(s)) \\ &\geq u(s, \phi_Q(s)) + \beta Q(\phi_Q(s)) \equiv T(Q)(s) \end{aligned}$$

Show example with $W(s) = \log(s^2), Q(s) = \log(s), s > 0$

The Blackwell (1965) sufficiency conditions: Discounting

- Adding constant a to W leads $T(W)$ to increase less than a.
- In other words

$$T(W+a)(s) \leq T(W)(s) + \beta a, \beta \in [0,1)$$

- *discounting* because $\beta < 1$.
- To verify on the Bellman operator:

$$T(W+a)(s) = \max_{s' \in \Gamma(s)} u(s,s') + \beta \left[W(s') + a\right] = T(W)(s) + \beta a$$

- Intuition: the discounting property is key for a **contraction**.
- In successive iterations on $T(W)$ we add only a fraction β of W.

Contraction Mapping Theorem (CMT)

- The CMT tells us that for a function of type $T(\cdot)$
  1. There is a unique fixed point. (from previous Stokey-Lucas proof.)
  2. This fixed point can be reached by iterating on T in (20) **using an arbitrary starting point.**
- Very useful to find a solution to (19):
  1. Start with an initial guess $V_0(s)$.
  2. Apply the Bellman operator to get $V_1 = T(V_0)$
     1. if $V_1(s) = V_0(s)$ we have a solution, done.
     2. if not, continue:
  3. Apply the Bellman operator to get $V_2 = T(V_1)$
  4. etc until $T(V) = V$.
- Again: if $T(V)$ is a contraction, this **will** converge.
- This technique is called **value function iteration.**

Value Function inherits Properties of u

Theorem

*Assume $\underline{u(s, s')}$ is real-values, continuous, **concave** and bounded, $0 < \beta < 1$, that S is a convex subset of $\mathbb{R}^k$ and that the constraint set $\Gamma(s)$ is non-empty, compact-valued, convex, and continuous. Then the unique solution to (19) is **strictly concave.** Furthermore, the policy $\phi(s)$ is a continuous, single-valued function.*

Proof.

See theorem 4.8 in Stokey and Lucas (1989). □

Value Function inherits Properties of u

- proof shows that if V is concave, so is $T(V)$.
- Given $u(s, s')$ is concave, let the initial guess be

$$V_0(s) = \max_{s' \in \Gamma(s)} u(s, s')$$

  and therefore $V_0(s)$ is concave.
- Since T preserves concavity, $V_1 = T(V_0)$ is concave etc.

VFI Example: Growth Model

$$V(k) = \max_{0<k'<f(k)} \ln(f(k) - k') + \beta V(k') \tag{21}$$

$$f(k) = k^{\alpha} \tag{22}$$

$$k_0 \text{ given} \tag{23}$$

VFI Example: Growth Model

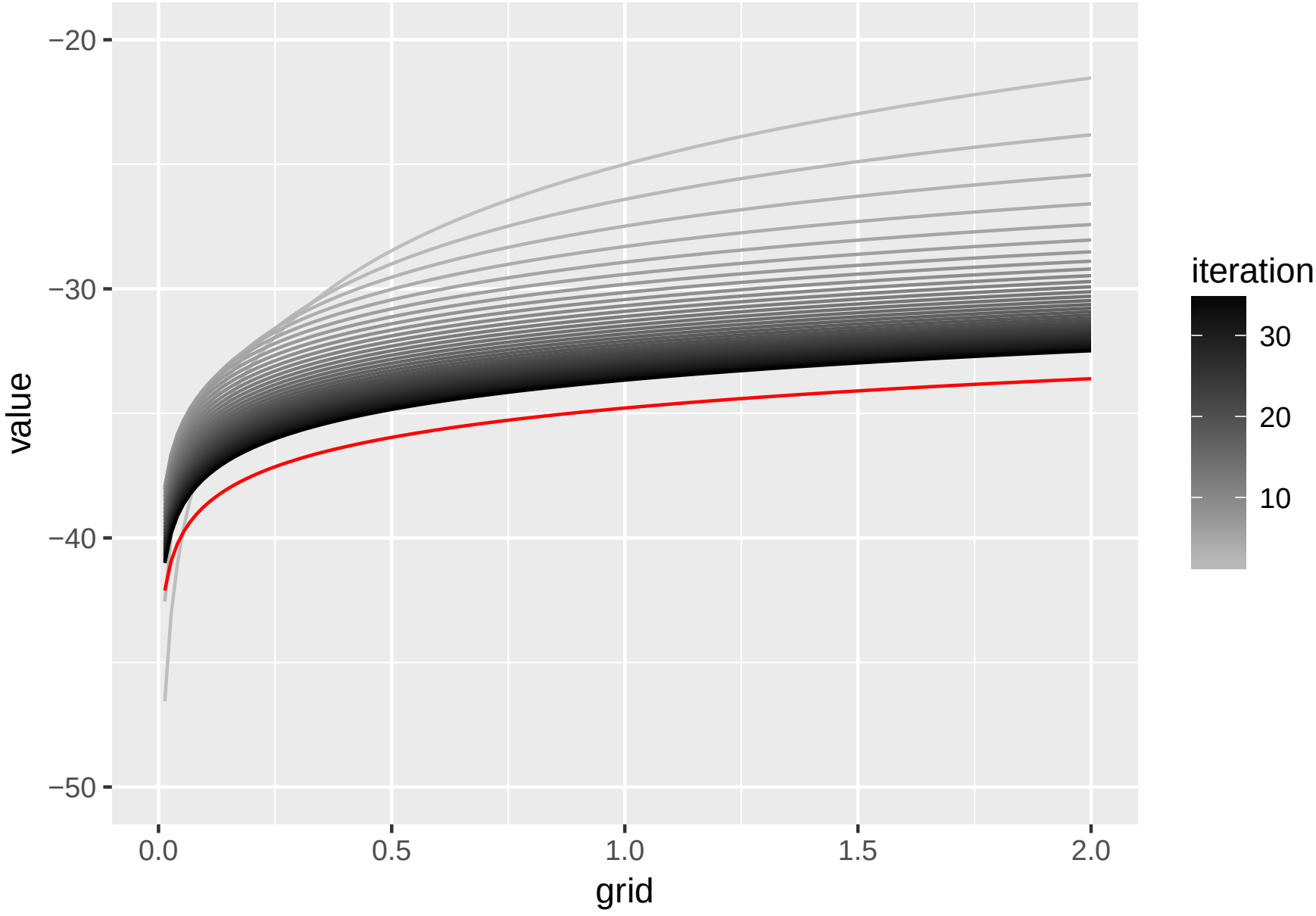
R Code

```
# parameters
alpha      = 0.65
beta       = 0.95
grid_max   = 2  # upper bound of capital grid
n          = 150
kgrid      = seq(from=1e-6,to=grid_max,len=n)  # equispaced
f <- function(x,alpha){x^alpha}  # defines the production

# value function iteration (VFI)
VFI <- function (grid,V0,maxIter){
    w = matrix(0,length(grid),maxIter)
    w[,1] = V0 # initial condition
    for (i in 2:maxIter){
        w[ ,i] = bellman_operator(grid, w[ ,i-1])
    }
    return(w)
}
```

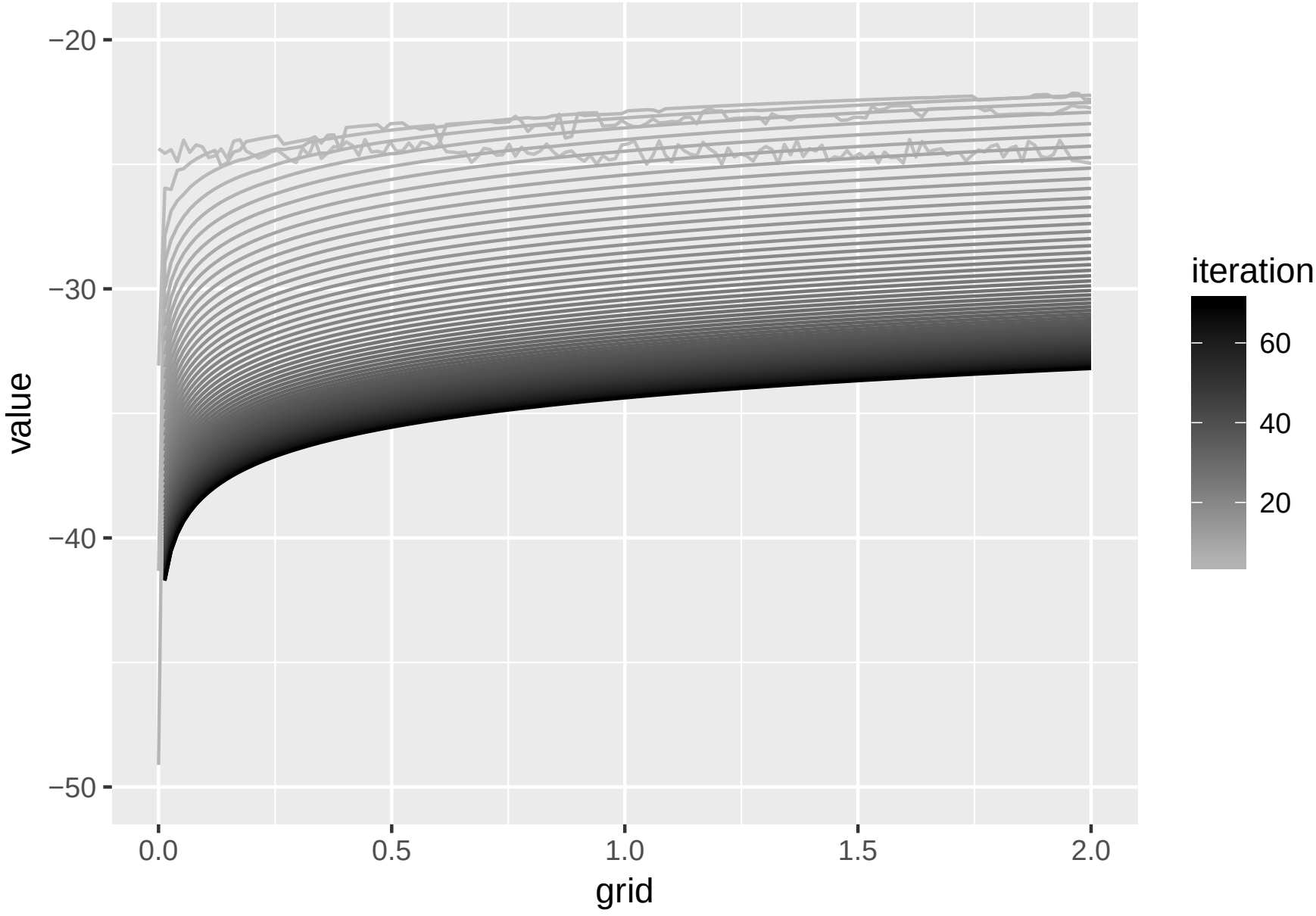
VFI Example

Starting from a $\log(k)$ scaled initial value



VFI Example

Starting from a random initial value



Stochastic Dynamic Programming

- There are several ways to include uncertainty into this framework - here is one:
- Let's assume the existence of a variable ϵ, representing a *shock*.
- Assumptions:
  1. ϵ_t affects the agent's payoff in period t.
  2. ϵ_t is exogenous: the agent cannot influence it.
  3. ϵ_t depends only on ϵ_{t-1} (and not on ϵ_{t-2}. although we could add ϵ_{t-1} as a state variable!)
  4. The distribution of $\epsilon' | \epsilon$ is time-invariant.
- Defined in this way, we call ϵ a *first-order Markov process*.

The Markov Property

Definition

A stochastic process $\{x_t\}$ is said to have the *Markov property* if for all $k \geq 1$ and all t,

$$\Pr\left(x_{t+1} | x_t, x_{t-1}, \ldots, x_{t-k}\right) = \Pr\left(x_{t+1} | x_t\right).$$

We assume that $\{\epsilon_t\}$ has this property, and characterize it by a *Markov Chain*.

Markov Chains

Definition

A time-invariant n**-State Markov Chain** consists of:

1. n vectors of size $(n, 1)$: $e_i, i = 1, \ldots, n$ such that the i-th entry of e_i is one and all others zero,
2. one (n, n) **transition matrix** P, giving the probability of moving from state i to state j, and
3. a vector $\pi_{0i} = \Pr(x_0 = e_i)$ holding the probability of being in state i at time 0.

- $e_1 = \begin{bmatrix} 1 & 0 & \ldots & 0 \end{bmatrix}', e_2 = \begin{bmatrix} 0 & 1 & \ldots & 0 \end{bmatrix}', \ldots$ are just a way of saying "x is in state i".
- The elements of P are

$$P_{ij} = \Pr(x_{t+1} = e_j | x_t = e_i)$$

Assumptions on P and π_0

1. For $i = 1, \ldots, n$, the matrix P satisfies

$$\sum_{j=1}^{n} P_{ij} = 1$$

2. The vector π_0 satisfies

$$\sum_{i=1}^{n} \pi_{0i} = 1$$

- In other words, P is a *stochastic matrix*, where each row sums to one:
  - row i has the probabilities to move to any possible state j. A valid probability distribution must sum to one.
- P defines the probabilities of moving from current state i to future state j.
- π_0 is a valid initial probability distribution.

Transition over two periods

- The probability to move from i to j over two periods is given by P_{ij}^2.
- Why:

$$\begin{aligned}
\Pr\left(x_{t+2} = e_j | x_t = e_i\right) &= \\
\sum_{h=1}^{n} \Pr\left(x_{t+2} = e_j | x_{t+1} = e_h\right) \Pr\left(x_{t+1} = e_h | x_{t+1} = e_i\right) &= \\
\sum_{h=1}^{n} P_{ih} P_{hj} &= P_{ij}^{(2)}
\end{aligned}$$

- Show 3-State example to illustrate this.

Conditional Expectation from a Markov Chain

- What is expected value of x_{t+1} given $x_t = e_i$?
- Simple:

$$\begin{aligned} E\left[x_{t+1}|x_t = e_i\right] &= \text{values of x} \times \text{Prob of those values} \\ &= \sum_{j=1}^{n} e_j \times \Pr\left(x_{t+1} = e_j|e_i\right) \\ &= \begin{bmatrix} x_1 & x_2 & \dots & x_n \end{bmatrix} (P_i)' \end{aligned}$$

  where P_i is the i-th row of P, and $(P_i)'$ is the transpose of that row (i.e. a column vector).
- What is the conditional expectation of a function $f(x)$, i.e. what is

$$E\left[f(x_{t+1})|x_t = e_i\right]?$$

Back to Stochastic DP

- With the Markovian setup, we can rewrite (19) as

$$v(s,\epsilon) = \max_{s' \in \Gamma(s,\epsilon)} u(s,s',\epsilon) + \beta E\left[v(s',\epsilon')|\epsilon\right] \tag{24}$$

Theorem

If $u(s,s',\epsilon)$ is real-valued, continuous, concave, and bounded, if $\beta \in (0,1)$, and constraint set is compact and convex, then

1. *there exists a unique value function $v(s,\epsilon)$ that solves (24).*
2. *there exists a stationary policy function $\phi(s,\epsilon)$.*

Proof.

This is a direct application of Blackwell's sufficiency conditions:

1. with $\beta < 1$ discounting holds for the operator on (24).
2. Monotonicity can be established as before.

Optimality in the Stochastic DP

- As before, we can derive the first order conditions on (24):

$$u_{s'}(s, s', \epsilon) + \beta E\left[V_{s'}(s', \epsilon')|\epsilon\right] = 0$$

- differentiating (24) w.r.t. s to find $V_{s'}(s', \epsilon')$ we find

$$u_{s'}(s, s', \epsilon) + \beta E\left[u_{s'}(s', s'', \epsilon')|\epsilon\right] = 0$$

DP Application 1: The Deterministic Growth Model

- We will now solve the deterministic growth model with dynamic programming.
- Remember:

$$V(k) = \max_{c=f(k)-k' \geq 0} u(c) + \beta V(k') \tag{25}$$

- Assume $f(k) = k^{\alpha}, u(c) = \ln c$.
- We will use *discrete state DP*. We cannot hope to know V at all $k \in \mathbb{R}_+$. Therefore we compute V at a finite set of points, called a *grid*.
- Hence, we must also choose those grid points.

DP Application: Discretize state and solution space

- There are many ways to approach this problem:

$$V(k) = \max_{k' \in [0, k^\alpha]} \ln(k^\alpha - k') + \beta V(k') \tag{26}$$

- Probably the easiest goes like this:
  1. Discretize V onto a grid of n points $\mathcal{K} \equiv \{k_1, k_2, \ldots, k_n\}$.
  2. Discretize control k': change $\max_{k' \in [0, k^\alpha]}$ to $\max_{k' \in \mathcal{K}}$, i.e. choose k' from the discrete grid.
  3. Guess an initial function $V_0(k)$.
  4. Iterate on (26) until $d\left(V_{t+1} - V_t\right) < \varepsilon$, where $d()$ is a measure of distance, and $\varepsilon > 0$ is a tolerance level chosen by you.

References

Jerome Adda and Russell W Cooper. *Dynamic economics: quantitative methods and applications*. MIT press, 2003.

Lars Ljungqvist and Thomas J Sargent. *Recursive macroeconomic theory*. MIT press, 2012.

RE Lucas and NL Stokey. *Recursive Methods in Dynamic Economics*. Harvard University Press, Cambridge MA, 1989.