THOMAS DEMUYNCK

DYNAMIC OPTIMIZATION

Contents

Introduction

In these notes we wil deal with the following class of problems,

$$\begin{aligned} \max_{(a_t)_{t=0}^{\infty}} & \sum_{t=0}^{\infty} \beta^t F(x_t, a_t), \\ \text{subject to } & x_t \in X, \\ & a_t \in A, \\ & a_t \in \Gamma(x_t), \\ & x_{t+1} = r(x_t, a_t), \\ & x_0 \text{ given.} \end{aligned}$$

Such problems contain the following ingredients.

1. A set of states, denoted by X and a set of actions, denoted by A. A state is denoted by $x \in X$ and an action is denoted by $a \in A$.
2. A correspondence $\Gamma : X \to A$ that determines for each state $x \in X$, which actions $a \in \Gamma(x)$ can be taken by the decision maker when the current state is x.[1]
3. An instantaneous payoff function $F(x, a)$ that determines the immediate benefit of taking an action $a \in A$ when the state is $x \in X$.
4. A transition function $r : X \times A \to X$ where $r(x, a)$ gives the state $y = r(x, a) \in X$ in the next period given that the current state is $x \in X$ and the current action is $a \in A$.
5. A discount rate $\beta \in (0, 1)$ that determines the trade-off between future and current payoffs.

[1] Think $\Gamma(x)$ as a budget constraint where x is the vector of prices and income and a is the consumption bundle.

The problem is to determine the optimal (infinite) sequence of actions $a_0, a_1, \ldots$ that should be taken in order to optimize the infinite horizon discounted payoff function:

$$\sum_{t=0}^{\infty} \beta^t F(x_t, a_t),$$

Although this problem may look like a standard optimization problem, there is one key difference. Namely, the optimization problem requires us to find an infinite number of values $(a_t)_{t=0}^{\infty}$ rather than a finite number of values. As such, it is not certain that the usual approach to solve standard optimization problems can also be used to solve this problem.[2]

[2] By usual, we mean the act of setting up the Lagrangian and take the corresponding Kuhn-Tucker first order conditions.

Before we attack the problem in full force, let us start by considering an example. We will choose the Ramsey-Cass-Koopmans model which extended the famous Solow model by permitting elastic savings rates.[3] The Ramsey-Cass-Koopman model is a representative consumer model with endogenous capital formation. In this model, we have an economy where capital is the only input in the production process. The output for a given amount of capital k is determined by a production function:

[3] Ramsey, Frank P. (1928), "A Mathematical Theory of Saving," Economic Journal. 38: 543-559.
Cass, David, (1965), "Optimum Growth in an Aggregative Model of Capital Accumulation," Review of Economic Studies. 32: 233-240.
Koopmans, T. C., (1965), "On the Concept of Optimal Economic Growth," The Economic Approach to Development Planning. Chicago: Rand McNally. pp. 225-287.

$$f(k) = Ak^{\alpha}.$$

Where $\alpha \in (0, 1)$ is the output elasticity of capital. There is a representative household that chooses a sequence of consumption levels $(c_t)_{t=0}^{\infty}$. The period t payoff of choosing c_t gives an instantaneous payoff of

$$u(c_t) = \ln(c_t).$$

The problem faced by the representative household is to choose a sequence of consumption amounts $(c_t)_{t=0}^{\infty}$ and a corresponding sequence of capital holdings $(k_t)_{t=0}^{\infty}$ to maximize discounted lifetime utility,

$$\sum_{t=0}^{\infty} \beta^t \ln(c_t).$$

where $\beta \in (0, 1)$ is an exogenous discount rate. The law of motion for the capital stock is given by:

$$k_{t+1} = f(k_t) - c_t = Ak_t^{\alpha} - c_t.$$

Here k_{t+1} is the stock of capital in period $t+1$. It is equal to the total amount of output, $f(k_t) = Ak_t^{\alpha}$, minus the part of output that is used for immediate consumption, c_t. This law of motion gives a clear trade off. Consumption increases the payoff but decreases future consumption by lowering the next period's amount of capital. The final piece of information to set up the model is a fixed initial level of capital k_0. Combining all pieces, we obtain the following problem,

$$\begin{aligned}
&\max_{(c_t)_{t=0}^{\infty}} \sum_{t=0}^{\infty} \beta^t \ln(c_t),\\
&\text{s.t. } k_{t+1} = Ak_t^{\alpha} - c_t,\\
&\quad k_t, c_t \geq 0,\\
&\quad k_0 \text{ given.}
\end{aligned}$$

Translating this into the dynamic optimization framework from the beginning of the chapter, we obtain the following ingredients.

1. The state space X is given by the possible amounts of capital, $(= \mathbb{R}_+)$. A state is given by a stock of capital $k \in X$. The action space, A, is the possible set of consumption levels $(= \mathbb{R}_+)$.[4] An action is an amount of consumption $c \in A$.

2. The correspondence $\Gamma(k)$ determines the possible consumption levels when the level of capital is equal to k. It is determined by,

$$\Gamma(k) = \{c \in \mathbb{R}_+ : c \leq Ak^{\alpha}\}.$$

3. The instantaneous payoff function is given by, $F(k, c) = \ln(c)$. In this setting, it is independent of the state k (for given c).

4. The transition function $r(k, c)$ that determines the next periods amount of capital is given by $r(k, c) = Ak^{\alpha} - c$.

5. The discount rate is given by β.

[4] Actually, the set of possible consumption levels is $\mathbb{R}_{++}$ as $\ln(0)$ is not defined.

It is instructive to first solve this problem when the time horizon is finite instead of infinite. Let T be the final period. If $T = 0$, we obtain a static optimization problem whose solution depends on the initial capital stock k_0.

$$v_0(k_0) = \max_{c_0} \ln(c_0) \text{ s.t. } k_1 = Ak_0^{\alpha} - c_0; k_1, c_0 \geq 0.$$

Given that $k_1 \geq 0$ and $\ln(.)$ is strictly increasing, the optimal solution is to set $k_1 = 0$ and $c_0 = Ak_0^{\alpha}$.[5] The function $v_0(k_0)$ is called the value function. It only depends only on the initial capital stock as all future capital stocks are determined by the optimal choice of the consumption levels. Substituting $k_1 = 0$ and $c_1 = Ak_0^{\alpha}$ into the problem gives,

$$v_0(k_0) = \ln(Ak_0^{\alpha}) = \ln(A) + \alpha \ln(k_0).$$

[5] As positive amounts of k_1 generate no additional utility, it is optimal to leave no money on the table after the final period.

Now, let look at the problem when the final time period $T = 1$. In this case, we need to choose two consumption levels c_0 and c_1 and we obtain the problem:

$$\begin{aligned} v_1(k_0) &= \max_{c_0,c_1}\{\ln(c_0)+\beta\ln(c_1)\}, \\ \text{s.t. } k_1 &= Ak_0^\alpha - c_0, \\ k_2 &= Ak_1^\alpha - c_1, \\ & c_0, c_1, k_1, k_2 \geq 0. \end{aligned}$$

Given that $k_2 \geq 0$, one clearly sees that $k_2 = 0$ should hold at the optimum.[6] Given this, we can substitute the constraints $c_1 = Ak_1^\alpha$ and $c_0 = Ak_0^\alpha - k_1$ into the objective function.

[6] Again, there should be no money left on the table.

$$v_1(k_0) = \max_{k_1}\{\ln(Ak_0^\alpha - k_1) + \beta\ln(Ak_1^\alpha)\}.$$

The first order condition gives,

$$\begin{aligned} &-\frac{1}{Ak_0^\alpha - k_1} + \beta\alpha\frac{Ak_1^{\alpha-1}}{Ak_1^\alpha} = 0, \\ \rightarrow &-\frac{1}{Ak_0^\alpha - k_1} + \beta\alpha\frac{1}{k_1} = 0, \\ \rightarrow & k_1 = \frac{\alpha\beta}{1+\alpha\beta}Ak_0^\alpha. \end{aligned}$$

The last line gives the optimal solution for k_1. Plugging this solution back into the objective function gives the value of $v_1(k_0)$.

Observe that here $Ak_0^\alpha - k_1 = \frac{1}{1+\alpha\beta}Ak_0^\alpha = c_0 \geq 0$, so the constraints $c_0, c_1, k_1 \geq 0$ are satisfied. Additionally, it is easily verified that the objective function is strictly concave in k_1, so the solution characterized by the first order conditions is a global maximum.

$$\begin{aligned} v_1(k_0) &= \ln\left(Ak_0^\alpha - \frac{\alpha\beta}{1+\alpha\beta}Ak_0^\alpha\right) + \beta\ln\left(A\left(\frac{\alpha\beta}{1+\alpha\beta}Ak_0^\alpha\right)^\alpha\right), \\ &= \ln\left(\frac{A}{1+\alpha\beta}\right) + \alpha\ln(k_0) + \beta\ln\left(\frac{A^{1+\alpha}(\alpha\beta)^\alpha}{(1+\alpha\beta)^\alpha}\right) + \alpha^2\beta\ln(k_0), \\ &= \ln\left(\frac{A}{1+\alpha\beta}\right) + \beta\ln\left(\frac{A^{1+\alpha}(\alpha\beta)^\alpha}{(1+\alpha\beta)^\alpha}\right) + \alpha(1+\alpha\beta)\ln(k_0), \end{aligned}$$

So far so good. extending the final period once more, we set $T = 2$. Then we can write the problem as,[7]

[7] In this case, we can set $k_2 = 0$ and substitute the constraints into the objective function.

$$v_2(k_0) = \max_{k_1,k_2}\{\ln(Ak_0^\alpha - k_1) + \beta\ln(Ak_1^\alpha - k_2) + \beta^2\ln(Ak_2^\alpha)\}.$$

The two first order conditions are,

$$\begin{aligned} \frac{1}{Ak_0^\alpha - k_1} &= \frac{\alpha\beta Ak_1^{\alpha-1}}{Ak_1^\alpha - k_2}, \\ \frac{1}{Ak_1^\alpha - k_2} &= \frac{\alpha\beta Ak_2^{\alpha-1}}{Ak_2^\alpha}. \end{aligned}$$

The solution is,[8]

[8] It is readily verified that this implies $c_0, c_1, c_2, k_1, k_2 \geq 0$. Also the objective function is strictly concave in (k_1, k_2) so the first order conditions are sufficient for a global maximum.

$$\begin{aligned} k_1 &= \frac{\alpha\beta + (\alpha\beta)^2}{1+\alpha\beta+(\alpha\beta)^2}Ak_0^\alpha, \\ k_2 &= \frac{\alpha\beta}{1+\alpha\beta}Ak_1^\alpha. \end{aligned}$$

Substituting these solutions into the objective function gives the value function $v_2(k_0)$. This expression is big mess.[9] We can iterate this procedure, and solve the problem for $T = 3, 4, 5, \ldots$. Doing this, it can be shown that the solution converges for $T \to \infty$ to the values,

[9] Try it.

$$
\begin{aligned}
&v_\infty(k_0) = a + b\ln(k_0), \text{ with,}\\
&a = \frac{1}{1-\beta}\left[\ln(A(1-\alpha\beta)) + \frac{\alpha\beta}{1-\alpha\beta}\ln(A\alpha\beta)\right],\\
&b = \frac{\alpha}{1-\alpha\beta}.
\end{aligned}
$$

This motivate the following procedure to solve the infinite horizon maximization problem: repeatedly solve the dynamic optimization problem for T finite, i.e. $T = 0, 1, 2, 3, \ldots$, and look whether the solution converges when $T \to \infty$.

There are several problems with this approach. First of all, it is not sure whether we will always get a clean functional form for $v_t(k_0)$. In our special setting where $f(k) = Ak^\alpha$ and $u(c) = \ln(c)$, we did have a closed form expression, but this is not the case in general. If we don't have a closed form solution for $v_t(k_0)$ is not clear how we should proceed. Second, even if we obtain a closed form solution, the method is rather cumbersome. We need to solve the optimization problem for various time periods in order to if see some convergence is going on. Third, even assuming that we are able to solve the problem for several finite time periods, it is not certain that these solutions converge to some limiting solution. Let alone that we are able to proof such convergence. Fourth, even if this convergence happens, nothing guarantees us that the limit of the finite horizon optimization problem also provides a solution for the infinite horizon problem. Finally, we have no idea that this limit solution is also the unique solution.

Given the large number of unresolved issues, it might be a good idea to have a fresh look at the initial problem.

$$
v(k_0) = \max_{(c_t)_{t=0}^\infty} \sum_{t=0}^{\infty} \beta^t \ln(c_t) \text{ s.t. } c_t + k_{t+1} \leq Ak_t^\alpha,
$$

Suppose that at time $t = 0$ we choose the value c_0. Then we have that at state 1, capital is equal to $k_1 = Ak_0^\alpha - c_0$. What, then, is the optimal choice of c_1. In order to solve this problem, we need to solve:

$$\begin{aligned}
&\max_{(c_t)_{t=1}^{\infty}} \left\{ \ln(c_0) + \sum_{t=1}^{\infty} \beta^t \ln(c_t) \right\} \text{ s.t. } c_t + k_{t+1} \leq Ak_t^{\alpha}, k_1 \text{ given},\\
&= \ln(c_0) + \beta \max_{(c_t)_{t=1}^{\infty}} \sum_{t=0}^{\infty} \beta^t \ln(c_{t+1}) \text{ s.t. } c_t + k_{t+1} \leq Ak_t^{\alpha}, k_1 \text{ given},\\
&= \ln(c_0) + \beta v(k_1).
\end{aligned}$$

This means that we can replace our original problem by solving:

$$\begin{aligned}
v(k_0) &= \max_{c_0} \{\ln(c_0) + \beta v(k_1)\} \text{ s.t. } k_1 = Ak_0^{\alpha} - c_0,\\
&= \max_{c_0} \{\ln(c_0) + \beta v(Ak_0^{\alpha} - c_0)\}.
\end{aligned}$$

This shows that we can reformulate the infinite horizon problem as a recursive problem. The optimal value $v(k_0)$ for an initial capital stock k_0 is determined by choosing c_0 to maximize current payoff $\ln(c_0)$ and the value of the future payoff which is conveniently written down as $\beta v(k_1) = \beta v(Ak_0^{\alpha} - c_0)$, i.e. the optimal payoff that can be obtained by starting with a capital stock of k_1 tomorrow. The functional equation

$$v(k) = \max_{c \leq Ak^{\alpha}} \{\ln(c) + \beta v(Ak^{\alpha} - c)\}.$$

is called the **Bellman equation** of the dynamic optimization problem.[10] If we could somehow find out the value of the function $v(.)$, we could simply insert it into the right hand side, maximize this right hand side with respect to c and find out the optimal value for c for any initial level of capital k.

[10] A functional equation is an equation of where the unknown is an entire function instead of a single variable.

One way to find out $v(.)$ is to make an educated guess. Before, we found that the limiting value of the value function of the finite horizon problem was of the form $v(k) = a + b\ln(k)$. Substituting this into the Bellman equation gives,

$$a + b\ln(k) = \max_{c}\{\ln(c) + a\beta + b\beta \ln(Ak^{\alpha} - c)\},$$

The maximization problem on the right hand side gives the following first order conditions,[11]

[11] We see that the right hand side is concave in c and $c \geq 0$ so the first order conditions give a global maximum.

$$\begin{aligned}
&\frac{1}{c} - \frac{\beta b}{Ak^{\alpha} - c} = 0,\\
&\to c = \frac{Ak^{\alpha}}{1 + \beta b},\\
&\to Ak^{\alpha} - c = \frac{\beta b}{1 + \beta b} Ak^{\alpha}.
\end{aligned}$$

Plugging this into the Bellman equation gives,

$$a + b\ln(k) = \ln\left(\frac{Ak^{\alpha}}{1 + \beta b}\right) + a\beta + b\beta \ln\left(\frac{\beta b}{1 + \beta b} Ak^{\alpha}\right),$$

Matching up the coefficients on $\ln(k)$ gives,

$$\begin{aligned} b &= \alpha(1+\beta b), \\ \rightarrow b &= \frac{\alpha}{1-\alpha\beta}. \end{aligned}$$

Matching up the constants gives,

$$a = \ln\left(\frac{A}{1+\beta b}\right) + \beta a + \beta b \ln\left(\frac{\beta b A}{1+\beta b}\right).$$

Substituting for b and solving for a finally gives,

$$a = \frac{1}{1-\beta}\left[\ln(A(1-\alpha\beta)) + \frac{\alpha\beta}{1-\alpha\beta}\ln(A\alpha\beta)\right],$$

This gives the same solution as before.[12] In this case, we do have a closed form solution for the value function and for every initial capital stock k we know the optimal consumption level $c = \frac{Ak^{\alpha}}{1+\beta b}$. As such, the optimal solution can be found iteratively,

[12] Eureka!

$$\begin{aligned} c_0 &= \frac{Ak_0^{\alpha}}{1+\beta b}, k_1 = Ak_0^{\alpha} - c_0, \\ c_1 &= \frac{Ak_1^{\alpha}}{1+\beta b}, k_2 = Ak_1^{\alpha} - c_0, \\ &\ldots, \\ c_t &= \frac{Ak_t^{\alpha}}{1+\beta b}, k_{t+1} = Ak_t^{\alpha} - c_t, \\ &\ldots \end{aligned}$$

The tricky part of this approach is, obviously, that we have to guess the functional form of the value function $v(.)$ and there are only a few very specific instances where we can make a good guess about this functional form.

What then should we do if we don't know the form of the value function? Let's go back to the Bellman equation.

$$v(k) = \max_{c \leq f(k)} \{u(c) + \beta v(k')\} \text{ s.t. } k' = Ak^{\alpha} - c.$$

Can we still somehow use this equation to solve our problem? The answer is yes and the key to the solution lies in the "recursiveness" of the equation.

Assume that we start with an "arbitrary" guess for the function $v(.)$, say $v_0(.)$. We know that v_0 does not satisfy the Bellman equation, but let us substitute it into the right hand side anyway. Doing this gives us on the left hand side a new function, say $v_1(.)$.[13]

[13] Observe that we start with a function $v_0(.)$ and get an entire new function $v_1(.)$ out of this by varying the level of k on the left and right hand side.

$$v_1(k) = \max_c \{u(c) + \beta v_0(Ak^\alpha - c)\}.$$

Now, we can do the same thing with $v_1(.)$: plug it into the right hand side of the Bellman equation and look at the values that it generates on the left hand, giving us a new function $v_2(.)$.

$$v_2(k) = \max_c \{u(c) + \beta v_1(Ak^\alpha - c)\}.$$

We can continue this process indefinitely, and generate functions $v_1(.), v_2(.), v_3(.), \ldots, v_n(.), \ldots$ What happens if we allow $n \to \infty$. We would hope that finally the function $v_n(.)$ converges to some limiting function v_∞ that satisfies our Bellman equation,

$$v_\infty(k) = \max_{c \leq f(k)} \{u(c) + \beta v_\infty(Ak^\alpha - c)\}.$$

This is the function we were looking for all along. Of course, currently, we don't know whether this iteration will converge to something useful or even that different starting functions for $v_0(.)$ will converge to the same limiting function $v_\infty(.)$. Studying the conditions for which this iteration does converge is the main objective of the theory developed in these notes.

Mathematical Preliminaries

In this chapter, we will introduce the necessary mathematical tools and results for the following chapters. We will need to have a look at the concepts of vector spaces and normed vector spaces. A special subclass of these spaces have the property that every Cauchy sequence has a limit, called Banach spaces.

Banach spaces will provide the necessary structure for our space of value functions. We will define contraction mappings on these spaces and show that these have a unique fixed point. Additionally, we will present a useful result called Blackwell's theorem that gives an easy to verify set of conditions for a mapping to be a contraction mapping.

In a second part of the chapter, we will have a look at the theorem of the maximum. This celebrated result in economics gives us conditions for which the result of a maximization exists and satisfies some convenient continuity conditions.

Banach spaces

Before we can introduce the concept of a Banach space, we first need to define vector spaces.

Definition 1 (vector space)**.** *A* ***real vector space*** *X is a set of elements together with two operations, addition and scalar multiplication.*[14] *For any two vectors $x, y \in X$, addition gives a vector $x + y \in X$ and for any vector $x \in X$ and a real number $\alpha \in \mathbb{R}$, scalar multiplication gives $\alpha x \in X$. We have the following conditions on the operations of a vector space:*

[14] The adjective real simply indicates that scalar multiplication is defined taking the reals, not elements of the complex plane or some other set.

1. $x + y = y + x$;
2. $(x + y) + z = x + (y + z)$
3. $\alpha(x + y) = \alpha x + \alpha y$,
4. $(\alpha + \beta)x = \alpha x + \beta x$,
5. $(\alpha\beta)x = \alpha(\beta x)$.

6. $1x = x$.

Additionally, there is a unique zero element $\theta \in X$ such that,

7. *$x + \theta = x$ for all $x \in X$*

8. *for every $x \in X$ there is a $-x$ such that $x + (-x) = \theta$*

Calculating with vectors produces what you would expect. First of all, notice that $0x = \theta$ for all x. Indeed we have the derivation:

$$\begin{aligned}
0x &= \theta + 0x \\
&= x + (-x) + 0x \\
&= (x + 0x) + (-x) \\
&= (1 + 0)x + (-x) \\
&= 1x + (-x) \\
&= x + (-x) \\
&= \theta.
\end{aligned}$$

Next, notice that $-x = (-1)x$. Indeed:

$$\begin{aligned}
(-1)x &= \theta + (-1)x \\
&= (x + (-x)) + (-1)x, \\
&= (1x + (-1)x) + (-x) \\
&= (1 - 1)x + (-x) \\
&= 0x + (-x) \\
&= \theta + (-x) \\
&= (-x).
\end{aligned}$$

So $\alpha x + (\beta)(-y)$ is equal to $\alpha x + (-\beta)y$ which we simply denote by $\alpha x - \beta y$.

A first well known example of a vector space is the set of n-dimensional real vectors $\mathbb{R}^n$.[15] However, the concept of a vector space is much broader than vectors of numbers.

[15] Verify that this set satisfies all conditions.

We will mainly work with vector spaces that have real valued functions as elements. Consider two functions $f : D \to \mathbb{R}$ and $g : D \to \mathbb{R}$ defined on some common domain D. Then we can define their sum $f + g$ as the function $h : D \to \mathbb{R}$ such that:

$$h(x) \equiv (f + g)(x) = f(x) + g(x),$$

and the scalar product, αf ($\alpha \in \mathbb{R}$) as the function $h : D \to \mathbb{R}$ such that:

$$h(x) \equiv (\alpha f)(x) = \alpha f(x).$$

It is clear that these operations satisfy all eight conditions of a vector space.[16] As such, the set $F(D)$ of real valued functions on a common

[16] Here we define the null-vector θ to be the function $\theta(x) = 0$ for all $x \in X$.

domain D forms a vector space. We can actually go further. If D is a topological space and if f and g are bounded, continuous functions from D to $\mathbb{R}$, then $f+g$ and αf are also bounded and continuous functions, so the set of all bounded, continuous real valued functions with domain D is also a vector space. Let us call this space $C(D)$.[17]

[17] A function $f : D \to \mathbb{R}$ is bounded if there exists a number $M > 0$ such that for all $x \in D$, $|f(x)| \le M$. Notice that if D is compact, then any continuous function $f : D \to \mathbb{R}$ is bounded.

The following are real vector spaces:

- The finite Euclidean space $\mathbb{R}^n$
- The set $X = \{x \in \mathbb{R}^2 : x = \alpha z\}$, where $z \in \mathbb{R}^2$
- The set of all continuous functions on $[a, b]$,

The following are not vector spaces

- The unit circle in $\mathbb{R}^2$ with the usual addition
- the set of all integers
- The set of non-negative functions on $[a, b]$.

The following are normed vector spaces:

- $\mathbb{R}^n$ with $\|x\| = \left(\sum_{i=1}^n x_i^2\right)^{1/2}$.
- $\mathbb{R}^n$ with $\|x\| = \max_i |x_i|$,
- $\mathbb{R}^n$ with $\|x\| = \sum_{i=1}^n |x_i|$.
- The set of all bonded infinite sequences $(x_1, \dots)$ with $\|x\| = \sup_k |x_k|$ this space is called ℓ_∞.
- The set of continuous functions on $[a, b]$ with $\|x\| = \sup_{a \le t \le b} |x(t)|$ this space is called $C[a, b]$.
- The set of continuous functions on $[a, b]$ with $\|x\| = \int_a^b |x(t)| dt$.

We are now ready to define the notion of a norm on a vector space.

Definition 2 (normed vector space)**.** *A **norm** on a vector space X is a function $\|.\| : X \to \mathbb{R}_+$ such that for all $x, y \in X$ and $\alpha \in \mathbb{R}$,*

- $\|x\| \ge 0$, *with equality if and only if* $x = \theta$,
- $\|\alpha x\| = |\alpha| \|x\|$,
- $\|x + y\| \le \|x\| + \|y\|$.

A vector space X together with a norm $\|.\|$ is called a normed vector space. Intuitively, the idea is that $\|x - y\|$ measures the distance between x and y. In particular, $\|x\|$ measures the distance between the zero element θ and x given that $\|x\| = \|x - \theta\|$. The last condition is called the triangle inequality. Substituting x by $x - z$ and y by $z - y$ gives:

$$\|x - y\| \le \|x - z\| + \|z - y\|.$$

In other words, the distance between x and y is always smaller than the distance between x and z plus the distance from z to y.

Now, consider our previously defined vector space $C(D)$ of continuous, bounded real valued functions on a common domain D. What would be a good norm on this set. In other words, if we take two bounded and continuous functions $f : D \to \mathbb{R}$ and $g : D \to \mathbb{R}$, how can we measure the 'distance' $\|f - g\|$ between these two functions?

A first idea would be to take one particular value $x_0 \in X$ and to define,

$$\|f - g\| = |f(x_0) - g(x_0)|.$$

In particular $\|f\| = |f(x_0)|$. The problem with this 'norm', however, is that it does not satisfy the first condition: it is possible that $|f(x_0) - g(x_0)| = 0$, i.e. $f(x_0) = g(x_0)$, but f is not equal to g the zero function. This can be fixed by taking the maximal distance between f and g over the set D,

$$\|f - g\| = \max_{x \in D} |f(x) - g(x)| \text{ in particular } \|f\| = \max_{x \in D} |f(x)|.$$

The problem with this proposal is that the maximum may not exist (if for example D is not compact). We can solve this by taking the supremum instead of the maximum.[18]

[18] The sup exists because we assumed that both f and g are bounded.

$$\|f - g\| = \sup_{x \in D} |f(x) - g(x)| \text{ in particular } \|f\| = \sup_{x \in D} |f(x)|.$$

This metric is called the sup or infinity norm.[19]

[19] Show that it satisfies all three conditions to be a norm.

For the analysis in the next chapters, it will be useful to generalize the notion of the sup-norm. Let us go back to the set of continuous functions on the set D. Let $\phi : D \to \mathbb{R}_{++}$ be a continuous function that takes only strictly positive values. For such given function ϕ, we consider the set of functions $C_\phi(D)$ for which the function $\frac{f(x)}{\phi(x)}$ is bounded. In other words, $f \in C_\phi(D)$ if f is there exists an M such that for all $x \in D$, $\frac{f(x)}{\phi(x)} \leq M$. If so, we say that f is bounded in the ϕ-norm.

For these functions, we can consider the following norm,

$$\|f\|_\phi = \sup_{x \in D} \frac{|f(x)|}{\phi(x)}.$$

Let us first show that this is indeed a norm. First,

$$\|f\|_\phi \geq 0,$$

is easily established.[20] If $\|f\|_\phi = 0$. Then we have that for all $x \in D$,

$$0 = \frac{|f(x)|}{\phi(x)}.$$

Given that $\phi(x) > 0$, we have indeed that $f(x) = 0$ for all $x \in D$, so f is the zero function. Next,

$$\|\alpha f\|_\phi = \sup_{x \in D} \frac{|\alpha f(x)|}{\phi(x)} = |\alpha| \sup_{x \in D} \frac{|f(x)|}{\phi(x)} = |\alpha| \|f\|_\phi.$$

finally,

$$\begin{aligned} \|f + g\|_\phi &= \sup_{x \in D} \frac{|f(x) + g(x)|}{\phi(x)}, \\ &\leq \sup_{x \in D} \frac{|f(x)| + |g(x)|}{\phi(x)}, \\ &\leq \sup_{x \in D} \frac{|f(x)|}{\phi(x)} + \sup_{x \in X} \frac{|g(x)|}{\phi(x)}, \\ &= \|f\|_\phi + \|g\|_\phi. \end{aligned}$$

Observe that if we consider the constant function $\phi(x) = 1$ for all $x \in D$, then $\|f\|_\phi = \|f\|$. As such, the sup norm is a special case of the ϕ-norm with $\phi(x) = 1$ for all x. However, the ϕ-norm covers other cases too. Consider, for example $D = \mathbb{R}$ and $\phi(x) = |x| + 1$ then we see that f is bounded in the norm $\|.\|_\phi$, if f does not grow faster than $|x|$.[21] In other words, f can be unbounded but not 'more' unbounded than the function $\phi(x) = |x| + 1$. In general f will be bounded in the ϕ-norm if the value of $|f(x)|$ does not 'grow' faster than the value of $\phi(x)$.

We have that $C_\phi(D)$ together with the norm $\|.\|_\phi$ is a normed vector space.

[20] Indeed, both $|f(x)| \geq 0$ and $\phi(x) > 0$.

[21] For example, $f(x) = x^2$ is not bounded using this norm on $\mathbb{R}$. but $f(x) = ax + b$ is bounded although $f(x) = ax + b$ is not bounded in the sup-norm: $\sup_{x \in \mathbb{R}} |f(x)| = \infty$

A MAIN REASON for introducing norms is to measure distance between different elements of a vector space. Once we can measure distances, we can also start talking about convergence.

Definition 3 (convergence). *Let $(X, \|.\|)$ be a normed vector space. A sequence $(x_n)_{n\in\mathbb{N}}$ of elements in X is said to **converge** to an element $x \in X$ if for all $\varepsilon > 0$, there exists a N_ε such that for all $n \geq N_\varepsilon$,*[22]

$$\|x_n - x\| < \varepsilon.$$

We also write this as $x_n \xrightarrow{n} x$ or $\lim_{n\to\infty} x_n = x$.

[22] We write N_ε to make it clear that N_ε may be different for different values of ε.

In words, a sequence $(x_n)_{n\in\mathbb{N}}$ converges to an element x if for all strictly positive numbers ε, it is possible to go far enough in the sequence, say further than the N_ε'th element such that for all elements x_n beyond this element the distance between x_n and x is smaller than ε.[23]

[23] Alternatively, you could say that for any strictly positive number ε there are only a finite number of elements in the sequence $(x_n)_{n\in\mathbb{N}}$ that are at a distance greater than ε from x.

Next, we need the definition of a Cauchy sequence.

Definition 4 (Cauchy sequence). *Let $(X, \|.\|)$ be a normed vector space. A sequence $(x_n)_{n\in\mathbb{N}}$ in S is a **Cauchy sequence** if for all $\varepsilon > 0$, there is a number N_ε such that for all $n, m \geq N_\varepsilon$,*

$$\|x_n - x_m\| < \varepsilon.$$

So a sequence $(x_n)_{n\in\mathbb{N}}$ is a Cauchy sequence if for any strictly positive number ε it is possible to go far enough in the sequence, further than N_ε such that the distance between any two elements beyond the N_ε'th position is less than ε.

Exercises:

- Show that if $x_n \to x$ and $x_n \to y$ then $x = y$.
- Show that if a sequence is convergent, then it is a Cauchy sequence.
- Show that $x_n \to x$ if and only if every subsequence of $(x_n)_{n\in\mathbb{N}}$ converges to x.

Complete normed spaces

IT IS ALWAYS the case that a convergent sequence $x_n \xrightarrow{n} x$ in a normed vector space is also a Cauchy sequence. The reverse, however is not necessarily the case. In other words, it is possible that $(x_n)_{n\in\mathbb{N}}$ is a Cauchy sequence, but it does not converge to an element in X. Normed vector spaces where this is true are called complete vector spaces.

Definition 5 (complete metric spaces). *A normed vector space $(X, \|.\|)$ is complete if every Cauchy sequence in X converges to an element in X.*

Not every vector space is complete. As an example, consider the set $C([0,1])$ of continuous, bounded functions on the closed interval $[0,1]$ and consider the L^2 norm:

$$\|f\| = \left(\int_0^1 |f(t)|^2 dt\right)^{1/2}.$$

Let f_n be the step function that is equal to 0 on the interval $[0, 1-2^{-n}]$ and linearly rises to 1 on the interval $[1-2^{-n}, 1]$. One can show that this is a Cauchy sequence. However, its limit is not a continuous function.

Intuitively, a complete vector space is a space without any 'points' missing, where the missing points could either lie inside or at its boundary. We take it as a fact that the set of real numbers $\mathbb{R}$ with the norm $|x-y|$ is a complete vector space.[24] A complete normed vector space is also called a **Banach space**. We will use the term Banach space from now on.

[24] This is a consequence of the Bolzano-Weierstrass theorem.

The following theorem shows that $C_\phi(D)$ is a Banach space.

Theorem 1. *Let $\phi : D \to R_{++}$ be a continuous function and let $C_\phi(D)$ be the set of all continuous functions $f : D \to \mathbb{R}$ that are bounded in the ϕ-norm, $\|f\|_\phi = \sup_{x\in D} \frac{f(x)}{\phi(x)}$. Then $C_\phi(D)$ is a Banach space.*

Proof. That $C_\phi(D)$ is a normed vector space was shown above. Let $(f_n)_{n\in\mathbb{N}}$ be a Cauchy sequence in $C_\phi(D)$. We need to show that there exists an $f \in C_\phi(D)$ such that:

$$\lim_{n\to\infty} f_n = f \text{ or equivalently} \|f_n - f\|_\phi \xrightarrow{n} 0.$$

There are three steps. First, we find a candidate function f, second, we show that $(f_n)_{n\in\mathbb{N}}$ converges to this candidate function (in the $\|.\|_\phi$ norm). Third we show that the candidate function is in $C_\phi(D)$.

For step one, fix $x \in X$, then the sequence of real numbers $f_n(x)$ satisfies,

$$\begin{aligned}
|f_n(x) - f_m(x)| &= \phi(x)\frac{|f_n(x) - f_m(x)|}{\phi(x)}, \\
&\leq \phi(x) \sup_{y\in D} \frac{|f_n(y) - f_m(y)|}{\phi(y)}, \\
&= \phi(x)\|f_n - f_m\|_\phi \xrightarrow{n,m} 0.
\end{aligned}$$

As such, the sequence $(f_n(x))_{n\in\mathbb{N}}$ is a Cauchy sequence in $\mathbb{R}$. As $\mathbb{R}$ is complete, the sequence $(f_n(x))_{n\in\mathbb{N}}$ has a limit, call it $f(x)$, i.e. $\lim_{n\to\infty} f_n(x) = f(x)$.[25] Doing this for all $x \in D$, defines a function $f : D \to \mathbb{R}$ that we take to be our candidate function.

[25] This keeps x fixed and regards $(f_n(x))_{n\in\mathbb{N}}$ as a sequence of numbers in $\mathbb{R}$.

For step 2, we need to show that $\|f_n - f\|_\phi \xrightarrow{n} 0$. Let $\varepsilon > 0$ and let N be such that for $n, m \geq N$, $\|f_n - f_m\|_\phi < \varepsilon$. Then for all $n \geq N$

$$\begin{aligned}
\frac{|f_n(x) - f(x)|}{\phi(x)} &= \frac{|f_n(x) - \lim_m f_m(x)|}{\phi(x)}, \\
&= \lim_m \frac{|f_n(x) - f_m(x)|}{\phi(x)}, \\
&\leq \lim_m \|f_n - f_m\|_\phi < \varepsilon.
\end{aligned}$$

This holds for all x. As such, $\|f_n - f\|_\phi < \varepsilon$.

Finally, we need to show that $f \in C_\phi(D)$. Boundedness of $\|f\|_\phi$ is obvious.[26] Let us first show that $\frac{f(x)}{\phi(x)}$ is continuous. As $\phi(x)$ is continuous, this also shows that $f(x)$ is continuous. So, let us show that if $x_n \xrightarrow{n} x$ then $f(x_n)/\phi(x_n) \to f(x)/\phi(x)$.

[26] This follows from the fact that the sequence $(f_n)_{n\in\mathbb{N}}$ is a Cauchy sequence and Cauchy sequences are bounded.

$$\begin{aligned}
\left|\frac{f(x_n)}{\phi(x_n)} - \frac{f(x)}{\phi(x)}\right| &= \left|\frac{f(x_n)}{\phi(x_n)} - \frac{f_m(x_n)}{\phi(x_n)} + \frac{f_m(x_n)}{\phi(x_n)} - \frac{f_m(x)}{\phi(x)} + \frac{f_m(x)}{\phi(x)} - \frac{f(x)}{\phi(x)}\right|, \\
&\le \frac{|f(x_n) - f_m(x_n)|}{\phi(x_n)} + \left|\frac{f_m(x_n)}{\phi(x_n)} - \frac{f_m(x)}{\phi(x)}\right| + \frac{|f_m(x) - f(x)|}{\phi(x)}, \\
&\le \|f - f_m\|_\phi + \left|\frac{f_m(x_n)}{\phi(x_n)} - \frac{f_m(x)}{\phi(x)}\right| + \|f_m - f\|_\phi.
\end{aligned}$$

The first and last term can be set arbitrarily small by picking m large enough as $(f_m)_{m\in\mathbb{N}}$ converges to f. The middle term goes to zero by continuity of f_m and ϕ. □

When we take $\phi(x) = 1$ for all x, this theorem shows that $C(D)$ being the set of all bounded, continuous functions is also a Banach space.

Corollary 1. *Let $D \subseteq \mathbb{R}^n$ and let $C(D)$ be the set of all bounded, continuous functions $f : X \to \mathbb{R}$ with the sup norm $\|f\| = \sup_{x\in D} |f(x)|$. Then $C(D)$ is a Banach space.*

Contraction mappings

Now that we are equipped with the notion of a Banach space, we can have a look at contraction mappings.

Definition 6 (contraction mapping)**.** *Let $(X, \|.\|)$ be a normed vector space and let $T : X \to X$ be a function mapping X into itself. The operator T is a **contraction mapping** with modulus $\beta \in [0, 1[$ if for all $x, y \in X$:*

$$\|T(x) - T(y)\| \le \beta\|x - y\|.$$

A function is a contraction mapping if the distance between the two images of points x and y are closer together than the original points x and y. Intuitively, when we iterate such mapping, the points will at each step come closer and closer together. Eventually, we expect these iterations to converge to what we call a **fixed point**.

Definition 7. *Let $(X, \|.\|)$ be a normed vector space and let $T : X \to X$. Then $x \in X$ is called a fixed point of T if*

$$T(x) = x.$$

Let $C_\phi(D)$ be our set of continuous functions on D that are bounded in the ϕ-norm. A mapping T from $C_\phi(D)$ to $C_\phi(D)$ takes a function $f \in C_\phi(D)$ as input and produces another function $g = T(f) \in C_\phi(D)$. A fixed point of T is a function $f \in C_\phi(D)$ such that T maps f to itself: $f = T(f)$. A function T that takes functions to functions is called, for clarity, an operator. Often we omit the brackets when using operators, so we write Tf instead of $T(f)$. If we are interested in the value of $T(f)$ at a particular point $x \in D$, we can write this as $Tf(x)$ or sometimes (to avoid confusion) as $(Tf)(x)$.

The following important results states that every contraction mapping (operator) on a Banach space has a unique fixed point.

Theorem 2 (Banach's contraction mapping theorem)**.** *Let $(X, \|.\|)$ be a Banach space and let $T : X \to X$ be a contraction mapping on X with modulus β, then*

- *T has exactly one fixed point $x \in X$,*
- *For any $x_0 \in X$, $\|(T^n x_0) - x\| \leq \beta^n \|x_0 - x\|$.*

Proof. Define the iterates of T, the mappings $(T^n)_{n\in\mathbb{N}}$ by $T^0 x = x$, $T^n x = T(T^{n-1}x) = (T \circ T^{n-1})(x)$. Choose $x_0 \in S$ and let $(x_n)_{n\in\mathbb{N}}$ be defined as $x_n = T^n x_0$. By the contraction mapping property,

$$\|x_2 - x_1\| = \|Tx_1 - Tx_0\| \leq \beta \|x_1 - x_0\|.$$

By induction, we can show that,

$$\|x_{n+1} - x_n\| \leq \beta^n \|x_1 - x_0\|.$$

As such, for any $m \geq n$,

$$\begin{aligned}
\|x_m - x_n\| &\leq \|x_m - x_{m-1}\| + \|x_{m-1} - x_{m-2}\| + \ldots + \|x_{n+1} - x_n\|, \\
&\leq \left[\beta^{m-1} + \ldots, + \beta^{n+1} + \beta^n\right] \|x_1 - x_0\|, \\
&= \beta^n \left[\beta^{m-n-1} + \ldots + \beta + 1\right] \|x_1 - x_0\|, \\
&\leq \frac{\beta^n}{1-\beta} \|x_1 - x_0\| \xrightarrow{n} 0.
\end{aligned}$$

This shows that $(x_n)_{n\in\mathbb{N}}$ is a Cauchy sequence, so it has a limit $x_n \xrightarrow{n} x \in X$. To show that $Tx = x$ note that

$$\begin{aligned}
\|Tx - x\| &\leq \|Tx - T^n x_0\| + \|T^n x_0 - x\|, \\
&\leq \beta \|x - T^{n-1} x_0\| + \|T^n x_0 - x\|.
\end{aligned}$$

Both terms on the right hand side converge to zero as $n \to \infty$ and the left hand side is independent of n. As such, $\|Tx - x\| = 0$, or equivalently, $Tx = x$. For uniqueness, assume that $x, \hat{x}$ are both fixed points of T, then

$$\|\hat{x} - x\| = \|T\hat{x} - Tx\| \leq \beta\|\hat{x} - x\|.$$

This can only be true if $\|\hat{x} - x\| = 0$ or $\hat{x} = x$. □

It will often be convenient to restrict the region in the set X where the fixed point is situated. Let $(X, \|.\|)$ be a Banach space and let X' be a closed subset of X. It can be shown that the smaller set $(X', \|.\|)$ is also a Banach space.[27] If $T : X \to X$ is a contraction mapping and if T maps X' to X' then T is also contraction mapping on the smaller set $(X', \|.\|)$.[28] As X' is closed, the unique fixed point of T will lie in X'. This is the gist of the following lemma.

[27] A set X' is closed if for all sequences $(x_n)_{n\in\mathbb{N}}$ in X', $x_n \xrightarrow{n} x$ (according to the norm $\|.\|$) implies that $x \in X'$.

[28] This requires that $T(X') \subseteq X'$.

Lemma 1. *Let $(X, \|.\|)$ be a Banach space and let $T : X \to X$ be a contraction mapping with fixed point $x^* \in X$. If X' is a closed subset of X and $T(X') \subseteq X'$, then $x^* \in X'$. If in addition $T(X') \subseteq X'' \subseteq X'$ then $x^* \in X''$.*

Proof. Choose $x_0 \in X'$ and note that $(T^n x_0)_{n\in\mathbb{N}}$ is a sequence in X' converging to the fixed point x^* of T. Since X' is closed, it follows that $x^* \in X'$, so the unique fixed point is also in X'. If in addition $T(X') \subseteq X''$ then it follows that $x^* = Tx^* \in X''$ so x^* is also in X''. □

The second part of the contraction mapping theorem provides a bound on the distance from the n-th iterate $T^n x_0$ to the fixed point x,

$$\|T^n x_0 - x\| \leq \beta^n \|x_0 - x\|.$$

This bound, however, is not very useful as it involves the 'unknown' limit x. The following gives a computationally more relevant bound.

Lemma 2. *Let $(X, \|.\|)$ be a Banach space, T a contraction mapping and x the fixed point of T. Then*

$$\|T^n x_0 - x\| \leq \frac{1}{1-\beta}\|T^n x_0 - T^{n+1} x_0\|.$$

Proof. Notice that,

$$\begin{aligned} \|T^n x_0 - x\| &\leq \|T^n x_0 - T^{n+1} x_0\| + \|T^{n+1} x_0 - x\|, \\ &\leq \|T^n x_0 - T^{n+1} x_0\| + \beta\|T^n x_0 - x\|. \end{aligned}$$

Rearranging this inequality gives the desired result. □

Previously, we saw that the set of continuous functions $f : D \to \mathbb{R}$ that are bounded in the $\|.\|_\phi$ norm, i.e. $C_\phi(D)$ was a Banach space.

We will mainly be interested in contraction mappings from $C_\phi(D) \to C_\phi(D)$. These contraction mappings take functions in $C_\phi(D)$ to other functions in $C_\phi(D)$. The following theorem, known as Blackwell's theorem provides sufficient, easy to verify, conditions for an operator T to be a contraction mapping on $C_\phi(D)$.

Theorem 3 (Blackwell's sufficient conditions)**.** *Let $\phi : D \to R_{++}$ be a continuous function and let $C_\phi(D)$ be the space of continuous functions $f : D \to \mathbb{R}$, that are bounded in the ϕ-norm. Let $T : C_\phi(D) \to C_\phi(D)$ be an operator satisfying:*

- *(monotonicity) If $f, g \in C_\phi(D)$ and $f(x) \leq g(x)$ for all $x \in D$, then $(Tf)(x) \leq (Tg)(x)$ for all $x \in D$.*
- *(discounting) There is a $\beta \in (0,1)$ such that for all $x \in D$, $f \in C_\phi(D)$ and $a \geq 0$:*

$$T(f + a\phi)(x) \leq (Tf)(x) + (\beta a)\phi(x),$$

Then T is a contraction with modulus β.

Proof. Observe that

$$\begin{aligned} \frac{f(x)}{\phi(x)} &= \frac{g(x)}{\phi(x)} + \frac{f(x) - g(x)}{\phi(x)}, \\ &\leq \frac{g(x)}{\phi(x)} + \frac{|f(x) - g(x)|}{\phi(x)}, \\ &\leq \frac{g(x)}{\phi(x)} + \|f - g\|_\phi. \end{aligned}$$

as such, multiplying both sides by $\phi(x) > 0$ gives,

$$f(x) \leq g(x) + \phi(x)\|f - g\|_\phi.$$

So, by monotonicity and discounting:

$$(Tf)(x) \leq T(g + \|f - g\|_\phi \phi)(x) \leq (Tg)(x) + \beta\|f - g\|_\phi \phi(x).$$

Equivalently,

$$\frac{(Tf)(x) - (Tg)(x)}{\phi(x)} \leq \beta\|f - g\|_\phi.$$

Reversing the roles of f and g gives

$$\frac{(Tg)(x) - (Tf)(x)}{\phi(x)} \leq \beta\|f - g\|_\phi.$$

This holds for all x and the right hand side does not depend on x, so taking the sup on the left hand side gives:

$$\|Tf - Tg\|_\phi \leq \beta\|f - g\|_\phi,$$

so T is a contraction mapping with modulus β. □

Applying above theorem to the case $\phi(x) = 1$,[29] we obtain the following, better known, version of Blackwell's theorem.

[29] This is the case where $\|.\|_\phi$ is the sup norm and $C(X)$ is the set of bounded, continuous functions on X.

Corollary 2 (Blackwell's sufficient conditions)**.** *Let $D \subseteq \mathbb{R}^l$ and let $B(D)$ be the space of bounded functions $f : D \to \mathbb{R}$, with the sup norm. Let $T : C(D) \to C(D)$ be an operator satisfying:*

- *(monotonicity) If $f, g \in C(D)$ and $f(x) \leq g(x)$ for all $x \in D$, then $(Tf)(x) \leq (Tg)(x)$ for all $x \in D$.*
- *(discounting) There is a $\beta \in (0,1)$ such that for all $x \in D$, all $f \in C(D)$ an all $a \geq 0$.*

  $$T(f + a)(x) \leq (Tf)(x) + \beta a,$$

  for all $f \in B(D)$ and $a \geq 0$.

Then T is a contraction with modulus β.

Theorem of the maximum

In the second part of this chapter we'll have a look at a seminal result in mathematics, the theorem of the maximum. Consider two sets $X \subseteq \mathbb{R}^l$, and $A \subseteq \mathbb{R}^m$, and let $f : X \times A \to \mathbb{R}$ be a real valued function that takes a vector $x \in X$, a vector in $a \in A$ and produces a real number $f(x, a)$. Here A will be the set of decision variables (actions), while X will denote the set of parameters (states).

Let $G : X \to A$ be a non-empty correspondence.[30] The theorem of the maximum deals with optimization problems of the following form,

$$\max_{a \in G(x)} f(x, a).$$

[30] A correspondence $\Gamma : X \to A$ is a mathematical object that takes a vector $x \in X$ and delivers a subset $\Gamma(x) \subseteq A$.

This problem optimizes a function $f(x, a)$ with respect to a, when a is restricted to lie in the set $G(x)$. In the optimization problem, the value x is kept fixed, so it is a parameter of the optimization problem. If for each x, $f(x, a)$ is continuous in a and the set $G(x)$ is nonempty and compact,[31] then for all x the maximum is attained.[32] In this case, we can define the function

$$v(x) = \max_{a \in G(x)} f(x, a),$$

and the correspondence:

$$\Gamma(x) = \{a \in G(x) : f(x, a) = v(x)\},$$

of values in $G(x)$ that attain this maximum. We would like to place additional restrictions such that the function v and the set Γ vary 'continuously' with the 'parameter' x.

[31] Here, compactness means that for each $x \in X$, $G(x)$ is closed and bounded.

[32] This follows from the extreme value theorem.

Towards this end, we need to define the concepts of lower and upper hemi-continuity.

Definition 8 (Lower hemi-continuity). *The correspondence $G : X \to A$ is* ***lower hemi-continuous*** *(l.h.c.) at $x \in X$ if*

1. *$G(x)$ is non-empty*
2. *for every $a \in G(x)$ and every sequence $x_n \xrightarrow{n} x$, there exists an $N \geq 1$ and a sequence $(a_n)_{n \geq N}$ such that $a_n \xrightarrow{n} a$ and $a_n \in G(x_n)$ for all $n \geq N$.*[33]

[33] If $G(x_n)$ is nonempty for all n, we can always take $N = 1$.

Definition 9 (Upper hemi-continuity). *A compact-valued correspondence $G : X \to A$ is* ***upper hemi-continuous and compact valued*** *(u.h.c.) at $x \in X$ if for every sequence $x_n \xrightarrow{n} x$ and every sequence $(a_n)_{n \geq N}$ such that $a_n \in G(x_n)$ for all n, the sequence $(a_n)_{t \in \mathbb{N}}$ is bounded and if $(a_n)_{n \in N}$ converges (say to a), then its limit point a is in $G(x)$.*

Definition 10 (continuity). *A correspondence $G : X \to A$ is continuous at $x \in X$ if it is both u.h.c. and l.h.c. at x. A correspondence is continuous if it is continuous at each point in its domain.*

The following lemma is a well known result concerning convergence of sequences and will be useful in the proof of the following theorem.

Lemma 3. *Let $v : \mathbb{R}^n \to \mathbb{R}$ be a real valued function. The function v is continuous at x if and only if for all sequences $(x_n)_{n \in \mathbb{N}}$ with $x_n \xrightarrow{n} x$ there is a subsequence $(x_{\varphi(n)})_{n \in \mathbb{N}}$ such that $v(x_{\varphi(n)}) \xrightarrow{n} v(x)$.*

Proof. ($\rightarrow$) Let v be continuous and $x_n \xrightarrow{n} x$. Then evidently $v(x_n) \xrightarrow{n} v(x)$ so for all subsequences $(x_{\varphi(n)})_{n \in \mathbb{N}}$ of $(x_n)_{n \in \mathbb{N}}$ we should have that $v(x_{\varphi(n)}) \xrightarrow{n} v(x)$.

($\leftarrow$) For the reverse, assume that v is not continuous at x. Then there is a sequence $(x_n)_{n \in \mathbb{N}}$ such that $v(x_n) \stackrel{n}{\not\to} v(x)$. From this, we will construct a sequence $z_t \to x$ that has no subsequence $z_{\varphi(n)}$ such that $v(z_{\varphi(n)})$ converges to $v(x)$.

As $v(x_n) \not\to v(x)$, there exists a $\varepsilon > 0$ such that for all T, there is an $n \geq T$ such that $|v(x) - v(x_n)| > \varepsilon$. This generates a sequence of numbers $n_1, n_2, n_3, \ldots$ such that for all k, $|v(x) - v(x_{n_k})| > \varepsilon$. Without loss of generality, we can assume that this sequence of numbers is increasing. Let $z_k = x_{n_k}$. We see that $(z_n)_{n \in N}$ is a sequence such that $z_n \xrightarrow{n} x$. Additionally, $(v(z_n))_{n \in \mathbb{N}}$ has no subsequence $(v(z_{\varphi(n)}))_{n \in \mathbb{N}}$ that converges to $v(x)$ as fro every n, $|v(z_n) - v(x)| \geq \varepsilon$. $\square$

Theorem 4 (Theorem of the maximum). *Let $X \subseteq \mathbb{R}^l, A \subseteq \mathbb{R}^m$. Let $f : X \times A \to \mathbb{R}$ be a continuous function, and let $G : X \to A$ be a* *continuous correspondence. Then the function $v : X \to \mathbb{R}$,*

$$v(x) = \max_{a \in G(x)} f(x,a),$$

is continuous, and the correspondence $\Gamma : X \to A$:

$$\Gamma(x) = \{a \in G(x) : f(x,a) = v(x)\},$$

is non-empty and u.h.c.

Proof. Let us first show that v is continuous. By the above lemma, it suffices to show that any sequence $x_n \to x$ has a subsequence $(x_{\varphi(n)})_{n \in N}$ such that $v(x_{\varphi(n)}) \xrightarrow{n} v(x)$. Take any $x \in X$ and consider any sequence $(x_n)_{n \in \mathbb{N}}$ such that $x_n \xrightarrow{n} x$. We need to construct a subsequence $(x_{\varphi(n)})_{n \in \mathbb{N}}$ such that $v(x_{\varphi(n)}) \xrightarrow{n} v(x)$.

As $x_n \xrightarrow{n} x$ we have for all n an element $a_n \in G(x_n)$ and $v(x_n) = f(x_n, a_n)$. As G is u.h.c. we have that there exists a subsequence $(a_{\varphi(n)})_{n \in N}$ converging to $a \in G(x)$ (apply u.h.c. twice). Also, as f is continuous, $\lim_n f(x_{\varphi(n)}, a_{\varphi(n)}) = \lim_n v(x_{\varphi(n)}) = f(x,a)$. Let us finish the proof by showing that $f(x,a) = v(x)$. If not, then there is an element $a' \in G(x)$ such that $v(x) = f(x,a') > f(x,a)$.

We have that $a' \in G(x)$, so by l.h.c. we have that there is a sequence $a'_{\varphi(n)} \xrightarrow{n} a'$ such that $a'_{\varphi(n)} \in G(x_{\varphi(n)})$. By optimality of $a_{\varphi(n)}$ we have that for all $n \in \mathbb{N}$:

$$f(x_\varphi(n), a_\varphi(n)) \geq f(x_\varphi(n), a'_{\varphi(n)}).$$

Taking limits on both sides and using continuity of f gives $f(x,a) \geq f(x,a')$, which gives the contradiction.

Next, let us show that $\Gamma(x)$ is u.h.c. Fix x and let $(x_n)_{n \in \mathbb{N}}$ be a sequence converging to x and let $a_n \in \Gamma(x_n)$. As $a_n \in G(x_n)$ for all n, we know that $(a_n)_{n \in \mathbb{N}}$ is bounded. Next if $a_n \xrightarrow{n} a$ then by continuity of v and f.

$$v(x) = \lim_n v(x_n) = \lim_n f(x_n, a_n) = f(x,a),$$

by continuity of f and v. As such, $a \in \Gamma(x)$ which shows that Γ is u.h.c. □

A correspondence $G : X \to A$ is convex valued if for all $x \in X$, the set $G(x) \subseteq A$ is a convex set. A function $f : X \times A \to \mathbb{R}$ is strictly quasi-concave in the second argument if for all $x \in X$ and all $a_1, a_2 \in A$ and all $\alpha \in [0,1]$:

$$f(x, \alpha a_1 + (1-\alpha) a_2) \geq \min\{f(x,a_1), f(x,a_2)\},$$

with a strict inequality if $\alpha \in (0,1)$ and $a_1 \neq a_2$.

In general the optimal value correspondence is not l.h.c. Consider the example where $X = \mathbb{R}$, $f(x,a) = xa^2$ and $G(x) = [-1,1]$ for all x. Then

$$\Gamma(x) = \begin{cases} \{-1,1\} \text{ if } x > 0 \\ [-1,1] \text{ if } x = 0, \\ 0 \text{ if } x < 0. \end{cases}$$

Take a sequence $x_t \to 0$ where $x_t < 0$ for all t. Then $0.5 \in \Gamma(0)$ but $\Gamma(x_t) = \{-1,1\}$ for all x_t so there is not sequence in $\Gamma(x_t)$ that converges to 0.5 which means that the correspondence Γ is not l.h.c. at $x = 0$.

Theorem 5. *Let $X \subseteq \mathbb{R}^l$ and $A \subseteq \mathbb{R}^k$. Let $G : X \to A$ be convex valued, compact valued and continuous. Assume that $f : X \times A \to \mathbb{R}$ is continuous and that $f(x,a)$ is strictly (quasi)-concave in its second argument then if*

$$v(x) = \max_{a \in G(x)} f(x,a),$$

we have that

$$\Gamma(x) = \{y \in G(x) : f(x,y) = v(x).\}$$

is single valued, and the function $\gamma(x)$ where $\Gamma(x) = \{\gamma(x)\}$ is continuous.

Proof. From the theorem of the maximum, we know that the optimal solution correspondence Γ is compact valued and u.h.c. Let us first show that Γ is single valued. Assume, towards a contradiction, that $a_1, a_2 \in \Gamma(x)$, i.e. $f(x,a_1) = f(x,a_2) = v(x)$. Then, by convex valuedness of G, for $\alpha \in (0,1)$, $\alpha a_1 + (1-\alpha)a_2 \in G(x)$. So by strict quasi-concavity:

$$f(x, \alpha a_1 + (1-\alpha)a_2) > \min\{f(x,a_1), f(x,a_2)\} = v(x).$$

This contradicts the optimality of a_1 and a_2. This shows that Γ is a single valued function. To show that γ is continuous, let $x_n \to x$. It suffices to show that $(x_n)_{n \in \mathbb{N}}$ has a subsequence $(x_{\varphi(n)})_{n \in \mathbb{N}}$ such that $\gamma(x_{\varphi(n)}) \overset{n}{\to} \gamma(x)$. Obviously, $\gamma(x_n) \in \Gamma(x_n)$. Then by u.h.c. of Γ, $(g(x_n))_{n \in \mathbb{N}}$ is bounded. So it has a convergent subsequence, say $\gamma(x_{\varphi(n)}) \overset{n}{\to} y$. Also $\gamma(x_{\varphi(n)}) \in \Gamma(x_{\varphi(n)})$ so again by u.h.c., $y \in \Gamma(x)$. By single valuedness of Γ, we have $y = \gamma(x)$. □

The second part of the theorem is actually true in general. If Γ is single valued and u.h.c. then the function γ where $\Gamma(x) = \{\gamma(x)\}$ is continuous. Also if Γ is single valued and l.h.c. then γ it is also continuous.

Consider a sequence of optimisation problems that converge to a certain optimisation problem in a sense that we will define in the next theorem. Under which conditions will the limit of the solutions (if it exists) equals the limiting solution?

Theorem 6. *Let $X \subseteq \mathbb{R}^l$ and $A \subseteq \mathbb{R}^k$. Consider a convex values, and continuous correspondence $G : X \to A$. Let $(f_n)_{n \in \mathbb{N}}$ be a sequence of continuous, real valued functions on $X \times A$ and assume that for all n, $f_n(x,a)$ is strictly concave in its second argument. Assume that f has the same properties and that for all $a \in A$:*

$$\sup_{x \in X, a \in G(x)} \frac{|f_n(x,a) - f(x,a)|}{\phi(x)} \overset{n}{\to} 0.$$

Let:

$$\gamma_n(x) = \arg\max_{a \in G(x)} f_n(x,a),$$

and

$$\gamma(x) = \arg\max_{a \in G(x)} f(x,a).$$

then for all $x \in X$, $\|\gamma_n(x) - \gamma(x)\| \xrightarrow{n} 0$.[34] *If* X *is compact then:*[35]

$$\|\gamma_n - \gamma\|_\phi \equiv \sup_{x \in X} \frac{\|\gamma_n(x) - \gamma(x)\|}{\phi(x)} \xrightarrow{n} 0.$$

[34] Here $\|.\|$ is the Euclidean norm on $\mathbb{R}^m$. This states that γ_n converges pointwise to γ.

[35] This is the notion of uniform convergence.

Proof. We have that for all $x \in X$:

$$\begin{aligned}
0 &\leq f(x,\gamma(x)) - f(x,\gamma_n(x)), \\
&\leq (f(x,\gamma(x)) - f(x,\gamma_n(x))) + \underbrace{(f_n(x,\gamma_n(x)) - f_n(x,\gamma(x)))}_{\geq 0}, \\
&= (f(x,\gamma(x)) - f_n(x,\gamma(x)) + (f_n(x,\gamma_n(x)) - f(x,\gamma_n(x))), \\
&\leq \phi(x) \sup_{y \in X, a \in G(y)} \frac{|f(y,a) - f_n(y,a)|}{\phi(y)} + \phi(x) \sup_{y \in X, a \in G(y)} \frac{|f_n(y,a) - f(y,a)|}{\phi(y)}.
\end{aligned}$$

This shows that:

$$\sup_{x \in X} \left| \frac{f(x,\gamma(x)) - f(x,\gamma_n(x))}{\phi(x)} \right| \xrightarrow{n} 0.$$

First, to show that for all $x \in X$, $\|\gamma_n(x) - \gamma(x)\| \xrightarrow{n} 0$, assume, towards a contradiction that for some $x \in X$, $\|\gamma_n(x) - \gamma(x)\| \stackrel{n}{\not\to} 0$. If so, there must be a subsequence $(\gamma_{\varphi(n)}(x))_{n \in \mathbb{N}}$ and a $\varepsilon > 0$ such that for all n:

$$\|\gamma_{\varphi(n)}(x) - \gamma(x)\| \geq \varepsilon.$$

Let,

$$A_\varepsilon = \{a \in G(x) : \|a - \gamma(x)\| \geq \varepsilon\}.$$

We know that for all n, $\gamma_{\varphi(n)}(x) \in A_\varepsilon$, so A_ε is non-empty. Also the element $\gamma(x) \notin A_\varepsilon$. The set A_ε is also a compact subset of A.

Let:

$$\delta = \min_{a \in A_\varepsilon} f(x,\gamma(x)) - f(x,a).$$

This problem is well defined as A_ε is a compact set and the objective function is continuous. Also as $f(x,\gamma(x)) \geq f(x,a)$ for all $a \in A_\varepsilon \subseteq G(x)$, we have that $\delta \geq 0$. In fact, the value $f(x,\gamma(x)) - f(x,a)$ is equal to 0 only if a solves the maximization problem which means that in this case $\gamma(x) = a$. By uniqueness of the solution and the fact that $\gamma(x) \notin A_\varepsilon$, we therefore conclude that the solution to this minimization problem satisfies $\delta > 0$. As $\gamma_{\varphi(n)}(x) \in A_\varepsilon$ it follows that for all n,

$$f(x,\gamma(x)) - f(x,\gamma_{\varphi(n)}(x)) \geq \delta > 0.$$

However, this contradicts, $\sup_{x \in X} \left| \frac{f(x,\gamma(x)) - f(x,\gamma_n(x))}{\phi(x)} \right| \xrightarrow{n} 0$.

For the second part, assume that X is compact. Let us show that $\|\gamma_n - \gamma\|_\phi \xrightarrow{n} 0$. If not then there exists a subsequence $(\gamma_{\varphi(n)})_{n \in \mathbb{N}}$ such that for all n,

$$\|\gamma_{\varphi(n)} - \gamma\|_\phi \geq \varepsilon.$$

In particular, by the definition of the ϕ-norm, this means that for all n there exists an $x_n \in X$ such that

$$\frac{\|\gamma_{\varphi(n)}(x_n) - \gamma(x_n)\|}{\phi(x_n)} \geq \varepsilon.$$

Let,

$$B_\varepsilon = \left\{ (x,a) \in X \times A : a \in G(x) \text{ and } \frac{\|a - \gamma(x)\|}{\phi(x)} \geq \varepsilon \right\}.$$

One sees that B_ε is a compact subset of $X \times A$. Also, for all n, there is an $x_n \in X$ and an $\gamma_{\varphi(n)}(x_n) \in G(x_n)$, such that $(x_n, \gamma_{\varphi(n)}(x_n)) \in B_\varepsilon$. This shows that B_ε is non-empty. Finally, for all $x \in X$, $(x, \gamma(x)) \notin B_\varepsilon$.

Let:

$$\delta = \min_{(x,a)\in B_\varepsilon} \frac{f(x,\gamma(x)) - f(x,a)}{\phi(x)}.$$

Due to the non-emptyness and compactness of B_ε this minimization problem is well defined. Observe that the objective function is non-negative and zero only if $f(x,\gamma(x)) = f(x,a)$ which can only happen (due to uniqueness of the optimum) if $a = \gamma(x)$. However, we know that for all $x \in X$: $(x,\gamma(x)) \notin B_\varepsilon$. This implies that $\delta > 0$. As for all n, $(x_n, \gamma_{\varphi(n)}(x_n)) \in B_\varepsilon$, we therefore have that for all n:

$$\frac{f(x_n,\gamma(x_n)) - f(x_n, \gamma_{\varphi(n)})(x_n)}{\phi(x_n)} \geq \delta > 0.$$

Then:

$$\delta \leq \frac{f(x_n,\gamma(x_n)) - f(x_n, \gamma_{\varphi(n)}(x_n))}{\phi(x_n)} \leq \sup_{x\in X} \left| \frac{f(x,\gamma(x)) - f(x,\gamma_{\varphi(n)}(x))}{\phi(x)} \right| \xrightarrow{n} 0,$$

a contradiction. □

Dynamic programming under certainty

In this chapter we will investigate the infinite horizon optimization problem that we presented in chapter 1.

We denote by $X \subseteq \mathbb{R}^l$ the state space. An element $x \in X$ captures the state of the world at a particular point in time. We denote by $A \subseteq \mathbb{R}^k$ the set of controls variables and we denote by $G : X \to A$ the correspondence that determines for all states x the possible values of the control variable. The next period feasible states are determined by a function $r : X \times A \to X$ such that $y = r(x, a) \in X$ is the next period's state if the current state is x and the chosen control variable has the value a. We also denote by $F : X \times A \to \mathbb{R}$ the instantaneous payoff function that depends on the values of the current state and control. Finally, let $\beta \in (0, 1)$ be a discount factor. In this section, we will be interested in finding solutions to the following infinite horizon optimization problem:

$$\begin{aligned} v(x_0) = \max_{a_0, a_1, a_2, \ldots} & \sum_{t=0}^{\infty} \beta^t F(x_t, a_t), \\ \text{s.t. } & x_{t+1} = r(x_t, a_t), \\ & a_t \in \Gamma(x_t), \\ & x_0 \text{ given.} \end{aligned}$$

We will do this by relating it to the fixed point of the so called Bellman operator, $T : C_\phi(X) \to C_\phi(X)$ defined by:

$$(Tv)(x) = \max_{a \in G(x)} \{F(x, a) + \beta v(r(x, a))\}.$$

In particular, we will show that under certain conditions the first problem has a solution, and its solution is equivalent to the fixed point of the Bellman operator Tv.

Definition 11 (Regularity condition)**.** *The problem* $(X, A, \Gamma, F, r, \beta)$ *is* ***regular*** *if the one period return function* $F : X \times A \to \mathbb{R}$ *and the transition function* $r : X \times A \to \mathbb{R}$ *are continuous, if the transition correspondence* $\Gamma : X \to A$ *is non-empty, continuous (u.h.c. and l.h.c.) and, additionally, there is a continuous function* $\phi : X \to \mathbb{R}_{++}$ *such that,*

1. *There exists an $M \geq 0$ such that for all $x \in X$:*

$$\max_{a \in \Gamma(x)} |F(x,a)| \leq M\phi(x).$$

2. *There exists a $\theta \in (0,1)$ such that for all $x \in X$:*

$$\beta \max_{a \in \Gamma(x)} \phi(r(x,a)) \leq \theta\phi(x).$$

Observe that if the function F is bounded, then the regularity conditions are satisfied by choosing $\phi(x) = 1$ for all x.

Theorem 7. *If the problem (X, A, Γ, F, β) is regular then the Bellman operator is a contraction mapping from $C_\phi(X)$ to $C_\phi(X)$.*

Proof. Let T be the Bellman operator. Let us first show that T maps from $C_\phi(X)$ to $C_\phi(X)$. If $v \in C_\phi(X)$, then v is by definition continuous. Also F and r are continuous by assumption and G is continuous and compact valued. As such, the optimization problem:

$$(Tv)(x) = \max_{a \in G(x)} \{F(x,a) + \beta v(r(x,a))\},$$

is well defined for all $x \in X$. By the theorem of the maximum, the maximum value function is continuous in x. As such, $(Tv)(x)$ is a continuous function of x.

In order to show that $T : C_\phi(X) \to C_\phi(X)$ it suffices to show that $\|Tv\|_\phi$ is bounded whenever $\|v\|_\phi$ is bounded. Assume that $\|v\|_\phi < N$ then:

$$\begin{aligned}
|(Tv)(x)| &= \left| \max_{a \in G(x)} \{F(x,a) + \beta v(r(x,a))\} \right|, \\
&\leq \max_{a \in G(x)} |F(x,a)| + \beta \max_{a \in G(x)} |v(r(x,a))|, \\
&\leq M\phi(x) + \beta N \max_{a \in G(x)} \phi(r(x,a)), \\
&\leq M\phi(x) + N\theta\phi(x) = (M + N\theta)\phi(x).
\end{aligned}$$

This shows that $\|Tv\|_\phi \leq M + N\theta$, so $\|Tv\|_\phi$ is bounded. Conclude that $Tv \in C_\phi(X)$.

In order to show that T is a contraction mapping, we use Blackwell's theorem. For monotonicity, assume that $v \geq w$. Then:

$$\begin{aligned}
(Tv)(x) &= \max_{a \in G(x)} \{F(x,a) + \beta v(r(x,a))\}, \\
&\geq \max_{a \in \Gamma(x)} \{F(x,a) + \beta w(r(x,a))\} = (Tw)(x).
\end{aligned}$$

For additivity, let $a \geq 0$. Then:

$$\begin{aligned}
(T(v + a\phi))(x) &= \max_{a \in G(x)} \{F(x,a) + \beta\,(v + a\phi)\,(r(x,a))\}, \\
&\leq \max_{a \in G(x)} \{F(x,a) + \beta v(r(x,a))\} + a \max_{a \in G(x)} \beta\phi(r(x,a)), \\
&\leq (Tv)(x) + a\theta\phi(x).
\end{aligned}$$

□

Above theorem shows that for a regular problem, the Bellman operator has a fixed point, say $v \in C_\phi(X)$. Then associated with the Bellman operator, T we can find a policy correspondence, Γ where

$$\Gamma(x) = \{a \in G(x) : F(x,a) + \beta v(r(x,a)) = v(x)\}.$$

The correspondence Γ gives the optimal action for a given state x.

Consider our infinite horizon optimization problem.

$$\begin{aligned} \max_{(a_t)_{t=0}^\infty} & \sum_{t=0}^{\infty} \beta^t F(x_t, a_t), \\ \text{s.t. } & a_t \in G(x_t), \\ & x_{t+1} = r(x_t, a_t), \\ & x_0 \in X \text{ given.} \end{aligned}$$

We need to resolve two issues. First, we need to show that this problem is well defined. Second, we would like to show that the solution to this problem coincides with the fixed point of the Bellman operator.

Definition 12 (feasible path)**.** *A **feasible path** is a sequence $((x_0, a_0), (x_1, a_1), \ldots)$ such that for all $t \in \mathbb{N}$,*

1. $a_{t-1} \in G(x_{t-1})$,
2. $x_t = r(x_{t-1}, a_{t-1})$.

Let $\Pi(x_0)$ be the set of all feasible paths that start at the state $x_0 \in X$. Then for such paths $p = ((x_0, a_0), (x_1, a_1), \ldots) \in \Pi(x_0)$ we define:

$$w_p = \sum_{t=0}^{\infty} \beta^t F(x_t, a_t) \equiv \lim_{T\to\infty} \sum_{t=0}^{T} \beta^t F(x_t, a_t),$$

whenever this limit is well defined.

Lemma 4. *Assume that (X, Γ, F, r, β) is regular and let $x_0 \in X$. Then for all paths $p \in \Pi(x_0)$, w_p is well defined.*

Proof. Fix a path $p = \{(x_0, a_0), (x_1, a_1), \ldots, \} \in \Pi(x_0)$ and the partial sums:

$$u_n = \sum_{t=0}^{n} \beta^t F(x_t, a_t).$$

Notice that we defined $w_p = \lim_n u_n$. As such, in order to show that w_p is well defined, we need to show that the sequence $u_1, u_2, u_3, \ldots$ converges. As $(u_n)_{n\in\mathbb{N}}$ is a sequence in $\mathbb{R}$, we can do this by showing that $(u_n)_{n\in\mathbb{N}}$ is a Cauchy sequence.[36] For any $n, m \in \mathbb{N}$ (without loss

[36] Remember that $\mathbb{R}$ is a Banach space, so every Cauchy sequence of real numbers converges.

of generality, we take $n > m$):

$$\begin{aligned} |u_n - u_m| &= \left| \sum_{t=m+1}^{n} \beta^t F(x_t, a_t) \right|, \\ &\leq \sum_{t=m+1}^{n} \beta^t |F(x_t, a_t)|, \\ &\leq \sum_{t=m+1}^{n} M\beta^t \phi(x_t). \end{aligned}$$

Now,

$$\phi(x_t) \leq \max_{a \in G(x_{t-1})} \phi(r(x_{t-1}, a)) \leq \frac{\theta}{\beta}\phi(x_{t-1}).$$

Iterating this further, we see that: $\phi(x_t) \leq \left(\frac{\theta}{\beta}\right)^t \phi(x_0)$. As such,

$$\begin{aligned} |u_n - u_m| &\leq \sum_{t=m+1}^{n} M\theta^t \phi(x_0), \\ &\leq M\theta^{m+1}\phi(x_0) \sum_{t=0}^{n-m-1} \theta^t, \\ &\leq M\theta^{m+1} \frac{\phi(x_0)}{1-\theta} \xrightarrow{m} 0 \end{aligned}$$

This shows that $(u_n)_{n \in \mathbb{N}}$ is Cauchy. So $\lim_{n\to\infty} u_n = w_p$ is well defined. □

Above lemma allows us to rewrite the infinite horizon optimization problem in the following way:

$$\max_{p \in \Pi(x_0)} w_p.$$

In order to show that this problem is well defined, we will show that that the fixed point of the Bellman operator v (which we know exists) satisfies that for all $x_0 \in X$ and all $p \in \Pi(x_0)$, (i) $v(x_0) \geq w_p$ and (ii) that for all $x_0 \in X$, there exists a path $p \in \Pi(x_0)$ such that $v(x_0) = w_p$.

Lemma 5. *Let (X, A, G, F, β) be a regular problem and let v be the fixed point of the Bellman operator. Then for all paths $p \in \Pi(x_0)$ $v(x_0) \geq w_p$.*[37]

[37] In other words, $v(x_0)$ is an upper bound for $\{w_p : p \in \Pi(x_0)\}$.

Proof. Let v be the fixed point of the Bellman operator and let $p = \{((x_0, a_0), (x_1, a_1), \ldots\} \in \Pi(x_0)$ be a path. We will show that $v(x_0) \geq w_p$.

Now, by definition $v(x) = \max_{a \in G(x)} F(x, a) + \beta v(r(x, a))$, so:

$$\begin{aligned} v(x_0) &\geq F(x_0, a_0) + \beta v(x_1), \\ &\geq F(x_0, a_0) + \beta F(x_1, a_1) + \beta^2 v(x_2), \\ &\ldots \\ &\geq \sum_{t=0}^{T} \beta^t F(x_t, a_t) + \beta^{T+1} v(x_{T+1}). \end{aligned}$$

Taking the limit for $T \to \infty$, gives[38],

$$v(x_0) \geq \lim_{T\to\infty} \sum_{t=0}^{T} \beta^t F(x_t, a_t) + \lim_{T\to\infty} \beta^{T+1} v(x_{T+1}) = w_p + \lim_{T\to\infty} \beta^{T+1} v(x_{T+1}).$$

[38] Observe that the limit $\lim_T \sum_{t=0}^{T} \beta^t F(x_t, a_t) = w_p$ is well defined by the previous lemma.

As such, it suffices to show that $|\beta^T v(x_T)| \xrightarrow{T} = 0$. Now:

$$|v(x_T)| = \frac{|v(x_T)|}{\phi(x_T)} \frac{\phi(x_T)}{\phi(x_{T-1})} \cdots \frac{\phi(x_1)}{\phi(x_0)} \phi(x_0).$$

Also, v is bounded in the ϕ-norm, so:

$$\frac{|v(x_T)|}{\phi(x_T)} \leq \|v\|_\phi.$$

Next, for all t:

$$\begin{aligned}
&\phi(x_t) \leq \max_{a \in G(x_{t-1})} \phi(r(a, x_{t-1}) \leq \frac{\theta}{\beta} \phi(x_{t-1}), \\
\to &\frac{\phi(x_t)}{\phi(x_{t-1})} \leq \frac{\theta}{\beta}.
\end{aligned}$$

From this, it follows that:[39]

$$\begin{aligned}
&|v(x_T)| \leq \left(\frac{\theta}{\beta}\right)^T \|v\|_\phi \phi(x_0), \\
\to &|\beta^T v(x_T)| \leq \theta^T \|v\|_\phi \phi(x_0) \xrightarrow{T} 0.
\end{aligned}$$

[39] Notice that x_0 is kept fixed here.

This shows that the fixed point of the Bellman provides an upper bound to any feasible solution of the infinite horizon optimization problem: $v(x_0) \geq w_p$. □

Now, we are going to show that there actually exists a path $p \in \Pi(x_0)$ such that $v(x_0) = w_p$. Given the fixed point v of the Bellman operator, define recursively a path $p^* = ((x_0, a_0), (x_1, a_1), \ldots) \in \Pi(x_0)$ by:[40]

$$\begin{aligned}
&a_t \in \arg\max_{a \in G(x_t)} \{F(x_t, a) + \beta v(r(x_t, a))\}, \\
&x_{t+1} = r(x_t, a_t).
\end{aligned}$$

[40] The right hand side is a maximization problem of a continuous function $(F(x_t, a) + \beta v(r(x_t, a))$ over a compact set $G(x_t)$ so the solution a_t and therefore the path p^* is well defined.

Lemma 6. *Let (X, A, Γ, F, β) be a regular problem and let v be the fixed point of the Bellman operator. Then for all $x_0 \in X$, $v(x_0) = w_{p^*}$.*

Proof. Let $p^* = ((x_0, a_0), (x_1, a_1), \ldots)$ be the path as defined above. Then, we have,

$$\begin{aligned}
v(x_0) &= F(x_0, a_0) + \beta v(x_1), \\
&= F(x_0, a_0) + \beta F(x_1, a_1) + \beta v(x_2), \\
&\vdots \\
&= \sum_{t=0}^{T} \beta^t F(x_t, a_t) + \beta^{T+1} v(x_{T+1}).
\end{aligned}$$

Taking limits gives, $v(x_0) = w_{p^*} + \lim_{T\to\infty} \beta^{T+1} v(x_{T+1})$ and similar to the proof of the previous lemma, we have that:

$$\lim_{T\to\infty} \beta^T v(x_T) = 0.$$

So $v(x_0) = w_{p^*}$ as was to be shown. □

Let us consider the following AK-model:

$$\begin{aligned}
&\max_{c_0, c_1, \dots} \sum_{t=0}^{\infty} \beta^t u(c_t) \text{ s.t. } k_{t+1} = Ak_t - c_t,\\
&\quad \text{s.t. } 0 \leq c_t \leq Ak_t,\\
&\qquad k_0 \text{ given.}
\end{aligned}$$

Here, capital k is the state variable and consumption c is the control variable, $u : \mathbb{R}_+ \to \mathbb{R}_+$ is the instantaneous payoff function, which we assume to be concave and monotone. Next, $\beta \in (0,1)$ is the discount factor. In terms of the model outlined above, we have that $X = \mathbb{R}_+, A = \mathbb{R}$, $x = k$, $a = c$, $F(x,a) = u(c)$, $\Gamma(x) = \{c \in \mathbb{R}_+ : 0 \leq c \leq Ak\}$ and $r(k,c) = Ak - c$.

We normalize utility such that: $u(0) = 0$, so u maps to $\mathbb{R}_+$.

The Bellman operator is given by,

$$(Tv)(k) = \max_{0\leq c\leq Ak} \{u(c) + \beta v(Ak - c)\}$$

In order for the fixed point of the Bellman operator to solve the problem, we need to find a function $\phi(k) > 0$ a number M and a $\theta < 1$ such that,

- $\max_{0\leq c\leq Ak} u(c) \leq M\phi(k)$.
- $\beta \max_{0\leq c\leq Ak} \phi(Ak - c) \leq \theta\phi(k)$.

Let us first consider the case where $A \leq 1$. Let's try the function

$$\phi(k) = \max\{1, u(k)\} > 0.$$

which is strictly positive and continuous. For the first condition, we have:

$$\max_{0\leq c\leq Ak} |u(c)| = |u(Ak)| \leq u(k) = \phi(k).$$

The first inequality uses $Ak \leq k$ (i.e. $A \leq 1$) and the fact that $u(.)$ is increasing. For the second property:

$$\begin{aligned}
\beta \max_{0\leq c\leq Ak} \max\{1, u(Ak - c)\} &= \beta \max\{1, u(Ak)\},\\
&\leq \beta \max\{1, u(k)\} = \beta\phi(k).
\end{aligned}$$

So we see that all conditions are satisfied and the fixed point of the Bellman operator will coincide with the value function of the in finite horizon optimisation problem.

Now, consider the more interesting case where $A > 1$. First, notice that $1/A \in (0,1)$ and by concavity of u, we therefore have that:

$$\left(1 - \frac{1}{A}\right) u(0) + \frac{1}{A} u(Ak) \leq u(k).$$

As $u(0) = 0$, this gives:

$$u(Ak) \leq Au(k).$$

Then for the first property:

$$\max_{0 \leq c \leq Ak} |u(c)| = |u(Ak)| \leq Au(k) \leq A \max\{1, u(k)\} = A\phi(k).$$

So if we set $M = A$, the condition is satisfied.

For the second condition:

$$\begin{aligned} \beta \left(\max_{0 \leq c \leq Ak} \max\{1, u(Ak - c)\} \right) &= \beta \max\{1, u(Ak)\}, \\ &\leq \beta A \max\{1, u(Ak)\} = \beta A \phi(k). \end{aligned}$$

This shows the second condition is satified if $\beta A < 1$. Observe that this requires that $A < 1/\beta$ so A can be greater than one but not too high for this to work.

If the utility function is sufficiently concave, we can improve upon this restriction however. Assume for instance that $u(c) = c^\alpha$ with $\alpha \in (0,1)$. Then:

$$\begin{aligned} \beta \left(\max_{0 \leq c \leq Ak} \max\{1, u(Ak - c)\} \right) &= \beta \max\{1, u(Ak)\}, \\ &= \beta \max\{1, (Ak)^\alpha\} \quad \leq \beta A^\alpha \max\{1, k^\alpha\} = \beta A^\alpha \phi(k). \end{aligned}$$

Now the condition is satisfied as long as $\beta A^\alpha < 1$.

Properties of the Bellman fixed point

So far we obtained sufficient conditions for the Bellman operator to be a contraction mapping and for the fixed point of the Bellman equation to be equal to the optimal value function of the infinite horizon model.

We can use this feature to derive some of the properties of this fixed point (under some conditions). For this part, we will take the simplifying assumption that $A = X$ and $r(x, a) = a$, so the action taken coincides with the state in the next period. Then, the optimization problem can be rewritten as:

$$v(x) = \max_{a \in \Gamma(x)} \{F(x,a) + \beta v(a)\}.$$

Theorem 8. *If the problem (X, Γ, F, β) is regular, if $F(x,a)$ is strictly increasing in each of its first arguments and if Γ is monotone in the sense that for all $x, x' \in X$ with $x \leq x'$,*

$$\Gamma(x) \subseteq \Gamma(x').$$

Then, the fixed point v of the Bellman operator is strictly increasing.

Proof. Let $C'_\phi(X)$ be the subset of continuous, weakly increasing functions on X that are bounded in the ϕ-norm. Let $C''_\phi(X) \subset C'_\phi(X)$ be the subset of strictly increasing continuous functions on X that are bounded in the ϕ-norm. Since $C'_\phi(X)$ is a closed subset of $C_\phi(X)$ it suffices to show that the Bellman operator T maps from $C'_\phi(X)$ to $C''_\phi(X)$, i.e. $T[C'_\phi(X)] \subseteq C''_\phi(X)$.[41]

If $x' > x$ then by monotonicity of Γ: $\Gamma(x) \subseteq \Gamma(x')$. Let a^* solve $\max_{a \in \Gamma(x)}\{F(x,a) + \beta v(a)\}$. Notice that $a^* \in \Gamma(x')$. Then,

$$\begin{aligned}(Tv)(x) = F(x,a^*) + \beta v(a^*) &< F(x',a^*) + \beta v(a^*), \\ &\leq \max_{a \in \Gamma(x')} \{F(x',a) + \beta v(a)\} = (Tv)(x').\end{aligned}$$

□

[41] In other words, the Bellman operator maps weakly increasing functions into the set of strictly increasing functions.

Theorem 9. *If the problem (X, Γ, F, β) is regular, if F is strictly concave, i.e. for all $\alpha \in (0,1)$ and all $x, x' \in X$ and $a, a' \in A$ with $(x,a) \neq (x',a')$:*

$$F(\alpha x + (1-\alpha)x', \alpha a + (1-\alpha)a') > \alpha F(x,a) + (1-\alpha)F(x',a'),$$

and if Γ is convex in the sense that for all $\alpha \in [0,1]$ and $x, x' \in X$:

$$a \in \Gamma(x), a' \in \Gamma(x') \textit{ implies } \alpha a + (1-\alpha)a' \in \Gamma(\alpha x + (1-\alpha)x').$$

Then v is strictly concave and the policy function $\gamma(x) = \arg\max_{a \in \Gamma(x)}\{F(x,a) + \beta v(a)\}$ is a continuous, single-valued function.

Proof. Let $C'_\phi(X)$ be the subset of continuous, weakly concave functions bounded in the ϕ-norm and let $C''_\phi(X)$ be the subset of continuous, strictly concave functions that are bounded in the ϕ-norm. It suffices to show that the Bellman operator maps $C'_\phi(X)$ into $C''_\phi(X)$, i.e. $T[C'_\phi(X)] \subseteq C''_\phi(X)$.

Let v be concave and let $x_0 \neq x_1$, $\alpha \in (0,1)$ and set $x_\alpha = \alpha x_0 + (1-\alpha)x_1$. Also let a_0 solve $\max_{a \in \Gamma(x_0)}\{F(x_0,a) + \beta v(a)\}$ and a_1 solve

$\max_{a\in\Gamma(x_1)}\{F(x_1,a)+\beta v(a)\}$. Let $a_\alpha = \alpha a_0 + (1-\alpha)a_1$. Then, by assumption $a_\alpha \in \Gamma(x_\alpha)$:

$$\begin{aligned}(Tv)(x_\alpha) &\geq F(x_\alpha, a_\alpha) + \beta v(a_\alpha),\\ &> \alpha F(x_0,a_0) + (1-\alpha)F(x_1,a_1) + \beta\alpha v(a_0) + \alpha(1-\theta)v(a_1),\\ &= \alpha(Tv)(x_0) + (1-\alpha)(Tv)(x_1),\end{aligned}$$

This shows that the Bellman fixed point function, say v, is strictly concave. Given strict concavity we have that:

$$\max_{a\in\Gamma(x)}\{F(x,a)+\beta v(a)\},$$

maximizes a strictly concave function and that Γ is convex valued. As such, the optimal value is unique and it corresponds to a continuous function. □

Theorem 10. *Let (X,Γ,F,β) be regular and let $F(x,a)$ be strictly concave in a and let Γ be convex. Let $C'_\phi(X)$ be the set of concave, continuous functions that are bounded in the ϕ-norm. and let $v_0 \in C'_\phi(X)$. Let $(v_n, g_n)_{n\in\mathbb{N}}$ be defined as,*

$$\begin{aligned}v_{n+1} &= Tv_n,\\ \gamma_n(x) &= \arg\max_{a\in\Gamma(x)}\{F(x,a)+\beta v_n(a)\}.\end{aligned}$$

Then for all $x \in X$: $\|\gamma_n(x)-\gamma(x)\| \xrightarrow{n} 0$ pointwise where $\gamma(x)$ is the optimal policy function. If X is compact, then $\|\gamma_n-\gamma\|_\phi \to 0$.

Proof. Let $C''_\phi(X)$ be the set of strictly concave, continuous functions that are bounded in the ϕ-norm. We know that for all n, $v_n \in C''_\phi(X)$. For $a \in \Gamma(x)$, let $f_n(x,a) = F(x,a)+\beta v_n(a)$. We have that every function $f_n(x,y)$ is strictly concave. Also let $f(x,a) = F(x,a)+\beta v(a)$ where v is the fixed point of the Bellman operator. Then,

$$\begin{aligned}|f_n(x,a)-f(x,a)| &= \beta|v_n(a)-v(a)|,\\ &= \phi(a)\beta\frac{|v_n(a)-v(a)|}{\phi(a)},\\ &\leq \phi(a)\beta\|v_n-v\|_\phi,\\ &\leq \theta\phi(x)\|v_n-v\|_\phi.\end{aligned}$$

By the contraction mapping theorem, we have that $\|v_n-v\|_\phi \xrightarrow{n} 0$. This shows that:

$$\sup_{x\in X, a\in\Gamma(x)}\frac{|f_n(x,a)-f(x,a)|}{\phi(x)} \xrightarrow{n} 0.$$

As such, we have that for all $x \in X$, $\|\gamma_n(x)-\gamma(x)\| \xrightarrow{n} 0$. If X is compact, we get that $\|\gamma_n-\gamma\|_\phi \xrightarrow{n} 0$. □

The following part provides assumptions for which the value function can be assumed to be differentiable. It uses the Benveniste-Scheinkman theorem.

Theorem 11 (Benveniste and Scheinkman)**.** *Let $X \subseteq \mathbb{R}^l$ be a convex set, let $V : X \to \mathbb{R}$ be concave and continuous, let x_0 be in the interior of X and let D be a convex open neighbourhood of x_0. If there is a concave, differentiable function $W : D \to \mathbb{R}$ with $W(x_0) = V(x_0)$ and $W(x) \leq V(x)$ for all $x \in D$ then V is differentiable at x_0 and,*

$$\nabla_x V(x_0) = \nabla_x W(x_0).$$

Proof. Let V be concave. Any supgradient δ of $V(x_0)$ must satisfy for all $x \in D$,

$$W(x) - W(x_0) \leq V(x) - V(x_0) \leq \delta(x - x_0).$$

This shows that δ is also a supgradient of W at x_0. Since W is differentiable, its supgradient is unique and $\delta = \nabla_x W(x_0)$. This means that V also has a unique subgradient and:

$$\nabla_x W(x_0) = \delta = \nabla_x V(x_0).$$

□

Theorem 12. *Let (X, Γ, F, β) be a regular problem and assume that F is strictly concave and Γ is convex. Assume that F is C^1. Let v be the fixed point of the Bellman operator and let γ be the continuous optimal value function. If x_0 is in the interior of X and $\gamma(x_0)$ is in the interior of $\Gamma(x_0)$, then v is C^1 at x_0 and*

$$\nabla_x v(x_0) = F_1(x_0, \gamma(x_0)).$$

where $F_1(x_0, \gamma(x_0)) = \nabla_x F(x, a)|_{(x,a)=(x_0,\gamma(x_0))}$.

Proof. As F is strictly concave and G is convex, γ is a function. Also, since $\gamma(x_0)$ is in the interior of $\Gamma(x_0)$ and G is lower hemi-continuous and convex valued, $\gamma(x_0)$ is in the interior of $\Gamma(x)$ for all x in a convex open neighbourhood D of x_0.[42] Define W on D by

[42] The proof of this is a bit too elaborate to give here, so I will omit it.

$$W(x) = F(x, \gamma(x_0)) + \beta v(\gamma(x_0)).$$

The function W is concave and differentiable, with:

$$\nabla_x W(x_0) = F_1(x_0, \gamma(x_0)).$$

In addition

$$W(x_0) = F(x_0, \gamma(x_0)) + \beta v(\gamma(x_0)) = v(x_0).$$

Also, for all x in D:

$$W(x) = F(x, g(x_0)) + \beta v(g(x_0)),$$
$$\leq \max_{a \in \Gamma(x)} \{F(x, a) + \beta v(a)\} = v(x).$$

The inequality uses the fact that $\gamma(x_0) \in \Gamma(x)$ for all $x \in D$. The result then follows from Benveniste and Sheinkman theorem. □

Above theorem (also called the envelope theorem) is convenient to derive so-called Euler equations. Assume that the conditions in the theorem are satisfied and that $X = A \subseteq \mathbb{R}$. Let us add time index notation so the state at time t is denoted by x_t and the action taken at period t is denoted by x_{t+1}. Then the Bellman equation takes the form:

$$v(x_t) = \max_{x_{t+1} \in \mathbb{R}} \{F(x_t, x_{t+1}) + \beta v(x_{t+1})\}.$$

Here we omitted the condition $x_{t+1} \in \Gamma(x_t)$ as the solution is interior anyway. If the conditions in the theorem are satisfied and if $X = A \subseteq \mathbb{R}$. Then above lemma shows that:

$$v'(x_t) = F_1(x_t, x_{t+1}).$$

Next, by the first order condition in the previous period, we obtain:

$$F_2(x_t, x_{t-1}) + \beta v'(x_t) = 0.$$

Substitution gives:

$$F_2(x_t, x_{t-1}) + \beta F_1(x_t, x_{t+1}) = 0$$

This is called the Euler equation.

Euler equations

There is a second more classical way to derive the Euler equation. As before, consider the following infinite horizon optimization problem.

$$\max_{x_1, x_2, \ldots} \sum_{t=0}^{\infty} \beta^t F(x_t, x_{t+1}) \text{ s.t. } x_{t+1} \in \Gamma(x_t).$$

Assume that this problem can be solved and has a unique optimal solution $(x_n^*)_{n \in \mathbb{N}}$. Now, consider the following problem,

$$\max_{x_{t+1}} F(x_t^*, x_{t+1}) + \beta F(x_{t+1}, x_{t+2}^*) \text{ s.t. } x_{t+1} \in \Gamma(x_t^*), x_{t+2}^* = \Gamma(x_{t+1}).$$

Given the optimality of x_t^* and x_{t+2}^*, we need that x_{t+1}^* also solves this smaller problem. If F is differentiable, strictly concave and if

x^*_{t+1}, x^*_{t+2} are in the interior of $\Gamma(x^*_t)$ and $\Gamma(x^*_{t+1})$ (so the two constraints are not binding), then we obtain the following first order condition:

$$F_2(x^*_t, x^*_{t+1}) + \beta F_1(x^*_{t+1}, x^*_{t+2}) = 0.$$

This again obtains the Euler equation. If the solution is interior, then as we have seen before, this condition is also necessary for optimality. Usually the set of Euler equations is completed by adding a so called **transversality condition** namely,

$$\lim_{t\to\infty} \beta^t F_1(x^*_t, x^*_{t+1}) \cdot x^*_t \leq 0.$$

It can be shown that if the Euler equations are satisfied, F is concave, $F_1(x, y) > 0$, $x \geq 0$, and the transversality hold, then it must be that any solution $(x^*_0, x^*_1, \dots, x^*_n, \dots)$ that satisfies the Euler equation is indeed optimal. To see this, let $(x_0, x_1, x_2, \dots)$ be another feasible path. If F is concave, then, by concavity of F:

$$\begin{aligned}
\sum_{t=0}^{T} \beta^t \left(F(x_t, x_{t+1}) - F(x^*_t, x^*_{t+1})\right) &\leq \sum_{t=0}^{T} \beta^t F_1(x^*_t, x^*_{t+1}) \cdot (x_t - x^*_t) + \sum_{t=0}^{T} \beta^t F_2(x^*_t, x^*_{t+1}) \cdot (x_{t+1} - x^*_{t+1}), \\
&= F_1(x^*_0, x^*_1)(x_0 - x^*_0) + \sum_{k=0}^{T-1} \beta^{k+1} F_1(x_{k+1}, x_{k+2}) \cdot (x_{k+1} - x^*_{k+1}), \\
&+ \sum_{k=0}^{T} \beta^k F_2(x^*_k, x^*_{k+1}) \cdot (x_{k+1} - x^*_{k+1}),
\end{aligned}$$

The first term on the right hand side is zero as $x^*_0 = x_0$ is fixed. Then rearranging terms gives,

$$\begin{aligned}
\sum_{t=0}^{T} \beta^t (F(x_t, x_{t+1}) - F(x^*_t, x^*_{t+1})) &\leq \sum_{k=0}^{T-1} \beta^k \left[\beta F_1(x^*_{k+1}, x^*_{k+2}) + F_2(x^*_k, x^*_{k+1})\right] \cdot (x_{k+1} - x^*_{k+1}), \\
&+ \beta^T F_2(x^*_T, x^*_{T+1}) \cdot (x_{T+1} - x^*_{T+1}).
\end{aligned}$$

The summations in the first line is equal to zero by the Euler equation. Then:

$$\begin{aligned}
\sum_{t=0}^{T} \beta^t (F(x_t, x_{t+1}) - F(x^*_t, x^*_{t+1})) &\leq \beta^T F_2(x^*_T, x^*_{T+1}) \cdot (x_{T+1} - x^*_{T+1}), \\
&= \beta^T \left(-\beta F_1(x^*_{T+1}, x^*_{T+2})\right) \cdot (x_{T+1} - x^*_{T+1}) \\
&\leq \beta^{T+1} F_1(x^*_{T+1}, x^*_{T+2}) \cdot x^*_{T+1}.
\end{aligned}$$

where we used the Euler equation, the fact that $F_1(x, y) \geq 0$ and $x_t \geq 0$ for all T. Taking $T \to \infty$, we see that the solution $(x^*_t)_{t\in\mathbb{N}}$ is indeed optimal, whenever:

$$\lim_{T\to\infty} \beta^T F_1(x^*_T, x_{T+1}) x^*_T \leq 0,$$

which is the transversality condition.

As an example, consider the Bellman equation of the AK-model:

$$v(k_t) = \max_{k_{t+1} \leq Ak_t} u(Ak_t - k_{t+1}) + \beta v(k_{t+1}).$$

The first order condition gives:

$$-u'(Ak_t - k_{t+1}) + \beta v'(k_{t+1}) = 0.$$

Then from the envelope theorem we get,

$$v'(k_t) = Au'(Ak_t - k_{t+1})$$

Updating one period and substitution gives the Euler equation,

$$u'(Ak_t - k_{t+1}) = A\beta u'(Ak_{t+1} - k_{t+2}).$$

Which is a second order difference equation in terms of capital.
Equivalently, in terms of consumption:

$$u'(c_t) = A\beta u'(c_{t+1}).$$

The transversality condition requires that,

$$\lim_{t \to \infty} A\beta^t u'(Ak_t - k_{t+1})k_t \leq 0.$$

Numerical methods

In the previous chapter we say that the solution of the infinite horizon dynamic programming problem could be restated in terms of a solution of the Bellman equation. In many cases there is no closed form solutions to this Bellman equation[43] On the other hand, we have also seen that the unique solution of the Bellman equation can be found as the fixed point of a contraction mapping: the Bellman operator. This fixed point can be approximated very precisely by iterating over this operator a sufficient number of times. In this way, it is possible to approximate the fixed point via simulation methods. These methods are of course finite in nature and provide therefore only an approximation to the true fixed point.

[43] If there are solutions, they are mainly used for pedagogical purposes and are only available in a few special cases.

The simplest (but not necessarily most efficient) method is the value function iteration.

Value function iteration

The contraction mapping theorem tells us that the solution of the Bellman equation can be found by iterating the Bellman operator T

$$(Tv)(x) = \max_{a \in G(x)} \{F(x,a) + \beta v(r(x,a))\}.$$

As computers only work with finite entities, a first step is to approximates the state space X using a finite grid for the possible values of x,

$$X = \{x_1, x_2, \ldots, x_n\}.$$

Also the space of all possible actions A must be approximated using a finite grid:

$$A = \{a_1, \ldots, a_m\}.$$

The correspondence $G : X \to A$ is now replaced by a non-empty correspondence from the finite set X to the finite set A. Also, the instantaneous return function $F : X \times A \to \mathbb{R}$ is now a function from the finite set $X \times A$ to $\mathbb{R}$, so it is bounded by definition. Finally, the updating rule $r(x, a)$ must take values from $X \times A$ to the finite set X. It can be shown that when restricted to such finite setting, the Bellman operator T is still is a contraction mapping.[44] As such, finding, or approximating, the fixed point of T takes the following steps:

1. Decide on a (fine enough) grid for the state space X and control space A.[45]
2. Decide on some tolerance level $\varepsilon > 0$.[46]
3. Decide on an initial bounded function $v_0 : X \to \mathbb{R}$. Initiate the iteration round $n = 0$.
4. (a) Compute for all x in the finite grid X:

$$v_{n+1}(x) = \max_{a \in G(x)} \{F(x, a) + \beta v_n(r(x, a))\}.$$

   A naive way to do this could be to, for given x, compute for all $a \in G(x)$ the value $F(x, a) + \beta v_n(r(x, a)))$ and then take the maximum over all $a \in G(x)$. More sophisticated algorithms can exist if some properties (like concavity) of the objective function are known. Given the optimal values of a, we can also find the policy correspondence:

$$\Gamma_{n+1}(x) = \arg\max_{a \in G(x)} \{F(x, a) + \beta v_n(r(x, a))\}.$$

   (b) Iterate step (a) as long as $\|v_{n+1} - v_n\| = max_{x \in X}|v_{n+1}(x) - v_n(x)| > \varepsilon$, each time updating the counter $n \leftarrow n + 1$.
5. The final update gives a function v_n and policy correspondence Γ_n that should be a good approximation to the fixed point of the Bellman operator and the corresponding policy correspondence.

In order to get a better grasp of the algorithm, let us work out a particular example. Consider a representative consumer model with CRRA utility function,[47]

$$u(c) = \frac{c^{1-\sigma} - 1}{1 - \sigma}.$$

The consumer maximizes her infinite horizon discounted utility:

$$\sum_{t=0}^{\infty} \beta^t u(c_t).$$

There is a stock of capital k_t in period t that she can use to produce an amount of capital in the next period using a production function

[44] Observe that F is bounded, so the conditions of Definition 11 are satisfied with $\phi(x) = 1$ which means that the Bellman operator $T : B(X) \to B(X)$ where $B(X)$ is the set of bounded functions on X has a unique fixed point by Blackwell's theorem. Also, the corresponding policy correspondence:

$$\Gamma(x) = \arg\max_{a \in G(x)} F(x, a) + v(r(x, a)),$$

is non-empty.

[45] Often the problem can be reformulated such that the control space and state space coincide, i.e. $A = X$. In this case, the action is to decide on the state in the next period.

[46] This should be a sufficiently small number.

[47] CRRA stands for constant relative risk aversion. The relative risk aversion of the utility function $u(.)$ is given by,

$$-\frac{u''(c)}{u'(c)}c.$$

$f(k_t) = k_t^\alpha$. There is also a depreciation rate of δ. This gives the following law of motion:

$$k_{t+1} = k_t^\alpha - c_t + (1-\delta)k_t.$$

As such, we obtain the following dynamic program:

$$\begin{aligned} &\max_{c_0, c_1, \dots} \sum_{t=0}^{\infty} \beta^t u(c_t), \\ &\text{s.t. } k_{t+1} = k_t^\alpha - c_t + (1-\delta)k_t, \\ &\quad c_t \in [0, k_t^\alpha - (1-\delta)k_t], \\ &\quad k_0 \text{ given} \end{aligned}$$

To make our lives a bit easier, we can substitute out the choice variable c_t from this problem:

$$\begin{aligned} &\max_{k_1, k_2, \dots} \sum_{t=0}^{\infty} \beta^t u(k_t^\alpha + (1-\delta)k_t - k_{t+1}), \\ &\text{s.t. } k_{t+1} \in [0, k_t^\alpha + (1-\delta)k_t], \\ &\quad k_0 \text{ given.} \end{aligned}$$

The Bellman operator for this problem is given by:

$$(Tv)(k) = \max_{k' \in [0, k^\alpha + (1-\delta)k]} \left\{ u(k^\alpha + (1-\delta)k - k') + \beta v(k') \right\}.$$

In particular, using the CRRA utilty function, we obtain:

$$(Tv)(k) = \max_{k' \in [0, k^\alpha + (1-\delta)k]} \left\{ \frac{(k^\alpha + (1-\delta)k - k')^{1-\sigma} - 1}{1-\sigma} + \beta v(k') \right\}.$$

Our aim is to write a program that computes the fixed point of T by, starting at some function v_0, sequentially computing $v_1 = Tv_0, v_2 = Tv_1, \dots$

1. First of all, we need to initialize some parameters. Let's pick the values:

   $$\sigma = 1.5, \qquad \delta = 0.1, \qquad \beta = 0.95, \qquad \alpha = 0.3.$$

   We also need to set the threshold for convergence which is a small number, say $\varepsilon = 10^{-3}$.

2. Next, we need to decide on a grid size, N, say $N = 1000$.

3. The grid size determines the number of values that we consider for our state variable, i.e. capital stock. As such, we intialize a vector $K = [k_1, \dots, k_N]$ of size 1000, say equally spaced between 0 and 5. The vector K represents our state space.

4. Next, we need to initialize the value function v and the updated value function Tv. These two things can easily be enoded using N-dimensional vectors $V = [v_1, \dots, v_N]$ where $V[i] \equiv v(k_i)$ gives the value of v at state k_i and $TV = [Tv_1, \dots, Tv_N]$ where $TV[i] \equiv Tv(k_i)$ gives the value of the function (Tv) at state k_i.

5. Finally, we need to encode the policy correspondence (or function) γ. We will do this by representing γ as an N-dimensional vector of integers $\Gamma = [\gamma_1, \dots, \gamma_N]$. The idea is that the ith component of Γ is equal to j, i.e. $\Gamma[i] = j$ if the value of γ at state k_i is given by k_j, i.e. $\gamma(k_i) = k_j$. In other words, if $\Gamma[i] = j$ then at k_i it will be optimal to set the next state equal to k_j.

6. Let's now go to the main part of the program. This embeds a loop that computes for each iteration the next update of the Bellman operator, i.e. given v, it computes Tv, until we have that:

$$\|TV - V\| = \max_i |TV[i] - V[i]| < \varepsilon.$$

   We program this as a while loop that iterates until this condition is satisfied.

7. Inside the loop we first have to update the value of V to TV. Notice that in order to do this, we first need to assign the value of TV to V ($V \equiv TV$).[48]

8. Next, we need to compute the new values of $TV[i] = Tv(k_i)$ for all states k_i in the grid. In order to do this, we iterate through the values $i = 1, \dots N$ and compute each time the values of $TV[i]$ and $\Gamma[i]$. This can be done using a For-loop.

   Given a particular value k_i (i.e. in the i-th iteration of the For-loop), we construct the values of the function,

$$f(k') = \begin{cases} \frac{(k_i^\alpha + (1-\delta)k_i - k')^{1-\sigma} - 1}{1-\sigma} + \beta v(k') & \text{if } k' \leq k_i^\alpha + (1-\delta)k_i, \\ -C & \text{if } k' > k_i^\alpha + (1-\delta)k_i \end{cases}$$

   where C is a very big number. This can be done by constructing an N-dimensional vector $F_i = [f(k_i), \dots, f(k_N)]$.

9. Next, the aim is to find the maximal element of the vector F. This maximal element will be the value $TV[i] = \max_j F_i[j]$. The index j at which this element is found, will be the new value of $\Gamma[i]$.

10. Don't forget to close the for and while loops.

Try to count the number of iterations that the program takes in order to converge (i.e. the number of iterations of the While-loop) and the time it takes to converge.

[48] If you do this in Python or Julia, the command $V = TV$ is not going to work as this only gives an alias. Instead you need to use the command $V = copy(Tv)$.

Figures 1, 2 and 3 give a plot of the policy function, the value function and the optimal level of consumption as functions of k.

The convergence rate of the Bellman operator has a rate of β. For many economic models, it is natural to choose β close to 1. Convergence of value function iteration method is particularly slow if β is chosen to be close to one.

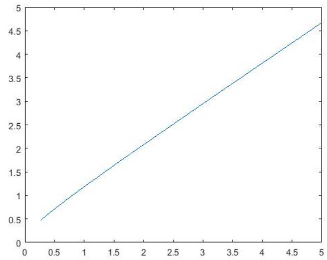


Figure 1: Present capital stock versus next periods capital stock.

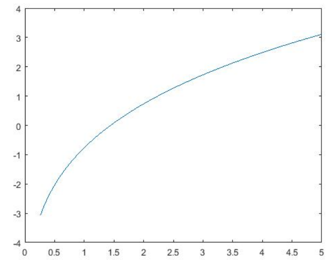


Figure 2: Value function.

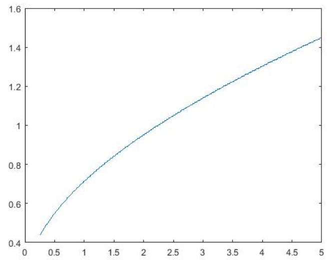


Figure 3: Consumption as a function of capital.

Interpolation

THE SPEED OF THE value function iteration depends on the size of the grid X. The larger X the more values of $Tv(x)$ we need to compute, and each involves an optimization procedure. If the state space is multidimensional, the size of the grid is even increasing exponentially in the dimension. This severely limits the speed of the algorithm. A first possible improvement for the speed of the algorithm is to decrease the size of this grid. However, we still would like to have a reasonable good estimate of the value of $Tv(x)$:

$$(Tv)(x) = \max_{a \in G(x)} \{F(x,a) + \beta v(r(x,a))\}.$$

Keeping x fixed, the quality of this estimate will depend on the grid size of A. The problem, however, is to obtain the value $v(r(x,a))$ on the right hand side when v is only known on the finite grid X. Indeed, it might be that $r(x,a)$ takes on a value that is not in this grid, which means that we cannot evaluate $v(r(x,a))$ for this particular value of a and x.

In our example, we had the Bellman operator:

$$(Tv)(k) = \max_{k' \leq k^\alpha + (1-\delta)k} \frac{(k^\alpha + (1-\delta)k - k')^{1-\sigma} - 1}{1-\sigma} + \beta v(k').$$

Here, we only have a small grid grid K on which we evaluate k, but we would like to perform the maximization using a finer grid for k'. As such, we need to be able to evaluate $v(k')$ also for values $k' \notin K$. The main change in our algorithm will therefore take place in step 8 of our algorithm. In particular, the function:

$$f(k') = \begin{cases} \frac{(k_i^\alpha + (1-\delta)k_i - k')^{1-\sigma} - 1}{1-\sigma} + \beta \tilde{v}(k') & \text{if } k' \leq k_i^\alpha + (1-\delta)k_i, \\ -C & \text{if } k' > k_i^\alpha + (1-\delta)k_i \end{cases}$$

is now computed on a denser grid. As such, the dimension for f will be larger than N. The function $\tilde{v}(k')$ can be approximated using interpolation of v on the N-dimensional vector v.

As an example, let us put the grid for the k equal to 20 and let the denser grid be equal to 1000. Adjust the code as indicated above using, for example, linear interpolation. After how many rounds does the code converge? What is the time of convergence?

The increase in speed is maily due to the large decrease of the grid size. Unfortunately, the value function is now only known at a smaller number of points and the interpolation function might be a bad guess for true value function. It is also not possible to prove convergence of the algorithm and convergence might even fail if, for example, the interpolation function is not well chosen.

Howard improvement (policy iteration)

The most important factor that determines the speed of the value function iteration algorithm is the optimization routine. Optimization is costly. Therefore, computational improvements that are aimed at reducing the number of times the optimization routine is called are very interesting. This is the idea behind the Howard improvement algorithm. Let H be the set of all potential policy functions:

$$H = \{g : X \to A : g(x) \in G(x)\}.$$

For any $g \in H$, we can define an operator R_g such that,

$$(R_g v)(x) = F(x, g(x)) + \beta v(r(x, g(x))).$$

This operator determines the value function resulting from using the function g as the choice rule. It is easily verified that the operator R_g from, say the set of bounded functions to the set of bounded functions on X, satisfies the conditions of Blackwell's theorem so it has a fixed point which can be obtained by iteration. This fixed point, say v, satisfies the condition:

$$v(x) = F(x, g(x)) + \beta v(r(x, g(x)))$$

which means that it computes the value of the infinite horizon problem under the constraint that the policy function g is used in every period. Importantly, the computation of this fixed point does not require any optimization routine, so it can be computed very fast.

The Howard improvement procedure takes the following form.

1. Decide on a grid X and A.
2. Pick any value function v_0.
3. Initiate the loop at $t = 1$ for all t, do the following
   (a) Find the policy function g_n such that,

$$g_{n(x)} = \arg\max_{a \in \Gamma(x)} F(x, a) + \beta v_{n-1}(r(x, a)).$$

(b) Find v_n as the unique fixed point of R_{g_n}, i.e.

$$v_n(x) = F(x, g_n(x)) + \beta v_n(r(x, g_n(x))).$$

(c) Iterate steps (a) and (b) each time updating $n \leftarrow n+1$, until convergence is met: $\|v_n - v_{n-1}\| < \epsilon$.

The Howard algorithm first converges on the value function given the policy function g_n. Once this function is found, the policy function g_n is updated using a maximization step. The advantage of this algorithm is that it requires fewer optimization iterations. Given that this is the most costly step, the algorithm is usually (much) faster.

The following theorem shows the validity of the Howard improvement algorithm.

Theorem 13. *The sequence of functions $(v_n)_{n\in\mathbb{N}}$ of the Howard algorithm converges to the fixed point of the Bellman operator T.*

Proof. Let T be the Bellman operator and let R_g be the policy function iterator for a given a policy function $g \in H$. Let v_n be the policy function obtained by the nth step of the algorithm. We will show that

$$v_0 \leq Tv_0 \leq v_1 \leq Tv_1 \leq \ldots$$

This is an increasing sequence in a bounded set,[49], so this sequence converges to the value $\sup_t v_t$ which is a fixed point of the Bellman operator T.

[49] Notice that X is a finite grid, so the number of distinct policy functions is finite and therefore also the number of value functions.

Let us first show that for all t, $Tv_n \geq v_n$. Indeed,

$$\begin{aligned}(Tv_n)(x) &= \max_{a\in G(x)} F(x,a) + \beta v_n(r(x,a)),\\ &\geq F(x, g_n(x)) + \beta v_t(r(x, g_n(x))) = v_n(x).\end{aligned}$$

The last equality follows from the fact that v_n is a fixed point of the operator R_{g_n}, so:

$$v_n(x) = F(x, g_n(x)) + \beta v_n(r(x, g_n(x))).$$

Next, we can show that $(Tv_n) = (R_{g_{n+1}} v_n)$. Indeed, by definition of g_{n+1}, we have:

$$(Tv_n)(x) = F(x, g_{n+1}(x)) + \beta v_t(r(x, g_{n+1}(x))) = (R_{g_{n+1}} v_t)(x).$$

Then, if we iterate $R_{g_{n+1}}$ a second time, we get:

$$\begin{aligned}(R^2_{g_{n+1}} v_n)(x) - (R_{g_n} v_n)(x) =& F(x, g_{n+1}(x)) + \beta (R_{g_{n+1}} v_n)(r(x, g_{n+1}(x))),\\ &- F(x, g_{n+1}(x)) - \beta v_n(r(x, g_{n+1}(x))),\\ &= \beta \left[(Tv_n)(r(x, g_{n+1}(x))) - v_n(r(x, g_{n+1}(x)))\right] \geq 0.\end{aligned}$$

where the last inequality follows from the fact that $Tv_t \geq v_t$. This shows that $(R^2_{g_{n+1}} v_n) \geq R_{g_{n+1}} v_n = Tv_n$.

Now let v_{n+1} be the fixed point of $(R_{g_{n+1}} v_n)$. We will show that $v_{n+1} \geq Tv_n$. In order to do this, we show that $(R^m_{g_{n+1}} v_t) \geq (R^{m-1}_{g_{n+1}} v_t)$ for all $m \geq 2$. As v_{n+1} is the limit of $(R^m_{g_{n+1}} v_n)$ for m going to infinity, this proves the assertion.

For $m = 2$, the proof is given above. Now for the induction step, we have:

$$
\begin{aligned}
&(R^m_{g_{n+1}} v_n)(x) \geq (R^{m-1}_{g_{n+1}} v_n)(x), \\
\leftrightarrow &F(x, g_{n+1}(x)) + \beta(R^{m-1}_{g_{n+1}} v_t)(r(x, g_{n+1}(x))) \geq F(x, g_{n+1}(x)) + \beta(R^{m-2}_{g_{n+1}} v_t)(r(x, g_{n+1}(x))), \\
\leftrightarrow &(R^{m-1}_{g_{n+1}} v_n)(r(x, g_{n+1}(x))) \geq (R^{m-2}_{g_{n+1}} v_n)(r(x, g_{n+1}(x))).
\end{aligned}
$$

which is indeed true by the induction hypothesis. Given that $v_{n+1} = \lim_m (R^m_{g_{n+1}} v_t)$, we have that

$$v_{n+1} \geq (R_{g_{n+1}} v_n) = Tv_n,$$

as was to be shown. □

The Howard algorithm can be implemented by using a call to a new function that computes the fixed point of the policy function mapping after each optimization routine of the value function iteration (i.e. after each For-loop iteration).

In order to compute the fixed point of the policy function, one can take the following steps.

1. Let the policy function Γ and the value function TV be the output of the maximization step of the value function iteration.

2. Initialize vectors W and $RW = V$

3. Do the following until $\|W - RW\| < \varepsilon$.

4. Assign $W = RW$ and compute the updated value for RW:

$$RW[i] = \frac{(k_i^\alpha + (1-\delta)K[i] - K[G[i]])^{1-\sigma} - 1}{1-\sigma} + \beta W[G[i]].$$

   Here we use index notation, where $K[G[i]]$ uses the index in $G[i]$ to get to the element $G[i] = j$ whenever $g(k_i) = k_j$. The same goes for $W[G[i]]$.

5. Close the while loop

6. Assign the updated value $TV = W$.

Compute the number of times the outer value function While-loop iterates until convergence and compute the time it takes to converge.

Instead of using a loop to compute the fixed point of R_g, it is sometimes possible to explicitly solve this step. Observe that the fixed point of the operator R_g satisfies,

$$v(k) = F(k, g(k)) + \beta v(g(k)).$$

This can be written in vector notation as,

$$V = F(K, K[G]) + \beta QV.$$

where Q is an $N \times N$ matrix with a 1 at position i, j if and only if $G[i] = 1$. This system can be solved for V,

$$V = (I - \beta Q)^{-1} F(K, K[G]).$$

This necessitates the inversion of the matrix $I - \beta Q$, which is computationally also costly (especially if the size of the grid is large). So it is not always the case that this gives a more efficient way of computing the fixed point of R_g.

Try to code the policy function iteration in this alternative way. For this, you first need to compute the matrix Q and invert $(I - \beta Q)$.

Parametric interpolation

THE SPEED OF THE iteration is slowed down by the size of the grid. The denser the grid, the better the approximation but also the longer the computation time. This is especially true when analyzing problem where the state space is more dimensional. If we have a grid of 100 points in a one dimensional setting to reach a certain level of accuracy, it takes approximately 100^2 grid points in a two dimensional problem, 100^3 grid points in a three dimensional problem and so on. As such, the problem becomes untractable even if there are only a moderate number of dimensions.

A possible way to avoid this problem is to exploiting the smoothness of typical economic examples to approximate the value function by some flexible functional form.[50] This flexible functional form usually depends on some some parameter values. The idea is then to iterate on the value of these parameters to approximate the value function as close as possible. The algorithm then takes the following steps,

[50] Notice however, that the convergence of the function iteration algorithm with nonlinear approximations of the value function is not guaranteed.

1. Decide on some flexible functional form $v(x, \sigma)$ and a grid of possible values X. Here σ is a set of parameters that fully determines the function $v(., \sigma)$.

2. Choose a critical value ε and a starting value for the parameters σ_0. Set $t = 1$,

3. Compute,
$$(Tv)(x) = \max_{y\in\Gamma(x)} F(x,y) + \beta v(y, \sigma_{t-1}).$$

4. Update the vector of parameters σ_t to make $v(x, \sigma_t)$ as close as possible to $(Tv)(x)$.

5. if $\|v(x, \sigma_t) - v(x, \sigma_{t-1})\| < \varepsilon$ then stop. Else set $t = t + 1$, and go back to step 2.

For this algorithm to be implementable we need to choose a specific functional form $v(., \sigma)$ and we need to decide on the updateing rule in step 4. For the first issue, one usually choses either a combination of polynomials, neural networks or splines. Concerning step 4, one can uses the value of σ_{t+1} that minimizes the sum of squares.

$$\sigma_{t+1} = \arg\min_{\sigma} \sum_{x\in X} \left[(Tv)(x) - v(x, \sigma)\right]^2.$$

Fortunately, the algorithm is usually much faster in high dimensional settings, with a comparable accuracy.[51]

[51] For the least squares estimates to be of high quality, the number of grid points should be considerably larger than the number of parameters θ to be estimated.

Let us have a look at an example where $v(.)$ is approximated using polynomials. One attractive option is the use of Chebychev polynomials. The n-th Chebychev polynomial is defined on the interval $[-1, 1]$ and has the form,

$$T_n(x) = \cos(n \cos^{-1} x).$$

Or equivalently, $T_n(\cos(\theta)) = \cos(n\theta)$. We know that,

$$\begin{aligned}
T_0(\cos(0)) &= \cos(0\theta) = 1, \\
T_1(\cos(\theta)) &= \cos(\theta), \\
T_2(\cos(\theta)) &= \cos(2\theta) = 2\cos^2\theta - 1, \\
T_3(\cos(\theta)) &= \cos(3\theta) = 4\cos^3\theta - 3\cos\theta
\end{aligned}$$

This gives for the first terms,

$$\begin{aligned}
T_0(x) &= 1, \\
T_1(x) &= x, \\
T_2(x) &= 2x^2 - 1, \\
T_3(x) &= 4x^3 - 3x.
\end{aligned}$$

Now,

$$\begin{aligned}
e^{i(n+1)\theta} &= e^{in\theta} e^{i\theta}, \\
\rightarrow \cos((n-1)\theta) &= \cos(n\theta)\cos(\theta) - \sin(n\theta)\sin(\theta).
\end{aligned}$$

We have that $e^{iz} = (\cos z + i \sin z)$. The second line is obtained by gathering the real terms of the two expansions.

We would like to get rid of the sin functions. We can do this by considering the following expansion,

$$e^{i(n-1)\theta} = e^{in\theta}e^{-i\theta},$$
$$\rightarrow \cos((n-1)\theta) = \cos(n\theta)\cos(\theta) + \sin(n\theta)\sin(\theta).$$

Adding the two together gives,

$$\cos((n+1)\theta) + \cos((n-1)\theta) = 2\cos(n\theta)\cos(\theta).$$

As such,

$$T_{n+1}(x) + T_{n-1}(x) = 2T_n(x)x,$$
$$\rightarrow T_{n+1}(x) = 2T_n(x)x - T_{n-1}(x).$$

From this, we can compute,

$$T_4(x) = 8x^4 - 6x^2 - 2x^2 + 1 = 8x^4 - 8x^2 + 1,$$
$$T_5(x) = 16x^5 - 16x^3 + 2x - 4x^3 + 3x = 16x^5 - 20x^3 + 5x,$$
$$\ldots$$

This shows that they can easily be computed iteratively. The roots of the polynomial T_n are equal to the values $\cos\left(\frac{(2k+1)\pi}{2n}\right)$. Indeed, we must have that,

$$T_n(\cos(\theta)) = \cos(n\theta) = 0,$$
$$\rightarrow n\theta = \frac{\pi}{2} + k\pi,$$
$$\rightarrow \theta = \frac{\pi(2k+1)}{2n},$$
$$\rightarrow x = \cos\left(\frac{\pi(2k+1)}{2n}\right).$$

The grid points are often chosen to be these zeros.

The value function $v(x;\sigma)$ can be chosen such that,

$$v(x;\sigma) = \sum_{t=1}^{N} \sigma_t T_n\left(2\frac{x-\underline{x}}{\overline{x}-\underline{x}} - 1\right).$$

This is a linear combination of Chebychev polynomials. The normalization $2\frac{(x-\underline{x})}{\overline{x}-\underline{x}} - 1$ is done to rescale the state variables to the interval $[-1,1]$.

To start we begin by some parameter initializations. The parameters are the same as before. On line 30, we define the Chebychev polynomials and we initiate estimates for our parameters by regressing these on the estimate of our value function.[52]

[52] In matlab the OLS estimates are quickly computes using the command $X\backslash v$.

```
%parameter values
par.sigma   = 1.50;
par.delta   = 0.10;
par.beta    = 0.95;
par.alpha   = 0.30;

par.ngrids  = 20;       %size of the grid
par.n       = 10;       %number of Chebychev polynomials
par.epsi    = 1e-6;     %small value


crit    = 1;         %initiation critical value
iter    = 1;         %counter for number of iterations


%steady state
ks = ((1-par.beta*(1-par.delta))/(par.alpha*par.beta))^(1/(par.alpha-1));
dev = 0.9;

par.kmin = (1-dev)*ks;      %minimal capital stock
par.kmax = (1+dev)*ks;      %maximal capital stock

rk = cos((2*[1:par.ngrids]'-1)*pi/(2*par.ngrids));   %grid for Chebychev polynomials
K = par.kmin+(rk+1)*(par.kmax-par.kmin)/2;      %grid for capital stock values

%starting value
v = ((K.^par.alpha).^(1-par.sigma)-1)/((1-par.sigma)*(1-par.beta));

%chebychev functions
X = chebychev(rk,par.n);

%starting value for coefficients
sig = X\v;


Tv = zeros(par.ngrids,1);
Knext = K;
```

The function *chebychev*(rk, n) creates the values of the n Chebychev polynomials on a grid given by rk. In order to compute theis, we consider some cases where $n = 0$ or 1. If n is larger or equal to two, we define the polynomials iterative,

$$T_n(x) = 2xT_{n-1}(x) - T_{n_2}(x).$$

```
function Tx = chebychev(x,n)

X = x(:);
lx = size(X,1);
if n<0;
error('n should be a positive integer');
end

switch n;
case 0;
    Tx = [ones(lx,1)];
case 1;
```

```
    Tx = [ones(lx,1) X];
otherwise
    Tx = [ones(lx,1) X];
    for i= 3:n+1
        Tx = [Tx 2*X.*Tx(:,i-1)-Tx(:,i-2)];
    end
end
```

Let's go back to the main program. The next part is the main loop. On line 11 we are minimizing a function *tv* which computes the function value

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

Then the estimates of σ are updated and the criterium $\|Tv - v\|$ is computed to determine when to stop the algorithm. The function *fmincon* is a minimization routine, so we have to minimize $-F(x,y) - \beta v(y,\sigma)$. This is why we change signs on line 12.

```
1  options = optimset('Display','off');
2  while crit > par.epsi;
3      k0 = K(1);
4      for i = 1:par.ngrids
5          %captial stock
6          k = K(i);
7          par.k = k;
8          %upper bound
9          b = k^par.alpha + (1-par.delta)*k;
10         %optimal value of next period capital stock
11         [Knext(i), Tv(i)] = fmincon(@(x)tv(x,par, sig),Knext(i), [], [],[],[],0,b,[], options);
12         Tv(i) = -Tv(i);
13     end
14     %update estimators
15     sig = X\Tv;
16
17     crit = max(abs(Tv-v));
18     disp(crit);
19
20     v = Tv;
21     iter = iter+1;
22 end
```

The function to be minimized is given by *tv* and is given by the following code. After retrieving some parameter values, it computes the value function $v(y,\sigma)$ on line 13 via a separate function *value*, the consumption and utility and returns the value of $res = -F(x,y) - \beta v(y,\sigma)$.

```
1      function res = tv(kp, par, theta)
2
3      alpha   = par.alpha;
4      beta    = par.beta;
5      delta   = par.delta;
6      sigma   = par.sigma;
```

```
kmin     = par.kmin;
kmax     = par.kmax;
n        = par.n;
k        = par.k;

%value next period
v = value(kp,[kmin kmax n], theta);
%consumption
c = k.^alpha + (1-delta)*k - kp;
%utility
util = (c.^(1-sigma)-1)/(1-sigma);
%objective function to be minmized
res = -(util+beta*v);
```

The final program is *value* which computes $v(x,\sigma)$. It does so by rescaling the capital stocks and computing,

$$v(x,\sigma) = \sum_{i=1}^{N} \sigma_n T_n(w).$$

where w are the rescaled capital stock values.

```
function v = value(k,param,theta)

kmin = param(1);
kmax = param(2);
n = param(3);
k = 2*(k - kmin)/(kmax-kmin)-1;
v = chebychev(k,n)*theta;
```

The program terminates after 243 iterations in 32.30 seconds.[53] The final parameter values are given by,

parameter	value
σ_0	0.8237
σ_1	2.7804
σ_2	-0.6601
σ_3	0.2370
σ_4	-0.1028
σ_5	0.0515
σ_6	-0.0260
σ_7	0.0113
σ_8	-0.0062
σ_9	0.0050
σ_{10}	-0.0028

[53] Here, we see that the computation time is much longer than before. However, the advantage is that we can approximate the value function for any particular value of k. Judd and Solnick (1994) successfully applied this technique to the optimal growth model and found that the approximations was very good and dominates to a lot of other methods.

Some applications

Let us have a look at some applications of dynamic programming under certainty.

Optimal tree growth

Consider a tree whose growth is described by a function $h : \mathbb{R}_+ \to \mathbb{R}_+$. In particular, if k_t is the length of the tree in period t then $k_{t+1} = h(k_t)$ is the height of the tree tomorrow. Assume that the price of wood is one per meter of tree, and the interest rate r are both constant over time. Set $\beta = 1/(1+r)$. It is costless to cut down the tree.

If the tree cannot be replanted, the problem in each period is either to cut the tree or not. If the tree is cut in period t then the value is given by $v(k_t) = k_t$ and there is no value thereafter. If the tree is not cut in period t then the value is given by $v(k_t) = \beta v(h(k_t))$. Each period, the problem is either to cut the tree or not. As such:

$$v(k_t) = \max_{c=\{0,1\}} \{k_t c + (1-c)\beta v(h(k_t))\}.$$

Here c is a binary variable that decides whether to cut the tree or not. Observe that his problem can be rewritten as:

$$v(k_t) = \max\{k_t ; \beta v(h(k_t))\}.$$

The first choice is taken when the tree is cut while if the second option is take the tree is not cut. Assume that there is a maximum height that the tree can take, $k \in [0, H]$.

Theorem 14. *The operator* $(Tv)(k) = \max\{k, \beta v(h(k))\}$ *is a contraction mapping from the set of bounded functions* $B([0, H])$ *to* $B([0, H])$*.*

Proof. We check Blackwell's theorem. If $v \leq w$ then

$$(Tv)(k) = \max\{k, \beta v(h(k))\} \leq \max\{k, \beta w(h(k))\} = (Tw)(k),$$

which shows monotonicity. For additivity,

$$
\begin{aligned}
(Tv + a)(k) &= \max\{k, \beta(v+a)(h(k))\}, \\
&= \max\{k, \beta v(h(k)) + \beta a\}, \\
&\leq \max\{k + \beta a, \beta v(h(k)) + \beta a\} = (Tv)(k) + \beta a.
\end{aligned}
$$

□

As such, we know that T has a fixed point. In order to get an idea of the shape of v, we start by a simulation. We set $H = 15$ and consider a grid of fifteen values of $k = 1, 2, \ldots, 15$. We specify $h(k) = k + 0.25(H - k)$ so every period the growth of the tree equals one fourth of the distance between its height and the maximal height.[54] The value function is given in Figure 4. We see that for low values of k, $v(k)$ is above the diagonal, which means that the tree will not be cut as:

$$v(k) > k.$$

[54] I don't know if this is realistic but it leads to a concave growth path.

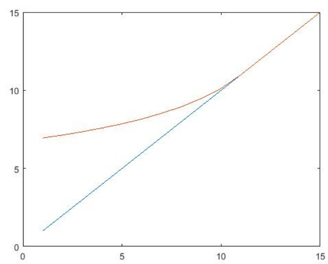


Figure 4: Value function and main diagonal.

For high values of k, we have that $v(k) = k$, which means that the tree will be cut. This indicates that (at least in this case) there is a unique cutoff height k^* that determines the minimal height for the tree to be cut.

At k^*, the decision maker should then be indifferent between cutting the tree or not. As such, k^* should satisfy the condition:

$$k^* = \beta v(h(k^*)).$$

In addition $h(k^*) \geq k^*$ so we know that, if we adhere to the conjecture that the tree is cut if it is longer than k^*, for height $h(k^*)$, the tree will also be cut, i.e. $v(h(k^*)) = h(k^*)$. As such:

$$
\begin{aligned}
k^* &= \beta v(h(k^*)) = \beta h(k^*), \\
\rightarrow \frac{h(k^*)}{k^*} &= 1/\beta.
\end{aligned}
$$

The left hand side gives the proportional growth of a tree of height k^*. The right hand side gives the interest rate $(1 + r) = 1/\beta$, i.e. the cost of waiting. Intuitively, if the left hand side is greater than the right hand side, it will be optimal not to cut the tree. Otherwise, cutting is optimal.

The following puts an assumption on the function $h(k)$ that guarantees that this reasoning is correct:

Assumption 1. *Assume that there is a unique $k^* \in [0, H]$ such that,*

- *if $k \geq k^*$ then $\frac{h(k)}{k} \leq \frac{1}{\beta}$*
- *if $k < k^*$ then $\frac{h(k)}{k} > \frac{1}{\beta}$.*

Theorem 15. *If assumption 1 is satisfied, then it is optimal to cut the tree for all $k \geq k^*$.*

Proof. Consider the fixed point v of the Bellman operator. It suffices to show that $v(k) = k$ whenever $k \geq k^*$, i.e. it is optimal to cut the tree if $k \geq k^*$. First, notice that the Bellman operator T with:

$$(Tv) = \max\{k, \beta v(h(k))\}.$$

is a contraction mapping. Let

$$C = \{v \in B([0, H]) : \forall k \geq k^*, v(k) = k\}.$$

The proofs works by showing that (i) C is a closed set and (ii) for all $v \in C$: $Tv \in C$, as then the fixed point of T will also be in C.

For the first, let $(v_n)_{n\in\mathbb{N}}$ be a sequence in C and $\|v_n - v\| \overset{n}{\to} 0$. Now, if $k \geq k^*$ then for all n, $v_n(k) = k$, so by taking pointwise limits: $v(k) = k$.

For the second part, let $v \in C$. We need to show that $Tv \in C$ or equivalently, for all $k \geq k^*$, $(Tv)(k) = k$. Assume that $k \geq k^*$. Then,

$$(Tv)(k) = \max\{k, \beta v(h(k))\}.$$

We know that $h(k) \geq k \geq k^*$, and $v \in C$, so $v(h(k)) = h(k)$. This gives,

$$(Tv)(k) = \max\{k, \beta h(k)\}.$$

As $k \geq k^*$ we also know that, by assumption, $h(k) < k/\beta$, so,

$$(Tv)(k) = \max\{k, \beta h(k)\} = k.$$

This shows that $Tv \in C$ as desired. □

The next result states that it is not optimal to cut the tree if $k < k^*$.

Theorem 16. *If assumption 1 is satisfied, then for all $k < k^*$ it is better to wait.*

Proof. As before let v be the fixed point of the Bellman operator. We need to show that for $k < k^*$, $\beta v(h(k)) > k$, i.e. it is optimal to wait.

Let

$$C = \{v \in B([0, H]) : \forall k < k^*, \beta(v(h(k))) > k\}.$$

Notice that it suffices to show that for all functions v, $Tv \in C$, i.e. for $k < k^*$, $\beta(Tv)(h(k)) > k$.

Indeed, let $k < k^*$. Then:

$$\beta(Tv)(h(k)) = \beta \max\{h(k), \beta v(h(h(k)))\} \geq \beta h(k) > k.$$

The last inequality follows from the fact that for $k < k^*$: $h(k)/k > 1/\beta$.

□

Above two results show that if assumption 1 is satisfied. Then there is a unique $k^*(= h(k^*)/\beta)$ such that for all $k < k^*$ the tree is not cut and for all $k \geq k^*$ the tree will be cut.

Optimal policy business cycles

The effectiveness of monetary economic policy depends on the expectations of the agents in the economy. Assume that the deviation of y_t which is the log of output from its natural level y^* is given by the following Philips curve:

This model is borrowed from Ginsburgh and Michel, 1998, Optimal policy business cycles, Journal of Economic Dynamics and Control.

$$(y_t - y^*) = \gamma(\pi_t - \pi_t^e),$$

where π_t and π_t^e are the actual and expected rate on inflation in period t. Here $\gamma > 0$. This claims that only unexpected inflation can push output above its natural level. The policy maker's objective in each period is given by a trade off between more output and less inflation:

$$g(y_t, \pi_t) = \alpha(y_t - y^*) - \frac{\pi_t^2}{2} = \alpha\gamma(\pi_t - \pi_t^e) - \frac{\pi_t^2}{2}.$$

The forward looking government has a discount factor δ so the problem is to maximize:

$$\sum_{t=0}^{\infty} \delta^t \left(\alpha\gamma(\pi_t - \pi_t^e) - \frac{\pi_t^2}{2} \right).$$

If agents have rational expectations then expected inflation equals actual inflation, so $\pi_t^e = \pi_t$ and therefore $y_t = y^*$. In this case, the optimal policy is to set $\pi_t = 0$ at every point in time. Now, assume that not all agents have rational expectations. Some agents have adaptive expectations in the sense that:[55]

[55] For example, they form expectations that are adaptive.

$$\pi_{t+1}^a = \lambda\pi_t + (1-\lambda)\pi_t^a,$$

where $\lambda \in (0, 1]$. Assume that a proportion x_t of agents use rational expectations while a fraction $(1 - x_t)$ form adaptive expectations. We assume that the average expected rate of inflation is a weighted average of the rates expected by rational and adaptive agents:

$$\pi_t^e = x\pi_t + (1-x)\pi_t^a.$$

then,

$$\begin{aligned}
&\sum_{t=0}^{\infty} \delta^t \left(\alpha\gamma(\pi_t - \pi_t^e) - \frac{\pi_t^2}{2} \right), \\
&= \sum_{t=0}^{\infty} \delta^t \left(\alpha\gamma(1-x)(\pi_t - \pi_t^a) - \frac{\pi_t^2}{2} \right).
\end{aligned}$$

The government will try to set π_t such as to maximize this payoff. The Bellman equation is:

$$v(\pi_t^a) = \max_{\pi} \left\{ \alpha\gamma((1-x)(\pi_t - \pi_t^a) - \frac{\pi_t^2}{2} + \delta v(\lambda \pi_t + (1-\lambda)\pi_t^a) \right\}.$$

Let us try to derive the Euler equations. Let $q_t = v'(\pi_t^a)$ then the first order condition and envelope theorem give:

$$\begin{aligned} q_t &= -\alpha\gamma(1-x) + \delta(1-\lambda)q_{t+1}, \\ 0 &= \alpha\gamma(1-x) - \pi_t + \delta\lambda q_{t+1}. \end{aligned}$$

So, eliminating the q_t, q_{t+1} variables gives:

$$\begin{aligned} &\frac{\pi_{t-1} - \alpha\gamma(1-x)}{\delta\lambda} = -\alpha\gamma(1-x) + \delta(1-\lambda)\frac{\pi_t - \alpha\gamma(1-x)}{\delta\lambda}, \\ \leftrightarrow &\pi_{t-1} - \alpha\gamma(1-x) = -\alpha\gamma(1-x)\delta\lambda + \delta(1-\lambda)\pi_t - \alpha\gamma(1-x)\delta(1-\lambda), \\ \leftrightarrow &\pi_{t-1} - \delta(1-\lambda)\pi_t = \alpha\gamma(1-x)(1-\delta), \\ \leftrightarrow &\pi_t - \frac{\pi_{t-1}}{\delta(1-\lambda)} = -\frac{\alpha\gamma(1-x)(1-\delta)}{\delta(1-\lambda)}. \end{aligned}$$

This is an explosive difference equation, so the only solution is the one at the steady state, where

$$\pi^* = \frac{\alpha\gamma(1-x)(1-\delta)}{1-\delta(1-\lambda)}.$$

Now let us endogeneize the share of rational agents x_t. Assume that at time t decisions are made at no cost on the basis of adaptive expectations π_t^a. An agent θ can modify this decision at a fixed cost c using the new information π_t. There is a continuum of agents $\theta \in [0,1]$. Agent θ makes a cost equal to $\theta(\pi_t^a - \pi_t)^2$ when he uses π_t^a instead of π_t. let $\underline{\theta}_t$ be defined by,

$$\underline{\theta}_t(\pi_t^a - \pi_t)^2 = c.$$

The loss of agent θ is larger than c if $\theta \geq \underline{\theta}_t$. and the proportion x_t of agents that decide to change their decision is,

$$x_t = x(\pi_t^a, \pi_t) = \max\{0, 1 - c(\pi_t^a - \pi_t)^{-2}\}.$$

The Bellman equation is now,

$$\begin{aligned} v(\pi_t^a) = \max_{\pi_t} &\left\{ \beta(1-x_t)(\pi_t - \pi_t^a) - \frac{\pi_t^2}{2} + \delta v(\lambda\pi_t + (1-\lambda)\pi_t^a) \right\}, \\ \text{s.t. } &x_t = \max\{0, 1 - c(\pi^a - \pi_t)^{-2}\}. \end{aligned}$$

The model is a bit daunting to analyze analytically, so we will resort to a simulation exercise. We use parameter values $\beta = 0.1$,

$c = 0.0001$, $\lambda = 0.75$ and $\delta = 0.95$. Also, we use grid of 1000 values of π^a between -0.1 and 0.05. Figure 5 plots π^a_{t+1} against the value of π^a_t. The stable state is situated at the point where the curve intersects with the diagonal. One sees that below the steady state the best response is above the diagonal. So, π^a increases over time. Suddenly the best response drops to below the diagonal. This shows that π^a will show cyclical behaviour. It will gradually increase and then suddenly drop to a lower value after which it will start increasing again.

Figure 6 shows a the evolution of inflation over time. Here the cyclical behaviour is clearly visible. For this example, we have cycles of length 6. In 5 periods, inflation increases stepwise. In the sixth period inflation drops again to its starting value.

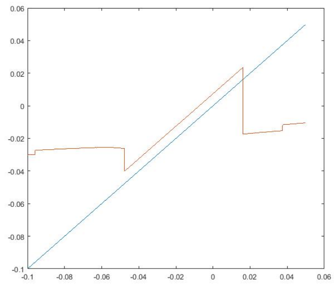


Figure 5: Value of π^a_{t+1} against the value of π^a_t.

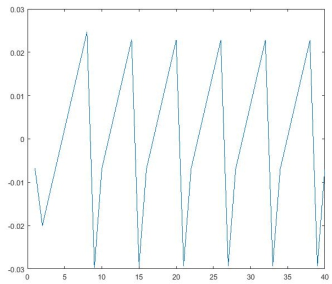


Figure 6: Evolution of π over time.

Stochastic dynamic programming

INTUITIVELY, A STOCHASTIC dynamic program has the same components as a deterministic one. The only (major) difference is that the transition that governs the process of going from one state to another is no longer certain (i.e. deterministic). When transitions occur probabilistically, then the state and action today only gives a distribution over possible states tomorrow.

As before, we use X to be the set of states and let A be the set of actions. Also similar as before, we define a correspondence $G : X \to A$ that determines which actions the agent can take for a given state $x \in X$. In the deterministic case, the state in the next period was given by the function $r(x, a)$. This is no longer the case in the stochastic world. Instead of the transition function we now introduce a Markov transition kernel.

$$Q((x, a), B),$$

Here $Q((x, a), B)$ gives the probability that the state in the next period is in the (measurable) set $B \subseteq X$ if the state today is x and the action taken today is a.[56] Notice that the probability law of the state tomorrow only depends on the state and action taken today. In particular, it does not depend on what happened before today. A stochastic process that does not depend on the past, given the present is called a Markov process. We will only look at Markov processes.

[56] Formally, $Q((x, a), B)$ is a kernel if $Q((x, a), .)$ is a probability measure for all (x, a) and $Q(., B)$ is a measurable function for all measurable sets B.

Let $f : X \times A \to \mathbb{R}$ be a payoff function. The expected value of f tomorrow if (x, a) is the current state and $a' \in G(x)$ is chosen tomorrow is given by:

$$\int_X Q((x, a), d\tilde{x}) f(\tilde{x}, a')$$

In setting without uncertainty, the intertemporal maximization problem was to look for a feasible paths of actions $a_0, a_1, \ldots, a_n, \ldots$. In a stochastic world, we have to be a bit more careful. As the evolution of the states is stochastic, the path of actions may also be stochastic.

In the stochastic framework, we will assume that the object of choice is a policy function $\gamma : X \to A$, that determines for each state

$x \in X$ today an action $a \in G(x)$ that is taken by the decision maker. This implies that (i) the action is not stochastic (given the current state) and (ii) the action taken only depends on the current state and not on the history of states in the past.

Assume that the current state is x_0 and assume that the decision maker follows the policy rule γ, then the next period's expected payoff is given by:

$$E[\beta F(x_1, \gamma(x_1))|x_0] = \int_X Q((x_0, \gamma(x_0)), dx_1)\beta F(x_1, \gamma(x_1)).$$

Here we write the expectation conditional on x_0. The expected payoff within two periods F two periods from now is given by:

$$\begin{aligned} &E[\beta^2 F(x_2, \gamma(x_2))|x_0], \\ &= \int_X Q((x_0, \gamma(x_0)), dx_1) \int_X Q((x_1, \gamma(x_1), dx_2)\beta^2 F(x_2, \gamma(x_2)). \end{aligned}$$

In general, the expected payoff that is obtained in period n is given by,

$$\begin{aligned} u_n(\gamma) &= E\left[\sum_{t=1}^n \beta^t F(x_t, \gamma(x_t)) \middle| x_0\right], \\ &= F(x_0, \gamma(x_0)) + \int_X Q((x_0, \gamma(x_0)), dx_1)\beta F(x_1, \gamma(x_1)), \\ &+ \int_X Q((x_0, \gamma(x_0)), dx_1) \int_X Q((x_1, \gamma(x_1)), dx_2)\beta^2 F(x_2, \gamma(x_2)), \\ &+ \ldots, \\ &+ \int_X Q((x_0, \gamma(x_0)), dx_1) \int_X Q((x_1, \gamma(x_1)), dx_2) \int_X \ldots \int_X Q((x_{n-1}, \gamma(x_{n-1})), dx_n)\beta^n F(x_n, \gamma(x_n)). \end{aligned}$$

Let $u_\infty(\gamma) = \lim_n u_n(\gamma)$, if it exists. The aim of the decision maker is to find a policy function γ to maximize $u_\infty(\gamma)$,[57]

$$\max_{\gamma: X \to A} u_\infty(\gamma).$$

[57] Observe that we have not showed yet that this maximization problem is well defined.

The aim of this chapter is to relate the solution of this problem (if it exists) to the solution of the following functional equation,

$$\begin{aligned} v(x) &= \max_{a \in G(x)} \left\{ F(x, a) + \beta \int_X Q((x, a), d\tilde{x}) v(\tilde{x}) \right\}, \\ &= \max_{a \in G(x)} \{F(x, a) + \beta \mathbb{E}\left[v(\tilde{x})|x, a\right]\} \end{aligned}$$

This is the Bellman equation for the stochastic problem. It is related to the following Bellman operator T,

$$(Tv)(x) = \max_{a\in G(x)} \{F(x,a) + \beta\mathbb{E}[v(\tilde{x})|x,a]\},$$
$$= \max_{a\in G(x)} \left\{F(x,a) + \beta \int_X Q((x,a),d\tilde{x})v(\tilde{x})\right\}.$$

Definition 13 (regularity). *The stochastic dynamic programming problem (X, A, G, F, β, Q) is regular if the instantaneous payoff function $F : X \times A \to \mathbb{R}$ is continuous, the transition function $G : \Omega \to X$ is non-empty and continuous (u.h.c. and l.h.c.) and there exists a continuous function $\phi : \Omega \to \mathbb{R}_{++}$ such that,*

1. *There exists an $M \geq 0$ such that for all $x \in X$,*

$$\max_{a\in G(x)} |F(x,a)| \leq M\phi(x).$$

2. *There exists a $\theta \in (0,1)$ such that for all $x \in X$,*

$$\beta \max_{a\in G(x)} \int_X Q((x,a),d\tilde{x})\phi(\tilde{x}) \equiv \beta \max_{a\in G(x)} \mathbb{E}[\phi(\tilde{x})|x,a] \leq \theta\phi(x).$$

3. *If $f : X \to \mathbb{R}$ is an element of $C_\phi(X)$, then for all $a \in A$:*

$$R(x,a) = \int_X Q((x,a),d\tilde{x})f(\tilde{x}) \equiv \mathbb{E}[f(\tilde{x})|x,a].$$

is continuous.

Condition 3 is called the Feller condition.

Theorem 17. *If the problem (X, A, G, F, β, Q) is regular then the Bellman operator maps $C_\phi(X)$ into $C_\phi(X)$ and is a contraction mapping.*

Proof. Let $v \in C_\phi(X)$. We have that:

$$(Tv)(x) = \max_{a\in G(x)} \{F(x,a) + \beta\mathbb{E}[v(\tilde{x})|x,a]\}$$

By the regularity condition (3), the objective function is continuous. As $G(.)$ is also continuous, we have that the optimization problem is well defined and $(TV)(x)$ is a continuous function.

Let us show that T maps $C_\phi(X)$ into $C_\phi(X)$, i.e. $\|Tv\|_\phi$ is bounded. In order to see this, let $\|v\|_\phi \leq N$. Then:

$$|(Tv)(x)| = \left|\max_{a\in G(x)} \left\{F(x,a) + \beta \int_X Q((x,a),d\tilde{x})v(\tilde{x})\right\}\right|,$$
$$\leq \max_{a\in G(x)} |F(x,a)| + \beta \max_{a\in G(x)} \int_X Q((x,a),d\tilde{x})|v(\tilde{x})|,$$
$$\leq \max_{a\in G(x)} |F(x,a)| + \beta \max_{a\in G(x)} \int_X Q((x,a),d\tilde{x})N\phi(\tilde{x}),$$
$$\leq M\phi(x) + \theta N\phi(x),$$
$$= (M + \theta N)\phi(x).$$

so $|(Tv)(x)|/\phi(x)$ is bounded by $M + \theta N$ which is finite.

For a contraction mapping, we verify Blackwell's conditions. For monotonicity, let $v \geq w$ then

$$\begin{aligned}
(Tv)(x) &= \max_{a \in G(x)} \left\{ F(x,a) + \beta \int_X Q((x,a), d\tilde{x}) v(\tilde{x}) \right\}, \\
&\geq \max_{a \in G(x)} \left\{ F(x,a) + \beta \int_X Q((x,a), d\tilde{x}) w(\tilde{x}) \right\}, \\
&= (Tw)(x).
\end{aligned}$$

For additivity,

$$\begin{aligned}
(T(v + \alpha\phi))(x) &= \max_{a \in G(x)} \left\{ F(x,a) + \beta \int_X Q((x,a), d\tilde{x})(v + \alpha\phi)(\tilde{x}) \right\}, \\
&\leq (Tv)(x) + \beta\alpha \max_{a \in G(x)} \int_X Q((x,a), d\tilde{x}) \phi(\tilde{x}), \\
&\leq (Tv)(x) + \theta\alpha\phi(x),
\end{aligned}$$

as was to be shown. □

Now, let's go back to our original problem,

$$\max_{\gamma : X \to A} u_\infty(\gamma).$$

We will relate the solution to this problem with the fixed point of the Bellman operator.

Lemma 7. *Let (X, A, F, G, β, Q) be a regular problem. Let $\gamma : X \to A$ be a policy function (i.e. $\gamma(x) \in G(x)$). Then $u_\infty(\gamma)$ exists.*

Proof. For a given policy function γ, we have that:

$$u_T(\gamma) = \sum_{t=0}^{T} \mathbb{E}[\beta^t F(x_t, \gamma(x_t)) | x_0].$$

Consider the t-th term in this summation:

$$\mathbb{E}[\beta^t F(x_t, \gamma(x_t)) | x_0] = \beta^t \int_X Q((x_0, \gamma(x_0)), dx_1) \ldots \int_X Q((x_{t-1}, \gamma(x_{t-1})), dx_t) F(x_t, \gamma(x_t)).$$

We will establish an upper bound on this term. First, take the innermost integral:

$$\begin{aligned}
\int_X Q((x_{t-1}, \gamma(x_{t-1})), dx_t) F(x_t, \gamma(x_t)) &\leq \int_X Q((x_{t-1}, \gamma(x_{t-1})), dx_t) |F(x_t, \gamma(x_t))|, \\
&\leq \int_X Q((x_{t-1}, \gamma(x_{t-1})), dx_t) M\phi(x_t), \\
&\leq M \frac{\theta}{\beta} \phi(x_{t-1})
\end{aligned}$$

Then taking the two inner integrals,

$$\begin{aligned}
&\int_X Q((x_{t-2}, \gamma(x_{t-2})), dx_{t-1}) \int_X Q((x_{t-1}, \gamma(x_{t-1})), dx_t) F(x_t, \gamma(x_t)),\\
&\leq \int_X Q((x_{t-2}, \gamma(x_{t-2})), dx_{t-1}) M \frac{\theta}{\beta} \phi(x_{t-1}),\\
&\leq M \frac{\theta^2}{\beta^2} \phi(x_{t-2}).
\end{aligned}$$

Iterating over all integrals, gives that this $t-$th term satisfies:

$$\mathbb{E}[\beta^t F(x_t, \gamma(x_t)) | x_0] \leq \beta^t M \frac{\theta^t}{\beta^t} \phi(x_0) = M\theta^t \phi(x_0).$$

Now let us show that $u_n(\gamma)$ is a Cauchy sequence. Using the bound established above, we have that:

$$\begin{aligned}
|u_n(\gamma) - u_m(\gamma)| &= \sum_{t=m+1}^{n} \mathbb{E}[\beta^t F(x_t, \gamma(x_t)) | x_0],\\
&\leq \sum_{t=m+1}^{n} \beta^t \mathbb{E}\left[|F(x_t, \gamma(x_t))| | x_0 \right],\\
&\leq \sum_{t=m+1}^{n} M\theta^t \phi(x_0),\\
&= M\phi(x_0) \sum_{t=m+1}^{n} \theta^t\\
&\leq M\phi(x_0) \frac{\theta^{m+1}}{1-\theta} \to^m 0.
\end{aligned}$$

This shows that $(u_n(\gamma))_{n\in\mathbb{N}}$ is Cauchy, so the sequence converges, which means that $u_\infty(\gamma) = \lim_{n\to\infty} u_n(\gamma)$ exists.

□

Next, let us show that the fixed point of the Bellman operator is greater than $u_\infty(\gamma)$ for any policy function γ.

Lemma 8. *Let (X, A, F, G, β, Q) be a regular problem. Let γ be a policy function and let v be the fixed point of the Bellman operator, then $v(x_0) \geq u_\infty(\gamma)$.*

Proof. We have that,

$$\begin{aligned}
v(x_0) &\geq F(x_0, \gamma(x_0)) + \beta \int_X Q((x_0, \gamma(x_0), dx_1) v(x_1),\\
&\geq F(x_0, \gamma(x_0)) + \beta \int_X Q((x_0, \gamma(x_0)), dx_1) F(x_1, \gamma(x_1)),\\
&+ \beta^2 \int_X Q((x_0, \gamma(x_0)), dx_1) \int_X Q((x_1, \gamma(x_1)), dx_2) v(x_2),\\
&= \dots\\
&= u_n(\gamma) + \beta^{n+1} \int_X Q((x_0, \gamma(x_0)), dx_1) ... \int_X Q((x_n, \gamma(x_n)), dx_{n+1}) v(x_{n+1}).
\end{aligned}$$

Taking the limit to infinity, the first term goes to $u_\infty(\gamma)$. So we only need to show that the second term goes to zero. The inner integral of this term is bounded by,

$$\begin{aligned}
&\int_X Q((x_n, \gamma(x_n)), dx_{n+1})v(x_{n+1}),\\
&\leq \|v\|_\phi \int_X Q((x_n, \gamma(x_n)), dx_{n+1})\phi(x_{n+1}),\\
&\leq \|v\|_\phi \frac{\theta}{\beta}\phi(x_n)
\end{aligned}$$

Iterating further over all other integrals finally gives that the term is bounded from above by:

$$\beta^{n+1}\|v\|_\phi \frac{\theta^{n+1}}{\beta^{n+1}}\phi(x_0)$$

This goes to zero as $n \to \infty$. □

For the fixed point v of the Bellman operator, define the policy function γ^* as,

$$\gamma^*(x) \in \arg\max_{a\in G(x)} \left\{ F(x,a) + \beta \int_X Q((x,a), d\tilde{x})v(\tilde{x}) \right\}.$$

This function is well defined.

Lemma 9. *Let (X, A, F, G, β, Q) be a regular problem, then $v(x_0) = u_\infty(\gamma^*)$.*

Proof. We have that,

$$\begin{aligned}
v(x_0) &= F(x_0, \gamma^*(x_0)) + \beta \int_X Q((x_0, \gamma^*(x_0)), dx_1)v(x_1),\\
&= F(x_0, \gamma^*(x_0)) + \beta \int_X Q((x_0, \gamma^*(x_0)), dx_1)F(x_1, \gamma^*(x_1)),\\
&+ \beta^2 \int_X Q((x_0, g^*(x_0)), dx_1) \int_X Q(dx_2, (x_1, \gamma^*(x_1)))v(x_2),\\
&= \ldots,\\
&= u_n(\gamma^*) + \beta^{n+1} \int_X Q((x_0, g^*(x_0)), dx_1) ... \int_X Q((x_n, \gamma^*(x_n)), dx_{n+1})v(x_{n+1}).
\end{aligned}$$

Taking the limit to infinity, the first term goes to $u_\infty(\gamma^*)$. So we only need to show that the second term goes to zero. However, the inner integral is bounded by,

$$\begin{aligned}
&\int_X Q((x_n, \gamma^*(x_n)), dx_{n+1})v(x_{n+1}),\\
&\leq \|v\|_\phi \int_X Q((x_n, \gamma^*(x_n)), dx_{n+1})\phi(x_{n+1}),\\
&\leq \|v\|_\phi \frac{\theta}{\beta}\phi(x_n)
\end{aligned}$$

Iterating further over all other integrations gives finally, that the term is bounded from above by,

$$\|v\|_\phi \theta^{n+1} \phi(x_0)$$

This goes to zero as $n \to \infty$. $\square$

Simulations for models of uncertainty

Let us first look at a very simple model of optimal growth with stochastic shocks. We take the utility of the consumer to be $u(c) = \ln(c+1)$. Output is produced using outputs in the previous period net of consumption. In particular, the output in period $t+1$ is given by,

$$y_{t+1} = \eta(y_t - c_t)^\alpha,$$

where η is a stochastic (random) shock with distribution P, realized in period $t+1$. We assume that the values of η are i.i.d. over time. The maximization problem reads:

$$\begin{aligned} &\max \sum_{t=0}^{\infty} \mathbb{E}\left(\beta^t \ln(c_t+1)|x_0\right), \\ \text{s.t. } &c_t \leq y_t, \\ &y_{t+1} = \eta(y_t - c_t)^\alpha. \end{aligned}$$

This gives rise to the Bellman equation:

$$v(y) = \max_{c \leq y} \left(\ln(c+1) + \beta \int_{\mathbb{R}} P(d\eta) v(\eta(y-c)^\alpha) \right).$$

This corresponds to a specification where $Q((x,a), d\tilde{x}) = P(d\eta)$. Simulation of this model is analogues as for the case under certainty. The only difference is here to estimate the integral. This can be done using Monte-Carlo simulation.

- Draw a large number of random variables $\eta_1, \ldots, \eta_N$ according to the distribution P.

- Compute the mean,

  $$\frac{1}{N} \sum_{n=1}^{N} v(\eta_n(y_t - c_t)^\alpha).$$

  Here $v(\eta_n(y_t - c_t)^\alpha)$ should be computed by interpolation.

For the algorithm, it is important to draw the values of $\eta_1, \eta_2, \ldots$ before entering the loop on the function value iteration. This guarantees the convergence of the algorithm. If you draw for each loop new random values, convergence is not guaranteed.

Try to code this problem using Howard improvement assuming that $\eta = e^{\mu + s\varepsilon}$ where ε has a standard normal distribution. You can use the parameters

$$\alpha = 0.4, \qquad \beta = 0.96, \qquad \mu = 0, \qquad s = 0.1$$

Take a grid size of 100 and let the values of y be equally spaced between 0 and 7. Compute the mean based on a sample of 1000 draws of η.

The previous example was rather easy in the sense that the value function (and policy function) where independent of the stochastic component. In particular P did not depend on the state or the action taken. In more interesting examples, however, this is no longer the case. Let's consider a growth model with a representative consumer with utility function $u(c) = \frac{c^{1-\sigma}-1}{1-\sigma}$. Capital is accumulated according to the law of motion:

$$k_{t+1} = e^s k_t^\alpha - c_t + (1-\delta)k_t.$$

Here s is a random variable that takes on two possible values s_1 and s_2. If the state is s_1, then output is multiplied by e^{s_1} if the state is s_2, output is multiplied by s_2. The transition probability between the states over time is determined by a Markov transition matrix:

$$\Pi = \begin{bmatrix} \pi_1 & 1-\pi_1 \\ 1-\pi_2 & \pi_2 \end{bmatrix}.$$

Here π_i is the probability of being in a_i next period, given that a_i is the current state. The optimization problem is then,

$$\begin{aligned} &\max \sum_{t=0}^{\infty} \beta^t \mathbb{E}\left(\frac{c_t^{1-\sigma}-1}{1-\sigma} \middle| k_0 \right), \\ \text{s.t. } & k_{t+1} = e^s k_t^\alpha - c_t + (1-\delta)k_t, \\ & \Pr(s_i|s_i) = \pi_i. \end{aligned}$$

A state is now given by a combination of a level of capital k and the value of the shock s. In terms of the Bellman equation, we have:

$$v(k,s) = \max_{c \le e^s k^\alpha + (1-\delta)k} \left\{ \frac{c^{1-\sigma}-1}{1-\sigma} + \beta \sum_{i=1}^{2} \Pr(s_i|s) v(e^{s_i}k^\alpha - c + (1-\delta)k, s_i) \right\}.$$

The value function $v(k,s)$ now depends on two variables, the capital stock k and the shock s. Likewise, the policy function will now be a function that takes a value for k and s and gives a level of consumption, $g(k, s)$. Given that we have two levels for the shocks, we have two functions $v(k, s_1)$ and $v(k, s_2)$ and two policy functions $g(k, s_1)$ and $g(k, s_2)$.

In this sense, we can encode the value function v and policy function g as an $N \times 2$ dimensional vector.

Try to code this problem using a Howard improvement with parameter values,

$$\sigma = 1.5, \qquad \delta = 0.1, \qquad \beta = 0.95, \qquad \alpha = 0.3,$$
$$s_1 = 0.8, \qquad s_2 = 1.2, \qquad \pi_1 = \pi_2 = 0.9,$$

and a grid size of 1000 where k is equally spaced between 0.2 and 6.

Try to use the output of the program to simulate trajectories of the capital stock and trajectories of consumption paths over time.

Applications

Consider the problem of a cake of size x that has to be eaten in its entirety in one single period. There is a taste shock z that takes on two possible values $0 < z_\ell < z_h$. Let p_ℓ be the probability of the taste next period equals z_ℓ when it is z_ℓ today and let p_h be the probability that the taste tomorrow is z_h given that it is z_h today. As such:

$$\begin{aligned}
\Pr(z_\ell|z_\ell) &= p_\ell, \\
\Pr(z_h|z_\ell) &= 1 - p_\ell, \\
\Pr(z_h|z_h) &= p_h, \\
\Pr(z_\ell|z_h) &= 1 - p_h.
\end{aligned}$$

Eating the cake of size x gives a value of $zu(x)$ where z is either equal to z_h or z_ℓ, depending on the taste value and $u(x) > 0$ is assumed to be strictly increasing.

To add an interesting twist, assume that if the cake is not eaten today, then a fraction $(1 - \delta)$ of the cake is lost. The state of the system depends on the size of the cake and the value of the taste shock. The Bellman equation takes the following expression:

$$\begin{aligned}
v(x, z_\ell) &= \max\{z_\ell u(x); \beta[p_\ell v(\delta x, z_\ell) + (1 - p_\ell)v(\delta x, z_h)]\}, \\
v(x, z_h) &= \max\{z_h u(x); \beta[p_h v(\delta x, z_h) + (1 - p_h)v(\delta x, z_\ell)]\}.
\end{aligned}$$

If the maximum on the right hand side is attained for the first component, then the cake will be eaten. Else, the decision maker will wait one more period. It is easy to see that the function $v(x, z)$ should be non-decreasing in x.

Lemma 10. *The fixed point of the Bellman equation satisfies that for all sizes x, $v(x, z_h) \geq v(x, z_\ell)$ and $z_h u(x) \geq \beta v(\delta x, z_h)$. As a result, the cake is always eaten if $z = z_h$.*

Proof. Let,

$$D = \{v : \text{ for all } x, v(x, z_h) \geq v(x, z_\ell) \text{ and } z_h u(x) \geq \beta v(\delta x, z_h)\}.$$

Notice that D is a closed set, so we only need to show that $T(D) \subseteq D$. Let $v \in D$. We need to show that $Tv \in D$. If the state is z_ℓ, there are two possibilities if $(Tv)(x, z_\ell) = z_\ell u(x)$, we have:

$$(Tv)(x, z_\ell) = z_\ell u(x) < z_h u(x) \leq (Tv)(x, z_h).$$

Next, let $(Tv)(x, z_\ell) = \beta[p_\ell v(\delta x, z_\ell) + (1 - p_\ell) v(\delta x, z_h)]$. Then:

$$\begin{aligned}(Tv)(x, z_\ell) &= \beta p_\ell v(\delta x, z_\ell) + \beta(1 - p_\ell) v(\delta x, z_h), \\ &\leq \beta v(\delta x, z_h) \leq z_h u(x) \leq (Tv)(x, z_h).\end{aligned}$$

The second and third inequality follows from the assumption that $v \in D$. This shows the first part of the proof.

Next, we need to show that $z_h u(x) \geq \beta(Tv)(\delta x, z_h)$.

$$\begin{aligned}\beta(Tv)(\delta x, z_h) &= \beta \max\{z_h u(\delta x), \beta[p_h v(\delta^2 x, z_h) + (1 - p_h) v(\delta^2 x, z_\ell)]\}, \\ &\leq \beta \max\{z_h u(\delta x), \beta v(\delta^2 x, z_h)\}, \\ &\leq \beta \max\{z_h u(\delta x), z_h u(\delta x)\} = \beta z_h u(\delta x) \leq z_h u(x).\end{aligned}$$

The second line uses $v \in D$ which implies $v(\delta^2 x, z_\ell) \leq v(\delta^2, z_h)$. The third line again uses $v \in D$ which implies $\beta v(\delta^2, z_h) \leq z_u u(\delta x)$. The last inequality follows from $\beta, \delta < 1$.

Given that the fixed point, v, is in D, we conclude that:

$$\begin{aligned}z_h u(x) &\geq \beta v(\delta x, z_h), \\ &\geq \beta[p_h v(\delta x, z_h) + \beta(1 - p_h) v(\delta x, z_\ell)].\end{aligned}$$

As such, $v(x, z_h) = z_h u(x)$ which means that the cake will be eaten if $z = z_h$. □

Given this result, we can omit the Bellman equation for the state z_h and focus only on the Bellman equation in case the state is z_ℓ. For this, however, we can now use the fact that $v(\delta x, z_h) = z_h u(\delta x)$:

$$v(x, z_\ell) = \max\{z_\ell u(x), \beta[p_\ell v(\delta x, z_\ell) + (1 - p_\ell) z_h u(\delta x)]\}.$$

Now, we would like to determine what happens if $z = z_\ell$. In this state, the decision maker faces a trade off between consuming now immediately and getting $z_\ell u(x)$ or waiting one period and hoping that the state changes to $z = z_h$, in which case, she will eat the cake.

Consider the following decision rule for some x^*: If $z = z_\ell$ and $x \leq x^*$, eat the cake. If $z = z_\ell$ and $x > x^*$, don't eat the cake. We would like to know when this is an optimal strategy.

Assume that we are at x^*. In this case, the decision maker should be indifferent between eating and not eating. As such:

$$z_\ell u(x^*) = \beta[p_\ell v(\delta x^*, z_\ell) + (1 - p_\ell) z_h u(\delta x^*, z_h)]$$

Also $\delta x^* < x^*$, so it is optimal to eat at state $(\delta x^*, z_\ell)$ which gives $v(\delta x^*, z_\ell) = z_\ell u(\delta x)$. This gives:

$$\begin{aligned} z_\ell u(x^*) &= \beta[p_\ell z_\ell u(\delta x^*) + (1-p_\ell) z_h u(\delta x, z_h)], \\ &= \beta(p_\ell z_\ell + (1-p_\ell) z_h) u(\delta x^*), \\ \leftrightarrow \frac{u(\delta x^*)}{u(x^*)} &= \frac{z_\ell}{\beta(p_\ell z_\ell + (1-p_\ell) z_h)}. \end{aligned}$$

Lemma 11. *Assume that there is an x^* such that:*

$$x \leq x^* \leftrightarrow \frac{z_\ell}{\beta\left(z_\ell p_\ell + z_h(1-p_\ell)\right)} \geq \frac{u(\delta x)}{u(x)}.$$

Then it is optimal to eat at $z = z_\ell$ if and only if $x \leq x^*$.

Proof. Let D be the set such that:

$$D = \{v : \forall x \leq x^*, v(x, z_\ell) = z_\ell u(x)\}.$$

Let us show that $T(D) \subseteq D$. We have that for $x \leq x^*$:

$$\begin{aligned} (Tv)(x, z_\ell) &= \max\{z_\ell u(x), \beta[p_\ell v(\delta x, z_\ell) + (1-p_\ell) z_h u(\delta x)]\}, \\ &= \max\{z_\ell u(x), \beta[p_\ell z_\ell u(\delta x) + (1-p_\ell) z_h u(\delta x)]\}, \\ &\leq \max\{z_\ell u(x), z_\ell u(x)\} = z_\ell u(x). \end{aligned}$$

The second line uses the fact that $\delta x \leq x^*$ so $v(\delta x, z_\ell) = z_\ell u(\delta x)$. Together with $(Tv)(x, z_\ell) \geq z_\ell u(x)$, we obtain that $(Tv)(x, z_\ell) = z_\ell u(x)$ as was to be shown.

Next, let $\tilde{D}$ be the set such that:

$$\tilde{D} = \{v : \forall x > x^*, v(x, z_\ell) > z_\ell u(x)\}$$

Let us show that $T(D) \subseteq \tilde{D}$. Let $x > x^*$. Then:

$$\begin{aligned} (Tv)(x, z_\ell) &= \max\{z_\ell u(x), \beta[p_\ell v(\delta x, z_\ell) + (1-p_\ell) z_h u(\delta x)]\}, \\ &\geq \max\{z_\ell u(x), \beta[p_\ell z_\ell u(\delta x) + (1-p_\ell) z_h u(\delta x)]\}, \\ &= \beta[p_\ell z_\ell u(\delta x) + (1-p_\ell) z_h u(\delta x)] > z_\ell u(x). \end{aligned}$$

The first inequality follows from the fact that $v(\delta x, z_\ell) \geq z_\ell u(\delta x)$. The equality and last strict inequality follow from the assumption. □

Optimal stopping problems

Optimal stopping problems are a special class of problems in where the discrete choice is a single decision to put an end to an ongoing problem.[58]

As a first example, consider a burglar who loots a house every day. The daily gains are independent and identically distributed on $\mathbb{R}_+$.

[58] For example, a student has to decide when to give up trying to solve a homework problem. A firms decides when to leave an industry, a firm decides when to stop working on the development of a new product or an unemployed worker has to decide when to accept a job from a sequence of offers.

With a certain probability $1 - p \in (0, 1)$, the burglar is caught and all her fortune is gone. The utility function of the burglar (with fortune x) is given by $1 - e^{-\alpha x}$.

The Bellman equation takes the form:

$$v(x) = \max\left\{1 - e^{-\alpha x}, \beta p \int_{\mathbb{R}_+} v(x+g)P(dg)\right\}$$

where P is the distribution of gains.

We will assume that there is an x^* such that the Burglar stops when $x \geq x^*$ and continues when $x < x^*$. In this case, she should be indifferent between stopping and continuing at x^*. This gives:

$$\begin{aligned}
1 - e^{-\alpha x^*} &= \beta p \int_{\mathbb{R}_+} v(x^* + g)P(dg), \\
&= \beta p \int_{\mathbb{R}_+} (1 - e^{-\alpha(x^*+g)})P(dg), \\
&= \beta p \left(1 - e^{-\alpha x^*} \int_{\mathbb{R}_+} e^{-\alpha g} P(dg)\right), \\
&= \beta p (1 - Re^{-\alpha x^*})
\end{aligned}$$

where $R = \int e^{-\alpha g} P(dg) \leq 1$ is a fixed number. So:

$$(1 - \beta p) = (1 - \beta p R)e^{-\alpha x^*}.$$

The right hand side is decreasing in x^*. So there will be a unique value of x^* that satisfies this equation.

Lemma 12. *Let x^* be the value that satisfies*

$$(1 - \beta p) = (1 - \beta p R)e^{-\alpha x^*},$$

then it is optimal to stop if and only if $x \geq x^*$.

Proof. Let $D = \{v : \text{ if } x \geq x^*, \text{ then } v(x) = 1 - e^{-\alpha x}\}$. As usual, we will show that for all $v \in D$, $Tv \in D$.

Let $v \in D$ and assume that $x \geq x^*$. Then:

$$\begin{aligned}
(Tv)(x) &= \max\{1 - e^{-\alpha x}, \beta p \int_{\mathbb{R}_+} v(x+g)P(dg)\}, \\
&= \max\{1 - e^{-\alpha x}, \beta p \int_{\mathbb{R}_+} (1 - e^{-\alpha(x+g)})Pdg\}, \\
&= \max\{1 - e^{-\alpha x}, \beta p - e^{-\alpha x}\beta p R\} = 1 - e^{-\alpha x},
\end{aligned}$$

The second equality uses the fact that $x + g \geq x \geq x^*$ and $v \in D$.

Next let

$$D = \{v : \text{ if } x \geq x^* \text{ then } v(x) = 1 - e^{-\alpha x} \text{ and if } x \leq x^*, \text{ then } v(x) \geq 1 - e^{-\alpha x}\}.$$

and let

$\tilde{D} = \{v: \text{ if } x \geq x^* \text{ then } v(x) = 1 - e^{-\alpha x} \text{ and if } x < x^*, \text{ then } v(x) > 1 - e^{-\alpha x}\}.$

Let us show that $T(D) \subseteq \tilde{D}$. Let $v \in D$ and assume that $x < x^*$. Then:

$$\begin{aligned}(Tv)(x) &= \max\{1 - e^{-\alpha x}, \beta p \int_{\mathbb{R}_+} v(x+g) P(dg)\},\\ &\geq \max\{1 - e^{-\alpha x}, \beta p \int_{\mathbb{R}_+} (1 - e^{-\alpha(x+g)}) P dg\},\\ &= \max\{1 - e^{-\alpha x}, \beta p - e^{-\alpha x} \beta p R\} > 1 - e^{-\alpha x},\end{aligned}$$

□

Consider an agent that visits stores at a rate of one per period. Then given that the price quoted in the current period is p, the individual can choose to stop now and purchase the good or go to the next store. If he stops, he gets $u - p$ where u is the value of the good bought. If he continuous, he enters the next period as an active searcher and faces an additional search cost of c. The Bellman equation is,

$$v(p) = \max\{u - p; -\beta c + \beta \int_0^\infty v(\tilde{p}) F(d\tilde{p})\}.$$

Observe that the second term $-\beta c + \beta \int_0^\infty v(\tilde{p}) F(d\tilde{p}) \equiv \bar{v}$ is independent of the current price p as we assumed that prices are i.i.d. drawn. The first term is declining in p so there is a unique value p^* where $u - p^* = -\beta c + \beta \int_0^\infty v(\tilde{p}) F(d\tilde{p})$. From this, it follows that $\bar{v} = u - p^*$.

Any price greater than p^* induces further search while any value below p^* lets the agent buy the good. We have that,

$$\begin{aligned}u - p^* &= -\beta c + \beta \int_0^{p^*} v(\tilde{p}) F(d\tilde{p}) + \beta \int_{p^*}^\infty v(\tilde{p}) F(d\tilde{p}),\\ &= -\beta c + \beta \int_0^{p^*} (u - \tilde{p}) F(d\tilde{p}) + \beta \int_{p^*}^\infty (u - p^*) F(d\tilde{p}),\\ &= -\beta c + \beta(u - p^*) + \beta \int_0^{p^*} (p^* - \tilde{p}) F(d\tilde{p}).\end{aligned}$$

So,

$$\begin{aligned}(u - p^*)(1 - \beta) &= -\beta c + \beta \int_0^{p^*} (p^* - \tilde{p}) F(d\tilde{p}),\\ \rightarrow p^* &= u + \frac{\beta}{1 - \beta} \left[c - \int_0^{p^*} (p^* - \tilde{p}) F(d\tilde{p}) \right].\end{aligned}$$

This is the fundamental reservation price equation of the problem. The first term gives the immediate benefit of purchasing. The second term gives the option value (cost) of waiting.

Finite horizon dynamic optimization

A FINITE HORIZON dynamic programming problem differs from the infinite version in the sense that the horizon over which the problem is considered is finite. In this chapter, we will demonstrate the power of dynamic programming by means of various examples.

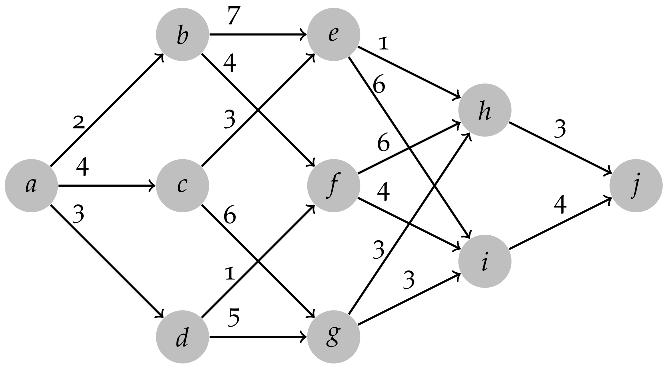
The most important feature of a dynamic programming problem is that it expresses the solution of a problem in terms of the solutions of smaller sub-problems. As these subproblems are themselves defined into smaller subproblems, the original problem can be solved recursively: first solve the smallest subproblems and use their solutions to solve the bigger problems.

The way a problem is defined in terms of smaller subproblems is done using the Bellman equation of the problem.

Shortest path problem

A shortest path problem is defined in terms of a network. This network consists of a number of nodes and directed edges between the nodes. In the easiest setting, the network contains no cycles. There are two special nodes, called the source or starting node and the sink or final node. We try to find the shortest path from the starting node to the final node.

As an example, consider the network below. Here a is the starting node and j is the end node. Connections between node have a certain cost. For example, to go from a to c, it costs 4. The problem that we try to solve is how to go from a to j with the lowest amount of total cost. For example, the path $a - b - e - h - j$ has a total cost of $2 + 7 + 1 + 3 = 13$. The path $a - c - g - i - j$ has a total cost of $4 + 6 + 3 + 4 = 17$ and so on.



One way to solve this problem would be to apply a brute force method: enumerate all possible paths, compute for each one its total cost and look for the path that minimizes the cost. For our particular example, there are a total of 12 paths, so this is a not too difficult problem. However, if we consider networks of larger size, the number of paths will be come exponentially big, so this brute force approach is no longer efficient.

An alternative approach is to use dynamic programming. Let x be a node in the network and let us denote by $v(x)$ the shortest distance from x to the final node j. Notice here that we look at the subproblem that starts at x (instead of at the starting node a) and tries to find the minimal path from x to j.

Let us denote by $s(x)$ the set of all nodes that can be reached from x in one step. For example, for node b: $s(b) = \{e, f\}$. Now, the optimal path from x to j also has to pass through one of the nodes in $s(x)$, say $y \in s(x)$. If so, then the shortest path from x to j can be written as the sum of the cost of going from x to y, say $c(x, y)$ plus the shortest distance from y to j:

$$v(x) = c(x, y) + v(y).$$

Of course, we don't know ex-ante whether the shortest path from x to j really goes through the node $y \in s(x)$. However, it must go through one of the nodes in $s(x)$. Moreover, given minimality of $v(x)$ we have that:

$$v(x) = \min_{y \in s(x)} \{c(x, y) + v(y)\}.$$

This is the Bellman equation for the problem. This equation states that the minimal cost of going from x to j can be found by finding the successor $y \in s(x)$ that minimizes the cost of going from x to y, i.e. $c(x, y)$ plus the minimal cost of going from y to j. Notice that if $\{j\} = s(x)$, i.e. the only successor of x is j, then:

$$v(x) = c(x, j).$$

So we know the value of $v(x)$ for all nodes whose only successor is j. These give us the base cases we need to solve the Bellman equation. Let's apply this principle to our example network. For the base case, there are two nodes that have j as their successor: h and i. We find:

$$v(h) = 3 \qquad v(i) = 4.$$

Next we can solve the minimal path for e:

$$v(e) = \min\{1 + v(h), 6 + v(i)\} = \min\{1 + 3, 6 + 4\} = 4,$$

We find that $v(e) = 4$ which is obtained via the path $e - h - j$. Similarly, we can solve the minimal paths for f and g:

$$v(f) = \min\{6 + v(h), 4 + v(i)\} = \min\{6 + 3, 4 + 4\} = 8 \to f - i - j,$$
$$v(g) = \min\{3 + v(h), 3 + v(i)\} = \min\{3 + 3, 3 + 4\} = 6 \to g - h - j.$$

Given these solutions, we can subsequently find the values of $v(b), v(c)$ and $v(d)$:

$$v(b) = \min\{7 + v(e), 4 + v(f)\} = \min\{7 + 4, 4 + 8\} = 11 \to b - e - h - j,$$
$$v(c) = \min\{3 + v(e), 6 + v(g)\} = \min\{3 + 4, 6 + 6\} = 7 \to c - e - h - j,$$
$$v(d) = \min\{1 + v(f), 5 + v(g)\} = \min\{1 + 8, 5 + 6\} = 9 \to d - g - h - j.$$

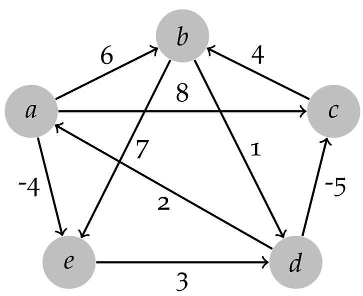
Finally, we can find the value of $v(a)$:

$$\begin{aligned} v(a) &= \min\{2 + v(b), 4 + v(c), 3 + v(d)\}, \\ &= \min\{2 + 11, 4 + 7, 3 + 9\} = 11 \to a - c - e - h - j. \end{aligned}$$

The advantage of the dynamic optimization formulation is that we do not need to go over all paths, but only solve one problem for each vertex. By first finding the optimal paths from all intermediary nodes, the procedure exclude paths that will never be taken early onwards and it thereby severely restricts the space of paths we need to consider.

Networks with cycles

Let us have a look at a slight generalization of the previous minimal cost problem. Consider the following network:



This network differs from the previous one in the sense that now, we allow for cycles in the graph. Again, similar to before, we would like to find out the shortest path between any two nodes in the network. For example, the direct connection from a to b has a cost of 6. However, there is an alternative path $a - e - d - c - b$ whose total cost equals -2 which is smaller.

Let $v(x, y)$ be the value of the shortest path from node x to node y in the network. We would like to write this function as a recursive problem. In other words, can we express the value $v(x, y)$ in terms of the values of some sub-problem?

Notice that in this case, we can not simply write it in terms of subproblems of the successors of x as this recursion will never give us any base problems whose value we already know.

As such, we need to find a different structure. Let S be a subset of all nodes and let us write by $v(x, y, S)$ the cost of the shortest path from x to y that only takes paths whose intermediate nodes are contained in S. For example, $v(a, b, \emptyset) = 6$ as the cost of going from a to b without involving any intermediary nodes is given by the direct edge with cost 6. However, $v(a, b, \{e, d, c\}) = -2$ as this allows the path $a - e - d - c - b$.

Now, consider a minimal path from x to y whose intermediate nodes are in some subset S of the total set of nodes; with total cost $v(x, y, S)$. Then for some $z \in S$, there are two options. First, it is possible that the path does not contain z. In this case, we have that:

$$v(x, y, S) = v(x, y, S - \{z\}).$$

As the minimal path from x to y through S is the same as the path from x to y through $S - \{z\}$. Second, it is possible that the minimal path from x to y in S does contains the node z. For such instances, it is possible to break this minimal path in two sub-paths: a first sub-path goes from x to z. The second sub-path goes from z to y. As the path is minimal, we have that in this case

$$v(x, y, S) = v(x, z, S - \{z\}) + v(z, y, S - \{z\}).$$

As $v(x, y, S)$ is the minimal path, it must be that:

$$v(x, y, S) = \min \left\{ \begin{array}{l} v(x, y, S - \{z\}), \\ v(x, z, S - \{z\}) + v(z, y, S - \{z\}) \end{array} \right\}$$

This allows us to compute $v(x, y, S)$ recursively as it expresses it in terms of smaller subproblems in terms of the set S.

Let us see how this algorithm works. First we need a list of nexted sets, say $\emptyset, \{a\}, \{a, b\}, \{a, b, c\}, \{a, b, c, d\}, \{a, b, c, d, e\}$.

Let us begin by the base case, where $S = \emptyset$, giving the values $v(x, y, \emptyset)$. This is easy. The shortest path from x to y that has no intermediary nodes is simply the path $x - y$. As such, $v(x,y) = c(x,y)$ where $c(x,y)$ is the cost of the edge from x to y. Let us put these in a table where in row x and column y, we put $c(x,y)$ and we enter ∞ if there is no direct path from x to y (if there is no path from x to y its cost is ∞).

	a	b	c	d	e
a	0	6	8	∞	-4
b	∞	0	∞	1	7
c	∞	4	0	∞	∞
d	2	∞	-5	0	∞
e	∞	∞	∞	3	0

For the next step, we look at the set $S = \{a\}$. Then for all x, y, we have:

$$v(x,y,\{a\}) = \min\{v(x,y,\emptyset), v(x,a,\emptyset) + v(a,y,\emptyset)\} = \min\{c(x,y), c(x,a) + c(a,y)\}.$$

Applying this gives the following table:

	a	b	c	d	e
a	0	6	8	∞	-4
b	∞	0	∞	1	7
c	∞	4	0	∞	∞
d	2	8	-5	0	-2
e	∞	∞	∞	3	0

The numbers in red shows the modifications. For example, we have that in row d and column b:

$$\begin{aligned} v(d,b,\{a\}) &= \min\{v(d,b,\emptyset), v(d,a,\emptyset) + v(a,b,\emptyset)\}, \\ &= \min\{\infty, c(d,a) + c(a,b)\} = 8. \end{aligned}$$

Next, we extend S to $\{a,b\}$. Then:

$$v(x,y,\{a,b\}) = \min\{v(x,y,\{a\}), v(x,b,\{a\}) + v(b,y,\{a\})\}.$$

This gives the following table:

	a	b	c	d	e
a	0	6	8	7	-4
b	∞	0	∞	1	7
c	∞	4	0	5	11
d	2	8	-5	0	-2
e	∞	∞	∞	3	0

Next, we expand to $S = \{a,b,c\}$ giving:

	a	b	c	d	e
a	0	6	8	7	-4
b	∞	0	∞	1	7
c	∞	4	0	5	11
d	2	-1	-5	0	-2
e	∞	∞	∞	3	0

Then $S = \{a, b, c, d\}$:

	a	b	c	d	e
a	0	6	2	7	-4
b	3	0	-4	1	-1
c	7	4	0	5	3
d	2	-1	-5	0	-2
e	5	2	-2	3	0

And finally for $S = \{a, b, c, d, e\}$:

	a	b	c	d	e
a	0	-2	-6	-1	-4
b	3	0	-4	1	-1
c	7	4	0	5	3
d	2	-1	-5	0	-2
e	5	2	-2	3	0

This shows that indeed, $v(a, b, S) = -2$. We can easily write the computation of $v(x, y, S)$ as the output of the following algorithm.[59]

[59] This algorithm is known as the Floyd-Warshall algorithm.

For i in $1 : N$

For x in $1 : N$

For y in $1 : N$

$v(x, y) = \min\{v(x, y), v(x, i) + v(i, y)\}$.

end

end

end

The outer loop iterates over the composition of the set $S = \{1, \ldots, i\}$. The second loop is over the starting nodes x in $v(x, y, S)$ and the final loop is over the end nodes y in $v(x, y, S)$. The algorithm has worst time complexity $\theta(n^3)$ where n is the number of nodes in the network.

Currency exchange

Consider a set $C = \{c_1, \ldots, c_N\}$ of coins of various values and let m be an amount of money. What is the minimal total number of coins we need in order to exactly pay the amount m.

As an example we can have $C = [1,2,5]$ so we have a coin of value 1, a coin of value 2 and a coin of value 5. Let $m = 11$. In this case, we can pay the amount m by using 1 piece of 5 and 3 pieces of 2, 4 coins. Alternatively, more efficiently is to use 2 piece of 5 and one piece of 1.

Let $v(m)$ be the minimal number of coins needed to pay an amount m. We would like to write $v(m)$ as a recursive problem.

Consider a coin in C with value c. If c is used to pay the amount m, then we can first pay the amount $m - c$ and then pay the remaining amount with one additional coin. As such:

$$v(m) = 1 + v(m - c).$$

however, we don't know if coin c is used in the payment of m. As such, we should take the minimum over all $c \in C$. this gives:

$$v(m) = \min_{c \in C}\{1 + v(m - c_i)\} \text{ subject to } c_i \leq m.$$

As a starting value we can set $v(0) = 0$, as we need zero coins to pay an amount of zero.

The table below gives the computation for the particular example. The first column gives the amounts m. The 2nd to fourth column gives the values for,

$$1 + v(m - c) \text{ when } c \leq m.$$

Finally, the last column gives the minimum over all coins and is therefore equal to $v(m)$.

For example, in row 6, we would like to determine $v(6)$. This is given by:

$$\min\{1 + v(5), 1 + v(4), 1 + v(1)\} = \min\{2, 3, 2\} = 2.$$

m	$c_1 = 1$	$c_2 = 2$	$c_3 = 5$	$v(m)$
0	NA	NA	NA	0
1	1	NA	NA	1
2	2	1	NA	1
3	2	2	NA	2
4	3	2	NA	2
5	3	3	1	1
6	2	3	2	2
7	3	2	2	2
8	3	3	3	3
9	4	3	3	3
10	4	4	2	2
11	3	4	3	3

Based on the last row, we see that $v(11) = 3$. Tracing back the minima, we see that this can be obtained by two pieces of 5 and one piece of 1.

Subset addition

Consider a set of strict positive numbers S and a number m. How many combinations of numbers in S have the property that they add up to the number m.

To take a concrete example, let $S = \{2, 4, 6, 8, 10\}$ and let $m = 16$. Then we have that $16 = 2 + 4 + 10$ but also $16 = 10 + 6$ and $16 = 6 + 2 + 8$. A careful check shows that there are no other subset of numbers that add up to 16, so here the answer is 3.

A brute force approach would be to look at all possible subsets of S and see which ones add up to m. If S contains N elements, this would require us to go over $N! = N(N-1)(N-2)\ldots 2.1$ subsets which is huge even for moderate N. As before, let us write this problem in a recursive form. Let $v(m, S)$ be the number of ways one can add together distinct numbers in S to get m. Let $x \in S$. Then any sum of numbers that add up to m either includes x or it does not include x. Consider first the ones that do not include x. For this, we have that there are $v(m, S - \{x\})$ ways to sum to m among all numbers in S excluding x. Next, for the sums that include x, there are a total of $v(m - x, S - \{x\})$ such combinations as including x to these combinations indeed adds up to m. As such:

$$v(m, S) = \begin{cases} v(m, S - \{x\}) & \text{if } x > m, \\ v(m, S - \{x\}) + v(m - x, S - \{x\}) & \text{if } x \leq m. \end{cases}$$

Here as starting values we should set $v(0, S) = 1$ as there is exactly one way to sum to zero. Next, we also set $v(m, \emptyset) = 0$ whenever $m > 0$ as there is no way to obtain $m > 0$ if there are no terms in S.

The table below gives the computations

m	$\emptyset$	$\{2\}$	$\{2,4\}$	$\{2,4,6\}$	$\{2,4,6,10\}$	$\{2,4,6,10,8\}$
0	1	1	1	1	1	1
1	0	0	0	0	0	0
2	0	1	1	1	1	1
3	0	0	0	0	0	0
4	0	0	1	1	1	1
5	0	0	0	0	0	0
6	0	0	1	2	2	2
7	0	0	0	0	0	0
8	0	0	0	1	1	2
9	0	0	0	0	0	0
10	0	0	0	1	2	3
11	0	0	0	0	0	0
12	0	0	0	1	2	3
13	0	0	0	0	0	0
14	0	0	0	0	1	3
15	0	0	0	0	0	0
16	0	0	0	0	2	3

As an example for how this table is filled, consider the element at row 12 and column $\{2,4,6,10\}$. We have that $10 \leq 12$ so,

$$v(12, \{2,4,6,10\}) = v(12, \{2,4,6\}) + v(2, \{2,4,6\}) = 1 + 1 = 2.$$

Knapsack problem

Consider a set of items S. Each item $i \in S$ has a weight w_i and a value x_i. You also have a knapsack to put items inside. The problem, however is that the total weight of the bag cannot exceed some threshold m. What items should you put in the bag in order to maximize the total value in the bag.

Let $v(m, S)$ be the maximal value of the bag with threshold m and items S. For an item i, it is either in the bag or not. If it is in the bag then the value is given by x_i plus the value of the bag without the item i. The value of the bag without the item should be equal to the maximal value of a knapsack with capacity $m - v_i$ and item set $S \setminus \{i\}$. As such:

$$v(m, S) = v(m - w_i, S - \{i\}).$$

If the item i is not in the optimal knapsack, then the maximal value is given by $v(m, S - \{i\})$. As such:

$$v(m, S) = v(m, S - \{i\}).$$

As the bag is optimal, the value of $v(m, S)$ should equal the maximum of these two.

$$v(m,S) = \begin{cases} \max \left\{ \begin{array}{l} v(m - w_i, S - \{i\}) + x_i, \\ v(m, S - \{i\}) \end{array} \right\} & \text{if } w_i \leq m, \\ v(m, S - \{i\}) & \text{if } w_i > m \end{cases}$$

The starting values are $v(0, S) = 0$ as a knapsack with capacity zero cannot have any value. Also, $v(m, \emptyset) = 0$ as there are no items to put into the bag in this case.

As an example consider a set S of 4 items with weights and values given as below:

item i	weight v_i	value x_i
1	1	1
2	3	4
3	4	5
4	5	7

Consider a bag of total weight 7. We can solve the problem by making a table as below. The table is filled in column by column. For example in column $\{1,2,3\}$ at row $m = 6$ one compares $v(6, \{1,2\})$ which is 5 with the value of $v(6-4, \{1,2\}) + x_3 = 1 + 5 = 6$ and choose the maximal.

m	$\emptyset$	$\{1\}$	$\{1,2\}$	$\{1,2,3\}$	$\{1,2,3,4\}$
0	0	0	0	0	0
1	0	1	1	1	1
2	0	1	1	1	1
3	0	1	4	4	4
4	0	1	5	5	5
5	0	1	5	5	7
6	0	1	5	6	8
7	0	1	5	9	9

We see that in this case, the bag has total value 9, obtained by including the items 3 and 2.

Longest common subsequence

Consider two sequence $s_1 = x_1 x_2 \ldots x_N$ and $s_2 = y_1 y_2 \ldots y_M$ of letters. What is the longest common subsequence of s_1 and s_2. For example if $s_1 = abcdaf$ and $s_2 = acbcf$ then the solution is given by $abcf$ which gives four.

As before, we try to model this as a recursive problem. Let us denote by $v(s_1, s_2)$ the length of the longest common subsequence. Let x_N and y_M be the last letter of the sequence s_1 and s_2. There are three cases, either $x_N = y_M$ are part of the common subsequence. In this case, we have that:

$$v(s_1, s_2) = v(s_1 - x_N, s_2 - y_M) + 1.$$

where we denote by $s_1 - x_N$ the sequence s_1 without its last letter and $s_2 - y_M$ the sequence s_2 without its last letter. Second, it is possible that x_N is not part of the subsequence, then:

$$v(s_1, s_2) = v(s_1 - x_N, s_2).$$

finally if y_M is not part of the subsequence, then:

$$v(s_1, s_2) = v(s_1, s_2 - y_M).$$

As such:

$$v(s_1, s_2) = \max \left\{ \begin{array}{l} v(s_1 - x_N, s_2 - y_M) + 1_{x_N = y_M}, \\ v(s_1 - x_N, s_2), \\ v(s_1, s_2 - y_M) \end{array} \right\}.$$

The starting values are $v(\emptyset, s_2) = v(s_1, \emptyset) = 0$. For the sequences $s_1 = abcdaf$ and $s_2 = acbcf$, the computation is given in the table below

	∅	a	b	c	d	a	f
∅	0	0	0	0	0	0	0
a	0	1	1	1	1	1	1
c	0	1	1	2	2	2	2
b	0	1	2	2	2	2	2
c	0	1	2	3	3	3	3
f	0	1	2	3	3	3	4

Ever cell is computed by taking the max of one plus the cell to the top left (in case the value in the row and column are equal) or taking the max of the value to the left or the top.

Efficient matrix multiplication

If you multiply a matrix of size $[n, m]$ with a matrix of size $[m, k]$ then the total number of operations is given by $n \times m \times k$: you need to compute a new matrix of size $n \times k$ and each time this is done by multiplying k numbers together and adding them up. As such, if you have to multiply a sequence of matrices together in the shortest amount of time, it does matter in which order you do the multiplication.

As an example, consider 4 matrices A_1, A_2, A_3, A_4 and assume that the sizes are given by:

Matrix	size $[r_i, c_i]$
A_1	[2,3]
A_2	[3,6]
A_3	[6,4]
A_4	[4,5]

Here we denote by r_i the number of rows of A_i and c_i the number of columns of A_i. Assume we need to take the product $A_1 \times A_2 \times A_3 \times A_4$.

Let us denote by $v(i,j)$ the cost of multiplying $A_i \times \ldots \times A_j$. Then for any $k \in \{i, i+1, \ldots, j-1\}$ we can first multiply $A_i \times \ldots \times A_k$, then $A_{k+1} \times \ldots \times A_j$ and finally we multiply the result of these two matrices together. (If $k = 1$ or $k = j-1$ the first or second product is simply the matrix A_i or A_j which has a cost of zero). As such, the minimal cost should equal:

$$v(i,j) \leq v(i,k) + v(k+1,j) + (r_i \times c_k \times c_j).$$

Notice that $A_i \times \ldots \times A_k$ has r_i rows and c_k columns and that the matrix $A_{k+1} \times \ldots \times A_j$ has $r_{k+1} = c_k$ rows and c_j columns. the first gives the cost of mulitplying the matrices A_i up to A_k together. The second the cost of multiplying A_{k+1} up to A_j together and finally the cost of multiplying the resulting two matrices together. Initial costs are obtained by setting $v(i,i) = 0$ for all i. The recursion is given by:

$$v(i,j) = \min_{k=i,i+1,\ldots,j-1} \{v(i,k) + v(k+1,j) + (r_i \times c_k \times c_j)\}.$$

For the example, we have that:

$$\begin{aligned}
v(1,2) &= 0 + 0 + 2 \times 3 \times 6 = 36, \\
v(2,3) &= 0 + 0 + 3 \times 6 \times 4 = 72, \\
v(3,4) &= 0 + 0 + 6 \times 4 \times 5 = 120.
\end{aligned}$$

Next:

$$\begin{aligned}
v(1,3) &= \min\{v(1,2) + v(3,3) + r_1 \times c_2 \times c_3, v(2,3) + v(1,1) + r_1 \times c_1 \times c_3\}, \\
&= \min\{36 + 2 \cdot 6 \cdot 4, 72 + 2 \cdot 3 \cdot 4\} = 84, \\
v(2,4) &= \min\{v(2,3) + v(4,4) + r_2 \times c_3 \times c_4, v(3,4) + v(2,2) + r_2 \times c_2 \times c_4, \}, \\
&= \min\{72 + 3 \cdot 4 \cdot 5, 120 + 3 \cdot 6 \cdot 5\} = 132.
\end{aligned}$$

Finally:

$$\begin{aligned}
v(1,4) &= \min\{v(1,1) + v(2,3) + r_1 \times c_1 \times c_4, v(1,2) + v(2,4) + r_1 \times c_2 \times c_4, v(1,3) + v(4,4) + r_1 \times c_3 \times c_4\}, \\
&= \min\{132 + 2 \cdot 3 \cdot 5, 36 + 120 + 2 \cdot 6 \cdot 5, 84 + 2 \cdot 4 \cdot 5\} = 124.
\end{aligned}$$

One can see that it is therefore optimal to first multiply A_1 and A_2 together (cost 36). Then this result should be multiplied by A_3 giving total cost of 84. Finally, this matrix is multiplied by A_4 giving a total cost of 124.

The egg drop problem

Assume that you have a number of eggs and an apartment building. You need to figure out the lowest floor from which you break an egg when it is dropped. What is the minimal amount of times you need to drop an egg in order to find this floor for all possible cases.

As an example, consider three floors and 1 egg. In this case, you first need to drop the egg from the first floor. If it breaks, your answer is 1. If not, you drop the egg from the second floor. If it breaks, your answer is 2. Finally, you drop the egg from the third floor. If it breaks, your answer is 3. If not, no floor is high enough to break an egg. As such, in order to figure out the answer to the egg-drop problem, you must have 3 tries at worst.

Notice that this cannot be improved upon. In particular, if you first drop the egg from the 2nd or 3rd floor, it might break immediately, in this case, you will never find out if the egg might have also broken from the first floor. In other words if you have only one egg and m floors, the largest number of tries is equal to m. So:

$$v(m,1) = m.$$

If you have more eggs, you can improve upon this by being a bit more aggressive. Assume, for example if you have two eggs and 3 floors. In this case, you might start to drop one egg from floor 2. If it breaks, you drop your second egg from the first floor and you find your answer in 2 tries. If the egg does not break from the second floor, you drop it again from the third floor to find you final answer. In this case, two drops is sufficient to find out the answer, i.e.

$$v(3,2) = 2 \leq v(3,1) = 3.$$

Let us try to find the recursion. Assume that you have m floors and e eggs. If you drop an egg from floor i two things can happen. Either it breaks. In this case, you can restrict yourself to floors $1, \ldots, i-1$ and have $e-1$ eggs left, so $v(i-1, e-1)$ tries are remaining at most.

If the egg does not break, you can restrict yourself to floors $i + 1, \ldots, m$ with a total of e eggs, so at most $v(m-i, e)$ tries remain. This gives that:

$$v(m,e) \leq 1 + \max\{v(i-1, e-1), v(m-i, e)\}.$$

you will choose the floor that minimizes the right hand side. So:

$$v(m,e) = 1 + \min_{i \leq m} \{\max\{v(i-1, e-1), v(m-i, e)\}\}.$$

The default values are given by $v(1,e) = 1$ and $v(m,1) = m$.

As an example, assume we hat 2 eggs and 6 floors. First we have:

$$v(1,1) = v(1,2) = v(1,3) = 1,$$

and

$$v(2,1) = 2; v(3,1) = 3, v(4,1) = 4, v(5,1) = 5, v(6,1) = 6.$$

Next, consider $v(2,2)$. Then if we drop the egg from the first floor we have:

$$1 + \max\{v(0,1), v(1,2)\} = 2.$$

If we drop it from the second floor we have:

$$1 + \max\{v(1,1), v(0,2)\} = 2.$$

This gives $v(2,2) = 2$. Next, let's have a look at $v(3,2)$. If we drop the egg from the first floor, we obtain:

$$1 + \max\{v(0,1), v(2,2)\} = 3,$$

If we drop it from the second floor, we get:

$$1 + \max\{v(1,1), v(1,2)\} = 2.$$

If we drop it from the third floor, we have:

$$1 + \max\{v(2,1), v(1,2)\} = 3.$$

As such, $v(3,2) = 2$. Now consider $v(4,2)$ going over the different cases gives:

$$\begin{aligned}
1 + \max\{v(0,1), v(3,2)\} &= 3,\\
1 + \max\{v(1,1), v(2,2)\} &= 3,\\
1 + \max\{v(2,1), v(1,2)\} &= 3,\\
1 + \max\{v(3,1), v(0,2)\} &= 4.
\end{aligned}$$

As such, $v(4,2) = 3$. Next consider 5 floors. A similar calculation gives that $v(5,2)$ is obtained as the minimum of:

$$\begin{aligned}
1 + \max\{v(0,1), v(4,2)\} &= 4,\\
1 + \max\{v(1,1), v(3,2)\} &= 3,\\
1 + \max\{v(2,1), v(2,2)\} &= 3,\\
1 + \max\{v(3,1), v(1,2)\} &= 4,\\
1 + \max\{v(4,1), v(0,2)\} &= 5.
\end{aligned}$$

We see that $v(5,2) = 3$. Finally for $v(6,2)$ we have that it equals the minimum of:

$$\begin{aligned}
&1 + \max\{v(0,1), v(5,2)\} = 4,\\
&1 + \max\{v(1,1), v(4,2)\} = 4,\\
&1 + \max\{v(2,1), v(3,2)\} = 3,\\
&1 + \max\{v(3,1), v(2,2)\} = 4,\\
&1 + \max\{v(4,1), v(1,2)\} = 5,\\
&1 + \max\{v(5,1), v(0,2)\} = 6.
\end{aligned}$$