1 Dynamic optimisation when the future is unpredictable: backward inducation and dynamic programming

1.1 Introduction

A major problem in natural resource management is that a lot of factors are unknown. For instance, we do not know how much fossil fuel is available exactly, what the quality of the resources is, and how easy or difficult it is to extract those resources. Water management depends strongly on weather conditions. Growth of renewable resources, such as fish and game, depends on many factors, some of which are unobservable.

The economic literature distinguishes 'risk' from uncertainty. The term 'risk' is generally reserved for situations where the exact value of one or more variables is unknown, but we do have a probability distribution. This allows us to make statements like "the expected value of x is 2" or "the probability that $y > 10$ is 0.2". The term 'uncertainty' refers to a situation where not only the value of a variable is unknown, but also its distribution. At best we could indicate a range ("x is somewhere between 2 and 5"), but in many cases we cannot even do that.

If you are analyzing a problem with some missing information, you could of course always do a thorough sensitivity analysis. You could run your model to evaluate different exploitation rates under several assumptions with regard to the unknown variable, and choose the exploitation rate with the highest expected utility. In many examples of natural resource management, however, some of the information gaps are resolved along the way. For instance, if you know the current size of fish populations it is difficult enough to predict next years stocks because of many different stochastic factors that play a role. It is even more difficult to predict the stocks in two years time, because they depend on two growth seasons instead of one, and in each growth season the stochastic factors may be different. However, you will be able to measure next years stocks, which will help to make your predictions more accurate. The trick is to take into account not only the unknown factors, but also the possibility to change your plans when you have new information.

In this document I will concentrate on such situations where unknown information is gradually disclosed, and we have, or can at least assume, a probability distribution. The reason is very practically that this is the situation where we still have quite a number of analytical tools at our disposal, and it is also the most interesting case. If we do not have probability distributions we can only resort to very crude strategies. If our unknowns are not resolved over time there is no new information to act on and we have to resort to sensitivity analysis.

1.2 Backward induction

1.2.1 Backward induction in a deterministic model

Suppose you have a tank of water that you use to irrigate your crops. Let X_t be the stock of water in time t, and C_t be your water consumption. For simplicity we assume the stock of water can have only four possible values, so that you have either no water, 1 unit, 2 units, or 3 units. This also implies that your water consumption can be either 0, 1, 2, or 3 units. Finally, we assume your utility from water consumption in each period is $\sqrt{C_t}$, and a discount rate of 10%. Figure 1 shows all possible consumption paths from an initial stock of 3 units.

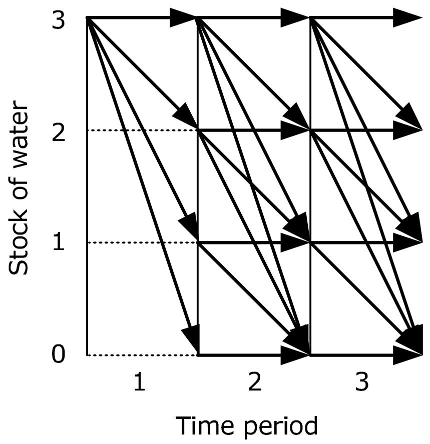

Figure 1: Possible consumption paths from a starting stock of 3 units of water

What is the optimal consumption path in this problem? You could identify each possible consumption path and calculate the sum of its discounted utilities. You may have noticed that there are 20 possible paths, and there is a faster way to do this. First, notice that in time period 3 it will always be optimal to consume everything. So we can eliminate all actions that do not deplete the water stock. We also calculate the immediate utility of the actions that do deplete all water (Figure 2).

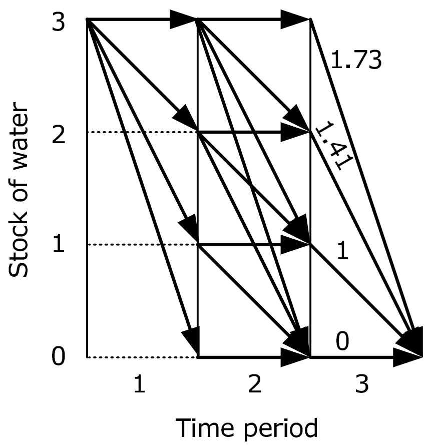

Figure 2: Optimal consumption decisions, and their immediate utility, for all possible stock sizes in time period 3

Now we have reduced our three-period problem to a two-period problem. We now see that for each stock size we face a trade-off between the immediate utility from consumption in time period 2, and the future consumption in time period 3, which we discount by 10%. For example, if in time period 2 we have 3 units, of which we would consume 1, our total utility would be $\sqrt{1} + \sqrt{2}/1.1 \approx 2.29$. Consuming 2 units would give a total utility of $\sqrt{2} + \sqrt{1}/1.1 \approx 2.32$; consuming 3 units would give a total utility of $\sqrt{3} \approx 1.73$. Hence, if in time period we have 3 units left we should consume 2 of them and leave 1 for next time period. Figure 3 shows the other optimal decisions in period 2.

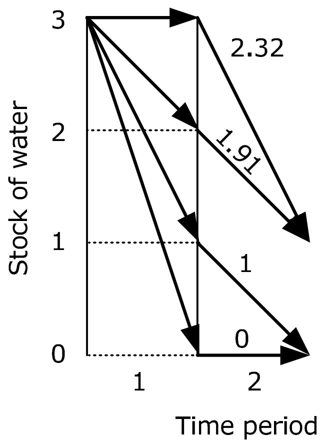

Figure 3: Optimal consumption decisions, and their total utility, for all possible stock sizes in time period 2

Now we arrive at the first decision, and we repeat the same exercise. If we consume nothing our total utility is $2.32/1.1 \approx 2.11$; if we consume 1 unit our total utility is $\sqrt{1} + 1.91/1.1 \approx 2.74$; if we consume 2 units our total utility is $\sqrt{2} + 1/1.1 \approx 2.32$; if we consume 3 units our total utility is $\sqrt{3} \approx 1.73$. Hence, we consume 1 unit in period 1, 1 unit in period 2, and 1 unit in period 3.

What happens if the stock is replenished every year, for instance by rainfall? Let's assume that every year the rain adds 1 unit to the stock of water. It is convenient to depict this problem as an interaction between the decision maker (let's call the decision maker 'Human') and Nature (Figure 4). Now, if we have, say, 3 units at the start of time period t, and we consume 2, we have 1 left at the end of t; however, before we get to the start of $t + 1$ Nature adds 1 unit of water to the stock, so that we start $t + 1$ with 2 units. Note that Nature's behaviour is perfectly predictable, so that we can determine an optimal consumption path over time.

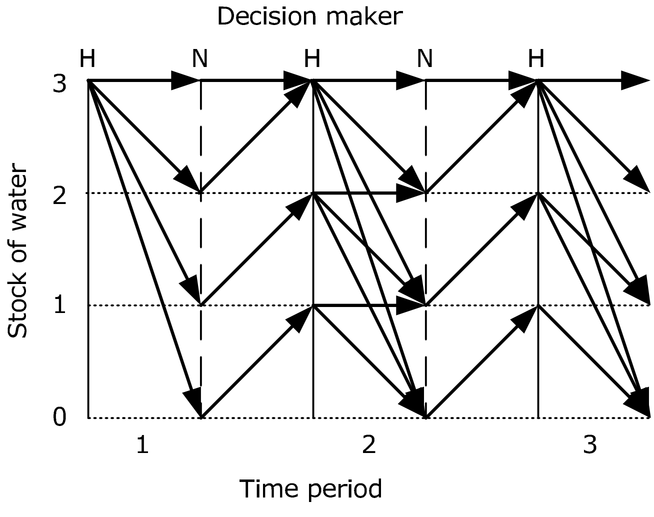

Figure 4: Decision tree when the water stock is replenished every time period by one unit. H = action by the decision maker ('Human'); N = action by Nature

Exercise 1
Solve the problem depicted in Figure 4 by backward induction, assuming a discount rate of 10% and utility function $\sqrt{C_t}$, where C_t denotes water consumption. How much water should you consume in time period 1, 2, and 3? What would your optimal consumption path look like from an initial stock size of 0, 1, and 2 units?

1.2.2 Backward induction in a stochastic world

So far we assumed that the size of our stock is perfectly predictable. This enabled us to describe the optimal consumption path as a function of time: consume 2 units in period 1, 2 units in period 2, and 1 unit in period 3. However, if stock size cannot be predicted with such accuracy, for instance because it depends on such unpredictable events as the weather, planning ahead becomes impossible and we need to define our decisions as a function of stock size rather than time. In other words, what we need is an *adaptive management strategy*, i.e. a strategy that plans for the near future but leaves the option open to change longer-term plans according to new information.

To illustrate this problem, let's assume that the replenishment of our stock of water depends on the weather. In a hot year, when there is little rain, 1 unit of our stock of water evaporates; in a wet year our stock of water is replenished by 1 unit. We do not know in advance which years will be hot and which years will be wet, but let's assume that we know the probability that any given year is a hot year (0.5). Because we assume only two types of weather this also defines the probability that any given year is a wet year (0.5). Furthermore, assume that if we have no water there is nothing to evaporate, so our stock of water remains empty because stock size cannot be negative. The problem now looks as depicted in Figure 5.

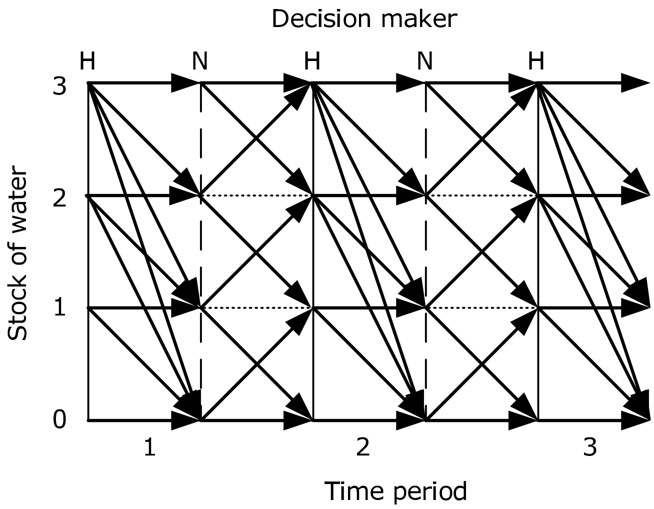

Figure 5: Decision tree for stochastic three-period, four-value problem (H: decision made by Human; N: 'decision' made by Nature)

Again we can solve this problem by backward induction. It should be obvious by now that in the last period we should consume all water there is left, so $C_3 = X_3$ and $U_3 = \sqrt{C_3} = \sqrt{X_3}$. In period 2, however, we don't know what replenishment to expect after we have made (and exercised) our consumption decision for that period. For example, if we have 3 units of water at the start of period 2, and we consume 1, then we might have 3 units again in period 3 (wet year), or we might have only 1 unit (hot year). That makes an expected value of 2, but actually that does not help us much. What we need to know is the expected value of the *utility* of the water stock, which in this example is $0.5 * \sqrt{3} + 0.5 * \sqrt{1} \approx 1.37$. And because we need to wait a year before we can enjoy that utility, we discount this by 10%, so that consuming 1 out of 3 units in period 2 gives us a utility of $\sqrt{1} + 1.37/1.1 \approx 2.24$. So you see the overall procedure is very similar to that in the deterministic model; the only difference is that for Nature's actions you calculate the discounted expected value of future utility.

> *Exercise* 2
> Solve the problem depicted in Figure 5 by backward induction, assuming a discount rate of 10% and utility function $\sqrt{C_t}$, where C_t denotes water consumption. How much water should you consume in time period 1, 2, and 3, depending on the size of your stock of water?

1.3 Bellman's Principle of Optimality

You see that backward induction by hand is quite labor intensive. It is much more efficient to do the procedure in a numerical model, for instance in Matlab or R. In fact, if you model backward induction this way you come very close to one of the applications of Bellman's Principle of Optimality. This principle is at the heart of what we call Dynamic Programming. Dynamic Programming was developed by the American mathematician Richard Bellman around the same time when Lev Pontryagin published the Maximum Principle. The underlying principle of dynamic programming, also called Bellman's Principle of Optimality, is as follows:

> An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision.

To understand this principle, take a look again at the stochastic water management problem in Exercise 2: in fact, we applied it when we used backward induction to solve it. Our results tell us that if, say, we have one unit of water in time period 2, the optimal decision is to use all that water. It makes no difference what we did before time period 2: whether we consumed one out of three units in period 1 and suffered a hot year after that, or consumed 2 out of 2 units and enjoyed a wet year, or consumed 1 out of 1 unit and enjoyed a wet year, the optimal water use in time period 2 is to consume 1 unit.

This also means that to make the right decision, we do not need to know all the details of the remaining actions. Once we know the optimal decisions from time period 1 onwards, and the present value of their payoffs, we only have to consider the tradeoff between our immediate utility from consumption and the present value of the expected utility that we'll derive from our remaining stock.

Bellman's Principle of Optimality is expressed by the so-called Bellman Equation:

$$V(X_t) = \max_{C_t} \left\{ U(X_t, C_t) + \frac{1}{1+r} V(X_{t+1}(X_t, C_t)) \right\} \tag{1}$$

where $V(X_t)$ denotes the so-called value function, which expresses how much a stock of size X_t is worth to us now; $U(X_t, C_t)$ is the immediate utility function; C_t is the control variable in period t; and r is the discrete-time discount rate. The Bellman Equation tells us that in each time period we maximize the sum of two values. The first value is the immediate utility we derive from a given combination of the stock variable and the control variable. The second value is the discounted value of the problem we face next year. The value of this problem depends on the stock size next year, which in turn depends on the combination of stock size and our control variable this year.

Bellman's Principle of Optimality can be applied analytically, but its main virtue lies in numerical applications. First of all, its analytical application is much more difficult than the analytical application of the Maximum Principle. One can solve a few simple optimization problems using the Bellman equation, but the Maximum Principle is more widely applicable. Second, Bellman's Principle of Optimality allows for inclusion of stochastic dynamics. The Maximum Principle allows you to derive an optimal extraction path: in other words, the control variable becomes a function of time. But such an optimal path is like a fisheries manager who has written in his calendar for the coming, say, 50 or 100 years the optimal TAC for each year. The uncertainty in fish stock dynamics makes such long-term planning worthless because the plan will be outdated within a few years. It would be much better if this fisheries manager makes his decision dependent not on time, but on stock size. In that case he would each year consider the most recent estimate of the size of the fish stock, and use that information to decide on the optimal TAC for that year. The next year, he will have new information that will determine another TAC decision.

1.4 Dynamic Programming and the Maximum Principle

Despite the difficulty in using the Bellman equation in analytical models, getting a bit deeper into the theory helps to see the links between the two approaches.

1.4.1 Deriving the dynamic optimality condition for a search fishery

Consider a fishery where the costs of catching a given amount of fish increase with stock abundance. The Bellman Equation for this problem is[1]

[1]This section draws heavily on Conrad, J. 2010. Resource Economics. Second edition, Cambridge University Press, NY.

$$V(X_t) = \max_{Y_t} \left\{ pY_t - C\left(X_t, Y_t\right) + \frac{1}{1+r} V(X_t + G(X_t) - Y_t) \right\} \quad (2)$$

Assume a steady state so that $X_t = X_{t+1} = X$ and $Y_t = Y_{t+1} = Y$. We first derive the optimality condition for Y, so we take the first derivative of the RHS of Equation 2 with respect to Y and put it equal to zero:

$$p - C_Y + \frac{1}{1+r} V_X X_Y = 0 \Rightarrow p - C_Y = \frac{1}{1+r} V_X \quad (3)$$

So in the optimum the marginal net benefits from catching more fish (LHS) are equal to the marginal net benefits from leaving more fish in the sea (RHS). To find an expression for V_X we take the first derivative of both sides of Equation 2:

$$V_X = -C_X + \frac{1}{1+r} V_X \left[1 + G_X\right] \Rightarrow V_X = -\frac{1+r}{r - G_X} C_X \quad (4)$$

Substituting Equation 4 into Equation 3 gives

$$p - C_Y = -\frac{1}{r - G_X} C_X \Rightarrow r\left[p - C_Y\right] = \left[p - C_Y\right] G_X - C_X \quad (5)$$

This is identical to what we found with the Maximum Principle, except for the discrete-time discount rate r instead of the continuous-time discount rate ρ.

1.4.2 The Bellman Equation and the Hamiltonian

In fact, the Bellman Equation conveys the same information as the Hamiltonian, and seeing the link between the two helps understand Hamiltonian. We can see this if we convert the discrete-time Bellman Equation to its continuous-time equivalent[2]. First, Δt be the length of our time period. Second, let u be the immediate utility enjoyed *per time unit*, so that the total immediate utility is $U = u\Delta t$. Likewise we define the control variable c such that $C = c\Delta t$ and the continuous-time discount rate ρ such that $r = \rho\Delta t$. The Bellman Equation (1) then becomes

$$V(X, t) = \max_{c} \left\{ u(X, c, t)\Delta t + \frac{1}{1 + \rho\Delta t} V(X_{t+\Delta t}(X, c), t + \Delta t) \right\} \quad (6)$$

Multiplying this by $1 + \rho\Delta t$ gives

$$\left[1 + \rho\Delta t\right] V(X, t) = \max_{c} \left\{ u(X, c, t)\Delta t \left[1 + \rho\Delta t\right] + V(X_{t+\Delta t}(X, c), t + \Delta t) \right\} \quad (7)$$

Which we can rewrite to

$$\rho\Delta t V(X, t) = \max_{c} \left\{ u(X, c, t)\Delta t \left[1 + \rho\Delta t\right] + V(X_{t+\Delta t}(X, c), t + \Delta t) - V(X, t) \right\} \quad (8)$$

Divide this by Δt:

$$\rho V(X, t) = \max_{c} \left\{ u(X, c, t) \left[1 + \rho\Delta t\right] + \frac{1}{\Delta t} \left[V(X_{t+\Delta t}(X, c), t + \Delta t) - V(X, t)\right] \right\} \quad (9)$$

Letting Δt go to zero gives

$$\rho V(X, t) = \max_{c} \left\{ u(X, c, t) + \frac{dV}{dX} \frac{dX}{dt} \right\} \quad (10)$$

Note the similarity with the Hamiltonian: this illustrates how the costate variable expresses the increase in the value of the Hamiltonian due to an increase in the stock variable.

[2]This section draws heavily on Dixit, A.K. and Pindyck, R.S. 1994. Investment under Uncertainty. Princeton University Press, Princeton, NJ.

1.5 Numerical Stochastic Dynamic Programming

There are several ways to apply Bellman's Principle of Optimality in numerical analysis. One of the easiest methods is called Value Function Iteration (VFI), and you can interpret VFI as a form of backward induction. The overall procedure of VFI is as follows:

1. Discretize the state variable X
2. Guess some value for next time period's value function V'
3. For each possible value of the state variable X, identify the value of the control variable C that maximizes the sum of immediate utility $U(X, C)$ and the present value of future utility $[1/[1+r]]V'(X'(X, C))$
4. For each possible value of the state variable X, set the value of V' equal to the maximum value found in step 3
5. Repeat steps 3 and 4 until you reach your stopping criterion.

1.5.1 Step 1: Discretize the state space

In our water management example we discretized the stock of water into four values: 0, 1, 2, or 3 units. In other words, we had defined a vector $\mathbf{X}$ that contains all possible values of X:

$$\mathbf{X} = \begin{bmatrix} 0 \\ 1 \\ 2 \\ 3 \end{bmatrix} \tag{11}$$

1.5.2 Step 2: Guess some value for next year's value function

We also need a vector $\mathbf{V}$ that represents the value function. $\mathbf{V}$ has the same size as $\mathbf{X}$, because it will tell us for each possible value of the stock variable X its value in utility terms, taking into account all possible future uses. Interpreting Value Function Iteration as a form of backward induction it is logical to start with a value function of all zeros:

$$\mathbf{V} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{12}$$

Exercise 3
Open a new script file in R and write a piece of code to make the vectors $\mathbf{X}$ and $\mathbf{V}$. Remember to write your code in a general fashion, to give your variables descriptive names, and to add comments to explain what the code is supposed to do.

1.5.3 Step 3: Find the optimal value of the control variable

In this step we maximize, for each entry of $\mathbf{X}$, the following objective:

$$\max_{C} \left\{ \sqrt{C} + \frac{1}{1+r} \left[pV(X - C - 1) + [1-p]V(X - C + 1) \right] \right\} \tag{13}$$

Where p denotes the probability of a hot year. To simplify the modelling somewhat we define water use as the difference between stock size before consumption and stock size after consumption:

$$\max_{X'} \left\{ \sqrt{X - X'} + \frac{1}{1+r} \left[pV(X' - 1) + [1-p]V(X' + 1) \right] \right\} \tag{14}$$

Also note that if $X' = 0$ there is no water to evaporate; on the other hand, if $X' = 3$ any rain water will spill over and wash away. So we should define the objective function such that

$$X' - 1 = \begin{cases} 0 & \text{if } X' = 0 \\ X' - 1 & \text{if } X' > 0 \end{cases} \tag{15}$$

and

$$X' + 1 = \begin{cases} X' + 1 & \text{if } X' < 3 \\ 3 & \text{if } X' = 3 \end{cases} \tag{16}$$

We can denote this as

$$\max_{X'} \left\{ \sqrt{X - X'} + \frac{1}{1+r} \left[pV(\max(0, X' - 1)) + [1 - p]V(\min(3, X' + 1)) \right] \right\} \tag{17}$$

Let's first address the immediate utility part, i.e. the first term in this objective function. We define a matrix that contains the utility for all combinations of stock size before consumption and stock size after consumption:

$$\mathbf{U} = \begin{bmatrix} \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \end{bmatrix} \tag{18}$$

Then we calculate, for each combination of X and X', the value of the corresponding entry of $\mathbf{U}$. Note, however, that consumption cannot be negative! Hence, we calculate immediate utility as follows:

$$U = \begin{cases} \sqrt{X - X'} & \text{if } X \geq X' \\ -\infty & \text{if } X < X' \end{cases} \tag{19}$$

By setting $U = -\infty$ if $X < X'$ we make sure that it is never optimal to choose this combination of X and X'.

> *Exercise* 4
> In your R script file, write a piece of code to make the matrix $\mathbf{U}$. Use the R function `array` to do this. To see how `array` works, type `help(array)`.

> *Exercise* 5
> Once you have made a matrix for $\mathbf{U}$, write a script to calculate its value. (You need to use `for` in this step.) Remember to write your code in a general fashion, to give your variables descriptive names, and to add comments to explain what the code is supposed to do.

If the calculation was done correctly $\mathbf{U}$ should look like this:

$$\mathbf{U} = \begin{bmatrix} 0 & -\infty & -\infty & -\infty \\ 1 & 0 & -\infty & -\infty \\ 1.41 & 1 & 0 & -\infty \\ 1.73 & 1.41 & 1 & 0 \end{bmatrix} \tag{20}$$

With these values we can calculate the sum of this period's immediate utility and the value of next period's stock. The latter is read from the value vector $\mathbf{V}$. First we define a matrix $\mathbf{W}$ that expresses total utility for all combinations of X and X':

$$\mathbf{W} = \begin{bmatrix} \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \end{bmatrix} \tag{21}$$

Then we calculate the values in this matrix according to Equation (17).

Exercise 6
In your R script file, write a piece of code to make the matrix **W**, and to calculate its value. Remember to write your code in a general fashion, to give your variables descriptive names, and to add comments to explain what the code is supposed to do.

Note that in the first iteration all entries in **V** are zero, so if the coding is correct we should have

$$\mathbf{W} = \mathbf{U} = \begin{bmatrix} 0 & -\infty & -\infty & -\infty \\ 1 & 0 & -\infty & -\infty \\ 1.41 & 1 & 0 & -\infty \\ 1.73 & 1.41 & 1 & 0 \end{bmatrix} \tag{22}$$

We can now, for each row in **W**, identify the column that maximizes total utility. We therefore define a vector **C** of the same length as **X**, that denotes the optimal consumption for each stock level.

Exercise 7
In your R script file, write a piece of code to make the vector **C**, and to calculate its value. Remember to write your code in a general fashion, to give your variables descriptive names, and to add comments to explain what the code is supposed to do.

1.5.4 Step 4: Set the value of the value function equal to the maximum found

Now that we have identified, for each stock size, the optimal consumption, we can assign the total utility derived from it to the value function.

Exercise 8
In your R script file, write a piece of code to calculate the new value of **V**. Remember to write your code in a general fashion, to give your variables descriptive names, and to add comments to explain what the code is supposed to do.

1.5.5 Step 5: Repeat steps 3-4 until convergence

The value of **V** that we have now calculated reflects the utility value of each possible stock size in the final period. By iterating backwards in time we can calculate the utility value of the stock size in the second to last period, third to last period, and so on. The most convenient way to do this in R is to put the iterations within a loop. Mind, however, that the immediate utility matrix **U** is independent of the time period, so we only have to repeat the calculation of **W**, **C** and **V**.

Exercise 9
In your R script file, write a piece of code to perform the calculation of **W**, **C** and **V** three times. Remember to write your code in a general fashion, to give your variables descriptive names, and to add comments to explain what the code is supposed to do.

If the model was programmed correctly the optimal water consumption in each year should be as given in Table 1.

Table 1: Optimal consumption in the three-period, four-level water management example under constant replenishment

X	C_0^*	C_1^*	C_2^*
0	0	0	0
1	1	1	1
2	2	2	2
3	2	2	3