Dynamic Programming Part B

1 Functional Equations and Value Function Properties

Dynamic programming problems often lead to **functional equations**—equations in which the *unknown* is a function rather than a scalar. The **Bellman equation** is the most important example in economics. Instead of solving for a scalar like x, we are solving for an entire function $V(\cdot)$ that satisfies a relationship involving itself.

1.1 What Is a Functional Equation?

- A **functional equation** relates the value of a function at one point to its value at another point.
- In its simplest form, it can look like:

$$V(s) = F(s, V(g(s)))$$

- where the function V appears on both sides.

- We are not trying to find a single number, but rather a function $V(\cdot)$ that makes this equation true for every s.
- Functional equations appear whenever we describe **recursive behavior** — situations where "the value of something today depends on the value of something tomorrow."

A functional equation specifies a relationship between a function and its transformed version.

In dynamic optimization:

$$V(s) = \max_{x \in X(s)} \{\pi(s, x) + \beta V(f(s, x))\}$$

where:

- $V(s)$ is the **value function**, giving the maximum attainable value from state s.
- $\pi(s, x)$ is the **current payoff** (or profit, utility).
- $f(s, x)$ gives the **next-period state**.
- $\beta \in (0, 1)$ is the **discount factor**.

The equation says: *The value of being in state s today equals the current payoff plus the discounted value of the next state, assuming optimal choice x.*

1.1.1 Functional Equations in Economics

Example: A Simple Savings Problem

Suppose an agent chooses consumption c_t and next-period assets a_{t+1}.

$$
\begin{aligned}
&\max_{\{c_t, a_{t+1}\}_{t=0}^{\infty}} \sum_{t=0}^{\infty} \beta^t u(c_t) \\
&\text{s.t. } c_t + a_{t+1} = (1+r)a_t + y_t, \quad a_t \geq 0
\end{aligned}
$$

where:

- $u(c)$ is the utility function (e.g., $u(c) = \log(c)$),
- r is the interest rate,
- y_t is income,
- $\beta \in (0, 1)$ is the discount factor.

Then the Bellman equation is

$$V(a_t) = \max_{a_{t+1} \geq 0} \{u((1+r)a_t + y - a_{t+1}) + \beta V(a_{t+1})\}.$$

FOC (interior) with respect to a_{t+1}

Let $c_t = (1+r)a_t + y - a_{t+1}$. Then

$$\frac{\partial}{\partial a_{t+1}} \Big[u((1+r)a_t + y - a_{t+1}) + \beta V(a_{t+1})\Big] = -u'(c_t) + \beta V'(a_{t+1}) = 0,$$

The agent trades off **current utility** $u(c)$ versus the **future value** $\beta V(a')$.

Firm Investment Problem

A firm with capital stock k_t chooses investment i_t:

$$\max_{i_t} \sum_{t=0}^{\infty} \beta^t \pi(k_t, i_t) \quad \text{s.t.} \quad k_{t+1} = (1-\delta)k_t + i_t, \quad k_t \geq 0.$$

The recursive form:

$$V(k_t) = \max_{i_t \geq 0} \{\pi(k_t, i_t) + \beta V((1-\delta)k_t + i_t)\}.$$

The function $V(k_t)$ gives the value of capital k_t as current profit plus discounted future (optimal) value of next-period capital. Again, the value function appears on both sides.

FOC with respect to i_t:

$$\frac{\partial}{\partial i_t} \Big[\pi(k_t, i_t) + \beta V((1-\delta)k_t + i_t)\Big] = \pi_i(k_t, i_t) + \beta V'((1-\delta)k_t + i_t) = 0.$$

Resource Extraction Problem

A resource owner decides how much to extract x_t from stock s_t:

$$V(s_t) = \max_{0 \leq x_t \leq s_t} \{px_t - c(x_t) + \beta V(s_t - x_t)\}.$$

Here:

- The **state** is remaining stock s_t.
- The **control** is extraction x_t.
- The **transition** is $s_{t+1} = s_t - x_t$.

The functional equation states that the value of the resource today equals profit from extraction plus the discounted value of what remains for tomorrow.

1.2 The Bellman Operator

The Bellman equation tells us that the value of being in a given state today, $V(s)$, equals the best possible current payoff plus the discounted value of what happens next.

We can think of this process as an **operator** — a kind of "machine" that takes a guess about the value function and produces a new, updated guess.

1.2.1 How it works

Start with any function $V(s)$ that tells you what the value might be for each state.
Then define a new function:

$$(\mathcal{T}V)(s) = \max_{x \in X(s)} \big\{\pi(s,x) + \beta V(f(s,x))\big\}.$$

What this says in words:

> "Given my current guess about how valuable future states are (that's V), what would be the total value of making the best decision today?"

So $\mathcal{T}$ takes the *old* value function and gives you a *new* one that's a little closer to the truth.
It's a way to **think one step ahead**.

1.2.2 The Fixed Point Idea

The true value function, $V^*(s)$, is the one that **doesn't change** when we apply this operation again:

$$V^*(s) = (\mathcal{T}V^*)(s).$$

In other words, if you already know the correct V^*, thinking one step ahead doesn't change your beliefs because you are already correct about the future.

Why this matters

- The operator gives us a **recipe for computing** V^*:
  - start with a guess and keep applying $\mathcal{T}$ repeatedly.
  - Each time, you're improving your estimate of the value of being in each state.
- Economically, this process mirrors **learning or planning**:
  - we evaluate today's decisions using our expectations of tomorrow, adjust, and repeat until everything is internally consistent.

Visual Intuition

- If you plotted $V_0(s)$ (your first guess) and then $V_1(s) = \mathcal{T}V_0(s)$, the curves would move closer and closer together until they line up at $V^*(s)$.
- That's what it means for the Bellman equation to be a **fixed point** — a steady state in your expectations about value.

In the context of economics

- For a **consumer**, $\mathcal{T}$ means re-evaluating how much future consumption is worth.
- For a **firm**, it means updating the expected profitability of holding or investing capital.

- For a **resource owner**, it means revising how valuable it is to leave part of the stock for tomorrow.

In all cases, the Bellman operator captures the logic of **forward-looking behavior**:
today's value depends on how optimally we plan for tomorrow.

Each application of $\mathcal{T}$ corresponds to "thinking one step further ahead."

- Starting with any initial guess $V_0(s)$,
- Repeatedly applying $\mathcal{T}$, $V_{k+1} = \mathcal{T}V_k$,
- Converges to the true value function V^* under mild conditions.

This is **Value Function Iteration (VFI)**.

1.3 Existence and Uniqueness: Why the Bellman Equation Has a Single Solution

Once we define the Bellman operator $\mathcal{T}$ - the rule that takes a guess about future value and updates it - we can ask two key questions:

1. Does this process always lead to a stable value function?
2. Will it always settle on the **same** function, no matter where we start?

The answer is yes — as long as future payoffs are **discounted** (so $\beta < 1$).

The reason is the **contraction mapping property**.

A **contraction mapping** is a transformation that *pulls things closer together* every time you apply it.

$$||\mathcal{T}V_1 - \mathcal{T}V_2|| \leq \gamma\, ||V_1 - V_2||$$

Imagine taking two different guesses about the value function, say $V_1(s)$ and $V_2(s)$. When we apply the Bellman operator to both, the resulting functions $\mathcal{T}V_1$ and $\mathcal{T}V_2$ are **closer to each other** than the originals.

Each round of updating reduces the distance between our guesses — eventually, all sequences converge to the same point, the **true value function** V^*.

Discounting is what makes this work. Because future rewards are multiplied by $\beta < 1$, any disagreement about the future is automatically **shrunk** when we think one step ahead.

Small differences in how we value the future can't explode backward into large differences today — they fade over time.

This gives the Bellman operator its *gravitational pull* toward a single stable value function.

If applying $\mathcal{T}$ repeatedly always pulls guesses closer together, there must be one and only one function that can't be improved upon — that's the **fixed point**:

$$V^*(s) = \mathcal{T}V^*(s).$$

Mathematically, this follows from the **Contraction Mapping Theorem**, but economically, you can think of it as saying:

There is one internally consistent way to value the future that agrees with itself when we plan forward.

1.3.1 How this helps us in practice

Because $\mathcal{T}$ is a contraction, we can find V^* by **value function iteration**:

1. Start with any initial guess $V_0(s)$ — even something crude.
2. Apply $\mathcal{T}$ to get an updated guess $V_1 = \mathcal{T}V_0$.

3. Keep repeating: $V_{n+1} = \mathcal{T} V_n$.

Each iteration gets us closer to the truth.
No calculus tricks, no global search — just forward iteration guided by economic logic.

1.3.2 The big takeaway

- Discounting and diminishing returns make the future "well-behaved."
- Together they guarantee that the Bellman equation has **one** solution, and that repeated forward-looking reasoning will find it.
- This is why we can compute dynamic equilibria with confidence: as long as the problem is discounted and well-behaved, there is a single, stable value function waiting to be found.

1.4 Euler Equations and the Envelope Condition

Dynamic optimization gives two complementary characterizations of optimal behavior:

- **Euler equation (FOC in the control):** how the agent trades off current vs. future returns when choosing x.
- **Envelope condition (FOC "in the state"):** how the lifetime value V changes with the state s.

The envelope condition is what lets us **eliminate messy derivatives of the policy function** and express the Euler equation in terms of primitives and V' only.

We work with the same notation as above:

$$V(s) = \max_{x \in X(s)} \{\pi(s, x) + \beta V(f(s, x))\}, \quad 0 < \beta < 1,$$

with optimal policy $x^*(s)$.

1.4.1 From Bellman to First-Order Conditions

Assume an interior, differentiable solution for intuition (we add bounds/KKT below).

- **Stationarity (optimum in x):**

$$\pi_x(s, x^*(s)) + \beta\, V'(f(s, x^*(s)))\, f_x(s, x^*(s)) = 0.$$

- **Envelope (optimum in s):**

$$V'(s) = \pi_s(s, x^*(s)) + \beta\, V'(f(s, x^*(s)))\, f_s(s, x^*(s)).$$

Why envelope works: when differentiating the max wrt s, the chain-rule term involving $x_s^*(s)$ vanishes because the stationarity condition sets the derivative wrt x to zero at the optimum. Intuitively, a marginal change in s doesn't induce a first-order change through the (already optimized) $x^*(s)$.

1.4.2 The (One-Step-Ahead) Euler Equation

Write the time-t Bellman equation at (s_t, x_t) with $s_{t+1} = f(s_t, x_t)$:

$$\pi_x(s_t, x_t) + \beta\, V'(s_{t+1})\, f_x(s_t, x_t) = 0.$$

This is the **Euler equation**: the current marginal payoff from x_t equals the discounted marginal value of how x_t moves the state into the future.

We can **eliminate** $V'(s_{t+1})$ using the envelope condition at $t+1$:

$$V'(s_{t+1}) = \pi_s(s_{t+1}, x_{t+1}) + \beta\, V'(s_{t+2})\, f_s(s_{t+1}, x_{t+1}),$$

which yields a purely **primitive** intertemporal tradeoff once substituted back. In many applications π does not depend directly on s (only through feasibility), making this especially clean.

1.4.2.1 Consumption–Savings (simple, separable)

Let's illustrate with the **savings–consumption problem**.

The household chooses next period's assets a_{t+1} each period:

$$V(a_t) = \max_{a_{t+1} \geq 0} \Big\{ u(c_t) + \beta V(a_{t+1}) \Big\},$$

subject to the budget constraint

$$c_t + a_{t+1} = (1+r)a_t + y.$$

Substitute $c_t = (1+r)a_t + y - a_{t+1}$, so

$$V(a_t) = \max_{a_{t+1} \geq 0} \{ u((1+r)a_t + y - a_{t+1}) + \beta V(a_{t+1}) \}.$$

The First-Order Condition (Euler Equation)

Take the derivative of the Bellman RHS with respect to a_{t+1}:

$$-u'(c_t) + \beta V'(a_{t+1}) = 0,$$

which gives

$$u'(c_t) = \beta V'(a_{t+1}).$$

This is the **Euler condition** in implicit form: the marginal utility today equals the discounted shadow value of next period's assets.

The Envelope Condition

Differentiate the Bellman equation with respect to a_t, treating a_{t+1} as constant (by the envelope theorem):

$$V'(a_t) = u'(c_t)(1+r).$$

> Why We Can "Treat a_{t+1} as Constant"
>
> $$\frac{dV(a_t)}{da_t} = u'(c_t)\big[(1+r) - \frac{da^*_{t+1}(a_t)}{da_t}\big] + \beta V'(a_{t+1}) \frac{da^*_{t+1}(a_t)}{da_t}.$$
>
> Now, note that the **first-order condition** for optimal a_{t+1} is:
>
> $$-\,u'(c_t) + \beta V'(a_{t+1}) = 0.$$
>
> Rearrange and substitute this into the derivative:
>
> $$\frac{dV(a_t)}{da_t} = u'(c_t)(1+r) - \big[u'(c_t) - \beta V'(a_{t+1})\big] \frac{da^*_{t+1}(a_t)}{da_t}.$$
>
> The term in brackets equals zero by the FOC, so the entire last term drops out:
>
> $$V'(a_t) = u'(c_t)(1+r).$$

Intuitively: a marginal increase in assets a_t raises consumption by $(1+r)$, increasing utility by $u'(c_t)(1+r)$.

1.4.3 Why This Matters

- The envelope theorem allows us to link **marginal value** ($V'(s)$) to **marginal utility** or **marginal profit**, depending on context.
- It simplifies derivations of **Euler equations**, **costate equations**, and **first-order necessary conditions**.
- It provides a recursive way to compute derivatives of value functions in both analytical and numerical work.