Macroeconomic Theory:
Value Function Iterations: Numerical Methods

Chien-Chiang Wang

November 1, 2021

Value Function Iteration

- I demonstrate how to solve (FE) numerically

$$v(k) = \max_{k' \in [0, g(k)]} \left\{ u(g(k) - k') + \beta v(k') \right\} \quad \text{(FE (Growth))}$$

- Our goal is to solve for the fixed point of the contraction mapping: $T : V \to V$

Goal : Find v^* s.t. $Tv^* = v^*$

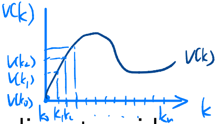
$$Tv(k) = \max_{k' \in [0, g(k)]} \left\{ u(g(k) - k') + \beta v(k') \right\}$$

- Let $u(c) = \frac{c^{1-\theta} - 1}{1 - \theta}$, $f(k) = Ak^{\alpha}$

Value Function Iteration

$V_0, \; V_1 = TV_0, \; T_2 = TV_1, \; \ldots\ldots, \; V^*$ → Given any V_0, we can converge to V^*.

↓ step 1. choose V_0,



Step 1 [Conjecture an initial value function $v(k)$]:

- We approximate continuous functions by using discrete grids
- We use two vectors together to represent a function
- The first vector denotes the domain of v, and the second vector denotes the image of v
- Discretize the domain by constructing a vector:

不一定是第0期,只是我們的初始值

$$\bar{X} = \{0 = \underline{k_0}, k_1, k_2, \ldots, k_n = \bar{k}\}$$

[Matlab]: $kseq = 0 : diff : kbar;$

Value Function Iteration

- We create another vector to represent the image of the value function $v(k)$ on $\bar{X}$:

$$\{v_0 = v(k_0), v_1 = v(k_1), v_2 = v(k_2), \ldots, v_n = v(k_n)\}$$

[Matlab]: $vseq = zeros(1, length(k));$

- By creating a zero vector, the initial value function we create is $\underline{v(k) = 0}$

↳ 用 zero func. 當作 initial V

cf. 也可以用 one func. vseq = ones(1, length(k))

Value Function Iteration

initial func.

$k' \in [k_0, k_1, \ldots k_n]$

找一个能 max 的 k'

Step 2. [Given $v(k)$, solve for $Tv(k)$]:

utility func.

$$\underbrace{Tv(k)}_{k_i} = \max_{\underbrace{k'}_{\text{Goal}}} \{ u(\underbrace{g}_{\text{product func. } f}(k) - k') + \beta \underbrace{v}_{\text{我们 let 成 zero func.}}(k') \} \tag{1}$$

- We create an vector to represent $Tv(k)$:

$$\{Tv_0 = Tv(k_0), Tv_1 = Tv(k_1), Tv_2 = Tv(k_2), \ldots, Tv_n = Tv(k_n)\}$$

[Matlab]: $Tvseq = zeros(1, length(k));$

先用0，接下来填数字进去

$[Tv(k_0), Tv(k_1), \ldots\ldots, Tv(k_n)]$

Value Function Iteration

↳ 說明上一頁 Tv 怎麼求

- Given $k_i \in \bar{X}$, we want to solve for $Tv(k_i)$: find $k' \in \bar{X}$ that maximizes $u(g(k_i) - k') + \beta v(k')$
- We compute $u(g(k_i) - k_j) + \beta v(k_j)$ for all j

  $k_j = [k_0, k_1, \ldots, k_n]$

  不用每個都試 有些根本不 feasible.
- However, the feasible k' must satisfy $k' \leq g(k_i)$
- Therefore, for $k' > g(k_i)$, we set $u(g(k_i) - k') + \beta v(k')$ to be negative infinite $\to -\infty$, i.e. 直接說不可能選到它

  ↳ In matlab, this means lower than anything else.

Value Function Iteration

- **Step 2.1** Given i, solve for the value of $u(g(k_i) - k_j) + \beta v_j$ for all j
- [Matlab ($\star$1)]

$$
\begin{array}{l}
\textit{for } j = 1 : length(kseq) \\
\quad \textit{if } g(kseq(i)) - kseq(j) >= 0 \quad \text{只有这样才要考虑} \\
\quad\quad Tvtemp(j) = u(g(kseq(i)) - kseq(j)) + \beta vseq(j); \\
\quad \textit{else} \\
\quad\quad Tvtemp(j) = -\inf; \quad \text{其他直接令成} -\infty\text{，不管} \\
\textit{end}
\end{array}
$$

Value Function Iteration

$Tv\,temp = (Tv\,temp_0,\ Tv\,temp_1,\ \ldots\ldots,\ Tv\,temp_n)$

$Tv\,temp_0 = u(g(k_i) - k_0) + \beta v(k_0)$

$\vdots \qquad\qquad \vdots$

$Tv\,temp_n = u(g(k_i) - k_n) + \beta v(k_n)$

- **Step 2.2:** find $Tv(k_i)$ and the policy function $h(k_i)$
- [Matlab (⋆2)]
  $[M, m] = max(Tvtemp);$
  $Tvseq(i) = M;$
  $ipol(i) = m;$

M 是最大值
m 是使得最大值的那個 k

policy func.
$h(k_i) = k_j$

當我capital是k_i時，那個capital會跑出 max(Tvtemp)

我們要找到最大 那個 Tvtemp

$$\begin{pmatrix} Tv\,temp_0 \\ Tv\,temp_1 \\ \vdots \\ \vdots \\ Tv\,temp_n \end{pmatrix} \begin{matrix} \\ \\ = -\infty \\ = -\infty \\ = -\infty \end{matrix}$$

超過某個k之後 剩下都會是 $-\infty$

Value Function Iteration

- **Step 2.2** Conduct the calculation for $i = 1, 2, \ldots, \mathit{length}(\mathit{kseq})$ → 每個 i 都跑一次
  for $i = 1 : \mathit{length}(\mathit{kseq})$
    [Matlab ($\star$1)]
    [Matlab ($\star$2)]
  end

Given v ⇒ Get Tv

Now, finish CM once,

Next, repeat CM multiple times.

Value Function Iteration

跑多少次 CMT

$d_\infty = \max|Tv - v|$

Step 3. Replace the original *vseq* vector by *Tvseq*, and repeat Step 2 for multiple times

- $for\ \ h = 1 : num$

  $\quad for\ \ i = 1 : length(kseq)$

  $\qquad$[Matlab ($\star$1)]

  $\qquad$[Matlab ($\star$2)]

  $\quad end$

  $\quad d = \max(abs(Tvseq - vseq));$

  $\quad vseq = Tvseq;$

  end

迭代要定義很多Tvseq大麻煩

$\Downarrow$

∴直接把Tvseq取代vseq然後繼續跑

$$
\begin{cases}
vseq = 0 & (V_0) \\
Tvseq = ? & (V_1) \\
TTvseq = \vdots & V_2 \\
TTTvseq = \vdots & V_3 \\
\vdots & \vdots
\end{cases}
$$

- If d is smaller than the criteria we set (for example, $d < \epsilon \equiv 1e^{-5}$), we say that the value function converges

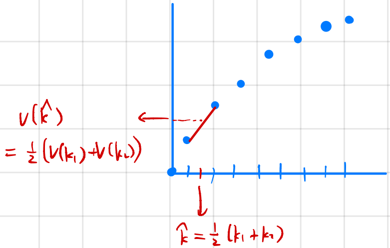
自行一個收斂的標準 不然電腦會跑不會end

- So far we have pinned down the value function and the policy function

Value Function Iteration

求解 v 是什么后，就能用 v 求出 k path

- **Step 4.** As the policy function is also pinned down, we can characterize the whole dynamic paths of capital and consumption
- In Solow model: the dynamic path is pinned down by the law of motion of capital: $k_{t+1} = sf(k_t) + (1-\delta)k_t$
- In Ramsey model, steady state linearization: the dynamic path is pinned down by the saddle path: $\hat{k}_{t+1} = \lambda_1 \hat{k}_t$
- In Ramsey model, value function iteration: the dynamic path is pinned down by the policy function: $k_{t+1} = h(k_t)$



$$\begin{cases} v(k) = \max\ u(g(k)-k') + \beta v(k') \\ k' \in \mathbb{X} = \{0, k_1, k_2, \dots, \bar{k}\} \\ k' \leq g(k) \end{cases}$$

$\Rightarrow$ Interpolation 插值法

$$k' = \hat{k} \in (k_1, k_2)$$

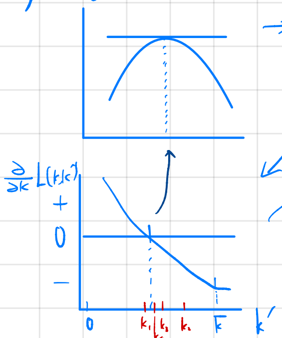
$\Rightarrow$ $k' \in [0, \bar{k}] \longrightarrow$ if 完全不知 func. 的資訊
then 就必須試 ∞ 點，非常難找到 k^*

But if V_0 is concave,
then $u(g(k)-k') + \beta v(k')$ would be concave.

e.g. Let $L(k,k') = u(g(k)-k') + \beta v(k')$

$\rightarrow$ 可算一階微分

how? By root finding.



就是平常用的證明方法

how to solve? By bisection.
$\bar{k}$ 和 0 之間一定有一點讓 func. 為 0
(by IVT)