Iteração da Função valor
Macroeconomia I

Yuri Passuelo – yuripassuelo@usp.br

6 de maio de 2025

Aqui vamos revisar e detalhar um pouco mais a respeito do algoritmo da iteração da função valor

---

Problema Básico

Inicialmente vamos considerar o problema mais básico do modelo de crescimento neoclássico.

Dado o problema vamos ao passo a passo do algoritmo:

Passo 1: Criação do Grid de Capital

Obs: O motivo de criar um Grid de capital se dá pois o algoritmo se trata de uma implementação que seria colocada dentro de um computador, como não conseguimos representar um continuo dentro de um computador, usamos essa definição de Grid que se trata basicamente da discretização de um intervalo continuo.

$$\mathcal{K} = \{k_1, k_2, ..., k_n\}$$

Aonde temos que:

$$\forall i(k_i \geq 0)$$

$$k_i < k_j, \text{ para } i < j$$

Ou seja, nosso vetor de capital se trata de uma lista ordenada de valores de capital, com a restrição de não negatividade desses valores de capital.

Passo 2: Parametrização da Economia

Dentro do nosso problema básico assim como visto acima trabalhamos com funções utilidade, funções de produção, e temos presente parâmetros em cada uma dessas funções assim como os parâmetros de depreciação do capital δ e de desconto intertemporal β.

Dado que escolhemos por exemplo uma função de utilidade CRRA:

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

E uma função de produção *Cobb-Douglas*:

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

Devemos declarar tanto as funções utilizadas como os parâmetros escolhidos, assim nessa etapa parametrizamos escolhendo δ, β, α, A e σ.

Passo 3: Criação dos objetivos de iteração

Aqui definiremos uma série de objetos que nos ajudaram na etapa de iteração, são basicamente vetores que vamos guardar os números.

- Vetor de iteração da função valor com dimensão $1 \times n$

$$\text{v\_it} = \{0, ..., 0\}$$

- Vetor de iteração da função política com dimensão $1 \times n$

$$\text{g\_it} = \{0, ..., 0\}$$

- Vetor de armazenamento da função valor com dimensão $1 \times n$

$$V = \{0, ..., 0\}$$

Por último vamos definir nosso parâmetro de tolerância para convergência dado aqui como:

$$\text{tol } = 1 \times 10^{-5}$$

Passo 4: Iteração da função Valor

Criar operador Tv de dimensão $1 \times n$ para operar a função valor

$$Tv = \{0, ..., 0\}$$

- Para cada $k \in \mathcal{K}$
  - Para cada $k' \in \mathcal{K}$

    Calcularemos o Consumo como:

    $$c = f(k) - k'$$

    Se $c \leq 0$:

    $$c = -99999$$

    Obs: Caso $c > 0$ não precisamos modificar em nada c

    Com c calculado, calcularemos:

    $$u(c) + \beta V[k']$$

    Obs: Aqui $V[k']$ se refere ao elemento de índice k' do vetor V

    Guardamos o valor calculado em Tv:

    $$Tv[k'] = u(c) + \beta V[k']$$

    Obs: Aqui $Tv[k']$ se refere ao k'-ésimo elemento do vetor Tv

  Finalizado o *Loop* em relação à k' dado um k, pegamos aqui tanto o valor máximo do vetor Tv quanto seu argumento máximo e salvamo-os em duas variáveis:

  $$\text{max\_k} = \text{arg max } (Tv)$$

  $$\text{max\_V} = \text{max } (Tv)$$

  Guardamos os valores nos vetores de iteração v_it e g_it:

  $$\text{v\_it}[k] = \text{max\_v}$$

  $$\text{g\_it}[k] = \text{max\_k}$$

  Obs: Aqui v_it[k] e g_it[k] referem-se ao k-ésimo elemento dos vetores

Passo 5: Checagem de Convergência

Terminado o processo de iteração vamos a checagem de convergência. Primeiro calcularemos a distância de V para v_ini

$$\text{dist} = \max |V - \text{v\_it}|$$

Caso dist < tol então paramos o processo e temos que v_it será nossa função valor $v(k)$ e g_it será nossa função política $g(k)$.

Caso contrário voltamos ao passo 4, mas antes:

$$V = \text{v\_it}$$