2 Lista 2

2.0 [☒] Definições, Proposições, Lemas e Observações

Configuração do Modelo.

- k_t: capital em t
- n_t: horas trabalhadas em t
- y_t: bens de consumo
- $y_t = F(k_t, n_t)$: função de produção agregada
- δ: taxa de depreciação
- i_t: investimento
- $k_{t+1} = (1-\delta)\, k_t + i_t$: lei de movimento do capital
- $y_t \geq i_t + c_t$: factibilidade

Definição 10 (Krueger, 2017). Uma alocação $\{c_t, k_t, n_t\}_{t=0}^{\infty}$ é factível se, para todo $t \geq 0$,

$$\begin{aligned} F(k_t, n_t) &= c_t + k_{t+1} - (1-\delta)k_t \\ c_t &\geq 0, \quad k_t \geq 0, \quad 0 \leq n_t \leq 1 \\ k_0 &\leq \bar{k}_0 \end{aligned}$$

Definição 11 (Krueger, 2017). Uma alocação $\{c_t, k_t, n_t\}_{t=0}^{\infty}$ é Pareto Eficiente se ela é factível e não existe outra alocação factível $\left\{\tilde{c}_t, \tilde{k}_t, \tilde{n}_t\right\}_{t=0}^{\infty}$ tal que

$$\sum_{t=0}^{\infty} \beta^t U(\tilde{c}_t) > \sum_{t=0}^{\infty} \beta^t U(c_t)$$

Formulação Sequencial do Problema do Planejador Social.

$$w\left(\bar{k}_0\right) = \max_{\{c_t, k_t, n_t\}_{t=0}^{\infty}} \sum_{t=0}^{\infty} \beta^t U(c_t)$$

$$s.a. \begin{cases} F(k_t, n_t) = c_t + k_{t+1} - (1-\delta)\, k_t \\ c_t \geq 0 \\ k_t \geq 0 \\ 0 \leq n_t \leq 1 \\ 0 \leq k_0 \leq \bar{k}_0 \end{cases}$$

Hipótese 1. U continuamente diferenciável, estritamente crescente, estritamente côncava e limitada. Satisfaz as condições de Inada $\lim_{c \to 0} U'(c) = 0$. O fator de desconto β satisfaz $\beta \in (0, 1)$.

Hipótese 2. F é continuamente diferenciável e homogênea de grau 1, estritamente crescente e estritamente côncava. Além disso, $F(0,n) = F(k,0) = 0$ para todo $k, n > 0$. Também, F satisfaz as condições de Inada $\lim_{k\to 0} F_k(k,1) = \infty$ e $\lim_{k\to\infty} F_k(k,1) = 0$.

Formulação Recursiva do Problema do Planejador Social.

$$v(k) = \max_{0\leq' f(k)} \{U(f(k) - k') + \beta v(k')\} \tag{3.2}$$

em que:

- (3.2): Equação Funcional ou "Equação de Bellman"
- v: função valor
- k: variável de estado
- k': variável de controle
- $k' = g(k)$: função política

Problema do Planejador com Horizonte Finito T.

$$w^T(\bar{k}_0) = \max_{\{k_{t+1}\}_{t=0}^T} \sum_{t=0}^{T} \beta^t U(f(k_t) - k_{t+1}), \quad s.a. \begin{cases} 0 \leq k_{t+1} \leq f(k_t) \\ k_0 = \bar{k}_0 > 0 \text{ dado} \end{cases}$$

em que:

- $(t_0, t_1, ..., t_T)$ e $(k_1,\ k_2, \ldots, k_T, k_{T+1})$
- $\beta^t U(f(k_t) - k_{t+1}) = \beta^t U(c_t)$
- $\beta^T U[f(k_T) - k_{T+1}]$
- escolha de $T+1$ variáveis em um conjunto compacto
- $k_{T+1} = 0$ no ótimo
- $0 \leq k_{t+1} \leq f(k_t) \leq f(K)$

Problema do Planejador com Horizonte Infinito $t \to \infty$.

$$w(\bar{k}_0) = \max_{\{k_{t+1}\}_{t=0}^\infty} \sum_{t=0}^{\infty} \beta^t U(f(k_t) - k_t), \quad s.a. \begin{cases} 0 \leq k_{t+1} \leq f(k_t) \\ k_0 = \bar{k}_0 > 0 \text{ dado} \end{cases}$$

Teorema 12 (Krueger, 2017). Suponha que U, β e F (e portanto f) satisfaçam Hipótese 1 e Hipótese 2. Então, uma alocação $\{k_{t+1}\}_{t=0}^\infty$ que satisfaça as equações de Euler e a condição de transversalidade soluciona o problema sequencial do planejador social, para um dado k_0.

2.1 [⊠]

Considere a seguinte modificação no modelo de crescimento neoclássico determinístico. As pessoas da economia têm formação de hábitos de consumo, ou seja, a utilidade é dada por

$$u(\{c_t\}_{t=0}^{\infty}) = \sum_{t=0}^{\infty} \beta^t U(c_t, c_{t-1})$$

dado um certo c_{-1}, em que $U(c_t, c_{t-1}) = \ln c_t + \gamma \ln c_{t-1}$. Além disso, a função de produção é dada por $f(k) = Ak^{\alpha}$ e todo estoque de capital se deprecia a cada período. (Podemos ignorar o trabalho aqui). Desse modo, o problema do planejador de escolher a trajetória de consumo que maximiza o bem-estar do consumidor pode ser escrito como:

$$\begin{aligned} \max_{\{c_t, k_{t+1}\}_{t=0}^{\infty}} \sum_{t=0}^{\infty} \beta^t (\ln c_t + \gamma \ln c_{t-1}) \\ \text{s.a. } c_t + k_{t+1} \leq Ak_t^{\alpha}, \\ c_t, k_{t+1} \geq 0 \\ A > 0, \alpha \in (0,1) \\ \text{dados } k_0 > 0 \text{ e } c_{-1} > 0, \end{aligned}$$

em que $\beta \in (0,1)$ e $\gamma > 0$. Aqui, c_t representa o consumo da data t, k_t é o estoque de capital no começo do período t.

(a) Escreva a equação de Bellman associada ao problema sequencial acima.

Estratégia da Prova:

- Subseção 3.2.1 (Krueger, 2017).

Podemos reescrever o problema do planejador, considerando $\bar{k}_0 = k_0$ no equilíbrio e dado $U(c_t, c_{t-1}) = \ln c_t + \gamma \ln c_{t-1}$, como:

$$\begin{aligned} w(k_0, c_{-1}) &= \max_{\substack{\{c_t, k_{t+1}\}_{t=0}^{\infty} \\ 0 \leq k_{t+1} \leq f(k_t) \\ k_0 \text{ dado}}} \left\{ \sum_{t=0}^{\infty} \beta^t (\ln c_t + \gamma \ln c_{t-1}) \right\} \\ &= \max_{\substack{\{c_t, k_{t+1}\}_{t=0}^{\infty} \\ 0 \leq k_{t+1} \leq f(k_t) \\ k_0 \text{ dado}}} \left\{ \ln c_0 + \gamma \ln c_{-1} + \beta \sum_{t=1}^{\infty} \frac{\beta^t}{\beta} (\ln c_t + \gamma \ln c_{t-1}) \right\} \quad \left(\text{tirando } t = 0 \text{ do } \textstyle\sum\right) \\ &= \max_{\substack{0 \leq k_1 \leq f(k_0) \\ k_0 \text{ dado}}} \left\{ \ln c_0 + \gamma \ln c_{-1} + \beta \left[\max_{\substack{\{k_{t+1}\}_{t=1}^{\infty} \\ 0 \leq k_{t+1} \leq f(k_t) \\ k_1 \text{ dado}}} \left\{ \sum_{t=1}^{\infty} \beta^{t-1} (\ln c_t + \gamma \ln c_{t-1}) \right\} \right] \right\} \\ &= \max_{\substack{0 \leq k_1 \leq f(k_0) \\ k_0 \text{ dado}}} \left\{ \ln c_0 + \gamma \ln c_{-1} + \beta \left[\max_{\substack{\{k_{t+2}\}_{t=0}^{\infty} \\ 0 \leq k_{t+2} \leq f(k_{t+1}) \\ k_1 \text{ dado}}} \left\{ \sum_{t=0}^{\infty} \beta^{t} (\ln c_{t+1} + \gamma \ln c_{t}) \right\} \right] \right\} \end{aligned}$$

Assim,

$$w(k_0, c_{-1}) = \max_{\substack{0 \le k_1 \le f(k_0) \\ k_0 \text{ dado}}} \{\ln c_0 + \gamma \ln c_{-1} + \beta w(k_1, c_0)\}, \qquad \text{s.a.} \;\; c_t + k_{t+1} \le A k_t^\alpha.$$

Logo, o problema recursivo é dado pela equação de Bellman:

$$v(k, c_{-1}) = \max_{\substack{0 \le k' \le f(k) \\ k \text{ dado}}} \{\ln c + \gamma \ln c_{-1} + \beta v(k', c)\}, \qquad \text{s.a.} \;\; c + k' \le A k^\alpha. \tag{2.1.1}$$

A restrição $c + k' \le Ak^\alpha$ se dá a partir da Definição 10 (Krueger, 2017), em que $F(k_t) = c_t + k_{t+1} - (1-\delta)k_t$ e, como temos depreciação total em um período ($\delta = 1$), segue que

$$F(k_t) = c_t + k_{t+1}$$

Como, no equilíbrio, $f(k_t) = F(k_t) + (1-\delta)k_t$ e $\delta = 1$, então $f(k_t) = F(k_t)$. Pelo enunciado, temos que $f(k_t) = Ak_t^\alpha$ (sendo $A > 0$ constante) e, portanto,

$$\begin{aligned} f(k_t) &= F(k_t) \\ Ak_t^\alpha &= c_t + k_{t+1}, \end{aligned} \tag{2.1.2}$$

que pode ser reescrita como $c + k' = Ak^\alpha$. ■

(b) Seja $v^*(k_0, c_{-1})$ a função valor que resolve a equação funcional estabelecida no item anterior. Mostre que $v^*(k_0, c_{-1}) = E + F \ln k_0 + G \ln c_{-1}$ e que a trajetória ótima de capital $\{k_{t+1}^*\}_{t=0}^\infty$ satisfaz $\ln k_{t+1}^* = I + H \ln k_t^*$, em que E, F, G, H e I são constantes. Ache fórmulas explícitas para E, F, G, H e I em função dos parâmetros A, β, α e γ.

Estratégia da Prova:

- Subseção 3.2.3 (Krueger, 2017): *Guess-and-Verify* (pág. 45)
  - Aplicar suposta forma da função valor na equação de Bellman obtida no item (a).
  - Maximizar via CPO e obter k'.
  - Aplicar k' na equação de Bellman e reescrever no formato suposto.
  - Resolver sistema com as possíveis equivalências entre E, F, G e obter seus valores em função de A, α, β, γ e aplicá-los na funções valor e política.
  - Supor forma da função política e estabelecer equivalências com I, H.
  - Construir, via indução matemática, uma sequência de capital e calcular o capital estacionário $\bar{k}$ em que $t \to \infty$.

Daremos os palpites sobre os formatos da função valor, v, e da função política $k' = g(k)$ no equilíbrio. Suponha que solução de (2.1.1) tenha o formato

$$v(k, c_{-1}) = E + F \ln k + G \ln c_{-1}, \quad \forall k \in \mathbb{R}_+ \tag{2.1.3}$$

em que $E, F, G \in \mathbb{R}$ e são constantes.

Aplicando $v(k', c) = E + F \ln(k') + G \ln c$ (período seguinte do formato da função que supomos no início) no problema de otimização/equação de Bellman em (2.1.1) temos:

$$\max_{\substack{0 \le k' \le f(k) \\ k \text{ dado}}} \{\ln c + \gamma \ln c_{-1} + \beta[E + F \ln k' + G \ln c]\}, \quad \text{s.a.} \;\; c + k' \le Ak^\alpha. \tag{2.1.4}$$

e substituindo $c = Ak^\alpha - k'$, segue que

$$\max_{\substack{0\leq k'\leq f(k)\\ k \text{ dado}}} \{\ln(Ak^\alpha - k') + \gamma \ln c_{-1} + \beta[E + F\ln k' + G\ln(Ak^\alpha - k')]\}.$$

Note que c_{-1} não está em função de k', apenas de k e k_{-1}. Observe também que a função objetiva (log) é côncava, satisfaz Inada e não deve ter solução de canto. Portanto, por CPO:

$$\begin{aligned}
[k']: \quad 0 &= -\frac{1}{Ak^\alpha - k'} + \frac{\beta F}{k'} - \frac{\beta G}{Ak^\alpha - k'} \\
0 &= -\frac{1+\beta G}{Ak^\alpha - k'} + \frac{\beta F}{k'} \\
\frac{\beta F}{k'} &= \frac{1+\beta G}{Ak^\alpha - k'} \\
Ak^\alpha - k' &= \frac{(1+\beta G)k'}{\beta F} \\
k'\left(1 + \frac{1+\beta G}{\beta F}\right) &= Ak^\alpha \\
k'\left(\frac{1+\beta F+\beta G}{\beta F}\right) &= Ak^\alpha \\
k'_* &= \left(\frac{A\beta F}{1+\beta F+\beta G}\right)k^\alpha
\end{aligned} \tag{2.1.5}$$

Aplicando (2.1.5) na equação de Bellman (2.1.4)

$$v(k, c_{-1}) = \max_{\substack{0\leq k'\leq f(k)\\ k \text{ dado}}} \{\ln(Ak^\alpha - k') + \gamma \ln c_{-1} + \beta[E + F\ln k' + G\ln(Ak^\alpha - k')]\}$$

$$(c = Ak^\alpha - k')$$

$$\begin{aligned}
&\overset{(*)}{=} \ln\left(Ak^\alpha - \frac{A\beta F}{1+\beta F+\beta G}k^\alpha\right) + \gamma\ln c_{-1} \\
&\quad + \beta\left[E + F\ln\left(\frac{A\beta F}{1+\beta F+\beta G}k^\alpha\right) + G\ln\left(Ak^\alpha - \frac{A\beta F}{1+\beta F+\beta G}k^\alpha\right)\right] \\
&= \ln k^\alpha + \ln\left(A - \frac{A\beta F}{1+\beta F+\beta G}\right) + \gamma\ln c_{-1} \\
&\quad + \beta\left[E + F\ln k^\alpha + F\ln\left(\frac{A\beta F}{1+\beta F+\beta G}\right) + G\ln k^\alpha + G\ln\left(A - \frac{A\beta F}{1+\beta F+\beta G}\right)\right] \\
&= \alpha\ln k + \ln\left(\frac{A(1+\beta G)}{1+\beta F+\beta G}\right) + \gamma\ln c_{-1} \\
&\quad + \beta\left[E + \alpha F\ln k + F\ln\left(\frac{A\beta F}{1+\beta F+\beta G}\right) + \alpha G\ln k + G\ln\left(\frac{A(1+\beta G)}{1+\beta F+\beta G}\right)\right] \\
&= (1+\beta G)\ln\left(\frac{A(1+\beta G)}{1+\beta F+\beta G}\right) + \beta F\ln\left(\frac{A\beta F}{1+\beta F+\beta G}\right) + \beta E \\
&\quad + \alpha(1+\beta F+\beta G)\ln k + \gamma\ln c_{-1}
\end{aligned} \tag{2.1.6}$$

Em $(*)$, o max "sumiu" pela aplicação função política ótima, $\hat{g}(k) = k'_*$, que maximiza a função valor obtida em (2.1.4). Note que (2.1.6) é comparável à forma suposta no início do exercício, $v(k, c_{-1}) = E + F\ln k + G\ln c_{-1}$:

$$\underbrace{(1+\beta G)\ln\left(\frac{A(1+\beta G)}{1+\beta F+\beta G}\right) + \beta F\ln\left(\frac{A\beta F}{1+\beta F+\beta G}\right) + \beta E}_{E} + \underbrace{\alpha(1+\beta F+\beta G)\ln k}_{F\ln k} + \underbrace{\gamma\ln c_{-1}}_{G\ln c_{-1}}$$

Para verificar essas equivalências, solucionaremos o seguinte sistema:

$$\begin{cases} E &= (1+\beta G)\ln\left(\frac{A(1+\beta G)}{1+\beta F+\beta G}\right) + \beta F\ln\left(\frac{A\beta F}{1+\beta F+\beta G}\right) + \beta E \\ F &= \alpha(1+\beta F+\beta G) \\ G &= \gamma \end{cases}$$

Aplicando G em F, segue que

$$\begin{aligned} F &= \alpha + \alpha\beta F + \alpha\beta G \\ F - \alpha\beta F &= \alpha + \alpha\beta\gamma && (G=\gamma) \\ F(1-\alpha\beta) &= \alpha + \alpha\beta\gamma \\ F &= \frac{\alpha(1+\beta\gamma)}{1-\alpha\beta} && (2.1.7) \end{aligned}$$

Substituindo F e G em E:

$$\begin{aligned} E - \beta E &= (1+\beta G)\ln\left(\frac{A(1+\beta G)}{1+\beta F+\beta G}\right) + \beta F\ln\left(\frac{A\beta F}{1+\beta F+\beta G}\right) \\ E(1-\beta) &= (1+\beta\gamma)\ln\left(\frac{A(1+\beta\gamma)}{1+\beta\frac{\alpha(1+\beta\gamma)}{1-\alpha\beta}+\beta\gamma}\right) + \beta\frac{\alpha(1+\beta\gamma)}{1-\alpha\beta}\ln\left(\frac{A\beta\frac{\alpha(1+\beta\gamma)}{1-\alpha\beta}}{1+\beta\frac{\alpha(1+\beta\gamma)}{1-\alpha\beta}+\beta\gamma}\right) \\ &= (1+\beta\gamma)\ln\left(\frac{A(1+\beta\gamma)}{\frac{(1-\alpha\beta)+\beta\alpha(1+\beta\gamma)+(1-\alpha\beta)\beta\gamma}{1-\alpha\beta}}\right) + \beta\frac{\alpha(1+\beta\gamma)}{1-\alpha\beta}\ln\left(\frac{\frac{A\beta\alpha(1+\beta\gamma)}{1-\alpha\beta}}{\frac{(1-\alpha\beta)+\beta\alpha(1+\beta\gamma)+(1-\alpha\beta)\beta\gamma}{1-\alpha\beta}}\right) \\ &= (1+\beta\gamma)\ln\left(\frac{A(1+\beta\gamma)(1-\alpha\beta)}{1-\alpha\beta+\beta\alpha(1+\beta\gamma)+(1-\alpha\beta)\beta\gamma}\right) \\ &\quad + \frac{\beta\alpha(1+\beta\gamma)}{1-\alpha\beta}\ln\left(\frac{A\beta\alpha(1+\beta\gamma)}{1-\alpha\beta+\beta\alpha(1+\beta\gamma)+(1-\alpha\beta)\beta\gamma}\right) \\ &= (1+\beta\gamma)\ln\left(\frac{A(1+\beta\gamma)(1-\alpha\beta)}{\beta\alpha(1+\beta\gamma)+(1-\alpha\beta)(1+\beta\gamma)}\right) \\ &\quad + \frac{\beta\alpha(1+\beta\gamma)}{1-\alpha\beta}\ln\left(\frac{A\beta\alpha(1+\beta\gamma)}{\beta\alpha(1+\beta\gamma)+(1-\alpha\beta)(1+\beta\gamma)}\right) \\ &= (1+\beta\gamma)\ln\left(\frac{A(1+\beta\gamma)(1-\alpha\beta)}{(\alpha\beta+1-\alpha\beta)(1+\beta\gamma)}\right) + \frac{\beta\alpha(1+\beta\gamma)}{1-\alpha\beta}\ln\left(\frac{A\beta\alpha(1+\beta\gamma)}{(\alpha\beta+1-\alpha\beta)(1+\beta\gamma)}\right) \\ &= (1+\beta\gamma)\ln\left(A(1-\alpha\beta)\right) + \frac{\beta\alpha(1+\beta\gamma)}{1-\alpha\beta}\ln\left(A\beta\alpha\right) \\ E &= \frac{1}{1-\beta}\left[(1+\beta\gamma)\ln\left(A(1-\alpha\beta)\right) + \frac{\beta\alpha(1+\beta\gamma)}{1-\alpha\beta}\ln\left(A\beta\alpha\right)\right] && (2.1.8) \end{aligned}$$

Substituindo E, F, G na função valor, temos:

$$v^*(k, c_{-1}) = \frac{1}{1-\beta}\left[(1+\beta\gamma)\ln\left(A(1-\alpha\beta)\right) + \frac{\beta\alpha(1+\beta\gamma)}{1-\alpha\beta}\ln\left(A\beta\alpha\right)\right] + \frac{\alpha(1+\beta\gamma)}{1-\alpha\beta}\ln k + \gamma\ln c_{-1}$$

Aplicando F e G na função política (2.1.5), segue

$$\begin{aligned} k'_* &= \left(\frac{A\beta F}{1+\beta F+\beta G}\right)k^\alpha = \left(\frac{A\beta\frac{\alpha(1+\beta\gamma)}{1-\alpha\beta}}{1+\beta\frac{\alpha(1+\beta\gamma)}{1-\alpha\beta}+\beta\gamma}\right)k^\alpha \\ &= \left(\frac{\frac{A\beta\alpha(1+\beta\gamma)}{1-\alpha\beta}}{\frac{1-\alpha\beta+\beta\alpha(1+\beta\gamma)+\beta\gamma(1-\alpha\beta)}{1-\alpha\beta}}\right)k^\alpha = \left(\frac{A\beta\alpha(1+\beta\gamma)}{(\alpha\beta+1-\alpha\beta)(1+\beta\gamma)}\right)k^\alpha \\ &= (A\beta\alpha)k^\alpha && (2.1.9) \end{aligned}$$

Suponha a forma da função política com as constantes H, I:

$$\ln k^*_{t+1} = I + H \ln k^*_t \tag{2.1.10}$$

Por (2.1.9), sabemos que $k^*_{t+1} = (A\beta\alpha)(k^*_t)^\alpha$ e, ao tomarmos o ln, obtemos:

$$\begin{aligned}\ln(k^*_{t+1}) &= \ln[(A\beta\alpha)(k^*_t)^\alpha] \\ &= \ln(A\beta\alpha) + \ln(k^*_t)^\alpha \\ &= \ln(A\beta\alpha) + \alpha \ln(k^*_t)\end{aligned}$$

Portanto, fazendo a equivalência com os termos de (2.1.11) (e ignorando que não tem $\ln(\cdot)$ no lado esquerdo da expressão), concluímos que:

$$\begin{cases} I &= \ln(A\beta\alpha) \\ H &= \alpha \end{cases} \tag{2.1.11}$$

Agora, a partir a da função política em (2.1.9) e usando indução matemática, encontraremos a sequência de capital solução de (2.1.1):

$$\begin{aligned}k_1 &= g(k_0) = (A\beta\alpha)k_0^\alpha \\ k_2 &= g(k_1) = (A\beta\alpha)k_1^\alpha = (A\beta\alpha)[(A\beta\alpha)k_0^\alpha]^\alpha = (A\beta\alpha)^{1+\alpha}k_0^{\alpha^2} \\ k_3 &= g(k_2) = (A\beta\alpha)k_2^\alpha = (A\beta\alpha)[(A\beta\alpha)^{1+\alpha}k_0^{\alpha^2}]^\alpha = (A\beta\alpha)^{1+\alpha+\alpha^2}k_0^{\alpha^3} \\ &\vdots \\ k^*_{t+1} &= g(k_t) = (A\beta\alpha)^{\sum_{i=0}^t \alpha^t}.k_0^{\alpha^t} = (A\beta\alpha)^{\frac{1}{1-\alpha}}k_0^{\alpha^t} \end{aligned} \tag{2.1.12}$$

Agora, calcularemos o capital estacionário $\bar{k}$ a partir de k_{t+1} obtido em (2.1.12)

$$\bar{k} = \lim_{t\to\infty} k_{t+1} = \lim_{t\to\infty} (A\beta\alpha)^{\frac{1}{1-\alpha}}k_0^{\alpha^t}$$

Note que, por $\alpha \in (0,1)$, quando $t \to \infty$, $\alpha^t \to 0$, logo

$$\lim_{t\to\infty} k_0^{\alpha^t} = 1$$

e, portanto,

$$\bar{k} = (A\beta\alpha)^{\frac{1}{1-\alpha}}.$$

Note que

$$\begin{aligned}\bar{k} &= g(\bar{k}) \\ (A\beta\alpha)^{\frac{1}{1-\alpha}} &= (A\beta\alpha)[(A\beta\alpha)^{\frac{1}{1-\alpha}}]^\alpha \\ &= (A\beta\alpha)^{1+\frac{\alpha}{1-\alpha}} \\ &= (A\beta\alpha)^{\frac{1-\alpha+\alpha}{1-\alpha}} \\ &= (A\beta\alpha)^{\frac{1}{1-\alpha}}.\end{aligned}$$

Portanto, $\bar{k}$ é ponto fixo de $g(k)$ e a sequência construída em (2.1.12) é solução para o problema sequencial do planejador social. ∎

2.2 [☒]

Considere novamente o modelo de crescimento neoclássico. Em que o planejador escolhe uma sequência de capital para $t = 0, 1, 2, \dots$ para maximizar

$$\sum_{t=0}^{\infty} \beta^t U(c_t), \qquad \beta \in (0,1)$$

sujeito a uma sequência de restrições de recursos dada por

$$c_t + k_{t+1} \leq f(k_t) + (1-\delta)k_t, \qquad \delta \in (0,1)$$

com uma condição inicial $k_0 > 0$.

(a) Seja $v(k)$ a função valor deste problema. Escreva e explique a equação de Bellman que determina $v(k)$.

Estratégia da Prova:

- Subseções 3.2.1 e 3.2.2 (Krueger, 2017).
- Note que $\delta \neq 1$. Logo, para $c_t \geq 0$, precisamos de $k_{t+1} \leq f(k_t) + (1-\delta)k_t$

Note que, no equilíbrio, a sequência de restrições de recursos é válida para a igualdade e, logo, sendo $\delta \in (0,1)$,

$$\begin{aligned} c_t + k_{t+1} &= f(k_t) + (1-\delta)k_t \\ c_t &= f(k_t) + (1-\delta)k_t - k_{t+1}. \end{aligned}$$

Normalizando $n_t = 1$, aplicando c_t na Formulação Sequencial do Problema do Planejador Social e tomando $k_0 = \bar{k}_0$, obtemos:

$$\begin{aligned} w(\bar{k}_0) &= \max_{\substack{\{c_t, k_{t+1}\}_{t=0}^{\infty} \\ k_0 \text{ dado}}} \left\{ \sum_{t=0}^{\infty} \beta^t U(c_t) \right\} \\ &= \max_{\substack{\{k_{t+1}\}_{t=0}^{\infty} \\ k_0 \text{ dado}}} \left\{ \sum_{t=0}^{\infty} \beta^t U[f(k_t) + (1-\delta)k_t - k_{t+1}] \right\} \qquad (2.2.1) \\ &= \max_{\substack{\{k_{t+1}\}_{t=0}^{\infty} \\ k_0 \text{ dado}}} \left\{ U[f(k_0) + (1-\delta)k_0 - k_1] + \beta \sum_{t=1}^{\infty} \frac{\beta^t}{\beta} U[f(k_t) + (1-\delta)k_t - k_{t+1}] \right\} \\ &\qquad\qquad (\text{tirando } t = 0 \text{ do } \textstyle\sum) \\ &= \max_{\substack{k_1 \\ k_0 \text{ dado}}} \left\{ U[f(k_0) + (1-\delta)k_0 - k_1] + \beta \left[\max_{\substack{\{k_{t+1}\}_{t=1}^{\infty} \\ k_1 \text{ dado}}} \left\{ \sum_{t=1}^{\infty} \beta^{t-1} U[f(k_t) + (1-\delta)k_t - k_{t+1}] \right\} \right] \right\} \\ &= \max_{\substack{k_1 \\ k_0 \text{ dado}}} \left\{ U[f(k_0) + (1-\delta)k_0 - k_1] + \beta \left[\max_{\substack{\{k_{t+2}\}_{t=0}^{\infty} \\ k_1 \text{ dado}}} \left\{ \sum_{t=0}^{\infty} \beta^t U[f(k_{t+1}) + (1-\delta)k_{t+1} - k_{t+2}] \right\} \right] \right\} \\ &= \max_{\substack{k_1 \\ k_0 \text{ dado}}} \left\{ U[f(k_0) + (1-\delta)k_0 - k_1] + \beta w(k_1) \right\}, \quad \text{s.a. } 0 \leq k_1 \leq f(k_0) + (1-\delta)k_0. \end{aligned}$$

Portanto, o problema recursivo é dado pela equação de Bellman:

$$v(k) = \max_{\substack{0 \leq k' \leq f(k)+(1-\delta)k \\ k \text{ dado}}} \left\{ U[f(k) + (1-\delta)k - k'] + \beta v(k') \right\}. \tag{2.2.2}$$

A equação de Bellman é um equação funcional em que a solução é uma função, ao invés de ser um número ou um vetor. Ela postula que a utilidade ao longo da vida do agente representativo é dado pela utilidade que o agente recebe hoje, $U[f(k) + (1-\delta)k - k']$, mais a utilidade trazida a valor presente das utilidades do próximo período em diante, $\beta v(k')$. Então, esta formulação considera o trade-off do planejador: maior consumo hoje $\times$ maior estoque de capital. (Krueger, 2017, pág. 43) ■

(b) Para este e os próximos itens suponha que

$$U(c) = \begin{cases} \dfrac{c^{1-\sigma} - 1}{1-\sigma}, & \text{se } \sigma > 0, \sigma \neq 1 \\ \ln(c), & \text{se } \sigma = 1 \end{cases},$$

e que $f(k) = zk^{\alpha}$ com $\alpha \in (0,1)$ e $z > 0$. Resolva o modelo para os valores de steady state c^* e k^*. (Dica: Você pode fazer isso utilizando os valores de c^* e k^* constantes ($c = c' = c^*$ ou $k = k' = k^*$) na equação de Euler.) Qual a razão capital/produto em steady state? E consumo/produto?

Estratégia da Prova:

- Subseção 3.2.4 (Krueger, 2017) - Solução do Problema Sequencial por Equação de Euler e Condições de Transversalidade (TVC).
- Usar relação de estado estacionário $k = k' = k^*$ na equação de Euler para encontrar k^*
- A partir de k^*, encontrar c^* e y^*.

Utilizaremos o problema na formulação sequencial (2.2.1), dado por:

$$w(\bar{k}_0) = \max_{\{k_{t+1}\}_{t=0}^{\infty}} \left\{ \sum_{t=0}^{\infty} \beta^t U[f(k_t) + (1-\delta)k_t - k_{t+1}] \right\}$$

Supondo Hipótese 1 e Hipótese 2 para as funções U, β, F e f, temos que as restrições são inativas, ou seja, $0 < k_{t+1} < f(k_t) + (1-\delta)k_t$. Portanto, podemos utilizar CPO's para solucionar o problema (são necessárias e suficientes).

Primeiro, note que dentro do somatório temos, para todo t,

$$\ldots + \beta^t U[(k_t) + (1-\delta)k_t - k_{t+1}] + \beta^{t+1} U[(k_{t+1}) + (1-\delta)k_{t+1} - k_{t+2}] + \ldots$$

logo, para derivarmos o somatório por k_{t+1}, precisamos considerar 2 termos que possuem k_{t+1}.

Pelas CPO's:

$$[k_{t+1}]: -\beta^t U'[f(k_t) + (1-\delta)k_t - k_{t+1}] + \beta^{t+1} U[f(k_{t+1}) + (1-\delta)k_{t+1} - k_{t+2}][f'(k_{t+1}) + (1-\delta)] = 0$$

$$\begin{aligned} \beta^t U'[f(k_t) + (1-\delta)k_t - k_{t+1}] &= \beta^{t+1} U'[f(k_{t+1}) + (1-\delta)k_{t+1} - k_{t+2}][f'(k_{t+1}) + (1-\delta)] \\ U'[f(k_t) + (1-\delta)k_t - k_{t+1}] &= \beta U'[f(k_{t+1}) + (1-\delta)k_{t+1} - k_{t+2}][f'(k_{t+1}) + (1-\delta)], \end{aligned} \tag{2.2.3}$$

para $t = 0, 1, 2, ...$

Note que, no estado estacionário (steady state), $k_t = k_{t+1} = k_*$, portanto:

$$
\begin{aligned}
U'[f(k_*) + (1-\delta)k_* - k_*] &= \beta U'[f(k_*) + (1-\delta)k_* - k_*][f'(k_*) + (1-\delta)] \\
1 &= \beta[f'(k_*) + (1-\delta)] && \text{(dividindo por } U'\text{)} \\
\frac{1}{\beta} &= (zk_*^\alpha)' + (1-\delta) && (f(k) = zk^\alpha) \\
\frac{1}{\beta} &= \alpha z k_*^{\alpha-1} + (1-\delta) \\
\frac{1}{\beta} - (1-\delta) &= \alpha z k_*^{\alpha-1} \\
\frac{1-\beta+\delta\beta}{\beta\alpha z} &= k_*^{\alpha-1} \\
\frac{\beta\alpha z}{1-\beta+\delta\beta} &= k_*^{1-\alpha} && \text{(elevado a } -1\text{, pois } \alpha - 1 < 0\text{)} \\
\left(\frac{\beta\alpha z}{1-\beta+\delta\beta}\right)^{\frac{1}{1-\alpha}} &= \left(k_*^{1-\alpha}\right)^{\frac{1}{1-\alpha}} \\
k_* &= \left(\frac{\beta\alpha z}{1-\beta+\delta\beta}\right)^{\frac{1}{1-\alpha}}. && (2.2.4)
\end{aligned}
$$

Agora, precisamos encontrar o produto em estado estacionário, y_*, tal que

$$
\begin{aligned}
y_* = f(k_*) = zk_*^\alpha = z\left[\left(\frac{\beta\alpha z}{1-\beta+\delta\beta}\right)^{\frac{1}{1-\alpha}}\right]^\alpha && \text{(usando (2.2.4))} \\
= \left(\frac{\beta\alpha}{1-\beta+\delta\beta}\right)^{\frac{\alpha}{1-\alpha}} z^{1+\frac{\alpha}{1-\alpha}} \\
= \left(\frac{\beta\alpha}{1-\beta+\delta\beta}\right)^{\frac{\alpha}{1-\alpha}} z^{\frac{1-\alpha+\alpha}{1-\alpha}} \\
= \left(\frac{\beta\alpha}{1-\beta+\delta\beta}\right)^{\frac{\alpha}{1-\alpha}} z^{\frac{1}{1-\alpha}} && (2.2.5)
\end{aligned}
$$

Portanto, a razão capital/produto no estado estacionário é

$$
\frac{k_*}{y_*} = \frac{\left(\frac{\beta\alpha}{1-\beta+\delta\beta}\right)^{\frac{1}{1-\alpha}} z^{\frac{1}{1-\alpha}}}{\left(\frac{\beta\alpha}{1-\beta+\delta\beta}\right)^{\frac{\alpha}{1-\alpha}} z^{\frac{1}{1-\alpha}}} = \left(\frac{\beta\alpha}{1-\beta+\delta\beta}\right)^{\frac{1-\alpha}{1-\alpha}} z^0 = \frac{\beta\alpha}{1-\beta+\delta\beta} \tag{2.2.6}
$$

Note que, como $k = k' = k_*$ no estado estacionário, o consumo é dado por

$$
c_* = f(k_*) + (1-\delta)k_* - k_* = f(k_*) - \delta k_* = y_* - \delta k_* \tag{2.2.7}
$$

Logo, a razão consumo/produto é

$$
\begin{aligned}
\frac{c_*}{y_*} &= \frac{y_* - \delta k_*}{y_*} = 1 - \delta\frac{k_*}{y_*} = 1 - \delta\frac{\beta\alpha}{1-\beta+\delta\beta} && \text{(usando (2.2.6))} \\
&= \frac{(1-\beta+\delta\beta) - \delta\beta\alpha}{1-\beta+\delta\beta} = \frac{1-\beta+\delta\beta(1-\alpha)}{1-\beta+\delta\beta}.
\end{aligned}
$$

$\blacksquare$

(c) Descreva em detalhes os passos de um algoritmo para computar as funções valor e política associadas ao problema do planejador.

Podemos computar as funções valor e política numericamente via programação dinâmica. Para isso, definiremos uma função v(·) de dimensão infinita, e a preencheremos com os resultados das iterações da função valor. O algoritmo pode ser descrito em 5 passos, são eles:

(i) **Parametrizar a economia.**
Neste tópico definiremos os parâmetros e as funções a serem utilizadas. Então, para o exercício proposto, neste item definiremos a função utilidade como uma CRRA:

$$U(c) = \begin{cases} \dfrac{c^{1-\sigma} - 1}{1-\sigma}, & \text{se } \sigma > 0, \sigma \neq 1 \\ \ln(c), & \text{se } \sigma = 1 \end{cases},$$

E a função de tecnologia de produção:

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

Além das funções, precisamos atribuir valores para os parâmetros a serem utilizados, a partir da calibração. Como exemplo, utilizaremos como parâmetros:

- $\alpha = 0.3$
- $\beta = 0.96$
- $\sigma = 2.5$
- $\delta = 0.02$
- $z = 1$

Além disso, precisamos definir um grid para os possíveis valores de k e k', nele estará o ponto fixo que o algoritmo procurará. Portanto a definição correta dos pontos do grid é crucial para o obtenção da solução do algoritmo. Um grid pode ser definido da seguinte forma:

$$K = \{\underline{k}, k_1, k_2, k_3, ..., \bar{k}\}$$

em que $\underline{k} < k_1 < k_2 < k_3 < ... < \bar{k}$.

(ii) **Defina um chute Inicial.**
Agora, faremos um chute inicial para a primeira função valor a entrar no loop. Podemos escolher qualquer valor, pois as propriedades da iteração da função valor garantem que haverá convergência do algoritmo para o ponto fixo, qualquer que seja o chute inicial. Porém, quanto mais distante do ponto fixo o chute inicial estiver, seu algoritmo fará mais iterações e demorará mais tempo para chegar ao resultado. Desta forma, é aconselhável dar um "chute educado". Um exemplo de chute inicial pode ser:

$$v_0(k) = 0 \qquad \forall k \in K$$

(iii) **Resolva o modelo.**
Este tópico representa a procura do algoritmo pela solução ótima. Neste caso, operaremos

pelo método da força bruta, utilizaremos a capacidade de processamento dos computadores para realizar o número necessário de iterações da função valor para procurar em todos os k' possíveis a função v(·) para cada k, até esgotar todas as entradas do grid. Então, resolveremos o seguinte problema de maximização

$$v_1(k) = \max_{\substack{0 \leq k' \leq f(k) \\ k' \in K}} \{U(f(k) - k') + \beta v_0(k')\}$$

e guardaremos v_1 e a escolha da função política $g_1(k) = k'$.

(iv) **Repita o passo anterior para as funções valor subsequentes.**
Agora faremos o mesmo processo do passo anterior e encontraremos $v_2, v_3, v_4, v_5, ..., v_n$ até que

$$\|v_n - v_{n-1}\| = 0$$

Assim, teremos que fazer as iterações até que encontremos a iteração convergida, ou seja, a iteração que satisfaça $\|v_n - v_{n-1}\| = 0$. Porém, note que tal norma pode ser muito pequena mas nunca será exatamente zero. Então, ao invés de zero, devemos definir um valor bastante pequeno e próximo de zero como critério de convergência da função valor.

(v) $\boldsymbol{v = v_n}$ **e** $\boldsymbol{g(k) = g_n(k)}$
No último passo basta igualar a função v à v_n e a função política à $g_n(k)$ e encontraremos a solução do problema de programação dinâmica.

(d) Sejam $z = 1$, $\alpha = 0.3$, $\beta = 1/1.05$, $\delta = 0.05$ e $\sigma = 1$. Usando estes valores de parâmetros, discretize o espaço de estados para o capital com $n = 1001$ pontos, calcule e plote a função valor neste grid. Seja $c(k)$ a função política para o consumo. Calcule e plote $c(k)$. Como o comportamento de consumo e poupança implicado por essa função se compara com o steady state do item (b)?[2]

Usando os parâmetros z, α, β, δ dados, em k_*, y_*, c_*, obtidos respectivamente em (2.2.4), (2.2.5) e (2.2.7), temos

$$k_* = \left(\frac{\beta\alpha z}{1-\beta+\delta\beta}\right)^{\frac{1}{1-\alpha}} = \left(\frac{\frac{1}{1.05}0.3(1)}{1-\frac{1}{1.05}+0.05\frac{1}{1.05}}\right)^{\frac{1}{1-0.3}} = 4.8$$

$$y_* = \left(\frac{\beta\alpha}{1-\beta+\delta\beta}\right)^{\frac{\alpha}{1-\alpha}} z^{\frac{1}{1-\alpha}} = \left(\frac{\frac{1}{1.05}0.3}{1-\frac{1}{1.05}+0.05\frac{1}{1.05}}\right)^{\frac{0.3}{1-0.3}} 1^{\frac{1}{1-0.3}} = 1.6$$

$$c_* = y_* - \delta k_* = 4.8 - (0.05)1.6 = 1.36$$

Note também que:

$$\frac{c^*}{y^*} = 0.85 \qquad \text{e} \qquad \frac{\delta k^*}{y^*} = 0.15$$

ou seja, 85% é consumido e 15% é investido/poupado no estado estacionário.

[2]Dica: Seu grid deve ter o ponto k^* em seu interior. Por exemplo, você pode usar um grid $K = \{0.7k^* < ... < k^* < ... < 1.3k^*\}$. Note que dado um n a escolha do grid pode não gerar uma função política muito suave. Faça experimentos em que você altera n e os limites inferiores e superiores do grid para observar esse fato.

Para resolver numericamente a função valor, como $\sigma = 1$, precisamos aplicar $U(c) = \ln c$ e $f(k) = zk^\alpha$ na função valor (2.2.2), obtida no item (a):

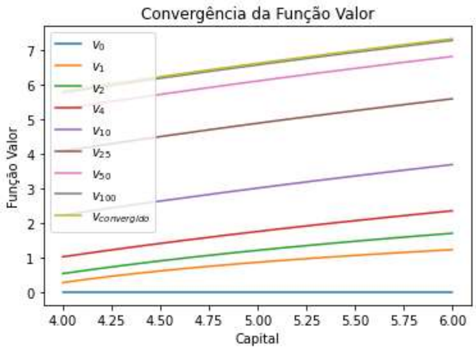
$$v(k) = \max_{\substack{0 \leq k' \leq zk^\alpha + (1-\delta)k \\ k \text{ dado}}} \left\{ \ln[zk^\alpha + (1-\delta)k - k'] + \beta v(k') \right\}. \tag{2.2.8}$$

```python
# Fazendo as suposições iniciais
k_grid = np.linspace(4, 6, 201)  # Possíveis valores de k e k'
beta = 1 / 1.05
alpha = 0.3
delta = 0.05
sigma = 1  # Para itens (c) e (d), usar 1; para item (e), usar 0.5 e 2
z = 1
n_k = len(k_grid)  # Tamanho do vetor de valores de k e k'

# Criando lista de arrays de função valor e função política
v0 = np.zeros(n_k)
vn = [v0]

g0 = np.zeros(n_k)
gn = [g0]

# Loop das iterações enquanto (while) é menor do que dada dist ncia
tol_norma = 1e-5  # Toler ncia de dist ncia entre funções valor (0.00001)
norma = np.inf  # Valor inicial da norma = infinito
n = 0  # Contador de iterações

while norma > tol_norma:
    # Aplicar a cada iteração Operador de Bellman em objetos genéricos Tv e Tg
    Tv = np.zeros(n_k)
    Tg = np.zeros(n_k)
    f_obj = np.zeros((n_k, n_k))
    n += 1

    for i in range(n_k):
        for j in range(n_k):
            if k_grid[j] >= 0 and k_grid[j] <= z*k_grid[i]**alpha + (1 - delta)*k_grid[i]:
                if sigma == 1:  # Utilidade na forma log
                    f_obj[i,j] = log(z*k_grid[i]**alpha + (1 - delta)*k_grid[i] - k_grid[j]) + beta*vn[n-1][j]
                else:  # Utilidade na forma de razão
                    f_obj[i,j] = ((z*k_grid[i]**alpha + (1 - delta)*k_grid[i] - k_grid[j])**(1 - sigma) - 1) / (1 - sigma) + beta*vn[n-1][j]
            else:
                f_obj[i,j] = - np.inf
        Tv[i] = np.max(f_obj[i,:])
        Tg[i] = np.argmax(f_obj[i,:])

    # Quando acabar loop de linha, jogar função valor em vn e política em gn
    vn.append(Tv)
    gn.append(Tg)
    norma = max(abs(vn[n] - vn[n-1]))

# Trocando índice em gn por valores de k'
for funcao in gn:
    for i in range(len(funcao)):
        funcao[i] = k_grid[int(funcao[i])]
```

Geradas as funções valor e política de capital, faremos as visualizações das convergências de acordo com o número de iterações:

```python
# Visualização gráfica da converg ncia da função valor
fig, ax = plt.subplots()

```

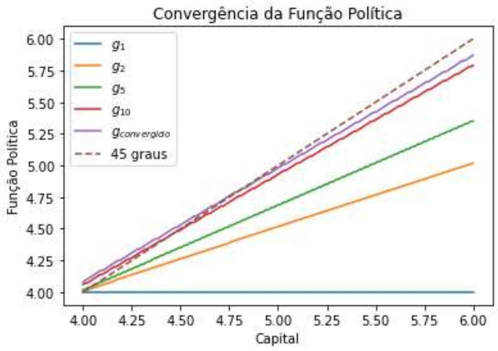
```python
# Inclusão de cada função valor no gráfico
ax.plot(k_grid, vn[0], label='$v_0$')
ax.plot(k_grid, vn[1], label='$v_1$')
ax.plot(k_grid, vn[2], label='$v_2$')
ax.plot(k_grid, vn[4], label='$v_4$')
ax.plot(k_grid, vn[10], label='$v_{10}$')
ax.plot(k_grid, vn[25], label='$v_{25}$')
ax.plot(k_grid, vn[50], label='$v_{50}$')
ax.plot(k_grid, vn[100], label='$v_{100}$')
ax.plot(k_grid, vn[n], label='$v_{convergido}$')

# Legendas
ax.set_xlabel('Capital')
ax.set_ylabel('Função Valor')
ax.set_title('Converg ncia da Função Valor')
ax.legend()

plt.show()
```

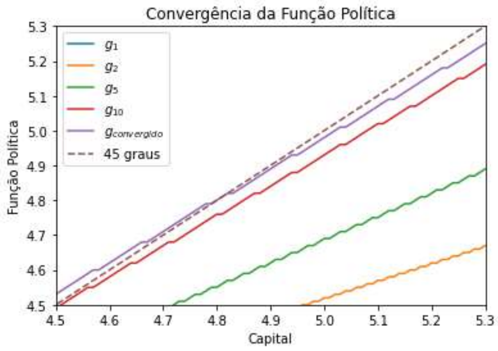



```python
# Visualização gráfica da converg ncia da função política
fig, ax = plt.subplots()

# Inclusão de cada função valor no gráfico
ax.plot(k_grid, gn[1], label='$g_1$')
ax.plot(k_grid, gn[2], label='$g_2$')
ax.plot(k_grid, gn[3], label='$g_5$')
ax.plot(k_grid, gn[10], label='$g_{10}$')
ax.plot(k_grid, gn[n], label='$g_{convergido}$')
ax.plot(k_grid, k_grid, '--', label='45 graus')

# Limites do gráfico
# ax.set_ylim([4.5, 5.3])  # tamanho mínimo e máximo vertical
# ax.set_xlim([4.5, 5.3])  # tamanho mínimo e máximo horizontal

# Legendas
ax.set_xlabel('Capital')
ax.set_ylabel('Função Política')
ax.set_title('Converg ncia da Função Política')
ax.legend()

plt.show()
```

Os gráficos acima mostram a convergência da função política, sendo que o da direita é um zoom do gráfico da esquerda. Nelas, vemos que a reta de 45 graus, que corresponde os pontos em que $k = k'$, cruza a função política de capital $g(k)$ próximo a 4.8 (valor calculado acima para o capital estacionário). Podemos também encontrar numericamente o capital estacionário:

```python
p = int(np.round(np.random.uniform(0, n_k - 1, size = 1), 0))  # aleatório
print("Índice inicial do capital (aleatorizado):", p)  # Índice randomizado

while k_grid[p] != gn[n][p]:  # até termos k = k'
```



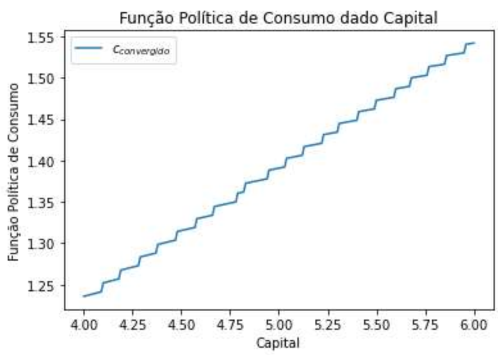



```
    p = np.where(k_grid == gn[n][p])[0][0]  # Aplica o índice de k' em k

print("Índice do capital estacionário:", p)  # Índice do capital estacionário
g_ss = k_grid[p]
print("Capital estacionário k* =", np.round(g_ss, 3))

# Consumo estacionário c* - a partir do k*
c_ss = z * k_grid[p]**alpha + (1-delta)*k_grid[p] - k_grid[p]
print("Consumo estacionário c* =", np.round(c_ss, 3))
```

```
Índice inicial do capital (aleatorizado): 104
Índice do capital estacionário: 82
Capital estacionário k* = 4.82
Consumo estacionário c* = 1.362
```

Note que o capital estacionário encontrado é bastante próximo ao valor teórico de 4.8 (calculado acima). Inserir mais dos que os 201 pontos definidos para k e/ou diminuir o tamanho do intervalo pode auxiliar numa melhor precisão deste valor. Agora, visualizaremos a função política de consumo a partir da função política do capital convergida:

```
# Calcular a função consumo c = f(k) + (1 - \delta) k - k'
cn = z * k_grid**alpha + (1-delta)*k_grid - gn[n]

# Visualização gráfica da função política de consumo
fig, ax = plt.subplots()

# Inclusão de cada função valor no gráfico
ax.plot(k_grid, cn, label='$c_{convergido}$')

# Legendas
ax.set_xlabel('Capital')
ax.set_ylabel('Função Política de Consumo')
ax.set_title('Função Política de Consumo dado Capital')
ax.legend()

plt.show()
```



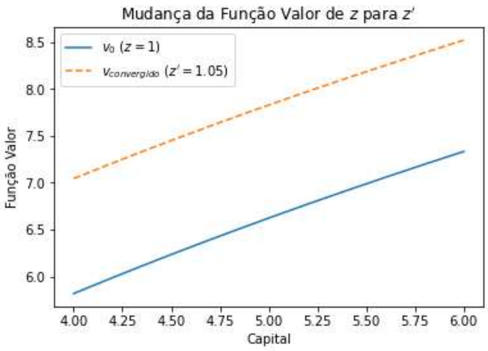
■

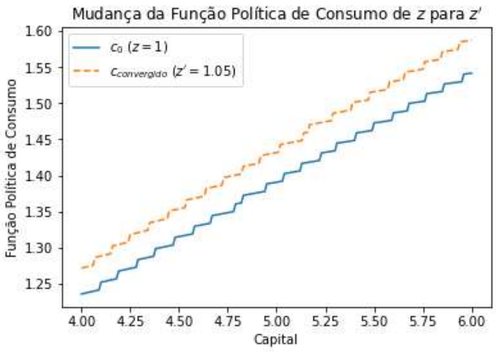
(e) Suponha que a economia está em steady state em $t = 0$. E há uma mudança permanente de $z = 1$ para $z' = 1.05$. Calcule e plote a nova função valor associada com z'. Compare ela com a função valor encontrada em (c). Calcule e plote a dinâmica de transição do capital e consumo dessa economia para o ajuste ao novo steady state.

Nesta seção, usaremos as funções convergidas (obtidas no item anterior) como "ponto de partida" para a nova economia, em que $z = 1$ se torna $z' = 1.05$.

```
# Fazendo a alteração em z
z = 1.05

# Funções iniciais são as convergidas na economia anterior
vn = [vn[n]]
gn = [gn[n]]

norma = np.inf  # Valor inicial da norma = infinito
n = 0  # Contador de iterações

while norma > tol_norma:
    # Aplicar a cada iteração Operador de Bellman em objetos genéricos Tv e Tg
    Tv = np.zeros(n_k)
    Tg = np.zeros(n_k)
    f_obj = np.zeros((n_k, n_k))
    n += 1

    for i in range(n_k):
        for j in range(n_k):
            if k_grid[j] >= 0 and k_grid[j] <= z*k_grid[i]**alpha + (1 - delta)*k_grid[i]:
                if sigma == 1:  # Utilidade na forma log
                    f_obj[i,j] = log(z*k_grid[i]**alpha + (1 - delta)*k_grid[i] - k_grid[j]) + beta*vn[n-1][j]
                else:  # Utilidade na forma de razão
                    f_obj[i,j] = ((z*k_grid[i]**alpha + (1 - delta)*k_grid[i] - k_grid[j])**(1 - sigma) - 1) / (1 - sigma) + beta*vn[n-1][j]
            else:
                f_obj[i,j] = - np.inf
        Tv[i] = np.max(f_obj[i,:])
        Tg[i] = np.argmax(f_obj[i,:])

    # Quando acabar loop de linha, jogar função valor em vn e política em gn
    vn.append(Tv)
    gn.append(Tg)
    norma = max(abs(vn[n] - vn[n-1]))


# Trocando índice em gn por valores de k' (retirando o 1   - já trocado)
for j in range(1, len(gn)):
    for i in range(len(gn[int(j)])):
        gn[int(j)][i] = k_grid[int(gn[int(j)][i])]


# Visualização gráfica da converg ncia da função valor
fig, ax = plt.subplots()

# Inclusão de cada função valor no gráfico
ax.plot(k_grid, vn[0], label='$v_0\ (z = 1)$')
ax.plot(k_grid, vn[n], "--", label='$v_{convergido}\ (z^\prime = 1.05)$')

```

```
# Legendas
ax.set_xlabel('Capital')
ax.set_ylabel('Função Valor')
ax.set_title('Mudança da Função Valor de $z$ para $z^\prime$')
ax.legend()

plt.show()


# Calcular o consumo c = f(k) + (1 - \delta) k - k'
c0 = cn  # função política de consumo anterior
cn = z * k_grid**alpha + (1-delta)*k_grid - gn[n]  # função política de consumo nova

# Visualização gráfica da função política de consumo
fig, ax = plt.subplots()

# Inclusão de cada função valor no gráfico
ax.plot(k_grid, c0, label='$c_0\ (z = 1)$')
ax.plot(k_grid, cn, "--", label='$c_{convergido}\ (z^\prime = 1.05)$')

# Legendas
ax.set_xlabel('Capital')
ax.set_ylabel('Função Política de Consumo')
ax.set_title('Mudança da Função Política de Consumo de $z$ para $z^\prime$')
ax.legend()

plt.show()
```



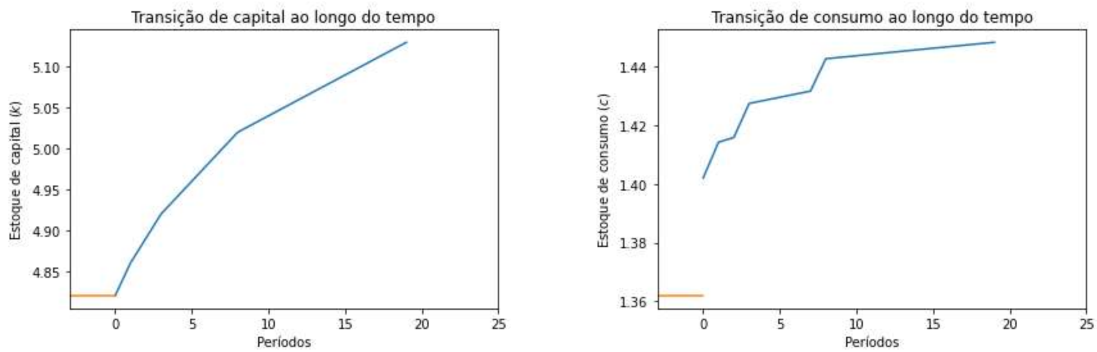



```
# Encontrar numericamente o capital estocástico k*  e guardar variáveis de transição
print("Índice inicial do capital (do k* anterior):", p)  # Índice randomizado
print("Capital inicial (k* de z = 1) =", np.round(k_grid[p], 3))

g_transicao = []
c_transicao = []
t_ss = 0  # Contador de períodos para estado estacionário

while k_grid[p] != gn[n][p]:  # até termos k = k'
    aux = p
    g_transicao.append(k_grid[p])
    p = np.where(k_grid == gn[n][p])[0][0]  # Aplica o índice de k' em k
    c_transicao.append(z * k_grid[aux]**alpha + (1-delta)*k_grid[aux] - k_grid[p])
    t_ss += 1

print("Foram necessários", t_ss, "períodos para atingir o novo estado estacionário.")
print("Índice do capital estacionário (novo):", p)  # Índice do capital estacionário
g_ss2 = k_grid[p]
print("Capital estacionário k* (z' = 1.05) =", np.round(g_ss2, 3))

# Consumo estacionário c* - a partir do k*
c_ss2 = z * k_grid[p]**alpha + (1-delta)*k_grid[p] - k_grid[p]
print("Consumo estacionário c* =", np.round(c_ss2, 3))
```

```
Índice inicial do capital (do k* anterior): 82
Capital inicial (k* de z = 1) = 4.82
Foram necessários 20 períodos para atingir o novo estado estacionário.
Índice do capital estacionário (novo): 114
Capital estacionário k* (z‘ = 1.05) = 5.14
Consumo estacionário c* = 1.459
```

A mudança de $z = 1$ para $z' = 1.05$ faz aumentar o capital estacionário de 4.8 para 5.1, e o consumo estacionário de 1.36 para 1.46 (as curvas deslocam-se para cima). Também podemos verificar como se deu a transição destes valores ao longo do tempo:

```python
# Visualização da transição do capital
fig, ax = plt.subplots()

# Inclusão de cada função valor no gráfico
ax.plot(range(len(g_transicao)), g_transicao)
ax.plot([-3, 0], [g_ss, g_ss])

# Legendas
ax.set_xlabel('Períodos')
ax.set_ylabel('Estoque de capital ($k$)')
ax.set_title('Transição de capital ao longo do tempo')
ax.set_xlim(-3, 25)

plt.show()


# Visualização da transição do consumo
fig, ax = plt.subplots()

# Inclusão de cada função valor no gráfico
ax.plot(range(len(c_transicao)), c_transicao)
ax.plot([-3, 0], [c_ss, c_ss])

# Legendas
ax.set_xlabel('Períodos')
ax.set_ylabel('Estoque de consumo ($c$)')
ax.set_title('Transição de consumo ao longo do tempo')
ax.set_xlim(-3, 25)

plt.show()
```



Ambas quantidade de capital e de consumo crescem. No entanto, o consumo tem um aumento brusco já em $t = 0$ com a mudança de $z = 1$ para $z' = 1.05$, diferente do capital que depende das decisões de investimento realizadas em períodos anteriores à mudança. ■

(f) Refaça os itens (b), (d) e (d) considerando $\sigma = 0.5$ e $\sigma = 2$. O que muda? Dê a intuição dos seus resultados.

A mudança de σ altera o formato a função de utilidade, dada por:

$$U(c) = \frac{c^{1-\sigma} - 1}{1 - \sigma}, \qquad \text{se } \sigma > 0, \sigma \neq 1$$

Para resolver numericamente a função valor, como $\sigma = 0.5$, a função valor com a forma de utilidade acima será dada por:

$$v(k) = \max_{\substack{0 \leq k' \leq zk^{\alpha} + (1-\delta)k \\ k \text{ dado}}} \left\{ \frac{[zk^{\alpha} + (1-\delta)k - k']^{1-\sigma} - 1}{1 - \sigma} + \beta v(k') \right\}. \tag{2.2.9}$$

As únicas alterações são as trocas, no código, do σ e da função de utilidade

```
log(z*k_grid[i]**alpha + (1 - delta)*k_grid[i] - k_grid[j]) + beta*vn[n-1][j]
```

para

```
((z*k_grid[i]**alpha + (1 - delta)*k_grid[i] - k_grid[j])**(1 - sigma) - 1) / (1 -
    sigma) + beta*vn[n-1][j]
```

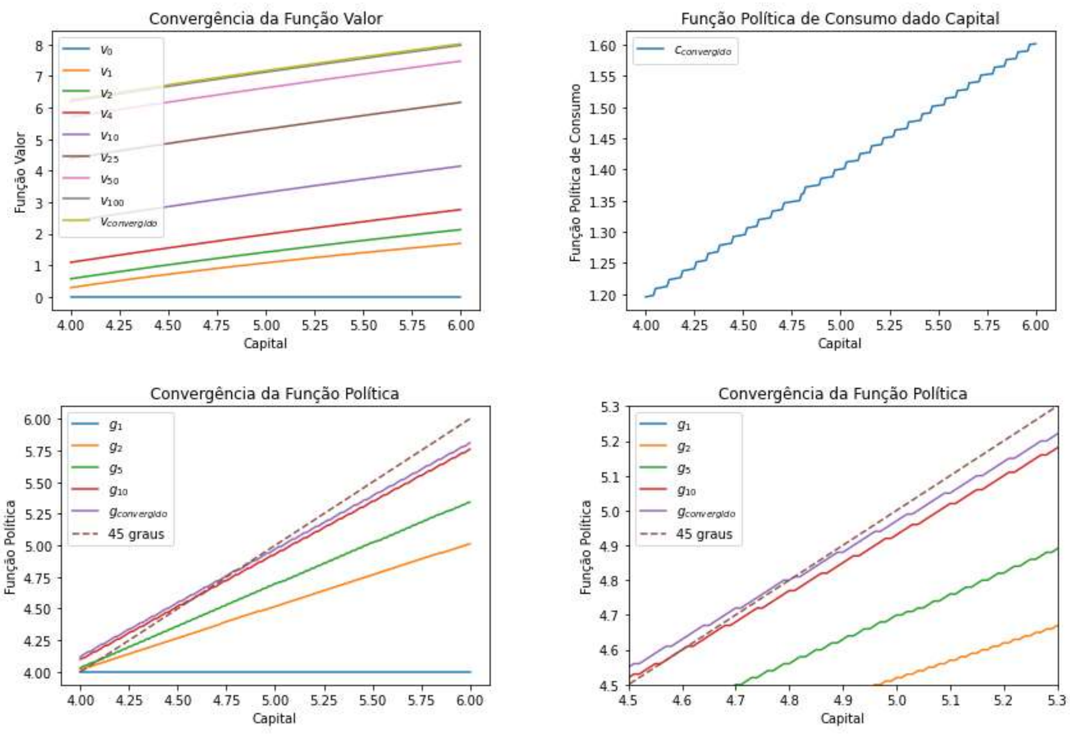
e, portanto, o código não será incluído neste item, apenas os resultados e gráficos.

<u>Caso $\sigma = 0.5$</u>:

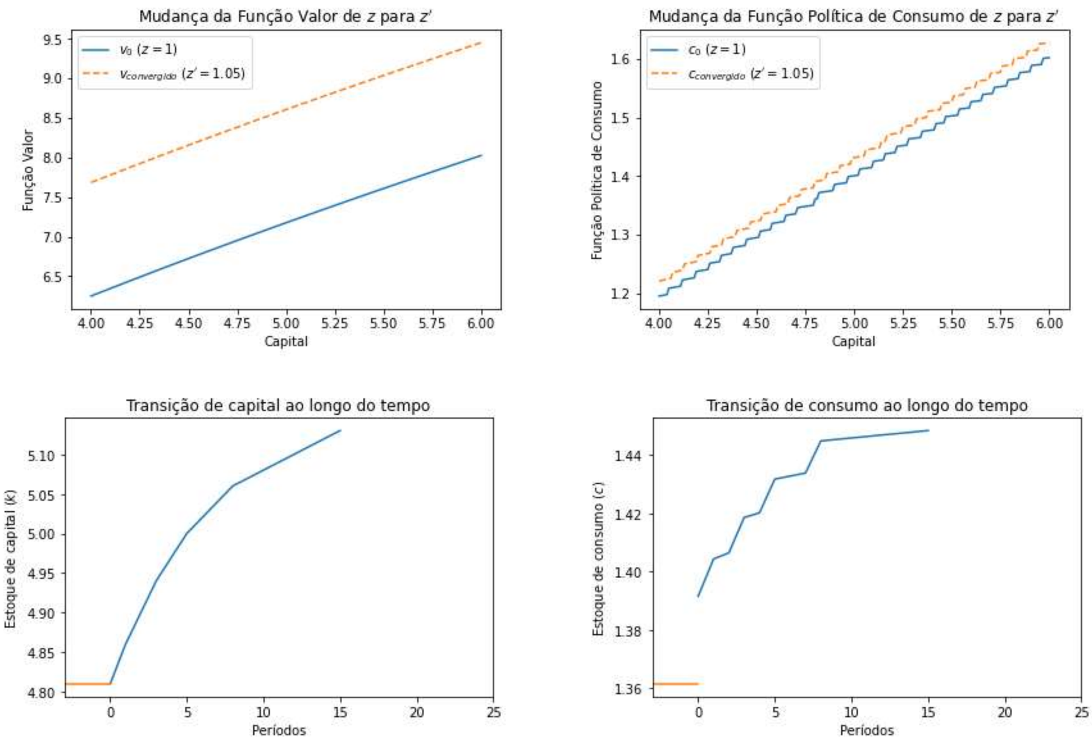
Quando $z = 1$, temos os seguintes resultados:

```
Índice inicial do capital (aleatorizado): 94
Índice do capital estacionário: 81
Capital estacionário k* = 4.81
Consumo estacionário c* = 1.361
```

Note que os resultados, mesmo com a alteração de $\sigma = 1$ para $\sigma = 0.5$, o capital estacionário, k^*, e o consumo estacionário, c^*, permanecem iguais.



Quando $z = 1.05$, temos os seguintes resultados:




```
Índice inicial do capital (do k* anterior): 81
Capital inicial (k* de z = 1) = 4.81
Foram necessários 16 períodos para atingir o novo estado estacionário.
Índice do capital estacionário (novo): 114
Capital estacionário k* (z‘ = 1.05) = 5.14
Consumo estacionário c* = 1.459
```

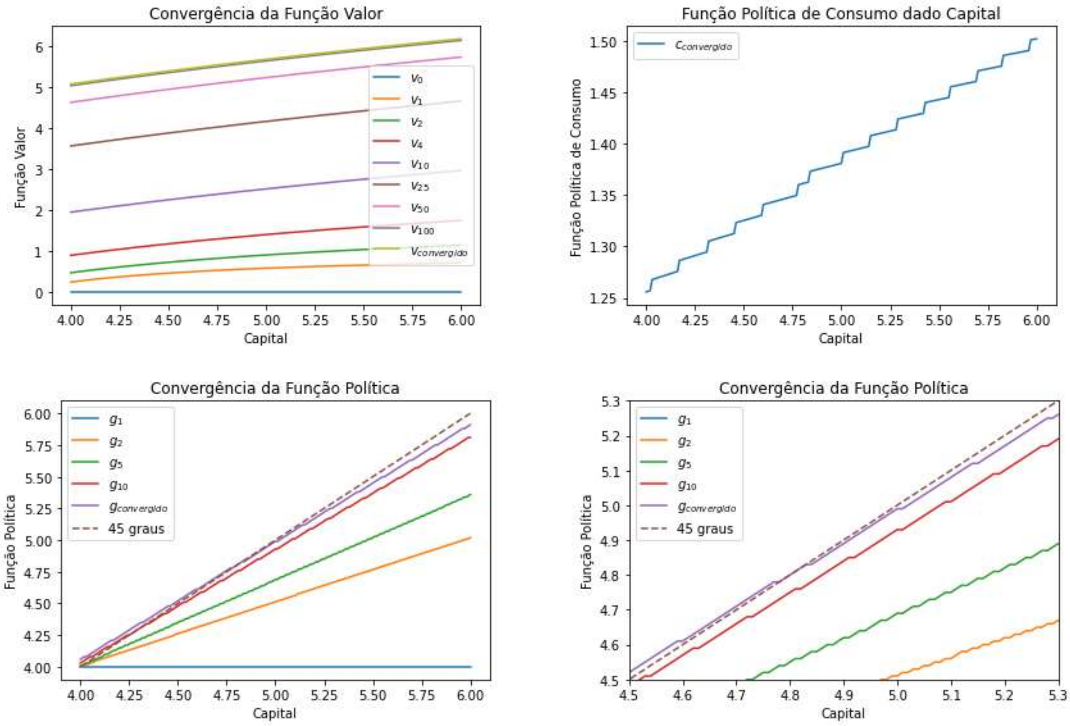
Ao alterar para $z' = 1.05$, os valores no estado estacionário também são os mesmos de quando $\sigma = 1$. Note que, para atingir o novo estado estacionário, foram necessários 16 períodos (ao invés de 20 no caso em que $\sigma = 1$). Intuitivamente, o consumo é altamente *substituível* ao longo do tempo - ou seja, o a elasticidade intertemporal de substituição é relativamente alta: $1/\sigma = 2$. Neste caso, a necessidade de suavização do consumo é baixa, logo, o planejador realiza a transição de maneira mais rápida.

<u>Caso $\sigma = 2$</u>:

Quando $z = 1$, temos os seguintes resultados:

```
Índice inicial do capital (aleatorizado): 74
Índice do capital estacionário: 78
Capital estacionário k* = 4.78
Consumo estacionário c* = 1.36
```

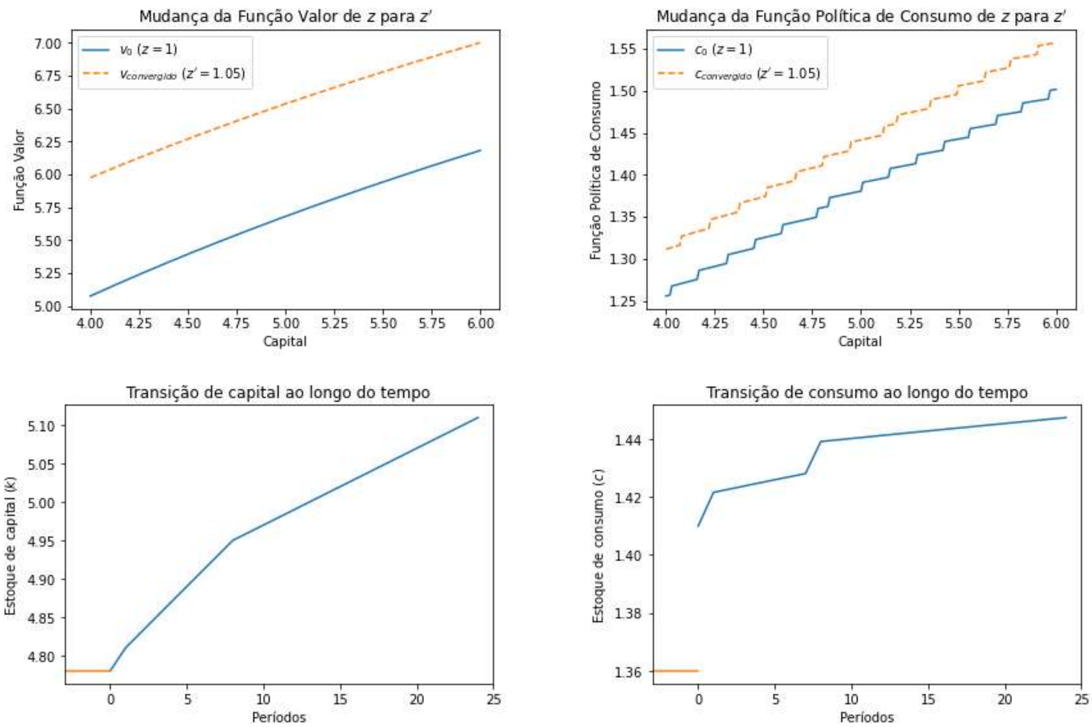
Assim como em $\sigma = 0.5$, os resultados para $\sigma = 2$ são iguais do item anterior ($\sigma = 1$).



Quando $z = 1.05$, temos os seguintes resultados:

```
Índice inicial do capital (do k* anterior): 78
Capital inicial (k* de z = 1) = 4.78
Foram necessários 25 períodos para atingir o novo estado estacionário.
Índice do capital estacionário (novo): 112
Capital estacionário k* (z‘ = 1.05) = 5.12
Consumo estacionário c* = 1.458
```

Note que, para atingir o novo estado estacionário, foram necessários 25 períodos (ao invés de 20 no caso em que $\sigma = 1$). Intuitivamente, o consumo é altamente *complementar* ao longo do tempo - ou seja, o a elasticidade intertemporal de substituição é relativamente baixa: $1/\sigma = 0.5$. Neste caso, a necessidade de suavização do consumo é alta, logo, o planejador realiza a transição em um período maior e a convergência ao estado estacionário demora mais.


Mudança da Função Valor de z para z'
v_0 $(z = 1)$
$v_{convergido}$ $(z' = 1.05)$
Função Valor
Capital
Mudança da Função Política de Consumo de z para z'
c_0 $(z = 1)$
$c_{convergido}$ $(z' = 1.05)$
Função Política de Consumo
Capital
Transição de capital ao longo do tempo
Estoque de capital (k)
Períodos
Transição de consumo ao longo do tempo
Estoque de consumo (c)
Períodos

■