Métodos Numéricos - Lista 02

Samuel de S. Barbosa

Abril 2018

Nesta lista vamos solucionar o modelo de *Real Business Cycles* (RBC) usando diferentes técnicas de iteração da função valor.

Preferências

$$U(c) = \mathbb{E}_0 \sum_{t=0}^{\infty} \beta^t u(c_t)$$

$$u(c_t) = \frac{c^{1-\mu} - 1}{1 - \mu}$$
$$\mu = 1/(1+r)$$

Tecnologia

$$Y_t = z_t F(K_t, N_t) = z_t K_t^{\alpha} N_t^{1-\alpha}$$

$$\log z_t = \rho \log z_{t-1} + \epsilon_t$$

$$\varepsilon_t \sim N(0, \sigma^2)$$

Calibração

$$\beta = 0.987, \quad \mu = 2, \quad \alpha = 0.3, \quad \delta = 0.012, \quad \rho = 0.95, \quad \sigma = 0.007$$

1. Escreva o problema do planejador na forma recursiva

O problema do planejador consiste em maximizar $U(c)$ sujeito à restrição de factibilidade $Y_t = C_t + S_t$ e à lei de movimento do capital $K_{t+1} = (1-\delta)K_t + S_t$.

Podemos escrever este problema na forma recursiva como

$$\begin{aligned} V(k,z) = \max_{k',\,c} \quad & u(c) + \beta \mathbb{E} V(k',z'). \\ \text{s.a.} \quad & c = y - s \\ & k' = (1-\delta)k + s \end{aligned} \tag{1}$$

Equivalentemente,

$$V(k,z) = \max_{k' \geq 0} \quad u(zk^\alpha + (1-\delta)k - k') + \beta \mathbb{E} V(k',z') \tag{2}$$

2. Assuma que não há incerteza, i.e., $\sigma = 0$. Derive a equação de Euler e encontre o capital de estado estacionário k_{ss}.

Da condição de otimalidade e Teorema do Envelope obtemos a eq. de Euler:

$$\begin{aligned} u'(c) = & \quad \beta \mathbb{E}[\frac{\partial}{\partial k'} V(k',z')] \\ = & \quad \beta \mathbb{E}[u'(c')(\alpha z k^{\alpha-1} + 1 - \delta)]. \end{aligned} \tag{3}$$

No estado estacionário, sem incerteza, temos $c' = c = c_{ss}$ e $k' = k = k_{ss}$.

Além disso,

$$\begin{aligned} & \log(z_t) = \rho \log(z_{t-1}) + \epsilon_t \\ \implies & (1-\rho L)\log(z_t) = \epsilon_t = 0 \\ \implies & \log(z_t) = 0 \\ \implies & z_t = 1. \end{aligned} \tag{4}$$

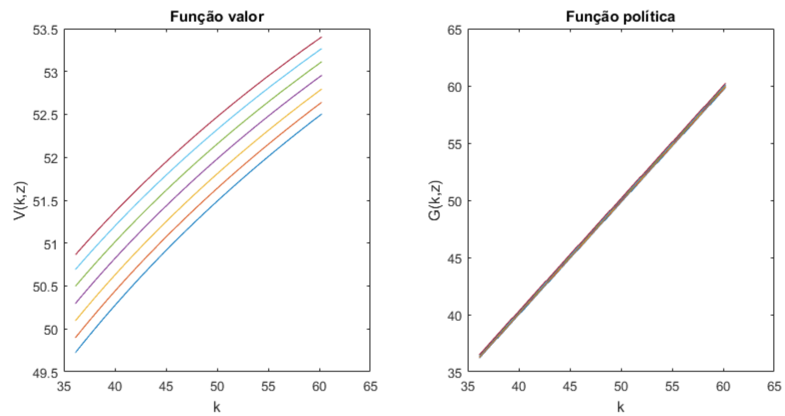
Logo

$$\begin{aligned} 1 = & \quad \beta(\alpha k_{ss}^{\alpha-1} + 1 - \delta) \iff \\ k_{ss}^{1-\alpha} = & \quad \frac{1-\beta(1-\delta)}{\alpha\beta} \iff \\ k_{ss} = & \quad \left[\frac{1-\beta(1-\delta)}{\alpha\beta}\right]^{\frac{1}{\alpha-1}} \end{aligned} \tag{5}$$

3. Resolva o problema utilizando método de iteração da função valor.

Para a choque de TFP, vamos utilizar o método de Tauchen (1986) com 7 pontos. Para o grid de capital, usaremos 500 pontos linearmente espaçados no intervalo $[0.75k_{ss}, 1.25k_{ss}]$.

Utilizando o método da força bruta como benchmark, obtemos a seguinte solução:



O tempo necessário para resolver o problema utilizando a força bruta (de forma vetorizada) foi de 14,73 segundos.

Como a função política é crescente no capital, podemos explorar essa monotonicidade para tentar obter uma solução mais rápida. Dada a primeira iteração no método da força bruta, sabemos que a escolha ótima de capital estará restrita ao intervalo $[k_1, k_{\max}]$. Ao restringir a maximização a este intervalo, obtemos a solução em 21,79 segundos. Embora este método reduza o tempo da solução não vetorizada, implementá-lo no algoritmo vetorizado causou, de fato, uma perda de eficiência.

Em seguida, podemos explorar a concavidade da função valor. A concavidade nos permite obter a solução somente observando a diferença da função valor estimada entre pontos consecutivos no grid. Utilizando esta técnica obtemos a solução em 20,13 segundos, um tempo similar ao obtido explorando-se a monotonicidade, porém não mais eficiente do que a força bruta vetorizada.

4. Refaça o item anterior usando o acelerador. Isto é, só realize a maximização em algumas iterações (10% delas, por exemplo). Compare os resultados com o item anterior.

Utilizando a técnica do acelerador (10%) com o método da força bruta vetorizada, obtemos a solução em 9,50 segundos, reduzindo o tempo em 35,5%, um ganho considerável.

5. Para este item, refaça o problema usando múltiplos grids (multigrid). Primeiro, resolva o problema usando um grid de 100 pontos, depois 500 e, finalmente, 5000. Para cada grid posterior, utilize a solução anterior como chute inicial (você precisará interpolar). Compare com os itens anteriores.

No primeiro passo (100 pontos), utilizando apenas força bruta vetorizada, obtemos convergência da V_1 em 0,28 segundos. Em seguida, utilizando a interpolação de V_1 em um grid de 500 pontos, obtemos V_2 em 7,27 segundos.

No último passo, interpolamos V_2 em um grid de 5000 pontos. Neste passo, obtemos convergência em 46,25 segundos.

No total, obtemos a função valor em um grid com 5000 pontos em aproximadamente 54 segundos.

6. Para este item, resolva o problema usando o método do grid endógeno (endogenous grid method). Compare com os itens anteriores.

O método do grid endógeno convergiu em um segundo. Para verificar os resultados obtidos calculamos os *Euler Errors* em cada estado. O gráfico a seguir mostra os resultados obtidos: