Crescimento Econômico: O Modelo de Solow

Ian Teixeira Barreiro

October 2021

1 Introdução

O que determina a renda e, consequentemente, o bem-estar de um país? Por que alguns países crescem mais e mais rápido que outros? Que papel os fatores de produção (capital e mão de obra) tem nesse crescimento e como podemos desenhar políticas que levem a um maior crescimento a longo prazo? Que papel tem o avanço tecnológico nesse crescimento? Essas e outras perguntas, relativas ao crescimento a longo prazo de uma economia serão respondidas no presente texto a partir de um resumo do Modelo de Crescimento de Solow, tendo como referência o livro texto "Macroeconomia, 8a Edição", do autor N. Gregory Mankiw. O Modelo de Solow apresenta o crescimento como função do estoque de capital, da mão de obra e do avanço tecnológico.

A estrutura do presente texto será a seguinte: i) primeiro descreveremos uma versão simplificada do modelo (em que excluímos a mão de obra e o avanço tecnológico como determinantes do produto) em termos de suas partes constituintes - a oferta ou produto agregado e a demanda agregada; ii) em seguida descreveremos o importante conceito do estado estacionário; iii) descreveremos então a implementação computacional do modelo simplificado descrito, com função de produção Cobb-Douglas, que encontra o estado estacionário da economia; iv) discutiremos então aprofundamentos do modelo simples apresentado, tais como o efeito da poupança no crescimento e a "regra de ouro".

2 O Modelo de Solow

O Modelo de Solow descreve como o estoque de capital, a força de trabalho e o avanço tecnológico, e em especial o crescimento dessas variáveis, interagem para levar ao crescimento do produto agregado a longo prazo. Para descrever o modelo, começaremos com uma economia fechada e sem atuação do governo.

2.1 A Função de Produção

No Modelo de Solow, a produção é uma função do estoque de capital e da força de trabalho, tendo a seguinte forma:

$$Y = F(K, L) \tag{1}$$

Tomamos o produto como tendo efeitos constantes de escala, ou seja, aumentos no estoque de capital e na força de trabalho a uma certa constante levam a crescimentos proporcionais do produto. Sabendo disso temos:

$$\begin{aligned} Y &= F(K, L) \\ Y \cdot 1/L &= F(K/L, 1) \\ y &= f(k) \end{aligned}$$

em que y é o produto por trabalhador e k é o capital por trabalhador. Deste modo, incorporamos a informação sobre o mercado de trabalho junto à variável do estoque de capital, tornando nossa análise independente do tamanho da economia quando medido através do tamanho da força de trabalho. A primeira derivada dessa função mostra o produto marginal da economia dado um acréscimo em k, ou seja, o aumento no produto dado um incremento infinitesimal no estoque de capital. Tomamos a segunda derivada como sendo negativa, para representar que o produto tem retornos marginais decrescentes em relação ao capital - quando k é alto, acréscimos adicionais nessa variável levam a um baixo aumento no produto. Uma função Cobb-Douglas possui as características acima descritas, por isso, nós adotaremos a seguinte função nas análises que seguem:

$$y = k^{1/3} = \sqrt[3]{k}$$

cujo gráfico é:

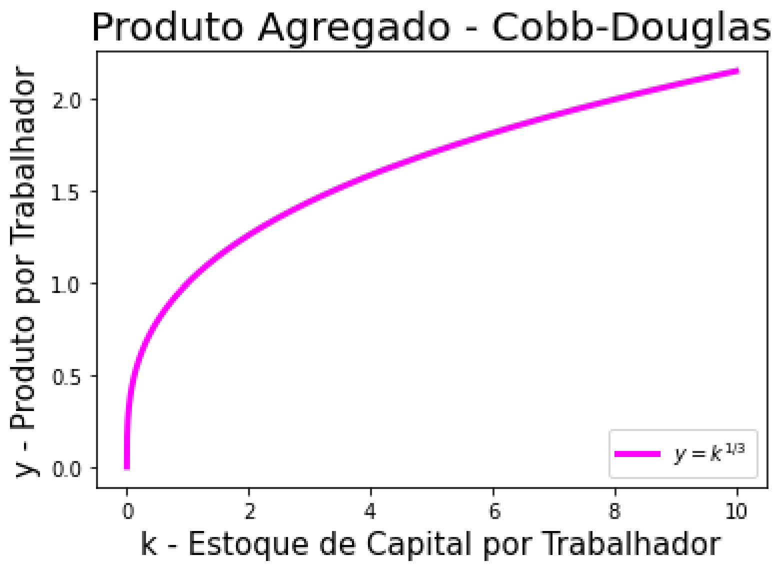

Figura 1: Um produto agregado Cobb-Douglas

2.2 A Função Consumo

Como dissemos anteriormente, nossa economia é fechada e sem gastos do governo. Deste modo, nossa demanda agregada é bastante simples, e compreende apenas o consumo e o investimento agregados. Para facilitar nossa análise, e tendo em vista que a produção está expressa em termos de cada trabalhador, dividiremos nossa função consumo por L para obter a demanda por trabalhador, como está expresso a seguir:

$$Y = C + I$$
$$\frac{Y}{L} = \frac{C}{L} + \frac{I}{L}$$
$$y = c + i$$

em que y é a demanda por trabalhador, c é o consumo por trabalhador e i é o investimento por trabalhador.

Suponhamos que s seja a taxa de poupança, valor entre 0 e 1, que representa a fração da renda poupada na economia. Nesse caso, poderíamos dizer que toda renda não poupada é usada no consumo, e assim expressar o consumo como:

$$c = (1 - s)y$$

e substituir essa equação na equação de demanda agregada obtendo:

$$\begin{aligned} y &= (1 - s)y + i \\ y(1 - 1 + s) &= i \\ i &= sy \\ i &= sf(k) \end{aligned}$$

de modo que agora podemos expressar o investimento na economia em termos de uma função do estoque de capital existente no presente. Tomando $s = 0.3$, e adotando ainda a função de produção Cobb-Douglas obtemos que $i = 0.3k^{1/3}$. A curva dessa função, juntamente com a da função de produção, está registrada na figura 2.

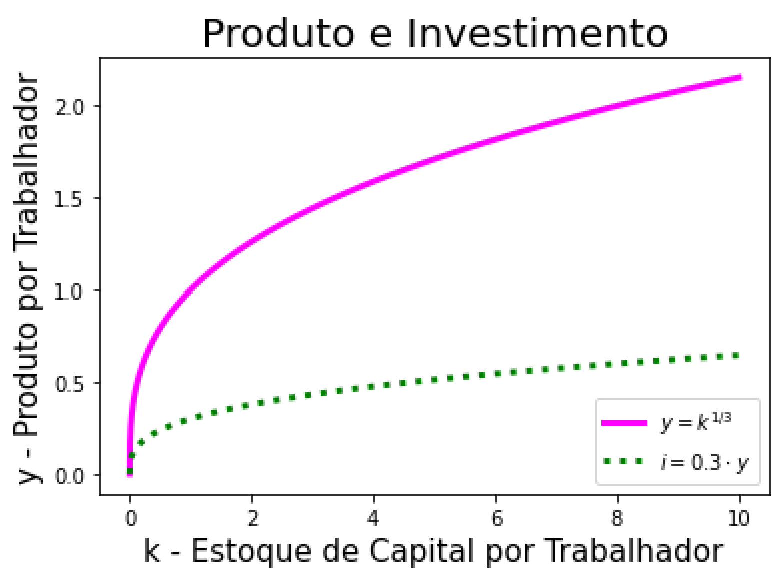

Figura 2: O produto e o investimento da economia

2.3 O Estado Estacionário

Para que possamos abordar o estado estacionário, é necessário acrescentar mais um elemento adicional, que nos ajudará a determinar em cada momento do tempo qual será a variação no estoque de capital: a depreciação. O investimento, abordado anteriormente, leva a um aumento no estoque de capital (estamos excluindo a manutenção de estoques). A depreciação atua deteriorando o capital. Ela expressa o tempo de vida útil a partir do qual o capital se torna inutilizável. No presente exemplo consideraremos que a depreciação atua anualmente a taxas constantes sobre o estoque de capital por trabalhador, de modo que a função depreciação seja linear. Supondo por exemplo uma depreciação de 0.1 teremos as curvas que seguem.

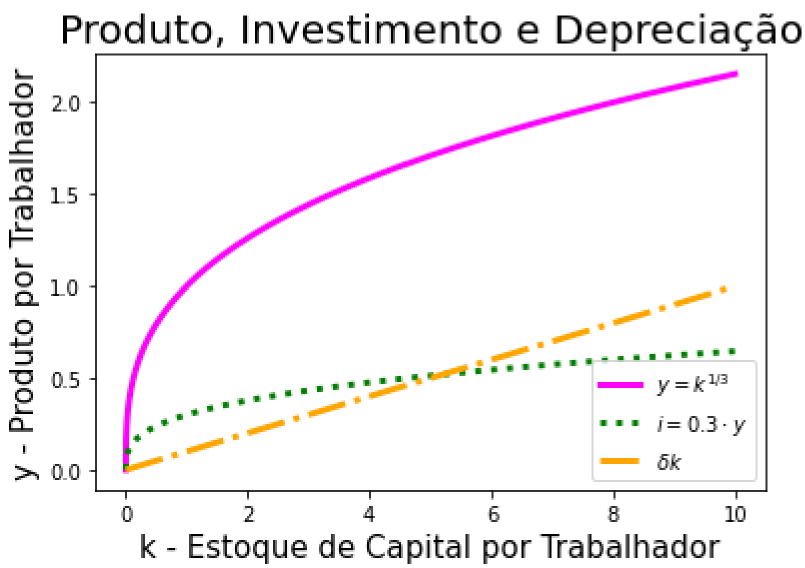

Figura 3: O produto, investimento e a depreciação de uma economia

Como observamos no gráfico, para certos valores de capital por trabalhador, o investimento é superior à depreciação. Devido a isso a variação do estoque de capital será positiva: mais capital é introduzido na economia do que deteriorado. Esse padrão de crescimento do estoque de capital continuará, até o ponto em que a reta de depreciação se encontra com a curva do investimento. Nesse ponto, a variação do estoque de capital é zero: a quantidade de capital novo introduzido na economia é igual à quantidade de capital depreciado. Esse ponto é chamado de estado estacionário, justamente pelo fato de que ao entrar nesse equilíbrio a economia permanecerá nesse ponto, não aumentando o seu estoque de capital. O estado estacionário representa o equilíbrio a longo prazo da economia.

2.4 Implementação Computacional

A seguir apresentamos a implementação computacional do programa que encontra o estado estacionário de uma economia dado uma função produção que depende apenas do estoque de capital.

```
 import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
sns.set

def produto(K, L, exp, K_por_trab = True):

    """
    Uma funcao de producao Cobb-Douglas:
    -------------------------------------
    K           = estoque de capital
    L           = forca de trabalho
    exp         = o expoente da funcao Cobb-Douglas
    K_por_trab = True -> a funcao produto se torna produto por trabalhador.
    Nesse caso recomenda-se usar L = 1.
    -------------------------------------
    """

    if K_por_trab:
        return (K / L) ** exp
    else:
        return (K ** exp) * (L ** exp)

def estado_estacionario(K, exp, s, depr, precisao = 4):

    """
    Encontra o equilibrio estacionario para um modelo com forca de
    trabalho constante e uma funcao de producao Cobb-Douglas.
    -------------------------------------
    K         = estoque de capital
    exp       = o expoente da funcao Cobb-Douglas
```

```
    s        = taxa de poupanca (entre 0 e 1)
    depr     = taxa de depreciacao (entre 0 e 1)
    precisao = a quantas casas decimais se deseja arredondar
    ------------------------------------
    """

    assert s >= 0 and s <= 1
    assert depr >= 0 and depr <= 1

    k_t_mais = K
    delta = s * produto(K, 1, exp) - depr * K

    while round(delta, precisao) != 0:
        k_t_mais = K + delta
        delta = s * produto(k_t_mais, 1, exp) - depr * k_t_mais
        K = k_t_mais

    return round(K, precisao)
```

A primeira função é uma função que nos retorna um produto Cobb-Douglas dados os parâmetros. Caso K_por_trab seja avaliada como True, a função retornará o produto por trabalhador. Nesse caso, como escrito na documentação, recomenda-se o uso de L = 1, ainda que isso não seja necessário.

A função estado_estacionario é a função efetivamente usada para encontrar o estado estacionário da economia, no caso de uma função de produção por trabalhador. Primeiramente verifica-se se a taxa de poupança está entre 0 e 1. Em seguida, o mesmo é feito para a taxa de depreciação. Definimos em seguida uma variável temporária que armazena o que sera o capital no período subsequente ao que está sendo analisado. Delta é definido como sendo a diferença do investimento e do capital depreciado para um dado período. Como dito anteriormente, no estado estacionário o capital investido se iguala ao capital depreciado, de modo que delta seria avaliado para 0. Precisão é um parâmetro que indica a quantas casas decimais queremos arredondas delta no momento de avaliar se este se iguala a 0. Definimos uma iteração por *while* que cessa apenas quando delta arredondado a precisão casas decimais se iguala a 0. Enquanto a condição não for satisfeita, atualizamos o valor do capital no período subsequente, somando o valor atual com o valor de delta (a variação no estoque de capital) e recalculamos delta com base no novo estoque de capital. Quando a iteração cessa, retornamos o valor do capital no ponto estacionário arredondado a precisão casas decimais.

A seguir temos um exemplo do programa acima com os parâmetros utilizados para as funções anteriores, ou seja, uma função de produção Cobb-Douglas com expoente $\frac{1}{3}$, uma taxa de poupança de 0.3 e uma depreciação de 0.1. Rodamos o programa com um capital inicial de K = 6.

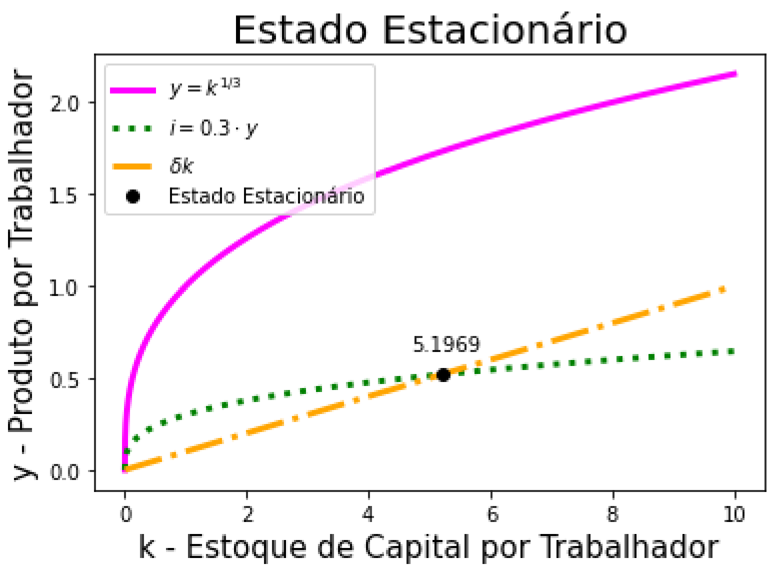

Figura 4: O estado estacionário para o exemplo dado

Como se vê o equilíbrio de longo prazo é alcançado no ponto K = 5.1969, arredondado com precisão = 4.

2.5 A Regra de Ouro

Diferentes níveis de poupança em uma economia levam a diferentes níveis de estoque de capital no estado estacionário. No entanto, não necessariamente níveis maiores de poupança e, portanto, capital, significam um aumento de bem estar da população de um país. Isso por que o bem estar percebido a nível individual muitas vezes está muito mais atrelado ao consumo das famílias do que do nível de investimento das empresas. Desse modo, cabe uma análise de quais níveis de poupança levam a um estado estacionário com maior consumo. A taxa de poupança que leva a economia ao

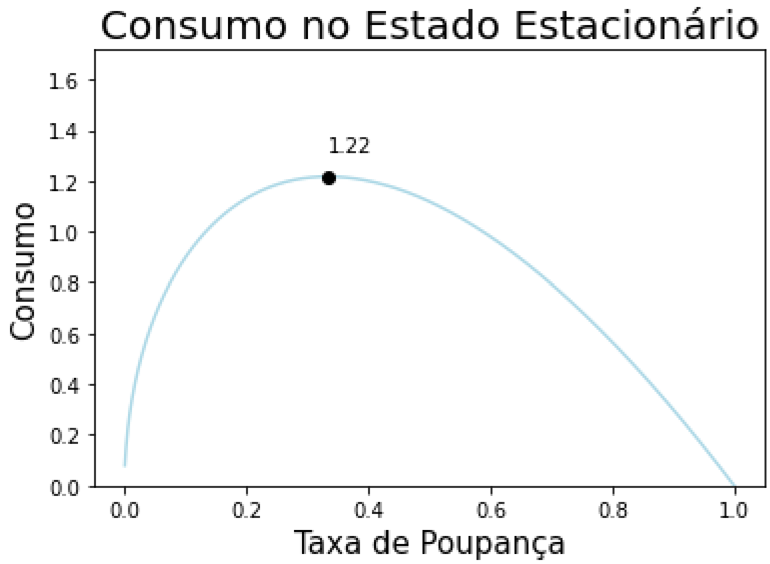

Figura 5: Trajetória do Consumo ao se Variar a Taxa de Poupança

maior nível de consumo possível no estado estacionário determina o chamado nível da regra de ouro na economia.

Dado que estamos em uma economia fechada e sem governo, o consumo pode ser calculado como $c = y - i$. Tomado $y = f(k)$ e que no equilíbrio o investimento se equivale à depreciação, ou seja, $i = \delta k$, temos que $c = f(k) - \delta k$. Assim, caso queiramos tomar uma abordagem numérica baseada em grids para encontrar o ponto da regra de ouro, poderíamos iterar por valores possíveis da taxa de poupança, encontrar os estados estacionários para cada uma dessas taxas e encontrar o consumo para esse estoque de capital, escolhendo então a taxa de poupança que rende o maior consumo como a regra de ouro. Essa abordagem é utilizada no programa que segue. Outra abordagem possível é perceber que no ponto da regra de ouro, a primeira derivada da produção deve ser igual à inclinação da reta de depreciação, já que se a produção for mais inclinada, o consumo crescerá com o aumento do estoque de capital, e caso contrário o consumo diminuirá com o aumento do estoque de capital.

Abaixo está um exemplo da trajetória do consumo para a economia com os parâmetros que indicamos anteriormente.

```
def regra_de_ouro(exp, depr, k_ini = 0.001, st = 1000, tol =
```

```
    4):

    """
    Encontra a taxa de poupanca que maximiza o consumo
    --------------------------------------
    exp    = expoente da cobb-douglas
    depr   = taxa de depreciacao
    k_ini = valor inicial para o estoque de capital
    st     = tamanho do vetor de taxas de poupanca
    tol    = valor de tolerancia para arredondamento (default
   = 4)
    --------------------------------------
    """

    assert depr >= 0 and depr <= 1

    # Cria vetor de taxas de poupanca
    s_vec = np.linspace(0, 1, st)
    c_vec = []
    k_vec = []

    # Encontra o capital no estado estacionario
    # para cada valor de poupanca
    for s in s_vec:

        k = estado_estacionario(k_ini, exp, s, depr)
        c = produto(k, 1, exp) - depr * k
        k_vec.append(k)
        c_vec.append(c)

    # Encontra o consumo maximo e a taxa de
    # poupanca e capital correspondentes
    c_max = max(c_vec)
    s_max = s_vec[np.argmax(c_vec)]
    k_max = k_vec[np.argmax(c_vec)]

    return c_max, s_max, k_max, c_vec, k_vec, s_vec

a = regra_de_ouro(0.5, 0.1)
```

3 Extensão do Modelo

Para a extensão do modelo de Solow analisaremos a situação em que haja crescimento populacional e progresso tecnológico a uma taxa constante. Em termos algorítmicos esse aprofundamento traz poucas complicações. No entanto, em termos teóricos, a extensão do modelo revela como é possível que economias sustentem crescimento constante a longo prazo.

3.1 Modelo com Crescimento Populacional

Suponhamos agora que L, ou seja, o número de trabalhadores na economia cresça a uma taxa n por período. Dessa modo, a todo período novos trabalhadores entrarão na economia. Como explicitado pela nossa análise anterior, já que estávamos lidando com o produto por trabalhador e o capital por trabalhador, ao aumentar o número de trabalhadores, mantendo constante o capital, haverá uma queda progressiva no estoque de capital por trabalhador e, portanto, no produto. Trataremos então o crescimento populacional de um modo semelhante à depreciação, ou seja, o investimento agora deve cobrir tanto o capital deteriorado que não serve mais para a produção quanto prover os novos trabalhadores de capital suficiente para produzirem. Desse modo, a variação no estoque de capital no tempo será $\Delta k = i-(\delta+n)k$. Como se pode ver, o crescimento populacional altera a inclinação da reta que determinava a depreciação de capital no modelo anterior. O estado estacionário ainda se encontra no ponto de intersecção dessa reta com a curva de investimento, ou seja, no ponto que que o investimento supre a depreciação e provê novos trabalhadores de capital, mantendo o estoque de capital por trabalhador constante, ainda que o montante de capital cresça à mesma taxa do crescimento populacional. Abaixo está um exemplo no caso em que o expoente da Cobb-Douglas é $\frac{1}{2}$, a taxa de poupança é 0.4, a taxa de depreciação é 0.1 e a taxa de crescimento populacional é 0.01.

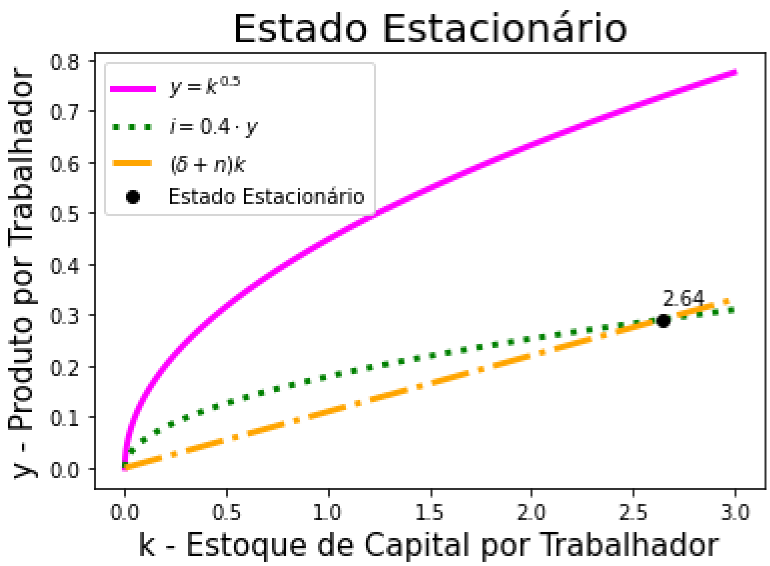

Figura 6: Estado Estacionário com uma Taxa de Crescimento Populacional de n = 0.01

```
# Modelo com crescimento populacional

def estado_estacionario_cresc_pop(exp, s, depr, n, K, L, tol
    = 4):

    """
    Encontra o estado estacionario para uma economia com
    crescimento
    populacional
    -------------------------------------
    exp  = o expoente da funcao cobb-douglas
    s    = a taxa de poupanca (entre 0 e 1)
    depr = a taxa de depreciacao (entre 0 e 1)
    n    = a taxa de crescimento populacional
    K    = o estoque de capital inicial
    L    = o numero de trabalhadores inicial
    -------------------------------------
    """

    assert s >= 0 and s <= 1
    assert depr >= 0 and depr <= 1

    # Inicializa o estoque de capital
```

```
    # do periodo t+1 e a variacao no estoque
    k_t_mais = K
    delta = s * produto(K, L, exp) - (depr + n) * K

    # Enquanto o investimento for diferente da
    # depreciacao, atualiza o estoque de capital
    while round(delta, tol) != 0:

        k_t_mais = K + delta
        delta = s * produto(k_t_mais, L, exp) - (depr + n) * k_t_mais
        K = k_t_mais

    return round(K, tol)
```

Para encontrar a taxa de poupança que maximiza o consumo fazemos o mesmo procedimento do programa anterior, iterando por várias taxas de poupança possíveis e calculando o estado estacionário para essa taxa. Então, escolhemos aquele que provê o maior consumo. Um ponto que diferencia este modelo do anterior é referente ao modo analítico de encontrar a regra de ouro. Agora devemos procurar o ponto em que o produto marginal se iguala à soma da depreciação com a taxa de crescimento populacional para encontrar a poupança maximizadora do consumo.

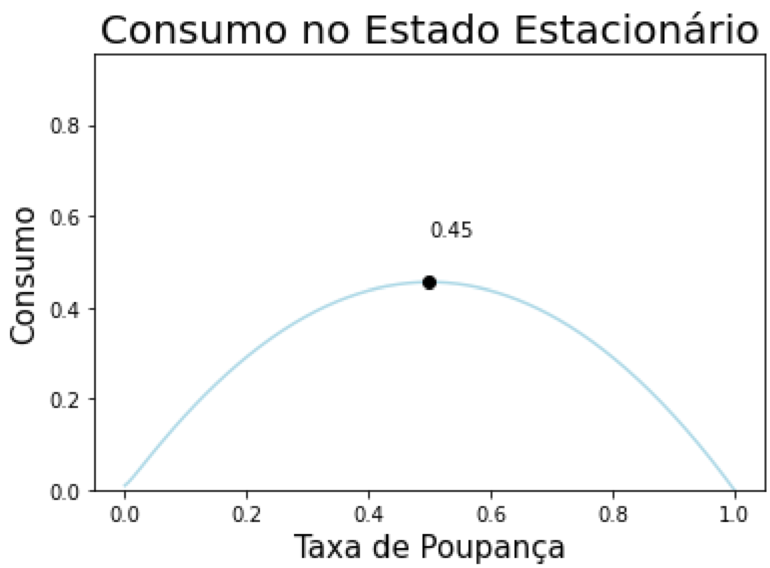

Figura 7: Trajetória do Consumo em uma Economia com Crescimento Populacional

```python
def regra_de_ouro_cresc_pop(exp, depr, n, k_ini, l_ini, st =
    1000, tol = 4):

    """
    Encontra a taxa de poupanca que maximiza o consumo para
    uma economia com crescimento populacional
    ---------------------------------------
    exp = o expoente da funcao cobb-douglas
    depr = a taxa de depreciacao (entre 0 e 1)
    n = a taxa de crescimento populacional
    k_ini = valor inicial para o estoque de capital
    l_ini = valor inicial para o numero de trabalhadores
    st = tamanho do vetor de taxas de poupanca
    ---------------------------------------
    """

    assert depr >= 0 and depr <= 1

    # Criando o vetor de poupancas
    s_vec = np.linspace(0, 1, st)
    k_vec = []
    c_vec = []

```

```
    # Encontra o capital no estado estacionario
    # para cada valor de poupanca
    for s in s_vec:

        k = estado_estacionario_cresc_pop(exp, s, depr, n, k_ini, l_ini)
        c = produto(k, l_ini, exp) - (depr + n) * k
        k_vec.append(k)
        c_vec.append(c)

    c_max = max(c_vec)
    k_max = k_vec[np.argmax(c_vec)]
    s_max = s_vec[np.argmax(c_vec)]

    return c_max, s_max, k_max, c_vec, k_vec, s_vec
```

3.2 Modelo com Crescimento Populacional e Progresso Tecnológico

Até agora consideramos que a tecnologia que determinava a produção era constante. No entanto, sabemos que isso não é verdade na prática e que existe progresso tecnológico nas economias. Modelaremos esse fenômeno tomando o parâmetro E como a eficiência de cada trabalhador. Desse modo, progressos tecnológicos podem ser expressos como aumentos de eficiência de cada trabalhador, fazendo com que seja possível produzir o mesmo com menos trabalhadores. Passaremos a fazer nossa analise pautado no produto e no capital por trabalhador eficiente, ou seja $\frac{Y}{L \cdot X}$ e $\frac{K}{L \cdot X}$. Caso a eficiência por trabalhador cresça, mantendo o estoque de capital constante, haverá uma queda na quantidade de capital por trabalhador eficiente, de modo semelhante ao que ocorre na situação do aumento populacional. Desse modo, expressando o crescimento da eficiência pelo parâmetro g, podemos tratar o progresso tecnológico de modo semelhante ao crescimento populacional. A variação do capital por trabalhador eficiente será expresso do seguinte modo $\Delta k = i - (\delta + n + g)k$. O estado estacionário será no ponto em que o investimento supre capital suficiente para repor a depreciação e cada novo trabalhador eficiente. Segue abaixo o estado estacionário de uma economia em que o expoente da Cobb-Douglas é meio, a taxa de poupança é 0.3, a taxa de depreciação é 0.1, a taxa de crescimento populacional é 0.01 e a taxa de progresso tecnológico é 0.02.

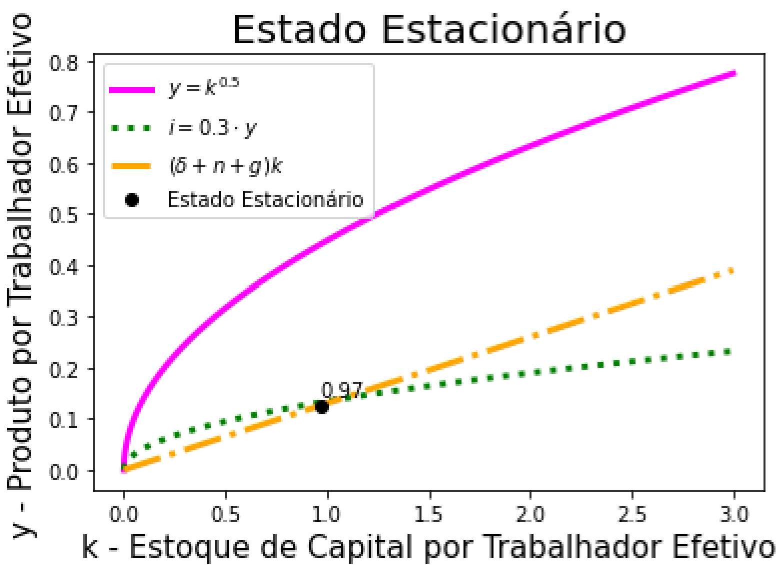

Figura 8: Estado Estacionário com uma Taxa de Crescimento Populacional de n = 0.01 e uma Taxa de Progresso Tecnológico de g = 0.02

```python
# Modelo com tecnologia

def produto_com_tech(K, L, E, exp, k_por_trab_ef = True):

    if k_por_trab_ef:

        return (K / (L * E)) ** exp

    else:

        return K ** exp * L ** exp * E ** exp

def estado_estacionario_tech(exp, s, depr, n, g, K, L, E, tol = 4):

    assert s >= 0 and s <= 1
    assert depr >= 0 and depr <= 1

    k_t_mais = K
    delta = s * produto_com_tech(K, L, E, exp) - (depr + n + g) * K

```

```
    while round(delta, tol) != 0:

        k_t_mais = K + delta
        delta = s * produto_com_tech(k_t_mais, L, E, exp) - (depr + n + g) * K
        K = k_t_mais

    return round(K, tol)
```

A taxa de poupança que leva ao consumo máximo é encontrado da mesma maneira que nas funções anteriores.

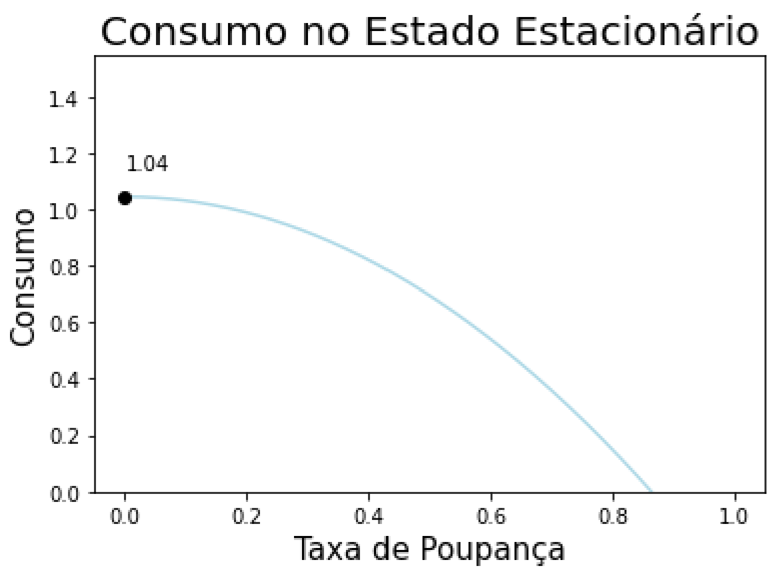

Figura 9: Trajetória do Consumo para uma Economia com Crescimento Populacional e Progresso Tecnológico

```
def regra_de_ouro_tech(exp, depr, n, g, k_ini, l_ini, e_ini,
    st = 1000):

    assert depr >= 0 and depr <= 1

    s_vec = np.linspace(0, 1, st)
    k_vec = []
    c_vec = []

    for s in s_vec:
```

```python

        k = estado_estacionario_tech(exp, s, depr, n, g, k_ini, l_ini, e_ini)
        c = produto_com_tech(k_ini, l_ini, e_ini, exp) - (depr + n + g) * k
        k_vec.append(k)
        c_vec.append(c)

    c_max = max(c_vec)
    k_max = k_vec[np.argmax(c_vec)]
    s_max = s_vec[np.argmax(c_vec)]

    return c_max, s_max, k_max, c_vec, k_vec, s_vec
```